sensors

MDPI

Article

Automatic Registration of TLS-TLS and TLS-MLS Point Clouds Using a Genetic Algorithm

Li Yan, Junxiang Tan *, Hua Liu, Hong Xie and Changjun Chen *

School of Geodesy and Geomatics, Wuhan University, Luoyu Road 129, Wuhan 430079, China;
lyan@sgg.whu.edu.cn (L.Y.); liuhua@whu.edu.cn (H.L.); hxie@sgg.whu.edu.cn (H.X.)
* Correspondence: tanjunxiang@whu.edu.cn (J.T.); chencj@whu.edu.cn (C.C.);
Tel.: +86-27-6877-8166 (J.T.); +86-27-6875-8449 (C.C.)

Received: 4 July 2017; Accepted: 25 August 2017; Published: 29 August 2017

Abstract: Registration of point clouds is a fundamental issue in Light Detection and Ranging (LiDAR) remote sensing because point clouds scanned from multiple scan stations or by different platforms need to be transformed to a uniform coordinate reference frame. This paper proposes an efficient registration method based on genetic algorithm (GA) for automatic alignment of two terrestrial LiDAR scanning (TLS) point clouds (TLS-TLS point clouds) and alignment between TLS and mobile LiDAR scanning (MLS) point clouds (TLS-MLS point clouds). The scanning station position acquired by the TLS built-in GPS and the quasi-horizontal orientation of the LiDAR sensor in data acquisition are used as constraints to narrow the search space in GA. A new fitness function to evaluate the solutions for GA, named as Normalized Sum of Matching Scores, is proposed for accurate registration. Our method is divided into five steps: selection of matching points, initialization of population, transformation of matching points, calculation of fitness values, and genetic operation. The method is verified using a TLS-TLS data set and a TLS-MLS data set. The experimental results indicate that the RMSE of registration of TLS-TLS point clouds is 3~5 mm, and that of TLS-MLS point clouds is 2~4 cm. The registration integrating the existing well-known ICP with GA is further proposed to accelerate the optimization and its optimizing time decreases by about 50%.

Keywords: terrestrial LiDAR scanning; mobile LiDAR scanning; point cloud; registration; genetic algorithm

1. Introduction

Light detection and ranging (LiDAR) remote sensing technology has rapidly developed since it can collect 3D point clouds of object surfaces efficiently [1]. Terrestrial LiDAR scanning (TLS) with a LiDAR sensor mounted on a fixed platform [2] is widely used in various fields such as reverse engineering [3], cultural heritage documentation [4,5] and environmental monitoring [6–8]. Mobile LiDAR scanning (MLS) by integrating with several LiDAR sensors, a high-accuracy positioning and orientation system and a high-precision controlling system on a van or car provides a safer and more efficient way to capture large-scale geo-referenced point clouds [9,10]. It is being used at an increasing rate in the transportation industry [11,12], especially for road asset inventory [13,14] and in the production of high accuracy driving maps for intelligence driving [15].

The registration of point clouds is a fundamental issue in LiDAR remote sensing because point clouds are scanned from multiple scan stations or by different platforms, and they should be merged to obtain full coverage of a scene [16]. The aim of registration of different point clouds is to transform point clouds in different coordinate frames to a uniform coordinate reference frame. This paper deals with the registration problem of two TLS point clouds and the registration problem between TLS and MLS point clouds.

The traditional registration method is based on artificial markers that are placed in the scene during data acquisition. The positions of the markers must be manually extracted as tie points for registration. The marker-based registration is very reliable, but it requires very careful arrangements and is time-consuming [17]. Many automatic registration algorithms are proposed to improve the efficiency of the registration. The published methods can be classified into two categories: auxiliary data-based methods and 3D point-based methods [18]. Auxiliary data-based methods usually incorporate photographic images and point clouds for registration [19,20], or incorporate intensity images and point clouds for registration [21,22]. The image-assisted or intensity-assisted registration may bring up extra calibration of cameras and scanners, and the quality of the third-party data also has a direct effect on the registration process.

Many experts have directly used 3D point-based methods to solve the registration problem. The 3D point-based methods are usually divided into a coarse registration step and a fine registration step [23]. The feature-based algorithm is a common way to achieve coarse registration, which establishes correspondences between two point clouds using extracted features. The used features, such as geometric curvature, main frame and point signature, are invariant with rotation and translation [24,25]. Johnson and Hebert proposed a local shape descriptor called 'spin image' to match or recognize an object [26]. This histogram-feature registration is strongly affected by the given parameters—bin size, image width, and support angle. Rusu, et al. proposed a local non-linear optimizer called 'sample consensus initial alignment' (SAC-IA) to accomplish the registration, which employed their own fast point feature histogram (FPFH) as the feature descriptor [27]. Tombari, et al. and Hänsch, et al. reviewed the existing feature descriptors and compared their performance, and the results indicate that the specific method to extract features has to be carefully chosen [28,29]. It is hard to establish robust correspondences by the extracted features because of the uneven point density, clutters and outliers, repeated objects, sheer size, and partial overlap of the point clouds [17,30]. The optimization of feature-based registration also needs exhausted computation [31].

A well-known fine registration method is the iterative closest points (ICP) algorithm [32], where each point in a one point cloud is paired with the closest point in the other point cloud to form correspondences, and then a point-to-point error metric (the mean of the squared distances of the correspondences) is minimized. The process is iterated until the error becomes smaller than a threshold or the maximum iteration is achieved. Chen and Medioni proposed a point-to-plane error metric ICP [33], which is more accurate than the point-to-point ICP [34]. Hereafter, some modified ICP methods are proposed and they are distinct from four aspects [35]: (1) selection of candidate points; (2) search strategy of nearest points to establish correspondences; (3) weighting relationship of correspondences or rejection of invalid correspondences; and (4) error metric and optimization. Many ICP algorithms have been achieved in the Point Cloud Library (PCL) [36]. However, ICP is easy to fall into local optimum, which is inherently determined by the employed local optimizer [37]. Hence, their performance critically relies on the initialization quality (the quality of coarse registration), and only local optimality is guaranteed.

Unlike the above coarse-fine registration, registration based on genetic algorithm (GA) uses a global search strategy which automatically finds the optimal solution in the search space [38]. GA is a heuristic optimizer that simulates the evolution of nature—selection (survival of the fittest), crossover, and mutation. Jacq and Roux presented a framework of 3D medical point cloud registration based on GA [39]. Brunnstrom and Stoddart proposed a GA registration method for free-form surface matching for the first time, which achieved finding correspondences rather than searching optimal solutions in search space [40]. This method is not applicable when there are too many matching points. Yamany, et al. presented a GA registration method whose fitness function was to maximize the reciprocal of the sum of squared errors (SSE) between correspondences [41]. Silva, et al. gave more details of GA registration and proposed a better objective function to minimize the mean squared errors (MSE) between correspondences with outlier rejection [42,43]. Hereafter, the scholars also presented some different GA registration methods and their fitness functions were also based on the MSE [37,44,45]. MSE is originated from local optimization and it may not be globally optimal. The fitness values are often needed to scale into maximum values by a negative exponential function. In addition, GA registration is complicated and time-consuming because of the global search strategy of GA and the complexity of scanned point clouds.

This paper proposes an efficient GA registration method for registering two TLS point clouds or two point clouds where one is scanned by TLS and the other is scanned by MLS. In order to make the GA registration workable and improve the registration efficiency, the selection of matching points is first applied to eliminate the far, redundant and noisy points and to select partial points representing the main features before GA evolution. Besides, the scanning station position acquired by the TLS built-in GPS and the quasi-horizontal orientation of LiDAR sensor in data acquisition are used as constraints to narrow the search space in GA. Furthermore, the calculation of fitness values—the most time-consuming step of GA registration—is parallel-computed. Instead of the MSE-based fitness function, a new and more accurate fitness function, named 'Normalized Sum of Matching Scores' (NSMS), is proposed to evaluate the solutions.

The remainder of the paper is structured as follows. The GA is firstly introduced in Section 2. The proposed GA registration is described in Section 3. Then the experimental results and the discussions are described in Section 4, where the registration integrating ICP with GA is presented. The conclusions are described in the last section.

2. Genetic Algorithm

Genetic algorithm (GA) is a global and heuristic optimizer which simulates biological evolution. It maintains a population of candidate solutions and evolves by iteratively applying three genetic operators: selection (survival of the fittest), crossover, and mutation. In this section, the GA is briefly introduced. More details can be referred in [38,46,47].

In order to apply GA, the encoding mode should be first determined. Encoding is to present a solution in the search space as a chromosome that is composed of genes and can be computed by the genetic operation. The inverse mode is called decoding. The common numerical encoding methods are binary encoding and float encoding. The binary encoding is to convert a real to a binary and the corresponding decoding is to inversely transform a binary to a real. The float encoding directly uses a parameter as an encoding gene, which the genes do not need in order to be decoded. Float encoding is therefore more efficient in the optimization of multivariable functions and there is no conversion accuracy loss.

A standard GA is divided into three steps—initialization of population, calculation of fitness values, and genetic operation. A population is a set of chromosomes $\{\boldsymbol{P} \mid \boldsymbol{Ch}_1, \boldsymbol{Ch}_2, \ldots\}$. In its initialization, a chromosome is often uniform-randomly generated in the search space. The pseudo code of initialization for float encoding is presented in Algorithm 1. The size of the population M is often empirically set to a few hundred. The search space is the solution domain of the optimization problem, which is defined between the negative and positive upper bound vector. Defining search space is a core issue in actual optimization.

Algorithm 1. The pseudo code of population initialization.

```
Input the upper bound vector of the solution
domain $U$ and population size $M$;
For $I$ = 1:$M$
    For $k$ = 1:DIMENSION($U$)
        Randomly generate $r$ in $[-U_k, U_k]$;
        $Ch_{i,k} = r$;
    End
End
```

The calculation of fitness values is based on a fitness function. The fitness function that is defined to evaluate the solutions is another core issue in actual optimization. It is scaled from the objective function of the optimization problem and provides the guidance for the GA selection operation. Therefore it directly affects the GA performance and should be carefully designed.

The genetic operation is a simulation of biological gene manipulation, including selection, crossover, and mutation. The selection operator is to select M chromosomes from the parent population for reproduction. The chance of selecting one chromosome as a father or mother should be proportional to the population size M. Remainder stochastic selection is a better method than the direct stochastic selection by fitness proportion. It can ensure the chromosomes with higher fitness proportions are chosen and have higher proportions in the selected population. Firstly, each chromosome in the population is copied several times for reproduction. The copied number of the ith chromosome is

$$Num_i = \left\lfloor M \cdot F_i / \sum_k F_k \right\rfloor \tag{1}$$

where, F_i is the fitness value of the ith chromosome. $\sum Num_i$ chromosomes are copied, and then the remained fitness value of the ith chromosome is

$$F_i' = F_i - \sum_k F_k \cdot Num_i / M \tag{2}$$

All remaining fitness values of the population are used to produce the other $M - \sum Num_i$ chromosomes by direct stochastic selection.

The crossover operator mates parents to produce two new offspring and the mutation operator alters one or more gene values in a chromosome from its initial state. To ensure that the genes of the optimal chromosome at each generation are not eliminated and damaged, the chromosome with the highest fitness is directly copied into the next generation. The crossover operator and mutation operator are dependent on the encoding. If the float encoding is applied, arithmetic crossover and non-uniform mutation are suitable for generating new chromosomes. Their pseudo codes are given in Algorithm 2. Here, two parameters, the crossover probability P_c and mutation probability P_m, should be set. Empirically, P_c is 0.6~1 and P_m is not more than 0.1.

Algorithm 2. The pseudo code of arithmetic crossover and non-uniform mutation.

arithmetic crossover

```
Input a mother chromosome $Ch_1$ and
a father chromosome $Ch_2$;
Randomly generate $p_c$ in [0, 1];
If $p_c < P_c$
  For $k$ = 1:DIMENSION($Ch_1$)
    Randomly generate $r$ in [0, 1];
    $T_k = r*(Ch_{2,k} - Ch_{1,k})$;
    $Ch_{1,k} = Ch_{1,k} + T_k$;
    $Ch_{2,k} = Ch_{1,k} - T_k$;
  End
End
```

non-uniform mutation

```
Input the upper bound vector of the solution
domain $U$ and a chromosome $Ch$;
Randomly generate $p_m$ in [0, 1];
If $p_m < P_m$
  $T = (1 - Current_g / MAX_g)^2$;
  For $k$ = 1:DIMENSION($Ch$)
    Randomly generate $r$ in [0, 1];
    If $r > 0.5$
      $Ch_k = Ch_k + (U_k - Ch_k)*r*T$;
    Else
      $Ch_k = Ch_k - (U_k + Ch_k)*r*T$;
    End
  End
End
```

The calculation of fitness values and genetic operation are iteratively applied, which constitutes the evolution of GA. The GA evolution is terminated when either of the following two criteria is satisfied:

- Maximum number of generations: exceeding the maximum generation MAX_g makes the evolution stopped.

- Maximum number of generations that the best fitness keeps stable: if the number of generations, which the current best fitness value is unchanged, is equal to a given value MAX_b, the evolution is also stopped. To find out the globally optimal solution, MAX_b should not be too small.

3. Proposed GA Registration

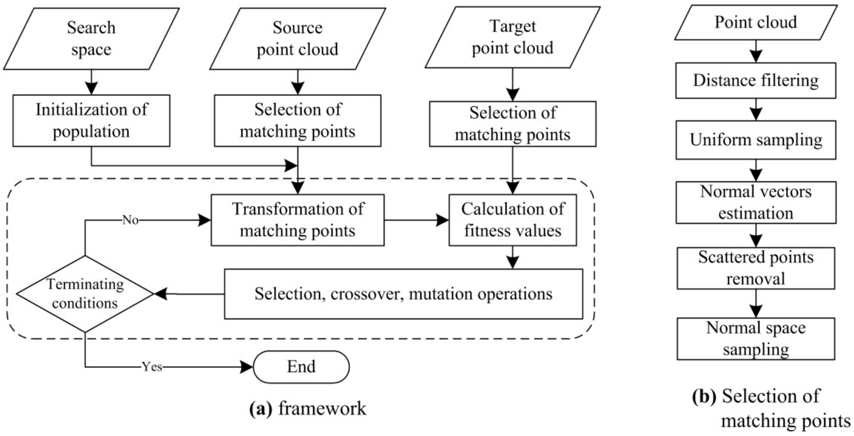
3.1. Framework

Given the source point cloud $\boldsymbol{S}$ with points $\boldsymbol{S}_i \in \boldsymbol{S}$ and the target point cloud $\boldsymbol{T}$ with points $\boldsymbol{T}_j \in \boldsymbol{T}$, the registration problem is to find correspondences $(\boldsymbol{S}_i, \boldsymbol{T}_j)$ between $\boldsymbol{S}$ and $\boldsymbol{T}$, and to estimate the rigid transformation [36]

$$\boldsymbol{T}_j = \boldsymbol{t} + \boldsymbol{R}\boldsymbol{S}_i \tag{3}$$

where, $\boldsymbol{t} = [t_x, t_y, t_z]^T$ is the unknown translation vector; $\boldsymbol{R}$ is the unknown rotation matrix that is expressed by a function of three Euler rotating angles α, β, γ around x, y, z-axes.

The pipeline of the proposed GA registration to estimate optimal transformation parameters is divided into five steps—selection of matching points, initialization of population, transformation of matching points, calculation of fitness values, and genetic operation. This framework is illustrated in Figure 1a. As illustrated in Figure 1b, the selection of matching points includes five sub-steps—distance filtering, uniform sampling, normal vectors estimation, scattered points removal, and normal space sampling. Its purpose is to make the GA registration efficient by eliminating the far, redundant, noisy points and by selecting some percentage of points that can express the main features. Its details are presented in Section 3.2.


Search space
Source point cloud
Target point cloud
Initialization of population
Selection of matching points
Selection of matching points
No
Transformation of matching points
Calculation of fitness values
Terminating conditions
Selection, crossover, mutation operations
Yes
End
Point cloud
Distance filtering
Uniform sampling
Normal vectors estimation
Scattered points removal
Normal space sampling

(a) framework

(b) Selection of matching points

Figure 1. The framework and selection of matching points of the proposed GA registration. The GA evolution is showed in the dashed box. (**a**) The flow chart of GA registration; (**b**) The process of selection of mathing points.

To further understand the proposed method, the pseudo code of the GA registration is given in Algorithm 3. The GA evolution consists of the steps from line 8 to line 27 in Algorithm 3 (the steps from 2 to 5 in the pipeline). Because the optimization contains multiple unknown parameters, the float encoding for GA is better compared with binary encoding. Each chromosome in the population is a six-dimensional vector $\boldsymbol{Ch}$ = $[\alpha, \beta, \gamma, x, y, z]$. Before GA evolution, the population must be initialized. The pseudo code of "Initialization of Population" is shown in Algorithm 1.

The "Transformation" using Equation Equation (3) is conducted for each chromosome, and then the closest points of the transformed $\boldsymbol{S}$ in $\boldsymbol{T}$ are searched and the chromosomes are evaluated by a fitness function in "Calculation of Fitness". According to the computed fitness values, the GA operations can be implemented to produce a new population. The GA evolution is stopped until the termination conditions are satisfied. The "Selection Operation" using the remainder of the stochastic selection has been described in Section 2. The float encoding is applied, so the arithmetic crossover and non-uniform mutation operation (Algorithm 2) are suitable for generating new chromosomes.

The "Calculation of Fitness" is the most time-consuming step of the GA registration because of the nearest neighbor searching between $\boldsymbol{S}$ and $\boldsymbol{T}$. The "Transformation" and "Calculation of Fitness" are independently conducted for each chromosome in the population. Therefore, these two steps can be accelerated by multi-thread parallel computing technology. In our program, the Open Multi-Processing (OpenMP) [48] is utilized to accelerate the computation.

Algorithm 3. The pseudo code of the Genetic Algorithm (GA) registration.

```
Input $S$ and $T$; Input the upper bound
vector $U$ ;Input the given parameters;
Selection Of Matching Points ($S$);
Selection Of Matching Points ($T$);
Initialization of Population ($P$);
Iter = 0; Iter_best = 0; cur_best = 0.0;
last_best = 0.0; //_best means best fitness
While (Iter < $Max_g$ and Iter_best < $Max_b$)
    Foreach $Ch_i$ in $P$
        Transformation ($S$);
        $F_i$ = Calculation Of Fitness ($S$,$T$);
    End
    cur_best = Selection Operation ($P$);
    Iter = Iter + 1;
    If cur_best = last_best
        Iter_best = Iter_best + 1;
    Else
        Iter_best = 0;
        last_best = cur_best;
    End
    Foreach $Ch_i$ and $Ch_{i+1}$ in $P$
        Crossover Operation ($Ch_i$, $Ch_{i+1}$);
    End
    Foreach $Ch_i$ in $P$
        Mutation Operation ($Ch_i$);
    End
End //While
```

To solve the GA registration, two core issues should be defined—(1) search space (solution domain); and (2) a fitness function to evaluate the solutions. The search space is expressed in Section 3.3, where the auxiliary constraints of the scanning station are implemented to narrow the search space. The proposed NSMS fitness function is presented in Section 3.4.

3.2. Selection of Matching Points

A large volume of noisy and unevenly distributed point clouds make the registration computation expensive. This section introduces some methods for selecting the candidate points for matching. The purpose is to make the registration efficient by eliminating the far, redundant, noisy points and then selecting some percentage of points that can still express the main features.

The points far away from the scanners are sparse and have significant noise. Hence, they are first eliminated by distance filtering. The max distance threshold D_{max} for distance filtering is set according to the specifications of the scanning system. The D_{max} should not be larger than the max range and can be often set to near the effective range of the scanner. e.g., under Riegl test conditions, the Riegl VZ400's max range with target reflectivity 10% is 120 m and its accuracy is 5 mm when the range is 100 m [49], so the D_{max} can be set to 100 m.

After distance filtering, uniform sampling is applied to avoid the near-field bias inherent in regular angular sampling and to achieve a reduction of the point count by voxel grid filtering [31]. The voxel grid filtering is that the points in a grid are replaced with the nearest point of the grid center. Then the normal vectors and curvatures are estimated by a local covariance matrix algorithm [50]. The voxel size V_g for uniform sampling can be set several times that of the point resolution. The vertical resolution can be computed by the vertical angular step width of the scanner, and the horizontal resolution can be calculated by the horizontal angular step width in TLS or the scan speed and the driving speed in MLS. e.g., if the angular step is θ rad, the resolution at scanning distance D is $\theta \times D$; if the scan speed is L lines/sec and the driving speed is V, the horizontal resolution (scan line spacing) of MLS point cloud is V/L. Obviously, it is difficult to assign the V_g to a certain value because the resolution is changed with the scanning distance. In order not to affect the registration accuracy as much as possible, the V_g should be set as small as possible.

The point clouds contain many outliers and tree leave points. The tree leaves are easy to shake with the wind. The outliers and tree leave points will affect the registration. They should be removed and the point count can be further reduced. These points are located in irregular shapes and are scattered (i.e., scattered points), so their curvatures are often larger than those of most other man-made objects (i.e., smooth points). It has been found experimentally that most of the scattered points can be removed by curvature filtering. The points whose curvatures are larger than a threshold C are regarded as scattered points. When C is set to 0.05, most scattered points can be removed (Figure 2).

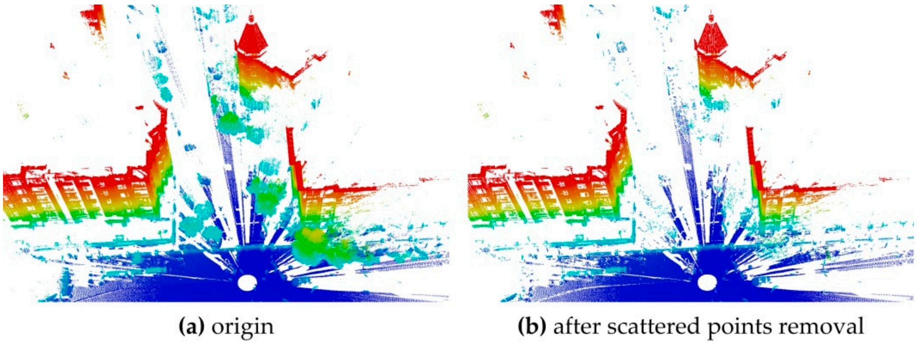

(a) origin **(b)** after scattered points removal

Figure 2. Scattered points removal. The point clouds are rendered by elevation. (**a**) The point cloud before scattered points removal; (**b**) the point cloud after scattered points removal.

After the above steps, there may be still plenty of points. A normal space sampling strategy [35] is employed to select p_S points from a source point cloud and p_T points from a target point cloud as candidate matching points for registration. In tests of the well-known ICP registration [35], the normal space sampling is proved to be superior compared with random sampling. Figure 3 gives an example of the normal space sampling. It can be seen that this algorithm is able to maintain the main features when the sampling ratio is small. The sampling ratio is crucial to improvement of registration efficiency but it would affect the registration accuracy if the ratio is too small. It can be tested in actual registration.

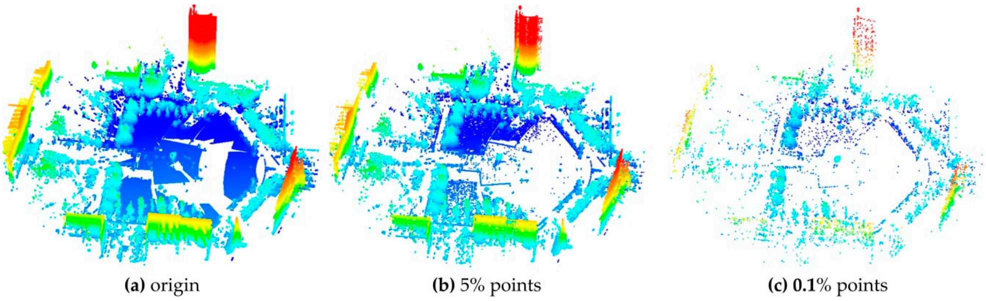

(a) origin **(b)** 5% points **(c) 0.1%** points

Figure 3. Normal space sampling. The point clouds are rendered by elevation. (**a**) The point cloud before normal space sampling; (**b**) the point cloud after 5% points were selected; (**c**) the point cloud after 0.1% points were selected.

3.3. Search Space

The GA registration is a process of evolution which finds the optimal solution in the search space. The search space of point cloud registration is defined between the negative and positive six-dimensional upper bound vector of the six transformation parameters. The upper bounds of α,

β, γ are all 180° and the upper bounds of t_x, t_y, t_z are unlimited when they are not constrained. It is a very large search space. If the search space is too large, the registration may become slow or may be easily premature, or even degenerate into a random search process. Hence, getting a limited search space is important.

In the TLS-TLS and TLS-MLS registration, the prior information of the point clouds is used to limit the search space. The MLS point clouds are directly geo-referenced through the high-accuracy positioning and orientation system (e.g., a GPS/INS system) [10], so they are located in the uniform geodetic coordinate reference frame. However, the TLS point clouds are located in local scanner coordinate systems. To narrow the search space, the position of the scan station acquired by TLS built-in GPS and the quasi-horizontal setup of TLS are used as constraints in our method.

If a built-in GPS is installed in the TLS laser scanner, the position of the scan station can be acquired by the built-in GPS in the field work. There is not much information about the specifications of the built-in GPS. What can be known is that the positioning mode of the built-in GPS is point positioning, whose accuracy is meter-level or even decimetre-level [51], so the upper bounds of t_x, t_y, t_z can be set to 10 m. During scanning, the scanners are usually placed near horizontally, so α, β are close to 0°. Taking into account the actual errors, their upper bounds are set to 5°. Therefore, the search space is set to (α, β, γ, t_x, t_y, t_z | α, $\beta \in [-5°, 5°]$, $\gamma \in [-180°, 180°]$, t_x, t_y, $t_z \in [-10\text{ m}, 10\text{ m}]$), i.e., the upper bound vector U is (5,5,180,10,10,10).

The used TLS laser scanner may not provide a built-in GPS. Under this circumstance, the other auxiliary measurements of the scan station are indispensable—e.g., measuring by an external GPS RTK is a good choice. The GPS RTK—the accuracy of which is centimetre-level—is more accurate than the point positioning mode [51], and then the upper bounds of t_x, t_y, t_z can be set to 0.1 m. The search space for GPS RTK is 1 million times less than that for built-in GPS. Other measuring methods—for example by a Total Station with millimeter-level accuracy—can also be implemented in the field work if necessary. Either way, the upper bounds of t_x, t_y, t_z can be set on the basis of the accuracy level of the auxiliary measurements.

3.4. Proposed NSMS Fitness Function

A fitness function defined to evaluate the solution domain provides the guidance for the GA selection operation. The proposed NSMS fitness function is presented here.

Given correspondences between S and T, a common way to estimate the transformation parameters is to minimize the error metric E between all selected correspondences.

$$E = \frac{1}{N}\sum_{i=1}^{N} d_i,\ d_i = \|\boldsymbol{T}_j - (\boldsymbol{t} + \boldsymbol{R}\boldsymbol{S}_i)\| \tag{4}$$

where, N is the number of correspondences.

In the traditional GA registration method, the error metric is required to convert into a maximum form to define the fitness function. The negative exponential function is a common way to convert the objective function to fitness function

$$F = \mathrm{e}^{-E} \tag{5}$$

where, $\boldsymbol{F}$ is the fitness value.

Differently, the NSMS fitness function is proposed here to avoid the conversion from E to F and to evaluate the solutions accurately. It is expressed as

$$F = \frac{1}{N}\sum_{i=1}^{N} Sc(d_i) \tag{6}$$

where, Sc is a score function of d_i, which must be satisfied with two criteria: (1) 0 < $Sc \leq 1$; (2) Sc is monotonically decreasing. The first criteria is to make the fitness function is optimal even if d_i is larger than 1; the second criteria is utilized to directly access to the maximum F.

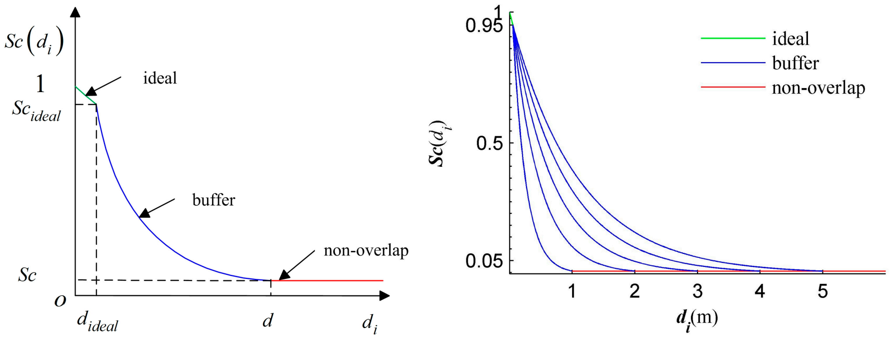
For partially overlapped point cloud registration, a continuous score function satisfied with the above criteria is designed

$$Sc(d_i) = \begin{cases} \exp\left(\log(Sc_{ideal})\frac{d_i}{d_{ideal}}\right) & 0 \leq d_i \leq d_{ideal} \\ Sc\cdot \exp\left(\left(\log\frac{Sc}{Sc_{ideal}}\right)\frac{d_i - d}{d - d_i}\right) & d_{ideal} < d_i \leq d \\ Sc & d_i > d \end{cases} \tag{7}$$

where:

d is the distance threshold that is applied to separate the correspondences into two parts: Inliers in the overlap area and outliers in the non-overlap area;
d_{ideal} is the ideal distance that is applied to further separate the correspondences in the overlap area into two parts: correspondences in the ideal area and correspondences in the buffer area;
Sc is the score of d; and Sc_{ideal} is the score of d_{ideal}.

The *NSMS* fitness function means that the optimization problem is to minimize the sum of the distances between correspondences and to maximize the number of inliers. Just as the negative exponential function is used to convert the objective function in the Equation (5), it is also used to define the score function. The score function and its applied instances are illustrated in Figure 4.



(a) the score function **(b)** the instances

Figure 4. The score function and its instances. (**a**) The diagram of score function; (**b**) the instances of score function under different distance thresholds.

Four parameters should be given for the score function. The Sc is set to a low confidence level value 0.05 and the Sc_{ideal} is set to a high confidence level value 0.95. That is to say, the higher the score, the greater the probability that the correspondence belongs to the inliers. The d_{ideal} need to be set to a small value to ensure that the correspondences with small distances are given high scores. In the registration, the d_{ideal} is set to 5 cm. The d is determined by the attributes of score function. It is shown in Figure 4b that the smaller the d, the steeper the curve; the bigger the d, the flatter the curve. Hence, the smaller the d, the fewer the solutions with high fitness values, and then the more difficult it is to find out the globally optimal solution; the bigger the d, the closer the scores of inliers and outliers, and the more inaccurate the final solution. In the proposed GA registration, the search space is limited by auxiliary constraints of the scanner station (described in Section 3.3), and so the d can be intermediately set to 2 m.

4. Results and Discussions

4.1. Test Datasets

Two datasets were scanned to validate the effectiveness of the proposed GA registration. Each dataset contains two point clouds: a source point cloud and a target point cloud. The dataset 1 as shown in Figure 5a was scanned by a TLS 3D Riegl VZ400 scanner. The number of points in each point cloud is about one hundred million. The station positions providing constraints of the search space were measured by the built-in GPS of the scanner. The important specifications of the scanner from Riegl's official website [49] are displayed in Table 1. The horizontal and vertical scan angular step widths are selectable. Towards the dataset 1, both were set to 0.015° before scanning. The resolution of the point clouds at distance D is about $0.015 \times \pi/180 \times D$ (2.5 cm@100 m).

Table 1. The important specifications of 3D Riegl VZ400 and 2D Riegl VUX [49].

	VZ400	**VUX**
Max range target reflectivity 80%	320 m	420 m
Max range target reflectivity 10%	120 m	150 m
Accuracy	5 mm@100 m	5 mm@100 m
Horizontal angular step width/Scan speed	0.0024°~0.5°	10–250 scan/s
Vertical angular step width	0.0024°~0.288°	0.0036°~0.3°
Laser Beam Divergence	0.3 mrad	0.5 mrad

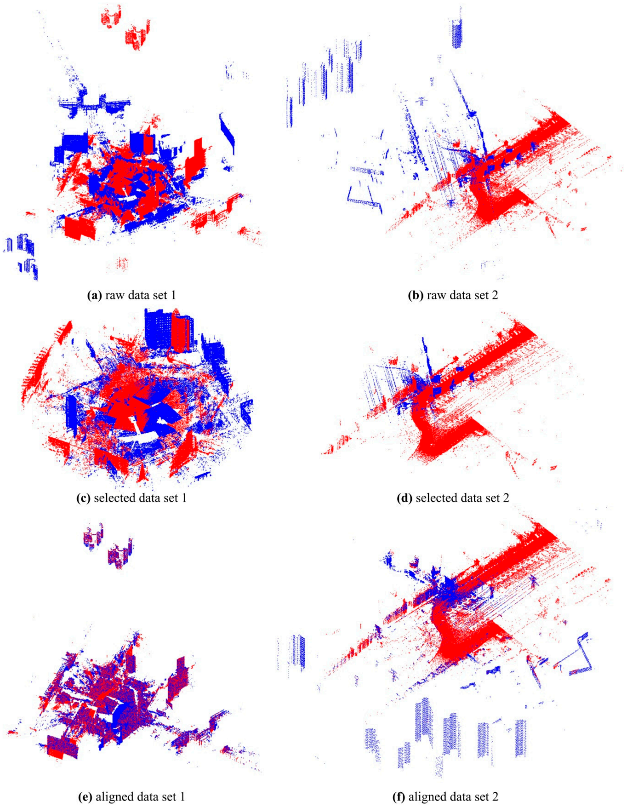

(a) raw data set 1
(b) raw data set 2
(c) selected data set 1
(d) selected data set 2
(e) aligned data set 1
(f) aligned data set 2

Figure 5. The test datasets. The red data are target point clouds and the blue data are source point clouds. (**a**) The raw point clouds of data set 1; (**b**) the raw point clouds of data set 2; (**c**) the point clouds of data set 1 after selection of matching points; (**d**) the point clouds of data set 2 after selection of matching points; (**e**) the registered point clouds of data set 1; (**f**) the registered point clouds of data set 2.

The dataset 2 as shown in Figure 5b includes a source point cloud scanned by the Riegl VZ400 scanner and a target point cloud scanned by MLS. The number of points in each point cloud is about ten million. The positioning vector of the source point cloud for GA registration was also measured by the built-in GPS of the Riegl VZ400. The horizontal and vertical scan angular step widths of the Riegl VZ400 were set to 0.15° before scanning. The resolution of the source point clouds at distance D was about $0.15 \times \pi/180 \times D$ (5.2 cm@20 m). The MLS system contains a 2D Riegl VUX scanner. Its important specifications from Riegl's official website are displayed in Table 1. The scan angular step width and the scan speed are selectable. Towards the MLS system, they were set to 0.5° and 200 scan/s respectively. The vertical resolution of the target point cloud at distance D was about $0.5 \times \pi/180 \times D$ (5.2 cm@10 m). During scanning, the driving speed was approximately 40–45 km/h, and then the horizontal resolution (scan line space) of the target point cloud was about 5–6 cm. The comparison of the two datasets is displayed in Table 2.

Table 2. The comparison of the two test datasets.

	Data Set 1		**Data Set 2**	
	Source	**Target**	**Source**	**Target**
Horizontal angular step width/Scan speed	0.015°	0.015°	0.15°	200 scan/s
Horizontal resolution	2.6 cm@100 m	2.6 cm@100 m	5.2 cm@20 m	5–6 cm
Vertical angular step width	0.015°	0.015°	0.15°	0.3°
Vertical resolution	2.6 cm@100 m	2.6 cm@100 m	5.2 cm@20 m	5.2 cm@10 m
point count	one hundred million		ten million	
Accuracy	millimeter level		centimeter level	

To measure the registration accuracy quantitatively, the root mean square error (RMSE) between the S and its reference was calculated. The reference of the dataset 1 was computed by 5 spherical targets (Figure 6) in the scene. The distances from the targets to the scan station are between 20 m and 40 m. The center for each target was extracted. The center point pairs of the source and target point clouds were registered and then the source point cloud was transformed to form the reference. The extraction and transformation were operated on the Riegl's software RiSCAN PRO [49]. The reference of the dataset 2 was estimated by artificial rough matching (Figure 7) and then the fine ICP registration. 5 point pairs were manually selected for rough matching. The rough matching was operated on the open source software CloudCompare [52] and then the ICP registration was computed by the open source Point Cloud Library (PCL) [36].

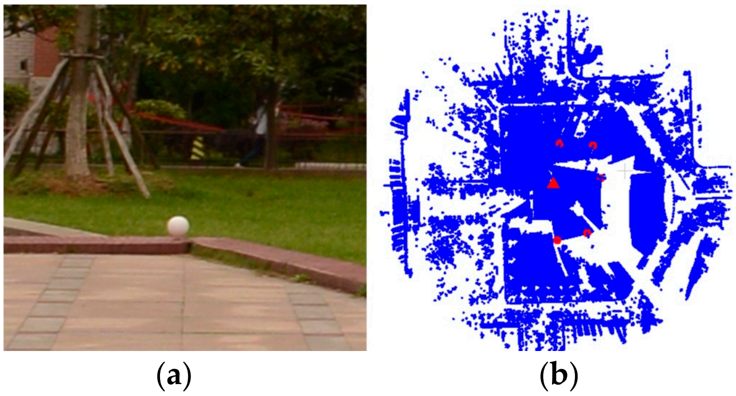
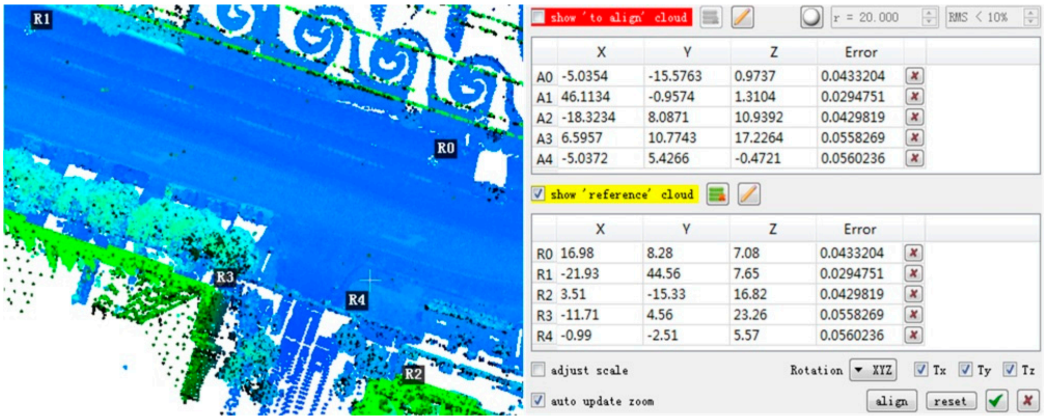


Figure 6. The spherical targets of data set 1. (**a**) the used target (white sphere). (**b**) the distribution of the targets (red circle points). The red triangle denotes the scan station.



□ show 'to align' cloud r = 20.000 RMS < 10%

	X	Y	Z	Error
A0	-5.0354	-15.5763	0.9737	0.0433204
A1	46.1134	-0.9574	1.3104	0.0294751
A2	-18.3234	8.0871	10.9392	0.0429819
A3	6.5957	10.7743	17.2264	0.0558269
A4	-5.0372	5.4266	-0.4721	0.0560236

☑ show 'reference' cloud

	X	Y	Z	Error
R0	16.98	8.28	7.08	0.0433204
R1	-21.93	44.56	7.65	0.0294751
R2	3.51	-15.33	16.82	0.0429819
R3	-11.71	4.56	23.26	0.0558269
R4	-0.99	-2.51	5.57	0.0560236

□ adjust scale Rotation XYZ ☑ Tx ☑ Ty ☑ Tz

☑ auto update zoom

Figure 7. The rough matching for dataset 2 by software Cloud Compare.

4.2. Evaluation of the Proposed GA Registration

The algorithms were programmed using the C++ language and the verification was operated on a computer with Intel Core i7-4790 CPU @ 3.60 GHz, 8 G memory and quad-core processors. Since GA is a stochastic optimizer and the matching points are randomly selected, 50 experiments were carried out for each test. The test was considered a failure if the RMSE was larger than 10 cm. The failure rate was computed to evaluate whether the registration is robust.

The running time of the GA registration mainly consists of two parts—the running time of selection of matching points and the optimizing time of GA evolution. The selection of matching points is a pre-processing step. Without the normal space sampling in the preprocessing, it is the same for all tests of a data set. Hence, the optimizing time of the GA evolution indicates the efficiency of the GA registration. The number of generations (iterations) of the GA evolution was also used to evaluate the efficiency.

The specified algorithm parameters of the evaluation are given in Table 3, in which the first row is the parameters of the selection of matching points, and the second row is the parameters of the GA evolution. As described in Section 3.2, the D_{max} was set to near the effective range of the scanner (Table 1) and the C was set by experiments. According to Table 2, the V_g was set near to the point resolution of dataset 1 with the range at D_{max}. The p_T was set to 5% which could ensure enough object features for registration (Figure 3b), and the p_S effect on the efficiency was tested. The parameters of the GA evolution were set empirically as described in Section 2. The parameters of the fitness function not included in Table 3 were d = 2 m, d_{ideal} = 0.05 m, Sc = 0.05, and Sc_{ideal} = 0.95 (Section 3.4).

Table 3. The specified algorithm parameters of the evaluation of the GA registration.

D_{max}	100 m	V_g	2.5 cm	C	0.05	p_T	5%	p_S	tested
M	100	P_c	0.9	P_m	0.1	MAX_g	300	MAX_b	20

The results of selection of matching points without normal space sampling are illustrated in Table 4 and Figure 5c,d. It took about 255 s for dataset 1 and 51 s for dataset 2. It can be seen that many noisy, redundant and scattered points were removed and the point count was greatly reduced while the main parts of the point clouds were retained. The failure rates, RMSEs and mean optimizing times in different sampling ratio cases are shown in Figure 8. The results showed in Figure 8 indicate that the increase of p_S causes the decrease of failure rate and the RMSE. Through manual inspection, the failures were really the situations where the registration goes wrong as shown in Figure 9. The optimizing time is linear complexity with the sampling ratio. For dataset 1, when p_S was more than 0.1%, the failure rate was 0 and the RMSE became unchanged, so p_S was set to 0.1% in the subsequent experiments and results. Similarly, the p_S of dataset 2 was set to 0.5% in the subsequent experiments and results.

Table 4. The results of selection of matching points without normal space sampling.

Datasets	Point Clouds	Remained Points (%)	Running Time (s)
Data set 1	*S*	11.56	155
	T	11.42	100
Data set 2	*S*	19.46	20
	T	58.38	31

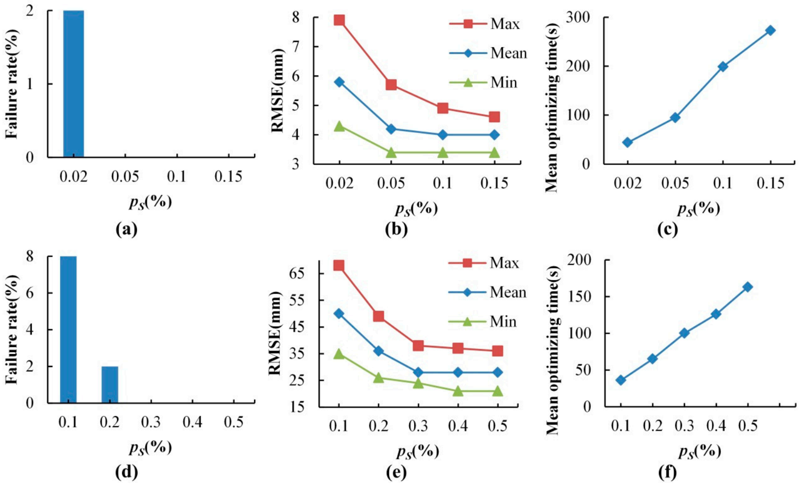
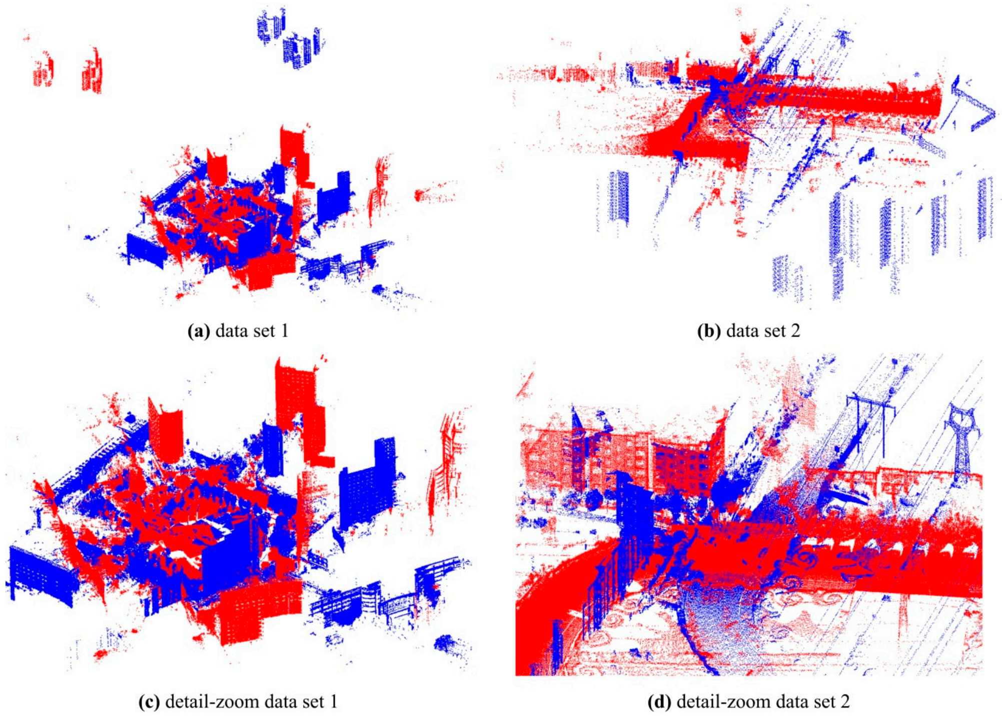


Figure 8. The failure rates, RMSEs and mean optimizing time of different sampling ratios. (**a**–**c**) are the results of data set 1. (**d**–**f**) are the results of data set 2.



(**a**) data set 1

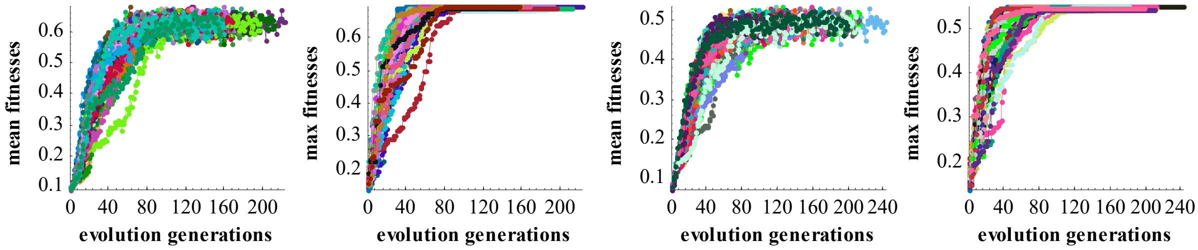
(**b**) data set 2

(**c**) detail-zoom data set 1

(**d**) detail-zoom data set 2

Figure 9. The examples of the situations where the registration goes wrong. The red data are target point clouds and the blue data are source point clouds. (**a**) The mismathed point clouds of data set 1; (**b**) the mismathed point clouds of data set 2; (**c**) a detail-zoom of (**a**); (**d**) a detail-zoom of (**b**).

The proposed GA registration can accurately align the laser scanning point clouds, where the registration accuracy of dataset 1 is 3~5 mm and that of data set 2 is 2~4 cm. The accuracy of dataset 2 is lower one order of magnitude than that of the data set 1. The main reason is that the target point cloud of data set 2 is scanned by MLS and it is much noisier than the TLS point clouds. The aligned datasets are qualitatively shown in Figure 5e,f. Additionally, the mean and maximum fitnesses of GA evolution are shown in Figure 10. It can be seen that the number of evolution generations for the flat parts of the fitness curves is more than half of the total number of evolution generations. It means that the GA's local convergence rate is slow.



(a) data set 1 **(b)** data set 2

Figure 10. The mean and maximum fitness values of GA. A polyline with random color represents the change of the fitness value of one experiment. (**a**) The change of the fitness values of data set 1; (**b**) the change of the fitness values of data set 2.

4.3. Comparative Study of the Fitness Functions

A study was conducted to compare our proposed NSMS fitness function with the published Silva fitness function [43]. The Silva fitness function is a typical MSE-based fitness function as given in Equation (5), which is converted from the objective function as given in Equation (4). Differently, the NSMS fitness function is directly mapped from the distances of the correspondences. In Silva fitness function, the distance threshold d like the algorithm parameter of the score function in the Equation Equation (7) is also applied to separate the correspondences into two parts:

$$d_i = \begin{cases} d_i & d_i \leq d \\ d & d_i > d \end{cases} \tag{8}$$

In the comparative test, the sampling ratio p_S was set to 0.1%, 0.5% for the dataset 1 and dataset 2 respectively according to the experimental results displayed in Figure 8, and the other algorithm parameters were the same as the parameters used in Section 4.2. 50 experiments were also carried out. The accuracy and efficiency of the Silva and NSMS fitness function for proposed GA registration are listed in Table 5. The results show that the NSMS fitness function was more accurate and efficient than the Silva fitness function. The optimizing time of the NSMS fitness function was about 20% less than that of the Silva fitness function.

Table 5. The accuracy and efficiency of different fitness functions.

Datasets	Fitness Functions	RMSEs (mm)			Number of Generations			Mean Optimizing Time (s)
		Min	Max	Mean	Min	Max	Mean	
Dataset 1	Silva	4.2	5.5	4.8	130	219	171	250
	NSMS	3.4	4.9	4.0	119	222	160	199
Dataset 2	Silva	61.1	97.9	75.1	107	224	162	202
	NSMS	22.9	36.1	28.8	93	242	154	163

4.4. Registration Integrating ICP with GA

To accelerate the convergence of optimization, the ICP was integrated with GA for registration. The combined method is called GA + ICP here. The integrating strategy is that ICP is executed after GA evolves some generations and most of the individuals in the population are located in a narrowed search space. Because the fitness values in the narrowed search space were close, to control the evolution generations of GA, the second terminating condition of GA (Section 2) could be modified to "if the number of generations, which the difference between the best fitness of the current generation and that of the previous generation is less than a given minor number e, is equal to a given value MAX_b, the evolution is also stopped". e is the searching control parameter. This is the only changed content of GA.

The GA + ICP was tested, where e was set to 0.001 and 50 experiments were also carried out. The point-to-plane ICP [33] was applied. To reject invalid correspondences in ICP, the correspondence distance threshold was set to 0.2 m and the constraint normal angle is set to 10°. The results illustrated in Figure 11 indicate that the optimizing efficiency of GA + ICP increased by about 50% compared with the GA registration alone.

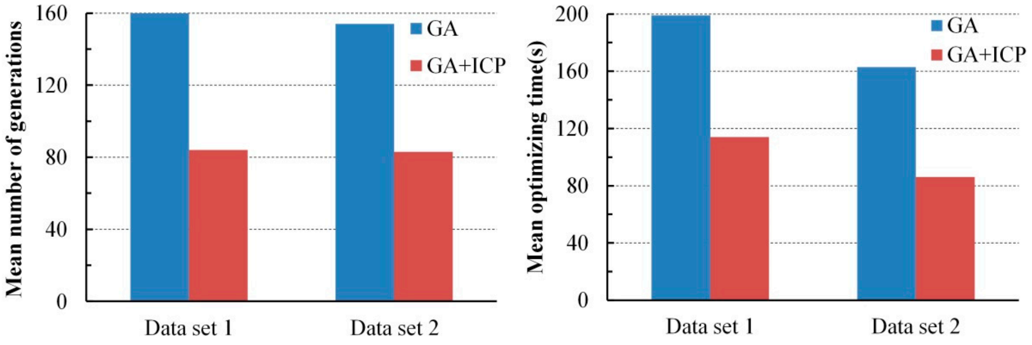


Figure 11. The mean number of generations and optimizing time of GA registration and GA + ICP.

5. Conclusions

This paper proposes an accurate and efficient GA registration method for automatic alignment of two TLS point clouds or two point clouds scanned by TLS and MLS respectively. It is divided into five main steps: selection of matching points, initialization of the population, transformation of matching points, calculation of fitness values, and genetic operation. In order to make the GA registration workable, the aided localization and priori quasi-horizontal orientation of the scan station were used as constraints to narrow the search space. To get accurate results, a new fitness function named 'normalized sum of matching scores' (NSMS) is proposed to evaluate the solutions instead of the MSE-based fitness function. To improve the registration efficiency, the selection of matching points was first applied to eliminate the far, redundant and noisy points and to select partial points representing the main features before GA evolution. Besides, the calculation of fitness values, the most time-consuming step of GA evolution, was parallel-computed.

Two test datasets including a TLS-TLS data set and a TLS-MLS data set were scanned to validate the effectiveness of the proposed GA registration. The experimental results indicate that the RMSE of TLS point clouds registration is 3~5 mm and the RMSE of registration between TLS and MLS point clouds is 2~4 cm. In addition, the proposed NSMS fitness function is more accurate and efficient than the existing Silva fitness function.

To accelerate the convergence of optimization, the ICP was integrated with GA for registration. The integrating strategy is that ICP is executed after GA evolves some generations and most of the individuals in the population are located in a narrowed search space. The combined method was also tested with the two test datasets. The optimizing efficiency of the integrated method was increased by about 50% compared with that of GA registration alone.

The proposed GA registration method can get globally optimal solutions in the search space without initial solutions and the feature extraction is also not required. However, in current algorithms of the first step, only the moving tree leave points can be removed out. The other moving points—e.g., the moving car point—can not be removed. A few moving points may not effect the registration. But it may not be true when there are too many moving objects in the scene. This special case would be considered and perfected in the follow-up work. Further research will mainly focus on extending the proposed method to automatically align multi-view TLS point clouds, multi-strip MLS point clouds or hybrid multi-view point clouds.

Acknowledgments: This research was supported by the National Natural Science Foundation of China (Grant No. 41771486 and Grant No. 41401527). The authors would like to thank the Superscene Information Technology Co., Ltd. (Wuhan, China) for providing the laser scanning systems.

Author Contributions: Li Yan and Junxiang Tan conceived and designed the experiments; Junxiang Tan, Changjun Chen and Hua Liu performed the experiments; Junxiang Tan and Hua Liu analyzed the data; Junxiang Tan, Changjun Chen and Hong Xie contributed reagents/materials/analysis tools; Changjun Chen , Hua Liu and Hong Xie helped to prepare the manuscript; Li Yan, Junxiang Tan, Hua Liu and Hong Xie wrote the paper. All authors read and approved the final manuscript.

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

References

1. Yang, B.; Zang, Y.; Dong, Z.; Huang, R. An automated method to register airborne and terrestrial laser scanning point clouds. *ISPRS J. Photogramm. Remote Sens.* **2015**, *109*, 62–76. [CrossRef]
2. Abbas, M.A.; Lichti, D.D.; Chong, A.K.; Setan, H.; Majid, Z. An on-site approach for the self-calibration of terrestrial laser scanner. *Measurement* **2014**, *52*, 111–123. [CrossRef]
3. Larsson, S.; Kjellander, J.A.P. Motion control and data capturing for laser scanning with an industrial robot. *Robot. Auton. Syst.* **2006**, *54*, 453–460. [CrossRef]
4. Yastikli, N. Documentation of cultural heritage using digital photogrammetry and laser scanning. *J. Cult. Herit.* **2007**, *8*, 423–427. [CrossRef]
5. Ergincan, F.; Çabuk, A.; Avdan, U.; Tün, M. Advanced technologies for archaeological documentation: Patara case. *Sci. Res. Essays* **2010**, *5*, 2615–2629.
6. Abellán, A.; Calvet, J.; Vilaplana, J.M.; Blanchard, J. Detection and spatial prediction of rockfalls by means of terrestrial laser scanner monitoring. *Geomorphology* **2010**, *119*, 162–171. [CrossRef]
7. Corsini, A.; Castagnetti, C.; Bertacchini, E.; Rivola, R.; Ronchetti, F.; Capra, A. Integrating airborne and multi-temporal long-range terrestrial laser scanning with total station measurements for mapping and monitoring a compound slow moving rock slide. *Earth Surf. Process. Landf.* **2013**, *38*, 1330–1338. [CrossRef]
8. Akgul, M.; Yurtseven, H.; Akburak, S.; Demir, M.; Cigizoglu, H.K.; Ozturk, T.; Eksi, M.; Akay, A.O. Short term monitoring of forest road pavement degradation using terrestrial laser scanning. *Measurement* **2017**, *103*, 283–293. [CrossRef]
9. Puente, I.; González-Jorge, H.; Martínez-Sánchez, J.; Arias, P. Review of mobile mapping and surveying technologies. *Measurement* **2013**, *46*, 2127–2145. [CrossRef]
10. Barber, D.; Mills, J.; Smith-Voysey, S. Geometric validation of a ground-based mobile laser scanning system. *ISPRS J. Photogramm. Remote Sens.* **2008**, *63*, 128–141. [CrossRef]
11. Williams, K.; Olsen, M.J.; Roe, G.V.; Glennie, C. Synthesis of transportation applications of mobile LiDAR. *Remote Sens.* **2013**, *5*, 4652–4692. [CrossRef]
12. Yan, L.; Liu, H.; Tan, J.; Li, Z.; Xie, H.; Chen, C. Scan Line Based Road Marking Extraction from Mobile LiDAR Point Clouds. *Sensors* **2016**, *16*, 903. [CrossRef] [PubMed]
13. Guan, H.; Li, J.; Cao, S.; Yu, Y. Use of mobile LiDAR in road information inventory: A review. *Int. J. Image Data Fusion* **2016**, *7*, 219–242. [CrossRef]
14. Sairam, N.; Nagarajan, S.; Ornitz, S. Development of Mobile Mapping System for 3D Road Asset Inventory. *Sensors* **2016**, *16*, 367. [CrossRef] [PubMed]

15. Yang, B.; Liu, Y.; Liang, F.; Dong, Z. Using Mobile Laser Scanning Data for Features Extraction of High Accuracy Driving Maps. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *XLI-B3*, 433–439. [CrossRef]
16. Wu, H.; Scaioni, M.; Li, H.; Li, N.; Lu, M.; Liu, C. Feature-constrained registration of building point clouds acquired by terrestrial and airborne laser scanners. *J. Appl. Remote Sens.* **2014**, *8*, 83587. [CrossRef]
17. Theiler, P.W.; Wegner, J.D.; Schindler, K. Globally consistent registration of terrestrial laser scans via graph optimization. *ISPRS J. Photogramm. Remote Sens.* **2015**, *109*, 126–138. [CrossRef]
18. Yang, B.; Zang, Y. Automated registration of dense terrestrial laser-scanning point clouds using curves. *ISPRS J. Photogramm. Remote Sens.* **2014**, *95*, 109–121. [CrossRef]
19. Yang, M.Y.; Cao, Y.; Mcdonald, J. Fusion of camera images and laser scans for wide baseline 3D scene alignment in urban environments. *ISPRS J. Photogramm. Remote Sens.* **2011**, *66*, S52–S61. [CrossRef]
20. Al-Manasir, K.; Fraser, C.S. Registration of terrestrial laser scanner data using imagery. *Photogramm. Rec.* **2006**, *21*, 255–268. [CrossRef]
21. Weinmann, M.; Weinmann, M.; Hinz, S.; Jutzi, B. Fast and automatic image-based registration of TLS data. *ISPRS J. Photogramm. Remote Sens.* **2011**, *66*, S62–S70. [CrossRef]
22. Kang, Z.; Li, J.; Zhang, L.; Zhao, Q.; Sisi, Z. Automatic Registration of Terrestrial Laser Scanning Point Clouds using Panoramic Reflectance Images. *Sensors* **2009**, *9*, 2621–2646. [CrossRef] [PubMed]
23. Salvi, J.; Matabosch, C.; Fofi, D.; Forest, J. A review of recent range image registration methods with accuracy evaluation. *Image Vis. Comput.* **2007**, *25*, 578–596. [CrossRef]
24. Bernardini, F.; Rushmeier, H. *The 3D Model Acquisition Pipeline*; Eurographics 2000 State of the Art Report (STAR), 2002; Wiley Online Library: Hoboken, NJ, USA, 2002; pp. 149–172.
25. Seeger, S.; Laboureux, X. Feature extraction and registration: An overview. In *Principles of 3D Image Analysis and Synthesis*; Springer: Boston, MA, USA, 2002; pp. 153–166.
26. Johnson, A.E.; Hebert, M. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Trans. Pattern Anal. Mach. Intell.* **1999**, *21*, 433–449. [CrossRef]
27. Rusu, R.B.; Blodow, N.; Beetz, M. Fast point feature histograms (FPFH) for 3D registration. In Proceedings of the IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009; pp. 3212–3217.
28. Tombari, F.; Salti, S.; Stefano, L.D. Performance Evaluation of 3D Keypoint Detectors. *Int. J. Comput. Vis.* **2013**, *102*, 236–243. [CrossRef]
29. Hänsch, R.; Hellwich, O.; Weber, T. Comparison of 3D Interest Point Detectors and Descriptors for Point Cloud Fusion. In Proceedings of the PCV 2014, Zurich, Switzerland, 5–7 September 2014; pp. 57–64.
30. Rodrigues, M.; Fisher, R.; Liu, Y. Special issue on registration and fusion of range images. *Comput. Vis. Image Underst.* **2002**, *87*, 1–7. [CrossRef]
31. Theiler, P.W.; Wegner, J.D.; Schindler, K. Keypoint-based 4-Points Congruent Sets—Automated marker-less registration of laser scans. *ISPRS J. Photogramm. Remote Sens.* **2014**, *96*, 149–163. [CrossRef]
32. Besl, P.J.; Mckay, N.D. A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **1992**, *14*, 239–256. [CrossRef]
33. Chen, Y.; Medioni, G. Object modelling by registration of multiple range images. *Image Vis. Comput.* **1992**, *10*, 145–155. [CrossRef]
34. Low, K. *Linear Least-Squares Optimization for Point-to-Plane ICP Surface Registration*; Technical Report; University of North Carolina: Chapel Hill, NC, USA, 2004.
35. Rusinkiewicz, S.; Levoy, M. Efficient variants of the ICP algorithm. In Proceedings of the Third International Conference on 3-D Digital Imaging and Modeling, Quebec City, QC, Canada, 28 May–1 June 2001; pp. 145–152.
36. Holz, D.; Ichim, A.E.; Tombari, F.; Rusu, R.B. Registration with the Point Cloud Library: A Modular Framework for Aligning in 3-D. *IEEE Robot. Autom. Mag.* **2015**, *22*, 110–124. [CrossRef]
37. Zhu, J.; Meng, D.; Li, Z.; Du, S.; Yuan, Z. Robust registration of partially overlapping point sets via genetic algorithm with growth operator. *IET Image Process.* **2014**, *8*, 582–590. [CrossRef]
38. Man, K.F.; Tang, K.S.; Kwong, S. Genetic algorithms: Concepts and applications. *IEEE Trans. Ind. Electron.* **1996**, *43*, 519–534. [CrossRef]
39. Jacq, J.J.; Roux, C. Registration of 3-D images by genetic optimization. *Pattern Recognit. Lett.* **1995**, *16*, 823–841. [CrossRef]
40. Brunnstrom, K.; Stoddart, A.J. Genetic Algorithms for Free-Form Surface Matching. In Proceedings of the International Conference on Pattern Recognition, Vienna, Austria, 25–29 August 1996; pp. 689–693.

41. Yamany, S.M.; Ahmed, M.N.; Farag, A.A. A new genetic-based technique for matching 3-D curves and surfaces. *Pattern Recognit.* **1999**, *32*, 1817–1820. [CrossRef]
42. Silva, L.; Bellon, O.R.; Boyer, K.L. Enhanced, robust genetic algorithms for multiview range image registration. In Proceedings of the Fourth International Conference on 3-D Digital Imaging and Modeling, Banff, AB, Canada, 6–10 October 2003; pp. 268–275.
43. Silva, L.; Bellon, O.R.; Boyer, K.L. Precision range image registration using a robust surface interpenetration measure and enhanced genetic algorithms. *IEEE Trans. Pattern Anal.* **2005**, *27*, 762–776. [CrossRef] [PubMed]
44. Schenk, S.; Hanke, K. Genetic Algorithms for Automatic Registration of Laser Scans with Imperfect and Subdivided Features (GAReg-ISF). *Photogramm. Fernerkund. Geoinf.* **2009**, *2009*, 23–32. [CrossRef]
45. Lomonosov, E.; Chetverikov, D.; Rt, A. Pre-registration of arbitrarily oriented 3D surfaces using a genetic algorithm. *Pattern Recognit. Lett.* **2006**, *27*, 1201–1208. [CrossRef]
46. Brindle, A. Genetic Algorithms for Function Optimization. Ph.D. Thesis, University of Alberta, Edmonton, AB, Canada, 1981.
47. Michalewicz, Z.; Janikow, C.Z.; Krawczyk, J.B. A modified genetic algorithm for optimal control problems. *Comput. Math. Appl.* **1992**, *23*, 83–94. [CrossRef]
48. OpenMP Home Page. Available online: www.openmp.org (accessed on 20 June 2017).
49. RIEGL Laser Measurement Systems. Available online: www.riegl.com (accessed on 20 June 2017).
50. Pauly, M.; Keiser, R.; Gross, M. Multi-scale Feature Extraction on Point-Sampled Surfaces. *Comput. Graph. Forum* **2003**, *22*, 281–289. [CrossRef]
51. Colomina, I. On Trajectory Determination for Photogrammetry and Remote Sensing: Sensors, Models and Exploitation. In Proceedings of the Photogrammetric Week, Stuttgart, Germany, 7–11 September 2015; pp. 131–142.
52. CloudCompare 3D Point Cloud and Mesh Processing Software. Available online: www.cloudcompare.org (accessed on 3 August 2017).