A Fast, Versatile, and User-friendly Plugin for Kernel Density Analysis

Tsz Nam Chan*¶, Bojian Zhu†¶, Leong Hou U‡, Dingming Wu*‖, Wei Tu§, Jianliang Xu†
*College of Computer Science and Software Engineering, Shenzhen University
{edisonchan, dingming}@szu.edu.cn
†Department of Computer Science, Hong Kong Baptist University
{csbjzhu, xujl}@comp.hkbu.edu.hk
‡Department of Computer and Information Science, University of Macau
ryanlhu@um.edu.mo
§School of Architecture and Urban Planning, Shenzhen University
tuwei@szu.edu.cn

Abstract—**Kernel density analysis (KDA) is a commonly used tool in various application domains. Some representative examples include disease outbreak detection and crime/traffic/traffic accident hotspot detection. However, KDA is a slow operation, which cannot scale to large resolution sizes and large-scale datasets. In addition, domain experts also adopt the more complicated variants of KDA, e.g., spatiotemporal kernel density analysis (STKDA) and network kernel density analysis (NKDA), which deteriorates the efficiency issues for analysis. Furthermore, existing software packages cannot simultaneously support KDA, STKDA, and NKDA and some of them require the prior programming skills, which may not be equipped by domain experts. In order to address the above efficiency, versatility, and usability issues, we develop the QGIS plugin, called Fast Density Analysis, which is based on our complexity-reduced algorithms for supporting KDA, STKDA, and NKDA. With the concise user interface, users do not need to have any prior programming skills for using this plugin. In this demonstration, we offer three location datasets (with up to 8.32 million data points) for participants to (1) compare the efficiency between this plugin and existing software packages and (2) conduct case studies to discover hotspots/hidden patterns in these datasets.**

I. INTRODUCTION

Kernel density analysis (KDA) [13], [15] is deemed to be an accurate and important tool for analyzing location datasets (especially for hotspot visualization), which has often been used in different application domains. Epidemiologists [17] adopt this tool to analyze disease outbreaks in different cities and countries (e.g., COVID-19 cases in Hong Kong). Crime and traffic experts [18] also adopt this tool to analyze crime and traffic/traffic accident hotspots, respectively, in order to understand the underlying phenomena of these geographical events. Yet, KDA is a computationally expensive tool, which has already been complained by many domain experts for more than two decades [15], [16]. The main reason is that most of the existing methods still suffer from high time complexity for supporting this tool. Worse still, domain experts further adopt the more complicated variants of KDA, including spatiotemporal kernel density analysis (STKDA) [17] and network kernel density analysis (NKDA) [18], which aim to discover spatiotemporal hotspots and hotspots that are on/align with a road network, respectively, leading to even higher time complexities for analysis. Furthermore, there is currently no all-in-one software package that can simultaneously support these three tools. In addition, some software packages that can support these tools also require the prior programming skills (e.g., python and R), which can act as the barrier for some domain experts who do not acquire these skills.

To overcome the efficiency, versatility, and usability issues of using KDA, STKDA, and NKDA, our preliminary research studies have successfully proposed three complexity-reduced algorithms, called sweep-line algorithm (SLAM) [13], prefix-matrix-based algorithm (PREFIX) [11], and aggregate distance augmentation algorithm (ADA) [12], respectively. Furthermore, we integrate these algorithms into a new QGIS plugin, namely Fast Density Analysis [7], for supporting all these tools simultaneously in this work. Since this plugin is used in QGIS, domain experts can interact with the easy-to-use user interface for using these three tools, without prior programming skills. With the low time complexities of these algorithms, this plugin can achieve significant speedups compared with the corresponding tools from other commonly used software packages, including QGIS [9], ArcGIS [5], PySAL [2] (a python package), spatstat [3] (an R package), spNetwork [4] (an R package). Although we have already incorporated SLAM and ADA into the python libraries, LIBKDV [10] and PyNKDV [14], respectively, using these two libraries requires the python programming skills and each of these libraries cannot support all three tools. Table I compares different software packages. Observe that Fast Density Analysis can address efficiency, versatility, and usability issues. Therefore, this plugin has received significant attention in the GIS community. **Until 1st February 2026, the number of downloads of this plugin has already reached 19,799.**

In this paper, we discuss the technical overview and user interface of our plugin in Sections II and III, respectively. In Section IV, we provide the demonstration plan.

¶ Tsz Nam Chan and Bojian Zhu are the co-first authors of this paper.
‖ Dingming Wu is the corresponding author of this paper.

TABLE I: Comparisons of software packages for supporting KDA, STKDA, and NKDA, where n.a. denotes that the software package cannot support the corresponding tool.

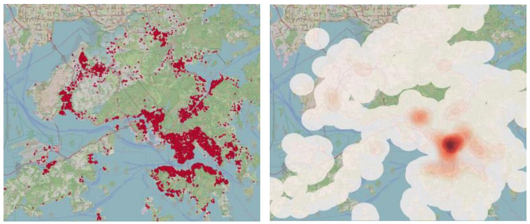
Software package	KDA	STKDA	NKDA	Prior skill
QGIS [9]	slow	n.a.	n.a.	no
ArcGIS [5]	slow	n.a.	n.a.	no
PySAL [2]	slow	n.a.	n.a.	python
spatstat [3]	slow	n.a.	n.a.	R
spNetwork [4]	n.a.	n.a.	slow	R
LIBKDV [10]	fast	medium	n.a.	python
PyNKDV [14]	n.a.	n.a.	fast	python
Fast Density Analysis (ours)	fast	fast	fast	no

II. TECHNICAL OVERVIEW

We first formally define the KDA, STKDA, and NKDA problems in Section II-A. Then, we briefly discuss the corresponding state-of-the-art solutions, which are SLAM (for KDA) [13], PREFIX (for STKDA) [11], and ADA (for NKDA) [12], in Sections II-B, II-C, and II-D, respectively.

A. Problem Definitions

To generate a hotspot map for a location dataset (see the red data points in Figure 1a) based on KDA, we need to color each pixel by computing its kernel density value (see Figure 1b), which is formally defined in Problem 1.



(a) Hong Kong COVID-19 cases (b) Hotspot map

Fig. 1: Generating a hotspot map (based on KDA) for the Hong Kong COVID-19 location dataset.

Problem 1: (KDA [13]) Given a spatial dataset $P = \{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_n\}$ with size n and a plane with $X \times Y$ resolution size, we need to compute the kernel density value $\mathcal{F}_P(\mathbf{q})$ (see Equation 1) for each pixel $\mathbf{q}$.

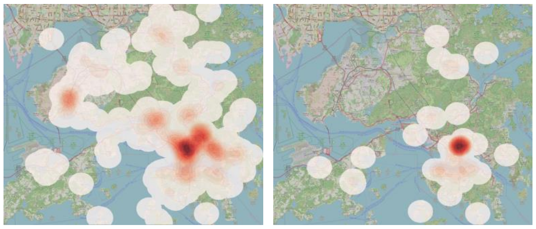
$$\mathcal{F}_P(\mathbf{q}) = \frac{1}{|P|} \sum_{\mathbf{p} \in P} K_{\text{space}}(\mathbf{q}, \mathbf{p}) \tag{1}$$

where $K_{\text{space}}(\mathbf{q}, \mathbf{p})$ denotes the spatial kernel (see Table II).

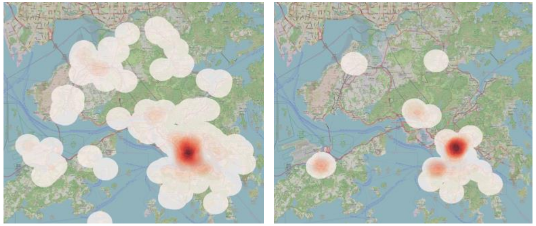
TABLE II: Some commonly used spatial kernels in QGIS and ArcGIS, where $d(\mathbf{q}, \mathbf{p})$ and b_σ denote the Euclidean distance and the spatial bandwidth, respectively.

Kernel	$K_{\text{space}}(\mathbf{q}, \mathbf{p})$
Epanechnikov	$\begin{cases} 1 - \frac{1}{b_\sigma^2} d(\mathbf{q}, \mathbf{p})^2 & \text{if } d(\mathbf{q}, \mathbf{p}) \leq b_\sigma \\ 0 & \text{otherwise} \end{cases}$
Quartic	$\begin{cases} (1 - \frac{1}{b_\sigma^2} d(\mathbf{q}, \mathbf{p})^2)^2 & \text{if } d(\mathbf{q}, \mathbf{p}) \leq b_\sigma \\ 0 & \text{otherwise} \end{cases}$

Note that various geographical phenomena depend on the occurrence time of each event (e.g., different waves of COVID-19 cases). As such, domain experts aim to generate time-dependent hotspot maps based on STKDA (see Figure 2), which is defined in Problem 2.



(a) 26th July 2020 (b) 28th October 2020



(c) 29th January 2021 (d) 2nd May 2021

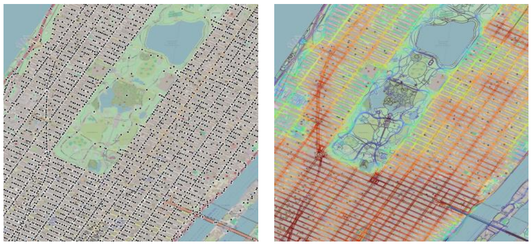
Fig. 2: Generating time-dependent hotspot maps with respect to different timestamps based on STKDA for the Hong Kong COVID-19 location dataset (in Figure 1a).

Problem 2: (STKDA [11]) Given a spatiotemporal dataset $\widehat{P} = \{(\mathbf{p}_1, t_{\mathbf{p}_1}), (\mathbf{p}_2, t_{\mathbf{p}_2}), ..., (\mathbf{p}_n, t_{\mathbf{p}_n})\}$ with size n, T timestamps, and a plane with $X \times Y$ resolution size, we need to compute the spatiotemporal kernel density value $\mathcal{F}_{\widehat{P}}(\mathbf{q}, t_i)$ for each pixel-timestamp pair $(\mathbf{q}, t_i)$ (see Equation 2).

$$\mathcal{F}_{\widehat{P}}(\mathbf{q}, t_i) = \frac{1}{|\widehat{P}|} \sum_{(\mathbf{p}, t_\mathbf{p}) \in \widehat{P}} K_{\text{space}}(\mathbf{q}, \mathbf{p}) \cdot K_{\text{time}}(t_i, t_\mathbf{p}) \tag{2}$$

where $K_{\text{time}}(t_i, t_\mathbf{p})$ denotes the temporal kernel (by replacing $\mathbf{q}$, $\mathbf{p}$, and b_σ with t_i, $t_\mathbf{p}$, and b_τ (the temporal bandwidth), respectively, in Table II).

In addition, some geographical events (e.g., traffic accidents) are mainly on/along with road networks. To accurately analyze hotspots for these events, we need to generate a hotspot map on a road network based on NKDA (see Figure 3), which is defined in Problem 3.



(a) New York traffic accidents (b) Hotspot map on a road network

Fig. 3: Generating a hotspot map (based on NKDA) on a road network for the New York traffic accident dataset.

Problem 3: (NKDA [12]) Given a road network $G = (V, E)$ and a location dataset P with size n that is mapped on G, we need to compute the network kernel density value $\mathcal{F}_P^{(G)}(\mathbf{q})$ (see Equation 3) for each lixel $\mathbf{q}$ (i.e., the center position of the equal-size line segment on each edge).

$$\mathcal{F}_P^{(G)}(\mathbf{q}) = \frac{1}{|P|} \sum_{\mathbf{p} \in P} K_G(\mathbf{q}, \mathbf{p}) \tag{3}$$

where $K_G(\mathbf{q}, \mathbf{p})$ is the kernel based on the shortest path distance (i.e., replace d by d_G in Table II).

B. SLAM for KDA

To efficiently solve the KDA problem, our preliminary work [13] first discovers that the evaluation of $\mathcal{F}_P(\mathbf{q})$ (see Equation 1) for each row of pixels (e.g., blue pixels in Figure 4) can be converted to an interval stabbing problem. Based on this, we then further develop SLAM for solving this problem in $O(X+n)$ time. Since there are Y rows in total, the time complexity of SLAM is $O(Y(X+n))$.

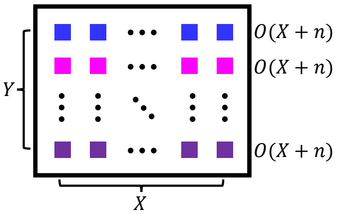


Fig. 4: An overview of SLAM.

C. PREFIX for STKDA

To efficiently solve the STKDA problem, our preliminary work [11] formally shows that $\mathcal{F}_{\widehat{P}}(\mathbf{q}, t_i)$ (see Equation 2) can be represented by each statistical matrix $S^{(u)}_{W(t_i)}(\mathbf{q})$ with each u (see the blue matrix in Figure 5).

$$S^{(u)}_{W(t_i)}(\mathbf{q}) = \sum_{(\mathbf{p}, t_{\mathbf{p}}) \in W(t_i)} t_{\mathbf{p}}^u \cdot K_{\text{space}}(\mathbf{q}, \mathbf{p}) \tag{4}$$

where $W(t_i)$ and u denote the set of data points $(\mathbf{p}, t_{\mathbf{p}})$ that are inside the window $[t_i - b_\tau, t_i + b_\tau]$ (e.g., green circles) and a positive integer, respectively. Note that u depends on the kernel (e.g., $u = 0, 1, 2$ for the Epanechnikov kernel).

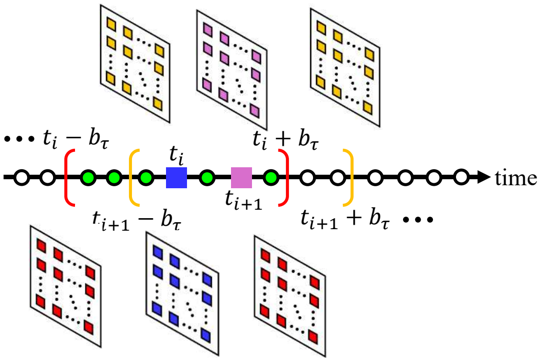


Fig. 5: An overview of PREFIX.

With this concept, we further develop the solution, called PREFIX, which first utilizes SLAM to compute the statistical matrix $S^{(u)}_{\widehat{P}(t)}(\mathbf{q})$ (by replacing $W(t_i)$ with $\widehat{P}(t)$ in Equation 4) for the prefix set $\widehat{P}(t)$ in each end position t of the window (e.g., $t_i - b_\tau$ in Figure 5), where

$$\widehat{P}(t) = \{(\mathbf{p}, t_{\mathbf{p}}) \in \widehat{P} : t_{\mathbf{p}} \leq t\} \tag{5}$$

Based on these statistical matrices $S^{(u)}_{\widehat{P}(t)}(\mathbf{q})$ (see the red and orange matrices), we can compute $S^{(u)}_{W(t_i)}(\mathbf{q})$ (see the blue and pink matrices) based on the following equation.

$$S^{(u)}_{W(t_i)}(\mathbf{q}) = S^{(u)}_{\widehat{P}(t_i + b_\tau)}(\mathbf{q}) - S^{(u)}_{\widehat{P}(t_i - b_\tau)}(\mathbf{q}) \tag{6}$$

Our preliminary work further shows that PREFIX can achieve $O(XYT + Yn)$ time for solving the STKDA problem.

D. ADA for NKDA

In [12], we reveal that once we have augmented the aggregate terms $a^{(deg)}_{P(\mathbf{u},\mathbf{p})}$ and $a^{(deg)}_{P(\mathbf{v},\mathbf{p})}$ (see Equation 7) in advance for the data point $\mathbf{p}$ on each edge $e = (\mathbf{u}, \mathbf{v})$ in G (see Figure 6), we can efficiently compute $\mathcal{F}^{(G)}_P(\mathbf{q})$.

$$a^{(deg)}_{P(\mathbf{u},\mathbf{p})} = \sum_{\mathbf{p} \in P(\mathbf{u},\mathbf{p})} d_G(\mathbf{u}, \mathbf{p})^{deg} \text{ and } a^{(deg)}_{P(\mathbf{v},\mathbf{p})} = \sum_{\mathbf{p} \in P(\mathbf{v},\mathbf{p})} d_G(\mathbf{v}, \mathbf{p})^{deg} \tag{7}$$

where deg is the positive integer, which is based on the kernel (e.g., $deg = 0, 1, 2$ for the Epanechnikov kernel).

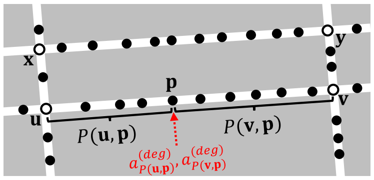


Fig. 6: An overview of ADA.

We further show that the time complexity for solving the NKDA problem based on ADA is $O\big(|E|\big(T_{\text{SP}} + L \log\big(\frac{n}{|E|}\big)\big)\big)$, where L and T_{SP} denote the number of lixels and the time complexity of the shortest path algorithm, respectively.

III. USER INTERFACE

Figure 7 illustrates the user interface (UI) of Fast Density Analysis so that users can call those fast algorithms in Section II-B, Section II-C, and Section II-D to compute KDA, STKDA, and NKDA, respectively.

KDA. Observe from Figure 7a that users need to provide the dataset (stored in a csv file) in the "Input point layer" box and specify the longitude and latitude attribute names (e.g., lon and lat, respectively) in the dataset. In addition, they also need to set the number of row pixels X, the number of column pixels Y, and the spatial bandwidth b_σ in the "Width", "Height", and "Spatial bandwidth (meters)" boxes, respectively.

STKDA. Figure 7b shows the UI of STKDA, which is similar to the one of KDA. The main differences are that users also need to provide the temporal attribute name (e.g., day_idx) in the "Time" box, the number of timestamps T (e.g., 32) in the "Time-axis" box, and the temporal bandwidth in "Temporal bandwidth (days)" box. In addition, users can also specify the starting and ending dates for the time-dependent hotspot maps of STKDA in the "Start" and "End" boxes, respectively.

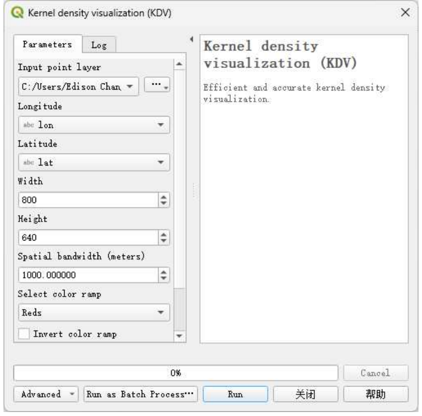
NKDA. In Figure 7c, users need to provide the dataset[1] in the "Input point layer" box and set the bandwidth and the lixel length (i.e., the length of $\mathbf{q}$) in the "Bandwidth (meters)" and "Lixel size (meters)" boxes, respectively.

IV. DEMONSTRATION PLAN

We will use three datasets (with up to 8.32 million data points), which are (1) Hong Kong COVID-19 dataset [1], (2) New York traffic accident dataset [8], and (3) Chicago crime dataset [6], for conducting the following demonstrations to compare the efficiency of Fast Density Analysis with those software packages in Table I. In addition, we also conduct case studies for using our plugin.

---

[1] The format is the same as the one in KDA. Note that this plugin extracts the corresponding road network and maps those data points on it.

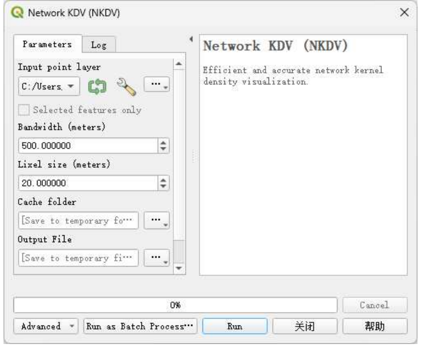


Input point layer: C:/Users/Edison Chan
Longitude: lon
Latitude: lat
Width: 800
Height: 640
Spatial bandwidth (meters): 1000.000000
Select color ramp: Reds
☐ Invert color ramp

Kernel density visualization (KDV)

Efficient and accurate kernel density visualization.

(a) KDA

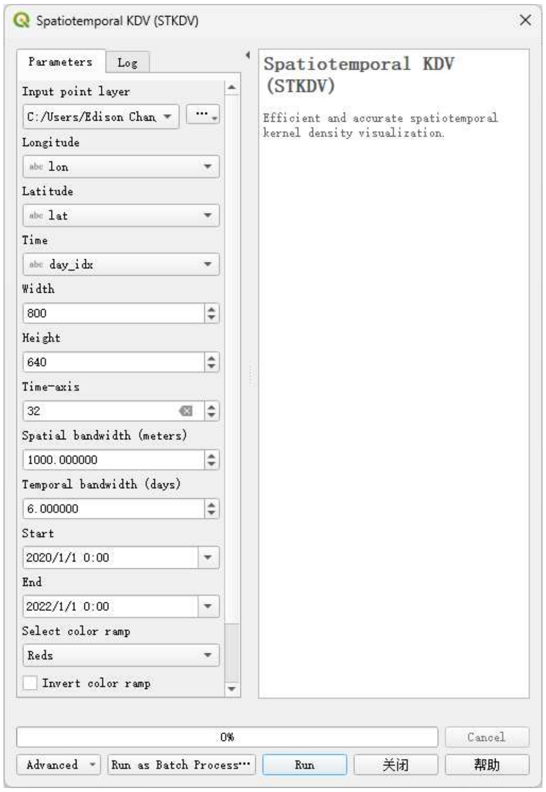


Input point layer: C:/Users.
☐ Selected features only
Bandwidth (meters): 500.000000
Lixel size (meters): 20.000000
Cache folder: [Save to temporary fo…
Output File: [Save to temporary fi…

Network KDV (NKDV)

Efficient and accurate network kernel density visualization.

(c) NKDA



Input point layer: C:/Users/Edison Chan
Longitude: lon
Latitude: lat
Time: day_idx
Width: 800
Height: 640
Time-axis: 32
Spatial bandwidth (meters): 1000.000000
Temporal bandwidth (days): 6.000000
Start: 2020/1/1 0:00
End: 2022/1/1 0:00
Select color ramp: Reds
☐ Invert color ramp

Spatiotemporal KDV (STKDV)

Efficient and accurate spatiotemporal kernel density visualization.

(b) STKDA

Fig. 7: The user interface of Fast Density Analysis for supporting KDA, STKDA, and NKDA.

Varying the resolution sizes. In KDA, STKDA, and NKDA, we will choose multiple resolution sizes $X \times Y$, $X \times Y \times T$, and the lixel size, respectively, for testing the response time of each software package. As an example, we will choose four resolution sizes, 160×120, 320×240, 640×480, and 1280×960, for the demonstration of KDA.

Varying the bandwidths. Recall that we need to set the bandwidths (which can significantly affect the visual quality) for KDA, STKDA, and NKDA. Therefore, we will vary them in this demonstration. As an example, we will choose 250m, 500m, 1000m, 2000m, 4000m for the demonstration of KDA.

Case studies. We will adopt different approaches for using our plugin to discover hotspots/hidden patterns in these three datasets, including (1) the exploration of hotspots in different geographical regions (e.g., investigate traffic accident hotspots in the Manhattan region of New York) and (2) the exploration of the reasonable visualization with different bandwidths.

ACKNOWLEDGEMENT

This work was supported by the NSFC grants 23IAA00610, 62572326, 62202401, 62372308, 42471496, Scientific Foundation for Youth Scholars of Shenzhen University, the Science and Technology Development Fund Macau SAR (0003/2023/RIC, 0052/2023/RIA1, 0011/2025/RIC, 001/2024/SKL for SKL-IOTSC), Shenzhen-Hong Kong-Macau Science and Technology Program Category C (SGDX20230821095159012), and RGC RIF R1015-23.

REFERENCES

[1] Common Spatial Data Infrastructure. https://portal.csdi.gov.hk/csdi-webpage/.
[2] Python spatial analysis library. https://pysal.org/.
[3] Spatstat analysing spatial point patterns. https://spatstat.org/.
[4] spnetwork. https://jeremygelb.github.io/spNetwork/.
[5] ArcGIS. https://www.arcgis.com/index.html, 2025.
[6] Chicago data portal. https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-Present/ijzp-q8t2/about_data, 2025.
[7] Fast Density Analysis. https://plugins.qgis.org/plugins/fast_kernel_density_analysis/, 2025.
[8] NYC open data. https://data.cityofnewyork.us/Public-Safety/Motor-Vehicle-Collisions-Crashes/h9gi-nx95, 2025.
[9] QGIS. https://www.qgis.org/en/site/, 2025.
[10] 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.
[11] T. N. Chan, P. L. Ip, B. Zhu, L. H. U, D. Wu, J. Xu, and C. S. Jensen. Large-scale spatiotemporal kernel density visualization. In *ICDE*, pages 99–113, 2025.
[12] 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.
[13] T. N. Chan, L. H. U, B. Choi, and J. Xu. SLAM: efficient sweep line algorithms for kernel density visualization. In *SIGMOD*, page 2120–2134, 2022.
[14] 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-Companion*, pages 99–102. ACM, 2023.
[15] E. Gan and P. Bailis. Scalable kernel density classification via threshold-based pruning. In *SIGMOD*, pages 945–959, 2017.
[16] A. G. Gray and A. W. Moore. Nonparametric density estimation: Toward computational tractability. In *SDM*, pages 203–211, 2003.
[17] J. Huang and M.-P. Kwan. Uncertainties in the assessment of covid-19 risk: A study of people's exposure to high-risk environments using individual-level activity data. *Annals of the American Association of Geographers*, 112(4):968–987, 2022.
[18] I. G. B. Putra, P.-F. Kuo, and D. Lord. Estimating the effectiveness of marked sidewalks: An application of the spatial causality approach. *Accident Analysis & Prevention*, 206:107699, 2024.