Combining road network data from OpenStreetMap with an authoritative database

Grzegorz Szwoch, Ph.D.
Gdansk University of Technology, Department of Multimedia Systems
80-223 Gdansk, Poland, Narutowicza 11/12
grzegorz.szwoch@pg.edu.pl

Abstract

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

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

INTRODUCTION

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

Utilizing separate data sources brings the problem of combining information from distinct databases. The main contribution of this paper is proposing a solution to two most important problems related to combining the OSM data with an authoritative database. The first one is related to the fact that OSM data describes only short road sections and there is no simple way to obtain ordered data on a complete trunk road, including separate lanes. This problem is addressed by a proposed algorithm that constructs a continuous route from partial OSM data. The second, more difficult problem is how to relate data from both sources with each other, so that any location in one database may be found in another one. For example, an authoritative database may identify road elements only by their reference position within the road, while the OSM data is based solely on geographic coordinates. The proposed algorithm provides a method of conversion between these coordinates. As a result, a combined data source, suitable e.g. for computer modeling of road network, is obtained.

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

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

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

CONSTRUCTING A ROUTE FROM THE OSM DATA

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

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

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

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

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

$$w_{i,j} = \langle n_i, n_{k,1}, n_{k,2}, \ldots, n_j \rangle \qquad (2)$$

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

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

$$R_k = \{ w_{k,1}, w_{k,2}, \ldots, w_{k,N} \}. \qquad (3)$$

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

$$s_k = \langle w_{k,1}, w_{k,2}, \ldots, w_{k,N} \rangle, \; w_k \in R_k . \qquad (4)$$

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

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

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

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

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

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

ESTABLISHING A RELATION BETWEEN OSM DATA AND THE AUTHORITATIVE DATASET

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

Calculating route distances

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

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

where:

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

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

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

where r_1 = 6378137 m, r_2 = 6356752 m.

The haversine formula may introduce errors up to 0.3%. A more accurate method of computing the distance is based on an inverse solution to the Vincenty's formula (Vincenty 1975), providing accuracy up to 0.5 mm, at the cost of increased computation time. Both methods are used in the proposed algorithm Other methods are also possible, such as projection of points to the Cartesian coordinate system, e.g. the Universal Transverse Mercator (UTM) (Snyder 1987) and computing an Euclidean distance between the points.

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

$$d_i^{'} = d_i \frac{d_f + d_b}{2d_f}. \qquad (9)$$

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

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

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

where *A* is the polygon area:

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

The latitude and longitude values are averaged directly, which is inaccurate, but sufficient for this particular purpose, because the error is not large due to proximity of the nodes, and determining the exact centroid of a roundabout is not essential here. The centroid replaces all nodes in a roundabout and is used in the distance calculation.

Calculating geographic coordinates given route distance

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

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

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

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

$$\theta = \mathrm{arctan2}\left(\sin\left(\lambda_2 - \lambda_1\right) \cdot \cos\varphi_2,\ \cos\varphi_1 \cdot \sin\varphi_2 - \sin\varphi_1 \cdot \cos\varphi_2 \cdot \cos\left(\lambda_2 - \lambda_1\right)\right) \qquad (15)$$

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

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

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

$$\lambda_p = \lambda_1 + \operatorname{arctan2}\left(\sin\theta \cdot \cos\varphi_1 \cdot \sin\frac{d-d_1}{r_E}, \cos\frac{d-d_1}{r_E} - \sin\varphi_1 \cdot \sin\varphi_p\right) \qquad (17)$$

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

Calculating route distance given geographic coordinates

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

$$d = \min_d\left(\left|p - p_d\right|\right) \qquad (18)$$

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

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

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

Relationship between route distance and mileage

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

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

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

$$m = m_p + \left(d - d_p\right), \quad \text{where } d_p = \max_i d_i \middle| d_i < d. \qquad (20)$$

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

ANALYZING THE ROUTE GEOMETRY

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

Mathematically, road curvature may be defined as a radius of a circle on a perimeter of which the road bend is located. The proposed method of finding the curvature radius r_c is based on a circle fitting method. First, geographic coordinates $p_i = (\varphi_i, \lambda_i)$ of nodes on the bend section are converted to Cartesian coordinates $q_i = (x_i, y_i)$, using the Universal Transverse Mercator (UTM) projection (Snyder 1987), where x and y (called easting and northing, respectively) are expressed in physical units, e.g. meters. It is assumed here that all points are located in the same UTM zone. It is also convenient to normalize all q values so that the first point on the bend is situated in the origin, i.e. $q_1 = (0, 0)$.

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

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

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

$$u_i = x_i - x_m, \; v_i = y_i - y_i, \; i = 1,\ldots,N \, . \; (22)$$

The following sums are computed:

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

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

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

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

$$x_c = x_m + u_c, \; y_c = y_m + v_c \, . \qquad (25)$$

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

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

with the residual error equal to:

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

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

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

The procedure for conversion of geographic coordinates to the route distance was also tested on a set of points that are not necessarily situated exactly on the OSM ways. For this experiment, 20 points on Route 55 were selected by manually reading their coordinates in the Google Earth service (on the satellite images). These coordinates were then converted using the iterative procedure described in the paper. Fig. 3 shows the obtained residual error, i.e. the distance of the calculated point on the route from the initial point. It can be observed that convergence is achieved very quickly, in five to eight iterations, after which the error variations are negligible. The residual value is below 10 meters for all points, but while it is close to zero for some points, it remains at a relatively high level (3 to 10 m) for the others. It was confirmed that in each case, a point on the route closest to the initial point, was found. The observed residual values are therefore not conversion errors, they are a consequence of inaccuracy of point selection and the OSM data. Using the additional optimization step with Brent's algorithm resulted in only a small improvement in the residual error (slightly below 1 m and only for points with larger error values).

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

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

```
node[highway=milestone][ref=6](around:100);
out body geom qt;
```

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

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

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

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

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

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

CONCLUSION

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

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

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

ACKNOWLEDGEMENTS

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

NOTATION

The following symbols are used in this paper:

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

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

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

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

p = a point (φ, λ)

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

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

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

s = route, an ordered sequence of connected ways

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

λ = longitude

φ = latitude

θ = bearing between two geographical locations

REFERENCES

Alarabi, L., Eldawy, A., Alghamdi, R., and Mokbel, M. F. (2014). “TAREEG: a MapReduce-based system for extracting spatial data from OpenStreetMap.” *Proc. 22nd ACM SIGSPATIAL Int. Conf. Advances in Geographic Information Systems*, ACM, New York, 83–92.

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

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

BDD – Bank Danych Drogowych [Road Bank Database] (2018). <https://www.gddkia.gov.pl/pl/995/bank-danych-drogowych> (In Polish, Apr. 20, 2018).

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

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

Bullock, R. (2006). “Least-Squares Circle Fit.” <http://www.dtcenter.org/met/users/docs/write_ups/circle_fit.pdf> (Apr. 20, 2018).

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

Int. Geoscience and Remote Sensing Symp. (IGARSS), IEEE, Piscataway, doi: 10.1109/IGARSS.2014.6947608.

Ciepluch, B., Jacob, R., Mooney, P., and Winstanley, A.C. (2010). “Comparison of the accuracy of OpenStreetMap for Ireland with Google Maps and Bing Maps.” *Proc., 9th Int. Symp. Spatial Accuracy Assessment in Natural Resuorces and Enviromental Sciences*, Univ. of Leicester, Leicester, 337.

Corcorana, P., Mooneyb, P., and Bertolottoa, M. (2013). “Analysing the growth of OpenStreetMap networks.” *Spatial Statistics*, 3, 21–32.

Fan, H., Yang, B., Zipf,, A., and Rousell, A. (2016). “A polygon-based approach for matching OpenStreetMap road networks with regional transit authority data.” *Int. J. Geographical Information Science*, 30(4), 748–764.

Farr, T. G., Rosen, P. A., Caro, E., Crippen, R., Duren, R., Hensley, S., Kobrick, M., Paller, M., Rodriguez, E., Roth, L., Shaffer, S., Shimada, J., Umlaud, J., Werner, M., Oskin, M., Burbank, D., and Alsdorf, D. (2007). “The Shuttle Radar Topography Mission.” *Reviews of Geophysics*, 45(2), doi:10.1029/2005RG000183.

GDDKiA (General Director for National Roads and Motorways) (2012) “Instrukcja ustalania i prowadzenia kilometrażu dróg. Zarządzenie nr 18. [Road mileage determination instructions].” https://www.gddkia.gov.pl/pl/1641/Rok-2012 (in Polish, Apr. 20, 2018).

Gil, J. (2015). “Building a multimodal urban network model using OpenStreetMap data for the analysis of sustainable accessibility.” In: Arsanjani, J. J., Zipf, A., Mooney, P., and Helbich, M. (eds.), *OpenStreetMap in GIScience, Lecture Notes in Geoinformation and Cartography*, Springer, Cham, 229–251.

Girres, J.-F., and Touya, G. (2010). “Quality assessment of the French OpenStreetMap dataset.” *Trans. in GIS*, 14(4), 435–459.

Hagberg, A. A., Schult, D. A., Swart, P. J. (2008). “Exploring network structure, dynamics, and function using NetworkX.” Proc. 7th Python in Science Conference (SciPy2008), Enthought, Austin, TX, 11–15.

Haklay, M. (2010). “How good is volunteered geographical information? A comparative study of OpenStreetMap and ordnance survey datasets.” *Environment and Planning B: Urban Analytics and City Science*, 37, 682–703.

Huber, S., and Rust, C. (2016), “Osrmtime: calculate travel time and distance with OpenStreetMap data using the Open Source Routing Machine (OSRM).” *The Stata Journal*, 16(2), 416–423.

Jilani, M., Corcoran, P., and Bertolotto, M. (2013). “Multi-granular street network representation towards quality assessment of OpenStreetMap data.” *Proc., 6th ACM SIGSPATIAL Int. Workshop on Computational Transportation Science*, ACM, New York, 19.

Jilani, M., Corcoran, P., and Bertolotto, M. (2014). “Automated highway tag assessment of OpenStreetMap road networks.” *Proc., 22nd ACM SIGSPATIAL Int. Conf. Advances in Geographic Information Systems*, ACM, New York, 449–452.

Li, Q., Fan, H., Luan, X., Yang, B., and Liu, L. (2014). “Polygon-based approach for extracting multilane roads from OpenStreetMap urban road networks.” *Int. J. Geographical Information Science*, 28(11), 2200–2219.

Long, Y., and Liu, X. (2016). “Automated identification and characterization of parcels (AICP) with OpenStreetMap and Points of Interest.” *Environment and Planning B: Urban Analytics and City Science*, 43(2), 341–360.

Mooney, P., Corcoran, P., and Winstanley, A. C. (2010). “Towards quality metrics for OpenStreetMap.” Proc., 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM, New York, 514–517.

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

Olbricht, R. (2015). “Data retrieval for small spatial regions in OpenStreetMap.” In: Arsanjani, J. J., Zipf, A., Mooney, P. and Helbich, M. (eds.), *OpenStreetMap in GIScience*, *Lecture Notes in Geoinformation and Cartography*, Springer, Cham, 101–122.

OpenStreetMap (OSM) contributors (2018). “Planet dump retrieved from https://planet.openstreetmap.org.” <https://www.openstreetmap.org> (Apr. 20, 2018).

OpenStretMap Wiki (2018). “Elements.” <http://wiki.openstreetmap.org/wiki/Elements> (Apr. 20, 2018).

Osmapa.pl: “Pikietaż w bazie OpenStreetMap [Milestones in OpenStreetMap]” (2018). <http://osmapa.pl/konkursy/pikietaz/> (In Polish, Apr. 20, 2018).

Over, M., Schilling, A., Neubauer, S., and Zipf, A. (2010). “Generating web-based 3D City Models from OpenStreetMap: The current situation in Germany.” *Computers, Environment and Urban Systems*, 34(6), 496–507.

Rahmig, C., and Simon, A. (2014). “Extracting topology and geometry information from OpenStreetMap data for digital maps for railway applications.” *10th ITS European Congress*, Ertico – ITS Europe, Brussels.

Schelhorn, S. J., Herfort, B., Leiner, R., and Zipf, A. (2014). “Identifying elements at risk from OpenStreetMap: the case of flooding.” *11th Int. Conf. Information Systems for Crisis Response and Management ISCRAM 2014*, Pennsylvania State Univ.

Schellekensa, J., Brolsmaa, R. J., Dahma, R. J., Donchytsa, G. V., and Winsemiusa, H.C. (2014). “Rapid setup of hydrological and hydraulic models using OpenStreetMap and the SRTM derived digital elevation model.” *Environmental Modelling & Software*, 61, 98–105.

Sedgewick, R., and Wayne, K. (2011). *Algorithms, 4th ed.* Addison-Wesley, Boston, MA.

Sinnott, R. W. (1984). “Virtues of the haversine.” *Sky and Telescope*, 68(2), 159.

Snyder, J. P.. (1987) “Map projections – a working manual.” *U.S. Geological Survey Professional Paper 1395*. United States Government Printing Office, Washington, D.C..

Vincenty, T. (1975). “Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations.” *Survey Review*, 23(176), 88–93.

Zilske, M., Neumann, A., and Nagel, K. (2011). “OpenStreetMap for traffic simulation.” Proc. 1st European State of the Map Conf.,, OpenStreetMap Foundation, Sutton Coldfield, 126–134.