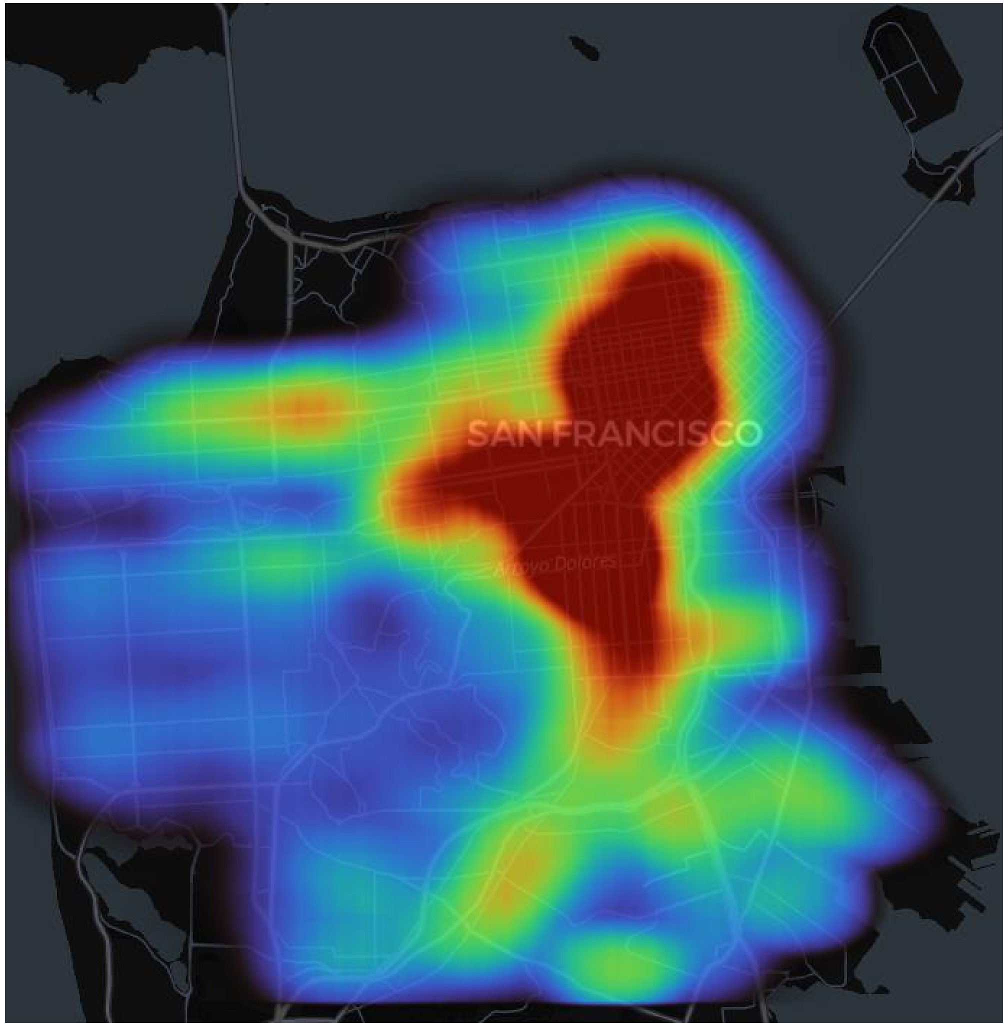
A Fast and Accurate Block Compression Solution for Spatiotemporal Kernel Density Visualization

Yue Zhong[1], Tsz Nam Chan[1], Leong Hou U[2], Dingming Wu[1], Wei Tu[1], Ruisheng Wang[1], Joshua Zhexue Huang[1]

[1]Shenzhen University [2]University of Macau

What is a heatmap?

Heatmap is a common tool to visualize dense and sparse areas in location-based data.



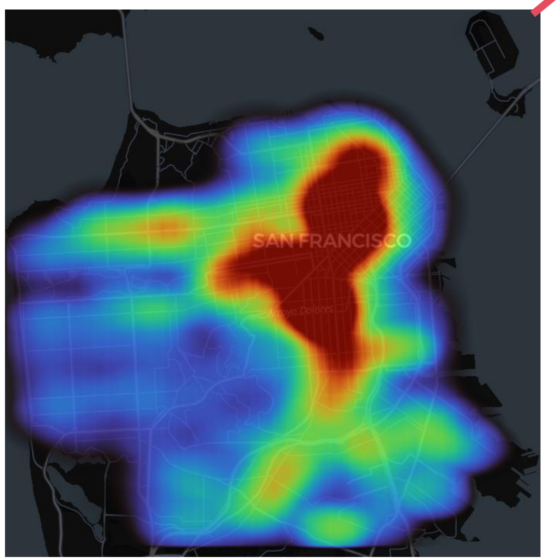
What is a heatmap?

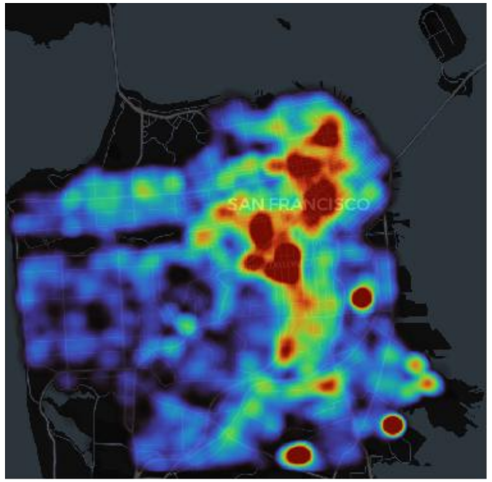
Heatmap is a common tool to visualize dense and sparse areas in location-based data.

$$\mathcal{F}_{\widehat{P}}(\mathbf{q}) = \sum_{(\mathbf{p},\, t_{\mathbf{p}}) \in \widehat{P}} K_{space}(\mathbf{q}, \mathbf{p})$$

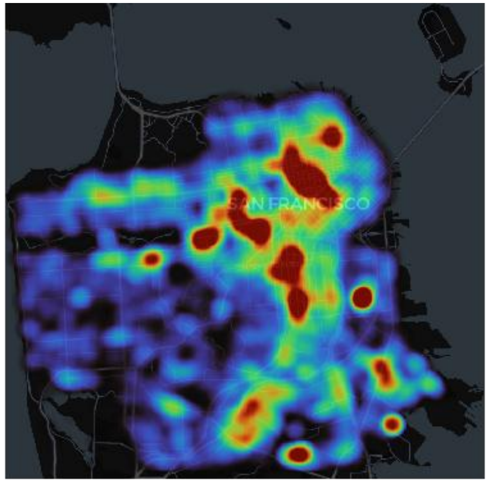
How to generate a heatmap?

Kernel Density Visualization (KDV), a method that estimates density across space.

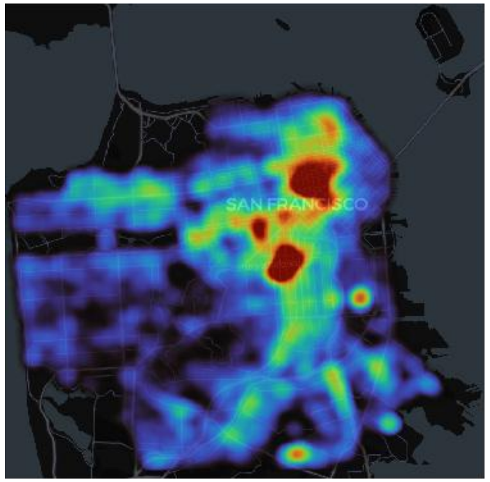




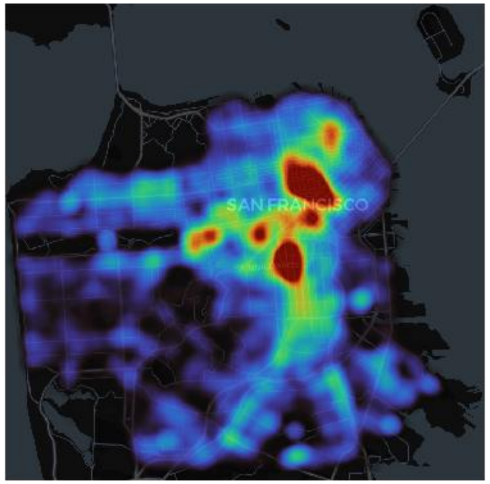
(a) 1st September 2008



(b) 17th January 2010



(c) 7th October 2011



(d) 26th June 2013

Spatiotemporal Kernel Density Visualization (STKDV)

The STKDV at each pixel (q, t_i) is computed as follows:

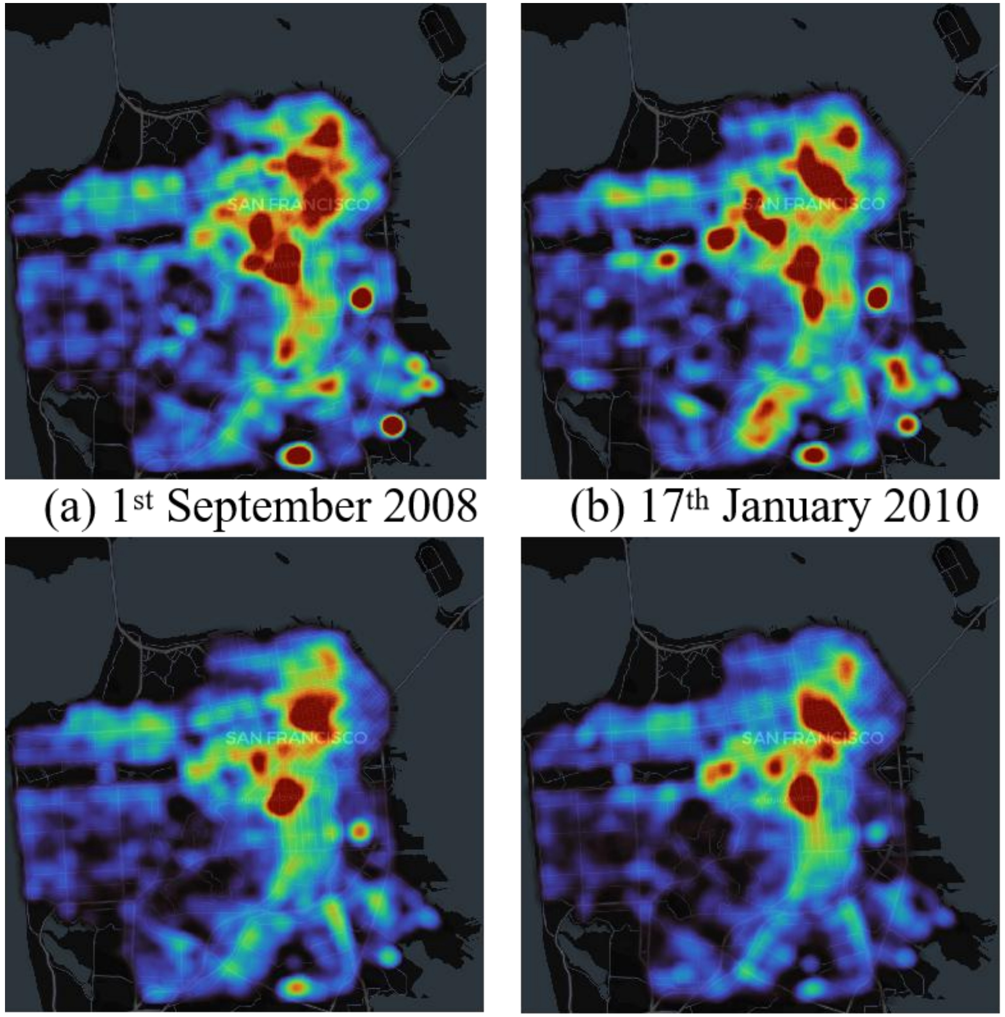
$$\mathcal{F}_P(\mathbf{q}, t_i) = \frac{1}{n} \sum_{(\mathbf{p}, t_\mathbf{p}) \in P} K_{space}(\mathbf{q}, \mathbf{p}) \cdot K_{time}(t_i, t_\mathbf{p})$$

CHALLENGE

Method	Time complexity
SCAN	*O(XYTn)*

Example:

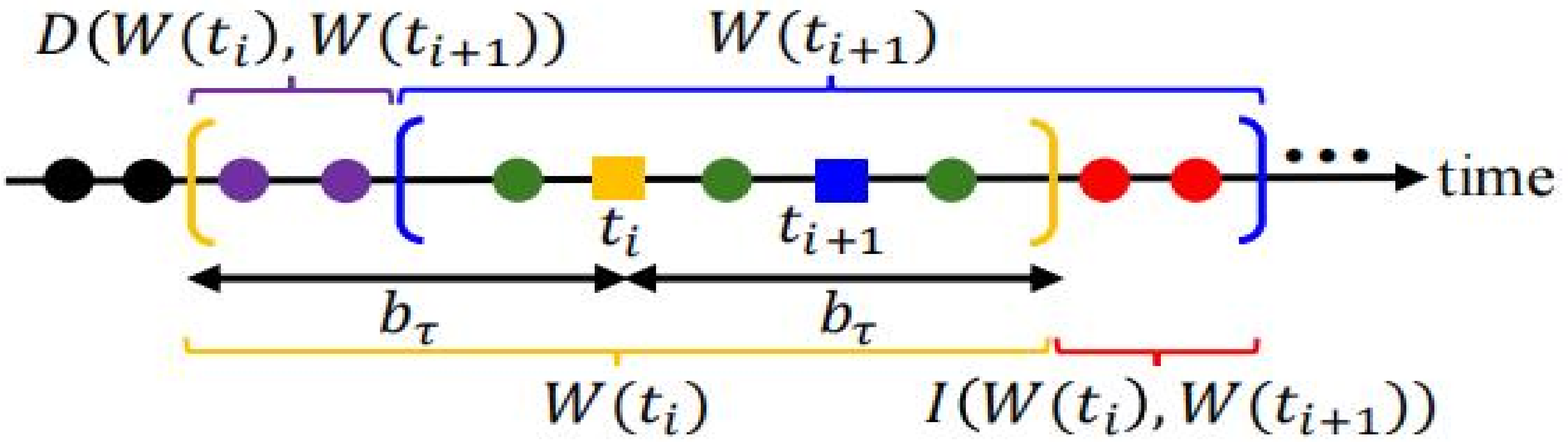
- The resolution size ($X \times Y$): 1280×960
- The number of timestamps (T): 32
- The total number of data points (n): 1.83 million
- The total cost is: **71.96 trillion operations** ☹



(a) 1st September 2008 (b) 17th January 2010

(c) 7th October 2011 (d) 26th June 2013

SWS:



Only those data points $(\mathbf{p},\ t_{\mathbf{p}})$ in $W(t_i)$ can contribute to $\mathcal{F}_{\widehat{P}}(\mathbf{q}, t_i)$.

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

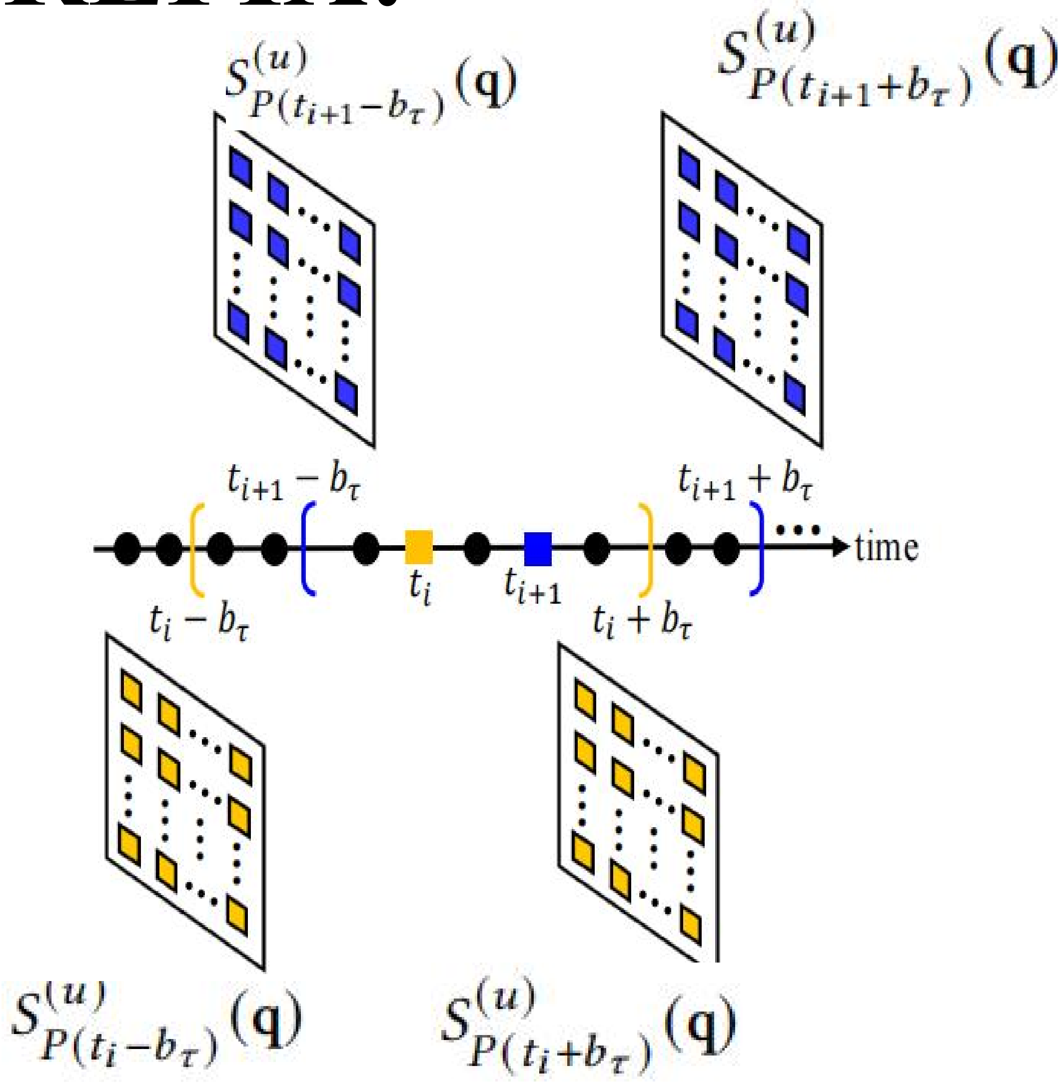
$$= w\left(1 - \frac{t_i^2}{b_\tau^2}\right) S_{W(t_i)}^{(0)}(\mathbf{q}) + \frac{2wt_i}{b_\tau^2} S_{W(t_i)}^{(1)}(\mathbf{q}) - \frac{w}{b_\tau^2} S_{W(t_i)}^{(2)}(\mathbf{q})$$

where $$S_{W(t_i)}^{(u)}(\mathbf{q}) = \sum_{(\mathbf{p}, t_{\mathbf{p}})\in W(t_i)} t_{\mathbf{p}}^u \cdot K_{\text{space}}^{(b_\sigma)}(\mathbf{q}, \mathbf{p})$$

Efficiently update from $S_{W(t_i)}^{(u)}(\mathbf{q})$ to $S_{W(t_{i+1})}^{(u)}(\mathbf{q})$.

The time complexity of SWS is $O(XY(T + n))$.

PREFIX:



First maintain the prefix-matrix with respect to all X×Y pixels for each end point in the time axis.

Once they have all prefix-matrices, they then evaluate each $S^{(u)}_{W(t_i)}(\mathbf{q})$ based on the following equation and compute $\mathcal{F}_{\widehat{P}}(\mathbf{q}, t_i)$ for all pixels $\mathbf{q}$.

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

The time complexity of PREFIX is $O(XYT + Yn)$.

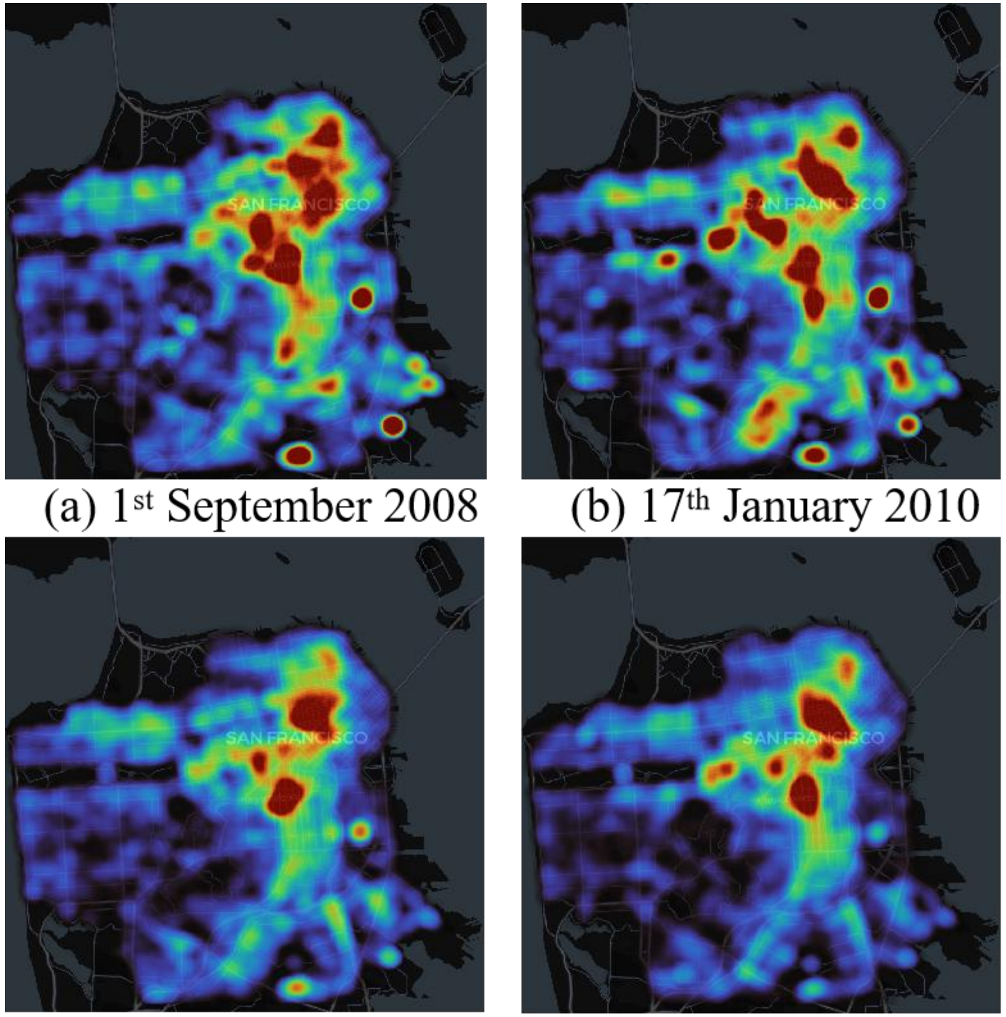
Weakness of Existing Solution : PREFIX

Method	Time complexity
SCAN	$O(XYTn)$
SWS	$O(XY(T+n))$
PREFIX	$O(XYT+Yn)$

In big data scenarios, the data size

n

is very large, resulting in high time complexity for these methods.



(a) 1st September 2008

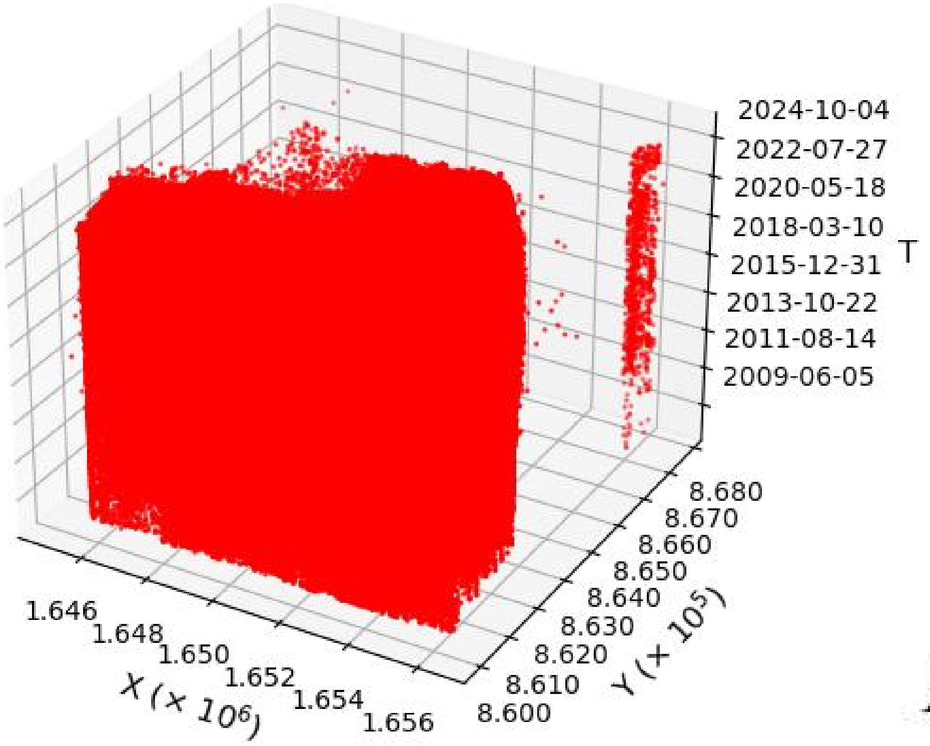
(b) 17th January 2010

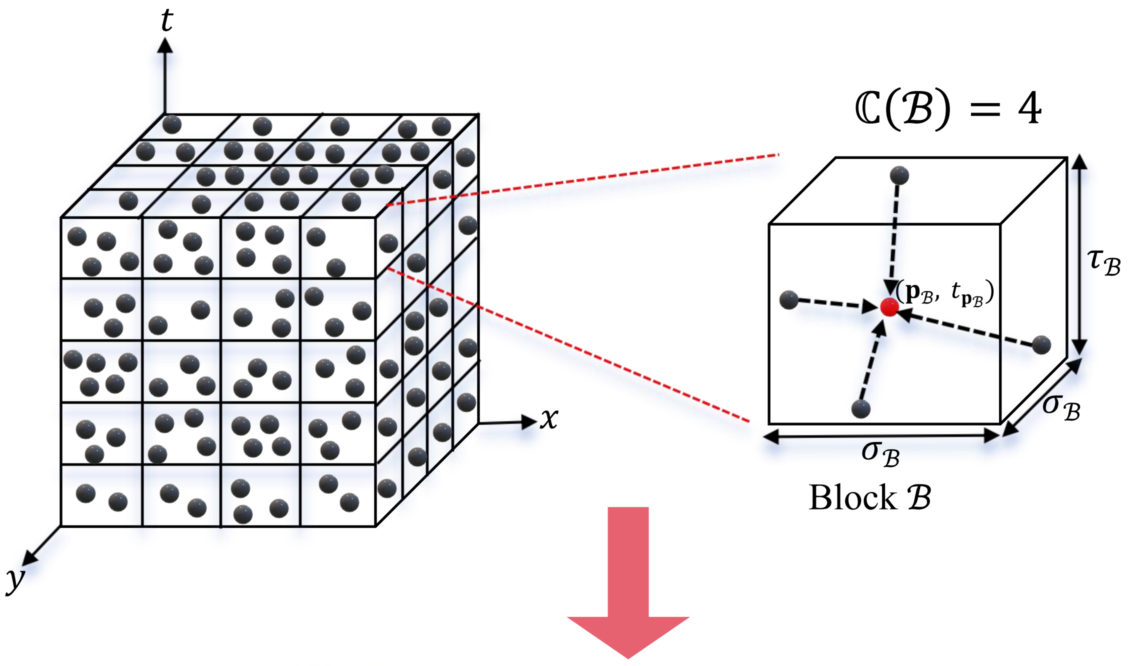
(c) 7th October 2011

(d) 26th June 2013

The Challenge of STKDV: n

Many points in real-world datasets are very close in both space and time.





$$A_{\mathcal{S}}(\mathbf{q}, t_i) = \frac{1}{n} \sum_{\mathcal{B} \in \mathcal{S}} \mathbb{C}(\mathcal{B}) \cdot K_{\text{space}}(\mathbf{q}, \mathbf{p}_{\mathcal{B}}) \cdot K_{\text{time}}(t_i, t_{\mathbf{p}_{\mathcal{B}}})$$

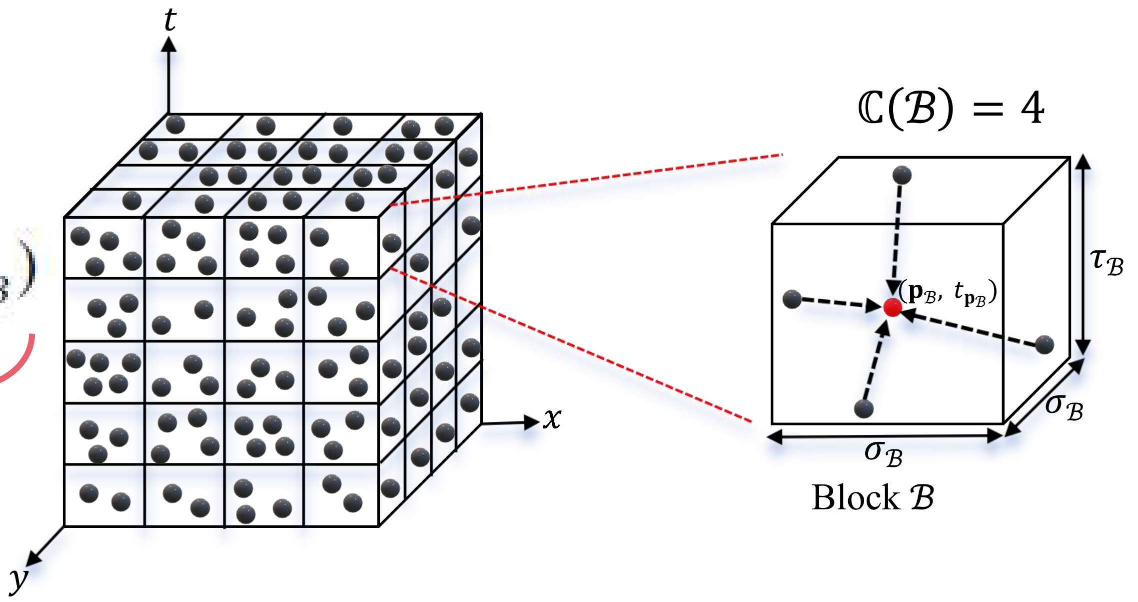
The Challenge of COMP: ϵ

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

$$\underbrace{A_\mathcal{S}(\mathbf{q}, t_i) = \frac{1}{n} \sum_{\mathcal{B} \in \mathcal{S}} \mathbb{C}(\mathcal{B}) \cdot K_{\text{space}}(\mathbf{q}, \mathbf{p}_\mathcal{B}) \cdot K_{\text{time}}(t_i, t_{\mathbf{p}_\mathcal{B}})}$$

$$\boxed{|A_\mathcal{S}(\mathbf{q}, t_i) - \mathscr{F}_P(\mathbf{q}, t_i)| \le \epsilon}$$

$$|K_{\text{space}}(\mathbf{q}, \mathbf{p}_\mathcal{B}) \cdot K_{\text{time}}(t_i, t_{\mathbf{p}_\mathcal{B}}) - K_{\text{space}}(\mathbf{q}, \mathbf{p}) \cdot K_{\text{time}}(t_i, t_\mathbf{p})| \le \epsilon$$



The Challenge of COMP: ϵ

$$|K_{\text{space}}(\mathbf{q}, \mathbf{p}_{\mathcal{B}}) \cdot K_{\text{time}}(t_i, t_{\mathbf{p}_{\mathcal{B}}}) - K_{\text{space}}(\mathbf{q}, \mathbf{p}) \cdot K_{\text{time}}(t_i, t_{\mathbf{p}})| \leq \epsilon$$

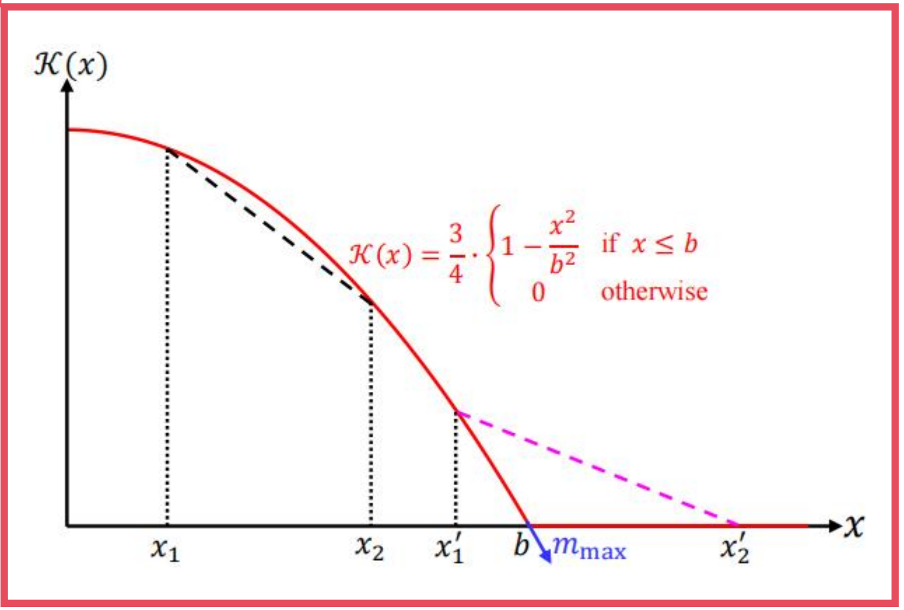
$$\left|\frac{\mathcal{K}(x_1) - \mathcal{K}(x_2)}{x_1 - x_2}\right| \leq m_{\max}$$

$$m_{\max} = \max_x \left(\left|\frac{d\mathcal{K}(x)}{dx}\right|\right)$$

$$|K_{\text{space}}(\mathbf{q}, \mathbf{p}_{\mathcal{B}}) - K_{\text{space}}(\mathbf{q}, \mathbf{p})| \leq \frac{\sqrt{2} \cdot \omega_{\mathcal{B}} \cdot m_{\max}^{(\sigma)}}{2}$$

$$|K_{\text{time}}(t_i, t_{\mathbf{p}_{\mathcal{B}}}) - K_{\text{time}}(t_i, t_{\mathbf{p}})| \leq \frac{\lambda_{\mathcal{B}} \cdot m_{\max}^{(\tau)}}{2}$$

$$\begin{aligned}
& |K_{\text{space}}(\mathbf{q}, \mathbf{p}_{\mathcal{B}}) \cdot K_{\text{time}}(t_i, t_{\mathbf{p}_{\mathcal{B}}}) - K_{\text{space}}(\mathbf{q}, \mathbf{p}) \cdot K_{\text{time}}(t_i, t_{\mathbf{p}})| \\
= & |K_{\text{space}}(\mathbf{q}, \mathbf{p}_{\mathcal{B}}) \cdot K_{\text{time}}(t_i, t_{\mathbf{p}_{\mathcal{B}}}) - K_{\text{space}}(\mathbf{q}, \mathbf{p}_{\mathcal{B}}) \cdot K_{\text{time}}(t_i, t_{\mathbf{p}}) \\
+ & K_{\text{space}}(\mathbf{q}, \mathbf{p}_{\mathcal{B}}) \cdot K_{\text{time}}(t_i, t_{\mathbf{p}}) - K_{\text{space}}(\mathbf{q}, \mathbf{p}) \cdot K_{\text{time}}(t_i, t_{\mathbf{p}})| \\
\leq & |K_{\text{space}}(\mathbf{q}, \mathbf{p}_{\mathcal{B}}) \cdot K_{\text{time}}(t_i, t_{\mathbf{p}_{\mathcal{B}}}) - K_{\text{space}}(\mathbf{q}, \mathbf{p}_{\mathcal{B}}) \cdot K_{\text{time}}(t_i, t_{\mathbf{p}})| \\
+ & |K_{\text{space}}(\mathbf{q}, \mathbf{p}_{\mathcal{B}}) \cdot K_{\text{time}}(t_i, t_{\mathbf{p}}) - K_{\text{space}}(\mathbf{q}, \mathbf{p}) \cdot K_{\text{time}}(t_i, t_{\mathbf{p}})| \\
\leq & \frac{\lambda_{\mathcal{B}} \cdot m_{\max}^{(\tau)}}{2} K_{\text{space}}(\mathbf{q}, \mathbf{p}_{\mathcal{B}}) + \frac{\sqrt{2} \cdot \omega_{\mathcal{B}} \cdot m_{\max}^{(\sigma)}}{2} K_{\text{time}}(t_i, t_{\mathbf{p}})
\end{aligned}$$



The Challenge of COMP: ϵ

Theorem 1. *Given a location dataset P and an absolute error ϵ, we can achieve the absolute error guarantee ϵ in Definition 2 if $\omega_{\mathcal{B}}$ and $\lambda_{\mathcal{B}}$ of each block $\mathcal{B}$ in $\mathcal{S}$ have the following settings.*

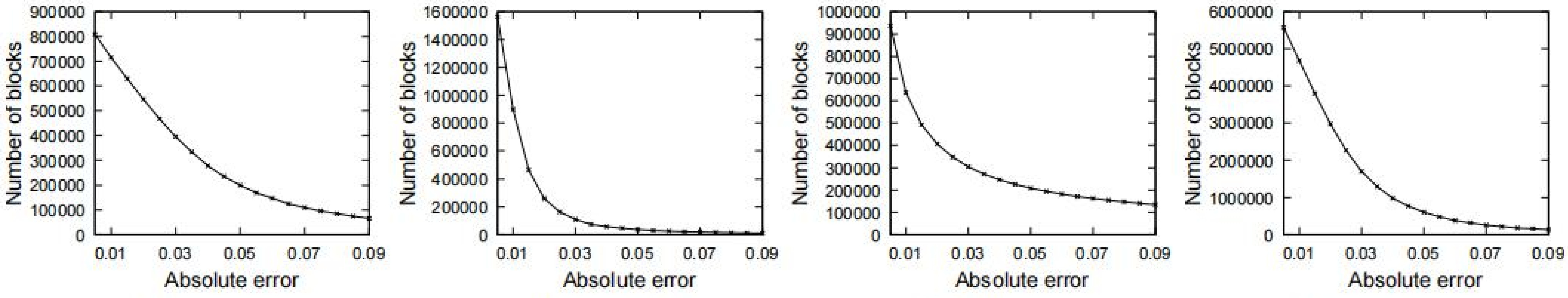
(1) $\omega_{\mathcal{B}} = \frac{\sqrt{2}\cdot\epsilon\cdot b_\sigma}{2}$ *and* $\lambda_{\mathcal{B}} = \epsilon \cdot b_\tau$ *using the triangular spatial and temporal kernels.*

(2) $\omega_{\mathcal{B}} = \frac{4\sqrt{2}\cdot\epsilon\cdot b_\sigma}{9}$ *and* $\lambda_{\mathcal{B}} = \frac{8\cdot\epsilon\cdot b_\tau}{9}$ *using the Epanechnikov spatial and temporal kernels.*

(3) $\omega_{\mathcal{B}} = \frac{16\sqrt{6}\cdot\epsilon\cdot b_\sigma}{75}$ *and* $\lambda_{\mathcal{B}} = \frac{32\sqrt{3}\cdot\epsilon\cdot b_\tau}{75}$ *using the quartic spatial and temporal kernels.*

Algorithm 1 Block Compression Algorithm for Generating Approximate STKDV

procedure COMP($P = \{(\mathbf{p}_1, t_{\mathbf{p}_1}), (\mathbf{p}_2, t_{\mathbf{p}_2}), ..., (\mathbf{p}_n, t_{\mathbf{p}_n})\}$, spatial bandwidth b_σ, temporal bandwidth b_τ, error parameter ϵ)
    Compute $\omega_{\mathcal{B}}$ and $\lambda_{\mathcal{B}}$ ▷ Theorem 1
    $\mathcal{S} \leftarrow \phi$
    for each $(\mathbf{p}, t_\mathbf{p}) \in P$ **do**
        Identify the correct block $\mathcal{B}$ for $(\mathbf{p}, t_\mathbf{p})$
        if $\mathcal{B} \in \mathcal{S}$ **then**
            $\mathbb{C}(\mathcal{B}) \leftarrow \mathbb{C}(\mathcal{B}) + 1$
        else
            $\mathbb{C}(\mathcal{B}) \leftarrow 1$
            $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{B}$
    Generate STKDV with $A_\mathcal{S}(\mathbf{q}, t_i)$ (based on $\mathcal{S}$)



(a) Montgomery (b) New York (c) Chicago (d) San Francisco

Combine COMP with SWS and PREFIX

COMP: does not modify the core STKDV algorithm (only reduces data size).

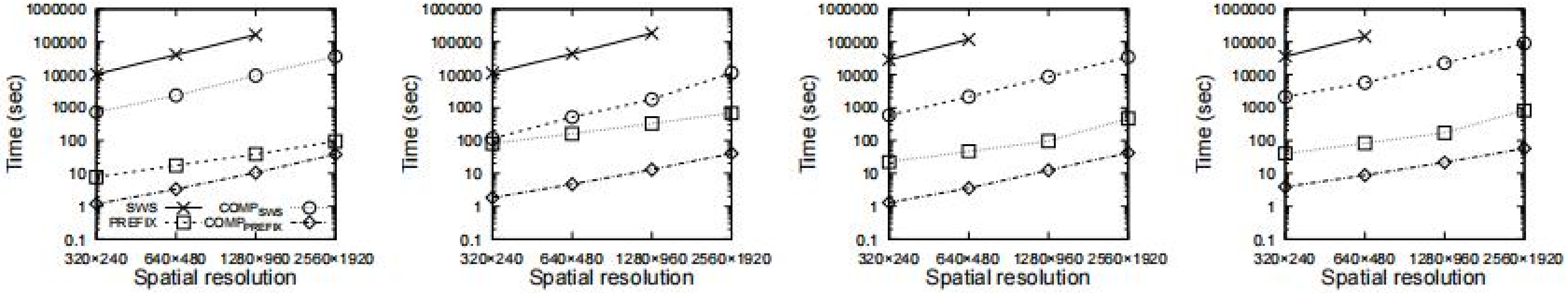
COMP_{SWS} $\text{COMP}_{\text{PREFIX}}$

Exact:

Approximation:

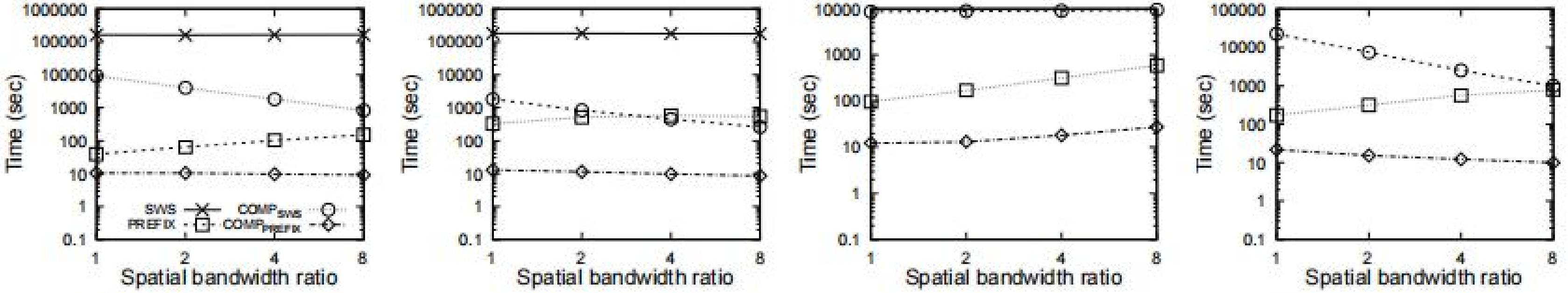
(a) 10th January 2024 (b) 5th April 2024 (c) 9th July 2024 (d) 22nd October 2024

Combine COMP with SWS and PREFIX



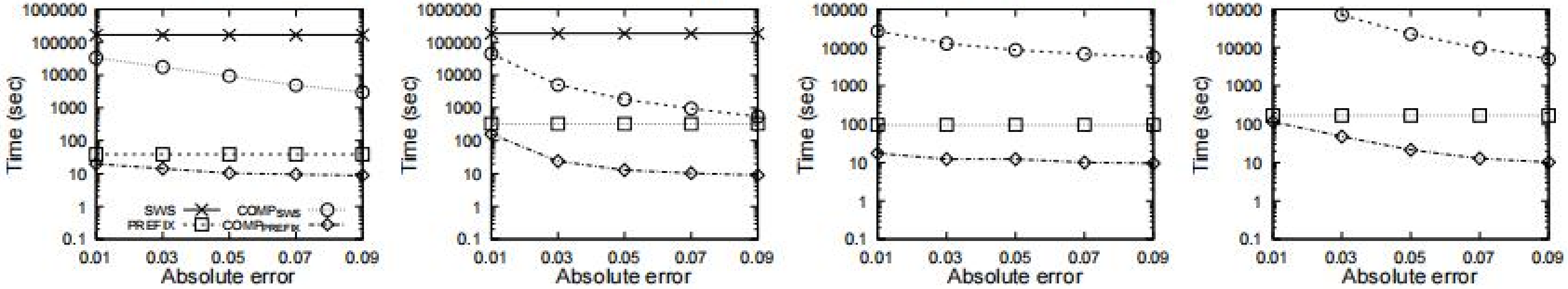
(a) Montgomery (b) New York (c) Chicago (d) San Francisco

Figure 9: Response time for computing STKDV, varying the spatial resolution.



(a) Montgomery (b) New York (c) Chicago (d) San Francisco

Figure 10: Response time for computing STKDV, varying the spatial bandwidth.



(a) Montgomery (b) New York (c) Chicago (d) San Francisco

Figure 11: Response time for computing STKDV, varying the absolute error.

By combining COMP with the existing methods, our experimental results verify that $COMP_{SWS}$ and $COMP_{PREFIX}$ can achieve speedups of 4.1x to 677.16x and 1.45x to 143.52x compared with SWS and PREFIX, respectively, without degrading the visualization results.

Future Work

- *Investigate the extension of block compression to additional geospatial analysis methods, including the K-function and inverse distance weighted interpolation.*

- *Develop Python and R packages based on the COMP method to support efficient spatiotemporal kernel density visualization.*

- *Develop a QGIS/ArcGIS plugin based on the COMP method to facilitate practical applications in geospatial analysis.*

Thanks.