L I F C

LABORATOIRE D'INFORMATIQUE DE L'UNIVERSITE DE FRANCHE-COMTE

EA 4269

Wi-Fi-Based Indoor Positioning: Basic Techniques, Hybrid Algorithms and Open Software Platform

Matteo Cypriani* — Frédéric Lassabe** — Philippe Canalda* — François Spies*
*Université de Franche-Comté, LIFC, Montbéliard, France.
Email: firstname.name@univ-fcomte.fr
**Université de Technologie de Belfort-Montbéliard, SeT, Belfort, France.
Email: frederic.lassabe@utbm.fr

Rapport de Recherche no RR 2011–06

THÈME 4 – September 2010

L I F C

Wi-Fi-Based Indoor Positioning: Basic Techniques, Hybrid Algorithms and Open Software Platform

Matteo Cypriani*, Frédéric Lassabe**, Philippe Canalda*, François Spies*
*Université de Franche-Comté, LIFC, Montbéliard, France.
Email: firstname.name@univ-fcomte.fr
**Université de Technologie de Belfort-Montbéliard, SeT, Belfort, France.
Email: frederic.lassabe@utbm.fr

Thème 4
OMNI

September 2010

Abstract: In urbanized and indoor environments, outdoor positioning systems, such as *Global Navigation Satellite Systems* (GNSSs), are often inaccurate and adaptations of such systems to those contexts are expensive and hard to deploy. Nowadays, a lot of indoor positioning techniques have been studied, but it it quite difficult to objectively evaluate and compare their accuracies in the same environment.

Our *Open Wireless Positioning System* (OWLPS) allows a comparison of indoor positioning algorithms and techniques with the same input data. Those techniques include propagation models, signal strength map, building topology, description of access points… Few algorithms use a Viterbi-like algorithm to take into account the pathway of the mobile terminal.

This paper presents a new hybrid algorithm combining a signal strength cartography and a calibrated propagation model. Finally, we compare our solution with well known algorithms in order to evaluate the results in a real context.

Key-words: Indoor positioning, Wi-Fi, IEEE 802.11, Positioning algorithm, Viterbi, Topology, Signal strength cartography, Fingerprinting location, Multilateration, FBCM, FRBHM, Friis-based building propagation model

Laboratoire d'Informatique de l'Université de Franche-Comté,
Antenne de Montbéliard — UFR STGI,
Pôle universitaire du Pays de Montbéliard,
25200 Montbéliard Cedex (France)
Téléphone : +33 (0)3 81 99 47 43 — Télécopie +33 (0)3 81 99 47 91

Positionnement Wi-Fi en intérieur : techniques de base, algorithmes hybrides et plate-forme logicielle ouverte

Résumé : En environnement urbain, et à l'intérieur des bâtiments, les systèmes de positionnement conçus pour fonctionner à l'extérieur, comme les systèmes satellitaires (GNSS, pour *Global Navigation Satellite System*), sont souvent très imprécis, et l'adaptation de ces systèmes à des environnements confinés est coûteuse et difficile à déployer. De nombreuses techniques de géopositionnement en intérieur ont été étudiées, mais il est ardu de les évaluer et de comparer objectivement leurs précisions dans le même environnement.

Notre plate-forme *Open Wireless Positioning System* (OWLPS), permet une comparaison d'algorithmes et de techniques de positionnement en intérieur, à partir des mêmes données d'entrée. Ces techniques peuvent utiliser des modèles de propagation, une cartographie des puissances du signal, la topologie de la zone de déploiement, la description des points d'accès… Quelques algorithmes tiennent compte des positions successives du terminal mobile grâce à un algorithme à la Viterbi.

Cet article présente un nouvel algorithme hybride, qui combine une cartographie des puissances du signal et un modèle de propagation calibré. Nous comparons également nos solutions et d'autres algorithmes de positionnement issus de la littérature, et évaluons les résultats dans un contexte réel.

Mots-clés : Géopositionnement en intérieur, Géolocalisation, Wi-Fi, IEEE 802.11, Algorithme de positionnement, Viterbi, Topologie, Cartographie des puissances du signal, Empreinte de l'environnement radio, Multilatération, FBCM, FRBHM, Modèle de propagation en intérieur

Laboratoire d'Informatique de l'Université de Franche-Comté,
Antenne de Montbéliard — UFR STGI,
Pôle universitaire du Pays de Montbéliard,
25200 Montbéliard Cedex (France)
Téléphone : +33 (0)3 81 99 47 43 — Télécopie +33 (0)3 81 99 47 91

I. Introduction

The democratization of 802.11 networks, combined with new mobile devices and services, emphasizes the interest of service continuity. Thanks to the availability of positioning services, the need for contextual knowledge is growing.

Delivering an accurate position in a heterogeneous environment is still a hard challenge. Moreover, experiments show that propagation results vary according to the activity inside the building. New generation positioning algorithms will be able to adapt dynamically themselves to the outside condition. Some steps need to be taken in order to progress towards this new generation, such as heterogeneous and variation detections.

After our state of the art, considering characteristics of indoor and outdoor heterogeneous environment, we introduce a set of new contributions from a topological model, a history memorization algorithm derived from Viterbi and its implementation in positioning algorithms from the literature. We then introduce our system OWLPS (Open WireLess Positioning System) with the description of basic components, positioning algorithms and hardware platform. We also propose a new design platform (OWLPS-1.0) addressing the dynamic changes in the environment and composing new algorithms to reduce the calibration and cartography cost as well as to minimize the distortion of signal strength dynamic variations in modern buildings.

II. State of the art

While outdoor positioning is widely treated and achieved by GNSS, indoor positioning is currently under development. Wi-Fi indoor positioning can be divided into two main categories. One category is based on wave propagation and relies on computing distances between mobile devices and points whose coordinates are known. The second family is based on mapping by combination of signal strength measurements and geographical coordinates, called a signal strength (SS) map. Locating a mobile device with a SS map consists in matching a measurement with some point of the SS map. Measurements matching is either deterministic [1] or probabilistic [2].

A. Propagation models

In propagation-based approaches, the main problem is to compute distances between transmitters and receivers based on signal strength measurements. Distance computation requires radio wave propagation modeling to express distance according to signal strength value. We identified two main choices. One is based on polynomial regression. The other one is based on the Friis formula.

Polynomial regression tries to match several signal strength measurements linked to transmitter-receiver distance with a polynomial expression. In SNAP-WPS [3], the authors conclude that third degree polynomial regression works best.

The other approach is based on the Friis formula [4]. The Friis formula gives signal strength according to wave parameters and distance between transmitter and receiver:

$$\frac{P_R}{P_T} = G_R G_T \left(\frac{\lambda}{4\pi d}\right)^2$$

where:

- P_R and P_T are power received and power transmitted,
- G_R and G_T and receiver and transmitter antenna gains,
- λ is signal wavelength,
- d is the distance between transmitter and receiver.

Although the Friis formula applies on free space propagation (i.e. in earth atmosphere), several contributions intend to adapt it to indoor propagation modeling. All are based on a change of the square power applied to distance. The new value is called the “Friis index”. As indoor environments absorb more signal strength through obstacles, using a Friis index greater than 2.0 allows for greater absorption.

After studying wave propagation in various buildings, Interlink Networks propose a Friis index equal to 3.5 [5]. However, it is suited only for some buildings, as we observed when using the same formula. So, we proposed a Friis-Based Calibrated Model (FBCM) [6] which requires calibration data, such as polynomial regression, to compute Friis indices. FBCM considers access points (AP) individually, so each AP has its own Friis index.

Knowing distances towards surrounding APs, a mobile device can compute its location through multilateration. Multilateration should take into account the unavoidable bias in distance calculation. Xinrong Li [7] proposes to merge distance estimation and location computation by relying on least square estimation, similar to GPS method. Simulation shows good results and same method is tested empirically in [8]. Empirical test shows 3.97 meters mean error and 1.18 meters standard deviation.

B. Signal strength maps

Signal strength map systems are based on mapping by combination of geographical coordinates and signal strength values. Geographical coordinates contain at least basic coordinates, for example cartesian (x, y, z) coordinates, and may be extended with other useful data, such as terminal orientation [1].

Two main steps are identified in signal strength map-based systems: an offline training step builds a signal strength map. Then, the online positioning step relies on the signal strength map previously built. For both steps, two approaches exist. The offline step is performed either by measurements or by simulation. The online step consists in matching a signal strength measurement to the signal strength map content. Matching can be either deterministic or probabilistic.

Building a signal strength map by measurements implies moving physically to every location in the map and perform a measurement [1]. Whereas this method is simple to understand and use, and gives real measurements, it requires a lot of time. On the other hand, building the signal strength map by simulation requires a lot of work to build a propagation model [9] used to compute the signal strength map. One would think that simulating signal propagation to build a signal strength map will rely on propagation-based positioning systems techniques. However, there is a key difference between both systems: a propagation-based positioning system doesn't know the mobile's location, therefore it cannot take into account the obstacles between the mobile and the transmitters. On the opposite, the signal strength map associates signal strength values to known geographical coordinates. It is able to take into account the obstacles with models like that of Motley-Keenan [10].

When a signal strength map is available, mobile positioning is achieved by matching the map content with a signal strength measurement provided by the mobile or the wireless network architecture. Such matching is either deterministic or probabilistic.

Deterministic matching uses a simple signal strength map in which each location has a list of access points within range and an average value of signal strength for each AP. Matching might be either on one point [11] or on several points, whose coordinates are averaged [12].

Probabilistic matching requires more data in the signal strength map. Signal strength values must be described by a probability distribution. Then, matching is achieved by probabilistic methods based on Gaussian models like the CMTA [13] or the kernel method [2]. Other systems model signal strength distribution by histograms [14].

One property of indoor radio wave propagation is its inconsistency. This implies that two signal strength measurements very close in signal strength space can describe two geographical points very far and the opposite. Bahl et al. [9] propose to enhance positioning by considering the past locations to eliminate ambiguous locations. They rely on a Viterbi-like [15] algorithm.

C. Analysis

Evaluated in the same testbed, propagation-based models have poor accuracy, about 8 meters to 15 meters average positioning error. Such accuracy is unable to provide context-aware services based on users location. On the opposite, signal strength map systems offer an accurate positioning but require a lot of work during their setup. Best systems manage to reach about 4 meters mean error but they still assume equal distance errors for all APs in one positioning computation.

These properties are extended to dynamicity: calibrated propagation-based systems and signal strength map-based systems require to perform again the offline step if the environment changes (typically, one AP location is changed). However, there is far less work load for calibrated models than for signal strength map-based ones.

Considering the shortcomings of both approaches, we propose to merge both of them in a hybrid model in order to get their benefits without their drawbacks.

III. Input data

To build a hybrid, topology-aware, indoor positioning system, several base data are required:

- A minimal SS map, with at least one point in each room. It allows a first, coarse, positioning of mobile device.
- A propagation model, for example FBCM. This model is calibrated and used locally, after coarse positioning based on SS map.
- A topology model, either discrete or continuous. Such a model aims at refining the positioning process with device tracking. It eliminates ambiguous locations based on past movements. A Viterbi-like algorithm performs the elimination of all candidate points but one. It requires storing several candidate locations for each positioning iteration.

Of course, we also need a radio signal. We chose to measure on the infrastructure side: the mobile send a positioning request, and the infrastructure elements measure the signal strength. In the sequel, we use "measurements" to designate a list of SS corresponding to a positioning request sent by a mobile and captured by the infrastructure.

IV. Base algorithms

Several base algorithms exist, from which complex techniques are developed.

A. Friis-Based Calibrated Model

The *Friis-Based Calibrated Model* (FBCM) is a multilateration-based positioning algorithm that was presented in earlier publications [6], [16], [17]. It consists in calibrating a propagation model with a priori measurements. Calibration determines which weight to give to transmitter-receiver distance in a Friis-like formula. It aims at computing accurately distances between mobile devices and access points.

B. Multilateration algorithm

The multilateration algorithm used by the current implementation of FBCM is a simple "brute-force-based" approximation method to find the closest point to the distance circles (or spheres, in a 3D space) of the APs. It is called "MinMax". The idea behind this method is to assume that a point that is close to all the distance circles is close to the probable intersection of the circles − if they had only one intersection point.

Pseudo-code for the 2D version is detailed in Fig. 1. The algorithm parses the sub-plan delimited by the points min and max, with a given step. The list of APs and the estimated distances between each AP and the mobile are given by the parameters ap_list and $distance\_list$. At each step, the distance between the current point and the perimeter of the distance circle of each AP is computed; the biggest distance (that is, the farthest circle) is memorized. The final position will be the point with the least memorized distance.

C. Extracting k nearest points from the SS map

Several of our algorithms need to search for k nearest points in a SS map, given a measurement. In the algorithms presented in the next section, we call *k_nss()* the corresponding function, which takes as parameters a SS map and an integer k.

Input: Array of APs: ap_list
Input: Array of Floats: $distance\_list$
Input: Point: min
Input: Point: max
Input: Float: $step$
Integer: $nb\_ap \leftarrow ap\_list$.size()
Float: $d\_min \leftarrow +\infty$
Float: $xm \leftarrow x \leftarrow min_x$
Float: $ym \leftarrow y \leftarrow min_y$
while $x \leq max_x$ **do**
  $y \leftarrow min_y$
  while $y \leq max_y$ **do**
    Float: $d\_max \leftarrow 0$
    for $i \leftarrow 0$ **to** nb_ap **do**
      if distance(x, y, $ap\_list[i]$, $distance\_list[i]$) $\geq$ d_max **then**
        $d\_max \leftarrow$ distance(x, y, $ap\_list[i]$, $distance\_list[i]$)
      end if
    end for
    if $d\_max \leq d\_min$ **then**
      $d\_min \leftarrow d\_max$
      $xm \leftarrow x$
      $ym \leftarrow y$
    end if
    $y \leftarrow y + step$
  end while
  $x \leftarrow x + step$
end while
return xm, ym

Fig. 1. Trilateration algorithm (MinMax), selects the nearest point to the distance circles

V. Contributions

From these base algorithms, we derive refinement techniques that combine a SS cartography, as in [1], and multilateration using FBCM. We called these techniques *FBCM and Reference-Based Hybrid Model* (FRBHM).

Furthermore, we can take into account the building topology. Describing precisely the room layout allows us to estimate the real distance between two points, instead of using a simple euclidean distance. Two variants of the FRBHM use the topology, combined with a Viterbi-like algorithm that uses the past positions of the mobile to compute the current position. Our implementation of this Viterbi-like algorithm is an optimization called *Fast Viterbi-Like* (FVL). When it takes into account past positions of the mobile, it is called *Fast Viterbi-Like Improved* (FVLI).

In the next sections, we present the three FRBHM variants. These algorithms are called when a localization request is received by the system.

Input: Array of SS data: $measurement$
Integer: $nb\_ap \leftarrow measurement$.size()
Array of nb_ap Floats: idx_list
Point: $p \leftarrow$ k_nss($measurement$, 1)
for $i \leftarrow 1$ **to** nb_ap **do**
  $idx\_list[i] \leftarrow$ friis_idx(p, $measurement[i]$)
end for
return FBCM(idx_list, $measurement$)

Fig. 2. Basic FRBHM algorithm

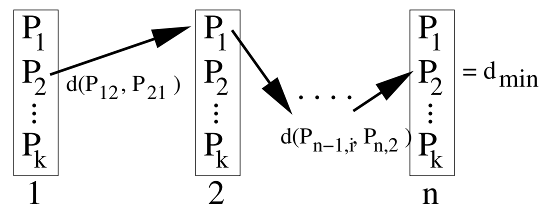

Fig. 3. Viterbi-like algortihm principle

A. Basic FRBHM

The Basic FRBHM is a simple fusion between the SS cartography technique and the propagation model with multilateration. It proceeds in two steps:

1) It selects, in the SS map, the closest point.
2) It calibrates the FBCM with this point's data, and applies it to current measurements to compute the position.

The algorithm is presented in Fig. 2. *nb_ap* is the number of access points for which SS data is present. The *k_nss()* function returns the nearest reference point in SS to the measurement (the parameter 1 indicates that only one reference point is requested). This nearest reference point is called p. The Friis index at p of each access point is computed by the *friis_idx()* function and stored into *idx_list*. The final position is computed by multilateration using FBCM.

B. Viterbi-like algorithms

Viterbi-like algorithm's principle is to avoid ambiguous locations and jumping between distant locations. Two parameters are used: n as history depth and k as history size. History depth defines how many positioning requests results are used. History size defines how many locations a positioning request result is composed. A positioning request result is further defined as a "history set".

Given these parameters, the Viterbi-like algorithm will elect, in the last history set, the point allowing for the shortest path from the first to the last history set. This path goes through exactly one location of each history set according to their order.

The Viterbi-like algorithm has a high complexity: k^n. We aim at improving this algorithm to reduce its complexity. To do so, we need to maintain two history sets: the last one and penultimate one. We also maintain a distance vector $D = \{d_i\}$ of size k (same size as the history sets). Each element d_i in

Input: Integer: n, k
Input: Array of k Floats: v_1, v_2, D
Array of k Floats: new_D
for $i \leftarrow 1$ **to** k **do**
  $new\_D_i = \min(D_j + v_{1_j} v_{2_i}), j \in [1, k]$
end for
return new_D

Fig. 4. Viterbi-like algorithm

it contains the cumulative min. distance from the first history set to element i of the penultimate history set.

On each iteration, i.e. new positioning request result, the distance vector is updated and both history sets are rotated: the penultimate one is set to last one's content and the last one is updated to receive the positioning request result. When the last history set equals the nth history set, the shortest path is selected in the distance vector and the matching point from the last history set is returned as the true mobile location. Fig. 4 shows the algorithm that updates cumulative distances.

C. Discrete FRBHM

The Discrete FRBHM is a bit more evolved than the Basic variant. Its main interest is to take into account the past positions of the mobile, and the building topology, thanks to the FVLI algorithm. When a localization request is received, it is processed as follows:

1) The algorithm searches the reference point database for the k nearest points to the measurement.
2) The FVLI algorithm selects a point from these k points, using the previous k-point groups.
3) FBCM is calibrated with data from the selected point, and applied to compute the position.

The algorithm is presented in Fig. 5. It operates in a way similar to the Basic FRBHM's. The *k_nss()* function now takes k as second parameter, and therefore selects the k nearest reference points in SS. The *queue()* function adds those k points to the Viterbi history. The *FVLI* algorithm is then run on the n series of k points from the Viterbi history, and selects a point p. This point p is used exactly as in the Basic FRBHM: Friis indexes are computed and FBCM is called to obtain the final solution.

D. Continuous FRBHM

In the Discrete FRBHM, we first select a bunch of points from the cartography; therefore, these points belong to a discrete space (the set of reference points) – at this step we cannot have random points. We start using continuous coordinates only at the last step, when the FBCM is called. This can be problematic because with few reference points (which is normally the case with FRBHM), the first selected point is selected by *k_nss()* during several iterations, and therefore selected by FVLI until it disappears from the k-point set. If k is so big that all reference points are selected at each iteration, the solution will never change.

Input: Array of SS data: $measurement$
Input: Array of $k \times n$ Points: $fvli\_history$
Integer: $nb\_ap \leftarrow measurement$.size()
Array of nb_ap Floats: idx_list
Array of k Points: $k\_pts\_list \leftarrow$ k_nss($measurement$, k)
queue($fvli\_history$, k_pts_list)
Point: $p \leftarrow$ FVLI($fvli\_history$)
for $i \leftarrow 1$ **to** nb_ap **do**
  $idx\_list[i] \leftarrow$ friis_idx(p, $measurement[i]$)
end for
return FBCM(idx_list, $measurement$)

Fig. 5. Discrete FRBHM algorithm

Input: Array of SS data: $measurement$
Input: Array of $k \times n$ Points: $fvli\_history$
Integer: $nb\_ap \leftarrow measurement$.size()
Array of nb_ap Floats: idx_list
Array of k Points: $k\_pts\_list \leftarrow$ k_nss($measurement$, k)
$k\_pts\_list \leftarrow$ k_nss($measurement$, k)
for $i \leftarrow 1$ **to** k **do**
  for $j \leftarrow 1$ **to** nb_ap **do**
    $idx\_list[j] \leftarrow$ friis_idx($k\_pts\_list[i]$, $measurement[j]$)
  end for
  $k\_pts\_list[i] \leftarrow$ FBCM(idx_list, $measurement$)
end for
queue($fvli\_history$, k_pts_list)
return FVLI($fvli\_history$)

Fig. 6. Continuous FRBHM algorithm

The aim of the Continuous FRBHM is to work earlier with continuous coordinates, so that the variability of the distances in the Viterbi history is maximized, allowing FVLI to eliminate earlier wrong points.

It operates in three steps:

1) As the Discrete variant, it selects the k nearest reference points to the measurement.
2) For each selected point, Friis indexes are computed and FBCM is called to compute new coordinates.
3) FVLI selects a point amongst all the points computed by FBCM.

The algorithm is presented in Fig. 6. The difference with the Discrete FRBHM is that the two last steps are inverted, so that the FBCM is first executed on each of the k nearest reference points, and FVLI selects one of them last.

VI. OWLPS

All these base and new algorithms are implemented in the *Open Wireless Positioning System* (OWLPS). It is an experimental system, in which we can implement new positioning techniques, then deploy the system in a building and conduct

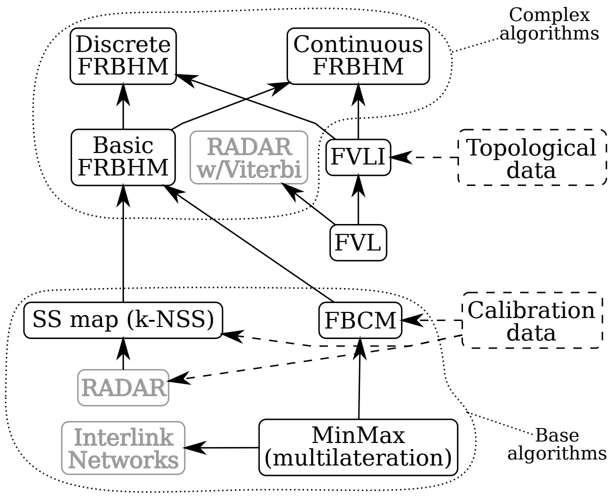

Fig. 8. Algorithms implemented in OWLPS Positioning. Boxes in gray represent algorithms from the literature. Dashed boxes represent data.

experiments. The software architecture makes easy to compare results of several techniques with the same input data.

A previous article [18] presents the system architecture and experiments we conducted that compare our algorithms to some others from the literature. Fig. 7 synthetically presents the system hardware and software architecture. Each hardware element of the system runs a software module: mobile terminals run the *owlps-client* module, access points run *owlps-listener*, the aggregation server runs *owlps-aggregator* and the positioning server run *owlps-positioning*. Note that the two server modules can be installed on the same computer, or even on one of the APs if powerful enough. On a large installation, several aggregation servers could coexist. Mobile terminals can be any kind of Wi-Fi enabled devices: laptop, PDA, cell phone, handheld game console, etc. APs need to have a IEEE 802.11 interface supporting *radiotap* headers; we tested our system with mini-PCs equipped with *Intel BG2200* cards, and with Foneras 2.0g (small APs equipped with an *Atheros* chip).

All software modules run on GNU/Linux systems and are developped in C, except *owlps-positioning* for which we use C++. Source code is versionned with the Git version control system.

Fig. 8 shows a schematic view of the implemented algorithms. Besides our own algorithms, we implemented three techniques from the literature: RADAR, in its two version (without [1] and with [9] Viterbi-like algorithm), and the Friis-like formula of Interlink Networks [5].

VII. Experiments

OWLPS has been deployed in our building and experiments have been conducted. The building is a dense office environment, with thick concrete interior walls and floor separations, glass exterior walls and metallic window shades. The experiment area is about 30m long and 10m large, and includes the two first floors of the building. We used five APs, two at the first floor and three at the second floor, one of them

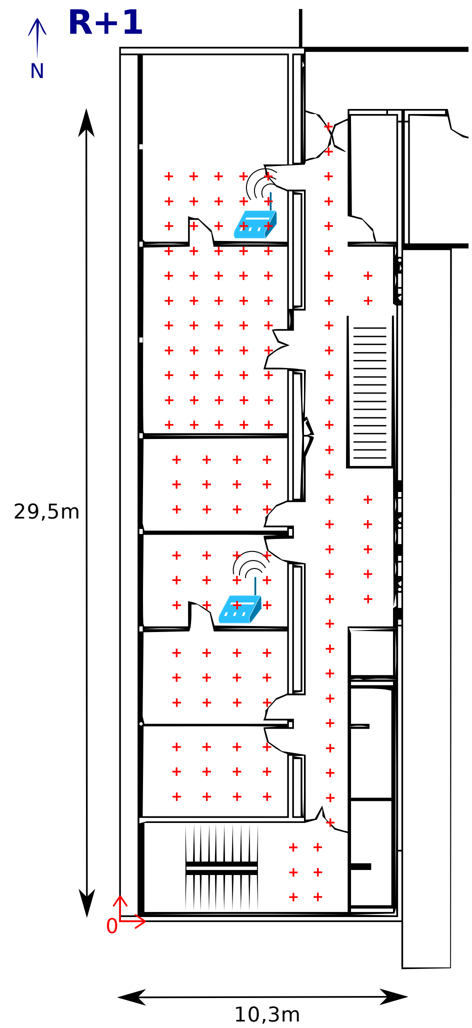

Fig. 10. Map of the second floor of the experiment area. Red crosses represent calibration points.

being out of the experiment area. Fig. 9 shows the building map and the location of the APs. Fig. 10 shows the location of the calibration points at the second floor of the experiment area.

During the calibration phase, we created a database containing 308 calibration points, with measurements in four directions (north, south, east, west) for each of these points. The meshing size of this set of points is 1 meter. We then extracted points from this database in order to create larger meshings: 2 meters (113 points), 3 meters (62 points) and 4 meters (35 points, which is about one per room).

Table I presents the results obtained with a reference path of 86 points taken with a mobile terminal in the two floors of our experiment area. The last line shows the best average error of each algorithm, specifying which meshing gave the best result amongst 1, 2, 3 and 4 meters.

Fig. 11 shows the path we took during our experiment, and the best solution provided by the system, all algorithms taken together. Fig. 12 displays tridimensional curves representing

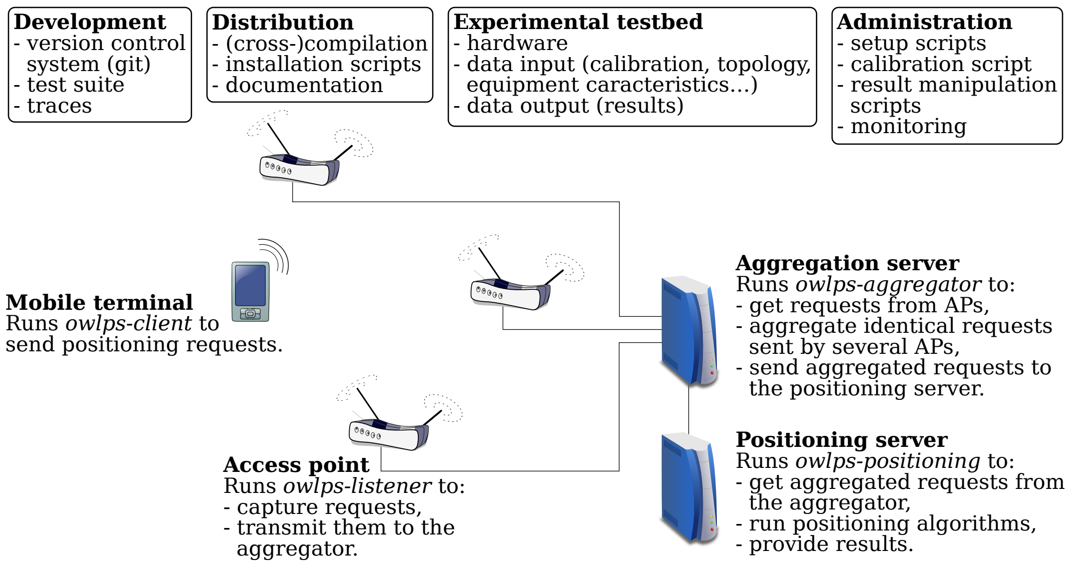

Fig. 7. OWLPS hardware and software architecture

TABLE I
EXPERIMENTAL RESULTS: AVERAGE ERROR ("AVG.") AND STANDARD DEVIATION ("STD. D.").

Meshing	Interlink Networks[a]		RADAR		FBCM		Basic FRBHM		RADAR + VL		Discrete FRBHM		Continuous FRBHM	
	Avg.	Std. d.	Avg.	Std. d.	Avg.	Std. d.	Avg.	Std. d.	Avg.	Std. d.	Avg.	Std. d.	Avg.	Std. d.
1m (308pts)	11.63	5.3	4.74	3.24	10.75	5.69	4.95	2.72	4.85	2.55	5.09	2.57	5.13	2.7
2m (113pts)	11.63	5.3	4.48	3.2	10.1	5.13	4.79	2.6	4.52	2.52	5.03	2.38	5.01	2.74
3m (62pts)	11.63	5.3	5.26	3.35	13.56	6.02	5.09	2.77	5.3	3.13	5.25	3.31	5.25	3.28
4m (35pts)	11.63	5.3	5.03	3.31	7	3.36	5.94	2.3	4.77	2.92	5.78	2.29	6.07	2.53
Best avg. (meshing)	11.63		4.48 (2m)		7 (4m)		4.79 (2m)		4.52 (2m)		5.03 (2m)		5.01 (2m)	

[a]Since it is based on a pure propagation model, the accuracy of this algorithm is not dependent on the meshing.

the error of this best computed position: the higher is the curve, the worse is the computed solution.

We can notice that the meshing does not seem to have a big impact on the accuracy of the algorithms, at least between 1 and 4 meters. Surprisingly the meshing that gives the better results is often 2 meters and not 1 meter, just as bigger meshings (3 and 4 meters) do not give notably worse results. For these meshings, the pure fingerprinting algorithm RADAR gives the best results most of the time. However, a more in-depth test has showed that hybrid algorithms (FRBHM variants) give more accurate results than RADAR variants with a very large meshing. We have therefore to determine what is the best compromise between the meshing size (of which depends the deployment duration) and the accuracy of the system.

Experiments in our building do not show significant accuracy improvement when using the topological data (Discrete and Continuous FRBHM, RADAR with Viterbi-like). With a one-meter meshing, the three Viterbi-enabled algorithms all together provide only 11% of the best solutions, whereas RADAR alone provides the best solution in 45% of the cases. It would be interesting to evaluate the contribution of Viterbi-like algorithms with other types of buildings.

We would like to estimate the error of each algorithm with criteria more complex than the simple euclidean distance. For instance, it is important to know if the mobile is located in a wrong room: a little euclidean distance error may be enough to imply a room change, whereas a bigger error having no impact on which room the mobile is located in is generally preferred.

VIII. NEXT ISSUES & CONCLUSION

OWLPS is an experimental indoor positioning platform based on the Wi-Fi, in which we implement algorithms from the literature and compare them to those from our research by testing in a real environment. We proposed and implemented algorithms, basic techniques and specifications that use the SS cartography of the positioning area, propagation models, and position history techniques (Viterbi-like algorithms) that use the building topology when known.

Our field experiments validate the use of Wi-Fi-based positioning in indoor environments, with low-cost equipments and a three-dimensional accuracy.

The next version of OWLPS should include an enhanced iterative multilateration algorithm, resulting in a new Iterative FRBHM variant. We are currently implementing an auto-calibration subsystem, in order to better adapt Friis indexes to

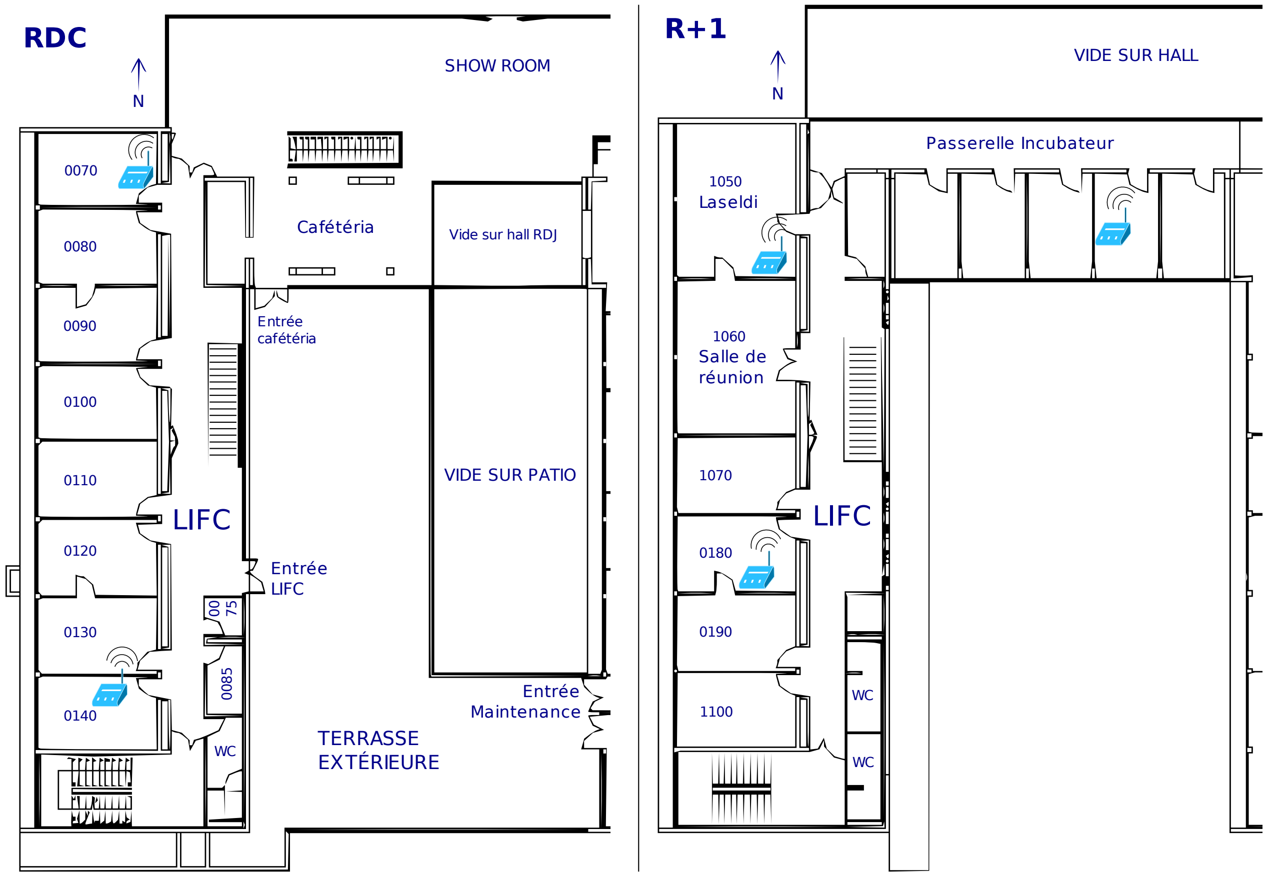

Fig. 9. Map of the building where the experiments take place. On the left is the first floor, where two APs are deployed, on the right the second floor, where three APs are deployed. The experiment area is only the west wing, not the north one.

the fluctuations of the radio environment and handle technical troubles (e.g. breakdown of an AP). We are also considering the implementation of the ability for the system to self-deploy; this would also allow us to add, move or remove APs and the system to automatically change its configuration to keep a good accuracy.

We will also extend our experiments to new contexts and conditions, such as combination with other positioning services [19] at the neighborhood of buildings.

References

[1] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *INFOCOM (2)*, 2000, pp. 775–784.

[2] R. Roos, P. Myllymäki, H. Tirri, P. Misikangas, and J. Sievänen, "A Probabilistic Approach to WLAN User Location Estimation," *International Journal of Wireless Information Networks*, vol. 9, no. 3, pp. 155–164, Jul. 2002.

[3] Y. Wang, X. Jia, and H. Lee, "An indoors wireless positioning system based on wireless local area network infrastructure," in *6th Int. Symp. on Satellite Navigation Technology Including Mobile Positioning & Location Services*, no. 54, Melbourne, Jul. 2003.

[4] L. Blake, *Radar Range-Performance Analysis*. Artech House Radar Library, Dec. 1986.

[5] Interlink Networks, Inc., "A practical approach to identifying and tracking unauthorized 802.11 cards and access points," Tech. Rep., 2002.

[6] F. Lassabe, O. Baala, P. Canalda, P. Chatonnay, and F. Spies, "A Friis-based calibrated model for WiFi terminals positioning," in *Proceedings of IEEE Int. Symp. on a World of Wireless, Mobile and Multimedia Networks*, Taormina, Italy, Jun. 2005, pp. 382–387.

[7] X. Li, "Rss-based location estimation with unknown pathloss model," *Wireless Communications, IEEE Transactions on*, vol. 5, no. 12, pp. 3626–3633, Dec. 2006.

[8] S. Mazuelas, A. Bahillo, R. Lorenzo, P. Fernandez, F. Lago, E. Garcia, J. Blas, and E. Abril, "Robust indoor positioning provided by real-time rssi values in unmodified wlan networks," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 3, no. 5, pp. 821–831, Oct. 2009.

[9] P. Bahl, A. Balachandran, and V. Padmanabhan, "Enhancements to the radar user location and tracking system," Microsoft Research, Tech. Rep., 2000.

[10] J. M. Keenan and A. J. Motley, "Radio coverage in buildings," *British Telecom Technology Journal*, vol. 8, no. 1, pp. pp. 19–24, 1990.

[11] A. Smailagic and D. Kogan, "Location sensing and privacy in a context-aware computing environment," *IEEE Wireless Communications*, pp. 10–17, 2002.

[12] M. Brunato and C. K. Kalló, "Transparent location fingerprinting for wireless services," in *Proceedings of Med-Hoc-Net 2002*, Sep. 2002.

[13] O. Baala and A. Caminada, "WLAN-based Indoor Positioning System: experimental results for stationary and tracking MS," in *Proc. IEEE International Conference on Communication Technology (ICCT'06)*, Guilin, China, Nov. 2006, pp. 1–4.

[14] M. A. Youssef, A. Agrawala, A. U. Shankar, and S. H. Noh, "A probabilistic clustering-based indoor location determination system," University of Maryland, Tech. Report CS-TR-4350, Mar. 2002.

[15] G. D. Forney, "The viterbi algorithm," in *Proceedings of the IEEE*, vol. 61, no. 3, Mar. 1973, pp. 268–278.

[16] F. Lassabe, P. Canalda, P. Chatonnay, and F. Spies, "Indoor Wi-Fi positioning: Techniques and systems," *Annals of telecommunications*, vol. 64, no. 9/10, pp. 651–664, Sep. 2009. [Online]. Available: http://dx.doi.org/10.1007/s12243-009-0122-1

[17] F. Lassabe, "Géolocalisation et prédiction dans les réseaux Wi-Fi en intérieur," Ph.D. dissertation, École doctorale SPIM, 2009.

[18] M. Cypriani, F. Lassabe, P. Canalda, and F. Spies, "Open wireless positionning system: a Wi-Fi-based indoor positionning system," in *VTC-fall 2009, 70th IEEE Vehicular Technologie Conference*. Anchorage, Alaska: IEEE Vehicular Technology Society, Sep. 2009, 5 pages. [Online]. Available: http://dx.doi.org/10.1109/VETECF.2009.5378966

[19] S. Zirari, P. Canalda, and F. Spies, "WiFi GPS based combined positioning algorithm," in *WCNIS'10, IEEE Int. Conf. on Wireless Communications, Networking and Information Security*, Beijing, China, Jun. 2010, pp. 684–688, 5 pages. To appear.

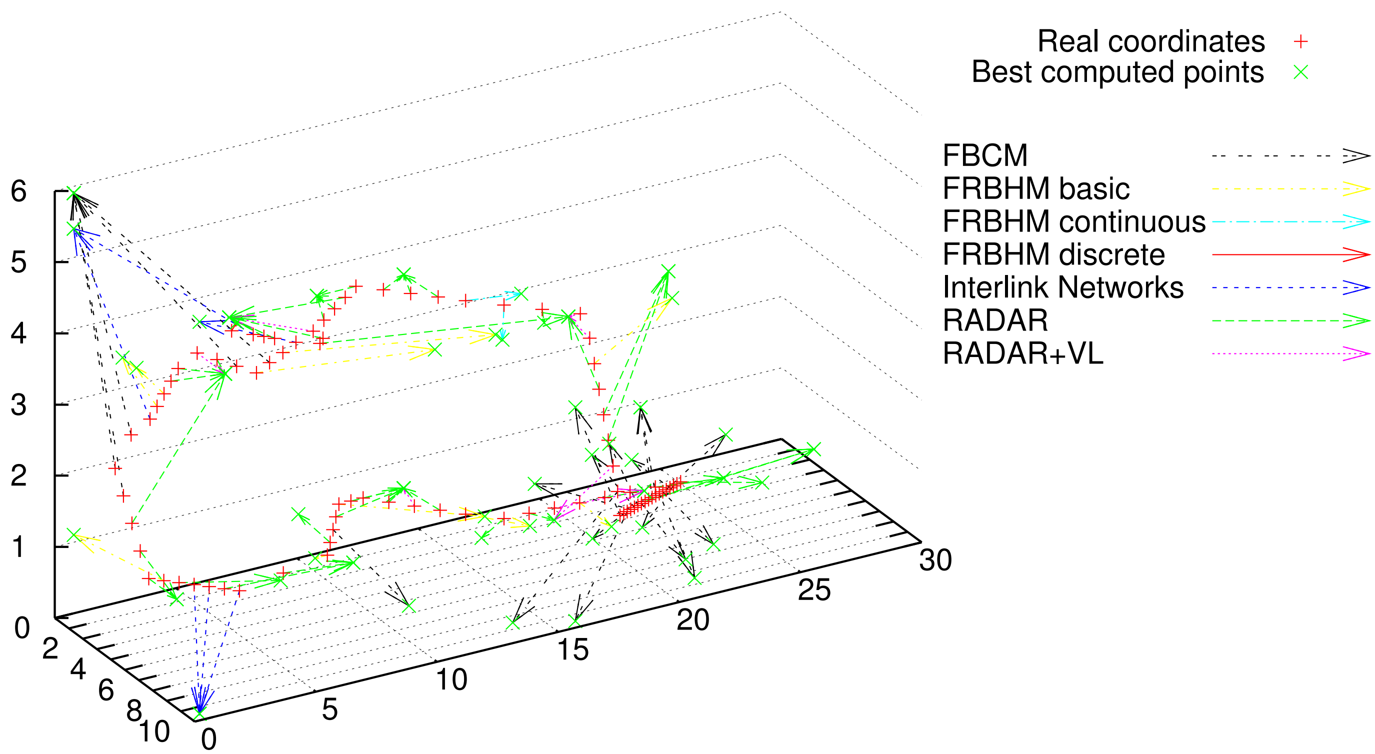

Fig. 11. Real path of the mobile, compared to the best computed position with a one-meter meshing. Each real position (red cross) is linked to its best computed position (green cross) by a colored arrow indicating which algorithm gave this position.

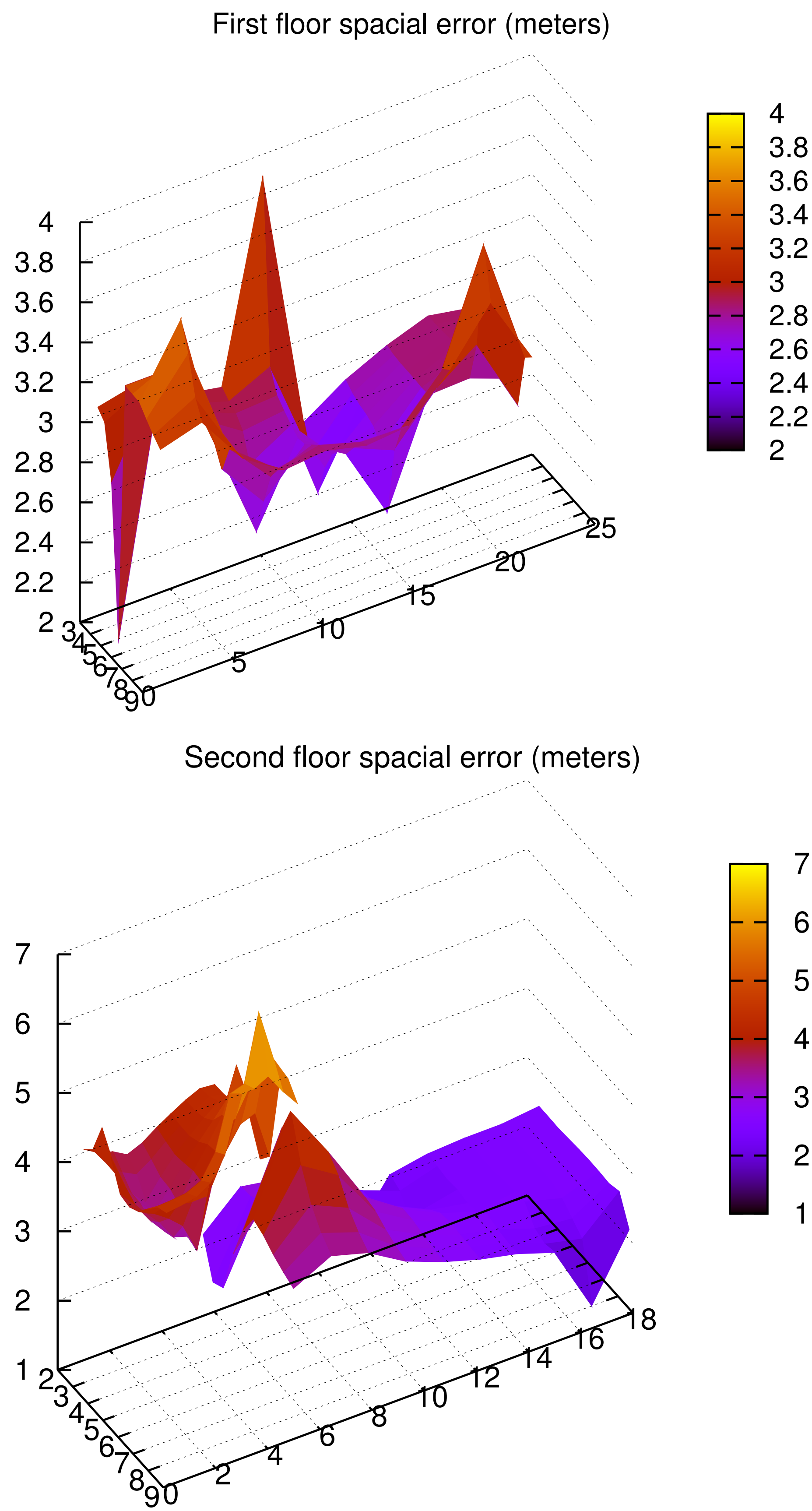

Fig. 12. The error obtained at each floor, in function of the real position (horizontal axes), is displayed on the vertical axis. The error displayed for a given point is the *best* error obtained, all algorithms taken together. The meshing used is 1 meter.

L I F C

Laboratoire d'Informatique de l'université de Franche-Comté
UFR Sciences et Techniques, 16, route de Gray - 25030 Besançon Cedex (France)

LIFC - Antenne de Belfort : IUT Belfort-Montbéliard, rue Engel Gros, BP 527 - 90016 Belfort Cedex (France)
LIFC - Antenne de Montbéliard : UFR STGI, Pôle universitaire du Pays de Montbéliard - 25200 Montbéliard Cedex (France)

http://lifc.univ-fcomte.fr