(Meta-)Data Modelling: Gathering Spatio-Temporal Data for Indoor-Outdoor Queries

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

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].

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. The different algorithms are developed in section 5. Section 6 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 track log 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.

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

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

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 benefits of spatial queries interpretation on past trajectories of objects or predicted ones have 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] presents a prototype (AQMIS) which provides the user with 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" office 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.

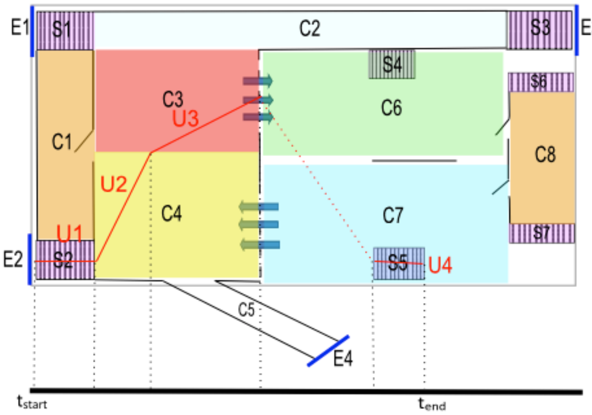


Figure 1: Example of trajectory one

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 that 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.

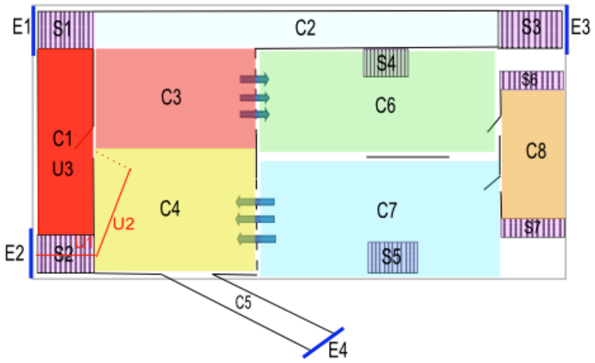


Figure 2: Example of trajectory two

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 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.

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:

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

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

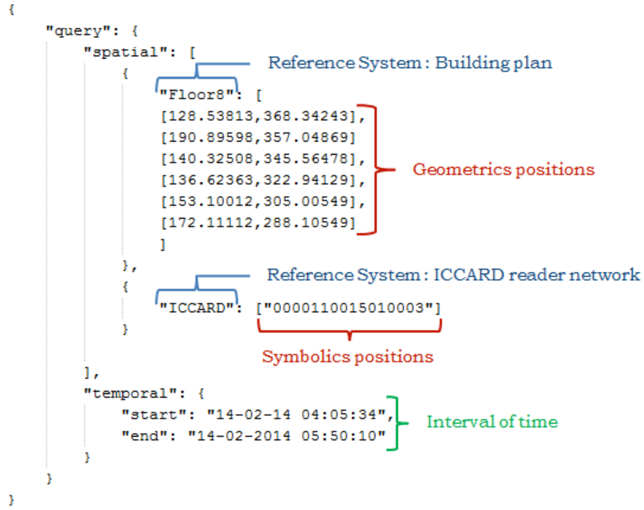
The definition of a path segment is as follows:

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

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.



```
{
    "query": {
        "spatial": [
            {
                "Floor8": [
                [128.53813,368.34243],
                [190.89598,357.04869]
                [140.32508,345.56478],
                [136.62363,322.94129],
                [153.10012,305.00549],
                [172.11112,288.10549]
                ]
            },
            {
                "ICCARD": ["0000110015010003"]
            }
        ],
        "temporal": {
            "start": "14-02-14 04:05:34",
            "end": "14-02-2014 05:50:10"
        }
    }
}
```

Reference System : Building plan
Geometrics positions
Reference System : ICCARD reader network
Symbolics positions
Interval of time

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 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,…, p_n) and a time interval [t_1, t_2]. These points form a sequence of spatial segments (u_1, u_2,…, u_n) for the geometrical positions and circles center (c_1, c_2, …, 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.

5.1 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.

```
geometry ← draw _geometry ($u_k$)
      list_object ($obj_i$, $pos_i$) ← get_object ([$t_1$, $t_2$])
for each $obj_i$ of list_object ($obj_i$, $pos_i$) do
    if $pos_i$ intersects geometry then
            add_result ($obj_i$, $pos_i$)
    end if
end for
```

5.2 Searching for objects positions related to ICCARD readers

In this case, we are dealing with symbolic positions that 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.

```
list_object ($obj_i$, $pos_i$) ← get_object ([$t_1$, $t_2$])
for each $p_n$ of query do
    geometry$_i$ ← draw_circle ($p_n$, $rayon_i$)
    for each $obj_i$ of list_object ($obj_i$, $pos_i$) do
            if $obj_i$ within geometry$_i$ then
                    add_result ($obj_i$, $pos_i$)
            end if
    end for
end for
```

5.3 Camera selection Algorithm

[2] proposes an algorithm for selecting 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 ci 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.

6 Conclusion

Our contribution in the field of indoor location is an extension of previous works on the construction of trajectories and meta-data management [27]. We define an indoor coordinate system, associated to the building plan, floor or any 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 gathering geometric and symbolic positions expressed in relation to different reference systems. The developed algorithms used to find objects (people, devices, etc.) 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 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.

Our approach has been continuously assessed, first within the context of the French national projects on video surveillance and forensic METHODEO, funded by the National French research Agency. Next steps are FILTER2 French ANR project and VICTORIA H2020 European project.

References

[1] D. Codreanu, A. Péninou, F. Sèdes (2015). Video Spatio-Temporal Filtering Based on Cameras and Target Objects Trajectories - Videosurveillance Forensic Framework. *ARES 2015: 611-617.*

[2] F. Sèdes, J. Sulzer, D. Marraud, C. Mulat, and B. Cepas (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 : Application dans le contexte de la vidéosurveillance. *PhD thesis, Université Paul Sabatier, mai 2015.*

[4] K.-J Li (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.*

[5] U. Leonhardt (1998). Supporting Location-Awareness in Open Distributed Systems. *PhD thesis, Imperial College of Science, Technology and Medicine University of London.*

[6] H. Ledoux and C. M. Gold (2008). Modelling three- dimensional geoscientific fields with the voronoi diagram and it's dual. *Int. J. Geogr. Inf. Sci., 22(5):547–574.*

[7] M. Mekni (2010). Automated Generation of Geometrically-Precise and Semantically-Informed Virtual Geographic Environments Populated with Spatially-Reasoning Agents. *Universal-Publishers.*

[8] C. Parent, S. Spaccapietra, C. Renso, G. Andrienko, N. Andrienko, V. Bogorny , M. L. Damiani, A. Gkoulalas-Divanis, J. Macedo, N. Pelekis, Y. Theodoridis, and Z. Yan (2013). Semantic trajectories modeling and analysis. *ACM Comput. Surv., 45(4):42:1-42:32.*

[9] C. S. Jensen, H. Lu, and B. Yang (2009). Graph model based indoor tracking. *In Mobile Data Management: Systems, Services and Middleware, 2009. MDM '09. pp 122-131.*

[10] P. Buschka (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] D. Mountain and J. Raper (2001). Modelling human spatio-temporal behaviour: a challenge for location based services. *6th International Conference on GeoComputation, pp. 2-9.*

[13] C. Parent, S. Spaccapietra, and E. Zim´anyi (2006). Conceptual Modeling for Traditional and Spatio-Temporal Applications: The MADS Approach. *Springer-Verlag.*

[14] S. Ilarri, E. Mena, and A. Illarramendi (2010). Location dependent query processing: Where we are and where we are heading. *ACM Comput. Surv., 42(3) :12 :1–12 :73.*

[15] C. Becker and F. Durr (2005). On location models for ubiquitous computing. *Personal Ubiquitous Comput. 9(1):20–31.*

[16] Y. Gu, G. Yu, N. Guo, and Y. Chen (2009). Probabilistic moving range query over rfid spatio-temporal data streams. *ACM CIKM '09, pp 1413-1416.*

[17] S. Arslan Ay, R. Zimmermann, and S. H. Kim (2008). Viewable scene modeling for geospatial video search. *16th ACM Intl Conf on Multimedia, MM '08, pp 309–318.*

[18] V.-M. Scuturici and D. Ejigu (2006). Positioning support in pervasive environments. *In 2006 ACS/IEEE International Conference on Pervasive Services, pp. 19–26.*

[19] W. Yuan and M. Schneider (2010). Supporting continuous range queries in indoor space. *11th Intl Conf on Mobile Data Management (MDM), pp 209–214. IEEE.*

[20] J. Zhang, M. Zhu, D. Papadias, Y. Tao, and D. L. Lee (2003). Location-based spatial queries. *2003 ACM SIGMOD Intl Conf on Management of Data, SIGMOD'03, pp 443–454.*

[21] I. Afyouni, C. Ray, S. Ilarri, and C. Claramunt (2012). 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] D. Al Kukhun, B. Soukkarieh, C. Zayani, F. Sèdes (2009). 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] M. Vazirgiannis and O. Wolfson (2001). A spatiotemporalmodel and language for moving objects on road networks. *In Proceedingsof the 7th International Symposium on Advances in Spatial and Temporal Databases, SSTD'01, pp 20–35, London, UK, Springer-Verlag.*

[24] F. Sèdes, G. Calvary, T. Delot, J.-Y. Tigli (2012). Computer Science and Ambient Intelligence, *Wiley Interscience Publications, ISBN 9781848214378, december 2012, ch3*

[25] C. Jiang and P. Steenkiste (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. Springer-Verlag.*

[26] F. Sèdes, A. Alboody, J. Inglada (2009). Enriching The Spatial Reasoning System RCC8. ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS 2008), Irvine, CA, USA, Vol. 1, ACM DL, The SIGSPATIAL Special Number 1, p. 14-20, march 2009.

[27] F. Sèdes, S. Laborie, A.-M. Manzat (2009). Managing and querying efficiently distributed semantic multimedia metadata collections. *IEEE MultiMedia, Vol. 16 N. 4, p. 12-20, dec. 2009.*