Organisation of Indoor Navigation Data from a Data Query Perspective

Martin Werner and Moritz Kessel
Mobile and Distributed Systems Group
Institute for Informatics of the LMU Munich
Munich, Germany
{martin.werner |moritz.kessel}@ifi.lmu.de

Abstract—**Indoor Navigation is becoming a more and more important topic. It is a key tool for the integration of disabled people into environments which they are not familiar with. With this paper we want to address the problem of indoor navigation being much more complex than outdoor navigation with a thorough modelling of queries which a indoor navigation GIS database shall be able to answer quickly and present an organisation of indoor navigation data tailored to scalable and flexible indoor navigation.**

Indoor Navigation, Geospatial Information

I. Introduction

In recent years localisation and navigation have been important topics in research. This includes outdoor navigation as well as indoor navigation. While for outdoor navigation a good data basis is available in the network of streets, this is not true for indoor navigation. The topological relations inside a building can be very complex while outdoors there is a limited complexity. Furthermore the way people interact with their surroundings is much more complex for indoor navigation as compared to the situation of outdoor navigation where a car on a specific part of a highway has no choice but to follow the road. Inside buildings the actual way that is taken by a human person is relatively free especially in bigger rooms containing much free space.

Inside a building the environment plays a more complex role than outdoors. As it is merely impossible to use satellite navigation inside buildings (for now), it is evident that there has been much research into alternative ways of positioning. Any sensor technology that enables a person to analyse aspects of the surroundings can be used to estimate the position. There are systems using received signal strength indication (RSSI) of wireless networks such as WLAN or Bluetooth as well as approaches with specialised sensors and tags (infrared light, UWB, ultrasonics).

In practice all these approaches need a check to weight measurement and walking plausibility in advanced filters for minimum-variance estimation of linear and non-linear stochastic systems. So it is evident that the positioning will need very detailed information about the building which is not true outdoors. While the error of GPS positioning is easy to remedy with filtering using a map, the positioning error in indoor environments is usually so high, that a simple filtering scheme does not provide sufficient accuracy.

With this paper we want to provide an extensive analysis of the questions a navigation system will have about the surrounding environment. On this background we propose to set up a system optimised around these questions. It is, of course, important to have a general model of the environment in a central data basis, but for the purpose of actually implementing a navigation system such a data basis is seldom feasible.

Indoor navigation data tailored to a specific query of the navigation system can be generated by hand which is extremely expensive as the editor of such data has to know very much information about a building as well as about indoor navigation and the pitfalls with e. g. turn by turn guiding in indoor environments. This is a real problem prohibiting the exploitation of mobile navigation technology for indoor environments in large scale. It is thus one of the most important challenges to simplify this creation process as much as possible.

The best situation would be to have an extensive model of the surroundings which is easy to generate, edit and understand combined with an automated way to reorganise this data such that it is ready for use in the components of an indoor navigation system.

II. The Art of Indoor Navigation

In this section we want to give an overview over existing research work in the area of Indoor Navigation. We decided to divide this topic into the three topics Positioning, Way-Planning and User-Interface. Though this leads to overlappings (usually a positioning system is evaluated with a navigation application) we get a more structural overview of those topics.

A. Positioning

There are many positioning technologies available for indoor use. It is possible to bring out a sensor network which sniffs identifications of active tags, e.g. as in the Active-Badge system [1]. Orthogonal to this approach one could bring out

978-1-4244-7879-8/10/$26.00 ©2010 IEEE

infrastructure that sends identifications out which can be received by the items to be located, e.g. as in the Cricket positioning system [2]. This type of systems is very expensive and its accuracy is limited by the sensor network density.

A better class of positioning systems is based on radio technology. Most of these systems use an existing WLAN infrastructure to derive positioning information. There are again two orthogonal approaches: Namely tracking the position of items by having strong time synchronisation between different access points and estimating the position of an item from signal strength information and Time-Difference-of-Arrival (TDoA) information. Alternatively a client can measure signal strength information and use this RSSI-fingerprint for a search in a database of known RSSI-fingerprints ([3], [4]).These systems typically work at an accuracy level making rooms distinguishable and have severe problems with localisation inside free space, because the difference between fingerprints is strongly related to attenuation effects induced by walls. Though this can be taken into account by using non-linear signal processing strategies such as particle filters ([5], [6]) it is impossible to locate a user precise enough for a navigation application experience comparable to the outdoor situation. There have been some WLAN-based positioning systems which used a non-probabilistic method using an indoor WLAN propagation model. But these systems did not survive because no good general propagation model was found and the results of fingerprinting-based systems are better.

A newer approach to indoor positioning with radio technology is based on Ultra-Wideband (UWB). Ultra-Wideband is any radio signal with a fractional bandwidth of more than 20% or an absolute bandwidth of more than 500 MHz [7]. With Ultra-Wideband technology it is possible to transmit and measure very short pulses and hence to detect and remedy effects based on multi-path propagation. This makes it possible in a tightly time-synchronised network of sensors to use both Angle-Of-Arrival (AoA) and Time-Difference-Of-Arrival (TDoA) information for position estimation. The commercial Ubisense Real-Time Localisation System (Series 7000 [8]) for example provides three-dimensional positioning within an accuracy of 30cm. This accuracy can be optimised with special filtering based on the movement model or known parameters such as fixing the height of a tag.

Another alternative way of positioning is to use some presence detection technology. This can be an RFID reader as in [9] where the only localisation technology is interaction with a screen or some bar codes brought out to known places in the environment. These technologies enable very exact positioning for some (typically few) positions. If for example a user is standing in front of a desk with a display and interacting with a card reader we know his position and his viewing direction very exactly.

Another upcoming positioning technology tries to use sensor-enabled smart phones for inertial navigation. Though the sensors have good accuracy nowadays the problem is here that the data used for navigation (distance, speed) is to be calculated as an integral over the measured values. So rather small errors will add up to large errors over time. Inertial navigation with such sensors is only possible for a very short period of time. However these sensor data can be integrated into a WiFi positioning system such as in [10] resulting in better positioning performance.

B. Way-Planning

Another common task of a navigation system is way-planning. Way-planning typically constructs a walkable way out of a list of points-of-interests with some constraints. These constraints can be the ability to walk stairs or some weak ordering on the points-of-interests. The most basic data structure for routing is a graph consisting of a set of vertices $\boldsymbol{V}$ and a subset $\boldsymbol{E}$ of the product $\boldsymbol{VxV}$. A vertex is typically assigned to a point in navigation space and an edge typically models the possibility to walk from the first vertex of this edge to the second vertex of this edge.

Algorithms for navigation on a graph are manifold. The most basic algorithms just try to walk any way in the graph starting at a vertex and find out which was the shortest to another vertex. Algorithms of this class are breadth-first-search (BFS), depth-first-search (DFS) and Dijkstra. Then there are some algorithms which try to be faster than these basic searches by using some guidance information. A* for example needs a heuristical function which estimates the distance to the goal in a consistent way. Though it is obvious that such a function always exists (i.e. just take the correct topological distance), it is not clear whether such a function exists that is computable in constant time. Especially for indoor navigation where multiple floors are interconnected at very few positions, it is not easy to formulate a good heuristical function computable in constant time.

Though graphs are the foundation of navigation they are often implicit. Navigation meshes for example organise the walkable space in polygons (often triangles) and the (implicit) graph consists of some interconnection of the middle points of the triangles and the three edges of the triangles.

On top of this basic search technology there are plenty of optimisations which can be done and which can (sometimes) optimise the running time of way planning algorithms.

C. Visualisation and User-Interface

Indoor navigation systems have been set up with a broad diversity of interfacing devices including wireless ear-phones, small cellular phones, smart phones and desktop computers.

From this diversity many implications to the way-planning and geometry organisation come up. While a pure audio guide can be based on a graph which does not allow for an embedding into the navigation space and where every non-trivial edge (e.g. an edge that cannot be left out for an audio description) produces a message, the situation for a navigation system which wants to visually display the route is different. For this type of application it is helpful if the navigation graph comes with an embedding into the navigation space allowing us to silently identify vertices and points as well as edges and walkable lines. When it comes to visualisation on a smart phone or a computer, we might want to have a fully detailed three-dimensional model of the environment which makes some visualisation tasks difficult (e.g. special staircases, camera movement while changing a floor).

III. CONCEPT

A comprehensive indoor navigation system will more or less perform the following activities to complete a navigation request. The ordering of course depends on the situation as well as whether a specific operation can be dropped depending on the individual application.

- Obtaining navigation dataset, task information and profile information
- Estimation of possible positions in a local coordinate system
- Position refinement including navigational situation
- Constrained way-planning for several points of interest including up-to-date context information
- Evaluation of whether the last way-planning fits profile information (e.g. are there constraints which might prohibit this solution)
- Enrichment of the solution with semantical information (room names, useful landmarks, ...)
- Generation of desired multimedial representation (3D graphics, 2D graphics, text, audio, video, ...)

All these operations use information about the surroundings in some specific form. A radio-based positioning system is for example often interested in the position of walls due to predict attenuation effects. To further optimise the quality of the position it can make sense to evaluate several predictions of the positioning engine with a plausibility check based on particle filtering using the actual navigation task and the probability of choosing a wrong way as well as the measurements to expect if such a wrong way has been chosen. The way planning engine again uses completely different data.

Classical way planning uses a graph structure of walkable ways (similar to the outdoor situation). Modern approaches use navigation meshes or free-form navigation without a reduced explicit discrete graph structure. Such data is only implicit in most geometric modelling languages. Nevertheless it is possible to extract such information from these models. Visualisation and instruction generation need much more semantic information (room names, landmarks, doors, stairs, elevators, ...) for their operation. In the following we will describe the information requests which are issued by an imaginary indoor navigation system.

A. GIS-Queries of an Indoor Navigation System

With this paper we want to analyse the data that an indoor navigation system will need to perform the most important tasks as described before. After a discussion of the queries that should be issued, we will explain implications for indoor environmental models.

1) Queries related to positioning and refinement of positions

As explained above many indoor positioning systems are based on a probabilistic model comparing a database of known measurement-position combinations with actual measurement parameters. For further refinements interpolation based on a deterministic model can be used and therefore the positioning engine needs to know about the positions of walls, doors and free space. Now it is still possible that the system has several different positions that are possibly correct. Classical systems would now trust in a specific fixed mobility model of the user and rule out all positions which are "implausible" with respect to this movement model. This can be enhanced by using the navigational context (goal, target, user constitution, proposed route, ...) as an oracle for the plausibility of positions. The positioning engine will query for the target and the route that the user shall be on and compare the measurements made with stored measurements for the correct way and for possible wrong way decisions. In this way we are able to detect with high probability the situation of the user leaving the route.

2) Queries related to way planning

Way planning is a big research topic in itself. Though it is easy to find the shortest way in many structures this shortest way can have many disadvantages. It is seldom the case that this way is comparable to a way that a human being would choose. Especially in the area of artificial intelligence for computer games and virtual reality many algorithms have been developed to remedy this.

The queries in this area are queries for the distance to walls, the size of free space surrounding a position, the area that can be looked over from a position and many similar local requests. As the interaction with the surroundings is very complex in the indoor area, we need a means of describing special areas and their implications to way-planning. This includes stairs, elevators and waiting areas.

3) Queries related to visualisation

The queries related to visualisation are manifold. While we first of all need some geometric information to be shown (e.g. a list of lines, a bitmap, a three-dimensional model) the complexity comes with visual augmentation of the navigation instructions. We need to search for landmarks which are visible for the user while he is walking his way and can be used to enhance orientation inside the building. As a typical building has many objects which can serve as a good orientation landmark in some situations it is difficult to reduce the set of all possible landmarks to a sensibly small set such that a user is able to remember this information.

4) Queries related to textual instruction generation

For the generation of textual instructions for a navigation application we need semantical knowledge about a building. The geometric queries in this area are checks for the line of sight (e.g. whether a point can be seen from another point). To remedy the problem of generating too many instructions, we need a semantical search method to evaluate the relevance of a given pre-instruction to the way-finding problem. Again profile information can be needed to select the correct language and granularity of navigation instructions. The most challenging part is here of course the integration of geometry and semantics

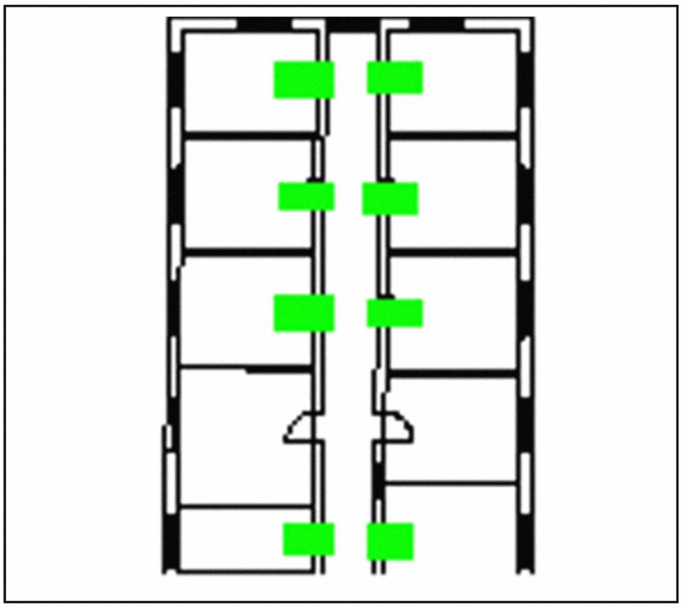

Figure 1: Local Topology Search Bitmap

Figure 2: An Accessibility Graph modelling access in a fixed discretisation

that is translation of semantic information into geometric information and vice versa.

B. Building-Blocks for a Complete Navigation Dataset

We identified several different data structures which can make several queries computable in acceptable time. The most often overlooked problem in constructing good indoor navigation systems are the difficulties with intersecting complex geometric objects such as arcs, splines and circles, which come up very often in buildings. Though it is often possible to approximate those rounded surfaces with lines this leads to pretty big line lists.

To support collision detection with a predictable running time and precision, we decided to render such data into a bitmap once and do line-painting on the resulting map. In this way, we have better running times for short distance collision detection.

The running time of a line painting algorithm linearly depends on the length of the line while the same request on top of a line-list depends linearly on the number of lines. Hence with our data structure local requests are faster than global requests which is not true for line-list based tests. As we have seen in our discussion of navigation queries most parts of a navigation system are only interested in local data in the sense that information that is not visible to the user or otherwise sufficiently near to the proposed route is irrelevant.

We decided to organise the data of an indoor navigation system into building blocks where each building block is well-suited to support one or more of the elaborated queries of an indoor navigation with high performance. A building block is some self-contained set of data that supports a specific query. Building blocks can be distributed in the network or on different cores of a multi-core system. Of course such an organisation implies overlapping between the information in different building blocks.

1) Building Block 1: Global Topology Information

The topological information which is used for indoor navigation models the relations between rooms and ways and shall enable us to compute a path inside a building possibly including context-information. Though it is possible to generate one general graph or to generate good graphs on the fly out of other building blocks, we want to allow the storage of several different graphs which are completely independent from each other. To be correct, we do not store the graph in itself, but we store a graph with an implicit embedding storing a coordinate in navigation space for each vertex and identifying edges with straight lines interconnecting those two vertices. Hence a graph for indoor navigation consists of a list of vertices with an assigned point for each vertex and a list of edges interconnecting vertices.

In our actual navigation application we use the following graphs for navigation depending on the situation.

- An Accessibility Graph which models the accessibility of all free space in a specific fixed discretisation and a locally self-similar structure
- A PoI Interconnection Graph which is a complete graph containing only PoI's as vertices and aggregated weights
- An Inter-Floor-Connection Graph which contains an edge for each connection of two different floors. This graph has to be a subgraph of the Accessibility Graph
- A reduced navigation graph which is suitable for efficient textual instruction generation

All these graphs exist in two favours: One which comes with a weight-map modelling the distance in meter where special decisions have to be made for connections between different floors and another graph which has an implicit weight-map modelling walking time depending on profile information about the users abilities and assumed speed and context information about the building infrastructure such as waiting times at a passport control or average waiting times for an elevator. An example of an accessibility graph is given in figure 2.

2) Building Block 2: Visualisation Information}

This building block consists of several datasets which allow for visualisation for all different device-classes taking part in the navigation applications. Visualisation information can come as a coloured bitmap floor-plan for each floor with a specific projection mapping two-dimensional bitmap coordinates to three-dimensional navigation coordinates and vice versa. Furthermore visualisation data can be arranged in a device-specific form to reduce the impact of decoding a generic description. Examples range from explicit OpenGL vertex arrays in several levels of detail which make it possible to transport them over the air to a mobile device where we make use of OpenGL ES to visualise this information to a CAD dataset which is directly used for visualisation. The queries related to visualisation are only partially resolved inside this building block, the landmark search is supported by Building Block 3.

3) Building Block 3: Local Topology Information

This building block consists of several bitmaps which contain the original geometry of the building enriched with several additional information such as the projection parameters to project points from CAD-space to bitmap space and vice versa. Areas of a specific colour will represent special objects such as landmarks and points-of-interest where a colour table holds a mapping between these colours and the semantical objects. Moreover we define a fixed specific colour to mark areas which do not represent a real obstacle for a navigation user though there is geometry inside the CAD drawing (e.g. organisational lines, small objects of no relevance). An example is given in figure 1.

4) Building Block 4: Semantical Information

This building block consists of semantical information such as room names, room polygons, information needed for textual instruction generation (patterns, translations, evaluators) and functional groupings of objects (restrooms, stairs, elevators). The most challenging part for this building block is to have a geometric representative of each semantical information and a translation between the semantical world and the geometrical world. Additionally this building block will serve as a connection between a semantical representation (e.g. restroom), a visual representation (e.g. a pictogram) and a geometric representation (e.g. a position in navigation space where this facility is located).

IV. Prototype and Performance Evaluation for Line-Of-Sight Queries

As a measurable reason of considering the queries issued by a navigation system during design of the environmental model we have implemented a Line-Of-Sight query in two ways: On a CAD database (essentially a list of lines in world coordinates) implementing a quad-tree spatial organisation and on a projection of the same CAD data to a bitmap with a pixel size in real world coordinates of 0.15m x 0.15m. We have then performed 100.000 random Line-Of-Sight queries on these data bases and compared the speed of the result. We performed the experiment 10 times and the results are given in the following table:

TABLE I. PERFORMANCE EVALUATION RESULTS

No	Operations per Second	
	On Map Data	***On CAD Data***
1	9049,77	307,88
2	8869,18	307,13
3	8984,73	292,06
4	8992,81	292,14
5	9000,9	292,48
6	8996,85	291,97
7	8583,69	308,17
8	8960,57	290,87
9	9004,95	291,29
10	8932,56	292,23
Average	8937,6	296,62
Std. Dev.	1,49 %	2,59%
Max. Dev.	5,21%	5,83%

The experiments were conducted on a single core without using possible hardware accelerations. As a result we can conclude that we sped up this operation just by choosing a good data structure by a factor of 30. It was of course easy to generate this optimised data structure out of the highly structured form (e.g. the CAD drawings). Of course, we have to pay a price for this speedup in terms of memory usage. The bitmap we constructed out of the CAD drawings of Munich Airport is 20 MB large whereas the line-quadtree is only 300 KB in size.

V. Conclusion

With the work presented in this paper we actually made possible a large-scale indoor navigation application. This shows the feasibility of our approach. Most indoor navigation systems that have been studied before suffer from scalability problems if the area of navigation is big. With our analysis of the queries which a navigation system shall be able to issue we are a step nearer to a general and scalable navigation application. The two main benefits are that subsystems such as the positing engine can use navigation information for better position estimation and that we have sped up some requests so much that they can be used very often and without much harm bringing possibilities for visualisation and augmentation that were not there before. Our future work will focus on integrated navigation applications sharing data and algorithms for better functionality, performance and scalability. As a last sentence we want to emphasise that this approach does not conflict with other work on standardisation of geometric modelling languages as most of the information in our building blocks can be derived from any good modelling language.

REFERENCES

[1] R. Want, A. Hopper, V. Falc and J. Gibbons, "The active badge location system", *ACM Trans. Inf. Syst.,* vol. 10, no. 1, pp. 91-102, 1992.

[2] N. B. Priyantha, A. Chakraborty and H. Balakrishman, "The cricket location-support system", in *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking.* New York, NY, 2000.

[3] P. Bahl and V.N. Padmanabhan, "Radar: an in-building RF-based user location and tracking system", in *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies,* vol. 2, pp. 775-784, 2000.

[4] Y. Chen and H. Kobayashi, "Signal strength based indoor geolocation", in *Proceedings of IEEE International Conference on Communications (ICC '02),* vol. 1, pp. 436-439, 2002.

[5] M. Arulampalam, S. Maskell, N. Gordon and T. Clapp, "A tutorial on particle filters for online non-linear / non-gaussian bayesian tracking", *IEEE Transactions on Signal Processing,* vol. 50, no. 2, pp. 174-188, 2002.

[6] F. Gustafsson et. al, "Particle filters for positioning, navigation and tracking", in *IEEE Transactions on Signal processing*, vol. 50, no. 2, pp. 425-437, 2002.

[7] D. Harmer, "Ultra-wide band (uwb) indoor positioning", online, Accessed 10.7.2010, Available: http://esamultimedia.esa.int/docs/NavigationProjects/UWB_tech_Proj_Overview.pdf

[8] Ubisense Inc, "Series 7000 slim tag factsheet", online, Accessed 10.7.2010, Available: http://www.ubisense.net/en/resources/factsheets/series-7000-slim-tag.html.

[9] P. Ruppel, F. Gschwandtner, C. K. Schindhelm, C. Linnhoff-Popien, "Indoor navigation on distributed stationary display systems", in *Proceedings of the International Conference on Computer Software and Applications,* vol. 1, pp. 37-44, 2009.

[10] F. Evennou, F. Marx, "Advanced Integration of wifi and inertial navigation systems for indoor mobile positioning", in *EURASIP J. of Applied Signal Processing,* 2006.