Kernel Density Visualization for Big Geospatial Data: Algorithms and Applications

(Edison) Tsz Nam Chan[1] Leong Hou U[2] Byron Choi[1]

Jianliang Xu[1] Reynold Cheng[3]

[1]Hong Kong Baptist University [2]University of Macau [3]The University of Hong Kong

Tutorial Outline

1. Background of hotspot visualization

2. Background of kernel density visualization (KDV)

3. State-of-the-art methods of generating KDV

4. Other variants of KDV

5. Software development of KDV and its variants

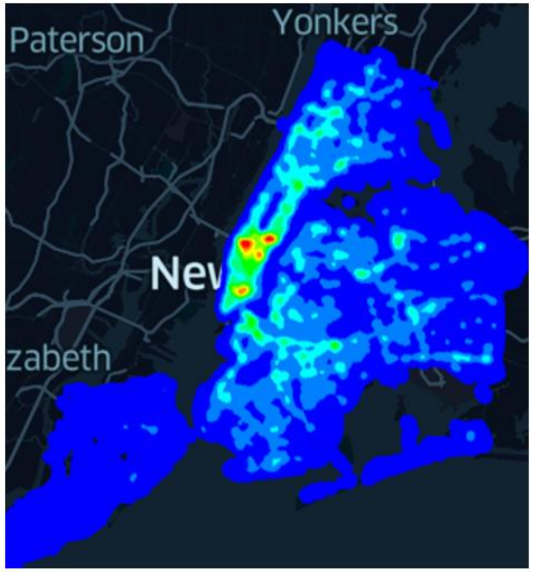
6. Future opportunities

Background of Hotspot Visualization

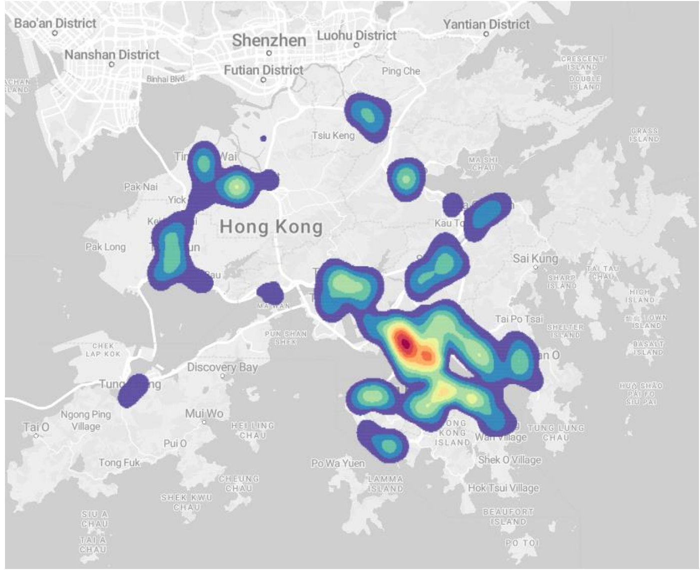
Why Hotspot Visualization?

- Finding the hidden patterns of location data points in different regions.
  - COVID-19 cases
  - Traffic accidents
  - Traffic flows
  - Crime events

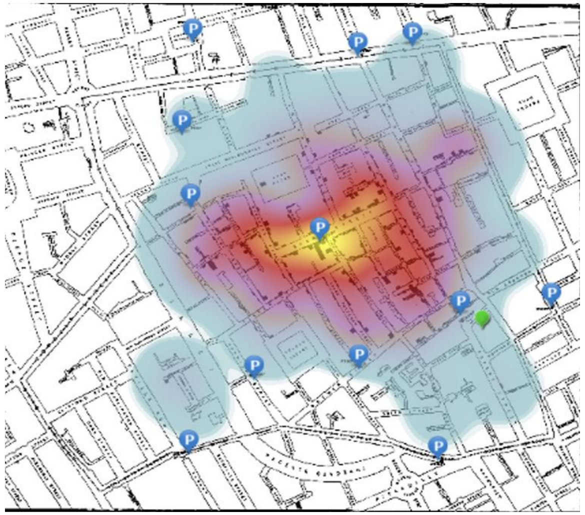
- Providing an intuitive analysis of different location datasets.



New York traffic accident heatmap



Hong Kong COVID-19 heatmap



1854 London cholera epidemic

Scatter Plot

- Directly plots data points in the map.

Scatter plot of the data points of
1854 London cholera epidemic

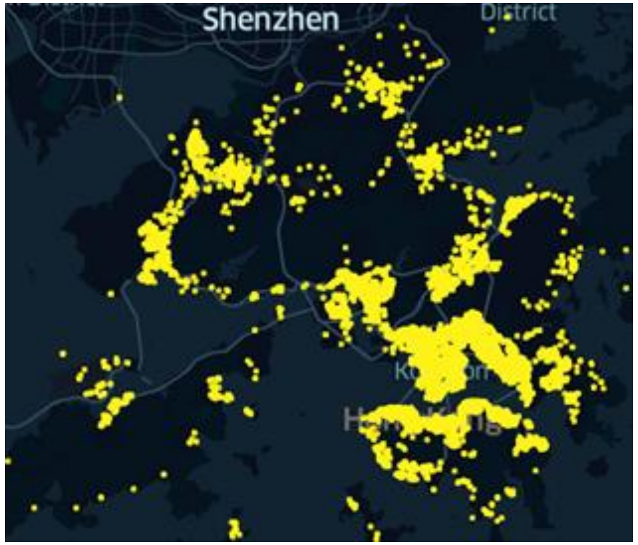
[LDAV15] A. Perrot, R. Bourqui, N. Hanusse F. Lalanne D. Auber, "Large Interactive Visualization of Density Functions on Big Data Infrastructure" LDAV2015

Advantages of Scatter Plot

- Simple ☺
- Show the patterns clearly for small data ☺
- Efficient ☺

Overplotting Issues of Scatter Plot

- Difficult to find which parts contain more data points (Overplotting) ☹
  - This issue is more serious if the number of data points is much larger than the resolution size.

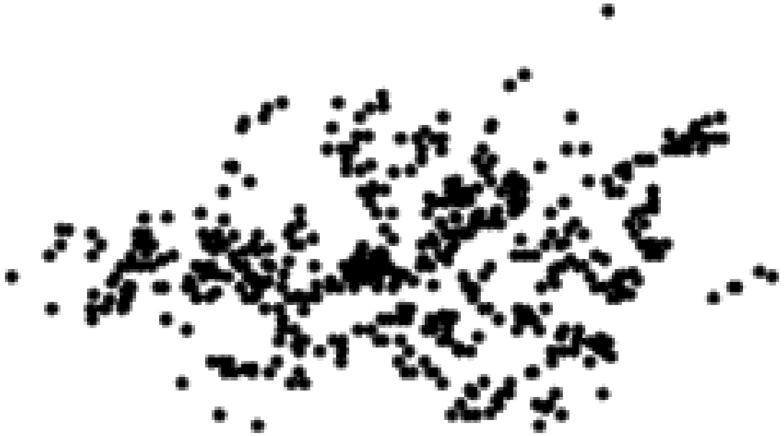


Hong Kong COVID-19 cases

Overplotting Issues of Scatter Plot

- Seriously suffers from the resolution changes ☹

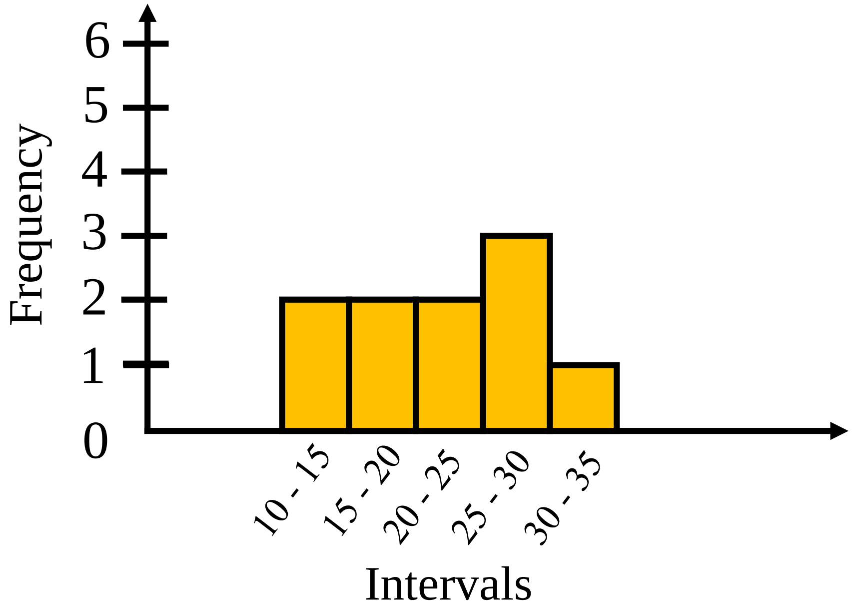
Resolution: 512 x 512

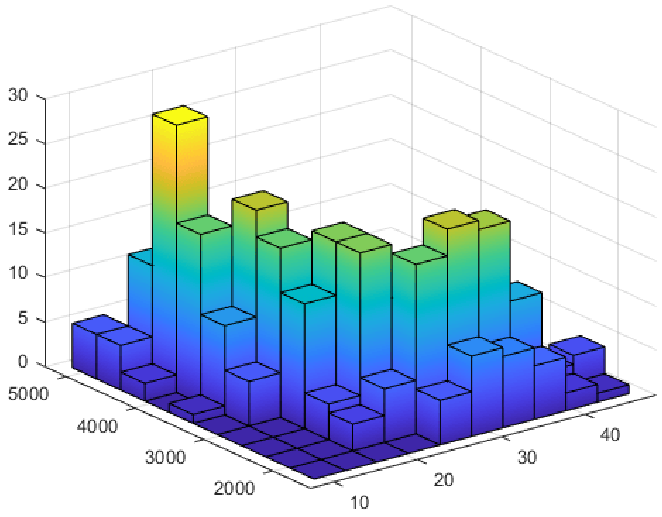


Resolution: 256 x 256

Histogram

- Divides the space into different intervals/ sub-regions with the same size.
- Counts the frequency in each interval/sub-region.



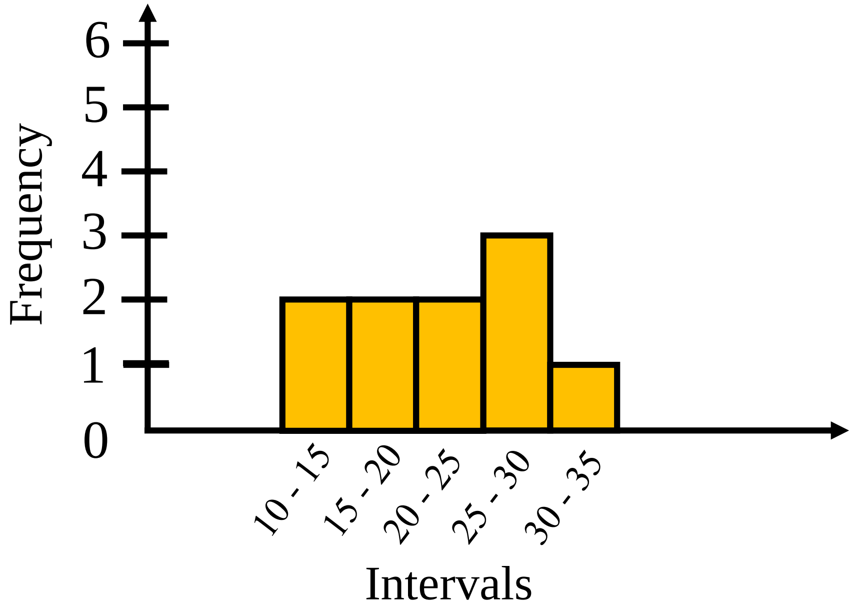


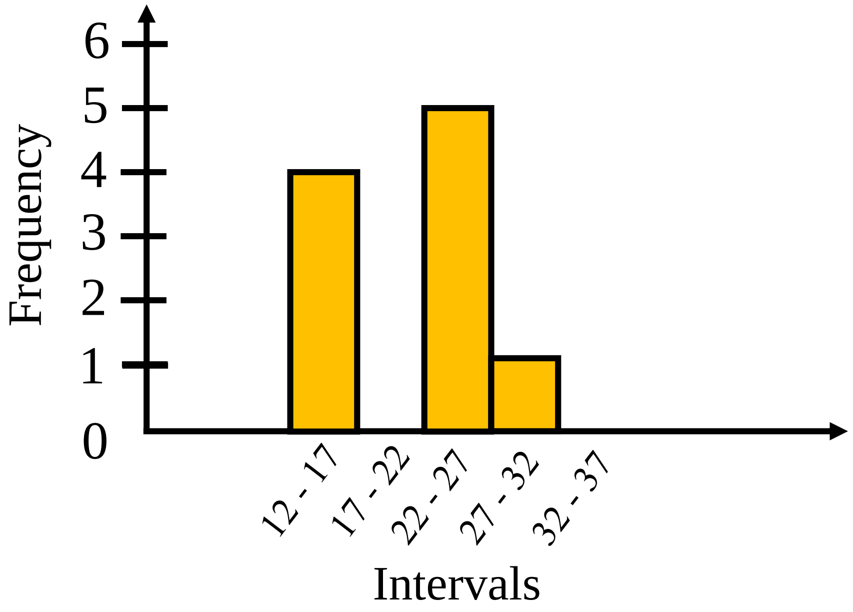
Advantages of Histogram

- Simple ☺
- Efficient ☺
- Solve the overplotting issues ☺

Histogram is Sensitive to the Pixel Positions

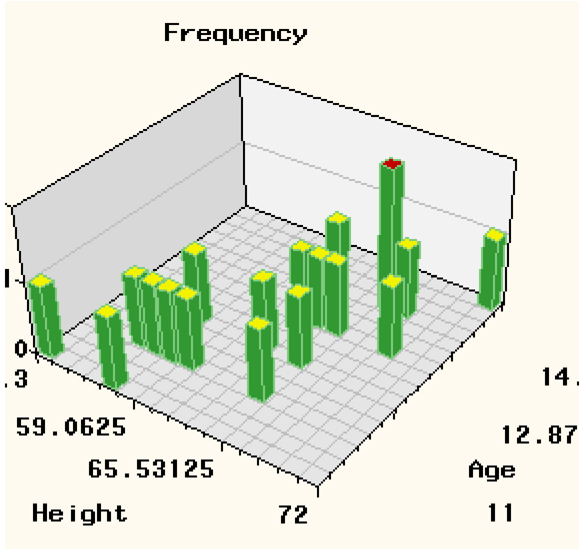
- Different starting points in the x-axis can significantly affect the visualization (link) ☹





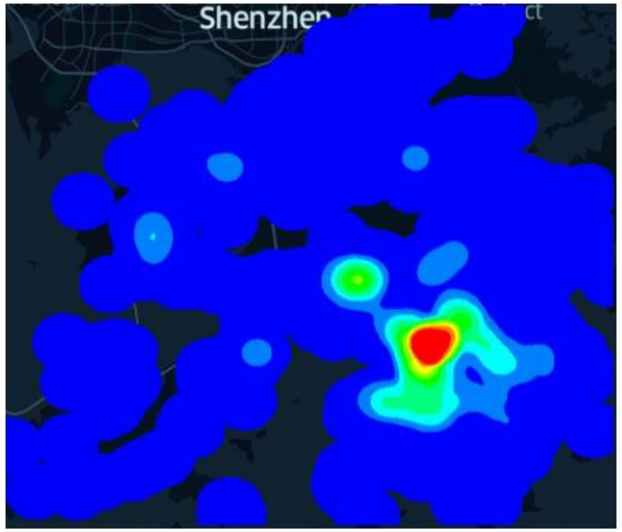
Histogram is Not Smooth

- The visualization is not smooth (There can be a huge change between two consecutive bins) ☹



Background of Kernel Density Visualization (KDV)

Kernel Density Visualization (KDV)



- Each **p** (yellow dot) represents the location of a COVID-19 case.
- Predict the risk of a given location **q** by computing the ***kernel density function*** $\mathcal{F}_P(\mathbf{q})$.

$$\underbrace{\mathcal{F}_P}_{\text{dataset}}(\overbrace{\mathbf{q}}^{\text{2D pixel}}) = \sum_{\mathbf{p}\in P} \overbrace{w}^{\text{weighting}} \cdot \begin{cases} 1 - \dfrac{1}{\underbrace{b^2}_{\text{bandwidth}}} \overbrace{dist(\mathbf{q},\mathbf{p})}^{\text{Euclidean distance}}{}^2 & \text{If } dist(\mathbf{q},\mathbf{p}) \le b \\ 0 & \text{Otherwise} \end{cases}$$

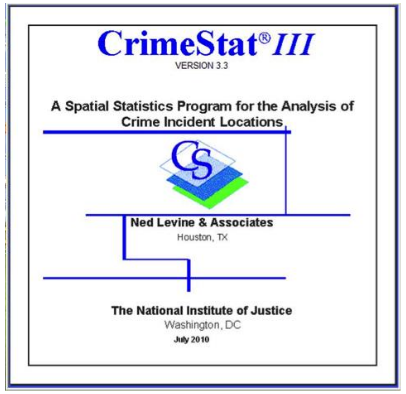
Software Packages for Supporting KDV

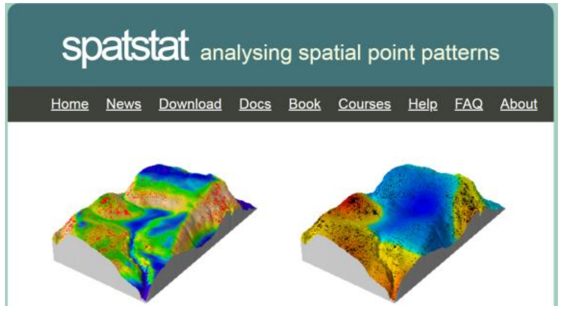
ArcGIS

PySAL

SciPy

scikit learn


CrimeStat® III
VERSION 3.3
A Spatial Statistics Program for the Analysis of Crime Incident Locations
Ned Levine & Associates
Houston, TX
The National Institute of Justice
Washington, DC
July 2010



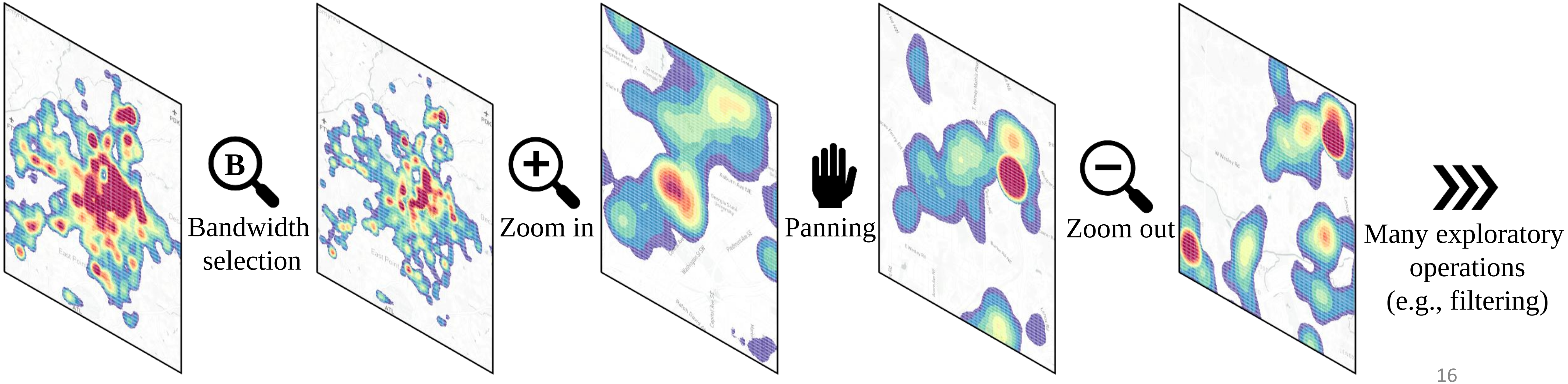
DECK.GL

seaborn

KDV is Slow!

- The time complexity is $O(XYn)$ ☹
  - $X \times Y$ denotes the number of pixels.
  - n denotes the number of location data points.

- Domain experts need to generate multiple KDVs ☹



KDV is Slow!

- Many complaints from domain experts:
  - Zhang [TGIS22] “Taking computation processes underlying the KDE-based method (modeling, prediction, and integration) literally, one would apply the sequence of computation steps cell-by-cell to generate a final suitability map. That is, using the covariate value at one cell as an input to compute a final suitability value before repeating the same computations at the next cell. However, this is computationally inefficient as computations may be unnecessarily repeated.”
  - Gramacki et al. [SIP17] “However, many (or even most) of the practical algorithms and solutions designed in the context of KDE are very time-consuming with quadratic computational complexity being a commonplace.”
  - Gan et al. [SIGMOD17] “Kernel Density Estimation (KDE) is a powerful technique for computing these densities, offering excellent statistical accuracy but quadratic total runtime.”

[TGIS22] G. Zhang. PyCLKDE: A big data-enabled high-performance computational framework for species habitat suitability modeling and mapping. Transactions in GIS, 2022.
[SIP17] A. Gramacki. Nonparametric Kernel Density Estimation and Its Computational Aspects. Springer International Publishing, 2017.

[SIGMOD17] E. Gan and P. Bailis. Scalable Kernel Density Classification via Threshold-Based Pruning. SIGMOD 2017.

State-of-the-art Methods of Generating KDV

State-of-the-art Methods for Generating KDV

- Function approximation [**TKDE22**, **SIGMOD20**, **ICDE19**, SIGMOD17, SDM03]
- Data sampling [SOCG18, SODA18, SODA13, SIGMOD13]
- Computational sharing [**VLDB22a**, AISTATS03]
- Computational geometry [**SIGMOD22**]

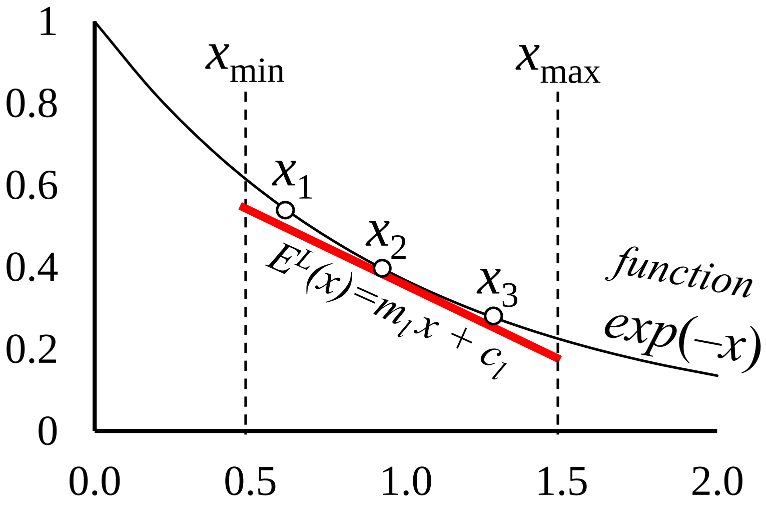
[TKDE22] T. N. Chan, L. H. U, R. Cheng, M. L. Yiu, Shivansh Mittal. Efficient Algorithms for Kernel Aggregation Queries. TKDE 2022.
[SIGMOD22] T. N. Chan, L. H. U, B. Choi, J. Xu. SLAM: Efficient Sweep Line Algorithms for Kernel Density Visualization. SIGMOD 2022.
[VLDB22a] T. N. Chan, P. L. Ip, L. H. U, B. Choi, J. Xu. SAFE: A Share-and-Aggregate Bandwidth Exploration Framework for Kernel Density Visualization. VLDB 2022.
[SIGMOD20] T. N. Chan, R. Cheng, M. L. Yiu. QUAD: Quadratic-Bound-based Kernel Density Visualization. SIGMOD 2020.
[ICDE19] T. N. Chan, M. L. Yiu, L. H. U. KARL: Fast Kernel Aggregation Queries. ICDE 2019.
[SOCG18] J. M. Phillips and W. M. Tai. Near-Optimal Coresets of Kernel Density Estimates. SOCG 2018.
[SODA18] J. M. Phillips and W. M. Tai. Improved Coresets for Kernel Density Estimates. SODA 2018.
[SIGMOD17] E. Gan and P. Bailis. Scalable Kernel Density Classification via Threshold-Based Pruning. SIGMOD 2017.
[SODA13] J. M. Phillips. ϵ-Samples for Kernels. In SODA 2013.
[SIGMOD13] Y. Zheng, J. Jestes, J. M. Phillips, F. Li. Quality and Efficiency for Kernel Density Estimates in Large Data. SIGMOD 2013.
[SDM03] A. G. Gray and A. W. Moore. Nonparametric Density Estimation: Toward Computational Tractability. SDM 2003.
[AISTATS03] A. G. Gray and A. W. Moore. Rapid Evaluation of Multiple Density Models. AISTATS 2003.

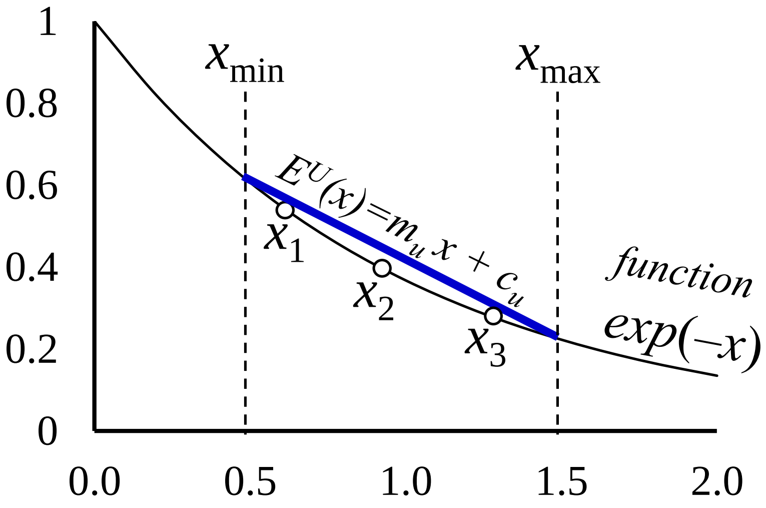
Function Approximation

- Consider the kernel density function (with the Gaussian kernel).

$$\mathcal{F}_P(\mathbf{q}) = \sum_{\mathbf{p}\in P} w \cdot \exp\left(\underbrace{-\frac{1}{b^2} dist(\mathbf{q},\mathbf{p})^2}_{x}\right)$$

- Use some simple functions (e.g., linear functions) to approximate the exponential function so that we can obtain the lower and upper bounds of $\mathcal{F}_P(\mathbf{q})$.





Function Approximation

$$\mathcal{F}_P(\mathbf{q}) = \sum_{\mathbf{p}\in P} w \cdot \exp\left(\underbrace{-\frac{1}{b^2}dist(\mathbf{q},\mathbf{p})^2}_{x}\right) \qquad O(n) \text{ time}$$

We have $LB_P(\mathbf{q}) \leq \mathcal{F}_P(\mathbf{q}) \leq UB_P(\mathbf{q})$.

Lower bound of $\mathcal{F}_P(\mathbf{q})$:

$$LB_P(\mathbf{q}) = \sum_{\mathbf{p_i}\in P} w\left(m_l\underbrace{\left(\frac{1}{b^2}dist(\mathbf{q},\mathbf{p})^2\right)}_{x} + c_l\right)$$

$$= wm\frac{1}{b^2}(\underbrace{|P|\|\mathbf{q}\|^2}_{\text{O(1)}} - \underbrace{2\mathbf{q}\cdot\mathbf{a}_P}_{\text{O(1)}} + b_P) + wc|P| \qquad O(1) \text{ time}$$

We can further tighten these bound values using some index structures (e.g., kd-tree) until they fulfill the relative error guarantees.

Advantages and Disadvantages of Function Approximation

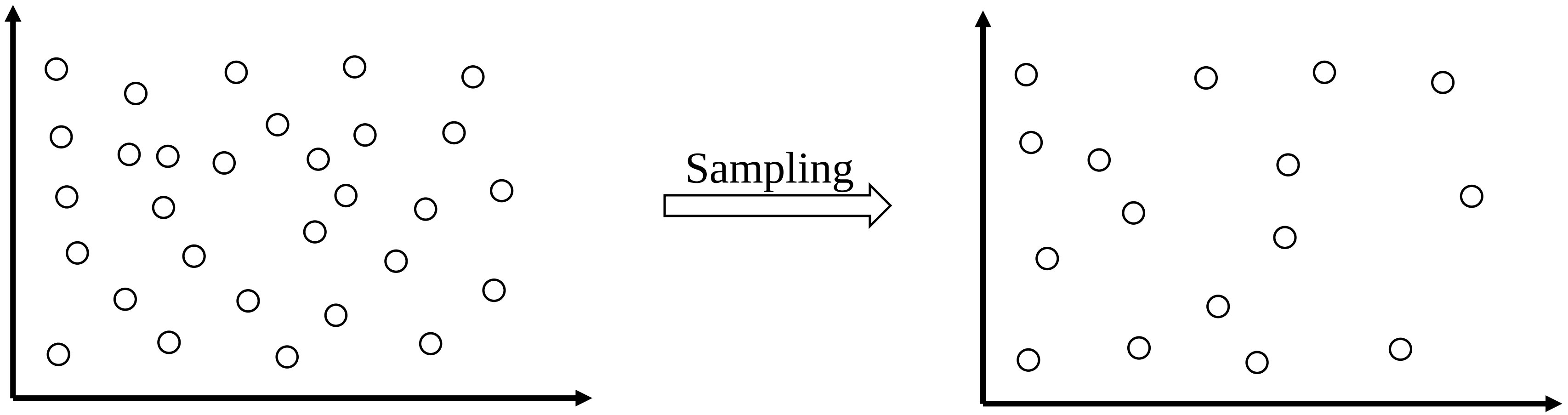
- Advantages ☺
  - Achieve better practical performance.
  - Can handle all kernel functions.
  - Can achieve approximation guarantees for generating KDV.

- Disadvantages ☹
  - Cannot reduce the worst-case time complexity for generating KDV.
  - Cannot achieve exact solution.
  - Can still be slow for generating KDV with some famous kernel functions (Epanechnikov and quartic kernels).

Data Sampling

- Consider the kernel density function (with the Gaussian kernel).

$$\mathcal{F}_P(\mathbf{q}) = \sum_{\mathbf{p}\in P} w \cdot \exp\left(-\frac{1}{b^2} dist(\mathbf{q}, \mathbf{p})^2\right)$$



- Compute the modified kernel density function based on the sampled dataset S.

$$\mathcal{F}_S^{(M)}(\mathbf{q}) = \sum_{\mathbf{p}_i\in S} w_i \cdot \exp\left(-\frac{1}{b^2} dist(\mathbf{q}, \mathbf{p}_i)^2\right)$$

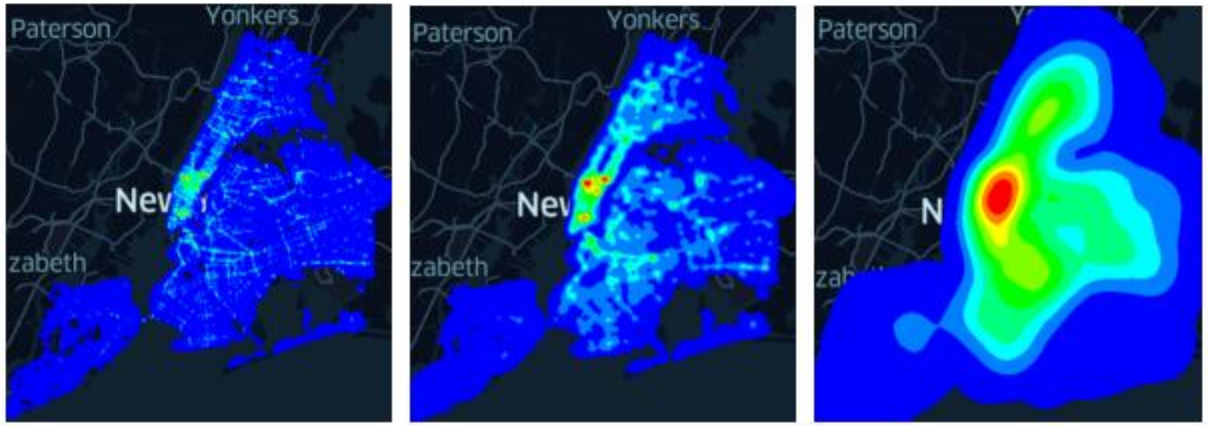
Advantages and Disadvantages of Data Sampling

- Advantages ☺
  - Can achieve probabilistic approximation guarantees for generating KDV.
  - Can reduce the worst-case time complexity for generating KDV.
  - Can handle all kernel functions.

- Disadvantages ☹
  - Cannot achieve exact solution.
  - Can still be slow for generating KDV.
  - Can degrade the practical visualization quality.

Computational Sharing

- Different bandwidth parameters b can significantly affect the visualization.



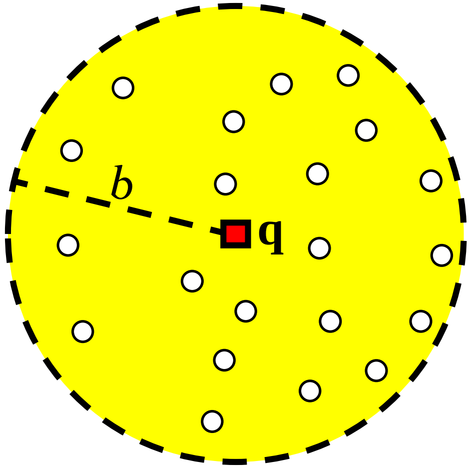
(a) Small b (b) Moderate b (c) Large b

Computational Sharing

- Consider the kernel density function (with Epanechnikov kernel).

$$\mathcal{F}_P(\mathbf{q}) = \sum_{\mathbf{p}\in P} w \cdot \begin{cases} 1 - \frac{1}{b^2} dist(\mathbf{q},\mathbf{p})^2 & \text{If } dist(\mathbf{q},\mathbf{p}) \leq b \\ 0 & \text{Otherwise} \end{cases}$$

- Kernel density function $\mathcal{F}_P(\mathbf{q})$ can be decomposed as follows.



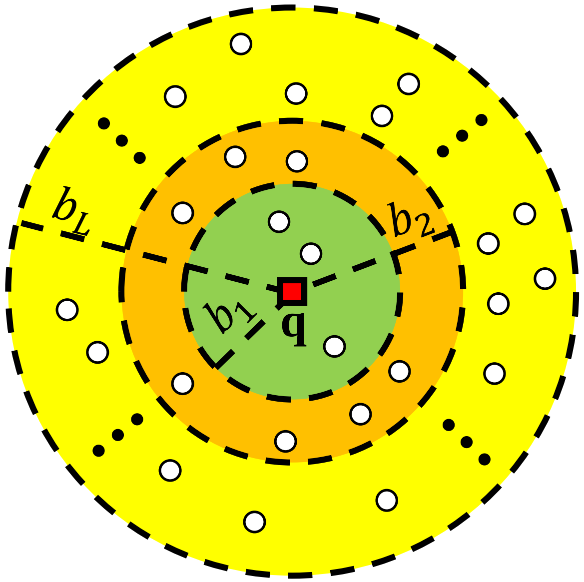
Range query set $R_{\mathbf{q}}^{(b)}$

$$\mathcal{F}_P(\mathbf{q}) = \sum_{\mathbf{p}\in R_{\mathbf{q}}^{(b)}} w \cdot \left(1 - \frac{1}{b^2} dist(\mathbf{q},\mathbf{p})^2\right)$$

$$= w\left|R_{\mathbf{q}}^{(b)}\right| - \frac{w}{b^2} S_{R_{\mathbf{q}}^{(b)}}$$

where $S_{R_{\mathbf{q}}^{(b)}} = \sum_{\mathbf{p}\in R_{\mathbf{q}}^{(b)}} dist(\mathbf{q},\mathbf{p})^2$

Computational Sharing



- Suppose that $b_1 \leq b_2 \leq \cdots \leq b_L$, we can conclude that

$$R_{\mathbf{q}}^{(b_1)} \subseteq R_{\mathbf{q}}^{(b_2)} \subseteq \cdots \subseteq R_{\mathbf{q}}^{(b_L)}$$

- $R_{\mathbf{q}}^{(b_L)}$ can be shared for computing $\mathcal{F}_P(\mathbf{q})$ with other bandwidths.

Advantages and Disadvantages of Computational Sharing

- Advantages ☺
  - Can achieve the exact solution for generating multiple KDVs.
  - Can reduce the worst-case time complexity for generating multiple KDVs.
  - Can achieve better practical efficiency for generating multiple KDVs.
  - Can combine with data sampling methods for generating multiple KDVs.

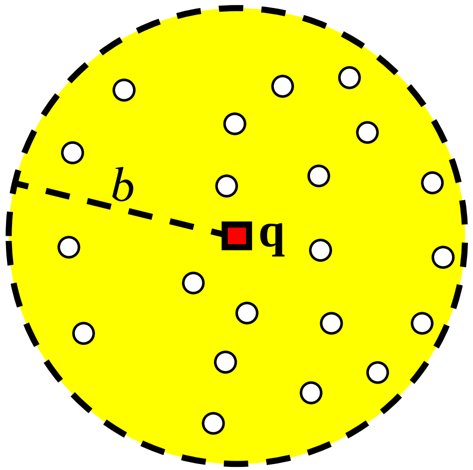
- Disadvantages ☹
  - Cannot support all kernel functions (e.g., cannot support Gaussian kernel).
  - Cannot achieve optimal worst-case time complexity.
  - Cannot reduce the worst-case time complexity for generating a single KDV.

Computational Geometry

- Consider the kernel density function (with Epanechnikov kernel).

$$\mathcal{F}_P(\mathbf{q}) = \sum_{\mathbf{p}\in P} w \cdot \begin{cases} 1 - \frac{1}{b^2} dist(\mathbf{q},\mathbf{p})^2 & \text{If } dist(\mathbf{q},\mathbf{p}) \leq b \\ 0 & \text{Otherwise} \end{cases}$$

- Kernel density function $\mathcal{F}_P(\mathbf{q})$ can be decomposed as follows.



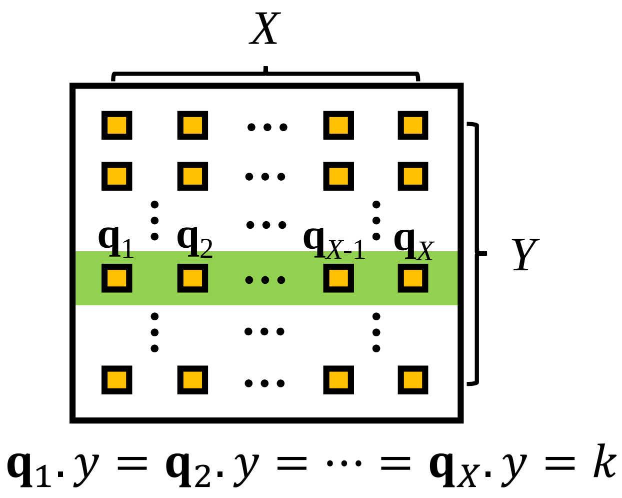
Range query set $R(\mathbf{q})$

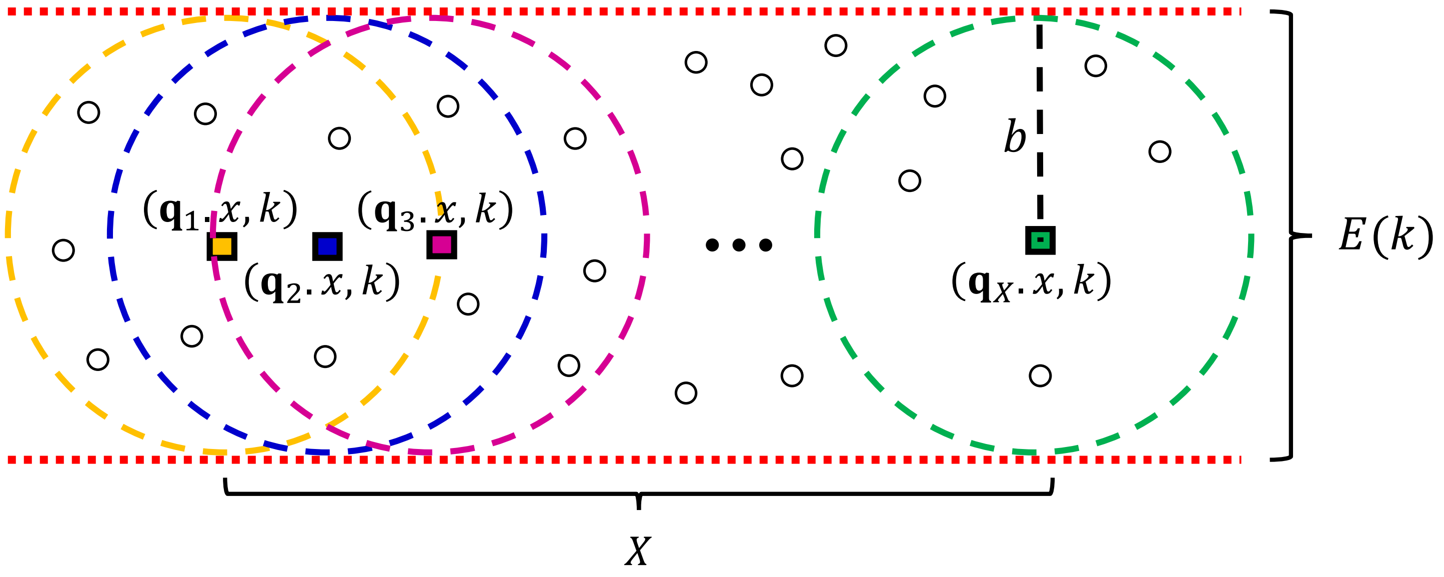
$$\mathcal{F}_P(\mathbf{q}) = \sum_{\mathbf{p}\in R(\mathbf{q})} w \cdot \left(1 - \frac{1}{b^2} dist(\mathbf{q},\mathbf{p})^2\right)$$

$$= w|R(\mathbf{q})| - \frac{w}{b^2}\left(|R(\mathbf{q})| \times \|\mathbf{q}\|_2^2 - 2\mathbf{q}^T \underbrace{\mathbf{A}_{R_\mathbf{q}}}_{\sum_{\mathbf{p}\in R(\mathbf{q})} \mathbf{p}} + \underbrace{S_{R_\mathbf{q}}}_{\sum_{\mathbf{p}\in R(\mathbf{q})} \|\mathbf{p}\|_2^2}\right)$$

How to efficiently maintain $R(\mathbf{q})$?

Computational Geometry

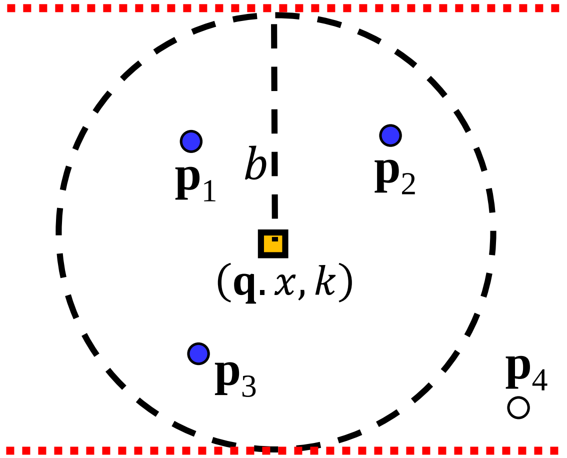




Use $O(n)$ time to find the envelope $E(k)$.

$$E(k) = \{\mathbf{p} \in P : |k - \mathbf{p}.y| \leq b\}$$

Computational Geometry



- Consider the blue data points **p** that are within the range b of the pixel **q**.

$$
\begin{aligned}
dist(\mathbf{q}, \mathbf{p}) &\leq b \\
(\mathbf{q}.x - \mathbf{p}.x)^2 &\leq b^2 - (k - \mathbf{p}.y)^2
\end{aligned}
$$

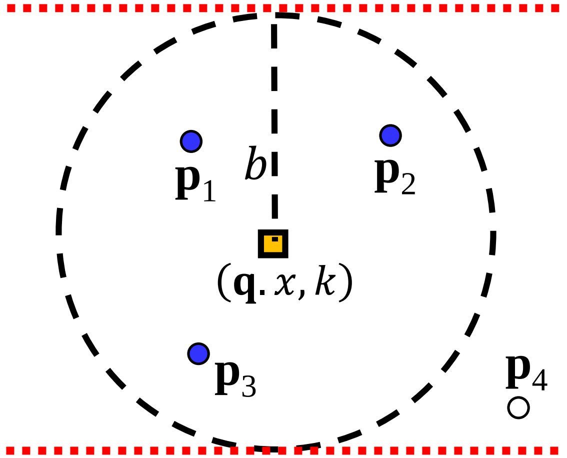
$$\mathbf{p}.x - \sqrt{b^2 - (k - \mathbf{p}.y)^2} \leq \mathbf{q}.x \leq \mathbf{p}.x + \sqrt{b^2 - (k - \mathbf{p}.y)^2}$$

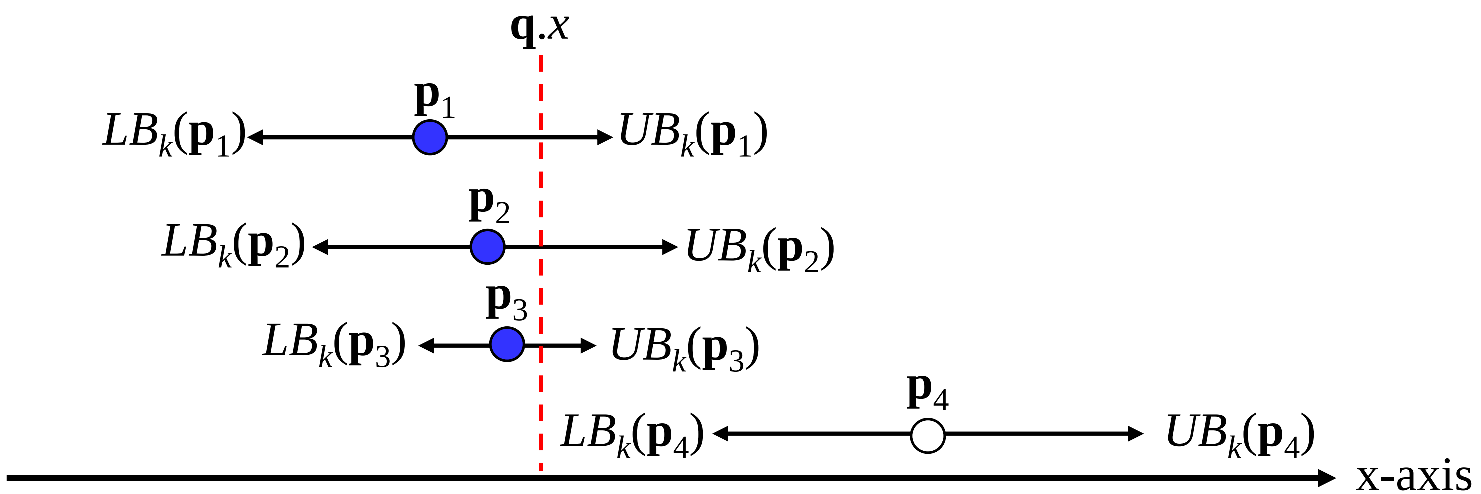
- We can let:

$$
\begin{aligned}
LB_k(\mathbf{p}) &= \mathbf{p}.x - \sqrt{b^2 - (k - \mathbf{p}.y)^2} \\
UB_k(\mathbf{p}) &= \mathbf{p}.x + \sqrt{b^2 - (k - \mathbf{p}.y)^2}
\end{aligned}
$$

- $O(n)$ time to find the bound functions for all data points in $E(k)$.

Computational Geometry





- Range search problem = Interval stabbing problem.
- Our sweep line algorithm (SLAM) can generate KDV in $O(Y(X+n))$ time ☺

Advantages and Disadvantages of Computational Geometry

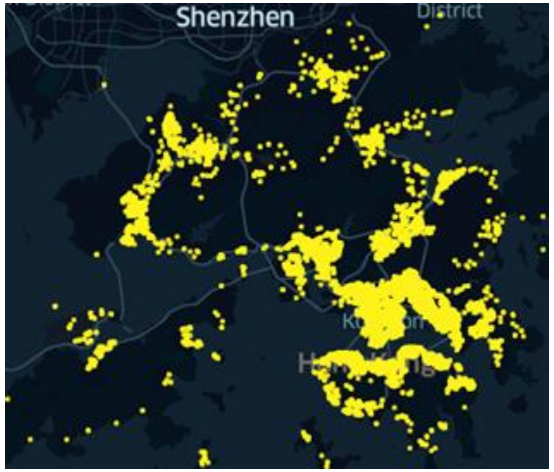
- Advantages ☺
  - Can achieve the exact solution
  - Can reduce the worst-case time complexity
  - Can achieve the best practical efficiency
  - Can combine with data sampling methods

- Disadvantages ☹
  - Cannot support all kernel functions (e.g., cannot support Gaussian kernel).
  - Cannot achieve optimal worst-case time complexity.

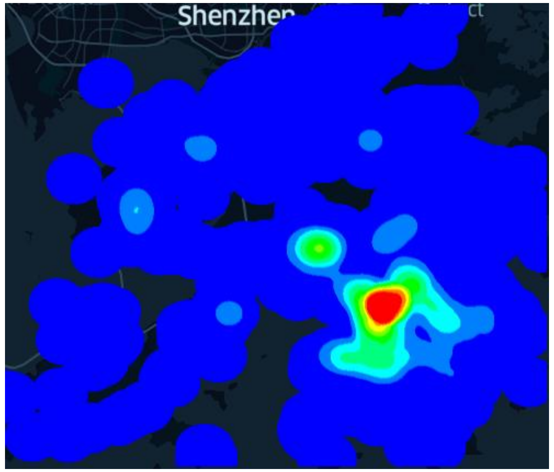
Other Variants of KDV

Variant 1: Spatiotemporal Kernel Density Visualization (STKDV)

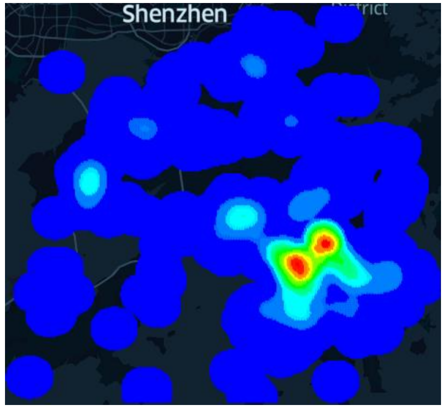
- KDV does not consider the occurrence time of each geographical event, which may provide misleading visualization results.



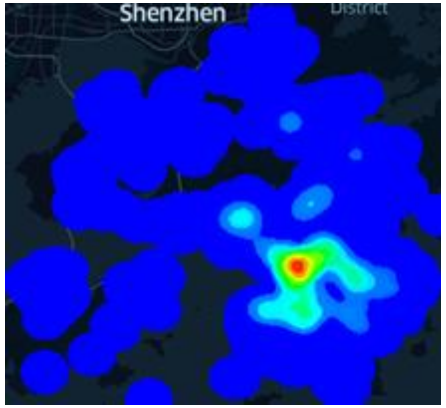
Hong Kong COVID-19 cases



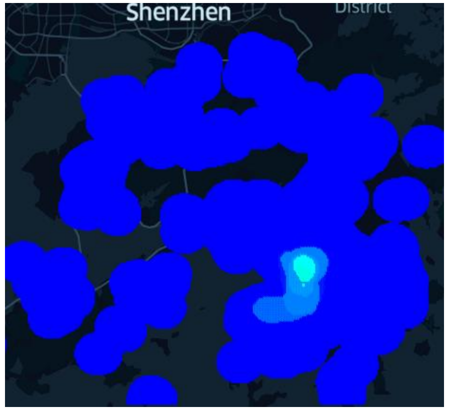
Hotspot map (based on KDV)



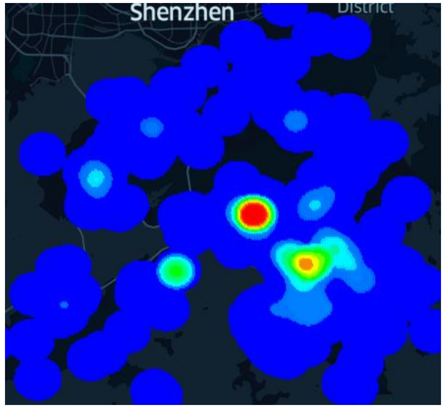
2nd August 2020



6th December 2020

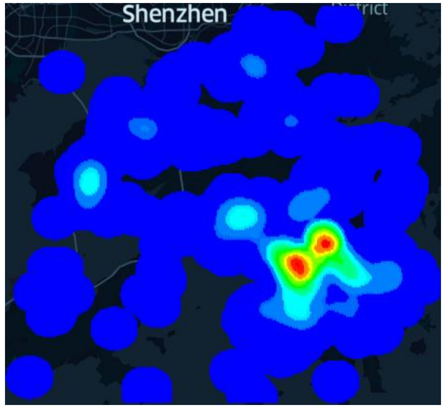


28th February 2021

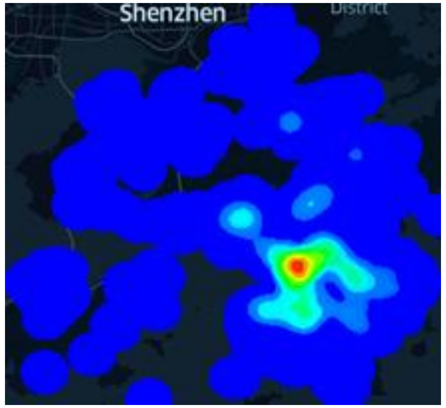


28th January 2022

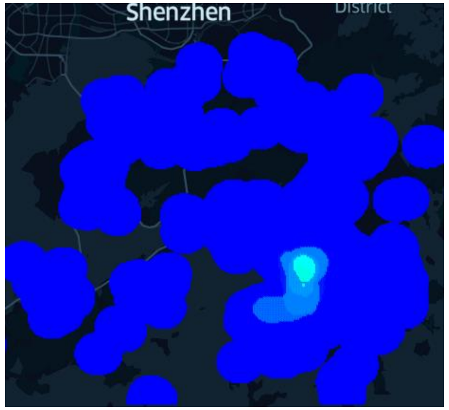
Variant 1: Spatiotemporal Kernel Density Visualization (STKDV)



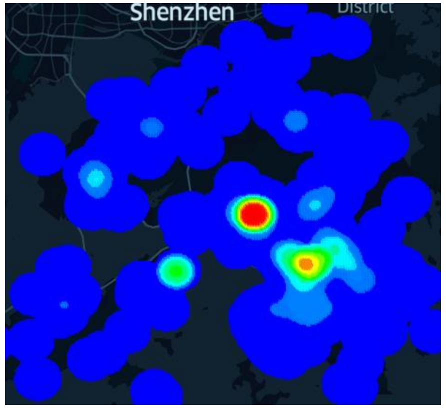
2nd August 2020



6th December 2020



28th February 2021



28th January 2022

- Consider a location dataset $\hat{P} = \{(\mathbf{p}_1, t_{\mathbf{p}_1}), (\mathbf{p}_2, t_{\mathbf{p}_2}), \ldots, (\mathbf{p}_n, t_{\mathbf{p}_n})\}$ with size n.

- Color each pixel $\mathbf{q}$ with the timestamp $t_{\mathbf{q}}$ based on the spatial-temporal kernel density function $\mathcal{F}_{\hat{P}}(\mathbf{q}, t_{\mathbf{q}})$.

$$\mathcal{F}_{\hat{P}}(\mathbf{q}, t_{\mathbf{q}}) = \sum_{(\mathbf{p}, t_{\mathbf{p}}) \in \hat{P}} w \cdot K_{\text{space}}(\mathbf{q}, \mathbf{p}) \cdot K_{\text{time}}(t_{\mathbf{q}}, t_{\mathbf{p}})$$

Variant 1: Spatiotemporal Kernel Density Visualization (STKDV)

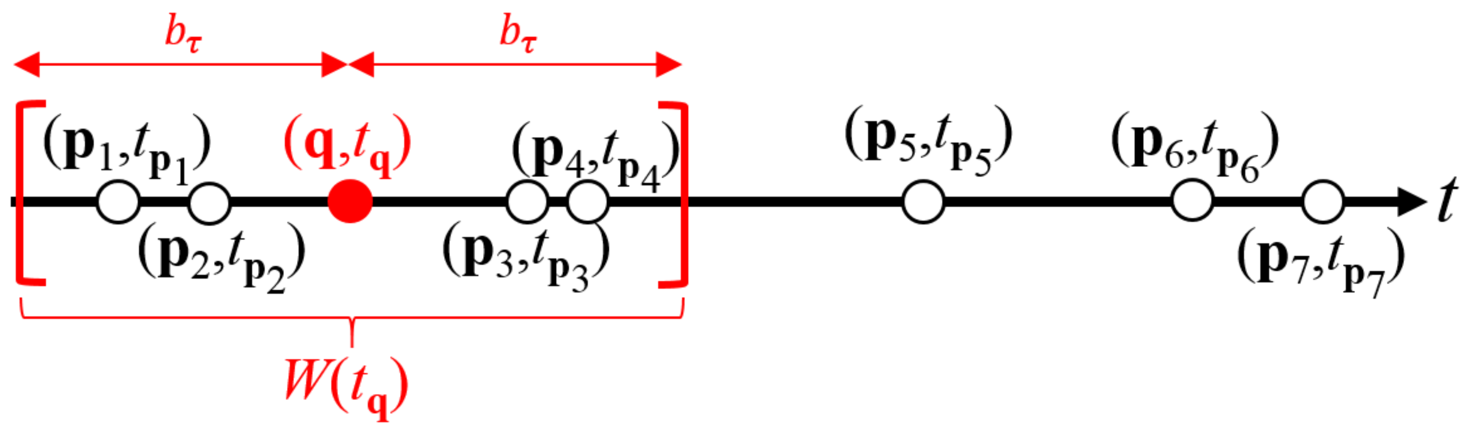
- Time complexity of a naïve solution is $O(XYTn)$ (Very slow!) ☹

- The time complexity of the best solution, called SWS [**VLDB22b**], is $O(XY(T+n))$ ☺

[VLDB22b] T. N. Chan, P. L. Ip, L. H. U, B. Choi, J. Xu. SWS: A Complexity-Optimized Solution for Spatial-Temporal Kernel Density Visualization. VLDB 2022.

Core Idea 1 of SWS: Sliding Window

- Establish the sliding window in the temporal dimension.



- Only $(\mathbf{p}_1, t_{\mathbf{p}_1})$, $(\mathbf{p}_2, t_{\mathbf{p}_2})$, $(\mathbf{p}_3, t_{\mathbf{p}_3})$, and $(\mathbf{p}_4, t_{\mathbf{p}_4})$ can contribute to $\mathcal{F}_{\hat{P}}(\mathbf{q}, t_{\mathbf{q}})$.

$$\mathcal{F}_{\hat{P}}(\mathbf{q}, t_{\mathbf{q}}) = \sum_{(\mathbf{p}, t_{\mathbf{p}}) \in \hat{P}} w \cdot K_{\text{space}}(\mathbf{q}, \mathbf{p}) \cdot \begin{cases} 1 - \frac{1}{b_\tau^2} dist(t_{\mathbf{q}}, t_{\mathbf{p}})^2 & \text{If } dist(t_{\mathbf{q}}, t_{\mathbf{p}}) \leq b_\tau \\ 0 & \text{Otherwise} \end{cases}$$

$$= \sum_{(\mathbf{p}, t_{\mathbf{p}}) \in W(t_{\mathbf{q}})} w \cdot K_{\text{space}}(\mathbf{q}, \mathbf{p}) \cdot \left(1 - \frac{1}{b_\tau^2} dist(t_{\mathbf{q}}, t_{\mathbf{p}})^2\right)$$

Core Idea 1 of SWS: Sliding Window

- Establish the sliding window in the temporal dimension.

$$\mathcal{F}_{\hat{P}}(\mathbf{q}, t_{\mathbf{q}}) = \sum_{(\mathbf{p}, t_{\mathbf{p}}) \in W(t_{\mathbf{q}})} w \cdot K_{\text{space}}(\mathbf{q}, \mathbf{p}) \cdot \left(1 - \frac{1}{b_\tau^2} dist(t_{\mathbf{q}}, t_{\mathbf{p}})^2\right)$$

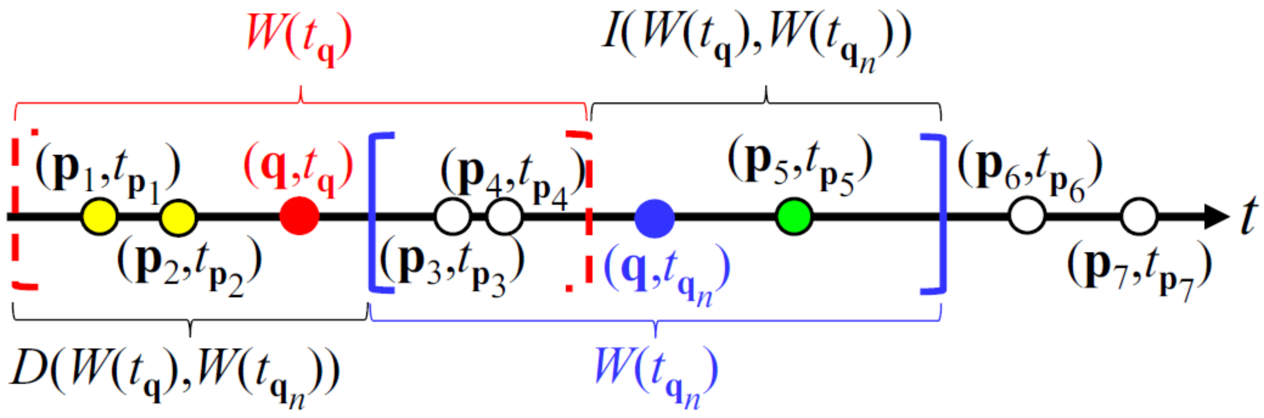
- Only $(\mathbf{p}_1, t_{\mathbf{p}_1})$, $(\mathbf{p}_2, t_{\mathbf{p}_2})$, $(\mathbf{p}_3, t_{\mathbf{p}_3})$, and $(\mathbf{p}_4, t_{\mathbf{p}_4})$ can contribute to $\mathcal{F}_{\hat{P}}(\mathbf{q}, t_{\mathbf{q}})$.

$$\mathcal{F}_{\hat{P}}(\mathbf{q}, t_{\mathbf{q}}) = \sum_{(\mathbf{p}, t_{\mathbf{p}}) \in W(t_{\mathbf{q}})} w \cdot K_{\text{space}}(\mathbf{q}, \mathbf{p}) \cdot \left(1 - \frac{1}{b_\tau^2} dist(t_{\mathbf{q}}, t_{\mathbf{p}})^2\right)$$

$$= w\left(1 - \frac{1}{b_\tau^2} t_{\mathbf{q}}^2\right) \cdot S_{W(t_{\mathbf{q}})}^{(0)}(\mathbf{q}) + \frac{2w}{b_\tau^2} t_{\mathbf{q}} \cdot S_{W(t_{\mathbf{q}})}^{(1)}(\mathbf{q}) - \frac{w}{b_\tau^2} \cdot S_{W(t_{\mathbf{q}})}^{(2)}(\mathbf{q})$$

$$S_{W(t_{\mathbf{q}})}^{(i)}(\mathbf{q}) = \sum_{(\mathbf{p}, t_{\mathbf{p}}) \in W(t_{\mathbf{q}})} t_{\mathbf{p}}^i \cdot K_{\text{space}}(\mathbf{q}, \mathbf{p})$$

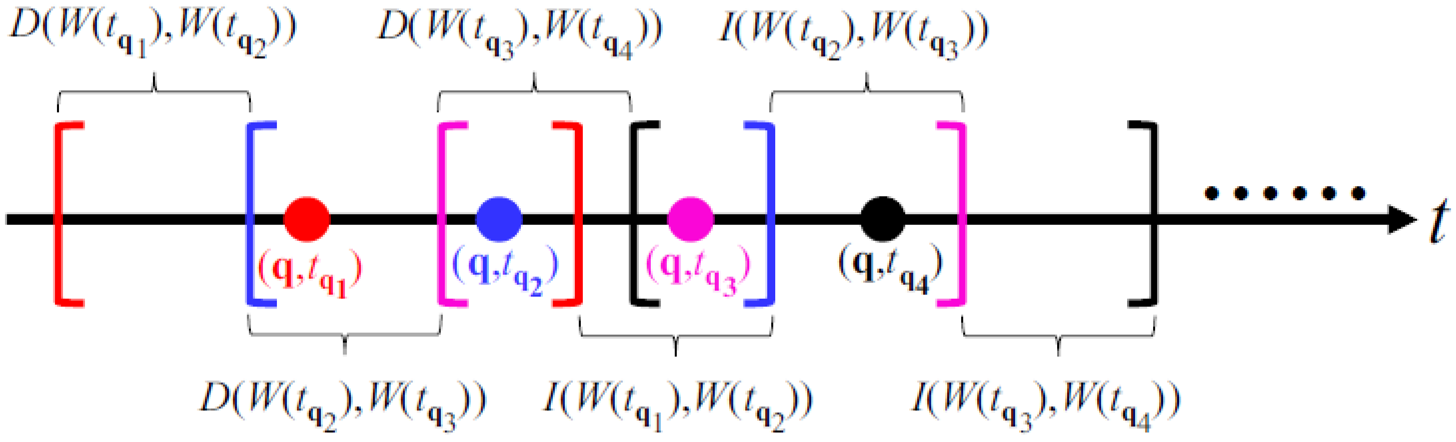
Core Idea 2 of SWS: Incremental Computation



$$S^{(i)}_{W(t_{\mathbf{q}n})}(\mathbf{q}) = S^{(i)}_{W(t_{\mathbf{q}})}(\mathbf{q}) - \sum_{(\mathbf{p},t_{\mathbf{p}})\in D\left(W(t_{\mathbf{q}}),W(t_{\mathbf{q}n})\right)} t^i_{\mathbf{p}} \cdot K_{\mathrm{space}}(\mathbf{q},\mathbf{p}) + \sum_{(\mathbf{p},t_{\mathbf{p}})\in I\left(W(t_{\mathbf{q}}),W(t_{\mathbf{q}n})\right)} t^i_{\mathbf{p}} \cdot K_{\mathrm{space}}(\mathbf{q},\mathbf{p})$$

The time complexity is $O\left(\left|I\left(W(t_{\mathbf{q}}),W(t_{\mathbf{q}n})\right)\right| + \left|D\left(W(t_{\mathbf{q}}),W(t_{\mathbf{q}n})\right)\right|\right)$

Core Idea 2 of SWS: Incremental Computation



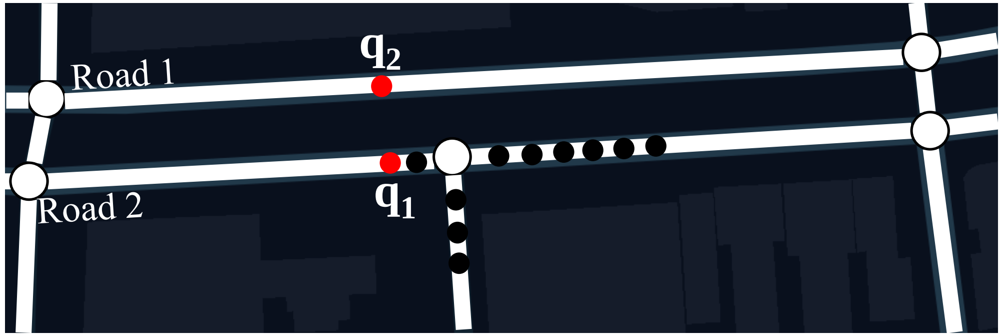
The time complexity is $O\left(\left|W_{t_{\mathbf{q}_1}}\right| + \sum_{i=1}^{T-1}\left|I\left(W(t_{\mathbf{q}_i}),W(t_{\mathbf{q}_{i+1}})\right)\right| + \sum_{i=1}^{T-1}\left|D\left(W(t_{\mathbf{q}_i}),W(t_{\mathbf{q}_{i+1}})\right)\right| + T\right)$

$= O(T+n)$

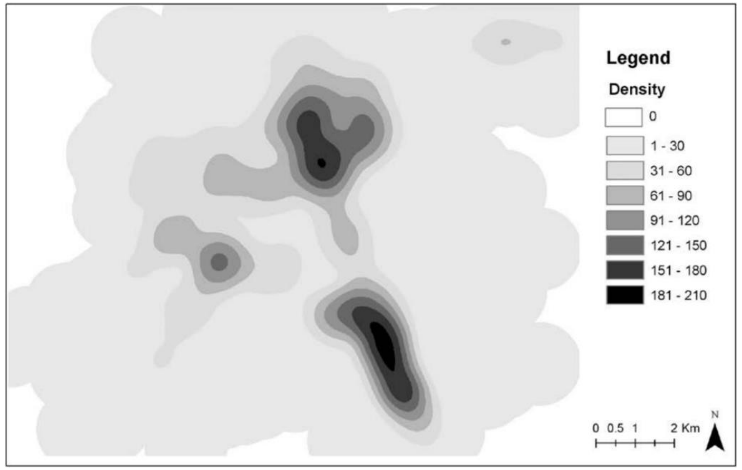
There are $X \times Y$ pixels $\Rightarrow$ Generating STKDV is $O(XY(T+n))$ time ☺

Variant 2: Network Kernel Density Visualization (NKDV)

- KDV ignores the road network
  1. Can overestimate the density value of some regions (e.g., $\mathbf{q}_2$)



  2. Cannot correctly identify which road segments are the hotspot.



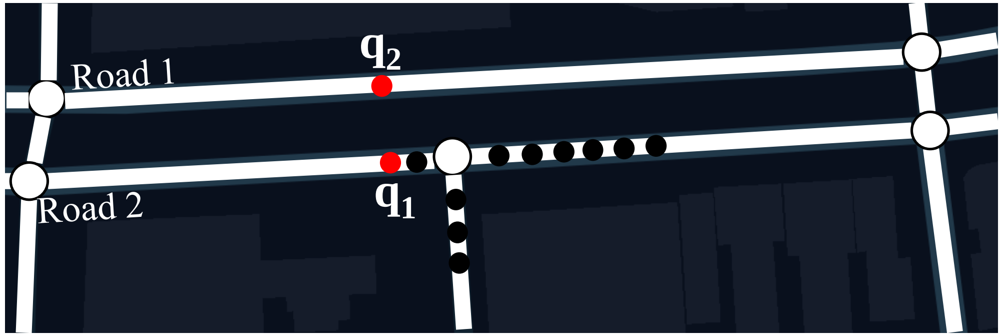
Kernel Density Visualization (KDV)

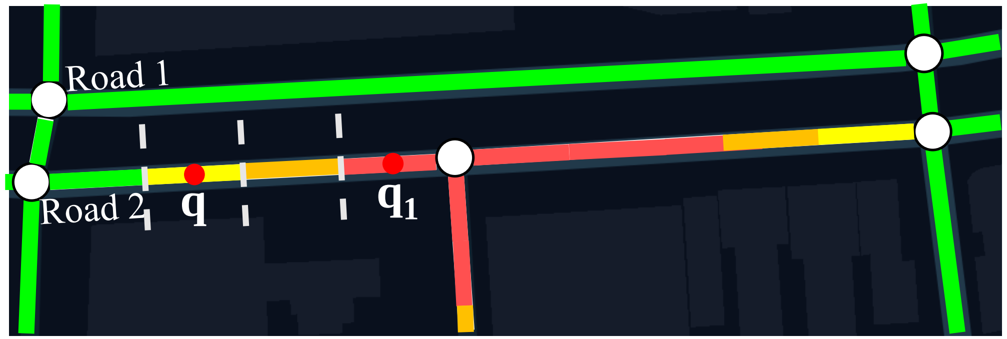
Network Kernel Density Visualization (NKDV)

Variant 2: Network Kernel Density Visualization (NKDV)

- Divide each road in the road network $G = (V, E)$ into a set of lixels.
- Color each lixel $\mathbf{q}$, based on the network kernel density function.

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





Variant 2: Network Kernel Density Visualization (NKDV)

- Time complexity of a naïve solution is $O(L(T_{\text{SP}} + n))$ (Very slow!) ☹
  - L is the number of lixels.
  - T_{SP} is the time complexity of a shortest path algorithm.
  - n is the number of data points.

- Time complexity of the best solution, ADA [**VLDB21a**], is $O\left(|E|\left(T_{\text{SP}} + L\log\left(\frac{n}{|E|}\right)\right)\right)$ time (Why?).

$$O\left(\log\left(\frac{n}{|E|}\right)\right) \quad < \quad O\left(\frac{n}{|E|}\right)$$

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

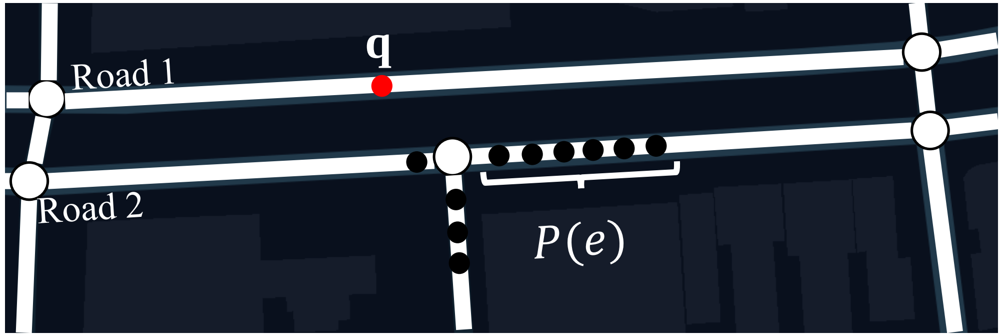
[VLDB21a] T. N. Chan, Z. Li, L. H. U, J. Xu, R. Cheng. Fast Augmentation Algorithms for Network Kernel Density Visualization. VLDB 2021.

Core Idea 1 of ADA: Decomposition of Kernel Density Function

- The kernel density function can be represented by multiple edge-e kernel density functions.

$$\mathcal{F}_P(\mathbf{q}) = \sum_{\mathbf{p}\in P} w \cdot \begin{cases} 1 - \frac{1}{b^2} dist_G(\mathbf{q},\mathbf{p})^2 & \text{if } dist_G(\mathbf{q},\mathbf{p}) \leq b \\ 0 & \text{otherwise} \end{cases}$$

$$\mathcal{F}_P(\mathbf{q}) = \sum_{e\in E} \sum_{\mathbf{p}\in P(e)} w \cdot \begin{cases} 1 - \frac{1}{b^2} dist_G(\mathbf{q},\mathbf{p})^2 & \text{if } dist_G(\mathbf{q},\mathbf{p}) \leq b \\ 0 & \text{otherwise} \end{cases}$$



$$\mathcal{F}_P(\mathbf{q}) = \sum_{e\in E} f_e(\mathbf{q}) \quad \text{where} \quad f_e(\mathbf{q}) = \sum_{\mathbf{p}\in P(e)} w \cdot \begin{cases} 1 - \frac{1}{b^2} dist_G(\mathbf{q},\mathbf{p})^2 & \text{if } dist_G(\mathbf{q},\mathbf{p}) \leq b \\ 0 & \text{otherwise} \end{cases}$$

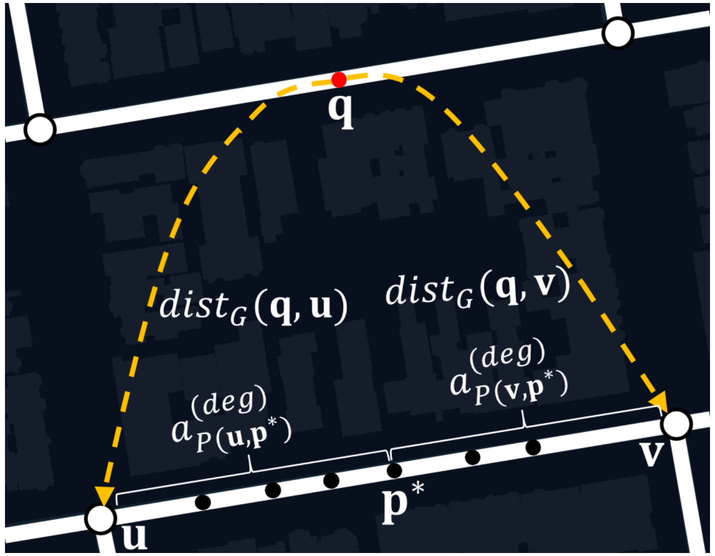
Core Idea 2 of ADA: Binary Search

- Consider one possible case ($dist_G(\mathbf{q},\mathbf{u}) \leq b$ and $dist_G(\mathbf{q},\mathbf{v}) > b$).

$$f_e(\mathbf{q}) = \sum_{\mathbf{p}\in P(e)} w \cdot \begin{cases} 1 - \frac{1}{b^2} dist_G(\mathbf{q},\mathbf{p})^2 & \text{if } dist_G(\mathbf{q},\mathbf{p}) \leq b \\ 0 & \text{otherwise} \end{cases}$$

$$= \sum_{\mathbf{p}\in P(\mathbf{u},\mathbf{p}^*)} w \cdot \left(1 - \frac{1}{b^2}\left(dist_G(\mathbf{q},\mathbf{u}) + dist_G(\mathbf{u},\mathbf{p})\right)^2\right)$$

$$= w\left(1 - \frac{dist_G(\mathbf{q},\mathbf{u})^2}{b^2}\right) a^{(0)}_{P(\mathbf{u},\mathbf{p}^*)} - \frac{2w \cdot dist_G(\mathbf{q},\mathbf{u})}{b^2} a^{(1)}_{P(\mathbf{u},\mathbf{p}^*)} - \frac{w}{b^2} a^{(2)}_{P(\mathbf{u},\mathbf{p}^*)}$$



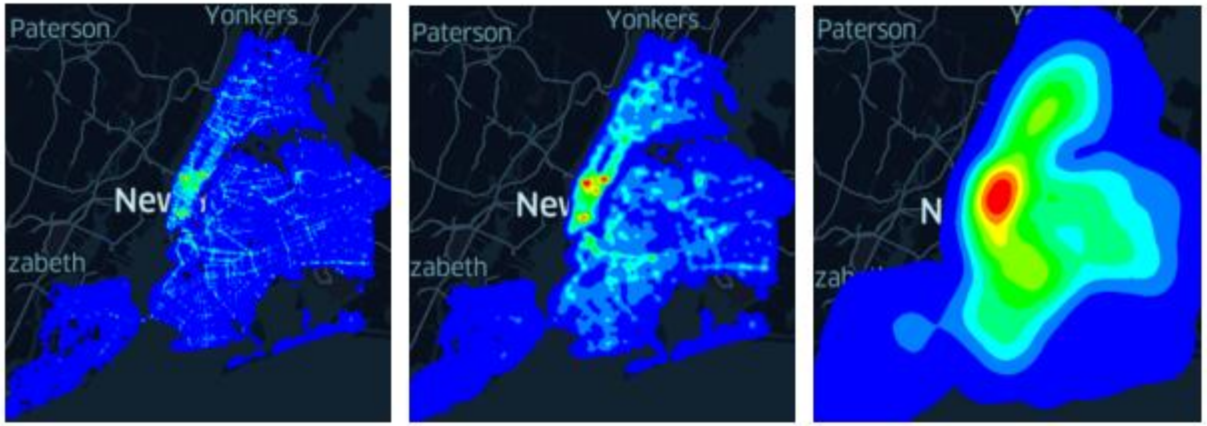
where $a^{(deg)}_{P(\mathbf{u},\mathbf{p}^*)} = \sum_{\mathbf{p}\in P(\mathbf{u},\mathbf{p}^*)} dist_G(\mathbf{u},\mathbf{p})^{deg}$

- Can compute $f_e(\mathbf{q})$ in $O(\log|P(e)|)$ time! Why?

Software Development of KDV and its Variants

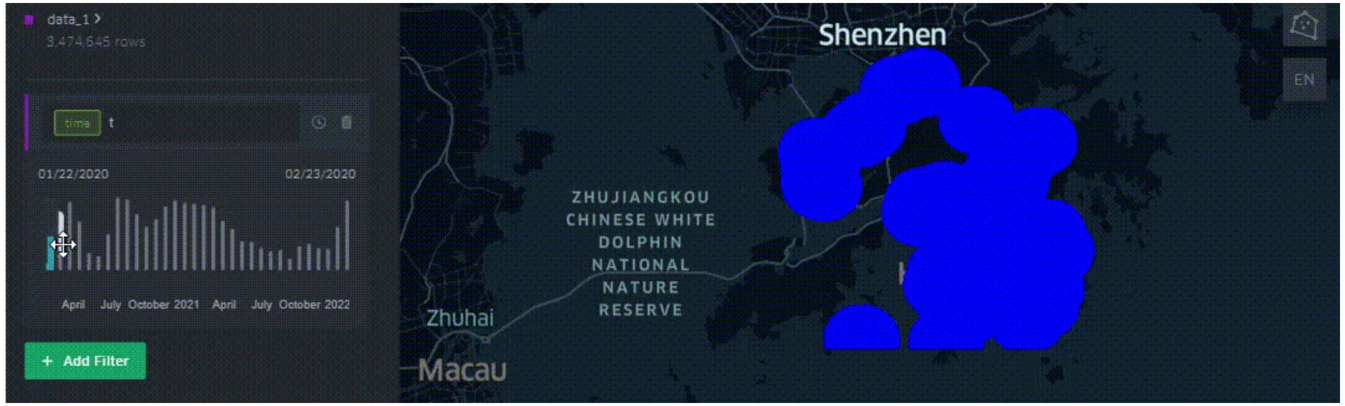
LIBKDV

- A python library for supporting KDV and STKDV.
  - Adopt our solution, SLAM, for computing KDV
  - Adopt our solution, SWS, for computing STKDV
- Webpage: https://github.com/libkdv/libkdv
- Functionalities:



(a) Small b (b) Moderate b (c) Large b

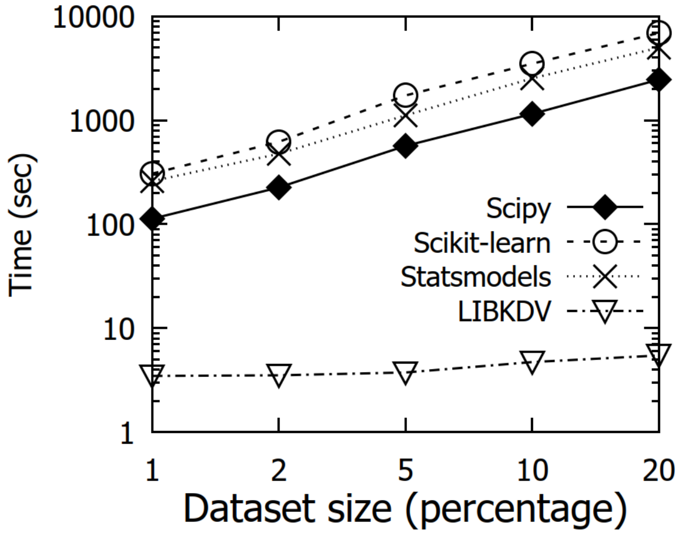
Generate multiple KDVs (based on LIBKDV) with different bandwidths b for the New York traffic accident dataset.



Generate STKDV (based on LIBKDV) with different bandwidths b for the Hong Kong COVID-19 cases

LIBKDV

- Fast ☺



- Easy to use ☺

```
NewYork = pd.read_csv('./Datasets/New_York.csv')
traffic_kdv = kdv(NewYork,KDV_type="KDV",bandwidth_s=1000)
traffic_kdv.compute()
```

KDV

```
libkdv_obj = kdv(dataset, KDV_type,
                 GPS=true,
                 bandwidth_s=1000, row_pixels=800, col_pixels=640,
                 bandwidth_t=6, t_pixels=32,
                 num_threads=8)
libkdv_obj.compute()
```

STKDV

Hong Kong and Macau COVID-19 Hotspot Maps

- Websites:
  - Hong Kong version (https://covid19.comp.hkbu.edu.hk/)
  - Macau version (http://degroup.cis.um.edu.mo/covid-19/)

- Powered by LIBKDV (https://github.com/libkdv/libkdv)

- Can achieve real-time performance (< 0.5 sec) for computing KDV ☺

- Can achieve nearly real-time performance for computing STKDV ☺

Hong Kong and Macau COVID-19 Hotspot Maps

浸大推新冠確診個案分布圖　實時掌握各地區風險水平

新聞觀看次數：4.5k

11月14日(一) 13:43　推介 9　分享　Tweet　分享

浸大推出「香港新冠病毒熱點分析圖」，可呈現確診個案的地理位置分布。

新冠肺炎疫情仍未平息，為助公眾了解不同地區的感染風險，香港浸會大學領導的研究團隊推出「香港新冠病毒熱點分析圖」，以直觀、實時和動態的方式，呈現新冠病毒個案的地理位置分布。此線上地圖有助及時和準確地掌握新冠感染個案位置分布資訊，並根據新冠個案

Oriental Daily Hong Kong (in Chinese)

浸大推確診分析圖 實時掌握各區風險

【大公報訊】香港浸會大學領導的研究團隊推出「香港新冠病毒熱點分析圖」，以直觀、實時和動態的方式，呈現新冠病毒個案分布。該地圖採用由團隊開發的時空大數據分析演算法，其運算時間，較現有最先進的方法快100倍。研究成果已發表於今年舉行的兩個大數據管理領域最頂級國際會議「國際數據管理會議」及「國際超大型數據庫會議」。

運算時間較現時快100倍

「香港新冠病毒熱點分析圖」由浸大計算機科學系系主任徐建良教授領導的團隊開發，目的是在線上地圖顯示出全港新冠病毒感染個案的數據。團隊的其他浸大學者包括計算機科學系副系主任蔡冠球教授及研究助理教授陳梓楠博士。此地圖亦由澳門大學及香港大學共同開發。

該分析圖以政府發布的香港互動地圖儀表板作為實時數據，直觀、實時和動態化地呈現新型冠狀病毒個案的地理位置分布。然而，現時應用於時空數據分析的「核密度可視化」計算工具，未能支援「香港新冠病毒熱點分析圖」運行所需，故浸大領導的團隊共同開發出一套新的演算方法。新算法配合漸進式可視化框架，產生持續的局部成像，以減少「核密度可視化」的運算時間。團隊運用大規模數據集進行實驗，結果顯示新算法的運算時間，較現有最先進的方法快100倍。而解像度亦提高至1376×960像素（高清解像度），並能以少於0.5秒的計算時間處理100萬個數據點。

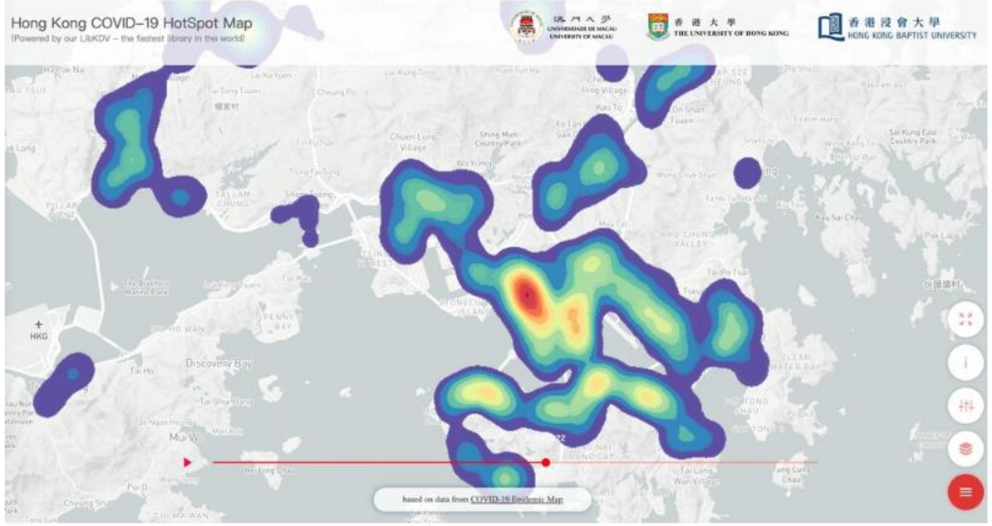
徐教授表示，新開發的演算法可以支援更多以「核密度可視化」為基礎的時空大數據分析工作。例如，交通熱點偵測，景區人流控制，樓價可視化分析，以及實時氣象資源管理等。

▲浸大領導的研究團隊推出「香港新冠病毒熱點分析圖」，可實時以動態方式呈現新冠病毒個案分布。

Ta Kung Pao News (in Chinese)

HKBU-led research team launches Hong Kong COVID-19 hotspot map

Local | 14 Nov 2022 7:16 pm



A research team led by Hong Kong Baptist University has launched the Hong Kong COVID-19 Hotspot Map, which allows the visualisation of the real-time and dynamic geographic distribution of Covid cases in the city.

The standard

Hong Kong and Macau COVID-19 Hotspot Maps

Macau TDM (Video news in Cantonese)

UM COVID-19 research team proposes key points for combating epidemic

NEWS PROVIDED BY
Macao Government News
July 15, 2022, 21:48 GMT

SHARE THIS ARTICLE

MACAU, July 15 - Omicron BA.5 is raging around the world and Macao has also been deeply affected by it. To closely monitor the impact of the virus on the community, the COVID-19 research team of the University of Macau (UM) continues to analyse the latest development trend of the epidemic in the city and underscores some key areas of concerns.

The research team points out that since the discovery of five positive cases in Macao on 18 June, the number of new single-day positive cases had been increasing, with the highest number of new single-day positive cases recorded on 5 July at 146. Since the implementation of 'relatively static' management measures on 9 July, the number of new daily positive cases has been gradually decreasing, with 29 new positive cases on 13 July. Between 18 June and 13 July, 1,644 positive cases have been recorded in the current outbreak, of which over 60 per cent were classified as non-symptomatic. Based on a cautious reading of the development trend of the epidemic in Macao, there are three points worth noting:

1. Strictly adhere to the principles of epidemic prevention to avoid a large increase of cases: Although BA.5 is more transmissible than other subvariants, there has not been an exponential increase in the number of positive cases in Macao as there has been in other affected areas. This reflects the key role that Macao's long-standing anti-epidemic principle of early detection, early

Contact

email us here

More From This Source

Employment survey for August-October 2022

MICE statistics for the 3rd quarter of 2022

[Infographic] Protective measures as part of routine anti-epidemic steps -- Advice for participants in festive, ...

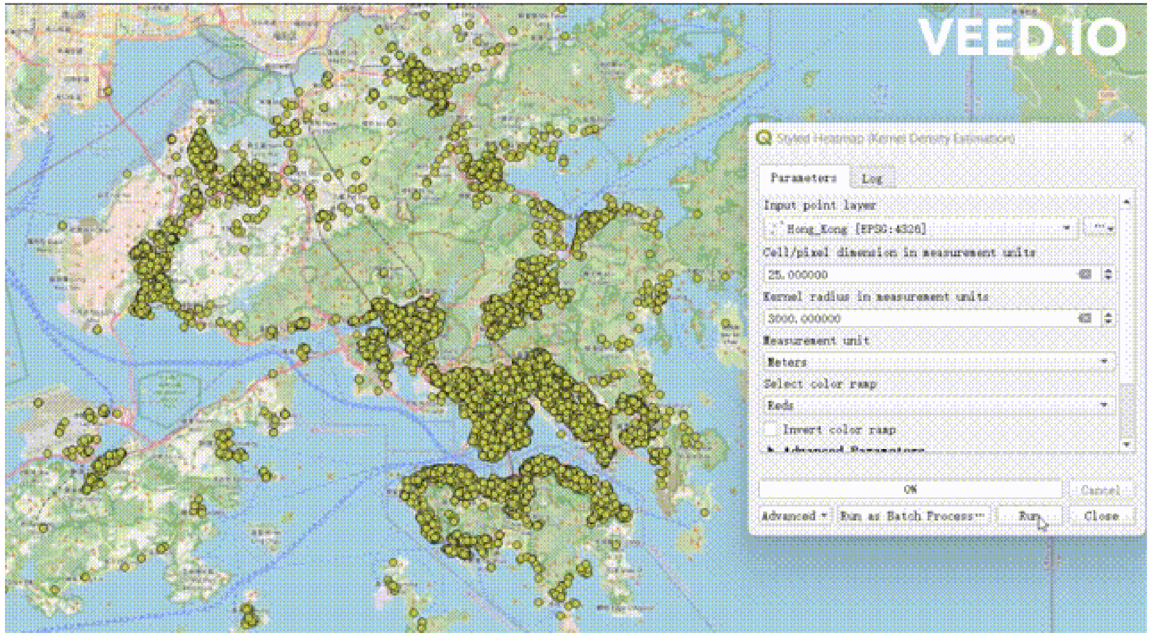
View All Stories From This Source

Newswires

A Plugin for GIS Systems

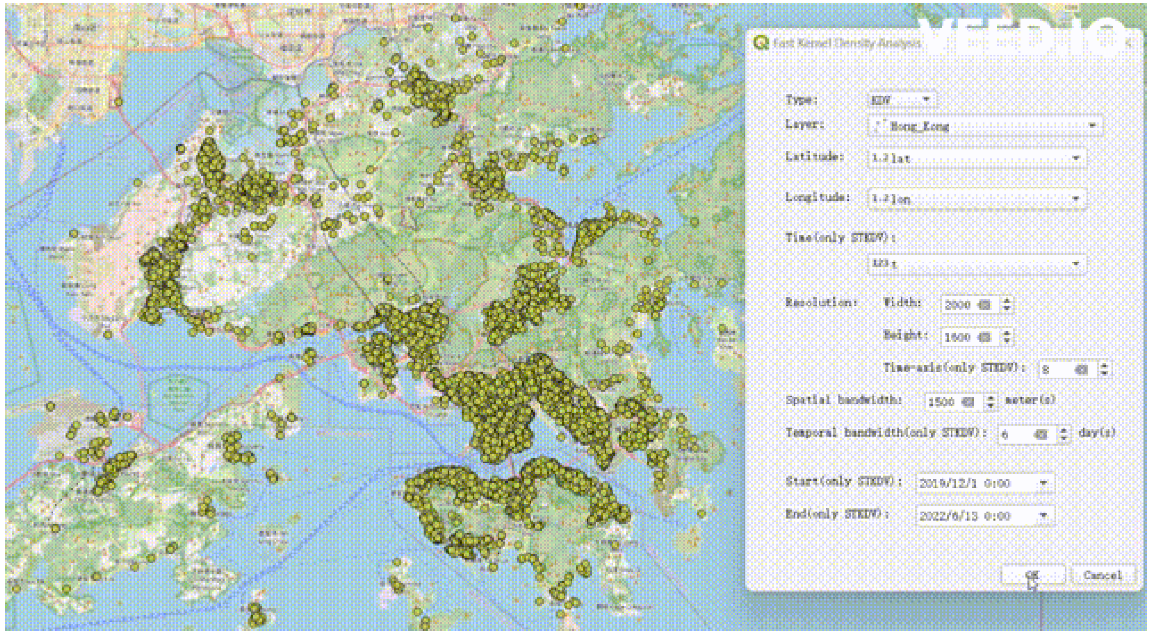
- The fastest plugin for generating KDV ☺

5x



QGIS internal function for KDV (3 minutes 32 seconds)
with the resolution size 1994 x 1566

5x

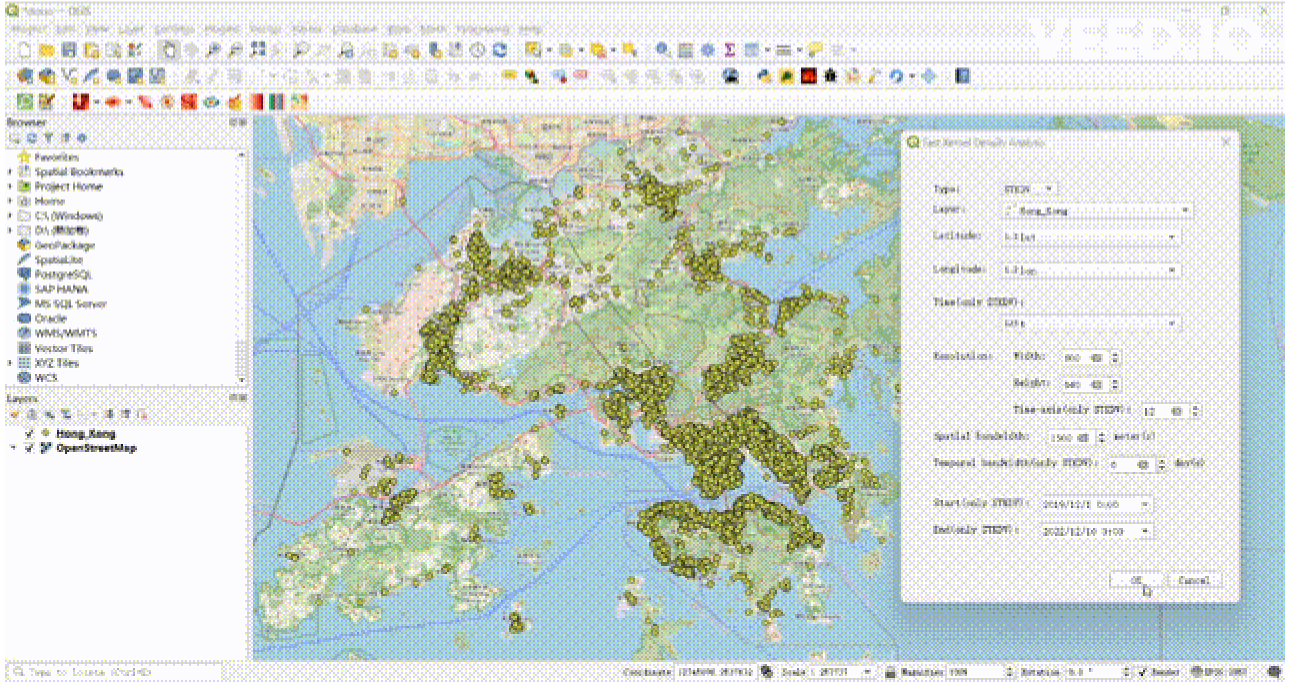


Our KDV plugin for QGIS (50 seconds)
with the resolution size 2000 x 1600

A Plugin for GIS Systems

- The fastest plugin for generating STKDV ☺

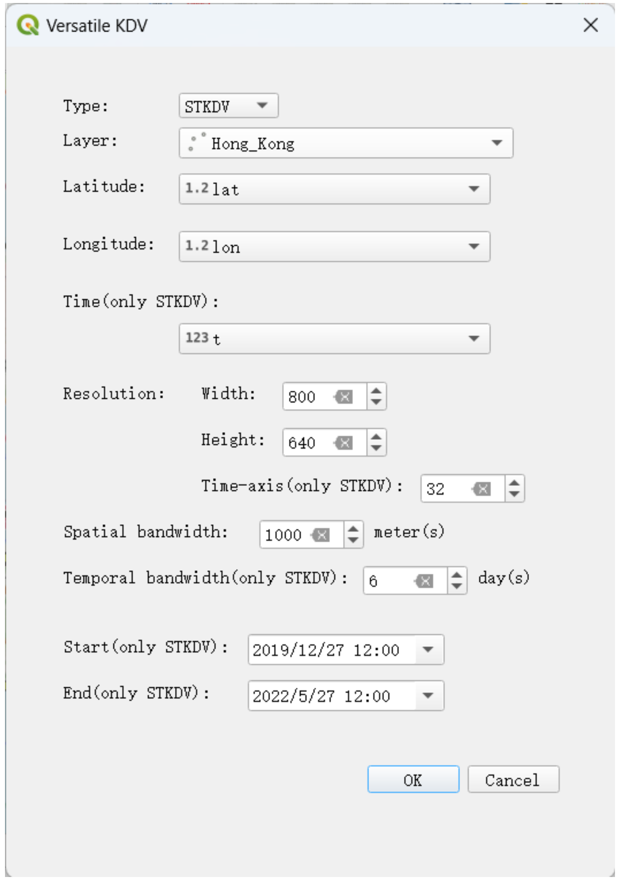
10x



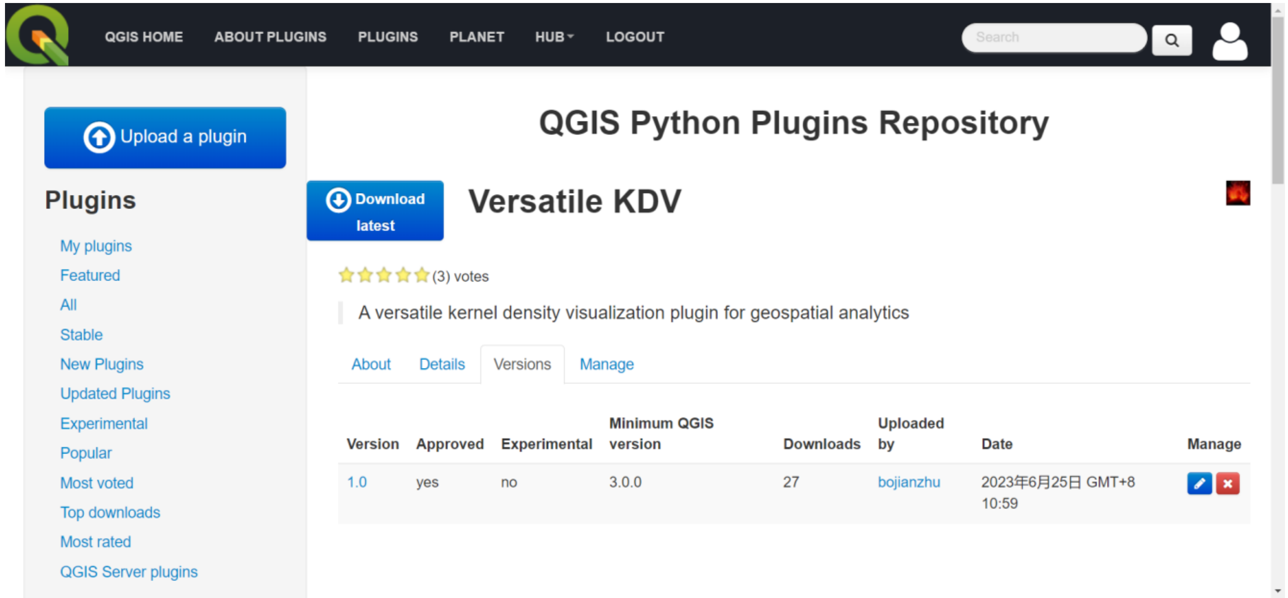
Our KDV plugin for QGIS (128 seconds) with
the resolution size 2000 x 1600 and 12 timestamps

A Plugin for GIS Systems

- QGIS plugin ([GitHub link]()).



- Approved by QGIS ([link]()).



PyNKDV

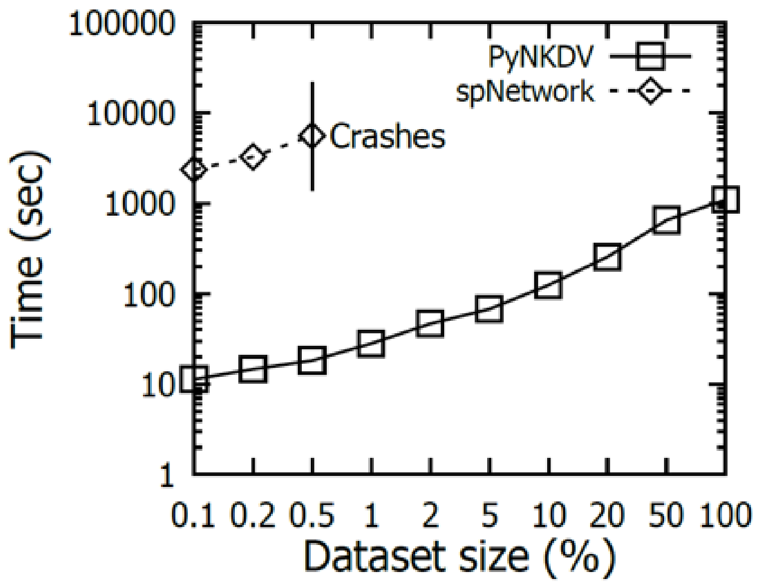
- A python library for generating NKDV (based on our ADA method).

- Webpage: https://github.com/edisonchan2013928/PyNKDV

NKDV for the 311-call location dataset in San Francisco (generated by PyNKDV)

PyNKDV

- Fast ☺



- Easy to use ☺

```
road_data = map_road_network(location_data)
model = PyNKDV(road_data, bandwidth=1000,
                            lixel_size=5, num_threads=8)
results = model.compute()
output(results, output_file_name)
```

Future Opportunities

Future Opportunities for Research

1. Can we further develop the optimal solution for KDV?
   - Current lower bound time complexity: $\Omega(XY + n)$.
   - State-of-the-art upper bound time complexity: $O(Y(X + n))$/ $O(X(Y + n))$.

2. Can we further develop the optimal solution for STKDV?
   - Current lower bound time complexity: $\Omega(XYT + n)$.
   - State-of-the-art upper bound time complexity: $O(XY(T + n))$.

3. Can we extend the complexity-reduced solutions to other kernel functions (e.g., Gaussian kernel and exponential kernel)?

Future Opportunities for Research

4. Can we extend our solution to support other types of spatial visualization tasks?
    - Kriging
    - Inverse distance weighting (IDW)

5. Can we extend our solution to support other spatial analysis tasks?
    - K-function
    - DBSCAN clustering

Future Opportunities for Software Development

1. Can we further develop some python libraries to support more visualization tools and data analysis operations (e.g., Kriging, IDW, and K-function)?

2. Can we further integrate our libraries in (1) into the commonly used software packages?
   - Develop plugins for QGIS and ArcGIS
   - Integrate our methods into Scikit-learn and Scipy.

3. Can we further develop some R packages for supporting those operations in (1)?

Future Opportunities for Software Development

4. Can we further extend our web-based system (Hong Kong COVID-19 hotspot map) to support more visualization tools and data analysis operations?

5. (**Long-term goal**) Can we develop a software package (like ArcGIS and Scikit-learn) that includes our complexity-reduced algorithms for different operations?