

# Towards the generation of 3D OpenStreetMap building models from single contributed photographs

Eliana Bshouty, Alexander Shafir, Sagi Dalyot\*

*Mapping and Geo-Information Engineering, Civil and Environmental Engineering, The Technion, Technion City, 3200003, Israel*

## ARTICLE INFO

**Keywords:**  
 3D city models  
 Building height  
 LoD1  
 Contributed photographs  
 OpenStreetMap  
 Newton's method in optimization

## ABSTRACT

3D city models are valuable and useful for many experts using this information for a wide array of environmental and sustainable analyses and services. To produce 3D city models, conventional production processes are typically conducted by authoritative mapping agencies, which mostly rely on aerial photogrammetry and laser scanning. Consequently, obtaining public domain 3D city models is challenging and limited, where the majority of open data is collected and mapped by participatory mapping driven communities. These are still limited to 2D data collection proficiencies due to the used mapping infrastructures and technological limitations. Thus, the 3rd (height) dimension is mostly missing from these maps and models, resulting in the fact that public domain 3D city models are still limited, and only scarcely used for environmental applications. Perhaps one of the most important features in 3D city models are the buildings, since they serve as a major geospatial element in many environmental applications. Our objective is to use a single contributed photograph and OpenStreetMap vector data to precisely calculate the photographed building height, and add this data to the OpenStreetMap map to enable the creation of open source Level of Detail 1 (LoD1) city models. To this end, we have developed a Newton's-based method in optimization to accurately calculate building heights from single contributed photographs taken by citizens using smartphones or tablets. An Android app, OpenStreetHeight, is developed to carry out the experiments. Based on the various medium-height buildings that were photographed using the app and processed using the developed algorithms we received accurate building height values. When compared to reliable reference field measurements, the average height mean absolute error was 30 cm. Combined with the OpenStreetMap footprint vector data, we were able to produce an average LoD1 volume mean absolute error of less than 5%, satisfying the CityGML standard quality. This research presents a framework for a semi-automatic crowdsourced user-generated content calculation of OpenStreetMap building heights, and the creation of reliable and accurate LoD1 building models, as a first step to enhance the already established 2D OpenStreetMap map infrastructure to the 3D domain. This enables expanding the use of OpenStreetMap as a comprehensive and detailed representation of our urban environment for various environmental and sustainable applications and analyses.

## 1. Introduction

3D city models are valuable and useful for many experts using this spatial information for a wide array of environmental and sustainable analyses and services, including simulations and facility management, emergency responses and rescue operations, architecture, city planning, and digital tourism – to name a few. Perhaps one of the most important features in 3D city models are the buildings, since they serve as a major geospatial element in many applications, such as the propagation of noise and air pollution through cities, shadowing and microclimate, and the estimation of real estate taxes. Currently, to produce detailed and accurate 3D city models conventional authoritative production

processes are used, mostly relying on advanced and expensive technologies, such as aerial photogrammetry and laser scanning, which are lengthy, costly and require the involvement of experts in the field.

In recent years, there has been an explosion in interest in using the Web to create, assemble, and disseminate user-generated geographic information provided voluntarily by individuals, where the most prominent Internet map information source is OpenStreetMap (OSM). OSM is one of the most popular data sources existing today of crowdsourced Volunteered Geographic Information (VGI) that involves participatory mapping, with more than 5 Million registered contributors, and more than 335 Million mapped building footprints. This mapping project empowers citizens to create public domain global maps, which are

\* Corresponding author.

E-mail addresses: [seliana@campus.technion.ac.il](mailto:seliana@campus.technion.ac.il) (E. Bshouty), [shafir@campus.technion.ac.il](mailto:shafir@campus.technion.ac.il) (A. Shafir), [dalyot@technion.ac.il](mailto:dalyot@technion.ac.il) (S. Dalyot).

available for everyone everywhere. This notion is led by the concept that humans are acting as sensors, especially in urban regions, contributing to the valuable coverage of geographic data, leading to an increase in geospatial information for urban data and environmental management.

The majority of data collected by the VGI-driven communities is being used for creating two-dimensional maps, resulting in usage that is mostly restricted to the 2D (planar) domain. Although efforts are made to generate and visualize 3D models of OSM data, e.g., OSM-3D<sup>1</sup> and OSM Buildings,<sup>2</sup> these are generally limited to several major cities, focusing on visualization purposes with no standardized usage for exchanging urban city and building models. Even the Digital Terrain Model (DTM) used in OSM relies on the authoritative-produced Shuttle Radar Topography Mission (SRTM) infrastructure, although the OSM contributors actively collect their geospatial location, as in GNSS trajectories. Preliminary efforts are beginning to be developed to add the height data and augment OSM, but these are still limited.

2D footprints of building features are mapped with increasing accuracy and detail in OSM, mostly from satellite-based orthoimages base-maps and on-site mobile consumer GNSS sensor observations. Accordingly, building height values are almost impossible to determine, therefore only sporadically inserted into the OSM map. Thus, it is very rare to find buildings in OSM that store height information (data key). As of February 2019, according to the OSM statistics,<sup>3</sup> only 3% of the 335 million building footprints in OSM have the 'height' data key (and 3.5% use the 'levels' data key), mostly concentrated in Europe and North America.<sup>4</sup> These data keys are mostly inserted while relying on existing authoritative data or from on-site approximated observations, e.g., by counting the number of building floors, which can be very inaccurate. Other methods include the use of remotely-sensed data, such as using shadow detection to calculate buildings heights. That is, the usage of the relevant building height keys in OSM should be promoted to improve the overall spatial modeling and information, and more importantly - to augment the OSM 3D building models to create comprehensive 3D city models.

In most OSM on-site observations, contributors use their embedded smartphone GNSS sensor to map the building 2D footprint, overlooking the building height. To this end, we aim to use the embedded smartphone camera sensor, and use a single contributed photograph together with the existing mapped OSM vector data to precisely calculate the photographed building height. This value is automatically inserted into the OSM database, enabling the creation of fundamental building models (Level of Detail 1, LoD1) as part of a comprehensive public domain 3D city models. The examination of Newton's method in optimization is made in this research to accurately calculate the building heights from a single contributed image photographed by citizens via smartphones or tablets.

The approach presented here handles the problem of calculating the building's height by relying on the building footprint vector data that exists in the OSM database to optimize and match the corresponding building depicted in the photograph. To do so, several geometric processes are developed, relying on the photograph and camera parameters (location, azimuth, focal length, and camera sensor size). The semi-automatic implementation of Newton's method in optimization is realized by finding the minimum reprojection error, while relying on the footprint data of all candidate OSM buildings in the vicinity of the photograph's geo-tagged location. The objective reprojection error function assumes that the 3D points in the real world coordinate system, and their matched 2D points on the

photograph plane are known, such that we can calculate the minimum reprojection error.

To analyze and evaluate our approach and algorithms, we have developed an android app – OpenStreetHeight. Various medium-height buildings were photographed using the app, and processed with the developed algorithms. The results achieved by implementing our methodology are very promising, validating our framework for a semi-automatic crowdsourced VGI-based calculation of OSM building heights, and the creation of OSM LoD1 building models. We believe that this is a first step to enhancing the already established 2D OSM map infrastructure to the 3D domain.

## 2. Related work

### 2.1. 3D city models

The creation of 3D building and city models is mostly a tedious task. Generating accurate and detailed models is still fairly expensive in both time and money, where efforts are undertaken towards an automatic model generation by using different types of data sources, e.g., airborne Light Detection And Ranging (LiDAR) (Kada & McKinley, 2009; Maas & Vosselman, 1999). Image-based reconstruction methods using Unmanned Aerial Vehicle (UAV) lead to robust and accurate 3D models that are comparable to those of the LiDAR approaches (Wefelscheid, Hänsch, & Hellwich, 2011). In Vanegas, Aliaga, and Beneš (2010), multiple images downloaded from the Internet were used to create a detailed 3D building model, while in Cheng, Keller, and Kumar (2008), a single oblique image taken by UAVs was used for building 3D reconstruction, using geometric relations and attributes of a building, combining it with reference coordinate information retrieved from 2D GIS databases. An automatic generation of 3D city models can also be done by a combination of using laser profiler data, 2D digital map and aerial images (Wang, You, & Neumann, 2007). LoD1 building models are also obtained by using DSMs derived from calibrated satellite stereo-pair images with high spatial resolution (e.g., QuickBird2 and WorldView2), reaching a spatial accuracy of several meters (Duan & Lafarge, 2016). Although the abovementioned solutions produce reliable and accurate 3D models, they still require fairly expansive technologies, and take a lot of processing time handled by experts in the field; thus, these are commonly not available in the public domain, and integrating them with volunteer-centered campaigns is principally not practical.

### 2.2. 3D VGI

The contribution of new data to OSM can be made in different ways. The simplest one is through the digitization of objects (such as building footprints and road centrelines) that are visible on openly licensed satellite-based orthoimages base-maps. While it is very simple to add new data, mostly planar (although distortions, e.g., slanted buildings, and aliasing effects may occur), allowing the contributors to map remote places, the attributed height data is mostly missing, since it is basically impossible to retrieve and edit data that is related to the 3<sup>rd</sup> dimension (Over, Schilling, Neubauer, & Zipf, 2010). Accordingly, building heights are common attribute data that is missing in OSM. This is also the case with topographic models (DTMs), whereas OSM uses the existing SRTM databases, and not the topographic information that can be generated by processing the contributed trajectory data, as proposed in Massad and Dalyot (2018), which use a 2D Kalman filter to create DTMs from A-GPS trajectory observations.

Typically, the contributor will provide building attribute data that is easy to observe and collect, using the existing OSM tags, for example, address name and building type (Goetz, 2013), such that the height data is mostly missing. The amount of building features in some areas surpassed the number of streets, and nowadays there are > 335 million building footprints, where according to Goetz and Zipf (2013) their

<sup>1</sup> <http://www.osm-3d.org/map.htm>.

<sup>2</sup> <https://osmbuildings.org/>.

<sup>3</sup> <https://taginfo.openstreetmap.org/keys/building#combinations>.

<sup>4</sup> <https://taginfo.openstreetmap.org/keys/height#map>.

number increases by 1% monthly. The positional accuracy and completeness of these features is mostly high, having a positional offset value of several meters that is frequently the result of the oblique distortion existing in the satellite imagery used for the building digitization (e.g., Brovelli & Zamboni, 2018; Fan, Zipf, Fu, & Neis, 2014; Hecht, Kunze, & Hahmann, 2013). OSM does not yet support advanced types of features, such as Google Maps' street-view images, although Mapillary, which is a service for crowdsourcing street-level photographs using smartphones and computer vision, had in 2017 more than 200 million geotagged street-level photographs (Juhász & Hochmair, 2016; Mapillary, 2017), which can be shared in OSM with the focus of creating 3D information.

Relying on the fact that the positional accuracy of the mapped building features is high, attempts are made to exploit various OSM key-values to generate 3D information, namely the building height. The OSM-3D project, for example, created in 2008, is aimed at generating realistic 3D models (especially building models) by purely using VGI from OSM (Goetz & Zipf, 2013), as well as the KOSMOS WorldFlier and the OSM3D (Kolbe, 2009). OSM Buildings<sup>2</sup> is another example, using JavaScript library, but mainly for visualization purposes of OSM building geometry on 2D and 3D maps. Attempts are made to generate LoD1 building models by extruding the building footprint according to its height, usually the median or maximum of all elevation samples located within the building footprint (Ledoux & Meijers, 2011). If such data is not available, the key 'building: levels' can be used as an alternative to approximate the building height (multiplying the number of levels/floors with an average floor height value of 3.5 m) (Goetz & Zipf, 2012); still, it was found that quite commonly this value is incorrect (Dorn, Törnros, & Zipf, 2015). In Biljecki, Ledoux, and Stoter (2017), the LoD1 building heights were predicted using a combination of the buildings' attributes (building use, year of construction and number of levels above ground) together with machine learning techniques. Although the accuracy of this method was relatively high, with a mean height absolute error of 0.8 m, yet it is very difficult to carry out this method with user-generated data only, due to the absence of the needed attributes and the heterogeneity of different areas. Another way to predict the height of buildings is to use the maximum allowed height value as prescribed in the local building regulations, together with other architectural principles (Chen & Norford, 2016); yet, for most countries, there is no such corresponding dataset in the public domain.

The building height can also be estimated by using remotely-sensed high spatial resolution (< 1 m) data to calculate the length of its shadow, and the solar altitude value at a given location, date and time at which the image was captured (Peeters & Etzion, 2012; Peeters, 2016). Adeline, Chen, Briottet, Pang, and Paparoditis (2013) used high-resolution aerial imagery, with the NIR and RGB channels, to detect shadow in various dense urban areas, showing good results mostly in the range of 80–90% when compared to reference datasets. Ok (2013) used single very-high-resolution multispectral satellite imagery to detect buildings using shadow information and graph cuts. The author showed qualitative results, with mostly 80–90%  $F_1$  scores for building detection. Still, when using medium spatial resolution satellite imagery ( $\sim 5$  m), it was found that using shadow analysis worked well for relatively tall buildings, where the height accuracy of this method can be limited, reaching up to 13 m (Lee & Kim, 2013; Shao, Taff, & Walsh, 2011).

To sum, although the OSM 2D building footprints are detailed and accurate, the missing height data still prevents the creation of detailed LoD1 building models, as a first step towards the creation of VGI-driven public domain 3D city models. Thus, additional techniques to retrieve accurate 3D information are still required, techniques that will rely on and augment the existing practice embraced today by the OSM contributors of on-site observations – instead of relying on remotely-sensed data.

### 3. Methodology

#### 3.1. Input data

Several assumptions are made in our approach in relation to the user-contributed photographs:

- 1) They present a perspective view of a single building under the Manhattan-World assumption, i.e., the scene (the photographed object) has a natural Cartesian {x, y, z} coordinate system, and the building appears in the photographs in its entirety: two building facades and its roof; and,
- 2) The photographs are close to being horizontal/vertical in respect to the ground (zenith), with an acceptable angular rotation difference of up to  $\pm 10^\circ$  in the tilt and the roll angles.

The data required to carry out the process are: a) the camera location (latitude and longitude coordinates, acquired from the embedded smartphone GNSS sensor); b) the azimuth angular value of the photograph in decimal degrees (usually measured by the embedded smartphone or camera digital compass sensor); c) the camera's model and focal length in millimeters. These data are retrieved from the metadata listed in the EXIF file (Exchangeable Image File Format) stored on the device, and are extracted automatically from the photographs. The smartphone's GNSS planar location accuracy is estimated at  $\pm 10$  m (Petovello, 2014), and the smartphone's compass accuracy is estimated at  $\pm 5^\circ$  (Hölzl, Neumeier, & Ostermayer, 2013); these are nominal accuracy values, which are acknowledged today for most modern smartphones and tablets equipped with these sensors, measured under normal conditions.

#### 3.2. Retrieval of candidate OSM buildings

Since the optimization process relies on the OSM building footprint vector data, we find the candidate buildings that can be matched to the photographed building. The process relies primarily on both the photograph location and direction data to calculate a Field of View (FoV) polygon visible from the camera location, and retrieve the candidate buildings that fall inside it. The FoV of the photograph, which describes the scene captured in front of the camera by the camera's sensor, is calculated relying on the camera focal length and sensor size values. The FoV polygon is defined by six vertices: P<sub>1</sub> to P<sub>6</sub>, depicted in Fig. 1. The coordinates of the three tangent points P<sub>1</sub>, P<sub>2</sub> and P<sub>3</sub>, and the points P<sub>4</sub>, P<sub>5</sub>, and P<sub>6</sub>, are calculated using Eq. (1). Parameter values are [ $\varphi_{GNSS}$ ,

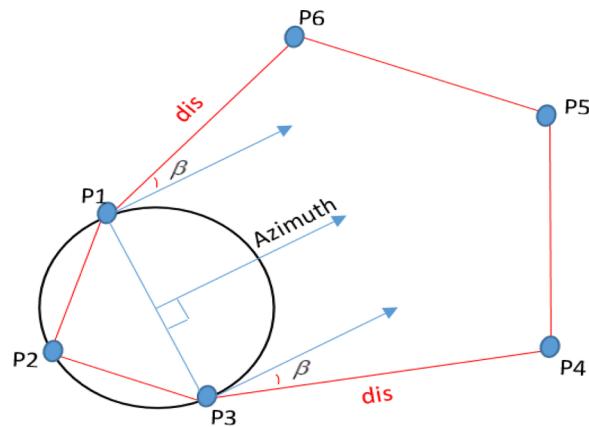
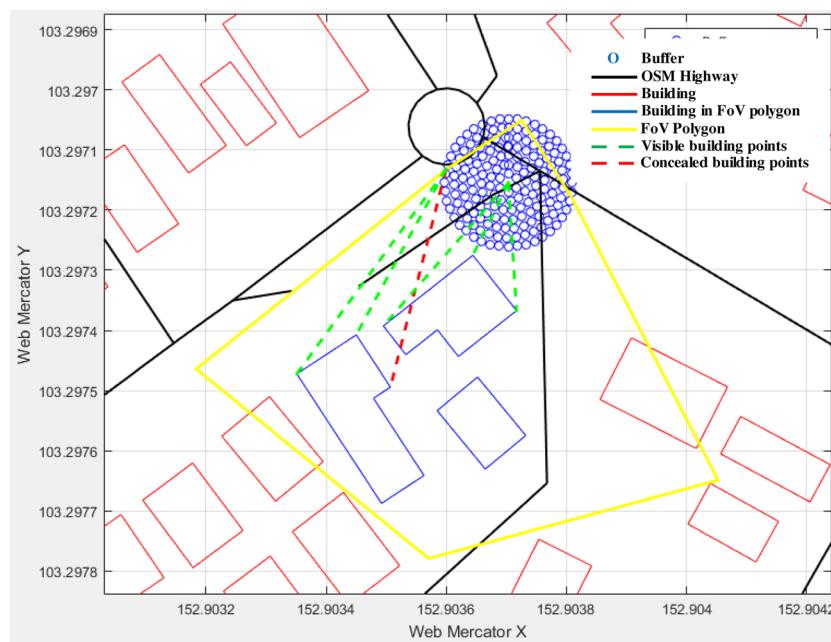


Fig. 1. The schematics of the FoV polygon (in red) calculation that is based on the measured values. The circle defines the possible camera locations, taking under consideration errors in position and azimuth. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



**Fig. 2.** An example of the FoV search polygon (in yellow) with several building footprints (in blue) falling inside its area. In this example, only one building footprint is considered a candidate (the closest footprint to the presumable camera location). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$\lambda_{GNSS}$ ] for the camera's location,  $dis = 11.7$  m (handling location accuracy errors of the GNSS and OSM footprint data) for points  $P_1, P_2$  and  $P_3$ , and  $dis = 100$  m (assumed visibility distance in urban areas) for points  $P_4, P_5$  and  $P_6$ .  $\theta = [\alpha - 90^\circ, \alpha + 180^\circ, \alpha + 90^\circ, \alpha - 90^\circ, \alpha - \beta, \alpha + \beta]$ , respectively, for all six points, where  $\alpha$  is the measured azimuth value, and  $\beta$  is the calculated view angle from Eq. (2).

$$\begin{aligned} \varphi_{Pi} &= \arcsin\left(\sin(\varphi_{GNSS}) \cdot \cos\left(\frac{dis}{R_{earth}}\right) + \cos(\varphi_{GNSS}) \cdot \sin\left(\frac{dis}{R_{earth}}\right) \cdot \cos(\theta)\right) \\ \lambda_{Pi} &= \lambda_{GNSS} + \arcsin\left(\frac{\sin(\theta) \cdot \sin\left(\frac{dis}{R_{earth}}\right)}{\cos(\varphi_{GNSS})} + \pi\right) \cdot MOD[2\pi] - \pi \end{aligned} \quad (1)$$

$$\beta = \arctan\left(\frac{\text{sensor\_width}}{2 \cdot \text{focal\_length}}\right) + 5^\circ \quad (2)$$

An example of this process is depicted in Fig. 2. The assumption is that at least two orthogonal building footprint lines should fall inside the FoV (following the requirement that at least two building facades are visible in the photograph), while checking that the corresponding three consecutive nodes forming these lines are visible from the camera position (green dashed lines). This, as long as these lines do not intersect any building footprint lines (red dashed line) within the FoV. If true, as in the building depicted in Fig. 2 that is the closest to the camera possible locations circle in the FoV (yellow polygon), which has two orthogonal building footprint lines that are not intersecting other buildings in the FoV, the OSM ID of the building is retrieved and stored as a candidate building. This process is presented in the first part of the pseudocode depicted in Fig. 6.

### 3.3. Calculating the photographed building height using Newton's method in optimization

The calculation of the photographed building height is achieved using Newton's method in optimization. This is achieved by finding the minimum projection error between the 3D points in the real world and the 2D points that define the building corners in the image in the camera's coordinate system. This is realized while relying on the footprint vector data of all candidate OSM buildings retrieved in the previous stage. The camera is described by the usual pinhole camera

model, with a projection from the world coordinates system [X, Y, Z] to the image coordinates [u, v], depicted in Eq. (3), where K is the camera  $3 \times 3$  intrinsic matrix, R is the  $3 \times 3$  orthonormal matrix representing the camera's orientation, and T is the  $3 \times 1$  vector representing the camera position. In reality, the camera can exhibit significant lens distortions, which can be modeled as a  $5 \times 1$  parameter vector consisting of radial and tangent distortion coefficients. Here, we assume that the camera has no significant lens distortions, or that the photographs are warped to eliminate these values. Accordingly, the general pinhole camera model,  $P = KR[I| - T]$ , has 9 degrees of freedom: 3 for the K matrix (including the focal length f, and the camera principle point: [px, py]), 3 for the Euler angles in the rotation matrix R, and, 3 for the camera position T.

We use only 6 degrees of freedom (R, T), under the assumption that K is fixed, the focal length is known from the EXIF data, and the principle point is equal to the center of the image. Then,  $\lambda$  is a scalar that according to Hartley and Zisserman (2003) can be interpreted as the depth of the 3D point in the real world from the camera center T in the direction of the principal ray.

$$\begin{aligned} \lambda \cdot \begin{bmatrix} a1 \\ a2 \\ a3 \end{bmatrix} &= \begin{bmatrix} f & 0 & px \\ 0 & f & py \\ 0 & 0 & 1 \end{bmatrix} \cdot R(rx, ry, rz) \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{\text{World}} + \begin{bmatrix} tx \\ ty \\ tz \end{bmatrix} \\ (u)_{\text{image}} &= \begin{bmatrix} a1/a3 \\ a2/a3 \end{bmatrix} \end{aligned} \quad (3)$$

The solution of an optimization problem is a set of possible variable values for which the objective function is assumed as an optimal value; in a mathematical term, optimization usually involves the maximizing or minimizing of the objective function. Newton's method in optimization converges quadratically to  $x^*$  if the initial solution  $x^0$  is sufficiently close to  $x^*$ , and the Hessian matrix (H) is a positive definite (assuming that H is non-singular at each iteration). Moreover, if H is not a positive definite, the search direction is not guaranteed to be in a descent direction, which is critical in finding the minimum of a function. If H is not a positive definite, the modified Cholesky factorization algorithm is used (Gill, Murray, & Wright, 1981). The search direction  $\Delta x$  is computed by solving the new linear system, depicted in Eq. (4), where E is 0 if  $A \in R_{n \times n}$  is a positive definite, otherwise, E is a nonnegative diagonal matrix, and g is the gradient vector. This method



**Fig. 3.** An example of the 6 building 2D (in pixels) corner points (blue dots) on the photograph used in the optimization process. A and B depict the footprint values. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

keeps the search direction close to Newton's direction, and is numerically stable when computing the search direction.

$$\Delta x = (H + E)^{-1} g_k \quad (4)$$

The objective function relies on the reprojection error function, outlined in Eqs. (3) and (5). Assuming that the 3D points in the world coordinate system and their corresponding 2D points on the photograph plane are known, we can measure the reprojection error, depicted in Eq. (5). There exist six parameters for the optimization, which are the angles of the rotation matrix R, the camera position vector T, and the number of points N. The goal is to find the height value of the corresponding building (from the candidate building footprint list) that gives the minimum reprojection error - meaning that we aim to find the optimal parameters for R and T.

$$\Delta = \begin{bmatrix} u \\ v \end{bmatrix}_{\text{Image}}^{\text{Known}} - \begin{bmatrix} u \\ v \end{bmatrix}_{\text{Image}}^{\text{Projected}}$$

$$\sum_i^N \Delta u^2 + \Delta v^2$$

$$f(R, T) = \frac{\sum_i^N \Delta u^2 + \Delta v^2}{N} \quad (5)$$

The 3D points' matrix is built using the OSM candidate building vector footprints according to six 2D points that represent the building corners in the photograph, as in the example depicted in Fig. 3. A and B are the footprint lengths in meters retrieved from the OSM vector data for all candidate buildings, while h is an array of possible building height values. During the implementation, h is iterated in a specified interval (in meters). The local Cartesian coordinate system is created under the assumption that the camera location is at [0, 0, 0] and [A, h, B] are the axis directions (Manhattan-World). Due to the fact that the photographs show a perspective view of the buildings, the 3D points are rotated by 45° in the horizon plane of AB (around the h axis), and translated with the horizontal distance between the EXIF camera location and the middle corner point of the building, depicted as distances T<sub>1</sub>, T<sub>2</sub>, and T<sub>3</sub> in Fig. 4. The 2D points in the pixel image plane (photograph) are retrieved by manually marking them after taking the photograph.

For each iteration, the value of h (building height) is updated, while A and B are fixed. For each h, we calculate the optimal values of R and T using Newton's method in optimization (the initial values are set to 0), and measure two error estimators:

- The reprojection error in pixel value relying on the values of  $\Delta u$  and  $\Delta v$  in Eq. (5); and,
- The average standard deviation error for  $\lambda$  in Eq. (6), which estimates how 'close' the 3D points are to the rays.

$$3D\_point_{new} = R_{opti} \cdot 3D_{point} + T_{opti}$$

$$\lambda_{3Xn} = \frac{3D_{point}_{new}^l}{[u_{image}^i, v_{image}^i, f]}$$

$$\lambda_{3Xn}^{norm} = \frac{\lambda}{|\lambda|}$$

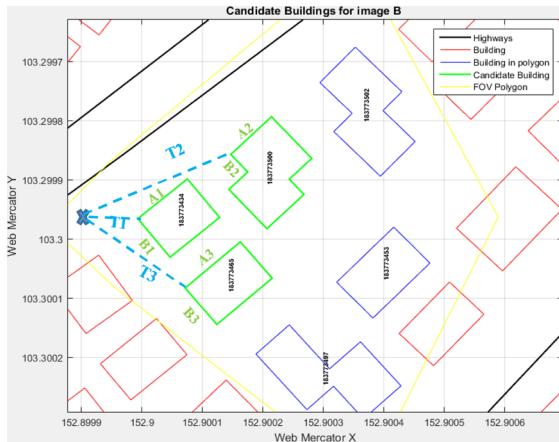
$$SD\_error = \text{means}\left( SD\left( \lambda_{3Xn}^{norm} \right) \right) \quad (6)$$

After going over all the h array values for all OSM candidate building footprints, the h value that gives the lowest (optimal) value for each error estimator is chosen as the optimal height for the analyzed building, as depicted in Fig. 5. In case both error estimators result in a different h value (mostly  $\pm 1$  h interval), the resulting optimal h value is calculated based on the average of both. At the end of this process, the building that gives the lowest error value in pixels, with a horizontal optimal translation value of the camera that is below the given threshold (11.7 m), is chosen as the corresponding photographed building. Its calculated optimal height value h is updated in the OSM database with building.height = h. The pseudocode of the complete process is depicted in Fig. 6.

## 4. Experimental results

### 4.1. OpenStreetHeight smartphone application

The field experiments were carried out using an Android-based



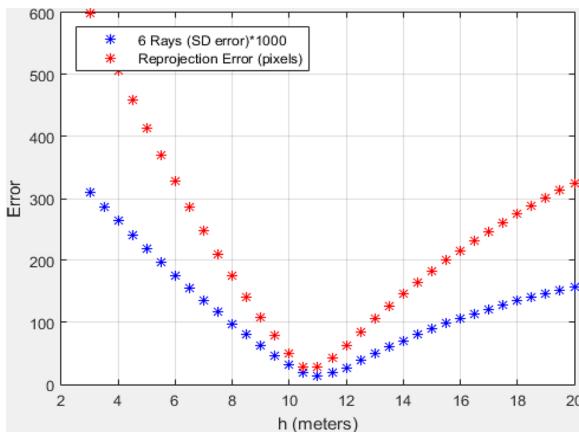
**Fig. 4.** The footprint vector data ( $A_i$ ,  $B_i$ ) for all candidate OSM building footprints, and the translation vectors  $T_1$ ,  $T_2$ ,  $T_3$  for the 3D point coordinate system.

application, OpenStreetHeight (OSH), which was developed specifically for this research, that includes all the algorithms required to carry out the height calculation process. The technical details related to the development and implementation of the OSH mobile application are detailed in the Appendix A.

The workflow for obtaining the building heights corresponds to the consecutive algorithm steps presented in the methodology chapter. In general, after a photograph is taken by the contributor, all arguments required for the height calculation function are retrieved automatically from the EXIF file (Section 3.1). The only requirement from the user is to mark the six building corners on the photograph. Fig. 7 (left) depicts the OSH User-Interface and an example of the building corners marking (showing building #8 from Table 1). After that, the candidate footprint building list is generated, and Newton's method in optimization is implemented, calculating the optimal building height. The user can then approve the calculated value, depicted in Fig. 7 (right), before it is inserted to the OSM database for the photographed building OSM-ID.

#### 4.2. Building height calculation and evaluation

Twelve medium-height buildings - up to 40 m, 2–8 stories high - having at least two orthogonal façades that match the Manhattan-World assumption were used for evaluating our approach and developments. The reference building footprint and height values were measured using



**Fig. 5.** An example of the iterative optimization process for a single OSM candidate building: the red asterisks correspond to the reprojection error in pixels, and the blue asterisks correspond to the  $\lambda$  standard deviation error. In this example, the lowest error value for both corresponds to an  $h$  value of 11 ms. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

#### Pseudocode

```

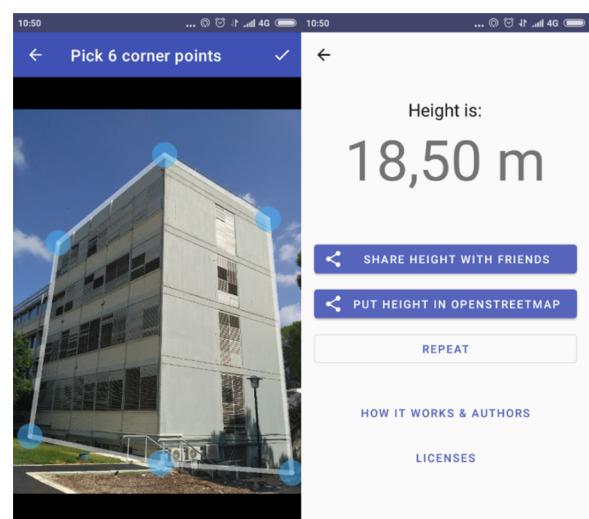
Input: EXIF data
L  $\leftarrow$  Circular buffer with radius: 11.7 m, center:  $[\varphi_{GPS}, \lambda_{GPS}]$ .
P  $\leftarrow$  Polygon of 'Field Of View'.
For each way in OSM rectangle:
    If way.tag is building & inside P :
        Else if 2 orthogonal lines can be seen from L :
            Do: Add this building to candidate list N.
    End for
For each building  $b_n$  in N:
    Build the 2D point matrix.
    For  $h_0=3$  to 20:
        Build the 3D point matrix.
        Rotate the 3D points with  $45^\circ$ .
        Translate the 3D points with distance  $T = |b_n - \text{camera}|$ .
         $x_0 \leftarrow [0,0,0,0,0]$ .
        Calculate the gradient 'g' and the Hessian 'H' variable matrices.
        While iterations < 1000 or  $|xDiff| > 1e^{-6}$  or  $\text{funcDiff} > 1e^{-9}$ 
            Do: Modified Cholesky for H.
            Calculate step size  $\alpha_k$ .
             $x_{k+1} \leftarrow x_k - \alpha_k H_k^{-1} g_k$ .
            Update H & g.
        End while
        Measure the Reprojection_Error in pixels.
        Measure the average Standard Deviation  $\bar{\sigma}$  for  $\lambda$ .
         $h_{k+1} \leftarrow h_k + 0.5$ 
    End For
     $h_1 \leftarrow$  Optimal height with the lowest Reprojection error of  $b_n$ .
     $h_2 \leftarrow$  Optimal height with the lowest  $\bar{\sigma}$  of  $b_n$ .
     $h_{optimal} \leftarrow$  Average of  $h_1$  &  $h_2$ .
    If  $\sqrt{T[1]^2_{optimal} + T[3]^2_{optimal}} > 11.7$  :
        Remove building  $b_n$  from N.
    End for
bmatched  $\leftarrow b_n$  with the lowest Reprojection_Error.
Set  $b_{matched}$ .height  $\leftarrow h_{optimal}$  in OSM.
End

```

**Fig. 6.** The pseudocode of the complete process.

a total station with an accuracy of  $\pm 5$  cm; these values are used for the accuracy evaluation. The buildings were photographed using two different smartphone models: 1) Samsung S7, with a smartphone camera having a full-size resolution of  $4032 \times 3024$  pixels, and a focal length of 4.2 mm, which is equal to 2971 pixels according to the camera sensor model (S5K2L1); 2) Xiaomi Redmi 3S, with a smartphone camera having a full-size resolution of  $4208 \times 3120$  pixels, and a focal length of 4.23 mm, which is equal to 3100 pixels according to the camera sensor model (S5K3L8). The EXIF data of the photographs were automatically extracted.

The complete process is outlined for two buildings, with their photographs depicted in Fig. 8. The candidate OSM buildings were



**Fig. 7.** OpenStreetHeight User-Interface: marking of photographed building corners (left) and the optimal building height value calculated by the application (right).

**Table 1**

The optimization processes and error estimators results for all analysed photographs.  $\Delta H_i$  depicts the height difference value between the reference building height value and the calculated one.

Photograph	OSM Building ID	Reprojection error (pixel)	$\Delta H_i$ (m)	$\lambda$ SD error (-)	$\Delta H_i$ (m)
#1	534119161	<b>75</b>	-0.16	<b>60</b>	0.34
	100108347	160	0.34	82	0.44
	100108348	200	-2.16	152	-2.26
#2	183773434	<b>54</b>	0.68	<b>61</b>	-0.82
	183773465	100	1.18	82	-0.82
#3	183773500	185	-3.32	156	-3.42
	184223028	<b>51</b>	-0.62	<b>25</b>	-0.12
#4	184223016	69	-0.62	35	-0.62
	184223016	<b>27</b>	-0.39	<b>14</b>	-0.39
#5	184223075	121	3.62	75	2.62
	184223080	55	-0.89	30	-0.89
#6	183803672	<b>53</b>	-0.79	<b>40</b>	-0.29
	183803587	61	-2.79	43	-3.29
#7	96580000	13	0.19	<b>53</b>	0.19
	97980065	<b>21</b>	-0.14	<b>31</b>	0.15
#8	96580000	<b>24</b>	-0.26	<b>39</b>	0.30
	67537797	<b>25</b>	-0.78	<b>31</b>	-0.44
#9	67591571	59	-2.69	45	-2.29
	102252768	31	-0.95	38	-0.84
#10	183773494	<b>30</b>	0.44	<b>22</b>	0.82
	183773439	35	0.85	41	0.92
#11	183773439	<b>61</b>	0.68	<b>58</b>	0.30
	183773459	97	1.06	78	0.82
#12	183773433	44	0.42	<b>55</b>	-0.72

automatically identified, depicted in Fig. 9; a visual field examination showed that for both examples the photographed building was amongst the OSM building candidate list. For each photographed building, the user manually marks the six building corners in the OSH app, and Newton's method in optimization is implemented. In the experiments, we used a minimum and maximum possible building height values of 3 and 40 m, correspondingly; these values were used since all the buildings in the area are below 40 m, and if required these values can be modified. The h interval (step size) of 0.5 m is used in the implementation; again, this value can be modified, although an empirical analysis showed that a smaller value of 0.1 ms did not improve the results. Fig. 10 depicts a graph showing the results of both measures used in the optimization for the two photographs depicted in Fig. 8: the reprojection error and the standard average deviation error, where both converged to a global minima error value near the same h value. The resulted calculated height values of the buildings are 10.75 m (left) and 11.25 m (right). A visual examination confirmed that for both photographed buildings the corresponding OSM buildings were correctly matched and used in the optimization process. Fig. 10 (left) depicts the

optimal T vector value that was calculated for the building in the optimization process. T ([x,y,z]) represents the translation vector of the camera from its initial position that was set as [0,0,0]. This result shows that based on the OSM building footprint geometry and values, there exists a horizontal distance in which the camera has moved, in this case, close to 7.5 m. To ascertain that we rely on the corresponding OSM building footprint, we use a threshold value of 11.7 m according to a local coordinate system. This value resembles the position errors derived from the GNSS sensor and OSM footprint data.

Table 1 depicts the results of all optimization processes for the twelve buildings that were photographed. The second column depicts the OSM building ID candidate list for each photograph. For each error estimators, reprojection and  $\lambda$  SD, we show the height difference value -  $\Delta H_i$ , between the reference building height value (measured with a total station) and the value calculated from the optimization process. For all photographs we can see that the minimum value for both estimators (in bold) – and accordingly the calculated optimal height values - correspond to the OSM building ID that was photographed, proving that the optimization process is robust and reliable. We can see that for some processes the reprojection error estimator produced better results (lower absolute  $\Delta H$  value) (photographs #1, #2, #7, #8, #10 and #12), while for some buildings the  $\lambda$  SD error estimator produced better results (photographs #3, #5, #9 and #11), and for the rest both estimators gave the same result (photographs #4 and #6). Accordingly, the mean absolute error (MAE) of the height difference value for all buildings is 0.46 m according to the reprojection error, and 0.41 ms according to the  $\lambda$  SD error. When the MAE of the height difference value is calculated based on the average value of both error estimator values, it is reduced to 0.30 m. Thus, we concluded that the final optimal height value will rely on the average value of both optimal height value calculations. The final optimal building height results for all the photographed buildings are depicted in Table 2. These results are very promising, presenting the high accuracy this optimization process can produce, which can reach several centimeters only - when compared to the accurate reference values.

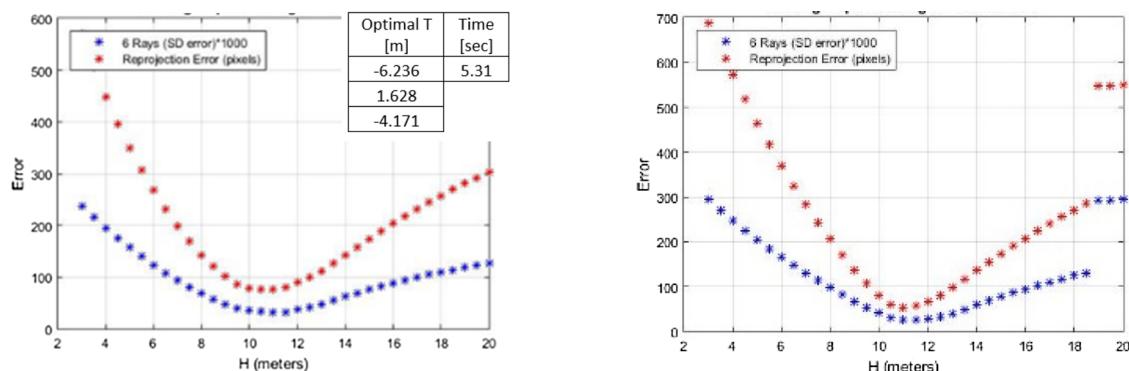
The calculated building height values were used alongside the OSM vector footprint values to produce LoD1 building models. When compared to the reference values measured using a total station, the MAE of the volume for all buildings was less than 4.0%. The qualitative assessment of this result is not straightforward, since in theory there is no commonly agreed standard in terms of LoD1 accuracy. For example, the CityGML (Gröger, Kolbe, Nagel, & Häfele, 2012) mentions an accuracy benchmark as a recommendation only: "...In LoD1, the positional and height accuracy of points should be 5 m - or less". Accordingly, and following the notions presented in Biljecki et al. (2017) and Tack, Buyukalih, and Goossens (2012), for example, where the accuracy of the 3D models is in the same error range - and even worse - than the accuracy calculated by our methodology, we have concluded that our



Fig. 8. The photographs of two of the buildings used in the experiments.



**Fig. 9.** Generating the candidate OSM building list for the two photographs: OSM buildings that fall inside the FoV (yellow polygon), candidate OSM buildings (green footprints), and corresponding photographed buildings (red arrow). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



**Fig. 10.** The height values calculated for the photographed buildings using Newton's method in optimization: X-axis represents the height value in h interval; Y-axis depicts the reprojection error in pixels (red asterisks), and the standard average deviation error ( $1000 \times \text{SD\_error}$ ) (blue asterisks).

approach is capable to produce reliable LoD1 building models, out-classing the commonly used traditional LoD1 model construction (e.g., satellite stereo-pair images with high spatial resolution). Moreover, the small difference of the volume values ascertains our assumption that the OSM footprint data in the study area are accurate, and can be used to generate LoD1 building models as a first step to generating more comprehensive city models.

To check the correlation between the accuracy of the OSM footprint

**Table 2**

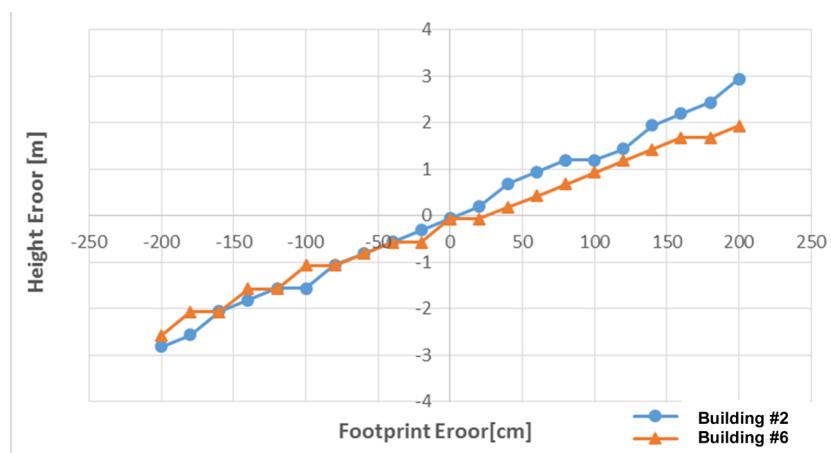
The optimal building height values calculated via Newton's method in optimization.

Building	Reference Measured Height (m)	Optimal Calculated Height (m)	Building Height Difference (m)
#1	10.66	10.75	0.09
#2	11.82	11.75	-0.07
#3	11.62	11.25	-0.37
#4	11.39	11.00	-0.39
#5	11.29	10.75	-0.54
#6	15.31	15.50	0.19
#7	12.24	12.25	0.01
#8	18.48	18.50	0.02
#9	31.11	30.50	-0.61
#10	7.87	8.50	0.63
#11	7.76	8.25	0.49
#12	7.90	7.75	-0.15

vector data and the building height calculation via the optimization process, the footprint data values of buildings #2 and #6 were modified with a defined error value, while the optimal height value was calculated based on these erroneous values. The results are displayed in Fig. 11. Observing the results, the two graphs behave similarly to the footprint error values, i.e., close to being linear: up to approximately 1 m error in A and B (footprint values), the height calculation results also change in approximately 1 m, while bigger error values of 2 m in A and B lead to height calculation change of close to 3 m. This shows that the location accuracy of the OSM footprint vector data affects the reliability of the building height calculation. Still, as was proved earlier, for most cases the OSM footprint data is accurate, and in cases where gross errors do exist, they lead to height errors of several meters, which although not optimal, are still less or in the range of other non-authoritative building height calculation methods that were presented in Chapter 2.

## 5. Conclusions and future work

This research presents a framework for a semi-automatic VGI-based calculation of OSM building heights from single contributed photographs, and the creation of LoD1 building models as a first step to enhancing the already established 2D OSM map infrastructure to the 3D domain. This is achieved by implementing Newton's method in optimization, which relies on the OSM footprint vector data, the building corners and the EXIF data of the photograph. This approach and the



**Fig. 11.** Correlation analysis of errors in the OSM footprint vector data and the resulting calculated building heights for buildings #2 and #6. The X-axis is the footprint error compared with the reference measured footprint, and the Y-axis is the optimal height value change (error). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

developed algorithms offer several advantages compared to LiDAR or photogrammetry, which are to date perhaps the most commonly used and accurate technologies for the creation of detailed 3D city models. The data produced by these technologies are mostly not available in the public domain, together with the fact that both are still not amalgamated with crowdsourcing, and using these technologies can be costly and require large processing times involving experts in the field. Alternatively, using contributed photographs can reduce these limitations quite considerably, relying on and augmenting the existing practice embraced today by the OSM contributors of on-site observations, to produce accurate results.

Experimental results of medium-height buildings show accurate building height calculation values when compared to the reference values, with an average height difference value of 30 cm for the analysed buildings. Moreover, for all photographs the corresponding OSM building was correctly detected and matched. When the calculated building height values were used to generate a LoD1 building model, the volume difference from the reference values was small, with an average volume difference of 4.0%. Future research is planned to analyze more complex building geometries and footprints with photographs that can be downloaded from the Internet. We also plan to improve the existing algorithms to handle the automatic detection of the building corner points in the photographs by developing a boundary detection algorithm, thus making the process fully automatic. Future research is also planned to develop a complementary process that solves the problem that is related to the fact that in dense built-up urban environments there might not be enough space near the photographed building to take a single photo that completely covers tall or wide buildings. Nonetheless, since a fairly large percentage of buildings in the urban setting suits the methodology definitions, the proposed implementation allows the OSM users to conduct on-site observations and map buildings in 3D with their smartphones.

Examining the CityGML standard with an accuracy benchmark of 5 m (or better), we conclude that it is feasible to generate reliable LoD1 models by implementing our approach. Whereas the quality of the output models is not as accurate as airborne LiDAR, the experimental results surpass the traditional LoD1 model construction (e.g., satellite stereo-pair images with high resolution), offering new perspectives in bringing the 2D OSM to the third dimension (3D OSM). Since the usage of 3D OSM is still limited, the outcome of this research can contribute to the creation of a general framework to overcome existing limitations, adding to the overall perception of using the public to create a comprehensive and more detailed representation of our environment. Namely, public domain 3D city models and 3D geographic information, which are essential for a variety of reliable environmental and sustainable applications, such as morphological-based analysis of urban environments and urban-climate studies.

## Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Appendix A. The OpenStreetHeight Smartphone Application

The OpenStreetHeight (OSH) application consists of front-end and back-end that communicate with each other via the HTTP protocol (i.e., POST requests only, body – JSON). The front-end is an Android application, developed using the Model-View-Presenter (MVP) architectural pattern. The back-end is a Java server application running on Linux OS, deployed in the cloud (virtual machine). The back-end application accepts HTTP POST requests only with JSON body and sends JSON object in return. We have decided to split the back-end from the front-end mainly to reduce the Android application size: Matlab Runtime is implemented to carry out all calculation functions in the back-end, which requires 3 GB of storage space, resulting in the front-end application that uses 40 MB only in its current configuration. This allows a relatively fast computation time of approximately 5 s to calculate the photographed building height value.

The building height calculation workflow corresponds to the consecutive algorithm steps discussed in the methodology chapter:

- 1 The contributor inserts his/her OSM authentication login credentials (i.e., username and password) to allow OSM editing.
- 2 The front-end UI input collects the arguments required for the height calculation function, including: a) the camera location in the real world (from the GNSS sensor); b) the camera azimuth (from the compass sensor); c) the camera parameters (focal length and sensor size); d) the photograph dimensions (length and width in pixels); d) the six building corners marked by the user on the photograph (in pixels).
- 3 The application then sends the abovementioned argument values to the back-end to:
  - a Obtain the list of the potential buildings from the OSM database.
  - b Select the building (OSM-ID) that corresponds to the photographed building according to the developed Newton's method in optimization algorithm.
  - c The building height is calculated, and the value is sent from the back-end and presented in the front-end (Android application) to

- the user.
- 4 If the calculated building height value is approved by the user, it is sent to the OSM database together with the OSM-ID (OSM WAY feature) of the building, and the height tag value of the building is inserted; accordingly, the OSM database is updated and the new information is shared in the OSM map.

## References

- Adeline, K. R. M., Chen, M., Briottet, X., Pang, S. K., & Paparoditis, N. (2013). Shadow detection in very high spatial resolution aerial images: A comparative study. *ISPRS Journal of Photogrammetry and Remote Sensing*, 80, 21–38.
- Biljecki, F., Ledoux, H., & Stoter, J. (2017). Generating 3D city models without elevation data. *Computers, Environment and Urban Systems*, 64, 1–18.
- Brovelli, M., & Zamboni, G. (2018). A new method for the assessment of spatial accuracy and completeness of OpenStreetMap building footprints. *ISPRS International Journal of Geo-Information*, 7(8), 289.
- Chen, K. W., & Norford, L. K. (2016). Workflow for generating 3D urban models from open city data for performance-based urban design. *Proceedings of the Asim 2016 IBPSA Asia Conference* (pp. 27–29).
- Cheng, P., Keller, J., & Kumar, V. (2008). Time-optimal UAV trajectory planning for 3D urban structure coverage. *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008. IROS 2008* (pp. 2750–2757).
- Dorn, H., Törnros, T., & Zipf, A. (2015). Quality evaluation of VGI using authoritative data—A comparison with land use data in Southern Germany. *ISPRS International Journal of Geo-Information*, 4(3), 1657–1671.
- Duan, L., & Lafarge, F. (2016). Towards large-scale city reconstruction from satellites. *Spatial information theory. Cognitive and computational foundations of geographic information science*. Cham: Springer International Publishing89–104.
- Gill, P. E., Murray, W., & Wright, M. H. (1981). *Practical optimization*.
- Goetz, M. (2013). Towards generating highly detailed 3D CityGML models from OpenStreetMap. *International Journal of Geographical Information Science*, 27(5), 845–865.
- Goetz, M., & Zipf, A. (2012). OpenStreetMap in 3D—detailed insights on the current situation in Germany. *Proceedings of 15th AGILE International Conference on Geographic Information Science* (pp. 24–27).
- Goetz, M., & Zipf, A. (2013). *The evolution of geo-crowdsourcing: Bringing volunteered geographic information to the third dimension. Crowdsourcing geographic knowledge*. Dordrecht: Springer139–159.
- Gröger, G., Kolbe, T. H., Nagel, C., & Häfele, K. H. (2012). OGC city geography markup language (CityGML) encoding standard, version 2.0. *OGC Doc. 12-019*.
- Fan, H., Zipf, A., Fu, Q., & Neis, P. (2014). Quality assessment for building footprints data on OpenStreetMap. *International Journal of Geographical Information Science*, 28(4), 700–719.
- Hartley, R., & Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge University Press.
- Hecht, R., Kunze, C., & Hahmann, S. (2013). Measuring completeness of building footprints in OpenStreetMap over space and time. *ISPRS International Journal of Geo-Information*, 2(4), 1066–1091.
- Hölzl, M., Neumeier, R., & Ostermayer, G. (2013). Analysis of compass sensor accuracy on several mobile devices in an industrial environment. *International Conference on Computer Aided Systems Theory* (pp. 381–389).
- Juhász, L., & Hochmair, H. H. (2016). User contribution patterns and completeness evaluation of Mapillary, a crowdsourced street level photo service. *Transactions in GIS*, 20(6), 925–947.
- Kada, M., & McKinley, L. (2009). 3D building reconstruction from LiDAR based on a cell decomposition approach. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(Part 3), W4.
- Kolbe, T. H. (2009). *Representing and exchanging 3D city models with CityGML. 3D geoinformation sciences*. Berlin, Heidelberg: Springer15–31.
- Ledoux, H., & Meijers, M. (2011). Topologically consistent 3D city models obtained by extrusion. *International Journal of Geographical Information Science*, 25(4), 557–574.
- Lee, T., & Kim, T. (2013). Automatic building height extraction by volumetric shadow analysis of monoscopic imagery. *International Journal of Remote Sensing*, 34(16), 5834–5850.
- Maas, H. G., & Vosselman, G. (1999). Two algorithms for extracting building models from raw laser altimetry data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 54(2–3), 153–163.
- Massad, I., & Dalyot, S. (2018). Towards the Crowdsourcing of Massive Smartphone Assisted-GPS Sensor Ground Observations for the Production of Digital Terrain Models. *Sensors*, 18, 898.
- Mapillary (2017). *Celebrating 200 million images—The mapillary blog*. Available online: <https://blog.mapillary.com/update/2017/10/05/200-million-images.html> (Accessed on 20 December 2018).
- Ok, A. O. (2013). Automated detection of buildings from single VHR multispectral images using shadow information and graph cuts. *ISPRS Journal of Photogrammetry and Remote Sensing*, 86, 21–40.
- Peeters, A., & Etzion, Y. (2012). Automated recognition of urban objects for morphological urban analysis. *Computers, Environment and Urban Systems*, 36(6), 573–582.
- Peeters, A. (2016). A GIS-based method for modeling urban-climate parameters using automated recognition of shadows cast by buildings. *Computers, Environment and Urban Systems*, 59, 107–115.
- Petovello, M. (2014). What are the actual performances of GNSS positioning using smartphone technology. *InsideGNSS*, 9, 34–37.
- Over, M., Schilling, A., Neubauer, S., & 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.
- Shao, Y., Taff, G. N., & Walsh, S. J. (2011). Shadow detection and building-height estimation using IKONOS data. *International Journal of Remote Sensing*, 32(22), 6929–6944.
- Tack, F., Buyuksalih, G., & Goossens, R. (2012). 3D building reconstruction based on given ground plan information and surface models extracted from spaceborne imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 67, 52–64.
- Wefelscheid, C., Hänsch, R., & Hellwich, O. (2011). Three-dimensional building reconstruction using images obtained by unmanned aerial vehicles. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(1).