



**HAL**  
open science

# Querying indoor spatio-temporal data by hybrid trajectories

Franck Jeveme Panta, Florence Sèdes

► **To cite this version:**

Franck Jeveme Panta, Florence Sèdes. Querying indoor spatio-temporal data by hybrid trajectories. 8th ACM International Workshop on Indoor Spatial Awareness (ISA 2016), ACM SIGSPATIAL, Oct 2016, Burlingame, California, United States. pp.11–18, 10.1145/3005422.3005424 . hal-03625887

**HAL Id: hal-03625887**

**<https://hal.science/hal-03625887v1>**

Submitted on 31 Mar 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



## Open Archive TOULOUSE Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible.

This is an author-deposited version published in : <http://oatao.univ-toulouse.fr/>  
Eprints ID : 18814

The contribution was presented at ISA 2016 :  
<https://sites.google.com/site/indoorspatialawareness2016/>

**To cite this version** : Jeveme Panta, Franck and Sèdes, Florence *Querying indoor spatio-temporal data by hybrid trajectories*. (2016) In: 8th ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness (ISA 2016), 31 October 2016 (San Francisco Bay, CA, United States).

Any correspondence concerning this service should be sent to the repository administrator: [staff-oatao@listes-diff.inp-toulouse.fr](mailto:staff-oatao@listes-diff.inp-toulouse.fr)

# Querying indoor spatio-temporal data by hybrid trajectories

Franck Jeveme Panta  
IRIT, Université Paul Sabatier  
118, Route de Narbonne  
31062, Toulouse cedex 9

Franck.panta@irit.fr

Florence Sèdes  
IRIT, Université Paul Sabatier  
118, Route de Narbonne  
31062, Toulouse cedex 9

Florence.sedes@irit.fr

## ABSTRACT

GPS has become the reference for outdoor positioning, implementing a direct connection between the GPS satellite and a receiver device. Indoor positioning raises new challenges, locating a target device requires the wireless sensors networks and other technologies. Sensor networks deployed in buildings are commonly used for many applications based on location: surveillance, detection, navigation, etc. These indoor locating sensors generate lot of data related to tracking information. Exploiting this information for investigation issues for example remains a relevant purpose. The context of this paper is related to indoor locations systems based on wireless cell, ICCARD sensors and video surveillance cameras. In this context, as no global reference system similar to GPS is available, the location information issued from various systems, platforms, devices, etc. have neither standards nor common formats, and remain heterogeneous. This heterogeneity is mainly due to the different types of positions (geometric, symbolic, etc.) expressed w.r.t. various reference systems. In order to manage them in a given framework, it is necessary to homogenize the relevant (Meta) data to process the global knowledge they can give. This paper presents a contribution to extend our framework [1] to information generated by location sensor networks deployed in an indoor environment. The use case is illustrated in a forensic application [2].

## Keywords

Indoor location, spatio-temporal data, hybrid queries, forensic.

## 1. INTRODUCTION

Providing users, via spatio-temporal queries, with geo-located data based on sensor networks captures aims at locating, identifying, extracting metadata, from connected devices or objects in a given area, at a distance from other objects or sensors, at crossing paths, etc. Such applications are based on object positioning that can be expressed according to two ways, depending on different reference systems: geometric (coordinates relative to a reference system as geodesic datum, map of a building, etc.) and symbolic (more semantic description related to points of interest, address, transport networks, parts of a building, etc.). Coping with these positions, processing predicates related to

a common reference highlight how the gap between the two ways of expressing positions is accurate, a geometric position being expressed by a point while a symbolic one is usually represented by geometry (a room, a building or an RFID cell).

In outdoor environments, applications use in most cases location models based on GPS sensors that are widely embedded (cars, smartphones, etc.). The indoor environments are becoming more complex and there is no single reference system to which any measure can refer to: company buildings, subways infrastructure, airports, etc., generate a significant need to manage indoor location in terms of applications.

Facing the lack of satisfactory and generic answer, various applications have been developed in our team about outdoor location problems, in particular in forensic-oriented projects [2], [3]. These applications led us to the need for extending it to indoor environments with sensors such as Wi-Fi hotspots, ICCARD readers and video surveillance cameras. The main problem in the indoor environment is the heterogeneity of location information given by the various sensors. This heterogeneity concerns the following cases:

- positions are either geometric or symbolic: while the location based on the wireless gives geometric positions relative to the 2-dimensional coordinate system, the location based on the cells in the case of the cellular network gives symbolic positions, that is also the case with technologies such as infrared (IR) and RFID (Radio Frequency Identification),
- positions are expressed according to different coordinate systems.

Our work focuses on these two points and consists in checking that the generic modeling we propose can be extended and that our trajectory query framework can integrate indoor environment.

This paper is organized as follows: after a review of related work on indoor location models, trajectories and types of spatial queries in section 2, section 3 presents the requirements for a heterogeneous trajectory base query framework and section 4 states the definition of a model of hybrid trajectory based query. Section 5 presents indoor coordinates system. The different algorithms are developed in section 6. In section 7, the proposed architecture is illustrated within our framework and finally, Section 8 concludes and discusses future researches.

## 2. STATE OF THE ART

### 2.1 Indoor location models

Indoor location aims at defining a user's/object position in an indoor environment. An indoor environment is generally made of

a three-dimensional space (2D + floor for example): building in which people live, work, transit, and usually spend their time (houses, shopping centers, companies, stations, etc.) [4]. To represent this space, spatial models based on the definition of a topology model that translates space have been proposed, defining the location of objects and trajectories. This representation is called location model. [5] defines a location model as an expressive, flexible and efficient representation of location information.

The indoor space is partitioned either using geometrical methods that generate a space division in uniform or non-uniform cells [6], [7] or using symbolic elements, most of the time related to the topological elements of the building (rooms, hallways, doors, floors, entrances, etc.). In our case, to be consistent with the use cases our applications state, we only consider approaches relying on building structure elements for space partitioning, since human trajectories are mainly guided by these structures. Based on this space partitioning, geometric and symbolic location models are defined.

### 2.1.1 Geometric location models

Within geometric location models, the space (or partitions of the space) is seen as an Euclidean space so entities locations as a 2D (X, Y) or 3D (X, Y, Z) set of coordinates with regards to global or local reference systems [8]. The main advantages of these kinds of models are the location high accuracy (ignoring the sensor measurements errors) and the straight forward computation of distances. Their main drawbacks are related to the non-representation of topological relation (e.g., connections between different rooms) and the fact that this representation is non-intuitive for users.

### 2.1.2 Symbolic location models

In symbolic location models, positions are identified by names or abstract symbols (e.g. "Room C1", "Scale S9") [9]. Many research works addressed the problems of lack of topological relations and of semantics of the geometrical location representations. These drawbacks are addressed by symbolic location modeling based on much more semantic descriptions about moving objects based on structural entities and/or points of interest (e.g., room/floor identifier, building name, etc.). The drawbacks of this type of models are that: (i) the accuracy depends on the level of abstraction of the space model, (ii) the model is less suitable for distance computation, (iii) the choice of named locations depends on the application (lack of interoperability), (iv) the modeling effort is sometimes valuable (symbols for locations often needing to be "manually" created and managed).

Some works also combine advantages from both geometrical and symbolic models and propose hybrid location models [10], [5].

Hybrid models are combinations of geometric and symbolic models [11]. We are also interested in how positions vary in time; taking into account that dynamicity leads us to present in the following a review of the literature on the trajectories.

## 2.2 Trajectory definition

A trajectory (or tracklog or path) is defined by the sequence of recorded positions of a moving object during a period of time. A trajectory can be segmented according to several criteria. The segment of a trajectory is defined as a maximum subsequence of the trajectory such that all space-time positions match a given predicate (e.g. constant direction) [12].

[3] gives a formal and generic definition of a trajectory and segment of trajectory. Trajectory (see equation 1) is defined as a tuple composed of a trajectory identifier, an object identifier, a set of trajectory segments and a time interval giving the "start" and "end" time of the trajectory.

$$Tr = (tr_{id}, obj_{id}, \{u_k\}, [t_{start}, t_{end}]) \quad (1)$$

$$u_k = (u_{id}, \{position_i\} / 1 \leq i, [t_{ks1tart}, t_{kend}]) \quad (2)$$

A trajectory segment (see equation 2) is a tuple that contains a segment identifier, a uniform set of positions and a time interval which is optional.

The objects movement can be performed freely (e.g. air moving) [13] or forced by the architecture of a building, for example [9] (e.g. moving inside a subway station is constrained by the arrangement of the different spaces and their connections).

In the following, we present the types of potential spatial queries in an indoor environment.

## 2.3 Spatial queries

A spatial query includes at least one geo- or spatial location element (expressed as geometric coordinates or a symbolic name) and/or a time reference (date, time, interval) [14]. The benefit of spatial queries interpretation on past trajectories of objects or predicted ones has been proved by many research works and software. There are several works about spatial queries like (i) position queries [15], returning the position of a moving or static object (geometric or symbolic position), (ii) region queries [16], [17], [18], [19], e.g. "what are the pharmacies located in the area?" (iii) nearest neighbors' queries [20], e.g. "what are the nearest houses of the intersection?", (iv) navigation or trajectory queries [21], e.g. "what is the shortest path to reach home?".

[22] present a prototype (AQMISS) which returns to the user a set of semi-structured documents about his interests or desired location. To implement it, they allow the user different types of spatial queries.

Most of the works on objects trajectories aim to use the old trajectories to detect regular travel and thus to predict the future position of objects. There are less trajectories-oriented works that consider reconstitution queries and, as far as we can know, no framework of query that manages heterogeneous trajectories (containing both symbolic and geometric segments) in indoor environment. In the following, we will present two use cases that demonstrate the need to cope with such queries.

Let's suppose an "intelligent" working building. Each employee has a smartphone that is legally tracked by positioning systems based on different location sensors. Other types of sensors are monitoring the energy consumption and the temperature everywhere in the building.

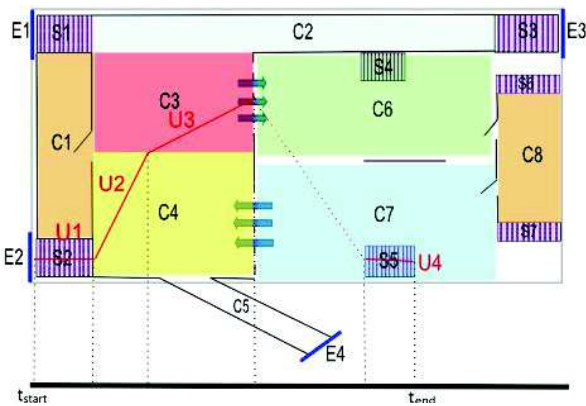


Figure 1. Example of trajectory one

Figure 1 illustrates the trajectory of a person inside the building. The trajectory is composed of four segments (U1, U2, U3 and U4). There might be a lack of position information for a given period of time (a person turned off his smartphone, so no more information comes about his position). The dotted line represents a gap in the definition of the trajectory (called "hole") that can be "filled" using interpolation functions [23]. Based on this trajectory, we want to compute different operators like aggregation of the energy consumption, identifying the sensors situated on the path or at a given distance of the path.

Let's consider the next case, surveillance in a working space; [24] explains the notion of managing the surveillance coverage zone using sensor networks. The classical scenario is triggered when a problem occurs in a given area of the building: someone is moving inside a restricted area. His/her movement track is generated based on the detection of his/her smartphone by the surrounding Wi-Fi cells. This generates coordinates (x, y) with regards to the building map. At a given place, the signal gets lost and the only available information is that he/she entered a secured room (his badge was validated at the entrance of the room). In this case the person's trajectory would look like the one in figure 2. The first two segments of the trajectory are U1 and U2, there is a period where the object gets lost and the last part of the trajectory is U3 which takes place in room C1. In that case we want to identify the video cameras on the given path in order to visualize the recordings.

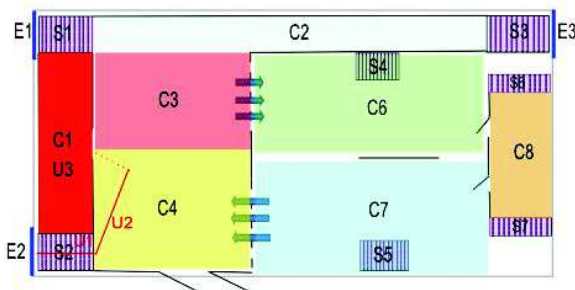


Figure 2. Example of trajectory two

In the following we present the requirements for designing a system that could support the queries identified in the two use cases.

### 3. REQUIREMENTS FOR A HETEROGENEOUS TRAJECTORY BASED QUERY FRAMEWORK

The main problems when designing an object trajectory querying framework are:

- integrating multiple (geometric and symbolic) segments within a trajectory,
- integrating multiple coordinate systems (local coordinate system or global coordinate systems, indoor or outdoor),
- integrating different trajectories coming from different sensors (e.g., the same object being tracked by different sensors, how to combine the two paths).

Based on this model, the requirements for the query framework that we developed followed this strategy:

- specification of the location model: even if the purpose of our work isn't to define yet another indoor location model since lots of efforts have been done in this direction, we will present a state of the art of the different types of existing approaches and we will describe and motivate our choices, by emphasizing the particularities of our model.
- specification of the query language (definition of a query template)
- query transformation to SQL (definition of a query interpreter module)
- data and query management tools (for a single coordinate system, multiple modeling)
- aggregation operators within or after the query
- geocoding and reverse geocoding within a coordinate system

The identified requirements for the underlying location model are:

- geometric and symbolic location: as illustrated by the given use cases, it is important for a location model to support both geometric and symbolic positions, as sensor outputs are heterogeneous and users formulates more semantic location descriptions,
- connectivity relation: the representation of trajectories requires using the connectivity information between locations. For that purpose it is crucial to use the topological properties of the environment (map of the building) because the objects movement is constrained by that topology. That information requires to be explicitly modeled in a location model,
- in order to integrate symbolic and geometrical coordinates, we need to define mappings between the spatial model (the representation of the environment) and the geometric coordinates. For that purpose, the geometry information related to each section (room in our case) of the spatial model needs to be integrated in the data model,
- distance: since many queries imply detecting the objects that are situated in the visibility range of a sensor, it is important for information stored in the location model to enable distance calculus,
- coordinate system transformation: when dealing with positioning technologies that express locations with respect to different coordinate systems, it is important to describe transformations between these reference systems.

#### 4. TEMPLATE OF HYBRID TRAJECTORY BASED QUERY

According to the steps of our strategy, we will start by defining a query template that the system will be able to support based on the two examples of use cases presented in Section 2.3. The innovation that we introduce into the trajectory definition is the fact that each position is expressed relatively to a reference system (e.g., building plan, ICCARD reader's network). Positions are homogeneous within the segment (expressed according to a single reference system) but the individual segments may contain heterogeneous positions as in the examples in Section 2.3.

Hybrid trajectory query is defined as a tuple shown below:

$$T_r = (tr_{id}, \{u_k\}, [t_{start}, t_{end}]) \quad (3)$$

where  $tr_{id}$  is the trajectory identifier,  $\{u_k\}$  is the set of spatial segments constituting the trajectory and  $[t_{start}, t_{end}]$  represents the time interval of the trajectory query.

The definition of a path segment is as follows:

$$u_k = (u_{id}, refS_{id}, \{position_i\}) \quad (4)$$

where  $u_{id}$  is the segment identifier,  $refS_{id}$  is the identifier of the reference system about which the positions are defined and  $position_i$  is the set of positions that make up the trajectory segment.

To manage the heterogeneity of path segments, we define the position of an object as being composed of two parts:

- one identifying the reference system,
- another one referring to the location relative to the reference system.

The idea goes in the same sense as that proposed by [25] which defines the syntax of a hybrid indoor location based on the syntax of a URI (Universal Resource Identifier).

A hybrid trajectory is composed of two parts, a spatial one and a temporal one. The spatial part consists of a sequence of segments, each segment consisting of a reference system identifier and a sequence of positions (geometric or symbolic) expressed relatively to the corresponding reference system. The temporal part is an interval of time. This hybrid trajectory will constitute the entry point of our query framework. Figure 3 shows an example of hybrid trajectory that contains two segments, one composed of geometrical positions expressed with regards to the floor map and one composed of one symbolic position which represents the ID of the sensor (ICCARD reader) which is expressed with regards to the ICCARD reader's network deployed in the building.

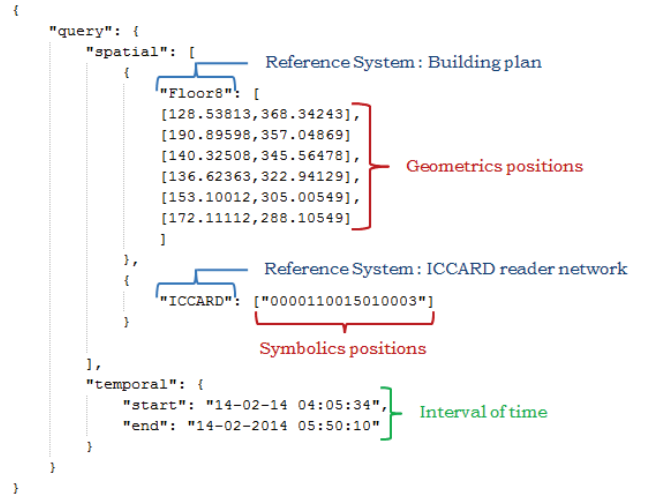


Figure 3. Example of a hybrid trajectory query

This query gives the trajectory possibly followed by the person to identify. The spatio-temporal information generated by the various sensors and saved will be queried by this trajectory to determine the relevant objects (people, devices, etc.), as well as surveillance cameras that could have filmed this path during the time interval of the query.

#### 5. INDOOR COORDINATE SYSTEM

Expressing localization information with regards to different local reference systems is one of the main sources of heterogeneity and lack of interoperability among location models. In outdoor environment, GPS reference system makes things much easier. In general, spatial models have a multi-layer structure with associated mapping functions that can be used to project any local coordinates with regards to the GPS reference system.

In indoor environments, as there is no global reference system, local coordinate systems might be associated to the whole building (for 3D positions), to one floor or to each room. Given the map of the building we are going to consider one coordinate system per floor as in Figure 4.

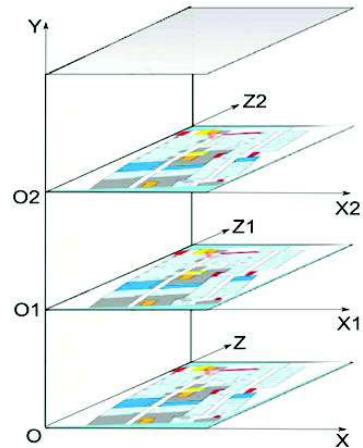


Figure 4. Coordinates system of a building

Positions are expressed using 2D coordinates with regards to each plan of the floor. A third coordinate is associated to each position to indicate the floor number.

In the next section, we present the different algorithms for query results.

## 6. ALGORITHMS

The entry point of our system is a hybrid trajectory query that mixes geometric and symbolic positions which are expressed according to different reference systems and a time interval. Once this query is interpreted by a query interpreter module, it is expressed as a set of points  $(p_1, p_2, \dots, p_n)$  and a time interval  $[t_1, t_2]$ . These points form a sequence of spatial segments  $(u_1, u_2, \dots, u_n)$  for the geometrical positions and circles center  $(c_1, c_2, \dots, c_n)$  for the symbolic positions. This sequence of points and the time stamps are the inputs of the search algorithms. The aim is:

- to propose a set of objects (person, device, etc.) whose positions have "intersected" the trajectory given by the query in the time interval  $[t_1, t_2]$ ,
- to propose a set of objects within the coverage area (range) of each ICCARD reader of the query trajectory in the time interval  $[t_1, t_2]$ ,
- to propose a list of cameras whose field of view (which may be variable) has "intersected" the trajectory of the query in the time interval.

### 6.1 Searching for the objects positions related to wireless terminals

Drawing a (poly-line) trajectory from the query given positions enables to define points. Then it is necessary to expand these points in geometries in order to cope with uncertainty and imprecision or approximation and human subjectivity (e.g. 168 instead of 168.142154). It implies to perform translations for the poly-lines from one end to the other of this trajectory in order to define a geometry that includes it enhancing a larger surface.

Based on this trajectory, we have to find the objects/persons that are "on" (or within) it. To succeed, we will "intersect" the poly-line with Wi-Fi data set (positions of objects/persons) stored in the Oracle database (Wi-Fi data set whose time stamp is in the interval  $[t_1, t_2]$  given by the query); the objects/persons which are on the path constitute our results.

#### Operators:

The result is the set of pairs:  $R = \{r = (obj_i, pos_i)\}$ ; function  $draw\_geometry(u_k)$  helps to draw a geometry using the poly-line formed by segments  $u_k$ ; function  $get\_object([t_1, t_2])$  returns a list of objects whose time (date time) is in the interval  $[t_1, t_2]$  and the corresponding positions.

#### Algorithm:

---

**Algorithm 1:** algorithm for identifying objects whose positions intersected the query path

---

**Inputs:** a sequence of path segments:  $u_k$  and an interval of time:  $[t_1, t_2]$ .

**Output:** A list of objects that have intersected the path and the corresponding positions.

```

1:  $geometry \leftarrow draw\_geometry(u_k)$ 
2:  $list\_object(obj_i, pos_i) \leftarrow get\_object([t_1, t_2])$ 
3: for each  $obj_i$  of  $list\_object(obj_i, pos_i)$  do
4:   if  $pos_i$  intersects  $geometry$  then
5:      $add\_result(obj_i, pos_i)$ 
6:   end if
7: end for

```

---

### 6.2 Searching for objects positions related to ICCARD readers

In this case, we are dealing with symbolic positions which are specific surfaces in the building. Each ICCARD reader has a field of action or a scope. When the trajectory query passed in the query interpreter module, the reader names are represented on the plan of the building by points (algorithm inputs) which are the positions assigned to readers; these points help to draw geometries that are circles, whose diameter is scope of each reader. The rays of the circles are customizable. Search objects are in the surface area of the circles. So we search here objects within the constructed geometries.

#### Operators:

The result for ICCARD readers is all pairs:  $R = \{r = (obj_i, pos_i)\}$ ; function  $draw\_circle(p_n, rayon_i)$  allows to draw circles whose centers are the points  $p_n$  and the rays are  $rayon_i$ ; function  $get\_object([t_1, t_2])$  returns a list of objects within circles and whose time (date time) is in the interval  $[t_1, t_2]$  along with the corresponding positions.

#### Algorithm:

---

**Algorithm 2:** algorithm for identifying objects within geometry

---

**Inputs:** a sequence of points:  $p_n$  and an interval of time:  $[t_1, t_2]$ .

**Outputs:** list of objects within geometry and the corresponding positions.

```

1:  $list\_object(obj_i, pos_i) \leftarrow get\_object([t_1, t_2])$ 
2: for each  $p_n$  of query do
3:    $geometry_i \leftarrow draw\_circle(p_n, rayon_i)$ 
4:   for each  $obj_i$  of  $list\_object(obj_i, pos_i)$  do
5:     if  $obj_i$  within  $geometry_i$  then
6:        $add\_result(obj_i, pos_i)$ 
7:     end if
8:   end for
9: end for

```

---

### 6.3 Camera selection Algorithm

[2] propose an algorithm for selection of fixed cameras. As our work is an extension to indoor environment, we will adapt this algorithm to this context.

The aim is to propose to the operator a list of cameras that may have filmed segments of the trajectory given as query. For each camera we must give the interval of time in which she filmed the trajectory.

The algorithm proceeds in two steps, the filtering step and the refining of results:

- filtering: selecting cameras that are at a distance less than the distance of maximum visibility of the trajectories segments;
- refining: for the selected cameras in the filtering step, we calculate the geometry of the field of view, and select those whose geometries touch the trajectories segments.

The result is the set of triples  $R = \{r = (c_i, u_k, [t_a, t_b])\}$  where  $c_i$  is the selected camera,  $u_k$  is the filmed segment and  $[t_a, t_b]$  the interval of time in which the camera filmed the segment of the trajectory.

These algorithms are implemented in the Searching engine module of the framework developed architecture.

In the next section we present the framework architecture.

## 7. FRAMEWORK ARCHITECTURE AND IMPLEMENTATION

The developed framework implements the model of hybrid trajectory query and algorithms presented in the previous sections. The implementation of the prototype was done using JDK 1.7 with JDBC drivers for connecting to the Oracle database.

### 7.1 Framework architecture

The main modules of this architecture are:

- User Interface: it is formed by the various blocks which allow interaction with user, data and results visualization;
- Query interpreter: it performs the transformation of the hybrid trajectory query submitted by the user by spatio-temporal query (a set of 2D points expressed relative to building plan forming a sequence of segments and/or geometry and an interval of time) ; Figure 5 shows how Query interpreter module manages a given query.

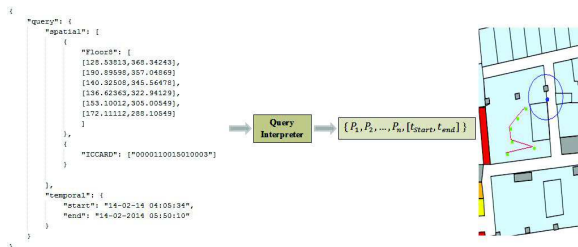


Figure 5. Query interpretation

- Search Engine: it implements the algorithms.
- Storage: it contains the spatial database, the data collection as well as modules that allow interaction with this data;
- SQL Query Builder: it manages communication between the Searching engine and the Storage module;
- Data Collection: it enables collecting of data generated by the different location sensors and organizes them in a given format.

Relations between the different modules are shown in Figure 6.

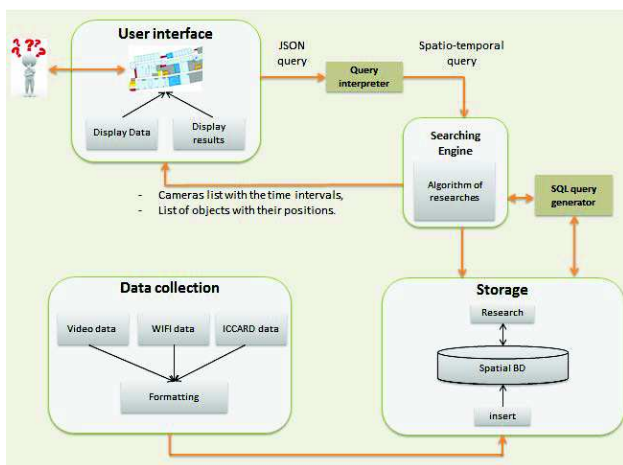


Figure 6. Framework architecture

### 7.2 Implementation

The aim is to implement the search for objects using queries: search for objects on a given trajectory as query, objects located within reach of ICCARD readers, cameras could have filmed the trajectory during the time period given by the query.

We use dataset generated by the sensors (ICCARD, Wi-Fi, cameras) installed on the campus of Kyushu University in Japan for the implementation.

Datasets associated to the different sensors are organized in XML files which the structure shown in Figure 7:

```
<?xml version="1.0" encoding="utf-8"?>
<LocalTracking>
  <SensorInfo>
    <Type> </Type>
    <Name> </Name>
  </SensorInfo>
  <PersonInfo>
    <LocalPersonID> </LocalPersonID>
    <Feature> </Feature>
  </PersonInfo>
  <Tracks>
    <Record>
      <DateTime> </DateTime>
      <Coordinates X=" " Y=" " Z=" "/>
      <Vagueness> </Vagueness>
    </Record>
  </Tracks>
</LocalTracking>
```

Figure 7. Dataset structure files

Each XML file contains:

- Information on the sensor: sensor type and name.
- Information on the localized person: the local identifier and characteristics of the person.
- Tracking information: date and time, the location coordinates. The coordinates may be those of the sensor or those of the person to locate. The sensors provide coordinates in a two-dimensional system (X, Y), the coordinate Z is the number of the floor.

We defined the query "indoorQuery" in a file (Figure 8):



```

1 {
2   "query": {
3     "spatial": [
4       {
5         "Floor8": [
6           [45,425],
7           [92,468]
8           [206,495],
9           [269,481],
10          [278,382],
11          [526,345],
12          [526,221],
13          [546,158],
14          [411,186],
15          [371,175],
16          [160,211],
17          [149,280]
18        ]
19      },
20      {
21        "ICCARD": ["0000110015010005"]
22      }
23    ],
24    "temporal":
25    {
26      "start": "14-02-2014 05:00:00",
27      "end": "14-02-2014 05:50:00"
28    }
29  }
30 }

```

Figure 8. Hybrid trajectory

The interpretation of this query is detailed in Figure 9:

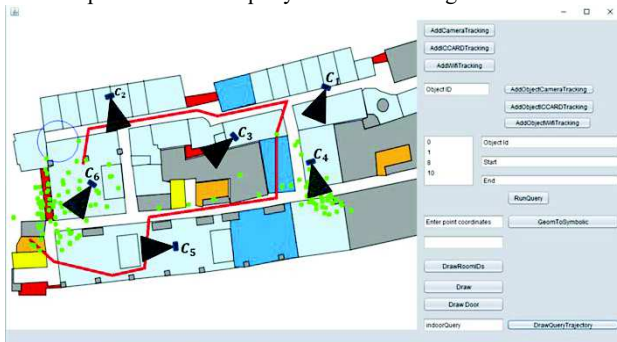


Figure 9. Application

the trajectory (red polyline) represents the first segment of the query which is constituted by the set of geometric positions ([45;425], [92;468], [206;495], [269;481], [278;382], [526;345], [526;221], [546;158], [411;186], [371;175]) ; blue circle represents the scope of the single ICCARD reader ("0000110015010003") present in the query. The green points are the positions of Persons generated by the wireless terminals and stored in the Oracle database.

We focus on people on the red poly-line trajectory (in this case 6) and those located in or on the blue circle (in this case 2). Relevant cameras are here those whose fields of view are "intersected" the trajectory (in this case C2 and C5).

The queries implementation in Oracle Spatial enables to visualize the results by returning the list of people identifiers and their

positions, selecting cameras that have "intersected" segments of the trajectory and the time intervals during which these fields view have "intersected" the trajectory.

## 8. CONCLUSION

Our contribution in the field of indoor location is an extension of previous works on the construction of trajectories. We define an indoor coordinate system which can be associated to the building plan, a floor or to each space. Positions are expressed using 2D coordinates in relation with floor plan; a third coordinate is associated to each position in order to indicate the number of the floor for example. We have proposed a model of hybrid trajectory query which comprised geometric and symbolic positions expressed in relation to different reference systems. The developed algorithms used to find objects (people, devices, etc.) which are on a trajectory given as a query during a specified interval of time, and to look for cameras that could have filmed this trajectory.

Our approach has been tested within the context of the French national projects on video surveillance and forensic METHODEO, funded by the National French research Agency. It has also been implemented in the context of OVALIE platform<sup>1</sup>. This platform implements a modular space (family room, restaurant room, airplane cabin, etc.) equipped with data collection devices (different types of cameras, microphones, static or moving, directional or not, etc.) in order to analyze social interaction from facial recognition, posture, congruence, etc.

Our work can be instantiated in the context of forensic, for example during an investigation [1]. We plan to extend this context in order to be able to deal with real-time queries or predict the trajectories based on statistics realized on stored data. For indoor location systems based on surveillance with cameras, the result of the proposed operator is a list of cameras whose intervals of time may overlap. A perspective therefore relates to the scheduling of results depending on the distance from the cameras relative to the segments of the request for example. In our algorithms, we search topological relations between objects and geometries. The issue of research topological relations between regions of space is topical, especially between the regions of space with uncertainties. In the processing of a request, it will take into account the uncertainty (temporal fuzzy and spatial fuzzy) [26]. The research issue of topological relations between regions of space is topical, especially between the regions of space with uncertainties. During the query processing, the inclusion of the uncertainty is capital (temporal fuzzy and spatial fuzzy) [26]. These contributions gave the basis for designing a framework whose architecture has been developed w.r.t. different contexts, as a tool to support hybrid querying.

## 9. REFERENCES

- [1] Dana Codreanu, André Péninou, Florence Sèdes, Video Spatio-Temporal Filtering Based on Cameras and Target Objects Trajectories - Videosurveillance Forensic Framework. ARES 2015: 611-617.
- [2] Sèdes, F., Sulzer, J., Marraud, D., Mulat, C., and Cepas, B. (2012). Intelligent Video Surveillance Systems, chapter A Posteriori Analysis for Investigative Purposes. Wiley, ch A
- [3] D. Codreanu (2015). Modélisation des métadonnées spatio-temporelles associées aux contenus vidéo et interrogation de ces métadonnées à partir des trajectoires hybrides :

<sup>1</sup> <http://www.certop.cnrs.fr/OVALIE-une-plateforme-pour-l-etude>

Application dans le contexte de la vidéosurveillance. Thèse de doctorat, Université Paul Sabatier, mai 2015.

- [4] Li, K.-J. (2008). Indoor space: A new notion of space. In Proc. 8th International Symposium on Web and Wireless Geographical Information Systems (W2GIS) (2008), pp. 1–3. Loi: 10.1007/978-3-540-89903-71.
- [5] Leonhardt, U. (1998). Supporting Location-Awareness in Open Distributed Systems. PhD thesis, Imperial College of Science, Technology and Medicine University of London.
- [6] Ledoux, H. and Gold, C. M. (2008). Modelling three-dimensional geoscientific fields with the voronoi diagram and its dual. *Int. J. Geogr. Inf. Sci.*, 22(5):547–574.
- [7] Mekni, M. (2010). Automated Generation of Geometrically-Precise and Semantically-Informed Virtual Geographic Environments Populated with Spatially-Reasoning Agents. Universal-Publishers.
- [8] Parent, C., Spaccapietra, S., Renso, C., Andrienko, G., Andrienko, N., Bogorny, V., Damiani, M. L., Gkoulalas-Divanis, A., Macedo, J., Pelekis, N., Theodoridis, Y., and Yan, Z. (2013). Semantic trajectories modeling and analysis. *ACM Comput. Surv.*, 45(4):42:1-42:32.
- [9] Jensen, C. S., Lu, H., and Yang, B. (2009). Graph model based indoor tracking. In *Mobile Data Management: Systems, Services and Middleware, 2009. MDM '09.* pp 122-131.
- [10] Buschka, P. (2005). An Investigation of Hybrid Maps for Mobile Robots. PhD thesis, Dept. of Technology, Orebro University, Sweden.
- [11] C. Becker and F. Durr (2005). On location models for ubiquitous computing. *Personal and Ubiquitous Computing*, vol. 9, pp. 20. [Online]. Available: <http://dx.doi.org/10.1007/s00779-004-0270-2>.
- [12] Mountain, D. and Raper, J. (2001). Modelling human spatio-temporal behaviour: a challenge for location based services. 6th International Conference on GeoComputation, pp. 2-9.
- [13] Parent, C., Spaccapietra, S., and Zimányi, E. (2006). *Conceptual Modeling for Traditional and Spatio-Temporal Applications: The MADS Approach.* Springer-Verlag.
- [14] Ilarri, S., Mena, E., and Illarramendi, A. (2010). Location dependent query processing: Where we are and where we are heading. *ACM Comput. Surv.*, 42(3) :12 :1–12 :73.
- [15] Becker, C. and Durr, F. (2005). On location models for ubiquitous computing. *Personal Ubiquitous Comput.* 9(1):20–31.
- [16] Gu, Y., Yu, G., Guo, N., and Chen, Y. (2009). Probabilistic moving range query over rfid spatio-temporal data streams. *ACM CIKM '09*, pp 1413-1416.
- [17] Arslan Ay, S., Zimmermann, R., and Kim, S. H. (2008). Viewable scene modeling for geospatial video search. 16th ACM Intl Conf on Multimedia, MM '08, pp 309–318.
- [18] Scuturici, V.-M. and Ejigu, D. (2006). Positioning support in pervasive environments. In 2006 ACS/IEEE International Conference on Pervasive Services, pp. 19–26.
- [19] Yuan, W. and Schneider, M. (2010). Supporting continuous range queries in indoor space. 11th Intl Conf on Mobile Data Management (MDM), pp 209–214. IEEE.
- [20] Zhang, J., Zhu, M., Papadias, D., Tao, Y., and Lee, D. L. (2003). Location-based spatial queries. 2003 ACM SIGMOD Intl Conf on Management of Data, SIGMOD'03, pp 443–454.
- [21] Afyouni, I., Ray, C., Ilarri, S., and Claramunt, C. (2012b). Algorithms for continuous location-dependent and context-aware queries in indoor environments. 20th Intl Conf on Advances in Geographic Information Systems, ACM SIGSPATIAL '12, pp 329-338.
- [22] Dana Al Kukhun, Bouhra Soukkarieh, Corinne Zayani, Florence Sèdes. AQMIS: An Adaptive Querying Prototype for Mobile Information Systems. *Communications of SIWN, The Foresight Academy of Technology, UK, Vol. 6, p. 29-35, april 2009.*
- [23] Vazirgiannis, M. and Wolfson, O. (2001). A spatiotemporal model and language for moving objects on road networks. In *Proceedings of the 7th International Symposium on Advances in Spatial and Temporal Databases, SSTD'01*, pages 20–35, London, UK, UK. Springer-Verlag.
- [24] Florence Sèdes, Gaele Calvary, Thierry Delot, Jean-Yves Tigli. *Computer Science and Ambient Intelligence*, Wiley Interscience Publications, ISBN 9781848214378, december 2012, ch3
- [25] Jiang, C. and Steenkiste, P. (2002). A hybrid location model with a computable location identifier for ubiquitous computing. In *Proceedings of the 4th International Conference on Ubiquitous Computing, UbiComp'02*, pages 246–263, London, UK, UK. Springer-Verlag.
- [26] Florence Sèdes, Ahed Alboody, Jordi Inglada (2010). Modeling Topological Relations between Uncertain Spatial Regions in Geo-spatial Databases: Uncertain Intersection and Difference Topological Model. *DBKDA 2010, IEEE Computer Society*, pp. 56-68.