PATH PLANNING FOR INDOOR CONTACT INSPECTION TASKS WITH UAVS

L.M. González-deSantos[1,*], J. Martínez-Sánchez[1], H. González-Jorge[2], P. Arias[1]

[1]Geotech Group, Mining and Energy Engineering School, University of Vigo, Vigo, Spain -(luismgonzalez, joaquin.martinez,parias)@uvigo.es
[2]Geotech Group, Aerospace Engineering School, University of Vigo, Ourense, Spain-higiniog@uvigo.es

Commission IV, WG IV/5

KEY WORDS: Unmanned Aerial System, Industrial Inspection, Path Planning, navigation, Flight control, Non-destructive testing

ABSTRACT:

UAV technology has become a useful tool for the inspection of infrastructures. Structural Health Monitoring methods are already implementing these vehicles to obtain information about the condition of the structure. Several systems based on close range remote sensing and contact sensors have been developed. In both cases, in order to perform autonomous missions in hard accessible areas or with obstacles, a path planning algorithm that calculates the trajectory to be followed by the UAV to navigate these areas is mandatory. This works presents a UAV path planning algorithm developed to navigate indoors and outdoors. This algorithm does not only calculate the waypoints of the path, but the orientation of the vehicle for each location. This algorithm will support a specific UAV-based contact inspection of vertical structures. The required input data consist of a point cloud of the environment, the initial position of the UAV and the target point of the structure where the contact inspection will be performed.

1. INTRODUCTION

UAVs systems have become a useful tool for SHM (Structural Health Monitoring) tasks, since they can navigate in complex and hard-accessible areas easily. This property makes this technology perfect for inspection tasks in big infrastructures (Duque et al., 2018). Several UAV systems for infrastructure inspection have been developed, using close remote sensing payloads, such as cameras (Seo et al., 2018) or LiDAR (Light Detection and Ranging) sensors (Teng et al., 2017), or contact inspection sensors, such as ultrasonic sensors (Zhang et al., 2018). In both cases, it is needed a path planning algorithm to perform this inspection tasks autonomously.

For outdoor missions, the path planning is usually founded on a GNSS way-point based algorithm using open software program (Ardupilot, n.d.; Dronecode, n.d.). In these missions, the system is less dependent on the environment where the mission is to be performed. For more complex environments, where the system is expected to fly in reduced or confined spaces, a more sophisticated path-planning method must be used.

Several indoor navigation algorithms have been also developed, in which obstacle detection is fundamental, using 3D point clouds of the environment using LiDAR sensors (Díaz-Vilariño et al., 2016). In the indoor case, it is fundamental to found the doors in the room, that are the component that connects different rooms (Liu and Zlatanova, 2011)

2. RELATED WORK

There are many two-dimensional (2D) path planning methods (Liu and Zlatanova, 2013), where the data used for the path planning is the existing building information. This methods can be classified in two large groups: surface-based and volume-based models (Fernández-Prieto et al., 2019; Li et al., 2018; Tomic et al., 2012). One of the main restrictions of the current developed 3-D path finding methods for UAVs is that they focused only on obtaining path positions. The vehicle orientation (Roll, Pitch, Yaw angles) along such path is also a key aspect to perform autonomous missions.

Volume-based algorithms are based on the discretization of the volumetric information, frequently by means of a voxelization: a discretization method, that is based on dividing the volume in smaller cubes of a fixed size, voxels. Each voxel is classified depending on the area that they represent (González-de Santos et al., 2018). Also, a more efficient octree structure can be applied to volume discretization, obtaining an equivalent resolution with less data to be processed (González-deSantos et al., 2018).

The aim of this work is to develop a path planning algorithm for UAV combining different developed methods and adding an orientation planning for the calculated path. The developed methodology is applied to contact inspection tasks indoors and outdoors. Authors have already developed a payload for UAVs to perform semi-autonomous contact inspection tasks in vertical structures (González-deSantos et al., 2020, 2019) , which control the approaching and stabilized contact of the UAV to the structure to be inspected. The main objective of the developed algorithm is to obtain and plan the path to navigate from the initial point, where the UAV is landed, to an oriented point in front of the structure where the payload control is capable of managing the approaching movements to perform the contact inspection.

3. METHODOLOGY

The data used by the developed path planning algorithm is a 3D point cloud obtained by TLS (Terrestrial Laser Scanner). Figure 1 presents the workflow of the developed methodology.

[*] Corresponding author

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLIII-B4-2020-345-2020 | © Authors 2020. CC BY 4.0 License.

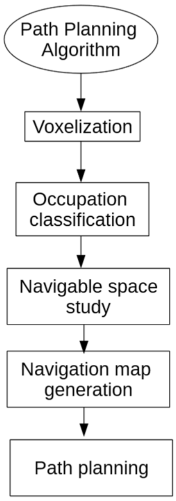


Figure 1. General workflow of the method

3.1 Voxelization.

This is the first step of the method, that consist of the discretization of the point cloud of the environment into a voxel-based structure, where each voxel is classified depending on the properties of the space it represents. In this step, the voxel size has to be defined. Choose a correct voxel size is important, because it implies a trade-off between the loss of detail in the point cloud and the computational cost of the classification.

A pre-processing of the point cloud was performed before the data discretization. In this process, the point cloud is rotated in order to make the floor plane completely horizontal. To calculate the floor's plane, a RANSAC (Random Sample Consensus) algorithm is used (Fischler and Bolles, 1981). Some LS (Laser Scanner) use a barometer to collect absolute elevations to reference the Z coordinate of each point. A rigid body transformation is applied in order to reference the height of each point to the floor, making its plane coincident with the XY plane. Then, the point cloud is segmented in 3 parts, separating the floor and the ceiling from the rest of the room (Figure 2).

In most of the buildings, walls are perpendicular and parallel to each other. According to this, the point cloud is rotated around the Z axis in order to align walls with the voxel faces, since they are aligned with XYZ coordinates. In this way the voxelization can be optimized. To make this, a small horizontal section of the upper part of the room is segmented, that is usually the less occluded part in most of the non-industrial buildings. In this section, the plane of each wall is fitted, locating the longest wall of the room. The normal vector of this wall is used to rotate the entire point cloud of the room to align it with the X axis. When the point cloud is rotated, the voxelization structure can be created, taking into account the room limits in each axis and the voxel size, calculating the voxel matrix's dimensions.

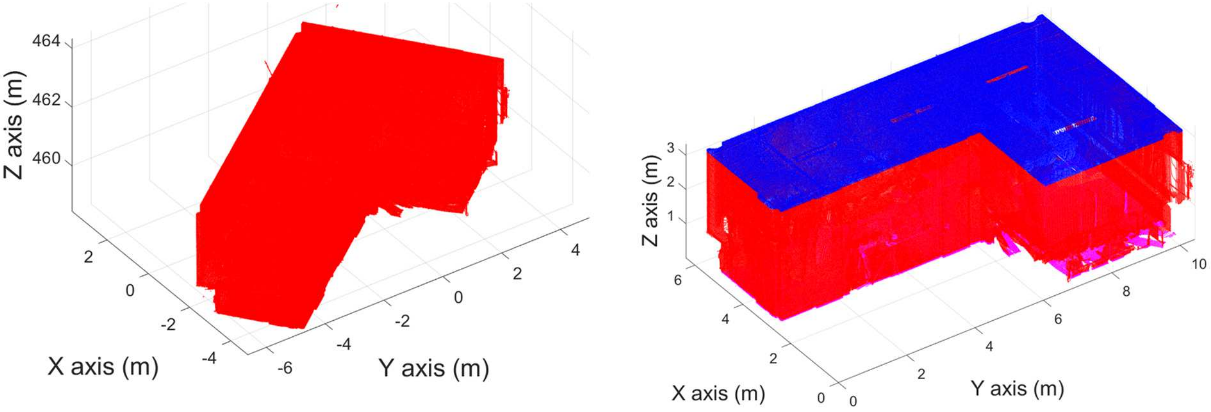
Each voxel of the structure is classified depending on its occupancy as one of the following four types of voxels:

- Empty: Voxels that do not contain points
- Occupied: Voxels that contain points.
- Security offset: Empty voxel that is near to one or more occupied voxels. They are defined by a security distance.
- Exterior: Empty voxels that are out of the study room.

Initially, all the voxels are defined as empty voxels. The following sections explain how these voxels are reclassified.

3.2 Occupation classification

In this step, each voxel is studied individually, classifying them depending on their inner point density. If a voxel has a point density higher than a predefined threshold, the voxel is classified as occupied. The value of this density threshold is of great importance, because it helps to discriminate noisy voxels or reflections in the point cloud that otherwise could affect the classification.



a) **b)**

Figure 2. a) Point cloud before pre-processing. b) Point cloud segmented in floor, ceiling and room. (magenta, blue and red respectively).

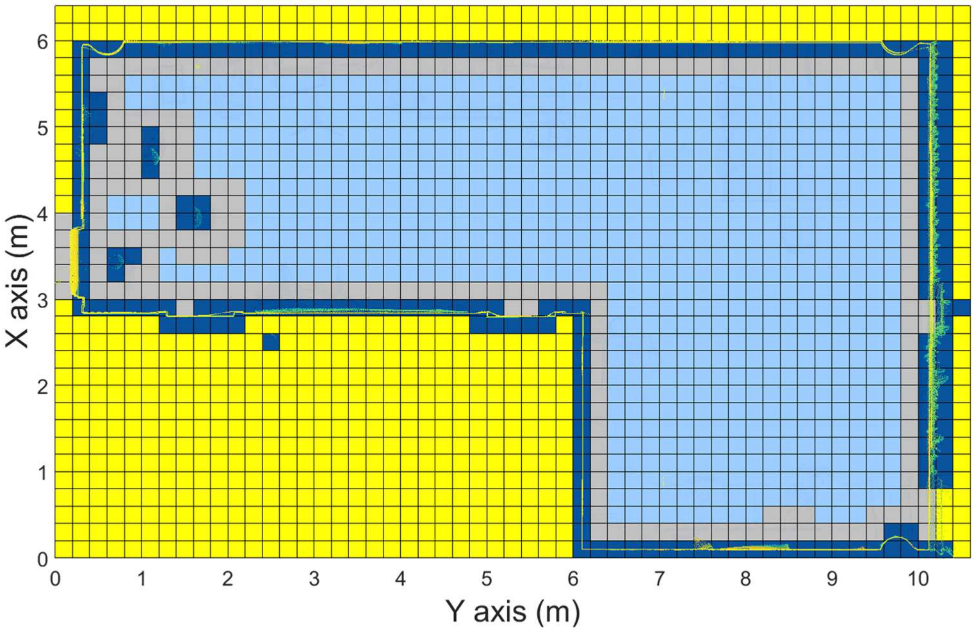


Figure 3. Horizontal plane of voxels. Dark blue: occupied voxels; Grey: security offset voxels; Yellow: exterior voxels; Light blue: empty voxels or navigable area.

3.3 Navigability study

After the first classification of the occupied/empty voxels, the point cloud is evaluated to localize and delimit the room, classifying the outer voxels as exterior voxels. In order to carry out this classification, the point cloud of the ceiling is used, applying a 2D discretization based on a uniform grid in this case. The cell-size of the grid is equal to the voxel-size and, accordingly, the cells of the grid are coincident with the voxel top faces. An occupation classification, similar to the classification followed in the previous section, is derived but, in this case, the squares with a point density lower than the threshold are classified as exterior, and this classification is applied to all empty voxels that are below the square.

Also, a security distance depending on the UAV size is defined. With this distance, a security offset is generated around the occupied voxels in an operation similar to a morphological dilation but with structuring element (s=NxNxN) of the operation is a 3D matrix. The matrix dimensions (N) is calculated using the security distance selected and the voxel size (Equation 1).

$$N = \left(ceil\left(\frac{security\ distance}{voxel\ size}\right) * 2\right) + 1 \quad (1)$$

Only empty voxels can change their classification during this process, occupied and exterior voxels do not change. In this way, the navigable area, that is defined as the area where the UAV could be positioned, is calculated (Figure 3).

3.4 Navigation map generation

Once the navigable area has been defined, the navigation map can be generated, using just the target position that the UAV has to reach, but not the initial position. Basically, this navigation map (*Map*) is a 3D matrix with the same dimensions as the voxel discretization where each voxel is given a value according to the following rules:

- Non-navigable voxels or navigable voxels not directly connected with the target position, meaning that there is not possible path between them and the target position, are given a value of -1.
- Navigable voxels directly connected to the target position are given a value depending on the voxels that surround it and the distance.
- The target voxel, that is, the voxel that contains the target position is given a value of 1.

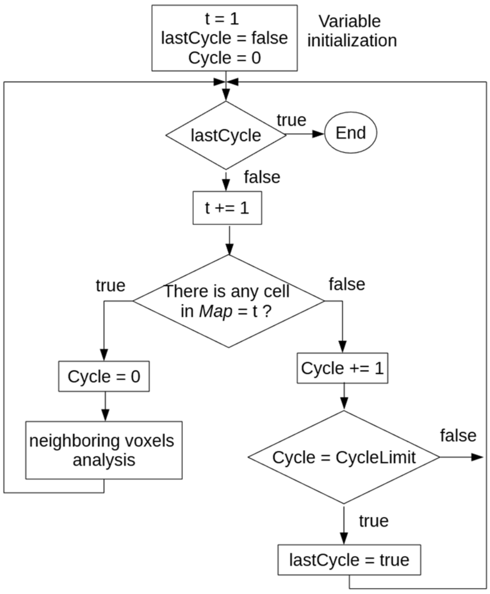


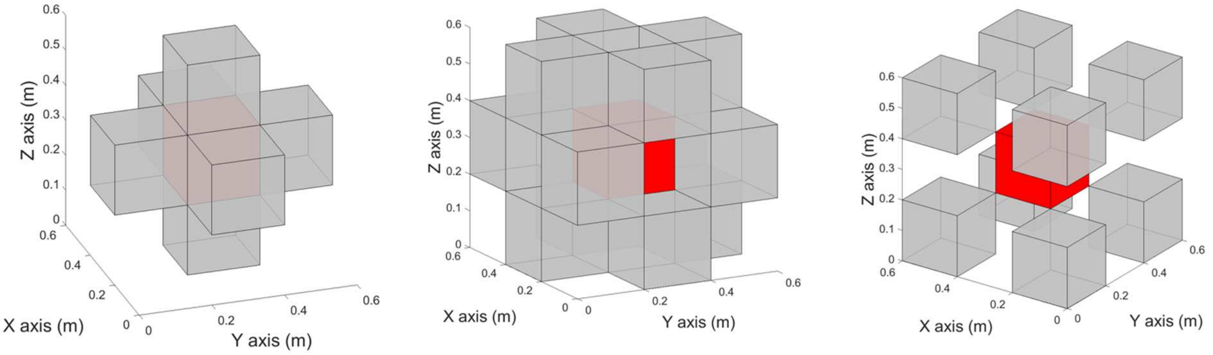
Figure 4. Navigation map algorithm workflow

The workflow of the map generation is described in Figure 4. As can be seen, we start the generation of the map in the target position, that has value 1. Then, we start expanding the navigation map, analysing the cells that surround this voxel, giving them a value depending on their position. Depending on the relation between the indexed of the voxels in the volume, three classification groups can be distinguished (Figure 5):

- *Common face*: voxels surrounding the study voxel that have a common face. This means that 2 of the indexed coordinates of this voxels and the study one are coincident
- *Common edge*: voxels surrounding the study voxel that have a common edge. This means that 1 of the indexed coordinates of this voxel and the study one are coincident.
- *Common vertex*: voxels surrounding the study voxels that have a common vertex. This means that none of the coordinates are coincident.

For each of the 26 voxels that surround the study voxel, a new weight value is given. Each relationship type has a standard weight. In this study, a value of 1 is assigned to *common face voxels,* the *common edge voxels* obtain a weight value of 2 and a weight value of 3 is assigned to *common vertex voxels.* The weight value has to coincide with the stop criteria *CycleLimit*. This weight is added to the value of the voxel under study and is assigned to empty voxels only if it has not been previously assigned a lower value. In this way each voxel of the map has the lowest possible value.

As can be seen in Figure 6, the Navigation map generated is similar to a gradient that decreases near to the target position, that has the lowest value of the map.



a) b) c)

Figure 5. Types of position relationship between voxels (red: study voxel; grey: voxels surrounding the study voxel). a) Common face. b) Common edge. c) Common vertex

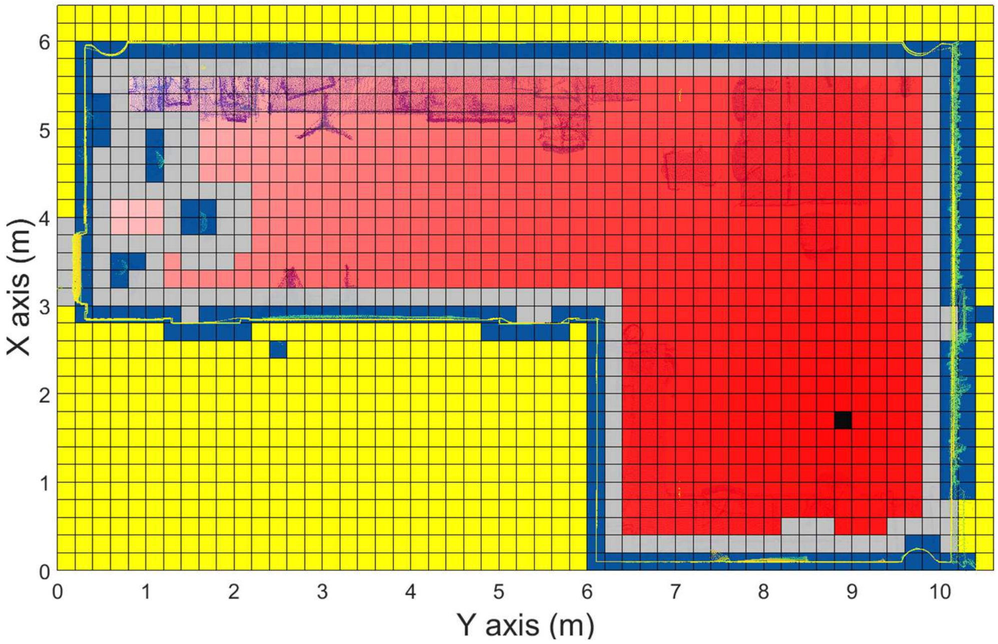


Figure 6. Horizontal plane of voxels that shows the Navigation Map generated (red voxels) around the objective position (black voxel)

3.5 Path planning

This section describes how the path is calculated, using the generated map. A voxel has to fulfil only one requirement to be assigned as initial position, which is having a value different than -1 in the navigation map. As a result, an initial voxel fulfilling this requirement means also that it is an empty voxel directly connected to the objective position.

In order to calculate the path, a process similar to the neighbouring voxels analysis explained before is carried out. In this case, we start with the initial position and analyse the values of the surrounding voxels in the navigation map, looking for the lowest value voxel, which is selected as the next position in the path, and turns the study voxel. This process is repeated until target voxel is reached.

As it has been said before, this path planning algorithm was developed specifically for UAVs to navigate and perform a completely autonomous contact inspection task with the system developed in previous works (González-de Santos et al., 2019; González-deSantos et al., 2020, 2019) (Figure 8). The general operation consists of a UAV landed on the floor and has to take off at the beginning of the mission. Taking into consideration the requirements of the contact payload, the system has navigate to a location and orientation one meter in front of the target point. With this information, the path planning software must calculate the path to be followed by the UAV to perform the contact inspection from an initial position (Figure 9).

When the path has been calculated, the orientation, consisting of three angles: roll, pitch and yaw, has to be obtained for each position in the path. In this work, only the yaw angle is calculated and fixed because roll and pitch are used for the UAV navigation and, thus, left to be calculated by the flight controller. The Yaw angle calculation supports the UAV rotation to fit the target orientation, perpendicular to the wall where the contact will be performed. In this work, three methods have been considered for the Yaw calculation:

- Initial/final rotation: The UAV runs the full rotation in the initial (or final) position, maintaining the orientation in the waypoints of the path.
- Progressive rotation: The UAV rotates progressively in Yaw between the initial orientation and the target orientation.
- Tangent to the path: The changes in UAV orientation are proportional to the azimuth changes in the path, i.e. the orientation of the path projection onto the XY plane.

The user has to choose between these options. For this work, the progressive rotation has been selected (Figure 7), since is the one that generates the most fluent path.

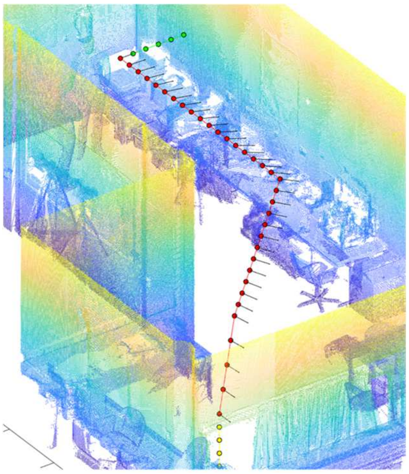


Figure 7. Progressive rotation (black arrows shows the

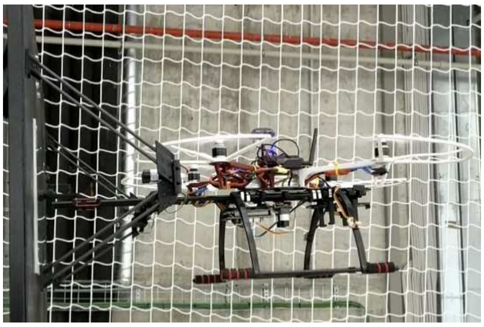
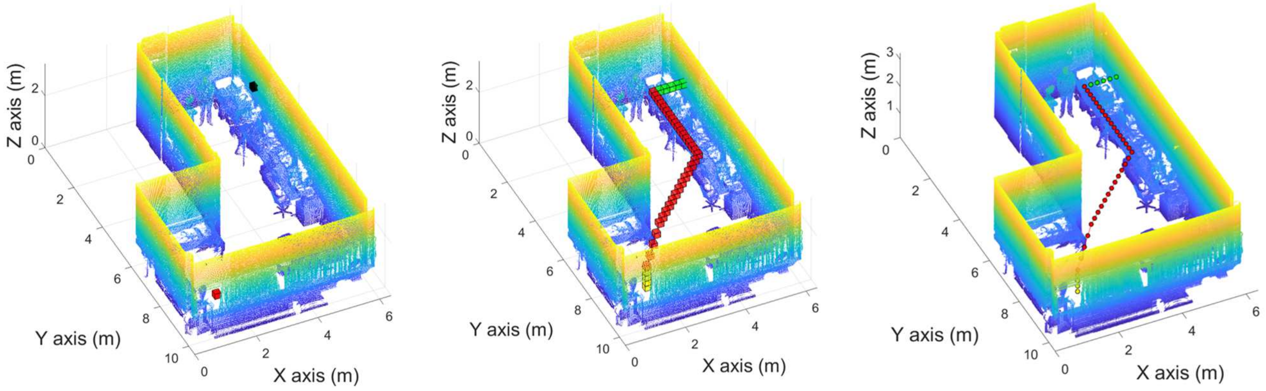


Figure 8. Contact inspection system developed in previous works.



a) b) c)

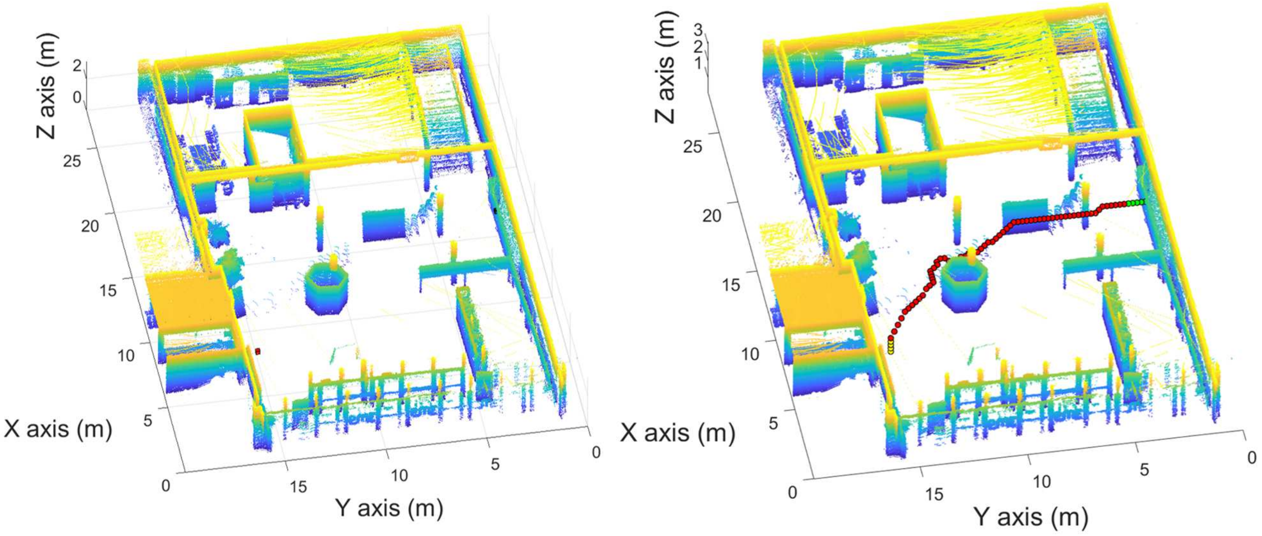
Figure 9. Path calculation. a) Initial position (red voxel) and objective position (black voxel). b) Voxels selected by the path planning algorithm (yellow: take off; red: path; green: contact approaching movements). c) Points of the calculated path.

4. RESULTS AND DISCUSSION

The path planning algorithm developed in this study has been tested in two study cases. The first one has been carried out in a laboratory of the Mining and Energy School at the University of Vigo, which size is approximately 10m x 6m. In this first test, a Faro Focus 3D x330 LiDAR scanner was used to obtain the point cloud, that contains 0.66 million points. The results of this test were shown in the Methodology section. The second study case has been carried out in the Hall of the same School, the point cloud was obtained using a prototype of a backpack-based mobile system (Filgueira et al., 2016). This point cloud is available in the ISPRS Benchmark on Indoor Modelling (Khoshelham et al., 2017). The size of this second room is approximately 14m x 26m. The point cloud used contains 5.5 million points. In both cases, a voxels size of 0.2m has been selected, that is enough to capture critical obstacles like chairs and other furniture.

All the algorithms for the path planning processing were implemented in MATLAB 2018b, using the Point Cloud Library. The PC used for this study has an Intel core i7 and 16GB of RAM.

Figure 10. shows the results obtained in the second study case, where a longer path has been calculated. Figure 11. shows the navigation map generated for this room. As can be seen in the figures, this second point cloud is noisy, but the software is able to process this data to calculate the path.



a) b)

Figure 10. Path calculation. a) Initial position (red) and objective position (black voxel). b) Path calculated (yellow: take off; red: path; green: contact approaching movements)

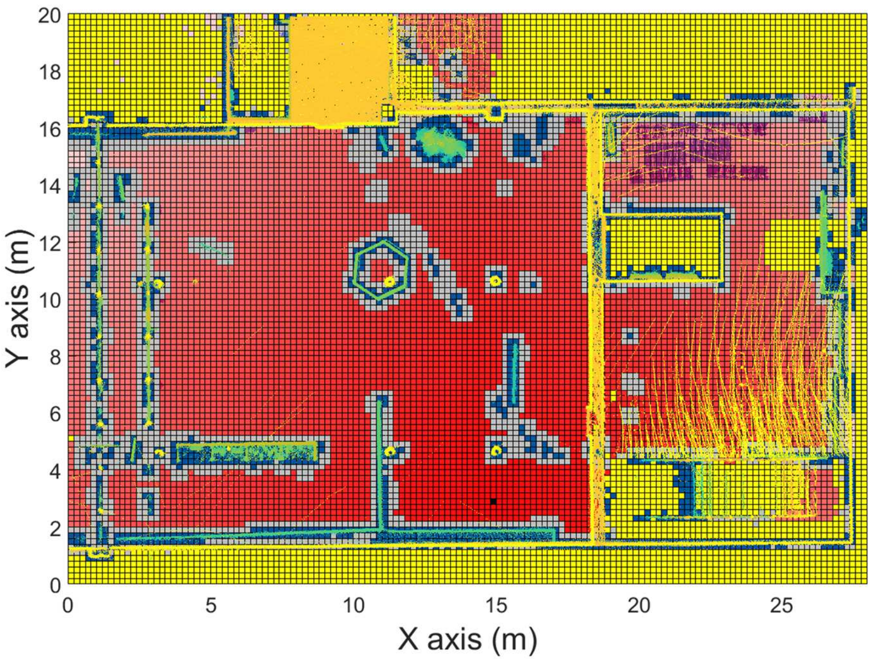


Figure 11. Navigation Map generated (red voxels) around the objective position (black voxel)

5. CONCLUSSIONS

This paper presents a path planning algorithm developed for the navigation of a UAV to carry out contact inspection tasks completely autonomously. The algorithm has been tested in two study cases, with a point cloud obtained with a TLS and another obtained with a backpack-based mobile scanner, that contains more noise. In both cases good results have been obtained. This work completes previous works where a payload for UAV was developed to perform contact inspection tasks.

As future work, the developed algorithm will be optimized, to being applied for real study cases. As part of this future work, a software communicating the results of the path planning to the flight controller will be developed.

ACKNOWLEDGEMENTS

This research is funded by the Universidade de Vigo (grant ref. 00Vl 131H 641.02) and the Ministerio de Economia, Industria y Competitividad, Gobierno de España (TIN2016-77158-C4-2-R, RTC-2016-5257-7). This project is co-financed by the INTERREG Atlantic Area Programme through the European Regional Development Fund (ERDF) with application code: EAPA_826/2018.
The sole responsibility for the content of this document lies with the authors. It does not necessarily reflect the opinion of the European Union. Neither the INTERREG Europe programme authorities are responsible for any use that may be made of the information contained therein.

REFERENCES

Ardupilot, n.d. Mission Planner [WWW Document]. URL https://ardupilot.org/planner/ (accessed 4.30.20).

Díaz-Vilariño, L., Boguslawski, P., Khoshelham, K., Lorenzo, H., Mahdjoubi, L., 2016. Indoor navigation from point clouds: 3D modelling and Obstacle Detection, in: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives. https://doi.org/10.5194/isprsarchives-XLI-B4-275-2016

Dronecode, n.d. QGroundControl [WWW Document]. URL http://qgroundcontrol.com/ (accessed 4.30.20).
Duque, L., Seo, J., Wacker, J., 2018. Synthesis of Unmanned Aerial Vehicle Applications for Infrastructures. J. Perform. Constr. Facil. 32, 04018046. https://doi.org/10.1061/(ASCE)CF.1943-5509.0001185

Fernández-Prieto, J., Díaz-Vilariño, L., Arias, P., 2019. Automatic processing of point clouds for the path planning of UAVs in indoor environmets. University of Vigo.

Filgueira, A., Arias, P., Bueno, M., Lagüela, S., 2016. Novel inspection system, backpack-based, for 3D modelling of indoor scenes, in: International Conference on Indoor Positioning and Indoor Navigation (IPIN), 4-7 October 2016, Alcalá de Henares, Spain.

Fischler, M.A., Bolles, R.C., 1981. Random sample consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Commun. ACM. https://doi.org/10.1145/358669.358692

González-de Santos, L.M., Díaz-Vilariño, L., Balado, J., Martínez-Sánchez, J., González-Jorge, H., Sánchez-Rodríguez, A., 2018. Autonomous point cloud acquisition of unknown indoor scenes. ISPRS Int. J. Geo-Information. https://doi.org/10.3390/ijgi7070250

González-de Santos, L.M., Martínez-Sánchez, J., González-Jorge, H., Novo, A., Arias, P., 2019. FIRST APPROACH TO UAV-BASED CONTACT INSPECTION: A SMART PAYLOAD FOR NAVIGATION IN THE NEIGHBOURHOOD OF STRUCTURES. ISPRS - Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. XLII-2/W13, 323–328. https://doi.org/10.5194/isprs-archives-XLII-2-W13-323-2019

González-deSantos, L.M., Martínez-Sánchez, J., González-Jorge, H., Díaz-Vilariño, L., Riveiro, B., 2018. New discretization method applied to NBV problem: Semioctree. PLoS One 13, e0206259. https://doi.org/10.1371/journal.pone.0206259

González-deSantos, L.M., Martínez-Sánchez, J., González-Jorge, H., Navarro-Medina, F., Arias, P., 2020. UAV payload with collision mitigation for contact inspection. Autom. Constr. https://doi.org/10.1016/j.autcon.2020.103200

González-deSantos, L.M., Martínez-Sánchez, J., González-Jorge, H., Ribeiro, M., de Sousa, J.B., Arias, P., 2019. Payload for Contact Inspection Tasks with UAV Systems. Sensors 19, 3752. https://doi.org/10.3390/s19173752

Khoshelham, K., Vilariño, L.D., Peter, M., Kang, Z., Acharya, D., 2017. The ISPRS benchmark on indoor modelling, in: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives. https://doi.org/10.5194/isprs-archives-XLII-2-W7-367-2017

Li, F., Zlatanova, S., Koopman, M., Bai, X., Diakité, A., 2018. Universal path planning for an indoor drone. Autom. Constr. https://doi.org/10.1016/j.autcon.2018.07.025

Liu, L., Zlatanova, S., 2013. A two-level path-finding strategy for indoor navigation, in: Lecture Notes in Geoinformation and Cartography. https://doi.org/10.1007/978-3-642-33218-0_3

Liu, L., Zlatanova, S., 2011. A “door-to-door” path-finding approach for indoor navigation, in: Gi4DM 2011 - GeoInformation for Disaster Management.

Seo, J., Duque, L., Wacker, J.P., 2018. Field Application of UAS-Based Bridge Inspection. Transp. Res. Rec. J. Transp. Res. Board 2672, 72–81. https://doi.org/10.1177/0361198118780825

Teng, G.E., Zhou, M., Li, C.R., Wu, H.H., Li, W., Meng, F.R., Zhou, C.C., Ma, L., 2017. MINI-UAV LIDAR FOR POWER LINE INSPECTION. ISPRS - Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. XLII-2/W7, 297–300. https://doi.org/10.5194/isprs-archives-XLII-2-W7-297-2017

Tomic, T., Schmid, K., Lutz, P., Domel, A., Kassecker, M., Mair, E., Grixa, I., Ruess, F., Suppa, M., Burschka, D., 2012. Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue. IEEE Robot. Autom. Mag. https://doi.org/10.1109/MRA.2012.2206473

Zhang, D., Watson, R., Dobie, G., MacLeod, C., Pierce, G., 2018. Autonomous Ultrasonic Inspection Using Unmanned Aerial Vehicle, in: 2018 IEEE International Ultrasonics Symposium (IUS). IEEE, pp. 1–4. https://doi.org/10.1109/ULTSYM.2018.8579727