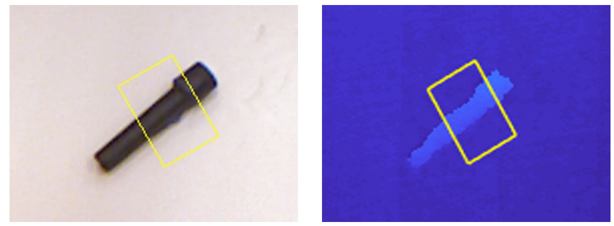
2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
September 24–28, 2017, Vancouver, BC, Canada

A Fast Search Algorithm Based on Image Pyramid for Robotic Grasping

Guangli Ren[1], Zhenzhou Shao[1*], Yong Guan[1], Ying Qu[2], Jindong Tan[2], Hongxing Wei[3] and Guofeng Tong[4]

***Abstract*— To improve the search efficiency of robotic grasping detection, this paper presents a novel search algorithm based on the image pyramid. It significantly reduces the search space for grasping position detection using the coarse-to-fine strategy. The proposed method searches the positions from the top layer of the pyramid, and initializes the search area at the next layer. The sparse automatic encoder is employed to construct the model which is used to evaluate the grasp quality. The experimental results demonstrate that the proposed search algorithm can improve efficiency of the robotic grasping detection with the comparative performance on the grasp quality.**



(a) RGB image (b) Depth image

Fig. 1: Grasping position detection based on RGB-D data. The yellow rectangle indicates the grasp position.

I. INTRODUCTION

Robotic grasping is of importance in the field of intelligent robot and is being applied to the home service, industrial production, space exploration, etc. [1]. The grasping procedure generally consists of object recognition, robotic grasping detection, generation of grasping configuration, motion planning and execution [2], [3], [4]. The detection of proper grasping position is the prerequisite for implementation of grasping task.

The grasping position detection is generally transferred into a search-evaluation problem based on machine vision [5], [6], [7], [8]. That is, given an image of target, the candidates of grasping positions are firstly obtained, and then we find the best one by means of the evaluation based on a score function. Along with the emergence of RGB-D sensors with the capability of real-time recording, the color and depth information is used in the grasping task [9], [10], [11]. Fig. 1 shows the RGB and depth images of a flashlight, and the region in the yellow rectangle represents the grasp position.

Unlike the traditional pattern recognition in machine vision, the robotic grasping detection has strict requirements with respect to the speed and precision of recognition. The most basic detection method is performed by traversing all possible candidate locations in the given target scene image and selecting the optimal one based on the evaluation matrix. However, it is a time-consuming task with the heavy overhead to find the best one from a large number of candidates. Recently, Jiang *et al.* [5] present a two-step learning algorithm to deal with the computationally expensive issue due to the quite large search space. However, this method only applies to linear evaluation problem using a linear score function. A two-stage closed-loop grasping candidate evaluator is proposed based on the reinforcement learning approach [7]. It improves the search efficiency using a feedback mechanism, in which the sliding window should be initialized by priori knowledge and spatial features. Similarly, Ian Lenz *et al.* [6] perform a two-stage cascaded detection method based on both deep learning networks. The one with less features is used to prune out the unlikely candidate grasps, and the other one to determine the best candidate with the highest score.

In this paper, we propose a grasping position search algorithm based on the image pyramid. The RGB-D images are processed by down-sampling with Gaussian filter to generate the Gaussian image pyramid. The detection of grasping position starts from the top of pyramid with the lowest resolution. In the case of inter-layer information transmission, the detection region of the high-resolution layer is initialized using the search result of the low-resolution layer. Finally, the optimal grasping position will be obtained at the bottom of pyramid with the highest resolution. The evaluation model is built using the deep neural network [12], [13]. Considering the complexity of features and huge amount of data used in the evaluation, we construct the deep network for the evaluation model based on the sparse

*Corresponding author

[1]Guangli Ren, Zhenzhou Shao and Yong Guan are with the College of Information Engineering, Beijing Advanced Innovation Center for Imaging Technology and Beijing Key Laboratory of Light Industrial Robot and Safety Verification, Capital Normal University, Beijing, 100048, China {2151002027,zshao,guanyong}@cnu.edu.cn

[2]Ying Qu and Jindong Tan are with the Engineering College, The University of Tennessee, Knoxville, TN, 37996, USA {yqu3,tan}@utk.edu

[3]Hongxing Wei is with the School of Mechanical Engineering and Automation, Beihang University, Beijing 100191, China weihongxing@buaa.edu.cn

[4]Guofeng Tong is with the College of Information Science and Engineering, Northeastern University, Shenyang, Liaoning, 110819, China tongguofeng@ise.neu.edu.cn

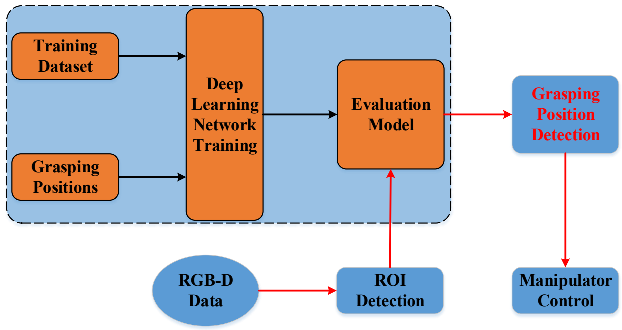


Fig. 2: Implementation of the grasping task.

automatic encoder (SAE) [14] which can extract the features automatically.

The rest of the paper is organized as follows. In Section II, we describe the problem of robotic grasping detection briefly. We present our search method based on image pyramid and the construction of evaluation model in Section III. Section IV demonstrates the experimental results. Conclusion is drawn in Section V.

II. PROBLEM DESCRIPTION

In this paper, we focus on the grasping process based on the RGB-D images captured by Kinect, as shown in Fig. 2. Through the pre-trained evaluation model, a large amount of potential grasping candidates are evaluated. The position in the top-ranked rectangle will be sent to the robotic arm, and then the grasping action is implemented by the manipulator control. In order to mathematically describe the problem of robotic grasping and explain the specific algorithmic steps in detail, the notations employed in this paper are tabulated in TABLE I.

In the process of grasping, the grasping position is represented by the rectangle $G^{(i)}$, which contains the position of the grasping rectangle. And the grasp space consists of all the grasping rectangles $G^{(i)}$ of the same object. Each grasping rectangle $G^{(i)}$ corresponds to a class $y^{(i)} \in \{0,1\}$, which indicates whether or not the rectangle satisfies the requirement of grasp. $\phi(G^{(i)})$ is used to represent the quality of the rectangle $G^{(i)}$.

To determine the optimal grasping rectangle G^*, a evaluation matrix in [6] is employed, as shown in Equation (1). In the target grasp space $GSpace$, the eigenvector $\psi(G^{(i)})$ of each rectangle $G^{(i)}$ is considered as the input data of the probability evaluation model P to calculate the evaluation score of the corresponding rectangle. The rectangle with the highest score is the optimal rectangle.

$$G^* = \underset{G^{(i)}}{\operatorname{argmax}}\ P(y^{(i)} = 1 | \psi(G^{(i)}); \Theta), \tag{1}$$

where Θ denotes the weight parameter of the probability evaluation model which is constructed by the deep learning method.

III. SEARCH ALGORITHM BASED ON IMAGE PYRAMID FOR OPTIMAL GRASPING POSITION

In this paper, the proposed search algorithm uses the coarse-to-fine strategy based on Gaussian image pyramid with the multi-resolution characteristic. It allows to deal with the target image at different resolutions. We use the evaluation model constructed by the deep neural network to evaluate a score for each candidate rectangle depending on the color, depth, and surface normal vectors that are learned from the candidate rectangle. The score is used as the quality standard for rectangle. A rectangle is graspable, if the score is greater than 5. In particular, the one greater than 10 results in a high-quality level. The evaluation model chosen in our work is a four-layer network, including two hidden layers, as shown in Fig. 3. The hidden layers are mainly used to extract the evaluation feature vectors for the evaluation model. The sparse auto-encoder is used to extract the features, which avoids the time-consuming effort on feature design. Although features can be learned automatically, the evaluation model still requires off-line training using a supervised approach.

A. Search Model Based on Gaussian Pyramid

The image pyramid [15] is an effective but simple concept structure that explains the image in multiple resolutions. The Gaussian pyramid and Laplacian pyramid are commonly

TABLE I: Notations

$GSpace$	Grasp space
$GSpace(L_j, \beta, G^*, k)$	Grasp space constrained by L_j, β , G^* and k
k	Step size of the sliding rectangle window
I	A rectangular patch image
L_j	The jth layer of Image pyramid
$G^{(i)}$	The ith Grasping rectangle
G^*	The optimal grasping rectangle
$\psi(G^{(i)})$	The feature of the ith rectangle
$y^{(i)}$	The class of the ith rectangle
$\phi(G^{(i)})$	The score of the ith rectangle
Θ	The weight parameters of model
$h^{(i)}$	The vector of the ith hidden layer
λ	Penalty coefficient of the sparse representation
α, β	Rotation angle
$g(z)$	$Sigmoid$ activate function
P	Probability evaluation model
$y^{(i)}$	The estimated value of class of the ith rectangle
$\Gamma(G)$	Mapping between different pyramid layers

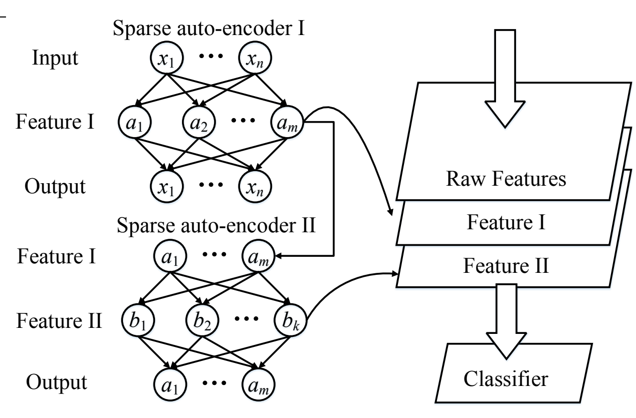


Fig. 3: Construction of the evaluation model.

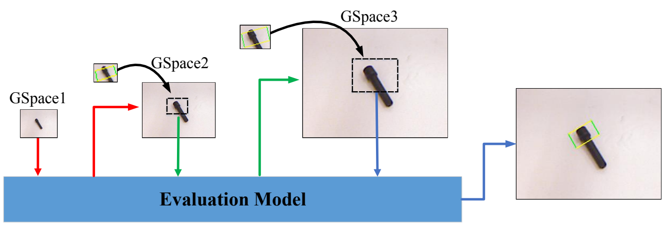


Fig. 4: Search model based on the three-level pyramid. The arrows in different colors represent the search process at different levels, the black dashed box indicates the detection area for initialization.

used. We applied the Gaussian pyramid in this paper. Assuming that the original target scene I_0 is the bottom of the Gaussian pyramid with the highest resolution (layer 0), then the lth layer of the Gaussian pyramid image I_l is:

$$I_l = \Sigma_{m=-2}^{2}\Sigma_{n=-2}^{2} w(m,n) I_{l-1}(2i+m, 2j+n), \quad (2)$$

where $w(m,n)$ is the product nucleus by the factor of 5×5, i and j are the position of pixel in the image.

According to the Equation (2), a Gaussian pyramid is constructed by down-sampling with Gaussian filter. The highest-resolution representation of the image is at the bottom of pyramid, and the top is the lowest-resolution image. The data transmission between layers is done using the mapping relationship $\Gamma(G)$, which is a coordinate transformation operation that returns the coordinates of the search area at the next layer, according to the coordinates of the optimal grasping rectangle obtained at the current layer. Fig. 4 shows the search algorithm model based on the three-layer pyramid. The search space $GSpace1$ is generated from the lowest-resolution image at the top of pyramid, and all the candidate rectangles in $GSpace1$ are evaluated by the evaluation model P, then the optimal rectangle of the top-layer of pyramid is obtained. A new grasp space $GSpace2$ is generated by applying the inter-layer mapping $\Gamma(G)$ to the current search area. It is used to initialize the search area at the middle layer, as shown in the black dashed box in Fig. 4. According to the same principle, the search area at the bottom layer is initialized, then the optimal rectangle can be obtained at the bottom layer of pyramid.

B. Feature Extraction Based on SAE

Sparse auto-encoder is a deep learning network with a single hidden layer, which has unsupervised feature learning capability. It introduces the L_1 regularization to the data based on the auto-encoder method. The features extracted from input data become sparse, which is helpful to improve the robustness of network. The structure of SAE is shown in top-left part of Fig. 3, the number of nodes in the input layer is the same as in the output layer, and the number of hidden layer nodes is generally less than input nodes. SAE can be divided into two steps, including the encoding and decoding. The first step is used to apply the constraints to the input data. Suppose that $x = \{x_1, x_2, \cdots, x_n\}$ is the input vector, the hidden layer vector $a = \{a_1, a_2, \cdots, a_m\}$ is obtained by Equation (3). The second step is the reconstruction of the input layer. We decode the vector a to obtain the reconstruction $x^{'}$ of the input vector x by Equation (4). While the training of SAE is the process of feature extraction.

$$a = Wx + b, \quad (3)$$

$$x^{'} = W^{'}a + b_1, \quad (4)$$

where W and $W^{'}$ denote the weight matrices and b and b_1 are bias matrices.

In general, the reconstruction error function is:

$$\min_{a, W^{'}, b_1} \|W^{'}a + b_1 - x\|^2 + \lambda \Sigma_j |a_j|, \quad (5)$$

where λ denotes the penalty coefficient of constraint. The more sparse of feature vectors, the more robust of them, which can improve the recognition accuracy and reduce the computational overload. The purpose of SAE training is to minimize the reconstruction error by adjusting the weights and biases between the layers. The sparse vector $a = \{a_1, a_2, \cdots, a_m\}$ is the extracted feature vector we want.

C. Construction of Evaluation Model

In this paper, a deep learning model with two hidden layers, $h^{(1)}$ and $h^{(2)}$, is chosen as the evaluation model of rectangles, as shown in Fig. 3. Firstly, the feature $Feature\ I$ of the hidden layer $h^{(1)}$ is obtained by abstracting the raw features using the first SAE, and the weight matrix $\Theta^{(1)}$ in the SAE network is taken as the initial weights of evaluation network $h^{(1)}$. Then the advanced feature $Feature\ II$ is obtained using the second SAE network fed by $Feature\ I$. And we consider the weight matrix $\Theta^{(2)}$ of the second SAE as the initial weights of $h^{(2)}$. The feature extraction is done with two abstraction.

Since the two hidden layers have only completed the feature extraction which cannot be used to classify and evaluate rectangles. We then employ a classification layer using the softmax regression [16] based on supervised learning. The output of network can be obtained according to the Equation (6), and use the gradient descent method for network adjustment, so as to complete the learning for the weights of classifier. At this point, the construction and training of the evaluation model are basically completed.

$$\begin{cases} h_j^1 = g(\Sigma_{i=1}^{N})\psi(G^{(i)})\Theta_{i,j}^{(1)} \\ h_j^2 = g(\Sigma_{i=1}^{K^1})h_j^1\Theta_{i,j}^{(2)} \\ P(y^{(i)} = 1|\psi(G^{(i)});\Theta) = g(\Sigma_{i=1}^{K^2} h_j^2 \Theta_{i,j}^{(3)}) \end{cases} \quad (6)$$

To evaluate the grasping position, we calculate the scores of candidate rectangles according to Equation (7), and finally find the rectangle with the highest score using Equation (8).

$$\phi(G^{(i)}) = \Sigma_{i=1}^{k} \Theta^T \psi(G^{(i)}) \quad (7)$$

Algorithm 1 Search Algorithm Based on Image Pyramid for Robotic Grasping.

Input: A rectangular patch image I, initialize the maximum layer of pyramids L_{max}, rotation angle of the sliding rectangle window α, step size of the sliding rectangle window k, score of the optimal rectangle $\phi^*(G)$.

for $j = 1 : L_{max}$ **do**
  for $\beta = 0 : \alpha : 180°$ **do**
    Initialize the current grasp space $GSpace(L_j, \beta, G^*, k)$.
    for t = 1:sizeof($GSpace(L_j, \beta, G^*, k)$) **do**
      Score calculation: $\phi(G) = \Theta^T \psi(G^{(i)})$
      if $\phi(G) > \phi^*(G)$ **then**
        $\phi^*(G) \leftarrow \phi(G)$
        $G^* \leftarrow G$
      end if
    end for
  end for
  Initialize the next layer of pyramid:
  $G^* = \Gamma(G^*)$
end for

Output: G^* (Optimal grasping rectangle).

$$G^* = \underset{G^{(i)}}{\mathrm{argmax}}\ \phi(G^{(i)}) \tag{8}$$

where K^i is the number of nodes in the ith hidden layer, and in order to prevent data divergence during transmission, the output data is bounded by the sigmoid activation function, by which ensure that the results correspond to the category of grasping rectangles.

The search algorithm based on image pyramid is summarized in Algorithm 1.

IV. EXPERIMENTS AND RESULTS

In order to verify the performance of proposed algorithm, two set of experiments are conducted in this section. In the first experiment, different search methods are implemented followed by the same evaluation model from the off-line training to compare proposed method with the algorithms of global search and the two-stage cascaded search in [6]. And we analyzed the running time and evaluation scores of the three methods. In the second experiment, we explore the impact on running time and recognition accuracy using the image pyramids with different layers.

These experiments use the dataset from Cornell grasping dataset [17], which contains 885 images of objects, each of which is labeled with 1-4 graspable rectangles and ungraspable rectangles. The dataset contains the RGB image and point cloud data of the object. Fig. 5 shows the part of data used in this experiment with the different size, shape and direction of the objects, including flashlight, cosmetic and shovel.

The seven channels of features are chosen in the experiments, including YUV, surface normal vector and depth features. These feature vectors are extracted from the rectangles, and represented by seven vectors with the size of 24×24, so the number of input features per network is 4032. The evaluation model used in the experiment is a double hidden layer deep network obtained by off-line training. The configuration used in the experiments is shown in TABLE II.

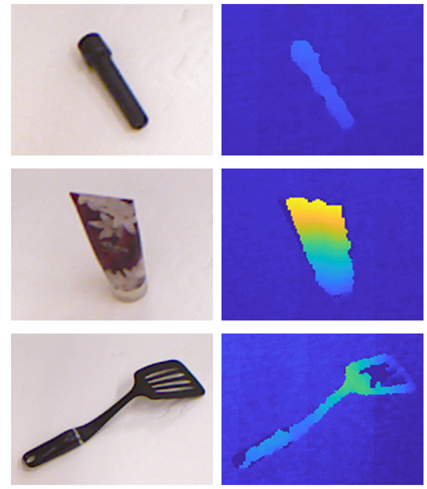


Fig. 5: Partial data in Cornell grasping dataset.

TABLE II: Configuration used in the experiments.

Category	Specification
The Operating System	Windows 7
CPU	AMD A8-3520M APU
RAM	6 GB
Basic Frequency	1.6 GHz
Software Environment	MATLAB R2014a

A. Comparison of Different Grasp Search Methods

In this experiment, the same evaluation model is used to evaluate the optimal grasping positions of different objects using the global search method, the two-stage cascaded search method and proposed method. The process of the global search method is similar to the first layer of the two-stage cascaded search method, but the evaluation function used is the evaluation function mentioned above. The experimental targets include flashlight, comb, plastic bottle, shovel, scissor and cosmetic.

The experimental results are shown in Fig. 6. We can see that the three search methods are basically the same for the grasping results of flashlight, comb and plastic bottle. However, for the remaining three objects, there are differences in the optimal position of the three methods. However, from the point of evaluation score to view, the selected rectangles are all high quality.

TABLE III shows that the running time using the proposed method is less than one third of the time of other ones. Especially in the grasping detection for scissor, it is ten times faster than other methods. Compared to the global search method, the search time of two-stage cascaded detection is increased by a few seconds. It is mainly caused by the fact that the evaluation procedure is implemented again at

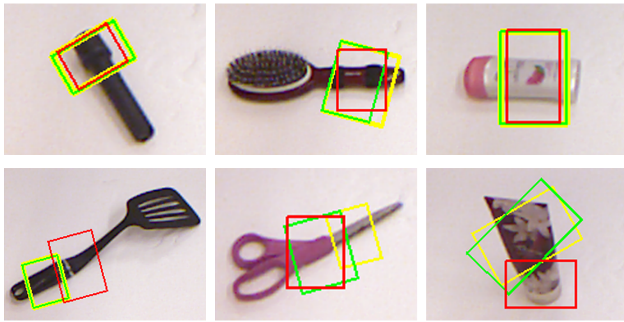


Fig. 6: Grasp search results using different methods. The yellow rectangle represents the result of the global search algorithm, the green rectangle represents the result of the two-stage cascaded detection and the red rectangle represents the result of the three-layer pyramid search algorithm.

TABLE III: Running time comparison of different search methods (s).

Method / Object	Global search	Two-stage cascaded search	Pyramid based search
Flashlight	381.62	399.22	123.13
Comb	499.01	552.04	180.14
Plastic bottle	229.73	261.47	59.73
Shovel	2185.08	2265.97	487.15
Scissor	714.05	717.66	58.40
Cosmetic	523.71	514.31	59.92

the second stage. As shown in TABLE IV, in terms of evaluation scores, the evaluation score using the pyramid search method is about one evaluation unit smaller than other two methods, but the evaluation scores of the three methods are all greater than 10, which means the selected rectangles are all the positions with high quality. This suggests that the pyramid search method not only improves the search efficiency by more than three times compared with the global search method and the two-stage cascaded search method, but also can obtain a high-quality grasping position.

From the statistical histogram shown in Fig. 7, the quality scores of image pyramid based search method are comparable, and the running time is much less than others.

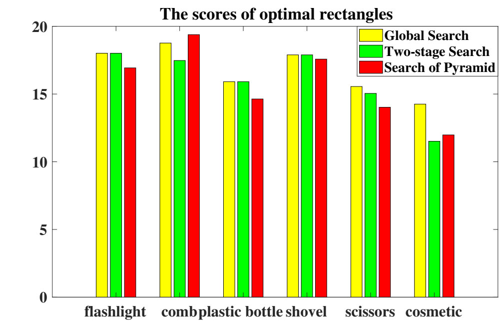
B. Comparison of Pyramid Methods with Different Layers

In this part, we compare the performance of methods based on two-layer and three-layer pyramid on the same targets used in the first experiment. The running time and quality of rectangles are also compared, the results are shown in Fig. 8 and TABLE V. We can see that the search methods based on different levels of pyramids are almost the same for the rectangles of plastic bottle and the scissor. For the remaining four targets, the rectangles are different.

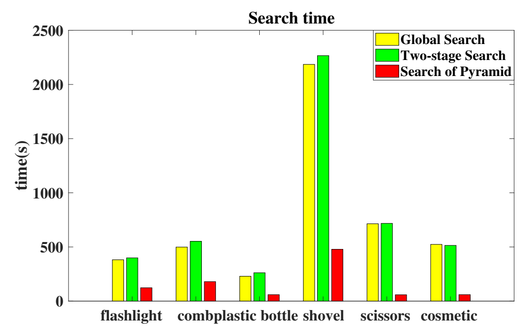
Fig. 9 intuitively shows that the evaluation scores obtained through search methods based on different levels of pyramids are all greater than 10. It indicates that both search methods can yield the high-quality positions. The evaluation scores of the three-layer pyramid are generally less than the scores of two-layer search, except for the plastic bottle. However, the search time of three-layer pyramid is shorter than the

TABLE IV: Comparison of Grasping quality using different search methods.

Method / Object	Global search	Two-stage cascaded search	Pyramid based search
Flashlight	18.0128	18.0128	16.9321
Comb	18.7650	17.4759	19.3859
Plastic bottle	15.9125	15.9125	14.6344
Shovel	17.8924	17.8924	17.5781
Scissor	15.5550	15.0469	14.0250
Cosmetic	14.2562	11.5092	11.9808



(a) Grasping quality



(b) Running time

Fig. 7: Comparison on the grasping quality and running time using the different methods.

two-layer search algorithm. This shows that increasing the number of layers of pyramid is able to improve performance on search time, but the quality of rectangle will drop a little.

V. CONCLUSION

In this paper, we present a novel robotic grasping search algorithm based on image pyramid. In this method, we deal with the large amount of candidate grasps at different layers of the image pyramid. Through the forward initialization mechanism, the existing results are used to initialize the search area of the next layer of image pyramid. In this way, a large number of ungraspable positions are filtered out, which makes the computational complexity reduce significantly. We link the method and a evaluation model of grasps trained off-line together to achieve a fast and accurate search for optimal grasping position. The experiments and results show that both the efficiency and accuracy are performed well compared with the methods of global search and two-stage cascaded search. Due to limitation of hardware condition, the algorithm can not be used in the real scene to carry out

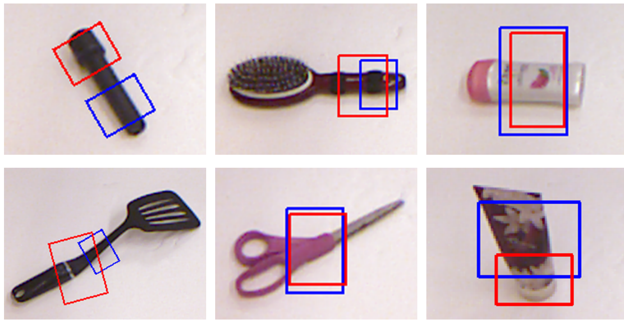


Fig. 8: Grasp search results based on different image pyramids. The red rectangle represents the result of the three-layer pyramid search algorithm, and the blue rectangle represents the result of a two-layer pyramid search algorithm.

TABLE V: Search results based on different layers of image pyramids.

	Running time(s)		Grasping quality	
Object	Two-layer	Three-layer	Two-layer	Three-layer
Flashlight	201.00	123.13	18.4046	16.9321
Comb	224.21	180.14	18.2347	19.3859
Plastic bottle	87.82	59.73	15.1408	14.6344
Shovel	520.37	478.15	17.6995	17.5781
Scissor	107.50	58.40	14.9500	14.0250
Cosmetic	239.81	59.92	13.9191	11.9808

the experiments. In the future, we will consider to combine the algorithm and robotic arm control system to complete the task of robotic grasp.

ACKNOWLEDGMENT

This work was supported by the Project of Beijing Municipal Commission of Education (KM201710028017), the National Natural Science Foundation of China (61572331,61472468,61602325, 61373034), the National Key Technology Research and Development Program (2015BAF13B01), the International Cooperation Program on Science and Technology (2011DFG13000), the Project of Beijing Municipal Science & Technology Commission(Z141100002014001), the Project of Construction of Innovative Teams and Teacher Career Development for Universities and Colleges Under Beijing Municipality(IDHT20150507), and Training young backbone talents personal projects (2014000020124G135).

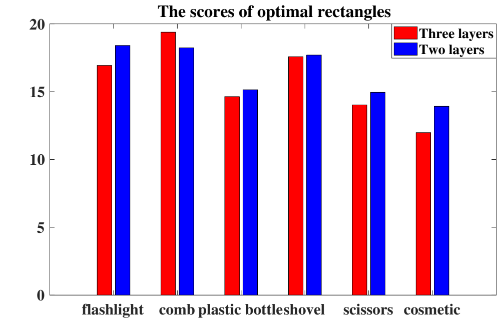
REFERENCES

[1] J. Yu, K. Weng, G. Liang, and G. Xie, "A vision-based robotic grasping system using deep learning for 3d object recognition and pose estimation," in *Robotics and Biomimetics (ROBIO), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1175–1180.

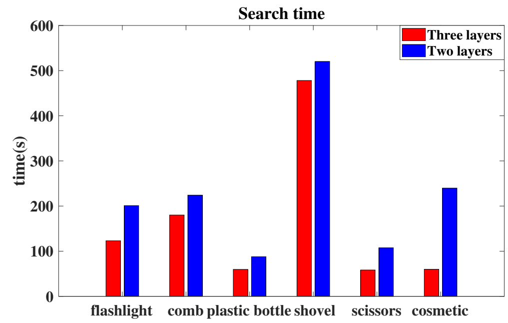
[2] R. Paolini, A. Rodriguez, S. S. Srinivasa, and M. T. Mason, *A Data-Driven Statistical Framework for Post-Grasp Manipulation*. Springer International Publishing, 2013.

[3] M. Ciocarlie, K. Hsiao, E. G. Jones, S. Chitta, R. B. Rusu, and I. A. ucan, "Towards reliable grasping and manipulation in household environments," *Springer Tracts in Advanced Robotics*, vol. 79, 2014.

[4] H. Jin, Q. Chen, Z. Chen, Y. Hu, and J. Zhang, "Multi-leapmotion sensor based demonstration for robotic refine tabletop object manipulation task," *Caai Transactions on Intelligence Technology*, vol. 1, no. 1, pp. 104–113, 2016.



(a) Grasping quality



(b) Running time

Fig. 9: Comparison on the grasping quality and running time using the pyramid methods with different layers.

[5] Y. Jiang, S. Moseson, and A. Saxena, "Efficient grasping from rgbd images: Learning using a new rectangle representation," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 3304–3311.

[6] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 705–724, 2015.

[7] Z. Wang, Z. Li, B. Wang, and H. Liu, "Robot grasp detection using multimodal deep convolutional neural networks," *Advances in Mechanical Engineering*, vol. 8, no. 9, 2016.

[8] C. M. Zhang LE and H. K, "Grasp evaluation with graspable feature matching," *RSS Workshop on Mobile Manipulation: Learning to Manipulate*, 2011.

[9] E. Johns, S. Leutenegger, and A. J. Davison, "Deep learning a grasp function for grasping under gripper pose uncertainty," 2016.

[10] Q. V. Le, D. Kamm, A. F. Kara, and A. Y. Ng, "Learning to grasp objects with multiple contact points," in *IEEE International Conference on Robotics and Automation*, 2010, pp. 5062–5069.

[11] J. Redmon and A. Angelova, "Real-time grasp detection using convolutional neural networks," vol. 2015, pp. 1316–1322, 2015.

[12] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K. R. Muller, "Evaluating the visualization of what a deep neural network has learned." *IEEE Transactions on Neural Networks and Learning Systems*, 2016.

[13] J. Schmidhuber, "Deep learning in neural networks: an overview." *Neural Networks*, vol. 61, pp. 85–117, 2014.

[14] H. L. Rong and Y. X. Xia, "A vehicle type recognition method based on sparse auto encoder," 2015.

[15] B. K. Choudhary, N. K. Sinha, and P. Shanker, "Pyramid method in image processing," *Journal of Information Systems and Communication*, vol. 3, no. 1, p. 269, 2012.

[16] P. A. Gutierrez, C. Hervas-Martinez, and F. J. Martinez-Estudillo, "Logistic regression by means of evolutionary radial basis function neural networks." *Neural Networks IEEE Transactions on*, vol. 22, no. 2, pp. 246–263, 2011.

[17] "Cornell grasping dataset," http://pr.cs.cornell.edu/grasping/rect_data/data.php.