



## Article

# A Practical Algorithm for the Viewpoint Planning of Terrestrial Laser Scanners

Fengman Jia \*  and Derek D. Lichti 

Department of Geomatics Engineering, University of Calgary, 2500 University Drive NW, Calgary, AB T2N 1N4, Canada; ddlichti@ucalgary.ca

\* Correspondence: fengman.jia@ucalgary.ca

**Abstract:** Applications using terrestrial laser scanners (TLS) have been skyrocketing in the past two decades. In a scanning project, the configuration of scans is a critical issue as it has significant effects on the project cost and the quality of the product. In this paper, a practical strategy is proposed to resolve the problem of the optimal placement of the terrestrial laser scanner. The method attempts to reduce the number of viewpoints under the premise that the scenes are fully covered. In addition, the approach is designed in a way that the solutions can be efficiently explored. The method has been tested on 540 polygons simulated with different sizes and complexities. The results have also been compared with a benchmark strategy in terms of the optimality of the solutions and runtime. It is concluded that our proposed algorithm ties or reduces the number of viewpoints in the benchmark paper in 85.6% of the 540 tests. For complex environments, the method can potentially reduce the project cost by 10%. Although with relatively lower efficiency, our method can still reach the solution within a few minutes for a polygon with up to 500 vertices.

**Keywords:** viewpoint planning; network design; terrestrial laser scanning; art gallery problem



**Citation:** Jia, F.; Lichti, D.D. A Practical Algorithm for the Viewpoint Planning of Terrestrial Laser Scanners. *Geomatics* **2022**, *2*, 181–196. <https://doi.org/10.3390/geomatics2020011>

Academic Editor: Filiberto Chiabrando

Received: 14 March 2022

Accepted: 20 April 2022

Published: 22 April 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

With the change in the variety and complexity of the modern survey tasks in recent decades, surveying techniques have advanced rapidly as well. Light Detection And Ranging (LiDAR) systems, represented by terrestrial laser scanners, represent a new technique that can provide rapid and high-density data acquirement. Thus, LiDAR has been widely accepted in various fields of application such as civil engineering surveying [1,2], archaeology surveying [3,4], robotic mapping [5,6], and environment monitoring [7,8].

A surveying project is usually conducted in three stages: network design, network execution, and network analysis [9]. A well-designed network is the core to the success of the project. A network should be designed in terms of the surveying datum, the equipment and observation, the network enhancement, and most importantly, the network configuration. The focus of this research is to find the optimal network configuration for terrestrial laser scanning surveys, which should usually meet the following requirements:

1. Coverage. Regardless of the complexity, all components of the scene should be fully scanned;
2. Number of viewpoints. This is of great significance as the number of scans directly affects the project cost in terms of the time and labor in both the field and office works;
3. Viewpoint locations. The environment should be scanned with strong geometry so that the point cloud meets quality requirements. In addition, sufficient overlap between scans should be provided to allow for registration; and
4. Design efficiency. The program should not take hours to process a design. Preferably, it should provide the plan within a few minutes, so an on-site design is possible.

Finding a design that meets the above requirements is known as the viewpoint planning problem. Although some attention has been focused on this problem, it remains an open issue and is still often empirically resolved based on an operator's experience.

In this paper, we propose a practical viewpoint planning algorithm for terrestrial laser scanners. Our method determines the viewpoint locations for a given environment and achieves the goals of full coverage and a minimum number of scans within a few minutes. The main contributions of this research are summarized as follows:

1. The planning strategy is proposed with two tools, the visibility analyzer and the optimization solver.
  - By constructing the visibility polygon, the visibility heatmap, and the visibility matrix, the visibility analyzer guarantees the complete scans of the scene.
  - The optimization solver is integrated with an optimization method previously proposed by the authors. It is applied on the visibility analysis results in exploring the optimal/near-optimal solution.
2. The proposed method is validated with 540 simulated polygons that vary in type, size, and complexity, which confirms its versatility and robustness. The results are also compared with available benchmark paper solutions.
3. The computation time in processing different polygons has been tracked to evaluate the efficiency of the method. Our method ensures a scanning plan for common scenes in the real world within a few minutes.
4. The main scientific contribution of this paper is that the method proposed for viewpoint planning is tested on massive random scenarios, which validates the method and provides a solid foundation for its further investigation and improvement.

The remainder of this paper is structured as follows. The research work related to this paper is reviewed in Section 2. The proposed viewpoint planning strategy is explained in detail in Section 3. A comprehensive comparison is made in Section 4 between our solutions and a benchmark paper. The performances are compared in terms of the optimality and the computational efficiency. Finally, conclusions and some future topics are discussed in Section 5.

## 2. Background Research

Two categories of related research are reviewed in this section: (1) the Art Gallery Problem and (2) TLS viewpoint planning methods.

### 2.1. Art Gallery Problem

The Art Gallery Problem (AGP) is concerned with how to position as few guards as possible to ensure that an art gallery represented by a non-self-intersecting polygon would be fully guarded, assuming the guard's field of view covers 360 degrees as well as an unbounded distance [10]. The original Art Gallery Problem was firstly proposed by Victor Klee in 1973, and has been developed into many variants since then. For example, if the guards are fixed at the polygon vertices, it is an AGP with vertex guards. Similarly, if the guards can be freely placed at any point on the edge or anywhere within the polygon, it is the AGP with edge guards, or the AGP with point guards.

Over the past 15 years, the Art Gallery Problem and its variants have been most extensively investigated by the research group at State University of Campinas Institute of Computing. The earliest research can be traced back to 2007. Since then, the group has published about ten papers regarding this problem. In [11–16], several exact algorithms were progressively proposed by discretizing the problem to a Set Cover Problem, then having it solved with the Integral Linear Programming (ILP). However, their research has mainly focused on the AGP with vertex guards. Refs. [10,17,18] have extended the algorithm to a more general AGP with point guards. They used a similar ILP scheme but improved the previous method by representing the polygon with finite points called witness candidates, and by iteratively adding additional witnesses to compensate the initial solution.

## 2.2. TLS Viewpoint Planning Methods

The study of the Art Gallery Problem and its variants also has value in more practical areas in the real world. For example, the method can be applied to the optimal placement of the surveying sensors. Surveying using terrestrial laser scanners seeks the minimum number of viewpoints (VPs) to record a site. However, the biggest difference between this task and the Art Gallery Problem is that a set of acquisition constraints and requirements must be satisfied. Constraints must be imposed on the instrument locations so that the data quality meets project specifications. According to [19], point cloud quality is largely influenced by the instrument capability and the scanning geometry. Two factors that define the scanning geometry need to be constrained when planning a scanning survey: the range and the incidence angle.

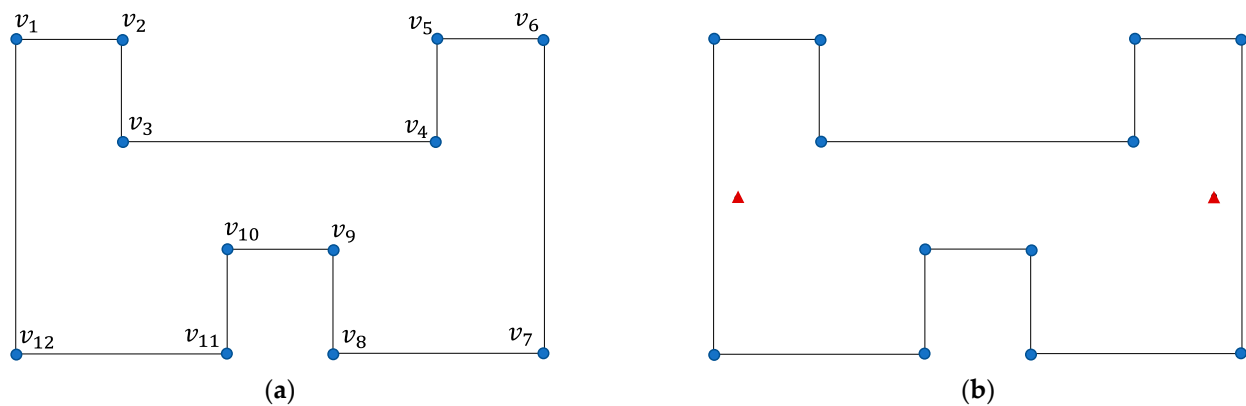
Contributions in the above paper were brought forth by several researchers. Ref. [20] evaluated the visibility of the discretized environment model based on the scanning geometry and searched for the optimal solution using the well-known heuristic optimization method of greedy algorithm. Ref. [21] utilized a similar strategy by iteratively adding up the viewpoints that scanned the most wall lengths. However, the viewpoints were selected interactively and could be adjusted by the user. Refs. [22,23] implemented several other heuristic methods on planning a relatively small environment, including the simulated annealing method, the genetic algorithm, and the particle swarm optimization. Ref. [24] proposed an improved greedy algorithm by adding a weighting scheme into the optimality assessment. Research with a similar workflow but different weighting strategy can be found in [25]. A further improvement was introduced in [26], which presented a hierarchical planning strategy to speed up the “brute force” searching process. Ref. [27] advanced the previous contributions with the floor coverage and the scan overlap being considered. Ref. [28] introduced a measurement of the surface topography into the planning system to fulfill the overlap requirement of the surface-based registration.

The research reviewed above assumes the existence of a prior model of the environment. Another type of viewpoint planning problem tends to manage the planning and modelling simultaneously. Known as the Next Best View (NBV) problem, it can be resolved without the knowledge of the object scene, and selects the next viewpoint based on a certain criterion with the information collected or predicted from the existing scans. Research focused on this topic can be found in [29–32].

The Art Gallery Problem has been extensively investigated and the above-mentioned algorithms have been tested over massive polygon instances, and the results are available for comparison [33]. However, the scanning geometry constraints have not been included. The viewpoint planning problem research for terrestrial laser scanners has achieved some progress, but the methods were mostly applied on certain environments and might only meet restricted specifications. In this research, we propose a method that has been widely validated on the benchmark data and can also be extended to solve the viewpoint planning problem.

## 3. Methods

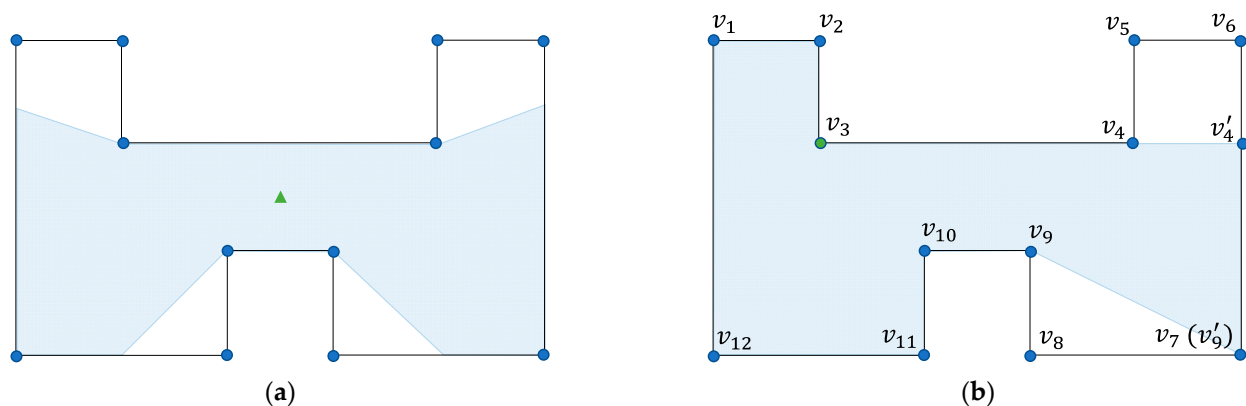
The goal of this research is to propose a practical methodology that can minimize the number of viewpoints for full coverage of an indoor environment. By representing an indoor environment with a polygon  $\mathcal{P}$  of  $n$  vertices, the solution of the method is a set of viewpoints that covers all the edges, i.e., walls. The polygon in Figure 1a is used as an example to introduce the proposed method in this section. It consists of twelve vertices from  $v_1$  to  $v_{12}$ , and two viewpoints represented by red triangles in Figure 1b have been found as the solution. Detailed steps in achieving this solution are presented in the following content.



**Figure 1.** An example. (a) Input polygon; (b) viewpoint solution.

### 3.1. Vertex Visibility Polygons

Starting from Figure 1a, the first step is to construct the visibility polygons. In most existing methods, the visibility polygon represents the visible area from an arbitrary point within the site polygon, as depicted in Figure 2a. In this way, the visibility test is required once a new viewpoint is generated.

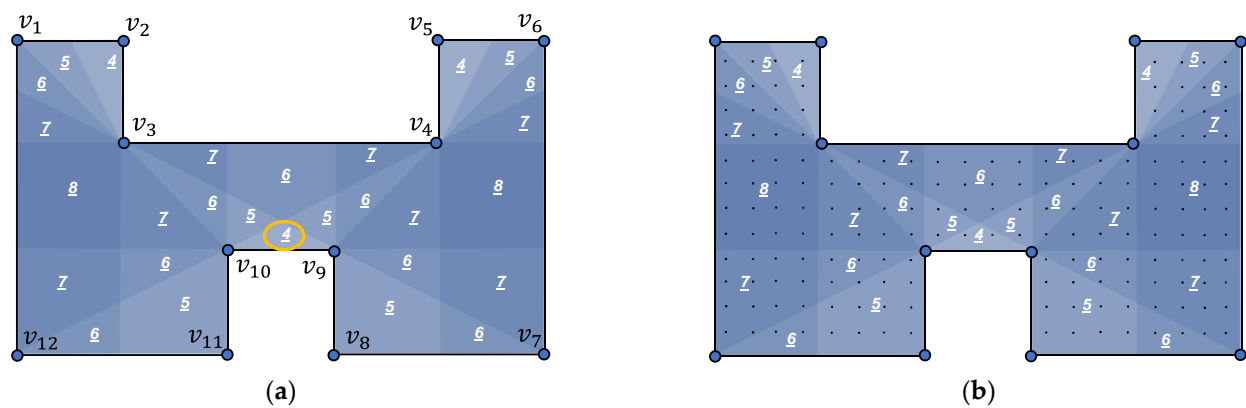


**Figure 2.** Visibility polygons. (a) For an arbitrary viewpoint; (b) for a polygon vertex.

In this research, the construction of visibility polygons is only executed on the site polygon vertices. This means that for a site polygon with  $n$  vertices, only a total of  $n$  visibility polygons will be generated. The visibility polygon for  $v_3$  in the example is depicted in Figure 2b. The procedure to obtain the visibility polygon is quite straightforward. Since the vertices are known inputs, the visibilities between them can be quickly determined. In our program, they are represented as an  $n \times n$  matrix whose elements indicate the visibilities between the corresponding vertices. Pseudo-vertices are the points of intersection between the extended sightline between two vertices and any site polygon edges, which are also searched during the visibility check. In Figure 2b,  $v'_4$  is the pseudo-vertex from the sightline  $l_{v_3v_4}$ . The scenarios when pseudo-vertices fall on the real vertices, such as  $v_7$  and  $v'_9$ , are also checked, and any redundant vertex will be eliminated. For each vertex, all its visible vertices and pseudo-vertices will be sorted to accomplish the visibility polygon construction.

### 3.2. Discretized Visibility Heatmap

A vertex visibility polygon indicates the locations within the site polygon that can observe that vertex. Overlaying all the vertex visibility polygons provides a visibility heatmap, as shown in Figure 3a, which represents the visibility score of each location for the polygon vertices.



**Figure 3.** Visibility heatmaps. (a) Non-discretized heatmap; (b) discretized heatmap. Underlined numbers indicate the number of visible vertices.

In the heatmap in Figure 3a, the entire site polygon is divided into several segments by the vertex visibility polygons. The score marked on each piece represents how many vertices can be observed from the corresponding area. This is also indicated by the shading. For example, the circled score of 4 means that any point within the triangular area resulting from the polygon intersection can see four vertices ( $v_3$ ,  $v_4$ ,  $v_9$ , and  $v_{10}$ ). Once the polygon intersection has been performed, a viewpoint and its inherent visibility can be found.

As only one viewpoint is needed within each polygon segment, a discretized heatmap is constructed and a single location is chosen. This is executed by discretizing the polygon  $P$  into a finite set of viewpoint candidates (VPCs), as shown in Figure 3b. The candidate falls into any polygon segment will be assigned with the corresponding visibility score. The score indicates how many of the  $n$  visibility polygons a viewpoint candidate lies within. Hence, the score can be determined by implementing the *point in polygon* check.

### 3.3. Edge Visibility Matrix

A  $k \times n$  vertex visibility matrix, as shown in Table 1, can then be constructed.  $k$  is the number of discretized viewpoint candidates and  $n$  is the number of polygon vertices. The visible vertices are marked as 1 while the invisible ones are scored 0, following the result of the *point in polygon* check.

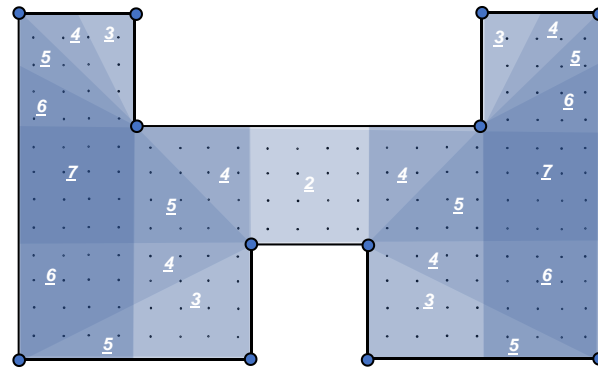
**Table 1.** Vertex visibility matrix.

	$v_1$	$v_2$	...	$v_n$
$VPC_1$	1	1	...	0
$VPC_2$	1	1	...	1
...	...	...	...	...
$VPC_k$	0	0	...	1

Since the goal of this research is to achieve the full coverage of the walls, i.e., the polygon edges, the vertex visibility matrix is further assessed to obtain the edge visibility matrix. For a polygon without holes, an edge is marked as visible if its two vertices are both visible from a VPC. Then, the edge visibility matrix in Table 2 can be constructed to indicate the visibility between VPCs and polygon edges from  $E_1$  to  $E_n$ . The corresponding discretized heatmap of edge visibility in the example polygon is shown in Figure 4. Scores marked on each piece represent the number of edges observed from the corresponding areas.

**Table 2.** Edge visibility matrix.

	$E_1$	$E_2$	$\dots$	$E_n$
$VPC_1$	1	0	$\dots$	0
$VPC_2$	1	1	$\dots$	1
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
$VPC_k$	0	0	$\dots$	0

**Figure 4.** Discretized heatmap for the edge visibility.

### 3.4. Optimization Method

The edge visibility matrix builds the foundation for terrestrial laser scanner viewpoint planning. Afterwards, a suitable optimization method must be selected to resolve this problem. The result of the optimization should be a set of minimum number of viewpoint candidates so that all the edges can be observed at least once.

If the inputs are described as:

- The polygon edges:  $\mathcal{E} = \{E_i | i = 1, \dots, n\}$
- The discretized viewpoint candidates:  $\mathcal{C} = \{VPC_j | j = 1, \dots, k\}$

Then the viewpoint planning problem can be represented by the linear programming formulation:

$$\min \sum_{j=1}^k VPC_j \quad VPC_j \in \{0, 1\} \quad (1)$$

subject to

$$\sum_{j=1, \dots, k} c_{ij} VPC_j \geq 1 \quad c_{ij} \in \{0, 1\}$$

The constraints in Equation (1) are the system of equations that comprise the following  $n$  equations representing the full coverage of the polygon edges. That is, each polygon edge is covered at least once among all the viewpoint candidates.

$$\begin{cases} c_{11}VPC_1 + c_{12}VPC_2 + \dots + c_{1k}VPC_k \geq 1 \\ c_{21}VPC_1 + c_{22}VPC_2 + \dots + c_{2n}VPC_k \geq 1 \\ \vdots \\ c_{n1}VPC_1 + c_{n2}VPC_2 + \dots + c_{nk}VPC_k \geq 1 \end{cases} \quad (2)$$

where

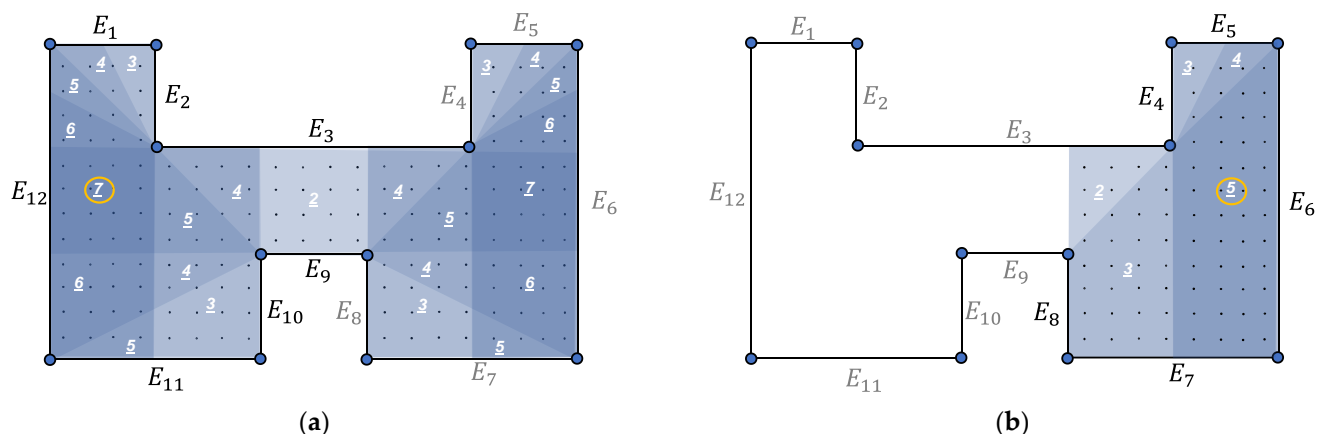
$$c_{ij} = \begin{cases} 0, & \text{if } E_i \text{ is not visible from } VPC_j \\ 1, & \text{if } E_i \text{ is visible from } VPC_j \end{cases}$$

$$VPC_j = \begin{cases} 0, & \text{if } VPC_j \text{ is not selected} \\ 1, & \text{if } VPC_j \text{ is selected} \end{cases}$$

The value of  $c_{ij}$  depends on the visibility between each viewpoint candidate and each edge, which can be found in the edge visibility matrix. The value of  $VPC_j$  determines whether a viewpoint candidate belongs to the solution set  $S$ . If  $VPC_j$  equals 1, the  $j$ th candidate is selected in the optimal solution; otherwise,  $VPC_j$  is not a necessary scan location. The solution set for the viewpoint planning problem is  $S = \{VPC_j | VPC_j = 1\}$ , which contains the minimum number of  $VPC_j$  that satisfies the constraints in Equation (2). In the TLS viewpoint planning problem, the values in the optimization are all binary; that is,  $VPC_j \in \{0, 1\}$ . Thus, this problem belongs to the special case of 0–1 integer linear programming, which is an NP-complete problem [18]. For any of the known methods, the computational complexity in searching the optimal solution increases rapidly with the size of the problem, which in this case is the number of vertices.

In this research, the optimization methods of original greedy algorithm (OGA) and the weighted greedy algorithm (WGA) were applied to solve Equation (1). The idea of OGA is to “greedily” pick the current optimal solution and then converge to the optimal/near optimal solution in a few computation steps. The WGA takes a step forward by reassigning the visibilities in the matrix based on a weighting scheme that prioritizes VPs that can see walls with lower coverage. The detailed description of these methods can be found in the authors’ previous publications [23,26].

In this example, the first optimal viewpoint is selected within the circled area in Figure 5a, which covers seven out of twelve edges ( $E_1$  to  $E_3$  and  $E_9$  to  $E_{12}$ ). After that, the heatmap is updated with the visibility polygons of the unobserved edges  $E_4$  to  $E_8$ , and the second optimal viewpoint is selected as shown in Figure 5b with the same strategy. In total, the solution of this viewpoint planning problem includes two optimal viewpoints.

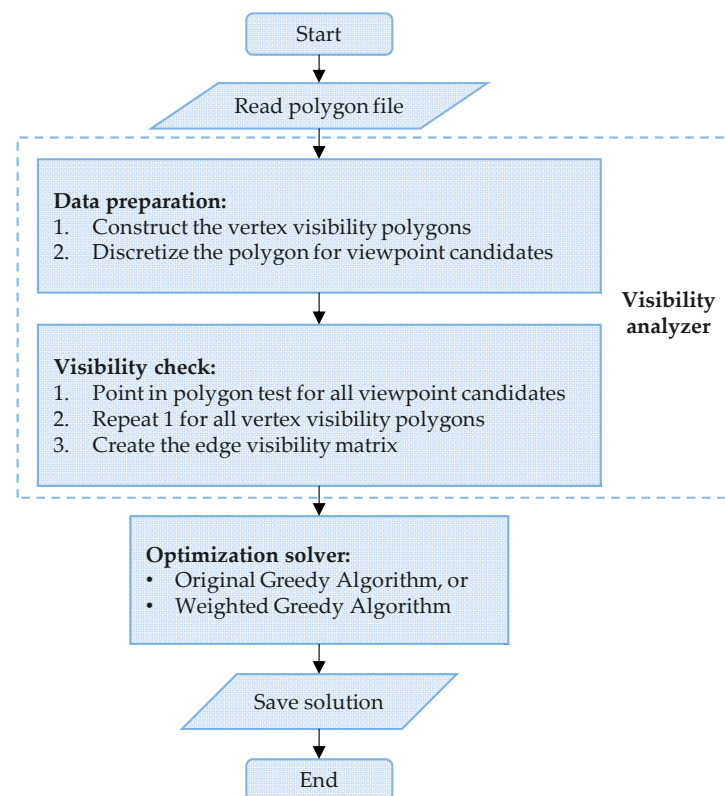


**Figure 5.** Find the optimal viewpoints. (a) First viewpoint selection based on the initial heatmap; (b) updated heatmap after selecting a viewpoint.

### 3.5. Work Flow

The flowchart in Figure 6 summarizes the proposed method. The four steps mentioned above are performed for all the instances tested in next section.





**Figure 6.** Flowchart of the proposed viewpoint planning method.

#### 4. Experiments and Discussion

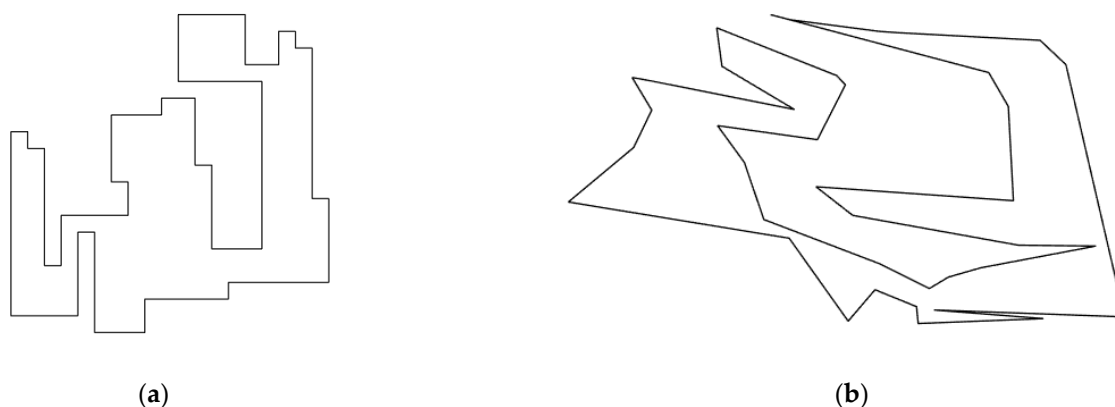
The experimental results are discussed in this section. The viewpoint planning results from the proposed method were compared with a benchmark technique. The comparisons were performed over 540 instances in terms of the number of optimal viewpoints and the computational efficiency. Our methods were coded in C++ in the Microsoft Visual Studio 2019 environment, and the tests were conducted on a computer featuring an Intel® Core™ i5-7500 at 3.40 GHz and 24 GB of RAM.

##### 4.1. Experimental Data

Instead of evaluating a particular laser scanning scenario, tests were performed on a large group of datasets so that different techniques can be compared regarding their versatility and robustness. A total of 540 polygon simulations, also called instances, were used in this test, varying in type, size, and complexity. The polygons were retrieved from [33], where more than 2700 polygons are publicly available for evaluating the Art Gallery Problem techniques. The benchmark solutions are also presented for comparison.

Only the hole-free polygons are evaluated in this research, and they are classified into random orthogonal and random simple polygons. All edges of a random orthogonal polygon are parallel to Cartesian coordinate axes, while the random simple polygon does not meet this condition. The generation of these polygons is described in [15]. In each group, 270 polygons are selected, with sizes of 20, 40, 60, 80, 100, 200, 300, 400, and 500 vertices. Each group also contains 30 different instances of each polygon size. Two sample polygons with 40 vertices are shown in Figure 7.





**Figure 7.** Sample polygons with 40 vertices. (a) Random orthogonal polygons; (b) random simple polygons.

Three methods summarized in Table 3 have been compared in this research: the benchmark technique proposed by Couto et al. in [16], and our methods based on two different optimization strategies. A significant reason for the selection of the benchmark paper is that it has the dataset and results of thousands of polygons available for download. The data includes different types of polygons with and without holes, which establishes a solid foundation for this research and its further improvements. By testing our methods with massive random scenarios, their validities can be proved, and eventually, they can be applied on the viewpoint planning of any TLS scanning project.

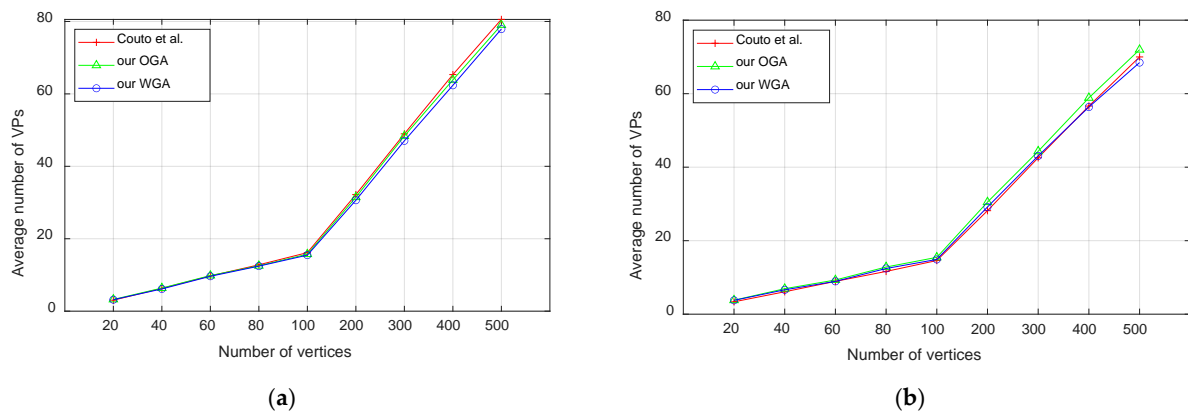
**Table 3.** Methods for comparison.

Name	Description
Couto et al. [16,33]	The benchmark paper where the simulated data were created and tested on.
Our OGA	Our proposed method using the original greedy algorithm for optimization.
Our WGA	Our proposed method using the weighted greedy algorithm for optimization.

#### 4.2. Optimality

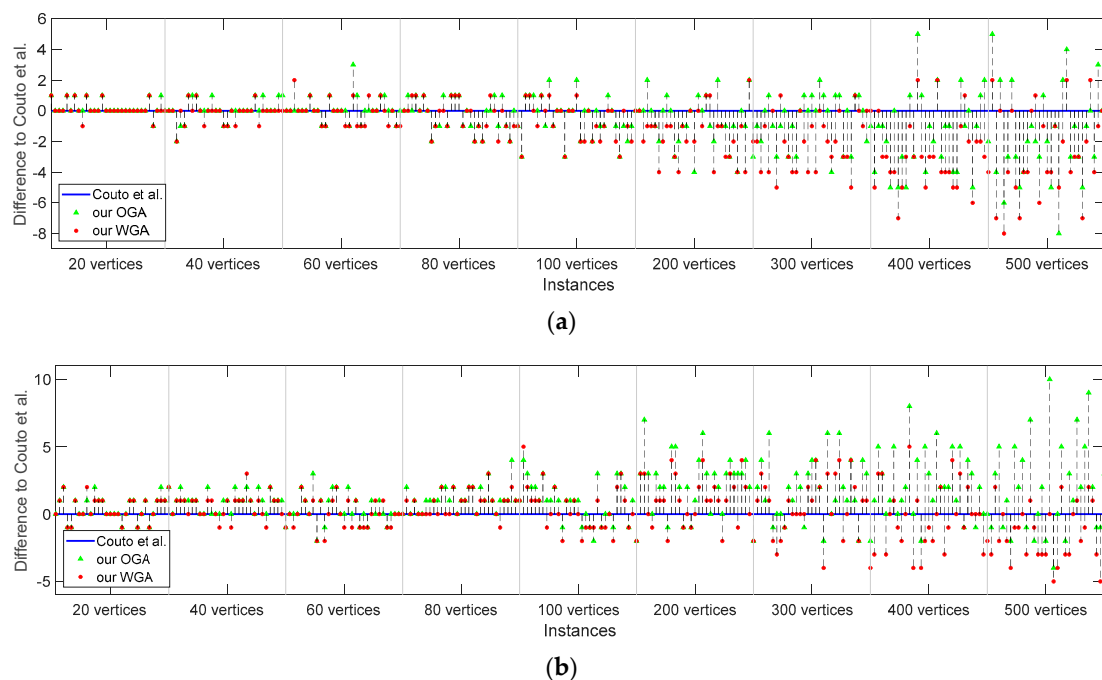
The optimality of a designed TLS network can be assessed with several criteria such as the achieved coverage and the number of viewpoints. Full coverage of the polygon edges was achieved by both of our methods and the benchmark technique. In this subsection, the optimality in terms of the number of optimal viewpoints is discussed.

Figure 8 shows the average number of selected viewpoints for each group of polygon instances. Each data point was obtained by averaging the results of the 30 instances at each number of vertices. It is obvious that the number of the required viewpoints increases with the size of the polygons, ranging from less than 5 up to 80 viewpoints. The difference between all three methods also becomes larger when the polygon size grows. However, the performance of the methods is not coherent over all the instances, which means that our methods are not always the best. This situation is addressed more specifically in the subsequent discussion. By comparing the results from two types of polygons, it seems that overall, the random simple polygons require fewer viewpoints than the orthogonal ones. It is hard to probe what really causes this difference because the polygons are generated with different strategies. For example, if the simple polygons generally have fewer concave vertices, they might need fewer viewpoints.



**Figure 8.** Average number of optimal viewpoints for (a) Random orthogonal polygons; (b) Random simple polygons.

To further investigate the performance of the methods, results from different sources are compared instance by instance. Figure 9 illustrates the results for all 540 instances, 270 for each polygon type. The number of viewpoints found by Couto et al. was subtracted from our results. Figure 9 thus shows the difference in the number of viewpoints between our methods and Couto et al. The blue line indicates the results from the benchmark paper, which are constant (zero). Values above zero indicate more viewpoints and, hence, less optimal solutions than Couto et al. Markers on or below zeros represent equivalent or better solutions. The x-axes are divided into nine sections, each containing 30 instances corresponding to nine polygon sizes.



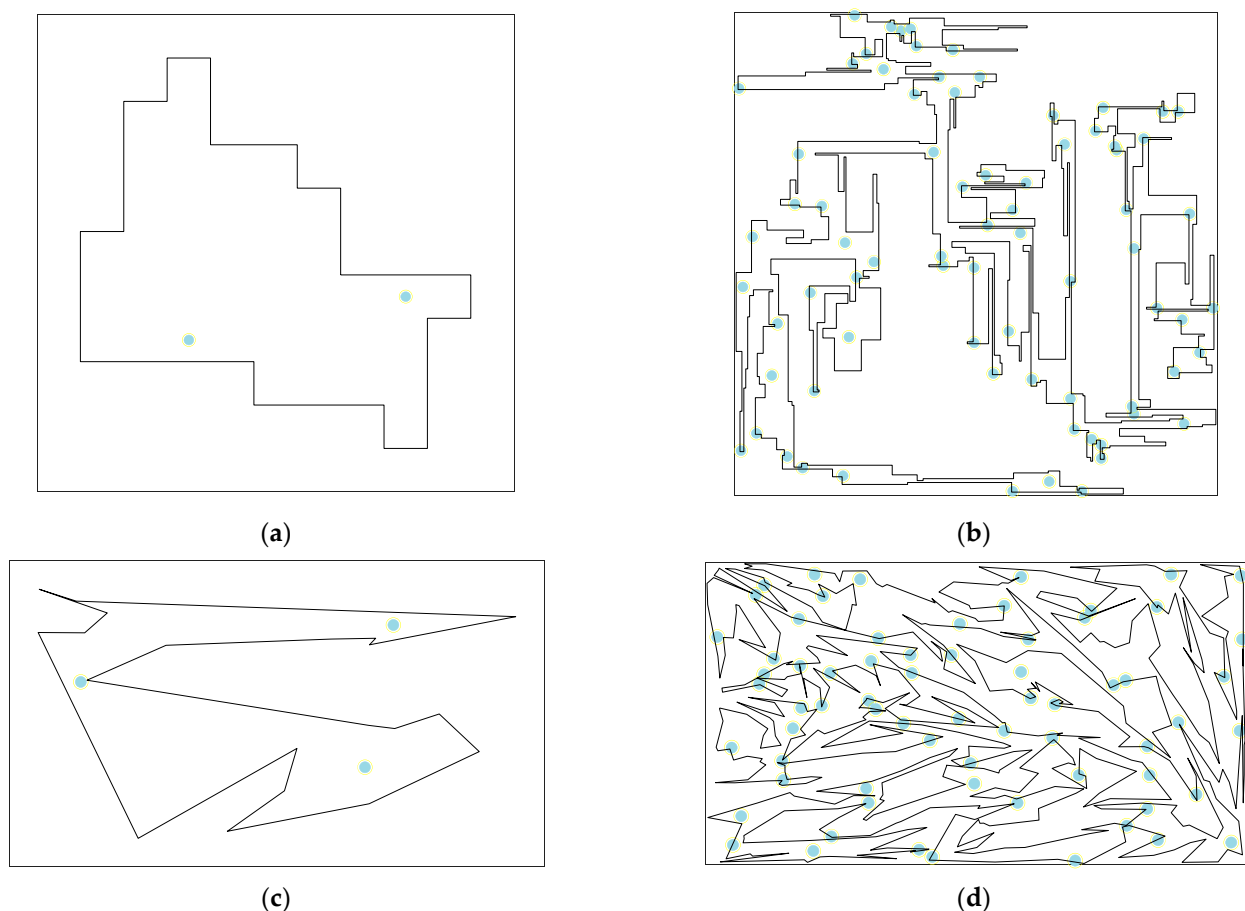
**Figure 9.** Difference in the number of optimal viewpoints between Couto et al. and our methods for (a) random orthogonal polygons; (b) random simple polygons.

Figure 9 shows that there are differences between our methods and the benchmark paper. In some instances, our performance is superior, but in others, it is not. For small numbers of vertices, our solutions agree closely with those of Couto et al., i.e., generally within one viewpoint. As the number of vertices increases, the differences in the number of VPs increase. This suggests that all three methods tend to fall into a local optimal solution when the polygon becomes more complicated. Thus, it may be more critical to find the

best method in this scenario. In addition, performance differs for two types of polygons. Our methods, particularly the WGA, outperform Couto et al. for the random orthogonal polygons. For the random simple polygon cases, the OGA is inferior and, overall, the performance of the WGA is neutral across all instance sections.

These outcomes indicate that our methods perform better in environments comprising orthogonal polygons. This is quite an acceptable outcome since our research focus is terrestrial scanning of complex buildings that are generally designed with orthogonal walls. We can also compare the results from our proposed OGA and WGA methods, which are represented by green triangles and red dots in Figure 9. WGA beats the OGA method in almost all instances by finding an equivalent or lower number of viewpoints. The numerical comparison is provided later in Section 4.4.

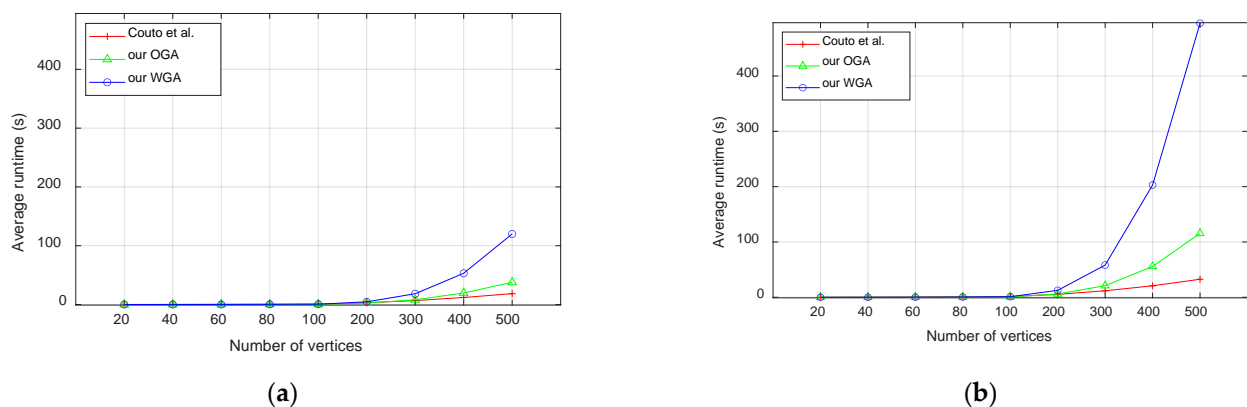
Figure 10 plots four environments where our WGA method yields fewer viewpoints. As the actual viewpoint locations of the benchmark paper are not provided by the authors, only our results are presented here. For the less complicated scenarios in Figure 10a,c, it is quite clear that they can be fully covered by our solution, which is two VPs for the orthogonal environment and three VPs for the simple environment. They are both one viewpoint less than Couto et al., meaning that the benchmark solutions contain redundant viewpoints. Improvements found in our results shown in Figure 10b,d are more significant, reducing the number of viewpoints by 8 and 5 out of 83 and 73 VPs found by Couto et al. Results in the 500 vertices cases have also been visually checked for the guarantee of full coverage.



**Figure 10.** The WGA viewpoint planning results for the sample polygons. (a) Random orthogonal polygons with 20 vertices (instance #20-9); (b) random orthogonal polygons with 500 vertices (instance #500-4); (c) random simple polygons with 20 vertices (instance #20-4); (d) random simple polygons with 500 vertices (instance #500-29).

#### 4.3. Computational Efficiency

Figure 11 compares the runtime in achieving the optimality between different methods. Like in Figure 8, the data were averaged for each size of the polygon instances. While the runtime increases with the number of vertices, the difference remains negligible until 300 vertices instances. For the large-sized polygons, the runtime of our methods increases considerably, especially for the WGA method. It is hard to directly compare the run time, as the computation environment could be rather different. The linear programming solvers used in the benchmark method might also accelerate the data processing. In our methods, most of the time is spent on the iterative *point in polygon* check and the visibility matrix construction, both of which could be optimized through advanced programming techniques. The causes of the slow execution of the WGA are the extra calculations in weighting the visibility matrix. Comparing the runtime between two types of polygons, we can tell that by using our methods, the random simple polygons take roughly up to three times longer than the orthogonal ones. This could be caused by the arbitrariness of the polygon edge direction that increases the complexity of the visibility check. This is supported by the results from Couto et al., whose time for the simple polygons also doubles the orthogonal ones. In general, our methods can obtain the solution within 2 min to 8 min for polygons with up to 500 vertices and various complexities. This is quite a reasonable time to complete a TLS network design.



**Figure 11.** Average runtime in reaching the optimality for (a) random orthogonal polygons; (b) random simple polygons. (Direct comparison is not possible due to different computation environments.)

#### 4.4. Summary of Results

Some of the results corresponding to Figures 8 and 11 are summarized in Table 4. Statistically, the proposed WGA method performs better, as it requires fewer viewpoints than the other methods, but the tradeoff is the run time. A normal real TLS scanning project usually has no more than 60 vertices. In Table 4, one can find that in these smaller vertices scenarios, our WGA not only provides a compatible average number of viewpoints, but also has faster computation.

Figure 12 displays the optimality rate of the proposed methods in two types of polygons. The optimality rate at each data point is calculated by counting how many results among 30 instances are equal to or better than the benchmark paper. For example, the rate of 80% for 20 vertices polygons means that the WGA method achieves equivalent or better results in 24 out of 30 instances. This figure proves again that our methods adapt better in the orthogonal environment than the simple polygons, as discussed before. The average optimality rate for nine groups of the orthogonal polygons using the WGA method is 85.6% compared to 75.6% for the OGA method. In the simple polygons, the numbers are 54.8% (WGA) and 36.3% (OGA), respectively. In scenarios with no more than 60 vertices, the optimality rate of the WGA method is higher than 80%.

**Table 4.** Summary of the numerical results.

Polygon Type	#Vertices	Average #VPs			Average Runtime (s)		
		Couto et al.	OGA	WGA	Couto et al.	OGA	WGA
Orthogonal	20	3.03 ± 0.93	3.23 ± 1.38	3.17 ± 1.42	0.04 ± 0.00	0.01 ± 0.00	0.01 ± 0.00
	60	9.73 ± 1.58	9.83 ± 1.94	9.63 ± 1.90	0.29 ± 0.05	0.10 ± 0.03	0.12 ± 0.04
	200	32.20 ± 4.49	31.47 ± 5.51	30.63 ± 5.84	3.00 ± 0.36	2.48 ± 0.69	4.56 ± 1.41
	500	80.57 ± 5.31	79.03 ± 7.20	77.90 ± 7.22	18.44 ± 1.39	37.74 ± 8.85	120.03 ± 33.79
Simple	20	3.37 ± 1.17	3.80 ± 1.53	3.77 ± 1.47	0.07 ± 0.02	0.01 ± 0.00	0.02 ± 0.02
	60	8.93 ± 1.87	9.30 ± 2.07	8.93 ± 1.99	0.52 ± 0.08	0.17 ± 0.05	0.22 ± 0.05
	200	28.17 ± 3.71	30.50 ± 4.68	29.17 ± 4.66	5.26 ± 0.41	5.96 ± 0.68	12.51 ± 0.71
	500	70.00 ± 5.96	71.97 ± 6.21	68.47 ± 5.87	32.57 ± 2.50	115.45 ± 10.06	495.38 ± 55.10

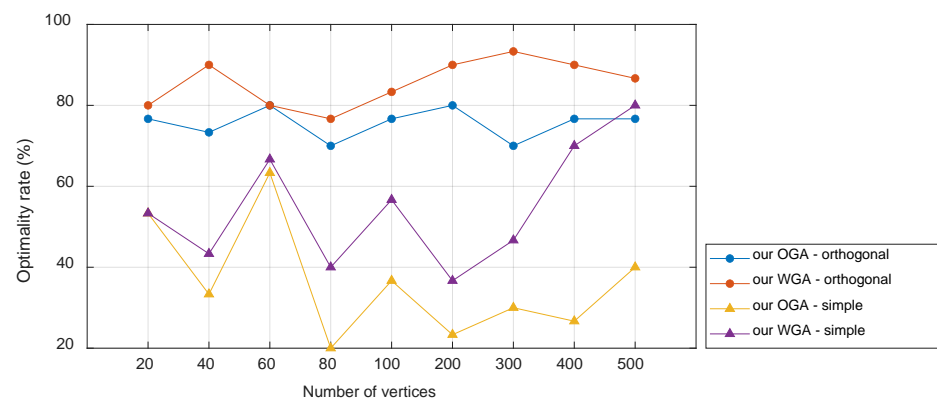
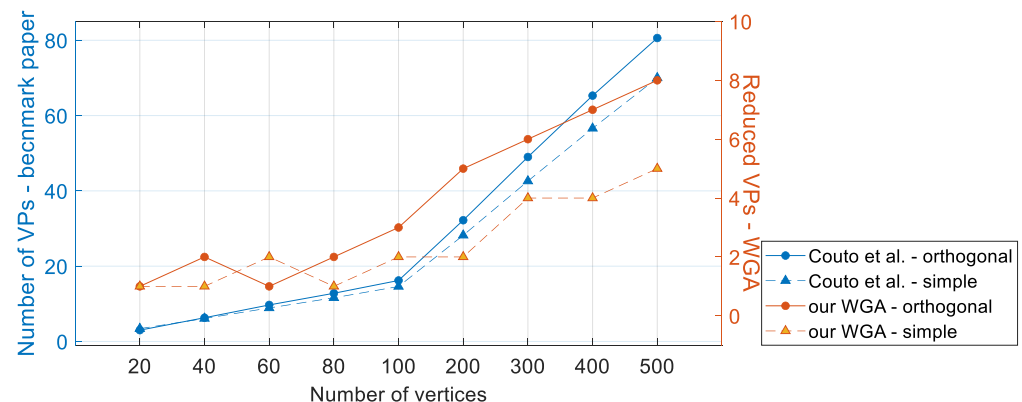
**Figure 12.** Optimality rate of the proposed methods to the benchmark paper method.

Figure 13 exhibits a plot with two groups of results. The left y-axis indicates the average number of optimal viewpoints saved by Couto et al., represented by the blue markers. The right y-axis displays the maximum number of viewpoints reduced by the WGA method among 30 instances, which indicates the scenario when our WGA outperforms the benchmark paper the most. For example, according to Figure 9a, the WGA method reduces as many as eight viewpoints for the orthogonal polygons with 500 vertices; thus, the corresponding data point in Figure 13 is eight. Combining these two sets of numbers gives the reduced viewpoints in percentages, which essentially is the lowered workload in the field. This can also be used to assess the cost of a scanning project. For example, to scan an orthogonal environment with 500 vertices, our WGA method requires eight scans less than the 80 scans from the benchmark paper, reducing the project cost by 10%. Since the time and cost saved from the decreased number of VPs in the field is far more than a few minutes in the office, it justifies that the relatively lower efficiency of our method is acceptable.

**Figure 13.** Reduced viewpoints of our WGA method to the benchmark paper method.

## 5. Conclusions

In this paper, a viewpoint planning strategy for complex scenes to be recorded using terrestrial laser scanners is presented. It is realized by the visibility analyzer and the optimization solver. Firstly, the scene to be scanned is outlined by a 2D polygon. Then, the site polygon is assessed by the visibility analyzer through the construction of a vertex visibility polygon, the discretization of a visibility heat map, and the construction of an edge visibility matrix. The results of the visibility analysis are fed into the optimization solver. Integrated with the OGA and our previously proposed WGA, the solver provides the locations of the viewpoints required to fully cover the scene of interest.

A total of 540 simulated polygons were used as the validation data. Categorized into the random orthogonal and the random simple polygons, the data were simulated with vertices ranging from 20 up to 500. Each polygon was processed with our method with two optimization strategies and compared with the results from the benchmark paper. It showed that on average, our method performed no worse than the benchmark paper in terms of the number of viewpoints in up to 85.6% of the instances. For a specific instance, the WGA-based method reduced eight viewpoints for a scene that required 80 viewpoints. Overall, our method was found to be superior in random orthogonal instances, so it is better suited for planning real-world scenes. Regarding efficiency, the proposed method requires longer processing time than the benchmark paper. However, with a computer configured like ours, the method can provide the solution in under eight minutes for applications with no more than 500 vertices, which suits most of the common scenes in the real world. With a more powerful configuration, this time can be further reduced. Thus, the method can already be used for on-site planning.

Along with the promising results, additional problems for future work have been revealed. Further investigations in the following aspects are suggested. The algorithm should be first extended to the polygons with holes, which improves the method by having the obstacles considered. Better polygon construction and processing strategy are required for our software, which can also reduce the time wasted on the polygon analyzer. The ILP scheme used in [16] should also be tested for exploring exact solutions. The location of the viewpoint should be provided by incorporating the geometry constraints and the overlap rate. In doing so, the method can provide solutions with better point cloud quality and meet the requirement of cloud-by-cloud registration. With these problems resolved, an advanced method will be available for a fast and reliable viewpoint planning of terrestrial laser scanning projects.

**Author Contributions:** Conceptualization, F.J.; methodology, F.J.; software, F.J.; validation, F.J.; formal analysis, F.J.; resources, F.J. and D.D.L.; writing—original draft preparation, F.J.; writing—review and editing, D.D.L. and F.J.; supervision, D.D.L.; funding acquisition, D.D.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** Funding for this research was provided by the Natural Sciences and Engineering Research Council of Canada (NSERC; grant number: RGPIN/03775-2018).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data will be used exclusively for this research.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Rashidi, M.; Mohammadi, M.; Sadeghlou Kivi, S.; Abdolvand, M.M.; Truong-Hong, L.; Samali, B. A decade of modern bridge monitoring using terrestrial laser scanning: Review and future directions. *Remote Sens.* **2020**, *12*, 3796. [\[CrossRef\]](#)
2. Truong-Hong, L.; Lindenbergh, R. Automatically extracting surfaces of reinforced concrete bridges from terrestrial laser scanning point clouds. *Autom. Constr.* **2022**, *135*, 104127. [\[CrossRef\]](#)
3. Ulvi, A. Documentation, Three-Dimensional (3D) Modelling and visualization of cultural heritage by using Unmanned Aerial Vehicle (UAV) photogrammetry and terrestrial laser scanners. *Int. J. Remote Sens.* **2021**, *42*, 1994–2021. [\[CrossRef\]](#)



4. Jo, Y.H.; Hong, S. Three-dimensional digital documentation of cultural heritage site based on the convergence of terrestrial laser scanning and unmanned aerial vehicle photogrammetry. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 53. [\[CrossRef\]](#)
5. Che, E.; Olsen, M.J.; Jung, J. Efficient segment-based ground filtering and adaptive road detection from mobile light detection and ranging (LiDAR) data. *Int. J. Remote Sens.* **2021**, *42*, 3633–3659. [\[CrossRef\]](#)
6. Pierzchała, M.; Giguère, P.; Astrup, R. Mapping forests using an unmanned ground vehicle with 3D LiDAR and graph-SLAM. *Comput. Electron. Agric.* **2018**, *145*, 217–225. [\[CrossRef\]](#)
7. Aghababaei, M.; Okamoto, C.; Koliou, M.; Nagae, T.; Pantelides, C.P.; Ryan, K.L.; Barbosa, A.R.; Pei, S.; van de Lindt, J.W.; Dashti, S. Full-scale shake table test damage data collection using terrestrial laser-scanning techniques. *J. Struct. Eng.* **2021**, *147*, 04020356. [\[CrossRef\]](#)
8. Wilson, L.; Rawlinson, A.; Frost, A.; Hephher, J. 3D digital documentation for disaster management in historic buildings: Applications following fire damage at the Mackintosh building, The Glasgow School of Art. *J. Cult. Herit.* **2018**, *31*, 24–32. [\[CrossRef\]](#)
9. Kuang, S.L. *Geodetic Network Analysis and Optimal Design: Concepts and Applications*; Ann Arbor Press: Chelsea, MI, USA, 1996.
10. Tozoni, D.C.; de Rezende, P.J.; de Souza, C.C. A practical iterative algorithm for the art gallery problem using integer linear programming. *ACM Trans. Math. Softw.* **2017**, *43*, 1–27. [\[CrossRef\]](#)
11. Couto, M.C.; De Souza, C.C. An exact and efficient algorithm for the orthogonal art gallery problem. In Proceedings of the XX Brazilian Symposium on Computer Graphics and Image Processing, Minas Gerais, Brazil, 7–10 October 2007.
12. Couto, M.C.; Souza, C.C.D.; Rezende, P.J.D. Experimental evaluation of an exact algorithm for the orthogonal art gallery problem. In Proceedings of the International Workshop on Experimental and Efficient Algorithms, Provincetown, MA, USA, 30 May–1 June 2008.
13. Couto, M.; Souza, C.; Rezende, P. Strategies for optimal placement of surveillance cameras in art galleries. In Proceedings of the 18th International Conference on Computer Graphics and Vision, Moscow, Russia, 23–28 June 2008.
14. Couto, M.C.; de Rezende, P.J.; de Souza, C.C. An IP solution to the art gallery problem. In Proceedings of the Twenty-Fifth Annual Symposium on Computational Geometry, Aarhus, Denmark, 8–10 June 2009.
15. Couto, M.C.; de Rezende, P.J.; de Souza, C.C. *An Exact Algorithm for an Art Gallery Problem*; Technical Report IC-09-46; Institute of Computing, The State University of Campinas: Campinas, Brazil, 2009.
16. Couto, M.C.; de Rezende, P.J.; de Souza, C.C. An exact algorithm for minimizing vertex guards on art galleries. *Int. Trans. Oper. Res.* **2011**, *18*, 425–448. [\[CrossRef\]](#)
17. Borrmann, D.; De Rezende, P.J.; De Souza, C.C.; Fekete, S.P.; Friedrichs, S.; Kröller, A.; Nüchter, A.; Schmidt, C.; Tozoni, D.C. Point guards and point clouds: Solving general art gallery problems. In Proceedings of the Twenty-Ninth Annual Symposium on Computational Geometry, Rio de Janeiro, Brazil, 17–20 June 2013.
18. Tozoni, D.C.; Rezende, P.J.D.; Souza, C.C.D. The quest for optimal solutions for the art gallery problem: A practical iterative algorithm. In Proceedings of the International Symposium on Experimental Algorithms, Rome, Italy, 5–7 June 2013.
19. Soudarissanane, S.; Lindenbergh, R.; Menenti, M.; Teunissen, P. Scanning geometry: Influencing factor on the quality of terrestrial laser scanning points. *ISPRS J. Photogramm. Remote Sens.* **2011**, *66*, 389–399. [\[CrossRef\]](#)
20. Soudarissanane, S.; Lindenbergh, R. Optimizing terrestrial laser scanning measurement set-up. In Proceedings of the ISPRS Workshop Laser Scanning 2011, Calgary, AB, Canada, 29–31 August 2011.
21. Ahn, J.; Wohn, K. Interactive scan planning for heritage recording. *Multimed. Tools Appl.* **2015**, *75*, 3655–3675. [\[CrossRef\]](#)
22. Jia, F.; Lichti, D.D. A comparison of Simulated Annealing, Genetic Algorithm and Particle Swarm Optimization in optimal First-Order Design of indoor TLS networks. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *4*, 75–82. [\[CrossRef\]](#)
23. Jia, F.; Lichti, D.D. An efficient, hierarchical viewpoint planning strategy for terrestrial laser scanner networks. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2018**, *4*, 137–144. [\[CrossRef\]](#)
24. Revuelta, E.C.; Chávez, M.J.; Vera, J.A.B.; Rodríguez, Y.F.; Sánchez, M.C. Optimization of laser scanner positioning networks for architectural surveys through the design of genetic algorithms. *Measurement* **2021**, *174*, 108898. [\[CrossRef\]](#)
25. Mozaffar, M.; Varshosaz, M. Optimal placement of a terrestrial laser scanner with an emphasis on reducing occlusions. *Photogramm. Rec.* **2016**, *31*, 374–393. [\[CrossRef\]](#)
26. Jia, F.; Lichti, D.D. A model-based design system for terrestrial laser scanning networks in complex sites. *Remote Sens.* **2019**, *11*, 1749. [\[CrossRef\]](#)
27. Dehbi, Y.; Leonhardt, J.; Oehrlin, J.; Haunert, J.H. Optimal scan planning with enforced network connectivity for the acquisition of three-dimensional indoor models. *ISPRS J. Photogramm. Remote Sens.* **2021**, *180*, 103–116. [\[CrossRef\]](#)
28. Wujanz, D.; Neitzel, F. Model based viewpoint planning for terrestrial laser scanning from an economic perspective. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *41*, 607–614. [\[CrossRef\]](#)
29. Blaer, P.; Allen, P. Data acquisition and viewpoint planning for 3-D modeling tasks. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA, 29 October–2 November 2007.
30. Kawashima, K.; Yamanishi, S.; Kanai, S.; Date, H. Finding the next-best scanner position for as-built modelling of piping systems. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2014**, *40*, 313. [\[CrossRef\]](#)
31. Wakisaka, E.; Kanai, S.; Date, H. Model-based next-best-view planning of terrestrial laser scanner for HVAC facility renovation. *Comput.-Aided Des. Appl.* **2018**, *15*, 353–366. [\[CrossRef\]](#)



- 
32. Aryan, A.; Bosché, F.; Tang, P. Planning for terrestrial laser scanning in construction: A review. *Autom. Constr.* **2021**, *125*, 103551. [[CrossRef](#)]
  33. State University of Campinas Institute of Computing Art Gallery Problem Project. Available online: <https://www.ic.unicamp.br/~cid/Problem-instances/Art-Gallery/index.html> (accessed on 9 March 2022).