A Context-enhanced Sector-based Indoor Positioning Library

Alexander Stelzhammer and Jens Krösche

University of Applied Sciences Upper Austria
Faculty for Informatics, Communications and Media
Department of Mobility and Energy
Softwarepark 11
4232 Hagenberg im Mühlkreis, Austria
Email: firstname.lastname@fh-hagenberg.at

***Abstract*—The position context is still one of the key features when it comes to building context-aware mobile applications. In the outdoors, today's GNSS provide the necessary information without much problem to all interested users but indoors reliable data and systems are still a big problem. The presented approach provides a novel approach to evaluate the position of an entity in such an indoor environment. It trades accuracy for robustness and environmental flexibility by neglecting concrete position coordinates and concentrating on sector-based positioning.**

***Keywords*–sector-based position; context-aware; indoor position**

I. Introduction

With the global proliferation of smartphones, the chances to disseminate context-aware applications and initiate the ubiquitous computing era envisioned by Weiser [1] on a large scale are higher than ever. In the wake of this development, the knowledge of one's own position has become one of the most crucial factors. Even as pioneers of the context-aware computing domain like Abowd and Dey [2] or Schmidt et al. [3] discussed its significance, the position information was and still is one of the most used key features when it comes to building adaptable and mobile context-aware systems. Whereas context-aware systems should be considered as systems that not only take into account the direct input of its users, but also consider environmental and situational parameters to provide suitable services, matching information or adaptive behaviour. The provision of the needed position information in today's world is strictly separated between the outdoor and the indoor domain. In the outdoor domain, modern Global Navigation Satellite Systems (GNSS) like GPS, Galileo, GLONASS or Baidou provide a reliable positioning service. However, in the indoor domain, no industrial standard has been established even after decades of research and technological advances.

Approaches used by today's indoor positioning systems usually rely strongly on highly complex algorithms trying to deduce the users' positions facilitating intensive pre-measuring, modelling, training and/or computing power. Based on the assumption that a very accurate position information is not needed in many indoor scenarios the presented system tries to approach the problem domain from a different angle. It is assumed that a general information in which part of the building an entity resides in is a good enough position information. Therefore, a robust and adaptable (regarding the used indoor environment) library for mobile applications that provides a sector-based position information is presented in this paper.

Based on a simple descriptive file containing the positions and provided networks of a building's WiFi infrastructure, a building is divided into sectors. These sectors are later used to reference the position of an entity. The main input to the system are regular scans of the current WiFi environment (particularly BSSID and RSSI values). The decision which sector is the "target" sector is calculated based on a pipes and filters pattern [4] approach. The measured access point values together with historical measurements are filtered and prioritized by a set of parameterizable filters (s. Section III) until one or more sectors are identified as "target" sectors reflecting the position sector(s) of the entity. The filters themselves can be rather simple or sometimes more complex. They range from simple white list or threshold ones over plausibility checking graph filters to context regarding filters. Taking into account the positioning taxonomy of Küpper [5], the presented approach uses proximity sensing to deduce the position. Our library tries to address the problem of multiple signals being received simultaneously, and the strongest signal not always reflecting the best matching signal respectively position, by applying a set of filters and plausibility checks. Even if the provided system is not adaptable on its own, it uses context to improve its calculation results and provides the foundation for mobile adaptable systems targeting the indoor domain by providing a novel, robust, and reliable indoor positioning system.

The rest of the paper is structured as follows. We start with a short overview of the current developments in indoor positioning in Section II which we use to motivate our own novel approach in Section III. In order to test and evaluate the presented approach we developed a simulation environment which is presented in Section IV. Section V shows the first promising results of our positioning library in four different scenarios and we close with a short conclusion and an outlook on the next steps to improve our approach.

II. Related Work

As indoor positioning is still an ongoing topic for context-aware systems even after decades of research, many different approaches have been tried to address the problem. What started with the Active Badge system from Want et al. [6] and the PARKTAB system by Schilit et al. [7] only slowly gained momentum in the beginning. In today's world where nearly everyone is using mobile applications on smartphones and the Internet of Things (IoT) is reality, the big companies like Google and Apple try their best to support indoor positioning but current solutions are still unsatisfying.

A rather recent indoor localization system based on WLAN fingerprinting was Sectjunction presented by He and Chan [8].

 ISBN: 978-1-61208-781-8

The special aim of the project was to reduce measurement uncertainty by dividing the coverage area into sectors. Each sector corresponded to a distinct access point and had its size modelled according to its access point's signal strength. The location of a target could then be constrained to the overlap of those sectors with the strongest RSS value, thus tightening the search space without leading to a dispersed set of reference points. Another project was Redpin from Bollinger [9], which was also a fingerprint-based indoor localization system based on WiFi, GSM and Bluetooth. One of the key factors of Redpin was that the system tried to omit the time-consuming offline/training-phase by training and improving itself during usage time by the operators. The LoCo framework from Biehl et al. [10] used a supervised classification scheme to provide a highly accurate room-level localization. The classification was based on the relative ordering of access points regarding their RSSI. Kumar et al. [11] aimed at a low "cost" (regarding training) implementation of an indoor localization method. They facilitated an angular approach by emulating a large antenna array using a new kind of Synthetic Aperture Radar (SAR) to orientate and localize the device in 3-D space. In order to gain a good overview on indoor positioning systems Zafari et al. [12] together with Jang and Kim [13] conducted some surveys.

What can be seen in recent projects is that one common aspect is the attempt to omit time consuming training and measuring phases. One of the main problems here is the lifespan of the collected data which typically is only valid as long as the environment isn't changed. It is also striking that many indoor systems do not specify concrete coordinates as a location reference. These developments are reflected by the proposed system by working without a training phase, being applicable to any building and facilitating the use of location sectors.

III. Positioning Library

In order to estimate the most probable position sector, the data collected from scanning the environment has to be processed. Because of this, the main part of the positioning library is structured as a filter pipeline (depicted in Figure 1) that analyses, processes and refines the provided raw data step by step.

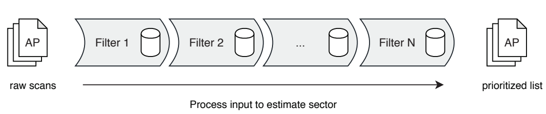

Figure 1. Structure of filter pipeline

A. Structure

The filter pipeline consists of a set of filters, each applying transformations to the original input data. Each filter in the pipeline has a single streamlined responsibility. With each step in the pipeline, the input data is further processed, taking the input from the previous filter and producing output for the next. Each filter can be applied an arbitrary number of times and is configurable on its own using a set of internal parameters. Each occurrence of a filter in the pipeline may therefore use a different configuration. Additionally, filters are able to hold a state (its memory) to keep track of previously analyzed scans. This way, the form of movement can be derived based on the changes in sensor values over time. In the following, we will present two example filters namely the Pressure Filter and the Graph Filter.

1) Pressure Filter

As buildings typically contain multiple floors it is important to know on which floor an entity is. The Pressure Filter, which keeps track of the contextual value ambient air pressure, is used to recognize possible floor changes. Looking at the temporal progression of the measured air pressure, the elevation changes of the target can be derived. The filter stores the most recent air pressure measurements, provided by the smartphone's barometer, ordered from least to most recent measurement. The number of measurements stored is determined by its configurable parameter capacity defining the internal window size. For these measurements, the change in pressure from one measurement to the next is calculated. In case of a significant increase or decrease a possible change of the floor level was deduced.

2) Graph Filter

Another important aspect of buildings is that floor changes cannot be carried out at any given location and that one can only move within a building on defined paths namely the hallways. Jumping from one building wing to another is considered not realistic. The Graph Filter restructures this topological information based on the installed access points forming a so called Access Point Graph as seen in Figure 2 where each node represents an access point and therefore a possible positioning sector. Nodes are only connected if they are directly reachable through, i.e., a hallway or a stair case if a floor change is depicted. The graph can therefore be used to establish a plausibility check to recognize impossible access point respectively sector jumps. It prioritizes the possible sectors in its calculation set based on a certain distance from the previously estimated sector. This prevents the algorithm from jumping from one sector to the other without passing through the connected sectors between them and therefore jumping floors or building wings. The maximum jump distance is defined using the parameter n and describes the smallest number of hops it accepts to reach one sector from the other. If $n = 0$, then the estimation prioritizes only sectors that have been part of the previous estimation. If $n = 1$ only sectors of the previous estimation and those adjacent to them are considered to be relevant estimates. The graph representation of the access points installed in the building is passed as another part of the filter input.

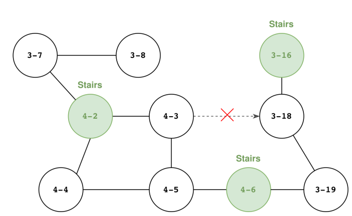

Figure 2. An Access Point Graph representing the topology of a building

B. Input

The input received by a filter contains on the one hand the scanned access points (namely RSSI and BSSID) and on the other hand additional parameters and contextual data that aid in filtering the access point scans. The contextual data, which is passed into the filter pipeline, is based on an Android smartphone's sensors and the additionally needed data like the building's access point structure is integrated as an additional parameter file and will be described in more detail in Section IV.

C. Output

The last filter in the pipeline provides the final set of access points. This set then serves as the prioritized list of sectors with the target's most likely position at the top and the least likely at the bottom. Only this final result set is then considered for the position estimation, providing the ID of the most probable access point-based positioning sector.

IV. Testing Environment

To test different configurations of the filter pipeline and the integrated filters in different scenarios (buildings and routes) a simulation system has been constructed. This system consists of two applications: a simulator and a smartphone application to record the necessary data. Both provide the means to simulate and evaluate pre-recorded walks through different buildings.

A. Labeled WiFi/Context Recording

The required data to simulate walks is collected by an Android application. The application initiates a number of scans of the environment along so-called recording route. These recording routes are defined using a set of arbitrary way points, which are defined in special JSON files. Each way point serves as a calibration point and is, in turn, being defined by its location using X and Y coordinates regarding the blueprint of the building as well as the floor it resides on. Any arbitrary location can be used as a way point. Connecting these way points form the describted recording routes.

During a recording process, the users move in a steady pace on the recording route. When passing a way point, the users signal (by pressing a corresponding application button) that a way point has been reached and the application will automatically start scanning the environment. These scans are referred to as way point scans. Way point scans are automatically associated with the respective way point's location and reflect the internal calibration points. While moving from one way point to the next, additional scans are applied automatically in the background at a fixed interval. These scans are referred to as intermediate scans. Unlike way point scans, these are not explicitly labelled with a location but their location will later be interpolated.

With each scan, information on the installed access points in the near proximity are collected. Next to RSS values and frequency, physical measurements taken by the sensors installed in the user's smartphone are recorded as well. For example, barometers, photometers, and thermometers are environmental sensors used to capture air pressure, illumination, and temperature respectively. Each scan is tagged with a timestamp, representing the time of recording the data and is later exported as a JSON file to be used in the simulator.

B. Simulator

The simulator provides the means to evaluate and adapt the indoor positioning library. Pre-recorded walks in form of JSON files can be simulated while the results of the indoor positioning library are visualized on a map of the building. A screenshot of the user interface can be seen in Figure 3.

The main view of the application displays a map of the current floor. The sectors of the installed access points are calculated based on a Voronoi algorithm and provided as an overlay of the simulator's map. Using the recorded scans, the simulator reconstructs different walks through the building. These recorded walks are visualized by blue dots on the map indicating the way points of the recording route. When a recording is playing, the actual position of the target is represented by a bright green dot on the map. If a way point scan is processed, the position of the way point is used to paint the indicator on the map. If an intermediate scan is processed, the position to draw on the map is interpolated based on its surrounding way point scans. During playback, the recorded scans are fed to the filter pipeline in sequence. With each scan, the current position is calculated based on the scan that is currently being processed. The sector that is estimated to most likely contain the target is highlighted in light-green, but the simulator can be configured to extend the highlighting to multiple sectors if necessary taking into account the calculated prioritization.

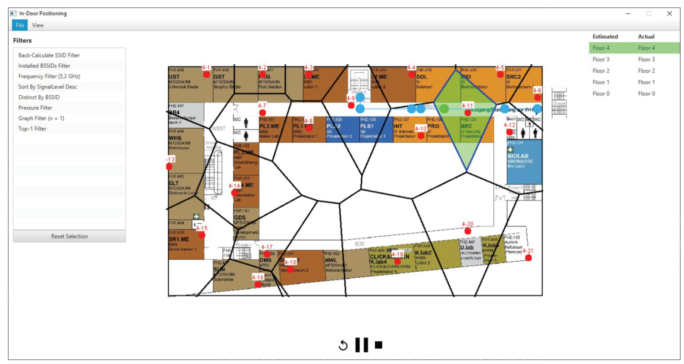

Figure 3. Screenshot of simulator

In addition, the currently configured filter pipeline is displayed left of the map. The filters in the pipeline are displayed from top to bottom, with the top most filter being applied first, and the lower most filter being applied last as part of a position estimation iteration. Via the context menu of the application, a dialogue can be accessed that enables the adaptation of the configured pipeline. Furthermore, the settings dialogue can be opened with which the simulation properties can be adapted and filters added or moved. When adding a filter to the pipeline, however, a dialogue window pops up where the parameters of the added filter can be adapted.

V. Evaluation

A. Scenarios

In order to evaluate the performance of the system, different tests at the campus Hagenberg of the University of Applied Sciences Upper Austria with different filter combinations in different pipelines have been conducted. In total, four scenarios have been defined to collect the test data and five different pipelines (three of them presented here) have been used to prioritize the access points. Each of the scenarios' focused on a different type of movement.

Scenario A focused on the change of direction. At a certain point on the route, the target turned around, followed the route back to where it came from, and finished the route in a different

 ISBN: 978-1-61208-781-8

direction. This route covered the case that sectors were passed through multiple times.

Scenario B aimed at the changing of floors. Two different flight of steps were used on this route to change from a higher to a lower floor and back to the higher floor again.

Scenario C concentrated on the target standing still for an arbitrary amount of time at a certain point on the recording route. The target started walking the predefined route, paused for several seconds before walking again.

Scenario D consisted of a regular round trip, starting from a distinct location and arriving at the same location again. The route was finished without a change of floors.

B. Metrics

In order to determine and compare the accuracy of different iterations and therefore filter combinations and configurations of the positioning library, two metrics have been defined and used. In this paper, we reference the results of the Number of Correctly Estimated Sectors (NCES) metric.

As the proposed system tries to estimate in which sector the target resides in at the current point in time, an estimation can either be correct or incorrect. The evaluated target's position is considered to be correct when the target's actual position is inside the estimated sector. So, the NCES metric determines how many of the overall performed estimations have been correct. The metric can be configured using the *toleratedDistance* parameter, which, by default, is set to 0. If a distance larger than 0 is set, an estimated sector is considered to be correct, when it is within the specified distance from the actual sector in the Access Point Graph.

C. Results

The results presented in this paper reflect the results for the four scenarios with regards to the NCES metric. For each iteration of the pipeline, the NCES metric has been calculated once with a tolerated distance of 0 (only the sector with the highest priority is regarded valid) and once with a tolerance value of 1 (the sector with the highest priority and its direct neighbours are regarded valid). The Table I displays the results of three pipeline configurations: Pressure, Graph ($n = 1$), and Graph ($n = 2$).

TABLE I. MATRIX DISPLAYING THE PERFORMANCE OF EACH ITERATION OF THE PIPELINES

		Pressure	*Graph* ($n = 1$)	*Graph* ($n = 2$)
Scenario A:	NCES (td = 0)	0.640	0.680	0.640
Change of Direction	NCES (td = 1)	0.980	1.000	0.980
Scenario B:	NCES (td = 0)	0.479	0.352	0.479
Change of Floors	NCES (td = 1)	0.930	0.634	0.986
Scenario C:	NCES (td = 0)	0.718	0.205	0.744
Standing Still	NCES (td = 1)	0.974	0.359	1.000
Scenario D:	NCES (td = 0)	0.580	0.420	0.470
Roundtrip	NCES (td = 1)	0.960	0.710	0.950

The worst result was achieved by the Graph ($n = 1$) pipeline. Even with a tolerated distance of 1, the pipeline only estimated 67.57 % of the sectors correctly on average. This can be traced back to the limited range of potential candidates considered to be the target's position. The pipeline tended to get stuck within its previous estimation after a while because the newly scanned AP at this point were too far off from the previous estimations to be considered realistic.

Overall, the best results were achieved using the Graph ($n = 2$) pipeline, with an average accuracy of 0.611 using a tolerated distance of 0 and an average accuracy of 0.979 using a tolerated distance of 1. With a tolerated distance of 1, almost perfect results were achieved. This variation ($n = 1$ vs. $n = 2$) indicates that the cell approximation of each sector is not accurate enough and more sophisticated approaches, such as signal propagation models for a better approximation of a sector's true coverage area, would be very promising extensions to improve the estimation for the Graph ($n = 1$) pipeline.

VI. CONCLUSION

In this paper, we presented the first results of our novel indoor localization library together with its evaluation system. Our first results are very promising but some challenges are still visible. One of the main error sources could be traced back to the unrealistic sectorization of the building based on its access point infrastructure by using Voronoi-based cells. Another promising improvement will be the integration of additional context sources. This should provide the system with more calibration points and help to further deduce realistic movement trajectories throughout a building. Therefore, the integration of movement trajectory and context-based calibration filters will be our next steps to improve the system.

REFERENCES

[1] M. Weiser, "The Computer for the 21st Century," Scientific American, vol. 265, no. 3, 1991, pp. 94–104.

[2] G. D. Abowd et al., "Towards a Better Understanding of Context and Context-Awareness," in Proceeding 1st International Symposium on Handheld and Ubiquitous Computing (HUC '99), June 1999, pp. 304–307.

[3] A. Schmidt, M. Beigl, and H.-W. Gellersen, "There is more to Context than Location," Computers and Graphics, vol. 23, 1998, pp. 893–901.

[4] F. Buschmann, R. Meunier, P. Sommerlad, and M. Stal, Pattern-Oriented Software Architecture, Volume 1, A System of Patterns. Wiley, Chichester, England, 1996.

[5] A. Küpper, Location-Based Services: Fundamentals and Operation. Wiley, Chichester, England, 2005.

[6] R. Want, A. Hopper, V. Falcao, and J. Gibbons, "The Active Badge Location System," ACM Transactions on Information Systems, vol. 10, no. 1, 1992, pp. 91–102.

[7] B. Schilit, N. Adams, R. Gold, M. Tso, and R. Want, "The PARCTAB mobile computing system," in Proceedings of the Fourth Workshop on Workstation Operating Systems, 1993, pp. 34–39.

[8] S. He and S. . G. Chan, "Sectjunction: Wi-Fi indoor localization based on junction of signal sectors," in Proceedings of the IEEE International Conference on Communications (ICC), 2014, pp. 2605–2610.

[9] P. Bolliger, "Redpin – Adaptive, zero-configuration indoor localization through user collaboration," in Proceedings of the ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments, MELT 2008, 2008, pp. 55–60.

[10] J. Biehl, M. Cooper, G. Filby, and S. Kratz, "LoCo: A ready-to-deploy framework for efficient room localization using Wi-Fi," in Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing – UbiComp 2014, 2014, pp. 183–187.

[11] S. Kumar, S. Gil, D. Katabi, and D. Rus, "Accurate indoor localization with zero start-up cost," in Proceedings of the Annual Intern. Conf. on Mobile Computing and Networking, MOBICOM, 2014, p. 483–494.

[12] F. Zafari, A. Gkelias, and K. K. Leung, "A Survey of Indoor Localization Systems and Technologies," IEEE Communications Surveys Tutorials, vol. 21, no. 3, 2019, pp. 2568–2599.

[13] B. Jang and H. Kim, "Indoor Positioning Technologies Without Offline Fingerprinting Map: A Survey," IEEE Communications Surveys Tutorials, vol. 21, no. 1, 2019, pp. 508–525.