

Depth Data Reconstruction Based on Gaussian Mixture Model

Zhe Li¹, Chen Ma², Tian-Fan Zhang¹

¹College of Technology, Hubei Engineering University, Xiao Gan 432000, China, Department of Automatic Control, Northwestern Polytechnical University, Xi'an 710072, China

²University of Victoria, Victoria BC V8P 5C2, Canada

Emails: lizhe_hbeu@vip.163.com machenhb@hotmail.com alitasoft@hotmail.com

Abstract: Depth data is an effective tool to locate the intelligent agent in space because it accurately records the 3D geometry information on the surface of the scanned object, and is not affected by factors like shadow and light. However, if there are many planes in the work scene, it is difficult to identify objects and process the resulting huge amount of data. In view of this problem and targeted at object calibration, this paper puts forward a depth data calibration method based on Gauss mixture model. The method converts the depth data to point cloud, filters the noise and collects samples, which effectively reduces the computational load in the following steps. Besides, the authors cluster the point cloud vector with the Gaussian mixture model, and obtain the target and background planes by using the random sampling consensus algorithm to fit the planes. The combination of target Region Of Intelligent agent (ROI) and point cloud significantly reduces the computational load and improves the computing speed. The effect and accuracy of the algorithm is verified by the test of the actual object.

Keywords: Depth data, point cloud, normal vector clustering, Gaussian mixture model, random sampling consensus algorithm, object calibration, CAMShift.

1. Introduction

The spatial calibration of intelligent agent is a key issue in autonomous robot control [1]. Traditional spatial calibration methods have some shortcomings. For instance, ultrasonic calibration has a poor accuracy [2], 2D image calibration requires assistive tools and is vulnerable to ambient light interference. In contrast, the depth image data, which reflects the distance between the imaging plane of the camera and the object, is not affected by factors like shadow and light, and is therefore widely used in intelligent systems in need of high calibration accuracy.

Traditional depth sensors can obtain high quality data, but they are too complex and costly.

With the development of imaging technologies, new depth sensors are invented. For example, Microsoft launches an optimized structured light depth sensor [3] called Kinect [4]. Striking a balance between price and performance, the sensors meet the basic needs of research and application.

All depth sensors create a huge amount of data. Let's take Kinect as an example. It creates 118 MB/s of data in 1920×1080 at 30 fps. An effective way to reduce the data volume is to convert the depth data to 3D data points and extract the advanced features of the image by image segmentation. Unlike general images, which are cut into Region Of Intelligent (ROIs), or target contours, the depth image is broken down into several regions with the same or relevant characteristic information based on the similarity principle [5]. With a direct impact on the final results of processing, the effect of segmentation is the focus and difficult point of the research on depth image [6].

The depth image carries information that facilitates spatial calibration, so it has become a research hotspot in recent years [7]. Generally speaking, the segmentation methods fall into three categories:

Edge segmentation. This method can locate the regional boundaries swiftly and accurately but it is easily influenced by factors like noise or occlusion. Wang et al. [8] uses several types of gradient operators to extract the edges of the depth image, while Benlami [9] detects the edges by the surface curvature changes.

Region segmentation. This method classifies neighbouring pixels with similar characteristics as one region. Under certain conditions, the regions are clustered to obtain the characteristics. The method can yield closed edge information, but it is likely to encounter boundary dislocation due to its complexity. P. J. Besl and R. C. Jain (see [10]) classify surfaces into eight types based on the Gaussian curvature and use quadric surface fitting for regional growth.

Combination of edge segmentation and region segmentation. Attaching equal importance to speed and accuracy, the combined method identifies the location and number of regions, and further divides the regions [11-13]. On the basis of RGB and the depth image, Mirante, Georgiev and Gotchev [14] constructs a real-time segmentation algorithm. Paris and Durand [16] put forward a mean shift algorithm which searches for modality based on pixel density, and thereby develop a parameterized Gaussian mixture segmentation method [17].

This paper proposes an improved method for depth data segmentation and object calibration. Firstly, the authors convert the original depth data to discrete 3D point cloud, filter the noise and collect samples, thereby effectively reducing the computational load in the following steps. Secondly, the authors cluster the normal vectors of the sampled data with the Gaussian mixture model to obtain the preliminary plan features, extract and decompose the characteristic plane by the RANSAC algorithm. Thirdly, the authors realize the calibration of intelligent agent by the combination of ROI and point cloud, which significantly reduces the computational load by 98%. Finally, the authors verify the effect and accuracy of the algorithm through the test of an actual object.

2. 3D data reconstruction

The structured light depth sensor usually has two cameras: an infrared camera and a RGB camera, which are used to obtain the depth image and RGB image, respectively. The two cameras have different spatial locations, so the original data needs to be reconstructed. The authors align the two sets of data through coordinates transformation; the x, y coordinates of spatial data points are calculated by the triangle similarity theory, and the 3D point cloud data is saved in the format (x, y, z, R, G, B) . See Fig. 1 for the reconstruction process.

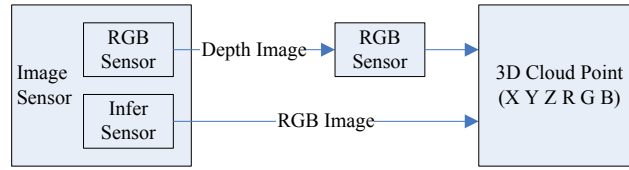


Fig. 1. Reconstruction process

The depth data should go through coordinate transformation because the imaging device uses right-handed coordinate system, while the depth data is expressed in projected coordinate system with the upper-left corner as the origin. Suppose both depth image and RGB image have a resolution of $Res = (width, height)$, the 2D coordinates of the depth data are (x'_i, y'_j) , $i \in [1, width], j \in [1, height]$, and the corresponding depth data is $depth_{ij}$, the real-world coordinates can be obtained by the similar triangle theory:

$$(1) \quad X_{ij} = \left(x'_i - \frac{width}{2} \right) \times Z_{ij} \times \frac{1}{f}, \quad Y_{ij} = \left(y'_j - \frac{height}{2} \right) \times Z_{ij} \times \frac{1}{f},$$

where Z is obtained directly from the depth data, i.e., $Z_{ij} = depth_{ij}$; f is the focal length of the camera, which is calculated by formula [18]

$$(2) \quad f = f_{SXGA} \times \frac{X_{ResOutput}}{X_{ResSXGA}} = 1050 \times \frac{640}{1280} = 525.$$

The authors obtain the 3D point cloud data sets of the depth image by Formula (1). See Fig. 2 for the results.

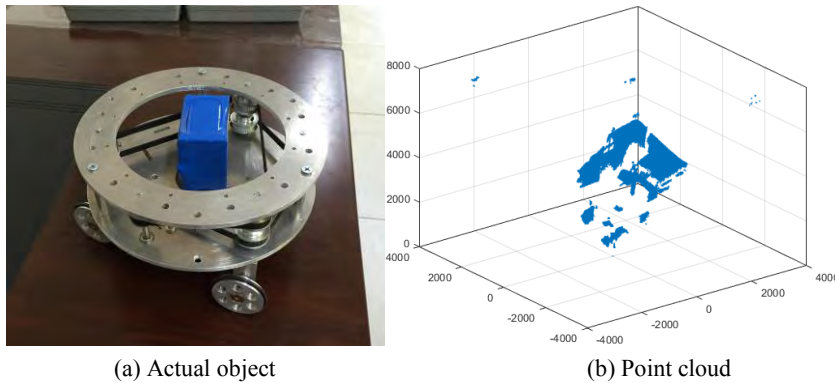


Fig. 2. Comparison of the actual object and the point cloud

To acquire depth information, the depth sensor obtains a speckle image of the space via the infrared sensor. If it is impossible to form a speckle on the surface of the object, there is no way to obtain the depth information. In this case, the depth data of the corresponding points is 0. According to Formula (1), the corresponding 3D point data should also be 0. Thus, it should be classified as noise. Moreover, speckles are not distributed evenly. In the point cloud data set, some data points or cluster points are far away from the main cloud. These data are also regarded as noise. Most of the noises can be filtered through conventional filters, such as median filter (Fig. 3).

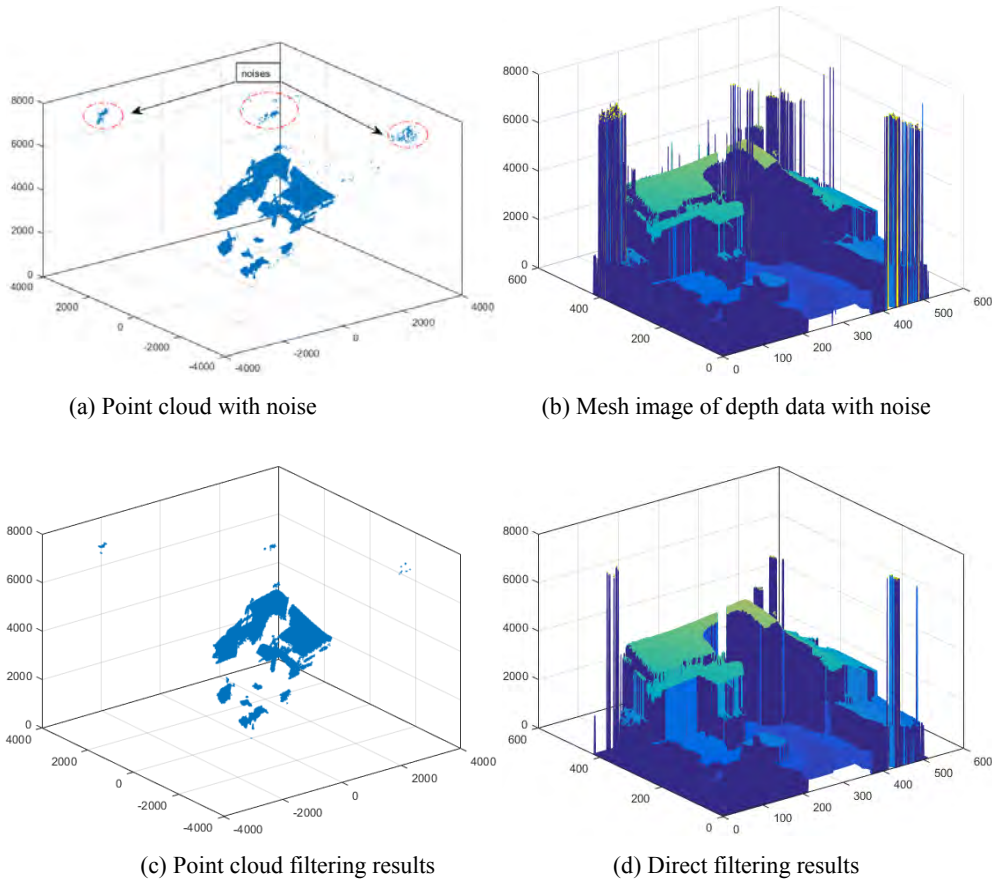


Fig. 3. Noise and filtering

However, the small clustered noises far away from the main point cloud are difficult to be filtered with conventional methods. These noises should be filtered by clustering method in the follow-up principal component analysis. Besides, even if the surface is very smooth, e.g., walls, the depth data still fluctuates because the depth image has a resolution of 3-4 mm (Fig. 4).

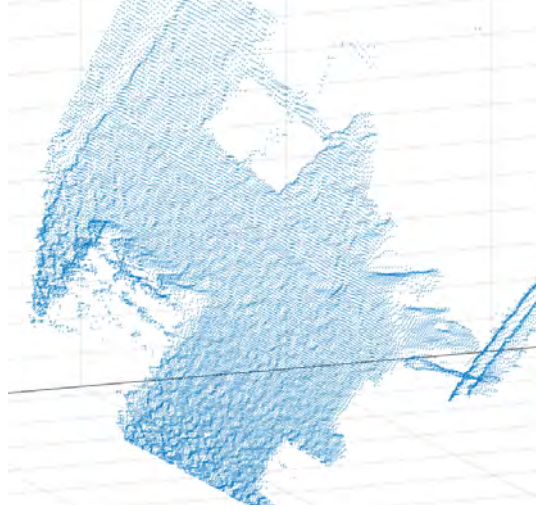


Fig. 4. Depth data difference resulted from the resolution of the sensor
(This is a partially enlarged view of Fig. 3a, which describes a wall, with a window above it.)

3. Point cloud data segmentation and object calibration

3.1. Calculation of point cloud vector

Whereas there is no association between the points in the initial point cloud, the authors have to establish a grid model that corresponds to point cloud data and calculate the normal vectors of data points before the segmentation. Normally, the normal vectors are obtained by calculating the weighted means of all adjacent points. As a global computing method, this approach involves heavy computational load. Thus, this paper resorts to the fitting process.

Let's assume that the local sampling plane is smooth in the point cloud, and any point and the points adjacent to it are fitted well to the plane. For every data point p of the point cloud, the authors solve the values of k adjacent points of p , and obtain a least squares plane P for these points. The plane is expressed as follows:

$$(3) \quad P(n, d) = \arg \min_{(n, d)} \sum_{i=1}^k (n \times p_i - d)^2,$$

where, n is the normal vector of plane P , and d is the distance between the origin and the plane P .

For a given point $p_i = (x_i, n_i)$, the authors calculate the approximate tangent plane of point p_i , and then find the approximate normal vector n_i of point p_i , so as to minimize the result of the following formula:

$$(4) \quad \text{ErrorFun} = \sum_{j=1}^k \left((x_j, v_j) \cdot n_j \right)^2.$$

After solving the values of k adjacent points of p , the authors obtain the centroid \bar{p} of the set of adjacent points through simple calculation. For any normal

vector n , there is $\|n\|_2 = 1$. On this basis, the authors get the following covariance matrix:

$$(5) \quad C = \sum_{i=1}^k (p_i - \bar{p}) \cdot (p_i - \bar{p})^T, \quad C \cdot \bar{v}_j = \lambda_j \bar{v}_j, \quad j \in \{0, 1, 2\},$$

where λ_j is the j -th eigenvalue of the covariance matrix, and \bar{v}_j is the j -th feature vector of the covariance matrix. The authors carry out covariance analysis of the adjacent points of p , acquire the surface changes in the local region of p , and estimate the value of curvature. The value at the minimum ErrorFun illustrates the fit of the plane. All normal vectors are adjusted in accordance with next formula, so that every vector \bar{n}_j always points towards the viewpoint:

$$(6) \quad \bar{n}_j \cdot (v_j - p_j) > 0.$$

3.2. Clustering based on Gaussian mixture model

3.2.1. Gaussian mixture model

Due to the particularity of the indoor scene, the approximate planes made up of normal vector data are oriented towards the direction of X -axis, Y -axis or Z -axis. The planes can be clearly classified by the normal value of the data. The authors adopt the Gaussian Mixture Model (GMM) to cluster the planes because the method can approximate any continuous probability distribution. As a linear combination of M Gaussian density functions, the GMM is defined as

$$(7) \quad P(x) = \sum_{i=1}^M \pi_i N_i(x, \mu_i, \Sigma_i),$$

where $N_i(x, \mu_i, \Sigma_i)$ is Gaussian distribution at the mean value of μ_i and the covariance of Σ_i . The probability distribution function is

$$(8) \quad N_i(x) = \left(1 / (2\pi)^{d/2} |\Sigma_i|^{1/2}\right) \cdot \exp \left\{ -\frac{1}{2} (x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i) \right\},$$

π_i stands for the mixing parameter, the weight of the i -th Gaussian distribution or the prior probability,

$$(9) \quad \sum_{i=1}^M \pi_i = 1, \quad 0 \leq \pi_i \leq 1.$$

The parameter values of the Gaussian distributions are estimated in light of the observed value of the data set X . In preparation for plane segmentation, the given data are fitted sufficiently by EM algorithm to get the final parameter values of all Gaussian distributions.

3.2.2. EM Algorithm

As an efficient MLE method, the EM Algorithm mostly applies to parameter estimation of incomplete data. Based on real data and variables containing hidden variables, it is possible to establish a Gaussian mixture probability model by the definition of the GMM. The most commonly used method is the MLE, which obtains the estimated values of parameters by the MLE function. The authors mark

the undetermined parameters of the function π_i, μ_i, Σ_i uniformly as θ , and use X to describe all observable data sets. In this case, the MLE function is as follows:

$$(10) \quad P(X|\theta) = \prod_{i=1}^N P(x_i|\theta) \Rightarrow \theta^* = \arg \max_{\theta} P(X|\theta).$$

To simplify the issue, the authors find the logarithm of function (10):

$$(11) \quad \log(P(X|\theta)) = \sum_{i=1}^N \log(P(x_i|\theta)) = \sum_{i=1}^N \log\left(\sum_{m=1}^M \pi_m N(x_i; \mu_m, \Sigma_m)\right).$$

Formula (11) is still too complex. It can be rewritten to contain the hidden variable Z . Let Gaussian distribution set is $Z = (z_1, z_2, \dots, z_n)$, $z_i \in \{1, 2, \dots, M\}$, $i = 1, 2, \dots, N$, and the data set $\{(x_1, z_1), (x_2, z_2), (x_3, z_3)\}$ is formed by x and z . When $z_i = k$, the observed value x_i of the i -th sample is generated by the k -th component. If the set Z is known, the MLE issue of Formula (11) can be converted to

$$(12) \quad \log(P(X|\theta)) = \sum_{i=1}^N \log p(x_i, z_i|\theta) = \sum_{i=1}^N \log \log(\pi_{iN} N(x; \mu_{z_i}, \Sigma_{z_i})).$$

However, it is often unknown to which distribution belongs each x , i.e., z is often unknown. Hence, the distribution of Z should be determined according to the estimated parameter in the last round. The expectation of Formula (12) is defined as

$$(13) \quad Q(\theta, \theta^{\text{old}}) = E_z[\log p(X, Z|\theta) | X, \theta^{\text{old}}],$$

whereas $p(z_i = k | x_i, \theta^{\text{old}})$ is the probability density of the k -th component of sample x_i , Formula (13) can be expressed as

$$(14) \quad Q(\theta, \theta^{\text{old}}) = \sum_{k=1}^M \sum_{i=1}^N \log(\pi_k) p(k | x_i, \theta^{\text{old}}) + \sum_{k=1}^M \sum_{i=1}^N \log(p(x_i; \mu_k, \Sigma_k)) p(k | x_i, \theta^{\text{old}}).$$

The new value of π_k should not be solved with the partial derivatives alone because the sum of all π_k must be equal to 1. It is necessary to use the Lagrange multiplier, thus Formula (14) is changed to

$$(15) \quad \pi_k^{\text{new}} = \frac{1}{N} \sum_{i=1}^N p(k | x_i, \theta^{\text{old}}).$$

The updated values of π_k and Σ_k can be obtained by seeking the derivatives of π_k and Σ_k in Formula (15). Let the derivative values be 0, there is:

$$(16) \quad \pi_k^{\text{new}} = \sum_{i=1}^N x_i p(k | x_i, \theta^{\text{old}}) / \sum_{i=1}^N p(k | x_i, \theta^{\text{old}}),$$

$$(17) \quad \Sigma_k^{\text{new}} = \sum_{i=1}^N p(k | x_i, \theta^{\text{old}}) (x_i - \mu_k^{\text{new}})^T / \sum_{i=1}^N p(k | x_i, \theta^{\text{old}}),$$

where

$$(18) \quad p(k | x_i, \theta^{\text{old}}) = \pi_k^{\text{old}} N(x_i | \mu_k^{\text{old}}, \Sigma_k^{\text{old}}) / \sum_{m=1}^M \pi_m^{\text{old}} N(x_i | \mu_m^{\text{old}}, \Sigma_m^{\text{old}}).$$

3.3. Plane extraction and merging

3.3.1. Plane extraction

The parameters of each point on cloud surface are calculated and the plane is extracted. Normally, all points located on the same plane satisfy the following equation:

$$(19) \quad ax + by + cz = d,$$

where (a, b, c) is the normal vector of the plane, $a^2 + b^2 + c^2 = 1$, (x, y, z) is the spatial coordinates of a point on the plane, and d is the distance between the plane and the origin of coordinates.

If the point cloud is expressed as $\{x_i, y_i, z_i\}$, $i = 1, 2, \dots, n$, where n stands for the total number of data points, the parameters of the plane can be expressed in the form of a fundamental matrix, and the parameter calculation can be converted into the estimation of the fundamental matrix. Let F be the fundamental matrix, there is

$$(20) \quad [x_i y_i z_i - 1]F = 0,$$

where $F = [a \ b \ c \ d]^T$.

It is known that the fundamental matrix F has three degrees of freedom. The authors select three points to constitute an “interior point” by the RANDOM SAMPLING Consensus (RANSAC) method for the purpose of calculating the initial values of parameters and looking for other interior points. This approach effectively reduces exterior points and noise and improves the speed of data processing,

$$(21) \quad p = 1 - \left(1 - (1 - \varepsilon)^t\right)^T,$$

where p is the data error rate and t is the minimum amount of data needed to calculate the model parameters, $t = 0.2$. If the probability that at least one subset is entirely made up of interior points $p > 99\%$, the iterative sampling times $P \geq 1000$.

Through the above steps, the authors set up a parameter model. The quality of the model is judged by the Euclidean distance, i.e., the distance between plane (a, b, c, d) to $p(x, y, z)$:

$$(22) \quad d(p, \text{plane}) = |ax + by + cz - d|.$$

Due to the sampling error of the depth sensor, the plane of the model is an approximate plane. Therefore, when judging the quality of the model by Euclidean distance, the authors set a tolerance threshold. Besides, they also consider the relationship between adjacent points. That is, point cloud data is concentrated more interiorly and sparser – exteriorly. Eventually, it is decided to judge the density threshold of interior data with a density circle of radius r , and remove the data below the threshold from the plane.

3.3.2. Plane merging

Through the above steps, the authors acquire a localized collection of point cloud surfaces, also known as the Collection of Cloud Domains (CCD). The domains on the same plane can be merged by calculating the angle and distance of the CCD. plane1 and plane2 are two planes in the CCD and $p1$ and $p2$ are two points on the

planes. \vec{r}_{12} is the distance between two points. \vec{n}_1 and \vec{n}_2 are the normal of $p1$ and $p2$. The angle between the two planes is ϕ . See Fig. 5 for the plane merger model.

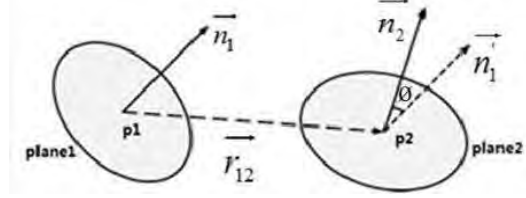


Fig. 5. Horizontal Merger model

The merger must meet several conditions: plane angle ϕ is smaller than the set threshold thres_ϕ and difference of the planes to the origin must be shorter than the set threshold thres_d :

$$(23) \quad \phi = \cos^{-1}(\vec{n}_1 \cdot \vec{n}_2).$$

When distance between the planes and the origin increases, the error of ϕ also increases, resulting in significant error in the distance difference $\Delta d = |d_1 - d_2|$. Therefore, Δd is defined as

$$(24) \quad \Delta d = \max\left(\left|\left(\vec{r}_{12} \cdot \vec{n}_1\right)\right|, \left|\left(\vec{r}_{12} \cdot \vec{n}_2\right)\right|\right).$$

If the two subsets (planes) in the CCD meet the two thresholds requirements, the two subsets should be merged into one.

3.4. Object calibration

The spatial information of the intelligent agent is obtained in two steps. First, process the RGB video data with a Kalman filter and by the Continuously Adaptive Mean-Shift (CAMShift) Tracking Algorithm [20]. CAMShift is known as an effective way to solve object deformation and occlusion, and does not require a lot of system resources, or consumption of long time. To improve tracking accuracy, the authors install a monochrome image calibration board on the intelligent agent [21]. To be specific, the plate is installed on the top agent of each plane (Fig. 6).



Fig. 6. The calibration plate on the intelligent agent

C is the image centroid, $S(x, y)$ is the pixel coordinates for the object, and T is the projected pixel coordinates. According to the description by Li et al. [21], the CAMShift Algorithm is illustrated on Fig. 7.

- 1: Selecting appropriate target search window,
for example, using optical flow method to get the ROI,
and set the gray value range of image is [0,255];
- 2: Calculating the zero order matrix and first order moments of the search window;
- 3: $M_{00} = \sum_x \sum_y I(x, y)$
- 4: The first-order matrix of x and y is:
- 5: $M_{10} = \sum_x \sum_y xI(x, y), M_{01} = \sum_x \sum_y yI(x, y)$
- 6: Calculating the center of the search window:
- 7: $x_c = \frac{M_{10}}{M_{00}}, y_c = \frac{M_{01}}{M_{00}}$
- 8: Setting the size of the search window:
- 9: $s = 2\sqrt{M_{00}/256} = 0.125\sqrt{M_{00}}$
- 10: Iterative computation for the next frame.

Fig. 7. The process of CAMShift Algorithm

In the calibration of intelligent agent, it is not necessary to process the entire depth image. Instead, the researcher only has to process the ROI extracted from the intelligent agent by CAMShift. The ROI is very small, taking up 1-5% of the entire image area. Coupled with the aforementioned point cloud data segmentation method, the ROI can greatly reduce the overall computational load because the data which have to be processed is only 5-15% of the original data.

4. Test and result analysis

In view of the MAS architecture multi-agent system, the authors select three scenes of different complexity for the tests, namely a simple corner with low resolution, a deep corridor and an area beyond the measuring distance. The mixture control of the MAS architecture multi-agent system is conducted by Li et al. [22]. The test results are displayed on Figs 8-10.

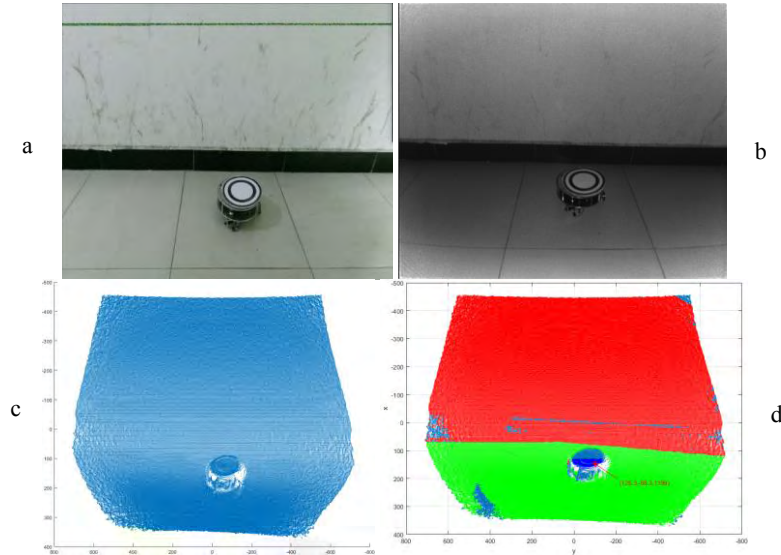


Fig. 8. Test 1: A simple corner with low resolution:
RGB (a); depth (b); point cloud (c); segmentation planes (d)

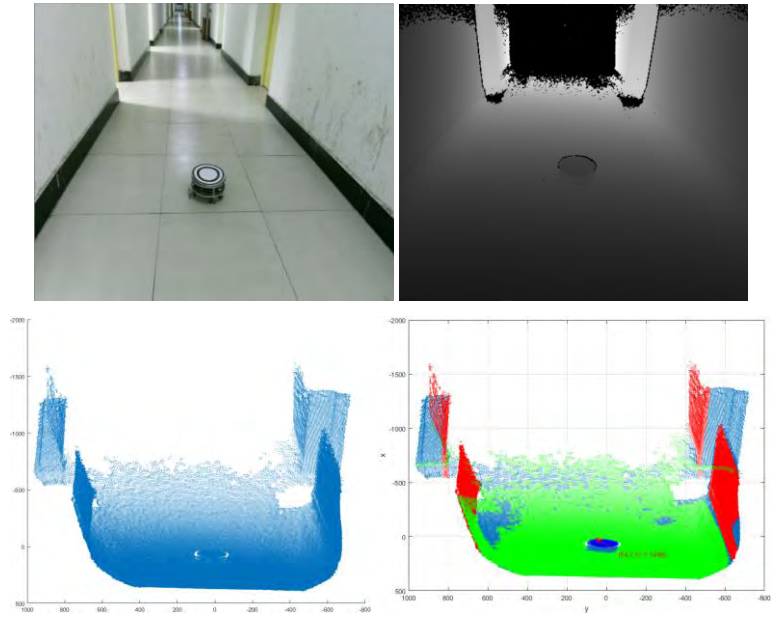


Fig. 9. Test 2: A deep corridor

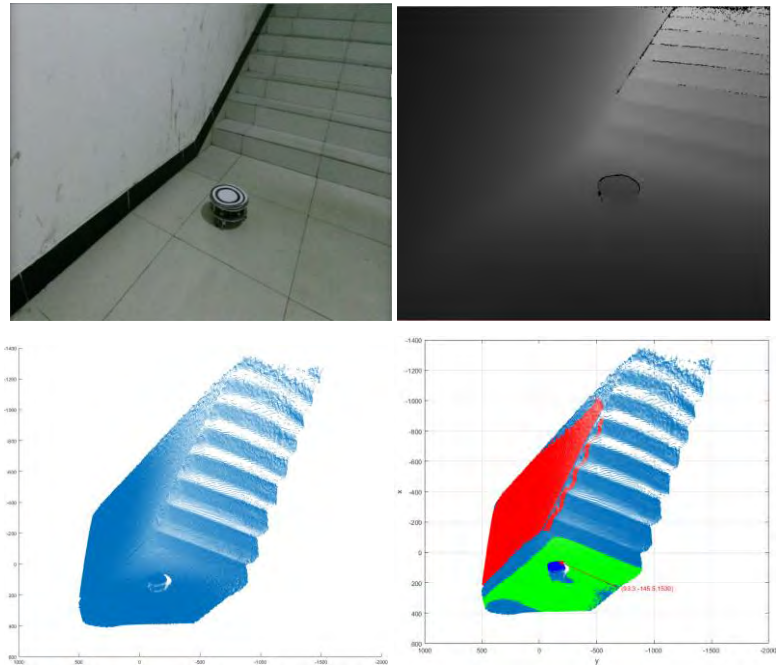


Fig. 10. Test 3: An area beyond the measuring distance

According to the experiment results, the data of the scenes and the intelligent agent can be segmented clearly. To a certain extent, the method is noise resistant. However, there are quite a few interferences and burrs on edges. That is because plane segmentation is affected by the significant deviation in normal vector value

on edges and other areas with obvious data changes, where the normal vector of the data points is replaced by that of its adjacent points. Similarly, the effect is rather poor on objects with large curvature, such as balls and barrels.

To sum up, the ROI of the RGB image not only reduces the computational load but also eliminates the deviation caused by the global computing method.

5. Conclusion

This paper studies the depth data reconstruction for the spatial calibration of intelligent agent. The authors discover that high quality depth data are easily obtained with the new structured light depth sensor, and the data volume can be reduced by more than 80% if the depth data is converted to 3D point cloud and filtered. The authors decompose the characteristic plane by the EM algorithm based on the GMM, obtain the ROI of the intelligent agent by processing the RGB image by CAMShift, and realize spatial calibration in light of the local plane of the depth data of the object. The combination of ROI and point cloud can reduce the data volume by 98% and greatly improve the processing speed.

Moreover, the authors test the proposed MAS architecture system to evaluate the performance of the algorithm. Compared with the results obtained with a calibration plate under ideal lighting, the results are satisfactory when the scene has a simpler layout, but the results are less accurate due to the interference of the background and other objects.

Acknowledgments: This study is made possible by the support from the Program of Science Foundation of Zhejiang Province (LY13F020023, LY14F020030), Hubei Provincial Department of Education (B2016517), and the College of Technology, Hubei Engineering University (Hgxy01).

References

1. Chris, O., W. Ben. Derivation of Spatiotemporal Data for Cyclists (from Video) to Enable Agent-Based Model Calibration. – *Procedia Computer Science*, Vol. **52**, 2015, pp. 932-937.
2. Syamsul, H., T. Oiwa, T. Tanaka, J. Asama. Calibration Error Improvement Based on Ultrasonic Oscillation for a Linear Motion Rolling Bearing During Sinusoidal Motion. – *Precision Engineering*, Vol. **38**, 2014, No 3, pp. 617-627.
3. Ito, M. A Three-Level Checkboard Pattern (PCT) Projection Method for Curved Surface Measurement. – *Pattern Recognition*, Vol. **28**, 1995, No 1, pp. 27-40.
4. Wu, Y. L., X. B. Zhang. Dynamic Arm Three Dimensional Posture Recognition Based on Device and Simulation. – *Computer Simulation*, Vol. **7**, 2016, pp. 367-387.
5. Yu, H. Y., X. B. Zhi, J. L. Fan. Image Segmentation Based on Weak Fuzzy Partition Entropy. – *Neurocomputing*, Vol. **168**, 2016, pp. 994-1010.
6. Ji, R. G., L. J. Cao, Y. Wang. Joint Depth and Semantic Inference from a Single Image via Elastic Conditional Random Field. – *Pattern Recognition*, Vol. **59**, 2016, pp. 268-281.
7. Pablo, M., I. Oscar, C. Oscar, C. Stefano. A Survey on Image Segmentation Using Metaheuristic-Based Deformable Models: State of the Art and Critical Analysis. – *Applied Soft Computing*, Vol. **44**, 2016, pp. 1-29.
8. Wang, J., Y. H. Wang, M. Jiang, X. Y. Yan, M. M. Song. Moving Cast Shadow Detection Using Online Sub-Scene Shadow Modelling and Object Inner-Edges Analysis. – *Journal of Visual Communication and Image Representation*, Vol. **25**, 2014, No 5, pp. 978-993.

9. Benlamri, R. Curved Shapes Construction for Object Recognition. – IEEE Theory and Applications, Vol. **10**, 2002, No 3, pp. 167-172.
10. Egorov, A. V., M. C. Hansen, D. P. Roy, A. Kommarreddy, P. V. Potapov. Image Interpretation-Guided Supervised Classification Using Nested Segmentation. – Remote Sensing of Environment, Vol. **165**, 2015, pp. 135-147.
11. Gotardo, P., O. Bellon. Range Image Segmentation into Planar and Quadric Surfaces Using an Improved Robust Estimator and Genetic Algorithm. – IEEE Transactions on System, Man, Cybernetics B, Vol. **34**, 2004, No 6, pp. 2303-2316.
12. Angel, D. S. Surface Model Generation from Range Images of Industrial Environments. – In: Proc. of 2nd International Symposium on 3DPVT, Barcelona, Spain, 2004, pp. 868-871.
13. Qian, C., F. T. Li, G. H. Ge. Feature Extraction from Range Images in 3D Modelling of Urban Scenes. – In: Proc. of International Conference on RISSP, Changsha, China, 2003, pp. 909-915.
14. Mirante, E., M. Georgiev, A. Gotchev. A Fast Image Segmentation Algorithm Using Color and Depth Map. – In: 3DTV Conference: The True Vision – Capture, Transmission and Display of 3D Video (3DTV-CON), 2011, pp. 1-4.
15. Wang, H., J. Oliensis. Shape Matching by Segmentation Averaging. – IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. **32**, 2010, No 4, pp. 619-635.
16. Paris, S., F. Durand. A Topological Approach to Hierarchical Segmentation Using Mean Shift. – In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'07), Minneapolis, MN, 2007.
17. Ma, Y., H. Derksen, W. Hong, J. Wright. Segmentation of Multivariate Mixed Data via Lossy Data Coding and Compression. – IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. **29**, 2007, No 9, pp. 1546-1562.
18. Fischler, M. A., R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. – Communications of the ACM, Vol. **24**, 1981, No 6, pp. 381-395.
19. Abhishek, A., S. K. Hema, J. Thorsten, S. Ashutosh. Contextually Guided Semantic Labelling and Search for 3D Point Clouds. – International Journal of Robotics Research January, Vol. **32**, 2013, No 1, pp. 19-34.
20. Deng, Z. H., T. T. Li, T. T. Zhang. An Adaptive Tracking Algorithm Based on Mean Shift. – Advanced Materials Research, Vol. **538**, 2012, pp. 2607-2613.
21. Li, Z., D. J. Mu, T. F. Zhang, X. L. Huang, M. Y. Fu. Design of Moving Target Detection and Tracking System Based on Cortex-A7 and Open CV. – In: Proc. of 2nd International Conference on Computational Intelligence, Communication and Signal Processing, South Korea, 2016, pp. 16-20.
22. Li, Z., H. X. Zhang, D. J. Mu, L. T. Guo. Random Time Delay Effect on Out-Of-Sequence Measurements. – In IEEE ACCESS Analysis and Synthesis of Large-Scale Systems, 2016 (In Press). DOI 10.1109/ ACCESS.2016.2610098.