2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA) | 979-8-3503-3991-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ETFA54631.2023.10275406

GeoGraspEvo: grasping points for multifingered grippers

1st Ignacio de Loyola Páez-Ubieta
AUtomatics, ROBotics, and Artificial Vision Lab
University Miguel Hernández and University of Alicante
Elche and Alicante, Spain
ipaez@umh.es and ignacio.paez@ua.es -

2th Edison Velasco-Sánchez
AUROVA Lab
University of Alicante
Alicante, Spain
edison.velasco@ua.es -

3nd Santiago T. Puente
AUROVA Lab
University of Alicante
Alicante, Spain
santiago.puente@ua.es -

4rd Pablo Gil
AUROVA Lab
University of Alicante
Alicante, Spain
pablo.gil@ua.es -

5th Francisco A. Candelas
AUROVA Lab
University of Alicante
Alicante, Spain
francisco.candelas@ua.es -

***Abstract*—The task of grasping objects is a simple and routinely action for humans but it is complex for robots. To integrate robots into everyday tasks, they have to be equipped with capabilities human-like dexterity. In this line, we propose an analytic method, called GeoGraspEvo, to compute grasping points to be used by robotic hands with three, four or more fingers. Our proposal uses features computed from visible surface objects captured by a single RGBD image of a scene. Additionally, it uses as input some configurable kinematic parameters to be able to carry out the grasping depending on the hand morphology. The method compute grasping points with no training process.**

***Index Terms*—Robotics, RGB-D Perception, Perception for Grasping, Manipulation Planning, Grasping points.**

I. Introduction and related works

Robots have appeared to remain with us. Little by little, they have been equipped with vision sensors so as to increase its capabilities for grasping and manipulating objects [1] [2]. The aforementioned task of grasping is not new but, as the object possibilities are endless, more and more methods appear every year [3] [4]. Several approaches have been applied to get the best combination of grasping points in order to handle objects. However, the core method used in nearly all of them are based on Neural Network (NN) or mathematical processes [5] [6].

On the one hand, the data driven methods use huge amount of data for training in order to learn all the possible variability of objects. For example, [7] described a system that learnt to handle different kind of objects with a two fingered gripper using RGB images. For that purpose, a NN was trained so as to learn the grasp success while exploring different paths to be followed by the robotic arm. Although they achieved great success, they point out that the task of collecting and labeling the dataset to train the system is a tough work. Also, a huge quantity of objects, robots and cameras were needed. Another work is [8], in which they trained a Convolutional Neural Network (CNN) with real grasping scenes using as input a n-channel image. They obtained as a result the coordinates of the center point and the orientation, width and height of the grasping rectangle. One of the things against this method was that sometimes it failed with complex objects, as well as with small ones. The only way to solve this issue was increasing the diversity during the training phase.

On the other hand, the analytical methods provide the ability of being able to generalise well to previously unseen objects and do not need any kind of previous training. In [9], they developed an algorithm for grasping general objects starting with a complete mesh model of the object or a 360º RGBD image. The initial object was decomposed into several submeshes and approximated into basic shapes in order to choose the most suitable type of grasping. However, they needed as much as 0.3 s to get its results and the method failed for objects with symmetry, as well as they needed as input a complete RGBD image or mesh. Other works like [10] used a 2 fingered gripper to, starting with a RGBD image, unclutter the environment and grasp the objects. Their method achieved great results but they left aside multi-fingered grippers. Also, if the objects in the scene changed its size, the weights of the ranking function should be recalculated, which is unnecessary in our proposed method.

In this work in progress, the baselines for our grasping algorithm called GeoGraspEvo are established. This algorithm is an evolution of the previous version called GeoGrasp [11] but, instead of working with just two fingered grippers, this new version can handle the calculus of grasping points with grippers with three, four or more fingers.

This paper is organised as follows: section II covers the proposed algorithm to clean the point cloud, obtain candidates for each region of the fingerprint, and rank them to obtain the combination that provides the most stable grip, section III shows some examples of the result of applying the proposed algorithm to the point clouds and section IV discuss its

Research work was funded by the Valencian Regional Government through the PROMETEO/2021/075 project and the Spanish Government through the Research Staff Formation (FPI) under Grant PRE2019-088069.

advantages and limitations and deals with current and future planned work to finish the experimental setup for testing the proposed algorithm .

II. Methodology

Our method, called GeoGraspEvo, consists of three stages. It comprises a preprocessing step, in which possible objects are clustered from its background. Later, candidate grasping points are obtained for each finger in the gripper. Finally, those candidates are sorted and punctuated following a formula that takes into account some of their properties.

A. Preprocessing

First of all, a segmentation process of the point cloud is needed so as to isolate each object in the captured scene. The complete process can be seen in Fig. 1. Different algorithms were presented in the literature for this task [12]. Our approach consists of getting a RGDB image such as Fig. 1a, which can be obtained by using a depth camera like the Intel® RealSense™ D435i. This input is used since the algorithm tries to be as general as possible so as to handle unknown objects without basing its success in any previous knowledge of the environment.

Once it is obtained, we apply a band-pass filter to eliminate points of the cloud which correspond with noise and background. This band-pass filter is static and can be modified by the user when running their application. The default case will isolate the points that are in a squared area of 1.0 m x 1.0 m and have between 0.0 and 1.5 m of depth, having as result Fig. 1b. Our method assumes that the objects lay on a flat surface like a table or floor but without overlapping. Thereby, we use RANdom SAmple Consensus (RANSAC) algorithm [13] to obtain an approximation of the table or floor plane. After, the method uses a KdTree algorithm for running a clustering process so as to segregate the scene in point clouds of objects. The flat surface can be seen painted in green in Fig. 1c, meanwhile each of the objects are shown in different colours.

The result is having each object of the scene in a cluster in order to analyse each of them separately with the following stages.

B. Candidate points

In the second stage, the proposed method takes as input the dominant plane and each of the object clusters, both calculated in section II-A, but also some configurable parameters. These parameters are the tip size of the gripper Gts, the number of fingers of the gripper $N+1$, the apertures range for each finger in the gripper $AF_{k\,min}$ $AF_{k\,max}$ with $k \in \mathbb{N}\,[1, N]$, as well as all sets for candidates to grasping point for each finger. Some of these parameters are shown in Fig. 2.

Given an object cluster like Fig. 3a, some parameters like the main axis, the object centroid or its normals are needed for posterior procedures. With the points of the object cluster, a voxelization process is applied in order to reduce the number of points. Once the point cloud has been simplified (black

(a) Complete RGBD scene.

(b) Part of interest after removing noise and background.

(c) Clustered objects with dominant plane.

Fig. 1: Preprocessing step of GeoGraspEvo algorithm.

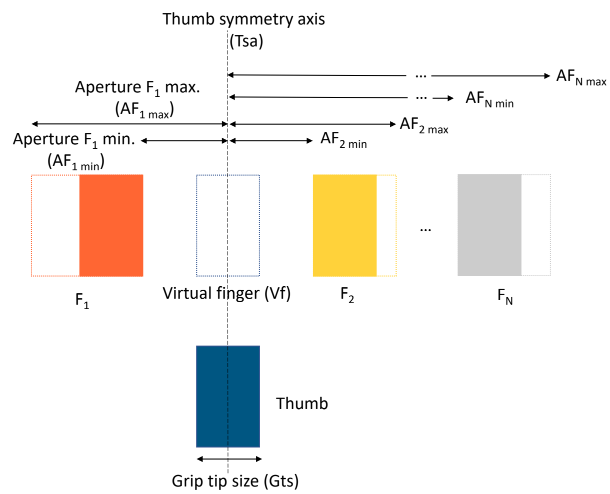


Fig. 2: Configurable parameters of a generic gripper.

points in Fig. 3b), its normals are obtained, as well as the object's main axis (magenta line in Fig. 3b) and centroid (yellow point in Fig. 3b). The voxel radius V_r is defined by (1), from which Gts is the grip tip size of the gripper in mm.

$$V_r = \frac{\frac{Gts}{1000.0}}{2.0} \tag{1}$$

The voxelization process is done following two reasons. The first one is to light up the process in order to save time and computational resources, since close points in an object will have similar properties. The second one is because there is a minimum physical resolution obtained from the specified finger tip.

The next issue to figure out is to find the camera orientation by getting the angle between a standard z axis and the dominant plane. The result will indicate if the object is being observed from the upside or side part. The points that are within 1 cm to the minor axis (generated with the normal of the major axis and the objects' centroid) of the object are subsampled from the object point cloud since these possible candidates have been found to constitute valid possible initial grasping points. These points will form what is called the grasping plane (see green line in Fig. 3b. Initially, it is composed of more points but the following processes overlaps their colour to the initial one). From them, those two points that are the furthest from the dominant plane will be our first two points of interest. They are represented in Fig. 3b by the big red and blue points.

Starting from these two points, some more points will be added. By using a KdTree operation, the points from the object point cloud that are inside the grasping radius G_r are chosen, defined by (2). This radius could be modified if the width of the object w (distance between the aforementioned two points) is smaller than twice the G_r. In this case, the grasping radius would be defined by (3).

$$G_r = \frac{2.0 * Gts}{1000.0} \tag{2}$$

$$G_r = \frac{w * 0.7}{2.0} \tag{3}$$

These candidate grasping points for the thumb have been represented in Fig. 3b by the small red and blue points on both sides of the object. Now it is time get the normalization parameters for each group of points. So, starting from the points of each side of the object, the maximum and minimum values of the following metrics are calculated: a) distance between the centroid of the object and each of the candidate points, b) distance between the dominant plane and each of the candidate points and c) the curvature.

The following steps are applied for getting the candidates for the rest of the fingers of the gripper F_k with $k \in \mathbb{N}\,[1, N]$. In order to get them, the points that are candidates to thumb finger in the other side of the object are chosen (blue points in Fig. 3c and red points in Fig. 3d). From them, the minimum and maximum aperture for each finger is applied in order to get a group of candidate points and its associated normals. Then the KdTree method is applied in order to get more possible candidates. Once obtained, the same normalization process from previous paragraph is applied. The thumb grasping candidate points are represented in red and blue in Fig. 3c and Fig. 3d respectively. In both cases, the other fingers grasping candidate points are represented in magenta and yellow colours.

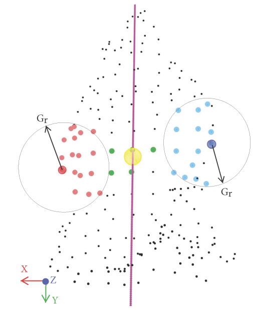
C. Ranking

A combination among all candidates is needed in order to see which tuple provides the highest punctuation. We do it by using the ranking function R, based on [11] but with modifications so as to consider multifingered hands. It is composed of two parts named GP and SP in (4).

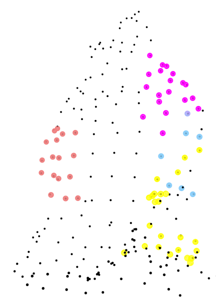
$$R = GP + SP \tag{4}$$

GP indicates geometrical properties, concretely the antipodality, which is if the points are opposite, generating collinear forces. This parameter should be maximized and it is expressed by r_1 and pondered by a weight w_1 in (5).

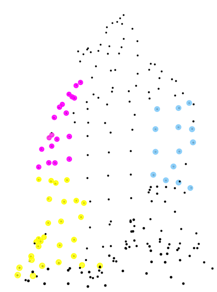
(a) Clustered object to work with.



(b) Centroid, major axis, grasping plane, first two points and neighbours in voxelized point cloud.



(c) Candidates to all three fingers considering thumb at the left part of the object.



(d) Candidates to all three fingers considering thumb at the right part of the object.

Fig. 3: Candidate points step of GeoGraspEvo algorithm.

$$GP = w_1 * r_1 \tag{5}$$

SP indicates stability properties, which are the curvature r_2, perpendicularity r_3, distance r_4 and linearity r_5. Each of them are pondered by a weight w_2, w_3, w_4 and w_5 in (6). The curvature is minimum when the surface of the object is planar, which will give more stability on the grasp. The perpendicularity is the angle between the minor axis and the normal of each point, which should be also minimum. The distance between the fingers should be maximized in order to have a more stable grasping. It should be in range of the proportioned minimum $AF_{k\,min}$ and maximum $AF_{k\,max}$ distance for each finger with $k \in \mathbb{N}\,[1, N]$. The linearity, which is the distance between each grasping point and the line that shows the average value, provide us information about if the object will slide or not when grasping it, being close to zero in the best case.

$$SP = w_2 * r_2 + w_3 * r_3 + w_4 * r_4 + w_5 * r_5 \tag{6}$$

The result of applying the ranking function is shown in Fig. 4. The darker the point is shown, the higher its punctuation

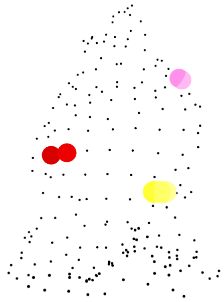


Fig. 4: Ranking step of GeoGraspEvo algorithm.

obtained by the ranking function was.

III. Partial results

A sample scene is provided in order to test the proposed algorithm. The same scene is used in order to compare two different grippers over the same objects. It is worth mentioning that the used objects are from the Yale-CMU-Berkeley (YCB) object dataset [14], concretely from the food items subset. This dataset is well known and broad tested and will guarantee the repeatability of the experiments.

In Fig. 5 the grip tip size is 25 mm. However, in Fig. 5a, a 3 fingered gripper is applied with the apertures (-0.040,-0.045) and (0.035,0.040), meanwhile in Fig. 5b, a 4 fingered gripper is applied with the apertures (-0.015,-0.025), (-0.005,0.010) and (0.035,0.040).

The result shows the thumb's position with a red point, while the magenta and yellow points in Fig. 5a and the same colour points plus cyan points in Fig. 5b are the position of the other fingers of the gripper. With the proportioned configuration, it is impossible to grasp the orange object since it is too small.

(a) Result with a 3 fingered gripper.

(b) Result with a 4 fingered gripper.

Fig. 5: Experiments applying the proposed algorithm.

IV. Conclusions and future works

In this work in progress, GeoGraspEvo algorithm has been introduced. It is an update to the previous version, named GeoGrasp, that provides grasping points in order to manipulate objects with multifingered grippers by using the math properties of the surface points of the objects.

The process is composed of three steps: a) preprocessing, b) candidate points and c) ranking. The first one consists of clustering the objects in the scene and, in the second step, candidates for both the thumb and each of the other fingers of the gripper are obtained. The algorithm is finished in the third step by combining all possible candidates between them in order to check which combination provides the highest score.

Some examples were seen by applying the algorithm with several configurations to a unique point cloud and is has shown promising results.

Future developments will consist of testing the algorithm with simulated and real robots while picking up objects. Currently, a simulation of a complete pipeline to apply the proposed algorithm into a robotic arm (UR5e) with several multifingered grippers (3f ROBOTIQ gripper and Allegro and Shadow hands) is being developed. The dataset is also planned to be extended by incorporating other subset of objects of the YCB dataset.

References

[1] J. Castaño-Amorós, I. d. L. Páez-Ubieta, P. Gil, and S. T. Puente, "Visual-tactile manipulation to collect household waste in outdoor," *Revista Iberoamericana de Automática e Informática industrial*, vol. 20, no. 2, pp. 163–174, 2023. doi:10.4995/riai.2022.18534

[2] R. Grimm, M. Grotz, S. Ottenhaus, and T. Asfour, "Vision-based robotic pushing and grasping for stone sample collection under computing resource constraints," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2021, pp. 6498–6504. doi:10.1109/ICRA48506.2021.9560889

[3] S. Duan, G. Tian, Z. Wang, S. Liu, and C. Feng, "A semantic robotic grasping framework based on multi-task learning in stacking scenes," *Engineering Applications of Artificial Intelligence*, vol. 121, p. 106059, 2023. doi:10.1016/j.engappai.2023.106059

[4] B. Wei, X. Ye, C. Long, Z. Du, B. Li, B. Yin, and X. Yang, "Discriminative active learning for robotic grasping in cluttered scene," *IEEE Robotics and Automation Letters*, vol. 8, no. 3, pp. 1858–1865, 2023. doi:10.1109/LRA.2023.3243474

[5] K. Kleeberger, R. Bormann, W. Kraus, and M. F. Huber, "A survey on learning-based robotic grasping," *Current Robotics Reports*, vol. 1, pp. 239–249, 2020. doi:10.1007/s43154-020-00021-6

[6] H. Tian, K. Song, S. Li, S. Ma, J. Xu, and Y. Yan, "Data-driven robotic visual grasping detection for unknown objects: A problem-oriented review," *Expert Systems with Applications*, vol. 211, p. 118624, 2022. doi:10.1016/j.eswa.2022.118624

[7] 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. doi:10.1177/0278364917710318

[8] H. Cao, G. Chen, Z. Li, J. Lin, and A. Knoll, "Lightweight convolutional neural network with gaussian-based grasping representation for robotic grasping detection," *arXiv preprint*, 2021. doi:10.48550/arXiv.2101.10226

[9] S. I. Ansary, S. Deb, and A. K. Deb, "A novel object slicing-based grasp planner for unknown 3d objects," *Intelligent Service Robotics*, vol. 15, no. 1, pp. 9–26, 2022. doi:10.1007/s11370-021-00397-0

[10] P. Raj, A. Singhal, V. Sanap, L. Behera, and R. Sinha, "Domain-independent disperse and pick method for robotic grasping," in *Int. Joint Conf. on Neural Networks (IJCNN)*, 2022, pp. 1–8. doi:10.1109/IJCNN55064.2022.9892672

[11] B. S. Zapata-Impata, P. Gil, J. Pomares, and F. Torres, "Fast geometry-based computation of grasping points on three-dimensional point clouds," *International Journal of Advanced Robotic Systems*, vol. 16, no. 1, p. 1729881419831846, 2019. doi:10.1177/1729881419831846

[12] L. Tchapmi, C. Choy, I. Armeni, J. Gwak, and S. Savarese, "Segcloud: Semantic segmentation of 3d point clouds," in *Int. Conf. on 3D vision (3DV)*, 2017, pp. 537–547. doi:10.1109/3DV.2017.00067

[13] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981. doi:10.1145/358669.358692

[14] B. Calli, A. Singh, J. Bruce, A. Walsman, K. Konolige, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Yale-cmu-berkeley dataset for robotic manipulation research," *The International Journal of Robotics Research*, vol. 36, no. 3, pp. 261–268, 2017. doi:10.1177/0278364917700714