Fitting the space line of high repeatability to noisy point cloud data

Shuai Huang\textsuperscript{1,a}, Ming Chen\textsuperscript{1,b*}, ShengLian Lu\textsuperscript{1,e}, ShouXin Chen\textsuperscript{1,d}, YongJian Zha\textsuperscript{1,e}

\textsuperscript{1}School of Computer Science and Information Engineering, Guangxi Normal University, Guilin City, China
\textsuperscript{a}email: hsldan0817@163.com, \textsuperscript{e}email:lsl@gxnu.edu.cn, \textsuperscript{d}email:shouxin1492@gmail.com, \textsuperscript{e}email: 972431785@qq.com
\textsuperscript{*}Corresponding author: Ming Chen, \textsuperscript{b}email: mingchen@gxnu.edu.cn

Abstract: The spatial line fitting algorithm with high repeatability for noise point cloud data is challenging and it is of great significance for high-precision measurement, visual guidance of robotics and CNC machining, but little study has been done at this topic. In this paper, a new algorithm is proposed to fit the spatial line with high repeatability to adapt to the point cloud data with noises. This method combines the idea of repeated residuals and the least square algorithm to filter out the noises and then solve the problem. The algorithm can successfully filter the noises. Four popular straight lines fitting algorithms are implemented as benchmarks. A large number of experimental results show that this algorithm can obtain highly repeatable and accurate straight lines.

1. Introduction

Line fitting is one of the basic problems of data processing in the field of computer graphics, which widely exists in the fields of computer-aided design, computer-aided manufacturing, robot path planning and optical measurement, etc.\cite{1-3}. There is little study on line fitting with high repeatability in noisy environment. High repeatability fitting is defined as: the same object is repeatedly scanned by different people at different time to obtain a series of point cloud data, on which one fitting algorithm is done; the less difference between each resultant fitting line, the higher repeatability the algorithm has. There is little study is done on line fitting regarding the high repeatability. In this paper, a feasible method of space line fitting is proposed.

Generally, 2D linear data fitting is performed using Least Squares (LS)\cite{4} or Total Least Squares (TLS)\cite{5}. For 3D point data set, the linear parameter equation is transformed into four parameters in\cite{7}, and then TLS algorithm is adopted to fit spatial lines. Hu\cite{4} uses LS and TLS to fit a spatial line by projecting it along XYZ coordinate axis, respectively. However, the above methods are all sensitive to noises. Principal component analysis\cite{8} is explored to minimize the sum of squares of distances between points and fitting lines. Li\cite{9} decentralizes the data and performs singular value decomposition to fit the spatial linear data. Du\cite{9} uses Newton-Gradient optimization algorithm to optimally achieve spatial line fitting. Many methods have been proposed to handle outliers among which RANdom SAmple Consensus (RANSAC)\cite{10} algorithm is most popular. RANSAC algorithm remove of noise point data through random sampling robustly; M-estimate SAmple Consensus (MSAC)\cite{11-12} is also proposed to improve RANSAC algorithm.
In this paper, a new robust line fitting algorithm is proposed to fit highly repetitive spatial lines. The repeated least trimmed squares idea is integrated into the total least squares method, improving the robustness. In the algorithm, repeated residuals are first applied to the line fitting such that the line with high repeatability can be fitted.

2. Methods

2.1 TLS method for line fitting
A space line passing a point \( K(x_0, y_0, z_0) \) with the direction \( N(A, B, C) \) can be expressed as below:

\[
\frac{x-x_0}{A} = \frac{y-y_0}{B} = \frac{z-z_0}{C}
\]  
(1)

Eq.(1) can be also represented as Eq.(2):

\[
\begin{cases}
  x = A \frac{z-z_0}{C} + x_0 \\
  y = B \frac{z-z_0}{C} + y_0
\end{cases}
\]  
(2)

Let \( a = \frac{A}{C}, b = x_0 - \frac{A}{C}z_0, c = \frac{B}{C}, d = y_0 - \frac{B}{C}z_0 \)

Eq. (1) can be further written as follows:

\[
\begin{cases}
  x = az + b \\
  y = cz + d
\end{cases}
\]  
(3)

Eq. (3) can be rewritten in matrix format:

\[
\begin{bmatrix}
x \\
y
\end{bmatrix} = \begin{bmatrix}
  z & 1 & 0 & 0 \\
  0 & 0 & z & 1
\end{bmatrix} \begin{bmatrix}
a & b & c & d
\end{bmatrix}^T
\]  
(4)

According to the least square indirect adjustment model \([4]\), the error of Eq. (4) can be expressed as:

\[
V = \begin{bmatrix}
z \\
0 \\
0 \\
0 \\
z
\end{bmatrix} \begin{bmatrix}
a & b & c & d
\end{bmatrix}^T
\]  
(5)

Let

\[
B = \begin{bmatrix}
z & 1 & 0 & 0 \\
0 & 0 & z & 1
\end{bmatrix}, L = \begin{bmatrix}
x & y \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
z & 1
\end{bmatrix}, \hat{X} = \begin{bmatrix}
a & b & c & d
\end{bmatrix}^T
\]

It yields Eq. (6):

\[
V = B\hat{X} - L
\]  
(6)

Suppose the point data is denoted as \( L_i(x_i, y_i, z_i) \), the error of Eq. (6) can be evaluated as follows:

\[
L = \begin{bmatrix}
x_1 \\
y_1 \\
\vdots \\
x_n
\end{bmatrix}, B = \begin{bmatrix}
z_1 & 1 & 0 & 0 \\
0 & 0 & z_1 & 1 \\
\vdots & \vdots & \vdots & \vdots \\
z_n & 1 & 0 & 0 \\
0 & 0 & z_n & 1
\end{bmatrix}, \hat{X} = \begin{bmatrix}
a & b & c & d
\end{bmatrix}^T
\]

The parameter vector \( \hat{X} \) is the objective unknowns for solution in Eq.(3).

2.2 Optimized line fitting
Since the fitting result of the RANSAC algorithm is sensitive to the threshold value. In this paper, the repeated minimum residual is introduced for line fitting optimization. The distance between spatial points and straight lines can be defined in Figure 1 and Eq.(7).

\[d = \sqrt{(x-x_1)^2 + (y-y_1)^2}
\]  
(7)

![Figure 1](image)

Figure 1. The distance between point \( Q \) and a straight line
In Figure 1, \( Q_i \) is the objective point, \( R \) one point on \( L \). \( N(A, B, C) \) is the normal vector of \( L \). The distance \( d_i \) for point \( Q_i \) to \( L \) can be calculated by Eq.(7):

\[
d_i = \sqrt{|M|^2 - (M \cdot N)^2}
\]

(7)

Where \( M = Q_i - R \), and operator represents dot product. The proposed algorithm is outlined by the following pseudocode of Fitting Line Func.

**Fitting Line Func.**

| Input | Output | Step 1 | Step 2 | Step 3 |
|-------|--------|--------|--------|--------|
| \( P \): the point cloud data in ROI; \( h \): trimming size; \( h_o \): the minimum sample size \( (h_o = 10\% \text{ of } P) \); \( T \): number of iteration; \( n \) | \( N(A, B, C) \): the line normal vector; \( K(x_o, y_o, z_o) \): A point on the line in Eq. (1); \( \tilde{N} \) | Initialize: Let sum of squared distance \( SSr d = \text{Inf} \); evaluate the number of \( P \) as \( n \); | For \( i = 1: T \) Do | Output \( N(A, B, C) \) and \( K(x_o, y_o, z_o) \) |
| Step 2.1: Randomly select \( h_o \) point; | Step 2.2: Fit one line to \( h_o \) points using TLS method in 2.1, obtaining \( (A, B, C) \) and \( K(x_o, y_o, z_o) \) in Eq. (1); | | Step 2.3: Calculate point to line distance \( d_j \) for each point in \( P \) as below: \( d_j = \sqrt{|M|^2 - (M \cdot N)^2} \) for each point in \( P \) as below: \( d_j = \sqrt{|M|^2 - (M \cdot N)^2} \) | \( SSrd_i < SSrd_{i-1} \) | \( N(A, B, C) = N_h(A_h, B_h, C_h) \) and \( K(x_o, y_o, z_o) = K_h(x_o, y_o, z_o) \); |
| Step 2.4: Sort \( P \) by \( d_j \) and select the \( h \) minimum points to the line obtained in Step 2.3; | Step 2.5: Fit one line to \( h \) points using TLS method, obtaining \( N_h(A_h, B_h, C_h) \) and \( K(x_o, y_o, z_o) \) in Eq. (1); | | Step 2.6: Calculate point to line distance \( d_j \) for each point in \( P \) as below: \( d_j = \sqrt{|M|^2 - (M \cdot N)^2} \) | \( SSrd_i < \text{threshold} \) | Goto Step 3 |
| Step 2.7: Calculate \( SSrd_i = \sum_{j=1}^{h} d_j^2 \) where \( (d_1^2 \leq d_2^2 \leq \ldots \leq d_{n-1}^2 \leq d_n^2) \) | | | \( SSrd_i < \text{threshold} \) | Goto Step 3 |
| Step 2.8: IF \( SSrd_i < SSrd_{i-1} \) End If | | | \( SSrd_i < \text{threshold} \) | Goto Step 3 |
| | End If | | \( SSrd_i < \text{threshold} \) | Goto Step 3 |
| Step 3: Output \( N(A, B, C) \) and \( K(x_o, y_o, z_o) \) | | | | |

The number of iterations \( T \) in Step 2 is determined according to the probability \(^{11}\) by Eq.(8).

\[
T = \frac{\log(1-p)}{\log(1-w^h)}
\]

(8)

where \( p \) is the selected points are the probability of valid points, usually \( >97\% \), which is decided by the performance of the 3D camera and the onsite sampling environment; \( w \) is the probability of any point being an noise point, usually less than 20\% and it can be calculated as:

\[
w = \frac{\text{Number of interior points in } P}{\text{Number of points in } P}
\]

\( h \) is the number of points participating in the fitting. In Eq.(8), \( w^h \) stands for the probabilities of the selected \( h \) points being all interior points and \( (1 - w^h) \) represents the probability of the \( h \) points having at least one noise point.

3. Results & Discussions

To verify the effectiveness of the proposed algorithm. The algorithm is implemented using MATLAB 2018a and run on the PC with OS Win 7, CPU Intel i7-7700 and 8 GB DDR4. Four algorithms are implemented as benchmarks, including LS \(^4\), TLS \(^7\), RANSAC \(^{10}\) and MSAC \(^{11}\). To evaluate the repeatability and accuracy of the algorithms, three quantity standards are defined by Eq.(9)-Eq.(11):

\[
A(D) = \frac{1}{n} \sum_{i=1}^{n} A_i
\]

(9)

\[
RMSE(N) = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (N_i - \bar{N})^2}
\]

(10)

\[
RMSE(K) = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (K_i - \bar{K})^2}
\]

(11)
Where $\sum_{i=1}^{n} \Delta_i$ is the sum of the distances from the fitting points to the fitting line; $A(\Delta)$ is the mean value of the distance sum; $N(A,B,C)$ is the normal vector and $K(x_0,y_0,z_0)$ is the point, through which the line passes; $\bar{N}$ is the mean value of the normal vectors of the line and $\bar{K}$ is the mean value of the points which the space line passes. $RMSE(N)$ is the Root Mean Square Error of the spatial straight line normal vectors, and $RMSE(K)$ is the Root Mean Square Error of the passing points. The more repeatable the line fitting algorithm is, the closer to 0 the value of $RMSE$ is.

$$\frac{x-1}{2} = \frac{y-2}{2} = \frac{z-5}{5}$$

(12)

**Example I:** Add fitting noisy points of 0%, 30%, 40% and 45% to the line in Eq. (12), the results by the benchmarks and the proposed algorithm are shown in Figure 2.

![Figure 2](image-url)

(a) Results without noisy points  
(b) Results of 40% outlier point clusters  
(c) Results of 30% noisy points  
(d) Results of 45% noisy points

Figure 2. Results of spatial line fitting by the benchmarks and the proposed algorithm.

It can be seen from Fig.2(a) that all the five fitting methods can obtain good result for the data set without noisy points. When the data set contains outlier clusters (see Fig.2 (b)), the fitting results of LT and TLS algorithms have very large deviation, which indicates that such methods are particularly affected by noisy point clusters. For the noise ratio shown in Fig.2(c) and Fig.2(d), LS and TLS fitting algorithms have a large deviation. RANSAC and MSAC algorithms are better, but a deviation obviously occurs. However, the algorithm in this paper can achieve a good result. To study the influence of the ratio of outliers on the line fitting algorithm, a line of 200 sampling points is generated. Thirty experiments are conducted by adjusting the ratio of noise points (15%, 30% and 45%). The statistical data, i.e., $A(\Delta), RMSE(N)$ and $RMSE(K)$ are evaluated and listed in Table 1 for repeatability evaluation.
Table 1. Results of different ratios of noise points

| Methods | 15% | 30% | 45% |
|---------|-----|-----|-----|
|         | A(Δ) | RMSE(N) | RMSE(K) | A(Δ) | RMSE(N) | RMSE(K) | A(Δ) | RMSE(N) | RMSE(K) |
| LS      | 7.56E+04 | 0.1662 | 0.1383 | 2.47E+06 | 0.2030 | 0.4319 | 6.04E+06 | 0.6054 | 0.6036 |
| TLS     | 7.49E+04 | 0.1511 | 0.1504 | 2.44E+06 | 0.1906 | 0.4037 | 5.82E+06 | 0.5541 | 0.5882 |
| RANSAC  | 4.11E+04 | 0.0929 | 0.0862 | 4.29E+05 | 0.1386 | 0.1782 | 2.26E+06 | 0.2042 | 0.2300 |
| MSAC    | 3.81E+04 | 0.0861 | 0.0692 | 4.19E+05 | 0.1005 | 0.1109 | 2.24E+06 | 0.2029 | 0.2030 |
| New     | 1.42E+04 | 0.0409 | 0.0398 | 8.03E+04 | 0.0597 | 0.0599 | 9.25E+05 | 0.0981 | 0.1002 |

It can be seen from Table 1 that the results of all methods are getting worse with the increase of noise point ratios. However, in this example, the proposed algorithm performs best among all the algorithms.

**Example II:** In Figure 3, the straight line of the outer edge of one workpiece needs to be measured. To quickly measure it, laser scanner is generally used to obtain the point cloud data, then run the line fitting algorithm to fit the data into one line, measuring the length of the line. To test the repeatability of the proposed algorithm, the workpiece will be placed ten times with different poses and at different positions. After each placement, one scanning will be done and ten data sets for the same workpiece can be acquired. All the tested algorithms will test each data set and then fit to obtain a series of spatial line.

Table 2. Results of Example II

| Methods | A(Δ) | RMSE(N) | RMSE(K) |
|---------|------|---------|---------|
| LS      | 282.1550 | 0.2691 | 0.4436 |
| TLS     | 280.8809 | 0.2601 | 0.3945 |
| RANSAC  | 67.8321  | 0.0441 | 0.1735 |
| MSAC    | 65.9582  | 0.0408 | 0.1682 |
| New     | 27.2413  | 0.0030 | 0.0120 |

The fitting results of five algorithms are shown in Figure 3, and the corresponding statistical calculation data are shown in Table 2. It can be seen from Table 2 that the minimum mean $A(\Delta)$ of the sum of distances between space points and lines is 27.2413. At the same time, the root mean square error of the normal vector and the coordinates of the points passing by the proposed algorithm is far less than that of the other four algorithms. Therefore, it shows that this algorithm can fit the spatial lines with higher accuracy and repeatability. For example, the value of $RMSE(N)$ by RANSAC algorithm is 0.0441, while the algorithm proposed in this paper is significantly reduced to be 0.003.

The above experimental results show that for the spatial point cloud data with different noise point ratios, the proposed algorithm is more accurate and repeatable than benchmarks.
4. Conclusions
It is important for precise measurement, robot technology and numerical control system to fit the point cloud data with noise into highly repeatable lines. It is a challenging job but little study has been done on this topic. In this paper, a new method is proposed to fit the point cloud data of noises to obtain high repeatable spatial lines. In this method, the minimum residual is used to filter outliers and then the total least square is used to iteratively and robustly solve the line fitting. In this paper, four commonly used linear fitting methods are implemented and used as benchmarks. Experiments show that the proposed algorithm is the most repeatable among all the test algorithms. In future work, how to realize the automatic determine the Region of Interest (ROI) can be further studied.

Acknowledgments
The authors of this paper are supported by the funding of Natural Science Foundation of China (No: 61662006, 62062015) and the Innovation Project of School of Computer Science and Information Engineering, Guangxi Normal University under the contract number JXXYYJSCXXM-002.

References
[1] Snow K , Schaffrin B. (2016) Line fitting in Euclidean 3D space. Stud Geophys Geod, 60:210–227.
[2] Zhou Y, Gong J, Fang X. (2017) Accurate coupled lines fitting in an errors-in-variables framework. Survey Review, 1-11.
[3] Wang Y, Liu Y, Li X. et al. (2020) GORFLM: totally Optimal Robust Fitting for Linear Model. Signal Processing Image Communication, 84:115834.
[4] Hu C, Chen Y, Zhu W. D. et al. (2015) Comparisons of total least squares and least squares for fitting spatial lines. Journal of Geodesy and Geodynamics, 35(4):689-692.
[5] Schaffrin B, Wieser A. (2008) On weighted total least-squares adjustment for linear regression. Geodesy, 82: 415-421.
[6] Li J. (2019) Spatial linear fitting based on singular value decomposition of the Matrix. College Mathematics, 35(02):5-8.
[7] Cui L. L, Yang R, Qian J. Y. et al. (2019) Spatial straight lines fitting algorithm based on total least squares. Journal of Chengdu University(Natural Science Edition), 129(01):106-109.
[8] Cao H. R, Wang F. C. (2010) Space fitting line with criteria of total least square and total weighted least absolute deviation. Computer Applications And Software, 27(9) : 111-112.
[9] Du M. F. (1996) Fitting of space straight line. Journal of Beijing Institute of Printing, 2:27-31.
[10] Fischler M. A., Bolles R. C. (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM, 24 (6):381–395.
[11] Torr P. H. S, Zisserman A. (2000) MLESAC: a new robust estimator with application to estimating image geometry. J. Computer. Vision Image Understanding, 78 (1):138–156.
[12] Choi S, Kim T, Yu W. (2009) Performance evaluation of RANSAC family, in: British Machine Vision Conference. DBLP, 1–12.