Non-Model-Based approach for complete digitization by TLS or mobile scanner

Farouk Achakir, Sanaa El Fkihi, El Mustapha Mouaddib

Non-Model-Based approach for complete digitization by TLS or mobile scanner

Farouk Achakir[a,*], Sanaa El Fkihi[b], El Mustapha Mouaddib[a]

[a]MIS Laboratory, University of Picardie Jules Verne (UPJV), 14 quai de la Somme, Amiens, 80000, France
[b]IRDA Group, ADMIR Laboratory, Rabat IT Center, ENSIAS, Mohammed V University in Rabat, Rabat, Morocco

Abstract

This paper investigates automatic digitization with complete coverage of large and complex environments using a TLS or a mobile scanner. We propose an adaptive multi-objective view-planner that can operate in an unknown environment to provide guidance for the human operator and ease the scanning task or by a mobile robot for an automatic exploration of the environment. The proposed view-planner is adapted to environments where the sensor is operating on a flat surface such as office spaces, urban areas, open fields or in some cultural heritage applications. First, we propose an adaptive gap-based method to extract occluded areas in a point cloud, which is completely automated and does not require extensive computations in a large environment such as ray-tracing or level-set methods. Then, we introduce a novel exploration strategy that uses specific regions of the environment called ”Conservative-Cells” to drastically reduce the number of sensing positions to achieve complete digitization of the environment.

Both methods were validated with simulated and real point clouds. The proposed approach has been applied to a scanner carried by a mobile robot, then to data acquired by a TLS used by a human operator in a large, complex environment. Experimental results on both TLS and mobile robot show that our view-planning approach is effective in finding a sequence of positions that leads to a complete reconstruction of the environment. Moreover, the proposed approach shows efficient performance in terms of coverage rate and computational time compared to others view-planning approaches as well as the results of an experienced human operator in a large, complex environment.

Keywords: TLS, Mobile scanner, Visibility analysis, View Planning, Next-Best-View, Non-Model-Based

1. INTRODUCTION

In recent decades, environment digitization with a laser scanner has become popular due to the high demand for virtual models in various fields of application. A laser scanner can be embedded on a mobile robot or positioned by a human in the case of a Terrestrial Laser Scanner (TLS) to remotely digitize objects and provides fast measurements over a wide angular range.
In environments with complex object geometry and unavoidable obstacles, multiple scans from different locations are required to achieve maximum coverage. The registration process is used to ”link” the scans together by using artificial targets references, among other methods [1, 2, 3].

The key factor in acquiring a point cloud is to estimate the best sequence of the scanner's position, allowing the scanned area to environment to be expanded rapidly, this increasing the coverage / completeness. Nevertheless, in large and complex environments, even an experienced human operator cannot accurately identify the remaining occluded areas to achieve the best coverage rate with the minimum number of the scanner positions.

Therefore, a computer-aid algorithm is needed to assist the task of environment digitization. Usually, this class of algorithm is called ”View-Planner”. It intends to achieve visibility analysis by extracting the locations of occluded areas and estimating the best sequence of sensing positions based on an exploration strategy to ensure global coverage of the environment. Besides, a view-planner can be used by a human operator to provide guidance and ease the scanning task or by a mobile robot for an automatic exploration of the environment.

The existing research on view-planning algorithms can be categorized into two groups: view-planner with a prior model, view-planner with no prior model.

Planning with a prior model, also known as ”model-based planning” algorithms, was examined first with the art gallery problems [4, 5] and terrain guarding [6]. In the context of environment digitization, the emphasis has been primarily laid on the development of model-based algorithms [7] motivated by the widespread use of CAD software for sketching 2D floor plans or 3D modeling of the environment to be scanned. Yet these approaches are used before the site visit to plan a sequence of TLS positions to ensure the required completeness and quality of the final digitization. Several works [8, 9, 10, 11] have already been conducted to solve model-based

*Corresponding author.
Email addresses: `achakir.farouk@u-picadrie.fr` (Farouk Achakir), `s.elfkihi@um5s.net.ma` (Sanaa El Fkihi), `mouaddib@u-picardie.fr` (El Mustapha Mouaddib)

planning based on three main point cloud quality metrics: Completeness (coverage) [12, 13, 14], Level of Accuracy (LOA) / Level of Density (LOD) [15, 16] and the registration accuracy [17]. In addition, other metrics such as acquisition time [18] or other factors that affect the quality of the data, like the properties of the scanned surfaces [19] or the effect of range and incidence angle [20, 21], are considered in other works.

However, the 2D floor plan of the environment to be scanned is not always available and collecting accurate information about the environment for 2D floor plan sketching requires more time and labor. Such limitations motivated the development of automated approaches for planning with no prior model commonly known as Next-Best-View planning algorithms. These methods [22, 23, 24, 10] are mainly used with autonomous robots that operate in unknown environments with no prior knowledge of the scene. During the scanning process, the coverage of the scanned model is extended and updated until the desired termination condition is reached.

In this paper, we propose a non-model-based view-planner for achieving the task of autonomous exploration when the scanner is operating on a flat surface. In fact, it is usually the case in many environments such as office spaces, urban areas, open fields or in some cultural heritage applications. Several recent works in TLS view-planning [18, 14, 25] show that in this case, the complexity of the Next-Best-View planning can be reduced by considering the 3D environment as a 2D horizontal cross-section at the height of the scanner. Moreover, in mobile robotics exploration [26, 27, 28], using a 2D map which is efficient for navigation with a 3D scanner embedded on the robot for digitization, reduces the exploration time and provides a complete 3D model of the environment.

The first step of our view-planner called "Projective-Method" is a gap-based method used to extract occluded areas with the finest resolution. In contrast to the previous gap-based methods [28, 29, 30, 31] this avoids using predefined parameters which vary according to the scene's geometry. Moreover, the gap representation is simple, adapted to large environments, and does not require extensive computations such as ray-tracing volumetric methods [32, 33, 34] or Level-set based methods [35, 36].

The second step of our view-planner called Hierarchical Conservative Cells (HCC), is the proposed exploration strategy used to select the next-best sensing position according to the visibility computation. Unlike other non-model-based planning algorithms that generate a huge number of candidates either on frontiers [32, 34, 37], or by sampling the candidates over the entire scanned space [38, 39], our planner uses specific regions of the environment called conservative cells to drastically reduce the number of candidates. Both proposed algorithms are combined to achieve the best coverage, even for large and complex environments.

The rest of the paper is organized as follows. The proposed multi-objective view planning strategy is presented in section 2 and experimentally validated and compared with other view-planning approaches [28, 32, 33, 34] in section 3. Section 4 concludes the paper.

2. Proposed View Planning Strategy

In the context of environment reconstruction, efficient laser scanning digitization can be achieved through a Next-Best-View exploration strategy. The problem of non-model-based planning has attracted increasing interest from the research community, and a number of approaches have been proposed. This section presents the proposed view planning strategy, which is a novel approach for solving the non-model-based planning problem. The first work contribution is a method called "Projective-Method" used for solving the task of visibility analysis. The second part of this section describes the proposed Next-Best-View planner called "Hierarchical Conservative Cells (HCC)".

2.1. Visibility Analysis: Projective-Method

In complex environments, multiple scans acquired from different viewpoints are required to sample all visible surfaces of the scene. Each scanner at the location L_i provides a point cloud $C_i = \{{}^i\rho_0, {}^i\rho_1, ..., {}^i\rho_k\}$ within a local frame $\mathcal{F}_i$, with ${}^i\rho_k \in \mathbf{R}^2$ denotes the Cartesian coordinates of the k-th measurement returned by a single laser beam. The set of data points C_i bounds a region of free obstacles called the "visibility polygon"; the occluded regions in the environment from the scanner's position L_i are defined outside the boundary of the visibility polygon. Then, the registration process determines the transformation ${}^0T_i$ between each local frame $\mathcal{F}_i$ and a common global frame $\mathcal{F}_0$ through targets or appropriate registration algorithm. Depending on the use cases, for map building tasks where a scanner is embedded on a mobile robot, local registration methods such as Iterative-Closest-Point [1] or Normal-Distributions-Transform [3] can be used to converge faster. For better accuracy in registration results, global methods such as Coherent-Point-Drift [2] can be used with a Terrestrial laser scanner. In both cases, a final registered point cloud denoted C_0 is produced in the global frame $\mathcal{F}_0$ and is defined by:

$$C_0 = \bigcup_{i=1}^{N} {}^0T_i \cdot C_i = \bigcup_{i=1}^{N} {}^0C_i \tag{1}$$

where N is the number of scanner's positions.

Our aim is to produce a highly accurate representation of the visibility in the global point cloud C_0. To achieve this, our approach consists of solving the visibility problem in each local point cloud C_i. Then, the registration process is used to propagate the local visibility to the whole scanned space defined by C_0. We call this method: the Projective-Method.
We summarize this approach into four main steps, which are:

1. Gap detection step. The goal here is to produce a geometric representation (gaps) of the occluded areas (local visibility) in each local point cloud C_i.
2. Gap classification step. The aim of this step is to deal with the effect of the scanner angular resolution to enable a robust detection of occluded areas to point cloud sparsity.
3. Visibility polygon correction step. This step is used to identify in which areas the scanner cannot detect any object in its maximum range and then correct the point cloud and produce a valid local visibility polygon.
4. Gap propagation step. It allows occluded areas in the registered point cloud to be extracted by propagating the local visibility from each position L_i to the global scanned space.

Figure 1 illustrates the workflow of the proposed Projective-Method. We will give more details for each step in the following sections.

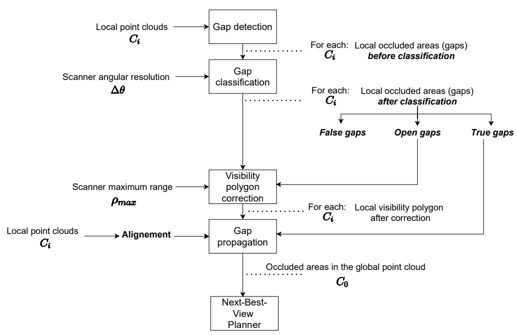


Figure 1: Flowchart of the Projective-Method which describes the interaction between the four mains steps: Gap detection, Gap classification, Visibility polygon correction and Gap propagation.

2.1.1. *Gap detection*

Computational geometry, level-set method, or ray-tracing methods are approaches to represent visibility in both local and global point clouds. However, gap analysis methods provide a simple geometric representation of visibility called a "gap". In fact, each gap corresponds to a segment that connects two measurements in the point cloud and indicates the existence of an occluded region behind it.

The gap detection is defined in the previous works [28, 29, 30, 31] as the process to detect a line-segment called "gap" denoted ${}^ig_k = ({}^i\rho_k, {}^i\rho_{k+1})$ that connects two measurements ${}^i\rho_k$ and ${}^i\rho_{k+1}$ in a local 2D point cloud C_i organized according to the polar angle of the measurements, such that:

$$|{}^i\rho_k, {}^i\rho_{k+1}| \geq \delta_{min} \tag{2}$$

Where δ_{min} is defined as a positive threshold.

The main challenge in these methods is to select an optimal threshold that is highly dependent on the characteristics of the environment (doorway widths or corridor widths in an indoor environment, or spaces between buildings in an urban setting). One can empirically select a static value for this threshold which remains constant throughout the gap detection process. However, in spite of the user's best efforts, the static threshold does not allow small gaps in the environment to be accurately detected. Indeed, this threshold is not necessarily suitable for all point clouds C_i because it does not take into account future needs.

We solve the problem by avoiding the use of a threshold. Thus, we detect the local-maxima in the variation of the euclidean distance between consecutive measurements in a 2D point cloud organized according to the polar angle of the scanned measurements. Each local-maximum at a position k is used to calculate a line-segment (gap) that connect two measurements ${}^i\rho_k$ and ${}^i\rho_{k+1}$.

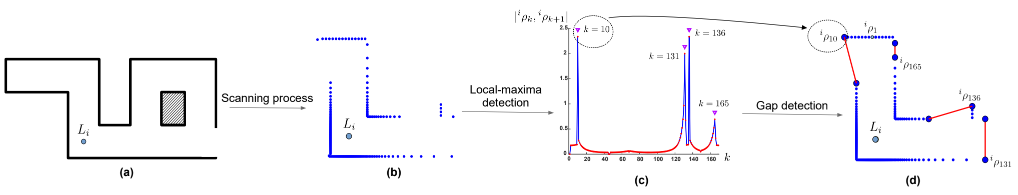


Figure 2: **(a)** Environment with the scanner position L_i. **(b)** Point cloud C_i with detected measurements ${}^i\rho_k$. **(c)** Local-maxima detection. **(d)** Gaps detected in C_i with the proposed approach.

Fig. 2a shows a laser scanner in the middle of a simple environment to illustrate the gap detection process with the proposed method. First, the scanner produces a point cloud C_i. Then, four local-maxima are detected at the positions $k = \{10, 131, 136, 165\}$, in this case (see Fig. 2c) in the point cloud C_i. For each position k, the measurement ${}^i\rho_k$ is linked to ${}^i\rho_{k+1}$ to create a gap denoted ${}^ig_k$. As the Fig. 2d shows, the proposed approach allows detection in the point cloud C_i of the final set of gaps $G_i = \{{}^ig_{10}, {}^ig_{131}, {}^ig_{136}, {}^ig_{164}\}$.

2.1.2. Gap classification

The gaps detection process provides a set of gaps (see Fig. 2d) that are supposed to show the presence of occluded areas in the environment. However, this process is directly related to the intrinsic characteristics of the scanner. Consequently, the maximal range of the scanner ρ_{max} and the angular resolution $\Delta\theta$ lead to production of a distance jump between two successive measurements similar to true gaps.

The gap classification process allows us to analyze each gap and classify it into one of the following three categories (see Fig. 3): "true gaps", "open gaps" and "false gaps". The first category defined as "true gaps" refers to all gaps that are aligned with the scanner's position L_i. Therefore, ${}^ig_k = ({}^i\rho_k, {}^i\rho_{k+1})$ is a "true gap" if:

$$d_\perp(L_i, {}^ig_k) \simeq 0 \tag{3}$$

where $d_\perp(L_i, {}^ig_k)$ denotes the orthogonal projection of the scanner position L_i on the extension of the segment gap ${}^ig_k$.

In addition, the further away the objects are from the scanner position, the larger is the distance between two successive points, which results in creating "false gaps". In fact, gaps of this kind are not aligned with the position L_i and the two points delimiting the gap are successive so the angle ${}^i\theta_{k,k+1} = \angle\, {}^i\rho_k L_i {}^i\rho_{k+1}$ is equal to the scanner's angular resolution $\Delta\theta$. The constraints that allow detection of "false gaps" ${}^ig_k$ are formulated as follows:

$$\begin{cases} d_\perp(L_i, {}^ig_k) > 0 \\ {}^i\theta_{k,k+1} \simeq \Delta\theta \end{cases} \tag{4}$$

When objects in the environment are beyond the maximum range of the scanner ρ_{max}, the laser beam cannot reach any obstacle. Therefore, it cannot produce any measurement during the scanning process and the gap detection creates an "open gap". In this case, the angle ${}^i\theta_{k,k+1}$ is higher than $\Delta\theta$ and the constraints that allow the detection of "open gaps" are:

$$\begin{cases} d_{\perp}(L_i, {}^i g_k) > 0 \\ {}^i\theta_{k,k+1} > \Delta\theta \end{cases} \tag{5}$$

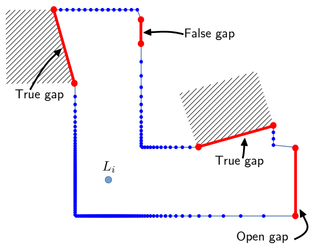


Figure 3: Detection of the three kind of gaps from the scanner's position L_i

After the gap classification, "false gaps" are eliminated from the point cloud, "open gaps" are used in the visibility polygon correction step (subsection 2.1.3) and "true gaps" are used in the gap propagation step (subsection 2.1.4).

2.1.3. Visibility polygon correction

Each point cloud C_i is a discrete representation that bounds an obstacles-free region called the visibility polygon, denoted Ω_i. This polygon is usually calculated to determine where the sensor can freely move. Therefore, the correct reconstruction of the visibility polygon is essential to ensure that the sensor does not hit any obstacles in the environment during its movements. However, the presence of an "open gap" in the point cloud C_i after the classification process indicates that the point cloud is not complete, which leads to a false visibility polygon.

The visibility polygon correction process is used to identify the endpoints of each "open gap", denoted $({}^i\rho_k, {}^i\rho_{k+1})$. Considering that all laser beams that pass through the open gap do not return any measurements, we reproduce the rotation of the laser beam from the point ${}^i\rho_k$ until it reaches the other endpoint ${}^i\rho_{k+1}$ and we substitute the real ranges by the simulated ranges. Fig. 4 shows the visibility polygon Ω_i after the correction process. We notice that the "open gap" in the point cloud is turned into two "true gaps".

2.1.4. Gaps propagation

In the case of a complex environment, it is generally necessary to move the scanner multiple times and generate several point clouds C_i that must be merged into a single point cloud C_0.

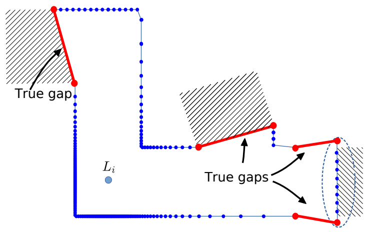


Figure 4: Visibility polygon correction allows to identify the open gap and recalculates the visibility polygon Ω_i.

Since it is no longer possible to organize the point cloud C_0 according to the polar angle of the measurements, the gap-based methods become useless after the alignment process.

Besides this, determining the local visibility in each point cloud C_i does not allow the hidden areas in C_0 to be guessed after the registration process. For example, a gap can be occluded from a scanner position L_i but visible from another position $L_{j\neq i}$.

To deal with this, we propose a process that we name "gap propagation" to evaluate the impact of all scans on the visibility of all gaps. This process allows us to create the set of fragments of gaps G_0 in the global point cloud C_0 using the geometric transformations ${}^0T_i$ previously defined in the equation 1 and the Boolean operators of polygon topology: (OR) denoted $\cup$, (AND) denoted $\cap$, the Boolean complement (NOT) such as the complement of the polygon Ω_i, denoted Ω_i^c and the Boolean exclusive union (XOR) denoted $\oplus$.

These operators will be applied to the two geometric objects that are involved in our process: The visibility polygons Ω_i and the sets of "true gaps" G_i.

The main idea behind our proposed solution is to project each set of "true gaps" $G_i = \{{}^ig_1; {}^ig_2; ...; {}^ig_k; ..\}$ from the scanner position L_i in the visibility polygons $\Omega_{j\neq i}$ of the other positions of the scanner. Then, we identify and classify which parts of each gap in the set G_i are visible (Definition 1) or invisible (Definition 2) in the others scanner positions $L_{j\neq i}$. We call these gap parts "Fragments".

Definition 1. *The visible fragments in the set G_i associated with the position L_i consists of gaps parts that are visible from at least one other position $L_{j\neq i}$. This set denoted ${}^iF_{vis}$ is calculated* *using the following equation:*

$$
\begin{aligned}
{}^{i}F_{vis} &= \bigcup_{j \neq i} \left({}^{0}T_i \cdot G_i \cap {}^{0}T_j \cdot \Omega_j\right) \\
&= \bigcup_{j \neq i} \left({}^{0}G_i \cap {}^{0}\Omega_j\right) \\
&= {}^{0}G_i \cap \bigcup_{j \neq i} {}^{0}\Omega_j
\end{aligned}
\tag{6}
$$

where ${}^{0}G_i = {}^{0}T_i \cdot G_i$ *and* ${}^{0}\Omega_i = {}^{0}T_i \cdot \Omega_i$

Definition 2. *The invisible fragments in the set* G_i *associated with the position* L_i *consists of gaps parts that are invisible from all the other positions* $L_{j \neq i}$*. This set is denoted* ${}^{i}F_{inv}$ *and calculated using the following equation:*

$$
\begin{aligned}
{}^{i}F_{inv} &= {}^{0}T_i \cdot G_i \oplus {}^{i}F_{vis} \\
&= {}^{0}G_i \oplus {}^{i}F_{vis}
\end{aligned}
\tag{7}
$$

Fig. 5 summarizes the gap propagation of the set G_1 in the case of three scanner's positions L_1, L_2 and L_3. The final set ${}^{1}F_{inv}$ is illustrated as the subset of G_1 after the propagation process. First, the set G_1 and the visibility polygons Ω_2 and Ω_3 associated with the scanner's positions L_2 and L_3 are projected in the same global frame $\mathcal{F}_0$. Then, the *definition 1* is used to extract the set of visible fragments ${}^{1}F_{vis}$ from the set of gaps G_1. Finally, the *definition 2* is used to extract the set of invisible fragments ${}^{1}F_{inv}$.

Therefore, the set of invisible fragments G_0 in the global point cloud C_0 can be defined by the fusion of all invisible fragments ${}^{i}F_{inv}$:

$$
G_0 = \bigcup_{i=1}^{N} {}^{i}F_{inv} \tag{8}
$$

Fig. 6 shows the results of the gap propagation process using the equation 8 with the same three scanner's positions L_1, L_2 and L_3. Then, the gap propagation process extracts the sets of invisible fragments ${}^{1}F_{inv}$, ${}^{2}F_{inv}$ and ${}^{3}F_{inv}$ associated with each position of the scanner. Finally, the fusion of these sets allows G_0 (red segments) to be found. This represents the fragments in the global point cloud C_0. Moreover, the fusion of all visibility polygons represents the scanned region, also known as the free space Ω_0 where:

$$
\Omega_0 = \bigcup_{i=1}^{N} {}^{0}\Omega_i \tag{9}
$$

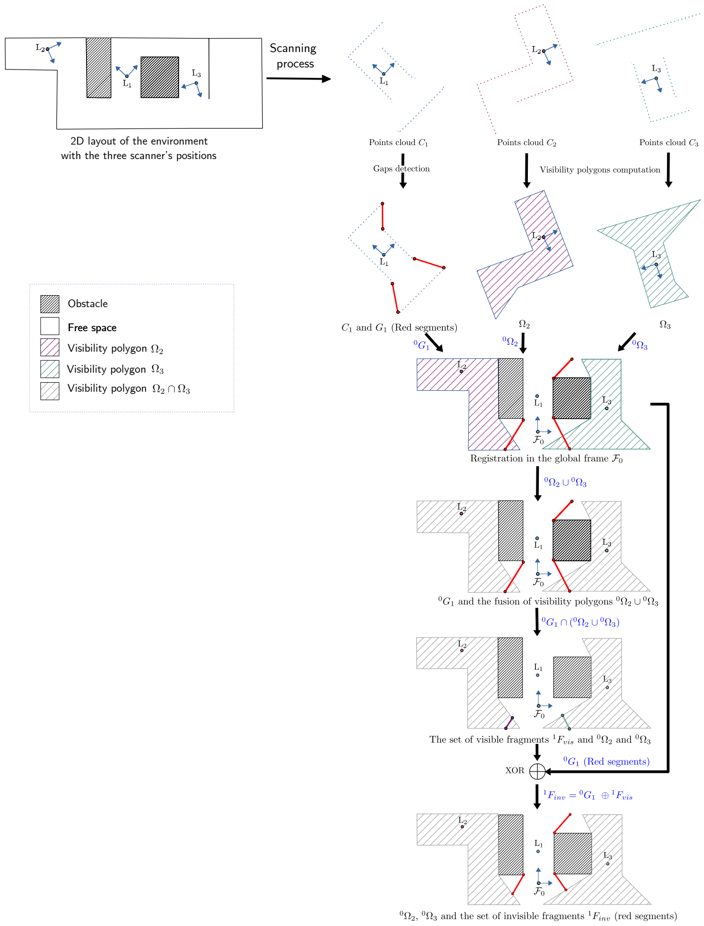


Figure 5: The gap propagation process of the set G_1 in the case of three scanner's positions (L_1, L_2, L_3) allows extraction of the set of occluded fragments ${}^1F_{inv}$. After the alignment process, the set G_1 is projected in the fusion of visibility polygons $\Omega_2 \cap \Omega_3$ to extract the subset of invisible fragments ${}^1F_{inv}$

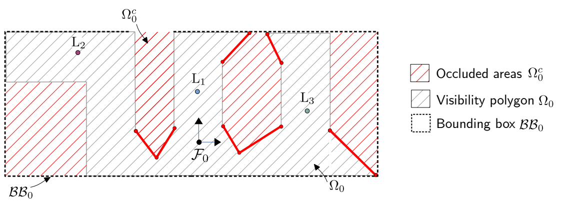


Figure 6: The gap propagation process in the case of three scanner's position L_1, L_2 and L_3. The red segments represents the set of fragments $G_0 = {}^1F_{inv} \cup {}^2F_{inv} \cup {}^3F_{inv}$. The red dashed areas inside the bounding box $\mathcal{BB}_0$ represent the non-scanned region Ω_0^c. The green dashed areas represent the fusion of all visibility polygons Ω_0

This region Ω_0 is delimited within a rectangular area called the bounding box $\mathcal{BB}_0$.

Thereafter, the non-scanned region Ω_0^c in the environment is defined as the complement of Ω_0 inside the bounding box $\mathcal{BB}_0$ where:

$$\Omega_0^c = (\bigcup_{i=1}^{N} {}^0\Omega_i)^c \tag{10}$$

2.2. *Next-Best-View planning*

The Projective-Method described in the previous section allows the visibility problem to be solved in a point cloud defined as the merging data C_i acquired from several positions L_i of the scanner. Then, the visibility is represented by the set of fragments G_0, the free space Ω_0, and the non-scanned region of the environment Ω_0^c.

The frontier between the free space Ω_0 and the non-scanned region Ω_0^c is defined by a set of regular edges and the fragments G_0. The regular edges are a simplified polygonal representation of the scanned environment. However, a fragment is only a link between two measurements in the point cloud, it doesn't represent any existing part of the real environment. Consequently, the only available apertures to look inside the region Ω_0^c are the set of fragments $G_0 = \{{}^0g_1, {}^0g_2, ..., {}^0g_k\}$ calculated by the Projective-Method (Equation 8).

Based on this observation, we propose an approach to maximize the scanner visibility through the fragments G_0 and to optimize the motion of the scanner.

In order to do this, we propose an approach called Hierarchical Conservative Cell Exploration Strategy that consists of three major steps:

- Generation of a set of potential candidates for the next sensing position.
- Evaluation of each generated candidate according to the selected criteria.
- Selection of the best candidate as the next sensing position.

The following subsections detail this approach.

2.2.1. *Candidates generation*

A human operator or a robot carrying the scanner can only move and choose a specific position in the free space Ω_0. A number of proposed approaches [40, 41, 42] for Next-best-view planning use this constraint to generate a set of candidates to subsequently select the best one based on chosen criteria. However, the major limitation of these approaches is the huge number of the generated candidates in the entire free space Ω_0 after each scanner displacement.

Consequently, the design of our view-planning approach is mainly focused on reducing the number of generated candidates by working on specific regions of the free space that we called "conservative cells" (definition 4).

Definition 3. *An inflection is a variation in the sign of the curvature of the visibility polygon Ω_0 boundary. The inflection ray is calculated by extending a ray from the inflection.*

Definition 4. *The conservative cell associated with a segment is defined as the area delimited by the segment-line itself and the extended inflection rays from each of the segment's endpoints until the rays hits each others.*

The particularity of a conservative cell is that the visibility of a specific sensor is the same regardless of its position inside it. It is therefore possible to use both vertices of a fragment in the point cloud C_0 to create the conservative cells. As shown in Fig. 7, regardless of the blue candidate's positions inside this region, the same visibility through the gap will be obtained. Therefore, it is no longer necessary to generate several candidates in the free space but only one position associated with each conservative cell.

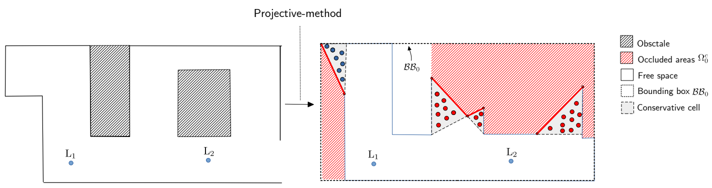


Figure 7: The Projective-Method allows detection of the fragments (red segments) and the occluded areas in the environment (red dashed area). Then, each fragment is associated with a conservative cell. The blue candidates inside the conservative cell associated with the convex region have the same visibility. However, the red candidates that observe a concave region have different visibility.

Definition 4 is only possible in the case of fragments associated with a convex region. As we can see in Fig. 7, the visibility of the red points changes according to their positions, which confirms that the use of conservative cells is no longer possible in the case of concave regions. Therefore, we introduced the *proposition 1* that allows a specific position to be selected where the sensor's visibility is maximized through a fragment associated with a concave region.

Proposition 1. *The visibility through a fragment associated with a concave region is maximized on one of the fragment's endpoints and minimized on the other one.*

PROOF. A concave region can be subdivided into at least two convex sub-regions. Each convex sub-region can be associated with a conservative cell represented by a candidate. The candidate that is contained in all conservative cells allows observation of the maximum number of convex sub-regions. Therefore, the visibility is maximized when the sensor is placed on this candidate position.

In Fig. 8, we present an example of the candidate generation process. In the case of a fragment associated with a convex region, only one candidate (blue) is generated inside the conservative cell. In the other case where the concave region is bounded by three fragments, a candidate (red) is generated on the endpoint that maximizes the visibility of each of them according to the *proposition 1*.

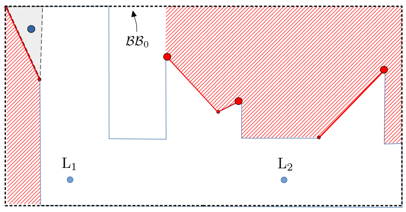


Figure 8: Results after the candidates generation process.

The following subsection proposes some criteria for evaluation of the generated candidates.

2.2.2. Evaluation of Candidates

A candidate location for the next sensing position c_i allows observation of a part or all the non-scanned region Ω_0^c through the set of fragments G_0.

Each candidate associated with a specific fragment might be able to scan regions associated with other fragments. In this case, some areas of the environment may obstruct the visibility of the candidate. To address this problem, we propose subdividing the candidate's field of view into two levels of visibility.

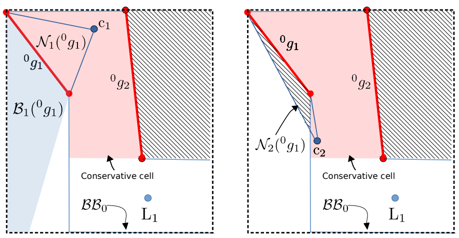
The first level of visibility denoted $\mathcal{N}_i({}^0g_k)$ is represented by the region between c_i and the fragment ${}^0g_k$. It determines whether a fragment is directly visible or not from c_i. Therefore, a fragment ${}^0g_k$ is invisible from the candidate position c_i if:

$$\mathcal{N}_i({}^0g_k) \cap \Omega_0^c \neq \emptyset \tag{11}$$

When a fragment is visible, the candidate uses the second level of visibility to observe the area beyond the fragment. This level of visibility is denoted $\mathcal{B}_i({}^0g_k)$ and allows measurement of the area of Ω_0^c inside the region between the fragment ${}^0g_k$ and the bounding box $\mathcal{BB}_0$ of the point cloud C_0. This region is denoted $\mathcal{R}({}^0g_k, \mathcal{BB}_0)$ and the level of visibility $\mathcal{B}_i({}^0g_k)$ is defined by:

$$\mathcal{B}_i({}^0g_k) = \mathit{Area}(\mathcal{R}({}^0g_k, \mathcal{BB}_0) \cap \Omega_0^c) \tag{12}$$

Fig. 9 illustrates two candidates c_1 and c_2 inside the conservative cell associated with ${}^0g_2$ that observe another fragment ${}^0g_1$ using two levels of visibility. The fragment ${}^0g_1$ is visible for c_1 but hidden for c_2 because $\mathcal{N}_2({}^0g_1) \neq \emptyset$



(a) (b)

Figure 9: **(a)** The fragment ${}^0g_1$ is visible for the candidate c_1. **(b)** The same fragment is hidden for c_2 because $\mathcal{N}_2({}^0g_1) \neq \emptyset$

As, the candidate c_i can observe more than one fragment in the set G_0, the information gain $\mathcal{I}_g(c_i)$ of c_i is defined by:

$$\mathcal{I}_g(c_i) = \sum_{k=1}^{\mathcal{V}(c_i)} \mathcal{B}_i({}^0g_k) \tag{13}$$

where $\mathcal{V}(c_i)$ is defined as the number of visible fragments.

In addition to the information gain criterion, the distance between each candidate's position and the current scanner's position is an important factor in optimizing the time required for the scanning process. Therefore we add the length of the collision-free path $\mathcal{L}_{\mathcal{P}}(c_i)$ between the current scanner's position and the candidate c_i as an additional criterion to define the final score s_i:

$$s(c_i) = \lambda \cdot \mathcal{I}_g(c_i) + (1-\lambda) \cdot {\mathcal{L}_{\mathcal{P}}}^{-1}(c_i) \tag{14}$$

The positive constant $\lambda \in [0, 1]$ is used to weight the cost of the motion against the expected gain of information.

In order to select the value of λ that must provides the best compromise between the final path length and the minimum number of candidates to cover the entire environment, we evaluated our planning algorithm in different scenarios (indoor offices, outdoor open fields). To show how we carried out the selection of λ in this paper, we use the following scenario: a scanner is positioned at L_1 in (Fig. 10).

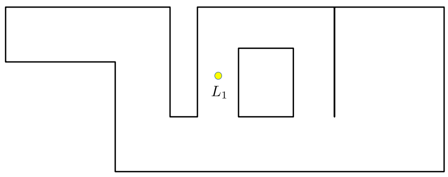


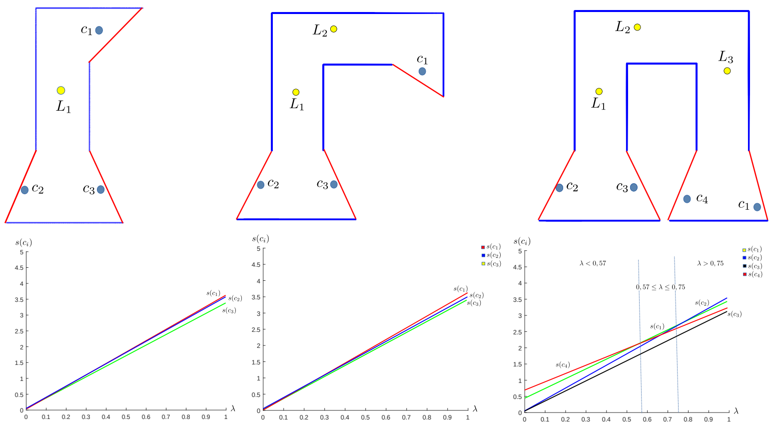
Figure 10: An example of environment with one scanner positioned at L_1

Then, our Projective-Method detected three gaps represented by three candidates c_1, c_2 and c_3 (Fig. 11a). For each candidate, we have plotted the variation of scores $s(c_1)$, $s(c_2)$, $s(c_3)$ according to λ. These plots show that the score of the candidate c_1 is the highest one for all values of λ. It is therefore selected to become the second scanner's position L_2. From the position L_2, the same process selects the candidate c_1 (Fig. 11b).

However, the third scan from the position L_3 (Fig. 11c) reveals four candidates and none of them shows a maximum score for all values of λ. In this case there are three different situations:

- $\lambda < 0.57$: The score $s(c_4)$ is the highest and the candidate c_4 is selected to be the position for L_4. This case is illustrated in Fig. 16a.

- $0.57 \leq \lambda \leq 0.75$: The score $s(c_1)$ is the highest and the candidate c_1 is selected to be the position L_4. This case corresponds to the choice of our view-planner and is illustrated in Fig. 15c.



(a) (b) (c)

Figure 11: Candidates score with different values of λ after **(a)** the first **(b)** the second **(c)** and the third scanner displacement.

- $0.75 < \lambda$: The score $s(c_2)$ is the highest and the candidate c_2 is selected to be the position L_4. This case is illustrated in Fig. 16b.

We notice that the shortest path to explore the environment is generated with $0.57 \leq \lambda \leq 0.75$. Thus, after conducting further similar experiments in other environments we consider the value $\lambda = 0.6$.

2.2.3. Selection of the candidate

After the candidates generation and evaluation process, the HCC view planning algorithm must select the next best view. Therefore, each fragment is represented by a single candidate selected according to the associated region (convex, concave):

- **Fragment associated with a convex region:** All candidates inside the conservative cell of the fragment share the same visibility. Thus, only one candidate inside the cell can be chosen to represent the fragment. The selected candidate in our proposed strategy is located at $\mathcal{L}(c_i)$ which is the center of the conservative cell (*definition 4*).

- **Fragment associated with a concave region:** The selected candidate is located at $\mathcal{L}(c_i)$ which is the end-point of the fragment that maximizes the visibility (*proposition 1*).

The location $\mathcal{L}(c_i)$ of each candidate c_i allows calculation of the associated score $s(c_i)$ and number of visible fragments $\mathcal{V}(c_i)$. The candidates are then prioritized according to their scores and level.

The HCC selection strategy consists of identifying a candidate c_i that strikes a balance between a high score and the maximum number of visible fragments.

On the one hand, a high score means that the candidate provides significant information gain at a short distance from the current scanner's position. On the other hand, observing a large number of fragments allows a large number of invisible areas to be scanned in a short amount of time.

For example, as shown in Fig. 12a, c_3 is a candidate with a very high score but it can observe only one fragment $\mathcal{V}(c_1) = 1$. However, observing many fragments, as candidate c_6, does not necessarily guarantee the highest score.

As a solution, we propose to increase the initial score of each candidate c_i according to the number of observable fragments $\mathcal{V}(c_i)$. First, the disparity in the scores of the candidates is evaluated by the mean square of successive differences (MSSD), denoted $\delta^2(s)$.

Therefore, a candidate observing more than one fragment $\mathcal{V}(c_i) > 1$ gets an initial score increased by $(\mathcal{V}(c_i) - 1) \cdot \delta^2(s)$. However, a candidate c_i observing a single fragment $\mathcal{V}(c_i) = 1$ remains at the same initial score.

Finally, the next sensing position is defined according to the following formula:

$$NBV_{i+1} = \mathcal{L}(c_k \mid k = \underset{x}{\operatorname{argmax}} \{s(c_x) + \delta^2(s) \cdot (\mathcal{V}(c_x) - 1)\} \tag{15}$$

where L_i is the current scanner position.

Fig. 12 shows a set of candidates before and after the NBV selection where the MSSD is equal to 5.22 and $\lambda = 0.5$. The candidate c_7 is selected as the next-best-view since it allows the right balance between the score and the number of visible fragments $\mathcal{V}(c_7) = 3$.

2.2.4. ***Termination condition***

The NBV process is stopped when the length of each fragment in the set G_0 is smaller than a predefined threshold. If this test is verified after the gap propagation process, then the environment digitization is complete. Otherwise, the selection strategy passes to the candidates generation step. This termination condition allows the handling of environments with small geometric features.

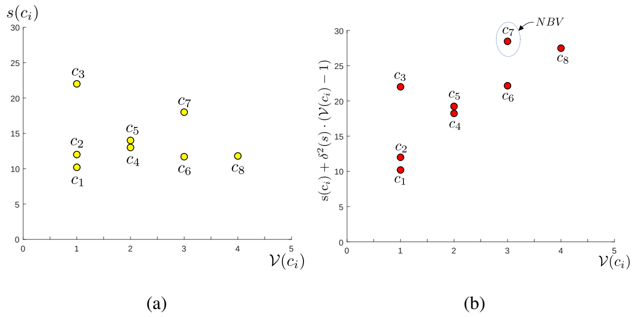


Figure 12: **(a)** The set of candidates before the NBV selection. **(b)** The candidate c_7 is selected as the NBV.

3. Results

To validate our approach and evaluate the entire process of our methods, we performed tests on synthetic data generated by a simulator that we developed for this specific task. This simulator allows the creation of 2D environments, selection of the desired laser scanner positions, and generation of the associated local and global 2D point clouds. Thereafter, we consider the case of a real point cloud. For this purpose, a mobile robot equipped with a 2D scanner is used to explore and perform a complete scan of an indoor environment. Finally, the performances are evaluated by making a comparative study with four others view-planners [28, 32, 33, 34] and the performance of an experienced operator in a large environment (The esplanade of Hassan in Morocco).

3.1. Simulated experiments

As a first step, results obtained in a simple simulated environment are presented within the context of two scenarios. The first simulated experiment demonstrates the concept of our approach when the sensor is a Terrestrial Laser Scanner (TLS) used by a human operator. The second one presents the situation of a sensor embedded on a mobile robot.

3.1.1. Terrestrial laser scanner

In this scenario, the displacement of the sensor is performed by a human operator based on its observations and prior knowledge of the environment. The main purpose of this experiment is to demonstrate the interest of using our process as a computer-aid solution to guide the decision of the operator in selecting the next sensing position. We assume that he is equipped with a TLS and aims to cover the whole environment illustrated in Fig. 13a. In order to achieve this, he observes the environment and intuitively moves the scanner four times in such a way that the global point cloud covers a large area of the environment. Afterwards, he decides to evaluate the completeness of the point cloud to make the next decision. To do this, the Projective-Method is used as a computer-aid solution to analyze the visibility in the global point cloud and generate the fragments associated with the non-scanned areas (Fig. 13b).

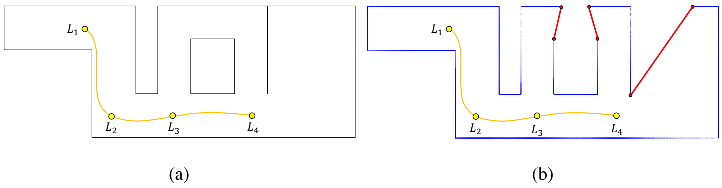


Figure 13: **(a)** Real (and unknown) environment (black lines) with all four scanner positions (yellow dots). **(b)** The global point cloud C_0 with the invisible area surrounded by the set of gaps.

Then, the next position for the scanner can be chosen based on the information provided by the Projective-Method. Nevertheless, the use of a view-planning method brings additional information that can help the human operator to choose the next best position for the scanner. As the Fig. 14 shows, the use of our HCC method allows generation of a NBV position denoted L_5. Once the TLS is moved, the exploration strategy detects that a non-scanned area remains in the point cloud and therefore provides a last position L_6 to cover all the environment.

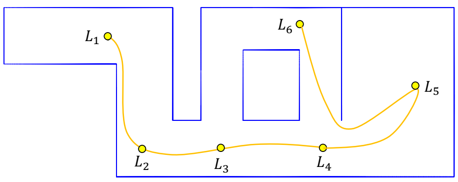


Figure 14: The human operator moves the scanner four times to positions L_1, L_2, L_3 and L_4. Then, our view-planner (HCC) automatically calculates the positions L_5 then L_6 which close all the gaps in this case.

3.1.2. Mobile robot

The following simulated experiment reproduces the case where a mobile robot equipped with a laser scanner attempts to automatically explore its environment. Unlike a human, the robot is unable to do multiple scans by itself. Consequently, our exploration strategy will not be used as a computer-aid solution in the middle of the scanning process but will be necessary in the first scan until the termination condition is verified.

As shown in Fig. 15, we assume that the robot (represented by the sensor location) is initially positioned at L_1. First, from this position, the gap detection process is used to determine the initial non-scanned areas to be explored. The result shown in the Fig. 15a reveals that there are two distinct unexplored areas denoted R_1 and R_2 behind three gaps (red fragments). Then, the HCC method generates three candidates (c_1, c_2, c_3) and selects c_1 as the NBV using the equation 15.

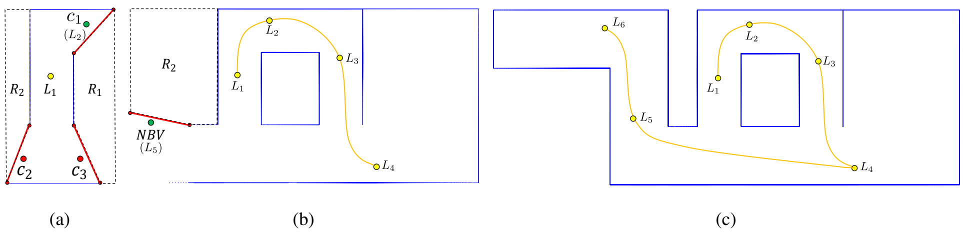
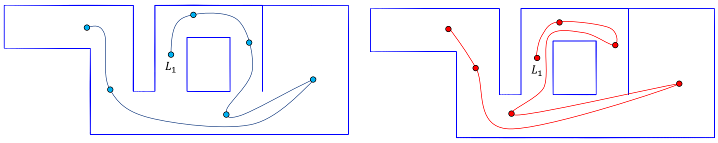


Figure 15: **(a)** The gap detection process reveals two non-scanned areas near to the initial robot position L_1. **(b)** The robot remains inside the region R_1 during the exploration process. **(c)** The final robot path using the exploration strategy HCC.

Fig. 15b illustrates an intermediate step in the exploration process and we notice that the robot remained inside the region R_1. The reason is that the HCC uses a distance constraint which allows the complete exploration of a region before moving to another one. For example, Fig. 15c shows that the final path of the robot avoids unnecessary back and forth displacement between the region R_1 and R_2 during the exploration process.

In comparison with the path generated with our view-planning strategy (Fig. 15c), a strategy where the next sensing position is selected as the nearest candidate to the robot's current position (distance-based strategy) (Fig. 16a) or otherwise an exploration strategy where the next sensing position is the candidate that provides the maximum information gain (information gain-based strategy) (Fig. 16b) usually produces paths with many round trips, which greatly increases the time required to explore the entire environment.

Subsequently, the HCC was also evaluated in more complex environments, as shown in the Fig. 17. In the case of the chosen environments, it is relatively complicated even for a human to find a minimal number of TLS positions to scan the entire space. The simulated results show



(a) (b)

Figure 16: The blue path is produced by the distance based strategy with **(a)**: $\lambda < 0.57$ and **(b):** The red one is based on the information gain criterion with $\lambda > 0.75$

that our exploration strategy allows us to generate a sequence of positions until the termination condition is verified with no more gaps in the global point cloud. The path that connects the generated positions shows that the HCC avoids unnecessary back and forth movement in the environment.

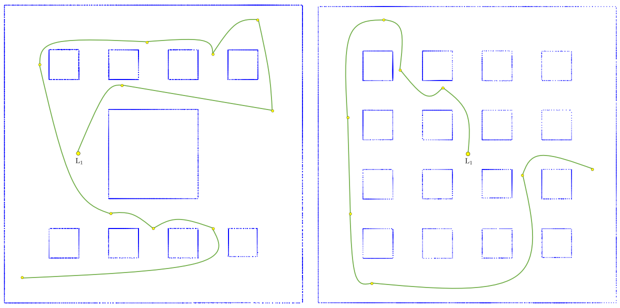


Figure 17: The path generated by the HCC in the case of more complex environments

3.2. *Real-time experimentation: Mobile robot*

The experiment's objective was to evaluate the correct working of the experimental workflow that is subdivided into two mains stacks summarized in the flowchart on Fig. 18. The first stack is dedicated to the processing of the point cloud generated by the robot's laser. It allows mainly to detect and classify the gaps, to identify the invisible areas in the environment with the Projective-Method and to compute the NBV using the HCC method. The robot stack is primarily composed of ROS nodes. It allows the robot localization in the environment, path planning and following.

This experiment was conducted with a mobile robot equipped with a 2D laser scanner with a range of 3m and a field of view of 120° (Fig. 19).

The robot operates inside an indoor environment (length around 15 meters), mostly made up of corridors and several office rooms (Fig. 20).

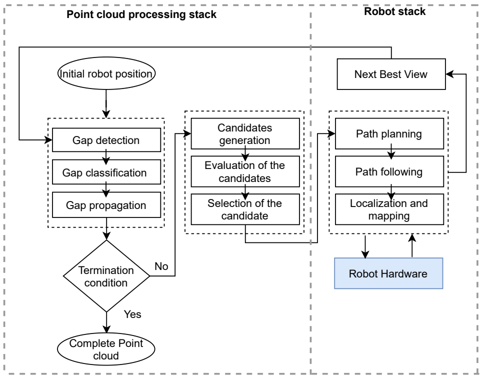


Figure 18: Experimental workflow which describes the interaction between the point cloud processing and the robot stacks that allows interaction with the robot hardware (laser scanner and motors)

(a) (b)

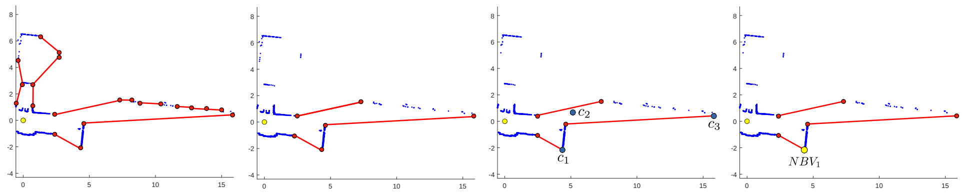
Figure 19: **(a)** Omron Adept MobileRobots Pioneer LX. **(b)** SICK LMS 300

(a) (b)

Figure 20: **(a)** Initial position of the Pionner LX in the environment **(b)** Another point of view of the environment

The first position of the robot provides a single point cloud C_1 with the corresponding gaps (Fig. 21a). Then, the gap classification allows "true gaps" to be selected (Fig. 21b). With only one position of the scanner, there is no reason for the gap propagation step. Consequently, the three "true gaps" of the point cloud C_1 are used to generate three HCC candidates (c_1, c_2, c_3) (Fig. 21c) and then to select c_1 as the next sensing position NBV_1 of the robot (Fig. 21d).

Subsequently, this position is used into the robot stack and allows the robot to automatically move into the environment to the desired position.



(a) (b) (c) (d)

Figure 21: **(a)** Set of gap detected in the point cloud C_1 before gaps classification. **(b)** Set of true gaps in the point cloud C_1 after the classification process. **(c)** HCC candidates (blue dots). **(d)** Selected Next Best View.

Once the robot reaches the goal position, and since its field of view is limited to $120°$, it makes a complete turn on itself to scan the surrounding environment. At the end of this operation, a point cloud C_2 is produced.
The second step of the experimentation allows us to validate the Projective-Method. For this purpose, the point clouds C_1 and C_2 are used to determine the areas that remain to be explored (Fig. 22), and the HCC method recalculates a new NBV position.

This cycling process of observing and deciding where to move for the next sensing position stops when the length of the remaining fragments are less than a predefined threshold (0.1 m). Fig. 23 shows the final result of the path followed by the robot to explore its environment and the acquired point cloud.

3.3. *A comparative study*

Automated view-planning has been addressed in the literature by several approaches using different strategies to achieve a complete coverage of the explored environment. Each strategy aims to calculate the shortest path to explore the entire environment in the least amount of time. Therefore, in order to evaluate the performances of our view-planner, we propose to compare it with four 2D exploration approaches: Gap navigation tree (GNT) [28], Frontier-Based Approach for Autonomous Exploration (Yamauchi) [32], Histogram based frontiers (HFB) [34] and a volumetric-based strategy for exploring indoor environments (Gonzalez) [33]. The following table (Table 1) summarizes each of the selected methods for the performances assessment.

We have simulated the selected methods in the same environment illustrated in the Fig. 24. All the methods are initialized from the same scanner's position L_1, and at each new position,

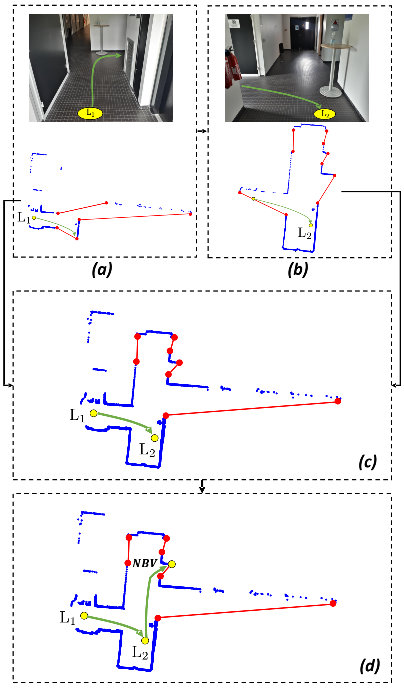


Figure 22: **(a)** Set of gaps in the first point cloud C_1 and **(b)** the second point cloud C_2 after the gaps classification process. **(c)** Set of gaps in the global point cloud after the Projective-Method. **(d)** Selected Next Best View after the HCC method.

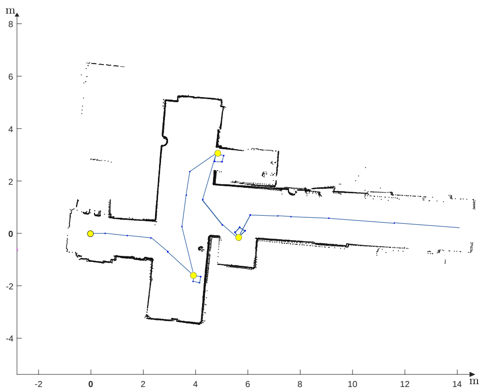


Figure 23: Final computed path using the Projective-Method to solve visibility and the HCC method to find the next sensing position.

we have recorded the computation time as well as the accumulated path length.

As shown in Fig. 25a, Yamauchi [32] and HBF [34] are very time consuming because of

View-planner	Environment representation	Exploration strategy
Yamauchi [32]	2D occupancy grid map	The nearest unexplored frontiers in the grid map is selected as the NBV.
HBF [34]	2D occupancy grid map	An utility function (distance from the current position, size of the frontier) is used to evaluate a score for each frontier in the grid map. The frontier with the highest score is selected as the NBV.
Gonzalez [33]	Point cloud / gap-based	A set of candidates is generated randomly across each gap. An utility function is used to select the nearest candidate with the highest expected gain of information is selected as the NBV.
GNT [28]	Point cloud / gap-based	The gap navigation tree is used to classify the detected gaps. Then, the tree is explored from the parents-nodes to the children-nodes.

Table 1: Exploration strategy of the selected view-planner for the performances assessment

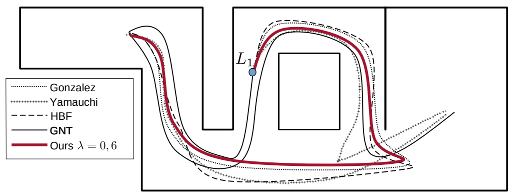


Figure 24: Final paths generated by each of the selected view-planners.

the ray-tracing used with the occupancy grid map to extract the frontiers. Gonzalez [33] is a gap-based strategy that avoids ray-tracing but the use of a large number of candidates for the next sensing position leads to an increase in the computational time. However, it seems that the GNT produces a very interesting results in terms of calculation time but the simplicity of this exploration strategy results in a large cumulative path as shown in Fig. 25b.

In contrast, our method (Ours $\lambda = 0.6$) avoids using ray-tracing and reduces the number of candidates thanks to ”conservative cells”. Moreover, using an utility function to evaluate the

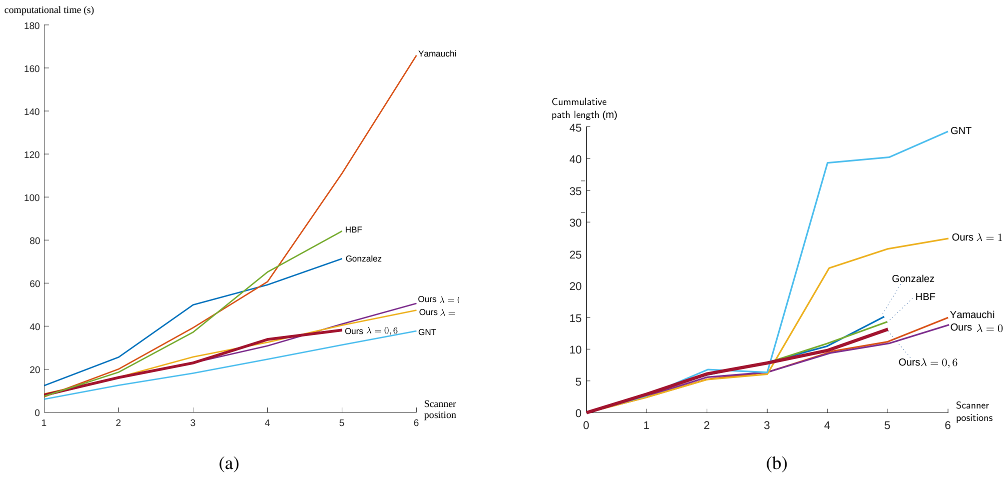


Figure 25: **(a)** Performances comparison according to computational time. **(b)** Performances comparison according to the cumulative path length.

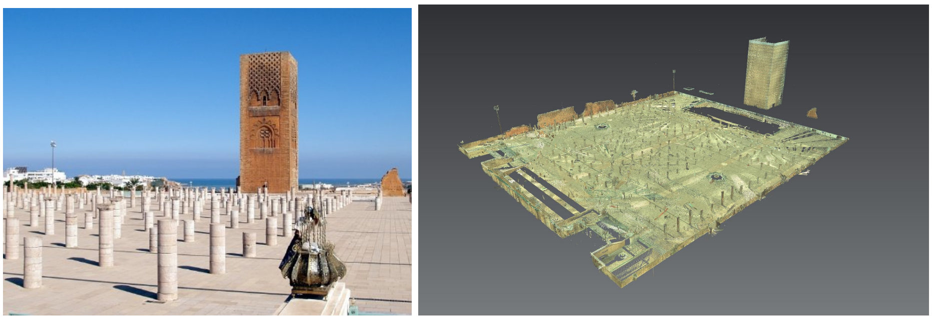
score of the candidates allows us to avoid a large cumulative path with only five scanner positions. A strategy based solely on distance (Yamauchi, Ours $\lambda = 0$) or only expected information gain (Ours $\lambda = 1$) generates six positions to cover the entire environment.

3.4. Offline experimentation: HCC vs Human operator

Humans are efficient at view-planning for coverage of a simple environment but even experienced operators will encounter considerable difficulty in complex environments to provide optimal scanner placement. Therefore, in a scanning project, these locations are determined empirically according to the operator's experience and the site conditions. To assist the human operator, view-planning algorithms such as Hierarchical conservative cells (HCC) are used to find the optimal placement and minimize the number of scanner locations.

In order to assess the performance of our view-planning algorithm (HCC), we proposed to compare the results achieved by an experienced human operator and our view-planing algorithm in the same complex environment.

The chosen environment is the esplanade of Hassan in Morocco (Fig. 26a). The challenge in scanning this environment lies in the surface area that exceeds 2000 m^2 and also in the presence of around 500 columns surrounded by a wall. In June 2015, as a part of the ATHAR3D project [43], we carried out the digitization of this esplanade using lasergrammetry (Leica C10 scanner).

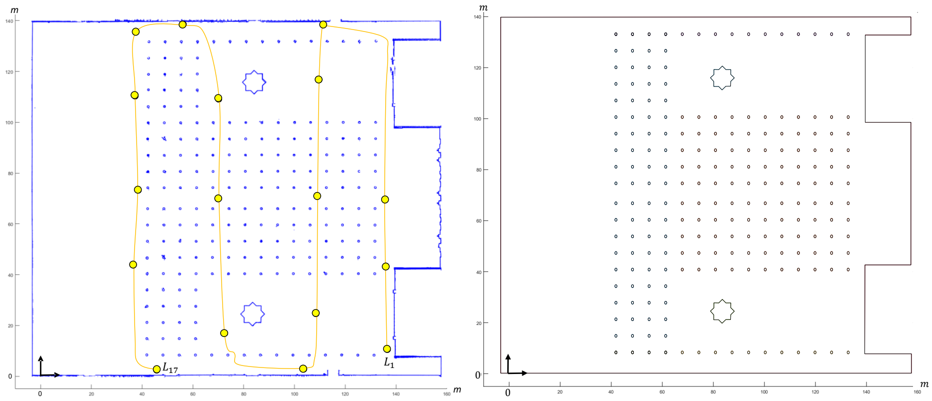


(a) (b)

Figure 26: **(a)** The esplanade of Hassan in Rabat (Morocco). **(b)** The 3D points cloud of of the entire esplanade after 17 scans.

We made 17 scans over the entire esplanade to cover all the columns. The final 3D point cloud is shown in figure 26b.

In order to make a comparison with the human operator results, we selected a 2D slice of this point cloud (Fig. 27a) to create a high-fidelity environment (Fig. 27b) on our simulator.



(a) (b)

Figure 27: **(a)** The 17 scanner positions used to cover the centre of the esplanade with the corresponding 2D slice of the points cloud **(b)** The simulated environment of the centre of the esplanade.

Thereafter, the simulator was configured to start the HCC method from the same initial position chosen by the human operator and the results obtained are shown in Fig. 28.

In order to estimate the performance of our view-planning method, we use the coverage rate of the scanning scene. The simulator allows us to estimate the coverage rate inside the region A_0

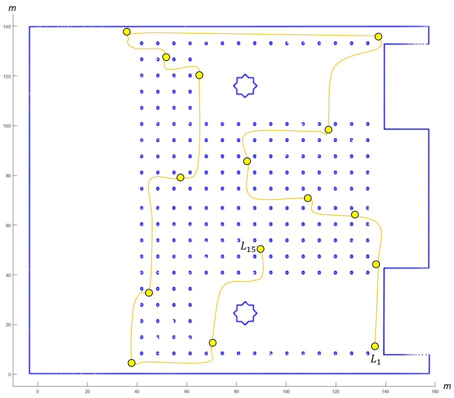


Figure 28: Final computed candidates with the HCC in the simulated environment of the center of the esplanade of Hassan.

(Fig. 29a) after each scanner displacement in both HCC and human operator placement of the viewpoints. The coverage rate is defined as:

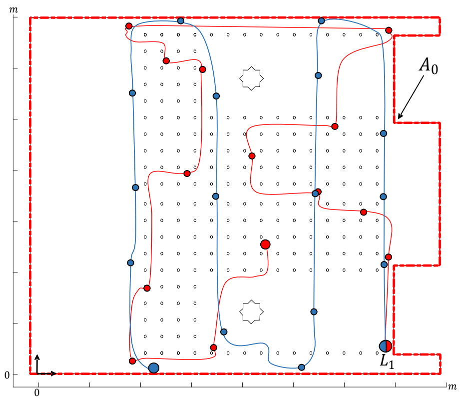
$$coverage(\%) = \frac{Area(\Omega_0)}{Area(A_0)} \cdot 100 \tag{16}$$

Where Ω_0 is defined in the equation 9 as the fusion of local visibility polygons also knows as the free space.

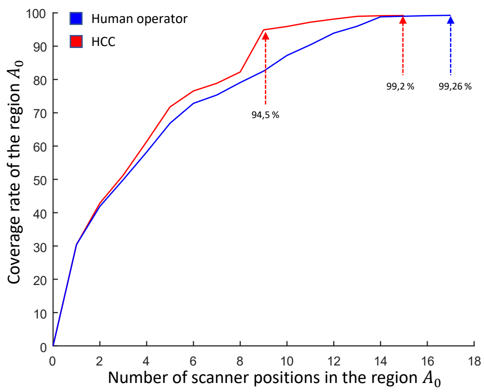
Results in Fig. 29b shows that the HCC method generates 15 scanner positions to ensure a coverage rate close to 99.2% which is similar to the results obtained by the human operator (99.08% after 15 scanner displacements and 99.26% after 17 scanner displacements). The scanning process is stopped when the lengths of all remaining fragments in the environment are smaller than a chosen threshold which is $0.5m$ in this example. We notice that the viewpoints of the HCC make it possible to achieve a higher coverage rate (94.5% with 9 positions) with fewer scanner positions than the human operator (95% with 12 positions). We attempted to achieve the maximum coverage rate (100%) with the HCC by selecting a small stop threshold (0.01m) as a termination condition. We notice that the number of HCC viewpoints increases significantly to reach 27 scanner positions.

Beyond this, our NBV planner can be used in online configuration with a TLS to assist the human operator and ensure the completeness of the scanned environment. Unfortunately, we were unable to perform any experiments in this configuration because of our TLS device (Leica C10) does not provide a real-time API.

However, the latest generation of TLS is now provides mobile applications that enable the user to track the scanner in real time. The integration of our approach into one of these applications is an interesting prospect, as long as the TLS manufacturers release the source code.



(a)



(b)

Figure 29: **(a)** Comparison between the HCC views (red) and the choice of the human operator (blue) inside the region A_0 that bounds the center of the esplanade. **(b)** The coverage rate inside the region A_0 after each scanner displacement.

4. CONCLUSIONS

In this paper, a Non-Model-Based exploration strategy was proposed for solving the view-planning problem. Our pipeline, the "Projective-Method + Hierarchical Conservative Cells (HCC)" allows visibility analysis and optimal scanner placement in the case of TLS and mobile scanner. The design of our approach was focused on two aspects. One is to have the ability to perform visibility analysis in large, complex environments. This objective is achieved with the Projective-Method with high accuracy without hardware acceleration. The second aspect is achieved by HCC by reducing the number of candidates for the next sensing position, based on the concept of conservative cells.

The strategy has been evaluated in both simulated and real environments. In the case of a mobile scanner embedded on a robot platform, the proposed approach provided efficient results in terms of coverage rate within an indoor environment (MIS laboratory). Furthermore, the comparison of our method with an experienced human operator in the case of a large (2000 m^2), complex outdoor environment (The esplanade of Hassan in Rabat) showed that the proposed pipeline provided similar coverage rate with fewer scanner positions.

In future work, we will add the overlap between scans as a criterion in the NBV selection. This overlap will guarantee the success of the registration process and will improve the quality of the scanned point cloud. This paper also presents a solution for the 2D view-planning problem. This was motivated by the previous work in the literature which focuses mainly on a 2D approach and confirmed by our experiments carried out using the mobile robot in the case of an indoor environment and also the TLS in the case of a large outdoor environment. Indeed, experimental results show that a unique slice in the 3D points cloud is sufficient to ensure accurate view-planning and a high coverage rate of the scanned environment. However, the extension to 3D view-planning in more complex environments and the problems mentioned above are expected to be solved in the future work.

References

[1] P. J. Besl, N. D. McKay, Method for registration of 3-d shapes, in: Sensor fusion IV: control paradigms and data structures, Vol. 1611, International Society for Optics and Photonics, 1992, pp. 586–606.

[2] A. Myronenko, X. Song, Point set registration: Coherent point drift, IEEE transactions on pattern analysis and machine intelligence 32 (12) (2010) 2262–2275.

[3] M. Magnusson, The three-dimensional normal-distributions transform: an efficient representation for registration, surface analysis, and loop detection, Ph.D. thesis, Örebro universitet (2009).

[4] J. O'rourke, et al., Art gallery theorems and algorithms, Vol. 57, Oxford University Press Oxford, 1987.

[5] J. Urrutia, Art gallery and illumination problems, in: Handbook of computational geometry, Elsevier, 2000, pp. 973–1027.

[6] P. Ashok, F. V. Fomin, S. Kolay, S. Saurabh, M. Zehavi, Exact algorithms for terrain guarding, ACM Transactions on Algorithms (TALG) 14 (2) (2018) 1–20.

[7] A. Aryan, F. Bosché, P. Tang, Planning for terrestrial laser scanning in construction: A review, Automation in Construction 125 (2021) 103551.

[8] S. Soudarissanane, R. Lindenbergh, Optimizing terrestrial laser scanning measurement set-up, in: ISPRS Workshop Laser Scanning 2011, Calgary, Canada, 29-31 August 2011; IAPRS, XXXVIII (5/W12), 2011, International Society for Photogrammetry and Remote Sensing (ISPRS), 2011, pp. 127–132.

[9] T. Rabbani, S. Dijkman, F. van den Heuvel, G. Vosselman, An integrated approach for modelling and global registration of point clouds, ISPRS journal of Photogrammetry and Remote Sensing 61 (6) (2007) 355–370.

[10] K. Kawashima, S. Yamanishi, S. Kanai, et al., Finding the next-best scanner position for as-built modeling of piping systems., International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences 45 (2014).

[11] L. Díaz-Vilariño, E. Frías, J. Balado, H. González-Jorge, Scan planning and route optimization for control of execution of as-designed bim., International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences (2018).

[12] H. K. Biswasa, F. Boschéa, M. Suna, Planning for scanning using building information models: A novel approach with occlusion handling, in: Symposium on Automation and Robotics in Construction and Mining (ISARC 2015), Vol. 15, 2015, p. 18.

[13] H. Son, C. Kim, C. Kim, Fully automated as-built 3d pipeline extraction method from laser-scanned data based on curvature computation, Journal of Computing in Civil Engineering 29 (4) (2015) B4014003.

[14] F. Jia, D. Lichti, An efficient, hierarchical viewpoint planning strategy for terrestrial laser scanner networks., ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences 4 (2) (2018).

[15] W. Boehler, M. B. Vicent, A. Marbs, et al., Investigating laser scanner accuracy, The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 34 (Part 5) (2003) 696–701.

[16] D. D. Lichti, Error modelling, calibration and analysis of an am–cw terrestrial laser scanner system, ISPRS journal of photogrammetry and remote sensing 61 (5) (2007) 307–324.

[17] M. Heidari Mozaffar, M. Varshosaz, Optimal placement of a terrestrial laser scanner with an emphasis on reducing occlusions, The Photogrammetric Record 31 (156) (2016) 374–393.

[18] D. Wujanz, F. Neitzel, Model based viewpoint planning for terrestrial laser scanning from an economic perspective., International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences 41 (2016).

[19] D. D. Lichti, A method to test differences between additional parameter sets with a case study in terrestrial laser scanner self-calibration stability analysis, ISPRS Journal of Photogrammetry and Remote Sensing 63 (2) (2008) 169–180.

[20] S. Soudarissanane, R. Lindenbergh, M. Menenti, P. Teunissen, Incidence angle influence on the quality of terrestrial laser scanning points, in: Proceedings ISPRS Workshop Laserscanning 2009, 1-2 Sept 2009, Paris, France, ISPRS, 2009, pp. 183–188.

[21] S. Soudarissanane, R. Lindenbergh, M. Menenti, P. Teunissen, Scanning geometry: Influencing factor on the quality of terrestrial laser scanning points, ISPRS journal of photogrammetry and remote sensing 66 (4) (2011) 389–399.

[22] R. Monica, J. Aleotti, Contour-based next-best view planning from point cloud segmentation of unknown objects, Autonomous Robots 42 (2) (2018) 443–458.

[23] R. Monica, J. Aleotti, Surfel-based next best view planning, IEEE Robotics and Automation Letters 3 (4) (2018) 3324–3331.

[24] W. Jing, K. Shimada, Model-based view planning for building inspection and surveillance using voxel dilation, medial objects, and random-key genetic algorithm, Journal of Computational Design and Engineering 5 (3) (2018) 337–347.

[25] F. Jia, D. D. Lichti, A model-based design system for terrestrial laser scanning networks in complex sites, Remote Sensing 11 (15) (2019) 1749.

[26] C. Wang, D. Zhu, T. Li, M. Q.-H. Meng, C. W. de Silva, Efficient autonomous robotic exploration with semantic road map in indoor environments, IEEE Robotics and Automation Letters 4 (3) (2019) 2989–2996.

[27] X. Zuo, F. Yang, Y. Liang, Z. Gang, F. Su, H. Zhu, L. Li, An improved autonomous exploration framework for indoor mobile robotics using reduced approximated generalized voronoi graphs, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences 1 (2020) 351–359.

[28] R. Nasir, A. Elnagar, et al., Gap navigation trees for discovering unknown environments, Intelligent Control and Automation 6 (04) (2015) 229.

[29] B. Tovar, L. Guilamo, S. M. LaValle, Gap navigation trees: Minimal representation for visibility-based tasks, in: Algorithmic Foundations of Robotics VI, Springer, 2004, pp. 425–440.

[30] L. Murphy, P. Newman, Using incomplete online metric maps for topological exploration with the gap navigation tree, in: 2008 IEEE International Conference on Robotics and Automation, IEEE, 2008, pp. 2792–2797.

[31] R. Lopez-Padilla, R. Murrieta-Cid, S. M. LaValle, Optimal gap navigation for a disc robot, in: Algorithmic Foundations of Robotics X, Springer, 2013, pp. 123–138.

[32] B. Yamauchi, A frontier-based approach for autonomous exploration, in: Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation', IEEE, 1997, pp. 146–151.

[33] H. H. González-Banos, J.-C. Latombe, Navigation strategies for exploring indoor environments, The International Journal of Robotics Research 21 (10-11) (2002) 829–848.

[34] A. Mobarhani, S. Nazari, A. H. Tamjidi, H. D. Taghirad, Histogram based frontier exploration, in: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2011, pp. 1128–1133.

[35] S. Osher, R. P. Fedkiw, Level set methods: an overview and some recent results, Journal of Computational physics 169 (2) (2001) 463–502.

[36] D. N. Subramani, P. F. Lermusiaux, Energy-optimal path planning by stochastic dynamically orthogonal level-set optimization, Ocean Modelling 100 (2016) 57–77.

[37] W. Gao, M. Booker, A. Adiwahono, M. Yuan, J. Wang, Y. W. Yun, An improved frontier-based approach for autonomous exploration, in: 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), IEEE, 2018, pp. 292–297.

[38] Y. Wang, R. Zhang, L. Qian, An improved a* algorithm based on hesitant fuzzy set theory for multi-criteria arctic route planning, Symmetry 10 (12) (2018) 765.

[39] L. R. Foulds, Graph theory applications, Springer Science & Business Media, 2012.

[40] C. Connolly, The determination of next best views, in: Proceedings. 1985 IEEE international conference on robotics and automation, Vol. 2, IEEE, 1985, pp. 432–435.

[41] J. I. Vásquez-Gómez, E. Löpez-Damian, L. E. Sucar, View planning for 3d object reconstruction, in: 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2009, pp. 4015–4020.

[42] C. Dornhege, A. Kleiner, A frontier-void-based approach for autonomous exploration in 3d, Advanced Robotics 27 (6) (2013) 459–468.

[43] F. Achakir, M. Deseilligny, S. Fkihi, M. Mghari, M. Ettarid, E. Mouaddib, A. Radgui, The hassan mosque at the digital era., Frontiers in Science and Engineering International Journal 7 (1) (2017) 45–58.