Journal: ISPRS Journal of Photogrammetry and Remote Sensing

Point clouds for direct pedestrian pathfinding in urban environments

Jesús Balado[a]*, Lucía Díaz-Vilariño[ab], Pedro Arias[a], Henrique Lorenzo[a]

[a]Applied Geotechnologies Group, Dept. Natural Resources and Environmental Engineering, School of Mining and Energy Engineering, University of Vigo, Campus Lagoas-Marcosende, CP 36310 Vigo, Spain

[b]Dept. Civil Engineering, University of Porto, s/n, R. Dr. Roberto Frias, 4200-465 Porto

---

Highlights:

- Direct use of point clouds to generate navigable graphs for pedestrians in urban environments
- Real routes generated according to two motor skills: pedestrians without reduced mobility and wheelchairs
- Navigable space based on ground elements and static objects
- Occlusion correction of sidewalks by applying morphological operations

---

Abstract

Pathfinding applications for the citizen in urban environments are usually designed from the perspective of a driver, not being effective for pedestrians. In addition, urban scenes have multiple elements that interfere with pedestrian routes and navigable space. In this paper, a methodology for the direct use of point clouds for pathfinding in urban environments is presented, solving the main limitations for this purpose: a) the excessive number of points is reduced for transformation into nodes on the final graph, b) urban static elements acting as permanent obstacles, such as furniture and trees, are delimited and differentiated from dynamic elements such as pedestrians, c) occlusions on ground elements are corrected to enable a complete graph modelling, and d) navigable space is delimited from free unobstructed space according to two motor skills (pedestrians without reduced mobility and wheelchairs). The methodology is tested into three different streets sampled as point clouds by mobile laser scanning (MLS) systems: an intersection of several streets with ground composed of sidewalks at different heights; an avenue with wide sidewalks, trees and cars parked on one side; and a street with a single-lane road and narrow sidewalks. By applying Dijkstra pathfinding algorithm to the resulting graphs, the correct viability of the generated routes has been verified based on a visual analysis of the generated routes on the point cloud and on the knowledge of the urban study area. The methodology enables the automatic generation of graphs representing the navigable urban space, on which safe and real routes for different motor skills can be calculated.

Keywords: spatial analysis, physical accessibility, pedestrian path planning, navigable space, graph modelling, Mobile Laser Scanning

---

1. Introduction

Pedestrian pathfinding is a current challenge that still subsides in many cities. More and more cities are being adapted to new street designs that promote the displacement of the citizen by foot, bike and public transport. Regardless of distance, either go to the nearest bus stop or walk several blocks, pedestrian displacements in urban environment entail difficulties.

Applications as Google Maps, Baidu Maps, Bing Maps, etc. have inbuilt pedestrian navigation modules. However, these options do not provide a real solution to the problem because navigable ground elements, such as sidewalks and pedestrian crossings, are mostly not being considered in the network. Pedestrian routes calculated from road networks (Gerke et al., 2004) present two serious problems. Firstly, they are focused on traveling along the road, assuming that there are sidewalks close to it, which is not always true; and they do not consider crosswalks to cross the road. The provided route is not safe, proposing to walk on the road and to cross by prohibited areas. Secondly, the generated routes are not adapted to the different motor profiles of the citizens. A person without reduced mobility can walk through any ground element, however, for a person in a wheelchair, one small step or curb turns into an impassable barrier.

LiDAR technology allows the acquisition of small elements in a quick and accurate way. Specifically, mobile laser scanning (MLS) is capable of acquiring entire streets with high point density (Kumar et al., 2017) and, thanks to the recent advances in the field of point cloud classification, it is possible to label most of the elements forming the ground (Balado et al., 2018). However, there are still some limitations to perform a correct pathfinding directly on point clouds:

- Large number of points existing in the cloud, useful for classification, becomes a problem of over-information and processing cost when all points are considered as graph nodes for the application of path finding algorithms.
- The density of MLS point clouds is not uniform. Point density is higher in the road areas closer to the sensor than in distant areas. Even in horizontal elements relatively close to the MLS trajectory, such as sidewalks, different levels of density can be obtained.
- Occlusions, absence of information, are common in urban MLS point clouds due to the large number of existing objects in the urban environment. These occlusions produce missing nodes in the final graph and, therefore, the generated model for pathfinding does not conform to reality.

In this paper, specific methods to address and solve the previous limitations are developed. The aim of this paper is to demonstrate the potentiality of the direct use of point clouds to solve pedestrian pathfinding problems. The proposed methodology begins with the use of point clouds where ground elements are previously classified by methodologies already presented in other works (Balado et al., 2018; Riveiro et al., 2015). Occlusions existing on sidewalks are corrected by applying morphological operations (in detail in Section 4.2), so that the final graph is more similar to reality. The point cloud is simplified and internal relationships are established for each ground element, between adjacent elements and obstacles. Once the final graph model is created, Dijkstra algorithm is applied just to verify that the resulting route is safe and viable according to different person's motor skill.

The methodology takes advantage of the LIDAR surveys that many cities are currently carrying out for various purposes (inventory of mobility, control of parking areas, state of the road, etc.) and gives it untested use so far, aimed at pedestrians and people with reduced mobility. The results shown in this work do not need manual processing.

This paper is structured as follows. In Section 2 works related with graph generation to pathfinding are reviewed. The methodologies used for the classification of the input point cloud are summarized in Section 3. A detailed description of the proposed methodology is provided in Section 4. Results obtained from the application of the methodology to several case studies are presented in Section 5. Section 6 is devoted to conclude the work.

2. Related work

Graphs for pedestrian pathfinding are non-directed graphs with positive values in each arc. In urban modelling, there are mainly two types of graphs for navigation: navigation graphs and cellular automata (Singhal and Kundra, 2014). Navigation graphs assign nodes/arches to constructed elements related to the end use, the intersections between them and the represented space. They are similar to graphs used for road network representations (Beneš et al., 2014; Gang and Guangshun, 2010). Navigation graphs are very useful to represent large-scale models with low precision. Cellular automata consists on a regular discretization of the space (Eckel, 2015), normally in grid or voxel-grid, maintaining constant dimensions and number of adjacency connections (Pettré et al., 2005). They are used in small study areas that require a surface based modelling with high level of detail of the environment (Butenuth et al., 2011; Izzati et al., 2015). Some studies combine both representations (Applegate et al., 2010). In this work, a discretization of the navigable space, similar to cellular automata, is chosen to distribute the nodes and generate 3D navigation graphs. Due to the high density of the point clouds, 3D ground elements must be discretized in order to reduce the size of the existing data in the final models without renouncing to a high level of detail. Although cellular automata allows 3D representations, an approximation of the built environment is necessary when the model is generated, thus losing part of the precision provided by point clouds, and the final model (such as 3D image) contains more information that is not necessary and more difficult to interpret than a navigation graph. By contrast, navigation graphs based on the downsampled points have the same accuracy of the acquired point cloud with a smaller number of nodes than its equivalent in cellular automata.

Graph modelling for pedestrian pathfinding has been studied in recent years mainly in indoor environments. Walking is the main way to move inside buildings and over time, buildings became larger and more complex.. These studies start from consistent geometric 2D and 3D models, extracted from BIM (Building Information Models) or point clouds. In point clouds, elements that take part of the route (floor) or that limit it (obstacles) must be located with segmentation and classification methods based on point or object features. Fichtner et al., (2018) structure the point cloud in an octree to proceed with the subdivision of the space and the classification into walkable elements (floor and stairs) and not walkable (walls and obstacles). The octree structure allows to model the navigable space inside the rooms. In a similar way, Staats et al., (2017) use voxels to structure and classify the point cloud. In their methodology, dynamic elements are subtracted from the point cloud based on the fact that they are not a constant part of the scene. The authors consider dynamic elements to be pedestrians and small vehicles in movement. Likewise, in later phases furniture is also subtracted to obtain the navigable space.

The most basic model for a pedestrian graph relates rooms of buildings (Boguslawski et al., 2016). This topological model is created from adjacent rooms sharing doors. Nodes representing rooms are located in the centre of each room and distances associated with arcs are distances between room centres. The graph can be improved considering doors as intermediate nodes between room nodes (Lorenz et al., 2006). As result, the created graph represents more realistic distances between rooms than the previous one: only given distances between room centres, the real route would involve crossing walls. Another improvement involves applying visibility techniques for computational geometry to detect corners of rooms and consider them as new nodes (Liu and Zlatanova, 2011), obtaining a more realistic graph into each room. Indoor navigable space, delimited by walls and other obstacles, can be modelled using triangulation techniques, such as Delaunay (Jamali et al., 2017), or diagrams, such as Voronoi (Lamarche and Donikian, 2004). Boguslawski et al., (2016) focus on the use of graph modelling to establish routes in emergencies. In that case, the triangulation has a low density for the outer rooms and increases as it approaches the possible evacuation routes in building centre and emergency exits. Other authors preferred a different type of node distribution in space. In (Czogalla and Naumann, 2015), nodes have a hexagonal distribution to model transfer stations between means of transport. They also locate the obstacles on a map in order to avoid them. Nasir et al., (2014) compare a distribution based on a grid with a triangulation based on navigable space vertices. Different distributions change adjacencies and number of nodes of the generated graph. These models are integrated in a simulation to obtain different possible trajectories for pedestrians. In the urban environment, Y.F. Tang and S.C. Pun-Cheng, (2004) define some elements (buildings, zebra crossing and roads) as polylines, similar to a road network map. By contrast, in this work, only safe and accessible ground elements (sidewalks, stairs and pedestrian crossings) are considered to generate the navigable graph from a mesh that connects the points belonging to those elements. Vertical elements are not used as navigable nodes, but they influence the free unobstructed space of pedestrian space. The input of the methodology is a labelled point cloud of the urban environment.

Serna and Marcotegui, (2013) carry out an physical accessibility study from point clouds, although these are converted into images in the early stages of the methodology. They segment obstacles, such as façades and objects, and consider the entire ground as accessible, except curbs. Soilán et al. (2018) detect curbs at the borders of pedestrian crossings to analyse the accessibility. In indoors, Maruyama et al., (2017, 2016) use simulations from 3D as-in environment models: walk surface points, navigation graphs and textured 3D environmental geometry. With Digital Human Models generated from Terrestrial Laser Scanning data, they check how people move and orientate themselves inside buildings. They analyse visibility and legibility of signals, as well as their motion planning in relation to their motor skills. The methodology proposed in the present work distinguishes between static (obstacles) and dynamic elements and only considers sidewalks, stairs and pedestrian crossings as navigable surfaces (not road). Risers of stairs and curbs are considered accessibility barriers. The present work does not study the visibility or legibility of the environment.

With regard to previous approaches, in this work a graph is directly created from classified point clouds. The main advantages of the direct use of point clouds over theoretical models for the generation of navigable graphs are with the use of real information from urban environments. Final nodes correspond to real and precise locations, existing in the input point cloud. Therefore, there is no risk that the theoretical models do not correspond to the as-built reality. The location and precision of the nodes are no modified during the modelling process. Points belonging to dynamic urban elements are removed. Free unobstructed space is used as navigable space. The nodes of the final graph are distributed in a mesh grid, not in a triangulation or visibility approach, since in point clouds the objects are not defined by simple geometries as in 3D models. In addition, the graph model proposed in this paper has a higher resolution with respect to previous urban modelling works and road networks maps, allowing a precise route for pedestrians.

Preliminary results of this work were presented in (López-Pazos et al., 2017). With respect to them, the following improvements have been made.

- A SVM classifier has been implemented to differentiate non-ground elements into static and dynamic urban elements. Only static urban elements are considered as obstacles that generate gaps in the final graph.
- The simplification of points for the final graph is now carried out through a spatial downsampling, not *k-means*, being faster and leaving nodes more uniformly distributed in a mesh grid (Beneš et al., 2014).
- A methodology for correcting occlusions in sidewalks is proposed. The absence of data on the ground implies an absence of nodes in the final graph, even being able to generate unconnected graphs that do not fit the reality.
- The final graph is generated from the free unobstructed navigable space and not from the entire ground surface, since pedestrians are modelled as a volume in the navigation space.

3. Datasets and overview of urban ground classification

In this section, the three datasets used for the evaluation of this work are described along with the methodologies that allow them to be generated. Both point clouds have labelled ground elements (sidewalks, roads, treads, risers, curbs and crosswalks). Non-ground elements are in an independent class without dividing. Each ground element has characteristics associated with mobility, physical accessibility and safety, therefore, its knowledge is essential for the creation of a graph that allows the correct pathfinding application.

The first point cloud (Fig. 1) corresponds to the intersection between *Humilladero* Street and *Portugal* Avenue in *Avila* (Spain). It is a complex area with a variety of ground elements and with a lot of connections between them. The point cloud contains 20.5 million points and has two strong occlusion zones, one on the ramp that joins the sidewalks of both streets and another produced by cars in *Portugal* Avenue. The second (Fig. 2) is a fragment of *Florida* Avenue in *Vigo* (Spain), it is a straight street with sidewalks on both sides and parked cars on one side of the street. The point cloud has 21.8 million points and occlusions on the sidewalk with parked cars. Both point clouds have been acquired using MLS LYNX Mobile Mapper of Optech (Puente et al., 2013). The third point cloud (Fig. 3) is provided by IQmulus & TerraMobilita Contest dataset (Vallet et al., 2015). It is *Cassete* Street in the city of *Paris* (France), a one-lane street with an intersection, narrow sidewalks and parked cars on both sides that cause occlusions. It contains 12 million points. It has been acquired by Stereopolis II MLS.

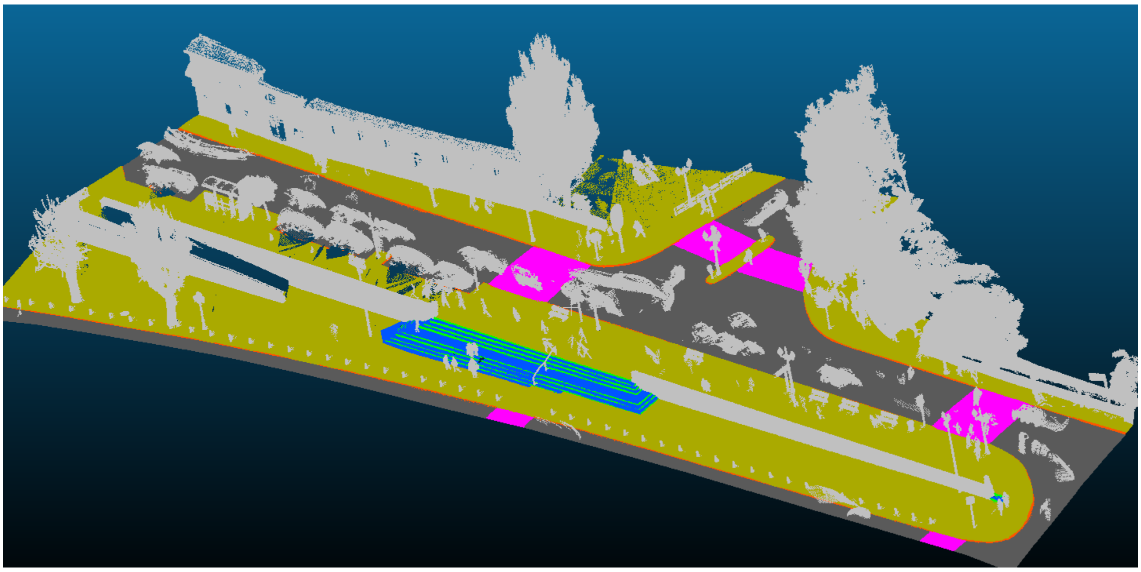


Fig. 1. Classified point cloud of case study 1. Colour code: non-ground elements in grey, sidewalk in olive, road in dark grey, crosswalks in rose, curbs in orange, risers in green and treads in blue. Model scale: 110x47x20 m.

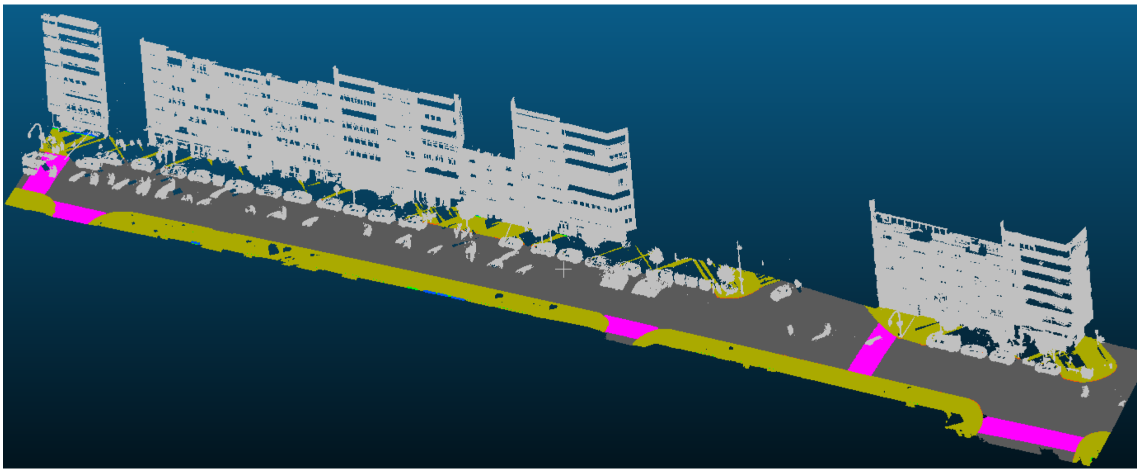


Fig. 2. Classified point cloud of case study 2. Note: the front façade line has been removed from the image to improve visualization. Model scale: 157x145x30 m.

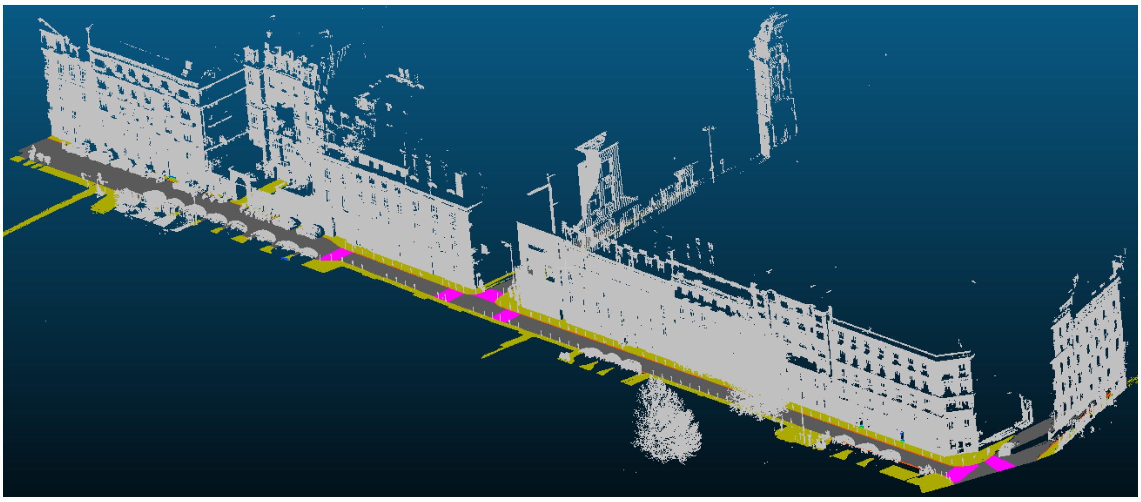


Fig. 3. Classified point cloud of case study 3. Note: the front façade line has been removed from the image to improve visualization. Model scale: 155x215x47 m.

The methodology for the classification of ground elements is collected in (Balado et al., 2018). The input of the methodology is an urban point cloud without RGB-intensity information and the MLS acquisition trajectory. The methodology begins with a planar segmentation based on point cloud curvature, since each ground element can be approximated by a planar element. After planar segmentation, refining operations are performed to obtain a greater level of detail and accuracy between real elements and segmented planar element. The refining operations are: split (separation of real elements segmented in the same planar region), merge (joint of adjacent elements with similar curvature in the same region), coplanar refinement (separation of different elements contained in the same plane, usually, risers that are part of walls) and road-sidewalk segmentation (separation road from sidewalk based on the MLS trajectory and the curvature that delimits roads edges).

Once each real element is segmented into a planar region, a double classification is implemented: first a geometry-based classification followed by a topology-based classification. The geometry-based classification employs a decision tree built with features defined by ISO-21542 (ISO, 2011): tilt, height and width. The objective of the topology-based classification is to differentiate those elements with a similar geometry through their relationships with other elements. The topology-based classification is based on a verification of the adjacency relationships of each geometric element in the point cloud versus predefined relations of each element (previously stored in a graph library). Once the entire process is done, the point cloud is classified into: roads, sidewalks, curbs, treads, risers and non-ground elements.

The methodology for crosswalks detection is available in (Riveiro et al., 2015). The input of the methodology is a point cloud with intensity information. The detection is based on the high intensity of the points belonging to road marks due to reflective painting. The point cloud is rasterized on XY plane and intensity values associated with pixels. Finally, the Hough transform is applied to detect the lines that form zebra crossing.

The result of the application of both methodologies is a point cloud $\mathrm{P} = (\mathrm{X}, \mathrm{Y}, \mathrm{Z})$ where ground elements are labeled L in: sidewalks, roads, curbs, treads, risers and crosswalks. The rest of the elements (urban objects, people, cars, trees and buildings) are in a class defined as non-ground elements. This point cloud is used as the input to the methodology presented in this work.

4. Methodology

The methodology is divided into three main phases. Pre-processing phase prepares the point cloud for the following steps by reducing and standardizing point cloud density and eliminating non-relevant points. Occlusion correction regenerates empty areas in sidewalk in where there are no points due to the presence of elements between the sidewalk and the MLS during the acquisition such as parked cars. Finally, the transited areas are delimited by adjacency with other elements and they are modelled in a graph. The workflow of the methodology is shown in Fig. 4.

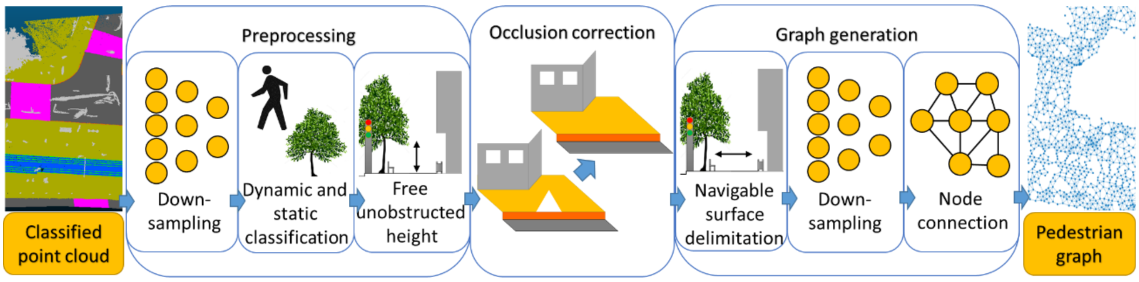


Fig. 4. Workflow of the methodology.

4.1. Pre-processing

The pre-processing consists of two objectives, a reduction and uniformisation of point cloud density and the elimination of points belonging to obstacles without interest for people mobility.

4.1.1. Downsampling

The first phase is a downsampling, since the amount of points in the cloud is excessive for the creation of the final graph and there are strong changes in density between near and fast areas to the MLS trajectory. Density uniformisation is achieved by reducing the number of points based on the distance between them (Pomerleau et al., 2013). The remaining points are at distance d_1 from their neighbours, being $d_1 > d_{initial\_point\_cloud}$. With this operation, processing time in the following processes are reduced.

4.1.2. Dynamic and static classification

Not all points classified as non-ground elements have the same utility for pedestrian pathfinding. Elements that have a static behaviour are obstacles to navigation, such as lampposts, benches, mailboxes, etc.; while those that have a dynamic behaviour are only found in the scene during the acquisition, such as pedestrians walking or motorcycles and cars circulating. Dynamic elements change place or disappear, therefore they are not considered as obstacles. The differentiation between static and dynamic elements can be carried out as an object classification, a well-studied subject in urban environment and in point clouds (Serna and Marcotegui, 2014; Vallet et al., 2015; Yang et al., 2017, 2015), or as a change detection (Chen and Yang, 2016; J Schauer and Nüchter, 2018; Johannes Schauer and Nüchter, 2018; Xiao et al., 2016, 2015). In this paper, object-based classification is used (Aijazi et al., 2013; Huang and You, 2015), since different multi-temporal observations are not needed. The objects considered as static are: buildings, parked cars (although they are a dynamic element, they occupy a fixed place in the scene that is rarely empty), urban furniture, trees and pole-like objects (traffic lights, lamps and traffic signals). The dynamic elements are cars in motion, pedestrians, bikes and motorcycles. Table 1 shows the classification of the main types of elements in the urban scene: the branch of the classified ground elements, and the branch of non-ground elements in which static and dynamic must be differentiated.

Table 1: Classification of elements in the urban scene.

Ground elements	Passable elements	Accessible elements	Sidewalks
			Crosswalks

		Non-accessible elements	Treads
	Non-passable elements	Accessible barriers	Risers
			Curbs
		Horizontal elements	Roads
Non-ground elements	Static elements		Buildings
			Parked cars
			Urban furniture
			Trees
			Pole likes objects
	Dynamic elements		Circulating cars
			Pedestrian
			Bikes and motorbikes

In this work, a machine learning (ML) classifier based on Support Vector Machines (SVM) is used (Mountrakis et al., 2011), because it is a classification technique that achieves high accuracy rates by using few features and training with a relatively low number of objects, in comparison with other techniques such as deep neural networks (Serna and Marcotegui, 2014). For the classification, the features defined by (Roynard et al., 2016) are used: height, standard deviation height, width, distribution of points based on histogram of five bins, area and volume of convex hull.

The objects to be classified are points labelled in the point cloud Pd as non-ground class. The objects are individualized by connected components (Trevor et al., 2013) and from each one the features are extracted. From these features the objects are classified with the SVM classifier as dynamics or static objects. The points classified as belonging to dynamic objects are removed from point cloud, while those of static objects should be refined in the next step.

4.1.3. Free unobstructed height

Static obstacles may interfere with people movement by two ways: based on free unobstructed width (discussed in section 4.3) and on free unobstructed height. Free unobstructed space establishes a minimum height h for pedestrians to comfortably transit, therefore, the points of greater height than h of each element have no interest for this study. The points that exceed the height h of each object are removed from point cloud. For each object previously classified as static, the lowest Z coordinate z_{min} is calculated. Then, points with $z > z_{min} + h$ are removed. The remaining points are returned to the main cloud leaving a refined cloud for the following processes Pr.

4.2. Occlusion correction

This phase allows point generation in highly occluded sidewalk areas, because of all objects (mainly cars) between sidewalks and MLS trajectory. The regeneration of occlusions assuming the planarity is not as a reliable solution as the use of data without occlusions, but based on prior knowledge of the urban environment, it is a better alternative than obtaining an incomplete graph or the need for multiple scans with different acquisition methods. The process of correction of these zones is based on a region growing of sidewalks controlled and delimited mathematical morphology.

The developed pseudo-code is indicated in Algorithm 1. This process begins with the rasterization (Díaz-Vilariño et al., 2015) of the previously refined point cloud Pr (Fig. 5.a) with a grid size gs of twice the size of d distance $gs = 2d_1$ between downsampled points performed in Section 4.1. This grid size allows all pixels to be populated with points and there are not a lot of empty pixels, except those that form properly occlusions. In each pixel of the rasterized image I, the mode value of the labels L of existing points is saved (Fig. 5.b).

Once the point cloud is rasterized, it is necessary to create separated binary images that correspond to the sidewalks IS (Fig. 5.c) and the rest of the elements IE (Fig. 5.d). Likewise, a global binary image IL is created that delimits the existence of the point cloud in the image (Fig. 5.e). IL combined with IE suppose the mask image IM that limits the growth of sidewalks image (Fig. 5.g). In this way, it is ensured that the growth of the sidewalks does not exceed the external limit of the point cloud or elements existing within it. In the next phase, within a loop, IS is expanded through morphological dilation (Jackway and Deriche, 1996) ISd (Fig. 5.f) and the pixels that coincide IM are removed (Fig. 5.h). This loop is executed while IS contains empty pixels where it could grow without conflict with IM. At the end of the loop, the resulting image ISn is the complete sidewalks image (Fig. 5.i).

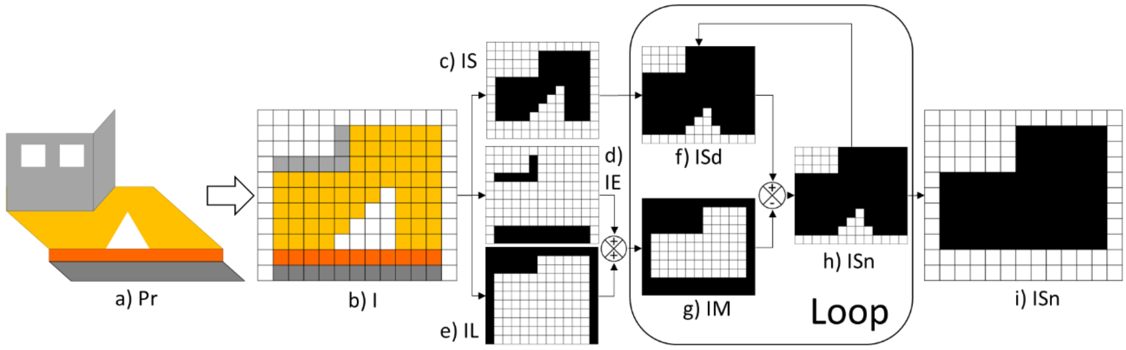


Fig. 5. First part of the occlusion correction: a) Refined point cloud Pr, b) point cloud rasterized by mode I, c) binarised sidewalk image IS, d) binarised image of other elements IE, d) binarised image of the contour IL, f) dilated sidewalk image ISd, g) mask image IM, h) sidewalk dilated image after subtract mask image ISn, i) at the end of the loop, binarised complete sidewalk image ISn.

By subtracting IS (Fig. 6.b) of ISn (Fig. 6.a), the image with only occlusions is obtained IO (Fig. 6.c), to which a morphological aperture is applied to eliminate the small ones and to focus on the large occlusions. In order to individualize occlusions, connected components are applied. Each set of pixels belonging to an occlusion IOcc is dilated (Fig. 6.d) and sidewalk points PS that belong to occlusions contour are searched in the refined point cloud Pr (Fig. 6.f). PS is structured in a polygon Pol and, inside, new random points PSnXY are generated with XY coordinates (Fig. 6.e) with density similar to Pr. For the generation of Z coordinate, a multiple linear regression model (Preacher et al., 2006) is used (Fig. 6.g). Z coordinates of new points PSnZ are calculated as a linear function of PSnXY from XYZ coordinates in PS. This allows the new sidewalk points to present a coherent inclination with the limits of the occlusion. At last, PSnXY and PSnZ are saved as new points in R (Fig. 6.h).

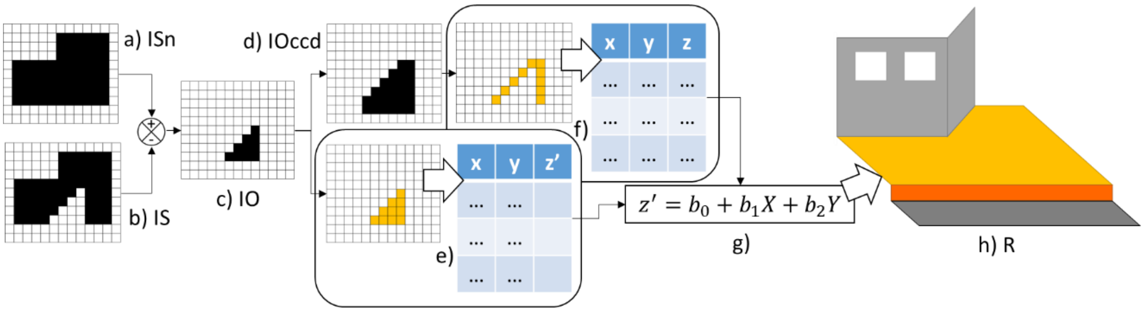


Fig. 6. Second part of occlusion correction: a) complete binarised sidewalk image ISn, b) initial binarised sidewalk image IS, c) binarised image of occluded sidewalk IO, d) dilated image of occluded sidewalk IOccd, e) generation of random XY points in the occlusion, f) extraction of XYZ points corresponding to the occlusion border in point cloud Pr, g) implementation of the multiple linear regression to complete the Z' coordinate of points generated randomly in the occlusion, h) occlusion correction and insertion of new points in the new refined cloud R.

***Algorithm 1:* Occlusion correction**

```
Inputs: Refined_Point_Cloud {Pr}, Labels {L}, grid_size gs
Outputs: Regenereted_Point_Cloud {R}

Raster_image {I} ← Raster (P,L,gs)
Sidewalk_image {IS}, Rest_of_Elements_image {IE} ← binary (I)
Limit_image {IL} ← fill (I)
Mask_image {IM} ← complement (IL) + IE
Sidewalk_image_prev {ISp} ← zeros (IS)
Sidewalk_image_new {ISn} ← IS
While ISp ≠ ISn do
        ISp ← ISn
        Sidewalk_image_dilated {ISd} ← dilate (ISn)
        ISn ← ISd – IM
End_While
Occlusion_image {IO} ← morphological_opening (ISn - IS)
CC_oclusion_image {IOcc} ← ConnectedComponents (IO)
R ← Pr
For each IOcc(i)
        IOcc_dilated { IOccd} ← dilate (IOcc (i))
        IOcc_dilated_Sidewalk { IOccdS} ← IOccd - IE
        Sidewalk_Points {PS} ← {P : P ∈ IOccdS}
        Polygon {Pol} ← polygon{PS}
        New_Sidewalk_Points_XY {PSnXY} ← random (2*gs,inpolygon(Pol))
        New_Sidewalk_Points_Z {PSnZ} ← linear_regresion_model (PSnXY, PS)
        R ← Add (PSnXY, PSnZ)
End_For
Return {R}
```

4.3. Graph generation

In this phase, the final graphs are generated from the point cloud belonging to passable elements, delimited by non-passable elements and static objects bounded by the free unobstructed height. For the generation of the final graphs, the points of those elements that are passable must be selected as nodes. These elements are sidewalks, crosswalks and treads for pedestrian without reduced mobility graph, and only sidewalks and crosswalks for wheelchairs graph. Roads, although they are physically accessible elements, are not safe areas to walk. From these elements, points less than a free unobstructed width *w* to obstacles (non-ground static elements) and roads are subtracted. For the wheelchair graph, accessibility barriers (risers and curbs) are added as delimiters of free unobstructed space (Fig. 7). Free unobstructed space is defined in ISO-21542 as the minimum space between elements that a person can pass comfortably. It is different for pedestrians *wp* and for wheelchairs *ww*. Once the navigable surfaces for pedestrians and accessible surfaces for wheelchair are defined, the downsampling generates the nodes of the final graph. The process for the graph generation is as follows and the pseudo-code used is collected in Algorithm 2.

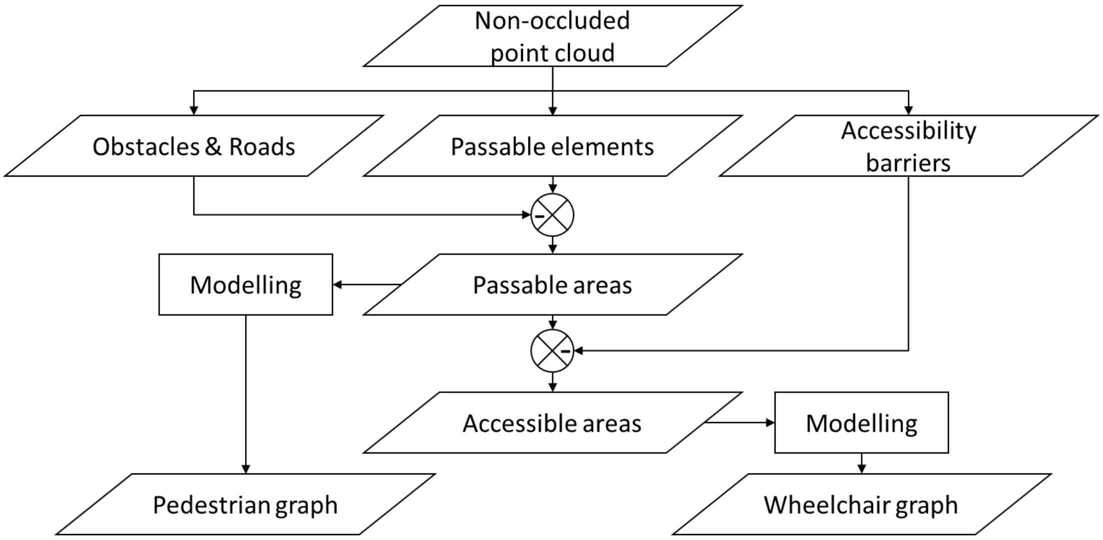


Fig. 7. Interaction between elements during the graph generation phase.

4.3.1. Navigable surface delimitation

First, the elements that make up the cloud are separated and grouped into passable surface elements Sp (sidewalks, crosswalks and treads) and accessible Sa (sidewalks and crosswalks). The rest of the elements (roads, risers, curbs and obstacles) are individualized by connected components. Because the passable elements are distributed horizontally, the rest of the elements are projected onto the XY plane and the contour *B* is calculated. To obtain the free unobstructed space, the contour of each element is dilated by a distance *w*/2 and subtracted from the passable surfaces (Fig. 8).

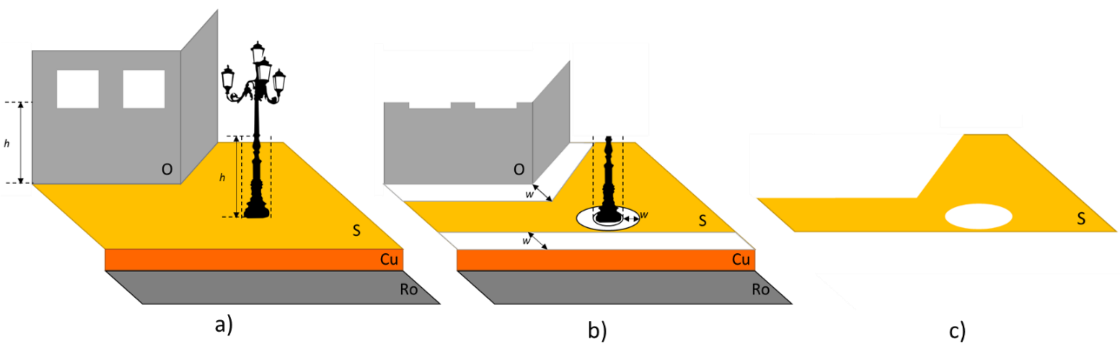


Fig. 8. Subtraction of areas near elements to calculate the free unobstructed space: a) point cloud with static objects and calculation of free unobstructed height *h*, b) projection of the limited objects by *h*, roads and

accessible barriers over the plane XY and dilation $w/2$ of the contour, c) free unobstructed space of sidewalk after substring element dilations.

Since there are different free unobstructed widths defined by ISO-21542 and different elements of influence, contour dilations and subtractions are made separately for the two final graphs. For the pedestrian graph, only contours of static obstacles and roads are dilated with distance *wp* and are subtracted from Sp. For wheelchair graph, static obstacles, roads, curbs and risers are dilated with distance *ww* and subtracted from Sa.

4.3.2. Downsampling

Once surfaces navigable by pedestrians and accessible by wheelchairs are delimited, the downsampling that leaves the final nodes Np and Na of the graphs is applied. This downsampling leaves a distance *nd* between nodes that approximately follows a grid distribution. *nd* must be sufficient for future trajectories can be followed without an over-saturation of nodes and to ensure that there are no connections between nodes which there may be an obstacle.

4.3.3. Node connection

Finally, nodes are connected and final graphs are generated. The number of arcs in final graphs depends on the distance *gd* that relates the nodes (Fig. 9) (Bunn et al., 2000). If $gd < nd$, there is no adjacency between nodes; if $gd \approx nd$, there is adjacency with four nodes; and if $gd \approx nd\sqrt{2}$, there is adjacency with 8 nodes. The latter one is used for the generation of final graphs Gp and Gw. The value of each arc in the final graphs is the Euclidean distance between the corresponding nodes.

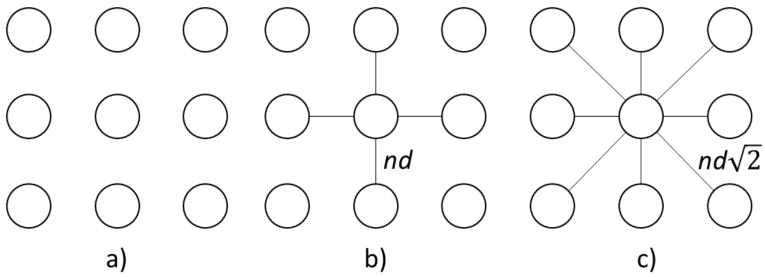


Fig. 9. Relations between the nodes based on the creation distance of the graph: a) $gd < nd$, b) $gd \approx nd$, c) $gd \approx nd\sqrt{2}$.

***Algorithm 2:* Graph generation**

Inputs: Regenereted_Point_Cloud {R}, node_distance *nd*, unobstructed_width_pedestrians *wp*, unobstructed_width_wheelchairrrs *ww*
Outputs: Graph_pedestrians {Gp}, Graph_wheelchairs {Gw}

Sidewalks {S}, Crosswalks {C}, Treads{T} ← R (X,Y)
Roads {Ro}, Risers {Ri}, Curbs{Cu}, Obstacles {O} ← R (X,Y)
Surface_points_unobstructed_passable {Sp} ← (S,C,T)
Surface_points_unobstructed_acessible {Sa} ← (S,C)
Obstacles_passable {Op} ← ConnectedComponents (Ro,O)
Obstacles_acessible {Oa} ← ConnectedComponents (Ri,Cu)

For each Op (i)
  Border {B} ← boundary (Op(i))
  Sp ← Sp - inpolygon (buffer (B,*wp*))
  Sa ← Sa - inpolygon (buffer (B,*ww*))
End_For
For each Oa (i)
  Border {B} ← boundary (Oa(i))
  Sa ← Sa - inpolygon (buffer (B,*ww*))
End_For
Nodes_passable {Np} ← downsampling (Sp,*nd*)
Nodes_accessible {Na} ← downsampling (Sa,*nd*)
Gp ← points2graph (Np,$nd\sqrt{2}$)
Gw ← points2graph (Na,$nd\sqrt{2}$))
Return {Gp, Gw}

5. Experiments

In this section, the results of the application of the methodology to three case studies are presented and analysed.

5.1. Results of methodology application

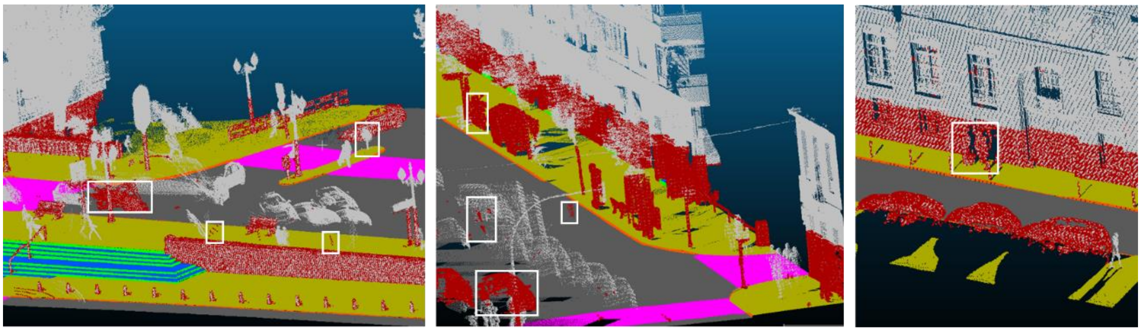
The methodology is composed of several phases whose results are shown below. In order to apply the methodology and guarantee its reproducibility, the parameter values must be established (shown in Table 2). Values *h*, *wp* and *ww* are defined by ISO-21542, while value *d* has been set based on the recommendations described in Section 4.1.1. The value *nd* is based on processing times and surface free widths (in detail in Section 5.3).

Table 2. Parameter values.

Section and parameter	Abbreviation	Value
4.1.1. Donwsampling distance	*d*	0.05 m
4.1.3. Free unobstructed height	*h*	2.1 m
4.3. Free unobstructed width for pedestrians	*wp*	0.8 m
4.3. Free unobstructed width for wheelchairs	*ww*	1 m
4.3. Distance between nodes	*nd*	0.5 m

In the pre-processing phase, the point cloud density is standardized and not relevant points for graph generation are eliminated. In Fig. 10, removed points are coloured in grey and non-ground remaining point are coloured in red. As can be seen, most elements are well classified, but not all, misclassified elements are highlighted in white boxes. The classifier implemented in the methodology must be previously trained. For that purpose, 906 non-ground elements of a point cloud acquired on *Camelias* Street in *Vigo* with the same MLS have been individually and classified manually. The number of non-ground elements for training is collected in Table 3 and the features of each object have been extracted (Section 4.1.2). After training the SVM classifier with cross-validation (Golub et al., 1979; Kohavi, 1995), the accuracy of the system is

94.5%, enough for this job since it is not the main objective. Even so, the misclassified elements have relevance in the final graph.



a) b) c)

Fig. 10. Fragments of classified point clouds by elements with points belonging to static elements at a height less than *h* in red for the case of study 1 (a), 2 (b) and 3 (c). Misclassified elements are highlighted in white boxes.

Table 3. Number of elements used to train the classifier.

Static elements	**593**
Buildings	23
Parked cars	150
Trees	203
Urban furniture	113
Pole-like objects	104
Dynamic elements	**313**
Cars in motion	137
Pedestrians	153
Bikes and motorbikes	23

Once the pre-processing phase is completed, the correction of occlusions is implemented. Fig. 11 shows the comparison before and after the occlusion correction for the most important areas of the datasets.

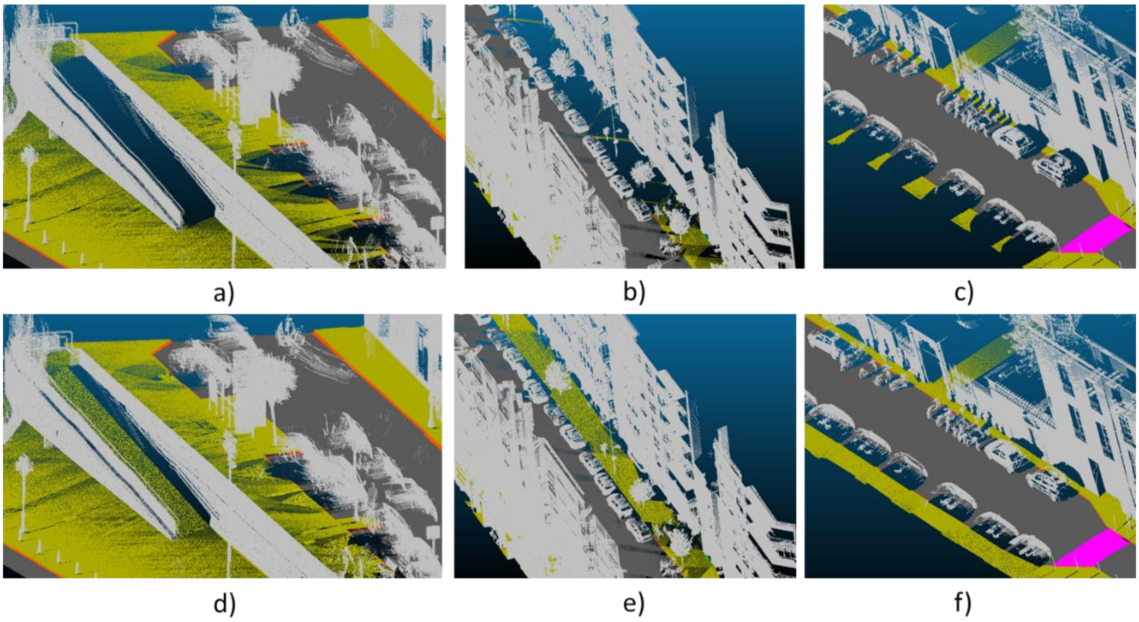


Fig. 11. Result of the application of the occlusion correction to point clouds before the correction: case study 1 (a), case study 2 (b) and case study 3 (c); and after correction case study 1 (d), case study 2 (e) and case study 3 (f).

Once the occlusions have been corrected, navigable space is delimited for graph generation, according to the motor skill of pedestrians and wheelchairs, free unobstructed height width *wp* and *ww*, and passable elements. Fig. 12 shows the classified point cloud (Fig. 12.a), with navigable surface for pedestrian in red (Fig. 12.b) and for wheelchairs in white (Fig. 12.c) respecting the distance (*wp* and *ww*) to static objects. As can be seen, the stairs are not considered navigable surface for wheelchairs. The last phase is the generation of the nodes of the graph from a last downsampling. The nodes are uniformly distributed at a distance *nd* from each other on passable surfaces. Fig. 13 shows the nodes with different *nd* for pedestrian graph in a fragment of case study 1.

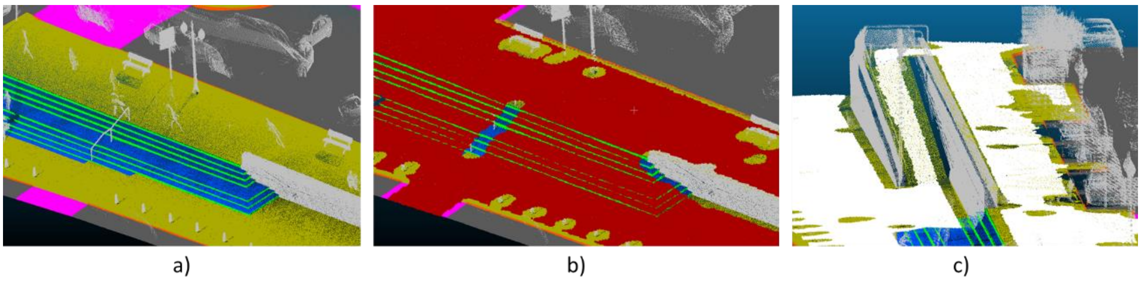


Fig. 12. Navigable surface in the case of study 1: a) classified point cloud, b) pedestrian passable surface in red and c) wheelchair passable surface in white.

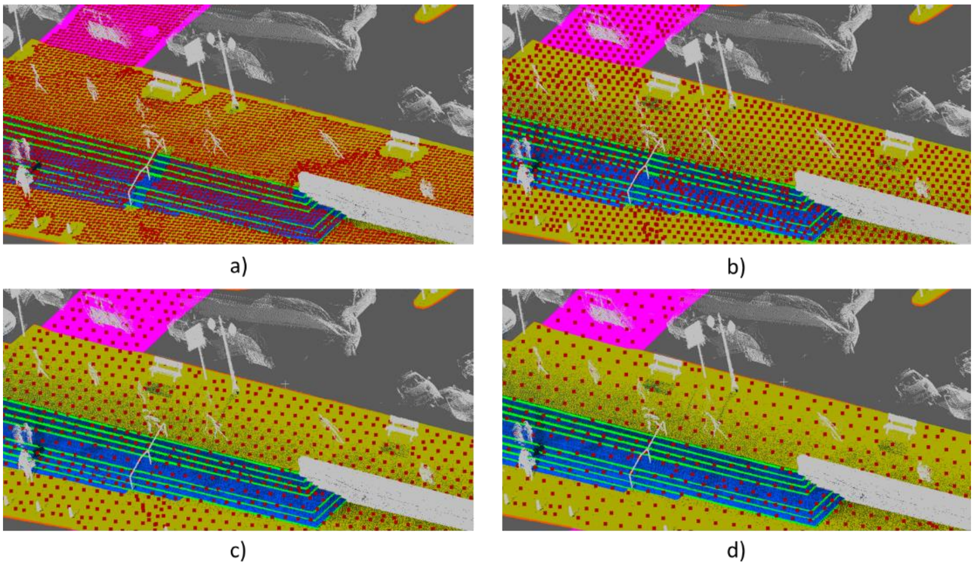


Fig. 13. Distribution of the nodes (red) on the passable surface by pedestrians in the case of study 1 with *nd* = 0.25 m (a), *nd* = 0.5 m (b). *nd* = 0.75 m (c) and *nd* = 1 m (d).

An erroneous classification of objects into static and dynamic causes errors in the generation of the navigable surface. Static false positives subtract zones from the navigable surface, while dynamic false positives add it. The classifier used has misclassified 47 objects as static and 1 as dynamic. The static false objects have subtracted a total of 87.9 m^2 and 72.1 m^2 from the navigable surface by pedestrians and wheelchairs respectively, while the dynamic false objects have generated 0.6 m^2 and 0.1 m^2. The total area correctly generated was 5400 m^2 for pedestrians and 3950 m^2 for wheelchairs, which quantifies the error caused by the classification at 1.6% for pedestrian and 1.8% for wheelchairs navigable surfaces (detailed in Table 4).

Table 4. Relation between object classification and generated navigable surface.

Case study	User	Generated navigable area (m^2)	Static false positives	Lost navigable area (m^2)	Dynamic false positives	Added navigable area (m^2)	Erroneous area
CS1	Ped.	1835	24	40.6	1	0.6	2.2%
	Wheel.	1714		32.6		0.1	1.9%
CS2	Ped.	2687	13	33.5	0	0	1.2%
	Wheel.	1511		30.4		0	2.0%
CS3	Ped.	878	10	13.8	0	0	1.5%
	Wheel.	725		9.1		0	1.2%
TOTAL	**Ped.**	**5400**	**47**	**87.9**	**1**	**0.6**	**1.6%**
	Wheel.	**3950**		**72.1**		**0.1**	**1.8%**

Through the phases, a continuous reduction of points is obtained, from the initial point cloud to the final graph. Its objective is to obtain a better performance of the methodology and the calculation of final routes, discarding non-relevant points. For the case studies presented in this paper, with *nd* = 0.5 m, the final number of nodes in the pedestrian graph is 8059 (case study 1), 5304 (case study 2) and 3341 (case study 3); and for wheelchairs 6826, 4904 and 2660 respectively. Only 0.1% of ground points of each point cloud is used in the generation of the final graph.

The methodology has been implemented in Matlab and the three case studies has been processed on an Intel Core i7-7700HQ CPU 2.80 GHz with 16GB RAM. The details of processing times are reflected in Table 5. The processing time is good for the case study 1 and 3 with 412 seconds and 416 seconds respectively (7 minutes approximately), and acceptable in the case study 2 with 1578 seconds (26 minutes approximately). Although the first two datasets have a similar size, this time difference is due to the large number of points in the non-ground elements class, 8.2 million points vs. 11.6 million points respectively, and the geometry of the street. The operations that consume more time are those that involve the use of connected components to large amounts of points, that are repeated in the pre-processing and in graph generation.

Table 5: Processing times of the methodology application to the case studies.

	Number of points	**Pre-processing**	**Occlusion correction**	**Graph generation**	**Total**
CS1	20.5M	223.6 s	21.1 s	167.3 s	412.0 s
CS2	21.8M	472.5 s	68.1 s	1038.2 s	1578.8 s
CS3	12.0M	254.9 s	38.2 s	123.1 s	416.2 s

5.2. Results of pathfinding application

To test the viability of the graphs generated with the methodology, the Dijkstra algorithm is used to calculate routes in a series of points of origin and destination. Dijkstra algorithm is a simple and well-known algorithm used in many applications (Kang et al., 2008; Ngoc Nha et al., 2012; Soltani et al., 2002).

The time in which the algorithm finds a correct route is directly related to the number of nodes, this parameter is selectable by *nd* in the methodology. Table 6 shows the relationship between the time it takes for the algorithm to find a route and the distance between nodes *nd* for a start and end node located at a linear distance of 100 m. All *nd* distances guarantee the application of the real-time pathfinding, although an increase in time is observed decreasing *nd*. Considering the time increase in low *nd* and distances close to *wp* and *ww* do not guarantee a correct obstacle representation in the final graph, *nd* = 0.5 m have chosen for the final graph generation.

Table 6: Relationship between the distance between nodes nd and the time to calculate a route with Dijkstra algorithm.

***nd* (m)**	1	0.75	0.5	0.25	0.1
time (s)	0.0160	0.0194	0.0357	0.0398	0.0758

The results for the routes calculated by Dijkstra algorithm are shown in Fig. 14 for case study 1, Fig. 15 for case study 2 and Fig 16. for case study 3. In Fig. 14, a distinction is made between pedestrian routes without reduced mobility and wheelchairs. Pedestrian routes (Fig. 14.a and Fig 14.c) follow a direct route crossing stairs, by contrast, equivalent routes for wheelchairs (Fig. 14.b and Fig 14.d) deviate until find no accessibility barriers. The proposed routes follow the shortest suitable trajectory according to their motor skill, avoiding obstacles and following a safe course, crossing road by crosswalks and not by dangerous areas. This can be seen especially in Fig. 15, where the proposed route joins one sidewalk to the opposite one crossing the road by the crosswalks. The routes shown in Fig. 15 are for pedestrians since there are no accessibility barriers in the area. The routes shown in Fig. 16 correspond to case study 3. Fig. 16.a shows the generated route for the displacement of a pedestrian. The route cannot run on one sidewalk because its width is insufficient according to ISO-21542, so there is no navigable surface on it (Fig 16.d). Fig. 16.b shows a possible route for crossing the street. The route shown in 16.c shows a small wheelchair route from a crosswalk to the opposite sidewalk. Fig. 16.e and Fig. 16.f show the routes taking into account the navigable surface for pedestrians and wheelchairs. It can be seen as one of the sidewalks is not accessible due to the proximity between the bollards. For the same reason, the route takes different layouts for pedestrians and wheelchairs on the opposite sidewalk. In addition, all the generated routes do not collide with any static object and cross areas that in the initial cloud did not have points such as the ramp in case study 1 and the sidewalk in front of parked cars in case study 2.

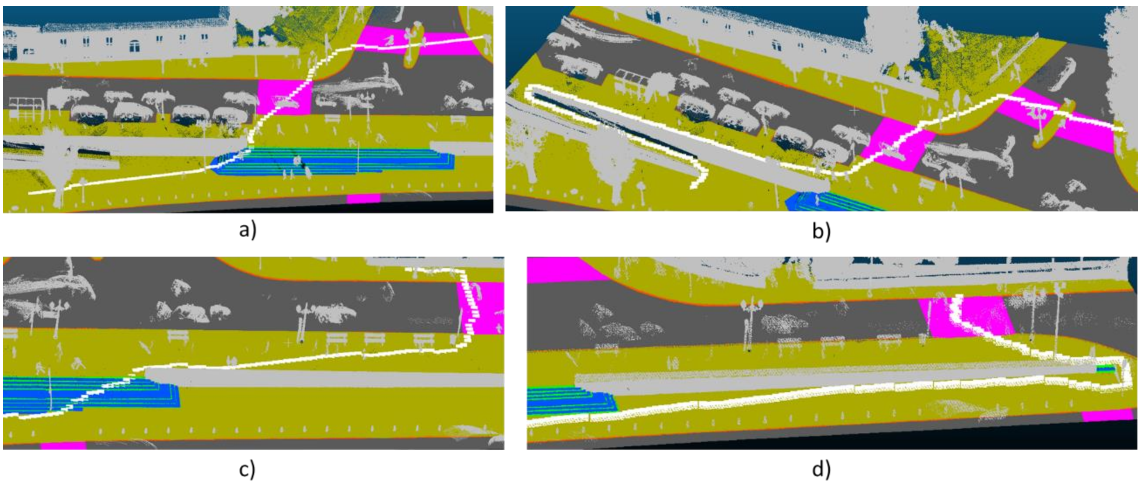


Fig. 14. Pedestrian routes (coloured in white) obtained with the application of Dijkstra algorithm in graphs generated by the proposed methodology over point cloud of case study 1: a) route for pedestrians, b) equivalent route for wheelchairs, c) route for pedestrians and d) route for wheelchairs.

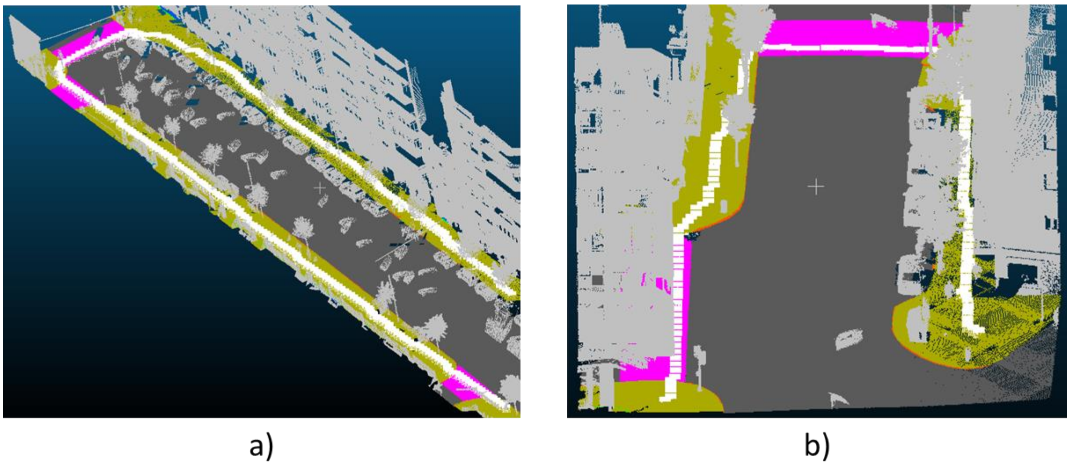


Fig. 15. Pedestrian routes (coloured in white) obtained with the application of Dijkstra algorithm in pedestrian graph generated by the proposed methodology over point cloud of case study 2.

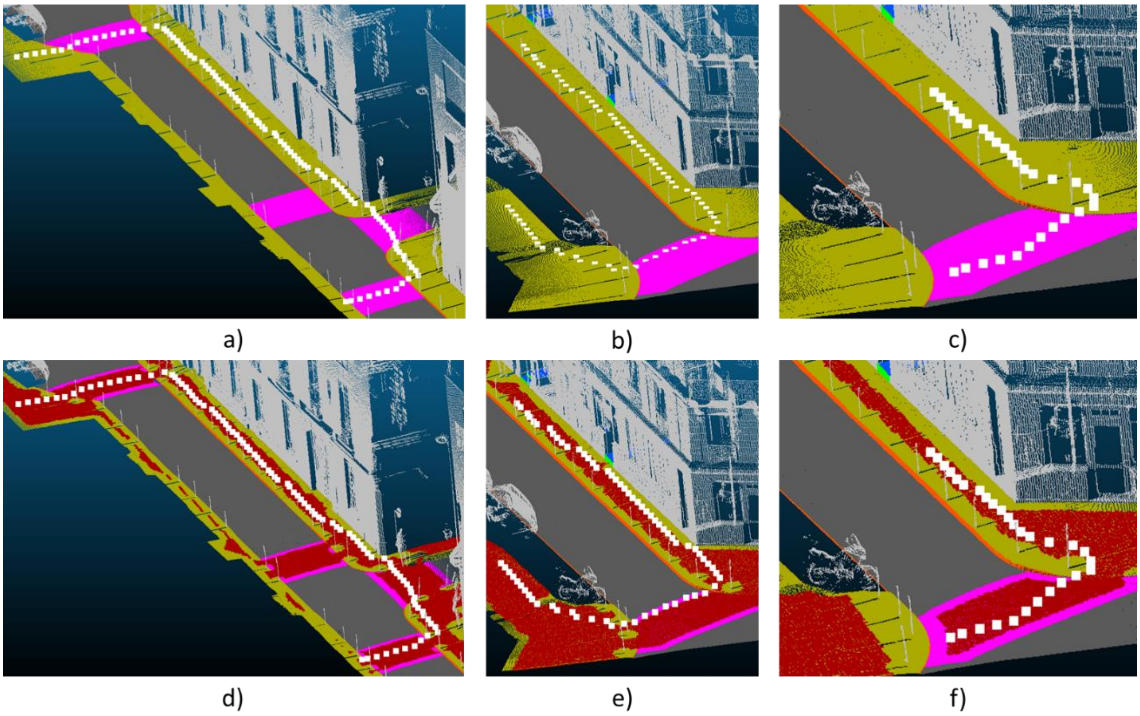


Fig. 16. Routes (coloured in white) obtained with the application of Dijkstra algorithm in graphs generated by the proposed methodology over point cloud of case study 3 with navigable surface in red: a-b) pedestrian route, c) wheelchair route, d-e) pedestrian route over navigable pedestrian surface, f) wheelchair route over navigable wheelchair surface.

5.3. Discussion

Routes generated by the Dijkstra algorithm are perfectly valid for use by both pedestrians and wheelchairs, adapt to ground elements, avoid static obstacles and do not take into account the dynamics that appear in point clouds. Routes can run through areas where input cloud had no points (as seen in Fig. 14 and Fig. 15) while taking into account regulations to establish a minimum free unobstructed space so that people can transit comfortably. In addition, while the generation of the graph involves processing of minutes, the generation of the path with Dijkstra algorithm can be done in real time.

The *nd* variable is the one that most influences the number of nodes in the final graph. A smaller *nd* distance allows a more detailed map, it increases the complexity of the graph and the time in route calculation. In the results presented in this work has been selected an nd = 0.5 m, because it has not seen an improvement in the use of a more detailed graph due to the inability of pedestrians to follow a route with few centimetres in an urban environment. By contrast, a larger size (more than the free unobstructed widths defined by ISO-21542) can lead to omitting small static elements, making the graph lose reality with the as-built environment, one of the main objectives of the work.

The results show a robust behaviour of the proposed methodology, but it is not exempt from some limitations. The generated graph at the end of the methodology depends on these two processes: object classification and occlusion correction. Errors in graph generation will result in erroneous routes.. The classification of non-ground elements in static and dynamic is preceded by a phase of individualization. If objects are very close to others, they may not be identified correctly and they are classified as one. It is common for this error to occur with dynamic elements that are interacting with statics or near them (Fig 17.a). In addition, the classification between static and dynamic is not perfect and there may be misclassified elements. The authors have chosen to use a classifier with few parameters to gain processing time and to only need XYZ coordinate information in its implementation. Errors in individualization and classification mean that the graph does not truly fit the built environment and holes appear on the navigable surface (Fig 17.b). Although the erroneous area accounts for less than 2% of the navigable area generated in the case studies (Table 4), each misclassified object carries an error in the navigable area of approximately 1.8 m^2. In certain locations, such as narrow sidewalks, a hole can break the continuity in the graph and lead to major changes in the generated route. Regarding the occlusion correction, the algorithm is only effective when occlusions are not at the border of the point cloud (Fig 17.c). If the occlusion is at the border, the zone is considered as an external part and no points are generated in it, therefore, it is not represented in the final graph either. In addition, occlusions can hide small static objects, which are not regenerated in the process of occlusion correction and represent a false reality. These two limitations have occurred in specific cases, not being common occurrence. An alternative to the use of an SVM for static and dynamic object classification and occlusion regeneration could be the acquisition of clouds at different multi-temporal observations, but this entails more expense in the acquisition of the built environment. In addition, while it would eliminate dynamic objects and some occlusions, others caused by static objects will continue to exist.

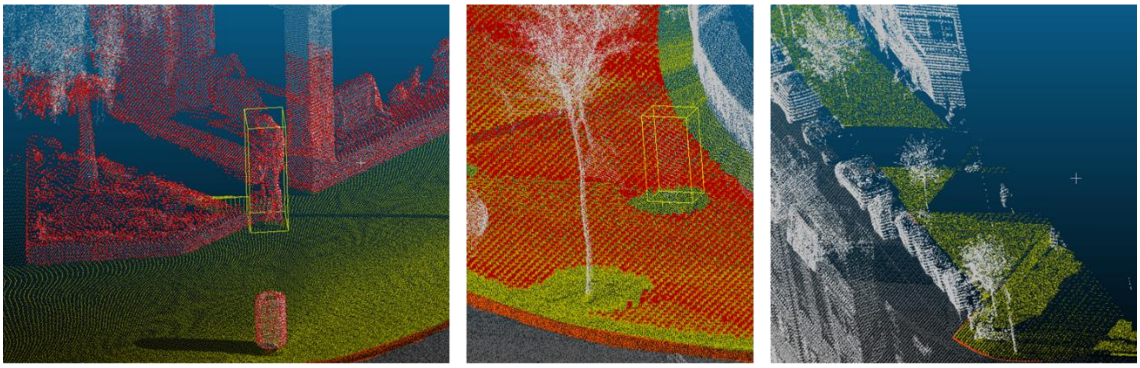


Fig. 17. Example of limitations in the application of the methodology: a) dynamic element merged to a static element, b) holes in the passable surface caused by a correct classification (tree) and an incorrect classification (dynamic object confused as static), and c) uncorrected occlusion at the point cloud border in the case study 2.

6. Conclusions

In this paper, a methodology for the direct use of point clouds for pathfinding is presented. As a result, a graph representing in detail the urban environment is automatically generated, providing a very accurate pathfinding solution for both pedestrians and wheelchairs. The methodology has been designed to solve the main limitations presented by use point clouds in real pathfinding:

- The large number of points is reduced through successive spatial downsampling and density standardization.
- A distinction is made between static and dynamic elements in the scene, considering static elements as obstacles and discarding dynamic ones.
- Occlusions presented in point clouds are corrected and no-data areas regenerated with new points.
- The navigable surfaces are selected and their free unobstructed space delimited according to their interaction with obstacles and physical accessibility barriers.

The methodology has been tested in three case studies acquired in real urban environments and their navigation graphs are generated successfully. The routes are obtained in real time by applying Dijkstra algorithm to the navigation graphs. The generated routes are perfectly viable, are adjusted to ground elements, respect safety distances with obstacles and take into account accessibility barriers. In summary, the proposed routes present a high level of detail at the same time that they are safe and viable according to two motor skills (pedestrians and wheelchairs). Therefore, point clouds have proven to be optimal input data for pedestrian pathfinding in urban environments because: they provide real data of the environment, therefore there are routes based on no theoretical models; it is possible to work on 3D routes and not only on flat maps; and the accuracy of the data is sufficient to detect obstacles and include them in the navigation graph.

Future work will focus on improving the classification between static and dynamic elements with Deep Learning techniques, since an incorrect differentiation reduces realism to the final model. Also the optimization and export for integration of the generated models in GIS (Geographic Information Systems) will be evaluated with the aim of reaching more users.

7. Acknowledgements

Authors would like to thank to the *Universidade de Vigo* for the financial support (00VI 131H 641.02), the *Xunta de Galicia* given through human resources grant (ED481B 2016/079-0) and competitive reference groups (ED431C 2016-038), and the *Ministerio de Economia, Industria y Competitividad -Gobierno de España-* (TIN2016-77158-C4-2-R, RTC-2016-5257-7). This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769255. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains. The statements made herein are solely the responsibility of the authors.

8. References and Notes

Aijazi, A.K., Checchin, P., Trassoudaine, L., 2013. Segmentation based classification of 3D urban point clouds: A super-voxel based approach with evaluation. Remote Sens. 5, 1624–1650. https://doi.org/10.3390/rs5041624

Applegate, C., Laycock, S., Day, A., 2010. Real-Time Traffic Simulation Using Cellular Automata., Theory and Practice of Computer Graphics 2010, TPCG 2010 - Eurographics UK Chapter Proceedings. https://doi.org/10.2312/LocalChapterEvents/TPCG/TPCG10/091-098

Balado, J., Díaz-Vilariño, L., Arias, P., González-Jorge, H., 2018. Automatic classification of urban ground elements from mobile laser scanning data. Autom. Constr. 86, 226–239. https://doi.org/https://doi.org/10.1016/j.autcon.2017.09.004

Beneš, J., Wilkie, A., Křivánek, J., 2014. Procedural modelling of urban road networks. Comput. Graph. Forum 33, 132–142. https://doi.org/10.1111/cgf.12283

Boguslawski, P., Mahdjoubi, L., Zverovich, V., Fadli, F., 2016. Two-Graph Building Interior Representation for Emergency Response Applications. ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. III-2, 9–14. https://doi.org/10.5194/isprsannals-III-2-9-2016

Bunn, A.G., Urban, D.L., Keitt, T.H., 2000. Landscape connectivity: A conservation application of graph theory. J. Environ. Manage. 59, 265–278. https://doi.org/https://doi.org/10.1006/jema.2000.0373

Butenuth, M., Burkert, F., Schmidt, F., Hinz, S., Hartmann, D., Kneidl, A., Borrmann, A., Sirmacek, B., 2011. Integrating pedestrian simulation, tracking and event detection for crowd analysis, in: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops). pp. 150–157. https://doi.org/10.1109/ICCVW.2011.6130237

Chen, C., Yang, B., 2016. Dynamic occlusion detection and inpainting of in situ captured terrestrial laser scanning point clouds sequence. ISPRS J. Photogramm. Remote Sens. 119, 90–107. https://doi.org/https://doi.org/10.1016/j.isprsjprs.2016.05.007

Czogalla, O., Naumann, S., 2015. Pedestrian Guidance for Public Transport Users in Indoor Stations Using Smartphones. IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC 2015–Octob, 2539–2544. https://doi.org/10.1109/ITSC.2015.403

Díaz-Vilariño, L., Conde, B., Lagüela, S., Lorenzo, H., 2015. Automatic detection and segmentation of columns in as-built buildings from point clouds. Remote Sens. 7, 15651–15667. https://doi.org/10.3390/rs71115651

Eckel, W., 2015. Creating an Artificial World with a New Kind of Cellular Automata. ArXiv e-prints.

Fichtner, F.W., Diakité, A.A., Zlatanova, S., Voûte, R., 2018. Semantic enrichment of octree structured point clouds for multi-story 3D pathfinding. Trans. GIS 22, 233–248.

https://doi.org/10.1111/tgis.12308

Gang, L., Guangshun, S., 2010. Procedural Modeling of Urban Road Network, in: 2010 International Forum on Information Technology and Applications. pp. 75–79. https://doi.org/10.1109/IFITA.2010.122

Gerke, M., Butenuth, M., Heipke, C., Willrich, F., 2004. Graph-supported verification of road databases. ISPRS J. Photogramm. Remote Sens. 58, 152–165. https://doi.org/https://doi.org/10.1016/j.isprsjprs.2003.09.003

Golub, G.H., Heath, M., Wahba, G., 1979. Generalized Cross-Validation as a Method for Choosing a Good Ridge Parameter. Technometrics 21, 215–223. https://doi.org/10.1080/00401706.1979.10489751

Huang, J., You, S., 2015. Pole-like object detection and classification from urban point clouds. 2015 IEEE Int. Conf. Robot. Autom. 3032–3038. https://doi.org/10.1109/ICRA.2015.7139615

ISO, 2011. ISO-21542 Building construction — Accessibility and usability of the built environment. ISO Int. Organ. Stand.

Izzati, N., Shamsudin, H., abdul razak, nurdiyana, Yusof, U., Khalid, mohd nor akmal, 2015. A Cellular Automata and Navigation Graph Framework in Simulating Crowd Behaviour at Bus Terminal Platform.

Jackway, P.T., Deriche, M., 1996. Scale-space properties of the multiscale morphological dilation-erosion. IEEE Trans. Pattern Anal. Mach. Intell. 18, 38–51. https://doi.org/10.1109/34.476009

Jamali, A., Abdul Rahman, A., Boguslawski, P., Kumar, P., Gold, C.M., 2017. An automated 3D modeling of topological indoor navigation network. GeoJournal 82, 157–170. https://doi.org/10.1007/s10708-015-9675-x

Kang, H. Il, Lee, B., Kim, K., 2008. Path Planning Algorithm Using the Particle Swarm Optimization and the Improved Dijkstra Algorithm. 2008 IEEE Pacific-Asia Work. Comput. Intell. Ind. Appl. 1002–1004. https://doi.org/10.1109/PACIIA.2008.376

Kohavi, R., 1995. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection, in: IJCAI.

Kumar, P., Lewis, P., McCarthy, T., 2017. The Potential of Active Contour Models in Extracting Road Edges from Mobile Laser Scanning Data. Infrastructures . https://doi.org/10.3390/infrastructures2030009

Lamarche, F., Donikian, S., 2004. Crowd of virtual humans: A new approach for real time navigation in complex and structured environments. Comput. Graph. Forum 23, 509–518. https://doi.org/10.1111/j.1467-8659.2004.00782.x

Liu, L., Zlatanova, S., 2011. A “Door-To-Door” Path-Finding Approach For Indoor Navigation. Proc. Gi4DM 2011 Geoinf. Disaster Manag. Antalya, Turkey, 3-8 May 2011 3–8.

López-Pazos, G., Balado, J., Díaz-Vilariño, L., Arias, P., Scaioni, M., 2017. Pedestrian pathfinding in urban environments: Preliminary results, in: ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Kyiv, Ukraine, pp. 4–6. https://doi.org/doi.org/10.5194/isprs-annals-IV-5-W1-35-2017

Lorenz, B., Ohlbach, H.J., Stoffel, E.-P., 2006. A Hybrid Spatial Model for Representing Indoor Environments, in: Proceedings of the 6th International Conference on Web and Wireless Geographical Information Systems, W2GIS’06. Springer-Verlag, Berlin, Heidelberg, pp. 102–112. https://doi.org/10.1007/11935148_10

Maruyama, T., Kanai, S., Date, H., Tada, M., 2017. Simulation-Based Evaluation of Ease of Wayfinding Using Digital Human and As-Is Environment Models. ISPRS Int. J. Geo-Information . https://doi.org/10.3390/ijgi6090267

Maruyama, T., Kanai, S., Date, H., Tada, M., 2016. Motion-capture-based walking simulation of digital human adapted to laser-scanned 3D as-is environments for accessibility evaluation. J. Comput. Des. Eng. 3, 250–265. https://doi.org/https://doi.org/10.1016/j.jcde.2016.03.001

Mountrakis, G., Im, J., Ogole, C., 2011. Support vector machines in remote sensing: A review. ISPRS J. Photogramm. Remote Sens. 66, 247–259. https://doi.org/https://doi.org/10.1016/j.isprsjprs.2010.11.001

Nasir, M., Lim, C.P., Nahavandi, S., Creighton, D., 2014. Prediction of pedestrians routes within a built environment in normal conditions. Expert Syst. Appl. 41, 4975–4988. https://doi.org/10.1016/j.eswa.2014.02.034

Ngoc Nha, V.T., Djahel, S., Murphy, J., 2012. A Comparative Study of Vehicles' Routing Algorithms for Route Planning in Smart Cities. Veh. Traffic Manag. Smart Cities (VTM), First Int. Work. IEEE. 1–6. https://doi.org/10.1109/VTM.2012.6398701

Pettré, J., Laumond, J.-P., Thalmann, D., 2005. A navigation graph for real-time crowd animation on multilayered and uneven terrain.

Pomerleau, F., Colas, F., Siegwart, R., Magnenat, S., 2013. Comparing ICP variants on real-world data sets. Auton. Robots 34, 133–148. https://doi.org/10.1007/s10514-013-9327-2

Preacher, K.J., Curran, P.J., Bauer, D.J., 2006. Computational Tools for Probing Interactions in Multiple Linear Regression, Multilevel Modeling, and Latent Curve Analysis. J. Educ. Behav. Stat. 31, 437–448. https://doi.org/10.3102/10769986031004437

Puente, I., González-Jorge, H., Martínez-Sánchez, J., Arias, P., 2013. Review of mobile mapping and surveying technologies. Meas. J. Int. Meas. Confed. 46, 2127–2145. https://doi.org/10.1016/j.measurement.2013.03.006

Riveiro, B., González-Jorge, H., Martínez-Sánchez, J., Díaz-Vilariño, L., Arias, P., 2015. Automatic detection of zebra crossings from mobile LiDAR data. Opt. Laser Technol. 70, 63–70. https://doi.org/10.1016/j.optlastec.2015.01.011

Roynard, X., Deschaud, J.-E., Goulette, F., 2016. Fast and Robust Segmentation and Classification for Change Detection in Urban Point Clouds. ISPRS - Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. XLI-B3, 693–699. https://doi.org/10.5194/isprsarchives-XLI-B3-693-2016

Schauer, J., Nüchter, A., 2018. Removing non-static objects from 3D laser scan data. ISPRS J. Photogramm. Remote Sens. https://doi.org/https://doi.org/10.1016/j.isprsjprs.2018.05.019

Schauer, J., Nüchter, A., 2018. The Peopleremover—Removing Dynamic Objects From 3-D Point Cloud Data by Traversing a Voxel Occupancy Grid. IEEE Robot. Autom. Lett. 3, 1679–1686. https://doi.org/10.1109/LRA.2018.2801797

Serna, A., Marcotegui, B., 2014. Detection , segmentation and classification of 3D urban objects using mathematical morphology and supervised learning. ISPRS J. Photogramm. Remote Sens. 93, 243–255. https://doi.org/10.1016/j.isprsjprs.2014.03.015

Serna, A., Marcotegui, B., 2013. Urban accessibility diagnosis from mobile laser scanning data. ISPRS J. Photogramm. Remote Sens. 84, 23–32. https://doi.org/10.1016/j.isprsjprs.2013.07.001

Singhal, P., Kundra, H., 2014. A Review paper of Navigation and Pathfinding using Mobile Cellular Automata.

Soilán, M., Riveiro, B., Sánchez-Rodríguez, A., Arias, P., 2018. Safety assessment on pedestrian crossing environments using MLS data. Accid. Anal. Prev. 111, 328–337. https://doi.org/https://doi.org/10.1016/j.aap.2017.12.009

Soltani, A.R., Tawfik, H., Goulermas, J.Y., Fernando, T., 2002. Path planning in construction sites: Performance evaluation of the dijkstra, a*, and GA search algorithms. Adv. Eng. Informatics 16, 291–303. https://doi.org/10.1016/S1474-0346(03)00018-1

Staats, B.R., Diakité, A.A., Voûte, R.L., Zlatanova, S., 2017. Automatic generation of indoor navigable space using a point cloud and its scanner trajectory. ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. 4, 393–400. https://doi.org/10.5194/isprs-annals-IV-2-W4-393-2017

Trevor, A.J.B., Gedikli, S., Rusu, R.B., Christensen, H.I., 2013. Efficient Organized Point Cloud Segmentation with Connected Components. Semant. Percept. Mapp. Explor.

Vallet, B., Brédif, M., Serna, A., Marcotegui, B., Paparoditis, N., 2015. TerraMobilita/iQmulus urban point cloud analysis benchmark. Comput. Graph. 49, 126–133. https://doi.org/10.1016/j.cag.2015.03.004

Xiao, W., Vallet, B., Brédif, M., Paparoditis, N., 2015. Street environment change detection from mobile laser scanning point clouds. ISPRS J. Photogramm. Remote Sens. 107, 38–49. https://doi.org/https://doi.org/10.1016/j.isprsjprs.2015.04.011

Xiao, W., Vallet, B., Schindler, K., Paparoditis, N., 2016. Street-side vehicle detection, classification and change detection using mobile laser scanning data. ISPRS J. Photogramm. Remote Sens. 114, 166–178. https://doi.org/https://doi.org/10.1016/j.isprsjprs.2016.02.007

Y.F. Tang, Ma., S.C. Pun-Cheng, L., 2004. Algorithmic development of an optimal path computation model based on topographic map features. Isprs 2004.

Yang, B., Dong, Z., Liu, Y., Liang, F., Wang, Y., 2017. Computing multiple aggregation levels and contextual features for road facilities recognition using mobile laser scanning data. ISPRS J. Photogramm. Remote Sens. 126, 180–194. https://doi.org/10.1016/j.isprsjprs.2017.02.014

Yang, B., Dong, Z., Zhao, G., Dai, W., 2015. Hierarchical extraction of urban objects from mobile laser scanning data. ISPRS J. Photogramm. Remote Sens. 99, 45–57. https://doi.org/https://doi.org/10.1016/j.isprsjprs.2014.10.005