Data Mining II

Mobility Data Mining

Mirco Nanni, ISTI-CNR

Main source: Jiawei Han, Dep. of CS, Univ. IL at Urbana-Champaign:
https://agora.cs.illinois.edu/display/cs512/Lectures

Mining Moving Object Data

- Introduction
- Movement Pattern Mining
- Periodic Pattern Mining
- Clustering
- Prediction
- Classification
- Outlier Detection

Why Mining Moving Object Data?

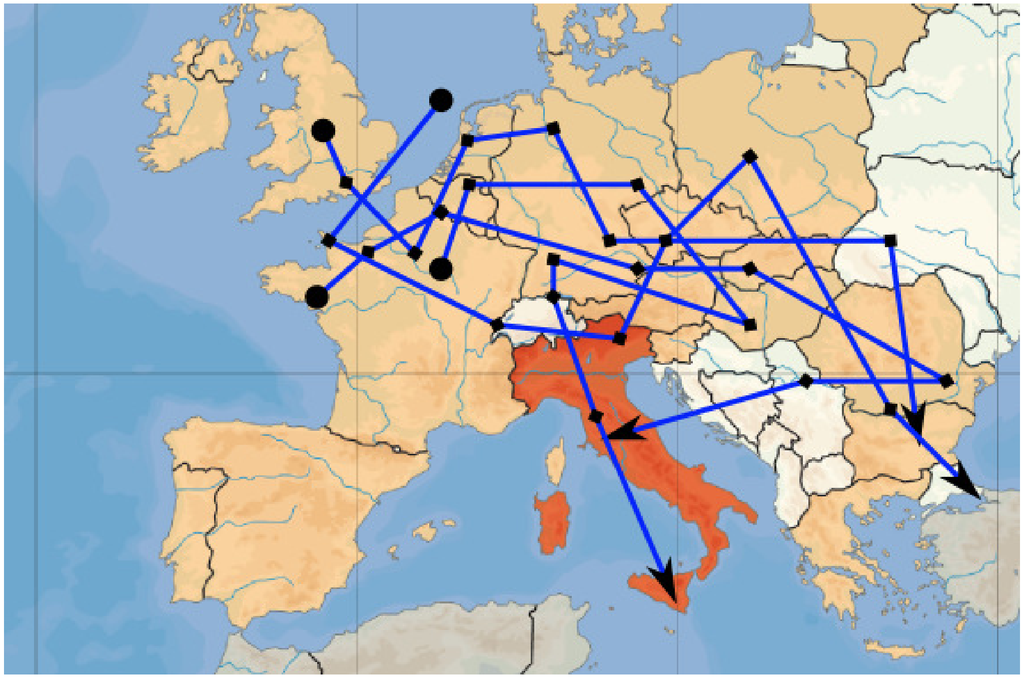
- Satellite, sensor, RFID, and wireless technologies have been improved rapidly
  - Prevalence of mobile devices, e.g., cell phones, smart phones and PDAs
  - GPS embedded in cars
  - Telemetry attached on animals
- Tremendous amounts of trajectory data of moving objects
  - Sampling rate could be every minute, or even every second
  - Data has been fast accumulated

Why Mining Moving Object Data?

- Large diffusion of mobile devices, mobile services and location-based services

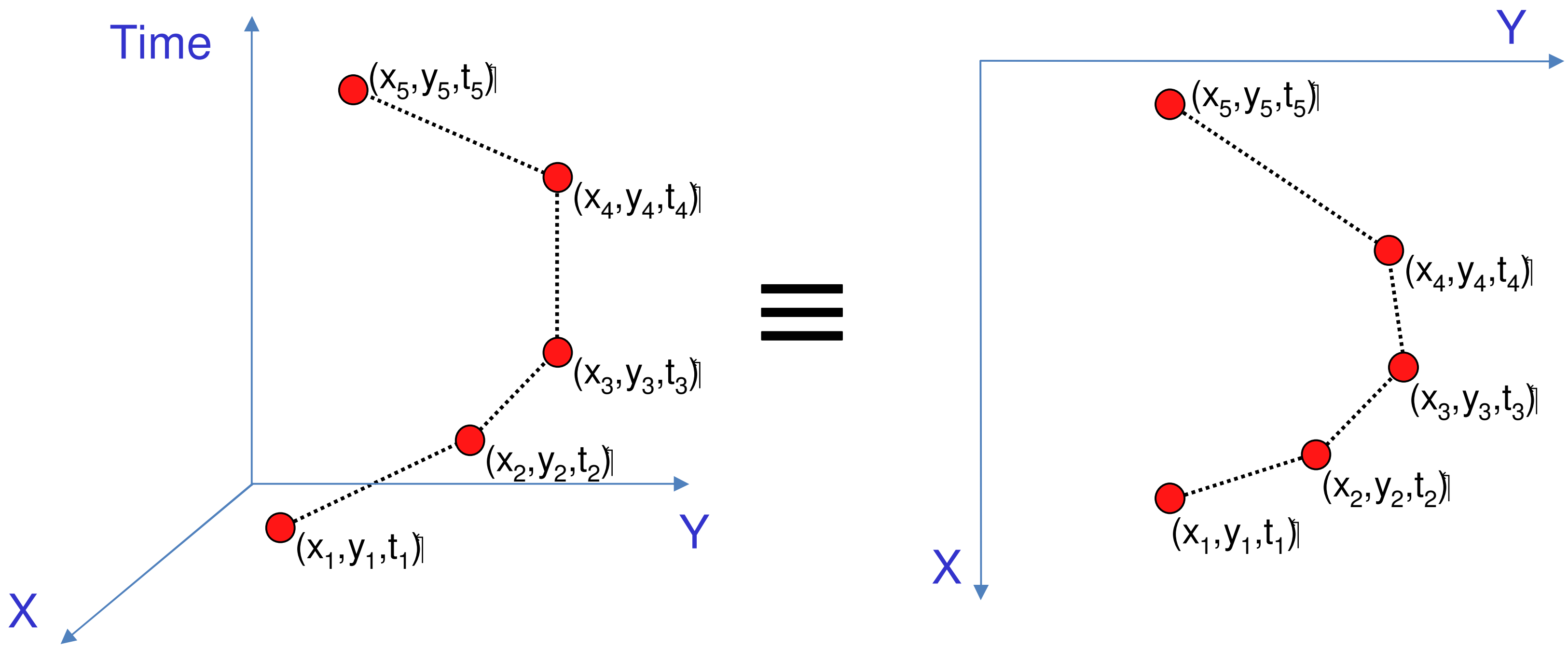
Why Mining Moving Object Data?

- Such devices leave digital traces that can be collected to obtrain *trajectories* describing the mobility behavior of its owner
- Trajectory: a sequence of the location and timestamp of a moving object



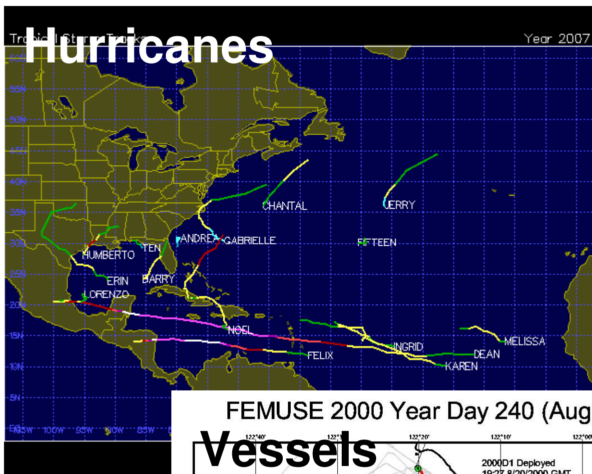
What is a trajectory

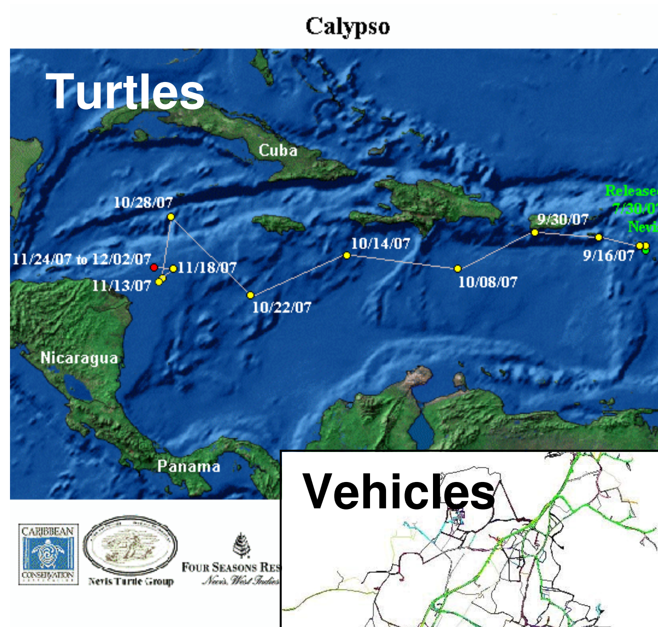
- Trajectories are usually given as *spatio-temporal (ST) sequences:* <(x_1,y_1,t_1), ..., (x_n,y_n,t_n)>

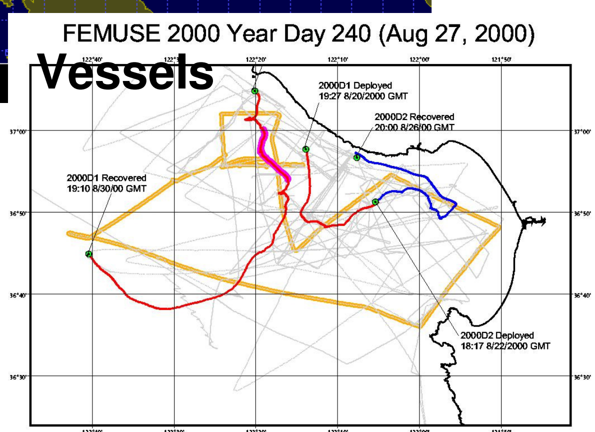


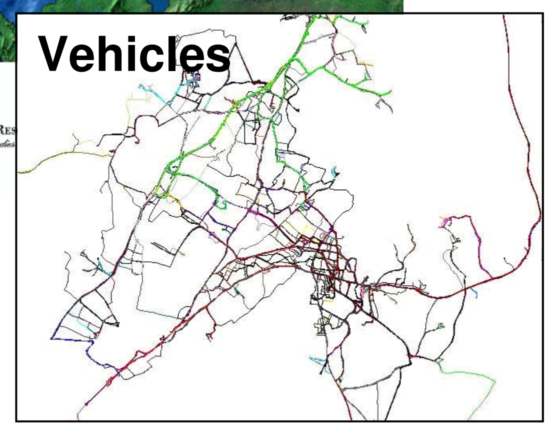
Moving Object Data

- Several domains:









Complexity of the Moving Object Data

- Uncertainty
  - Sampling rate could be inconstant: From every few seconds transmitting a signal to every few days transmitting one
  - Data can be sparse: A recorded location every 3 days
- Noise
  - Erroneous points (e.g., a point in the ocean)
- Background
  - Cars follow underlying road network
  - Animals movements relate to mountains, lakes, ...
- Movement interactions
  - Affected by nearby moving objects

Research Impacts

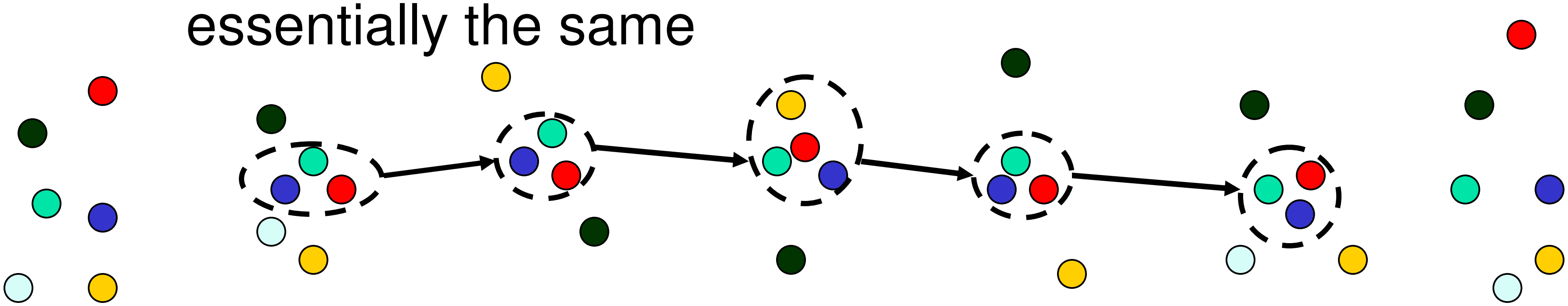
- Moving object and trajectory data mining has many important, real-world applications driven by the real need
  - Ecological analysis (e.g., animal scientists)
  - Weather forecast
  - Traffic control
  - Location-based services
  - Homeland security (*e.g.*, border monitoring)
  - Law enforcement (*e.g.*, video surveillance)
  - ...

Mining Moving Object Data

- Introduction
- Movement Pattern Mining
- Periodic Pattern Mining
- Clustering
- Prediction
- Classification
- Outlier Detection

Moving Clusters

- A ***moving cluster*** is a set of objects that move close to each other for a long time interval
  - **Note**: Moving clusters and flock patterns (see later) are essentially the same



- Formal Definition [Kalnis et al., SSTD'05]:
  - A ***moving cluster*** is a sequence of (snapshot) clusters $c_1, c_2, \ldots, c_k$ such that for each timestamp i ($1 \leq i < k$), $|c_i \cap c_{i+1}| / |c_i \cup c_{i+1}| \geq \theta \quad (0 < \theta \leq 1)$

Retrieval of Moving Clusters

(Kalnis et al. SSTD'05)

- Basic algorithm (MC1)

1. Perform DBSCAN for each time slice
2. For each pair of a cluster *c* and a moving cluster *g*, check if *g* can be extended by *c*
    - If yes, *g* is used at the next iteration
    - If no, *g* is returned as a result

- Improvements
  - MC2: Avoid redundant checks (Improve Step 2)
  - MC3: Reduce the number of executions of DBSCAN (Improve Step 1)

Relative Motion Patterns

(Laube et al. 04, Gudmundsson et al. 07)

- ***Flock:*** At least *m* entities are within a circular region of **radius *r*** and they move in the same direction

- ***Leadership*:** At least *m* entities are within a circular region of radius *r*, they move in the same direction, and **at least one of the entities was already heading in this direction for at least *s* time steps**

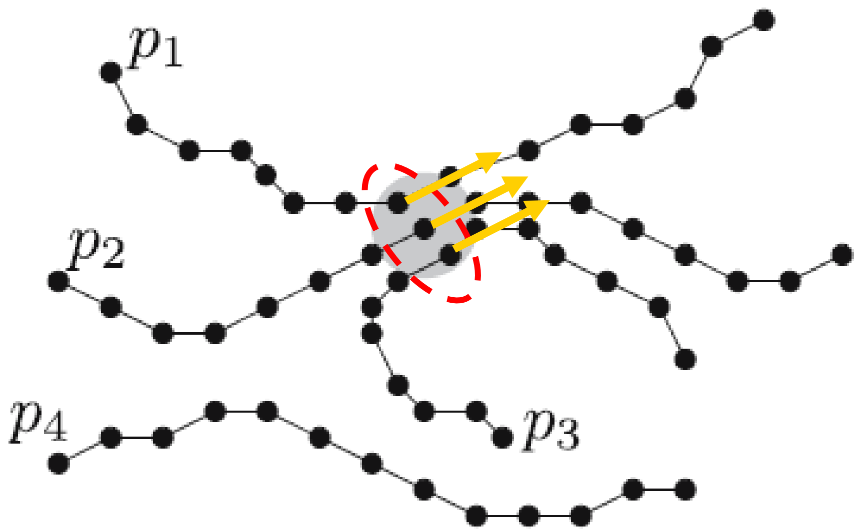
- ***Convergence:*** At least *m* entities will **pass through** the same circular region of radius *r* (assuming they keep their direction)

- ***Encounter:*** At least *m* entities will be **simultaneously inside** the same circular region of radius *r* (assuming they keep their speed and direction)

Relative Motion Patterns

(Laube et al. 04, Gudmundsson et al. 07)

- ***Flock*** ($m > 1$, $r > 0$): At least *m* entities are within a circular region of **radius *r*** and they move in the same direction

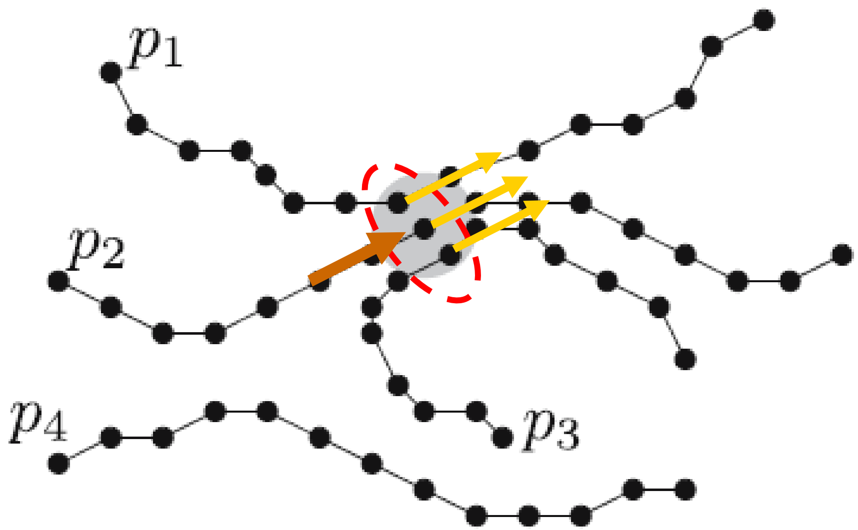


An example of a ***flock*** pattern for p_1, p_2, and p_3 at 8th time step; also a ***leadership*** pattern with p_2 as the leader

Relative Motion Patterns

(Laube et al. 04, Gudmundsson et al. 07)

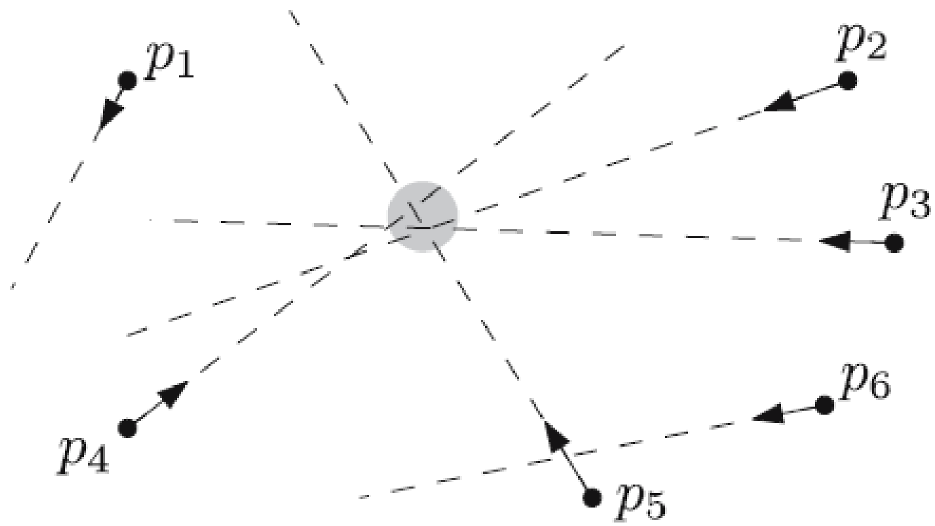
- ***Leadership*** ($m > 1$, $r > 0$, $s > 0$) At least m entities are within a circular region of radius r, they move in the same direction, and **at least one of the entities was already heading in this direction for at least *s* time steps**



An example of ***leadership*** pattern with p_2 as the leader

Relative Motion Patterns
(Laube et al. 04, Gudmundsson et al. 07)

- ***Convergence*** (*m* > 1, *r* > 0) At least *m* entities will **pass through** the same circular region of radius *r* (assuming they keep their direction)



A ***convergence*** pattern if $m = 4$ for p_2, p_3, p_4, and p_5

- ***Encounter*** (*m* > 1, *r* > 0). Variant: at least *m* entities will be **simultaneously inside** the same circular region of radius *r* (assuming they keep their speed and direction)

Complexity of Moving Relationship Pattern Mining

- Algorithms: Exact and approximate algorithms are developed

(Length *t* is multiplicative factor in all time bounds)

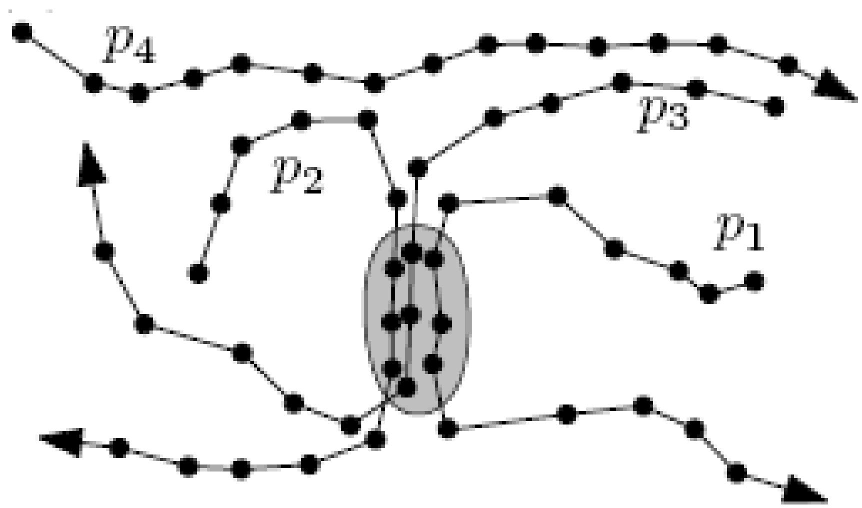
Pattern	Exact (from [15])	Exact (new)	Approximate
Flock	$O(nm^2 + n \log n)$	–	$O(\frac{n}{\varepsilon^2} \log \frac{1}{\varepsilon} + n \log n)$ (radius)
Leadership	$O(ns + nm^2 + n \log n)$	–	$O(ns + \frac{1}{\varepsilon^2} n \log \frac{1}{\varepsilon} + n \log n)$ (radius)
Convergence	$O(n^2)$	–	$O(n^{2+\delta}/(\varepsilon m))$ (subset)
		$O(n^3)$ (all)	$O(\frac{1}{\varepsilon} n^2 \log n)$ (radius)
Encounter	$O(n^4)$	$O((m + \log n)n^2)$ (detect) $O((M + \log n)n^2 \log M)$ (largest)	

  - Flock: Use the higher-order Voronoi diagram
  - Leadership: Check the leader condition additionally
  - …

An Extension of Flock Patterns

(Gudmundsson et al. GIS'06, Benkert et al. SAC'07)

- A new definition considers *multiple* time steps, whereas the previous definition *only one* time step
- ***Flock***: A *flock* in a time interval *I*, where the duration of *I* is at least *k*, consists of at least *m* entities such that for every point in time within *I*, there is a disk of radius *r* that contains all the *m* entities
  - e.g.,

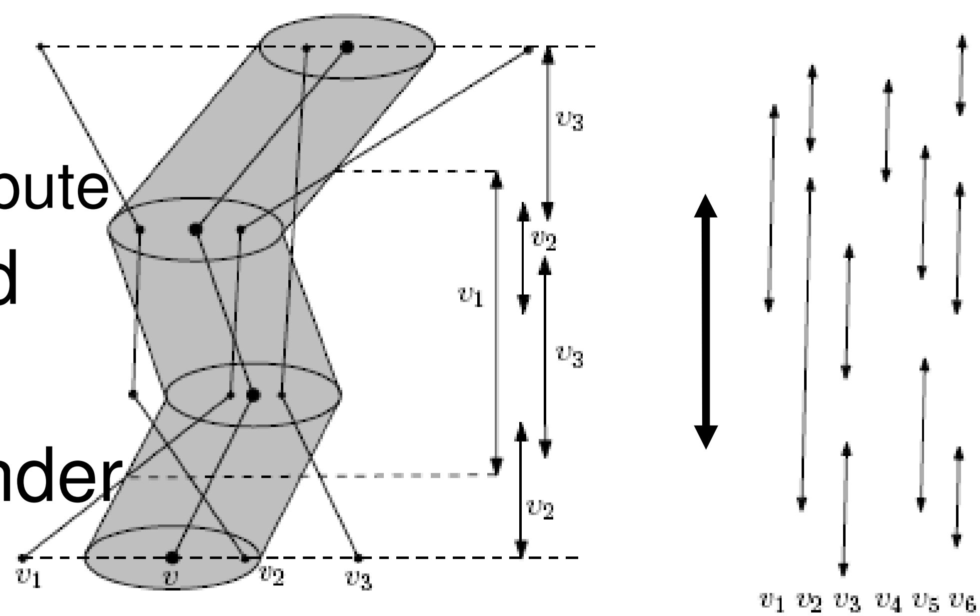


A flock through 3 time steps

Computing Flock Patterns

- *Approximate* flocks
  - Convert overlapping segments of length *k* to points in a *2k*-dimensional space
  - Find *2k*-d pipes that contain at least *m* points

- *Longest duration* flocks
  - For every entity *v*, compute a cylindrical region and the intervals from the intersection of the cylinder
  - Pick the longest one



Convoy: An Extension of Flock Pattern

(Jeung et al. ICDE'08 & VLDB'08)

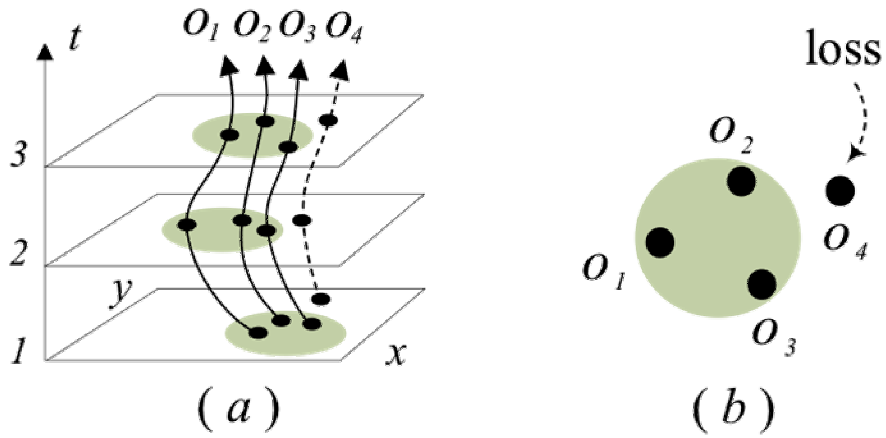


Figure 1: ***Lossy-flock*** **Problem**

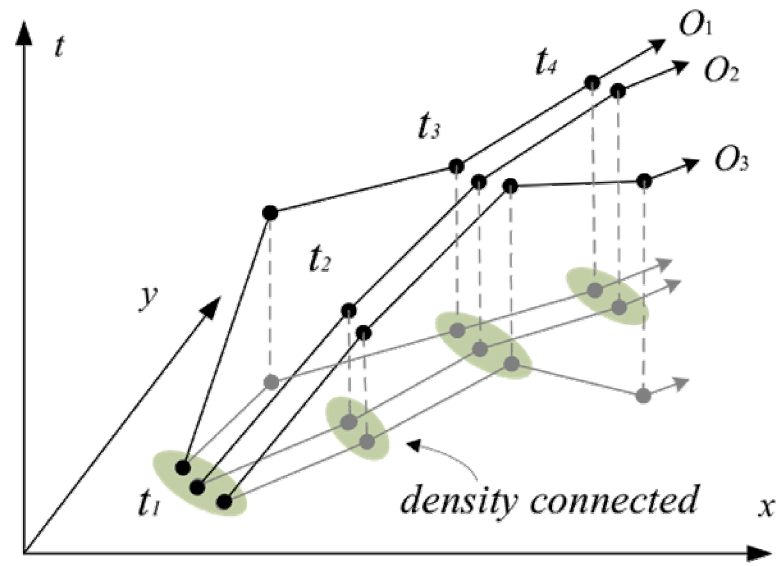


Figure 4: An Example of a Convoy

- Flock pattern has rigid definition with a circle
- Convoy use *density-based clustering* at each timestamp

Efficient Discovery of Convoys

- Base-line algorithm:
  - Calculate density-based clusters for each timestamp
  - Overlap clusters for every k consecutive timestamps
- Speedup algorithm using trajectory simplification
  - Trajectory simplification

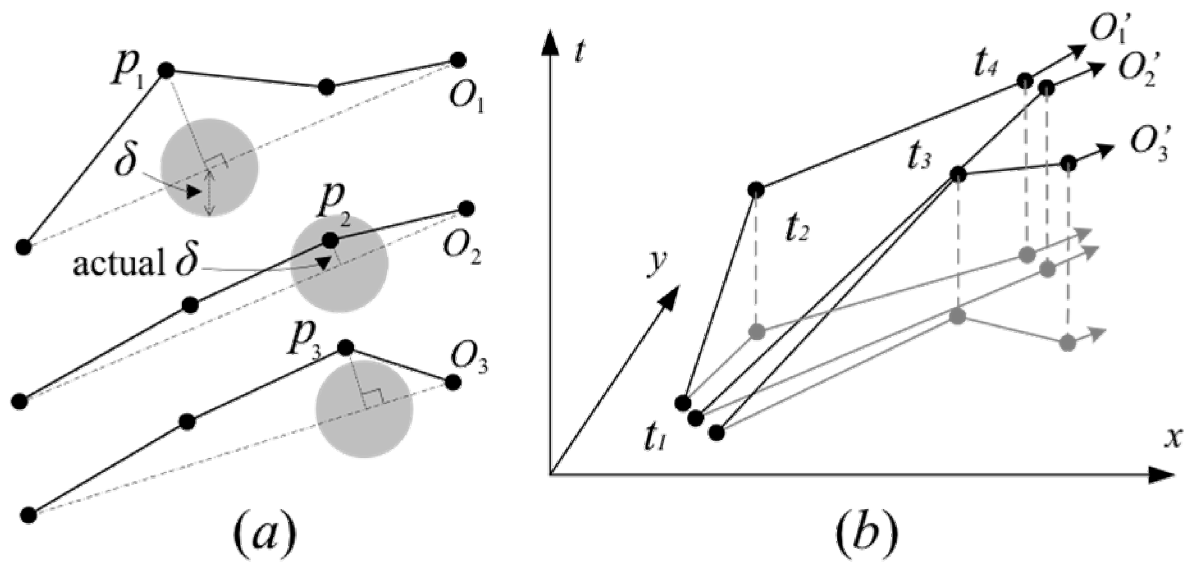


Figure 6: Trajectory Simplification

A Filter-and-Refine Framework for Convoy Mining

- Filter-and-refine framework
  - Filter: partition time into λ-size time slot; a trajectory is transformed into a set of segments; density-based clustering on segments.
  - Refine: Look into every λ-size time slot, refine the clusters based on points.

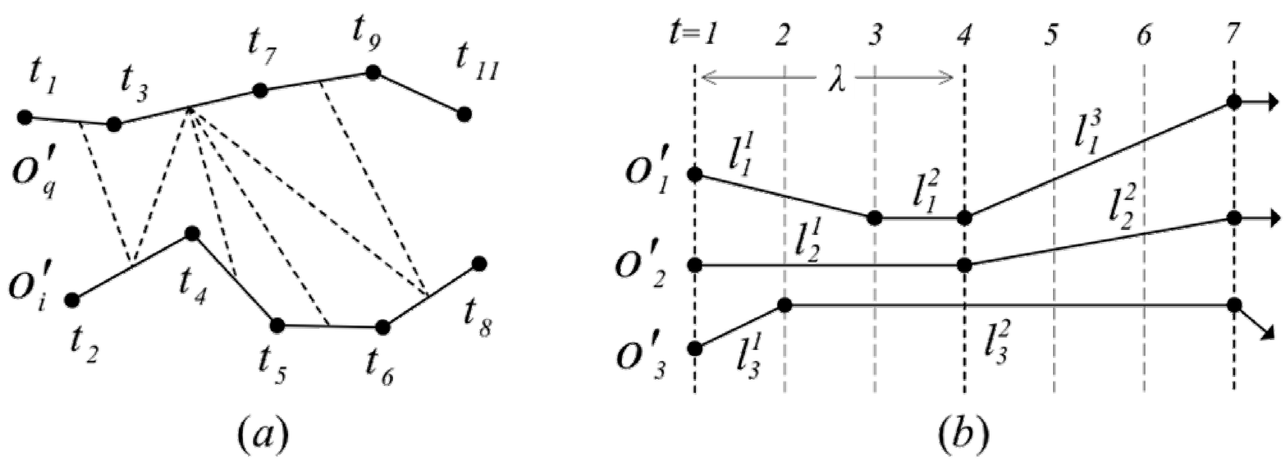


Figure 9: Measure of $\omega(o'_q, o'_i)$ and Time Partitioning

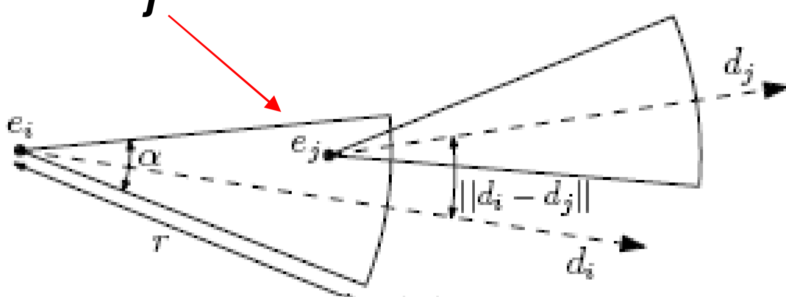
An Extension of Leadership Patterns

(Andersson et al. *GeoInformatica* 07)

- ***Leadership***: if there is an entity that is a leader of at least *m* entities for <u>at least *k* time units</u>
  - An entity e_j is said to be a *leader* at time $[t_x, t_y]$ for time-points t_x, t_y, if and only if e_j does not follow anyone at time $[t_x, t_y]$, and e_j is followed by sufficiently many entities at time $[t_x, t_y]$

e_j

e_i

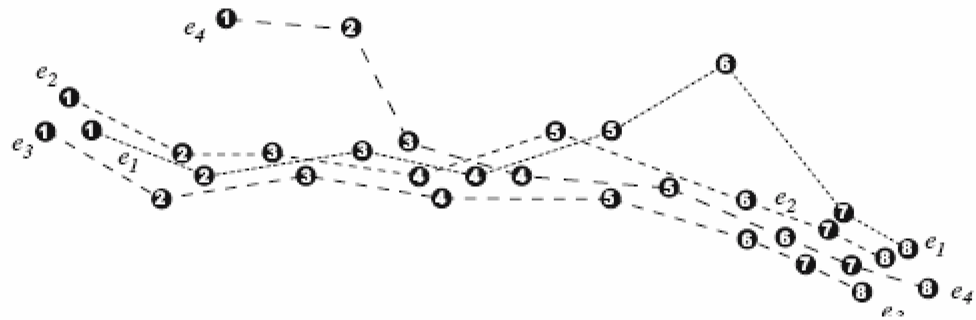


e_i follows e_j

$$||d_i - d_j|| \leq \beta$$

Reporting Leadership Patterns

- Algorithm: Build and use the follow-arrays

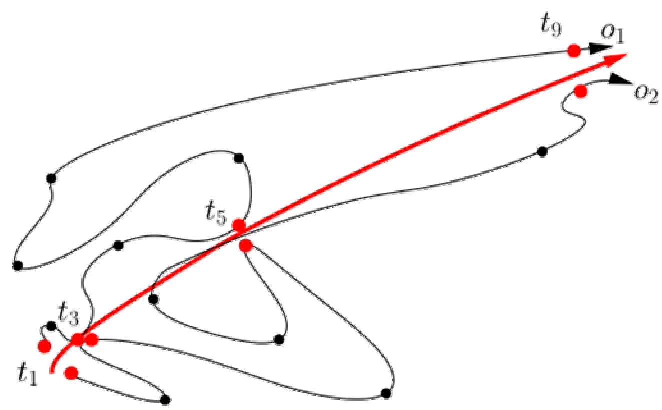


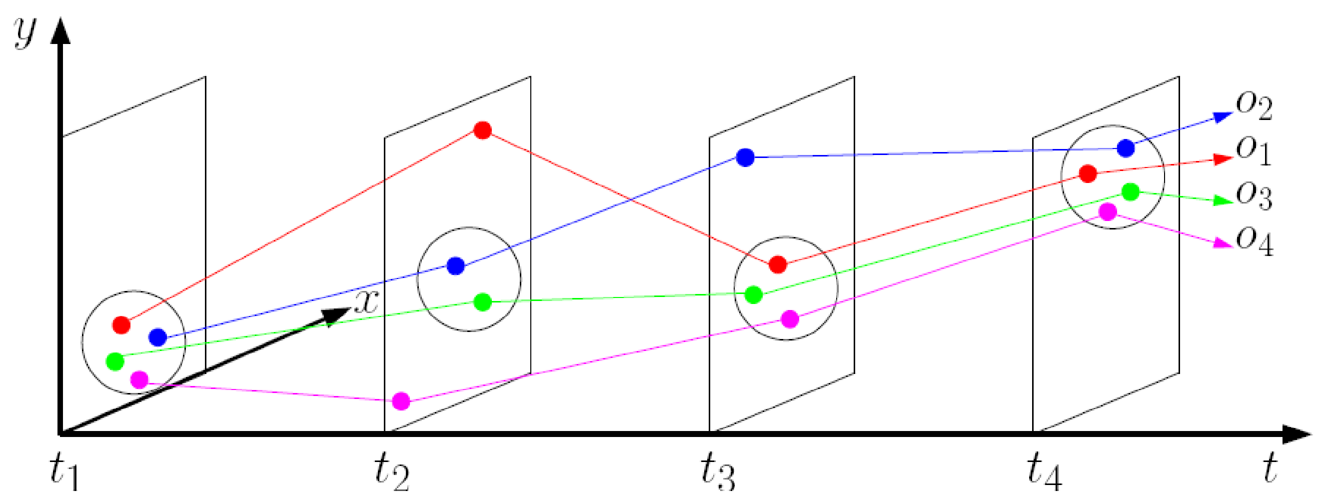
Array	Entity	Row	1	2	3	4	5	6	7	8
IntervalsNotFwg(t)	e_1	e_1	0	1	2	0	1	2	3	0
	e_2	e_2	0	0	0	0	0	1	0	0
	e_3	e_3	0	0	0	0	0	0	0	0
	e_4	e_4	0	1	2	3	4	5	6	7
IntervalsFwg(e',t)	e_1	e_2	0	1	2	3	4	0	0	0
		e_3	0	1	2	3	0	0	0	0
		e_4	0	0	0	0	0	0	0	0
	e_2	e_1	0	0	0	0	0	0	0	0
		e_3	0	0	0	0	0	0	0	0
		e_4	0	0	0	0	0	0	0	0
	e_3	e_1	0	0	0	0	0	0	0	0
		e_2	0	0	0	1	0	0	0	0
		e_4	0	0	0	0	0	0	0	0
	e_4	e_1	0	0	0	1	0	0	0	1
		e_2	0	0	0	0	0	0	1	2
		e_3	0	0	0	0	1	2	3	4
IntervalsFwd$_m$(t): *(m=1)*	e_1	e_1	0	1	2	3	4	0	0	0
	e_2	e_2	0	0	0	0	0	0	0	0
	e_3	e_3	0	0	0	1	0	0	0	0
	e_4	e_4	0	0	0	1	2	3	4	5
numFws(t)	e_1	e_1	0	2	2	2	1	0	0	0
	e_2	e_2	0	0	0	0	0	0	0	0
	e_3	e_3	0	0	0	1	0	0	0	0
	e_4	e_4	0	0	0	1	1	1	2	3

e.g., Store nonnegative integers specifying for how many past consecutive unit-time-intervals e_j is following e_i ($e_j \neq e_i$)

Swarms: A Relaxed but Real, Relative Movement Pattern

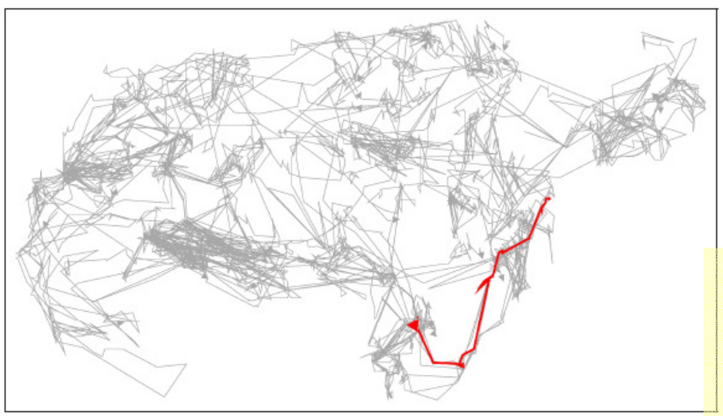
- Flock and convoy all require k consecutive time stamps (still very rigid definition)
- Moving objects may not be close to each other for consecutive time stamps (need to relax time constraint)





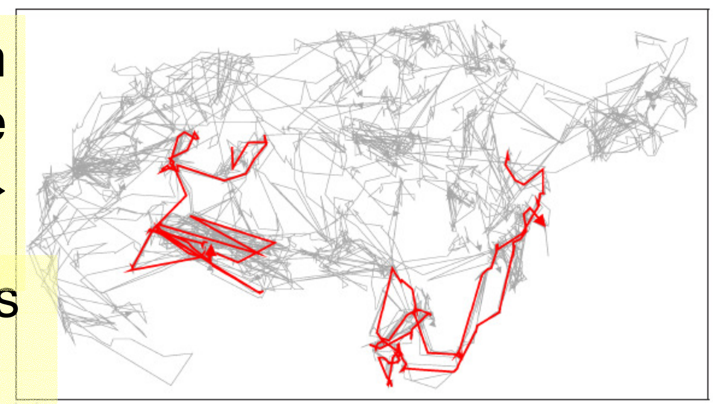
Discovery of Swarm Patterns

- A system that mines moving object patterns: Z. Li, et al., "**MoveMine: Mining Moving Object Databases**", SIGMOD'10 (system demo)
- Z. Li, B. Ding, J. Han, and R. Kays, "**Swarm: Mining Relaxed Temporal Moving Object Clusters**", in submission



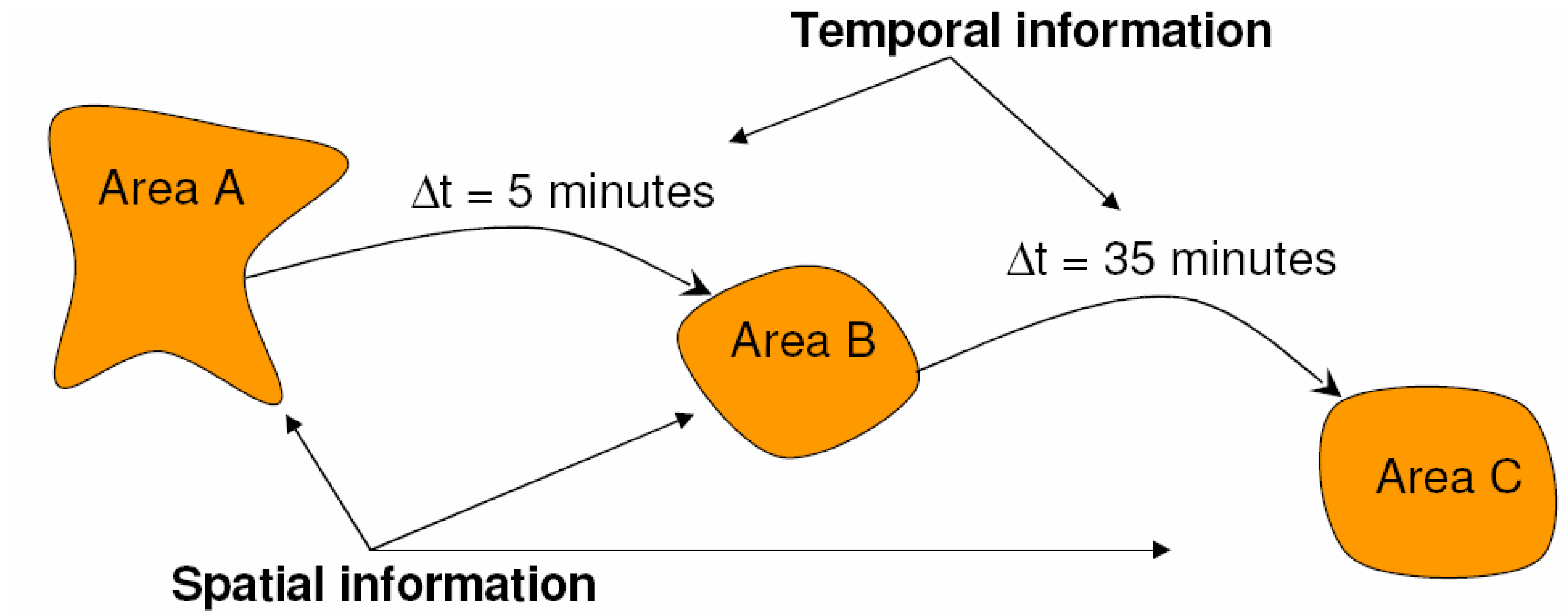
Swarm discovers more patterns →

← **Convoy** discovers only restricted patterns



Trajectory Pattern Mining
(Giannotti et al. KDD 07)

- A trajectory pattern should describe the movements of objects both in space and in time



Trajectory (T-) Patterns: Definition

- A *Trajectory Pattern* (*T-pattern*) is a couple (s,α):
  - $s = <(x_0,y_0),\ldots, (x_k,y_k)>$ is a sequence of k+1 locations
  - $\alpha = <\alpha_1,\ldots, \alpha_k>$ are the transition times (annotations)

  also written as:

$$(x_0,y_0) \xrightarrow{\alpha_1} (x_1,y_1) \xrightarrow{\alpha_2} \ldots\ldots \xrightarrow{\alpha_k} (x_k,y_k)$$

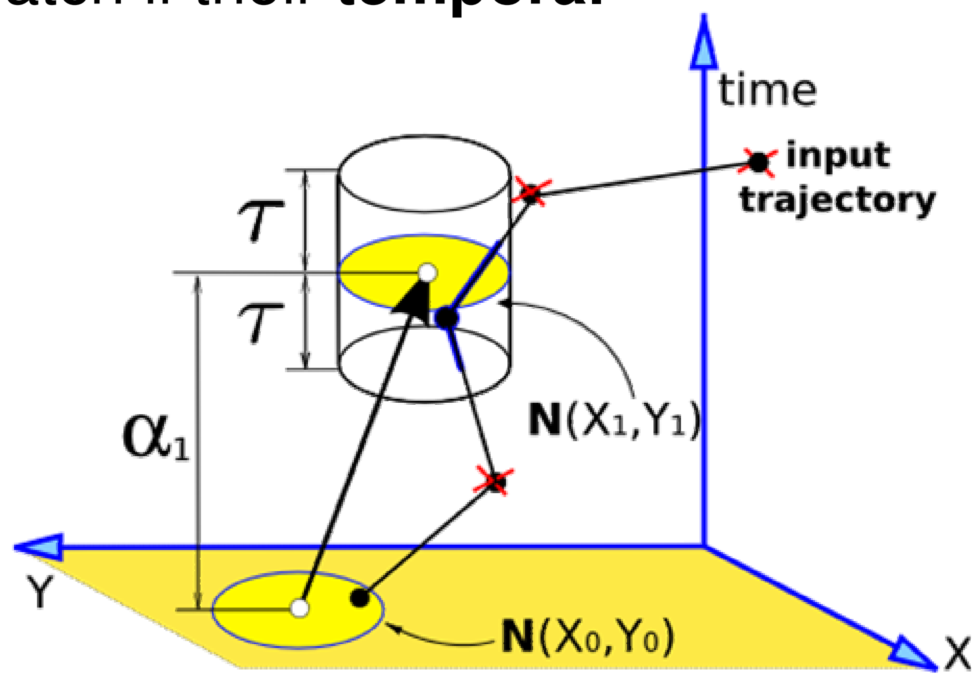
- A T-pattern T_p *occurs* in a trajectory if the trajectory contains a subsequence S such that:
  - Each (x_i,y_i) in T_p matches a point (x_i',y_i') in S, and
  - the transition times in Tp are similar to those in S

T-Pattern: *approximate* occurrence

- Two points match if one falls within a **spatial neighborhood N()** of the other
- Two transition times match if their **temporal difference is ≤ τ**

- Example:

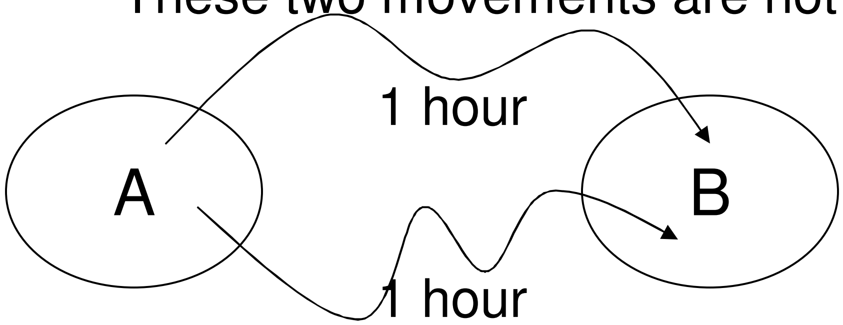
$$(x_0, y_0) \xrightarrow{\alpha_1} (x_1, y_1)$$



Characteristics of Trajectory-Patterns

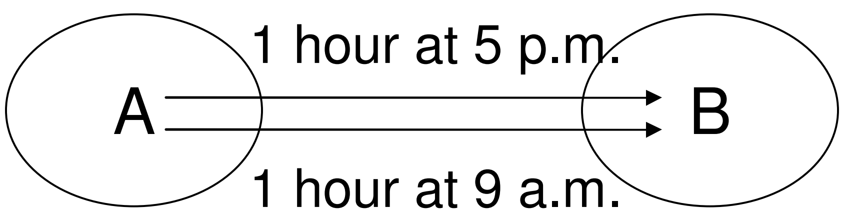
- Routes between two consecutive regions are not relevant

These two movements are not discriminated

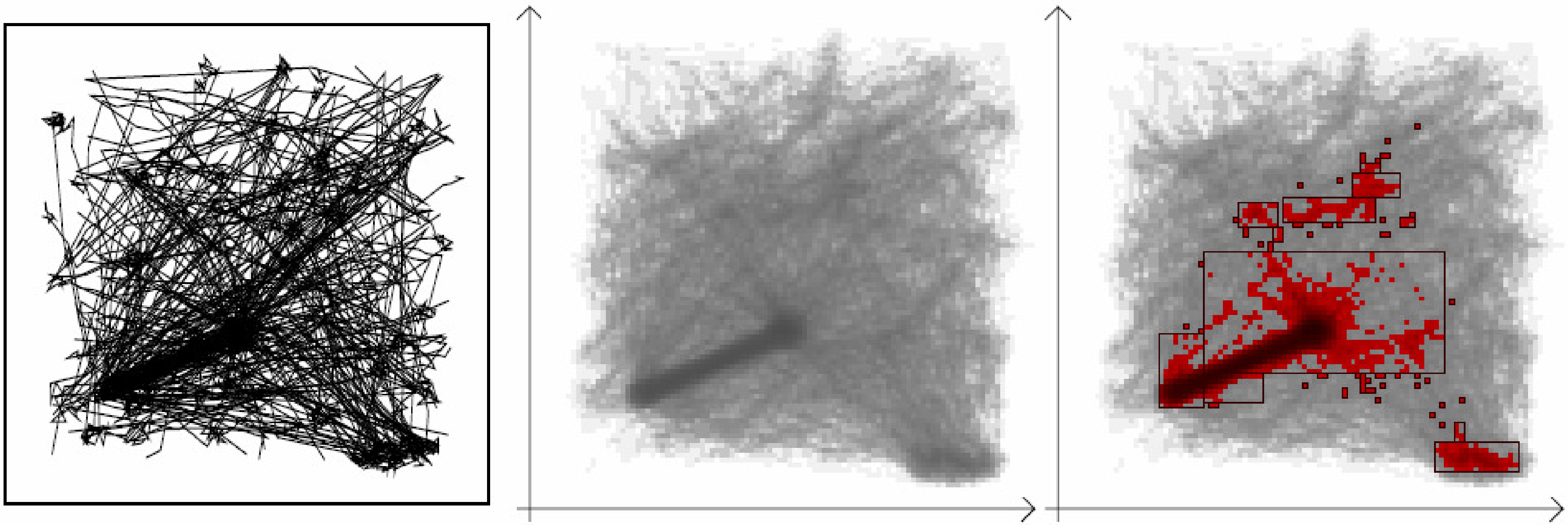


- Absolute times are not relevant

These two movements are not discriminated



Finding regions
A usage-based heuristic

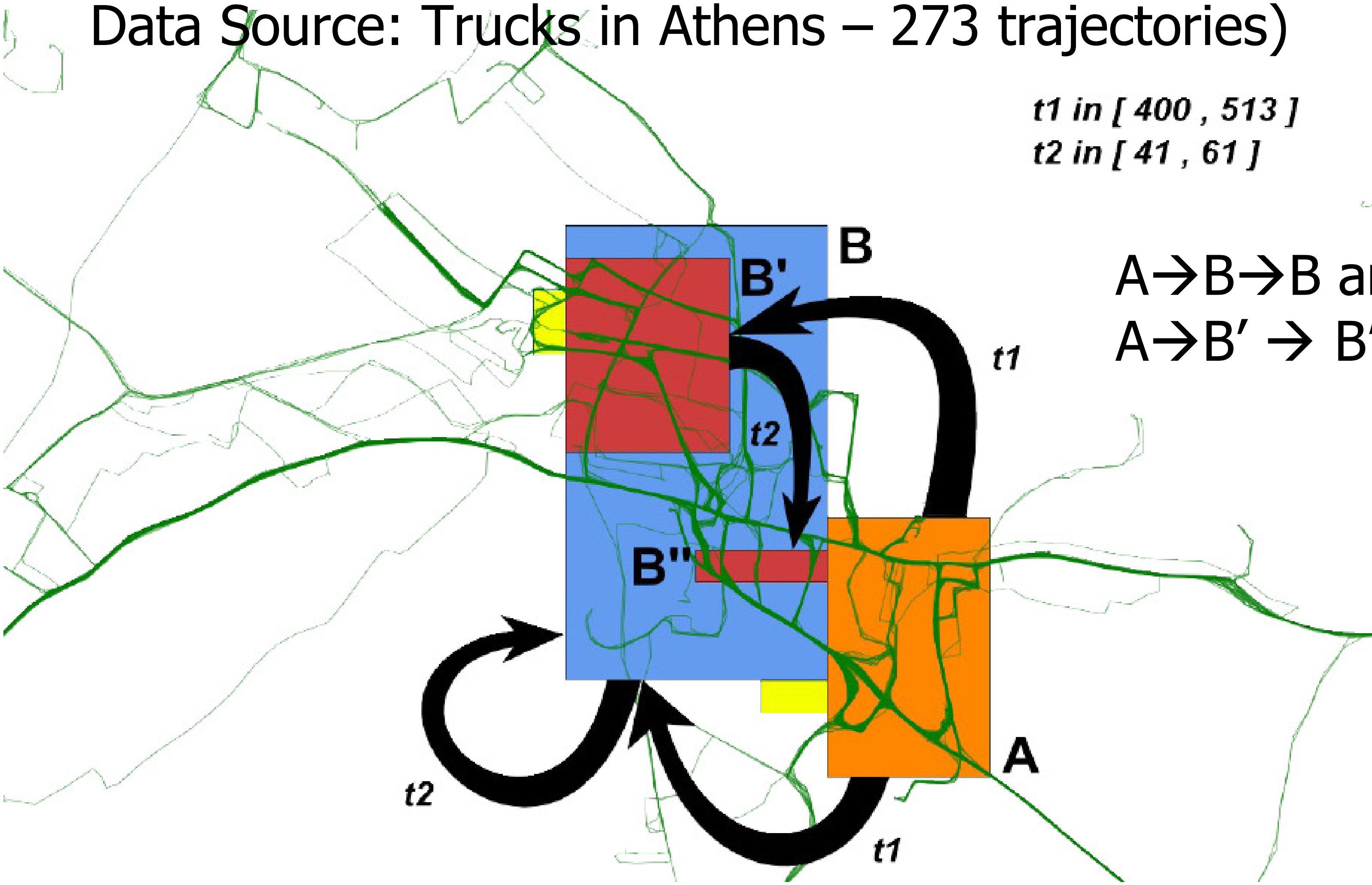


(a) input trajectories (b) density distribution (c) dense cells and extracted RoI

1. Impose a regular grid over space
2. Find dense cells (i.e., touched by many trajs.)
3. Coalesce cells into rectangles of bounded size

Sample Trajectory-Patterns

Data Source: Trucks in Athens – 273 trajectories)



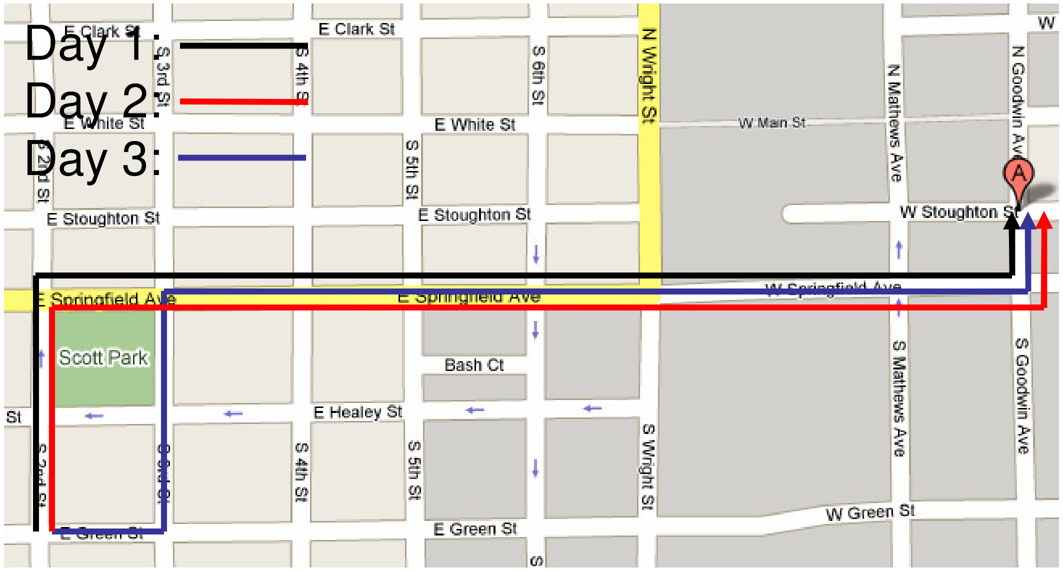
A→B→B and
A→B′ → B″

Mining Moving Object Data

- Introduction
- Movement Pattern Mining
- Periodic Pattern Mining
- Clustering
- Prediction
- Classification
- Outlier Detection

Spatiotemporal Periodic Pattern (Mamoulis et al. KDD 04)

- In many applications, objects follow the same routes (approximately) over regular time intervals
  - e.g., Bob wakes up at the same time and then follows, more or less, the same route to his work everyday



Period and Periodic Pattern

- Let ***S*** be a sequence of *n* spatial locations, $\{l_0, l_1, \ldots, l_{n-1}\}$, representing the movement of an object over a long history
- Let $T \ll n$ be an integer called *period, and T is given*
- A *periodic pattern P* is defined by a sequence $r_0 r_1 \ldots r_{T-1}$ of length T that appears in ***S*** by more than *min_sup* times
  - For every r_i in P, $r_i = *$ or l_{j*T+i} is inside r_i

Periodic Patterns of Moving objects

- Periodic behavior is the intrinsic behavior for most moving objects
  - Yearly migration of birds
    - Fly to south for winter, fly back to north for summer
  - People's daily routines
    - Go to office at 9:00am, back home around 6:00pm
- Detecting periodic behavior is useful for:
  - Summarizing over long historical movement
    - People's behavior could be summarized as some daily behavior and weekly behavior
  - Predicting future movement
    - E.g., predict the location at the *future* time (next day, next week, or next year)
  - Help detect abnormal events
    - A bird does not follow its usual migration path a signal of environment change

Challenges of Periodic Pattern Mining

interleaved periods

Raw data of David's movement

...
2009−02−05 07:01 (601, 254)
2009−02−05 09:14 (811, 60)
2009−02−05 10:58 (810, 55)
2009−02−05 14:29 (820, 100)
...
2009−06−12 09:56 (110, 98)
2009−06−12 11:20 (101, 65)
2009−06−12 20:08 (20, 97)
2009−06−12 22:19 (15, 100)
...

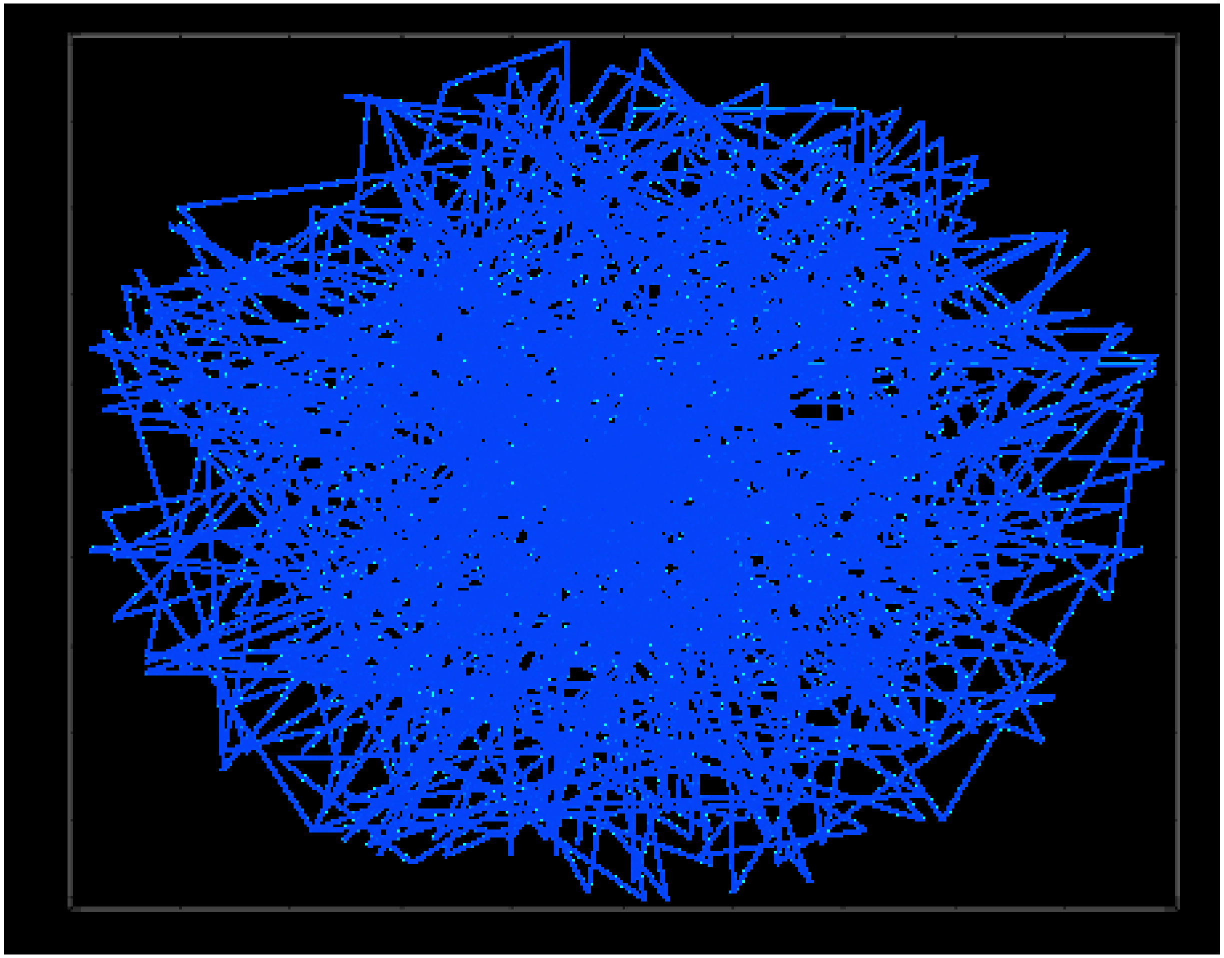
Hidden periodic behaviors

- Periodic Behavior #1
  (Period: day; Time span: Sept. − May)
  9:00−18:00 in the office
  20:00−8:00 in the dorm
- Periodic Behavior #2
  (Period: day; Time span: June − Aug.)
  8:00−18:00 in the company
  20:00−7:30 in the apartment
- Periodic Behavior #3
  (Period: week; Time span: Sept. − May)
  13:00−15:00 Mon. and Wed. in the classroom
  14:00−16:00 Tues. and Thurs. in the gym

multiple periods

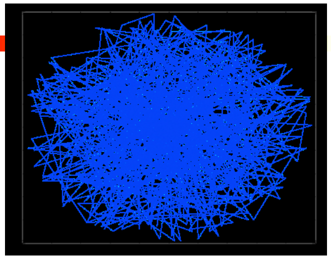
different locations

A Motivating Example: Trajectories of Bees



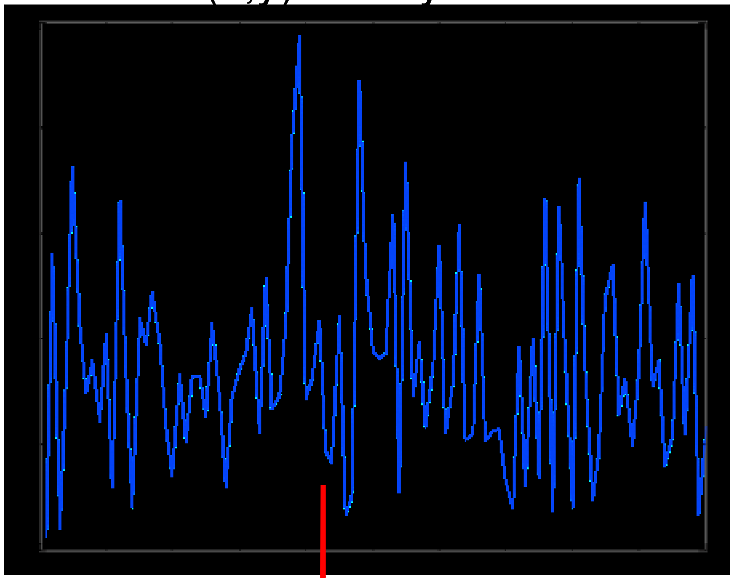
Bee and Flower:
8 hours stays in the nest
16 hours fly nearby

FFT Transformation Does Not Work

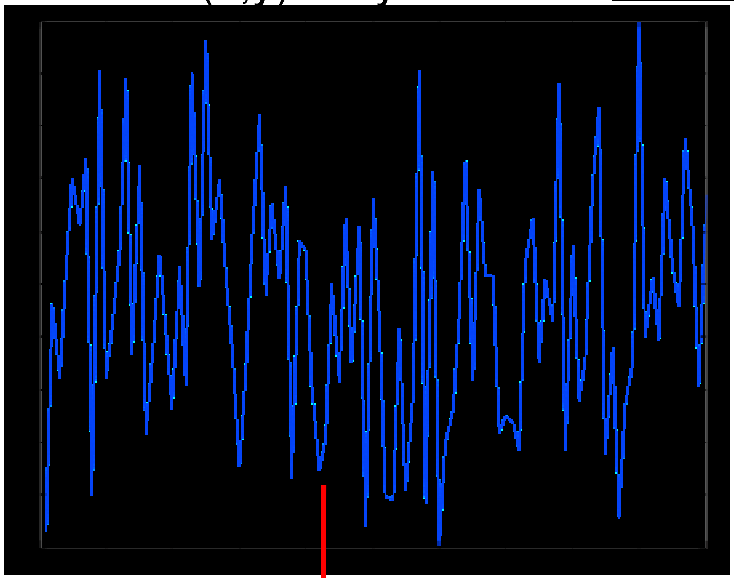


Transform (x,y) into complex plane (two ways to transform)

(x,y) => x-yi

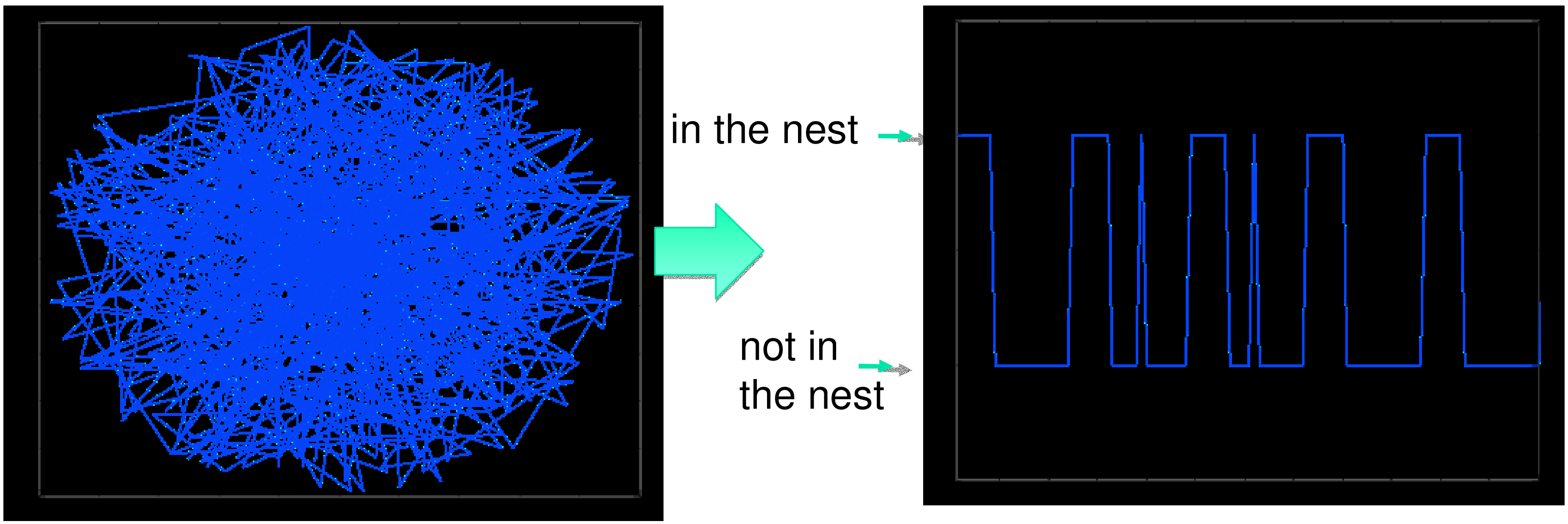


(x,y) => y-xi



FFT should have strongest power at **42.7** (T = 24, NFFT/T = 1024/24 = 42.7)
Failed!

Observation/Reference Spot: The Nest



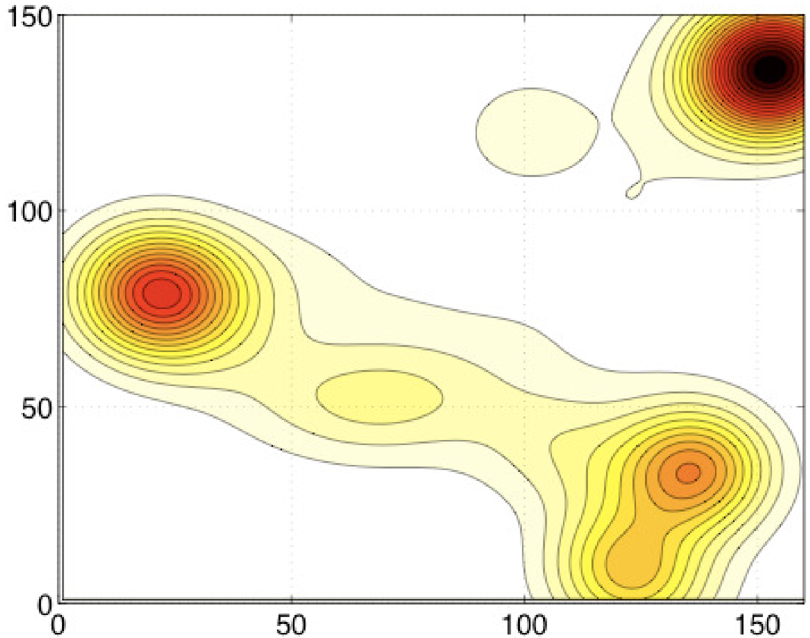
Period is more obvious in this binary sequence!

Algorithm General Framework

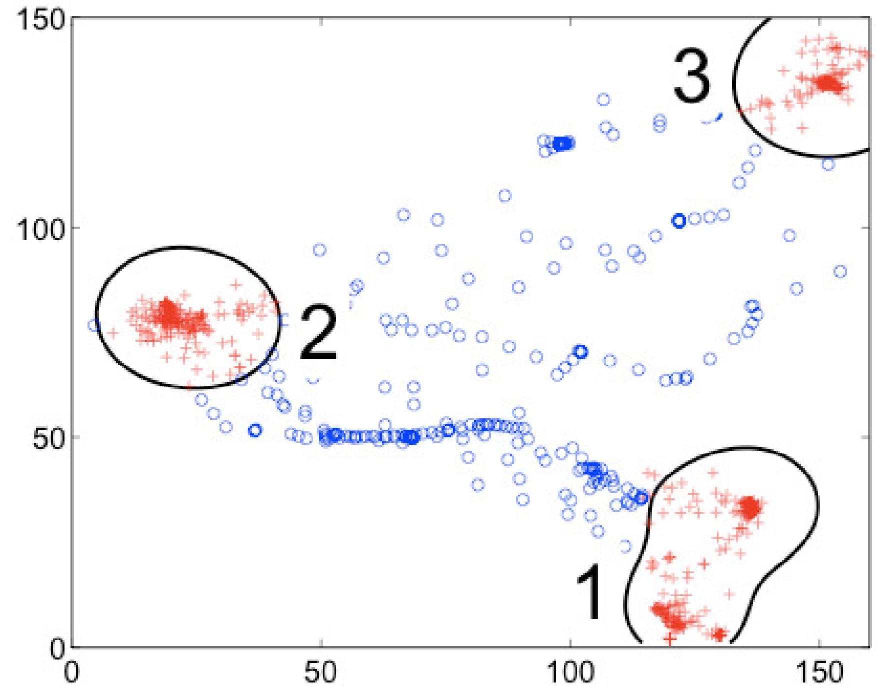
- **Detecting periods**: Use observation spots to find multiple interleaved periods
  - Observation spots are detected using **density-based method**
  - Periods are detected for each obs. spot using **Fourier Transform and auto-correlation**

- **Summarizing periodic behaviors:** via **clustering**
  - Give the statistical explanation of the behavior
  - E.g., “David has 80% probability to be at the office.”

Example: Finding Observation Spots



Density



Observation spots

Mining Moving Object Data

- Introduction
- Movement Pattern Mining
- Periodic Pattern Mining
- Clustering
- Prediction
- Classification
- Outlier Detection

Clustering: Distance-Based vs. Shape-Based

- Distance-based clustering: Find a group of objects moving together
  - For whole time span
    - high-dimensional clustering
    - probabilistic clustering
  - For partial continuous time span
    - density-based clustering
    - moving cluster, flock, convoy *(borderline case between clustering and patterns)*
  - For partial discrete time span
    - swarm *(borderline case between clustering and patterns)*
- Shape-based clustering: Find similar shape trajectories
  - Variants of shape: translation, rotation, scaling, and transformation
  - Sub-trajectory clustering

High-Dimensional Clustering & Distance Measures

- Treat each timestamp as one dimension
- Many high-dimensional clustering methods can be applied to cluster moving objects
- Most popular high-dimensional distance measure
  - Euclidean distance
  - Dynamic Time Warping
  - Longest Common Subsequence
  - Edit Distance with Real Penalty
  - Edit Distance on Real Sequence

High-Dimensional Distance Measures

Distance Measure	Local Time Shifting	Noise	Metric	Complexity
Euclidean			□	O(n)
DTW (Yi et al., ICDE'98)	□			O(n^2)
LCSS (Vlachos et al., KDD'03)	□	□		O(n^2)
ERP (Chen et al., VLDB'04)	□		□	O(n^2)
EDR (Chen et al., SIGMOD'05)	□	□ (consider gap)		O(n^2)

Probabilistic Trajectory Clustering

(Gaffney et al., KDD'00; Chudova et al., KDD'03)

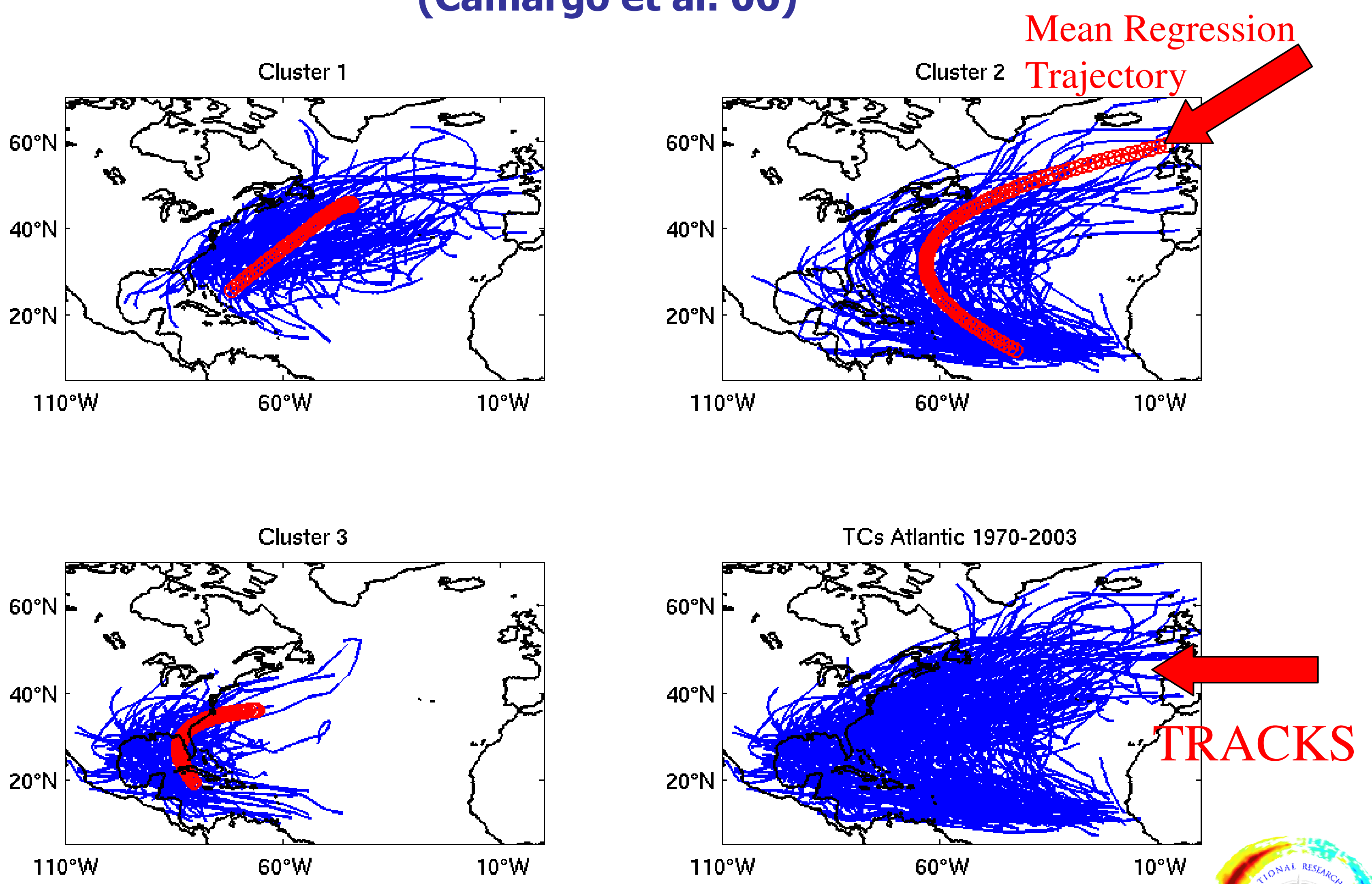
- Basic assumption: Data produced in the following ***generative*** manner
  - An individual is drawn randomly from the population of interest
  - The individual has been assigned to a cluster k with probability w_k, $\sum_{k=1}^{K} w_k = 1$, these are the *prior* weights on the K clusters
  - Given that an individual belongs to a cluster k, there is a density function $f_k(y_j \mid \theta_k)$ which generates an observed data item y_j for the individual j
- The probability density function of observed trajectories is a mixture density

$$P(y_j \mid x_j, \theta) = \sum_{k}^{K} f_k(y_j \mid x_j, \theta_k) w_k$$

  - $f_k(y_j \mid x_j, \theta_k)$ is the density component
  - w_k is the weight, and θ_k is the set of parameters for the k-th component
- θ_k and w_k can be estimated from the trajectory data using the *Expectation-Maximization* (*EM*) algorithm

Clustering Results For Hurricanes

(Camargo et al. 06)

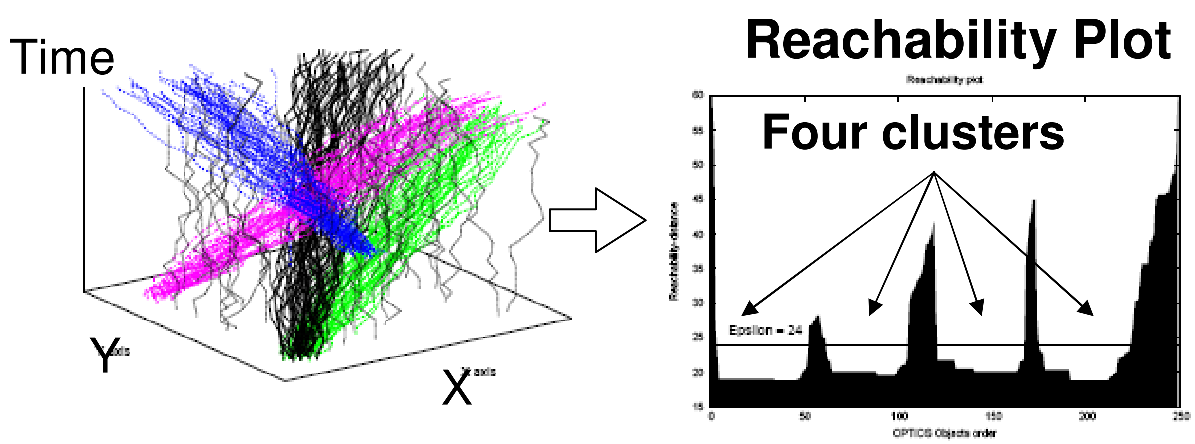


Tracks Atlantic named Tropical Cyclones 1970-2003.

Density-Based Trajectory Clustering

(M. Nanni & D. Pedreschi, JIIS'06)

- Define the distance between *whole* trajectories
  - A trajectory is represented as a sequence of location and timestamp
  - The distance between trajectories is the average distance between objects for every timestamp
- Use the OPTICS algorithm for trajectories
  - e.g.,



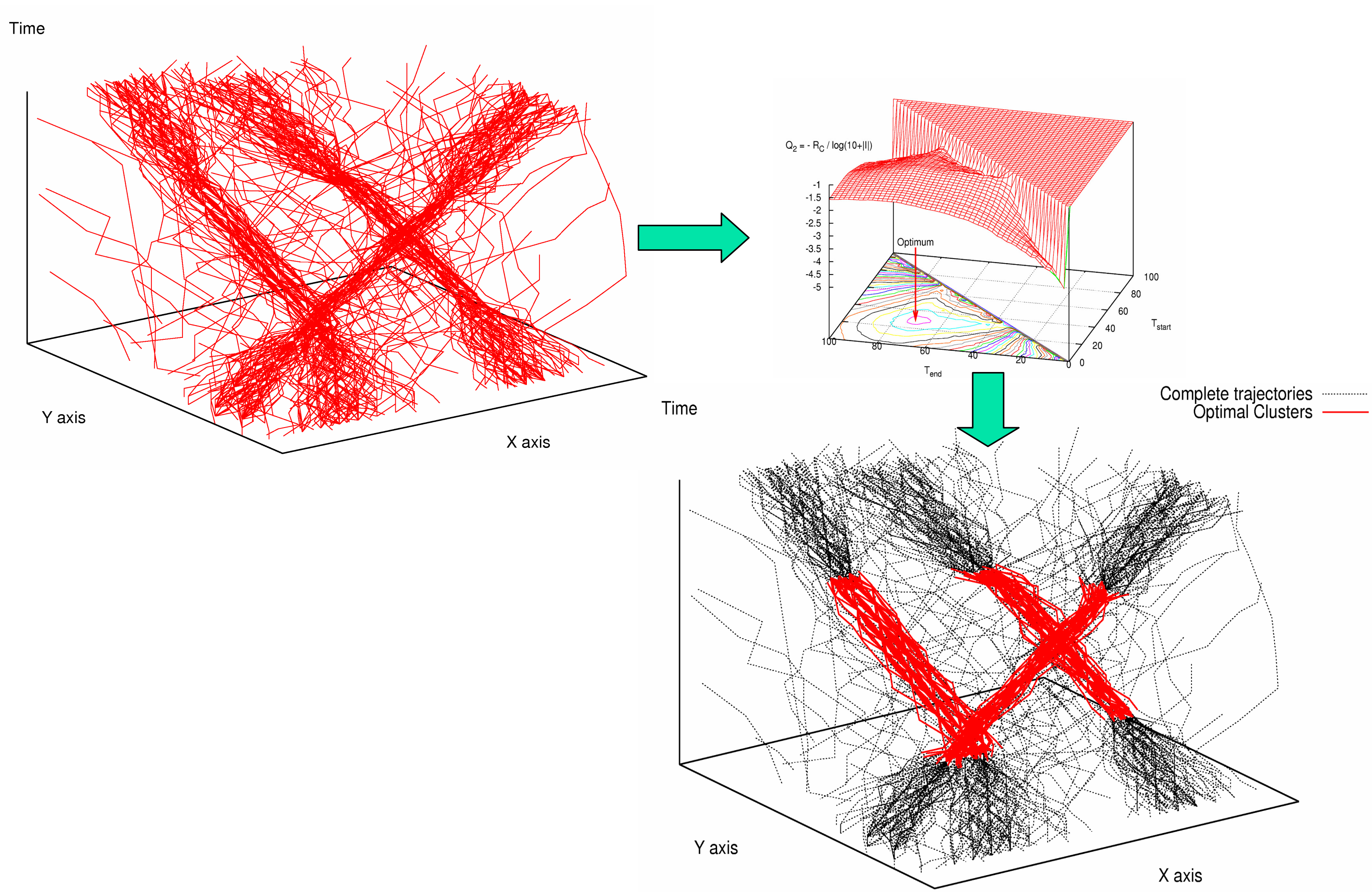
Temporal Focusing: TF-OPTICS
(M. Nanni & D. Pedreschi, JIIS'06)

- In a real environment, not all time intervals have the same importance
  - e.g., *in rush hours*, many people move from home to work or vice versa
- *TF-OPTICS* aims at **searching the most meaningful time intervals**, which allows us to isolate the clusters of higher quality
- Method:
  - Define the quality of a clustering
    - Take account of both high-density clusters and low-density noise
    - Can be computed directly from the reachability plot
  - Find the time interval that maximizes the quality

1. Choose an initial random time interval
2. Calculate the quality of neighborhood intervals generated by increasing or decreasing the starting or ending times
3. Repeat Step 2 as long as the quality increases

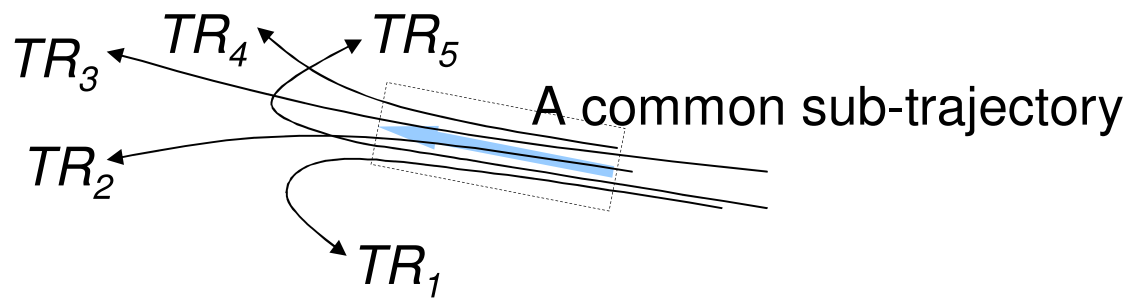
Temporal Focusing: TF-OPTICS

(M. Nanni & D. Pedreschi, JIIS'06)



Trajectory Clustering: A Partition-and-Group Framework (Lee et al., SIGMOD'07)

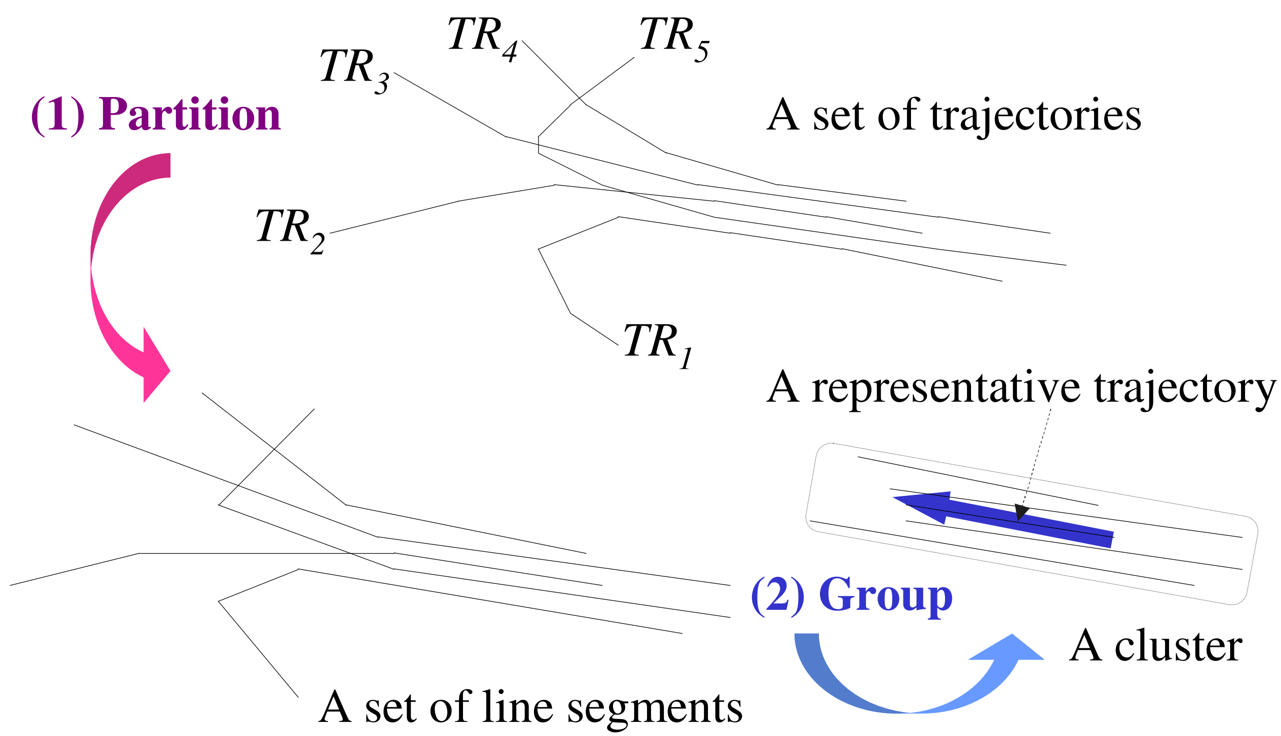
- Existing algorithms group trajectories ***as a whole*** □ They might not be able to find ***similar portions*** of trajectories
  - e.g., common behavior cannot be discovered since TR_1~TR_5 move to totally different directions



- ***Partition-and-group:*** discovers common ***sub***-trajectories
- Usage: Discover ***regions of special interest***
  - *Hurricane Landfall Forecasts:* Discovery of common behaviors of hurricanes ***near the coastline*** or ***at sea*** (*i.e.*, before landing)
  - *Effects of Roads and Traffic on Animal Movements: Discover* common behaviors of animals ***near the road***

Partition-and-Group: Overall Procedure

- Two phases: ***partitioning*** and ***grouping***

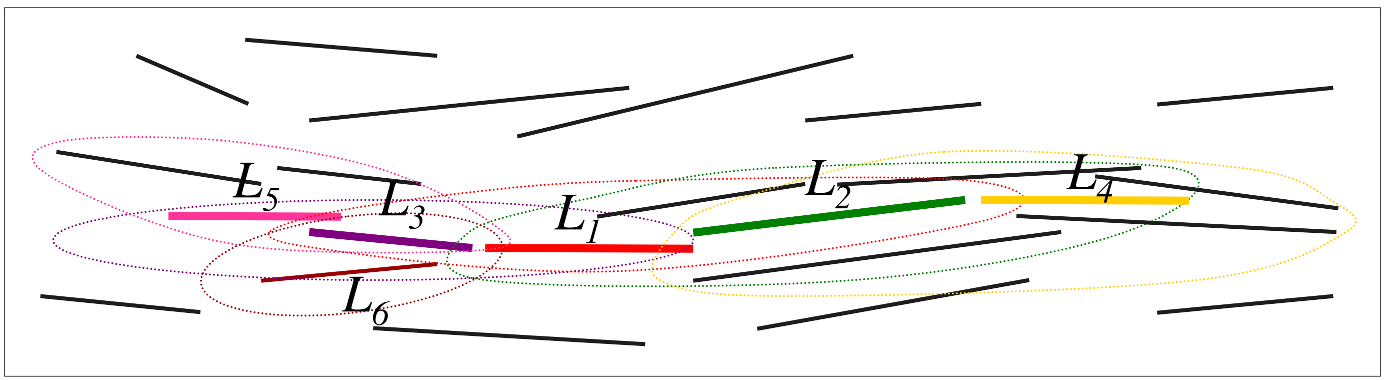


Note: A representative trajectory is a common sub-trajectory

Grouping Phase (1/2)

- Find the clusters of trajectory partitions using density-based clustering (*i.e.*, DBSCAN)
  - A density-connect component forms a cluster, *e.g.*, { L_1, L_2, L_3, L_4, L_5, L_6 }

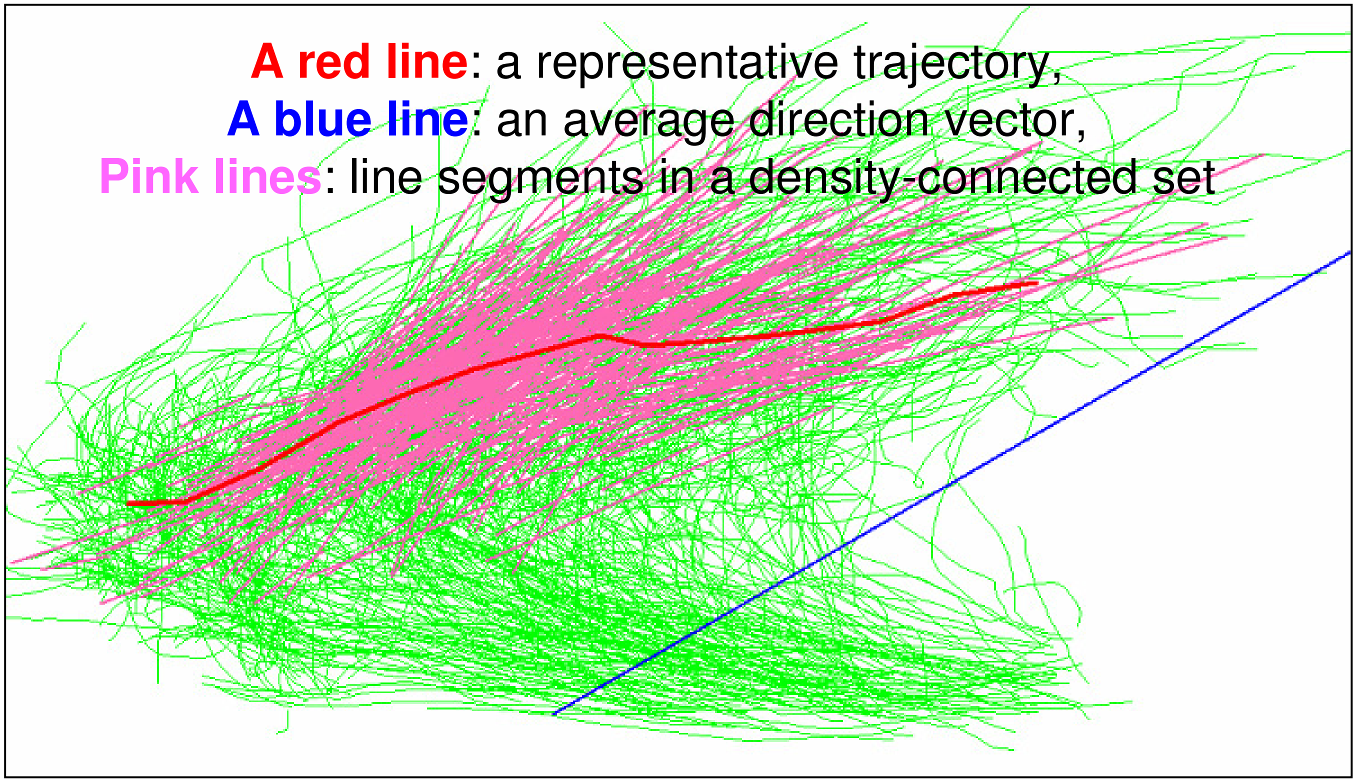
$MinLns = 3$



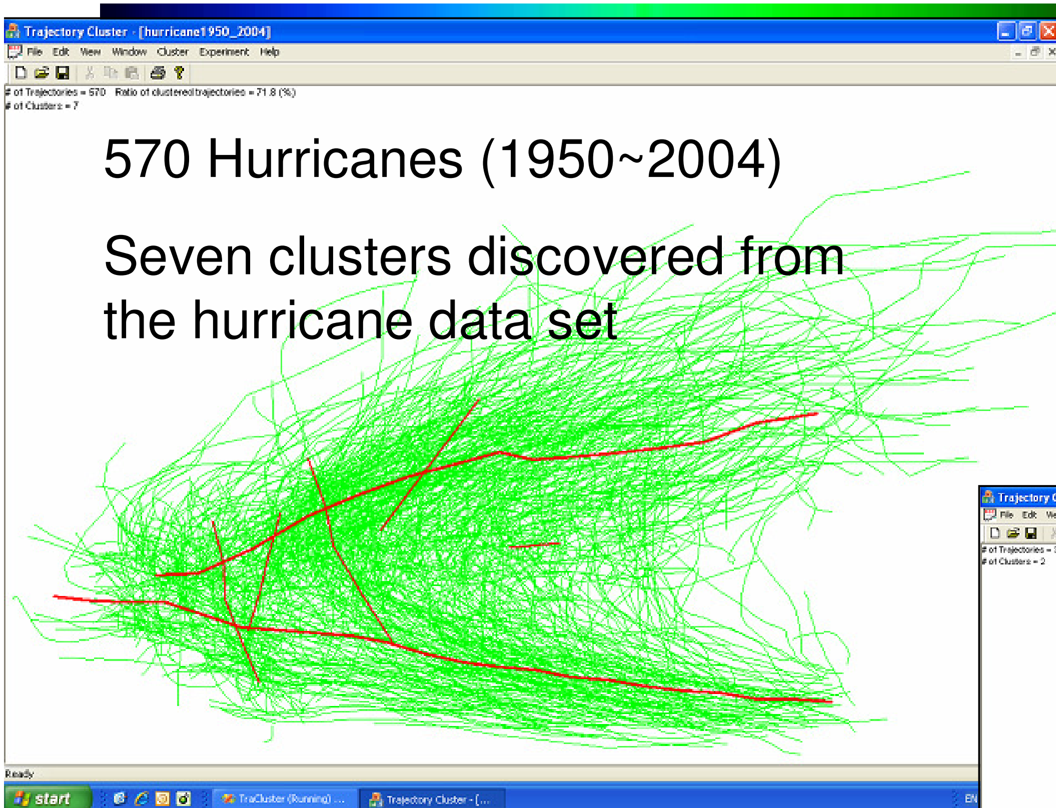
$L_6 \quad L_5 — L_3 — L_1 — L_2 — L_4$

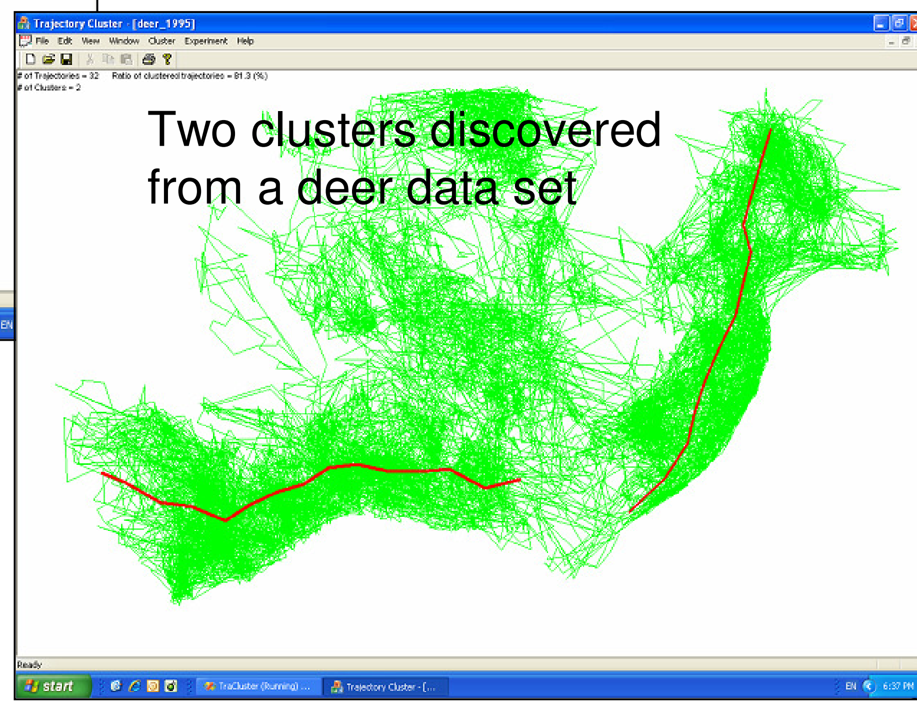
Grouping Phase (2/2)

- Describe the overall movement of the trajectory partitions that belong to the cluster



Example: Trajectory Clustering Results





Red line: a representative trajectory

Mining Moving Object Data

- Introduction
- Movement Pattern Mining
- Periodic Pattern Mining
- Clustering
- Prediction
- Classification
- Outlier Detection

Location Prediction for Moving Objects

- Predicting future location
  - Based on its own history of one moving object
    - Linear (not practical) vs. non-linear motion (more practical)
    - Vector based (predict near time, e.g., next minute) vs. pattern based (predict distant time, e.g., next month/year)
  - Based on all moving objects' trajectories
    - based on frequent patterns

Recursive Motion Function
(Tao et al., SIGMOD'04)

- Non-linear model, near time prediction, vector-based method
- Linear model is not practical in prediction, so better to use non-linear model
- Recursive motion function

  $$\boldsymbol{o}(t) = \mathbf{C}_1 \cdot \boldsymbol{o}(t-1) + \mathbf{C}_2 \cdot \boldsymbol{o}(t-2) + \ldots + \mathbf{C}_f \cdot \boldsymbol{o}(t-f)$$

C_i is a constant matrix expressing several complex movement types, including polynomials, ellipse, sinusoids, etc.

- Use basic motion matrices to model unknown motion matrices

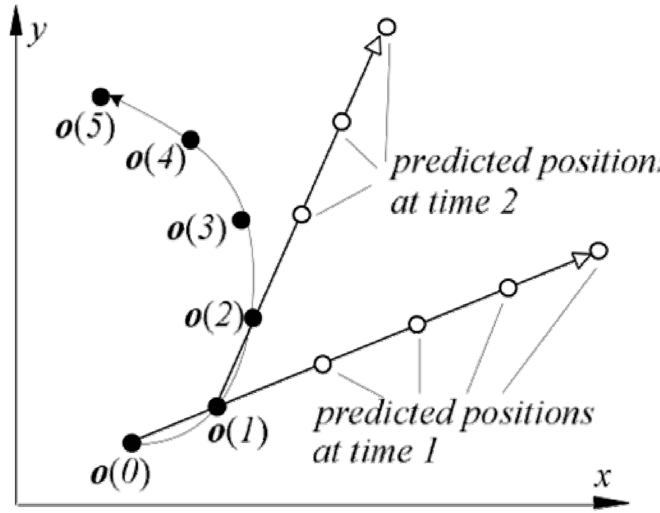


Figure 1.1: Failure of linear prediction

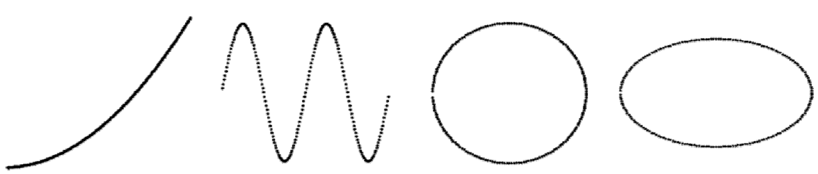


Figure 6.1: Movements with known motion matrices

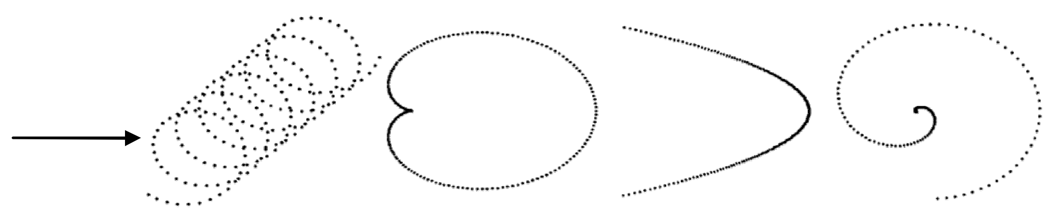
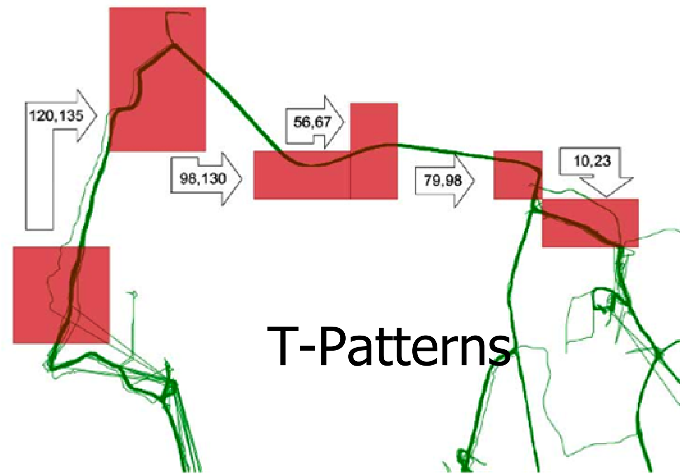


Figure 6.3: Movements with unknown motion matrices

Prediction Using Frequent Trajectory Patterns (Monreale et al., KDD'09)

- Use frequent T-patterns of other moving objects
- If many moving objects follow a pattern, it is likely that a moving object will also follow this pattern
- Method
  - Mine T-Patterns
  - Construct T-Pattern Tree
  - Predict using T-pattern tree



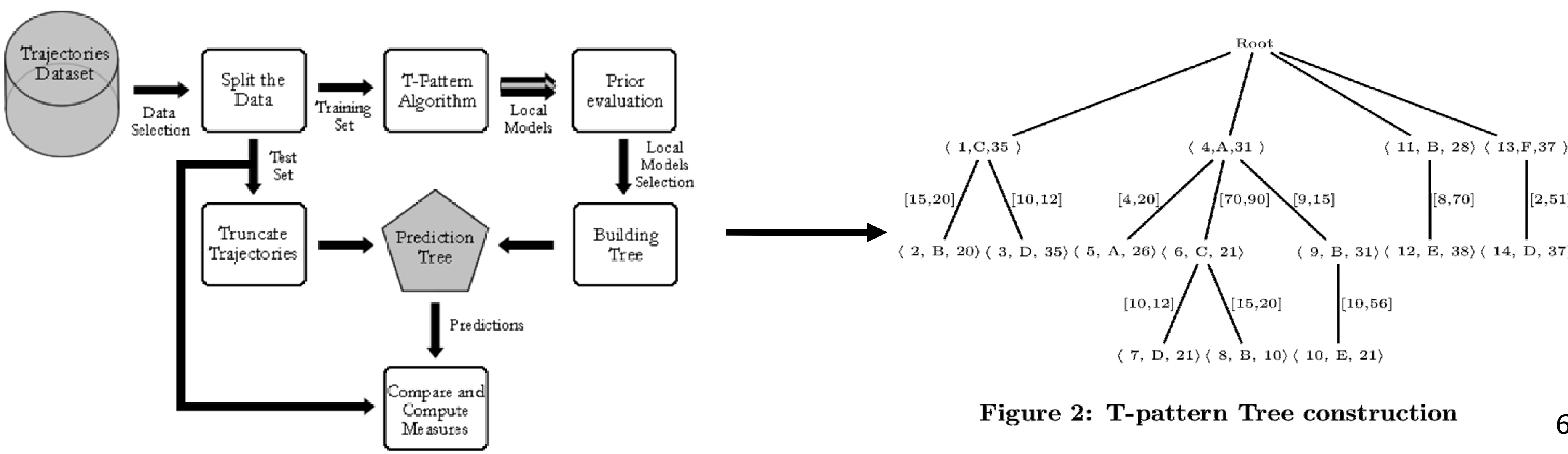


Figure 2: T-pattern Tree construction

Mining Moving Object Data

- Introduction
- Movement Pattern Mining
- Periodic Pattern Mining
- Clustering
- Prediction
- Classification
- Outlier Detection

Trajectory Classification

- Task: Predict the class labels of moving objects based on their trajectories and other features
- Two approaches
  - Machine learning techniques
    - Studied mostly in pattern recognition, bioengineering, and video surveillance
    - The hidden Markov model (HMM)
  - Trajectory-based classification (**TraClass**): Trajectory classification using hierarchical region-based and trajectory-based clustering

Vehicle Trajectory Classification

(Fraile and Maybank 98)

- The measurement sequence is divided into overlapping segments
- In each segment, the trajectory of the car is approximated by a smooth function and then assigned to one of four categories: *ahead*, *left*, *right*, or *stop*
- The list of segments is reduced to a string of symbols drawn from the set {*a*, *l*, *r*, *s*}
- The string of symbols is classified using the hidden Markov model (HMM)

Motion Trajectory Classification
(Bashir et al. 07)

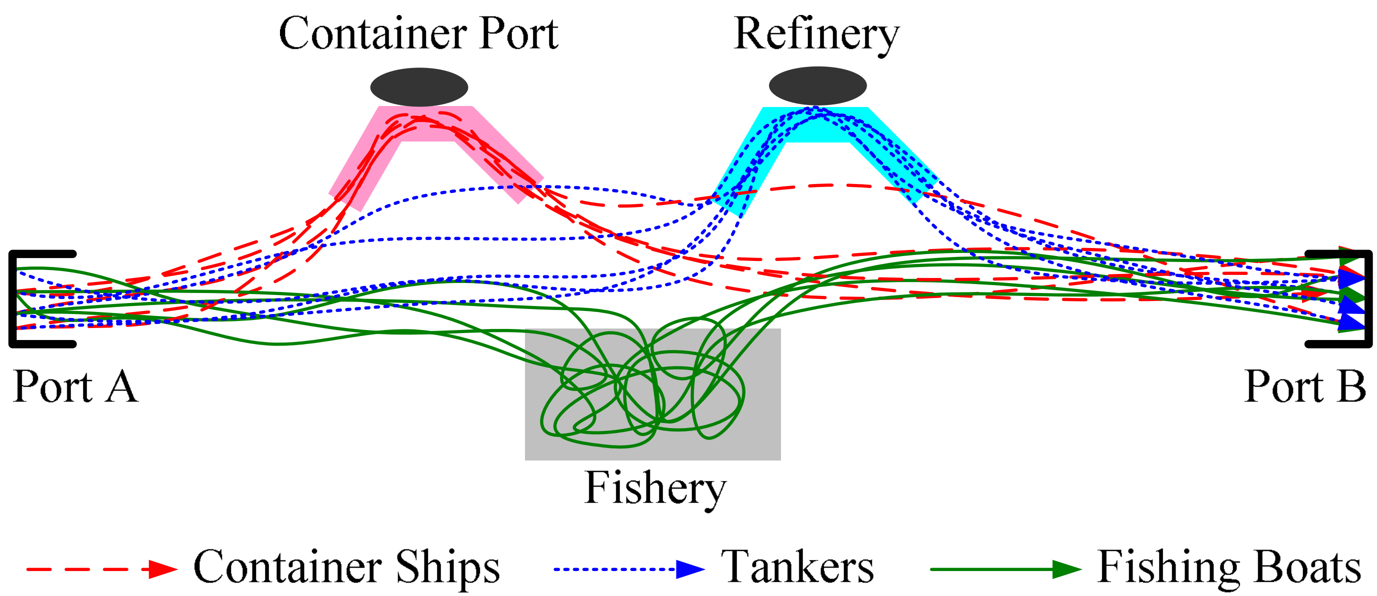
- Motion trajectories
  - Tracking results from video trackers, sign language data measurements gathered from wired glove interfaces, and so on
- Application scenarios
  - Sport video (*e.g.*, soccer video) analysis
    - Player movements → A strategy
  - Sign and gesture recognition
    - Hand movements ⟶ A particular word
- The HMM-Based Algorithm

1. Trajectories are segmented at points of change in curvature
2. Sub-trajectories are represented by their Principal Component Analysis (PCA) coefficients
3. The PCA coefficients are represented using a GMM for each class
4. An HMM is built for each class, where the state of the HMM is a sub-trajectory and is modeled by a mixture of Gaussians

TraClass: Trajectory Classification Based on Clustering

- Motivation
  - Discriminative features are likely to appear at ***parts*** of trajectories, not at whole trajectories
  - Discriminative features appear not only as common movement patterns, but also as ***regions***
- Solution
  - Extract features in a top-down fashion, first by ***region-based clustering*** and then by ***trajectory-based clustering***

Intuition and Working Example



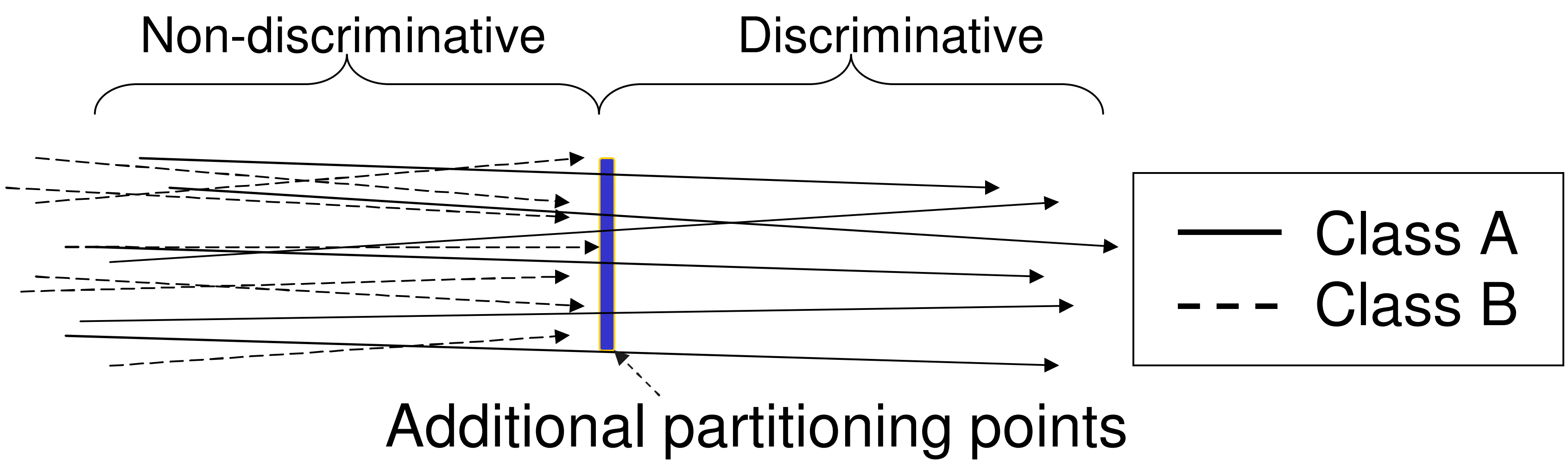
- Parts of trajectories near the container port and near the refinery enable us to distinguish between container ships and tankers even if they share common long paths
- Those in the fishery enable us to recognize fishing boats even if they have no common path there

Class-Conscious Trajectory Partitioning

1. Trajectories are partitioned based on their shapes as in the partition-and-group framework

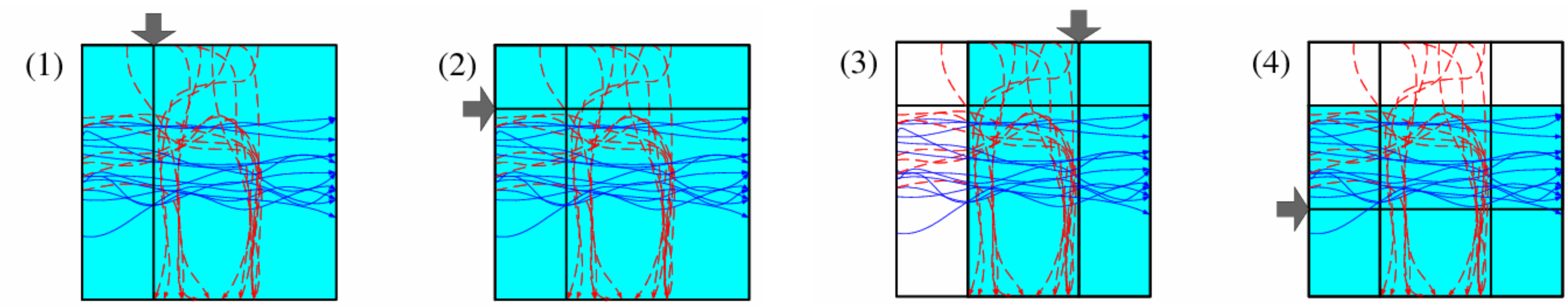
2. Trajectory partitions are further partitioned by ***the class labels***

- The real interest here is to guarantee that trajectory partitions do not span the class boundaries



Region-Based Clustering

- Objective: Discover regions that have trajectories mostly of one class regardless of their movement patterns



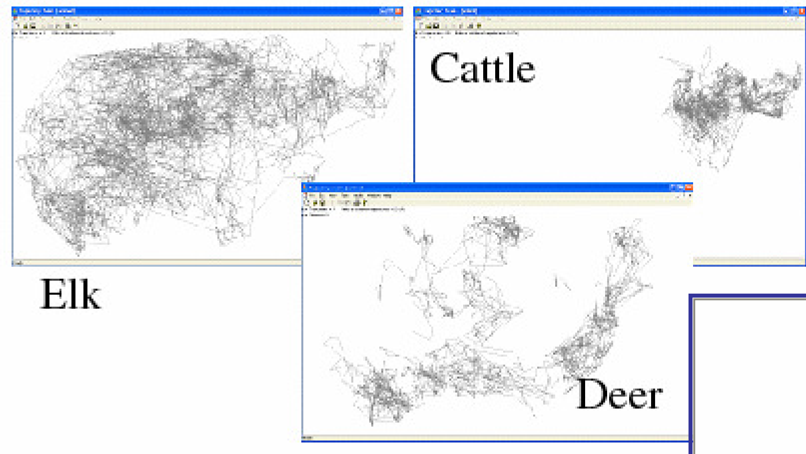
Trajectory-Based Clustering

- Objective: Discover sub-trajectories that indicate common movement patterns of each class
- Algorithm: Extend the partition-and-group framework for classification purposes so that the class labels are incorporated into trajectory clustering
  - If an ε-neighborhood contains trajectory partitions mostly of the same class, it is used for clustering; otherwise, it is discarded immediately

Overall Procedure of TraClass

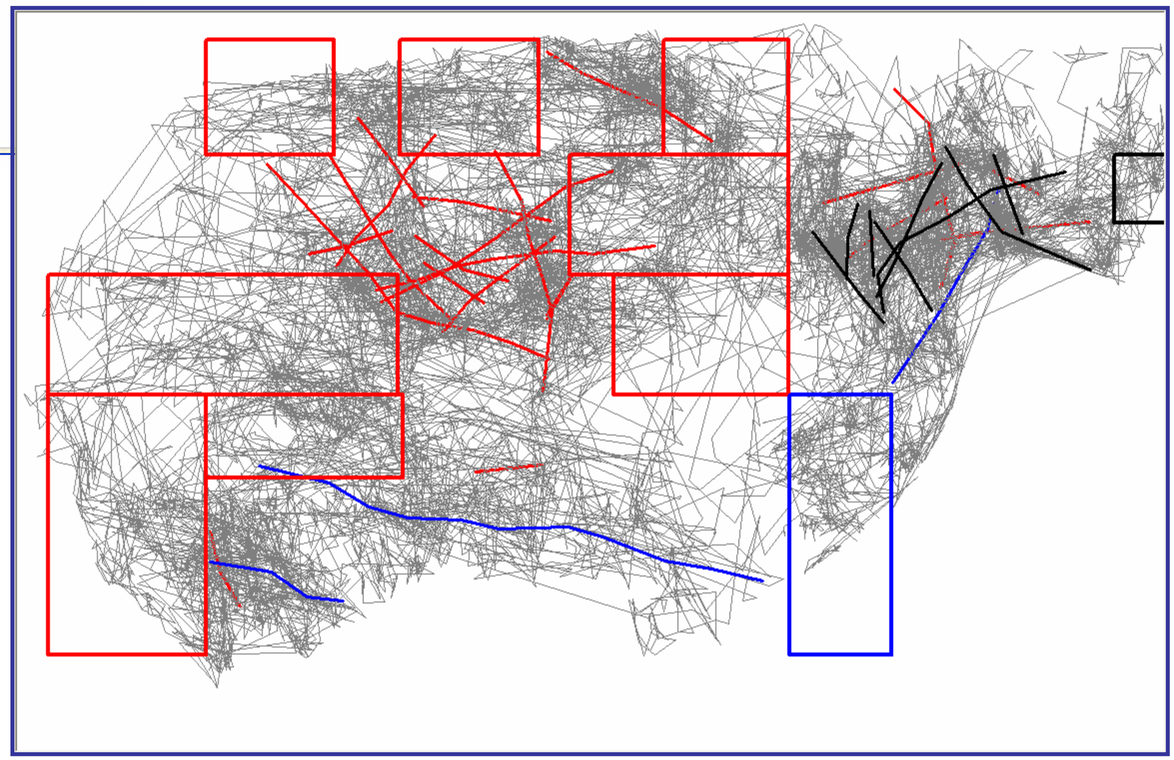
1. Partition trajectories
2. Perform region-based clustering
3. Perform trajectory-based clustering
4. Select discriminative trajectory-based clusters
5. Convert each trajectory into a feature vector
   - Each feature is either a region-based cluster or a trajectory-based cluster
   - The *i*-th entry of a feature vector is the frequency that the *i*-th feature occurs in the trajectory
6. Feed feature vectors to the SVM

Example: Extracted Features



Data (Three Classes)

Features:
10 Region-Based Clusters
37 Trajectory-Based Clusters



Accuracy = 83.3%

Mining Moving Object Data

- Introduction
- Movement Pattern Mining
- Periodic Pattern Mining
- Clustering
- Prediction
- Classification
- Outlier Detection

Trajectory Outlier Detection

- Task: Detect the trajectory outliers that are grossly different from or inconsistent with the remaining set of trajectories
- Methods and philosophy:

1. ***Whole*** trajectory outlier detection
    - A unsupervised method
    - A supervised method *based on classification*
2. Integration with multi-dimensional information
3. ***Partial*** trajectory outlier detection
    - A Partition-and-Detect framework

Outlier Detection: A Distance-Based Approach (Knorr et al. VLDBJ00)

- Define the distance between two *whole* trajectories
  - A whole trajectory is represented by

$$P = \begin{bmatrix} P_{start} \\ P_{end} \\ P_{heading} \\ P_{velocity} \end{bmatrix} \quad \text{where} \quad \begin{aligned} P_{start} &= (x_{start}, y_{start}) \\ P_{end} &= (x_{end}, y_{end}) \\ P_{heading} &= (avg_{heading}, max_{heading}, min_{heading}) \\ P_{velocity} &= (avg_{velocity}, max_{velocity}, min_{velocity}) \end{aligned}$$

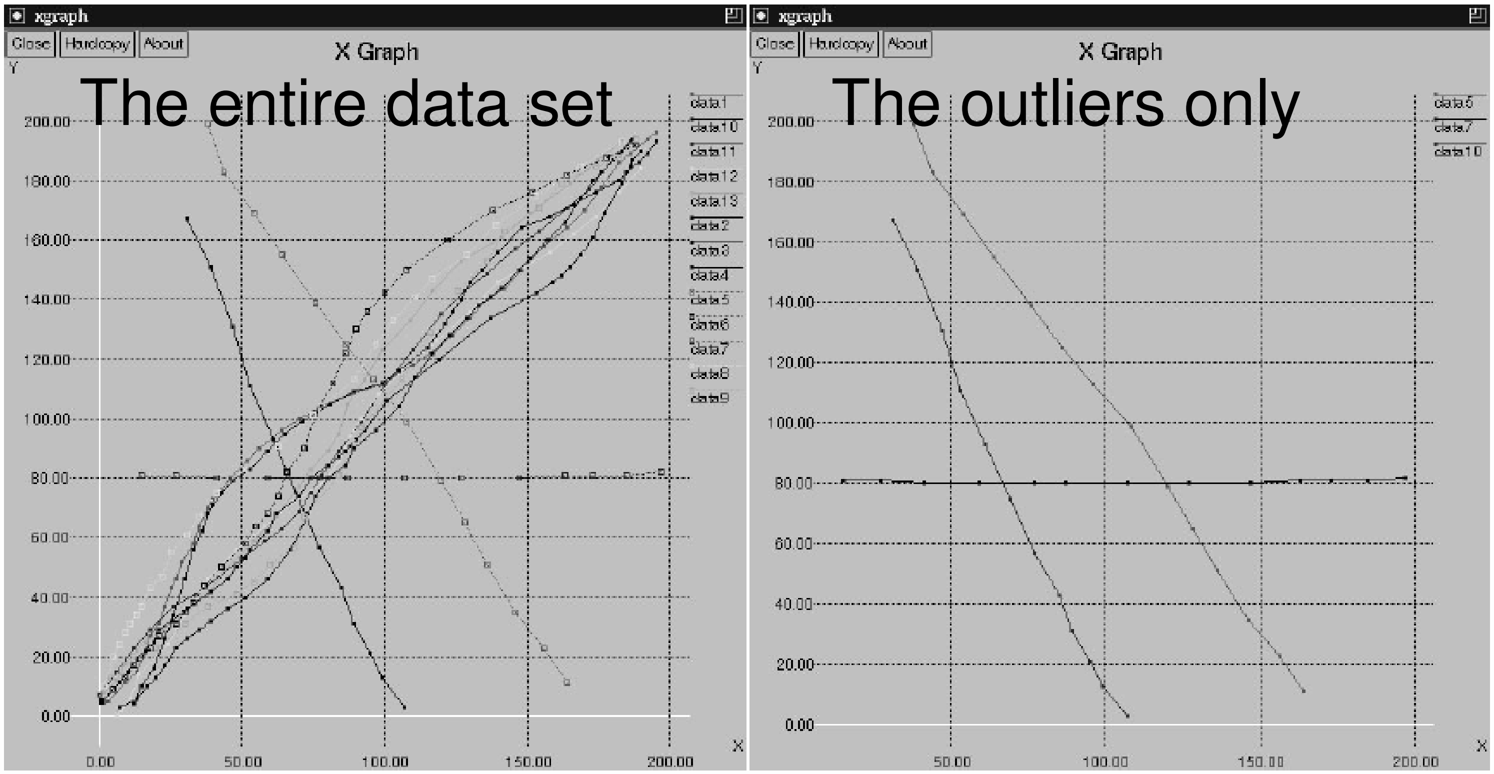
  - The distance between two whole trajectories is defined as

$$D(P_1, P_2) = \begin{bmatrix} D_{start}(P_1, P_2) \\ D_{end}(P_1, P_2) \\ D_{heading}(P_1, P_2) \\ D_{velocity}(P_1, P_2) \end{bmatrix} \cdot \begin{bmatrix} w_{start} & w_{end} & w_{heading} & w_{velocity} \end{bmatrix}$$

- Apply a distance-based approach to detection of trajectory outliers
  - An object *O* in a dataset *T* is a DB(*p*, *D*)-outlier if at least fraction *p* of the objects in *T* lies greater than distance *D* from *O*

Sample Trajectory Outliers

- Detect outliers from person trajectories in a room



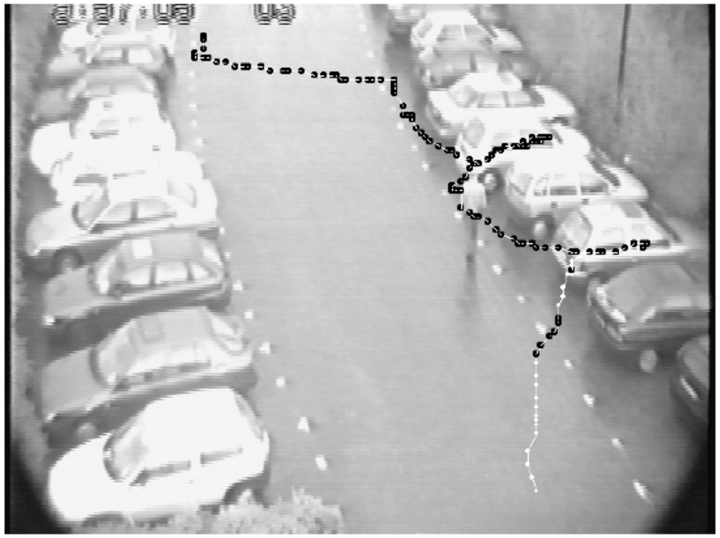
Use of Neural Networks (Owens and Hunter 00)

- A *whole* trajectory is encoded to a *feature vector*: **F** = [x, y, $s(x)$, $s(y)$, $s(dx)$, $s(dy)$, $s(|d^2x|)$, $s(|d^2y|)$]
  - $s()$ indicates a time smoothed average of the quantity
  - $dx = x_t - x_{t-1}$
  - $d^2x = x_t - 2x_{t-1} + x_{t-2}$
- A self-organizing feature map (SOFM) is trained using the feature vectors of training trajectories, and a new trajectory is classified into novel (i.e., suspicious) or not novel
- ***Supervised learning***

An Application: Video Surveillance

- Training dataset: 206 normal trajectories
- Test dataset: 23 unusual and 16 normal trajectories
- Classification accuracy: 92%

An example of a normal trajectory



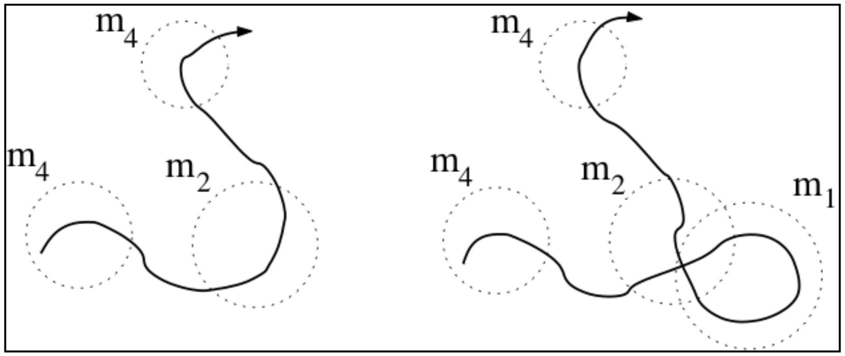
An unusual trajectory; The unusual points are shown in black

Anomaly Detection (Li et al. ISI’06, SSTD’07)

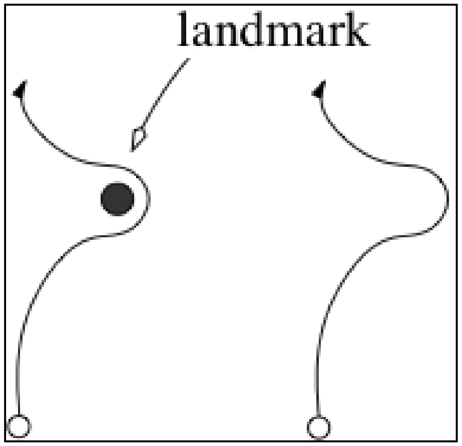
- Automated alerts of abnormal moving objects
- Current US Navy model: manual inspection
  - Started in the 1980s
  - 160,000 ships

Conditional Anomalies and Motif Representations

- Raw analysis of collected data does not fully convey "anomaly" information
- More effective analysis relies on higher semantic features
- Examples:
  - A speed boat moving quickly in open water
  - A fishing boat moving slowly into the docks
  - A yacht circling slowly around landmark during night hours
- Motif representation



a sequence of **motifs**



with **motif attributes**

Motif-Oriented Feature Space

- Each motif expression has attributes (*e.g.*, speed, location, size, time)
- Attributes express how a motif was expressed
  - A right-turn at 30mph near landmark Y at 5:30pm
  - A straight-line at 120mph (!!!) in location X at 2:01am
- Motif-Oriented Feature Space
  - Naïve feature space

1. Map each distinct motif-expression to a feature
2. Trajectories become feature vectors in the new space
   - Let there be *A* attributes attached to every motif, each trajectory is a set of motif-attribute tuples

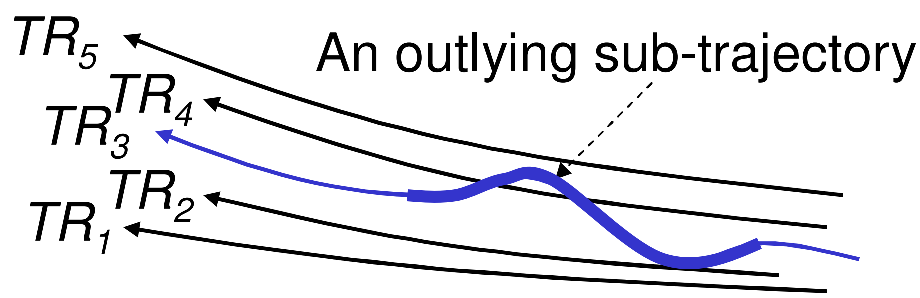
     $\{(m_i, v_1, v_2, \ldots, v_A), \ldots, (m_j, v_1, v_2, \ldots, v_A)\}$
   - Example:
     - Object 1: {(right-turn, 53mph, 3:43pm)} → (1, 0)
     - Object 2: {(right-turn, 50mph, 3:47pm)} → (0, 1)

Motif Feature Extraction

- Intuition: Should have features that describe general high-level concepts
  - "Early Morning" instead of 2:03am, 2:04am, …
  - "Near Location X" instead of "50m west of Location X"
- Solution: Hierarchical micro-clustering
  - For each motif attribute, cluster values to form higher level concepts
  - Hierarchy allows flexibility in describing objects
    - e.g., "afternoon" vs. "early afternoon" and "late afternoon"

Trajectory Outlier Detection: A Partition-and-Detect Framework (Lee et al. 08)

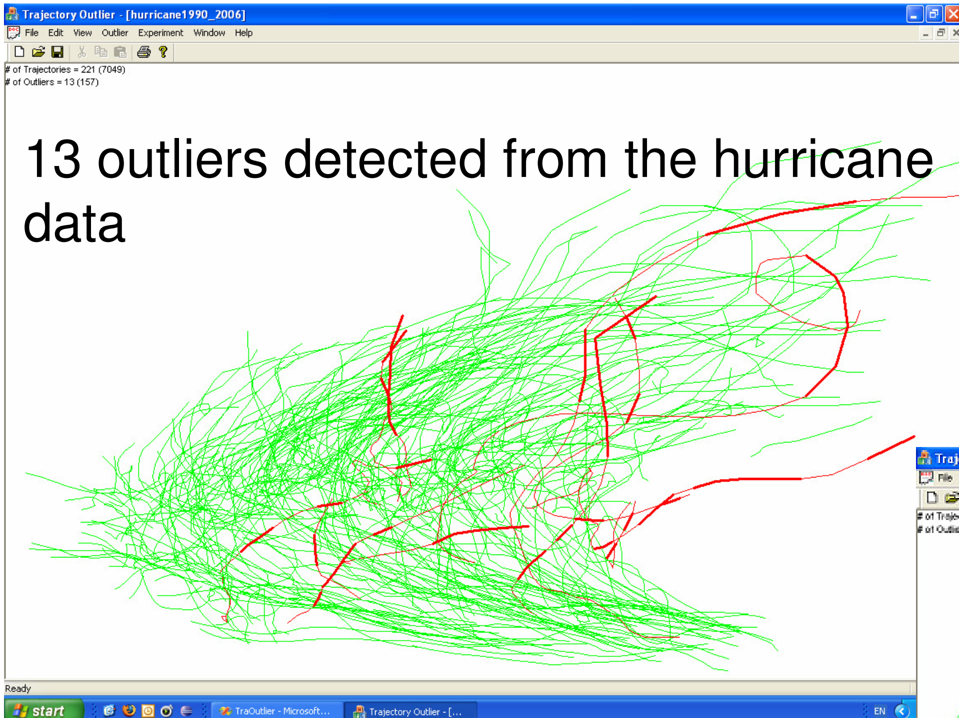
- Existing algorithms compare trajectories ***as a whole*** ➔ They might not be able to detect ***outlying portions*** of trajectories
  - e.g., TR_3 is not detected as an outlier since its overall behavior is similar to those of neighboring trajectories



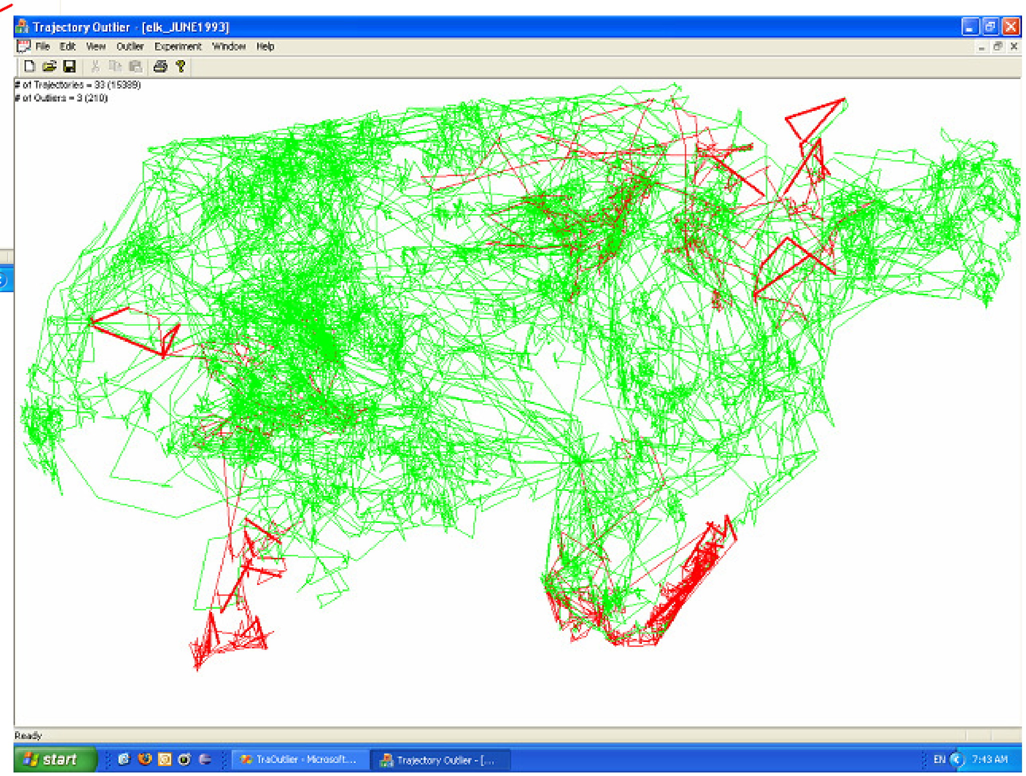
- The ***partition-and-detect framework*** is proposed to detect outlying ***sub***-trajectories

Experiments: Sample Detection Results

13 outliers detected from the hurricane data



Three outliers found from the Elk Data



Summary: Moving Object Mining

- Pattern Mining
  - Trajectory patterns, flock and leadership patterns, periodic patterns,
- Clustering
  - Probabilistic method, density-based method, partition-and-group framework
- Prediction
  - linear/non-linear model, vector-based method, pattern-based method
- Classification
  - Machine learning-based method, HMM-based method, *TraClass* using collaborative clustering
- Outlier Detection
  - Unsupervised method, supervised method, partition-and-detect framework

References: Moving Object Databases and Queries

- R. H. Gueting and M. Schneider. *Moving Objects Databases*. Morgan Kaufmann, 2005.
- S. R. Jeffrey, G. Alonso, M.J. Franklin, W. Hong, and J. Widom. A pipelined framework for online cleaning of sensor data streams. *ICDE'06.*
- N. Jing, Y.-W. Huang, and E. A. Rundensteiner. Hierarchical optimization of optimal path finding for transportation applications. *CIKM'96*.
- C. S. Jensen, D. Lin, and B. C. Ooi. Query and update efficient b+-tree based indexing of moving objects. *VLDB'04*.
- E. Kanoulas, Y. Du, T. Xia, and D. Zhang. Finding fastest paths on a road network with speed patterns. *ICDE'06*.
- J. Krumm and E. Horvitz. Predestination: Inferring destinations from partial trajectories. *Ubicomp'06.*
- E. M. Knorr, R. T. Ng, and V. Tucakov. Distance-based outliers: Algorithms and applications. *The VLDB Journal*, 8(3):237-253, February 2000.
- L. Liao, D. Fox, and H. Kautz. Learning and inferring transportation routines. *AAAI'04*.

References on Moving Object Pattern Mining (I)

- M. Andersson, Gudmundsson, J., Laube, P. & Wolle, T., "Reporting Leaders and Followers Among Trajectories of Moving Point Objects" , GeoInformatica, 2008.
- M. Benkert, J. Gudmundsson, F. Hubner, and T. Wolle. Reporting flock patterns. *Euro. Symp. Algorithms'06.*
- M. Benkert, J. Gudmundsson, F. Hubner, and T. Wolle. Reporting leadership patterns among trajectories. *SAC'07*.
- H. Cao, N. Mamoulis, and D. W. Cheung. Mining frequent spatiotemporal sequential patterns. *ICDM'05*.
- M. G. Elfeky, W. G. Aref, and A. K. Elmagarmid. Periodicity detection in time series databases. IEEE Trans. Knowl. Data Eng., 17(7), 2005.
- R. Fraile and S. J. Maybank. Vehicle trajectory approximation and classification. In *Proc. British Machine Vision Conf.*, Southampton, UK, September 1998.
- F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi. Trajectory pattern mining. KDD'07.
- S. Gaffney and P. Smyth. Trajectory clustering with mixtures of regression models. *KDD'99*.
- J. Gudmundsson and M. J. van Kreveld. Computing longest duration flocks in trajectory data. GIS'06.
- J. Gudmundsson, M. J. van Kreveld, and B. Speckmann. Efficient detection of patterns in 2d trajectories of moving points. *GeoInformatica*, 11(2):195-215, June 2007.
- H. Jeung, M. L. Yiu, X. Zhou, C. S. Jensen and H. T. Shen, "Discovery of Convoys in Trajectory Databases", VLDB 2008.

References on Moving Object Pattern Mining (II)

- P. Kalnis, N. Mamoulis, and S. Bakiras. On discovering moving clusters in spatiotemporal data. SSTD 2005.
- V. Kostov, J. Ozawa, M. Yoshioka, and T. Kudoh. Travel destination prediction using frequent crossing pattern from driving history. *Proc. Int. IEEE Conf. Intelligent Transportation Systems*, Vienna, Austria, Sept. 2005
- Y. Li, J. Han, and J. Yang. Clustering moving objects. *KDD'04*.
- Z. Li, et al., “MoveMine: Mining Moving Object Databases", SIGMOD’10 (system demo)
- Z. Li, B. Ding, J. Han, and R. Kays, “Swarm: Mining Relaxed Temporal Moving Object Clusters”, in submission
- Z. Li, B. Ding, J. Han, and R. Kays, “Mining Hidden Periodic Behaviors for Moving Objects”, in submission
- N. Mamoulis, H. Cao, G. Kollios, M. Hadjieleftheriou, Y. Tao, and D. W. Cheung. Mining, indexing, and querying historical spatiotemporal data. KDD 2004..
- M. Nanni and D. Pedreschi. Time-focused clustering of trajectories of moving objects. *JIIS*, 27(3), 2006.
- I. F. Sbalzariniy, J. Theriot, and P. Koumoutsakos. Machine learning for biological trajectory classification applications. In *Proc. 2002 Summer Program, Center for Turbulence Research*, August 2002.
- I. Tsoukatos and D. Gunopulos. Efficient mining of spatiotemporal patterns. *SSTD'01*.

References on Outlier Detection

- E. Horvitz, J. Apacible, R. Sarin, and L. Liao. Prediction, expectation, and surprise: Methods, designs, and study of a deployed traffic forecasting service. *UAI'05*
- E. M. Knorr, R. T. Ng, and V. Tucakov. Distance-based outliers: Algorithms and applications. The VLDB Journal, 8(3):237-253, February 2000.
- J.-G. Lee, J. Han, and X. Li, "Trajectory Outlier Detection: A Partition-and-Detect Framework", ICDE 2008
- J.-G. Lee, J. Han, and K.-Y. Whang, "Trajectory Clustering: A Partition-and-Group Framework", SIGMOD'07
- X. Li, J. Han, S. Kim, "Motion-alert: Automatic anomaly detection in massive moving objects", ISI 2006
- X. Li, J. Han, S. Kim, and H. Gonzalez, "ROAM: Rule- and Motif-Based Anomaly Detection in Massive Moving Object Data Sets", SDM'07
- J. Owens and A. Hunter. Application of the self-organizing map to trajectory classification. In . 3rd IEEE Int. Workshop on Visual Surveillance, Dublin, Ireland, July 2000

References on Prediction and Classification

- Faisal I. Bashir, Ashfaq A. Khokhar, Dan Schonfeld, View-invariant motion trajectory-based activity classification and recognition, Multimedia Syst. (MMS) 12(1):45-54 (2006)
- R. Fraile and S. J. Maybank. Vehicle trajectory approximation and classification. In Proc. British Machine Vision Conf., Southampton, UK, Sept. 1998
- H. Jeung, Q. Liu, H. T. Shen, X. Zhou: A Hybrid Prediction Model for Moving Objects. ICDE 2008
- J.-G. Lee, J. Han, X. Li, and H. Gonzalez, "TraClass: Trajectory Classification Using Hierarchical Region-Based and Trajectory-Based Clustering", VLDB 2008.
- Anna Monreale, Fabio Pinelli, Roberto Trasarti, Fosca Giannotti: WhereNext: a location predictor on trajectory pattern mining. KDD 2009
- I. F. Sbalzariniy, J. Theriot, and P. Koumoutsakos. Machine learning for biological trajectory classification applications. In Proc. 2002 Summer Program, Center for Turbulence Research, August 2002.
- Y. Tao, C. Faloutsos, D. Papadias, B. Liu: Prediction and Indexing of Moving Objects with Unknown Motion Patterns. SIGMOD 2004