

Simultaneous planning of standpoints and routing for laser scanning of buildings with network redundancy

Julius Knechtel ^a, Youness Dehbi ^b, Lasse Klingbeil ^a, Jan-Henrik Haunert ^a

^a University of Bonn, Institute for Geodesy and Geoinformation, Meckenheimer Allee 172, 53115, Bonn, Germany

^b HafenCity University, Computational Methods Lab, Henning-Voscherau-Platz 1, 20457, Hamburg, Germany

ARTICLE INFO

Keywords:

Laser scanning
Network survivability
Optimization
Redundancy
Mixed Integer Linear Programming
Route planning

ABSTRACT

Stop-and-go laser scanning is becoming increasingly prevalent in a variety of applications, e.g., the survey of the built environment. For this, a surveyor needs to select a set of standpoints as well as the route between them. This choice, however, has a high impact on both the economic efficiency of the respective survey as well as the completeness, accuracy, and subsequent registrability of the resulting point cloud.

Assuming a set of building footprints as input, this article proposes a one-step optimization method to find the minimal number of selected standpoints based on scanner-related constraints. At the same time, we incorporate the length of the shortest route connecting the standpoints in the objective function. A local search method to speed up the time for solving the corresponding Mixed-Integer Linear Program (MILP) is additionally presented. The results for different scenarios show constantly shorter routes in comparison to existing approaches while still maintaining the minimal number of standpoints.

Moreover, in our formulation we aim to minimize the effects of inaccuracies in the software-based registration. Inspired by the ideas of network survivability, we to this end propose a novel definition of connectivity tailored for laser scanning networks. On this basis, we enforce redundancy for the registration network of the survey. To prove the applicability of our formulation, we applied it to a large real-world scenario.

This paves the way for the future use of fully automatic autonomous systems to provide a complete and high-quality model of the underlying building scenery.

1. Introduction

In the context of the increasing digitization of the construction industry and the emergence of manifold digital methods, the retrieval of accurate as-built models of already existing objects, e.g., for Building Information Models (BIMs), is an important and frequently performed task. For example, during the construction phase of a building, an as-is BIM can be obtained in order to document the progress and to allow a comparison with an as-planned model (Meyer et al., 2022). Although the process of generating such a BIM from sensor data is already highly automated, the previous data capturing procedure is not. Often the required data is captured by a terrestrial laser scanner, e.g., using static or stop-and-go laser scanning. The planning of the standpoints where the scanner is placed and of the route to be followed with the scanner to observe the respective buildings is a largely manual process, which is often performed directly on-site by an expert, i.e., a surveyor. The purely visual assessment of visibility as well as assumptions about the

resulting point cloud quality, however, are contingent on a great degree of experience and are prone to errors (Dehbi et al., 2021). Additionally, a robot which can autonomously carry out a previously planned stop-and-go laser scanning survey could possibly facilitate more frequent observations during the building process without a significant increase in costs. To enable such a procedure and to mitigate the risks associated with the purely visual assessments of the scene, it appears to be desirable to employ automatic planning algorithms.

In general, offline planning for the stop-and-go survey of buildings consists of two parts, i.e., choosing (1) the number and the corresponding positions of the scanner standpoints before the scanning process as well as (2) the shortest route between them while not being on-site. There exist approaches for both respective tasks, but most of them rely on heuristics, i.e., they are unable to provide an optimal result. Additionally, the two tasks of choosing the standpoints and the route are performed subsequently and, thus, the selection of the standpoints

* Corresponding author.

E-mail addresses: knechtel@igg.uni-bonn.de (J. Knechtel), youness.dehbi@hcu-hamburg.de (Y. Dehbi), klingbeil@igg.uni-bonn.de (L. Klingbeil), haunert@igg.uni-bonn.de (J.-H. Haunert).

<https://doi.org/10.1016/j.isprsjprs.2025.03.017>

Received 20 December 2024; Received in revised form 13 March 2025; Accepted 17 March 2025

Available online 5 April 2025

0924-2716/© 2025 The Authors. Published by Elsevier B.V. on behalf of International Society for Photogrammetry and Remote Sensing, Inc. (ISPRS). This is an open access article under the CC BY-NC license (<http://creativecommons.org/licenses/by-nc/4.0/>).

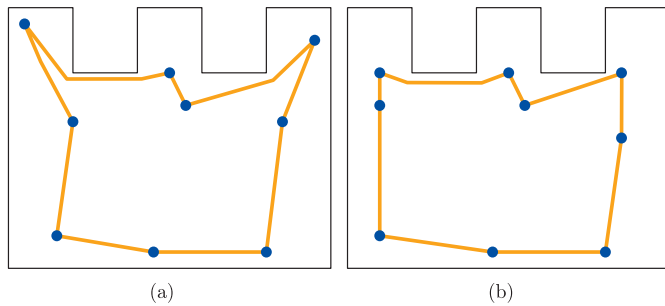


Fig. 1. Results of the two-step approach from Knechtel et al. (2022) (a) compared to our one-step optimization (b). Both yield full coverage and admit a subsequent software-based registration using the same number of standpoints. The route in (b) is, however, substantially shorter.

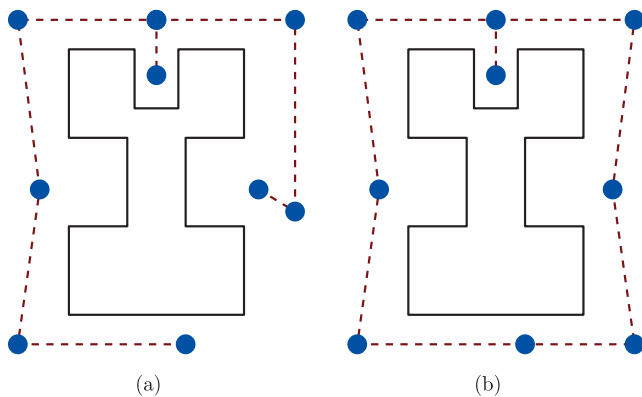


Fig. 2. The result of the one-step optimization without (a) and with (b) additional connectivity constraints. The red-dashed lines represent the edges from the registrability graph G . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

is not influenced by the length of the corresponding resulting shortest route between them. Since the selection of standpoints is mostly not unambiguous, i.e., there exist multiple sets of standpoints of equal size, considering both tasks at the same time to find the set comprising the shortest route seems to be an enticing task. Hence, the first contribution of this article is to present a formulation solving both problems in one step. The benefit is depicted in Fig. 1. On the left we show a result of the approach from Knechtel et al. (2022). The optimal set of standpoints is computed using Integer Linear Programming (ILP) followed by a subsequent calculation of the optimal route using Mixed Integer Linear Programming (MILP). In the following, we refer to this approach as TwoStep formulation. In contrast, on the right the result after a combined optimization as proposed in our article is shown. Both solutions fulfill all constraints of the standpoint planning and the number of standpoints is equal. However, the result of the combined approach requires substantially less traveling between the corresponding standpoints.

Since solving ILPs can be time-consuming, we present as our second contribution a heuristic based on a local search which is able to speed up the computational process. Furthermore, we thoroughly test our formulation and the heuristic on different building outlines and compare the results and running times of the different approaches.

The third contribution of this article is that, for the first time, we incorporate ideas from the concept of network survivability into the planning of laser scanner networks. In this context, we present the possibility of adding constraints to our problem formulation to enhance the connectivity of the registration network, i.e., the network indicating which single scans can be registered together. In general, the term

network survivability refers to the ability of an arbitrary network to maintain functional in spite of failures in the network (Abbas, 2006; Kuipers, 2012) and is often used in the context of communication networks. This concept, however, can also be applied in a survey context in which often the presence of redundancy is desired to be able to compensate for measurement inaccuracies or unexpected obstructions. This holds especially for the registration process. Previous methods already consider a simple software or target-based registration. However, when executing a laser scanning survey assumptions made in the planning process about the registrability of two standpoints may be violated, e.g., due to unforeseen circumstances which constrain the visibility. Thus, the overlap of the point clouds may be smaller than expected. This can lead to inaccuracies when estimating the registration parameters, which can in turn influence all subsequent registrations. In the worst case it can even cause a failed registration.

Fig. 2(a) shows the result of a combined optimization for an exemplary boundary from a building that should be surveyed from the exterior. The dashed red lines indicate that based on the planning a software-based registration is possible between the particular standpoints. In this case, an inaccuracy in one registration cannot be detected as it is not controlled by any other registration. Fig. 2(b) was generated using our additional constraint and in contrast shows a more desirable configuration because a closed exterior loop of standpoints exists. Here, redundancy for the software-based registration is present since the point clouds of each pair of scanner positions on this exterior loop can be registered together by following two distinct paths. Consequently, one registration can fail but a registration of all scans still remains possible. The registration network therefore has a higher survivability. However, to avoid a from a practical perspective unnecessary measurement overhead, our formulation is relaxed to a certain degree to still allow individual scanner locations to be easily attached to the remaining network as depicted at the recess at the top of the building. Since no common definition of connectivity fulfills these demands we propose a new definition, i.e., after-pruning- k -edge-connectivity. To the best of our knowledge, this is the first ILP formulation which enforces a stronger connectivity of the underlying network but explicitly allows the existence of pendant vertices. We additionally test our constraint by planning and executing the survey of a large real-world scenario and evaluating the resulting point cloud.

The remainder of this paper is structured as follows: Section 2 provides an overview of relevant articles dealing with the topics of survey planning for laser scanning and network survivability. In Section 3 we introduce our new ILP formulation and the constraints we applied to solve the problem 3.1 as well as a heuristic based on local search 3.2. Subsequently, in Section 3.3 we provide a suitable definition of connectivity as well as an overview of the additional constraints to guarantee an improved level of redundancy in the scanning network. In Section 4 we present our experiments, in which we compare our different methods on multiple scenarios 4.1 and show our survey and the resulting point cloud on a large real-world scenario 4.2. Section 5 summarizes our article and offers an outlook on future work.

2. Related work

The problem at hand can be linked to a well-studied problem from the field of computational geometry. The so-called Art Gallery Problem (AGP) was introduced by the mathematician Victor Klee in 1973. Lee and Lin (1986) proved that this problem is NP-hard. That means that it is highly improbable to find an algorithm to solve this problem optimally and in an efficient way, i.e., in polynomial time. Hence, it is reasonable to either apply a heuristic, which may not yield an optimal solution, or to employ Integer Linear Programming (ILP) as an exact algorithm. The aim is to compute the smallest possible set of guards observing all parts of an art gallery. A distinction can be made between vertex, edge, and point guards, which allow placement in the corners of the polygon, on the edges of the polygon, and freely in space,

Table 1

Overview on different related publications and their used inputs, constraints, and methods. SC describes if scanner constraints are considered, Reg if the registration is part of the planning.

Publication	Standpoint calculation					Route calculation	
	Input	Candidates	SC	Reg	Method	Method	Steps
Soudarissanane and Lindenberg (2012)	2D	2D Grid	x		Greedy		
Biswas et al. (2015)	3D	2D Grid	x		Integer Programming		
Ahn and Wohn (2016)	2D	2D Grid	x	x	Interactive Greedy		
Díaz-Vilariño et al. (2019)	2D	2D Grid/Triangulation	(x)		Greedy		
Jia and Lichti (2019)	2D	2D Grid	x	x	Weighted Greedy		
Dehbi et al. (2021)	2D	2D Grid	x	x	Integer Programming		
Frías et al. (2019)	2D	2D Grid/Triangulation	(x)		Greedy	Probabilistic	2
Knechtel et al. (2022)	2D	2D Grid	x	x	Integer Programming	Integer Programming	2
Noichl et al. (2024)	3D	2D Grid/ Poisson Disk Sampling	x	x	Greedy/ Genetic Algorithm	Approximation	2

respectively. Additional restrictions can be applied to the geometric structure, e.g., that only orthogonal polygons are accepted, or if holes in the polygon are allowed. Different combinations of restrictions lead to different bounds for the necessary number of guards that can be proven, e.g., by O'Rourke (1987). For example, in the case of vertex guards and assuming an orthogonal polygon without holes this problem is optimally solvable by addressing the AGP as a Set Cover Problem and using Integer Programming to solve the problem formulation (Couto et al., 2011). The AGP is closely related to the problem at hand, i.e., to find the minimal set of laser scanner positions to conduct a complete survey. In this context, a laser scanner needs to be placed as a point guard, i.e., the scanner can be placed anywhere inside the gallery. Additionally, we aim for no restrictions on the geometry of the building as well as to allow holes in the polygon to cover all realistic building polygons. This consequently corresponds to the most unrestricted case of the AGP. For this case, Kröller et al. (2012) presented a method to compute the lower bound for the minimum number of guards for general art gallery problems by applying a primal–dual algorithm based on linear programming.

In the art gallery problem, the vision of the guards is unrestricted. Yet, in the context of laser scan planning, this assumption is not valid, since the quality of the resulting point cloud depends on different factors. For each scanner, the manufacturer provides a minimum and maximum measurement distance. From the literature additionally a serviceable incidence angle can be retrieved based on its influence on the measurement accuracy (Soudarissanane et al., 2011). When exceeding one or more of these values, the quality of the resulting point cloud is degrading. Thus, these scanning-related constraints which limit the visibility need to be additionally incorporated.

A variety of approaches dealing with the selection of laser scanner positions exists. In this context, a distinction can be made between online and offline planning, i.e., whether the planning process is undertaken on-site or not. Additionally, it is possible to distinguish between model-based and non-model-based approaches, i.e., whether the algorithm is based on some input describing the underlying scenery. In this article, we focus on model-based approaches for offline planning. They differ with regard to

1. their input, i.e., whether it is a 2D or 3D input structure,
2. the constraints which are applied, e.g., the minimum and maximum range, the minimal incidence angle and a possible subsequent registrability, and
3. their method, i.e., if it is a (greedy) heuristic or exact algorithm and how the candidates for the scanner positions are generated, e.g., using a grid or a triangulation-based approach.

An exhaustive overview of different approaches and their respective characteristics as well as different categorizations are presented in the review article by Aryan et al. (2021). The approaches presented in this section are additionally listed in Table 1 to provide a concise overview of the differences with regard to the above-mentioned categories.

Most of the previous approaches focus on greedy methods. Soudarissanane and Lindenberg (2012), for example, incorporate range and angle constraints by considering the visibility of subsegments of walls from given standpoint candidates. The authors subsequently choose greedily the next best candidate, which covers most of the yet unobserved segments. An optimal approach was presented by Biswas et al. (2015) who employed Integer Linear Programming to find the minimum number of standpoints given an already existing 3D BIM model while incorporating sensor constraints. Nevertheless, the approach suffers from the fact that there is no guaranteed overlap between the point clouds. Specifically for archaeological sites, which are often characterized by their large size, Díaz-Vilariño et al. (2019) provide a triangulation-based system for the generation of candidate positions to obtain faster computation times for such large areas.

Alongside the sensor constraints, the subsequent registration of the individual scans also plays an important role in position planning for laser scanners. With a special focus on heritage surveying, Ahn and Wohn (2016) consider the overlap between two scans as an additional parameter. However, a user intervention in the planning process is needed, i.e., to select the position based on proposals. For a target-based registration procedure, Jia and Lichti (2019) again proposed a greedy, hierarchical strategy. They aim for minimizing the number of registration targets that need to be placed in the environment, since the placement is a further time-consuming working step and, hence, cost-intensive. In order to economize on this step, software-based methods can be employed, which combine a coarse and a fine registration. The coarse step is often performed by utilizing geometric structures in the measurement object, i.e., planar patches (Brenner et al., 2008) or the intersection of planes (Theiler and Schindler, 2012). For the subsequent fine registration, well-known algorithms, e.g., Iterative Closest Points (ICP, Besl and McKay 1992)) or Random Sample Consensus (RANSAC, Fishler (1981)), can be employed. Cheng et al. (2018) offer an in-depth overview on state-of-the art registration techniques for point clouds.

Dehbi et al. (2021) presented an approach to compute a minimal set of standpoints for static laser scanning while considering the aforementioned scanner-related constraints and additionally guaranteeing the feasibility of a subsequent software-based registration. The optimality is restricted to the use of a set of candidate positions. Assuming a software-based registration, the standpoints selected for the survey form a network, which is represented by a registrability graph. The corresponding registrability edges determine whether for two particular standpoints a registration can be performed, i.e., the overlap of the point clouds as well as the distribution of the normals corresponding to the overlapping objects are satisfying. In this context, it is also important to consider the robustness of this network, e.g., that the standpoints not only form a connected graph but the network is also resilient against failures. This property is in graph theory also known as network survivability, i.e., the ability of a network to maintain operation when one or more components fail, e.g., in our case the registration of two particular scanner positions. In the geodetic context,

this is known as redundancy. An overview of the general problem and algorithms to solve different survivability-related problems was given by Kuipers (2012). Different approaches exist for planning and evaluating models for power grids or fiber optic networks (e.g., Heegaard and Trivedi (2009)). In some publications, the Menger's theorem from mathematical graph theory has been exploited, e.g., to enforce region-based connectivity to gain fault-tolerant networks (Sen et al., 2009). However, to the best of our knowledge, network survivability has not been explicitly addressed in the context of laser scan planning.

It is noticeable that the approaches from the aforementioned publications focus on static laser scanning, although it poses some disadvantages: As stated in a comparative study by Lin et al. (2013) it is time-consuming and, hence, economically expensive. The common alternative is to use kinematic laser scanning, which is remarkably faster. The quality of the resulting point cloud, however, is often worse. This can be attributed to uncertainties in the referencing of the respective laser scanner positions, which is usually performed based on a combination of GNSS and Inertial Measurement Unit (IMU) measurements. The GNSS is more stable in the long-term, whereas the IMU, which usually combines measurements of accelerometers and gyroscopes, has a higher short-term accuracy but is prone to drift. The error in the resulting positions is directly propagated into the final point cloud. Hence, the authors suggest to use stop-and-go laser scanning, which allows for combining the strengths of both paradigms but compensates for the particular weaknesses (Lin et al., 2013).

The planning process for a stop-and-go laser scanning survey also requires the calculation of a suitable sequence of standpoints. For this, again different techniques can be applied, e.g., a probabilistic ant colony algorithm (Frías et al., 2019), which results in a heuristically determined path. From a more general point of view, the problem can be seen as an instance of the Traveling Salesperson Problem (TSP). This is a combinatorial optimization problem to find the sequence of cities a traveling salesperson needs to visit such that the distance traveled is as short as possible. At the same time, every city must not be visited more than once and the start and end point are identical. Although such a round tour is not always needed for stop-and-go laser scanning, not only in the context of autonomous or semi-autonomous robots this is often desired in order to return to the area where the additional equipment was kept. This problem is also proven to be NP-hard (Cormen et al., 2009). This property again justifies the use of a Mixed Integer Linear Program (MILP) to retrieve an optimal solution or non-exact heuristics to faster obtain a solution. Bolourian and Hammad (2020), for example, solve the TSP for bridge monitoring and damage assessment based on Uncrewed Aerial Vehicles (UAV) equipped with LiDAR. An in-depth overview of path planning in the UAV context is provided by Khoufi et al. (2019).

Recently, Noichl et al. (2024) published a set of different heuristic approaches, based on a greedy forward, backward, or oscillating search. Additionally, a genetic algorithm inspired by evolution theory and based on randomized selection is employed. The authors perform a subsequent sequence planning based on Christofides algorithm (Christofides, 1976), which yields an approximate solution for the TSP. Although this approach poses the possibility to handle both global or local coverage requirements, in this publication again no optimality can be guaranteed for both the number of selected standpoints and the subsequent routing. (Knechtel et al., 2022) apply the ILP formulation by Dantzig et al. (1954) to optimally solve the TSP and retrieve the shortest route between the set of precomputed standpoints. However, since the optimal route calculation is only performed after the optimal standpoints have been determined, its optimality is only guaranteed for this specific set of standpoints. Yet, the selection of the standpoint set is often ambiguous as neighboring standpoint candidates often offer the same characteristics with regard to the optimization criteria. It follows that the solution path of the applied ILP solver, which is influenced by a number of different, partly random parameters, ultimately determines which standpoints are selected. However, other

combinations of standpoints of equal size which possibly comprise a shorter route remain unexplored.

In conclusion, the existing approaches are deficient in at least one of three aspects: First, the registrability of the scans is not considered, or at least without redundancy. Second, the solutions are only greedy and not optimal. Third, the calculation of points and routes, if available, is carried out separately. The main contribution of this paper to overcome the previously elaborated research gap is threefold: (1) we develop a formulation for the combined optimization of both problems. This represents a baseline to evaluate the (2) additionally introduced local search method, as well as possible future heuristics. Lastly, we (3) explicitly incorporate and address ideas from the concept of network survivability to introduce a suitable level of redundancy for the software-based registration. At the same time, we still allow certain relaxed topologies, that can be frequently found in practice and are important to ensure an economic efficacy of the survey by avoiding large measurement overheads.

3. Methodology

The following section is divided into three parts. First, in Section 3.1 our one-step formulation for the calculation of an optimal stop-and-go laser scanning survey which employs a simple connectivity formulation to allow for a software-based registration is presented. Thereafter, we introduce our heuristic based on local search to enable a speed-up of the calculation process (Section 3.2). In Section 3.3, we present a method to enhance the connectivity, i.e., topology, of the underlying registration network by enforcing redundancy. For this, we first elaborate a suitable definition of connectivity in the context of laser scanning and afterwards present corresponding constraints for the ILP.

3.1. Combined standpoint and route calculation

For our combined approach to find a solution with the minimal route length among all possible solutions with the minimal number of standpoints we extend the TwoStep formulation from Knechtel et al. (2022). From now on we will refer to our formulation as the OneStep approach. The optimality is as in the preceding approaches restricted to the use of a set of candidate positions.

Calculating the standpoints and route for an optimal stop-and-go laser scan planning in one step still comprises two different objective functions. First, the number of standpoints needs to be as small as possible, this objective is from now on considered as $O_{\text{Standpoints}}$. Second, the length of the shortest route connecting the standpoints, which is described by O_{Route} , needs to be as short as possible. Generally spoken, we aim to

$$\underset{l \in L}{\text{Minimize}}(O_{\text{Standpoints}}(l), O_{\text{Route}}(l)) \quad (1)$$

where L is the feasible set of decision vectors. Obviously, both objectives could be jointly considered, i.e., blended via a linear combination utilizing a weight parameter λ . The application of such a linear combination would be easily employable in our implementation. However, the parameter λ needs to be tuned to obtain the desired results, which can be time-consuming due to the runtime of the ILP and, moreover, is dependent on the specific scenery. Additionally, minimizing the number of standpoints is in our context, i.e., the retrieval of highly accurate building models, more important due to the high time consumption associated with additional standpoints when retrieving a reasonable dense and accurate point cloud (Wujanz et al., 2016), whereas a slightly longer route is less harmful to the duration of the survey. Furthermore, each additional standpoint necessitates more work in the postprocessing of the point cloud, for example, the registration, which also increases the cost of this step. Hence, applying a hierarchical objective function in the optimization seems to be a promising approach to resolve the ambiguities. In this context, we

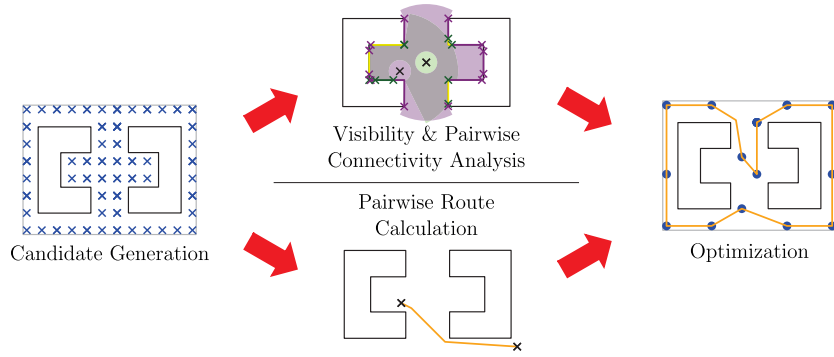


Fig. 3. Schematic depiction of our OneStep algorithm. In the preprocessing, candidates are generated and subsequently analyzed with regard to visibility and routing. The core of our approach is the optimization step using Mixed Integer Linear Programming (MILP).

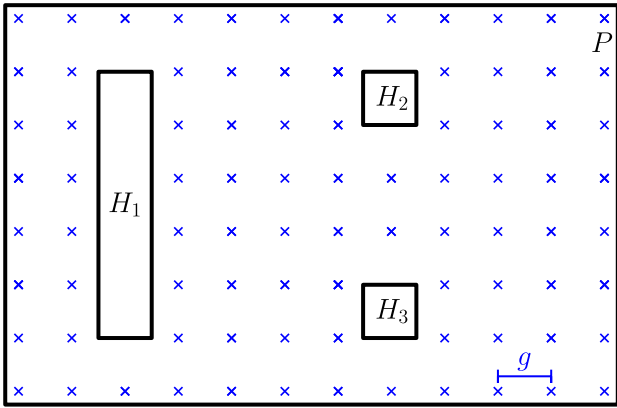


Fig. 4. Exemplary indoor scenery for the proposed algorithms. The blue crosses depict the grid of candidate points, i.e., possible laser scanner positions. The exterior wall, i.e., the exterior ring of P , as well as the interior building parts which are modeled as holes H_1, H_2 and H_3 are marked in black. This indicates that they comprise a continuous set $\mathcal{R} \subseteq \partial P$ of points which need to be observed. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

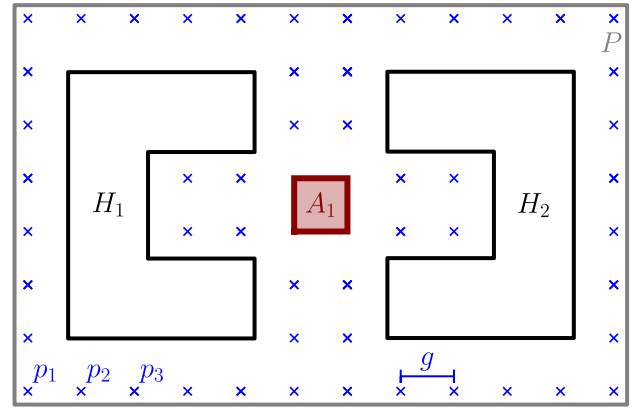


Fig. 5. Exemplary outdoor scenery which serves as the input for the proposed algorithms. The blue crosses depict the grid of candidate points. The red polygon A_1 represents a non-viable area prohibiting the introduction of standpoints. H_1 and H_2 correspond to buildings and are marked in black, i.e., they define the continuous set $\mathcal{R}$ of points which need to be observed. The outer ring of P is marked gray and is not part of $\mathcal{R}$, since it does not correspond to an object but is a user-selected boundary for the scenery. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

assign that the objective function connected to the viewpoint minimization, i.e., $O_{\text{Standpoints}}$, needs to dominate the objective function for the route planning (O_{Route}), i.e., $O_{\text{Standpoints}}$ has a higher priority. Consequently, the solver finds among all solutions comprising the minimal number of standpoints a solution with the minimal route length. In other words, given the result of our formulation, there exists no combination of standpoints which feature a shorter route without degrading the solution for the viewpoint minimization, i.e., by introducing an additional standpoint.

It is evident that this joint optimization of two NP-hard problems in one step is computationally substantially more expensive than finding solutions for both problems subsequently, but at the same time the quality of the results can be improved substantially as well. This trade-off will be discussed in detail in Section 4. In the following subsections, first the input is described as well as some notations are introduced. Thereafter we present the variables and constraints of the formulation, clustered into the different respective tasks. The general pipeline of the preprocessing which defines the variables and constraints as well as the subsequent optimization of our OneStep approach is depicted in Fig. 3.

3.1.1. Input and candidate generation

The scenery that needs to be surveyed can be described by a 2D polygon P with holes, to which a 3D scenery can be safely reduced (Dehbi et al., 2021). This polygon could be created by the user or retrieved

from other sources, e.g., *OpenStreetMap*. In an indoor scenario (cf. Fig. 4), the outer ring of P is directly given by the exterior walls of the building, while the set of holes H describes the interior building elements, e.g., walls and columns. These holes define the inner rings of P . Additionally, $\mathcal{R} \subseteq \partial P$ describes a continuous set of points, which need to be observed. In the indoor scenario, this corresponds to the complete boundary ∂P of the polygon describing the building, i.e., the inner rings as well as the outer ring. In the outdoor scenario as exemplarily depicted in Fig. 5, however, $\mathcal{R}$ only comprises the inner rings of P , since the outer ring is only a user-defined boundary which does not correspond to any object. The holes $H = H_1, \dots, H_n$ describe in this context the n buildings comprising the scenery and correspond to the inner rings of P . Additionally, for both scenarios another set of polygons $A = A_1, \dots, A_m$ is introduced to describe areas which are restricted, e.g., not viable for the laser scanner. All polygons in A are contained in the interior of P . Hence, $P_{\text{candidate}} = P \setminus \{A_1 \cup \dots \cup A_m\}$ is again a polygon with holes.

Given this polygon, a set CP of candidate positions is subsequently instantiated as a discrete grid in $P_{\text{candidate}}$, which is shown in Figs. 4 and 5 in blue color. In this context, the grid width g is a crucial parameter, as it on the one hand determines the density of the candidate positions, but on the other hand also has a substantial impact on the runtime of the approach. An Integer Linear Program (ILP) is employed to calculate an optimal subset of these standpoints with the variables

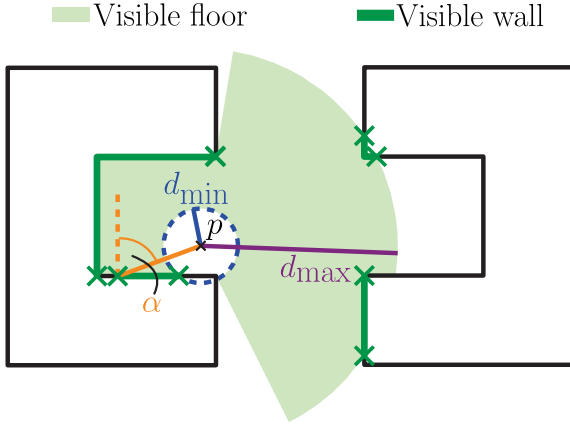


Fig. 6. Visibility polygon based on a position p . Scanner-related constraints, i.e., minimal ($d_{\min}$) and maximal range ($d_{\max}$), as well as the incidence angle α and obstacles blocking the view are considered. The crosses depict critical points, i.e., points which delineate the visible parts of the walls of the underlying building from non-visible parts.

$x_p \in \{0, 1\} \quad \forall \quad p \in CP$, which can be interpreted as follows:

$$x_p = \begin{cases} 1 & , \text{ if } p \text{ is selected as a scanner position} \\ 0 & , \text{ otherwise} \end{cases} \quad (2)$$

Since we aim to find a subset which is as small as possible but admits an appropriate survey of the scenery, the first objective is to minimize

$$O_{\text{Standpoints}} = \sum_{p \in CP} x_p. \quad (3)$$

This objective is one part of the hierarchical objective function (cf. Eq. (1)). In most cases the selection of the standpoints is unconstrained, however, in some cases the selection of specific standpoints can be useful. The selection of any standpoint p_f can therefore be enforced by adding a constraint that sets the corresponding variable to 1 ($x_f = 1$).

3.1.2. Visibility constraints

Additional constraints need to be added to our model regarding the surveying capabilities of the respective laser scanner. For each candidate point $p \in CP$ a visibility polygon $\mathcal{V}(p)$ can be calculated. For this, we take different parameters into account which highly influence the quality of the resulting point cloud: The minimum measuring distance $d_{\min}$, the maximum measuring distance $d_{\max}$, and the maximum incidence angle α . All parameters as well as an exemplary visibility polygon are illustrated in Fig. 6. The visible parts of the walls of the buildings that constitute the scenery that need to be observed, i.e., $\mathcal{V}(p) \cap \mathcal{R}$, are depicted separately.

The most important aim of a laser scanning survey is to observe all parts of all walls, which are as previously stated represented by a continuous set of wall points $\mathcal{R} \subseteq \partial B$, from at least one of the candidate positions $p \in CP$. To limit the computational complexity without introducing a discretization of the scenery, which could potentially cause some parts of the wall to be invisible, we compute a set of critical points (Dehbi et al., 2021). For each $p \in CP$, the critical points delineate the visible and non-visible parts of the boundary ∂P . The critical points for one viewpoint are exemplarily depicted in Fig. 6. When computing the set of all critical points for all standpoint candidates, ∂P and consequently $\mathcal{R}$ are split by this set into different segments. This is exemplarily shown for two candidate positions in Fig. 7. The critical points from both positions divide the boundary of the building polygons. The corresponding segments differ from one another in terms of the viewpoints from which they can be observed. In this example, all blue segments are visible from both standpoints, green segments are visible only from p_1 and red segments only from p_2 . Thus, for each of these visibility segments also holds, that any given point $r \subseteq \mathcal{R}$ in the

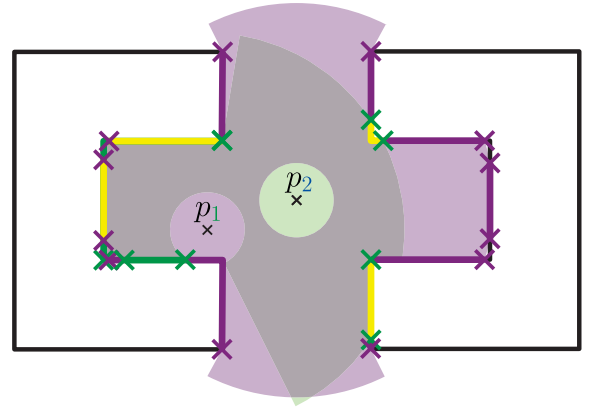


Fig. 7. Visibility polygons for two positions p_1 and p_2 as well as their critical points (colored crosses). The critical points define segments on the boundary of the buildings ∂P , which define the continuous set $\mathcal{R}$ of points which need to be observed. The segments are colored with respect to the standpoints which are able to observe the segment, i.e., red (observable from p_2), green (observable from p_1), or blue (observable from both points). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

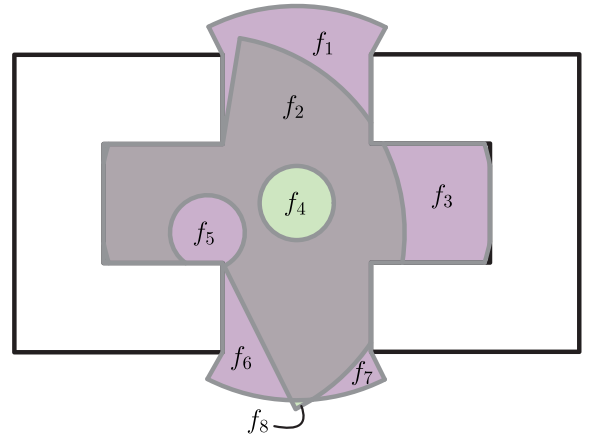


Fig. 8. Faces F of the arrangement defined by the two visibility polygons from Fig. 7.

respective segment is visible from the same set of candidate points. It is therefore sufficient to define a constraint only for one representative point r for each of the segments, which guarantees the visibility for the continuous set of all points on the particular segment. The discrete point set $R \subseteq \mathcal{R}$ consequently comprises one representative point for each segment. The following constraint is formulated to ensure that each representative point $r \in R$ is visible from at least one candidate point p , i.e., r must lie in at least one visibility polygon $\mathcal{V}(p)$ of a selected standpoint:

$$\sum_{p \in \{q \in CP \mid r \in \mathcal{V}(q)\}} x_p \geq 1 \quad \forall \quad r \in R \quad (4)$$

If additionally the floor needs to be observed, a similar approach can be used. Fig. 8 shows the arrangement which originates from the two visibility polygons. For each face $f \in F$ of the arrangement it holds, that each point in this particular face is visible from the same set of candidate points. Therefore, it is again sufficient to add a constraint only for one representative point of each face. For more details on the calculation of representative points, the interested reader is referred to Dehbi et al. (2021).

3.1.3. Software-based registration

The second aim is to ensure that a subsequent software-based registration of the scan s can be performed successfully. In this subsection,

we will first enforce a basic connectivity constraint based on the ILP formulation by Dehbi et al. (2021) ensuring that a registration is generally possible, an improved constraint will be presented in Section 3.3. Since in the end all standpoints need to be registered together, the problem has to be considered globally with the help of a registrability graph $G = (V, E)$. The candidate points serve as nodes, i.e., $V = CP$, and a registrability edge between two nodes exists if a registration of the two point clouds from the particular standpoints is possible. We assume that this graph is computable before the optimization process. In general, there exist various ways of assessing whether two scans can be registered software-based. When employing an ICP algorithm it is expected that a specific portion of both point clouds will cover the same objects, thereby ensuring a certain degree of overlap between the two datasets. Additionally, the distribution of normals can be considered to ensure that all registration parameters can be estimated.

For our experiments, the registrability is defined only by the overlap of the particular scans. However, other metrics to define registrability could be included in the workflow. Such an overlap may be observed both on the floor and on the walls of the buildings in question. We define $A_{\min}$ as the minimum size for the overlapping parts of the floor in the visibility polygon of two standpoints p and q with $A_{pq} = \text{area}(\mathcal{V}(p) \cap \mathcal{V}(q))$. Likewise, taking into account the walls of the buildings which need to be observed, i.e., $\mathcal{R} \subseteq \partial P$, a minimal overlap $L_{\min}$ can be considered necessary for a successful software-based registration. This is defined as the overlapping part of the walls visible from two standpoints p and q , i.e., $L_{pq} = \text{length}(\mathcal{V}(p) \cap \mathcal{R}) \cap (\mathcal{V}(q) \cap \mathcal{R})$. Thus, a successful software-based registration between two standpoints can be defined by the user as an arbitrary combination of values for $L_{\min}$ and $A_{\min}$, respectively.

Consequently, for each pair of standpoints p, q one edge $\{p, q\}$ is added to the registrability graph if the overlap between the visibility polygons of the two respective standpoints exceeds the defined values for $L_{\min}$ and $A_{\min}$. With G' we refer to a subgraph which is induced by the selected standpoints. A global connectivity of all standpoints in G' is in the following guaranteed by formulating a flow problem as introduced by Shirabe (2004). Following this paradigm, for each selected standpoint a unit of flow is injected, but only one standpoint is selected as sink which is capable of taking up all the flow. Since no flow is allowed to get lost, there must exist a way for each unit of flow to reach the sink. Thus, connectivity in the subgraph of selected standpoints G' will be enforced. To this end, a variable $w_p \in \{0, 1\} \forall p \in CP$ is introduced for each standpoint, indicating whether the particular point represents the sink. The constraint

$$\sum_{p \in CP} w_p = 1 \quad (5)$$

ensures that exactly one standpoint is selected as a sink. Additionally, only a selected standpoint ($x_p = 1$) can be chosen as a sink:

$$w_p \leq x_p \quad \forall p \in CP \quad (6)$$

Moreover, two variables $f_{pq}, f_{qp} \in \mathbb{R}^+ \forall (p, q) \in E$ are introduced for each edge (p, q) in the registrability graph G , indicating the direction and amount of flow. Two constraints are used to enforce all the above-mentioned flow properties. The first constraint forbids any outflow from the particular standpoint if it is not selected ($x_p = 0$). Additionally, it restricts the outflow if the node is selected ($x_p = 1$):

$$\sum_{\{p, q\} \in E} f_{pq} \leq (|V| - 1) \cdot x_p \quad \forall p \in CP \quad (7)$$

The second constraint forbids any inflow to a not selected standpoint. However, if the standpoint is selected ($x_p = 1$) an outflow is enforced since the difference of in- and outflow must be greater than x_p . This holds as long as the standpoint is not selected as sink ($w_p = 0$). If $w_p = 1$ the constraint allows the standpoint, which in this case is the sink, to take the complete flow of the network.

$$\sum_{\{p, q\} \in E} f_{pq} - \sum_{\{q, p\} \in E} f_{qp} \geq x_p - |V| \cdot w_p \quad \forall p \in CP \quad (8)$$

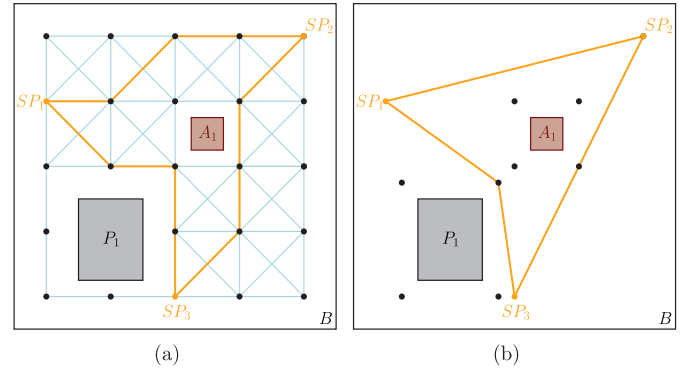


Fig. 9. Exemplary shortest round tour (orange) for three standpoints (a) based on an octilinear grid (Knechtel et al., 2022) and (b) using projected convex corner points. In (a) there exist multiple solutions for the shortest path with the exact same length and the octilinear grid structure leads to small detours. In contrast, the shortest path in (b) is usually unambiguous and offers a direct routing without detours, i.e., the geodesic path. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

As in this formulation also continuous variables are used, the problem now needs to be considered as a Mixed-Integer Linear Program (MILP).

3.1.4. Route planning

Last but not least, the route the scanner needs to travel for conducting the survey should be as short as possible. To this end, an ILP formulation of the Traveling Salesperson Problem (TSP) by Dantzig et al. (1954) is incorporated into our model to compute the best round tour that connects all selected standpoints and is minimal with respect to the total weight w_{tot} . The weight, i.e., cost of traveling between two points, can be arbitrarily determined, for example, as the travel time in order to account for difficult terrain conditions. In our study we employ a distance measure as weight without accounting for slope and other factors. Nevertheless, this is easily adaptable as part of the preprocessing. A round tour potentially is a practical option for both (semi)-autonomous and manual measurements considering a return to the start place where the equipment is possibly stored. However, the formulation can also be easily adapted if no round tour is desired.

In the preceding publication by Knechtel et al. (2022), the distance has been calculated on an octilinear grid (cf. Fig. 9(a)). However, with this approach there often exist multiple optimal solutions with the exact same length. This can influence the running time of the ILP solver. Moreover, the model tends to overestimate the length of the shortest path due to the octilinear grid structure. To make the result less ambiguous and more practice-oriented, we use the geodesic distance. The geodesic distance is here defined as the geometrical length of a geometrically shortest path between two points while considering obstacles. For that, on each corner of the polygon $P_{\text{candidate}}$ an intermediate point is projected into the routable space, as shown in Fig. 9(b). If an obstacle, e.g., a wall or a restricted area between the two respective standpoints exists, the routing is performed utilizing the intermediate corner points. To guarantee a safe routing, the polygons are additionally buffered by a certain distance ϵ_{buff} . For setting the weights of the edges of the routing graph $\hat{G}$, an all-pairs shortest paths calculation is performed.

A variable $y_{pq} \in \{0, 1\}$ is introduced for each edge $\{p, q\} \in \hat{E}$, which can be interpreted as

$$y_{pq} = \begin{cases} 1 & \text{if the output route connects } p \text{ and } q \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

Correspondingly, the objective is to minimize the selected weights, i.e., to minimize

$$O_{\text{Route}} = \sum_{p, q \in V} w_{pq} \cdot y_{pq} \quad (10)$$

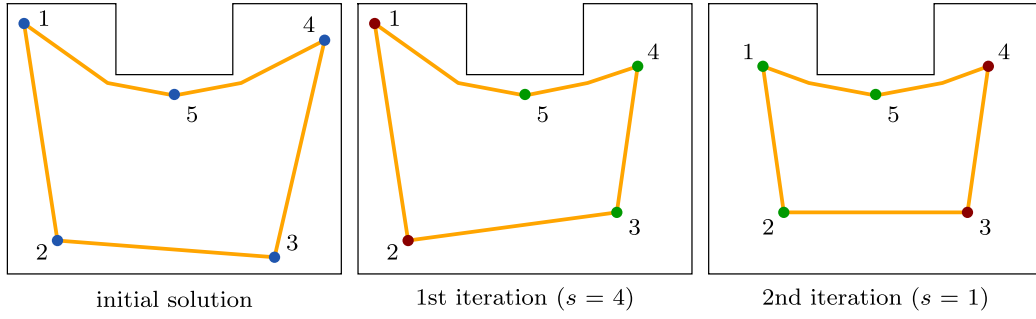


Fig. 10. Applying a local combined optimization as a heuristic given an initial solution. In this example, only one standpoint s and its $np = 2$ neighbors (green) on the route (orange) are optimized using the OneStep ILP formulation while the others (red) remain fixed. This is repeatedly done to achieve a globally better solution with respect to the path length. The first two iterations are here exemplarily depicted. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Algorithm 1 LocalSearch

Require: number np (even) of neighbors to be optimized

Require: number i of iterations

$S \leftarrow$ compute set of standpoints with TwoStep approach

repeat

$S_r \leftarrow$ shuffled(S)

for all $s \in S_r$ **do**

$S_{lo} \leftarrow S[s.id - np/2], \dots, S[s.id], \dots, S[s.id + np/2]$

$S_{lo} \leftarrow$ optimizeOneStep(S_{lo})

$S \leftarrow S_f \cup S_{lo}$

end for

until i iterations performed

This objective is the second part of the hierarchical objective function (cf. Eq. (1)). Each node which has been selected as a standpoint requires at least one incoming and one outgoing edge, and since the graph is undirected for each node it must hold

$$\sum_{q \in V, q \neq p} y_{pq} = 2 \cdot x_p \quad \forall p \in V. \quad (11)$$

As the resulting path needs to contain all nodes, the last constraint aims to eliminate isolated subtours by enforcing each subgraph $V' \subseteq V$ to be connected to the other vertices by at least two edges by constraining the number of selected edges in the subset:

$$\sum_{p \in V'} \sum_{q \in V', p \neq q} y_{pq} \leq |V'| - 1 \quad \forall V' \subseteq V \quad (12)$$

However, the number of subsets of V is exponential. Thus, adding all possible constraints to the model at the beginning is exhaustive and comes with huge memory requirements. The concept of separating subtour elimination constraints has been well-known and practiced since the first formulation of Dantzig et al. (1954). We therefore in the first instantiation omit these constraints and check for every feasible solution that is found during the solving process whether subtours exist. By explicitly adding constraint (12) for these particular subtours they are consequently forbidden afterwards. This iterative procedure of adding constraints to the model is supported by state-of-the-art solvers and the number of constraints added in this fashion is normally moderate (Pfersch and Stanek, 2017).

3.2. Speed-up with local search

As previously stated, solving two NP-hard problems using an ILP solver can, depending on the specific problem configuration, result in longer computation times. A speed-up possibility is to compute a starting solution S , e.g., with the TwoStep approach, and solve the OneStep formulation iteratively only for different subsets of this

solution. Meanwhile, the rest of the standpoints remains fixed. The corresponding pseudocode for this heuristic can be found in Algorithm 1. The subset S_{lo} of points which are locally optimized always consists of one standpoint s and np neighboring points, i.e., points which are consecutive in the sequence of points which comprises the current route. In our case, np is always an even number to enforce a symmetric distribution of the neighbored points. The remaining standpoints $S_f = S \setminus S_{lo}$ remain fixed during this procedure. To gain an additional speed-up, not only the selection of the remaining standpoints remains fixed, but also the route between them. Applying the OneStep formulation for only the subset S_{lo} yields a result for which holds that given the fixed standpoints there exists no smaller combination of standpoints which additionally comprises a smaller route. This result is then subsequently incorporated into the global solution S . This step needs to be iteratively repeated, i.e., for each standpoint the local search is applied i times.

In order to achieve enhanced stability for the solution, the points for the local optimization are not selected in their sequential order along the route. Instead, the order is randomized. Nevertheless, the selection of the neighboring points is still performed with regard to the original ordering in the route. Obviously, np , i.e., the size of the subset which is optimized, as well as i , i.e., the number of repetitions of the local search for each standpoint, are parameters which need to be determined.

The first steps of this procedure are for a small example exemplarily shown in Fig. 10. The initial solution is computed from the TwoStep approach. After shuffling the optimization order, standpoint 4 is selected first as s for local optimization. For this example, we chose to optimize $np = 2$ neighboring standpoints, i.e., the succeeding and preceding standpoint of s . Hence, the set of standpoints for the local search is $S_{lo} = \{3, 4, 5\}$. During this process, better positions regarding the overall route length for standpoints 3 and 4, respectively, are found and incorporated into the solution. Nevertheless, all constraints regarding visibility and software-based registration are still globally preserved. For the second step with standpoint 1 and the corresponding set $S_{lo} = \{5, 1, 2\}$, two new standpoints are selected, while standpoint 5 remains unchanged. Although the figure only exemplarily shows the first two steps, this is repeatedly performed i times for each standpoint. In the end, by iteratively applying the local optimization we receive a heuristic solution. In Section 4 the quality of this solution will be evaluated and compared to the optimal solution from the OneStep approach as well as to results from other speed-up techniques and from the TwoStep formulation.

3.3. Enhanced survivability constraints

In this article we furthermore aim to provide the option to introduce additional constraints regarding the network topology to enhance the survivability of the network, i.e., its resilience to potential failures induced by registration inaccuracies. We will in the following first

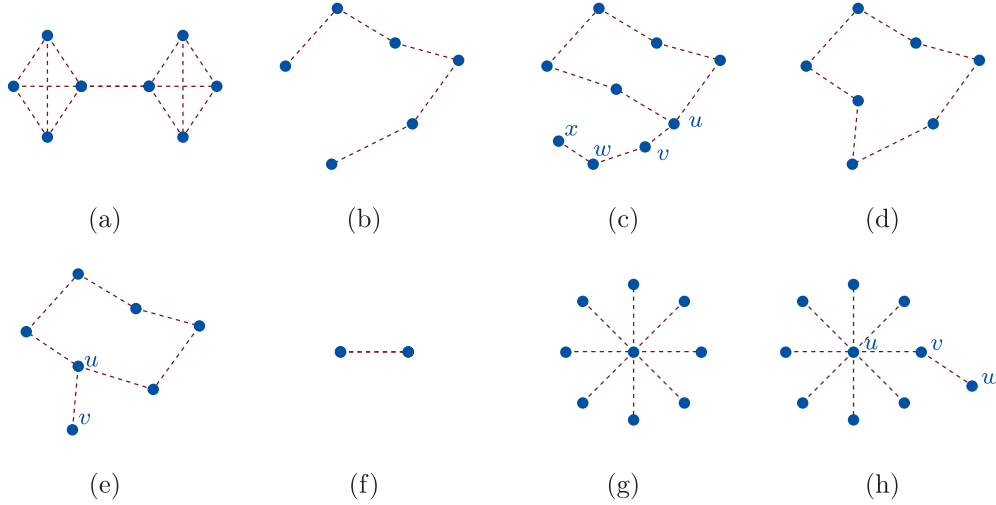


Fig. 11. Different registration networks. The graphs in (a) and (b) are 1-edge-connected. For (c), the upper part of the graph is 2-edge-connected, but the lower part consisting of u, v, w, x is only 1-edge-connected. Graph (d) is 2-edge-connected. (e) depicts a pendant edge $\{u, v\}$ connected to an otherwise 2-edge-connected graph, i.e., it is after-pruning-2-edge-connected. Same holds for (f) and (g) as they only comprise pendant edges. (h) is again 1-edge-connected.

elaborate a suitable definition of connectivity in the context of laser scanning. Subsequently, we will present the corresponding constraints for our ILP formulation.

3.3.1. Suitable definition of connectivity

In a general context, the survivability of a network is defined by its particular connectivity (Kuipers, 2012). Each network can be conceptualized as a graph, therefore we can approach this problem from a graph-theoretical standpoint. In general, it is defined that k -edge-connectivity for a graph is given, if the graph remains connected whenever fewer than k edges are removed, i.e., there still exists a path between every pair of vertices in the graph (Diestel, 2017). The formal definition is as follows: Let $G = (V, E)$ be an arbitrary graph. G is k -edge-connected if and only if the subgraph $G' = (V, E \setminus X)$ is connected for all $X \subseteq E$ with the size $|X|$ being smaller than k . In other words, the minimum edge cut, i.e., the smallest set of edges which need to be removed to separate the graph into multiple components, has a size of at least k . A connected graph is therefore at least 1-edge-connected. Menger's theorem from graph theory additionally characterizes the connectivity of a graph by connecting the size of the minimum cut-set to the number of edge-disjoint paths (Menger, 1927):

Theorem 1 (Menger's-Theorem). *Given a finite undirected graph and two distinct vertices, the size of the minimum edge cut is equal to the maximum number of pairwise edge-disjoint paths between the two vertices.*

Thus, a graph is k -edge-connected, if and only if every pair of nodes has k edge-disjoint paths between them. Generally, two paths are said to be edge-disjoint if they do not have any edge in common.

The flow formulation from Section 3.1 enforces that the registration graph is connected, i.e., at least 1-edge-connected. Corresponding registration networks are shown in Figs. 11(a) and 11(b). As previously elaborated, both of them are prone to errors, since each contains at least one registration edge which is not controllable, but influences the result of subsequent standpoints. The registration network depicted in Fig. 11(c) is not sufficiently redundant as well. Although the top part is even 2-edge-connected, the path at the bottom is only 1-edge-connected. A deviation in the registration between u and v is therefore also propagated into the registration of the point cloud resulting from w and x . In conclusion, simply enforcing 1-edge-connectivity to gain redundancy is too weak and hence not sufficient with respect to the redundancy demands of a laser scanner network.

Enforcing 2-edge-connectivity can potentially overcome these shortcomings. This could, for example, lead to the use of an additional

standpoint (cf. Fig. 11(d)). The example from Fig. 11(e) would not be sufficient when enforcing 2-edge-connectivity, since the vertex v at the bottom would be disconnected from the remaining graph by just removing the corresponding edge $\{u, v\}$. Yet, an error in the registration would not influence any subsequent standpoints and such configurations are very common in scanning networks designed by experts. Even the small example in Fig. 11(f) would be forbidden when enforcing 2-edge-connectivity, although this is a network which from a practical perspective would be suitable for a small scenery. Therefore, simply enforcing 2-edge-connectivity can be considered too strong.

Since in conclusion a connectivity definition purely based on k -edge-connectivity is not suitable, we define the term after-pruning- k -edge-connected:

Definition 1. A graph G is after-pruning- k -edge-connected if the subgraph G' of G that is induced by the set of nodes with degree of at least two is k -edge-connected.

In other words, the subgraph which originates from one pruning operation, i.e., the deletion of all vertices with degree 1, needs to be k -edge-connected. Consequently, the networks in Fig. 11(d), 11(e) and 11(f), i.e., the networks with a desirable registration network topology, are after-pruning-2-edge-connected. Even the scanning network in Figs. 11(g) is allowed in this context. All edges are pendant edges and therefore a registration inaccuracy would only influence the standpoint itself. In contrast, 11(h) is again not a valid result. As desired, also the examples in Figs. 11(a)–11(c) are not after-pruning-2-edge-connected. In conclusion, enforcing after-pruning-2-edge-connectivity for the underlying registration network only yields results comprising the desired network topology, i.e., scanning networks with a desirable amount of redundancy.

3.3.2. Constraint formulation

To enforce said after-pruning-2-edge-connectivity, an additional binary variable for each registrability edge $\{p, q\} \in E$ of the registrability graph presented in Section 3.1.3 is introduced. The variable indicates if this edge is part of the registration network:

$$z_{pq} = \begin{cases} 1 & , \text{ if registrability edge } \{pq\} \text{ is selected} \\ 0 & , \text{ otherwise} \end{cases} \quad (13)$$

First, we introduce a constraint, which enforces that the edge between two nodes cannot be selected if at least one of the nodes, i.e., standpoints, is not selected:

$$2z_{pq} \leq x_p + x_q \quad \forall \{p, q\} \in E \quad (14)$$

The core of the formulation is a constraint, that aims to enforce after-pruning-2-edge-connectivity for a graph by enforcing a size of at least two for the minimum cut-set for all parts of the graph that are not a pendant vertex. In general, a cut $C = (A, B)$ is a partition of the node set of a graph $G = (V, E)$ into two subsets A and B . The cut-set CS of a cut C is the set of edges that have one incident vertex in S and the other in T , i.e., $CS(C) = \{\{u, v\} \in E \mid u \in A, v \in B\}$. Their removal would consequently disconnect the graph into the two particular subsets. For each pair of edges $a, b \in E$ and each cut C for which a and b are in different subsets, i.e., $a \subseteq A$ and $b \subseteq B$, we employ the following constraint:

$$2z_a + 2z_b - 2 \leq \sum_{e \in CS(C)} z_e \quad \forall C(A, B), a \subseteq A, b \subseteq B \quad (15)$$

If both edges a, b are selected, the constraint enforces that at least two of the edges $e \in CS$ of the cut-set need to be selected. Otherwise, the constraint is relaxed. In other words, the application of this constraint for a pair of edges necessitates, if both should be part of the final registration network, the selection of at least two edges of the cut-set of each possible cut disconnecting the particular edges. With this, the capacity of the minimum cut between these edges is constrained to at least 2, and as this is done for each pair of edges the resulting registrability graph is after-pruning-2-edge connected. This holds as pendant vertices are still allowed: Assume again a cut $C = (A, B)$, but one of the partitions, e.g., B , consists of only one vertex v and consequently has no edges ($E(B) = \emptyset$). Thus, there exists no edge $b \in B$ for which the constraint can be set up. Hence, no additional constraint is active and consequently no 2-edge-connectivity is enforced. The flow formulation from Section 3.1, however, still enforces a simple connectivity for this vertex.

3.3.3. Speed-up of the computation

Setting up this constraint for all combinations of edges and cuts, however, has a high impact on the number of constraints of the ILP and consequently can influence the solving time. When assuming that our registrability graph has n edges, there exist $O(n^2)$ combinations of edges and for these combinations there possibly exists an exponential number of cuts. Consequently, we aim to not include all possible constraints in the ILP instance directly at the beginning, but only the constraints which are actually violated for solutions found during the solution procedure. This is possible since we employ the branch-and-cut technique as combinatorial optimization method to solve our problem formulation, which involves the use of branch-and-bound algorithms and cutting-planes. In the cutting-plane method, one adds consecutive constraints to a relaxed LP formulation. For the solution of each LP relaxation, a cutting plane is said to be an inequality, which is violated by the current solution. This inequality is consequently added to the ILP formulation. In a similar procedure, we can check in a callback for every produced feasible integer solution if the corresponding registrability graph is 2-edge-connected. If this is not the case, we add the particular violated constraints to the ILP, which therefore cuts off the solution.

This check can be easily performed by computing the bridges of the registration graph. A bridge is defined as an edge whose deletion disconnects the graph, i.e., the number of components is increased. This is exemplarily depicted in Fig. 12 for one possible solution. Here, the edge $\{u, v\}$ is a bridge. Without this edge, the graph would consist of two components A and B . Therefore, the after-pruning-2-edge-connectivity constraint is definitely violated for one edge $a \in A$ and one edge $b \in B$ and two cut-sets, consisting of the outgoing edges $CS_A = \delta^-(A)$ for component A , and $CS_B = \delta^-(B)$ for B , respectively. For these two edges a and b and the two particular cuts CS_A, CS_B , our constraint is added in the callback. Therefore, this specific solution is not feasible anymore. This could either lead to a scanning network in which at least one of a and b are not part of the registrability graph since the constraint is relaxed if either a or b are not selected, or one of the outgoing edges for A and B , respectively, are selected in addition to

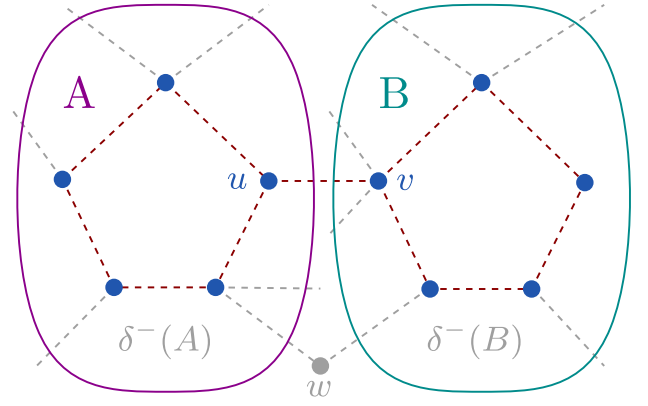


Fig. 12. Depiction of a registration graph of a possible solution before violated constraints are enforced via a callback. The edge $\{u, v\}$ is a bridge, i.e., its removal would separate the graph into two components A, B . To enforce 2-edge-connectivity for each of the two components A, B at least one of the outgoing registration edges in $\delta^-(A)$ and $\delta^-(B)$ (gray dashed) need to be selected. For example, the gray standpoint at the bottom could be included. Consequently, the two incident registration edges would be selected and 2-edge-connectivity would be obtained. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the bridge. This could happen, for example, by additionally selecting the standpoint w . Consequently, the two incident registrability edges, which are part of $\delta^-(A)$ and $\delta^-(B)$, respectively, can be selected. This process is repeated iteratively for all feasible integer solutions produced by the solver so that the final solution does not violate constraint (15). With this procedure, the number of added constraints can amount to a fraction of all possible constraints enforcing after-pruning-2-edge-connectivity. Therefore, the memory usage as well as the computation time can be substantially reduced.

To further speed up the solving process of the ILP, the following two constraints are additionally introduced as valid inequalities. First, an edge needs to be selected when any amount of flow is allocated to the edge, as this is equivalent to being part of the connectivity network:

$$\begin{aligned} (|V| - 1) \cdot z_{pq} &\geq f_{pq} \quad \forall \{p, q\} \in E \\ (|V| - 1) \cdot z_{qp} &\geq f_{qp} \end{aligned} \quad (16)$$

In the same context, the edge needs to be definitely selected if both adjacent nodes are selected:

$$z_{pq} \geq x_p + x_q - 1 \quad \forall \{p, q\} \in E \quad (17)$$

The results of this formulation in comparison to the simple connectivity will be assessed on a real-world example in Section 4.2.

4. Experiments

The experiments are divided into two parts. First, the OneStep approach and the LocalSearch heuristic are compared to the previous TwoStep optimization method as well as to different techniques to speed up the solving process on a variety of different building scenarios. In the second part, we apply for the survey of a large real-world scenario the OneStep algorithm together with our extended survivability constraint formulation which enforces redundancy in the registration network.

4.1. Comparison on different building outlines

To assess the performance of our approaches, we applied the different algorithms on various building scenarios with varying extents, for indoor scan planning as well as for outdoor scanning. All corresponding floorplans can be found as supplementary materials. The respective results are compared with regard to (1) their runtime and (2) the length of the corresponding shortest route between the standpoints.

Table 2

Comparison of the runtime and the length of the route using the different proposed optimization algorithms. For the OneStep formulation we applied a warm start with the result of the TwoStep formulation as starting solution.

	grid points	stand- points	TwoStep		OneStep		LocalSearch		LS+OneStep	
			time [s]	tour [m]	time [s]	tour [m]	time [s]	tour [m]	time [s]	tour [m]
ex1	279	5	18	211.9	573	175.0	593	175.0	2443	175.0
ex2	231	9	2	2149.6	4669	1856.5	38	1929.5	4125	1856.5
ex3	162	8	2	1709.1	806	1390.7	38	1399.9	647	1390.7
ex4	297	11	16	138.4	16164	114.9	253	114.9	9593	114.9
ex5	399	9	5	2631.0	28	2381.6	81	2381.6	112	2381.6
ex6	346	9	17	165.1	3860	129.5	214	131.6	3416	129.5
ex7	370	13	71	3463.0	585621	3044.0	843	3044.0	-	-
delft	548	9	27	83.95	1466	73.7	611	73.7	1315	73.7
campus	462	16	16	1381.2	792	1341.5	279	1341.7	885	1341.5

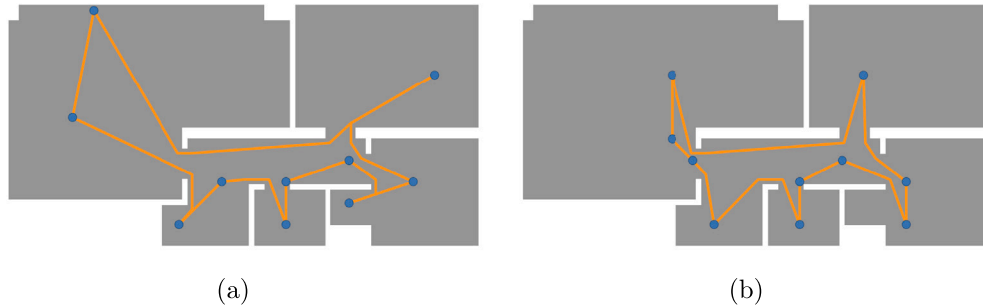


Fig. 13. Standpoints (blue) and the corresponding route (orange) resulting from the TwoStep (a) and OneStep (b) optimization for the building example ex6. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

For this, all experiments were conducted on a computer equipped with an AMD Ryzen 9 7950X processor and 64 GB RAM. As solver for the ILP, Gurobi¹ in version 9.5 was employed in combination with Java in version 22.0.1. The corresponding code is available on gitlab.²

The results are shown in Table 2. The number of standpoints is the same for all methods for the respective building due to our hierarchical objective function, in which the standpoint planning was introduced with a higher priority. Hence, only the length of the corresponding route is changing. In all tested scenarios the tour resulting from the TwoStep approach is longer than the result stemming from our OneStep formulation, the difference is as much as up to 33%. The mean difference is 17%, with only one instance where the difference was less than 10%. The result for ex6 is exemplarily depicted in Fig. 13. The aforementioned numbers justify the assertion that the OneStep formulation poses a distinct advantage in terms of its applicability and economic efficiency with regard to the surveying time when compared to the preceding TwoStep solution.

Since the solver we applied is capable of including a starting solution to reduce the solving time in the so-called warm start procedure, we used the result generated from the TwoStep formulation for our OneStep approach. Nevertheless, the time needed to solve our formulation is recognizably longer. However, notwithstanding the extended calculation time the solver needs to yield an optimal solution, the majority of instances can be solved within a maximum of a few hours. One rare exception is ex7, but it should be noted that small variations, for example in the density of the standpoints, can bring the calculation time into similar ranges.

In Table 3 we show the influence the selection of different measurement-related parameters can have exemplarily for two parameters with high impact on the building ex2. Nevertheless, of course also the choice of the other parameters plays an important role for the runtime. In general, a denser grid results in more parameters and

Table 3

Influence of different parameter settings on the runtime of the OneStep approach for floorplan example ex2.

Overlap wall [m]	Grid size [m]	# Candidate points	Runtime [s]
30	2	231	56171
30	1.5	430	144
20	2	231	5581

correspondingly more constraints. Hence, this is accordingly extending the solution time. It can, however, in some cases also lead to a more straightforward problem constellation for the solver, resulting in a better solution path. In this case, a denser grid of 1.5 m is substantially reducing the runtime. For more details on the influence of the selection of the grid size on the quality of the result and the runtime of the solver the interested reader is referred to Dehbi et al. (2021). Additionally, only changing one other parameter, e.g., the desired overlap of the walls, can reduce the solution time by more than 90%. This can be attributed to the fact that the registrability graph is getting denser if the required overlap is smaller. This in turn leads to more possible combinations of standpoints that need to be explored by the solver. A further observation on the runtime can be made regarding the characteristics of the underlying scenery. Buildings which comprise a very uniform layout pose a bigger challenge to the solver than a building with many protrusions and recesses. This is due to the fact that for the uniform layout the choice of the standpoints can be highly arbitrary. For example, if a long straight wall needs 3 standpoints to be fully observed, all standpoints candidates on a straight line have the same influence on the objective functions. To prove the optimality of the solution, the solver therefore needs to exhaustively explore a high number of equally good solutions.

In conclusion, specific choices of different variables as well as characteristics of the scenery to observe can lead to longer solving times. Potential application scenarios, in which an overnight calculation of the corresponding solution appears to be not suitable, therefore motivate the use of a heuristic with a faster computation time. In this context, we in the following evaluate our proposed LocalSearch method. For the reported results we chose $np = 2$, i.e., we locally optimized

¹ <https://www.gurobi.com/solutions/gurobi-optimizer/>

² <https://gitlab.igg.uni-bonn.de/geoinfo/laserscan-optimization-redundancy>

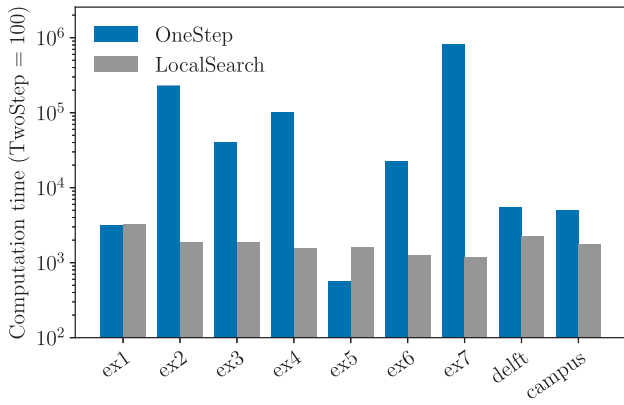


Fig. 14. Computation time of the OneStep formulation and the LocalSearch method compared to the TwoStep approach. The scale in this figure is logarithmic.

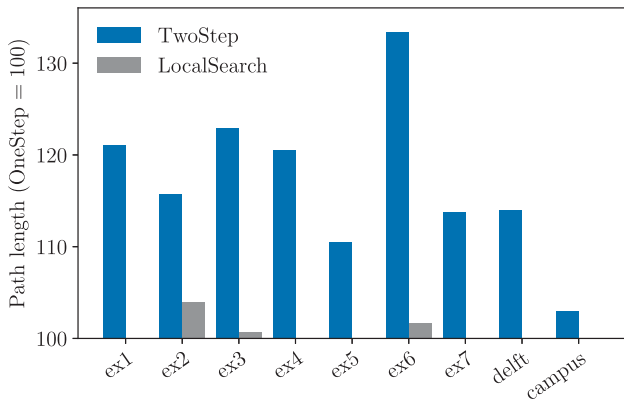


Fig. 15. Length of the route resulting from the TwoStep approach and LocalSearch method compared to the optimal path length from the OneStep method.

3 neighboring standpoints while fixing the remaining points. Additionally, we set $i = 2$, leading to each standpoint being selected twice for the local search. In general, as depicted in Table 2, the LocalSearch method can solve the problem instances substantially faster than the OneStep approach, although not as fast as the TwoStep formulation. The solving times for the three approaches for our exemplary buildings are additionally shown in Fig. 14, in which the TwoStep approach serves as a baseline to which the difference in computation time is depicted on a logarithmic scale. ex5 represents an exception since the computation time of the heuristic is longer than for the combined ILP. This can be attributed to the very fast solving time of 28 s for the OneStep approach. In this specific case, the repeated call of the local search and its optimization produces an overhead. Nevertheless, for instances which are harder to solve, the local search heuristic demonstrates the potential to substantially reduce the time required to obtain an optimal solution.

In addition to the achieved speed-up in comparison to the solution time of the OneStep approach, the heuristic consistently yields superior results regarding the solution quality compared to the TwoStep formulation. This again is depicted in Fig. 15, in which the optimal result of OneStep serves as a baseline. In five out of the eight cases, our heuristic yields the optimal result for the route from the OneStep formulation. In the remaining three cases, the length difference of the route amounts to a maximum of 4%. The mean difference is only 0.7%, which is substantially less than the 17% mean difference of the TwoStep approach. In conclusion, these examples illustrate the efficacy of our LocalSearch heuristic, as high-quality results can be attained in a mere fraction of the time compared to the OneStep approach.

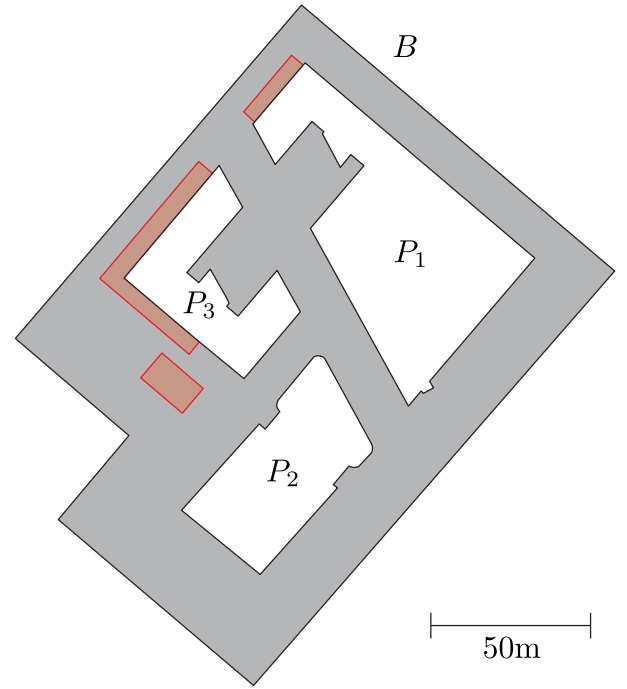


Fig. 16. The surveyed scenery on Campus Poppelsdorf of the University of Bonn, Germany. P_1 , P_2 and P_3 are the buildings which need to be surveyed, red areas are not viable. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Different additional techniques to speed up the computation time of the exact algorithm have been tested. In general, we applied a warm start for the OneStep method, i.e., the solution of the twoStep approach was given to the solver as starting solution. LS+OneStep, which can also be found in Table 2, works in a similar way, however, as starting solution we calculate the solution of the LocalSearch method, which yields as previously stated better results than the TwoStep solution but is solved faster than the OneStep approach. As it can be seen in the table, obviously the method always yields the same result as the OneStep method, since in the second step the identical problem formulation is applied. Yet, it offers a speed-up possibility by introducing a better start solution. Thus, the solving time can be reduced compared to the OneStep formulation. This is, however, not the case for the building ex1. This can be accounted to the fact that the start solution also influences the solution path Gurobi takes while trying to find the optimal solution. Setting a starting solution can therefore in rare cases lead to a disadvantageous solution path. Additionally, the solution path is also influenced by the seed parameter of Gurobi, i.e., a number which is normally randomly chosen at the beginning of the optimization and which acts as a small perturbation to the solver. In general, Gurobi offers the possibility to tune a huge set of parameters which all influence the solving time. For example, it is possible to set a gap parameter ϵ , allowing the solver to terminate with a solution that is at most $(1 + \epsilon)$ times worse than the optimal solution. However, for our experiments we used the standard parameter settings.

4.2. Application of the survivability constraint on a real-world scenario

In addition to the evaluation in the previous subsection, we also performed a planning based on the OneStep approach to apply and assess the influence of our connectivity formulation, which enforces redundancy. The planning was subsequently executed to prove its applicability in a real-world scenario. The scenery, which is depicted in Fig. 16, is located on the campus of the University of Bonn and comprises

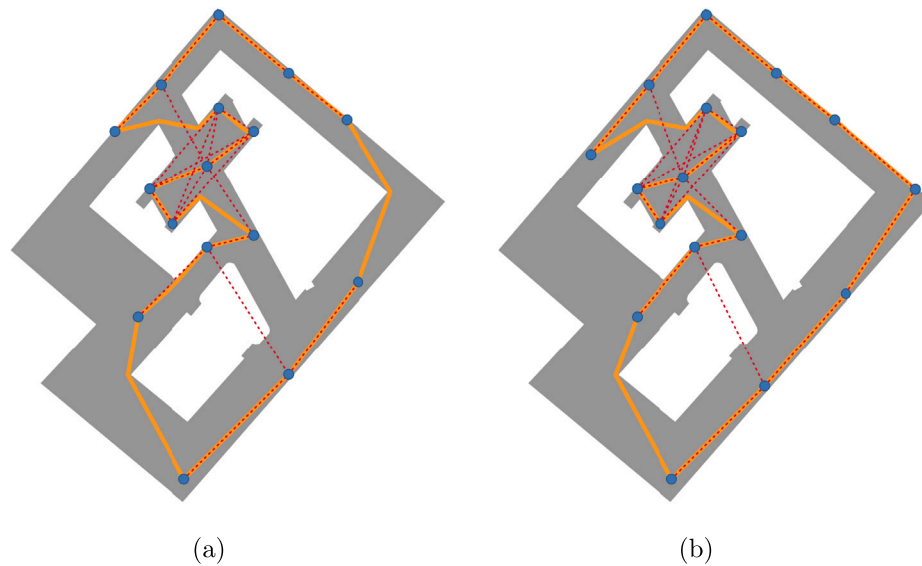


Fig. 17. Solution of our OneStep approach with the simple connectivity formulation (a) and with our survivability constraint to enforce after-pruning-2-edge-connectivity (b). The orange line indicates the route between the respective standpoints. The red dotted lines depict the edges of the registrability graph and, hence, indicate which standpoints can be registered together software-based. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

three major buildings P_1 , P_2 and P_3 . The aim is to survey these buildings from the outside. The input for our algorithm consists of georeferenced building polygons, which originate from *OpenStreetMap*.³ The size of the whole area bounded by the polygon B amounts to approximately 18,000 m². The upper left and upper right border of B are determined by the presence of vegetation and stables, respectively. The red areas indicate locations where setting up the laser scanner is not possible, for example, due to an inaccessible educational garden. The grid size was set to 5 m resulting in 462 candidate positions. For the scanner-related constraints, we chose a maximum measurement distance of 60 m and a minimum distance of 1 m. The minimum incidence angle is set to 70°. The minimal length of overlapping walls between two standpoints to allow for a subsequent software-based registration amounts to 20 m, additionally an overlapping floor area of 20 m² is required.

When directly applying the OneStep optimization approach with the simple connectivity constraints this results in a survey planning for stop-and-go laser scanning as shown in Fig. 17(a). The orange line illustrates the route connecting the blue standpoints. The dashed red lines are the registrability edges and indicate between which standpoints a software-based registration according to our selected parameters is possible. Although this is a valid solution with regard to all constraints, the five upper and 6 lower standpoints are each only weakly connected by one registration edge to the central part of the scanning network, which lies in between the buildings P_1 and P_3 . Therefore, an error in this registration could lead to a tilting of a large part of the point cloud. The network resulting from applying the additional survivability constraint which enforces after-pruning-2-edge-connectivity as presented in Section 3.3 is shown in Fig. 17(b). By introducing one additional standpoint, a loop closure between the upper and lower parts of the network is obtained. Additionally, three pendant edges can be found, as they are still explicitly permitted by our constraints. Obviously, one registrability edge in the lower part of the scenery is crossing the building P_2 . This is, however, intended, since for the software-based registration no direct visibility between two standpoints is necessary. Yet, only the overlap in the resulting point cloud needs to be sufficient and since both standpoints observe large parts of P_3 as well as a large portion of the floor, the chosen requirements regarding the overlap are fulfilled.

This resulting scan planning was used for a survey, which was performed in a stop-and-go manner utilizing a *Leica ScanStation P50*.⁴ The subsequent software-based registration was carried out with *Leica Cyclone* resulting in a mean absolute error of 0.2 cm, ranging from 0.1 cm to 0.5 cm for the respective standpoints. The point cloud of our survey is depicted in Fig. 18. To analyze the completeness of the point cloud, models of the buildings are provided. These building models originate from georeferenced authoritative cadastral data from the state of North Rhine-Westphalia, Germany,⁵ and correspond to the Level-of-Detail (LoD) 2 as defined in the CityGML OGC standard.⁶ Hence, they pose an error of at maximum few centimeters on the ground. It can be seen that all parts of the building have been captured by the scanner and are reflected in the point cloud. Even narrow, twisted parts of the building ensemble as well as protrusions and recesses with rounded corners were fully recorded, e.g., (d) and (e). This holds for any parts, which are observable at all by a terrestrial laser scanner. For roofs, for example, other measurement methods like airborne laser scanning need to be employed. One part of the southern wall of P_3 is missing, as seen in the bottom left part of Fig. 18(b), since a construction container was set up there spontaneously which made an observation impossible. For the same reason, one standpoint needed to be moved by roughly 0.5 m, since its planned position was occupied. The influence of small deviations of the scanner positions has already been discussed by Dehbi et al. (2021). It was found that the influence on the quality of the point cloud is usually small. However, in edge cases at corners this may lead to unobserved parts of the object.

Although this also had no noticeable effect in our case, it is a clear illustration of the risk involved in actually implementing plans in a real-world scenario. A second factor influencing our approach is the quality of the underlying input, i.e., building outlines. In the event that the plan fails to accurately depict the actual situation, for instance, due to a generalization process, there exists the possibility that the resulting scan plan might fail to observe all parts of the building in accordance with the aforementioned quality constraints. Although for small deviations the plan can potentially still be valid, larger deviations

³ <https://www.openstreetmap.org>

⁴ <https://leica-geosystems.com/en-us/products/laser-scanners/scanners/leica-scanstation-p50>

⁵ https://www.opengeodata.nrw.de/produkte/geobasis/3dg/lo2_gml/

⁶ <https://www.ogc.org/publications/standard/citygml/>

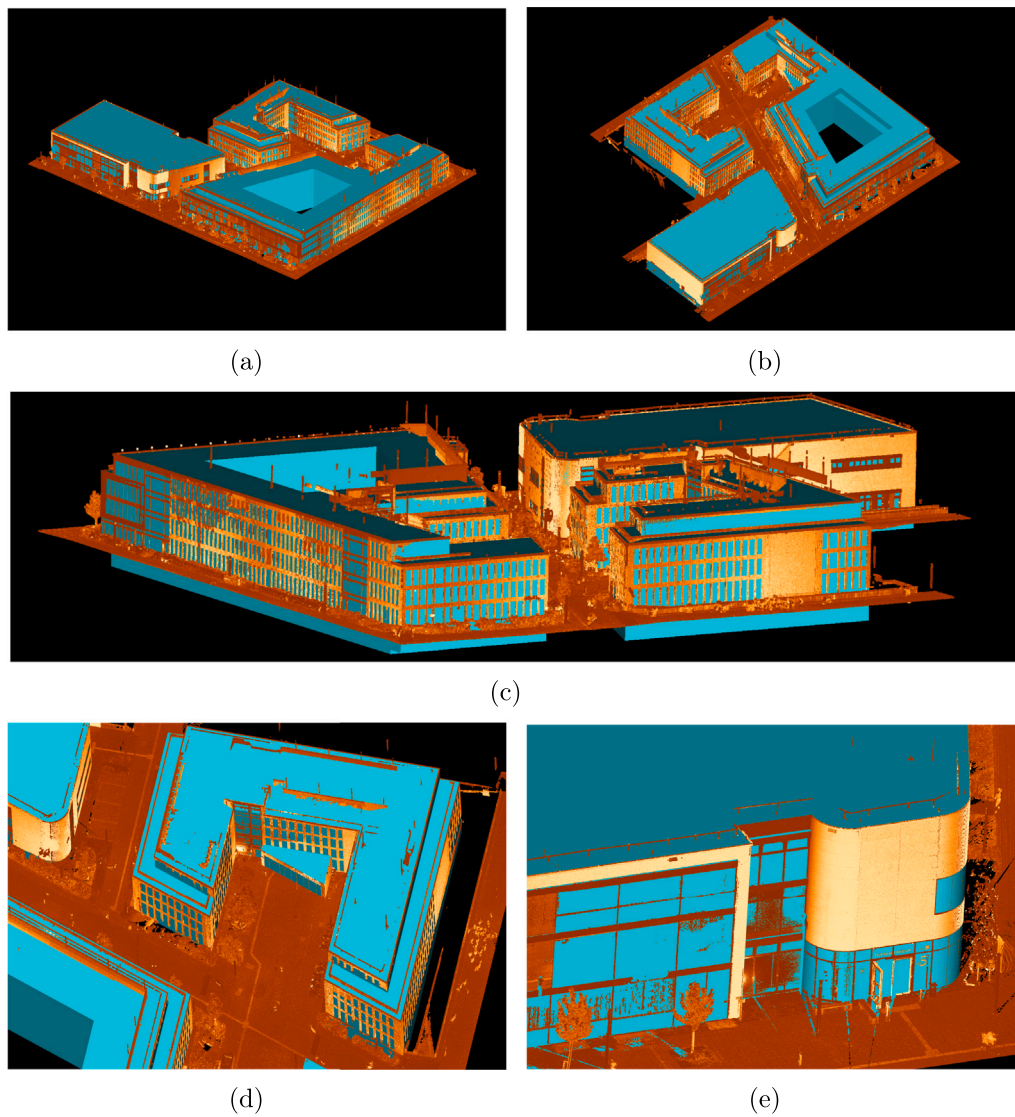


Fig. 18. Resulting point cloud (orange) from our survey overlaid on models of the corresponding buildings (blue) according to the CITYGML LoD2 standard. The coloring corresponds to the intensity of the particular measurements. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

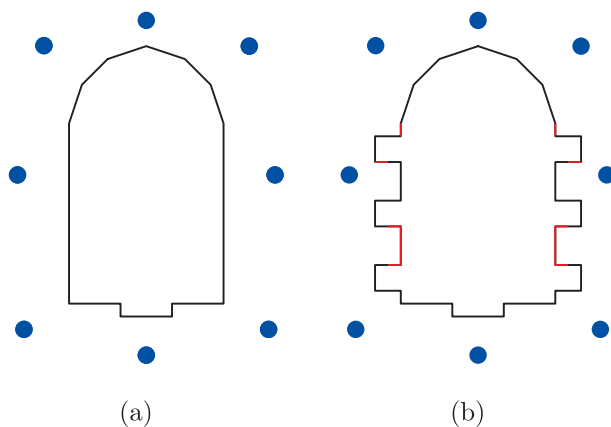


Fig. 19. Influence of deviations between the input building outline (a) and the real building footprint (b). Due to obstructions and resulting incidence angles, some parts of the buildings (red) are not observable with regard to the scanner-related constraints using the original standpoints. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

also increase the likelihood of an inappropriate planning. An example of this is shown in Fig. 19. On the left the outline which serves as input is depicted as well as a possible set of standpoints comprising the survey. The actual footprint of the building is shown on the right. The red parts of the boundary are either not observable from the preplanned standpoints due to obstructions or do not meet the scanner-related constraints, specifically the incidence angle. Beyond the completeness of the resulting point cloud also the successful registration could be endangered because the overlap between two scans is not sufficient anymore. Hence, a strategy to evaluate any kind of deviation to the offline plan and to implement any necessary changes without changing the features of the optimal solution, potentially on-site, seems to be a valuable goal for future research.

5. Conclusion and future work

In this article we presented a combined ILP formulation to calculate the minimum number of standpoints and the corresponding shortest route between them in a OneStep approach. We outperform the preceding TwoStep approach regarding the length of the route by up to 33%, on average by 17%. The approach therefore offers on the one hand better economic efficiency when performing the respective

survey. On the other hand it can provide an optimal baseline based on the candidate positions for further comparisons of future approaches. In order to accelerate the calculation, a method utilizing local search for iterative optimization of smaller sub-problems was additionally proposed. This method was observed to provide the optimal solution in the majority of cases in a mere fraction of the time compared to the OneStep approach. In the remaining instances, it was found to be no more than 4% inferior to the optimal solution with regard to the length of the route, while on average being 0.7% inferior.

Incorporating ideas from network survivability, we additionally provide an extra set of constraints for the ILP formulation to add desired redundancy to the registration network. As common connectivity properties are not suitable in the context of laser scanning, we define the term after-pruning- k -edge-connectivity. With this, we are able to enforce a more favorable topology of the scanning network with regard to the subsequent software-based registration. In general, each registration can be controlled by another registration. However, pendant vertices, i.e., vertices with degree 1, are explicitly allowed, since in practice single laser scanner standpoints need to be easily connected to the remaining network to avoid a measurement overhead. The constraints were additionally tested on a real-world scenario, allowing for a highly accurate software-based registration with a mean absolute registration error of 0.2 cm.

Obviously, the proposed formulation heavily relies on the input polygon which describes the scenery to be surveyed. The influence of deviations of the input, e.g., due to a generalization of the input data, or the standpoint positions, e.g., due to an unforeseen obstacle, as well as strategies to overcome possible issues is ongoing work and will be addressed in a future publication. Furthermore, the second factor influencing the result and highly impacting the running time is the number of candidate points. Although a grid yields meaningful results, the investigation of other methods to generate candidate points based on the underlying geometry of the buildings to be surveyed, or even the use of no fixed candidate points seem to be a promising field of research for future work.

CRediT authorship contribution statement

Julius Knechtel: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Youness Dehbi:** Writing – review & editing, Conceptualization. **Lasse Klingbeil:** Writing – review & editing, Resources. **Jan-Henrik Haunert:** Writing – review & editing, Supervision, Resources, Project administration, Methodology, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We gratefully acknowledge the open data sets from Geobasis NRW and OpenStreetMap used in this article. The authors would like to express their gratitude to Prof. Heiner Kuhlmann and Martin Blome for their valuable assistance with the survey campaign. This work was partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under (1) Germany's Excellence Strategy - EXC2070 - 390732324 and (2) FOR5361 - 459420781.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.isprsjprs.2025.03.017>.

References

- Abbas, G., 2006. Network survivability. In: Kazi, K. (Ed.), *Optical Networking Standards: A Comprehensive Guide*. Springer US, Boston, MA, pp. 295–319.
- Ahn, P.J., Wahn, K., 2016. Interactive scan planning for heritage recording. *Multimed. Tools Appl.* 75 (7), 3655–3675.
- Aryan, A., Bosché, F., Tang, P., 2021. Planning for terrestrial laser scanning in construction: A review. *Autom. Constr.* 125, 103551.
- Besl, P.J., McKay, N.D., 1992. Method for registration of 3-D shapes. In: *Sensor Fusion IV: Control Paradigms and Data Structures*, Vol. 1611. International Society for Optics and Photonics, pp. 586–606.
- Biswas, H., Bosché, F., Sun, M., 2015. Planning for scanning using building information models: A novel approach with occlusion handling. In: *Proc. Int. Symp. Autom. Robot. Constr. ISARC 2015 Proceedings of the 32nd ISARC*, Oulu, Finland. pp. 1–8.
- Bolourian, N., Hammad, A., 2020. LiDAR-equipped UAV path planning considering potential locations of defects for bridge inspection. *Autom. Constr.* 117, 103250.
- Brenner, C., Dold, C., Ripperda, N., 2008. Coarse orientation of terrestrial laser scans in urban environments. *ISPRS J. Photogramm. Remote Sens.* 63 (1), 4–18.
- Cheng, L., Chen, S., Liu, X., Xu, H., Wu, Y., Li, M., Chen, Y., 2018. Registration of laser scanning point clouds: A review. *Sensors* 18 (5), 1641.
- Christofides, N., 1976. Worst-case analysis of a new heuristic for the travelling salesman problem. *Oper. Res. Forum* 3.
- Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C., 2009. *Introduction to Algorithms*. MIT Press.
- Couto, M., de Rezende, P., Souza, C., 2011. An exact algorithm for minimizing vertex guards on art galleries. *Int. Trans. Oper. Res.* 18, 425–448.
- Dantzig, G., Fulkerson, R., Johnson, S., 1954. Solution of a large-scale traveling-salesman problem. *J. Oper. Res. Soc. Am.* 2 (4), 393–410.
- Dehbi, Y., Leonhardt, J., Oehrlin, J., Haunert, J.-H., 2021. Optimal scan planning with enforced network connectivity for the acquisition of three-dimensional indoor models. *ISPRS J. Photogramm. Remote Sens.* 180, 103–116.
- Díaz-Vilariño, L., Frías, E., Previtali, M., Scaioni, M., Balado, J., 2019. Scan planning optimization for outdoor archaeological sites. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* XLII-2/W11, 489–494.
- Diestel, R., 2017. *Graph Theory*, fifth ed. Springer Publishing Company, Incorporated.
- Fishler, M.A., 1981. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24 (6), 381–395.
- Frías, E., Díaz-Vilariño, L., Balado, J., Lorenzo, H., 2019. From BIM to scan planning and optimization for construction control. *Remote Sens.* 11 (17), 1963.
- Heegaard, P.E., Trivedi, K.S., 2009. Network survivability modeling. *Comput. Netw.* 53 (8), 1215–1234. *Performance Modeling of Computer Networks: Special Issue in Memory of Dr. Gunter Bolch*.
- Jia, F., Lichti, D., 2019. A model-based design system for terrestrial laser scanning networks in complex sites. *Remote Sens.* 11 (15), 1749.
- Khoufi, I., Laouiti, A., Adjih, C., 2019. A survey of recent extended variants of the traveling salesman and vehicle routing problems for unmanned aerial vehicles. *Drones* 3 (3).
- 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 Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* V-4-2022, 129–136.
- Kröller, A., Baumgartner, T., Fekete, S.P., Schmidt, C., 2012. Exact solutions and bounds for general art gallery problems. *J. Exp. Algorithmics (JEA)* 17, 2.1–2.23.
- Kuipers, F.A., 2012. An overview of algorithms for network survivability. *CN* 2012, 24:24.
- Lee, D., Lin, A., 1986. Computational complexity of art gallery problems. *IEEE Trans. Inform. Theory* 32 (2), 276–282.
- 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.
- Menger, K., 1927. Zur allgemeinen kurventheorie. *Fund. Math.* 10 (1), 96–115, URL <http://eudml.org/doc/211191>.
- Meyer, T., Brunn, A., Stilla, U., 2022. Change detection for indoor construction progress monitoring based on BIM, point clouds and uncertainties. *Autom. Constr.* 141, 104442.
- Noichl, F., Lichti, D.D., Borrmann, A., 2024. Automating adaptive scan planning for static laser scanning in complex 3D environments. *Autom. Constr.* 165, 105511.
- O'Rourke, J., 1987. *Art Gallery Theorems and Algorithms*. Oxford University Press Inc., USA.
- Pferschy, U., Stanek, R., 2017. Generating subtour elimination constraints for the TSP from pure integer solutions. *Central Eur. J. Oper. Res.* 25, 231–260.
- Sen, A., Murthy, S., Banerjee, S., 2009. Region-based connectivity: a new paradigm for design of fault-tolerant networks. In: *Proceedings of the 15th International Conference on High Performance Switching and Routing. HPSR '09*, IEEE Press, pp. 94–100.
- Shirabe, T., 2004. A model of contiguity for spatial unit allocation. *Geogr. Anal.* 37, 2–16.
- Soudarissanane, S., Lindenberg, R., 2012. Optimizing terrestrial laser scanning measurement set-up. *ISPRS - Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* XXXVIII-5/W12, 127–132.

- Soudarissanane, S., Lindenbergh, R., Menenti, M., Teunissen, P., 2011. Scanning geometry: Influencing factor on the quality of terrestrial laser scanning points. *ISPRS J. Photogramm. Remote Sens.* 66 (4), 389–399.
- Theiler, P., Schindler, K., 2012. Automatic registration of terrestrial laser scanner point clouds using natural planar surfaces. *ISPRS Ann. Photogramm. Remote. Sens. Spat. Inf. Sci.* I-3, 173–178.
- Wujanz, D., Holst, C., Neitzel, F., Kuhlmann, H., Niemeier, W., Schwieger, V., 2016. Survey configuration for terrestrial laser scanning: Aufnahmekonfiguration für terrestrisches Laserscanning. *Allg. Vermessungs- Nachrichten: AVN; Z. Für Alle Bereiche der Geodäsie Und Geoinf.* 123.2016 (6), 158–169.