Spatio-Temporal Data Warehouses:

Current Status and Research Issues

Esteban ZIMÁNYI
Department of Computer & Decision Engineering (CoDE)
Université Libre de Bruxelles
ezimanyi@ulb.ac.be

Invited Talk
19th International Symposium on Temporal Representation and Reasoning
Leicester, UK
12-14 September 2012

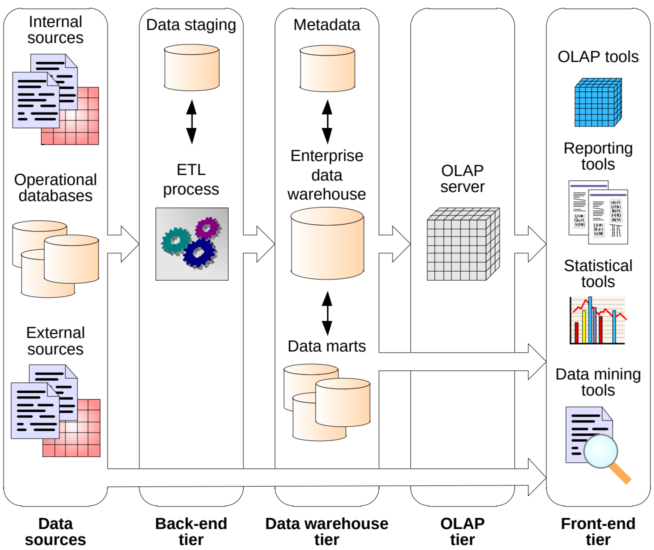
Contents

- **Background**
  - Data Warehouses and OLAP
  - Nested Relational Calculus
  - Temporal Aggregation
- Spatio-Temporal Data Warehouses
- From Spatio-Temporal Data to Trajectory Data
- Conclusions & Future Work

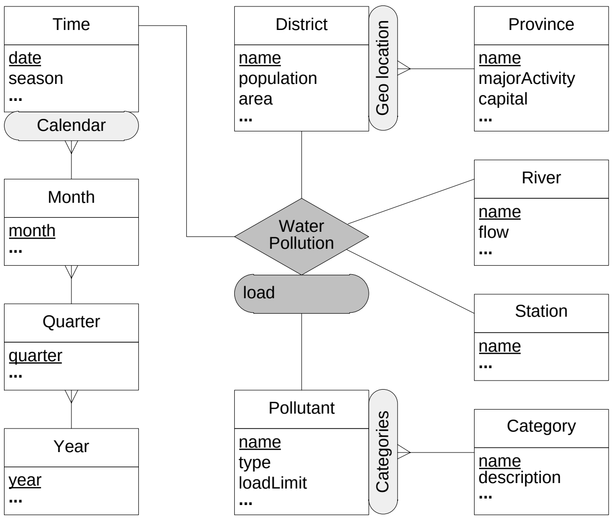
Data Warehouses

- **Operational databases** (**OLTP**) are not suitable for data analysis
  - Contain detailed data
  - Do not include historical data
  - Perform poorly for complex queries due to normalization
- **Data Warehouses** (DW) address requirements of decision-making users
  ⇒ **Business Intelligence**
- A data warehouse is a collection of **subject-oriented**, **integrated**, **nonvolatile**, and **time-varying** data to support data management decisions
- **Online analytical processing** (**OLAP**) allow decision-making users to perform **interactive** analysis of data
- **Data Warehouses** typically store **huge amounts of data**

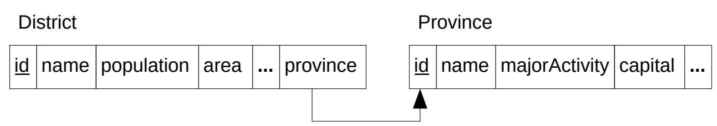
Typical Data Warehouse Architecture



Conventional Data Warehouses: Example



Nested Relational Calculus



- **Simple query**: Name and population of districts of the Antwerp province

  $\{d.\text{name}, d.\text{population} \mid \text{District}(d) \wedge \exists p\ (\text{Province}(p) \wedge$
  $d.\text{province} = p.\text{id} \wedge p.\text{name} = \text{'Antwerp'})\}$

- **Aggregation query with group filtering**: Total population by province provided that it is greater than 100,000

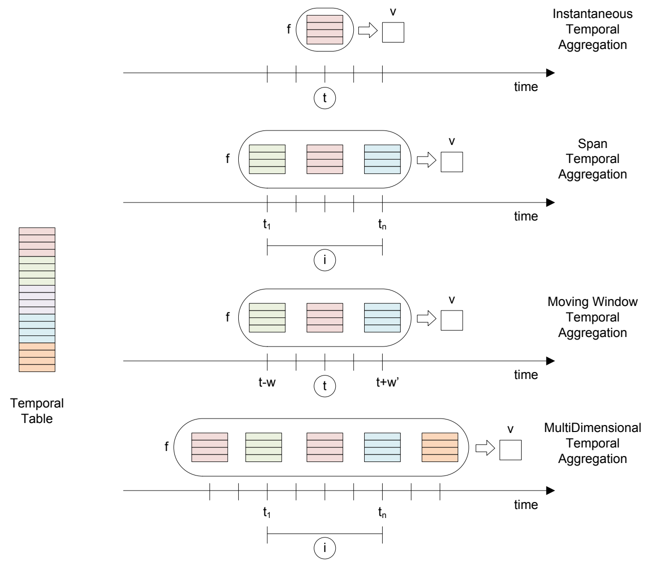
  $\{p.\text{name}, \text{totalPop} \mid \text{Province}(p) \wedge$
  $\text{totalPop} = \text{sum}_2(\{d.\text{id}, d.\text{population} \mid \text{District}(d) \wedge d.\text{province} = p.\text{id}\}) \wedge$
  $\text{totalPop} > 100,000\ \}$

- Corresponds to an SQL query with the GROUP BY and HAVING clauses

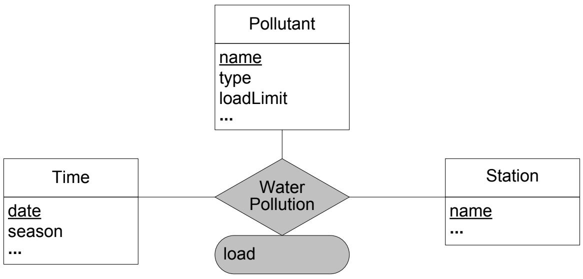
Temporal Aggregation [Gamper et al. 2009]

- **Instantaneous Temporal Aggregation**
  - To each time instant t is associated an aggregation group valid at t
  - Aggregation function applied to each group $\Rightarrow$ a single aggregate value at t
  - **Span Temporal Aggregation**: Similar, at a coarser granularity
- **Moving Window Temporal Aggregation**
  - To each time instant t is associated a window $w = [t-w, t+w']$ and an aggregation group valid at w
  - Aggregation function applied to each group $\Rightarrow$ a single aggregate value at t
  - **Multi-Dimensional Temporal Aggregation**: Similar, at a coarser granularity
- Equivalent to **temporal group composition** and **temporal partition composition** in [Vega López et al. 2005]

Temporal Aggregation



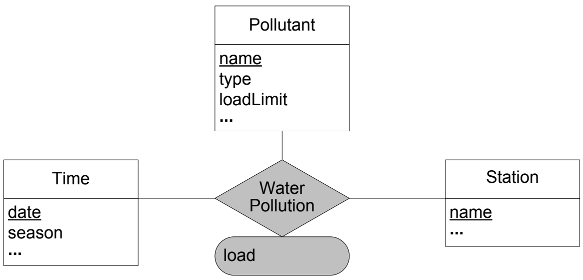
OLAP Queries: Instantaneous Temporal Aggregation



- By pollutant and day count the number of stations that exceeded the load limit

$$\{p.\text{name}, t.\text{date}, \text{countExc} \mid \text{Pollutant}(p) \wedge \text{Time}(t) \wedge \\ \text{countExc} = \text{count}_1(\{w.\text{station} \mid \text{WaterPollution}(w) \wedge w.\text{pollutant} = p.\text{id} \wedge \\ w.\text{time} = t.\text{id} \wedge w.\text{load} > p.\text{loadLimit}\})\}$$

OLAP Queries: Moving Window Temporal Aggregation



- By station, pollutant, and day, give the 3-day moving average of load

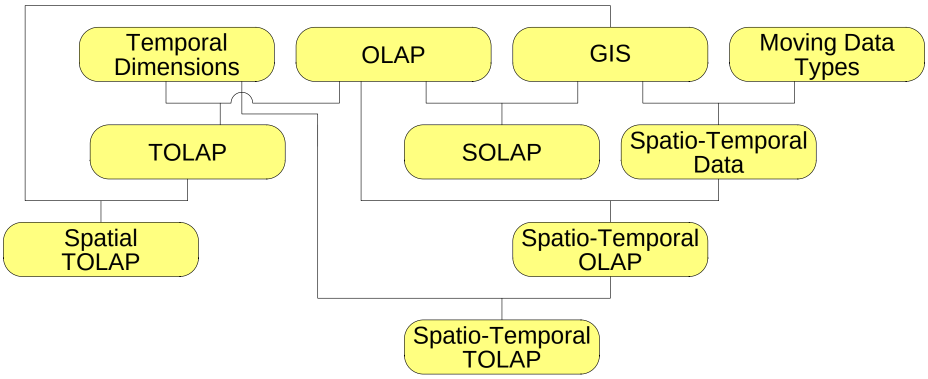
$$\{s.\text{name}, p.\text{name}, t.\text{date}, \text{3dMovAvg} \mid \text{Station}(s) \wedge \text{Pollutant}(p) \wedge \text{Time}(t) \wedge \\ \text{3dMovAvg} = \text{avg}_2(\{w.\text{id}, w.\text{load} \mid \text{WaterPollution}(w) \wedge w.\text{station} = s.\text{id} \wedge \\ w.\text{pollutant} = p.\text{id} \wedge \exists t_1\ (\text{Time}(t_1) \wedge w.\text{time} = t_1.\text{id} \wedge \\ 0 \leq t.\text{date} - t_1.\text{date} \wedge t.\text{date} - t_1.\text{date} \leq 3)\ \})\ \}$$

Contents

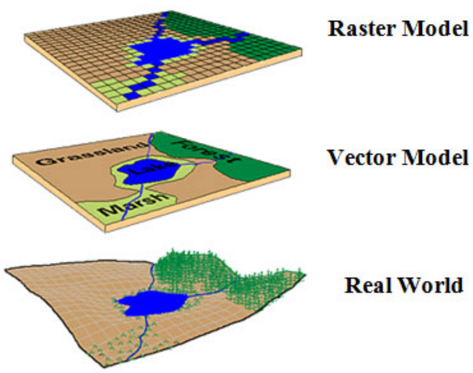
Spatio-Temporal Data Warehouses

- Several proposals aim at extending DW and OLAP with **spatial/temporal features**
- **No commonly agreed definition** of what is a spatio-temporal DW and what functionality it should support
- Proposed solutions **vary considerably** in the kind of **data that can be represented** and the kind of **queries that can be expressed**
- [Vaisman & Zimányi 2009] defined
  - **Conceptual framework** for spatio-temporal DWs using an extensible type system
  - **Taxonomy of several classes of queries** of increasing expressive power extending the tuple relational calculus with aggregated functions [Klug 1982]
- This provides the **underlying basis** for **implementing** spatio-temporal DWs

A Taxonomy for Spatio-Temporal Data Warehouses

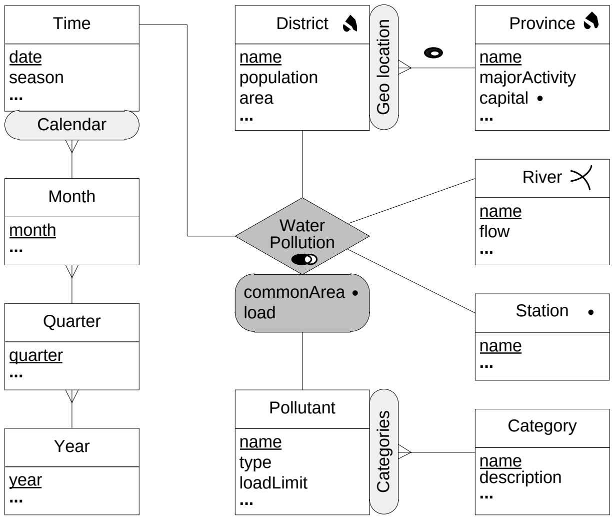


Representing Spatial Data



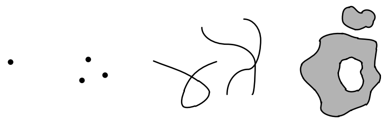
- Two complementary ways to represent spatial data
- **Object-based** (vector) model: Objects of interest are stored with their spatial extent
- **Space-based** (raster) model: Space is represented as a continuum, to each point in space is associated a value of a phenomenon of interest

Spatial Data Warehouses: Example



Spatial Data Types [Güting et al. 2000]

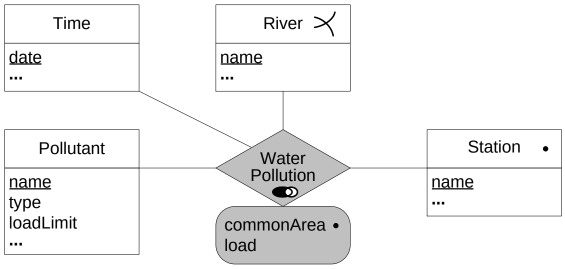
- **Spatial types**: point, points, line, region



- These types have an associated set of operations

Class	Operations
Predicates	isempty, =, ≠, intersects, inside, <, ≤, ≥, >, before, touches, attached, overlaps, on_border, in_interior
Set Operations	intersection, union, minus, crossings, touch_points, common_border
Aggregation	min, max, avg, center, single
Numeric	no_components, size, duration, length, area, perimeter
Distance and Direction	distance, direction

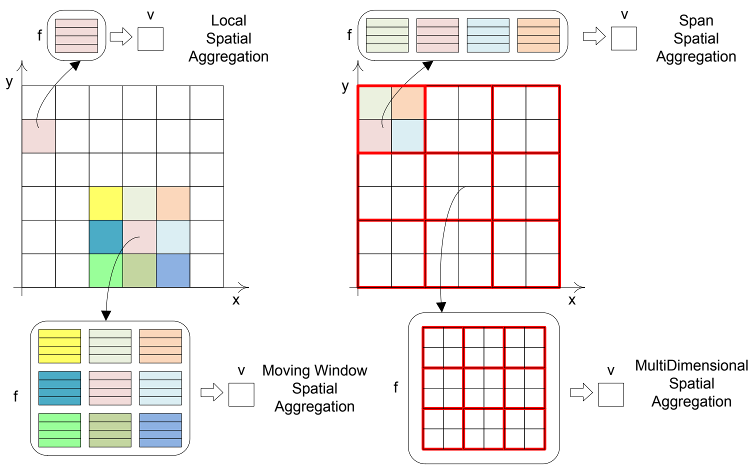
Spatial OLAP (SOLAP) Queries



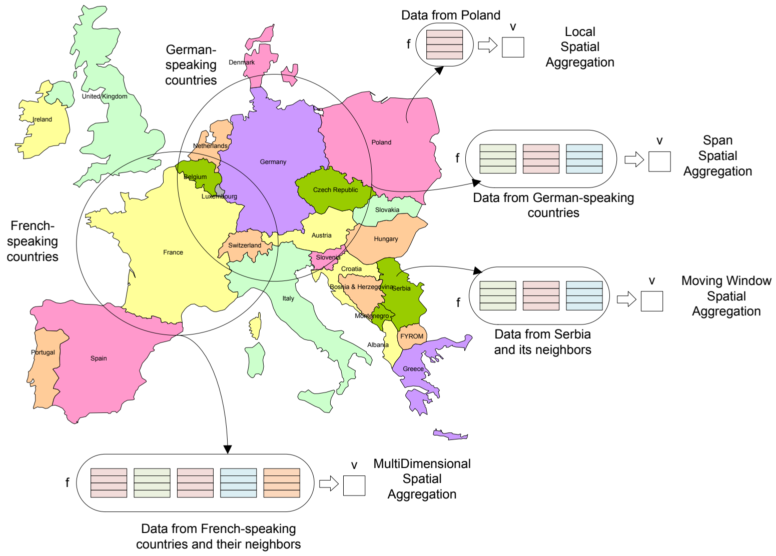
- For stations located on the Schelde river, give the average content of lead in the last quarter of 2008

$$\{s.\text{name}, \text{avgLead} \mid \text{Station}(s) \land \exists r, \exists p\ (\text{River}(r) \land r.\text{name} = \text{`Schelde'} \land \boxed{\text{intersects}(r.\text{geometry}, s.\text{geometry})} \land \text{Pollutant}(p) \land p.\text{name} = \text{`Lead'} \land \text{avgLead} = \text{avg}_2(\{w.\text{id}, w.\text{load} \mid \text{WaterPollution}(w) \land w.\text{station} = s.\text{id} \land w.\text{pollutant} = p.\text{id} \land \exists t\ (\text{Time}(t) \land w.\text{time} = t.\text{id} \land t.\text{date} \geq 1/10/2008 \land t.\text{date} \leq 31/12/2008)\})\,)\}$$

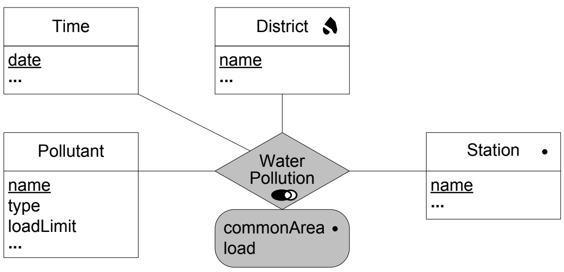
Spatial Aggregation: Space Based



Spatial Aggregation: Object Based



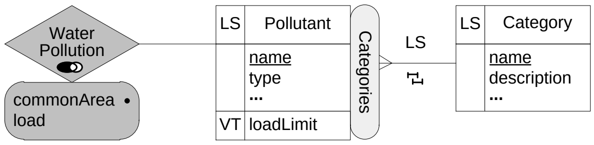
Spatial Aggregation Queries



- Union of the geometries of the districts in which the average content of lead in the last quarter of 2008 was greater than the load limit

$$\boxed{\text{union}(\{d.\text{geometry}} \mid \text{District}(d) \wedge \exists p\, (\text{Pollutant}(p) \wedge p.\text{name} = \text{`Lead'} \wedge$$
$$\text{avg}_2(\{w.\text{id}, w.\text{load} \mid \text{WaterPollution}(w) \wedge w.\text{district} = d.\text{id} \wedge$$
$$w.\text{pollutant} = p.\text{id} \wedge \exists t\, (\text{Time}(t) \wedge w.\text{time} = t.\text{id} \wedge$$
$$t.\text{date} \geq 1/10/2008 \wedge t.\text{date} \leq 31/12/2008)\, \})$$
$$> p.\text{loadLimit})\, \}$$

Temporal Data Warehouses



- Arise when **evolution of dimension instances** is supported
  - Also referred to as **slowly changing dimensions** [Kimball 96]
- **Temporality types**: Valid time (VT), Transaction Time (TT), Bitemporal Time (BT), Lifespan (LS), DW loading time (DWLT)
- Temporality represented using **moving types** moving(α) where α is a base type
  - Lifespan of Pollutant is of type moving(bool)
  - Temporal attribute loadLimit is of type moving(real)
  - Temporal relationship between Pollutant and Category is of type moving(id)

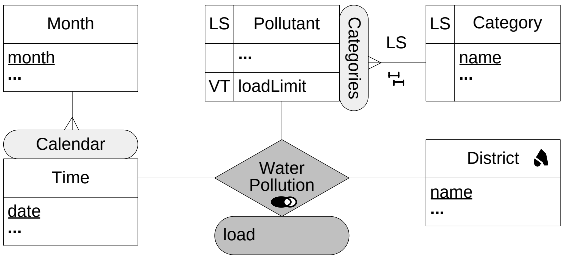
Moving Types [Güting et al. 2000]

- Capture the evolution over time of base types and spatial types
- Obtained by applying a constructor moving(α)
  - A value of type moving(point) is a continuous function f : instant $\rightarrow$ point
- Operations on moving types

Class	Operations
Projection to Domain/Range	deftime, rangevalues, locations, trajectory, routes, traversed, inst, val
Interaction with Domain/Range	atinstant, atperiods, initial, final, present, at, atmin, atmax, passes
Rate of change	derivative, speed, turn, velocity
Lifting	(all new operations inferred)

- **Lifting**: Operations of moving types generalize those of the nontemporal types
  - A distance function with signature moving(point)$\times$moving(point) $\rightarrow$ moving(real) calculates the distance between two moving points
- **Semantics**: result is **computed at each time instant** using the non-lifted operation

Temporal OLAP (TOLAP) Queries

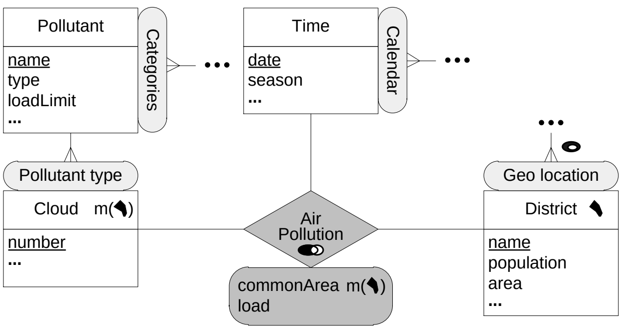


- By district, pollutant category, and month, give the average load

$$\{d.\mathsf{name}, c.\mathsf{name}, m.\mathsf{month}, \mathsf{avgLoad} \mid \mathsf{District}(d) \wedge \mathsf{Category}(c) \wedge \mathsf{Month}(m) \wedge \mathsf{avgLoad} = \mathsf{avg}_2(\{w.\mathsf{id}, w.\mathsf{load} \mid \mathsf{WaterPollution}(w) \wedge \exists t, \exists p\, (\mathsf{Time}(t) \wedge \mathsf{Pollutant}(p) \wedge w.\mathsf{district} = d.\mathsf{id} \wedge w.\mathsf{time} = t.\mathsf{id} \wedge t.\mathsf{month} = m.\mathsf{id} \wedge w.\mathsf{pollutant} = p.\mathsf{id} \wedge \boxed{\mathsf{val}(\mathsf{initial}(\mathsf{atperiods}(p.\mathsf{category}, t))) = c.\mathsf{id}}\,)\})\}$$

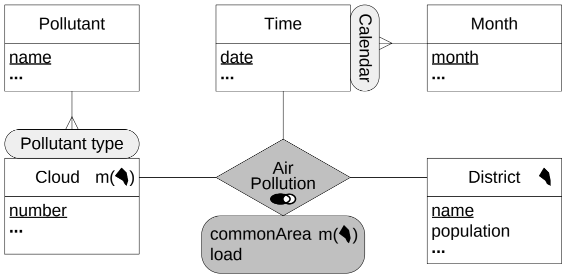
- Here we consider the category of a pollutant **valid at the day of the measure**
- Alternative: consider the **current category** of a pollutant

Spatio-Temporal OLAP (ST-OLAP)



- Arise when **spatial objects evolve over time**
- Evolution captured by **moving types** $\mathsf{moving}(\alpha)$ where α is a **spatial type**

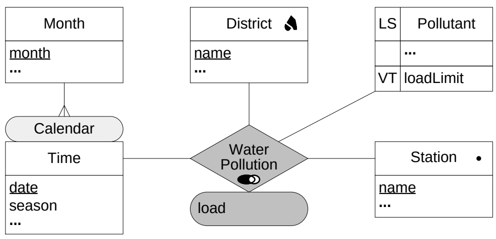
Spatio-Temporal OLAP (ST-OLAP) Queries



- By district and month, give the total number of persons affected by polluting clouds

$$\{d.\text{name}, m.\text{month}, \text{totalNo} \mid \text{District}(d) \wedge \text{Month}(m) \wedge \text{totalNo} = \text{area}(\text{union}(\{\boxed{\text{traversed(p.commonArea)}} \mid \text{AirPollution}(p) \wedge p.\text{district} = d.\text{id} \wedge \exists t\ (\text{Time}(t) \wedge t.\text{month} = m.\text{id})\}))/ \text{area}(d.\text{geometry}) \times d.\text{population}\}$$

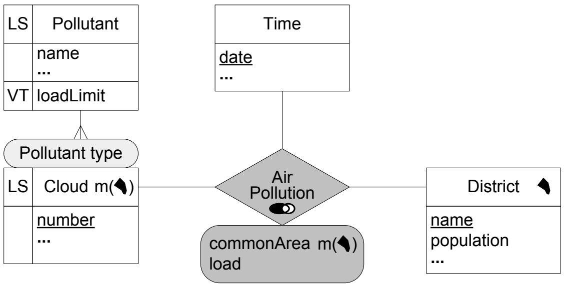
Spatial TOLAP (S-TOLAP)



- Arise when there are **spatial objects/attributes and temporal dimensions**
- By pollutant and month, give the average load in stations of the Namur district, if it is larger than the load limit during the reported month

$$\{p.\text{name}, m.\text{month}, \text{avgLoad} \mid \text{Pollutant}(p) \wedge \text{Month}(m) \wedge \exists d, \exists s\ (\text{District}(d) \wedge d.\text{name} = \text{'Namur'} \wedge \text{Station}(s) \wedge \boxed{\text{inside(s.geometry,d.geometry)}} \wedge \text{avgLoad} = \text{avg}_2(\{w.\text{id}, w.\text{load} \mid \text{WaterPollution}(w) \wedge \exists t\ (\text{Time}(t) \wedge w.\text{station} = s.\text{id} \wedge w.\text{time} = t.\text{id} \wedge t.\text{month} = m.\text{id} \wedge w.\text{pollutant} = p.\text{id})\}) \wedge \boxed{\text{avgLoad} > \text{val}(\text{initial}(\text{atperiods}(p.\text{loadLimit}, m.\text{month})))})\}$$

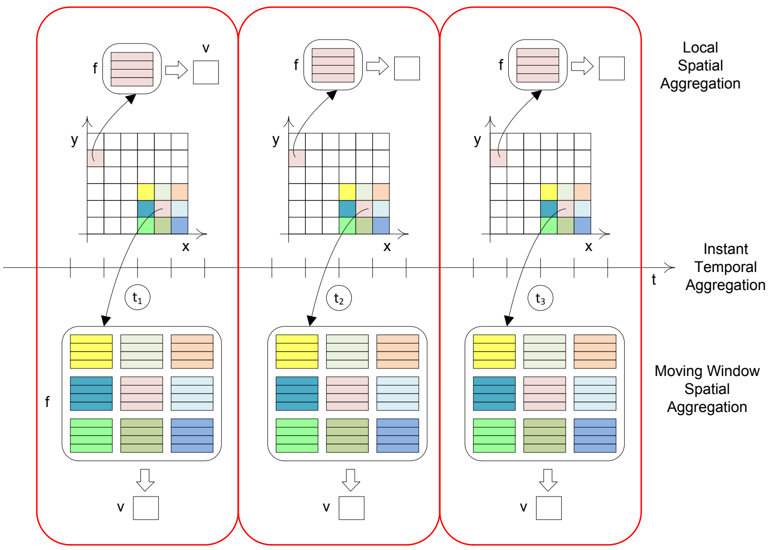
Spatio-Temporal TOLAP (ST-TOLAP)



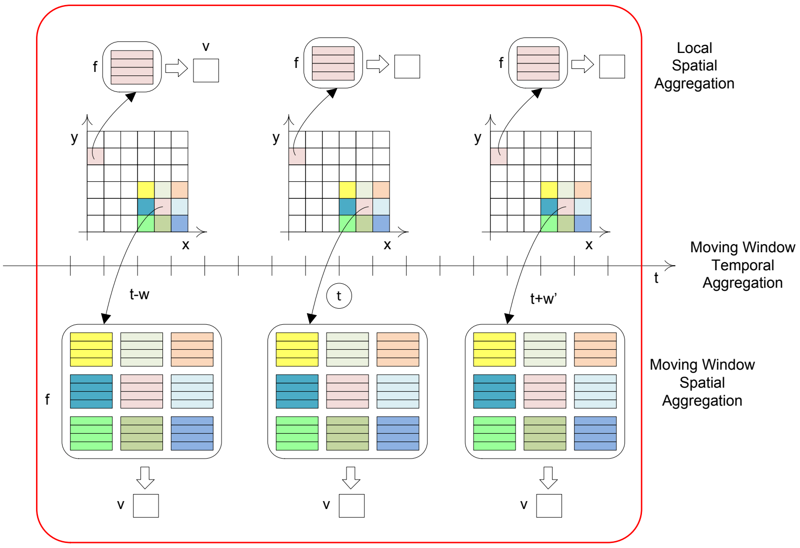
- **Most general case**: there are **moving geometries and temporal dimensions**
- Number of days when the Gent district was under at least one cloud of carbon monoxide (CO) with a load larger than the load limit valid when the cloud appeared

$$\text{duration}(\text{union}(\{t.\text{date} \mid \text{Time}(t) \wedge \exists a, \exists d, \exists c, \exists p\ (\text{AirPollution}(a) \wedge \text{District}(d) \wedge \text{Cloud}(c) \wedge \text{Pollutant}(p) \wedge a.\text{time} = t.\text{id} \wedge a.\text{district} = d.\text{id} \wedge d.\text{name} = \text{'Gent'} \wedge a.\text{cloud} = c.\text{id} \wedge c.\text{pollutant} = p.\text{id} \wedge p.\text{name} = \text{'CO'} \wedge \boxed{a.\text{load} > \text{val}(\text{atinstant}(p.\text{loadLimit}, \text{inst}(\text{initial}(\text{at}(c.\text{lifespan}, \text{true})))))}\ )\}))$$

Spatio-Temporal Aggregation: Space Based (1)



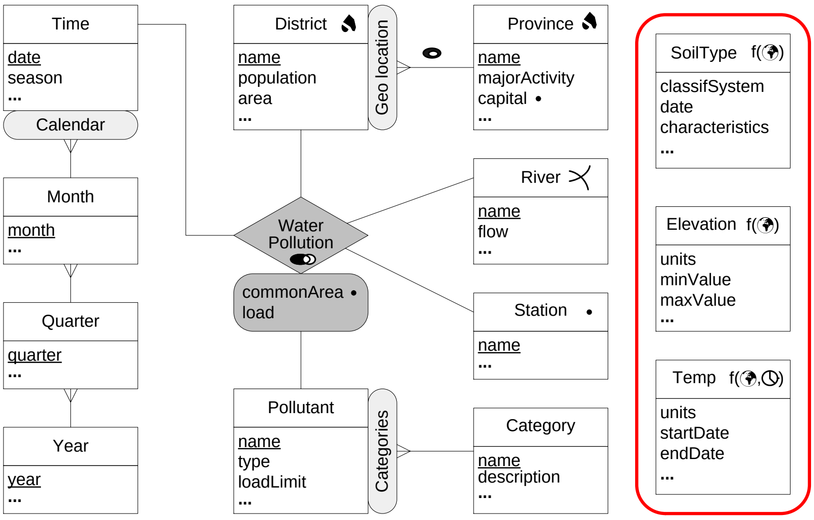
Spatio-Temporal Aggregation: Space Based (2)



Continuous Fields

- **Non-temporal** fields describe phenomena that change continuously in space
  - land elevation, soil type, . . .
- Represented by **field types** field(α) where α is a base type
- **Temporal** fields describe phenomena that change continuously in space and time
  - temperature, precipitation, . . .
- Represented by **field types** moving(field(α)) where α is a base type
  - moving(field(real)) defines a continuous function f : instant $\rightarrow$ (point $\rightarrow$ real)
- Operators for moving fields as before
- Field levels have a geometry attribute of type field(α) or moving(field(α))
- Field dimensions are **not connected to a fact relationship**

Spatial Data Warehouses with Continuous Fields: Example



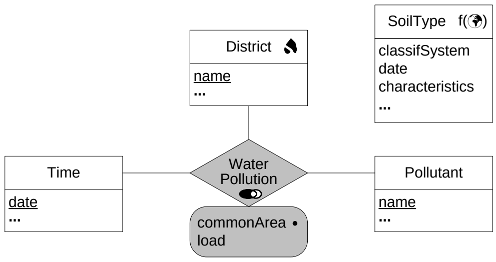
Field Types [Vaisman & Zimányi 2009]

- Capture the variation in space of base types
- Obtained applying a constructor field(α)
  - A value of type field(real) is a continuous function f : point $\rightarrow$ real
- Operations on field types

Class	Operations
Projection to Domain/Range	defspace, rangevalues, point, val
Interaction with Domain/Range	atpoint, atpoints, atline, atregion, at, atmin, atmax, defined, takes,concave, convex, flex
Lifting	(all new operations inferred)
Rate of change	partialder_x, partialder_y
Aggregation operators	integral, area, surface, favg, fvariance, fstdev

- **Lifting** applies to fields
  - The + operator with signature $\alpha \times \alpha \rightarrow \alpha$ generalized by allowing any argument to be a field as in field(α) $\times$ field(α) $\rightarrow$ field(α)
- **Semantics**: result is **computed at each point in space** using the non-lifted operation

Spatio-Temporal OLAP with Continuous Fields



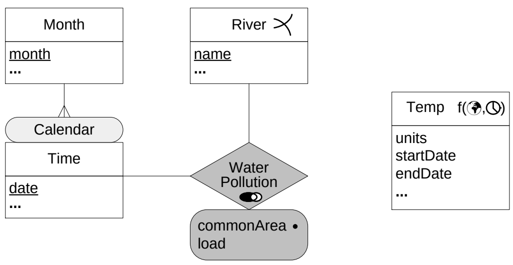
- For districts having at least 30% of their surface of clay soil, give the average load of lead on February 1st, 2009

$$\{d.\text{name}, \text{avgLead} \mid \text{District}(d) \wedge \exists p, \exists s\ (\text{Pollutant}(p) \wedge p.\text{name} = \text{`Lead'} \wedge$$

$$\boxed{\begin{array}{l}\text{SoilType}(s) \wedge \text{area}(\text{defspace}(\text{atregion}(\text{at}(s.\text{geometry}, \text{`Clay'}), d.\text{geometry})))/ \\ \text{area}(d.\text{geometry}) \geq 0.3\end{array}}$$

$$\text{avgLead} = \text{avg}_2(\{w.\text{id}, w.\text{load} \mid \text{WaterPollution}(w) \wedge \exists t\ (\text{Time}(t) \wedge$$
$$w.\text{district} = d.\text{id} \wedge w.\text{time} = t.\text{id} \ \wedge t.\text{date} = 1/2/2009 \wedge w.\text{pollutant} = p.\text{id})\ \})\ )\}$$

Spatio-Temporal OLAP with Temporal Continuous Fields



- For each river and month, give a field computing the average temperature of the month at each point in the river

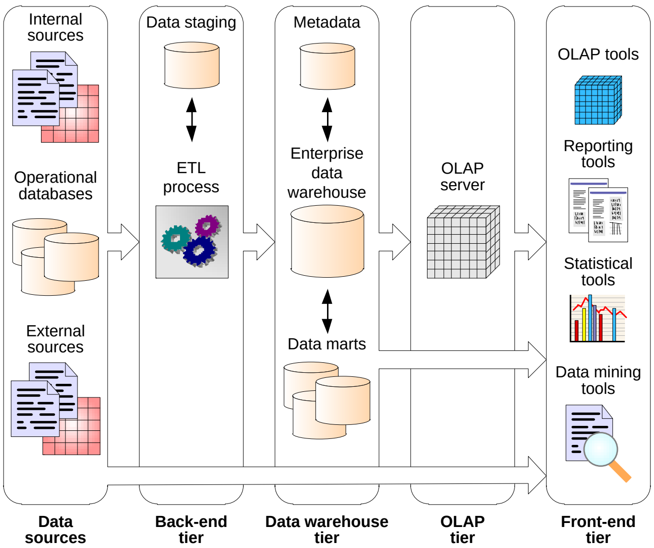
$$\{r.\text{name}, m.\text{month}, \text{temp} \mid \text{River}(r) \wedge \text{Month}(m) \wedge$$
$$\text{first} = \min(\{t.\text{date} \mid \text{Time}(t) \wedge t.\text{month} = m.\text{id}\} \wedge$$
$$\text{last} = \max(\{t.\text{date} \mid \text{Time}(t) \wedge t.\text{month} = m.\text{id}\} \wedge$$
$$\text{temp} = \text{avg}(\{\ \boxed{\text{atperiods}(\text{atregion}(f.\text{geometry}, r.\text{geometry}), \text{range}(\text{first}, \text{last}))} \mid$$
$$\text{Temp}(f)\ \})\ \}$$

Spatio-Temporal Data Warehouses: Conclusions

- Spatio-temporal DWs result from **combining GIS, OLAP, and temporal data types**
  - Temporal data types model geometries that evolve over time (moving objects) and evolving (slowly changing) dimensions
  - Field data types model continuous fields that change in space
  - Temporal fields obtained by composing field and temporal data types
- Our **taxonomy for spatio-temporal OLAP queries**
  - characterizes features required by spatio-temporal DWs
  - allows to classify different work in the literature
- Our framework is at a **conceptual level**: implementation issues were omitted
  - From abstract to concrete models: e.g., grid and TIN models for continuous fields
  - Optimization issues: index structures, pre-aggregation, query optimization, . . .

Contents

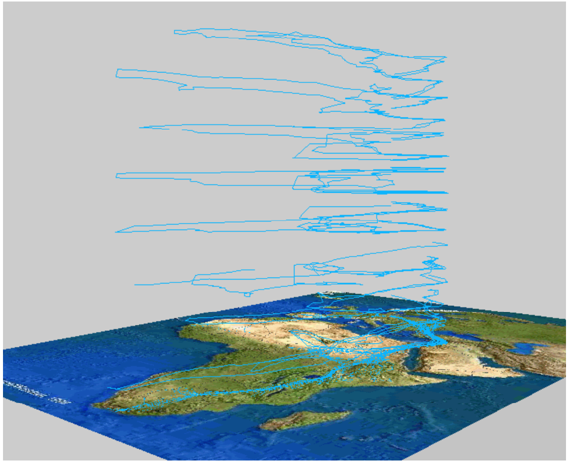
Typical Data Warehouse Architecture (Reminder)



Movement Data

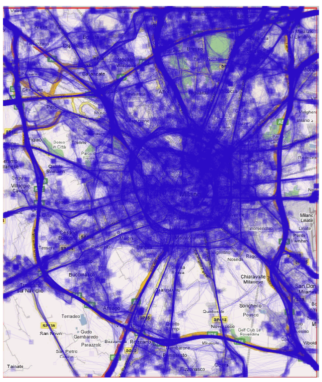
- Typically, a **temporal sequence** of position records (id, x, y, t)
- Collected nowadays in **rapidly growing amounts** due to the development of tracking technologies
  - GPS, RFID, WiFi, mobile phones, banking transactions, sensor networks, ...
- Complexities
  - **Huge amounts**: number of moving entities, number of records
  - **Geographic space** with its structure and complexity
  - **Time domain** at multiple granularities, linear and also multiple nested and overlapping cycles
  - **Data properties**: imprecision (errors in location, time, attributes), irregular and/or sparse sampling, missing values, ...
  - Large **diversity of types of movement**: constrained vs. unconstrained, vehicles/persons/animals, 2D vs 3D ...
  - **Real-world**: ill-defined, application-dependent problems

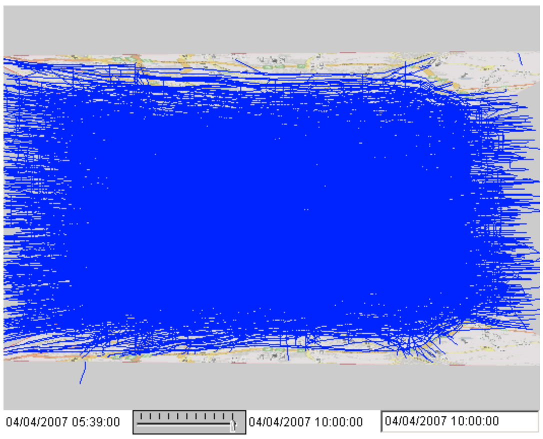
Examples of Movement Data: Migration of White Storks



Tracks of 35 storks during 8 years, about 2,000 positions ⇒ **Animals**

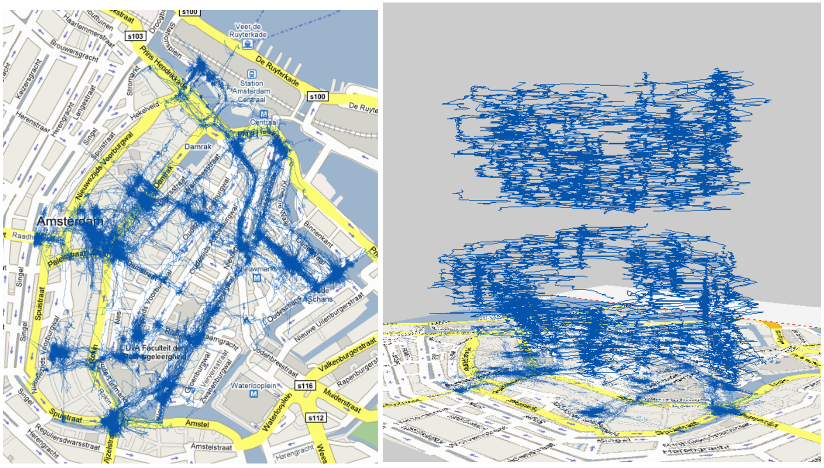
Examples of Movement Data: Cars in Milan





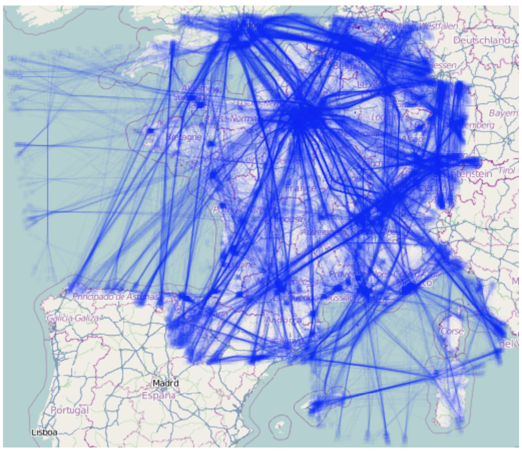
2,075,216 position records of 17,241 cars during 1 week ⇒ **Vehicles, network-constrained**

Examples of Movement Data: Children in Amsterdam



GPS tracks of 303 school children playing an educational game in Amsterdam, about 57,000 points ⇒ **Pedestrians**

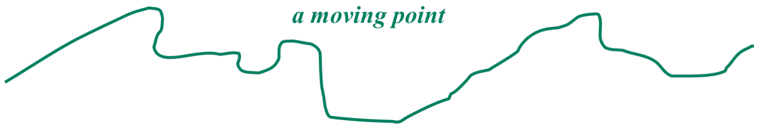
Examples of Movement Data: Air Traffic



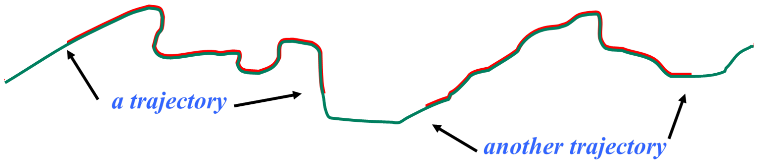
427,652 position records of 17,851 planes during 1 day ⇒ **3D movement**

From Movement to Trajectories [Parent et al. 2013]

- Movement is **continuous and never-ending**

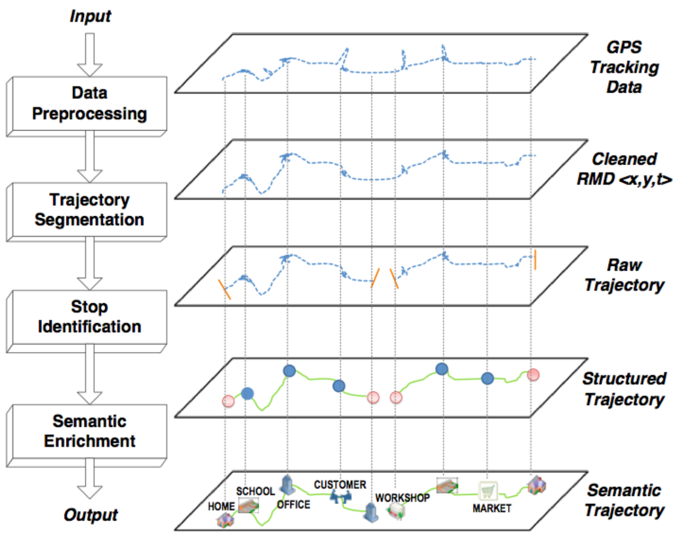


- A trajectory is a finite **meaningful** movement segment



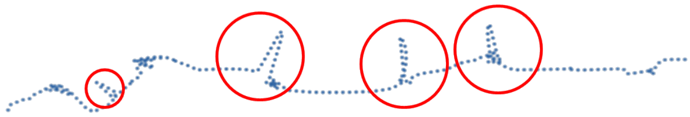
- Trajectory data is **semantically richer** than spatio-temporal data ⚠

Trajectory Construction



Cleaning Raw Data

- Input: Raw data



- Output: Cleaned raw data

- Methods: filtering, smoothing, outlier removal, missing point interpolation, map-matching, data compression, etc.

Segmentation into Trajectories

- Input: Cleaned raw data

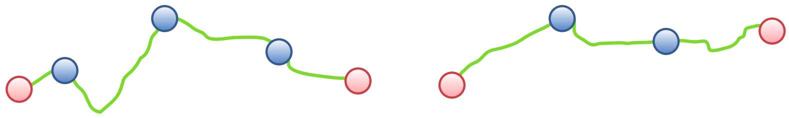
- Output: Trajectories

- Methods: various segmentation algorithms, based on spatial gaps, temporal gaps, time intervals, time series, etc.

Trajectory Structuring: Stop & Move Episodes

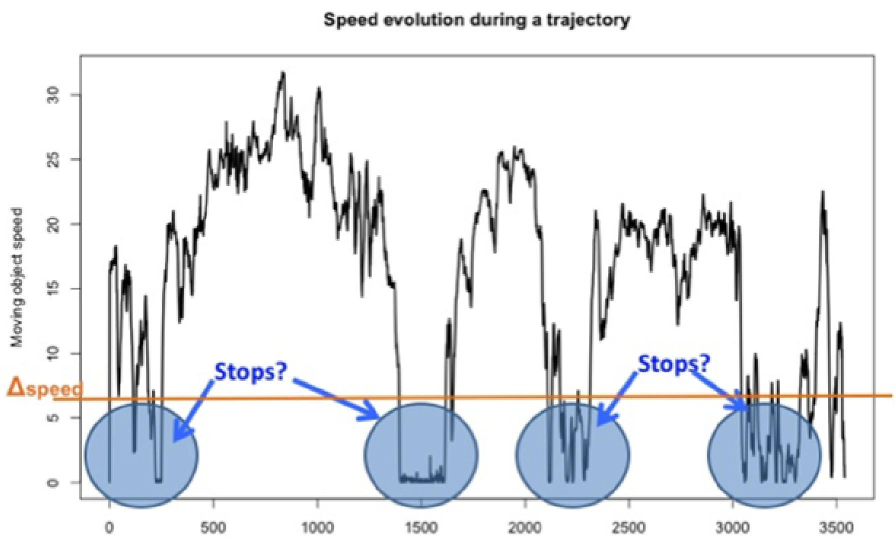
- Input: Trajectories

- Output: Structured trajectories (Stop, Move)



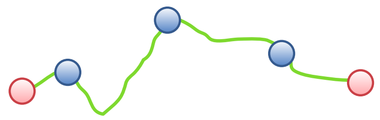
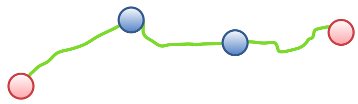
- Methods: various stop identification algorithms, based on velocity, density, etc.

Velocity-Based Stop Identification



Semantic Enrichment

- Input: Structured trajectories

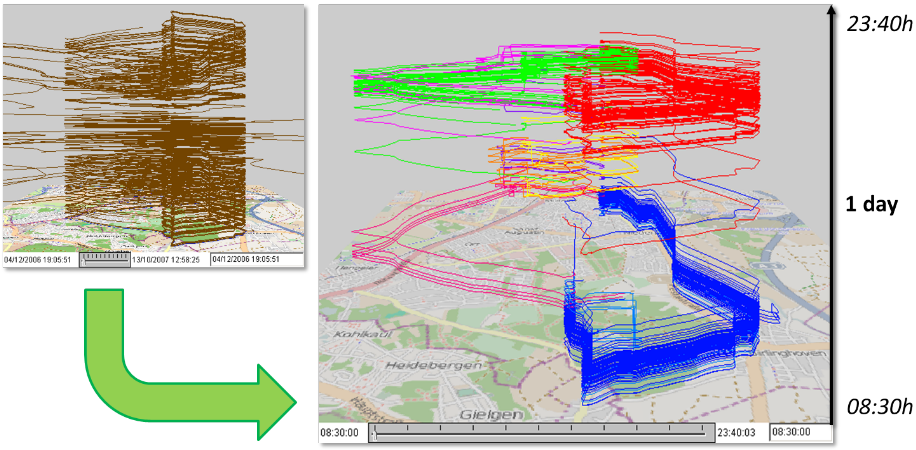
- Output: Semantic trajectories

- Methods: relate structured components (begin, end, stops, moves, . . .) to application knowledge (i.e., meaningful objects)

Visual Analytics [Andrienko & Andrienko 2012]

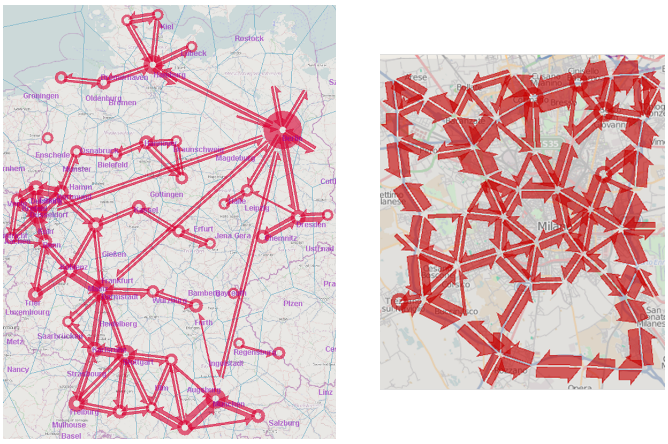
- Aims at helping people in
  - distilling relevant nuggets of information from large amounts of data
  - understanding the connections among relevant information
  - gaining insight from data
- Focuses on the division of labour between humans and machines
- Goal: computational power amplifies human perceptual and cognitive capabilities
- Visual representations: most effective means to convey information to human's mind, prompt human cognition and reasoning
- Combines interactive visualizations with automated analysis techniques such as
  - database processing
  - data mining algorithms
  - statistics
  - geographical analysis methods

Interactive Transformation of Time



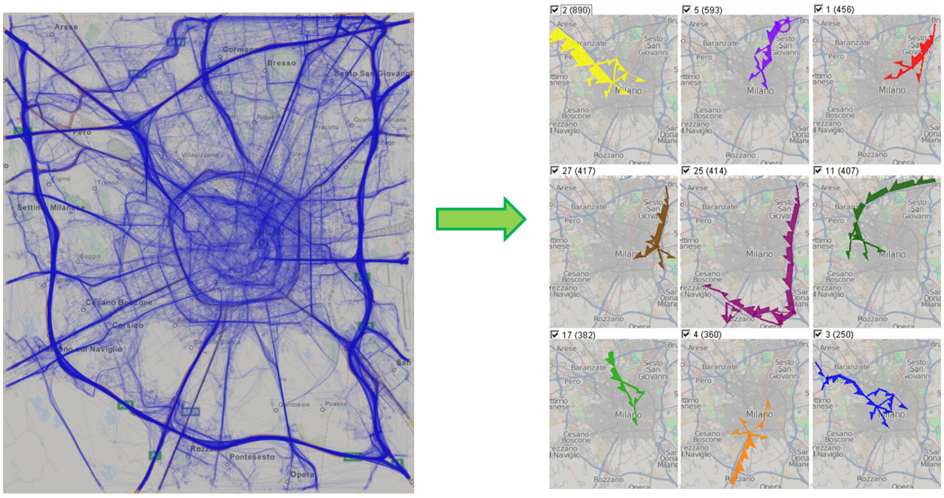
- Space-time cube on the right uses time transformation in respect to daily cycle
- This technique enables interpretation of repeated trajectories

Trajectory Summarization



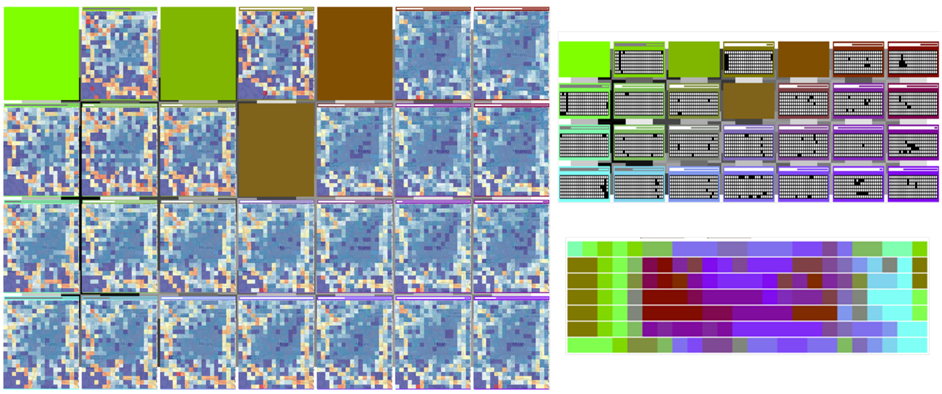
- Left: major flows of tourists in Germany, according to panoramio.com photos
- Right: major traffic flows in Milano, based on trajectories of about 20,000 cars

Scalable Trajectory Clustering



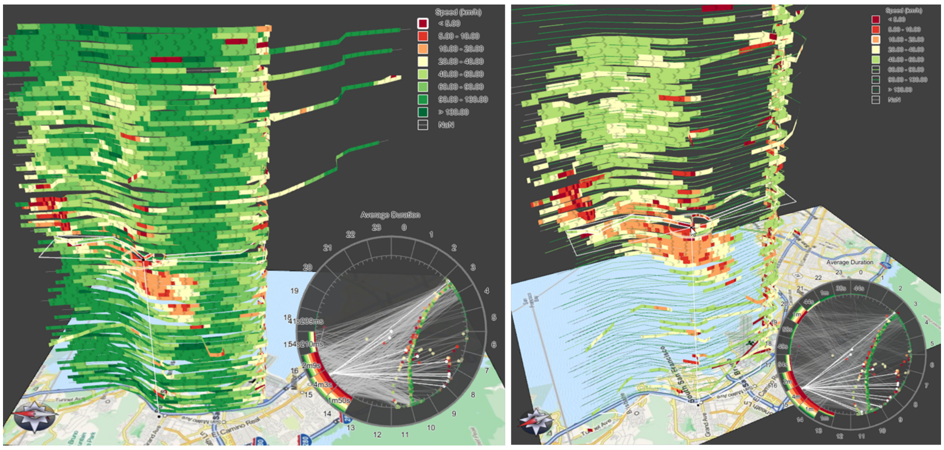
- Major clusters of trajectories extracted from the same data set (a week of 20K cars in Milano) presented by the trajectory summarization method

Similarity of Situations



- Hourly traffic situations in Milano (spatial distributions of counts of cars) are clustered; colors are assigned to clusters according to the similarity of situations
- The colors are projected to 2D time plot (bottom right) showing similarities of situations over 7 days x 24 hours

Visualization of Trajectory Attributes



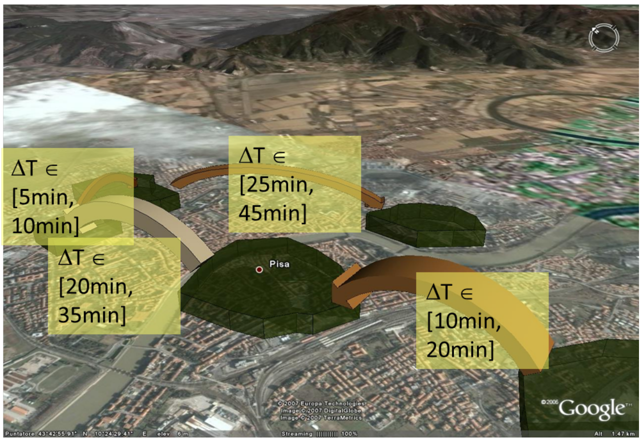
- For a cluster of trajectories, attribute values (speeds) are displayed on the trajectory wall display
- This enables investigation of traffic jams

Trajectory Data Mining [Giannotti et al. 2011]

- The process of analyzing large amounts of trajectory data to **identify unsuspected or unknown patterns** that might be of value to an application
- A particular step in the knowledge discovery process
- Some key research questions
  - Which spatio-temporal patterns are **useful abstractions** of mobility data?
  - How to **classify trajectories** according to specific behavior?
  - How to **interpret** in a meaningful way the discovered patterns?
  - How to make such analysis **privacy preserving** in a measurable way?

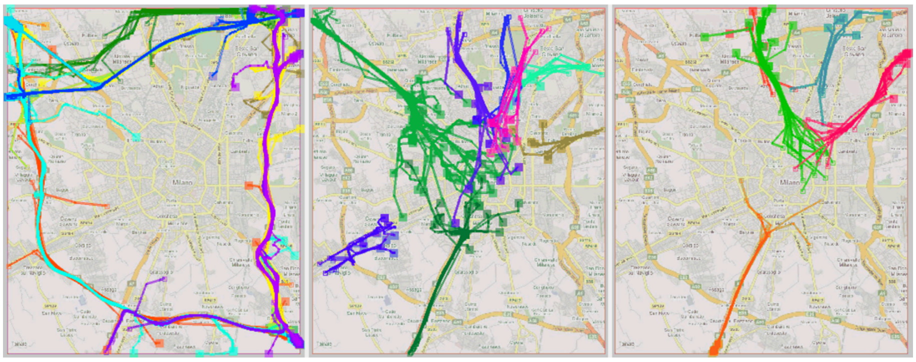
What are Trajectory Patterns (1) ?

- Frequent sets/sequences of places visited



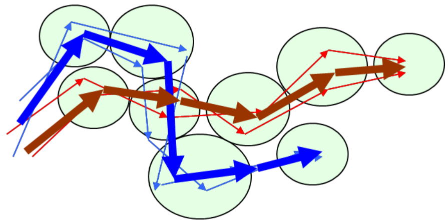
What are Trajectory Patterns (2) ?

- Groups of objects moving together



Trajectory Clustering

- **Cluster analysis**: Find groups where objects in a group are **similar** (or near) to one another and **different** from (or distant from) the objects in other groups

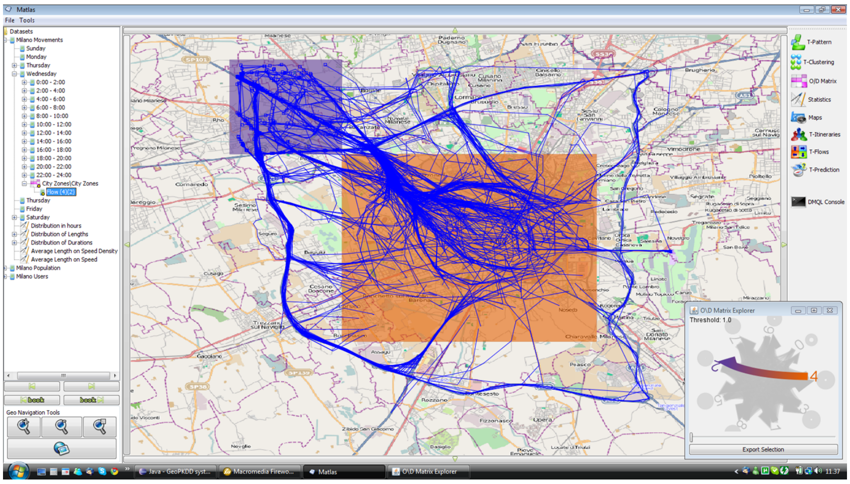


- Some research questions:
  - Which distance between trajectories?
  - Which kind of clustering?
  - What does a cluster mean?
    - A representative trajectory?

The M-Atlas tool

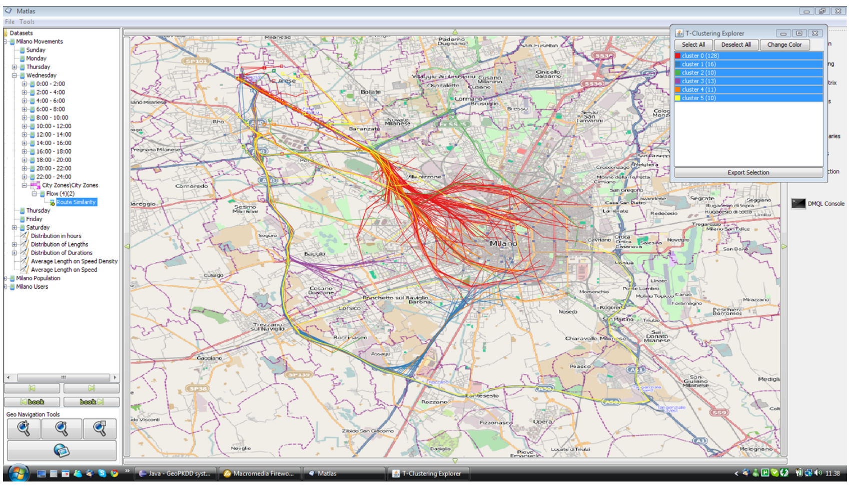
- A knowledge discovery support environment for trajectory data
- Data, patterns, and background knowledge need to be progressively combined
- T-patterns, T-clusters, etc. are the basic primitives within a Data Mining Query Language (DMQL), supporting the entire knowledge discovery process
- T-patterns, T-clusters, etc., once mined from a trajectory dataset, can be stored and later used for query, mining, and interpreting in a progressive way

Looking for Movement Patterns: From Where to Where



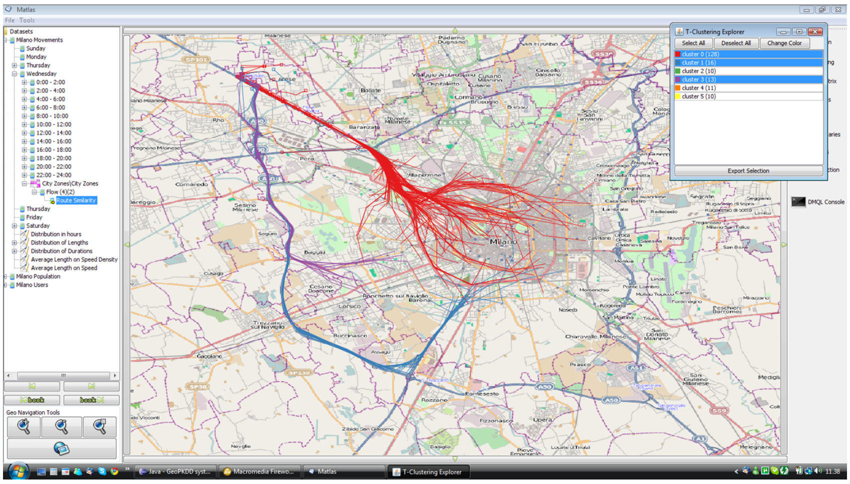
Select trajectories exiting the city from the centre towards North-West

Discovering Typical Routes



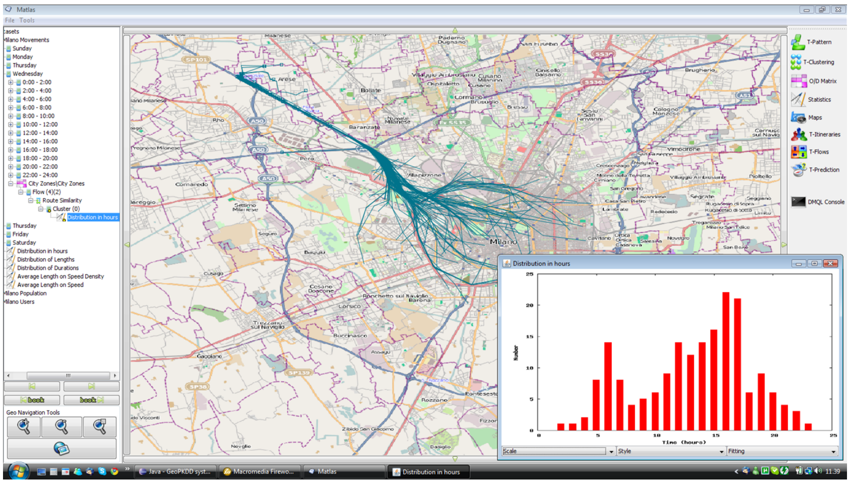
T-clustering divides trips based on route similarity

Focus on the Three Larger Clusters



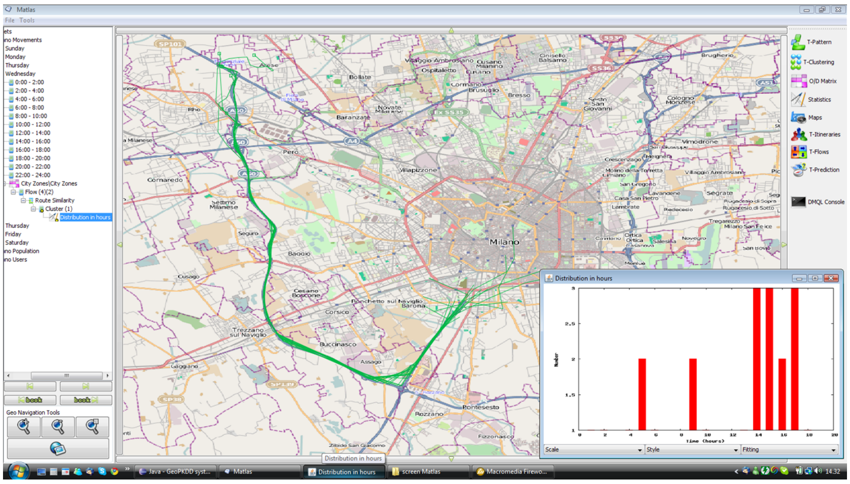
One group (red) goes straight to NW, other follow alternative routes

Temporal Analysis on Each Group



A small group of commuters in the morning and a larger one in the afternoon

Temporal Analysis on Each Group



A group of commuters in the afternoon follows an alternative route: are they smart?

Conclusions

- A large number of applications in a variety of domains are interested in analyzing movement of some type of objects or phenomena
- Nowadays, a huge amount of movement data that is being **continuously captured**
- Current approaches to process these massive data sets are **innapropriate**
- A complex and pipelined process is needed for **transforming raw samples to insightful knowledge**
- This process is by definition **iterative, semi-automatic, application-dependent, and multi-disciplinary**
- This raises **many theoretical and implementation issues**
  - A forthcoming book [Renso et al. 2013] surveys some of them
- Solutions proposed so far are just the **first steps** in the direction of mobility data understanding
- Exciting research domain, **huge potential implications** in our lives and in our planet

References (1)

- N. Andrienko, G. Andrienko. Visual Analytics of Movement: an Overview of Methods, Tools, and Procedures. *Information Visualization*, 2012. Forthcoming.
- A. Klug. Equivalence of Relational Algebra and Relational Calculus Query Languages Having. Aggregate Functions. *J. ACM* 29(3): 699-717, 1982.
- J. Gamper, M.H. Böhlen, C.S. Jensen. Temporal Aggregation. *Encyclopedia of Database Systems*: 2924-2929, Springer, 2009.
- F. Giannotti et al. Unveiling the complexity of human mobility by querying and mining massive trajectory data. *The VLDB Journal*, 20:695-719, 2011.
- R.H. Güting et al. A foundation for representing and querying moving objects. *ACM Trans. Database Syst.* 25(1): 1-42, 2000.
- R.H. Güting, M. Schneider. *Moving Objects Databases*. Morgan Kaufmann, 2005.

References (2)

- E. Malinowski, E. Zimányi. *Advanced Data Warehouse Design: From Conventional to Spatial and Temporal Applications*. Springer, 2008.
- C. Parent et al. Semantic Trajectories: Modeling and Analysis. *ACM Computing Surveys*, 45(4), 2013. Forthcoming.
- C. Renso, S. Spaccapietra, E. Zimányi (Eds.). *Mobility Data: Modeling, Management, and Understanding*. Cambridge Press, 2013. Forthcoming.
- A. Vaisman, E. Zimányi. What Is Spatio-Temporal Data Warehousing? *DaWaK*: 9-23, 2009.
- A. Vaisman, E. Zimányi. A Multidimensional Model Representing Continuous Fields in Spatial Data Warehouses. *ACM GIS*: 168-177, 2009.
- I.F. Vega López, R.T. Snodgrass, B. Moon. Spatiotemporal Aggregate Computation: A Survey. *IEEE Trans. Knowl. Data Eng.*, 17(2): 271-286, 2005.

Spatio-Temporal Data Warehouses:

Current Status and Research Issues

Esteban ZIMÁNYI

ULB

Questions ?