



# Article A Point Cloud Registration Method Based on Histogram and Vector Operations

Yanan Zhang <sup>1,2</sup>, Dayong Qiao <sup>1,2,3,\*</sup>, Changfeng Xia <sup>4</sup> and Qing He <sup>5</sup>

- Key Laboratory of Micro/Nano Systems for Aerospace, Ministry of Education, Northwestern Polytechnical University, Xi'an 710072, China
- <sup>2</sup> Shaanxi Province Key Laboratory of Micro and Nano Electro-Mechanical Systems, Northwestern Polytechnical University, Xi'an 710072, China
- <sup>3</sup> Ningbo Institute of Northwestern Polytechnical University, 218 Qingyi Road, Ningbo 315103, China
- <sup>4</sup> Xi'An Zhisensor Technologies Co., Ltd., Xi'an 710077, China
- <sup>5</sup> Konka Group Company Limted, Kejinan 12 Road 28, Shenzhen 518063, China
- \* Correspondence: dyqiao@nwpu.edu.cn

**Abstract:** Point-pair registration in a real scene remains a challenging task, due to the complexity of solving three transformations (scale, rotation, and displacement) simultaneously, and the influence of noise and outliers. Aimed at this problem, a registration algorithm based on histogram and vector operations is proposed in this paper. This approach converts point-based operations into vector-based operations, thereby decomposing the registration process into three independent steps solving for scale transformation factors, rotation matrices, and displacement vectors, which reduces the complexity of the solution and avoids the effects of scaling in the other two processes. The influence of outliers on the global transformation matrix is simultaneously eliminated using a histogram-based approach. Algorithm performance was evaluated through a comparison with the most commonly used SVD method in a series of validation experiments, with results showing that our methodology was superior to SVD in the cases with scaling transformation or outliers.

Keywords: point-pair registration; vector-based operation; histogram-based operation

### 1. Introduction

The use of 3D point cloud processing has seen rapid expansion over the last several years, promoted by developments in computer hardware and increased processing power. Recent applications have included a broad range of fields, such as industrial production, autonomous driving, and cultural relic restoration [1–3]. However, it remains difficult to completely reconstruct objects using a single depth sensor, due to field-of-view limitations and occlusions. As such, multiple depth sensors are typically required to measure an object from multiple angles and positions. Reconstructed results can then be converted to the same coordinate system and fused into a complete point cloud [4]. This process requires point cloud registration, a challenging task in pattern recognition [5], computer vision [6], robotics [7], and industrial measurements [8]. Its purpose is to identify a transformation matrix from one coordinate system to another, allowing the point clouds from different sensors to be converted into the same coordinate system, thereby forming a complete point cloud for the measured object [9–11].

Point cloud registration algorithms are typically implemented in three steps. The first step involves determining the correspondence of points between two sets. The second step requires calculating a transformation matrix between the two point sets using this correspondence. The third step applies iterative methods to optimize the transformation matrix [12]. The first step can be achieved using a variety of techniques, including distance-based [13] and feature-based methods [14]. The third step can primarily be divided into optimization methods based on distance [15], annealing, and soft correspondence [16],



**Citation:** Zhang, Y.; Qiao, D.; Xia, C.; He, Q. A Point Cloud Registration Method Based on Histogram and Vector Operations. *Electronics* **2022**, *11*, 4172. https://doi.org/10.3390/ electronics11244172

Academic Editors: Peter Sarcevic, Sašo Tomažič, Akos Odry, Sara Stančin and Gábor Kertész

Received: 11 November 2022 Accepted: 7 December 2022 Published: 14 December 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/). similarity measurements [17], and probability density estimation [18]. In terms of the second step, most techniques use an algorithm based on SVD (singular value decomposition) to calculate the transformation matrix [19–23]. This approach is not only fast, but it also produces the smallest errors when only rotation and translation transformations are applied between point sets [24].

When only rotation and translation operations are conducted between point sets, the algorithm can achieve good results. However, It is very difficult for the SVD algorithm to identify and handle the scale transformation. Here, scale transformation is a linear transformation that enlarges or shrinks objects by a scale factor that is the same in all directions. In practice, the sizes of two reconstructed point clouds are always different. Therefore, if scale transformation is neglected, not only can the transformation relationship of two point clouds not be correctly expressed, but also unavoidable error will be introduced into rotation and translation processes [25]. These issues are the result of errors and variability in different sensors, which are essentially unavoidable. In addition, all points are involved in calculating the transformation matrix and, as such, calculation errors increase when local noise is too large or outliers are included [26]. Therefore, point clouds must be filtered before this registration process, and it has high requirements for filtering [27–29].

In response to the above problems, scholars have proposed various solutions. The PFH (point feature histogram) algorithm proposed by Radu computes the registration results by extracting an optimal set that best characterizes a given point cloud through feature histograms [30]. This method is insensitive to outliers, but the computational efficiency is generally low due to the high computational dimension (16D). Baowei utilizes the PCA (Principal Component Analysis) algorithm to calculate the covariance matrix of the source and target point clouds, and finds the rotation and transformation matrices between the source and target point clouds based on the covariance matrix [25]. This method works well in the case of scale transformation, but the computation time is also long, taking several minutes to compute 100,000 points. In addition, Xuyan's proposed method based on triangle similarity ratio consistency uses LASH (Local Angle Statistics Histogram) to detect triangle matching points with the same similarity for computing multiple transformations between two point clouds, which has been proved to be robust to noise [31]. Xu's method applies the NVP (normal vector and particle swarm optimization) algorithm to find the correspondence points and employs the quaternion method to align the correspondence points, which achieves high accuracy even when some data are lost [32]. Yet, this method generally requires a long time when evaluating multiple transformation matrices.

Aimed at this complex registration process, this study proposes a registration method based on histogram and vector operations. To avoid the mutual influence of registration parameters, our method converts point-based operations into vector-based operations, thereby decomposing the registration process into three independent steps solving for scale transformation factors, rotation matrices, and displacement vectors, which reduces the complexity of the solution. In case of noise and outlier situations, histograms are utilized to obtain Rodrigues parameters (i.e., scale factor, rotation axis vector, sine and cosine values of the rotation angle), following the processes to calculate the transformation matrices based on Rodrigues' rotation formula. Finally, four experiments demonstrate that the proposed method has good registration accuracy both in the case of scale transformation and outliers.

Compared to existing methods, the novelty of this paper is: (i) Other methods address the scale and outlier problem in terms of feature extraction, and our method addresses both of these problems in the computation of the transformation matrix. (ii) We calculated the scale factor, rotation axis vector, rotation angle, and displacement vector separately using vector and histogram methods to reduce the error interactions of these variables. (iii) We make full use of the relationship between vectors (Equations (5)–(8)) to calculate the rotation axis vector and the rotation angle. (iv) Even under Gaussian noise, the distribution of Rodrigues parameters will not be the standard positive-terrestrial distribution, which makes the error of methods such as averaging very large. In contrast, our method uses histograms to greatly avoid this systematic error and provides a way for others to find the maximum likelihood variable for non-Gaussian distributions.

This paper is organized as follows. The principles of the proposed registration methodology are introduced in Section 2. Experimental details are provided in Section 3. The discussions are presented in Section 4. The conclusion is given in the last section.

#### 2. Methods

2.1. Symbol Description

Since many symbols are used in this paper, Table 1 explains the meaning of these symbols.

Table 1. The list of symbols used in this paper.

| Symbol                          | Description  |  |  |  |
|---------------------------------|--|--|--|--|
| $\{p_i\}$                       | The initial point cloud  |  |  |  |
| $\{p'_i\}$                      | The target point cloud   |  |  |  |
| $p_i$                           | a point in the point cloud $\{p_i\}$                                 |  |  |  |
| $p'_i$                          | The corresponding point of $p_i$ in point cloud $\{p'_i\}$ .         |  |  |  |
| $\vec{p}_i$                     | The coordinates of point $p_i$                                       |  |  |  |
| $\vec{p}'_i$                    | The coordinates of point $p'_i$                                      |  |  |  |
| Ĥ                               | Homogeneous transformation matrix between $\{p_i\}$ and $\{p'_i\}$ . |  |  |  |
| Ι                               | $3 \times 3$ unit matrix.  |  |  |  |
| ñ                               | The rotation axis of $\{p_i\}$ transformed to $\{p'_i\}$             |  |  |  |
| $(k_x, k_y, k_z)$               | The coordinates of the rotation axis $\vec{n}$                       |  |  |  |
| Κ                               | The anti-symmetric matrix representation of $(k_x, k_y, k_z)$ .      |  |  |  |
| heta                            | The rotation angle of $\{p_i\}$ transformed to $\{p'_i\}$            |  |  |  |
| R                               | The rotation matrix of $\{p_i\}$ transformed to $\{p'_i\}$           |  |  |  |
| R <sub>real</sub>               | The real value of <i>R</i>   |  |  |  |
| $\vec{R}_{calculate}$           | The calculated value of <i>R</i>                                     |  |  |  |
| $\stackrel{t}{\longrightarrow}$ | The translation vector of $\{p_i\}$ transformed to $\{p'_i\}$        |  |  |  |
| t <sub>real</sub>               | The real value of $\vec{t}$  |  |  |  |
| t <sub>calculate</sub>          | The calculated value of $\vec{t}$                                    |  |  |  |
| scale                           | The scale factor of $\{p_i\}$ transformed to $\{p'_i\}$              |  |  |  |
| scale <sub>real</sub>           | The real value of <i>scale</i>                                       |  |  |  |
| scale <sub>calculate</sub>      | The calculated value of <i>scale</i>                                 |  |  |  |
| ABC                             | Three points in the point cloud $\{p_i\}$                            |  |  |  |
| A' B' C'                        | The corresponding points of points A, B and C in $\{p'_i\}$ .        |  |  |  |
| axis <sub>ABC</sub>             | The rotation axis of $A B C$ transformed to $A' B' C'$               |  |  |  |
| $\overrightarrow{p_m p_n}$      | The vector from point $p_m$ to point $p_n$                           |  |  |  |
| $\ \overrightarrow{p_mp_n}\ $   | The norm of vector $\overline{p_m p_n}$                              |  |  |  |
| Х                               | Cross product of vectors   |  |  |  |
| •                               | Dot product of vectors   |  |  |  |
| normalize()                     | Normalization of vectors   |  |  |  |
| ε                               | The accuracy of registration algorithms                              |  |  |  |
| σ                               | Gaussian noise standard deviation                                    |  |  |  |

2.2. Understanding of the Registration Process

We all know that the registration process between two point clouds  $\{p_i\}$  and  $\{p'_i\}$  is actually to find the transformation matrix H, which makes the corresponding points  $p_i$  and  $p'_i$  between the two point clouds meet the following relationship:

$$p_i' = H p_i \tag{1}$$

In homogeneous coordinates,

$$H = \begin{bmatrix} R & \vec{t} \\ \vec{0} & \frac{1}{scale} \end{bmatrix}$$
(2)

Among them, R,  $\vec{t}$ , and *scale* are the rotation matrix, displacement vector, and scale factor for transforming point cloud  $\{p_i\}$  to point cloud  $\{p'_i\}$ . Therefore, the registration process of two point clouds is actually the process of solving the R,  $\vec{t}$ , and *scale* components of the two point cloud transformations.

#### 2.3. Rodrigues' Rotation Formula

Rodrigues' rotation formula states that if a point cloud  $\{p_i\}$  is rotated by an angle  $\theta$  around the unit vector  $\vec{n}$  to form the point cloud  $\{p'_i\}$ , the associated rotation matrix *R* can be represented by:

$$R = I + (1 - \cos\theta)K^2 + \sin\theta K \tag{3}$$

Here, *I* is a 3 \* 3 unit matrix and *K* can be expressed as:

$$K = \begin{bmatrix} 0 & -k_z & k_y \\ k_z & 0 & -k_x \\ -k_y & k_x & 0 \end{bmatrix}$$
(4)

where the coordinates of the rotation axis  $\vec{n}$  are denoted  $(k_x, k_y, k_z)$ . Once  $\vec{n}$  and the rotation angle  $\theta$  are determined, the rotation matrix R can be calculated from the point cloud  $\{p_i\}$  to the point cloud  $\{p_i'\}$ .

### 2.4. Registration of Three Pairs of Corresponding Points

Three points (*A*, *B*, and *C*) can be identified in a given point cloud  $\{p_i\}$ , corresponding to three points (*A*', *B*', and *C*') in the point cloud  $\{p'_i\}$ . A scale transformation factor from points *A*, *B*, and *C* to points *A*', *B*', and *C*' can then be calculated as:

$$scale = \frac{\|\overrightarrow{A'B'}\|}{\|\overrightarrow{AB}\|} = \sqrt{\frac{(x'_A - x'_B)^2 + (y'_A - y'_B)^2 + (z'_A - z'_B)^2}{(x_A - x_B)^2 + (y_A - y_B)^2 + (z_A - z_B)^2}}$$
(5)

where  $(x_A, y_A, z_A)$ ,  $(x_B, y_B, z_B)$ ,  $(x'_A, y'_A, z'_A)$ , and  $(x'_B, y'_B, z'_B)$  are the coordinates of points *A*, *B*, *A*' and *B*', respectively. The axis of the rotation vector can be expressed as:

$$\overrightarrow{axis}_{ABC} = normalize((\frac{\overrightarrow{AB}}{\|\overrightarrow{AB}\|} - \frac{\overrightarrow{A'B'}}{\|\overrightarrow{A'B'}\|}) \times (\frac{\overrightarrow{AC}}{\|\overrightarrow{AC}\|} - \frac{\overrightarrow{A'C'}}{\|\overrightarrow{A'C'}\|}))$$
(6)

Since the unit vector  $\overrightarrow{AB} = \overrightarrow{AB}$  is rotated around the rotation axis  $\overrightarrow{axis}_{ABC}$  to the vector  $\overrightarrow{A'B'}$ . The vector  $\overrightarrow{axis}_{ABC}$  is then perpendicular to the vector  $\overrightarrow{AB} = -\overrightarrow{A'B'}$  and  $\overrightarrow{axis}_{ABC}$  is also perpendicular to  $\overrightarrow{AC} = -\overrightarrow{A'C'}$ . Thus, Equation (6) can be used to calculate the unit vector  $\overrightarrow{axis}_{ABC}$ , where *normalize*() represents a normalization operation. The cosine of the angle  $\theta$  is given by:

$$\cos\theta = \frac{\left(\frac{\overrightarrow{AB}}{\|\overrightarrow{AB}\|} - \frac{\overrightarrow{A'B'}}{\|\overrightarrow{A'B'}\|} \bullet \overrightarrow{axis_{ABC}}\right) \bullet \left(\frac{\overrightarrow{AC}}{\|\overrightarrow{AC}\|} - \frac{\overrightarrow{A'C'}}{\|\overrightarrow{A'C'}\|} \bullet \overrightarrow{axis_{ABC}}\right)}{\left\|\frac{\overrightarrow{AB}}{\|\overrightarrow{AB}\|} - \frac{\overrightarrow{A'B'}}{\|\overrightarrow{A'B'}\|} \bullet \overrightarrow{axis_{ABC}}\right\| \cdot \left\|\frac{\overrightarrow{AC}}{\|\overrightarrow{AC}\|} - \frac{\overrightarrow{A'C'}}{\|\overrightarrow{A'C'}\|} \bullet \overrightarrow{axis_{ABC}}\right\|}$$
(7)

The sine value of the angle  $\theta$  is given by:

$$\sin \theta = \frac{\left(\frac{\overrightarrow{AB}}{\|\overrightarrow{AB}\|} - \frac{\overrightarrow{A'B'}}{\|\overrightarrow{A'B'}\|} \bullet \overrightarrow{axis_{ABC}}\right) \times \left(\frac{\overrightarrow{AC}}{\|\overrightarrow{AC}\|} - \frac{\overrightarrow{A'C'}}{\|\overrightarrow{A'C'}\|} \bullet \overrightarrow{axis_{ABC}}\right)}{\left\|\frac{\overrightarrow{AB}}{\|\overrightarrow{AB}\|} - \frac{\overrightarrow{A'B'}}{\|\overrightarrow{A'B'}\|} \bullet \overrightarrow{axis_{ABC}}\right\| \cdot \left\|\frac{\overrightarrow{AC}}{\|\overrightarrow{AC}\|} - \frac{\overrightarrow{A'C'}}{\|\overrightarrow{A'C'}\|} \bullet \overrightarrow{axis_{ABC}}\right\|} \bullet \overrightarrow{axis_{ABC}}$$
(8)

When calculating the rotation angle of two vectors, the components in the direction of the rotation axis must be removed first. The angle can then be acquired using Equations (7) and (8) and the rotation matrix R can be calculated from Equation (3). The displacement vector  $\vec{t}$  from points A, B, and C to points A', B', and C' can be calculated as:

$$\vec{t} = \frac{1}{3} \sum_{i=1}^{3} (\frac{1}{scale} \cdot \vec{p}'_i - R \cdot \vec{p}_i)$$
(9)

where  $\vec{p}_i(i = 1, 2, 3)$  represents the coordinates of points *A*, *B*, and *C*, respectively, and  $\vec{p}'_i(i = 1, 2, 3)$  denotes the coordinates of points *A*', *B*', and *C*', respectively.

#### 2.5. Registration of Two Sets of Point Clouds

Point clouds can be grouped using three points, which can uniquely determine a registration relationship. The global registration relationship can then be determined from the registration relationship of each group as follows.

- Step 1: The point clouds  $\{p_i\}$  and  $\{p'_i\}$  are separately grouped using three points.
- Step 2: The technique presented in Section 2.3 is used to calculate the Rodrigues parameters (i.e., scale factor, rotation axis vector, and sine and cosine values of the rotation angle) for each group.
- Step 3: A histogram is used to acquire the global Rodrigues parameters, taking the parameter with the highest probability in the histogram as the global parameter.
- Step 4: A global rotation matrix is acquired using the global Rodrigues parameters.
- Step 5: A global displacement vector is calculated using:

$$\vec{t} = \frac{1}{N} \sum_{i=1}^{N} \left( \frac{1}{scale} \cdot \vec{p}'_i - R \cdot \vec{p}_i \right) \tag{10}$$

where  $\vec{p}_i$  represents the coordinates of the point  $p_i$  in the point cloud  $\{p_i\}$  and  $p'_i$  represents the coordinates of the point  $p'_i$  in the point cloud  $\{p'_i\}$ .

#### 3. Experiments

Algorithm effectiveness was evaluated on the i5-10400 platform. We used a point cloud from the Bremen dataset as the source point cloud  $\{p_i\}$ , which can be seen in Figure 1. Then, we transform the original point cloud according to Equation (11) to obtain the destination point cloud  $\{p'_i\}$ . Registration processing was performed on the two sets of point clouds using our proposed technique and the SVD algorithm.

$$\vec{p}'_i = scale_{real} \cdot (R_{real} \cdot \vec{p}_i + \vec{t_{real}}) \tag{11}$$



Figure 1. The Bremen point cloud used in the validation experiments.

The accuracy of the two algorithms was determined by calculating the error using the following formula:

$$\varepsilon = \frac{1}{N} \sum_{i=1}^{N} \left( \frac{1}{scale_{calculate}} \cdot \vec{p}'_{i} - R_{calculate} \cdot \vec{p}_{i} - \overrightarrow{t_{calculate}} \right)^{2}$$
(12)

where  $scale_{calculate}$ ,  $R_{calculate}$ , and  $\overrightarrow{t_{calculate}}$  are the scale factor, rotation matrix, and displacement vector calculated using our method and the SVD algorithm, respectively. The default SVD scaling factor was set to  $scale_{calculate} = 1$ . Algorithm performance was verified in four distinct cases:

Case 1: Ideal conditions (no scaling transformation, noise, or outliers).

Case 2: Noisy conditions (no scaling transformation or outliers, but noise included).

Case 3: Scaled conditions (no outliers, but noise and a scaling transformation included). Case 4: Outlier conditions (no scaling transformation, but noise and outliers included).

In Equation (11), the rotation matrix and displacement vector of the original point cloud transformation to the target point cloud are:

$$R_{real} = \begin{bmatrix} 0.3536 & 0.3536 & 0.866 \\ -0.7071 & 0.7071 & 0 \\ -0.6124 & -0.6124 & 0.5 \end{bmatrix}$$
(13)

$$\overrightarrow{t_{real}} = \begin{bmatrix} 100 & 200 & 50 \end{bmatrix} \tag{14}$$

When there is no scale transformation, we define the factor of the scale transformation as 1. When there is scale transformation, we define the range of scale transformation as 0.9–1.1. In addition, the Gaussian noise standard deviation in this paper is 0.2, and the outliers are square point clouds with side lengths of 14 \* 14 and spacing of 0.1.

#### 3.1. The Case of No Scale Transformation and No Noise

In this case, the transformations from the origin point cloud to the destination point cloud are rotation and translation transformations. The corresponding transformed point cloud  $\{p'_i\}$  is shown in Figure 2. Then, we calculated the registration parameters according to the steps in Section 2.5.



Figure 2. The point cloud transformed by rotation and translation.

After the grouping process in step 1 and the vector operation in step 2, histograms for the scaling factor, cosine, and sine values of the rotation angle and rotation axis vector in step 3 are shown in Figure 3. We took the value of the maximum probability from the histogram of each Rodrigues parameter in step 3 as the global Rodrigues parameter and calculated the rotation matrix of the registration by Equation (3) (step 4) as:



Figure 3. Histograms for the Rodrigues parameters calculated using our method in Case 1.

$$R_{calculate} = \begin{bmatrix} 0.3536 & 0.3535 & 0.866 \\ -0.707 & 0.7072 & 0 \\ -0.6124 & -0.6123 & 0.5 \end{bmatrix}$$
(15)

In step 5, we calculated the registered displacement vector by Equation (10) as:

$$\overrightarrow{t_{calculate}} = \begin{bmatrix} 100 & 200 & 50 \end{bmatrix}$$
(16)

and those calculated using SVD are given by:

$$R_{calculate} = \begin{bmatrix} 0.3536 & 0.3536 & 0.866 \\ -0.7071 & 0.7071 & 0 \\ -0.6124 & -0.6124 & 0.5 \end{bmatrix}$$
(17)

$$\overrightarrow{t_{calculate}} = \begin{bmatrix} 100 & 200 & 50 \end{bmatrix}$$
(18)

#### 3.2. The Case without a Scaling Transformation but with Gaussian Noise

Gaussian noise with a standard deviation of two times the point spacing ( $\sigma = 0.2$ ) was added to the destination point cloud  $\{p'_i\}$  described in Section 3.1. The destination point cloud  $\{p'_i\}$  with Gaussian noise is shown in Figure 4. Then, we calculated the registration parameters according to the steps in Section 2.5. In this case, histograms for the scaling factor, cosine, and sine values of the rotation angle and rotation axis vector are shown in Figure 5.

The scaling factor was set to 1 and the rotation matrix and displacement vector calculated using our method are:

$$R_{calculate} = \begin{bmatrix} 0.3546 & 0.3539 & 0.8655 \\ -0.7066 & 0.7077 & 0 \\ -0.6124 & -0.6115 & 0.501 \end{bmatrix}$$
(19)

$$\overrightarrow{t_{calculate}} = \begin{bmatrix} 100.0063 & 199.9923 & 49.9844 \end{bmatrix}$$
(20)

Those produced by the SVD method are:

$$R_{calculate} = \begin{bmatrix} 0.3536 & 0.3536 & 0.866 \\ -0.7071 & 0.7071 & 0 \\ -0.6124 & -0.6124 & 0.5 \end{bmatrix}$$
(21)  
$$\overrightarrow{t_{calculate}} = \begin{bmatrix} 100.0005 & 200.0001 & 49.9999 \end{bmatrix}$$
(22)



**Figure 4.** The destination point cloud  $\{p'_i\}$  with noise.



Figure 5. Histograms for the Rodrigues parameters calculated using our method in Case 2.

## 3.3. The Case with Scaling Information and Gaussian Noise

The scaling effects produced by each of these two methods were observed using scale factors (*scale<sub>real</sub>*) ranging from 0.9 to 1.1, applied to the point cloud  $\{p'_i\}$  discussed in Section 3.2. The destination point cloud  $\{p'_i\}$  with Gaussian noise and scale transformation is shown in Figure 6. We then registered the point clouds  $\{p_i\}$  and  $\{p'_i\}$  using each technique. Among them, when the scale factor is 0.9, histograms for the scaling factor, cosine, and sine values of the rotation angle and rotation axis vector are shown in Figure 7.



**Figure 6.** The destination point cloud  $\{p'_i\}$  with scale and noise.



Figure 7. Histograms for the Rodrigues parameters calculated using our method in Case 3.

The scale factor calculated in step 3 is:

$$scale_{calculate} = 0.9 \tag{23}$$

The rotation matrix calculated in step 4 is:

$$R_{calculate} = \begin{bmatrix} 0.3538 & 0.3531 & 0.8659 \\ -0.7067 & 0.7074 & 0 \\ -0.6124 & -0.6122 & 0.4999 \end{bmatrix}$$
(24)

The displacement vector calculated in step 5 is:

$$\overrightarrow{t_{calculate}} = \begin{bmatrix} 100.0035 & 199.9940 & 50.0002 \end{bmatrix}$$
(25)

The rotation matrix and displacement vector produced by the SVD method are:

$$R_{calculate} = \begin{bmatrix} 0.3484 & 0.3578 & 0.8664 \\ -0.7095 & 0.7047 & -0.0057 \\ -0.6125 & -0.6127 & 0.4994 \end{bmatrix}$$
(26)

$$\overrightarrow{t_{calculate}} = \begin{bmatrix} 90.7114 & 180.9782 & 45.3664 \end{bmatrix}$$
(27)

The results for varying scale factors are shown in Figure 8.

It can be seen in Figure 8a that in the process of increasing the scaling factor from 0.9 to 1.1, the maximum error produced by our method is 0.05% and close to zero in most cases. Errors calculated using both our method and SVD are shown in Figure 8b, where it is evident that the SVD method produced the smallest errors for a scaling factor equal to 1, increasing as the scale factor deviated from 1 (e.g., reaching 60 for a scale factor of 1.1). As shown, our method is far less sensitive to changes in the scaling factor.



**Figure 8.** The experimental results for Case 3. (a) The relative error for varying scale factors, calculated using our method. (b) A comparison of errors produced using our method and SVD, applied to Case 3.

#### 3.4. The Case with Outliers and Gaussian Noise

Some outliers will inevitably be generated in point clouds, due to the influence of noise, varying fields of view, and moving objects in the environment. To evaluate the impact on each technique, we added outliers (square point clouds with side lengths of 14 \* 14 and spacing of 0.1) to  $\{p'_i\}$ , as shown in Figure 9.

We then registered the source and destination point clouds using our method and SVD. Among them, the histogram in step 3 for the proposed technique is shown in Figure 10.

The rotation matrix and displacement vector calculated in step 4 and step 5 can be represented as:

$$R_{calculate} = \begin{bmatrix} 0.3532 & 0.3544 & 0.866 \\ -0.7079 & 0.7064 & 0 \\ -0.6119 & -0.6128 & 0.5003 \end{bmatrix}$$
(28)

$$\vec{t_{calculate}} = \begin{bmatrix} 99.9914 & 200.0110 & 49.9973 \end{bmatrix}$$
(29)

Those produced by the SVD method are given by:

$$R_{calculate} = \begin{bmatrix} 0.3484 & 0.3578 & 0.8663 \\ -0.7095 & 0.7046 & -0.0057 \\ -0.6125 & -0.6127 & 0.4994 \end{bmatrix}$$
(30)  
$$\overrightarrow{t_{calculate}} = \begin{bmatrix} 99.9515 & 200.0389 & 50.0006 \end{bmatrix}$$
(31)



**Figure 9.** The point cloud  $\{p'_i\}$  with outliers.



Figure 10. Histograms for the Rodrigues parameters calculated using our method in Case 4.

#### 4. Results and Discussions

The experiments implemented above demonstrate that both our method and the SVD method are able to perform point cloud registration in each of the four cases. The results are shown in Table 2.

| Experiment No.              | 1                  | 2                  | 3        | 4        |
|-----------------------------|--------------------|--------------------|----------|----------|
| Noise                       | No                 | Gaussian           | Gaussian | Gaussian |
| Scaling                     | 1.0                | 1.0                | 0.9      | 1.0      |
| Outliers                    | No                 | No                 | No       | Yes      |
| Error of our method         | $6.59	imes10^{-5}$ | $2.70	imes10^{-3}$ | 0.0017   | 0.0062   |
| Error of SVD method         | 0                  | $7.69	imes10^{-7}$ | 58.1891  | 0.2541   |
| Time consumed by our method | 0.161 s            | 0.178 s            | 0.177 s  | 0.159 s  |
| Time consumed by SVD method | 0.011 s            | 0.013 s            | 0.015 s  | 0.012 s  |

Table 2. Comparisons of our method and the SVD method.

As shown in Table 2, the SVD results are superior in ideal and noisy conditions and the errors of both methods are much smaller than the point spacing of  $\{p_i\}$  and  $\{p'_i\}$ . This is because SVD is optimal for least squares errors, while our method suffers from histogram quantization effects. In the case of scale transformations, our method is better than the SVD method, because it cannot calculate scale transformation factors, which will eventually influence the registration results for angle and displacement transformations. It can be seen from Table 2 that even small-scale transformations could lead to large errors. In contrast, our method was unaffected by scale transformations.

The SVD method calculates a global (optimal) point cloud solution. As such, when mismatches (such as outliers) are included in the correspondence of the two sets, these point pairs participate in the calculation of point clouds. This can negatively affect the final registration results, as was the case for the fourth experiment. However, our method uses a statistical histogram approach involving a rotation axis vector, rotation angle, and scale transformation factor to determine the maximum probability for the global rotation axis, rotation angle, and scale transformation factor, thereby eliminating local outliers and noise. Compared with the SVD algorithm, which uses global points in the calculation of a transformation matrix, the influence of outliers on the registration process is reduced significantly, which improves the resulting accuracy. However, our technique does involve multiple vector operations and a time-consuming normalization step. As such, reducing the computational complexity and runtime will be the subject of a future study.

In addition, it is difficult to calculate scale, angle, and displacement transformations between point clouds while registering pairs of points with similar relationships. Thus, we convert these relationships into vectors, eliminating displacement transformations, and then normalize the vectors to eliminate scale transformations between the vectors. In this way, the purpose of separately calculating each transformation between point clouds is achieved, thereby reducing the difficulty. This study also utilizes the relationships between vectors to directly identify a rotation axis and rotation angle from a source point cloud to a destination point cloud, using Rodrigues' rotation formula to calculate the angle transformation matrix. This approach does not require a matrix decomposition or inversion, eliminating errors caused by failures in these steps.

#### 5. Conclusions

Since SVD algorithm failed to solve the point-pair registration problem when scale transformation and outliers exist, a registration method based on histogram and vector operations is proposed in this paper. This method can reduce the complexity of point-pair registration by converting point-based operations into vector-based operations, and decomposing the registration process into three independent steps solving for scale transformation factors, rotation matrices, and displacement. This method can also eliminate the influence of outliers on the global transformation matrix by using a histogram-based approach. A series of validation experiments were carried out to validate the point-pair registration effect compared to the SVD-based method. The experimental results proved that our method has better performance in terms of relative error and registration error when scale transformations and outliers, our method also has the disadvantage of

being time-consuming due to the involvement of vector operations. How to reduce the computational complexity of vector operations will be the focus of our future study. I believe that our method can become a reliable and efficient registration method in the field of 3D reconstruction and SLAM (Simultaneous Localization and Mapping) in the future.

**Author Contributions:** Methodology, Y.Z.; experiments, Y.Z.; writing—original draft preparation, Y.Z.; writing—review and editing, D.Q., C.X. and Q.H.; supervision, D.Q. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received support from the National Natural Science Foundation of China (Grant No. U21B2035, 62074128) and the Key-Area Research and Development Program of Guang-dong Province (Grant No. 2021B0101410001).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

#### References

- Yang, S.; Lu, H.; Li, J. Multifeature Fusion-Based Object Detection for Intelligent Transportation Systems. *IEEE Trans. Intell. Transp.* Syst. 2022, 1–8. [CrossRef]
- Chen, F.; Brown, G.M.; Song, M. Overview of three-dimensional shape measurement using optical methods. *Opt. Eng.* 2000, 39, 10–21.
- 3. Zhang, Z.H. Review of single-shot 3D shape measurement by phase calculation-based fringe projection techniques. *Opt. Lasers Eng.* **2012**, *50*, 1097–1106. [CrossRef]
- 4. Chen, S.; Nan, L.; Xia, R.; Zhao, J.; Wonka, P. PLADE: A plane-based descriptor for point cloud registration with small overlap. *IEEE Trans. Geosci. Remote. Sens.* 2019, *58*, 2530–2540. [CrossRef]
- 5. Evangelidis, G.D.; Horaud, R. Joint alignment of multiple point sets with batch and incremental expectation-maximization. *IEEE Trans. Pattern Anal. Mach. Intell.* 2017, 40, 1397–1410. [CrossRef]
- Cheng, L.; Chen, S.; Liu, X.; Xu, H.; Wu, Y.; Li, M.; Chen, Y. Registration of laser scanning point clouds: A review. Sensors 2018, 18, 1641. [CrossRef]
- Zhang, X.; Wang, J.; Fang, Y.; Yuan, J. Multilevel humanlike motion planning for mobile robots in complex indoor environments. *IEEE Trans. Autom. Sci. Eng.* 2018, 16, 1244–1258. [CrossRef]
- 8. Zhu, H.; Li, Y.; Yu, J.; Leung, H.; Li, Y. Ensemble registration of multisensor images by a variational Bayesian approach. *IEEE Sensors J.* **2014**, *14*, 2698–2705. [CrossRef]
- Li, L.; Yang, M.; Wang, C.; Wang, B. Cubature Kalman Filter based point set registration for SLAM. In Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1–4 November 2016; pp. 847–852.
- 10. Li, L.; Yang, M.; Wang, C.; Wang, B. Rigid point set registration based on cubature Kalman filter and its application in intelligent vehicles. *IEEE Trans. Intell. Transp. Syst.* **2017**, *19*, 1754–1765. [CrossRef]
- 11. Wang, G.; Zhou, Q.; Chen, Y. Robust non-rigid point set registration using spatially constrained Gaussian fields. *IEEE Trans. Image Process.* **2017**, *26*, 1759–1769. [CrossRef]
- 12. Lu, Y.; Duan, M.; Dai, S. A method of point cloud data block registration with considering distance from point to surface. *Bull. Pol. Acad. Sci.-Tech. Sci.* **2022**, *70*, e140259.
- 13. Besl, P.J.; McKay, N.D. Method for registration of 3-D shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*; SPIE: Bellingham, WA, USA, 1992; Volume 1611, pp. 586–606.
- 14. Renaudin, E.; Habib, A.; Kersting, A.P. Featured-Based Registration of Terrestrial Laser Scans with Minimum Overlap Using Photogrammetric Data. *Etri J.* 2011, *33*, 517–527. [CrossRef]
- 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.
- 16. Gold, S.; Rangarajan, A.; Lu, C.; Pappu, S.; Mjolsness, E. New algorithms for 2D and 3D point matching: Pose estimation and correspondence. *Pattern Recognit.* **1998**, *31*, 1019–1031. [CrossRef]
- 17. Tsin, Y.; Kanade, T. A correlation-based approach to robust point set registration. In Proceedings of the European Conference on Computer Vision, Prague, Czech Republic, 11–14 May 2004; pp. 558–569.
- Myronenko, A.; Song, X. Point set registration: Coherent point drift. *IEEE Trans. Pattern Anal. Mach. Intell.* 2010, 32, 2262–2275. [CrossRef]
- 19. Andrews, C.M.; Henry, A.B.; Soriano, I.M.; Southworth, M.K.; Silva, J.R. Registration techniques for clinical applications of three-dimensional augmented reality devices. *IEEE J. Transl. Eng. Health Med.* **2020**, *9*, 1–14. [CrossRef]

- 20. Gruen, A.; Akca, D. Least squares 3D surface and curve matching. *ISPRS J. Photogramm. Remote. Sens.* 2005, 59, 151–174. [CrossRef]
- 21. Zhang, Z. Iterative point matching for registration of free-form curves and surfaces. *Int. J. Comput. Vis.* **1994**, *13*, 119–152. [CrossRef]
- Cheng, J.; Zhang, L.; Chen, Q.; Hu, X.; Cai, J. A review of visual SLAM methods for autonomous driving vehicles. *Eng. Appl. Artif. Intell.* 2022, 114, 104992. [CrossRef]
- 23. Sequeira, V.; Ng, K.; Wolfart, E.; Gonçalves, J.G.M.; Hogg, D. Automated reconstruction of 3D models from real environments. *ISPRS J. Photogramm. Remote. Sens.* **1999**, 54, 1–22. [CrossRef]
- 24. Sorkine-Hornung, O.; Rabinovich, M. Least-squares rigid motion using svd. Computing 2017, 1, 1–5.
- 25. Lin, B.; Tamaki, T.; Zhao, F.; Raytchev, B.; Kaneda, K.; Ichii, K. Scale alignment of 3D point clouds with different scales. *Mach. Vis. Appl.* **2014**, *25*, 1989–2002. [CrossRef]
- Bolkas, D.; Walton, G.; Kromer, R.; Sichler, T. Registration of multi-platform point clouds using edge detection for rockfall monitoring. *ISPRS J. Photogramm. Remote. Sens.* 2021, 175, 366–385. [CrossRef]
- Shi, X.; Liu, T.; Han, X. Improved Iterative Closest Point (ICP) 3D point cloud registration algorithm based on point cloud filtering and adaptive fireworks for coarse registration. *Int. J. Remote. Sens.* 2020, *41*, 3197–3220. [CrossRef]
- Xu, G.; Pang, Y.; Bai, Z.; Wang, Y.; Lu, Z. A fast point clouds registration algorithm for laser scanners. *Appl. Sci.* 2021, 11, 3426. [CrossRef]
- Liu, H.; Zhang, Y.; Lei, L.; Xie, H.; Li, Y.; Sun, S. Hierarchical Optimization of 3D Point Cloud Registration. Sensors 2020, 20, 6999. [CrossRef]
- Rusu, R.B.; Blodow, N.; Marton, Z.C.; Beetz, M. Aligning point cloud views using persistent feature histograms. In Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, 22–26 September 2008; pp. 3384–3391.
- 31. Zou, X.; He, H.; Wu, Y.; Chen, Y.; Xu, M. Automatic 3D point cloud registration algorithm based on triangle similarity ratio consistency. *IET Image Process.* 2020, *14*, 3314–3323. [CrossRef]
- Zhan, X.; Cai, Y.; Li, H.; Li, Y.; He, P. A point cloud registration algorithm based on normal vector and particle swarm optimization. *Meas. Control.* 2020, 53, 265–275. [CrossRef]