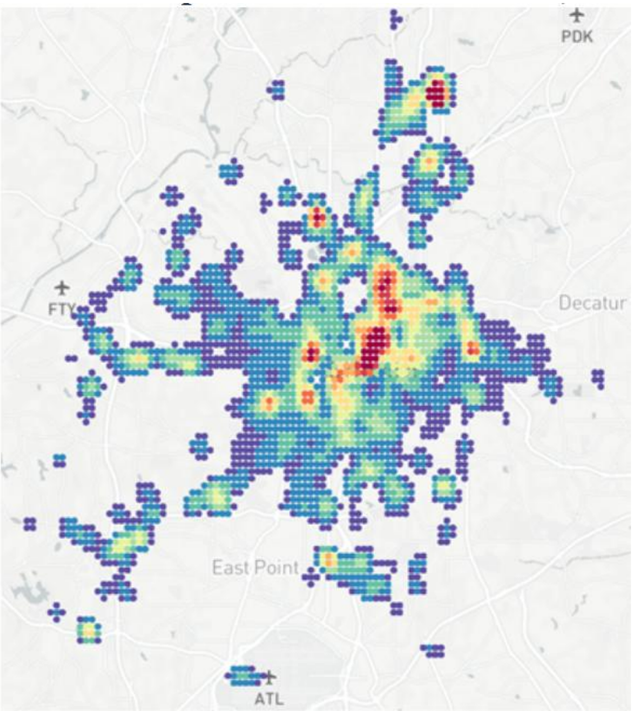
Data Visualization

Why Data Visualization?

- Easy to understand/discover the hidden information from data
- More impressive and intuitive
- Guide the data analysts to analyze data (first step for many data analysis tasks)

270688	2
33.77101	-84.38895
33.74057	-84.4168
33.71803	-84.40774
33.70731	-84.39674
33.75947	-84.36626
33.82838	-84.40133
33.70537	-84.45498
33.70121	-84.45724
33.83193	-84.42627
33.7604	-84.38746
33.76309	-84.3516
33.77725	-84.46072
33.84922	-84.36056
33.82674	-84.36131
33.75946	-84.38769
33.77101	-84.38895
33.74075	-84.39454
33.70604	-84.35916
33.82543	-84.36706
33.73611	-84.37812
33.75795	-84.34501
33.8199	-84.35326
33.80253	-84.39776
33.77948	-84.49901
33.76089	-84.38855
33.69935	-84.40073
33.7456	-84.40378
33.77368	-84.38477

Crime events in Atlanta, USA



Density visualization of crime events in Atlanta, USA

Visualization Tools

- **Scatter plot** (link)
- **Histogram** (link)
- **Kernel density visualization (KDV)**
  - **Spatial kernel density visualization (SKDV)** (link 1, link 2, link 3, link 4, link 5)
  - **Spatiotemporal kernel density visualization (STKDV)** (link 1, link 2, link 3)
  - **Network kernel density visualization (NKDV)** (link 1, link 2)
  - Spatiotemporal network kernel density visualization (STNKDV) (link)
- Kriging (link)

Scatter Plot

- Directly plots data points in the map

Scatter plot of the data points of
1854 London cholera epidemic

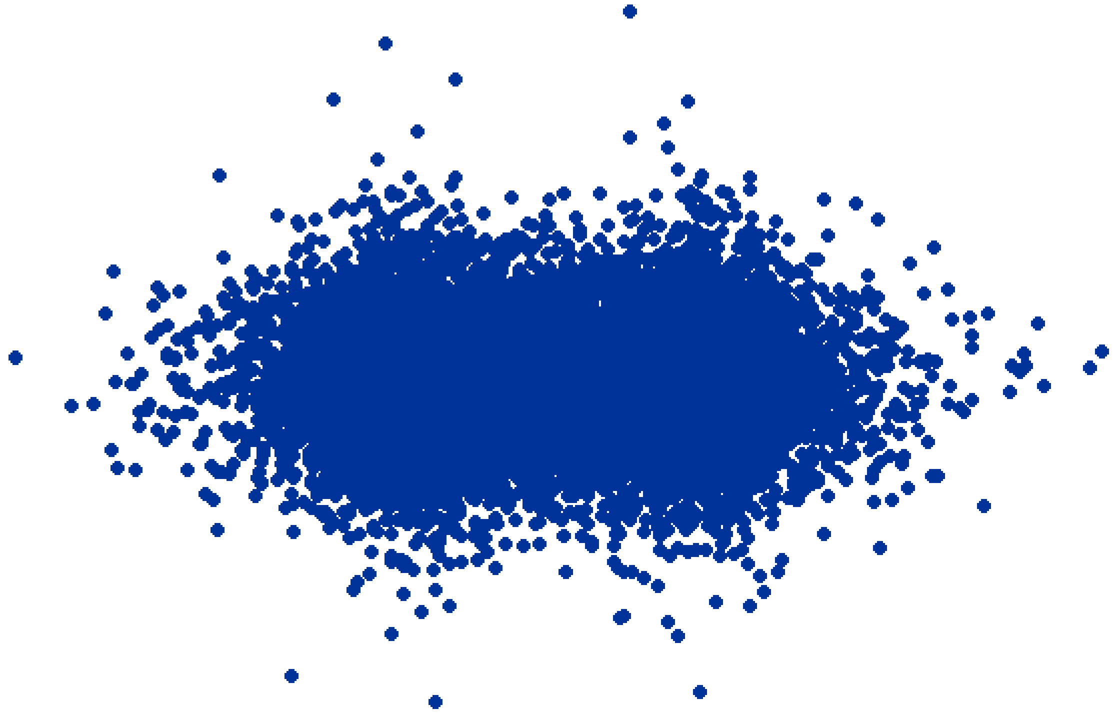
[1] A. Perrot, R. Bourqui, N. Hanusse F. Lalanne D. Auber, "Large Interactive Visualization of Density Functions on Big Data Infrastructure" LDAV2015

Advantages of Scatter Plot

- Simple ☺
- Show the patterns clearly for small data ☺
- Time-efficient ☺

Overplotting Issues of Scatter Plot

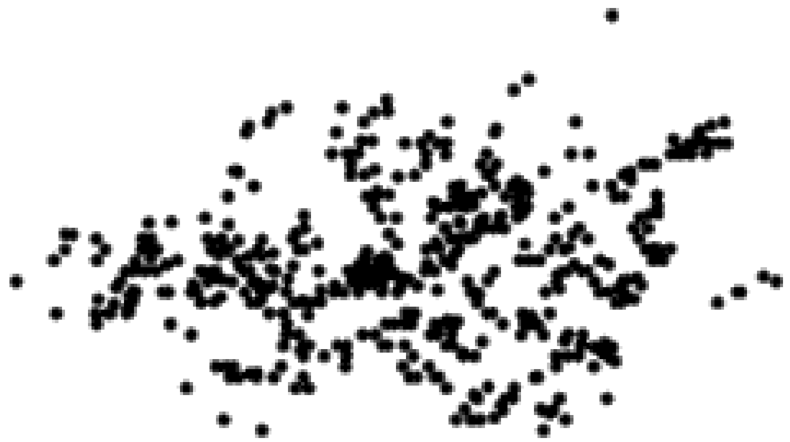
- Difficult to find which parts contain more data points (Overplotting) ☹
  - This issue is more serious if the number of data points is much larger than the resolution size.



Overplotting Issues of Scatter Plot

- Seriously suffer from the resolution changes ☹

Resolution: 512 x 512



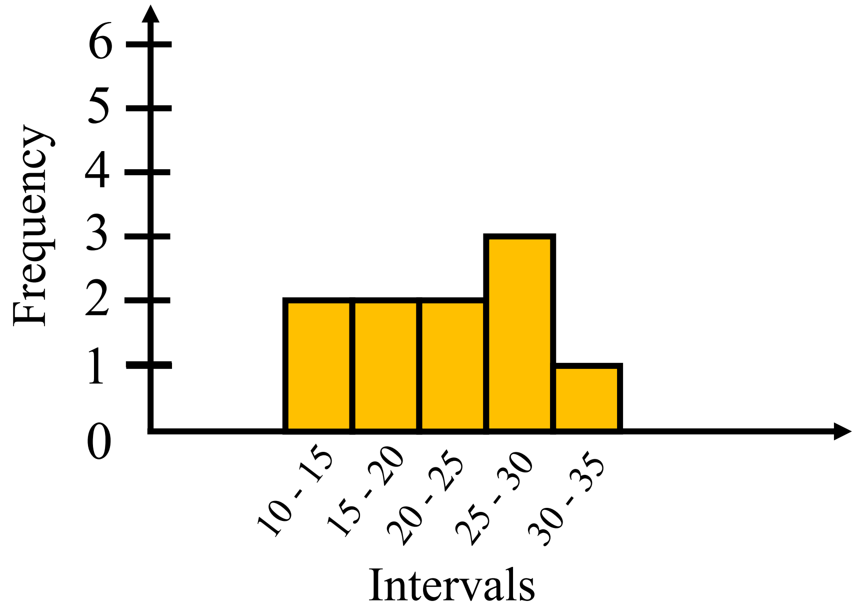
Resolution: 256 x 256

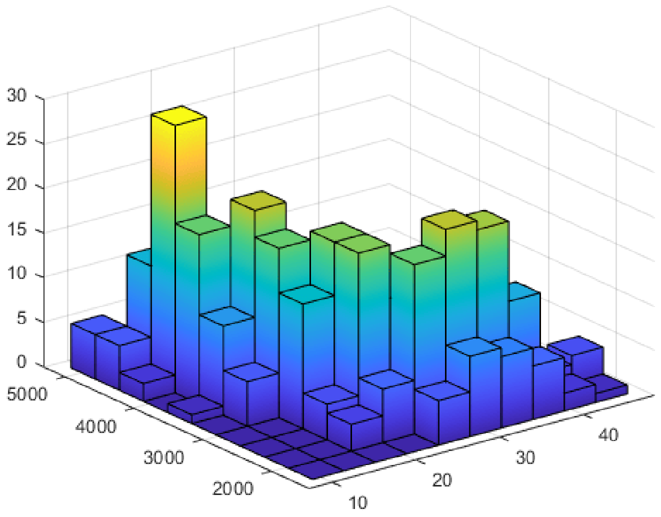
Histogram

- Divide the space into different intervals/ sub-regions with the same size
- Count the frequency in each interval/sub-region
- Example: The grade for students

  12.5, 14.8, 16.1, 16.8, 22.3, 24.1, 26.1, 26.6, 26.9, 31.2

- Generalize to multi-dimensional histogram



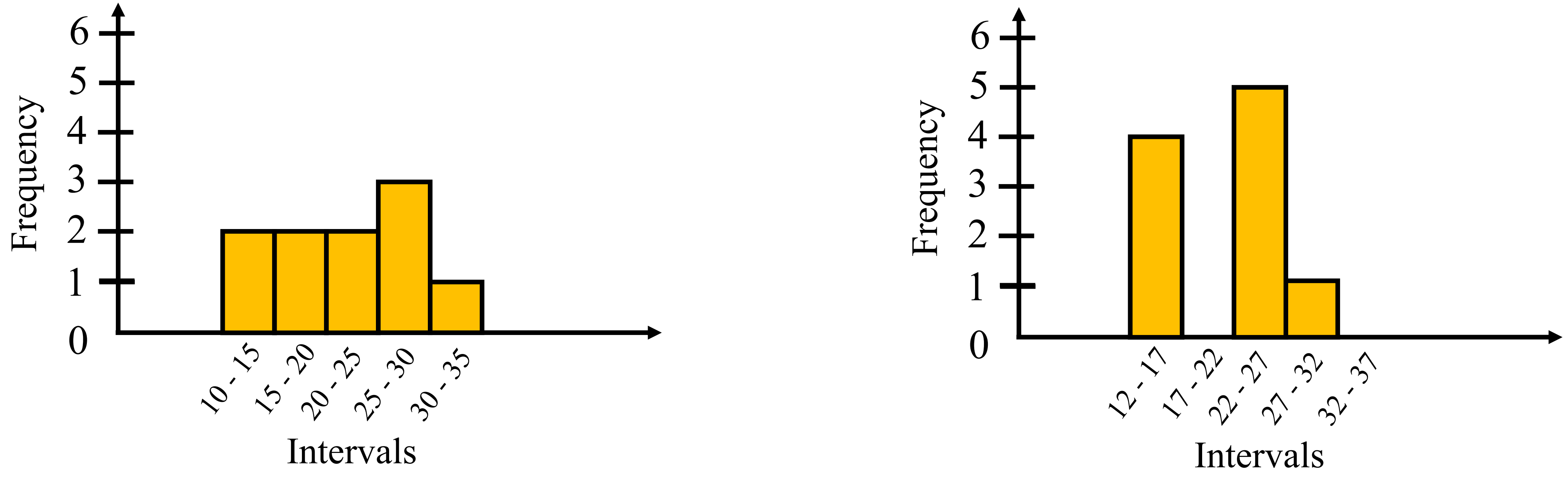


Advantages of Histogram

- Simple ☺
- Time-efficient ☺
- Solve the overplotting issues ☺

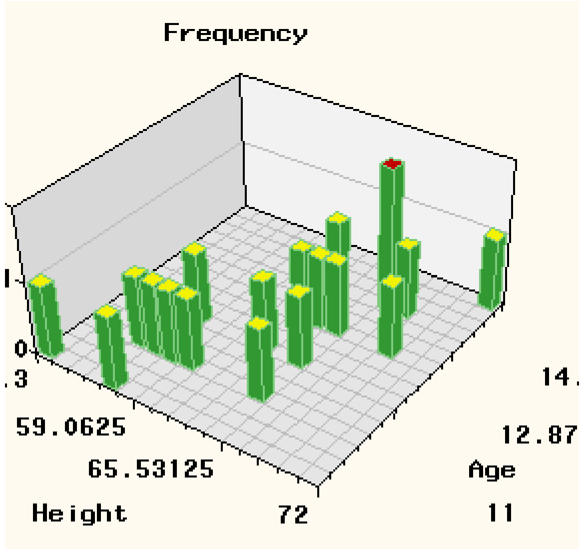
Histogram is Sensitive to the Pixel Positions

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



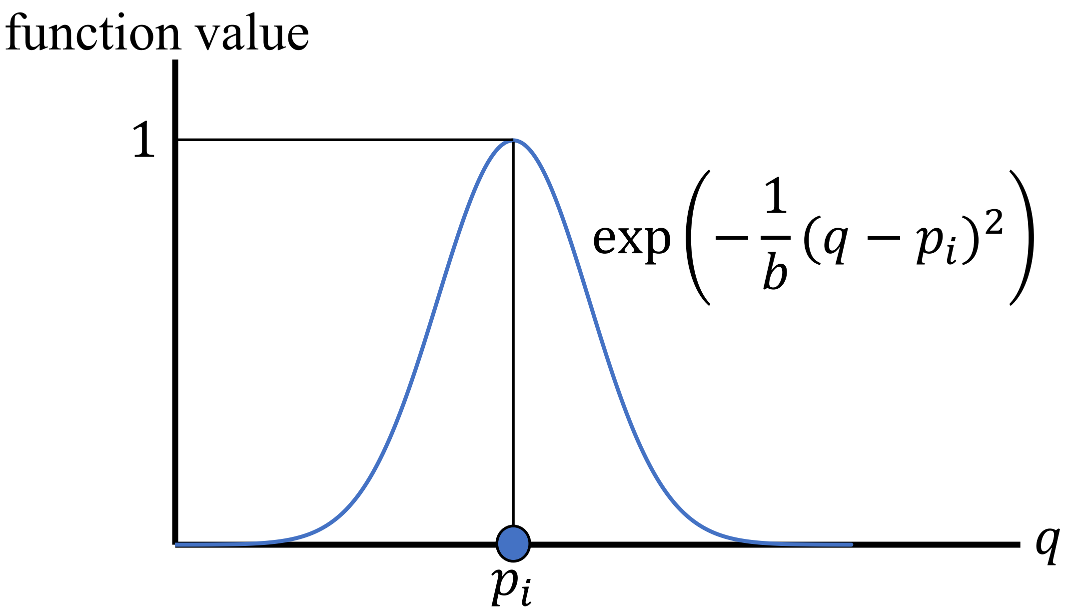
Histogram is Not Smooth

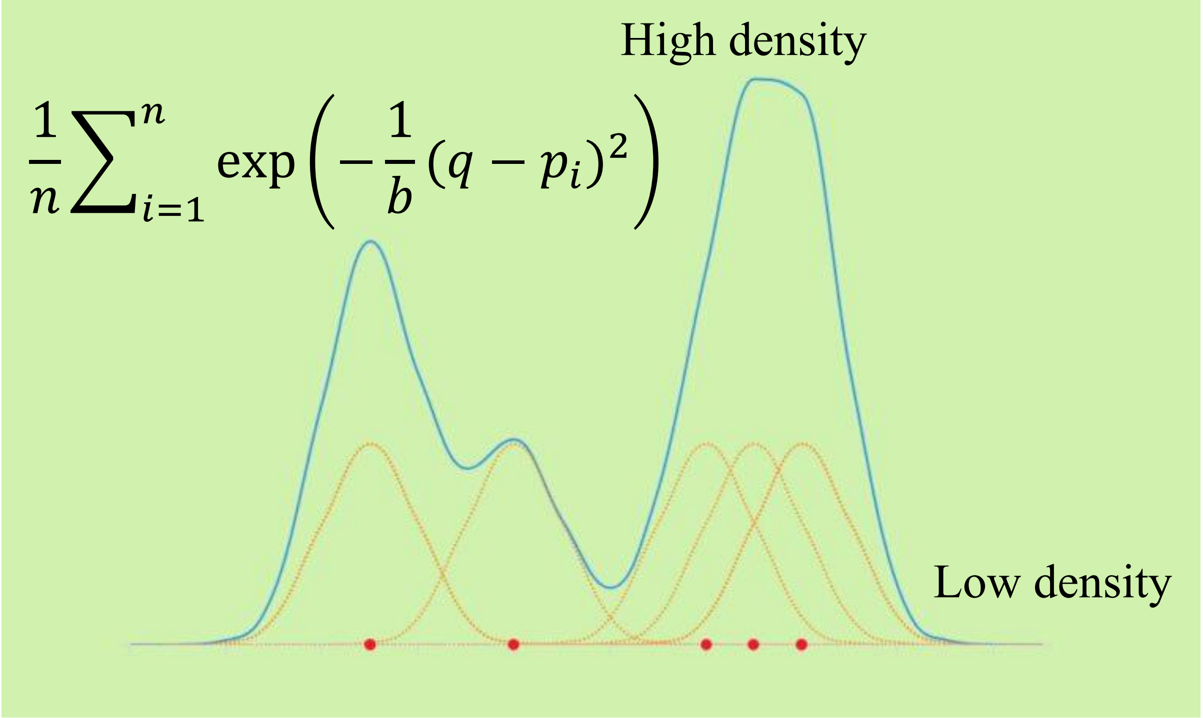
- The visualization is not smooth (There can be a huge change between two consecutive bins) ☹



Spatial Kernel Density Visualization (SKDV)

- Based on kernel density estimation
- One-dimensional case:

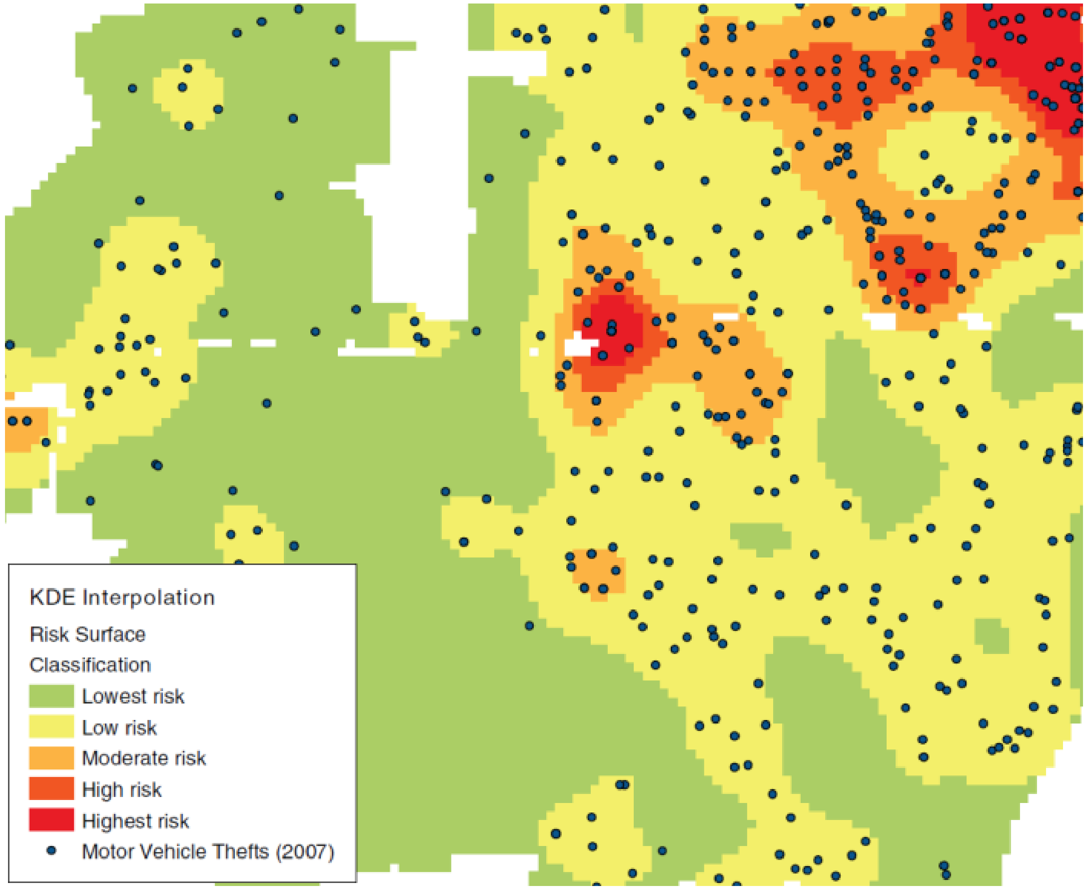




SKDV

- Generalize to two-dimensional case

- Motor vehicle thefts in Arlington, Texas 2007 (link)
  - Each black dot denotes the crime event.
  - Region with red color denotes high density of crime.
  - Region with green color denotes low density of crime.

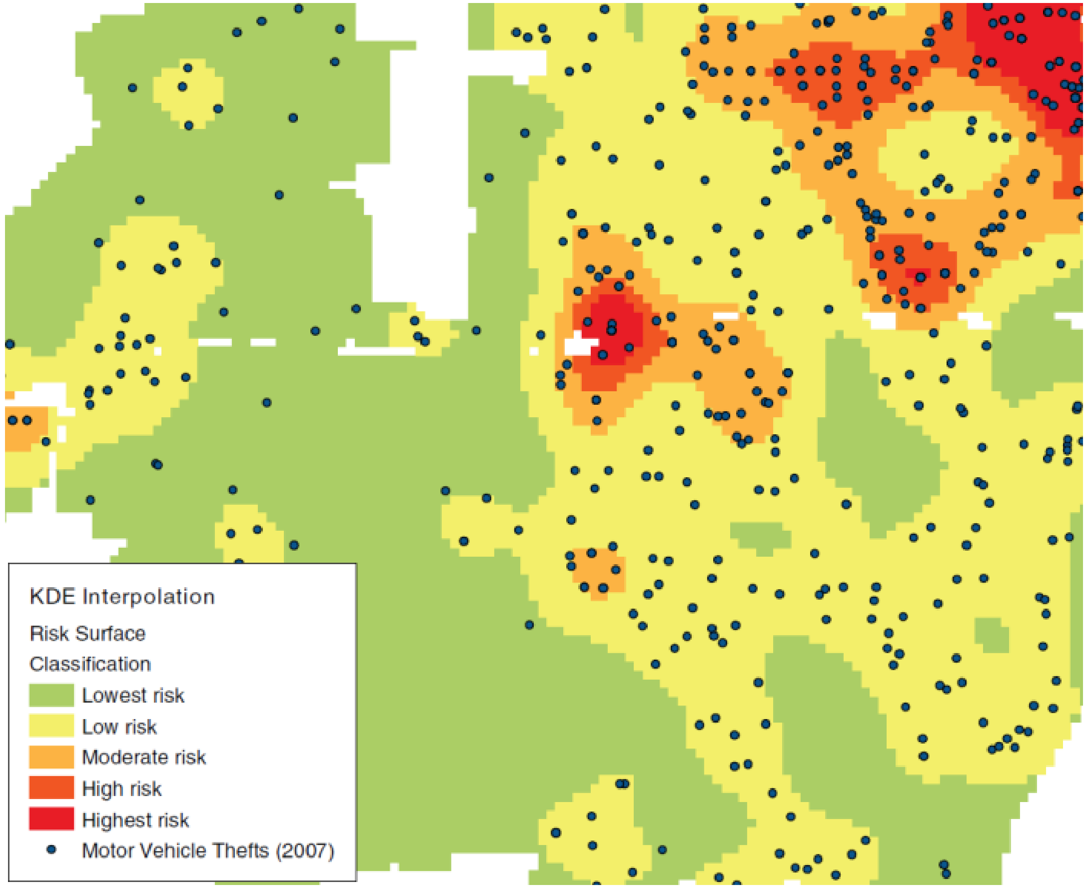


Problem Definition of SKDV

- Given a set of two-dimensional data points $P = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n\}$ with size n, the resolution size $X \times Y$, we need to compute the density of each pixel $\mathbf{q}$ using the following kernel density function.

$$\mathcal{F}_P(\mathbf{q}) = \frac{1}{n} \sum_{\mathbf{p} \in P} K(\mathbf{q}, \mathbf{p})$$

- $K(\mathbf{q}, \mathbf{p})$ is the kernel function.



Representative Kernel Functions

- Uniform kernel function

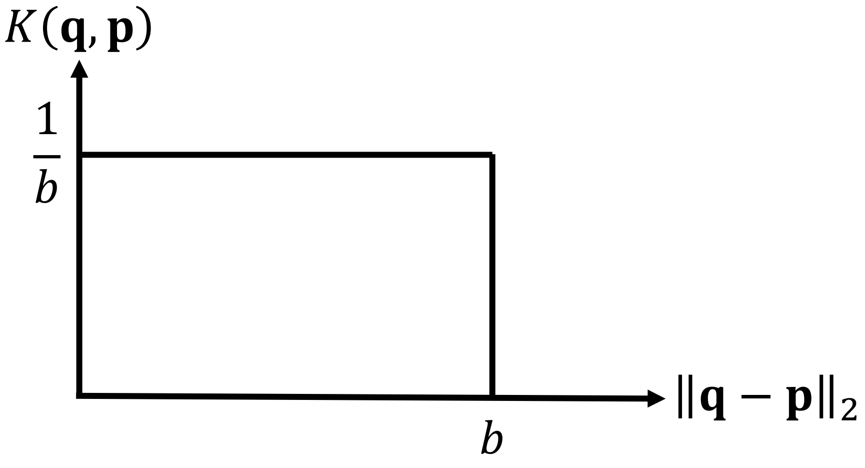
$$K(\mathbf{q},\mathbf{p}) = \begin{cases} \frac{1}{b} & \text{if } \|\mathbf{q}-\mathbf{p}\|_2 \le b \\ 0 & \text{otherwise} \end{cases}$$

- Gaussian kernel function

$$K(\mathbf{q},\mathbf{p}) = \exp\left(-\frac{1}{b^2}\|\mathbf{q}-\mathbf{p}\|_2^2\right)$$

- Epanechnikov kernel

$$K(\mathbf{q},\mathbf{p}) = \begin{cases} 1-\frac{1}{b^2}\|\mathbf{q}-\mathbf{p}\|_2^2 & \text{if } \|\mathbf{q}-\mathbf{p}\|_2 \le b \\ 0 & \text{otherwise} \end{cases}$$



Uniform kernel function

Advantages of SKDV

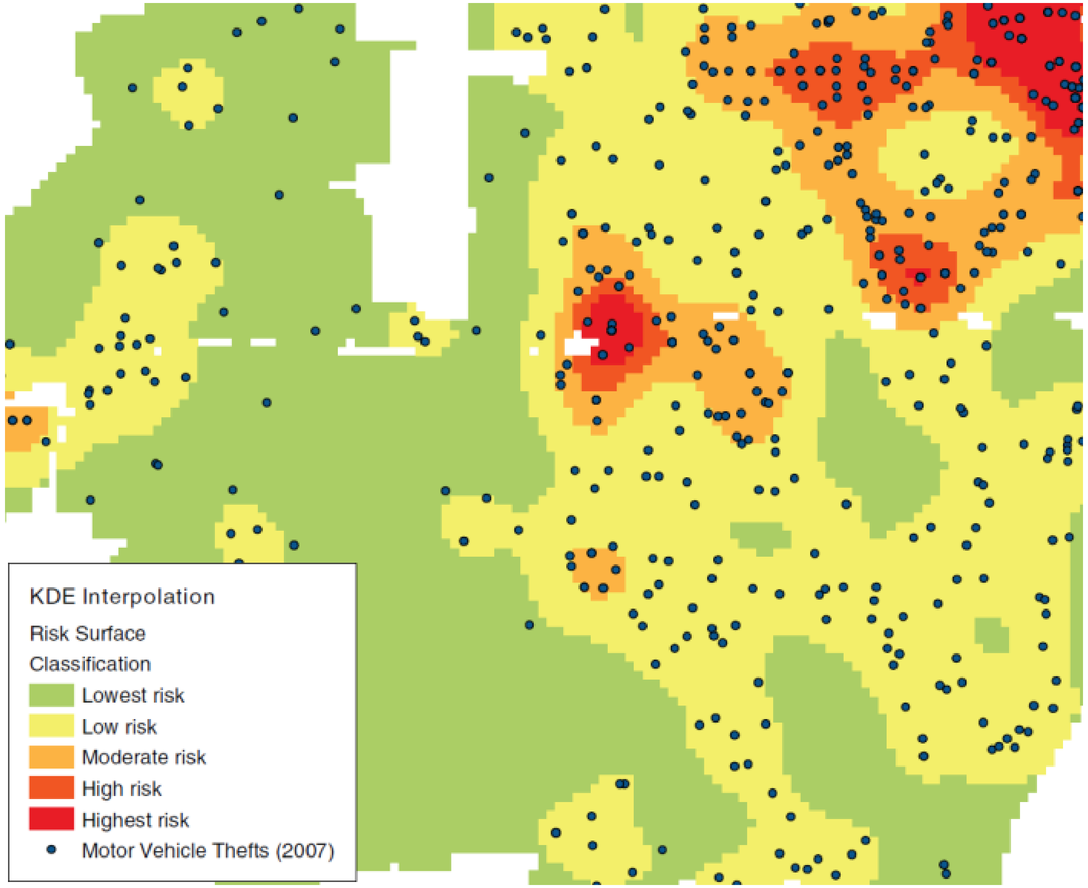
- Solve the overplotting issues ☺

- Slightly shifting the region does not significantly affect the visualization ☺

- Good visualization quality (Smooth) ☺

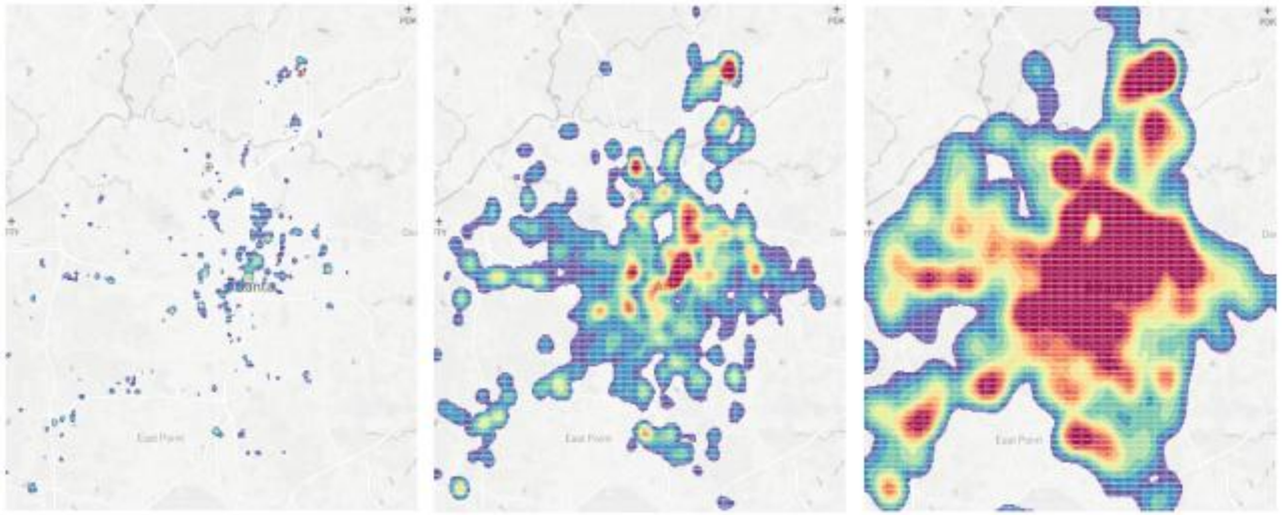
SKDV is Slow

- Resolution size: $X \times Y$
- Number of data points: n
- Time complexity: O(XYn) ☹
- Example:
  $X = 512$, $Y = 512$, and $n = 2000000$
  Time cost = 0.524 **trillion**
  Infeasible to handle this operation



Slow to Tune the Correct Bandwidth Parameter for SKDV

- Bandwidth parameter can significantly affect the visualization quality.



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

- Many domain experts adopt the trial-and-error approach to choose this bandwidth parameter b, which further deteriorates the inefficiency issue (link) ☹

Efficient Algorithms for SKDV

- SAFE (link): the complexity-optimized solution for generating SKDVs with multiple bandwidths using some kernel functions, including uniform kernel and Epanechnikov kernel.

- SLAM (link): the complexity-optimized solution for generating a single SKDV with some kernel functions, including uniform kernel and Epanechnikov kernel.

- QUAD (link): the practically efficient solution for generating a single SKDV with all kernel functions.

No Time Information for SKDV

- Time information is important for many applications.
  - Different waves of COVID-19 cases
  - Crime/ traffic accident blackspot patterns significantly depend on time.

- May provide misleading visualization ☹

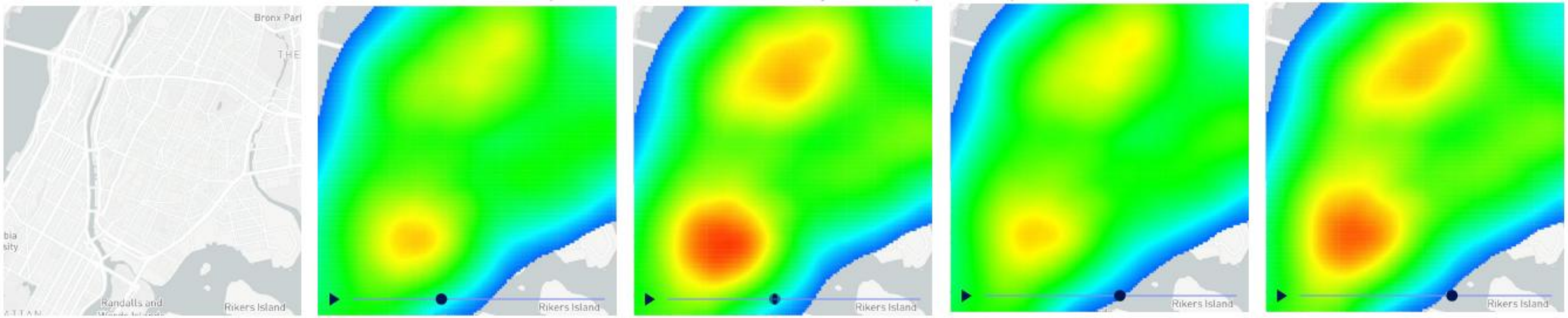
COVID-19 cases in Hong Kong from Feb 2020 to Feb 2021

Spatial-Temporal Kernel Density Visualization (STKDV)

- The visualization of the COVID-19 density distribution in Hong Kong

(a) Hong Kong (b) 1st April 2020 (c) 15th May 2020 (d) 30th July 2020 (e) 1st Oct 2020 (f) 10th Dec 2020

- The visualization of the traffic accident density distribution in New York

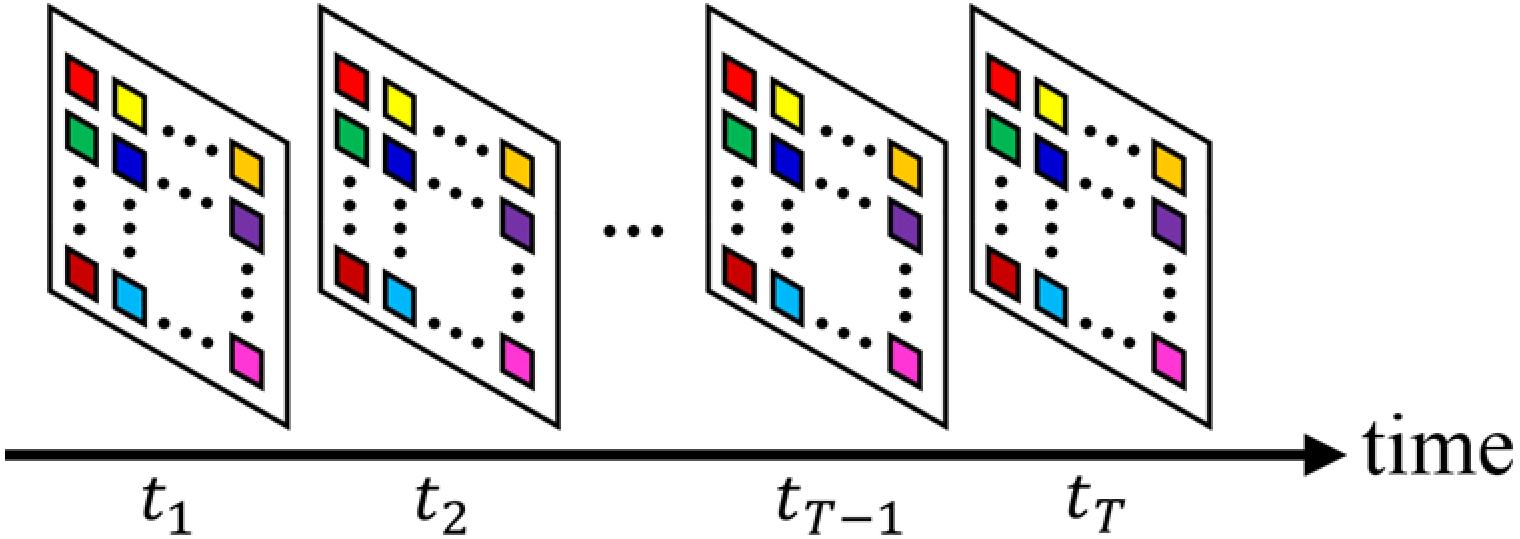


(a) Upper Manhattan (b) 31st December 2016 (c) 22nd May 2017 (d) 3rd January 2018 (e) 17th May 2018

Problem Definition of STKDV

- Given a set of data points $\hat{P} = \{(\mathbf{p}_1, t_{\mathbf{p}_1}), (\mathbf{p}_2, t_{\mathbf{p}_2}), \ldots, (\mathbf{p}_n, t_{\mathbf{p}_n})\}$ with size n, the resolution size $X \times Y$, and T timestamps t_1, t_2,…, t_T, we need to color each pixel $\mathbf{q}$ with the timestamp t_i, where $1 \leq i \leq T$, using the following density spatial-temporal kernel density function.

$$\mathcal{F}_P(\mathbf{q}, t_i) = \frac{1}{n} \sum_{(\mathbf{p}, t_\mathbf{p}) \in \hat{P}} K_{\text{space}}(\mathbf{q}, \mathbf{p}) \cdot K_{\text{time}}(t_i, t_\mathbf{p})$$



Representative Temporal Kernel Functions

- Uniform kernel function

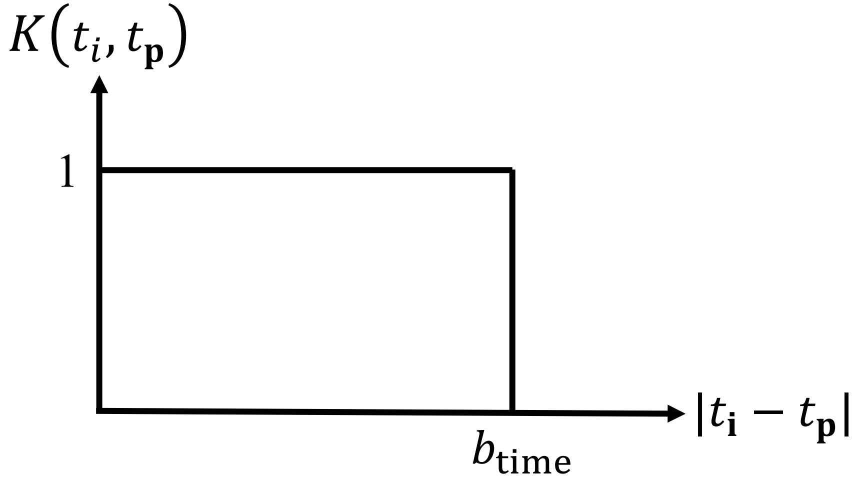
$$K(t_i, t_{\mathbf{p}}) = \begin{cases} \dfrac{1}{b_{\text{time}}} & \text{if } |t_{\mathbf{i}} - t_{\mathbf{p}}| \le b_{\text{time}} \\ 0 & \text{otherwise} \end{cases}$$

- Gaussian kernel function

$$K(t_i, t_{\mathbf{p}}) = \exp\left(-\frac{1}{b_{\text{time}}^2}(t_{\mathbf{i}} - t_{\mathbf{p}})^2\right)$$

- Epanechnikov kernel

$$K(t_i, t_{\mathbf{p}}) = \begin{cases} 1 - \dfrac{1}{b_{\text{time}}^2}(t_{\mathbf{i}} - t_{\mathbf{p}})^2 & \text{if } |t_{\mathbf{i}} - t_{\mathbf{p}}| \le b_{\text{time}} \\ 0 & \text{otherwise} \end{cases}$$



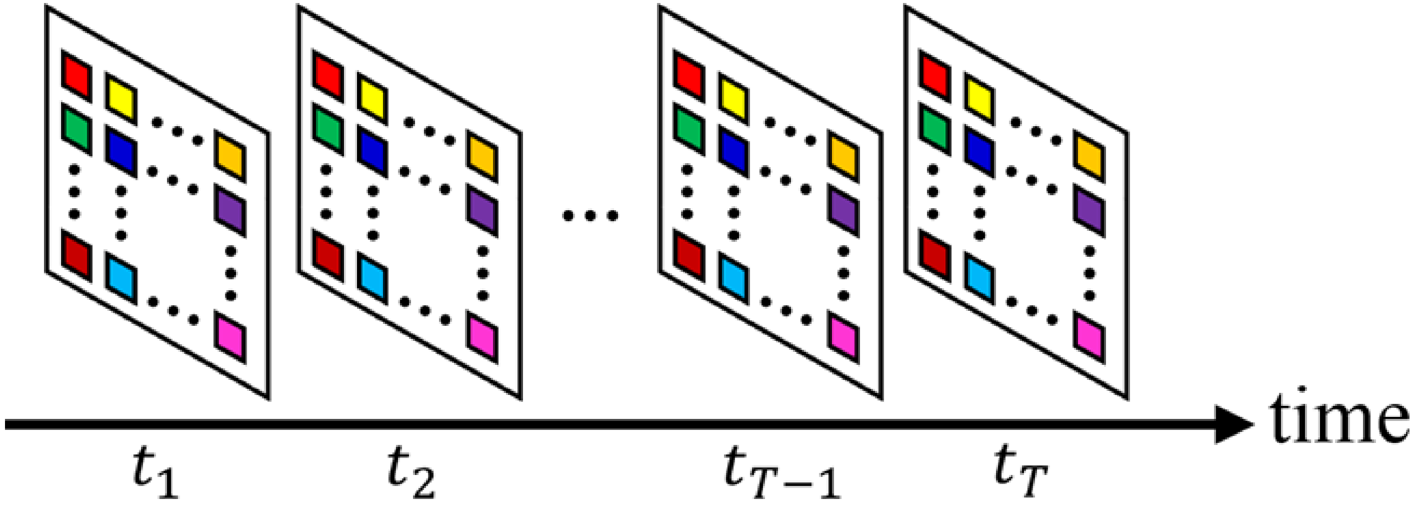
Uniform temporal kernel function

Advantages of STKDV

- Solve the overplotting issues ☺
- Slightly shifting the region does not significantly affect the visualization ☺
- Good visualization quality (Smooth) ☺
- Capture the time information ☺

STKDV is Slow

- Resolution size: $X \times Y$
- Number of data points: n
- Number of timestamps: T
- Time complexity: $O(XYTn)$ ☹
- Slower than SKDV ☹
- Example:
  $X = 512$, $Y = 512$, $n = 2000000$, and $T = 32$
  Time cost = 16.777 **trillion**
  Infeasible to handle this operation



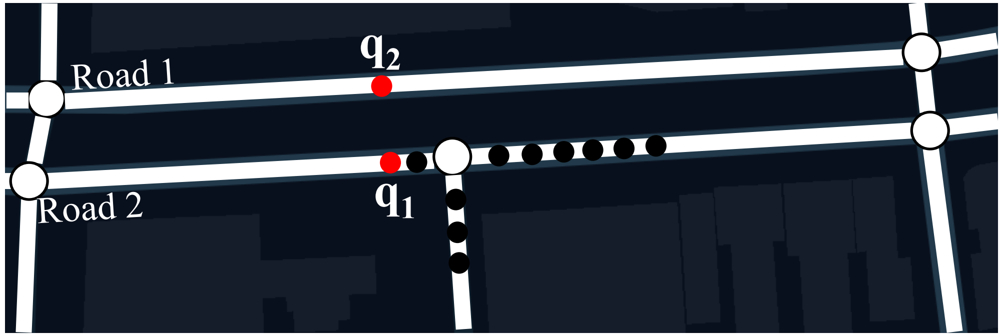
Efficient Algorithms for STKDV

- Parallel solution (link): A parallel approach for generating STKDV.

- SWS (link 1) and PREFIX (link 2): the complexity-optimized solutions for generating STKDV.
  - Theoretically reduce the time complexity.
    - SWS: $O(XY(T+n))$
    - PREFIX: $O(XYT+Yn)$
  - Do not increase the space complexity.
  - Can incorporate the parallel approach to further improve the efficiency (Section 9.5 in this link 1).

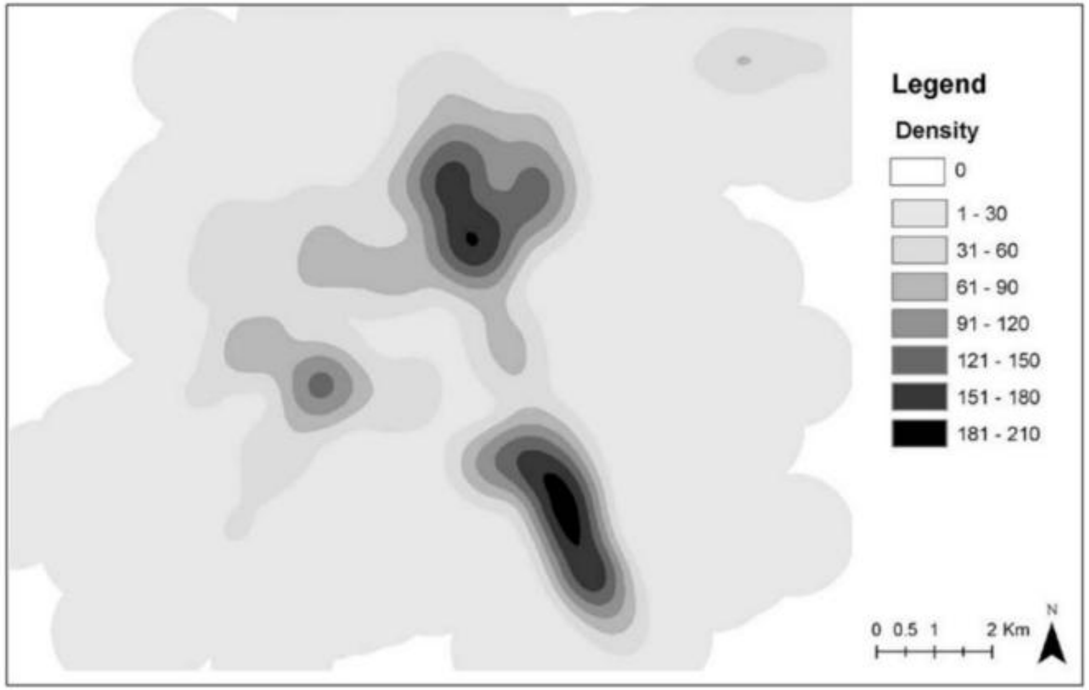
STKDV does not Consider the Road Network Information

- Many data points can be in (or along with) the road network (link).
  - Traffic accidents
  - Crime events



- (Spatial-temporal) Kernel density function can regard the density values of $\mathbf{q}_1$ and $\mathbf{q}_2$ to be similar since they are close in terms of Euclidean distance.

Network Kernel Density Visualization (NKDV)



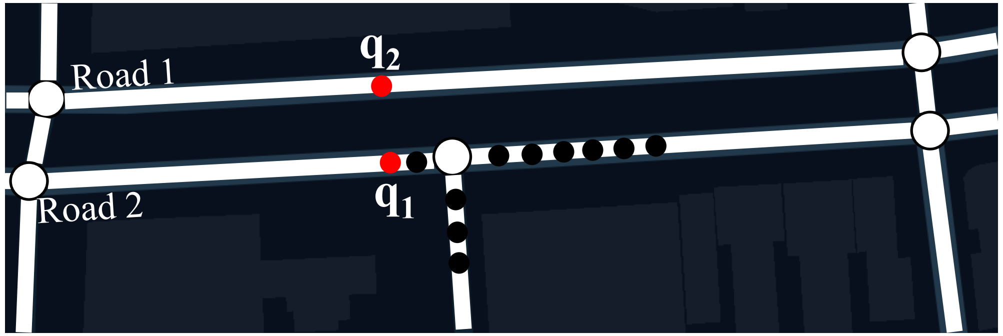
(a) Planar KDV

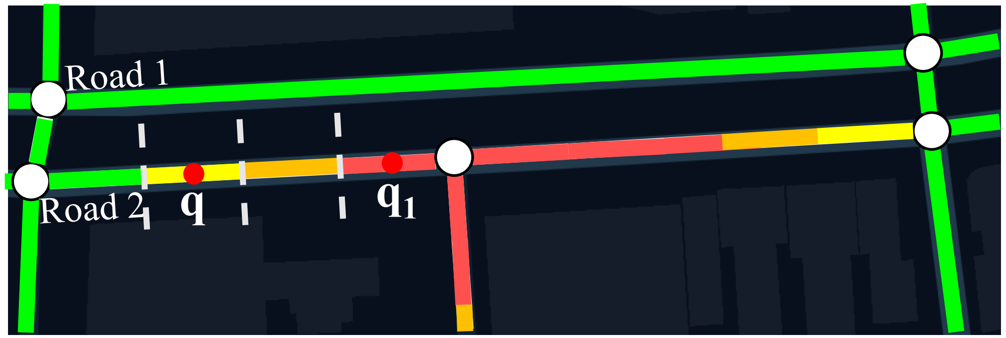
(b) Network KDV

Problem Definition of NKDV

- Given a road network $G = (V, E)$ and a set of two-dimensional data points $P = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n\}$ with size n, the set of lixels, we need to compute the density of each lixel $\mathbf{q}$ using the following network kernel density function.

$$\mathcal{F}_P(\mathbf{q}) = \frac{1}{n} \sum_{\mathbf{p} \in P} K_G(\mathbf{q}, \mathbf{p})$$





- $K_G(\mathbf{q}, \mathbf{p})$ is the kernel function, where we replace the Euclidean distance by the shortest path distance.

Advantages of NKDV

- Solve the overplotting issues ☺

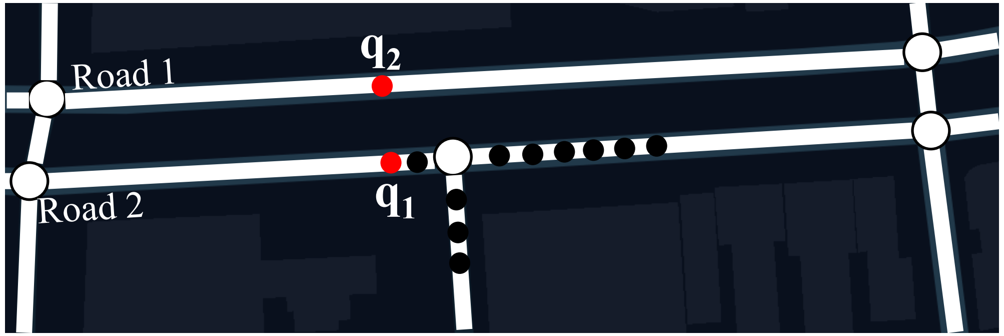
- Slightly shifting the region does not significantly affect the visualization ☺

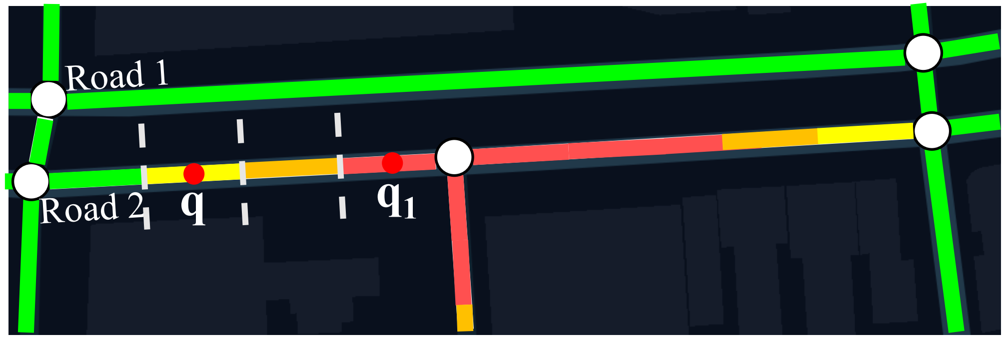
- Good visualization quality (Smooth) ☺

- Capture the road network information.

NKDV is Slow

- Graph: $G = (V, E)$
- Number of lixels: L
- Number of data points: n
- Time complexity: $O(L(|V|\log_2|V| + |E| + n))$
- Example:
  In the New York road network, $|V| = 41467$, $|E| = 116081$, and $n = 1294779$. The time cost is at least 0.2376 trillion.





Efficient Algorithms for NKDV

- Augmentation approach (link 1) and LION (link 2): the complexity-optimized solution for generating NKDV.
  - Theoretically reduce the time complexity.
  - Do not increase the space complexity.

Software Packages

- Python packages
  - LIBKDV (link)
  - PyNKDV (link)

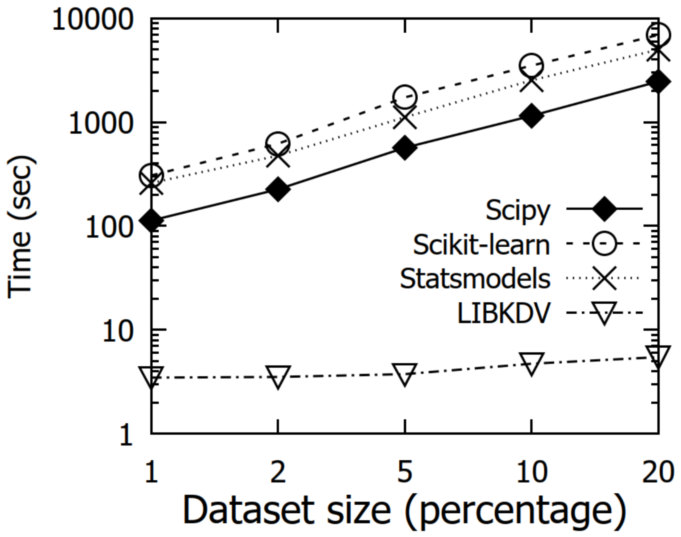
- QGIS plugin
  - Fast Density Analysis (link)

- R package
  - Rlibkdv (link)

- Web-based spatial analysis systems
  - Hong Kong/Macau COVID-19 hotspot map (link 1) (link 2)
  - Spatial-Temporal Analytics with Rapid System (STARS) (link)

LIBKDV

- Efficient python library for supporting both SKDV and STKDV ([link]())
  - Based on SLAM and SWS
  - Has incorporated the parallel implementation for both SLAM and SWS



- Demonstration video of LIBKDV ([link]())

Fast Density Analysis

- An efficient QGIS plugin (link)
  - SKDV: based on SLAM
  - STKDV: based on PREFIX
  - NKDV: based on the augmentation approach

Version		QGIS >=	QGIS <=			Date
1.7	-	3.0.0	3.99.0	3896	bojianzhu	2025年5月30日 GMT+8 11:42
1.6	-	3.0.0	3.99.0	13198	bojianzhu	2023年7月12日 GMT+8 15:55
1.5	-	3.0.0	3.99.0	444	bojianzhu	2023年7月6日 GMT+8 13:14
1.0	-	3.0.0	3.99.0	563	bojianzhu	2023年6月28日 GMT+8 02:24

- Demonstration video of Fast Density Analysis (link 1, link 2)

Spatial-Temporal Analytics with Rapid System (STARS)

- Support KDV, STKDV, and NKDV
- Support exploratory operations in (near) real-time (< 0.5 seconds)
  - Zoom in
  - Zoom out
  - Panning
- Available online (link)

Take Home Messages (For Your Career)

- Foundation (e.g., mathematics, data structures, algorithms, and computational theory) is very important.

- Many applied courses (e.g., web-programming, IoT, and blockchain) may be fun and useful for finding a job. However, only foundational courses can make you competitive.

- Computer science is a fast-changing subject. Most of the knowledge that I learnt five years ago can be outdated. However, foundational courses can never be outdated.