18th International Conference on Information Fusion
Washington, DC - July 6-9, 2015

Improving Indoor Localization by User Feedback

Lukas Köping, Marcin Grzegorzek
Research Group for Pattern Recognition
University of Siegen
Siegen, Germany
Email: lukas.koeping@uni-siegen.de
Email: marcin.grzegorzek@uni-siegen.de

Frank Deinzer
University of Applied Sciences
Wuerzburg-Schweinfurt
Wuerzburg, Germany
Email: frank.deinzer@fhws.de

Szymon Bobek, Mateusz Ślażyński
Grzegorz J. Nalepa
AGH University of Science and Technology
Krakow, Poland
Email: szymon.bobek@agh.edu.pl
Email: mslaz@agh.edu.pl
Email: gjn@agh.edu.pl

***Abstract*—In this paper we introduce a new method to incorporate the user as an additional information source for the purpose of indoor localization. Therefore, the user is interrogated about certain characteristics in his/her environment. The questions are generated by a knowledge-based system built on ontologies. We provide a new statistical model to evaluate the user's answer and integrate it into the particle filter, which is used for the localization system. Our results show that the user's feedback significantly improves localization results.**

I. INTRODUCTION

The field of pattern recognition has achieved significant progress over the last decades, however there are still tasks that computers cannot accomplish. Indoor localization is a typical example for this, even though it is considered to be a key enabler to a large number of new applications in ubiquitous computing. Although smartphones are equipped with a lot of sensors like gyroscopes and accelerometers and the additional availability of external infrastructure like Wifi, it is still not possible to estimate the position of a user with a constantly high indoor accuracy.

Though, one information source for localization, which is generally overlooked, is the user himself. Humans are still superior to computers regarding tasks like object detection and description. Thus, we make use of this "sensor" by incorporating the user into the indoor localization process. We find the advantages to be as follows: Firstly, the system can ask the user detailed questions which he/she can answer easily while this task would be hard to accomplish for a machine. Secondly, interrogating the user is much more efficient in terms of energy and calculation time than letting the machine execute this work. Especially in mobile computing this is an important requirement, since battery lifetime is a precious good. Of course we are aware of the fact that an ubiquitous system should be as unintrusive as possible. This is why our proposed method will only ask the user for help if it becomes uncertain about its current position.

The most important scientific contribution of this paper is the way we incorporate the user as an additional information source into the localization system. To achieve this we developed a knowledge-based system which inquires the user about possible objects in his surrounding environment by using semantic description of maps and domain ontology. This clearly separates our approach from other methods that incorporate user feedback, but make the assumption that the user has knowledge about his/her current position. To be as unintrusive as possible we only trigger a question if the systems reaches a certain level of uncertainty. Therefore we introduce a measurement for the degree of uncertainty of the current estimation. For the implementation of the particle filter we provide detailed probabilistic models of the state transition and the state evaluation, whereby we put our focus on the modeling of the user's feedback.

The paper is structured as follows. In section II we review recent work on the incorporation of user feedback into the localization process. In section III we describe the semantic mediation system. Section IV explains all components of the probabilistic localization system as well as the step and turn detection process. In section V we present experimental results of our system and finally in section VI we draw conclusions and inspect future work.

II. RELATED WORK

In the field of indoor localization, a considerable number of publications has been made over the last decade. Despite this effort, currently no method has reached a level that makes it practicable for daily use. Either the long-term accuracy is inadequate or the installing and maintenance of additional infrastructure is too time and cost consuming. The former affects dead-reckoning systems, which avoid the use of additional infrastructure. These systems track a person relative to a starting position, thereby relying on step detection and heading information [1], [2], [3], [4]. However, estimation errors accumulate and lead to false position results over time.

One of the most promising indoor localization systems in terms of long-term positional accuracy is based on Wifi signal strength measurements [5]. Setting up such a system requires the creation of a so called *radio map* which associates a location with a Wifi fingerprint. Therein a fingerprint is the combination of the received signal strengths (RSSs) of all available wireless Access Points (APs) in this position. During localization the system compares the latest fingerprint measured by the user's device to the entries in the database. The most similar entry corresponds to the user's probable position. Although this localization method delivers good results, taking measurements all over the building in order to create the radio map is tremendously time consuming. Additionally, infrastructural changes like renovations enforce the renewal of the radio map.

Several attempts have been made to utilize users' feedback to avoid or at least reduce the effort of this aggravating procedure. Whenever the system provides a wrong position estimation or cannot provide one at all, the user is asked to fix this estimation by pointing to his/her location on a map [6], [7], [8], [9], [10]. This user-provided position is then connected with the current fingerprint. To encounter (willfully) wrongly contributed position corrections, some authors also introduce a confidence model for user information. It weights information given by highly trusted users higher than that of unreliable users [8]. Nevertheless, the drawback of these approaches for localization systems seems to be the absurdity of requiring the user to manually provide his/her position, since we cannot assume he/she has any knowledge about the current position.

In [11] the authors suggest incorporating user's feedback for the purpose of giving locations and traces semantically meaningful names. A place is considered interesting if the user stays there for a prolonged period of time. In this case the system asks the user to label it. Whenever the user returns to this location, the system notices this with the help of Wifi fingerprints and records it accordingly.

The authors in [12] provide a theoretical framework for the incorporation of user feedback, called human-assisted graph search (HumanGS). Interrogating the user is formalized as a graph search problem, where a directed acyclic graph represents the semantic information about the subject in question. However, their method mainly focuses on tasks in which it is impractical to generate questions in real-time, based on previously given answers like in crowd-sourced human-intelligence tasks. Consequently, to achieve the given task, HumanGS not only picks a single question, but a best set of questions. Yet in our case the choice of questions should depend on the localization's current level of certainty and the estimated position, making it inappropriate to choose a set of questions in indoor localization beforehand since both conditions may change with every new sensor measurement.

In literature, all of the aforementioned methods which allow modeling knowledge interchange between the system and the external components are known as mediation techniques [13]. The concept of mediation was first introduced in 1992 in the domain of databases to cope with the integration of knowledge from heterogeneous sources [14]. Over time, the notion of mediation has changed and nowadays the expression refers to methods for modeling dialogue between the system components and the user. The primary objective of these mechanisms is to query mediators (i.e. users) for information that can be later fused to resolve ambiguity or to improve system accuracy. Although the idea of incorporating user feedback is not new, all of the current locational systems require the user to know his/her position. In our work we do not make any assumptions about this. Instead, we question the user about characteristic features within the area in order to exclude those that do not match the information provided by the user.

III. SEMANTIC MEDIATION FOR LOCALIZATION

To improve localization we aim at including the user's information. We want to make the process least intrusive for him and at the same time as efficient as possible. In this section we propose a knowledge-based mediation system tightly integrated with the main localization module to obtain such a solution. The main objective of this system is to get on demand information from the user which makes the localization process unambiguous.

The system consists of two main parts: the static and the dynamic knowledge component. The static knowledge component provides a semantic representation of the microloclization maps. This representation allows us to communicate with the user while employing easy-to-grasp concepts. The representation itself is dual: the microlocalization map is first created with OpenJUMP[1] software, and is then serialized to a well known GML[2] format. The GML map is next compiled to the form of a semantic map with the use of a dedicated ontology generator. For simplicity, the ontology itself is encoded using logic programming language Prolog. It provides domain knowledge about the user's environment. In fact, it is an input for the dynamic knowledge component which is responsible for question generation, where questions apply concepts from the ontology. Furthermore, the dynamic knowledge component allows an online mediation between the user and the sensor fusion system. In detail, the following section describes these two components.

A. Maps representation and knowledge compilation

In order to represent the knowledge about the floor plan and its features in a semantic and structuralised way, the building plan is first created using OpenJUMP software; the output of this process then is serialized in a GML format. There are at least four advantages of this representation: 1.) it is a XML based format, therefore it is easy to process using the vast choice of supported libraries and technologies; 2.) as a default format for GIS software, it is perfectly suited to represent the spatial data; 3.) there are already existing solutions to create and manipulate models in this format; 4) it does not put constraints on possible features of objects – in fact there are even supported schemas to validate data for specific domains.

```
class(place).
class(classroom).
class(desk).

attributes(place, [
    size: [small, medium, big],
]).
relationships(classroom, [
    has: [desk]
]).

subclass(classroom, place).

relation(has).
relation(consists_of).
properties(has, [
    distributes_over: consists_of
]).
properties(consists_of, [
    transitivity: true
]).
```

Fig. 1. Simplified TBox from the indoor localization ontology.

---

[1] OpenJUMP is an open source Geographic Information System (GIS) written in the Java programming language.

[2] Geography Markup Language, see http://www.opengeospatial.org/standards/gml

In the next step the GML representation has to be compiled into a knowledge base. Translation is done according to the semantic meta-information contained in a terminological part of the ontology (so called TBox). Finally, every GML entity should correspond to an instance inside of an assertional part of the ontology (so called ABox). While there is a vast choice of existing technologies to describe the ontological systems, due to the specific inferential tasks, we propose a dedicated representation based on the logic programming language Prolog (specifically on a dialect supported by SWI-Prolog[3] development version). Samples of this notation are presented in Figures 1 and 2. The main features of the proposed solution include: 1.) human-friendly format; 2.) support for conjunctive queries; 3.) enhanceability using the Prolog rules. Despite being strictly less expressive than the popular OWL[4] family of languages, it is sufficient to represent semantically annotated unambiguous data[5]. In the TBox part there are six types of statements supported: definitions of classes; attributes and relationships necessary for every instance of the class; subclassing relationships between the defined classes; definitions of relations; properties holding the defined relations. An example of the TBox can be found in Figure 1.

```
classroom{id: lab316, size: big} has [
    desk{color:brown} has [
        pc{},
        display{type:flat}
    ] * 16,
    blackboard{id: smart316},
    more
].

blackboard{id: smart316} stands_near [
    tv{display:flat},
    window{}
].
```

Fig. 2. Simplified ABox from the indoor localization ontology.

An ABox part consists of the ontology instances — every statement represents a one-to-many relation between specific instances (currently, to assert a mere existence of an object it has to stand in an unary relation like `exists`). Every instance is described with a set of attributes listed inside the curly braces similarly to the popular JSON notation. Among the attributes there is a reserved unique id, explicitly stated in a model or implicitly generated during model loading; explicitly stated ids can be used to reference the same instance in different statements. By default every relation is assumed to be exhaustive; to suppress this behavior the keyword `more` has to be added to the list of related objects. A multiplication sign can be used to state a number of the qualitatively same statements in an intuitive manner. An example of the ABox is presented in Figure 2.

The process of training data generation from the ontology model consists of two steps; first the model is compiled into a more basic representation, consisting of the two types of statements: 1.) class assertions assigning the instance with its plain attributes to the specified class; 2.) relation assertions between instances referenced with `id` value, for example: `has(lab316, smart316)`. Relations' properties (like transitivity) are represented by the appropriate Prolog rules. The main motivation behind the translation to this lower level representation is to allow fine grained semantic queries. During this phase, a TBox is also used to validate ABox instances, infer information about attributes, and fill missing attribute values with a reserved `unspecified` value. Afterwards, the training data set is generated based on the alternatives received from a Sensor Fusion Mechanism. Every instance related (directly or indirectly) to an alternative becomes a new training feature with a name created by concatenation of classes' and relations' names leading from alternative's class to this instance. The same applies to their plain attributes. For example, training features based on the `classroom{id:316}` alternative from Figure 2 include: `size`, `has_desk`, `has_desk_has_pc`, `has_desk_color`, etc. Training features are then grouped into entries conforming to the following requirements: 1.) there is one training entry with all the plain attributes of the alternative (for example `{size}`); 2.) other training entries contain a training feature representing an instance related directly to the alternative and its descendants. The resulting training set consists of training entries generated for every alternative and tree structure representing relations between training features. This is an input for the dynamic knowledge component described next.

B. Mediating user localisation

The dynamic knowledge component is responsible for the discovery of the most probable user location in a process of mediation. It is based on the information acquired by the user answering the questions from a structure called question forest. The question forest is generated based on four types of information: 1.) Possible user positions P that are obtained from the sensor fusion mechanism; 2.) Semantic map M, with all the information about the characteristics of the user environment; 3.) Set of facts F about the user location obtained from the user himself in a process of mediation (this is an empty set in the initial state) and 4.) Set of question roots R which are defined as independent objects or features of every room from P. Having information encapsulated in P and M, a training set T is generated. The training set T and question roots R are the input for the question forest generation algorithm. The algorithm takes every question root from R, and generates a decision tree for it. The decision tree algorithm is loosely based on ID3 algorithm. It uses information gain as a criterion for splitting nodes into branches. However, in our approach the original information gain formula is enriched with an answer acquisition cost factor assigned to every attribute in training set T. This factor allows us to choose the node that not only best splits the dataset but also minimizes the cost of obtaining an answer to the question. The cost-sensitive information gain formula for an attribute A and a training set T is defined as follows:

$$Gain(A) = \frac{H(T) - \sum_{v \in Domain(A)} \frac{|T_v|}{|T|} H(T_v)}{cost(A)} \quad , \qquad (1)$$

---

[3] SWI-Prolog is a free, open-source Prolog implementation, see: http://www.swi-prolog.org/

[4] Web Ontology Language (OWL) is a family of languages designed to represent ontologies.

[5] Every instance has an a priori known class, so there is no need for the inferential capabilities of OWL regarding classification of data.

where T_v is a subset of T, such that for every $t \in T$ the value of $A = v$, and $cost(A)$ is a cost associated with obtaining the value of A from the user. The entropy for the training set T is defined as follows:

$$H(T) = -\sum_{x \in X} p(x) \log_2 p(x) \quad , \tag{2}$$

where X is a set of all classes in T and $p(x)$ is a ratio of the number of elements of class x to all the elements in T.

Every tree that is built with the aforementioned formulae also has an aggregated cost assigned, indicating the degree in which this tree allows us to correctly classify a room where the user is located. An example of such a tree is presented in Figure 3. There are four different types of elements that take part in the process of cost calculation:

1) Best Prediction factor – it is a maximum probability value from the leaf that best classifies the data.
2) Mean Prediction factor – it represents an arithmetic mean of all the maximum probabilities from every leaf of the tree.
3) Question cost – it represents the total cost of obtaining all the values of the attributes from the tree nodes.
4) Aggregated cost – this is the *Question cost* divided either by the *Mean prediction factor*, or *Best prediction factor*. The choice is arbitrary and should be made based on the experiments.

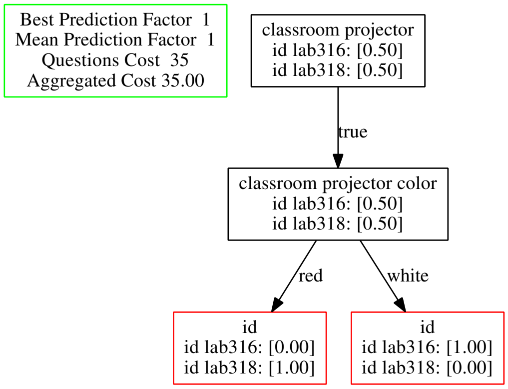

Fig. 3. Example of a decision tree from a question forest

All this information is used by the mediation procedure. The mediation itself is an iterative process and is presented in Algorithm 1. The algorithm begins with the generation of a training set, based on P, F and M. The training set is generated according to the procedure described in Section III-A. After the training set is generated, the question roots are obtained. Question roots are the questions which, from a logical point of view, should be asked first in the mediation process. For instance, a question about the existence of a TV in the room should precede the question about the TV's color. In such a case, the question about the existence of a TV is a question root. Based on the question roots a question forest QF is created. After that, the question A, for which the aggregated cost is minimal, is selected from the QF. The procedure continues picking question trees from QF until the optimal classification provided by A is better than provided by C.

```
Data: $P$ – set of all possible user positions, equally
      probable
      $M$ – ontology-based semantic map
Result: $C$ – set of all possible user positions with
        certainties assigned
1  $F = \phi$;
2  $C = P$;
3  $T = generateTrainingSet(P, F, M)$;
4  $R = obtainQuestionRoots(P, M)$;
5  $QF = generateQuestionForest(T, R)$;
6  if not empty $QF$ then
7      pop $A$ from $QF$, such that:
       $\forall X \in QF : cost(A) \leq cost(X)$;
8      assign $V = root(A)$;
9  else
10     return $C$;
11 end
12 if $optimalClassification(A)$ not better than $C$ then
13     goto 6;
14 end
15 while $V$ is not leaf(A) do
16     ask the user question associated with $V$;
17     save user answer $r$ as a fact in $F$;
18     locate the edge $(V, r, N)$ in question tree $A$;
19     assign $V = N$;
20     if $classification(V)$ better than $C$ then
21         assign $C = classification(V)$;
22     end
23 end
24 if $classificationQuality(C) \geq \epsilon$ then
25     return $C$;
26 else
27     goto 3;
28 end
```

Algorithm 1: Algorithm for mediating user position

If all of the questions from QF provide worse classification then C, C is returned and the algorithm stops. The optimal classification is defined as the probability associated with the best classifying branch in the tree. For example the optimal classification equals 1.0 for the tree presented in Figure 3, as this is the certainty of the classification provided by the best classifying leaf (in this case, both leaves are optimal). If the optimal classification provided by A is better than the one provided by C, the interrogation of the user is initiated. Every obtained answer is saved in F and is later used to generate a better, narrowed training set. Finally, when the user answers question A, and the certainty of the classification is sufficient with respect to some constant ϵ, C is returned. Alternatively, the algorithm goes back to line 3 and the mediation starts over.

IV. INDOOR LOCALIZATION

Next we explain our indoor localization system. We model the indoor localization problem as a recursive density estimation problem, which can be estimated by employing the Bayesian framework, namely particle filters. Because particle filters are extensively discussed in many papers, we forgo a detailed review and refer the reader to [15], [16], [17]. In the following we will provide the necessary statistical models.

A. Localization Model

Indoor localization can be treated as a recursive density estimation problem, where the goal is to estimate the posterior probability density $p(\boldsymbol{q}_t \mid \langle \boldsymbol{o} \rangle_t)$ of a hidden variable $\boldsymbol{q}_t$ by successively incorporating new observations $\boldsymbol{o}_t$. We denote the series of observations from the beginning up until t as $\langle \boldsymbol{o} \rangle_t = \boldsymbol{o}_1, \boldsymbol{o}_2, ..., \boldsymbol{o}_t$. In this work the hidden state $\boldsymbol{q}_t$ at time t consists of a two-dimensional position x_t, y_t and the user's heading α_t,

$$\boldsymbol{q}_t = (x_t, y_t, \alpha_t)^T \quad . \tag{3}$$

Our observation is given by

$$\boldsymbol{o}_t = (d_{\text{obs}}, \Delta\alpha_{\text{obs}}, \phi_{\text{obs}})^T \tag{4}$$

and consists of the observed covered distance d_{obs}, the heading change $\Delta\alpha_{\text{obs}}$ and the user feedback ϕ_{obs}.

The posterior $p(\boldsymbol{q}_t \mid \langle \boldsymbol{o} \rangle_t)$ can be calculated recursively using

$$p(\boldsymbol{q}_t \mid \langle \boldsymbol{o} \rangle_t) = \underbrace{p(\boldsymbol{o}_t \mid \boldsymbol{q}_t)}_{\text{observation}} \int \underbrace{p(\boldsymbol{q}_t \mid \boldsymbol{q}_{t-1})}_{\text{transition}} \underbrace{p(\boldsymbol{q}_{t-1} \mid \langle \boldsymbol{o} \rangle_{t-1}) d\boldsymbol{q}_{t-1}}_{\text{recursion}} . \tag{5}$$

The *observation* component evaluates the probability of a hypothetical state, if we take the current sensor data into account. The *transition* models the possibilities of moving from one state to another. The *recursion* is the posterior probability of the previous time step and thus contains all information up to time $t-1$. Since (5) cannot be calculated in an analytically closed form, particle filters estimate (5) with the help of N weighted samples. In the following, we will illustrate the *transition* and *observation* models in detail. For our starting distribution p_0 we assume the initial position and heading to be known.

B. Transition model

For the purpose of localization the state transition model estimates the probability of a pedestrian moving from one position to another within two points in time. Such a model must incorporate physical limitations like the speed of walking and obstacles impeding the path. Since we rely on the CONDENSATION algorithm [18] our transition does not incorporate any observations. Consequently, we must take into consideration every possible movement, whether the pedestrian has taken a step or he/she has changed his/her direction.

Our transition model is defined as composite of independent terms

$$p(\boldsymbol{q}_t \mid \boldsymbol{q}_{t-1}) = p(\boldsymbol{q}_t \mid \boldsymbol{q}_{t-1})_{\text{head}}\, p(\boldsymbol{q}_t \mid \boldsymbol{q}_{t-1})_{\text{dist}} \tag{6}$$

where

$$p(\boldsymbol{q}_t \mid \boldsymbol{q}_{t-1})_{\text{head}} = \mathcal{U}(-180°, 180°) \tag{7}$$

is the probability for a heading change, which is uniform between all possible values because a pedestrian can turn around rapidly in every possible direction. It must be noted that we introduce this term for the sake of completeness, since a uniform distribution over the whole state space does not provide new information. However, one can introduce a different distribution if better assumptions about the pedestrian's movement can be made.

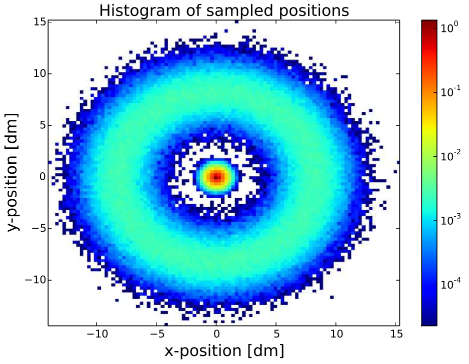

Fig. 4. Example of sampling our state transition model. Particles that are assumed to stand still are generating the inner circle, while "walking" particles generate the outer circle.

The state is updated by

$$\begin{aligned} \alpha_t &\sim \mathcal{U}(-180°, 180°) && (8)\\ x_t &= x_{t-1} + d_t \cos \alpha_t && (9)\\ y_t &= y_{t-1} + d_t \sin \alpha_t \quad , && (10) \end{aligned}$$

where d_t is randomly chosen from a Gaussian mixture

$$d_t \sim a_w \mathcal{N}(\mu_w, \sigma_w^2) + a_s \mathcal{N}(\mu_s, \sigma_s^2) \quad , \tag{11}$$

which describes the probability of a step length depending on whether a step is assumed or not. Thereby the parameters a_w, μ_w and σ_w refer to the assumption that the pedestrian was walking, while a_s, μ_s and σ_s refer to the assumption that the pedestrian was standing, respectively. Since we do not have any further information we can safely assume $a_w = a_s = 0.5$. μ_w represents the step length when walking with variance σ_w^2 and μ_s the step length when standing, which typically should be zero, with variance σ_s^2. Figure 4 shows an exemplary sampling of our state transition model.

C. Observation model

The observation model weights every state hypothesis given the current sensor measurements and is split up in three parts:

$$\begin{aligned} &p(\boldsymbol{o}_t \mid \boldsymbol{q}_t, \boldsymbol{q}_{t-1}) = \\ &p(\boldsymbol{o}_t \mid \boldsymbol{q}_t, \boldsymbol{q}_{t-1})_{\text{dist}}\, p(\boldsymbol{o}_t \mid \boldsymbol{q}_t, \boldsymbol{q}_{t-1})_{\text{head}}\, p(\boldsymbol{o}_t \mid \boldsymbol{q}_t)_{\text{user}} \quad . \end{aligned} \tag{12}$$

Here $p(\boldsymbol{o}_t \mid \boldsymbol{q}_t, \boldsymbol{q}_{t-1})_{\text{dist}}$ evaluates if the new state's covered distance corresponds to the step detection measurement and $p(\boldsymbol{o}_t \mid \boldsymbol{q}_t, \boldsymbol{q}_{t-1})_{\text{head}}$ if the heading change is according to the estimation of the turn detection. The positional feedback of the user is incorporated into $p(\boldsymbol{o}_t \mid \boldsymbol{q}_t)_{\text{user}}$ and evaluates the current position given by x_t, y_t. Note that we deviate from the convenient observation model that only utilizes the current state $\boldsymbol{q}_t$ by also making use of the previous state $\boldsymbol{q}_{t-1}$. The additional information of the previous state is needed for the calculation of the covered distance and the directional change. We assume all three observation components to be statistically independent.

Picture?	Answer	Position information
✓	Yes	**User in room with red picture**
✗	No	**User in room without red picture**
✓	No	No information about position
✗	Yes	No information about position
✚	Yes	No information about position
✚	No	No information about position

TABLE I. POSSIBLE OUTCOMES TO THE QUESTION: "CAN YOU SEE A RED PICTURE IN THE ROOM?". ✓DENOTES THAT THERE IS A RED PICTURE IN THE ROOM, ✗THAT THERE IS NO RED PICTURE IN THE ROOM AND ✚THAT THE QUESTION ASSUMES WRONG FACTS.

1) Distance evaluation: The distance evaluation is based upon a step detection process and floor map information and is given by:

$$p(\boldsymbol{o}_t \mid \boldsymbol{q}_t, \boldsymbol{q}_{t-1})_{\text{dist}} = \mathcal{N}(f_{\text{map}}(\boldsymbol{q}_t, \boldsymbol{q}_{t-1}) \mid d_{\text{obs}}, \sigma^2_{\text{step,obs}}) \quad . \tag{13}$$

If a step was detected in the last time interval, d_{obs} is set to the estimated step length, otherwise to zero. $\sigma^2_{\text{step,obs}}$ should be selected according to the certainty of the step length estimation process. $f_{\text{map}}(\cdot)$ is a function that uses the floor map information and returns either the covered distance d_t or infinity, if during the sample transition a wall was crossed.

$$f_{\text{map}}(\boldsymbol{q}_t, \boldsymbol{q}_{t-1}) = \begin{cases} d_t = \sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2} \\ \infty, \text{if wall crossed} \end{cases} , \tag{14}$$

where x_t and y_t are the location coordinates given by the state vector $\boldsymbol{q}_t$, and x_{t-1}, y_{t-1} are given by $\boldsymbol{q}_{t-1}$.

In localization, the distance measurement $f_{\text{map}}(\cdot)$ should integrate the knowledge that people must bypass obstacles like walls, which makes the Euclidean usually the wrong choice of distance measurement. However, by keeping the filtering update intervals short enough (here 250ms) the assumption holds that a pedestrian can only walk a maximum of one step within this interval. In addition, we assume that people don't change their heading within one step, which leads to a piecewise linearization of the walking trace and, consequently the Euclidean to be the correct choice.

2) Heading evaluation: The heading change is evaluated by

$$p(\boldsymbol{o}_t \mid \boldsymbol{q}_t, \boldsymbol{q}_{t-1})_{\text{head}} = \mathcal{N}(\Delta\alpha_t \mid \Delta\alpha_{\text{obs}}, \sigma^2_{\text{head,obs}}) \quad , \tag{15}$$

where $\Delta\alpha_t$ is the difference between the current and previous state heading

$$\Delta\alpha_t = \alpha_t - \alpha_{t-1} \quad . \tag{16}$$

$\Delta\alpha_{\text{obs}}$ is the observed heading change, which is estimated by a turn detection process (see section IV-D1). We rely on this process instead of the smartphone's magnetometer because magnetometer readings within buildings are heavily disturbed by metallic objects and other electric devices.

3) User feedback evaluation: The localization system questions the user whenever it reaches a degree of uncertainty (we elaborate on this point in section IV-E2). The mediation system returns a probability for different rooms depending on the answer of the user.

Table I shows the different outcomes and positional information we gain if we ask the user the exemplary question "Can you see a red picture in the room?". Thereby we summarize table I into three different cases:

1) The user's answer corresponds to the fact he is asked about, meaning that if there is a red picture in his present room he will answer "yes" and "no" if the red picture is missing. These are the only situations that provide additional information (line 1 and 2 of table I) and should occur the most often. The information our system can extract from this case is that the user's positions either is in a room with a red picture (line 1) or that he is in any room without a red picture (line 2).
2) The user's answer deviates from the true facts. This can happen e.g. if the user does not recognize the object he is asked about or he/she (intentionally) provides wrong information (line 3 and 4 of table I).
3) The third case considers the situation where the localization system itself provides wrong information. This might happen if the system's information is outdated and the specific object is no longer in the room, e.g. somebody removed the red picture without updating the system (line 5 and 6 of table I).

Since it is not possible for the system to identify situations 2) and 3) automatically, we treat these cases as sensor noise.

To formalize this model we introduce the sets $\mathcal{M}$, $\mathcal{A}$ and $\mathcal{B}$. Therein $\mathcal{M}$ denotes the set of all possible locations (x_t, y_t), $\mathcal{A}$ is the set of all locations belonging to one or more specific rooms and $\mathcal{B} = \mathcal{M} \setminus \mathcal{A}$ is the set of all locations that do not belong to the room specified by $\mathcal{A}$. With this information our model of the user's feedback can be written as

$$\begin{aligned} &p(\boldsymbol{o}_t \mid \boldsymbol{q}_t)_{\text{user}} = \\ &\underbrace{k_1(m_1\mathcal{U}(\mathcal{A}) + m_2\mathcal{U}(\mathcal{B}))}_{\text{Position information}} + \underbrace{(k_2 + k_3)\mathcal{U}(\mathcal{M})}_{\text{Noise}} \quad . \end{aligned} \tag{17}$$

Equation (17) is a mixture model with the weights k_1, k_2, k_3 that correspond to the above described cases. The term $m_1\mathcal{U}(\mathcal{A}) + m_2\mathcal{U}(\mathcal{B})$ itself is a mixture model and is described by case 1). If the user's answers reduces possible locations to some specific rooms, we have a uniform distribution over these rooms ($m_1 = 1$), otherwise we can assume that he/she is at any other location different to the specified rooms ($m_2 = 1$). Since cases 2) and 3) do not provide any information to the localization process, the noise part of (17) is determined by the uniform distribution $\mathcal{U}(\mathcal{M})$ over all possible locations on the map. Determining exact weights for k_1, k_2, k_3 is not an easy task, but the methods referred to in section II which build a model of the user's trustworthiness could provide a solution to this. Nevertheless, k_1 should outweigh the other weights significantly to achieve good results.

Also note that we have to choose a uniform distribution because our knowledge is limited to which room the user is located without any additional details. One can select a different distribution, like a Gaussian centered at a determined location, if the answer to a question provides more specific information. This will be part of our future research.

D. Step and turn detection

In the following, we shortly explain the methods to estimate steps and heading changes.

1) Turn estimation: At this stage of our research we assume that the user holds the smartphone in his/her hand and is looking at it, since this is the most common way to hold your phone while using it as a guide. We make use of this assumption by calculating the integral over the values $\theta_{z,t}$ of the gyroscope's z-axis for a given time interval Δt,

$$\Delta\alpha_{\text{obs},t} = \int_{t-\Delta t}^{t} \theta_{z,t} dt \ . \tag{18}$$

2) Step detection: We use a step detection process to estimate the covered distance of the pedestrian. In [19] different step detection methods are evaluated, of which we chose a threshold based one, because it provides the best performance with minimal calculation effort.

E. State estimation and divergence measurement

Next we will answer two important questions. Firstly, given the estimated probability density $p(\boldsymbol{q}_t \mid \langle \boldsymbol{o} \rangle_t)$, how do we extract the current position of the user? Secondly, when do we trigger the system to interrogate the user about his/her position? We will start to elaborate on the first question, since this also leads to the answer of the second question.

1) State estimation: Since the result of the particle filter is a probability density over the whole state space we must infer the position from this density. For this, we fit the particle set to a Gaussian distribution, whereby the mean value represents the estimated position. However, such an estimation suffers heavily when the estimated density is multimodal, because the position estimation will be somewhere in the middle between the modes. We can solve this issue with two approaches. The first one is simply to wait. If the user keeps walking, multimodalities often vanish because of map information. That is, some walking routes become impossible because walls would be crossed. The second approach utilizes the mediation of the user. If the probability density possess two or more peaks, this is a typical sign that the localization system got lost.

2) Divergence measurement: Multimodalities in the density can be seen in localization if the particle set is spread in two or more clusters. Making use of this observation we cluster the particle set with a Gaussian mixture model (GMM) consisting of two Gaussians. The GMM estimates the means, the covariances and the weights of both Gaussians. Typically, if no ambiguities exist the Gaussian's mean values will be close together distance-wise. However, we can identify ambiguities as soon as both mean values diverge.

A decision whether the user should be interrogated or not should be made wisely, and should be performed in a way that is least intrusive for him. For that, a measure of uncertainty of the current location estimation was introduced that allows to trigger mediation only when needed. We define this measure in a form of two conditions that both have to be true in order to start the questioning process. The necessary condition is that both of the weights assigned to Gaussian mean values are smaller than a defined constant $\epsilon_{\text{weight}} \in (0.5; 1)$. The sufficient condition was defined in a form of two alternative options, with one of them to choose. These are: *centroid based option* and *density based option*. Both of them are evaluated based on the map information. In *centroid based option* mediation is triggered when the line connecting two centroids intersects more than one room. In *density based option* points for which density of the Gaussian estimates is greater than some defined constant $\epsilon_{\text{density}} \in (0; 1)$ are calculated first. Then, convex hulls created for these points are projected on the map. Finally, mediation is triggered when projected convex hulls intersect more than one room. The choice of the sufficient condition option is made arbitrarily, and should depend on the architecture of the area where the localisation shall be performed.

V. EXPERIMENTAL RESULTS

In the following section we will evaluate the localization system and its improvement by the user feedback. First of all, we provide experimental results regarding the accuracy of the localization system on its own. The results will show that, while the system can provide decent results, one typically needs additional information. Secondly, we contribute the results that the mediation improves the localization results. We collected all the data using a Nexus 5. The gyroscope data is rotated into the earth's coordinate system using Android's rotation matrix. For the accuracy estimation a test person walked 11 different routes through the university's building. For every route we recorded the timestamps at characteristic locations. Floor layout and the sample routes are presented in Figure 5.

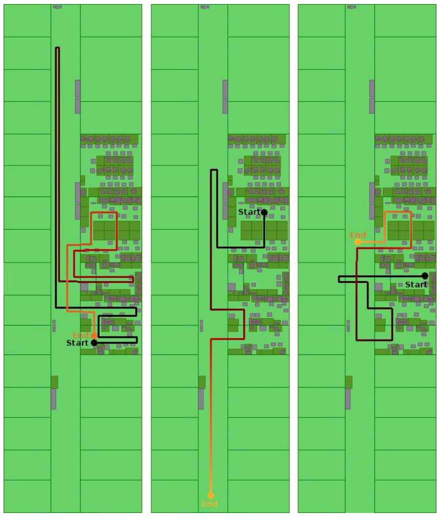

Fig. 5. Floor layout of the experiment environment with three sample routes.

A. Localization accuracy

Our first experiment evaluates the accuracy of the localization system. Therein we neglect any additional information given by the user. The initial starting position and heading are assumed to be known. We compare the room estimated by the system with the room provided by the ground truth data, since it was shown that room accuracy is enough for most applications [7]. Within all of the 11 routes the user visited 80 locations, whereby the system estimated 46 correctly, resulting in an accuracy of 57% using 10000 particles. This low number is not surprising considering the fact that the system makes no use of external information. Typical for such a dead reckoning

#Particles	Dead reckoning	Dead reckoning + User feedback	#Mediations
10000	57% (46/80)	67% (54/80)	21
50000	67% (54/80)	87% (70/80)	36

TABLE II. COMPARISON OF ROOM ACCURACY USING ONLY DEAD RECKONING COMPARED TO DEAD RECKONING + USER FEEDBACK

approach, position estimations are often accurate at an early stage of the localization, while the results become worse the longer the system is not corrected by external information. As expected, the results become much better if the number of particles is increased. Using 50000 particles the results improve to 67%.

B. Accuracy improvement by user feedback

Next, we demonstrate that the incorporation of the user's feedback improves the overall system accuracy. We tested the system setting the feedback model's weighting factors to $k_1 = 0.9$ and, $k_2 = k_3 = 0.05$. As seen in table II, the results are significantly improved by the user feedback. The user was interrogated 21 times during all of the 11 runs using 10000 particles and 36 times using 50000 particles. However, using only 10000 particles the results are still not as promising as expected. This is due to the mediation system being unable to recognize the need to interrogate the user about his/her position if particles are not spread across different rooms. These situations occur, whenever particles are gathering within a single room, so that consequently, we only have a single cluster. However, this cluster may not even be located in the correct room. Such situations can be avoided by increasing the number of particles. Nevertheless, one drawback of this is that number of mediations also goes up, since particles then might be spread around a lot of locations. Figure 6 shows a situation when a mediation will be triggered because the system is unsure of its position.

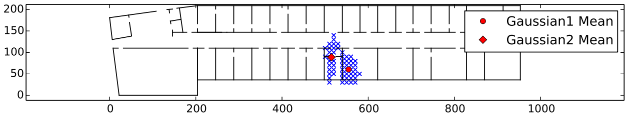

Fig. 6. Example of a situation when a mediation is triggered, because the means of both Gaussians are at two different rooms. Particles are shown as blue crosses.

VI. DISCUSSION AND CONCLUSION

In this paper we presented a localization system that incorporates the user by asking him/her about certain characteristics of the room he/she currently finds him-/herself in. For this we provided a semantic knowledge based system that chooses an appropriate question given the current density estimation of the localization system. For the particle filter based localization system we introduce a statistical model to evaluate the user's answer and integrate it as additional sensor observation. We are aware that a localization system should be as unintrusive as possible, so "asking the user" might be counterintuitive at first sight. Nevertheless, our approach can be combined with techniques for automatic fingerprinting, as mentioned in the related work section.

ACKNOWLEDGMENT

This work was partially funded by the German Research Foundation (DFG) as part of the research training group GRK 1564 "Imaging New Modalities" and by the National Science Centre, Poland as a part of the KnowMe project (registration number 2014/13/N/ST6/01786).

REFERENCES

[1] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao, "A Reliable and Accurate Indoor Localization Method Using Phone Inertial Sensors," in *UbiComp '12*, Sept. 2012, pp. 421–430.

[2] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: Zero-effort Crowdsourcing for Indoor Localization," in *Mobicom '12*, Aug. 2012, pp. 293–304.

[3] K. Lan and W.-Y. Shih, "Using Smart-Phones and Floor Plans for Indoor Location Tracking," *Human-Machine Systems, IEEE Trans. on*, vol. 44, no. 2, pp. 211–221, April 2014.

[4] L. Köping, M. Grzegorzek, and F. Deinzer, "Probabilistic Step and Turn Detection in Indoor Localization," in *Data Fusion and Target Tracking 2014*, Liverpool, UK, April 2014, pp. 1–7.

[5] M. Abdat, T.-C. Wan, and S. Supramaniam, "Survey on Indoor Wireless Positioning Techniques: Towards Adaptive Systems," *Int. Conf. on Distributed Framework and Applications 2010*, pp. 1–5, Aug. 2010.

[6] E. Bhasker, S. Brown, and W. G. Griswold, "Employing user feedback for fast, accurate, low-maintenance geolocationing," in *PerCom 2004.*, March 2004, pp. 111–120.

[7] P. Bolliger, "Redpin - Adaptive, Zero-configuration Indoor Localization Through User Collaboration," in *Workshop on Mobile Entity Localization and Tracking in GPS-less Environments*, Sept. 2008, pp. 55–60.

[8] A. K. M. Mahtab Hossain, H. Nguyen Van, and W.-S. Soh, "Utilization of User Feedback in Indoor Positioning System," *Pervasive and Mobile Computing*, vol. 6, no. 4, pp. 467–481, August 2010.

[9] Y. Luo, Y. Chen, and O. Hoeber, "Wi-Fi-Based Indoor Positioning Using Human-Centric Collaborative Feedback," in *Int. Conf. on Communications (ICC 2011)*, June 2011, pp. 1–6.

[10] L. T. Nguyen and J. Zhang, "Wi-Fi Fingerprinting Through Active Learning Using Smartphones," in *UbiComp '13 Adjunct*, Sept. 2013, pp. 969–976.

[11] D. H. Kim, K. Han, and D. Estrin, "Employing User Feedback for Semantic Location Services," in *UbiComp '11*, Sept. 2011, pp. 217–226.

[12] A. Parameswaran, A. D. Sarma, H. Garcia-Molina, N. Polyzotis, and J. Widom, "Human-assisted graph search: It's okay to ask questions," *VLDB Endowment*, vol. 4, no. 5, pp. 267–278, Feb. 2011.

[13] A. K. Dey and J. Mankoff, "Designing mediation for context-aware applications," *ACM Trans. Comput.-Hum. Interact.*, vol. 12, no. 1, pp. 53–80, Mar. 2005.

[14] G. Wiederhold, "Mediators in the architecture of future information systems," *Computer*, vol. 25, no. 3, pp. 38–49, 1992.

[15] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking," *IEEE Trans. on Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[16] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. Springer, 2001, ch. An Introduction to Sequential Monte Carlo Methods.

[17] A. Doucet and A. M. Johansen, "A Tutorial on Particle Filtering and Smoothing: Fifteen years later," in *The Oxford Handbook of Nonlinear Filtering*, D. Crisan and B. Rozovsky, Eds. Oxford University Press, 2011.

[18] M. Isard and A. Blake, "CONDENSATION - Conditional Density Propagation for Visual Tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, August 1998.

[19] A. Brajdic and R. Harle, "Walk detection and step counting on unconstrained smartphones," in *UbiComp '13*, 2013, pp. 225–234.