



# Genetic Algorithm-Based Optimization for Color Point Cloud Registration

Dongsheng Liu<sup>1</sup>, Deyan Hong<sup>1\*</sup>, Siting Wang<sup>2</sup> and Yahui Chen<sup>3</sup>

<sup>1</sup>School of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou, China, <sup>2</sup>School of Management Engineering and E-commerce, Zhejiang Gongshang University, Hangzhou, China, <sup>3</sup>Zhejiang Gongshang University, Hangzhou, China

Point cloud registration is an important technique for 3D environment map construction. Traditional point cloud registration algorithms rely on color features or geometric features, which leave problems such as color affected by environmental lighting. This article introduced a color point cloud registration algorithm optimized by a genetic algorithm, which has good robustness for different lighting environments. We extracted the HSV color data from the point cloud color information and made the HSV distribution of the tangent plane continuous, and we used the genetic algorithm to optimize the point cloud color information consistently. The Gauss–Newton method was utilized to realize the optimal registration of color point clouds for the joint error function of color and geometry. The contribution of this study was that the genetic algorithm was used to optimize HSV color information of the point cloud and was applied to the point cloud registration algorithm, which reduces the influence of illumination on color information and improves the accuracy of registration. The experimental results showed that the square error of color information saturation and lightness optimized by the genetic algorithm was reduced by 14.07% and 37.16%, respectively. The color point cloud registration algorithm in this article was reduced by 12.53% on average compared with the optimal result algorithm RMSE.

## OPEN ACCESS

### Edited by:

Zhihua Cui,  
Taiyuan University of Science and  
Technology, China

### Reviewed by:

Shiyuan Han,  
University of Jinan, China  
Anfeng Liu,  
Central South University, China

### \*Correspondence:

Deyan Hong  
hongdeyan1997@gmail.com

### Specialty section:

This article was submitted to  
Bionics and Biomimetics,  
a section of the journal  
Frontiers in Bioengineering and  
Biotechnology

**Received:** 19 April 2022

**Accepted:** 13 May 2022

**Published:** 29 June 2022

### Citation:

Liu D, Hong D, Wang S and Chen Y  
(2022) Genetic Algorithm-Based  
Optimization for Color Point  
Cloud Registration.  
*Front. Bioeng. Biotechnol.* 10:923736.  
doi: 10.3389/fbioe.2022.923736

**Keywords:** genetic algorithm, point cloud, hsv, registration, optimization

## 1 INTRODUCTION

Point cloud registration is an important technique in the fields of pose estimation, three-dimensional positioning, and computer vision. It is commonly used in simultaneous localization and mapping (SLAM), three-dimensional reconstruction, unmanned driving, and automation applications. Robots use devices such as depth cameras to make three-dimensional maps of the environment to realize the perception of the environment. Due to the limitations of shooting equipment and physical occlusion, scanning devices can only scan part of an object or scene. Therefore, the object or scene is scanned in different angles and directions to obtain the complete point cloud data. The purpose of point cloud registration is to compute the rotation and translation matrices between multiple point clouds and finally merge them into complete point cloud data.

Point cloud registration relies on geometry or color information for the moment. Douadi et al. (2006) emphasized the influence of color information on improving the registration accuracy of high point cloud for the existing problems of ICP. Tong and Ying (2018) proposed a rough matching algorithm based on texture information and point cloud curvature characteristics. Park et al. (2017) proposed a way to constrain both geometric and color information. By using

the distribution of color on the tangent plane, a continuous color gradient was defined to represent the functional change of color with the location. A joint error function consisting of geometric and color errors was constructed, effectively constraining both geometric and color information. However, the color space used is the RGB color space, and then, the color information needs to be converted to gray scale information, which is easily affected by ambient light, resulting in large data fluctuations.

The aforementioned method combines color information with geometric information to improve the registration's effectiveness, but there is inconsistency in the original color data. When the depth camera collects data, the color information collected by the depth camera changes due to the inconsistency in scene illumination and the change in the acquisition of the angle. Therefore, Ren et al. (2021) converted the RGB color space to a hue color space to improve robustness under different lighting conditions. However, saturation and brightness information is ignored.

Since the saturation and brightness of images are greatly different from hues owing to the influence of ambient lighting, it is difficult to effectively apply the saturation and brightness information to point cloud registration. With the help of previous researchers, this article was based on the study of two-dimensional image illumination uniformity; the global brightness transformation can effectively reduce the differences in brightness. The saturation feedback algorithm can effectively change the image of saturation, but this kind of algorithm relies heavily on parameter setting; setting the same parameters is not effective for all situations. The genetic algorithm has the abilities of fast random search, simple process, easy to integrate with other algorithms, can effectively haphazard search for appropriate parameters, and reduce the impact of the environment on color information.

This article proposed a color point cloud registration algorithm optimized by a genetic algorithm to solve the aforementioned problems in point cloud registration. In this algorithm, the global luminance of color information is logarithmic transformation, and the HSV (hexcone model) color information is optimized by using a genetic algorithm. According to the normal vector and the surface curvature of the point, the matching correlation points of the point cloud are calculated. Finally, the error function composed of geometric error and color error is constructed. Experiments showed that the algorithm proposed in this article can also realize accurate point cloud registration with the help of color information and geometric information in the case of geometric degradation and complex scenes, which proves that the algorithm can improve the accuracy of the color point cloud registration.

The contributions of this article are as follows:

- 1) Aiming at the fact that the saturation and brightness of the HSV color space are greatly affected by illumination, the global brightness transformation and saturation feedback algorithm of the 2D image is applied to 3D point cloud registration.
- 2) We constructed a fitness index to evaluate saturation information, optimized the consistency of point cloud color information by using a genetic algorithm, and effectively reduced the influence of environmental illumination on point cloud registration.

The remaining article is organized as follows. The *Related Work* section discusses some work of the point cloud registration in recent years, and the *Algorithm Implementation* section shows the specific algorithm steps. The *Experimental Result and Analysis* section shows the experimental results and analysis, and the *Conclusion* summarizes the article and the future research direction.

## 2 RELATED WORK

Point cloud registration has been extensively investigated in history. The point cloud registration method mainly includes an optimization-based registration algorithm, feature extraction-based registration algorithm, and end-to-end registration algorithm (Huang et al., 2021). The most classical point cloud registration algorithm based on optimization is the ICP algorithm (Besl and McKay, 1992). ICP achieves registration of target point clouds by finding the nearest point neighbor in space as a matching pair and calculating a three-dimensional rigid transformation using matching points. However, the challenging assignment corresponding to the nearest point based on spatial distance is sensitive to the initial rigid transformation and outliers. This often leads to ICP convergence to the wrong local minimum. To solve the problems being completed in ICP, many studies have put forward improvement schemes for the shortcomings of the original ICP. Rusinkiewicz and Levoy (2001) considered the distance between the vertex and the target vertex and proposed a point-to-surface ICP algorithm, which made the iteration process tough to fall into local optimum. Grant et al. (2019) proposed a surface-to-surface ICP algorithm, which not only considered the local structure of the target point cloud but also the local structure of the source point cloud. Segal et al. (2009) proposed a generalized ICP (GICP). It combines the point-to-point and surface-to-surface strategies to improve accuracy and robustness. Serafin and Grisetti (2015) added the curvature constraints of the normal vector and point cloud to point cloud registration and considered the distance from the point to the tangent and the angle difference of the normal vector, which improved the accuracy of the registration. But these algorithms only constrain the geometrical information of the point cloud and do not make effective use of color information.

With the popularization of 3D lidar and depth cameras, depth maps and color map data can be obtained simultaneously with corresponding equipment. Therefore, many articles have been proposed to improve the accuracy of the point cloud registration by combining depth map and color map data. Traditional point cloud registration methods based on color information extend the color

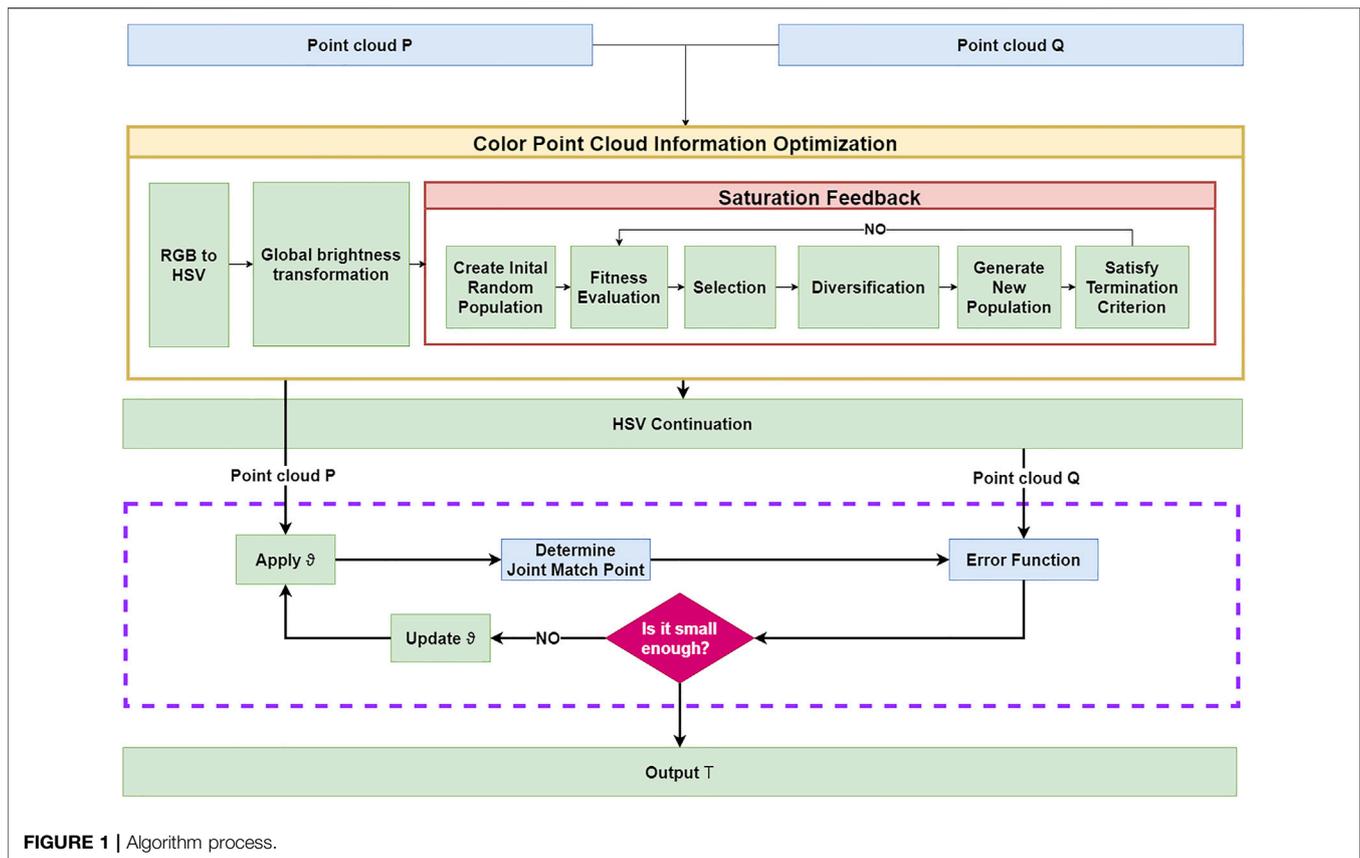


FIGURE 1 | Algorithm process.

information vector from three-dimensional to a higher dimension (Besl and McKay, 1992; Men et al., 2011; Korn et al., 2014; Jia et al., 2016). Jia et al. (2016) improved homogeneity and registration accuracy by converting the RGB color space into the LAB color space and combining the NICP algorithm. Ren et al. (2021) reduced the impact of ambient brightness changes by converting the RGB color space into hue color space.

In addition, some research practices construct a point cloud registration algorithm based on feature extraction. Hu et al. (2021) used the scale-invariant feature transform (SIFT) algorithm to extract scale-invariant features from the 2D gray image, which improves the matching accuracy. Ran and Xu (2020) proposed a point cloud registration method that integrates sift and geometric features. The feature points are produced by SIFT, and incorrect matching points are evaluated by curvature.

In addition to the traditional point cloud registration algorithm mentioned previously, with the progress of deep learning in recent years, many researchers have proposed the registration algorithm based on deep learning (Guo et al., 2020; Kurobe et al., 2020; Huang et al., 2021). The current depth-based registration algorithms are mainly divided into feature-based learning point cloud registration algorithms and end-to-end-based point cloud registration algorithms. The feature-based point cloud registration algorithm mainly evaluates features through input point cloud and deep network, then

solves the rotation translation matrix, and outputs the results. In recent years, DCP (deep closest point) (Wang and Solomon, 2019), RPMNet (Yew and Lee, 2020), AlignNet (Groß et al., 2019), and other models have been developed. The main approach of the point cloud registration algorithm based on end-to-end is to input two pieces of the point cloud. Generally, the neural network fitting regression model is used to estimate the rotation translation matrix or to combine the neural network with optimization. In recent years, DeepGMR (Yuan et al., 2020) and 3D RegNet (Pais et al., 2020) have been used as models. The main advantage of a registration algorithm based on deep learning is that it can combine the advantages of a traditional mathematical theory and neural network well. But calculating the in-depth learning model at the same time requires a lot of training data and often only works well for scenarios contained in the training set. When facing untrained scenarios, the accuracy will be greatly reduced. In addition, there is a high requirement for training effort (Guo et al., 2020).

### 3 ALGORITHM IMPLEMENTATION

In this article, a color point cloud registration algorithm based on genetic image enhancement and geometric features is designed. The algorithm flow is shown in Figure 1. The algorithm consists of four steps. Step 1: the global

luminance logarithmic transformation should be performed on the color information, and then, the HSV color information should be optimized based on a genetic algorithm to calculate the color information gradient. Step 2: the association points of the joint image features and geometric features query point cloud should be extracted, and the association points that do not meet the conditions should be filtered. Step 3: the error function combining the geometric error and color error should be constructed. Step 4: The Gauss–Newton method should be used to solve the point cloud iteratively.

### 3.1 Color Point Cloud Information Optimization

In general, the original image acquired by the depth camera is a RGB image. The RGB color space is the most frequently used color space mode. It composes of three primary colors: red, green, and blue. The range of red, green, and blue is between 0 and 255, which is widely used in the field of computer vision. However, the uniformity of the RGB color space is insignificant; the color difference cannot be judged directly by spatial distance, and the brightness cannot be handled well. At the same time, due to the influence of ambient light brightness, images from different angles lead to coloring deviation.

To solve the aforementioned problem, this article puts forward after a genetic algorithm to optimize the HSV color information with the introduction of point cloud registration methods; this article is first to study the global image brightness logarithmic transformation and complete the dark areas in the image enhancement and dynamic range compression. To further improve the optimization of HSV color information, this article used a genetic algorithm to dynamically calculate the saturation optimization coefficient and to improve the stability of saturation under different lighting conditions.

The global logarithmic brightness transformation is applicable to the original image. The global logarithmic brightness transformation formula is as follows:

$$V'(x, y) = \ln(V(x, y) + 1), \quad (1)$$

where  $V'(x, y)$  is the logarithmic brightness, and  $V(x, y)$  is the original brightness at the point  $(x, y)$ . To reduce the influence of illumination on brightness, this article first performed a global logarithmic brightness transformation on the original image. Setting the red, green, and blue channels in each original RGB image as  $R_e$ ,  $G_e$ ,  $B_e$ , and  $C_{\max}$  as the largest of the three channels, the global logarithmic brightness transformation formula is as follows:

$$R_e' = R_e/255, G_e' = G_e/255, B_e' = B_e/255; \quad (2)$$

$$C_{\max} = \max(R_e', G_e', B_e'), C_{\min} = \min(R_e', G_e', B_e'), \Delta = C_{\max} - C_{\min}; \quad (3)$$

where  $R_e'$ ,  $G_e'$ , and  $B_e'$ , respectively, represent the normalized results of  $R_e$ ,  $G_e$ , and  $B_e$  in corresponding places;  $C_{\min}$  is the minimum value of  $R'$ ,  $G'$ , and  $B'$ ; and  $\Delta$  is the difference between the maximum value and minimum value of  $R_e'$ ,  $G_e'$ , and  $B_e'$ .

$$H = \begin{cases} \Delta = 0 & 0^\circ; \\ \frac{60^\circ \times \left( \frac{G_e' - B_e'}{\Delta} \bmod 6 \right)}{360} & , C_{\max}' = R_e'; \\ \frac{60^\circ \times \left( \frac{B_e' - R_e'}{\Delta} + 2 \right)}{360} & , C_{\max}' = G_e'; \\ \frac{60^\circ \times \left( \frac{R_e' - G_e'}{\Delta} + 4 \right)}{360} & , C_{\max}' = B_e'; \end{cases} \quad (4)$$

$$S = \begin{cases} 0 & , C_{\max} = 0; \\ \frac{\Delta}{C_{\max}} & , C_{\max} \neq 0; \end{cases} \quad (5)$$

$$V = C_{\max}. \quad (6)$$

$V \in [0, 1]$ ,  $S \in [0, 1]$ , and  $H \in [0, 1)$ . because the global logarithmic brightness transformation only enhances the brightness while the saturation remains essentially the same. Therefore, to optimize saturation effectively, this article is based on a genetic algorithm and feedback saturation enhancement algorithm to enhance the saturation layer (Strickland et al., 1987).

The genetic algorithm simulates the survival of the fittest in nature and uses selection, crossover, and variation to solve a satisfactory solution. According to the saturation feedback formula, this article modified it as follows:

$$S^*(x, y) = k_s [S'(x, y) - \bar{S}'(x, y)] + k_v [V'(x, y) - \bar{V}'(x, y)] + S(x, y), \quad (7)$$

where  $S^*(x, y)$  is the enhanced saturation value,  $S'(x, y) = 1 - S(x, y)$ , and  $S(x, y)$  is the saturation at the point  $(x, y)$ ;  $\bar{S}'(x, y)$  is the local saturation mean at the point  $(x, y)$ ,  $V'(x, y) = 1 - V(x, y)$ , and  $\bar{V}'(x, y)$  is the local saturation mean at the point  $(x, y)$ . Because the coefficients of  $k_s$  and  $k_v$  are very important for saturation processing, the manual setting method cannot effectively meet the actual need. For this reason, this article used a genetic algorithm to calculate the saturation coefficient dynamically so that it can adjust the values of  $k_s$  and  $k_v$  adaptively.

The genetic algorithm in this article is divided into four steps; the specific steps are as follows:

- 1) Confirming the the encoding and fitness functions: float number encoding is employed in this article.  $k_s$  and  $k_v$  are the coefficients to be optimized, and their value ranges are 0–4. To test and evaluate the enhancement effect, we defined a fitness function, and it is constructed as follows:

$$f_{fitness} = \exp\left(\left[\frac{1}{M \times N} \sum_{x=1}^M \sum_{y=1}^N S(x, y)\right]^2 - \frac{1}{M \times N} \sum_{x=1}^M \sum_{y=1}^N S(x, y)^2\right), \quad (8)$$

For an image of  $M \times N$  size, the smaller the  $f_{fitness}$  value, the larger is the average change of image saturation.

- 2) Identifying the selection strategy and the genetic operator: to ensure the optimal individuals can be obtained, this article used the sorting selection strategy, as shown in Eq. 9. In addition, in order to maintain the diversity of the group, it is not permitted to choose the same parent.

$$P(S^*(x, y)_i) = \frac{f_{fitness}(S^*(x, y)_i)}{\sum_{j=1}^N f_{fitness}(S^*(x, y)_j)} \quad (9)$$

- 3) Determination of the control parameters of the genetic algorithm: it mainly includes population size, variation rate, and crossover rate. If the size of the population is too small and lacks diversity, it can easily lead to regional convergence. A large population size makes the algorithm converge slowly, which influences the processing speed. Therefore, the size of the population should be large enough. In the experiment, we set the population size at 30 and the crossover rate and variation rate at 0.9 and 0.05, respectively.
- 4) Determination of downtime criteria: the calculation is aborted based on the maximum number of iterations and the rate of change between the average fitness value of the current population and that of the previous generation. The maximum number of iterations is set to 100, and the change rate of the average fitness is 0.2%. When the number of iterations is greater than 100 or the change rate of average fitness is less than 0.2%, the calculation is aborted.

In order to optimize the error of color information, it is necessary to obtain the corresponding color information gradient. Therefore,  $H(\mathbf{p})$  needs to be converted into the continuous color function  $H_p(v)$ , where  $v$  is the tangent plane vector, and  $N_p$  is the neighborhood of  $\mathbf{p}$ ;  $\mathbf{p}'$  is a point in  $N_p$ , and  $\mathbf{n}_p$  is the normal of the point  $\mathbf{p}$ , making  $v^* \mathbf{n}_p = 0$ . We assumed that  $H_p(v)$  is a continuous function and represented the distribution of HSV on the tangent plane; then it can define  $H_p(v)$  as follows:

$$H_p(v) \approx H(\mathbf{p}) + d_p^T v, \mathbf{p} \in P. \quad (10)$$

For each point  $\mathbf{p}$ ,  $f(s)$  is the plane function projected by point  $P$  to the section plane:

$$f(s) = s - \mathbf{n}_p (s - \mathbf{p})^T \mathbf{n}_p. \quad (11)$$

The least square fitting objective of  $d_p$  is as follows:

$$\begin{aligned} L(d_p) &= \sum_{\mathbf{p}' \in N_p} (H_p(f(\mathbf{p}') - \mathbf{p}) - H(\mathbf{p}'))^2 \\ &\approx \sum_{\mathbf{p}' \in N_p} (H(\mathbf{p}) + d_p^T (f(\mathbf{p}') - \mathbf{p}) - H(\mathbf{p}'))^2. \end{aligned} \quad (12)$$

There is a constraint that requires  $d_p^T \mathbf{n}_p = 0$ , which is solved by using the Lagrange multiplier method in this article.

### 3.2 Point Cloud-Associated Points

The traditional way to find the associated points of the point cloud usually only depends on the geometric information to find the matching points and does not make full use of the color

information. Due to the scale invariance, rotation invariance, and angle invariance of sphere features, the same feature information can still be maintained at different angles and distances. At the same time, the extraction speed of orb features is significantly improved compared with SIFT (scale-invariant feature transform) and surf (speed up robot features) (Rublee et al., 2011). Therefore, in order to effectively utilize the feature and geometric information of color images, the algorithm uses the orb feature as the image feature extraction algorithm and combines the KD-tree nearest neighbor algorithm with the set of matching points found by the orb feature algorithm.

We set the orb feature point set as  $\mathcal{K}_{orb}$  and the geometric point set as  $\mathcal{K}_{kd} = \{(\mathbf{p}_i, \mathbf{q}_j)\}, \mathbf{p}_i \in P, \mathbf{q}_j \in Q$  feature point set.

$$\mathcal{K} = \mathcal{K}_{orb} \cup \mathcal{K}_{kd}. \quad (13)$$

KD-Tree is used to search for  $\mathbf{p}$  radius, and the search parameter is set at 0.04. Center  $\mu_i^S$  of all points centered on  $\mathbf{p}_i$ , and the covariance  $\Sigma_i^S$  of the Gaussian distribution is calculated.

$$\mu_i^S = \frac{1}{|\mathcal{V}_i|} \sum_{\mathbf{p}_i \in \mathcal{V}_i} \mathbf{p}_i, \quad (14)$$

$$\Sigma_i^S = \frac{1}{|\mathcal{V}_i|} \sum_{\mathbf{p}_i \in \mathcal{V}_i} (\mathbf{p}_i - \mu_i) (\mathbf{p}_i - \mu_i)^T, \quad (15)$$

where  $\mathcal{V}_i$  is the set of adjacent points of point  $\mathbf{p}_i$ , and  $\mu_i$  is the center of  $\mathcal{V}_i$ . By matrix decomposition of  $\Sigma_i^S$ , the eigenvalues  $\lambda_1$ ,  $\lambda_2$ , and  $\lambda_3$  of  $\Sigma_i^S$  are obtained. In this article,  $\sigma_i$  is used to represent the curvature of the current point to evaluate the degree of coincidence between planes. The smaller  $\sigma_i$  is, the flatter is the plane.

$$\sigma_i = \lambda_1 / (\lambda_1 + \lambda_2 + \lambda_3). \quad (16)$$

Detail curvature filtering principle in the NICP algorithm is used to filter the matching points (Serafin and Grisetti, 2015).

### 3.3 Error Function

The traditional error function only considers the geometric information. When similar geometric structures occur, it is not difficult to fall into the local optimal solution. In view of the aforementioned problems, the error function set out in this article not only considers the geometric information but also the color information.  $P$  and  $Q$  are color point clouds, respectively, and  $T$  is the initial alignment matrix. The goal is to determine the optimal solution  $T$  from  $P$  and  $Q$ .

The following error functions  $E(T)$  are defined in this article:

$$E(T) = (1 - \lambda)E_H(T) + \lambda E_G(T). \quad (17)$$

$E_H(T)$  and  $E_G(T)$  correspond to the color error and geometric error, respectively. They have the following relationship:

$$q^\dagger = R(w)q + t, \quad (18)$$

$$q^p = f(q^\dagger), \quad (19)$$

where  $R(w) \in R^{3 \times 3}$  is the rotation matrix of  $w$ , and  $t$  is the displacement vector.  $q^\dagger$  is the new point after  $q$  rotation and translation transformation.  $q^p$  is the projection of  $q^\dagger$  on the

tangent plane; then, the HSV color error  $E_H(T)$  is defined as follows:

$$E_H(T) = \sum_{(p,q) \in \mathcal{K}} (H_p(q^p) - H_p(q))^2. \quad (20)$$

The geometric error  $E_G$  is defined as the tangent plane distance between  $q^t$  and  $p$ ,  $\mathbf{n}_p$  is the normal of the point  $p$ ; then, the geometric error is as follows:

$$E_G(T) = \sum_{(p,q) \in \mathcal{K}} ((q^t - p)^T \mathbf{n}_p)^2. \quad (21)$$

$E_H$  and  $E_G$  are combined to construct a complete error function:

$$E(T) = (1 - \lambda) \sum_{(p,q) \in \mathcal{K}} (e_H^{(p,q)}(T))^2 + \lambda \sum_{(p,q) \in \mathcal{K}} (e_G^{(p,q)}(T))^2, \quad (22)$$

where  $e_H^{(p,q)}$  and  $e_G^{(p,q)}$  represent the residuals of color information and geometric information, respectively:

$$e_H^{(p,q)}(T) = H_p(q^p) - H_p(q). \quad (23)$$

$$e_G^{(p,q)}(T) = (q^t - p)^T \mathbf{n}_p. \quad (24)$$

### 3.4 Point Cloud Registration Iteration

The Gauss–Newton method is used to solve the minimum error of  $E(T)$ . In order to facilitate the algorithm calculation, we defined six-dimensional vector  $\vartheta$  to represent  $T$ , and its structure is as follows:

$$\vartheta = (\Delta\delta, \Delta\epsilon, \Delta\epsilon, \Delta a, \Delta b, \Delta c). \quad (25)$$

Since  $E(T)$  is composed of  $E_H(T)$  and  $E_G(T)$ , the Jacobian matrix also needs to be composed of  $J_{e_H}$  and  $J_{e_G}$ :

$$J_e = \begin{bmatrix} \sqrt{1-\lambda} J_{e_H} \\ \sqrt{\lambda} J_{e_G} \end{bmatrix}. \quad (26)$$

$\vartheta^k$  represents the transformation vector after  $k$  iterations, and then, the Jacobian matrix calculation of  $\vartheta^k$  is as follows:

$$\begin{aligned} J_{e_H} &= \nabla e_H^{(p,q)}(\vartheta) \Big|_{\vartheta=\vartheta^k}, \\ J_{e_G} &= \nabla e_G^{(p,q)}(\vartheta) \Big|_{\vartheta=\vartheta^k}. \end{aligned} \quad (27)$$

According to the chain rule,  $\nabla e_H^{(p,q)}(\vartheta)$  and  $e_G^{(p,q)}(\vartheta)$  can be calculated by the following formula:

$$\nabla e_H^{(p,q)}(\vartheta) = \left[ \frac{\partial}{\partial \vartheta_i} (q^p - p) - H(q) \right] = \nabla H_p(f) J_f(s) J_s(\theta), \quad (28)$$

$$\nabla e_G^{(p,q)}(\vartheta) = \left[ \frac{\partial}{\partial \vartheta_i} (q^t - p)^T \mathbf{n}_p \right] = \mathbf{n}_p^T J_s(\theta), \quad (29)$$

where  $\nabla H_p(f) = d_p$ ,  $J_f(s)$  is the Jacobian of  $f$  with respect to  $s$ , and  $J_s(\theta)$  is the Jacobian with respect to  $\theta$ . The Gauss–Newton method

is used for optimization iteration of  $\vartheta$ , and the specific update steps are as follows:

$$\vartheta^{k+1} = \vartheta^k - (J_e^T J_e)^{-1} J_e^T e, \quad (30)$$

where  $e = \begin{bmatrix} \sqrt{1-\lambda} e_H \\ \sqrt{\lambda} J_{e_G} e_G \end{bmatrix}$ , when the value of error function is less than the threshold value through repeated iterative optimization of  $\vartheta$ , the iteration is stopped, and the final result is the output.

## 4 EXPERIMENTAL RESULT AND ANALYSIS

### 4.1 Experimental Environment Configuration

We evaluated the effectiveness of the color point cloud matching algorithm based on the genetic algorithm. RawFoot (Cusano et al., 2015) and TUM datasets (Sturm et al., 2012) were used for verification, respectively. From the corn sample set in the RawFoot dataset, 64 images with different light intensities and angles are selected and used to assess color consistency. In the depth and color images, four datasets were extracted from the tums with clear 254 overlaps. It can be further used to verify the rationality and effectiveness of the registration algorithm. The cloud registration algorithm in this experiment was run on a desktop computer equipped with Ubuntu18.04 LTS, AMD Ryzen 9 5950 × 3.4ghz CPU, 64G DDR4 memory, and GTX 3070 Ti graphics card. Using Open3D (Zhou et al., 2018) and OpenCV (Bradski, 2000) and other tools and running the corresponding C++ algorithm program (Xu et al., 2021), the experimental parameters are set as follows: the maximum iteration time was set to 50, error function threshold to  $10^{-6}$ ,  $\epsilon_d$  to 0.5,  $\epsilon_\sigma$  to 1.3,  $\epsilon_n$  to 0.523, Kd-Tree search radius to 0.08, orb feature number to 500, orb pyramid layer number to 8, and orb pyramid extraction ratio to 1.2. The oriented BRIEF descriptor size was set to 31.

### 4.2 Color Information Optimization

To verify the optimization performance of the genetic algorithm for HSV color information, this article used the RawFoot as test datasets. The RawFoot database is designed specifically for the descriptor; and the classification method in robustness on light condition changes and pays special attention to the change in the light source color; the light conditions in the direction of the light, light color, and brightness on the combination of these factors are different.

The sample set of corn is selected from RawFoot as test data. The sample set of corn is composed of 46 pictures of different light sources and angles, as shown in Figure 2. D40 is regarded as the reference image, and the original RGB, original HSV, and enhanced HSV data are compared to verify the effectiveness of image enhancement. Meanwhile, in order to effectively evaluate the consistency of the image enhancement algorithm in different lighting and light angles, the sum square distance SSD algorithm is used to calculate the degree of difference between the image under different lighting and the benchmark images:



**FIGURE 2** | Image of the original corn sample set.

$$SSD(O, E) = \frac{1}{N} \sum_{i=1}^N (C(o_i) - C(e_i))^2. \quad (31)$$

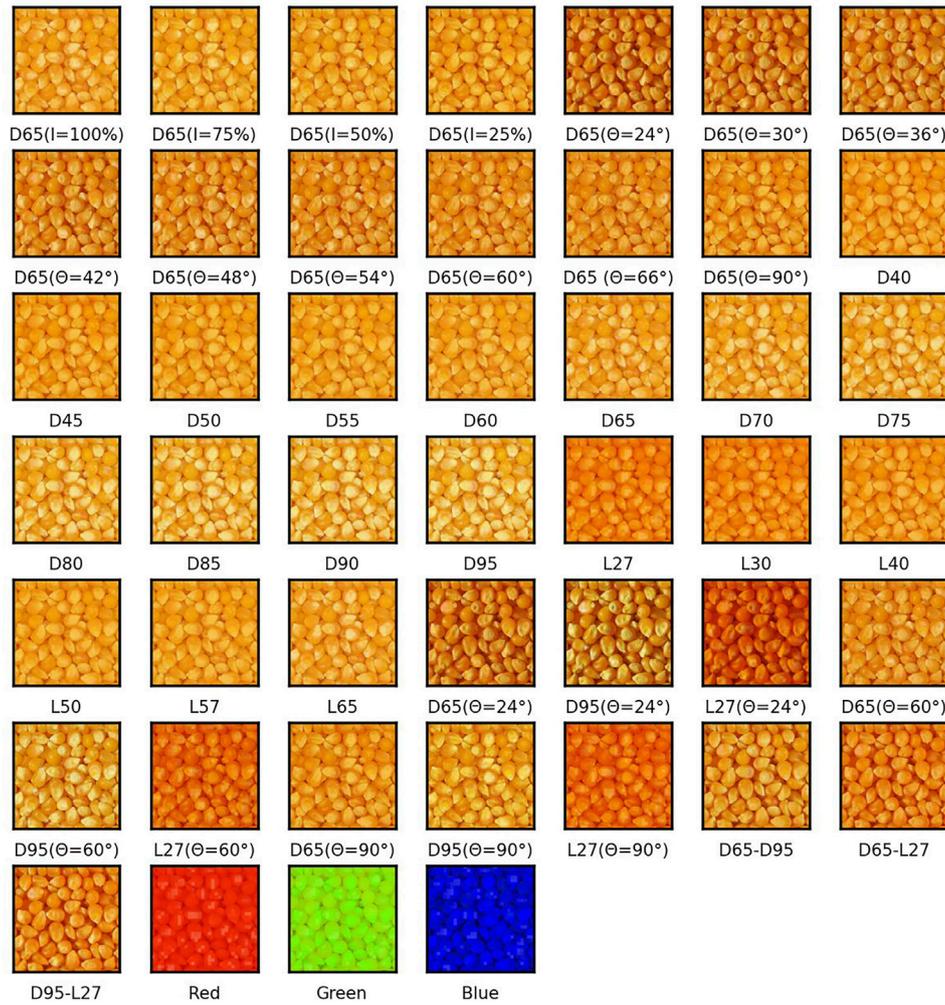
$O$  stands for reference pictures, on behalf of the need to compute the degree of the difference image,  $o_i$  and  $e_i$  represent  $O$  and  $E$  with a single channel of pixels, where the function represents a pixel value on a single channel of the picture. The smaller the pixel value difference between the basic picture and different lighting conditions, the closer the SSD value is. It represents the calculated value under different lighting conditions, which is closer to the original value of the reference picture.

**Figure 2** shows the image of the corn sample set in the original RawFoot dataset. Due to light angle and brightness, there are obvious differences between the images. **Figure 3** is the result of the image enhancement algorithm. After the algorithm processing, the difference between images is significantly reduced, and the details in the dark area are more noticeable due to the improvement of brightness and contrast.

As shown in **Figure 4**, **Figures 4A–C** are the comparison diagrams of the original hue, saturation, and value data and the hue, saturation, and value data after enhancement; **Figures 4D–F** are the differences of the original R, G, and B channels, respectively. Compared with the unenhanced HSV data, the average SSD index of saturation and value after image enhancement decreased by 14.07% and 37.16% compared with the original saturation and value. Experimental results showed that the influence of illumination and light angle on saturation and value is decreased after the image enhancement algorithm.

### 4.3 Color Point Cloud Registration

The TUM dataset is a well-known RGB-D dataset, which provides rich scenes. The Kinect camera is used in the TUM dataset to collect depth information and color information, among which the resolution of the depth image and a color image is 640 \* 480. In this article, `rgbd_dataset_freiburg1_xyz`, `rgbd_dataset_freiburg1_teddy`, `rgbd_dataset_freiburg1_360` and



**FIGURE 3** | Image of the corn sample set after enhancement.

rgb\_dataset\_freiburg1\_plant were selected in the TUM dataset, respectively. Partial scenes were selected from four datasets of the plant for testing; ICP and NICP algorithms (Park et al., 2017) (referred to as Color 3D ICP) and the method proposed in this article were used to register overlapping color point clouds, respectively. In order to effectively evaluate other algorithms and the accuracy of the algorithm proposed in this article, we used the root mean square error (RMSE) index and the fitness index, for a given two pieces of point cloud  $(p, \hat{p})$  and their matching points  $(p_i, \hat{p}_i) \in \mathcal{K}_{p\hat{p}}$ , pit for after the rigid body transformation, including  $\|p_i^t - \hat{p}_i\|_2^2$ , a representative and the distance between the  $p_i^t$  and  $\hat{p}_i$ ; RMSE metrics are defined as follows:

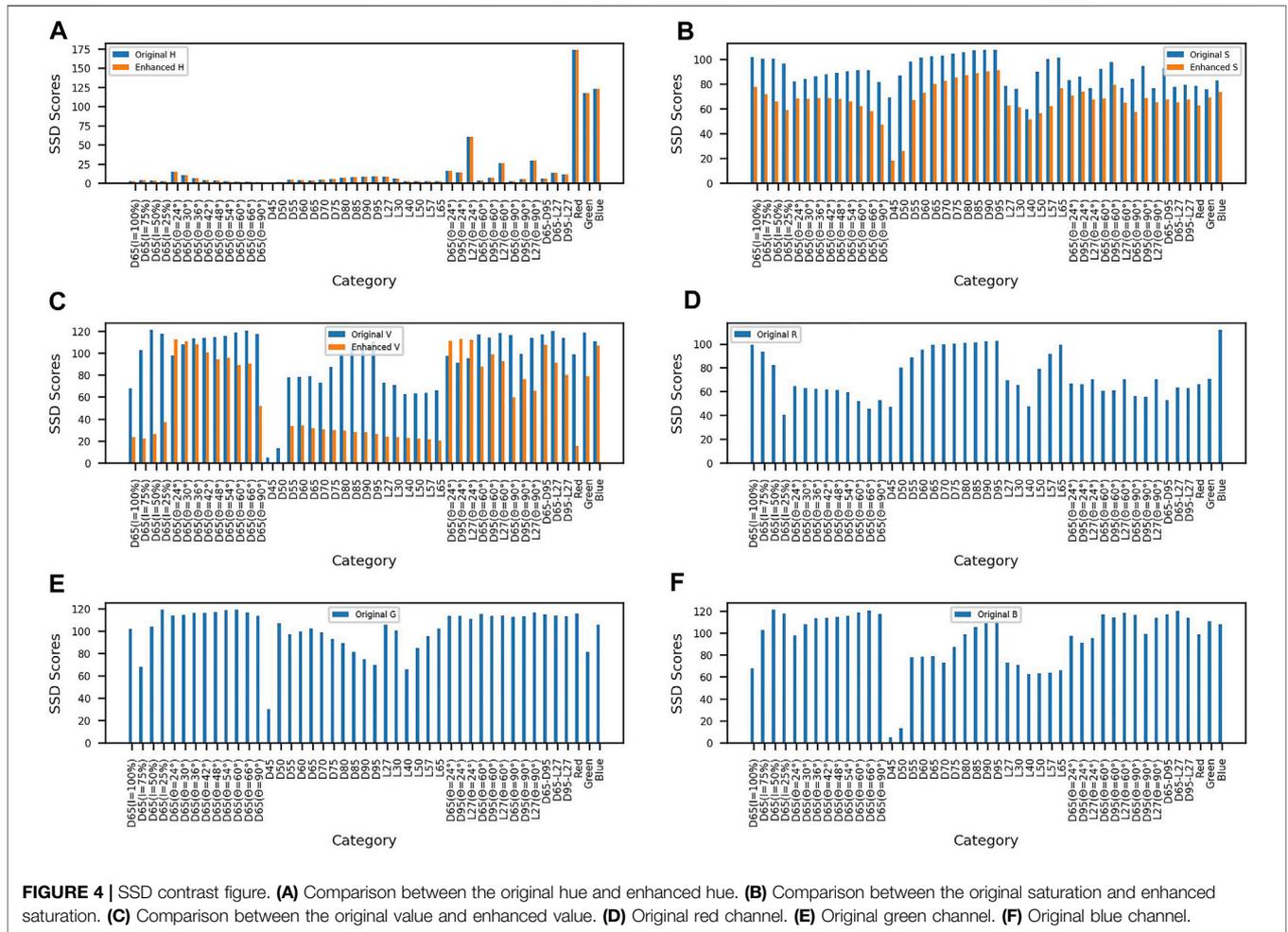
$$\text{RMSE}(p, \hat{p}) = \sqrt{\frac{1}{n} \sum_{i=1}^n \|p_i^t - \hat{p}_i\|_2^2}, p_i^t = R(w)p_i + t_i. \quad (32)$$

In order to reflect the relationship between the number of matching points and the number of source point clouds, fitness indicators are defined as follows:

$$\text{Fitness}(p, \mathcal{K}_{p\hat{p}}) = \frac{|\mathcal{K}_{p\hat{p}}|}{|p|}. \quad (33)$$

In this article, rgb\_dataset\_freiburg1\_xyz, rgb\_dataset\_freiburg1\_teddy, rgb\_dataset\_freiburg1\_360 and rgb\_dataset\_freiburg1\_plant were selected in the TUM dataset, respectively. Some scenes were selected from the four datasets of the plant for testing. ICP, NICP, Color 3D ICP, and the method proposed in this article were used to register the overlapping color point clouds and calculate the corresponding RMSE and fitness indicators.

**Tables 1, 2**, respectively, show RMSE and fitness index test results of ICP, NICP, Color 3D ICP, and our algorithm, and the best results of these algorithms are shown in bold. It can be seen from the data in **Table 1** that the RMSE index of the algorithm is



**TABLE 1 |** RMSE index comparison table.

Dataset	ICP	NICP	Color 3D ICP	Our
rgbd_dataset_freiburg1_xyz	0.0225	0.0223	0.0253	<b>0.0185</b>
rgbd_dataset_freiburg1_teddy	0.0498	0.0731	0.0380	<b>0.0369</b>
rgbd_dataset_freiburg1_360	0.0469	0.0330	0.0447	<b>0.0326</b>
rgbd_dataset_freiburg1_plant	0.0483	0.0379	0.0324	<b>0.0321</b>
Average	0.0426	0.0494	0.0351	<b>0.0300</b>

*Bold values represents the best result of these algorithms.*

**TABLE 2 |** Fitness index comparison table.

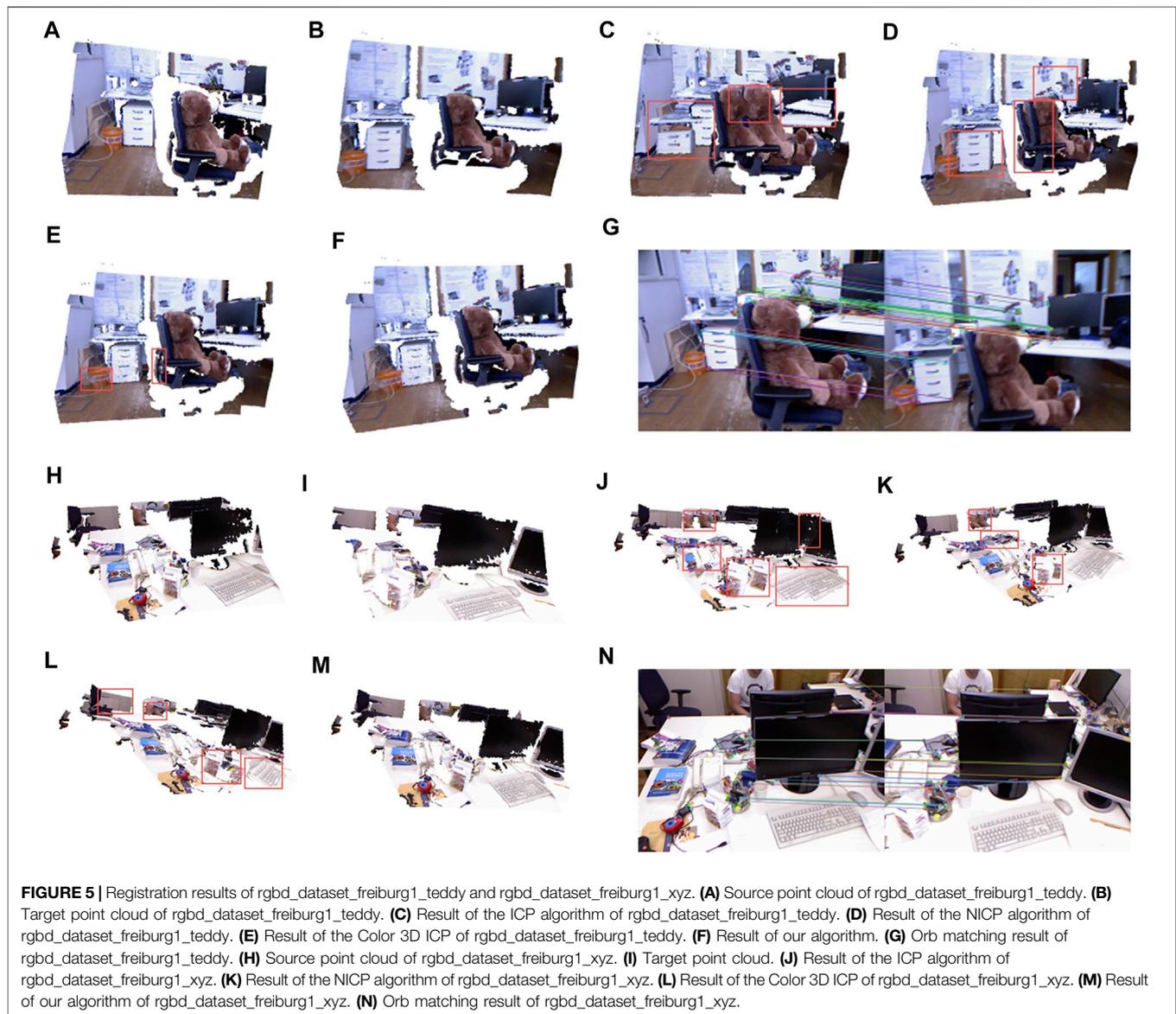
Dataset	ICP	NICP	Color 3D ICP	Our
rgbd_dataset_freiburg1_xyz	0.3804	0.3625	0.3058	<b>0.6742</b>
rgbd_dataset_freiburg1_teddy	0.2804	0.1922	0.7393	<b>0.7421</b>
rgbd_dataset_freiburg1_360	0.2967	0.3453	0.3653	<b>0.7121</b>
rgbd_dataset_freiburg1_plant	0.5116	0.3747	<b>0.9009</b>	0.9004
Average	0.3672	0.3186	0.5778	<b>0.7572</b>

*Bold values represents the best result of these algorithms.*

almost lower than that of the aforementioned algorithm, and the average RMSE is also the lowest. Meanwhile, it can be seen from

Table 2 that fitness is significantly improved compared with the aforementioned algorithm, indicating that the number of feature points selected by the algorithm accounts for a large proportion of the original point cloud.

Figures 5A,B,H, and I, are the adjacent original point clouds and target point clouds randomly selected in the rgbd\_dataset\_freiburg1\_teddy and rgbd\_dataset\_freiburg1\_xyz data sets, respectively. Figures 5C,J are the ICP algorithm, as shown in Figure 5C. The algorithm does not fit the point cloud well and has a serious dislocation. Figures 5D,K are the NICP algorithm, which restricts the geometric structure of the point cloud by restricting the normal and curvature of the point cloud, making the algorithm better for plane registration, but the registration effect for complex objects is not ideal. Figures 5E, L show the color 3D ICP registration of point clouds by combining the color information and geometric information. However, due to the incorrect association between matching points, registration errors still exist. Figures 5F, M, N show the algorithm in this article. The algorithm in this article not only constrained the color information and geometric information but also calculated the curvatures and normals of matching points and filtered the matching points that did not



meet the requirements. The results demonstrated that the algorithm in this article correctly registered the original point cloud and the target point cloud well. **Figures 5G, N** are orb feature point matching, and the results showed that orb feature matching for two-color images also has a good matching effect.

## 5 CONCLUSION

Point cloud acquisition is easy to be affected by the shooting environment. Changing the light will lead to the deviation of the brightness and color of the shooting object, which will affect the accuracy of the point cloud registration. Based on the aforementioned problems, a genetic algorithm is proposed to pre-process the color information of the point cloud. It can

further eliminate the inconsistency of brightness and color caused by different lights and reduce the impact on the point cloud registration.

In this article, using the logarithmic transformation of two-dimensional image brightness and the saturation feedback formula optimized by the genetic algorithm, the algorithm is transplanted to three-dimensional point cloud illumination processing to study the sensitivity of three-dimensional point cloud registration to illumination. Its innovation lies in the combination of the saturation feedback formula and genetic algorithm to optimize the color consistency of the point cloud color information, reducing the interference of illumination factors to a certain extent.

The experiment shows that the HSV data after bionic image enhancement decrease 14.07% and 37.16% on average compared

with the SSD indexes of original saturation and brightness. In addition, during the experiment, it is found that the input color image may be blurred due to jitter during camera acquisition, which will lead to blur in the registration process. Therefore, how to reduce the influence of color image blur on the accuracy of color point cloud registration is a problem that this article needs to study in the future.

## DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. These data can be found at: <https://vision.in.tum.de/data/datasets/rgbd-dataset/download> <http://projects.ivl.disco.unimib.it/minisites/rawfoot/>.

## REFERENCES

- Besl, P. J., and McKay, N. D. (1992). "Method for Registration of 3-d Shapes," in *Sensor Fusion IV: Control Paradigms and Data Structures (International Society for Optics and Photonics)*, 1611, 586–606.
- Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's J. Softw. Tools* 25 (11), 120–123.
- Cusano, C., Napoletano, P., and Schettini, R. (2015). "Local Angular Patterns for Color Texture Classification," in *International Conference on Image Analysis and Processing (Springer)*, 111–118. doi:10.1007/978-3-319-23222-5\_14
- Douadi, L., Aldon, M.-J., and Crosnier, A. (2006). "Pair-wise Registration of 3d/color Data Sets with Icp," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IEEE)*, 663–668. doi:10.1109/iroso.2006.282551
- Grant, W. S., Voorhies, R. C., and Itti, L. (2019). Efficient Velodyne Slam with Point and Plane Features. *Auton. Robot.* 43, 1207–1224. doi:10.1007/s10514-018-9794-6
- Groß, J., Ošep, A., and Leibe, B. (2019). "Alignnet-3d: Fast Point Cloud Registration of Partially Observed Objects," in *2019 International Conference on 3D Vision (3DV) (IEEE)*, 623–632. doi:10.1109/3dv.2019.00074
- Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., and Bennamoun, M. (2020). "Deep Learning for 3d Point Clouds: A Survey," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Hu, Q., Niu, J., Wang, Z., and Wang, S. (2021). Improved Point Cloud Registration with Scale Invariant Feature Extracted. *J. Russ. Laser Res.* 42, 219–225. doi:10.1007/s10946-021-09953-6
- Huang, X., Mei, G., Zhang, J., and Abbas, R. (2021). *A Comprehensive Survey on Point Cloud Registration. arXiv preprint arXiv:2103.02690*.
- Jia, S., Ding, M., Zhang, G., and Li, X. (2016). "Improved Normal Iterative Closest Point Algorithm with Multi-Information," in *2016 IEEE International Conference on Information and Automation (ICIA) (IEEE)*, 876–881. doi:10.1109/icinfa.2016.7831942
- Korn, M., Holzkothen, M., and Pauli, J. (2014). "Color Supported Generalized-Icp," in *2014 International Conference on Computer Vision Theory and Applications (VISAPP) (IEEE)*, 3, 592–599.
- Kurobe, A., Sekikawa, Y., Ishikawa, K., and Saito, H. (2020). Corsnet: 3d Point Cloud Registration by Deep Neural Network. *IEEE Robot. Autom. Lett.* 5, 3960–3966. doi:10.1109/ira.2020.2970946
- Men, H., Gebre, B., and Pochiraju, K. (2011). "Color Point Cloud Registration with 4d Icp Algorithm," in *2011 IEEE International Conference on Robotics and Automation (IEEE)*, 1511–1516. doi:10.1109/icra.2011.5980407

## AUTHOR CONTRIBUTIONS

DL was responsible for the overall direction of the paper. DH as corresponding authors provides research ideas and writes programs for experiments. SW is responsible for data collection and translation. YC revise the manuscript. All authors participated in the writing, review, and editing of the manuscript.

## FUNDING

This work was supported by the "Pioneer" and "Leading Goose" RD Program of Zhejiang (no. 2022C01005), the Basic Public Welfare Projects in Zhejiang Province (nos LGF20G010002 and LGF20G010003), and the Public Projects of Zhejiang Province (no. LGN20G010002).

- Pais, G. D., Ramalingam, S., Govindu, V. M., Nascimento, J. C., Chellappa, R., and Miraldo, P. (2020). "3dregnet: A Deep Neural Network for 3d Point Registration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7193–7203. doi:10.1109/cvpr42600.2020.00722
- Park, J., Zhou, Q.-Y., and Koltun, V. (2017). "Colored Point Cloud Registration Revisited," in *Proceedings of the IEEE International Conference on Computer Vision*, 143–152. doi:10.1109/iccv.2017.25
- Ran, Y., and Xu, X. (2020). Point Cloud Registration Method Based on Sift and Geometry Feature. *Optik* 203, 163902. doi:10.1016/j.ijleo.2019.163902
- Ren, S., Chen, X., Cai, H., Wang, Y., Liang, H., and Li, H. (2021). Color Point Cloud Registration Algorithm Based on Hue. *Appl. Sci.* 11, 5431. doi:10.3390/app11125431
- Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). "Orb: An Efficient Alternative to Sift or Surf," in *International Conference on Computer Vision IEEE*, 2564–2571. doi:10.1109/iccv.2011.6126544
- Rusinkiewicz, S., and Levoy, M. (2001). "Efficient Variants of the Icp Algorithm," in *Proceedings Third International Conference on 3-D Digital Imaging and Modeling (IEEE)*, 145–152.
- Segal, A., Haehnel, D., and Thrun, S. (2009). Generalized-icp. *Robotics Sci. Syst.* 2, 435. doi:10.15607/rss.2009.v.021
- Serafin, J., and Grisetti, G. (2015). "Nicip: Dense Normal Based Point Cloud Registration," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE)*, 742–749. doi:10.1109/iroso.2015.7353455
- Strickland, R. N., Kim, C.-S., and McDonnell, W. F. (1987). Digital Color Image Enhancement Based on the Saturation Component. *Opt. Eng.* 26, 267609. doi:10.1117/12.7974125
- Sturm, J., Engelhard, N., Endres, F., Burgard, W., and Cremers, D. (2012). "A Benchmark for the Evaluation of Rgb-D Slam Systems," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IEEE)*, 573–580. doi:10.1109/iroso.2012.6385773
- Tong, L., and Ying, X. (2018). 3d Point Cloud Initial Registration Using Surface Curvature and Surf Matching. *3D Res.* 9, 1–16. doi:10.1007/s13319-018-0193-8
- Wang, Y., and Solomon, J. M. (2019). "Deep Closest Point: Learning Representations for Point Cloud Registration," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3523–3532. doi:10.1109/iccv.2019.00362
- Xu, C., Ding, A. S., and Zhao, K. (2021). A Novel POI Recommendation Method Based on Trust Relationship and Spatial-Temporal Factors. *Electron. Commer. Res. Appl.* 48, 101060. doi:10.1016/j.elerap.2021.101060

- Yew, Z. J., and Lee, G. H. (2020). "Rpm-net: Robust Point Matching Using Learned Features," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11824–11833. doi:10.1109/cvpr42600.2020.01184
- Yuan, W., Eckart, B., Kim, K., Jampani, V., Fox, D., and Kautz, J. (2020). "Deepgmr: Learning Latent Gaussian Mixture Models for Registration," in *European Conference on Computer Vision* (Springer), 733–750. doi:10.1007/978-3-030-58558-7\_43
- Zhou, Q.-Y., Park, J., and Koltun, V. (2018). *Open3d: A Modern Library for 3d Data Processing*. *arXiv preprint arXiv:1801.09847*.

**Conflict of Interest:** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

**Publisher's Note:** All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

*Copyright © 2022 Liu, Hong, Wang and Chen. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.*