Criteria for Selecting Small Sets of Alternative Routes in Constrained Free Space Scenarios

Sebastian Feld, Martin Werner, Claudia Linnhoff-Popien

Mobile and Distributed Systems Group, LMU Munich, Munich, Germany
sebastian.feld@ifi.lmu.de, martin.werner@ifi.lmu.de, linnhoff@ifi.lmu.de

Abstract. Recently, alternative routes have gained momentum in the creation of Location-Based Services. This paper gathers and sorts existing work regarding quality metrics of alternative routes and alternative graphs in road networks and discusses their commonalities. Based on this, the paper clarifies what challenges need to be tackled in order to create such metrics for constrained free space scenarios and discusses possible courses of action, opportunities, and limitations. The general goal of this paper is to stimulate the discussion on and the development of quality metrics for alternative routes and alternative graphs for constrained free spaces like pedestrian navigation, or even maritime or aviation scenarios.

Keywords. Alternative Routes, Alternative Graphs, Quality Metrics, Constrained Free Space, Indoor Navigation

1. Introduction

The technical progress in reducing the physical size of processing power, storage, and connectivity supports the enlarged spread of Location-Based Services (LBS) not only on powerful mobile devices like smartphones, but also on mobile robots or in distributed sensor networks. The importance of single computing devices vanishes and leads to an active construction of ubiquitous computing and the Internet of Things. Navigation is surely a central topic in LBS consisting, amongst others, of positioning, path finding, path representation, and (interactive) guidance. The calculation of a shortest path between two points in a street network is one of the most famous applications.

Routing in street networks is usually handled using a directed graph where edges represent streets and interconnections of streets represent nodes.

LBS 2016
13th Conference on Location-Based Services
Vienna, 14–16 November 2016

Published in "Proceedings of the 13th International Conference on Location-Based Services", edited by Georg Gartner and Haosheng Huang, LBS 2016, 14-16 November 2016, Vienna, Austria.

Another field of research and application is routing in constrained free space scenarios, i.e., wayfinding for subjects that can freely operate in a scenario like pedestrians in large buildings or characters in computer games. The main question is to find an appropriate map representation that handles the tradeoff between the map's simplicity and expressive power. Basically, the simpler the map the faster the algorithms can perform. On the other hand the scenario may suffer from missing details or imprecisions. There exist numerous considerations on how to create simple maps out of complex maps, consider topology-aware shape simplification, for example.

The calculation of shortest paths is widely investigated leading to speed-up techniques that enable continental-sized path finding in real-time applications. A further, much less examined, field of research is the question of how to calculate alternative routes and the subsequent creation of an alternative graph. An alternative graph is the aggregation of several alternative routes into one representation. Often, there are several highly different ways to traverse a map from a given start and to a goal that are nearly as optimal as the shortest path. Depending on the use case, the optimal path may not match the personal preferences of the concrete user, or the preferences may be hard to obtain or calculate. They may depend on local knowledge (Abraham et al., 2013), or contextual information like toll pricing, scenic value, fuel consumption, or the risk of traffic jams (Bader et al., 2011). Thus, true to the slogan “Human-in-the-loop”, we want to compute a set of good alternatives in order to let the user choose based on personal needs. Besides that, there are motivations for alternative routes for non-human users, i.e. robots, or as a background service: multiple mobile robots get different routes from start to goal to increase the chance that at least one robot will find the goal; proactive avoidance of bottlenecks; offline calculation of pathways for action forces like firefighters; non-player characters in computer games that tactically adapt their travel path. The task of calculating alternative routes and alternative graphs is quite difficult since there is no clear definition of what represents a good alternative; the topic is very subjective, almost philosophical.

Testing the quality of possible alternative routes and alternative graphs is not trivial and – at least for street network scenarios – in the focus of many researchers (Kobitzsch, 2013). There is much work on alternative routes and alternative graphs in streets networks, but not for constrained free space scenarios like navigation in buildings, maritime navigation in harbors, pedestrian navigation, or airplane navigation. With this paper we want to close the gap and discuss the reasons why the proposed solutions for outdoor scenarios cannot directly be transferred to constrained free space scenarios. Our goal is to stimulate research and discussion on measures and target functions for comparing sets of routes in said constrained free space scenarios.

The remainder of this paper is structured as follows: After introducing the motivation of this paper, Section 2 introduces needed background including map representations, algorithms for calculating alternative routes, and existing quality metrics for alternative routes and alternative graphs in street networks. Section 3 discusses the transferability of the measures to free space scenarios and their mutual interference. Section 4 concludes the paper.

2. Background

This Section deals with the presentation of different ways to represent constrained free space scenarios and algorithms for calculating alternative routes. Based on this, existing measures for calculating the quality of alternative routes and alternative graphs in street networks are discussed.

2.1. Map Representations

Street networks form a graph structure in a quite natural way: streets are modelled as sequences of linear segments and crossings of streets as well as turning points inside a street are usually becoming a vertex of the graph. Then, the graph is an embedded and high-quality abstraction of the real abilities of movement in a street network.

For constrained free space, the situation is different. First of all, movements in all directions are possible. However, we might have some constraints regarding the possible movements, e.g., a mobile robot may only change movement by steering leading to smooth trajectories with bounded curvature. Dealing with such complicated objects is computationally demanding and often intractable. Therefore, trajectories are often modelled as polylines, that is, by points connected by linear segments. In the sequel, we will always assume that routes are given as polygonal lines and not as some parametrized curves.

Still, dealing with polylines in free space is difficult. An example is the optimization over all possible polylines for finding the shortest connection between two points. At this point, map representations come into play, which effectively reduce the space of possible movements for more efficient planning. Map representations involve the tradeoff between the map representation's ability to describe any given path in the building and the computational complexity of algorithms performing choices in the environmental model.

Recently, relatively free movement is being planned in the field of continuous planning. More classically, however, synthetic graphs are being used such that the paths through the graph represent possibly large sets of movements through constrained free space.

A very simple form of map representation is given by constrained grids. In this map representation, one chooses an arbitrary length and a regular grid which is then put over the map. The points of the grid represent vertices and two adjacent points in the grid are connected if and only if they are connected by free space. In occupancy grids, for example, a two-dimensional bitmap is being used to represent walkable space in one color (e.g., white) and obstacles in another color (e.g., black). Then, each pixel creates a vertex and two pixels are connected in the graph if they are both white and direct neighbors. At a first glance, grids are large and inefficient; however, as computers are actually able to manipulate large bitmaps, they can be very efficient in practice. Additionally, the graphs have bounded degree of 4 when connecting vertical and horizontal neighbors, or 8 including diagonals. Finally, the algorithms' performance depends only on the amount of free space and not on the complexity of involved geometry.

A second form of creating a graph representation of constrained free space is given by polygonal maps. For polygonal maps, the constraints are modelled as polygons and a graph is created by using vertices and edges of the polygons. There are several constructions for maps given as sets of such obstructive polygons. If, for example, all vertices of these polygons are connected as long as the direct line between them does only cross free space, we call it visibility graph. This representation has some good properties: for example, it can be relatively small for few and simple polygons and it contains all shortest paths between vertices of the involved polygons. However, it is not obvious, how this graph can be used for motion planning starting in arbitrary locations in free space and shortest paths in this graph tend to scrape along walls or walk in diagonals.

Another approach called navigation mesh is given by using a polygonal tessellation of free space and adding edges to a graph for each edge shared by two polygons meaning that one could go from one polygon to the other. In such constructions, each polygon represents some space and simple polygons have the advantage of having low degree in the resulting graph. Therefore, navigation meshes are often built from very simple polygons such as triangles or rectangles. In a sense, the grid-based approach is also a navigation mesh in which small and regular squares are being used for the polygons. There are some choices of building a graph representation from a navigation mesh: one could just represent each polygon as a vertex and connect neighboring vertices. When the polygons are convex, one often uses centers of the polygons' edges as navigation points leading to slightly better paths as they become embeddable into free space. However, the resulting paths often have an unusual shape. A third way of extracting a navigation graph from a navigation mesh is by traversing edges of the involved polygons. This leads, again, to wall-scraping behavior, which can have positive impact (e.g., knowing which obstacles are being passed) or negative impact (e.g., not being able to directly visualize the path) depending on the application.

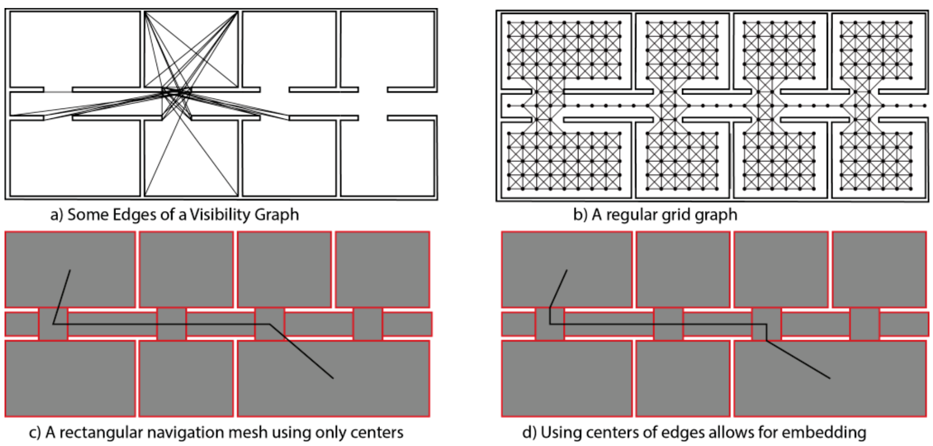

Figure 1. Examples of map representations, adapted from (Werner, 2014).

Figure 1 depicts examples for the three basic approaches in an indoor scenario. *Figure 1a* shows that the amount of edges in visibility graphs can get high very quick and that the average degree of vertices will be high, too. This effect is even worse, when GIS systems are being used in which round objects are tessellated into many line segments. A circular pillar can easily generate hundreds of additional lines. In contrast to that, a regular grid graph is depicted in *Figure 1b*. It has low-degree vertices, however, there are many of them. *Figure 1c* and *Figure 1d* depict a rectangular navigation mesh. In *Figure 1c* the shortest path is calculated only on the rectangles' center points leading to the situation that the lower-right line segment is not fully inside free space. In *Figure 1d*, however, the shortest path remains in free space due to the use of the middle points of the rectangle's edges and also the rectangles' convexity.

These basic approaches are being used to make path planning in constrained free space tractable and simple. However, they all have in common that their scale is limited. In this case, space subdivisions are commonly used to combine several "local" models in a general view. Usually, space is subdivided recursively (e.g., Binary Space Partitioning) or in a regular way (e.g., Grids) and local models can be combined to models that are small, but large enough to contain all relevant geometry.

Finally, reducing the combinatorial complexity of free space introduces inaccuracies as not all movements are being represented. From an observer perspective, one has to discuss, how a given trajectory is represented in the graph, and, from a planning perspective, one should consider post-processing to alleviate artifacts like diagonals, wall-scraping, and detours.

2.2. Algorithms for Calculating Alternative Routes

Finding alternative routes relies on the problem of calculating a single shortest path between two given points. Starting with Dijkstra's algorithm from the year 1959 (Dijkstra, 1959), there are considerable extensions and speed-ups, see the survey (Bast et al., 2015), for example.

The next step is to calculate not a single shortest path, but multiple routes. The class of K-shortest-path algorithms (Eppstein, 1994, Yen, 1971) seem to solve the problem but they do not, since, in general, the resulting paths do not change significantly until a high value of k. An extension is to calculate k disjoint paths (Scott, 1997). That in turn introduces the notion of overlap, a very important term in the further course of this paper. Currently, there do not exist suitable implementations to efficiently enumerate the paths for continental sized road networks.

Another class of algorithms is based on multicriteria optimization, i.e the use of multiple edge weights like distance, travel time, or fuel cost. The idea is to calculate so-called Pareto-optimal paths, i.e. paths that have the lowest edge weights for at least one criterion (Martins, 1984, Delling & Wagner, 2009, Geisberger et al., 2010, Graf et al., 2010).

The penalty algorithm tries to compute alternative routes by iteratively calculating the shortest path and increasing certain edge weights (Chen et al., 2007). There are several improvements of the algorithm together with the use for generating alternative graphs (Bader et al., 2011).

Finally, there is a class of algorithms for creating alternative routes that uses via-nodes or plateaus (Abraham et al., 2010, Bader et al., 2011, Kobitzsch, 2013, Luxen & Schieferdecker, 2102, Werner & Feld, 2014). The basic idea is to concatenate a shortest path from a start to a supporting via-node with a shortest path from the said node to the goal. Additionally, there is a proprietary algorithm called choice routing by Camvit (Camvit, 2009).

2.3. Quality Metrics for Alternative Routes

The central reference for quality metrics for alternative routes is (Abraham et al., 2013), a paper that extends its predecessor (Abraham et al., 2010) by various extensions and an elaborate evaluation. The authors try to find good alternative routes and define an “admissible path” as a path that is substantially different to the reference path, not much longer than that, and that is natural without unnecessary detours. The just stated three properties have been defined formally and will be described successive:

1. **Limited Sharing**: The alternative path has to be significantly different to the reference path, i.e. the total length of the edges they share must be a small fraction of the reference route’s length.
2. **Local Optimality**: The alternative path must be reasonable, i.e. no unnecessary detours are allowed. Every local decision must make sense, so local optimality is given if every subpath up to a certain length is a shortest path.
3. **Uniformly Bounded Stretch**: The alternative path must not be much longer than the reference path, i.e. every subpath needs to have a good stretch. This condition also enhances local optimality, since there may be the case that a path has good high optimality, but a shortcut could skip an unnecessary part of the route.

Based on these prosaic descriptions, Abraham et al. formally define the class of paths to be found as “admissible alternative paths”. Let $G = (V, E)$ be a directed graph with nonnegative edge weights, with $|V| = n$ being the number of nodes and $|E| = m$ the number of edges. Given a path P in G, $|P|$ is the number of the path’s edges and $l(P)$ is the sum of the edge weights. Furthermore, $l(P \cap Q)$ is the sum of the edge weights shared by paths P and Q, and $l(P \setminus Q)$ is $l(P) - l(P \cap Q)$. Given two vertices, s and t, finding the shortest path – denoted by $Opt(s, t = Opt)$ – is called the point-to-point shortest path problem. Given three tuning parameters $0 < \alpha < 1$, $\epsilon \leq 0$, and $0 \leq \gamma \leq 1$, and a shortest path Opt between s and t, a s-t-path P is an admissible alternative if the following criteria are fulfilled:

1. **Limited Sharing**: $l(Opt \cap P) \leq \gamma \cdot l(Opt)$
2. **Local Optimality**: P is T-locally optimal for $T = \alpha \cdot l(Opt)$. A path P is T-locally optimal if every subpath P' of P with $l(P') \leq T$ is a shortest path
3. **Uniformly Bounded Stretch (UBS)**: P is $(1 + \epsilon)$-UBS. A path P has $(1 + \epsilon)$-UBS if for every subpath P' of P with end points s', t', the inequality $l(P') \leq (1 + \epsilon) \cdot l\big(Opt(s', t')\big)$ holds

Due to the fact that too many admissible paths can be found, Abraham et al. introduce a limited yet useful subset of the class “admissible alternative paths", namely "single via paths": Given a start s, a goal t, and a via-node v, a via path P_v is the concatenation of the shortest path from s to v and the shortest path from v to t. This definition is an unnecessary restriction, as Bader et al. (2013) will work out later on, but such routes have interesting properties. P_v has got the lowest stretch of all routes going through node v, and, more importantly, the local optimality can be violated just around node v. This fact will be used, amongst others, for further improvements of the conditions (Abraham et al., 2013).

The conditions just described are used as hard constraints for a not further defined target function $f(\cdot)$. This function is used to sort candidates and return the first admissible path. A possible target function may seek for routes having low "limited sharing", high "local optimality", and low "uniformly bounded stretch". If the use case is to find multiple alternative routes, then just "limited sharing" needs to be calculated regarding the shortest path and all alternative routes already found. Both, "local optimality" and "uniformly bounded stretch" refer to only the path in question.

(Luxen & Schieferdecker, 2012) present an improvement of the algorithm of Abraham et al. regarding query times. The authors propose a fast algorithm that stores a small precomputed set of via-nodes for pairs of regions within the graph, i.e. they focus on a small candidate set to be tested efficiently. They base their via-node routing on top of Contraction Hierarchies (Geisberger et al., 2009, Bauer et al., 2010) and achieve routes with higher quality with faster query times while having negligible memory overhead.

Another improvement is done in (Kobitzsch, 2013), where the author motivates the contribution with the fact, that the selection process of the known via-node algorithm is unsatisfying, since testing all potential candidates proves expensive due to many shortest path queries. Existing algorithms order the candidates heuristically with potentially better candidates being discarded. Kobitzsch uses a different approach by making the viability check fast enough to test all potential candidates. Thus, the problem is reduced to a small graph representing all potentially viable alternative paths.

2.4. Quality Metrics for Alternative Graphs

The central reference for quality metrics for alternative graphs is (Bader et al., 2011), a paper that is based on Dees' master's thesis (Dees, 2010). Besides that, preliminary aspects have been published before in (Dees et al., 2010). The main concept is to compute a set of alternative routes which, in general, can share nodes and edges, and subpaths of the alternative routes can potentially be combined to new alternative routes. Thus, Bader et al. motivate and define the concept of an alternative graph (AG) as the union of several paths having the same start and goal as a compact representation of multiple alternative routes. They define several attributes quantifying the quality of an alternative graph using mathematical definitions of the graph structure and show that it is already NP-hard to optimize a simple objective function combining just two of the proposed attributes and therefore turn to heuristics. The three measures are described as follows:

1. **Total Distance**: This measure describes the extent to which the routes defined by the AG are non-overlapping. The maximum value is reached when the AG consists of disjoint paths only. The mathematical definition of total distance will need a scaling, since otherwise long non-optimal paths would be encouraged.
2. **Average Distance**: This measure describes the path quality as the average stretch on an alternative path. The mathematical definition will need an averaging in order to avoid a high weight to large numbers of alternative paths that are all very similar.
3. **Decision Edges**: This measure describes the complexity of the AG and is used to retain the representation easily understandable for human users.

After depicting the measures prosaically, we turn to the formal definitions. Let $G = (V, E)$ be a graph with an edge weight function $w: E \to \mathbb{R}_+$. Given a source node s and a target node t, an AG $H = (V', E')$ is a graph with $V' \subseteq V$ such that for every edge $e \in E'$ there exists a simple s-t-path in H containing e. For every edge (u, v) in E' there must be a path from u to v in G and the edge weight $w(u, v)$ must be equal to the path's weight. $d_G(u, v)$ denotes the shortest path distance from u to v in G, analog $d_H(u, v)$ is the shortest path distance from u to v in H. Based on that, the formal definition of the quality metrics for a given alternative graph $H = (V', E')$ is:

Total Distance: $\displaystyle\sum_{e=(u,v)\in E'} \frac{w(e)}{d_H(s,u) + w(e) + d_H(v,t)}$

Average Distance: $\displaystyle\frac{\sum_{e\in E'} w(e)}{d_G(s,t) \cdot totalDistance}$

Decision Edges: $\displaystyle\sum_{v\in V'\setminus\{t\}} outdegree(v) - 1$

As already stated above, it is NP-hard to optimize a reasonable combination of the measures just explained, thus Bader et al. use heuristics to compute an AG. They calculate a shortest path, insert it into the AG, gradually calculate further alternative paths, and insert them greedily into the AG regarding the optimization of a target function.

(Radermacher, 2012) is a bachelor's thesis that efficiently implements the concept of (Bader et al., 2011). Basically, Radermacher combines a Multi-Level-Dijkstra (Delling et al., 2011) with the penalty method that is augmented with path analysis.

(Kobitzsch et al., 2013) present a viable implementation of (Bader et al., 2011) such that it can be used interactively. They modify the penalty algorithm through a multi-level partitioning together with the penalization scheme for the route and its adjacent edges. Furthermore, they modify the function that tests if a candidate is feasible and introduce further speed-ups using Customizable Route Planning (Delling et al., 2011, Delling & Werneck, 2013) and Dynamic Level Selection. Kobitzsch et al. state that (Bader et al., 2011) compute up to 20 paths and perform a selection based on priority terms afterwards. Since Kobitzsch et al. focus on query times, they take a different approach by considering the potential value to the alternative graph. Thus, a path must offer at least one deviation of a certain length, and the detours are checked for their stretch.

At the same time, (Paraskevopoulos & Zaroliagis, 2013) has been published proposing an algorithm that creates alternative graphs with higher quality. They suggest a pruning stage preceding the heuristic method for finding alternative paths, and introduce filtering and fine tuning of both, plateau algorithm and penalty algorithm.

3. Quality Metrics in Constrained Free Space

The previous section illustrated related work regarding quality metrics for alternative routes and alternative graphs in street networks. Abraham et al. state that a proper alternative route should be substantially different from a reference path ("limited sharing"), should not have unnecessary detours ("local optimality"), and should be not much longer than the shortest path ("uniformly bounded stretch"). Similarly, Bader et al. proposed that a good alternative graph should have low overlap of the included routes (high "total distance"), low stretch of included alternatives (low "average distance"), and low complexity (few "decision edges").

In this section, we discuss the transferability of the measures from road networks to constrained free space scenarios. From a general point of view, we can consider only four instead of six measures, since some measures of alternative routes and alternative graphs are similar.

In the following, "limited sharing" and "total distance" will get a unified treatment as they both limit the amount of edges that alternatives have in common. Similarly, "uniformly bounded stretch" and "average distance" are treated together as they both rely on the idea that alternatives should not be excessive in length. The measure of “local optimality”, that means even subpaths should be short, and the idea of counting “decision edges” are the remaining concepts from literature to be adapted for constrained free space scenarios.

3.1. Limited Sharing & Total Distance

One central idea for both, alternative routes and alternative graphs, is linked to the length of the edges that alternatives have in common. The terminology was “limited sharing” for alternative routes and “total distance” for alternative graphs. The central difference between these two ideas is that “limited sharing” applies to pairs of paths while “total distance” measures whether the integration of a given path into an already existing alternative graph is sensible. In practice, optimizing an alternative graph using this measure depends on the ordering of alternatives considered.

The most important question for the application of these measures in free space scenarios is how overlap or sharing can be defined in an unambiguous way.

Even the implementation of a simple shortest path algorithm such as Dijkstra’s algorithm tends to keep the shortest path as much as possible to the left (or right). As constrained free space scenarios potentially have plenty of equal-length paths, this tendency realizes to significantly different shortest path from A to B and vice versa. *Figure 2* illustrates this idea.

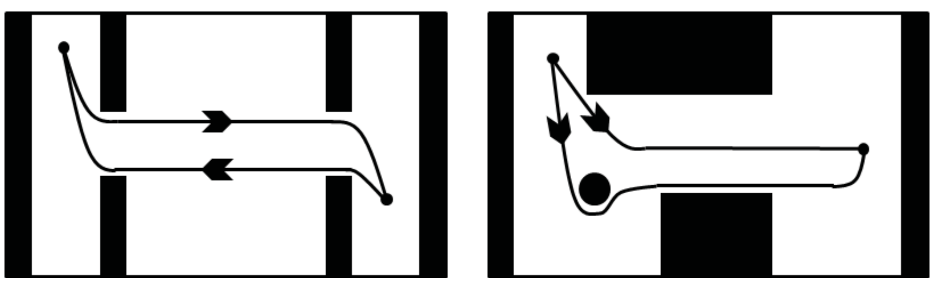

Figure 2. Two examples representing the problem of non-overlapping routes due to implementation details.

The reason why the overlap definitions inspired by street networks is not applicable lies in the high ambiguity of equivalent routes of equivalent meaning. For example, the four routes depicted in *Figure 3* are clearly different albeit human users would discuss whether the two middle routes are actually alternatives. If the map was at continental scale such that no interaction between both routes is to be expected (e.g., airplane navigation), we would accept them as proper alternatives. If, however, this is an indoor scenario where pedestrians on these two routes can see and talk to each other, we would likely identify them as equivalent.

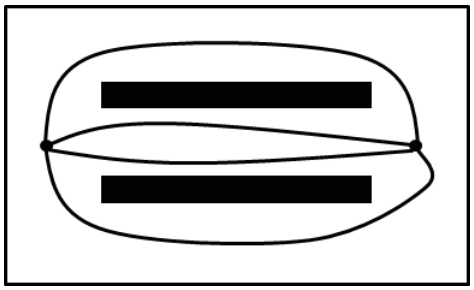

Figure 3. Three or four alternative routes, based on the map scale.

In order to combat this problem, we need a way to decide which points of two different routes actually "overlap" in an application-dependent and map-dependent way. We identify the following three general strategies to extend the measures of overlap to our scenario.

As the original measures were defined for paths in graphs, it would be desirable to consistently map all free space paths to a graph such that different edges mean semantically different movements. Then, two paths that share a specific semantic movement in a building, like going through a specific hallway, will have increased overlap.

There are many ideas of automatically or manually creating such graphs for specific classes of constrained free space navigation. However, realizing a graph means to realize a tradeoff between map complexity and expressiveness. When, for example, a long hallway is represented by a single edge such that all users of the hallway have overlap, this edge can hardly get a sensible embedding into the map. If, however, the graph shall get a visual representation in the map, the complexity will be higher and overlap will become more and more unclear. For the concepts to automatically generate navigation graphs including navigation meshes, straight skeletons, grid graphs and the like, the number of edges and at the same time the quality of a possible overlap measurement depends on the complexity of the involved geometry. Therefore, a general solution to define overlap will not be reached. However, if the application uses any of these techniques anyways, one should consider the possibility of using the graph at least to some extent. Obvious ideas to define a (potentially ambiguous) overlap is to consider the joint traversal of polygons or the identical sequence of passed edges.

A second approach would be the identification of connected spaces such that overlap can be defined from the spaces that two routes have in common. From a general point of view, this amounts to some sense of map topology. Ideally, it would be great if every point was part of one and only one such connected space. This is not always the case, but room names in buildings are a remarkable example where this works. Another sense of topology can be generated from homotopy of routes. That is, two routes have overlap as long as they can be continuously deformed into each other inside a map (see *Figure 3*). However, this concept is, again, a binary one and does not distinguish between small obstacles and large detours. Additionally, homotopy is only applicable to pairs of routes and, therefore, not obviously applicable in alternative graph situations.

This fact motivates the third general approach to extend overlap to constrained free space: We can just replace the binary and counting nature of all overlap measures by a fully continuous framework in which distances of routes are used to assess overlap. This is elegant in the sense that the classical graph-based definitions of overlap can be seen as special cases in which the distance is zero or non-zero. A threshold can then be used to differentiate overlapping and non-overlapping trajectories. On the other hand, one should also consider the integration of continuous measures of overlap in alternative route generation.

3.2. Uniformly Bounded Stretch & Average Distance

Bounding the stretch of a path is a straightforward technique to limit the amount of detours while generating alternative routes. However, when stretch is chosen too small, some alternatives cannot be uncovered. The ideal amount of stretch depends on the application, for example, how many alternatives will be used in the next step, and the environment: At an airport, longer alternatives might be acceptable and lead to completely different journeys through shops; in manufacturing and robotics, however, every detour costs money.

Additionally, the length of a route in a map representation for constrained free space can be highly different from the length in reality. For example, navigation meshes often use central points of larger polygons or edges as waypoints. This makes routes longer than they might be in reality. To the contrary, corner graphs have the property that every length in the derived graph is the shortest among all routes between start and goal.

3.3. Local Optimality

In order to avoid unnecessary detours, the measure of local optimality was introduced in (Abraham et al, 2013). In their situation this was an important feature, because they created alternatives by concatenating two shortest paths leading to situations, where the combined path has got extreme turning points. They optimized such situations by checking around the point of contact of these two shortest paths that no sensible shortcuts exist. In the context of alternative graphs, Bader et al. (2011) realize the same problem, but they push the solution into a post-processing step. We think that this is due to the situation that even alternative graphs can be used to generate overly long alternatives, which imply the need for post-processing anyways.

While local optimality can be strictly defined on a graph level, it is not clear how to define it for continuous free space. The question is how to handle detours that occur only due to inaccuracies in the map-to-graph translation. Additionally, the vast amount of possible choices in free space makes the approach computationally demanding or even infeasible. This all boils down to two questions: What is an "optimal" path and what is a "locally optimal" path. When applying local optimization outside the area of street networks, we think that one should consider local optimization as an application-dependent post-processing step, as the meaning of optimal and local are otherwise unclear.

Additionally, local optimality could be implemented with a different optimization goal: Not the shortest alternative route is to be found, but the simplest one. If one is really interested in optimizing the length, there is a large body of research on line simplification which can be adapted to this situation. But, in many cases this is in contrast to application goals as shortest paths tend to scrape along geometry or cross on unnatural diagonals.

3.4. Decision Edges

"Decision edges" count the number of decisions that can be made in an alternative graph. For a vertex with one edge going out, it is zero as there is no alternative decision (see left-hand side of *Figure 4*). For a vertex with three outgoing edges, however, it is two, as one edge has to be taken, but there would have been two alternatives. For the complete alternative graph, one takes the sum over all alternative choices. The goal in designing alternative graphs is to keep the value of "decision edges" small in order to have a small set of alternatives. However, if it is too small, too many additional alternatives might be discarded.

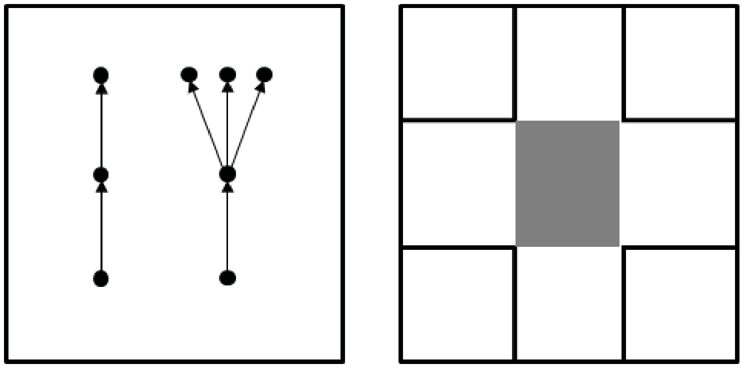

Figure 4. Left: An outdegree of 0 means that no decision can be made, an outdegree of 3 means two decisions can be made besides the necessary flow. Right: Schematic representation of a crossroads where the gray area may represent the location of a decision.

Translating this feature to constrained free space scenarios is similar to the discussion for overlap. One can even define decisions in free space from overlap: A decision is taken at a time where overlap changes. Thus, in some sense, it amounts to extend overlap from edges to vertices.

From a practical perspective, one can again use any existing graph and the associated decisions at graph vertices. However, it is unclear what their actual meaning is. With additional modeling effort, one can integrate the definition of decision in constrained free space into the map creation process. For buildings, for example, a decision edge could be represented as a polygon completely filling a crossing (see right-hand side of *Figure 4*). A decision is taken once a route crosses a different set of edges of this polygon.

While it is clear that in a street network every turn means some sort of decision, it is not clear whether any isolated decision comes up in maritime, pedestrian or plane navigation. Therefore, we think that one should be careful in adaptation of this feature.

3.5. Mutual Interference of Quality Metrics

In the previous subsections, we have presented several approaches to transform the measures for generating alternative routes and alternative graphs to constrained free space scenarios. While we propose various possible ways for each of these measures, one should keep in mind that they actually form deep-rooted interwoven features. When changing overlap, one has to carefully think about what this means for decision edges and vice versa; local optimality can have a strong impact on the definition of both, decision edges and overlap. Furthermore, local optimization can bring bad alternatives into the limits given by the stretch. Figure E depicts the most important links between those measures.

4. Conclusion

In this paper, we first introduced various concepts in the area of alternative routes and alternative graphs as they have been invented for street networks. We showed how constrained free space can be transformed into graphs such that the classical algorithms designed for street networks would be applicable. As we explained, however, the semantic correspondence of vertices and edges with real-world streets and crossings is not evident for constrained free space models, especially not, when they are automatically generated.

Therefore, we discuss the ideas from both worlds, alternative routes and alternative graphs, with which the quality of alternative routes has been measured in street networks in order to extract small sets of truly alternative routes or sufficiently small alternative graphs from the vast amount of possible alternative routes. We discuss methods of extending the classical quality metrics to constrained free space and balance their opportunities and limitations.

Finally, we discuss that all of the numerous choices are interlinked and need to be considered together instead of in isolation. Choosing a map representation has implications for the viability of measures such as overlap and decision edges, the maps themselves have impact on how these measures help in identifying useful alternatives, and changing one of the involved abstractions impacts the overall system.

Therefore, we propose to discuss alternative routes in constrained free space scenarios in an application-dependent context. Many choices that might be reasonable for indoor navigation and computer games could be unreasonable for maritime navigation and airplanes. Still, we believe that our aim of integrating constrained free space scenarios into the area of alternative route research will stimulate discussion and the development of novel and possibly universal ways of selecting alternatives. Additionally, the expected integration of complicated modes of mobility in intermodal navigation (pedestrian, bicycle, mobile robots in production scenarios) illustrates the need for an integrated treatment of environments and alternatives.

References

Abraham I, Delling D, Goldberg A, Werneck R (2010) Alternative Routes in Road Networks. In: Proceedings of the 9th International Symposium on Experimental Algorithms (SEA). Lecture Notes in Computer Science 6049, Springer, pp. 23-34

Abraham I, Delling D, Goldberg A, Werneck R (2013) Alternative routes in road networks. Journal of Experimental Algorithmics (JEA) 18. ACM, pp. 1-3

Bader R, Dees J, Geisberger R, Sanders P (2011) Alternative route graphs in road networks. Theory and Practice of Algorithms in (Computer) Systems. Springer, pp. 21-32

Bast H, Delling D, Goldberg A, Müller-Hannemann M, Pajor T, Sanders P, Wagner D, Werneck R (2015) Route planning in transportation networks. arXiv preprint arXiv:1504.05140

Bauer R, Delling D, Sanders P, Schieferdecker D, Schultes D, Wagner D (2010) Combining hierarchical and goal-directed speed-up techniques for dijkstra's algorithm. Journal of Experimental Algorithmics (JEA) 15. ACM, pp.1-31

Camvit (2009) Cambridge Vehicle Information Technology Ltd. – Choice Routing. Online. http://www.camvit.com/camvit-technical-english/Camvit-Choice-Routing-Explanation-english.pdf

Chen Y, Bell M, Bogenberger K (2007) Reliable pre-trip multi-path planning and dynamic adaption for a centralized road navigation system. In: Proceedings of the 8th International IEEE Conference of Intelligent Transportation Systems. IEEE, pp. 14-20

Dees (2010) Computing Alternative Routes in Road Networks. Master's thesis, Karlsruhe Institute of Technology, Fakultät für Informatik

Dees J, Geisberger R, Sanders P, Bader R (2010) Defining and computing alternative routes in road networks. arXiv preprint arXiv:1002.4330

Delling D, Goldberg A, Pajor T, Werneck R (2011) Customizable Route Planning. In: Proceedings of the 10th International Symposium on Experimental Algorithms (SEA). Lecture Notes in Computer Science 6630, Springer, pp. 376-387

Delling D, Wagner D (2009) Pareto Paths with SHARC. In: Proceedings of the 8th International Symposium on Experimental Algorithms (SEA). Lecture Notes in Computer Science 5526, Springer, pp. 125-136

Delling D, Werneck R (2013) Faster Customization of Road Networks. In: International Symposium on Experimental Algorithms (SEA). Lecture Notes in Computer Science 7933, Springer, pp. 30-42

Dijkstra E (1959) A note on two problems in connexion with graphs. Numerische mathematik 1(1). Springer, pp. 269-271

Eppstein D (1994) Finding the K Shortest Loopless Paths in a Network. In: Proceedings of the 35th Annual IEEE Symposium of Foundations of Computer Science (FOCS), pp. 154-165

Geisberger R, Kobitzsch M, Sanders P (2010) Route planning with flexible objective functions. In: Proceedings of the 12th Workshop on Algorithm Engineering and Experiments (ALENEX). SIAM, pp. 124-137

Geisberger R, Sanders P, Schultes D, Delling D (2008) Contraction hierarchies: Faster and simpler hierarchical routing in road networks. International Workshop on Experimental and Efficient Algorithms. Springer, pp. 319-333

Graf F, Kriegel HP, Renz M, Schubert M (2010) PAROS: pareto optimal route selection. In: Proceedings of the 2010 ACM SIGMOD International Conference on Management of data. ACM, pp. 1100-1202

Kobitzsch M (2013) An alternative approach to alternative routes: HiDAR. European Symposium on Algorithms. Springer, pp. 613-624

Kobitzsch M, Radermacher M, Schieferdecker D (2013) Evolution and evaluation of the penalty method for alternative graphs. In: Proceedings of the 13th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, pp. 94-107

Luxen D, Schieferdecker D (2012) Candidate sets for alternative routes in road networks. Journal of Experimental Algorithmics (JEA) 19. ACM, pp. 2-7

Martins E (1984) On a Multicriteria Shortest Path Problem. European Journal of Operational Research 26(3), pp. 236-245

Paraskevopoulos A, Zaroliagis C (2013) Improved alternative route planning. OASIcs – OpenAccess Series in Informatics 33. Schloss Dagstuhl – Leibniz-Zentrum für Informatik

Radermacher M (2012) Schnelle Berechnung von Alternativgraphen. Bachelor's thesis, Karlsruhe Institute of Technology, Fakultät für Informatik

Scott K (1997) Finding alternatives to the best path

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). IEEE, pp. 230-238

Yen J (1971) Finding the K Shortest Loopless Paths in a Network. Management Science 17(11), pp. 712-716