



HAL
open science

Energy function algorithm for detection of openings in indoor point clouds

Rami Assi, Tania Landes, H el ene Macher, Pierre Grussenmeyer

► To cite this version:

Rami Assi, Tania Landes, H el ene Macher, Pierre Grussenmeyer. Energy function algorithm for detection of openings in indoor point clouds. International Society for Photogrammetry and Remote Sensing, 2019, <10.5194/isprs-archives-XLII-2-W13-747-2019>. <hal-04234126>

HAL Id: hal-04234126

<https://hal.science/hal-04234126v1>

Submitted on 9 Oct 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destin e au d ep ot et  a la diffusion de documents scientifiques de niveau recherche, publi es ou non,  emanant des  tablissements d'enseignement et de recherche fran ais ou  trangers, des laboratoires publics ou priv es.



HAL Authorization

ENERGY FUNCTION ALGORITHM FOR DETECTION OF OPENINGS IN INDOOR POINT CLOUDS

Rami Assi, Tania Landes, H el ene Macher, Pierre Grussenmeyer

ICube Laboratory UMR 7357, Photogrammetry and Geomatics Group, National Institute of Applied Sciences (INSA),
24 Boulevard de la Victoire, 67084 Strasbourg Cedex, France

rami.assi@insa-strasbourg.fr; tania.landes@insa-strasbourg.fr; helene.macher@insa-strasbourg.fr;
pierre.grussenmeyer@insa-strasbourg.fr

KEY WORDS: Indoor point clouds, Building information model, Deep Learning, Segmentation, Registration

ABSTRACT:

As the use of building information model (BIM) for architectural heritage becomes more relevant, this paper explores different solutions to further automatize the modelling process. The scan-to-BIM process still requires manual intervention that is time consuming, subject to errors and user-dependent. In this paper, the main focus is the automated segmentation of windows. In the first part of our paper, we will review and compare several state-of-the-art methods for automatic detection and segmentation of openings in a point cloud. Based on the most pertinent aspects of those methods, a new algorithm focusing on indoor point clouds is proposed. After walls are already detected, they are converted in 2D binary images. Holes in those images correspond to openings. We submit each opening to an energy function with two terms: data and coherence. The data term depends on the shape of the opening. The coherence term considers the position of the opening in the scene. Those function let us determine if an opening in the point cloud is due to a window/door or an object obstructing the acquisition. In the third part we discuss the results obtained by applying the method to different datasets.

1. INTRODUCTION

As the use of building information model (BIM) for architectural heritage becomes more relevant, this paper explores different solutions to further automatize the modelling process. The scan-to-BIM process still requires manual intervention that is time consuming, subject to errors and user-dependent. The work toward automatization of the process is in the continuity of the lab's previous research. Semi-automatic methods were already developed for indoor (Macher et al., 2017), outdoor segmentation (Boulaassal et al., 2010) and roof segmentation (Tarsha-Kurdi et al., 2008). The implicit next step is the judicious combination and improvement of those methods to deliver the complete segmentation. However, in this paper, the main focus is the segmentation of windows. An opening in the fa ade is the unique common entity that can be seen from inside and outside. As such, it can help the registration of indoor and outdoor point clouds. Besides, as we strive to enhance segmentation, being able to automatically model and label windows is pertinent on its own.

In the first section of our paper, we will review and compare several state-of-the-art methods for automatic detection and segmentation of openings in a point cloud. Those methods differ on their degree of automatization and their field of application. Some of them were especially designed for indoor segmentation (Wenzhong et al., 2019) while others were only applied to exterior fa ades (Zolanvari et al., 2018). Recent studies highlight the success of deep learning convolutional neural network (CNN) for point cloud segmentation. While deep learning has already proven to be very efficient for many image processing problems, its applications to 3D point clouds are an active research field.

Based on the most pertinent aspects of those methods, a new algorithm focusing on indoor point clouds is proposed. After

walls are already detected, they are converted in 2D binary images. Holes in those images correspond to openings. Some of those openings are neither windows nor doors but the shape of objects that obstructed the wall during the acquisition. We associate each cluster to an energy function with two terms: data and coherence as suggested by Boykov et al. (2001) and Wenzhong et al. (2019). The data term depends on the shape of the opening. We can define it so that a rectangular opening has higher data energy. This will discard any opening due to non-rectangular objects obstructing the wall. The coherence term considers the position of the opening in the scene. It includes several criteria, such as for instance: if the centroid of the opening is too close to the floor or the ceiling, it cannot be a door or a window.

The third and final section of the paper compares the results obtained by applying the previously mentioned methods to our dataset. Algorithms were tested on data acquired in and around the zoological museum of Strasbourg (France) and an individual house (only indoor). Indoor data were acquired with static laser scanning technique and outdoor data with a mobile mapping system (Stereopolis from IGN). The museum consists of four floors and a vast attic, with an internal courtyard. The ground floor is composed of class rooms and labs. Two floors are dedicated to the visiting tour and are filled with life-sized animal models as well as complete skeletons. The third floor houses the employees' offices. Pieces of furniture obstruct walls of most rooms. In the attic, old exposition models are stored. The individual house consists of two floors with a garden. All rooms are filled with household furniture. All those obstacles that cannot be removed during data acquisition create holes and therefore openings when segmenting the walls of each room. This apparent drawback makes those datasets particularly pertinent to test the robustness of our algorithm. Results are evaluated and discussed to validate our geometrical approach,

as well as to evaluate the pertinence of the deep learning approach for future research.

2. STATE-OF-THE-ART

2.1 Previous works

The lab's previous works include two segmentation pipelines, one specifically designed for façade segmentation (Boulaassal et al., 2010) and the other for indoor (Macher et al., 2017) point cloud segmentation. The former uses several iterations of the RANSAC algorithm (Fischler and Bolles, 1981) to find the main planes fitting the points of the façade. Independently for each plane, the Delaunay triangulation is calculated in order to use one of its underlying property. Areas with high density of points will form triangles with short edges whereas areas with fewer points will form triangles with longer edges. With the correct threshold, it is possible to extract only points of the latter areas. In the point cloud of a façade, the less populated areas are the boundaries of the façade and its openings (windows and doors). This pipeline allows quick and automated modelling of façades. It also models the openings. However, the resulting wireframe models are not yet identified and segmented as windows. Besides, information about the façade's width is missing, because all extracted planes are modelled as surfaces rather than volumes.

The second method (Macher et al., 2017), focused on indoor segmentation, automatically transforms an indoor point cloud into an *obj* file modelling walls, floors and ceiling as volumes. First, the peaks of the histogram showing the points' altitude distribution are alternatively identified as the floors and ceilings. For each floor, a horizontal slice close to the ceiling is considered. Most furniture and doors do not reach the ceiling, so this slice will most likely only contain points belonging to walls. The slice of points is projected on a horizontal plane and converted into an image. From this perspective, a simple region growing segmentation can separate each room. For each room point cloud, the application of MLSAC algorithm (Torr and Zisserman, 2000) in an iterative way allows to find vertical planes and consequently identify walls. Each plane is associated with parallel planes that are close enough so that wall point clouds are identified. Finally, walls and slabs are reconstructed in an *obj* file before being converted into an *ifc* file. The algorithm processes large point clouds (up to 45 million points) within 30 minutes. It is relatively short considering the quality of the segmentation.

2.2 Related works

While it does not directly address the problematic of joint indoor/outdoor segmentation, many other researches were led in the field of opening detection.

Most LiDAR acquisition systems (both static and dynamic) store for each detected point, the properties of the laser that was reflected on it i.e. mainly the station-to-point distance and direction of the laser. Assuming that the position of a wall is known, those properties can be exploited for opening detection (Colleu and Benitez, 2013);(Tuttas and Stilla, 2013). For each point, the corresponding laser direction is calculated, as well as the intersection between the laser direction and the known plane of the wall. At this stage, there are three possibilities:

- a) If the point under study is closer to the station than to the intersection point, the point belongs to an object obstructing the wall (furniture for indoor clouds, vehicle or tree outdoor clouds);
- b) If the point coincides with the intersection area, the point belongs to the wall;
- c) If the point is farther than the intersection, the laser went through a window. This simple process allows the segmentation of indoor or outdoor in 3 labels i.e. wall, obstruction and window.

The next method (Li and al., 2018) takes advantage of an architectural property found in building façades: windows are usually aligned along horizontal and vertical lines. This means that when a window is detected, there are probably similar windows next to it on the same floor, but also above and under it on each floor. In this regard, façades are processed with horizontal and vertical slices rather than point by point. Gradient of different variables (density of points, colours, intensity) can be calculated across the slice. A brutal change in gradient value indicates that the slice is probably the frontier between a wall and a window. Like the previous method (Zolanvari et al., 2018) processes façade horizontally and vertically. The façade is decomposed in horizontal slices with a chosen width. Each slice is a point cloud of the wall split by segments of windows. Those windows appear as holes in the point cloud's slice, separating it in two distinct clusters. Counting and locating the clusters is enough to infer the number and positions of the windows' vertical edges. The same method is applied on vertical slices to find windows' horizontal edges. The speed and accuracy of the method mostly depends on the width chosen for the slices.

2.3 Deep learning approaches

In topography and geomatics as in many other fields, deep learning has recently been the focus of several researches. Deep is a branch of machine learning often used for 2D image recognition problems. It can be decomposed in two successive steps: training and testing. Training means that data is fed to several layers of randomly initialized functions called the convolutional neural network (CNN). The randomly segmented output is compared to the manually segmented ground truth to obtain the error. Then the functions' parameters will be slightly changed to minimize the error. This step is repeated until the error is small enough to consider that the parameters of the functions have converged. The test step means that new data is fed to the CNN after convergence. If the training step was successful, the output should correspond to the ground truth with a small marge of error.

In a 3D point cloud, there is no convenient way to order points, unlike in 2D matrices (left to right and up to down). As an object in 3D can be considered from all perspectives, its labelling must be invariant to rotations and translations. Point clouds are also much heavier data than 2D images and require more time and/or power to process. The design of a 3D CNN must meet those criteria. Different approaches already exist.

SnapNet (Boulch et al., 2017) transposes the problem to 2D segmentation. Point clouds are roughly textured, then 2D screenshots are taken from different perspectives. Those images can be segmented by 2D CNN. When taking the screenshot, the depth of the object relative to the point of view brings one more parameter to feed to the CNN.

Rather than processing every point, SuperpointGraph (Landrieu and Simonovsky, 2018) regroups points with the same geometric properties into clusters they call “superpoint”. For instance, a superpoint can be a group of points belonging to the same plane. Then the graph representing the superpoint is fed to a 3D CNN that recognizes object as a specific combination of superpoints (geometries).

In our study, we explored the method that directly takes a 3D point cloud as an input: PointNet (Qi et al., 2017). It was already tested on indoor point clouds. As such we wanted to test it to validate the pertinence of the deep learning approach for opening segmentation.

We implemented the method described on PointNet’s github. We first tested the training algorithm using the dependencies and OS specified on the git. Since most of those are not supported any more (Ubuntu 14.04), we also tried to use the code on Ubuntu 18.04 with the newest version of Cuda and Tensorflow. Training achieved 78.6% accuracy with their datasets.

3. METHOD

Researching the state-of-the-art let us develop a novel method inspired by previous promising work.

3.1 Overview

The proposed opening detection method is just one step in a pipeline that aims at the joint automated segmentation of indoor and outdoor point cloud (figure 1).

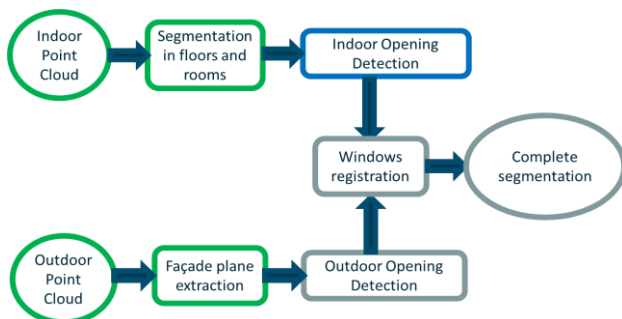


Figure 1. Current pipeline of the project. Green: previous works. Blue: ongoing development. Grey: future works.

Macher et al. (2017) and Boulaassal et al. (2010) respectively developed methods for indoor segmentation in floors and rooms and extraction of planes in façade. We currently focus on opening detection to use windows as common objects to register indoor and outdoor point clouds. In this article we focus on indoor opening detection. As you can infer from figure 1, our method takes as input an indoor point cloud where floors, rooms and walls were already segmented. The output is an .obj file representing floors, ceilings and walls as volume (to the exception of walls of the outer façade), with windows segmented out of walls.

3.2 Developed approach for indoor openings detection

The code is developed in Matlab (MathWorks). The indoor opening detection method on its own can be described in different steps, as detailed in figure 2.

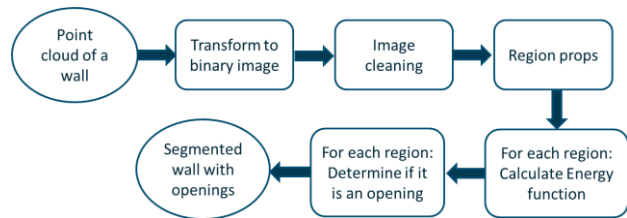


Figure 2. Pipeline of the point cloud processing chain for indoor opening segmentation

Our previous work on indoor segmentation let us retrieve the indexes of the points belonging to each floor, each room and each wall. That is the sequence in which we process the input point cloud.

For each wall, we know its centroid, Cartesian equation and the indexes of the points that belong to it.

3.2.1 Transform to binary image

As points of a wall approximately belong to the same plane, we transpose the problem to 2D like Boulch et al. (2017). First, we project all points on the plane defined by the Cartesian equation. We look for farthest points on the right and left side of the wall. We also look for the highest and lowest. Those 4 points will define the width and height of the wall in the binary image.

We tried different values for the sampling of the image i.e. pixel size. If the value is too low, the image will represent every hole between the points resulting in a lot of noises. If the sampling is set too high the details of openings will be lost and it will be hard to distinguish windows and obstructions. We chose a pixel size of 6 mm x 6 mm.

For later steps, pixels of walls are represented with zeros and pixels of openings with ones.

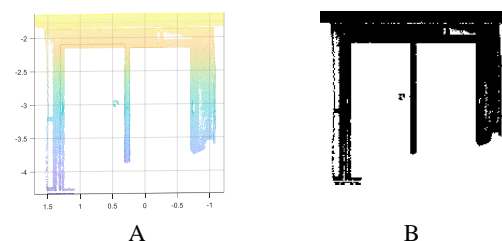


Figure 3. A) a 3D point cloud of a wall with two windows. B) the right 2D binary projection of the wall. Windows and noises appear in white on the black wall.

3.2.2 Image Cleaning

After looking at several datasets, it appeared that most openings due to obstructions are close to the ground. We discarded those openings by cleaning each column of the image that includes h/2 consecutive black pixels, where h is the height of the image of the wall.

A morphological opening (erosion followed by a dilatation) is applied on the binary image. It cleans most the remaining isolated noise (figure 4).



Figure 4. Binary image of the wall after cleaning.

3.2.3 Region Props

This part uses Matlab's *regionprops* function provided by the Image Processing Toolbox. It takes a binary image as input and finds the different region composing it. It uses a region growing algorithms. For each region, *regionprops* returns additional information that will be used in the next step: a binary image of the region, its extremums, its centroid and the number of pixels in the region.

3.2.4 Energy Function

To each region we associate two values that depends on the properties of the region. The term energy function and its primary decomposition was inspired by Boykov et al. (2001) and Whenzong et al. (2019).

The coherence energy (1) verifies that the region do not have aberrant properties preventing this region from being a window.

$$E_{coherence} = \begin{cases} 1 & \text{if } h0 < \tau_1 \\ 1 & \text{if } N < \tau_2 \\ 1 & \text{if } H/W < \tau_3 \\ 1 & \text{if } H/W > \tau_4 \\ 0 & \text{else} \end{cases} \quad (1)$$

Where $E_{coherence}$ = coherence term

- $h0$ = height of the centroid
- τ_1 = minimum height for a window's centroid
- N = number of points in the region
- τ_2 = minimum number of points to discard noise
- H = height of the region
- W = width of the region
- τ_3 = minimum H/W acceptable for a window
- τ_4 = maximum H/W acceptable for a window

In (1) the first criteria allows us to discard objects that are too close to the ceiling to be anything else than an object obstructing the wall. The second criteria discards any opening that is too small to be a window, such as remaining noise or other obstruction. The third and fourth criteria are threshold that we defined to discard objects that are too large and small or too thin and high. The region can be an opening only if its coherence energy is null.

The data energy (2) depends on the repartition of non-null points in the image of the region.

$$E_{data} = \frac{\sum_i \sum_j R(i,j)}{NbC * NbL} \quad (2)$$

Where E_{data} = data energy

- $R(i,j)$ = region's image(i,j) pixel (with value 0 or 1)
- NbC = width of the region's image
- NbL = height of the region's image

For each line, we look at the repartition of pixels per columns. If all columns contain a non-null pixel, the opening is perfectly rectangular, and the data energy will be equal to the height of the region's image. It is reduced for each line that is not filled. With the correct value for threshold τ_5 , this data term allows us to discard objects that passed the coherence term but do not have the usual mostly rectangular shape of windows or doors.

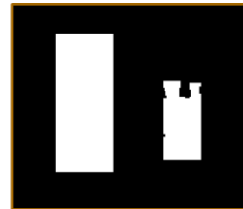


Figure 5. The region on the left was obtained using the *regionprops* on figure 4. It is perfectly rectangular, so its data energy is equal to one. The region on the right is not completely rectangular, its energy will be lower.

3.2.5 Segmentation

The energy functions let us conclude for each region if it is a window/door or if it is not a real opening. For each window and door, we calculate the coordinates of its centroid and extrema relatively to the centroid of the wall. Then, we edit an *.obj* file of the walls to extrude doors and windows. For this step, we used the Boolean modifier of the software Blender. We select the *.obj* of the walls and the *.obj* file containing the windows. Blender computes the difference i.e. walls with extruded windows.

4. RESULTS

Since the zoological museum is a large point cloud (60 million points), the algorithm was first tested and tuned using the point cloud of a smaller habitation (10 million points) to observe results faster. In this dataset, most windows are partially occulted by furnitures. But it was possible to see through all windows (no curtains or shutter). This dataset has the advantage of including different sizes of rectangular windows. The data acquisition was performed by the office David Pierrot (Mandelieu, France).

With 10 million points, the point cloud to BIM with segmented windows process took 9 minutes. After several experiments we chose the following thresholds

Parameter	Value
τ_1	30 pixels
τ_2	300
τ_3	3
τ_4	3
τ_5	0.8

Table 1. Threshold values for the energy function

For the individual house, 69% of windows were detected and segmented. 100 % of unobstructed windows were detected and segmented. The algorithm failed to detect windows that were partially obstructed by furnitures. In the binary image, those windows are merged with the obstructions as a single region. The resulting openings were discarded as they had a very low data energy.

For each segmented window we manually checked the position of its centroid to evaluate the error of positioning in the final model. The average error is below 5 cm. See figure 6 for segmentation results.

On the museum, only the offices' windows were detected. Indeed, most windows were covered, closed with shutters in order to protect the models exposed in the museum. Those windows did not appear as holes in the point clouds of the walls.

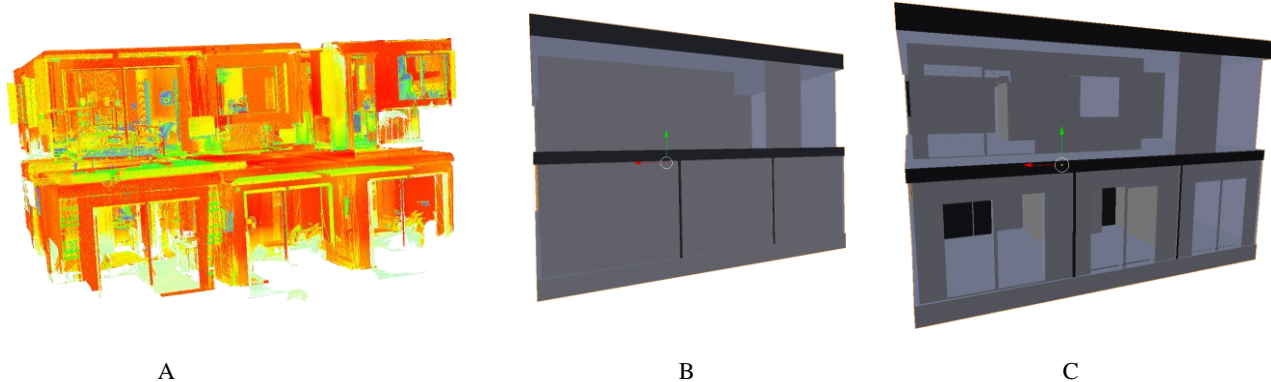


Figure 6. A) point cloud of the individual habitation. B) *.obj* model after segmentation of floors, ceilings and walls (Macher et al., 2017); (Acquisition: David Pierrot, Mandelieu). C) *.obj* model after implementation of the energy function.

5. CONCLUSION AND FUTURE WORKS

We developed an algorithm to automatically detect and segment openings in indoor point cloud. While this purpose is pertinent on its own, it is part of a bigger project that aims at automatic registration of indoor and outdoor point cloud. In this regard, the results are very promising. Indeed, in order to register indoor and outdoor point clouds, only a few common points are necessary. For this purpose, detecting 100% of openings is not required. Yet the algorithm is capable of reaching 100% accuracy on point clouds acquired with unobstructed windows. This drawback had the algorithm perform poorly on our main dataset, the museum. However, it proved rather efficient on point clouds corresponding to this requirement. With the individual house, 100% of unobstructed windows were segmented. Including all windows (partially obstructed and completely obstructed) 69 % were segmented. The restriction of the method is not an issue for many point clouds. However, it can still be improved to detect and segment windows that are partially obstructed. For instance, we could implement the method that analyses the laser's trajectory (Colleu and Benitez, 2013). It would let us separate holes in the point cloud that were caused by windows and objects.

In order to reduce the error of positioning of the windows, we could consider the pattern of the façade. If we detect windows of the same type on the same floor, their centroids should be positioned periodically on the same height. Each detected window would enhance the positioning of similar windows.

As for the museum, we intend to look more into deep learning approaches for window detection. Deep learning algorithm seem to be more appropriate to cover the large variety of windows that must be detected.

REFERENCES

- Boulaassal, H., Landes, T. and Chevrier, C. (2010) From Laser Data to Parametric Models: Towards an Automatic Method for Building Façade Modelling. EuroMed: Digital Heritage, pp. 42-55.
- Boulch, A., Guerry, J., Le Saux, B. and Audebert, N. (2017). SnapNet: 3D point cloud semantic labeling with 2D deep segmentation networks. Computers and Graphics (Pergamon), 71, pp. 189–198. <https://doi.org/10.1016/j.cag.2017.11.010>
- Boykov Y., Veksler O., and Zabih R. (2001) Fast approximate energy minimization via graph cuts, IEEE Transactions on pattern analysis and machine intelligence, volume 23, issue 11, p 1222-1239.
- Colleu, E. and Benitez, S. (2013) Ray-based detection of openings in urban areas using mobiles Lidar data. ISPRS Archives, vol 41, July pp. 591-598
- Fischler, M.a and Bolles, R. C. (1981) Random Sample Consensus: A paradigm for model fitting with. In communications of the ACM, volume 24, pp. 381-395
- Landrieu L. and Simonovsky M. (2018) Large-scale Point Cloud Semantic Segmentation with Superpoint Graphs, 2018 IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR 2018).
- Li, J., Xiong, B., Biljecki, and F., Schrotter, G. (2018). A sliding window method for detecting corners of openings from terrestrial LiDAR data. XLII-4/W10. 97-103. 10.5194/isprs-archives-XLII-4-W10-97.
- Macher, H., Landes, T. and Grussenmeyer, P. (2017) From Point Clouds to Building Information Models: 3D Semi-Automatic Reconstruction of Indoors of Existing Buildings. Applied science 7(10), 1030, 30 pages. doi:10.3390/app7101030

Qi C.R., Su H., Mo K., and Guibas, L.J. (2017). PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 77-85.

Tarsha-Kurdi, F., Landes, T., and Grussenmeyer, P. (2008). Extended RANSAC algorithm for automatic detection of building roof planes from Lidar data. *The Photogrammetric Journal of Finland*. Vol. 21, n°1, 2008, pp. 97-109.

Gago-Silva, A., 2016. GRASS GIS in Grid Environment. doi.org/10.6084/m9.figshare.3188950.

Torr, P.H. and Zisserman, A. (2000) Mlesac : a new robust estimator with application to estimating image geometry. *Computer vision and image understanding*, 78(1):138-156

Tuttas, S. and Stilla U. (2013) Window detection in sparse point clouds using indoor points, *ISPRS archives, Remote sensing and spatial and spatial information sciences*. March pp.131-136.

Wenzhong, S., Wael, A., Na, L., Wenzheng, F., Haodong, X., and Muyang, W. (2019) Semantic Geometric Modelling of Unstructured Indoor Point Cloud. *ISPRS Int. J. Geo-Inf.* 8, 9. 20 pages; doi:10.3390/ijgi8010009

Zolanvari, S. M. I., Laefer, D. F. and Natanzi, A. S. (2018) Three-dimensional building façade segmentation and opening area detection from point clouds, *ISPRS J. Photogramm. Remote Sens.*, vol. 143, pp. 134–149.