TRANSPORT PROBLEMS 2022 Volume 17 Issue 1
PROBLEMY TRANSPORTU DOI: 10.20858/tp.2022.17.1.02

Keywords: routing; cost distance analysis; remote sensing

Armin HAHN[1]*, Wiard FRÜHLING[2], Jan SCHLÜTER[3]

DETERMINATION OF OPTIMISED PICK-UP AND DROP-OFF LOCATIONS IN TRANSPORT ROUTING – A COST DISTANCE APPROACH

Summary. With the emergence of dynamic passenger transport systems, such as demand-responsive transport (DRT) and ride-sharing without predetermined stop locations as used for static bus routes, accurate routing for these flexible door-to-door transport services is needed. Routing between two addresses requires the assignment of addresses to suitable, so-called snapping points as reference points on the road network. Therefore, many conventional routing machines use perpendicular distance to identify the nearest point on the road network. However, this technique tends to produce inaccurate results if the access to a building is not reachable from the road segment with the shortest perpendicular distance. We provide a novel approach to identify the access to buildings (paths) based on remote sensing data to obtain more reasonable stop locations for passenger transport. Multispectral images, OpenStreetMap data, and light detection and ranging (LiDAR) data were used to perform a cost distance analysis based on vegetation cover, building footprints, and the slope of the terrain to identify such optimised stop locations. We assumed that the access to buildings on the shortest route to the building's entrance consists of little vegetation cover and minimal slope of the terrain; furthermore, the calculated path should not cross building footprints. Thus, snapping points on the road network can be determined based on the most likely path between a building and the road network. We validated our results based on a predetermined ideal snapping area considering different weightings for the parameters slope, vegetation, and building footprints. The results were compared with a conventional routing machine that uses perpendicular distance. This routing machine shows a validation rate of 81.4%, whereas the validation rate of our presented approach is as high as 90.3%. This new approach provides increased accuracy and better comfort for flexible passenger transport systems.

1. INTRODUCTION

Map matching is a technique that relates a geographical point or a sequence of points to a logical real-world model, such as a road network [1]. A typical application of offline map matching is the reconstruction of the most likely real driven path of vehicles from sequences of global positioning system (GPS) points. This technique is used when GPS points are not directly located on the road network due to common inaccuracies. There are many publications on real-time map matching [1 - 6],

[1] Max-Planck-Institute for Dynamics and Self-Organization; Am Fassberg 17, 37077 Göttingen, Germany; email: armin.hahn@ds.mpg.de; orcid.org/0000-0002-6255-0361
[2] Max-Planck-Institute for Dynamics and Self-Organization; Am Fassberg 17, 37077 Göttingen, Germany; email: wiard.fruehling@ds.mpg.de; orcid.org/0000-0002-2864-1809
[3] Max-Planck-Institute for Dynamics and Self-Organization; Am Fassberg 17, 37077 Göttingen, Germany; email: jan.schlueter@ds.mpg.de; orcid.org/0000-0002-3363-8663
* Corresponding author. E-mail: armin.hahn@ds.mpg.de

but few studies have investigated offline map matching [7]. In this work, we focus on so-called road snapping, which can be seen as a unique form of offline map matching.

The basic ideas underlying the already known offline map matching and the special road snapping technique that we focus on in this study are similar; however, the aim and, hence, the methods of road snapping are different. In road snapping, a single coordinate, usually an address, is related to a point on a road network (the snapping point) to identify the starting point of the route for the given coordinate or address. Therefore, road snapping by conventional routing machines is based on the perpendicular distance. However, some routing machines also use the matching of the name of the address and the road name, which is not considered in typical offline map matching.

The literature on road snapping is scarce [7] for a number of reasons. In the past, offline and, especially, real-time map matching was generally more relevant for enhancing the routing and post-processing of GPS tracks, thus improving map matching. The accuracy of snapping points for pick-up and drop-off locations was considered less important in the past by transportation network companies since door-to-door transport services were mostly offered by taxi companies, and taxi drivers do not rely solely on routing machines due to their local knowledge. With the emergence of more door-to-door transport services besides taxis, such as Uber and flexible DRT systems, the accuracy of snapping points has become more relevant as drivers of such systems often have less local knowledge than taxi drivers and, therefore, often use routing machines. Furthermore, accurate snapping points can play an important role in autonomous driving in the future.

Road snapping is offered, for example, by Google in the Roads application programming interface (API) [8] or by the Nearest API from an Open Source Routing Machine (OSRM) [9]. The common technique for road snapping is based on perpendicular distance, according to the formula in Equation 1:

$$d = \frac{|Ax+By+C|}{\sqrt{(A^2+B^2)}} \quad , \tag{1}$$

where d is the distance from a point defined by (x, y) to a line defined by $Ax + By + C = 0$. However, road snapping from a geographical point to the road network based on perpendicular distance is not sufficient in some cases. One problem with this technique is that, in the real world, the destination building might not be connected to the road segment with the shortest distance but to another road segment that is further away. This situation occurs, for instance, when a building's driveway is not part of the public road network (e.g., private properties) or when this data is missing and cannot be used to determine the snapping point. However, we focus on enhancing suboptimal pick-up and drop-off locations, which are also referred to here as stop locations, that result from deficiencies in road snapping based on perpendicular distance.

Figs. 1 and 2 show examples from Google Maps with suboptimal stop locations. In Fig. 1, the stop location is on a road without a parking spot or proper access to the building. Since the perpendicular distance was used, more suitable parking spots on the northern side of the building are ignored because the centroid of the building is closer to the road on the southern side.

Fig. 2 shows a typical road snapping problem in urban areas. In this case, it can be assumed that Google Maps used perpendicular distance in combination with a matching name of the road and the address. Consequently, the road segment in the north is chosen instead of the road segment in the southeast, even though the southeast road segment has the shortest perpendicular distance to the building's centroid. Nevertheless, the actual access to the building is shown with an orange dashed line. These two examples show that stop locations provided by conventional routing machines are sometimes inaccurate, which can lead to misleading or even dangerous stop locations in transport routing.

We provide an alternative approach to identify optimised stop locations that can be used to enhance routing for passenger transport services (e.g., door-to-door services). Our aim in this work is to investigate the potential of the cost distance approach for road snapping purposes. We also compare our results with results from a conventional routing machine. For the alternative approach, we looked for parameters that could be helpful in determining the path to buildings entrances. Finally, we made the assumption that access to building entrances consists of few vegetation cover, minimal slope of the terrain, and that the calculated path should not cross building footprints. We performed a cost distance analysis with preprocessed remote sensing data (multispectral imagery and LiDAR) and OpenStreetMap data to identify the so-called least cost paths. These can be used to determine the most likely path to a

road segment for any building or from any other point. The methodology of a cost distance analysis is explained in more detail in the next section.

2. FUNDAMENTALS OF COST DISTANCE

Even though the methodology of a cost distance analysis is well known and documented in the field of geoinformatics, we want to address a broad audience of mobility researchers. Therefore, we explain the basics of a cost distance analysis in this chapter.

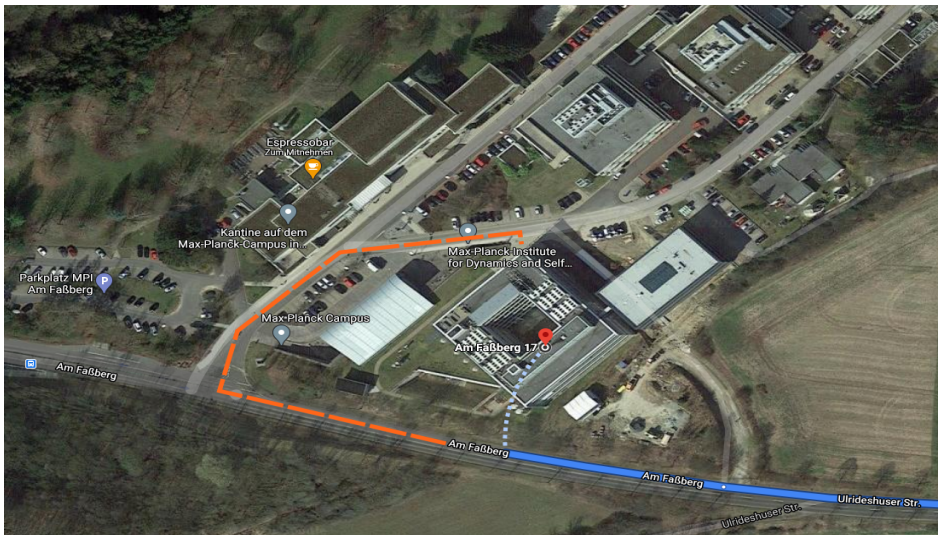

Fig. 1. Road snapping based on perpendicular distance from Google Maps shows a suboptimal snapping point without access to the building. The orange dashed line shows the correct access to the building. The dotted line depicts the access to the building by [10]

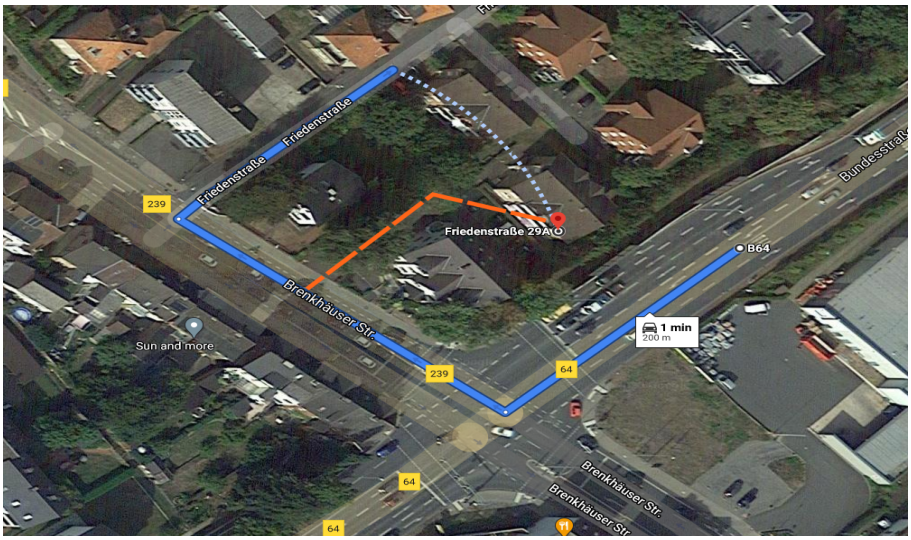

Fig. 2. Problems in current road snapping based on perpendicular distance. Even if a matching name of the road and the given address is used as an additional feature, the actual access to the building (orange dashed line) differs from the results provided by Google Maps. The dotted line depicts the access to the building by [11]

Cost distance is a ‘procedure for determining least cost paths across continuous surfaces, typically using grid representations’ [12]. Cost distance is based on the concept that movement in a continuous space requires different kinds of efforts. Therefore, both the length of a route and its difficulty influence

the time or cost of completing the route. The concepts of perpendicular distance and cost distance are compared in Fig. 3.

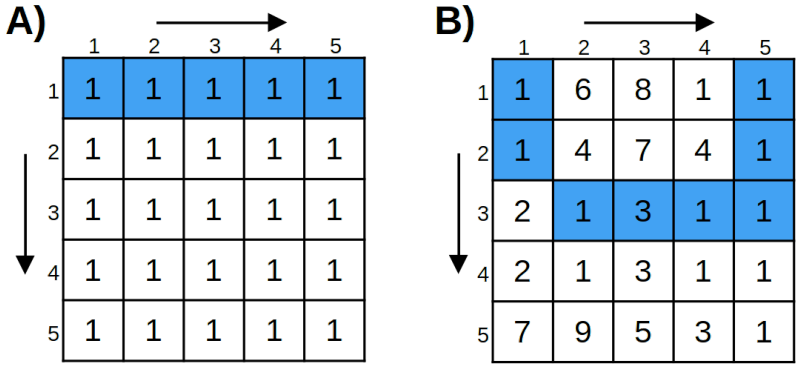

Fig. 3. Different distance metrics for finding the least cost path from (1:1) to (1:5) are highlighted in blue. A) Least cost path by the perpendicular distance (cost = 5). B) Least cost path by cost distance (cost~=~10)

Cost distance analyses, first defined in the 1950s as *cost based proximity analysis* [13], are widely used in areas such as cartography, archaeology, and computer science. Some possible applications are road planning [14] and reconstructing ancient roads with known start points and end points [15]. In many studies, the resulting paths are considered realistic [14; 16]. Today, the calculation of the least cost path based on cost distance is implemented in most geographic information systems (GISs) [17].

The least cost path between two points on a grid can be identified as follows. So-called source cells are given points that are referred to as possible destinations. In a cost distance analysis for every cell in a grid, the costs of paths to all source cells are computed and compared. The computational time scales with the number of source cells and the resolution of the grid or raster. A cost surface is needed as an input, which is a gridded representation of a graph that describes the cost per grid-cell. In a grid representation of a graph, cell centres represent the nodes of a graph with costs passing the nodes. They are connected via edges with adjacent nodes in a graph representation, respectively with adjacent cells in a grid representation of a graph [18]. Several possible neighbourhood types can determine the number and relations of the adjacent nodes of a cell. The most common ones are shown in Fig. 4.

The weights of the edges are calculated for horizontal and vertical neighbours as shown in Equation 2. Equation 3 was used for diagonal neighbours.

$$a_1 = \frac{(cost_1 + cost_2)}{2} \tag{2}$$

$$b_1 = \sqrt{2} * \frac{(cost_1 + cost_2)}{2} \tag{3}$$

When multiple parameters should be considered, different cost surfaces can be amalgamated into a merged cost surface. For a single cost surface or a merged cost surface, an accumulative cost surface and a backlink raster are calculated based on the cost surfaces. Therefore, in most implementations of cost distance analysis, Dijkstra's shortest path algorithm is used [19]. We used a well-established modification of this algorithm [14] to calculate the cost from each cell to the next source cell with the least cost, resulting in an accumulative cost surface and a backlink raster. The cells of a backlink raster contain coded direction values linked to the next cell on the least cost path to the source cell. The cells of an accumulative cost surface contain the actual cost of the path to the source cell. The backlink raster can then be used to track the least cost path from any cell to the next source cell with the least cost [20]. Fig. 5 shows a simplified illustration of a cost distance analysis with equally weighted cost surfaces.

3. METHODOLOGY

Our aim was to utilise cost distance to consider remote sensing data to calculate reasonable snapping points rather than to merely utilise perpendicular distance or relations between buildings and streets.

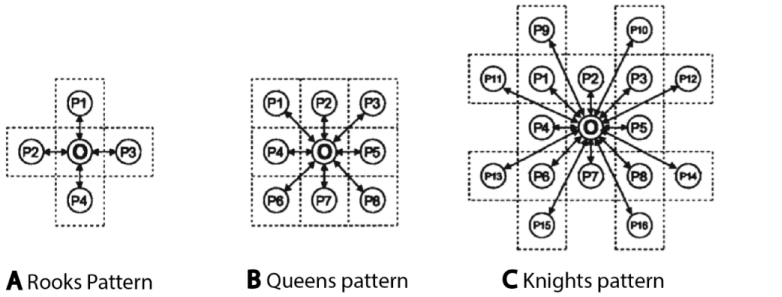

Fig. 4. Different neighbourhood types in gridded data representation: A) Rooks pattern - 4 linked neighbours for each cell. B) Queens pattern - 8 linked neighbours for each cell. C) Knights pattern - 16 linked neighbours for each cell [14]

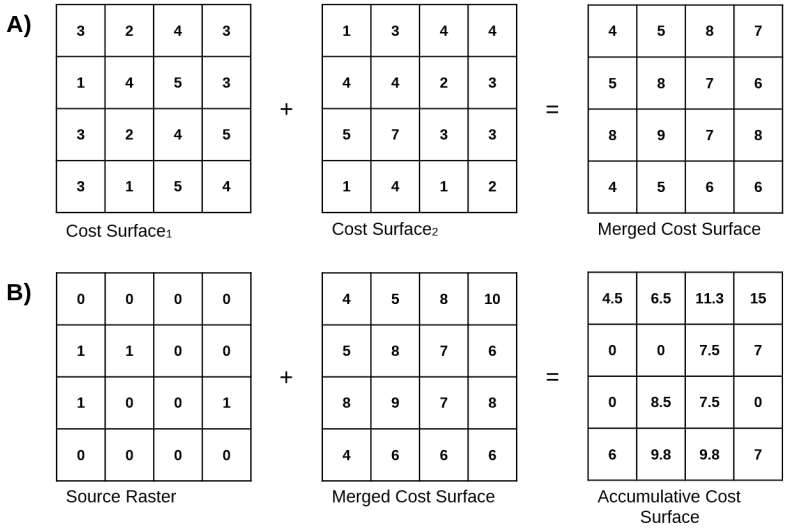

Fig. 5. A) Generation of a merged cost surface with equally weighted cost surfaces. B) Generation of the accumulative cost surface using the Queens pattern and Equation 2 and Equation 3. In each cell, the least costs to the next source cell are stored

We performed a cost distance analysis with cost surfaces of vegetation, slope, and building footprints. Therefore, we used multispectral images to determine the vegetation cover of cells based on the vegetation index normalised digital vegetation index (NDVI), LiDAR data when modelling the slope for each cell, thereby building footprints from OpenStreetMap to identify building cells and the road network from OpenStreetMap. We used thresholds for vegetation and slope to distinguish cells with no-vegetation from those with vegetation, as well as cells with passable slopes from those with not-passable slopes (*cf.* subsection 3.3).

The road network was then transformed to the source raster, where cells represent the existence of road segments, also referred to as source cells. To avoid unnecessary computation complexity due to a high number of source cells, we generated a source cell only every three metres along the road network, which we still consider as sufficiently accurate. Based on the building footprints, a binary raster with cells representing the existence or absence of buildings was created. The centroids of these building footprints will be called destination cells. Then, a merged cost surface was generated by merging and weighting the cost surfaces of vegetation, slope, and building footprints. Further, the accumulative cost surface and the backlink raster were calculated, with each cell in the accumulative cost surface representing the costs from the cell of interest to the source cell that can be reached with the least cost using the merged cost surface. The complete set of least cost paths between destination and source cells could then be generated using the coded direction values in the backlink raster. The last point of the least cost path described the source cell and, thus, the snapping point for the corresponding destination cell on the road network.

We evaluated our results by comparing the snapping points calculated using cost distance with the snapping points calculated using the conventional routing machine [21], which uses perpendicular distance. Therefore, we applied a so-called ideal snapping area, which represents the area where snapping points are considered correct. This area is based on manually set geographical points, which were used as ground truth data. For the considered area of interest (AOI), we set 495 ideal snapping points. We also evaluated the weighting of the classes (vegetation, slope, and building footprints). Only the odd numbers from 1 to 9 were used as weights for each class to reduce the time needed to perform calculations.

The number of total weight combinations was calculated according to Equation 4:

$$iterations_{weighting} = n^k, \tag{4}$$

where n is the number of possible weights (1,3,5,7,9), and k is the number of classes (vegetation, slope, and building footprints). Accordingly, we still had 125 different weighting combinations in total, meaning that we had 125 iterations of a complete cost distance analysis. As a result, we had obtained validation rates describing the percentage of snapping points within the ideal snapping area for each weight combination. This enabled a detailed analysis of a reasonable weighting and a comparison between the calculated snapping points based on cost distance and the snapping points determined by an OSRM.

3.1. Area of interest

Our area of interest (AOI) is located in the town of Höxter. Höxter is a medium-sized town in the southwest part of North Rhine-Westphalia (NRW) in Germany, which extends over an area of 158.16 km² and has a population of 29,112 [22]. The considered AOI is square-shaped, 1 km² in size, and located at the centre of Höxter. Most of the landcover of the AOI is residential areas, but there are some industrial complexes in the centre, north-west, and east areas of the AOI. The coordinates confining the geographical extent of the AOI are shown in Fig. 6.

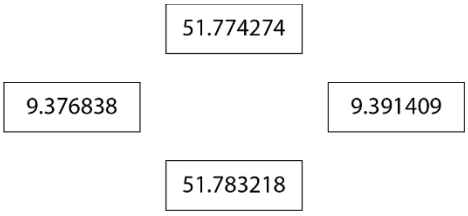

Fig. 6. Geographical extent of the AOI in Höxter with coordinates (EPSG:4326)

3.2. Data

We derived the road network and the building footprints from [23]. The aerial imagery and the LiDAR data can be obtained from the OpenGeoData project [24] for NRW. We filtered the road network data, resulting in a road network comprising only publicly accessible roads. Aerial imagery is a multispectral image containing red and near-infrared (NIR) wavelengths. Hence, the NDVI can easily be calculated using the following equation:

$$NDVI = \frac{NIR - Red}{NIR + Red} \tag{5}$$

The point cloud from the LiDAR data was used to generate a grid, each cell of which represents the slope value of the terrain. For each data point, a grid representation was generated with a cell size of 0.2 x 0.2 metres. Thus, the cells of the cost surfaces overlap exactly, and cells from different cost surfaces refer to the same position in the AOI.

3.3. Generation of snapping points by cost distance

We determined the thresholds for the classes of vegetation and slope on an empiric basis. The results are shown in Tab. 1.

Table 1

Classes and related weights for the cost surfaces

Class	Derived from	Threshold
vegetation	NDVI grid	> 0.2
no-vegetation	NDVI grid	<= 0.2
passable	Slope grid	<= 11
not-passable	Slope grid	> 11

A conservative threshold for no-vegetation cells is used with an NDVI of 0.2, and a threshold of a slope of 11 was chosen for not-passable cells. The threshold for not-passable cells depends on the resolution of the given cost surfaces. As for neighbourhood type, the Queens pattern was used (*cf.* Fig. 4).

3.4. Evaluation of snapping points

The ideal snapping point for a building was predetermined as the point on the road network that is most likely accessible from the building. If a building had more than one possible access point from the road network, multiple ideal snapping points were set. A line from each building's centroid to its ideal snapping point on the road network was generated, ensuring a spatial relation between the two points and meaning that no additional relation is required. Since we were not able to find entities providing pick-up and drop-off locations from passenger transport services as ground truth data, we manually set 495 reference points as ground truth data. The first vertex of the line represents the building's centroid, and the second vertex represents the ideal snapping point. This line and its second vertex can be compared to lines from the building's centroid to calculated snapping points by cost distance and to the snapping points obtained from OSRM.

Considering the maximum acceptable distance from the ideal snapping point to calculated snapping points results in a circular area around the ideal snapping point. In the first validation step, we checked whether the calculated snapping point was located inside this area. However, if the distance between the building and the road is short, a calculated snapping point might be validated even if the ideal snapping point is in another direction from the building's centroid position. We also considered the direction in the next validation step by comparing the difference in bearings between the two lines from the building's centroid to the ideal snapping point and to the calculated snapping point. Thus, a maximum acceptable degree for the angle between these two lines was used as a threshold. This led to an area around the ideal snapping point that is called the ideal snapping area (Fig. 7).

Consequently, a calculated snapping point was validated only if the difference between the calculated and manually set ideal snapping point regarding distance and direction was below the predefined thresholds. As the distance between the points and the same maximum acceptable difference in bearings grows, the size of the ideal snapping area increases until the area is defined only by the radius of the circle based on the maximum allowable distance. During evaluations, the maximum acceptable distance between the points was set to 25 metres, and the maximum difference in bearings between the line from the building's centroid to the calculated snapping point and the ideal snapping point was set to 70°.

4. RESULTS AND DISCUSSION

In the AOI, 403 out of 495 calculated snapping points obtained according to perpendicular distance using the nearest API from OSRM are inside the ideal snapping area, which translates to a validation rate of 81.4%.

The calculated snapping points by cost distance result in different validation rates depending on the weight of each class. Thus, the validation rate varies from 84.8% to 90.3%.

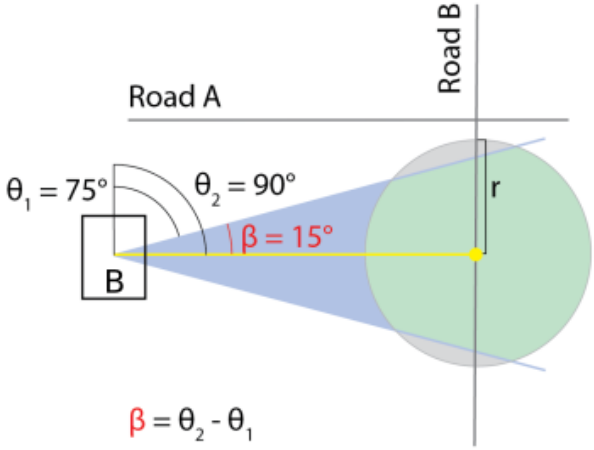

Fig. 7. Concept of the ideal snapping area. The ideal snapping area is defined by a vector (yellow line) from the centroid of building *B* to the ideal snapping point (yellow point), a maximum distance (defined by *r*), and a direction t (defined by the maximum allowed difference in bearings). In this example, the allowable difference in bearings β between θ_1 and θ_2 is 15°. The ideal snapping area is shown in green. An ideal snapping area restricted only by *r* could lead to acceptable snapping points on Road A and Road B if *r* is sufficiently large

A detailed analysis of the weighting and the validation rate allows the weighting of each parameter to be scored. Fig. 8 shows the enhancement of the validation rate compared to the validation rate without weighting parameters for each weight combination. A trend can be seen – specifically, the higher the cost of the parameter slope, the higher the validation rate. Meanwhile, the lower the cost of the parameter slope, the lower the validation rate. For the parameters of vegetation and building footprints, no such trend can clearly be identified. However, the highest validation rates were achieved with a medium cost of vegetation and building footprints.

Fig. 8 depicts the distribution of the validation rates. All validation rates of our cost distance approach are higher than the validation rates based on perpendicular distance, with the highest validation rates being achieved most frequently.

We performed an additional cost distance analysis for a small extract of the AOI where the building in Fig. 2 is located with more weight combinations. Fig. 10 shows the impact of weighting on the quality of the least cost path. The most reasonable least cost path (blue) is achieved with a weighting combination of 5 (vegetation), 7 (slope), and 3 (building footprints), whereas other weighting combinations did not reflect the actual access to the building.

It has been shown that the cost distance analysis outperforms the perpendicular distance approach in the AOI. However, the presented approach by cost distance still has to be tested in other regions and on a larger scale to draw a generalised conclusion. Further investigations are difficult to perform due to the lack of access to sufficiently high-resolution open-source data. When such data are available, the presented approach should be tested and investigated in other regions with different terrains (e.g. mountainous, flat, urban, and rural areas). Since weighting has a substantial impact on the quality of the least cost paths, the proportion of vegetation, the geomorphology, and the buildings in a selected AOI influence the optimal combination of weights; hence, the optimal weighting depends on the AOI. Unlike satellite imagery, cloud cover is hardly an issue in the aerial imagery used, as these images are taken irregularly by aircraft during good weather conditions. As a result, not much data is available. The time of capture of such aerial images (season) may also play a role, as the condition of vegetation varies across seasons. Consequently, the quality of the results may be affected when aerial images are used in winter. If further sufficient data become available, we recommend investigating this assumption in more detail in the future, for example, by comparing aerial imagery from summer and winter.

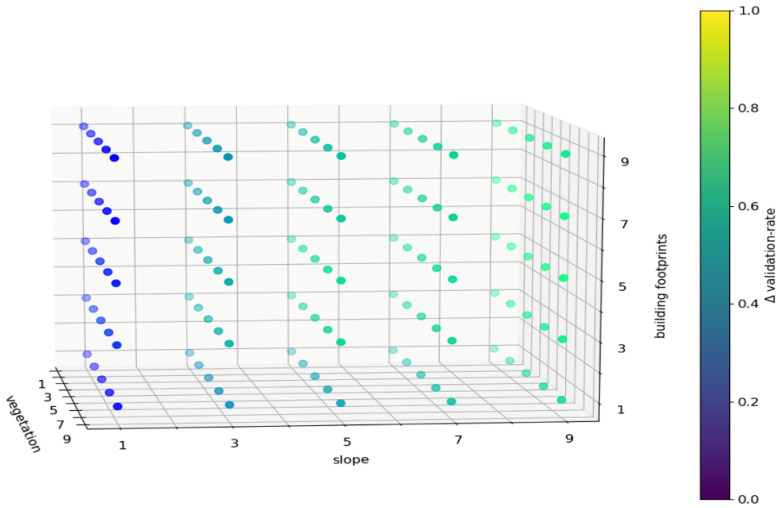

Fig. 8. Effect of weights on the validation rate

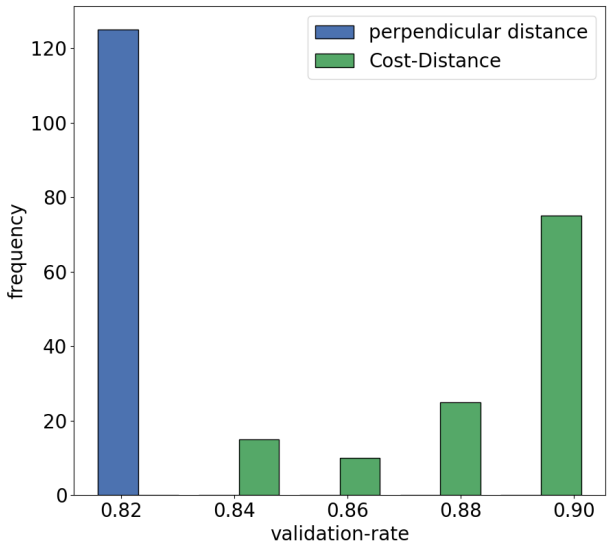

Fig. 9. A histogram of the validation rates for calculated snapping points based on perpendicular distance and cost distance

Due to the high computational complexity of cost distance analysis, we recommend neglecting the accuracy of the parameters to a certain degree. In particular, reducing the number of source cells can notably improve the calculation time. At the same time, a density of the source cells on the road network with a spacing of three to five metres can provide adequately accurate results. Coarsening the grid resolution would also improve the calculation time. However, at a resolution of 0.3 x 0.3 metres, objects like fences or other obstacles could not be considered sufficient. Regarding this problem, Shannon, Whittaker and Nyquist state that the cell size of the grid should be at least $2*\sqrt{2}$ times smaller than the smallest detail to be kept [25; 26]. Hence, a sufficiently high resolution is required for small objects, but the resolution of the source cells and, thus, the accuracy of the snapping points can be reduced to improve performance. Alternatively, the object-based image classification method can be used when working with lower resolutions in point cloud data, similar to the detection and filtering of vehicles and trees using LiDAR or aerial imagery [27; 28]. Nevertheless, if the duration of the preprocessing stage is not relevant, our approach could be used in the future (e.g., in passenger transportation with door-to-door services to precompute optimised pick-up and drop-off locations for any address within a certain area).

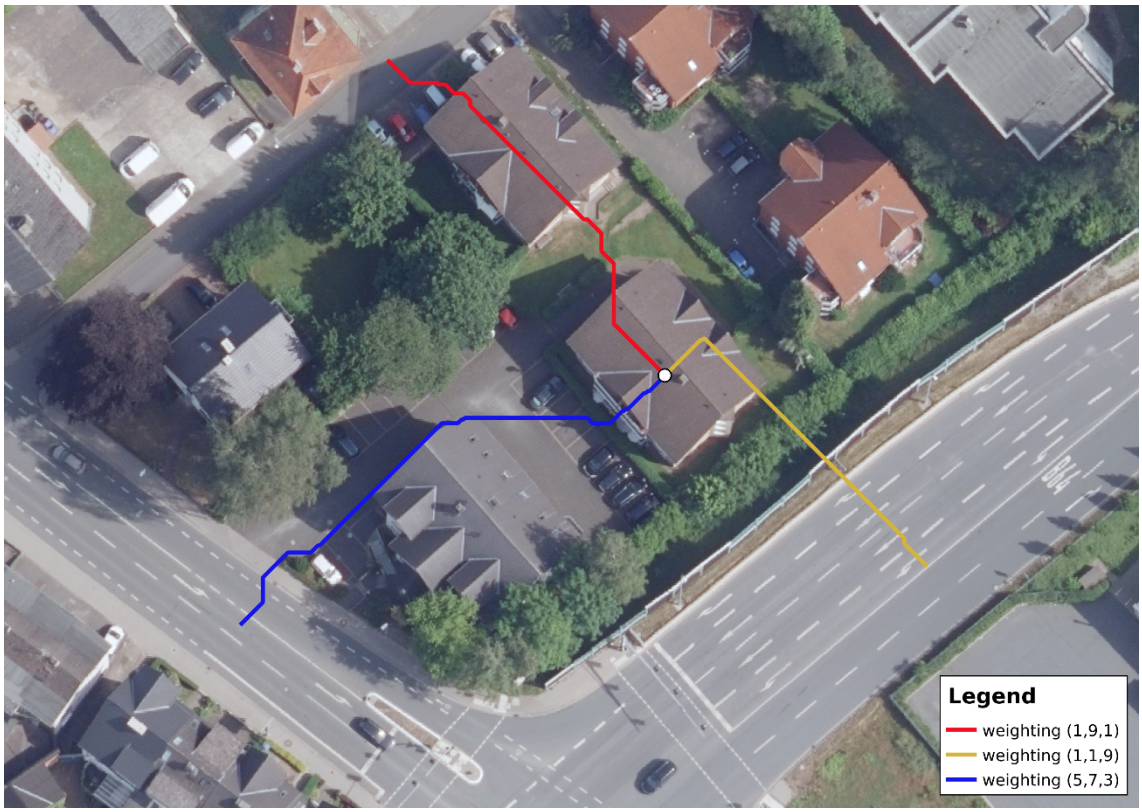

Fig. 10. The influence of weighting using the example from Fig. 2. Weighting with a cost of 5 (vegetation), 7 (slope), and 3 (building footprints) results in the least cost path (blue), which reflects the most realistic access to the building, whereas the other shown weighting combinations do not lead to realistic results

Performing a cost distance analysis on grids results in least cost paths that tend to zig-zag and are longer than the direct paths in reality. This can be enhanced by using larger neighbourhoods for each cell (*cf.* Fig. 4) or using the approach presented in [17; 29]. However, in this study, only the point-to-point relation is important, and the geometry of the path is not relevant for the snapping point.

Obviously, transportation data is required to obtain a sufficient number of realistic stop locations of vehicles using buildings' addresses as reference data to perform a more detailed evaluation. However, such data is hard to obtain due to privacy issues.

The approach that we have presented could be used in the future to increase the accuracy of stop locations for door-to-door transport services. This can be useful for transportation network companies that offer such services, especially in combination with public transport or multimodal transportation. Time delays that are caused by finding a reasonable stop location can interfere with the plans of future trips and time schedules. This could be prevented by applying pre-calculated optimised stop locations.

5. CONCLUSION AND OUTLOOK

We have shown that classical road snapping techniques based on perpendicular distance have some weaknesses. The cost distance approach using a weighted grid presented here outperforms the road snapping by perpendicular distance in the AOI.

The given conditions of the AOI (vegetation cover, the slope of the terrain, and the structure of the settlement) notably impact the cost distance analysis and the optimal weight combination.

Applying higher-resolution remote sensing data could increase the accuracy of the classification process and, consequently, the least cost paths and snapping points. However, this would also further increase the already high computation time of this process.

The cost distance approach presented in this study provides more reasonable snapping points than the snapping points derived by perpendicular distance, as evidenced by the higher validation rate (up to 90.3% vs 81.4%). The highest validation rates are achieved most frequently, indicating the reliability of the proposed approach.

We have demonstrated the basic idea and the possible potential of the cost distance approach for optimizing pick-up and drop-off locations in transport routing and encourage further research in this field. Such research could improve the quality and accuracy of road snapping in real-world applications.

References

1. Pereira, F.C. & Costa, H. & Pereira, N.M. An off-line map-matching algorithm for incomplete map databases. *European Transport Research Review.* 2009. Vol. 1. P. 107-124.
2. Hashemi, M. & Karimi, H.A. A critical review of real-time map-matching algorithms: Current issues and future directions. *Computers, Environment and Urban Systems*. 2014. Vol. 48. P. 153-165. Elsevier BV.
3. He, M. & et al. An enhanced weight-based real-time map matching algorithm for complex urban networks. *Physica A: Statistical Mechanics and its Applications.* 2019. Vol. 534. Elsevier BV.
4. Knapen, L. & et al. Likelihood-based offline map matching of GPS recordings using global trace information. *Transportation Research Part C: Emerging Technologies.* 2018. Vol. 93. P. 13-35.
5. Quddus, M.A. & Ochieng, W.Y. & Noland, R.B. Current map-matching algorithms for transport applications: State-of-the art and future research directions. *Transportation Research Part C: Emerging Technologies*. 2007. Vol. 15. P. 312-328.
6. Zhang, D. & Dong, Y. & Guo, Z. A turning point-based offline map matching algorithm for urban road networks. *Information Sciences.* 2021. Vol. 565. P. 32-45.
7. Huabei, Y. & Wolfson, O. A weight-based map matching method in moving objects databases. 2004. In: *Proceedings. 16th International Conference on Scientific and Statistical Database Management.* 2004. P. 437-438.
8. *Google Maps. Offline Map Matching.* Available at: https://developers.google.com/maps/documentation/roads/snap. 2021.
9. *Open Source Routing Machine. Nearest Service.* Available at: http://project-osrm.org/docs/v5.5.1/api/#services 2021.
10. *Google Maps. Query Am Fassberg 17, Goettingen, Germany.* Available at: https://www.google.de/maps/place/Am+Fa%C3%9Fberg+17,+37077+G%C3%B6ttingen/@51.5605163,9.9660005,681m/data=!3m2!1e3!4b1!4m5!3m4!1s0x47a4d503fe8feebf:0x2c2b721e7a1fcf62!8m2!3d51.5605163!4d9.9681893 2021.
11. *Google Maps. Query Friedenstrasse 29A, Hoexter, Germany.* Available at: https://www.google.de/maps/place/Am+Fa%C3%9Fberg+17,+37077+G%C3%B6ttingen/@51.5605163,9.9660005,681m/data=!3m2!1e3!4b1!4m5!3m4!1s0x47a4d503fe8feebf:0x2c2b721e7a1fcf62!8m2!3d51.5605163!4d9.9681893 2021.
12. Smith, M.J. & Goodchild, M.F. & Longley, P. *Geospatial Analysis*. 2018. Winchelsea Press. ISBN: 9781912556038.
13. Warntz, W. Transportation, social physics, and the law of refraction. *The Professional Geographer.* 2005. Vol. 9. P. 2-7.
14. Yu, C. & Lee, J. & Mandy, J. Extensions to least-cost path algorithms for roadway planning. *International Journal of Geographical Information Science.* 2003. Vol. 17. P. 361-376. Taylor & Francis.
15. van Leusen, M. *Pattern to process: methodological investigations into the formation and interpretation of spatial patterns in archaeological landscapes*. 2002. Rijksuniversiteit Groningen.
16. Rees, G. Least-cost paths in mountainous terrain. *Computers & Geosciences*. 2004. Vol. 30. P. 203-209.

17. Douglas, D. Least-cost path in GIS using an accumulated cost surface and slopelines. University of Toronto Press Inc. (UTPress). *Cartographica: The International Journal for Geographic Information and Geovisualization*. 1994. Vol. 31. P. 37-51.
18. Collischonn, W. & Pilar, J. A direction dependent least-cost-path algorithm for roads and canals. *International Journal of Geographical Information Science.* 2000. Vol. 14. P. 397-406.
19. Cormen, T. H. & et al. *Introduction to Algorithms*. MIT Press. 2001.
20. Xu, J. & Lathrop, R.G. Improving cost-path tracing in a raster data format. *Computers & Geosciences.* 1994. Vol. 20. P. 1455-1465. ISSN: 0098-3004.
21. Open Source Routing Machine. Routing Engine. Available at: http://project-osrm.org/ 2021.
22. Landesdatenbank-NRW. Die Landesdatenbank NRW. Available at: https://www.landesdatenbank.nrw.de/ldbnrw/online/dat 2018.
23. OpenStreetMap contributors. Available at: https://planet.osm.org. https://www.openstreetmap.org/ 2017.
24. Information und Technik - Nordrhein-Westfalen. OpenGeoData. Available at: https://www.opengeodata.nrw.de/produkte/ 2017.
25. Huijsmans, D.P. & Vossepoel, A.M. *Informatie in Gedigitaliseerde Beelden.* Volume One: Introduction. 1989.
26. Oppenheim, A.V. & et al. *Signals & Systems.* Prentice Hall. 1983. ISBN: 9780138147570LCCN: 96019945.
27. Liu, Y. & Monteiro, S. & Saber, E. Vehicle detection from aerial color imagery and airborne LiDAR data. 2016. *IGARSS.* P. 1384-1387.
28. Yang, B. & Sharma, P. & Nevatia, R. Vehicle detection from low quality aerial LIDAR data. In: *2011 IEEE Workshop on Applications of Computer Vision (WACV)*. 2011. P. 541-548.
29. Xu, J. & Lathrop, R.G. Improving simulation accuracy of spread phenomena in a raster-based Geographic Information System. *International Journal of Geographical Information Systems.* 1995. Vol. 9. No. 2. P. 153-168.

Received 12.02.2021; accepted in revised form 08.03.2022