Exploring Quality in MultiSensor Pervasive Systems - A Localization Case Study

Stefano Bonetti #1, Sharad Mehrotra *2, Nalini Venkatasubramanian *3

# *University of Bologna and* * *University of California, Irvine*

1 stefano.bonetti2@studio.unibo.it, 2 sharad@ics.uci.edu and 3 nalini@ics.uci.edu

Abstract—**This paper addresses quality vs. cost tradeoffs in multisensor pervasive spaces. Specifically, we focus on a case study that uses location sensing in instrumented pervasive spaces executing a variety of applications ranging from social networking to surveillance and security. Location sensing is vital in creating effective situational awareness for mission critical applications, e.g. localization of firefighters and rescue personnel in emergency response scenarios as well as general purpose location based services. While advances in GPS and other technologies have enabled high quality outdoor localization, the realization of accurate indoor location technologies is significantly more complex. We present a generalized indoor localization framework that composes inputs from multiple localization technologies in a quality-aware manner so as to meet the accuracy/cost demands of diverse applications.**

I. Introduction

Advances in embedded computing, networking, and related information technologies allow the creation of Instrumented Pervasive Spaces that deploy a variety of sensing devices, communication networks and IT services in the real world. Such IPS enable monitoring of the state of the physical world, its artifacts (including humans), as well as activities within this world. The resulting pervasive spaces offer numerous benefits by making it feasible to realize applications where wherein the system observes its own state (including humans and other objects) to dynamically adapt itself based on its current situation to offer new functionalities or bring unprecedented improvements in efficiencies to real-world tasks. Of particular interest are pervasive environments where human activities and interactions are monitored to support decision making tasks. Such applications include surveillance, critical infrastructure protection, personalized environments, social networks and smart spaces. The Responsphere infrastructure at UC Irvine is one such instrumented pervasive space; it consists of of a variety of sensors (video cameras, sensor mounted mobile robots, people counters, RFID, acoustic sensors, thermal and gas sensors) dispersed over approximately a third of the campus connected via a variety of network and communication technologies (802.11, cellular, mesh, and powerline networks). Responsphere is already used to implement variety of pervasive applications and functionalities within UCI. For instance, using a mixture of video and RFID technologies it has been used to implement social policies for the use of shared common facilities in campus buildings (kitchens, labs); it has been used to study social experiments on artifact usage, as well as to conduct and monitor a variety of emergency drills (such as building and region evacuations). Enabling cost-effective and high quality data collection from sensors will improve the functionality and QoS of the deployed applications; however, there exist several challenges in its realization and deployment. Continuous multimodal sensing requires significant network and storage resources. Capturing activities using sensors, e.g. video sensors, may violate the privacy of the target individuals in the space. Data captured from sensors may be erroneous due to inherent imprecision of the sensing devices and dynamicity of the underlying sensing and communication infrastructure. The goal of data collection service is to efficiently capture information from the various sensors/data producers at desired levels of accuracy and granularity in order to meet the information quality, timeliness and reliability needs of data consumers given storage and communication constraints.

This paper explores the notion of quality-aware data collection in multisensor pervasive spaces via one specific case study, localization in instrumented pervasive spaces, where cost vs. quality tradeoffs are inherent. Localization is an important task in pervasive applications; it is a fundamental service used for target tracking, monitoring individuals, providing location based services etc.. Localization technologies have been extensively researched and developed resulting in a multitude of indoor and outdoor location systems. For outdoor localization, GPS technology is currently the de facto standard - it enables a cheap, robust solution for a majority of location based services. On the other hand, indoor localization mechanisms are more complex to design since they (a)depend on structural characteristics of the building, (b)differ in terms of levels of accuracy and precisions they can achieve, (c)have varying deployment and operational costs, e.g. the level of infrastructure support required, and (d) pose inherent limitations in diverse settings (e.g., sunlight in IR based mechanisms).

In this paper, we outline an general extensible localization framework that enables seamless fusion of multiple localization technologies. Such a framework will support the needs of multiple applications that need indoor and outdoor localization. Consider, as an example, the special needs of firefighters, rescue personnel and the emergency response community in general. Location sensing is amongst the most vital and important needs for creating effective situational awareness in firefighter settings; various studies have established a need for such technology for fire fighter safety during search and rescue operations. Any general-purpose localization framework should encompass three aspects. First, it must be able to accomodate the plethora of localization technologies that have been developed, such as GPS, GSM, Wi-Fi, Ultrasound, UWB, Inertial sensors, IR, etc. These technologies differ in many aspects - operational costs, infrastructure requirements, levels of accuracy they can achieve, efforts needed to calibrate and use the technology, etc.[?] For instance, Wi-Fi based localization requires little new infrastructure (assuming that the structure in which localization is to be implemented is already instrumented for networking), and coupled with significant training and calibration using fingerprinting/scene analysis could lead to accuracy levels of about 2-3 meters. UWB technology, on the other hand, can provide significant accuracy, and unlike WI-FI fingerprinting approach can be rapidly deployed; however, it is expensive and also requires appropriate placement of outdoor base units which may not be feasible in all situations.

Secondly, the framework must address diverse location query needs that come from different applications. Location queries of interest could range from simple position queries (e.g., "where am I?") that request specific (either absolute or relative) location information or Boolean queries that evaluate a boolean predicate based on location of entities. An example of a Boolean query of particular importance for firefighters are *proximity queries*(e.g., "are the members of a firefighter *company* within a specified range of each other?"). Note that most location based queries can be appropriately answered if accurate location information is available for every target/entity, meaningful results for a large class of queries does not require accurate localization of every entity involved. For instance, proximity queries can still be meaningfully answered even though the location technology might not provide accurate location information for all individuals.

Finally, interesting design tradeoffs arise based on the distributed nature of computing localization information - i.e. where the location query originates (at a centralized server, or on a mobile device), where the location information is computed (at the server or on the located object itself), and who needs the localization to be performed (the located object or a different entity). For instance, GPS receivers use triangulation to compute their location, the GPS satellites and corresponding infrastructure are oblivious to who needs that information. The choice of how, where and for whom localization information is computed has numerous implications on factors such as privacy, power consumption, timeliness, etc. Tradeoffs and design choices can be made in determining design choices for indoor localization.

Proposed Framework: The key observation on which our localization framework is based is that localization applications can often tolerate different levels of quality (since no technology can localize perfectly is ubiquitously available, applications typically have to be designed to accomodate such quality degradations). The notion of quality here is multifaceted and refers to properties such as precision/accuracy, confidence in the location accuracy, timeliness of the location information, etc. Likewise, we wish to exploit knowledge of which technologies work well in specific situations to combine/fuse localization mechanisms to improve the overall localization performance. For instance, Wi-Fi fingerprinting based location information can be coupled with techniques such as signal strength analysis to more accurately locate targets. Indeed such fusion techniques have been explored in the past e.g., coupling inertial sensing with UWB to improve on the localization accuracy [?].

The goal of our framework is two folds: (i) develop a generic approach whereby diverse location technologies can be fused together to meet the diverse needs of the location applications (e.g., different location queries), (ii) to enable such location fusion to occur in a cost-effective manner. Location applications/queries can use our proposed framework in two complementary fashions. The first approach would be to identify the best answers (with least uncertainty) given the total cost budget. An alternate formulation would be to minimize cost to produce results, if possible, at a given level of quality. Cost metrics can be defined to subsume a variety of operational and deployment factors. In this paper, we describe an initial prototype of such a framework and use it to integrate a variety of scene analysis approaches (Wi-Fi based and Bluetooth based localization technologies among many others). We have experimented with combining such technologies for direct localization queries in the context of fire fighter drills within our CS Building (Bren Hall); our initial results show significant accuracy in distance estimation.

II. Information quality in localization

Every location-based application or service requires a certain accuracy and a certain precision to correctly carry out its task. For instance, intelligent transportation applications such as routing assistance requires position information whose errors cannot exceed 15-20 meters. To determine whether a person is at work, it would be sufficient to know if whether he or she is inside a building or not. Robotic control applications may require that the positioning error not exceed a few millimetres. The quality of location information can be expressed using two concepts: *accuracy* and *precision* [?]. The accuracy is the average distance between the actual and the computed position of a generic located object. The precision is a percentage value which indicates how often we can expect to get a certain accuracy from a measurement. For example, cheap GPS receivers can locate positions within 10 meters 95% of the time. These two attributes cannot be analysed separately, and every location system must find a satisfying trade-off between them. In this section we try to formalize the problem of localization in the broadest and most generic fashion, to meet the needs of most applications that express their location needs via a querying framework. The following is a taxonomy of location query types:

1) Absolute location queries: these queries require the system to output an absolute value for the target's location. Applications may specify a bound on the accuracy of the answer, e.g. "Where am I (with 100 meters accuracy)?".
2) Relative location queries: this addresses situations in which users are not interested in absolute positions.

To correctly answer the query "How far am I from firefighter F?" - it may be unnecessary to obtain absolute location information (based on available technologies).
3) Absolute boolean queries: in this case the boolean predicate is used to check the relationship between targets and fixed location or regions, e.g. "Is firefighter F in building B?".
4) Relative boolean queries: relative location can be important when checking the status of a group of targets, e.g. "Are John and Jane within 10 meters from each other?".

In other words, queries can express precision constraints and certainty parameters as inputs. Furthermore, location queries are usually submitted specifying, either implicitly or explicitly, a proper answer set. For instance, the query "In which room am I?" explicitly indicates a symbolic answer set, whose elements are the different rooms in a considered environment. On the contrary, the simplest natural language location query, "Where am I?" does not contain that information. However, since a query must specify either a granularity value or region, these values must refer to a specific answer set.

III. Localization framework definition

In this section, we develop a generalized localization framework in which any location technique can be modeled as a generic location component. The main purpose of such a framework is to answer location queries with the best trade-off between accuracy, precision and cost, choosing the fittest location technology or the best combination of technologies for each query. We now present elements of our framework.

A. Location components

In our framework, we characterize the different location technologies as location *components*. Components are able provide non-boolean, uncertain answers among a defined *answer set*, which is the space of the possible query answers. It is reasonable to assume that all components' outputs are on the same answer set, since they are collaborating to solve the same query. For example, a physical absolute coordinate system (*(latitude, longitude, height)*) represents the most obvious choice. This assumption might not be correct if, for instance, we start considering location components which can output a relative location. For the sake of simplicity we will now focus on components which share the same answer set, which are called *homogeneous* components. Non-homogeneous components will be further briefly discussed. The *input set* is the space of input data that localization components need to actually compute their answers. To model *granularity* and *uncertainty* of answers, a component should actually output a probability mass function (*pmf*) on the answer set. A valid query solution could then be extracted from such function.

Definition (nondeterministic component) Let A be a countable set and $\mathcal{F}$ the space of all *pmf*s on A. A *nondeterministic component* is a function $c_{D\to\mathcal{F}(A)}$ being D a generic input domain. Let's assume that $f = c(d)$

- D is called *input set*
- A is called *answer set*
- $\forall A^* \subseteq A$ is called *answer subset*
- if A^* is an answer subset:
  - $g_{A^*} = |A^*|$ is called *granularity* or *accuracy* of the subset
  - $p_{A^*,f} = \sum_{a\in A^*} f(a)$ is called *f-certainty* or *f-precision* of the subset

Definition (homogeneous components) Given two nondeterministic components $c_{D\to\mathcal{F}(A)}$ and $c'_{D'\to\mathcal{F}(A')}$, they are said to be homogeneous if $A \equiv A'$.

A component can be qualitatively compared to other components on the base of the trade-off between cost and efficacy. We define the two properties cost and selectivity [1] of a component. The *cost* information is modeled as a per-input cost. In other words, it is a measure of computational effort to output the *pmf*, given a certain element of the input set. Such cost can be considered to be not dependent on the specific formulated query. Given a location query, the *selectivity* of a component is the probability that it will not manage to output a valid answer for that query. This property is directly query-dependent; it can be estimated by choosing a query samples and noting if the component execution is successful.

B. Location queries

Queries can be solved by components in relation to a certain input *d*. Answers given by components can be valid answers for a certain query or not, e.g. the requested precision bounds may not be satisfied, or answer granularity is too coarse.

Definition (query) Let A be an answer set. Then,

- a *simple query* q on A is a tuple (g_q, p_q) such that $g_q \in \mathbb{N}, p_q \in \mathbb{R}, 0 \leq p_q \leq 1$. g_q is called *granularity* of q. p_q is called *certainty* of q.
- a *boolean query* b on A is a tuple (A_q^*, p_q) such that $A_q^* \subseteq A, p_q \in \mathbb{R}, 0 \leq p_q \leq 1$. A_q is called *region* of q. p_q is called *certainty* of q.

Definition (query satisfaction) Let

- D be an input set
- A be an answer set
- $q = (g_q, p_q)$ be a simple query on A
- $b = (A_q^*, p_q)$ be a boolean query on A
- $c_{D\to\mathcal{F}(A)}$ be a component
- $f = c(d)$ where $d \in D$.

Then,

- q is said to be *satisfied* by c with input d iff $\exists A^* \subseteq A$ such that $(g_{A^*} \leq g_q) \wedge (p_{A^*,f} \geq p_q)$
- b is said to be *satisfied* by c with input d iff $\exists A^* \subseteq A$ such that $(A_q^* \subseteq A^*) \wedge (p_{A^*,f} \geq p_q)$.

The output of a component indicates how probability is distributed among answers, according to that component. The *pmf* can cover several answer subsets which meet the the query parameters.

Example Let's consider the following function f: $f(l_1) = 0.6; f(l_2) = 0.3; f(l_3) = 0.1; f(x) = 0,\ \forall x \neq l_1, l_2, l_3$ where $l_1, l_2, l_3 \in A$. Let q be a simple query $(g_q = 3, p_q = 0.8)$ on A. Both the answer subsets (l_1, l_2) or (l_1, l_2, l_3) fit the query parameter.

Given an answer subset A^*, the ratio $\frac{p_{A^*}}{g_{A^*}}$ represents the average probability of the single answers in the subset. The subset to look for is the one which maximizes this ratio.

Definition (best answer subset) Given an answer space A and a pmf $f_{A \to \mathbb{R}}$. A *best answer subset* for f, according to a query (g_q, p_q) on A is a subset $A^*_{best} \subseteq A$ such that:

1) $g_{A^*_{best}} \leq g_q$ (*granularity acceptability*)
2) $p_{A^*_{best}} \geq p_q$ (*certainty acceptability*)
3) $\forall A^* \subseteq A$, $\frac{p_{A^*_{best}}}{g_{A^*_{best}}} \geq \frac{p_{A^*}}{g_{A^*}}$ (*maximum certainty/granularity ratio*)

Example The answer subset ratios are:

- For subset $A^{'} = (l_1, l_2)$, $\frac{p_{A'}}{g_{A'}} = \frac{0.6+0.2}{2} = 0.45$.
- For subset $A^{''} = (l_1, l_2, l_3)$, $\frac{p_{A''}}{g_{A''}} = \frac{0.6+0.2+0.1}{2} = 0.333$.

C. Component Aggregation and Status Networks

This section deals with the composition rules for components, discussing a structure to link components and to combine their capabilities. We explore two methods using which diverse localization components can be merged together for the purpose of properly answering location queries - (a) pipelined execution and (b) parallel evaluation. *Pipelining* is useful to split a technology into a pipeline of stages with increasing efficacies and costs. For example, in Wi-Fi fingerprinting, the basic nearest neighbor approach [2] can be refined with more techniques exploiting directional information [3]. A pipeline of technologies can be used to progressively refine estimates as new information becomes available. The answer coming from the last stage of the pipeline includes information from all the previous stages. *Parallel evaluation* can be used to enhance accuracy. When two or more technologies used in combination yield the same result, the outputs can be combine to obtain an aggregate estimate providing increased confidence in the result. In other words, a greater precision will be achieved.

We combine these two orthogonal composition methods into a component graph called *status network.* In the most general situation, we consider a set of pipelines, each of which may have a different number of components. Assume that we have N pipelines with respectively $l_1, \cdots, l_N$ components. The status network is defined by its nodes and its edges. The set of nodes, called *statuses*, is $S = \{0, \cdots, l_1\} \times \cdots \times \{0, \cdots, l_N\}$. Each status is actually a vector of integer values, the *i*-th values representing which version of *i*-th pipeline has been evaluated. For what concern edges, given a certain status $\mathbf{s} = (s_1, \cdots, s_N)$, its *following statuses* are the elements of the set $F_\mathbf{s} = \{f = (f_1, \cdots, f_N) \mid \exists i \in 1 \cdots N$ s.t. $s_i < f_i \wedge s_j = f_j\ \forall j \neq i\}$. An oriented edge is assumed to be established from $\mathbf{s}$ to $\mathbf{s}'$, $\forall \mathbf{s} \in S$, $\forall \mathbf{s}' \in F_\mathbf{s}$.
The designed framework is a broad generalization of the multi-version predicate framework designed by Lazaridis and Mehrotra in [1]. Figure 1 shows a graphical representation of component pipelines and the correspondent status network.

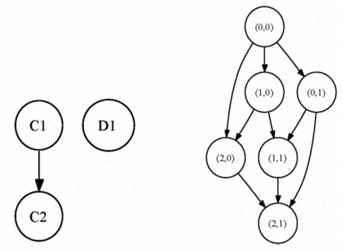

Fig. 1. A two-component pipeline and a one-component pipeline (LHS), and the resulting status network (RHS)

During the query solution process, each status will output a *pmf* consistent with the components which have been evaluated so far. The output of the start status $(0, \cdots, 0)$, when no components has been evaluated yet, can be defined as a uniform probability distribution on the answer set. This directly proceeds from the fact that there are no clues about the target's location, so each answer is as good as any other. What is missing is a way to combine the contributions of different components, i.e. a way to aggregate *pmf*s in a consistent way. As briefly indicated before, parallel use of different location components can provide a precision-enhanced answer. Components which are evaluated in parallel output different pmfs which must be combined to obtain a global answer. The problem is similar to the probability aggregation problem studied in probabilistic risk analysis where in the presence of limited information, decision-makers and risk analysts usually turn to judgments coming from different experts/sources [4] to obtain as much information as possible. One may use a straightforward Bayesian updating scheme to merge results from different sources [5]. Let $p_i (i = 1, \cdots, n)$ denote expert i's stated probability that θ occurs. Expressed in terms of the posterior odds of the occurrence of θ, $q* = \frac{p^*}{1-p^*}$ the model is defined as follows:

$$q^* = \frac{p_0}{1 - p_0} \prod_{i=1}^{n} \frac{f_{1i}(p_i | q = 1)}{f_{0i}(p_i | q = 0)},$$

where f_{1i} (f_{0i}) represents the probability of source i generating probability p_i conditional on the occurrence (non-occurrence) of θ and p_0 denotes the prior probability $p(\theta = 1)$. Since in our case we are dealing with a probability mass function, the single computed probabilities will have to be normalized to make sure that the aggregation function still meets the pmf requirements (i.e. the values sum up to 1). Thus, the final aggregation formula is:

$$p_j^* = \frac{\frac{q_j^*}{1+q_j^*}}{\sum_{j=1}^{N} \frac{q_j^*}{1+q_j^*}},$$

p_j^* indicating the aggregated probability of the j-th answer of the answer set, and N being the answer set cardinality.

So far, we made the assumption that all the components which form a certain network share the same answer set. In the following the problem of using components with different answer sets is addressed by defining the concept of *translator.*

Definition (translator) Given two answer spaces A and B, let $\mathcal{F}_A$ and $\mathcal{F}_B$ be the spaces of all pmfs on A and B respectively. A translator from A to B is a function $t_{\mathcal{F}_A \rightarrow \mathcal{F}_B}$.

In practice, such a function will have to be implemented for a certain transition from an answer set to another (e.g. mapping symbolic places to geographical coordinates). Since they output a pmf on a specific answer set, the translators can be treated like components, therefore they can be inserted into the status network to be evaluated.

IV. Evaluating Localization Queries

Given the formalization and definitions above, we now focus on finding a way to properly choose the most suitable location components to answer a query while minimizing the total cost of the operation. Our solution leverages a prior technique used in database query optimization [1] that generates optimal query plans to evaluate *multi-version predicates* which are sequences ("trains") of selection predicates with increasing selectivity (m) and cost (c); this work shows how the cost of predicate evaluation can be reduced by utilizing cheaper, but less accurate, versions of the predicates to pre-filter tuples. The optimal generalized plan generation algorithm proposed in [1] can be adapted to our problem with non-deterministic localization components where the status network happens to be a generalization of the predicate train. We propose a novel algorithm, called EOGP (Extended Optimal Generalized Plan).

Extended Optimal Generalized Plan: The defined status network is more specifically a directed acyclic graph with one start and one terminal. Each edge represents the execution of a specific component starting from a specific status. Therefore, following a certain edge means to evaluate a certain component knowing which other components have already been evaluated. As a consequence, the cost of following a certain edge is given by the per-input cost of the component, times the number of inputs which are expected not to be filtered by the edge source status. As a result, not only the selectivity of each component, but in general the selectivity of each status has to be estimated.
Consistently with what stated in [1] a certain component per-input cost might not be constant. A pipelined component could exploit the results previously outputted by the pipeline, reducing the cost of computing its *pmf.* This concept can be modeled actually putting the costs on the status network edges instead of directly associating them with the components.

Definition (conditional cost) Given an edge $e_{\mathbf{st}}$ of a network, and being $\mathbf{s}$ and $\mathbf{t}$ respectively the source and the destination statuses of $e_{\mathbf{st}}$, the *conditional cost* of $e_{\mathbf{st}}$ is $C_{\mathbf{st}} = m^*_{\mathbf{s}} c_{\mathbf{st}}$, where $m^*_{\mathbf{s}}$ is the status selectivity of $\mathbf{s}$ and $c_{\mathbf{st}}$ is the shared cost of the edge $e_{\mathbf{st}}$.

All the edges of a network can now be marked with their conditional costs. Since a path on the status network can be mapped to a sequence of components, the total cost incurred by a sequence is the sum of the conditional costs on its path's edges. This means that the less expensive path from start to terminal identifies the less expensive component sequence. The problem is equivalent to the single-pair shortest path problem solved by
Dijkstra [6].

Algorithm 1 Extended Optimal Generalized Plan

```
Require: $Network$,$start$,$terminal$ {a status network, its start node
    and its terminal node}
Ensure: $S$ {the sequence of status on the minimum cost path}
 1: for all status $s$ in network do
 2:     $dist[s] \leftarrow \infty$
 3:     $previous[s] \leftarrow$ undefined
 4: end for
 5: $dist[start] \leftarrow 0$
 6: $Q \leftarrow$the set of all statuses in $Network$
 7: while $Q \neq \emptyset$ do
 8:     $s \leftarrow$ status in $Q$ with smallest $condCost[]$
 9:     $Q \leftarrow Q \setminus \{s\}$
10:     for all sibling $t$ of $s$, where $t$ has not yet been removed from
        $Q$ do
11:         $alt \leftarrow dist[s] + condCost(s,t)$
12:         if $alt < dist[t]$ then
13:             $dist[t] \leftarrow alt$
14:             $previous[t] \leftarrow s$
15:         end if
16:     end for
17: end while
18: $S \leftarrow$empty sequence
19: $s \leftarrow$terminal
20: while $previous[s] \neq$ undefined do
21:     insert $s$ at the beginning of $S$
22:     $s \leftarrow previous[s]$
23: end while
24: return $S$
```

The running time of this algorithm is known to be $O(|S|^2 + |E|) = O(|S|^2)$ where S is the network nodes set and E the edges' set.

V. Implementation and Evaluation

The proposed framework was implemented and evaluated in the Responsphere (http://www.responsphere.org) multisensor infrastructure at UCI. Figure 2 illustrates a basic schematic of the implemented framework with several simple localization techniques that have been adapted to fit in the framework (i.e. to provide probabilistic answers). Components involving the following localization techniques were implemented:

1) the *Wi-Fi fingerprinting-based* component exploits deployed WLAN access points using a database matching technique to localize targets. This technique involves a nearest-neighbor search on a data space of previously collected signal strength readings (fingerprints) [3];
2) the *GPS-based* component takes the coordinates provided by a GPS receiver and uses them to build a truncated Rayleigh distribution based on the accuracy value provided by the receiver itself [7];
3) the *Bluetooth-based* component is a simple anchor-based proximity localization system. This component outputs a uniform truncated pmf around fixed Bluetooth anchors;

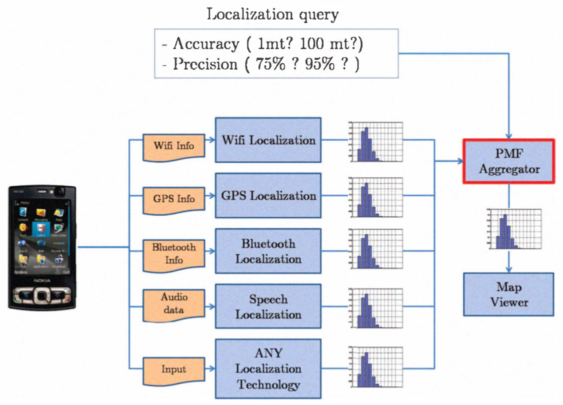

Fig. 2. Localization framework general architecture

4) the *speech recognition-based* component consists of a simple natural language parser was written to extract location information from recognized speech. These information are used to retrieve pmfs which were previously written and which are stored in a database;
5) the *accelerometer-based* component uses previously calculated pmfs as a prior. Movement information coming from an accelerometer are also used to better exploit location information from the past.

The whole localization system has been developed in two tiers: (1) the client tier, running on N95 devices, is made of a Python script and several Nokia packages to access the hardware modules of the smartphone. This script periodically inquires the device modules for information about the status of the Wi-Fi network, GPS receiver and all the input data needed by the components which might be available at every instant; (2) the server tier, which hosts all the business logic of the framework. All the framework modules were implemented inside SATware [8], a middleware for pervasive spaces developed at UCI, which offers a support for mobile software agents communication amongst many other features. Several components and the aggregation algorithm were incorporated in a number of SATware mobile agents. SATware middleware is suitable to host the defined localization system, because the modularity of the framework, defined formally, is preserved. During the tests, communication between the client and the server was carried out using the WLAN capabilities of the device: all collected information were packed in a UDP datagram and sent to the server. Quantitative tests for the selectivity estimation process and the pmf aggregation technique have been carried out using Wi-Fi fingerprinting and Bluetooth-based components and are briefly described in the following. The testbed for the experiments was the UCI-ICS Bren Hall building. To test in real-time the answers given by components the demo GUI has been created. The results obtained using the Bluetooth-based component and the Wi-Fi component are shown respectively in figure 3(a) and in figure 3(b). The plotted surfaces show how selectivity varies with certainty and granularity. The volume bounded by the selectivity surface can vary from 0 (every possible query always satisfied) to $100 \cdot 100$ (every possible query never satisfied). The lower is the surface-bounded volume, the higher is the efficacy of the component in the query solving process. Figure 3(c) shows that the two components aggregated in our localization framework allow to provide a better accuracy/precision answer.

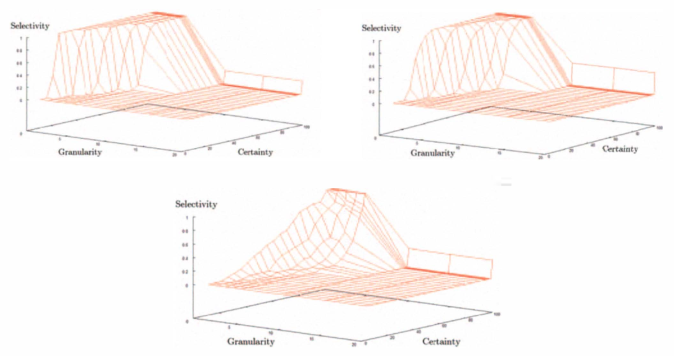

Fig. 3. Estimated selectivities of the single test components (a) Bluetooth, (b)Wi-Fi fingerprinting and their aggregation

Discussion and Future Research Directions: We plan to extend our current framework design in many ways. First, we plan to expand the repertoire of base level localization algorithms into the framework. E.g., implementation and incorporation of particle filtering methods, incorporation of inertial sensors. Second, we will explore techniques to effectively incorporate multimodal input such as speech as a localization methodology into the framework. Third, we will explore a range of options to combine diverse localization mechanisms into a localization strategy. While we have estimation techniques incorporated into the framework, current approaches are somewhat ad-hoc and also developed at coarse granularity (e.g., quality is considered for an algorithm at an average in the aggregate sense). Better estimation mechanisms are expected to result in better (faster/higher quality) answers. Finally, we will address a variety of application needs manifested through diverse query types.

References

[1] I. Lazaridis and S. Mehrotra, "Optimization of multi-version expensive predicates," in *Proceedings of ACM SIGMOD 2007*, 2007.
[2] P. Bahl and V. N. Padmanabhan, "Radar: an in-building rf-based user location and tracking system," 2000.
[3] B. Li, J. Salter, A. G. Dempster, and C. Rizos, "Indoor positioning techniques based on wireless," in *LAN, First IEEE International Conference on Wireless Broadband and Ultra Wideband Communications*, 2006.
[4] R. Clemen, "Combining probability distributions from experts in risk analysis," *Risk Analysis*, vol. 19, April 1999.
[5] P. Morris, "Combining expert judgments: A bayesian approach," *Management Science*, vol. 23(7), 1977.
[6] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, December 1959. [Online]. Available: http://dx.doi.org/10.1007/BF01386390
[7] M. Kobayashi, F. Ingels, and G. Bennett, "Determination of the probability density function of GPS(Global Positioning Systems) positioning error," in *Inst. of Navigation, Annual Meeting, 48 th, Washington.*
[8] B. Hore, H. Jafarpour, R. Jain, S. Ji, D. Massaguer, S. Mehrotra, N. Venkatasubramanian, and U. Westermann, "SATware: Middleware for Sentient Spaces," *Multimodal Surveillance: Sensors, Algorithms and Systems*, 2007.