Adaptable Path Planning in Regionalized Environments

Kai-Florian Richter

Transregional Collaborative Research Center SFB/TR 8 Spatial Cognition
Universität Bremen, Germany
richter@sfbtr8.uni-bremen.de

Abstract. Human path planning relies on several more aspects than only geometric distance between two locations. These additional aspects mostly relate to the complexity of the traveled path. Accordingly, in recent years several cognitively motivated path search algorithms have been developed that try to minimize wayfinding complexity. However, the calculated paths may result in large detours as geometric properties of the network wayfinding occurs in are ignored. Simply adding distance as an additional factor to the cost function is a possible, but insufficient way of dealing with this problem. Instead, taking a global view on an environment by accounting for the heterogeneity of its structure allows for adapting the path search strategy. This heterogeneity can be used to regionalize the environment; each emerging region may require a different strategy for path planning. This paper presents such an approach to *regionalized path planning*. It argues for the advantages of the chosen approach, develops a measure for calculating wayfinding complexity that accounts for structural and functional aspects of wayfinding, and states a generic algorithm for regionalization. Finally, regionalized path planning is demonstrated in a sample scenario.

Keywords: Path planning, regionalization, wayfinding complexity, cognitive ergonomics

1 Introduction

Wayfinding is defined to be a purposive, directed, motivated activity to follow a route from origin to destination that reflects the cognitive processes going on during navigation [1]. It is based on a wayfinder's mental representation of an environment or, in case of not well known environments, on external representations delivered as a means of wayfinding assistance. In urban environments, planning one's route through the environment is largely determined by the network of streets [2]. This network is the predominant structure movement occurs in, i.e., the structure of the environment drives planning and execution of the wayfinding task—at least on the large-scale, environmental level [3, 4].

Automatically planning a path through such networks classically only accounts for geometric properties of the network, predominantly path length and the distance between two nodes. In recent years, several approaches have been presented that instead account for human principles of path planning and the cognitive complexity of traveling through a network. Their calculated paths may result in large detours. This is because they focus only on one aspect of path planning, namely the reduction of cognitive complexity. The approaches ignore geometric aspects and the overall structure of an environment. They only account for the neighborhood of the current location for deciding on how to proceed. This paper presents an approach to regionalized path planning in network environments where the strategy employed for path search can be adapted to properties of the part of the environment planning currently occurs in.

In the following section, some background on human wayfinding in urban environments, wayfinding complexity and cognitively motivated path search is presented. Section 3 argues for a regionalized approach to path planning that acknowledges possible heterogeneity of an environment. Section 4, then, develops a measure for wayfinding complexity and a generic algorithm for performing regionalization of an environment, followed by an illustration of regionalized path planning in Section 5. Section 6, finally, concludes the paper with a summary and an outline of some future work.

2 Wayfinding in Urban Environments

In the following, human path planning and execution (Section 2.1) and its relation to environmental complexity (Section 2.2) is discussed. Different approaches allow for measuring environmental complexity (Section 2.3). Finally, different path search algorithms accounting for these findings are presented in Section 2.4.

2.1 Path Planning in Urban Environments

In their planning, humans often do not take the shortest or fastest path, which today's automatic assistance systems (e.g., internet route planners or car navigation systems) usually calculate. Instead, their conceptualization or perception of the environment determines path choices. Golledge [5] lists a number of factors that influence human path choice. Next to distance and time these include number of turns, shortest or longest leg first, many curves or turns, first noticed and most scenic route. Others (e.g., [6]) have explored how the angle with which the direction of the current or initial segment deviates from the direction to the destination influences path choice. Since these factors are based on cognitive and perceptual aspects and do not rely on distance (or at least just on perceived, not actual distance), routes may differ between traveling to a place and back from that place again [5].

In [7], Wiener et al. found that there is a difference between the chosen path depending on whether it is communicated to others, planned for being traveled by oneself, or actually traveled along. There have been significant differences between the routes that participants (who knew the environment well) had chosen to communicate to others and those they actually traveled themselves. While this may be due to communication considerations, i.e., selecting a route that is easier to describe and remember, there are also differences between the route planned to travel and the actually traveled one. This seems to indicate that routes are not fully planned ahead in every detail, but some options can only be verified and, thus, chosen in situ. This supports the concept of hierarchical planning and execution in wayfinding as argued for by Timpf [2, 3], for instance.

2.2 Environmental Complexity

The structure of an environment has a strong influence on people's wayfinding behavior. The complexity of this structure to a large part determines the complexity of wayfinding. In complex environments, people have more difficulties building up a mental representation, i.e., learning the environment (e.g., [8]). Wayfinding itself is also more difficult. People take longer and make more mistakes [9, 10]. Different factors influence the complexity of an environment's structure. Heye and Timpf [11] elicited some of these factors for traveling in public transport. Here, complexity mostly depends on the structure of the stations and the paths between places to be traveled there. Generally, structural complexity of built environments depends on architectural differentiation, the degree of visual access, and the complexity of the layout [12, 13]. In urban environments, a layout's complexity is determined by the types and orientation of streets [14] or competing spatial reference systems [15], among others.

2.3 Measuring Environmental Complexity

There are different approaches to calculating measures of the structural complexity of an environment. One prominent example is the approach of space syntax [16], which uses largely topological measures to capture the influence of a space's structure on human (social) behavior. A key concept is intelligibility that describes how predictable (understandable) the global structure of space is from observations of local properties. The structure of an environment may further be described using methods of integration or choice, among others.

Others have focused on the structure of the underlying network. O'Neill [17] introduced interconnection density (ICD) as a measure. The ICD value is the average number of nodes connected to every node. In that it captures the connectedness of an environment. It relates to measures of integration (as in space syntax). With increasing ICD, wayfinding performance decreases [17]. This indicates that many possible ways through an environment aggravate constructing mental representations of that environment (there is more to be stored) and increase the chance of making errors (there are more options to choose a wrong turn). In a similar line of thinking, Mark [18] characterized intersections of an environment according to the number of possible choices at them. For each type of intersection, different costs are assigned that reflect the difficulty of navigating it. The simplest action to perform is to go along a straight segment, followed by turning around a corner, which requires a mental update of one's heading. Intersections, i.e., points in a network where there are several options to continue the path, are weighted according to their number of branches. Coming to the dead end of a T-intersection, which forces a decision on how to continue, is treated as a special case with lower costs than other intersections.

In his theory of wayfinding choremes, Klippel [4] elicited human conceptualizations of turning actions in networks. One of the main results shows that humans have prototypical mental representations of the actions performed in the network, rather than prototypical representations of an intersection's structure. Taking this into account, an intersection's complexity increases if it offers several turns into the same conceptual direction (cf. also [19]).

2.4 Cognitively Motivated Path Search Algorithms

To capture wayfinding complexity as discussed above in automatically planning a path through an environment, in recent years different path search algorithms have been proposed that account for principles of cognitive ergonomics. These principles are human-centered, in that the optimization criterion used to determine a path is based on aspects that emerge from the conceptualization of wayfinding situations, for example, the likelihood of going wrong or the ease of describing the path. The criteria are not directly dependent on geometric properties of the path network, such as distances between nodes. All the different approaches can be implemented as a variant of Dijkstra's shortest path algorithm [20]. In the following, these approaches will be presented.

Duckham and Kulik [21] extend standard shortest path search by a heuristic that reflects the complexity of negotiating the decision point represented by the two adjacent edges (e.g., turning from one edge onto another). The specific weighting used is based on an adaptation of Mark's measure [18]. Duckham and Kulik term their algorithm *simplest path* algorithm. In a simulation experiment, they show that their algorithm generally results in paths that are only slightly longer than the shortest path. While the costs employed account for structural differences of intersections, they do not account for functional aspects, for example, (possible) ambiguity in the direction to take at an intersection, nor landmarks or other environmental characteristics that might be exploited in instructions. Like shortest paths, the simplest path finds the cheapest path according to a cost function. Unlike shortest paths, the cost function used applies to the complexity of navigation decisions rather than travel distance or time.

The *landmark spider* [22] is an approach that acknowledges landmarks as crucial elements in wayfinding and performs routing along (point-like) landmarks located in the environment. For these landmarks, a salience value is assumed to be known (e.g., [23]). The landmark spider accounts for three parameters in determining edge weights: the salience of a landmark, its orientation with respect to the wayfinder and its distance to the edge. To compute orientation, a simple homogeneous four sector model is used, dividing the plane in "front," "left," "right" and "back" direction. The landmark spider approach calculates the 'clearest' path, i.e., the path that leads a wayfinder along the most suitable landmarks. The algorithm does not account for any distance information.

The approach to *most reliable paths* presented by [24] aims at minimizing the possibility of choosing a wrong turn at an intersection. As in simplest paths, the structure of an intersection influences path choice. Here, the assumption is that while instructions, such as "turn left," are easy to understand, they may still result in wrong decisions given that there are several options to turn left at an intersection. Accordingly, paths in this approach are optimized with respect to the ambiguity of instructions describing the decision to be taken at intersections. For each turn, i.e., for passing from an edge e to an edge e' that are both connected by a middle vertex, the unreliability measure is calculated. This measure is defined as the number of turns that are instruction equivalent. Two turns are instruction equivalent if they are described using the same linguistic variable. Geometrically, both turns head approximately in the same direction. To compute the linguistic variable, a qualitative direction model is used (cf. [25, 26] for a discussion of the adequacy of different direction models).

In its original implementation, geometric length of edges is used as a secondary criterion to distinguish equally unreliable paths—the shorter one is selected. Consequently, Haque and coworkers extended this original algorithm to allow for a weighting between unreliability (r) and path length (w):

$$op(e, e') = \lambda_d w(e') + \lambda_u r(e, e')$$

This way, the trade-off between reliability and length can be shifted in one direction or the other. In simulation experiments, Haque and coworkers have been able to show that an agent using most reliable paths to navigate through a street network performs better than an agent relying on shortest paths. This also holds for the optimized most reliable paths.

Richter and Duckham [27] combine the reasoning behind simplest paths and context-specific route directions [19]. Just as simplest paths, the algorithm for *simplest instructions paths* finds the best route, i.e., the route associated with the lowest costs in terms of instruction complexity. However, the algorithm utilizes instruction principles and optimization criteria that are related to functional aspects of human direction giving and avoid ambiguity in instructions. The algorithm makes use of the systematics of route direction elements developed in [28]. Consequently, it allows for multiple alternative instructions (labels) to navigate a pair of edges. Furthermore, it realizes spatial chunking [29]. When selecting the node with the currently lowest cost, the algorithm tries to spread all instructions that node has been reached with forward through the graph. To this end, it checks whether neighboring nodes are chunkable using these instructions as well. Using superordinate chunking rules that check for cognitive and structural plausibility, this spreading of instructions allows for traversing several nodes with a single instruction, thus reducing the travel costs, or, in terms of route directions, the number of instructions that need to be communicated.

3 Path Planning in Regionalized Environments

An analysis of the different cognitively motivated path search algorithms shows that compared to the shortest path each approach results in detours in the path from origin to destination [30]. The length of the detour largely depends on the structure of the environment. Since the algorithms are designed to account for complexity in wayfinding and aim at minimizing this complexity one way or other, all avoid complex parts of the environment by navigating around them. Depending on the environment, the resulting detours may turn the calculated paths nearly useless as the increased effort (time and distance) is much greater than the decreased complexity and, thus, users would not accept this path as the solution to their wayfinding problem.

As a consequence, cognitively motivated path search algorithms need to counteract these detours somehow in order to render them useful in any given situation. One way to achieve this is to take the geometric aspects that have been excluded previously into account again. In most reliable paths [24], for example, a weighting is possible between unreliability (a cognitive aspect) and path length (a geometric aspect). Setting the weights is an optimization problem. Favoring the cognitive aspects too much bears the danger of getting large detours, while a strong favor of the geometric aspects may counter the original intention of the algorithms, namely accounting for the complexity of wayfinding.

Introducing this second parameter into the path planning algorithm tries to mediate between the ease of following a route and the length of paths by weighting the parameters. This has two fundamental drawbacks, which are related. First, using a weighting between two factors is a bottom-up approach that tries to fix the detour problem on a uniform level. Everywhere across the environment the same weighting is used. This ignores that the problem domain at hand (the environment) may be heterogenous and may require different strategies in different areas. Second, because of this ignorance of the global structure of an environment the adaptation of the weights has to be done for each new environment individually and mostly from scratch in order to find a sensible balance. It is largely a trial-and-error process that runs until the results "look good."

Instead, since the structure of an environment plays such a crucial role in wayfinding complexity and the performance of the path search algorithms, an obvious approach is to directly exploit environmental structure in countering the detour behavior of cognitively motivated path search algorithms. This is a top-down approach, acknowledging that the environment at hand may be heterogenous with respect to its structure and, as a consequence, its wayfinding complexity. To enable this approach, the environment needs to be analyzed to identify crucial differences within its structure. If no such differences exist, the path search algorithms behave uniformly across the environment. In this case emerging detours will either be small or the environment is so complex that it would be hard to find a sensible weighting that counters the detours anyway.

There are different aspects and different methods to identify the structural differences within an environment (see Section 4). Independent of the chosen method, the analysis results in the identification of different regions, i.e., a regionalization of the environment. Those parts of a network that are in the same region share the same properties according to the chosen regionalization method. They belong to the same *class*. The number of different classes may vary depending on both the environment and the chosen method. For example, there may be just two classes of regions—simple and complex, or the differentiation may be more elaborate, for example, "highway system," "major streets," "suburb," "downtown," and so on. Having a representation of an environment's network that contains information about the regionalization, it is possible to use different path search strategies for each region class. This adapts searching for an optimal path to the environment's structure at hand. The general algorithm and its properties are discussed next, Sections 4 and 5 then illustrate this approach with some examples.

The Regionalized Path Planning Algorithm

The *regionalized path planning algorithm* takes as its input a graph G representing the environment's network of streets and a node o that represents the origin of the route to be taken. The algorithm works on both the original graph and the complete linegraph [21]. The original graph reflects the geometry of the environment, i.e., each node has a position coordinate. Also, each node is annotated with the region class it belongs to. The algorithm also takes a list of region-tuples (class,function); class is a value for the region class at hand, function represents the cost function to be used for path search for this region class. The algorithm's result is a path from origin to destination, represented as a sequence of nodes that need to be traversed. Since these nodes have a position, this sequence can be directly mapped back to the geographic data the graph is derived from.

Path planning is performed as it is done in Dijkstra's shortest path algorithm [20]. To account for the heterogeneous structure of the environment that is identified by regionalization, the algorithm is adapted to allow for using different cost functions depending on the region a node is in. To this end, when expanding a new node it is checked to which region class it belongs. According to this, the costs to reach the neighboring nodes is calculated using the corresponding cost function. Algorithm 1 summarizes the regionalized path planning algorithm. In this notation, the algorithm works on the linegraph [21], which is used by most cognitively motivated path search algorithms. Searching on the original graph can be realized accordingly.

Algorithm 1 is very similar to the well known Dijkstra algorithm. The only addition is the dependence of the cost function on the region class. This reflects that regionalized path planning is a top-down approach. Once the representation of the environment—the graph—is set up with region information, path planning itself is a matter of correctly adapting to the region of the environment search currently occurs in. In that, Algorithm 1 can be seen as a kind of meta-level path search algorithm. It selects the right strategy depending on the current situation. Note that it is not possible to stop search the first time the destination is reached. This is because cost functions, such as the one used in simplest instruction paths [27], might not only depend on local information (the current edge), but on global information (e.g., previously traversed edges or the current region) as well. This may result in situations where a path that reaches the destination later on in the search process may be better than the one that has reached the destination first.

Algorithm 1: The path planning algorithm for regionalized environments.

```
Data: $G = (V, E)$ is a connected, simple, directed graph; $G' = (E', \mathcal{E})$ is the
      complete linegraph; $o \in E$ is the origin (starting) edge; $w : R \times E \to \mathbb{R}^+$
      is the edge weighting function, which depends on the region class $R$.
Result: Function $p : E \to E$ that stores for each edge the preceding edge in the
        least cost path.
// Initialize values;
forall $e \in E$ do
    Initialize $c : E \to \mathbb{R}^+$ such that $c(e) \leftarrow \infty$;
Set $S \leftarrow \{\}$, a set of visited edges;
Set $p(o) \leftarrow o$;
Set $c(o) = 0$;
// Process lowest cost edge until all edges are visited
while $|E \backslash S| > 0$ do
    Find $e \in E \backslash S$ such that $c(e)$ is minimized;
    Add $e$ to $S$;
    forall $e' \in E \backslash S$ such that $(e, e') \in \mathcal{E}$ do
        // Update costs to $e'$ based on region's cost functon
        Set $r \leftarrow$ region($e$), the region class of $e$;
        if $c(e') > c(e) + w(r, e')$ then
            Set $c(e') \leftarrow c(e) + w(r, e')$;
            Set $p(e') \leftarrow e$;
```

4 Regionalization of Environments

As pointed out in Section 3, there are different aspects and methods that may cause a regionalization of an environment. It may be possible to use existing regions, for example, districts of a city to structure the environment. However, this does not tell anything about differences in environmental complexity since these districts are just administrative boundaries that are drawn for historical or census reasons. Another approach could be to divide the streets represented in the network in different classes according to their road status hierarchy. It can be assumed that traveling along highways is less complex than traveling through small streets in a downtown area. However, this default assumption does not always hold. Further, using this simple approach there is no distinction possible within hierarchy levels, i.e., the actual complexity of the configurations of streets on the same hierarchy level cannot be judged.

Thus, for the purpose of path planning that accounts for environmental complexity, regionalization must be performed on the network based on (some of) its properties. For example, it is possible to use the measures explained in Section 2.3 or those underlying the cognitively motivated path search algorithms (Section 2.4). Both ICD [17] and Mark's complexity measure [18] can be used to determine the complexity of the nodes in the network. These two measures, as well as the derived measure used in simplest paths [21], rely on topological and ordering information only. And they account for local information only. They are restricted to information concerning the number of branches at a given node.

In the following, regionalization is performed by, first, calculating a complexity value for every node of the network and, second, clustering these nodes based on their values to form region classes. Based on the discussion in Section 2, three parameters that are derived from the network are used in the process:

1. *Number of branches*: according to ICD and Mark's measure, the more branches there are at a node, the more difficult it is to correctly navigate the corresponding intersection. While this parameter ignores function in wayfinding (cf. [4]), it reflects the fact that a high number of branches provides more opportunity to take the wrong turn, i.e., increases the chances for wayfinding errors. Further, intersections with many streets meeting there become more complex as there is more information to parse before taking a decision.
2. *Average deviation from prototypical angles*: people have prototypical conceptualizations of turning actions in wayfinding [4]. Street configurations that do not adhere to the prototypical angles are hard to conceptualize and to correctly integrate in the environment's mental representation [14]. Thus, covering a functional aspect of wayfinding, intersections with oblique turns are deemed more complex than those following the prototypical angles of 90 and/or 45 degree turns. As wayfinding complexity is calculated for individual nodes (not the edges), the average deviation for all combinations between two of the node's branches needs to be calculated.
3. *Average segment length*: decision points are the crucial parts of route following [31]. In street networks, decision points correspond to intersections. These are the spots along a route a wayfinder needs to decide on the further way to take. They are the spots where wayfinding errors occur. Therefore, along long segments there are fewer possibilities to make errors since there are fewer decision points. Areas with short segments, on the other hand, have a higher density of decision points. Thus, longer segments indicate an area that is less complex for wayfinding, simply because there are fewer options to decide from. Again, since complexity is calculated for nodes, the average segment length over all branches of a node is used.

These parameters capture both structural and functional aspects of wayfinding. They aim for reflecting human assessment of wayfinding complexity.

4.1 The Combined Wayfinding Complexity Measure

Using the three parameters *number of branches*, *average deviation*, and *average segment length* we can define a combined wayfinding complexity measure *CWC*. The complexity *cwc* of an individual node n_k is the sum of the values for the three parameters. The individual parameters can be weighted to account for differences in the relevance of each parameter. The order of magnitude of the different parameters depends on the environment's structure and the geographic data at hand. The number of nodes can be expected to be in the range of lower positive integers—there will hardly be intersections with more than 8 branches. The length of street segments, however, may range from a few meters to several kilometers. Further, when calculating distances, the values depend on the coordinate system's representation. For Gauss-Krüger, for example, coordinates are represented as 7 digit numbers. Therefore, values of the parameters are normalized to be in the range $[0, 1]$ first before adding them up. To this end, the maximum value for each parameter needs to be known, i.e., the maximum number of branches nb_{max}, the maximum average deviation ad_{max}, and the maximum average segment length al_{max}. As discussed above, for number of branches and deviation, small values denote low complexity, for segment length, however, high values correspond to low complexity. Therefore, the length value is subtracted from 1 to account for this difference in the value's semantics.

Equation 1 states how to calculate the wayfinding complexity value of an individual node. Here, nb_k is the number of branches of the node n_k (its degree), the different λ's are the weighting factors, $d(\gamma)$ is the deviation of an angle γ from 90 (45) degrees, $\gamma(b_1, b_2)$ is the angle between two branches b_1 and b_2, and $l(b)$ gives the length of a branch b.

$$cwc(n_k) = \lambda_{nb} \frac{nb_k}{nb_{max}} + \lambda_d \frac{\frac{\sum_{j=0}^{nb_k} \sum_{i=0, i \neq j}^{nb_k} d(\gamma(b_i, b_j))}{nb_k}}{ad_{max}} + \lambda_l (1 - \frac{\frac{\sum_{i=0}^{nb_k} l(b_i)}{nb_k}}{al_{max}}) \qquad (1)$$

A low CWC value for a node corresponds to an intersection that can be expected to be easy to navigate, a high value to a complex intersection.

4.2 Regionalization by Clustering

Applying Equation 1 to all nodes, i.e., calculating the CWC value for each node, results in a distribution of complexity values across the environment. This can now be used to form region classes and, based on the classes, to perform regionalization of an environment.

It can be assumed that the distribution of complexity values is not random, i.e., that there are clusters of nodes which have similar complexity values. However, the values will not necessarily change gradually between neighboring nodes. While this will often be the case, there will also be leaps in values, for example, at those points where local, dense parts of the network connect to main streets that define the global structure of an environment. These considerations can be exploited in the regionalization process.

As a first step, region classes need to be defined based on the CWC values. In principle, there can be an arbitrary number of classes. Usually, a small number of classes will suffice to reflect the environment's structure, though. Each class corresponds to a mutual exclusive interval in the range of $[0, 1]$, i.e. the interval $[0, 1]$ is divided into sub-intervals to form the region classes. According to these intervals, each node belongs to a region class. In order to form regions from the individual nodes, nodes of the same class that are spatially near need to be subsumed into clusters.

In general, neighboring nodes that are in the same region class are subsumed. This way, all nodes that are connected by at least one path that contains only nodes of the same region class form a cluster of that region class. Two further aspects need to be considered: 1) the size of clusters; 2) nodes surrounded by nodes of a different region class. The first aspect relates to the willingness of calling a collection of nodes a "region." Clusters should have a minimum size in order to count as a region. Having two neighbored nodes belonging to the same class hardly can be considered a region. Nodes correspond to intersections in the real world (or to streets when looking at the linegraph). Even if two neighbored intersections are much more complex than their neighbors, they may be remarkable for humans, but most probably are not considered to be a region in their own right. Rather, they will be considered to be special in their neighborhood. Further, the path search algorithms presented in Section 2.4 will avoid such small clusters without much of a detour.

Regarding nodes surrounded by nodes of another class the reverse argument holds. Even though they belong to a different region class, these nodes should be subsumed within the same cluster. They can be considered to be outliers within a cluster. Often, these outliers will belong to a neighboring region class. They will have a range value one below or above the cluster's class and, thus, can be considered to be similar to the other nodes. Using the same reasoning as above, such a node would correspond to a single intersection that differs in complexity to its surrounding intersections within a part of the street network. While being remarkable, conceptually the node will still belong to this part of the network.

The regionalization process consists of two steps. In a first step, connected nodes belonging to the same region class are subsumed (this may also be done using a minimum spanning tree of the street network with similarity of CWC values as edge weight [32]). This results in a set of potential clusters. Each of these clusters is checked against the size threshold for forming a region. If they are smaller the cluster is disbanded again. In a second step, for each node that does not belong to a cluster (anymore), it is checked whether it can be subsumed with a neighboring cluster of a different region class (see Figure 1). To this end, for all neighbors of the node it is checked which cluster they belong to. If two or more of these neighbors belong to the same cluster, the node is added to that cluster. This threshold of two neighbors is used to avoid meandering clusters that consist of a single line of nodes. At the same time clusters are less likely to occur that show large differences in CWC values across the cluster. Using only a single neighbor that needs to belong to a cluster, it becomes more likely that a chain of nodes with slowly increasing (or decreasing) CWC values will be added to that cluster. If a node has more than three neighbors and there are two or more potential clusters a node can be subsumed in, the cluster with the most connections to the node is chosen. The second step is repeated until no more nodes can be subsumed to neighboring clusters. Algorithm 2 summarizes the regionalization by clustering process.

In principle, using this algorithm nodes may end up belonging to two or more regions at the same time. This reflects the fuzziness of many real world's

Algorithm 2: The regionalization by clustering algorithm

```
Data: $G(V, E)$ is the graph representing the street network; for each node $v \in V$
      the CWC value $\texttt{cwc}(v)$ has been calculated using Equation 1; a list of
      tuples (regioninterval,regionclass) that defines which sub-interval of $[0, 1]$
      corresponds to which region class; cts is the threshold size of clusters.
Result: regions, a set of regions
Set regions $\leftarrow \varnothing$;
Set tempclusters $\leftarrow \varnothing$;
Set unclustered $\leftarrow \varnothing$;
// Set the region class for each node.
forall $v \in V$ do
    forall (regioninterval,regionclass) do
        if $\texttt{cwc}(v) \in$ regioninterval then
            Set $\texttt{rc}(v) \leftarrow$ regionclass;
// Construct potential clusters.
forall $v \in V$ do
    forall $n \in neighbors(v)$ do
        if $\texttt{rc}(v) = \texttt{rc}(n)$ then
            if $\exists c : c \in$ tempclusters $\wedge\, n \in c$ then
                Add $v$ to $n$'s cluster;
            else
                // Add $\{v,n\}$ as new cluster.
                Set tempclusters $\leftarrow$ tempclusters $\cup \{v, n\}$;
// Check for size threshold.
forall cluster $\in$ tempclusters do
    if $\texttt{size}$(cluster) $< cts$ then
        Set unclustered $\leftarrow$ unclustered $\cup$ cluster;
        Set tempclusters $\leftarrow$ tempclusters$\setminus$cluster;
// Add unclustered nodes.
repeat
    forall $v \in$ unclustered do
        if $\exists n_0, .., n_{k-1} : n_0.., n_{k-1} \in neighbors(v) \wedge n_0, ., n_{k-1} \in$ cluster,
        $k \geq 2$, cluster $\in$ tempclusters then
            Add $v$ to cluster;
            Set unclustered $\leftarrow$ unclustered$\setminus\{v\}$;
until no node has been added to any cluster ;
// Setting regions to the set of clusters joined with the set of remaining
   unclustered nodes as default region.
Set regions $\leftarrow$ tempclusters $\cup$ unclustered;
```

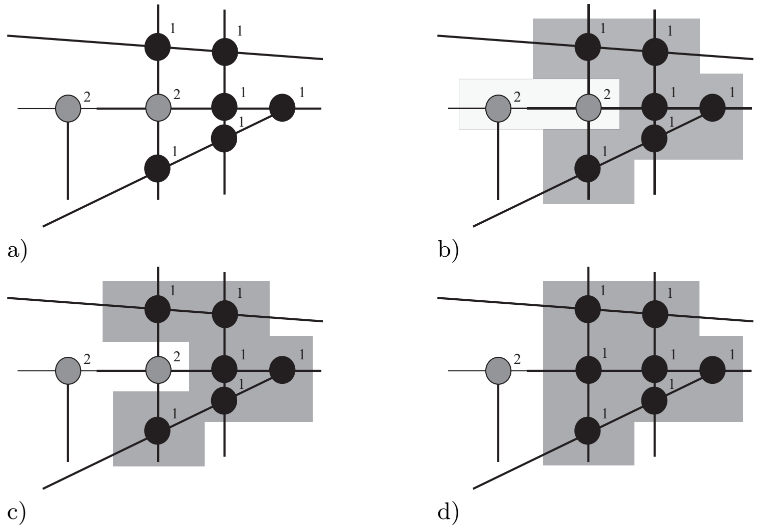

Fig. 1. The clustering process: a) each node is assigned to a region class; b) nodes of the same class are clustered if they are all on the same path; c) clusters smaller than the region threshold (here 3) are removed; d) nodes not belonging to a cluster may be assigned to a neighboring cluster if two or more of their neighbors are within this cluster.

regions [33]. For the application of the regionalized path planning algorithm (Algorithm 1) this would result in ambiguous situations. Therefore, only one of the clusters is kept. Different strategies can be used to decide this: the first cluster the node has been added to may be kept, or the largest or smallest it belongs to. Likewise, some nodes may end up not belonging to any region. Therefore, there always needs to be a default "region" that subsumes all these nodes.

4.3 Discussion

The wayfinding complexity measure CWC has three factors λ_{nb}, λ_d, and λ_l, which are used for weighting the three measure's components. At a first glance, this seems to result in the same effect discussed above for most reliable paths [24]. Setting different weights changes the outcome of the regionalization algorithm. In most reliable paths, setting different weights changes the ratio between accounting for geometric and for cognitive aspects. As argued above, this has drawbacks related to the need to adapt the weights to each environment individually. However, in Equation 1 the weights are used to determine the influence each parameter has on wayfinding complexity. It reflects the influence each parameter has in general; it is not bound to a specific environmental setting. The weights used do not influence the path search strategies employed, but rather determine the regionalized structure path search is performed on.

5 Example

The approach to regionalized path planning as discussed above has been tested using different scenarios. Here, one scenario is used for illustration: path planning in the inner-city area of Bremen. A map of that area can be seen in Figure 2a. The area comprises of some big streets connecting different districts of the city, some parts with regular street patterns, and some dense, old parts of the city, which are especially located along the river.

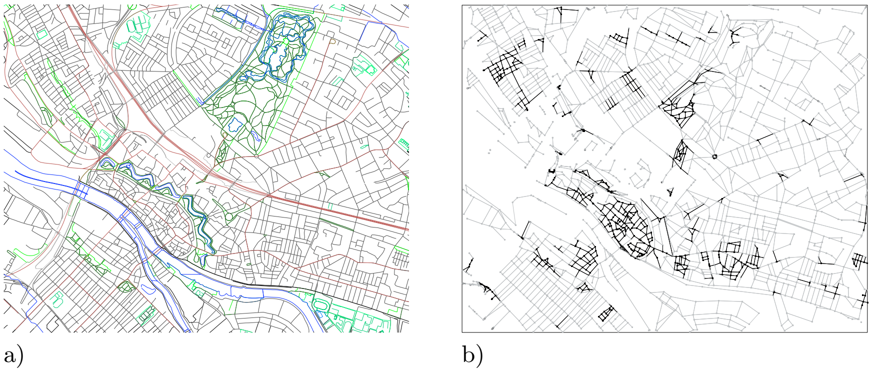

Fig. 2. a) Part of Bremen, the test environment; b) the complex (dark colored) and simple (light) regions identified for this environment.

Figure 2b shows a regionalization of the environment. In this example, only two kinds of regions are distinguished: those regions with complex intersections and regions that are easy to navigate. The region threshold in Algorithm 2 is set to 5. This threshold is chosen for illustration purposes to ensure that regions emerge which will result in a clear detour if circumvented by a path search algorithm. Any node that is not part of a region after clustering is assigned to the "easy" region, which is used as default region. A complex region is a region with nodes having a CWC value greater than 0.6 (this value is chosen ad hoc for illustration purposes only—the CWC value which actually delineates easy and complex intersections would need to be determined empirically; cf. Section 6). As can be seen, the 'complex' regions are mostly located along the river in the dense old-town parts where streets meet at odd angles.

To illustrate regionalized path planning in this environment, for the 'simple' regions the chosen path search strategy is 'shortest path,' for the 'complex' regions the 'simplest paths' [21] is chosen as strategy. The reasoning behind this combination is that in those parts with low wayfinding complexity people will have few problems in understanding the shortest path. Often, shortest and simplest path do not differ much here anyway. It is in the complex parts of the environment that people need instructions that take away the complexity of the environment. Forcing the simplest path algorithm to find a path through complex parts of the environment achieves just that. The resulting paths get wayfinders out of these parts in the simplest way available without resulting in large deviations from the course to the destination. In an actual assistance scenario, where verbal or graphical instructions are provided to wayfinders, understanding these paths can be made easier by employing strategies for generating cognitively ergonomic route instructions (cf. [34, 29, 28]).

Figure 3 shows an example of the shortest, simplest and regionalized path between the same origin / destination pair. As can be seen, the shortest path runs through complex and simple parts of the environment as it only relies on distances between nodes as a geometric property of the network. The simplest path avoids the complex parts, thus, taking a large detour to reach the destination in this example. The regionalized path, finally, lowers complexity in the complex parts of the environment according to the simplest path approach and gets through simple parts of the environment as fast as possible, as predicted.

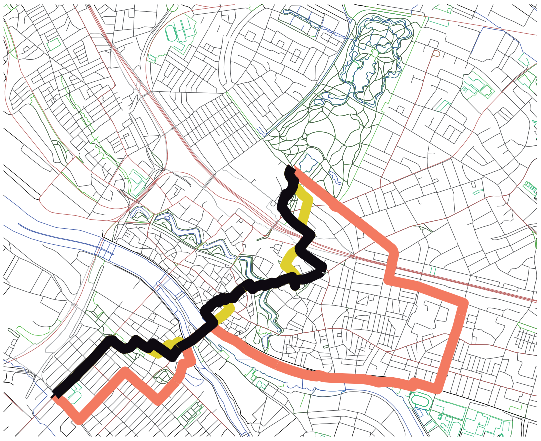

Fig. 3. Planning a path between a sample origin / destination pair. Illustrated are the differences between the shortest path (light colored line), the simplest path (medium colored line taking the bow to the right), and the regionalized path (dark line).

6 Conclusions

Human path planning relies on several more aspects than only geometric properties of an environment, i.e., the shortest distance between two locations. For wayfinding and path planning in network environments, such as urban street systems, in recent years several cognitively motivated path search algorithms have been presented. These approaches cover different aspects of principles of human path planning. Essentially, all the approaches are extensions of Dijkstra's classical shortest path algorithm. By taking into account cognitive aspects of path planning, the approaches largely ignore geometric aspects. They try to minimize wayfinding complexity, not the length of the traveled path. Accordingly, they avoid complex parts of an environment and, thus, may result in large detours. This renders the calculated paths unacceptable.

In this paper, an approach to regionalized path planning has been presented that acknowledges that environments may be heterogeneous, i.e., that environmental complexity usually is not uniformly distributed across an environment, which is an assumption other cognitively motivated approaches implicitly make. This approach employs adaptable strategies depending on the region path search currently occurs in. This way, the strategy that allows for optimal assistance in a given region of the environment may be chosen. The paper presented the overall approach to regionalized path planning arguing for its advantages, developed a measure for an intersection's wayfinding complexity and a generic algorithm that allows for regionalizing an environment based on this measure, and presented an example applying regionalized path planning to adequately cope with changing complexity in an environment's structure.

Regionalized path planning takes a global approach when it comes to accounting for complexity of an environment. This is different to other approaches to cognitively ergonomic path planning that in each step only account for the immediate surrounding of a single node. These approaches never leave their local perspective and, thus, can only take local decisions in their planning. However, path planning is known to be hierarchic [3] and to be based on regions as well [35]. Accordingly, future work on regionalized path planning comprises path search on different levels—the level of regions and the level of individual nodes—and the implementation of more global path search strategies, for example, region-based ones. The global view will also be more strongly reflected in the complexity analysis where aspects of monotony of an environment and similarity of neighboring intersections will be integrated, in line with [36]. Further, both for the weighting of the different factors in calculating the CWC values and for setting a threshold value for complex intersections future (empirical) work is required to elicit values that reflect human conceptualization and behavior.

Acknowledgments

This work is supported by the Transregional Collaborative Research Center SFB/TR 8 Spatial Cognition which is funded by the Deutsche Forschungsgemeinschaft (DFG). The author likes to thank Jan Oliver Wallgrün and four anonymous reviewers for valuable input.

References

1. Montello, D.R.: Navigation. In Shah, P., Miyake, A., eds.: Handbook of Visuospatial Thinking. Cambridge University Press, Cambridge (2005) 257–294

2. Timpf, S., Volta, G.S., Pollock, D.W., Frank, A.U., Egenhofer, M.J.: A conceptual model of wayfinding using multiple levels of abstraction. In Frank, A.U., Campari, I., Formentini, U., eds.: Proceedings of Theories and Methods of Spatio-Temporal Reasoning in Geographic Space. LNCS 639, Berlin, Springer (1992) 348–367
3. Timpf, S., Kuhn, W.: Granularity transformations in wayfinding. In Freksa, C., Brauer, W., Habel, C., Wender, K.F., eds.: Spatial Cognition III. LNAI 2685, Berlin, Springer (2003) 77–88
4. Klippel, A.: Wayfinding choremes. In Kuhn, W., Worboys, M., Timpf, S., eds.: Spatial Information Theory. LNCS 2825, Berlin, Springer (2003) 320–334
5. Golledge, R.G.: Path selection and route preference in human navigation: A progress report. In Frank, A.U., Kuhn, W., eds.: Spatial Information Theory. LNCS 988, Berlin, Springer (1995) 207–222
6. Conroy Dalton, R.: The secret is to follow your nose — route path selection and angularity. In Peponis, J., Wineman, J., Bafna, S., eds.: Proceedings of the Third International Space Syntax Symposium, Ann Arbor, University of Michigan (2001) 47.1–47.14
7. Wiener, J.M., Tenbrink, T., Henschel, J., Hölscher, C.: Situated and prospective path planning: Route choice in an urban environment. In: CogSci 2008: 30th Annual Conference of the Cognitive Science Society, Washington, D.C. (2008)
8. Kim, Y.O.: The role of spatial configuration in spatial cognition. In Peponis, J., Wineman, J., Bafna, S., eds.: Proceedings of the Third International Space Syntax Symposium, Ann Arbor, University of Michigan (2001) 49.1–49.21
9. Butler, D.L., Acquino, A.L., Hissong, A.A., Scott, P.A.: Wayfinding by newcomers in a complex building. Human Factors **35**(1) (1993) 159–173
10. Dogu, U., Erkip, F.: Spatial factors affecting wayfinding and orientation — a case study in a shopping mall. Environment and Behavior **32**(6) (2000) 731–755
11. Heye, C., Timpf, S.: Factors influencing the physical complexity of routes in public transportation networks. In Axhausen, K., ed.: 10th International Conference on Travel Behaviour Research, Lucerne, Switzerland (2003)
12. Weisman, J.: Evaluating architectural legibility: Way-finding in the built environment. Environment and Behaviour **13**(2) (March 1981) 189–204
13. Gärling, T., Böök, A., Linberg, E.: Spatial orientation and wayfinding in the designed environment – a conceptual analysis and some suggestions for postoccupancy evaluation. Journal of Architectural and Planning Research **3** (1986) 55–64
14. Montello, D.R.: Spatial orientation and the angularity of urban routes — a field study. Environment and Behavior **23**(1) (1991) 47–69
15. Werner, S., Long, P.: Cognition meets Le Corbusier – cognitive principles of architectural design. In Freksa, C., Brauer, W., Habel, C., Wender, K.F., eds.: Spatial Cognition III. LNAI 2685, Berlin, Springer (2003) 112–126
16. Hillier, B., Hanson, J.: The Social Logic of Space. Cambridge University Press, Cambridge (1984)
17. O'Neill, M.J.: Evaluation of a conceptual model of architectural legibility. Environment and Behaviour **23**(3) (1991) 259–284
18. Mark, D.: Automated route selection for navigation. IEEE Aerospace and Electronic Systems Magazine **1** (1986) 2–5
19. Richter, K.F., Klippel, A.: A model for context-specific route directions. In Freksa, C., Knauff, M., Krieg-Brückner, B., Nebel, B., Barkowsky, T., eds.: Spatial Cognition IV. LNAI 3343, Berlin, Springer (2005) 58–78
20. Dijkstra, E.: A note on two problems in connexion with graphs. Numerische Mathematik **1** (1959) 269–271

21. Duckham, M., Kulik, L.: "Simplest" paths: Automated route selection for navigation. In Kuhn, W., Worboys, M., Timpf, S., eds.: Spatial Information Theory. LNCS 2825, Berlin, Springer (2003) 169–185
22. Caduff, D., Timpf, S.: The landmark spider: Representing landmark knowledge for wayfinding tasks. In Barkowsky, T., Freksa, C., Hegarty, M., Lowe, R., eds.: Reasoning with mental and external diagrams: computational modeling and spatial assistance - Papers from the 2005 AAAI Spring Symposium, Menlo Park, CA (2005) 30–35
23. Raubal, M., Winter, S.: Enriching wayfinding instructions with local landmarks. In Egenhofer, M., Mark, D., eds.: Geographic Information Science. LNCS 2478, Berlin, Springer (2002) 243–259
24. Haque, S., Kulik, L., Klippel, A.: Algorithms for reliable navigation and wayfinding. In Barkowsky, T., Knauff, M., Ligozat, G., Montello, D.R., eds.: Spatial Cognition V. LNCS 4387, Berlin, Springer (2007) 308–326
25. Montello, D.R., Frank, A.U.: Modeling directional knowledge and reasoning in environmental space: Testing qualitative metrics. In Portugali, J., ed.: The Construction of Cognitive Maps. Kluwer Academic Publishers (1996) 321–344
26. Klippel, A., Montello, D.R.: Linguistic and nonlinguistic turn direction concepts. In Winter, S., Duckham, M., Kulik, L., Kuipers, B., eds.: Spatial Information Theory. LNCS 4736, Berlin, Springer (2007) 373–389
27. Richter, K.F., Duckham, M.: Simplest instructions: Finding easy-to-describe routes for navigation. In Cova, T.J., Miller, H.J., Beard, K., Frank, A.U., Goodchild, M.F., eds.: Geographic Information Science. LNCS 5266, Berlin, Springer (2008) 274–289
28. Richter, K.F.: Context-Specific Route Directions - Generation of Cognitively Motivated Wayfinding Instructions. DisKi 314 / SFB/TR 8 Monographs Volume 3. IOS Press; Amsterdam (2008)
29. Klippel, A., Tappe, H., Kulik, L., Lee, P.U.: Wayfinding choremes — a language for modeling conceptual route knowledge. Journal of Visual Languages and Computing **16**(4) (2005) 311–329
30. Richter, K.F.: Properties and performance of cognitively motivated path search algorithms (under revision)
31. Daniel, M.P., Denis, M.: Spatial descriptions as navigational aids: A cognitive analysis of route directions. Kognitionswissenschaft **7** (1998) 45–52
32. Assunção, R.M., Neves, M.C., Câmara, G., Freitas, C.D.C.: Efficient regionalization techniques for socio-economic geographical units using minimum spanning trees. International Journal of Geographical Information Science **20** (2006) 797–811
33. Kulik, L.: A geometric theory of vague boundaries based on supervaluation. In Montello, D.R., ed.: Spatial Information Theory. LNCS 2205 (2001) 44–59
34. Dale, R., Geldof, S., Prost, J.P.: Using natural language generation in automatic route description. Journal of Research and Practice in Information Technology **37**(1) (2005) 89–105
35. Wiener, J.M., Mallot, H.A.: 'Fine to coarse' route planning and navigation in regionalized environments. Spatial Cognition and Computation **3**(4) (2003) 331–358
36. Schmid, F., Peters, D., Richter, K.F.: You are not lost - you are somewhere here. In Klippel, A., Hirtle, S., eds.: You-Are-Here-Maps: Creating a Sense of Place through Map-like Representations. (2008) Workshop at Spatial Cognition 2008.