Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 118 (2023) 253–258

www.elsevier.com/locate/procedia

16th CIRP Conference on Intelligent Computation in Manufacturing Engineering, CIRP ICME '22, Italy

Automatic End Tool Alignment through Plane Detection with a RANSAC-Algorithm for Robotic Grasping

Ludwig Vogt[a,*], Robert Ludwig[a], Johannes Schilp[a,b]

[a]*Augsburg University, Am Technologiezentrum 8, 86159 Augsburg, Germany*

[b]*Fraunhofer IGCV, Am Technologiezentrum 2, 86159 Augsburg, Germany*

* Corresponding author. Tel.: +49 821-598-69315; *E-mail address:* Ludwig.Vogt@uni-a.de

Abstract

Camera based grasping algorithms enable the handling of unknown objects without a complete CAD model. In some scenarios, the captured information from a single view is not sufficient or no grasp is possible. For these cases, the precise realignment of the gripper is difficult because a suitable rotation is part of an infinite solution space. In this paper, we propose a framework which automatically identifies correct rotations from point clouds to adjust the gripper. We validate our approach in a virtual environment for a parallel jaw gripper with multiple isolated and grouped industrial objects.

Keywords: Camera-based grasping; Grasping simulation, RANSAC-algorithm

1. Introduction

A lot of focus in automatic handling tasks for robotic grasping lies in detecting a suitable grasping position for the combination of handling object and gripper. To calculate grasping positions, the existing algorithms can be categorized as offline algorithms [1, 2], where the grasping points are calculated in advance and as online [3, 4] algorithms where the grasps are calculated during the handling task. While the offline algorithms offer a bigger solution set and often more comprehensive evaluation criteria, e.g., friction cone analysis and load analysis, these methods are computationally demanding and restrictive regarding the dynamics of the environmental conditions. Often, these methods demand a static object and fail if a displacement of the object occurs prior to the grasp. In contrast to that, camera-based grasping algorithms [5] offer the possibility to dynamically react to orientation changes and displacements and are therefore suitable for dynamic production scenarios with a high demand for flexibility [6]. The drawback of these methods is a lack of evaluation criteria since only a fraction of the object is detected and not all parameters, e.g., volume, weight and material are known. Another unique property of the camera-based methods is the identification of possible grasps if no grasp can be identified from the initial recording. This occurs if no planar areas exist, parts of the objects are occluded or if the size of the parts does not fit to the dimensions of the gripper. In these cases, a reorientation of the object or camera is necessary to generate a different view of the object and perform a new calculation. In the case of a necessary reorientation a possible infinite solution space exists and offers the challenge to determine a practical transformation.

To conquer this problem, we develop a Random Sample Consensus (RANSAC)-based Algorithm [7] which calculates appropriate transformations for the robotic end tool. Based on the data of depth images from a vision sensor, surface clusters are derived and a grasp evaluation is performed through the positioning of an auxiliary object [8]. If no result can be found neighbor surfaces are identified to determine possible reorientations. Because the accuracy of depth images can be

10.1016/j.procir.2023.06.044

coarse and noisy at times [9] we adapted a RANSAC algorithm to cluster the points of the point cloud. From this set the best solution is selected. Therefore, our method contributes to the field of camera-based grasping algorithms with an efficient reorientation algorithm.

The paper is organized as follows. Section 2 covers the past work in the fields of camera-based grasping algorithms. Afterwards, the methodology of our RANSAC-based reorientation algorithm and the grasping point determination is described. An evaluation and validation of our algorithm with different objects and in different production scenarios is performed in Section 4, where we also discuss our results.

2. State of the Arts

Because our algorithm can be classified as an online algorithm, this Section only convers the relevant literature in this scientific field.

Herzog et al. [10] use a template based algorithm which searches the best shape conformity between heatmaps, which are derived from the point cloud, and the templates from generated training data by demonstration. An empirical algorithm is also used by Nieuwenhuisen et al. [11]. In their work objects are segmented into primitive shapes and represented as graphs. The previously known grasps for these shapes are then be applied to the objects in the scene. While most algorithms try to identify force closure contacts, Pokorny et al. [12] search for form closure contacts to ensure a stable grasp. Their grasping algorithm is based on the identification of graspable loops i.e., “holes” in a convex hull which is derived from the surface mesh of a point cloud. Klingbeil et al. [13] search for convex contours, where the shape of the gripper has a shape conformity. For their evaluation 2D slices are derived from the object and compared to the shape of the gripper, which necessities a comprehensive slicing in multiple directions of the object. Contrary to approaches only using vision data, Schiebner et al. [14] combine visual information with tactile sensor feedback to generate a reactive grasping approach. In their work, the actual grasping point is determined via the calculation of its principal axis and the object extension.

Various methods use neural networks to identify grasping positions. Mahler et al. [3] developed a CNN based grasping algorithm which calculates grasping positions through the use of a depth camera. For the training of their CNN, a large number of annotated training examples is necessary. Over 80% of their grasps are performed with a vacuum gripper and only ~20% with the parallel jaw gripper. Instead of calculating the grasps through a CNN, Varley et al. [15] use a neural network to generate a complete shape of the object and perform a grasp planning in the software *GraspIt!* [16].

Kleeberger et al. [8] use a hybrid system of offline grasping point determination and online 6D-pose estimation. Based on a point cloud representation of the object antipodal point pairs, which satisfy their boundary conditions, e.g., angle, distance and collision in a circular area, are identified.

Huang et al. [17] perform a partial reorientation of the gripper after the initial grasping point determination to optimize the alignment of the gripper and the object. The calculation of their necessary transformation is based on PCA to identify the principal axis of the object.

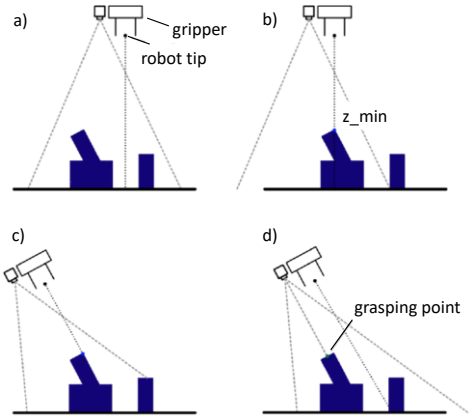

Fig. 1. Procedure of initial alignment a), nearest object detection through z_{min} b), an aimed reorientation c) and the grasping point determination of our proposed algorithm.

In summary, the literature shows many different camera-based grasping approaches which can detect grasps even if only partial information of the object is given. Though, most algorithms are restricted to a single view for the grasping point determination and neglect the efficient search for a reorientation of the camera completely. In other cases, e.g. [3], an evasion strategy (vacuum gripper) is used to perform successful grasps.

3. Methodology

Our methodology contains two main tasks, which can be executed iteratively until a successful grasp is determined. One part is the grasping point detection (cf. Section 0) based on a geometrical auxiliary object, which is similar to the strategy used in [8]. The other part is the end tool alignment (cf. Section 3.2) of the gripper in the case no successful grasp can be determined from the current perspective. Our algorithm starts at an initial positioning above an object or multiple objects with a top-down view on the scene (cf. Fig. 1 a). From this starting position an object of the scene is selected through the value z_{min}, which classifies the nearest object to the end tool and the center of the vision sensor is aligned with the handling object (cf. Fig. 1 b). The alignment is necessary to get an accurate representation of the object, otherwise the influence of shadows increases. From there on, both tasks can be executed in succession until a valid grasping position is detected.

3.1. Grasping Point detection

The grasping point detection calculates if a grasp is possible and with which orientation the object can be grasped from the present perspective.

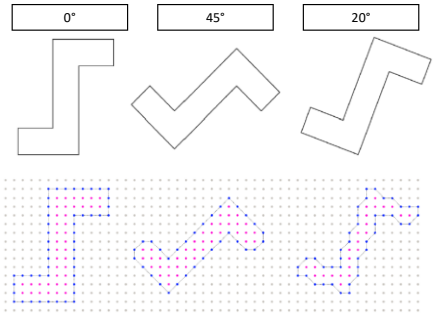

Fig. 2. Classification of surface points and contour points with 0°, 20° and 45° orientation.

Through our parallel positioning of the camera in relation to the surface of the handling object (cf. Section 3.2) all surface points have nearly identical depth values. This depth value z is used to cluster all surface points in a plane where the main part of the grasping point determination is executed. After clustering the points, two sets containing the interior points and the edges of the geometry are derived. Because the next steps of the grasping point determination are based on the contour of the geometry, a good quality of the contour is necessary for an accurate calculation. While a simple connection of the edge points is successful if the edges are aligned with the grid axis, a rotation of the geometry can lead to a bad contour and therefore inadequate input data (cf. Fig. 2).

Therefore, we use a set of rules to group the edge points and calculate the contour to prevent jagged edges (cf. Fig. 2 20° orientation).
Based on the extracted edges of the geometry an actual grasping position is identified with the help of a double T-beam which equals the size of the opened gripper (cf. Fig. 3). The auxiliary object is positioned at the center C of every edge. For a successful grasp, two antipodal edges must fit into the double T-beam. After a successful evaluation, an optimized position of the double T-beam needs to be performed because the center of the double T-beam specifies the plane of the TCP and must therefore be set carefully. If positioned decentral, the closing of the gripper leads to a movement of the object along the closing

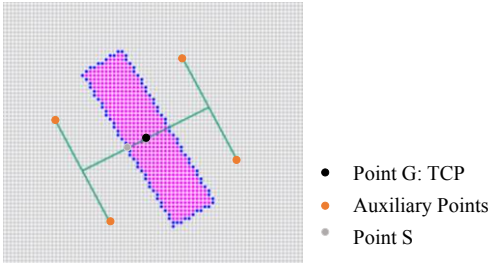

Fig. 3. Positioning of various double t-beams for the grasping point determination at a detected object surface.

axis which we want to avoid. To calculate the grasping middle point G the following formula is used:

$$= \begin{cases} x = S_x + \dfrac{\cos\delta \cdot \frac{c}{2}}{d_p} \\ y = S_y - \dfrac{\sin\delta \cdot \frac{c}{2}}{d_p} \\ z = y \cdot x + x \end{cases} \quad (1)$$

where δ is the angle to the edge of the contour, S the middle of an edge, c the distance between two antipodal edges and d_p the distance between two points. In the next step, the center of the double T-beam is aligned with the calculated middle points G. From the double T-beam 4 points are evaluated if they are positioned inside the geometry or outside. If an inside point is detected, the position is dismissed because a collision between the gripper and object occurs and the grasping position cannot be reached. The same evaluation is performed in depth direction to confirm a sufficient depth for the gripper. When all criteria are fulfilled, the position is verified and represents a grasping position.

3.2. End Tool Alignment

If the performed grasping point detection finds no solution, the end tool performs a rotation into a calculated plane to detect the object from another angle and perform the grasping point detection. To identify the reorientation and detect neighbor surfaces, we use eight triangles (cf. Fig. 4) rotated around a point in the z_{min} to cluster the surface points. From the edge points of the triangles the span vectors of each plane are calculated and afterwards the normal vector and the equation of each plane are determined. Before these calculations can be performed, the units of each axis must be set identical. Initially, the x and y-values are coordinates and the z (depth)-direction is given in absolute values, so the z-values are converted into the dimension of the x,y-grid. For the transformation, the

absolute distance $d_p$between two points is calculated with the following equation:

$$d_p = \frac{z \cdot \tan\frac{\theta}{2}}{res_x}, \quad (2)$$

where z denotes the distance between the camera and the edge of the object, θ denotes the opening angle in the x-direction and res_x the number of pixels after downsampling the original data. Next, the z-values can be converted through a division with d_p. After converting the z-values and deriving the equations of the neighbor planes, the idea of a RANSAC-algorithm is used to identify the optimal surface for the next grasping point evaluation. The basic idea of the RANSAC-algorithm is to create random models and evaluate if all remaining objects fit into the model [7]. Instead of using randomly created models, we use the equations for the derived planes. The assignment of each point to a plane is done through the calculation of its distance to the plane.

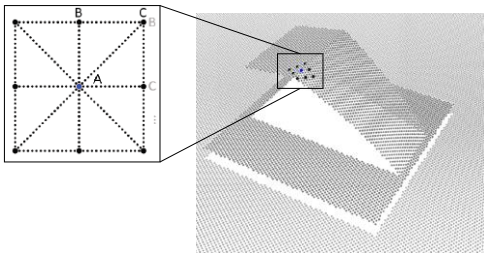

Fig. 4. Positioning of the triangles and its edge points *A, B, C* at z_{min} to detect adjacent surfaces.

The basic idea of the RANSAC-algorithm is to create random models and evaluate if all remaining objects fit into the model [7]. Instead of using randomly created models, we use the equations for the derived planes. The assignment of each point to a plane is done through the calculation of its distance to the plane. Only clustering the points in a plane can lead to false assignments (cf. Fig. 5) because points of other objects or the ground from the point cloud can be assigned to the plane.

Therefore, another restriction was implemented so only points close to the center (point A in Fig. 5) are considered. Performed tests with different distances showed the best results for 15 pixels, so this value is used as the default value. The classification of the points to a plane P_i, where i = 1,…,8 denotes the number of planes can be summarized with the following formulation:

$$P_i = \{n_j | d < 0.3 \wedge A_x - n_{j,x} < 15 \wedge A_y - n_{j,y} < 15 \wedge A_z - n_{j,z} > 15\}, \quad (3)$$

where d denotes the distance between the point n_j and plane P_i and A the centroid of the auxiliary object. The resulting clusters are ranked through the number of its point entries and from this set, the first cluster is selected for the reorientation. While at this stage it is not known which of the selected clusters offers the best option to find a successful grasp, selecting the plane with the most entries usually led to the best results in the considered test cases. After choosing the cluster, the tip of the robot is oriented in negative direction of the normal vector of the selected plane. Thus, both are aligned parallel and the grasping point detection (cf. Section 3.1) can be performed again.

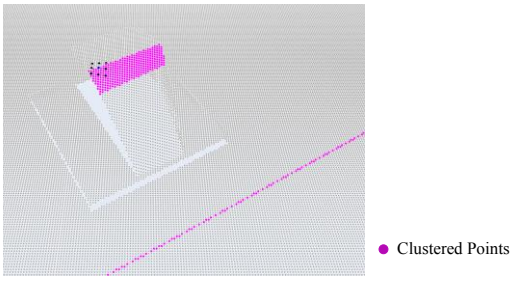

Fig. 5. Partly incorrect clustered points of the floor for a specific plane due to a selection without further restrictions.

4. Evaluation

In order to test our grasping point detection and reorientation algorithm we use the robotic simulation CoppeliaSim (v. 3.6.2) [19]. As Hardware models already integrated models of the UR5e from Universal Robots, a standard parallel jaw gripper and a Microsoft Kinect as the vision sensor are used. The initial data from the vision Sensor is downsampled to reduce the number of pixels and therefore in turn reduce the computation time. For our initial tests, we use a total of 12 objects (cf. Fig. 6). Six objects (*a-f*) are self-created and represent basic geometries. The remaining objects (*g-l*) were selected from a Fraunhofer IPA dataset [18] which was originally adapted from the Sileane dataset [20]. In contrast to the basic geometries of the self-created objects, objects *g-l* are a mixture of industrial and everyday objects with concave and curved geometries. From the latter six objects some had to be scaled to even fit into the dimension of our parallel jaw gripper.

In our test setup all objects are placed on a table in front of the robot and for all objects a pick & place task is performed. To evaluate the performance in different production scenarios we performed the tests with varying distances between the objects (cf. Fig. 7).

In Setup 1, the objects are placed individually on the pickup table. In Setup 2 all objects are placed next to each other with a distance of a few centimeters between each object. The last Setup resembles a bin picking scenario, so the objects are positioned close to each other within minimal distance and sometimes even have overlapping. Objects *a-f* were evaluated in setup 1,2 and 3 and objects *g-l* were only evaluated in setup1. For each setup, we performed 10 tests and positioned the objects randomly to evaluate the procedure with different orientations.

While our setup (cf. Section 3.1 and Section 3.2) offers the possibility to perform the grasping point determination and reorientation in an infinite loop until a valid grasping point is detected, we restricted the algorithm to one reorientation in our evaluation settings.

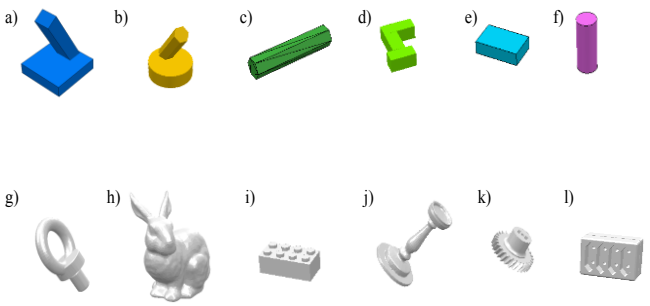

Fig. 6. Used test objects for our evaluation. Objects *a-f* represent self-created objects and objects *g-l* are selected from the data set in [18].

Fig. 7. Different setups used for the evaluation. Setup1: Detached placement of a single object. Setup2: Groups of multiple objects with a few cm between each object. Setup3: Dense cluttered scenes with multiple overlapping objects.

4.1. Results of the Grasping Point Evaluation

As common with detailed grasping simulations with force interaction, the contact can sometimes be unstable and lead to a movement or slipping of the object during the handling task. Therefore, we classified a successful grasp when the object was successfully grasped and lifted from its initial position. Also, for objects *g-l* the grasps had to be manually classified from the calculated grasping positions due to special simulation features. The import of STL data and the assignment of detailed physical properties can be difficult for complex geometries in V-Rep. So, the only physical property which was activated for these objects was the "rendering" option which made the objects visible for the vision sensor, while preserving the original shape. The results of our algorithm for objects *a-f* in setup 1,2 and 3 are displayed in table Table 1:

Table 1. Grasp success rates in setup 1,2 and 3 for objects *a-f*.

	Object					
	a)	b)	c)	d)	e)	f)
Success Setup 1	90 %	60 %	100 %	90 %	100 %	100 %
Success Setup 2	90 %	70 %	70 %	80 %	100 %	100 %
Success Setup 3	60 %	50 %	70 %	50 %	50 %	70 %

In setup 1, where all objects are freestanding without any obstacles, the success rate of objects *a, c, d, e, f* was at or above 90 %. Object *c* could be grasped with a success rate of 60 %. In setup 2 the success rate of objects *a, e, f* was unchanged in comparison to setup 1. Object *b* and *c* had a success rate of 70 % and object d had a success rate of 80 %. The total success rate for setup 2 is 85 %. During the 10 runs, in 3 cases all objects where successfully grasped, 5 times all but one object was successfully grasped and 3 times for two objects a failed grasp occurred. In setup 3, the success rate decreased for all objects but was still at or above 50 percent for all objects. Objects *b, d, e* had a success rate of 50 percent, while object *a* had a success rate of 60% and objects *c* and f a success rate of 70%. The total success rate of setup 3 for objects *a-f* was ~58 % while in all runs 2 or more failures occurred.

As we had already seen the performance development for different production scenarios at objects *a-f* and expected a similar trend for other objects, we only evaluated objects *g-l* in setup 1 (cf. Table 2).

Table 2. Evaluation of the grasp success for objects *g-l* in setup 1.

	Object					
	g)	h)	i)	j)	k)	l)
Success Setup 1	60 %	30 %	60 %	70 %	50 %	40 %

The objects from the Sileane dataset (*g-l*) all had significantly lower success rates compared to the self-created objects. Here, most of the objects had a success rate of around 50 % with a maximum of 70 % for object *j* and minimum of 30 % for object *h*. The total success rate for objects *g-l* was ~52 %.

4.2. Discussion

While the results showed the succession of our grasping point determination and reorientation, a more detailed discussion is needed in some aspects. Comparing the two data sets, a clear performance difference is noticeable (90 % vs 52 %), which leads to the conclusion that our setup performs significantly better for basic shaped geometries. As basic geometries often have unique defined edges and plane surfaces, the clustering of the points is more accurate and leads to better results compared to complex shapes. However, the gap between the two data sets was exaggerated through several reasons. The utilized gripper had a very small opening and is better suited for smaller objects and even though we scaled the objects *g-l* to fit to the parameters of our gripper, a greater grasping space of the gripper would lead to a higher success rate. The results show a good performance with high success rates for basic geometries (objects *a-g*) and if the objects are positioned individually. When executed in a bin picking scenario the success rate of our methodology decreases noticeably from 85 % to ca. 58 %. In this scenario the objects often overlap with each other and therefore the positioning and depth evaluation of the auxiliary object is influenced.

As mentioned in Section 4, we restricted the algorithm to perform only one reorientation to enhance the runtime of our algorithm. If this threshold is set at a higher value a more elaborate analysis of the objects is possible and therefore with great likelihood higher success rates.

Further analyzing the unsuccessful grasps shows that the positioning of the auxiliary object at the edge and the contour itself caused the failures sometimes. In our setup, both steps are based on relatively simple analytical models and therefore in some cases do not detect the whole contour of the surface. Combined with a single positioning of the auxiliary at the middle of the edge this led to some failures (cf. Fig. 8). A possible way of improving success rates in this case would be a more frequent positioning of the double T-beam to enhance the reviewed area.

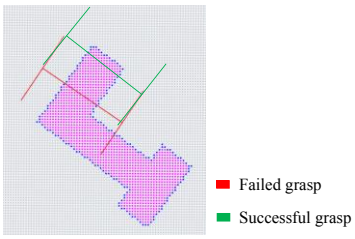

Fig. 8. False and correct alignment of the double T-beam at an extracted surface.

The performance decrease in a bin picking scenario can also be explained with our grasping determination framework. In cluttered scenes, the clustering of surface points is extremely difficult and therefore failures during the contour extraction occur.

Processing times ranged from a few seconds to a minute. The time increases significantly if a reorientation is necessary from the initial position, as the clustering and reorientation of the point clouds often accounts for more than 50% of the overall processing time. While a comparison with other strategies is difficult due to differences in hardware and computing power, computation times in [8] where comparable to ours only considering the grasping point determination. In comparison, the CNN based Dex-Net architecture [3] takes only a few seconds to calculate the grasping positions but mainly uses a vacuum gripper.

5. Summary and Conclusion

In this paper, we developed a RANSAC-based algorithm to perform efficient reorientations and determine grasping points for unknown objects with only partial object information available. The grasping point determination is performed in our setup with a double T-beam as an auxiliary object. The validity of our reorientation calculation and grasping point determination was shown through different setups where we reached a success rate of 90 % for basic geometries in a detached setting. For slightly cluttered scenes with multiple objects the performance was comparable (85 %). Tests for dense cluttered scenes (bin picking scenarios) and complex object geometries showed a significant performance decrease, 58 % for the former and 52 % for the latter. It can be concluded that the existing setup of our algorithm is better suited for pick and place applications with basic objects in slightly cluttered scenes. To further enhance the performance in all settings a few points will be optimized in the future.

Besides performing more cycles between the reorientation and grasping point determination, we plan to optimize the contour detection and generate the full contour through an extension of the original ruleset. Also, a more frequent division and positioning of the auxiliary object will be implemented which will enlarge the solution set further and enhance the success rate. Finally, we want to evaluate our setup on a real robot in a physical production setup.

References

[1] K. Harada *et al.,* „Grasp planning for parallel grippers with flexibility on its grasping surface“ in *2011 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Karon Beach, Thailand, 2011, S. 1540–1546, doi: 10.1109/ROBIO.2011.6181508.

[2] C. Papazov, S. Haddadin, S. Parusel, K. Krieger und D. Burschka, „Rigid 3D geometry matching for grasping of known objects in cluttered scenes“, *The International Journal of Robotics Research*, Jg. 31, Nr. 4, S. 538–553, 2012, doi: 10.1177/0278364911436019.

[3] J. Mahler *et al.,* „Learning ambidextrous robot grasping policies“ (eng), *Science robotics*, Jg. 4, Nr. 26, 2019, doi: 10.1126/scirobotics.aau4984.

[4] P. Schmidt, N. Vahrenkamp, M. Wachter und T. Asfour, „Grasping of Unknown Objects Using Deep Convolutional Neural Networks Based on Depth Images“ in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, QLD, 2018, S. 6831–6838, doi: 10.1109/ICRA.2018.8463204.

[5] Z. Yin und Y. Li, „Overview of robotic grasp detection from 2D to 3D“, *Cognitive Robotics*, Jg. 2, S. 73–82, 2022, doi: 10.1016/j.cogr.2022.03.002.

[6] J. P. C. de Souza, L. F. Rocha, P. M. Oliveira, A. P. Moreira und J. Boaventura-Cunha, „Robotic grasping: from wrench space heuristics to deep learning policies“, *Robotics and Computer-Integrated Manufacturing*, Jg. 71, S. 102176, 2021, doi: 10.1016/j.rcim.2021.102176.

[7] M. A. Fischler und R. C. Bolles, „Random sample consensus“, *Commun. ACM*, Jg. 24, Nr. 6, S. 381–395, 1981, doi: 10.1145/358669.358692.

[8] K. Kleeberger, F. Roth, R. Bormann und M. F. Huber, „Automatic Grasp Pose Generation for Parallel Jaw Grippers“, 23. Apr. 2021. [Online]. Verfügbar unter: http://arxiv.org/pdf/2104.11660v1.

[9] Y. Fang, X. Wang, Z. Sun, B. Su und J. Xue, „Method to improve the accuracy of depth images based on differential entropy“, *Opt. Eng.*, Jg. 60, Nr. 03, 2021, doi: 10.1117/1.OE.60.3.033105.

[10] A. Herzog *et al.,* „Learning of grasp selection based on shape-templates“, *Auton Robot*, Jg. 36, 1-2, S. 51–65, 2014, doi: 10.1007/s10514-013-9366-8.

[11] M. Nieuwenhuisen, J. Stueckler, A. Berner, R. Klein und S. Behnke, „Shape-Primitive Based Object Recognition and Grasping“, *Proceedings of ROBOTIK*, Jg. 2012, 2012. [Online]. Verfügbar unter: http://ais.uni-bonn.de/papers/robotik2012_nieuwenhuisen_grasping.pdf

[12] F. T. Pokorny, J. A. Stork und D. Kragic, „Grasping objects with holes: A topological approach“ in *2013 IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany, 06.05.2013 - 10.05.2013, S. 1100–1107, doi: 10.1109/ICRA.2013.6630710.

[13] E. Klingbeil, D. Rao, B. Carpenter, V. Ganapathi, A. Y. Ng und O. Khatib, „Grasping with application to an autonomous checkout robot“ in *2011 IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, 09.05.2011 - 13.05.2011, S. 2837–2844, doi: 10.1109/ICRA.2011.5980287.

[14] D. Schiebener, J. Schill und T. Asfour, „Discovery, segmentation and reactive grasping of unknown objects“ in *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, Osaka, Japan, 2012, S. 71–77, doi: 10.1109/HUMANOIDS.2012.6651501.

[15] J. Varley, C. DeChant, A. Richardson, J. Ruales und P. Allen, „Shape completion enabled robotic grasping“ in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, BC, 24.09.2017 - 28.09.2017, S. 2442–2447, doi: 10.1109/IROS.2017.8206060.

[16] A. T. Miller und P. K. Allen, „GraspIt!“, *IEEE Robot. Automat. Mag.*, Jg. 11, Nr. 4, S. 110–122, 2004, doi: 10.1109/MRA.2004.1371616.

[17] X. Huang *et al.,* „Real-time grasping strategies using event camera“, *J Intell Manuf*, Jg. 33, Nr. 2, S. 593–615, 2022, doi: 10.1007/s10845-021-01887-9.

[18] K. Kleeberger, C. Landgraf und M. F. Huber, „Large-scale 6D Object Pose Estimation Dataset for Industrial Bin-Picking“, 6. Dez. 2019. [Online]. Verfügbar unter: http://arxiv.org/pdf/1912.12125v1.

[19] E. Rohmer, S. P. N. Singh und M. Freese, „V-REP: A versatile and scalable robot simulation framework“ in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2013)*, Tokyo, 03.11.2013 - 07.11.2013, S. 1321–1326, doi: 10.1109/IROS.2013.6696520.

[20] R. Bregier, F. Devernay, L. Leyrit und J. L. Crowley, „Symmetry Aware Evaluation of 3D Object Detection and Pose Estimation in Scenes of Many Parts in Bulk“ in *2017 IEEE International Conference on Computer Vision Workshop (ICCVW)*, Venice, 22.10.2017 - 29.10.2017, S. 2209–2218, doi: 10.1109/ICCVW.2017.258.