LION: Fast and High-Resolution Network Kernel Density Visualization

Tsz Nam Chan
Shenzhen University
edisonchan@szu.edu.cn

Rui Zang
Hong Kong Baptist University
19251017@life.hkbu.edu.hk

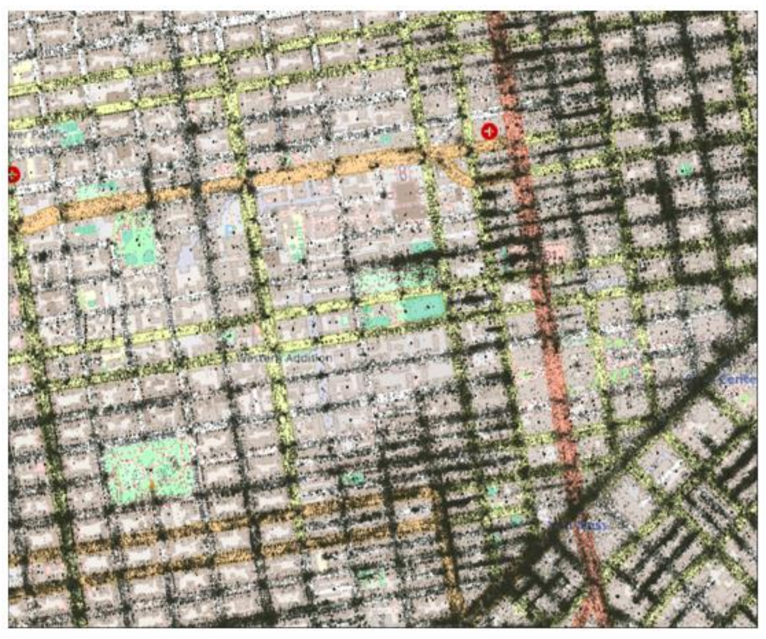
Bojian Zhu
Hong Kong Baptist University
csbjzhu@comp.hkbu.edu.hk

Leong Hou U
University of Macau
ryanlhu@um.edu.mo

Dingming Wu
Shenzhen University
dingming@szu.edu.cn

Jianliang Xu
Hong Kong Baptist University
xujl@comp.hkbu.edu.hk

Overview of Network Kernel Density Visualization (NKDV)



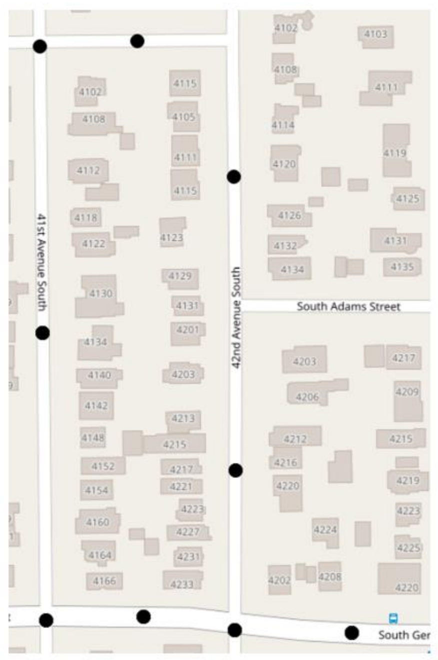
311-call location data points in San Francisco

NKDV

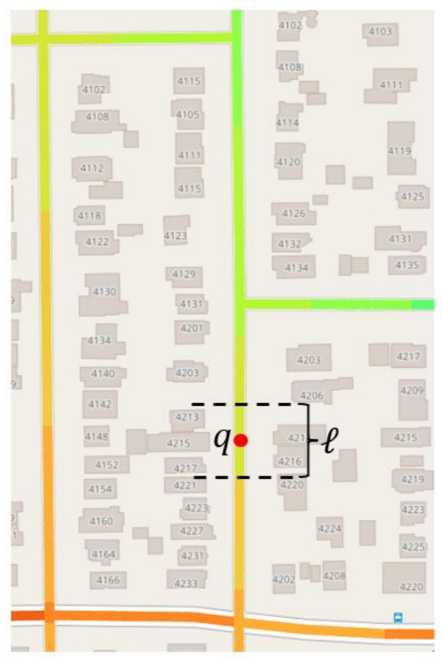
Consider a set of data points in a road network $G = (V, E)$.

Color each lixel q based on the network kernel density function $\mathcal{F}_P(q)$.

$$\mathcal{F}_{\underbrace{P}_{\text{dataset}}}(\overbrace{q}^{\text{lixel}}) = \sum_{p_i \in P} \overbrace{w}^{\text{weighting}} \cdot \begin{cases} 1 - \frac{1}{\underbrace{b^2}_{\text{bandwidth}}} \overbrace{d_G(q, p_i)^2}^{\text{shortest path distance}} & \text{if } d_G(q, p_i) \le b \\ 0 & \text{otherwise} \end{cases}$$



Location data points



NKDV

NKDV is Slow!

Suffer from high time complexity: $O(|E|T_{SP} + nL)$. ☹

Real example (London traffic accident dataset [a]):

- Number of lixels $L = 2.95$ million (with $\ell = 5m$)
- Number of data points $n = 0.838$ million
- NKDV takes at least 2.4721 trillion operations. ☹

Many complaints from domain experts. For example:

Rakshit et al. [b] "Kernel smoothing of point events, which is simple to define and very fast to compute in two dimensions (Diggle 1985), is mathematically complicated and can be extremely time-consuming to perform on a network..."

[a] Road Safety Data. https://data.gov.uk/dataset/cb7ae6f0-4be6-4935-9277-47e5ce24a11f/road-safety-data.
[b] 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.

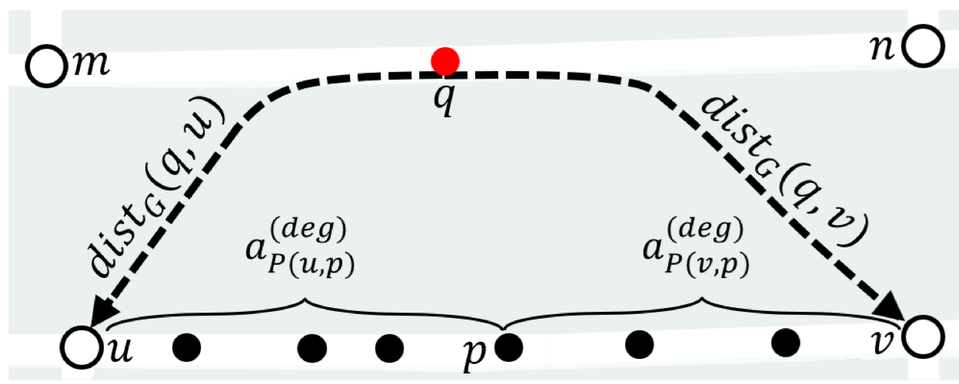
State-of-the-art Solution (ADA)

$$\mathcal{F}_P(q) = \sum_{p_i \in P} \mathcal{F}_{P(e)}(q) \quad \text{where} \quad \mathcal{F}_{P(e)}(q) = \sum_{p_i \in P(e)} w \cdot \begin{cases} 1 - \frac{1}{b^2} d_G(q, p_i)^2 & \text{if } d_G(q, p_i) \le b \\ 0 & \text{otherwise} \end{cases}$$

Augment $a^{(deg)}_{P(u,p)}$ and $a^{(deg)}_{P(v,p)}$ in each data point p of the edge $e = (u, v)$.

$$a^{(deg)}_{P(u,p)} = \sum_{p_i \in P(u,p)} d_G(u, p_i)^{deg}$$

$$a^{(deg)}_{P(v,p)} = \sum_{p_i \in P(v,p)} d_G(v, p_i)^{deg}$$



Use the binary search method to compute $\mathcal{F}_{P(e)}(q)$.

The time complexity of ADA is $O\left(|E|T_{SP} + L|E| \log\left(\frac{n}{|E|}\right)\right)$.

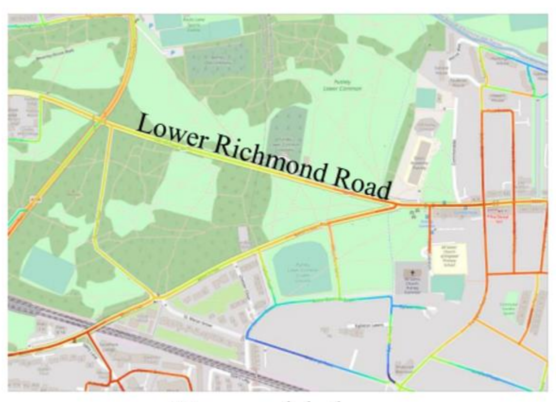
ADA is theoretically faster than the previous solution since

$$O\left(\log\left(\frac{n}{|E|}\right)\right) < O\left(\frac{n}{|E|}\right) \Rightarrow O\left(L|E| \log\left(\frac{n}{|E|}\right)\right) < O(nL)$$

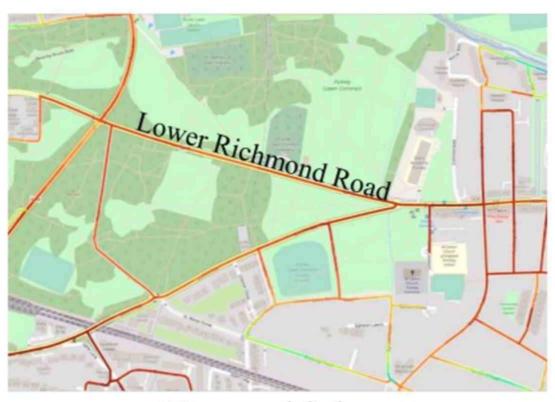
Weakness of ADA

In practice, we have $L > n$. As an example, generating NKDV for the London traffic accident dataset with $\ell = 5m$.

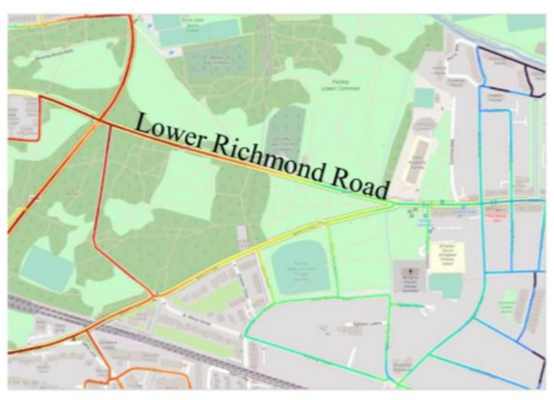
Domain experts can perform exploratory operations (e.g., filtering), making $L > n$.



One vehicle



Two vehicles

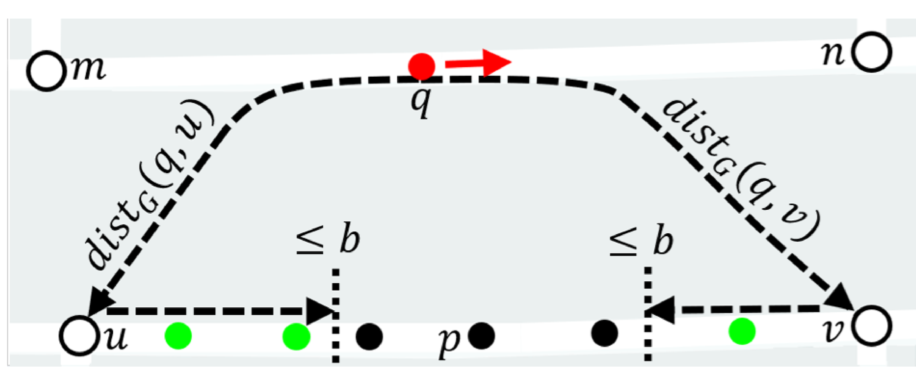


Three vehicles

$O\left(|E|T_{SP} + L|E| \log\left(\frac{n}{|E|}\right)\right)$ can still be slow! ☹

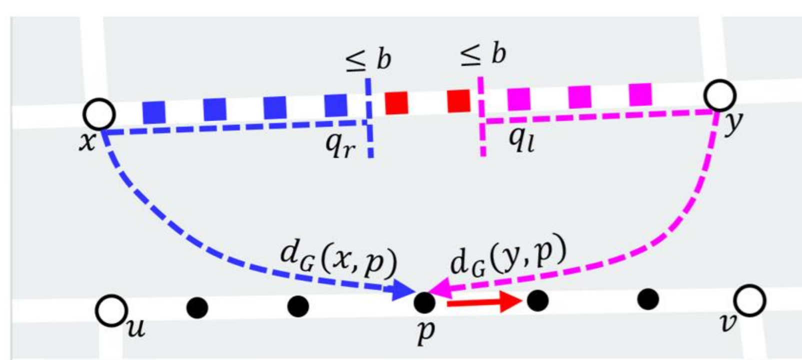
Our Solution: LION

Core idea: Change from pixel-scanning to point-scanning.



Pixel-scanning

$O\left(|E|T_{SP} + L|E| \log\left(\frac{n}{|E|}\right)\right)$



Point-scanning

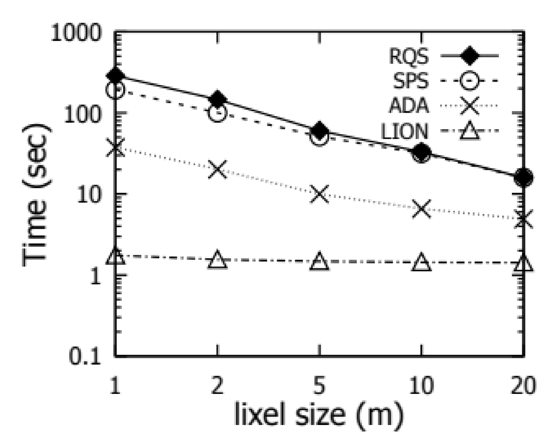
$O(|E|T_{SP} + n|E| + |E|^2 + L)$

The details of lixel augmentation and lixel aggregation can be found in our paper.

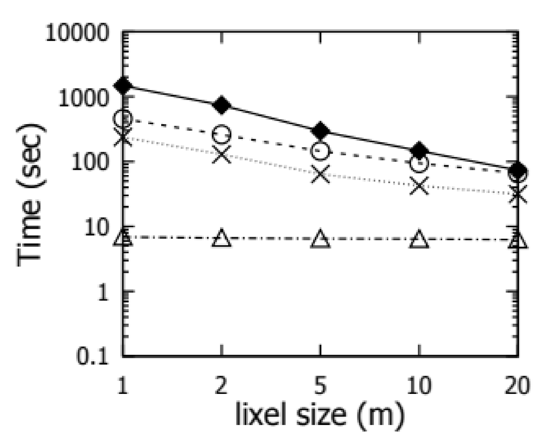
If $L > n$, LION achieves smaller time complexity compared with ADA.

Experimental Evaluation

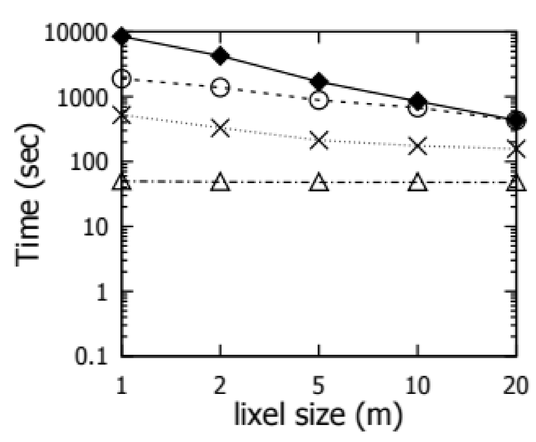
Dataset	$\lvert V\rvert$	$\lvert E\rvert$	n	Category
Gainesville	5,352	7,522	193,795	Crime events
Seattle	12,030	20,369	241,599	Traffic accidents
Chicago	40,428	69,219	719,372	Traffic accidents
Detroit	57,029	92,646	1,931,000	911 calls



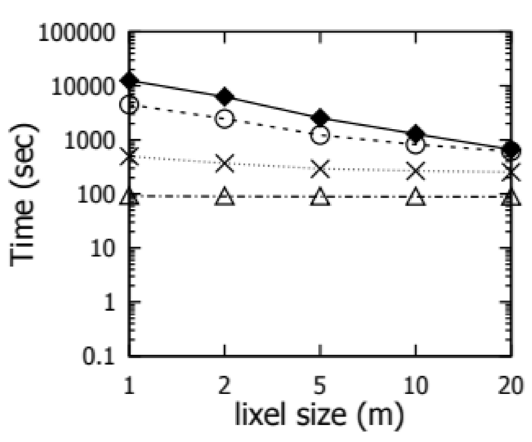
(a) Gainesville



(b) Seattle



(c) Chicago



(d) Detroit

Response time for generating NKDV, varying the lixel size.