JOURNAL OF SPATIAL INFORMATION SCIENCE
Number 21 (2020), pp. xx–yy
doi:10.5311/JOSIS.2020.21.589

RESEARCH ARTICLE

Route schematization with landmarks

Marcelo Galvão[1], Jakub Krukar[1], Martin Nöllenburg[2], and Angela Schwering[1]

[1]Institute for Geoinformatics, University of Münster, Germany
[2]Institute of Logic and Computation, Technische Universität Wien, Austria

Received: November 15, 2019; returned: March 7, 2020; revised: May 25, 2020; accepted: August 13, 2020.

Abstract: Modern navigation applications make use of a turn-by-turn instructions approach and are mostly supported by digital devices with limited display size. This combination does little to improve users' orientation or spatial knowledge acquisition. Considering this limitation, we propose a route schematization method to facilitate the readability of route information and survey knowledge acquisition. Current schematization methods focus on the route path and ignore context information, specially polygonal landmarks such as lakes, parks, and regions, which is crucial for promoting orientation. Our schematization method, in addition to the route path, takes as input: adjacent streets, point-like landmarks, and polygonal landmarks. Moreover, the schematic layout highlights spatial relations between route and context information, improves the readability of turns at decision points, and the visibility of the surroundings. The focus of the paper is the schematization method that combines geometric transformations and integer linear programming to produce the maps. Two routes are used as examples to present the execution information and the outputs. We complement our results with a user study that indicates a preference for our schematic layout in matching textual route instructions. The contribution of this paper is a method that produces schematic route maps with context information to support the user in wayfinding and orientation.

Keywords: schematic map, geovisualization, cartographic generalization, route map, wayfinding, landmarks, orientation

1 Introduction

The use of traditional static road maps (paper maps) is becoming less common in wayfinding and navigation tasks. Compared to modern GPS-based navigation systems, road maps require considerably more cognitive effort in performing such tasks. However, the literature suggests that travelers who used paper maps for navigation have gained orientation and a better comprehension of the geography [18, 25, 33, 52].

GPS-based navigation devices use large-scale visualizations of the route, where the focus lies on turn instructions. Although efficient in the wayfinding task, this approach impairs survey knowledge acquisition, which is essential for the user to build up a mental spatial representation and obtain orientation. Another visualization option is the small-scale topographic overview of the route. However, in this case, turns at decision points and landmarks are oftentimes not recognizable with enough detail to be interpreted by the user.

Schematic visualizations make use of abstract and symbolic representations to improve cognitive ergonomics of map interpretations, and they can be particularly useful for navigation with small screen devices [12]. The most common schematic layout is the metro map, which emphasizes the sequence of stations and connections between metro lines. Route maps have a different function and therefore need different criteria for the schematization. Schematic route maps are expected to emphasize turns at decision points, crossing streets, and their spatial relation with landmarks, be it point-like or polygonal.

Polygonal landmarks are examples of context information that facilitates wayfinding [14, 21, 40, 42], also, they play an essential role as global landmarks for survey knowledge acquisition and self-orientation [2, 41, 46]. Information such as the "route goes around the city center" or "turn right after crossing the lake" represents spatial relations between landmarks and a route. Schematic visualizations can be used for a cognitively adequate representation of such spatial relations [20]. Removing granularity and unnecessary shape complexity improves the focus on such information; however, this needs to be carefully balanced in order to not disturb the user's sense of distance and directions.

(a) Topographic map of the region of Munich.

(b) Official suburban train network map of Munich.

Figure 1: Example of polygonal landmarks in schematic maps. The schematic map (b) indicates that line S8 goes towards the Ammer lake, and the line S6 passes along the Starnberger lake.

The contribution of this work is a method to produce cognitive adequate schematizations of routes with context information such as side streets, point-like landmarks, and mainly, polygonal landmarks. Our drawing method allows a controlled abstraction of the shape of the polygonal landmark. Moreover, it explicitly considers the generalization of the landmarks' topological relations with the route. Figure 1 exemplifies how cartographers use schematic visualizations to highlight topological relations between paths and polygonal landmarks.

Formally speaking, our schematization method takes as its input the target route R as a geometrically embedded path and adjacent roads represented as stubs. Moreover, the spatial context of R is given as a set of point-like and polygonal landmarks. The goal is to compute a topologically correct schematic representation of the route R and its context, which satisfies a number of hard constraints and optimizes an aesthetic quality measure. We implement our approach as a combination of integer linear programming (ILP) for map schematization and local geometric transformations. Figure 2 illustrates the different steps of our method.

In order to evaluate our method, we asked participants to follow textual route descriptions together with two map layouts, one with the original shape and another with the schematic layout generated by our algorithm. In 82.8% of the evaluations, participants reported that following the route descriptions was easier with the schematic maps.

The remainder of this article is structured as follows: In Section 2 we review related work on map schematization with emphasis in routes and polygons. In Section 3 we describe the schematization method which includes pre-processing steps (Sections 3.1, 3.2, 3.1 and 3.3), the ILP model (Section 3.4), and the specificities of the route and polygonal landmark schematization (Sections 3.5 and 3.7 respectively). In Section 4 we present the results of our drawing method as well as the results of the user study on the readability of the maps. Contribution, limitations and future work are discussed in Section 5. We conclude our work in Section 6.

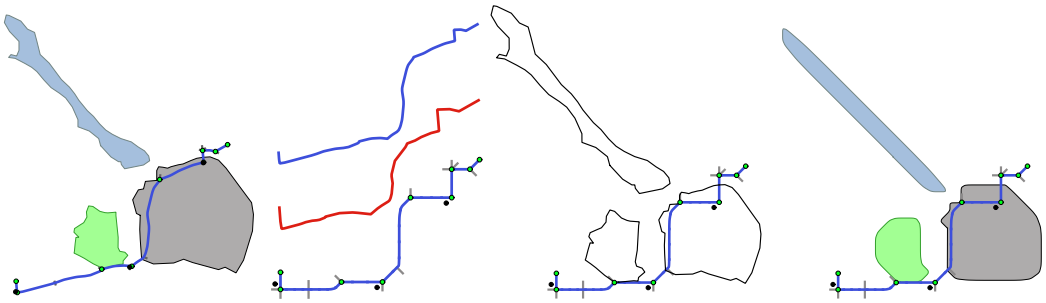


Figure 2: Schematization process flow. (a) As input, we get the route path, adjacent streets, point-like (black dots) and polygonal landmarks. (b) Original route path in blue (top), rescaled route path in red (middle), and ILP schematization of route path and with adjacent streets and point-like landmarks (bottom). (c) Affine transformation to transpose the polygonal landmarks into the schematic route. (d) The polygons are schematized to highlight spatial relations and to fix topological inconsistencies.

2 Related work

We divide the related work section into three topics. First, in Section 2.1 we review the bibliography on conceptual models and layout criteria for route maps as an informative tool for wayfinding and spatial learning. Second, in Section 2.2, we review publications on automatic schematization of routes and polygons. Finally, in Section 2.3, we specifically review publications that use ILP for schematic map drawing, since ILP is the core technology used in our method.

2.1 Route schematization layout criteria for wayfinding and spatial learning

Schematic cartography, different from conventional cartography, applies generalizations to improve a map's functionality that goes beyond the adaptation of the map's features into a specific map scale [50] i.e., independent of the scale the map is presented at, a generalization is applied because it improves the map's function as a task solving tool. The cognitive adequacy and the usability related to the map's functionality justifies the schematic layout criteria that are later transformed into drawing rules.

According to Klippel [22], adequate characterization of routes as a visual wayfinding tool must emphasize its primitive elements that are mentally conceptualized as route knowledge. Klippel named these primitives wayfinding choremes, and divided them as choremes for turn directions at decision points and choremes for spatial chunking. For turn directions, Klippel proposed seven choremes (SHARP RIGHT, RIGHT, HALF RIGHT, STRAIGHT, HALF LEFT, LEFT, and SHARP LEFT) as a natural human conceptualization of directions and adequate for schematic layout criteria for route knowledge acquisition. Spatial chunking is a combination of two or more elementary route instructions into a single one. For the spatial chunking of routes, Klippel proposed three kinds: numerical chunking that usually makes use of streets adjacent to the route, e.g., "turn left at the third intersection"; structural chunking that uses the structure of the network, e.g., " at the roundabout, take the second exit"; and landmark chunking, e.g., " turn left after the gas station". Chunking route instructions is a natural efficient way to mentally store and communicate route knowledge [21], and therefore, an adequate route schematization must facilitate the readability of features that support spatial chunking.

Richter [40, 41] extended Klippel's model and provided a systematic way to analyze route environments in order to produce optimized cognitively adequate instruction chunks that consider landmarks and their spatial relation with the route. The produced chunked route instructions are abstract; it means that the system stores them in a data structure, and they need to be converted to human-readable text, pictorial representations, or schematic route maps. Our schematization method, in its limitation, addresses the requirements of Richter's system for cognitively adequate instructions. Some features of the system, such as environmental structures or linear landmarks, are not covered by our method.

Beyond the function of supporting route instructions, global (off the route) or polygonal landmarks allow map users to extend the unidimensionality of route paths. They enrich route maps and promote orientation [26,31,41]. Anacta et al. [2] revealed that people make use of local and global landmarks (even if not visible from the route) in maps in order to structure a mental map of the region and orient themselves. Löwen et al. [30] showed that highlighting global landmarks (point-like or polygonal) in assisted navigation devices

promotes survey knowledge acquisition. Wiener and Mallot [53] also showed that human route planning takes regions and their spatial relations into account. Schmid et al. [44] highlighted the relevance of presenting selected regions and polygonal landmarks in route maps to support structured spatial learning and orientation.

We base our layout criteria for the polygonal landmarks schematization on public metro map examples. We note that the general principles of schematization by removing unnecessary information (shape, angle, and length generalizations) are applied to polygonal landmarks. Moreover, path-polygon relations used in natural languages [24] are emphasized. Figure 1 exemplifies how a relation "along" is more evident in the schematic layout. Although the Line 6 is not running precisely parallel to the Starnberger lake and in some places the distance between them is over 1km, in the schematic layout the west side of the lake and the train line are collinear.

Another layout criterion frequently applied in route maps is the scale variation, i.e., when different scales are applied to the same map. As pointed out by Delling et al. [9], the information density in a route varies along its path, for example, inside an urban area the number of decision points tends to be much larger than in a highway, and the proper scale to visualize a route depends on this complexity. Schematic cartography allows the application of different scale levels in different parts of a map within a single visualization, what could mitigate the fragmentation of spatial information caused by constant zooming and panning [43]. In *The bounds of distortion* [12], Godfrey and Mackaness discuss different kinds of scale variation applied in cartography; they also highlight the relevance of such distortions for the efficacy of navigational information in the context of small devices.

2.2 Route and polygon schematization methods

Although automatic generalization has been an important topic since the beginning of digital cartography, research on automatic schematization started only in the late '90s and since then has been a prolific topic. The most recognizable example of a schematic map layout is the Tube Map of London's metro system. Almost all cities that have a metro system also provide their users with a schematic map of the network. For this reason, most of the research on schematization algorithms focused on transit networks [3, 11, 17, 28, 37, 51, 54]. Those methods were surveyed in [35].

In the category of **route schematization**, Agrawala and Stolte [1] proposed a method to schematize routes inspired by handmade route sketch maps. First, short roads are extended to a minimal length to make them more visible. Second, the route path is simplified in a simulated annealing process. Topological inconsistencies and incoherence in the turning angles are penalized in the objective function. This approach is only a path schematization, context information like crossing-roads and point-like landmarks are placed in a post-processing step just as edge decoration. The schematization does not follow the principles of cognitively motivated instructions [41], and neither considers polygonal landmarks.

Delling et al. [8] presented a co-orientated (octilinear) route schematization method that preserves the orthogonal order among all vertices. Their 2-step method uses linear programming to schematize monotone paths in polynomial time. To be applied in any case, the route path is submitted to a heuristic process that decomposes the route into a minimal set of monotone paths that are joined together after the schematization. The method was further extended [9] to be executed in a single step using ILP. By simplifying the route in a pre-processing step, their solution was mostly able to produce results in less than one sec-

ond. The distortions in the final results tend to be very exaggerated because the total edge length is minimized in the objective function. Delling et al. did not consider landmarks at all.

Polygon schematization can be made using simplification methods. De Berg et al. [7], Barkowsky et al. [4], and da Silva and WU [6] presented line simplification algorithms (applicable to polygons) that consider surrounding geometric features to prevent topological inconsistencies. Such simplifications are useful for schematization because they reduce complexity keeping the original shape perception. However, they are evolutionary and the input instance must be topologically correct. Our ILP method allows an input instance with a wrong topology; we illustrate the fix of such topological inconsistencies in Section 3.7.

Polygon schematization methods that consider typical schematic style constraints gained attention more recently. Buchin et al. [5] presented an area-preserving quadratic time schematization method for polygons or subdivisions. Their method iteratively reduces the complexity of the shapes by aligning edges to remove bends. In order to preserve area, a second alignment is selected to compensate the loss or gain of the area. Their method allows octilinear simplification if the input is octilinear. Van Dijk et al. [49] presented a curvilinear polygon schematization method. The algorithms iteratively substitute a pair of adjacent edges for a single circular arc with minimal Fréchet-distance. A substitution is rejected if a topological inconsistency is created or if the distance is out of a given tolerance. Both methods, Buchin et al. [5] and van Dijk et al. [49], show good results for single polygons or polygonal subdivisions, but they were not applied in the context of highlighting spatial relations of regional landmarks in route maps.

2.3 Schematic map drawing with integer linear programming

A schematization method based on integer linear programming (ILP) was first proposed by Nöllenburg and Wolff for the automatic drawing of metro maps [36]. Since then, several publications proposed schematization methods using a similar framework. Oke and Siddiqui [38] simplified the original model to reduce execution time. Wu et al. [54] extended the model to create space for pictures (annotations) that can be associated with the metro stations. Delling et al. [9], already mentioned in Section 2.2, adapted the model for a path schematization that preserves the orthogonal order of the nodes. Lan et al. [27] extended the model to preserve the representation of major structures of the metro network, e.g., ring lines.

Linear programming is an optimization technique used to find optimal values for real-valued variables that minimize (or maximize) a linear objective function and are subjected to restrictions that must be modeled as linear inequalities [45]. There exist polynomial-time algorithms to solve linear programming problems. However, some of the constraints require extra variables to represent discrete information, like restricting edge orientations to a fixed set of angles, or to guarantee the correct topology, e.g., a test whether one point is to the left or right of another. Discrete information requires the use of binary or integer variables. Linear programming with integer variables (ILP) is NP-hard, but due to its high relevance in discrete optimization, several practically efficient solvers exist.

For map schematization, the desired variable values are, for example, point coordinates; the hard layout constraints are the restrictions; and soft layout constraints form the objective function as a weighted linear equation that is minimized in the optimization process. For instance, Nöllenburg and Wolff's model [37] has three soft layout constraints that are

minimized together in a weighted objective function: bends, relative positions, and total length:

$$\text{Minimize} \quad \alpha_{S1}\text{cost}_{S1} + \alpha_{S2}\text{cost}_{S2} + \alpha_{S3}\text{cost}_{S3} \tag{1}$$

Where cost_{Si} is the cost of a soft constraint and α_{Si} the weight of the soft constraint defined as a parameter.

An ILP-based approach has an important advantage over other methods because the schematization is not limited to a locally optimal solution like in hill-climbing [47] or simulated annealing methods [1, 3], but rather the whole solution space can be inspected to find the solution that exactly minimizes the soft layout constraints. Moreover, topological correctness can be guaranteed as a hard constraint, unlike some other optimizations that include violations in the topology as a cost in the objective function [1]. Specifically, the minimal edge spacing constraint (preserves planarity), and the circular order constraint (preserves the embedding) guarantee a solution with the correct network topology. Notwithstanding, the ILP solution has limitations. First, it is its NP-hardness, which means that too large instances make it impractical, especially for a real-time application. Second, restricted linearity is mandatory, i.e., the orientations of the edges are limited to a fixed set of angles. The original approach [36] is octilinear (8 directions), but Nickel and Nöllenburg [34] demonstrated how to extend the model to be applicable with $2k$ directions for any integer k.

The major difference in the conceptual model between our method and previous ILP models for schematization [9, 37] is that we do not minimize the total edge lengths in the objective function. This constraint results in a very compact layout adequate for metro maps; however, the scale variation tends to be extreme, causing a distortion that is not always desired for route maps in a driving situation. In our approach, we limit the scale variation level with a parameter in a process before the ILP schematization (Section 3.3). Another important difference is that, instead of running a single ILP model for the whole instance, we run several ILP optimizations for subsets of the input. As a result, we do not obtain a globally optimal solution for the whole instance, but in compensation, there are some advantages of our approach. First, there is a considerable reduction in the total runtime because of the NP-hardness of the problem. Second, partial results can be delivered while other parts of the instances are processed. And third, distinct ILP models or weights in the objective function can be applied to the different parts of the instance. For example, we use distinct ILP models for the route and for polygonal landmarks.

3 Schematization method

The flowchart in Figure 3 illustrates the main steps involved in our algorithm. We divide the main processing steps into five independent process boxes, and we will refer to their numbers throughout this section. The route data is loaded from a street network database that allows routing algorithms. The street data is derived from OpenStreetMap [39] data of Münsterland (Germany). This street network contains only the topological information, meaning that parallel segments of the same street are collapsed into a single one, the roundabouts were collapsed into a single node, and street links, such as highway exits, were removed.

The process box 1 contains pre-processing steps that prepare the data structures required for the schematization. It gets as input, in addition to the route path, surrounding

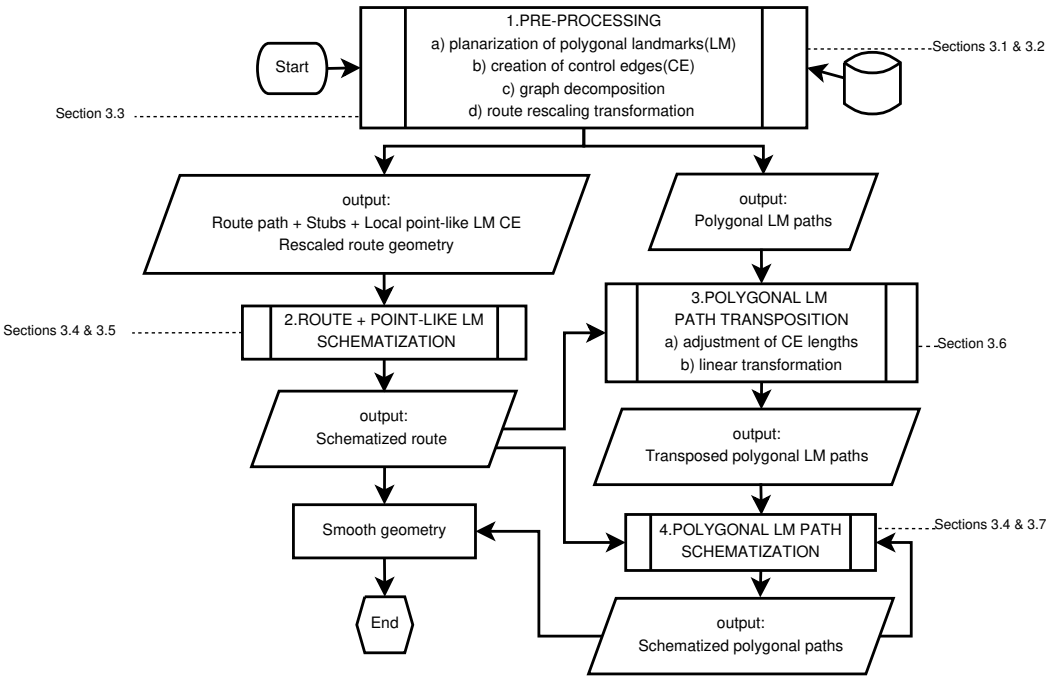


Figure 3: Schematization method flowchart. The five process boxes are associated with the document sections where they are described in detail.

streets, point-like landmarks, and polygonal landmarks. First, we planarize the polygons with the route. Second, we create special edges (control edges) to connect the point-like landmarks and the disconnected polygonal landmarks to the route; this step is explained in detail in Sections 3.1 and 3.2. Third, we decompose the graph into paths, separating the route, street edges adjacent to the route (stubs), and polygonal landmark paths. Finally, we apply our route rescaling transformation that will define the scale variation on the final map; we explain this step in Section 3.3.

The process box 2 shows the route schematization using ILP. It gets as input the route path, the adjacent streets, and the control edges of the local point-like landmarks. The ILP model uses the rescaled route geometry as a reference for the node positions and proportion of the edges. The output is the schematized route, with the adjacent streets and the point-like landmarks. We describe the ILP model constraints in Section 3.4, and how they are applied for the route schematization in Section 3.5.

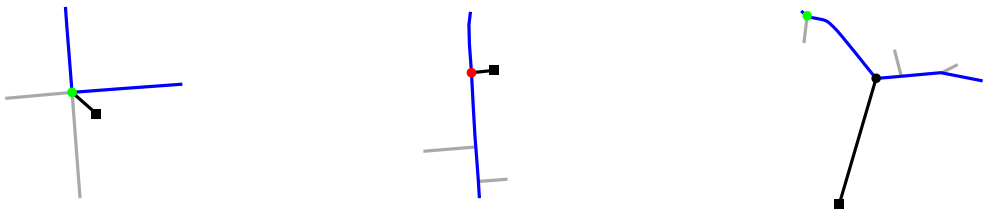
The process box 3 contains the steps required before the schematization of the polygonal landmark paths. First, we adjust the length of the control edges to make it adequate to the scale variation of the schematized route. As explained in Section 3.6, the new length of the control edges depends on the length of the edges of the schematized route. Second, we transpose the polygonal landmark paths to the schematized route; for that, we apply a linear transformation that uses the crossing nodes or the control edges nodes as control points.

The process box 4 shows the schematization of the polygonal landmark paths using ILP. It gets as input the transposed paths that are here sequentially schematized. The schema-

tized route and previously schematized paths need to be submitted to the process in order to guarantee the correct topology. The polygonal landmarks ILP schematization is explained in Section 3.7. Finally, for aesthetics reasons, we submit the resulting schematized path (and the route) to a smoothing generalization.

3.1 Control edges for point-like landmarks

A control edge is a distinct edge created in a pre-processing step (process box 1 in Figure 3) that is used to preserve the relative position of the landmarks with the route. The point-like landmarks are initially loaded as disconnected from the route, so we use control edges to connect them. The way we create the control edges depends on the type of the landmark. We classify the point-like landmarks into three types: landmarks at decision points (DPs), landmarks along the route, and global landmarks.



(a) *At DP* landmark. (b) *Along* landmark. (c) *Global* landmark.

Figure 4: Examples of control edges (black edges) created for different classes of point-like landmarks (black squares). The control edges connect to: a DP, green dot in (a); to a new node on the route, red dot in (b); or to an existing route node, black dot in (c).

For landmarks at DPs, we create an edge connecting the landmark to the DP node (Figure 4a). For landmarks along the route, i.e., close to the route but not at a DP, we add to the route path an extra node at the closest point to the landmark and create an edge connecting this node to the landmark (Figure 4b). For global landmarks, we create an edge connecting the landmark to the closest node in the route path (Figure 4c).

The control edges of the landmarks at DPs and along the route are schematized together with the route and treated as an adjacent edge (Section 3.5). This way, their topology concerning the route and the adjacent streets will be preserved. As for the global landmarks, their control edges are not considered by the ILP model. They are used to preserve the relative position of the landmarks with respect to the route. So, after the route is schematized, the new position of the global landmarks are calculated based on the length and the incident angle of their respective control edges. The incident angle is preserved, but the length needs to be adjusted to be compatible with the rescaled route. This length adjustment is the same as for polygonal landmarks' control edges and is explained in Section 3.6.

3.2 Control edges for polygonal landmarks

Before the polygonal landmarks are schematized, we use a linear transformation to adjust their position in relation to the schematized route (process box 3 in Figure 3). To support this linear transformation with control points, we create control edges that connect the

route with the landmarks. The creation of the control edges depends on the type of the polygonal landmark.

Along the route landmarks

Along the route landmarks are polygonal landmarks that are not crossed but that lie within a certain distance to the route. For this case, we set a pair of control edges. Figure 5 illustrates the pair of control edges created for a polygonal landmark along the route. The control edge is defined by a control node n (polygon node) and the closest point on the route, where an extra node is created. The control nodes are selected based on two values: the distance to the route (d_r); and the distance to the orthogonal line at the beginning (d_{l1}) or end (d_{l2}) of the linear referencing of the polygon against the route, one for each control edge. The values d_r and d_l are weighted in a linear function, and the node with the lowest resulting value is selected.

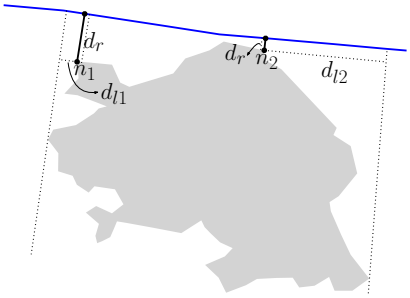


Figure 5: Control edges (black solid line) for polygonal landmarks *Along the Route*. The control nodes n_1 and n_2 are selected because they mutually minimizes d_r and d_{l1}, and d_r and d_{l2}, respectively.

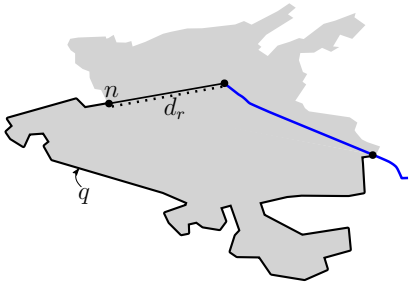


Figure 6: A single control edge for *Origin/Destination* polygonal landmarks. The control node n is selected because it mutually minimizes the d_r and $|q - p/2|$, with p being the perimeter of the polygon.

Crossed polygonal landmarks

Crossed landmarks are polygonal landmarks whose boundaries intersect with the route an even number of times. They do not need control edges since the crossing nodes suffice as control nodes for a proper adjustment in the affine transformation.

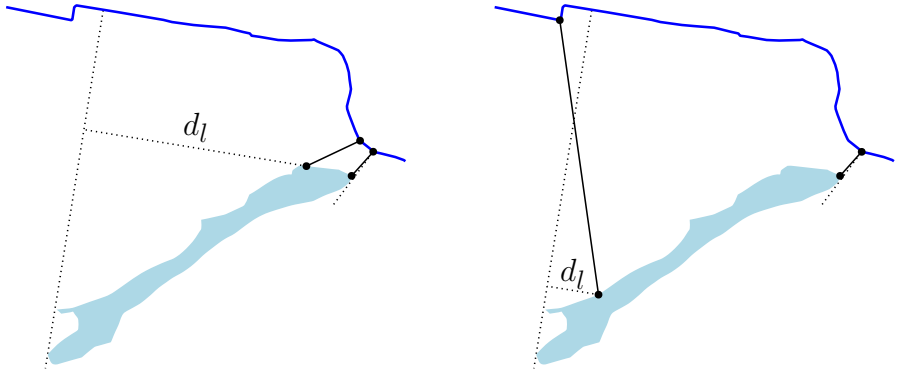
Origin or destination polygonal landmarks

Origin or *destination* landmarks are polygonal landmarks whose boundaries intersect with the route an odd number of times. Depending on whether the first or the last node of the route is inside the polygon, we classify it as an origin or a destination. One of the crossings is always one control node, therefore only a single control edge is created (Fig. 6). Again, we select the polygon node (n) of the control edges based on two arguments: (1) the distance of the route (d_r), and (2) the value specifying how the node, together with the crossing control node, divide the polygon into two paths of a similar length. Let p be the perimeter of the

polygon, and q the length of the path connecting the node to the crossing control node. The second argument of the evaluation function is calculated as $|q - p/2|$.

Global polygonal landmarks

Polygonal landmarks are classified as *global* if they are neither *crossed* nor *along the route*, nor the *origin* or *destination*. Polygonal landmarks containing the entire route (e.g., a route inside a city) are treated as *global*, too. For *global* polygonal landmarks, we set a pair of control edges the same way we set it for *along* landmarks. The only difference is a greater weight given to the d_l values in a evaluation function. Figure 7 demonstrates how the control edges are selected if the same landmark is treated as *along* (7a), or *global* (7b).



(a) Control edges as *Along the Route*. (b) Control edges as *Global*.

Figure 7: Difference in the creation of control edges if the same landmark is treated as *Along* (a) and *Global* (b). The value d_l for *Global* (b) received a 5 times higher weight in the node evaluation equation.

3.3 Route rescaling transformation

The rescaling of the route is a pre-process (process box 1 in Figure 3) of the schematization that creates a scale variation along the route. The Route ILP model uses the rescaled route geometry as a reference for the position of the nodes and the proportion of the edges. The rescaling allocates more space to parts of the route that have a higher concentration of decision points (DPs), making the map more suitable to modern navigation devices with a small screen [12].

Strictly, a DP is a point in space or time at which a choice must be made. In our method, a DP has a more flexible definition. Let a DP be a node in the route where a turn is made at a new street, a roundabout, or any salient intersection; we call a path connecting two DPs a route section. The scale of each route section is reduced proportionally to their length using a common parameter that defines the level of distortion. The resulting length $l(s_k)$ of a route section s_k is calculated as

$$l(s_k) = l'(s_{min}) + (l'(s_k) - l'(s_{min})) * (1 - p^{\phi(i)}) \quad (2)$$

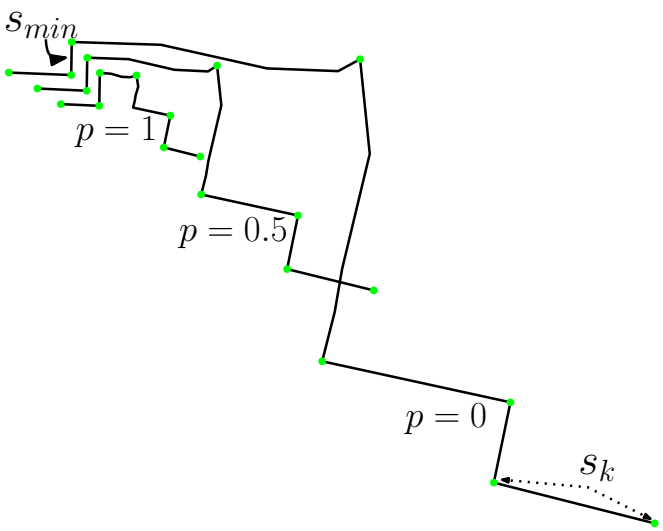


Figure 8: The effect of rescaling of the route using different values for p. In the extreme case ($p = 1$), all route sections (sub-paths between the green dots) have the length of the shortest route section (s_{min}).

Where $l'(s_k)$ is the original length of s_k and $l'(s_{min})$ is the given length of the shortest route section. The float parameter $p = [0, 1]$ specifies by how much the sections are reduced. In the minimal distortion ($p = 0$), the sections keep their original length, and in the maximal distortion ($p = 1$), the resulting length is the length of the shortest section. We raise p to $\phi(i) = (\frac{l'(s_{min})}{l'(s_k)})^i$ to increase the effect of p proportionally to the length of the route sections ($l'(s_k)$), where i is a float variable in the range $[0, 1]$. The higher the value of i, the higher the effect of p. All examples present in this paper use $i = 0.2$.

Figure 8 exemplifies the effect of the rescaling transformation in a route path for three different p values: (i) the original route path ($p = 0$), (ii) an intermediate rescaling distortion ($p = 0.5$), and (iii) the maximum rescaling distortion ($p = 1$). Note that, for ($p = 1$), all route sections get the same length. Moreover, the rescaling transformation shrinks the route geometry, and only the coordinates of the nodes in the shortest route section are preserved.

Figures 9 shows the effect of the rescaling process for four different routes and four different p values ($p = 0$ is the original geometry). For each route, we expand the rescaled geometries to have them in a similar extension. Note that the rescaling process allows shorter route sections to be more visible, and at the same time, the resulting geometry maintains similarity to the original geometry. The rescaled route geometry is used as an input in the ILP model that tries to preserve the proportion of its route sections (process box 2 in Figure 3).

It is important to mention that the rescaling process is not a topologically safe transformation for non-monotone paths. For example, route 4 and $p = 0.9$ in Figure 9 (red path) contains a violation of the planarity. Some routes have a limit for p until planarity is violated; for route 4 in Figure 9, this limit is 0.86. Nevertheless, the rescaling process, in the distortion limit ($p = 1$), did not violate planarity for more than 95% of the routes we tested.

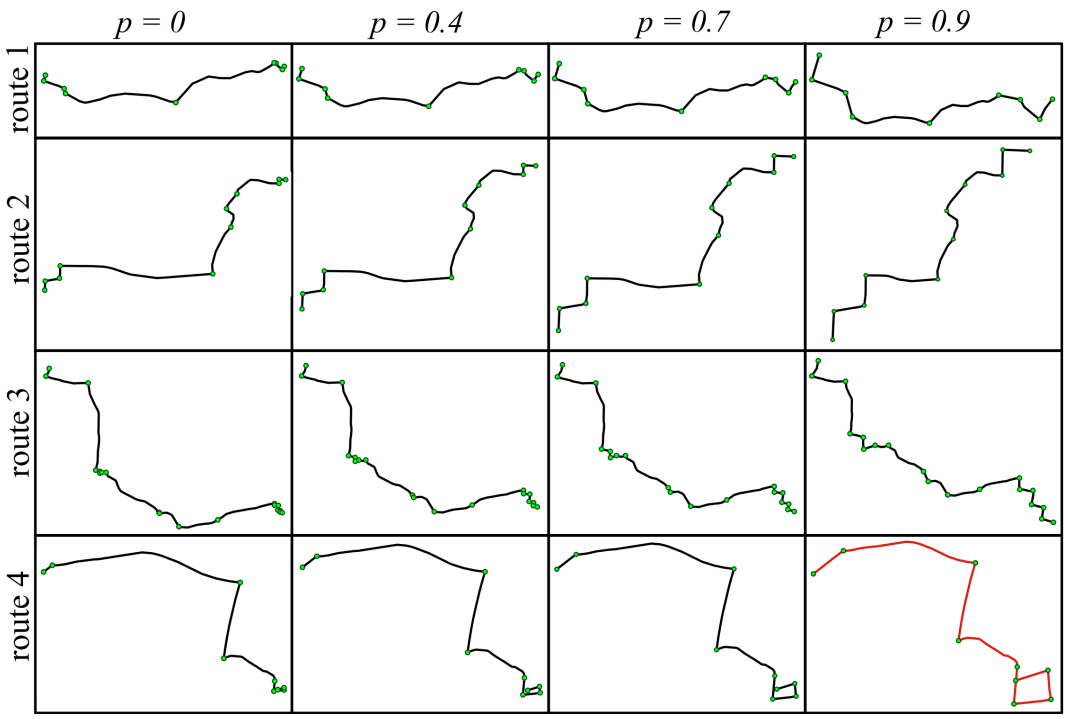


Figure 9: Example of rescaling transformation varying the distortion parameter p applied to four different routes. Shorter route sections (sub-paths between the green dots) are more visible, and the resulting geometry preserves recognizability to the input geometry ($p = 0$). Route 4 and $p = 0.9$ (in red) is an invalid transformation because it violates planarity.

This is because single destination routes generated by shortest path algorithms have a high level of monotonicity.

3.4 The ILP constraints

This section describes the ILP constraints used for the route and for the polygonal landmark paths schematization (process boxes 2 and 4 in Figure 3 respectively). In Sections 3.5 and 3.7 we explain how we apply the constraints for each case.

An ILP schematization process gets as input a graph $G = (V, E)$, where V is the set of nodes, each of which has x and y Cartesian coordinates, represented here as $x'(v)$ and $y'(v)$ for all $v \in V$; E is the set of edges $e = uv$ composed by a pair of distinct nodes, where u and $v \in V$ and where there is a direct link between u and v. The expected output are the schematized coordinates $x(v)$ and $y(v)$ for all $v \in V$, such that the layout hard constraints are satisfied and the layout soft constraints are optimized.

The hard constraints are *octilinearity*, *best turn at DP*, *stub length*, *circular order*, and *planarity*. The last two guarantee the topologically correct output. The soft constraints are *bend minimization*, *edge orientation*, *node position*, and *route sections proportion*. The constraints *best turn at DP*, *stub length*, *node position*, and *route sections proportion* were newly implemented

to address the particularity of route maps; the others were modeled identically as those used in the original ILP model for metro map [37].

Hard constraint: Octilinearity

Octilinearity is a common rule for metro maps. For route maps, it makes intersection representations compatible with the 8-directional model of Klippel's wayfinding choreme theory [22]. According to Klippel, a lower level in the granularity of intersections' angles better reflects a cognitive conceptualization of directions, facilitating their interpretation. Moreover, octilinearity makes crossings of the route with polygonal landmarks more explicit. We modeled this constraint identically to the metro map model [37], nevertheless, an overview of the implementation is necessary to understand the implementation of further constraints.

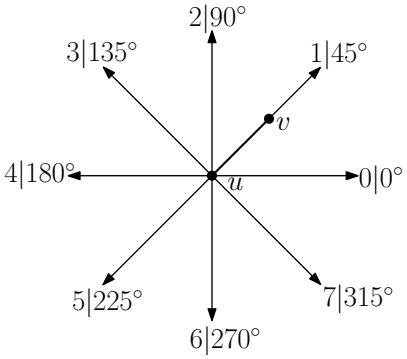


Figure 10: Octilinear edge directions. The variable dir(uv) defines the direction of uv. In this example dir$(uv) = 1$.

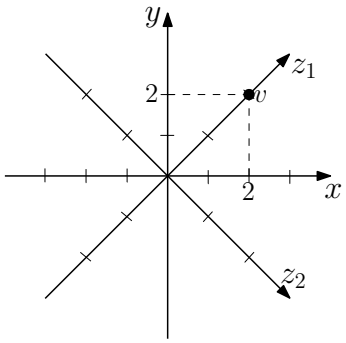


Figure 11: Octilinear coordinate system. If $x(v) = 2$ and $y(v) = 2$, then, by Equation 4, $z_1(v) = 2$ and $z_2(v) = 0$.

To guarantee that all edges get an octilinear direction ($0°$, $45°$, $90°$, $135°$, $180°$, $225°$, $270°$, $315°$), we define, for each edge $e \in E$, a supplementary integer variable dir(e) in the range $\{0, .., 7\}$. Figure 10 shows how the eight dir values are associated with the octilinear directions. The variable dir(uv) constraints the coordinates of u and v to force their alignment in the correct octilinear angle. For example, if dir$(uv) = 0$, i.e., the direction angle of uv is $0°$, the following must hold:

$$y(u) = y(v) \quad \text{and} \quad x(v) > x(u) \tag{3}$$

The dir(e) constraints for other orthogonal directions ($0°$, $90°$, $180°$ and $270°$) can be straightforwardly defined in a similar way as in equations 3. However, for the diagonal directions ($45°$, $135°$, $225°$ and $315°$), it would require trigonometric functions that are not linear and therefore cannot be used in ILP. To overcome this problem, two new redundant axes, in addition to the x and y axis, are defined in the coordinate system to represent the diagonals, z_1 and z_2 (Figure 11).

For all $v \in V$, we constrain $z_1(v)$ and $z_2(v)$ as a function of $x(v)$ and $y(v)$:

$$z_1(v) = \frac{x(v) + y(v)}{2} \quad \text{and} \quad z_2(v) = \frac{x(v) - y(v)}{2} \tag{4}$$

Now, using the z_1 and z_2 coordinates, we can constrain dir(uv) for the diagonal cases. For example, if dir$(uv) = 1$, i.e., the direction angle of uv is $45°$, the following must hold:

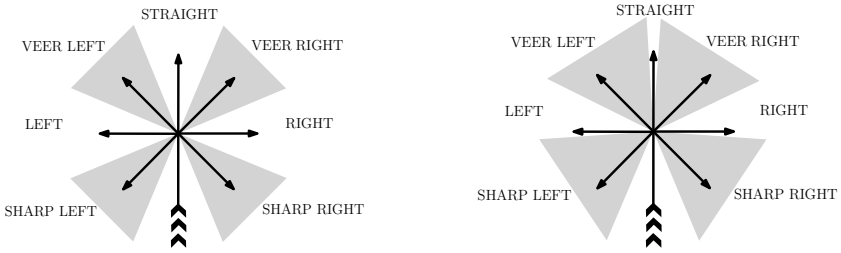
$$z_2(u) = z_2(v) \quad \text{and} \quad z_1(v) > z_1(u) \tag{5}$$

It is important to mention that dir(uv) also restricts dir(vu) to its inverse angle. For example, if dir$(uv) = 1$, then dir$(vu) = 5$. Figure 10 illustrates this example.

Finally, additional binary variables are necessary to model the case conditions statement that defines the direction of dir(e). Since dir(e) can assume eight directions, eight binary variables would be necessary for each edge; however, to reduce the number of binary variables, we allow dir to assume only one of the three octilinear directions that approximate most to the original orientation.

Hard constraint: Best turn representation at DPs

This constraint forces the turn's representations into the seven wayfinding choremes. For that, we constrain the direction of the route edges incident on DP nodes to the direction that best represents the original turn. So for each edge e incident on DP, we set dir(e) to the octilinear direction that best suits the original slope of uv. For example, if the slope of e is $40°$, we constrain dir$(e) = 1$ because this is the octilinear direction that best approximates to $40°$. To avoid the infeasibility of the model (when a solution cannot be found), we always make sure that there are at least two nodes between DPs.



(a) Turn directions based on the angle.

(b) Klippel et al.'s alternative for turn direction.

Figure 12: Klippel et al. proposed direction models alternatives for route maps [19]. Edge direction at turns can be defined by the region where the original bend lies.

It is worth noting that there are other ways to model the constraints for *best turns representation*. For instance, instead of edges' slope, we could constrain the angle between the edges incidents on the same DP. Inspired by the wayfinding choremes, we implemented several turn representation based on angles. In Figure 12, we see two alternatives to model octilinear turn directions based on the angle; notwithstanding, we observed consistently better results using the slope of the incident edges. All results presented in this paper use the edges' slope to model turns at DPs.

Hard constraint: Circular order

This *circular order* hard constraint is applied to all degree-3 or higher nodes (more than two neighbors). It preserves the circular order of the incident edges (embedding). Figure 13

illustrates a degree-k node v and its neighbours: $\{u_1, u_2, u_3, .., u_k\}$. For a topologically correct output, the circular order (indicated by the dashed line with an arrow) of the incident edges in v must be the same as the input.

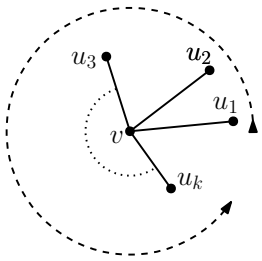


Figure 13: The circular order of the incident edges in a node must be preserved

In order to guarantee the circular order of the incident edges in v, we need to constrain dir(e) for all e incident to v as follows:

$$\text{dir}(vu_{i+1}) \geq \text{dir}(vu_i) + 1 \tag{6}$$

An exception is necessary for the last and the first edge relation (dir(vu_k) and dir(vu_1)). We omit details here, because this constraint is modeled identically to the metro map model (See [37, Section 3.2] for details).

Hard constraint: Planarity

The *planarity* hard constraint prevents pairs of non-incident edges from touching or crossing themselves (preserve planarity). Together with the *circular order* constraint, *planarity* guarantees a topologically correct schematization. We explain only the principle behind the modeling of the inequalities because it is modeled identically to the metro map model. (See [37, Section 3.3] for details).

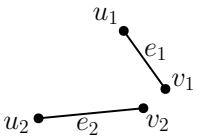


Figure 14: The conjunctive expression 7 holds true for e_1 and e_2, so e_2 is completely south from e_1.

Figure 14 shows two distinct edges, e_1 and e_2, and their respective nodes. To facilitate the explanation, let the octilinear directions be cardinal directions: $0°$ is east, $45°$ is northeast, $90°$ is north, $135°$ is northwest, and so on. To say that e_2 is completely south from e_1, the following conjunction of inequalities must hold:

$$y(u_2) < y(u_1) \quad \wedge \quad y(u_2) < y(v_1) \quad \wedge \quad y(v_2) < y(u_1) \quad \wedge \quad y(v_2) < y(v_1) \tag{7}$$

If we constrain e_2 to be completely south from e_1, we guarantee that e_2 and e_1 are not crossing. Analogous conjunctive expressions can be defined for all other 7 octilinear

cardinal directions using the x, y, z_1, and z_2 coordinates. Therefore, to guarantee that e_1 and e_2 do not cross themselves, at least one of eight conjunctive expressions must be true.

Hard constraint: Edge length

The *edge length* hard constraint is used to guarantee that all side streets edges have the same length or that all local point-like landmarks are equally spaced from the route. The constraint to fix the length of an edge e depends on the direction of e, i.e., dir(e). We present the case for dir$(e) = 0$ like illustrated in Figure 15. The constraints for other directions are modeled analogously.

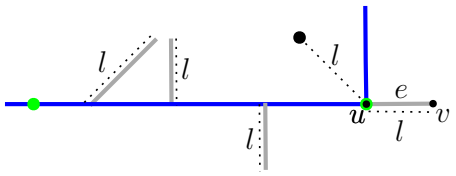


Figure 15: Hard constraint *edge length* fixes the lengths of the side streets (in gray) and local point-like landmarks (large black dot) control edges to a given value l.

Recall Equation 3 with the constraints to guarantee dir$(uv) = 0$ (and the analogous constraints for other directions). We modify this constraint by adding a parameter l that represents the desired length of uv, and by change the inequality *greater than* to *equal to* as follows:

$$y(u) = y(v) \quad \text{and} \quad x(v) = x(u) + l \tag{8}$$

Soft constraint: Node position

We use the *node position* soft constraint to minimize the distance of the output coordinates to the input coordinates of the nodes. They are applied to the route (rescaled coordinates) and to the polygonal landmarks (transposed path). The *node position* soft constraint improves coherence in the relative position among all nodes, i.e., it contributes for topographicity. This constraint is not present in the original metro map model; therefore, we describe its implementation in detail. The general idea is to add a cost in the objective function proportional to the distance of the input and output coordinates for each node.

The cost of the *nodes position* to be minimized in the objective function is:

$$\sum (d_x(v) + d_y(v)) \quad \forall v \in V \tag{9}$$

where $d_x(v) = |x(v) - x'(v)|$ and $d_y(v) = |y(v) - y'(v)|$. Note that we use Manhattan distance instead of Euclidean distance because square roots are not linear functions and therefore cannot be used in an ILP model.

We can divide our cost function into two terms, one for the x-coordinates and another for the y-coordinates and minimize them individually:

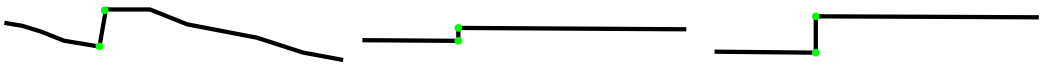
$$\sum d_x(v) \quad + \quad \sum d_y(v) \quad \forall v \in V \tag{10}$$

An absolute value operator inside an ILP model usually requires the use of a binary variable in order to treat the positive and the negative cases differently. However, an absolute value operation can be modeled using two inequalities if it is an increasing function inside an objective function to be minimized. We show the inequalities for the x-coordinates. The y-coordinates are modeled analogously:

$$d_x(v) \geq \quad x'(v) - x(v) \tag{11a}$$
$$d_x(v) \geq -(x'(v) - x(v)) \tag{11b}$$

Soft constraint: Proportion

The *proportion* soft constraint is used to minimize the difference between edge lengths in the input and output. It improves distance coherence between adjacent nodes. We also model this constraint to maintain proportionality of route section lengths, i.e., to reduce length distortions between decision points. The *node position* constraint already contributes to this proportionality, but it is not always sufficient to guarantee a balanced result in terms of route section lengths. Figure 16 contrasts a schematization *without* the *proportion* constraint for route sections with one *with* the proportion constraint for route sections.



(a) Original route path. (b) Schematization without proportion constraint. (c) Schematization with proportion constraint.

Figure 16: The effect of the *proportion* constraint for route sections.

The *proportion* constraint is not present in the original metro map model [37]. Therefore we describe it in detail here. For simplicity, we explain the constraint to keep length proportionality of all edges. Notwithstanding, the same principles can be applied to route sections; for that, consider only the DP nodes instead of all nodes.

For each edge $uv \in E$, we define a variable $\Delta p(uv)$ that expresses the difference of the resulting edge length and the input edge length as

$$\Delta p(uv) = |l'(uv) - l(uv)|, \tag{12}$$

where $l'(uv)$ is the original edge length of uv, and the variable $l(uv)$ denotes the resulting edge length

$$l(uv) = (|x(v) - x(u)| + |y(v) - y(u)|). \tag{13}$$

This time we have two absolute value operations to describe a variable in our ILP model. In this case, we need, for each of them, a pair of binary variables to model the absolute value operation in a case system of linear equations. We show our model for the x-coordinates. The y-coordinates are analogously modeled.

Let $d_x(uv) = |x(v) - x(u)|$ be the absolute value function, then $d_x(uv)$ can be expressed as

$$d_x(uv) = \begin{cases} x(v) - x(u) & \text{if } x(v) \geq x(u) \\ x(u) - x(v) & \text{if } x(u) \geq x(v). \end{cases} \tag{14}$$

To model this case distinction system in our ILP model we define for each edge a pair of binary variables $\lambda_{x1}(uv)$ and $\lambda_{x2}(uv)$ and constrain them as

$$\lambda_{x1}(uv) + \lambda_{x2}(uv) = 1. \tag{15}$$

Now we can define the following inequalities

$$d_x(uv) \leq (x(v) - x(u)) + M(1 - \lambda_{x1}) \tag{16a}$$
$$d_x(uv) \geq (x(v) - x(u)) - M(1 - \lambda_{x1}) \tag{16b}$$
$$d_x(uv) \geq -(x(v) - x(u)) - M(1 - \lambda_{x2}) \tag{16c}$$
$$d_x(uv) \leq -(x(v) - x(u)) + M(1 - \lambda_{x2}). \tag{16d}$$

Note that if $\lambda_{x1}(uv) = 1$, by Equations 16a and 16b, we get $d_x(uv) = (x(v) - x(u))$, and Equations 16c and 16d are trivially fulfilled (obsolete) because M is a large enough constant. Analogously, if $\lambda_{x2}(uv) = 1$, by Equations 16c and 16d we get $d_x(uv) = (x(u) - x(v))$, and Equations 16a and 16b are trivially fulfilled (obsolete).

Now, we can express

$$\Delta p(uv) = |l'(uv) - (d_x(uv) - d_y(uv))|. \tag{17}$$

Again we have an absolute value operation inside an objective function to be minimized. Similarly to Equation 11, we use two constraints to model Equation 17.

$$\Delta p(uv) \geq l'(uv) - d_x(uv) + d_y(uv) \tag{18a}$$
$$\Delta p(uv) \geq -l'(uv) + d_x(uv) - d_y(uv). \tag{18b}$$

Soft constraint: Bend minimization

The *bend minimization* soft constraint reduces the number of bends along the route. It diminishes unnecessary complexity along the route path in order to facilitate path following by the user. For polygonal landmarks, it contributes to increasing abstraction. We model this constraint in the same way it is modeled to reduce bends along the transit lines in the metro map model [37]. For this reason, we only give an overview of its implementation (See [37, Section 3.5] for details).

Let P be a path (the route path or a polygonal landmark path) and v_b a degree-2 node in P. The cost of the *bend minimization* in the objective function is:

$$\sum b(v_b) \quad \forall v_b \in P, \tag{19}$$

where $b(v_b)$ represents the individual cost of the bend formed by the incident edges in v_b, and is calculated as

$$b(v_b) = \begin{cases} |\text{dir}(u_1v_b) - \text{dir}(v_bu_2)| & \text{if} \quad |\text{dir}(u_1v_b) - \text{dir}(v_bu_2)| \leq 4 \\ 8 - |\text{dir}(u_1v_b) - \text{dir}(v_bu_2)| & \text{if} \quad |\text{dir}(u_1v_b) - \text{dir}(v_bu_2)| \geq 5. \end{cases} \tag{20}$$

Let u_1 and u_2 be the two nodes adjacent to v_b, as exemplified in Figure 17. In this example we have $\text{dir}(u_1v_b) = 0$ and $\text{dir}(v_bu_2) = 5$, so $|\text{dir}(u_1v_b) - \text{dir}(v_bu_2)| = 5$; then, by Equation 20, $b(v_b) = 3$. Equation 20 has the property to give higher costs to acute bend angles compared to obtuse ones: 0 for $0°$, 1 for $135°$, 2 for $90°$, and 3 for $45°$.

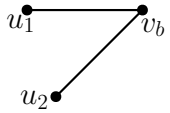


Figure 17: The bend angle formed by edges u_1v_b and v_bu_2 is $45°$. By Equation 20, the cost of this bend ($b(v_b)$) is 3.

Soft constraint: Edge orientation

The soft constraint *edge orientation* is used to minimize the difference of edges' direction between the input and output. This constraint counterbalances potential undesired distortion caused by the *bend minimization* constraint. For the route path, it improves the sense of direction along the route path, and, for the polygonal landmark paths, it contributes to a more genuine shape (less abstract). We modeled this constraint identically to the *edge direction* constraint in the metro map model [37, Section 3.6].

Let $\epsilon(e)$ be a binary variable constrained as

$$8\epsilon(e) \leq \text{dir}(e) - \text{dir}'(e) \leq 8\epsilon(e) \quad \forall e \in E, \tag{21}$$

where $\text{dir}'(e)$ is the octilinear direction angle of e that approximates best its original slope. For instance, if the slope angle of e is $40°$, then $\text{dir}'(e) = 1$. Because $|\text{dir}(e) - \text{dir}'(e)| < 8$, by Equation 21, $\epsilon(e) = 0$ if and only if $\text{dir}(e) = \text{dir}'(e)$, otherwise $\epsilon(e) = 1$.

The cost of the *edges orientation* in the objective function is:

$$\sum \epsilon(e) \quad \forall e \in E. \tag{22}$$

3.5 Route and point-like landmarks schematization

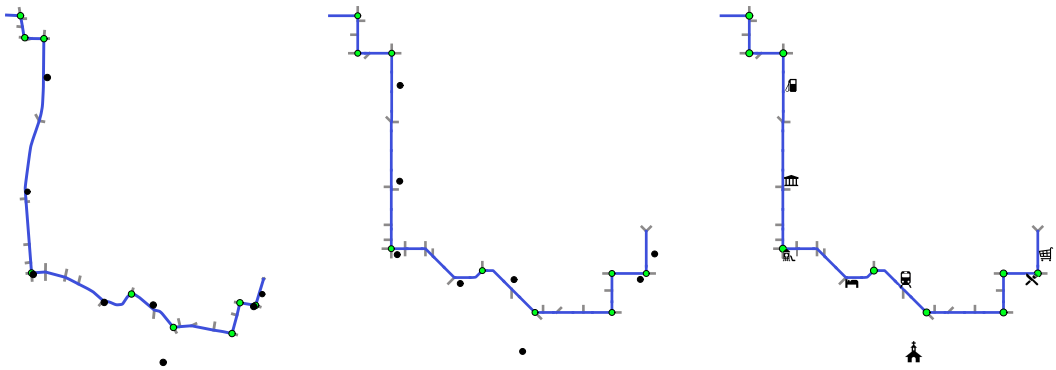
The ILP model for the route and point-like landmarks schematization gets as input the route path, its adjacent streets, and the control edges of the local point-like landmarks. In order to reduce the size of the instance, we simplify the route path using the Douglas-and-Peucker simplification method [10] to remove some degree-2 nodes. We use a sufficiently low enough tolerance in the simplification to keep the topographicity of curves along the route. we guarantee that at least two nodes remain between intersections (degree-3 or higher nodes). Keeping these two nodes improves the final representation of intersections and guarantees the feasibility of ILP in respect to the *best turns at DP* hard constraint.

ILP model hard constraints are *planarity*, *circular order*, *octilinearity*, *best turn at DP*, and *edge length*. The *edge length* fixes the lengths of the adjacent street edges and the local point-like landmark control edges to a fixed value. The longer the edges, the higher the chances of planarity violations, which potentially increases the execution time. The *circular order* and *planarity* constraints guarantee topologically correct results. However, the *planarity* constraint represents to the whole model 32 constraints and eight binary variables for each pair of non-incident edges. Since the number of non-incident edges pairs is roughly $m^2/2$, where m is the total number of edges, planarity is the most costly constraint of the model. In order to reduce the runtime of our application, we implement this constraint in a "lazy" way as used by Delling et al [9]. First, the instance is schematized without the *planarity* constraint, and, while the output contains edge crossings, we rerun the ILP only including in the *planarity* constraint for pairs of edges that crossed in the previous executions.

There are five soft constraints in the ILP model: *node position* (dist), *edge orientation* (dir), *bend minimization* (bend), *proportion* for route edges (prop), and *proportion* for route sections between DPs (propDP). The objective function is defined as follow:

$$\text{Minimize} \quad \frac{\alpha_{dist}}{n}\text{cost}_{\text{dist}} + \frac{\alpha_{\text{dir}}}{m}\text{cost}_{\text{dir}} + \frac{\alpha_{\text{bend}}}{b}\text{cost}_{\text{bend}} + \frac{\alpha_{\text{prop}}}{m}\text{cost}_{\text{prop}} + \frac{\alpha_{\text{propDP}}}{s}\text{cost}_{\text{propDP}} \tag{23}$$

The α values are parameters to weight the cost of each soft constraint. The variables n, m, b, and s are the number of nodes, edges, bends (degree-2 or higher nodes), and route sections, respectively; they normalize the given weights. Note that we use the *proportion* constraint twice. One is used to keep the length coherency of all edges, and the other applied only for the route sections.



(a) Original layout (b) Schematized layout (c) Dots replaced by icons

Figure 18: Example of a route schematized with point-like landmarks

Figure 18 demonstrates the results of the route schematization with point-like landmarks (black dots). Note that in the schematized layout (Figure 18b), landmarks along the route are evenly spaced away from the route, creating enough space to replace the point geometries with icons, as shown in Figure 18c.

3.6 Path transposition: adjustment of control edges' lengths

The control edges are used to preserve the relative positions of the polygons to the route. After the route is schematized, we use linear transformation to transpose the polygonal landmark paths to the schematized route using the nodes of the control edges as control points (process box 3 in Figure 3). However, because of the rescaling of the route, we need to rescale the control edges too. The scale along the schematized route varies from section to section, meaning that the scale factor cannot be the same for all control edges. For a coherent rescaling, we adjust the length of the control edges based on the length of the schematized route edges. Let E_r be the set of route edges, and $l'(e_r)$ and $l(e_r)$ the lengths of the original and schematized route edge $e_r \in E_r$, respectively. The ratio of the new length for each control edge is calculated as:

$$\sum \frac{l(e_r)}{d(e_r)} \Big/ \sum \frac{l'(e_r)}{d(e_r)} \quad \forall e_r \in E_r \quad \text{and} \quad d(e_r) < 0.8, \tag{24}$$

where $d(e_r)$ is the normalized (scaling $[0, 1]$) distance of the route edge e_r to the control edge. Because we weight each route edge by the inverse of its distance to the control edge, route edges closer to the control edges will have a higher influence on the resulting length of the control edge. Moreover, we ignore route edges that are too far away from the control edge ($d(e_r) \geq 0.8$).

3.7 Polygonal landmark path schematization

The ILP for polygonal landmark paths schematization gets as main input the transposed geometries of the polygons obtained from a linear transformation. The linear transformation uses crossing nodes or the control edges nodes as control points (process box 3 in Figure 3). Figure 19b illustrates the transposed paths (black lines) as a result of this process.

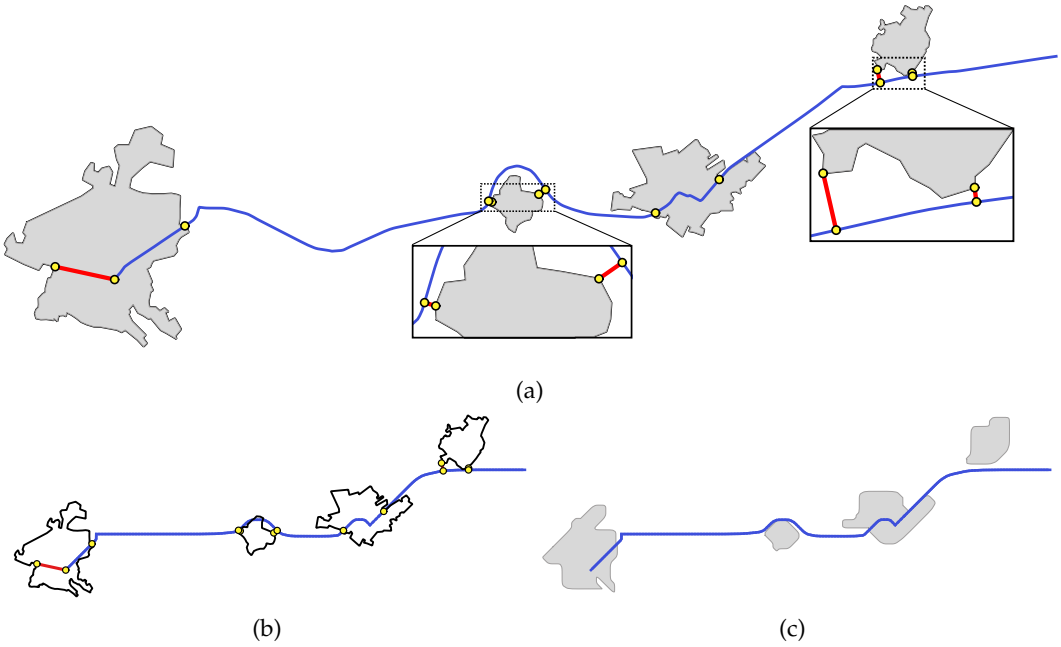


Figure 19: Polygon schematization workflow: (a) Before the route is schematized we create control nodes (yellow dots) and control edges (red lines) for each polygon. (b) After the route is rescaled and schematized, we adjust the control edges' length in order to use the control nodes in the affine transformation to transpose the polygons (black outlines). (c) Finally, the transposed polygons are schematized using ILP

Before running the ILP to schematize the polygons, we reduce the number of nodes of the polygon to reduce the ILP run time. Reducing the number of nodes is done using Douglas-Peucker simplification with the simplification tolerance limited in order to preserve the perceptual morphology of landmarks. This guarantees that the control nodes are maintained. Other simplification methods are equally possible such as discrete curve evolution [4] which shows good performance regarding the perceptual appearance of the polygon. Our method is independent of the simplification method. However, a simplifica-

tion method such as discrete curve evolution [4] could improve results, because it preserves better the perceptual appearance of the polygon.

The ILP model hard constraints for the polygonal landmark path schematization are *octilinearity* and *planarity*. We use the same iterative planarity check as for the route ILP. First, the instance is schematized without the planarity constraint, and, while the output contains edge crossings, we rerun the ILP including the planarity constraints for those pairs of edges that crossed in previous executions.

There are five soft constraints in the polygon schematization ILP model: *node position* (dist), *node position* for control nodes only (distCN), *edge orientation* (dir), *bend minimization* (bend), and *proportion* (prop). The objective function is defined as:

$$\text{Minimize} \quad \frac{\alpha_{\text{dist}}}{n}\text{cost}_{\text{dist}} + \frac{\alpha_{\text{distCN}}}{c}\text{cost}_{\text{distCN}} + \frac{\alpha_{\text{dir}}}{m}\text{cost}_{\text{dir}} + \frac{\alpha_{\text{bend}}}{b}\text{cost}_{\text{bend}} + \frac{\alpha_{\text{prop}}}{m}\text{cost}_{\text{prop}} \quad (25)$$

The α values are the parameters to weight the cost of each soft constraint. The variables n, m, b, and c are the number of nodes, edges, bends (non degree-1 nodes), and control nodes respectively; they normalize the given weights. Note that we use a second *node position* constraint only for the control nodes. This way we can discriminate the weight for the *node position* between control nodes and ordinary nodes.

The polygonal landmark ILP schematization is used to meet four goals: (i) reduce the shape complexity (abstraction); (ii) guarantee the correct topology with previously schematized features; (iii) emphasize spatial relations (crossings and alongness), and (iv) control the dimension of the landmarks.

Control of abstraction

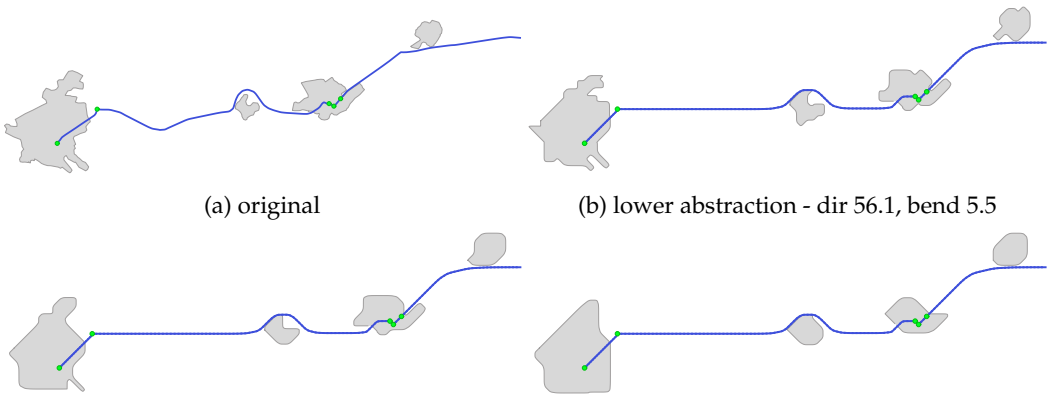
Abstraction involves reducing the shape complexity of the map's features. It is one of the main characteristics of schematic visualizations. Abstraction contributes to a cleaner layout by removing unnecessary information, driving the attention of the user to more functional elements of maps. Nevertheless, abstraction reduces recognizability of landmarks because it leads to severe changes in the original geometry.

We control the level of abstraction by weighting the cost of the soft constraints. The constraint *bend minimization* increases abstraction, and constraints *edge orientation* and *node position* increase topographicity. Figure 20 illustrates the abstraction's effect on the landmark geometries that results from variations in the soft constraints' weights.

The weights for the soft constraints can also be dynamically adjusted. This means that the level of abstraction can depend on attributes of landmarks, such as type, size or distance to the route.

Fixing topological inconsistencies

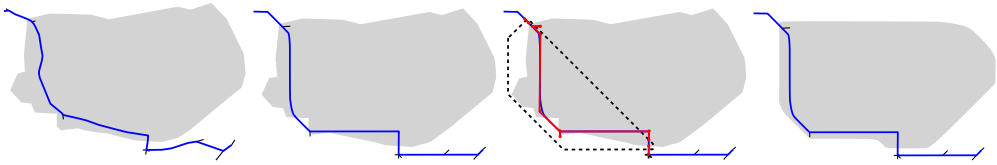
Merely transposing the polygon using an affine transformation does not guarantee topological consistency with previously schematized features. Figure 21b shows a polygon transposed to a schematized route. Note that the botton-left part of the polygon contains crossings with the route nonexistent in the original layout (Figure 21a). The *planarity* constraint can prevent such crossings in the resulting layout. For that, we need to input previously schematized edges from the route and other polygonal landmarks into the ILP process (process box 4 in Figure 3).



(a) original (b) lower abstraction - dir 56.1, bend 5.5

(c) medium abstraction - dir 45.1 - bend 30.5 (d) higher abstraction - dir 12.1 - bend 45.5

Figure 20: Balancing the level of abstraction by varying the weights of the soft constraints. The caption values are the weights for the *edge orientation* and *bend minimization* (α_{dir} and α_{bend} in Equation 25). The weights for the remaining constraints are constant (1580, 790, and 500 for α_{dist}, α_{distCN}, and α_{prop} respectively)



(a) (b) (c) (d)

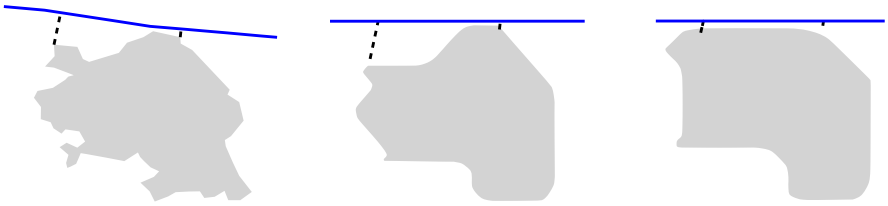
Figure 21: ILP fixes topological inconsistencies. (a) Original route crosses the polygon shape, (b) The polygon transposed to the schematized route contains extra edge crossings. (c) Octilinear bounding box (dotted line), and detection of edges (red) to avoid crossing. (d) Schematized polygon with correct topology

Every shape schematized in the ILP is bounded by an octilinear box defined by the limit of the octilinear coordinates. This way, it is necessary to check for planarity violations only for the edges intersecting this bounding box region. Figure 21c illustrates the octilinear bounding box (black dotted lines) that limits the position of the path nodes, and the edges of the schematized route (in red) that intersects the bounding box region. The resulting schematization with correct topology is shown in Figure 21d.

Emphasizing spatial relation

Landmarks along the route can be used in spatial chunks (e.g., go along the park and turn left). We can increase alongness (ratio of the region boundary being parallel to a path) with the route by reducing the length of its pair of control edges before transposing the polygon. Figure 22 illustrates the effect of this tighter adjustment. Because we shortened the control edges (little black line segments) in Figure 22c, the polygonal landmark moved closer to

the route after the transposition of its path. As a result, its schematic layout has a higher degree of parallelism with the route compared to Figure 22b.

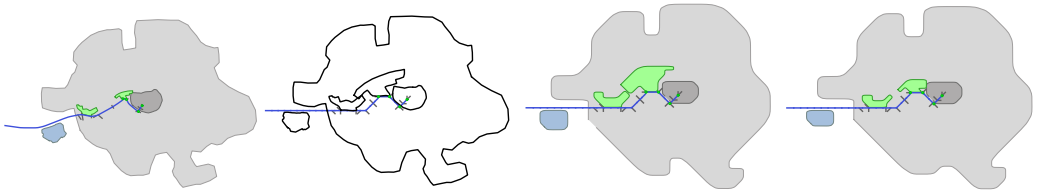


(a) original shape. (b) regular. (c) tight.

Figure 22: Emphasizing alongness: the control edges (dashed black lines) for (C) are shortened.

Scale control

The polygonal landmark ILP schematization can be used to control the size of the landmarks. For example, a landmark along a route section with an enlarged scale can have its size extended beyond the desired limit. Figure 23c illustrate such a case. Note that the parks have their size over-exaggerated compared to the original layout (Figure 23a). This happens because the route sections along the parks have their scale relatively enlarged after the route rescaling transformation, and this has an effect on the scale in the transposed landmark paths (black outline in Figure 23b).



(a) original. (b) transposed paths. (c) no scale control. (d) with scale control.

Figure 23: Effect of scale control.

We can control the size of landmarks using the *edge proportion* soft constraint. For that we add a parameter q to the Equation 12:

$$\Delta p(uv) = |q * l'(uv) - l(uv)|. \tag{26}$$

The parameter q modifies the resulting length of the edges. If q is equal to 1 the edges lengths will tend to have the same length ($l'(uv)$) as the respective edges of the input (transposed path). We calculate q by, first, estimating the ideal length of the landmark path using the same equation for the adjustment of the control edges (Equation 24 in Section 3.6). Then, the value q is calculated as the ratio of the ideal length over the transposed path length. Figure 23d shows the result of scale control of the landmarks along the route using

the parameter q. For instance, the rightmost park, q was set to 0.73, i.e., its edges' lengths are preferred to be 73% of the respective input edges' lengths.

The last change to control the scale of a landmark is to remove the *node position* constraint (setting $\alpha_{\text{dist}} = 0$ in Equation 25). We remove *node position* because we want to change the extension of the input path (transposed). Therefore the minimization of the nodes distances to the input positions is not necessary. We keep only the *node position* constraint for the control nodes (distCN in Equation 25). Then only the control nodes (landmark control edge nodes) will have their distance to the respective input nodes minimized.

4 Results

We present the results in detail for two routes. Results for more routes can be found in the appendix. The first example you find in this section is a route that connects two cities and passes by several landmarks and other cities (Figure 24). This route is longer and allows us to illustrate the main characteristics of our method (pointed with numbered annotations). The second example (Figure 25) is a shorter route that crosses a city center. This second example allows us to show a typical result on a larger scale. Also different from the first, it is an example where topological inconsistencies were found and fixed (Figure 21 presents a similar case). Complementary to the maps, we provide tables containing details about the instances and executions.

The first example is a 33.7km long route between two cities (Figure 24). The instance submitted to schematization is composed of 317 nodes, of which 172 belong to the route, adjacent streets, and the point-like landmarks, and 145 to the polygonal landmarks. We count here only the number of nodes after the Douglas-Peucker simplification, i.e., the actual size of the instance submitted to ILP process. The total execution time was 6.189 seconds, of which 5.2 seconds for the route, and 0.983 for 8 landmarks. We ran our application using an 8GB RAM Intel Core i7 2.8 GHz Windows 10 Laptop and IBM CPLEX®V12.7.1 to solve the ILP model. Also, for all ILP executions, we set the relative MIP gap tolerance to 0.1, i.e., CPLEX stops the execution as soon as it finds a feasible integer solution proved to be within ten percent of optimal.

General								
nodes	edges	rescaled	paths	poly. LM	point LM	fixed cross.	ILP exec.	runtime
317	304	0.77	12	8	13	0	13	6189ms

Route										
nodes	edges	DPs	stubs	α_{bend}	α_{dir}	α_{dist}	α_{prop}	α_{propDP}	ILP exec.	runtime
172	171	12	60	100	43.5	43.5	100	200	1	5206ms

Polygonal landmark paths									
paths	nodes	edges	α_{bend}	α_{dir}	$\alpha_{\text{dist}}{}^{*}$	α_{distCN}	α_{prop}	ILP exec.	runtime
12	145	13	25	52.8	1000	500	500	12	983ms

Table 1: Schematization summary of Figure 24.

Table 1 shows the main information on the instance and the weights used in the objective function. The weights are shown as the α values used in the Equation 23 for the route, and in the Equation 25 for the polygonal landmark paths. The 12 polygonal landmark paths

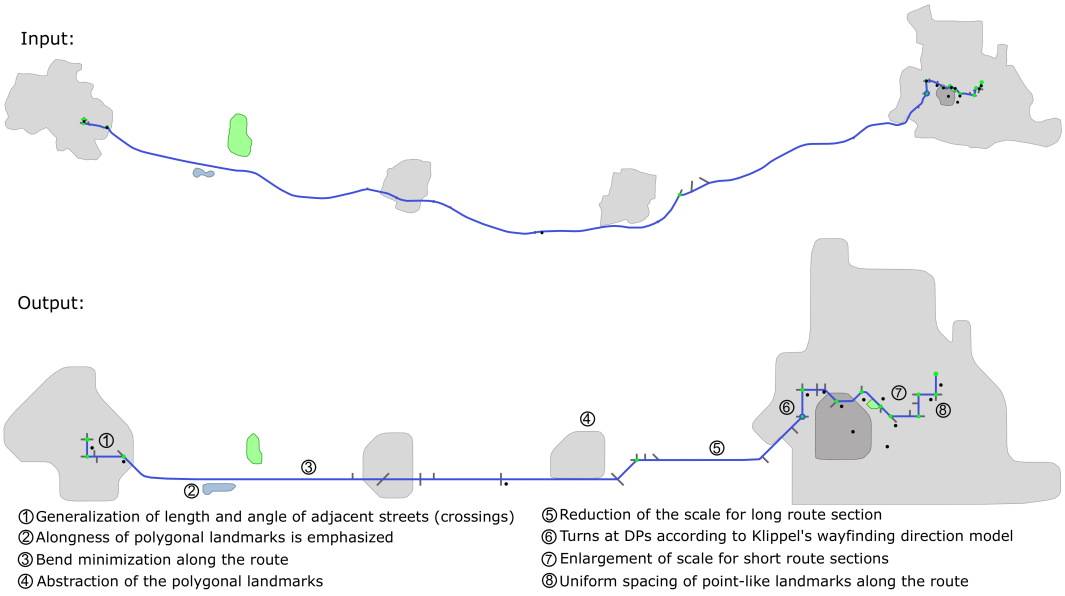
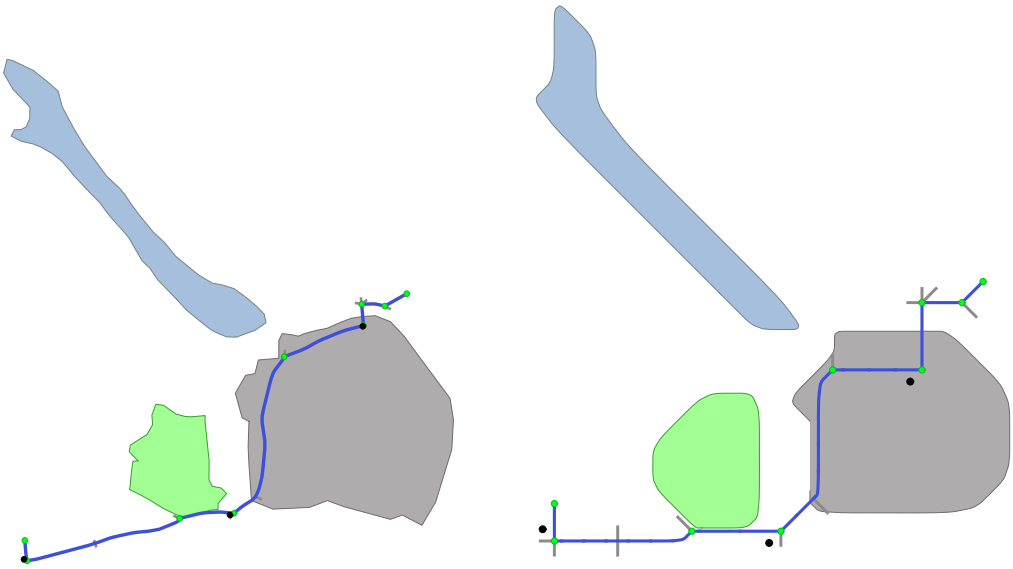


Figure 24: Example of input and output of our method with annotations.

are individually schematized but, to be more concise, we summarize their schematization information into a single table. For that we use the same weights in the objective function for all landmarks, except for the *node position* constraint of the landmarks along the route. For them, the value α_{dist} is zero because of the scale control explained in Section 3.7.

Yet in Figure 24, we compare the original with the resulting layout in the same space extension. In order to facilitate the description of the schematic layout criteria effects, we added reference annotations (numbers inside circles) to the input figure. Note that, for parts of the route with a higher density of DPs and landmarks, the scale is larger (7), and for parts of the route with fewer elements the scale is smaller (5); that way, short route sections and small polygonal landmarks are more visible in the schematic layout. Unnecessary bends along the route path were removed (3), and important crossings and turns were made more evident (1, 6), in line with the wayfinding choremes theory [22]. The placement of point-like landmarks (black dots) is consistent with the original (8); those that were too close were pushed away from the route, making it easier to identify on which crossing corner or which side of the route they are located. As for the polygonal landmarks, their shape was simplified making them more abstract (4), their crossing with the route became more orthogonal and, for those along the route, the parallelism became more evident (2).

The second example is a 3.7km route (Figure 25). This instance is composed of 135 nodes, of which 78 belong to the route, adjacent streets, and the point-like landmarks, and 57 to the polygonal landmarks. The total execution time was 2.281 seconds, of which 0.7 seconds for the route, and 1.58 seconds for 3 polygonal landmarks. Note that in this example 9 ILP executions were required for the 4 polygonal landmark paths (Table 2). The extra ILP executions were necessary to fix edge crossings that violated the topology as identified after the first executions. This example also shows the characteristics described in the



(a) input (b) output

Figure 25: Example of input and output. The route goes along a park (green) and crosses the city center (gray). The lake (blue) is a global landmark. There are seven DPs and three point-like landmarks.

General								
nodes	edges	rescaled	paths	poly. LM	point LM	fixed cross.	ILP exec.	runtime
135	130	0.5	4	4	3	7	10	2281ms

Route										
nodes	edges	DPs	stubs	α_{bend}	α_{dir}	α_{dist}	α_{prop}	α_{propDP}	ILP exec.	runtime
78	77	7	15	50	75	75	100	75	1	705ms

Polygonal landmark paths									
paths	nodes	edges	α_{bend}	α_{dir}	$\alpha_{\text{dist}}{}^{*}$	α_{distCN}	α_{prop}	ILP exec.	runtime
4	57	53	25	48.4	1000	500	500	9	1576ms

Table 2: Schematization summary of Figure 25. Five extras ILP executions were necessary to fix 7 edge crossings that violated the topology.

first example in a larger scale. Note that street junctions, turn directions, and topological relations with landmarks are more clear in the schematic route map.

4.1 User study: readability of route instructions on schematic layout

We conducted an empirical experiment to evaluate the preference of users between a non-schematic and a schematic route map in a route instructions matching task. The experiment consisted of personal judgments, and the goal was to understand if the proposed route schematization facilitates the readability of route information.

Participants. We collected data from 27 participants (17 female, 10 male; average age of 29.56 years). Most of them were recruited via social media groups and a general university mailing list. The study was approved by the institute's ethics committee. Participants signed informed consent forms before participating and received a EUR 5 compensation for their collaboration.

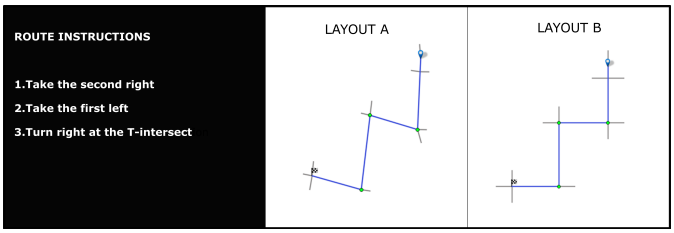
Materials. We prepared 19 question sheets, each for a single route. Each route was presented in a landscape orientated A4 sheet. Each sheet was divided into three boxes. On the left side, covering one-third of the page, there was a column containing route instructions. The rest of the page was vertically or horizontally (depending on the extension of the route dimensions) divided into two equally-sized boxes. Each box contained either the original or the schematized layout. The left-right location of the box containing the schematic version was randomised. Figure 26 illustrates three examples of routes used in the experiment.

The routes were calculated and schematized using the system and street data described in our methodology (Section 3). We selected the routes varying the levels of complexity (from 3 to 9 DPs) and varying the spatial relations with polygonal and point-like landmarks. The landmarks are geometries representing real geographic features, we only vary its type, e.g., we refer to a village as park or a lake in some routes. In order to evaluate different aspects of the algorithm we divided the routes into three focus group: (i) schematization of the route and side streets (no landmarks, as seen in Figure 26a), (ii) route and placement of the point-like landmarks (no polygonal landmarks, as seen in Figure 26b), and (iii) spatial relations of the route with the polygonal landmarks (no point-like landmarks, as seen in Figure 26c).

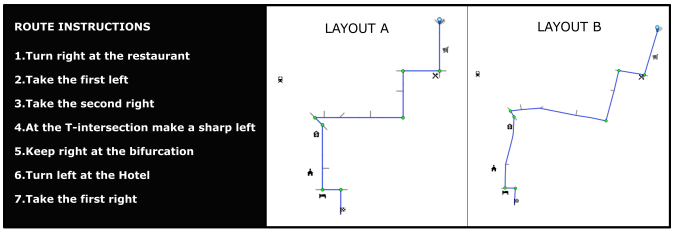
Experimental Design and Procedure. Participants were invited to sit in a regular office desk, read the experiment instructions, and read the map features legend. After that, each participant had to evaluate 19 routes. We presented the participants with a textual route description composed of an ordered list of instructions, and the two corresponding map layouts: the non-schematic version (with the input coordinates), and a schematic version generated by our algorithm. Then, we asked participants to read the instructions and follow them in both maps. Afterwards, participants were asked to answer two questions: *(1) On which layout is the route instructions easier to follow?* and *(2) Which layout symbolizes better the route instructions?* Participants needed to provide their preference on a separate answer-sheet in a forced-choice manner, i.e., they could select only one layout to answer each question.

Results. 27 participants provided a total of 513 evaluations. Table 3 shows the results of the evaluation reported by the participants divided by the aspect focus. In total, in 82.8% of the evaluations, participants reported route instructions in the schematic layout to be easier to follow, and in 74.5% of the evaluations, participants thought that the schematic layout symbolizes route instructions better.

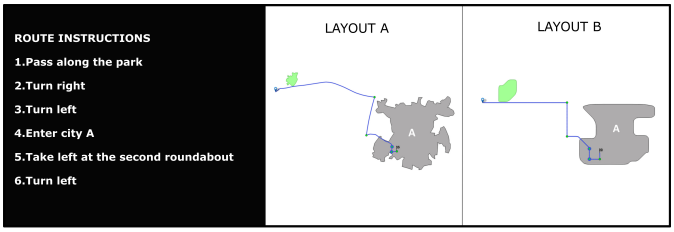
If the participants were indifferent to the map layout, we would expect a mean value closer to 50%. We tested whether these differences were statistically significant by performing a test of equal proportions (a one-sample binomial test) against the null hypothesis that the preference for schematized and non-schematized layout is equal. The Result was significant for both questions, and for all aspects (Table 4). This confirms that the preference for the schematic layout was significantly higher than for the non-schematic layout, in all three aspect focus categories. We did not find any significant difference between genders.



(a) route and side streets.



(b) point-like landmarks.



(c) polygonal landmarks.

Figure 26: Example of routes used in the experiment: in Figure 26a the focus is on the route and side streets only, in Figure 26b the focus is on the point-like landmarks, and in Figure 26c the focus is on the polygonal landmarks.

	Easier to follow			Symbolizes better		
Aspect focus	Original	Schematic	Total	Original	Schematic	Total
Route only	19	143	162	32	130	162
Point-like landmarks	23	112	135	28	107	135
Polygonal landmarks	46	170	216	71	145	216
All routes	88	425	513	131	382	513

Table 3: Evaluation results. Participants layout preference (original vs. schematic) on two questions: *On which layout are the route instructions easier to follow? Which layout symbolizes better the route instructions?*

5 Discussion

Schematic cartography makes use of generalizations to improve specific functionalities of a map. As discussed in the introduction, we are investigating visualizations whose primary functions are to facilitate the readability of route information and the acquisition of survey knowledge. As presented in the Results section, our route schematization method

	Easier to follow			Symbolizes better		
Aspect focus	χ^2	df	p	χ^2	df	p
Route only	93.39	1	$< .001$	58.08	1	$< .001$
Point-like landmarks	57.36	1	$< .001$	45.07	1	$< .001$
Polygonal landmarks	70.04	1	$< .001$	24.67	1	$< .001$

Table 4: Tests of equal proportions demonstrating statistically significant difference in the choices for schematized layouts, for both questions and all three aspects.

achieved the layout criteria for a functional route map, i.e., emphasized turns at decision points, crossings, elements for spatial chunking [21], and the route's spatial relation with contextual information [41].

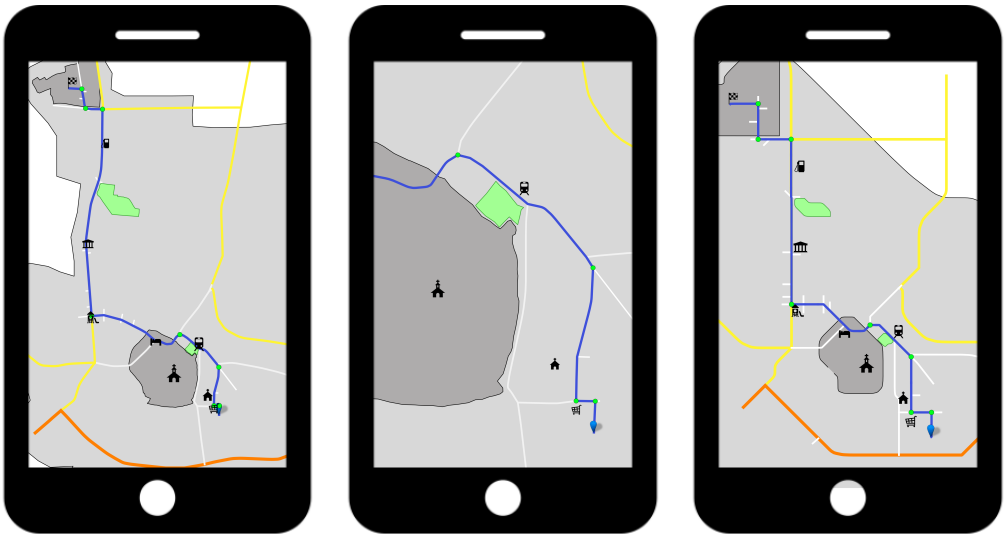
Our method cares strictly for the transformation of geometric route features to cognitively adequate schematizations. The problem of selecting the streets and landmarks that are useful for schematic route maps is out of the scope. Richter [41] defined a system that analyses route environments and generates spatial chunks (abstracts route directions or instructions) with selected landmarks. Our method allows drawings of route directions and spatial chunks with point-like and polygonal landmarks in the same level of granularity required by Richter's system. We expect that automatic schematization of route maps can be applied in dynamic route information applications. For that, we suggest for future work to test our method combined with methods of automatic selection of landmarks for route maps.

We conducted a user study that indicates the preference of users for our schematized layout in following route instructions. In total, 27 participants evaluated 19 different route maps presented in two version, non-schematic and schematized by our method. In 82.8% of the evaluations, participants reported that following route instructions in the schematic maps was easier. This user study cannot access objective measurements of usability or performance between different map layouts and their characteristics. A separate user study with simulated wayfinding and navigational scenarios is required to obtain more accurate evaluations.

In order to discuss the advantages of the schematic layout for survey knowledge acquisition, we created a montage of maps within a small-display device frame (Figure 27). We used that to compare the visibility of the surroundings. Besides the global landmarks, we also included parts of the surrounding street network, which are not in the scope of the method, but they help to illustrate the case. Note that at a first look, both maps, the original layout (Figure 27a) and the schematic layout (Figure 27c), are similar. However, in the original layout, some turns at DPs are not easily recognizable. Recognizing those turns would require a zoom-in operation, as shown in Figure 27b, but in this case, the user would lose the overview of the whole route and its surrounding geographical context. In contrast, using the schematic layout, it is possible to recognize all turns and have a broader overview of the region within a single zoom level. This characteristic is mainly due to the scale variation that allows a larger scale for smaller elements or dense areas. The capability of maps with scale variation to improve the practical use of maps in small screen devices is well discussed in [12, 16, 29], it allows details of important parts to be shown together with the surroundings.

A known disadvantage of the scale variation in cartography is the misinterpretation of geographic nearness by the map user [13]. The perception of geographic nearness from

graphical proximity in geovisualization is a topic that yet requires a better comprehension [15]. Nevertheless, we hope that the diagrammatic layout of the schematizations can work as a "visual cue" of geographic inaccuracy [32], while extra annotations (e.g., labels or marks) can be used as symbols to indicate accurate distances.



(a) original route overview. (b) original route zoom-in. (c) schematized route overview.

Figure 27: Practical applicability of our schematic layout in small screen devices. In the topographic map (a) it is possible to have an overview of the route and of the surroundings, but turns at DP are not recognizable; or when zoomed (b) the turns are recognizable, but few of the surroundings are visible. In the schematic map (c) all turns are recognizable and, in the same canvas, a lot of the surrounding is visible.

We expect that cognitively adequate route schematizations with scale variation contribute to survey knowledge acquisition in visual means with limited viewport, and that they could be a suitable informative tool for wayfinding applications that fosters orientation, as discussed by Schwering et al. [46]. We suggest as future work a behavioral study that simulates wayfinding and navigation tasks using route maps in small devices in order to objectively assess the level of survey knowledge acquisition with the schematic route layout.

As for current limitations, our algorithm is not applicable to polygons structured as subdivisions. This will be addressed in future work. Another important feature we consider essential for a functional route map, and not covered in this paper, is the surrounding street network. The street network, like global landmarks, gives an extra dimension to the one-dimensional route path, promoting orientation and survey-knowledge aquisition. In Figure 27 we showed a mock-up example that includes a naive network schematization just for a matter of illustration. For future work we propose network schematization aimed for route maps that highlights major street structures of cities or broader regions, such as rings, radial avenues (exits) and grids. Kopf et al. [23] and van Dijk and Haunert [48] addressed the problem of adapting a road network to maps with scale variation. The challenge for our case is to develop a method for the network schematization that gets as input,

in addition to the network data, a route path inside the network, and the schematic route coordinates that need to be preserved. Both methods, Kopf et al. [23] and van Dijk and Haunert [48], are not straightforwardly applicable to this formulation of the problem of the network schematization. Therefore, further investigations are required.

6 Conclusion

The goal of this paper was to present a method for the transformation of geometric route features into schematizations that facilitates route and survey knowledge acquisition. Current route schematization methods do not comply with principles of cognitive adequacy or overlook contextual information such as side streets, point-like landmarks, and, especially, regional landmarks. Contextual information is relevant to both route and survey information acquisition. In this contribution, we presented a route schematization method that includes landmarks and side streets.

Our method makes use of ILP and geometric transformations to emphasize route information (crossings, turns at decision points) and spatial relations between local and global landmarks. The schematization follows principles of ergonomics for route information interpretation and survey knowledge acquisition. To evaluate the applied layout rules, we conduct an empirical study that shows that users prefer to follow route instructions with our schematized features over the original shape. We expect that automatic schematization of route maps can support dynamic route information applications that contribute to the spatial learning of map users.

Acknowledgments

This work was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 637645) and by the Austrian Science Fund (FWF) Grant P 31119. Map data copyrighted OpenStreetMap contributors and available from https://www.openstreetmap.org .

References

[1] AGRAWALA, M., AND STOLTE, C. Rendering effective route maps: improving usability through generalization. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), vol. 1 of *SIGGRAPH '01*, ACM, pp. 241–249. doi:10.1145/383259.383286.

[2] ANACTA, V. J. A., SCHWERING, A., LI, R., AND MUENZER, S. Orientation information in wayfinding instructions: evidences from human verbal and visual instructions. *GeoJournal 82*, 3 (2017), 567–583. doi:10.1007/s10708-016-9703-5.

[3] AVELAR, S. *Schematic maps on demand: design, modeling and visualization*. PhD thesis, ETH Zurich, 2002.

[4] BARKOWSKY, T., LATECKI, L. J., AND RICHTER, K. F. Schematizing maps: Simplification of geographic shape by discrete curve evolution. In *Spatial Cognition II: Integrating*

Abstract Theories, Empirical Studies, Formal Methods, and Practical Applications (Berlin, Heidelberg, 2000), C. Freksa, C. Habel, W. Brauer, and K. F. Wender, Eds., Springer Berlin Heidelberg, pp. 41–53. doi:10.1007/3-540-45460-8_4.

[5] BUCHIN, K., MEULEMANS, W., RENSSEN, A. V., AND SPECKMANN, B. Area-preserving simplification and schematization of polygonal subdivisions. *ACM Transactions on Spatial Algorithms and Systems 2*, 1 (2016), 1–36. doi:10.1145/2818373.

[6] DA SILVA, A. C., AND WU, S.-T. A robust strategy for handling linear features in topologically consistent polyline simplification. In *GEOINFO 2006 - 8th Brazilian Symposium on GeoInformatics* (Campos do Jordao-SP, 2006), p. 15.

[7] DE BERG, M., VAN KREVELD, M., AND SCHIRRA, S. Topologically correct subdivision simplification using the bandwidth criterion. *Cartography and Geographic Information Systems 25*, 4 (1998), 243–257. doi:10.1559/152304098782383007.

[8] DELLING, D., GEMSA, A., NÖLLENBURG, M., AND PAJOR, T. Path schematization for route sketches. In *Scandinavian Workshop on Algorithm Theory* (2010), Springer, pp. 285–296. doi:10.1007/978-3-642-13731-0_27.

[9] DELLING, D., GEMSA, A., NÖLLENBURG, M., PAJOR, T., AND RUTTER, I. On d-regular schematization of embedded paths. *Computational Geometry 47*, 3 (2014), 381–406. doi:10.1016/j.comgeo.2013.10.002.

[10] DOUGLAS, D. H., AND PEUCKER, T. K. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: the international journal for geographic information and geovisualization 10*, 2 (1973), 112–122. doi:10.3138/FM57-6770-U75U-7727.

[11] GALVAO, M., RAMOS, F., LAMAR, M., AND TACO, P. Dynamic visualization of transit information using genetic algorithms for path schematization. In *Dynamics in GIscience. GIS OSTRAVA 2017* (2017), I. Ivan, J. Horak, and T. Inspektor, Eds., Lecture Notes in Geoinformation and Cartography, Springer, Cham, pp. 99–113. doi:10.1007/978-3-319-61297-3_8.

[12] GODFREY, L., AND MACKANESS, W. The bounds of distortion: truth, meaning and efficacy in digital geographic representation. *International Journal of Cartography 3*, 1 (2017), 31–44. doi:10.1080/23729333.2017.1301348.

[13] GUO, Z. Mind the map! the impact of transit maps on path choice in public transit. *Transportation Research Part A: Policy and Practice 45*, 7 (2011), 625–639.

[14] HANSEN, S., RICHTER, K.-F., AND KLIPPEL, A. Landmarks in openls — a data structure for cognitive ergonomic route directions. In *Geographic Information Science* (Berlin, Heidelberg, 2006), M. Raubal, H. J. Miller, A. U. Frank, and M. F. Goodchild, Eds., Springer Berlin Heidelberg, pp. 128–144.

[15] HARVEY, F. Considerations of Graphical Proximity and Geographical Nearness. In *10th International Conference on Geographic Information Science (GIScience 2018)* (Dagstuhl, Germany, 2018), S. Winter, A. Griffin, and M. Sester, Eds., vol. 114 of *Leibniz International Proceedings in Informatics (LIPIcs)*, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, pp. 4:1–4:18. doi:10.4230/LIPIcs.GISCIENCE.2018.4.

[16] HAUNERT, J., AND SERING, L. Drawing road networks with focus regions. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2555–2562.

[17] HONG, S.-H., MERRICK, D., AND DO NASCIMENTO, H. A. Automatic visualisation of metro maps. *Journal of Visual Languages & Computing 17*, 3 (2006), 203–224. doi:10.1016/j.jvlc.2005.09.001.

[18] ISHIKAWA, T., FUJIWARA, H., IMAI, O., AND OKABE, A. Wayfinding with a gps-based mobile navigation system: A comparison with maps and direct experience. *Journal of Environmental Psychology 28*, 1 (2008), 74–82. doi:10.1016/j.jenvp.2007.09.002.

[19] KLIPPEL, A., DEWEY, C., KNAUFF, M., RICHTER, K.-F., MONTELLO, D. R., FREKSA, C., AND LOELIGER, E.-A. Direction concepts in wayfinding assistance systems. In *Workshop on Artificial Intelligence in Mobile Systems (AIMS'04)* (2004).

[20] KLIPPEL, A., RICHTER, K. F., BARKOWSKY, T., AND FREKSA, C. The cognitive reality of schematic maps. In *Map-based mobile services*, L. Meng, T. Reichenbacher, and A. Zipf, Eds. Springer, Berlin, Heidelberg, 2005, pp. 55–71. doi:10.1007/3-540-26982-7-5.

[21] KLIPPEL, A., TAPPE, H., AND HABEL, C. Pictorial representations of routes: Chunking route segments during comprehension. In *Spatial Cognition III. Spatial Cognition 2002* (2003), C. Freksa, W. Brauer, C. Habel, and K. Wender, Eds., Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, pp. 11–33. doi:10.1007/3-540-45004-1_2.

[22] KLIPPEL, A., TAPPE, H., KULIK, L., AND LEE, P. U. Wayfinding choremes: a language for modeling conceptual route knowledge. *Journal of Visual Languages & Computing 16*, 4 (2005), 311–329. doi:10.1016/j.jvlc.2004.11.004.

[23] KOPF, J., AGRAWALA, M., BARGERON, D., SALESIN, D., AND COHEN, M. Automatic generation of destination maps. *ACM Trans. Graph. 29*, 6 (Dec. 2010), 158:1–158:12. doi:10.1145/1882261.1866184.

[24] KRAY, C., AND BLOCHER, A. Modeling the basic meanings of path relations. In *IJCAI International Joint Conference on Artificial Intelligence* (1999), vol. 1, pp. 384–393.

[25] KRÜGER, A., ASLAN, I., AND ZIMMER, H. The effects of mobile pedestrian navigation systems on the concurrent acquisition of route and survey knowledge. In *Mobile Human-Computer Interaction - MobileHCI 2004* (2004), S. Brewster and M. Dunlop, Eds., vol. 3160 of *Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, pp. 446–450. doi:10.1007/978-3-540-28637-0_54.

[26] KRUKAR, J., AND SCHWERING, A. What is orientation? In *Proceedings of the 13th Biannual Conference of the German Cognitive Science Society* (2016), pp. 115–118.

[27] LAN, T., LI, Z., AND TI, P. Integrating general principles into mixed-integer programming to optimize schematic network maps. *International Journal of Geographical Information Science 00*, 00 (2019), 1–29. doi:10.1080/13658816.2019.1620237.

[28] LI, Z., AND DONG, W. A stroke-based method for automated generation of schematic network maps. *International Journal of Geographical Information Science 24*, 11 (2010), 1631–1647. doi:10.1080/13658811003766936.

[29] LI, Z., AND TI, P. Adaptive generation of variable-scale network maps for small displays based on line density distribution. *GeoInformatica 19*, 2 (2015), 277–295. doi:10.1007/s10707-014-0212-8.

[30] LÖWEN, H., KRUKAR, J., AND SCHWERING, A. Spatial learning with orientation maps: The influence of different environmental features on spatial knowledge acquisition. *ISPRS International Journal of Geo-Information 8*, 3 (2019), 149. doi:10.3390/ijgi8030149.

[31] MEILINGER, T., HÖLSCHER, C., BÜCHNER, S. J., AND BRÖSAMLE, M. How much information do you need? Schematic maps in wayfinding and self localisation. *Spatial Cognition V Reasoning, Action, Interaction 4387* (2007), 381–400. doi:10.1007/978-3-540-75666-8_22.

[32] MEULEMANS, W. Discretized approaches to schematization. *CoRR abs/1606.06488* (2016).

[33] MÜNZER, S., LÖRCH, L., AND FRANKENSTEIN, J. Wayfinding and acquisition of spatial knowledge with navigation assistance. *Journal of Experimental Psychology: Applied* (2019). doi:10.1037/xap0000237.

[34] NICKEL, S., AND NÖLLENBURG, M. Towards data-driven multilinear metro maps. In *Diagrammatic Representation and Inference (DIAGRAMS'20)* (2020), A.-V. Pietarinen, P. Chapman, L. Bosveld de Smet, V. Giardino, J. Corter, and S. Linker, Eds., vol. 12169 of *LNAI*, Springer, pp. 153–161. doi:10.1007/978-3-030-54249-8_12.

[35] NÖLLENBURG, M. A survey on automated metro map layout methods. In *Schematic Mapping Workshop 2014* (University of Essex, England, April 2014).

[36] NÖLLENBURG, M., AND WOLFF, A. A mixed-integer program for drawing high-quality metro maps. In *Graph Drawing* (2006), P. Healy and N. S. Nikolov, Eds., Springer Berlin Heidelberg, pp. 321–333. doi:10.1007/11618058_29.

[37] NÖLLENBURG, M., AND WOLFF, A. Drawing and labeling high-quality metro maps by mixed-integer programming. *IEEE Transactions on Visualization and Computer Graphics 17*, 5 (2011), 626–641. doi:10.1109/TVCG.2010.81.

[38] OKE, O., AND SIDDIQUI, S. Efficient automated schematic map drawing using multiobjective mixed integer programming. *Computers and Operations Research 61* (2015), 1–17. doi:10.1016/j.cor.2015.02.010.

[39] OPENSTREETMAP CONTRIBUTORS. Planet dump retrieved from planet.osm.org . www.openstreetmap.org, 2017.

[40] RICHTER, K.-F. A uniform handling of different landmark types in route directions. In *Spatial Information Theory* (Berlin, Heidelberg, 2007), S. Winter, M. Duckham, L. Kulik, and B. Kuipers, Eds., Springer Berlin Heidelberg, pp. 373–389.

[41] RICHTER, K.-F. *Context-specific route directions: generation of cognitively motivated wayfinding instructions*, vol. 314. IOS Press, 2008.

[42] RICHTER, K.-F., AND KLIPPEL, A. A model for context-specific route directions. In *Spatial Cognition IV. Reasoning, Action, Interaction* (Berlin, Heidelberg, 2005), C. Freksa, M. Knauff, B. Krieg-Brückner, B. Nebel, and T. Barkowsky, Eds., Springer Berlin Heidelberg, pp. 58–78.

[43] SCHMID, F. Knowledge-based wayfinding maps for small display cartography. *Journal of Location Based Services 2*, 1 (2008), 57–83. doi:10.1080/17489720802279544.

[44] SCHMID, F., RICHTER, K.-F., AND PETERS, D. Route aware maps: Multigranular wayfinding assistance. *Spatial Cognition & Computation 10*, 2-3 (2010), 184–206. doi:10.1080/13875861003592748.

[45] SCHRIJVER, A. *Theory of linear and integer programming*. John Wiley & Sons, 1998.

[46] SCHWERING, A., KRUKAR, J., LI, R., ANACTA, V. J., AND FUEST, S. Wayfinding Through Orientation. *Spatial Cognition and Computation 17*, 4 (2017), 273–303. doi:10.1080/13875868.2017.1322597.

[47] STOTT, J. M., AND RODGERS, P. Metro map layout using multicriteria optimization. In *Proceedings. Eighth International Conference on Information Visualisation, 2004. IV 2004.* (July 2004), pp. 355–362. doi:10.1109/IV.2004.1320168.

[48] VAN DIJK, T. C., AND HAUNERT, J.-H. Interactive focus maps using least-squares optimization. *International Journal of Geographical Information Science 28*, 10 (2014), 2052–2075. doi:10.1080/13658816.2014.887718.

[49] VAN DIJK, T. C., VAN GOETHEM, A., HAUNERT, J.-H., MEULEMANS, W., AND SPECKMANN, B. Map schematization with circular arcs. In *Geographic Information Science. GIScience 2014* (2014), M. Duckham, E. Pebesma, K. Stewart, and A. Frank, Eds., Lecture Notes in Computer Science, Springer, Cham, pp. 1–17. doi:978-3-319-11593-1_1.

[50] VAN GOETHEM, A., MEULEMANS, W., SPECKMANN, B., AND WOOD, J. Exploring curved schematization of territorial outlines. *IEEE Transactions on Visualization and Computer Graphics 21*, 8 (Aug 2015), 889–902. doi:10.1109/TVCG.2015.2401025.

[51] WANG, Y.-S., AND CHI, M.-T. Focus plus Context Metro Maps. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2528–2535. doi:10.1109/TVCG.2011.205.

[52] WESSEL, G., ZIEMKIEWICZ, C., CHANG, R., AND SAUDA, E. Gps and road map navigation : The case for a spatial framework for semantic information. In *Proceedings of the International Conference on Advanced Visual Interfaces* (2010), ACM, pp. 207–214. doi:10.1145/1842993.1843030.

[53] WIENER, J. M., AND MALLOT, H. A. 'fine-to-coarse' route planning and navigation in regionalized environments. *Spatial Cognition & Computation 3*, 4 (2003), 331–358. doi:10.1207/s15427633scc0304_5.

[54] WU, H.-Y., TAKAHASHI, S., LIN, C.-C., AND YEN, H.-C. Travel-Route-Centered Metro Map Layout and Annotation. *Computer Graphics Forum 31* (2012), 925–934. doi:10.1111/j.1467-8659.2012.03085.x.