

1 **Combining road network data from OpenStreetMap with an authoritative database**

2 Grzegorz Szwoch, Ph.D.

3 Gdansk University of Technology, Department of Multimedia Systems

4 80-223 Gdansk, Poland, Narutowicza 11/12

5 grzegorz.szwoch@pg.edu.pl

6

7 **Abstract**

8 Computer modeling of road networks requires detailed and up-to-date dataset. This paper
9 proposes a method of combining authoritative databases with OpenStreetMap (OSM) system.
10 The complete route is established by finding paths in the graph constructed from partial data
11 obtained from OSM. In order to correlate data from both sources, a method of coordinate
12 conversion is proposed. The algorithm queries road data from OSM and provides means of
13 locating any point on the route in both datasets. A method of calculating the distance of any
14 route point from the origin, and conversion between the distance and geographic coordinates,
15 is described. Next, the location of any route point in the authoritative database is converted to
16 the calculated route distance, which establishes a relation between the two data sources.
17 Additionally, a method of estimating road curvature is proposed. The algorithm is validated in
18 series of experiments. The proposed algorithm may be beneficial for researchers who collect
19 datasets needed for computer simulations, e.g. for evaluation of optimal speed limits, and it
20 shows usefulness of OSM in transportation related research.

This material may be downloaded for personal use only. Any other use requires prior permission of the American Society of Civil Engineers. This material may be found at <https://ascelibrary.org/doi/10.1061/JTEPBS.0000215>

Post-print of: Szwoch G.. Combining Road Network Data from OpenStreetMap with an Authoritative Database. JOURNAL OF TRANSPORTATION ENGINEERING, PART A: SYSTEMS. Vol. 145, iss. 2 (2019). DOI: 10.1061/JTEPBS.0000215

21 INTRODUCTION

22 Computer models of road networks are valuable tools for simulation and evaluation of the
23 road infrastructure. Computer simulations require detailed, complete and up-to-date
24 information on all road network elements. Authoritative databases, maintained by the
25 authorities administering the road infrastructure in a specified region, constitute official
26 sources of such data. However, these datasets may not be sufficient for performing computer
27 simulations, as they may not have all necessary information (such as speed limits), and they
28 may not be updated with a sufficient frequency. Therefore, additional data may be obtained
29 from alternative sources in order to supplement the authoritative databases. Volunteered
30 Geographic Information (VGI) systems have been a hot topic in the last ten years. The most
31 popular example of a VGI system is the OpenStreetMap (OSM) service (OSM Contributors
32 2018) which is a collaborative, crowdsourced effort to create an open sourced, worldwide
33 geographic database. The main strength of the OSM is the number of volunteering editors
34 across the world. Thanks to that, the OSM has currently an advantage over commercial
35 mapping services in terms of coverage and up-to-date information. Data quality in OSM is an
36 important issue, as VGI systems are created mostly by amateurs. However, with a large
37 number of editors and automated maintenance algorithms, most errors are corrected.
38 Therefore, the OSM database is a valuable source of information that may be used to
39 complement authoritative road network databases. The official databases contain detailed data
40 collected with professional equipment, while the OSM provides up-to-date information
41 gathered by the community.

42 Utilizing separate data sources brings the problem of combining information from
43 distinct databases. The main contribution of this paper is proposing a solution to two most
44 important problems related to combining the OSM data with an authoritative database. The
45 first one is related to the fact that OSM data describes only short road sections and there is no



46 simple way to obtain ordered data on a complete trunk road, including separate lanes. This
47 problem is addressed by a proposed algorithm that constructs a continuous route from partial
48 OSM data. The second, more difficult problem is how to relate data from both sources with
49 each other, so that any location in one database may be found in another one. For example, an
50 authoritative database may identify road elements only by their reference position within the
51 road, while the OSM data is based solely on geographic coordinates. The proposed algorithm
52 provides a method of conversion between these coordinates. As a result, a combined data
53 source, suitable e.g. for computer modeling of road network, is obtained.

54 In the literature related to analysis of OSM data, many publications concern assessing
55 and enforcing data quality. For example, comparison of the OSM database with official
56 geospatial databases was described in (Haklay 2010) or (Brovelli et al 2017), and a
57 comparison with a commercial system was performed in (Ciepluch et al 2010). Quality of
58 OSM data in particular regions was evaluated e.g. for France (Girres and Touya 2010) and
59 Germany (Neis et al 2011). Methods of assessing the OSM data quality using metrics and
60 automated frameworks were researched e.g. in (Mooney et al 2010), (Barron et al 2014) and
61 (Jilani et al 2013). Research on the OSM data analysis includes geographic knowledge
62 extraction and semantic similarity (Ballatore et al 2013), analysis of the OSM networks
63 growth (Corcolara et al 2013), a method of automated highway tag assessment (Jilani et al
64 2014), a map-reduce system for extracting spatial data (Alarbi et al 2014) and extracting
65 multi-lane roads data (Li et al 2014). Application of the OSM data for traffic simulation was
66 described in (Zilske et al 2011), and for routing and travel time calculation in (Huber and Rust
67 2016). Other road-related research includes improving image-based characterization of roads
68 (Chen et al 2014) and building a multimodal urban network model (Gil 2015). There are also
69 publications on utilizing the OSM data in other areas, e.g. for generating a web-based 3D city
70 model (Over et al 2010), automated identification and characterization of parcels (Long and



71 Liu 2016), identifying elements at risk in case of flooding (Schelborn et al 2014),
72 hydrological and hydraulic modeling (Schellekens 2014) or railway applications (Rahmig
73 and Simon 2014). There are also numerous publications related to sociological aspects of the
74 OSM system. The only work related to the problem of combining the OSM data with
75 authoritative registers, known to the author, is based on polygon approach (Fan et al 2016)
76 which is not applicable to the problem presented here.

77 This paper proposes a novel approach to enhancing an authoritative road network
78 dataset with information obtained from the OSM database. Although the presented examples
79 are based on the Polish authoritative database, the proposed method should be applicable to
80 any road network worldwide, allowing researchers to supplement their road network datasets
81 with additional data. The remaining parts of the paper describe the algorithm that collects
82 road data from the OSM, constructs a continuous route from individual elements, and
83 establishes a relationship between coordinates in the OSM database and the authoritative data
84 source. Evaluation of the proposed method in series of experiments is presented and the paper
85 ends with Conclusions.

86

87 **CONSTRUCTING A ROUTE FROM THE OSM DATA**

88 The purpose of the algorithm presented in this Section is to explore the OSM database in
89 order to obtain data about continuous routes within a road network. As a specific example, a
90 complete trunk road within the administrative boundaries of a region, will be considered. The
91 OSM does not provide data in the desired form, the dataset has to be constructed from partial
92 OSM data. First, all road elements are obtained from the OSM database, then they are ordered
93 and merged, so that the full route geometry is created.



94 There are three main elements that constitute data stored in the OSM database
 95 (OpenStreetMap Wiki 2018). A *node* n is a single point, described by its geographic
 96 coordinates, an unique identifier and metadata (“tags”):

$$97 \quad n_i = \langle \varphi_i, \lambda_i \rangle \quad (1)$$

98 where φ_i is the latitude, λ_i is the longitude, i is the node identifier.

99 A *way* w is an ordered sequence of nodes forming a logical structure, e.g. a road section.
 100 Ways are described with an unique identifier, a list of node identifiers and metadata. A way
 101 connecting nodes n_i and n_j may be denoted as:

$$102 \quad w_{i,j} = \langle n_i, n_{k,1}, n_{k,2}, \dots, n_j \rangle \quad (2)$$

103 where n_k are the intermediate nodes. Ways may be unidirectional (traffic only in the direction
 104 indicated by the order of nodes) or bidirectional (traffic in both directions, i.e. $w_{i,j} \equiv w_{j,i}$). In
 105 case of dual carriageways, each separated lane has to be represented with an individual,
 106 unidirectional way.

107 *A relation* R is a set of ways that form a logical structure, e.g. the complete trunk road.
 108 Each way may belong to multiple relations. A relation is described by its identifier, a set of
 109 ways and metadata:

$$110 \quad R_k = \{w_{k,1}, w_{k,2}, \dots, w_{k,N}\}. \quad (3)$$

111 An example of a map view of ways and nodes belonging to a relation is shown in Fig. 1. It
 112 should be noted that a relation is an unordered collection of ways, there is no information on
 113 connectivity of the ways. Therefore, such information has to be established by analyzing the
 114 OSM data. Let’s define a *route* s as an ordered sequence of connected ways, as opposed to an
 115 unordered relation:

$$116 \quad s_k = \langle w_{k,1}, w_{k,2}, \dots, w_{k,N} \rangle, \quad w_k \in R_k. \quad (4)$$

117 This structure represents an actual route, e.g. a trunk road. Let's also define a *connector* c as a
118 node which is a junction between two or more ways:

$$119 \quad n_i \equiv c(j, k) \Leftrightarrow n_i \in w_j \wedge n_i \in w_k. \quad (5)$$

120 If the whole route is composed only of single carriageways, the problem of route construction
121 is trivial and it may be solved by starting with the first way and iteratively finding a way that
122 has a connector with a previously found way. However, typical trunk roads comprise both
123 single and dual carriageway sections, and roundabouts are prevalent, so the connectors often
124 merge more than two ways. It is convenient to view a route from one of its terminations, as
125 two separate routes: a *forward* and a *backward* route (the backward meaning a direction
126 opposite to the traffic). These two routes alternate between single and dual carriageway
127 sections, with junctions and roundabouts along the way, which results in constant splitting
128 and merging of these two routes.

129 In the proposed algorithm, the problem of establishing these routes in an unordered set
130 of ways was approached by employing the graph theory (Sedgewick and Wayne 2011). The
131 route is represented with a directed graph, in which connectors form the graph vertices and
132 ways are its edges. Additionally, nodes that terminate the relation on both its ends (*endpoints*)
133 are also added as vertices. In the next stage, all ways belonging to the relation are examined
134 one after another. For most ways, the first node n_i and the last node n_j are the only connectors.
135 In this case, the edge (n_i-n_j) is added to the graph, and if the way is bidirectional, the reversed
136 edge (n_j-n_i) is added as well. Unidirectional ways may be recognized by presence of the
137 *oneway=yes* tag in their metadata. If there are more than two connectors in a single way, this
138 way is divided into parts having two connectors, and each part is added to the graph as above.
139 After all ways are analyzed, a directed, cyclic graph, without self-loops and with non-
140 weighted edges, is obtained (Fig. 2).



141 Establishing the forward and the backward routes requires finding simple paths
142 between the pairs of related endpoints, using the depth-first search algorithm. Assuming that
143 the OSM data is valid and complete, there should be exactly one path for each route. The
144 backward path is then reversed, so that both routes originate at the same end of the road.
145 Finally, the route is constructed by iterating over the graph edges along both paths. Ways
146 representing single carriageways are present in both paths, while unidirectional ways are only
147 found in one path. Therefore, the final route data is composed of sections representing single
148 and dual carriageways.

149 Once the route is established, it is possible to traverse it and analyze metadata that was
150 obtained from the OSM database. For example, information on speed limits along the route
151 may be obtained. Metadata of ways may contain the *maxspeed* tag describing the speed limit
152 value, other tags describe limits for specific vehicle classes or conditions (e.g. day or night).
153 Additionally, the *source:maxspeed* tag explains the reason for imposing a speed limit, such as
154 a road sign (*sign*) or area type (*rural, urban*). As a result, speed limit data for various route
155 sections may be obtained and added to the authoritative dataset.

156

157 **ESTABLISHING A RELATION BETWEEN OSM DATA AND THE** 158 **AUTHORITATIVE DATASET**

159 In order to relate the constructed route data with the authoritative dataset, it is necessary to
160 provide a method which identifies any point on the route in both datasets. The main problem
161 is that these sources may use different methods of describing the location of any route
162 element. The authoritative databases often use cumulative distances computed within
163 reference sections, while the OSM database uses geographic coordinates. A problem of
164 converting the coordinates and combining both data sources is solved with the algorithm
165 presented in this Section.



166

167 **Calculating route distances**

168 The first problem is how to describe a location of any point p on the route. Let's define a
 169 *route distance* d of any route point p as the length of the route from its origin to p , expressed
 170 in physical units (meters, miles, etc.). In the OSM database, each way is represented with a
 171 list of nodes, and each node is described with its geographic coordinates. Therefore, route
 172 distances may be computed by summing up distances between pairs of nodes along the route.
 173 There are multiple ways of calculating the distance between two locations given by their
 174 geographic coordinates: $p_1 = (\varphi_1, \lambda_1)$ and $p_2 = (\varphi_2, \lambda_2)$. One of the most often used approaches
 175 utilizes the haversine formula (Sinnott 1984) which calculates the great circle distance r
 176 between two points as:

$$177 \quad r = 2r_E \cdot \arctan2(\sqrt{a}, \sqrt{1-a}), \quad (6)$$

178 where:

$$179 \quad a = \sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos(\varphi_1) \cdot \cos(\varphi_2) \cdot \sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right) \quad (7)$$

180 and r_E is the Earth radius. The value of r_E depends on the latitude, a mean value of 6371001 m
 181 is typically used in calculations. This value may also be estimated as:

$$182 \quad r_E(\varphi) = \sqrt{\frac{(r_1^2 \cos(\varphi))^2 + (r_2^2 \sin(\varphi))^2}{(r_1 \cos(\varphi))^2 + (r_2 \sin(\varphi))^2}} \quad (8)$$

183 where $r_1 = 6378137$ m, $r_2 = 6356752$ m.

184 The haversine formula may introduce errors up to 0.3%. A more accurate method of
 185 computing the distance is based on an inverse solution to the Vincenty's formula (Vincenty
 186 1975), providing accuracy up to 0.5 mm, at the cost of increased computation time. Both
 187 methods are used in the proposed algorithm. Other methods are also possible, such as



188 projection of points to the Cartesian coordinate system, e.g. the Universal Transverse
189 Mercator (UTM) (Snyder 1987) and computing an Euclidean distance between the points.

190 The procedure iterates over all nodes along the route, computes distances between
191 each pair of nodes, and stores the accumulated distance of each node from the route origin.
192 Regional regulations may declare that the distance has to be measured along the route axis,
193 and in case of dual carriageways, the axis is situated between the lanes (GDDKiA 2012).
194 Calculated distances of the final node may be different for the forward and the backward
195 route. Therefore, for each route section composed of dual carriageways, the local distance is
196 calculated from the beginning of the section, for the forward and the backward route
197 separately. For the final node of the section, two distances d_f and d_b are obtained. Then, for
198 each node i on the forward route section, the previously computed distance d_i is rescaled:

$$199 \quad d'_i = d_i \frac{d_f + d_b}{2d_f}. \quad (9)$$

200 With this approach, differences between the forward and the backward route are averaged.
201 Also, according to this recommendation, the distance should be calculated straight through
202 roundabouts. Therefore, all ways that belong to roundabouts (having a *junction=roundabout*
203 tag) are found. For each roundabout, all its nodes are replaced with a centroid, calculated as:

$$204 \quad \varphi_c = \frac{1}{6A} \sum_{i=0}^{n-1} (\varphi_i + \varphi_{i+1})(\varphi_i \lambda_{i+1} - \varphi_{i+1} \lambda_i) \quad (10)$$

$$205 \quad \lambda_c = \frac{1}{6A} \sum_{i=0}^{n-1} (\lambda_i + \lambda_{i+1})(\varphi_i \lambda_{i+1} - \varphi_{i+1} \lambda_i) \quad (11)$$

206 where A is the polygon area:

$$207 \quad A = \frac{1}{2} \sum_{i=0}^{n-1} (\varphi_i \lambda_{i+1} - \varphi_{i+1} \lambda_i) \quad (12)$$

208 The latitude and longitude values are averaged directly, which is inaccurate, but sufficient for
209 this particular purpose, because the error is not large due to proximity of the nodes, and



210 determining the exact centroid of a roundabout is not essential here. The centroid replaces all
 211 nodes in a roundabout and is used in the distance calculation.

212

213 **Calculating geographic coordinates given route distance**

214 Geographical coordinates of the route are obtained from the OSM database and they are only
 215 known for the nodes. The next step is to compute values (φ, λ) for any point on the route,
 216 given its distance d from the route origin. A point p with a known distance d is situated on the
 217 route section between two nodes:

$$218 \quad n_1 = \max_i(n_i \mid d_i \leq d) \quad (13)$$

$$219 \quad n_2 = \min_i(n_i \mid d_i > d) \quad (14)$$

220 where n_1 and n_2 are the previous and the next node on the route relative to p , respectively, d_i is
 221 the route distance of n_i . These two nodes are found with the bisection algorithm in the ordered
 222 list of node distances. Next, the direction of the route from n_1 to n_2 is calculated. A *bearing* θ_i
 223 of a node n_i is the angle between the North and the direction to the next node on the route.
 224 The value may be calculated with the haversine formula:

$$225 \quad \theta = \arctan2(\sin(\lambda_2 - \lambda_1) \cdot \cos \varphi_2, \cos \varphi_1 \cdot \sin \varphi_2 - \sin \varphi_1 \cdot \cos \varphi_2 \cdot \cos(\lambda_2 - \lambda_1)) \quad (15)$$

226 A more accurate estimation of bearing may also be calculated iteratively with the Vincenty's
 227 formula. In the presented algorithm, the distance and bearing between each pair of route
 228 nodes are pre-computed when the route is constructed.

229 A point with distance d is situated on a straight line connecting nodes n_1 and n_2 .
 230 Therefore, bearing between these nodes is the same as between n_1 and p , and the distance of p
 231 from n_1 is also known ($d - d_1$). As a result, calculation of p may be performed with the
 232 haversine formula again (Sinnott 1984):

$$233 \quad \varphi_p = \arcsin\left(\sin \varphi_1 \cos \frac{d - d_1}{r_E} + \cos \varphi_1 \sin \frac{d - d_1}{r_E} \cdot \cos \theta_1\right) \quad (16)$$



$$234 \quad \lambda_p = \lambda_1 + \arctan 2 \left(\sin \theta \cdot \cos \varphi_1 \cdot \sin \frac{d-d_1}{r_E}, \cos \frac{d-d_1}{r_E} - \sin \varphi_1 \cdot \sin \varphi_p \right) \quad (17)$$

235 where r_E is the Earth radius. It is also possible to calculate these values iteratively using a
 236 direct solution to the Vincenty's formula (Vincenty 1975).

237

238 **Calculating route distance given geographic coordinates**

239 The inverse problem of finding the route distance d of a point p with known coordinates (φ ,
 240 λ) is more complex. Assuming that p is situated sufficiently close to the route, the problem
 241 may be solved by finding a point on the route with a minimal distance from p :

$$242 \quad d = \min_d (|p - p_d|) \quad (18)$$

243 where p_d is a point with the route distance d . In the initial experiments, an approach based of
 244 minimization of distance between p and an iteratively found p_d , using methods such as
 245 Brent's algorithm (Brent 1973), was employed. However, this method failed when the route
 246 took sharp turns, because the minimization algorithm converged on a local minimum.
 247 Therefore, another method, which is a simple iterative algorithm of successive
 248 approximations, was developed and it proved to work reliably. The algorithm starts with
 249 computing the great circle distance r_0 between the starting endpoint ($d = 0$) and the searched
 250 point p , using e.g. the haversine formula, and finding point p_1 on the route with distance $d_1 =$
 251 r_0 . Then, for each iteration i :

- 252 ▪ compute the great circle distance r_i between p_i and p ;
- 253 ▪ find two points on the route: $p_{i,left}$ with the route distance $(d_i - r_i)$ and $p_{i,right}$ with the
 254 distance $(d_i + r_i)$;
- 255 ▪ compute the great circle distance of each point from p , obtaining $r_{i,left}$ and $r_{i,right}$;
- 256 ▪ if $r_{i,left} \leq r_{i,right}$ then set $d_{i+1} = d_{i,left}$ and $r_{i+1} = r_{i,left}$, otherwise set $d_{i+1} = d_{i,right}$ and $r_{i+1} =$
 257 $r_{i,right}$.



258 The algorithm stops at finding the route distance d_i of a point closest to p if there is no
 259 improvement in r with respect to the previous iteration. The value of r_i is the residual error. It
 260 is also possible to perform a further optimization of the result by applying a minimization
 261 algorithm, such as Brent's method (Brent 1973), using the range $(d_i - r_i, d_i + r_i)$, where i is the
 262 final iteration that was completed, as the bounds for minimization.

263

264 **Relationship between route distance and mileage**

265 In authoritative road network databases, location of any point on the route is often expressed
 266 as a cumulative distance calculated within reference sections. This method of describing the
 267 location will be referred to as a *mileage* in this paper, even if the distances are actually
 268 expressed in kilometers, because they are marked with *milestones* (signposts), usually at full
 269 kilometers or miles. The mileage is related to the route distance, but it is not guaranteed that
 270 the mileage is continuous along the route. Rules for the mileage calculation may vary by
 271 country (GDDKiA 2012). In order to combine the authoritative database with data retrieved
 272 from the OSM, a method of conversion between mileage, route distance and geographic
 273 coordinates is proposed. Geographical coordinates of milestones are retrieved from the OSM
 274 database, in which milestones are represented with nodes having the *highway=milestone* tag,
 275 and the *ref* tag describes the route that the milestone belongs to. Availability of milestones
 276 data in a given region depends on the community effort, competitions are often made to
 277 obtain complete data (Osmapa.pl 2018). In the proposed algorithm, geographic coordinates of
 278 each milestone are converted to the route distance d , using the procedure described earlier.
 279 The milestones are then ordered by d , forming pairs (d_i, m_i) , where m_i is the mileage indicated
 280 by a milestone. At this point, any mileage m may be converted to the route distance d as:

$$281 \quad d = d_p + (m - m_p), \quad \text{where } m_p = \max_i m_i | m_i < m \quad (19)$$



282 and d_p is the route distance corresponding to mileage m_p . It is assumed that d and m are
 283 expressed in the same physical units (usually kilometers or miles). Also, a reverse conversion
 284 of route distance to mileage may be performed using the formula:

$$285 \quad m = m_p + (d - d_p), \quad \text{where } d_p = \max_i d_i \mid d_i < d. \quad (20)$$

286 Mileage may also be converted to/from geographic coordinates using the route distance as an
 287 intermediate result. This way, it is possible to calculate latitude and longitude of a point that is
 288 represented only with the mileage in the official database, and also to estimate the mileage for
 289 any route point described with its geographic coordinates, e.g. marked on a digital map.

290

291 ANALYZING THE ROUTE GEOMETRY

292 Road geometry is one of the important factors for determining road safety. For example, sharp
 293 bends often have lowered speed limits. Route geometry data is usually not be present in the
 294 authoritative databases. However, once the route, consisting of nodes with known geographic
 295 positions, is established using the algorithm described earlier, it is possible to analyze its
 296 geometry. A discrete function $\theta(d)$, describing bearing changes along the route, was computed
 297 during the previous analysis stages. By analyzing this function and its derivative, it is possible
 298 to identify sections where the route changes its direction. For example, large steps in the
 299 bearing function and large peaks in its derivative indicate sharp turns (e.g. on the crossroads),
 300 while linearly increasing or decreasing segments indicate smooth road bends. The slope of
 301 such segments may be an indicator of the road curvature.

302 Mathematically, road curvature may be defined as a radius of a circle on a perimeter of
 303 which the road bend is located. The proposed method of finding the curvature radius r_c is
 304 based on a circle fitting method. First, geographic coordinates $p_i = (\varphi_i, \lambda_i)$ of nodes on the
 305 bend section are converted to Cartesian coordinates $q_i = (x_i, y_i)$, using the Universal
 306 Transverse Mercator (UTM) projection (Snyder 1987), where x and y (called easting and

307 northing, respectively) are expressed in physical units, e.g. meters. It is assumed here that all
 308 points are located in the same UTM zone. It is also convenient to normalize all q values so
 309 that the first point on the bend is situated in the origin, i.e. $q_1 = (0, 0)$.

310 The procedure of finding the radius r_c of a circle that fits to the points is realized with a
 311 method of a least-squares circle fit, as described in (Bullock 2006). First, a mean of all N
 312 analyzed points is computed:

$$313 \quad x_m = \frac{1}{N} \sum_{i=1}^N x_i, \quad y_m = \frac{1}{N} \sum_{i=1}^N y_i. \quad (21)$$

314 Next, points (x, y) are normalized to (u, v) by subtracting the mean:

$$315 \quad u_i = x_i - x_m, \quad v_i = y_i - y_m, \quad i = 1, \dots, N. \quad (22)$$

316 The following sums are computed:

$$317 \quad \begin{aligned} S_{uv} &= \sum_{i=1}^N u_i v_i, & S_{uu} &= \sum_{i=1}^N u_i^2, & S_{vv} &= \sum_{i=1}^N v_i^2, \\ S_{uuv} &= \sum_{i=1}^N u_i^2 v_i, & S_{uvv} &= \sum_{i=1}^N u_i v_i^2, & & \\ S_{uuu} &= \sum_{i=1}^N u_i^3, & S_{vvv} &= \sum_{i=1}^N v_i^3 & & \end{aligned} \quad (23)$$

318 The center (u_c, v_c) of a circle fitted to points (u_i, v_i) satisfies the linear equation:

$$319 \quad \begin{bmatrix} S_{uu} & S_{uv} \\ S_{uv} & S_{vv} \end{bmatrix} \cdot \begin{bmatrix} u_c \\ v_c \end{bmatrix} = \frac{1}{2} \cdot \begin{bmatrix} S_{uuu} + S_{uuv} \\ S_{uuv} + S_{vvv} \end{bmatrix}. \quad (24)$$

320 Solving this equation for (u_c, v_c) allows for computing the center point (x_c, y_c) of a circle that
 321 fits to points (x_i, y_i) :

$$322 \quad x_c = x_m + u_c, \quad y_c = y_m + v_c. \quad (25)$$

323 Finally, the fitted circle radius, which is also the searched curvature of the node set, is:

$$324 \quad r_c = \sqrt{u_c^2 + v_c^2 + \frac{S_{uu} + S_{vv}}{2}}, \quad (26)$$

325 with the residual error equal to:

$$e = \frac{1}{N} \sum_{i=1}^N \left(\sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} - r_c \right)^2. \quad (27)$$

In order to reduce the residual error, only nodes that form the road bend should be used in the calculation, nodes forming straight lines (approach to the bend or exit from it) should not be taken into consideration. Also, the accuracy depends on the number of points, so the procedure works best for longer bends, with a larger number of nodes. As it will be shown in the experiments Section, six nodes are sufficient to obtain the correct result.

IMPLEMENTATION AND EXPERIMENTS

The complete algorithm described here was implemented in a form of scripts written in Python 3 language. Route data was retrieved from the OSM database by connecting to the OSM web service through a REST (*REpresentational State Transfer*) interface, using a special query language, *Overpass QL* (Olbricht 2015). For example, in order to obtain the relation data for Route 6 in Poland, together with all ways belonging to the relation, the following query was issued:

```
[out:json];
rel[network="pl:national"][ref=6]; out;
way(r); out body geom qt;
```

The result is returned in JSON (*JavaScript Object Notation*) format, which is then converted into a nested combination of Python lists and dictionaries. In order to construct the graph and find paths between the endpoints, the *NetworkX* library was used (Hagberg 2008). After the routes are obtained from the graph, the remaining operations are performed using scripts written by the author.

In order to validate the presented algorithm, five complete trunk roads in Poland, with numbers: 6, 20, 21, 22 and 55, were analyzed. Table 1 presents the main parameters of each route, calculated by the algorithm. Three routes (6, 20, 22) were mainly longitudinal, two



351 others (21, 55) were dominantly latitudinal. Route 6 was the one with the most frequent use of
352 dual carriageways. Both long routes (20) and short ones (21) were examined. Therefore, the
353 test set is a representative, albeit small, selection of possible routes. In each case, the
354 algorithm was able to find the path between the manually selected endpoints, which confirms
355 that the algorithm was designed properly.

356 The aim of the next experiment was to assess the accuracy of the distance conversion
357 algorithm. For each route, 1000 distance values were randomly chosen from the range of the
358 route length. These values were first converted to geographic coordinates, and then back to
359 route distances. The conversion was performed using three methods, two of them were based
360 on the haversine formula (one used the standard mean Earth radius and another one estimated
361 the Earth radius with Eq. 8), and the third method used the Vincenty's formula. The additional
362 optimization step using Brent's algorithm did not improve the results, so it is not shown here.
363 The conversion accuracy was measured in terms of an absolute difference between the initial
364 distance and the distance after both conversions. Table 2 presents the root mean squared error
365 (RMSE) values in meters. It can be seen that the error is below 0.07 meters when the
366 haversine formula with mean Earth radius was used. The Vincenty's formula provides even
367 smaller RMSE values, about 10^{-4} m for most routes, except for the shortest Route 21.
368 Estimating the Earth radius introduced errors into the calculation, so this approach was
369 rejected from the further experiments. The obtained results prove that the conversion
370 procedure works reliably for points that are situated exactly on the route.

371 The procedure for conversion of geographic coordinates to the route distance was also
372 tested on a set of points that are not necessarily situated exactly on the OSM ways. For this
373 experiment, 20 points on Route 55 were selected by manually reading their coordinates in the
374 Google Earth service (on the satellite images). These coordinates were then converted using
375 the iterative procedure described in the paper. Fig. 3 shows the obtained residual error, i.e. the



376 distance of the calculated point on the route from the initial point. It can be observed that
377 convergence is achieved very quickly, in five to eight iterations, after which the error
378 variations are negligible. The residual value is below 10 meters for all points, but while it is
379 close to zero for some points, it remains at a relatively high level (3 to 10 m) for the others. It
380 was confirmed that in each case, a point on the route closest to the initial point, was found.
381 The observed residual values are therefore not conversion errors, they are a consequence of
382 inaccuracy of point selection and the OSM data. Using the additional optimization step with
383 Brent's algorithm resulted in only a small improvement in the residual error (slightly below 1
384 m and only for points with larger error values).

385 In order to confirm that the proposed algorithm allows for combining OSM data with
386 authoritative databases by means of an accurate coordinate conversion, the next experiment
387 evaluated the conversion between the calculated route distance and the official mileage. The
388 reference data on the examined trunk roads were obtained from the *Bank Danych Drogowych*
389 authoritative database (BDD, Polish for Road Data Bank, not publicly available) that is
390 managed by General Directorate for National Roads and Highways, central authority of
391 national administration set up to manage the national roads and implementation of the state
392 budget in Poland (BDD 2018). Each object in this database is described only with the official
393 mileage, e.g. "194+660" means that the point location is 194.66 km, measured within the
394 reference sections. Geographic coordinates are not used in this database at all. In order to
395 supplement this dataset with information retrieved from OSM, the coordinate conversion is
396 required. For the purpose of this experiment, milestone positions were first retrieved from the
397 OSM database by querying for nodes that represent the actual milestones, situated within 100
398 meters around the route. An example query for Route 6 is as follows:

```
399 [out:json];  
400 rel[network="pl:national"][ref=6];
```



```
401 node[highway=milestone][ref=6](around:100);
```

```
402 out body geom qt;
```

403 The obtained geographic coordinates of the milestones were converted to route distances with
404 the proposed algorithm, and sections of continuous mileage (in which changes in route
405 distance and the mileage are consistent) were found. For each milestone, its mileage relative
406 to the first milestone in a given section was compared with the difference in route distance
407 calculated between these two points. Ideally, both values should be identical. Table 3 presents
408 the obtained RMSE values (Vincenty's algorithm was used to convert milestone positions to
409 the route distance). It can be seen that these values are relatively large, with differences up to
410 70 m, 45 m on average. Since the conversion procedure itself has already been tested and the
411 observed errors were much smaller, it may be concluded that inaccuracy of the input
412 (milestones) data is the main source of these differences. Indeed, it was observed by checking
413 the milestone data on the OSM map that some milestones were not positioned accurately, and
414 errors from even a single incorrect milestone propagate throughout the whole section.
415 Therefore, in the presented algorithm and in further experiments, the route distances are
416 calculated from the nearest milestone that precedes a given point, and not from the beginning
417 of a section, thus avoiding such large errors.

418 In the next test, the accuracy of conversion between the official mileage used in the
419 BDD database and the route distances calculated by the algorithm, was examined. This test
420 validated the algorithm in terms of consistency of the data from the combined datasets. In
421 order to perform this test, 20 points for each route, spaced as regularly as possible, were
422 selected from the BDD database. The choice of the points was made so that it was easy to
423 identify each location on the satellite images. These points represented railway crossings,
424 small rivers crossings and crossroads. The mileage of each point was read from the BDD, and
425 geographic coordinates of the points were found by manually locating each one in the Google



426 Earth service, and reading its latitude and longitude. This dataset constituted the ground truth
427 for the experiment. The coordinates were then converted to the mileage with the evaluated
428 algorithm, and the difference between the computed result and the real mileage from BDD
429 was the error value, The results for each route, computed with two methods (haversine with
430 mean Earth radius and Vincenty's), without and with the optimization step using the Brent's
431 method, are presented in Table 4. The obtained average accuracy is below 35 meters for all
432 tested routes and below 24 meters for RMSE averaged over all the five routes. The accuracy
433 of the algorithm is limited mainly by that of the milestone positions in the OSM database.
434 Differences between variants of the algorithm are small, so the simplest approach based on
435 the haversine formula only (without the optimization step), may be used without decreasing
436 the conversion accuracy.

437 The results of the experiments show that the proposed algorithm allows for combining
438 information from the authoritative database and OSM, with sufficient accuracy. An example
439 of possible application of the proposed algorithm is extracting route segments with speed
440 limits and examining reasons for imposing these limits. Speed limit data may be obtained
441 from an authoritative database or/and from OSM, depending on data availability. If OSM is
442 used for this task, the complete route has to be constructed first, which is the task of the
443 algorithm presented earlier. In the next experiment, the algorithm traversed each route built
444 from OSM data, and collected information on speed limits and their reasons from the ways
445 metadata. The results are summarized in Tables 5 and 6. Although only the overall
446 distribution is shown, the algorithm finds individual route segments with given speed limits.
447 As it may be concluded from these results, completeness of the OSM metadata varies greatly
448 between the tested routes. For routes annotated with speed limits (6, 22, 55), data from OSM
449 may be combined with the official database.



450 Unfortunately, it was not possible to obtain the reference data on road curvature for
451 the tested routes, so the ground truth for the algorithm estimating road curvature from node
452 points, was not available. Therefore, only an example of the obtained results will be
453 presented. Fig. 4 shows a section of Route 55 with two sharp corners, as well as plots of the
454 route bearing and its derivative as a function of route distance. Sharp turns are easy to identify
455 on the plots, as large steps in the bearing and large spikes in its derivative. Therefore, all
456 similar turns on the route may be found by searching for such spikes. These turns may be
457 described, similarly to real life, as a change in bearing, in the presented case (going from the
458 North): a 116 degree turn left and a 93 degree turn right.

459 Fig. 5 presents a case of two bends on the road. On the bearing plot, these bends are
460 represented with slopes, and on the derivative plot – with values that consistently deviate
461 from zero. In this case, it is more suitable to describe these bends using the curvature radius,
462 as the nodes are situated on an arc. The radius of this arc, i.e. the curvature, was calculated
463 using the proposed algorithm. Fig. 6 shows the result of a successful fitting of the route nodes
464 to a circle. For the first bend (going from the North), the radius computed from 12 nodes was
465 152.707 m, with the residual error of 0.896 m. For the second bend, the radius computed from
466 6 points was 130.460 m with the residual error of 0.724 m. It can be observed that six nodes
467 were enough to obtain a good fit. The direction of bends may be obtained from the bearing
468 function (left, then right). Similar calculations were performed for other road bends and the
469 observed residual error was below 1 m, which confirms that the proposed method works as
470 expected. The only requirement is that the nodes have to form an arch. For sharp corners,
471 where the road is more accurately approximated with linear segments than an arch, the
472 method described earlier should be used.

473

474 **CONCLUSION**

475 The algorithm proposed in this paper allows for combining road network data from two
476 sources: the authoritative database and the OSM, in order to obtain a detailed dataset, suitable
477 e.g. for computer modeling of a road network. The official source may not have sufficiently
478 complete and up-to-date information. The data obtained from the OSM supplements the
479 authoritative dataset with useful information, e.g. speed limits and road geometry, it may also
480 update obsolete data. The problem of different representations of point location (mileage vs.
481 geographical coordinates) was solved by using the milestone data from the OSM database and
482 developing appropriate conversion methods. The experiments proved that the accuracy of
483 conversion between these two systems is satisfactory. As a result, any point in the
484 authoritative database may be easily located on the map, and also road information for a
485 specified geographic location may be obtained from this database. Additionally, the OSM
486 data proved to be useful in examining the route geometry, e.g. for finding road sections with
487 bends and computing its curvature. Metadata from the OSM database provide important
488 information on road conditions, such as speed limits. As a conclusion, merging information
489 from both sources provides a more detailed dataset, which may be used e.g. in computer
490 simulations of the road network, than the authoritative database alone. Some possible
491 enhancements to the road network dataset were not explored in the paper. For example,
492 information on points of interest in the vicinity of the route may also be obtained from the
493 OSM database. The algorithm also neglected elevation data which is not available from the
494 OSM, but may be acquired from other sources, such as the SRTM dataset (Farr 2007). These
495 issues are left for the future research.

496 Coverage of geographic data in OSM for Poland proved to be significantly better than
497 in any popular commercial mapping service (especially in rural areas) and it still improves
498 over time. The OSM also dominates over the commercial services in terms of rate of updating
499 data with changes. It is also inevitable that some errors are introduced by the amateur editors.



500 However, due to the collaborative character of the OSM system, each user is also the editor,
 501 able to correct errors directly in the database, which is not possible in commercial systems.
 502 Therefore, the open nature of the OSM is both its weakness and its strength at the same time.
 503 The proposed algorithm may also have an originally unintended side effect of functioning as a
 504 validation tool for ensuring completeness of relations in the OSM database.

505

506 **ACKNOWLEDGEMENTS**

507 Research was subsidized by the Polish National Centre for Research and Development and
 508 the General Directorate of Public Roads and Motorways within the grant No. OT4- 4B/AGH-
 509 PG-WSTKT. Map data copyrighted OpenStreetMap contributors and available from
 510 <https://www.openstreetmap.org>.

511

512 **NOTATION**

513 *The following symbols are used in this paper:*

514 c = connector, a node that connects two or more ways

515 d = route distance of a point, measured from the origin along the route (“driving distance”)

516 m = mileage, an official distance of a route point measured in reference sections

517 n = node in the OSM data, a single point described by its latitude and longitude

518 p = a point (φ, λ)

519 q = a point (x, y) in the Cartesian coordinate system

520 r = distance between two geographical locations, measured on the great circle

521 R = relation in the OSM data, an unordered collection of ways

522 s = route, an ordered sequence of connected ways

523 w = way in the OSM data, an ordered sequence of nodes

524 λ = longitude



525 φ = latitude

526 θ = bearing between two geographical locations

527

528 REFERENCES

529 Alarabi, L., Eldawy, A., Alghamdi, R., and Mokbel, M. F. (2014). “TAREEG: a MapReduce-
530 based system for extracting spatial data from OpenStreetMap.” *Proc. 22nd ACM*

531 *SIGSPATIAL Int. Conf. Advances in Geographic Information Systems*, ACM, New York, 83–
532 92.

533 Ballatore, A., Bertolotto, M., and Wilson, D. C. (2013). “Geographic knowledge extraction
534 and semantic similarity in OpenStreetMap.” *Knowledge & Information Systems*, 37(1), 61–81.

535 Barron, C., Neis, P., and Zipf, A. (2014). “A comprehensive framework for intrinsic
536 OpenStreetMap quality analysis.” *Trans. in GIS*, 18(6), 877–895.

537 *BDD – Bank Danych Drogowych* [Road Bank Database] (2018).

538 <<https://www.gddkia.gov.pl/pl/995/bank-danych-drogowych>> (In Polish, Apr. 20, 2018).

539 Brent, R. P. (1973). “An algorithm with guaranteed convergence for finding a zero of a
540 function.” *Algorithms for minimization without derivatives, Chapter 4*. Prentice-Hall,

541 Englewood Cliffs, NJ.

542 Brovelli, M. A., Minghini, M., Molinari, M., and Mooney, P. (2017). “Towards an automated
543 comparison of OpenStreetMap with authoritative road datasets.” *Trans. in GIS*, 21(2), 191–
544 206.

545 Bullock, R. (2006). “Least-Squares Circle Fit.”

546 <http://www.dtcenter.org/met/users/docs/write_ups/circle_fit.pdf> (Apr. 20, 2018).

547 Chen, B., Sun, W., and Vodacek, A. (2014). “Improving image-based characterization of road
548 junctions, widths, and connectivity by leveraging OpenStreetMap vector map.” 2014 IEEE



- 549 Int. Geoscience and Remote Sensing Symp. (IGARSS), IEEE, Piscataway, doi:
550 10.1109/IGARSS.2014.6947608.
- 551 Ciepluch, B., Jacob, R., Mooney, P., and Winstanley, A.C. (2010). “Comparison of the
552 accuracy of OpenStreetMap for Ireland with Google Maps and Bing Maps.” *Proc., 9th Int.*
553 *Symp. Spatial Accuracy Assessment in Natural Resources and Environmental Sciences*, Univ.
554 of Leicester, Leicester, 337.
- 555 Corcorana, P., Mooney, P., and Bertolotto, M. (2013). “Analysing the growth of
556 OpenStreetMap networks.” *Spatial Statistics*, 3, 21–32.
- 557 Fan, H., Yang, B., Zipf, A., and Rousell, A. (2016). “A polygon-based approach for matching
558 OpenStreetMap road networks with regional transit authority data.” *Int. J. Geographical*
559 *Information Science*, 30(4), 748–764.
- 560 Farr, T. G., Rosen, P. A., Caro, E., Crippen, R., Duren, R., Hensley, S., Kobrick, M., Paller,
561 M., Rodriguez, E., Roth, L., Shaffer, S., Shimada, J., Umlaud, J., Werner, M., Oskin, M.,
562 Burbank, D., and Alsdorf, D. (2007). “The Shuttle Radar Topography Mission.” *Reviews of*
563 *Geophysics*, 45(2), doi:10.1029/2005RG000183.
- 564 GDDKiA (General Director for National Roads and Motorways) (2012) “Instrukcja ustalania
565 i prowadzenia kilometrażu dróg. Zarządzenie nr 18. [Road mileage determination
566 instructions].” <https://www.gddkia.gov.pl/pl/1641/Rok-2012> (in Polish, Apr. 20, 2018).
- 567 Gil, J. (2015). “Building a multimodal urban network model using OpenStreetMap data for
568 the analysis of sustainable accessibility.” In: Arsanjani, J. J., Zipf, A., Mooney, P., and
569 Helbich, M. (eds.), *OpenStreetMap in GIScience, Lecture Notes in Geoinformation and*
570 *Cartography*, Springer, Cham, 229–251.
- 571 Girres, J.-F., and Touya, G. (2010). “Quality assessment of the French OpenStreetMap
572 dataset.” *Trans. in GIS*, 14(4), 435–459.



- 573 Hagberg, A. A., Schult, D. A., Swart, P. J. (2008). “Exploring network structure, dynamics,
574 and function using NetworkX.” Proc. 7th Python in Science Conference (SciPy2008),
575 Enthougt, Austin, TX, 11–15.
- 576 Haklay, M. (2010). “How good is volunteered geographical information? A comparative
577 study of OpenStreetMap and ordnance survey datasets.” *Environment and Planning B: Urban*
578 *Analytics and City Science*, 37, 682–703.
- 579 Huber, S., and Rust, C. (2016), “Osrmtime: calculate travel time and distance with
580 OpenStreetMap data using the Open Source Routing Machine (OSRM).” *The Stata Journal*,
581 16(2), 416–423.
- 582 Jilani, M., Corcoran, P., and Bertolotto, M. (2013). “Multi-granular street network
583 representation towards quality assessment of OpenStreetMap data.” *Proc., 6th ACM*
584 *SIGSPATIAL Int. Workshop on Computational Transportation Science*, ACM, New York, 19.
- 585 Jilani, M., Corcoran, P., and Bertolotto, M. (2014). “Automated highway tag assessment of
586 OpenStreetMap road networks.” *Proc., 22nd ACM SIGSPATIAL Int. Conf. Advances in*
587 *Geographic Information Systems*, ACM, New York, 449–452.
- 588 Li, Q., Fan, H., Luan, X., Yang, B., and Liu, L. (2014). “Polygon-based approach for
589 extracting multilane roads from OpenStreetMap urban road networks.” *Int. J. Geographical*
590 *Information Science*, 28(11), 2200–2219.
- 591 Long, Y., and Liu, X. (2016). “Automated identification and characterization of parcels
592 (AICP) with OpenStreetMap and Points of Interest.” *Environment and Planning B: Urban*
593 *Analytics and City Science*, 43(2), 341–360.
- 594 Mooney, P., Corcoran, P., and Winstanley, A. C. (2010). “Towards quality metrics for
595 OpenStreetMap.” Proc., 18th SIGSPATIAL International Conference on Advances in
596 Geographic Information Systems, ACM, New York, 514–517.



- 597 Neis, P., Zielstra, D., and Zipf, A. (2011). “The street network evolution of crowdsourced
598 maps: OpenStreetMap in Germany 2007–2011.” *Future Internet*, 4(1), 1–21.
- 599 Olbricht, R. (2015). “Data retrieval for small spatial regions in OpenStreetMap.” In:
600 Arsanjani, J. J., Zipf, A., Mooney, P. and Helbich, M. (eds.), *OpenStreetMap in GIScience*,
601 *Lecture Notes in Geoinformation and Cartography*, Springer, Cham, 101–122.
- 602 OpenStreetMap (OSM) contributors (2018). “Planet dump retrieved from
603 <https://planet.openstreetmap.org>.” <<https://www.openstreetmap.org>> (Apr. 20, 2018).
- 604 OpenStreetMap Wiki (2018). “Elements.” <<http://wiki.openstreetmap.org/wiki/Elements>>
605 (Apr. 20, 2018).
- 606 Osmapa.pl: “Pikietaż w bazie OpenStreetMap [Milestones in OpenStreetMap]” (2018).
607 <<http://osmapa.pl/konkursy/pikietaz/>> (In Polish, Apr. 20, 2018).
- 608 Over, M., Schilling, A., Neubauer, S., and Zipf, A. (2010). “Generating web-based 3D City
609 Models from OpenStreetMap: The current situation in Germany.” *Computers, Environment*
610 *and Urban Systems*, 34(6), 496–507.
- 611 Rahmig, C., and Simon, A. (2014). “Extracting topology and geometry information from
612 OpenStreetMap data for digital maps for railway applications.” *10th ITS European Congress*,
613 Ertico – ITS Europe, Brussels.
- 614 Schelhorn, S. J., Herfort, B., Leiner, R., and Zipf, A. (2014). “Identifying elements at risk
615 from OpenStreetMap: the case of flooding.” *11th Int. Conf. Information Systems for Crisis*
616 *Response and Management ISCRAM 2014*, Pennsylvania State Univ.
- 617 Schellekens, J., Broksmaa, R. J., Dahma, R. J., Donchytsa, G. V., and Winsemius, H.C.
618 (2014). “Rapid setup of hydrological and hydraulic models using OpenStreetMap and the
619 SRTM derived digital elevation model.” *Environmental Modelling & Software*, 61, 98–105.
- 620 Sedgewick, R., and Wayne, K. (2011). *Algorithms*, 4th ed. Addison-Wesley, Boston, MA.
- 621 Sinnott, R. W. (1984). “Virtues of the haversine.” *Sky and Telescope*, 68(2), 159.



- 622 Snyder, J. P.. (1987) “Map projections – a working manual.” *U.S. Geological Survey*
623 *Professional Paper 1395*. United States Government Printing Office, Washington, D.C..
- 624 Vincenty, T. (1975). “Direct and inverse solutions of geodesics on the ellipsoid with
625 application of nested equations.” *Survey Review*, 23(176), 88–93.
- 626 Zilske, M., Neumann, A., and Nagel, K. (2011). “OpenStreetMap for traffic simulation.”
627 Proc. 1st European State of the Map Conf., OpenStreetMap Foundation, Sutton Coldfield,
628 126–134.

