RESERACH ARTICLE

There is no way! Ternary qualitative spatial reasoning for error detection in map data

Ivan Majic[1] | Elham Naghizade[1*] | Stephan Winter[1] | Martin Tomko[1]

[1]Department of Infrastructure Engineering, The University of Melbourne, Parkville 3010 VIC, Australia

Correspondence
Ivan Majic, Department of Infrastructure Engineering, The University of Melbourne, Parkville 3010 VIC, Australia
Email: imajic@student.unimelb.edu.au

Present address
*School of Science, Computer Science and IT, RMIT University, Melbourne 3000 VIC, Australia

Funding information
Australian Research Council (DP170100153 and DP170100109)

Detection and correction of errors in map data based on spatial reasoning may be used to improve their quality. However, the majority of current spatial reasoning approaches are based on binary spatial relations and are not able to perform analyses involving more than two objects. This paper proposes building accessibility analysis with the ternary Ray Intersection Model (RIM) to detect potential map errors. Where buildings are not accessible from the road network, this may indicate potential errors in map data such as roads that are not mapped. The plausibility of the proposed method was tested in a case study on OpenStreetMap data. The results have been published in an online mapping challenge where volunteering mappers have used them to correct errors in map data, and have provided feedback on the analysis. The results show that the proposed method can detect errors in map data that are caused by incorrect classification of buildings, incorrect mapping of multi-part buildings, and missing road data.

KEYWORDS
building accessibility, visibility analysis, ternary spatial relations, spatial reasoning, Ray Intersection Model (RIM), OpenStreetMap

1 | INTRODUCTION

One of the most common spatial computation tasks is path-planning, also known as routing. From finding the way to the nearest Italian restaurant to planning ambulance dispatch, path-planning uses spatial information to determine the preferred route based on the selected criteria (e.g., shortest, fastest, or the simplest path). Digital maps and systems that rely on digital maps know about the physical world only what is captured by the mappers in map data. Thus, if map data are incomplete, outdated, or otherwise incorrect, this may negatively impact analysis results (e.g., Zhang and Ai (2015)), here the planned paths.

The particular path-planning problem that motivates this study is that destinations (i.e., usually buildings) are often snapped to the nearest point on a route segment for path-planning. The routing algorithm (e.g., Dijkstra) then uses this snapped point as a destination in its calculations. If road segments around the destination building are missing from the map, then the destination building will be snapped to a road that is further away and that perhaps cannot provide access to it, which will result in an incorrect route.

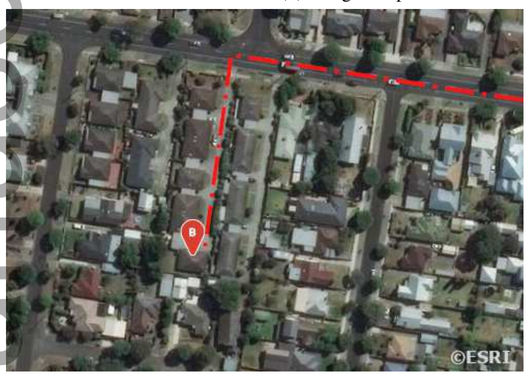
For example, Figure 1 shows that if a user searches for directions to the building at 7/147 Roslyn Road, Belmont, Victoria, Australia, the resulting paths planned based on OpenStreetMap (OSM) data (Figure 1a) and those using Google Maps (Figure 1b) will be incorrect. The proposed route will lead the user to the closest point on the road network to this building. However, the user is unable to access the building from the planned path's endpoint as private properties block access. The satellite image of the area, however, shows that the intended destination is accessible via a driveway off the Roslyn Road (Figure 1c). Although Google has recently partially fixed this issue by forcing the proposed route to end on the street that shares the same address as the destination building (i.e., Roslyn Road), the problem shown in Figure 1b still remains if the building is selected via the right-click and "Directions to here" option. This case, where a driveway is missing from the incomplete map data, leads to an approximately 400 meters detour. The misguided user would spend additional time backtracking to access the building. In the case of an emergency (Nicoară & Haidu, 2014), such delay might have critical consequences.

In the example, and in most implementations of routing algorithms (i.e., Google Maps, OpenRouteService.org, and algorithms in (Dalumpines & Scott, 2011) and (Graser, Straub, & Dragaschnig, 2015)), the start and destination points of a route are snapped to the closest road segment. When the destination is a building, this is based on the assumption that the road closest to the building provides access and that all buildings should be accessible from the road network. If the question is whether other buildings are in the way, we can take *visibility* as a proxy for *accessibility*, and then hypothesize that buildings with no access to the closest road can be identified by a 2D *visibility* analysis between buildings and the closest roads. Following the definition that "two points are mutually visible if the line segment connecting them is unoccluded" (Bittner & Wonka, 2003), this study considers a building to be accessible if there exists a direct unoccluded line of sight between it and the closest road in a 2D map dataset of buildings and roads. While there may be other objects that affect visibility and access to buildings such as walls, gates, fences, trees, and terrain, the scope of this study is limited to relations between buildings and roads. Other objects may be supported in the visibility analysis with a more nuanced version of the proposed approach.

This paper proposes a modified version of the Ray Intersection Model (RIM) (Majic, Naghizade, Winter, & Tomko, 2020), originally developed for the analysis of betweenness of spatial objects, to determine the accessibility of buildings. This method analyses rays between the building being analyzed and its closest road (i.e., a ray is a straight line between two objects that shares exactly one end point with each of them). The proposed method then tests if rays are intercepted by other objects. This can be further modeled as a binary classification task, where buildings are classified as either *accessible* if an unintercepted ray exists between the building and its closest road or *inaccessible* if it does not. The proposed approach for building accessibility analysis has been tested in a case study on the OSM dataset for the state of Victoria, Australia. Buildings that are completely occluded (i.e., all rays between the building and the road are intercepted) by other objects and are therefore considered *not visible* from their closest roads are classified as inaccessible. The factors that cause a building to be inaccessible can be categorised as: (1) buildings without the need for direct access (e.g., sheds); (2) buildings without designated road access (e.g., pedestrian zone and school complex buildings); (3) buildings where the access path is not a straight line; (4) data quality issues. In this paper we hypothesise that inaccessible buildings indicate potential errors in the map data. After the inaccessible buildings have been detected, they were presented via an online mapping challenge on maproulette.org, where mappers assessed the situation and corrected any potential errors in the map data. The outcomes of this challenge have been used to evaluate the performance of the

(a) OpenStreetMap directions.

(b) Google Maps directions.



(c) ESRI satellite imagery showing the real access to the building (dashed line).

FIGURE 1 Incorrect routing directions from Geelong, Victoria to the building at *7/147 Roslyn Rd, Belmont, Victoria, Australia* given by OSM and Google Maps.

proposed method in the case study and its plausibility.

This paper makes the following contributions: (1) it presents a computational refinement of the RIM to determine the occlusion between two spatial objects in a 2D plane and a set of potentially occluding objects; (2) it demonstrates a novel problem of building accessibility detection from the road network, where it provides qualitative spatial reasoning on ternary relationships; and (3) it demonstrates an approach for a large scale analysis of building accessibility in a way that enables automated detection of potential errors such as missing road data.

The remainder of this paper is structured as follows: Section 2 reviews the related work in the fields of accessibility, visibility analysis, and OSM data quality. Section 3 presents the four steps of the building accessibility analysis and shows an algorithmic approach to determine whether a visibility ray exists between the building and the closest road. Section 4 demonstrates the case study where inaccessible buildings in Victoria, Australia were detected as potential errors and assessed by mappers. Section 5 discusses the results of the building accessibility analysis and the human feedback on the inaccessible buildings as potential errors. Section 6 gives concluding remarks and plans for future work.

2 | RELATED WORK

2.1 | Accessibility

Most of the studies on accessibility documented in the literature are concerned either with how accessible the buildings are for people with disability (Alonso, 2002; Balado, Díaz-Vilariño, Arias, & Soilán, 2017), or with the ease of access to a building (Sakkas & Pérez, 2006), a location (Liu & Zhu, 2004; AlKahtani, Xia, Veenendaaland, Caulfield, & Hughes, 2015), or a service (e.g., such as transport (Ford, Barr, Dawson, & James, 2015), health care (Luo & Wang, 2003), or emergency services (Tansley, Schuurman, Amram, & Yanchar, 2015)). Similar to our study, the researchers in the latter group also see the road network as provider of access and they usually use travel time (Luo & Wang, 2003) or travel distance (Tansley et al., 2015) to assess the accessibility of a location. In this paper we focus on whether a building is accessible from the road network without there being an obstacle between, with the aim of detecting errors in the map database. The above mentioned studies use this database as a framework for routing and would benefit from the improved data quality. Other approaches to accessibility include statistical assessment of accessibility (e.g., walkability, variety of destinations). These are of not considered here. We focus on physical accessibility to the premises, and do not nuance by the mobility constraints of the agents (e.g., pedestrians, cyclists) which can be simply modelled by considering a different street network.

2.2 | Visibility and occlusion analysis

2.2.1 | Qualitative spatial reasoning

There have been several studies in qualitative spatial reasoning (QSR) on the visibility and occlusion of spatial objects. Galton (1994) was the first to model visibility with *line-of-sight* relations (LOS-14) inspired by the region connection calculus (RCC) (D. A. Randell, Cui, & Cohn, 1992). LOS-14 consists of 14 qualitatively distinct configurations of two convex spatial objects as seen by the observer in their field of view (e.g., A hides B or A is in front of B). LOS-14 was also used in (D. Randell & Witkowski, 2006) to interpret relations between objects in digital image. It was superseded by Köhler's occlusion calculus (OCC) (Köhler, 2002) which is slightly different in its calculation and in the composition table used for qualitative reasoning with these relations (e.g., if A partially hides B what are the possible occlusion relations from B to A?). D. Randell, Witkowski, and Shanahan (2001) have introduced the region occlusion calculus (ROC-20) which has expanded the LOS-14 by also allowing concave objects in their model. They have also studied the motion parallax and how does the occlusion change when the objects in the scene are moving, and when the viewpoint of the scene is changed. Some interesting applications of ROC-20 include formalization and exploration of shadows (Santos, Dee, & Fenelon, 2008) and celestial eclipses (Santos, Casati, Dee, Schultz, & Bhatt, 2016).

The next improvement was introduced by Tarquini, De Felice, Fogliaroni, and Clementini (2007) in their "qualitative model for visibility relations" that has enabled the "point of view" to be not only a point, but any convex geometry. They have achieved this by modeling the visibility as a ternary spatial relation based on the ternary projective relations model (Billen & Clementini, 2005a, 2005b; Clementini & Billen, 2006). Their method divides the space based on the projective lines between the point of view object C and the obstacle object B into three types of acceptance areas: a shadow zone, a twilight zone, and a light zone. The third object A is then assigned any of the seven distinct visibility relations (e.g., visible, partially visible or occluded) based on its placement within these acceptance zones. This model has also been applied to localization and navigation of autonomous agent with limited sensor abilities (Fogliaroni, Wallgrün, Clementini, Tarquini, & Wolter, 2009). In this application the autonomous agent in each position analyses the obstacle objects in its surroundings with the novel approach of visual cyclic ordering. It then creates a topological map of visible and invisible areas using the visibility relations, and uses this map for qualitative navigation.

Guha, Mukerjee, and Venkatesh (2011) have introduced their model that differentiates fourteen occlusion states (OCS-14) in which they additionally consider whether the visible parts in the scene are connected behind the occluder object or not, and whether the occluder object is static or moving, claiming that these distinctions are important for computer vision applications. A more recent Interval Occlusion Calculus (IOC) (Ligozat & Santos, 2015; Santos, Ligozat, & Safi-Samghabad, 2015) takes into consideration multiple distinct viewpoints of a scene. This enables multiple agents to explore the environment simultaneously. A more detailed overview of visibility relations in spatial cognition can be found in (Fogliaroni, 2015), while (Dylla et al., 2017) places these visibility models inside the larger scope of qualitative spatial and temporal calculi.

The method proposed in this paper uses rays cast between a line object (a road) and a polygon object (a building), which is similar to lines of sight used in some of the approaches mentioned above. However, what sets the proposed method apart is that it does not classify the entire scene into visible and occluded or light and shadow parts. Instead, it only searches for the existence of at least one line of sight between two designated objects. While this does not provide a comprehensive visibility analysis of the scene, it reduces the complexity of the algorithm. This can potentially enable faster computation for binary visibility analysis (i.e., there is / is no visibility between two objects), but the comprehensive comparison of approaches' computing speeds is out of the scope of this paper.

2.2.2 | Computer graphics

In contrast to QSR, visibility analysis research in computer graphics revolves around quantitative approaches where the focus is on the efficiency and the speed of algorithms. Multiple approaches for building visibility graphs (Welzl, 1985; de Berg, Cheong, van Kreveld, & Overmars, 2008) and visibility complexes (Pocchiola & Vegter, 1996; Rivière, 1997) have been developed over the years, always further optimizing the computation. Similar to some of the papers from QSR, the goal is usually to analyze the entire scene and determine the visible and invisible parts (Kasapakis & Gavalas, 2015), or light and shaded parts if the application is for lighting effects (Heidrich, Brabec, & Seidel, 2000).

A thorough overview of the visibility related research in computer graphics has been presented in (Bittner & Wonka, 2003). It presents a taxonomy of visibility problems according to the problem domain. This taxonomy differentiates five visibility problems: visibility along a line, visibility from a point, visibility from a line segment, visibility from a polygon, visibility from a region, and global visibility. In computer graphics it is the usual practice to map the problem of finding unoccluded lines of sight into the n-dimensional space called the *line space*, and the previously mentioned domains map visibility to line spaces of different dimensions. What is different about the approach proposed in this paper is that it does not need to analyze an entire scene for visible and occluded parts. It only needs to confirm the existence of a single line of sight between two objects – a road and a building. Given that these objects are usually represented as a polygon and a polyline, the visibility problem in this paper could be described as visibility from a line segment or a polygon depending on which object is designated as a source of visibility. However, our algorithm for assessing visibility (Section 3.2) uses functions and operators that are common in GIS computations, and it does not map the problem into the line space.

2.3 | OpenStreetMap data quality

There has been much work done on the quality assessment of the OpenStreetMap data. Some of the first studies have evaluated OSM by comparing it to proprietary datasets (Haklay, 2010; Girres & Touya, 2010; Zielstra & Zipf, 2010), mostly to find that the completeness and accuracy of the OSM data is higher in urban areas. A more recent study by Barrington-Leigh and Millard-Ball (2017) claims that OSM has achieved more than 80% completeness for world roads, with more than 40% of countries having a fully mapped street network (i.e., > 95% streets are mapped). Of interest here are also the studies that have tested OSM data and its fitness for use in car navigation and routing (Graser et al., 2015; Zhang & Ai, 2015), as well as the quality of mapped building footprints (Müller, Iosifescu, & Hurni, 2015). Graser et al. (2015) have compared the OSM data to the official Austrian reference graph for routing in Vienna, showing that the median difference of the route lengths in the two datasets is only 1%. Zhang and Ai (2015) has focused on quality issues of OSM data related to routing such as tagging, modelling of junctions as dual carriageways, and missing or inconsistent road names. They also briefly mention that missing segments are one of the problems that may cause incorrect routing results, which is the same problem we discuss here.

There is an increasing trend for developing intrinsic approaches for assessing the quality of OSM data (Barron, Neis, & Zipf, 2014; Degrossi, De Albuquerque, Dos Santos Rocha, & Zipf, 2017; Maguire & Tomko, 2017), as well as detecting and fixing errors (Ali, Falomir, Schmid, & Freksa, 2017; Chittor Sundaram, Naghizade, Borovica-Gajic, & Tomko, 2020; Majic, Naghizade, Winter, & Tomko, 2019; Majic, Winter, & Tomko, 2017; Zhang & Ai, 2015). Intrinsic approaches use solely OSM data, which solves the problem of reference dataset unavailability. Ali et al. (2017) used the Association Classification method to learn the associations between map objects of different classes and recommend correct OSM tags to users and improve the quality of data classification. Chittor Sundaram et al. (2020) used the Membership Imputation Algorithm (MIA) to impute street names in OSM, thus improving the quality of the attribute data. Majic et al. (2019) used the frequent itemset mining technique to learn the topological constraints for bridges in OSM and detect incorrectly classified objects. Majic et al. (2017) used a semantic similarity analysis to analyze undocumented attribute keys in the OSM and propose their documented equivalents as substitutions to improve the quality of the attribute data. Zhang and Ai (2015) use the concept of stroke to recognize named roads in the map dataset and then use it to fix missing or incorrect road names and inconsistent attributes. The approach proposed in this paper can be considered an intrinsic approach to detecting errors since it does not need any other dataset than the one being analyzed. It is different from the approaches described above as it focuses on the accessibility of buildings and the problem of inaccessible buildings in the OSM, which was has not been addressed yet. The shortcoming of intrinsic approaches is that they depend on the quality of the dataset itself. If the majority of the data is not correct, then the statistical outliers will no longer indicate errors.

2.4 | RIM

The ray intersection model (RIM) (Majic et al., 2020) is a model for expressing nuanced ternary spatial relationships with the emphasis on the betweenness relationship. RIM takes in a core object O and two peripheral objects A and B in a 2-dimensional plane such that they may touch but not overlap each other. It then defines rays cast between A and B and analyses their topological relationships with O using the 9-intersection model (9IM) (Egenhofer & Herring, 1990). Here, the rays between A and B define the space that is considered to be between them. RIM also distinguishes extreme rays that denote the edge of this space between A and B. This lets RIM know if O is also protruding outside of this space. RIM combines the distinct 9IM relationships between O and rays to determine its position relative to A and B (e.g., O is between or not between A and B).

The output of RIM is a 4×3 matrix called the RIM matrix. It shows the combined intersections that the interiors of all rays (1st row), the boundaries of all rays (2nd row), the interiors of the extreme rays (3rd row), and the boundaries of the extreme rays (4th row) have with the interior (1st column), the boundary (2nd column), and the exterior (3rd column) of O (see the rightmost matrix in Figure 2). Fields in the RIM matrix can have three different values, each of which has a graphical representation. The value □ means that there are no rays that have the respective intersection with O. The value ⬕ means that some but not all rays have the respective intersection with O. The value ■ means that all rays have the respective intersection with O.

3 | BUILDING ACCESSIBILITY ANALYSIS

This study is concerned with the accessibility of buildings from the road network. Here, we define accessibility of a building as the ability to approach it in a straight line without having to avoid other buildings. The method proposed here uses the building's visibility from the road network to determine if it is accessible. For visibility we adopt a definition that "two points are mutually

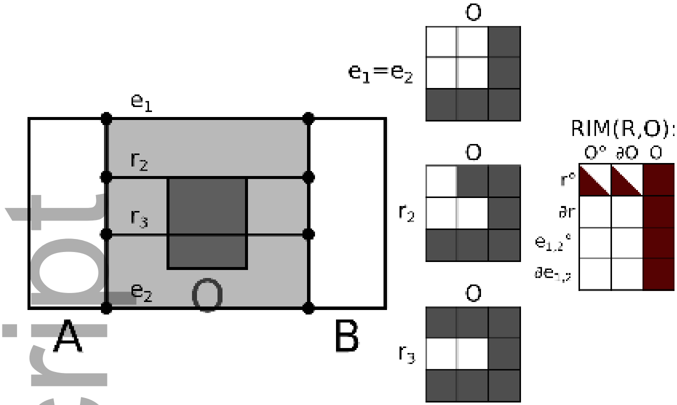


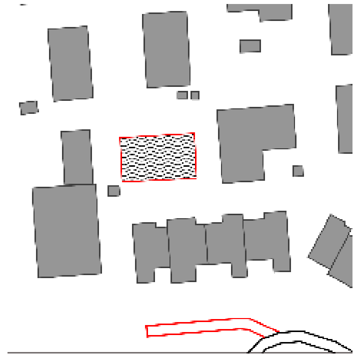
FIGURE 2 Peripheral spatial objects *A* and *B* (black outline), and the core object *O* (dark gray fill) in a 2D plane. The matrices show the distinct 9IM relationships between rays and *O*, and the RIM matrix shows the resulting RIM.

visible if the line segment connecting them is unoccluded" (Bittner & Wonka, 2003). The method also assumes that access to building if it is accessible will be provided by the road that is closest to it (see example in Section 1 for more detail). Thus, the problem addressed here is: *given a dataset of buildings and a road network, determine which buildings are accessible from the road network*. The presented method shows how to determine the accessibility of a single building and can be applied iteratively to analyse all buildings in a given dataset.

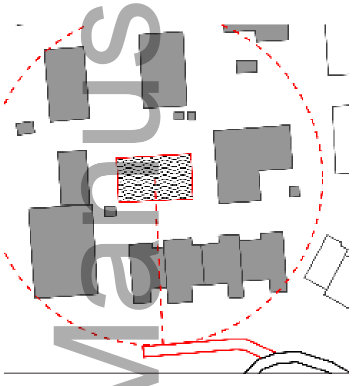
3.1 | Refinement of RIM

To find out if a building is accessible from its closest road, our method proposes a modified version of RIM. RIM originally focuses on the core object whose position is being tested in regards to the two peripheral objects. The aim in RIM is to determine if the core object is positioned *between* peripheral objects. In the building accessibility analysis, the spatial objects involved are the building being analysed – the *core building* C, its *closest road* R, and a set of *potentially occluding objects* O that may occlude the visibility between C and R. The distance between C and R will be denoted as d_{CR}. Here the focus changes from the object in between to one of the objects that would be considered peripheral in RIM – the *core building*. Figure 3 shows four major steps in the building accessibility analysis. First, the core building is designated (Figure 3a). Then, the roads in the proximity of the core building are analysed to find its closest road (Figure 3b). Then the potentially occluding objects are identified. Here, the algorithm only considers buildings for which the distance to the core building is less than d_{CR} (Figure 3c). It is still possible for buildings where the distance to the core building is greater than d_{CR} to be positioned between C and R and partially occlude access (e.g., if the closest road in Figure 3c was extended to the right, the white buildings in the bottom right of the scene would be between C and R). However, given the optimisation step introduced prior to the execution of this method where R is cut to the portion that is relevant to the context of C (see Section 4.3), such buildings are less likely to affect the outcome of the accessibility analysis. The last step is to analyse all the rays that exist between the core building and its closest road, and their intersections with the potentially occluding objects (Figure 3d). If there exists such a ray that is not intersected by any of the potentially occluding objects, then this ray represents a line of sight by which the core building is visible from its closest road. This also means that the core building can be accessed from the closest road and the algorithm will classify it as *accessible*. On the other hand, the potentially occluding objects between the core building and its closest road might be intersecting all of the rays completely occluding one from the other. This means that the core building cannot be accessed from its closest road and it will be classified as *inaccessible*.

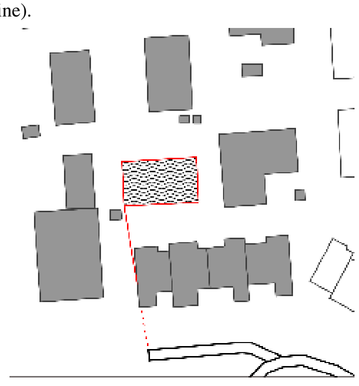
(a) Designate the *core building* to be analysed (pattern fill).



(b) Determine the *closest road* to the core building (red outline).



(c) Determine the relevant surrounding buildings that are within distance d_{CR} to the core building (grey fill).



(d) Test the visibility between the core building and its closest road, with respect to the surrounding buildings (red line shows an intercepted line of sight).

FIGURE 3 Illustration of steps in the proposed approach for the building accessibility analysis.

If we wish to describe the scenario in Figure 3 with RIM, the inputs would be C and R (i.e., peripheral objects in RIM), and O. The output of RIM(C,R,O) would be the RIM matrix calculated based on the intersections of rays between C and R, and all potentially occluding objects O (Figure 4). In the case of building accessibility, we are interested in the field RIM[1,1] that captures the intersections between rays' interiors and the interior of O. If the value in this field is □ or ◣, this means that there are some rays that are not intersecting the interior of O. These will be the access paths to the core building C which means that C is accessible. If the values in this field is ■, this means that all rays' interiors are intersecting O and C is inaccessible. The other fields in the RIM(C,R,O) matrix can help us further differentiate scenarios that are all accessible or inaccessible. However, this is outside of the scope of this paper. Thus, we do not utilise the entire RIM matrix in the rest of this study. The detailed explanation of how the value in the field RIM[1,1] is determined is explained in the next section.

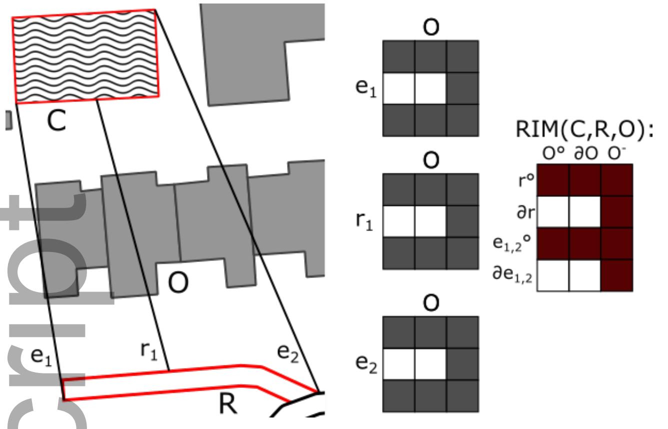


FIGURE 4 RIM shows that the core building C is not accessible from its closest road R because the occluding object O intersects all rays between C and R.

3.2 | Visibility between C and R given the occluder O

To determine whether the core building C can be seen and consequently accessed from its closest road R, visibility analysis is undertaken. The goal here is not to determine which parts of C are visible from R or obscured by O, but only to confirm if there exists at least one ray between C and R that is not intercepted by O. Such a ray represents the line of sight between C and R, and in the context of this method, proves that C is accessible from R. Algorithm 1 presents the computational steps of this analysis, while Figure 5 shows examples of three different scenarios of C, R, and O together with the visibility rays. Although we have referred to O as a set of objects, O is here treated as a single mutli-geometry object (i.e., a multipolygon consisting of one or more polygon parts).

The first step in this analysis is to find the area between C and R (Algorithm 1 line 1; gray areas in Figure 5). In the RIM, this area is called the ray area R_A and is defined as the area covered by all rays between C and R (for details of the ray area computation algorithm see (Majic et al., 2020)). The next step is to define all the rays between C and R that can potentially provide visibility between them. For this purpose, the algorithm extracts all the vertices of C, R, and O, and keeps only the ones that are intersecting R_A (e.g., $c_1 - c_4, o_1 - o_4, r_1 - r_4$ in Figure 5a). This is because any straight line passing through a vertex that is disjoint from the R_A is impossible to share a point with both C and R. Thus, such lines cannot be visibility rays between C and R and can be omitted from the rest of the analysis. After the algorithm has extracted and selected the vertices, it creates straight lines passing through pairs of vertices achieved by combining the vertices of C with the vertices of O, the vertices of R with the vertices of O, and the vertices of O with the vertices of O (Algorithm 1 line 5). These lines are also extended on both sides to reach both C and R. After each line is created, the algorithm tests if the line is crossing the occluder object O (Algorithm 1 line 10). If yes, then this means that the line does not provide visibility between C and R. If not, then the algorithm tests if this line has intersections with both C and R (Algorithm 1 line 12). It may happen that the line touches only one of these objects and in that case it still does not provide visibility between C and R. Only if the line does not intersect O and touches both C and R it can be considered as a valid line of sight between C and R. If this is the case, then the iterative search process can end and the algorithm can return `True` to confirm that there is visibility between C and R (Algorithm 1 line 14).

If O does not have any vertices inside the R_A, then the algorithm will not be able to find the visibility ray in the previously described iterative search process (e.g., see Figure 5c). In this case, the algorithm will enter a second search process where it creates candidate rays through the pairs of vertices achieved by combining the vertices of C with the vertices or R (Algorithm 1 line 15). Because these lines are created from the vertices of C and R, they are guaranteed to intersect both of them. Thus, it is sufficient only to check if these lines are crossing O to test if they provide visibility between C and R. In the end, the algorithm will return `True` if there is visibility between C and R and `False` if there is not.

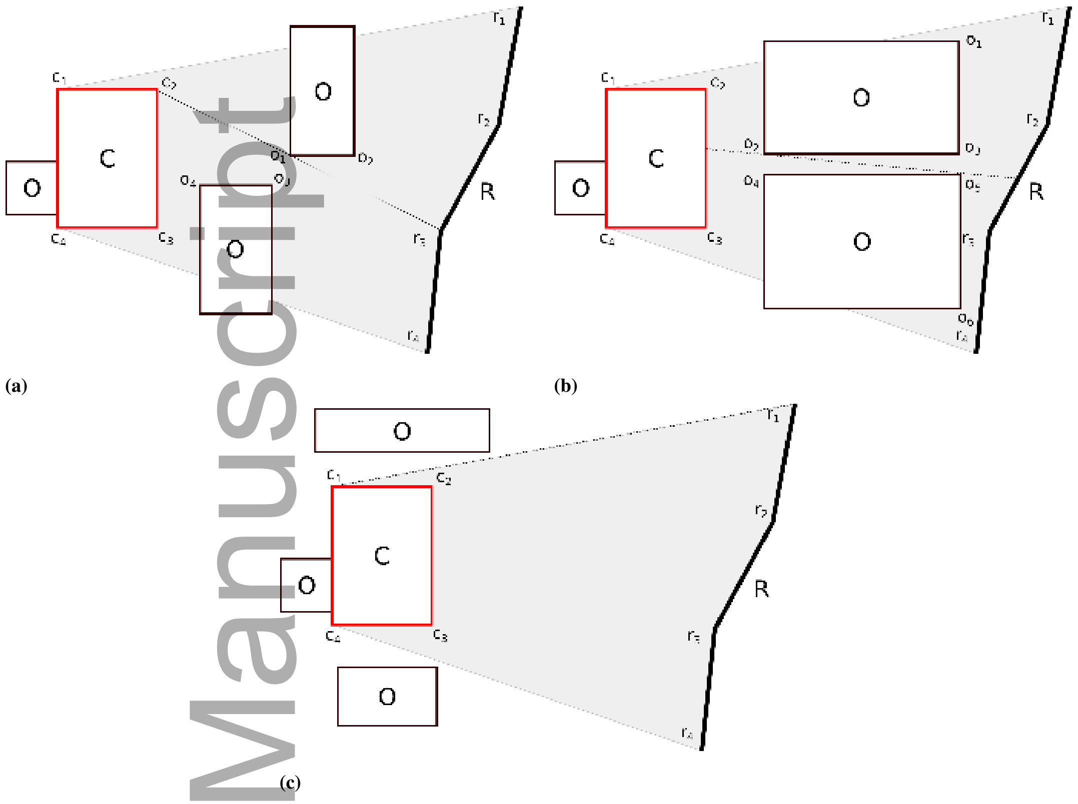


FIGURE 5 Illustration of the core object C, the closest road R, and potentially occluding objects O. The ray area between C and R is shown in gray and the dotted line between them is the line of sight.

Figure 5 shows examples of three different placements of O relative to C and R. The core buildings C in these figures are shown in red outline, the ray areas between C and R are shown in grey, and the vertices of C, R, and O that intersect the R_A are denoted with lowercase letters (i.e., $c_i, r_i,$ ando_i). In Figure 5a, after the Algorithm 1 analyses the ray that passes through c_1 and o_1, it rejects this ray as the possible visibility ray because it crosses O. However, when the algorithm tests the ray that passes through c_2 and o_1, it will conclude that there exists visibility between C and R because this ray does not cross O and touches both C and R.

In Figure 5b there is no visibility ray that passes through any vertex of C or R. Instead, the algorithm finds the visibility ray by analyzing rays that pass through different vertices of O (e.g., o_2 and o_5).

In Figure 5c, O is completely outside (i.e., topologically disjoint) of the R_A. In this case the algorithm analyses the rays that pass through vertices of C and R. Since the ray that passes through c_1 and r_1 does not cross O, the algorithm accepts it as the visibility ray between C and R.

Algorithm 1 Visibility test

Require: Geometries of two spatial objects (C and R) between which the line of sight is searched for, and a multi-geometry of the occluder object (O)

```
R_A ← CALCULATE_RAY_AREA(C,R)                          ▷ For this function, see (Majic et al., 2020)
cVertices ← Intersection(C.getVertices(),R_A)
rVertices ← Intersection(R.getVertices(),R_A)
oVertices ← Intersection(O.getVertices(),R_A)
for vertices in [cVertices, rVertices, oVertices] do
    for v1 in vertices do
        for v2 in oVertices do
            if v1 ≠ v2 then                            ▷ Two distinct points needed to construct a line
                candidateRay ← MakeLine(v1,v2).Extend()
                if candidateRay.Crosses(O) then
                    visibility ← False
                else if candidateRay.Touches(C) AND candidateRay.Touches(R) then
                    visibility ← True
                    return visibility                  ▷ Visibility ray found and search ends
for v1 in cVertices do
    for v2 in rVertices do      ▷ When there are no oVertices, construct rays between cVertices and rVertices
        if v1 ≠ v2 then
            candidateRay ← MakeLine(v1,v2).Extend()
            if candidateRay.Crosses(O) then
                visibility ← False
            else
                visibility ← True
                return visibility                      ▷ Visibility ray found and search ends
return visibility                                      ▷ Visibility ray not found, False returned
```

4 | CASE STUDY ON BUILDINGS IN VICTORIA, AUSTRALIA

Routing algorithms depend on the quality of map data they use to calculate and provide users with correct routes to their destinations. The proposed approach enables to assess the accessibility of buildings in a map dataset. By detecting inaccessible buildings, it points to potential errors in map data. To demonstrate and test the proposed approach, we have applied it to buildings in the OSM dataset for the State of Victoria, Australia. This study recognizes that there are some exceptions, such as sheds, storage buildings, and other outbuildings that are often accessible only *through* the main building on the property. Thus, in this case study, we have only analyzed buildings for which it is presumed they *should* be accessible from the road network. Using binary classification, the analyzed buildings have been classified as either *accessible* or *inaccessible*. In the end, *inaccessible* buildings have been presented via an online mapping challenge to mappers who have then assessed individual situations and corrected any potential errors in OSM.

4.1 | Dataset and experimental setup

We obtained the OSM data for the State of Victoria, Australia as the OSM data extract[1] on March 26, 2020. The downloaded data extract was already confined to the territory of Victoria and no additional steps were taken in regards to the geographical extents of the dataset. The setup consists of a PostgreSQL database (version 9.5.21) extended with PostGIS (version 2.4.2). The

[1] http://download.openstreetmap.fr/extracts/oceania/australia/

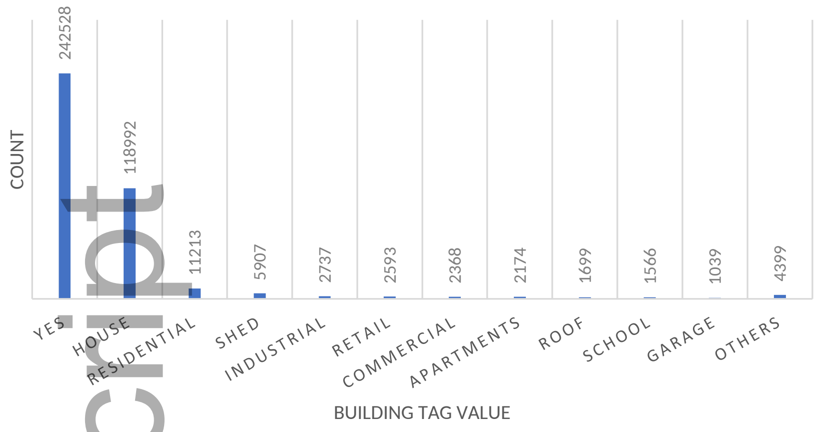


FIGURE 6 The most frequent building tags in the imported OSM dataset. *Others* represents all other tags that occur less frequently.

data were imported using the osm2pgsql importer (version 0.88.1), and a part of the processing in the experiment was done with Python3 scripts and the RIM Python3 package[2].

We have extracted the buildings and the road network data from the overall dataset since this is the only information that is needed for the proposed method. The osm2pgsql importer by default divides the data across four core tables: `planet_osm_point`, `planet_osm_line`, `planet_osm_roads`, and `planet_osm_polygon`. From the `planet_osm_polygon` table we have extracted all objects which are tagged with the *building=** [3] tag (i.e., * is a standard way of denoting any value used with a specific attribute key) and stored them into the `osm_buildings` table. Likewise, we have extracted all objects from the `planet_osm_line` table that are tagged with the *highway=** [4] tag and stored them into the `osm_highways` table. We have decided to use `planet_osm_line` table as a source of our road network data instead of the `planet_osm_roads` table because the latter also contains administrative boundaries and other objects missing the `highway=*` tag. Table 1 shows the relevant statistics for buildings and highways.

Figure 6 shows the most frequent building tags in the dataset. The majority of the buildings are tagged with the generic tag *building=yes*. This tag only confirms that the object is a building, but does not make it clear what is the type of the building. This is problematic when we want to focus our analysis on certain types of buildings and exclude others. Thus, we have used the area of the building in the results section to semantically differentiate main buildings from the outbuildings. From the other tags, it is notable that there are almost 6000 sheds, almost 1700 roofs, and over 1000 garages in the dataset. These can be considered outbuildings and are not expected to have access from the closest road. Instead, the access to these buildings is often provided through the main building on the property or other roads which are usually not mapped (e.g., private driveways and dirt tracks). Lastly, it is notable that there are over 4000 buildings tagged with other less frequent tags (135). These tags have not been analyzed individually, since there are 135 of them and they all have less than 1000 occurrences (and often much less), which translates to 0.25% of buildings in the dataset.

Table 2 shows all of the highway tags that are present in the dataset and how they have been divided them into five categories. Since this study is concerned with the accessibility of buildings from the road network, this categorisation helps to differentiate

[2] https://pypi.org/project/rim
[3] https://wiki.openstreetmap.org/wiki/Key:building
[4] https://wiki.openstreetmap.org/wiki/Key:highway

OSM highways that are assumed to be a part of that network (i.e., *car* category) from those that are not. However, there are also more than 47000 highways that are generically tagged as *highway=unclassified*. For these highways we cannot be certain if they are part of the road network, so they could not be excluded from the analysis. Thus, we only use highways categorised as car and unclassified in the rest of the case study.

4.2 | Experiment execution

The experiment was carried out following the four steps for building accessibility analysis as shown in Section 3 and in Figure 3. The analysis was partly executed in the database with the SQL queries and partly with Python scripts where the RIM Python3 package was utilized to finally assess the visibility of buildings from their closest roads.

The second step of the building accessibility analysis is to find the core building's closest road. In this experiment we have limited the search distance for closest roads to 400 m, as we believe this is the maximum realistic distance a person would walk from the road over empty ground to access a building. This distance is calculated from the bounding box of the building's geometry to the road. This approach does not take into consideration the location of the building's entrance as this information is not available for most buildings in the OSM dataset. Out of all roads that satisfy this criteria, the one that is closest to the core building is designated as its closest road. Buildings that have no roads within this search distance are by default classified as *inaccessible*.

Also, a pre-processing step is applied to cases where core buildings and the closest roads are intersecting each other (i.e., the distance between them is 0 m). In these cases the core building is automatically classified as *accessible* because it is impossible for its access to be completely occluded. This leaves less buildings for which the visibility analysis needs to be performed and speeds up the total execution time of the experiment.

4.3 | Analysis optimization

In all cases where core buildings and the closest roads are disjoint (i.e., the distance between them is > 0 m) the analysis is optimized by only considering a portion of the closest road that is relevant for the core building. This is needed because some road segments in OSM can be up to several kilometers long. Analyzing the whole road segment in such cases would be computationally demanding and analyzing the accessibility of a building to a point on the road which is far away (e.g., 1 km) would not produce useful results. Instead, only the part of the closest road that is relevant to the local context of the core building is cut out using a custom SQL function
ST_Building_Crop_Road(geometry building_geom, geometry road_geom, cut_distance numeric) which was created specifically for this task. This function first projects the core building's vertices onto the closest road. From these projected points it then finds two points (p_1, p_2) that are closest to the ends of the road linestring. The function's `cut_distance` parameter specifies how far from these points the road should be cut to leave only the portion that is relevant to the core building. In this experiment we have used a constant
`cut_distance` of 100 m (Figure 7). Each of the projected points p_1 and p_2 will be closer to one end of the road than the other projected point and this determines the direction for the interpolation in the next step. Then, two new points (p'_1, p'_2) that are `cut_distance` removed from the corresponding projected points are interpolated on the road in previously determined directions. These interpolated points will be the end points of the cut road.

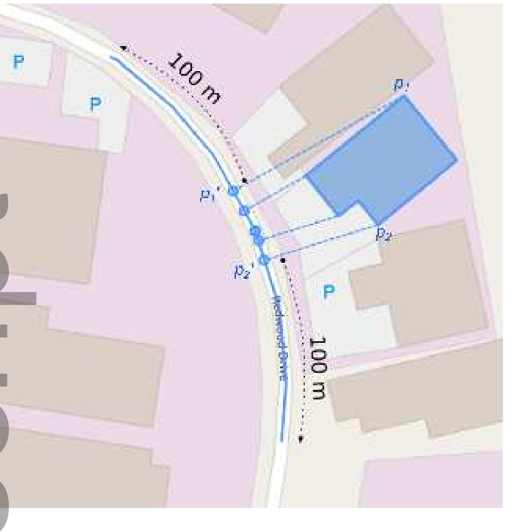


FIGURE 7 The closest roads were cut to a 100 m from the core building's projection points towards the ends of the road linestring to improve the speed of calculations.

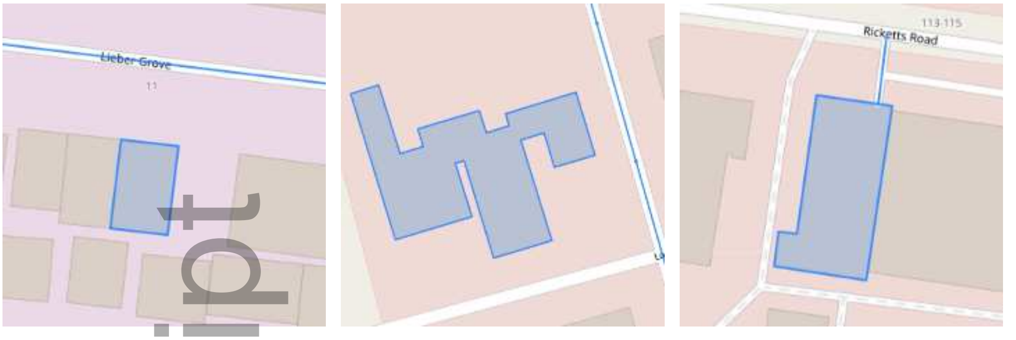
4.4 | Results

This section lays out the results of the case study in Victoria where almost 400000 buildings were classified based on their accessibility from the road network into *accessible* and *inaccessible* buildings. Table 3 shows the overall results of the classification. *Accessible* buildings in the dataset far outnumber the *inaccessible* buildings as one would expect in a well covered, well mapped area of OSM. However, there are still more than 2000 *inaccessible* buildings that potentially indicate erroneous data and need to be checked. There were also 152 buildings in Victoria for which the accessibility could not be computed due to problems such as invalid geometry of the core building, the closest road, or the occluder object.

Table 4 breaks down the *accessible* buildings based on the reason why they are classified as such. It is clear that for the vast majority of the buildings, the visibility analysis was needed to confirm that they are accessible from their closest roads. This table also shows the buildings that were classified as *accessible* by default, without the need to perform the full visibility. In 40447 cases the buildings were classified as *accessible* because there were no occluder buildings in the surroundings of the core building, so there was nothing to obstruct the access to it. There were also 2616 cases where the core building is intersecting its closest road which then means that the building can be accessed from the road.

Table 5 shows the results by the most frequent building tags in Victoria (i.e., building tags here correspond to building tags in Figure 6). It can be noticed that accessibility is very high for all building types, but this is expected given that more than 99% of all buildings in the dataset are *accessible* (Table 3). Although the ratios of *inaccessible* buildings across different types of buildings are low, they are the lowest in buildings where people live – namely *house (*0.10%), *residential (*0.34%), and *apartments (*0.78%). Buildings tagged as *roof (*3.18%) and *school(*2.87%) have the highest ratio of *inaccessible* buildings.

The frequencies of building tags that can be found in buildings that are classified as *inaccessible* are shown in the last column of Table 5. The frequencies of different building tags roughly follow the frequencies of tags for all buildings in the dataset (Figure 6). Once again, the majority of buildings (74%) are tagged with a generic *yes*. It can be seen that some of the most frequent tags found in *inaccessible* buildings are those that can be considered auxiliary to what are considered main buildings on the property. Buildings like *shed*, *roof*, *silo*, and *garage* are not always excepted to be accessible from the road and can often be accessible through main buildings on their properties.



(a) Accessibility analysis **(b)** No occluder buildings **(c)** $d_{CR} = 0m$

FIGURE 8 Examples of *accessible* buildings and different reasons for such classification.

We filtered the *inaccessible* buildings by selecting only buildings which are considered to be main buildings and hence expected to be accessible from the road network. Table 6 shows the two conditions used in this filtering process, where the conditions are connected with a logical OR operator. The first condition selects all buildings tagged with *house, school, residential, industrial, retail, commercial, apartments, university, train_station,* or *hospital*. The second condition selects buildings tagged with *yes*, but only if their areas are larger than 50 m^2.

After filtering, we identify 1585 buildings in the dataset which according to their tags and size should be accessible from the road network, but are classified as *inaccessible*.

4.5 | Online mapping challenge

The 1585 buildings in Victoria that were classified as *inaccessible* (Table 7) have been published online for mappers from the OSM community to assess and fix any potential errors[5]. The buildings were published together with the closest roads in the form of a *MapRoulette* challenge. *MapRoulette* is a micro-tasking platform for OSM. It presents mappers with small mapping tasks that can be completed in a short time and its goal is to improve the quality of OSM data. A *MapRoulette* challenge comprises of group of tasks dealing with the same problem. Multiple mappers can participate in the same challenge, and the progress of the challenge is tracked with various measures.

4.5.1 | Mapping tasks

Each mapping task presented mappers with an *inaccessible* building and its closest road drawn over aerial imagery. The mappers can complete tasks in the following ways:

- Edit - the mappers can edit the OSM features presented in the task, or other features in that locality and mark the task with “I fixed it!”;
- Skip - the mappers can skip the current task and go to the next one (the skipped task will still be available to other mappers);
- Not an issue - if they believe that the problem presented in the task is not an issue in the OSM, the mappers can mark the task as such;

[5] https://maproulette.org/browse/challenges/13044

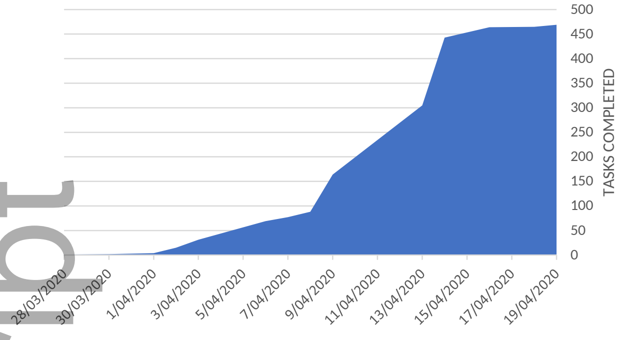


FIGURE 9 MapRoulette mapping challenge progress over time.

- Too hard / Can't see - the mappers can mark the task if they are not able to complete it due to poor imagery or the difficulty of the task;
- Already fixed - the mappers can use this mark when they believe that the problem presented in the task has already been solved.

In the task description for this specific challenge, mappers were given the following list of possible problems in the OSM data that might cause the building in the task to be *inaccessible*, and how to fix them:

- missing road - map the missing roads around the building and mark the task as fixed, alternatively if the roads are not visible in the satellite imagery mark the task as *too hard* and *hard to see*
- outbuildings - if the building is tagged as building=yes change the tag to a more specific tag, such as building=shed and mark the task as fixed; otherwise, if the building is correctly tagged with something other than building=yes, and still does not have an access road, mark the task as *not a problem* and add MR tag 'small building'
- building inside building - if one of the features is incorrectly tagged as a building you could remove the building=* tag from it and mark the task as fixed; otherwise if both objects are buildings and you believe they are mapped correctly you could mark the task as not a problem and add MR tag 'building inside the building'
- building accessible from another highway - if the access to the building provided by another road that is not its closest road; in this case the task can be marked as not a problem with MR tag 'another access'

4.5.2 | Mapping progress

This section shows the progress of the MapRoulette challenge until April 21, 2020. In 22 days that the challenge has been active, total of 469 tasks have been completed by mappers (Figure 9) which approximately covers 30% of all identified *inaccessible* buildings in Victoria (Figure 10). Figure 10 shows the statistics of how mappers have completed these tasks. The majority of tasks have either been fixed (29% of all completed tasks) or marked as not an issue (59% of all completed tasks). There were much fewer tasks that were skipped, too hard, or already fixed.

For the purposes of evaluating the proposed method, the results of binary accessibility classification have been compared to the mappers' responses in the mapping challenge. Buildings that were classified as *accessible* are considered *positives* and buildings classified as *inaccessible* are considered *negatives* in the contingency table (Table 8). The evaluation of *positives* was not possible in this study because buildings classified as *accessible* were not presented in the mapping challenge as they do not

FIGURE 10 Statistics of the MapRoulette mapping challenge.

(a) Comment says that building tagged *building=yes* is a shed. **(b)** Comment says that building is within another building.

FIGURE 11 Examples where tasks are marked as not an issue, but comments indicate that issue exists.

indicate potential errors. Thus, counts of *true positives and false positives* were not determined. The tasks that were completed as fixed can be seen as *true negatives* in the classification results because they imply that mappers have confirmed the presence of an error in data that was causing the building to be classified as *inaccessible*. Similarly, tasks marked as not an issue can be seen as *false negatives* in the classification results because they imply that the algorithm has incorrectly classified the building as *inaccessible*. However, we have observed that there have been cases where mappers have concluded that the building presented in the task is not expected to have access from the road based on their personal knowledge and experience of mapping practices in the OSM. In these cases, the proposed method has correctly identified that the building is *inaccessible* and the task should not contribute to the number of *false negatives*. This is also supported by our analysis of the comments that mappers have made when solving these tasks, where two predominant cases where this happens have been noticed. There are 6 tasks that were marked as not an issue, but the comment says that the *inaccessible* building is actually a shed that is tagged differently. Another 50 tasks marked as not an issue are commented as cases where the core building is located topologically within another building which always makes it completely occluded and thus *inaccessible*. Since both of these problems point to errors in OSM data that should be corrected, these tasks were counted towards *true negatives* in the classification evaluation shown in Table 8.

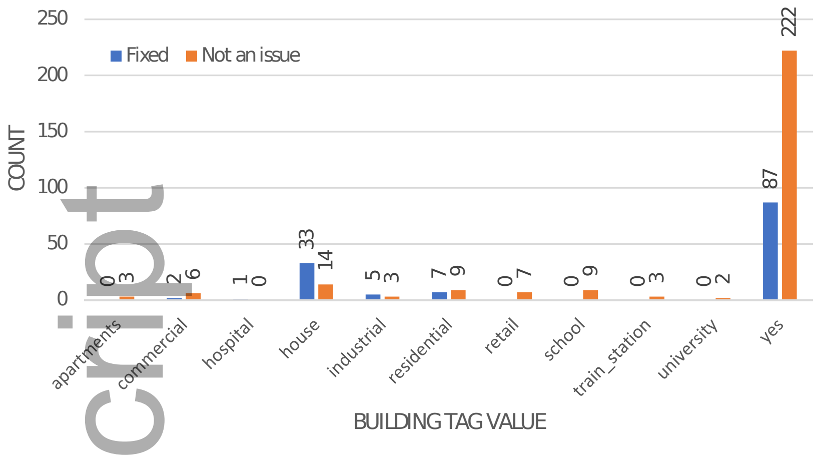


FIGURE 12 Distribution of fixed and not an issue MapRoulette tasks across different building tags.

Finally, Figure 12 shows the distribution of tasks marked as fixed and not an issue across different types of buildings. It can be seen that residential buildings (i.e., tagged with house and residential) are among the most common where the problem of inaccessibility was correctly detected. In public buildings tagged with commercial, retail and school there is a much higher ratio of false negatives.

5 | DISCUSSION

The case study shows how the proposed building accessibility approach can be applied to detect inaccessible buildings and thus point to potential errors in the dataset. The fact that the vast majority of the buildings in the dataset were found to be accessible (Table 3) only justifies the assumption made in the proposed approach that the buildings should be accessible from the road network. In terms of patterns in the data, buildings that do not have access from the road network can be described as outliers. Given that the outlier detection is a method that is often used in data cleaning to indicate errors in data, it is reasonable to assume that in this case inaccessible buildings can serve as indicators of potential errors as well. The difference in this study is that the pattern of majority buildings being accessible from the closest roads has not been learned from the data, but theoretically assumed and later confirmed by the analysis.

The online mapping challenge has been a great platform to present the detected potential errors to the OSM community so that they can be fixed. The response from the community has resulted in one third of detected inaccessible buildings to be assessed and potential errors corrected within 22 days. This highlights how useful the proposed method is when applied to improve map data. Even though Victoria can be considered a well mapped area in OSM where data are mature and have been repeatedly revised by active contributors to assure their quality, the proposed method has discovered errors that were previously not known. This might be due to the fact that the accessibility analysis presented in this paper is a complex spatial analysis that simultaneously considers multiple spatial objects. Such task is not trivial and these errors might be hard to notice when a person is looking at the map with a naked eye. But when potential errors are detected automatically, mappers are able to direct their efforts towards correction of errors. This can save considerable time and resources in the data quality assurance process.

Apart from the obvious benefit of improving the quality of OSM data, the challenge has also provided valuable feedback from 18 different mappers. This provides the opportunity to gain a deeper understanding of specific OSM dataset for Victoria as

FIGURE 13 Buildings on the waterfront near Lakes Entrance that can only be accessed by boat.

well as the performance of the proposed approach. At first it may seem as though the majority of tasks addressed by mappers have been buildings incorrectly classified as inaccessible (Figure 10). However, a further analysis of the mappers' comments associated with those tasks show that a meaningful portion of these cases are indeed inaccessible buildings, but for which the mappers' believe it is acceptable to remain without access. These are exactly the kinds of buildings we have partially filtered out prior to the building accessibility analysis with conditions shown in Table 6. For example, buildings annotated as sheds have not been analyzed in this experiment because it is expected for sheds to often be located behind main buildings and be accessible indirectly through them. Furthermore, buildings tagged with the generic *building=yes* tag that are smaller than 50 m^2 have also been removed because they could be sheds. We have instructed mappers to fix the tags of sheds that are annotated differently and have made it into the accessibility analysis because they are larger than 50 m^2. However, there were still cases where the mappers have recognized the sheds but have marked the task as not an issue instead of fixing the error (Figure 11a). Another example are buildings that are topologically positioned within other buildings (Figure 11b). Because the proposed approach is using topological relationships between the rays cast between the core building and the closest road and the surrounding buildings, in these cases the occluder building containing the core building will intersect all rays. A potential solution for such situations may be to disregard those occluder buildings that contain both the core building and its closest road. An example where this often occurs are the shopping malls where individual shops are often mapped inside the outline of the whole complex.

When the results of the online mapping challenge are adjusted for these cases the performance of the proposed approach improves as shown in Table 8. However, according to mappers there are still more false than true negative classification results. We suspect there are two major factors that affect this and that may further improve the performance of the proposed method. The first is the assumption that majority of buildings should have access from the road network. There are cases in Victoria where residential buildings are rightfully inaccessible from the road network. One example are the houses on the beach of Lakes Entrance that are accessible only by sea (Figure 13). The road network does not exist in that area and according to the proposed approach these buildings will always be inaccessible.

The other factor that affects the performance of the method is the way in which this assumption is realised in the proposed approach. We have only considered the building tags and areas of building objects. However, there are cases in OSM where the semantics of buildings are recorded with a combination of different tags. One such example are the silos that are usually annotated with *building = yes* and *man_made = silo*. Because they are usually larger than 50 m^2 they have not been filtered out of the analysis. Figure 14 shows a group of silos in a rural industrial area that are all classified as inaccessible. However, it is often not expected for them not to have a built access road or such road not to be mapped since it is on the private property.

Lastly, it may also happen that the access to a building is provided by a road that is not the closest one. Although there have

FIGURE 14 Silos that are classified as inaccessible.

FIGURE 15 Building is accessible from the road on the right, but the algorithm has tested the accessibility from the road on the left as it is slightly closer to the building.

not been many such cases in the case study, Figure 15 shows one such example. The building is classified as inaccessible by the proposed approach based on the analysis of the road on the left. However, the building is actually accessible from the road on the right which is slightly further away. For situations like these it may be worth analyzing the accessible from multiple closest road if the difference in distance the them is small. This would lower the number of buildings incorrectly classified as inaccessible, but would increase the time needed to perform the the analysis.

6 | CONCLUSIONS AND FUTURE WORK

The aim of this study was to make use of the assumption made in most routing algorithms that the access to a building should be provided by the road that is closest to it. It sometimes happens that there are other buildings in the way between the building and its closest road that are obstructing access. The proposed approach for building accessibility analysis uses 2D map visibility between the building and its closest road to determine if there exists at least one unoccluded line of sight between these objects in a map. If yes, then this line of sight is understood as a line of access and the building is classified as accessible. Otherwise, if there is no unoccluded line of sight the building is classified as inaccessible. The value of this analysis is that inaccessible buildings often indicate potential errors in the map dataset such as incomplete data (i.e., the access road is not mapped) or incorrect attribute information (i.e., an outbuilding may be mapped as a residential building). It also has the potential to be part of a richer data processing pipeline that ensures high quality of data.

The case study on buildings in Victoria has demonstrated how the proposed approach may be used on OSM data. Given that the proposed approach is intrinsic and does not use any external ground truth datasets, it will only be effective where data are rich enough – for OSM dataset in Victoria this seems to be the case. Also, we believe that this approach would be applicable to other locations in the world as we have detected inaccessible buildings in New Zealand as well. However, we could not evaluate the proposed approach in New Zealand as we did not get a meaningful number of edits and responses to the mapping tasks. There were 1585 buildings that were classified as inaccessible. These buildings were then published in the form of a MapRoulette mapping challenge online where other mappers from the OSM community can contribute by correcting errors in OSM and/or providing feedback on the analysis. With the feedback of almost 500 inaccessible buildings that were assessed by mappers in MapRoulette, there were 135 errors in the OSM data that were fixed. The number of data that were classified as inaccessible by the algorithm but were marked as "no an issue" by mappers initially seemed high. However, after some of the mappers' comments were analyzed, we conclude that 56 of those cases actually point to errors in the data, but were not fixed. After adjusting for those values, the evaluation of the proposed approach provides balance between the number of detected inaccessible buildings that point to actual problems in the data (i.e., 191), and the number of inaccessible buildings that mappers believed are acceptable to be inaccessible (i.e., 222).

Some drawbacks of the proposed approach have been noticed with the results of the case study, and have been discussed in detail (Section 5). These provide directions for future work, especially the problem regarding the analysis of more tags in the OSM buildings data, other than only the main tag *building=**. Another issue that could be addressed in future studies is that access may sometimes be provided by the second or third closest road. Repeating the proposed approach n times for n nearest roads could potentially solve this problem, but would also increase the time needed for the algorithm to execute.

All in all, we consider the case study has proved the hypothesis that 2D map visibility can be used as a proxy for accessibility and detection of inaccessible buildings. Although mappers report seemingly high number of incorrectly classified buildings, we consider the ability of the proposed approach to detect 135 errors in the map data proves its plausibility. As a result, accessible buildings will enable routing algorithms to calculate correct routes. Consequently, correct routes will benefit all routing applications ,and in the case of emergency dispatch, they may even help save lives.

ACKNOWLEDGMENTS

Map data ©OpenStreetMap contributors are available from https://www.openstreetmap.org.

Support by the Australian Research Council (DP170100153 and DP170100109) is acknowledged. We also acknowledge the help from the OSM contributors who have participated in our MapRoulette challenge. In particular we thank the Microsoft Open Maps team (https://github.com/Microsoft/Open-Maps) for the help and feedback provided.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

REFERENCES

Ali, A. L., Falomir, Z., Schmid, F., & Freksa, C. (2017). Rule-guided human classification of Volunteered Geographic Information. *ISPRS Journal of Photogrammetry and Remote Sensing*, *127*, 3–15. doi: 10.1016/j.isprsjprs.2016.06.003

AlKahtani, S. J. H., Xia, J. C., Veenendaaland, B., Caulfield, C., & Hughes, M. (2015). Building a conceptual framework for determining individual differences of accessibility to tourist attractions. *Tourism Management Perspectives*, *16*, 28–42. doi: 10.1016/j.tmp.2015.05.002

Alonso, F. (2002). The benefits of building barrier-free: a contingent valuation of accessibility as an attribute of housing. *European Journal of Housing Policy*, *2*(1), 25–44. doi: 10.1080/14616710110120577

Balado, J., Díaz-Vilariño, L., Arias, P., & Soilán, M. (2017). Automatic building accessibility diagnosis from point clouds. *Automation in Construction*, *82*, 103–111. doi: 10.1016/j.autcon.2017.06.026

Barrington-Leigh, C., & Millard-Ball, A. (2017). The world's user-generated road map is more than 80% complete. *PLOS ONE*, *12*(8), e0180698. doi: 10.1371/journal.pone.0180698

Barron, C., Neis, P., & Zipf, A. (2014). A Comprehensive Framework for Intrinsic OpenStreetMap Quality Analysis. *Transactions in GIS*, *18*(6), 877–895. doi: 10.1111/tgis.12073

Billen, R., & Clementini, E. (2005a). Introducing a Reasoning System Based on Ternary Projective Relations. In P. F. Fisher (Ed.), *Developments in spatial data handling* (pp. 381–394). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/3-540-26772-7_29

Billen, R., & Clementini, E. (2005b). Semantics of Collinearity Among Regions. In R. Meersman, Z. Tari, & P. Herrero (Eds.), *On the move to meaningful internet systems 2005: Otm 2005 workshops. otm 2005. lecture notes in computer science* (Vol. 3762, pp. 1066–1076). Berlin, Heidelberg: Springer, Berlin, Heidelberg. doi: 10.1007/11575863_129

Bittner, J., & Wonka, P. (2003). Visibility in Computer Graphics. *Environment and Planning B: Planning and Design*, *30*(5), 729–755. doi: 10.1068/b2957

Chittor Sundaram, R., Naghizade, E., Borovica-Gajic, R., & Tomko, M. (2020). Harnessing spatio-temporal patterns in data for nominal attribute imputation. *Transactions in GIS*(00), 1–32. doi: 10.1111/tgis.12617

Clementini, E., & Billen, R. (2006). Modeling and computing ternary projective relations between regions. *IEEE Transactions on Knowledge and Data Engineering*, *18*(6), 799–814. doi: 10.1109/TKDE.2006.102

Dalumpines, R., & Scott, D. M. (2011). GIS-based Map-matching: Development and Demonstration of a Postprocessing Map-matching Algorithm for Transportation Research. In S. Geertmanm, W. Reinhardt, & F. Toppen (Eds.), *Advancing geoinformation science for a changing world, lecture notes in geoinformation and cartography 1* (Vol. 1, pp. 101–120). Springer-Verlag Berlin Heidelberg. doi: 10.1007/978-3-642-19789-5_6

de Berg, M., Cheong, O., van Kreveld, M., & Overmars, M. (2008). Visibility Graphs: Finding the Shortest Route. In *Computational geometry algorithms and applications* (pp. 323–333).

Degrossi, L. C., De Albuquerque, J. P., Dos Santos Rocha, R., & Zipf, A. (2017). A framework of quality assessment methods for crowdsourced geographic information: A systematic literature review. In *Proceedings of the international iscram conference* (Vol. 2017-May, pp. 532–545).

Dylla, F., Lee, J. H., Mossakowski, T., Schneider, T., Van Delden, A., Van De Ven, J., & Wolter, D. (2017). A Survey of Qualitative Spatial and Temporal Calculi: Algebraic and Computational Properties. *ACM Computing Surveys*, *50*(1), 1–39. doi: 10.1145/3038927

Egenhofer, M. J., & Herring, J. R. (1990). *Categorizing binary topological relations between regions, lines, and points in geographic databases* (Tech. Rep.). University of Maine, National Center for Geographic Information and Analysis and Department of Surveying Engineering, Department of Computer Science.

Fogliaroni, P. (2015). Visibility Relations. In S. Shekhar & H. Xiong (Eds.), *Encyclopedia of gis* (Vol. 8, pp. 1–8). Cham: Springer International Publishing. doi: 10.1007/978-3-319-23519-6_1541-1

Fogliaroni, P., Wallgrün, J. O., Clementini, E., Tarquini, F., & Wolter, D. (2009). A Qualitative Approach to Localization and Navigation Based on Visibility Information. In K. Hornsby, C. Claramunt, M. Denis, & G. Ligozat (Eds.), *Spatial information theory. cosit 2009. lecture notes in computer science* (Vol. 5756, pp. 312–329). Springer, Berlin, Heidelberg.

doi: 10.1007/978-3-642-03832-7_19

Ford, A., Barr, S., Dawson, R., & James, P. (2015). Transport Accessibility Analysis Using GIS: Assessing Sustainable Transport in London. *ISPRS International Journal of Geo-Information*, *4*(1), 124–149. doi: 10.3390/ijgi4010124

Galton, A. (1994). Lines of sight. In M. Keane, P. Cunningham, M. Brady, & R. Byrne (Eds.), *Ai and cognitive science* (pp. 103–113). Dublin University Press.

Girres, J. F., & Touya, G. (2010). Quality Assessment of the French OpenStreetMap Dataset. *Transactions in GIS*, *14*(4), 435–459. doi: 10.1111/j.1467-9671.2010.01203.x

Graser, A., Straub, M., & Dragaschnig, M. (2015). Is OSM Good Enough for Vehicle Routing? A Study Comparing Street Networks in Vienna. In G. Gartner & H. Huang (Eds.), *Progress in location-based services 2014, lecture notes in geoinformation and cartography* (pp. 3–17). Springer International Publishing Switzerland. doi: 10.1007/978-3-319-11879-6_1

Guha, P., Mukerjee, A., & Venkatesh, K. S. (2011). OCS-14: You can get occluded in fourteen ways. *IJCAI International Joint Conference on Artificial Intelligence*(Figure 1), 1665–1670. doi: 10.5591/978-1-57735-516-8/IJCAI11-280

Haklay, M. (2010). How good is volunteered geographical information? A comparative study of OpenStreetMap and ordnance survey datasets. *Environment and Planning B: Planning and Design*, *37*(4), 682–703. doi: 10.1068/b35097

Heidrich, W., Brabec, S., & Seidel, H.-P. (2000). Soft Shadow Maps for Linear Lights. In B. Péroche & H. Rushmeier (Eds.), *Rendering techniques 2000* (pp. 269–280). Vienna: Springer Vienna. doi: 10.1007/978-3-7091-6303-0_24

Kasapakis, V., & Gavalas, D. (2015). Geolocative Raycasting for real-time buildings detection in pervasive games. In *2015 international workshop on network and systems support for games (netgames)* (pp. 1–3). IEEE. doi: 10.1109/NetGames.2015.7382999

Köhler, C. (2002). The Occlusion Calculus. In *Proceedings of the 1st cognitive vision workshop* (pp. 1–6). Zürich, Switzerland.

Ligozat, G., & Santos, P. E. (2015). Spatial occlusion within an interval algebra. In *2015 aaai spring symposium series* (pp. 103–106).

Liu, S., & Zhu, X. (2004). An Integrated GIS Approach to Accessibility Analysis. *Transactions in GIS*, *8*(1), 45–62. doi: 10.1111/j.1467-9671.2004.00167.x

Luo, W., & Wang, F. (2003). Measures of Spatial Accessibility to Health Care in a GIS Environment: Synthesis and a Case Study in the Chicago Region. *Environment and Planning B: Planning and Design*, *30*(6), 865–884. doi: 10.1068/b29120

Maguire, S., & Tomko, M. (2017). Ripe for the picking? Dataset maturity assessment based on temporal dynamics of feature definitions. *International Journal of Geographical Information Science*, *31*(7), 1334–1358. doi: 10.1080/13658816.2017.1287370

Majic, I., Naghizade, E., Winter, S., & Tomko, M. (2019). Discovery of topological constraints on spatial object classes using a refined topological model. *Journal of Spatial Information Science*, *18*(18), 1–30. doi: 10.5311/JOSIS.2019.18.459

Majic, I., Naghizade, E., Winter, S., & Tomko, M. (2020). RIM: a ray intersection model for the analysis of the between relationship of spatial objects in a 2D plane. *International Journal of Geographical Information Science*, 1–26. doi: 10.1080/13658816.2020.1778002

Majic, I., Winter, S., & Tomko, M. (2017). Finding equivalent keys in openstreetmap: semantic similarity computation based on extensional definitions. In *Proceedings of the 1st workshop on artificial intelligence and deep learning for geographic knowledge discovery - geoai '17* (pp. 24–32). Los Angeles, California, USA: ACM Press. doi: 10.1145/3149808.3149813

Müller, F., Iosifescu, I., & Hurni, L. (2015). Assessment and Visualization of OSM Building Footprint Quality. *International Cartographic Conference*.

Nicoară, P. S., & Haidu, I. (2014). A gis based network analysis for the identification of shortest route access to emergency medical facilities. *Geographia Technica*, *9*(2), 60–67.

Pocchiola, M., & Vegter, G. (1996). The visibility complex. *International Journal of Computational Geometry & Applications*, *06*(03), 279–308. doi: 10.1142/S0218195996000204

Randell, D., & Witkowski, M. (2006). Using occlusion calculi to interpret digital images. *Frontiers in Artificial Intelligence and Applications*, *141*, 432–436.

Randell, D., Witkowski, M., & Shanahan, M. (2001). From Images to Bodies: Modelling and Exploiting Spatial Occlusion and Motion Parallax. In *Proceedings of the 17th international joint conference on artificial intelligence - volume 1* (pp. 57–63). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. doi: 10.5555/1642090.1642098

Randell, D. A., Cui, Z., & Cohn, A. G. (1992). A spatial logic based on regions and connection. In B. Nebel, C. Rich, & W. Swartout (Eds.), *Proceedings of the third international conference on principles of knowledge representation and reasoning* (pp. 165–176). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Rivière, S. (1997). Dynamic visibility in polygonal scenes with the visibility complex. In *Proceedings of the thirteenth annual symposium on computational geometry - scg '97* (Vol. 1997, pp. 421–423). New York, New York, USA: ACM Press. doi: 10.1145/262839.263057

Sakkas, N., & Pérez, J. (2006). Elaborating metrics for the accessibility of buildings. *Computers, Environment and Urban Systems*, *30*(5), 661–685. doi: 10.1016/j.compenvurbsys.2005.06.002

Santos, P. E., Casati, R., Dee, H. M., Schultz, C., & Bhatt, M. (2016). Eclipse in Occlusion A perspectival mereotopological representation of celestial eclipses. In B. Bredeweg, K. Kansou, & M. Klenk (Eds.), *Proceedings of the 29th international workshop on qualitative reasoning. qr2016* (pp. 93–100). New York, USA.

Santos, P. E., Dee, H. M., & Fenelon, V. (2008). Notes on a qualitative theory of shadows. *Cognitive Robotics workshop at ECAI 2008*, 47–54.

Santos, P. E., Ligozat, G., & Safi-Samghabad, M. (2015). An Occlusion Calculus Based on an Interval Algebra. In *4th brazilian conference on intelligent systems (bracis 2015)* (pp. 128–133). Nadal, Brazil: IEEE. doi: 10.1109/BRACIS.2015.12

Tansley, G., Schuurman, N., Amram, O., & Yanchar, N. (2015). Spatial Access to Emergency Services in Low- and Middle-Income Countries: A GIS-Based Analysis. *PLOS ONE*, *10*(11), e0141113. doi: 10.1371/journal.pone.0141113

Tarquini, F., De Felice, G., Fogliaroni, P., & Clementini, E. (2007). A Qualitative Model for Visibility Relations. In J. Hertzberg, M. Beetz, & R. Englert (Eds.), *Ki 2007: Advances in artificial intelligence. ki 2007. lecture notes in computer science* (Vol. 4667, pp. 510–513). Springer, Berlin, Heidelberg. doi: 10.1007/978-3-540-74565-5_52

Welzl, E. (1985). CONSTRUCTING THE VISIBILITY GRAPH FOR n-LINE SEGMENTS IN O(nˆ2) TIME. *Information Processing Letters*, *20*(May), 167–171.

Zhang, X., & Ai, T. (2015). How to Model Roads in OpenStreetMap? A Method for Evaluating the Fitness-for-Use of the Network for Navigation. In F. Harvey & Y. Leung (Eds.), *Advances in spatial data handling and analysis, advances in geographic information science* (Vol. 19, pp. 143–162). Springer International Publishing Switzerland. doi: 10.1007/978-3-319-19950-4_9

Zielstra, D., & Zipf, A. (2010). A Comparative Study of Proprietary Geodata and Volunteered Geographic Information for Germany. In *13th agile international conference on geographic information science* (Vol. 1, pp. 1–15). doi: 10.1119/1.1736005

type of objects	count
Imported polygon objects	566265
Polygon objects annotated with *building*=*	397215
Imported line objects	616318
Line objects annotated with *highway*=*	471046

TABLE 1 Numbers of polygon objects, buildings, line objects, and highways in the imported OSM dataset.

highway category	highway	count
car	living_street, motorway, primary, residential, road, secondary, service, tertiary, trunk	292161
foot and cycle	cycleway, footway, path, pedestrian, steps, track	113761
unclassified	unclassified	47020
car link	motorway_link, primary_link, secondary_link, tertiary_link, trunk_link	11046
other	abandoned, bridleway, bus_stop, construction, corridor, crossing, disused, platform, proposed, raceway, rest_area, services, traffic_island, yes	7058

TABLE 2 Highway categories, the highway tags that belong to each category, and their counts. Only highways categorized as car or unclassified were considered in this case study.

accessibility	count	ratio
accessible	393123	99.44%
inaccessible	2055	0.52%
null	152	0.04%

TABLE 3 The overall results of binary classification of buildings into *accessible* and *inaccessible*. Null contains cases where the computation could not be performed.

justification	count	ratio of all accessible
The accessibility analysis has confirmed the core building is accessible from the closest road (Figure 8a)	350060	89.05%
No occluder buildings were found in the surroundings of the core building (Figure 8b)	40447	10.29%
The distance between the core building and the closest road is 0 m (Figure 8c)	2616	0.67%

TABLE 4 Breakdown of *accessible* buildings based on the reason why they are classified as *accessible*.

building	accessible	ratio accessible	inaccessible	ratio inaccessible	ratio of all inaccessible
yes	239209	98.63%	1521	0.63%	74.01%
house	118761	99.81%	122	0.10%	5.94%
residential	11170	99.62%	38	0.34%	1.85%
shed	5732	97.04%	103	1.74%	5.01%
industrial	2693	98.39%	34	1.24%	1.65%
retail	2564	98.88%	29	1.12%	1.41%
commercial	2347	99.11%	20	0.84%	0.97%
apartments	2156	99.17%	17	0.78%	0.83%
roof	1644	96.76%	54	3.18%	2.63%
school	1517	96.87%	45	2.87%	2.19%
garage	1031	99.23%	8	0.77%	0.39%
others (sum)	4299	97.73%	64	1.45%	3.11%

TABLE 5 The classification results for most frequent building tags in the imported OSM dataset.

condition 1	building IN ('house', 'school', 'residential', 'industrial', 'retail', 'commercial', 'apartments', 'university', 'train_station', 'hospital')
	OR
condition 2	building = 'yes' AND area > 50

TABLE 6 Criteria used to filter inaccessible buildings by removing small buildings and outbuildings from the results.

building	inaccessible	ratio of all inaccessible
yes	1257	79.31%
house	122	7.70%
school	45	2.84%
residential	38	2.40%
industrial	34	2.15%
retail	29	1.83%
commercial	20	1.26%
apartments	17	1.07%
university	11	0.69%
train_station	7	0.44%
hospital	5	0.32%

TABLE 7 Filtered results for the most frequent buildings classified as inaccessible.

		Mappers	
		accessible	*inaccessible*
Classifier	*accessible*	TP	FP
	inaccessible	222	191

TABLE 8 Binary classification contingency table.