Optimal Path Planning for Kinematic Laser Scanning

Julius Knechtel[1], Mohammad Kordgholiabad[2], Jan-Henrik Haunert[1]

[1] Institute for Geodesy and Geoinformation, University of Bonn, Germany - (knechtel, haunert)@igg.uni-bonn.de
[2] Politecnico Milano, Italy - mohammad.kordgholiabad@mail.polimi.it

Keywords: Scan Planning, Kinematic Laser Scanning, Watchman Route Problem, Optimization, Mixed Integer Linear Programming

Abstract

Prompted by the rapid advancements in software and hardware, 3D building data for numerous different applications is nowadays often captured via mobile or kinematic laser scanning. However, in contrast to other laser scanning methods, there exist only a few approaches tailored for the planning of a kinematic laser scan survey, and none of them provides an optimality guarantee. Therefore, we propose a novel approach based on Mixed Integer Linear Programming (MILP) to find the optimal trajectory for such a survey. To obtain a high-quality point cloud, we account for scanner-related constraints that influence the quality of the resulting point cloud. Moreover, we enable the introduction of tie points to mitigate the effects of uncertainties in the position estimation that are propagated in the acquired data. In our problem formulation, we aim to find the best tour in a properly weighted graph. For this, we propose two different weight settings to either enable a purely length-based optimization or to increase the redundancy in the measurements by incorporating a *Visibility Ratio Factor* (VRF) into the objective function.
To prove the applicability of our approach for offline panning, we apply our formulation to three different scenarios. In this context, the VRF-based weighting enables a significant speed-up of the solving process while resulting in only slightly prolonged routes. This approach paves the way for applying exact algorithms with an optimality guarantee in the planning process for efficient kinematic laser scanning surveys.

1. Introduction

The accurate acquisition of 3D building data is a crucial building block for many applications in the context of, for example, Building Information Modeling (BIM), change detection, construction control, and heritage preservation (e.g., Díaz Vilariño et al. (2019); Meyer et al. (2022)). In recent years, the rapid development in both hardware and software has prompted a growing use of laser scanning to obtain the required data. Depending on the use case, there are restrictions imposed on the quality of the acquired data, which is, among other things, reflected in the choice of the surveying method (Kim et al., 2019). In general, kinematic or mobile laser scanning (MLS) offers a lower point quality in the resulting point cloud compared to, e.g, stop-and-go laser scanning (Lin et al., 2013), since the position of the scanner needs to be determined during the survey, and errors in this process directly propagate to the point cloud. However, an advantage of kinematic laser scanning is that the survey can normally be performed faster compared to static and stop-and-go laser scanning. This characteristic renders it a favorable option in numerous scenarios (Kim et al., 2019).

Depending on the platform used, it is possible to distinguish between different variants of mobile laser scanning. Human-carried laser scanners are often referred to as personal or Portable Laser Scanners (PLS), or as Wearable Laser Scanners (Di Stefano et al., 2021). These systems enable rapid data acquisition even in confined and tight spaces that are hard to navigate. It is common practice that the surveyor plans the corresponding route directly on-site. However, this approach poses the risk that some parts of the object of interest remain unobserved and, consequently, the resulting point cloud is incomplete. Additionally, some scanner-related constraints that influence the quality of the acquired data need to be considered: the minimum and maximum distance between the scanner and the object, as well as the incidence angle of the laser beam with respect to the surface (Soudarissanane et al., 2011). Assessing these parameters in a purely visual way requires a significant degree of experience and is error-prone.

In addition to the human-carried scanners, other MLS platforms are, for example, wheel-based. This enables the application of (semi-) automatic approaches in which a robot follows a (predefined) trajectory in order to acquire the required data. This offers the possibility to survey even hazardous areas that pose risks to humans. In the robotics domain, researchers often aim to simultaneously map an unknown environment and localize the robot within said environment (Grisetti et al., 2025). In this context, the path planning is mostly performed from the perspective of *coverage mapping*. Here, the objective is to observe all parts of a scenery at least once, often by following the detected boundary of an object. However, the quality of the resulting point cloud is not considered. Additionally, the focus is not on finding the shortest possible path. The economic efficiency of a survey, however, is determined by the time allocated for its completion. For MLS, this directly corresponds to the length of the survey path. This is additionally important, since a path that is as short as possible is more likely to satisfy the battery requirements of an autonomous system.

The efficiency and effectiveness of a survey plan for laser scanning in general highly depend on (1) the duration of the survey campaign, (2) the completeness of the resulting point cloud, and (3) the quality, i.e., accuracy, of the acquired data. Although some recent approaches for calculating survey campaigns for static or stop-and-go laser scanning that consider all three criteria can be found (e.g., Jia and Lichti (2019); Noichl et al. (2024); Knechtel et al. (2025)), there exists little research with a focus on kinematic laser scanning. To the best of our knowledge, there exists no exact approach that comes with an optim-

ality guarantee with respect to the given input.

Thus, in this article, we propose an algorithm to calculate the optimal round tour tailored for the requirements of mobile laser scanning in an offline way, i.e., before the surveyor arrives at the scene. The two-dimensional input, which could be derived, e.g., from *OpenStreetMap*, describes an outdoor scene comprising multiple buildings that need to be surveyed. The problem of finding the shortest path to survey all parts of a building is closely related to the *Watchman route problem* (WRP), which is proven to be *NP-hard* for polygons with holes (Chin and Ntafos, 1988). Consequently, the existence of an algorithm that can solve this problem optimally and efficiently, i.e., with a polynomial worst-case running time, is highly improbable. Therefore, we employ Mixed Integer Linear Programming (MILP), which guarantees an optimal solution, but features an exponential worst-case running time. In practice, however, it often performs better. In our planning, we additionally incorporate the aforementioned scanner-related constraints to obtain a point cloud of high quality. The aim is to minimize the weighted tour in a properly weighted navigation graph. In addition to a purely length-based weighting, we propose the *Visibility Ratio Factor* that is included in the objective function. With this, we can obtain a higher level of redundancy in the point cloud while still maintaining the focus on the minimization of the length of the survey route. The influence of the modified weighting on the length of the resulting path, as well as on the time needed to calculate an optimal solution, is subsequently investigated.

The remainder of the paper is structured as follows: After presenting relevant related works in Section 2, we describe in Section 3 in detail the methodology of our approach as well as the underlying MILP formulation. Our experiments and the respective results are shown in Section 4, followed by a summary and an outlook on future work in Section 5.

2. Related Work

Different planning algorithms for all types of laser scan surveys have been studied in recent years. In general, when talking about laser scanning, one needs to distinguish between static, stop-and-go, and kinematic laser scanning. Due to the differing requirements for the respective techniques, different planning algorithms need to be employed. An overview of multiple applications, the applied techniques, and the respective quality requirements is provided by Kim et al. (2019).

For static scanning, a set of standpoints on which the scanner is placed is calculated. The main goal is to retrieve a complete point cloud for the object of interest. To obtain a point cloud of high quality, recent approaches consider not only plain visibility but also scanner-related constraints (e.g. Jia and Lichti (2019)). As examined by Soudarissanane et al. (2011), the object should be neither too close nor too far away from the scanner. Reasonable values for this are provided by the manufacturer of the scanner. Additionally, retroreflective surfaces are often associated with data voids and hotspot effects (Fong and Yan, 2025). Such effects can also be mitigated by maintaining an offset distance from the respective surface. Moreover, the incidence angle between the laser beam and the observed surface influences the point quality. For the planning under these constraints, we need to distinguish between heuristic approaches (e.g. Díaz Vilariño et al. (2019)) and approaches with optimality guarantees (Biswas et al., 2015; Dehbi et al., 2021) that apply (Mixed) Integer Linear Programming ((M)ILP). The latter approach additionally guarantees a successful software-based registration of the single scans to obtain a joint point cloud. An overview of different methods of Planning4Scanning (P4S) with the focus on static scanning is provided by Aryan et al. (2021).

For stop-and-go laser scanning, a route that a platform equipped with a laser scanner can follow is additionally required. Measurements, however, are still performed in a static fashion, i.e., the platform stops at predefined standpoints to conduct a complete scan of the environment. Although this technique features a high accuracy due to the static measurement process, the time needed for the complete survey is longer than for kinematic scanning (Lin et al., 2013). With a focus on the monitoring of the construction progress of a building, Frías et al. (2019) employ a probabilistic ant colony algorithm to compute a route between standpoints that are previously computed using a greedy algorithm. The approach by Noichl et al. (2024) consists of a greedy heuristic as well as a genetic algorithm to compute a suitable set of standpoints. The shortest route between them is subsequently calculated with an approximation algorithm for the Traveling Salesperson Problem. Knechtel et al. (2025) solve both problems, i.e., the calculation of the optimal set of standpoints and the shortest tour between them, simultaneously, enabling shorter routes compared to subsequently solving both problems. The approach again employs Integer Linear Programming, i.e., comes with an optimality guarantee with respect to the candidate positions. Additionally, it features specific constraints guaranteeing a certain level of redundancy in the registration process of the single point clouds, thereby overcoming uncertainties and decreasing the risk of failures during the software-based registration.

During a kinematic survey, in some contexts also referred to as mobile mapping, the scanner is again following a trajectory, thereby eliminating the necessity to register the individual scans together. To subsequently obtain a point cloud, the trajectory of the system needs to be retrieved. Oftentimes, this is done by performing a fusion of measurements from an Inertial Measurement Unit with measurements obtained from GNSS. This process can be enhanced, for example, by including landmarks with known positions (Esser et al., 2022). Calculating the optimal, i.e., shortest, route for kinematic scanning while maintaining full coverage of the object of interest is closely related to the Watchman Route Problem, i.e., to find the shortest route that a watchman has to follow to guard an art gallery. Chin and Ntafos (1988) proved that this problem is *NP-hard* for polygons with holes by reducing the geometric Traveling Salesperson Problem (Papadimitriou, 1977) to it. The problem presented in this article is even more complex, since the visibility is restricted due to the scanner-related constraints.

There exist only a few approaches tailored for planning for kinematic scanning. The method for mobile mapping on large sites presented by Frías et al. (2022) again relies first on a greedy weighted viewpoint calculation, based on two-dimensional input polygons. For the route planning, the approach enables, among other things, splitting the routes into multiple subtours to account for battery restrictions coming with some mobile mapping systems when acquiring data of large sceneries. For the dynamic acquisition, a profile scanner is assumed, scanning the objects lying on a 2D plane that is orthogonal to the system's movement direction. The approach by Poku-Agyemang and Reiterer (2025) is focused on the detailed acquisition of a point cloud of infrastructure, e.g., a bridge. The input is a 3D model of the bridge generated from the technical drawings.

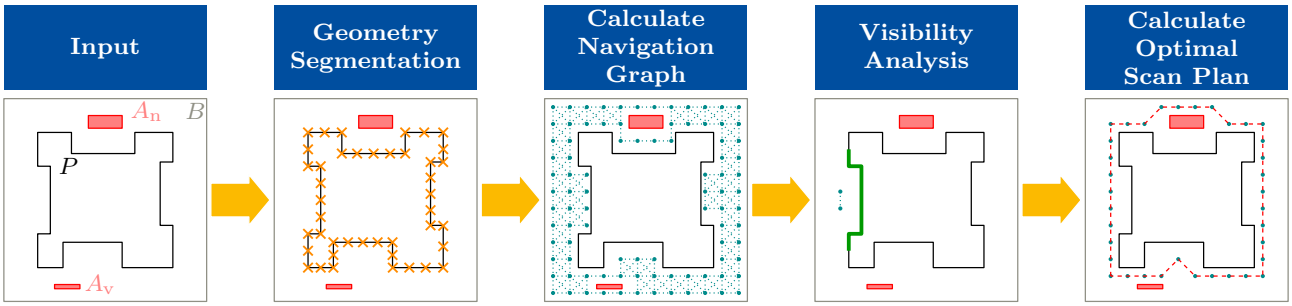


Figure 1. Overview of the different steps of our proposed algorithm.

Viewpoint candidates are weighted with their respective coverage and greedily selected. The kinematic scanning path is obtained by applying the A* algorithm on the selected viewpoints. The approach, however, does not explicitly consider the measurements while the scanner follows the route between the selected viewpoints. The importance of a good trajectory to obtain a satisfactory model of an object of interest is examined by Kuželka (2025), exemplary for a forest.

In the robotics domain, autonomous systems are often applied to map an unknown environment. The two tasks of mapping the environment and localizing the robot in the latter are simultaneously performed. An overview of different related works in this domain is given by Grisetti et al. (2025). In this context, however, the focus is on mapping all objects without directly considering the quality of the resulting point cloud from a survey perspective or the characteristics of a laser scanner.

Neither of the aforementioned approaches provides an optimality guarantee for the route length, which is, among other things, the crucial parameter for economic efficiency, since it influences the time needed for data acquisition. It is additionally directly related to the amount of collected data that needs to be subsequently processed. While a certain degree of redundancy may be considered beneficial, an excess of data can lead to excessive captured information and prolonged processing times. To overcome this research gap, and by extending a master’s thesis (Kordgholiabad, 2025), we propose in this article a novel, MILP-based formulation that minimizes the length of the route the system needs to traverse. Additionally, we propose a slightly modified weighting that is applied in the objective function, which aims to implicitly achieve a higher degree of redundancy in the results.

3. Methodology

Our proposed algorithm to calculate an optimal route for kinematic laser scanning consists of four steps, which are depicted in Figure 1. Based on two-dimensional input polygons comprising the scenery to observe, we (1) split the input geometries into a set of segments S that need to be observed. Subsequently, (2) based on a grid structure and while accounting for potential obstacles, we calculate a navigation graph G. The edges in this graph serve as candidate edges for the subsequent path planning. Furthermore, (3) for each navigation edge $e \in G$, it is calculated which $s \in S$ can be observed while traversing the particular edge. In this step, we account for the scanner-related constraints that influence the quality of the resulting point cloud. As a final step, we (4) employ Mixed Integer Linear Programming to calculate the optimal round tour on G which is capable of observing all objects of interest. The single steps are explained in more detail in the following subchapters.

3.1 Input

Our algorithm relies on a set of input geometries describing the scenery that needs to be surveyed with kinematic laser scanning. First, a boundary polygon B is introduced that bounds the navigable space. For the buildings comprising the scene and other objects of interest that should be surveyed, we obtain the set of their respective footprints P. This reduction from the three-dimensional to the two-dimensional space is often performed in the existing literature and was exemplary proven to be efficient for real-world outdoor sceneries, e.g., by Knechtel et al. (2022). This assumption is, of course, not applicable for high buildings or narrow urban canyons due to the resulting steep inclination angles. However, in these scenarios, the application of terrestrial laser scanning is often restricted. Moreover, we include obstacles $A = A_\mathrm{n} \cup A_\mathrm{v}$ in our approach. A_n describes areas that are not viable, i.e., through which the scanner cannot navigate, for example, due to the underground conditions. Other obstacles that not only influence the navigability but also obstruct the visibility, e.g., walls or trees, are again described as 2D polygons (A_v). On this basis, we calculate the routable space R in which the scanner can be placed and follow a route within, i.e., $R = B \setminus (P \cup A)$.

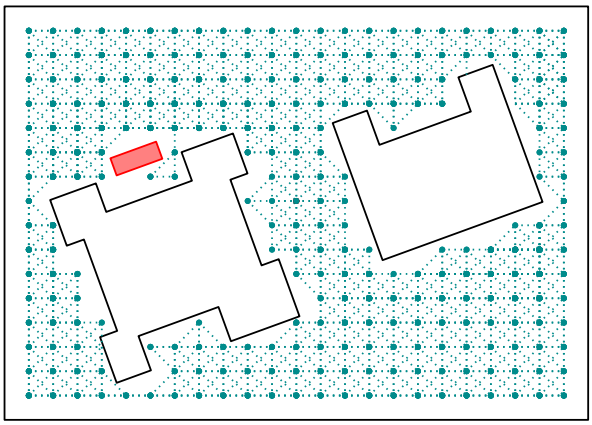
3.2 Segment Generation

In the second step, we discretize the outline of the objects of interest P to allow a subsequent determination of the observability of the façade. Each segment from the input polygons is equally subdivided into smaller segments. The result is a set of segments S that feature a length of at most l_seg. Each segment is a straight line between its respective start and end points s_s and s_e. l_seg is a user-selectable parameter that, on the one hand, determines the granularity of the observability calculations, but, on the other hand, heavily influences the calculation time by introducing a larger amount of façade segments. In a later step, for each segment, a constraint is imposed to guarantee the completeness of the acquired data.

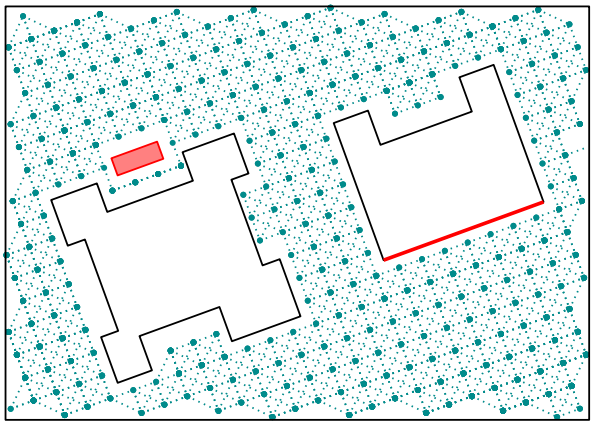
3.3 Navigation Graph

Subsequently, we calculate the directed graph $G = (V, E)$, which we refer to as the navigation graph. For this, a rectangular grid is generated inside the routable space R. The distance between the grid points is again a user-selectable parameter l_grid. A dense grid enables more and potentially shorter routes. However, it also results in prolonged calculation times for the optimal scan plan by introducing more candidate edges. The grid points serve as the nodes V in the navigation graph, i.e., are navigation way points. The nodes then form an 8-connected grid with their respective neighbors. Each connection from node v to a neighbor u leads to pairwise candidate

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-XI-2-2026-109-2026



(a)



(b)

Figure 2. Different orientations of the rectangular grid and the navigation graph derived from it. In (a), the grid follows the overarching coordinate grid. In (b), it is oriented on the longest building façade (red), thereby enabling more suitable paths.

edges (u, v) and (v, u) in the navigation graph. However, to ensure a safe navigation, it is necessary to take the obstacles A as well as the building itself into account. Consequently, we only introduce edges that do not intersect with any of the polygons contained in A and P. Furthermore, these edges must also maintain a distance of no less than a safety parameter (l_{safe}) from any buildings or obstacles to ensure a safe navigation.

Often, such a rectangular grid follows the orientation of the underlying coordinate system. The objects of interest usually also feature global coordinates for georeferencing. However, they seldom align with the coordinate grid. The effect of this misalignment is exemplarily depicted in Figure 2(a). Especially around recesses and obstacles, the navigation graph is poorly adapted to the situation, which can hinder the efficiency of the survey. Therefore, we align our navigation grid along the longest façade part present in the scenery to observe, which is marked in red in Figure 2(b). The resulting grid is better adapted and allows for more suitable measurement paths while maintaining the original coordinates of the objects of interest. Similar benefits of aligning the grid-based search space to the dominant scene orientation are reported in Navarro (2023).

3.4 Visibility Analysis

Each of the segments in S needs to be observed at least once during the scanning process. For now, we focus on a scanner

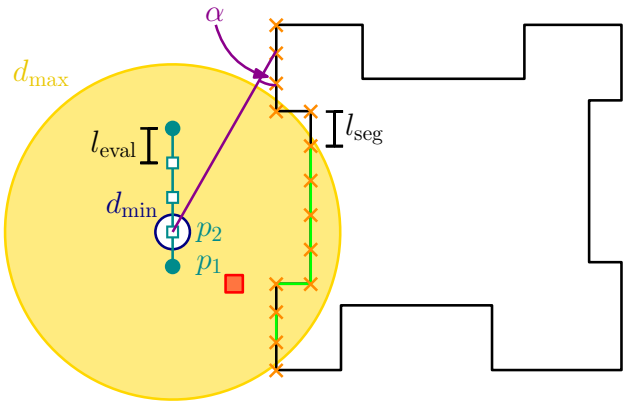


Figure 3. Exemplary visibility calculation performed for the second sample point p_2 on one navigation edge e. The segments of the building are indicated by orange crosses at their respective start and end points. A segment is observable from one sample point (green) if both start and end points are visible while considering the scanner-related constraints. Otherwise, the subsequent sample points need to be considered if they are able to overcome this deficiency.

that performs 360° scans. However, this calculation could be easily adapted for a profile laser scanner. In this step, we need to calculate which parts of the respective buildings are observable while traversing which navigation edge. This information is stored in an observability matrix O with binary entries:

$$O_{ij} = \begin{cases} 1 \text{ if segment } s_j \text{ fully observable from edge } e_i \\ 0 \text{ otherwise} \end{cases} \quad (1)$$

In Figure 3, the process of calculating the visible segments for one exemplary navigation edge (cyan) is depicted. For our work, we define that a segment s is observable from a particular navigation edge e if the start point s_s and end point s_e (orange crosses) can be observed at least once while the scanner traverses the edge. To keep the calculation times for the visibility within reasonable limits, we evaluate the visibility for each edge only for selected sample points. For this, the user can define a parameter l_{eval} that determines the maximal distance between two sample points on the visibility edge. These sample points are in Figure 3 depicted as cyan squares. The start and end nodes of the navigation edge serve as the first and last sample points. In this figure, we exemplarily show the visibility calculation for the second sample point p_2. In general, observability can only be given if a straight line of sight between the current position and the object of interest exists. For this check, the buildings themselves as well as the visibility obstacles A_{v} are taken into account. Additionally, the aforementioned scanner-related constraints need to be considered: the minimum measurement distance $d_{\min}$ (blue), the maximum measurement distance $d_{\max}$ (yellow), and the incidence angle α (lilac) between the laser beam and the surface. The entries in the observability matrix O for edges where both start and end points are observable (green) are consequently set to 1. If one or both of (s_s, s_e) are not observable, we need to proceed with the other sample points on the edge to check if they are capable of (partly) observing the missing segments.

3.5 Calculate Optimal Scan Plan using MILP

To calculate an optimal route for a kinematic laser scanning survey, one needs to decide which of the edges in the navigation

graph are part of the tour. For this, we introduce a binary variable $x_{uv} \in \{0, 1\}$ for each directed candidate edge $e = (u, v)$ that is part of the navigation graph G. This variable can be interpreted as follows:

$$x_{uv} = \begin{cases} 1 \text{ if edge } (u, v) \text{ is part of the survey route} \\ 0 \text{ otherwise} \end{cases} \tag{2}$$

The aim is to minimize the corresponding objective function:

$$\text{Minimize} \sum_{(u,v)\in E} w_{uv} \cdot x_{uv} \tag{3}$$

In this objective, we apply a specific weight factor w_{uv} to each candidate edge. In general, this factor can be arbitrarily set, indicating, for example, how long it takes to traverse an edge, considering environmental factors such as slope and underground conditions. In our experiments, we distinguish between two different weight factors. The first weight factor indeed corresponds to the length of the particular edge, i.e., the Euclidean distance between the respective start and end points. This serves as a simple indication of the travel time associated with traversing the edge. The second weight factor we utilize is designed to encourage the solver to obtain greater redundancy in the observations by selecting navigation edges that are capable of observing as many segments as possible. For this, we first define the *Visibility-Ratio Factor* (VRF):

$$\text{VRF}(e) = \frac{|\{s_j \in S \mid O_{ej} = 1\}|}{\text{len}(e)} \tag{4}$$

This factor indicates how many distinct parts of the façade can be observed while traversing the navigation edge. We can count for each candidate edge how many distinct segments can be observed by summing over the corresponding row in the visibility matrix O. This number is then normalized by the length of the edge. Consequently, the VRF can be described as how many of the segments S can be observed per unit of length traveled. To still emphasize the main goal of finding a short survey tour, the length of an edge $e = (u, v)$ is again incorporated in the calculation of the particular weight factor:

$$w_{uv} = \frac{\text{len}(e)}{\text{VRF}(e) + \epsilon} \tag{5}$$

A small value ϵ, e.g., 10^{-6}, prevents a division by zero if an edge does not observe any segment, but is incorporated into the tour to connect other candidate edges. Consequently, edges that feature a low VRF obtain a high weight and are, thus, less probable to be selected since the objective function is minimized.

The objective function is subject to various constraints that ensure that each part of the object of interest is observed at least once during the scanning process and that the resulting survey tour is a feasible round tour. The first set of constraints ensures said observability by considering the previously calculated observability matrix O. For every wall segment of the building, at least one of the navigation edges that are capable of observing this segment needs to be selected:

$$\sum_{e=(u,v)\in E} O_{es} \cdot x_{uv} \geq 1 \;\; \forall \;\; s \in S \tag{6}$$

Our kinematic scan plan should additionally feature a suitable round tour. The following set of constraints enforces that the number of selected incoming edges equals the number of selected outgoing edges for every node in the navigation graph:

$$\sum_{(v,u)\in E} x_{vu} - \sum_{(u,v)\in E} x_{uv} = 0 \;\; \forall \;\; v \in V \tag{7}$$

An additional set of constraints can optionally be added if the tour should bypass a set of user-selected tie points $T \subseteq V$. For each tie point $t \in T$, a parameter m_t can be chosen, determining how often the tour should bypass this point:

$$\sum_{(t,u)\in E} x_{tu} \geq m_t \;\; \forall \;\; t \in T \subseteq V \tag{8}$$

This constraint enforces that for each selected tie point, at least m_t outgoing navigation edges are selected for the final survey plan. Due to the balance enforced by Equation 7, it is sufficient to set up this constraint only for the outgoing navigation edges. With this, the user is able to force a loop closure at certain points to mitigate the effects of dead reckoning, i.e., a drift in the point cloud that can occur for long trajectories.

Applying only the aforementioned constraints, however, can lead to unintentionally disconnected subtours. To avoid this, we apply the flow conservation model of Shirabe (2004), which can be used to ensure the connectivity of a subgraph $G' \subseteq G$, consisting of the selected navigation edges and the nodes connecting them. Each node in the subgraph injects a portion of flow into the network that is carried upon the edges. Only one of the selected nodes is capable of taking up the flow. Thereby, a path from each selected node to this sink is enforced. Since this model is well-known and frequently applied, we provide only a brief overview and explanation of the necessary variables and constraints. Each navigation edge $(u, v) \in E$ receives an additional variable $f_{uv} \in \mathbb{R}_0^+$ modeling the flow on the edge. Additionally, each node $v \in V$ is assigned two binary variables that determine if v is part of the subgraph ($y_v \in \{0, 1\}$) and if v serves as the sink ($s_v \in \{0, 1\}$). The first constraints couple the variables with the edge-selection variable: Flow can only be present on selected navigation edges, and a node with at least one incident selected edge needs to be selected as well:

$$f_{uv} \leq (|V| - 1) \;\cdot\; x_{uv} \;\; \forall \;\; (u, v) \in E \tag{9}$$

$$y_v \geq x_{uv} \;\; \forall \;\; (u, v) \in E \tag{10}$$

Moreover, we ensure that only one sink exists, and the sink is a node that is selected:

$$\sum_{v\in V} s_v = 1 \tag{11}$$

$$s_v \leq y_v \;\; \forall \;\; v \in V \tag{12}$$

The flow conservation constraint ensures that each selected node v contributes flow to the network and that no flow gets lost. This, however, is relaxed if the node is the sink, then it is capable of taking up all the flow in the network:

$$\sum_{(v,u)\in E} f_{vu} - \sum_{(u,v)\in E} f_{uv} \geq y_v - |V| \;\cdot\; s_v \;\; \forall \;\; v \in V \tag{13}$$

With the presented constraints, we obtain the subgraph $G' = (V', E')$ with the minimum total weight from which the scanner is able to observe all parts of the building. This subgraph is (1) connected and (2) has only nodes where the in-degree

Table 1. Results for the three datasets with both weighting techniques applied.

	grid size [m]	weighting	path length [m]	# selected edges	mean VRF	computation time [s]
ex1	10	length	766.27	53	0.4528	3045
		VRF	796.98	71	0.5471	66
ex2	8	length	414.39	46	0.5743	50
		VRF	451.08	51	0.6274	40
ex3	10	length	477.99	36	0.5147	25200
		VRF	480.42	41	0.6682	57

equals the out-degree. These two properties are at the same time the conditions for the existence of an *Eulerian cycle* in a directed graph. Consequently, we can, in a final step, find such an Eulerian circle in the optimal subgraph in $\mathcal{O}(E')$ to obtain a suitable round tour (Biggs et al., 1986).

4. Experiments

In our experiments, we calculate a plan for a kinematic laser scan survey for three large outdoor scenarios to demonstrate the applicability of our formulation. Moreover, we evaluate the effect of the two proposed candidate edge weightings on the resulting survey plan.

During our simulations, we assume a non-dynamic environment in which the mapping system follows exactly the predefined path while continuously performing 360° scans. Moreover, the input polygons correspond to the actual buildings.

The code we used to evaluate our algorithm is completely written in *Python*. For geometric operations, we utilized *shapely*[1], and to maintain and update the graph structures *networkx*[2]. As the solver for our MILP formulation, we applied *gurobi*[3] in version 12.0.3. For our experiments, we set the minimum measurement distance $d_{\min}$ to 2m, the maximum measurement distance $d_{\max}$ to 20m, and the minimum incidence angle α to 30°. The grid consists of nodes with a distance of 10m between them (l_{grid}), while evaluating the visibility every 1m (l_{eval}).

The results for all three sceneries and both weighting techniques are presented in Table 1. As expected, the length of the route computed with the geometric length as edge weights features shorter tours for the kinematic laser scanning survey. Moreover, fewer edges are selected compared to the case where the VRF-based weighting was applied. Although the tour is only 0.5%, 3.9%, and 8.7% longer, respectively, the mean VRF is notably higher for every scenario.

To examine the influence of the weighting, in Figure 4 we additionally show the VRF, calculated as proposed in Equation 4, for the scenery of our first example (`ex1`). It is evident that, as expected, the edges between two or more buildings feature a higher VRF (yellow color), since segments from each building can be observed. Moreover, around the buildings, mostly the edges that are a bit distant from the façade feature a higher VRF than the edges lying closer to the building, which can be attributed to the incidence angle constraint that restricts the observability of more distant edges when traversing an edge close to the building. In contrast, the more distant edges also feature a lower VRF due to the maximum measurement distance.

[1] `https://pypi.org/project/shapely/`
[2] `https://networkx.org/`
[3] `https://www.gurobi.com/`

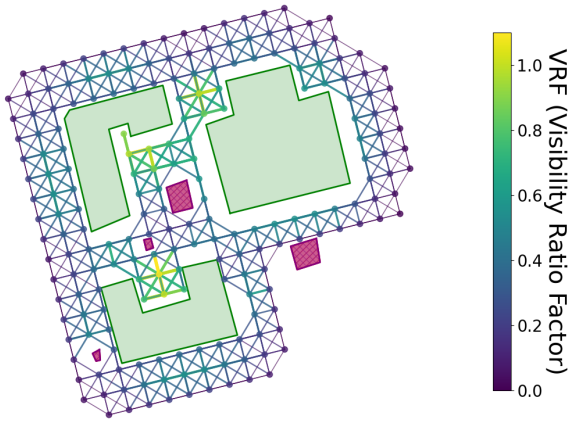


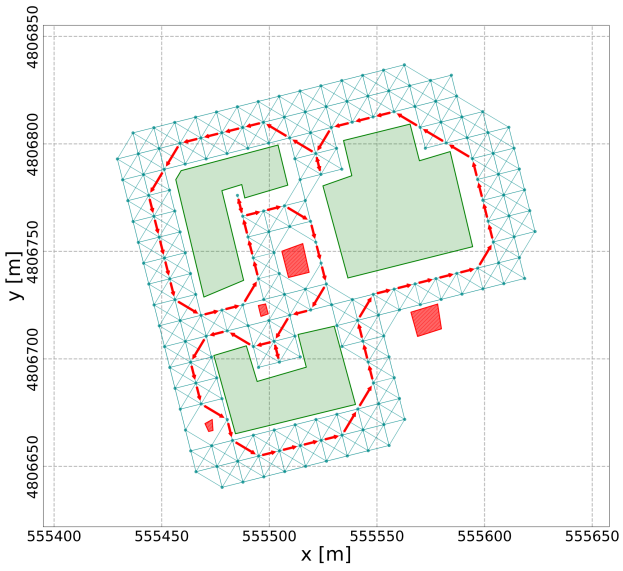
Figure 4. Heatmap illustrating the VRF for all candidate edges of the navigation graph on scenery `ex1`.

Consequently, the solution calculated with the VRF-based weighting, which is depicted in Figure 5(b), comprises these aforementioned edges with a medium distance to the building. In contrast, the length-based solution shown in Figure 5(a) features only the edges closer to the building, since they allow a shorter survey tour. Only on the upper left edge of the lower building, the length-based route does not utilize the navigation edges that are close to the building. This can be attributed to the application of the minimum measurement distance constraint.

In addition to the influence on the length of the resulting survey plan, the VRF also affects the time the solver needs to calculate the solution with an optimality guarantee. Most of our exemplary calculations were performed in minutes. However, the length-based solution for `ex1` took roughly one hour, and for `ex3` several hours. This is still reasonable in the context of offline planning, where the surveyor performs the planning before arriving at the scene, e.g., on the prior day. However, we observed that the optimal solution was already found after roughly 5 and 30 minutes, respectively. The remaining time was needed to guarantee the optimality of the solution. This process is more complex for the length-based weighting, since there are only two different weights (for the horizontal and vertical connections, as well as the diagonal ones). This leads to the existence of many solutions featuring the exact same route length that all need to be examined during the solving process. The VRF-based weights, however, are more diverse, easing the process of finding the optimal solution for the solver. This results in solving times of only a few minutes on our examples, while still featuring only slightly prolonged routes. However, also the other user-selectable parameters, such as l_{grid}, l_{seg}, or the scanner-related constraints, heavily influence the solving time,

https://doi.org/10.5194/isprs-annals-XI-2-2026-109-2026

(a)



(b)

Figure 5. Results for the first example scenery (`ex1`) using the geometric length as weight (a) and the VRF-based weighting (b) of the navigation edges in the objective function.

as they determine how many variables are set up, as well as how many constraints need to be imposed.

Our experiments show that applying the VRF-based weighting eases the solving process for the solver a lot, thereby making the application of our optimal algorithm even more attractive. However, up to now, we have not evaluated the influence of the weighting factor on the quality of the point clouds. In our experiments, we simulate the measurements in a 2D environment, which, obviously, does not allow for such an assessment. Simulating the measurement to obtain the resulting point cloud in software such as *Helios++* (Esmorís et al., 2022), or conducting a real-world survey based on the plans, should be future steps. In this context, it would be interesting to tune the VRF or even add more complex weighting factors for the candidate edges, for example, by considering the resolution of the respective observations on the object of interest.

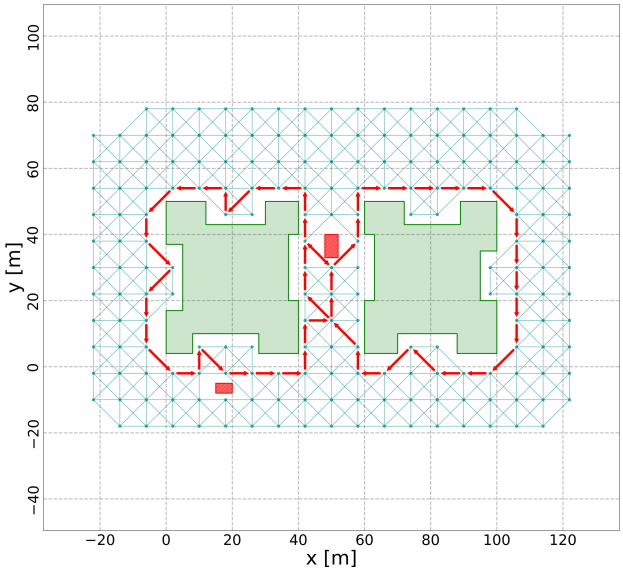


Figure 6. Result on `ex2` after introducing tie points in between the two buildings.

In Figure 6, we show the result of our optimization approach when introducing tie points that need to be traversed at least two times in between the two buildings comprising the scene. This leads to the tour passing through the gap between the buildings twice. In consequence, we obtain a loop closure to mitigate the effects of uncertainties in the estimation of the positions of the system that are directly propagated into the resulting point cloud.

Although our experiments demonstrate the general feasibility and applicability of our formulation, paving the way for the application of exact approaches for the planning of kinematic laser scanning, other factors influencing the quality of the resulting point cloud have not been taken into consideration, such as the quality of the GNSS position, or scanner-specific operational constraints (e.g., cooldown time of the scanner). Moreover, we restricted ourselves to two-dimensional scenarios, although especially for indoor scanning, the collection of multi-floor data poses a further challenge that needs to be addressed in future research.

5. Conclusion and Future Work

In this article, we propose a novel MILP-based formulation to calculate an optimal route for kinematic laser scanning. We guarantee full observability and consider sensor-related constraints that influence the quality of the acquired data. The user is additionally able to introduce tie points to the survey to mitigate the effects of uncertainties in the sensor position estimation that would directly propagate into the point cloud. Moreover, we propose a novel weight factor for the objective function, the *Visibility Ratio Factor* (VRF), which takes not only the length of the survey route into account, but also implicitly adds a certain amount of redundancy to the acquired data. In multiple experiments, we apply our formulation and the different weightings to different survey scenarios, proving the applicability of our formulation. The VRF additionally proves to be able to significantly speed up the solving time of the optimization process and features an increased amount of redundancy, while resulting in a route that is only slightly prolonged.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-XI-2-2026-109-2026 | © Author(s) 2026. CC BY 4.0 License.

To benefit from this property, we aim to assess in future work the influence of the VRF-based weighting, as well as even more sophisticated weighting techniques, e.g., based on the possible resolution of the observations, on the resulting point cloud. Moreover, introducing constraints to the ILP formulation in order to allow multiple subtours to satisfy battery constraints for large sceneries appears to be a promising enhancement. In a step further, optimizing the trajectories for multiple independent systems, similar to the approach presented by Zhang et al. (2023) on coordinated coverage path planning for UAV-ships, appears to be a promising research area.

References

Aryan, A., Bosché, F., Tang, P., 2021. Planning for Terrestrial Laser Scanning in Construction: A Review. *Automation in Construction*, 125, 103551.

Biggs, N., Lloyd, E. K., Wilson, R. J., 1986. *Graph Theory, 1736-1936*. Oxford University Press.

Biswas, H., Bosché, F., Sun, M., 2015. Planning for Scanning Using Building Information Models: A Novel Approach with Occlusion Handling. *Proceedings of the International Symposium on Automation and Robotics in Construction, ISARC*, 2015 Proceedings of the 32nd ISARC, Oulu, Finland, 1–8.

Chin, W.-P., Ntafos, S., 1988. Optimum Watchman Routes. *Information Processing Letters*, 28(1), 39-44.

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.

Di Stefano, F., Gorreja, A., Balestra, M., Pierdicca, R., 2021. Mobile 3D Scan LiDAR: A Literature Review. *Geomatics, Natural Hazards and Risk*, 12, 2387-2429.

Díaz Vilariño, L., Frías, E., Previtali, M., Scaioni, M., Balado Frias, J., 2019. Scan Planning Optimization for Outdoor Archaeological Sites. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W11, 489-494.

Esmorís, A. M., Yermo, M., Weiser, H., Winiwarter, L., Höfle, B., Rivera, F. F., 2022. Virtual LiDAR Simulation as a High Performance Computing Challenge: Toward HPC HELIOS++. *IEEE Access*, 10, 105052–105073.

Esser, F., Moraga, J., Klingbeil, L., Kuhlmann, H., 2022. Accuracy Improvement of Mobile Laser Scanning Point Clouds using Graph-based Trajectory Optimization. *Proceedings of the 5th Joint International Symposium on Deformation Monitoring - JISDM 2022*.

Fong, T. S., Yan, W. Y., 2025. Signs on Glasses: LiDAR Data Voids, Hotspot Effect, and Reflection Artifacts. *Automation in Construction*, 169, 105877.

Frías, E., Díaz Vilariño, L., Balado Frias, J., Lorenzo, H., 2019. From BIM to Scan Planning and Optimization for Construction Control. *Remote Sensing*, 11, 1963.

Frías, E., Previtali, M., Díaz-Vilariño, L., Scaioni, M., Lorenzo, H., 2022. Optimal Scan Planning for Surveying Large Sites with Static and Mobile Mapping Systems. *ISPRS Journal of Photogrammetry and Remote Sensing*, 192, 13-32.

Grisetti, G., Kümmerle, R., Stachniss, C., Burgard, W., 2025. An Updated Tutorial on Graph-Based SLAM. B. Siciliano (ed.), *Robotics Goes MOOC: Knowledge*, Springer Nature Switzerland, Cham, 255–285.

Jia, F., Lichti, D., 2019. A Model-based Design System for Terrestrial Laser Scanning Networks in Complex Sites. *Remote Sensing*, 11(15), 1749.

Kim, M.-K., Wang, Q., Li, H., 2019. Non-contact Sensing Based Geometric Quality Assessment of Buildings and Civil Structures: A Review. *Automation in Construction*, 100, 163–179.

Knechtel, J., Dehbi, Y., Klingbeil, L., Haunert, J.-H., 2025. Simultaneous planning of standpoints and routing for laser scanning of buildings with network redundancy. *ISPRS Journal of Photogrammetry and Remote Sensing*, 224, 59-74.

Knechtel, J., Klingbeil, L., Haunert, J.-H., Dehbi, Y., 2022. Optimal Position and Path Planning for Stop-and-go Laserscanning for the Acquisition of 3D Building Models. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, V-4-2022, 129–136.

Kordgholiabad, M., 2025. Optimal Path Planning for Kinematic Laser Scanning. Master’s thesis, Politecnico Milano.

Kuželka, K., 2025. Trajectory Design for Handheld Mobile Laser Scanning in Complex Natural Forests: A Simulation Approach. *ISPRS Journal of Photogrammetry and Remote Sensing*, 230, 524-546.

Lin, Y., Hyyppä, J., Kukko, A., 2013. Stop-and-go Mode: Sensor Manipulation as Essential as Sensor Development in Terrestrial Laser Scanning. *Sensors*, 13(7), 8140–8154.

Meyer, T., Brunn, A., Stilla, U., 2022. Change Detection for Indoor Construction Progress Monitoring based on BIM, Point Clouds and Uncertainties. *Automation in Construction*, 141, 104442.

Navarro, A. J., 2023. Sensor-aligned Coverage Planning for Mobile Robots in Complex 3D Environments. PhD thesis, The University of Texas at Austin.

Noichl, F., Lichti, D., Borrmann, A., 2024. Automating Adaptive Scan Planning for Static Laser Scanning in Complex 3D Environments. *Automation in Construction*, 165, 105511.

Papadimitriou, C. H., 1977. The Euclidean Travelling Salesman Problem is NP-complete. *Theoretical Computer Science*, 4(3), 237-244.

Poku-Agyemang, K. N., Reiterer, A., 2025. Model-based Planning of Complex 3D Laser Scanning Campaigns for Bridge Digitisation. *Automation in Construction*, 177, 106289.

Shirabe, T., 2004. A Model of Contiguity for Spatial Unit Allocation. *Geographical Analysis*, 37, 2-16.

Soudarissanane, S., Lindenbergh, R., Menenti, M., Teunissen, P., 2011. Scanning Geometry: Influencing Factor on the Quality of Terrestrial Laser Scanning Points. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(4), 389–399.

Zhang, X., Zhang, F., Tang, Z., Chen, X., 2023. A MILP Model on Coordinated Coverage Path Planning System for UAV-ship Hybrid Team Scheduling Software. *Journal of Systems and Software*, 206, 111854.