EXTRACTING DIMENSIONS AND LOCATIONS OF DOORS, WINDOWS, AND DOOR THRESHOLDS OUT OF MOBILE LIDAR DATA USING OBJECT DETECTION TO ESTIMATE THE IMPACT OF FLOODS

S. Van Ackere [1,*], J. Verbeurgt [1], L. De Sloover [1], A. De Wulf [1], N. Van de Weghe [1], P. De Maeyer [1]

[1] Departement of Geography, Belgium - (samuel.vanackere, jeffrey.verbeurgt, lars.desloover, alain.dewulf, nico.vandeweghe, philippe.demaeyer)@ugent.be

KEY WORDS: FLIAT, flood impact assessment, location of doors and windows, threshold height, object detection, machine learning

ABSTRACT:

Increasing urbanisation, changes in land use (e.g., more impervious area) and climate change have all led to an increasing frequency and severity of flood events and increased socio-economic impact. In order to deploy an urban flood disaster and risk management system, it is necessary to know what the consequences of a specific urban flood event are to adapt to a potential event and prepare for its impact. Therefore, an accurate socio-economic impact assessment must be conducted. Unfortunately, until now, there has been a lack of data regarding the design and construction of flood-prone building structures (e.g., locations and dimensions of doors and door thresholds and presence and dimensions of basement ventilation holes) to consider when calculating the flood impact on buildings. We propose a pipeline to detect the dimension and location of doors and windows based on mobile LiDAR data and 360° images. This paper reports on the current state of research in the domain of object detection and instance segmentation of images to detect doors and windows in mobile LiDAR data. The use and improvement of this algorithm can greatly enhance the accuracy of socio-economic impact of urban flood events and, therefore, can be of great importance for flood disaster management.

1. INTRODUCTION

For a variety of applications, like the evaluation of the effect of (architectural) design, various construction methods, and engineering applications on the damage due to flood events, flood damage and risk assessment would benefit from the consideration of the distinctiveness of buildings [1]. In such an effective case-by-case analysis of damage to a building at micro level, building components that resist against flood impacts and are unique to each building need to be taken into account [2]. Therefore, acquiring the dimensions of doors and windows is, among other things, of high importance in flood risk assessment studies on micro level. The locations and dimensions of these open, weak spots in buildings are decisive factors in whether or not the water of a flood can easily penetrate, damage or destroy building contents, and affect inhabitants [1,3,4]. Moreover, the information of location and dimensions of doors and windows, and other openings can be taken into account when evaluating local flood protection (e.g., temporary barriers like sand bags).

On the other hand, in some cases, openings in load-bearing walls (which for example support the elevated building) are necessary to relieve the pressure of standing or slow-moving water against the structure (called hydrostatic loads) [5]. As a result of these openings, the flood water reaches equal levels on all sides of the construction and thus lessen the potential for damage caused by a difference in hydrostatic loads on opposite sides of the structure.

Although it is already possible to extract the dimensions of doors, windows and basement holes from Energy Performance Certificates (EPC) [6], extracting the exact location of these objects or weak spots from these documents is not possible. On the other hand, it is possible to extract the orientation of the normal vector of these doors and windows from EPC documents, thus making it possible to align these doors and windows on walls of the building with the same normal vector orientation. Moreover, information on door threshold dimensions, for example, cannot be extracted from EPC documents. Therefore, an algorithm that can detect the exact location of doors and windows adds enormous value to flood risk management and flood disaster risk reduction in the future.

1.1 Indoor Social Impact

Regarding the activity and place of the victims at the time of a flood event, research shows that a significant percentage of fatalities occur indoors [7–9]. Diakakis, M. (2016) conducted research indicating that from mortality numbers due to flood events in Greece, 14.8% of all victims passed away indoors [7]. Research conducted by Jonckman et al. (2009) showed that even a higher portion of fatal incidents occurred indoors as a result of Hurricane Katherina. In this case, the majority of victims (53%) passed away in individual residences [8]. Important to mention is that fieldwork showed that many of these residential buildings were unelevated or elevated less than three feet, single-story homes [8]. Although a portion of these victims died when their houses collapsed due to the powerful force of the flood, many others drowned in their home due to a high horizontal and rising flood velocity.

Flood water can penetrate through the weak spots of buildings (e.g., doors and windows), affecting inhabitants. Therefore, it is crucial to determine the locations and dimensions of these weak spots. It then becomes possible to estimate and assess the flood risk of inhabitants and to calculate the indoor flood characteristics (e.g., indoor horizontal flood velocity, vertical flood velocity, water depth and duration) with specific flood models.

In addition to determining the dimensions and location of doors, windows and other weak spots against the force of floods, considering the human impact is also essential for estimating the direct economic impact due to flood events. The dimensions and locations of doors and windows determine the indoor flood characteristics and thus the direct indoor economic impact when a flood permeates these areas. Moreover, the location of doors and windows and the height of door thresholds can exclude houses to be affected by a flood event and enabling emergency services to work in a more effective manner.

For example, a recent pluvial flood simulation conducted by engineering company Arcadis for a vulnerability study of the city of Ghent showed that 72-88% of buildings are, in reality, not affected by this specific pluvial flood event (with a return period of 20 years) when a door threshold for every door of 10 cm or 15 cm respectively is assumed (see Table 1).

	T20		T100		T20 (2050)	
Total number of affected houses	9 764	[%]	16 383	[%]	17 973	[%]
< 0.15 m	8 634	88.4	13 717	83.7	15 101	84.0
< 0.10 m	7 628	78.1	11 725	71.6	12 934	71.9

Table 1. Percentage not-affected by a pluvial flood event (for a return period of T20, T100 and T20 for the in situ situation of 2050) buildings due to door threshold consideration

1.2 LiDAR Data and the Point Cloud Extension

LiDAR (Light Detection And Ranging) is an optical remote-sensing technique that uses laser light to produce highly dense and accurate (x, y, z) measurements. Besides containing only x, y and z values, LiDAR sensors can capture dozens of other variables, such as intensity and return number, red, green and blue colour values and return times. Handling LiDAR data is a complex challenge due to the millions of rapidly produced points with large numbers of variables measured on each point by LiDAR sensors. This data must be stored efficiently while allowing quick and convenient access to the stored point cloud data afterwards.

Many Lidar Information Systems (LIS), which have a spatial relational database architecture as a core, have been developed over the past years in response to storing difficulties (e.g., Point Cloud extension in PostgreSQL [10], Oracle [11] …). For this research, the Point Cloud extension, together with the PostgreSQL database, is used. The Point Cloud extension, created by Blottiere P., stores point clouds into so-called patches of several hundred points each (see Figure 1) [10]. Instead of having a table with billions of points, the table is reduced to tens of millions of rows, which is more tractable.

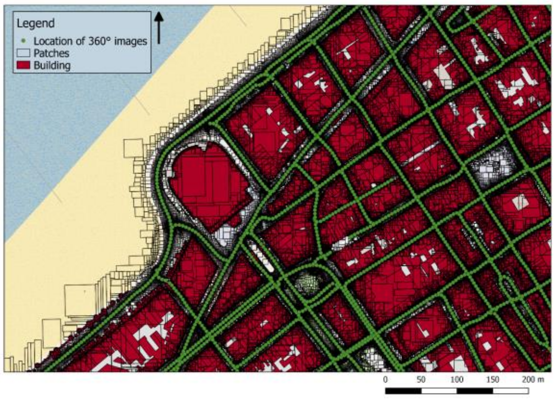


Figure 1. Point cloud of mobile LiDAR is stored in patches of 600 points, Ostend (Belgium)

PostgreSQL Pointcloud deals with all this variability by using a so-called schema document to describe the contents of any particular LiDAR point. Each point can contain several variables: X, Y, Z, intensity and return number, red, green, and blue values, return times, etc.

The schema document format used by PostgreSQL Pointcloud is the same one as used by the Point cloud Data Abstraction Library (PDAL) library [12]. The PDAL library is a C++ BSD library for translating and manipulating point cloud data quickly and fluently.

2. METHODOLOGY

2.1 Data Preparation

Although some research is conducted on running object detection and semantic segmentation on panorama images, in most scientific studies, spherical images are first converted into a less distorted format. 360° spherical panorama images are converted to cube boxes via the so-called cube mapping process. Cube mapping is a method of environment mapping that uses the six faces of a cube as the map shape, with every face of the cube consisting of undistorted, perspective images (up, down, left, right, forward and backward), whereas the equirectangular format is a single stitched image of 360° horizontally and 180° vertically. Because the cubic format suffers from less distortion than the equirectangular format, it becomes possible to detect objects more accurate.

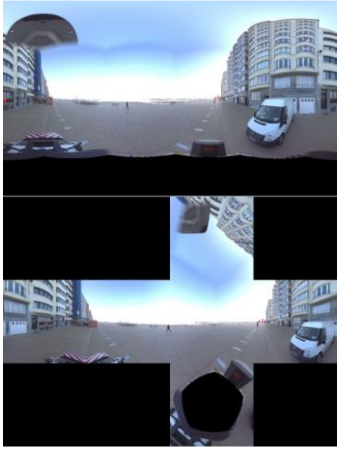


Figure 2. Cube mapping of spherical panorama images allows for more accurate detection of objects

For the case of Ostend, on average, 70,000 spherical images were first converted into cube boxes (see Figure 2).

In order to convert the equirectangular projection to cube box projection, the spherical coordinates are used. First, the pixel coordinates of the spherical image are normalised:

$$x = \frac{2i}{(w-1)} \quad (1)$$

$$x = \frac{2j}{(h-1)} \; or \; y = \frac{(1-2j)}{h} \quad (2)$$

depending on the position of the pixel

where (i, j) = pixel coordinates
h = height
w = width

Hereafter, the spherical pixel coordinates can be calculated with the following formulas:

$$\theta = x\,\pi \qquad (3)$$

$$\varphi = y\,\pi\,/\,2 \qquad (4)$$

These spherical coordinates (θ, φ) are turned into a unit vector (for the sphere with $r = 1$), by projecting these pixel coordinates onto a surrounding cube:

$$x = r \sin\theta \cos\varphi \qquad (5)$$

$$y = r \sin\theta \sin\varphi \qquad (6)$$

$$z = r \cos\theta \qquad (7)$$

where r = radius
$\theta \in [0, \pi]$
$\varphi \in [0, 2\pi]$

Based on these unit vectors, the cube boxes are created. Hereafter, for every cube box, four of the six faces are extracted as individual undistorted, perspective images (Figure 3).

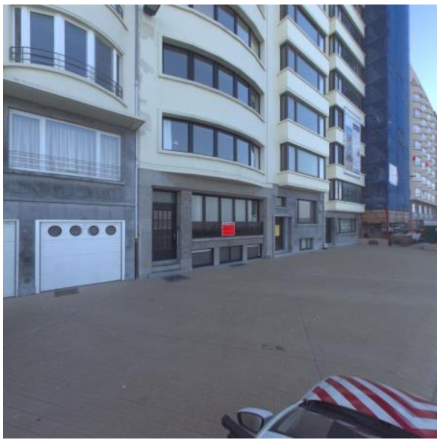


Figure 3. Example of an image extracted from the cube box after converting the 360° image

2.2 Detection of Doors and Windows

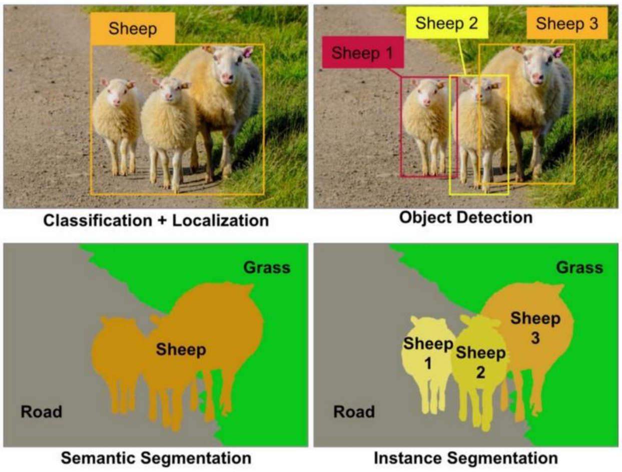
Over the past several years, a considerable amount of research has focused on the theme of object detection. Applications include face recognition [13], gesture recognition [14], semantic human activity recognition [15], vehicle and pedestrian detection for self-driving cars [16,17] and several other advanced, far-reaching applications.

In this time, door and window detection on images has also been studied extensively. Notably, approaches in scientific studies differ in the variability of the environment (e.g., indoor or outdoor) and the images and type of sensors they consider. Additionally, numerous studies try to find doors and windows based on the fact that these objects move, in contrast with static walls [18]. Although this methodology is highly effective, there are many more applications where doors need to be detected from its static, closed appearance.

The past two decades have seen a number of researchers who have sought to detect doors and windows using both visual information, whereby for many examples an additional remote sensing source of information is taken into account (e.g., sonar data, acoustic data, LiDAR data).

Because object detection alone is not enough (with a rectangle as output around the detected doors and windows, see Figure 4) to determine the location and the dimensions of doors and windows, segmentation is needed. Numerous studies have considered the problem of detecting the dimensions and locations of doors and windows by segmenting the pixels in images of building facades into different semantic classes.

After the door or window is detected (the location of the object on the image is found), it is possible to predict the best-fitted classification for every pixel, so that each pixel is labelled with the class of its enclosing object or region, so-called semantic segmentation (see Figure 4).



Classification + Localization

Object Detection

Semantic Segmentation

Instance Segmentation

Figure 4. Difference between classification and localisation, object detection, semantic segmentation and instance segmentation [19]

In order to calculate the dimensions and assess the location of doors and windows, pixel-wise masks for each object are needed, dividing each object with the use of instance segmentation.

Object detection or segmentation can be done by supervised machine learning approaches or unsupervised machine learning approaches. Supervised machine learning approaches use algorithms together with pre-defined extracted features to find instances of specific objects on new images [20,21]. These features are, for example, photometric and spatial statistical features, shape features (e.g., ratio of height to width) or contextual features. Unfortunately, supervised machine learning-based approaches are still prone to human error due to the manually labelled features on the pictures used during the training process of the model.

On the other hand, unsupervised machine learning approaches do not need predefined features to detect the object (e.g., a door or a window) or run a semantic segmentation. Instead, an artificial Neural Network (NN) automatically creates a model and defines the features or the definition of a door and window.

By labelling thousands and thousands of images, it becomes possible to train a neural network in detecting objects or creating semantic segmentation [22,23]. These NN approaches learn to perform tasks by considering examples without accounting for predefined features and generally without being programmed with any task-specific rules.

There are two different approaches when it comes to facade segmentation: top-down methods [24–26] and bottom-up methods [27–30]. The former method, top-down, uses shape grammar to parse a facade into a set of production rules and element attributes [25]. This method starts with the philosophy that building facades are highly structured due to architectural design choices and construction constraints [24–26]. For example, a door will often only appear on street-level, and windows are not placed randomly but typically at the same height as a vertical ordering. Therefore, this method searches for the best possible derivation of every object, using a specific shape grammar. Unfortunately, until now, grammar-based methods have achieved poor accuracy of pixel-wise classification [25,31]. Moreover, this method is time inefficient during training and inference [32].

On the other hand, bottom-up methods classify pixels, taking context (e.g., neighbouring pixels) into account [28,29]. This method employs a pipeline architecture in which each part of the pipeline tries to correct wrongly classified pixels or optimise the segments created by previous iterations. Currently, this method is more efficient and of a higher quality compared to the top-down method.

In recent years, much progress has been made on object detection, mainly by the development and use of convolutional neural networks (CNNs). We can consider Faster R-CNN (region-based convolutional neural networks) [33], R-FCN (region-based fully convolutional network) [34] and SSD (single-shot detector) [35]. Overall, the best instance segmentation algorithm depends on desirable accuracy versus speed and its necessary memory (see **Fout! Verwijzingsbron niet gevonden.**) [36]. Important to note is that a false positive object detection could indicate, in this case, a higher socio-economic damage which does not match the reality.

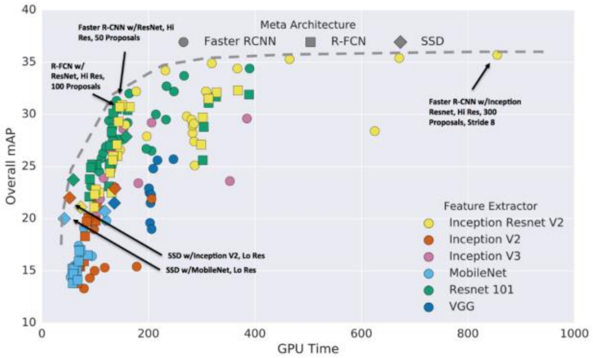


Figure 5. Accuracy versus speed for an instance segmentation algorithm [36]

As aforementioned, multiple algorithms can be used to train and run an instance segmentation on perspective images (converted from spherical images). For example, He, K. et al. (2017) developed the Mask-RCNN, which detects objects in an image while simultaneously generating a high-quality segmentation mask for each instance (see Figure 6) [37].

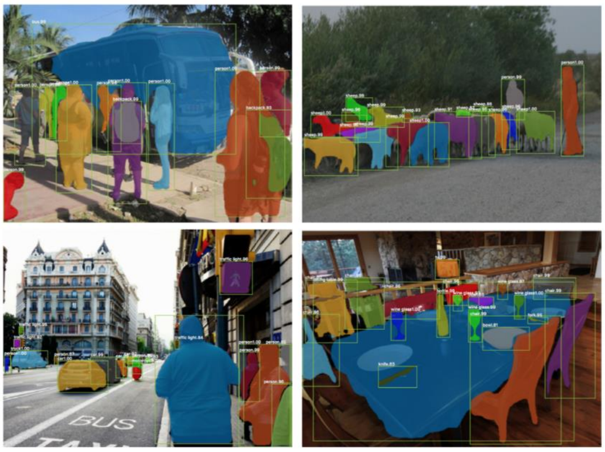


Figure 6. Examples of outputs from the Mask-RCNN algorithm [37]

Starting from an instance segmentation on perspective images (converted from spherical images) allows for detection of doors and windows in mobile LiDAR data.

2.3 Extraction of Door Dimensions out of Point Clouds

Images do not always visualise the whole object of interest (e.g., door or window) because the line of sight is often obstructed by other objects or part of the building itself. This is undoubtedly the case when the point of view of the image is located at a slight angle from the object (see Figure 7). Consequently, automatically extracting the exact dimensions of doors or windows out of the object segmentation is impossible. Therefore, the correct dimensions need to be extracted from the point cloud based on the instance segmentation.

Since the instance segmentation algorithm has yet to give desirable results, labelled training data is used to further develop the processing algorithm to extract dimension and location of doors and windows from a point cloud.

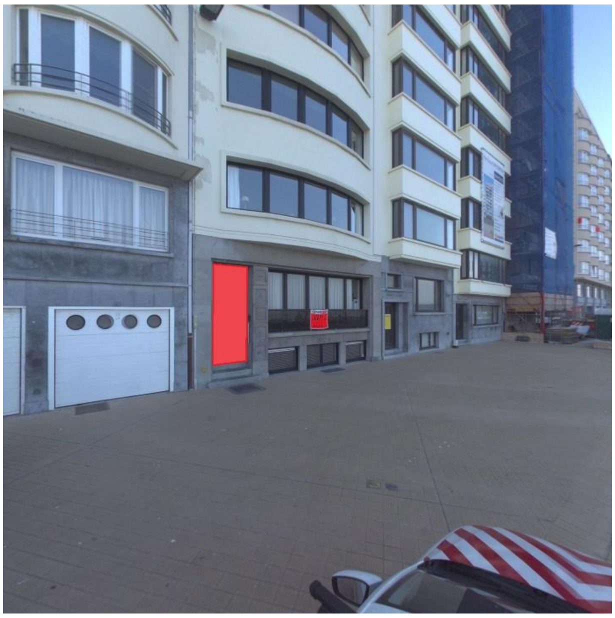


Figure 7. Training set data is used to further develop the point cloud processing algorithm (door labelled by red polygon)

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLII-3-W8-429-2019 | © Authors 2019. CC BY 4.0 License.

Detecting doors, windows and door thresholds and assessing their locations and dimensions can be done by running a semantic segmentation on point clouds [38–40]. Unfortunately, for the case of Flanders, the point cloud does not have extra metadata apart from the information about the location (e.g., no intensity or scan direction flag or edge of flight line and no classification). Therefore, a semantic segmentation on the point cloud is challenging or even impossible to perform accurately. Another method is required to detect the locations and dimensions of doors and windows.

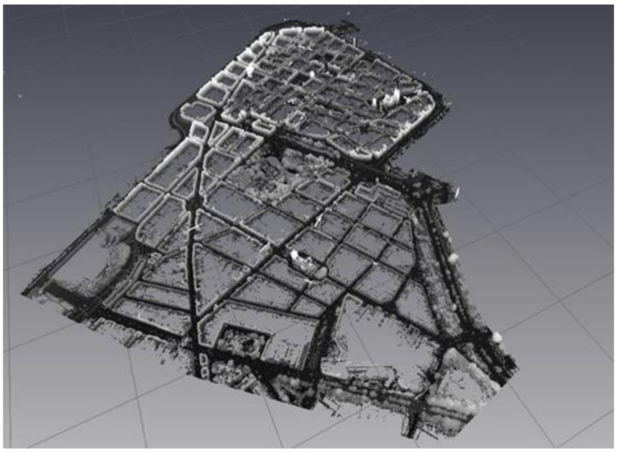


Figure 8. Point cloud of Ostend, captured from a mobile platform

Research conducted in 2005 showed that it is also possible to create a distance-value-added panoramic image [41], where every pixel holds the distance value measured from the location where the images are taken. Similarly, it is possible to create 'dimension-added-value' panorama images, making it possible to extract the location and dimensions after completing the object detection or semantic segmentation.

This method provides the benefit of quickly extracting only relevant point cloud data, whereby point cloud analysis has been reduced to a minimum. Moreover, with the use of multiple 'dimension-added-value' panorama images, it becomes possible to run semantic segmentation of multiple points of views. As a result, it is feasible to detect doors and windows even if an obstacle (e.g., a car or tree) blocks the line of sight from one specific point of view.

After detecting the object (e.g., door, window) with a semantic segmentation algorithm on the spherical images, the metadata of the pixels that are classified as a door or window is extracted and stored in a database.

Although mobile point clouds can give highly accurate measurements of dimensions, this geometry acquisition method, unfortunately, inevitably includes measurement noise at varying degrees. This noise is caused by signal backscattering of the measured targets and the materials of the targets' surface[42].

The Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is used to deal with the noise of the mobile mapping acquisition. DBSCAN groups points that are closely packed together (points with many nearby neighbours). In order to run the DBSCAN clustering, two parameters are required: maximum distance between points ε and the minimum number of points required to form a dense region [43]. First, all so-called core points with a predefined minimum number of points inside the ε neighbourhood of every point are selected. Next, a connected component is created of all core points that are in the neighbour graph. Hereafter, every non-core point is assigned to a formed cluster if the non-core point lies within the ε distance of a cluster. All remaining points are labelled as noise and can be ignored (see Figure 9). Consequently, all points within the DBSCAN cluster are mutually density-connected, and if one point is density-reachable from any point of the cluster, it is part of the cluster as well.

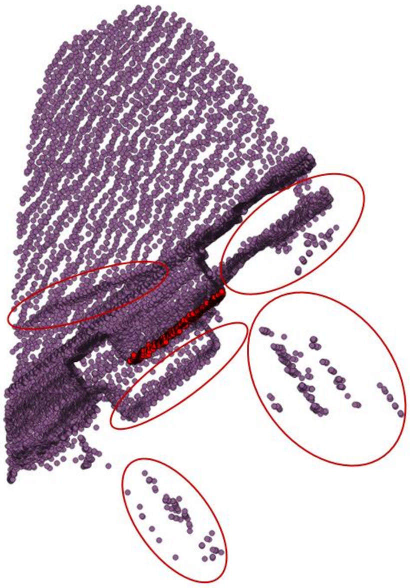


Figure 9. Besides containing the actual measurement of walls, doors and windows, mobile LiDAR data (purple points) contains noise (red circles), which makes it challenging to extract the points that represent doors (red points).

After reducing the noise in the point cloud samples, the detection of a door and window plane is completed. The planes are not created using normal vectors (see Discussion) but rather by calculating the line of best fit in the x,y plane (see Figure 10). Although noise is ignored with the use of DBSCAN clustering, the line of best fit is created by considering the possible existence of outliers. Because of this, it becomes possible to define doors quickly.

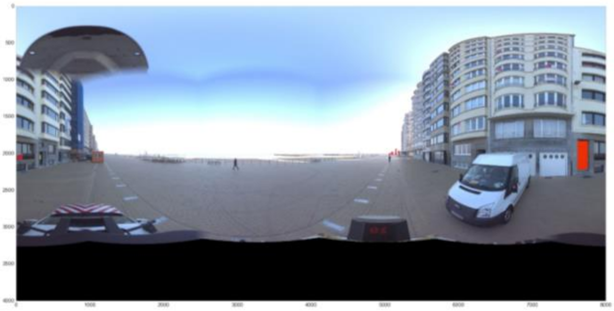


Figure 10. Pixels are automatically selected based on the training data set, whereafter the relevant point cloud dimensions can be used

https://doi.org/10.5194/isprs-archives-XLII-3-W8-429-2019 | © Authors 2019. CC BY 4.0 License.

Unfortunately, due to a wide range of door shapes (e.g., ornamentation and sculpting on front doors), the proposed algorithm does not give satisfactory results after a visual evaluation. Therefore, more extensive research is needed to improve this proposed algorithm pipeline so that the locations and dimensions of doors and windows can later be used in the flood risk assessment methodology in Flanders.

3. DISCUSSION

3.1 Accuracy of the Point Cloud

The point cloud of the mobile mapping has an accuracy between 1 to 2 cm, which means that the extracted location and dimensions of doors, windows and door thresholds will be accurate enough to use in flood risk assessment studies [44]. Nevertheless, this accuracy needs to be included and mentioned together with the output of this calculation so that it can be accounted for in the decision-making process of flood risk management.

3.2 Difference in Region-Dependent Appearance

While the appearance of doors and windows seem only to slightly change from region to region, sometimes these differences can be significant, resulting in decreased accuracy of instance segmentation algorithms. Thus, when an object detection algorithm is trained, special attention must be given to training the model based on images of doors and windows in the specific region of the application (e.g., Flanders).

3.3 Type of Materials

In contrast with algorithms that only use segmentation of LiDAR data to detect doors and windows, this algorithm can provide more information than the dimensions and location of these objects. Because this prototype contains an object detection script, it is possible to incorporate a material detection algorithm, which can detect whether a door or window is made of wood or metal. Although this material detection will remain a rudimentary estimate, this information can be used to estimate the stability of these weak spots for flood events in buildings. Furthermore, the object detection algorithm can also detect the presence of barrier gutters around doors and windows. Moreover, cat doors and mailboxes in front doors can be detected and considered in flood risk assessments.

3.4 Conversion Time Spherical to Cube Box Image

The conversion from the spherical image to cube box images take, on average, seven minutes since the algorithm does not support multithreading on the graphics processing unit (GPU). Instead, everything is purely calculated on central processing units (CPU), which is computation intensive. Fortunately, it was possible to convert the images in parallel using the High-Performance Computing (HPC) infrastructure of Flanders [45]. As a result, it was possible to convert 70,000 images in a few hours, instead of 34 days. Nevertheless, changing this conversion script into a script that supports multithreading on a GPU will be necessary for future use.

3.5 Applicability and Scale of Prototype

Although this prototype is tailored for the Flanders region, it can be used in other regions as well, after some additional script is embedded. The spherical panorama images and mobile LiDAR data can be extracted from the Google StreetView panorama images [46,47]. Cavello M. et al. (2015) suggested a method to reconstruct a point cloud based on multiple different Google StreetView panoramic images along a street [47]. By using the reconstructed point cloud and panorama images from Google StreetView, this prototype can also be used to detect the dimensions and locations of doors, windows and door thresholds.

3.6 Median Clustering of Normal Vectors

In the development of the prototype, the median cluster method of normal vectors was not used. With the clustering of normal vectors of a point cloud, it becomes possible to get segmentations of planes. Unfortunately, due to an overload of noise at windows and windows in doors and the lack of LiDAR data of the glass, it is challenging to extract planes of doors and windows by clustering normal vectors. Moreover, not all front doors have a perfectly flat surface. For example, ornamentation and sculpting on front doors make the detection of the door plane extremely challenging. Nevertheless, a combination of the normal vector clustering method and the line of best fit (in the x,y plane) method, could offer an improved, complementary methodology.

3.7 Upgrading the Prototype

As cited above, the mentioned script requires further research and development to detect the dimensions and locations of doors, windows and door thresholds automatically. At the moment, the script is not ready for valorisation without further improvement in object detection and the final door and window segmentation.

4. CONCLUSIONS

Consideration of the location and dimensions of doors and windows plays a crucial role in increasing the accuracy of flood risk assessment in Flanders. Until now, there has been a lack of data concerning the design and construction of flood-prone building structures. However, the combination of LiDAR data and panoramic images available in Flanders could be used to provide valuable insight into the matter. With the use of instance segmentation on 360° images and processing and analysis of point cloud data, it becomes possible to obtain information on weak spots. This paper reports on the current state of research in the areas of object detection and instance segmentation on images to detect doors and windows in mobile LiDAR data.

REFERENCES

1. Amirebrahimi, S.; Rajabifard, A.; Mendis, P.; Ngo, T. D.; Ngo, T. *A data model for integrating GIS and BIM for assessment and 3D visualisation of flood damage to building*;

2. Pistrika, A. *Flood Damage Estimation based on Flood Simulation Scenarios and a GIS Platform*; 2010; Vol. 30;

3. Amirebrahimi, S.; Rajabifard, A.; Mendis, P.; Ngo, T. A BIM-GIS integration method in support of the assessment and 3D visualisation of flood damage to a building. *J. Spat. Sci.* **2016**, doi:10.1080/14498596.2016.1189365.

4. Amirebrahimi, S.; Rajabifard, A.; Mendis, P.; Ngo, T. A Planning Decision Support Tool for Assessment and 3D Visualisation of Flood Risk to Buildings. In *2016 Floodplain Management Association National Conference*; 2016; pp. 1–15.

5. *Openings in Foundation Walls and Walls of Enclosures Below Elevated Buildings in Special Flood Hazard Areas in accordance with the National Flood Insurance Program*;

6. Van Ackere, S.; Beullens, J.; De Wulf, A.; De Maeyer, P. Data Extraction Algorithm for Energy Performance Certificates (EPC) to Estimate the Maximum Economic Damage of Buildings for Economic Impact Assessment of Floods in Flanders , Belgium. *Int. J. Geo-information* **2018**, *7*, doi:10.3390/ijgi7070272.

7. Diakakis, M. Have flood mortality qualitative characteristics changed during the last decades? The case study of Greece. *Environ. Hazards* **2016**, *15*, 148–159, doi:10.1080/17477891.2016.1147412.

8. Jonkman, S. N.; Maaskant, B.; Boyd, E.; Levitan, M. L. Loss of life caused by the flooding of New Orleans after hurricane Katrina: Analysis of the relationship between flood characteristics and mortality. *Risk Anal.* **2009**, *29*, 676–698, doi:10.1111/j.1539-6924.2008.01190.x.

9. Lew, E. O.; Wetli, C. V. Mortality from Hurricane Andrew. *J. Forensic Sci.* **2015**, *41*, 13933J, doi:10.1520/jfs13933j.

10. Paul, B. A PostgreSQL extension for storing point cloud (LIDAR) data Available online: https://github.com/pgpointcloud/pointcloud.

11. Ravada, S.; Kazar, B. M.; Kothuri, R. Query Processing in 3D Spatial Databases: Experiences with Oracle Spatial 11g. In *3D Geo-Information Sciences*; Springer Berlin Heidelberg: Berlin, Heidelberg, 2009; pp. 153–173.

12. Bell, A.; Chambers, B.; Butler, H.; Gerlek, M. PDAL - Point Data Abstraction Library — pdal.io Available online: https://pdal.io/ (accessed on May 27, 2019).

13. Zhao, W.; Chellappa, R.; Phillips, P. J.; Rosenfeld, A. Face recognition. *ACM Comput. Surv.* **2003**, *35*, 399–458, doi:10.1145/954339.954342.

14. Suarez, J.; Murphy, R. R. Hand gesture recognition with depth images: A review. In *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*; IEEE, 2012; pp. 411–417.

15. Ziaeefard, M.; Bergevin, R. Semantic human activity recognition: A literature review. *Pattern Recognit.* **2015**, *48*, 2329–2345, doi:10.1016/J.PATCOG.2015.03.006.

16. Gerónimo, D.; López, A. M.; Sappa, A. D.; Graf, T. Survey of Pedestrian Detection for Advanced Driver Assistance Systems. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 1239–1258, doi:10.1109/TPAMI.2009.122.

17. Zehang Sun; Bebis, G.; Miller, R. On-road vehicle detection: a review. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 694–711, doi:10.1109/TPAMI.2006.104.

18. Anguelov, D.; Koller, D.; Parker, E.; Thrun, S. *Detecting and Modeling Doors with Mobile Robots*;

19. Parmar, R. Detection and Segmentation through ConvNets Available online: https://towardsdatascience.com/detection-and-segmentation-through-convnets-47aa42de27ea (accessed on Jun 11, 2019).

20. Stoeter, S. A.; Le Mauff, F.; Papanikolopoulos, N. P. Real-time door detection in cluttered environments. In *Proceedings of the 2000 IEEE International Symposium on Intelligent Control. Held jointly with the 8th IEEE Mediterranean Conference on Control and Automation (Cat. No.00CH37147)*; IEEE; pp. 187–192.

21. Munoz-Salinas, R.; Aguirre, E.; Garcia-Silvente, M.; Gonzalez, A. *Door-detection using computer vision and fuzzy logic*;

22. Cicirelli, G.; D'orazio, T.; Distante, A. Target recognition by components for mobile robot navigation. *J. Exp. Theor. Artif. Intell.* **2003**, *15*, 281–297, doi:10.1080/0952813021000039430.

23. Cokal, E.; Erden, A. Development of an image processing system for a special purpose mobile robot navigation. In *Proceedings Fourth Annual Conference on Mechatronics and Machine Vision in Practice*; IEEE Comput. Soc; pp. 246–252.

24. Delo, A. ¯; Martinovi´c, M.; Van Gool, L. Bayesian Grammar Learning for Inverse Procedural Modeling. **2013**, doi:10.1109/CVPR.2013.33.

25. Teboul, O.; Kokkinos, I.; Simon, ıc; Koutsourakis, P.; Paragios, N. *Shape Grammar Parsing via Reinforcement Learning*;
26. Riemenschneider, H.; Krispel, U.; Thaller, W.; Donoser, M.; Havemann, S.; Fellner, D.; Bischof, H. Irregular lattices for complex shape grammar facade parsing. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*; IEEE, 2012; pp. 1640–1647.

27. Martinovi¢, A.; Mathias, M.; Weissenberg, J.; Gool, L. Van *A Three-Layered Approach to Facade Parsing*;

28. Tyleček, R.; Radimšára, R. R. *Spatial Pattern Templates for Recognition of Objects with Regular Structure*;

29. Yang, M. Y.; Forstner, W. A hierarchical conditional random field model for labeling and classifying images of man-made scenes. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*; IEEE, 2011; pp. 196–203.

30. Cohen, A.; Schwing, A. G.; Pollefeys, M. Efficient Structured Parsing of Facades Using Dynamic Programming. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*; IEEE, 2014; pp. 3206–3213.

31. Gadde, R.; Marlet, R.; Nikos, P. Learning Grammars for Architecture-Specific Facade Parsing. *Int. J. Comput. Vis.* **2014**, *117*, 290–316.

32. Kozinski, M.; Gadde, R.; Zagoruyko, S.; Obozinski, G.; Marlet, R. A MRF shape prior for facade parsing with occlusions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE, 2015; pp. 2820–2828.

33. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. **2015**.

34. Dai, J.; Li, Y.; He, K.; Sun, J. R-FCN: Object Detection via Region-based Fully Convolutional Networks. **2016**.

35. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A. C. SSD: Single Shot MultiBox Detector. **2015**, doi:10.1007/978-3-319-46448-0_2.

36. Huang, J.; Rathod, V.; Sun, C.; Zhu, M.; Korattikara, A.; Fathi, A.; Fischer, I.; Wojna, Z.; Song, Y.; Guadarrama, S.; Murphy, K. Speed/accuracy trade-offs for modern convolutional object detectors. **2016**.

37. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. **2017**.

38. Qi, C. R.; Su, H.; Mo, K.; Guibas, L. J. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation 2017, 652–660.

39. Koppula, H. S.; Anand, A.; Joachims, T.; Saxena, A. Semantic Labeling of 3D Point Clouds for Indoor Scenes 2011, 244–252.

40. Hackel, T.; Wegner, J. D.; Schindler, K. Fast semantic segmentation of 3D point clouds with strongly varying density., doi:10.5194/isprsannals-III-3-177-2016.

41. Verbree, E.; Zlatanova, S.; Dijkman, S. *Distance-value-added panoramic images as the base data model for 3D-GIS*;

42. Guan, H.; Yu, Y.; Li, J.; Liu, P.; Zhao, H.; Wang, C. Automated extraction of manhole covers using mobile LiDAR data. *Remote Sens. Lett.* **2014**, *5*, 1042–1050, doi:10.1080/2150704X.2014.994716.

43. Schubert, E.; Sander, J.; Ester, M.; Kriegel, H. P.; Xu, X. DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN. *ACM Trans. Database Syst.* **2017**, *42*, 1–21, doi:10.1145/3068335.

44. Teccon Mobile mapping in cijfers Available online: https://teccon.be/diensten/mobile-mapping/visualiseren/ (accessed on Jun 28, 2019).

45. Vlaams Supercomputer Centrum (VSC) Supercomputing | VSC | Vlaams Supercomputer Centrum | Flanders Available online: https://www.vscentrum.be/ (accessed on Jun 13, 2019).

46. nøcomputer Creating point clouds with Google Street View – nøcomputer – Medium Available online: https://medium.com/@nocomputer/creating-point-clouds-with-google-street-view-185faad9d4ee (accessed on May 27, 2019).

47. Cavallo, M. *3D City Reconstruction From Google Street View*;