

# Advanced Data Management (CSCI 490/680)

---

Spatial Data

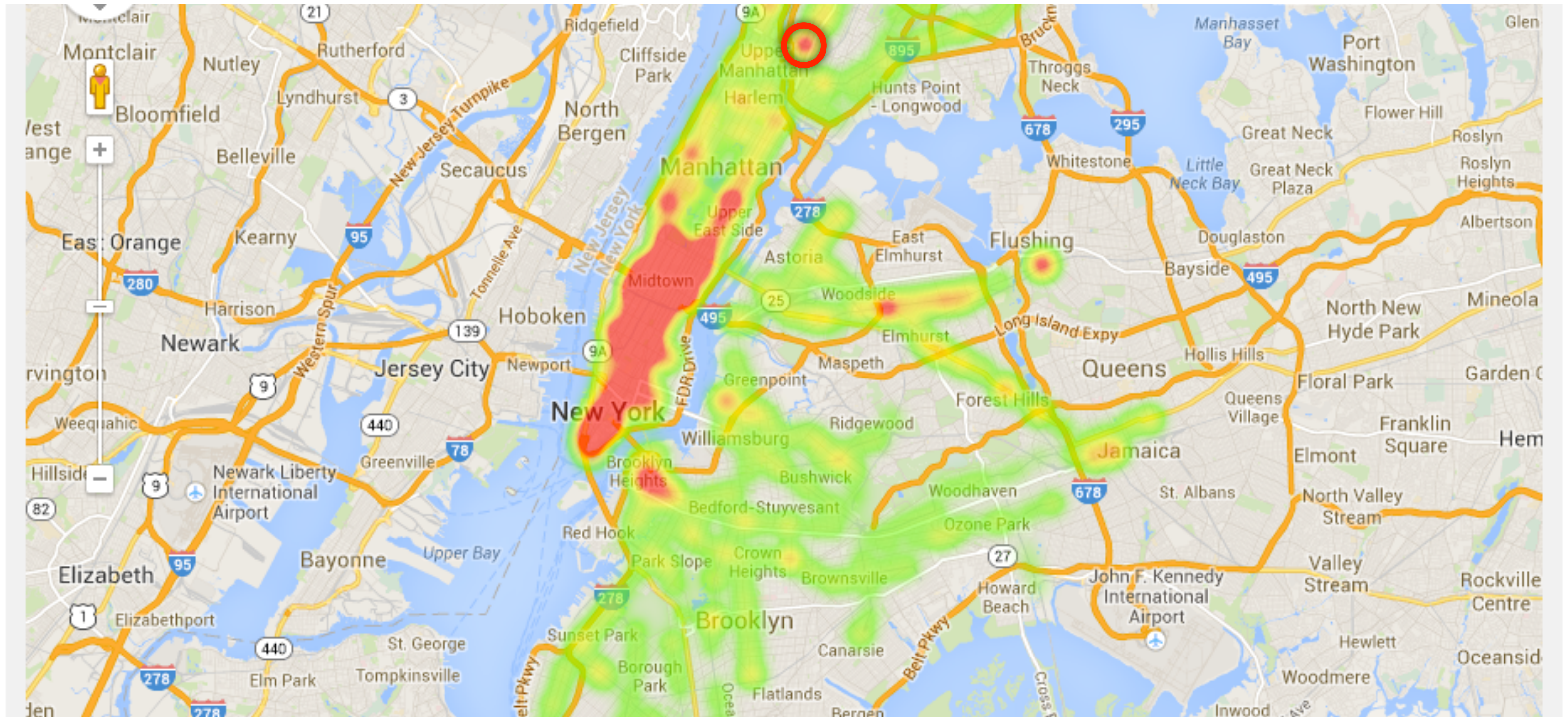
Dr. David Koop





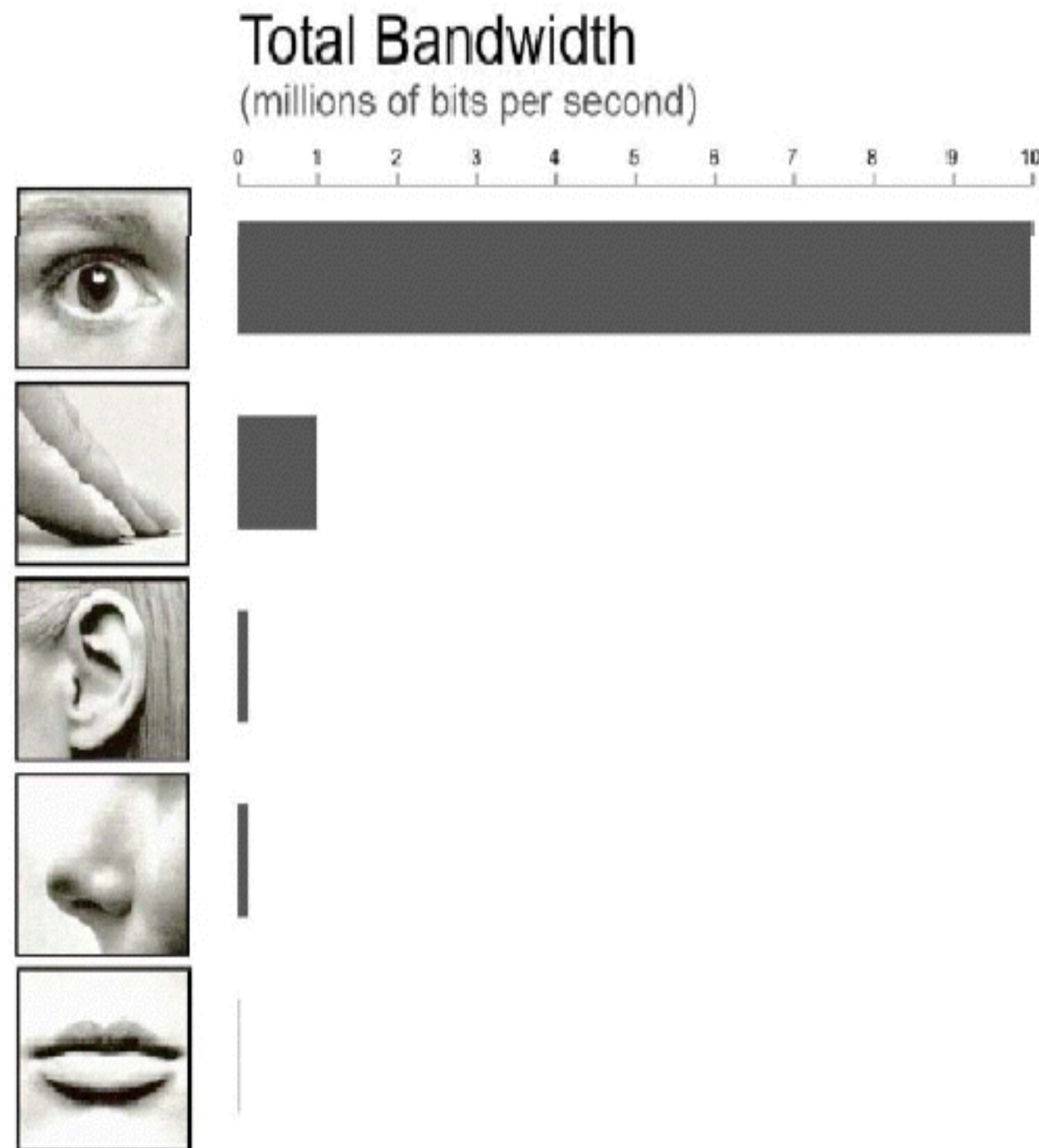


# Data Exploration Through Visualization

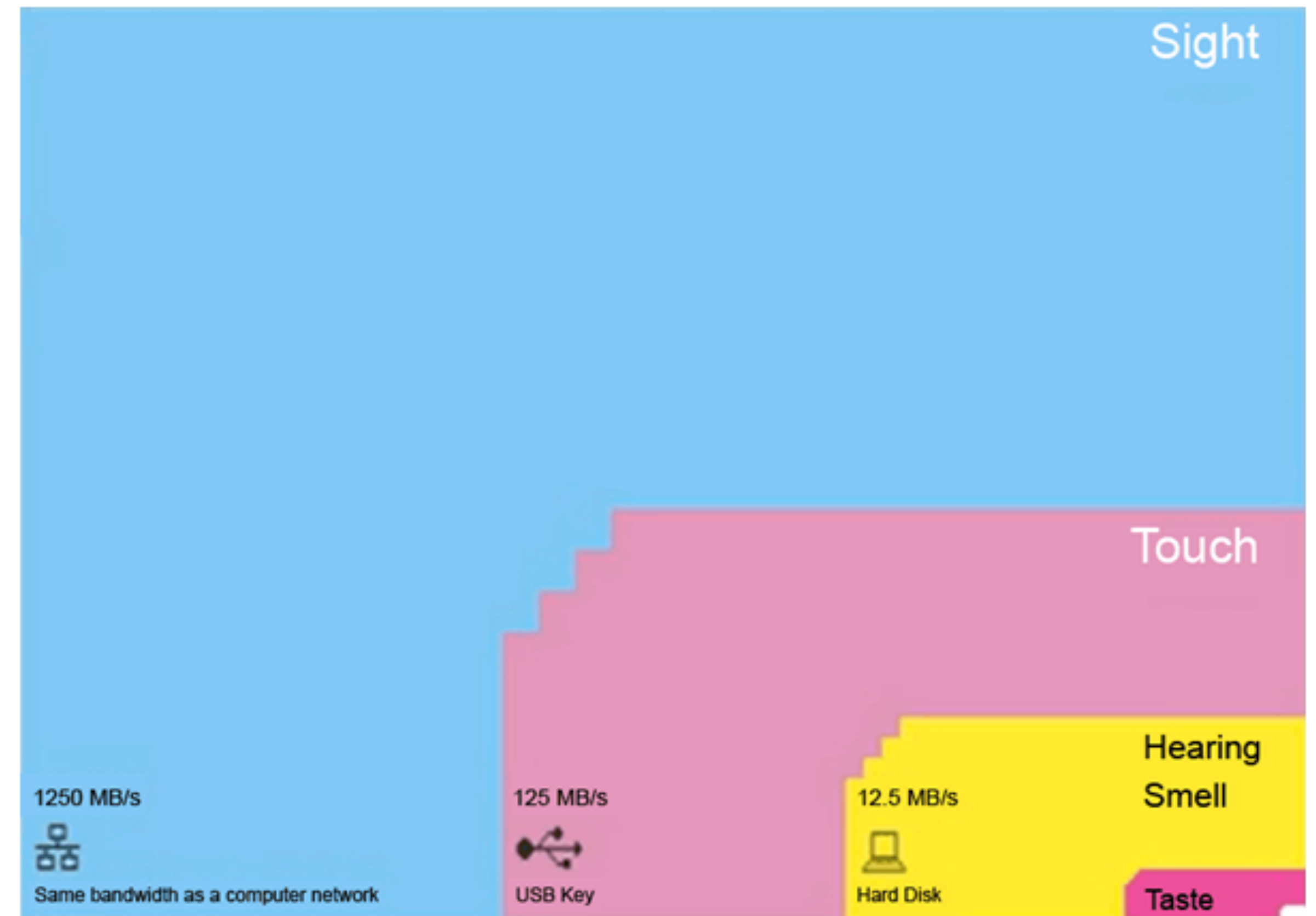




# Why do we visualize data?



[via A. Lex]



[T. Nørretranders]



# Why Visual?

I		II		III		IV	
x	y	x	y	x	y	x	y
10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71
9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47
14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04
6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25
4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.50
12.0	10.84	12.0	9.13	12.0	8.15	8.0	5.56
7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91
5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89

[F. J. Anscombe]



# Why Visual?

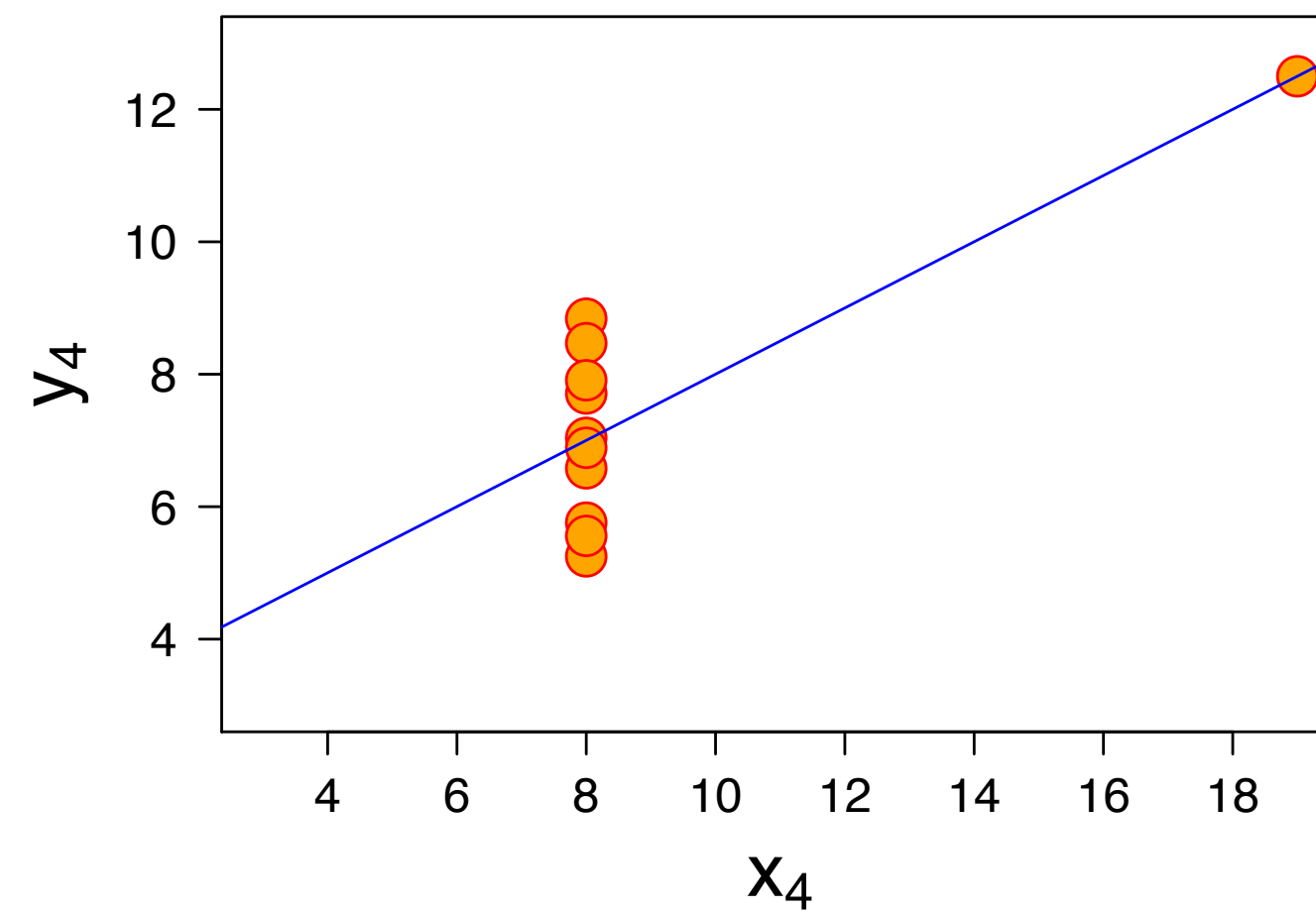
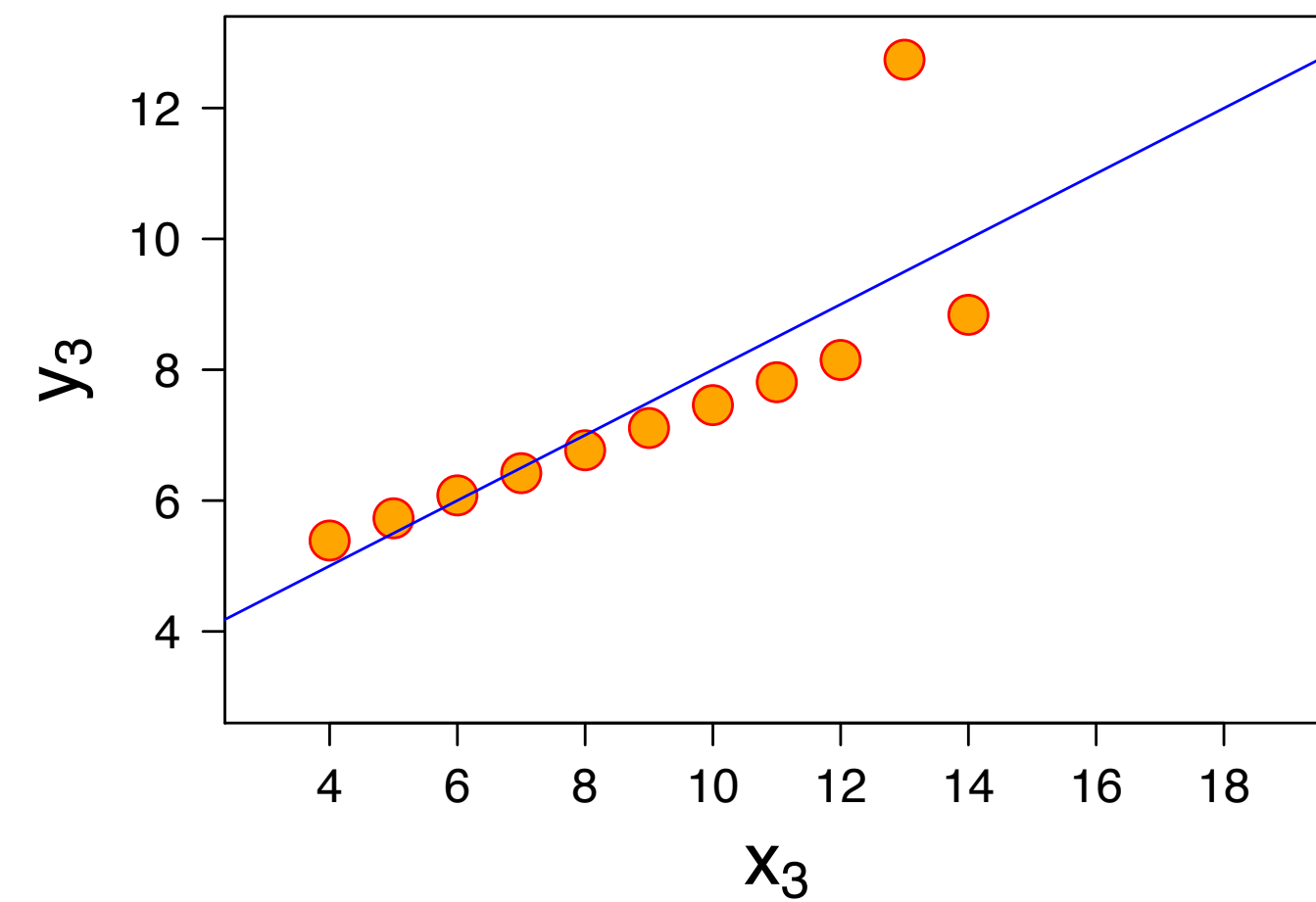
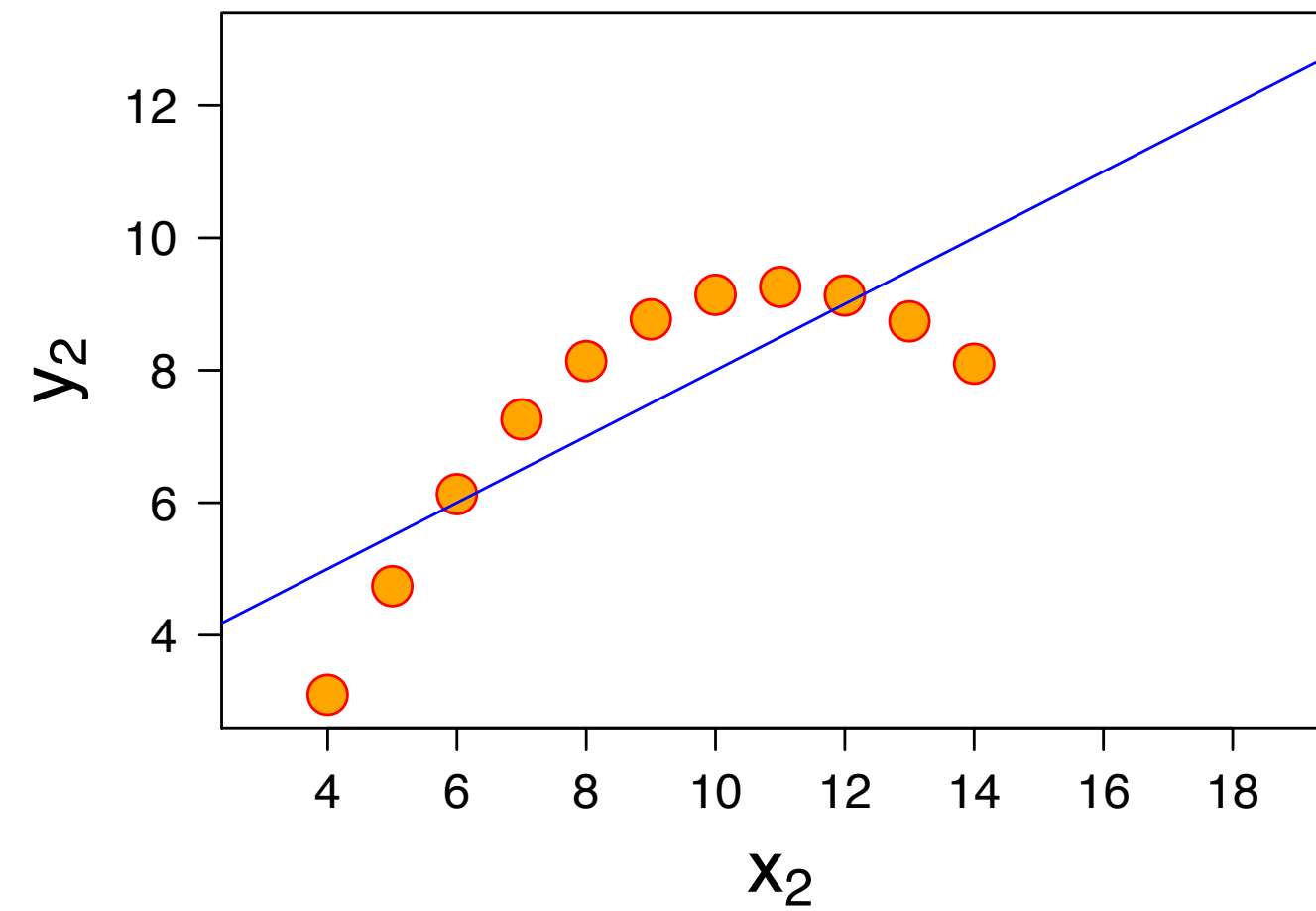
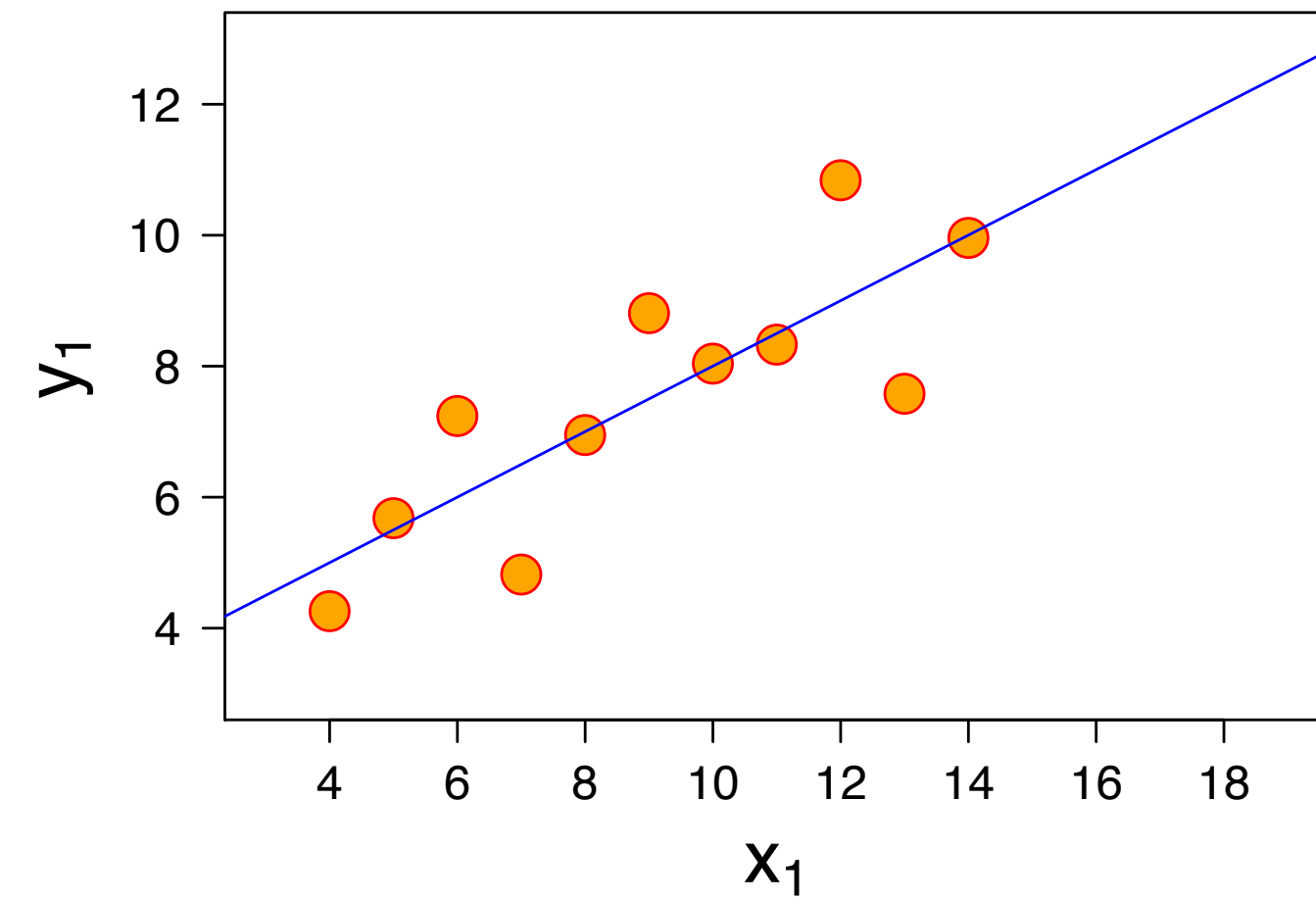
I		II		III		IV	
x	y	x	y	x	y	x	y
10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71
9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47
14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04
6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25
4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.50
12.0	10.84	12.0	9.13	12.0	8.15	8.0	5.56
7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91
5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89

Mean of x	9
Variance of x	11
Mean of y	7.50
Variance of y	4.122
Correlation	0.816

[F. J. Anscombe]



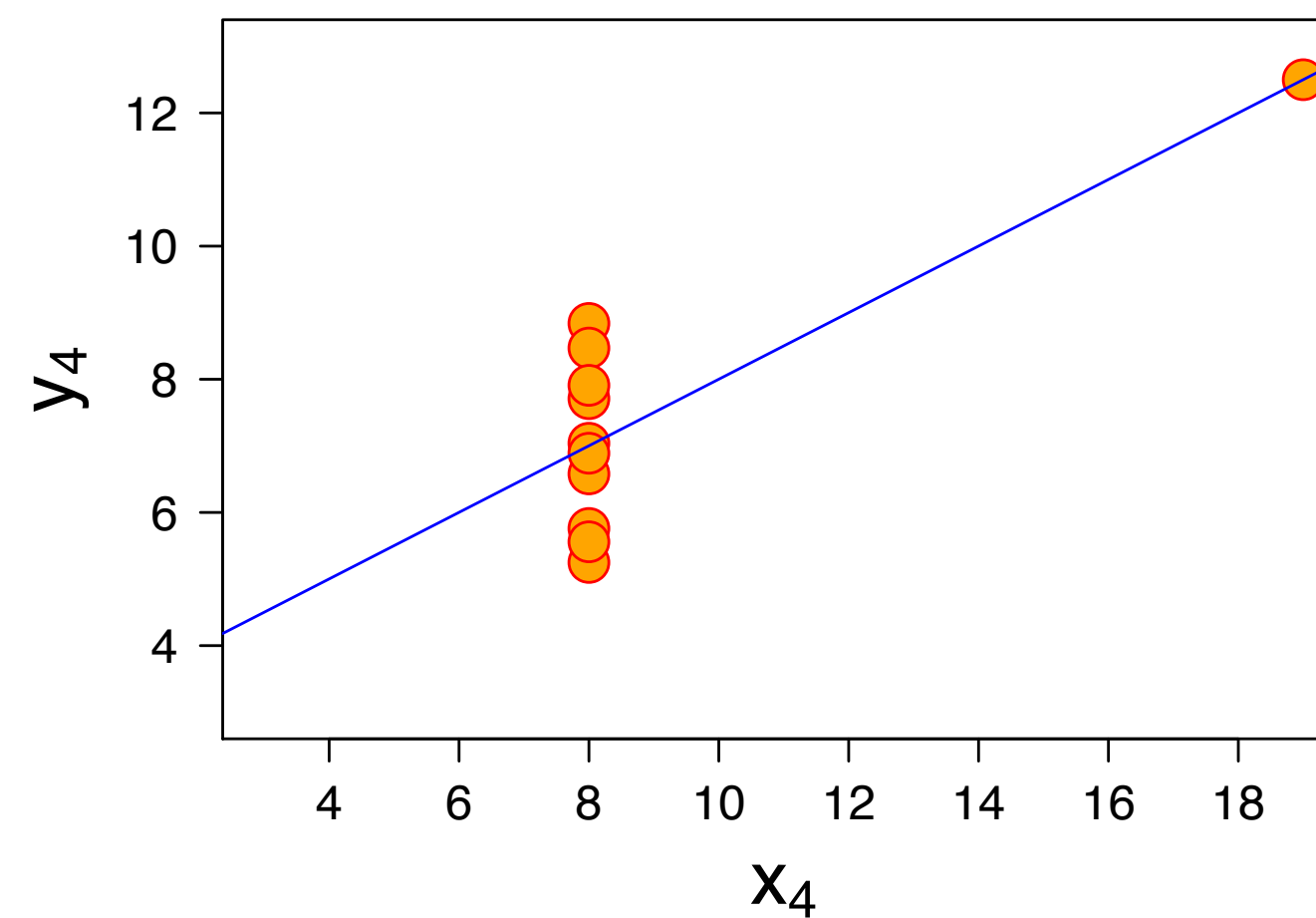
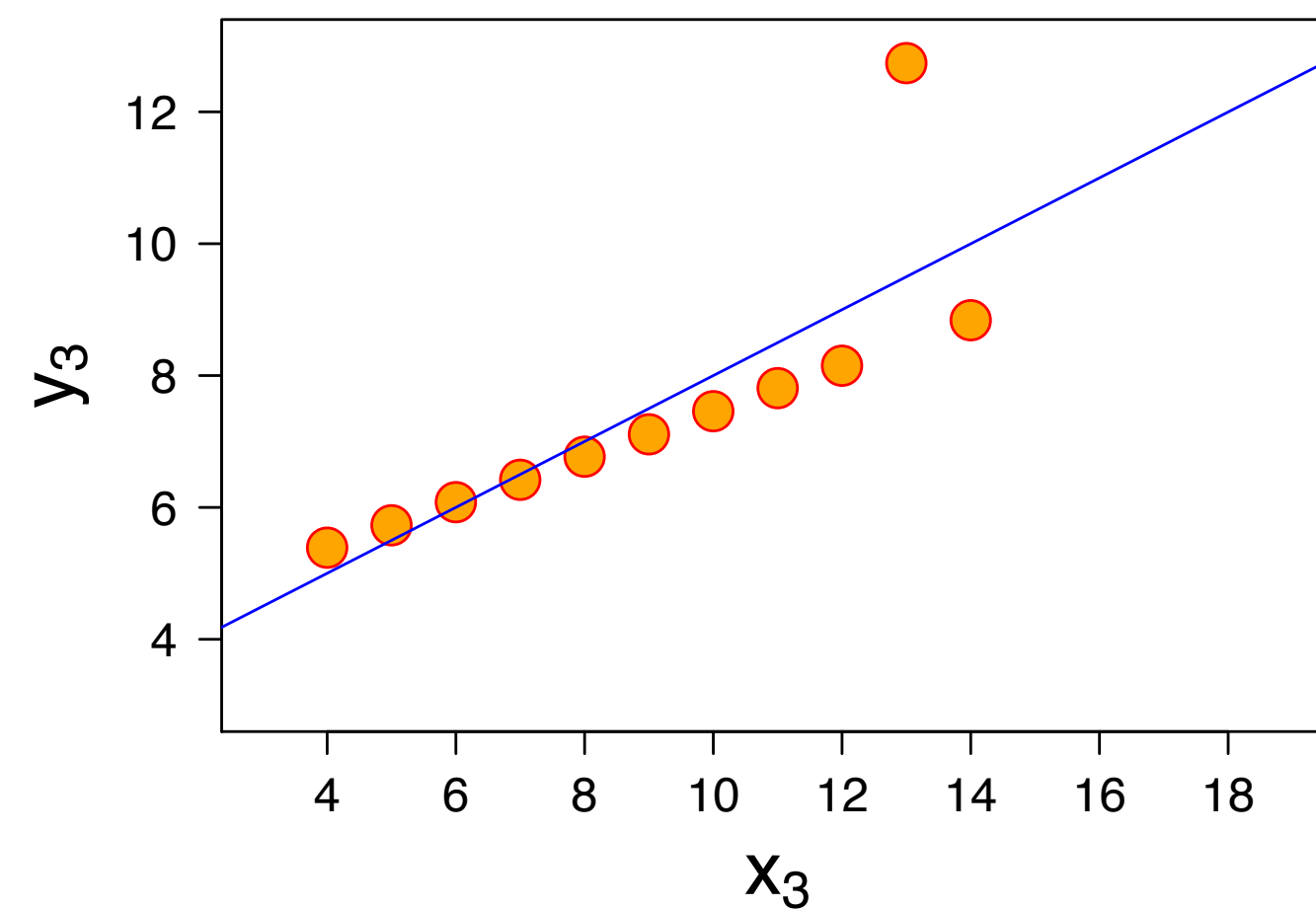
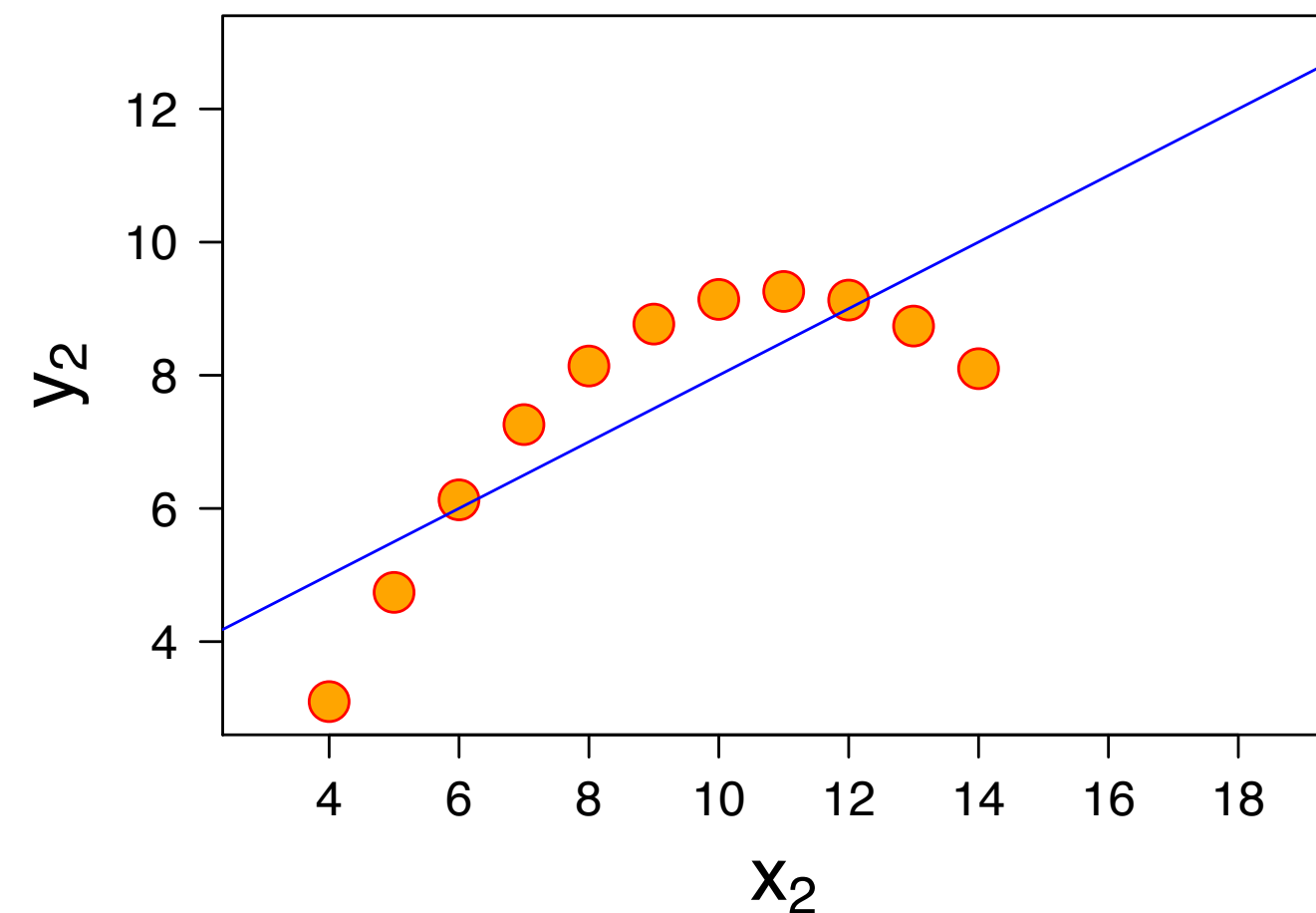
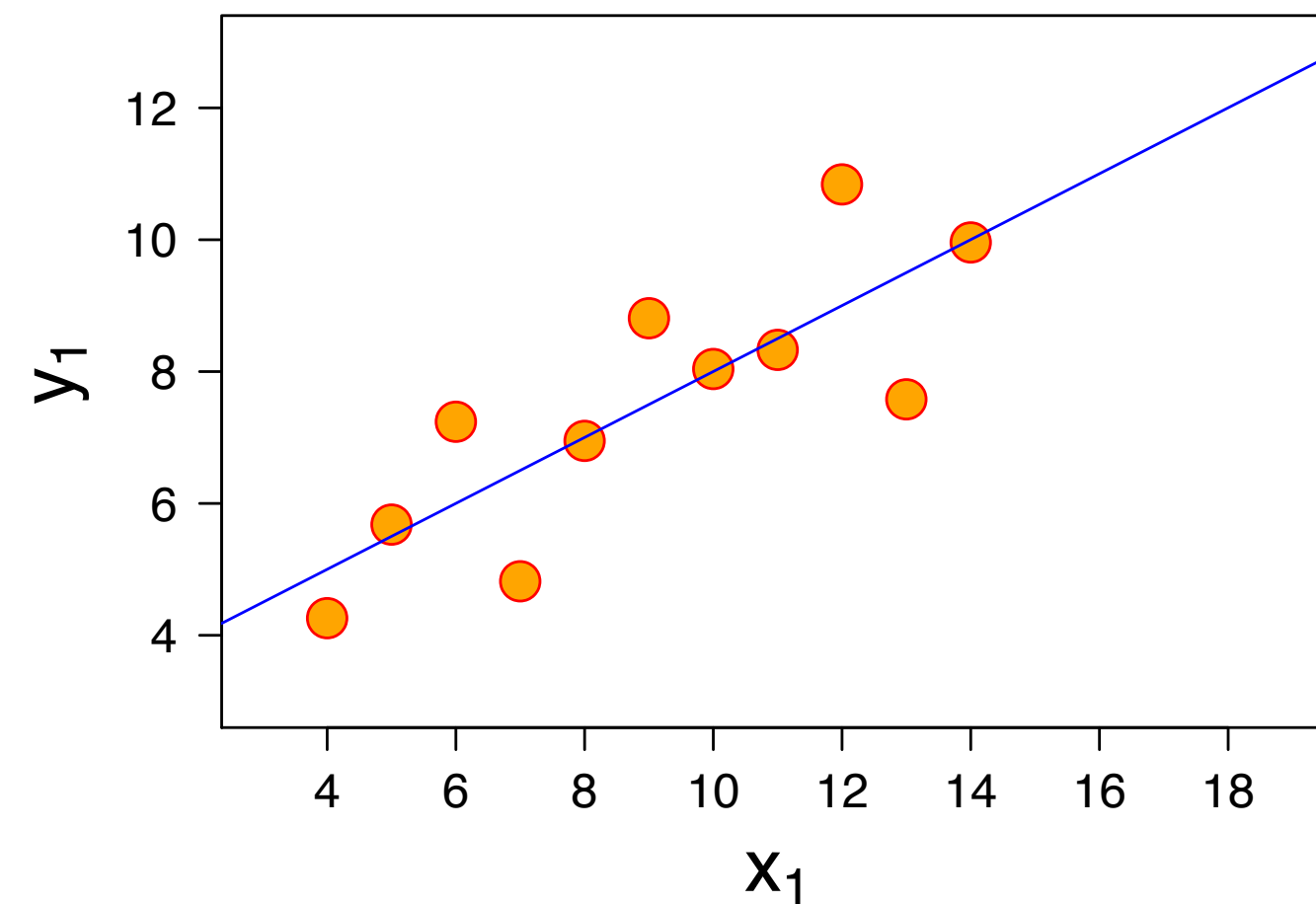
# Why Visual?



[F. J. Anscombe]



# Why Visual?



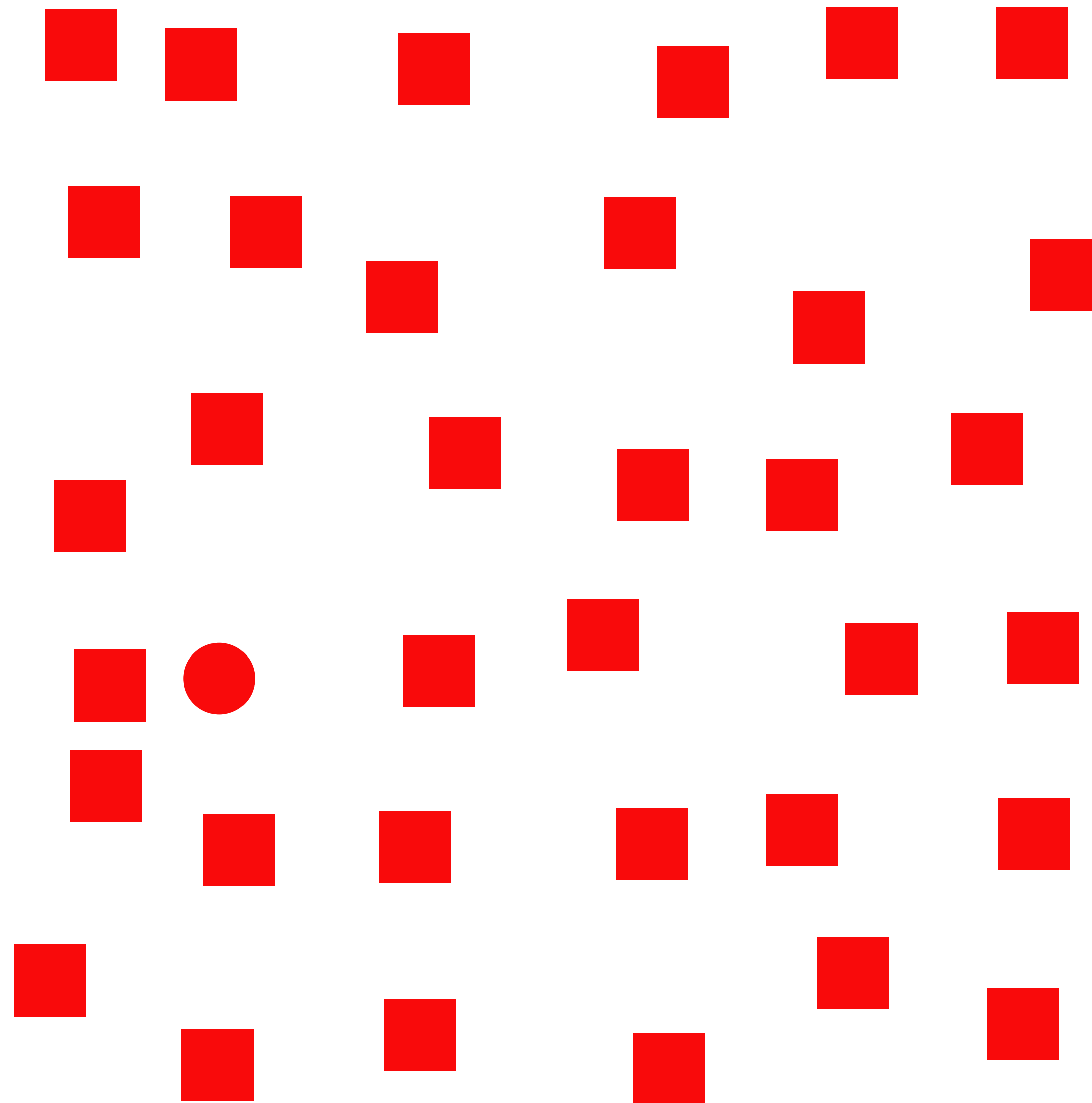
Mean of x	9
Variance of x	11
Mean of y	7.50
Variance of y	4.122
Correlation	0.816

[F. J. Anscombe]



# Visual Pop-out

---



[C. G. Healey]



# Supporting Scalable Visualization

---

- Two Problems:
  - **Lots of data**, how to display (encode) it
  - User **interaction** is key to gaining insight, requires **low latency**
- Addressing big data:
  - Encoding should focus on **available resolution**, not size of data
  - Approaches:
    - Sampling
    - Modeling
    - **Binning**
  - Bin → Aggregate (→ Smooth) → Plot

[J. Heer]



# Effects of Latency

---

- **Higher latency** leads to...
  - Reduced user activity and data set coverage
  - Significantly fewer brushing actions
  - Less observation, generalization & hypothesis
- **Interaction effect:** Exposure to delay reduces subsequent performance in low-latency interface.
- Different interactions exhibit **varied sensitivity** to latency. Brushing is highly sensitive!

[Liu et al. via [J. Heer](#)]



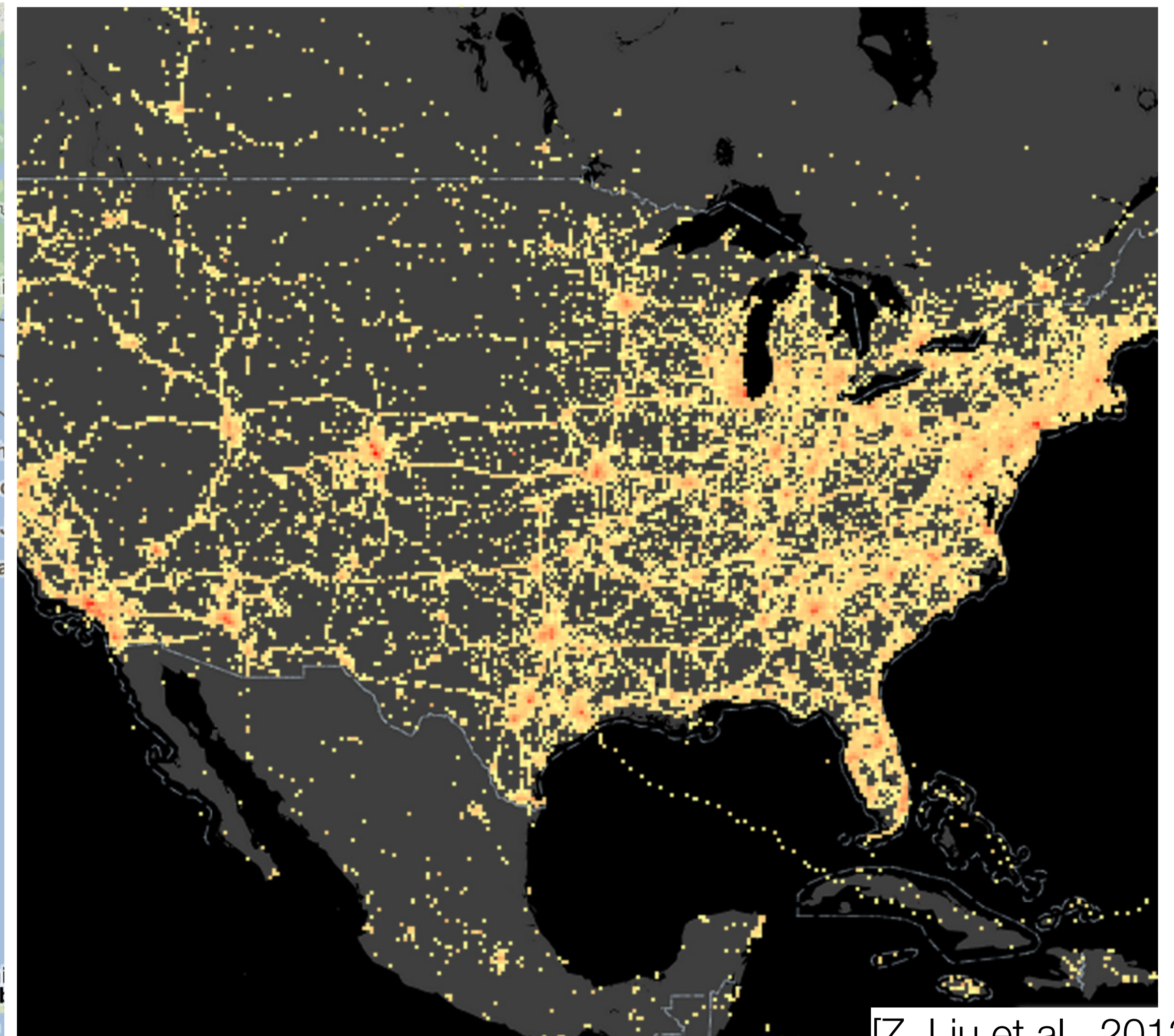
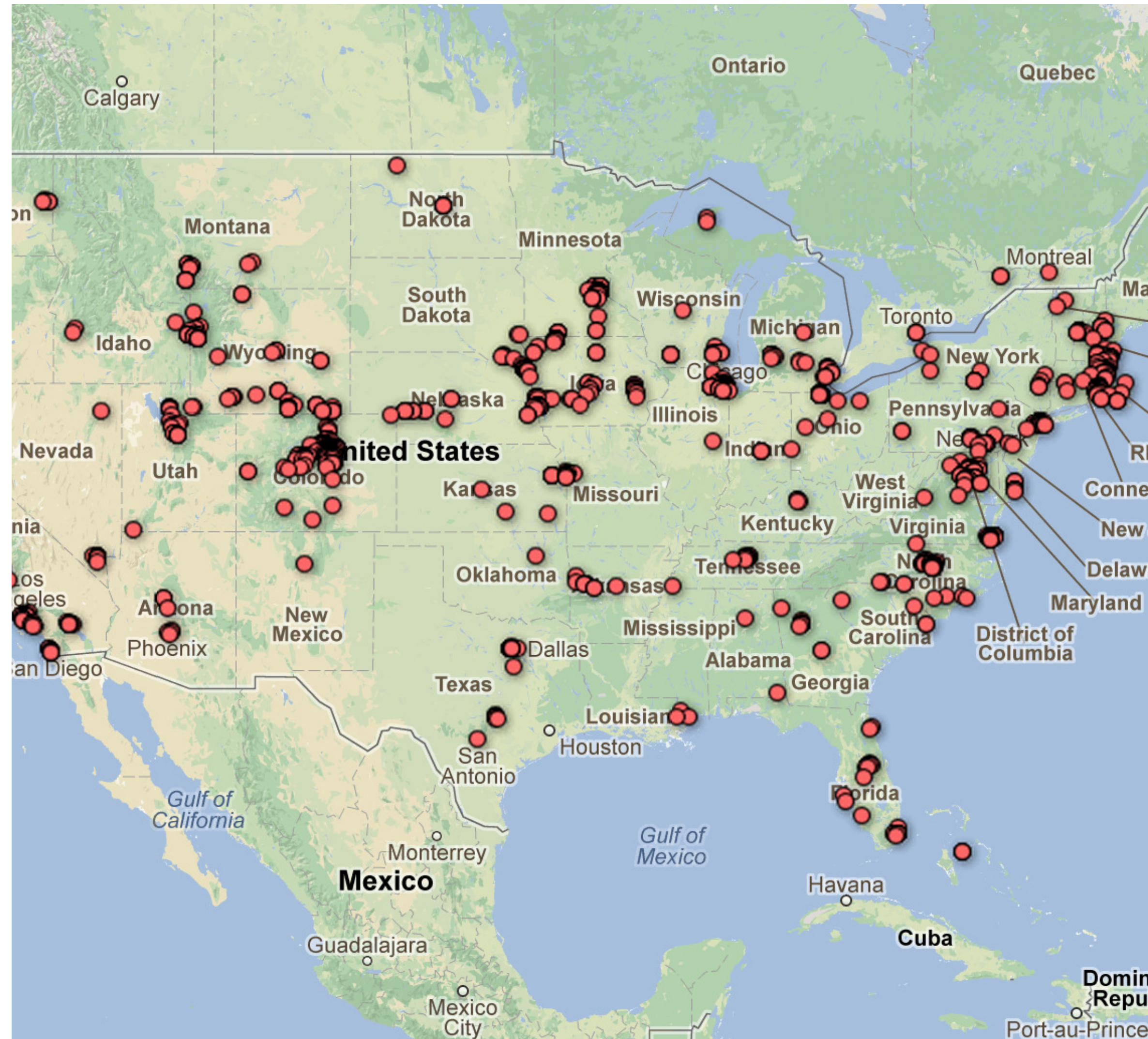
# Interactive Scalability Solutions

---

- Use Database Technology: databases built for scalability
- Client-side Indexing (Data Cubes): take advantage of data structure
- Prefetching: load data before requests based on predictions
- Approximation: show estimates early but with error information



# Sampling vs. Aggregation

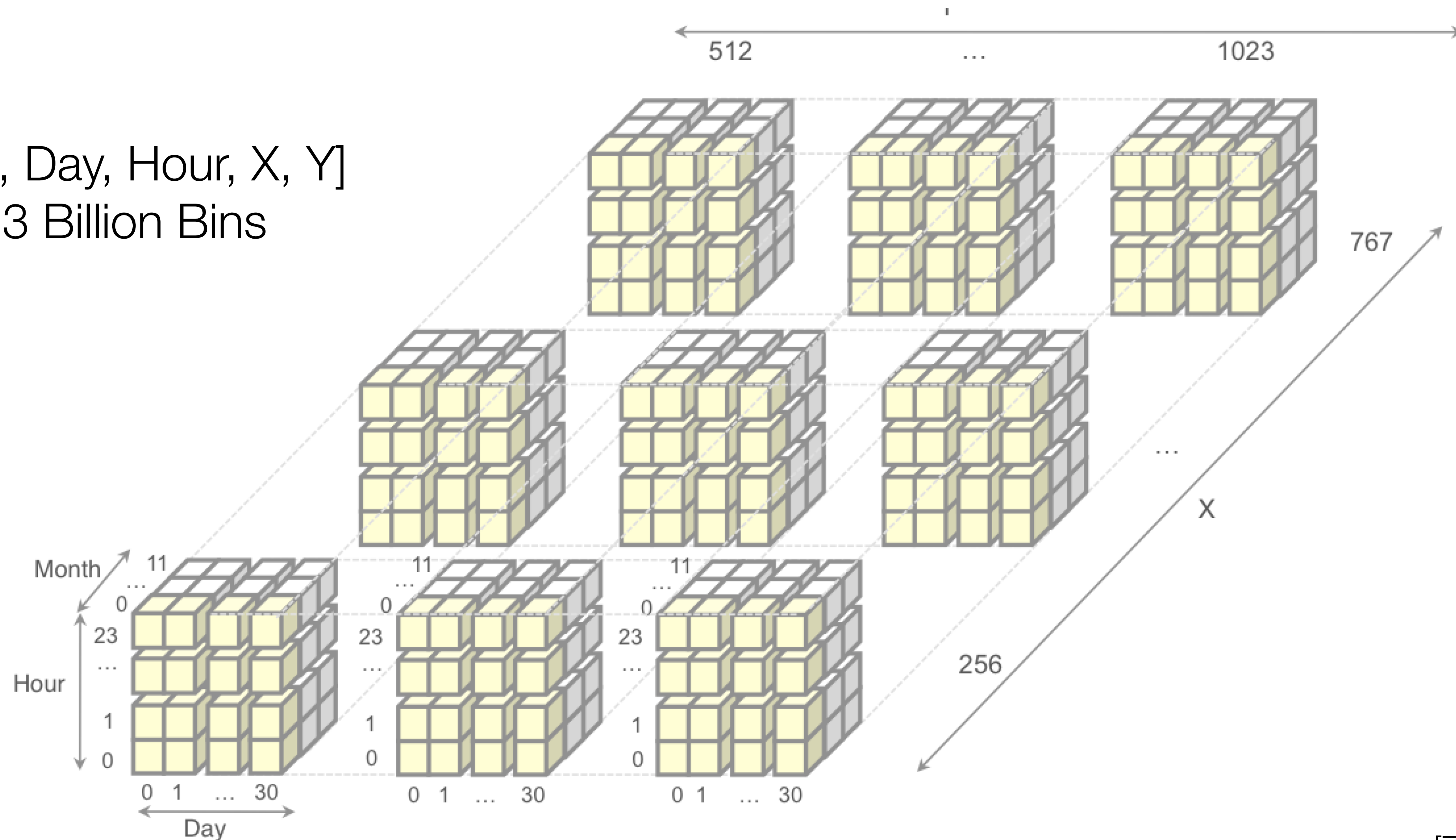


[Z. Liu et al., 2013]



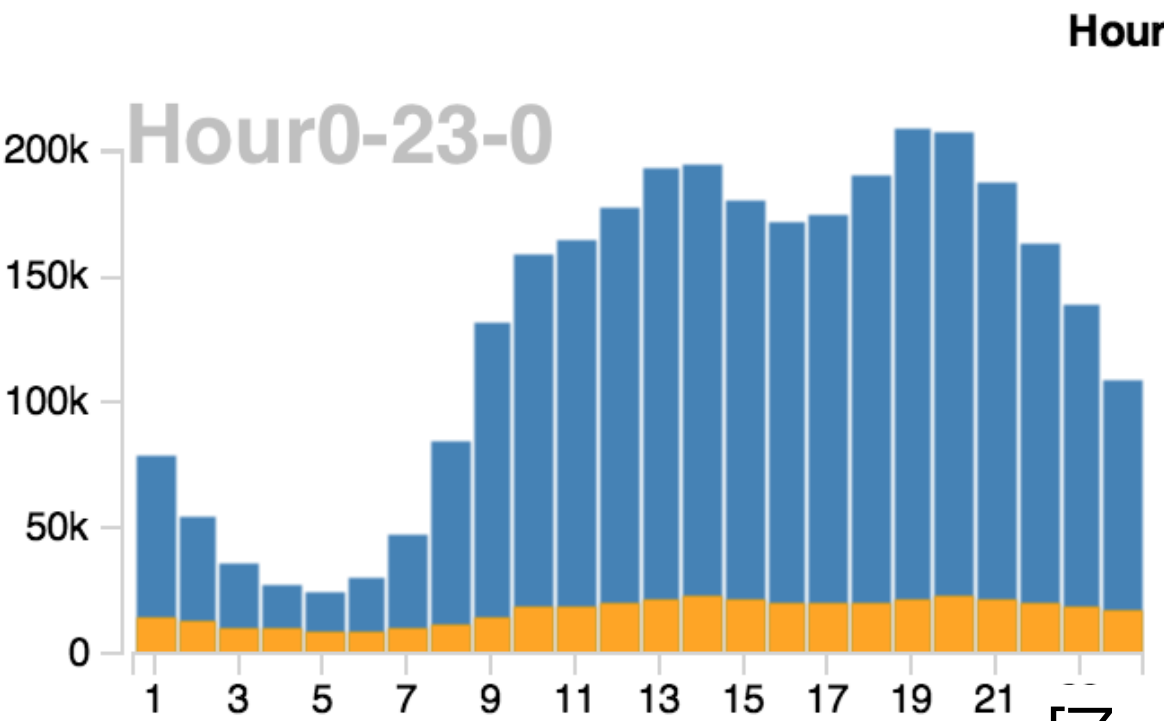
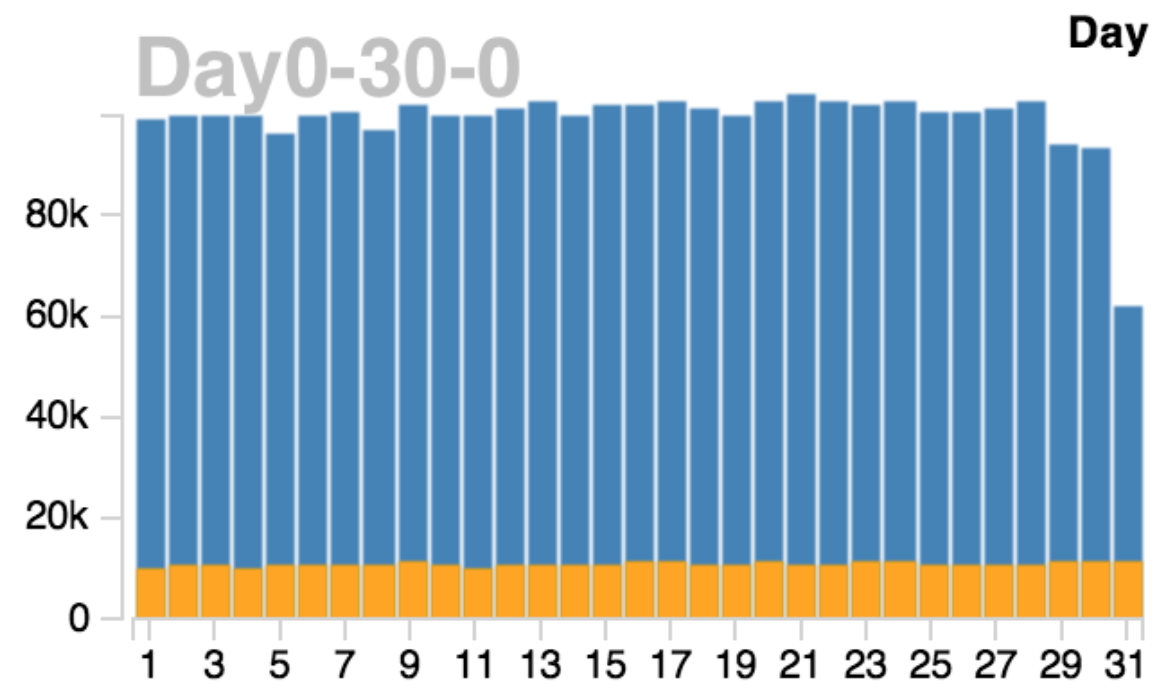
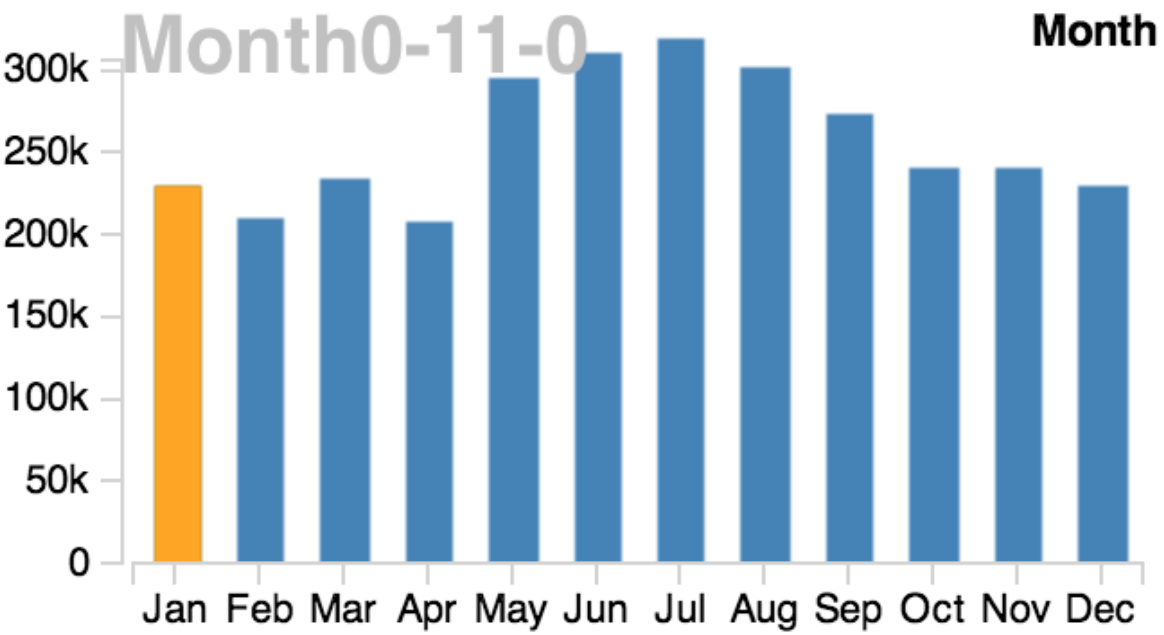
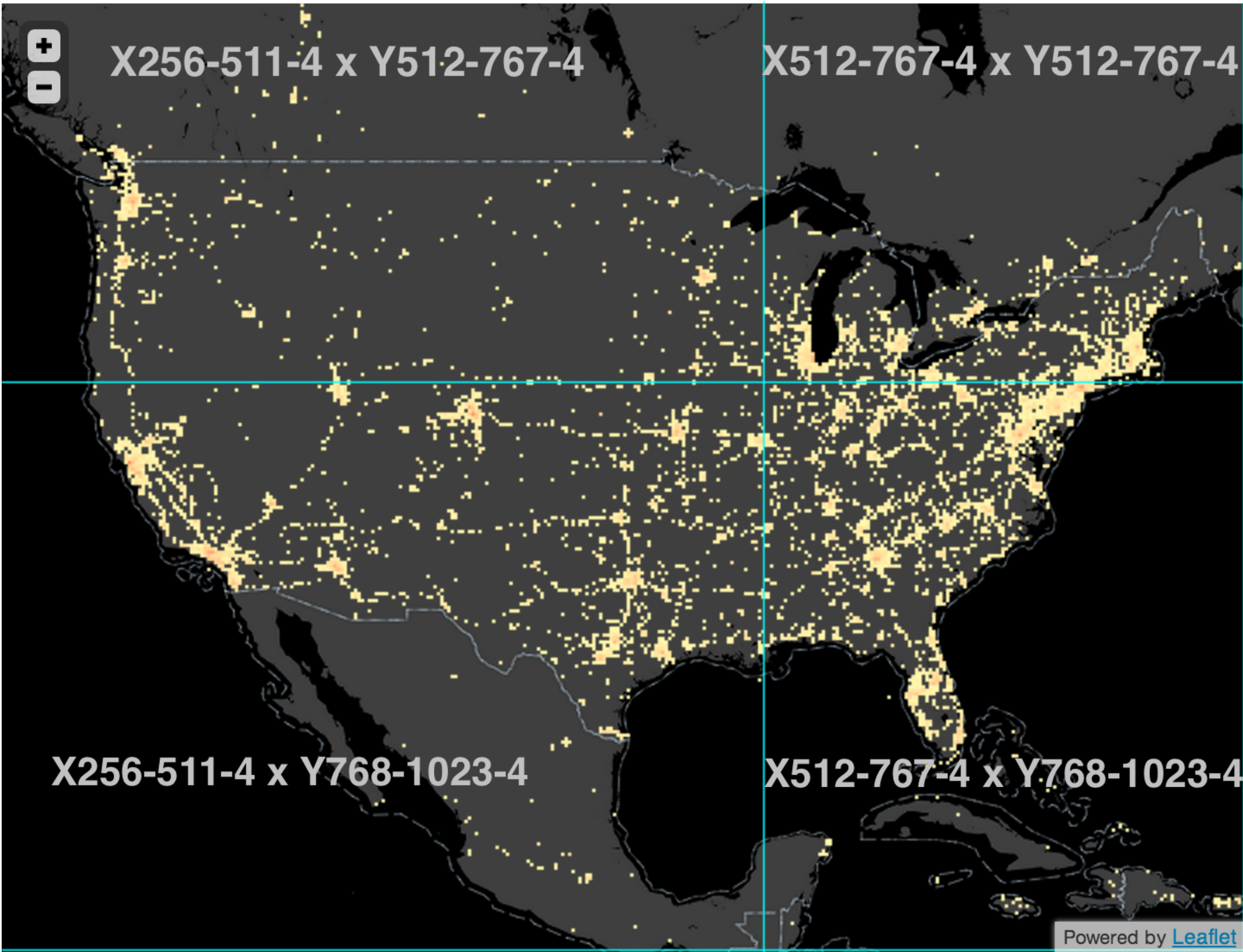
# Full 5-D Data Cube

[Month, Day, Hour, X, Y]  
~2.3 Billion Bins



[Z. Liu et al., 2013]

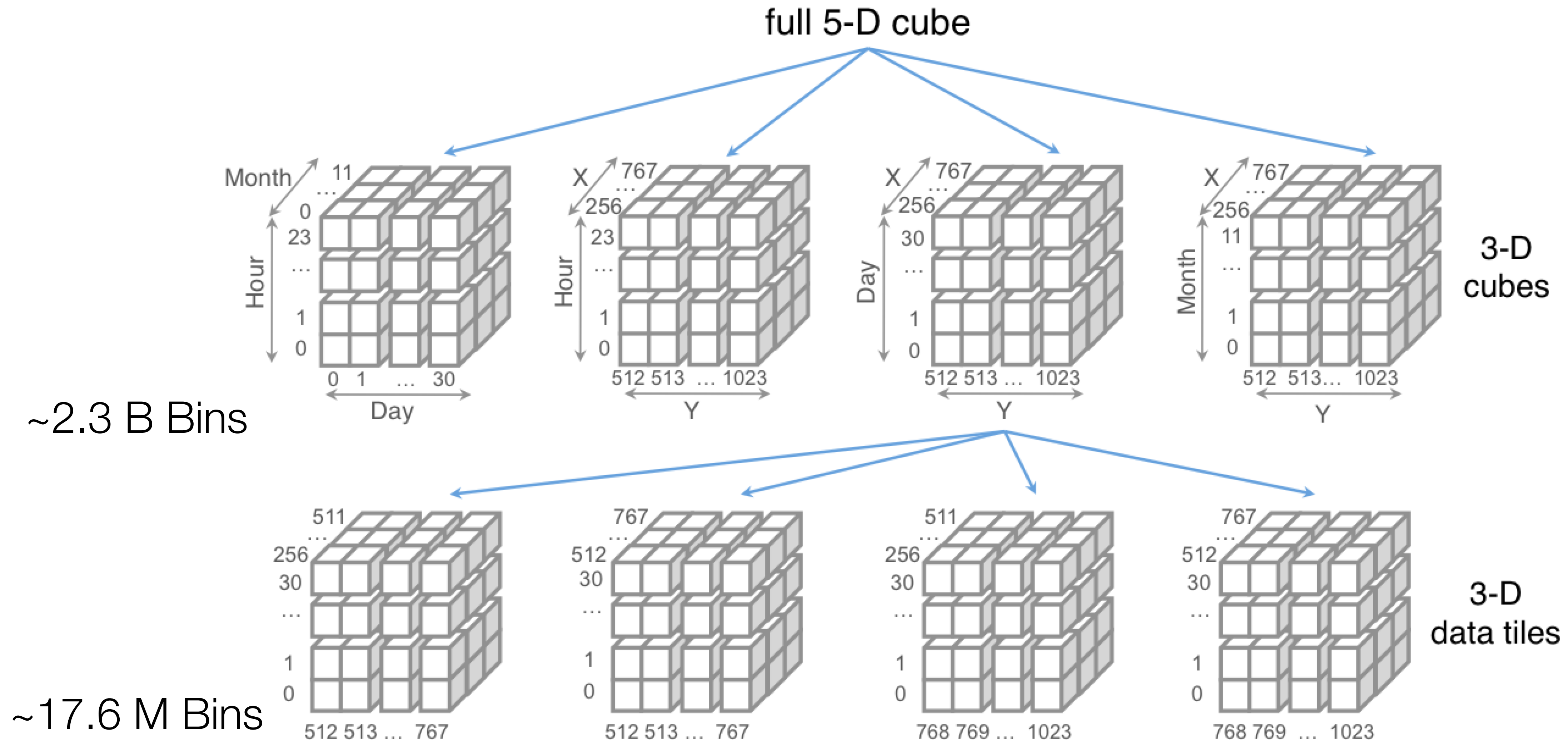
# Break into Tiles



[Z. Liu et al., 2013]

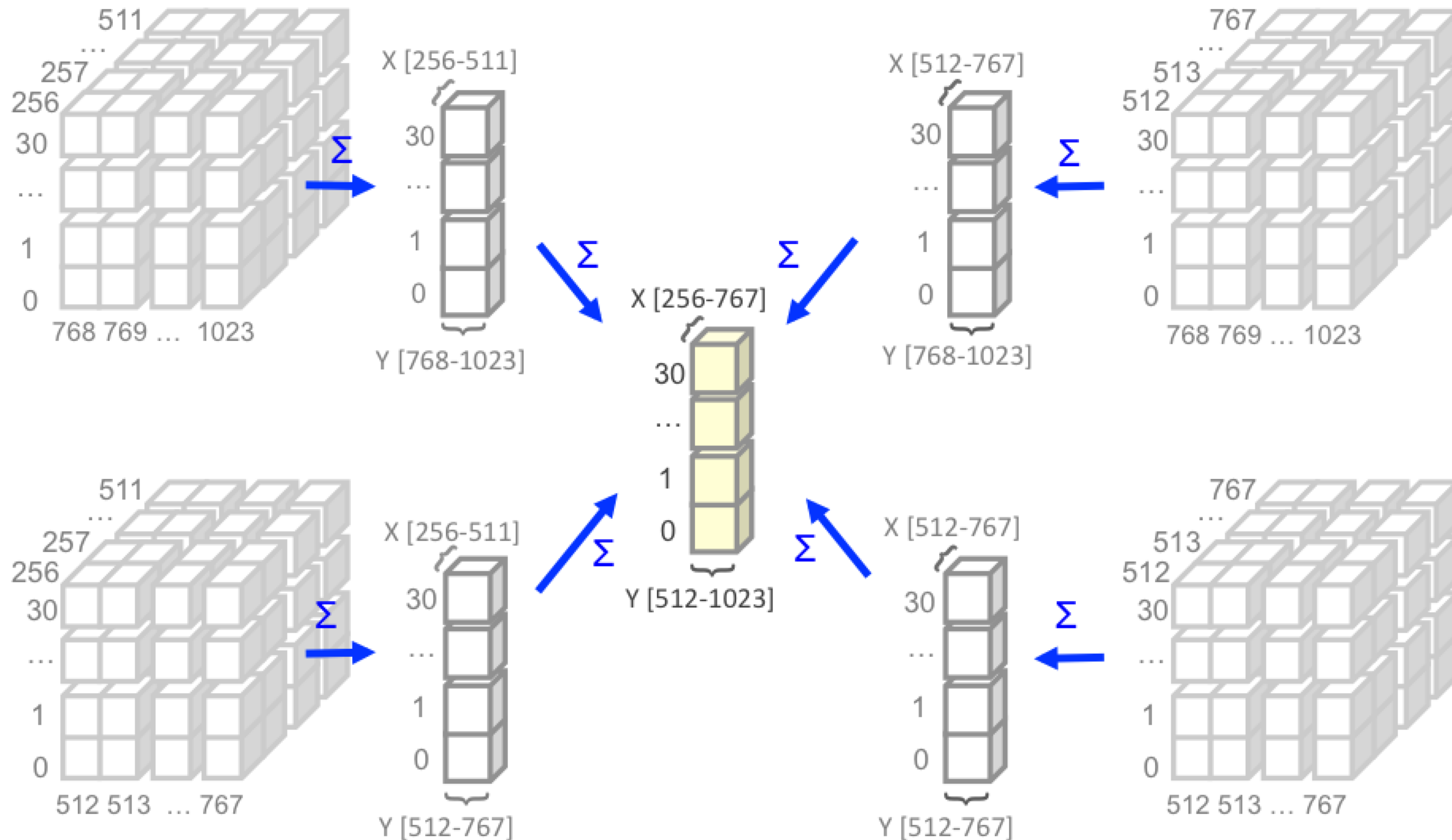


# Data Cube Decomposition



[Z. Liu et al., 2013]

# Aggregation of Tiles for Results

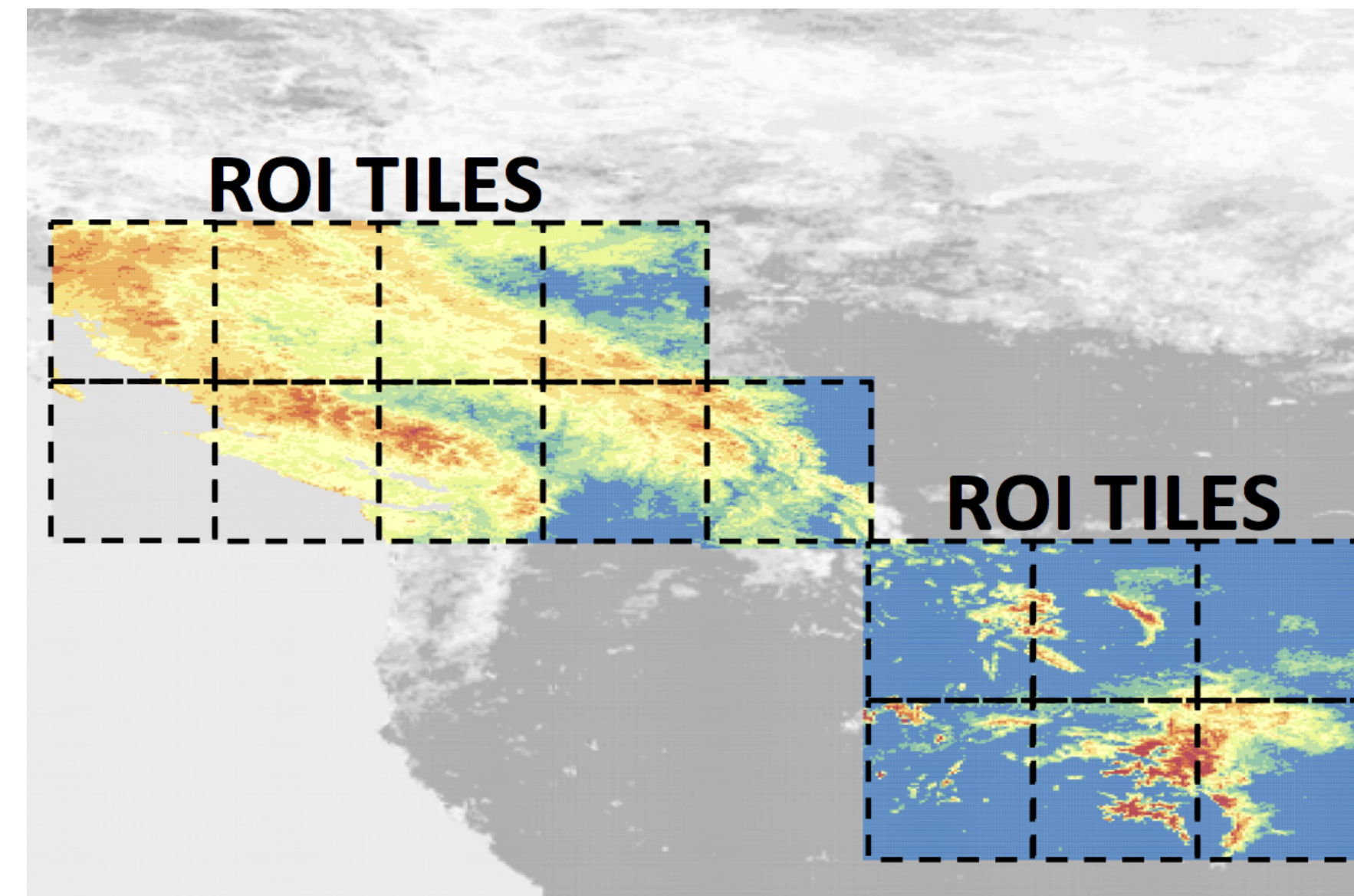
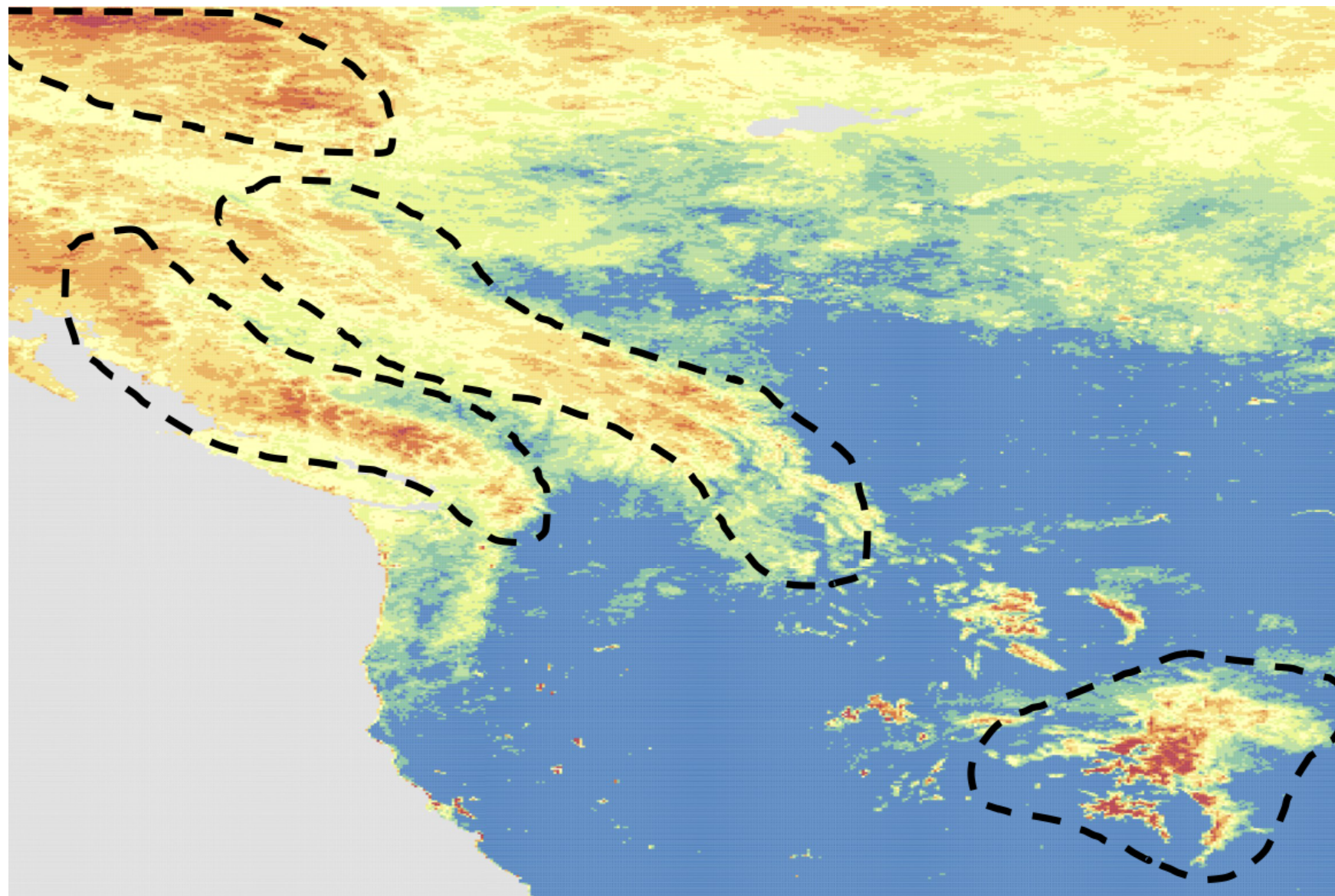


[Z. Liu et al., 2013]



# Prefetching (ForeCache)

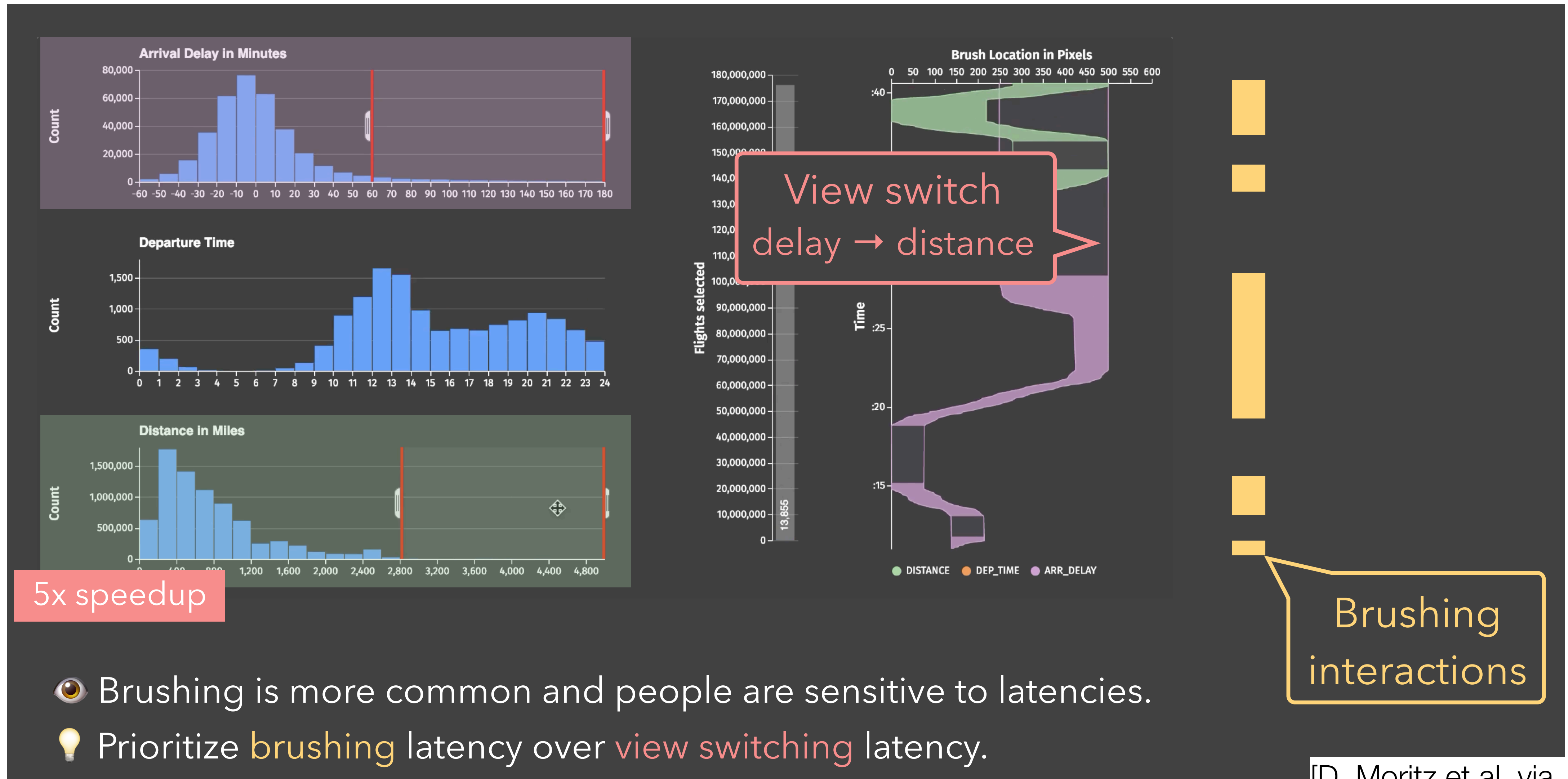
- Predict which tiles a user will need next and prefetch those
  - Use common patterns (zoom, pan)
  - Use regions of interest (ROIs)



[Battle et al., 2016]



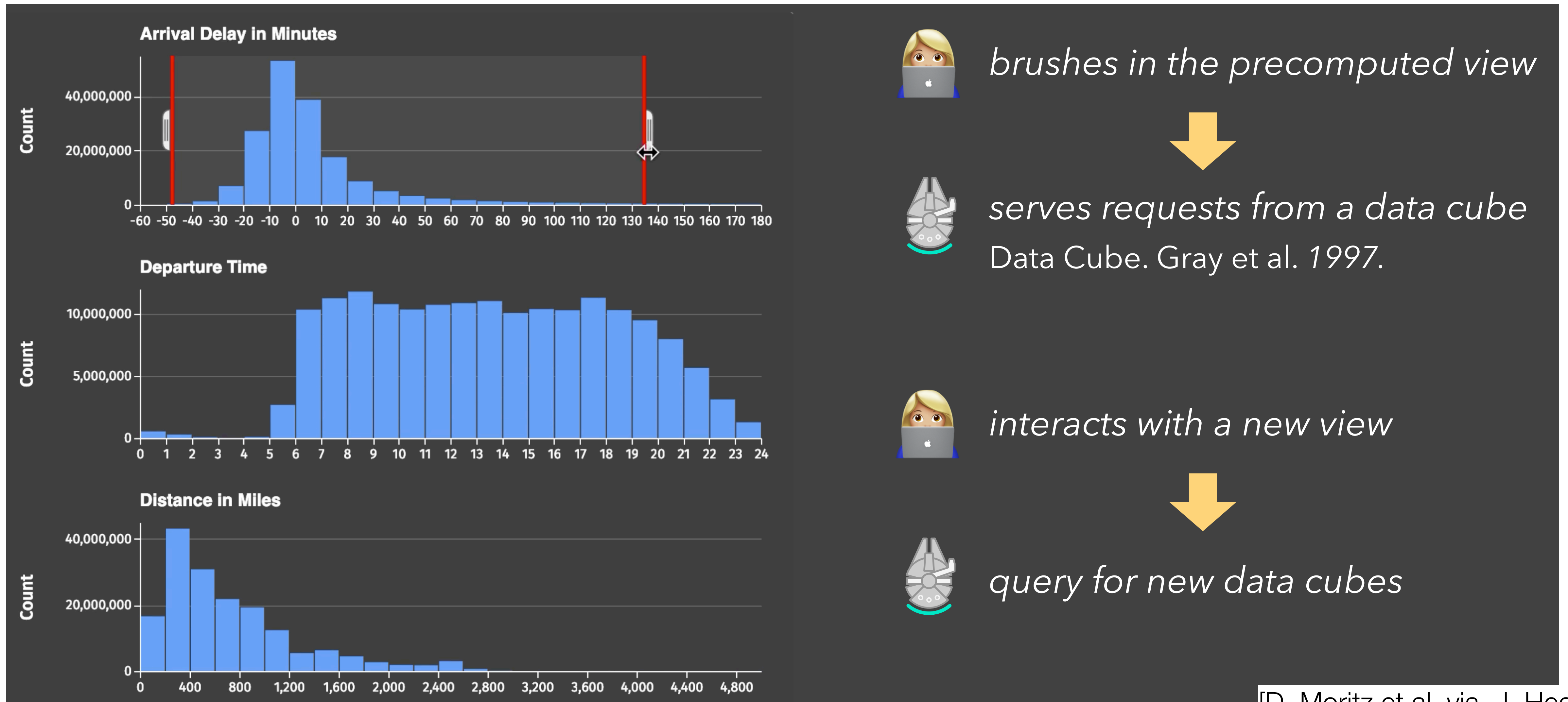
# Latency Differences in Tasks



[D. Moritz et al. via [J. Heer](#)]



# Task-Prioritized Prefetching



[D. Moritz et al. via [J. Heer](#)]



# Reading Quiz

---

# Test 2

---

- Monday, March 29, "in class" on Blackboard, 3:30-4:45pm
- Covers material from the beginning of course, emphasizing material since Test 1
- Similar Format to Test 1

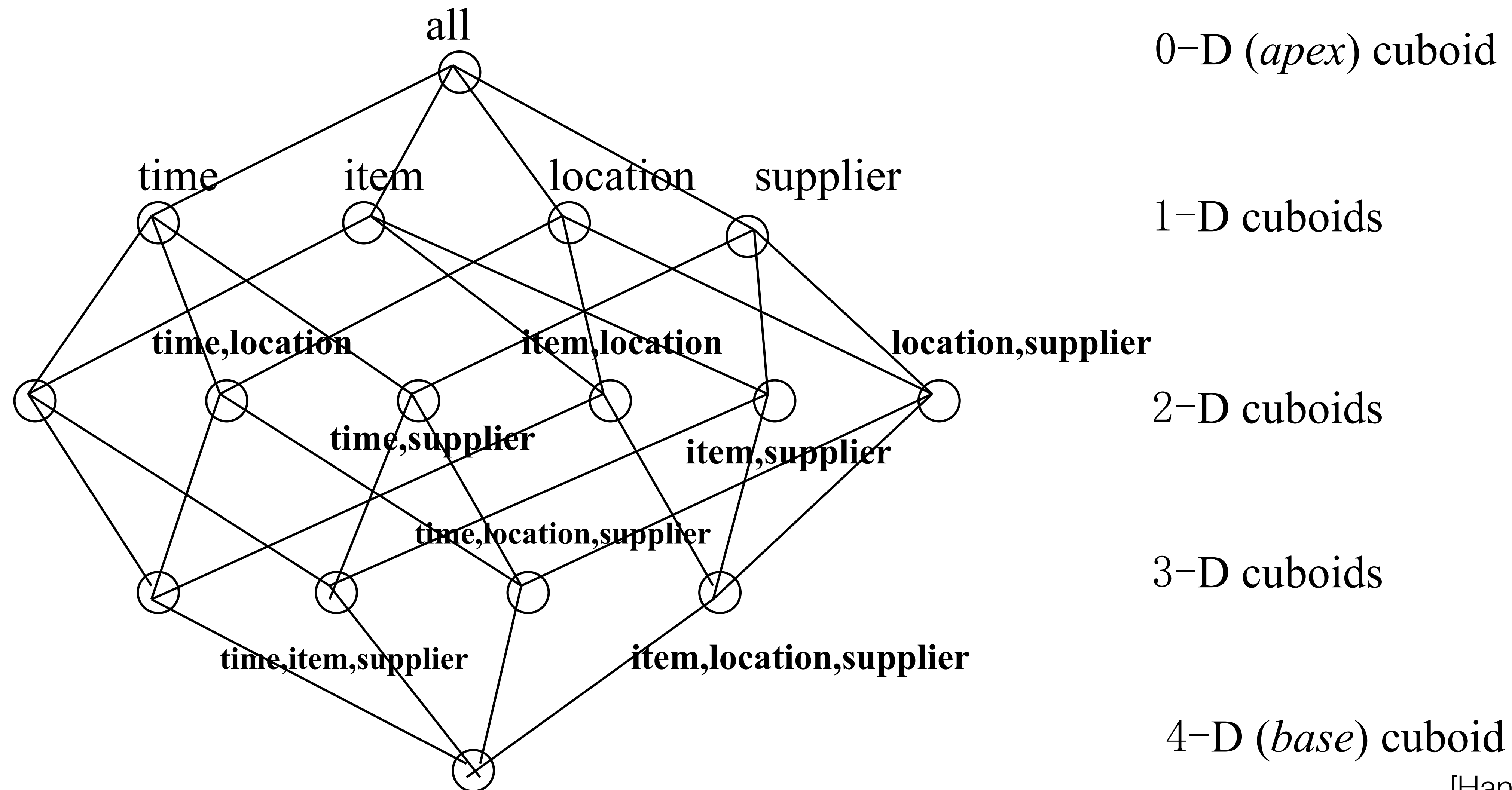


# Data Cubes

---

J. Han, M. Kamber, and J. Pei

# Data Cube: A Lattice of Cuboids



[Han et al., 2011]



# Cube Operations

---

- Roll-up: aggregate up the given hierarchy
- Drill-down: refine down the given hierarchy
- Roll-up and drill-down are "inverses"

# Spatial Data Exploration Motivation

---

L. Battle

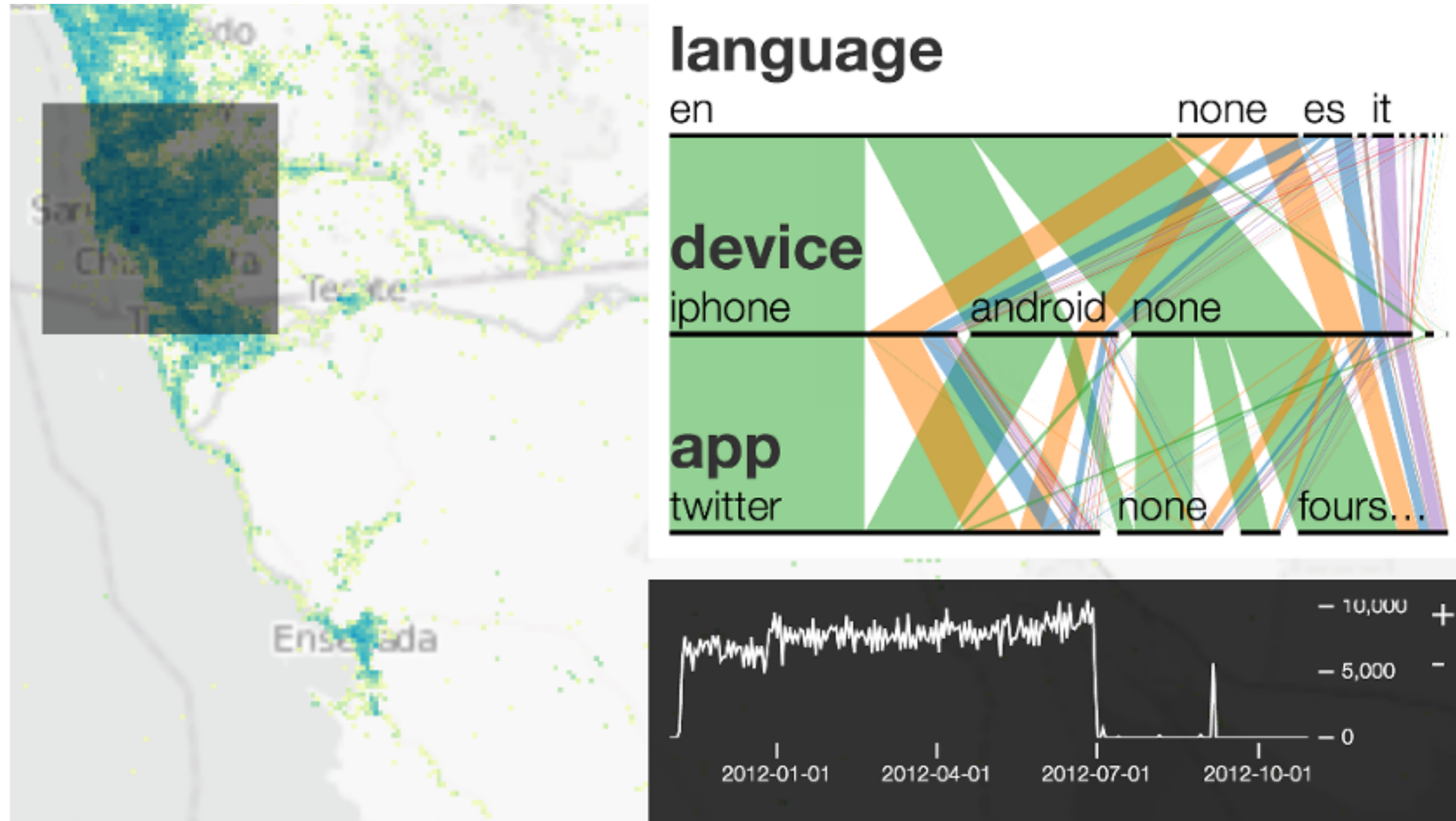


# Nanocubes for Real-Time Exploration of Spatiotemporal Datasets

---

L. Lins, J. T. Klosowski, and C. Scheidegger

# Goal: Interactive Exploration of Data Cubes

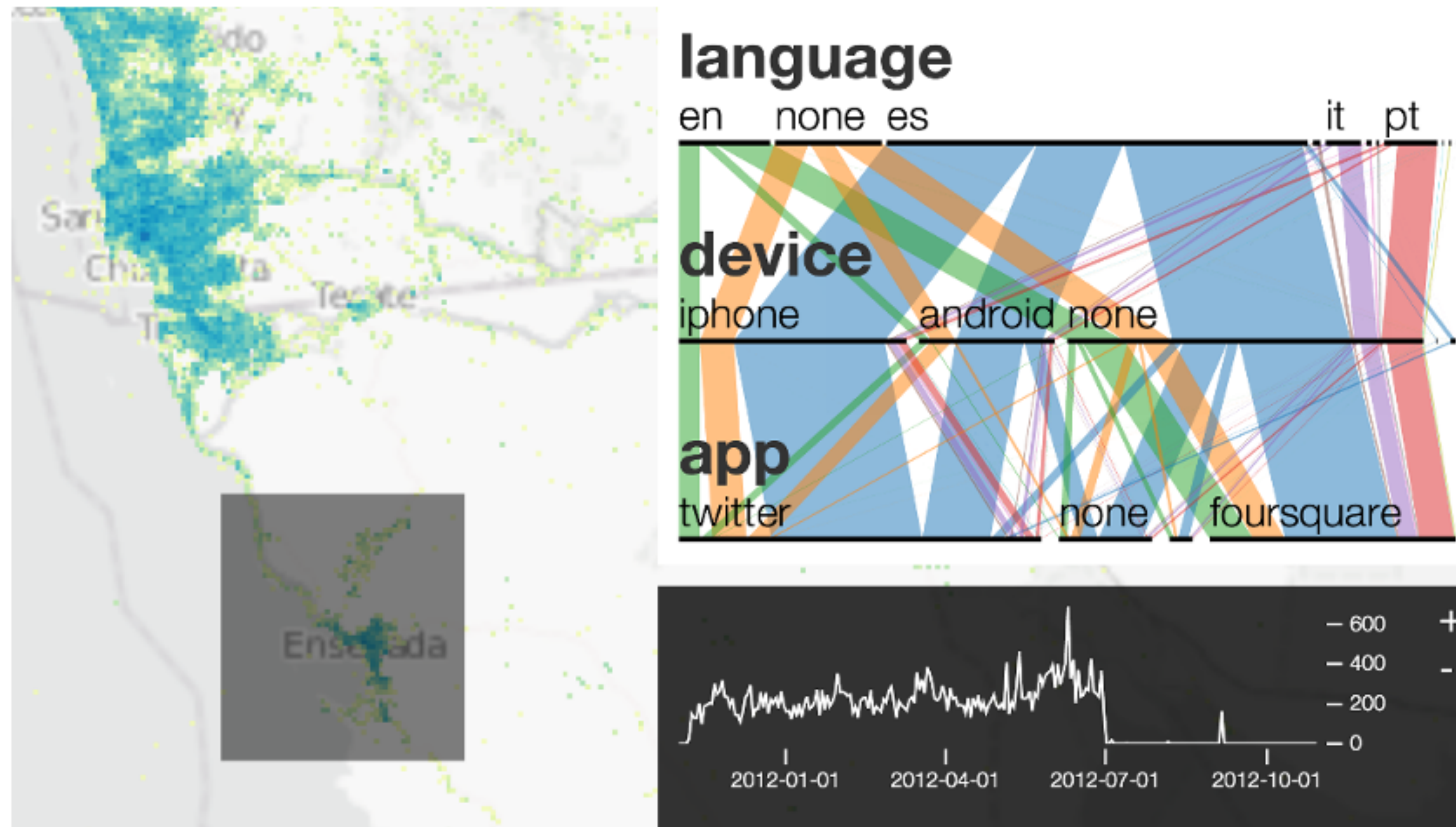


Linked view of tweets in San Diego, US

[Lins et. al, 2013]



# Move to Another Location

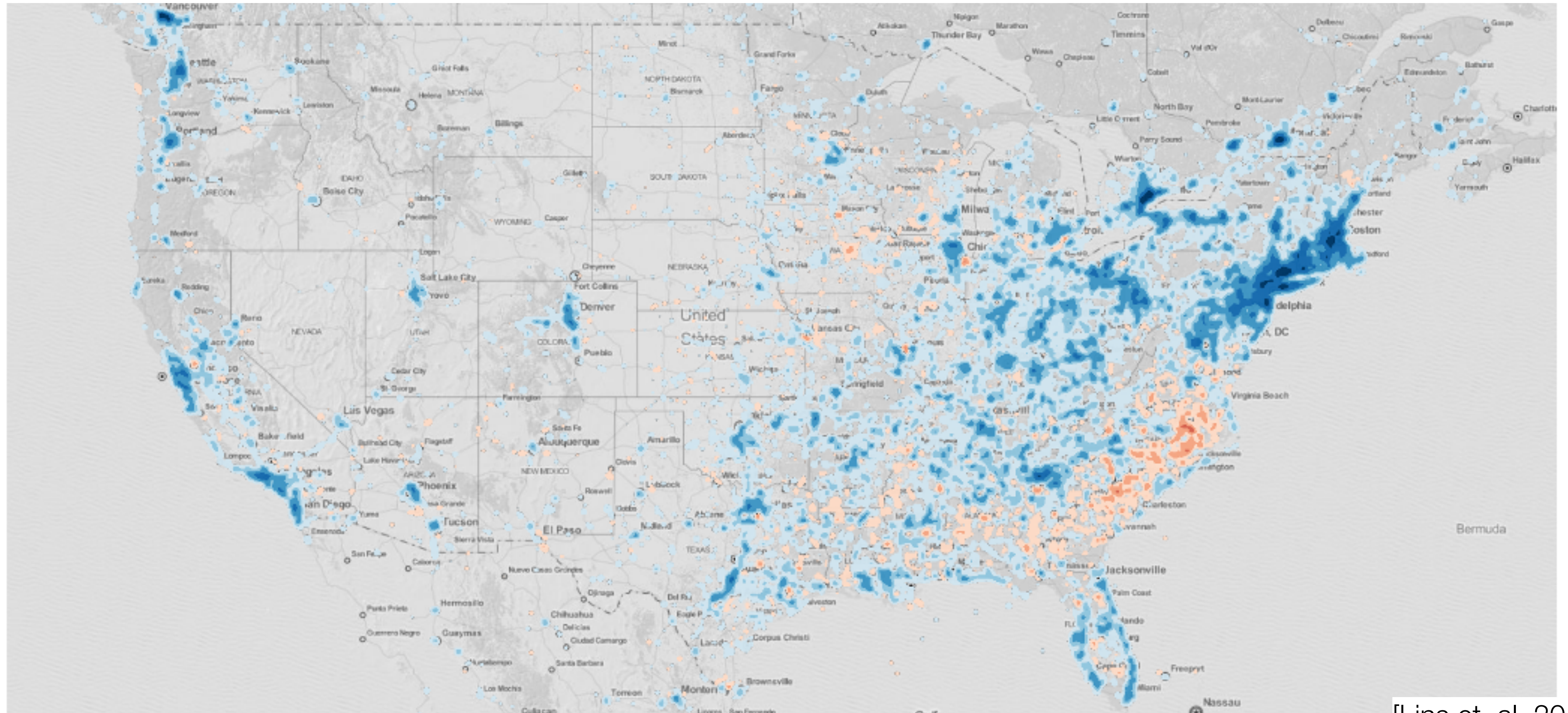


Linked view of tweets in Ensenada, Mexico

[Lins et. al, 2013]



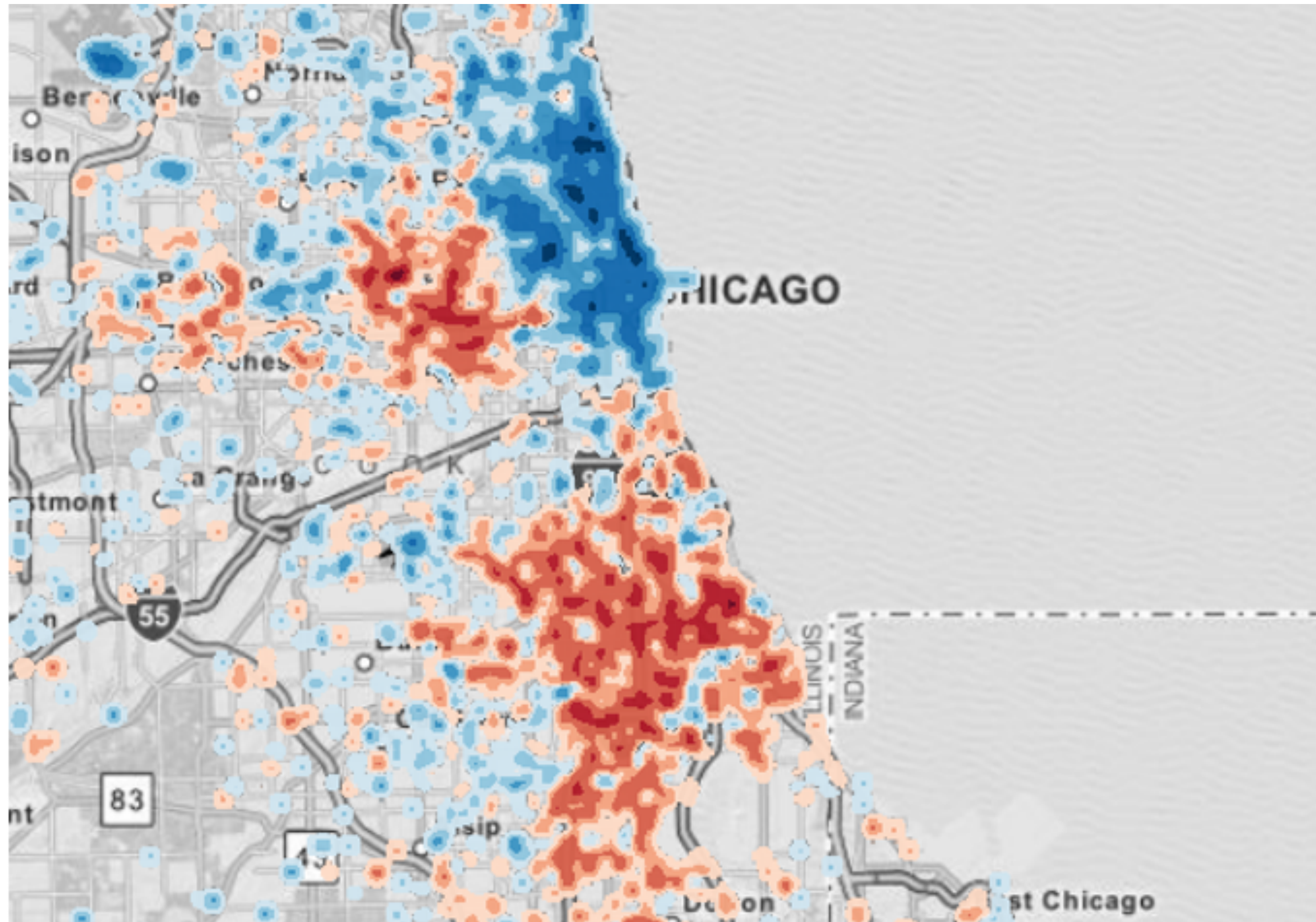
# iPhone vs. Android Map



[Lins et. al, 2013]



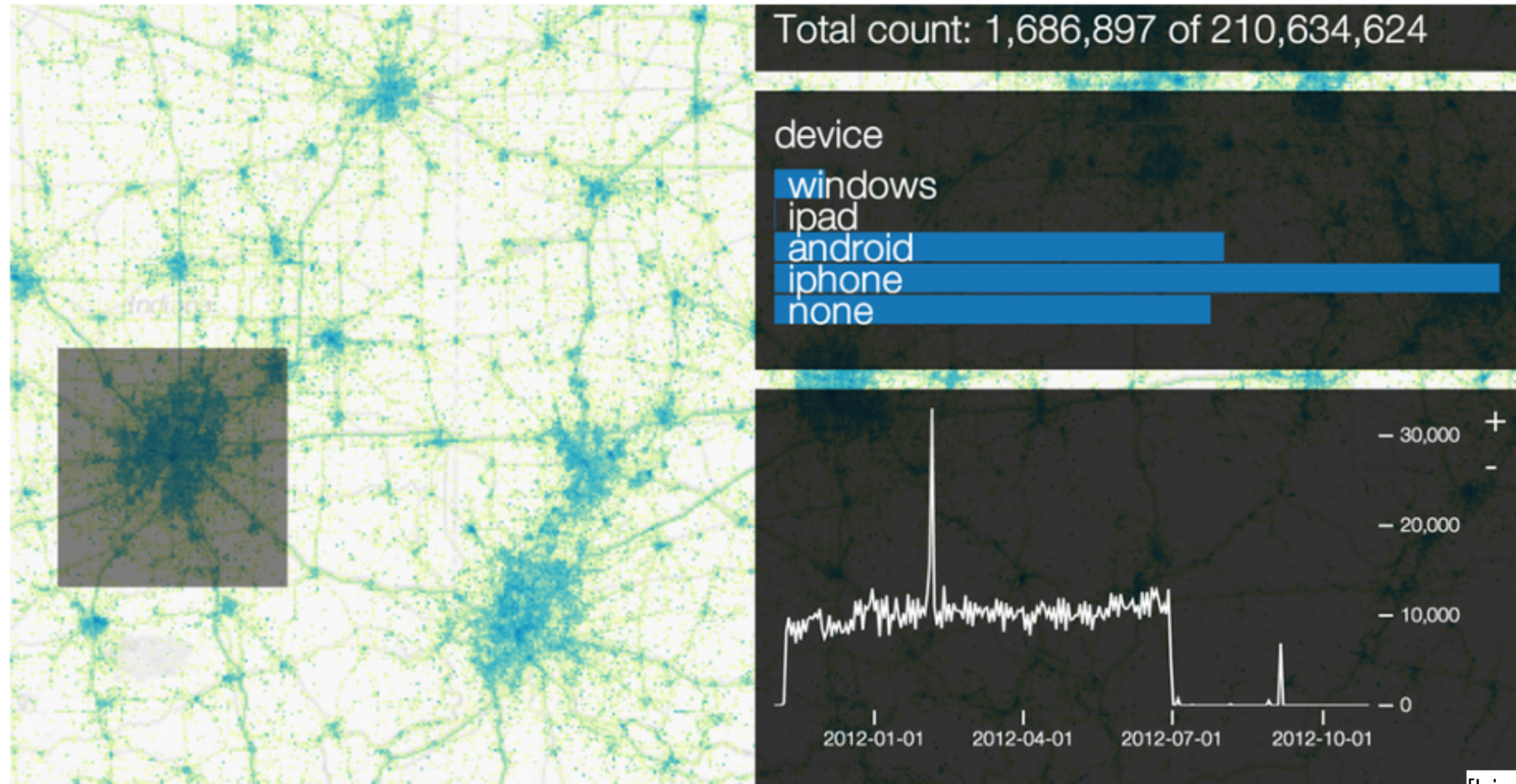
# Zoom into Chicago



[Lins et. al, 2013]



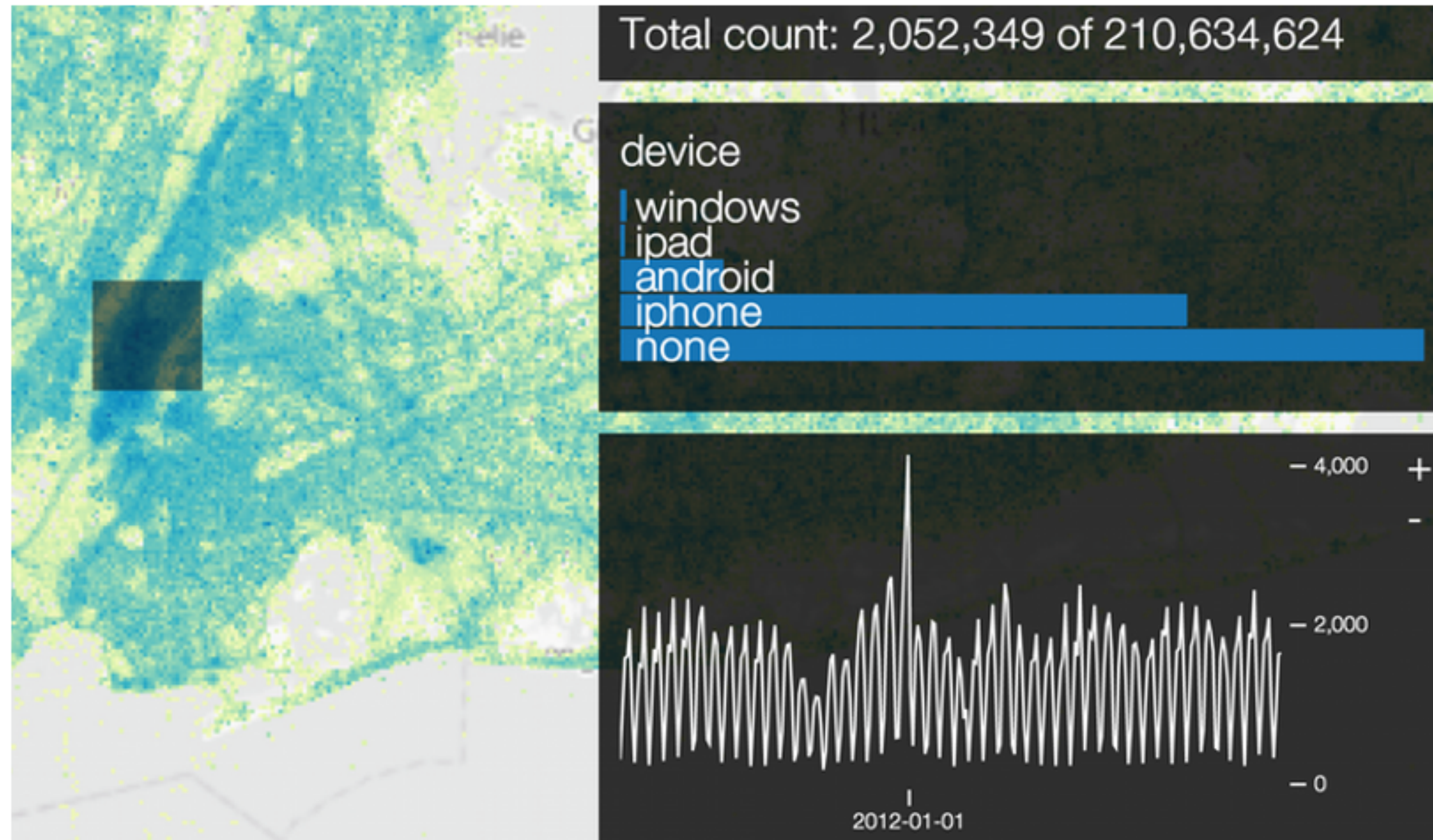
# SuperBowl in Indianapolis



[Lins et. al, 2013]



# New Year's Eve in Manhattan



[Lins et. al, 2013]



# Aggregations on Spatiotemporal Data

---

- Spatial: e.g. counting events in a spatial region (world or San Fran.)
- Temporal: e.g. queries at multiple scales (hour, day, week, month)
- Seek to address Visual Information Seeking Mantra:
- Overview first, zoom and filter, details-on-demand
- Multidimensional:
  - Latitude, Longitude, Time + **more**



# Data Cube Aggregations

Relation **A**

<i>Country</i>	<i>Device</i>	<i>Language</i>
US	Android	en
US	iPhone	ru
South Africa	iPhone	en
India	Android	en
Australia	iPhone	en

Aggregation **B**

<i>Country</i>	<i>Device</i>	<i>Language</i>	<i>Count</i>
<i>All</i>	<i>All</i>	<i>All</i>	5

Group By on *Device, Language* **C**

<i>Country</i>	<i>Device</i>	<i>Language</i>	<i>Count</i>
<i>All</i>	Android	en	2
<i>All</i>	iPhone	en	2
<i>All</i>	iPhone	ru	1

Cube on *Device, Language*

**D**

<i>Country</i>	<i>Device</i>	<i>Language</i>	<i>Count</i>
<i>All</i>	<i>All</i>	<i>All</i>	5
<i>All</i>	Android	<i>All</i>	2
<i>All</i>	iPhone	<i>All</i>	3
<i>All</i>	<i>All</i>	en	4
<i>All</i>	<i>All</i>	ru	1
<i>All</i>	iPhone	ru	1
<i>All</i>	Android	en	2
<i>All</i>	iPhone	en	2

Equivalent to Group By on  
all possible subsets of  
{*Device, Language*}

[Lins et. al, 2013]

# Nanocube Queries

- Representing natural language queries as data cube queries

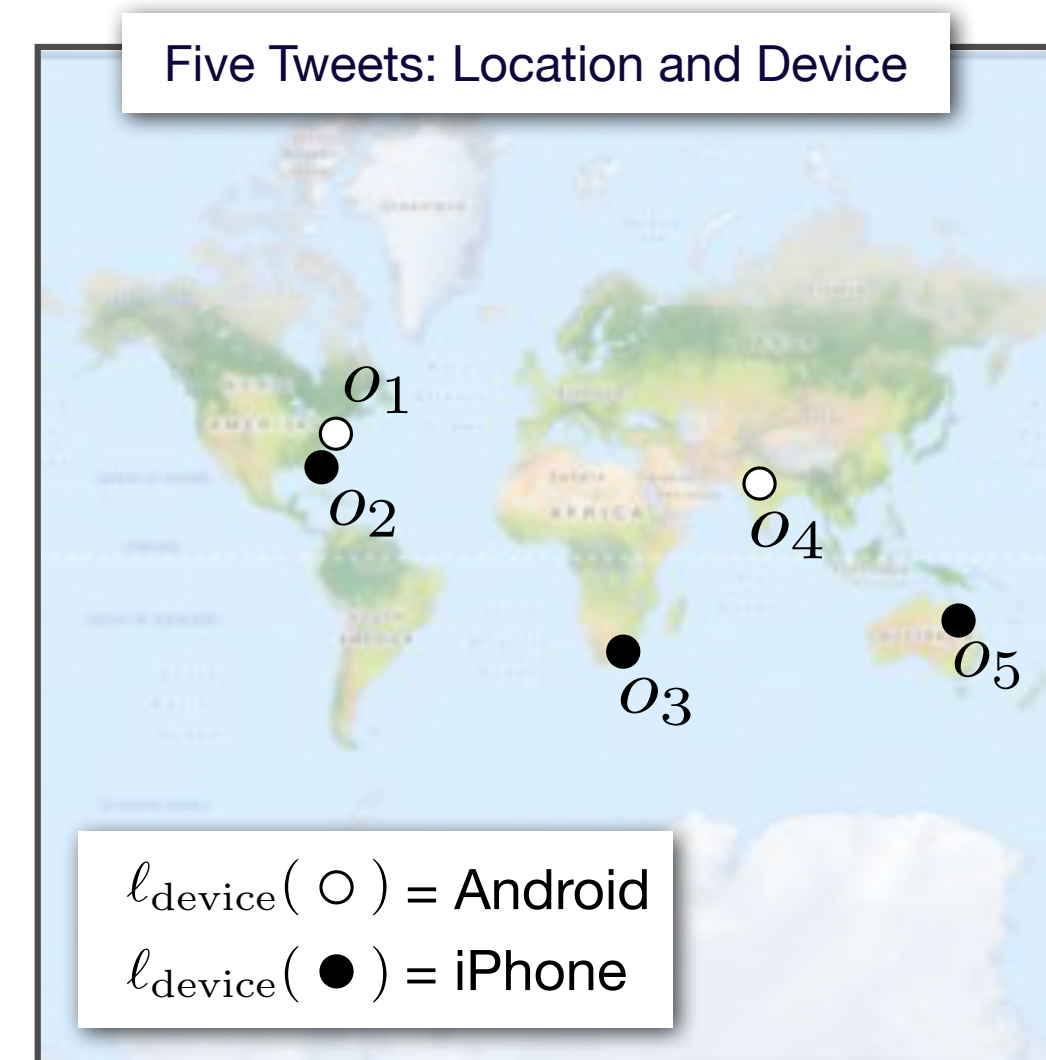
Natural language query	s	c	t	URL
count of all Delta flights	<i>R</i> <i>U</i>	<i>R</i> { Delta }	<i>R</i> <i>U</i>	/where/carrier=Delta
count of all Delta flights in the Midwest	<i>R</i> Midwest	<i>R</i> { Delta }	<i>R</i> <i>U</i>	/region/Midwest/where/carrier=Delta
count of all flights in 2010	<i>R</i> <i>U</i>	<i>D</i>	<i>R</i> 2010	/field/carrier/when/2010
time-series of all United flights in 2009	<i>R</i> <i>U</i>	<i>R</i> { United }	<i>D</i> 2009	/tseries/when/2009/where/carrier=United
heatmap of Delta flights in 2010	<i>D</i> tile0	<i>R</i> { Delta }	<i>R</i> 2010	/tile/tile0/when/2010/where/carrier=Delta

- s = space, c = category, t = time
- R = rollup, D = drill down
- <value> after RD = subset of dimension's domain, U = universe
- Note that time queries are stored in an array of cumulative counts

[Lins et. al, 2013]



# Building a Nanocube

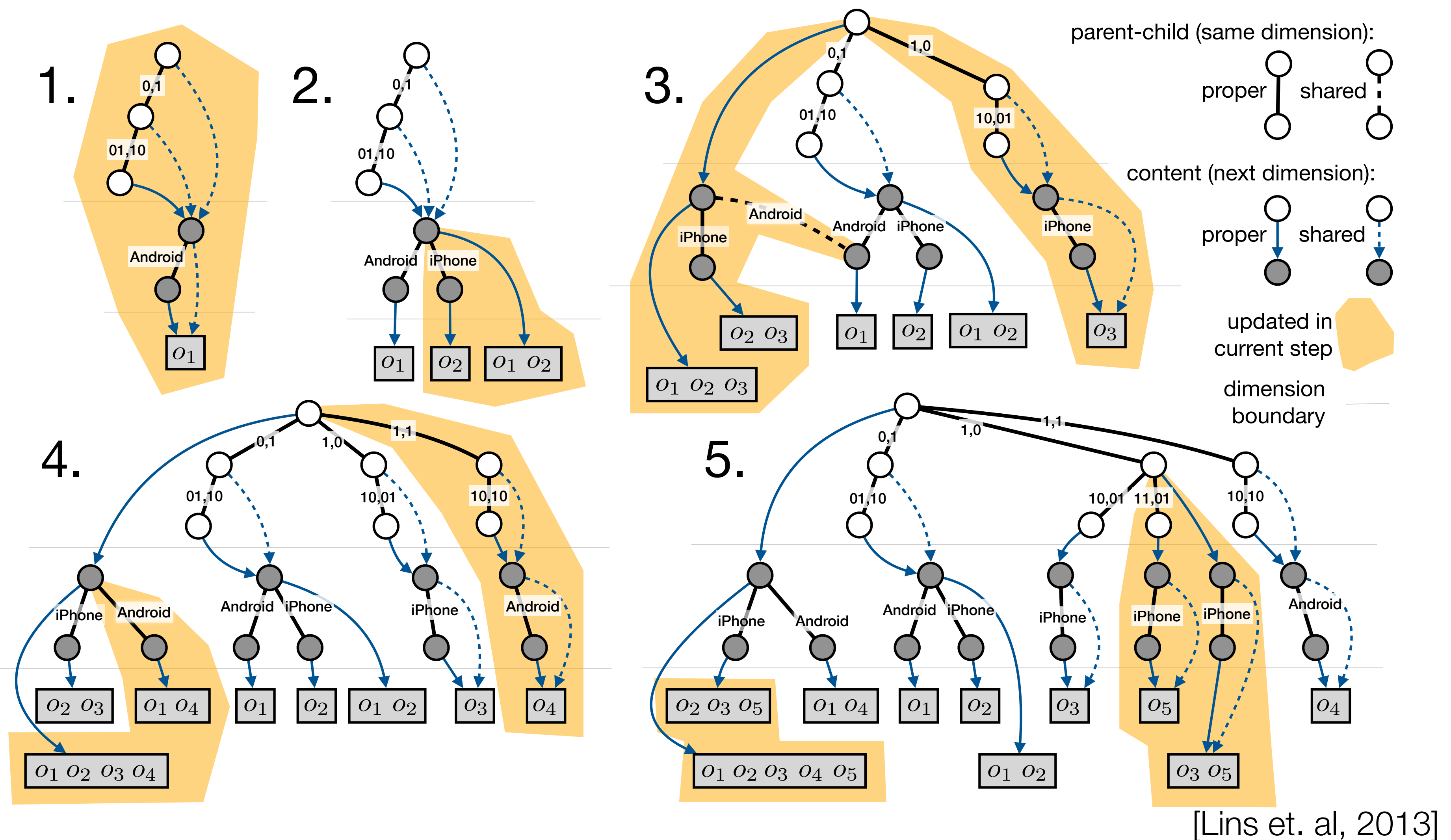


0,1 	1,1 
0,0 	1,0 

00,11	01,11	10,11	11,11
00,10	01,10	10,10	11,10
00,01	01,01	10,01	11,01
00,00	01,00	10,00	11,00

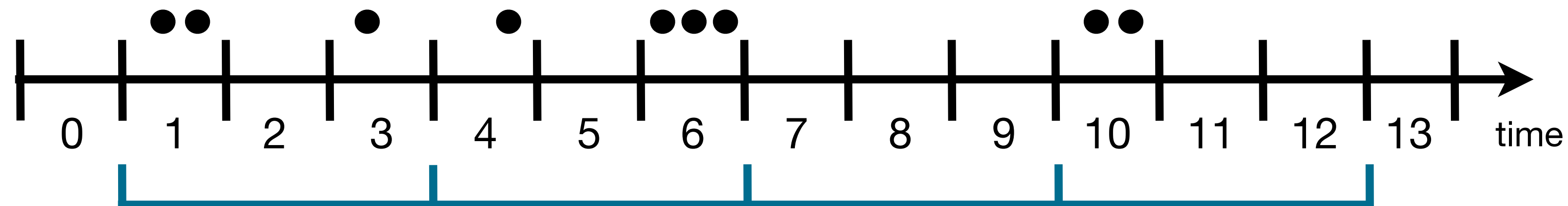
## Indexing Schema

$$S = [[\ell_{\text{spatial1}}, \ell_{\text{spatial2}}], [\ell_{\text{device}}]]$$



# Summed-area Table

- Every node in the previous figure stores an array of timestamped counts like this:



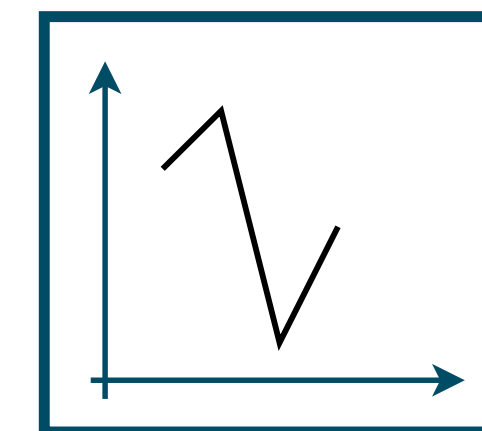
query/tseries/1/3/4

start at bin 1, use buckets of 3 bins each, and collect 4 of these buckets

solve using...

bin: 1 accum: 2	bin: 3 accum: 3	bin: 4 accum: 4	bin: 6 accum: 7	bin: 10 accum: 9
--------------------	--------------------	--------------------	--------------------	---------------------

result



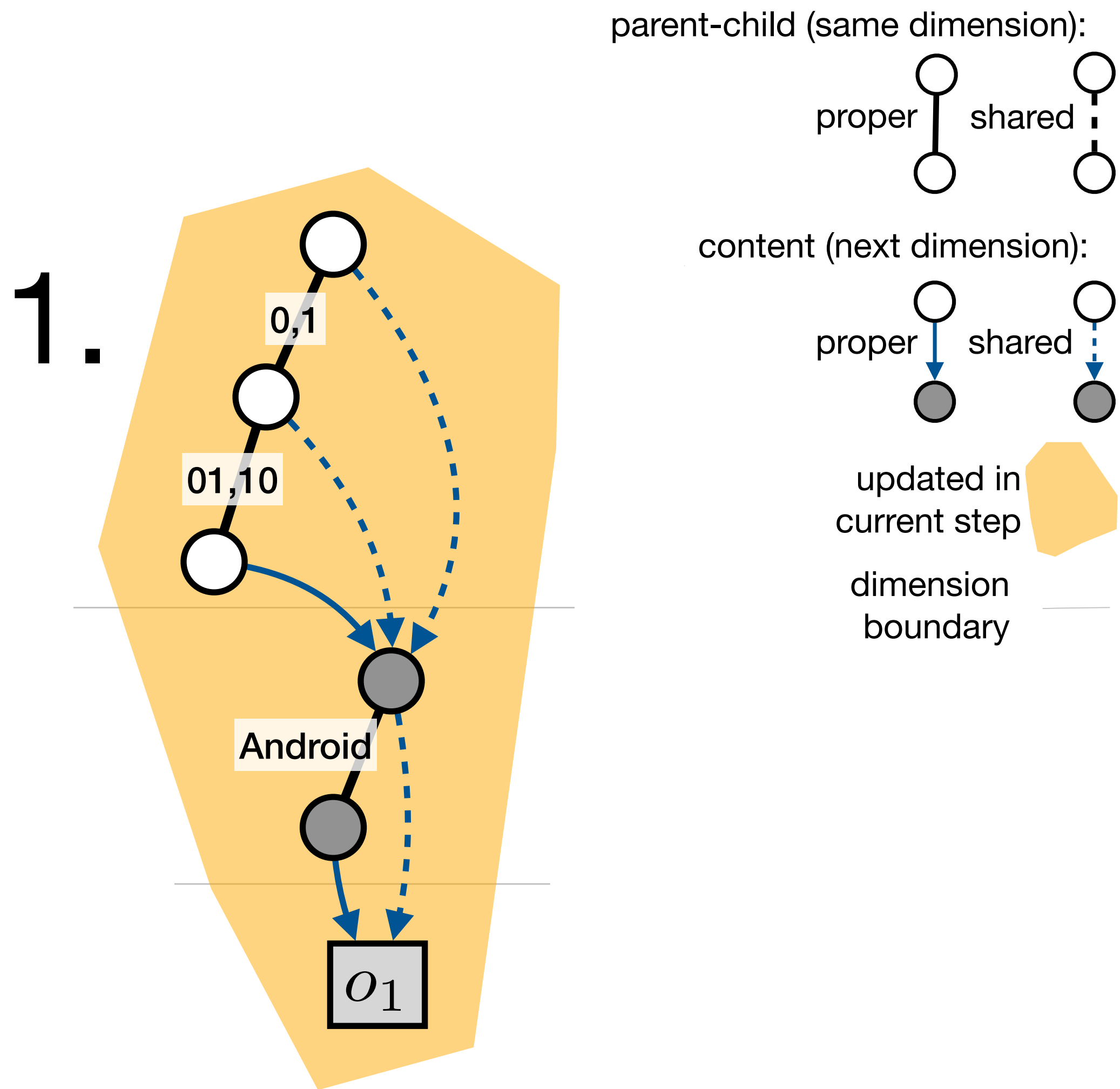
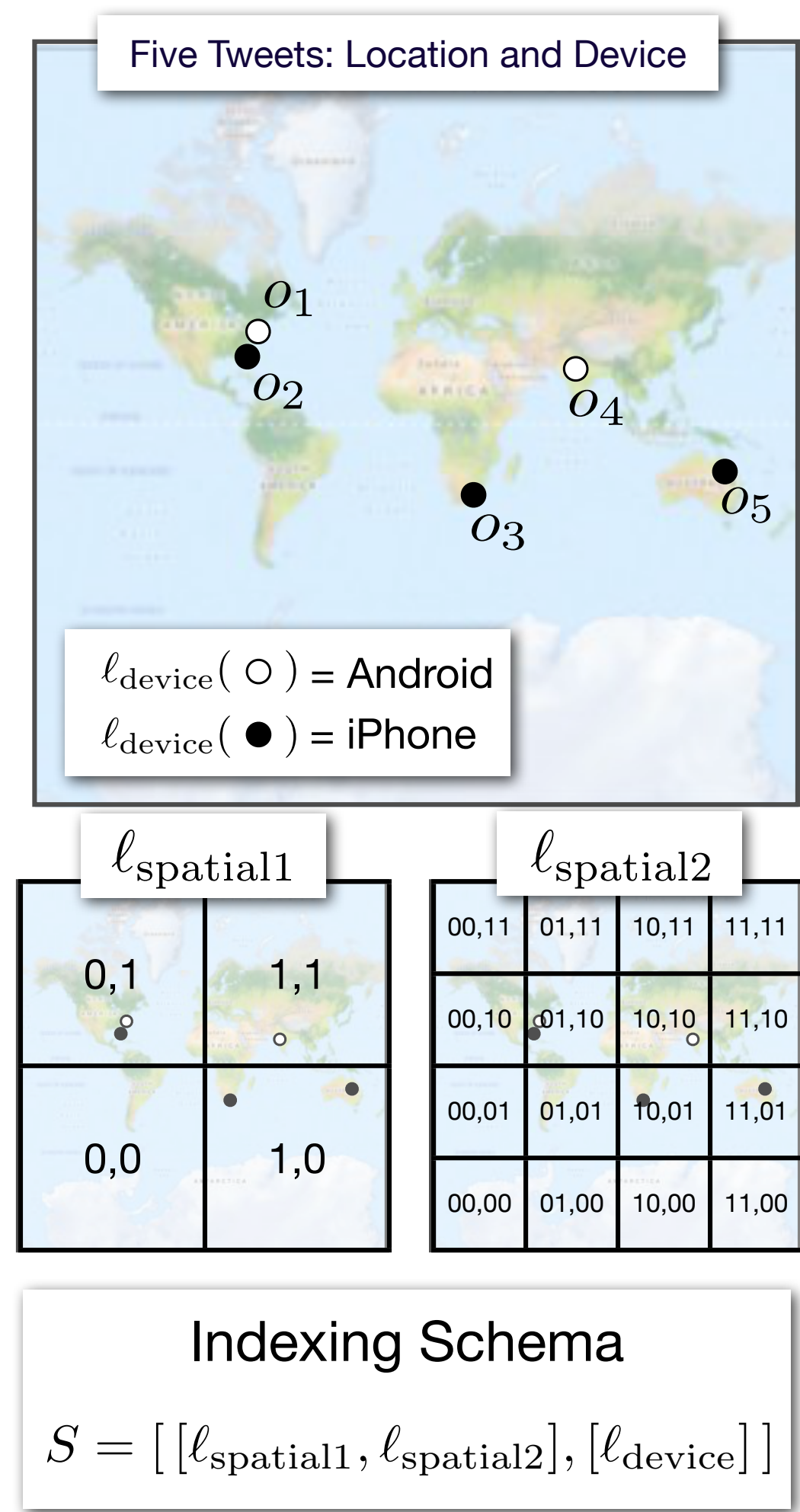
A Summed Table Sparse Representation for Counts

3	4	0	2
---	---	---	---

[Lins et. al, 2013]

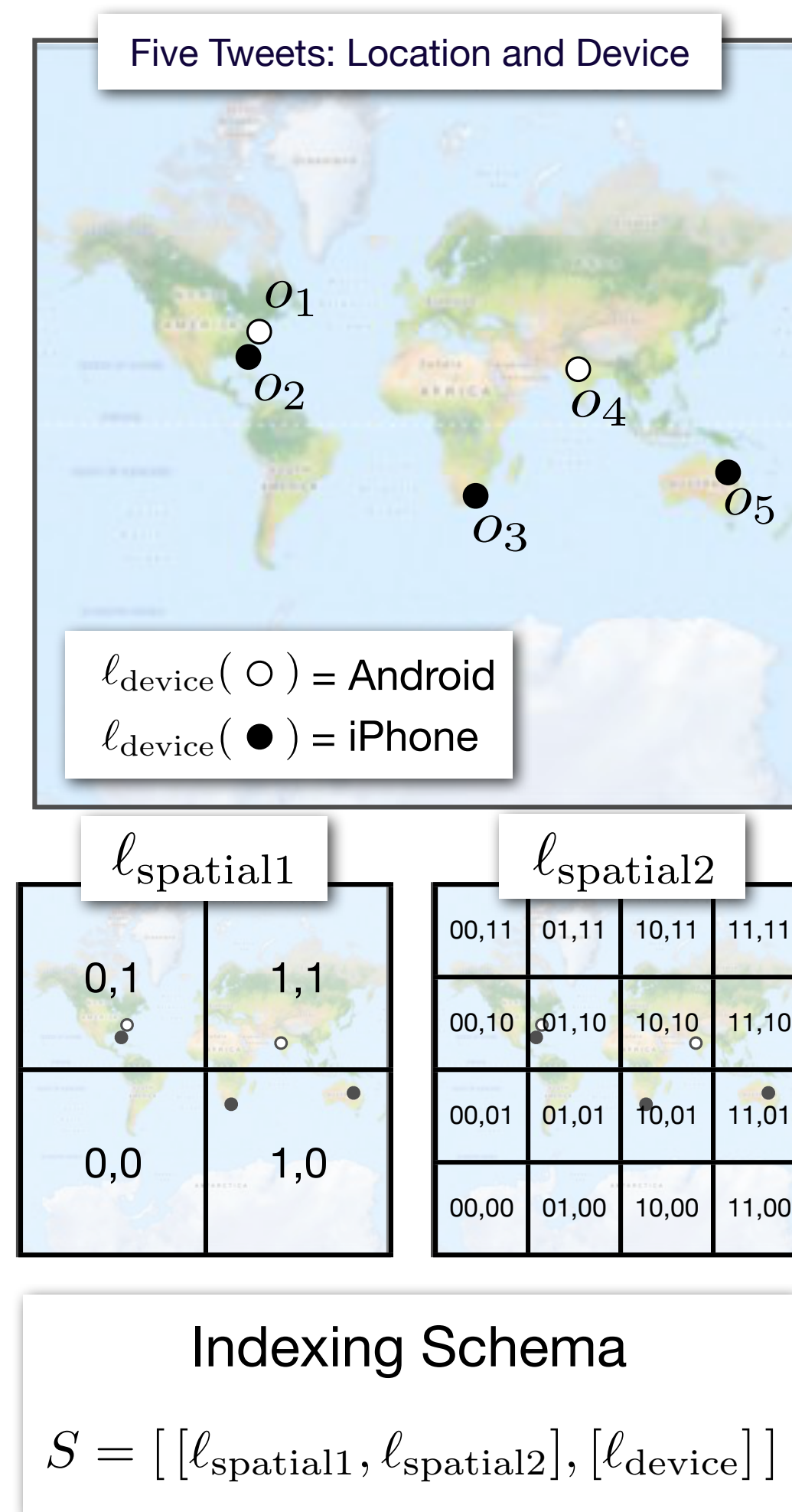


# Building a Nanocube: Step 1

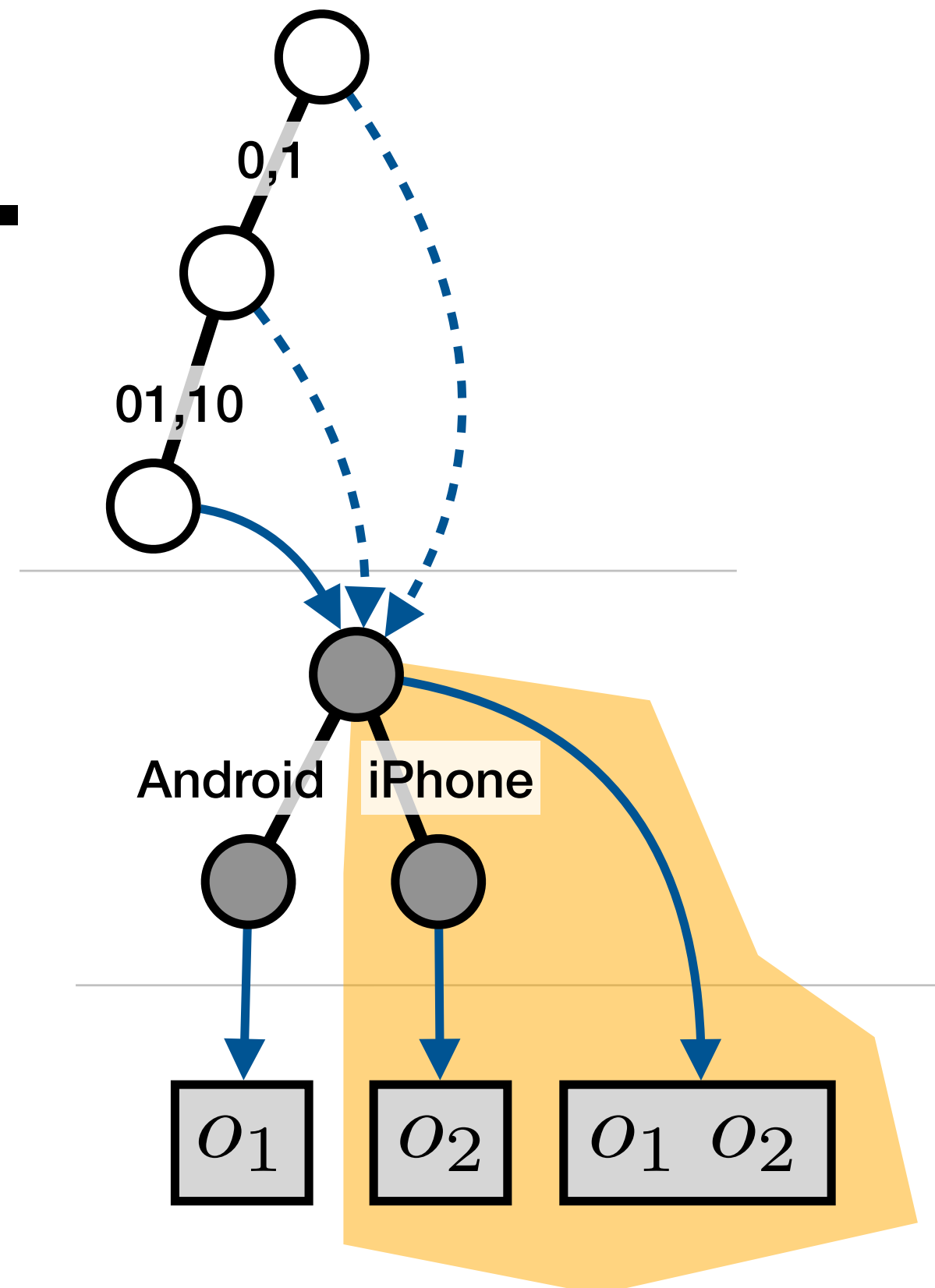


[Lins et. al, 2013]

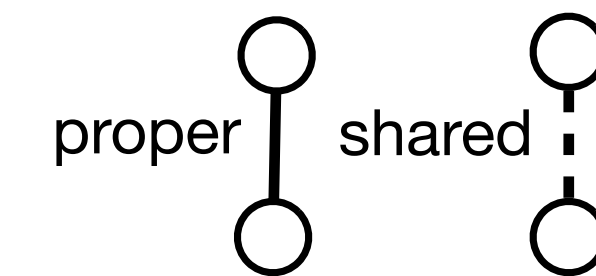
# Building a Nanocube: Step 2



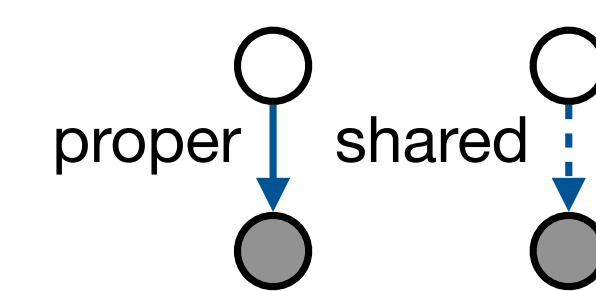
2.



parent-child (same dimension):



content (next dimension):



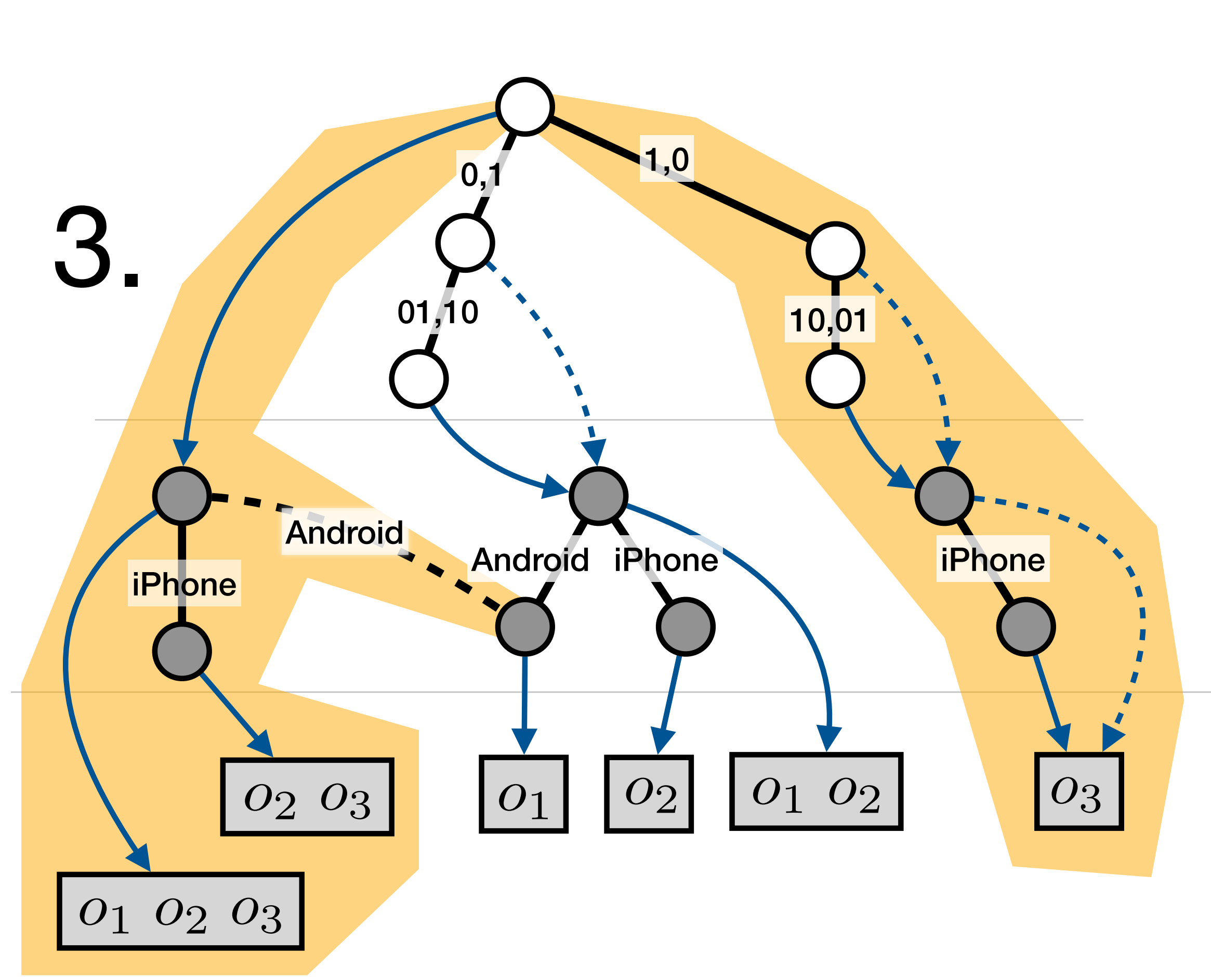
updated in  
current step

dimension  
boundary

[Lins et. al, 2013]



# Building a Nanocube: Step 3



parent-child (same dimension):

content (next dimension):

updated in current step

dimension boundary

Five Tweets: Location and Device

$\ell_{\text{device}}(\bigcirc) = \text{Android}$   
 $\ell_{\text{device}}(\bullet) = \text{iPhone}$

$\ell_{\text{spatial1}}$

0,1	1,1
0,0	1,0

$\ell_{\text{spatial2}}$

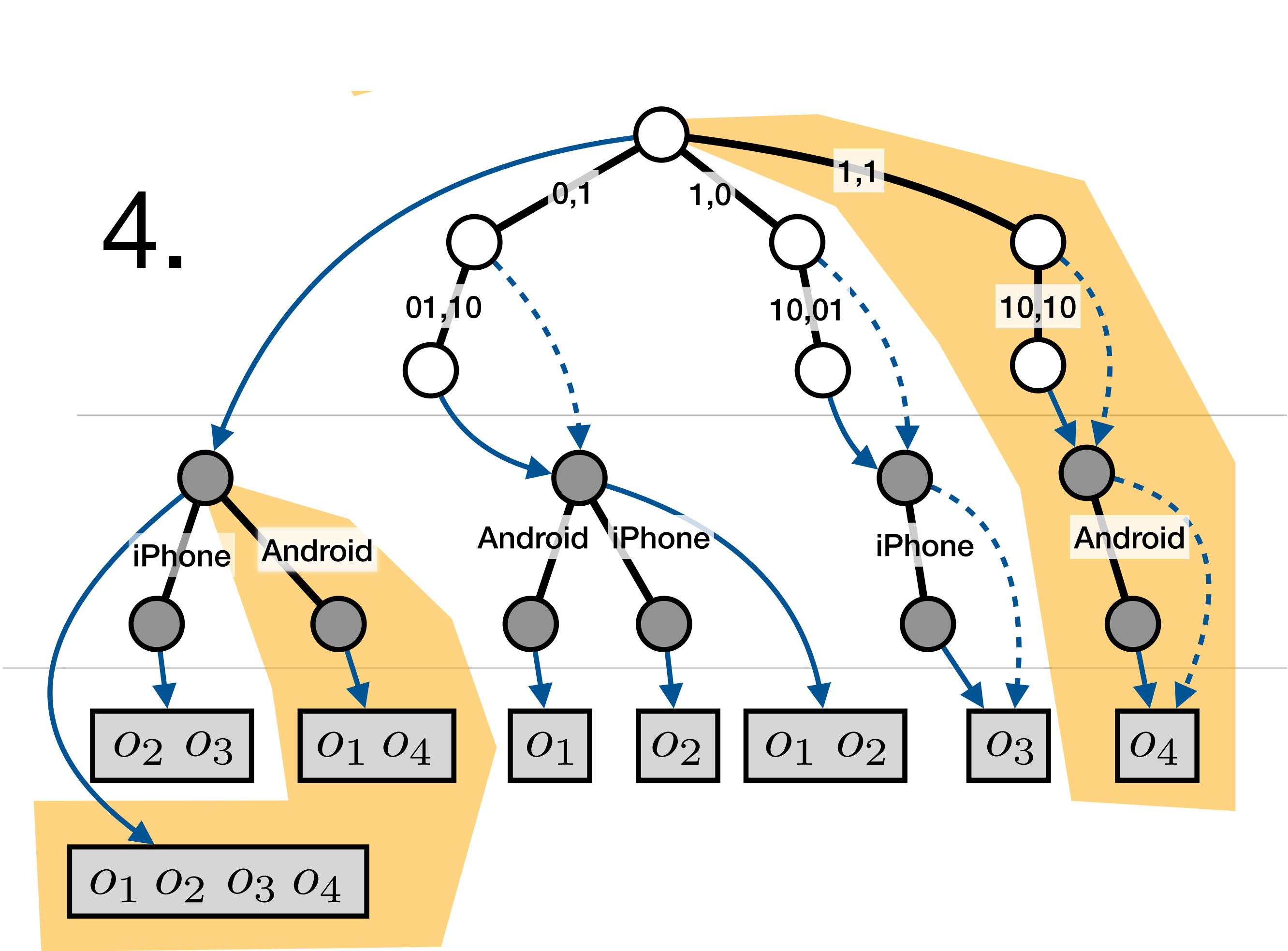
00,11	01,11	10,11	11,11
00,10	01,10	10,10	11,10
00,01	01,01	10,01	11,01
00,00	01,00	10,00	11,00

Indexing Schema

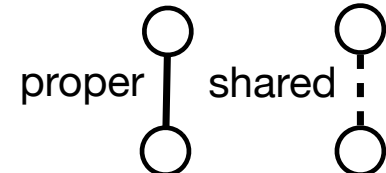
$S = [[\ell_{\text{spatial1}}, \ell_{\text{spatial2}}], [\ell_{\text{device}}]]$

[Lins et. al, 2013]

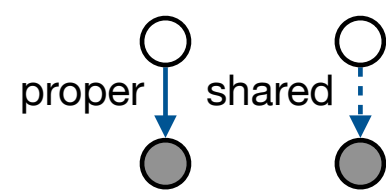
# Building a Nanocube: Step 4



parent-child (same dimension):

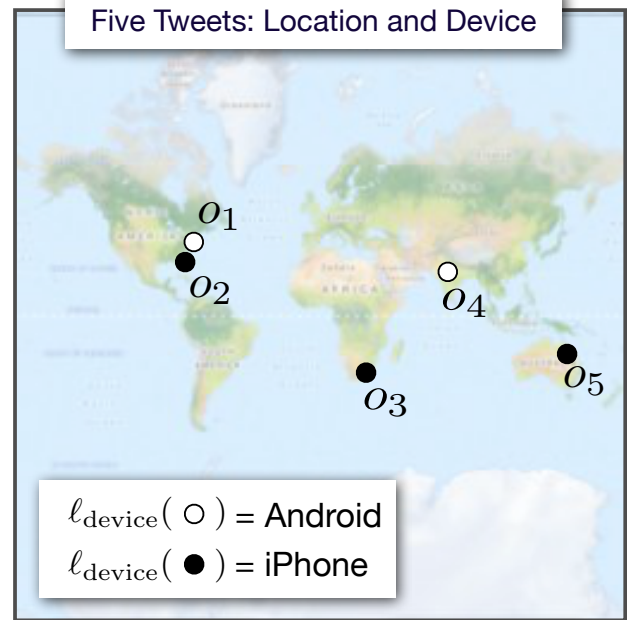


content (next dimension):



updated in current step

dimension boundary



$\ell_{\text{spatial1}}$		$\ell_{\text{spatial2}}$			
0,1	1,1	00,11	01,11	10,11	11,11
0,10	01,10	00,10	01,10	10,10	11,10
0,01	01,01	00,01	01,01	10,01	11,01
0,00	01,00	00,00	01,00	10,00	11,00

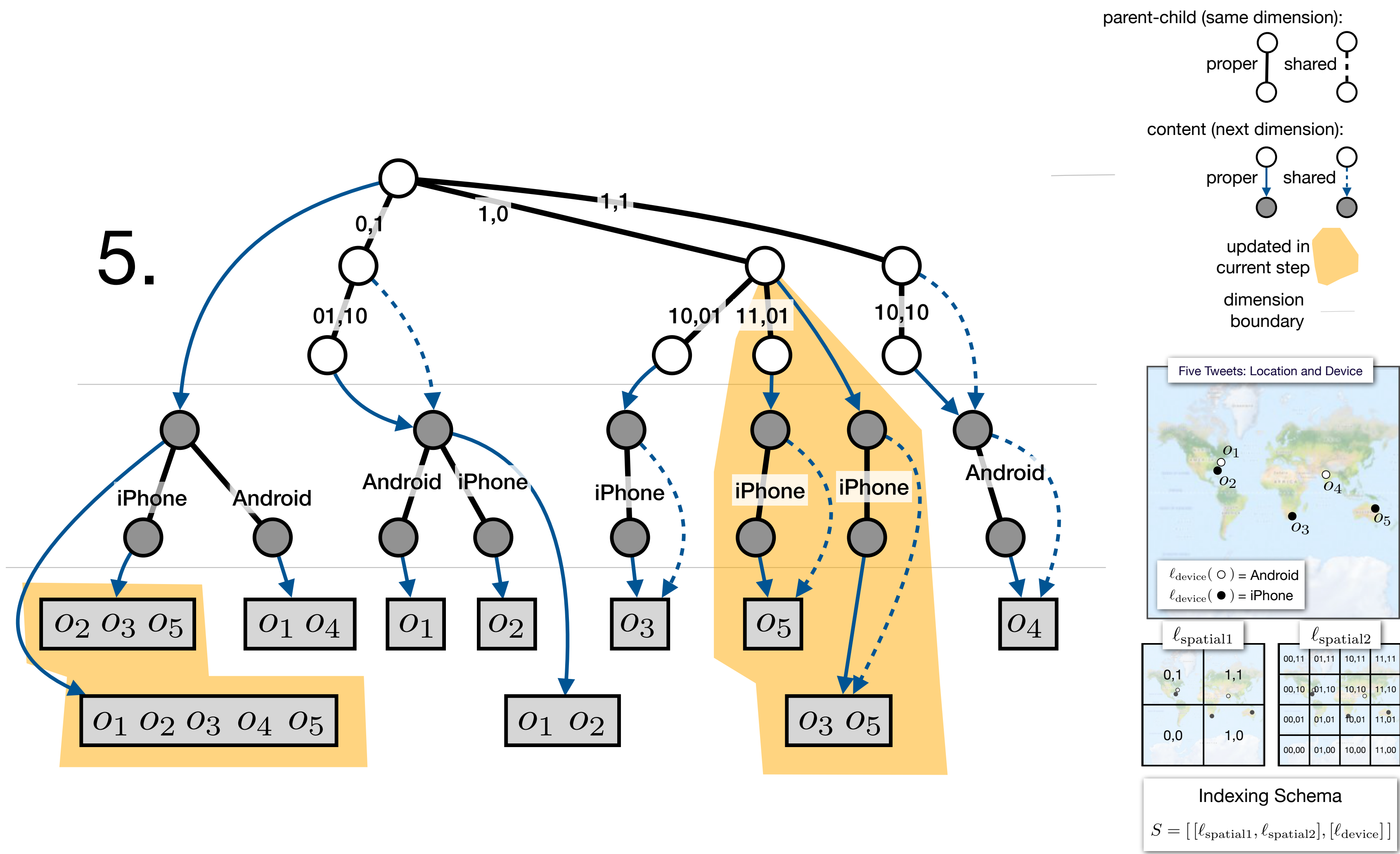
Indexing Schema

$$S = [[\ell_{\text{spatial1}}, \ell_{\text{spatial2}}], [\ell_{\text{device}}]]$$

[Lins et. al, 2013]



# Building a Nanocube: Step 5



[Lins et. al, 2013]

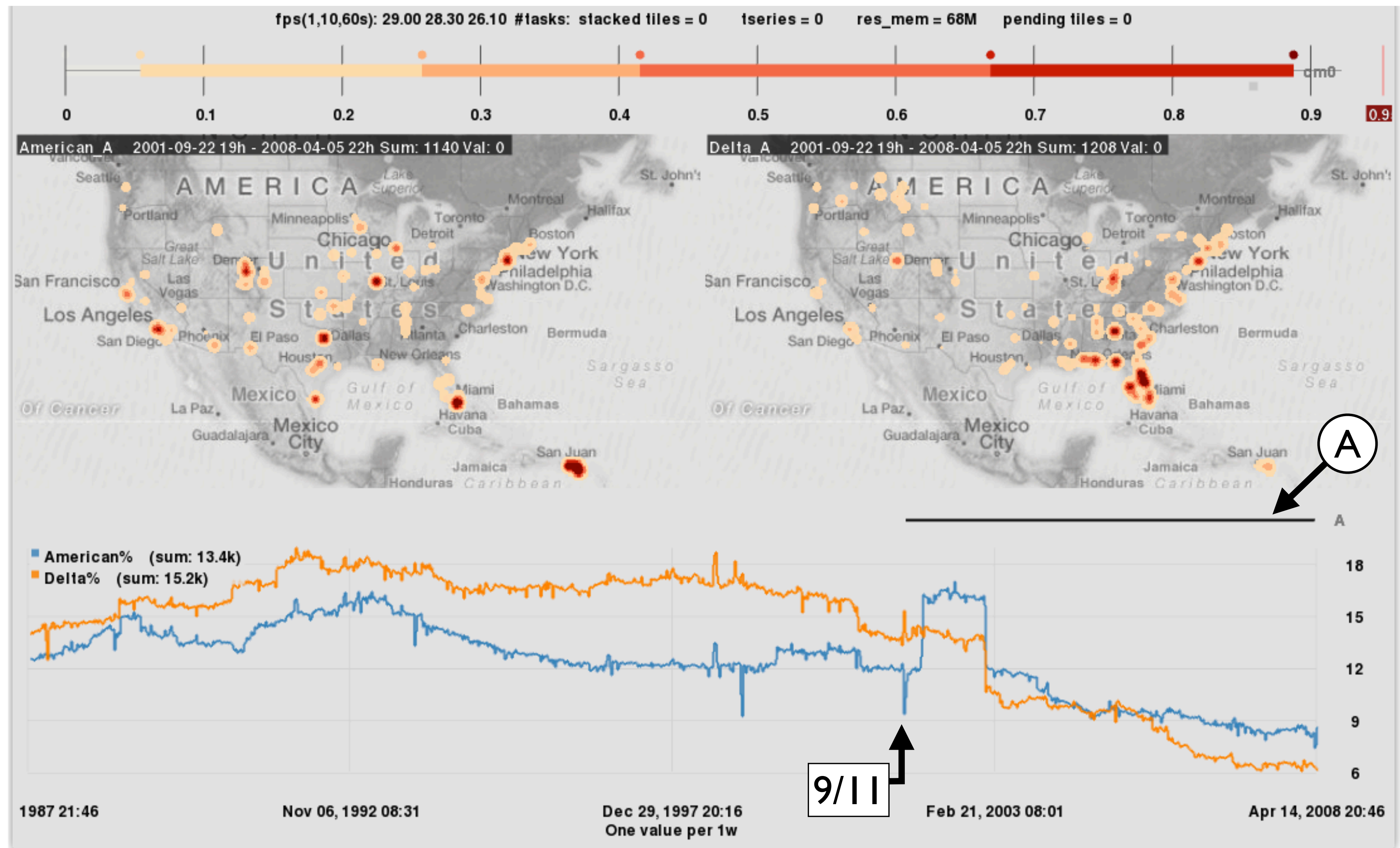
# Nanocubes Discussion

---

- Save space by organizing the data in a manner that takes advantage of data sparseness
- Limited to one spatial dimension, one temporal dimension
- Precompute **once**, then exploration has **low latency**



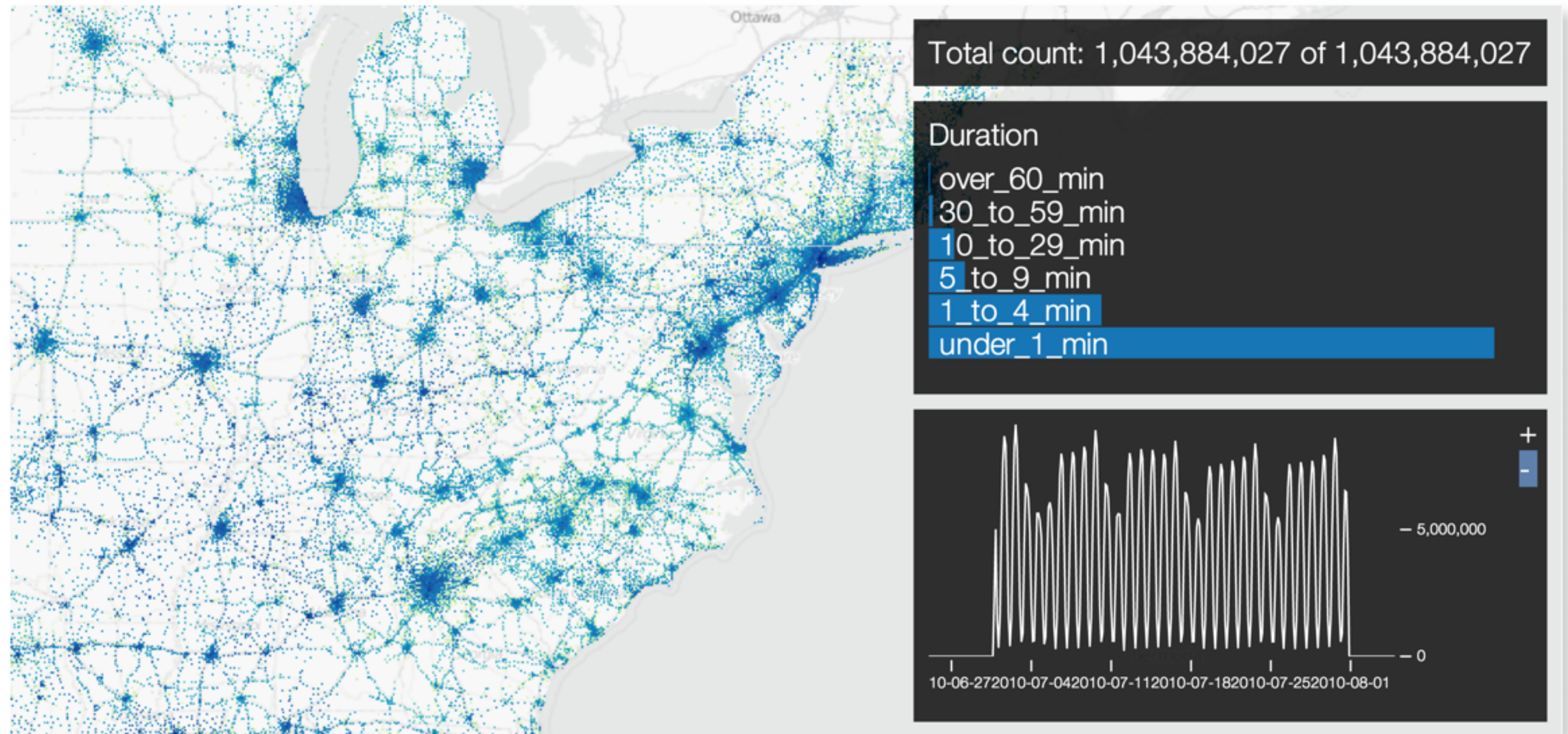
# Example: American vs. Delta



[Lins et. al, 2013]



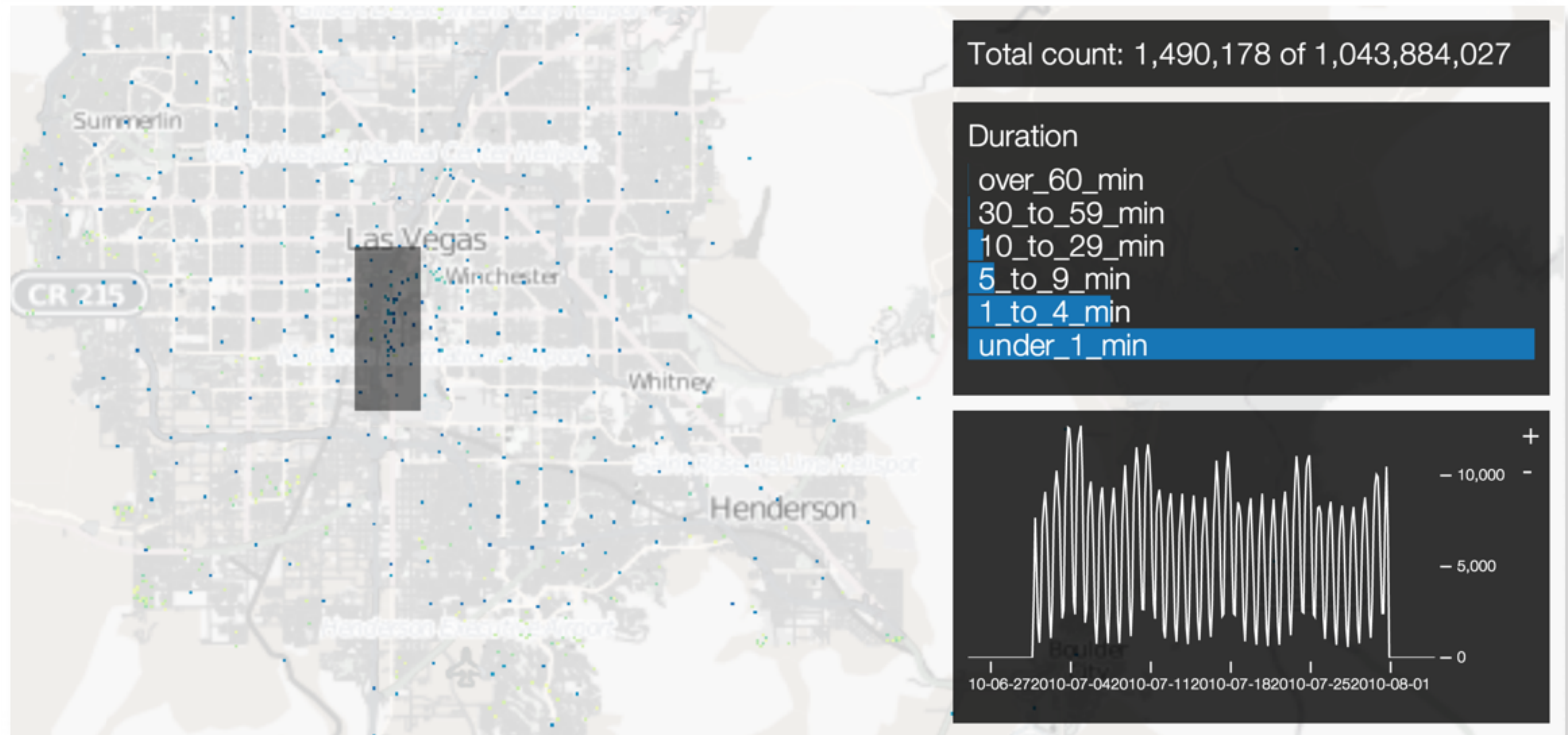
# Example: Cell Data Records



[Lins et. al, 2013]



# Example: Cell Data Records



[Lins et. al, 2013]

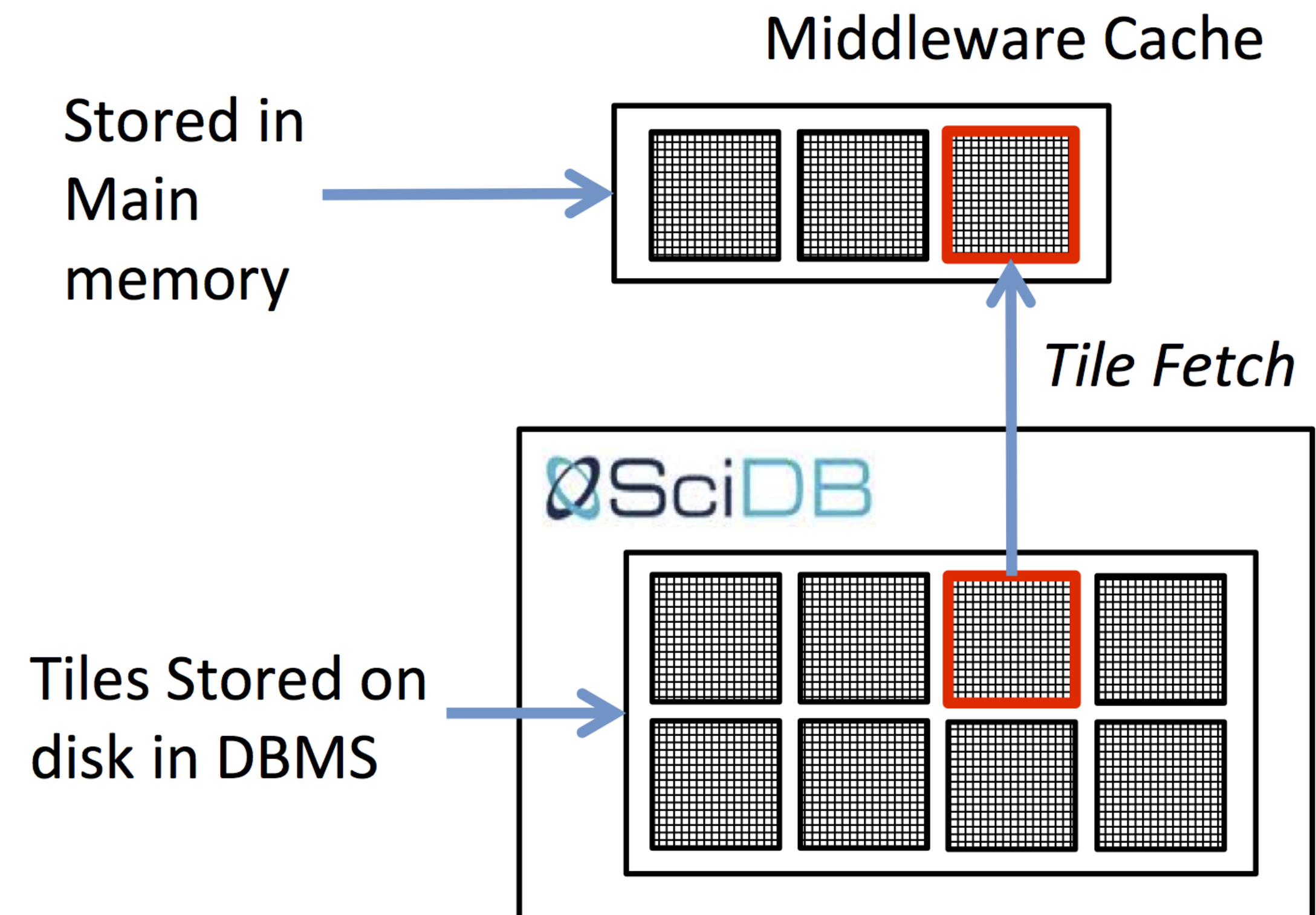
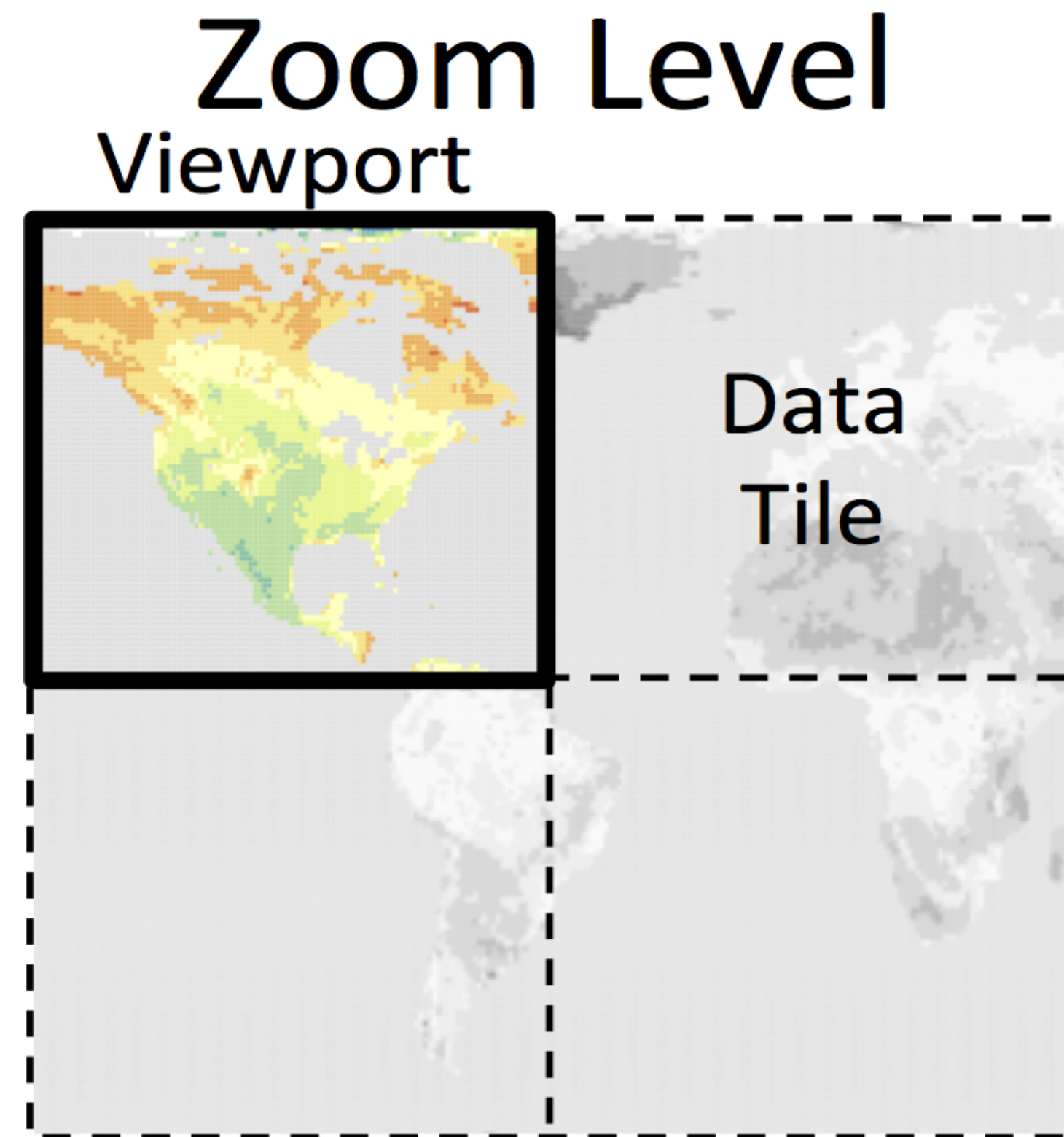
# ForeCache

---

L. Battle



# Tile Storage in ForeCache

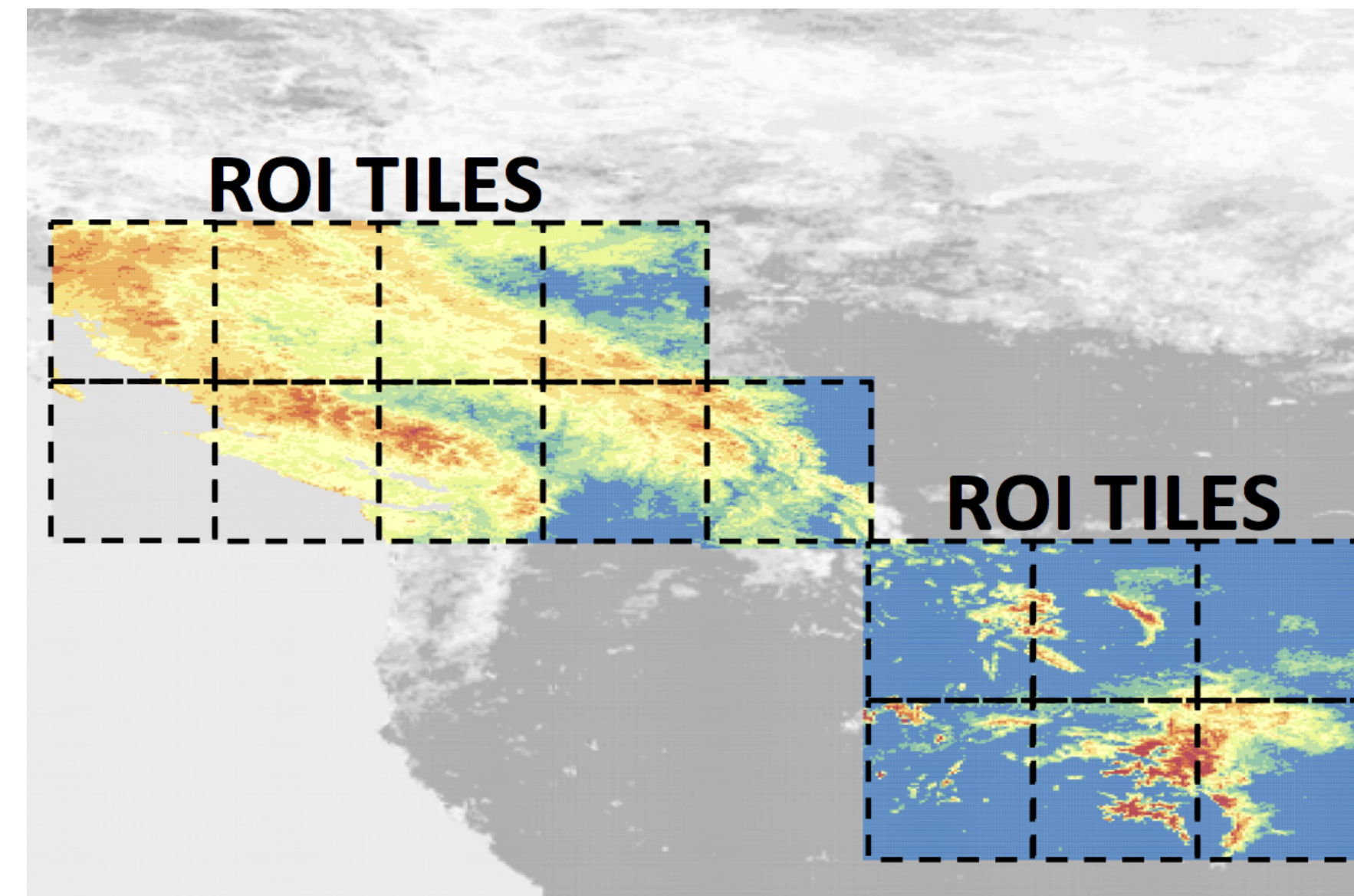
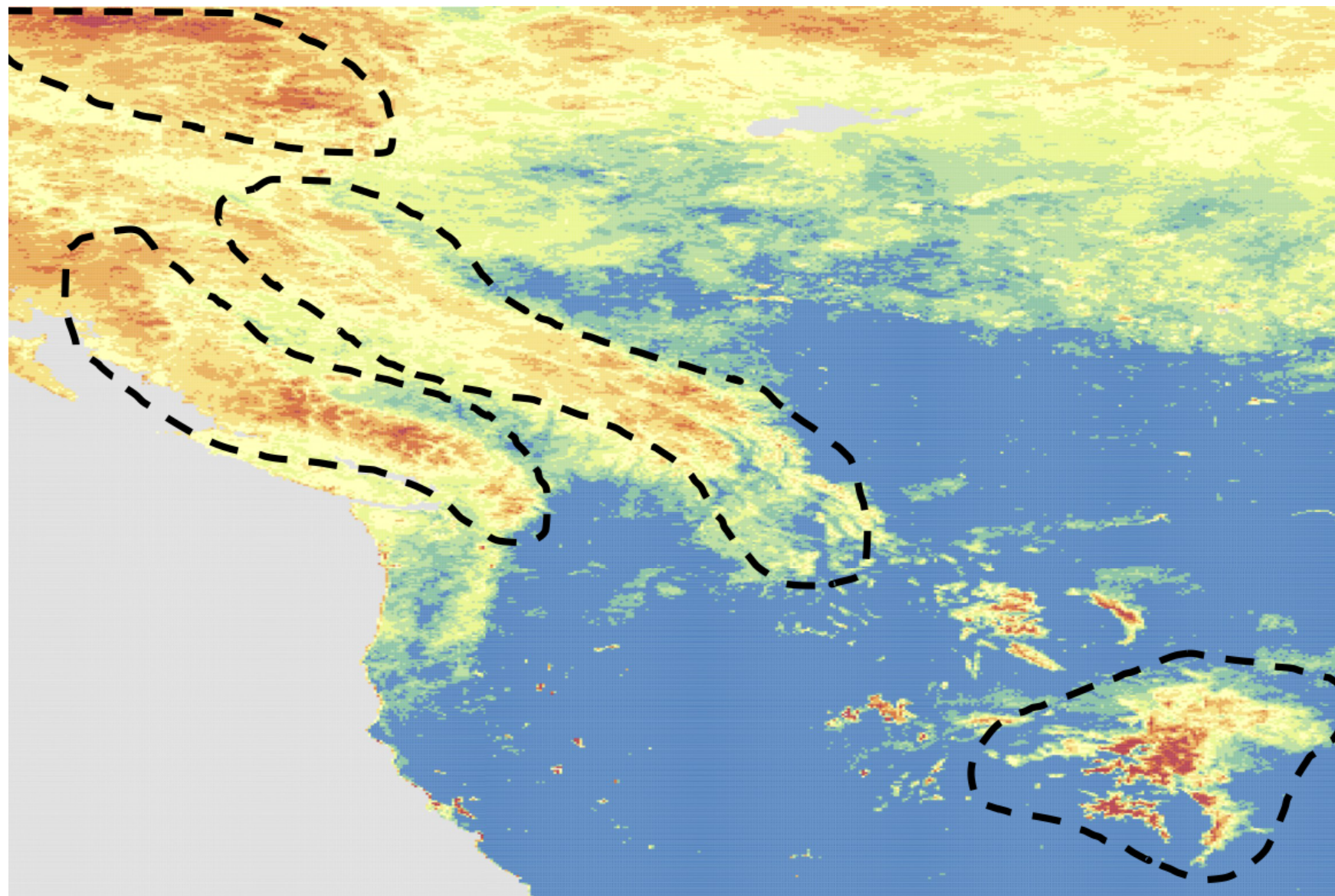


[Battle et al., 2016]



# ForeCache

- Predict which tiles a user will need next and prefetch those
  - Use common patterns (zoom, pan)
  - Use regions of interest (ROIs)



[Battle et al., 2016]