International Journal of Geographical Information Science
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tgis20

Car navigation - computing routes that avoid complicated crossings

Jukka M. Krisp[a] & Andreas Keler[a]
[a] Institute of Geography, Applied Geoinformatics, University of Augsburg, Augsburg, Bavaria, Germany
Published online: 18 Jun 2015.

International Journal of Geographical Information Science, 2015
http://dx.doi.org/10.1080/13658816.2015.1053485

Taylor & Francis
Taylor & Francis Group

Car navigation – computing routes that avoid complicated crossings

Jukka M. Krisp* and Andreas Keler

Institute of Geography, Applied Geoinformatics, University of Augsburg, Augsburg, Bavaria, Germany

(*Received 24 September 2014; final version received 4 May 2015*)

Personalized navigation and way-finding are prominent research areas of location-based service (LBSs). This includes innovative concepts for car navigation. Within this paper, we investigate the idea of providing drivers a routing suggestion which avoids 'complicated crossings' in urban areas. Inexperienced drivers include persons who have a driver's license but, for whatever reason, feel uncomfortable to drive in a city environment. Situations where the inexperienced driver has to depend on a navigation device and reach a destination in an unfamiliar territory may be difficult. Preferences of inexperienced drivers are investigated. 'Fears' include driving into 'complicated crossings'. Therefore, the definition and spatial characteristics of 'complicated crossings' are investigated. We use OpenStreetMap as a road dataset for the routing network. Based on the topological characteristics of the dataset, measured by the number of nodes, we identify crossings that are 'complicated'. The user can choose to compute an alternative route that avoids these complicated crossings. This methodology is one step in building a full 'inexperienced drivers' routing system, which includes additional preferences from the user group, for example, as avoiding left turns where no traffic light is present.

Keywords: location-based services (LBSs); routing; car navigation; way finding; vehicle navigation

Introduction and background

Currently common, in-car navigation systems allow the user to select a destination and provide detailed instructions on where to turn. These systems are developed by a number of commercial providers (e.g. Google, Nokia, Garmin, etc.) and are well established. Generally, these routing systems let the user choose several options on how to calculate the route, e.g. the shortest or the fastest, avoid toll roads or avoid highways. These options are based on the attributes of the underlying geographical dataset. Additionally, current navigation systems include 'near-real-time' information to be considered in the routing options, for example, traffic information or roadblocks due to an accident or construction site. This information is used to dynamically recalculate the route and provide the user with a 'better', generally meaning a faster route alternative. Recent attempts in the field of electric-powered cars offer the route using as little power as possible to extend the range of the vehicle.

Automated positioning functionalities in car navigation systems and smartphones enable the user to locate themselves constantly. Once location information is derived, it needs to be processed in several ways, including its transformation into the format of

*Corresponding author. Email: jukka.krisp@geo.uni-augsburg.de

another spatial reference system, its correlation with other location information or geographic content, the generation of maps, or the calculation of navigation instructions (Küpper 2005). Therefore, navigation and way-finding are prominent research areas of location-based services (LBSs). Krisp and Meng (2013) point out that LBSs have been undergoing a breathtaking development during the recent decade. A large number of theoretical and empirical findings of LBSs have already been accumulated in books on 'Location-Based Services and TeleCartography' (Gartner *et al*. 2007), 'Location-Based Services and TeleCartography II' (Gartner and Rehrl 2009), and 'Progress in Location-Based Services' (Gartner and Ortag 2011, Krisp 2013). LBS are investigated from different perspectives that include mobile positioning and tracking technologies, data capturing and computing devices, integrated software engineering, user studies for various applications.

Progress in LBSs includes innovative concepts on car routing and navigation. Early research by Mark *et al*. (1987) states that

> the development of search algorithms to find routes meeting different travel objectives is worth pursuing. Minimizing distance travelled is one objective. Others include the ease of describing or following a route, maximizing aesthetic quality of a journey (perhaps subject to a time constraint) and minimizing travel time. […]
>
> Algorithms permitting explicit trade-offs among these (or other) objectives would greatly enhance the utility of Vehicle Navigation Appliance (VNA), a term introduced by Vehicle Navigation Appliances (1985). More importantly, developing these algorithms would increase our understanding of how people think about and plan travel.

Shirabe (2005) investigated the theoretical basis for network data structures and Curtin (2007) reviewed several major types of network data structures as they have been implemented in several GI-systems.

With an increase of available dynamic data streams (weather, traffic, etc.) and growing computing functionalities on mobile devices, the parameters for routing functionalities need to be extended. Therefore, we aim on more personalized routing included in car navigation systems that have functionalities depending on the users' interests and abilities. As Huang *et al*. (2014) state, 'in order to offer services that more adequately support and suit users preferences, their affective responses to environments should be considered when computing a route on road networks'. A number of different parameters that could be considered may provide the user with the shortest, fastest, safest, most beautiful, least fuel/energy consumption, male/female (Häusler *et al*. 2010), easiest (to drive) (Krisp *et al*. 2014) or 'most difficult' (to drive) route. Additionally, the 'most difficult to drive route' might be useful for training purposes, for example, in driving schools.

Duckham and Kulik (2003) investigated 'the simplest route' in terms of how easy it is to explain, understand, memorize, or execute the navigation instructions for the route. Simple routes are based on cognitive complexity, for example how difficult is it to take the right decision at a particular crossing. This may imply an algorithm with the simplest instructions for routing that allows a faster path description (Mark 1986, Richter and Duckham 2008). Path description is, in most cases, connected with path selection and the route preference of humans (Golledge 1995).

Most automated navigation systems rely on computing the solution for the shortest path problem (Richter *et al*. 2004), and not the problem of finding the 'simplest' path. Technically, this way of computing a personalized route can assist users to drive a perhaps more 'reasonable' or more 'natural' route. Other approaches follow the minimize turns based on the connectivity of roads (Jiang and Liu 2011, Winter 2002) or make use of avoiding 'confusing' intersections (Haque *et al*. 2007). Ludwig *et al*. (2009) propose personalized routes for pedestrians using the public transport and the connection with event recommendation.

The indication of measuring connectivity relates to 'space syntax' (Hillier 1999, Ratti 2005). Its main idea consists of a computational language for describing spatial patterns in urban environments using relationships between spaces or their interactions with the society (Hillier and Hanson 1984, Jiang and Claramunt 2002). These spaces are represented by straight lines (axial lines) and topological graphs showing their intersections (Jiang and Liu 2010) with the aim of creating distance indices for accessibility approximation (Batty and Rana 2002, 2004).

Patel *et al*. (2006) use specific landmarks within an area for defining personalized routes and Duckham *et al*. (2010) investigate algorithms for generating routing instructions that include references to landmarks, which can be referred to as cues (Caduff and Timpf 2005). The use of personal landmarks enables routing services to generate personalized driving instructions. The main advantage of these personalized propositions is the reduction of turning instructions (steps) within one route, which makes the personalized route simpler. Rogers *et al*. (1997) propose personalization of routing as an application from the field of machine learning, which is built on a knowledge base from training data. They suggest a system that dynamically learns a driver's familiar route and incorporates this 'knowledge' into route planning, route description, and destination prediction. This would require a reasonably comprehensive base of routes from a driver. The data are meant to be the driver's familiar routes, which are used for the derivation of personalized route planning. These personalized routes might also include preferences for a route that is 'easy to drive' and avoids complicated crossings.

Problem statement and aim – what is an 'easy to drive' route?

Car navigation systems provide drivers with very limited options to choose from when calculating the route. This route should meet their specific preferences. In some cases, the fastest or shortest route may not be the route of choice. Wu *et al*. (2013) investigated the unfamiliarity of drivers with the road environment, including unfamiliar street signs and map language. Situations where the inexperienced driver has to depend on the navigation device and reach a destination in unfamiliar territory may be uncomfortable.

What are the traffic situations that could be avoided for inexperienced drivers and/or driving beginners? In other words, what is an 'easy to drive' route? To determine the uncomfortable situations, a survey had been conducted (Bärschmann 2011). Twenty-one driving situations have been described in short statements. Participants of the survey evaluated their agreement with these statements on a scale from 1 (I agree) to 6 (I do not agree). Eighty questionnaires had been distributed among potential driving beginners at selected driving schools in the Munich city area. Forty have been filled and returned for this survey. Table 1 summarizes the significant findings of this short survey, which include the statements that are ranked as most problematic when learning to drive.

From this survey, even though it can only give indications, we can derive preferences of inexperienced drivers. Table 1 includes statement 'Complicated crossings are no problem for me'. A majority of survey participants did not agree to this statement. Therefore, 'fears' include driving into 'complicated crossings' among other preferences, like avoiding left turns without a traffic light.

Table 1. Summary of a survey conducted regarding fears of driving beginners, investigated by Bärschmann (2011).

	Indication of answers						Sum	Avg.	Dev.
Situation (description in short)	1 (I agree)	2	3	4	5	6 (I do not agree)	∑	Ø	σ
Statement – 'Driving on the highway is no problem for me'	12	8	10	4	1	0	35	2,3	1,13
Statement – 'Left turns without a traffic light is not problem for me'	4	7	8	6	6	6	37	3,6	1,6
Statement – 'Complicated crossings are no problem for me'	**2**	**6**	**8**	**9**	**11**	**2**	**38**	**3,7**	**1,32**
…									

Note: Bold indicates the statement used in the further investigations.

Our aim is to provide routing options that avoid complicated crossings. Therefore, we need to automatically identify and extract complicated crossings from a geographical dataset and mark them as difficulties for inexperienced drivers. These crossings are used as obstacles that can be avoided when calculating a route.

Conceptually, a crossing is complicated for a driver, because many (multilane) streets meet, there might be a tram going through that crossing, bikes riding on the same street, and pedestrians may cross the street. Many of these features (tram, bike lanes, and roads) have nodes on a particular crossing. Therefore, from a technical point of view, we define a crossing therefore in our case we define a 'complicated crossing' as a crossing with a certain number of nodes that exceeds a defined threshold.

Road data – OpenStreetMap (OSM) data as a basis for routing

We use OpenStreetMap (OSM) data as a base road network for routing. 'OSM is a knowledge collective that creates Open Geodata as its main objective' (Haklay and Weber 2008). As OSM encourages a broad community (in the year 2014, there were about 25,000 active contributors a month out of about 1.7 million registered users[1]) to contribute to its development, wide regions are covered. OSM has been introduced to different research areas and developments like routing. OSM data need to be prepared, as its original design is not routable (Neis *et al*. 2012). A critical point is the completeness of downloaded OSM data, as certain providers reduce the amount of data or interpret content differently. Consequently, only unlimited access to unmodified OSM data, including all relevant attribute information such as directions, turns, and barriers, is a basis for proper routing.

A routable network needs topologically correct data when converted into edges and nodes. Neis *et al*. (2012) examined the topological correctness of OSM data and states 'such errors […] have decreased over the years and […] remain high only for routes of cyclists or pedestrians'. Figure 1 shows the study area, the city of Munich, and the selected road extracts (major roads, local roads) and other network lines (bike roads, trams) from the OSM.

To determine a complicated crossing, we have included a number of selected road types to include nodes of the vehicle infrastructure, plus cycle, tram, or pedestrian ways.

Figure 1. Study area Munich showing extracted nodes of the road network, bike roads, trams, and walkways (source OpenStreetMap).

From a computational standpoint, roads should topologically connect to represent a crossing. In our case, we are searching for a ‘complicated crossing’, and our assumption is that in a ‘complicated crossing’, large number of nodes are in close proximity to each other. That includes, for example, bike lanes and trams that intersect with a road network, but are not part of the road network used for routing. In this method, a simple cluster of nodes within a specific crossing gives an indication of its ‘complexity’ and based on that it can be defined as a ‘complicated crossing’. In other words, complicated crossings can appear within the locations where different road types are connected to each other.

Computing a complicated crossing using node density

Nodes are the basis to determine a complicated crossing. We suggest that a high number of nodes form hotspots within the investigation area. A high density of nodes appears in intersecting roads, normally in the locations of crossing. Particular crossings will include a higher amount of extracted points.

The nodes of selected road extracts (or road type values) are used to compute a node density layer. Based on these densities, we generate polygons which are used as obstacles in the further route calculations. The workflow is shown in Figure 2.

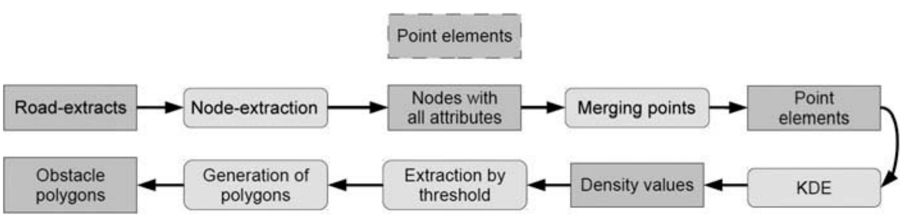


Figure 2. Workflow showing the process from extracting OSM nodes and generating obstacle polygons.

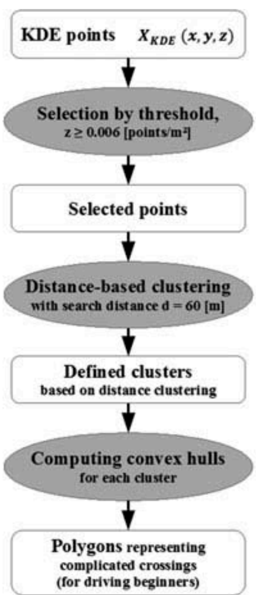
To detect hotspots of node density, we apply a kernel density estimation method. Kernel density estimation (Silverman 1986, Bailey and Gatrell 1995) is an efficient way to detect hotspots or clusters within point data. The choice of the search radius (or bandwidth) affects the kernel density estimation strongly. Krisp *et al*. (2009) suggest a visual determination of the kernel bandwidth. When the kernel density estimation is carried out with a small bandwidth, the estimation interacts with individual events. For this reason we experiment with different bandwidths to visually find an appropriate setting.

The search radius for this kernel density estimation is set to 60 m, which is the estimated average diameter of a roundabout in Munich, and there is a notable amount of roundabouts in included in the road network. The kernel density estimation for the data of the investigation area of Munich was selected with cell size of 5 m. This means that the calculated density has an accuracy of about 5 m. Areas of high density are the basis for deriving obstacle polygons.

Identified complicated crossings within the study area

The kernel density estimation calculates values for each cell of the density layer. We selected a threshold of 0.006 points per m2 after investigating different point density thresholds. We checked the complicated crossings on site or via Google StreetView. The threshold reflects our estimated boundary between a 'still simple' and an 'already complicated' crossing. The following commented workflow and pseudo code, shown in Figure 3, documents the process of a selection by threshold and obstacle generation.

With the extraction of polygons using a threshold for the estimated density values, the number of classified crossings within the investigation area of Munich can be estimated by 133 generated obstacles. The obstacle polygons in the Munich city center area are plotted in Figure 4.




```
PSEUDOCODE SELECTION BY THRESHOLD & OBSTACLE GENERATION

Input:      Density layer with values z from Kernel density functions f_h(x)
            represented as Points X_KDE(x,y,z) with z as the calculated value
            for the Kernel density function.

Output:     A set of obstacle polygons P (Convex Hulls P)

Function:

Select * from X_KDE(x,y,z)
      where z ≥ 0.006         // unit: points per square meter, [points/ m²]
      X_KDE,thresh   =   S    // Selected Points

d = 60                         // distance d; unit: meter, [m²]
T = Aggregate (S, d)           // Clustering based on distance d
k = 0

for each R in T do             // T = Groups of selected points

pointOnHull = leftmost point in R     // creation of obstacle polygons P
i = 0
 repeat
   C[i] = pointOnHull                 // C contains set of boundary points
   endpoint = R[0]  // initial endpoint for candidate edge on the hull
   for j from 1 to |R|
         if endpoint = pointOnHull
               or R[j] is on left of line from C[i] to endpoint then
            endpoint = R[j]   // found greater left turn, update endpoint
         end if
         i = i+1
         pointOnHull = endpoint
   end for
 until endpoint = C[0]         // reached first hull point
P[k] = createPolygon(C)        // connect points of C by lines
k = k+1                        // update polygon count for next iteration
end for
```

Figure 3. Pseudo code – 1 workflow and pseudo code selection by threshold and obstacle generation.

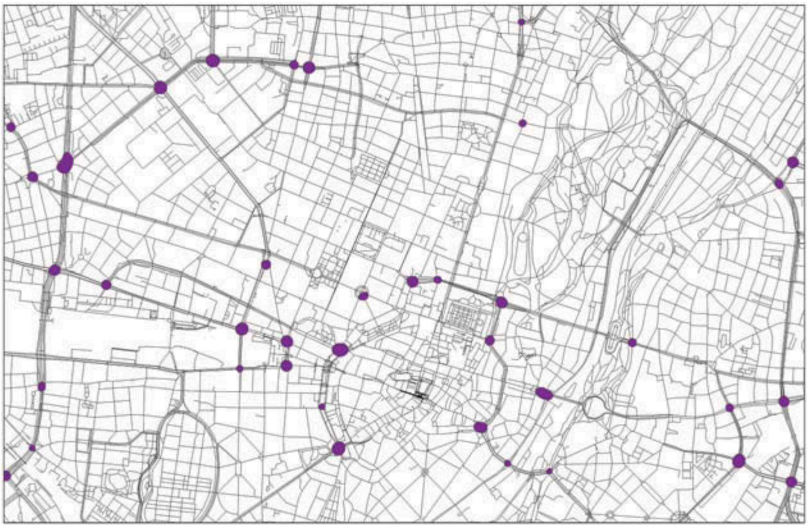


Figure 4. Overview of the obstacle polygons in the Munich city center area.

To what extent are these identified polygons (based on the node density) 'complicated crossings'? Obstacle polygons can be examined in more detail by site visits or by linked views, for example, using Google StreetView. Three obstacle polygons are selected based on the amount of nodes and are examined further. Figure 5 shows a view of three selected

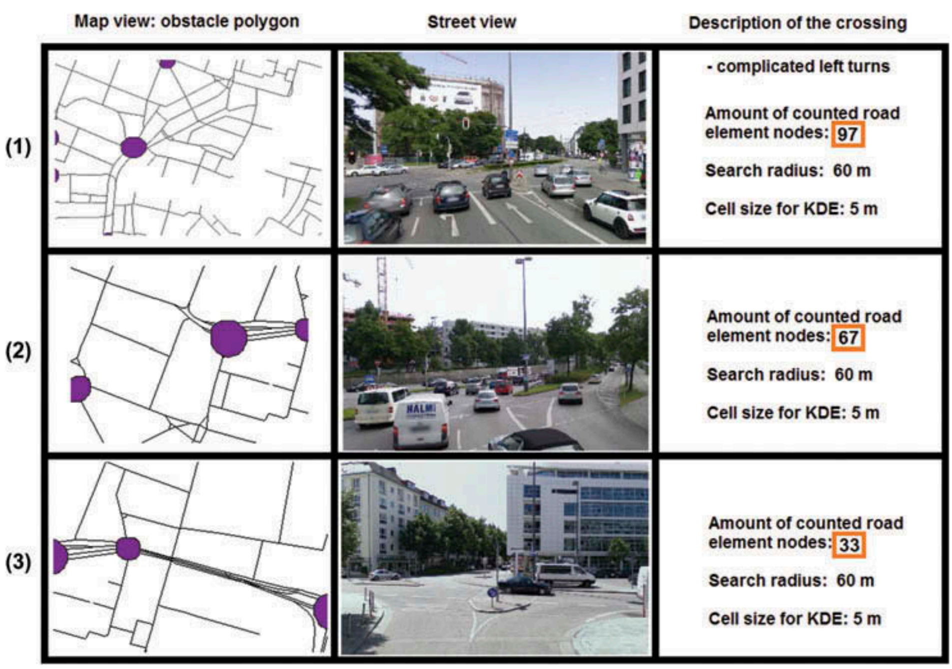


Figure 5. Selected obstacle polygons on the road network (1. to 3.), StreetView picture of the obstacle location (StreetView) and the list with the amount of nodes counted in the crossing area (description of the crossing).

obstacle polygons (complicated crossings) in the road network. It is linked in the table with a photograph from the driver's perspective acquired from Google StreetView.[2]

The StreetView picture shows the traffic situation from the drivers' perspective and the description of the crossing with properties of the generated obstacle polygon. The last column describes the specification of the amount of extracted road network nodes and the used density estimation for creating the certain obstacle polygon. As an example, the most complicated crossing within the study area is the 'Lenbachplatz' in Munich with 97 road nodes. As shown in the StreetView pictures, as well as on-site investigations, this crossing is 'complicated to drive'.

Routing algorithm – how can we compute a route avoiding complicated crossings within the road network?

As we have identified a number of 'complicated crossing' obstacles, the aim is to avoid these obstacles in the suggested routing functionality. Routing can be described as a search for the best path in a network. Different types of networks can be defined with regard to their purpose, such as street networks serving as mode of transport for motor vehicles, pipeline networks transporting fluent materials, or cable networks carrying digital information or electrical current. Finding the best path on a network generally is managed by defining the path with lowest costs, depending on regarded criteria. Several criteria influence path search. The most commonly used are distance and travel time measures, but there are also combined measures such as situation and capacity of a path (Halaoui 2010).

Most commonly known and used for routing is the Dijkstra's algorithm (Dijkstra 1959). This algorithm includes the search for 'the shortest path between two nodes in graph […], each having a non-negative weight' (De Berg *et al*. 2008). It 'maintains an array of tentative distances $D[u] \geq d(s, u)$ for each node' (Delling 2009), and by visiting the neighboring nodes via the edges connecting the initial node, their distance is saved. Based on Dijkstra's algorithm and its general principals of computing positive weighted, shortest graphs only, A* algorithm takes a general direction of the target node via Euclidean distance into consideration. The edge weight is modified reducing the distance to the target, still searching for the shortest path (Delling 2009), though limiting data and time selecting the best paths with regard to certain criteria. For routing on our network, Dijkstra's algorithm serves our purpose, as we only use positive weighted graphs.

Results – alternative routing through the study area

The routing function is applied to a test route with a starting point at the Theresienwiese and an end point at the Ludwig-Maximilians-Universität (LMU), both located within the Munich city area. The selected routes in Figures 6 and 7 connect these well-known places. Starting from Theresienwiese, which is the location where the world-famous Oktoberfest takes place each year, the routing is provided to the university LMU, which is one of the universities in Munich.

Figure 6 shows the 'classic' shortest path or 'standard route', which is calculated by the Dijkstra's algorithm. We regard this as the 'standard route', which online or onboard routing system will suggest to the user. The shortest possible route is shown here with the use of the inner ring road. This standard route crosses three complicated crossings, including the 'Lenbachplatz'.

Figure 6. Example for a standard route using the Dijkstra functions between Theresienwiese (1) and LMU (2).

Figure 7. Example for a route avoiding complicated crossings set as obstacles between Theresienwiese (1) and LMU (2).

Figure 7 shows a route in which obstacle polygons are avoided in the route calculation. In this route, we use the complicated crossings as fixed obstacle polygons. These obstacle polygons cannot be crossed. An alternative would be to assign weights to the obstacle (e.g. based on the number of nodes) and allow routes through these crossings, based on these weights. Standard routing algorithms search for the shortest (or the fastest) route. In comparison, this route is longer and slower than the standard route. Therefore, apparently current routing systems will not compute this 'route avoiding complicated crossings' for these particular start and end points. The selected differences are summarized in Table 2.

We have computed, respectively, 10 'random' routes crossing the city of Munich. Each route has individual start and end points. We computed the shortest path for the start and end points and, using the same start and end points, we computed the route 'avoiding

Table 2. Comparison of parameters for a ‘standard route’ (shortest path) and a ‘route avoiding complicated crossings’ shown in Figure 6.

Parameters for routes	Standard route	Route avoiding complicated crossings as obstacles
Length	5.14 [km]	6.19 [km]
Time to drive	11 [minutes]	14 [minutes]
Number of turns	7	10
Number of right turns	4	6
Number of left turns	3	4
Number of crossings	19	21
Number of avoided ‘complicated crossings’	0	3

complicated crossings’. Comparison these routes shows that, on average, the ‘avoiding complicated crossings’ route is 16.86% longer than the shortest route. They have, on average, 11 more turns – four more left turns and seven more right turns.

Discussion and conclusions

Within this research, we show a possibility of avoiding complicated crossings, defined as obstacles, which are based on the number of nodes within one crossing. Results show that it is possible to determine and avoid these crossings within a route computation. The methodology has shortcomings as it involves ‘false positives’, for example, intersections that kernel density estimation detects as complex, but they really are not complex (or the clusters are not even intersections). As our method is based on node density to detect complex crossings, each crossing has an individual amount of nodes (e.g. 97 nodes in case of the Lenbachplatz); therefore, we can set a threshold. In the case of computing an ‘easy to drive’ route, this point is less crucial, as, for example, the driving beginners would like to avoid complicated crossings. If a crossing is less (or not at all) complicated, we still may avoid it by this automated classification method. From an application point of view, it does not do any harm and still satisfies the user.

Currently, we have investigated selected start and end points in the area of Munich. To have a proof of concept, a comprehensive user test needs to be conducted. Is the route avoiding complicated crossings in every case easier to drive than the standard route? People who feel uncomfortable to drive are also not very inclined to turn into side streets, just with the purpose of avoiding complicated crossings. This might confuse them even more. Therefore, more personalized preferences are needed of a comprehensive system, which might suggest an ‘easy to drive’ route. Difficulties occur as the definitions for a ‘personalized route’ are not clear. For some users, the shortest path may be the optimal most ‘natural’ route, other users may prefer a different route. Technically, this way of computing an easy route can assist us to find a different, perhaps more ‘reasonable’ or more ‘natural’ route around defined obstacles in a test study area.

Future work

This is the first step in the development of a more comprehensive system that suggests users an 'easy to drive' route. Other personalized parameters need to be considered before such a system will give improved routing results. Additionally, the routing calculation needs to consider additional features (crossing tram lane, bike lane) which are stored in distributed databases. Dynamic data, such as weather and traffic density, need to be considered. Wang and Zlatanova (2012) suggest that an extension for routing with obstacle polygons might be a time-dependent movement of the obstacles, which may allow different routing solutions for the same start and end points but different time of day.

Theoretically, the concept of a 'complicated crossing' stretches on a fuzzy continuum ranging from an 'easy to drive' route to a 'difficult to drive route'. The concept of an 'easy to drive' route can be extended, for example, by including 'avoiding left turns without a traffic light'.

One additional field of application can be the use of a personalized routing system in driving schools. Since certain driving instructors are familiar with the local transport infrastructure, they can evaluate certain obstacles and change their calculated weighing to more realistic values. The routing database needs to include a reasonably complete number of obstacles, which makes it more complex to provide a realistic and correct route, including dynamic objects. Additionally, Ward and Hirst (1997) state that sometimes drivers can be distracted by a secondary task (e.g. interacting with in-vehicle information technology system such as GPS, cell phone).

A prominent research area for this kind of personalized routing services is privacy protection. That may include anonymous locations, routes, and privacy policies. User behaviors' for reducing group pressure for revealing the routing preference of the device need to be investigated.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Notes

1. OpenStreetMap Wiki Contributors (2014): OpenStreetMap Wiki, Stats. Online, http://wiki.openstreetmap.org/wiki/Stats [Accessed 18 Jun 2014].
2. https://www.google.com/maps/views/streetview?gl=de&hl=de

References

Bailey, T.C. and Gatrell, A.C., 1995. *Interactive spatial data analysis*. Reading: Addison-Wesley.

Bärschmann, A., 2011. *Anforderungen an ein Fahranfänger-Routingsystem*. Thesis. Technical University Munich.

Batty, M. and Rana, S., 2002. Reformulating space syntax: the automatic definition and generation of axial lines and axial maps [online].

Batty, M. and Rana, S., 2004. The automatic definition and generation of axial lines and axial maps. *Environment and Planning B: Planning and Design*, 31, 615–640. doi:10.1068/b2985

Caduff, D. and Timpf, S., 2005. The landmark spider: representing landmark knowledge for wayfinding tasks. *In*: T. Barkowsky, *et al*., eds. *Reasoning with mental and external diagrams: computational modeling and spatial assistance*. Menlo Park, CA: AAAI Spring Symposium, 30–35.

Curtin, K.M., 2007. Network analysis in geographic information science: review, assessment, and projections. *Cartography and Geographic Information Science*, 34 (2), 103–111. doi:10.1559/152304007781002163
De Berg, M., *et al*., 2008. *Computational geometry: algorithms and applications*. Berlin: Springer, 323–333.
Delling, D., 2009. Engineering route planning algorithms. *In*: J. Lerner *et al*., eds. *Algorithmics of large and complex networks*. Berlin: Springer, 117–139.
Dijkstra, E.W., 1959. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1 (1), 269–271. doi:10.1007/BF01386390
Duckham, M. and Kulik, L., 2003. “Simplest” paths: automated route selection for navigation. *In*: W. Kuhn, M.F. Worboys, and S. Timpf, eds. *Spatial information theory. Foundations of geographic information science, lecture notes in computer science*. Berlin: Springer, 182–199.
Duckham, M., Winter, S., and Robinson, M., 2010. Including landmarks in routing instructions. *Journal of Location Based Services*, 4 (1), 28–52. doi:10.1080/17489721003785602
Gartner, G., Cartwright, W., and Peterson, P., 2007. *Location based services and telecartography*. Berlin: Springer.
Gartner, G. and Ortag, F., 2011. *Advances in location-based services*. Berlin: Springer.
Gartner, G. and Rehrl, K., 2009. *Location based services and telecartography II*. Berlin: Springer.
Golledge, R.G., 1995. Path selection and route preference in human navigation: a progress report. *In*: A.U. Frank and W. Kuhn, eds. *Spatial information theory*. Berlin: Springer, 207–222.
Haklay, M. and Weber, P., 2008. OpenStreetMap: user-generated street maps. *IEEE Pervasive Computing*, 7 (4), 12–18. doi:10.1109/MPRV.2008.80
Halaoui, H.F., 2010. Intelligent traffic system: road networks with time-weighted graphs. *International Journal for Infonomics*, 3 (4), 350–359.
Haque, S., Kulik, L., and Klippel, A., 2007. Algorithms for reliable navigation and wayfinding. *In*: T. Barkowsky, *et al*., eds. *Spatial cognition V – reasoning, action, interaction, LNCS4387*. Berlin: Springer, 308–326.
Häusler, E., *et al*., 2010. The FEMroute project – a gender-sensitive approach to route planning systems for pedestrians. *In*: G. Gartner and Y. Li, eds. *Proceedings of the 7th international symposium on location based services & telecartography*, 20–22 September, 124–128.
Hillier, B., 1999. *Space is the machine: a configurational theory of architecture*. Cambridge: Cambridge University Press.
Hillier, B. and Hanson, J., 1984. *The social logic of space*. Cambridge: Cambridge University Press.
Huang, H., *et al*., 2014. AffectRoute – considering people’s affective responses to environments for enhancing route-planning services. *International Journal of Geographical Information Science*, 28, 2456–2473. doi:10.1080/13658816.2014.931585
Jiang, B. and Claramunt, C., 2002. Integration of space syntax into GIS: new perspectives for urban morphology. *Transactions in GIS*, 6 (3), 295–309. doi:10.1111/1467-9671.00112
Jiang, B. and Liu, X., 2010. Automatic generation of the axial lines of Urban environments to capture what we perceive. *International Journal of Geographical Information Science*, 24 (4), 545–558. doi:10.1080/13658810902950351
Jiang, B. and Liu, X., 2011. Computing the fewest-turn map directions based on the connectivity of natural roads. *International Journal of Geographical Information Science*, 25 (7), 1069–1082. doi:10.1080/13658816.2010.510799
Krisp, J.M., *et al*., 2009. Visual bandwidth selection for Kernel density maps. *PFG – Photogrammetrie Fernerkundung Geoinformation*, 5, 441–450.
Krisp, J.M., 2013. *Progress in location based services (LBS)*. Berlin: Springer.
Krisp, J.M., Keler, A., and Karrais, N., 2014. Personalized routing for car navigation systems. *In*: G. Gartner and H. Huang, eds. *Proceedings of the 11th international symposium on location-based services*, *LBS2014*, 26–28 November, Vienna.
Krisp, J.M. and Meng, L., 2013. Going local – evolution of location based services. *In*: J.M. Krisp, ed. *Progress in location based services*. Heidelberg: Springer, 11–15.
Küpper, A., 2005. *Location-based services fundamentals and operation*. Chichester: Wiley.
Ludwig, B., Zenker, B., and Schrader, J., 2009. Recommendation of personalized routes with public transport connections. *In*: D. Tavangarian, *et al*., eds. *Intelligent interactive assistance and mobile multimedia computing; communications in computer and information science*. Berlin: Springer, 97–107.

Mark, D., 1986. Automated route selection for navigation. *IEEE Aerospace and Electronic Systems Magazine*, 1, 2–5. doi:10.1109/MAES.1986.5005198

Mark, D.M., Gould, M.D., and McGranaghan, M., 1987. Computerized navigation assistance for drivers. *The Professional Geographer*, 39 (2), 215–220. doi:10.1111/j.0033-0124.1987.00215.x

Neis, P., Zielstra, D., and Zipf, A., 2012. The street network evolution of crowdsourced maps: OpenStreetMap in Germany 2007–2011. *Future Internet*, 4 (4), 1–21. doi:10.3390/fi4010001

Patel, K., *et al*., 2006. Personalizing routes. *In*: *Proceedings of the 19th annual ACM symposium on user interface software and technology (Uist'06*). New York, NY: Association for Computing Machinery (ACM), 187–190.

Ratti, C., 2005. Suggestions for developments in space syntax. *In*: *Proceedings of the 5th space syntax symposium*, 13–17 June, Delft. Amsterdam: Techne Press.

Richter, K.F. and Duckham, M., 2008. Simplest instructions: finding easy-to-describe routes for navigation. *In*: T.J. Cova, *et al*., eds. *Lecture notes in computer science – geographic information science*. Heidelberg: Springer, 274–289.

Richter, K.F., Klippel, A., and Freksa, C., 2004. Shortest, fastest – but what next? A different approach to route directions. *In*: M. Raubal, A. Sliwinski, and W. Kuhn, eds. *Geoinformation und Mobilität – von der Forschung zur praktischen Anwendung*. Münster: IfGI Prints, 205–217.

Rogers, S., *et al*., 1997. Personalization of the automotive information environment. *In*: *Proceedings of the workshop on machine learning in the real world; methodological aspects and implications*, 12 July, Nashville, TN, 28–33.

Shirabe, T., 2005. Shortest path search from a physical perspective. *In*: A.G. Cohn and D.M. Mark, eds. *Spatial information theory: cognitive and computational foundations*. Berlin: Springer, 83–95.

Silverman, B.W., 1986. *Density estimation for statistics and data analysis*. London: Chapman and Hall.

Cooke, D.F., Vehicle Navigation Appliances, ed., 1985. *Proceedings Auto-Carto7*. Falls Church, VA: American Society of Photogrammetry and American Congress on Surveying and Mapping, 108–115.

Wang, Z. and Zlatanova, S., 2012. An A*-based approach for navigation among moving obstacles. *In*: S. Zlatanova, *et al*., eds. *Intelligent systems for crisis management; geo-information for disaster management (Gi4dm)*. Berlin: Springer.

Ward, N.J. and Hirst, S., 1997. In-vehicle intelligent information technologies as safety benefit systems: consideration of philosophy and function. *Behaviour & Information Technology*, 16 (2), 88–97. doi:10.1080/014492997119932

Winter, S., 2002. Modeling costs of turns in route planning. *Geoinformatica*, 6 (4), 345–361. doi:10.1023/A:1020853410145

Wu, C., *et al*., 2013. Navigating a car in an unfamiliar country using an internet map: effects of street language formats, map orientation consistency, and gender on driver performance, workload and multitasking strategy. *Behaviour & Information Technology*, 32 (5), 425–437. doi:10.1080/0144929X.2011.566941