SWS: A Complexity-Optimized Solution for Spatial-Temporal Kernel Density Visualization

Tsz Nam Chan
Hong Kong Baptist University
edisonchan@comp.hkbu.edu.hk

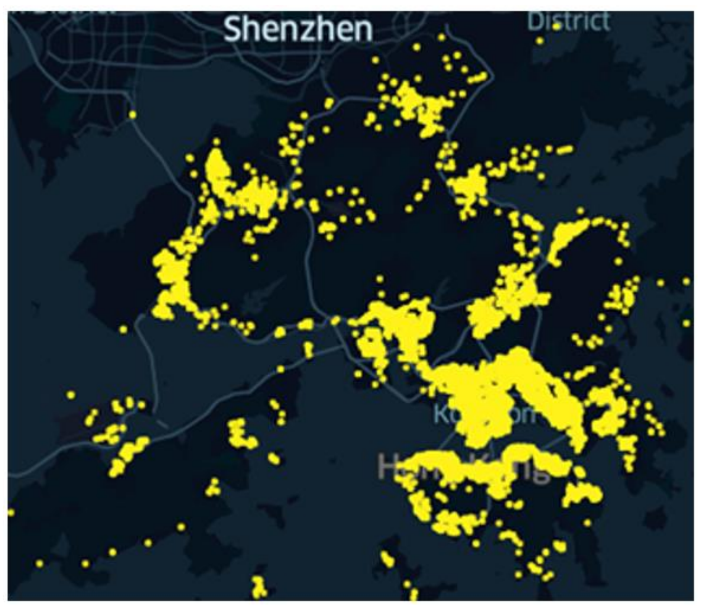
Pak Lon Ip
University of Macau
SKL of Internet of Things for Smart City
paklonip@um.edu.mo

Leong Hou U
University of Macau
SKL of Internet of Things for Smart City
ryanlhu@um.edu.mo

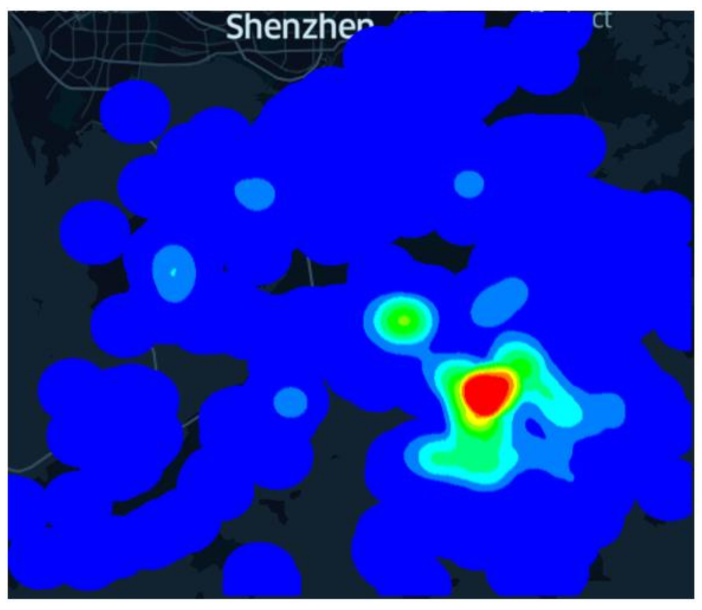
Byron Choi
Hong Kong Baptist University
bchoi@comp.hkbu.edu.hk

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

What is Kernel Density Visualization (KDV)?



Hong Kong COVID-19 cases



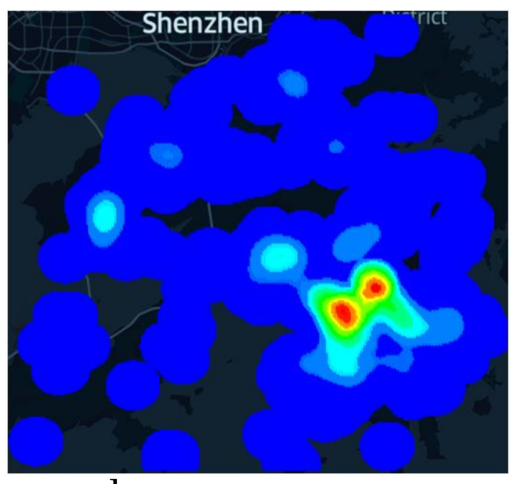
Hotspot map (based on KDV)

- Goal of KDV: Given a location dataset, we need to color each pixel $\mathbf{q}$ based on the kernel density function $\mathcal{F}_P(\mathbf{q})$, where:

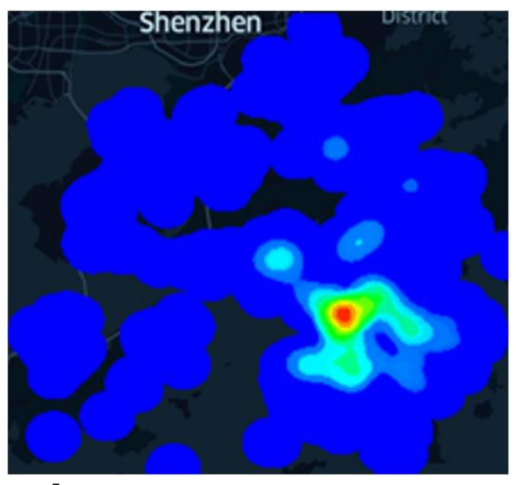
$$\mathcal{F}_P(\mathbf{q}) = \sum_{\mathbf{p}\in P} w \cdot K_{\text{space}}(\mathbf{q},\mathbf{p})$$

- Support many geographical applications:
  - COVID-19 hotspot detection
  - Crime hotspot detection
  - Traffic accident hotspot detection
  - Traffic hotspot detection
- Does not consider the occurrence time of each location data point ☹

Overview of Spatial-Temporal Kernel Density Visualization (STKDV)

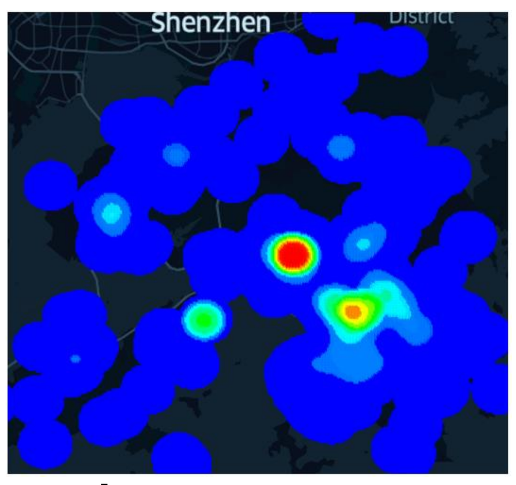


2nd August 2020



6th December 2020

28th February 2021



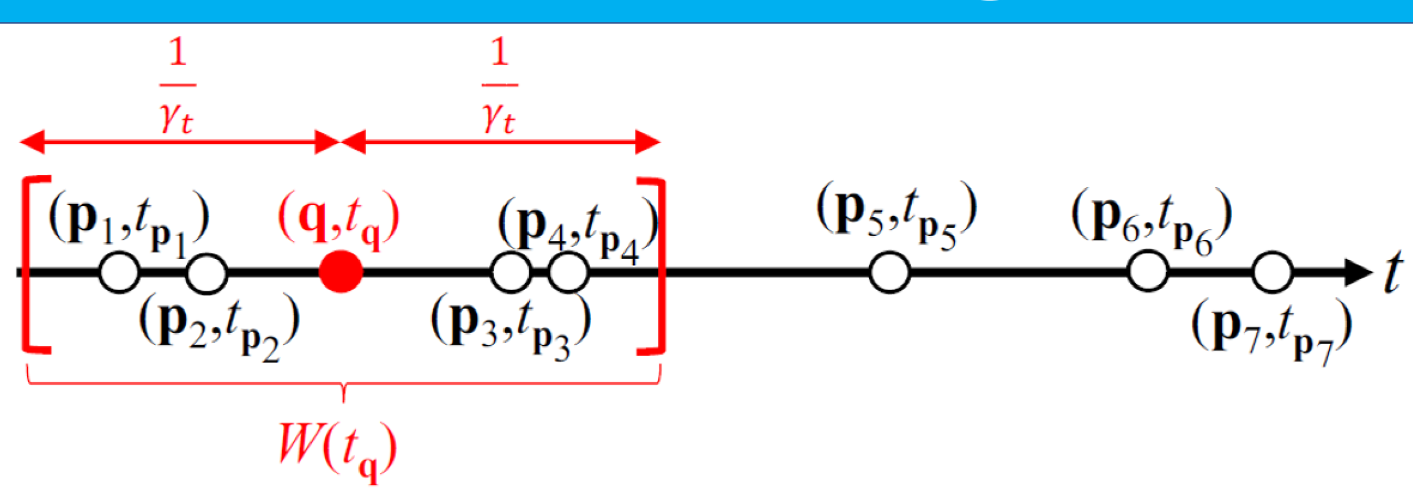
28th January 2022

- Consider a location dataset $\hat{P} = \{(\mathbf{p}_1, t_{\mathbf{p}_1}), (\mathbf{p}_2, t_{\mathbf{p}_2}), \dots, (\mathbf{p}_n, t_{\mathbf{p}_n})\}$ with size n.
- Color each pixel $\mathbf{q}$ with the timestamp $t_{\mathbf{q}}$ based on the spatial-temporal kernel density function $\mathcal{F}_{\hat{P}}(\mathbf{q}, t_{\mathbf{q}})$, where:

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

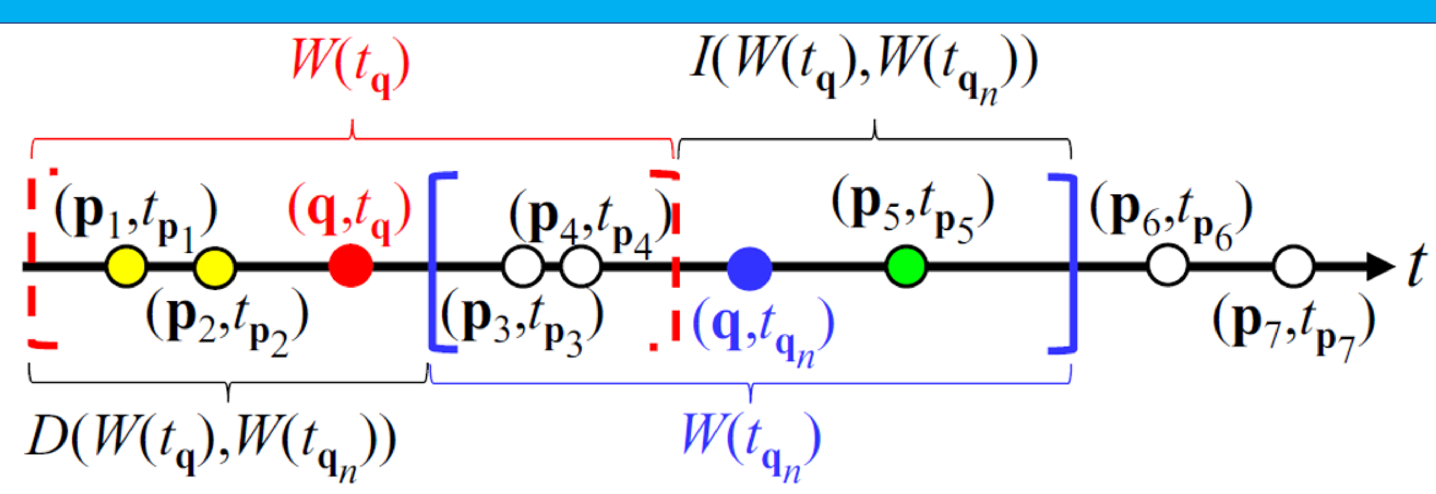
- The time complexity of a naïve solution for generating STKDV is $O(XYTn)$ ☹
- Example:
  - The resolution size ($X \times Y$): 128×128
  - The number of timestamps (T): 128
  - The total number of data points (n): 1,000,000
  - The total cost is: **2.09 trillion operations** ☹

Sliding-Window-based Solution (SWS)



$$\mathcal{F}_{\hat{P}}(\mathbf{q}, t_{\mathbf{q}}) = \sum_{(\mathbf{p},t_{\mathbf{p}})\in W(t_{\mathbf{q}})} w \cdot K_{\text{space}}(\mathbf{q},\mathbf{p}) \cdot \left(1 - \gamma_t^2 dist(t_{\mathbf{q}}, t_{\mathbf{p}})^2\right)$$
$$= w(1-\gamma_t^2 t_{\mathbf{q}}^2) \cdot S^{(0)}_{W(t_{\mathbf{q}})}(\mathbf{q}) + 2w\gamma_t^2 t_{\mathbf{q}} \cdot S^{(1)}_{W(t_{\mathbf{q}})}(\mathbf{q}) - w\gamma_t^2 \cdot S^{(2)}_{W(t_{\mathbf{q}})}(\mathbf{q})$$

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



$$S^{(i)}_{W(t_{\mathbf{q}_n})}(\mathbf{q}) = S^{(i)}_{W(t_{\mathbf{q}})}(\mathbf{q}) - S^{(i)}_{D(W(t_{\mathbf{q}}), W(t_{\mathbf{q}_n}))}(\mathbf{q}) + S^{(i)}_{I(W(t_{\mathbf{q}}), W(t_{\mathbf{q}_n}))}(\mathbf{q})$$

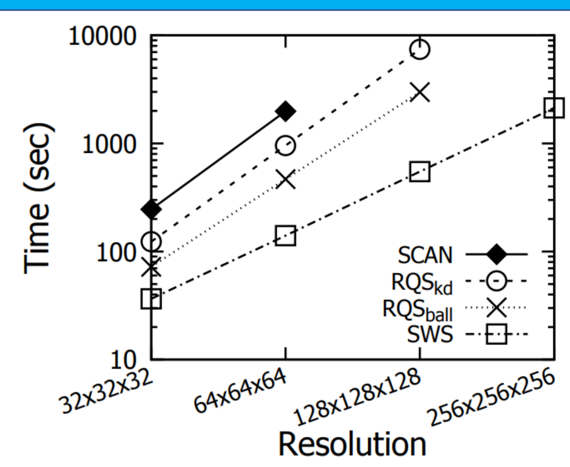
The time complexity for updating the window is $O\left(\left|I\left(W(t_{\mathbf{q}}), W(t_{\mathbf{q}_n})\right)\right| + \left|D\left(W(t_{\mathbf{q}}), W(t_{\mathbf{q}_n})\right)\right|\right)$.

The time complexity for finding the density values of each pixel $\mathbf{q}$ with T timestamps is: $O(T+n)$.

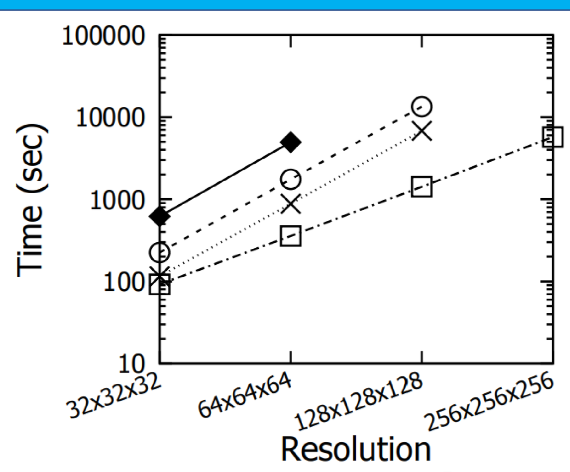
Theoretical Results

Method	Time complexity	Space complexity	Ref.
SCAN	$O(XYTn)$	$O(XYT+n)$	NIL
RQS_{kd}			[23, 42]
RQS_{ball}			[28, 42]
SWS	$O(XY(T+n))$		Sections 3-5

Experimental Results



Ontario



New York