Team Yellow on Agile Waste Sorting with Tossing

Biao Wang[1*], Mandan Chao[2*], Jinqi Wei[3*], Zerui Li[4*], Guowei Shi[5*], Haoze Sun[6*], Yulun Zhuang[7*], Zhi Xu[8*]

Abstract— **We investigate how fast the waste sorting speed can be improved by tossing with the robot arm in comparison with placing. Tossing has the potential to increase the physical reachability and picking speed of a robot arm. However, precisely throwing wastes in unstructured settings presents many challenges: from acquiring reliable initial pose of object in manipulator to handling varying object-centre properties (e.g. mass distribution, friction, shape) and aerodynamics. In this work, we investigate the process of grasping and tossing. Through UR5 robot arm, we achieve accurate 2D picking of wastes with orientation information on both stationary and running conveyor belt. For different tossing target, we optimize the trajectory of tossing including the position and velocity properties. The resulting system is able to classify, grasp and throw the waste into corresponding boxes at 521 times tossing per hour (614 times grasping per hour with 85% tossing accuracy), which is 60+ times more than sorting by picking and placing.**

Index Terms— **Robot Arm Tossing, Waste Sorting, Object Detection, 2D Picking**

CONTENTS

*This work was supported by Southern University of Science and Technology and AncoraSpring Inc. for the ME336 Collaborative Robot Learning (me336.ancorasir.com) under the supervision of Prof. Song Chaoyang songcy@ieee.org

[1*]Author Biao Wang: tossing trajectories designing and debugging, PnT and PnP methods realization and comparison, data collection and cleaning, Grippers comparison. [2*]Author Mandan Chao: grasping algorithm designing, grasping process test, data cleaning, presentation slides designing. [3*]Author Jinqi Wei: Object detection and classification, Grasping Points Computation,data cleaning. [4*]Author Zerui Li: data collection and cleaning, experiment environment building, experimental equipment purchasing. [5*]Author Guowei Shi: Design robot's posture and the trajectory of the tossing motion, implement the PNT model to sort the waste, label the data, construct the trash bins separator. [6*]Author Haoze Sun: data collection and cleaning, experiment environment building, calibration precision testing. [7*]Author Yulun Zhuang: grasping algorithm designing, grasping process test and data cleaning. [8*]Author Zhi Xu: Object detection and classification, Grasping Points Computation,data collection and cleaning.

I. INTRODUCTION

In recent years there has been an increasing demand for waste sorting, which can be solved properly by the application of modern robotic technology. Recent research on robot arms presents a growing interest in applying it to realize waste sorting with high accuracy. However, another important factor that determine the feasibility of the sorting robot arm is the general efficiency. In further research, tossing is an excellent means of exploiting dynamics to increase the capabilities of a manipulator. throwing enables a robot arm to rapidly place objects into boxes located outside its maximum kinematic range, which not only reduces the total physical space used by the robot, but also maximizes its picking efficiency.

In our work, we aim to use the robot arm to improve the waste sorting speed with certain sorting accuracy with tossing. The work is proved to achieve higher execution speed compared to those using usual methods. During the work, we train a YOLO-v5 model to classify garbage, the process of grasping and tossing is investigated using 2D and 6D calibration with robot arm control.[1] Then we can control the expected grasping position in both fixed and moving situation of conveyor belt. Finally, we designed and optimized the trajectory of tossing including the position and velocity.

A. Related Work

Robotic system sorting of mixed waste has a growing interest.[2] The system The system combines machine learning technology to categorize recyclables transferred on a conveyor belt into a set of predefined material classes with vacuum gripping to pick recyclable items. The implemented robotic system outperforms human quality-control workers who pick out about 30 to 40 items per minute, on a daily average. In contrast, our robot picks on average 60 items per minute which is similar to the processing rate of commercial systems.

In the previous robot arms for waste sorting, flexible robot arm is tested to enhance performance such as a fast motion in short operation time.[3] The system relies on handcrafting or approximating dynamics based on mechanical analysis, then optimizing control parameters to execute a throw. Other researches attempt to use learning method to optimize the sorting effects.

An end-to-end system was put forward[4], which uses trial and error to learn how to predict control parameters for grasping and throwing from visual observations. The formulation learns grasping and throwing jointly – discovering grasps that enable accurate throws, while learning throws that compensate for the dynamics of arbitrary objects.

B. *Proposed Method and Contributions*

To reduce data collection time for deep learning of robust robotic grasp plans, training from a point clouds is applicable. Using the resulting dataset, a Grasp Quality Convolutional Neural Network model can be built to rapidly detect the labels of waste from depth images.

The model we training is used with YOLOv5. YOLOv5 is a family of compound-scaled object detection models trained on the COCO dataset, and includes simple functionality for Test Time Augmentation model ensembling, hyperparameter evolution, and export to ONNX, CoreML and TFLite.

For tossing, our method is based on the result of theoretical model. Then adjust according to the actual situation.

II. METHOD

The integrated waste sorting system is composed by four parts, namely:(i) data, (ii) detection, (iii) grasping and (iv) tossing parts.

A. *Data*

In this experiment, we need to use the camera to collect the image data of the target object on the experimental platform built for subsequent recognition and capture training.

For data collection and preservation, first of all, the environment of the experimental platform should be installed, the camera should be connected, the transmission band rate should be set, the filling light should be turned on/off, and the objects to be collected should be prepared.Open the conveyor belt in the conveyor belt controller, set the knob 0.2m/s;Turn on the light source in the fill light controller.

When collecting and saving a single photo, there is no need to start the conveyor belt, and the given code is directly used to collect and save the image.When continuous photo collection is carried out, the conveyor belt is started and the code is run to realize the collection of dynamic objects.

After collecting and saving the data, we should carry out the corresponding labeling work on the required data part, as the truth value of subsequent algorithm training.

Labelimg and Labelme can be used for labeling. Labelimg is an ortho-rectangular box labeling format, and Labelme can label polygon regions.In this experiment, we believe that it is not necessary to carry out very accurate shape recognition of the target object, but the pose is highly required, so we choose to use the labelImg method for labeling.Considering that we need to obtain as much accurate data as possible within a limited time, the image information is roughly processed by automatic annotation first, and then detected by manual annotation, so as to make the collected data credible.

The result of automatic labeling is directly related to the workload of manual labeling, so to improve the accuracy of automatic labeling as much as possible is the key to improve efficiency.

Firstly, color filter is applied to process the pixel information obtained by the digital camera, which is convenient for subsequent processing.

When the Gaussian mixture model (MOG2) is applied to automatic labeling, the image gray histogram reflects the frequency of the occurrence of a certain gray value in the image, which can also be considered as the estimation of the probability density of image gray.The shadow is detected by controlling the threshold value, and the background is separated before the object moves to detect the moving object. Finally, the image is processed, and the smallest outer rectangle is drawn to achieve the segmentation and detection of the moving object.This method has a good effect on the detection of dynamic objects.

In this project, we collected the data of about 1,000 cans, 1,000 plastic bottles and 900 boxes in total. After that, we adopted the method of automatic labeling first, then manually checking the labeling effect and manually labeling, which greatly reduced the workload.

B. *Detection*

Object Detection and Classification

You only look once (YOLO) is a state-of-the-art, real-time object detection system.Based on the hashrate of the machine(i5-10400F@2.9GHz) and the size of the memory space of the GPU(GTX1660@6GB) we use, we used the YOLOv5s to train the data.The following figure shows the structure of the network used for training data.

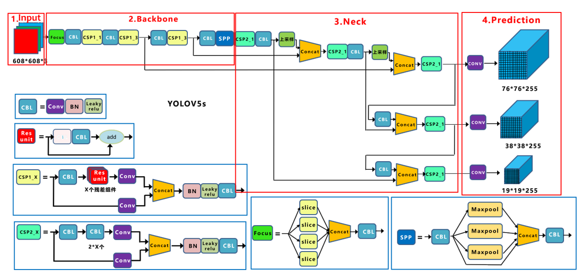


Fig. 1: The structure of YOLOv5s

In order to deploy the latest version of YOLOv5 locally, you first need to download the latest open source code locally, and configure the corresponding version of the python environment according to the model requirements. To train your own custom dataset with YOLOv5, we have to need to perform the following steps:

1. Create dataset.yaml.
   The dataset configuration file that defines 1) a path to a directory of training and validation images and labels 2) the number of classes, 3) a list of class names.
2. Organize directories.
   Organize the train and val images and labels to make the file names of images and labels correspond one-to-one,

and make the file naming in each folder start from 0. YOLOv5 locates labels automatically for each image by replacing the last instance of /images/ in each image path with /labels/.

3. Configuration parameter and Train.
   We train the data from a randomly initialized weight file, and set the batch-size equals 5.After 300 epochs training, all training results are saved to runs/train/ with incrementing run directories.

After about 20 hours of training, we got a well-represented result. The various parameters are shown in the figure below:

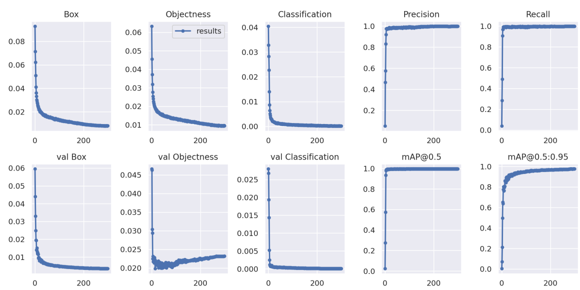


Fig. 2: Training performance

From the figure, we can find that the Precision, Recall and Mean Average Precision of the model are gradually increasing with training and remain stable, which means that the weight parameters of the network have converged, and the high mAP also shows that the training effect is good.

Next, we use the trained weights to detect the image input from the Real-Sense camera above the work space, and return the number, type, and location of waste in the current field of view for the system. Then the detected different types of targets are circled with different colored boxes and displayed on the operating terminal.

Grasping Points Computation

In addition, the detection system also uses the information returned by the depth camera to calculate a better posture of the gripping object for the robotic arm. Here we apply the open source project code of related research[5] [6] . Its core steps are as follows: First, at least 4 points are used to calibrate the plane where the conveyor belt is located. Then use the relationship between the point and the plane in the three-dimensional space to filter those points that are within the frame selection range but not on the plane, that is, the point cloud information corresponding to the object we need to clamp. Perform calculation and feature extraction on the point cloud information of the object, and return two points suitable for grasping on the object. And directly return to the posture where the gripper should be when gripping the object as needed.

C. *Grasping*

After the procedure of detection, we can obtain the specific type of waste from YOLO-v5 and the pose and position of waste from 6D point cloud. Due to known location and pose of waste, we can achieve the static grasp, which means the location of waste will not change with time. But based the situation of waste classification, they always are transported by conveyor belts. As a result, we need to get true grasping position by calculating or estimating.

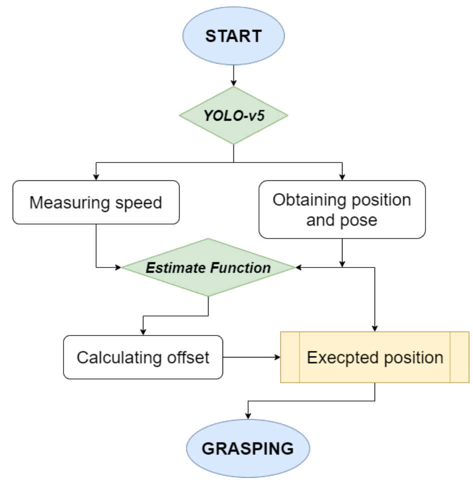


Fig. 3: Grasping Flow Chart

In the whole process of waste sorting task, the real-sense camera will take a picture after one tossing motion. Detection part will give the current position and pose. Actually, at this time, the robot arm still shows the home position predestinated which can not reach the according grasping position immediately. We take the distance between the planned capture position and the actual position of the waste in the camera as the **offset**.

In fact, calculating the accurate capture position in 3-dimensional coordination is quite complex and in our experiment we cannot directly control and obtain accurate conveyor speed. So we used the methods of measuring and fitting to calculate the planned offset.

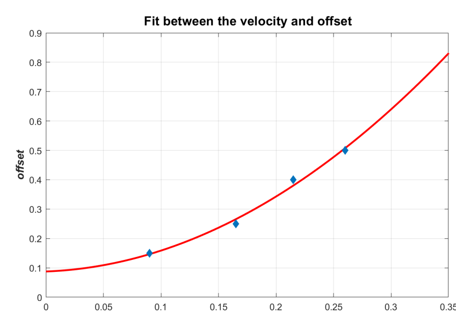


Fig. 4: Fitting Relationship between Velocity and Offset

In the fig.4 shown above, we measured and recorded several sets of data about the conveyor belt speed and the best offset value. The conveyor belt speed is indirectly from obtaining the distance between the center position of the object before and after the shooting 1s through the real-sense camera and the object detection algorithm. And in the according velocity, through experience we set the appropriate offset value to make the gripper grasp waste successfully at the right time. Then we chose quadratic and cubic function for fitting, and finally selected the quadratic fitting function by comparing goodness of two fits.

Due to this estimated fitting function, we planned to find the velocity of conveyor belt after every start. And due to different pose (horizontal/vertical) we generated according function to them.

In addition, one important factor to grasping is the gripper manipulated. In our experiment we use two different types of gripper, pneumatic and motor gripper. They all have advantages and disadvantages for grasping. Pneumatic gripper can provide faster grasping reaction but less force while motor gripper with soft finger[7] can provide enough force but slower grasping shown in fig.5.

(a) motor gripper (b) pneumatic gripper

Fig. 5: Comparison between different grippers

D. Tossing

Since there are too many factors can effect the result of tossing, we need to add some limitations to simplify the question:

1) Speed limit: Intuitively, the faster the robot arm operate, the better. Because that can speed up the whole process of garbage sorting. Nevertheless, the instability of the system can increase with the increase of the operation speed. If the speed is too high, the robot arm can even shakes. After many experiments, finally we find that using **64%** of the maximum speed can obtain a good balance between the stability and the efficiency.
2) Start and end points: To maximize the solution space, we want the operation space to be as large as possible. However, there are also some limitations. Firstly, the start point can not be too low in case the gripper hits the conveyor belt, especially with the garbage. Additionally, the start points should not to be too close to the base of the robot arm. Again, after many experiments, we set the start points to be **0.15m** higher than the conveyor belt and **0.2m** away from the arm base. As for end points, we do not want the arm exceed the range of the conveyor belt, for security reasons. So, we set a plane as the boundary, which is **0.7m** away from the arm base.

With these limitations, we then need to decide how to control the robot arm.[8]

When it comes to the control of the robot arm, there are two main ways: one is linear on the work space(move_L); the other one is linear in the joint space(move_p). With the linearity in the work space, we can obtain a straight line as the trace of the end-effector. While with the linearity in the joint space, we can make the motion of joints to be more fluent, which is friendly to the robot arm. Having the transformation matrix velocity Jacobian matrix, we can also map the motion of the joint space to the work space:

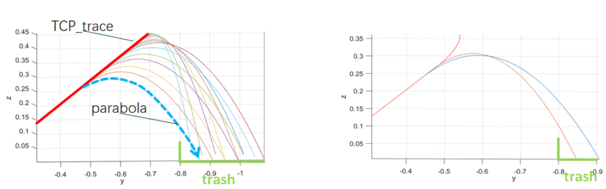
$$P_{ee} = P(\theta_1 \cdot \hat{v}_{j1o}) P(\theta_2 \cdot \hat{v}_{j2o}) \dots P(\theta_6 \cdot \hat{v}_{j6o}) P_{eeo}$$

$$J = [\hat{v}_1 \quad \hat{v}_2 \dots \hat{v}_6]$$

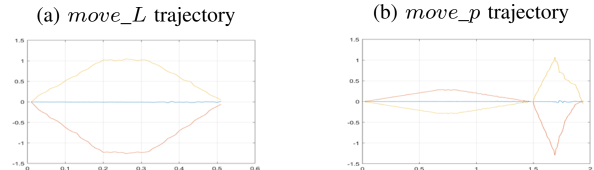
$$\hat{v}_{ee} = J \cdot \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \vdots \\ \dot{\theta}_6 \end{bmatrix}$$

So, it's hard to decide which controlling method can be better.

To compare these two methods, we find a stable and of relatively high success rate tossing trajectory. Using these two methods to control the arm to pass along the trajectory respectively, recording the position and corresponding velocity information at the frequency of about 100Hz. Analyze the data, we can obtain the trace of the end-effector during the whole tossing motion. Sample the trace, we can get some release points and corresponding velocity information. Then, calculate the parabola and obtain the drop point. Only if the drop points are in the range of the trash (0.8-1.3m away from the arm base), the release point can be valid. To count the number of valid release points, we can measure the success rate of tossing using these two methods. As shown in Fig 1. we can find that the success rate of move_L is significantly larger than move_p. Moreover, through the velocity information of these two curves, we can find that using move_p, the velocity of the end-effector is not fluent. Actually, when we use move_p to control the robot arm, the phenomenon of stuttering can occur. So, we can draw the preliminary conclusion that the performance of move_L is better than move_p.



(a) $move\_L$ trajectory (b) $move\_p$ trajectory



(c) $move\_L$ v_{ee} (d) $move\_p$ v_{ee}

Fig. 6: Comparison between two controlling methods

As we decide to use move_L to control the arm, we can find that the trajectory of tossing is almost a straight line and the direction of the end-effector velocity is almost the same as the trajectory (as shown in fig.6). With all these known conditions, to find a relatively optimal tossing trajectory , we also need to decide the angle θ between the trajectory and the horizontal line.

To find the optimal angle θ^*, we need to reverse the process we have done before. That is, with the position of target drop point, we can calculate and obtain the release point on the tossing trajectory with given θ. To simplify the problem, we just focus on three characteristic drop points: two boundary points(**0.8m** and **1.3m** away from the base) and one middle point(**1.05m** away from the base). Because of the inertia, air drag and many other factors, the distribution of drop points corresponding to one release point can seem to fit **Gaussian distribution**. So, among these points, the middle one matters most, since that represents the optimal tossing result.

After calculation, we can obtain the corresponding release points on the tossing trajectory. Because ideally, the map relationship between the drop point and release point is linear. So, to obtain the optimal drop result, the distribution of release points can also fit **Gaussian distribution**. In this case, the expectation can measure the tossing performance of different trajectories.

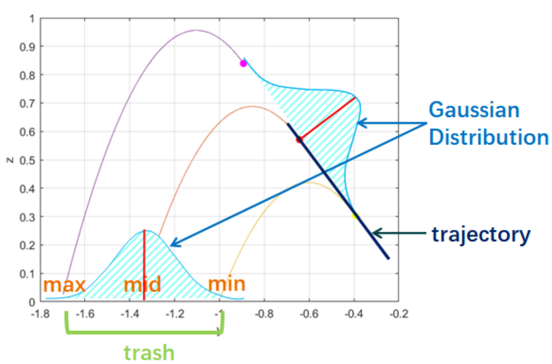
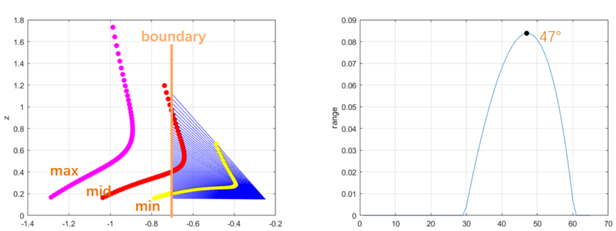


Fig. 7: calculate the parabola to obtain release points

After sample and calculation, we find that the expectation can reach peak when $\theta^* = 47°$.



(a) θ transversal.

(b) optimal θ

Fig. 8: tranverse to optimize the θ

Based on the θ^*, we set different tossing trajectories for different kind of garbage, because their target drop points are different.

III. Experiment Results

A. PNT with PNP

For this system, the processing cycle of the individual materials is separated into 3 sub-actions as described below:

1) Horizontal translation from the current position to the target position.
2) Gripper closes and grasp the waste.
3) Horizontal translation towards the bin that corresponds to the waste material. Gripper opens and place the waste.

The second step consume constant time. However, the first step and the third step are implemented at varying time. Because the time consumption is determined by (a) the bin that the previously separated material has been placed, (b) the position of the current material on the conveyor belt and (c) the location of the bin that the current material must be placed.

B. Detect

We use the YOLO model to detect the waste. As the figure shows, we can detect the waste rapidly and classify it precisely. In order to train the model, we collect a large amount of different types of waste and label them.

Fig. 9: Detect the waste

C. 2D&6D picking

In the project, we use the 2D picking method to grasp the waste. The grabbing location is calculated according to the color map obtained by the sensors and the visual recognition algorithm. Yolo5 landmark detection algorithm is used to obtain the boundary of the object, and the grabbing point is in the middle of the boundary. As for the 6D picking method, the grabbing location is calculated according to the color map obtained by the sensors, point cloud information and the visual recognition algorithm. In the project, the GeoGrasp algorithm is used to obtain the best grasping pose of the object in the camera coordinate system, and then the best grasping pose of the manipulator in the base coordinate is obtained through the transformation matrix. Because the waste is placed on the flat convey belt and their postures are unsophisticated. We find that the 2D picking method is suitable for the grasp experiment.

D. Gripper

In order to figure out the influence of the gripper's open and close time on the tossing, we record the open and close time that a pneumatic gripper and a soft gripper consumes. The grasp speed of the pneumatic gripper is much faster than the soft gripper, it needs 0.2 second and the soft gripper needs 2 seconds, which will result in a small horizontal initial velocity of the waste in the projectile motion.

IV. Discussion

In the present study, we implemented a comprehensive system based on RGB-D camera and robotic arm to identify different kinds of waste, pick them from a moving conveyor and toss them into the corresponding box. According to our experiment result, because of the use of the tossing strategy, the average time consumed for dealing with single garbage is significantly shorter than the conventional pick-and-place method. Nevertheless, there are still details remained to be improved.

A. Detection in the Real World

Although the YOLO algorithm we use can reach a high accuracy on the test data set, its real world performance isn't so good as we expected according to the experiment result. After comparing the classification results obtained in different environment, we discover that the performance would be affected by different light conditions. As a result, more data collected in different environment is needed. Besides, longer training time is also required considering the large data set and the complex network model.

B. Improvement of Tossing Process

According to the experiment results, the PnT method we used can not only save more time than the conventional PnP method, but also allow the waste to be tossed to places outside the workspace of the arm. Nonetheless, other improvement can be done in order to get a better performance. On one hand, a pneumatic gripper with faster operation speed is required so that the waste's release position and speed can be accurately controlled. On the other hand, the base of the robotic arm should be strengthened so that the arm can work in a higher speed without worry about wobble.

C. Optimize Multi-target PnT Strategy

In our present study, we considered the optimization of PnT method with single target. However, there must be more than one target on the conveyor at the same time. Taking the multi-target condition into consideration, the movement of the arm from one target to another would consumed considerable time if present strategy (pick, back to the fixed tossing position, toss, back to the initial position) is applied. Therefore, we can reasonably optimize the order of targets to pick to save the extra time spent on moving the arm. What's more, we can optimize the tossing position and the path of the arm so that the waste can be tossed in the process of moving to next target.[9]

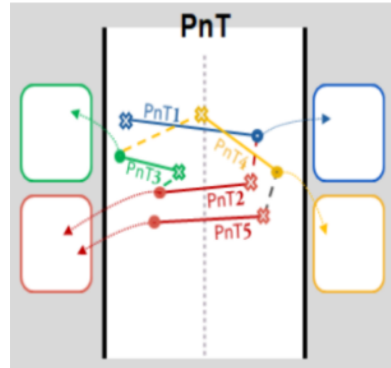


Fig. 10: Multi-target PnT

References

[1] F. Wan, H. Wang, X. Liu, L. Yang, and C. Song, "Deepclaw: A robotic hardware benchmarking platform for learning object manipulation*," in *2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 2020, pp. 2011–2018.

[2] F. Raptopoulos, M. Koskinopoulou, and M. Maniadakis, "Robotic pick-and-toss facilitates urban waste sorting *," in *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, 2020, pp. 1149–1154.

[3]

[4] A. Zeng, S. Song, S. Welker, J. Lee, A. Rodriguez, and T. Funkhouser, "Learning synergies between pushing and grasping with self-supervised deep reinforcement learning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 4238–4245.

[5] Zapata-Impata *et al.*, "Using geometry to detect grasping points on 3d unknown point cloud. in proceedings of the 14th international conference on informatics in control," *Automation and Robotics (ICINCO)*, vol. 2, p. 154–161, 2017.

[6] ——, "Fast geometry-based computation of grasping points on three-dimensional point clouds," *International Journal of Advanced Robotic Systems*, p. 16, 2019.

[7] Z. Yang, S. Ge, F. Wan, Y. Liu, and C. Song, "Scalable tactile sensing for an omni-adaptive soft robot finger," in *2020 3rd IEEE International Conference on Soft Robotics (RoboSoft)*, 2020, pp. 572–577.

[8] T. Senoo, A. Namiki, and M. Ishikawa, "High-speed throwing motion based on kinetic chain approach," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 3206–3211.

[9] F. Raptopoulos, M. Koskinopoulou, and M. Maniadakis, "Robotic pick-and-toss facilitates urban waste sorting *," in *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, 2020, pp. 1149–1154.