Uncertainty:
FINAL REPORT

Referring to the conceptual model defined in the paper
"A conceptual view on trajectories" [S. Spaccapietra et al.]

FABRIZIO CELLI
LAUSANNE, 26 AOÛT 2009

Abstract

- **Moving objects** are objects like cars, persons or animals equipped with a GPS devise that have a geometry changing over the time: they produce **trajectories**, that is to say descriptions of the movements of those objects
- We can define a **trajectory** as a time-space function that records the changing of the position of an object moving in space during a given time interval
- In our model, a trajectory is a sequence of *stops* and *moves*. A *stop* is a part of a trajectory in which the moving object does not move, while a *move* is a part of a trajectory in which the moving object changes its position

Context

- We collect data about the movements of persons driving cars in the city of Milan: cars are monitored by GPS devises
- During her movement by a car, a person can stop driving in order to reach a place walking: the problem is that we don't know exactly where she is going to, because only the car is monitored by a GPS devise
- What we want to do is to try to understand which place of interest (POI) the person is going to, in order to classify the person on the basis of her behavior (for example, tourist or worker, student or housewife…)
- To say the truth, we will not classify the physical person but her trajectory

What is Uncertainty

- In literature "uncertainty" has been defined as the measure of the difference between the actual content of a database and the content that the current user would have created by direct observation of reality
- We can find uncertainty in the acquisition process of the raw components of a model (the extraction of the sampling points): this is called **measurement uncertainty**
- Moreover, to obtain the curve representing the trajectory we apply an interpolation method to sample points, so the resulting curve will only be an approximation of the real trajectory: this is called **interpolation uncertainty**

Where is Uncertainty

- Due to continuous motion and **network delays**, the database location of a moving object will not always precisely represent its real location
- Spatio-temporal trajectories require a **geographical abstraction** and it is highly unlikely that geographical complexity can be reduced to models with perfect accuracy
- All information is stored in a database: in a digital environment, approximation must be expressed in a limited number of digits, as computer storage space is limited. This is the **raw problem**

Where is Uncertainty

- Then uncertainty can be connected to the **semantics** of the information stored in the database, i.e. to the interpretation given by the user. We must remember that raw data haven't any semantics associated
- Another kind of uncertainty can be found **inside the application** itself: for example, in our application there is uncertainty about the POI visited by a person during the stop (*we have developed an algorithm that aims to solve this problem, as we are going to show*)

Definitions

- A "**point of interest**" is a physical location of which we know the geographical coordinates: for our application, it is a place that can be visited by a person (e.g. a bar, a museum…)
- A "**classification**" for a set of points of interest is a multilevel hierarchy that assigns categories to each point of interest: it is multilevel because in this way we can choose the granularity of the classification (e.g. first level will have less classes)
- A "**category**" is a label associated to a point of interest: the definition of categories is an application-dependent issue and it is related to the semantics that the application gives to a trajectory

Pre-processing

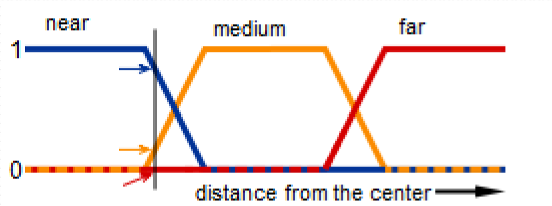
- For our application, a trajectory is a sequence of *stops* and each *stop* must be associated to a set of *points of interest* reachable from that *stop*
- We assign to each *stop* a set of *points of interest* on the basis of:
  - The **duration T** of the *stop.*
  - The **average velocity V** of a person walking: this is a parameter of the application and it changes in different contexts
  - The **measurement error E** of the input devise: in fact, when we consider the position of a *stop*, we must to be aware that the position is not accurate, because of the errors made by the input devise
  - The **average visit time AT** of each *point of interest*. This is also an application dependent parameter and represents the average time a person spends into a specific *point of interest*. We can extract this information by statistics.

Fuzzy geometries for POIs

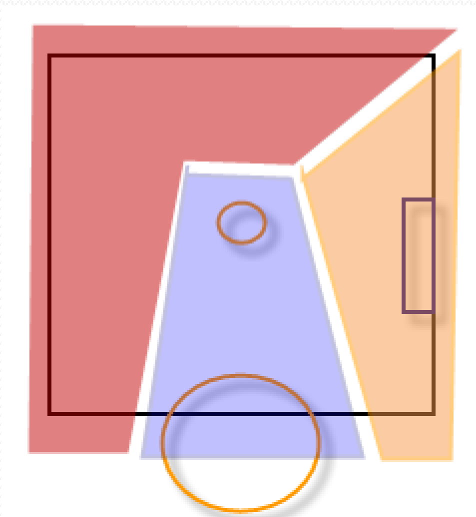
- Our POIs are points and not polygonal geometries: thus, for example, if a POI is a big square, we model that square as a point, so we have to decide which point of the square our POI represents and also which are the points around the POI that are interesting for our application
- Then we should consider that also the position of the POI can suffer of the measurement error
- We can solve these problems introducing the concept of "**fuzzy geometry**": we model the POI as a geometry, whose dimensions and shapes are decided by the user (note that the user can also choose dimensions equal to zero, that is to say that for that POI there is no uncertainty, i.e. it is a **crisp** point)

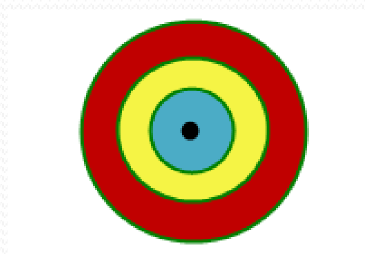
Fuzzy geometries for POIs

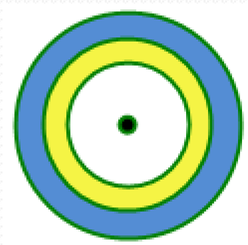
- Then we consider three fuzzy sets, “near” (blue), “medium” (yellow) and “far” (red), and three membership functions that specify the degree to which each point of the geometry (our Universe) belongs to a fuzzy set
- Obviously, **the boundaries of the considered sets are not precisely defined**

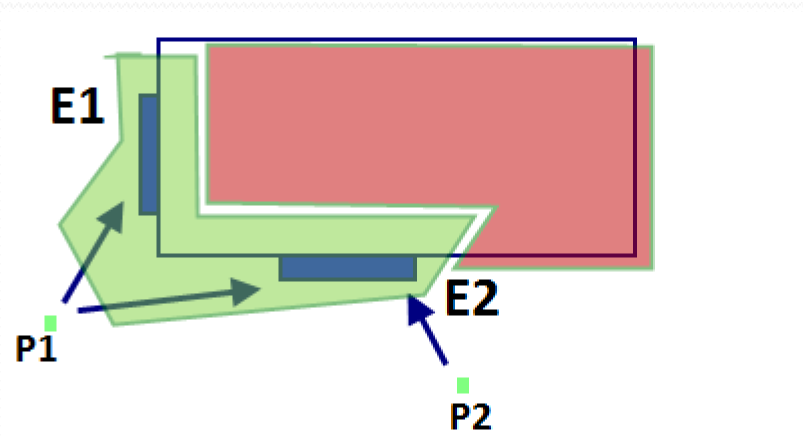


Fuzzy geometries for POIs

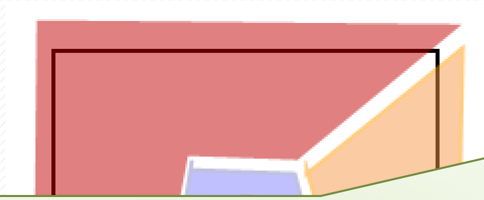


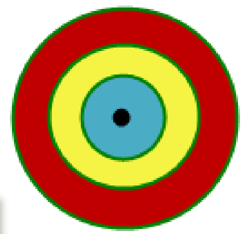






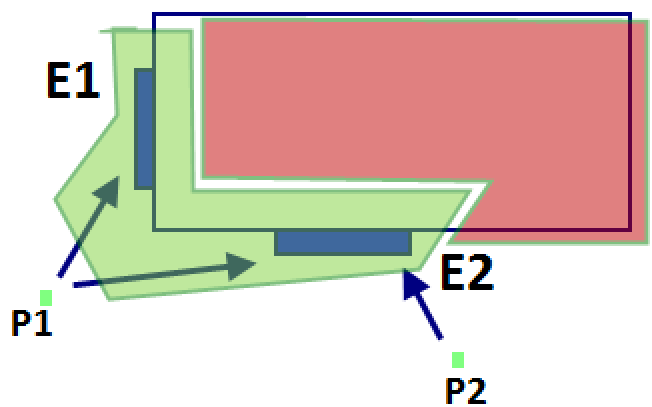
Fuzzy geometries



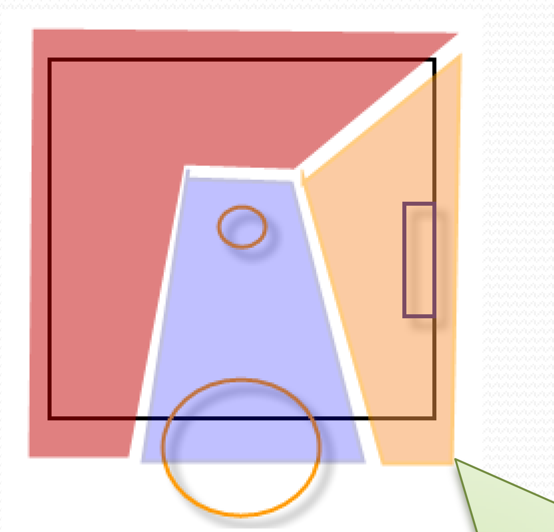


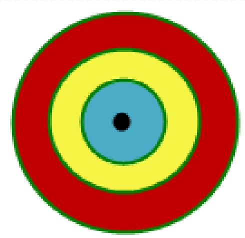
Fuzzy Circle: for a square, e.g., it means that out interesting area is around the POI, but we have to specify different levels of interest (the belongings of the points to the fuzzy sets) in order to cope with the measurement uncertainty and with the non-negligible size of the square.

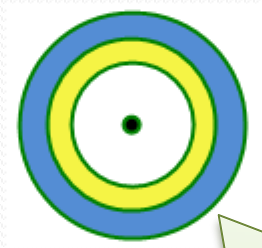
Local Fuzzy Geometry: The figure shows two entrances: E1 and E2. According to the position of the person P2 the point of interest is around E2 (we say “around” because of the measurement error), while according to P1 it is around E1 and E2. So we have a green area.

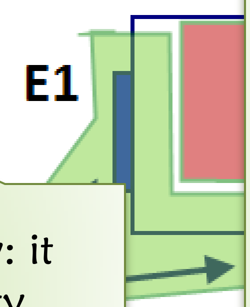


Fuzzy geometries for POIs







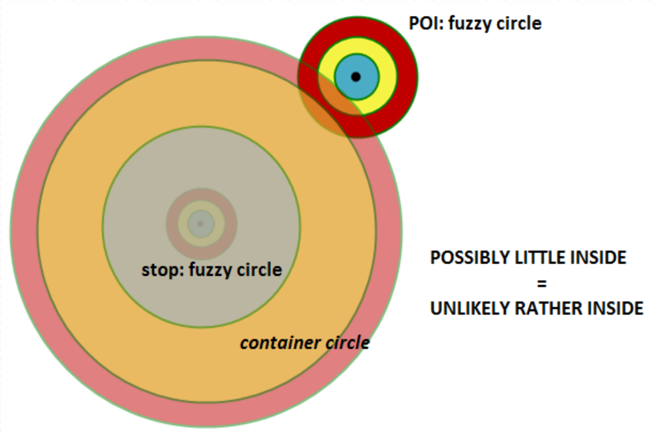


Distributed Fuzzy Geometry: it expresses the population density

Constrained Fuzzy Geometry: our area of interest is the external boundary of the place (e.g. a stadium), where there are the entrances. Then the application can take care about semantics and considering part of this area more probable (because of the ticket of a person). It is not a circle but a ring, because a person can't stay on the field (the center).

Association of POIs to Stops

- We model stops as fuzzy circles
- We compute the "container circle" using the maximum distance walkable by a person (it is also a fuzzy circle)
- We compute the possibility that a POI is associated to a stop on the basis of the covered area of the fuzzy POI (we have defined 11 inclusion relations)



The problem

- *"Given a trajectory described in terms of our conceptual model, we want to assign a probability to each POI to be the goal of the associated stop. In this way we can classify the trajectory, that is to say that we can assign a category to the trajectory on the basis of the POIs more probably visited by the moving person"*
- The **goal** of a *stop* is the *POI* visited by the moving person. This means that we will use an algorithm that assigns a probability to all the *POIs* of each *stop*: considering that each *POI* belongs to a category, the whole trajectory can be classified on the basis of the categories of the *POI* visited

The algorithm

- <u>INPUT</u>
  - A set of *Trajectories*, defined as a sequence of *Stops S*
  - A set of *POI* having a *position* <x,y>, an *average-visit-time* ***AT***, a category ***C*** and a fuzzy geometry
  - Each *Stop S* of each *Trajectory* must have a *duration* ***T***, a *position* <*x,y*>, a set of *POI* associated according to our conceptual model (each POI with a possibility to be associated to the stop)
  - The average velocity **V** of a walking person (application dependent parameter)

- <u>OUTPUT</u>
  - A probability for each *POI* of each *Stop* to be the *POI* visited in that *Stop*
  - A classification of all *Trajectories*: for each *Trajectory* we will have the most probable category with an associated probability

The algorithm

```
LIST<STOP> all = SORT(tutti gli Stop per numero crescente di POI)
LIST<STOP> past = null     //stop already visited

FOR EACH stop S OF all DO
        assignProbabilityUsingDistanceAndAverageTime(all)
        double threshold5 = 50+(50/size(L))
        double threshold7 = 90+(10/size(L))
        POI MP = getPoiHavingMaxProbability(L)
        IF(NOT(probability(MP)> threshold5) && size(past)!=0)
                THEN updateProbabilityUsingPastHistory(S, past)
POI MP = getPoiHavingMaxProbability(L)
IF(probability(MP)> threshold7)
                THEN backtracking(S,past)
past.add(S)
END FOR

assignProbabilityToTrajectory(all)
```

Classification Patterns

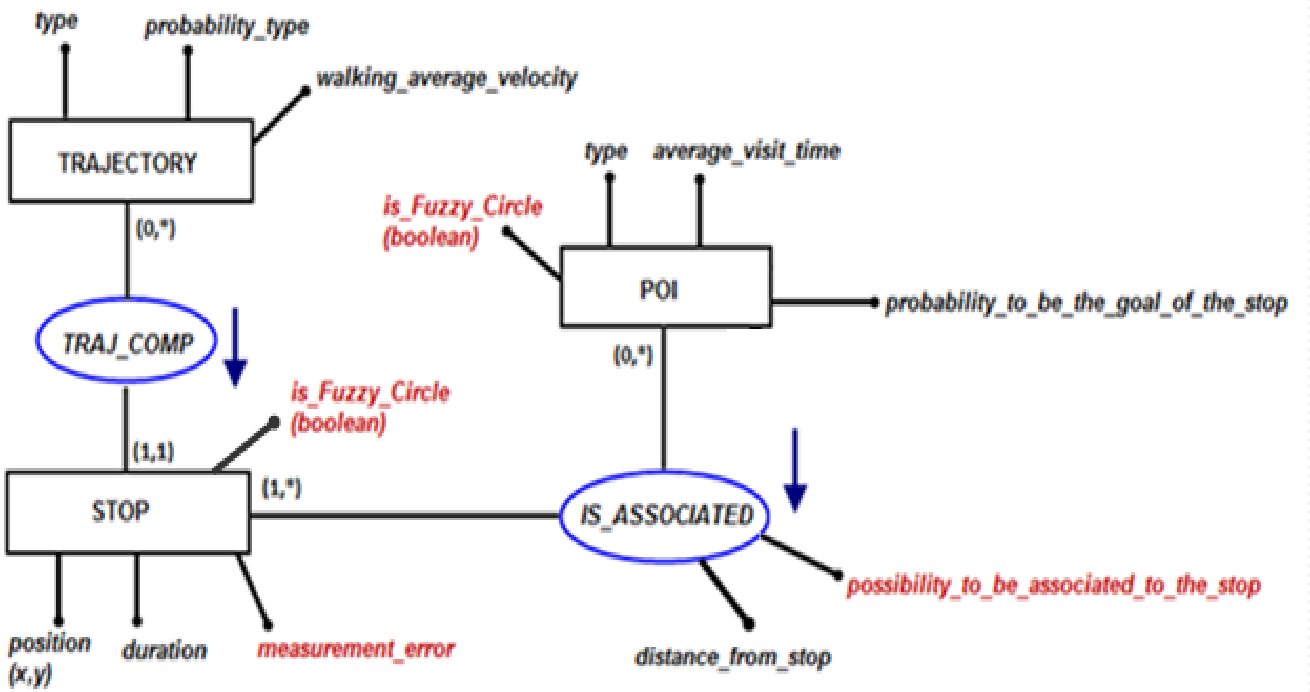
- Once we have yield an algorithm for trajectories classification, we can try to improve it by adding semantic information.
- We may think of a decision process, that is to say that, during classification, we can take care about some rules (or *patterns*) that help the classification of a trajectory.
- These **rules** are annotations that refer to the categories we are using for the current application and that say what can or cannot happen in a trajectory of a given category or in a stop with certain parameters.
- Rules are annotations associated to categories and they represent procedures that take in input some parameters and return a Boolean value
- In this way we can exclude those situations that don't satisfy our patterns: patterns can be used to bind the classification of a single stop or of a group of stops, as we will see next.

Rules

- **A-rules** Rules that bind the category of a single stop, independently from categories already assign to the other stops of the same trajectory
- **B-rules** Rules that foresee the category of the current stop on the basis of the categories assigned to past stops
- **C-rules** Rules similar to B-rules but that return also a probability
- **D-rules** Rules that are similar to C-rules but that don’t return a single value of probability, however a user-defined function used to compute a value of probability
- **E-rules** Rules that consider groups of trajectories, that is to say that they compare different trajectories of the same person

```
IF(poi_category_level1="TOURIST" AND
          (last_N_PAST_trajectory contains "LOUVRE MUSEUM") AND
          poi_category_level3="LOUVRE MUSEUM")
THEN RETURN MAX(50, [10+((N-1)*10)] % );
```

ER-schema of our application




Thank
You!