Topology Extraction from Occupancy Grids

Martin Werner

Abstract A fundamental problem in indoor location-based services is to compute the meaning of location with respect to an indoor location model. One specific challenge in this area is represented by the central tradeoff between two philosophies: a decent amount of the community tries to provide high-quality, high-fidelity models investing specialized knowledge and a lot of time in building such models for each building thereby increasing simplicity and quality of location-based services such as navigation or guidance. In contrast to that, other people argue that crowd sourcing and very simple representations of environmental information are the only way of generating indoor environmental information at scale. However, applications then have to tolerate errors and deal with oversimplified models. With this paper, we show for a specific widely accepted simple environmental model in which building floorplans are represented as black-and-white bitmaps, how we can provide algorithms for extracting higher order topological concepts from these trivial maps. We further illustrate how these can be applied to the hard problem of indoor shortest path calculation, indoor alternative path calculation, indoor spatial statistics, and path segmentation.

1 Introduction

Today, numerous location-based services enable new digital services and digital support for day-to-day life. Based on the wide availability of enabling technologies including GNSS positioning, satellite imagery, LiDAR scans, digital terrain models, and maps, many services have been developed especially for the outdoor space. The situation of Indoor Location-based services is, however, different (Werner 2014). First, it is not easy to derive a meaningful location from measurements of existing signals in the indoor area. Without investing much effort and money into a dedicated indoor location system, one is basically left with inertial sensory and signals that

---

M. Werner (✉)
German Aerospace Center (DLR), Remote Sensing Technology Institute (IMF),
Oberpfaffenhofen, 82234 Weling, Germany
e-mail: martin.werner@dlr.de

P. Kiefer et al. (eds.), *Progress in Location Based Services 2018*, Lecture Notes
in Geoinformation and Cartography, https://doi.org/10.1007/978-3-319-71470-7_7

have not been deployed with positioning in mind. This renders the indoor positioning problem highly ambiguous and challenging. Second, the acquisition of indoor location models is considerably harder as compared to the outdoor area. While for outdoors, a simple graph model in which edges represent ways and vertices represent corners is sufficient for navigation, indoor environments are more complex and cannot be represented by such a simple abstraction as a single graph. Instead, the full freedom of movement of humans should be taken into account. Based on this, a multitude of indoor location models has been defined ranging from a simple graph modelling movement over set-based and hybrid models to complex models based on occupancy grids or even 3D point clouds (Becker and Dürr 2005). The drawback of this wide range of modeling methodologies is that there is no good compromise to be reached: some of these models are extremely expressive and can be applied in any environment (e.g., 3D point clouds), but are difficult to exploit computationally. Other models are extremely efficient for computations (e.g., set-based models, graphs), however, are unable to express the full complexity of indoor navigation in a natural way. Additionally, the modeling effort varies greatly. Drawing new map information enhanced by a navigation graph consumes a lot of time. However, some sort of map information is often available: floorplans and building blueprints.

In my opinion, one must rely on techniques that are simple enough such that a majority of users understands these techniques and is able to model the environment themselves. One such model, though radical, is given by occupancy grids. An occupancy grid is a map representation scheme in which indoor spaces are first split into individual floors each of which is represented by a black-and-white floorplan bitmap. In this bitmap, black pixels are partially or fully obstructed and white pixel are walkable space. This model is simple enough such that even novice users are able to create, understand and modify models based on such information as this only needs basic image editing tools.

However, these representations have not been applied widely for complex queries about the environment. Instead, they are mainly used to find connectivity by calculating shortest paths in free space ignoring that they will scrape along walls and to filter erroneous sensor measurements in spatial particle filters. I believe that these simple models are more powerful and show that it is possible to automatically extract a lot of information from such occupancy grids, only.

The main contribution of this paper is the following: a method to extract higher order topological information from these maps in which spaces such as rooms and their interconnection (such as doors, hallways) are made explicitly available without user intervention. Additionally, it is shown how to use reduced topological maps for calculating shortest paths that do not scrape along geometry and, therefore, allow for better postprocessing when, e.g., identifying visible landmarks for route description. Third, the paper gives a scalable methods for extracting sets of sufficiently different short routes between two points, which applications can use in order to optimize complex additional criteria, e.g., find a sufficiently short path through the airport such that I am able to change money along the way.

Note that algorithms for selecting alternative routes in street networks are not applicable in this setting as there is a high number of similar paths sharing not a single edge, which would be seen as “perfect” alternatives given the criteria used for selecting alternative routes in street networks.

The remainder of the paper is structured as follows: Sect. 2 shortly reviews related work with respect to building modelling and information extraction. Section 3 introduces the contraction pyramid and explains its relation to the Reeb graph of the contraction process. Section 4 introduces illustratory applications including shortest path, alternative routes, Wi-Fi positioning as a classification problem, spatial statistics, and turn-by-turn guidance in buildings. Finally, Sect. 5 concludes the paper.

2 Related Work

Extracting environmental information is a fundamental prerequisite to many location-based services. When providing services based on the location of an entity, the meaning of this location to the application needs to be understood. Therefore, most location-based services are designed by setting the location of a mobile device into the context of an environment and only very simple, just about trivial, services can be provided without environmental information.

A very early work towards the representation of indoor spaces for location-based services has been presented by Becker and Dürr (2005). They distinguish a set-based model in which the subset relation, for example, models rooms being on a specific floor, graph-based models, in which graph edges model neighborhood relations and graph connectivity models space connectivity in a navigational sense, and present combinations of both. Another approach to indoor modeling is explicitly based on the two concepts of locations (e.g., rooms) and exits (e.g., doors) (Hu and Lee 2004).

In the last decades, several new approaches for scalable extraction of environmental information have been proposed. One direction is the automatic extraction of higher order topological information like rooms or doors from building blueprints (Werner and Kessel 2012), another direction of research is towards building a community of people explicitly modelling topological information in open data platforms such as OpenStreetMap (Openstreetmap wiki—indoor mapping 2016).

However, both approaches have not seen wide adoption yet: the complexities of extracting meaningful information from drawings for humans is not sufficiently solved and the communities have not agreed on a single, sufficient way to represent indoor spaces.

Another track of research originates in the robotics domain. Here, the environmental models are limited not by the limitations of algorithms, but rather by the limited ability of mobile robots to acquire and analyze data. In this area, two-dimensional occupancy grids are often created in which white pixels model free space and black pixels model occupied space. This concept of modelling spaces by what devices actually measure, is quite successful. However, much information about indoor spaces is lost. Recent representatives of this approach are given by high-end SLAM systems such as NavVis (NavVis corporation 2016), which generate very large amounts of three-dimensional point clouds.

This paper proposes a method to extract meaningful topological information from occupancy grid maps which can be created by mobile robots or modern measurement devices. The most similar work in literature with respect to our work is given by Fabrizi et al., who extract information topological relevant objects such as rooms and corridors as a computer vision task on fuzzy occupancy grids (Fabrizi and Saffiotti 2000). Due to the nature of their approach, this does not easily generalize to complex environments as it is based on morphology operations using structuring elements, which have a chosen size and shape. This size and shape determines a lot of properties of the output.

The approach provides a set of features implicitly describing the building topology, which can be used to extract a topological map of rooms similar to the approach of Fabrizi. Most importantly, it is not needed to set parameters related to the expected size of topological features as doors or hallways. Additionally, it is shown that the given features can be used for other tasks with respect to the environment such as calculating realistically shaped short paths and reasonable families of alternative paths through space.

3 The Contraction Pyramid

Indoor maps usually come in two different flavors depending on how they have been constructed. Either they are vector drawings or they are bitmaps and occupancy grids (Werner 2014).

Geographic Information Systems (GIS) drawings are usually created by architects and technicians to build, enhance, understand, or improve the building while user-generated or machine-generated maps are often in the form of occupancy grids either by creating such representations using sensory or by using bitmap manipulation software in order to create an occupancy grid map of a building from floorplans.

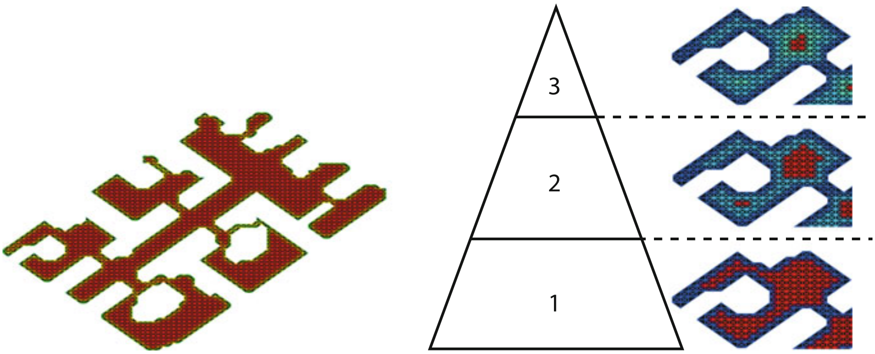
When GIS drawings of the building are available and of usable quality or if maps for indoor navigation are created using GIS software, the indoor maps are most often collections of two-dimensional drawings of primitives including lines, circles, and arcs. A specific set of symbols is being used to draw special objects such as doors, escalators, elevators, windows, and other buildings objects. However, these drawings are often unclear about some details of these building objects which have to be modelled manually (Werner and Kessel 2010). For topology detection in the sense of this paper, such GIS drawings have to be preprocessed in order to model walkable space as empty space—that is, given a starting point inside the building, a recursive eight corner graph traversal can walk the complete walkable space. This can be done either manually by deleting organizational lines and doors. However, it can be automated to a high degree (Werner and Kessel 2012). In fact, this preprocessing results in a GIS drawing, essentially a set of lines, such that the walkable space can be extracted as an occupancy grid. These occupancy grids are transformed into an eight corner navigation graph by creating a vertex per walkable pixel and eight edges connecting neighboring white pixels. Note that this map could have more than one connected component, especially, when some rooms can only be accessed via building objects such as elevators or staircases. For floorplans given as a bitmap, it is usually quite easy to modify the image to contain white pixels for free walkable space and black pixel for the building and the surroundings. Note that the three-dimensional connections are not to be considered in the context of topology extraction, as they connect different topological parts of the building. Thereby, it suffices to perform topology extraction and detection in 2D and handle three-dimensional connections as connecting different parts from different topological objects.

The concept proposed with this paper is based on the idea of using the vertex degree as an indicator of the border of free space components. The degree of a vertex in an undirected graph is defined to be the number of adjacent edges. Given an eight-corner-system navigation graph spanning walkable space, two different types of vertices exist: full degree vertices, which are called inner vertices and vertices with lower degrees, which are called border vertices. Border vertices are generated near obstructive geometry, which hinders the generation of some edges. The degree of a vertex is used as a color in Fig. 1a.

You can clearly see inner vertices of full degree depicted in red, border vertices in green or yellow depending on their number of neighbors.

The proposed approach proceeds in iterations. In an initialization step, the connected components of the given navigation graph are calculated and all vertices are labelled with a number representing their connected component.

After initializing the data structures, the approach removes all border vertices from the graph and then again calculate connected components. This gives a set of connected components each of which is fully contained in a connected component of the previous iteration. However, a connected component from the previous layer can split into more than one component in the current iteration.



(a) Degree of a navigation graph based on eight directions

(b) Contraction Pyramid

Fig. 1 A degree map and the contraction pyramid of a simple example

Considering the iteration number as a vertical coordinate, one can create a stack of shrinking and eventually splitting connected components similar to what we know from Morse theory or Reeb graphs. This leads to a geometric objects as depicted in Fig. 1b in which the lowest layer 0 is the full graph, layer 1 is given by removing all border vertices of layer 0, layer 2 is given by removing all border vertices from the graph at layer 1, and so on.

Connected regions will soon break into several disconnected regions when enough border vertices are removed, e.g., the graph shrinks and connected components start splitting. Considering the step from layer 1 to layer 2 in Fig. 1b, you see the red marked room splits into a small kernel down and a slightly larger patch up.

When a given connected component splits in parts, all vertices that have been removed in this step and are adjacent to the created connected components are called topological borders. The process ends when connected components are only represented by a set of border vertices, which would disappear in the next iteration. These vertices are called kernel of a room or hallway or otherwise relevant topological object. Now, these kernels are expanded into all directions until they meet with topological border vertices. These connected regions are called a topological room.

Figure 2 depicts the process for a larger map as well as a three-dimensional visualization of the different steps stacked one upon each other.

Again, the three-dimensional representation of the topology is based on stacking the same graph iteratively reduced by removing border vertices vertically on top of the previous graph. This transforms the two-dimensional occupancy grid graph into a three-dimensional object of finite height.

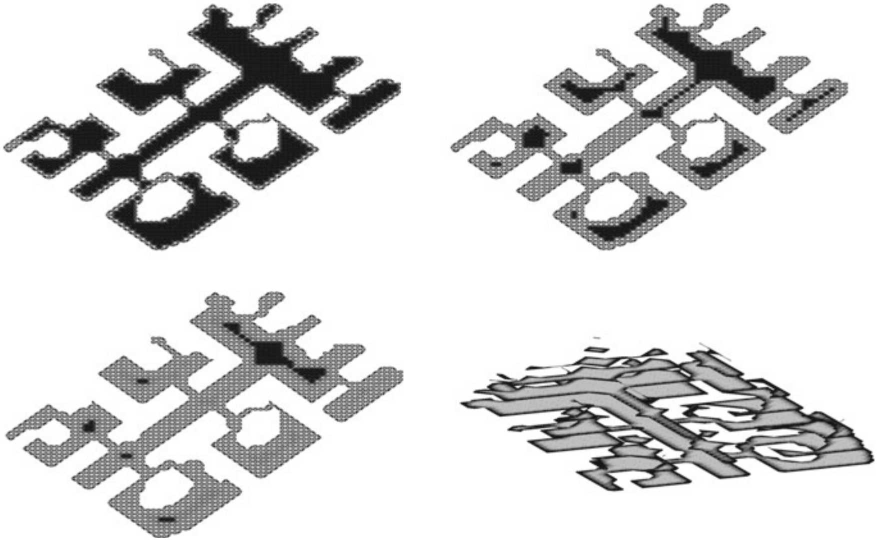


Fig. 2 The process of iterative shrinking and the pyramid as a stack of layers

From this object, one can vertically connect all vertices which are above each other and thereby form a full three-dimensional object. This motivates the following formal definitions:

Definition 1 The *height* of a vertex v representing a white pixel in an occupancy grid map is defined to be the number of layers of the pyramid in which this vertex exists.

In math, Morse theory as well as Reeb graph theory are conceptually similar. They use a concept of sweeping along the height of objects realizing their topological structure. The Reeb graph is very similar to our construction, however, with one more level of abstraction. The Reeb graph of a manifold is constructed by sweeping along one axis (let it be the vertical one) and adding a vertex if and only if a connected component appears, disappears, or splits. These three operations are represented by vertices in the Reeb graph. An edge in the Reeb graph means that some object (e.g., a smaller connected component) appeared from another component during a split. This Reeb graph is combinatorially equivalent to the information in our pyramid, however, purely combinatorial. Hence, there is no access to the vertices and height after creating the Reeb graph. With our pyramid, we are using a slightly larger graph as compared to the Reeb graph. But in a loose sense, the Reeb graph of the contraction pyramid taken as a threedimensional surface is similar to the following definitions extracting topological objects from the pyramid.

From the pyramid, one can as well derive some other objects describing a less local topological feature around a vertex.

Definition 2 The *threshold-connected component* of an occupancy grid vertex v and an integer threshold τ is defined to be the connected component of the layer of the pyramid at height τ.

An example of such a threshold-connected component is given by Fig. 1b. The red vertices on the second layer build, for example, three connected components. The sets of vertices of these components are the threshold-connected components at height τ. Similarly, in Fig. 2, the black connected region in the top left is for height $\tau = 1$, as you can see from the non-black border vertices. To the right, one finds a different set of components by reducing all these components.

Unfortunately, for a fixed height, not all interesting components are realized. Small components die out earlier than large components. So in higher heights, the small components are not visible anymore.

This motivates the following construction of maximal or complete components, which first needs the term of a kernel to be defined more clearly:

Definition 3 The *kernel component* of a vertex is the highest connected component from the pyramid. That is the highest set of vertices such that all of these vertices (including the vertex itself) would be removed in the next iteration.

This leads us to the definition of maximal component:

Definition 4 The set of *complete* or *maximal* components is extracted from a kernel by taking the largest (e.g., lowest) component in the pyramid that fully contains the kernel and does not split in the process.

In other words, a maximal component is the largest connected region that will not split by iterating the described algorithm. In the following sections, the usefulness of the pyramid, threshold-connected components, and maximal components is highlighted in various application scenarios from the domain of indoor location-based services. Note that we do not claim that any of those applications taken for themselves are new or extraordinarily innovative. The aim of the section is to show that the innovation of extracting a topological environmental model without a single value ϵ related to the expected size of topological features is expressive enough for these problems. More clearly, that there is a *single and simple* data structure represented by the contraction pyramid in which these operations can be done.

4 Applications

Understanding the building topology is a quite general and important task for indoor positioning systems. Consequently, there are very many application areas for automatically extracted topological information. The following applications may serve as a examples for the vast applicability of topology as extracted using this framework.

4.1 Topology-Aware Shortest Paths

The extrusion of the map space into a pyramid as explained before is a powerful concept. As a first tool, the height of a vertex can be used to support better shortest paths for visualization and computation along paths without severe additional overhead. One can compute the heights h_i of each vertex and scale each edge weight $\omega_{i,j}$ in the navigation graph by a factor:

$$\tilde{\omega}_{i,j} = \frac{1}{h_i + h_j}\omega_{i,j}$$

This makes edges between higher heights shorter and therefore leads to paths favoring a location for visualization and calculation inside free space as opposed to along walls and corners. Figure 3 depicts an example of a shortest path using the modified weights $\tilde{\omega}_{i,j}$.

As you can clearly see, the shortest path avoids scraping along walls and is a reasonable tradeoff between preferring high vertex edges and short paths. Furthermore, the height map can be used to classify the topology along the path directly. Figure 4a

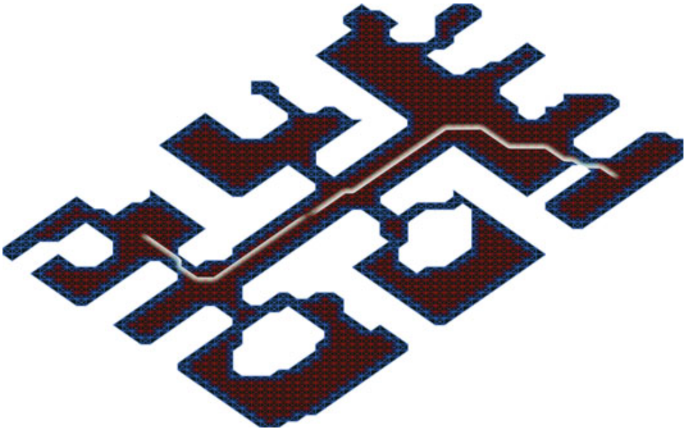
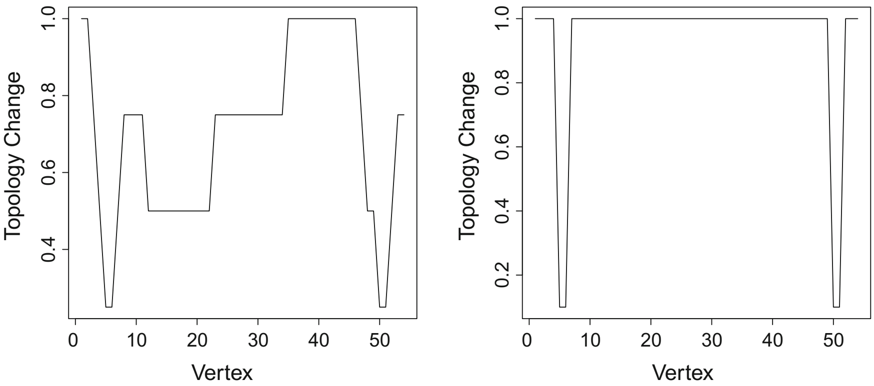


Fig. 3 A shortest path in the adapted weightmap keeping away from disturbing geometry



(a) Height (b) Binary Classification

Fig. 4 Height of vertices as observed along the shortest path

depicts the height as observed when following the shortest path depicted in Fig. 3 and the binary classification of this height.

In the first figure, we follow the depicted path from the left to the right. You can see that we are somehow inside a room, then leave the room through a bottleneck at vertices around eight. Then follow a path of varying height (e.g., a hallway) before we enter a larger space from vertex 35–48 followed by a bottleneck and entering another room. The binary classification on the right shows only the bottlenecks. These two figures can be used to create hints of interest for route description engines creating sentences like: "Start at the given location, then leave the room, follow a medium-sized hallway. When the hallway opens up widely, turn right and go through the door." Essentially, each arbitrary threshold θ on the height returns in a binary classification as depicted in Fig. 4b, which marks significant spatial events along the path with respect to the surroundings.

Additionally, note that computations along the path are more sensible as the path is more similar to the path a human would actually follow. If one wants to describe the shortest path in a navigation application, one likely wants to identify landmarks that are visible from the path. The visible space is, however, larger for the path preferring to stay away from obstacles and walls as much as sensible.

4.2 Topology-Aware Alternative Routes

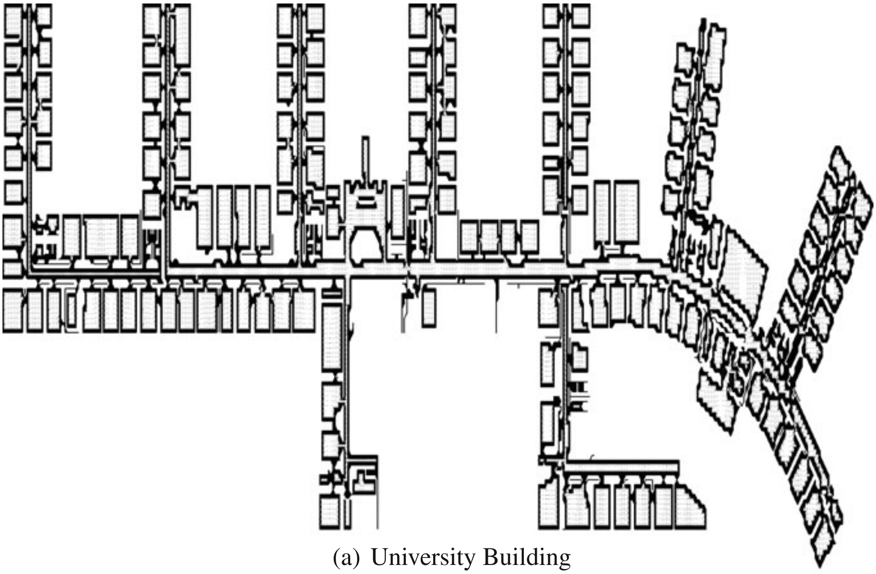
In the same situation as before, we might be interested in calculating alternative routes between two locations. Alternative routes are different routes between two locations that are reasonably short. It is quite complex to define and evaluate a good notion of different routes; the interested reader is referred to a definition of alternative routes in buildings (Werner and Feld 2014) and to general work regarding alternative routes (Dees et al. 2010; Bader et al. 2011).

Using the proposed pyramid, we can summarize routes between two locations by the set of labels of connected components the route visits. This is especially powerful, when a specific maximal height threshold is given. If we ignore all vertices whose height exceeds this given threshold, the connected components of this map can be labelled and two routes can be considered equivalent when they stride through the same sequence of connected components in this thresholded map. Figure 5 depicts such connected components for two example maps.

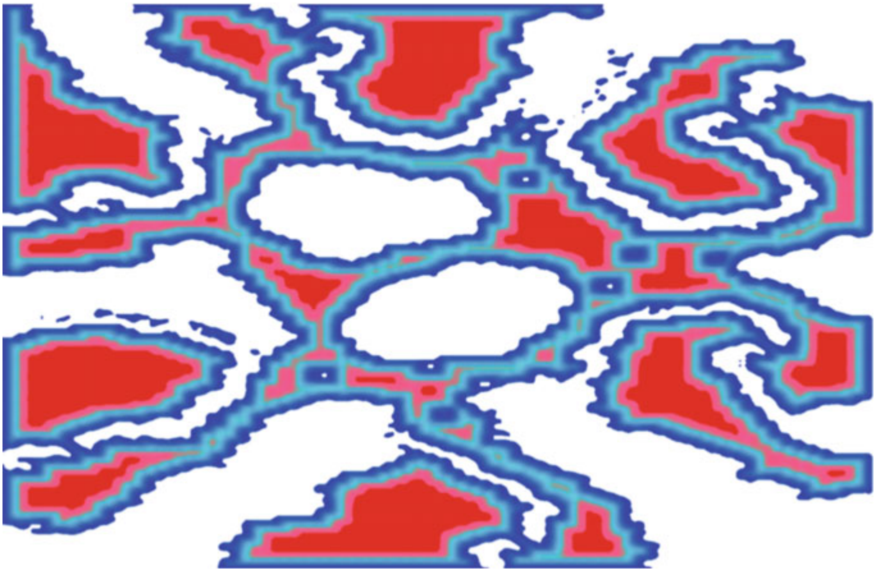
One can clearly see the rooms (and larger open spaces for the map taken from Starcraft) as red areas connected by blue corridors. The red areas are—for a constant threshold—the same as the connected components of the respective layer in the pyramid.

When it comes to the calculation of alternative routes, three important approaches can be identified. The most basic version of alternative route calculation is given by finding the top k shortest paths in a graph. This is relatively easy by first building a shortest path tree from the beginning and end of the intended route and then patching together a shortest path from the beginning, an edge (or a short sequence) that is not part of either shortest path tree, and the shortest path to the goal. Unfortunately, this approach requires calculation of two sufficiently complete shortest path trees which renders most optimizations of shortest path calculation useless and generates a huge amount of route candidates. Additionally, the patching nature of these routes makes many of those look quite unnatural.

There are two widely-used algorithms to reduce the amount of computation as well as the amount of candidates a bit. These two widely used algorithms can be enhanced by the extracted topology. One simple way to calculate alternative routes is given by the penalty method (Chen et al. 2007). This method is based on repeatedly calculating the shortest path between two points and increasing the weights of all edges on this shortest path by a certain amount. In this way, the shortest path gets



(a) University Building



(b) Computer Game Map

Fig. 5 Threshold connected components longer and alternative routes become shortest routes in the updated weighting. This creates a large candidate set of alternative routes of increasing length from which a reasonable set of alternatives has to be selected. The sets of routes generated in this way are by an order of magnitude smaller than the sets generated with the previously described sidestepping method. The reason is that partial overlap between routes is disfavored as all edges along a path get a penalty. Thereby, the candidates quickly start ranging forth and back over space. One problem of the penalty method is the fact that the candidates keep quite similar for a number of iterations and "good" alternatives are first found after many iterations.

To this end, one can use the topology to not only increase the weights of the shortest paths, but possibly all weights of edges inside the threshold connected components. In this way, it becomes more likely that the algorithm will avoid such a component in few iterations leading to quick identification of alternative routes crossing different threshold connected components. Essentially, this behaves like a graph compression in which threshold-connected components behave like a single edge with respect to the penalty algorithm.

The second widely used algorithm for detecting alternative routes is given by the plateau algorithm (CVIT Ltd. 2016). It is based on building shortest path trees from start and end vertex simultaneously with optimizations and identifying path segments in which both shortest path trees overlap. For each such overlap, the candidate alternative route is extracted by routing from the start to the overlapping region and towards the end. Once overlap is detected, this is possible in constant time. As compared to the penalty algorithm, the plateau algorithm is able to quickly detect alternatives from a larger spatial area due to the uniform growing of shortest path trees. However, this is also the most important downside of this approach: the shortest path trees should be optimized to avoid explosion and quickly generate a solution but still have to cover a decent amount of space in order to find these overlapping regions. Usually, both shortest path trees are pruned at a fixed multiple of the distance between the start and end vertex by exploiting the triangle inequality. Note that this method is similar to the sidestepping method with a heuristic on which sidesteps to use first.

This algorithm can also be augmented by extracted topological information: the shortest path trees might not overlap too much as most search algorithms tend to keep left or right during expansion in free space. Therefore, quite unnatural overlapping regions will be generated and candidates for alternative paths will be counterintuitively irregular. By using a height-scaled weightmap, however, the forward and backward search are pulled towards the areas of high height vertices leading to collisions of both search trees and reasonable overlapping regions at high heights and high quality candidates. Again, this can be seen as a graph compression in which the connectivity of the graph is summarized by the high height vertices that are locally maximal. That is, as shortest paths in forward and backward search are pulled towards the same local maxima of height, they will collide there and each such local maximum (or plateau of such locally maximal vertices) represents the "kernel" of a threshold-connected component.

4.3 Selection of Alternative Routes

As already mentioned in the previous section, the most puzzling question for selecting alternative routes is a measure of alternativity of candidates.

It is quite easy to calculate very large sets of different routes between two points. As an easy approach, one calculates two shortest path trees, one from the start and one from the end vertex in the reverse graph. Then, each edge not in the shortest path trees creates an "alternative" by first going a shortset path from the start to this edge, then along this edge, and finally to the goal on a shortest path. However, these large sets of routes are not very useful for applications.

With the extracted topology, one can create a family of "alternative" routes, possibly even generated by mobile devices. Then, one can use the set of height components, these routes traverse, to select one and only one, e.g., the shortest, for each of these sets. Figure 6 depicts a result of selecting only one alternative route from a search based on the penalty algorithm in which two "alternatives" are considered equivalent, if they cross the same set of labels using a fixed height threshold. One clearly sees that only a limited set of routes is generated without much overlap. Hence, a "good" set of alternative routes: it is small subset of alternative route candidates, but covers many sensible examples. It is much more restrictive to use maximal connected components. Figure 7 depicts the filtering result in which maximal components have been used to define equivalence. This generates another, less complete, but smaller set of truly alternative routes between the two points.

It depends on the application, how much filtering of candidates is reasonable. Note that it is even possible to not only use set equality in filtering (reject candidates that are equal) but also to use set similarity provided for example by the Jaccard index for a fine-grained adjustment of the result set size and variety.

4.4 Indoor Spatial Statistics

When collecting large amounts of spatial data, several analytic approaches have been very successful. One elementary analysis for spatial and spatiotemporal datasets is given by the Getis-Ord G_i^* statistics. Basically, this statistic is based on comparing feature values of spatial cells (e.g., the number of events in a specific region) with those feature values of their neighbors identifying hotspots, i.e., locations, where this feature value is significantly larger than expected. Therefore, this statistics compares sums of features of regions and neighbors with the globally expected value for these sums.

The regions of Getis-Ord are often calculated by first aggregating feature values on a grid and then using the neighborhood relation on grid cells in order to find hotspots. However, when thinking about meaningful hotspots in buildings, this will be misleading.

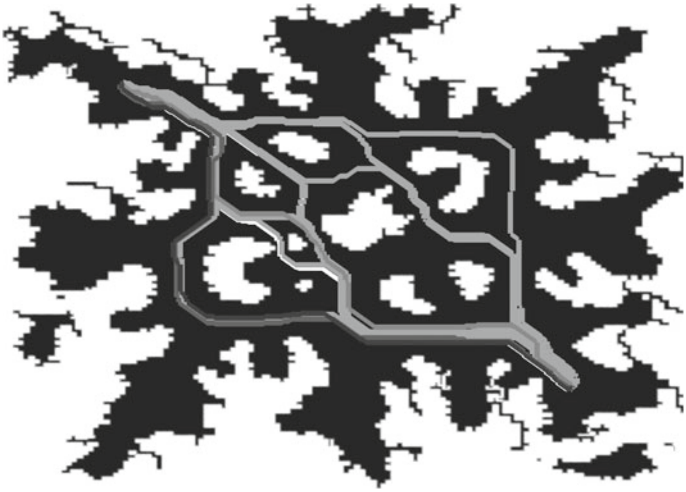


Fig. 6 Alternative routes selected using height components

Fig. 7 Alternative routes selected using maximal height components

Assume, we are able to count the number of people in every room of the building with some sensors. A hotspot in this context should be a room in which many more people reside than one should expect from the average number of persons in a room. The grid-based approach, however, would detect people density as hotspots, that is rooms in which many people are near each other—a completely different question.

In this context, I propose to use the topological subdivision as provided by complete or threshold-connected components based on height in order to analyze hotspots in a topologically correct way. Consider Fig. 5a for an example of the spatial splitting using complete height components. Again, the most useful fact is here that the topology does not need to be generated by hand, but is extracted automatically from connectivity and does not depend on choosing a suitable threshold as in related work.

This approach can easily be extended to other spatial statistics methods in which spatial divisions have to be chosen including Moran's *I* and Geary's *C* for spatial autocorrelation.

4.5 Turn-by-Turn Guidance in Buildings

One of the hardest unsolved problems in indoor pervasive computing might be the computational generation of descriptions of movements and paths, that is a suitable analog of turn-by-turn guidance (Chewar and McCrickard 2002; Raubal and Winter 2002). In this area, two aspects interfere making a solution to the problem extremely hard:

The first aspect is about the availability of suitable map information needed to describe ways. This includes information about landmarks as well as topological information similar to what is extracted in this paper.

The second aspect is the unavailability of continuous user interfaces as well as the inability of humans to measure or estimate distances and to remember large sets of instructions. It has been discussed that the optimal number of instructions for describing a route through an airport would be roughly five (Ruppel et al. 2009). If users are given more than that, they tend to forget or—even worse—to confuse instructions.

In this context, the challenge is to generate descriptions with few instructions, the number of instructions independent from the length and complexity of the route, and still comprehensible for humans.

With respect to this problem, we envision a system that calculates and readily explains shortest paths in buildings in order to make an audio guide through buildings feasible and increase memorability of navigation instructions.

The hierarchical structure of the topology represented by the pyramid can be used to split any shortest path into flexible numbers of subpaths by increasing the height in the pyramid. While the path crosses one component on the base layer, this component splits several times and we expect these splittings to induce useful information for textual instruction generation. This can be seen as a Morse theory perspective on the navigation space: by increasing height, one can increase the number of components step by step and select a height in which the number of components is suitable for generating few instructions.

Additionally, as depicted in Fig. 4a, the height along the shortest path is full of information about the direct surroundings such as when the path is crossing small space such as doors or large spaces such as halls.

Furthermore, there is evidence that self-localization in large environments is better performed on schematic maps, while detailed information is better to be extracted from detailed floorplans (Meilinger et al. 2006). Such schematic maps can be constructed from our framework, for example, using the complete height components and highlighting all components crossed by the shortest path to be visually described allowing the mobile user to ignore irrelevant parts of the floorplan.

By fusing this information with information from other sources, I conclude that a lot of information for the optimization of textual instruction generation and shortest path visualization is made available. However, a deep investigation of this approach including a usability analysis is beyond the scope of this paper.

5 Conclusion

This paper has shown how to use an occupancy grid map in order to understand how the building topology is split into smaller pieces such as rooms and hallways. Additionally, it has shown the impact of observing the height in the contraction pyramid as a simple and powerful topological feature in order to annotate ways, calculate alternatives, and visualize shortest paths in buildings.

Additionally, a splitting of the topology into neighboring objects with sensible spatial extent allows for the application of spatial statistics such as the Getis-Ord hotspot statistics and similar spatio-temporal tools in indoor situations. This has not been the case without manually creating sensible neighborhoods which is a tedious, time-consuming and error-prone task.

Furthermore, we motivate another direction of applications in the area of compressed descriptions of complex paths through buildings. This is an area of ongoing research and we plan to explore this direction in future work.

References

Bader R, Dees J, Geisberger R, Sanders P (2011) Alternative route graphs in road networks. In: Theory and practice of algorithms in (computer) systems. Springer, pp 21–32

Becker C, Dürr F (2005) On location models for ubiquitous computing. Pers Ubiquitous Comput 9(1):20–31

Chen Y, Bell MG, Bogenberger K (2007) Reliable pretrip multipath planning and dynamic adaptation for a centralized road navigation system. IEEE Trans Intell Transp Syst 8(1):14–20

Chewar C, McCrickard DS (2002) Dynamic route descriptions: tradeoffs by usage goals and user characteristics. In: Proceedings of the 2nd international symposium on Smart graphics. ACM, pp 71–78

CVIT Ltd. (2016) Choice routing. http://www.camvit.com/camvit-technical-english/Camvit-Choice-Routing-Explanation-english.pdf

Dees J, Geisberger R, Sanders P, Bader R (2010) Defining and computing alternative routes in road networks. arXiv:10024330

Fabrizi E, Saffiotti A (2000) Extracting topology-based maps from gridmaps. In: IEEE international conference on robotics and automation, 2000. Proceedings. ICRA'00, vol 3. IEEE, pp 2972–2978

Hu H, Lee DL (2004) Semantic location modeling for location navigation in mobile environment. In: 2004 IEEE international conference on mobile data management, 2004. Proceedings. IEEE, pp 52–61

Meilinger T, Hölscher C, Büchner SJ, Brösamle M (2006) How much information do you need? schematic maps in wayfinding and self localisation. Spatial cognition V reasoning, action, interaction. Springer, pp 381–400

NavVis corporation (2016). http://www.navvis.com/

Openstreetmap wiki—indoor mapping (2016). http://wiki.openstreetmap.org/wiki/Indoor_Mapping#Tagging_proposals

Raubal M, Winter S (2002) Enriching wayfinding instructions with local landmarks. Springer

Ruppel P, Gschwandtner F, Schindhelm CK, Linnhoff-Popien C (2009) Indoor navigation on distributed stationary display systems. In: 33rd annual IEEE international computer software and applications conference, 2009. COMPSAC'09, vol 1. IEEE, pp 37–44

Werner M (2014) Indoor location-based services: prerequisites and foundations. Springer

Werner M, Feld S (2014) Homotopy and alternative routes in indoor navigation scenarios. In: Proceedings of the 5th international conference on indoor positioning and indoor navigation (IPIN 2014)

Werner M, Kessel M (2010) Organisation of Indoor Navigation data from a data query perspective. Ubiquitous positioning indoor navigation and location based service (UPINLBS), pp 1–6

Werner M, Kessel M (2012) A bitmap-centric environmental model for mobile navigation inside buildings. Int J Adv Netw Serv 5(1 and 2):91–101