Viewpoint optimization for aiding grasp synthesis algorithms using reinforcement learning

Calli, B.; Caarls, W.; Wisse, M.; Jonker, P.

DOI
10.1080/01691864.2018.1520145

Publication date
2018

Document Version
Accepted author manuscript

Published in
Advanced Robotics

To appear in *Advanced Robotics*
Vol. 00, No. 00, January 2013, 1–16

FULL PAPER

Viewpoint Optimization for Aiding Grasp Synthesis Algorithms using Reinforcement Learning

B. Calli[a], W. Caarls[b], M. Wisse[c] and P. Jonker[c]

[a]Worcester Polytechnic Institute, Computer Science Department, Robotics Engineering Program, 85 Prescott St, Worcester, MA-01605, USA; [b]Pontifical Catholic University of Rio de Janeiro, Department of Electrical Engineering, R. Marquês de São Vicente, 225 - Gávea, RJ, 22451-900, Rio de Janeiro; [c]Delft University of Technology, BioMechanical Engineering Department, Mekelweg 2, 2628 CD, Delft The Netherlands.

(*Received 00 Month 201X; accepted 00 Month 201X*)

Grasp synthesis for unknown objects is a challenging problem as the algorithms are expected to cope with missing object shape information. This missing information is a function of the vision sensor viewpoint. The majority of the grasp synthesis algorithms in literature synthesize a grasp by using one single image of the target object and making assumptions on the missing shape information. On the contrary, this paper proposes the use of robot's depth sensor actively: we propose an active vision methodology that optimizes the viewpoint of the sensor for increasing the quality of the synthesized grasp over time. By this way we aim to relax the assumptions on the sensor's viewpoint and boost the success rates of the grasp synthesis algorithms. A reinforcement learning technique is employed to obtain a viewpoint optimization policy, and a training process and automated training data generation procedure are presented. The methodology is applied to a simple force-moment balance-based grasp synthesis algorithm, and a thousand simulations with five objects are conducted with random initial poses in which the grasp synthesis algorithm was not able to obtain a good grasp with the initial viewpoint. In 94% of these cases, the policy achieved to find a successful grasp.

Keywords: active vision, reinforcement learning, robotic grasping, viewpoint optimization

1. Introduction

Grasp synthesis for unknown objects aims to enable robotic grasping even when the shape of the target object is not known by a robot *a priori*. To achieve this goal, an option is to generate a full model of the target object first, and then apply a grasp synthesis algorithm for known objects [1–3]. This approach is far from efficient since the robot should acquire images from all the viewpoints that are necessary for a full shape reconstruction. Consequently, various techniques are developed in the literature to solve the grasp synthesis problem with partial shape information, and the majority of these methods use a single image of the target object and make implicit or explicit assumptions on the missing part of the shape information: for example, the feature-based methods in [4–9] aim to find visual features that correspond to good grasping locations. These approaches assume that the sensor observes the object from a specific viewpoint and train their system accordingly. The method in [10] assumes that the object is symmetric, completes the shape model by calculating the symmetry axis and then synthesizes a grasp. The method in [11] fits shape primitives to the partial shape information and uses predefined grasps for the assigned shape. This approach assumes that the correct shape primitive can be fit from the given viewpoint and that the assigned primitive represents the shape well enough for grasp

CONTACT B. Calli Email: bcalli@wpi.edu

synthesis. The knowledge base method in [12] uses a database of objects represented by oriented bounding boxes together with corresponding grasps. As the partial shape of the target object is acquired, the oriented bounding box model is generated and the best match is found from the database. This method assumes that the oriented bounding box model that is fit from the current viewpoint is sufficient to find a good match from the database.

The significance of the sensor viewpoint for grasp success can be observed by analyzing the experimental results of the above-mentioned papers. In these works, the experiments are conducted with several objects for various object poses (viewpoints), and 1) in none of the papers, 100% success rate is reported for all the objects, 2) for none of the objects, a 0% success rate is reported. This shows that the assumptions on the missing information (which is a function of the sensor viewpoint) do not always hold even for a limited set of objects, but if good viewpoints are supplied, these algorithms can achieve successful results even for challenging objects.

This paper presents an active vision methodology for changing the viewpoint of a depth sensor in order to aid grasp synthesis algorithms. For this purpose an exploration policy is obtained using a reinforcement learning method that provides exploration directions to an eye-in-hand system. In contrast to the majority of the active vision strategies in the robotic manipulation literature (e.g. [13–16]), our policy does not aim to obtain a full model of the target object, but focuses to find a sufficiently good grasp in an efficient way by brief camera motions. For each visited viewpoint, the collected data about the object shape is fused with the previous ones in order to benefit from all the data that is acquired during the process.

The viewpoint optimization policy is obtained by training the system with partial point cloud data that are labeled with a direction for the robot's sensor to follow in order to collect enough data for synthesizing a good grasp. The output of the training process is one single policy that provides an exploration direction for given partial point cloud data. The resulting policy is a generalized viewpoint optimization strategy that is also applicable to the objects outside the training data set. Since obtaining the training data with manual labelling is a cumbersome process, an automated training data generation procedure is also presented.

The proposed viewpoint optimization method can be applied to various grasp synthesis techniques in literature for increasing their success rate. Naturally, it can only be applied to systems with movable sensors. The methodology also requires a quality value both for training the system and terminating the viewpoint optimization process. This value can be the quality of the synthesized grasp as well as the quality of the data, both of which can be supplied by the majority of the algorithms in literature as most algorithms already use a quality measure for their internal grasp optimization process.

We applied the proposed methodology to an eye-in-hand system with a depth sensor, and adopted a simple force-moment balance-based grasp synthesis strategy. Simulations are conducted to evaluate the performance of our strategy and to compare it with random and heuristic-based exploration techniques. Five objects are chosen with various shapes, and a thousand simulations are conducted with each method for random poses of the objects. The results demonstrate that the proposed viewpoint optimization enables a good grasp for the 94% of the cases in which synthesizing a good grasp was not possible for the grasp synthesis algorithm from the initial viewpoint. It is also seen that the proposed method is superior to the random and heuristic-based exploration methods in terms of both success rate and efficiency.

This paper is organized as follows: we present the related work in Section 2. The viewpoint optimization methodology is explained in Section 3. Following that the details of our implementation is given in Section 4. The simulation results are presented and discussed in Section 5. In the last section, the paper is concluded with discussions and future work.

2. Related Work

Active vision methods are utilized in many robotics applications, e.g. surveillance, inspection, object recognition, tracking, path planning (a comprehensive list and a comparison of methods can be found in [17] and [18] respectively). Among these application, the next best view planning problem for object recognition (active object recognition) [19] has resemblance to the viewpoint optimization for grasp synthesis. Active object recognition algorithms provide methods to alter the viewpoint of the vision sensor in order to obtain a descriptive view of a 3D object for recognizing it. One of the common solutions in literature is view selection by increasing the mutual information [20–22]. By this approach, planning the next viewpoint is conducted by searching the whole action space for the maximum object class and observation mutuality considering the previous actions and observations. Another common approach is increasing the discriminative information among the class predictions by entropy minimization [23, 24]. Learning techniques are also utilized for active object recognition, in which a policy that maps states to actions is learned for increasing the discriminative information. A very common way of learning this policy is via reinforcement learning algorithms [25–27]. While designing our viewpoint optimization strategy for grasp synthesis, we applied a similar framework to [27] with the following differences:

(1) We train our system specifically for boosting grasp synthesis instead of object recognition performance, which requires a completely different process.
(2) Our method is designed to generate brief and continuous motions; the policy does not output discrete camera positions on the viewsphere for the next best view, but generates local exploration directions in the camera coordinate frame.
(3) While the camera moves on the viewsphere, we continuously fuse the acquired point clouds in order to benefit from all the available data. In this way, even brief motions can provide the required information for a good grasp. This approach requires a different state description than [27].

In the robotic manipulation literature, active vision is often utilized for known object models [28, 29]. In this case, the role of viewpoint optimization is to localize the target object. In [13–16], a complete 3D model of the object is generated prior to manipulation. In [30] a method is proposed to obtain optimal estimates of the intrinsic and extrinsic camera parameters for achieving higher quality 3D models of the target object. On the other hand, very few works utilize active vision that use partial object data. In [31], active vision is used to refine the surface reconstruction of the grasp location candidates, which results in a more reliable grasp execution. [32] addresses the problem of grasping unknown objects in the presence of occlusion. In this case, the active vision system aims to minimize occlusions on the grasp handles of the objects. In a similar vein, [33] uses active vision to minimize occlusions in a bin-picking problem. [34] utilizes active vision for recognizing the object in clutter, and estimating its pose. Different than these approaches, our method aims to conduct active vision directly for the purpose of boosting grasp success by obtaining a policy that searches for a successful grasp given the current state.

3. Viewpoint Optimization Methodology

The purpose of viewpoint optimization is to change the viewpoint of the sensor to increase the quality of a synthesized grasp. The proposed exploration scheme is as follows.

3.1 *The Viewpoint Optimization Scheme*

In this section, an overview of the viewpoint optimization scheme is presented. The sensor is constrained to move on a viewsphere around the object as in [35]. Therefore, the sensor is always pointed to the target object and its position on the virtual sphere is altered by a viewpoint

optimization strategy.

The proposed viewpoint optimization scheme can be seen in Figure 1. First, data are acquired from the sensor and fused with the data collected from the other visited viewpoints during the process (the fusion step is skipped in the first cycle since the system is still at the initial viewpoint). Then, a grasp is synthesized using the fused data. If the quality of the grasp is good enough (above a predefined grasp quality threshold), then the procedure is terminated and the grasp is executed. If the grasp quality is not sufficient, then the fused data are converted to a state representation and sent to the viewpoint optimization policy obtained by the training procedure. The policy returns a direction to follow in order to lead the robot to a successful grasp. This direction is defined with respect to the sensor's coordinate frame. The sensor is moved to this direction on the viewsphere for a certain distance, called a step in this paper. From this new position (viewpoint) new data are acquired, and the procedure is run again until a successful grasp is obtained.

This scheme can work with many grasp synthesis algorithms in literature; the only prerequisites to apply the scheme are the provision of a quality value and a quality threshold that signifies good grasps. As can be seen in Figure 1, the quality value and its threshold are used to terminate the viewpoint optimization process. Moreover, these values are necessary while training the system for obtaining the viewpoint optimization policy. Supplying a grasp quality is a common feature of many grasp synthesis algorithms as they already use a quality measure for optimizing grasping points and/or a grasp pose. Therefore, the quality of the best synthesized grasp can be used as the quality of the viewpoint. With this interpretation, the whole procedure can be seen as two cascaded optimizations: The first optimization stage is the grasp synthesis algorithm which aims to find the best grasp for given data. The second optimization is the viewpoint optimization that aims to supply better data to the grasp synthesis algorithm.

Even though the details of our implementation will be presented in Section 4, we would like to give a brief overview here to provide a concrete example for the components of the scheme in Figure 1. In our implementation, we used an eye-in-hand system: a manipulator with a depth sensor attached next to its gripper. The sensor supplies point cloud data of the scene. In the data fusion stage, we stitch the point clouds that are acquired during the process. As grasp synthesis algorithm, we used a force-moment balance metric in 3D. As a state representation, we utilized height accumulated features (HAF) [36] to model the stitched point cloud. We add to this representation the position of the sensor relative to its initial position and a model of the unexplored regions. The state is the input of the viewpoint optimization policy that is obtained by a reinforement learning procedure. The policy generates a direction that is relative to the depth sensor. We discretize the direction as north, northwest, west, southwest, south, southeast, east and northeast, and the policy returns one of these values. The robot moves the depth sensor for a step, so that the direction is followed while staying on the viewsphere. From the new viewpoint of the sensor the procedure is run again until a successful grasp is obtained.

The viewpoint optimization policy aims to collect data for achieving a successful grasp with the lowest number of iterations (steps) possible. The procedure of obtaining this policy by reinforcement learning is explained next.

3.2 *The Training Process*

The aim of the training is to obtain an exploration policy that supplies an exploration direction to the sensor for the current state (a representation of the partial shape data of the target object combined with other state information i.e. the position of the robot and a representation of the unexplored regions). For this purpose, we propose the use of approximate reinforcement learning techniques. This method requires us to estimate the expected utility of moving in a certain direction, starting from a given state. The optimal exploration direction, which maximizes the expected utility, should make the sensor collect enough information for synthesizing a sufficiently good grasp in the shortest amount of steps. As a result a policy is obtained, which returns an

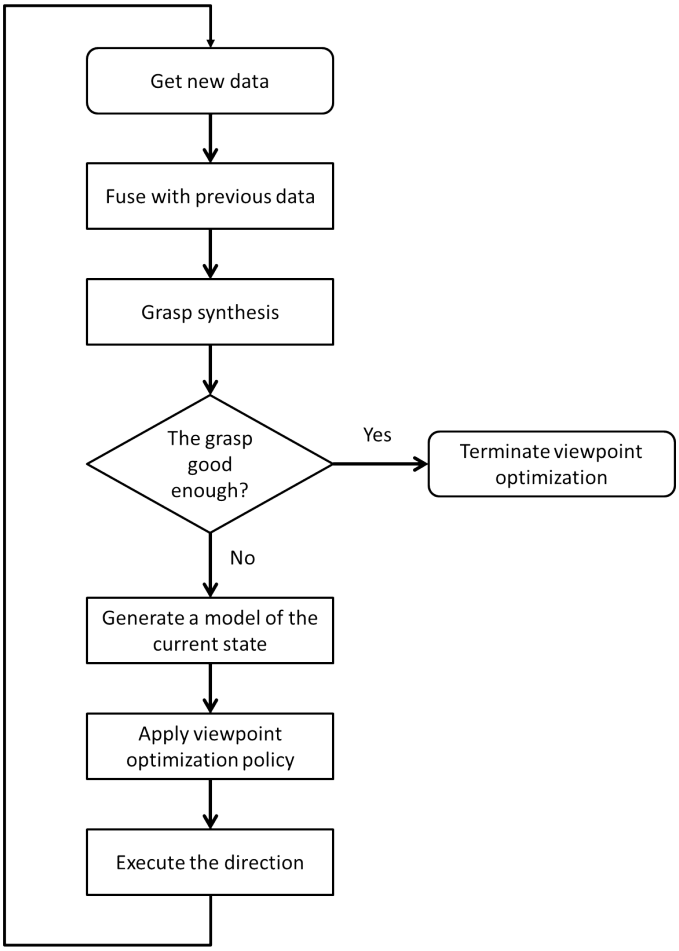

Figure 1.: The exploration scheme

exploration direction for a given state.

Formally speaking, the exploration policy is obtained by approximate Monte Carlo policy evaluation as follows: Define a policy π as a mapping from states to actions (directions). Its state-action value function $Q^{\pi}(s,a)$ encodes, for every state s, the expected value (grasp quality) of taking action a and following π afterwards. Let π_0 be the random policy, taking uniformly distributed random actions independent of the state. Then

$$\pi_1(s) = \arg\max_a Q^{\pi_0}(s,a), \tag{1}$$

which is greedy with respect to Q^{π_0}, can be used as a first approximation of the optimal policy π^*, which always takes the action that leads to the highest expected value [37]. Because s is continuous, we need to generalize over states. In order to reduce the required sample set, we do not continue the iteration $\pi_0, \pi_1, \ldots, \pi^*$, but directly approximate π_1 by supervised learning.

To create a sample set, we drive the system to a series of random states s, execute every action and record the value of following π_0. We then apply Eq. 1 and approximate the mapping

$$s \rightarrow \pi_1(s) \tag{2}$$

using a linear classifier. To reduce the dimensionality, principal component analysis and linear discriminant analysis are first applied to the state features.

It is important to note that the exploration policy depends on the grasp synthesis algorithm; since the best viewpoint can be different for different algorithms, the system should be trained

per algorithm. This means training data that is labeled according to the characteristics of the algorithm is necessary. Generating labeled training data manually is a cumbersome and difficult process, since determining the best exploration direction for a given state considering the characteristics of the algorithm may not be straightforward. For this purpose, an automated training data generation procedure that can be implemented in a simulation environment is also proposed in this paper. This procedure is explained next.

3.3 *Automated Training Data Generation Procedure*

The automated training data generation procedure aims to generate data for the policy approximation. This data is made up of various states and corresponding exploration directions. Here, we decouple direction determination and state representation stages; we assign the exploration directions to fused raw data obtained by the vision sensor, and then generate the state representation using the raw data. In this way, the learning performance with various representations can be examined without rerunning the data generation procedure.

The automated training data generation procedure can be seen in Appendix A. The procedure consists of the following main steps:

(1) Finding a random initial sensor position for which a good grasp cannot be generated,
(2) Finding the direction that leads to a good grasp with fewest number of steps,
(3) Assigning this direction to the raw data.

The procedure is run for all the objects in the training dataset for the requested data number. The main steps are explained below.

3.3.1 Finding a random initial sensor position

In the beginning of the process, first the target object is spawned. Then the robot is moved to a random initial position. Following that, an image is taken in that position and the best grasp for the given data is synthesized. If the quality of this grasp is already above a certain threshold, then the procedure is restarted since viewpoint optimization is not necessary. If the grasp quality is not sufficient, then the image is recorded to the *image_arr* array. Next, the best exploration direction for that position is found with the following procedure.

3.3.2 Finding the best exploration direction

The procedure for determining the exploration direction that leads to a good grasp in the fewest number of steps starts at line 13. The first step of this sub-procedure is to lead the robot in a certain exploration direction k for a step. If a sufficiently good grasp can already be achieved by this motion, then the grasp quality is recorded, and the step number necessary to achieve this grasp is set to 1. In this case, the rest of the loop (random search for a good grasp) is not necessary for this specific direction (k), since a good grasp is already obtained with the fewest number of steps, and the sub-procedure continues for the other directions. If a good grasp cannot be achieved by following this certain direction k for a step, a random search is started (line 23) for finding the step number necessary for obtaining a good grasp from this position. The random search is conducted *no_random_search* times and each search continues until a good grasp is found or the step number exceeds *max_search_step_no*. Within this random search, the minimum step number that leads to a good grasp and the quality of that grasp is recorded to the corresponding entries of arrays *rec_step_no* and *rec_grasp_quality* respectively. If a good grasp cannot be achieved by this random search, then the quality of the highest quality grasp is recorded, and as step number, *rec_step_no*[k] is assigned to *max_search_step_no*. These steps are applied to all directions.

3.3.3 Assigning direction to raw data

By the recorded values obtained from the previous sub-procedure, the direction that achieves a good grasp with fewest number of steps is assigned to the recorded image (line 42). If there

are directions with the same number of steps, then the one with higher grasp quality is chosen. The outputs of the procedure are the arrays *image_arr* and *dir_arr*, which are the raw data and assigned exploration directions respectively. For using these data for training, the raw data should be converted into a state representation.

Here the variety of the object shapes used in the training is important: By the training, we aim to obtain a policy that can work for objects with different shapes. As the shape variety in the training data set increases, the possibility of getting an efficient viewpoint optimization also increases. The other important point is the technique that is used to represent the states. This representation should be descriptive enough, so that the policy can differentiate between different states and make correct decisions.

An application of the proposed methodology is given in the following section.

4. An Implementation of the Viewpoint Optimization Methodology

As a grasp synthesis algorithm we used a force-moment balance-based method for antipodal grasps. We preferred using this method since it is simple and its working principles are straightforward. In this way, we aim to keep the emphasis on the viewpoint optimization strategy and make its results easier to analyze.

For a representation of a state we used a model of the object point cloud, the position of the robot relative to its initial position and a model of the unexplored regions. For modeling the point clouds we used Height Accumulated Features (HAF).

In this section the grasp synthesis algorithm and the formation of the state are explained in detail.

4.1 *Grasp Synthesis Algorithm*

The force-moment balance-based algorithm uses a 3D version of the gradient-angle representation proposed in [7]. This algorithm is efficient to compute and can be calculated as follows: Suppose that there are two grasping point candidates on the object. The quality of the grasp using these points can be calculated by the angles between force application vectors (the normals at those points) and the line that passes through the grasping points (for a 2D visualization see Fig. 2). As the angles get closer to 0 and *pi* radians, the quality of the grasp increases. This metric is utilized in the following way: The input to the algorithm is the latest stitched point cloud of the scene. First, the points that belong to the object are extracted from the point cloud by removing the major plane (points that belong to the table). The point cloud that belongs to the major plane is saved and will be used shortly. Next, downsampling is applied to the object point cloud in order to remove the sensor noise and provide efficiency to the following stages. For each point of the downsampled data, local surface normals are calculated by fitting a plane to a small patch around these points. Each possible pair of the normal vectors is considered as a possible antipodal grasp candidate.

The normal pair that provides the highest quality grasp is found by a brute force search with the following steps: For each normal pair, the force-moment balance-based grasp quality metric is calculated. If none of the pairs can achieve a grasp above a predefined quality (set by the user), the algorithm fails to synthesize a grasp. If there are pairs above the threshold, then starting from the pair with the highest quality grasp, collision is checked between the possible poses of the gripper and 1) the object point cloud, 2) the major plane point cloud and 3) a point cloud that represents the unexplored regions (forming this point cloud is explained shortly). If a pose of the gripper can be found for grasping the object from this pair without colliding any of these point clouds, then a successful grasp is found. If not, the algorithm fails. If there are more than one collision-free gripper pose, the one that is most perpendicular to the object major axis can be chosen.

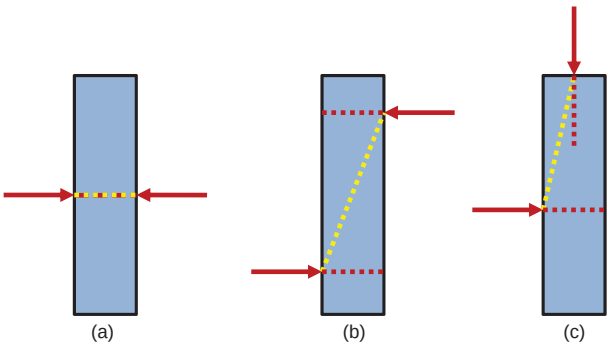

Figure 2.: An illustration of the gradient angle features that are introduced by [7]. For a given set of two grasping point locations, the gradient angle features can be calculated as angles between force application directions (the red dashed lines) and the line that connects the two grasping points (the yellow dashed line). These two angles are expected to be close to 0 and 180 degrees for a good grasp. These features are simple and efficient to calculate and can distinguish between good grasps (i.e. (a)) and bad grasps (i.e. (b) and (c)).

For obtaining a point cloud for unexplored regions, we first cast regularly distributed virtual points around the object in the beginning of the viewpoint optimization process. For each point in this point cloud, ray tracing is applied to check if the point is visible to the camera or obscured by the object. If the point is obscured, it is kept in the point cloud, otherwise it is removed. The point removal procedure is continuously run as the viewpoint of the sensor changes.

An important property of this grasp synthesis algorithm is that it does not make any assumptions on the missing shape: it only uses the data that is acquired during the process and avoids interacting with the unexplored regions of the scene.

4.2 *State Representation*

In this application, our state representation is composed of three components: a model of the partial object shape data, a model of the unexplored regions and the position of the robot relative to its initial position in the viewpoint optimization process.

For modeling the partial object shape point cloud and unexplored region point cloud, we adopt height accumulated features (HAF) explained in [36] as follows: We define a 2D five-by-five grid on the table plane centered at the object. The scale of the grid is calculated so that it covers the whole object point cloud with the smallest grid dimension possible (orientation of the grid is kept the same). In this way, we maintained scale invariance of the model. The distance of each point to the grid plane is calculated and recorded. The points are then projected to the planar grid, and the ones that correspond to the same grid cell are grouped together. For each cell, the largest distance value among the associated points is assigned as a feature value for that cell. In this way, a 25x1 feature vector is formed for each point cloud. For modeling the unexplored regions point cloud, we again use a five-by-five grid, but with a larger scale (1.5 times the scale of the object grid) since the backside of the object is also populated by this point cloud.

The position of the robot is defined on the viewsphere with respect to its initial position using two rotation parameters. Therefore, the overall state vector is the concatenation of the HAF values of the object point cloud (25x1), the HAF values of the unexplored region point cloud (25x1), and the robot position (2x1), in total 52x1 vector.

We conducted simulations to evaluate the performance of the viewpoint optimization strategy with the explained grasp synthesis and state representation techniques. The results are presented in the next section.

Figure 3.: The objects models used in the training: A square prism and a bowl.

Figure 4.: The objects models used in the simulations: A square prism, a rectangular prism, a driller, a glass and a bowl.

5. Simulation Results

The implementation presented in Section 4 is realized in simulation for analyzing the success and efficiency of the proposed viewpoint optimization method. Both for the automated training data generation and for obtaining simulation results, Robot Operating System (ROS) is used together with the Gazebo simulation environment and the Point Cloud Library (PCL).

The system is trained by using two objects, a square prism and a bowl shown in Figure 3. For the training, 200 data are collected for various poses of the square prism and 50 data points for the bowl. For testing the resulting policy, 5 objects are chosen from the household objects database package of ROS which supplies models of real household objects. The shapes and sizes of the square prism-like object and bowl are different from the ones used in the training process. These models can be seen in Figure 4. 200 simulations have been conducted for each object with various poses, standing and lying on a table. However, only poses that are in principle successful are considered. For example, for the rectangular prism, if the object lies on one of the largest two edges, then the gripper opening is not sufficient enough to grasp the object successfully. Therefore, this pose is not used while conducting simulations with this object.

In the simulations, the policy-based exploration is compared with a heuristic-based search technique and a random search technique. As mentioned previously, the search directions are discretized as north, northwest, west, southwest, south, southeast, east and northeast. The algorithms return one of these values for each decision cycle, and the robot follows this direction with respect to the sensor frame for a step. A step is set as 20 degrees on the view sphere. This value is determined experimentally; we chose a value that allows collecting enough new data that would trigger new possibilities for grasp synthesis. Selecting a smaller step size may end up with actions without any/significant outcome, which would make the learning process harder. The step size should be consistent with the one used in the training stage.

With some preliminary simulations that we led the camera manually by user input, we have

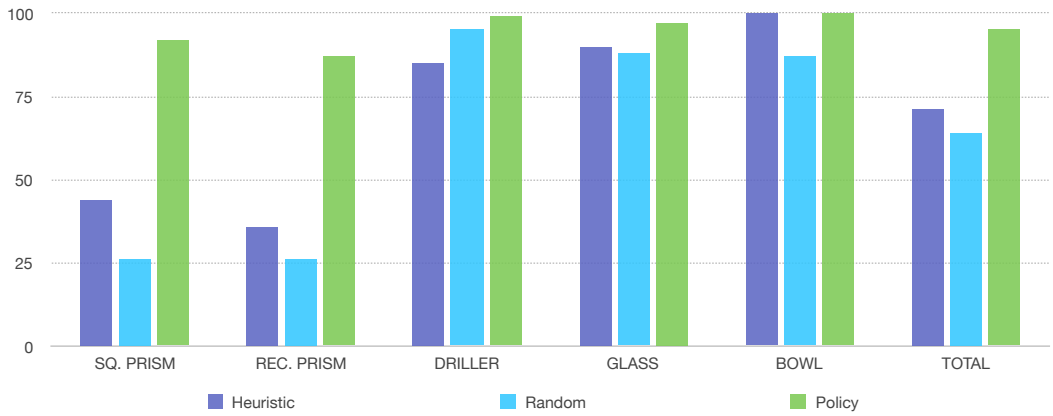

Figure 5.: Success rates of the heuristic-based, random and policy-based exploration techniques for each object after 5 steps (in total 1000 experiments).

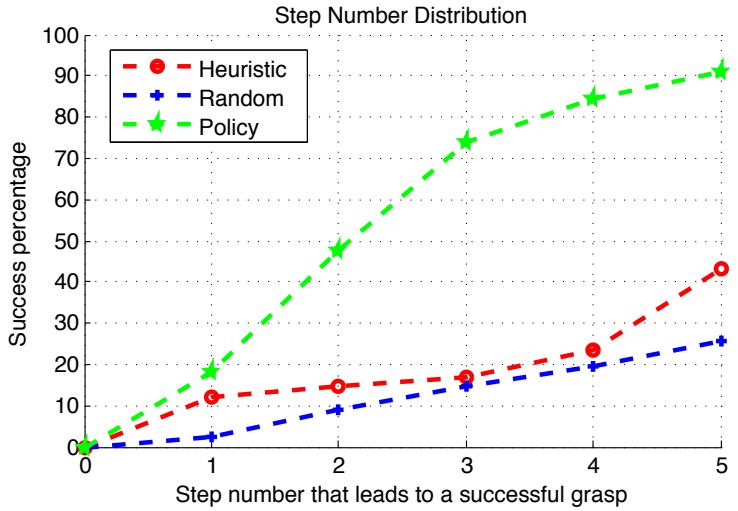

Figure 6.: The results of 200 simulation with each method when using the square prism as a target object.

observed that even in the most difficult cases, five iterations are sufficient to lead to a successful grasp. Therefore, we set the maximum number of iterations to five steps; if an algorithm can achieve a successful grasp within five steps, then the exploration is considered as successful. Otherwise, it is considered as failed. The heuristic based search always takes the steps west, north, north, east, east in this order. With such a strategy, the object can be explored in west, east and north directions of the initial position of the sensor. In the random search, random steps are taken. If any of the algorithms violates the workspace constraints of the robot, then a random step is taken which is forced to be different than the previous decision.

The success percentages for each method and object at the end of the viewpoint optimization process (after 5 steps) are presented in Figure 5. The step number distributions for successful cases are given in Figures 6, 7, 8, 9 and 10. For example, in Figure 6, around 18% success rate is obtained by taking only one step provided by the exploration policy. The success rate increases to around 48% as two steps are taken and so on. As we analyze the results in general we see that the performance of the exploration policy is superior to the heuristic based search and the random search in terms of both success rate and efficiency: For all objects, the policy exploration is more successful and obtains a good grasp within considerably less amount of steps.

Considering each object separately, we see that the policy-based exploration brings a huge advantage for the box shaped objects, square prism and rectangular prism. The sharp edges of

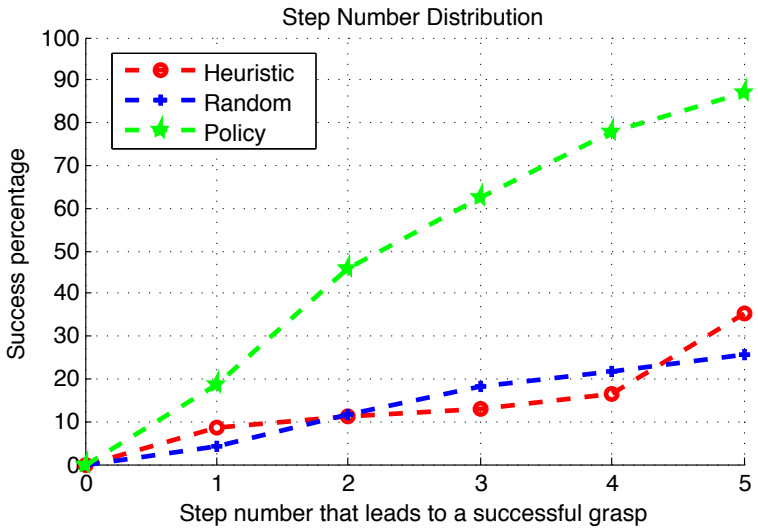

Figure 7.: The results of 200 simulation with each method when using the rectangular prism as a target object.

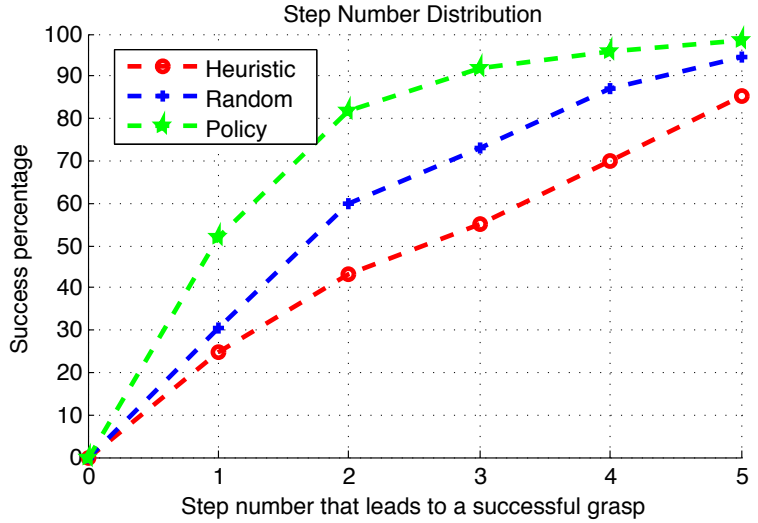

Figure 8.: The results of 200 simulation with each method when using the driller as a target object.

these objects make the job of the exploration algorithms harder than the smooth surfaces, as the algorithm should give correct decisions consecutively for revealing a new object surface. The exploration policy successfully does its job for the majority of the cases which brings 92% and 87% success rates for the square prism and the rectangular prism respectively, whereas the other algorithms perform below reliable rates. For the failed cases, we analyzed the motion trajectories and discovered that even though the algorithm leads the robot to a good viewpoint in the first three steps, bad decisions are made in the last steps. This can be overcome by adding more training data for the later stages of the process.

For simulations with the driller and the glass, the exploration policy obtain a success rate close to 100%. In this case, the heuristic based search and the random search also bring high success rates. This is due to the relatively smooth surfaces of these objects. This shows that using the vision sensor actively in grasp synthesis process, in some cases by not even using an intelligent algorithm, can bring a major benefit. For the bowl object, both the heuristic-based search and

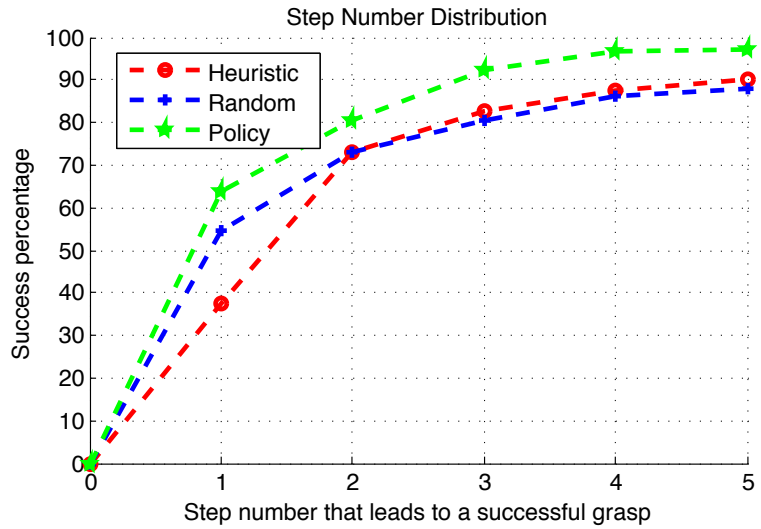

Figure 9.: The results of 200 simulation with each method when using the glass as a target object.

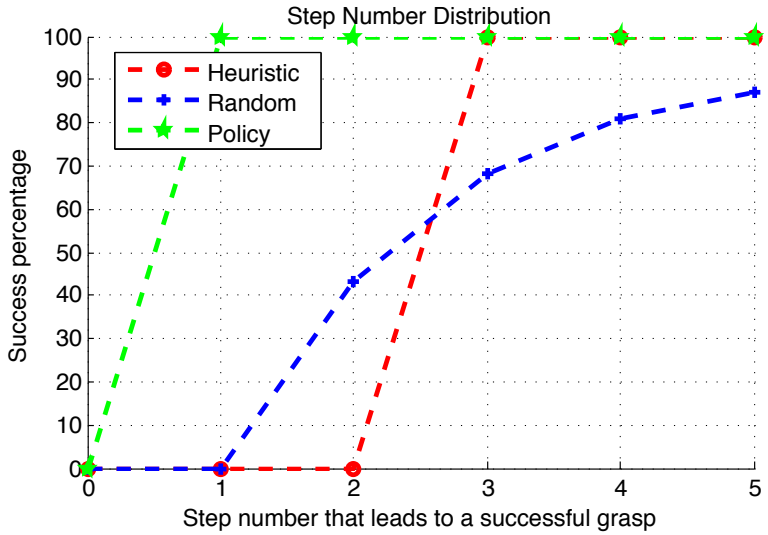

Figure 10.: The results of 200 simulation with each method when using the bowl as a target object.

the exploration policy bring a 100% success rate. The random search becomes successful in 87% of the cases.

Nevertheless, the exploration policy brings a major advantage when it comes to efficiency as can be seen in the step number distribution plots. For example, the average success rate of the exploration policy after two steps is 71.4%, whereas for random and heuristic-based methods, they are 39.4% and 28.4% respectively. For the driller, even though the success rates of all the methods are similar at the end of 5 steps, after only 2 steps the exploration policy performs significantly better with 82% than the random and heuristic methods that achieve 60% and 42% success rates. For the bowl, the exploration policy finds a good grasp in one step, whereas the heuristic method takes three.

Considering the high performance of the exploration policy, one can conclude that using the proposed state representation technique and the learning procedure, a good generalization is obtained even by using two objects in the training process.

6. Conclusions and Future Work

The novelty of this paper lies in the use of a robot's sensor actively in the grasp synthesis process and continuously fusing the gathered data. By the proposed viewpoint optimization methodology, the performance of grasp synthesis algorithms in literature can be boosted considerably. Since manual training data generation is a cumbersome and difficult process, we also presented an automated training data generation procedure. The application of the methodology to a force-moment balance-based grasp synthesis method is given. The simulation results with this method show that the exploration policy obtained via reinforcement learning is superior to a heuristic-based search and a random search in terms of both success rate and efficiency.

In the simulation results, it is shown that even with a limited training set a good generalization can be obtained for objects with various shapes. However, the performance of the algorithm can be improved even further by increasing the number of data and shape variety of the training data set. Also, choosing a more descriptive state representation would help to increase the performance of the algorithm, as with HAF models the concavities of the objects cannot be modelled for some poses of the objects.

References

[1] B. Wang, L. Jiang, J. W. Li, H. G. Cai, and H. Liu. "Grasping Unknown Objects Based on 3D Model Reconstruction". In: *Proc. of the IEEE/ASME International Conf. on Advanced Intelligent Mechatronics*. Monterey, CA, 2005, pp. 461–466.

[2] G. M. Bone, A. Lambert, and M. Edwards. "Automated Modeling and Robotic Grasping of Unknown Three-Dimensional Objects". In: *Proceedings of the IEEE International Conference on Robotics and Automation*. 2008, pp. 292–298.

[3] K. Yamazaki, M. Tomono, T. Tsubouchi, and S. Yuta. "A Grasp Planning for Picking up an Unknown Object for a Mobile Manipulator". In: *Proceedings of the IEEE International Conference on Robotics and Automation*. May 2005, pp. 2143–2149.

[4] A. Saxena, J. Driemeyer, J. Kearns, C. Osondu, and A. Y. Ng. "Learning to grasp novel objects using vision". In: *Proceedings of 10th International Symposium of Experimental Robotics*. 2006.

[5] A. Saxena, J. Driemeyer, and A. Y. Ng. "Robotic Grasping of Novel Objects using Vision". In: *The Int. Journal of Robotics Research* 27 (2008), pp. 157–173.

[6] A. Saxena, L. L. S. Wong, and A. Y. Ng. "Learning grasp strategies with partial shape information". In: *Proc. of the 23rd National Conference on Artificial intelligence*. Vol. 3. 2008, pp. 1491–1494.

[7] Q. Le, D. Kamm, A. Kara, and A. Ng. "Learning to grasp objects with multiple contact points". In: *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*. May 2010, pp. 5062–5069.

[8] E. Klingbeil, D. Rao, B. Carpenter, V. Ganapathi, A. Ng, and O. Khatib. "Grasping with application to an autonomous checkout robot". In: *Proc. of IEEE Int. Conference Robotics and Automation (ICRA)*. 2011, pp. 2837–2844.

[9] Y. Jiang, S. Moseson, and A. Saxena. "Efficient grasping from RGBD images: Learning using a new rectangle representation". In: *Proc. of the IEEE International Conference on Robotics and Automation (ICRA),* 2011, pp. 3304–3311.

[10] J. Bohg, M. Johnson-Roberson, B. Leon, J. Felip, X. Gratal, N. Bergstrom, D. Kragic, and A. Morales. "Mind the gap - robotic grasping under incomplete observation". In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. 2011, pp. 686–693.

[11] L. Gallardo and V. Kyrki. "Detection of parametrized 3-D primitives from stereo for robotic grasping". In: *Proceedings of the 15th International Conference on Advanced Robotics (ICAR)*. 2011, pp. 55–60.

[12] N. Curtis and J. Xiao. "Efficient and Effective Grasping of Novel Objects through Learning and Adapting a Knowledge Base". In: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2008, pp. 2252–2257.

[13] S. Kriegel, C. Rink, T. Bodenmuller, and M. Suppa. "Efficient next-best-scan planning for autonomous 3D surface reconstruction of unknown objects". In: *Journal of Real-Time Image Processing* 10.4 (2015), pp. 611–631.

[14] M. Krainin, D. Fox, and B. Curless. "Autonomous Generation of Complete 3D Object Models Using Next Best View Manipulation Planning". In: *Proc. of the IEEE Int. Conference on Robotics & Automation (ICRA)*. 2011, pp. 5031–5037.

[15] C. Dune, E. Marchand, C. Collowet, and C. Leroux. "Active rough shape estimation of unknown objects". In: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2008, pp. 3622–3627.

[16] J. Aleotti, D. Lodi Rizzini, and S. Caselli. "Perception and Grasping of Object Parts from Active Robot Exploration". In: *Journal of Intelligent and Robotic Systems: Theory and Applications* 76.3-4 (2014), pp. 401–425.

[17] S. Chen, Y. Li, and N. M. Kwok. "Active vision in robotic systems: A survey of recent developments". In: *The International Journal of Robotics Research* 30.11 (2011), pp. 1343–1377.

[18] G. de Croon, I. Sprinkhuizen-Kuyper, and E. Postma. "Comparing active vision models". In: *Image and Vision Computing* 27.4 (2009), pp. 374–384.

[19] S. Dutta Roy, S. Chaudhury, and S. Banerjee. "Active recognition through next view planning: a survey". In: *Pattern Recognition* 37.3 (2004), pp. 429–446.

[20] J. Denzler and C. M. Brown. "Information theoretic sensor data selection for active object recognition and state estimation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.2 (2002), pp. 145–157.

[21] M. F. Huber, T. Dencker, M. Roschani, and J. Beyerer. "Bayesian active object recognition via Gaussian process regression". In: *Proc. 15th Int Information Fusion (FUSION) Conf.* 2012, pp. 1718–1725.

[22] H. Borotschnig, L. Paletta, M. Prantl, and A. Pinz. "Appearance-based active object recognition". In: *Image and Vision Computing* 18.9 (2000), pp. 715–727.

[23] C. Laporte, R. Brooks, and T. Arbel. "A fast discriminant approach to active object recognition and pose estimation". In: *Proceedings of the 17th International Conference on Pattern Recognition. (ICPR)*. Vol. 3. Aug. 2004, pp. 91–94.

[24] X. S. Zhou, D. Comaniciu, and A. Krishnan. "Conditional feature sensitivity: a unifying view on active recognition and feature selection". In: *Proc. Ninth IEEE Int Computer Vision Conf.* 2003, pp. 1502–1509.

[25] K. Shibata, T. Nishino, and Y. Okabe. "Active perception and recognition learning system based on Actor-Q architecture". In: *Systems and Computers in Japan* 33.14 (2002), pp. 12–22.

[26] F. Deinzer, C. Derichs, H. Niemann, and J. Denzler. "A framework for actively selecting viewpoints in object recognition". In: *International Journal of Pattern Recognition and Artificial Intelligence* 23.4 (2009), pp. 765–799.

[27] J. Defretin, J. Marzat, and H. Piet-Lahanier. "Learning viewpoint planning in active recognition on a small sampling budget: a Kriging approach". In: *the Ninth International Conference on Machine Learning and Applications (ICMLA)*. 2010, pp. 169–174.

[28] D. Holz, M. Nieuwenhuisen, D. Droeschel, J. Stückler, A. Berner, J. Li, R. Klein, and S. Behnke. "Active Recognition and Manipulation for Mobile Robot Bin Picking". In: *Gearing Up and Accelerating Cross-fertilization between Academic and Industrial Robotics Research in Europe: Technology Transfer Experiments from the ECHORD Project*. Ed. by F. Röhrbein, G. Veiga, and C. Natale. Springer International Publishing, 2014, pp. 133–153.

[29] L. P. Kaelbling and T. Lozano-Pérez. "Integrated task and motion planning in belief space". In: *The Int. J. of Robotics Research* 32.9-10 (2013), pp. 1194–1227.

[30] Y. Motai and A. Kosaka. "Hand-eye calibration applied to viewpoint selection for robotic vision". In: *IEEE Transactions on Industrial Electronics* 55.10 (2008), pp. 3731–3741.

[31] E. Arruda, J. Wyatt, and M. Kopicki. "Active vision for dexterous grasping of novel objects". In: *IEEE International Conference on Intelligent Robots and Systems*. 2016, pp. 2881–2888.

[32] G. Kahn, P. Sujan, S. Patil, S. Bopardikar, J. Ryde, K. Goldberg, and P. Abbeel. "Active exploration using trajectory optimization for robotic grasping in the presence of occlusions". In: *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*. May 2015, pp. 4783–4790.

[33] K. Harada, W. Wan, T. Tsuji, K. Kikuchi, K. Nagata, and H. Onda. "Iterative Visual Recognition for Learning Based Randomized Bin-Picking BT - 2016 International Symposium on Experimental Robotics". In: ed. by D. Kulić, Y. Nakamura, O. Khatib, and G. Venture. 2017, pp. 646–655.

[34] S.-K. Kim and M. Likhachev. "Planning for grasp selection of partially occluded objects". In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. 2016, pp. 3971–3978.
[35] L. Paletta and A. Pinz. "Active object recognition by view integration and reinforcement learning". In: *Robotics and Autonomous Systems* 31.1 (2000), pp. 71–86.
[36] D. Fischinger and M. Vincze. "Empty the basket - a shape based learning approach for grasping piles of unknown objects". In: *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2012, pp. 2051–2057.
[37] R. S. Sutton and A. G. Barto. *Reinforcement Learning, An introduction*. 1st. Cambridge, MA, USA: The MIT Press, 1998.

Berk Calli received his Bachelor of Science and Master of Science degrees in Mechatronics Program of Sabanci University, Turkey. His master thesis was on integrated visual servoing and force control for robotic manipulation. He completed his PhD at Delft University of Technology, The Netherlands, in 2015. His Ph.D. thesis introduced active vision algorithms for improving manipulation performance in unstructured environments. He, then, joined Yale University GRAB Lab as a post-doc, and worked on vision-based dexterous manipulation algorithms. He is now an assistant professor in Worcester Polytechnic Institute, focusing on vision-based robotics and environmental sustainability-related robotics applications. He is also the administrator of the Yale-Carnegie Mellon-Berkeley (YCB) benchmarking project, which facilitates performance benchmarking efforts for robotics worldwide.

Wouter Caarls received his M.Sc. degree (with honors) in artificial intelligence from the University of Amsterdam, The Netherlands. He obtained a Ph.D. from the Delft University of Technology, The Netherlands on the subject of the automatic optimization of a parallel computer architecture for smart cameras. He is currently an assistant professor at the department of electrical engineering of the Pontifical Catholic University of Rio de Janeiro, Brazil, investigating the applications of reinforcement learning in robotics. His research interests include robotics, machine learning, optimization, parallel algorithms, and image processing.

Martijn Wisse received the M.Sc. and Ph.D. degrees in mechanical engineering from Delft University of Technology, Delft, The Netherlands. He is currently with Delft University of Technology as a Full Professor. His previous research interests included passive dynamic walking robots. His current research interests include the field of robot manipulators for agile manufacturing, underactuated grasping, open-loop stable manipulator control, design of robotic arms and robotic systems, agile manufacturing, and the creation of startup companies.

Pieter P. Jonker (M'91) received the M.Sc. degree in electrical engineering from Twente University of Technology, The Netherlands, in 1979 and the Ph.D. degree in physics from the Delft University of Technology, Delft, The Netherlands, in 1992. He is currently a Full Professor of Visionbased Robotics with the Bio-Mechanical Engineering Group of the Delft University of Technology (TUDelft), Delft, The Netherlands.With Dr.M.Wisse he runs the Dutch Bio-Robotics Laboratory at the TUDelft. His current research interests include bioinspired real-time embedded vision systems for robotics, surveillance, and augmented reality, and on hierarchical reinforcement learning for walking robots. He is a Fellow of the IAPR.

Appendix A. Automated training data generation procedure

Require:
no_of_objects: number of object models for training,
no_of_data_per_object: number of data per object,
no_search_dir: number of discrete search directions,
no_random_search:number of random search for a direction,
max_search_step_no: maximum number of steps within a random search,
quality_threshold: threshold for the quality of a sufficiently good grasp.

initialize *dir_arr* and *image_arr*.
for $i = 1 : no\_of\_objects$ **do**
  for $j = 1 : no\_of\_data\_per\_object$ **do**
    % *Find a random initial sensor pos. w/o good grasp*
    Spawn object model i with random pose.
    Move sensor to a random pos., while fusing data.
    Synthesize a grasp and get *grasp_quality*.
    if $grasp\_quality \geq quality\_threshold$ **then**
      Decrease j by one, skip rest and continue the loop.
    end if
    Record the image to $image\_arr[j]$.
    % *Find the best direction*
    for $k = 1 : no\_search\_dir$ **do**
      Move to direction k for a step.
      Take an image from the current position.
      Synthesize a grasp and get *grasp_quality*.
      if $grasp\_quality \geq quality\_threshold$ **then**
        $rec\_grasp\_quality[k] = grasp\_quality$
        $rec\_step\_no[k] = 1$
        Skip the rest and continue the loop.
      end if
      % *Find shortest path to grasp from this location*
      for $m = 1 : no\_random\_search$ **do**
        for $n = 1 : max\_search\_step\_no$ **do**
          if $rec\_step\_no[k] < n$ **then**
            Break the loop.
          end if
          Move to a random direction for a step.
          Take an image from the current position.
          Synthesize a grasp and get *grasp_quality*.
          if $grasp\_quality \geq quality\_threshold$ **OR** $u == max\_search\_step\_no$ **then**
            $rec\_grasp\_quality[k] = grasp\_quality$
            $rec\_step\_no[k] = u$
            Break the loop.
          end if
        end for
      end for
    end for

39: % *Update best direction and step limit*
40: Initialize *min_step_no* to *max_search_step_no*.
41: Initialize *max_grasp_quality* to 0.
42: **for** $k = 1 : no\_search\_dir$ **do**
43: **if** $(rec\_step\_no[k] < min\_step\_no)$ **OR** $(rec\_step\_no[k] == min\_step\_no$ **AND** $rec\_grasp\_quality[k] > max\_grasp\_quality)$ **then**
44: $min\_step\_no = rec\_step\_no[k]$
45: $max\_grasp\_quality = rec\_grasp\_quality[k]$
46: $exp\_dir = k$
47: **end if**
48: **end for**
49: $dir\_arr[j] = exp\_dir$
50: **end for**
51: **end for**