

# Fusing sensor information for location estimation

Odysseas Sekkas<sup>1</sup>, Stathes Hadjiefthymiades<sup>1</sup>, Evangelos Zervas<sup>2</sup>

<sup>1</sup>Communication Networks Laboratory, University of Athens, Dept. of Informatics and Telecommunications, Panepistimiopolis, Ilissia, 15784, Athens, Greece  
{sekkas, shadj}@di.uoa.gr

<sup>2</sup>Dept. of Electronics, TEI-Athens, Egaleo 12210, Athens, Greece  
zervas@ee.teiath.gr

**Abstract.** Pervasive computing is an emerging computing paradigm that provides new improved services for everyone, everywhere and at all times. In this paper\* we discuss a system which exploits data streams derived from sensors, in order to accurately estimate a key factor for pervasive computing and context aware applications: the location of a user. The term “sensors” includes Wi-Fi adapters, IR receivers, RFID tag readers, etc. The core of the system is the fusion engine which is based on Dynamic Bayesian Networks (DBNs), a powerful mathematical tool for integrating heterogeneous sensor observations. In closing, we provide an evaluation of the system as it comes out from the experimental results.

## 1 Introduction

In pervasive computing environments, location (position) is essential information as it is an important part of a user’s context. Applications can exploit this information for customizing their functionality. Such applications are termed location-aware applications and include tracking individuals, locating friends, offer services such as “nearest printer”, etc.

The location of a user is derived by various positioning methods. Triangulation, scene analysis, and proximity are the three main techniques for automatic location estimation [1]. Especially for indoor positioning, where GPS is not available, different approaches have been proposed. The majority of indoor positioning systems relies on different technologies (usually of the same kind) e.g., on Wireless LAN (Wi-Fi) signal strength measurements [3], IR beacons [4], or ultrasonic sounds.

At this point we will quote the definitions of accuracy and precision, the most important characteristics of a positioning system.

---

\* This work was performed in the context of the "PENED" Programme, co-funded by the European Union and the Hellenic Ministry of Development, General Secretariat for Research and Technology (research grant 03ED173).

- *accuracy* denotes the distance within which the system has the ability to locate a user, e.g. 1-10 meters
- *precision* denotes the percentage of time the system provides a specific accuracy, e.g. 80% of the time the system provides accuracy 1-5 meters (or else accuracy less than 5 meters).

The accuracy and precision are tradeable and it is clear that if we need less accuracy the precision that the system provides increases. For example, common GPS receivers can locate a user or object to within 10 meters (accuracy) for approximately 95% of time (precision). For accuracy less than 20 meters the precision would be greater than 95%.

During the last years several location systems have been proposed that use multiple technologies simultaneously in order to locate a user. The system we describe in this paper belongs to this category. It utilizes multiple sensors readings from Wi-Fi access points, IR Beacons, RFID tags, etc. to estimate the location of a user and provide input to location-aware applications. This technique is known better as sensor information fusion which aims to improve accuracy and precision by integrating heterogeneous sensor observations. The proposed location estimation system uses a fusion engine that is based on Dynamic Bayesian Networks (DBNs), thus, substantially improving the accuracy and precision in relation to other probabilistic approaches.

The paper is organized as follows: Section 2 discusses the related work and the differences of the proposed system from other indoor positioning systems. In Section 3 we present the basic location sensing technologies and the devices related with each technology. In Section 4, we present the layered architecture of the system and we discuss in detail the structure of each layer. In Section 5 we define the experimental setup environment and provide the experimental results from the evaluation of the system. Finally, Section 6 concludes the paper and provides some open research issues that we are currently working on.

## 2 Related work

Indoor positioning systems have been an active research area since the Active Badge [2] project in 1992. In this system, a badge worn by a person emits a unique IR signal. Sensors placed at known positions within a building detect the unique identifiers and relay these to the location manager software. Since then, several different indoor location systems have been proposed. A variety of indoor location systems use 802.11 (Wi-Fi) access points to estimate location. The RADAR [3] is a radio-frequency (RF) based system for locating and tracking users inside buildings. It operates by recording and processing received signal strength (RSS) information at multiple base stations positioned to provide overlapping coverage in the area of interest and combines empirical measurements with signal propagation modeling to determine user location. The RSS method is used also by the commercial system Ekahau [5]. This system uses a calibration-based approach and the location is calculated by means of RSS measurements at the client side. Another positioning system based on wireless LAN technology is described in [6].

The Cricket Location Support System [7] and Active Bat location system [8] are two systems that use the ultrasonic technology. Such systems use an ultrasound time-of-flight measurement technique to provide location information. They provide accurate location information but have also several drawbacks like poor scaling and a high installation and maintenance cost. For these reasons they are rather inaccessible to the majority of users.

Another category of location systems use multiple sensor readings (Wi-Fi access points, RFIDs) and sensor fusion techniques to estimate the location of a user. Location Stack [9] employs such techniques to fuse readings from multiple sensors. It is analogous to the OSI networking stack for organizing functionality in location aware applications. Another similar approach is described in [10]. The drawback of these systems is their inability of supporting mobile devices with limited capabilities (CPU, memory) because the location estimation is performed mainly on the client side, hence devices incur the cost of complex computations.

The proposed location estimation system described in this paper, utilizes data from sensors of different technologies to determine the location of a user. Our work differs from previous approaches in various aspects.

First, we use Dynamic Bayesian Networks (DBNs) for location inference. By using DBNs we obtain better location estimation results as along with heterogeneous sensor data that are processed in real-time we can also “fuse” past information about the user (previous location and historical data) and we can make predictions about the user’s future behavior.

Second, our system can support a variety of mobile devices (PDAs, palmtops) with low computational capabilities, as location computation takes place in a server residing in the fixed network infrastructure. Mobile devices send observations from sensors to this server whenever it is required and not all the time. Thus, the communication cost is reduced.

Finally, the system architecture we adopt has the advantage of easy management and renders the system more scalable. For example, the installation of a new access point is completely transparent to the users. Taking also into consideration that we can utilize the already existent wireless LAN infrastructure of a building and use a variety of inexpensive sensors, the proposed location estimation system offers a remarkable solution for indoor positioning.

### **3 Positioning Technologies**

In this section we present the principal technologies that are used for indoor positioning and describe their characteristics. We also discuss a categorization of the devices related with each technology.

#### **3.1 Wireless LAN (Wi-Fi)**

The most important wireless LAN standard today is the IEEE 802.11, also known as “Wi-Fi” that operates in the 2.4 GHz ISM band or 5GHz band and provides data rates

up to 54 Mbps. There are several specifications in the 802.11 family. The low-cost base stations (access points) can be easily installed in a building, providing wireless connectivity to those devices that are equipped with a wireless network card. Wireless LAN technology is used by several positioning systems which measure the signal strength from access points (RSS) to locate a user as described in the *Related work* section.

### **3.2 Bluetooth**

Bluetooth is a newer wireless local networking standard that also operates in the 2.4GHz ISM band. Compared to Wi-Fi, the bit rate is lower (1 Mbps), and the effective range is shorter. On the other hand, Bluetooth is a “lighter” standard, highly ubiquitous (embedded in most phones, PDAs, PC peripherals, etc.) and supports, in addition to IP, several other networking protocols. Bluetooth tags are small, pocket-size transceivers. As any other Bluetooth device, each tag has a unique ID which is periodically transmitted. A Bluetooth reader which intercepts the transmitted signals can be used to locate the tags.

### **3.3 Radio Frequency Identification**

Radio Frequency Identification is the technology used for security tags mainly on items in shops, ID cards, etc. Tags are powered by the magnetic field generated by a reader. The antenna of the tag exploits the magnetic energy and the tag transmits its ID or other information. These tags do not require any battery or maintenance and can be deployed in the doors of a building to detect object and person passing or in the furniture to detect proximity. The main problem with tags is their limited range. However, longer-range systems exist, but are more expensive and have higher power requirements.

### **3.4 Infrared technology**

Infrared (IR) Beacons are programmable devices that periodically emit their unique ID in the IR spectrum. Usually the range of these beacons is approximately 10-20 meters and the infrared receiver should have line of sight with the beacon in order to receive its ID, although sometimes this is not necessary due to reflections that occur in indoor environments.

### **3.5 Ultrasonic transmitters**

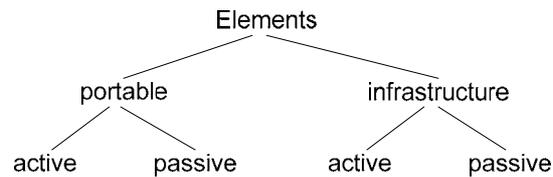
Ultrasonic signals are vibrations at a frequency above the range of human hearing, usually  $>20$  kHz. The devices used to receive and transmit ultrasonic signals are called transducers. Most of them are typically operating at frequencies between 40 kHz and 250 kHz. These devices are commonly used for distance measuring. In gen-

eral, they usually integrate a sensor that can receive or transmit an ultrasonic signal and another RF (or IR) transmitter/receiver which is used for synchronization.

### 3.6 Categorization

All the previously mentioned devices (elements) of different technologies (access points, beacons, tags, etc) can be found in indoor environments either deployed in the building or attached to mobile devices. Some of them emit information and others detect (read) information. According to their position and functionality the elements can be categorized as shown in **Fig 1**.

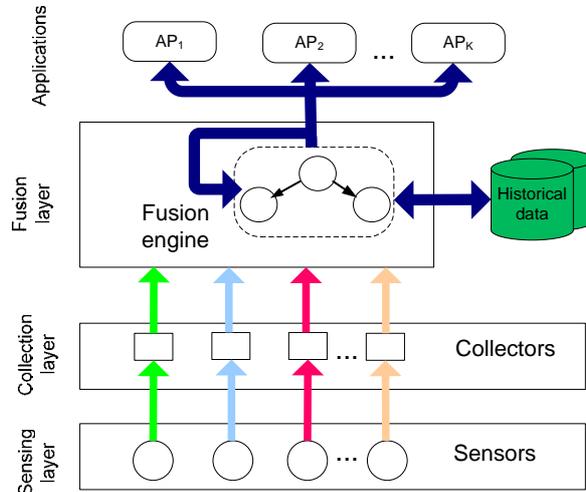
- Portable elements are those carried by users or attached to their mobile devices (e.g., RFID tags, Wi-Fi adapters, etc).
- Infrastructure elements are those that are attached to the building (e.g., Wi-Fi access points, IR beacons, RFID tag readers, etc).
- Active elements (also called also sensors) are those which detect a phenomenon or make measurement (e.g., RFID tag readers, Wi-Fi adapters, etc)
- Passive elements are those that emit information which is detected by active elements. Wi-Fi access points, IR beacons, Bluetooth tags, etc., are members of this category.



**Fig. 1.** Categorization of elements that are used for indoor positioning

## 4 System architecture

The architecture of the proposed location estimation system is organized in three layers: the sensing layer, the collection layer and the fusion layer. **Fig. 2** illustrates the generic architecture of the proposed system. In the same figure are also depicted location-aware applications which exploit the location information and data bases where is stored the personal profile of users or historical data about their behavior. The layered approach aims to facilitate effortless inclusion of new elements in order to improve the accuracy and the precision offered by the system. In the following paragraphs we provide a more detailed presentation of each layer and we discuss analytically their role.



**Fig. 2.** Architecture of the indoor location estimation system

#### 4.1 Sensing layer

This is the lowest layer of the system’s architecture and comprises sensors (active elements) of different technologies. Sensors (along with the appropriate software drivers) are attached either to the user’s mobile device (portable active elements) or are attached to the building (infrastructure active elements). Below, we briefly discuss these two categories.

##### Portable Sensors

A Wi-Fi adapter, which provides wireless connectivity and is built into most new laptops, can be also used to measure the received signal strength (RSS) from a Wi-Fi access point (passive infrastructure element). Similarly, the IR port of a handheld device or a laptop is used as reader for infrared transmissions from IR beacons that are attached to the building (wall-mounted). Aside from the “built-in” version of the Wi-Fi adapter and IR port, these sensors could be also added to the device via a PC card slot or USB.

##### Infrastructure Sensors

RFID tag readers belong to this category. Such readers detect an RFID tag and read its ID when the latter is in proximity. Users can carry RFID tags (portable passive elements) which have unique IDs. Furthermore, ultrasonic devices, which estimate the distance of a user from a known point and Bluetooth readers which can detect transmissions from Bluetooth tags (portable passive elements) carried by users, also belong to this category.

## 4.2 Collection Layer

This layer consists of software components called *collectors*. The main role of a collector is to interact with the appropriate sensor (active element) and retrieve (collect) the available measurements or events. Sensors may produce raw data in a variety of formats according to their type. Hence, the output of a Wi-Fi adapter is a stream consisting of RSS measurements from access points; IR and RFID readers generate a stream of proximity events. When these raw data arrive at the collection layer a pre-processing procedure is performed as described below.

### Preprocessing of raw data

Assume that a new RSS measurement arrives from a Wi-Fi adapter. Then, the appropriate collector (Wi-Fi collector) quantizes this on  $N$  discrete levels (values):  $S_1, S_2, \dots, S_N$ . If, for example, the value from the access point with ID  $AP2$  is between  $-70$  dBm and  $-60$  dBm the value “ $S_7$ ” is assigned to this infrastructure passive element. It is important to mention here that the number  $N$  of quantization levels depends mainly on the thresholds (lower and higher) of the access point’s transmitted power and the environmental conditions. According to these conditions (noise, etc.) we can adjust the number  $N$  as well as the spacing of the quantization levels.

An IR Beacon collector during this preprocessing procedure operates in a different and simpler way. The two possible states of an IR Beacon are: *Visible* and *Not Visible*. Assume that an IR receiver (portable active element) is in the range of the IR Beacon with ID  $IRB3$  (infrastructure passive element). This situation will cause a proximity event which will be detected and thus, the collector sets the  $IRB3$  to the value “*Visible*”. An RFID tag reader collector’s functionality is similar to the IR Beacon collector’s, as RFID tag readers detect proximity events, too.

### Tuple formation

After the preprocessing of raw data from the sensing layer, each collector forms a tuple of the type:

$$(user\_ID, IE\_ID, value)$$

where *user ID* is the unique identifier of a user, *IE ID* is the unique identifier of an infrastructure element and *value* is a measurement or an event.

For example, a Wi-Fi collector may form the following tuple:

$$(userA, API, S1)$$

which denotes that the Wi-Fi adapter (portable active element) of the mobile device of *userA* measures the RSS from access point  $API$  (infrastructure passive element) and the (quantized) RSS has value  $S1$ .

A possible tuple generated by a RFID tag reader collector would be

$$(userB, RFRI, Visible)$$

which denotes that an RFID tag (portable passive element) worn by *userB* (or attached to his/her mobile device ) is in proximity of the RFID tag reader with ID *RFR1*.

These tuples are then forwarded to the upper layer where a location estimation procedure is undertaken for each user.

### An example

We assume that all available infrastructure elements (and their identifiers) are the following:

Identifier	Type
AP1, AP2	Wi-Fi Access Points
IRB1	IR Beacon
US1, US2	Ultrasonic devices
RFR1, RFR2	RFID tag readers

In **Table 1** we present some possible inputs and outputs of collectors (for simplicity *userID* is omitted).

**Table 1.** Output of collectors after the raw data preprocessing and tuple formation

Collector	Input	Output
Wi-Fi	(AP1, -65 dBm) (AP2, -30 dBm)	AP1 = S1 AP2 = S3
IR Beacon	(IRB1, proximity event)	IRB1 = Visible
RFID tag reader	(RFR1, proximity event)	RFR1=Visible
Ultrasonic	(US1, 8 meters)	US1 = D1

Up to this point we know the values of infrastructure elements (active and passive) that are associated with a particular user. For the example shown in **Table 1** the data arriving to the *Fusion* layer are:

$$\{AP1 = S1, AP2 = S3, IRB1 = \text{Visible}, RFR1 = \text{Visible}, US1 = D1\}$$

In the following section where we describe the *Fusion* layer we will show how these infrastructure element values are exploited toward the location estimation of the user.

### 4.3 Fusion layer

This layer is the core of the system and includes the fusion engine. As mentioned in the *Introduction* section, this engine is based on a Dynamic Bayesian Network (DBN) which is used for the location inference. Below, we briefly discuss the basic concepts of Bayesian and Dynamic Bayesian Networks and, next, we adapt to our system the last ones. We assume that the reader is familiar with the theory of Bayesian and Dy-

dynamic Bayesian Networks and graphical models. For a more complete introduction the author is referred to [11], [12].

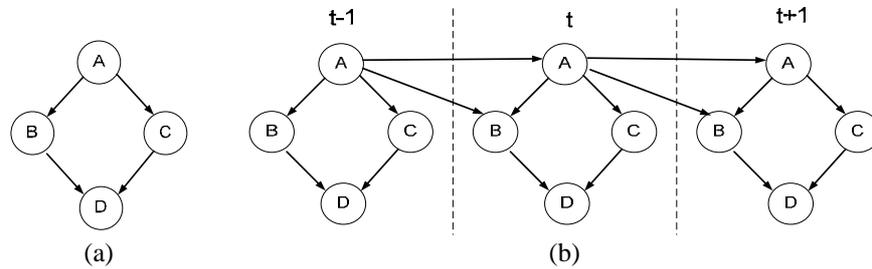
### Bayesian and Dynamic Bayesian Networks

Bayesian Networks (BNs) present a statistical tool that has become popular in the areas of machine learning and artificial intelligence [13]. They are well suited for inference because of their ability to model causal influence (cause - effect) between random variables.

A BN (see **Fig. 3a**) consists of two parts. The first part is a directed acyclic graph (DAG), representing random variables as nodes, and relationships between variables as arcs between the nodes. If there is an arc from node  $A$  to node  $B$  then we say that  $B$  is directly affected by  $A$  or  $A$  is the parent of  $B$ . Each node is conditionally independent from any other node given its parents. For example, in the Bayesian Network of **Fig.3a** we have:

$$P(D | A, B, C) = P(D | B, C). \quad (1)$$

The second part of a BN is a local probability distribution associated with each node of the graph. This describes the probability of all possible outcomes of the variable given all possible values of its parents. The parameters of this probability distribution would be estimated using observed data (Bayesian Network learning) [14]. The DAG and probability distributions together define the joint probability distribution.



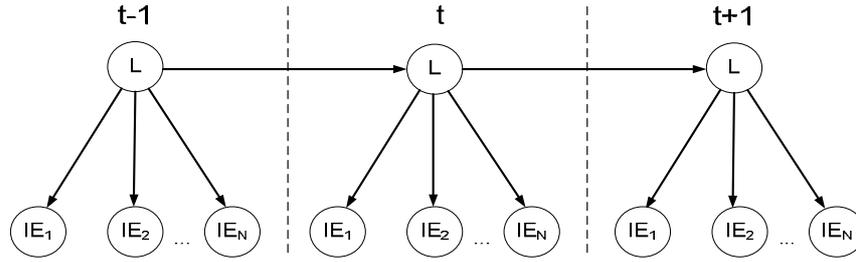
**Fig. 3.** (a) A Bayesian Network (BN) showing four (4) random variables and their dependencies. (b) A Dynamic Bayesian Network (DBN) showing dependencies between variables in different time-slots (A at  $t-1$  affects A and B at  $t$ )

A DBN extends the static BN by modeling changes of stochastic variables over time. Random variables in a DBN are also affected by variables from previous time slots (see **Fig. 3b**). For simplicity, it is assumed that the parents of a node are in the same or in the previous time slot (First Order Markov Chain).

### DBN integration in the location system

The DBN that is used in our location estimation system is depicted in **Fig. 4**. The incorporated random variables are:

- the location  $L$  of the user, which may take values from a set of  $K$  locations  $\{L1, L2, \dots, LK\}$ . This set may comprise symbolic locations e.g.  $\{room1, room2, corridorA, \dots\}$ .
- the  $N$  infrastructure elements  $IE1, IE2, \dots, IEN$ . The value field of those random variables depends on the type of element. Hence, an access point may take a value from the set  $\{S1, S2, \dots\}$  and an RFID tag reader from the set  $\{Not\_Visible, Visible\}$ , as described analytically in the *Collection layer* subsection.



**Fig. 4.** The Dynamic Bayesian Network of the location estimation system, representing the dependencies between random variables (location  $L$  and infrastructure elements  $IEi$ ) at different time slots

The random variable  $L$  at time  $t$ ,  $L^{(t)}$ , is directly affected by the random variable  $L$  at time  $t-1$ ,  $L^{(t-1)}$ , so  $L^{(t-1)}$  is the cause and  $L^{(t)}$  is the effect. This is a reasonable assumption as the location of a user is depended on his/her previous location. Also, the infrastructure elements at time  $t$  are affected by location at time  $t$ ,  $L^{(t)}$ ; the location of the user affects the value of an infrastructure element (e.g., the signal strength we measure from a Wi-Fi access point depends on our location). Random variable  $L^{(t)}$  is, simultaneously, cause and effect.

The local probability distributions that are associated with each node of the DBN are estimated with Bayesian network learning techniques. In particular, for every infrastructure element ( $IE_1, IE_2, \dots, IEN$ ) we estimate the probability distribution  $P(IE_i | L)$ . This can be achieved by taking into account the fixed positions of infrastructure elements, the indoor propagation models of RF and IR signals, the time of flight and indoor reflections of ultrasonic sounds, etc. A simpler technique of learning that we have adopted for our system is the method of sampling (signal, events) at every location for determining the values of infrastructure elements and the frequency of appearance of these values. According to this frequency we are able to form the probability distributions. In **Table 2** we present a probability distribution of a passive infrastructure element (Wi-Fi access point) with ID *API*.

**Table 2.** A possible probability distribution for access point with identifier  $API$ . We can see that the probability  $P(API=S2 | L=L_1) = 0.3$ , where random variable  $L$  denotes the location of the user

	$L_1$	$L_2$	...
$S1$	<b>0.5</b>	<b>0.0</b>	...
$S2$	<b>0.3</b>	<b>0.8</b>	...
$S3$	<b>0.1</b>	<b>0.2</b>	...
...	...	...	...

Furthermore, the probability distributions  $P(L^t | L^{t-1})$  (see **Table 3**) for location transition can be generated according to the structure of the building, the distance between two locations and the time required by a mobile user to cover this distance (e.g. is impossible for a person to cover a distance of 30 meters in just 2 seconds). It is important to note here that the determination of probability distributions takes place once, at system initialization (training phase).

**Table 3.** A possible probability distribution for location transition. The probability of transition from  $L1$  to  $L2$  is  $P(L^0=L_2 | L^{(-1)}=L_1)=0.1$

	$L_1$	$L_2$	...
$L_1$	<b>0.5</b>	<b>0.0</b>	...
$L_2$	<b>0.1</b>	<b>0.8</b>	...
$L_3$	<b>0.1</b>	<b>0.05</b>	...
...	...	...	...

### Location inference queries

After having structured the DBN of the fusion engine we can use it for location estimations. A location inference query might be: “*Where is user X (which is his/her location) given his/her previous location and given the values (observations) of infrastructure elements that are associated with this user?*”.

To answer this we compute for each of the  $K$  locations ( $L_1, L_2, \dots, L_K$ ) the following conditional probability:

$$P(L^{(t)} | L^{(t-1)}, O^{(t)}) . \quad (2)$$

which is the mathematical representation of the location inference query and denotes the probability of being at location  $L^{(t)}$  at time  $t$  (the requested location) given the already known value of the previous location  $L^{(t-1)}$  and given the values of the  $N$  infrastructure elements at time  $t$ ,  $O^{(t)}$ . For simplicity reasons we write:

$$\{IE_1^{(t)}, IE_2^{(t)}, \dots, IE_N^{(t)}\} = O^{(t)} . \quad (3)$$

Equation (2) can be converted to the following equation:

$$P(L^{(t)} | L^{(t-1)}, O^{(t)}) = \frac{P(L^{(t)}, L^{(t-1)}, O^{(t)})}{P(L^{(t-1)}, O^{(t)})}. \quad (4)$$

As the denominator of (4) does not depend upon the random variable  $L^{(t)}$ , it can be treated as a normalizing constant. Also, by taking into consideration that each node of our DBN is conditionally independent from any other node given its parents, we can compute the joint probability that appears in the numerator of (4). Hence, the following equation is derived:

$$P(L^{(t)} | L^{(t-1)}, O^{(t)}) = \alpha * P(L^{(t)} | L^{(t-1)}) * P(O^{(t)} | L^{(t)}). \quad (5)$$

The normalizing constant  $\alpha$  is computed easily as all probabilities (probabilities for all the  $K$  locations) must sum up to 1.

Finally, from (5) and (3) we obtain:

$$P(L^{(t)} | L^{(t-1)}, O^{(t)}) = \frac{P(L^{(t)} | L^{(t-1)}) * P(IE_1^{(t)} | L^{(t)}) * \dots * P(IE_N^{(t)} | L^{(t)})}{\sum_{i=1}^K P(L_i^{(t)} | L^{(t-1)}) * P(IE_1^{(t)} | L_i^{(t)}) * \dots * P(IE_N^{(t)} | L_i^{(t)})}. \quad (6)$$

The probability distributions  $P(IE_i | L)$  and  $P(L^{(t)} | L^{(t-1)})$  are known from the training phase, so we can now compute the probabilities for each location  $\{L_1, L_2, \dots, L_K\}$ . Whenever a probability on the numerator of (6) is equal to zero, it is unnecessary to compute the final probability as it is equal to zero, too. Thus, the calculations are pruned and the overall computation is optimized. The problem of location estimation is to find the location  $L_i$ , that maximizes the probability.

$$\max\{P(L_i^{(t)} | L^{(t-1)}, O^{(t)})\}.$$

The location with maximum probability is stored in the database and the profile of the user is updated. Moreover, the location information is forwarded to location-aware applications for the provision of LBS services. After that the system proceeds to the next location estimation in light of the current observations and the previous estimated location of the user.

## 5 System evaluation

### 5.1 Experimental setup

The evaluation of our system was performed using two technologies, Wi-Fi access points and IR Beacons. The experimental setup was the 2-floor building of the *De-*

partment of Informatics and Telecommunications (University Of Athens, Greece). Each floor has dimensions of  $30 \times 100$  meters.

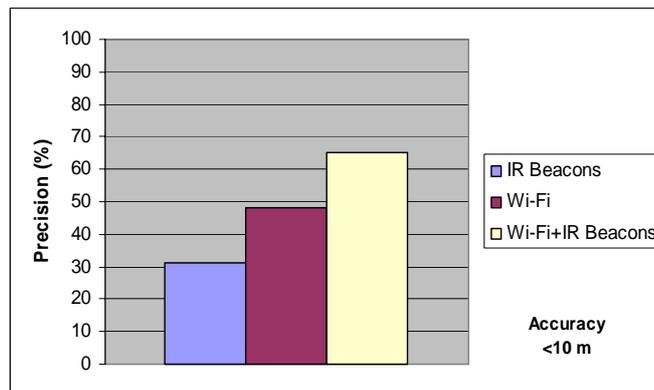
In total, we used 4 Wi-Fi access points (exploiting the wireless infrastructure of the building) and 5 IR Beacons of Lesswire AG company (<http://www.lesswire.com>). Moreover, we used 35 symbolic locations (*room1, room2, corridorA, entrance, ...*). During the DBN training phase, a training sequence (number of samples-measurements) from all locations and infrastructure elements was compiled and fed to the system. This resulted to the formation of probability distributions.

The whole architecture of the system (collectors, fusion engine, etc.) was implemented using Java™ technology which comprises an open and widely adopted standard. The server, where the fusion procedure and location inference is performed, was running on a PC powered by an Athlon 1800+ CPU with 512 MB memory. The operating system was Windows 2000. At the client (user) side was used an iPAQ™ Pocket PC equipped with an Orinoco™ wireless adapter and a “built-in” IR port.

The experiments were performed for a week and at different hours of a day in order to have a clear image of the system’s performance. The results are provided explicitly below.

## 5.2 Experimental results

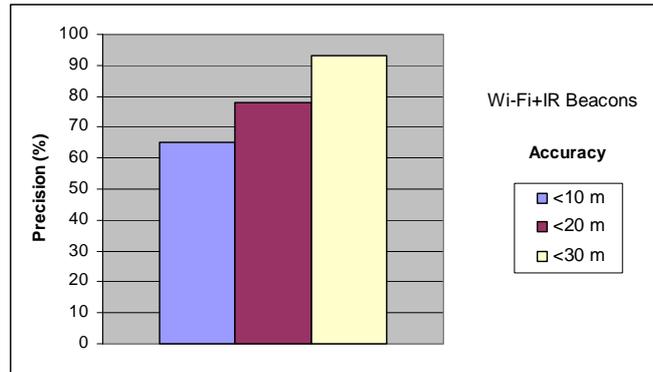
In **Fig. 5** is illustrated the precision obtained for accuracy less than  $10$  meters if we use only IR Beacons, only Wi-Fi access points and their combination. The precision for the first case (i.e. IR Beacons) is  $31\%$ . If we only use Wi-Fi access points precision climbs to  $48\%$ . Finally, our location estimation system that uses the combination (fusion) of the two heterogeneous technologies reaches precision  $65\%$ . As anticipated, the integration of heterogeneous technologies into the system improves its performance.



**Fig. 5.** Precision of different technologies for accuracy less than  $10$  meters

**Fig. 6** illustrates the precision of the system (Wi-Fi + IR Beacons) for different values of accuracy ( $<10$ ,  $<20$  and  $<30$  meters). As stated in the *Introduction* the improvement of the precision has a negative impact on accuracy. For accuracy less than

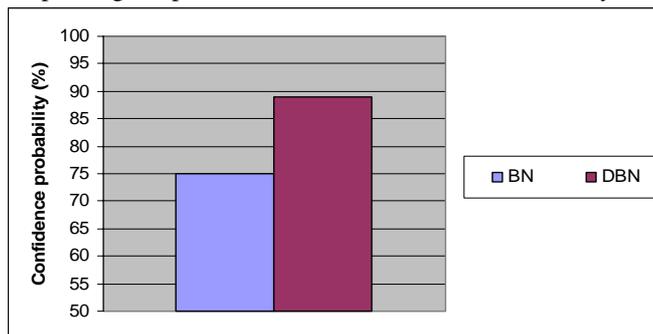
10 meters we obtain precision 65% and for accuracy less than 30 meters we can obtain precision 92%.



**Fig. 6.** Precision of the location system (4 Wi-Fi access points + 5 IR Beacons) for different values of accuracy (<10, <20, and <30 meters)

As discussed in the description of the *Fusion layer*, the system computes the probabilities for each location and the estimated location of the user is the location with the maximum probability. If the estimated location is correct, this maximum probability is defined as *confidence probability* of the system.

**Fig. 7** illustrates the *confidence probability* (mean value) of the system using a static Bayesian Network and a Dynamic Bayesian Network for the location inference process. In the first case (static BN) we do not take into consideration the previous location of the user for the estimation of the current location. The mean value of *confidence probability* was 75%. Conversely, through the use of a DBN, the mean value of *confidence probability* increased to 89%. It is obvious that the use of DBNs for location inference instead of static BNs increases the certainty on the user position estimation, thus improving the performance indicators of the location system.



**Fig. 7.** Mean confidence probability of the system, using static BN and DBN

## 6 Conclusion and future work

In this paper we presented a layered fusion system architecture which exploits information from sensors of different technologies to estimate the location of a user. A key difference from similar systems is the use of Dynamic Bayesian Networks for location inference. The use of DBNs improves our estimations. Along with sensor information we take into consideration the previous location of the user thus improving the system's performance. Additionally, the system supports a variety of mobile devices including those with restricted computational capabilities (PDAs, etc.) as they do not incur the burden of complex location calculations. Finally, the evaluation of the system in real conditions proved its appropriateness for indoor positioning.

Currently we are working on two issues which will have a direct impact on system's performance and scalability. The first issue is the use of "dead reckoning" techniques to improve the precision and accuracy that the system provides. In general "dead reckoning" is the process of estimating the position of a moving object based on speed and direction of travel and time elapsed since the last known position. In our case, for example, a user's mobile device which is equipped with an electronic compass and an accelerometer could provide information about the direction and speed of its owner. Taking also into account the last known position of the user and the time elapsed since then, we can predict the current position. Of course to estimate more accurately this position we have to include the current values of infrastructure elements.

The second issue that we are working on is the adoption of a distributed architecture for the system. As mentioned previously the location estimation takes place in a remote server. In the proposed distributed architecture the building is divided in regions or cells which their size depends on the dimension of whole building and the number of deployed elements. For each region there is one server responsible for location estimation. Servers of adjacent regions are interconnected in order to interchange information about the users (handovers between regions, etc). The distributed approach of the system enhances its performance, improves its scalability and makes it more robust in case of server failures.

## References

1. Hightower, J., Borriello, G.: Location Sensing Techniques. UW CSE 01-07-01, University of Washington, Department of Computer Science and Engineering, Seattle, WA, July 2001.
2. Want, R., Hopper, A., Falcao, V., Gibbons, J.: The Active Badge Location System. *ACM-Transactions on Information Systems*, 10, 91-102. (1992)
3. Bahl, P., Padmanabhan, V.: RADAR: An In-Building RF-Based User Location and Tracking System. *Proceedings of IEEE INFOCOM*, pp. 775-784 (Tel-Aviv, Israel).(2000)
4. Sonnenblick, Y.: An indoor navigation system for blind individuals. In *CSUN Center On Disabilities*, editor, CSUN 1998 Conference, California State University Northridge, Los Angeles, March 1998.
5. Ekahau, Inc : Ekahau Positioning Engine. <http://www.ekahau.com/>

6. Ladd A., Bekris K., Rudys A., Kavradi L., Wallach D., Marceau G.: Robotics-based location sensing using wireless ethernet. In Proceedings MOBICOM-02, pages 227–238, New York, September 23–28 2002. ACM Press
7. Nissanka, B. Priyantha, A., Balakrishnan, H.: The cricket location-support system. In Proceedings of MOBICOM 2000, pages 32–43, Boston, MA, August 2000. ACM Press.
8. Harter, A., Hopper, A., Steggle, P., Ward A., Webster P.: The anatomy of a context-aware application. In Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom '99), 1999.
9. Graumann, D., Lara, W., Hightower, J., Borriello, G.: Real-world implementation of the Location Stack: The Universal Location Framework. In Proceedings of the 5th IEEE Workshop on Mobile Computing Systems & Applications (WMCSA 2003), pp. 122–128, Oct. 2003
10. King, T., Kopf, S., Effelsberg, W.: A Location System based on Sensor Fusion: Research Areas and Software Architecture. Proc. of 2nd GI/ITG KuVS Fachgespräch "Ortsbezogene Anwendungen und Dienste", Stuttgart, Germany, June 2005
11. Jensen F.: An Introduction to Bayesian Networks. SpringerVerlag, New York, 1996.
12. Mihajlovic, V., Petkovic, M.: Dynamic Bayesian Networks: A State of the Art. Technical report (2001). Report no. TR-CTIT-01-34, Centre for Telematics and Information Technology, University of Twente, Enschede, The Netherlands.
13. Cowell, G., David, P., Lauritzen, L., Spiegelhalter J.: Probabilistic Networks and Expert Systems. Springer Verlag, Statistics in Engineering and Information Science, New York. (1999)
14. Heckerman, D.: A tutorial on learning with Bayesian networks. In *Learning in Graphical Models*, M. Jordan, ed.. MIT Press, Cambridge, MA, 1999