Int. J. Fuzzy Syst.
https://doi.org/10.1007/s40815-018-00604-8

Adaptive Fuzzy Gaussian Mixture Models for Shape Approximation in Robot Grasping

Huifeng Lin[1] · **Tong Zhang**[2] · **Zhaopeng Chen**[3] · **Haina Song**[2,4] · **Chenguang Yang**[5]

Received: 16 October 2018 / Revised: 10 December 2018 / Accepted: 19 December 2018

Abstract Robotic grasping has always been a challenging task for both service and industrial robots. The ability of grasp planning for novel objects is necessary for a robot to autonomously perform grasps under unknown environments. In this work, we consider the task of grasp planning for a parallel gripper to grasp a novel object, given an RGB image and its corresponding depth image taken from a single view. In this paper, we show that this problem can be simplified by modeling a novel object as a set of simple shape primitives, such as ellipses. We adopt fuzzy Gaussian mixture models (GMMs) for novel objects' shape approximation. With the obtained GMM, we decompose the object into several ellipses, while each ellipse is corresponding to a grasping rectangle. After comparing the grasp quality among these rectangles, we will obtain the most proper part for a gripper to grasp. Extensive experiments on a real robotic platform demonstrate that our algorithm assists the robot to grasp a variety of novel objects with good grasp quality and computational efficiency.

Keywords Grasp planning · Novel object grasping · Fuzzy Gaussian mixture models · Shape approximation

1 Introduction

Recently, robotic grasping has gained increasing attention because it is fundamental for robots' manipulation task. Finding a proper grasp pose is of great importance for implementation of the grasping task. [1] converts the robotic grasping problem into a detection problem. They use an oriented rectangle in the image plane to present the seven-dimensional grasping configuration which involves the location, orientation and opening width of the gripper, as shown in Fig. 1. The rectangle is called a 'grasping rectangle.' This grasp detection method has been successfully applied on a real robotic platform. It is flexible to employ in a real scene because only RGB-D image of the object is needed. It overcomes the shortcoming of some previous related works [2–4] which are only available when the precise 3D model of the object and other physical information such as the friction coefficient are known in advance. However, the inefficiency of searching for a good grasp of this grasping detection strategy is a great drawback, which cannot meet the demand in real industrial scene.

Deep learning methods have shown great power in many fields, especially for the visual recognition [5]. Lenz et al. [6] adopts a deep learning approach to extract the grasping features for grasp detection from the Cornell Grasp Dataset [1]. Though their work is remarkable, the computation

Tong Zhang and Huifeng Lin are joint first authors and they contributed equally to this work.

✉ Chenguang Yang
cyang@ieee.org

[1] Key Laboratory of Autonomous Systems and Networked Control, College of Automation Science and Engineering, South China University of Technology, Guangzhou 510640, China

[2] School of Electronics and Information, South China University of Technology, Guangzhou 510640, China

[3] Robotics and Mechatronics Center, German Aerospace Center, DLR, Wessling, Germany

[4] Haihua Electronics Enterprise (China) Co. Ltd, Guangzhou 510670, China

[5] Bristol Robotics Laboratory, University of the West of England, Bristol, BS16 1QY, UK

Published online: 25 February 2019

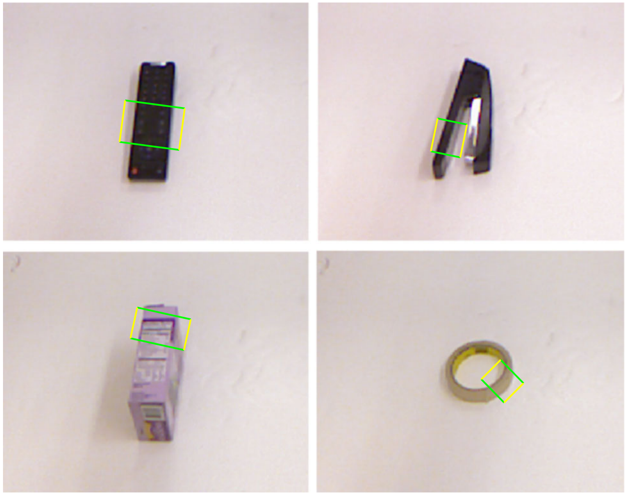


Fig. 1 Some example grasps for common objects, which are presented as oriented rectangles in 2D. Yellow lines represent the parallel plates of gripper, while green lines represent the opening width of the gripper before grasping

efficiency of their method is not satisfactory which needs 13.5 s to search for the best grasp in every single image. Using the same dataset, [7] employs the AlexNet [8] and [9] applies Resnet [10] for real-time grasp detection, both of which achieve great performance.

However, these deep learning-based approaches require a significant amount of computing capabilities. In the training phase, it takes several days on parallel high-performance GPUs to train a CNN (as the one in [7]) and takes several hours to fine-tune the network. In the testing phase, these deep networks need to run on a high-performance GPU for real-time grasp detection. These methods are computationally expensive. The performance requires large masses of manually labeled data, which is another limitation of these methods. And most importantly, the predicted grasp rectangle of these deep networks does not necessarily guarantee a stable grasp because it is solely learned from the training data.

This paper proposes a novel method for grasp planning based on shape approximation. First, we segment the object from the background by employing a proposed segmentation algorithm. Only use an RGB image for segmentation is hard to deal with problems such as indistinguishable background and shadows. To take advantage of the aligned depth image, we combine the RGB image and depth image to achieve a better segmentation. Second, an adaptive GMM-based shape approximation method will be employed to decompose the shape of the object into several ellipses. In order to accelerate the convergence speed of parameters estimation for GMM, we adopt the fuzzy EM algorithm instead of the normal EM algorithm. GMM with fuzzy EM algorithm by defining a dissimilarity function was introduced in [11], which they called fuzzy GMM. According to the experiment results of [11], fuzzy GMM can converge with less iterations and less computational time when compared to conventional GMM. By now, we have transformed the task of grasp planning into finding the most proper ellipse to grasp. The grasp for each ellipse can be represented as a grasping rectangle. In this algorithm, the number of components of GMM is adaptively chosen according to the complexity of the shape of the object. Considering the uncertainty of gripper pose caused by the inaccurate calibration between the robot and the camera [12], a pose error robust metric is proposed to evaluate the grasp quality for each grasping rectangle. We rank these grasping rectangles to obtain the most stable grasp under pose uncertainty. Finally, the best grasping rectangle is converted to the corresponding seven-dimensional gripper configuration according to the point cloud generated from the RGB image and depth image.

To sum up, the contributions of this work can be concluded as the following three points:

- An adaptive fuzzy GMM-based shape approximation method is proposed for robotic grasping. It does not need a training phase and has shown comparable performance with more computational efficiency compared to previous deep learning-based method.
- Taking the uncertainty in gripper pose into consideration, we introduce a grasp quality metric to obtain the most viable grasp.
- Experiments have been implemented on a real robotic platform which demonstrates the effectiveness of the proposed method.

The rest of this paper is structured as follows. We discuss related work about the robotic grasping in Sect. 2. Details of our proposed method are presented in Sects. 3 and 4. Section 5 shows the experimental results implemented with the proposed method, followed by the conclusion in Sect. 6.

2 Related Work

Precise information of 3D models and other physical information are required in most previous work. Based on these knowledge, methods focusing on force closure [13, 14] and form closure [15] aim to obtain theoretically stable grasps. Given the 3D model of an object, they tried to synthesize grasps fulfilling form closure and force closure and ranked them according to a specific grasp quality metric. The most commonly used metric is the epsilon quality (ϵ_{GWS}), which is corresponding to the radius of the maximum inscribed ball of the convex hull determined by the set of contact wrenches [16]. Some significant works [17–19] use physical simulation to find optimal grasps which also rely on a full 3D model. All the above approaches are theoretical methods instead of practical methods for grasp planning because the 3D model of the target object is usually impossible to obtain a priori.

Since inexpensive depth sensors like Microsoft Kinect are becoming available nowadays, RGB-D data have been leveraged in various robotics applications, like object detection and recognition [20, 21]. Though RGB-D data can only capture incomplete information of the object compared to the 3D model, it is more applicable in a real-world robotic setting. Recent work on robotic grasping focuses on finding appropriate grasps depending on RGB-D data of the object instead of its full physical model. [1] transforms it into a detection problem by encoding the seven-dimensional grasping configuration of a gripper into a 2D oriented rectangle in the RGB-D image. Two edges of the gripper are corresponding to the plates, and the surface normal of the point cloud is used to determine the grasp approach vector. In this paper, we will follow this representation of gripper configuration for the convenience of grasp planning.

Recently, deep learning method has shown powerful performance on multiple problems in computer vision, such as image classification [8], object recognition [22] and face verification [23]. It has been firstly introduced into grasp detection since Lenz's remarkable work [6]. [7] employs the AlexNet [8] and [9] applies Resnet [10] for real-time grasp detection, both of which achieve better performance. The deep learning models mentioned above are all trained on the Cornell Grasp Dataset [1]. All of these methods are computationally expensive in the training phase and testing phase and largely depend on the dataset. Since these methods only learn from labeled data, the stability of the planned grasp is not guaranteed. Furthermore, they give no consideration to the inevitable pose error for a robot to execute the grasp.

3 Data Acquisition And Preprocessing

3.1 Capturing Data From RGB-D Camera

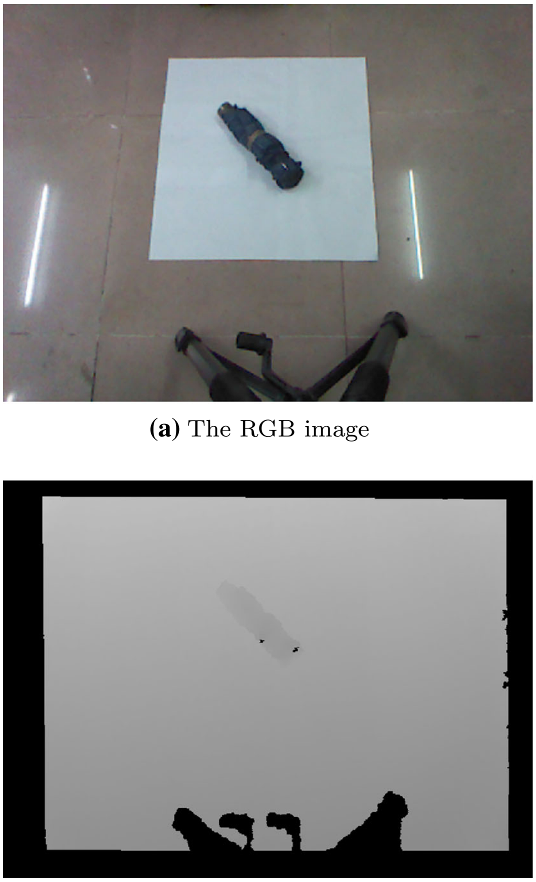
In this work, we use a Microsoft Kinect to obtain the raw 3D data. Kinect is commonly used to obtain the point cloud [9] since it is low cost. With a pair of additional infrared ray emitter and receiver, this RGB-D camera can acquire extra depth data compared to the common ones. The raw data generated by Kinect sensor are in the form of a depth image and an RGB image, as shown in Fig. 2. The additional depth information is used to recover the 3D coordinate [x, y, z] for every pixel of the RGB image, which is corresponding to each point in the point cloud. Use $P = \{p_j(x_j, y_j, z_j) | j = 0, 1, \ldots, m\}$ to denote the 3D point cloud and the coordinate of each point p_j is obtained by [24]:

$$\begin{aligned} x_j &= d_j(x_j^c - c_x)/f_x \\ y_j &= d_j(y_j^c - c_y)/f_y \qquad (1) \\ z_j &= d_j \end{aligned}$$

where (x_j^c, y_j^c) is the coordinate of p_j in the image coordinate frame, d_j is the depth value of p_j, c_x,f_x, c_y,f_y are Kinect sensor's intrinsic parameters. Figure 3 gives an intuitive explanation for Eq. 1. In this figure, p is in the image coordinate frame, while P is in the 3D camera coordinate frame. Equation 1 is employed to transform the coordinate of p to P.

3.2 Object Segmentation from Background

Raw data obtained by the Kinect consist of points belonging to the object and the background. The points belonging to the object are what we really need. For further processing, we must firstly segment the object from the background and the quality of the segmentation largely



(a) The RGB image

(b) The depth image

Fig. 2 Raw data captured by the Kinect

influences the performance of the following grasping planning [25].

At the beginning, a background image is taken from Kinect. An intuitive idea for segmentation is to subtract the background image from the foreground image using the following formulation,

$$\|I(x,y) - B(x,y)\|_2 > \tau \tag{2}$$

where I denotes the image which contains the object; B denotes the background image; $I(x,y) \in R^3$ denotes the intensity of the pixel (x, y) in I; and τ is a preset threshold. If the L_2 distance between these intensities is greater than the threshold τ, then this pixel is considered to belong to the object.

Since this method is solely based on pixel intensities, it cannot work as expected in scenes having indistinguishable background or shadows. It is easy to infer that only using color information is not an ideal method. To make full use of the depth data, we combine the RGB image and depth image to achieve a better segmentation. Denote the set of points belonging to the object as P_O, we have

$$P_O = \left\{(x,y) | \left\|\omega^T * (\bar{I}(x,y) - \bar{B}(x,y))\right\|_2 > \tau\right\} \tag{3}$$

In this formula, $I(x,y) \in R^3$ is augmented with depth value to become $\bar{I}(x,y) \in R^4$ and τ is a predefined threshold.

Considering that the RGB value and depth value may have different importance for segmentation, we introduce a weight vector $\omega \in R^4$ to assign weights to different elements of $\bar{I}(x,y)$ according to their importance. Experiments demonstrate that assigning greater weights to color intensities than the depth value results in better segmentation performance. According to the experimental results, setting weights to 0.4 for the depth channel and 0.6 for the color channels derives the best results.

The performance of these two methods, segmentation with and without depth data, is shown in Fig. 4. In this figure, we can see that without depth data, shadows of the stapler is segmented as part of the foreground, while a part of the tape are segmented as the background. In contrast, segmentation with depth data gives a better result, which can separate the object from the background compactly. Finally, we obtain a set of points P_O which construct the 2D shape of the object.

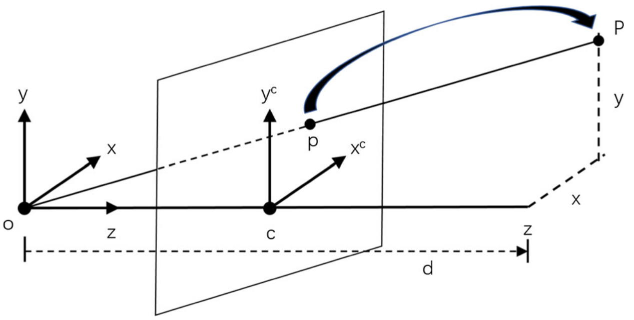


Fig. 3 Transformation between coordinate frames

4 Grasp Planning

4.1 Grasping an Ellipse and Its Rectangle Representation

Before detailed description of the proposed algorithm, we first discuss how to grasp an ellipse for better grasp stability by examining some examples.

Consider an ellipse given by the following equation, where $a > b$.

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \tag{4}$$

Consider grasping the ellipse with flat fingertips at different points shown in Fig. 5a–c. According to our daily experiences, the grasp in Fig. 5c seems more stable among these grasps. From this example, we can come to a preliminary conclusion that the curvature of the grasped object and distance between the two contact points are important in grasp stability.

In [26], the author makes a comparison between contact grasp stability [27] and spatial grasp stability [28]. They show that spatial stability cannot represent the essence of grasp stability and contact stability must be involved for a comprehensive evaluation of the grasp stability. According to their theory, the grasp in Fig. 5c is of good contact stability. Therefore, we consider it to be the best grasp for an ellipse with a parallel gripper.

For the convenience of the post-processing, we adopt the rectangle-based approach proposed in [1] to present the grasp, as shown in Fig. 5d. The center of the grasping rectangle and the center of the ellipse coincide. The long side of the rectangle is parallel to the short axis of the ellipse, while the short side of the rectangle is parallel to the long axis. The length of the long side of the rectangle is set a little greater to the length of the short axis to avoid collision.

4.2 Fuzzy Gaussian Mixture Models (GMMs)

Gaussian mixture models are widely employed in different pattern recognition problems acting as a powerful tool to classify or represent data, which we use in the proposed method. In a GMM, the probability density at the value of x is given by

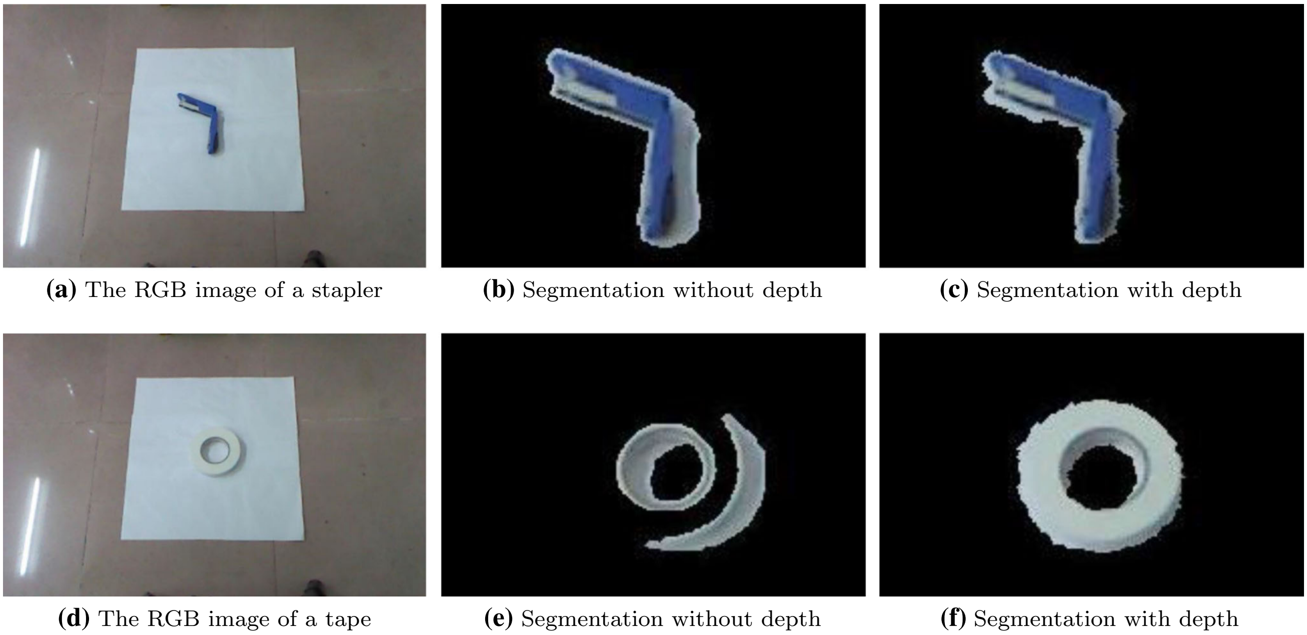

(a) The RGB image of a stapler
(b) Segmentation without depth
(c) Segmentation with depth
(d) The RGB image of a tape
(e) Segmentation without depth
(f) Segmentation with depth

Fig. 4 Segmentation results without and with depth

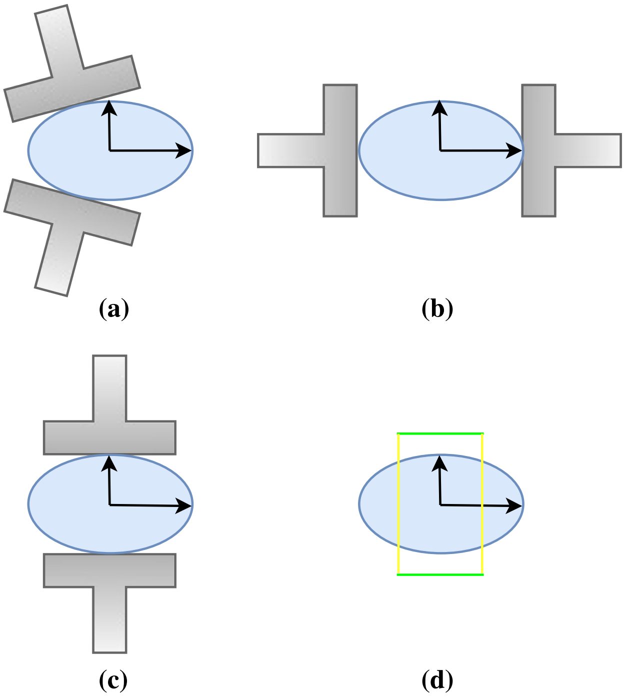


Fig. 5 Different grasps for an ellipse and the grasping rectangle

$$p(x|\Theta) = \sum_{i=1}^{K} w_i p_i(x|\mu_i, \Sigma_i) \quad (5)$$

$$p_i(x|\theta_i) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_i|}} \exp\left(-\frac{(x-\mu_i)^T \Sigma_i^{-1} (x-\mu_i)}{2}\right) \quad (6)$$

in which μ_i denotes the mean of the ith single Gaussian model, Σ_i is the covariance matrix and w_i is the mixture weight. Therefore, a GMM is determined by $\Theta = \{w_i, \mu_i, \Sigma_i | i = 1, \ldots, K\}$, where K denotes the number of components of the GMM.

Given a dataset of observations $X = \{x_1, x_2, \ldots, x_n\}$, our goal is to estimate Θ using maximum likelihood method. In other words, we need to find Θ that maximizes the log-likelihood function $L(X|\Theta)$.

$$L(X|\Theta) = \log\left(\prod_{t=1}^{n} p(x_t|\Theta)\right) \quad (7)$$

$$\hat{\Theta} = \arg\max_{\Theta} L(X|\Theta) \quad (8)$$

The common method for estimating parameters of GMM is to use expectation maximization (EM) algorithm [29], which is often utilized to estimate parameters with incomplete data. The following equations are applied to update the parameters of each component iteratively, where $p(i|x_t, \Theta)$ denotes the posteriori probability of data x_t belonging to the ith single Gaussian model.

$$\lambda_i^{\text{new}} = \sum_{t=1}^{n} p(i|x_t, \Theta) \quad (9)$$

$$w_i^{\text{new}} = \frac{\lambda_i^{\text{new}}}{n} \quad (10)$$

$$\mu_i^{\text{new}} = \frac{\sum_{t=1}^{n} p(i|x_t, \Theta) x_t}{\lambda_i^{\text{new}}} \quad (11)$$

$$\Sigma_i^{\text{new}} = \frac{\sum_{t=1}^{n} p(i|x_t, \Theta)(x_t - \mu_i^{\text{new}})(x_t - \mu_i^{\text{new}})^T}{\lambda_i^{\text{new}}} \tag{12}$$

Inspired by the mechanism of Fuzzy C-means, [11, 30] introduced the concept of fuzzy membership into the EM algorithm for GMM to accelerate the procedure of parameters estimation. They define a dissimilarity function d_{it} as follows,

$$d_{it}^2 = [w_i p_i(x_t|\theta_i)]^{-1} \tag{13}$$

Therefore, the degree of membership of x_t in the ith component u_{it} can be obtained according to Eq. 14, which is defined in fuzzy c-means clusteringl algorithm [31].

$$u_{it} = \left[\sum_{j=1}^{K}\left(\frac{d_{it}}{d_{jt}}\right)^{\frac{2}{m-1}}\right]^{-1} \tag{14}$$

Substitute Eq. 13 into Eq. 14, we can obtain

$$u_{it}^m = \frac{[w_i p_i(x_t|\theta_i)]^{\frac{m}{m-1}}}{\left[\sum_{j=1}^{K}(w_j p_j(x_t|\theta_j))^{\frac{1}{m-1}}\right]^m} \tag{15}$$

where m denotes the degree of fuzziness. And the equations for the update of parameters of GMM become

$$w_i^{\text{new}} = \frac{\sum_{t=1}^{n} u_{it}^m}{\sum_{i=1}^{K}\sum_{t=1}^{n} u_{it}^m} \tag{16}$$

$$\mu_i^{\text{new}} = \frac{\sum_{t=1}^{n} u_{it}^m x_t}{\sum_{t=1}^{n} u_{it}^m} \tag{17}$$

$$\Sigma_i^{\text{new}} = \frac{\sum_{t=1}^{n} u_{it}^m (x_t - \mu_i^{\text{new}})(x_t - \mu_i^{\text{new}})^T}{\sum_{t=1}^{n} u_{it}^m} \tag{18}$$

According to the experiment results in [11], when the number of components $K \geq 2$, fuzzy GMM can converge to similar results with fewer iterations and less computational time compared to conventional GMM, which reveals that the introduction of fuzziness does help the EM algorithm converge faster. For this reason, when $K \geq 2$ we adopt the fuzzy EM in our algorithm instead of the conventional one.

When $K = 1$, GMM degenerates to a single Gaussian model (SGM). In this case, we directly estimate the parameters using the following equations without iteration,

$$\mu = \frac{\sum_{t=1}^{n} x_t}{n} \tag{19}$$

$$\Sigma = \frac{\sum_{t=1}^{n}(x_t - \mu)(x_t - \mu)^T}{n} \tag{20}$$

which are derived directly from the maximum likelihood method.

4.3 Adaptive Fuzzy GMM for Shape Approximation

After filtering out the background points in the RGB image, the points belonging to the target object are left behind, which form the 2D shape of the object. Each of these points is represented by its location (x, y) in the image coordinate frame. We assume that these two-dimensional points are generated by some kind of probability distributions like Gaussian mixture model, without knowing its parameters. Therefore, we employ EM algorithm described in Sect. 4.2 to estimate the parameters of GMM using these points.

A Gaussian mixture model is composed of several single Gaussian models (SGM). Each SGM can be represented by an ellipse since the isoline of the probability density is also an ellipse which can be fully determined by the parameters of SGM. The center of the ellipse is determined by the mean μ_i. The axis are determined by the unit eigenvectors V_i of the covariance matrix Σ_i, where $[V_i, D_i] = eig(\Sigma_i)$. The short axis is determined by the eigenvector corresponding to the smallest eigenvalue. The length of these axis can be determined by the eigenvalues λs of the covariance matrix Σ_i.

Combining the analysis above and in Sect. 4.1, we can generate a grasping rectangle for each SGM. The center of the grasping rectangle is set equal to the mean μ of SGM so as to align their centers. The long side and short side are aligned to each unit eigenvector v of the covariance matrix Σ. The width and height of the grasping rectangle are determined by

$$\text{Width} = f \times 2 \times \sqrt{\lambda_2} \tag{21}$$

$$\text{Height} = \frac{1}{2} \times \text{width} \tag{22}$$

where λ_2 is the smallest eigenvalue which is the variance along the direction of the eigenvector. f is a scale factor to adjust the *width*, which is set to 2.5 for the best performance. For convenience, we simply set the *height* half of the *width*.

An example is shown in Fig. 6 for an intuitionistic explanation. In this example, we manually set the number of components $K = 1$ since the shape of the umbrella is not complex. In Fig. 6a, blue points are randomly sampled from P_O obtained according to Eq. 3 which filters out all of the background points and leaves behind points belonging to the target object. We estimate a probability density function for these points using GMM, and one of the isolines of the probability density is represented as an orange ellipse in the figure. Note that in this case, GMM degenerates to a single Gaussian model. The rectangle composed of green lines and yellow lines is the corresponding

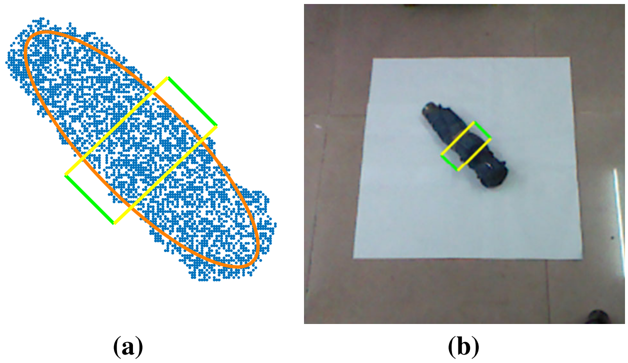


Fig. 6 GMM-based shape approximation for an umbrella

grasping rectangle of that ellipse. We can clearly see in Fig. 6b that it indicates a good grasp for the umbrella.

However, when the shape of the object is getting more complex, more components are required for a proper approximation. Sequentially, we come across some problems, such as how to choose the number of components adaptively and which rectangle to choose among all these rectangles. Because of the inaccurate calibration between the robot and the camera and noisy measurements from joint encoders, the error in gripper pose is inevitable. How to deal with uncertainty when executing the grasp becomes another problem.

To solve these problems, we introduce a grasp quality metric under uncertainty to evaluate each grasping rectangle. Given a pair of contact points of a grasp, if the forces it applies at these two points are opposite and collinear, it is called an antipodal grasp. It satisfies the force-closure condition and it is a theoretically stable grasp [32]. An antipodal grasp requires that the angle between the vector connecting two contact points and the normal of each contact point should be close to zero, which we formulate as follows.

$$s(p_1, p_2) = \cos(\alpha_{p_1}) * \cos(\alpha_{p_2}) \tag{23}$$

where p_1, p_2 denote the two contact points and α_{p_i} denotes the angle between the connecting vector and the normal of contact point p_i. The normal of contact point p_i can be easily estimated using the 3D point cloud. If a pair of contact points p_1 and p_2 forms an antipodal grasp, then $s(p_1, p_2)$ would be equal to 1. It can be inferred that contact points with bigger $s(p_1, p_2)$ would be more likely to form an antipodal grasp.

Due to the uncertainty in gripper pose, when the gripper closes, the actual contact points are likely not to lie along the short axis of the ellipse. Instead, they will randomly locate at a certain area. We divide the grasp rectangle into three equal parts as shown in Fig. 7 and denote them as S_1, S_2 and S_3. For the convenience of analysis, we assume that when the gripper closes, the actual contact points will locate at S_1 and S_3 randomly. For a reasonable evaluation of the grasp quality of the grasping rectangle, we randomly generate many pairs of contact points (by randomly selecting points located at S_1 and S_3, respectively), depicted as red points connected with black lines in Fig. 7, calculate the grasp quality score $s(p_{i1}, p_{i2})$ for each pair of contact points p_{i1}, p_{i2}, and average them all as the final grasp quality score, which is formulated as follows.

$$S(R) = \sum_{i=1}^{N} s(p_{i1}, p_{i2})/N \tag{24}$$

where R denotes a given rectangle and N denotes the number of pairs of contact points we generate. In theory, a larger N leads to a better estimation of the grasp quality of the grasping rectangle. For computational efficiency, we manually set N to 100. Note that we select these points in the 2D plane, but we must first transform them to their corresponding 3D points in the point cloud to calculate the grasp quality score $s(p_{i1}, p_{i2})$ since the contact points are actually 3D points. This metric is more robust because it takes an average of the grasp quality of many pairs of contact points instead of using only one pair of contact points.

Using this metric to evaluate each grasping rectangle, we propose the adaptive fuzzy GMM-based shape approximation algorithm for grasp planning. A grasping rectangle which can be considered to be a proper one should satisfy the following rules:

Firstly, the width of the rectangle should not exceed the actual opening width of the gripper.

Secondly, the gripper should avoid to crash into the object, which means that two short sides of the rectangle should not overlap the object in the image.

Meanwhile, the grasp quality score $S(R)$ of the grasping rectangle should satisfy the following formulation,

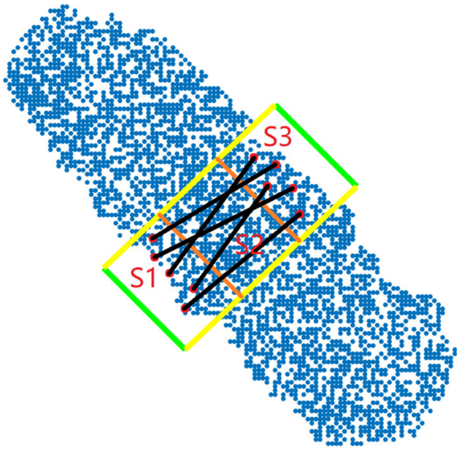


Fig. 7 Pairs of contact points

$$S(R) > \tau \qquad (25)$$

where τ is a predefined threshold. If this threshold is set too high, no proper grasp would be found. If this threshold is set too low, the quality of the grasp would be not guaranteed. According to our experiments, setting τ to 0.7 can lead to a satisfactory performance.

Testing all the grasping rectangles by the above rules, when there are several proper rectangles left, we consider the rectangle with the biggest $S(R)$ value as the best one. When none of the rectangles satisfies the conditions above, we set the number of components to $(K + 1)$ and repeat the fuzzy GMM algorithm again until the best rectangle is obtained. We give an example to better explain the proposed algorithm, which is illustrated in Fig. 8.

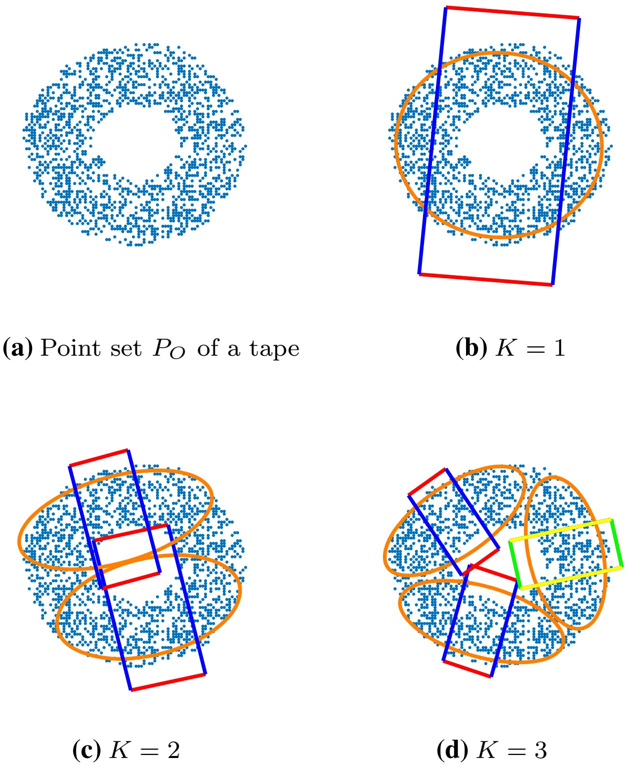
In Fig. 8a, blue points are randomly sampled from P_O in Fig. 4f, which construct the 2D shape of the tape. Figure 8b–d demonstrates the procedure of the proposed algorithm, in which final planned grasp is depicted as the rectangle with yellow and green lines, while other rectangles are the intermediate results. At the beginning, the number of components of GMM K is initially set to 1 and the obtained grasping rectangle exceeds maximum opening width of the gripper as can be seen in (b). Therefore, K is updated to 2. It can be seen in (c) that both of the obtained grasping rectangles will cause collision to the tape when executing the grasps. Sequentially, K is updated to 3 and all of the resulting grasping rectangles are viable. By evaluating grasp quality metric for each rectangle, the one with the maximal $S(R)$ value which satisfies Equation 25 is chosen as the best grasp. Actually, due to the symmetry of the tape, the calculated $S(R)$ values of these three rectangles are very close, which means that they have similar grasp quality.

5 Experiments

5.1 Off-line Experiments with Grasp Planning

The whole procedure of our algorithm is depicted as follows. Firstly, the RGB-D image of the object is captured by Kinect; secondly, the target object in the image will be segmented from the background with depth information as described in Sect. 3.2; and finally, the grasp planning algorithm described in Sect. 4.3 will be employed to generate the best grasp configuration.

In this experiment, we would like to visualize the result of grasp planning to analyze the performance of the proposed algorithm. We collected several common objects which vary in material, shape and size to test the algorithm off-line, and the result of our algorithm is given in Fig. 9a.


(a) Point set P_O of a tape

(b) $K = 1$

(c) $K = 2$

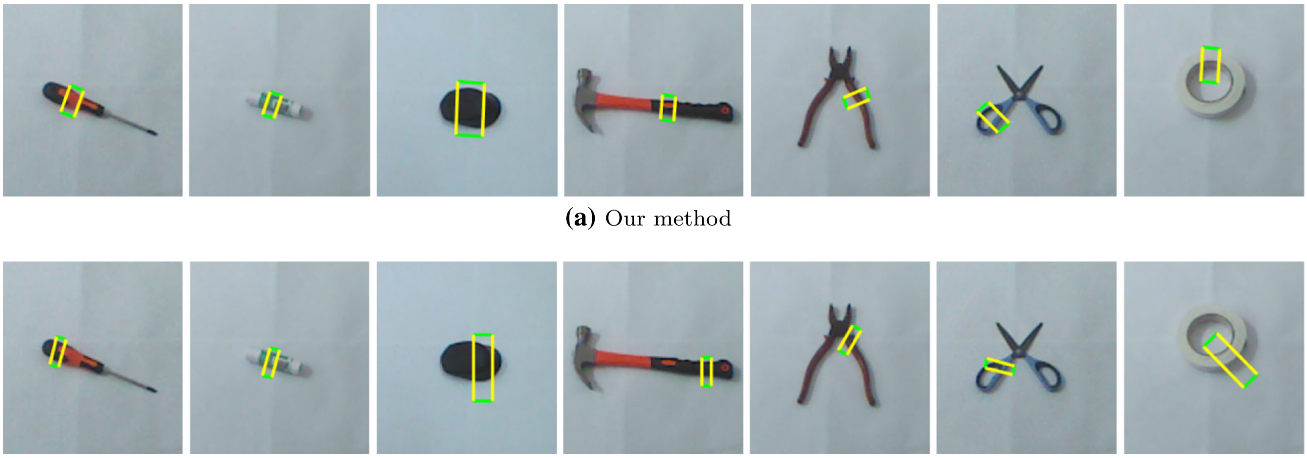
(d) $K = 3$

Fig. 8 Process of the grasp planning

In addition, we compare our algorithm with the approach in [6], which is the first to employ deep learning method in generating robotic grasps. The deep network they used is trained on the Cornell Grasp Dataset. Since the code for their paper is available on the Internet, we can easily make a comparison between their method and ours. The comparison results are shown in Fig. 9. We can see that most of the grasp planning results using Lenz's method indicate good grasps. And the performance of our method is comparable and some planned grasps seem to be more reasonable, such as the grasps for the scissors and the pliers. We can also see that the grasps generated by our method lie along the orientation of the objects. They are similar to the way in which humans grasp objects; therefore, they are more likely to succeed when executing these grasps.

We also make a comparison among the computational efficiency of Lenz's algorithm, our algorithm using normal EM and fuzzy EM. The CPU of the computer we use is Intel(R) Core(TM) i5-6400 CPU with basic frequency of 2.71 GHz and both of the algorithms run on MATLAB R2017a. We run the algorithm 10 times for each object and average the amount of time required to generate the grasp. Table 1 presents the comparison of these methods in terms of running time in seconds.

From Table 1, we can see that Lenz's method is very time-consuming, since it searches for the best grasp in an



(a) Our method

(b) Lenz's deep learning method[6]

Fig. 9 Comparison of grasp planning results using different methods

exhaustive way using sliding windows, which cannot meet the demand in real scenario. The execution time of these methods largely depends on the size of the objects, as we can see in Fig. 9 and Table 1. As for our method, since we directly estimate the parameters of GMM using maximum likelihood instead of EM when $K = 1$, the screwdriver, glue and mouse can be processed within less than 5 ms, respectively. For those objects with $K \geq 2$, we can observe that the algorithm using fuzzy EM consumes less time than using normal EM. From Table 1, we can obtain that fuzzy EM is 1.13× faster than normal EM on average. Besides, we also observed in our experiments that fuzzy EM can always converge using less iterations compared to the normal EM with regard to the same object. It can be concluded that the fuzzy EM algorithm does help to improve the computational efficiency in the aspects of running times and the number of iterations. Among the objects listed in the table which are very common in our daily life, our algorithm using fuzzy EM algorithm can plan the grasp for each object within 100 ms. In the field of robotic grasping, it is a relatively fast approach. The last column of the table K denotes the number of components required for shape approximation with our method for each object. Objects with complex shape need more components for shape approximation, such as the tape, which needs three components, while the glue only needs one. It can be observed that the execution time increases monotonically with K.

Table 1 Computational efficiency comparison

Object	Running times (s)			K
	Lenz et al. [6]	Ours		
		Normal EM	Fuzzy EM	
Screwdriver	21.42	0.0027		1
Glue	16.84	0.0028		1
Mouse	28.10	0.0033		1
Hammer	112.64	0.0132	**0.0114**	2
Pliers	72.92	0.0208	**0.0171**	2
Scissors	104.24	0.0341	**0.0307**	3
Tape	45.08	0.0560	**0.0496**	3

Bold value indicates the better performance of the proposed method

K is the number of components of GMM required in our method. Note that the K value of each object is not manually set, it is adaptively chosen by our algorithm

5.2 Real-World Grasping

To evaluate the performance of the proposed method in the real scenario, we conduct extensive experiments on a real robotic platform, a Baxter Research Robot. Baxter has two 7-DOF arms, both of which is equipped with a two-finger parallel gripper. Only the left arm is used for these experiments. There is a table in front of Baxter. A Kinect sensor is fixed near to the head of Baxter and angled downwards toward the table, which can capture the RGB-D image of the table.

In this experiment, the goal for Baxter was to grasp the target object using gripper and lift it. The whole process of the experiment is provided as follows. Firstly, we make the hand–eye calibration for transformation between Kinect's and Baxter's coordinate frames. Secondly, we take an RGB-D image of the table as a background image for segmentation, which contains no objects in the scene. Thirdly, we place a single object on the table where Baxter can reach. Then, we execute our algorithm with fuzzy GMM or Lenz's method to find the best grasping rectangle.

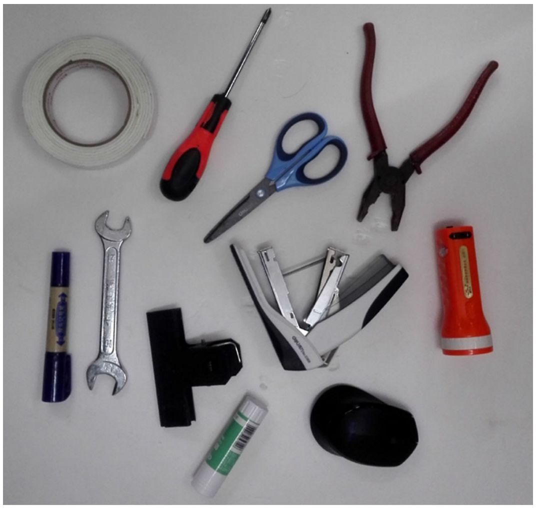


Fig. 10 Robotic experiment objects

Finally, we convert the obtained rectangle to the 7D gripper configuration and send a command to Baxter to execute the grasp.

In this experiments, we collected several objects from homes, our offices and laboratory, which are shown in Fig. 10. We executed the steps above in sequence to grasp these objects, each for 10 trials. The objects are placed on the table at different positions in different poses in different trials. If Baxter can grasp the object on the table, lift it up and keep it stable for 3 s, we consider it a successful grasp; otherwise, we record it as a failure. Some examples of successful grasps generated by our algorithm are given in Fig. 11.

The final experimental result is shown in Table 2. It reveals that our robotic grasping system is able to make successful grasps in 90.00% of cases, which demonstrates the good performance of our algorithm in grasping different objects. We can also observe that our algorithm can achieve comparable or even slightly better performance than Lenz's algorithm, the average success rate of which is 86.36%. From the experiment, we observed that most of the failure cases are because of the slight imprecision in calibration between the camera's and Baxter's coordinate frames and the inherent imprecision in Baxter's end-effector positioning, which may cause the gripper to crash into the object or grasp nothing. Though the obtained grasping rectangles generated by our algorithm have relatively high pose error robustness metric $S(R)$, the uncertainty in gripper pose is still disastrous. These pose error can be alleviated using force or tactile feedback, which is believed to complement our algorithm and improve the robustness of our grasping system.

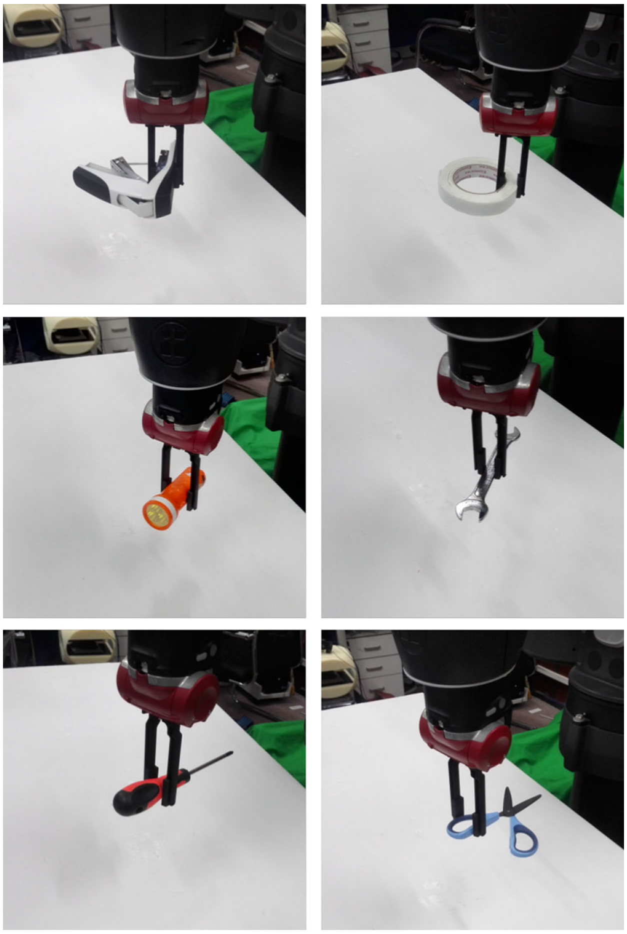


Fig. 11 Some screenshots of Baxter executing grasps

6 Conclusions

In this work, we consider the task of grasp planning for a parallel gripper to grasp a novel object, given an RGB image and its corresponding depth image taken from a single view. We propose a novel grasp planning algorithm based on shape approximation using Gaussian mixture models, which can adaptively decompose the target object into several ellipses and plan grasps upon these ellipses. We employ the fuzzy EM algorithm to improve the computational efficiency of the parameters estimation of GMM. Taking the pose uncertainty into consideration, we introduce a grasp quality metric to filter candidate grasps and obtain the most viable grasp. We also implement extensive grasping experiments on a real robotic platform. The results of both off-line and real robotic experiments demonstrate that our algorithm enables the robot to grasp a variety of novel objects with high success rate and high computational efficiency.

Table 2 Results on robotic experiments

Object	Accuracy(%)	
	Lenz et al. [6]	Ours
Tape	80	**90**
Screwdriver	90	90
Glue	100	100
Mouse	100	90
Pincher	80	**90**
Scissors	70	**80**
Stapler	70	**80**
Flashlight	90	**100**
Clip	80	80
Pen	100	100
Wrench	90	90
Average	86.36	**90.00**

Bold value indicates the better performance of the proposed method

However, there is still room for improvement. Our future work will focus on alleviating the uncertainty in gripper pose by leveraging the force, tactile or visual feedback in our grasping system to achieve a better performance.

Acknowledgements This work was partially supported in part by the Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/S001913/1, in part by the National Nature Science Foundation under Grants 61702195, 61751202, and U181320097.

References

1. Jiang, Y., Moseson, S., Saxena, A.: Efficient grasping from rgbd images: Learning using a new rectangle representation. In: 2011 IEEE International Conference on Robotics and Automation (ICRA), , IEEE, pp. 3304–3311 (2011)
2. Miller, A. T., Knoop, S., Christensen, H.I., Allen, P. K. : Automatic grasp planning using shape primitives. In: ICRA'03. IEEE International Conference on Robotics and Automation, 2003. Proceedings. Vol. 2, IEEE, pp. 1824–1829 (2003)
3. Miller, A.T., Allen, P.K.: Graspit! a versatile simulator for robotic grasping. IEEE Robot. Autom. Mag. **11**(4), 110–122 (2004)
4. Pelossof, R., Miller, A., Allen, P., Jebara, T.: An svm learning approach to robotic grasping. In: 2004 IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. Vol. 4, IEEE, pp. 3512–3518 (2004)
5. Le, Q. V.: Building high-level features using large scale unsupervised learning. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, pp. 8595–8598 (2013)
6. Lenz, I., Lee, H., Saxena, A.: Deep learning for detecting robotic grasps. Int. J. Robot. Res. **34**(4–5), 705–724 (2015)
7. Redmon, J., Angelova, A.: Real-time grasp detection using convolutional neural networks. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp. 1316–1322 (2015)
8. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
9. Kumra, S., Kanan, C.: Robotic grasp detection using deep convolutional neural networks. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp. 769–776 (2017)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
11. Ju, Z., Liu, H.: Fuzzy gaussian mixture models. Pattern Recognit. **45**, 1146–1158 (2012)
12. Johns, E., Leutenegger, S., Davison, A. J.: Deep learning a grasp function for grasping under gripper pose uncertainty. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2016, pp. 4461–4468 (2016)
13. Nguyen, V.-D.: Constructing stable force-closure grasps. In: Proceedings of 1986 ACM Fall joint Computer Conference, IEEE Computer Society Press, pp. 129–137 (1986)
14. Ponce, J., Stam, D., Faverjon, B.: On computing two-finger force-closure grasps of curved 2d objects. Int. J. Robot. Res. **12**(3), 263–273 (1993)
15. Dizioğlu, B., Lakshiminarayana, K.: Mechanics of form closure. Acta Mech. **52**(1–2), 107–118 (1984)
16. Ferrari, C., Canny, J.: Planning optimal grasps, in: IEEE International Conference on Robotics and Automation, 1992. Proceedings, 1992 , IEEE, pp. 2290–2295 (1992)
17. Dogar, M., Hsiao, K., Ciocarlie, M., Srinivasa, S.: Physics-based grasp planning through clutter. Robotics: Science and Systems (2012)
18. Goldfeder, C., Ciocarlie, M., Dang, H., Allen, P. K.: The columbia grasp database. In: IEEE International Conference on Robotics and Automation, 2009. ICRA'09. IEEE, pp. 1710–1716 (2009)
19. Weisz, J., Allen, P.K.: Pose error robust grasping from contact wrench space metrics. In: 2012 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp. 557–562 (2012)
20. Lai, K., Bo, L., Ren, X., Fox, D.: A large-scale hierarchical multiview rgb-d object dataset. In: 2011 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp. 1817–1824 (2011)
21. Blum, M., Springenberg, J.T., Wülfing, J., Riedmiller, M.: A learned feature descriptor for object recognition in rgb-d data. In: 2012 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp. 1298–1303 (2012)
22. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. **115**(3), 211–252 (2015)
23. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: Deepface: Closing the gap to human-level performance in face verification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1701–1708 (2014)
24. Wang, X., Yang, C., Ju, Z., Ma, H., Fu, M.: Robot manipulator self-identification for surrounding obstacle detection. Multimed. Tools Appl. **76**(5), 6495–6520 (2017)
25. Rao, D., Le, Q.V., Phoka, T., Quigley, M., Sudsang, A., Ng, A.Y.: Grasping novel objects with depth segmentation. In: 2010

IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp. 2578–2585 (2010)

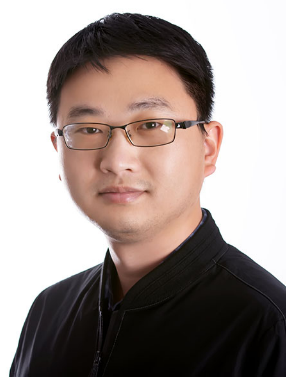
26. Montana, D.J.: The condition for contact grasp stability. In: 1991 IEEE International Conference on Robotics and Automation, 1991. Proceedings. IEEE, pp. 412–417 (1991)
27. Hanafusa, H.: Stable prehension by a robot hand with elastic fingers. In: Proceedings of the 7th International Symposium on Industrial Robots, Tokyo, 1977, pp. 361–368 (1977)
28. Salisbury, J.K., Roth, B.: Kinematic and force analysis of articulated mechanical hands. J. Mech. Transm. Autom. Des. **105**(1), 35–41 (1983)
29. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. J. R. Stat. Soc. Ser. B Methodol 1–38 (1977)
30. Peizhuang, W.: Pattern recognition with fuzzy objective function algorithms (james c. bezdek). SIAM Rev. **25**(3), 442 (1983)
31. Bezdek, J.C.: FCM: the fuzzy c-means clustering algorithm. Comput. Geosci. **10**, 191–203 (2002)
32. Murray, R.M.: A Mathematical Introduction to Robotic Manipulation. CRC Press, Boca Raton (2017)
33. Zhang, T., Chen, C.L., Chen, L., Xu, X., Hu, B.: Design of highly nonlinear substitution boxes based on I-Ching operators. IEEE Trans. Cybern. **48**, 3349–3358 (2018)
34. Chen, C.L., Zhang, T., Chen, L., Tam, S.C.: I-Ching divination evolutionary algorithm and its convergence analysis. IEEE Trans. Cybern. **47**, 2–13 (2017)

Huifeng Lin received the B.E. degree in automation from the South China University of Technology, Guangzhou, China, in 2018, and is currently pursuing the M.S. degree in the South China University of Technology, Guangzhou, China. His research interests include human–robot interaction, machine learning and image processing.

Tong Zhang received the B.S. degree in software engineering from Sun Yat-sen University, at Guangzhou, China, in 2009, and the M.S. degree in applied mathematics from University of Macau, at Macau, China, in 2011, and the Ph.D. degree in software engineering from the University of Macau, at Macau, China in 2016. Dr. Zhang currently is an assistant professor with School of Electronics and Information, South China University of Technology, at Guangzhou, China. His research interests include affective computing, evolutionary computation, neural network, and other machine learning techniques and their applications. Dr. Zhang has been working in publication matters for many IEEE conferences.

Zhaopeng Chen received B.S. degree in Mechanical Engineering and Automation and M.S. degree in Mechatronic Engineering from Harbin Institute of Technology, Harbin,P.R. China, in 2005 and 2007, respectively. From 2008 to 2010, he was a joint-train Ph.D. candidate in the Institute of Robotics and Mechatronics, DLR German Aerospace Center, Oberpfaffenhofen, Germany. Since 2010, he has been with DLR German Aerospace Center in Oberpfaffenhofen, Germany, as a research scientist. His current interests include dexterous robot hands and nonlinear control.

Haina Song is a Senior Engineer at Haihua Electronics Enterprise (China) Co.Ltd. She is also a Ph.D. candidate at South China University of Technology. Her research interests include computer vision and vehicle environmental perception in the field of Internet of Vehicles.



Chenguang Yang is a Professor of Robotics. He received the Ph.D. degree in control engineering from the National University of Singapore, Singapore, in 2010 and performed postdoctoral research in human robotics at Imperial College London, London, UK from 2009 to 2010. He has been awarded EU Marie Curie International Incoming Fellowship, UK EPSRC UKRI Innovation Fellowship, and the Best Paper Award of the IEEE Transactions on Robotics as well as over ten conference Best Paper Awards. His research interest lies in human robot interaction and intelligent system design.