A Hybrid Spatial Model for Representing Indoor Environments

Bernhard Lorenz, Hans Jürgen Ohlbach, and Edgar-Philipp Stoffel

University of Munich, Oettingenstr. 67, 80538 Munich, Germany,
{lorenz,ohlbach,stoffel}@pms.ifi.lmu.de,
http://www.pms.ifi.lmu.de/

Abstract. In this article we propose a hybrid spatial model for indoor environments. The model consists of hierarchically structured graphs with typed edges and nodes. The model is hybrid in the sense that nodes and edges can be labelled with qualitative as well as quantitative information. The graphs support wayfinding and, in addition, provide helpful information for generating human-oriented descriptions of an indoor (and outdoor) path.

Keywords: indoor wayfinding, indoor navigation, hybrid spatial representation, qualitative spatial representation, topology, graph, hierarchical graph

1 Introduction and Motivation

Car navigation systems are becoming a more or less standard commodity nowadays. Since they are a mass product, the problem of navigating cars through large road networks has been well investigated and the solutions are mature. Much less well investigated is the problem of navigating pedestrians through airports, train stations, libraries, hospitals, supermarkets, etc. Unfortunately, it turns out that navigation problems in large buildings are quite different from the ones encountered in large road networks. The main reason for this is that the topological structure of buildings is much more diverse than the topological structure of road networks: Whereas roads are primarily one-dimensional structures with landmarks aligned along them, areas in buildings are really two-dimensional (as in floor plans), or, when multiple storeys are taken into account, even 2.5-dimensional structures [4].

The most efficient wayfinding [8] solutions use shortest-path algorithms in graphs. Therefore, we propose a graph representation also for indoor environments, yet the graphs are much more structured and enriched with extra qualitative and quantitative information assorted. Beyond, when considering two-dimensional structures not only adjacency has to be modelled, but also containment. We introduce a hierarchical graph structure in section 3.

The main characteristics of the graph model are:

– the two-dimensional areas in buildings are partitioned into cells, and these cells are represented as nodes in the graph (Section 2). Doors and other passways which represent possibilities for persons to move from one cell to another are represented as edges;
– in order to facilitate hierarchical planning, there are different levels of abstraction in the graph (Section 3). For example, a storey in a building may be represented as a graph at a certain level, this entire graph being just a node in a graph at a higher level which stands for the whole building. The edges in the abstract graph connect the different storeys;
– the nodes and edges of the graph are labelled with hybrid information to support wayfinding as well as the generation of a human-understandable description of a path. Hence, we distinguish different types of nodes (Section 4). For example, rooms and corridors are both represented as nodes, but with different labels. As we shall see, it is quite useful to maintain a list of doors and windows in a room, all sorted by their angle against a fixed point of reference (Section 4.1). Corridors, on the other hand, are essentially one-dimensional structures for which it is useful to maintain the sequence of doors at the left hand side and the sequence of doors at the right hand side (Section 4.2).

The indoor model is described in more detail in the subsequent sections. However, we want to emphasise that the model is deliberately kept flexible. The node and the edge types as well as their labelling can be extended when it turns out that this is suitable for future applications.

2 Cell Decomposition

For buildings with simple rooms and corridors, that is to say rather small rooms (unlike, for instance, an entrance hall where hundreds of people fit in) and narrow enough corridors (not stretching over several parts of a building), there is a direct one-to-one mapping to a graph structure. Rooms and corridors are represented as nodes, and the passways between them as edges. In Fig. 1, where an extract from a blueprint of a university building is shown, such a graph structure is laid over the floor plan.[1] Two rooms which are connected by two or more doors have two or more edges between the corresponding nodes (like the entrance hall and the main corridor in Fig. 1).

However, strictly pursuing this naïve approach becomes difficult for larger buildings with large areas of open space, as for instance an airport. Following Bittner [1] we divide the free space C_{free} in this case into non-overlapping, disjoint cells C_r such that $C_{free} = \bigcup_r C_r \wedge \forall i \neq j : C_i \cap C_j = \emptyset$. Adjacent cells are connected by a link. The main corridor in Fig. 1 is actually split into several cells due to its length. Otherwise, impractical route descriptions like *"turn left to the main corridor and take the 32nd door on the right"* may result.

[1] The stairs to the other two storeys were omitted for keeping the example simple.

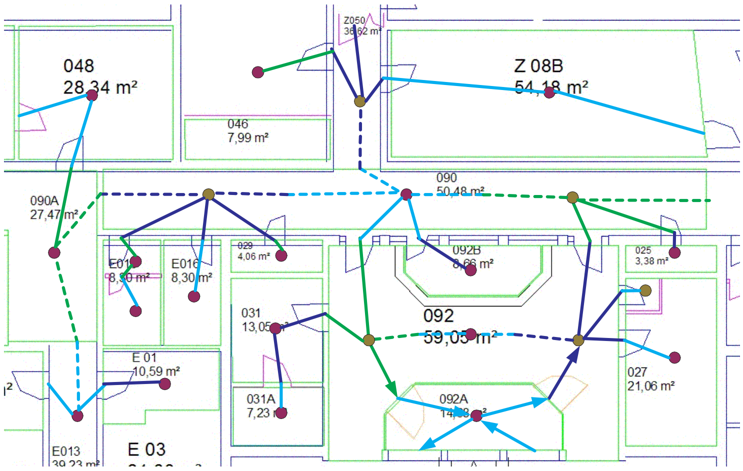

Fig. 1. Floor Plan Overlaid with Cell Centres and a Path System

The sheer size of a room may be a reason to decompose it into cells. Other reasons have to do with concavity of rooms, or with the functionality of certain areas in a larger open space. For example, an airport lounge may feature waiting areas, meeting points, areas in front of the different counters and security checks, passport control, etc. All of them serve a different purpose, and this must be represented in the graph.

Unfortunately, there is no obvious way to fully automate the cell decomposition. It has to be designed very carefully, taking into account the purpose of the different cells.

3 Hierarchical Graphs

If you are at the first floor of a large building, and you ask someone how to get to a particular room, the explanation may well start with *"Go to the third floor ..."*. What is behind this is a two-level (or, in general, multi-level) hierarchical model of the building. An example is depicted in Fig. 2. The upper hierarchy level consists of the storeys, and the lower level models the topology of each storey. In addition, the hierarchy shown in Fig. 2 also has an intermediate level which consists of wings. Navigation between different storeys usually consists of the steps *"go to the lift (staircase etc.)"*, *"go to the target storey"*, *"navigate the target storey"*. This is a typical case of hierarchical planning as it has been investigated in Artificial Intelligence for decades.

Our graph model supports hierarchical planning by providing hierarchical graphs. Each graph has a level (in the hierarchy) and an identifier. Graphs at higher levels can have as node labels the identifiers of graphs at lower levels. But this is not enough. There must also be a possibility to access graphs at higher levels from nodes of graphs at lower levels. This is done by classifying certain nodes of graphs at level n as "interface nodes" to graphs at level $n+1$. Physically, these interface nodes may represent access points to staircases, lift doors, etc. (see Fig. 2).

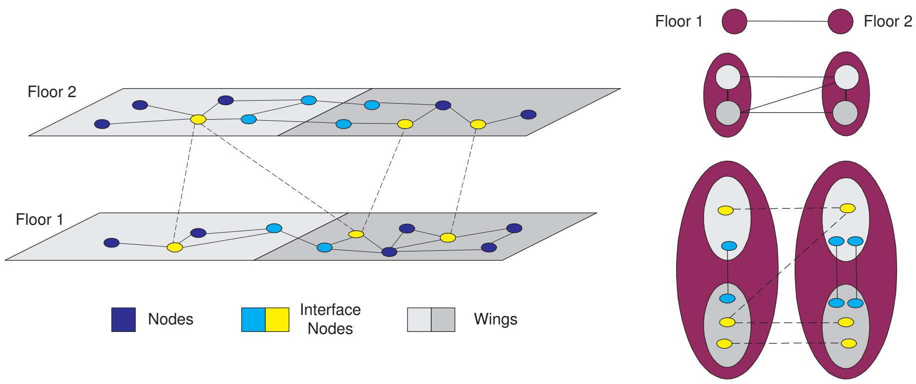

Fig. 2. Hierarchical Graph

The primary use for the graph hierarchy is of course the representation of different storeys in a building. Other use cases may necessitate the representation of different wings in a building (as in Fig. 2). Wings and storeys yield a hierarchy of three levels. If it makes sense to subdivide wings further, one may have four or more levels (see Fig. 3). On the other hand, there may also be further levels above the level of storeys. If we want to represent not only a single building, but, say, the whole campus of a university with many buildings, each building would be a node in a graph one level above the level of storeys.

A further use of hierarchical graphs can be the representation of areas which are contained within each other. As an example, consider the vegetable area in a hypermarket. The vegetable area may be subdivided into the area with the salad, the cucumbers, the carrots, etc. In the hierarchical graph model, we would have a node for the vegetable area at some level n, and this node refers to the graph of the salad, cucumber etc. areas at level $n-1$.

The edges in the graph at the 'building level' represent walkways or streets. In the simplest case, such an edge contains solely the information that it is *possible* to get to another building. If we want more detail on *how* to get to this building, we must link the edge with another graph which describes the walkways and the road network. Therefore not only nodes of a graph at level $n+1$ can represent graphs at level n, but also edges at level $n+1$ can represent graphs at level n. The only difference is that an edge at level $n+1$ must correspond to a graph at level n with two interface nodes, one for each end of the edge.

Granularity \ Model Element	Graph	Vertex	Edge
Level 4 (coarsest)	City	Building	StreetNetwork
Level 3	Building	Storey, *Staircase*, *Elevator*	
Level 2	Storey	Wing, Room, Corridor	Door, Window, Ladder, Ramp, Stairs
Level 1	Wing	Room, Corridor	
Level 0 (finest)	Room, Corridor	PartOfRoom, PartOfCorridor	

Fig. 3. Relations between Hierarchy Levels and Graph Elements

4 Node and Edge Types

Wayfinding by means of shortest path algorithms requires no more than a graph and a cost function. A simple cost function measures the geometrical distance between two places. More sophisticated cost functions can, for example, distinguish between lifts and staircases by making the staircases more "expensive". A minimum of semantic information is sufficient for this purpose. It turns out that the problem of wayfinding is considerably easier than the problem of describing an indoor path in a human-understandable manner. Humans use a combination of mostly qualitative information ("use the door *at the end* of the corridor") with little quantitative information ("take the *second* door to your left") for describing routes. Landmarks, which are very important in outdoor scenarios ("after passing by the church"), however, seem to be less important for indoor scenarios.

In order to support the generation of descriptions of a path through a building, we need to enrich the graphs with a lot more semantic information. Therefore it is necessary to classify indoor areas and to attach further class-dependent information to the nodes and edges. In this paper we illustrate the node types with two examples, namely 'rooms' and 'corridors'. Other types could be 'waiting area', 'meeting point' etc. In hierarchic graphs with many levels, node types like 'wing', 'storey', 'building' etc. are needed.

The node and edge types correspond directly to an ontology of building components. At present it is, however, not yet clear whether it is possible to describe the ontology in a formal description language like the Web Ontology Language, in short OWL[2], and to automatically incorporate the OWL concepts into the graph data structures. If this were indeed possible, it would make the graph framework much more elegant and flexible.

[2] http://www.w3.org/TR/owl-features/

4.1 Rooms

Rooms which are not further decomposed into cells are represented by a single node. Each door is represented by an edge leading to the neighbouring rooms. This is not enough information for generating instructions like *"take the second door on your left"* with the optional clarification "*[the door] directly opposite the window to your right"*. In the event of further information being available, one could of course add the coordinates of the corners of a room and those of all doors and windows. It turns out that for generating instructions like the ones above, it is sufficient to have a less complex data structure, such as a list of angles between the doors (or windows, respectively) and a reference line which goes from the centre of gravity of the room to a fixed reference point at the wall (we use the most north-western corner). An example is depicted in Fig. 4.

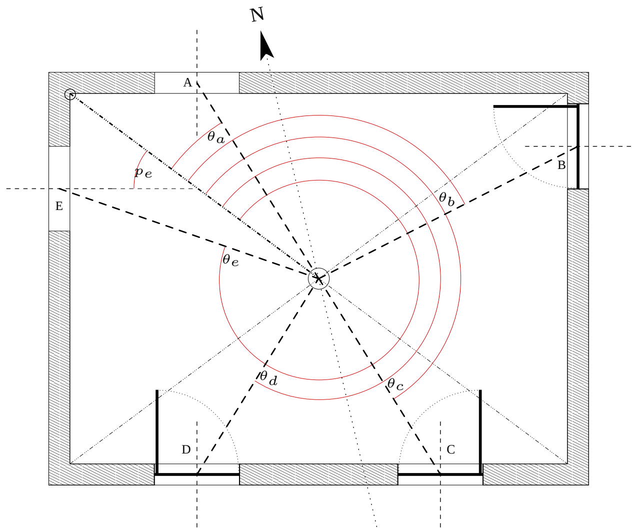

Fig. 4. Hybrid Model of a Room

A path crossing the room by entering through door B and leaving through door D may, for example, be described by the statement *"take the second door on your left"*. The information *"second door on your left"* can be computed as follows: the trajectory from B to the centre divides the room into *left* and *right*. Doors C and D are to the left and windows A and E are to the right. This can be derived from the angular distribution of the doors and windows. Thus, D is 'to your left'. The fact that D is the second door on your left can simply be obtained, by counting the number of doors in clockwise direction from B to D.

The further clarification "*[the door] directly opposite the window to your right"* can only be generated when the angular orientation of the walls is also stored. Together with the orientation of the doors and windows one can find out whether there exists another door or window which is situated opposite to door D.

The methods described above implicitly assume that the person entering at door B is looking towards the middle of the room. For this case it is sufficient to store a single angular distribution at the room node. If, however, the person looks straight forward when he enters the door, his notion of left and right may be different. Window E would now be to the left instead of right, for example. To account for this, one must compute the angular distribution for each door separately and store it at the corresponding end of the edge that represents the door. The line of reference for the angles crosses the middle of the door and is perpendicular to the door.

For many notions there are phrases in the human language which describe these notions with varying degree of precision. For example, there exist several degrees of *opposite*, such as *somewhat opposite*, *fairly opposite*, and *directly opposite*. A possible mathematical representation of these fuzzy notions are *fuzzy sets*, in our case fuzzy angular distributions (see Fig. 5)[3]: This has the advantage that deviations from an angle, like for the notion of being opposite, can still be regarded as being opposite, but only to a certain degree (determined by the membership, a fuzzy value between 0 and 1). The choice whether to use, for example, *somewhat opposite* or *fairly opposite* can be done by evaluating the corresponding fuzzy values on the distribution. If, say, the fuzzy value for the angle has been evaluated to 0.6, it would qualify as *somewhat opposite* whereas 0.95 would be considered as *directly opposite*. It is practical to use several intervals with decreasing threshold values for the various levels of 'opposing'.

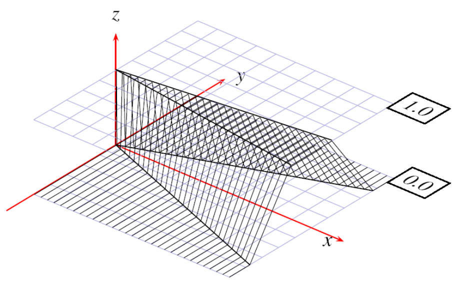

Fig. 5. Fuzzy Angular Distribution ($\theta_c > 0$, $\theta_s > \theta_c$)

4.2 Corridors

There are in fact two ways for modelling corridors. The first method is to decompose a corridor into cells such that each entrance to the corridor can be

[3] Depending on the application, core angles θ_c (fuzzy value of 1) and support angles θ_s (fuzzy value > 0) can vary.

associated with a representative cell. Adjacent cells are represented by edges between the corresponding nodes. The main corridor in Fig. 1 can modelled this way, leading to a representation of seven cells for the seven adjoining doors. This representation is completely sufficient for solving wayfinding problems. It is, however, very cumbersome to generate a statement like *"take the third door to the right"* and from a practical point of view, it is certainly not the most elegant and compact[4] data structure.

The second way is illustrated in Fig. 6. The whole corridor is represented by a single node. However, this node actually stands for a directed linear structure leading from the front to the end, with openings both on its left hand side and its right hand side.[5] It does not affect the general notion of linearity whether the corridor is distorted, since the notions of 'left' and 'right' are relative and thus change accordingly. The node must have labels which represent the entrances at the left side of the corridor, the entrances at the right hand side of the corridor, and the entrances at both ends of the corridor. The list of edges must reflect the real sequence of doors, stairways etc. It must keep the distances between two subsequent elements as well, or the offset from the front. Using these lists, it is easy to reconstruct from a particular door and a particular orientation an instruction like *"go to the second door on your left"*. The main corridor in Fig. 1 could be partitioned into several of such sequences.

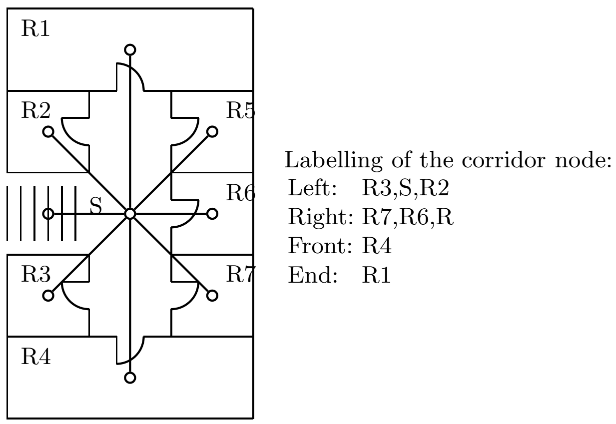

Fig. 6. Corridor

[4] in terms of storage
[5] in a way, the node is a dual to the linear structure

5 Related Work

The modelling of indoor environments involves various research areas. In the studies of Franz et al. [3], aspects of both architecture and cognitive science are investigated. Most importantly, the versatility of graphs as formal models of built spaces is pointed out. Beyond occupancy grid models used in artificial intelligence, place graphs and view graphs are concepts contributed by cognitive science. Visibility plays a major role in defining where a place ends and another one starts. The *space syntax*[6] is a method which covers three elementary aspects of wayfinding: Access graphs, axial maps and isovist (or visibility) fields. In the field of robot navigation, *topological* and *cognitive maps* [7, 2, 13] are common practice for indoor models. The hybrid approach presented by Kuipers et al. [7] creates large-scale topological maps from small-scale metrical maps (with a local reference system). Similarly do Broch et al. [2] create global maps in a bottom-up fashion.

Hierarchies in buildings where entities can be entered and left through exit points on the boundary were discussed by Hu and Lee [5]. These entities can be clustered. Particularly, it is emphasised that for indoor wayfinding, topological relations of regions should reflect *reachability semantics* rather than mere intersection. Jiang et al. [6] introduce a model with location identifiers for hierarchies. Approaches from cognitive science pursue a functional perspective in which the intrinsic, egocentric viewpoint of a human wayfinder is adopted, as opposed to a bird's eye view offered e.g. by maps. In order to describe indoor environments with their inherent hierarchy by cognitive elements, *image schemata* [9, 10] have been proposed. Examples include concepts like `CONTAINER`, `REGION`, `GATEWAY` and `PATH`, which are characterised by *affordances* (their specific function in social context), too.

Ontological aspects of indoor environments are covered e.g. by Bittner, Tsetsos et al. [1, 12]. Bittner [1] proposes a formal characterisation of built environments by *partitions* which represent approximate regions and their boundary relations. Tsetsos et al. [12] focus their attention on application aspects and enhanced personalised indoor routing. A strong point in Bittner's approach is the clear, unambiguous definition of regions by boundaries. Furthermore, a distinction between so-called *bona-fide* (hard) and *fiat* (soft) *boundaries* is given: Hard boundaries are impenetrable, tangible barriers (like walls or fences). In contrast, soft boundaries are sometimes invisible, nonetheless existent barriers which are often unconsciously passed by the wayfinder (e.g. a marked area on the ground of a subterranean garage which delimits a particular car-park).

6 Conclusion and Future Work

We have introduced a novel model for indoor environments. Not only does it support wayfinding algorithms, but also it supports the generation of human-understandable path descriptions. For this purpose it combines quantitative with

[6] `http://www.spacesyntax.org/`

qualitative spatial information into a hybrid structure. We have shown how angular and distal expressions enrich the basic graph structure and enable reasoning within a local frame of reference. This will be needed for generating route instructions like *"take the 2nd door to the right"*. Some scenarios can require quantitative information, for instance, when querying the total number of seats in a classroom or its capacity for storing goods.

The graphs presented in our model are hierarchical. Graphs at a lower level are refinements of nodes (or edges) at a higher level. The nodes and edges are of different types and carry different kinds of information, which is not solely the physical distance. The types reflect an ontology of rooms, buildings, and so on.

The graphs together with some of the standard algorithms, shortest paths, nearest neighbours etc., have been implemented in the prototypic `TransRoute` system [11]. In a next step we want to incorporate context information into the wayfinding algorithms. The context information can come from the environment itself (e.g. a door is closed at night), or it can date from user models (e.g. a preference for lifts instead of stairs).

For generating descriptions of paths, we are currently developing a markup language. The elements of this language describe operations for navigating indoor and outdoor environments.

Acknowledgements

This research has been co-funded by the European Commission and by the Swiss Federal Office for Education and Science within the 6th Framework Programme project REWERSE number 506779 (cf. http://rewerse.net).

References

1. Thomas Bittner. The Qualitative Structure of Built Environments. In *Fundamenta Informaticae*, volume 46 (nr. 1–2), pages 97–128, 2001.
2. Juan Carlos Peris Broch and Mara Teresa Escrig Monferrer. Cognitive Maps for Mobile Robot Navigation: A Hybrid Representation Using Reference Systems. In Michael Hofbaur, Bernhard Rinner, and Franz Wotawa, editors, *Proceedings of the 19th International Workshop on Qualitative Reasoning (QR-05)*, pages 179–186, 2005.
3. Gerald Franz, Hanspeter Mallot, and Jan Wiener. Graph-based Models of Space in Architecture and Cognitive Science - a Comparative Analysis. In Y-T. Leong and G.E. Lasker, editors, *Proceedings of the 17th International Conference on Systems Research, Informatics and Cybernetics (InterSymp'2005)*, Architecture, Engineering and Construction of Built Environments, pages 30–38. The International Institute for Advanced Studies in System Research and Cybernetics, 2005.
4. Christoph Hölscher, Tobias Meilinger, Georg Vrachliotis, and M. Knauff. The Floor Strategy: Wayfinding Cognition in a Multi-Level Building. In Akkelies van Nes, editor, *Proceedings of the 5th International Space Syntax Symposium*, volume 2, pages 823–824. Techne Press, June 2005.

5. Haibo Hu and Dik Lun Lee. Semantic Location Modeling for Location Navigation in Mobile Environment. In *Proceedings of the 5th IEEE International Conference on Mobile Data Management (MDM 2004)*, pages 52–61. IEEE Computer Society, 2004.
6. Changhao Jiang and Peter Steenkiste. A Hybrid Location Model with a Computable Location Identifier for Ubiquitous Computing. In Gaetano Borriello and Lars Erik Holmquist, editors, *Proceedings of the 4th International Conference on Ubiquitous Computing (UbiComp 2002)*, volume 2498 of *Lecture Notes in Computer Science*, pages 246–263. Springer, 2002.
7. Benjamin Kuipers, Joseph Modayil, Patrick Beeson, Matt MacMahon, and Francesco Savelli. Local metrical and global topological maps in the Hybrid Spatial Semantic Hierarchy. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation (ICRA 2004)*, volume 5 of *ICRA*, pages 4845–4851. IEEE Computer Society, 2004.
8. Kevin Lynch. *The Image of the City*. MIT Press, June 1960.
9. Martin Raubal and Michael Worboys. A Formal Model of the Process of Wayfinding in Built Environments. In Christian Freksa and David M. Mark, editors, *Spatial Information Theory: Cognitive and Computational Foundations of Geographic Information Science, Proceedings of the International Conference COSIT '99*, volume 1661 of *Lecture Notes in Computer Science*, pages 381–401. Springer, 1999.
10. Urs-Jakob Rüetschi and Sabine Timpf. Modelling Wayfinding in Public Transport: Network Space and Scene Space. In Christian Freksa et al., editor, *Spatial Cognition IV*, volume 3343 of *Lecture Notes in Artificial Intelligence*, pages 24–41. Springer, 2005.
11. Edgar-Philipp Stoffel. A Research Framework for Graph Theory in Routing Applications. Diplomarbeit/diploma thesis, Institute of Computer Science, LMU, Munich, 2005. http://www.pms.ifi.lmu.de/publikationen/#DA_Edgar.Stoffel.
12. Vassileios Tsetsos, Christos Anagnostopoulos, Panayiotis Kikiras, Tilemahos Hasiotis, and Stathes Hadjiefthymiades. A Human-centered Semantic Navigation System for Indoor Environments. In *Proceedings of the IEEE/ACS International Conference on Pervasive Services (ICPS'05)*, pages 146–155. IEEE Computer Society, 2005.
13. Jan Oliver Wallgrün. Hierarchical Voronoi-based Route Graph Representations for Planning, Spatial Reasoning, and Communication. In Patrick Doherty, editor, *Proceedings of the 4th International Cognitive Robotics Workshop (CogRob-2004)*, pages 64–69, 2004.