Translucent Object Grasping Using Robot Vision

Krishna Kodur*, Manizheh Zand†, Matthew Tognotti‡, Maria Kyrarini§
Dept. of Electrical & Computer Engineering
Santa Clara University
Santa Clara, CA, USA - 95053
kkodur@scu.edu*, mzand@scu.edu†, mtognotti@scu.edu‡, mkyrarini@scu.edu§

2024 10th International Conference on Automation, Robotics and Applications (ICARA) | 979-8-3503-9424-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICARA60736.2024.10552988

***Abstract*—Grasping translucent objects, such as open containers, poses a significant challenge when using RGB and Depth (RGBD) cameras, primarily due to the presence of cavities in their depth values. The need for effectively grasping translucent containers is especially important in kitchen environments, where easy visibility of the contents inside is essential, particularly for individuals with dementia. This paper addresses this challenge by introducing a novel method that combines an analytical approach with an object detection algorithm such as You Only Look Once (YOLO) to improve grasping performance. Traditional approaches often rely on depth-filling deep neural network models to mitigate the issues caused by these cavities. Although various deep learning methods have been developed for this purpose, they typically entail extensive data collection efforts for fine-tuning their models to work for the objects of interest. In contrast, the approach presented in this paper leverages an analytical method that is particularly well-suited for objects with simple geometries, effectively eliminating the necessity for extensive data collection to predict grasp points and fill cavities. The experimental results demonstrate the effectiveness of this novel approach, with an average grasping accuracy of 94.55% achieved on translucent open containers, establishing it as a viable and practical alternative to traditional deep learning-based methods. The source code is available at Link[1] and the dataset for training the object detection algorithm YOLO in this paper is available at Link[2].**

***Index Terms*—Computer Vision, Robot Grasping, Translucent Objects**

I. Introduction

The escalating prevalence of Alzheimer's dementia among the elderly, predicted to nearly triple by 2050 [1], presents a pressing concern, accentuating the challenges faced by this demographic in their daily lives. Addressing these challenges, more than 70% of homeowners have been making substantial modifications to their residences, with a predominant focus on integrating assistive technology to aid in day-to-day tasks, as reported by the National Aging in Place Council and National Association of Home Builders [2]. Despite these efforts, the execution of fundamental Activities of Daily Living (ADL) such as cooking remains a formidable task for individuals dealing with dementia. According to Wherton et al., [3], visibility issues further compound this difficulty, highlighting the specific challenges faced by individuals with dementia when navigating tasks in the kitchen environment.

Recognizing these challenges and the need for innovative solutions, practical recommendations have been made for inclusive kitchen design. One such recommendation, supported by the Social Care Institute for Excellence, involves using transparent or translucent containers [4]. This seemingly simple yet highly effective solution holds significant potential in enhancing the accessibility of objects for individuals with dementia, thereby facilitating their active participation in daily activities. In further exploring avenues to enhance accessibility and independence in kitchen environments, integrating robotic systems emerges as a promising solution. These advanced technologies can assist individuals by performing tasks such as opening cupboards, retrieving ingredients and utensils, and operating kitchen appliances, similar to how Sugiura et al. [5] conceptualize cooking as a collaborative scenario between humans and robots. To perform all the aforementioned tasks effectively, an assistive robot must accurately grasp objects.

One of the commonly used methods in grasping is the use of computer vision. The necessity of developing an accurate vision system capable of predicting and grasping translucent objects is even more challenging. Existing approaches in the realm of deep learning, primarily relying on RGB and Depth (RGBD) cameras, have demonstrated success in grasping opaque objects, as evidenced by notable studies [6], [7]. However, these methods face a significant challenge when it comes to translucent objects due to the presence of cavities in their depth values. These voids of depth information render the conventional deep learning approaches ineffective for grasping translucent objects [8].

Addressing this challenge requires rectifying the distorted depth values inherent to translucent objects. Several deep learning methods have been devised to fill these cavities accurately, as highlighted in studies such as [8]–[10]. A notable drawback of these methods is the requirement for extensive data collection and training. However, when the object's geometry is simple, e.g., open containers, collecting data for training would be superfluous. To circumvent the challenges associated with data collection and training, analytical methods offer a viable alternative, particularly when the object's geometry is simple. Analytical techniques can be employed to identify suitable grasp points on the object, thereby reducing the need for extensive data-driven approaches.

The contributions of this paper are as follows:

1) a novel analytical method that estimates the robotic grasping point for translucent open containers
2) a real-world implementation of the proposed method for evaluation with a 7-degrees of freedom robotic arm.

[1] https://github.com/HMI2-Research-Group/Analytical_BestGrab
[2] https://github.com/HMI2-Research-Group/Kitchen-YOLO-Dataset

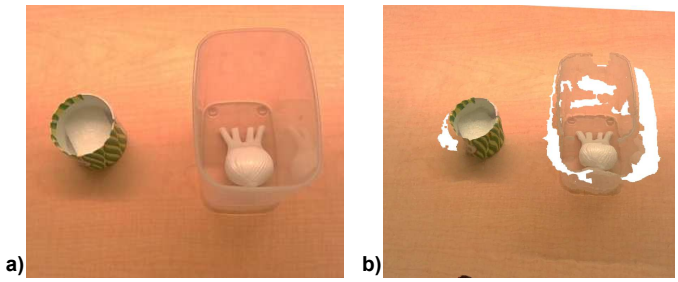

Fig. 1. a) The RGB image of an opaque container on the left and a translucent container on the right. b) The corresponding point cloud for the objects.

The rest of the paper is organized as follows: Section II explains the current state-of-the-art methods for grasping translucent objects, Section III presents the results of the baseline methods, and Section IV elaborates the proposed system architecture to grasp open container translucent objects. The results and comparison with baselines are discussed in Section V, followed by the conclusion in Section VI.

II. Related Work

Robotic manipulation refers to robots interacting with objects and their environments through physical interactions such as grasping. In a home environment, robots must adapt to challenging situations such as occlusion [11]. However, grasping is even more challenging when the objects are translucent. The realm of robotic grasping has been a focal point of extensive research efforts. A distinctive challenge arises when dealing with translucent objects, a complexity not encountered with their opaque counterparts. Unlike opaque objects, translucent objects pose a formidable hurdle in generating accurate depth estimates, often leading to distorted results and significant lacunae, as demonstrated in Figure 1. This added complexity makes adopting robots at home difficult, especially if the robot is designed to assist people with dementia who may require translucent objects for easier visibility. The current methods use 3D data to grasp the objects, including point clouds or images coupled with depth information (commonly known as a depth frame), so that grasp points and grasp angles can be detected. Grasp points are points at which the robot can grasp the object, and the grasp angle is the angle at which the gripper can approach the object. However, as there are cavities in the depth frame of translucent objects, the current methods cannot be applied directly. This inherent difficulty in grasping translucent objects remains an ongoing research challenge, as emphasized by Sun et al. [12].

To address this challenge, researchers have turned to deep learning models and proposed datasets containing translucent objects. These datasets can be categorized mainly into three types based on the generation method: (1) synthetic Datasets, (2) real datasets, and (3) a combination of real and synthetic Datasets. In synthetic datasets, translucent objects are simulated in realistic environment simulators, such as Blender. [13]. Li et al. [9] proposed enhancing the naturalness of the simulations in Blender to make better synthetic datasets. The reason for this enhancement is because of a phenomenon called the Sim-to-Real gap [14]. This gap refers to the disparity between the performance of deep learning models in simulated environments and their performance in the real world. Factors such as lighting conditions, sensor noise, and unmodeled dynamics occurring naturally can significantly impact the performance of algorithms. Li et al. proposed to use Blender with enhanced lighting simulations along with exact camera intrinsics for the Intel RealSense D435i to simulate realistic scenes. The authors also proposed using a Gaussian mask on the depth image. The peaks in the mask can act as a guiding factor for accurately finding the best grasp points. The authors concluded that using enhanced simulations and a Gaussian mask improved the grasp rate by 36.7% compared to direct grasping without using Gaussian mask annotation. Fang et al. [8] proposed using a real dataset called the TransCG to combat the Sim-to-Real gap. However, collecting a real dataset is time-consuming, especially annotating the grasp points needed for the Convolutional Neural Network (CNN) to train. To get around this issue, the authors proposed the use of an object tracking system that can detect objects mounted with an IR marker in real-time. The generated dataset is used to train Depth Filler Net (DFNet) to correct the depth estimates. In real-world testing, the authors established a perfect grasping rate on eight objects (six novel and two from the training). ClearGrasp by Sajjan et al. [10] employed the third method of combining real and synthetic data to train their CNN based network to correct the depth cavities. The authors concluded that using ClearGrasp increased the accuracy of the depth frame estimates, thus improving the grasp accuracy compared to only using the depth frame for detecting the grasping point.

The generation of real-world and synthetic datasets takes time and effort. Although deep learning methods can correct the depth frame of objects, fine-tuning might still be needed for some objects depending on the object's shape, as shown in the baseline comparison in Section III. Instead of fine-tuning to correct the depth frame, analytical methods can be used if the geometry of objects, e.g., open containers, is known. This novel analytical method can then be used for translucent open containers and is especially useful for robots to grasp translucent objects to assist people with dementia.

III. Baseline Comparison

It is imperative to test the baseline methods before proposing a new analytical approach for grasping open containers. Therefore, DFNet [8] is used to correct the depth values, and GraspNet [7] is used to generate the grasp location and orientation. Both models are used without fine-tuning. The combined DFNet+GraspNet is used for detecting the grasp location and orientation. To test the average accuracy of correct grasp locations and orientation detections on translucent objects, the model's output in each of the three translucent objects shown in Figure 2 is tested ten times, and the average accuracy is calculated to be 16.67%. However, in the case of testing on opaque objects, shown in Figure 2, there is no need

for depth correction using DFNet. Only GraspNet is used to detect the grasp location and orientation of the opaque objects with an average accuracy of 90%.

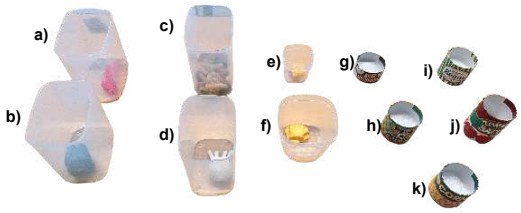

Fig. 2. Different opaque and translucent objects for testing: Translucent containers with a) Chilli b) bell Pepper c) Pasta d) Garlic e) Cheese f) Butter; Opaque containers with g) Mushrooms h) Peas and Carrots i) Green beans j) Tomato Sauce k) Corn.

Figure 3 shows the DFNet+GraspNet output grasp location and orientation on one of the translucent container test objects. This indicates that additional training using the test translucent container is required before DFNet can appropriately correct the depth values. To address this, it is essential to propose an analytical method for the detection of grasp points and orientations, serving as a substitute for deep neural networks like DFNet.

Fig. 3. Output of DFNet by Fang et al [8] on one of the translucent container test objects.

IV. Proposed Architecture

Aims and Assumptions: The aim is to develop a hybrid approach consisting of deep learning models and analytical methods for grasping open and translucent containers. The objects are detected by commonly used object detection algorithms, and the analytical methods are designed under the assumption that the gripper will approach the object from a top-down perspective.

System Architecture: The proposed architecture is depicted in Figure 4, and succinctly described in Algorithm 1. The process begins with the acquisition of RGB color and camera intrinsics using an RGBD camera, specifically the Intel RealSense D405 RGB-Depth camera (D405) model. Subsequently, object detection is performed using YOLO [15],

Algorithm 1: Analytical Grasp Point Detector

input : RGB Image, Camera Intrinsics, n
output: Best Grasp Point, Orientation Angle

BBox = YOLO(RGB Image) ;
`/* Assuming YOLO has given the Bounding Box of the object of interest only */`
SPC = BBox to PointCloud(BBox, PixeltoPointProjection) ;
TPC = $\{(x,y,z)|^{cam}_{world}T(x_s,y_s,z_s)\forall(x_s,y_s,z_s) \in SPC\}$;
MH = $\arg\max_z \forall(x,y,z) \in TPC$;
PGP = $\{(x,y,z)|(x,y,z) \in TPC \wedge z \geq MH - \delta\}$;
BGP = $\arg\max_{(x,y,z)\in PGP}\{||X - \frac{X_{tl}+X_{br}}{2}|| \; |(X,Y) \in PoPiPr(^{world}_{cam}T(x,y,z))\}$;
Let $(x_0,y_0,z_0) = GraspPoint$;
α = $\{(x,y,z)|(x,y,z) \in PGP \wedge ||(x,y,z) - (x_0,y_0,z_0)|| > 0\}$;
NGP = $\arg\max_{I\subset\alpha:|I|=n} \alpha$ = $\{(x_1,y_1,z_1)\dots(x_n,y_n,z_n)\}$;
Let $x = \begin{bmatrix} x_0 & x_1 & \dots & x_n \end{bmatrix}^T$;
Let $y = \begin{bmatrix} y_0 & y_1 & \dots & y_n \end{bmatrix}^T$;
A = $\begin{bmatrix} x^T x & x & 1 \end{bmatrix}$;
$\begin{pmatrix} a \\ b \\ c \end{pmatrix} = (A^T A)^{-1} A^T y$;
$\theta_{\text{orientation}} = atan2(\frac{-1}{2ax_0+b})$;
return BGP, $\theta_{\text{orientation}}$

a model for identifying bounding boxes around objects of interest. YOLO is finetuned using a curated dataset specifically to identify both translucent and opaque objects, which is made accessible **here**[2]. The dataset comprises 210 images. The image count for each class is elucidated in Table I. Notably,

TABLE I
Table depicting the distribution of image counts across various classes within the dataset, comprising 210 images.

Class	Image Count by Class
Cheese	18
Peas and Carrots	25
Butter	32
Green Beans	34
Corn	48
Mushrooms	52
Garlic	56
Bell Pepper	57
Tomato Sauce	58
Chilli	60
Pasta	62

the dataset includes a range of translucent and opaque objects, as can be observed in Figure 2. This variety ensures the

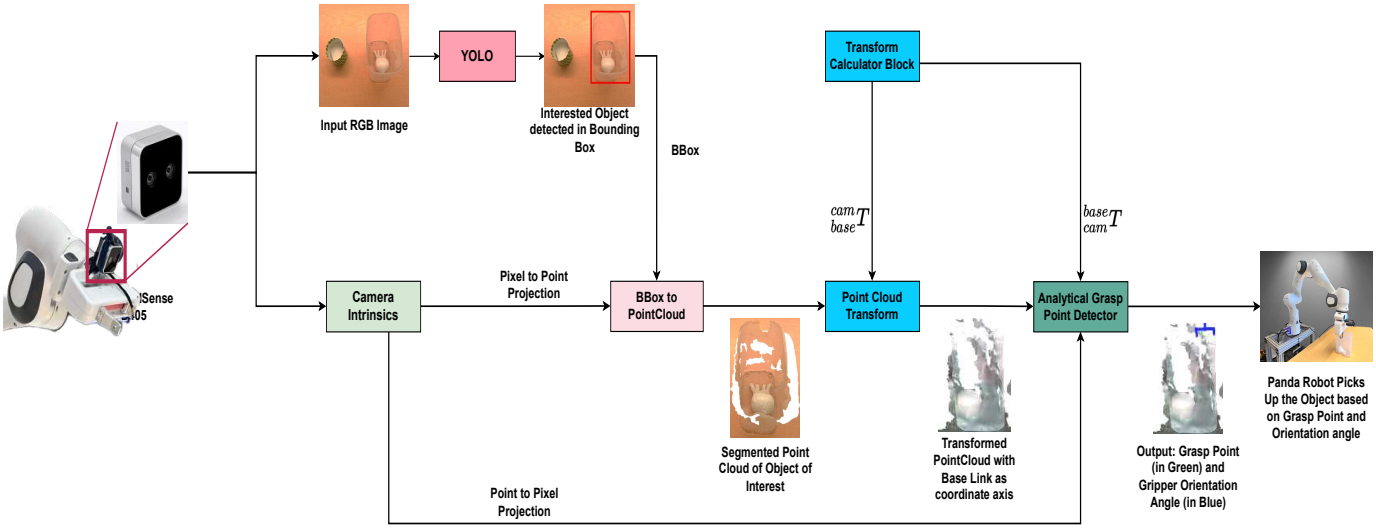

Fig. 4. The proposed pipeline for detecting the grasp point and grasp orientation angle from the camera inputs of RGB Image and Camera Intrinsics.

model's proficiency in detecting objects with different material properties.

Let the bounding box (BBox) output from the YOLO model be represented as

$$BBox = \{(X, Y) | X_{tl} \leq X \leq X_{br}, Y_{tl} \leq Y \leq Y_{br}\} \quad (1)$$

where (X_{tl}, Y_{tl}) are the top left pixel coordinates and (X_{br}, Y_{br}) are the bottom right coordinates of the bounding box.

Upon detection of the bounding box, the "BBox to PointCloud" block outputs the corresponding point cloud enclosed within the bounding box (line 2 in Algorithm 1). This function takes two inputs. One *BBox* and other Pixel to Point Projection ($PiPoPr : R^2 \rightarrow R^3$), which projects the pixel (X, Y) into a 3D point (x, y, z) in the camera coordinate frame.

The "Point Cloud Transform" block in the Figure 4 (line 3 in Algorithm 1) transforms the points in the point cloud generated by the "BBox to PointCloud" to a world frame of reference that is stationary. The world reference frame's Z-axis is aligned perpendicularly to the ground plane and points upwards. By applying this transformation, all the points are reoriented to align with the world reference frame. The transform is provided by "Transform Calculator Block", whose functionality is provided by the tf package [16], which transforms the points from the camera coordinate frame to the world frame and vice versa. Let the transformation from the camera coordinate frame to the world frame be defined as ${}^{cam}_{world}T : R^3 \rightarrow R^3$, the transformation from world frame to the camera coordinate frame ${}^{world}_{cam}T : R^3 \rightarrow R^3$ and let the transformed point cloud be defined as TPC. TPC is the output of "Point Cloud Transform" block. An interesting property of TPC is that the elevated Z coordinates in TPC correspond to the top edge points of the object of interest. The top edge points are particularly interesting because those are the potential points where the object can be grasped.

As the RGBD cameras cannot estimate the depth of translucent objects correctly, the best estimate of a viable grasp point is the point with the highest Z coordinate (line 4 in Algorithm 1). But getting as many top-edge points as possible is also essential. Therefore, all the points whose Z coordinate is greater than the highest Z coordinate minus a small offset δ, e.g., 1mm, can be sampled as potential grasp points PGP (line 5 in Algorithm 1).

From the RGB image, the best grasp point should lie as farthest as possible from the center line of the bounding box because that is where at least one of the faces or edges is present. The center line can be defined as $X = \frac{X_{tl}+X_{br}}{2}$. To measure the distance from this line, the 3D points must be projected back to the pixel plane using Point to Pixel Projection $PoPiPr : R^3 \rightarrow R^2$, and the distance can be calculated. The point with the highest distance from the line is chosen as the best grasp point BGP, which can also be referred to as (x_0, y_0, z_0) (line 6 in Algorithm 1).

To determine the approach angle of the gripper in a top-down manner (line 7-line 14 in Algorithm 1), a set of n nearest points to the optimal grasp point is chosen as NGP (line 9 in Algorithm 1). After trial and error, $n \geq 4$ was found to be the best. As the focus is grasping the object top-down, the grasp point and the n nearest points are projected onto the XY plane as the Z axis is not needed. A parabolic curve can be fitted by least squares approximation through those selected points. The normal to the parabola passing through the best grasp point can be considered the best gripper orientation. Let the best-fit parabola be denoted as $y = ax^2 + bx + c$. Finally, the gripper's orientation angle can be found by atan2(-1/(2ax_0 + b)). The source code of Algorithm 1 can be found **here**[1]. The next section discusses the results of the proposed method.

TABLE II
AVERAGE GRASPING ACCURACY OF EACH TEST OBJECT. TEN TRIALS ARE CONDUCTED TO COMPUTE THE AVERAGE.

Object Name	Grasp Accuracy (in %)
Tomato Sauce (Opaque)	90
Peas and Carrots (Opaque)	100
Green Beans (Opaque)	90
Mushrooms (Opaque)	100
Corn (Opaque)	90
Cheese (Translucent)	90
Butter (Translucent)	100
Garlic (Translucent)	100
Pasta (Translucent)	90
Chilli (Translucent)	100
Bell Pepper (Translucent)	90
Overall Accuracy	**94.55**
Translucent Objects Accuracy	**95**
Opaque Objects Accuracy	**94**

V. RESULTS AND DISCUSSION

To evaluate the proposed architecture in Section IV, the grasping accuracy is assessed with ten trials for 11 distinct objects, shown in Figure 2. The average accuracy of ten trials for each object is presented in Table II; the average grasping accuracy is 94.55%.

The authors of DFNet [8] report the average accuracy during real-world experiments is 80.4%. Therefore, analytical approaches provide a viable option in case the object's open and flat-topped geometry is known and performs at par with deep neural networks without training for depth correction. However, both analytical and deep learning methods have their own advantages and drawbacks. The analytical method demands less data because the object's geometry is known beforehand. Conversely, deep learning requires more data but can generalize across objects well, enabling the grasping of diverse objects. Therefore, it is important to determine which method to use depending on the types of objects used and whether data collection is desirable.

Hence, the choice of the method should be guided by the nature of the objects involved and the desirability of data collection.

VI. CONCLUSION

This paper introduced a novel method to grasp translucent open containers using an analytical method paired with an object detection algorithm. Previous approaches have succeeded in grasping opaque objects, but translucent objects are challenging to grasp due to cavities caused by distorted depth data. Several deep-learning methods have been devised to fill these cavities accurately, but they require extensive data collection. This proposed architecture enables a robot to accurately grasp translucent objects with simple geometries, such as an open container, with less data collection required for training when compared to previous methods. The results show an average grasping accuracy of 94.55% on open containers, which is comparable to deep neural network approaches.

ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation under Grant 2226165. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] "2017 alzheimer's disease facts and figures," *Alzheimer's & Dementia*, vol. 13, pp. 325–373, 4 2017. [Online]. Available: https://onlinelibrary.wiley.com/doi/full/10.1016/j.jalz.2017.02.001https://onlinelibrary.wiley.com/doi/abs/10.1016/j.jalz.2017.02.001https://alz-journals.onlinelibrary.wiley.com/doi/10.1016/j.jalz.2017.02.001

[2] "Aging in place survey report oct 2015." [Online]. Available: https://www.homeadvisor.com/r/wp-content/uploads/2015/10/HomeAdvisor-Aging-in-Place.pdf

[3] J. P. Wherton and A. F. Monk, "Technological opportunities for supporting people with dementia who are living at home," *International Journal of Human-Computer Studies*, vol. 66, pp. 571–586, 8 2008.

[4] "Dementia-friendly environments: Kitchens and dining areas," Oct 2020. [Online]. Available: https://www.scie.org.uk/dementia/supporting-people-with-dementia/dementia-friendly-environments/kitchens.asp

[5] Y. Sugiura, D. Sakamoto, A. Withana, M. Inami, and T. Igarashi, "Cooking with robots: designing a household system working in open environments," in *Proceedings of the SIGCHI conference on human factors in computing systems*, 2010, pp. 2427–2430.

[6] A. Mousavian, C. Eppner, and D. Fox, "6-dof graspnet: Variational grasp generation for object manipulation," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2019-October, pp. 2901–2910, 10 2019.

[7] H. S. Fang, C. Wang, M. Gou, and C. Lu, "Graspnet-1billion: A large-scale benchmark for general object grasping," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 11 441–11 450, 2020.

[8] H. Fang, H. S. Fang, S. Xu, and C. Lu, "Transcg: A large-scale real-world dataset for transparent object depth completion and a grasping baseline," *IEEE Robotics and Automation Letters*, vol. 7, pp. 7383–7390, 7 2022.

[9] S. Li, H. Yu, W. Ding, H. Liu, L. Ye, C. Xia, X. Wang, and X. P. Zhang, "Visual–tactile fusion for transparent object grasping in complex backgrounds," *IEEE Transactions on Robotics*, 10 2023.

[10] S. Sajjan, M. Moore, M. Pan, G. Nagaraja, J. Lee, A. Zeng, and S. Song, "Clear grasp: 3d shape estimation of transparent objects for manipulation," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 3634–3642, 5 2020.

[11] "Why household robot servants are so hard to build," Oct 2022.

[12] Y. Sun, J. Falco, M. A. Roa, and B. Calli, "Research challenges and progress in robotic grasping and manipulation competitions," *IEEE Robotics and Automation Letters*, vol. 7, pp. 874–881, 4 2022.

[13] Blender Online Community, *Blender - a 3D modelling and rendering package*, Blender Foundation, Blender Institute, Amsterdam, 2023. [Online]. Available: http://www.blender.org

[14] T. Stoffregen, C. Scheerlinck, D. Scaramuzza, T. Drummond, N. Barnes, L. Kleeman, and R. Mahony, "Reducing the sim-to-real gap for event cameras," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12372 LNCS, pp. 534–549, 2020. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-58583-9_32

[15] G. Jocher, "ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements," https://github.com/ultralytics/yolov5, Oct. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.4154370

[16] T. Foote, "tf: The transform library," in *Technologies for Practical Robot Applications (TePRA), 2013 IEEE International Conference on*, ser. Open-Source Software workshop, April 2013, pp. 1–6.