Highlights

Optimal Scan Planning For Surveying Large Sites with Static and Mobile Mapping Systems

Ernesto Frías, Mattia Previtali, Lucía Díaz-Vilariño, Marco Scaioni, Henrique Lorenzo

- a scan planning tool including optimal routing for both static and mobile scanning
- a mobile scanning planning strategy to determine key scan positions on the trajectory
- a balanced-clustering programming formulation for solving mobile scan planning
- a method for 3D data quality simulation including route soften based on Helios

Optimal Scan Planning For Surveying Large Sites with Static and Mobile Mapping Systems

Ernesto Frías[a], Mattia Previtali[b], Lucía Díaz-Vilariño[a,*], Marco Scaioni[b], Henrique Lorenzo[a]

[a] *Universidade de Vigo. CINTECX, Applied Geotechnologies Research Group, Campus universitario de Vigo, As Lagoas, Marcosende, Vigo, 36310, Spain*

[b] *Department of Architecture, Built Environment and Construction Engineering, Politecnico di Milano, Via Ponzio 31, Milano, 20133, Italy*

Abstract

Since the last two decades, the use of laser scanners for generating accurate and dense 3D models has been rapidly growing in multiple disciplines. The reliance on human-expertise to perform an efficient scanning in terms of completeness and quality encouraged the researchers to develop strategies for carrying out an optimized and automated scan planning. Nevertheless, due to the predominant use of static terrestrial laser scanners (TLS), the most of developed methods have been focused on scan optimization by fixing standpoints on basis of static scanning. The increasing use of portable mobile laser scanning systems (MLS) enables faster non-stop acquisition which demands appropriate approaches to optimise dynamic scan planning. Therefore, a novel method addressing the absence of dynamic scan planning is proposed considering specific MLS constraints such as maximum acquisition time or closed-loops requirement. First, an initial analysis is carried out to determinate key-positions to reach during data acquisition. From these positions a navigable graph is generated to compute routes satisfying specific MLS constraints by a three-step process. This starts by estimating the number of routes necessary to subsequently carry out a coarse graph partition based on Kmedoids clustering. Next, a balancing algorithm was implemented to compute a balanced graph partition by node exchanging. Finally, partitions are

*Corresponding author

Email address: lucia@uvigo.es (Lucía Díaz-Vilariño)

extended by adding key nodes from their adjacent ones in order to provide a desirable overlapping between scans. The method was tested by simulating three laser scanner configurations in four indoor and outdoor real case studies. The acquisition quality of the computed scan planning was evaluated in terms of 3D completeness and point cloud density with the simulator Helios++.

Keywords: 3D digitalization, autonomous navigation, computational geometry, path planning, scan-vs-BIM, spatial analysis

---

1. Introduction

With the development of reliable and rapid sensing methods for direct 3D point cloud acquisition, new opportunities for surveying and mapping complex sites have emerged. Starting in first early 2000s, static terrestrial laser scanning technology has developed and become a consolidate technique for 3D data acquisition (Watt and author, 2005; Alba et al., 2006; Buckley et al., 2008; Oppikofer et al., 2009; Abellan et al., 2014). These new opportunities have become more relevant with the progress of mobile mapping systems (MMSs), and more recently with the development of indoor mobile mapping systems (IMMSs) (Nocerino et al., 2017). Advances in the reduction of size and weight of laser scanning sensors, together with improvements in indoor positioning techniques such as the well-known simultaneous localization and mapping (SLAM), have contributed to the consolidation of new portable and mobile systems (Corso and Zakhor, 2013). MMSs are usually classified according to the platform in which sensors are placed: cart, backpack, UAV, and handheld. The variability of platforms where sensors are integrated gives an idea about how complex and challenging the sites to acquire may be (Otero et al., 2020).

Despite the usefulness of laser scanning systems, data capture is generally a time-consuming task especially when talking about terrestrial laser scanning (TLS). Minimizing the number of scanning operations while maximizing data completeness is essential for efficient data capture. In this context, the optimal positioning of TLS has been extensively studied in the last few years see, e.g., Ahn and Wohn (2015); Biswas (2020). On the contrary, the problem of scan planning for MMSs has not been addressed too much. This can be explained by the fact that acquiring data with MMS is much faster, and

acquisition time is no longer considered a problem except for the autonomy of the systems. Nevertheless, scan planning for MMSs becomes also necessary in the case of large and complex sites in a way to ensure data completeness, avoiding repetitiveness, and dealing with time restrictions. This need is even more essential for autonomous systems not following the Next Best View (NBV) problem, that is all case studies in which there is previous knowledge about the scene to capture, typically as 2D and 3D models. Applications such as construction control or scan-vs-BIM are among the most popular in recent times (Aryan et al., 2021).

Different to optimal scan planning for TLS, in which we typically define a set of minimum positions maximizing the coverage from which the laser scanners capturing data with a panoramic horizontal field of view, scan planning for MMSs requires from the definition of a complete trajectory. In this paper, we formulate the definition of a trajectory of acquisition as the definition of an optimal route passing through a set of key scan positions. Route planning has been widely addressed in the literature, and many related problems are known to be NP-hard (Kim et al., 2020). The complexity increases substantially with the size of the case study because more nodes conforming the navigable graph need to be considered in the analysis. In the same way, the coverage problem is becoming more complex when requirements from some MMS, such as the need of closed loops, is taken into account.

To overcome the above challenges, this paper describes a novel approach for the optimal scan planning for both static and mobile laser scanning systems. Previous developments focused on TLS scan planning (Frías et al., 2019) are taken as basis for the determination of the minimal scan positions from which the MMS needs to pass through. Then, the scan planning for MMS is formulated as an optimal-route planning problem in which main restriction is given by the maximum acquisition time for each specific sensor. In addition, the method also considers overlapping between laser scan trajectories for ensuring subsequent registration tasks. The method deals with polygons with holes and obstacles either from the perspective of the navigable graph calculation and for the visibility analysis.

To summarize, the main contributions of this paper are listed as follows:

- a scan planning tool including optimal routing, which is flexible in the sense that it can be applied to static and mobile scan planning, dealing with non-closed loop and closed-loop requirements;

- a mobile scan planning strategy based on optimal scan position placement followed by optimal navigable routing;
- a balanced graph partitioning programming formulation for solving mobile scan planning with closed-loops, which effectively find solutions in case of large-site acquisition. The method includes the addition of sharing nodes between adjacent clusters (loops) to ensure overlapping; and
- a method for 3D data quality simulation including route soften based on Helios, an open-source software developedby the Geoinformation Group at the University of Heidelbergh, Germany.

The remainder of this paper is organized as follows. Section 2 reviews the state of the art in terms of scan planning both for static and mobile mapping systems. A subsection about quality analysis of planned scan data is also included. Section 3 describes the proposed method while Section 4 analyses and discusses the results obtained from applying the method to several case studies, including indoor and outdoor scenarios. Finally, Section 5 is devoted to conclude this work and to address future development which will be necessary to go along with a continuously improving technology.

2. Related work

Scan planning problem is generally stated as a particular case of the extensively studied View Planning Problem (VPP) (Scott et al., 2003). Techniques to address this problem can be classified in two main categories depending on scene model availability. On the absence of this, non model-based methods are implemented by a Next Best View (NBV) planner designed to determinate the best scan position providing greater coverage gain (Connolly, 1985; Pito, 1999). Since no prior knowledge from the scene is available, heuristic function is often defined to decide the next scan position using only information obtained from previous scans. This online strategy is widely implemented for automatic scanning by autonomous mobile robots (Quintana et al., 2016) or Unmanned Aerial Systems (UASs) (González-de Santos et al., 2018).

On the contrary, model-based methods enable to perform a scan planning with complete scene information. Generally, the model consists of 2D layout

or is reduced to it from a 3D model to represent the scene as a polygon such that scan planning can be posed as the classical computational geometrical "art-gallery problem". This is formulated as the calculation of minimum number of positions inside the polygon needed to provide free-obstacle line-of-sight to any point composing the polygon. The NP-hard complexity of the problem (Lewis and Papadimitriou, 1997) leads to make assumptions to become good solutions. For example, González-Banos (2001) proposed a randomized strategy to reformulate the "art-gallery problem" as a "set-covering problem". In his work, candidate positions are randomly generated and polygon boundaries are dicretised on basis of the visibility from candidates. Then, the minimum covering problem is solved by applying Greedy strategy (Chvatal, 1979). This approach is also used by Blaer and Allen (2007) extending laser constraints for a initial coarse scanning embedded in a two-phase acquisition refined with a NBV scan planning.

A similar approach is adopted by Soudarissanane and Lindenbergh (2011) in which elements of interest are discretised in equal-length segments whereas a regular grid is used to generate candidates. Besides, laser constraints such as range and incidence angle are regarded in visibility analysis. In the following stage, the coverage optimisation is solved by Greedy selection again. A variant from this optimisation algorithm is proposed by Jia and Lichti (2019) coined Weighted Greedy Algorithm (WGA) that prioritises candidates covering segments not visible from other positions. Furthermore, the provided solution is exploited to remove redundant candidates hierarchically in order to carry out a reliable scan planning for large sites.

A different approach for a workable scanning of large scenes is proposed in Zhang et al. (2016). In addition to the aforementioned laser constrains, specific Level-of-Detail (LOD) requirement are included in the coverage optimisation involving an increase of computational complexity. To reduce this, points of interest in the elements to scan are grouped in compliance with required LOD for each point. After, the resultant clusters are arranged minimising geometrical scattering and coverage problem is solved separately per each cluster.

A more recent work presented by Kim et al. (2020) generalises sensor constraints to realise an efficient space representation oriented to a variety of indoor applications beyond scan planning. The polygon representing the scene is split into triangles which are then grouped according to the tar-

get application, so that the expanded groups may fulfil required constraints. Then, coverage optimisation (also the partitioning problem) is posed as a binary linear programming problem (Hu, 2016). In addition, the space representation is compatible with the well-known standard IndoorGML (Kang and Li, 2017). Previous set-covering problem formulations are extended to a mixed-integer programming problem in Dehbi et al. (2021) by adding non-discrete variables to address overlapping constraint for further point cloud registration. To do this, a graph is created representing candidate positions as nodes which are connected according visibility. Furthermore, coverage is not focused on vertical elements but also floor area acquisition is explicitly managed.

Despite the numerous works addressing scan planning, almost all of them are limited to static acquisition assuming that data is collected only from the selected scan positions as is the case TLS is adopted. In addition, scanning complex environments with TLS may become a tedious task because of the inaccessibility for mobile robots or craggy surfaces. A real situation is reported in (Rüther and Palumbo, 2012), where the objective of this study was to scan with TLS an extend heritage site of 700 x 400 m of extension. The 1200 scans collected to cover the entire scene required six weeks and its subsequent registration about four months. In contrast, Zlot et al. (2014) has evaluated the advantages of dynamic scan by a handheld laser scanner performing a scanning of an heritage site with an extension of approximately 400 x 250 m. The scanning time was about 3.6 hours while the processing time from raw data to provide a coherent 3D model was 2.6 hours. Despite the scanned area is almost three times smaller than the site scanned with TLS, the consumed time for both data capture and post-processing is drastically reduced by operating a dynamic acquisition with a mobile laser scanning (MLS). However, as the scan planning planning of the MLS was driven by human-expertise, some aspects about the redundant data capturing, minimum number of scans and time optimisation per scan were missed.

Actually, there is a shortage in the literature about dynamic scan planning requiring to deal with specific constraints of MLS devices such as tracing closed trajectories and maximum acquisition time per scan. Therefore, this work is focused to address this gap by extending previous static scan planning proposed by Frías et al. (2019) to dynamic acquisition in order to derive efficient trajectories in terms of time-saving, coverage and overlapping.

3. Method

A general overview of the method is represented in Fig. 1. Firstly, an initial analysis for determining the minimal scan positions maximizing coverage is performed. This process consists of a set of steps (space partitioning, visibility analysis and scan optimization) and it considers a panoramic horizontal field of view. The acquisition from these positions guarantees the minimal coverage required, therefore, route planning starts by generating a grid-based navigable graph ensuring that all initial scanning positions are reached. Then, an optimal route is computed using a heuristic ant "colony optimisation algorithm" (ACO). For TLS, only one route is sufficient to complete an acquisition because it does not have time restrictions. However, MLS planning may require several routes, especially on large environments. In this case, navigable graph is partitioned into smaller subgraphs taken into account MLS limitations, such as maximum acquisition time or closed route requirement. The graph division is addressed as a balanced graph partitioning problem deriving cluster subgraphs from which optimal routes are computed by ACO algorithm. Each procedure is extensively explained in the following subsections.

3.1. Initial analysis

Despite the optimisation of the minimal scan positions is similar to Frías et al. (2019), a brief summary of the processes conforming the initial analysis is included for contextualizing the route planning problem.

The input data consists of a 2D CAD file structured in layers, each one corresponding to an element type such as floor, walls or columns. In accordance with user selection, layers are classified into $\mathcal{L}_{acq}$ and $\mathcal{L}_{occ}$ groups containing element layers to be acquired and elements that cause occlusions in laser visibility, respectively. Also, a predefined layer called 'floor' or 'extension' represents the navigable space which is defined as the floor area where the acquisition system can be positioned by itself. Further, this layer is represented geometrically as a closed polygon $\mathcal{P}$. Then, elements of $\mathcal{L}_{acq}$ and $\mathcal{L}_{occ}$ are discretized so that their geometries are represented by evenly spaced points composing the sets $\mathcal{X}_{acq}$ and $\mathcal{X}_{occ}$.

In order to carry out a simulated scanning acquisition as realistic as possible, laser characteristics involving acquisition constraints are included in all

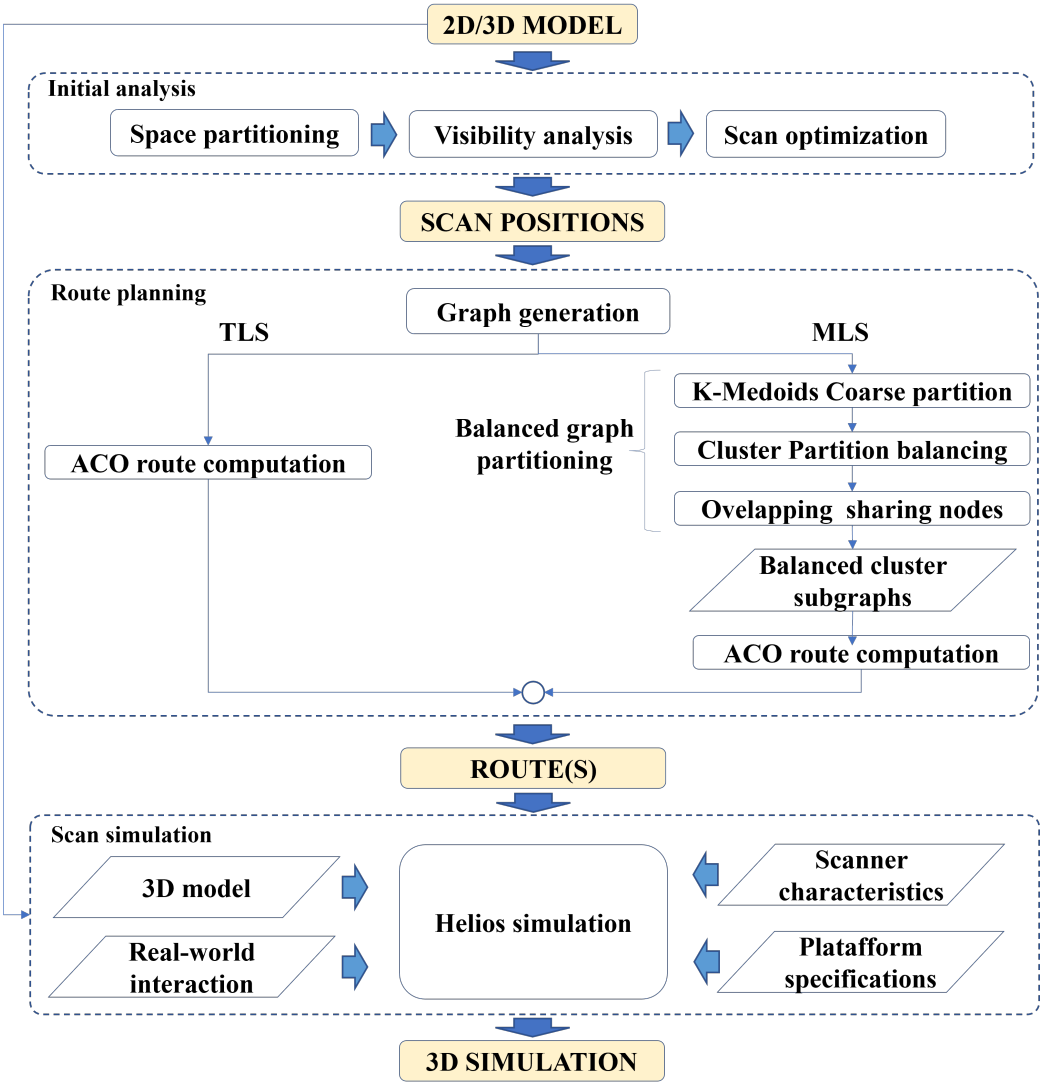


Figure 1: General workflow of the method.

procedures composing the method. Since this study addresses two acquisitions modes with different limitations, scanner parameters are divided into two categories: common and specific constraints collected in Table 1.

Table 1: Laser constraints classified in common and specific.

	Constraints	Static acquisition (TLS)	Dynamic acquisition (MLS)
Common	Laser range	3D	2D
	Minimum distance	Yes	Yes
	Visibility	Yes	Yes
Specific	Maximum acquisition time	No	Yes
	Closed-loops requirement	No	Yes/No
	Overlapping	-	User-defined

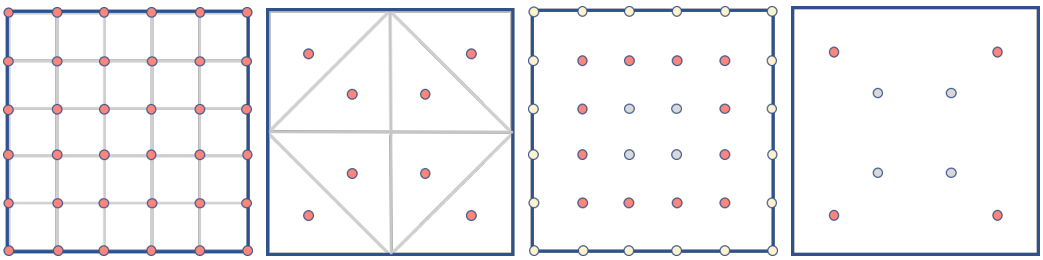
3.1.1. Space partitioning

After elements have been discretised and classified, the geometry of the navigable space is partitioned into smaller regions for a more efficient and manageable representation. Although multiple partitions are possible, a space partition can be generically defined as:

Definition 1. *(Space partition)*
Given a polygon $\mathcal{P}$ representing the navigable space, a set of the regions $\mathcal{R} \subset \mathcal{P}$ is a partition of $\mathcal{P}$ if given any region $r_i, r_j \in \mathcal{R}$ such that $\{r_i \cap r_j, \forall_{i,j} \mid i \neq j\} = \varnothing\}$ and $\{r_i \cup \ r_j, \forall_{i,j} \mid i \neq j\} = \mathcal{P}$.

For the initial analysis, two partitioning methods based on grid or triangulation distribution can be selected to generate a discrete number of potential positions $\mathcal{S}_{ini}$ to be scanning places. Unlike most of scan planning works, the boundary of $\mathcal{P}$ may not be the area of interest to be acquired, moreover, it may not be represented as a closed polygon as is the case of construction environments where outer walls have not already been built or as in open outdoor environments.

The set $\mathcal{S}_{ini}$ resultant from space partitioning (Figs. 2a and 2b) are filtered by laser range and minimum distance constraints before the visibility analysis. To do this, the positions from which no point of $\mathcal{X}_{acq}$ would be acquired due to range constraints and positions at a distance less than the minimum distance to occlusion elements are discarded. The filtering outcome is a set of candidate positions $\mathcal{S}_{cand}$ (Figs 2c and 2d) to be analysed in the next step.



(a) Grid-based partition (b) Triangulation-based partition (c) Constrained grid-based partition (d) Constrained triangulation-based partition

Figure 2: Points (red) representing navigable space generated by a) grid-based and b) triangulation-based methods. Candidate points (red) are depicted in c) and d) after filtering on basis of range (blue points) and minimum distance (light gold) constraints.

3.1.2. Visibility analysis

In this step, the coverage provided by each candidate position $p_{cand} \in S_{cand}$ is evaluated by a visibility analysis based on a ray-tracing strategy (Díaz-Vilariño et al., 2019) as shown in Fig. 3. Prior to this analysis, an occupancy binary map is generated projecting occlusion elements $\mathcal{X}_{occ}$ on it. Then, visible points $\mathcal{X}_{vis}$ from each candidate are determined by simulating laser beams on the map from p_{cand} to elements of interest $\mathcal{X}_{acq}$.

Generally, ray-tracing-based visibility analysis is a time-consuming process due to the larger number of evaluations from candidates to target points that have to be performed. Therefore, only points to be acquired $\mathcal{X}_{acq}$ falling in covering range are used for beam simulations from each candidate. In addition, auxiliary layers can be included in the input 2D model to lead a more efficient analysis. These should correspond to subspaces of the navigable space, i.e., rooms and corridors in indoors. In this case, only visibility from points to be acquired inside the subspace is checked.

3.1.3. Scan optimization

Finally, the selection of the best scanning positions is formulated as a minimum coverage problem. As already implemented in Frías et al. (2019), the candidate position covering unseen greater area to acquire is selected (maximum coverage) by a combinatorial iterative process as shown in Fig. 4. The implemented algorithm based on a 'backtracking' strategy stops when the minimum required completeness Cov_{min} is achieved. The selected po-

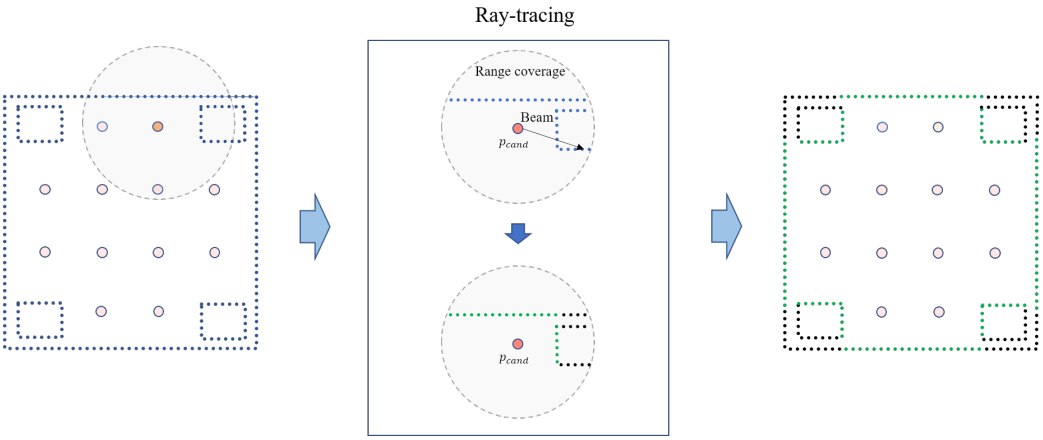


Figure 3: Visibility analysis process determines which points to be acquired (blue) would be reached by the simulated laser beam (green) from any candidate position (red).

sitions $\mathcal{S}$ are the minimum positions necessary to accomplish the required coverage in terms of completeness (minimum coverage).

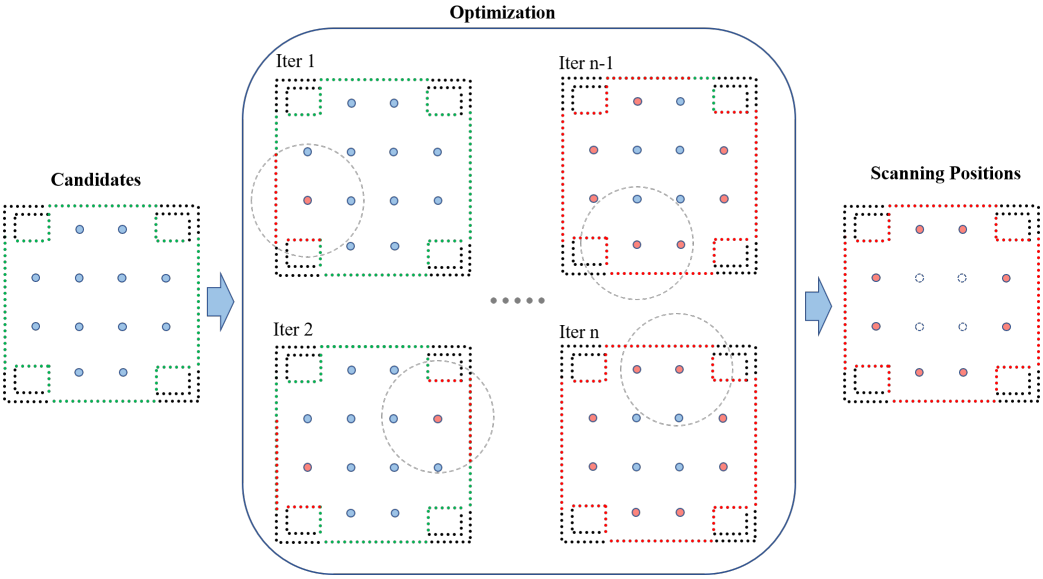


Figure 4: From candidates (blue points) resulting after visibility analysis, scan optimization procedure selects the candidate providing the most coverage (at the centre of dashed circle) in each iteration considering only visible points (green).

In summary, the initial analysis provides a set $\mathcal{S}$ containing the theoret-

ically optimal scan placements to acquire the points of interest $\mathcal{X}_{acq}$. The coverage/completeness provided by the selected locations satisfies scanning requirements. Moreover, the panoramic Field-of-View (FoV) considered for the analysis assumes all points of interest falling in a sphere (3D) centred at each scanner placement are visible. However, acquisition area of most iMMS consists in a 2D plane orthogonal to the system motion. Despite this difference, the obtained positions used in the further analysis performed for dynamic acquisition are valid since similar 3D coverage is obtained by making a U-turn at a scan position with a MLS system. Besides, the implemented procedure achieves to compute high quality placements with a reasonable computational effort avoiding the higher complexity entailed for a 2D acquisition.

3.2. Route planning

Although the optimal scanning positions calculated in the initial analysis contribute to reduce acquisition time, addressing route planning problem can minimize the route distance to reach all positions. This is more critical to carry out an efficient acquisition plan in large complex environments acquisition with a iMMS. Therefore, remaining part of the proposed method deals with route computation taking into account common and specific laser constraints. Route planning for static acquisitions is implemented as in Frías et al. (2019) but the novelty of this work lies in route computation for acquisition based on MLS.

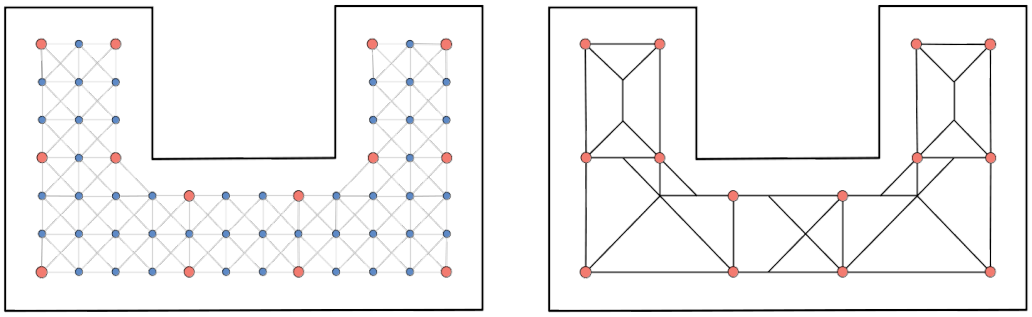
Route planning is divided into the three processing steps explained below. The first one consists in a graph generation to obtain routes for both laser operation modes. The second step corresponds to route computation for static acquisition with TLS while dynamic route planning is addressed in the latter process.

3.2.1. Graph generation

The most common representation of the navigable space for route computation is by means of a graph, therefore, route planning starts by generating a navigable graph. Since the method is geared to autonomous mobile robots, square grid was selected as graph nodes distribution. With regard the foregoing, a navigable graph can be formally defined as:

Definition 2. *(Navigable graph)*
A navigable graph is a simple undirected connected graph $G_n = \{N, E\}$ where N is the set of graph nodes and E the set of collision-free edges connecting graph nodes.

Furthermore, nodes composing N are categorized into two classes: scan nodes N_s corresponding to nodes located at scan locations from S and navigable nodes N_n that the mobile system can reach while travelling through the graph edges. An example of a G_n is depicted in Fig. 5a representing scan and navigable nodes in red and blue respectively.



(a) Navigable graph (b) Scanning graph

Figure 5: Graphical representation of a) a navigable graph and b) its corresponding high-level representation as a scanning graph.

As nodes N_s are the key-positions to route computation, a simpler higher hierarchy graph representation is made by abstracting navigable nodes. We call this high-level graph representation *scanning graph* (Fig. 5b) and it is defined as:

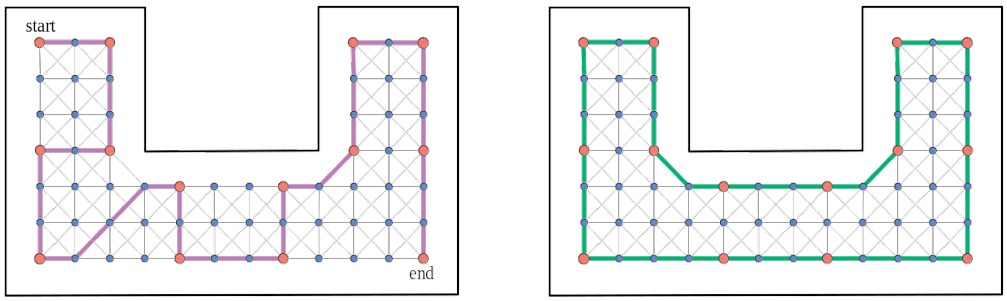
Definition 3. *(Scanning graph)*
Given a navigable graph $G_n = \{N, E\}$, a scanning graph is a complete graph $G_S = \{N_s, E_s\}$ where N_s is the set of all scan nodes contained in N and $E_s \subset E$ the set of collision-free edges connecting nodes.

The scanning graph is generated from G_n computing shortest path between each pair of nodes $(u_s, v_s) \in N_s$ using Dijkstra algorithm (Dijkstra, 1959). The obtained paths constitute the set of edges E_s representing the shortest route between scan nodes. Thus, only the relevant information to route computation is kept in the scanning graph.

3.2.2. Route computation for static acquisition

An optimal route should reach all scan locations travelling as short a distance as possible. Hence, the route computation can be formulated as the well-known *Travelling Salesman Problem (TSP)* defined as the search for the shortest route reaching all key-positions only once. The NP-hard complexity of this problem makes its applicability unfeasible in large environments where the number of scan positions would be high. This limitation is generally overcome by using available heuristic algorithms providing good results as the ACO selected for this work.

For a static acquisition with TLS, the optimal route is obtained by directly applying a ACO algorithm to the whole scanning subgraph. This is possible because no time restrictions have been considered for this acquisition mode, assuming that the scan planning can be completed with only one route. Fig. 6a depicts an optimal route passing through all of scan positions while Fig. 6b shows another route for the same graph finishing at the start position (closed route).



(a) No-closed optimal route

(b) Closed optimal route

Figure 6: Optimal routes

3.2.3. Route computation for dynamic acquisition

Unlike TLS-based systems, dynamic acquisition is generally performed with Mobile Mapping Systems (MMSs) based on SLAM that may cause positioning drift in large trajectories decreasing acquisition accuracy. To avoid this degradation, manufacturers recommend to limit the scanning time to a certain temporal value t_{max} or tracing closed trajectories during the acquisition. The temporal restriction together with the velocity of the mobile

system v_{sys} fix the maximum acquisition distance d_{max} computed by Eq. 1:

$$d_{max} = t_{max} * v_{sys} \tag{1}$$

Especially in large environments, the recommended maximum acquisition time leads to require multiple scans to complete the scan planning involving a route computation per each scan. To address this problem, a balanced-clustering algorithm is proposed to accomplish laser requirements while minimising the number of scans. The algorithm consists of three steps starting by a rough initial graph partitioning that is refined in the next step by a balancing process. Finally, clusters are extended to provide enough overlap between scans to facilitate the further registration of point clouds derived from each scan.

3.2.3.1. K-medoids coarse partition.

In this step, clustering K-medoids (Park and Jun, 2009) algorithm is used for an initial coarse grouping of scan nodes. As the popular K-means method (Krishna and Narasimha Murty, 1999), the number of goal clusters and the distance between key-points are the required input parameters. Unlike this, K-medoids can minimize other magnitudes beyond Euclidean distance. This property makes K-medoids more suitable for graph distance minimization. Hence, the distance matrix from scanning graph D_s is one input used as the dissimilarity measure to be minimised while the input number of clusters k must be estimated. The proposed solution for this problem consists in estimating the average number of scan nodes forming a cluster n_{npc} complying laser constraints. The estimation of n_{npc} is obtained by Eq. 2, which takes into account d_{max} and the overlapping fixed by an arbitrary number of scan nodes to be shared between clusters n_{sh}:

$$n_{npc} = \left\lfloor \frac{d_{max}/d_{avg}}{2} - n_{sh} \right\rfloor \tag{2}$$

Besides, the average distance between clusters d_{avg} is necessary to determinate n_{npc} but this not can be computed directly. Since all nodes composing a scanning graph are connected, the average distance between them cannot be taken as a reliable measure. This is because the value obtained would increase according to the graph´s length. To find a more robust value, an

adjacency relation between the scan nodes is established on the basis of graph connectivity. Thus, only adjacent nodes are used to compute the d_{avg} with respect to each other.

To determinate adjacency between scan nodes, the navigable graph is split into smaller subgraphs hereinafter referred to as *scanning subgraphs*. Before defining a *scanning subgraph*, the concept of induced subgraph in graph theory is introduced:

Definition 4. *(Induced subgraph)*
Given a graph $G_n = \{N, E\}$, let be Y a subset of nodes such as $Y \subset N$. The induced subgraph $G_n[Y]$ is a graph composed of nodes from Y and the edges from E whose both nodes belong to Y.

Relying on the above definition, a *scanning subgraph* can be defined as follows:

Definition 5. *(Scanning subgraph)*
Given a navigable graph $G_n = \{N, E\}$, let be N_s the set of scan nodes on the graph and $G_{SS} = \{N_{SS}, E_{SS}\}$ the subgraph induced by $N_{SS} \subset N$. G_{SS} is a *scanning subgraph* if the subset N_{SS} contains only one scan node $n_s^i \in N_s$ and the navigable nodes closer to n_s^i than to any other scan node.

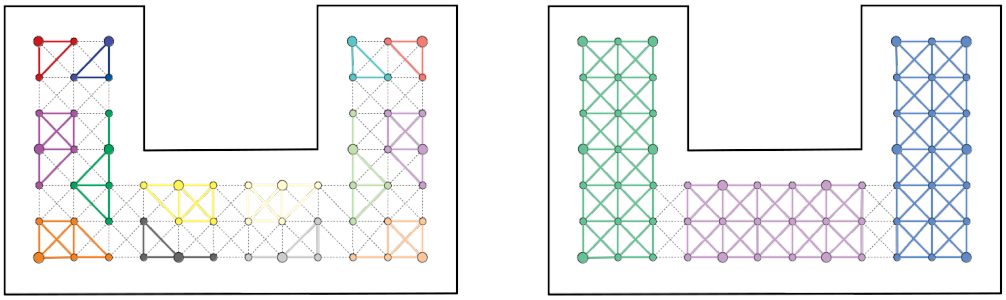
Definition 6. *(graph partition)*
Given a navigable graph $G_n = \{N, E\}$, $N_p = N_0, ..., N_k$ is a partition of G_n if $\bigcup_{i=0}^{k-1} N_i = N$ and $N_i \cap N_j = \emptyset, \forall_{i,j}, i \neq j$.

Hence, a navigable graph can be partitioned into $|N_s|$ scanning subgraphs deriving into a *scanning partition* formally defined as:

Definition 7. *(scanning partition)*
Given a navigable graph $G_n = \{N, E\}$, let be $N_p = N_0, ..., N_k$ a partition of G_n, N_p is a scanning partition of G_n if the induced subgraph $G_{SS}^i = G_n[N_i]$ is a scanning subgraph $\forall i = 0, ..., k$.

An example of a scanning partition is shown in Fig. 7a where each group of nodes is depicted by the same colour. Note that subgraphs are connected

to each other by edges of G_n not belonging to any scanning subgraph (black dashes edges). In this paper, these edges will be referred to as *adjacent scanning edges* since they determinate the adjacency relation between scanning subgraphs. Consequently, two scan nodes $n_s^i, n_s^j \in N_s$ are adjacent if their induced scanning subgraphs G_{SS}^i and G_{SS}^j are connected in the scanning partition by at least one adjacent scanning edge.



(a) Scanning partition.

(b) Cluster partition.

Figure 7: Graph partitioning from a) scan nodes and b) clusters.

After the mean distance of each scan node with respect to their adjacent ones is obtained and d_{avg} is computed by averaging all these distances the initial number of clusters k_{ini} is determined by Eq. 3. As conservative estimations have been assumed, the number of clusters is refined by the Algorithm 1 in order to optimise its value. In the first iteration, initial k_{ini} clusters are obtained by K-medoids. Then, the route distance of each cluster is estimated using the Greedy approach considering the closed-loops constraint. The mean of length of routes l_{avg} and the shortest route are evaluated to decide whether k is optimal or it has to be increased or decremented. This process is repeated until k is optimal. In addition, the k value used at each iteration is saved in a buffer k_{buffer} to avoid oscillations since evaluation conditions are not mutually exclusive:

$$k_{ini} = \left\lceil |S| \Big/ n_{npc} \right\rceil \tag{3}$$

Algorithm 1 Optimal k computation

```
procedure K_COMPUTATION(k_ini, D_s, d_max)
    k_c ← False
    k ← k_ini
    K_buffer ← k
    while k_c = False do
        C_ini ← Kmedoids(k, D_s)
        for c_i ← c_ini do
            D_est ← Greedy_estimation(c_i)
        l_avg = mean(D_est)
        if l_avg > d_max then
            k ← k + 1
            k_buffer ← k
        else
            if (l_avg + min(D_est)) < d_max and not (k − 1) in K_buffer then
                k ← k − 1
                K_buffer ← k
            else
                k_opt ← k
                k_c ← True
    Output: k_opt, C_ini
```

Since clustering is only based on the distance between scan nodes, initial clustering C_{ini} may be unbalanced such that the estimated route distance for some clusters can exceed d_{max}. In order to comply with laser requirements, next step consists in a balancing process that seeks to compensate the cluster distance difference through node exchange between adjacent clusters.

Like in the case of scanning subgraphs, adjacency between clusters is determined by the navigable graph partitioning (Fig. 7b). The sets derived from the partitioning correspond to nodes inducing to *clustersubgraphs* defined as follows:

Definition 8. *(Cluster subgraph)*
Given a navigable graph $G_n = \{N, E\}$, let be N_s the set of scan nodes on the graph and $N_c \subset N_s$ a set of scan nodes composing a cluster, the subgraph $G_{CS} = \{N_{CS}, E_{CS}\}$ is a *cluster subgraph* if:

1. N_{CS} is the set of nodes from N closer to N_c than to scan nodes not belonging to N_c.
2. G_{CS} is the induced subgraph $G_n[N_{CS}]$.

Therefore, the navigable graph can be partitioned into k node sets inducing a k cluster subgraphs. The obtained graph is refereed as *cluster partition* that is defined as:

Definition 9. *(Cluster partition)*
Given a navigable graph $G_n = \{N, E\}$, let be $N_p = N_0, ..., N_k$ a partition of G_n, N_p is a scanning partition of G_n if the induced subgraph $G^i_{SS} = G_n[N_i]$ is a cluster subgraph $\forall i = 0, ..., k$.

According to the previous definition, the relation between scan nodes composing a cluster C_i and the *cluster subgraph* G^i_{CS} induced from them is unambiguous. Like scanning partition, edges not belonging to any *cluster subgraph* are used to determinate adjacency relations between clusters. These edges are referred to as *adjacent cluster edges*.

Besides, since *cluster subgraph* connectivity is required in the balancing process to exchange nodes between adjacent clusters, the cluster subgraphs derived from the cluster graph partitioning must be connected. However, the induced subgraphs from clusters obtained by K-medoids clustering may not satisfy the connectivity condition. Therefore, before starting the balancing procedure, cluster subgraphs are analysed to meet connectivity requirement by moving scanning subgraphs between clusters, if necessary. The resultant clusters C_{con} are the input for the balancing process.

3.2.3.2. Cluster partition balancing.

The implemented balancing process consists in exchanging scan nodes between clusters for the purpose of making all cluster routes shorter than d_{max}, for forcing the distance variance from them be as small as possible. Main operations performed by the algorithm to attain a balanced cluster partition from the coarse clustering are visualised in Fig. 8.

First, the initial clustering graph is generated by partitioning the navigable graph according to the coarse clustering C_{con} computed in the previous step. At each iteration, the length of every *cluster subgraph* is estimated by Greedy-based route computation. Also, the clustering distance variance is calculated for the refinement of the balancing.

From the previous estimations, two clusters are selected to perform the node(s) exchange between them. The criteria for this selection depends on

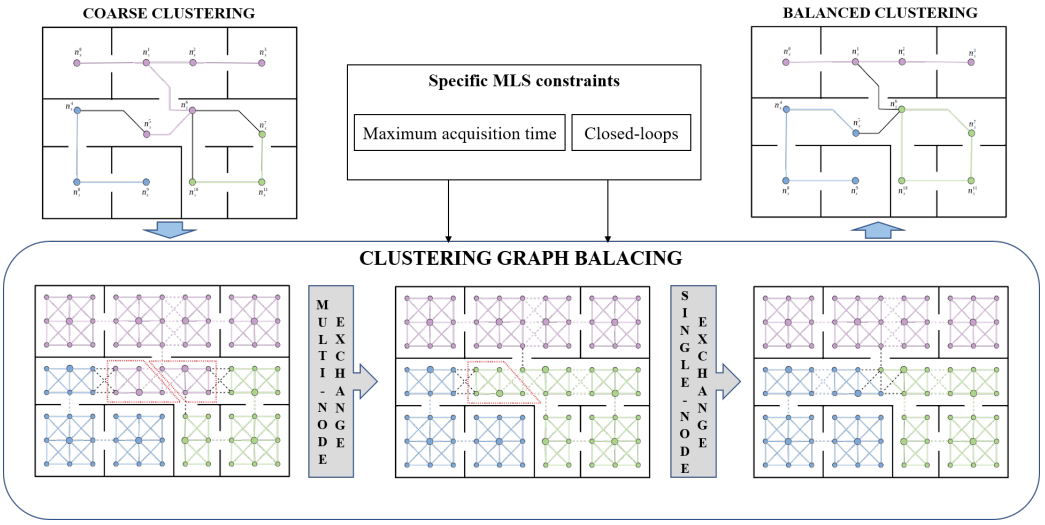


Figure 8: General clustering graph balancing process.

whether the estimated length of any cluster exceeds the maximum acquisition distance d_{max} or not. In the first case, the longest cluster subgraph is chosen as the cluster to be reduced C_{rdc}. Subsequently, the shorter cluster subgraph adjacent to C_{rdc} is the cluster selected to be expanded C_{exp}. Otherwise, the algorithm seeks to improve the balancing by clustering variance reduction. To do this, cluster selection is based on the length deviation of the clusters around the clustering average distance. First, the most deviated cluster c_d is selected in agreement with Eq. 4:

$$c_d \mid d = \max_d |D_d| \tag{4}$$

Then, Eq. 5 is used to determinate the cluster whose deviation better compensate that of c_d. Once both clusters have been determined, the cluster with the smallest deviation value is selected as the one to be expanded (C_{exp}) while the other one is the one to be reduced (C_{rdc}). In the example visualised in Fig. 9, the cluster c_2 is the most deviated cluster and it is selected as C_{exp} because D_2 is smaller than D_0:

$$c_o \mid o = \min_o |D_o + D_d| \tag{5}$$

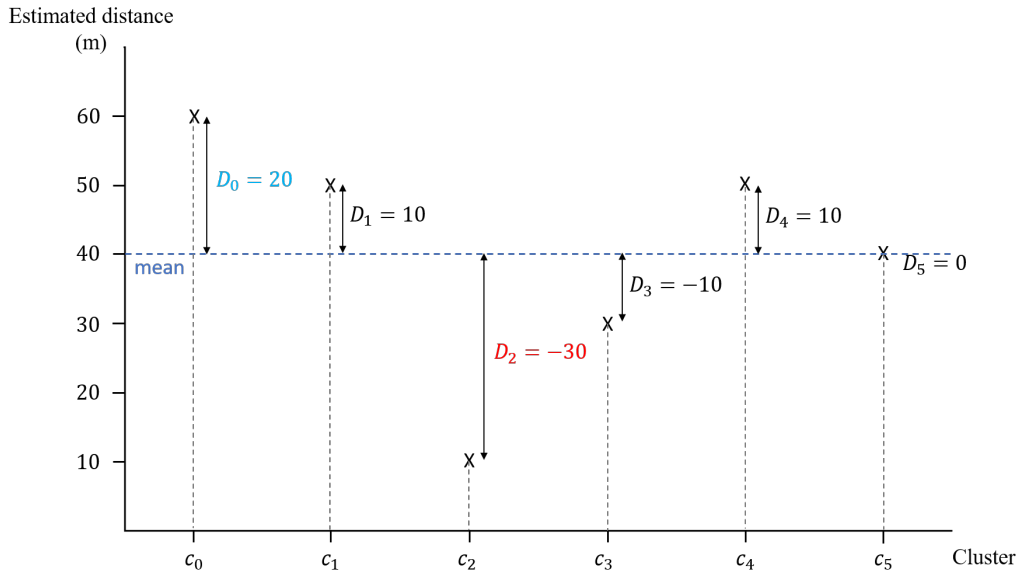


Figure 9: Graphical representation of cluster deviations around the mean distance. The largest deviation is visualized in red while the deviation that most offsets it is coloured in light blue.

Next step consists in determining the node(s) to move from C_{rdc} to C_{exp} regardless of the criteria previously used to select both clusters. Balancing process is based on nodes' connectivity, thus, all subgraphs composing C_{rdc} adjacent to C_{exp} are candidates to be exchanged. Following the example in Fig. 10a, let us suppose that C_1 (purple nodes) has been selected as C_{rdc} and C_2 (green nodes) as C_{exp}. Thus, the nodes corresponding to subgraphs G^5_{ss}, G^6_{ss}, G^{11}_{ss} are the candidates to exchange for being adjacent to C_2. Since multiple exchanges are possible, a criteria of prioritisation is defined with the aim of conducting the most advantageous exchange. Therefore, candidate nodes are evaluated in connectivity terms to establish node priority. Due to the fact that C_{exp} is adjacent to any candidate, the resultant expanded cluster will always be connected after the exchange. However, this is not true for the reduced cluster. Continuing with the example in Fig. 10a, the exchange of one subgraph such as G^5_{ss} or G^{11}_{ss} to C_2 does not break the connectivity of C_{rdc}. In contrast, C_{rdc} is split into three connected components if G^6_{ss} is exchanged generating a cluster graph partition of $k+1$ sets. To avoid this inconsistency, the candidate node associate to G^6_{ss} is discarded. After, the priority of the remaining candidates is determined according the number

of adjacent cluster edges (dashed black edges) joining candidate subgraph with C_{exp}. Thus, the subgraph G^5_{ss} have a level of priority 5 while that for subgraph G^{11}_{ss} is 1 being priority 1 the highest level.

The break in connectivity of C_{rdc} is an usual situation in graphs of indoor spaces since topological connections between corridors and rooms occur through by a door. Therefore, a more complex node exchange is addressed to provide greater flexibility to the algorithm. To visualise the problem, a new clustering graph represented in Figure 10b that corresponds to the previous example adding a slightly modification is used. In this case, subgraphs G^5_{ss} and G^6_{ss} are the adjacent candidate nodes to be exchanged. As in the example above, G^5_{ss} has priority 5 while the exchange of G^6_{ss} breaks continuity of C_{rdc}. But unlike the previous case, this exchange breaks the subgraph into two connected components only. In this case, a solution to allow the exchange of nodes that break cluster graph connectivity keeping cluster partition consistency is more feasible to implement. To solve clustering consistency, a new operation called *multi-node exchange* has been added. This exchange consists in moving the shorter component ($\{G^5_{ss}\}$) together with the candidate subgraph (G^6_{ss}) to C_{exp} which ensures connectivity of resultant subgraphs. Although this exchange is possible, the candidate G^5_{ss} is priority because this can be moved by *single-node operation*, that is, only one subgraph is moved.

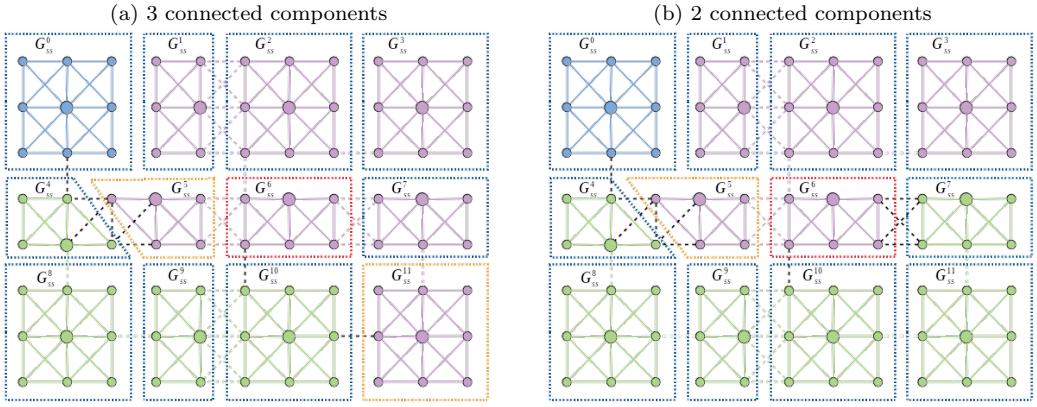


Figure 10: Unconnected cluster subgraph (purple) after moving one scanning subgraph (G^6_{ss}) generating a subgraph composed by a) 3 and b) 2 connected components.

Then, a new clustering is obtained after exchanging the highest-priority node(s) but it must be checked in order to avoid exchanges leading to oscillations in the balancing process. The clustering properties to evaluate

depend on the criteria used for cluster selection. The length of the resulting expanded cluster is computed from the comparison with the previous length of C_{rdc} if any cluster exceeded d_{max} in the previous clustering. In this case, the exchange is valid if length of C_{exp} is smaller. Otherwise, when clusters are selected on the basis of variance reduction, the exchange is valid if the variance of the new clustering is smaller than the previous variance and the length of C_{exp} does not exceed d_{max}. According to this check, the new clustering is ruled out if the exchange is not valid. Consequently, the remaining candidates are chosen according to priority until a valid exchange is performed.

In complex cases, exchange validation may be invalid for all candidates. To avoid the non-convergence of the algorithm in this situation, the node exchange between C_{rdc} and C_{exp} is blocked until one of them is modified in a further iteration. Thus, the balancing process concludes when any exchange is allowed.

3.2.3.3. Overlapping sharing nodes

Once scan nodes have been clustered, a route from each cluster can be computed separately with the expectation that the routes obtained comply with the specific MLS constraints. However, the overlapping between individual point clouds collected from each route may not be enough for further registration. To ensure this, clusters are extended by adding nodes from their adjacent clusters. The level of overlapping is determined by the user-selected parameter n_{sh}, which corresponds to the number of nodes that clusters must share with each other. Hence, from clusters do not sharing n_{sh}, the shortest one is selected to expand it with the closest node from an adjacency cluster and this process is repeated until all clusters share at less n_{sh}. The routes obtained after to apply this to the previous example with $n_{sh} = 1$ are depicted in Fig. 11b.

3.3. MLS visibility analysis

In order to make a coverage/completeness comparison between static and dynamic acquisition, a new visibility analysis is carried out with dynamic constraints. First, trajectory positions for laser beam simulation are computed on the basis of binary map obtained by projecting trajectory segments as shown Fig. 12a. The occupied cells by the projected trajectory are the source

(a) Routes without overlapping. (b) Routes with overlapping

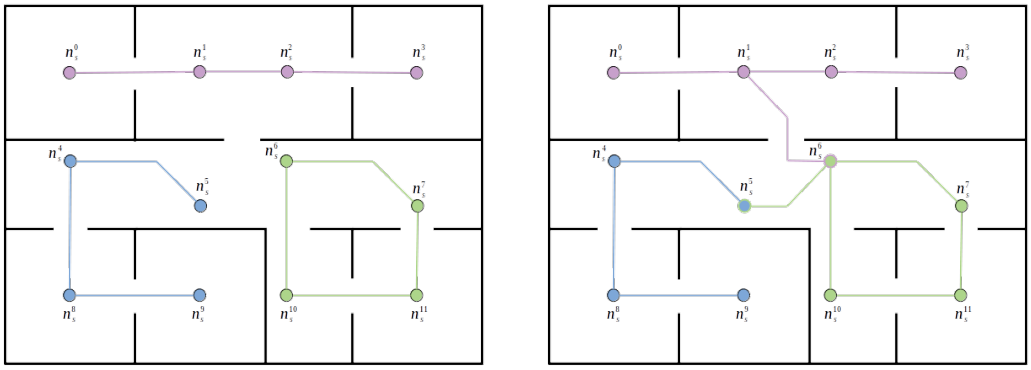


Figure 11: Routes computed a) directly from balanced clustering and b) after adding shared nodes

positions to perform the ray-tracing test. Second, the FoV is adapted to 2D laser range of MLS that limits the reachable points from source positions. Thus, the coverage area covered from a trajectory position corresponds to the orthogonal plane to the direction of motion. In addition, the yaw effect, inherent to mobile system movement is simulated by an angular sweeping of the coverage plane considering an offset of $\pm\Psi$ degrees. Abrupt changes in the direction along the trajectory generating gain in coverage are regarded as well by extending the plane sweeping with the angle Θ as is represented in Fig. 12b.

3.4. Scan simulation

The methodology developed for route planning is considering as searching space for the estimation of the optimal path a 2D domain. To evaluate the possible effects of this assumption in terms of scan quality, density and completeness in realistic 3D applications a scan simulation has been carried out for the four case study analysed.

Scan simulation is carried out by using the HELIOS++ library (Winiwarter et al., 2021). The Heidelberg LiDAR Operations Simulator (HELIOS++) is a laser scanning simulation framework originally implemented in C++ and allowing bindings with Python by using pyhelios extension. The HELIOS++ framework allows the virtual laser scanning simulations of LiDAR sensors placed on different platforms. For scan simulation five main

(a) Grid space (b) Euclidean space

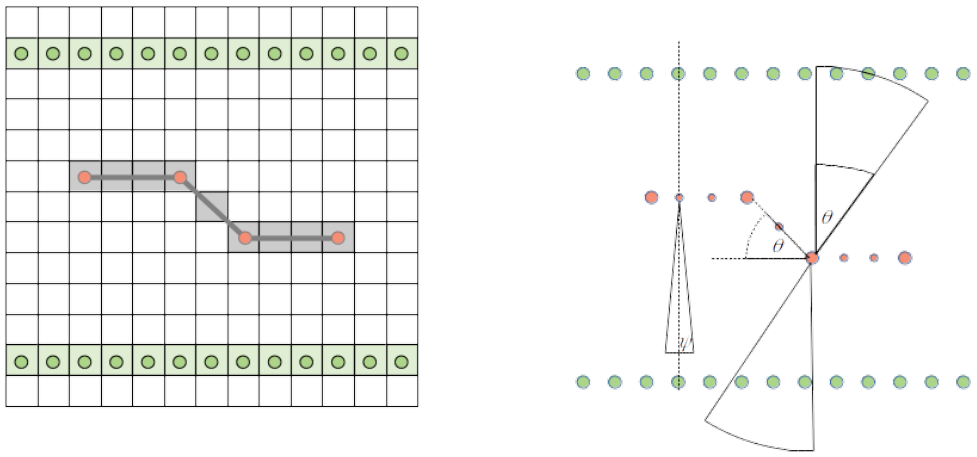


Figure 12: (a) Trajectory segments are projected to binary map to determinate unique trajectory positions which (b) are used to carry out visibility analysis in the Euclidean Space .

elements are necessary: i) the definition of the scanner characteristics, properties and the platform behaviour, ii) an adequate 3D model of the scene to be simulated, iii) a simplification of the real-world interaction between the laser beam and the objects in the 3D scene to be simulated.

The HELIOS++ framework is highly flexible and arranged into modules allowing for the definition of new sensors in terms of: the emitter (e.g., beam divergence full-angle, pulse frequencies, laser pulse length/duration, wavelength), the laser beam deflector (e.g., rotating mirror, fiber array, oscillating mirror, conic mirror, risley prims, etc.), characteristic of the range detector (e.g., minimum and maximum range, sensor accuracy) and scanner head properties (e.g., maximum speed, axis of rotation, etc.). Defined scanners can be coupled with static and dynamic as well as terrestrial and airborne platforms.

The model of the scene to be simulated can be provided as triangular meshes, digital elevation raster, voxel grids and point clouds. The model can be obtained either from a real survey or can be generated digitally. Materials can be applied to the 3D to simulate intensity behaviour and laser return intensity values are calculated using the laser radar equation.

The physical interaction among laser beam and simulated object is modelled using a set of parameters considering both physical and material behaviour of the object and characteristics of the laser beam such as: transmitted optical power, receiver aperture diameter, atmosphere transmission, beam divergence, angle of incidence, range distance to the target, angle of incidence to the target, reflectance of the target surface.

In this context simulations are carried out to evaluate the performance of the estimated optimal path considering two different metrics:

1. Data completeness, that considers if a certain area of the scene is captured or not by the scanner
2. Data density, that provides the amount of points recorded to approximate the surface of the reflector

Data completeness reflects if the surface of the investigated object is surveyed and represented in the point cloud or not. Lack of completeness can be the consequence of an incomplete acquisition due to wrong planning of the scan standpoints (static scanner) and paths (mobile scanner). In particular, data completeness is defined as the ratio between the surface of the scene covered by scan data and the total area of the scene. Completeness is evaluated considering a voxel based strategy: the original scene is represented using a regular voxel structure and if scanned data are present into a voxel occupied by the original simulated scene that voxel is labelled as "complete". Otherwise, if the voxel is not covered by scanner data it is labelled as "empty".

Data completeness accounts only if a certain area is covered by scan data or not. However, it is not accounting if the recorded point cloud has a sufficient resolution to reconstruct all the details of the investigated site. According to this concept, a further metrics is added in the evaluation of the estimated optimal path performance. Data density is locally evaluated on the surface within a prefixed diameter around each point, as proposed in Fugazza et al. (2018)). This metric can be depicted to model the point density distribution that can be obtained from the proposed data acquisition plan. A minimum threshold for the local point density can be established to check whether this parameter is acceptable or a revision of the scan plan is needed.

4. Results and discussion

Experiments carried out to validate the proposed method and the obtained results are outlined in this section. Datasets used for testing and the designing of the experiments are described in section4.1. Results presented in section 4.2 are divided in two parts in accordance with both main problems addressed: initial analysis and route planning. The results of the latter are further split into TLS and MLS analysis.

4.1. Case studies

With the aim to evaluate the robustness of the method for different applications, four case studies from indoor and outdoor environments have been selected (Fig.13). Case studies a) and b) correspond to indoor environments and they are provided by the ISPRS Benchmark on Indoor Modelling dataset (Khoshelham et al., 2017). An archaeological site is selected as case study c) in order to demonstrate the reliability of the method in complex and large outdoor scenarios. And finally, a large mall is selected as case study d). This mall has the particularity of being composed of rooms with a highly variable size. Special focus should be put into the large size of case studies; the total processed area is area is 304 m^2, 373 m^2, 18,364 m^2 and 35,503 m^2, for case studies 1, 2, 3 and 4, respectively.

Simulations were intended to assess the utility of the proposed method with three different theoretical laser scanners imitating core features of commercial devices. The technical specifications defined for the emulated systems are summarized in Table 2. One of the systems is a terrestrial laser scanner selected to simulate static acquisitions, while the other two are MLS systems used for dynamic mapping. The main differences between the two mobile systems in terms of scan planning are the maximum recommended acquisition time and the requirement of closed loops.

For each case study, five simulations are conducted. One simulation corresponds to a static TLS, two simulations are conceived for a MLS referred as IMMs-1 considering maximum acquisition times of 5 mins and 2 mins, respectively, and the last two simulations are conducted for another MLS coined IMMs-2 with 15 mins and 10 mins of acquisition time. For all MLS simulations, a travel speed of 4 km/h is considered. This value is similar to the base travel speed of non-disabled people (Montufar et al, 2007). The

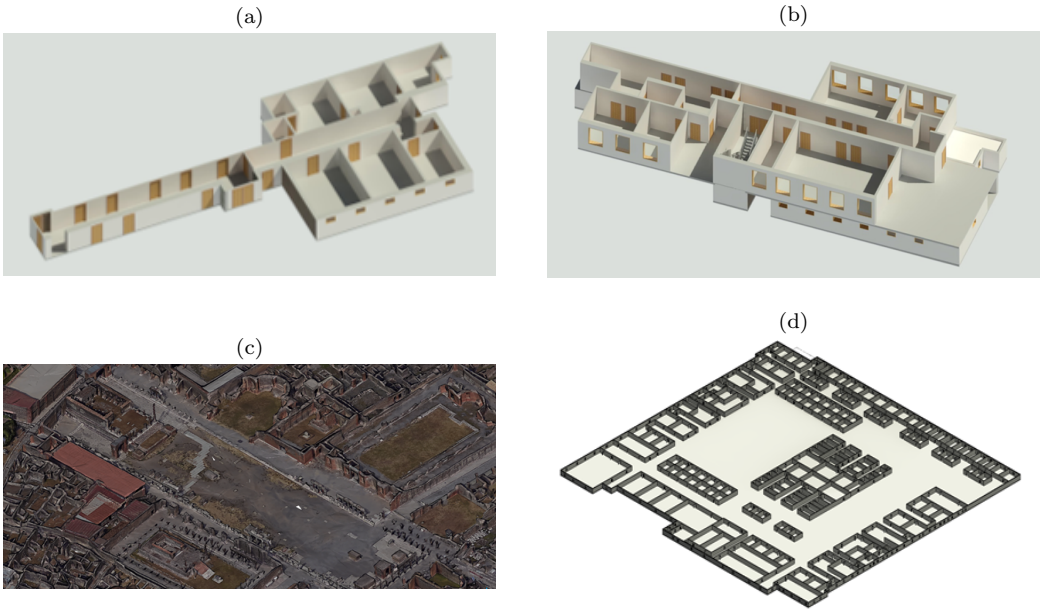


Figure 13: BIM of the case studies a) 1, b) 2, c) 3 and d) 4.

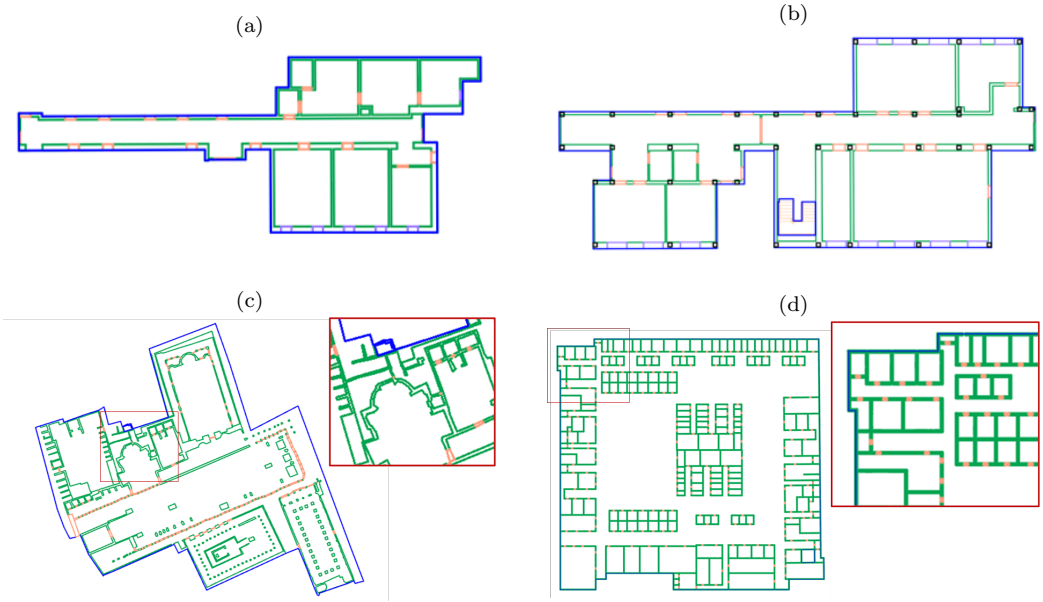


Figure 14: dxf models of the case studies a) 1, b) 2, c) 3 and d) 4.

Table 2: Technical specifications of laser scanner systems used to test the method.

	TLS-1 (TLS)	**IMMs-2 (MLS)**	**IMMs-1 (MLS)**
Field of view	360^{o}x360^{o}	360^{o}x270^{o}	360^{o}x270^{o}
Range	330	30m (optimal conditions)/ <10m (recommended)	0.5-25
Point measurement (rate pts/s)	976,000	43,000	420,000
Maximum recommended acquisition time	No limited	30 min	2 min
Closed-loops requirement	No	Yes	No

optimal route for TLS is also calculated to consider the option of mounting the system on an autonomous system for performing a Scan&Go procedure as in Frías et al. (2019). In this case, the optimal route corresponds to the shortest route not considering time restrictions.

4.2. Results

This section is devoted to show the results obtained from simulations. Section 4.2.1 collects the output data from initial analysis while the relevant results of route planning are presented and analysed in section 4.2.2. Results of simulations on Helios are reported in section 4.2.3.

4.2.1. Initial analysis

For initial scan candidate generation both partition methods were used, triangulation in the two first case studies, while a 2 m grid resolution was selected for cases 3 and 4. Regarding environment extension, two different laser ranges were chosen to carry out the planned simulations. A range of 5 m was used for the cases 1 and 2 whereas for the other much larger scenarios a laser operating distance was set to 10 m. From this configuration for simulations, the number of initial scan candidates obtained for studies case 1, 2, 3 and 4 were 151, 185, 3,261 and 6,675 respectively. Finally, scan positions were computed in order to ensure a theoretical coverage of 90% of the elements of interest for all of the cases resulting in 15, 18, 210 and 300 positions. The outcome of this analysis previous to route computation is depicted in Figs. 15 and 16

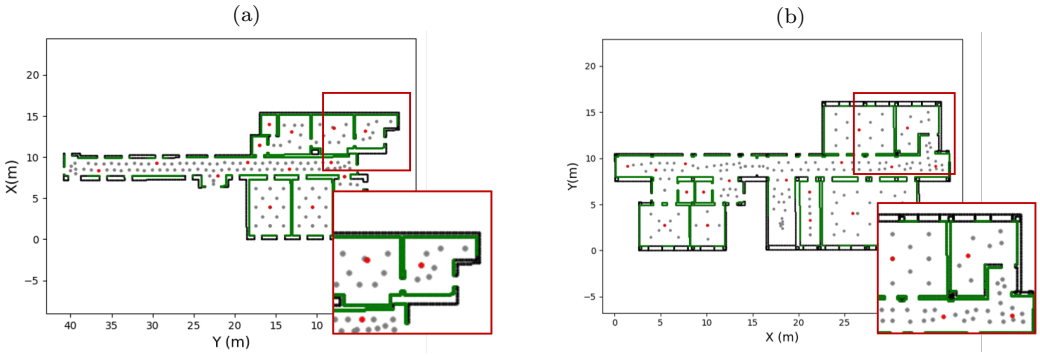


Figure 15: Initial analysis results depicting discarded candidates (grey) and scan positions (red).

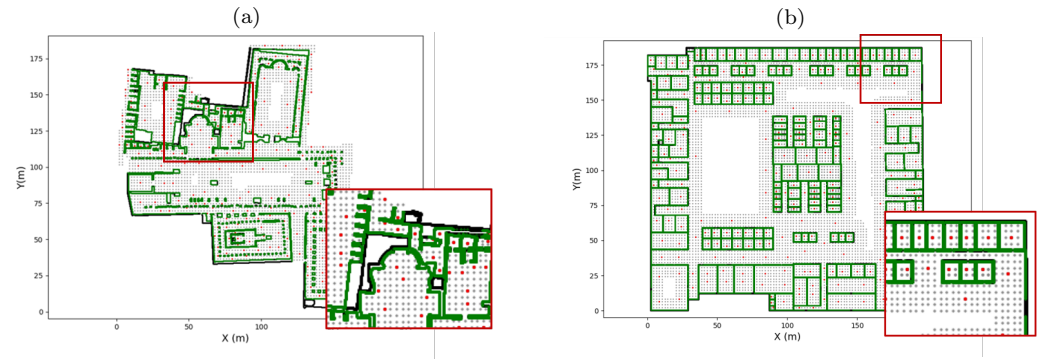


Figure 16: Initial analysis results depicting discarded candidates (grey) and scan positions (red).

4.2.2. Route planning

4.2.2.1. Route planning for TLS.

The main results for TLS simulation in the four case studies are collected in Table 3. The larger the case study extension, the longer the computed routes listed in the second column of Table 3. The processing time to compute the optimal route grows as well as the number of scan nodes increases. This is because the ACO algorithm complexity depends on the number of nodes involving a high computation time for route planning in large environments as in cases 3 and 4. The computed routes for the middle-sized buildings 1, 2 and large scenes 3, 4 are depicted in Figs. 17 and 18 respectively.

Table 3: Route length and processing time for TLS simulation with the four selected study cases.

Case study	Route length (m)	Processing time (s)
1	94.6	0.8
2	107.2	1.3
3	1,825.3	2,987.6
4	6,136.0	5,152.0

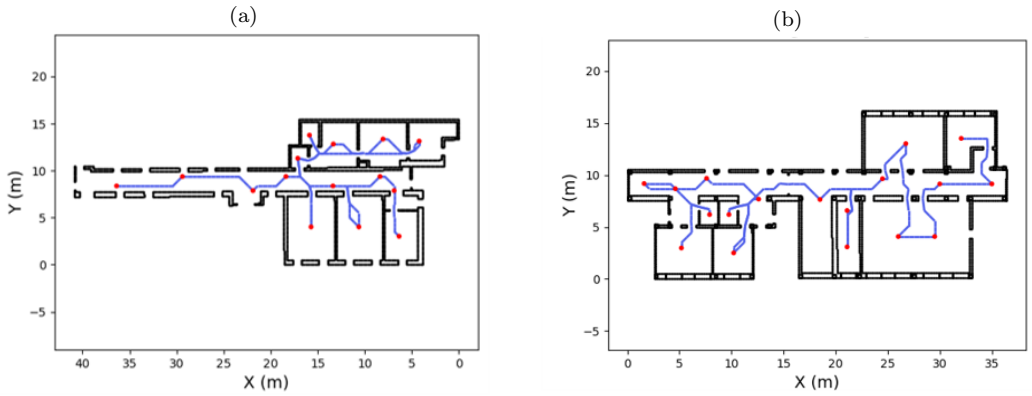


Figure 17: Routes obtained from TLS simulation for cases a) 1 and b) 2.

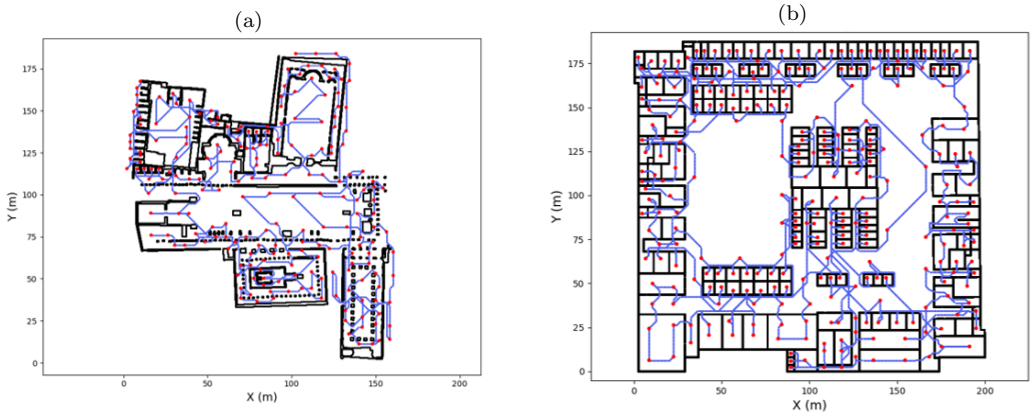


Figure 18: Routes obtained from TLS simulation for cases c) 3 and d) 4.

4.2.2.2. Route planning for MLS.

Some relevant results to analyse the computed routes for MLS are collected in Table 4 for the simulations described in section 4.1. The number

Table 4: Summarize of MLS results

Case study	Closed Loops	t_{max} (min)	k	Coarse clustering		Balanced clustering		Routes		
				ct	σ^2	ct	σ^2	ct	Avg. length (m)	Proc. time (s)
1	No	2	2	2	588.6	2	25.6	2	39.4	0.4
	No	5	1	1	0.0	1	0.0	1	94.4	0.9
	Yes	10	1	1	0.0	1	0.0	1	128.4	0.9
	Yes	15	1	1	0.0	1	0.0	1	129.3	0.9
2	No	2	2	2	0.3	2	0.3	2	52.0	0.6
	No	5	1	1	0.0	1	0.0	1	106.6	1.3
	Yes	10	1	1	0.0	1	0.0	1	138.0	1.3
	Yes	15	1	1	0.0	1	0.0	1	136.6	1.4
3	No	2	23	16	15 633.2	21	254.2	22	88.6	168.3
	No	5	13	9	77 454.0	13	2052.1	13	167.1	114.2
	Yes	10	8	5	58 888.5	8	1080.0	8	3.4	122.4
	Yes	15	6	4	87 776.0	6	2662.6	6	485.7	136.0
4	No	2	43	31	34 881.6	36	9709.8	38	95.4	276.6
	No	5	22	15	174 332.2	20	16 486.0	21	199.9	174.2
	Yes	10	14	9	219 839.8	13	18 642.6	14	394.3	160.0
	Yes	15	11	8	842 479.7	11	11 040.6	11	512.4	190.7

clusters k obtained in simulations, decreases longer the maximum acquisition time t_{max} and increases as the number of scan nodes is higher. This behavior is congruent with Eqs. 2 and 3 used for k calculation. The suitability of this estimation is evaluated in the analysis of final routes.

From k estimation, the balanced-clustering algorithm is tested by comparing balanced clusters with the coarse clustering previous to balance. The number of clusters complying time restriction (column labelled as 'ct') and the clustering distance variance (σ^2) have been the measures chosen to evaluate the performance of the process. In cases studies 1 and 2, the only simulation requiring more than one cluster is the 2 minutes without loops. For both cases, the two clusters obtained by coarse clustering satisfy time constraint but the balancing algorithm reduces clustering distance variance in case 1. Fig. 19 shows that cluster distribution is improved by applying the balancing algorithm. On the contrary, no balancing is performed in the case 2 since the result generated by the coarse clustering with variance of 0.32 m^2 is already valid.

Case studies 3 and 4 are really challenging for the balancing process since

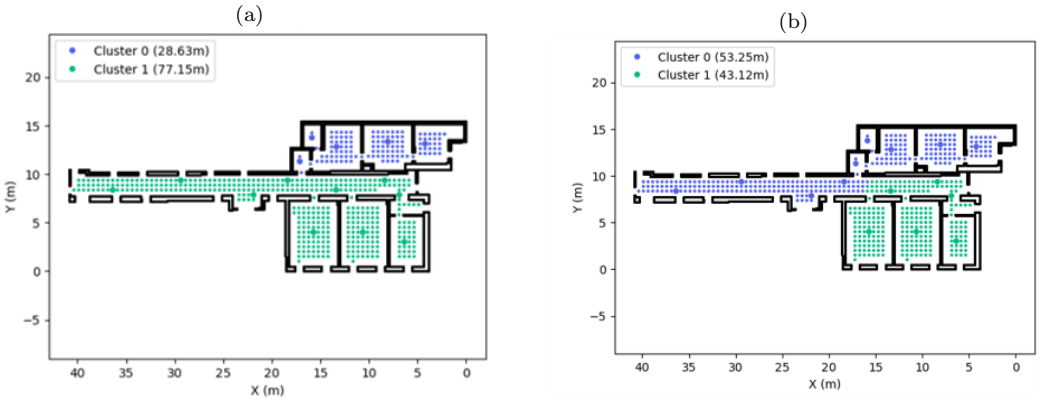


Figure 19: Clustering in the simulation of an acquisition performed by IMMs-1 with maximum acquisition time of 2 minutes in case study 1, before balancing (a) and after balancing (b).

the variance of distances of coarse clustering is high and several clusters exceed t_{max} condition. In case 3, the clustering distance variance is strongly reduced for all simulations and balanced clusters satisfy t_{max} restriction except in one simulation. This corresponds to the most restrictive simulation in time of 2 mins for which the coarse clustering generates clusters varying from 0 m (one scan node) to 400 m. Despite this difficulty the algorithm archives to improve significantly cluster distribution. Variance is also reduced but two clusters exceeds t_{max} by no more than 15 m due to a exceptional situation. Both clusters are almost the same length and they are adjacent to each other. Beyond, they are also adjacent to another cluster of length similar to d_{max} what does not allow any exchanges between them. Theoretically, the algorithm should be able to reduce this cluster in order to enable the exchange with its adjacent ones in a later iteration. But this is not reduced since the unique possible exchange with its adjacent cluster generates a longer expanded cluster, violating the exchange validation. Consequently, as the estimation of the distance is far from optimal route the balancing is not completed. Even more complex is the case study 4 both in extension and in space distribution. As in the other case studies, balancing algorithm improves significantly clustering distribution, however, some clusters exceed t_{max} limitation. By visual analysis we note that balanced clusters longer than d_{max} are located in the same areas during different simulations. These areas are composed of a narrow corridor connecting several small rooms adjacent

to each other on both sides. This distribution involves to perform a large multi-exchange operation for the balancing that, in many cases, infringes the the exchange condition locking node moving with their adjacent clusters. As these clusters are shorter than d_{max} when $t_{max} = 15$ min, the balancing is successfully completed as depicted in Figs. 20a and 20b, respectively.

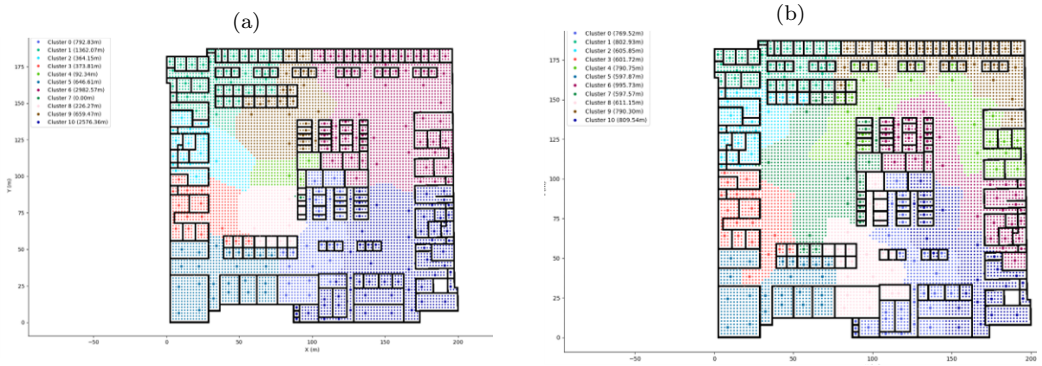


Figure 20: Clustering in the simulation of an acquisition performed by IMMs-2 with maximum acquisition time of 15 minutes in case study 4, before balancing (a) and after balancing (b).

The last three columns in the Table 4 contain the results of the computed optimal routes after adding shared nodes $n_{sh} = 1$. The average length of clusters for every simulation is at least a 10% shorter than the maximum acquisition distance. Therefore, the estimated number of clusters (k) is enough to carry out a cluster balancing fulfilling the adopted MLS constraints. An example of the final routes for each case study with open and closed loops are visualized in Figs. 21 and 22, respectively.

Processing time is the sum of the complete process from coarse clustering to route computation. While simulations requiring one route path are processed with a similar temporal cost as TLS route (case 1 and 2), processing time consumed to compute partitioned routes in the same cases is drastically reduced. Such as, the worst MLS simulation for case 4 the computation time was 276.6 seconds which is approximately a 5% of the route computation time in the TLS planning for the same case study.

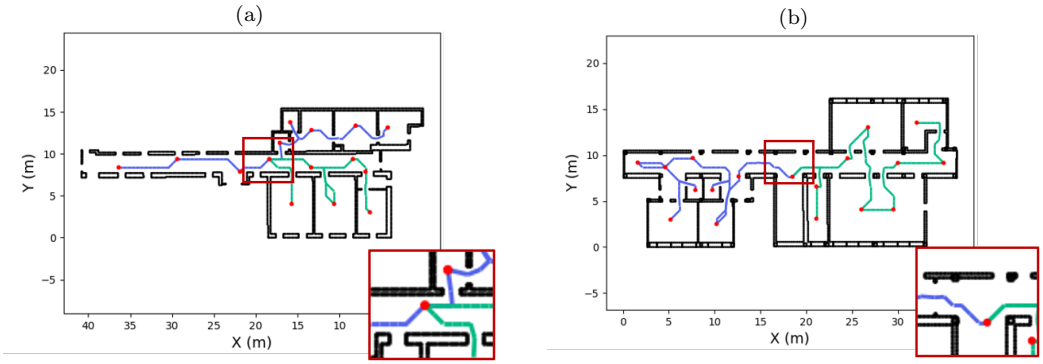


Figure 21: Final route with node sharing from IMMs-1 2min simulation for case study a) 1 and b) 2.

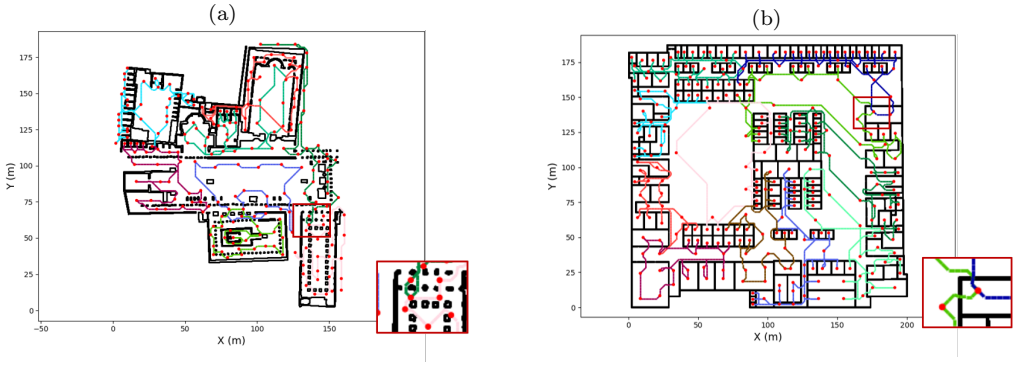


Figure 22: Final route with node sharing from IMMs-2 10 min simulation for case studies a) 3 and b) 4.

4.2.3. Scan simulation

4.2.3.1. Scan simulation for TLS.

The main results for TLS simulation in the four case studies are presented in Table 5 and Table 6. In Table 5, the second column represents the 2D completeness (ratio computed in 2D space) used as the stopping criteria for the definition of the optimal scanning positions to capture the scene. The remaining columns presents the completeness for the main elements composing the scene (i.e., wall, ceiling and floor) and the overall completeness of the scene. Completeness is evaluated considering a regular voxel size of (3.0 x 3.0 x 3.0 cm). As it can be observed the completeness of walls is

really close to the 2D completeness used as stopping criteria. Ceiling and floor present a lower completeness and this is mainly due to the fact that the acquisition is assumed on a tripod at an height with respect to the ground equal to 1.5 m so the area under the scanner tripod is not surveyed and the incidence angle between the laser ray and the reflector is unfavourable for those areas compared with walls.

Table 5: Data completeness for TLS simulation using Helios with the four selected study cases.

Case study	**Estimated 2D completeness**	**Overall 3D completeness (simulation)[%]**	**Wall completeness (simulation)[%]**	**Ceiling completeness (simulation)[%]**	**Floor completeness (simulation)[%]**
1	90	85.6	89.7	84.4	78.5
2	90	84.3	88.5	82.7	77.7
3	90	82.7	90.1	80.3	66.5
4	90	81.8	89.3	81.1	63.2

Table 6 presents data density as the percentage of points in the points cloud having a number of neighbouring points (computed in a circle of radius 0.2 m).

Table 6: Data density for TLS simulation in Helios with the four selected study cases.

Case study	**Point density > 5000 pts** [%]	**Point density > 15000 pts** [%]	**Point density > 25000 pts** [%]	**Point density > 35000 pts** [%]
1	87.55	66.86	37.10	15.50
2	86.24	64.52	35.98	15.23
3	63.30	21.57	5.68	3.61
4	62.45	19.96	6.49	2.87

4.2.3.2. Scan simulation for MLS.

The main results for MLS simulation in the four case studies are presented in Table 7 and Table 8. In Table 7 the results for the simulation of the for case study. Simulations were carried out considering a walking speed of 0.8 m/s.

Table 8 presents data density as the percentage of points in the points cloud having a number of neighbouring points (computed in a circle of radius 0.2 m).

Table 7: Data completeness for MLS simulation in Helios with the four selected study cases.

	Case study	Estimated 2D completeness	Overall 3D completeness (simulation)[%]	Wall completeness (simulation)[%]	Ceiling completeness (simulation)[%]	Floor completeness (simulation)[%]
IMMs-1 2 min	1	90	89.4	90.1	87.7	87.5
	2	90	89.7	86.9	86.9	86.8
	3	90	73.3	77.1	N.A.	61.8
	4	90	74.8	81.1	62.3	62.3
IMMs-1 5 min	1	90	90.1	90.3	89.3	89.3
	2	90	89.0	89.9	87.1	87.1
	3	90	75.2	79.0	N.A.	65.8
	4	90	76.8	81.3	65.3	65.3
IMMs-2 10 min	1	90	60.2	60.4	59.4	89.3
	2	90	61.0	61.5	60.7	60.7
	3	90	53.3	56.6	N.A.	40.1
	4	90	54.8	58.7	42.2	42.2
IMMs-2 15 min	1	90	60.2	60.4	59.4	59.3
	2	90	61.1	61.6	60.7	60.7
	3	90	52.4	55.4	N.A	38.3
	4	90	53.2	55.3	40.1	40.1

Table 8: Data density for MLS simulation with the four selected study cases.

	Case study	Point density > 5000 pts [%]	Point density > 15000 pts [%]	Point density > 25000 [%]	Point density > 35000 [%]
IMMs-1 2 min	1	53.44	14.52	6.66	4.24
	2	52.65	11.36	5.11	2.18
	3	19.80	9.24	2.12	0.35
	4	22.30	13.54	3.32	0.53
IMMs-1 5 min	1	50.87	14.01	5.95	3.72
	2	49.38	10.24	5.00	1.98
	3	20.13	10.43	3.29	0.43
	4	23.12	14.27	4.21	0.65
IMMs-2 10 min	1	15.67	5.12	0.87	N.A.
	2	12.45	4.79	0.54	N.A.
	3	0.40	0.02	N.A	N.A.
	4	0.52	0.03	N.A	N.A
IMMs-2 15 min	1	14.82	4.75	0.65	N.A.
	2	11.39	3.54	0.53	N.A.
	3	0.35	0.01	N.A.	N.A.
	4	0.50	0.02	N.A.	N.A.

Simulation results with IMMs-1 are showing for case study 1 and 2 a really high completeness (90%) even better with the one achieved in the simulation of the TLS. In particular, completeness of ceiling and floors is

high and this is mainly connected with the different acquisition pattern that is more favourable, for those surfaces, compared with TLS static acquisition. The high completeness is connected with the high point measurement rate. For case study 3 and 4 completeness decreases and this is, in our opinion, mainly due to the large scale of the case study, the relatively limited range of the instrument and the relatively small voxel size used for simulation.

Simulation results with IMMs-2 are showing a much lower completeness compared with IMMs-1 and, this could be mainly connected with the lower point measurement rate compared with IMMs-1. Results in terms of density are also confirming this.The lower measurement rate results in a much lower data density and keeping a quite tight voxel size (3.0 x 3.0 x 3.0 cm) influences the results also in terms of completeness. Considering density analysis the TLS data are showing as expected, a concentration of areas with high density points compared to MLS, that instead are showing a lower density but an higher homogeneity in the distribution.

5. Conclusion

In this paper, a novel approach for both static and mobile scan planning is presented. The method is flexible in the sense that it can deal with several MMSs constraints such as maximum acquisition time and non-closed loop and closed-loop requirement. These constraints configure the graph partition balancing process implemented to generate routes specific for an user-configurable MMS. The method is tested in four real case studies with variable size and complexity. Five simulations are performed for each case study considering different laser scanning system configurations. Simulation with Helios++ is used for demonstrate the reliability of the computed route in terms of 3D completeness, and validate the acquisition assumption for MLSs. Results show a good performance of the scan planning method, including optimal routing, in large scale and complex sites. Furthermore, scanning from multiple routes with MLS not only allows to capture the scene more quickly than using TLS but also the processing time to compute the optimal route is significantly reduced, particularly in large scenes.

In future work, some limitations should be addressed to improve the proposed method. The use of grid-based graph for large scenes involves a high computation cost in memory, therefore, a more efficient graph nodes dis-

tribution should be studied. In this line, a flexible graph generation for multiple platforms carrying the MLS or for different environments, i.e. indoor/outdoor would improve route accuracy.

6. Acknowledgements

This project has received funding from the Xunta de Galicia through project ED431C 2020/01, and from the Government of Spain through project PID2019-105221RB-C43 funded by MCIN/AEI/10.13039/501100011033 and through human resources grant RYC2020-029193-I funded by MCIN/AEI/10.13039/501100011033 y FSE "El FSE invierte en tu futuro". The open access fee has received funding from the University of Vigo/CISUG. This document reflects only the views of the authors. The statements made herein are solely the responsibility of the authors.

Appendix A. Balanced graph partitioning algorithm

This appendix is devoted to extend the balanced graph partitioning algorithm described in section 3.2.3.2. A detailed pseudocode of the entire process is collected in Algorithm 2. This takes the the coarse clustering computed in in the previous step C_{con}, the navigable graph G_n and the MLS constraints of maximum acquisition distance d_{max} and the closed loops requirementcl as input parameters. Before to start the main iterative process, estimated distances of each cluster D_{est} are computed by a Greedy-based algorithm to determinate the binary condition l_c which is true if any cluster is longer than d_{max}. To exchange control, the boolean matrix E is initially filled with true values where the elements $e_{l,m}$ with value true means that the exchange between cluster l and cluster $m$is enabled while the exchange is disable when the value is false. Consequently, the exchange between two clusters is allowed if they are adjacent and the corresponding change is enabled in E.

At each iteration, an adjacency cluster matrix A_c is computed to define adjacency relations between clusters from the clustering at this iteration C_b. Then, clusters to be reduced C_{rdc} and expanded C_{exp} are selected according to criteria explained in section 3.2.3.2. From C_{rdc}, adjacent nodes to C_{exp} are retrieved and arranged by the function *candidate_adj_nodes*. Next, the candidate exchange is evaluated in order of priority. Since candidate moving

may involve a multi-node exchange operation, the function *nodes_to_exchange* computes nodes which must be moved to keep connectivity of C_{rdc}. The resultant clusters after exchange are saved in $C_{rdc}^{'}$ and $C_{exp}^{'}$ to compose the clustering derived from the exchange $C^{'}$. From these, the length of new clusters $D_{est}^{'}$ and the clustering distance variance $\sigma_D^{2'}$ are calculated to determinate if the exchange satisfied the conditions (val_exc) defined in section 3.2.3.2. If not, the candidate n_a is remove from candidate list N_{adj} and the next candidate is evaluated. In case the exchange is valid, no more candi-

Algorithm 2 Cluster balancing

```
1: procedure CLUSTER_BALANCING(C_con, G_n, d_max, cl)
2:     C_b ← C_con
3:     E ← matrix_initialization()
4:     D_est ← greedy_estimation(C_con, cl)
5:     if max(D_est) > d_max then
6:         l_c ← True
7:     else
8:         l_c ← False
9:     σ²_D ← variance(D_est)
10:    while l_c == True or any(E) do
11:        A_c ← adjacent_cluster_matrix(C_b, G_n)
12:        D_est ← greedy_estimation(C_b)
13:        if l_c == True then
14:            C_rdc ← longest_cluster(C_b, E)
15:            C_exp ← adjacent_cluster(C_b, C_rdc, A_c, E)
16:        else
17:            d_avg ← mean(D_est)
18:            D ← D_est − D_avg
19:            C_max_dev ← abs_most_dev_cluster(C_b, D, E)
20:            C_adj_dev ← adj_offset_dev_cluster(C_b, D, A_c, E)
21:        C_rdc ← max_dev_cluster(C_max_dev, C_adj_dev)
22:        C_exp ← min_dev_cluster(C_max_dev, C_adj_dev)
23:        if C_rdc == null or C_exp == null then
24:            exit()
25:        N_adj ← candidate_adj_nodes(C_rdc, C_exp, A, G_n)
26:        val_exc ← False
27:        while N_adj ≠ ∅ and val_exc == False do
28:            n_a ← N_adj[0]
29:            N_e ← nodes_to_exchange(C_rdc, C_exp, A, G_n, n_a)
30:            C'_rdc ← C_rdc \ N_e
31:            C'_exp ← C_exp ∪ N_e
32:            C' ← C_b \ {C_rdc, C_exp}
```

33: $D^{'}_{est} \leftarrow$ **greedy_estimation**$(C^{'})$
34: $\sigma^{2'}_{D} \leftarrow$ **variance**$(D^{'}_{est})$
35: **if** $l_c == True$ **then**
36: **if** $D^{'}_{est}[C^{'}_{exp}] <= D_{est}[C_{rdc}]$ **then**
37: $val_{exc} \leftarrow True$
38: **if max**$(D^{'}_{est})$ **then**
39: $l_c \leftarrow False$
40: **else**
41: **if** $\sigma^{2'}_{D} < \sigma^{2}_{D}$ **and max**$(D^{'}_{est}) < d_{max}$ **then**
42: $val_{exc} \leftarrow True$
43: **if** $val_{exc} == False$ **then**
44: $N_{adj} \leftarrow N_{adj} \setminus n_a$
45: **if** $val_{exc} == True$ **then**
46: $C_b \leftarrow C^{'}$
47: $D_{est} \leftarrow D^{'}_{est}$
48: $\sigma^{2}_{D} \leftarrow \sigma^{2'}_{D}$
49: $E \leftarrow$ **unlock_exchange**(C_{rdc}, C_{exp})
50: **else**
51: $E \leftarrow$ **lock_exchange**(C_{rdc}, C_{exp})

Output:C_b

dates are tested and clustering is updated. Since C_{rdc} and C_{exp} have been modified due to the exchange, matrix E is also actualised to allow further exchanges from them.

References

Abellan, A., Oppikofer, T., Jaboyedoff, M., Rosser, N., Lim, M., Lato, M., 2014. Terrestrial laser scanning of rock slope instabilities. Earth Surface Processes and Landforms 39, 80–97. doi:10.1002/esp.3493.

Ahn, J., Wohn, K., 2015. Interactive scan planning for heritage recording. Multimedia Tools and Applications 75. doi:10.1007/s11042-015-2473-0.

Alba, M., Fregonese, L., Prandi, F., Scaioni, M., Valgoi, P., 2006. Structural monitoring of a large dam by terrestrial laser scanning. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 36.

Aryan, A., Bosché, F., Tang, P., 2021. Planning for terrestrial laser scanning in construction: A review. Automation in Construction 125, 103551. doi:https://doi.org/10.1016/j.autcon.2021.103551.

Biswas, H., 2020. Automatic planning for scanning: Optimizing 3d laser scanning operations using bim and tls. IAENG International Journal of Computer Science 46.

Blaer, P.S., Allen, P.K., 2007. Data acquisition and view planning for 3-d modeling tasks, in: 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 417–422. doi:10.1109/IROS.2007.4399581.

Buckley, S.J., HOWELL, J., ENGE, H., KURZ, T., 2008. Terrestrial laser scanning in geology: data acquisition, processing and accuracy considerations. Journal of the Geological Society 165, 625–638. doi:10.1144/0016-76492007-100.

Chvatal, V., 1979. A greedy heuristic for the set-covering problem. Math. Oper. Res. 4, 233–235. doi:10.1287/moor.4.3.233.

Connolly, C., 1985. The determination of next best views, in: Proceedings. 1985 IEEE International Conference on Robotics and Automation, pp. 432–435. doi:10.1109/ROBOT.1985.1087372.

Corso, N., Zakhor, A., 2013. Indoor localization algorithms for an ambulatory human operated 3d mobile mapping system. Remote Sensing 5, 6611–6646. doi:10.3390/rs5126611.

Dehbi, Y., Leonhardt, J., Oehrlein, J., Haunert, J.H., 2021. Optimal scan planning with enforced network connectivity for the acquisition of three-dimensional indoor models. ISPRS Journal of Photogrammetry and Remote Sensing 180, 103–116. doi:https://doi.org/10.1016/j.isprsjprs.2021.07.013.

Díaz-Vilariño, L., Frías, E., Previtali, M., Scaioni, M., Balado, J., 2019. Scan planning optimization for outdoor archaeological sites. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLII-2/W11, 489–494. doi:10.5194/isprs-archives-XLII-2-W11-489-2019.

Dijkstra, E.W., 1959. A note on two problems in connexion with graphs. Numer. Math. 1, 269–271. doi:10.1007/BF01386390.

Frías, E., Díaz-Vilariño, L., Balado, J., Lorenzo, H., 2019. From bim to scan planning and optimization for construction control. Remote Sensing 11. doi:10.3390/rs11171963.

Fugazza, D., Scaioni, M., Corti, M., D'Agata, C., Azzoni, R.S., Cernuschi, M., Smiraglia, C., Diolaiuti, G.A., 2018. Combination of uav and terrestrial photogrammetry to assess rapid glacier evolution and map glacier hazards. Natural Hazards and Earth System Sciences 18, 1055–1071. doi:10.5194/nhess-18-1055-2018.

González-Banos, H., 2001. A randomized art-gallery algorithm for sensor placement. Proc. 17th ACM Symp. Comp. Geom. , 232–240doi:10.1145/378583.378674.

Hu, T. C.and Kahng, A.B., 2016. Linear and Integer Programming in Practice. Springer International Publishing, Cham. pp. 117–130. doi:10.1007/978-3-319-24001-5.

Jia, F., Lichti, D., 2019. A model-based design system for terrestrial laser scanning networks in complex sites. Remote Sensing 11, 1749. doi:10.3390/rs11151749.

Kang, H.K., Li, K.J., 2017. A standard indoor spatial data model—ogc indoorgml and implementation approaches. ISPRS International Journal of Geo-Information 6. doi:10.3390/ijgi6040116.

Khoshelham, K., Díaz Vilariño, L., Peter, M., Kang, Z., Acharya, D., 2017. The isprs benchmark on indoor modelling. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLII-2/W7, 367–372. doi:10.5194/isprs-archives-XLII-2-W7-367-2017.

Kim, S.H., Li, K.J., Cho, H.G., 2020. A flexible framework for covering and partitioning problems in indoor spaces. ISPRS International Journal of Geo-Information 9. doi:10.3390/ijgi9110618.

Krishna, K., Narasimha Murty, M., 1999. Genetic k-means algorithm. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 29, 433–439. doi:10.1109/3477.764879.

Lewis, H.R., Papadimitriou, C.H., 1997. Elements of the Theory of Computation. 2nd ed., Prentice Hall PTR, USA.

Nocerino, E., Menna, F., Remondino, F., Toschi, I., Rodríguez-Gonzálvez, P., 2017. Investigation of indoor and outdoor performance of two portable mobile mapping systems, in: Remondino, F., Shortis, M.R. (Eds.), Videometrics, Range Imaging, and Applications XIV, International Society for Optics and Photonics. SPIE. pp. 125 – 139. doi:10.1117/12.2270761.

Oppikofer, T., Jaboyedoff, M., Blikra, L., Derron, M.H., Metzger, R., 2009. Characterization and monitoring of the Åknes rockslide using terrestrial laser scanning. Natural Hazards and Earth System Sciences 9, 1003–1019. doi:10.5194/nhess-9-1003-2009.

Otero, R., Garrido, I., Lagüela, S., Arias, P., 2020. Mobile indoor mapping technologies: A review. Automation in Construction 120, 103399. doi:10.1016/j.autcon.2020.103399.

Park, H.S., Jun, C.H., 2009. A simple and fast algorithm for k-medoids clustering. Expert Systems with Applications 36, 3336–3341. doi:https://doi.org/10.1016/j.eswa.2008.01.039.

Pito, R., 1999. A solution to the next best view problem for automated surface acquisition. IEEE Transactions on Pattern Analysis and Machine Intelligence 21, 1016–1030. doi:10.1109/34.799908.

Quintana, B., Prieto, S., Adán, A., Vázquez, A., 2016. Semantic scan planning for indoor structural elements of buildings. Advanced Engineering Informatics 30, 643–659. doi:https://doi.org/10.1016/j.aei.2016.08.003.

Rüther, H., Palumbo, G., 2012. 3d laser scanning for site monitoring and conservation in lalibela world heritage site, ethiopia. International Journal of Heritage in the Digital Era 1, 217–232. doi:10.1260/2047-4970.1.2.217.

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 International Journal of Geo-Information 7. doi:10.3390/ijgi7070250.

Scott, W.R., Roth, G., Rivest, J.F., 2003. View planning for automated three-dimensional object reconstruction and inspection. ACM Comput. Surv. 35, 64–96. doi:10.1145/641865.641868.

Soudarissanane, S., Lindenbergh, R., 2011. Optimizing Terrestrial Laser Scanning Measurement Set-Up. ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 3812, 127–132. doi:10.5194/isprsarchives-XXXVIII-5-W12-127-2011.

Watt, P.J., author, D.N.M.D.C., 2005. Measuring forest structure with terrestrial laser scanning. International Journal of Remote Sensing 26, 1437–1446. doi:10.1080/01431160512331337961, arXiv:https://doi.org/10.1080/01431160512331337961.

Winiwarter, L., Pena, A.M.E., Weiser, H., Anders, K., Sanchez, J.M., Searle, M., Höfle, B., 2021. Virtual laser scanning with helios++: A novel take on ray tracing-based simulation of topographic 3d laser scanning. arXiv:2101.09154.

Zhang, C., Kalasapudi, V.S., Tang, P., 2016. Rapid data quality oriented laser scan planning for dynamic construction environments. Advanced Engineering Informatics 30, 218–232. doi:https://doi.org/10.1016/j.aei.2016.03.004.

Zlot, R., Bosse, M., Greenop, K., Jarzab, Z., Juckes, E., Roberts, J., 2014. Efficiently capturing large, complex cultural heritage sites with a handheld mobile 3d laser mapping system. Journal of Cultural Heritage 15, 670–678. doi:https://doi.org/10.1016/j.culher.2013.11.009.