+

CSIT6000P Spatial and Multimedia Databases

2022 Spring

Managing Spatiotemporal Data

Prof Xiaofang Zhou

香港科技大學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

+ Learning Objectives

- What we will cover
  - Spatiotemporal data and queries
  - Spatiotemporal indexing and query processing
  - Trajectory similarity measures
  - Open issues and directions
- Goals
  - Understand the temporal dimension of spatial data
  - Understand basics of spatiotemporal data management
  - Learn from examples about how to deal with new data management challenges brought by new data types

+ The Temporal Dimension

- Location and time are two ubiquitous attributes of data
- RDBMS is designed to handle neither of them
- We have studied how the spatial dimension can be managed so far
- The temporal dimension of spatial data cannot be simply viewed as just another dimension
  - Data values and operations/queries are quite different

+ Spatiotemporal Data

- GPS recordings
- Sensor data, RFID data and surveillance data
- Use of smartphones and smartcards
- Use of location-aware apps
- Social media data: uploaded photos/messages and check-in data
- Much more spatiotemporal data to come with 5G, Smart City and IoT
- …and beyond the geographical domain

+ What is Trajectory Data

- Any data that record the locations of a moving object over time in a geographical space
- Simple form: $\langle id, (p_1,t_1), (p_2,t_2) \ldots (p_n,t_n)\rangle$

  ordered by time: $t_1 < t_2 < \ldots < t_n$
- General form:

  $\langle objId, trajID, trajProperties,$

  $(p_1,t_1,a_1),(p_2,t_2,a_2) \ldots (p_n,t_n,a_n)\rangle$

+ Many Dimensions of Trajectory Data

- Basic dimensions
  - Spatial dimension: locations $p_1, p_2 \ldots p_n$
  - Temporal dimension: time-stamps $t_1, t_2 \ldots t_n$
  - Attribute dimension: other data of interest $a_1, a_2 \ldots a_n$
- Other dimensions
  - Entity dimension: what type of objects?
  - Environment dimension: road networks, floor plans, water systems, sensor networks
  - Semantic dimension: what activities at a location or time?

+ How Much Trajectory Data?

- A back-of-the-envelope calculation:
  - A simple point data (`x`, `y`, `t`): 24 bytes
  - A car can generate 85KB a day (10 hours a day, 10 seconds interval)
  - Beijing has 60,000 taxis, that is 5GB a day, or 1.72 TB a year
- Actual trajectory data could be much larger
  - A multiplier of X: There are much more information than just a point data: taxi ID, trip ID, job status, direction, velocity, acceleration, fuel consumption, other sensor data (OBD/M2M data)
  - Even larger once processed: original data, plus map-matched data, other derived data, other forms of representation (e.g., OpenLR - http://www.openlr.org/)

+ Trajectory Data in a Company

- A car navigation service provider
- Total trajectory data: 32 TB in size, 10.9 billion matched trajectories

	Current	Daily
Company X (in-car navigation provider)	17.6TB	15M trajectories
Company Y (map app provider)	14.5TB	5M trajectories
Company Z (social network)	0.68TB	18M trajectories

- Every day, ~40M new trajectories, ~4 billion points
- Many types of related data: maps, accident reports, roadside data, surveillance video, weather, events, social media…

+ NavInfo DataHIVE (minedata.cn)

Vehicle	Infrastructure	Environment	People
Trajectories:	Standard maps	Weather	Voice and text
- taxis	High res maps	Events	User comments
- uber-like	Services POIs	Air quality	Search log
- monitored	Culture POIs	Water quality	Travel log
- commercial	Commercial POIs	Land & water info	Operators' OD
- user generated	Health POIs	DEM & EEC	Workplace info
Sensor/OBD data	Travel POIs	Satellite image	
Perception data	City models	Street views	
	City 3D Models	Roadside pictures	
	Business districts	Laser point cloud	
	Admin boundaries	Road condition	
	Organization maps	Traffic condition	
		Traffic incidents	

+ A Lot of Data!

		Total	Per Period
Vehicle Dynamics	Track (GPS and others)	1682 T	2010 G/day
	Sensor (OBD, cameras etc)	39 T	123 G/day
Environment Status	Weather and air/water quality	7 T	32 G/day
	Physiognomy	135 T	528 G/day
	Traffic	230 T	237 G/day
Infrastructures	Road	2236 T	62 G/mth
	POI		10 G/mth
	Building and admin boundary		20 G/quarter
People Information	Profile and behavior	488 T	310 G/day

+ Applications

- Understanding, monitoring, predicting mobility patterns
  - …from very large amount of movement data in real-time
- Some examples
  - Finding the nearest businesses or services
  - Location/movement-based event/service recommendations, alerts (sales, traffic jam, warnings…) and information push (e.g., advertisement)
  - Resource tracking and scheduling (e.g., for fleet management)
  - Safe drivers (e.g., for insurance industry)
  - Data-driven ITS, urban planning and smart cities
  - …

+ Alternative Names

- Moving objects data concern the current and future locations
- Spatial trajectory data is the movement history of moving objects
- A trajectory without the time dimension is also called a route

+ Spatiotemporal Queries

- Simple point/range queries are useful
  - Spatiotemporal point/window queries: a combination of timestamp/time interval and spatial point/region
  - Examples:
    - To find where an object is at time *t*
    - To find all the objects at/near location *p* at time *t*
    - To find all objects inside a given region during a given period…
- Query can also take a trajectory as an input
  - To find the nearest POI for a given trajectory
  - To find top-K most similar trajectories to a given trajectory

+ More Spatiotemporal Queries

- More challenging queries
  - Monitoring queries: also called continuous queries
  - Predicative queries: at a future point of time
- Data analytics queries
  - To find trajectory clusters
  - To find where traffic jams could occur in the next 30 minutes
  - To find where people come from to a given region

…can you give some examples for each type of queries?

+ Example: Continuous NN Query

- Where is the nearest petrol station?
- Static NN query
  - Concerning a given location
  - Optimization goal: minimize the number of objects to be examined
- Moving NN query
  - Concerning the current location which is moving
  - Optimization goal: minimize the number of calls to the NN algorithm
    - Safe region: the results don't change within the region

how?

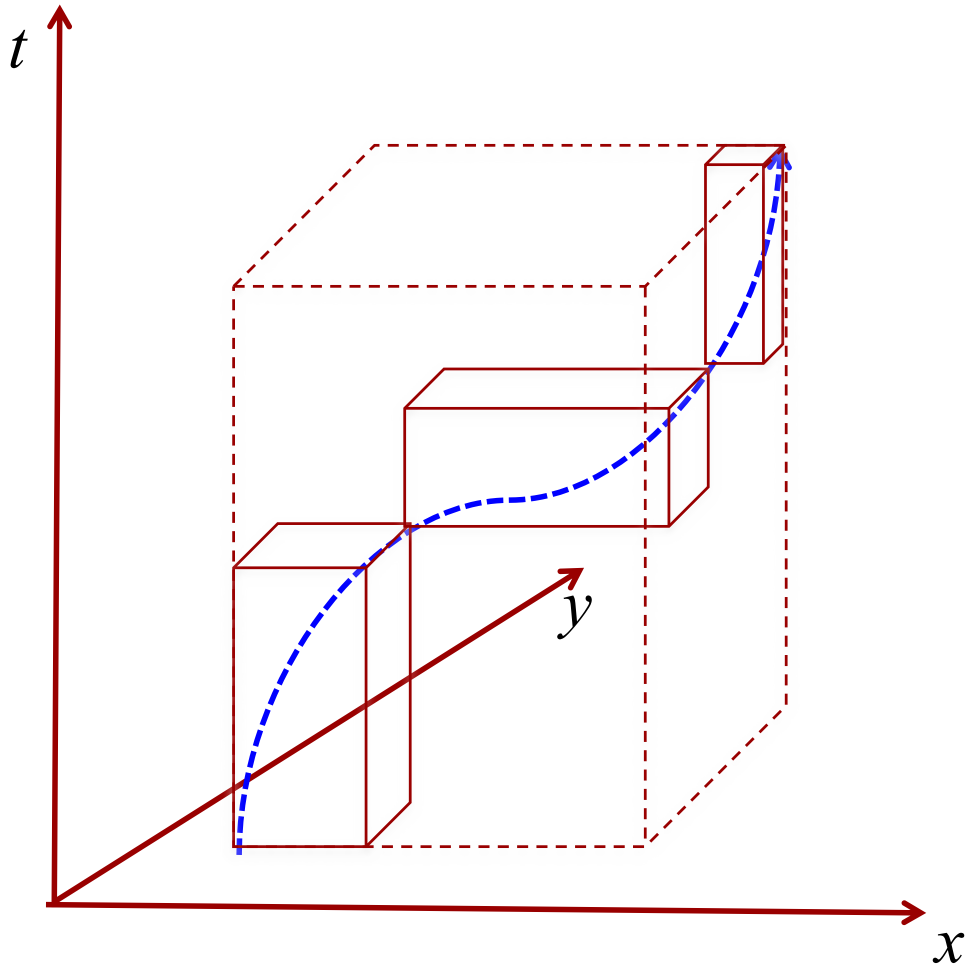
…path NN query: where is the nearest petrol station on my way from city to Surfer's Paradise?

+ Indexing Spatiotemporal Data

- R-tree has been the most efficient and widely used general purpose spatial indexing structure, so let's extend that for spatiotemporal data
- What's special now?
  - The time dimension increases monotonically and unbounded
  - An object can have a long sequence of locations (i.e., points), and those locations which are temporally close to each other are often spatially close too
  - MBR has been used as the foundation for R-tress, but now the MBR for an object or a group of objects either changes over time or occupies a huge space

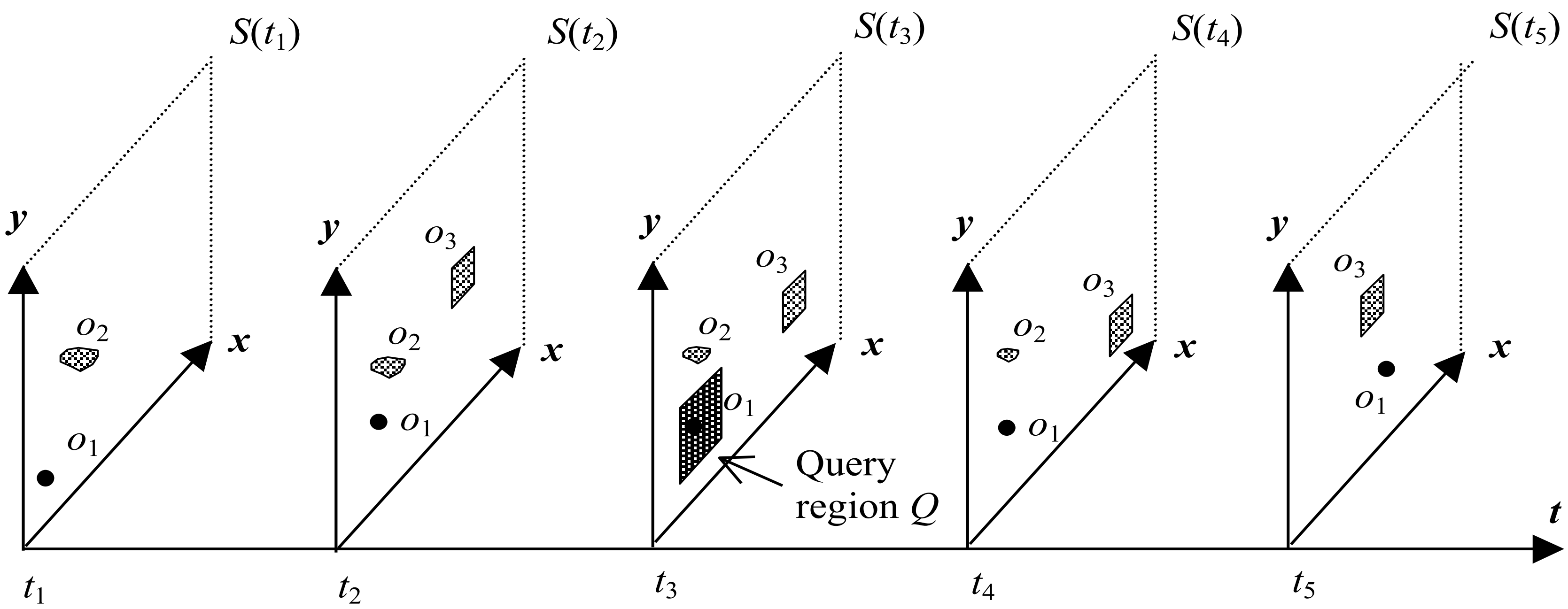
+ 3D R-tree?

- Adding time as another dimension
  - Conceptually simple: 2D for the spatial dimension + 1D for the temporal dimension
  - Problem 1: The t-dimension is unbounded, and the data about one trajectory is spread everywhere and severe overlapping in the time dimension
  - Problem 2: It's not effective to use boxes to approximate lines
  - Problem 3: Only efficient for coordinate-based queries (time slices and ranges), not for trajectory-based queries



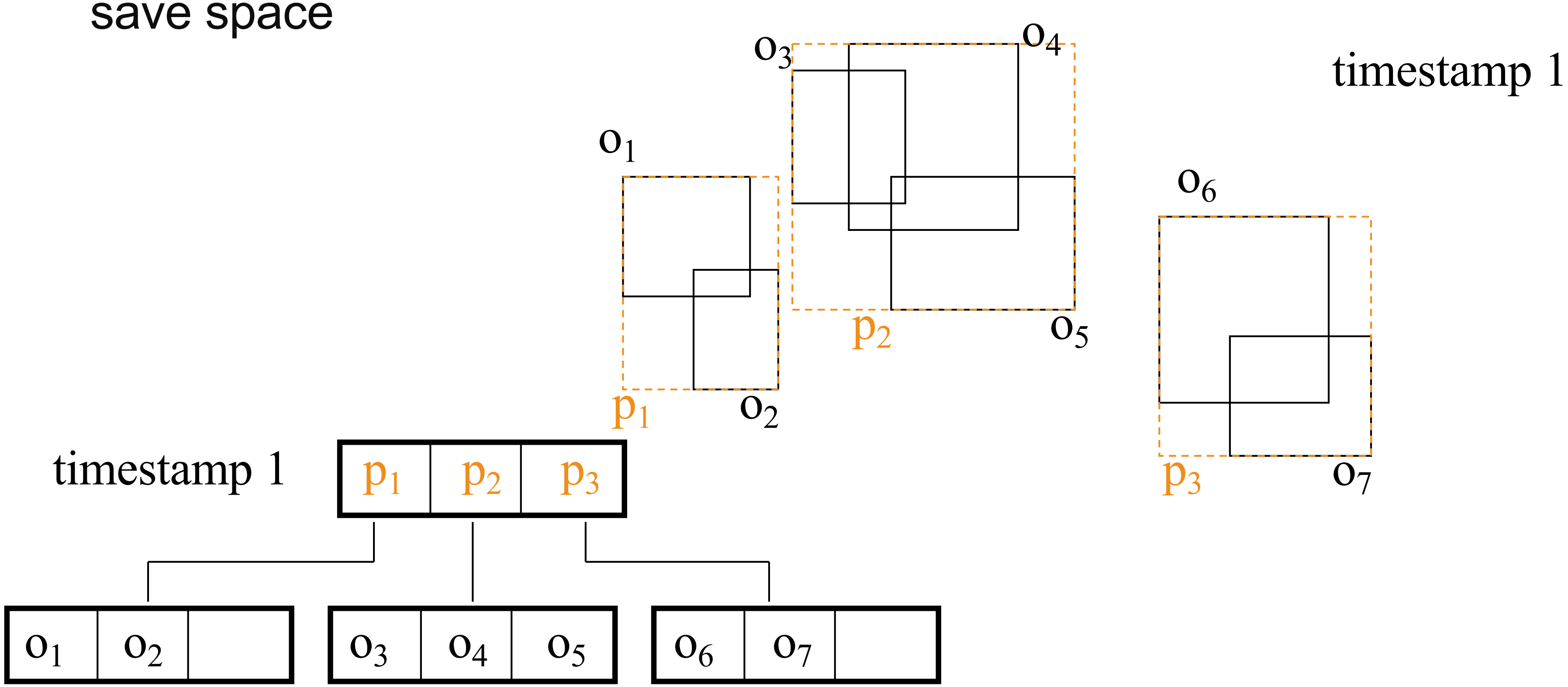
Snapshot-Based Indexing

- One R-tree for each snapshot?



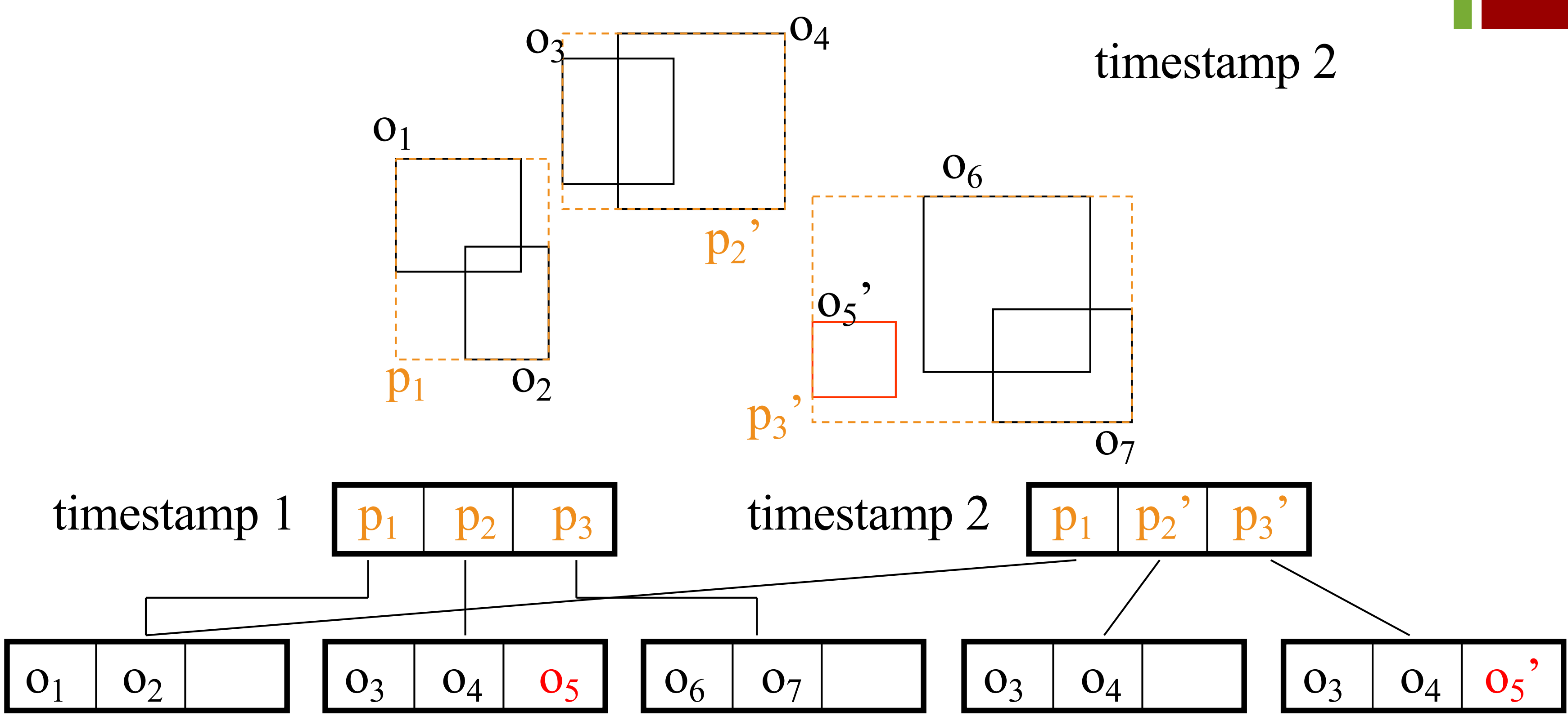
+ HR-Tree

- Historical R-tree
  - An R-tree is maintained for each timestamp in history
  - Trees at consecutive timestamps may share branches to save space



"Towards Historical R-trees", M. Nascimento and J. Silva, ACM Symposium on Applied Computing, 1998.

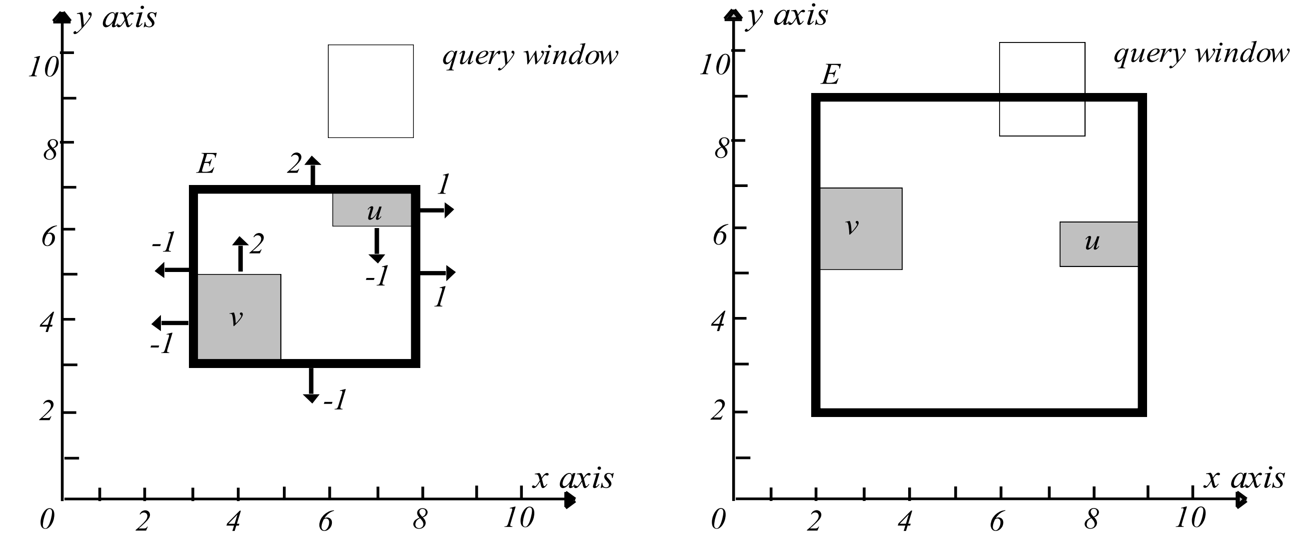
+ HR-Tree: Sharing Branches



…what do you think about this approach?

+ TPR-Tree

- Time-Parameterized R-tree
- Store locations and MBRs as functions of time
  - L(t) = L(t0) + V(t)   MBR(t) = MBR(t0) + V(t)
- VBR: V for velocity
  - An MBR grows with time, and can be estimated
  - Insertion of objects needs to minimize VBR



(a) The boundaries at current time 0 (b) The boundaries at future time 1

"Indexing the Position of Continuously Moving Objects", S Saltenis, C Jensen, S T Leutenegger and M. A. Lopez, SIGMOD 2000

+ Trajectory Data

- Spatial trajectory is object movement history in a space
  - Continuous in nature, but discrete once captured and stored
- Many location-update strategies
  - By time, by distance, by deviation…
  - A trade-off between accuracy and other overheads
  - Variations may not always be under control
- Movement can be in a free space (e.g., birds flying), or a constrained space (e.g., cars in road networks)

+ Trajectory Similarity Measures

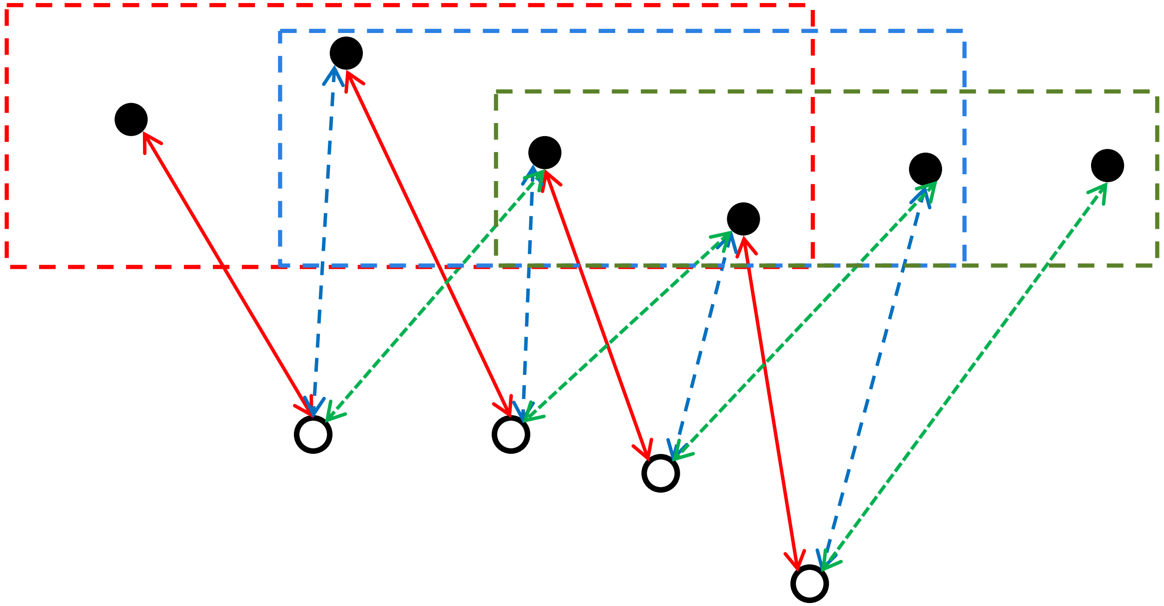
- The foundation to perform trajectory-based queries and analytics (such as clustering)
- Many types of similarities
  - Sequence-based: passing the same sequence of points?
  - Geometry-based: similar shapes?
  - With or without time or speed considerations
- Key factors to consider
  - Alignment of sampling points (to deal with non-uniform sampling)
  - Robustness to noise (to deal with data quality issues)
- There are many trajectory similarity measures

+ Classification

- Alignment of samples
  - Lock-step vs. adaptive alignment
- Similarity metric
  - Geographical distance vs. count based
- Continuity
  - Discrete vs. continuous
- Dimension
  - Spatial-only vs. spatio-temporal
- Underlying space
  - Euclidean space vs. road network

+ Lock-Step Alignment

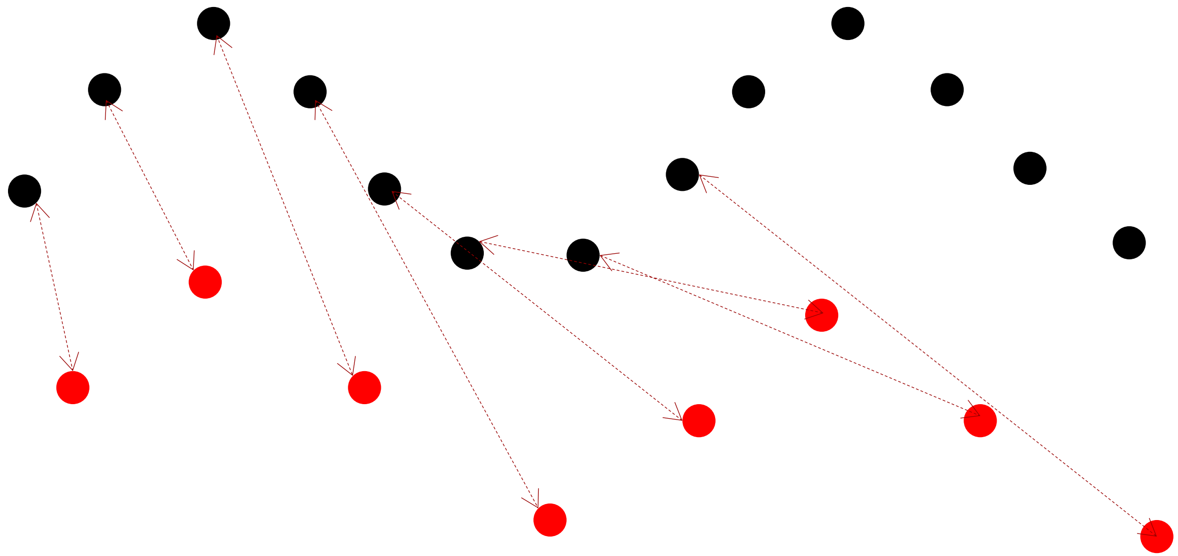
- The *k*-th point of a trajectory is aligned to the *k*-th point of the other trajectory
  - Sliding window based on the shorter trajectory
  - The distance is the best of window-based total Euclidian distance



… what is the time complexity of this approach?

+ Drawback

- Cannot find similar trajectories with different sampling rates, which are common in practice
- Sensitive to noise



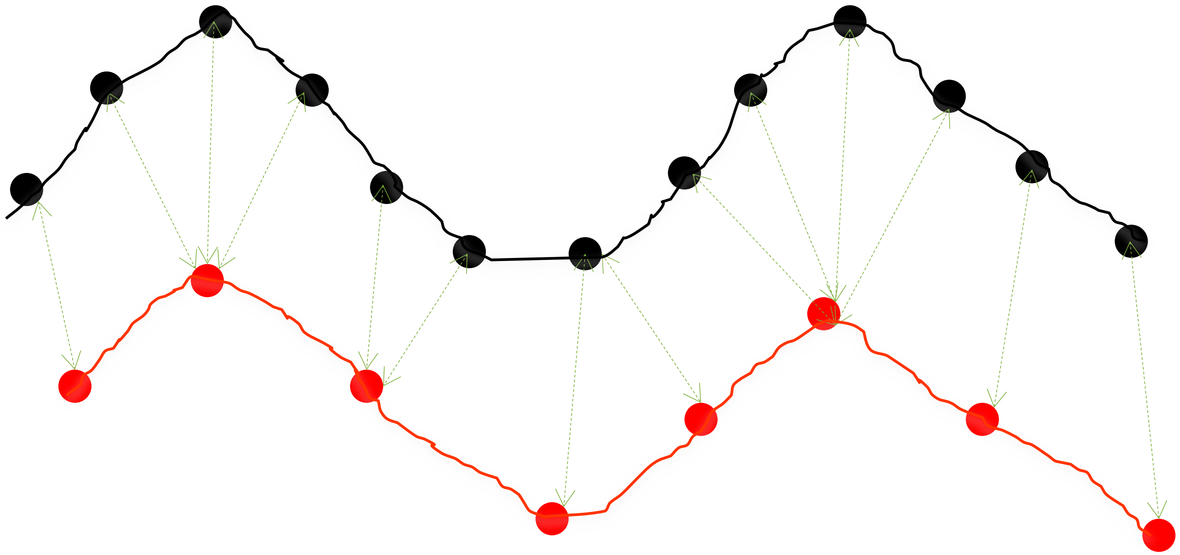
+ Adaptive Alignment

- Dynamic Time Warping (DTW) distance
  - Adaptation from time series distance measure
  - Used to handle time shift and scale in time series
- Optimal order-aware alignment between two sequences
  - Goal: minimize the aggregate distance between matched points
- 1-to-many mapping: one point in one sequence can be mapped to multiple points in another sequence

Yi, Byoung-Kee, Jagadish, HV and Faloutsos, Christos, Efficient retrieval of similar time sequences under time warping. ICDE 1998

+ DTW for Trajectories

- Useful when detecting similar trajectories with different sampling rates



Using Dynamic programming to compute DTW. Check the Wikipedia page
Time complexity: O(MN)

+ Count-Based Similarity

- So far, geographical distance based
  - Similarity is measured by the geographical distance between matched samples
- Count based
  - Similarity is measured by the number of ‘similar’/ ‘dissimilar’ samples
  - Based on Edit Distance
    - The distance between two strings is the minimum number of operations (insert, edit or replace) to transform one string to another
  - Now introducing a “close” threshold so two points are consider as the same when they are close enough
    - LCSS: count the similar sample pairs
    - EDR: count the dissimilar sample pairs

Edit Distance is computed using dynamic programming.
Check the Wikipedia page

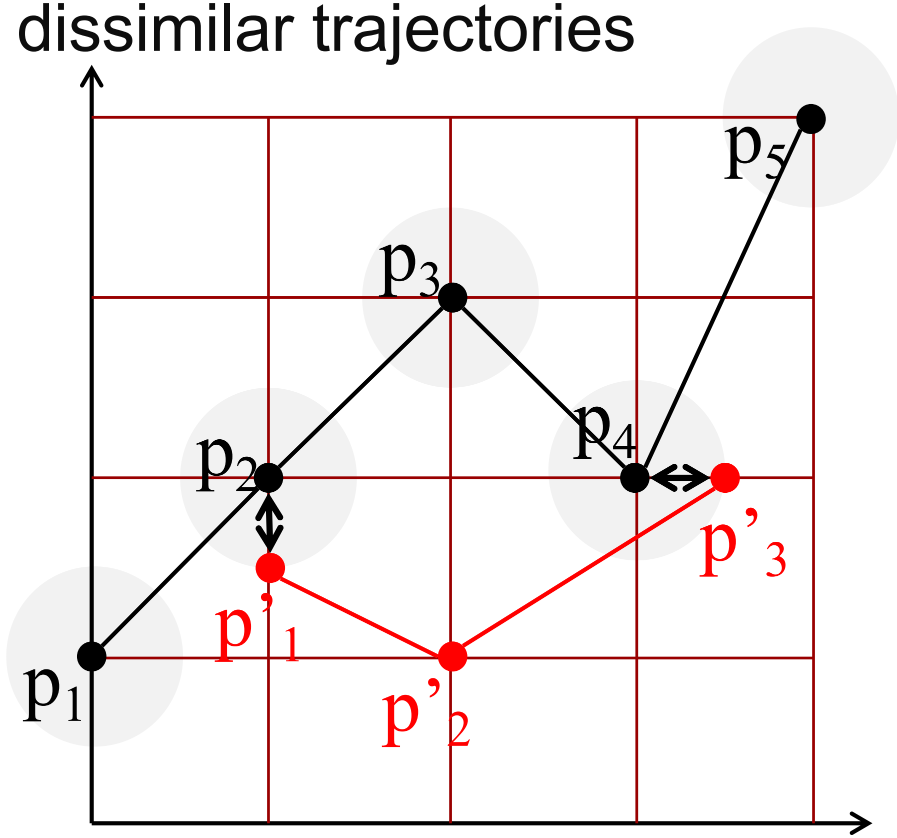
+ LCSS

- Longest Common Sub-Sequence
  - To find the longest common subsequence (which may not be consecutive) between two strings
  - LCSS(‘abcde’,’bd’) = ?   LCSS(‘abc’,’acb’) = ?
  - 1-to-(1 or null) mapping
  - This can also be computed using dynamic programming
    - Complexity O(mn) where m and n are the lengths of the two strings
- Adaptation of string similarity
  - Two locations are regarded as equal if they are ‘close’ enough (compared to a threshold)

VLACHOS, M., GUNOPULOS, D., AND KOLLIOS, G. Discovering similar multidimensional trajectories. ICDE 2002

+ LCSS

- Insensitive to noise
- Not easy to define threshold
- May return dissimilar trajectories



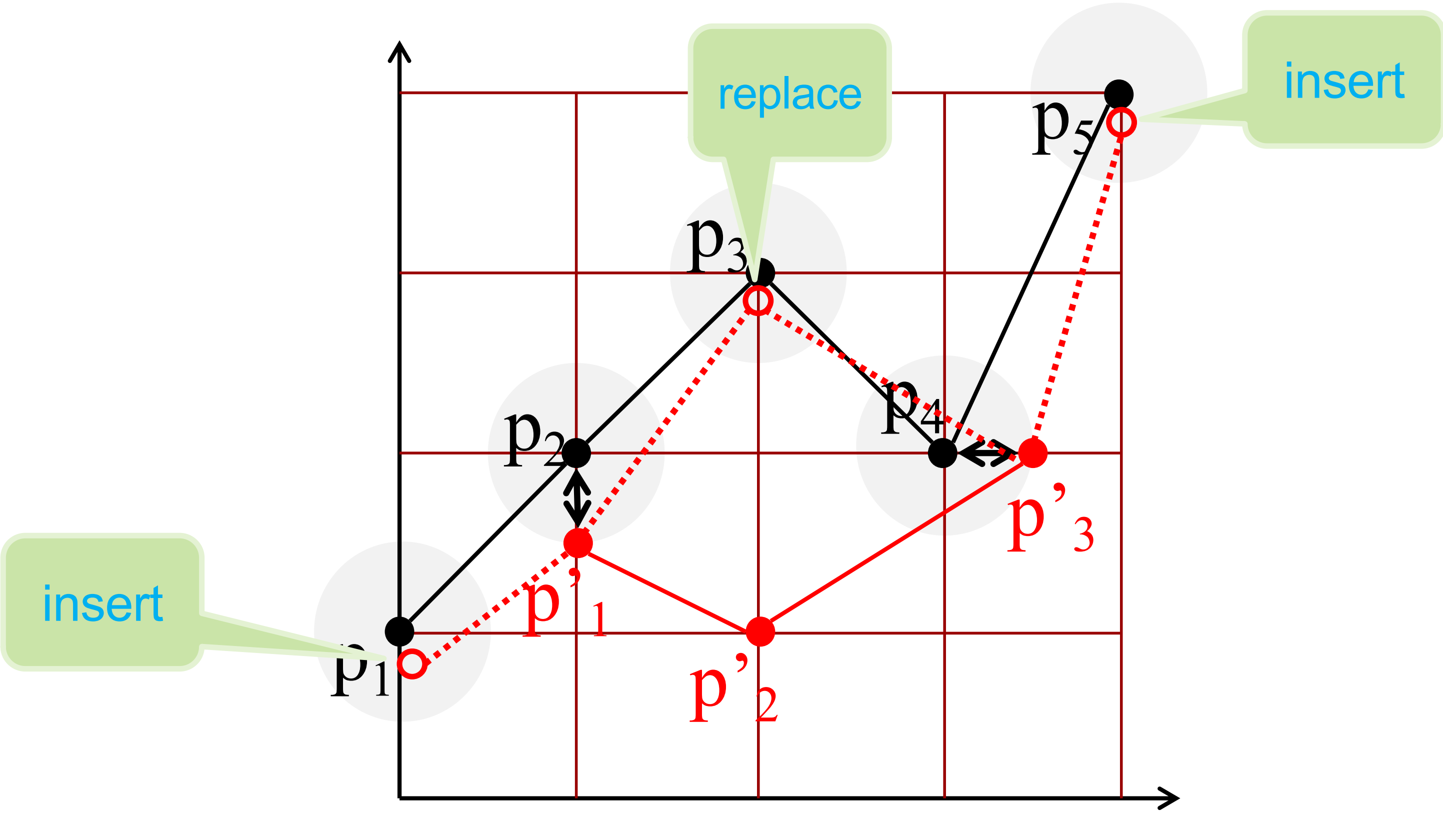
+ EDR

- Edit Distance on Real sequence
- Adaptation from Edit Distance on strings
  - Number of insert, delete, replace needed to convert one string into another
  - Two locations are regarded as equal if they're 'close' enough (compared to a threshold)

Lei Chen, M. Tamer Ozsu, Vincent Oria, Robust and Fast Similarity Search for Moving Object Trajectories. SIGMOD 2005

+ EDR

- Value means the number of operations, not "distance between locations"
  - Insensitive to noise



+ LCSS and EDR

- They are both count-based
  - LCSS counts the number of matched pairs
  - EDR counts the cost of operations needed to fix the unmatched pairs
- Higher LCSS, lower EDR

+ Continuity

- So far, discrete measures only
  - Only consider the sample points of trajectory
  - All previous measures are in this category
- Continuous measures
  - Consider the line segments between samples
  - OWD
  - LIP

+ OWD

- One Way Distance from T1 to T2 is:
  - Integral of the distance from points of T1 to T2
  - Divided by the length of T1

$$D_{\mathrm{owd}}(T_1, T_2) = \frac{1}{|T_1|}\left(\int_{p \in T_1} D_{\mathrm{point}}(p, T_2)\, dp\right)$$

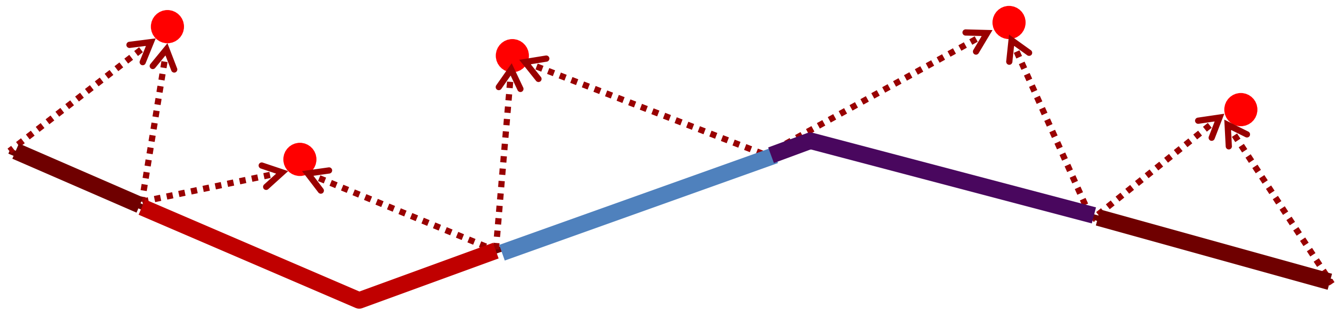
- Make it into symmetric measure

$$D(T_1, T_2) = \frac{1}{2}\left(D_{\mathrm{owd}}(T_1, T_2) + D_{\mathrm{owd}}(T_2, T_1)\right)$$

Bin Lin, Jianwen Su, One Way Distance: For Shape Based Similarity Search of Moving Object Trajectories. In Geoinformatica (2008)

+ OWD example

- Consider one trajectory as piece-wise line segment, and the other as discrete samples



+ LIP distance

- Locality In-between Polylines

$$LIP(Q,S) = \sum_{\forall\, polygon_i} Area_i \cdot w_i$$

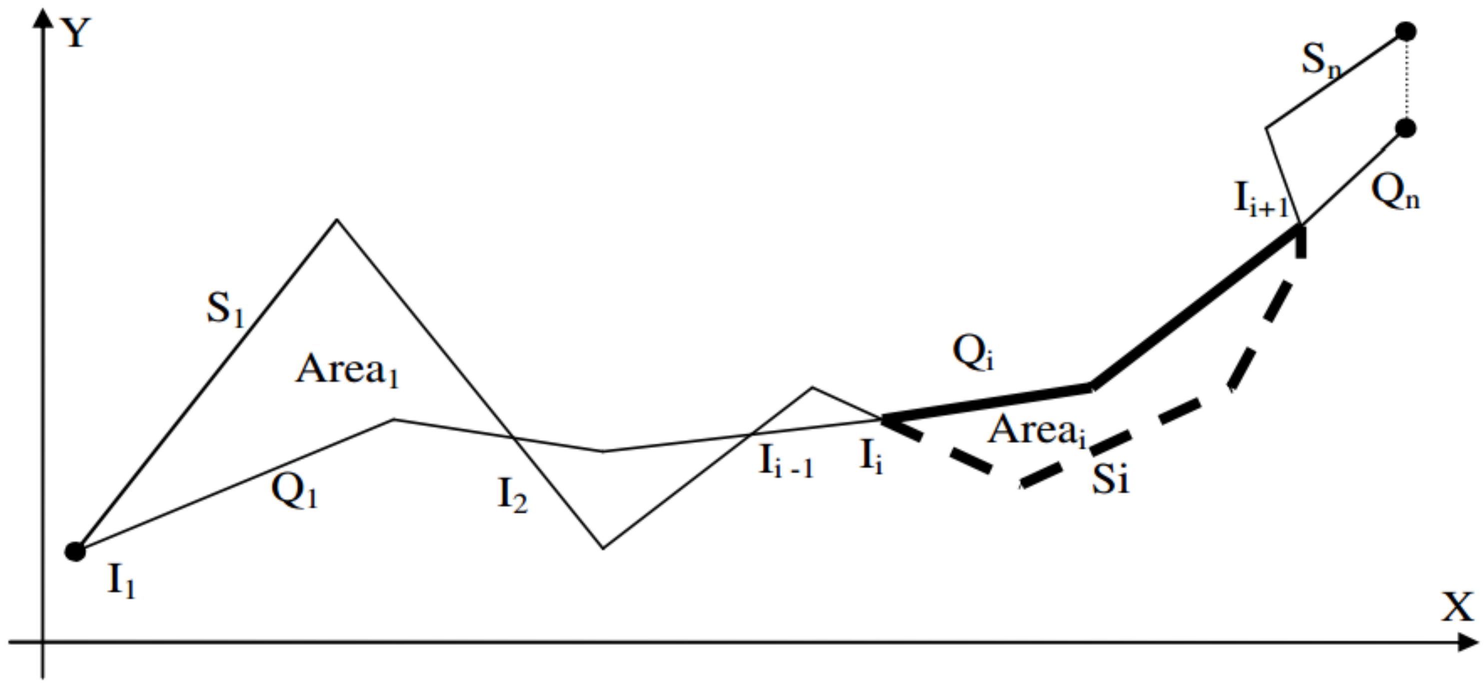
- *Polygon* is the set of polygons formed between intersection points

- $$w_i = \frac{Length_Q(I_i, I_{i+1}) + Length_S(I_i, I_{i+1})}{Length_Q + Length_S}$$

Nikos Pelekis et al, Similarity Search in Trajectory Databases. Symposium on Temporal Representation and Reasoning 2007

+ LIP distance

- Only works for 2-dimensional trajectories
- Polygon → polyhedron: non-trivial change

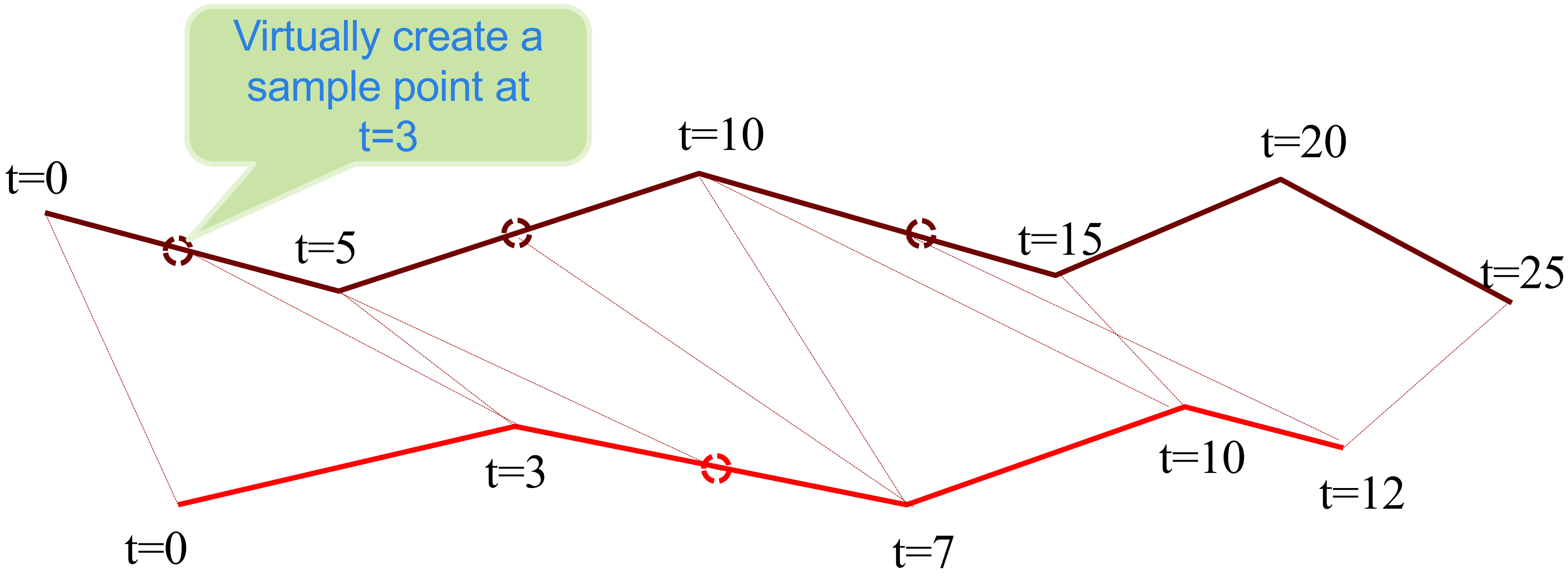


+ Spatiotemporal Distances

- So far, spatial only
  - Disregard the time information on sample points
- Spatiotemporal
  - Take the timestamp into consideration
  - Synchronous Euclidean Distance

+ SED

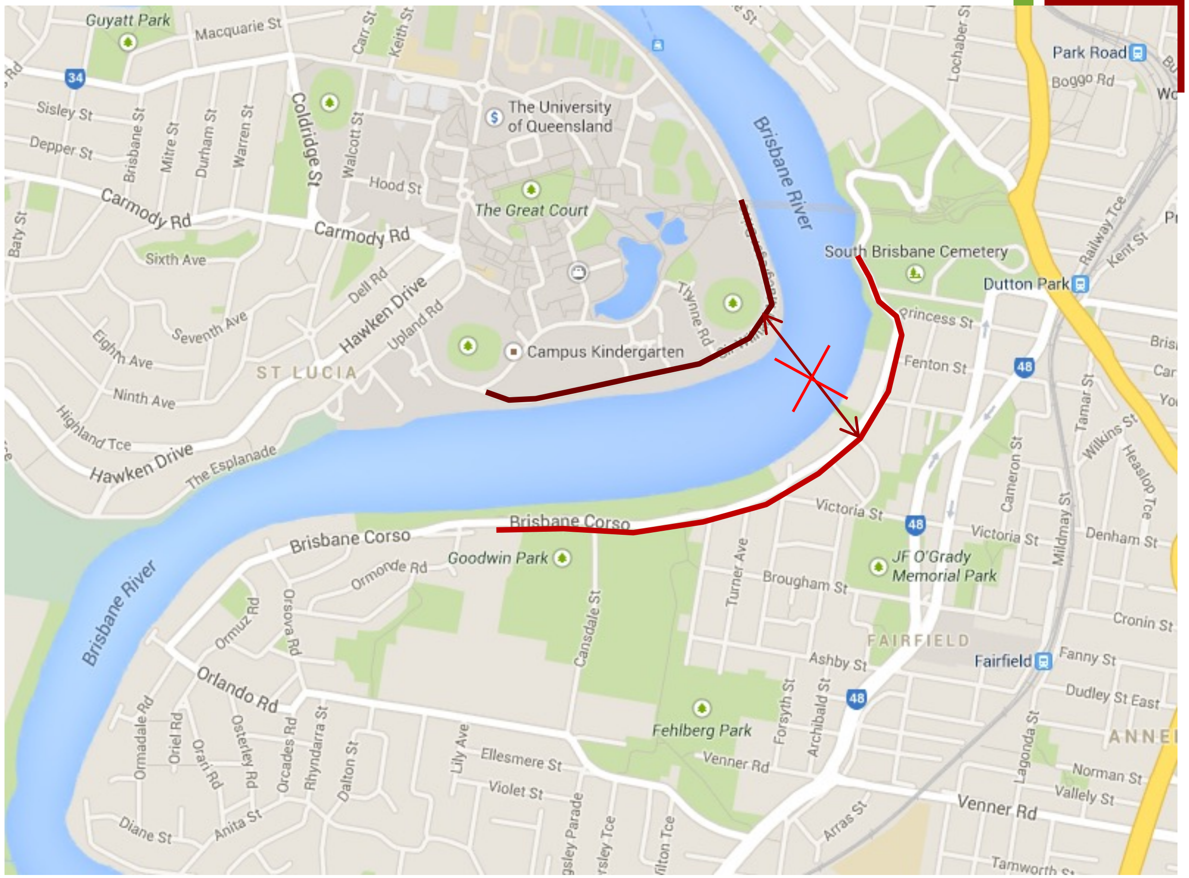
- Synchronous Euclidean Distance
  - Euclidean distance between locations at the same time instance of two trajectories



Mirco Nanni, Dino Pedreschi, Time-focused clustering of trajectories of moving objects. Journal of Intelligent Information Systems (2006)
POTAMIAS, M., PATROUMPAS, K., AND SELLIS, T. K. Sampling trajectory streams with spatiotemporal criteria. SSDBM 2006

+ Underlying Space

- We have considered on the Euclidean Space so far
- Road network
  - Distance is defined along shortest path



Jung-Rae Hwang, Hye-Young Kang, Ki-Joune Li, Searching for Similar Trajectories on Road Networks Using Spatio-temporal Similarity. Advances in Databases and Information Systems, pp 282 – 295, 2006

+ Trajectory Compression: The Need

- One record every 10 s, 24 bytes/record (min), 10 hrs/day

	Day	Month	Year
1 object	84KB	2.5MB	30MB
60,000 taxis	5GB	140GB	1.7TB

- Significant level of redundancy (i.e., the need for compression)
- Data quality can be poor (i.e., lossless compression may not be meaningful)

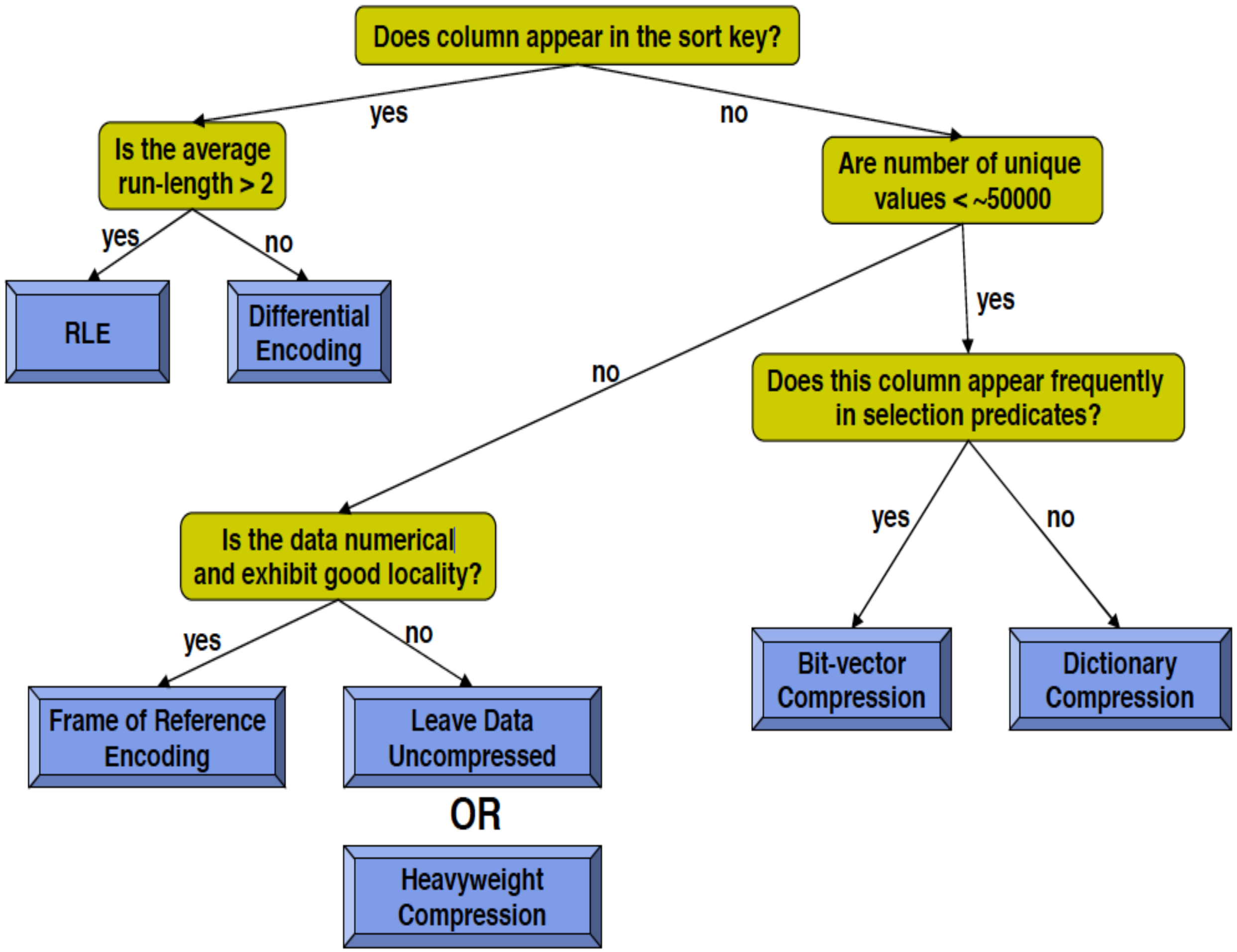
+ Simplification and Compression

- Trajectory simplification
  - Removing redundant information in a trajectory
- Trajectory compression
  - Reduce the amount of data without too much information loss
- Questions
  - Goals: size, quality, fitness for use, processing efficiency…
  - Intra-, inter- or knowledge-assisted?
    - That is, within one trajectory, among a set of trajectories, and use of other information such as road network information or some kind of dictionary and patterns
  - Geometric, spatiotemporal, semantic?

+ Data Compression

- Run length encoding
- Bit-vector
- Dictionary encoding
- Frame of reference encoding
- Differential encoding (escape sequence)
- Heavyweight encoding

What Compression Scheme To Use?



+ Line Simplification

- Piecewise linear approximation
  - Curves and spline based methods do exist but seldom used
- Naïve reduction methods
  - Every a few points, or every given distance
- Two subproblems
  - Min-ε: minimize error for a given maximum number of points
  - Min-#: minimize #points for a given error threshold
- Research areas: computational geometry, spatial database and GIS, pattern recognition, signal processing, image processing…
- Optimal: $O(n^2 \log n) \sim O(n^2)$, heuristics-based: $O(n \log n) \sim O(n)$

+ Criteria of Line Simplification

- In cartography (Weibel 1997)
  - (1) Reduction of data size and complexity
  - (2) Emphasizing the essential while suppressing the unimportant
  - (3) Maintaining logical and unambiguous relations
  - (4) Preserving aesthetic quality

- For us?
  - (1) and (2) above remain to be important
    - Need to establish the context and refine the criteria
  - (3) is hard to define, but is still important
    - Don't introduce misleading information (passing obstacle areas)
    - Support similarity-based operations
  - (4) might be unimportant
  - Any new criteria?

"Generalization of spatial data: Principles and selected algorithms", R. Weibel, in *Algorithmic Foundations of Geographic Info. Sys.*, 1997

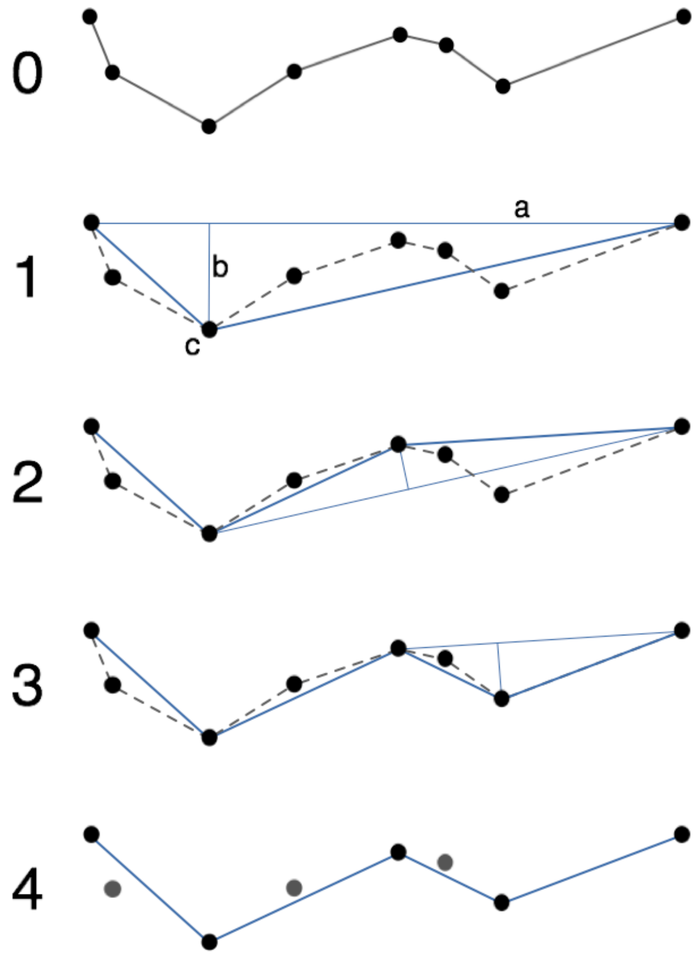
+ Lang's Algorithms

- A first step beyond removing random points (e.g. every 2nd point) or looking at the immediate neighboring points (all points within a pre-defined distance)
- From point i, check between i and i+k (i.e., looking ahead)
  - Draw a line L between i and i+k
  - Find perpendicular distance from i+1, …, i+k-1 to L
  - Remove all points i+1, …, i+k-1 if the distance is acceptable, and move the start point to i+k; otherwise k-- and repeat
- Notes
  - Only use a subset of existing points? Noisy data?
  - Prior-knowledge (such as a map) and post-processing (such as non-self crossing)

"Rules for robot draughtsmen", T. Lang, Geographical Magazine, 1969

+ Douglas-Peucker Algorithm

- A classic algorithm, simple, and "most superior"
- Top-down, to check if the deviation from a straight line is acceptable
- Computationally costly: $O(n^2)$, reducible to $O(n \log n)$
- 'Area difference' can be used to measure too



"Algorithms for the reduction of the number of points required to represent a digitized line or its caricature", D. Douglas & T. Peucker, The Canadian Cartographer, 1973

+ Li-Openshaw Algorithm

- Naturally adaptive to visual effects
  - SVS: smallest visible size
  - Aka Scale-specific or multiresolution generalization, a raster-vector approach
  - All points within a SVS (a pixel) collapse into one (the centroid)
    - Many benefits (see Weibel's criteria)

"Algorithms for objective generalization of line features based on the natural principle", Z. Li and S. Openshaw, IJGIS, 1992

+ Temporal DP for Trajectories

- Let (p_i, …, p_j) be the subcurve to simplify, the perpendicular distance from p_k, $i < k < j$, to line (p_i,p_j) is replaced by the Synchronous Euclidean Distance between (p_k, p'_k):

$$x_k' = x_i + \frac{t_k - t_i}{t_j - t_i}(x_j - x_i)$$

$$y_k' = y_i + \frac{t_k - t_i}{t_j - t_i}(y_j - y_i)$$

$$SED(p_k, p_k') = \sqrt{(x_k - x_k')^2 + (y_k - y_k')^2}$$

  - O(n log n)
  - Can use spatiotemporal error measures: Threshold-guided distance (SSDBM 2006), spatial join distance (VLDBJ 2006) and Fréchet distance (Int'l J Comp. Geometry and App.1995) – note: yet to study these
  - Used in our VLDB 2008 work

"A new perspective on trajectory compression techniques", N. Meratnia and R. A. de By, EDBT 2004

+ With Network Constraints

- Consider map-matching and trajectory compression at same time （by different orders)
  - Map-matching by Brakatsoulas, Pfoser, Salas and Wenk (VLDB 2005)
  - Trade-off between compression ratio and similarity
- New idea: using shortest path for compression
  - Between which pair of points?
  - Key technique: minimum description length (MDL)
    - L(H) + L(D|H) (for compression and differences caused)
    - Need to consider all-pair combination to optimize
    - Using a greedy approach, with pre-computed requires all-pair SP

"Trajectory compression under network constraints", G. Kellaris, N. Pelekis and Y. Theodoridis, SSTD 2009

+ Semantic Compression

- Only use those semantically more important points
  - Curve apexes? Road intersections? Transit points?
- Compress a trajectory into a sequence of *events*
  - Events: street intersections, public transport stops
  - Movement by direction (e.g., straight, left) or by event labels
  - Compression by combining consecutive homogeneous events
- Decompress to paths only with important events with estimated timestamps based on linear behavior of movement

“Semantic trajectory compression”, F. Schmid, K.-F. Richter and P. Laube, SSTD 2009

+ Mapping Trajectories to Events

- Map trajectories to activities by considering the nearest neighbor constraint and time duration constraint

- The focus is on efficient join processing
  - A set of trajectories
  - A set of POIs associated with activities and min/max durations
  - A part of a trajectory can be linked to an activity (or multiple activities) if the corresponding POI is the nearest of that part of trajectory and the duration meets the min/max time constraint

"From trajectories to activities: a spatiotemporal join approach", K. Xie and X. Zhou, ACM GIS-LBSN 2009

+ Trajectory Rewriting: The Need

- A case study
  - Using 4 popular similarity measures, average normalized distance values over 1000 dense trajectories rewritten to different sampling rates

Sampling rate	ED	DTW	LCSS	EDR
10	0.35	0.21	0.41	0.55
20	0.21	0.09	0.27	0.37
30	0	0	0	0
60	0.24	0.15	0.33	0.23
100	0.25	0.21	0.45	0.28

+ Reference System Based Rewriting

- To make heterogeneous trajectories *compatible* by rewriting
- Four reference systems
  - Grid-based
  - Data-based
  - POI-based
  - Feature-based
- Two rewriting operations
  - Alignment and compliment

"Calibrating trajectory data for similarity-based analysis", H. Su, K. Zheng, H. Wang and X. Zhou, SIGMOD 2013

+ Other Trajectory Related Topics

- Map-matching and map-inferencing
- Route planning
- Trajectory mining
- Spatiotemporal entity linking
- Location privacy protection
- Road speed profile prediction
- Predictive route planning
- Large-scale trajectory data management systems

+ Map Matching and Inferencing

- Map matching is to match recorded geographic coordinates to a digital map
  - Thus, a GPS trajectory will be mapped to a sequence of road segments
  - GPS data errors can be corrected
- Map inferencing is to correct maps based on GPS trajectories
  - To identify new roads, change of road conditions etc

- Pingfu Chao, Wen Hua, Rui Mao, Jiajie Xu, Xiaofang Zhou, “A Survey and Quantitative Study on Map Inference Algorithms from GPS Trajectories”, IEEE Transactions on Knowledge and Data Engineering, 34(1): 15-28 (2022)
- Pingfu Chao, Yehong Xu, Wen Hua, Xiaofang Zhou, “A Survey on Map-Matching Algorithms”. ADC 2020: 121-133

+ Route Planning

- Given a weighted road network, to find the shortest path from origin to destination
  - Weights can be distance, time, costs, fuel consumption etc
  - Basic algorithms: Dijkstra algorithm and A* algorithm
- Advanced routing algorithms
  - Time-dependent shortest path planning
  - Batch routing
  - New indexing structures (e.g., 2-hop, CH trees)
  - Multi-criteria routing and KSP problem
  - Routing algorithms for Evs
  - Equilibrium routing

…this is a very active research area, and my group is a leader in this research, check my homepage for papers

+ Trajectory Mining

- Trajectory clustering (SIGMOD 2007)
  - Partition-and-group, and partition based on MDL
- Trajectory pattern mining (ICDE 2008)
  - Long trajectories are divided based a duration T, and then they are aligned and points are density-based clustered
- Finding convoys (VLDB 2008)
  - A group of objects travelling together for long enough
  - Used line simplification algorithms for efficiency
  - Finding “swarms” (VLDB 2010)
    - Allow temporary divergence

+ Spatiotemporal Entity Linking

- For two sets of trajectories, linking the entities in these two datasets based on their trajectories
- Signatures can be created for each entity based on the first m locations with largest location frequency-inverse trajectory frequency values
  - Highly accurate linking accuracy
  - Highly efficient with a R-tree variation index

- Fengmei Jin, Wen Hua, Jiajie Xu, Xiaofang Zhou: Moving Object Linking Based on Historical Trace. ICDE 2019

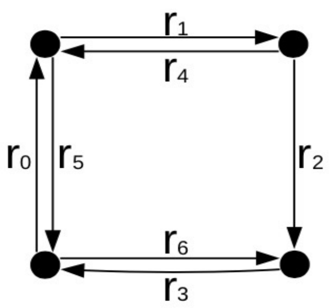
+ Location Privacy Protection

- Traditional methods
  - k-anonymity, i-diversity, t-closeness and differential privacy
  - Many ad hoc methods
- Privacy vs utility
- Signature-based point removal
- DP-based frequency modification

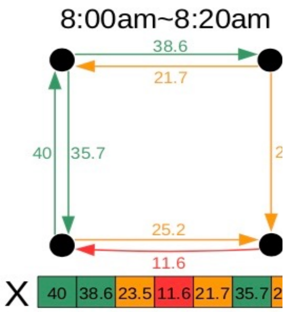
- F Jin, W Hua, M Francia, P Chao, M Orlowska, X Zhou , “A Survey and Experimental Study on Privacy-Preserving Trajectory Data Publishing”, TechRxiv 2021.
- F Jin, W Hua, B Ruan, X Zhou, "Frequency-based Randomization for Guaranteeing Differential Privacy in Spatial Trajectories', ICDE 2022.

+ Speed Profile Prediction

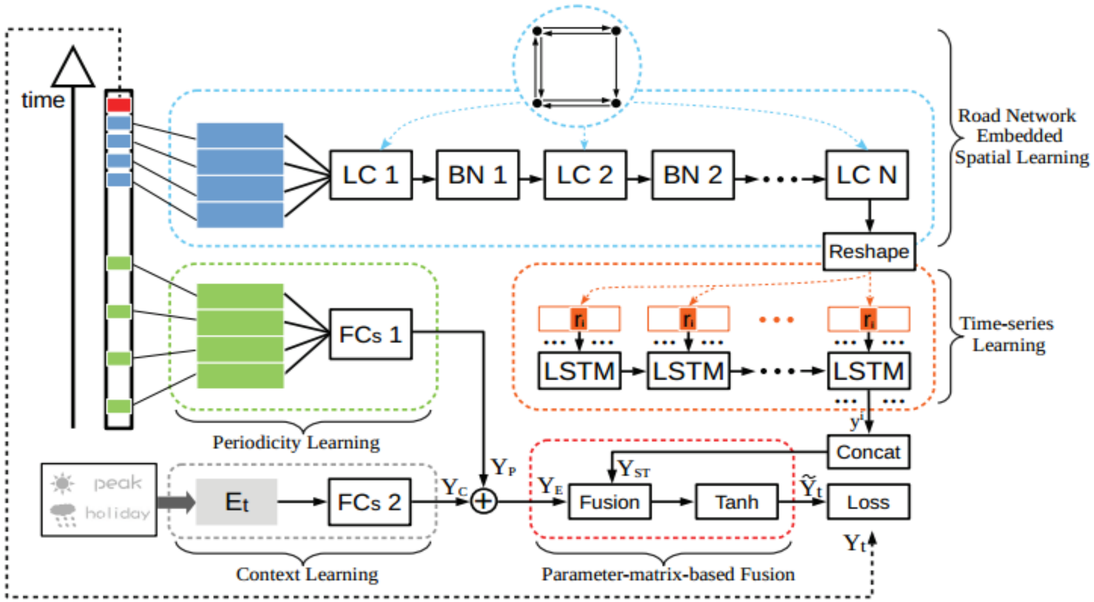
Problem: Given the historical observations $\{X_i | i = 1, ..., t\}$, this paper aims to predict $Y_t = \{X_j | j = t+1, ..., t+z\}$, where z is the number of time intervals to be predicted.



(a) A road network



(b) A speed vector 8:00am - 8:20am



LC-RNN model

- Previous approaches: ARIMA based (conventional), RNN based (consider time only), CNN based (spatial information but previously only at grid level)
- Look-up Convolution (LC): learn the latent features of surrounding area
- LSTM: learn the time-series pattern that is aware of surrounding area dynamics

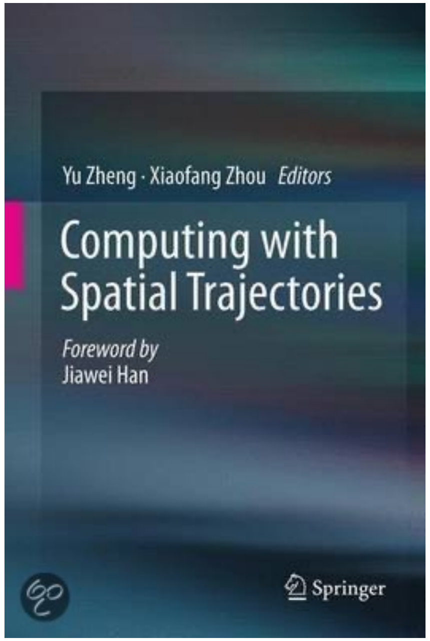
Z. Lv, J. Xu, K. Zheng, P. Zhao, H. Yin, X. Zhou, "LC-RNN: A Deep Learning Model for Traffic Speed Prediction", **IJCAI** 2018.

+ Predicative Routing

- Task: you want to do route planning for a future time
- What you have: many prediction models
- Questions: in the context of a given query
  - Which model to choose?
  - How to retrain efficiently?
- This is an example of predictive data analytics
  - Descriptive, predictive and prescriptive analytics
  - Data can come from databases, sensors and predication models
  - A new breed of "database system" is needed!

+ An Introduction Book

- ***Computing with Spatial Trajectories***
  - Yu Zheng and Xiaofang Zhou, 2011
- Part I Foundations
  - Trajectory Preprocessing *(W.-C. Lee, J.Krumm)*
  - Trajectory Indexing and Retrieval *(X. Zhou et al)*
- Part II Advanced Topics
  - Uncertainty in Spatial Trajectories *(G. Trajcevski)*
  - Privacy of Spatial Trajectories *(C.-Y. Chow, M. Mokbel)*
  - Trajectory Pattern Mining *(H. Young, K. L. Yiu, C. Jensen)*
  - Activity Recognition from Trajectory Data *(Y. Zhu, V. Zheng, Q. Yang)*
  - Trajectory Analysis for Driving *(J. Krumm)*
  - Location-Based Social Networks: Users *(Y. Zheng)*
  - Location-Based Social Networks: Locations *(Y. Zheng and X. Xie)*



+ Summary

- Spatiotemporal data is common
- Spatiotemporal data indexing and query processing are different from spatial ones
- Trajectory data is of particular importance with a wide range of applications
- Raw trajectory data cannot be compared directly
- Trajectory analytics is a new research direction