Hindawi
Complexity
Volume 2021, Article ID 7619794, 12 pages
https://doi.org/10.1155/2021/7619794

Research Article

Grasp Detection under Occlusions Using SIFT Features

Zhaojun Ye,[1,2] Yi Guo,[1,2] Chengguang Wang,[3] Haohui Huang,[1,2] and Genke Yang[1,2,3]

[1]*Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China*
[2]*Ningbo Artificial Intelligence Institute of Shanghai Jiao Tong University, Ningbo, Zhejiang 315000, China*
[3]*Ningbo Industrial Internet Institute, Ningbo, Zhejiang 315000, China*

Correspondence should be addressed to Yi Guo; guo.yi@sjtu.edu.cn and Genke Yang; gkyang@sjtu.edu.cn

Received 30 August 2021; Accepted 22 October 2021; Published 13 November 2021

Academic Editor: Zhenyu Lu

Distinguishing target object under occlusions has become the forefront of research to cope with grasping study in general. In this paper, a novel framework which is able to be utilized for a parallel robotic gripper is proposed. There are two key steps for the proposed method in the process of grasping occluded object: generating template information and grasp detection using the matching algorithm. A neural network, trained by the RGB-D data from the Cornell Grasp Dataset, predicts multiple grasp rectangles on template images. A proposed matching algorithm is utilized to eliminate the influence caused by occluded parts on scene images and generates multiple grasp rectangles for objects under occlusions using the grasp information of matched template images. In order to improve the quality of matching result, the proposed matching algorithm improves the SIFT algorithm and combines it with the improved RANSAC algorithm. In this way, this paper obtains suitable grasp rectangles on scene images and offers a new thought about grasping detection under occlusions. The validation results show the effectiveness and efficiency of this approach.

1. Introduction

Robotic grasping has been a hot-spot topic and drawn increasing attention from researchers. With the growing demand of intelligent robot, robotic grasping technique has been extensively adopted in our daily life, such as workshop assembly, service robotic grasping, and agricultural robotic grasping [1–5]. However, compared with human beings, robots still have great limitations in grasping, such as grasping in occlusion case. Meanwhile, manipulating objects in occlusion occasion is an inevitable application for robots, like grasping under household [5] and industrial [6] scenes. Consequently, improving the ability of grasping objects in the case of occlusion and overlap is a difficult but necessary work for robotic manipulation.

Many works focus on predicting grasp rectangles on single-object scene [1, 7–9]. Nevertheless, robots usually face the scenes of grasping target from multiple objects. This type of problem is also called "bin picking." Some works [10–13] offer solutions to such problem, and they settle multiple objects grasp detection to a certain extent. However, they pay no attention on overlapped situations and may predict grasp rectangles in the overlapping areas. It can cause collision between objects and the robotic manipulator. Other works [14, 15] consider the occlusions of objects and work out such problem in their own way but cause high cost of dataset acquisition.

The proposal method in this paper divides such question into two main stages and predicts suitable grasp configurations (grasp configuration will be shown in Section 3) using an RGB input image. Inspired by [1], the first stage predicts multiple grasp rectangles on template images using the neural network ResNet-50. Each template image is taken in advance and contains only one object. Template images and the corresponding grasp rectangles are integrated into the template information; the second stage utilizes the proposed matching algorithm to connect the scene information with the template information and decreases the influence caused by occluded parts on scene images. Then, several grasp rectangles are predicted using the template information and connection between the template information and the scene information. A matrix M is used to

represent scaling, translation, and rotation information of a same object between two images. The proposal matching algorithm consists of the scale-invariant feature transform (SIFT) algorithm and random sample consensus (RANSAC) algorithm and connects two images based on SIFT features. In order to obtain a better connection, this paper improved the SIFT algorithm and RANSAC algorithm, respectively, and received a more accurate transformation matrix M between the template image and the scene image. The proposed grasp detection algorithm predicts multiple grasp rectangles with corresponding quality scores for objects under occlusions.

The main contributions of this paper are as follows:

(1) A novel grasp detection algorithm is proposed to predict grasp configurations for objects under occlusions. This algorithm, which is composed of a grasp rectangle predicting neural network and our key-point matching algorithm, predicts multiple grasp configurations on template images and generates several grasp configurations for objects on scene images using the connection between scene information and template information, rather than using an end-to-end network.

(2) The paper proposes a new way to predict grasp rectangles under occlusions by matching template images with scene images using our matching algorithm. The matching algorithm combines the SIFT algorithm and RANSAC algorithm and improves the two algorithms in order to receive a better matching result.

The rest of the paper is organised as follows. Related work about grasp detection is presented in Section 2. Problem formulation and a brief introduction to the SIFT algorithm are provided in Section 3. The detail of the proposed algorithm is discussed in Section 4. Section 5 provides the description of our experiment setup and validation results, respectively. The conclusion part can be found in Section 6.

2. Related Work

2.1. Grasp Detection Using Neural Network. In previous works like [16–18], model-based methods have played a primary position in solving the grasp detection problem. Such method uses the complete 3D model of the target object to define the grasp operation. However, robots face different environments, and obtaining the accurate 3D model in advance seems to be impossible [19]. On the contrary, it is more convenient to capture RGB images than reconstructed 3D models. Meanwhile, the learning-based method offers a strong generalization ability over object classification, target detection, and regression [10, 20]. Many works about object grasp detection have been done using the learning-based method. Early research studies focus on solving grasp detection problem in single-object scenes. These works pay no attention on multiobject, occluded cases and instruct the robot to grasp objects only under ideal circumstances [19]. Lenz et al. [21] connected two single neural networks in series in order to detect grasp positions in an RGB-D image. Similarly, Guo et al. [22] considered grasp detection using multiple sensors and proposed a hybrid deep architecture fusing the visual and the tactile information for grasp detection. The author collected a THU grasp dataset with visual, tactile, and grasp configuration information for network training. Reference [1] proposed a deep learning architecture to predict graspable locations using an RGB-D image for robotic manipulation. Different from some previous works, the paper considered to define the angle learning problem as classification with null hypothesis competition rather than regression and detected multiple grasp candidates for each object in a single shot with the input of RGB-D image.

The above works are all single-object scenes with objects in Cornell Grasp Dataset. However, grasping in multiobject scenes is inevitable in reality. Guo et al. [23] proposed a convolutional neural network to detect the target object and its optimal grasp configuration simultaneously on a fruit dataset. However, this model can just predict the grasp rectangle information of the most exposed object, without considering about partially visible objects. Vohra et al. [24] proposed a real-time grasp pose detection strategy for novel objects in our daily life. The proposed technique predicts the contour of the object in the point cloud and detects the grasp configurations along with the object skeleton in the image plane. Reference [25] explained a robotic grasp detection algorithm named ROI-GD to detect objects and their possible grasp configurations at the same time based on region of interest (ROI). The experiment results showed that this algorithm solved grasp detection problem of the object in the case of contact to some extent but did not offer the results of occluded cases.

Employed the neural network in [1], this paper trains a model using the Cornell Grasp Dataset and succeeds to obtain multiple grasp rectangles on the template image containing only one object. Then, these template images and relevant grasp configurations are used to generate grasp configurations for the target objects under occlusions during experiments. In order to build the connection between scene images and template images, this paper adopts a matching algorithm based on SIFT features to realize image matching task.

2.2. SIFT-Based Matching Algorithm. SIFT is a feature point extraction and matching algorithm proposed by Lowe in 1999 and perfectly improved in 2004 [26–28]. SIFT is proposed to extract distinctive invariant features from images in order to perform reliable match between different views of an object or a scene [27]. SIFT features are invariant to rotation and scale and can match robustly across affine distortion, change in 3D viewpoint, disturbance of noise, illumination variation, and even partial occlusion [29]. However, the original SIFT algorithm matches the key points by comparing the distance of the closest neighbor and the second-closets neighbor, and that method makes the detecting quality of the SIFT algorithm sensitive to the threshold. Duo to the requirements of different works,

researchers improved the performance of the SIFT algorithm in many ways. Dellinger et al. [30] proposed a new algorithm named SAR-SIFT based on the SIFT algorithm to reduce the influence caused by speckle noise on synthetic aperture radar (SAR). In [31], Alhwarin et al. improved the original SIFT algorithm for the purpose of providing a more reliable object recognition. Before matching the features, they divided the features of both test and template images into several subcollections according to the different octaves. Compared with the original SIFT algorithm, the processing time of the improved SIFT algorithm reduced 40% for matching the stereo images. Reference [32] utilized the improved RANSAC algorithm to realize a better SIFT feature point matching result and received an obvious promotion. The paper eliminated the mismatches using the improved RANSAC algorithm and obtained a more accuracy connection between images, also improved efficiency of processing. Other works like [33, 34] also proposed algorithms based on the SIFT algorithm to realize better results.

3. Preliminaries

3.1. Problem Formulation. Given an RGB scene image containing several arbitrarily placed objects, the objective of this paper is to identify the suitable grasp configurations for the target object even the target object is occluded by other objects. Inspired by [21], a grasp configuration of the target object can be represented using a six-dimensional vector:

$$g = (x, y, w, h, \theta, s), \tag{1}$$

where the grasp configuration g describes the grasp location, orientation information, and approximate opening distance of a parallel plate gripper. As shown in Figure 1(a), point G is the gripper's location and also the center of grasp rectangle, (x, y) is the coordinate of point G, angle θ is the orientation information, it represents the angle of rotation of the gripper in a certain direction, w and h represent the width and opening size of parallel plate gripper, respectively, and s represents the grasp quality score of the grasp configuration and is used to be the criterion of selecting the best grasp configuration. For each object, there may have several possible grasp configurations, and a set of proper grasp configurations $S(g)$ is obtained; each element of $S(g)$ represents a proper grasp configuration information for the target object in the scene:

$$S(g) = \{g_1, g_2, \ldots, g_n\}, \tag{2}$$

where n represents the number of grasp configurations we predict for the target object. We choose the best one for robotic manipulation depending on the quality scores of these grasp configurations.

Note that, this paper only predicts such grasp configurations for template images; each template image contains only one object we want to grasp and then generating grasp configurations for scene images using our matching algorithm.

3.2. Introduction of SIFT Algorithm. SIFT features are invariant to rotation and scale and can match robustly across affine distortion, change in 3D viewpoint, disturbance of noise, illumination variation, and even partial occlusion [29]. The original SIFT algorithm includes four main parts. The detail of the SIFT algorithm can be obtained from [27].

3.3. Scale Space Extrema Detection. The SIFT algorithm selects the extreme points of scale space as candidate feature points. The scale space of an image $I(x, y)$ is defined as follows:

$$L(x, y, \delta) = G(x, y, \delta) * I(x, y), \tag{3}$$

where $L(x, y, \delta)$ defines the convolution of original image $I(x, y)$ and a Gaussian function. $*$ presents two-dimensional convolution, and δ is the standard deviation of normal Gaussian distribution.

The SIFT algorithm uses scale space difference-of-Gaussian (DoG) function to generate a large number of extremas. The DoG image $D(x, y, \delta)$ is defined as follows:

$$D(x, y, \delta) = L(x, y, k\delta) - L(x, y, \delta), \tag{4}$$

where k is a constant over all scales; thus, it does not influence extrema location [27].

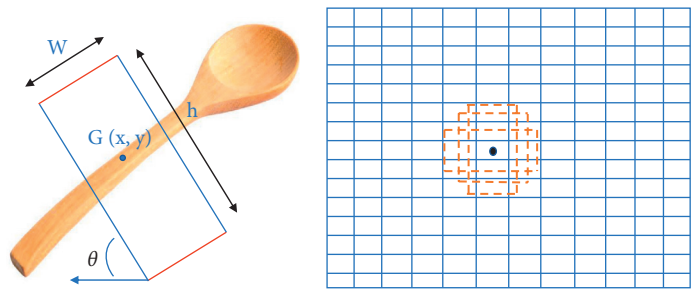
3.4. Key-Point Localization. This step is aimed to filter the key points in order to only retain the stable key points. The Taylor expansion of DoG function is constructed in scale space:

$$\begin{aligned} D(X) &= D + \frac{\partial D^T}{\partial X} X + \frac{1}{2} X^T \frac{\partial^2 D}{\partial X^2} X, \\ X &= (x, y, \delta)^T, \end{aligned} \tag{5}$$

and then the stable key point is obtained by solving the formula as follows:

$$\begin{aligned} \widehat{X} &= -\frac{\partial^2 D^{-}1}{\partial^2 X} \frac{\partial D}{\partial X}, \\ D(\widehat{X}) &= D + \frac{1}{2} \frac{\partial D}{\partial X} \widehat{X}. \end{aligned} \tag{6}$$

3.5. Orientation Assignment. In this step, every key point is assigned an orientation to make the descriptor invariant to rotation. Every direction contains gradient magnitude $m(x, y)$ and gradient direction $\theta(x, y)$ as follows:



(a) (b)

Figure 1: (a) Grasp configuration for parallel robotic gripper. Each grasp configuration contains the position information, orientation information, and grasp quality score. (b) Each anchor is corresponded to 9 possible grasp boxes with 3 scales and 3 aspect ratios (here only shows 3 of the possible grasp boxes of a same scale).

$$m(x,y) = \left[(L(x,y+1) - L(x,y-1))^2 + (L(x+1,y) - L(x-1,y))^2 \right]^{1/2},$$
$$\theta(x,y) = \arctan \frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}. \tag{7}$$

3.6. Key Point Descriptor. The last step divides the image region around the critical point to blocks; for each block, the gradient histogram of it is calculated, and then a 128-dimensional key point descriptor is generated.

In order to decrease the disturbance caused by occluded parts of scene images, this paper improves the original SIFT algorithm and receives fewer mismatches. Several matched SIFT features are extracted by the improved SIFT algorithm and fed into the remaining matching algorithm to build a more robust connection between the scene image and the corresponding template image.

4. Approach

The proposed algorithm can be divided into two stages (Figure 2): template generation and image matching. Firstly, before predicting grasp configurations during experiments with robotic gripper, several template images are taken in advance. Each template image contains only one object and performed grasp detection using neural network proposed in [1]. Then, a matching algorithm matches the target object in a scene image with template images and generates several grasp configurations for the target object under occlusions using the connection between the scene image and the template images.

The next three subsections describe the whole system in detail. It includes the architecture of the network and generates template information using the network, a description of obtaining connection between template images and scene images using the proposed matching algorithm, and a strategy of generating grasp configurations on scene images using the information of template images and connection.

4.1. Template Generation Using Neural Network. Currently, convolutional neural networks (CNNs) receive great performances on classification, detection, and regression problems. We use modified ResNet-50 with 50 layers to solve grasp detection problem. ResNet overcomes the challenge of learning mapping function by its residual learning concept. Every residual block is designed to be an incorporation which is a skip connection with the standard CNN. Meanwhile, ResNet can avoid time-consuming sliding-window approach shown in [8, 22] by utilizing the capacity of neural networks to execute bounding box regression and predict candidate regions on the whole image directly. The structure of our network is shown in Figure 3; we adopt the architecture proposed in [1] and train a model on Cornell Grasp Dataset.

The network takes RG-D images as input. The RG-D image is composed of the RGB image and corresponding depth image. Thus, the original dataset contains RGB images and depth images. Then, the data preprocessing part combines RGB images with the corresponding depth images to obtain RG-D images and crops them. After that, every cropped RG-D image is sent to intermediate convolutional layers (1–40 layer of ResNet-50). The intermediate convolutional layer extracts a common feature map with the size of $14 \times 14 \times 1024$. The feature map with r anchors is then sent to section Grasp Proposal Network. The Grasp Proposal Network slides a mini-network of 3×3 over the feature map and generates 9 possible grasp boxes with 3 aspect ratios and 3 scales for each anchors (Figure 1(b)). Thus, there are $r \times 9$ possible grasp boxes for each feature map. The Grasp Proposal Network outputs a $1 \times 1 \times 512$ feature map and then sends it into two sibling fully connected layers. Afterwards, the outputs of two layers represent the probability of grasp proposal and bounding box for each of r anchors on the feature map. The feature of each proposal bounding boxes is extracted by the ROI layer and sent to the remaining layers of the ResNet-50. Let t_i be the i-th grasp configuration with the form of (x, y, w, h) and p_i be the probability of the corresponding grasp proposal. *In* is an index set of all proposals; we use the formulation as follows to define the loss of grasp proposal net (gp).

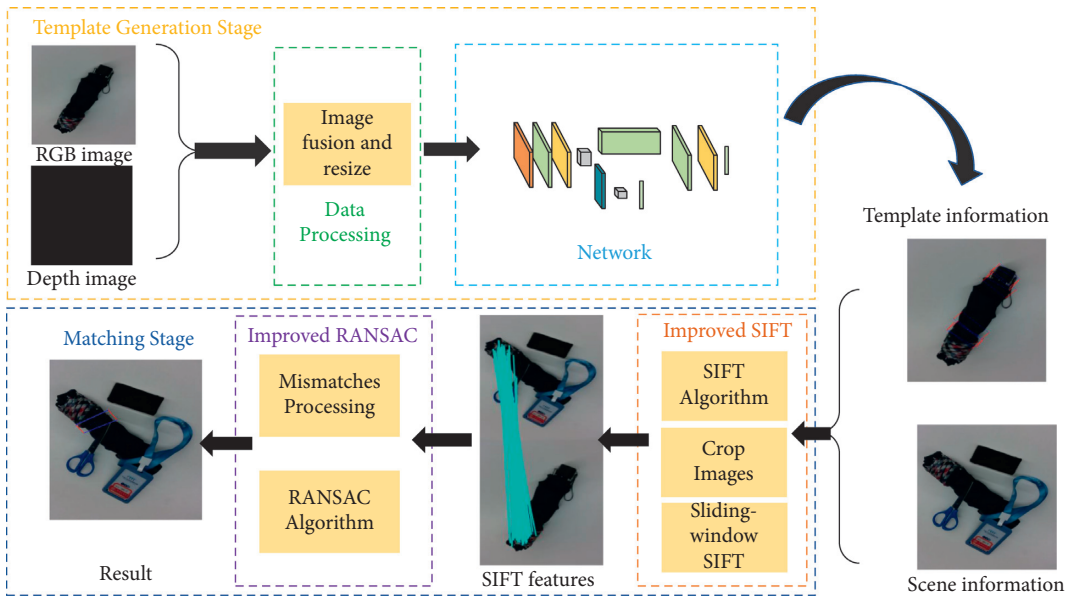


Figure 2: System diagram of our grasp detection algorithm. The template generation stage uses the network and template images to generate template information. For each scene image, the algorithm connects the scene image with template information in matching stage using a matching algorithm and obtains grasp rectangles for the target object in occluded condition.

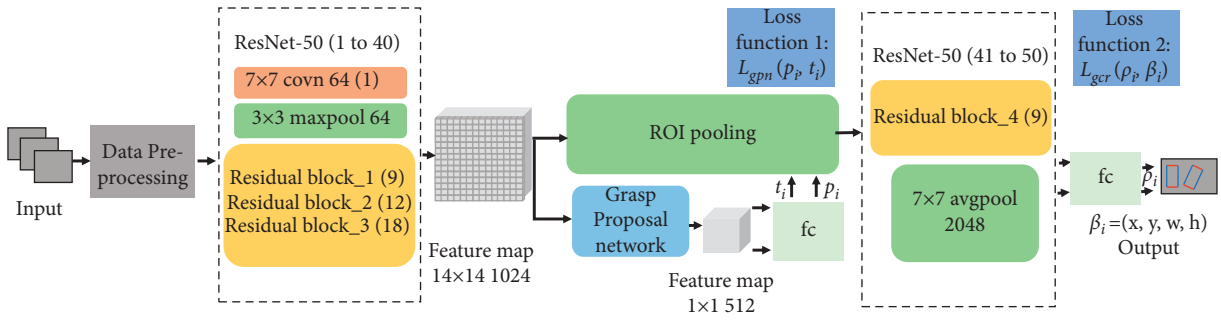


Figure 3: The structure of our network. The network uses RG-D images as inputs and predicts multiple grasp rectangles which contain position information, orientation information, and grasp quality score for each object. The data preprocessing part fuses the RGB image and the corresponding depth image to be the RG-D image and crops the RG-D image into a given size. The 1–40 layers of our network extract a feature map, the feature map is then fed into the Grasp Proposal Network and the ROI pooling. The rest of the network (41–50 layers) receives the output of the Grasp Proposal Network and generates several grasp rectangles.

$$L_{\mathrm{gp}}\left((p_i, c_i)_{i=1}^{In}\right) = \sum_i L_{\mathrm{gp\_}c}(p_i, p_i^*) + \lambda \sum_i p_i^* L_{\mathrm{gp\_}r}(c_i, c_i^*), \tag{8}$$

where $L_{\mathrm{gp\_}c}$ denotes the cross entropy loss of grasp proposal classification and $L_{\mathrm{gp\_}r}$ denotes the l_1 regression loss of grasp proposal with weight λ. p_i^* is an index with only two values. $p_i^* = 0$ represents no grasp, and $p_i^* = 1$ shows that a grasp is specified. c_i^* is the coordinate of ground-truth grasp corresponding to p_i^*.

As for θ, the orientation of each grasping configuration, we consider it as a classification task. The multigrasp detection pipeline quantizes θ into R equal-length intervals and generates multiple possible grasp configuration for each possible grasp proposal using θ. If none score of the possible grasp configuration is higher than the threshold we set, then we abandon the corresponding possible grasp proposal. In this paper, the total classes $C = R + 1$ and $R = 19$.

After the above processing, the last stage of the network classifies the grasp proposals into R regions and refines every proposal grasp bounding box to a bounding box (x, y, w, h) without orientation. A ROI pooling layer is added into ResNet-50 and shares the common feature map extracted by intermediate convolutional layer. Thus, it reduces the recomputation of feature extraction. All the features of the proposal grasps are stacked by ROI pooling layer and then fed into two sibling fully connected layers for the classification of orientation parameter l and regression of bounding box (x, y, w, h). The loss function of predicted grasp configuration (gc) is defined to be

$$L_{\mathrm{gc}}\left((\rho_l, \beta_l)_{c=0}^{C}\right) = \sum_c L_{\mathrm{gc\_}c}(\rho_l) + \lambda_2 \sum_c L_{\mathrm{gc\_}r}(\beta_c, \beta_c^*), \tag{9}$$

where ρ_i is the probability of class l, β_l is the corresponding grasp bounding box prediction, L_{gc_c} denotes the cross entropy loss of the angle classification, L_{gc_r} denotes the l_1 regression loss grasp bounding boxes with weight λ_2, and β_c^* is the ground-truth grasp bounding box. The total loss is defined to be

$$L_{total} = L_{gp} + L_{gc}. \tag{10}$$

The ROI layer generates grasp proposals, and grasp bounding boxes and orientations are received using the additional neurons of two sibling layers. Finally, we obtain the grasp configurations $S(g)$ of the target object on the template image. As mentioned above,

$$\begin{aligned} S(g) &= \{g_1, g_2, \ldots, g_n\}, \\ g &= (x, y, w, h, \theta, s). \end{aligned} \tag{11}$$

4.2. Connect Images Using SIFT Features. Using the proposed deep network, multiple grasp bounding boxes are obtained on the template images. Every template image contains a single object, and the predicted grasp bounding boxes show the suitable grasp configurations of the object. However, as it is described earlier, grasp detection may obtain some unsuitable grasp configurations in the occluded parts. In order to abandon the unsuitable grasp bounding boxes from all predicted grasp rectangles, we try to cut the occluded parts of scene image using our matching algorithm rather than predicting grasp rectangles on scene image using an end-to-end network.

SIFT features are invariant to rotation and scale and can match robustly across affine distortion, change in 3D viewpoint, disturbance of noise, illumination variation, and even partial occlusion [29]. The original SIFT algorithm matches the key points by comparing the distance of the closest neighbor and the second-closets neighbor, and that method makes the detecting quality of the SIFT algorithm sensitive to the threshold. Correct matching happens when the ratio is less than the threshold. Thus, as the threshold raises, the matching points increase, but mismatch increases too. Usually we want to generate more correct feature points and decrease the mismatches. Only by raising the threshold cannot solve such conflicting problem. Inspired by [32], we propose a matching algorithm which combines the improved SIFT algorithm with the improved RANSAC algorithm to improve the quality of matching. Our matching strategy is shown in Figure 4.

Firstly, we extract images SIFT feature points using the improved SIFT algorithm; this algorithm contains three parts: SIFT algorithm, crop images, and sliding-window SIFT. We utilize the SIFT algorithm to each pair of images and obtain many SIFT feature points of the two images. In order to decrease the influence of occluded parts, this paper uses the original matched SIFT features to detect the occluded parts. Our method divides the scene image into some patches and judges whether a patch is the occluded part by connecting it with template image using the SIFT algorithm. Note that the original matched SIFT features have proved the approximate position of the object on images; thus, unnecessary computation can be avoided by cropping the images. The crop image part crops images based on original matched SIFT features and obtains cropped parts of the images that contain the target object. Then, the sliding-window SIFT part slides the cropped scene image into several patches and judges whether a patch is the occluded part by the number of matched SIFT features. If a patch contains SIFT features, which are matched with template image, more than a given count (in this paper is 2), it is considered to be a part of the target object; otherwise, we classify it as occluded part and delete the corresponding SIFT features. After the improved SIFT algorithm, many matched SIFT features are obtained. The SIFT algorithm matches two images' SIFT feature points based on calculating the Euclidean distance between the two 128-dimensional key point descriptor. Such matching strategy makes the matching quality sensitive to the given threshold of the SIFT algorithm and cannot solve the trade-off between obtaining more matches and decreasing mismatches while it has to calculate the transformation matrix between two images, and mismatches may influence the accuracy of the result. In order to get a better result, we decide to eliminate some of the mismatches in advance and utilize the RANSAC algorithm to calculate a more accurate result afterwards.

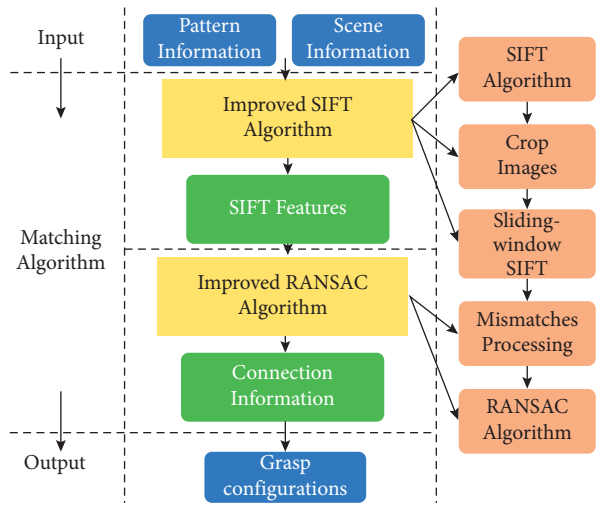


Figure 4: The diagram of our matching algorithm. The matching algorithm combines the improved SIFT algorithm and the improved RANSAC algorithm to connect the template information and the scene information. The connection information generated by our matching algorithm contains the transformation relationship between the matched pair of images. The output of the matching algorithm includes multiply grasp configurations of the target object in the occluded condition.

Inspired by [32], this paper considers the cross points as mismatches. Generally speaking, the size of same object is a constant. The transformation of the same object in two images can be considered as rotation and scaling. Thus, there should be no crossover between two correctly matched images. So, the proposed algorithm can eliminate part of the mismatches by abandoning the feature points which cause cross line with other lines. The main function of mismatches processing part is to delete such mismatches. After the

processing of the SIFT feature points, the RANSAC algorithm is used to calculate the transformation matrix M between each pair of images using matching feature point set F. F is defined as follows:

$$\begin{aligned} F &= f_1, f_2, \ldots, f_N, \\ f_k &= (t_k, s_k), \end{aligned} \tag{12}$$

where N is the number of matching point pair, f_k is the k-th matching point pair in F, k is an integer between 1 and N, and t_k and s_k are two corresponding feature points of template image and scene image, respectively.

Each pair of matching images can be connected using the corresponding transformation matrix M; the relationship between two matching points' coordinates and transformation matrix M is defined as follows:

$$\begin{aligned} P_s &= MP_t, \\ M &= \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}, \end{aligned} \tag{13}$$

where $P_s = \begin{bmatrix} x' & y' & 1 \end{bmatrix}^T$ is the coordinate of any feature point on scene image and $P_t = \begin{bmatrix} x & y & 1 \end{bmatrix}^T$ is the coordinate of the corresponding feature point on template image. By equation (13), we can connect two matched images at the pixel level and match each point on the template image to the pixel of the scene image. The improved RANSAC Algorithm 1 is described as follows.

4.3. Grasp Configurations Generating Strategy under Occlusions. Through the above steps, there obtains the grasp configurations $S(g)$ of the target object in the template image and uses the matching feature point pair set F to calculate the transformation matrix M between template image and scene image. Note that, the occluded parts of scene image cannot match with template image due to the specialty of the SIFT algorithm. At this step, this paper utilizes above information to generate grasp configurations for target object in the case of occlusion.

For every grasp configuration $g = (x, y, w, h, \theta, s)$ of $S(g)$, we denote $G = (x, y)$ as the grasp center point of g. Thus, we get a set $S(G)$ of grasp center point as follows:

$$S(G) = \{G_1, G_2, \ldots, G_n\}. \tag{14}$$

Meanwhile, we denote $S(t)$ and $S(s)$ as the matching feature point sets of template image and scene image, respectively:

$$\begin{aligned} S(t) &= \{t_1, t_2, \ldots, t_m\}, \\ S(s) &= \{s_1, s_2, \ldots, s_m\}, \end{aligned} \tag{15}$$

where n is the number of grasp center points of the target object in template image and m is the number of matching pairs which satisfy the transformation matrix M.

The proposed strategy is using the points in $S(G)$ to replace the points in $S(t)$ based on the Euclidean distance of pixels. Thus, for each $g = (G, w, h, \theta, s)$, we replace the grasp center point G with the closets SIFT feature point p'; the closets Euclidean distance is less than a given threshold, and a new set of grasp configurations $S(g')$ is created:

$$\begin{aligned} S(g') &= \{g_1', g_2', \ldots, g_r'\}, \\ g' &= (p', w, h, \theta, s), \end{aligned} \tag{16}$$

and then for each p', we have a corresponding SIFT feature point s' scene image, and the parameters of s' are as follows:

$$s' = (p', w, h, \theta + \beta, s), \tag{17}$$

where β is the orientation transformation of the transformation matrix M. Finally, we obtain the grasp configurations $S(s') = \{s_1', s_2', \ldots, s_r'\}$ of the target object in the case of occlusion.

5. Experiment

5.1. Dataset and Implementation for Network Training. In this paper, the goal is to predict grasp rectangles in occluded scenes. However, the proposed algorithm utilizes the connection between template information and scene information to obtain the grasp configurations in the multiobject, occluded cases, rather than an end-to-end deep neural network to predict grasp configurations on the scene images directly. Hence, the function of our network is to generate multiple grasp rectangles on the template image which contains only one target object. Note that, in order to increase the accuracy of detection, this paper stipulates the template image with only one object intentionally, because different matched objects between template image and scene image correspond to different transformation matrices in theory. On the basis of above condition, this paper chooses Cornell Grasp Dataset (Figure 5) as the dataset of our network.

The Cornell Grasp Dataset contains 855 images (RGB images and depth images) of 240 different objects [35]. Each image contains several ground-truth grasp rectangles with different orientations and positions. This paper takes the same procedure of data processing in [1, 8] and replaces the blue channel of each image with the corresponding depth channel. Because the data of blue channel are distributed between 0 and 255, we normalize the depth data to the same range. By combining RGB information and depth information, RG-D images are obtained. In order to generate plenty of training data and fit to the input size of ResNet-50, each image is performed extensive data augmentation by rotating randomly between 0 and 360 and resized to 227×227.

The network is implemented on Tensorflow framework and trained end-to-end on a single GPU of GTX1660Ti. We set the initial learning rate of our network to 0.0001 and divide it by 5 every 10000 iterations. We set the training epochs as 5.

Input: Input parameters set of SIFT matches F
Output: Output transformation matrix M
(1) Preprocessing: eliminate part of the mismatches by deleting the feature point pair that causes cross line. Denote the matching point set after preprocessing as F'.
(2) Select 4 pairs of points from F' randomly, and calculate transformation matrix M', create a new point set O.
(3) Judge whether other matching pairs satisfy the transformation of M' (a matching pair belongs to O if the error less than a given threshold), record the number of satisfied matching pair (elements in O) as m.
(4) If current m is the biggest than before, retain the current M'; otherwise, abandon it.
(5) Repeat 2, 3, and 4 a given times (5 in this paper), and obtain a transformation matrix M.

Algorithm 1: Improved RANSAC algorithm.

5.2. Evaluation Metrics of Predicted Result. In this paper, we take the metrics proposed in [21] to evaluate the grasp detection ability of our network. A grasp is considered to be a good grasp if it meets the following two criterions:

(1) The difference of grasp orientation between predicted grasp rotation angle and the corresponding grasp rotation angle of ground-truth is less than 30°.
(2) The Jaccard index of the predicted grasp configuration g' and the ground-truth g is more than 25%. The Jaccard index is defined as follows:

$$J(g, g') = \frac{g_A \cap g_A'}{g_A \cup g_A'}, \tag{18}$$

where g_A and g_A' are the areas of predicted grasp rectangle and ground-truth grasp rectangle, respectively. The Jaccard index is the ratio of intersection of the two rectangles to the union of the two rectangles.

5.3. Validation Results on Cornell Grasp Dataset and Household Objects. Three main types of experiments are performed to verify the ability of the proposed grasp detection algorithm. The first experiment makes a comparison with other typical methods on Cornell Grasp Dataset. In order to prove the validity of the proposed matching algorithm, the second experiment predicts grasp rectangles for objects under occlusions using the original SIFT algorithm and the proposed matching algorithm, respectively. The last experiment predicts grasp rectangles for some household objects under occlusions. The results of all experiments prove that the proposed grasp detection algorithm can ensure the detection accuracy on Cornell Grasp Dataset and solve the problem of grasp detection under occlusions to some extent.

Experiment 1 tests the proposed grasp detection algorithm on the Cornell Grasp Dataset and makes a comparison with prior works, and the result of comparison is shown in Table 1. We compare these works in two ways: image-wise split (*IW*) and object-wise split (*OW*):

(i) *IW*. The dataset is divided based on image randomly. Each image has an equal probability to be trained or tested. This is a common way to test the generalization of the network to new orientation and position about objects it has seen before.
(ii) *OW*. The dataset is divided based on object instances. Objects in training set and test set can be different. OW is used to test the generalization ability of a network about new object.

The performance of the proposed grasp detection algorithm on Cornell Grasp Dataset is shown at the last column of Table 1. For this grasp detection test on Cornell Grasp Dataset, we choose the best grasp rectangle from all the grasp candidates based on the corresponding output scores. Our grasp detection algorithm receives the accuracy of 97.2% on *IW* and 92.5% on *OW*, respectively. The detection accuracy is slightly inferior to the algorithm proposed in [1], the possible reason is that our matching algorithm may ignore some suitable grasp rectangles by mistake, we use the same network to generate several template images, but during calculating the transformation matrix, our matching algorithm may cut the right patch which contains the best grasp configuration, and the final result is the best of the remaining. Figure 6 shows the results of grasp detection on part of images of Cornell Grasp Dataset. We only show the grasp rectangle with the highest score.

Experiment 2 focuses on some household objects like umbrella, scissor, remote control, and so on. In order to validate the usefulness of our matching algorithm, we firstly compare our matching algorithm with the original SIFT algorithm. The result of comparison can be seen in Figure 7.

Figure 7(a) is the result of the original SIFT algorithm; the final prediction of grasp rectangle is at an occluded part. The original SIFT algorithm fails to decrease the influence of occluded parts and generates wrong grasp configurations. The reason may be that the position of grasp configuration with highest score in the template image is one of the occluded parts in the scene image; thus, without eliminating the influence of occluded parts, some SIFT feature pairs choose such grasp rectangle as the closest grasp position, and the result of matching goes wrong. Our matching algorithm deletes the occluded parts by combining the improved SIFT algorithm with the improved RANSAC algorithm and matches SIFT feature pair with the closest grasp position that is not in the occluded parts. The result is shown in Figure 7(b), and the robot can grasp the umbrella without grabbing other objects.

Figure 5: Several objects of Cornell Grasp Dataset.

Table 1: Performance of different methods on Cornell Grasp Dataset.

Approach	Algorithm	Accuracy (%)	
		IW	OW
Jiang et al. [36]	Fast search	60.5	58.3
Lenz et al. [21]	SAE, struct.	73.9	75.6
Redmon and Angelova [8]	AlexNet, MultiGrasp	88.0	87.1
Guo et al. [22]	ZF-net, hybrid network	93.2	89.1
Chu et al. [1]	ResNet-50 FCGN	97.7	94.9
Li et al. [19]	Key point-based scheme	96.05	96.5
This paper	The proposed algorithm	**97.2**	**92.5**

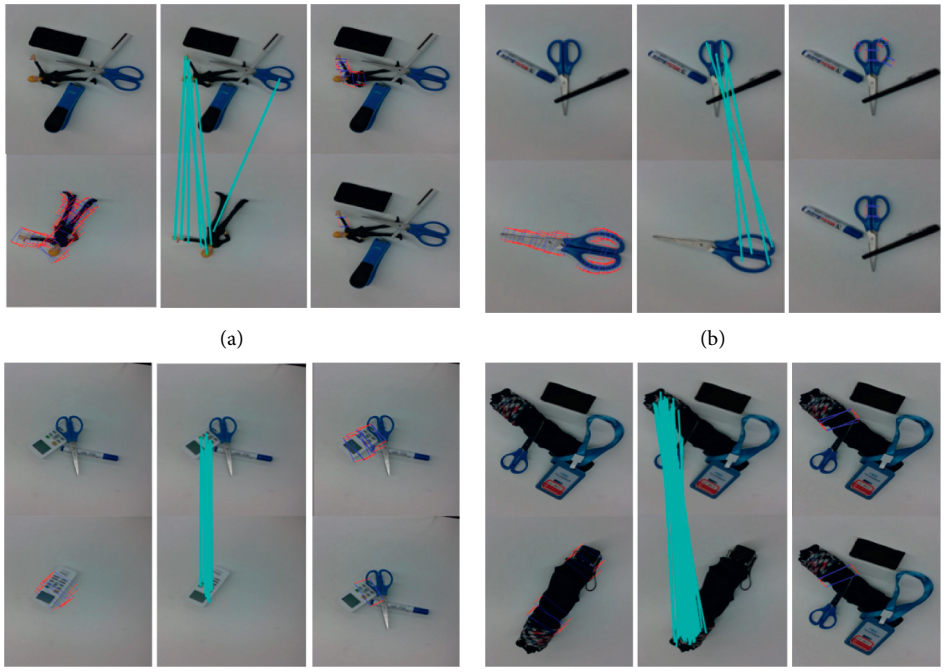
Bold values indicate the performance of our algorithm on Cornell Grasp Dataset. IW: image-wise. The dataset is divided based on image randomly. Each image has an equal probability to be trained or tested. This is a common way to test the generalization of the network to new orientation and position about objects it has seen before. OW: object-wise. The dataset is divided based on object instances. Objects in training set and test set can be different. OW is used to test the generalization ability of a network about new object.

Figure 6: The results of grasp detection on several images of Cornell Grasp Dataset.

Experiment 3 is grasping some household objects in the occluded case. Note that, our algorithm performs well when there contains plenty of SIFT features, and the degree of texture richness and occlusion of the object determine the performance of our algorithm. There are several randomly placed objects in the grasp range of the robot, our matching algorithm generates a certain number of matching points across two images, and the blue lines show the connection of each pair of matching feature points. From the matching feature pairs, we can obtain the transformation matrix M' using the feature points' location information of matching images, and finally we can get the transformation between template images and scene images. Our grasp detection algorithm can avoid the occluded parts and predict a suitable grasp configuration for robot. The results reveal the usefulness of our algorithm (see Figure 8). Our algorithm can predict suitable grasp configurations for the target objects and help the robot to grab the target objects without grabbing other objects.

(a) (b)

Figure 7: Comparsion of our matching algorithm with the original SIFT algorithm on household objects: (a) the result generated by the original SIFT algorithm; (b) the result obtained by our matching algorithm. Our matching algorithm decreases the influence of occluded parts and predicts a more suitable grasp configuration than the original SIFT algorithm.



(a) (b)

(c) (d)

Figure 8: Grasp detection results on household objects under occlusions: (a) detection results of a garage kit, the first row of (a) is the input information and template information, the target object on scene image is occluded by other objects, and the template image only contains the target object and its grasp configurations; the second row shows the SIFT feature pairs obtained by our matching algorithm, and we connect each matching SIFT features using a blue line; and the last row is the detecting results with multiple-rectangle above and most suitable rectangle below, and all the results avoid the occluded parts. (b), (c), and (d) have the same layout. We connect the matching features using blue lines.

6. Conclusion

This paper proposes a grasp detection algorithm to predict grasp rectangles for objects in occluded cases, which trains the model based on Cornell Grasp Dataset and obtains grasp rectangles for scene images using our matching algorithm. Every image in the dataset contains only one object, but our algorithm can predict grasp configurations for images with multiple objects. Experiment results demonstrate the improvement of our algorithm. We evaluate our algorithm on Cornell Grasp Dataset and receive the accuracy of 97.2% on image-wise and 92.5% on object-wise, respectively. In order to verify the effect of our algorithm in occluded cases, we preform experiment in multiobject, occluded condition. The outcome shows that this is a feasible method to utilize our grasp detection algorithm to obtain grasp rectangles in occluded condition; this is the advantage over [1]. The disadvantage of this method is that the result of the matching algorithm influences the final prediction and reduces the detection accuracy on Cornell Grasp Dataset in comparison to [1]. Future work will focus on improving the robustness of the matching algorithm while ensuring the detection accuracy of network.

Data Availability

The data used to support this study are available upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was partially funded by the China National R&D Key Research Program (2019YFB1705703 and 2020YFB1711204).

References

[1] F.-J. Chu, R. Xu, and P. A. Vela, "Real-world multiobject, multigrasp detection," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3355–3362, 2018.

[2] A. Bicchi and V. Kumar, "Robotic grasping and contact: a review," in *Proceedings of the 2000 ICRA, Millennium Conference, IEEE International Conference on Robotics and Automation, Symposia Proceedings (Cat. No.00CH37065)*, vol. 1, pp. 348–353, San Francisco, CA, USA, 2000.

[3] K. Bousmalis, A. Irpan, P. Wohlhart et al., "Using simulation and domain adaptation to improve efficiency of deep robotic grasping," ", in *Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4243–4250, Madrid, Spain, 2018.

[4] D. Quillen, E. Jang, O. Nachum, C. Finn, J. Ibarz, and S. Levine, "Deep reinforcement learning for vision-based robotic grasping: a simulated comparative evaluation of off-policy methods," in *Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6284–6291, Madrid, Spain, 2018.

[5] J. Zhang, M. Li, Y. Feng, and C. Yang, "Robotic grasp detection based on image processing and random forest," *Multimedia Tools and Applications*, vol. 79, no. 3, pp. 2427–2446, 2020.

[6] Y. Chao, X. Chen, and N. Xiao, "Deep learning-based grasp-detection method for a five-fingered industrial robot hand," *IET Computer Vision*, vol. 13, no. 1, pp. 61–70, 2019.

[7] S. Kumra and C. Kanan, "Robotic grasp detection using deep convolutional neural networks," in *Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 769–776, Vancouver, Canada, September 2017.

[8] J. Redmon and A. Angelova, "Real-time grasp detection using convolutional neural networks," in *Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1316–1322, IEEE, Seattle, WA, USA, 2015.

[9] H. Lin, T. Zhang, Z. Chen, H. Song, and C. Yang, "Adaptive fuzzy Gaussian mixture models for shape approximation in robot grasping," *International Journal of Fuzzy Systems*, vol. 21, no. 4, pp. 1026–1037, 2019.

[10] L. Bo, X. Ren, and D. Fox, "Unsupervised feature learning for rgb-d based object recognition," in *Experimental Robotics*, pp. 387–402, Springer, Berlin, Germany, 2013.

[11] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 421–436, 2018.

[12] J. Mahler and K. Goldberg, "Learning deep policies for robot bin picking by simulating robust grasping sequences," in *Proceedings of the Conference on Robot Learning, PMLR*, pp. 515–524, Mountain View, CA, USA, November 2017.

[13] J. Zhang, C. Yang, M. Li, and Y. Feng, "Grasping novel objects with real-time obstacle avoidance," in *Proceedings of the International Conference on Social Robotics*, pp. 160–169, Springer, Qingdao, China, 2018.

[14] D. Morrison, P. Corke, and J. Leitner, "Closing the loop for robotic grasping: a real-time, generative grasp synthesis approach," 2018, https://arxiv.org/abs/1804.05172.

[15] Z. Zhou, T. Pan, S. Wu, H. Chang, and O. C. Jenkins, "Glassloc: plenoptic grasp pose detection in transparent clutter," 2019, https://arxiv.org/abs/1909.04269.

[16] A. T. Miller and P. K. Allen, "Graspit! a versatile simulator for robotic grasping," *IEEE Robotics & Automation Magazine*, vol. 11, no. 4, pp. 110–122, 2004.

[17] R. Pelossof, A. Miller, P. Allen, and T. Jebara, "An SVM learning approach to robotic grasping," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 4, pp. 3512–3518, Washington, DC, USA, 2004.

[18] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis—a survey," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 289–309, 2013.

[19] T. Li, F. Wang, C. Ru, Y. Jiang, and J. Li, "Keypoint-based robotic grasp detection scheme in multi-object scenes," *Sensors*, vol. 21, no. 6, p. 2132, 2021.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.

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

[22] D. Guo, F. Sun, H. Liu, T. Kong, B. Fang, and N. Xi, "A hybrid deep architecture for robotic grasp detection," in *Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1609–1614, IEEE, Marina Bay Sands, Singapore, 2017.

[23] D. Guo, T. Kong, F. Sun, and H. Liu, "Object discovery and grasp detection with a shared convolutional neural network," in *Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2038–2043, IEEE, Stockholm, Sweden, 2016.

[24] M. Vohra, R. Prakash, and L. Behera, "Real-time grasp pose estimation for novel objects in densely cluttered environment," in *Proceedings of the 2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pp. 1–6, IEEE, New Delhi, India, 2019.

[25] H. Zhang, X. Lan, S. Bai, X. Zhou, Z. Tian, and N. Zheng, "Roi-based robotic grasp detection for object overlapping scenes," in *Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4768–4775, IEEE, Venetian Macao, Macau, 2019.

[26] D. G. Lowe, "Object recognition from local scale-invariant features,"vol. 2, pp. 1150–1157, in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, pp. 1150–1157, IEEE, Kerkyra, Greece, 1999.

[27] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[28] D. G. Lowe, "Local feature view clustering for 3d object recognition," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR*, Kauai, HI, USA, 2001.

[29] X. Hu, Y. Tang, and Z. Zhang, "Video object matching based on sift algorithm," in *Proceedings of the 2008 International Conference on Neural Networks and Signal Processing*, pp. 412–415, IEEE, Zhenjiang, China, 2008.

[30] F. Dellinger, J. Delon, Y. Gousseau, J. Michel, and F. Tupin, "Sar-sift: a sift-like algorithm for sar images," *IEEE*

Transactions on Geoscience and Remote Sensing, vol. 53, no. 1, pp. 453–466, 2014.

[31] F. Alhwarin, C. Wang, D. Ristić-Durrant, and A. Gräser, "Improved sift-features matching for object recognition," in *Proceedings of the Visions of Computer Science-BCS International Academic Conference*, pp. 179–190, London, UK, 2008.

[32] G. Shi, X. Xu, and Y. Dai, "Sift feature point matching based on improved RANSAC algorithm,"vol. 1, pp. 474–477, in *Proceedings of the 2013 5th International Conference on Intelligent Human-Machine Systems and Cybernetics*, vol. 1, pp. 474–477, IEEE, Hangzhou, China, 2013.

[33] E. Delponte, F. Isgro, F. Odone, and A. Verri, "SVD-matching using sift features," *Graphical Models*, vol. 68, no. 5-6, pp. 415–431, 2006.

[34] O. Pele and M. Werman, "A linear time histogram metric for improved sift matching," in *Proceedings of the European Conference on Computer Vision*, pp. 495–508, Springer, Marseille, France, 2008.

[35] R. I. Lab, Cornell Grasping Dataset, 2017, http://pr.cs.cornell.edu/grasping/rect_data/data.php/.

[36] Y. Jiang, S. Moseson, and A. Saxena, "Efficient grasping from RGBD images: learning using a new rectangle representation," in *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*, pp. 3304–3311, Shanghai, China, 2011.