A Mobile Order Fulfillment Robotic System for Warehouse Automation

Liguang Zhou[1,2†],Yanwei Huang[1,2†], Yiyao Zhu[1,3], Pizeng Zhou[2], Zhenglong Sun[1,2],
Yongquan Chen[1,3*],Yangsheng Xu[1,2]

Abstract— **Nowadays, robots hold the key to increase the efficiency of the logistics industry. To cope with the growing demands for higher productivity in logistic systems, we present a novel mobile order fulfillment robotic system in retail warehouses, to increase the efficiency of order fulfillment. In the warehouse, the robot is commanded to pick up all items from several places according to the order placed by customers and expected to travel through relatively shorter distance by fulfilling multiple items in a single trip. To this end, we first propose a warehouse management system (WMS) for reducing the time and traveling distance for order fulfillment. By formulating the order fulfillment problem in the warehouse as a variant of Generalized Travel Salesman Problem(GTSP), we combine the Local Search (LS) and Exchange Operation (EO) to solve this problem. Then, to achieve robust item picking, we enhance the suction point selection by keeping a relative position of the camera to objects with a lifter, and we also combine the deep-learning-based object detection algorithm with modeless geometric analysis for precision suction point selection.**

Numerous experiments have been conducted to validate the effectiveness of our proposed solution. For WMS system, both time and traveling distance of robot are significantly reduced, which save an average of 57.95% time and reduce 0.71 lengths compared with sequencing algorithm. Moreover, with the lifer mechanism, the trained model can reach 0.992 mAP over 15 different items. Furthermore, we conduct real experiments in a retail warehouse, which shows the proposed system can pick most items in the retail warehouse with an average success rate of 85.8%.

I. Introduction

With the cost of labor continues to grow, the use of automatic equipment such as robots in warehouses has become a mainstream trend. However, there is still a problem in the e-commerce warehouse system. Can our robot have better efficiency and stability over humans? Aims to boost the warehouse automation through robot, Amazon, as one of the leading e-commerce company in the world, held the first Amazon Picking Challenge (APC) in 2015, for seeking the solutions to robust item picking and high efficiency order fulfillment in unmanned warehouses, and the APC competition has been successfully held for three years [1]. JingDong, as one of the giant e-commerce company in China, also intends to push the limits and improves the capabilities of item picking robots for retail warehouse automation, held its robotic competitions in 2017 (JRC2017) and 2018 (JRC2018), which significantly connecting the academia and industry and boosting solutions to autonomous robotic system for order fulfillment in unmanned warehouses.

In JRC2018, Jingdong designed a retail warehouse as Figure 1 and 4 shows. The retail warehouse is equipped with two two-level shelves, two desks, one checkout desk for item delivering by robot and one entrance. Robot is required to pick up specific items according to the order placed by customers. The key challenges of JRC2018 include: the choose of an efficient picking strategy and the robustness of both object recognition and item picking.

As we can find in Figure 1, robots are allowed to pick one type of item from different positions such as on both sides of a shelf, and they are also allowed to fetch multiple items from one position as long as the items are within its reachable workspace. For example, in Figure 1, there are four types of object (from left to right they were 'spray', 'tablet', 'toothpaste', 'biscuit') placed on the lower layers of the shelves, and the robot can pick 'toothpaste', 'spray', and 'tablet' without moving its chassis. Based on this feature, we design an efficient Warehouse Management System (WMS) to improve the system's efficiency by picking several items from one position and optimizing the total traveling distance of order fulfillment.

Besides, according to the scenario, the items are well-placed on the shelves, so the accuracy of recognition can be greatly improved if the robot can capture images of the items from a comparatively stable direction. In order to find an accurate suction point for vacuum suction cup, we take a wide variety of adjustments. Firstly, to ensure a comparatively more stable relative position of items regarded with the camera frame, a lifter mechanism is designed, and the camera is installed in an eye-to-hand mode. Secondly, we combined the YoloV3 [2] and a modeless geometry analysis method for the suction point selection. Lastly, high-precision 2-D SLAM is used to offer a robust and repeatable localization.

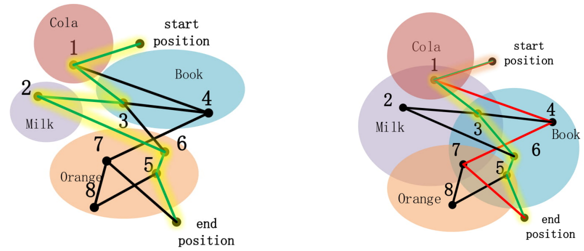
In summary, the contributions of our novel and efficient solution for order fulfillment robot in a retail warehouses are as follows:

- We propose a warehouse management system (WMS) for efficient order fulfillment with Local Search (LS) and Exchange Operation (EO).
- We design a perception pipeline by combining deep learning based modeless geometry analysis to provide

* Corresponding author: yqchen@cuhk.edu.cn
† Authors contributed equally
[1] Shenzhen Institute of Artificial Intelligence and Robotics for Society
[2] Robotics and Artificial Intelligence Lab, The Chinese University of Hong Kong, Shenzhen, 518172 P.R. China
[3] Institute of Robotics and Intelligent Manufacturing, The Chinese University of Hong Kong, Shenzhen

Manuscript 2507 submitted to 2020 International Conference on Robotics and Automation (ICRA). Received September 16, 2019.

Fig. 1: The real scenario of JCR2018



(a) (b)

Fig. 2: The differences between GTSP and this scenario

the high-accuracy suction point selection for vacuum suction cup.

We conduct experiments comparing the WMS and the most frequently used method, picking items in sequence based on 2000 testing sets. As a result, our proposed method save an average of 57.94% time-cost and reduce the 0.71 lengths of traveling distance. Besides, our system can reach an average success rate of 85.8% in item picking, which sufficiently proves the efficiency of our system.

II. Related Works

In this section, work related to item picking robot system in the unmanned retail warehouse is reviewed.

1) Planning Algorithm: Scheduling algorithms have been widely used in the unmanned retail warehouses to promote efficiency. Typically, robots are not only required to fetch all the wanted items but also expected to fulfill all the tasks with minimum time cost according to a target list. Surveys [3] [4]concerning this scenario shows that in most cases, robots are normally allowed to pick only one item in one position and a specific item can merely be found in one place. In the view of Operations Research, picking and dispatching problems are often transformed into a Generalized Travel Salesman Problem(GTSP) [5] [6] [7], in which the vertexes in a graph are partitioned into several clusters, and a tour passing through all those clusters are required to be found with the minimum sum of weight. Generally, people tend to tackle with this problem by using heuristic search methods such as Branch-Cut Algorithm [8], Genetic Algorithm [9] or Ant Colony Algorithm [10]. However, it seems that there is few related works having considered about making an optimization under this scenario, where robots are allowed to choose a path consisting of positions which have access to more than one items, and thus the robot can integrate several tasks into one and fulfill them with merely one visit.

2) Solutions for Well-organized Warehouses: In order to fulfill the orders from customers, the system are required to identify the items and estimate the pose of target items placed on the shelf. Team NimbRo proposed the deep object perception pipeline for quick and efficient object learning and new item adaption by leveraging the turnable table [11] [12] [13]. In addition, the pose was refined by the ICP algorithm [14]. However, these methods require the CAD model library of all objects for object matching and 6d pose estimation, whereas the construction of CAD model library require extra time and resources. Aims to improve the efficiency of the CAD model construction process, DoraPicker [15] explores the method that automatically scan the CAD models for pose estimation. Recently, one model-less estimation method was proposed for item picking and grasping using the 3d shape primitives for item shape and pose approximation [16]. However, it requires to recognize the object surfaces by using ShapeNet, which requires the preparation for 3D label data. Unlike the other methods, which are using the CAD model for pose estimation by evaluating the objects that are with similar shapes. However, the CAD model is tedious and extremely time-consuming to construct. For human beings, a more natural way to fetch an object is by their intuition, which is a modeless process because they don't need to estimate the pose of an object. In other words, by leveraging the prior information learned from daily lives, they already have an excellent and robust grasping strategy. Therefore, we propose a deep learning based model-less suction point selection method for item picking without 3D CAD model construction.

III. Warehouse Management System (WMS)

A. Planning Problem Description

The goal of WMS is to help the robot fetch all the designated items with minimum time. Generally, this problem is generally modeled as a Generalized Travel Salesman Problem(GTSP). However, there is the main difference between GTSP and this scenario. For example, FIGURE 2(a) shows the Venn diagram of GTSP. In this graph, every vertex in the graph can only belong to one cluster. By contrast, as can be seen in FIGURE 2(b), in this scenario, clusters may intersect with the others, leading to some vertexes partitioned into several clusters. The robot can pick more than one items at a station and can also obtain the same object from different stations, it leaves space to optimize the picking efficiency by seeking a set of stations allowing the robot to integrate more than one orders into one. As FIGURE 2 (b) shows, to travel through all the four clusters in the graph on the way from the start position to the end position with higher efficiency, two paths (the red and the green path) are found and compared.

Manuscript 2507 submitted to 2020 International Conference on Robotics and Automation (ICRA). Received September 16, 2019.

Although it is true that the red one (s-1-4-7-e) travels through the shortest vertexes in the graph, the green path (s-1-3-6-5-e) achieves a better performance in terms of the total time cost.

B. Order Planning Algorithm

As finding a solution to this question is an NP-hard process, the calculation can be unaffordable when the scale of the problem goes larger. To tackle this problem, we design the local search method and an Exchange Operation(EO) to speed up this process.

1) Local Search: The robot's efficiency under a time constraint is associated with the places the robot choose to visit and the visiting sequence. In this paper, we propose a fast local search method for upgrading the execution sequence of item picking. The main steps of our proposed search algorithm are described as follows.

- S1: Generate a solution S_0 by using the Greedy Algorithm(GA) [17];
- S2: Generate its neighbor solutions S_{01},S_{02} ...S_{0n} with EO;
- S3: Estimate a comparatively better visiting order by appiled metric TSP algorithm on each neighbor solutions [18], and then return the best solution S_{best}found in this process;
- S4: If the S_{best} changes, repeat S2 , S3 and S4. Otherwise, go to S5
- S5: Return the final feasible solution S_{best}

2) Exchange Operation: Before introducing the Exchange Operation, we first introduce the definition of the redundant and non-redundant feasible solution:

Definition 1: Redundant Feasible Solution is a feasible solution can remain feasible if a vertex in it is removed.

Definition 2: Non-redundant Feasible Solution is a feasible solution becomes infeasible if any vertex in it is removed.

The details of EO are described as follows.

- S0: Input a feasible solution S_0
- S1: Make sure the feasible solution S_0 is a non-redundant one. If yes, continue.
- S2: Make it an infeasible solution by deleting a vertex except the current and the destination vertex
- S3: Search S vertexes, which store available items, as a substitute of the deleted vertex and reconstruct the solution into a feasible one, in which $S = 1, 2...I.I < MAX$;
- S4: Make the solution a non-redundant feasible solution again by deleting all those redundant vertexes

3) Complexity Analysis: Based on an adjacency matrix storing all the weights of edges in a topology graph, we assume that the robot needs to visit E vertexes. Thus the EO can be conducted on totally $E-2$ vertexes except for both the current and the destination vertex. For each of them, assuming there are on average S vertexes available for substituting, the S2 in Local Search could yield out totally $(E-2)(C_S^1+C_S^2...+C_S^T) \leq (E-2)2^S$ solutions. The exponential calculation always makes the algorithm unaffordable. Nevertheless, in practice, we choose $2^S \leq 16$ as MAX usually is not greater than 4. Because the complexity of metric TSP is $O(ElogV)$, the upper bound of the complexity for a round of Local Search is $O(E^2logV)$.

IV. Enhanced Suction Point Selection

A. mechanical layout

In JRC2018, all the items are placed row by row on the desks and the shelves in the given warehouse as Figure 4 shows. In the experiment, we found that the bounding boxes predicted by YOLOV3 fluctuate in a small range, which increases the uncertainty picking process of the robotic system. If we can assume a fixed relative position and orientation of items to the Kinect camera, both the recognition accuracy of YOLOV3 and the successful picking rate of robotic system will be higher and makes a more robust robotic system. To reduce this uncertain and increase the robustness of robotic system, a lifter is designed, and the Kinect camera is mounted on the lifter and can adjust the relative height w.r.t the items in the field view. For example, if the robot is required to pick the items in the upper layer of the shelves, the lifter will first adjust the position of the Kinect camera to the upper position, which is about 18cm higher than the upper shelves. In practice, this solution can effectively improve the recognition accuracy of the predicted bounding boxes and therefore increase the robustness of the item picking.

B. perception pipeline

The perception pipeline of our system is shown in Figure 3. The perception algorithm takes visual information from Kinect V2 camera as the input and outputs the available local patch for suction. The pipeline can be described as following steps: first, RGB image and point cloud are captured by Kinect V2 camera, then, a deep learning based object detection algorithm YoloV3 is used to predict the bounding boxes from the input RGB image, and the corresponding point cloud with target objects are acquired by using the bounding boxes information. Next, the target objects will be segmented individually, and depending on the physical parameter of the target object, items are divided into two categories for suction, one is for suction side, and the other is for suction down. Last, the 4-connectivity neighbors patch for the individual point cloud of target items is depicted, according to the human's prior knowledge and visual information of the geometrical properties of target items, an optimal patch for suction will be picked up.

1) Point cloud Preprocessing: The raw point cloud captured from Microsoft Kinect V2 contains more than 500,000 3D points with color, which is very dense and noisy, and such a massive amount of point will slow down our perception pipeline. For instance, there are a lot of noisy outliers like doors and walls that will prevent our recognition pipeline from recognizing the target object. Therefore, we only take the points that are in the bounding boxes and generate the new point cloud as shown in Figure 3 by filtering out the points, which are not in the range that UR5 can reach.

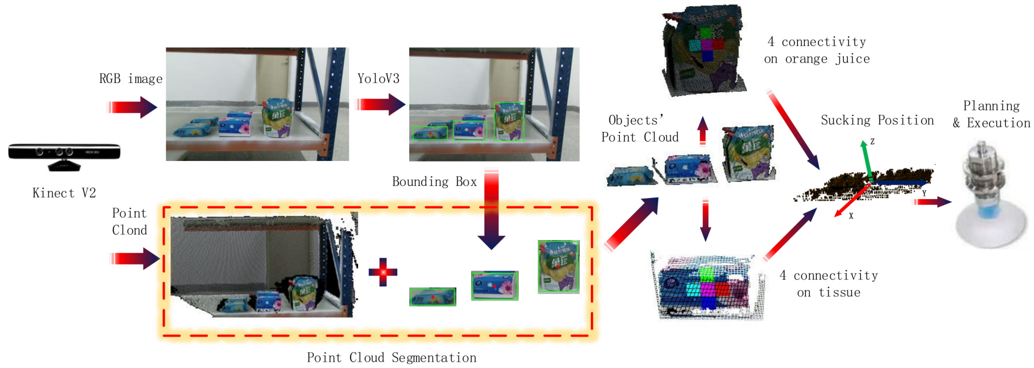


Fig. 3: The main procedure of suction patch selection of the proposed system

2) Deep learning for bounding boxes proposal: The object detection is very critical for item picking robot. In our paper, we used the open source deep learning based object detection algorithm YoloV3 for inferring the bounding boxes from the input RGB image. Yolov3 performs indubitably well on the object detection and have a high mAP on the testing set. Although the bounding boxes acquired by previous deep learning based algorithms reaches a start-of-art performance, the system requires more detailed information for end-effector to pick up the objects from the real position in the shelf. The fundamental challenge is how to select the point from bounding boxes accurately. Besides, there are some limitations for the bounding boxes inferred by YoloV3. For example, in JRC2018 scenario, there are multiple items placed in one column in the shelf, or the items are placed far away from the camera center.

In these cases, the predicted bounding boxes suffer from a lower IOU, which means the bounding boxes is deviating from the ground truth in some degrees, if the center of the bounding boxes are used to do grasping execution, it will lead to the picking failure case.

3) 4-connectivity neighbors patch: For the items that need suction side, inspired by the ideal from 4-connectivity neighbors [19] in image processing, a patch located on the center of bounding boxes are selected by a selective patch size, which will be tuned by grasping experiment. The patch size is related to the size of the suction cup and physical size of the target objects.

Next, 4-connectivity neighbors patches around the center patch of the point cloud are generated as the local patch proposal for suction point correction. By picking the local patch among five patches by experiments, we chose the local patch with most grasping success rate. With experiments, the system will learn the most feasible suction patch selection s-trategy with the prior information and sufficient experiments. For the items need suction down, the corresponding plane surface will be selected by filtering out the surfaces that not parallel to the shelf surface with the surface normals information. We first select the center of the plane from the segmented plane and find the four neighbor patches. Then, we evaluate each local path based on the surface normals and prior information from our experiment. Finally, we conducted several experiments on the candidate patch to validate our prior knowledge. Moreover, the turntable for automated object labelling will be introduced.

V. Experiment

A. Hardware

Our robot system consists of the following parts:

1) Mobile robot base - used for conveying the robot arm to the desired station and grasping the items from the order list
2) UR5 - produced by Universal Robots, whose accuracy is ± 0.1mm
3) Lifter - made to increase the reachable workspace UR5 and provide a stable FOV, whose accuracy is ± 0.1mm.
4) Industrial Computer - with a 3.4 GHz Intel®Core i7 6700 GPU, and NVIDIA GPU GTX1050Ti with a 4G memory. The OS is Ubuntu16.04, and all the codes were written with C++ or Python. The communication protocol among all programs is ROS-based
5) Camera - KinectV2, used as a visual sensor for capturing the RGB image and point cloud data of items placed on the shelf and desk
6) Gripper - is a suction cup with 4cm diameter

1) WMS Evaluation: To evaluate the performance of the algorithm given in Section IV, we built up a map according to the scenario proposed in JRC2018, where two double-layer shelves and two desks were placed inside a 6×8 m^2 court, and an estimated 4-5 different items were well-placed in a random sequence on the desks and each layer of the shelves. In addition, to record all the necessary information, a directed graph consisting of 10 vertexes standing for the crossroad positions in the warehouse and 26 vertexes standing for the available picking position near the shelves was generated to store the information of the warehouse. The weight of edges between vertexes represented the time cost the real robot needed to move through the corresponding positions in the real scenario.

The task list we sent to the robot includes the name and the number of items the customers want, and then WMS processes the information, plans the execution sequence and returns a solution. In this work, we generated tasks in a wide

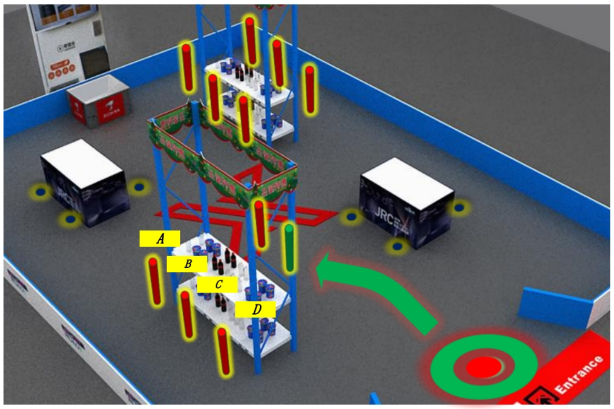


Fig. 4: A scenario given in WMS

TABLE I: RESULTS ON 5*4*100 TESTING SET

VARIABLE	VALUE				AVERAGE
$Number_T$	5	10	15	20	- -
$\hat{T}_{SEQ}$	2364.22	3921.97	5496.74	7293.40	- -
$\hat{T}_{LS}$	1033.43	1876.99	2349.16	2471.62	- -
$Improve$	56.26%	52.14%	57.26%	66.11%	**57.94%**
$\hat{Num}_{SEQ}$	4.60	6.90	8.55	10.01	- -
$\hat{Num}_{LS}$	4.25	6.05	7.54	9.40	- -
Num_{Imp}	-0.35	-0.85	-1.01	-0.61	**-0.71**

variety of scales, including 5, 10, 15 and 20 items, respectively. We conducted a group of experiments consisting 100 randomly generated tasks with items' position unchanged, and repeated five groups of experiments after changing items' position, totally leading to $4 * 100 * 5 = 2000$ times of task fulfillment. As could be seen in Table I, $Number_{Task}$ denotes the length of our task list. $\hat{T}_{SEQ}$ and $\hat{T}_{LS}$ denotes the average time cost of 4* 100* 5 tasks of sequencing-execution method and WMS respectively. $\hat{Num}_{SEQ}$ and $\hat{Num}_{LS}$ demotes the average number of vertexes the robot has to pass through. Overall, our proposed WMS could achieve an average improvement of 57.94 % in time cost compared with the most used method — executing in sequence. Meanwhile, we found it could make a reduction in the number of places that robot need to pay a visit by around 0.71, which fully prove that the WMS can improve the efficiency of the robot.

2) Object Perception: The RGB-D camera Kinect V2 is mounted on the base of the robot arm (UR5) for detecting the objects placed in the shelves and desks. In order to build up the accurate and robust detection algorithm for object searching, we adopt the advanced deep learning based technology to train a neural network for the detection task. To this end, a large-scale and manually labeled image dataset, including 15 different daily items in Figure 5 and 2812 images in total are collected for model training and testing. The images of objects have been collected in three different scenarios in a retail warehouse includes the upper shelf, bottom shelf of the two-level shelf and on the desk. In addition, aiming to increase the robustness of the object perception algorithm in multi-angles, we build a turnable table for single object capturing in 360 degrees. The total dataset is 2812 images, where the 2300 images are captured on the shelf and desk and the rest 512 images captured on the turntable table with a single object. We manually label these images. The dataset captured on the shelf and desk is then divided into a training dataset and testing dataset with a ratio of 4:1. We implemented YoloV3 to train a model for object detection. The image will be resized to 416x416 before feeding to the neural network model. Table II shows the trained model reach 0.992 mAP over 15 different objects, which is a relatively high detection precision, meaning the trained model has extremely high detection accuracy.

Fig. 5: All the target items: Row-by-row starting from top-left: milk, book, tissue, toothpaste, shampoo, french fries, cola, Oreo cookies, orange juice, gutta pertscha, toothbrush, orange, spray, chips, biscuit.

TABLE II: The experimental results of the YoloV3-416, 416 means the input image size is 416×416.

object	mAP
biscuit	1.000
book	1.000
chips	1.000
cola	0.998
french fries	0.996
gutta pertscha	0.983
milk	0.975
orange	1.000
orange juice	0.993
oreo	0.991
shampoo	1.000
spray	0.961
tissue	0.999
toothbrush	0.992
toothpaste	0.999
overall	**0.992**

3) Robot System Evaluation: The proposed order fulfillment robotic system will fulfill an order by executing the following five procedures:

- S0 Try to establish the environment map for robot localization and navigation
- S1 Wait for the orders from customers. After receiving the orders, the WMS system plans a task execution sequence for order fulfillment

Fig. 6: The experiments for 12 randomly generated orders, the system fulfills the order successfully three times. The worst case of the system is 6 items were picking out of 10 target items.

TABLE III: The summary of successful rate of item picking among 12 experiments

Items	Characteristics	Number of successful picks	Number of attempts
biscuit	Flat	10	10
book	book	6	7
chips	irregular	10	10
cola	cylinder	8	10
french fries	cylinder	7	7
gutta pertscha	irregular	6	9
milk	box	8	8
orange	sphere	8	8
orange juice	irregular	7	7
oreo	cylinder	7	7
shampoo	cylinder & irregular	3	6
spray	small flat	5	5
tissue	box	6	8
toothbrush	irregular	3	6
toothpaste	long box	7	7
overall		**103**	**120**

- S2 Move to the target positions with the help of the environmental map built-in S0
- S3 Process the RGB image and point cloud data captured by the RGB-D camera and detect the target objects and calculate the suction point
- S4 Execute the item picking procedure to get all the expected items and stow them into the tray

We randomly generated 12 orders in total for the experiment and each order with 10 items (15 items in total). We conducted 12 experiments as Figure 6 shows, the robot is waiting for the orders at the starting point. According to the tasks in the given order, our robot traversed the upper and lower layers of each shelf, and the desk, to find the target objects that are in the given order. Our proposed robot has fulfilled order successfully for 3 times (all items in the order list have been grasping to target tray). Table III is the result of our experiment. The experimental results show that our system is very robust, but it still has some shortcomings, for example, the successful rate of grasping toothbrush and shampoo is 50%, we didn't pick up items for 3 times out of 6 trials, indicating that the toothbrush and shampoo are extremely challenging for the suction cup. Also, for items like gutta pertscha, due to the irregular physical shape of it is a package, our system picks up six times over nine trials. Therefore, our system is more adaptable for the flat surface and hexahedron shape objects.

VI. Conclusion

In this paper, we presented a novel robotic system for effective and efficient warehouse automation. According to the retail store scenario offered by the JRC2018, our solution takes the advantages of the special scenario where the robot is allowed to grasping an object from different stations, and try to reduce the time wasting of task execution by integrating several tasks into one. Besides, a lifts mechanism is designed to maintain relatively fixed field-of-view (FOV) between camera and shelf or desk, leading to a more stable object perception performance. Furthermore, a set of point cloud processing methods based on geometry analysis and prior information of human knowledge is integrated for finding the suction point for grasping. These methods are ensuring the success rate of the grasping executed by vacuum suction cup. We carried out the simulation experiments to validate the effectiveness of our proposed WMS algorithm. The scheme can save 57.95% of time cost and reduce 0.71 lengths of traveling distance for the mobile base compared with the typical sequential execution. Also, we conducted 12 experiments on the real robot in a retail warehouse scenario to validate the effectiveness of our proposed item picking system. The experiment results show that the average success rate of picking can reach 85.8%. The success rate fully demonstrates the efficiency and robustness of our proposed system.

Acknowledgment

This paper is partially supported by Shenzhen Fundamental Research grant (JCYJ20180508162406177) and the National Natural Science Foundation of China (U1613216) from The Chinese University of Hong Kong, Shenzhen. This paper is also partially supported by funding from Shenzhen Institute of Artificial Intelligence and Robotics for Society. We would like to thank all the team members of the IRIM-Solvers at the CUHKSZ and IRIM for excellent robot system construction.

References

[1] N. Correll, K. E. Bekris, D. Berenson, O. Brock, A. Causo, K. Hauser, K. Okada, A. Rodriguez, J. M. Romano, and P. R. Wurman, "Analysis and observations from the first amazon picking challenge," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 1, pp. 172–188, 2018.

[2] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[3] K. Ilavarasi and K. S. Joseph, "Variants of travelling salesman problem: A survey," in *International Conference on Information Communication and Embedded Systems (ICICES2014)*. IEEE, 2014, pp. 1–7.

[4] C. Renzi, F. Leali, M. Cavazzuti, and A. Andrisano, "A review on artificial intelligence applications to the optimal design of dedicated and reconfigurable manufacturing systems," *The International Journal of Advanced Manufacturing Technology*, vol. 72, no. 1-4, pp. 403–418, 2014.

[5] B. Englot and F. Hover, "Planning complex inspection tasks using redundant roadmaps," in *Robotics Research*. Springer, 2017, pp. 327–343.

[6] K. Treleaven, M. Pavone, and E. Frazzoli, "Models and efficient algorithms for pickup and delivery problems on roadmaps," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. IEEE, 2012, pp. 5691–5698.

[7] J. M. Díaz-Báñez, G. Hernández, D. Oliveros, A. Ramírez-Vigueras, J. A. Sellarès, J. Urrutia, and I. Ventura, "Computing shortest heterochromatic monotone routes," *Operations Research Letters*, vol. 36, no. 6, pp. 684–687, 2008.

[8] M. Fischetti, J. J. Salazar González, and P. Toth, "A branch-and-cut algorithm for the symmetric generalized traveling salesman problem," *Operations Research*, vol. 45, no. 3, pp. 378–394, 1997.

[9] D. Karapetyan and G. Gutin, “Lin–kernighan heuristic adaptations for the generalized traveling salesman problem,” *European Journal of Operational Research*, vol. 208, no. 3, pp. 221–232, 2011.

[10] J. Yang, X. Shi, M. Marchese, and Y. Liang, “An ant colony optimization method for generalized tsp problem,” *Progress in Natural Science*, vol. 18, no. 11, pp. 1417–1422, 2008.

[11] M. Schwarz, C. Lenz, G. M. Garca, S. Koo, A. S. Periyasamy, M. Schreiber, and S. Behnke, “Fast object learning and dual-arm coordination for cluttered stowing, picking, and packing,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, Conference Proceedings, pp. 3347–3354.

[12] M. Schwarz, A. Milan, C. Lenz, A. Munoz, A. S. Periyasamy, M. Schreiber, S. Schller, and S. Behnke, “Nimbro picking: Versatile part handling for warehouse automation,” in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, Conference Proceedings, pp. 3032–3039.

[13] M. Schwarz, A. Milan, A. S. Periyasamy, and S. Behnke, “Rgb-d object detection and semantic segmentation for autonomous manipulation in clutter,” *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 437–451, 2018.

[14] A. Causo, Z.-H. Chong, R. Luxman, and I.-M. Chen, “Visual marker-guided mobile robot solution for automated item picking in a warehouse,” in *Advanced Intelligent Mechatronics (AIM), 2017 IEEE International Conference on*. IEEE, Conference Proceedings, pp. 201–206.

[15] H. Zhang, P. Long, D. Zhou, Z. Qian, Z. Wang, W. Wan, D. Manocha, C. Park, T. Hu, and C. Cao, “Dorapicker: An autonomous picking system for general objects,” in *Automation Science and Engineering (CASE), 2016 IEEE International Conference on*. IEEE, Conference Proceedings, pp. 721–726.

[16] T. Torii and M. Hashimoto, “Model-less estimation method for robot grasping parameters using 3d shape primitive approximation,” in *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*. IEEE, Conference Proceedings, pp. 580–585.

[17] G. Gutin, A. Yeo, and A. Zverovich, “Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the tsp,” *Discrete Applied Mathematics*, vol. 117, no. 1-3, pp. 81–86, 2002.

[18] H.-J. Böckenhauer, J. Hromkovič, R. Klasing, S. Seibert, and W. Unger, “An improved lower bound on the approximability of metric tsp and approximation algorithms for the tsp with sharpened triangle inequality,” in *Annual Symposium on Theoretical Aspects of Computer Science*. Springer, 2000, pp. 382–394.

[19] R. Haralick, “Some neighborhood operators,” in *Real-Time Parallel Computing*. Springer, 1981, pp. 11–35.