

Object Pose Estimation for Robot Loading in Accommodation Space Using Alpha-Shape Algorithm

Qingda Guo (✉ qingda0501@163.com)

South China University of Technology <https://orcid.org/0000-0001-5863-5541>

Lulu Tang

University of Macau

Jianchi Zhang

South China University of Technology

Research Article

Keywords: Robot, Pose estimation, Alpha-shape algorithm, Fruit fly optimization, Accommodation space

Posted Date: April 12th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-309578/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Object pose estimation for robot loading in accommodation space using alpha-shape algorithm

Guo Qingda^{1*}, Tang Lulu², Zhang Jianchi¹

¹School of Electronic and Information Engineering, South China University of Technology, Guangzhou, 510640, China

²State Key Laboratory of Internet of Things for Smart City, University of Macau, Macau SAR, China

E-mail: guoqd@scut.edu.cn; lulu.tang@connect.um.edu.mo; msj.c.zhang@scut.edu.cn;

Received **.**, 2021; Revision accepted **.**, 2021; Crosschecked **.**, 2021;

Abstract: Robots with visual sensors have been used in various goods logistics, such as bin picking or uploading. However, there are more and more demands for the automatic blanking and loading, it is necessary to solve the problem of object pose estimation in changing accommodation space. This paper proposes a method for pose estimation in the accommodation space using alpha-shape algorithm and improved Fruit fly Optimization Algorithm (FOA). The alpha-shape volume variety of object and measured space is set to the objective function and the pose variety of object is set to six variables of improved FOA. The experiments were performed by setting parameters of improved FOA and considering the four space types represented the common accommodation shapes. Compared with previous work using convex hull, the new study using alpha-shape algorithm not only keeps the object in the accommodation space, but also maintains the object pose is at the bottom of the space and can meet the practical requirement of object placement by robot arms.

Keywords: Robot, Pose estimation, Alpha-shape algorithm, Fruit fly optimization, Accommodation space

1 Introduction

With the increase of labor cost and the automation demand of production line equipment, industrial robots with stereo vision are applied to product processing, especially for bin picking and workpiece loading, such as the picking competition held by Amazon from the box or on the shelf. However, there are more and more demands for the automatic blanking and placement of objects in the "reversed picking process". Limited by the visual perception technology of industrial robots, the packing process is usually set by manual procedure or based on artificial prior knowledge. For example, objects are packaged into simplified geometric shapes (cubes, cylinders, etc.) and then loaded into the boxes with known shapes and fixed positions. For some industrial robots with complex shape and elastic objects, robot engineers often set up the procedure of goods placement. Wynright company of the United States [1] built a mobile robot system for automatic loading of tires in containers, but there will still be unexpected space changes in the container body and thus remaining

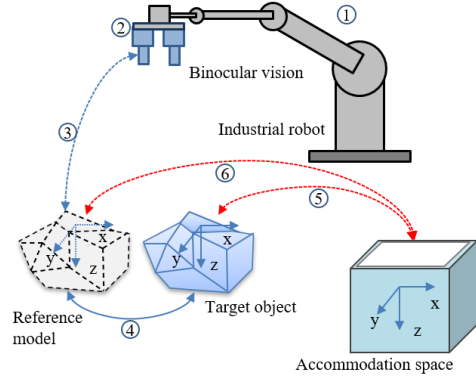
space (accommodation space). The key problem of automatic loading in an irregularly changing space is to obtain the pose of a target object in the remaining accommodation space, which eventually contains at least one object.

The effective pose matrix of the object in the accommodation space is the last crucial step in the process of intelligent robot's picking, running (obstacle avoidance) and placing. Automatic loading [2] using industrial robot needs the hand-eye matrix (①), pose matrix (②) between the camera coordinate system and workpiece coordinate system of reference model, pose matrix (③) between the reference model and the target model, and pose matrix (④) between the target object (or the reference model (⑤)) and the accommodation space, as shown in Fig. 1(a). In the past few years, object recognition of robot in bin picking has been a hot topic in the scientific research and robot companies. Many algorithms have been proposed in the aspect of 6 Degree Of Freedom (DOF) pose estimation, such as the application of Fast Point Feature Histograms (FPFH),

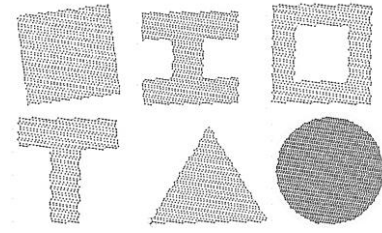
Signature of Histograms of Orientations (SHOT), LineMod algorithm with template [3], Point Pair Feature (PPF) with voting mechanism [4], DenseFusion [5], Pose-CNN based on regression [6], ssd-6d [7], etc.

The end-to-end trajectory of robot is also a hotspot research, which focuses on the trajectory planning based on optimal time or minimum energy. These planning algorithms mostly establish the dynamic and kinematic constraints of robot, using intelligent optimization algorithm to achieve trajectory acquisition, or apply sensors to generate Otscomap or three-dimensional (3D) data environment to establish space obstacle avoidance trajectory. Robot is usually based on the pre-setting pose matrix when it moves to the place where objects can be placed. Human beings have great flexibility in the operation of object placement, such as choosing a reasonable place for the object according to the size of the object or the environment on the plane, and packing in a limited space, which can be best placed according to different objects.

Alpha-shape algorithm is used to reconstruct object surface from an unorganized point cloud. The method was proposed in 2D points Edelsbrunner and was then extended to 3D points [8-9]. Compared with convex hull, alpha-shape can be used to reconstruct object shape of nonconvex body, as show in Fig.1(b). Alpha-shape was widely used in 3D object shape. Zhu et al. [10] proposed a novel approach for tree crown reconstruction based on an improvement of alpha shape modeling, where the data are points unevenly distributed in a volume rather than on a surface only. Lou et al. [11] used alpha-shape to extract topographical features from engineering surfaces, found that the alpha-shape method was more efficient in performance for large structuring elements. Santos et al. [12] proposed an adaptive method which estimates a local parameter for each edge based on local point spacing and used the method to extract the building roof boundaries from LiDAR Data.



(a) Structure diagram of automatic loading using industrial robots in the remaining accommodation space [2]



(b) The applications of alpha-shape algorithm [13]

Fig. 1. Structure diagram of the system and the applications

Based on previous study [2], this paper still aims to solve the problem of object pose estimation in the accommodation space assuming that accommodation space can hold one target object. The alpha-shape algorithm and improved FOA [14] are used to determine the object's final state given a pose matrix. Firstly, the alpha-shape algorithms are introduced, then, the alpha-shape volume variety of object and measured space is set to the objective function and the pose variety of object is set to six variables of improved FOA. Next, in the simulation experiments we at first present pose estimation process in cube space and obtain the convergence curves of four different spaces (namely the cube, hemisphere, cylindrical, and triangular prism) and the object's final pose in the four spaces. Finally, the reliability of the proposed method is verified experimentally, comparisons with previous work are given, and conclusions are drawn.

2 Proposed method

2.1 Alpha-shape algorithm

Alpha-shape algorithm can reconstruct geometry from a set of discrete points, as shown in Fig. 2. A circle with radius value α is used to scroll around the point set S.

After traversing all the points, the inner and outer contours of S can be obtained. When the radius α is large, the circle rolls outside S , and the trace of the external roll is the boundary contour of the point set. When α is small, the circle will roll to the interior of S . When α is small enough, every point in S is a boundary point. The value of circle radius α is closely related to the fineness of the detected points contour. When the radius is relatively small, the detected contour is relatively fine, and when the radius is relatively large, the detected contour is slightly rough.

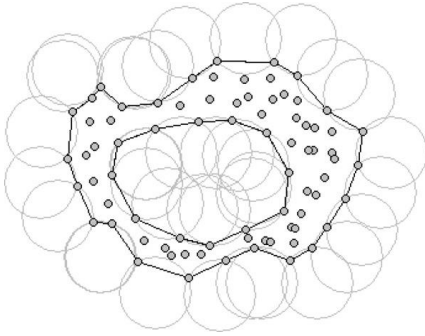


Fig. 2. Alpha shape of the planar point set

2.2 Pose estimation algorithm

When an industrial robot picks, runs, and places the target object in the accommodation space, the shape of accommodation space generally can be generally convex. The swarm intelligence optimization algorithm usually requires an objective function to determine the direction of iterative optimization. Compared with previous work [2], we just use the volume of alpha-shape to establish an objective function of FOA.

In the process of placing the object in the accommodation space, there are two groups of point clouds (the object and the accommodation space), in Fig. 3(a-c). The two groups of points are combined into one group, and the change of relative position can be expressed by the whole volume value V_t . The whole volume value V_t of the optimal position is minimal as the object are placed at the bottom of the accommodation space.

The fitness function of FOA is expressed by:

$$S_{\min} = V_t . \quad (1)$$

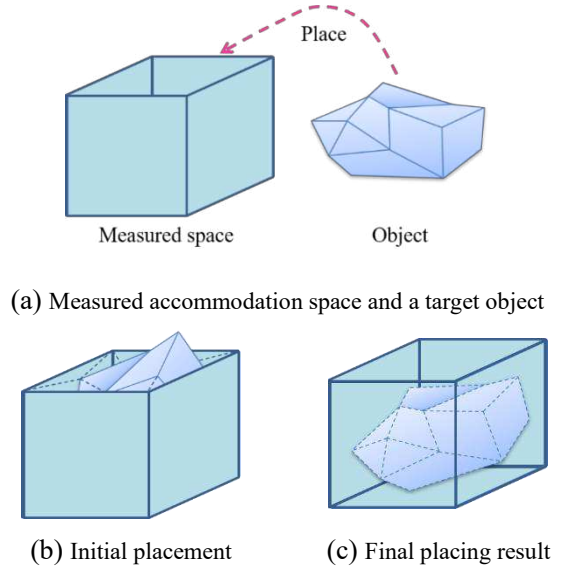


Fig. 3. The process of object placing into the measured space

Based on previous work [2], the target rotation is regarded as a separate rotation along each axis, and the posture adjustment of the object is simplified into three parameter variables, whose rotation matrixes are expressed as:

$$\mathbf{R}_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix} , \quad (2)$$

$$\mathbf{R}_y(\beta) = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix} , \quad (3)$$

$$\mathbf{R}_z(\gamma) = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} , \quad (4)$$

where θ , β and γ are the rotation parameters of x , y and z axes, respectively, and $\mathbf{R}_x(\theta)$, $\mathbf{R}_y(\beta)$ and $\mathbf{R}_z(\gamma)$ are the corresponding rotation matrixes, respectively.

To overcome local optimum, we still use multiple individuals to search for a global optimum, as shown in Fig. 4. We start by defining the point cloud of the target object as $O(x, y, z)$ and the point cloud of the measured accommodation space as $P(x, y, z)$.

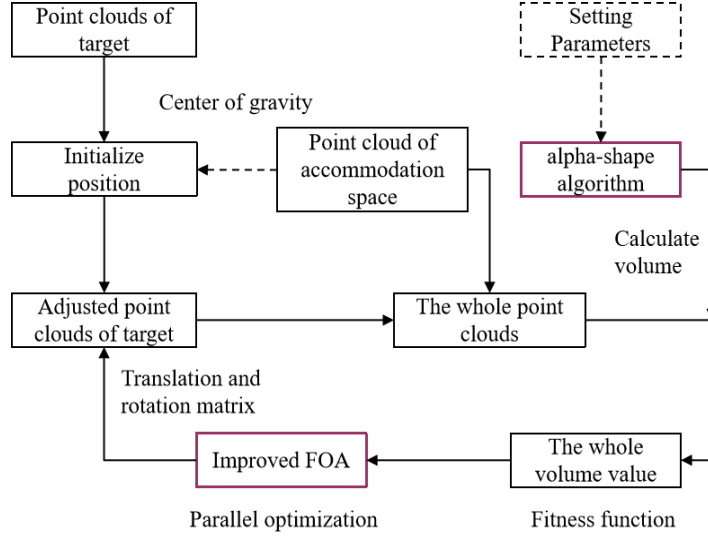


Fig. 4. Flowchart of pose estimation in the accommodation space based on improved FOA.

(1) Initialize the FOA parameters:

$$O_g(\bar{x}_m, \bar{y}_m, \bar{z}_m) = \sum_{i=1}^m O_i(x_i, y_i, z_i) / m, \quad (5)$$

$$P_g(\bar{x}_n, \bar{y}_n, \bar{z}_n) = \sum_{i=1}^n P_i(x_i, y_i, z_i) / n, \quad (6)$$

$$[O^1, 1]^T = \begin{bmatrix} E & O_g - P_g \\ 0 & 1 \end{bmatrix} \cdot [O, 1]^T, \quad (7)$$

$$V^t = \text{alphaShape}([O^1; P_g]), \quad (8)$$

where m and n are the corresponding point cloud numbers, E is the 3 identity matrix, V^t is the volume of the alpha-shape algorithm result for the two point clouds, and $\text{alphaShape}()$ represents the alpha-shape algorithm.

(2) Initialize the population position parameters of FOA:

$$\begin{cases} X_i^o = \bar{x}_n \\ Y_i^o = \bar{y}_n \\ Z_i^o = \bar{z}_n \end{cases}, \quad (9)$$

$$\begin{cases} \alpha_i^o = 2\pi \cdot \text{rand} - \pi \\ \beta_i^o = 2\pi \cdot \text{rand} - \pi \\ \gamma_i^o = 2\pi \cdot \text{rand} - \pi \end{cases}, \quad (10)$$

where rand denotes a random number within the range $(0, 1)$, X_i^o , Y_i^o , and Z_i^o denote the coordinates of the i^{th} initial given position, and α_i^o , β_i^o , and γ_i^o denote the i^{th} initial random angle parameters.

(3) The fruit flies use olfactory cues to search for the food in random directions and distances:

$$\begin{cases} X_i = X_i^o + a_1 \cdot \text{rand} - b_1 \\ Y_i = Y_i^o + a_2 \cdot \text{rand} - b_2 \\ Z_i = Z_i^o + a_3 \cdot \text{rand} - b_3 \end{cases}, \quad (11)$$

$$\begin{cases} \alpha_i = \alpha_i^o + a_4 \cdot \text{rand} - b_4 \\ \beta_i = \beta_i^o + a_5 \cdot \text{rand} - b_5 \\ \gamma_i = \gamma_i^o + a_6 \cdot \text{rand} - b_6 \end{cases}, \quad (12)$$

where $a_1 \sim a_6$ and $b_1 \sim b_6$ are constants used to constrain the range of random numbers.

(4) The minimum volume of alpha shape of the object and the accommodation space is taken as a judgment value of the taste concentration:

$$R_i = R_z(\gamma_i) \cdot R_y(\beta_i) \cdot R_x(\alpha_i), \quad (13)$$

$$T_i = [X_i, Y_i, Z_i]^T, \quad (14)$$

$$[O_i^2, 1]^T = \begin{bmatrix} R_i & T_i \\ 0 & 1 \end{bmatrix} \cdot [O_i^1, 1]^T, \quad (15)$$

$$PO_i(x, y, z) = [O_i^2; P], \quad (16)$$

$$V_i^t = \text{alphaShape}(PO_i(x, y, z)), \quad (17)$$

$$S_i = V_i^t. \quad (18)$$

(5) The groups are sorted in ascending order according to the concentration value:

$$[S_s \ S_index] = \text{sort}(S). \quad (19)$$

Where S_s is the volume value after sorting the groups and S_index is index of the corresponding individual.

(6) Half individuals with smaller concentration values are selected and marked again according to the concentration values:

$$\begin{cases} X_j = X(S_index) \\ Y_j = Y(S_index) \\ Z_j = Z(S_index) \end{cases}, \quad (20)$$

$$\begin{cases} \alpha_j = \alpha(S_index) \\ \beta_j = \beta(S_index) \\ \gamma_j = \gamma(S_index) \end{cases}, \quad (21)$$

where the range of S_index is determined by the number of individuals in half population, (X_j, Y_j, Z_j) denote new position, $(\alpha_j, \beta_j, \gamma_j)$ denote new posture and subscript j is index of individual.

(7) After step (6), return to step (3), and then to step (4); random direction and location are updated to obtain the taste concentration of a new half individuals. The pre-selected half individuals and the new half individuals are combined, and then all individuals are sorted by the taste concentration according to step (5). The algorithm will stop the iterative optimization loop until the end condition is satisfied.

3 Experimental results

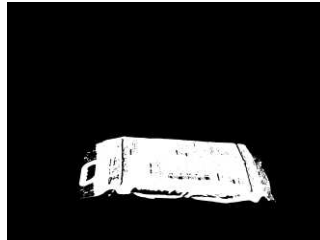
3.1 Obtaining the target object

In order to obtain the sparse point cloud of the target, the object was placed in the common field of view of

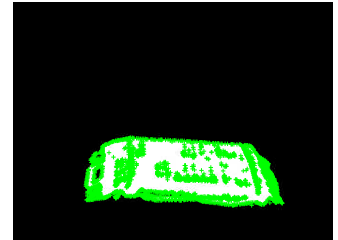
the left and right cameras, and the object images of the left and right cameras were obtained, as shown in Fig. 5 (a). According to the point cloud acquisition process, Otsu threshold segmentation was performed on the object image of the left camera; the background was removed, and region of the object was retained, as shown in Fig. 5 (b). Harris feature points were detected on the left image after threshold segmentation, as shown in Fig. 5 (c). The feature points of the right image were tracked by the obtained left image feature points, and the positions of features point pairs was obtained using KLT optical flow [15], as shown in Fig. 5(d). According to the corresponding feature point positions of left and right images and camera calibrated parameters, the top point clouds in the left camera coordinate system were calculated and finally rotated and translated into the world coordinate system, as shown in Fig. 5(e). The bottom point cloud was obtained using the same method and the top point cloud and the bottom point cloud were combined to a complete point cloud of the target object based the world coordinate system.



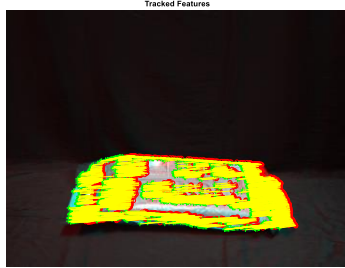
(a) left camera image



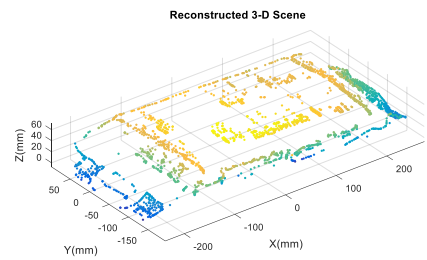
(b) OTSU threshold of left image



(c) Harris feature points of image



(d) feature point tracking



(e) the top point cloud of the target object

Fig.5. The process of point cloud derived from binocular vision.

3.2 Parameters setting result analysis

The initial parameters of the improved FOA are population size, number of iterations, and random direction and position parameters ($a_1 \sim a_6$ and $b_1 \sim b_6$).

In the experiments, the number of iterations is set to be 100 and the radius α is set to be 250. The range of random position variation was limited to $(-1, 1)$, and the range of random posture variation was limited to $(-0.05$

$\pi, 0.05\pi$); thus $a_1 \sim a_3$ and $b_1 \sim b_3$ are set to be 2 and 1, respectively, $a_4 \sim a_6$ and $b_4 \sim b_6$ are set to be 0.1π and 0.05π , respectively. Software environment is Win10 64bit and MATLAB 2015b, and the corresponding hardware configuration parameters were CPU i7-7700 and RAM 8GB.

Based on previous study, the point cloud of the target object is obtained by the Harris feature point detection

and KLT optical flow, in Fig. 6(a). 3D accommodation space is set as a cuboid with the size of $550 \times 100 \times 300 \text{ mm}^3$, in Fig. 6(b). The population number of fruit flies is 100. Figure 6(c) shows the initial state; Fig. 6(d) shows the 10th iteration; Fig. 6(e) shows the 20th iteration; Fig. 6(f) shows that pose adjustment in the accommodation space is completed at the 40th iteration.

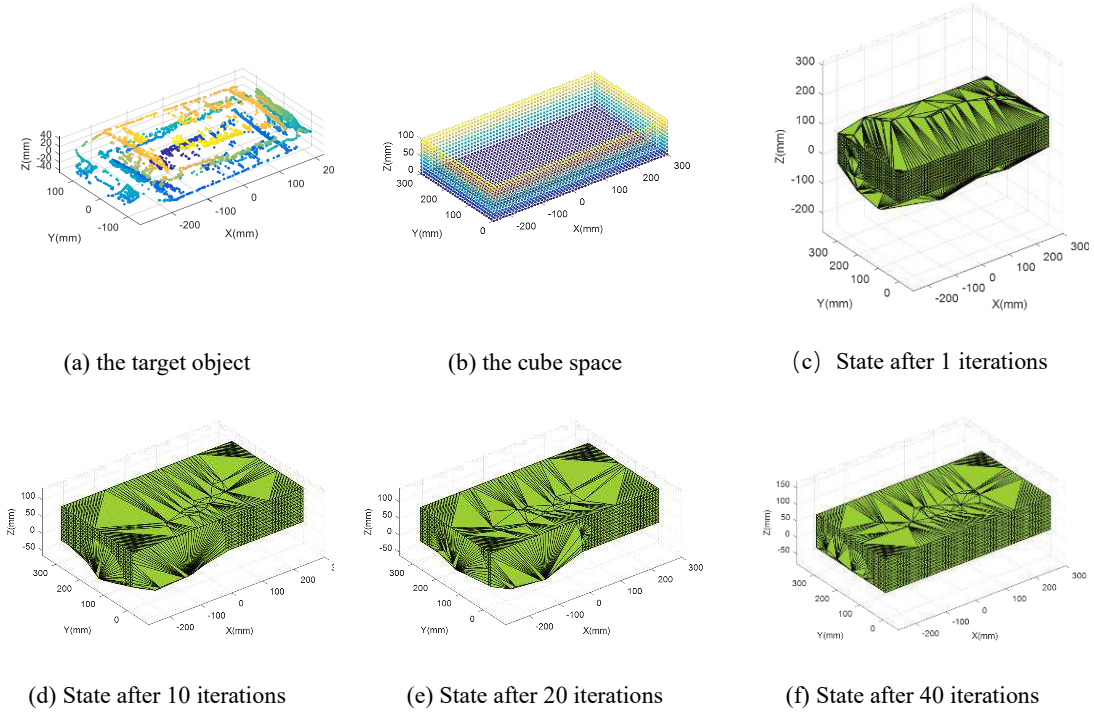


Fig. 6. Iteration process and its final results

The space types are designed only to accommodate the object and the target object's final posture in those three kinds of space as shown in Fig. 8. The four space types, namely the cube, hemisphere, cylindrical, and triangular prism, represent different shapes of the common accommodation space. The proposed method can be applied to different convex shaped spaces. The iterative convergence curves of the four space types are plotted in Fig. 7. The four curves show that the object is basically placed in four spaces after iterating the 10th iteration. To test the effect of different value of parameter α on the estimation pose, parameter α values are set to be 60, 80, 100, 150, respectively. As shown in Fig. 9, there will be different results for setting different values of parameter α and it is important to choose proper α value.

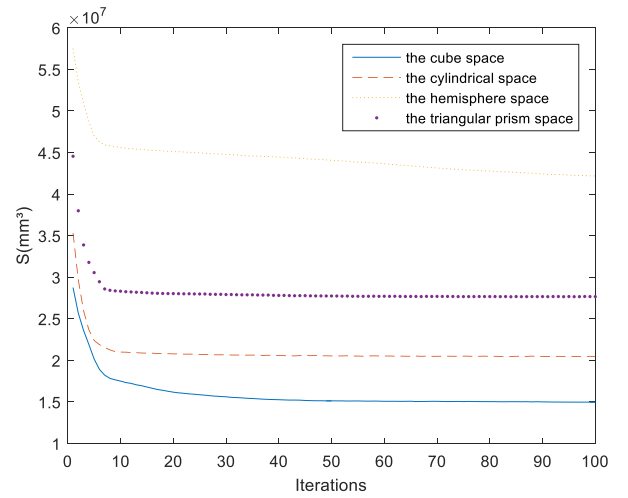
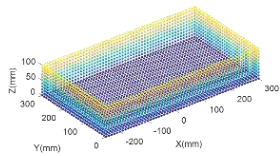
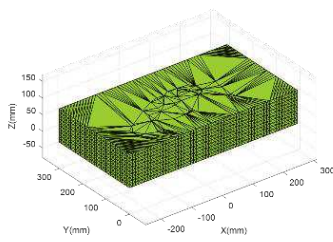


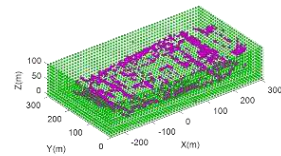
Fig. 7. The iterative convergence process



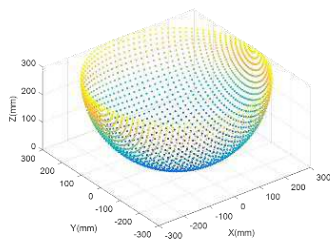
(a) the cube space



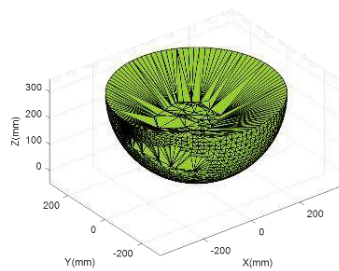
(b) the alpha shape of the final object in the cube space



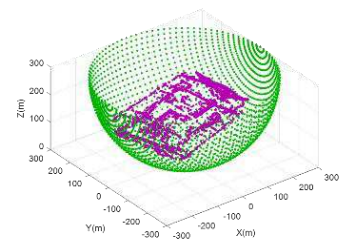
(c) object in the cube space



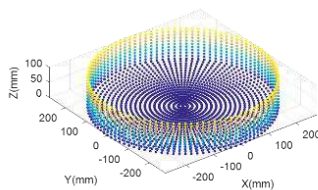
(d) the hemisphere space



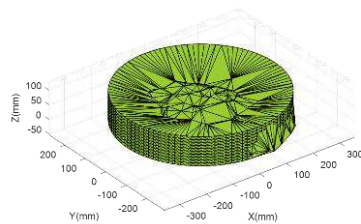
(e) the alpha shape of the final object and the semicircle space



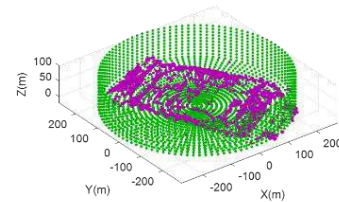
(f) object in the hemisphere space



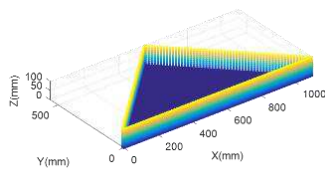
(g) the cylindrical space



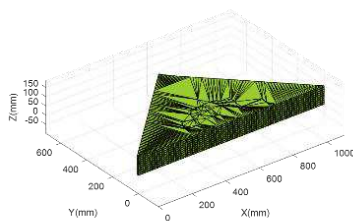
(h) the alpha shape of the final object and the cylindrical space



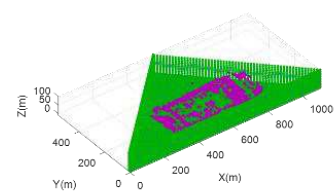
(i) object in the cylindrical space



(j) the triangular prism

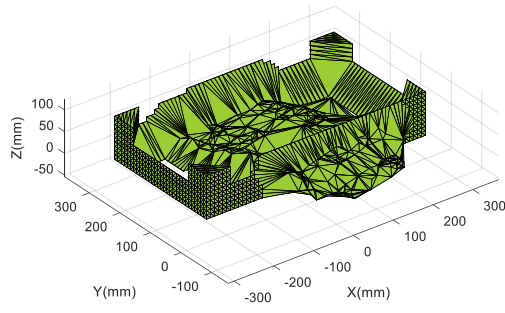


(k) the alpha shape of the final object and the triangular space

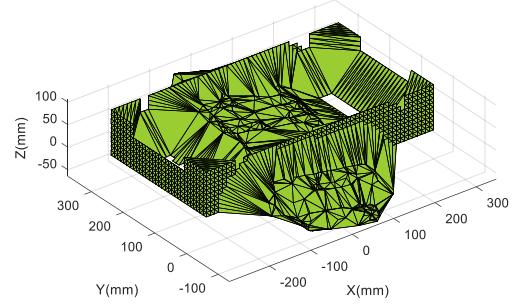


(l) Object in the triangular prism

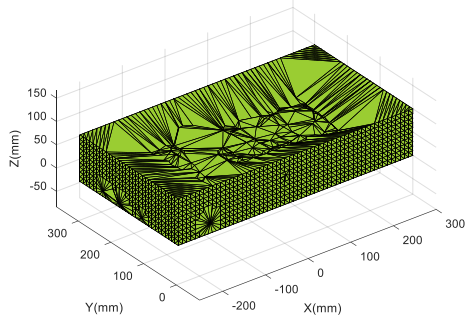
Fig. 8. The designed types of accommodation space



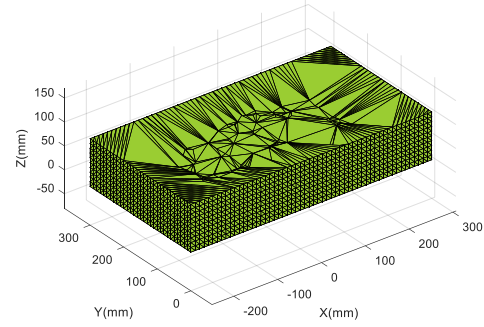
(a) $\alpha=60$



(b) $\alpha=75$



(c) $\alpha=100$



(d) $\alpha=150$

Fig.9. The alpha shape with different parameter α

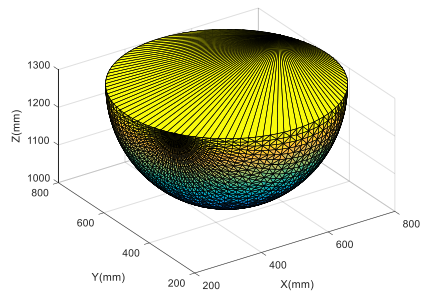
3.3 Method comparison

Our previous work uses convex hull to construct the point clouds of the object and the accommodation space. Object final pose in the hemisphere space is computed using previous method, as shown in Fig. 10. Comparing Fig. 8 (e-f), Fig.10(d) with Fig.10 (a-c), the result shows that alpha shape algorithm is more suitable for the placement requirements of robot real scenes than the convex hull algorithm. Previous work only considers that the accommodation space meets the placement of the target object, but not gives the best final pose in practical application; Our new study using alpha shape algorithm not only keeps the object in the

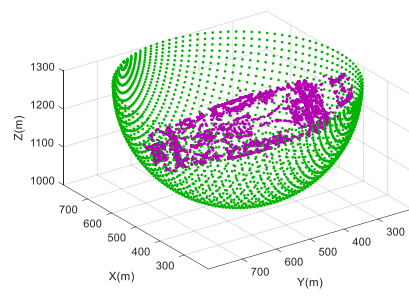
accommodation space, but also maintains the object pose is at the bottom of the space and can meet the actual robot placements requirements.

3.4 Hardware test

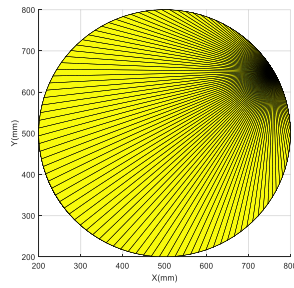
We used the SIASUN industrial robot (model: SR6C), binocular vision (or 3D vision sensor, model: Astra Mini), and vacuum sucker to establish the hardware test platform shown in Fig. 11(a). A more complex-shaped object and a randomly cube box were considered, and their corresponding point clouds were derived using a 3D sensor, Figs. 11 (b)(c). The final pose of the complex object in the box space was estimated using the proposed method, as shown in Fig. 11(d).



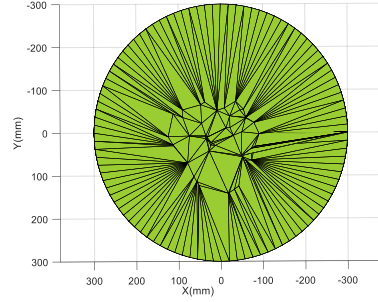
(a) the convex hull of the final object and the hemisphere space



(b) object in the hemisphere space

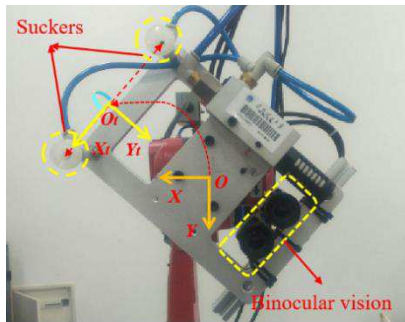


(c) top view of the convex hull

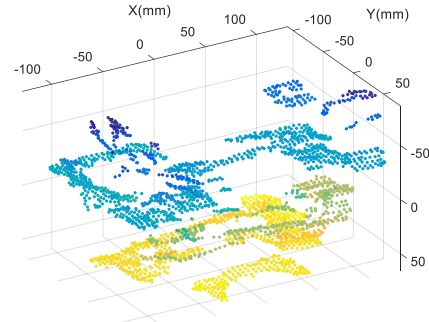


(d) top view of the alpha shape

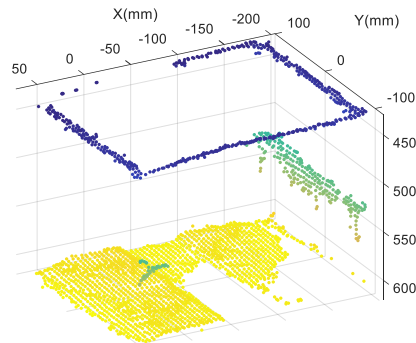
Fig.10. the object pose estimation using convex hull



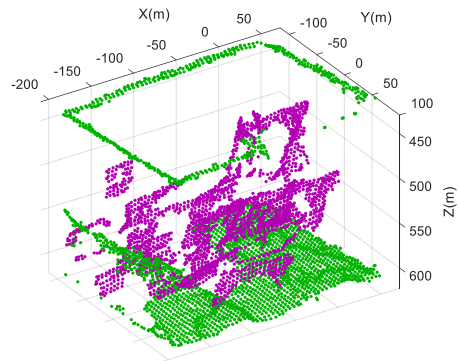
(a) The end-effector of the industrial robot



(b) Point cloud of the complex-shaped object



(c) Point cloud of the cube box



(d) The complex object pose in the cube box space

Fig. 11. Hardware platform test of the complex object

4 Conclusion

In this paper, the method of combining alpha-shape algorithm and improved fruit fly optimization is proposed to estimate the object posing in a 3D accommodation space. The proposed method uses the alpha-shape algorithm to establish the objective function, which uses the whole volume change to adjust the object pose. To obtain the best individual with 6 degrees of freedom, the iteration strategy of improved FOA choosing better half individuals to produce next half individuals. The experiments were performed setting parameters of improved FOA and considering the four space types and the obtained results show that proposed method can obtain the object pose in the common accommodation space. Compared with previous work using convex hull, the new study using alpha-shape could keep the object pose at the bottom of the accommodation space without limitation of gravity, which could meet the actual robot stacking. For an actual object placed in the enclosed space, the future study would increase the constraints (such as elasticity and collision) and adjust the parameters (or adds other algorithms) to shorten the estimation time. In addition, the objective function depends on the whole volume change of alpha-shape and parameter α directly related to point density and the level of detail of the boundary. As the proposed method needs choosing an appropriate parameter α , segmenting point clouds of the accommodation space and building 3D object, our future work will focus on using adaptive alpha-shape algorithm and obtaining the whole scene point clouds and the space point clouds segmentation.

Acknowledgments

This work was supported by Key-Area Research and Development Program of Guangdong Province, China (2019B010155001). In addition, thanks to all the people who are fighting the COVID-19.

Declarations

Conflict of interest All authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed

by any of the authors.

Authors' contributions

Guo Qingda contributed to the conception of the study and wrote the manuscript; Tang Lulu contributed significantly to analysis and manuscript preparation; Zhang Jianchi performed the experiment.

References

- [1] Tim Criswell, Arlington, TX (US). Automated Truck Unloader for Unloading Unpacking Product from Trailers and Containers: US, 2014/0205403 A1. 2014-07-24
- [2] Qingda Guo; Yanming Quan; Changcheng Jiang. Object Pose Estimation in Accommodation Space using an Improved Fruit Fly Optimization Algorithm[J], Journal of Intelligent & Robotic Systems, 2019, 95(2): 405-417.
- [3] Hinterstoisser S, Cagniart C, Ilic S, et al.. Gradient Response Maps for Real-Time Detection of Textureless Objects. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34(5):876-888.
- [4] Drost B, Ulrich M, Navab N, et al.. Model globally, match locally: Efficient and robust 3D object recognition. IEEE Conference on Computer Vision and Pattern Recognition, 2010.
- [5] Wang C, Xu D, Zhu Y, et al.. DenseFusion: 6D Object Pose Estimation by Iterative Dense Fusion. IEEE Conference on Computer Vision and Pattern Recognition, 2019.
- [6] Xiang Y, Schmidt T, Narayanan V, et al.. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. Robotics: Science and Systems, 2017.
- [7] Kehl W, Manhardt F, Tombari F, et al.. SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again. IEEE International Conference on Computer Vision, 2017.
- [8] Edelsbrunner H, Kirkpatrick D, Seidel R. On the shape of a set of points in the plane[J]. IEEE Transactions on Information Theory, 2003, 29(4):551-559.
- [9] Edelsbrunner H, Mücke, Ernst P. Three dimensional alpha shapes[J]. ACM Transactions on

Graphics, 1994, 13(1): 43–72.

[10] Zhu C , Zhang X , Hu B , et al.. Reconstruction of Tree Crown Shape from Scanned Data[C]. Proceedings of the 3rd International Conference on Technologies for E-learning and Digital Entertainment. Edutainment '08, Berlin, Heidelberg: Springer-Verlag, 2008: 745-756.

[11] Shan Lou, Xiangqian Jiang, Paul J. Scott. Application of the morphological alpha shape method to the extraction of topographical features from engineering surfaces[J]. Measurement, 2013, 46: 1002-1008.

[12] Santos R C D , Galo M , Carrilho A C . Extraction of Building Roof Boundaries From LiDAR Data Using

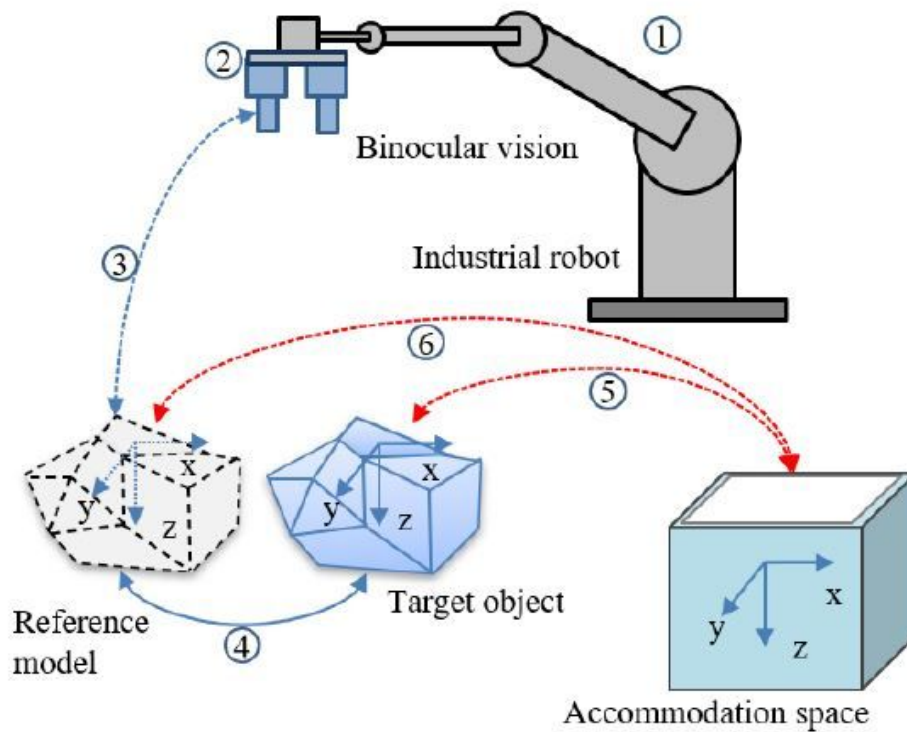
an Adaptive Alpha-Shape Algorithm[J]. IEEE Geoscience & Remote Sensing Letters, 2019:1-5.

[13] Wang Qiu-yan, Chen Xi-li. Building Contours Extraction from LIDAR Data Based on Alpha Shapes Algorithm[J]. Intelligent Building & Smart City, 2019(02):23-25.

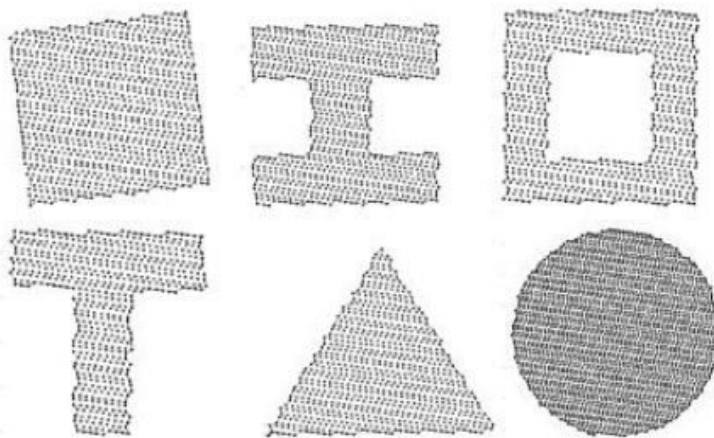
[14] Pan W T. A new Fruit Fly Optimization Algorithm: Taking the financial distress model as an example[J]. Knowledge-Based Systems, 2012, 26(2):69-74.

[15] Shi, Jianbo and Carlo Tomasi. Good Features to Track [C]. IEEE Conference on Computer Vision and Pattern Recognition, 1994: 593–600.

Figures



(a) Structure diagram of automatic loading using industrial robots in the remaining accommodation space [2]



(b) The applications of alpha-shape algorithm [13]

Figure 1

Structure diagram of the system and the applications

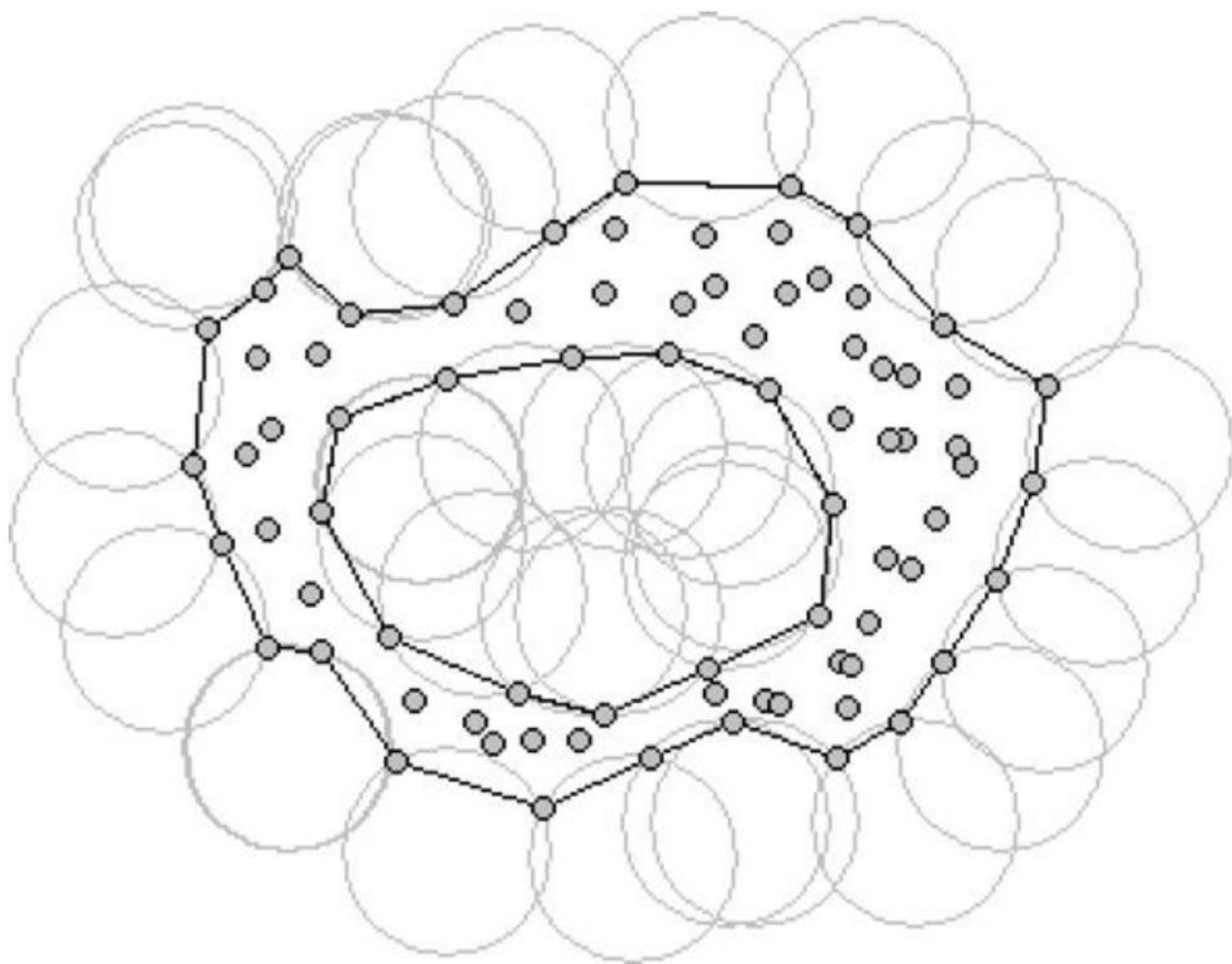
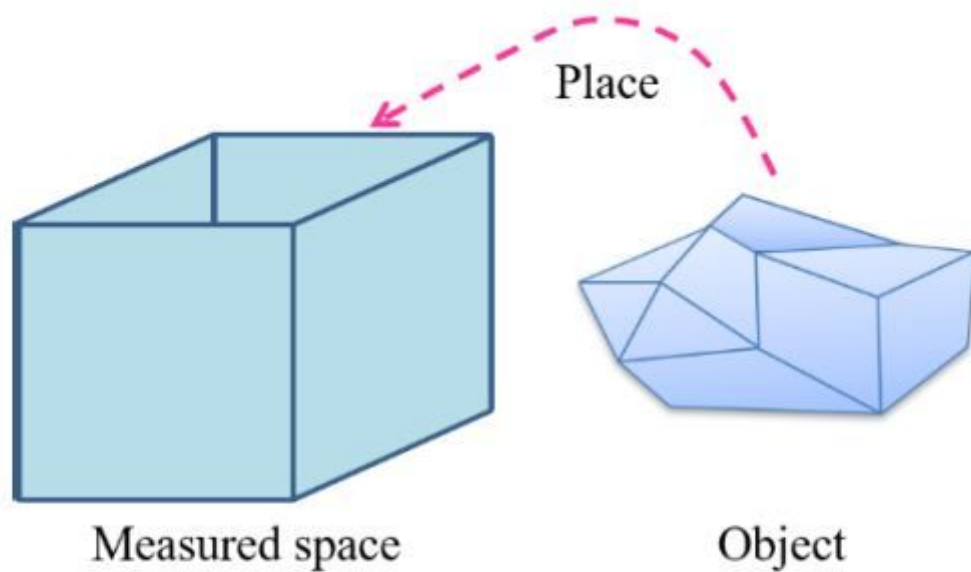
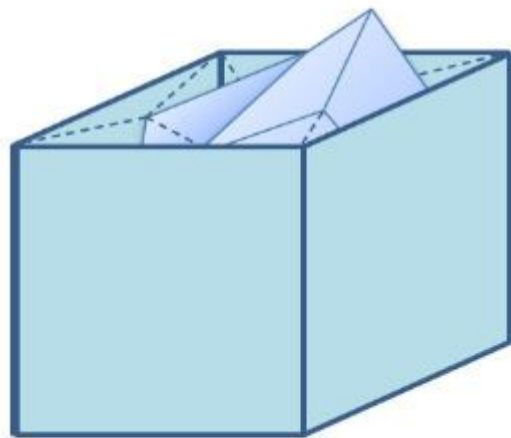


Figure 2

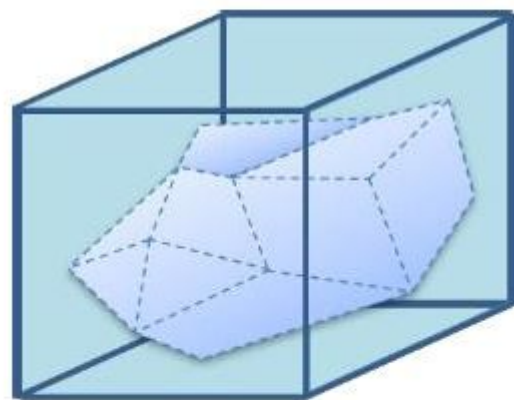
Alpha shape of the planar point set



(a) Measured accommodation space and a target object



(b) Initial placement



(c) Final placing result

Figure 3

The process of object placing into the measured space

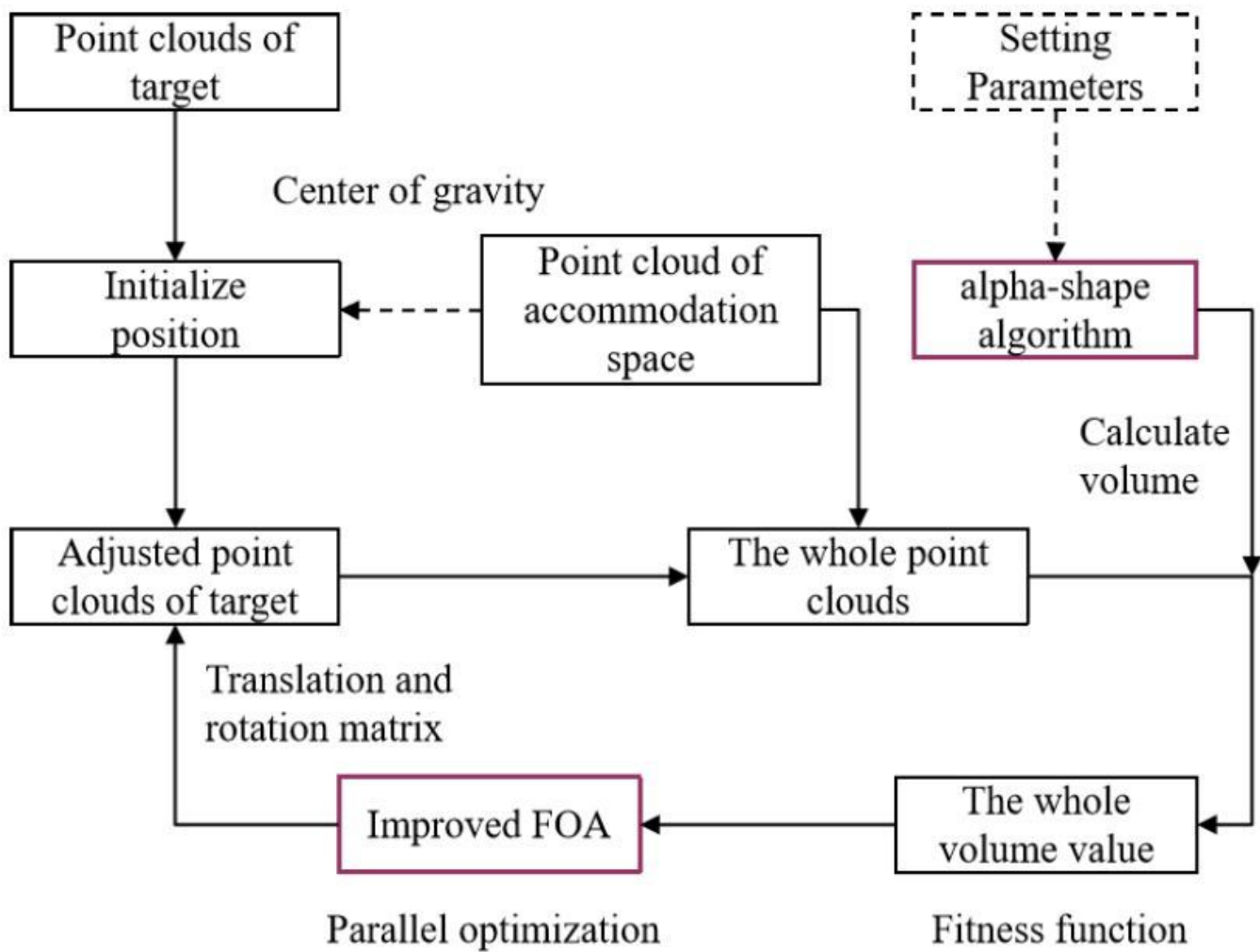
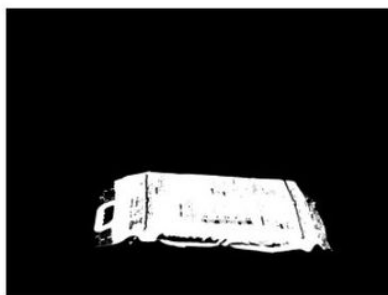


Figure 4

Flowchart of pose estimation in the accommodation space based on improved FOA.



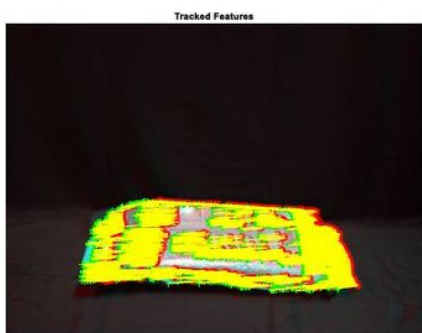
(a) left camera image



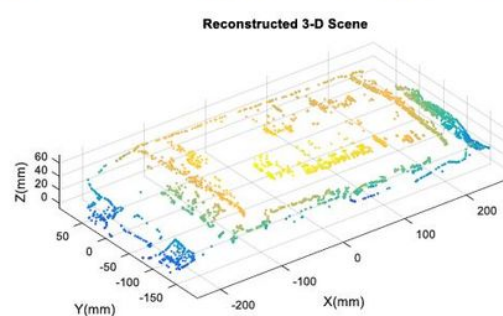
(b) OTSU threshold of left image



(c) Harris feature points of image



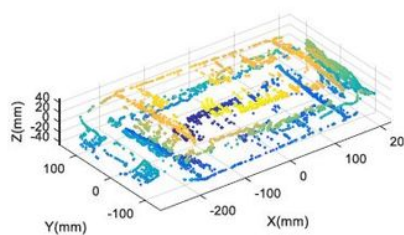
(d) feature point tracking



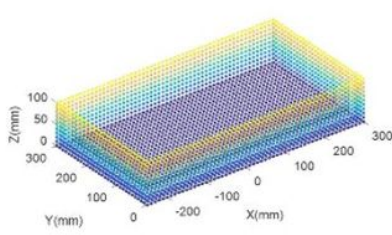
(e) the top point cloud of the target object

Figure 5

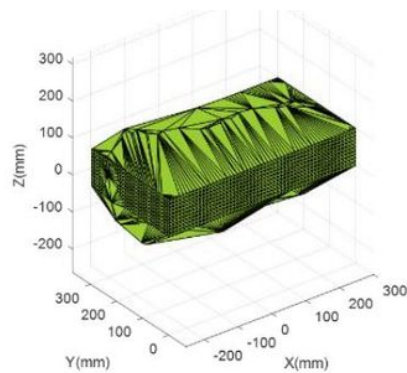
The process of point cloud derived from binocular vision.



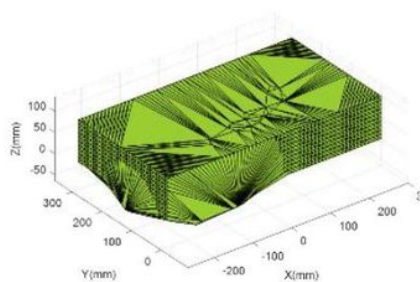
(a) the target object



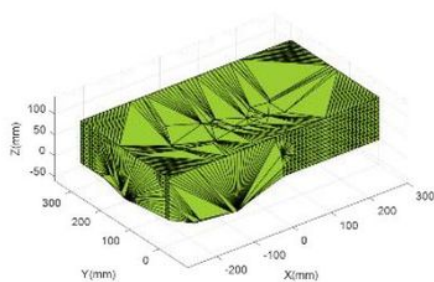
(b) the cube space



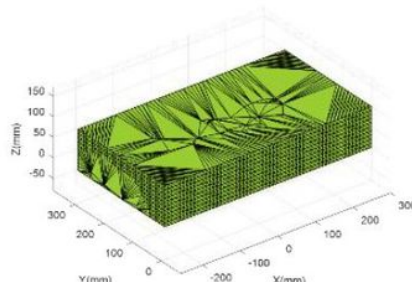
(c) State after 1 iterations



(d) State after 10 iterations



(e) State after 20 iterations



(f) State after 40 iterations

Figure 6

Iteration process and its final results

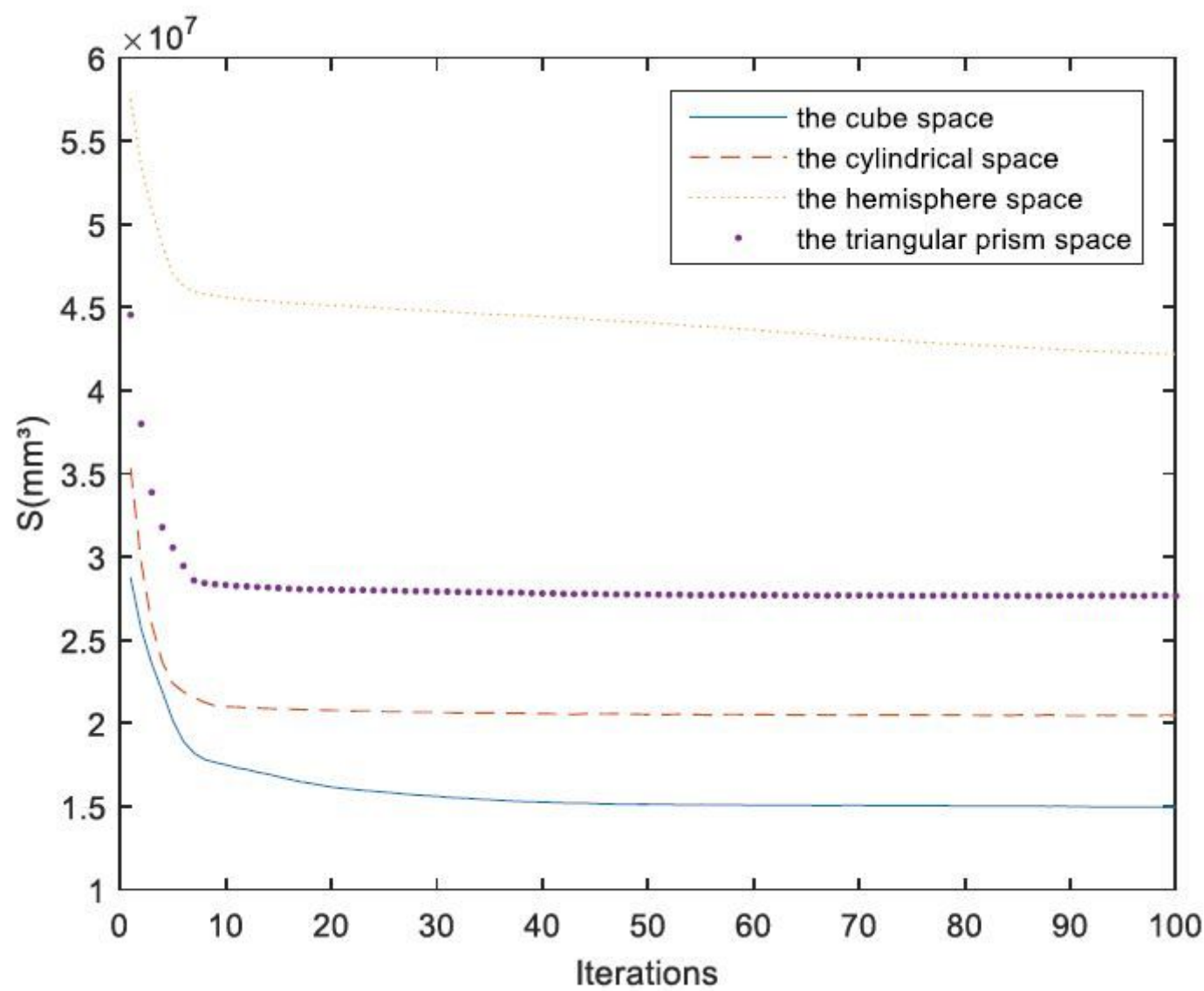
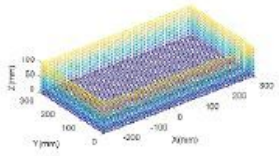
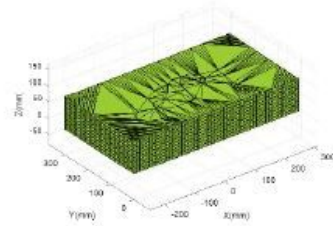


Figure 7

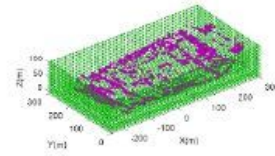
The iterative convergence process



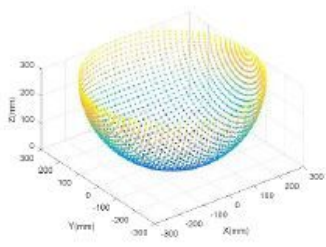
(a) the cube space



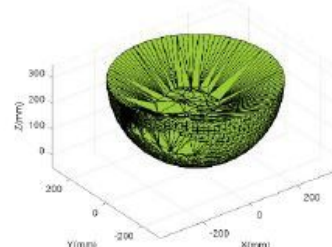
(b) the alpha shape of the final object in the cube space



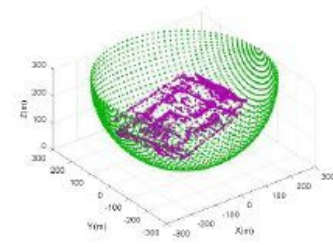
(c) object in the cube space



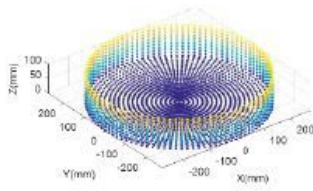
(d) the hemisphere space



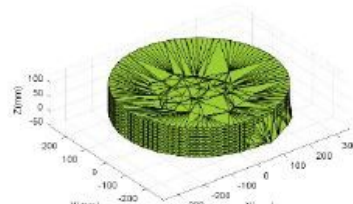
(e) the alpha shape of the final object and the semicircle space



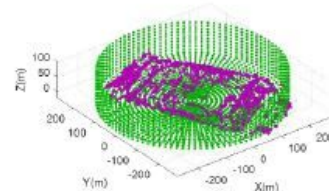
(f) object in the hemisphere space



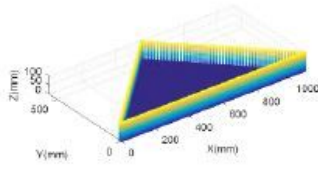
(g) the cylindrical space



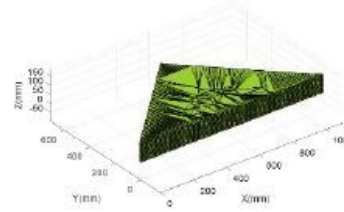
(h) the alpha shape of the final object and the cylindrical space



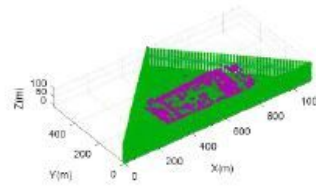
(i) object in the cylindrical space



(j) the triangular prism



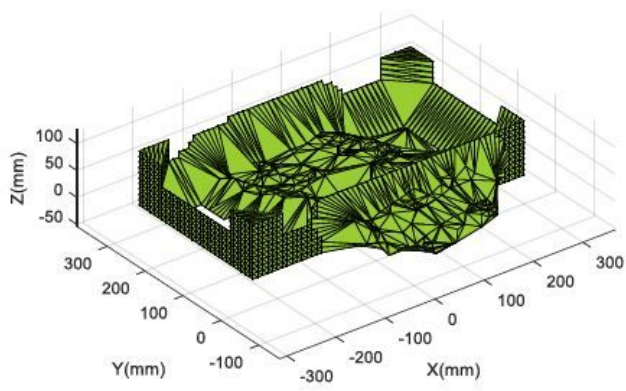
(k) the alpha shape of the final object and the triangular space



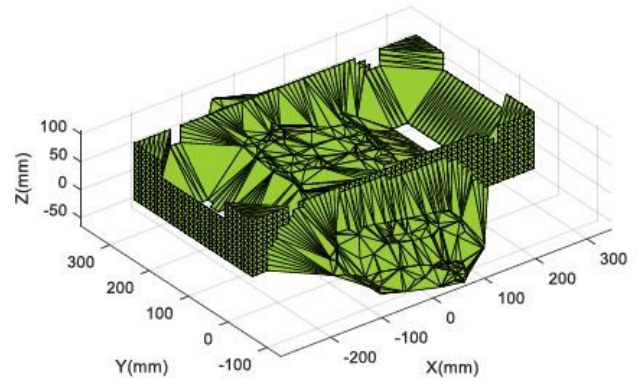
(l) Object in the triangular prism

Figure 8

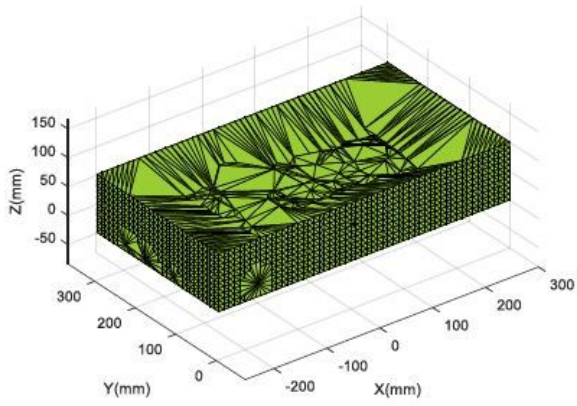
The designed types of accommodation space



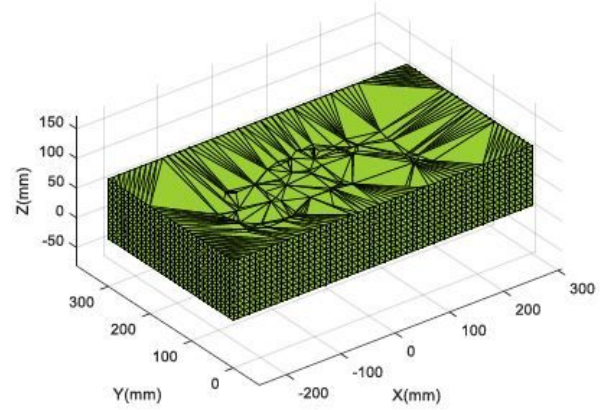
(a) $\alpha=60$



(b) $\alpha=75$



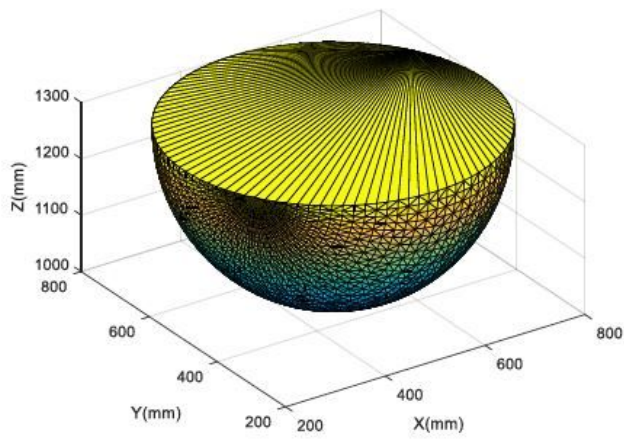
(c) $\alpha=100$



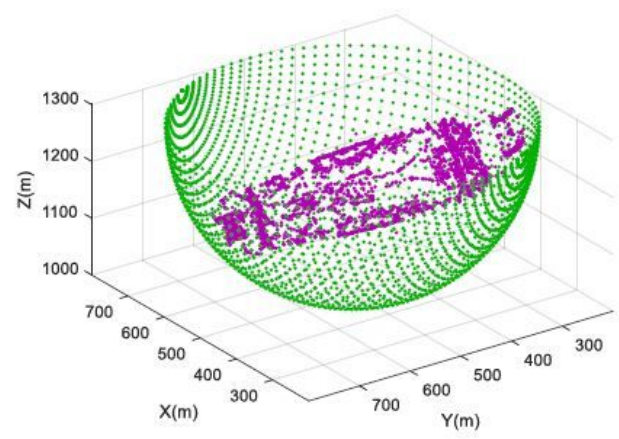
(d) $\alpha=150$

Figure 9

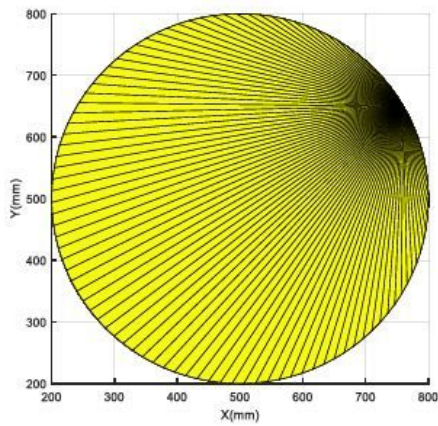
The alpha shape with different parameter α



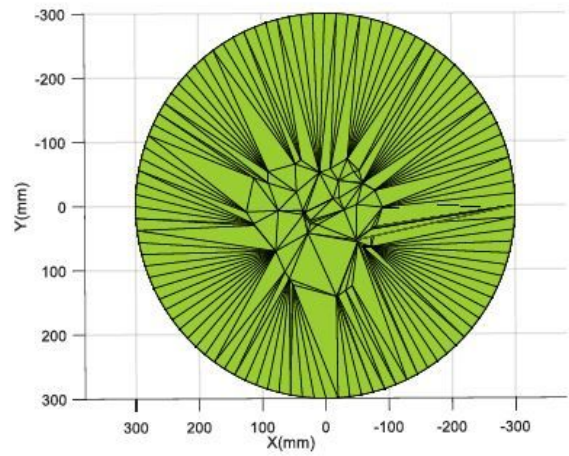
(a) the convex hull of the final object and the hemisphere space



(b) object in the hemisphere space



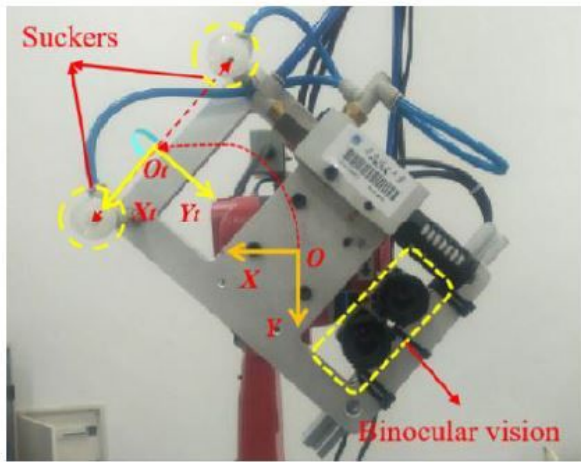
(c) top view of the convex hull



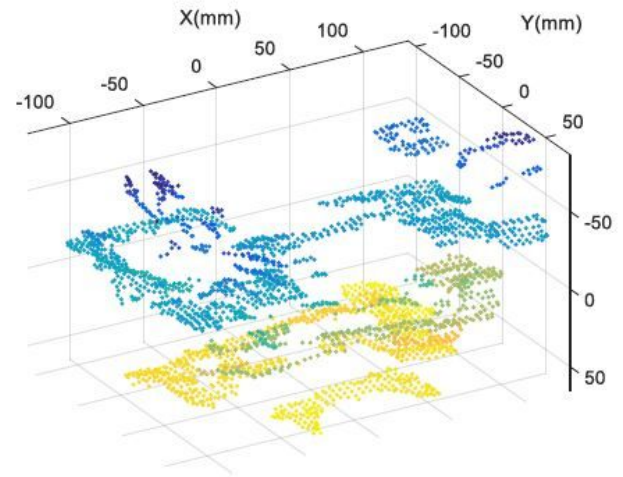
(d) top view of the alpha shape

Figure 10

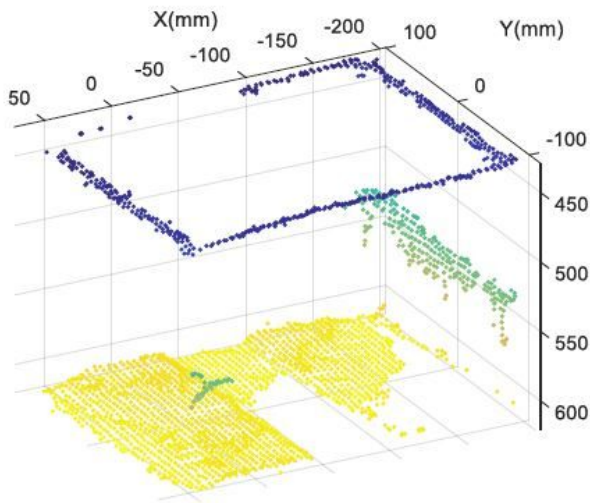
the object pose estimation using convex hull



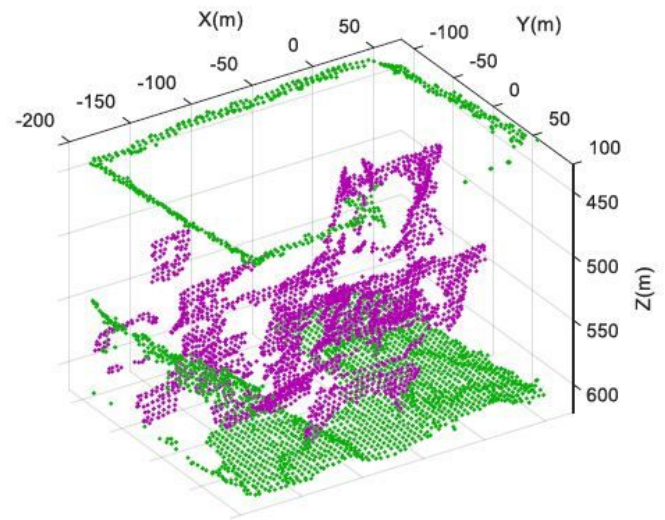
(a) The end-effector of the industrial robot



(b) Point cloud of the complex-shaped object



(c) Point cloud of the cube box



(d) The complex object pose in the cube box space

Figure 11

Hardware platform test of the complex object