Uncertainty in trajectories classification v2.0

This document starts from the definitions and the algorithm presented in "An algorithm for trajectories classification" .

Fabrizio Celli
28/08/2009

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. There are a lot of kinds of uncertainty that can be analyzed:

- We can find uncertainty in the acquisition process of the raw components of a model (the extraction of the sampling points): though progress of technologies (such as GPS) and improvements in storage techniques have greatly improved the quality of spatial data, it is simply not possible to eliminate errors in data acquisition. This is called **measurement uncertainty.** In addition to the errors made when recording the sample points, to obtain the curve representing the trajectory we have to apply an interpolation method to that sample points (usually a linear interpolation), so the resulting curve will only be an approximation of the real trajectory: in literature this is called **interpolation uncertainty**. *We will give a representation of this kind of uncertainty for our application.*
- Due to continuous motion and **network delays**, the database location of a moving object will not always precisely represent its real location. We can model uncertainty and introduce new operators to answer to queries like "find the objects that are *possibly/definitely* inside the region *sometime/always* during the time interval".
- The nature of movements of a moving object is continuous and unpredictable, so these movements can't be precisely stored in a database. Spatio-temporal trajectories require a **geographical abstraction** and it is highly unlikely that geographical complexity can be reduced to models with perfect accuracy. *We will analyze the state of art.*
- 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**. Beside the approximation problem, there is the problem of **compression**: compression is necessary to increase the response-time of queries when the amount of data collected by positioning devices is too big. This operation can lead to a loss of information in order to eliminate some redundant or unnecessary information, which increases the amount of errors.
- 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.

- As well as we are developing a conceptual model, another kind of uncertainty can be found **inside the model** itself: for example, in our application there is uncertainty about the POI visited by a person during the stop (*our algorithm aims to solve this problem, as we showed in the last document*).

MEASUREMENT AND INTERPOLATION UNCERTAINTY

In our approach we have chosen a representation model for measurement and interpolation uncertainty: in fact, as we have already explained in the first document, we model a *stop* as a circle, which radius is the known maximum measurement error of the input devise. In this way, the area of the circle is the place where our *stop* is contained.

This approach is similar to the one proposed in [G. Trajcevski et al. Managing uncertainty in moving objects databases]: that work introduces a threshold **r** that denotes the maximum distance of the point to the assumed location of the trajectory. Thus, each point *(x,y,t)* of the trajectory becomes the center of an horizontal disc of radius equals to the threshold and the trajectory is modeled as a sheared **cylindrical volume** in 3D space around the given trajectory polyline. This is shown in figure 1.

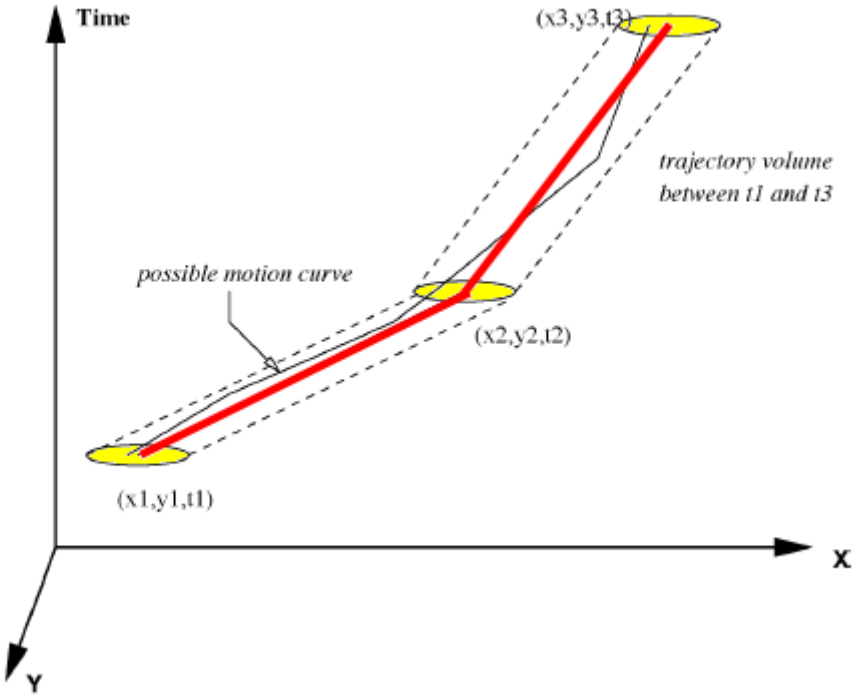


Figure 1: solution proposed in [3]

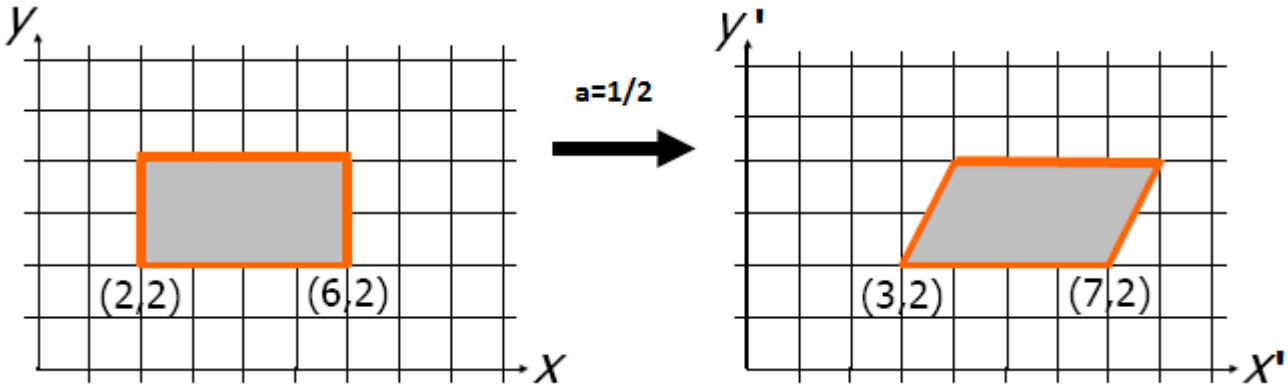
Looking at figure 1 we can observe that here it has been modeled only the uncertainty in space: in fact, if we want to model also uncertainty in time, each point *(x,y,t)* of the trajectory becomes the center of a cylindrical volume of radius equals to the chosen threshold, while the whole trajectory is given by the composition of all these cylinders translated in time and space. We could do other two observations:

- First, we can choose a different threshold for each point, in order to have cylinders different in radius. In this way we model uncertainty in space with more precision, even if the computational cost increases.
- Then, we must observe that, modeling also uncertainty in time, for each point we won't have a perfect cylinder, but a sheared one. In fact, uncertainty in time means that we are not sure about the time in which we do the measurement, therefore the measure could be carried out in an instant belonging to a time interval. But the monitored object is a moving object, so considering a time interval we must consider also a translation in space. It is easier to look at this situation in 2D. Starting from a rectangle (the projection of a cylinder in 2d) we apply the following transformation:

  $x' = x + ay$

  $y' = y$

  where (x',y') is the new coordinate system, x represents the space and y the time. The height expresses the uncertainty in time while the value "a" the fact that the monitored object is a moving object.



A more complex solution to the problem of measurement and interpolation uncertainty has been proposed in [B.Kuijpers, W.Othman. *Trajectory Databases: Data Models, Uncertainty and Complete Query Languages*] and it has been called "*bead model*". Here the uncertainty associated with the location of an object travelling between two endpoints of a line segment can be an ellipse

with foci at the endpoints. The problem is that this model works under the assumption that we must know an upper bound for the object's speed between sample points in order to calculate 3D cones centered in each sample point: the intersection of two cones is an ellipse. Bead model reduces the uncertainty by a factor 3 (a cone as 1/3 volume of its minimal bounding cylinder) but is not easy to handle and to query; moreover it is no easy to find an upper bound for velocity. Figure 2 shows this model.

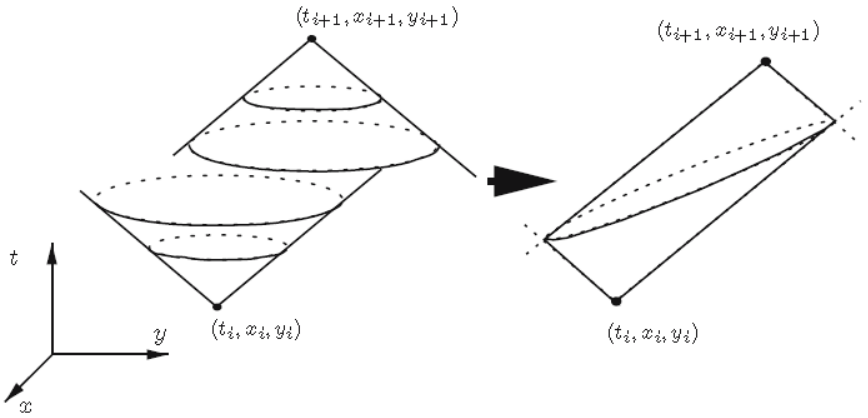


Figure 2: the bead model

We can improve our representation of uncertainty about the position of the moving object by considering the following aspect:

- For each stop we can state a different measurement error **E**, so each stop will be modeled with a circle of different radius and area. The reason of this assumption is that we can use different devises to capture the position of a moving objects and each devise can have a different measurement error. Another reason may be the fact that for a stop we have more certainty about its position, for example because it is delimited by natural or artificial boundaries.

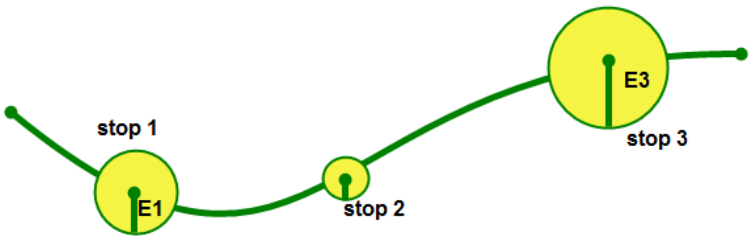


Figure 3: the stops representation

As we will see in the following sections, we can transform these crisp circles in fuzzy ones.

The article [M.Sabrakos, *Fuzzy Interoperable Geographical Object*] shows how to integrate the *fuzzy set theory* and the *object data modeling* to create a new approach of enhancing spatial objects. But a very interesting part of this article is the one that classifies the fuzziness in a spatio-temporal point of view. As far as time concerns, we have:

- Fuzzy time points: a time point become fuzzy if it is not known exactly when an element has a location. Such a point is delimited by two time elements and is defined by fuzzy functions.
- Fuzzy time period: it is a subset of the tile line, bounded by two time points.

In our work we don't use a fuzzy approach for uncertainty in time, but we model it as a cylinder over the fuzzy circle representing the uncertainty in space.

As far as space concerns, spatial fuzziness occurs when it is not possible to define an area with crisp boundaries or when it is not possible to describe precisely spatial relationships between objects. A spatial point can be only crisp or uncertain (i.e. its location is not known exactly): **in our work we model uncertain points as fuzzy polygons**. The article shows:

- Uncertain polygon: it's a polygon whose boundaries are not exactly known. We need a probability function to determine the probability an arbitrary point belongs to the actual boundary line.
- Fuzzy polygon: it's a polygon whose boundaries are not crisp but transitional, characterizing areas that for some reasons can't have (or doesn't have) sharply defined boundaries. **In our work we follow this approach.**

FUZZY GEOMETRIES FOR POIs

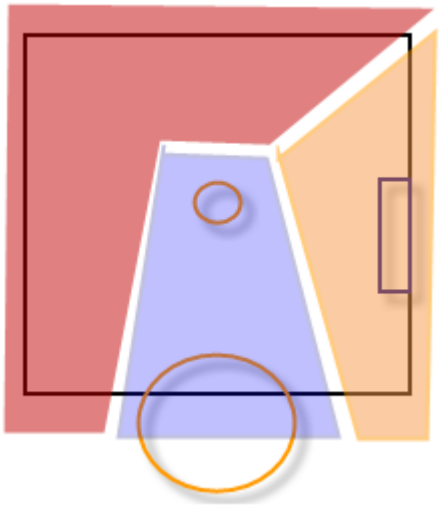
We can model also the **uncertainty about the position of the POIs**: up to now we have considered their position as precise. First of all we must say that our POIs are points and not polygonal geometries: thus, for example, if a POI is a big square, we model that square as a point, so we have to decide 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) and that for example can represent the dimensions of a square or the maximum measurement errors of the input devise. Then we introduce a **probability function** that assign to each point of the geometry a probability included in the real interval [0,1] that expresses the probability of that point to be our POI: this probability decreases with the increase of the distance from the center of the geometry. We can improve the definition of the probability function by considering the *fuzzy set theory*: 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**.

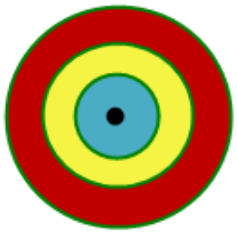
Maybe we have to specify better what we mean for "fuzzy geometry". In our database, POIs are points and, together with the measurement errors, this is a source of uncertainty: in fact, real objects can have a non-negligible size, so we can be interested not only in a single point but in an area and, moreover, we could not be sure about the precision of the measured point. Thus the user can choose to model this uncertainty by specifying for some (or all) POIs a geometry shape and a set of membership functions that map our fuzzy sets on this shape: this is a fuzzy geometry. For example, let us consider "*Piazza della Rotonda*" in Rome, the square where is situated the *Pantheon*. In the center of the square there is a fountain with an obelisk, in one side of the square there is the Pantheon, in another side there is an important bar, "bar Pantheon". We can model this square as a rectangle, having as center the position of the POI registered in our database, and we can create the membership functions in this way:

- The "near" fuzzy set covers the center of the square and the size with the Pantheon
- The "medium" fuzzy set covers the side with the bar
- The "far" fuzzy set covers the rest of the square

Here the membership function are simple mappings from a set of points of the square to a number, the possibility. We call this model as **distributed fuzzy geometry**.

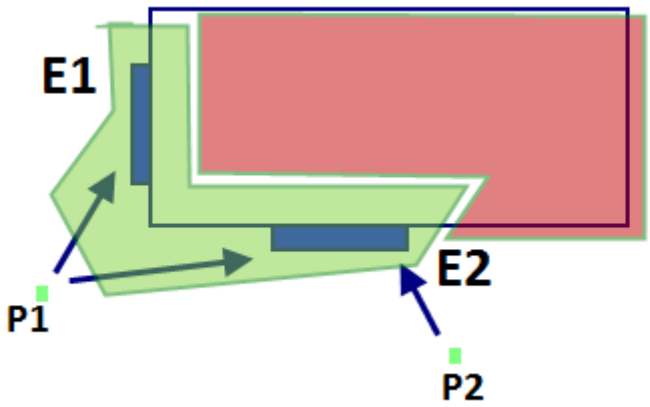


Things are simpler if we consider "*Place de la Concorde*" in Paris. This square has an obelisk in the middle while, around it, there is the street: so people will stay more probable in the center of the square. If we consider our POI as the center of the square, we can model the square with a "**fuzzy circle**" as following:



This fuzzy circle means that out interesting area is around the POI (i.e. the center of the square), 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.

We can identify other kinds of fuzzy geometries. For example, we could have a place of interest with a limited number of physical entrances, delimited by artificial boundaries. For example, if our place of interest is a great building with two entrance, the position of our POI depends also from the position of the moving person:



The figure shows a rectangular building with two entrances: E1 and E2. In correspondence of E1 and E2 we have together the two fuzzy sets “near” and “medium”, so we have a green area (yellow + blue): in fact, 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. We call this model as **local fuzzy geometry**.

A variant of the previous situation is the one in which the entrance is constrained by semantics: for example, if the building is a stadium, a person usually can’t enter from any entrance, but she has to follow the instructions on her ticket.

The example shows the “*Stadio Olimpico*” in Rome: here we can state that our area of interest is the external boundary of the 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). We call this model as **constrained fuzzy geometry**.

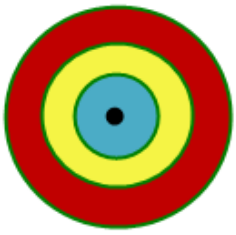
Now that we have analyzed some fuzzy geometry, we can answer the following question: *how will the user assign a fuzzy geometry to all POIs?*

As basic case, the user can decide to have all crisp point, i.e. she doesn't model the uncertainty. Then the user can choose a default geometry in order to model all the POIs of a certain category with such geometry: in fact we have to remember that each POI belongs to a set of categories, so the user can assign a geometry to a whole category. The user can choose different fuzzy geometries for different categories. Once this is done, the user can refine the model by assigning specific fuzzy geometries to specific POIs: for example, suppose that for a possible category "Square" (we have never used this category before) the user has chosen the "fuzzy circle", then she can choose to model the instance "*Piazza della Rotonda*" with a "distributed fuzzy geometry". In this way we can model the uncertainty in a very general way and then consider only the main instances (where "main" depends on the user).

FUZZY POINTS FOR STOPS

We have already discussed about the uncertainty of stops: we model stops with circles of radius equals to the known measurement error. We can use different radiuses for different stops because for some stops we can be more sure about their position or because we use different measurement devises.

Now we want to apply the fuzzy theory also to stops: instead of having crisp circles, we can model a stop with a fuzzy circle, where the measured point is the center, the radius is given by the user as it happened before, while the membership functions are related to the distance from the center. This means that we will have a small circle area around the center that will contain almost certainly the real position of the stop (the "near" fuzzy set) and other two rings around it having low possibility to contain the stop (low but not zero).



Moreover we can do another consideration. As we saw in the first document (*An algorithm for Trajectory Classification*), when we compute the distance between a POI and a stop we first map our trajectory on a road map to compute the real minimum distance between them: this road map (such as a Google Map®) contains information about the environment, so we can constrain fuzzy circles using the environment. Thus, if a part of the fuzzy circle overlaps a building, a river or another obstacle, we can cut that part because certainly the position of the stop can't be there. In this way we further reduce the uncertainty about the position of the stop.

The document [N. Hung, S. Spaccapietra; *Trajectory Data Cleaning*] studies an algorithm that maps a trajectory on a road map, repairing situations in which, for example, a point goes outside a physical or natural street. The paper [S.D.Prager; *Environmental Contextualization of Uncertainty for moving objects*] uses an ontology for reallocation of positional probability through the environmental contextualization, where obstacles like buildings reduce probability, while objects like roads increase it.

CHANGINGS IN THE ALGORITHM

Now that we have modeled the POI using the fuzzy set theory (and also the stops), we can do a new consideration about the association of POIs to a stop. As we have seen in the previous documents, we compute a "maximum distance walkable" in order to obtain a "**container circle**" - centered in the stop – that contains all POIs to associate to the stop:

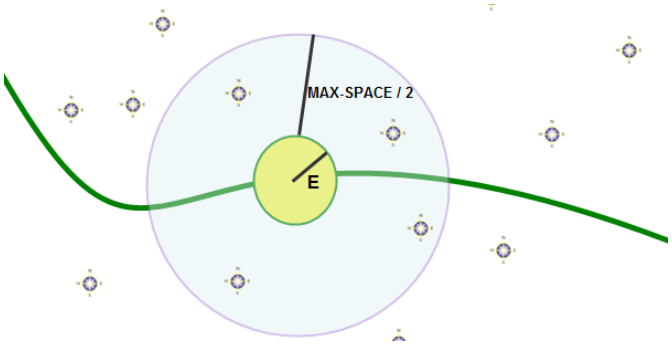


Figure 5: the container circle

But now a POI is no more a single point, so we have to refine the inclusion relationship of a POI in this container circle. We have to introduce the concept of possibility: in fact, if the POI is a point, the inclusion relationship says only if it is TRUE or FALSE that the POI is associated to a stop; but now that a POI is a fuzzy geometry, we have to talk about the *possibility* that the POI is associated to the stop:

- If the fuzzy geometry of the POI is entirely contained into the "container circle", then the POI can be associated to the stop with a *possibility* equals to 1.
- If the fuzzy geometry of the POI is entirely outside the "container circle", then the POI can be associated to the stop with a *possibility* equals to 0, that is to say that it is not associated to the stop.
- If the fuzzy geometry of the POI is partially contained into the "container circle", then the POI can be associated to the stop with a *possibility* that depends on the fuzzy set covered by the "container circle": in fact, the "container circle" can overlap the "near" set, the "medium" one or the "far" one, so we will assign a different possibility for each of them. This possibility arises from the membership functions for the covered fuzzy sets. In fact we can consider the points in the covered area and we compute from their membership functions the degree $\mathbf{d_S}$ to which they belong to the fuzzy set **S**: we use these degrees to compute the possibility **P** as we are going to explain. For simplicity, in the next figures we will draw POIs as fuzzy circles and stops as crisp circles.

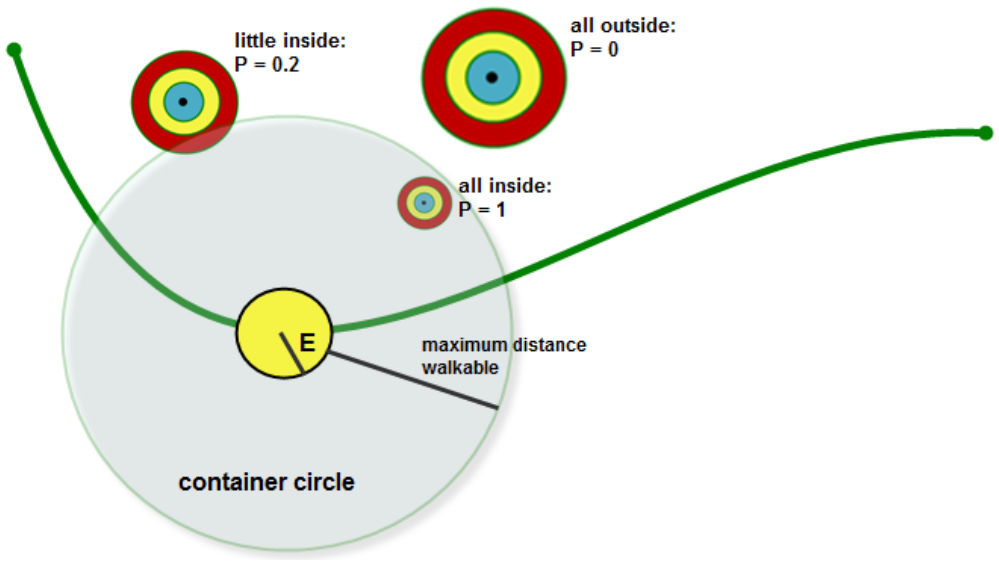


Figure 6: the new criterion for POIs association to stops

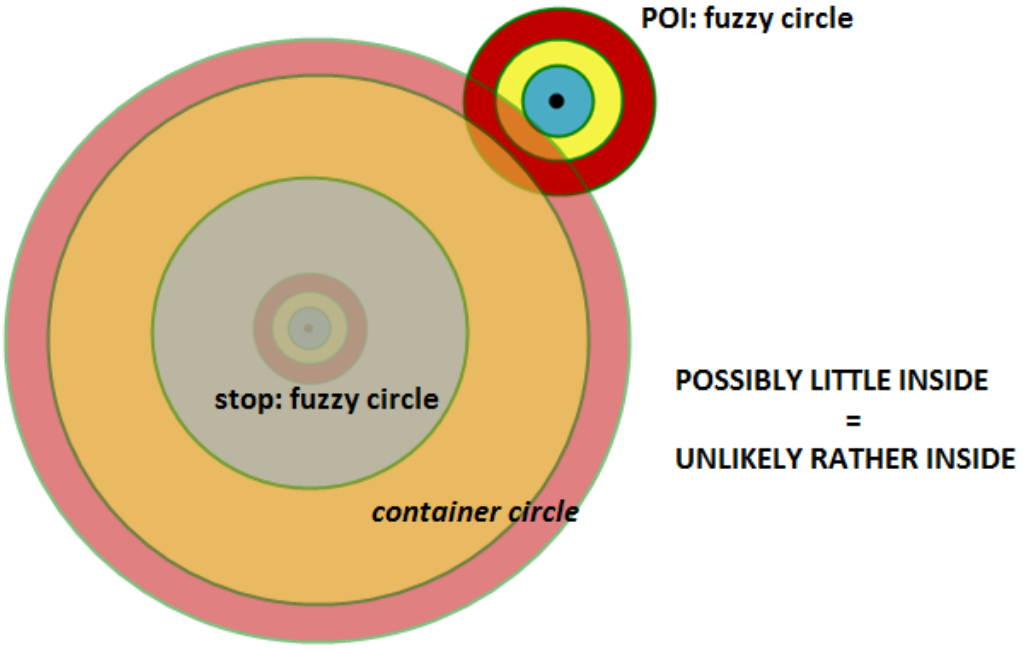
In this way, when we associate a set of POIs to each stop of the current trajectory, we can define five inclusion relations:

- ALL INSIDE: means that the POI is entirely inside the container circle. Here the possibility that the POI is associated to the stop is 1, that is to say that there is no uncertainty.
- ALL OUTSIDE: means that the POI is entirely outside the container circle. Here the possibility that the POI is associated to the stop is 0, that is to say that the POI is not associated to the stop and there is no uncertainty.
- **LITTLE** INSIDE: means that the circle covers only the "far" fuzzy set of the POI. We can assume this fact only if the points in the covered area show to belong more probable to the "far" fuzzy set, that is to say that $\mathbf{d_{Far}}$ is the biggest degree returned by their membership functions. We can assign here a small possibility, for example included between 0.1 and 0.4, depending on the value of $\mathbf{d_{Far}}$: if $\mathbf{d_{Far}}$ is equal to 1 we can assign 0.1, because it is unlikely that in the covered area there is the point of interest; if $\mathbf{d_{Far}}$ is next to 0 we can assign 0.4, because the covered area contains also to the "medium" fuzzy set with a high degree.
- **RATHER** INSIDE: means that the circle covers also the "medium" fuzzy set of the POI. In this situation the possibility is bigger than in the last one, so for example we can have p=0.5 or 0.7 if the covered area also contains the "near" fuzzy set.
- **WIDELY** INSIDE: means that the circle covers also a part of the "near" fuzzy set of the POI. Here the possibility can be, for example, 0.8.

To say the truth, there are other relations to be considered. In fact we model the stop as a fuzzy circle, thus also the "container circle" should be a fuzzy circle. The relations ALL INSIDE and ALL OUTSIDE don't change, but the other three relations have to be reconsidered. For example, as far as the relation LITTLE INSIDE concerns, we have three new operators:

- **UNLIKELY** LITTLE INSIDE: means that the a big area of the "far" part of the fuzzy container circle covers only the "far" fuzzy set of the POI. Here it is also possible that a small area of the "medium" part of the fuzzy container circle covers the "far" fuzzy set of the POI, in which case the resulting possibility will be bigger.
- **POSSIBLY** LITTLE INSIDE: means that the a big area of the "medium" part of the fuzzy container circle covers the "far" fuzzy set of the POI.
- **DEFINITELY** LITTLE INSIDE: means that the a big area of the "near" part of the fuzzy container circle covers the "far" fuzzy set of the POI.

We can apply these three operators (unlikely, possibly and definitely) also to the relations RATHER INSIDE and WIDELY INSIDE. Note also that some of these relations are dependant each other. For example, sometimes POSSIBLY LITTLE INSIDE and UNLIKELY RATHER INSIDE can overlap: it is the case of a POI modeled as a fuzzy circle, where if a big area of the "medium" part of the fuzzy container circle covers the "far" fuzzy set of the POI (POSSIBLY LITTLE INSIDE), then also a big area of the "far" part of the fuzzy container circle covers only the "medium" fuzzy set of the POI (UNLIKELY RATHER INSIDE). Obviously, using other geometries for POIs will change these dependences.



It's better to understand the meaning of possibility values. For example, what does it mean that a POI has a possibility of 0.5 to be associated to a stop?

- Considering the classical probability it means "maybe", that is to say that if we repeat infinitely the measurement of the positions of the stop and the POI, we will assign the POI to the stop half times.
- If we read that value as a "measure", it means that half POI is associated to the stop.
- Instead, considering the **fuzzy logic**, means that we can associate something to the stop, maybe not the POI, but a view of it or a knowledge about it (like it's type). Referring to the real life, it means that maybe there is the POI in that position or that the POI is observable from that position.

Once we have modeled POIs as fuzzy circles and have chosen a group of fuzzy sets with associated membership functions, our conceptual model changes as expressed in the following sentence:

- **Each stop of a trajectory has a set of POIs associated, each one with a known possibility that expresses the uncertainty about the association between a POI and the stop.**

Considering this new definition, also the algorithm that classifies trajectories will change: in fact, when we compute the probability of a POI to be the goal of a stop, we must take care about the *possibility* attribute of the POI to be associated to the stop. If this possibility is equals to 1 or 0, the computation will not change. But if it is included in the real interval (0,1) we have to combine it and the probability computed by the algorithm: in this way, POIs having a small value of possibilities will see reduced their probabilities to be the goal of the stop.

The last comment concerns the modeling of the POIs. As we have already said, the user can choose if a POI is a fuzzy geometry or a crisp point. The motivation for this representation is trivial: in fact a POI can be for example a bus station, so it can be modeled as a crisp point if we consider its position as precise; but a POI can be also "place de la Concorde" that covers a huge area, so we have to decide which part of the area is of our interest. Moreover, the choice for a fuzzy geometry arises also from the possible measurement error doing during the registration of the position of a POI, even if sometimes we can consider that position as precise because the POI is a well known POI.

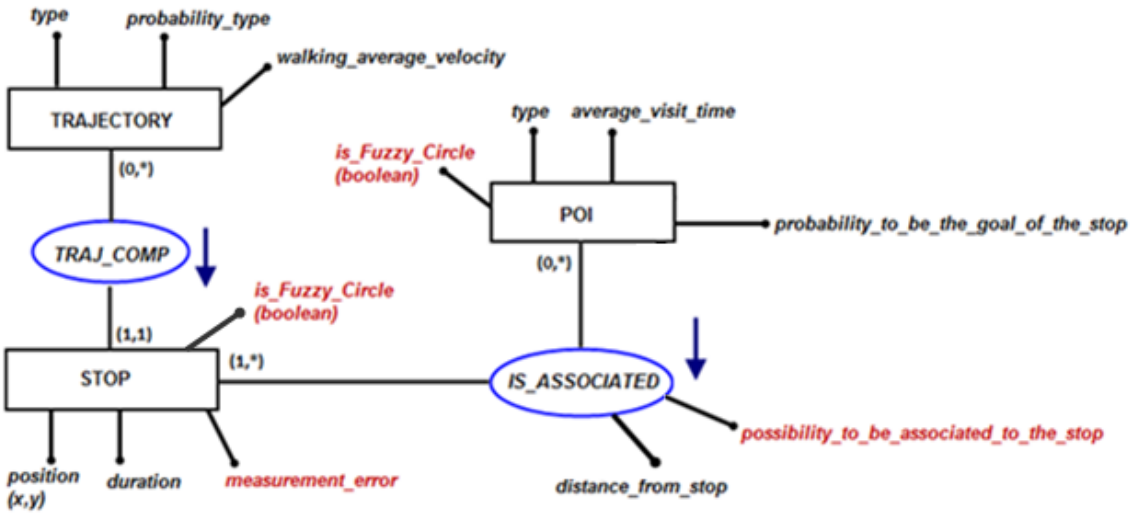


Figure 7: the ER schema of our application

GEOGRAPHICAL ABSTRACTION: STATE OF THE ART

Spatio-temporal trajectories require a geographical **abstraction** and it is highly unlikely that geographical complexity can be reduced to models with perfect accuracy. The inevitable discrepancy between the modeled and real worlds constitutes uncertainty and inaccuracy. Moreover abstraction may be affected by an imperfect observation of the real world.

In the paper [A.U. Frank, *Data quality ontology*] the author try to build up an ontology to classify the imperfections of data and formalize the influence of data quality on decisions. To say the truth, he doesn't build am ontology, but a simple classification of uncertainty in the following categories:

- Limitation to partial knowledge of the world and of the context
- Observation error
- Simplification in processing

This article gives us a set of commitments to cope with these (few) kinds of uncertainty, but it does not propose any solutions.

A problem that arises from the book [J.Zhang, *Uncertainty in Geographical Information*] is the **ambiguity:** together with vagueness, it reflects the inherent complexity of geographical world and the subjectivity involved in a high proportion of geographical data. In this book, the author classifies geographical uncertainty in this way:

- Approximation, due to the complexity of the real world and the limited storage space of a computer;
- Interpretation given by the user to data stored in a database;
- Perception of the world from the user viewpoint;
- Errors in data acquisition;
- Ambiguity, vagueness and Lack of information.

In this book there is an attempt to detect and measure uncertainty, in order to assess accuracy levels in geographical data, and then to analyze uncertainty propagation in order to predict the consequences of using a particular set of data for a specific purpose. It is a complete work, that gives the idea of geographical uncertainty and of the techniques to cope with it, also at a conceptual level. The only lack is that it doesn't give a precise classification of uncertainty: it introduces some types of it with examples, without trying to classify internally these types.

This classification is instead made in the article [H.Shu, S.Spaccapietra et al., *Spatio-Temporal Uncertainty modeling in databases*]. This article states that Geographical Uncertainty results from the differences among human cognition, computer representation and geographical reality: this is the Human-Machine-Hearth System (HMH). Generally, Geographical Information Uncertainty reflects the richness of geographic states, the inability of human cognition (e.g. visual haze, lack of knowledge, undecidability) and the limited computing capacity of machine. The following figure shows a classification of this type of uncertainty, produced in the aforesaid article:

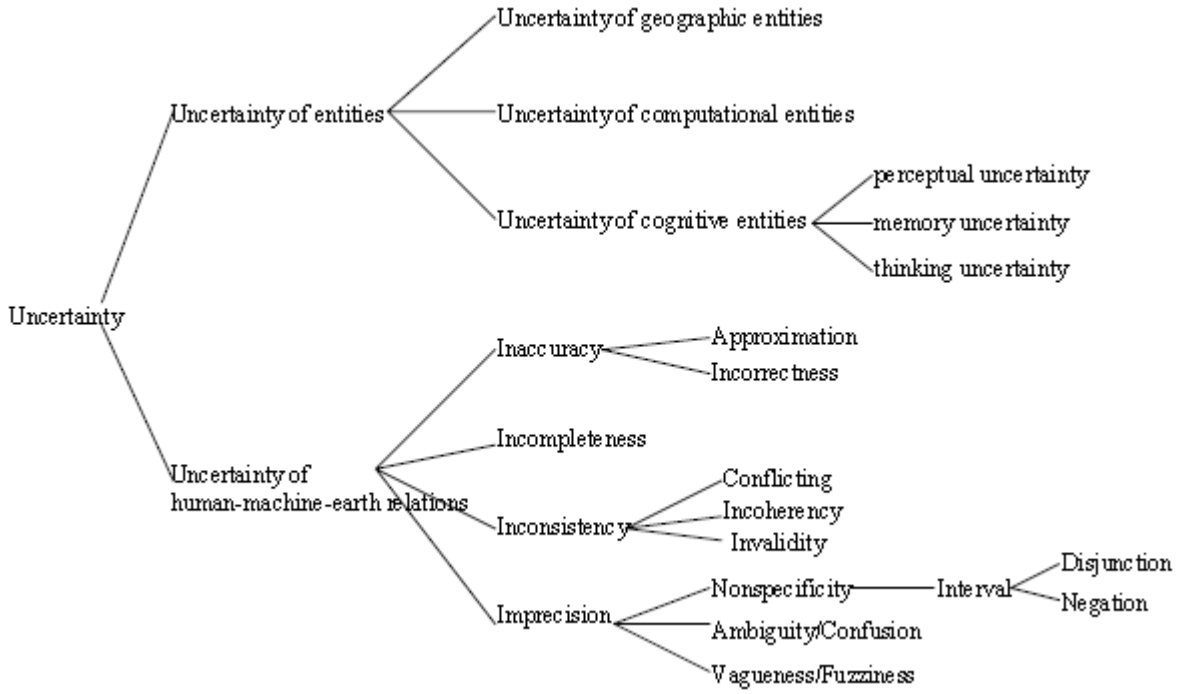


Figure 8: a classification of geographical uncertainty

where "uncertainty of entities" arises from the differences among entities **within** human cognition or machine or the earth, while "uncertainty of HMH relations" arises from the differences **among** cognitive, computational and geographic entities. We could spend some words for the meaning of the categories for "uncertainty of HMH relations":

- Inaccuracy refers to deviation of our value from the true value, so it represents an error.
- Incompleteness means that some value is missing.
- Inconsistency means that, for the same geographical entity, there are many different computational and cognitive statements.
- Imprecision refers to the degree of exactness of computational and cognitive values: ambiguity is imprecision with a low resolution and it's a formalization of the ambiguity discussed by [J.Zhang].

SUMMARY

KIND OF UNCERTAINTY	DESCRIPTION	SOLUTION
Measurement Uncertainty	Uncertainty that arises from the acquisition process, that is to say from the extraction of sample points. The measurement input devise can't be perfectly precise, so the values extracted are similar but not equal to the real ones.	We model stops as fuzzy circles, whose radius represent the known measurement error. We model POIs as fuzzy geometry, defining some fuzzy sets.
Interpolation Uncertainty	The trajectory has to be built from a set of sample points (the positions of the object) and the curve is obtained by applying interpolation methods on that set of sample points (e.g. linear interpolation or with Bezier curve). So the resulting curve will only be an approximation of the real trajectory.	We model stops as fuzzy circles, whose radius represent the known measurement error. We model POIs as fuzzy geometry, defining some fuzzy sets.
Network Delay	All information is transmitted through a network, that is characterized by delays, due for example to congestions or speed limits. Thus, we have uncertainty between the measured value and the time of measurement.	We can introduce new operators to answer to queries like "find all the objects that are *possibly-definitely* inside the region *sometime-always* during the time interval".
Geographical Abstraction	We have to model the real world but it is highly unlikely that geographical complexity can be reduced to models with perfect accuracy	
Raw Problem	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.	
Semantics	The interpretation given by the user to the information stored in a database is another source of uncertainty.	Valid documentation.
Application Dependent	Uncertainty connected to the application under analysis. For the application of classifying trajectories we have uncertainty about the POIs visited by a person during a stop	A probabilistic algorithm that assigns a probability to all POIs of a stop to be the goal of that stop.