This is a repository copy of *A fast Manhattan frame estimation method based on normal vectors*.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/183320/

Version: Accepted Version

Article:
Zhang, Y., Ding, Y., Song, J. et al. (2 more authors) (2022) A fast Manhattan frame estimation method based on normal vectors. Journal of Field Robotics, 39 (5). pp. 557-579. ISSN 1556-4959

https://doi.org/10.1002/rob.22064

This is the peer reviewed version of the following article: Zhang, Y., Ding, Y., Song, J., Li, J., & Wei, H.-L. (2022). A fast Manhattan frame estimation method based on normal vectors. Journal of Field Robotics, which has been published in final form at https://doi.org/10.1002/rob.22064. This article may be used for non-commercial purposes in accordance with Wiley Terms and Conditions for Use of Self-Archived Versions. This article may not be enhanced, enriched or otherwise transformed into a derivative work, without express permission from Wiley or by statutory rights under applicable legislation. Copyright notices must not be removed, obscured or modified. The article must be linked to Wiley's version of record on Wiley Online Library and any embedding, framing or otherwise making available the article or pages thereof by third parties from platforms, services and websites other than Wiley Online Library must be prohibited.

Reuse
Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

A Fast Manhattan Frame Estimation Method based on Normal Vectors

Yutong Zhang
Key Laboratory of Dynamics and Control of Flight Vehicle
Beijing Institute of Technology
Beijing 100081, China
3120160041@bit.edu.cn

Yan Ding*
Key Laboratory of Dynamics and Control of Flight Vehicle
Beijing Institute of Technology
Beijing 100081, China
dingyan@bit.edu.cn

Jianmei Song
Key Laboratory of Dynamics and Control of Flight Vehicle
Beijing Institute of Technology
Beijing 100081, China
sjm318@bit.edu.cn

Jiaxin Li
Key Laboratory of Dynamics and Control of Flight Vehicle
Beijing Institute of Technology
Beijing 100081, China
3120200057@bit.edu.cn

Hua-Liang Wei
Department of Automatic Control and Systems Engineering
University of Sheffield
Sheffield S1 3JD, UK
w.hualiang@sheffield.ac.uk

Abstract

In most human made scenes, such as high-rise urban city or indoor environment, the surface normal vectors or direction vectors are concentrated in three orthogonal principal directions. The scene of such a pattern is called Manhattan World (MW), and the coordinate frame formed by the three principal directions is called Manhattan Frame (MF). MF estimation methods have been applied to many different fields, such as scene reconstruction, Visual based Simultaneous Localization And Mapping (V-SLAM) and camera calibration. In this paper, we propose a novel MF estimation method based on a set of normal vectors. A cost function of normal vectors and MF axes is introduced based on the trigonometric function. For computational purpose, the cost function is significantly simplified by making use of vector dot and cross products, and the reduced cost function only involves 14 scalar parameters that need to be computed with O(n) complexity. The experimental results show that the proposed MF estimation method has excellent real-time performance and gives high accuracy on both the virtual and real-world benchmark datasets of different sizes.

1 Introduction

In most of the artificial scenes, such as high-rise urban city and indoor environment, there is often a unified scene structure pattern, where the floor, ceiling, wall corner and edge line are often perpendicular or parallel to each other, and their normal vectors or direction vectors are concentrated in three main orthogonal directions in 3D (three-dimensional) space. The coordinate frame, which defines these three main directions along X, Y and Z axes, is called the Manhattan Frame (MF) (Straub et al., 2014; Ghanem et al., 2015; Straub et al., 2015, 2017), and the scene with such a structural feature is called the Manhattan World (MW) (Coughlan and Yuille, 1999). MF estimation is widely used in many visual tasks, such as Visual based Simultaneous

*Corresponding author

Final Accepted Manuscript 22 January 2022 for publication in Journal of Field Robotics

Localization And Mapping (V-SLAM)(Wang and Wu, 2019; Zhou et al., 2015; Hsiao et al., 2017; Le and Košecka, 2017), 3D scene reconstruction (Furukawa et al., 2009; Sinha et al., 2009), scene understanding (Straub et al., 2014; Silberman et al., 2012; Gupta et al., 2013; Choi et al., 2013), scene layout estimation (Hedau et al., 2009; Lee et al., 2009; Hedau et al., 2010; Schwing et al., 2012) and camera calibration (Straub et al., 2014; Bazin et al., 2014).

In recent years, many methods (Ghanem et al., 2015; Furukawa et al., 2009; Gupta et al., 2013; Taylor and Cowley, 2013; Zhang et al., 2010) have been proposed for MF estimation, but these methods only provide suboptimal solutions (Joo et al., 2018). In order to ensure the global optimal result, Bazin et al. (2012a) and Parra Bustos et al. (2014) proposed new methods based on Branch-and-Bound (Bazin et al., 2012b) (BnB), but the amount of the calculation of these methods is very large. In order to reduce the amount of computations, Joo et al. (2018) proposed a BnB method based on Extended Gaussian Image (EGI). This method has reliable real-time performance, but it needs to adjust the resolution of EGI to achieve the balance of accuracy and computing speed.

In order to improve the real-time performance of MF estimation, while ensuring the accuracy and stability, this paper proposes a novel MF estimation method. Firstly, the cost function of a single MF axis direction is defined based on the trigonometric function. Then an easy-to-calculate cost function is defined by using vector dot and cross product operations, so that the part of the normal vector sample set in the cost function only contains 14 scalar parameters that need to be computed with O(n) complexity. Finally, the cost function of MF rotation matrix is defined by adding the cost functions of the three MF axes, and methods for determining the initial value and searching for an optimal solution are designed.

The number of scalar parameters of the designed cost function is fixed, so the calculation amount of the MF optimization process does not increase with the increase of the number of normal vectors. The computation process of the 14 scalar parameters in the cost function has O(n) complexity, and the optimization process of MF has O(1) complexity; these enable the method to have outstanding real-time performance. The proposed method can accurately estimate the MF which involves nearly 300000 unit normal vectors within 5 ms with CPU only. The cost function is smooth and convex, therefore it has a global minimum which guarantees an optimal result for the MF estimation. In addition, an initial value determination method is designed to speed up the MF estimation convergence to the global optimum. These features ensure both high accuracy and robustness of the MF estimation. The proposed method does not include any parameter that needs to be adjusted manually. The main contributions of the paper include:

(1) A new cost function with parallel or vertical vector constraints based on the trigonometric function.

(2) A novel easy-to-calculate cost function, involving only 14 scalar parameters to be computed with O(n) complexity.

(3) A new method for determining the MF initial value, which is significantly useful for speeding up the MF estimation convergence to the global optimum.

The remaining of the paper is arranged as follows. In Section 2, the related work of the MF estimation methods is presented. In Section 3, the methodology of the proposed MF estimation method in this paper is described in detail. In Section 4, several experiments are carried out to evaluate the performance of the proposed method. Section 5 briefly summarizes the work.

2 Related Work

In the early work (Lee et al., 2009; Ramalingam and Brand, 2013; Lee et al., 2010), RGB (Red-Green-Blue) image was used as input, and the MF was estimated based on perspective geometry. The image gradient or line features were extracted and clustered, then the MF was estimated by estimating the vanishing points

and lines.

In recent years, with the development of RGB-D (Red-Green-Blue-Depth) camera technology, RGB-D images are more and more easy to obtain. As a result, MF estimation methods based on RGB image and depth information are proposed (Ghanem et al., 2015; Taylor and Cowley, 2013; Wu and Wang, 2017). Compared with the MF estimation methods based only on RGB image, the methods based on RGB-D image utilize both 2D RGB information and 3D depth information, so the RGB-D based methods are more accurate and stable (Wang and Wu, 2019).

Furukawa et al. (2009) proposed an MF estimation method based on binocular vision, where binocular 3D reconstruction is used to extract the main plane direction and orientation points. Then, the hemispherical histogram is used to count these directions. The MF is estimated by finding three orthogonal clusters. The accuracy of this method depends on the resolution of the hemispherical histogram. A lower resolution usually cannot guarantee a high estimation accuracy. On the contrary, a higher resolution may make the process of MF estimation more sensitive to noise.

Silberman et al. (2012) proposed a method to explain the support relationship of indoor scene elements from RGB-D images. In this work, in order to estimate the MF, a series of candidate directions are selected firstly, and then the coordinate axis direction of MF is determined through exhaustive search with a scoring heuristic method. This method usually takes a long time, and the output result is suboptimal.

Taylor and Cowley (2013) proposed a method to analyze the Manhattan structure of indoor scene using Kinect camera. The method first finds the area of the ground below the camera's view, and then determines the MF by choosing a main wall plane direction perpendicular to the ground. This method is not suitable for the situation that there is no ground in the camera's field of view.

In some visual tasks, only the gravity direction needs to be estimated. Zhang et al. (2010) proposed a framework for semantic scene parsing and object recognition, where the normal information of the point cloud is extracted, then a group of normal vectors close to the upward direction are selected in the camera coordinate system. The gravity direction is estimated based on the RANSAC (RANdom SAmple Consensus) (Fischler and Bolles, 1981) algorithm. Gupta et al. (2013) proposed a method for object boundary detection and segmentation. The method selects the Y axis of the camera coordinate system as the initial direction of the gravity vector, then selects the normal vector parallel to the initial direction and construct the parallel constraint, and finally selects the normal vector perpendicular to the initial direction and constructs the vertical constraint to optimize the gravity vector. The limitation of the methods proposed in (Gupta et al., 2013; Zhang et al., 2010) is that the direction of the Y axis of the camera coordinate system should be near parallel to the gravity direction.

Ghanem et al. (2015) proposed an MF estimation method by introducing sparse constraints of scene normal information, and used it to estimate the surfaces aligned with the MF axes and the outlying surfaces. This approach is based on the assumption that the camera attitude is normal, where the angle between the normal vector of floor or ceiling and the Y-axis of camera coordinate system is small. The initial value of the optimization is selected as identity matrix. The global optimization result cannot be guaranteed when the camera is in abnormal posture.

Straub et al. (2014) proposed a new approach called MMF (Mixture of Manhattan Frame), where the multiple MFs of the scene can be estimated simultaneously. However, the estimation accuracy is limited, and the calculation time is up to 100s, which cannot meet the real-time requirements. They also proposed a method called RTMF (Real Time Manhattan Frame)(Straub et al., 2015) based on the normal vectors. With the acceleration of GPU (Graphics Processing Unit), RTMF meets the requirements of real-time application, but it is sensitive to initial conditions and cannot guarantee the global optimal result.

In order to ensure the global optimal estimation, Bazin et al. (2012a) proposed an MF estimation method by exploiting the Branch-and-Bound (BnB) framework (Bazin et al., 2012b), where a global search in the rotating

search space (Hartley and Kahl, 2009) is conducted based on interval analysis, and the MF is estimated by determining the orthogonal vanishing points. The main limitation of the method is that it is not suitable for real-time application because of the large amount of calculations (Parra Bustos et al., 2014).

In order to improve the calculation speed, Parra Bustos et al. (2014) proposed a more efficient boundary function for BnB framework. This method can register up to 1000 points in 2 seconds, but there is still a gap between the requirements of real-time application.

Joo et al. (2018) extended the MF estimation method presented in (Joo et al., 2016), and improved the BnB framework and proposed a near real-time MF estimation method. In this work, the normal vectors are projected onto an equal rectangular two-dimensional plane, and are discretized according to the manually set resolution to obtain the Extended Gaussian Image (EGI) (Horn, 1984), which contains the histogram information of normal vector distribution. The BnB method was then used to estimate MF in the obtained EGI. Compared with other BnB based methods (Bazin et al., 2012a; Parra Bustos et al., 2014), the computational efficiency of this method is greatly improved. However, this method needs to adjust the resolution of EGI to an appropriate value so as to achieve a good balance between accuracy and running time.

3 Methodology

Let $\{\mathbf{a}_i | i = 1, 2, ..., n\}$ be a set of unit normal vectors of a 3D scene point cloud. Most of the normal vector directions are concentrated in three orthogonal main directions, and there are a number of normal vectors whose directions are randomly distributed on the unit sphere, representing outliers. The goal of the MF estimation is to find the MF whose three axes are parallel to the three orthogonal main directions.

For an ideal case, any normal vector is parallel or perpendicular to any coordinate axis of MF, that is, the angle between any normal vector and any MF axis is 0°, 90° or 180°. In practice, however, because of the existence of noise and outliers, the normal vectors are not strictly parallel or perpendicular to the axis of MF; this can result in errors in MF estimation. This paper aims to MF estimation performance in terms of both accuracy and computational reduction to meet the requirements of real-time applications. In doing so, a cost function is designed, whose solution can be achieved by optimizing the corresponding rotation matrix of MF, so as to obtain the result of MF.

3.1 The Definition of the Cost Function

Let θ be the angle between a normal vector and a MF axis. In order to make θ close to 0, 90 or 180 degrees, the cost function $e(\theta)$ can be defined as

$$e(\theta) = \sin^2\theta\cos^2\theta \tag{1}$$

When the normal vector and the coordinate axis are parallel, θ is close to 0 or 180 degrees, $\sin^2\theta$ is close to zero. When the normal vector and the coordinate axis are vertical, θ is close to 90 degrees, $\cos^2\theta$ is close to zero. The Cartesian coordinate curve of $e(\theta) = \sin^2\theta\cos^2\theta$ is shown in Fig. 1.

Let $\mathbf{r}$ be the unit vector corresponding to one of the MF axes. The Single Normal vector and Single Axis (SNSA) cost function $E_i(\mathbf{r})$ associated with $\mathbf{r}$ and the i-th normal vector $\mathbf{a}_i$ in the normal vector set is defined as

$$E_i(\mathbf{r}) = \sin^2 < \mathbf{r}, \mathbf{a}_i > \cos^2 < \mathbf{r}, \mathbf{a}_i > \tag{2}$$

where $< \mathbf{r}, \mathbf{a}_i >$ denotes the angle between $\mathbf{r}$ and $\mathbf{a}_i$.

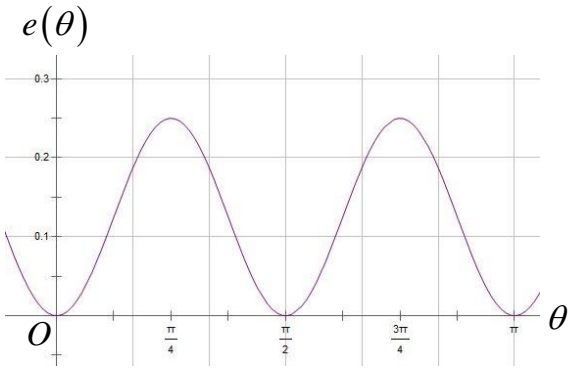


Figure 1: The Cartesian Coordinate Curve of the Cost Function $e(\theta) = \sin^2\theta\cos^2\theta$. When $\theta = \frac{k\pi}{2}, k \in Z$ (Z represents integer field), the function value is zero, and the closer to $\theta = \frac{k\pi}{2}, k \in Z$, the smaller the function value is. When $\theta = \frac{k\pi}{2} + \frac{\pi}{4}, k \in Z$, the function value is the largest. In addition, the function $e(\theta) = \sin^2\theta\cos^2\theta$ is smooth and continuous in all domains.

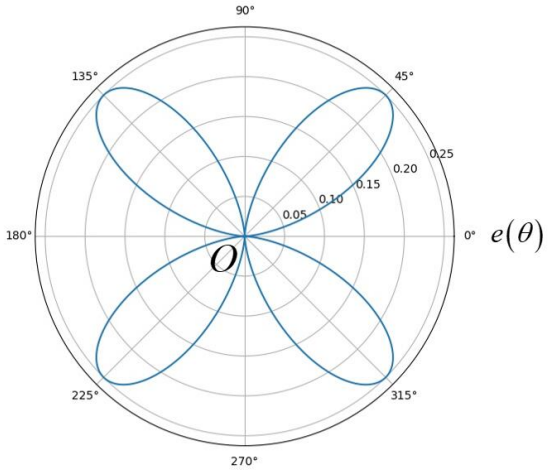


Figure 2: The Polar Coordinate Curve of $e(\theta) = \sin^2\theta\cos^2\theta$.

Fig.2 shows the polar coordinate curve of$e(\theta) = \sin^2\theta\cos^2\theta$, and Fig.3 is the spherical coordinate curve surface of $E_i = \sin^2 < \mathbf{r}, \mathbf{a}_i > \cos^2 < \mathbf{r}, \mathbf{a}_i >$ when the vector $\mathbf{a}_i$ and Z-axis are in the same direction.

It can be seen from Fig.2 and Fig.3 that the spherical coordinate curve surface of the cost function $E_i = \sin^2 < \mathbf{r}, \mathbf{a}_i > \cos^2 < \mathbf{r}, \mathbf{a}_i >$ is a hourglass shaped curve surface after the polar coordinate curve of Fig.2 is rotated 180 degrees around its polar axis. The upper, lower and the waist part of the curve surface are concave inward, while the rest are protruding outward.

The Multiple Normal vector and Single Axis (MNSA) cost function associated with the vector $\mathbf{r}$ and the vector set $\{\mathbf{a}_i | i = 1, 2, ..., n\}$ is defined as the arithmetic mean of the SNSA cost function (2) of each vector $\mathbf{a}_i$, shown as follow

$$E(\mathbf{r}) = \frac{1}{n}\sum_{i=1}^{n} E_i(\mathbf{r}) \tag{3}$$

Combining (2) and (3), we can obtain

$$E(\mathbf{r}) = \frac{1}{n}\sum_{i=1}^{n} \sin^2 < \mathbf{r}, \mathbf{a}_i > \cos^2 < \mathbf{r}, \mathbf{a}_i > \tag{4}$$

Fig.4 is the spherical coordinate curve surface of the cost function corresponding to two orthogonal normal vectors. Fig.5 is the spherical coordinate curve surface of the cost function corresponding to three orthogonal normal vectors. It can be seen that the MNSA cost has a minimum value when the vector $\mathbf{r}$ points to one of the MF axis, which satisfy the parallel or vertical constraints. In addition, the curve surface has no other minimum points which do not satisfy the constraints.

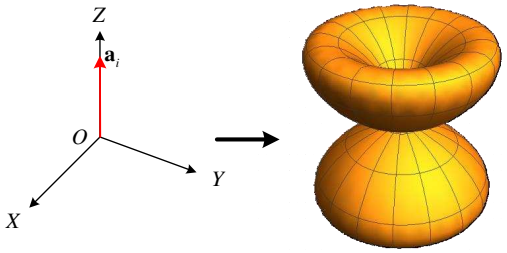


Figure 3: The Spherical Coordinate Curve Surface of $E_i(\mathbf{r}) = \sin^2 < \mathbf{r}, \mathbf{a}_i > \cos^2 < \mathbf{r}, \mathbf{a}_i >$. It is a hourglass shaped curve surface after the polar coordinate curve of Fig.2 is rotated 180 degrees around its polar axis.

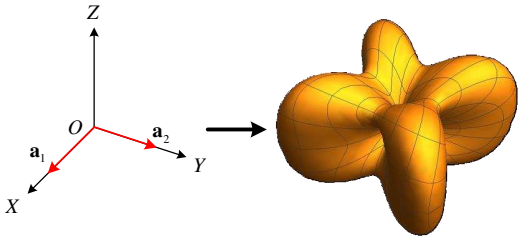


Figure 4: The Curve Surface of the MNSA Cost Function Corresponding to Two Orthogonal Normal Vectors $\mathbf{a}_1$ and $\mathbf{a}_2$. Along the $\mathbf{a}_1$, $\mathbf{a}_2$ and $\mathbf{a}_1 \times \mathbf{a}_2$ directions, the curve surface concave inward, and the rest parts are protruding outwards.

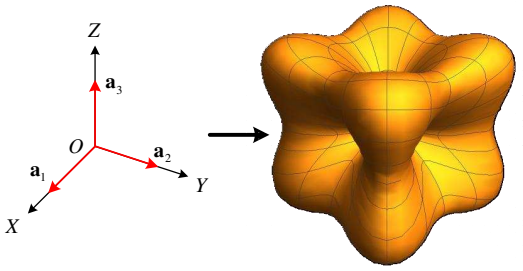


Figure 5: The Curve Surface of the MNSA Cost Function Corresponding to Three Orthogonal Normal Vectors $\mathbf{a}_1$, $\mathbf{a}_2$ and $\mathbf{a}_3$. The curve surface of the cost function is a round corner cube whose the six sides are concave inward. Along the $\mathbf{a}_1$, $\mathbf{a}_2$ and $\mathbf{a}_3$ direction, the curve surface concave inward, and the rest of the parts are protruding outwards.

3.2 Simplification of the MNSA Cost Function

In order to reduce the amount of calculations in the optimization process of MF estimation, the cost function defined in (4) is simplified in this section, so that the variables related to the normal vectors $\{\mathbf{a}_i | i = 1, 2, ..., n\}$ in the MNSA cost function are separated from the vector $\mathbf{r}$. In this way, the resulting cost function only contains 14 scalar parameters that need to be calculated with O(n) complexity. In the following, we use vector cross and dot product operations to simply the MNSA cost function.

The vector cross multiplication property is

$$|\sin < \mathbf{r}, \mathbf{a}_i >| = |\mathbf{a}_i \times \mathbf{r}| \tag{5}$$

The vector dot multiplication property is

$$|\cos < \mathbf{r}, \mathbf{a}_i >| = |\mathbf{a}_i \cdot \mathbf{r}| \tag{6}$$

For scalars, the operator $|\cdot|$ denotes an absolute value operation, whereas for vectors, it denotes the modular length operation.

Define $\mathbf{a}_i = \begin{bmatrix} a_i & b_i & c_i \end{bmatrix}^{\mathrm{T}}$ and $\mathbf{r} = \begin{bmatrix} x & y & z \end{bmatrix}^{\mathrm{T}}$, the cross and dot multiplications for vectors and matrices are respectively defined as

$$\mathbf{a}_i \times \mathbf{r} = [\mathbf{a}_i]_\times \mathbf{r} = \begin{bmatrix} 0 & -c_i & b_i \\ c_i & 0 & -a_i \\ -b_i & a_i & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{7}$$

$$\mathbf{a}_i \cdot \mathbf{r} = \mathbf{r}^{\mathrm{T}} \mathbf{a}_i = \begin{bmatrix} x & y & z \end{bmatrix} \begin{bmatrix} a_i \\ b_i \\ c_i \end{bmatrix} \tag{8}$$

where the operator $[\cdot]_\times$ represents the operation of converting a 3D vector to a 3×3 skew symmetric matrix.

We can obtain that

$$|\sin < \mathbf{r}, \mathbf{a}_i > \cos < \mathbf{r}, \mathbf{a}_i >| = \left| [\mathbf{a}_i]_\times \mathbf{r}\mathbf{r}^{\mathrm{T}} \mathbf{a}_i \right| \tag{9}$$

For $[\mathbf{a}_i]_\times \mathbf{r}\mathbf{r}^{\mathrm{T}} \mathbf{a}_i$, we separate the variables of $\mathbf{a}_i$ and $\mathbf{r}$ to two matrices, as follow

$$[\mathbf{a}_i]_\times \mathbf{r}\mathbf{r}^{\mathrm{T}} \mathbf{a}_i = \begin{bmatrix} 0 & -b_i c_i & b_i c_i & -a_i c_i & a_i b_i & {b_i}^2 - {c_i}^2 \\ a_i c_i & 0 & -a_i c_i & b_i c_i & {c_i}^2 - {a_i}^2 & -a_i b_i \\ -a_i b_i & a_i b_i & 0 & {a_i}^2 - {b_i}^2 & -b_i c_i & a_i c_i \end{bmatrix} \begin{bmatrix} x^2 \\ y^2 \\ z^2 \\ xy \\ xz \\ yz \end{bmatrix} \tag{10}$$

The detailed derivation process of formula (10) is in Appendix A.

Two matrix functions are defined as follow

$$\mathbf{V}_2(\mathbf{r}) = \begin{bmatrix} x^2 \\ y^2 \\ z^2 \\ xy \\ xz \\ yz \end{bmatrix} \tag{11}$$

$$\mathbf{A}(\mathbf{a}_i) = \begin{bmatrix} 0 & -b_i c_i & b_i c_i & -a_i c_i & a_i b_i & {b_i}^2 - {c_i}^2 \\ a_i c_i & 0 & -a_i c_i & b_i c_i & {c_i}^2 - {a_i}^2 & -a_i b_i \\ -a_i b_i & a_i b_i & 0 & {a_i}^2 - {b_i}^2 & -b_i c_i & a_i c_i \end{bmatrix} \tag{12}$$

So we can obtain

$$[\mathbf{a}_i]_\times \mathbf{r}\mathbf{r}^{\mathrm{T}}\mathbf{a}_i = \mathbf{A}(\mathbf{a}_i)\,\mathbf{V}_2(\mathbf{r}) \tag{13}$$

Combining (2), (9) and (13) gives:

$$E_i(\mathbf{r}) = {\mathbf{V}_2}^{\mathrm{T}}(\mathbf{r})\,\mathbf{A}^{\mathrm{T}}(\mathbf{a}_i)\,\mathbf{A}(\mathbf{a}_i)\,\mathbf{V}_2(\mathbf{r}) \tag{14}$$

where

$$\mathbf{A}^{\mathrm{T}}(\mathbf{a}_i)\,\mathbf{A}(\mathbf{a}_i) = \begin{bmatrix} {a_i}^2{b_i}^2 + {a_i}^2{c_i}^2 & -{a_i}^2{b_i}^2 & -{a_i}^2{c_i}^2 \\ -{a_i}^2{b_i}^2 & {a_i}^2{b_i}^2 + {b_i}^2{c_i}^2 & -{b_i}^2{c_i}^2 \\ -{a_i}^2{c_i}^2 & -{b_i}^2{c_i}^2 & {a_i}^2{c_i}^2 + {b_i}^2{c_i}^2 \\ a_i b_i {c_i}^2 - {a_i}^3 b_i + a_i {b_i}^3 & {a_i}^3 b_i - a_i {b_i}^3 + a_i b_i {c_i}^2 & -2 a_i b_i {c_i}^2 \\ a_i {c_i}^3 - {a_i}^3 c_i + a_i {b_i}^2 c_i & -2 a_i {b_i}^2 c_i & a_i {b_i}^2 c_i - a_i {c_i}^3 + {a_i}^3 c_i \\ -2 {a_i}^2 b_i c_i & {a_i}^2 b_i c_i - {b_i}^3 c_i + b_i {c_i}^3 & {b_i}^3 c_i - b_i {c_i}^3 + {a_i}^2 b_i c_i \\ a_i b_i {c_i}^2 - {a_i}^3 b_i + a_i {b_i}^3 & a_i {c_i}^3 - {a_i}^3 c_i + a_i {b_i}^2 c_i & -2 {a_i}^2 b_i c_i \\ {a_i}^3 b_i - a_i {b_i}^3 + a_i b_i {c_i}^2 & -2 a_i {b_i}^2 c_i & {a_i}^2 b_i c_i - {b_i}^3 c_i + b_i {c_i}^3 \\ -2 a_i b_i {c_i}^2 & a_i {b_i}^2 c_i - a_i {c_i}^3 + {a_i}^3 c_i & {b_i}^3 c_i - b_i {c_i}^3 + {a_i}^2 b_i c_i \\ {a_i}^2{c_i}^2 + {b_i}^2{c_i}^2 + {a_i}^4 - 2{a_i}^2{b_i}^2 + {b_i}^4 & -3{a_i}^2 b_i c_i + b_i {c_i}^3 + {b_i}^3 c_i & -3 a_i {b_i}^2 c_i + a_i {c_i}^3 + {a_i}^3 c_i \\ -3{a_i}^2 b_i c_i + b_i {c_i}^3 + {b_i}^3 c_i & {a_i}^2{b_i}^2 + {c_i}^4 - 2{a_i}^2{c_i}^2 + {a_i}^4 + {b_i}^2{c_i}^2 & a_i {b_i}^3 + {a_i}^3 b_i - 3 a_i b_i {c_i}^2 \\ -3 a_i {b_i}^2 c_i + a_i {c_i}^3 + {a_i}^3 c_i & a_i {b_i}^3 + {a_i}^3 b_i - 3 a_i b_i {c_i}^2 & {b_i}^4 - 2{b_i}^2{c_i}^2 + {c_i}^4 + {a_i}^2{b_i}^2 + {a_i}^2{c_i}^2 \end{bmatrix} \tag{15}$$

The detailed derivation process of formula (14) is in Appendix A.

Combining (4), (14) and (15) gives:

$$E(\mathbf{r}) = {\mathbf{V}_2}^{\mathrm{T}}(\mathbf{r}) \left(\frac{1}{n} \sum_{i=1}^{n} \mathbf{A}^{\mathrm{T}}(\mathbf{a}_i)\,\mathbf{A}(\mathbf{a}_i) \right) \mathbf{V}_2(\mathbf{r}) \tag{16}$$

Define

$$\mathbf{M} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{A}^{\mathrm{T}}(\mathbf{a}_i)\,\mathbf{A}(\mathbf{a}_i) \tag{17}$$

The MNSA cost function $E(\mathbf{r})$ can then be simplified as:

$$E(\mathbf{r}) = \mathbf{V}_2{}^{\mathrm{T}}(\mathbf{r})\,\mathbf{M}\mathbf{V}_2(\mathbf{r}) \tag{18}$$

Define

$$S_{uvw} = \frac{1}{n}\sum_{i=1}^{n} {a_i}^u {b_i}^v {c_i}^w \tag{19}$$

Combining (15), (17) and (19), yields,

$$\mathbf{M} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{A}^{\mathrm{T}}(\mathbf{a}_i)\,\mathbf{A}(\mathbf{a}_i) = \begin{bmatrix} S_{220}+S_{202} & -S_{220} & -S_{202} \\ -S_{220} & S_{220}+S_{022} & -S_{022} \\ -S_{202} & -S_{022} & S_{202}+S_{022} \\ S_{130}-S_{310}+S_{112} & S_{310}-S_{130}+S_{112} & -2S_{112} \\ S_{103}-S_{301}+S_{121} & -2S_{121} & S_{301}-S_{103}+S_{121} \\ -2S_{211} & S_{013}-S_{031}+S_{211} & S_{031}-S_{013}+S_{211} \end{bmatrix}$$
$$\begin{bmatrix} S_{130}-S_{310}+S_{112} & S_{103}-S_{301}+S_{121} & -2S_{211} \\ S_{310}-S_{130}+S_{112} & -2S_{121} & S_{013}-S_{031}+S_{211} \\ -2S_{112} & S_{301}-S_{103}+S_{121} & S_{031}-S_{013}+S_{211} \\ S_{202}+S_{022}+S_{400}-2S_{220}+S_{040} & S_{013}+S_{031}-3S_{211} & S_{103}+S_{301}-3S_{121} \\ S_{013}+S_{031}-3S_{211} & S_{220}+S_{004}-2S_{202}+S_{400}+S_{022} & S_{130}+S_{310}-3S_{112} \\ S_{103}+S_{301}-3S_{121} & S_{130}+S_{310}-3S_{112} & S_{040}-2S_{022}+S_{004}+S_{220}+S_{202} \end{bmatrix} \tag{20}$$

The matrix $\mathbf{M}$ is determined by the following 15 scalar parameters

$$\begin{aligned} &S_{400}, S_{310}, S_{220}, S_{130}, S_{040}, \\ &S_{301}, S_{211}, S_{121}, S_{031}, \\ &S_{202}, S_{112}, S_{022}, \\ &S_{103}, S_{013}, \\ &S_{004} \end{aligned} \tag{21}$$

Note that the vectors $\mathbf{a}_i$ $(i = 1, 2, ..., n)$ are unit vectors and there is a constraint on the three elements of each $\mathbf{a}_i$ as follows:

$$\frac{1}{n}\sum_{i=1}^{n}\left({a_i}^2 + {b_i}^2 + {c_i}^2\right)^2 = 1 \tag{22}$$

By expanding $\frac{1}{n}\sum_{i=1}^{n}\left({a_i}^2 + {b_i}^2 + {c_i}^2\right)^2$, we can get

$$\frac{1}{n}\sum_{i=1}^{n}\left({a_i}^2+{b_i}^2+{c_i}^2\right)^2=S_{400}+S_{040}+S_{004}+2S_{220}+2S_{202}+2S_{022} \tag{23}$$

Using (22) and (23), it can be known that:

$$S_{004}=1-\left(S_{400}+S_{040}+2S_{220}+2S_{202}+2S_{022}\right) \tag{24}$$

Therefore, in order to calculate each S_{uvw} with the normal vector set $\{\mathbf{a}_i\}$, only the following 14 scalar parameters are needed:

$$\begin{aligned}&S_{400},S_{310},S_{220},S_{130},S_{040},\\&S_{301},S_{211},S_{121},S_{031},\\&S_{202},S_{112},S_{022},\\&S_{103},S_{013}\end{aligned} \tag{25}$$

Finally, the 15th parameter S_{004} can be worked out using (24); this significantly reduces the overall computational load.

3.3 The Definition of MNMA Cost Function and the Estimation of the MF

In order to estimate the MF, we design the cost function of Multiple Normal vectors and Multiple MF Axis (MNMA), represented by the rotation matrix $\mathbf{R}$ of the MF.

Let $\mathbf{R}=\begin{bmatrix}\mathbf{r_1} & \mathbf{r_2} & \mathbf{r_3}\end{bmatrix}$, where $\mathbf{r_1}$,$\mathbf{r_2}$,$\mathbf{r_3}$ are the corresponding unit vector of the X, Y and Z axes of the MF in the camera coordinate system. We define MNMA cost function as the sum of the X, Y and Z axes' MNSA cost values, as follow:

$$\begin{aligned}E(\mathbf{R})&=E(\mathbf{r}_1)+E(\mathbf{r}_2)+E(\mathbf{r}_3)\\&=\begin{bmatrix}{\mathbf{V}_2}^{\mathrm{T}}(\mathbf{r}_1) & {\mathbf{V}_2}^{\mathrm{T}}(\mathbf{r}_2) & {\mathbf{V}_2}^{\mathrm{T}}(\mathbf{r}_3)\end{bmatrix}\begin{bmatrix}\mathbf{M} & & \\ & \mathbf{M} & \\ & & \mathbf{M}\end{bmatrix}\begin{bmatrix}\mathbf{V}_2(\mathbf{r}_1)\\ \mathbf{V}_2(\mathbf{r}_2)\\ \mathbf{V}_2(\mathbf{r}_3)\end{bmatrix}\end{aligned} \tag{26}$$

In this paper, the Levenberg-Marquardt (LM) optimization algorithm is used to estimate $\mathbf{R}$. In doing so, it needs to find a function $\mathbf{f}(\mathbf{R})$ that satisfies

$$E(\mathbf{R})=\mathbf{f}^{\mathrm{T}}(\mathbf{R})\,\mathbf{f}(\mathbf{R}) \tag{27}$$

The matrix $\mathbf{M}$ can be written as the product of a matrix and its transpose

$$\mathbf{M}=\left(\frac{1}{\sqrt{n}}\begin{bmatrix}\mathbf{A}(\mathbf{a}_1)\\ \mathbf{A}(\mathbf{a}_2)\\ \vdots\\ \mathbf{A}(\mathbf{a}_n)\end{bmatrix}\right)^{\mathrm{T}}\frac{1}{\sqrt{n}}\begin{bmatrix}\mathbf{A}(\mathbf{a}_1)\\ \mathbf{A}(\mathbf{a}_2)\\ \vdots\\ \mathbf{A}(\mathbf{a}_n)\end{bmatrix}={\mathbf{A}'}^{\mathrm{T}}\mathbf{A}' \tag{28}$$

The matrix $\mathbf{M}$ can be further decomposed as:

$$\mathbf{M} = \mathbf{Q} diag\left(\lambda_1, \lambda_2, ..., \lambda_6\right) \mathbf{Q}^{\mathrm{T}} \tag{29}$$

where $\lambda_k \geq 0, k = 1, 2, ..., 6$, and the matrix $\mathbf{Q}$ is an orthogonal matrix.

Define

$$\mathbf{H} = diag\left(\sqrt{\lambda_1}, \sqrt{\lambda_2}, ..., \sqrt{\lambda_6}\right) \mathbf{Q}^{\mathrm{T}} \tag{30}$$

So the matrix $\mathbf{M}$ is formed by

$$\mathbf{M} = \mathbf{H}^{\mathrm{T}}\mathbf{H} \tag{31}$$

Then

$$E\left(\mathbf{R}\right) = \mathbf{f}^{\mathrm{T}}\left(\mathbf{R}\right)\mathbf{f}\left(\mathbf{R}\right) = \begin{bmatrix} \mathbf{V}_2{}^{\mathrm{T}}\left(\mathbf{r}_1\right)\mathbf{H}^{\mathrm{T}} & \mathbf{V}_2{}^{\mathrm{T}}\left(\mathbf{r}_2\right)\mathbf{H}^{\mathrm{T}} & \mathbf{V}_2{}^{\mathrm{T}}\left(\mathbf{r}_3\right)\mathbf{H}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \mathbf{H}\mathbf{V}_2\left(\mathbf{r}_1\right) \\ \mathbf{H}\mathbf{V}_2\left(\mathbf{r}_2\right) \\ \mathbf{H}\mathbf{V}_2\left(\mathbf{r}_3\right) \end{bmatrix} \tag{32}$$

where $\mathbf{f}\left(\mathbf{R}\right) = \begin{bmatrix} \mathbf{H}\mathbf{V}_2\left(\mathbf{r}_1\right) \\ \mathbf{H}\mathbf{V}_2\left(\mathbf{r}_2\right) \\ \mathbf{H}\mathbf{V}_2\left(\mathbf{r}_3\right) \end{bmatrix}$.

From (11), the Jacobian matrix of $\mathbf{V}_2\left(\mathbf{r}\right)$ with respect to $\mathbf{r}$ is

$$\mathbf{J}_{\mathbf{V}_2}\left(\mathbf{r}\right) = \frac{\partial \mathbf{V}_2\left(\mathbf{r}\right)}{\partial \mathbf{r}} = \begin{bmatrix} 2x & 0 & 0 \\ 0 & 2y & 0 \\ 0 & 0 & 2z \\ y & x & 0 \\ z & 0 & x \\ 0 & z & y \end{bmatrix} \tag{33}$$

Let $\exp([\Delta\phi]_\times)\mathbf{R}$ be a left perturbed matrix of $\mathbf{R}$, where $\Delta\phi$ corresponds to the Lie algebra of the rotation perturbation, and $\exp([\Delta\phi]_\times)$ represents the perturbation rotation matrix corresponding to $\Delta\phi$. The Jacobian matrices of $\mathbf{r_1}$,$\mathbf{r_2}$,$\mathbf{r_3}$ with respect to $\Delta\phi$ are

$$\begin{aligned} \frac{\partial \mathbf{r}_1}{\partial \Delta\phi} &= -[\mathbf{r}_1]_\times \\ \frac{\partial \mathbf{r}_2}{\partial \Delta\phi} &= -[\mathbf{r}_2]_\times \\ \frac{\partial \mathbf{r}_3}{\partial \Delta\phi} &= -[\mathbf{r}_3]_\times \end{aligned} \tag{34}$$

So the Jacobian matrix of $\mathbf{f}\left(\mathbf{R}\right)$ with respect to $\Delta\phi$ is

$$\mathbf{J_f}(\mathbf{R}) = \frac{\partial \mathbf{f}(\mathbf{R})}{\partial \Delta\phi} = -\begin{bmatrix} \mathbf{H}\mathbf{J}_{\mathbf{V}_2}(\mathbf{r}_1)[\mathbf{r}_1]_\times \\ \mathbf{H}\mathbf{J}_{\mathbf{V}_2}(\mathbf{r}_2)[\mathbf{r}_2]_\times \\ \mathbf{H}\mathbf{J}_{\mathbf{V}_2}(\mathbf{r}_3)[\mathbf{r}_3]_\times \end{bmatrix} \tag{35}$$

Given an initial value $\mathbf{R}_{\text{init}}$, the minimum of the MNMA cost function $E(\mathbf{R})$, i.e., the MF estimation result $\mathbf{R}^*$, can be obtained by using the LM optimization algorithm to minimize the function as follow

$$\mathbf{R}^* = \arg\min_{\mathbf{R}} E(\mathbf{R}) \tag{36}$$

It is known that $\mathbf{R}_0$, the initial value of $\mathbf{R}$, is an orthogonal matrix. Assume that after the n-th iteration, the resulting matrix $\mathbf{R}_n$ is orthogonal (based on the definition of $\exp([\Delta\phi_n]_\times)$, then it is ready to know that $\mathbf{R}_{n+1} = \exp([\Delta\phi_n]_\times)\mathbf{R}_n$ is also orthogonal. So the optimization result $\mathbf{R}^*$ is orthogonal.

3.4 Determination of the Initial Value

Define $\mathbf{R}_0 = \begin{bmatrix}\mathbf{r}_{1,init} & \mathbf{r}_{2,init} & \mathbf{r}_{3,init}\end{bmatrix}^{\mathrm{T}}$. In order to avoid the influence of the zero gradient point on the cost function in the optimization process and ensure the global optimality, we design a method to select the initial value $\mathbf{R}_{\text{init}}$ through selecting the direction of the three coordinate axes $\mathbf{r}_{1,init}, \mathbf{r}_{2,init}, \mathbf{r}_{3,init}$.

Define the candidate axis vector set as

$$\begin{aligned} Candidate = \{\mathbf{v} = \frac{\mathbf{v}'}{|\mathbf{v}'|} \Big| \mathbf{v}' = \begin{bmatrix} x & y & \pm 4 \end{bmatrix}^{\mathrm{T}} \text{or} \begin{bmatrix} x & \pm 4 & y \end{bmatrix}^{\mathrm{T}} \\ \text{or} \begin{bmatrix} \pm 4 & x & y \end{bmatrix}^{\mathrm{T}}, x = -3, -2, ..., 3, y = -3, -2, ..., 3\} \end{aligned} \tag{37}$$

The set *Candidate* contains a total of $7 \times 7 \times 6 = 294$ unit vectors, which are roughly evenly distributed on the unit sphere. The purpose of constructing such a *Candidate* set is to ensure that, for any case, there is at least one vector in the set *Candidate* in every convergence domain of minimum value of $E(\mathbf{r})$.

Firstly, for each member vector $\mathbf{v}$ in the set *Candidate*, the corresponding MNSA cost value $E(\mathbf{v})$ is computed, and the vector $\mathbf{v}_1$ with the lowest MNSA cost is selected as the X-axis direction vector $\mathbf{r}_{\mathbf{1},init}$, that is

$$\mathbf{r}_{\mathbf{1},init} = \mathbf{v}_1 = \arg\min_{\mathbf{v} \in Candidate} E(\mathbf{v}) \tag{38}$$

Then, in the set *Candidate*, the vector $\mathbf{v}_2$ with the smallest MNSA cost value $E(\mathbf{v})$ is determined on the premise that the angle between $\mathbf{v}_1$ and $\mathbf{v}_2$ is between 60 and 120 degrees, that is

$$\mathbf{v}_2 = \arg\min_{\mathbf{v} \in Candidate, -\frac{1}{2} < \cos(\mathbf{v}_1 \cdot \mathbf{v}) < \frac{1}{2}} E(\mathbf{v}) \tag{39}$$

In order to ensure that $\mathbf{r}_{2,init}$ is perpendicular to $\mathbf{r}_{1,init}$, we define the following rules to determine $\mathbf{r}_{2,init}$

$$\mathbf{r}_{2,init} = \frac{\left(\mathbf{I} - \mathbf{v}_1 \mathbf{v}_1^{\mathrm{T}}\right)\mathbf{v}_2}{\left|\left(\mathbf{I} - \mathbf{v}_1 \mathbf{v}_1^{\mathrm{T}}\right)\mathbf{v}_2\right|} \tag{40}$$

where $\mathbf{I}$ is the identity matrix, $\left(\mathbf{I}-\mathbf{v}_1\mathbf{v}_1{}^{\mathrm{T}}\right)\mathbf{v}_2$ represents the vector after $\mathbf{v}_2$ removing the component of $\mathbf{v}_1$, $\frac{1}{\left|\left(\mathbf{I}-\mathbf{v}_1\mathbf{v}_1{}^{\mathrm{T}}\right)\mathbf{v}_2\right|}$ is the normalization factor. The vector $\mathbf{r}_{2,init}$ is a unit vector and coplanar with vectors $\mathbf{v}_2$ and $\mathbf{v}_1$. The construction of vector $\mathbf{r}_{2,init}$ is shown in Fig.6.

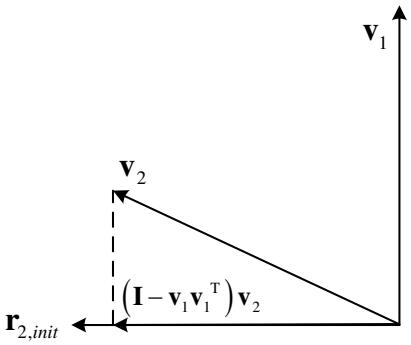


Figure 6: The Construction Process of $\mathbf{r}_{2,init}$

Finally, $\mathbf{r}_{3,init}$ is determined by the cross product of $\mathbf{r}_{1,init}$ and $\mathbf{r}_{2,init}$.

$$\mathbf{r}_{3,init} = \mathbf{r}_{1,init} \times \mathbf{r}_{2,init} \tag{41}$$

3.5 Complexity Analysis of the Proposed Method

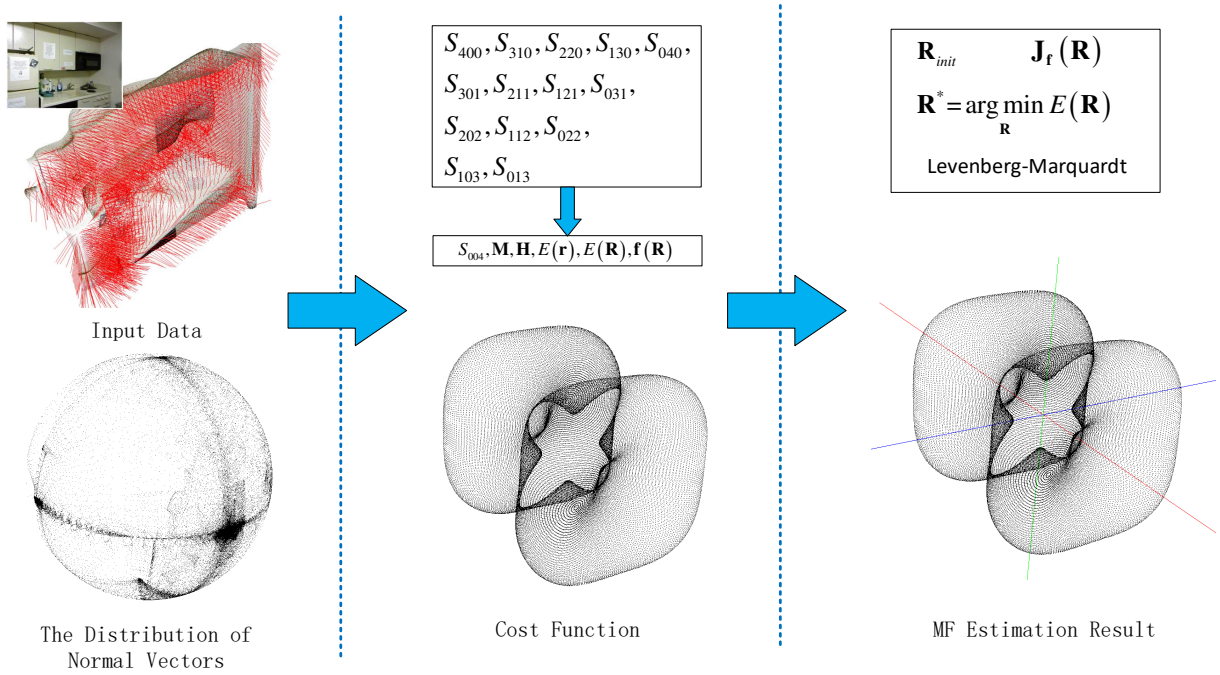


Figure 7: An illustration of the Proposed MF Estimation Algorithm.

A graphical illustration of the flow of the proposed method is shown in Fig.7. The left panel represents the input data and the corresponding normal vectors. The middle panel represents the determination of the cost function. In this process, 14 scalar parameters shown in (25) are computed from the normal vector set, then the 15th scalar parameter S_{004} is calculated, and the parameter matrix $\mathbf{M}$ and $\mathbf{H}$ are determined by the 15 scalar parameters. The MNSA cost function $E(\mathbf{r})$, the MNMA cost function $E(\mathbf{R})$ and the corresponding function $\mathbf{f}(\mathbf{R})$ are obtained. The right panel represents the optimization of MF rotation matrix. In this process, the initial value $\mathbf{R}_{init}$ is determined first, and the MF rotation matrix $\mathbf{R}$ is optimized by the Levenberg-Marquardt Algorithm based on the MNMA cost function $E(\mathbf{R})$, $\mathbf{f}(\mathbf{R})$ and the corresponding Jacobian matrix $\mathbf{J}_{\mathbf{f}}(\mathbf{R})$. The analysis of the global optimality of the proposed method is presented in Appendix B.

From equation (25), it can be seen that the complexity of the 14 scalar parameters computing process is O(n). The rest parts of the proposed method, namely, the parameter matrix $\mathbf{M}$, $\mathbf{H}$, and the optimization process of the MF rotation matrix $\mathbf{R}$, can be computed based on the 14 scalar parameters and do not need to use the original normal vector set, so the complexity is O(1). Thus, the overall complexity of the proposed method is O(n+1) which is linear.

4 Experimental Verification

To verify the performance of the proposed method, three groups of experiments are carried out. In Experiment 1, the accuracy and real-time performance of the proposed algorithm are evaluated on a generated virtual dataset. In experiment 2, the performance of the proposed algorithm is evaluated using real-world datasets. In Experiment 3, the performance of proposed method for estimating gravity direction in Atlanta Word is evaluated using Bremen dataset. All the experiments are carried out on a laptop with the following configuration:

CPU: Intel Corei7-4710MQ CPU with 2.50GHZ

Memory: 8G DDR4 RAM

Graphics Card: NVIDIA GT940M 2G DDR3

System: Ubuntu 16.04

4.1 Experiment 1: Performance Verification in Virtual Datasets

In this experiment, three groups of virtual datasets are generated to evaluate: 1) the accuracy of the proposed algorithm under different data dispersions and outlier ratios, and 2) the real-time performance under different data size.

In the process of dataset generation, 100 values are randomly selected in Lie Group SO(3) as the MF rotation matrix ground truth $\mathbf{R}_{gt,k}, k = 1, 2, ..., 100$. For each $\mathbf{R}_{gt,k}$, six directions, corresponding to the positive and negative directions of three axes, are taken as the distribution center, and n_1 normal vectors are randomly generated according to the vMF (von Mises-Fisher) (Ulrich, 1984) distribution law. The distribution law of vMF is as follow:

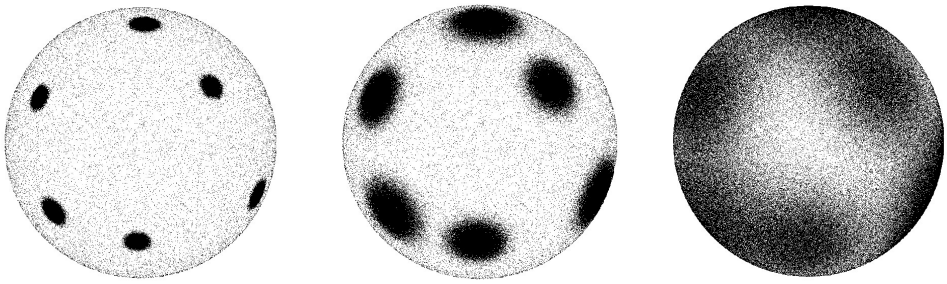
$$vMF(\mathbf{a}) = \frac{\kappa}{4\pi \sinh \kappa} e^{\kappa \mathbf{u}^{\mathrm{T}} \mathbf{a}} \tag{42}$$

where the unit vector $\mathbf{u}$ represents the distribution center of the sample normal vector, and κ describes the dispersion of the sample normal vector distribution relative to the distribution center. The smaller the value κ is, the more discrete the normal vector distribution is. Moreover, a number of n_2 additional unit vectors are generated uniformly and randomly to form the outlier component of the normal vector dataset.

In the experiments, the state-of-the-art MF estimation methods, EGI-BnB(Joo et al., 2018), MMF(Straub et al., 2014) and RTMF(Straub et al., 2015) are selected as baseline. For a fair comparison, we use the open source code provided by the original authors and use the default values of the associated parameters. The open source code of RTMF contains MMF algorithm implemented by GPU, and the GPU version of MMF is used in the experiment 1(a) to 1(c). Since MMF generates multiple MF solutions, we select the one closest to the ground truth as its evaluation result.

4.1.1 Experiment 1(a): Performance Evaluation under Different Data Dispersions

Referring to (Joo et al., 2018), seven different values of κ are taken, making the values of κ^{-1} be 0.0012, 0.0025, 0.005, 0.01, 0.02, 0.04 and 0.08 respectively. Taking the number of the inliers n_1 as 300000, and the number of the outliers n_2 as 20000, then 700 normal vector datasets are generated. Some examples of the generated datasets are shown in Fig.8.



(a) κ^{-1}=0.0012 (b) κ^{-1}=0.01 (c) κ^{-1}=0.08

Figure 8: The Generated Datasets under Different Data Dispersion. With the increase of κ^{-1}, the normal vector distribution of each pole becomes more discrete.

Table 1: The Mean and SD Values of the Evaluation Results under Different Data Dispersion

κ^{-1}	**Proposed**		**EGI-BnB**		**RTMF**		**MMF**	
	Mean(°)	***SD(°)***	***Mean(°)***	***SD(°)***	***Mean(°)***	***SD(°)***	***Mean(°)***	***SD(°)***
0.0012	0.009	0.0024	1.362	0.334	4.668	8.443	3.146	6.942
0.0025	0.011	0.003	1.356	0.332	5.28	8.483	3.836	8.004
0.005	0.0138	0.0038	1.363	0.328	6.393	9.146	3.325	5.598
0.01	0.0181	0.0048	1.388	0.374	8.264	10.405	3.633	3.306
0.02	0.0249	0.0059	1.562	0.794	11.097	11.954	5.631	4.098
0.04	0.0366	0.0069	1.821	1.433	15.109	13.211	8.264	4.372
0.08	0.0606	0.0119	2.789	5.97	20.494	13.473	12.484	6.244

The proposed method, together with EGI-BnB(Joo et al., 2018), MMF(Straub et al., 2014) and RTMF(Straub et al., 2015), is applied to the above generated data. For all the four methods, the MF is estimated with the generated normal vector datasets as input. Let $\mathbf{R}_{est,k}$ be the rotation matrix estimation result of the test methods. The error angle between $\mathbf{R}_{est}$ and the ground truth $\mathbf{R}_{gt,k}$ is calculated as follow

$$\boldsymbol{\theta}_k = \arccos\left(\max\left(abs\left(\mathbf{R}_{gt,k}^{\mathrm{T}}\mathbf{R}_{est,k}\right)\right)\right) \tag{43}$$

$$\boldsymbol{\theta}_k = \begin{bmatrix}\theta_{x,k} & \theta_{y,k} & \theta_{z,k}\end{bmatrix}^{\mathrm{T}} \tag{44}$$

$$\theta_{Avg,k} = \frac{\theta_{x,k} + \theta_{y,k} + \theta_{z,k}}{3} \tag{45}$$

where $\theta_{x,k}, \theta_{y,k}, \theta_{z,k}$ are the angular errors of X, Y and Z axes of the MF estimation result. Abs() represents the absolute value operation for each element of the input matrix. Max() represents the maximum value operation for each row of the input matrix. Arccos() represents the arccosine operation for each matrix element. $\theta_{Avg,k}$ is the average of the angular errors of the three axes.

The mean value $\theta_{Avg,mean}$ and standard deviation (SD) $\theta_{Avg,SD}$ of $\theta_{Avg,k}$ are calculated as follow

$$\theta_{Avg,mean} = \frac{1}{100} \sum_{k=1}^{100} \theta_{Avg,k} \tag{46}$$

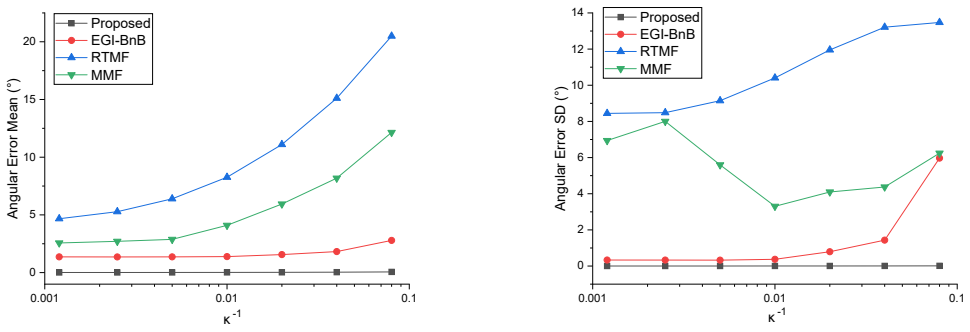
$$\theta_{Avg,SD} = \frac{1}{100} \sqrt{\sum_{k=1}^{100} \left(\theta_{Avg,k} - \theta_{Avg,mean} \right)^2} \tag{47}$$

The smaller the value of $\theta_{Avg,mean}$, the higher the accuracy. The smaller the value of $\theta_{Avg,SD}$, the higher the stability.

The evaluation results of angular error mean $\theta_{Avg,mean}$ and SD $\theta_{Avg,SD}$ in degree are shown in TABLE 1, and the corresponding line charts are shown in Fig.9. As an example, the curve surfaces of the MNSA cost function $E(\mathbf{r})$ and the X (red), Y (green), and Z (blue) axes of the corresponding MF estimation results, for the three cases of κ^{-1}=0.0012, 0.01 and 0.08, are shown in Fig.10.

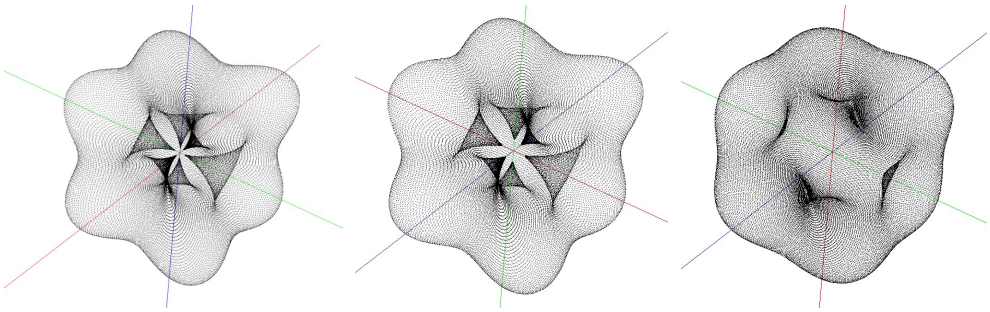
From TABLE 1 and Fig.9, it can be seen that the proposed method shows better accuracy and stability than the three compared methods under different data dispersion.

In Fig.10, it can be seen that with the increase of the dispersion of normal vector distribution, the minimum value of the cost function increases gradually, and the curve surface of the cost function tends to be flat gradually. Even so, the cost function still has obvious minimum value and sufficient convergence region in the corresponding MF axis direction, and each convergence region occupies the angular area covered by one face of the cube. Therefore, the proposed method produces stable and robust MF estimation under different data dispersions.



(a) $\theta_{Avg,mean}$ (b) $\theta_{Avg,SD}$

Figure 9: The Mean and SD Curves of Evaluation Results under Different Data Dispersion



(a) κ^{-1}=0.0012 (b) κ^{-1}=0.01 (c) κ^{-1}=0.08

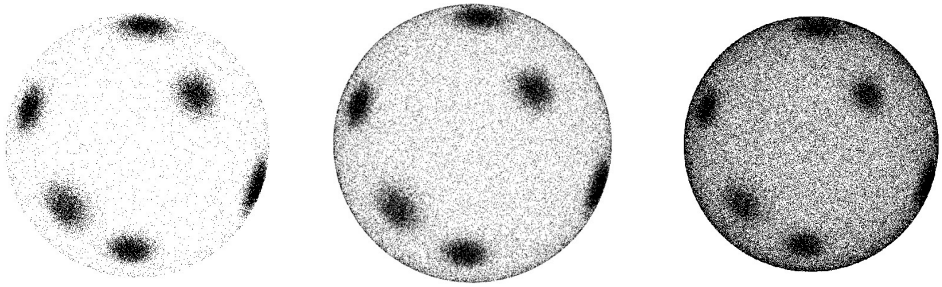
Figure 10: The Curve Surfaces of the MNSA Cost Function and Corresponding MF Estimation Results under Different Data Dispersion

4.1.2 Experiment 1(b): Performance Evaluation under Different Outlier Rates

In this experiment, the number of the inlier normal vectors is 30000, the dispersion parameter κ of inliers is 128, and the outlier rate η are chosen to be 10%, 20%, ... , 80% respectively. The number of outlier normal vectors is calculated as follows

$$outliers = inliers \times \frac{\eta}{1-\eta} \tag{48}$$

In this experiment, a total of 800 normal vector sets are generated. Some examples of the sets are shown in Fig.11.



(a) η=10% (b) η=50% (c) η=80%

Figure 11: The Generated Datasets under Different Outlier Rate

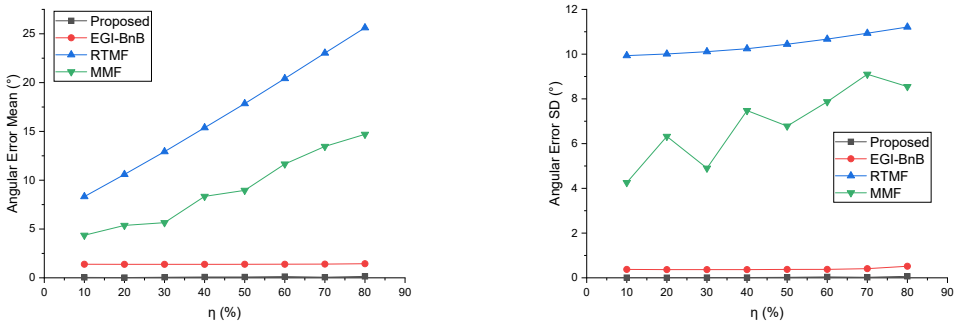
Similar to experiment 1(a), EGI-BnB, RTMF, MMF, together with the proposed method, are considered. Based on (43)-(47), the accuracy metric $\theta_{Avg,mean}$ and stability metric $\theta_{Avg,SD}$ under different outlier rates are calculated. The evaluation results of angular error mean $\theta_{Avg,mean}$ and SD $\theta_{Avg,SD}$ in degree are shown in TABLE 2, and the corresponding line charts are shown in Fig.12. As an example, the curve surfaces of the MNSA cost function $E(\mathbf{r})$ and the X (red), Y (green), and Z (blue) axes of the corresponding MF estimation results, for the case of η=10%, 50% and 80%, are shown in Fig.13.

It can be seen from TABLE 2 and Fig.12 that the proposed method has better accuracy and stability under different outlier rates.

From Fig.13, it can be seen that with the increase of the outlier rate η, the minimum value of the MNSA cost function gradually increases, and the curve surface of the cost function gradually flattens. When η = 80%, the minimum neighborhood of the curve surface of the cost function changes from concave to plane. Even so, the cost function still has obvious minimum values and sufficient convergence region in the corresponding MF axis directions, while each convergence area of MF axis direction occupies the angular area covered by one face of the cube. At η= 80%, the polar radius corresponding to the direction of MF axes are still a local minimum. All these show that the proposed method produces stable and robust MF estimation under different outlier rates.

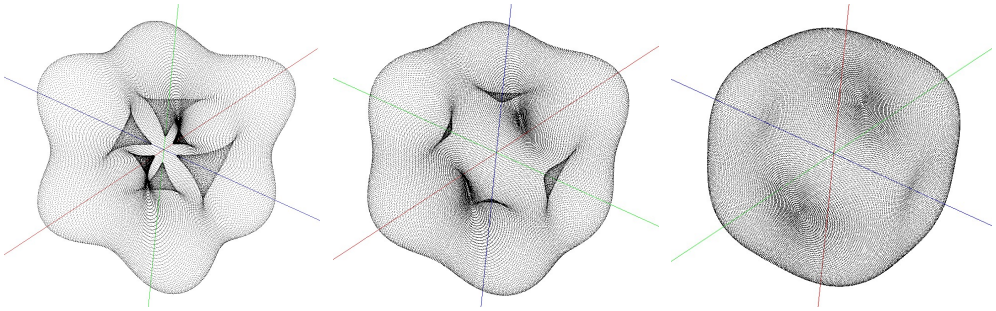
Table 2: The Mean and SD Values of the Evaluation Results under Different Outlier Rate

Outlier Rate (%)	**Proposed**		**EGI-BnB**		**RTMF**		**MMF**	
	Mean(°)	***SD(°)***	***Mean(°)***	***SD(°)***	***Mean(°)***	***SD(°)***	***Mean(°)***	***SD(°)***
10	0.0442	0.0076	1.389	0.375	8.32	9.934	4.36	4.259
20	0.0274	0.0048	1.385	0.368	10.593	10.008	5.371	6.322
30	0.0573	0.0121	1.385	0.368	12.925	10.111	5.65	4.901
40	0.0856	0.0131	1.384	0.368	15.378	10.243	8.343	7.478
50	0.0841	0.0212	1.39	0.376	17.845	10.443	8.963	6.783
60	0.128	0.0406	1.39	0.377	20.416	10.672	11.656	7.87
70	0.0626	0.0244	1.406	0.41	23.026	10.935	13.462	9.101
80	0.148	0.0675	1.445	0.517	25.617	11.212	14.701	8.549



(a) $\theta_{Avg,mean}$ (b) $\theta_{Avg,SD}$

Figure 12: The Mean and SD Curves of Evaluation Results under Different Outlier Rate



(a) η=10% (b) η=50% (c) η=80%

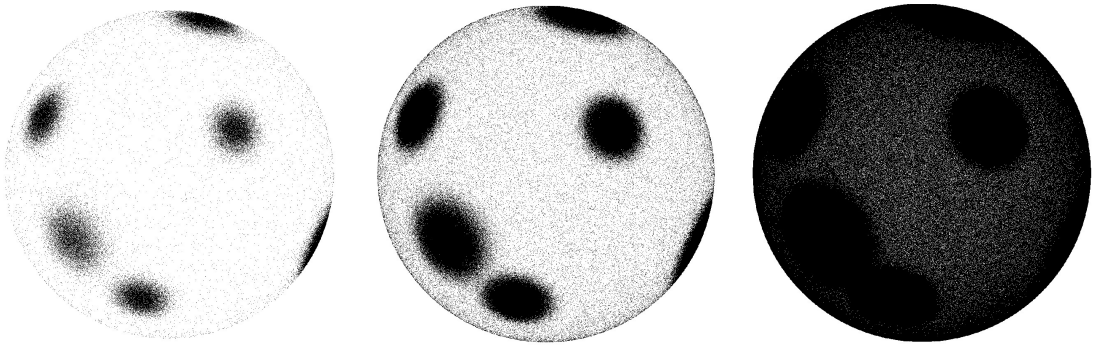
Figure 13: The Curve Surfaces of the MNSA Cost Function and Corresponding MF Estimation Results under Different Outlier Rate

4.1.3 Experiment 1(c): Real-Time Performance Evaluation under Different Data Sizes

Set κ as 128, and the ratio of the inliers and outliers is 30:2. The normal vector sets are generated according to the six data size levels shown in TABLE 3, and three examples of the generated vector sets are shown in Fig.14.

Table 3: Data Size Levels of the Normal Vector Sets

Level	1	2	3	4	5	6
Data Size	96000	320000	1056000	3200000	10656000	32000000



(a) *Data Size* $= 9.6 \times 10^4$ (b) *Data Size* $= 3.2 \times 10^6$ (c) *Data Size* $= 3.2 \times 10^7$

Figure 14: The Generated Datasets under Different Data Size

The computational performance of the proposed method, EGI-BnB, RTMF and MMF are carried out at each data size level. Our method and EGI-BnB run on CPU, whilst RTMF and MMF run on GPU. The results are shown in TABLE 4, and the corresponding logarithmic scale line chart is shown in Fig. 15. Besides, the time consuming performance of the 14 scalar parameters S_{uvw} computing process and the computational load associated with the cost functions and the MF optimization are shown in TABLE 5 and Fig. 16.

From the experimental results of TABLE 4 and Fig. 15, it can be seen that the time consuming curve of our method is significantly lower than the other three methods if only CPU is used; it is especially obviously lower than RTMF and MMF which are accelerated by GPU. The experimental results show that our method has outstanding real-time performance.

It can be seen from TABLE 5 and Fig. 16 that the time consuming of the computing process of the 14 scalar parameters is increasing with the increase of the data size. ~~The complexity of this process is O(n).~~ The time consuming of the rest part of the proposed method is constant. ~~Because this process is only based on the 14 scalar parameters , the complexity is O(1). Thus, the overall complexity of the proposed method is O(n+1) which is linear.~~ Therefore, the overall computational complexity of the proposed method is O(n+1).

Table 4: The Time Consuming Results under Different Data Size

Data Size	Time Consuming(s)			
	Proposed	EGI-BnB	RTMF(GPU)	MMF(GPU)
96000	0.000979	0.0358	0.00348	0.00442
320000	0.00294	0.0616	0.00994	0.0114
1056000	0.00681	0.134	0.0316	0.035
3200000	0.0197	0.348	0.0924	0.1
10656000	0.0657	1.08	0.299	0.321
32000000	0.196	3.13	0.922	0.996

Table 5: The Time Consuming of Two Parts of the Proposed Method under Different Data Size

Data Size	Time Consuming(s)	
	S_{uvw}	The Rest
96000	0.000878	0.000101
320000	0.00284	0.000103
1056000	0.00672	0.0000898
3200000	0.0196	0.000103
10656000	0.0656	0.00011
32000000	0.196	0.000113

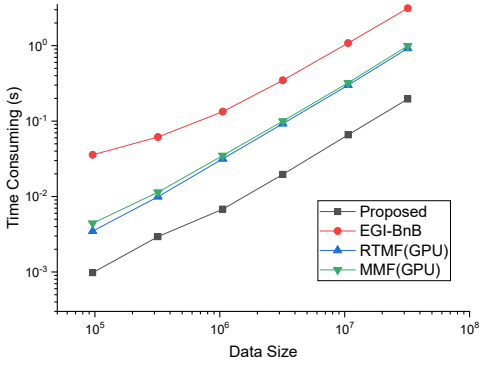


Figure 15: The Curves of Time Consuming under Different Data Size

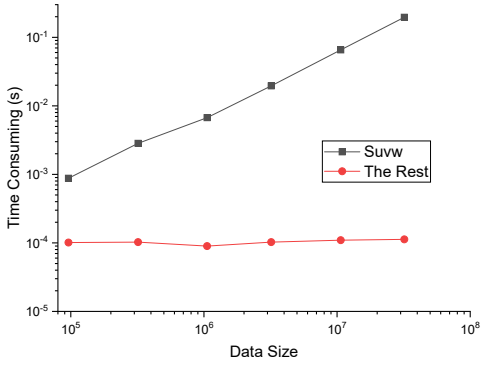


Figure 16: The Time Consuming Curves of Two Parts of the Proposed Method under Different Data Size

4.2 Experiment 2: Performance Evaluation in Real Word Datasets

Two experiments are carried out on NYUv2 dataset (Silberman et al., 2012) and self-captured RGB-D image sequences to verify the accuracy and real-time performance of the proposed algorithm.

4.2.1 Experiment 2(a): Performance Evaluation on the NYUv2 Dataset

The NYUv2 dataset contains 1449 RGB-D image samples of indoor scenes, four sample images of which are shown in Fig. 17. In each image sample of the dataset, we extract about 200000 to 300000 normal vectors as

input, and determine the MF ground truth according to the method proposed in RMFE (Robust Manhattan Frame Estimation)(Ghanem et al., 2015).

Figure 17: Some Example Images in NYUv2 Dataset

The corresponding angular error histogram and logarithmic scale time consuming histogram are shown in Fig.18. For our method, the computation of $\theta_x, \theta_y, \theta_z$ is as follow

$$\begin{aligned} \theta_x &= \frac{1}{1449}\sum_{k=1}^{1449}\theta_{x,k} \\ \theta_y &= \frac{1}{1449}\sum_{k=1}^{1449}\theta_{y,k} \\ \theta_z &= \frac{1}{1449}\sum_{k=1}^{1449}\theta_{z,k} \end{aligned} \tag{49}$$

where k is the index of the sample images. $\theta_{x,k}, \theta_{y,k}, \theta_{z,k}$ represent the angular error on X, Y and Z axis between MF estimation result and ground truth at k-th sample. θ_{Avg} is computed by (46).

We compare our method with other state-of-the-art methods including MPE (Main Plane Estimation) (Taylor and Cowley, 2013), MMF (Straub et al., 2014) , ES (Exhaustive Search) (Silberman et al., 2012) , RMFE (Ghanem et al., 2015) , RTMF (Straub et al., 2015) , Exhaustive BnB (Joo et al., 2018) and EGI-BnB (Joo et al., 2018). The details of the results are shown in TABLE 6. Some examples of the curve surfaces of MNSA cost functions and X (red), Y (green) and Z (blue) axis of MF estimation results in NYUv2 dataset are shown in 19.

From the experimental results of TABLE 6 and Fig.18, it can be seen that the angular error of the proposed method is within 2.0-2.5 degrees, which is similar to the Exhaustive BnB and EGI-BnB, and θ_z and θ_{Avg} are the lowest among all the methods involved in the comparison. Compared with the virtual datasets (in experiment 1) which only contains unbiased noise, the real dataset here contains more biased noise. Due to the influence of biased noise, the proposed method cannot achieve the same accuracy for the NYUv2 dataset as that obtained for the virtual datasets.

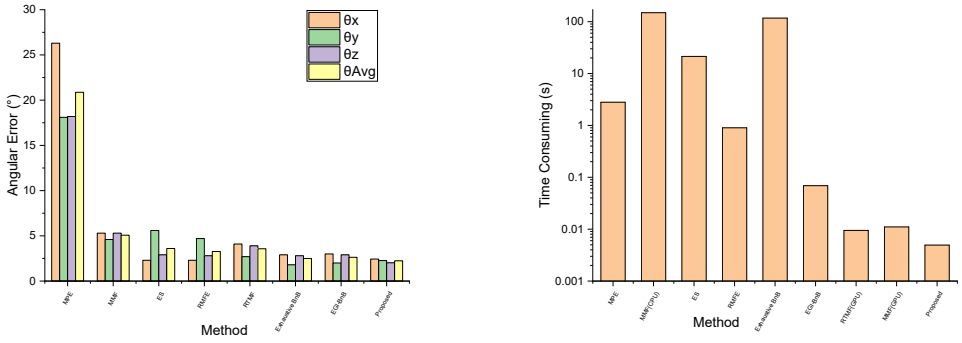
The average time consuming of our method is 4.93 ms, which is far less than 68.5 ms of EGI-BnB, and much less than 9.48 ms of RTMF and 11ms of MMF (GPU). Among all the methods involved in the comparison, the real-time performance of our method is much better than the compared methods.

The experimental results show that, compared with other MF estimation methods, the proposed method can produce more accurate estimation results and shows outstanding real-time performance for the NYUv2 dataset.

It can be seen from Fig.19 that the curve surface shape of MNSA cost function changes with the scene structure in NYUv2 dataset. Even so, the MNSA cost function still has obvious minimum values and sufficient convergence regions in the corresponding MF axis directions, and the MF can be estimated stably and accurately by our method.

Table 6: The Results of Angular Error and Time Consuming in NYUv2 Dataset

Method	MPE	MMF(CPU/GPU)	ES	RMFE	RTMF(GPU)	Exhaustive BnB	EGI-BnB	Proposed
θ_x (°)	26.3	5.3	**2.3**	**2.3**	4.1	2.9	3.0	2.435
θ_y (°)	18.1	4.6	5.6	4.7	2.7	**1.8**	2.0	2.265
θ_z (°)	18.2	5.3	2.9	2.8	3.9	2.8	2.9	**2.020**
θ_{Avg} (°)	20.87	5.07	3.6	3.27	3.57	2.5	2.63	**2.240**
Time Consuming(s)	2.8	148.7/0.011	21.4	0.9	0.0095	117.06	0.069	**0.0049**



(a) Angular Error (b) Time Consuming

Figure 18: The Histogram of the Angular Error and Time Consuming in NYUv2 Dataset

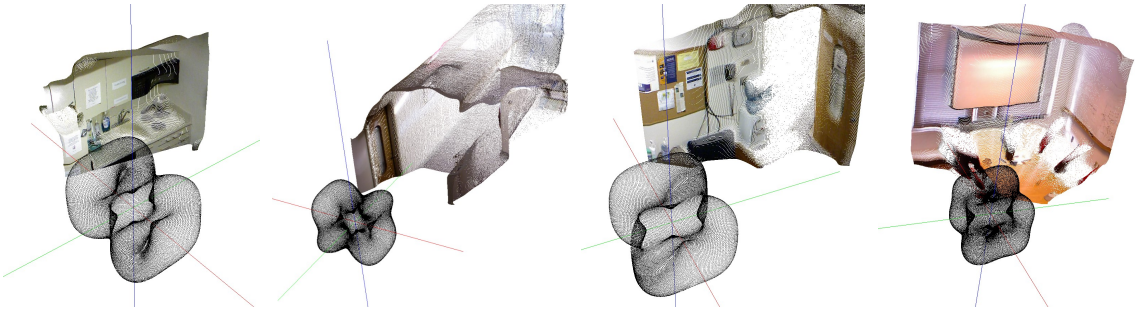


Figure 19: The Curve Surfaces of the MNSA Cost Function and Corresponding MF Estimation Results in NYUv2 Dataset

4.2.2 Experiment 2(b): Performance Evaluation in Self-captured Datasets

This experiment is carried out on a RGB-D image sequence captured in a self-structured Manhattan indoor scene to test the algorithm performance. In this scene, we put some boxes, boards and other items on the ground according to the Manhattan World pattern, and add some irregular objects such as cylinders and stools as noise. We use an RGB-D camera with a resolution of 640 * 480 to take the image sequence. Four sample images are shown in Fig.20. In order to obtain the MF ground truth of each frame, we use Optitrack localization system to record the camera pose, and set the scene coordinate system of the Optitrack coincidence with the MF of the scene.

Figure 20: Some Example Images in RGB-D Image Sequence

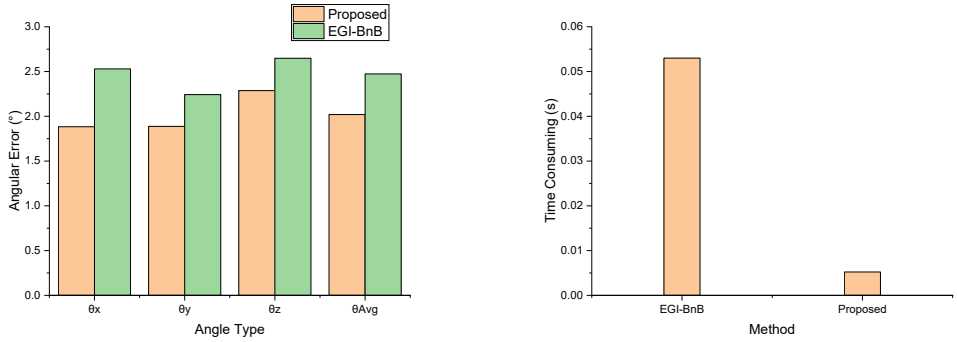
Firstly, based on each image's depth information, the point cloud is generated and the normal vectors are extracted. With the normal vectors as the input, the MFs are estimated and the average angular errors $\theta_x, \theta_y, \theta_z$and θ_{Avg} between the estimated MF and ground truth are determined. In this experiment, EGI-BnB is selected as the comparison benchmark. The results are shown in TABLE 7, from which it is clear that the proposed method outperforms the EGI-BnB method used in (Joo et al., 2018) for dealing with RGB-D images. The corresponding histogram is shown in Fig.21. Some examples of the curve surfaces of MNSA cost function $E(\mathbf{r})$ and X (red), Y (green) and Z (blue) axis of MF estimation results are shown in Fig.22.

From Fig.22, it can be seen that although there are various irregular objects in the scene, the MNSA cost function curve surface has obvious minimum values and sufficient convergence regions in the corresponding MF axis directions, and there are enough convergence regions near the minimum values of the cost function. The proposed method can still accurately estimate MF.

The experimental results show that the proposed method shows good accuracy and real-time performance for MF estimation in the RGB-D image sequences.

Table 7: The Angular Error and Time Consuming in Self-Captured Image Sequence

Method	Proposed	EGI-BnB
θ_x (°)	1.88	2.53
θ_y (°)	1.89	2.24
θ_z (°)	2.29	2.65
θ_{Avg} (°)	2.02	2.47
Time Consuming(s)	0.00522	0.053



(a) Angular Error

(b) Time Consuming

Figure 21: The Histogram of the Angular Error and Time Consuming in Self-Captured Image Sequence

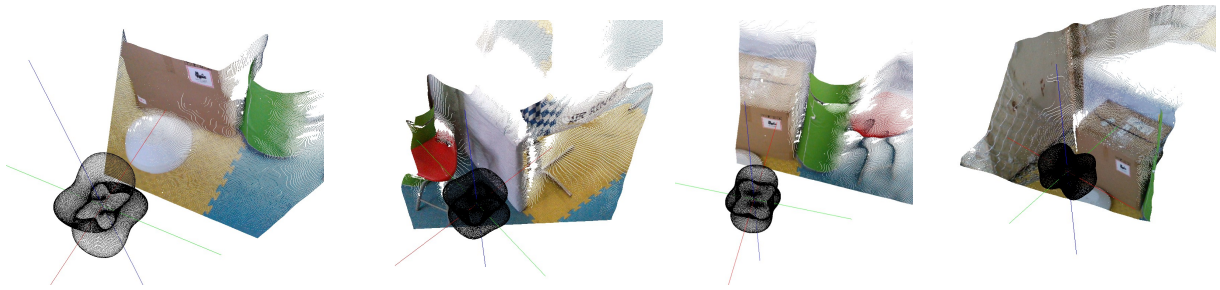


Figure 22: The Curve Surfaces of the MNSA Cost Function and Corresponding MF Estimation Results in Self-Captured Image Sequence

4.3 Experiment 3: Performance Evaluation of Gravity Direction Estimation in Atlanta World

In some cases, the artificial scenes do not strictly conform to the Manhattan World model, but the Atlanta World model. Different from Manhattan World, for Atlanta World pattern, the distribution of normal vectors in the horizontal direction is arbitrary, but all normal vectors are vertical or parallel to the gravity direction. The schematic diagram of Manhattan World and Atlanta World is shown in Fig.23 (Straub et al., 2017).

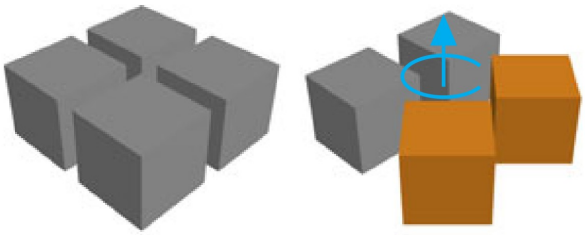


Figure 23: Manhattan World (Left) and Atlanta World (Right) (Straub et al., 2017)

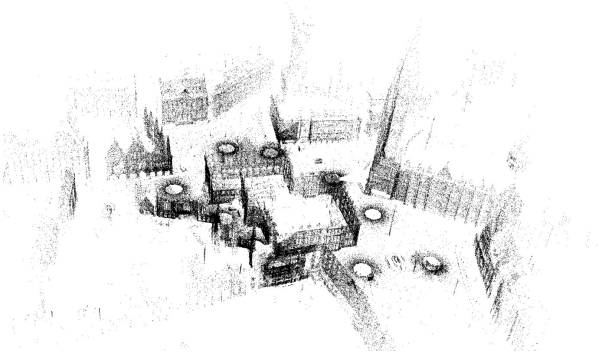


Figure 24: Bremen Point Cloud Dataset

For some tasks, it is not necessary to calculate all the horizontal directions in the Atlanta World, but only the gravity direction (Straub et al., 2017). In order to verify the performance of the proposed method in estimating gravity direction of Atlanta World, the point cloud dataset of Bremen is adopted, as shown in Fig.24, which is freely obtained from Robotic 3D scan repository (http://kos.informatik.uni-osnabrueck.de/3Dscans/). The Bremen dataset is collected by a laser scanner in 99 frames, which contains about 81 million 3D points and 99 laser scanner pose.

In the Bremen dataset, by manually selecting 3D points on the ground, we determined the ground truth of the gravity vector. Then, we extracted about 200000 normal vectors from the point cloud as the input of gravity direction estimation, and evaluate the accuracy and time consuming performance of our method.

For an ideal Atlanta World, the distribution of its normal vectors are all perpendicular or parallel to the gravity vector, so the MNSA cost function value of the gravity vector is zero. Unless it is Manhattan World, there is no orthogonal principal direction of the normal vectors in the horizontal direction, so the MNSA cost function value corresponding to any horizontal direction is not zero; in fact, it can be far greater than that in the gravity direction. Therefore, in this experiment, in the three output MF axes of the proposed method,

the axis with the least value of MNSA cost function is defined as the Z-axis, whilst the estimated gravity direction of Atlanta World, and the other two axes are defined as the X-axis and Y-axis according to the right-hand rule.

In this experiment, the EGI-BnB (Joo et al., 2018) and the state-of-the-art gravity vector estimation method of Liu et al. (2020) for Atlanta word, dubbed GOVDE (Globally Optimal Vertical Direction Estimation) in this paper, are selected as the baseline; we evaluate the accuracy performance and time consuming of our method on the Bremen dataset. For EGI-BnB, we calculate all the angular errors between the three output MF axes and the ground truth gravity vector, and take the minimum value as the angular error of gravity estimation.

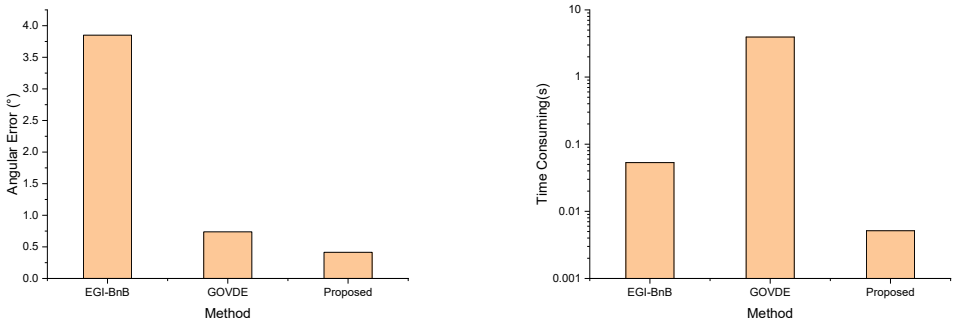
The experimental results are shown in TABLE 8 and the corresponding histogram is shown in Fig.25. The curve surface (red point cloud) of the MNSA cost function for the Bremen dataset, and the output along the X (red line), Y (green line) and Z (blue line) axes of our method are shown in Fig.26, where the Z-axis represents the gravity direction estimated by our method. The four views (oblique view, top view, front view and side view) of the MNSA cost function curve surface are shown in Fig.27.

It can be seen from Fig.26 that the minimum direction of the MNSA cost function is parallel to the gravity direction of the Bremen point cloud. From the four views of the MNSA cost function curve surface in Fig.27, it can be seen that the MNSA cost value of gravity direction is about 0.4 times of the MNSA cost value of the MF horizontal axes. The results show that, for the Bremen dataset, it is reasonable to find out the gravity direction of Atlanta World by the strategy of finding the axis with the least MNSA cost value.

The experimental results show that, for Atlanta World, the proposed method shows outstanding accuracy and real-time performance for gravity direction estimation task.

Table 8: The Angular Error and Time Consuming of Gravity Direction Estimation in Bremen Dataset

Method	EGI-BnB	GOVDE	Proposed
Angular Error(°)	3.85	0.738	0.414
Time Consuming(s)	0.0532	3.94	0.00514



(a) Angular Error

(b) Time Consuming

Figure 25: The Histogram of the Angualr Error and Time Consuming of Gravity Direction Estimation in Bremen Dataset

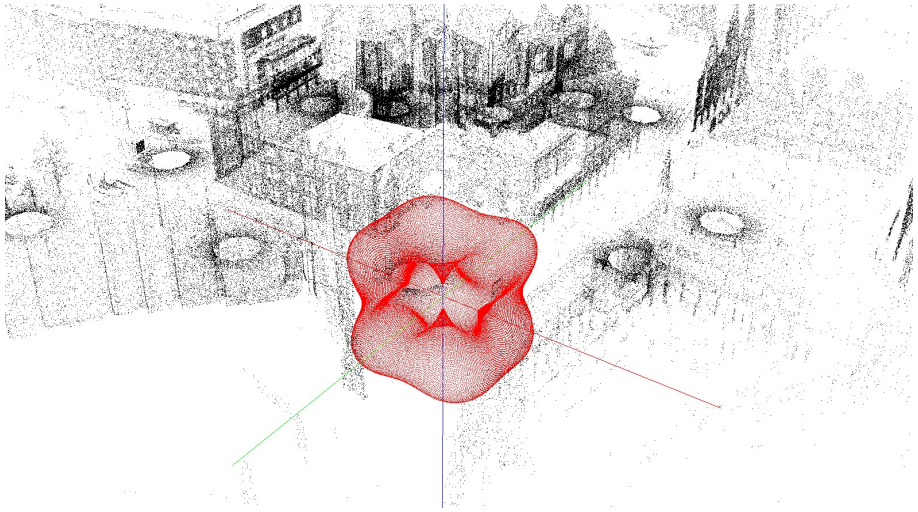
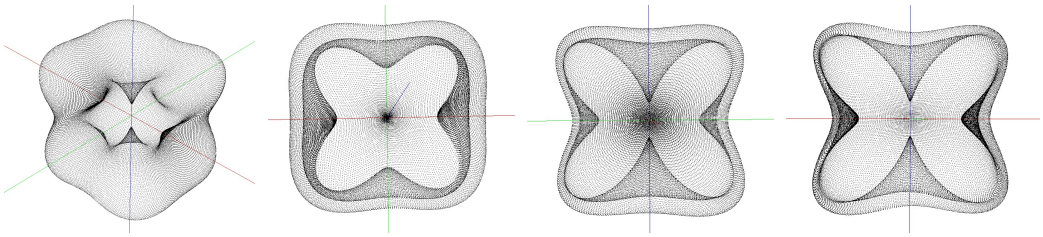


Figure 26: The Curve Surface (Red Point Cloud) of MNSA Cost Function and the Gravity Estimation Result (Blue Line) of Our Method in Bremen Dataset



(a) Oblique View (b) Top View (c) Front View (d) Side View

Figure 27: The Four View of the MNSA Cost Function Curve Surface in Bremen Dataset

5 Conclusion

To improve the real-time performance of MF estimation while ensure a high accuracy, a novel fast MF estimation method is proposed. Based on the trigonometric function, the cost function of Single Normal vector and Single MF Axis (SNSA) are designed, and the cost function of Multiple Normal vectors and Single MF Axis (MNSA) are defined as the arithmetic mean of the SNSA cost function of every normal vector. Then, the MNSA cost function is simplified utilizing vector dot and cross product operations, resulting in a simplified MNSA cost function where the normal vector set only contains 14 scalar parameters and the associated computational complexity is O(n). Finally, the cost function of MF rotation matrix, the initial value determination and the optimization method are given, whose computational complexity is O(1).

The accuracy and real-time performances of the proposed method are evaluated using three groups of experiments and compared with several state-of-the-art MF estimation methods. The experimental results in experiment 1 show that the proposed method performs excellently in terms of accuracy and computational speed for virtual datasets containing unbiased noise. In experiment 2, the experimental results show that the proposed method performs excellently in terms of computational efficiency for real-world datasets containing biased noise, and meanwhile ensures that the accuracy is comparable to the state-of-the-art methods. For the Bremen dataset in experiment 3, the proposed method shows outstanding real-time performance and high accuracy for estimating the gravity direction in Atlanta World.

The innovative design and the good properties shown by the proposed method on the benchmark tests set it aside from the state-of-the-art methods, in that it enables more effective and efficient MF estimation.

Appendix A Detailed Derivation of Some Formulas

The detailed derivation process of formula (10) is as follows

$$
\begin{aligned}
&[\mathbf{a}_i]_\times \mathbf{r}\mathbf{r}^{\mathrm{T}}\mathbf{a}_i \\
&= \begin{bmatrix} 0 & -c_i & b_i \\ c_i & 0 & -a_i \\ -b_i & a_i & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \begin{bmatrix} x & y & z \end{bmatrix} \begin{bmatrix} a_i \\ b_i \\ c_i \end{bmatrix} \\
&= \begin{bmatrix} 0 & -c_i & b_i \\ c_i & 0 & -a_i \\ -b_i & a_i & 0 \end{bmatrix} \begin{bmatrix} x^2 & xy & xz \\ xy & y^2 & yz \\ xz & yz & z^2 \end{bmatrix} \begin{bmatrix} a_i \\ b_i \\ c_i \end{bmatrix} \\
&= \begin{bmatrix} 0 & -c_i & b_i \\ c_i & 0 & -a_i \\ -b_i & a_i & 0 \end{bmatrix} \begin{bmatrix} a_i x^2 + b_i xy + c_i xz \\ a_i xy + b_i y^2 + c_i yz \\ a_i xz + b_i yz + c_i z^2 \end{bmatrix} \\
&= \begin{bmatrix} 0 & -c_i & b_i \\ c_i & 0 & -a_i \\ -b_i & a_i & 0 \end{bmatrix} \begin{bmatrix} a_i & 0 & 0 & b_i & c_i & 0 \\ 0 & b_i & 0 & a_i & 0 & c_i \\ 0 & 0 & c_i & 0 & a_i & b_i \end{bmatrix} \begin{bmatrix} x^2 \\ y^2 \\ z^2 \\ xy \\ xz \\ yz \end{bmatrix} \\
&= \begin{bmatrix} 0 & -b_i c_i & b_i c_i & -a_i c_i & a_i b_i & {b_i}^2 - {c_i}^2 \\ a_i c_i & 0 & -a_i c_i & b_i c_i & {c_i}^2 - {a_i}^2 & -a_i b_i \\ -a_i b_i & a_i b_i & 0 & {a_i}^2 - {b_i}^2 & -b_i c_i & a_i c_i \end{bmatrix} \begin{bmatrix} x^2 \\ y^2 \\ z^2 \\ xy \\ xz \\ yz \end{bmatrix}
\end{aligned}
\tag{A.1}
$$

The detailed derivation process of formula (14) is as follows

$$
\begin{aligned}
E_i(\mathbf{r}) &= \sin^2 < \mathbf{r}, \mathbf{a}_i > \cos^2 < \mathbf{r}, \mathbf{a}_i > \\
&= |\sin < \mathbf{r}, \mathbf{a}_i > \cos < \mathbf{r}, \mathbf{a}_i >|^2 \\
&= \left| [\mathbf{a}_i]_\times \mathbf{r}\mathbf{r}^{\mathrm{T}}\mathbf{a}_i \right|^2 \\
&= |\mathbf{A}(\mathbf{a}_i)\mathbf{V}_2(\mathbf{r})|^2 \\
&= {\mathbf{V}_2}^{\mathrm{T}}(\mathbf{r})\mathbf{A}^{\mathrm{T}}(\mathbf{a}_i)\mathbf{A}(\mathbf{a}_i)\mathbf{V}_2(\mathbf{r})
\end{aligned}
\tag{A.2}
$$

Appendix B Analysis of the Global Optimality

From (4), we can see that the cost function value is related to the angles between the axes of coordinate frame $\mathbf{R}$ to be optimized and the normal vector set $\{\mathbf{a}_i\}$, and independent with the selection of the Reference Coordinate Frame (RCF). For the convenience of description, the MF corresponding to the vector set $\{\mathbf{a}_i\}$ is selected as the RCF.

When the three coordinate axes of the Coordinate Frame to be Optimized (CFO) coincide with the three coordinate axes of the RCF, the attitude matrix $\mathbf{R}$ of the CFO is at a stationary point of the MNMA cost function. We intend to explain that, in this case, the global minimum of the MNMA cost value is obtained.

Since the MNSA cost function has central symmetry, and the three coordinate axes of CFO have the same mathematical status in the definition of MNMA cost function, no matter which axis of the RCF coincides with the X, Y, or Z axis of the CFO, whether the direction of any coordinate axis of CFO is the same as or opposite to the corresponding axis of the RCF, the value of MNMA cost value is the same. Therefore, in the following analysis, $\mathbf{R} = \mathbf{I}$ is taken as an example, where $\mathbf{I}$ is an identity matrix.

Combining (18) and (20), yields

$$\begin{aligned} E(\mathbf{r}) &= A_1x^4 + B_1y^4 + C_1z^4 \\ &+ A_2x^2y^2 + B_2x^2z^2 + C_2y^2z^2 \\ &+ 2A_3x^2yz + 2B_3xy^2z + 2C_3xyz^2 \\ &+ 2A_4x^3y + 2B_4x^3z + 2C_4y^3z \\ &+ 2A_5xy^3 + 2B_5xz^3 + 2C_5yz^3 \end{aligned} \tag{B.1}$$

where

$$\begin{aligned} A_1 &= S'_{220} + S'_{202} \\ B_1 &= S'_{220} + S'_{022} \\ C_1 &= S'_{202} + S'_{022} \\ A_2 &= S'_{400} + S'_{040} + S'_{202} + S'_{022} - 4S'_{220} \\ B_2 &= S'_{400} + S'_{004} + S'_{022} + S'_{220} - 4S'_{202} \\ C_2 &= S'_{040} + S'_{004} + S'_{202} + S'_{220} - 4S'_{022} \\ A_3 &= S'_{013} + S'_{031} - 5S'_{211} \\ B_3 &= S'_{103} + S'_{301} - 5S'_{121} \\ C_3 &= S'_{130} + S'_{310} - 5S'_{112} \\ A_4 &= S'_{130} - S'_{310} + S'_{112} \\ B_4 &= S'_{103} - S'_{301} + S'_{121} \\ C_4 &= S'_{013} - S'_{031} + S'_{211} \\ A_5 &= -S'_{130} + S'_{310} + S'_{112} \\ B_5 &= -S'_{103} + S'_{301} + S'_{121} \\ C_5 &= -S'_{013} + S'_{031} + S'_{211} \end{aligned} \tag{B.2}$$

In (B.2), $S'_{uvw} = \frac{1}{n}\sum_{i=1}^{n} a_{i,M}^{u} b_{i,M}^{v} c_{i,M}^{w}$ represents the 15 parameters S_{uvw} of $\{\mathbf{a}_i\}$ calculated in the MF as the RCF, where $a_{i,M}$,$b_{i,M}$,$c_{i,M}$ are the coordinates of the normal vector $\mathbf{a}_i$ in the MF.

In the Manhattan scene, assume that most of the normal vectors of $\{\mathbf{a}_i\}$ are uniformly distributed near the X, Y and Z axes of the MF, and a small number of vectors are distributed away from the MF axes in the form of noise. Therefore, the values of the non-negative parameters $S'_{400}, S'_{040}, S'_{004}$ are the three largest of the 15 parameters S'_{uvw}, and $A_2x^2y^2 + B_2x^2z^2 + C_2y^2z^2$ is the main component of the MNSA cost function (B.1).

For the parameters A_1, B_1, C_1, the related non-negative parameters $S'_{220}, S'_{202}, S'_{022}$ reflect the distribution proportion of the vector set $\{\mathbf{a}_i\}$ near the XOY plane, XOZ plane and YOZ plane and away from the MF axes. The greater the noise, the greater the A_1, B_1, C_1 values. We assume that most normal vectors in the

Manhattan scene are distributed near the MF axes, and the proportion of the noise normal vectors far away from the MF axes is very small, so $S'_{400}, S'_{040}, S'_{004}$ are much greater than $S'_{220}, S'_{202}, S'_{022}$, and A_2, B_2, C_2 are much greater than A_1, B_1, C_1 and positive.

$A_3, B_3, C_3, A_4, B_4, C_4, A_5, B_5, C_5$ are the parameters about the odd power of the normal vectors' coordinates in MF, which reflect the degree of non-uniformity of the noise normal vector distribution. If the noise normal vector distribution is absolutely uniform, the values of $A_3, B_3, C_3, A_4, B_4, C_4, A_5, B_5, C_5$ will be zero. It is assumed that in the Manhattan scene, the normal vectors near the Manhattan axes are uniformly distributed, and the normal vectors away from the Manhattan axis may be uneven, but the proportion is very small. Therefore, $A_3, B_3, C_3, A_4, B_4, C_4, A_5, B_5, C_5$ is much smaller than the parameters A_1, B_1, C_1 reflecting the proportional size of noise.

To sum up, the following assumption is made in the Manhattan scene:

$$A_2, B_2, C_2 \gg A_1, B_1, C_1 \gg A_3, B_3, C_3, A_4, B_4, C_4, A_5, B_5, C_5 \tag{B.3}$$

Let $\theta_z, \theta_y, \theta_x$ be the Euler angles of the CFO relative to the RCF in the order of Z-Y-X. The rotation matrix $\mathbf{R}$ between the CFO and the RCF is

$$\mathbf{R} = \begin{bmatrix} \cos\theta_z \cos\theta_y & -\sin\theta_z \cos\theta_x + \cos\theta_z \sin\theta_y \sin\theta_x & \sin\theta_z \sin\theta_x + \cos\theta_z \sin\theta_y \cos\theta_x \\ \sin\theta_z \cos\theta_y & \cos\theta_z \cos\theta_x + \sin\theta_z \sin\theta_y \sin\theta_x & -\cos\theta_z \sin\theta_x + \sin\theta_z \sin\theta_y \cos\theta_x \\ -\sin\theta_y & \cos\theta_y \sin\theta_x & \cos\theta_y \cos\theta_x \end{bmatrix} \tag{B.4}$$

When $\theta_x, \theta_y, \theta_z$ are very small, the rotation matrix $\mathbf{R}$ can be approximately expressed in the form of the second-order Taylor expansion as follow

$$\mathbf{R} = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 \end{bmatrix} = \begin{bmatrix} 1 - \frac{{\theta_z}^2}{2} - \frac{{\theta_y}^2}{2} & -\theta_z + \theta_y\theta_x & \theta_z\theta_x + \theta_y \\ \theta_z & 1 - \frac{{\theta_z}^2}{2} - \frac{{\theta_x}^2}{2} & -\theta_x + \theta_z\theta_y \\ -\theta_y & \theta_x & 1 - \frac{{\theta_y}^2}{2} - \frac{{\theta_x}^2}{2} \end{bmatrix} \tag{B.5}$$

where

$$\begin{aligned} \mathbf{r}_1 &= \begin{bmatrix} 1 - \frac{{\theta_z}^2}{2} - \frac{{\theta_y}^2}{2} & \theta_z & -\theta_y \end{bmatrix}^{\mathrm{T}} \\ \mathbf{r}_2 &= \begin{bmatrix} -\theta_z + \theta_y\theta_x & 1 - \frac{{\theta_z}^2}{2} - \frac{{\theta_x}^2}{2} & \theta_x \end{bmatrix}^{\mathrm{T}} \\ \mathbf{r}_3 &= \begin{bmatrix} \theta_z\theta_x + \theta_y & -\theta_x + \theta_z\theta_y & 1 - \frac{{\theta_y}^2}{2} - \frac{{\theta_x}^2}{2} \end{bmatrix}^{\mathrm{T}} \end{aligned} \tag{B.6}$$

$\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3$ represent the unit vectors of the CFO's X, Y and Z axis in the RCF.

The derivatives of the MNSA cost function at $\mathbf{r} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^{\mathrm{T}}$, $\mathbf{r} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^{\mathrm{T}}$, $\mathbf{r} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^{\mathrm{T}}$ are

$$\begin{aligned} \left.\frac{\partial E(\mathbf{r})}{\partial \mathbf{r}}\right|_{\mathbf{r}=\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^{\mathrm{T}}} &= \begin{bmatrix} 0 & 2A_4 & 2B_4 \end{bmatrix}^{\mathrm{T}} \\ \left.\frac{\partial E(\mathbf{r})}{\partial \mathbf{r}}\right|_{\mathbf{r}=\begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^{\mathrm{T}}} &= \begin{bmatrix} 2A_5 & 0 & 2C_4 \end{bmatrix}^{\mathrm{T}} \\ \left.\frac{\partial E(\mathbf{r})}{\partial \mathbf{r}}\right|_{\mathbf{r}=\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^{\mathrm{T}}} &= \begin{bmatrix} 2B_5 & 2C_5 & 0 \end{bmatrix}^{\mathrm{T}} \end{aligned} \tag{B.7}$$

Then, the derivative of MNMA cost function with respect to $\theta_x, \theta_y, \theta_z$ at $\theta_x, \theta_y, \theta_z = 0$ is

$$\begin{aligned}
&\left.\frac{\partial E\left(R_{rc}\left(\theta_x,\theta_y,\theta_z\right)\right)}{\partial\left[\theta_x,\theta_y,\theta_z\right]^{\mathrm{T}}}\right|_{\theta_x,\theta_y,\theta_z=0} \\
&= \left.\frac{\partial E\left(\mathbf{r}_1\right)}{\partial \mathbf{r}_1}\right|_{\mathbf{r}_1=\begin{bmatrix}1 & 0 & 0\end{bmatrix}^{\mathrm{T}}} \left.\frac{\partial \mathbf{r}_1}{\partial\left[\theta_x,\theta_y,\theta_z\right]^{\mathrm{T}}}\right|_{\theta_x,\theta_y,\theta_z=0} \\
&+ \left.\frac{\partial E\left(\mathbf{r}_2\right)}{\partial \mathbf{r}_2}\right|_{\mathbf{r}_2=\begin{bmatrix}0 & 1 & 0\end{bmatrix}^{\mathrm{T}}} \left.\frac{\partial \mathbf{r}_2}{\partial\left[\theta_x,\theta_y,\theta_z\right]^{\mathrm{T}}}\right|_{\theta_x,\theta_y,\theta_z=0} \\
&+ \left.\frac{\partial E\left(\mathbf{r}_3\right)}{\partial \mathbf{r}_3}\right|_{\mathbf{r}_3=\begin{bmatrix}0 & 0 & 1\end{bmatrix}^{\mathrm{T}}} \left.\frac{\partial \mathbf{r}_3}{\partial\left[\theta_x,\theta_y,\theta_z\right]^{\mathrm{T}}}\right|_{\theta_x,\theta_y,\theta_z=0} \\
&= \begin{bmatrix}2C_4 - 2C_5 & 2B_5 - 2B_4 & 2A_4 - 2A_5\end{bmatrix}
\end{aligned} \tag{B.8}$$

$\mathbf{R} = \mathbf{I}$ is a stationary point of the MNMA cost function, so the derivative (B.8) is a zero vector. Then we can get

$$\begin{aligned}
A_4 &= A_5 \\
B_4 &= B_5 \\
C_4 &= C_5
\end{aligned} \tag{B.9}$$

Then

$$\begin{aligned}
A_4 &= A_5 = S'_{112} \\
B_4 &= B_5 = S'_{121} \\
C_4 &= C_5 = S'_{211} \\
S'_{130} &= S'_{310} \\
S'_{103} &= S'_{301} \\
S'_{013} &= S'_{031}
\end{aligned} \tag{B.10}$$

So the MNSA cost function (B.1) can be simplified to

$$\begin{aligned}
E\left(\mathbf{r}\right) &= A_1x^4 + B_1y^4 + C_1z^4 \\
&+ A_2x^2y^2 + B_2x^2z^2 + C_2y^2z^2 \\
&+ 2A_3x^2yz + 2B_3xy^2z + 2C_3xyz^2 \\
&+ 2A_4xy\left(x^2+y^2\right) + 2B_4xz\left(x^2+z^2\right) + 2C_4yz\left(y^2+z^2\right)
\end{aligned} \tag{B.11}$$

Combining (B.6) and (B.11), and ignoring the small quantities of third order and above, we can obtain

$$
\begin{aligned}
E\left(\mathbf{r}_1\right) &\approx A_1 + 2A_4\theta_z - 2B_4\theta_y + \left(A_2 - 2A_1\right){\theta_z}^2 + \left(B_2 - 2A_1\right){\theta_y}^2 - 2A_3\theta_y\theta_z \\
E\left(\mathbf{r}_2\right) &\approx B_1 + 2C_4\theta_x - 2A_4\theta_z + \left(A_2 - 2B_1\right){\theta_z}^2 + \left(C_2 - 2B_1\right){\theta_x}^2 - 2B_3\theta_x\theta_z + 2A_4\theta_y\theta_x \\
E\left(\mathbf{r}_3\right) &\approx C_1 + 2B_4\theta_y - 2C_4\theta_x + \left(B_2 - 2C_1\right){\theta_y}^2 + \left(C_2 - 2C_1\right){\theta_x}^2 - 2C_3\theta_x\theta_y + 2B_4\theta_z\theta_x + 2C_4\theta_z\theta_y
\end{aligned} \tag{B.12}
$$

Combining (26) with (B.12), the MNMA cost function with respect to $\theta_x, \theta_y, \theta_z$ is

$$
\begin{aligned}
&E\left(\mathbf{R}\left(\theta_x, \theta_y, \theta_z\right)\right) \\
&\approx A_1 + B_1 + C_1 \\
&+ 2\left(C_2 - B_1 - C_1\right){\theta_x}^2 + 2\left(B_2 - A_1 - C_1\right){\theta_y}^2 + 2\left(A_2 - A_1 - B_1\right){\theta_z}^2 \\
&+ 2\left(A_4 - C_3\right)\theta_x\theta_y + 2\left(B_4 - B_3\right)\theta_x\theta_z + 2\left(C_4 - A_3\right)\theta_y\theta_z
\end{aligned} \tag{B.13}
$$

The Hessian matrix of MNMA cost function with respect to $\theta_x, \theta_y, \theta_z$ at $\theta_x, \theta_y, \theta_z = 0$ is

$$
\begin{aligned}
\mathbf{H}\left(E\left(\mathbf{R}\left(\theta_x, \theta_y, \theta_z\right)\right)\right)\big|_{\theta_x,\theta_y,\theta_z=\mathbf{0}} &= \left.\begin{bmatrix} \frac{\partial^2 E(\mathbf{R})}{\partial {\theta_x}^2} & \frac{\partial^2 E(\mathbf{R})}{\partial\theta_x\partial\theta_y} & \frac{\partial^2 E(\mathbf{R})}{\partial\theta_x\partial\theta_z} \\ \frac{\partial^2 E(\mathbf{R})}{\partial\theta_x\partial\theta_y} & \frac{\partial^2 E(\mathbf{R})}{\partial {\theta_y}^2} & \frac{\partial^2 E(\mathbf{R})}{\partial\theta_y\partial\theta_z} \\ \frac{\partial^2 E(\mathbf{R})}{\partial\theta_x\partial\theta_z} & \frac{\partial^2 E(\mathbf{R})}{\partial\theta_y\partial\theta_z} & \frac{\partial^2 E(\mathbf{R})}{\partial {\theta_z}^2} \end{bmatrix}\right|_{\theta_x,\theta_y,\theta_z=\mathbf{0}} \\
&= 2\begin{bmatrix} 2\left(C_2 - B_1 - C_1\right) & A_4 - C_3 & B_4 - B_3 \\ A_4 - C_3 & 2\left(B_2 - A_1 - C_1\right) & C_4 - A_3 \\ B_4 - B_3 & C_4 - A_3 & 2\left(A_2 - A_1 - B_1\right) \end{bmatrix}
\end{aligned} \tag{B.14}
$$

Based on the assumption (B.3), the Hessian matrix (B.14) is positive definite. So $\theta_x, \theta_y, \theta_z = 0$, and thus $\mathbf{R} = \mathbf{I}$, is a minimum point of the MNMA cost function.

In order to analyze whether $\mathbf{R} = \mathbf{I}$ is the global minimum point of the MNMA cost function, firstly, the distribution characteristic of MNSA cost function's stationary points is analyzed. For the convenience of description, the terms related to $A_3, B_3, C_3, A_4, B_4, C_4, A_5, B_5, C_5$ in MNSA function are ignored, then the MNSA cost function is simplified to

$$
E\left(\mathbf{r}\right) \approx A_1x^4 + B_1y^4 + C_1z^4 + A_2x^2y^2 + B_2x^2z^2 + C_2y^2z^2 \tag{B.15}
$$

Define

$$
E\left(\mathbf{r}, \lambda\right) = E\left(\mathbf{r}\right) + \lambda\left(x^2 + y^2 + z^2 - 1\right) \tag{B.16}
$$

Take the derivative of $E\left(\mathbf{r}, \lambda\right)$ and make its derivative zero, as follows

$$\begin{aligned}
\frac{\partial E\left(\mathbf{r},\lambda\right)}{\partial x} &= 2x\left(2A_1x^2+A_2y^2+B_2z^2\right)+\lambda\cdot 2x=0\\
\frac{\partial E\left(\mathbf{r},\lambda\right)}{\partial y} &= 2y\left(A_2x^2+2B_1y^2+C_2z^2\right)+\lambda\cdot 2y=0\\
\frac{\partial E\left(\mathbf{r},\lambda\right)}{\partial z} &= 2z\left(B_2x^2+C_2y^2+2C_1z^2\right)+\lambda\cdot 2z=0\\
\frac{\partial E\left(\mathbf{r},\lambda\right)}{\partial \lambda} &= x^2+y^2+z^2-1=0
\end{aligned} \tag{B.17}$$

Then we can get

$$\begin{aligned}
&xy\left(\left(2A_1-A_2\right)x^2+\left(A_2-2B_1\right)y^2+\left(B_2-C_2\right)z^2\right)=0\\
&yz\left(\left(A_2-B_2\right)x^2+\left(2B_1-C_2\right)y^2+\left(C_2-2C_1\right)z^2\right)=0\\
&xz\left(\left(2A_1-B_2\right)x^2+\left(A_2-C_2\right)y^2+\left(B_2-2C_1\right)z^2\right)=0\\
&x^2+y^2+z^2=1
\end{aligned} \tag{B.18}$$

When two coordinates of $\mathbf{r}$ are zero, (B.18) holds, there is

$$\mathbf{r}=\begin{bmatrix}0 & 0 & \pm 1\end{bmatrix}^{\mathrm{T}}\mathrm{or}\begin{bmatrix}0 & \pm 1 & 0\end{bmatrix}^{\mathrm{T}}\mathrm{or}\begin{bmatrix}\pm 1 & 0 & 0\end{bmatrix}^{\mathrm{T}} \tag{B.19}$$

In order to analyze whether (B.19) are the minimum points, $\mathbf{r}=\begin{bmatrix}0 & 0 & 1\end{bmatrix}^{\mathrm{T}}$ is taken as an example, with $\lambda=-2C_1$. Set $\mathbf{r}\left(\Delta x,\Delta y\right)=\begin{bmatrix}\Delta x & \Delta y & 1\end{bmatrix}^{\mathrm{T}}$, where $\Delta x,\Delta y$ are small quantities. The Hessian matrix of (B.16) about $\Delta x,\Delta y$ at $\Delta x,\Delta y=0$ is

$$\mathbf{H}\left(E\left(\mathbf{r}\left(\Delta x,\Delta y\right),\lambda\right)\right)\big|_{\Delta x,\Delta y=0}=\left[\begin{matrix}\frac{\partial^2 E(\mathbf{r},\lambda)}{\partial \Delta x^2} & \frac{\partial^2 E(\mathbf{r},\lambda)}{\partial \Delta x\partial \Delta y}\\ \frac{\partial^2 E(\mathbf{r},\lambda)}{\partial \Delta x\partial \Delta y} & \frac{\partial^2 E(\mathbf{r},\lambda)}{\partial \Delta y^2}\end{matrix}\right]\Bigg|_{\Delta x,\Delta y=0}=\begin{bmatrix}2\left(B_2-2C_1\right) & 0\\ 0 & 2\left(C_2-2C_1\right)\end{bmatrix} \tag{B.20}$$

B_2-2C_1 and C_2-2C_1 are all positive, so the Hessian matrix defined by (B.20) is positive definite. Therefore, $\mathbf{r}=\begin{bmatrix}0 & 0 & 1\end{bmatrix}^{\mathrm{T}}$ is a minimum point of the MNSA cost function (B.15). Through the similar analysis process, it can be shown that (B.19) are all minimum points of the MNSA cost function (B.15).

When only one coordinate of $\mathbf{r}$ is zero, $z=0$ is taken as an example. Combining with (B.18) and we can get

$$\left(A_2-2A_1\right)x^2=\left(A_2-2B_1\right)y^2 \tag{B.21}$$

Then we can get

$$\mathbf{r}=\begin{bmatrix}\pm\frac{\sqrt{A_2-2B_1}}{\sqrt{2A_2-2B_1-2A_1}} & \pm\frac{\sqrt{A_2-2A_1}}{\sqrt{2A_2-2B_1-2A_1}} & 0\end{bmatrix}^{\mathrm{T}} \tag{B.22}$$

To analyze whether (B.22) are the minimum points, $\mathbf{r}=\begin{bmatrix}\frac{\sqrt{A_2-2B_1}}{\sqrt{2A_2-2B_1-2A_1}} & \frac{\sqrt{A_2-2A_1}}{\sqrt{2A_2-2B_1-2A_1}} & 0\end{bmatrix}^{\mathrm{T}}$ is taken as an example, with $\lambda=-\frac{2A_1(A_2-2B_1)+A_2(A_2-2A_1)}{2A_2-2B_1-2A_1}$. Set

$$\mathbf{r}(\delta,\Delta z)=\begin{bmatrix}\frac{\sqrt{A_2-2B_1}}{\sqrt{2A_2-2B_1-2A_1}}+\frac{\sqrt{2A_2-2B_1-2A_1}}{\sqrt{A_2-2B_1}}\delta\\ \frac{\sqrt{A_2-2A_1}}{\sqrt{2A_2-2B_1-2A_1}}-\frac{\sqrt{2A_2-2B_1-2A_1}}{\sqrt{A_2-2A_1}}\delta\\ \Delta z\end{bmatrix} \tag{B.23}$$

where δ and Δz are small quantities. The Hessian matrix of (B.16) with respect to $\delta,\Delta z$ at $\delta,\Delta z=0$ is

$$\mathbf{H}\left(E\left(\mathbf{r}\left(\delta,\Delta z\right),\lambda\right)\right)|_{\delta,\Delta z=0}=\begin{bmatrix}\frac{\partial^2 E(\mathbf{r},\lambda)}{\partial\delta^2} & \frac{\partial^2 E(\mathbf{r},\lambda)}{\partial\delta\partial\Delta z}\\ \frac{\partial^2 E(\mathbf{r},\lambda)}{\partial\delta\partial\Delta z} & \frac{\partial^2 E(\mathbf{r},\lambda)}{\partial\Delta z^2}\end{bmatrix}\Bigg|_{\delta,\Delta z=0}=\begin{bmatrix}8\left(A_1+B_1-A_2\right) & 0\\ 0 & \frac{(B_2-A_2)(A_2-2B_1)+(C_2-2B_1)(A_2-2A_1)}{A_2-B_1-A_1}\end{bmatrix} \tag{B.24}$$

where $A_1+B_1-A_2$ is negative, so $\mathbf{r}=\begin{bmatrix}\frac{\sqrt{A_2-2B_1}}{\sqrt{2A_2-2B_1-2A_1}} & \frac{\sqrt{A_2-2A_1}}{\sqrt{2A_2-2B_1-2A_1}} & 0\end{bmatrix}^{\mathrm{T}}$ is not a minimum point of MNSA cost function (B.15). Through the similar analysis process, (B.22) are not minimum points.

When $y=0$, the stationary points are

$$\mathbf{r}=\begin{bmatrix}\pm\frac{\sqrt{B_2-2C_1}}{\sqrt{2B_2-2C_1-2A_1}} & 0 & \pm\frac{\sqrt{B_2-2A_1}}{\sqrt{2B_2-2C_1-2A_1}}\end{bmatrix}^{\mathrm{T}} \tag{B.25}$$

When $x=0$, the stationary points are

$$\mathbf{r}=\begin{bmatrix}0 & \pm\frac{\sqrt{C_2-2C_1}}{\sqrt{2C_2-2B_1-2C_1}} & \pm\frac{\sqrt{C_2-2B_1}}{\sqrt{2C_2-2B_1-2C_1}}\end{bmatrix}^{\mathrm{T}} \tag{B.26}$$

Through the similar analysis process, it can be shown that (B.25) and (B.26) are not minimum points.

When all the coordinates of $\mathbf{r}$ are not zero, from (B.18), we can get

$$2A_1x^2+A_2y^2+B_2z^2=A_2x^2+2B_1y^2+C_2z^2=B_2x^2+C_2y^2+2C_1z^2=-\lambda \tag{B.27}$$

Then we can get

$$\begin{aligned}&\frac{x^2}{k_1}=\frac{y^2}{k_2}=\frac{z^2}{k_3}\\ &k_1=(2B_1-C_2)(B_2-C_2)-(C_2-2C_1)(A_2-2B_1)\\ &k_2=(C_2-2C_1)(2A_1-A_2)-(A_2-B_2)(B_2-C_2)\\ &k_3=(A_2-B_2)(A_2-2B_1)-(2B_1-C_2)(2A_1-A_2)\end{aligned} \tag{B.28}$$

When the signs of k_1,k_2,k_3 are the same, there are stationary points as follow

$$
\begin{aligned}
&\mathbf{r} = \begin{bmatrix} \pm x_0 & \pm y_0 & \pm z_0 \end{bmatrix}^{\mathrm{T}} \\
&x_0 = \sqrt{\left| \frac{(2B_1 - C_2)(B_2 - C_2) - (C_2 - 2C_1)(A_2 - 2B_1)}{(2B_1 - C_2)(B_2 - C_2 - 2A_1 + A_2) + (C_2 - 2C_1)(2A_1 - 2A_2 + 2B_1) + (A_2 - B_2)(A_2 - 2B_1 - B_2 + C_2)} \right|} \\
&y_0 = \sqrt{\left| \frac{(C_2 - 2C_1)(2A_1 - A_2) - (A_2 - B_2)(B_2 - C_2)}{(2B_1 - C_2)(B_2 - C_2 - 2A_1 + A_2) + (C_2 - 2C_1)(2A_1 - 2A_2 + 2B_1) + (A_2 - B_2)(A_2 - 2B_1 - B_2 + C_2)} \right|} \\
&z_0 = \sqrt{\left| \frac{(A_2 - B_2)(A_2 - 2B_1) - (2B_1 - C_2)(2A_1 - A_2)}{(2B_1 - C_2)(B_2 - C_2 - 2A_1 + A_2) + (C_2 - 2C_1)(2A_1 - 2A_2 + 2B_1) + (A_2 - B_2)(A_2 - 2B_1 - B_2 + C_2)} \right|}
\end{aligned}
\tag{B.29}
$$

To analyze whether (B.29) are the minimum points, take $\mathbf{r} = \begin{bmatrix} x_0 & y_0 & z_0 \end{bmatrix}^{\mathrm{T}}$ as an example, with $\lambda = -2A_1{x_0}^2 - A_2{y_0}^2 - B_2{z_0}^2$. Set

$$
\mathbf{r} = \begin{bmatrix} x_0 + \frac{1}{x_0}\delta_x & y_0 + \frac{1}{y_0}\delta_y & z_0 - \frac{1}{z_0}\delta_x - \frac{1}{z_0}\delta_y \end{bmatrix}^{\mathrm{T}} \tag{B.30}
$$

where δ_x and δ_y are small quantities. The Hessian matrix of (B.16) with respect to δ_x, δ_y at $\delta_x, \delta_y = 0$ is

$$
\mathbf{H}\left(E\left(\mathbf{r}\left(\delta_x, \delta_y\right), \lambda\right)\right)\big|_{\delta_x, \delta_y = 0} = \begin{bmatrix} \frac{\partial^2 E}{\partial {\delta_x}^2} & \frac{\partial^2 E}{\partial \delta_x \partial \delta_y} \\ \frac{\partial^2 E}{\partial \delta_x \partial \delta_y} & \frac{\partial^2 E}{\partial {\delta_y}^2} \end{bmatrix} = \begin{bmatrix} 8(A_1 + C_1 - B_2) & 4(A_2 - B_2 - C_2 + 2C_1) \\ 4(A_2 - B_2 - C_2 + 2C_1) & 8(B_1 + C_1 - C_2) \end{bmatrix} \tag{B.31}
$$

where $A_1 + C_1 - B_2$ and $B_1 + C_1 - C_2$ are negative, so $\mathbf{r} = \begin{bmatrix} x_0 & y_0 & z_0 \end{bmatrix}^{\mathrm{T}}$ is not a minimum point. Through the similar analysis process, (B.29) are not minimum points.

When the signs of k_1, k_2, k_3 are not the same, there is no stationary point whose three coordinates are not zero.

In summary, all minimum points of the MNSA cost function (B.15) are

$$
\mathbf{r} = \begin{bmatrix} 0 & 0 & \pm 1 \end{bmatrix}^{\mathrm{T}} \text{or} \begin{bmatrix} 0 & \pm 1 & 0 \end{bmatrix}^{\mathrm{T}} \text{or} \begin{bmatrix} \pm 1 & 0 & 0 \end{bmatrix}^{\mathrm{T}} \tag{B.32}
$$

Because that the values of $A_3, B_3, C_3, A_4, B_4, C_4, A_5, B_5, C_5$ are very small, the characteristics of the minimum point of (B.1) and (B.15) are very similar. So it is considered that the MNSA cost function (B.1) has 6 minimum points near MF axes. According to the definition of the MNSA cost function (4), the MNSA cost value far away from the three MF axes is much greater than that near the three MF axes.

For the MNSA cost functions $E(\mathbf{r}_1), E(\mathbf{r}_2), E(\mathbf{r}_3)$ in (B.12), there are included angles between the minimum points of $E(\mathbf{r}_1), E(\mathbf{r}_2), E(\mathbf{r}_3)$ and the corresponding axes of the RCF due to the elements of $2A_4\theta_z - 2B_4\theta_y$, $2C_4\theta_x - 2A_4\theta_z$ and $2B_4\theta_y - 2C_4\theta_x$. Based on the assumption (B.3), the included angles are very small and the minimum points of $E(\mathbf{r}_1), E(\mathbf{r}_2), E(\mathbf{r}_3)$ approximately coincide with the corresponding axes of the RCF.

The MNSA cost function has central symmetry, that is $E(\mathbf{r}) = E(-\mathbf{r})$, so the two minimum points close to the positive and negative directions of the same MF axis are symmetrical about the center of the RCF's origin point. Therefore, near each of the three MF axes, there is a pair of centrosymmetric minimum points of MNSA cost function with equal cost value.

When $\mathbf{R} = \mathbf{I}$, all the three coordinate axes of CFO are near a pair of minimum points of MNSA cost function, and $\mathbf{R} = \mathbf{I}$ is a minimum point of MNMA cost function. Obviously, in the definition domain of MNMA cost function, there is no other value of $\mathbf{R}$ which can make the MNMA cost value smaller than that with $\mathbf{R} = \mathbf{I}$. Therefore, $\mathbf{R} = \mathbf{I}$, is a global minimum point of the MNMA function. Thus, when CFO coincides with RCF, the global minimum value of MNMA cost function is obtained.

Next, we analyse whether the proposed initial value determination method in section 3.4 can ensure the initial value $\mathbf{R}_{init}$ near a global minimum point of MNMA cost function. Assume that there is a unit vector $\mathbf{r}''$ in the camera coordinate system. Let $\mathbf{v}''$ be the vector with the smallest angle with vector $\mathbf{r}''$ in the candidate vector set (37). Let $\mathbf{r}''_c$ be the coordinate vector of $\mathbf{r}''$ in the camera coordinate system. It can be shown that when $\mathbf{r}''_c$ meets the following conditions, the angle between $\mathbf{r}''$ and $\mathbf{v}''$ obtains the maximum value which is 9.86 degrees.

$$\begin{aligned}
\mathbf{r}''_c &= \frac{1}{\sqrt{2549 - 490\sqrt{26}}}\left[\pm 5 \quad \pm 5 \quad \pm 7\left(\sqrt{26} - 5\right)\right]^{\mathrm{T}} \\
\text{or } &\frac{1}{\sqrt{2549 - 490\sqrt{26}}}\left[\pm 5 \quad \pm 7\left(\sqrt{26} - 5\right) \quad \pm 5\right]^{\mathrm{T}} \\
\text{or } &\frac{1}{\sqrt{2549 - 490\sqrt{26}}}\left[\pm 7\left(\sqrt{26} - 5\right) \quad \pm 5 \quad \pm 5\right]^{\mathrm{T}}
\end{aligned} \tag{B.33}$$

Therefore, the candidate vector set (37) can ensure that at least one candidate vector is included within the angle range of 10 degrees of each MNSA minimum point no matter what the attitude of the camera coordinate system is. In addition, the selection of vector $\mathbf{v}_1$ is to take the vector with the lowest MNSA cost value for all candidate vectors, so $\mathbf{r}_{1,init}$ will be selected close to one of the MNSA minimum points. Since $\mathbf{v}_2$ is selected from the candidate vectors with an angle of 60 degrees to 120 degrees with $\mathbf{v}_1$, the candidate vectors with a small angle with $\mathbf{v}_1$ will not be selected as $\mathbf{v}_2$. Therefore, the vector $\mathbf{r}_{2,init}$ will be selected near a MNSA minimum point different from that of $\mathbf{r}_{1,init}$. Since $\mathbf{r}_{3,init}$ is perpendicular to $\mathbf{r}_{1,init}$ and $\mathbf{r}_{2,init}$, $\mathbf{r}_{3,init}$ will be determined near the third MNSA minimum point according to the small angle assumption. Therefore, the initial value selection method can ensure that $\mathbf{R}_{init}$ is near a global minimum point of the MNMA cost function.

In Summary, the proposed MF estimation method can ensure the global optimality when the Manhattan scene assumption (B.3) holds.

References

Bazin, J.-C., Seo, Y., Demonceaux, C., Vasseur, P., Ikeuchi, K., Kweon, I., and Pollefeys, M. (2012a). Globally optimal line clustering and vanishing point estimation in manhattan world. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 638–645. IEEE.

Bazin, J.-C., Seo, Y., Hartley, R., and Pollefeys, M. (2014). Globally optimal inlier set maximization with unknown rotation and focal length. In *European Conference on Computer Vision*, pages 803–817. Springer.

Bazin, J.-C., Seo, Y., and Pollefeys, M. (2012b). Globally optimal consensus set maximization through rotation search. In *Asian Conference on Computer Vision*, pages 539–551. Springer.

Choi, W., Chao, Y.-W., Pantofaru, C., and Savarese, S. (2013). Understanding indoor scenes using 3d geometric phrases. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 33–40.

Coughlan, J. M. and Yuille, A. L. (1999). Manhattan world: Compass direction from a single image by bayesian inference. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 941–947. IEEE.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.

Furukawa, Y., Curless, B., Seitz, S. M., and Szeliski, R. (2009). Manhattan-world stereo. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1422–1429. IEEE.

Ghanem, B., Thabet, A., Carlos Niebles, J., and Caba Heilbron, F. (2015). Robust manhattan frame estimation from a single rgb-d image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3772–3780.

Gupta, S., Arbelaez, P., and Malik, J. (2013). Perceptual organization and recognition of indoor scenes from rgb-d images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 564–571.

Hartley, R. I. and Kahl, F. (2009). Global optimization through rotation space search. *International Journal of Computer Vision*, 82(1):64–79.

Hedau, V., Hoiem, D., and Forsyth, D. (2009). Recovering the spatial layout of cluttered rooms. In *2009 IEEE 12th international conference on computer vision*, pages 1849–1856. IEEE.

Hedau, V., Hoiem, D., and Forsyth, D. (2010). Thinking inside the box: Using appearance models and context based on room geometry. In *European Conference on Computer Vision*, pages 224–237. Springer.

Horn, B. K. P. (1984). Extended gaussian images. *Proceedings of the IEEE*, 72(12):1671–1686.

Hsiao, M., Westman, E., Zhang, G., and Kaess, M. (2017). Keyframe-based dense planar slam. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5110–5117. IEEE.

Joo, K., Oh, T.-H., Kim, J., and Kweon, I. S. (2016). Globally optimal manhattan frame estimation in real-time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1763–1771.

Joo, K., Oh, T.-H., Kim, J., and Kweon, I. S. (2018). Robust and globally optimal manhattan frame estimation in near real time. *IEEE transactions on pattern analysis and machine intelligence*, 41(3):682–696.

Le, P.-H. and Košecka, J. (2017). Dense piecewise planar rgb-d slam for indoor environments. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4944–4949. IEEE.

Lee, D. C., Gupta, A., Hebert, M., and Kanade, T. (2010). Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. In *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada.*

Lee, D. C., Hebert, M., and Kanade, T. (2009). Geometric reasoning for single image structure recovery. In *2009 IEEE conference on computer vision and pattern recognition*, pages 2136–2143. IEEE.

Liu, Y., Chen, G., and Knoll, A. (2020). Globally optimal vertical direction estimation in atlanta world. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Parra Bustos, A., Chin, T.-J., and Suter, D. (2014). Fast rotation search with stereographic projections for 3d registration. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3930–3937.

Ramalingam, S. and Brand, M. (2013). Lifting 3d manhattan lines from a single image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 497–504.

Schwing, A. G., Hazan, T., Pollefeys, M., and Urtasun, R. (2012). Efficient structured prediction for 3d indoor scene understanding. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2815–2822. IEEE.

Silberman, N., Hoiem, D., Kohli, P., and Fergus, R. (2012). Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pages 746–760. Springer.

Sinha, S. N., Steedly, D., and Szeliski, R. (2009). Piecewise planar stereo for image-based rendering. In *IEEE 12th International Conference on Computer Vision, ICCV 2009, Kyoto, Japan, September 27 - October 4, 2009*.

Straub, J., Bhandari, N., Leonard, J. J., and Fisher, J. W. (2015). Real-time manhattan world rotation estimation in 3d. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1913–1920. IEEE.

Straub, J., Freifeld, O., Rosman, G., Leonard, J. J., and Fisher, J. W. (2017). The manhattan frame model—manhattan world inference in the space of surface normals. *IEEE transactions on pattern analysis and machine intelligence*, 40(1):235–249.

Straub, J., Rosman, G., Freifeld, O., Leonard, J. J., and Fisher, J. W. (2014). A mixture of manhattan frames: Beyond the manhattan world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3770–3777.

Taylor, C. J. and Cowley, A. (2013). Parsing indoor scenes using rgb-d imagery. In *Robotics: Science and Systems*, volume 8, pages 401–408.

Ulrich, G. (1984). Computer generation of distributions on the m-sphere. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 33(2):158–163.

Wang, L. and Wu, Z. (2019). Rgb-d slam with manhattan frame estimation using orientation relevance. *Sensors*, 19(5):1050.

Wu, Z. and Wang, L. (2017). Recovering the manhattan frame from a single rgb-d image by using orientation relevance. In *2017 29th Chinese Control And Decision Conference (CCDC)*, pages 4574–4579. IEEE.

Zhang, C., Wang, L., and Yang, R. (2010). Semantic segmentation of urban scenes using dense depth maps. In *European Conference on Computer Vision*, pages 708–721. Springer.

Zhou, H., Zou, D., Pei, L., Ying, R., Liu, P., and Yu, W. (2015). Structslam: Visual slam with building structure lines. *IEEE Transactions on Vehicular Technology*, 64(4):1364–1375.