Optimal Laser Scan Planning for As-Built Modeling of Plant Renovations Using Mathematical Programming

E. Wakisaka[a], S. Kanai[b], and H. Date[b]

[a]Shinryo Corporation, Japan
[b]Hokkaido University, Japan
E-mail: wakisaka.ei@shinryo.com, kanai@ssi.ist.hokudai.ac.jp, hdate@ssi.ist.hokudai.ac.jp

Abstract –

In recent years, terrestrial laser scanners have been introduced to enable efficient as-built modeling. Since heating, ventilating, and air conditioning facilities often include many pipes and ducts packed into small spaces, it is difficult to manually determine optimal scanner placements that can capture their surfaces with high accuracy and quality and with few occlusions.

To solve this problem, we propose an optimal scan planning method based on mathematical programming that uses a coarse 3D model obtained from structure-from-motion as prior knowledge of the objects to be scanned. Integer programming enables us to identify optimal scanner locations that maximize scan coverage while satisfying general scan constraints. The proposed method can outperform experienced operators in terms of scan coverage and modeling accuracy. In addition, we extend our original method to address additional objectives and constraints encountered in practice, such as ensuring full scan coverage, minimizing travel time, and guaranteeing point cloud registration. We also confirm our methods' effectiveness via computer simulations.

Keywords –

Laser scanning; scan planning; Structure-from-motion; Next-best-view; As-built; Plant renovation

1 Introduction

Recently, increasing numbers of building facility renovations are being conducted in the heating, ventilating, and air conditioning (HVAC) industry. Laser scanning with terrestrial laser scanners (TLSs) is being used to reconstruct as-built three-dimensional (3D) models of these facilities, enabling shorter survey periods and in-depth construction planning. To reconstruct such 3D models to the precision required for renovation, the TLSs must be positioned appropriately and the scanning process must consider several different objectives and constraints. First, with a TLS, the scanning error depends on the scanning range and the incidence angle between the scanning beam and the surface to be scanned [1]. Second, the acceptable level of scanning error can vary for different parts of the facility depending on the scan's purpose. Finally, the aim is to maximize the scan coverage, with as few occlusions as possible, while respecting the first and second constraints.

Currently, however, the TLS positions are generally determined manually by experienced operators, so there is no guarantee that the scanner placements will satisfy the above scanning objectives and conditions.

To deal with these issues, in this paper, we propose a new optimal laser scan planning method for TLSs that utilizes a coarse 3D model reconstructed via a structure-from-motion (SfM) technique as prior knowledge. It uses integer programming to find optimal scanner placements that maximize scan coverage while satisfying the beam incidence angle and scanning range constraints. We also propose ways to address three further objectives and constraints encountered in practice: providing full scan coverage, minimizing travel time to increase scanning efficiency, and guaranteeing point cloud registration. To achieve this, we develop three extensions to the original method. Finally, we evaluate our methods' effectiveness via experiments and computer simulations.

2 Related Work

In the reverse engineering and robotics fields, the problem of automatically determining optimal sensor placements and measurement sequences is known as the *next-best-view* (NBV) problem [2]. NBV problems can be divided into two types: ones with prior knowledge about the objects to be scanned and ones without. It is known that better sensor placements can be found in the former case [3] than in the latter.

Several NBV problem studies have considered optimal laser scan planning problems involving prior knowledge. Soudarissanane et al. [4] and Ahn et al. [5]

proposed scanner placement estimation methods that maximize the building's measurable wall length using greedy methods, based on 2D drawings of the building's interior and exterior. However, since their methods treat scanner placement as a simple two-dimensional (2D) optimization problem, it is difficult to ensure the scanner placements provide sufficient scan coverage, minimizing occlusions, when applied to scanning plant facilities involving complex, 3D structures with TLS.

Turning now to 3D optimization approaches, Kitada et al. [6] and Zhang et al. [7] proposed methods for deriving optimal scanner placements that maximize scanned surface coverage. These apply integer programming [6] or full search [7] to a 3D model of the building's exterior, but they do not consider the beam incidence angle and scanning range constraints. In addition, these studies did not compare the optimal scanner placements found by their algorithms with those determined by experienced operators in terms of scanning efficiency and quality for as-built modeling.

In our previous study [8], we also proposed a method of solving the optimal TLS planning problem that uses the mesh model representation of an SfM model as prior knowledge. Our method formulates the problem of maximizing the number of measurable surfaces while satisfying the incidence angle and scanning range constraints as a 0–1 integer programming problem, deriving the optimal scanner placements from its solution. However, this method cannot provide full scan coverage, minimize travel time, or guarantee point cloud registration.

To address these issues, in this paper, we extend our previous approach [8] so as to derive the additional scan positions needed for full scan coverage, optimize the scan sequence to minimize travel time, and generate scanner placements that guarantee point cloud registration.

3 Planning Optimal Scans Using Mesh Models And Integer Programming

3.1 Algorithm Overview

In this section, we provide a brief overview of our previous planning algorithm [8]. As shown in Figure 1, the first step is to generate a coarse 3D SfM model from photos of the facility to be modeled. Next, since the required scan quality can differ substantially depending on the area and construction type, we assign a "scan significance level" to each region to specify the quality needed, which is later used to determine the constraint level. Each of the SfM model's surfaces is assigned one of three scan significance levels: *high*, *medium*, or *low*.

During the second step (Figure 2) the space enclosing the SfM model is decomposed into a set of

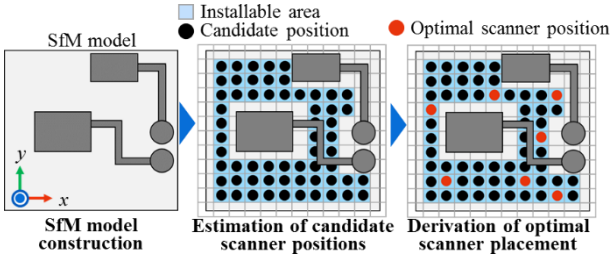

Figure 1. Outline of our optimal scan planning algorithm based on mesh-based integer programming

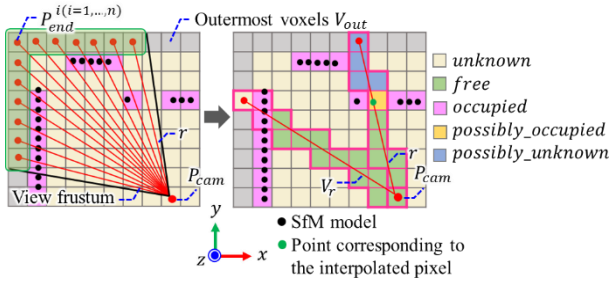

Figure 2. Spatial occupancy classification

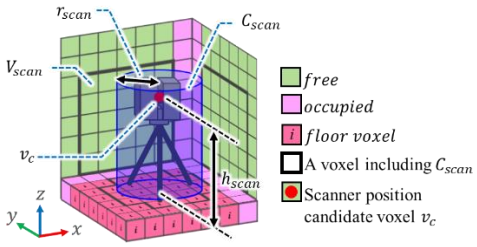

Figure 3. Candidate scanner positions

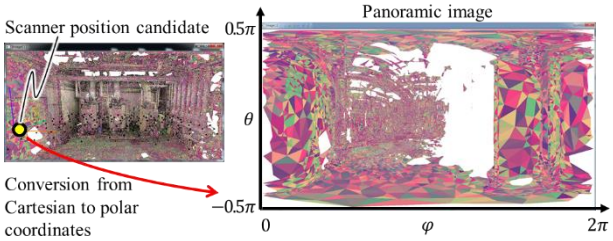

Figure 4. Rendering of an SfM model's surfaces as a panoramic image

voxels with a spatial resolution of l_v, and any voxels that include one of the SfM model's surfaces have their space occupancy attributes set to *occupied*. Next, rays are cast from each camera position toward the centroid of the outermost voxel P_{end} included in that camera's view frustum. Based on these results, the space occupancies of the remaining voxels are set to one of the remaining three types: *free*, *possibly_occupied*, and *possibly_unknown*. Then, as shown in Figure 3, based

on the voxels' space occupancy classifications, a set of candidate scanner positions v_c is extracted from the voxel space enclosing the SfM model.

The final step is to reduce the optimal scan planning problem to a 0–1 integer programming problem. First, we calculate the observation matrix $A = \{a_{v_c,f}\}$, where $a_{v_c,f}$ indicates whether surface f of the SfM model is visible from scanning position v_c. To efficiently determine the visibility, we render the sections of each surface of the SfM model in different colors on a spherical image generated from v_c (Figure 4). Then, surface f is visible if the color used to render it remains in the image produced by the graphics processing unit (GPU). If f also satisfies the incidence angle and scanning range conditions, it is labeled as *observable* $(a_{v_c,f} = 1)$. By repeating this process for all $v_c \in V_c$, we can generate the observation matrix A.

Finally, we derive the optimal scanner placements by integer programming, where the indicator variables represent whether or not a scanner should be placed at v_c. Here we use the Numerical Optimizer [9] package, with the branch and bound algorithm.

3.2 Performance

We compared the performance of our original method with that of experienced operators for a room containing a machine acting as a heat source (12.1 × 14.1 × 4.6 m^3) in terms of the number of scans, scan coverage ratio, modeling accuracy, and processing time. The scanning conditions and threshold constraint values were as shown in Tables 1 and 2, respectively. The optimal laser scanner positions found by both methods are shown in Figure 5.

These results confirm that the proposed method yielded higher scan coverage than the human operator could achieve. The geometric errors in the point cloud-based model captured with the optimal scanner positions were less than 5 mm, sufficient for practical use, while those in the model captured using the operator's placements were more than 7 mm and it is too high for the model to be useful. Furthermore, our method was able to generate optimal scanner positions in only a few minutes, giving it a significant advantage for use in practical scanning tasks.

4 Addressing Practical Constraints

4.1 Issues and Resolutions

Unfortunately, several issues still remain when applying the original optimal scan planning algorithm (Section 3) in practice, namely the following.

- When scanning complex facilities, the use of 2D candidate scanner positions, with the scanner placed at a constant height on a tripod, means that un-scanned SfM model surfaces often remain near ceilings, regardless of the scanner positions used.
- Since the sequence in which the scans are performed does not affect the coverage or scan quality, the sequence used is left to the operator's judgment. However, when scanning large-scale facilities, it is difficult for unskilled operators to move from one position to the next efficiently without getting lost.
- Since no constraints are imposed on the overlaps between point clouds from different scans, the scanner positions generated do not necessarily ensure point cloud registration.

To address these issues, in this study, we extend our original method in the following ways, then evaluate the effectiveness of these extensions in computer simulations.

1. We formulate an additional planning problem using integer programming to derive additional and optimal 3D scanner placements that can achieve full scan coverage with the minimum number of scans.

Table 1. Scanning parameters

Parameter	Value
Scanner height (h_{scan})	1.4 m
Scanner base radius (r_{scan})	0.3 m
Vertical field of view	320°
Horizontal field of view	360°
Vertical and horizontal scan pitch	0.072°

Table 2. Scanning constraints

Constraint	Scanning significance level		
	High	Medium	Low
Max. incidence angle	45°	90°	90°
Min. scan range	0.3 m	0.3 m	0.3 m
Max. scan range	5.0 m	8.0 m	20.0 m

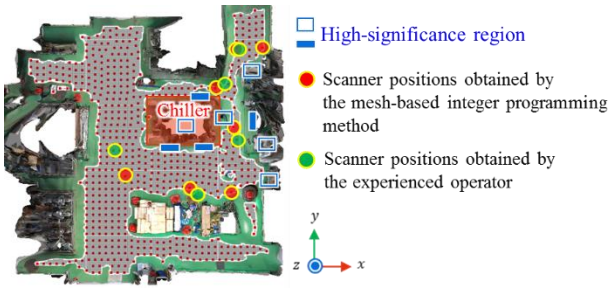

Figure 5. Scanner positions in a room with a machine-based heat source

2. We formulate an optimal scan ordering problem, deriving the shortest distance needed to travel among the scanner positions from an instance of the *traveling salesman problem*.
3. We formulate scan planning as a constraint satisfaction problem to derive optimal scanner placements that guarantee point cloud registration.

4.2 Full-Coverage Scan Planning Algorithm

4.2.1 Generating Additional Scanner Positions

First, based on the constraints given in Table 2, we define a view frustum (Figure 6) such that the viewpoint is located at the centroid f_u of the unmeasured surface, with the view direction directed toward the surface's normal vector n_i, the vertical and horizontal fields of view being $2\theta_\alpha$, and the *near* and *far* planes being d_{min} and d_{max}, respectively. Then, *free* or *possibly_occupied* voxels whose centroids are included in the view frustum are extracted as additional candidate scanner positions $v_s \in V_{scanner}$.

4.2.2 Checking Visibility and Generating the Observation Matrix

As with the original method [8], we formulate the problem of planning additional scans as a 0–1 integer programming problem. Therefore, we again need to generate an observation matrix $B = \{b_{f_u,v_s}\}$, where b_{f_u,v_s} indicates whether or not the un-scanned surface f_u is observable from scanning position v_s.

To efficiently evaluate these element values, we generate spherical images of the SfM model's surfaces as seen from each position v_s as before, labeling each surface as *observable* $(b_{f_u,v_s} = 1)$ or *unobservable* $(b_{f_u,v_s} = 0)$ using pixel-wise visibility checks and scan quality checks that consider the beam incidence angle and scan range. By repeating this process for all $v_s \in V_{scanner}$, we can generate the observation matrix B.

4.2.3 Generating Additional Scanning Positions

Using the observation matrix B (Section 4.2.2), we formulate the additional scanner placement problem as the following 0–1 integer programming problem.

$$\text{minimize}_{z_{v_s}} \quad \sum_{v_s \in V_s} z_{v_s}(1 + \alpha h_{v_s}) \qquad (1)\text{-}1$$

$$\text{subject to} \quad \sum_{v_s \in V_{scanner}} b_{f_u,v_s} z_{v_s} \geq 1 \ (f_u \in F_u) \qquad (1)\text{-}2$$

$$z_{v_s} \in \{0,1\} \ (v_s \in V_s) \qquad (1)\text{-}3$$

Here, the terms are defined as follows:
$v_s \in V_s$: a candidate additional scanning position,
$f_u \in F_u$:an unmeasured triangular SfM model surface,

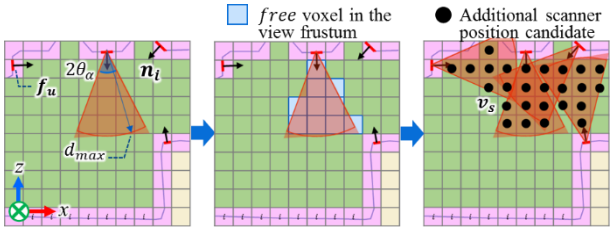

Figure 6. Generating additional scanner position candidates

Figure 7. Distribution of the unmeasured high-significance surfaces

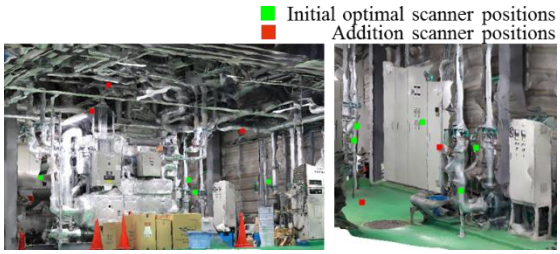

Figure 8. Distribution of additional scanner positions

$$z_{v_s} = \begin{cases} 1 & \text{if } v_s \text{ is used as a scanning position,} \\ 0 & \text{if } v_s \text{ is not used as a scanning position,} \end{cases}$$

$$x_f = \begin{cases} 1 & \text{if } f \text{ is measured,} \\ 0 & \text{if } f \text{ is not measured,} \end{cases}$$

and

h_{v_s}: z coordinate of v_s.

Here, the objective function (Equation (1)) aims to minimize both the number of additional scan positions and their z-coordinates, so lower v_s positions will tend to be selected as additional scan locations.

Finally, we derive the solution to this optimization problem by integer programming. The centroids of the voxels v_s for which $z_{v_s} = 1$ are adopted as additional scanning positions, and added to the set of scanner positions Z_{opt} derived in Section 3.

4.2.4 Results

We applied the above algorithm for planning additional scans to the room used for the previous evaluation (Section 3.2). Figure 7 shows the high-significance surfaces that could not be measured using only the previous scanner positions. The parameters and

constraints were as before (Tables 1 and 2, respectively).

Here, 2,258 additional candidate scanner positions v_s were considered, from which the proposed algorithm extracted the five positions shown in Figure 8. These included several high locations, needed to scan several pipes installed near the ceiling. We were able to confirm that the scan coverage reached 100% with these additional scanner positions added. The time required to solve the optimization problem was 9.6 min, which is reasonable for on-site planning.

The TLS unit used for these experiments weighed 9.8 kg, and its body was 170 mm wide, 286 mm deep, and 286 mm high. This made it difficult to position at the heights required on a tripod due to instability. Therefore, for these additional scans, another laser scanner would be needed that is smaller, lighter, and has a tripod capable of greater extension.

4.3 Optimal Scan Ordering Algorithm

4.3.1 Estimating the Passable Areas

First, the areas of the SfM model through which the operator can travel are extracted using the voxels' space occupancy attributes (Section 3.1).

This is achieved by approximating the shape of a human body as a closed cylinder C_{human} with a radius of r_{human} (Figure 9), and determining the minimum connected voxel set V_{human} that can include C_{human}. As shown in Figure 9, if the ratio of the area of the projected floor voxels (M) to that of V_{human} projected along the z-axis (N) is greater than the threshold value τ_{place}, the floor voxel is classified as *passable*. This condition can be represented as

$$M/N \geq \tau_{place}, \tag{2}$$

where we set $\tau_{place} = 0.80$ in this paper.

4.3.2 Constructing the Path Graph

Generating the optimal scan ordering that achieves the shortest possible travel distance among the given scanner positions can be formulated as an instance of the traveling salesman problem. In this problem, the costs, namely the distances $dist_{i,j}$ between pairs of scanner positions i, j, have to be evaluated and assigned. With this in mind, we construct candidate operator routes as path graphs.

First, as shown in Figure 10, the passable voxels are projected onto a horizontal plane and the centroids of passable voxels are designated as nodes, with pairs of adjacent nodes connected by edges. Then, the set of optimal scanner positions Z_{opt} is projected onto this horizontal plane and the closest nodes are regarded as the scanner nodes. Finally, the shortest path between each pair of scanner nodes, i.e., the path that minimizes

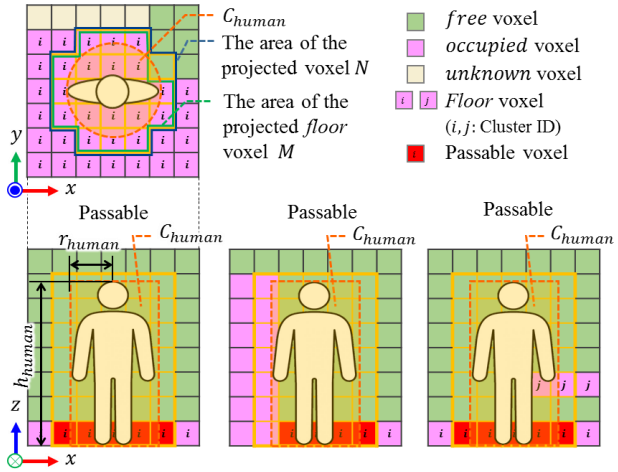

Figure 9. Estimating passable area

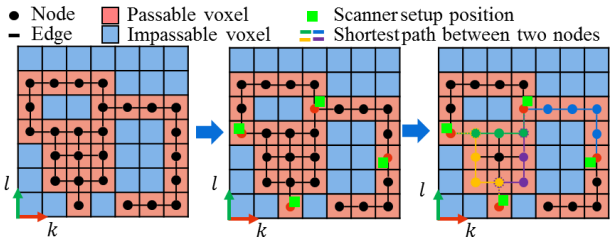

Figure 10. Constructing path graph

the distance between them, is evaluated using Dijkstra's algorithm. Here, the distances between scanner nodes are calculated in terms of the Manhattan distance. This enables us to define all the distances $dist_{i,j}$ between pairs of scanner nodes.

4.3.3 Calculating the Shortest Path and Optimal Scanning Order

The optimal scan ordering problem can be formulated as the following traveling salesman problem.

$$\text{minimize}_{x_{i,j}} \quad \sum_{i \in V_{opt}} \sum_{j \in V_{opt}} dist_{i,j}\, x_{i,j} \tag{3}\text{-1}$$

$$\text{subject to} \quad \sum_{j \in V_{opt}} x_{i,j} = 1 \left(\forall i \in V_{opt}\right) \tag{3}\text{-2}$$

$$\sum_{i \in V_{opt}} x_{i,j} = 1 \left(\forall j \in V_{opt}\right) \tag{3}\text{-3}$$

$$y_i - y_j + (n-1)x_{i,j} \leq n-2 \quad \left(\forall i, j \in V_{opt} \setminus \{1\}, i \neq j\right) \tag{3}\text{-4}$$

$$x_{i,j} \in \{0,1\} \left(\forall i, j \in V_{opt}\right) \tag{3}\text{-5}$$

Here, the terms are defined as follows:
$i \in V_{opt}$: a scanner node,
$x_{i,j} = \begin{cases} 1 & \text{if the path traverses from nodes } i \text{ to } j, \\ 0 & \text{otherwise,} \end{cases}$

$dist_{i,j}$: the distance from nodes i to j,
and
y_i: a dummy variable.

Finally, we can solve this optimization problem using integer programming.

4.3.4 Results

We used the above scan ordering optimization algorithm to plan the scanning sequence for the machine room example considered in Sections 3.2 and 4.2.4. The eight scanner nodes shown in Figure 11 had previously been derived using the original method [8]. Here, the numbers indicate scanner position IDs. In this case, since the entrance and exit were located at the lower left of the room, this was selected as the start point S, and a route derived that started and ended at *S*. For this experiment, we used $r_{human} = 0.3$ m and $h_{human} = 1.6$ m.

The optimal route is shown in Figure 11. Conducting the scans in the order $S \rightarrow 1 \rightarrow 5 \rightarrow 7 \rightarrow 6 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow S$ gave us a minimum travel distance of 35.1 m. Constructing path graph process (Section 4.3.2) took 309 s, and the optimization process (Section 4.3.3) took 0.4 s, meaning this approach could also be used for on-site scan planning.

Being able to calculate such optimal routes could be very helpful in practice, as it could enable unskilled operators to complete scans efficiently without getting lost.

4.4 Planning Optimal Scans that Guarantee Point Cloud Registration

4.4.1 Problem Setting

Point cloud registration is the process of aligning overlapping scanned points or point clouds with common target markers. It is essential to ensure point cloud registration in laser scanning to create as-built models, so we must generate scanning plans that guarantee point cloud registration. In this paper, we simplify this problem by using target markers to assist with point cloud registration when planning the scans.

In particular, we enforce the following two constraints to enable point cloud registration.

1. As shown in Figure 12, at least two common target markers must be visible from any two different scanner positions.
2. The registration graph must be fully connected. This graph defines the scanning positions as nodes, and has edges between positions that satisfy Condition 1.

Here, the proposed scanner placement method only attempts to satisfy Condition 1 because it is almost satisfied during the scanning of an open room with

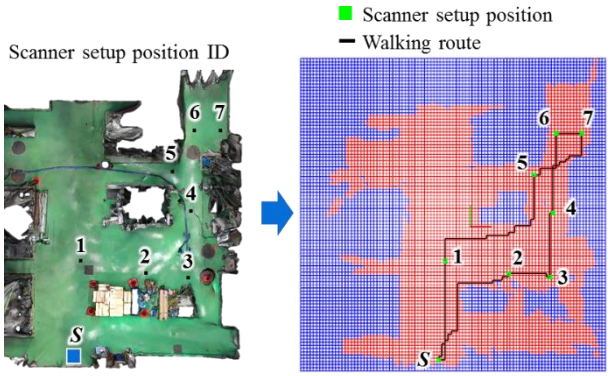

Figure 11. Shortest route and scanning order

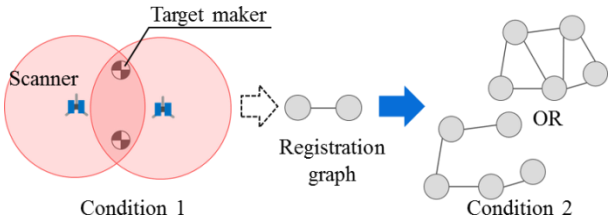

Figure 12. Conditions required to create a point cloud registration graph

plenty of planes. Conversely, when aligning plurality of rooms, second Condition 2 must be satisfied. Therefore, we plan to deal with Condition 2 in the future.

4.4.2 Selecting Planar Regions

First, the operator interactively designates planar areas in the SfM model as target markers, and assigns each one a marker ID.

4.4.3 Checking Visibility and Generating the Registration Matrix

Again, our approach is similar to that used for the original method [8]. First, we construct an observation matrix $A = \{a_{i,f}\}$ describing whether or not the surface f can be observed from the candidate scanner position i, in the same way as in Section 3.1.

In addition, we also calculate the registration matrix $G = \{g_{i,j}\}$ describing whether or not given pairs of scanner positions i, j enable registration between their point clouds. Here, we first calculate the marker observation matrix $E = \{e_{i,m}\}$ describing whether or not marker m can be observed from scanner position i. Next, as shown in Figure 13, if the marker plane f_m passes the pixel-wise visibility check (Section 4.2.2) then it is labeled as *visible*. If f_m also satisfies the incidence angle and scanning range conditions given in Equations (4) and (5), it is labeled as *observable*.

$$ang\{l(f_m, i), \boldsymbol{n_f}(f_m)\} < \theta_m \quad (4)$$

$$d_{mmin} < dist(f_m, i) \leq d_{mmax} \quad (5)$$

$$f_m / F_M \geq \tau_m \quad (6)$$

Then, as described in Equation (6), if the ratio of observable plane elements f_m to the complete set of plane elements F_M constituting the marker is larger than τ_m, we set $e_{i,m}$ to 1, i.e., observable.

Next, we calculate the inner product of the row vectors $\boldsymbol{e_i}$ and $\boldsymbol{e_{i'}}$ in the marker observation matrix E, which correspond to the i-th and i'-th scanner positions. When $\boldsymbol{e_i} \cdot \boldsymbol{e_{i'}} \geq 2$, the point clouds scanned at these two positions have at least two markers in common, so point cloud registration is possible and we set $g_{i,i'} = 1$:

$$g_{i,i'} = \begin{cases} 1 \text{ if } & \boldsymbol{e_i} \cdot \boldsymbol{e_{i'}} \geq 2, \\ 0 \text{ if } & \boldsymbol{e_i} \cdot \boldsymbol{e_{i'}} < 2. \end{cases} \quad (7)$$

4.4.4 Generating Optimal Scanner Placements that Allow for Registration

Using the registration matrix G (Section 4.4.3), we can generate optimal scanner placements that ensure point cloud registration as follows.

$$\begin{aligned}
&\underset{x_f, z_i}{\text{maximize}} && \sum_{f \in F} x_f && (8)\text{-}1 \\
&\text{subject to} && \sum_{i \in V_c} z_i \leq T && (8)\text{-}2 \\
& && \sum_{i \in V_c} a_{i,f} z_i \geq x_f \ (\forall f \in F) && (8)\text{-}3 \\
& && \sum_{i \in V_c} z_i z_{i'} g_{i,i'} \geq 1 \ (\forall i' \in V_c) && (8)\text{-}4 \\
& && z_i \in \{0,1\} \ (\forall i \in V_c) && (8)\text{-}5 \\
& && x_f \in \{0,1\} \ (\forall f \in F) && (8)\text{-}6
\end{aligned}$$

Here, the terms are defined as follows:
$i \in V_c$: a candidate scanning position,
$f \in F$: a triangular surface with high significance in the SfM model,
T: the upper limit on the number of scans,

$$z_i = \begin{cases} 1 & \text{if position i is adopted,} \\ 0 & \text{if position } i \text{ is not adopted,} \end{cases}$$

$$x_f = \begin{cases} 1 & \text{if } f \text{ is measured,} \\ 0 & \text{if } f \text{ is not measured,} \end{cases}$$

$$a_{i,f} = \begin{cases} 1 & \text{if } f \text{ is observable from } i, \\ 0 & \text{if } f \text{ is not observable from } i, \end{cases}$$

and

$$g_{i,i'} = \begin{cases} 1 & \text{if positions } i \text{ and } i' \text{are aligned,} \\ 0 & \text{if positions } i \text{ and } i' \text{ are not aligned.} \end{cases}$$

In the formulation of 4.2 and 4.3, as the objective function and constraints can be expressed as linear functions, 4.2 and 4.3 can be handled as an integer programming problem. Therefore, an optimal solution can be derived through the simplex method. In contrast,

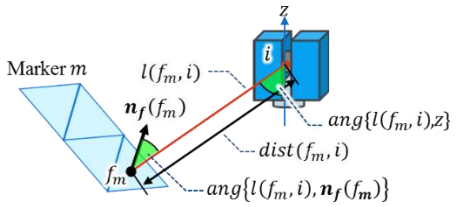

Figure 13. Checking the visibility of the surface f_m containing the marker

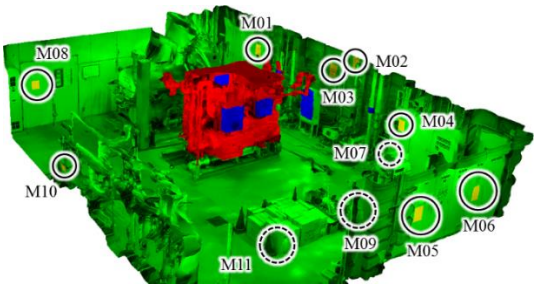

Figure 14. Planes selected as target markers

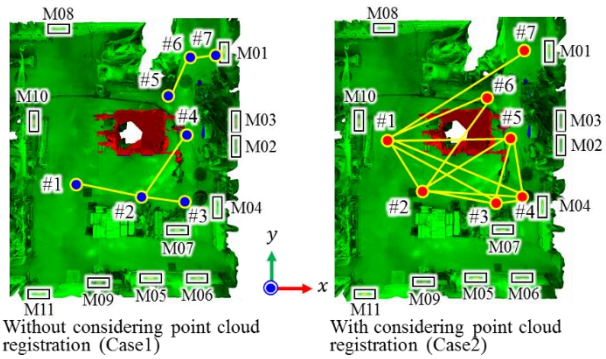

Figure 15. Scanner position pairs that enable point cloud registration

as the registration constraint in the formulation of 4.4 needs to be expressed by a quadric function, it cannot be handled as an integer programming problem. Therefore, we treat it as a constraint satisfaction problem, and derive a solution using tabu search, which is a metaheuristic [10], [11]. This algorithm solves the problem of minimizing total penalty for the violation of each constraint. Because this algorithm is an approximate solver, even if there is no solution that satisfies all constraints, we can derive a solution that satisfies the constraint to the maximum possible extent.

Finally, the scanner positions i for which $z_i = 1$ were added to the set of optimal scanning positions Z_{opt}.

4.4.5 Results

We again applied the algorithm to the machine room used in our previous evaluations (Sections 3.2, 4.2.4, and 4.3.4). As shown in Figure 14, eleven planes were

chosen as being representative of the given area, excluding the floor, and were selected as target markers. The same parameters and conditions were used as before (Tables 1 and 2). In addition, since the target markers had to be scanned with high precision, the conditions for Equations (4) and (5) were taken to be $\theta_m = 45°, d_{mmin} = 0.3\ m, d_{mmax} = 10.0\ m,$ and $\tau_m = 0.8$.

We compared the scanner placements obtained by the proposed method (Case 2) with those obtained via the original integer programming method (Section 3) that does not consider registration, with $T = 7$ (Case 1).

The resulting scanner placements are shown in Figure 15. The scan coverage of the high-significance areas was 85.0% in Case 2, which was 3.2% lower than in Case 1 (88.2%). We also confirmed that the proposed scanner setup (Case 2) yielded a fully-connected registration graph, with every point cloud pair including at least two markers that were visible from both clouds and it being possible to connect all the point clouds to each other by a sequence of registrations. By contrast, in Case 1, the registration graph was not fully connected and the scanner positions were divided into two connected components, making it impossible to register point clouds in different components.

5 Conclusions

In this paper, to ensure that the laser scans used for as-built modeling are sufficiently complete, efficient, and reliable for use in plant renovation, we have proposed three optimal scan planning methods that extend our original method.

First, we confirmed that we could achieve 100% scan coverage using the additional scanner positions generated by our extended planning method. Next, we confirmed that we could derive the scan order that minimized the total travel distance using our scan ordering method. Finally, we confirmed that the scanner positions generated by our registration-aware method ensured that every pair of point clouds included at least two markers that were visible from both clouds and that all the point clouds could be connected via a sequence of registrations.

In future work, we plan to introduce conditions on the scan overlap to ensure global registration in the optimal scan planning problem, and also to develop a navigation system that can present the scanner setup positions without requiring any survey instruments.

References

[1] Soudarissanane S., Lindenbergh R., Menenti M. and Teunissen P. Incidence angle influence on the quality of terrestrial laser scanning points, *International Archives of ISPRS*, 183–188, 2009.

[2] Scott W.R., Roth G. and Rivest J.F. View planning for automated three-dimensional object reconstruction and inspection, *ACM Computing Surveys (CSUR)*, 35(1):64–96, 2003.

[3] Munkelt C., Kühmstedt P. and Denzler J. Incorporation of a-priori information in planning the next best view, *International Archives of ISPRS*, XXXVI-5, 37–42, 2006.

[4] Soudarissanane S. and Lindenbergh R. Optimizing terrestrial laser scanning measurement set-up, *International Archives of ISPRS*, XXXVIII-5/W12, 127–132, 2011.

[5] Ahn J. and Wohn K. Interactive scan planning for heritage recording, *Multimedia Tools and Applications*, 1–21, 2016.

[6] Kitada Y., Dan H. and Yasumuro Y. Optimization scenario for 3D-scanning plans of outdoor constructions based on SFM, *Proceedings of CONVR 2015*, 65–68, 2015.

[7] Zhang C., Kalasapudi V. S. and Pingbo T. Rapid data quality oriented laser scan planning for dynamic construction environments, *Advanced Engineering Informatics*, 30(2):218–232, 2016.

[8] Wakisaka E., Kanai S. and Date H. Optimal laser scan planning of terrestrial laser scanner for as-built modeling of HVAC Systems, *Journal of the Japan Society for Precision Engineering*, 84(8):738–745, 2018.

[9] Numerical Optimizer. Online: https://www.msi.co.jp-/nuopt/. Accessed: 13/12/2018.

[10] Nonobe K. and Ibaraki T. a tabu search approach for the constraint satisfaction problem as a general problem solver, *European Journal of Operational Research*, 106:599–623, 1998.

[11] Nonobe K. and Ibaraki T. An improved tabu search method for the weighted constraint satisfaction problem, *INFOR* 39:131–151, 2001.