Benchmarking of Motion Planning Algorithms with Real-time 3D Occupancy Grid Map for an Agricultural Robotic Manipulator

Seyed Abdollah Vaghefi, Mohd Faisal Ibrahim, Mohd Hairi Mohd Zaman
Department of Electrical, Electronic and Systems Engineering
Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia
43600 Bangi, Selangor, Malaysia

***Abstract*—The performance evaluation of motion planning algorithms for agricultural robotic manipulators is commonly performed via benchmarking platforms. However, creating a realistic benchmarking scene that constrains the motion planning algorithms with the characteristic of a real-work environment has always been a challenge worthy of research. In this paper, we present a lab-setup benchmarking platform to evaluate Open Motion Planning Library (OMPL) motion planners for the application of a robotic harvester of a palm-like tree using a real-time 3D occupancy grid map. First, three motion problems were defined with different levels of complexity based on a real oil palm fruit harvesting task. To achieve reliable outcomes, the benchmarking scene was modeled by converting point cloud data from a stereo-depth sensor into a 3D occupancy grid map using the Octomap algorithm. Then the benchmarking was performed, all within a real-time process. According to the results, a fair performance evaluation was achieved by modeling a realistic benchmarking scene, which can help in choosing a high-performing algorithm and efficiently conducting such harvesting tasks in real practice.**

Keywords*—*Motion planning; agricultural; harvesting; robot manipulator; benchmarking; oil palm

I. Introduction

Over the last few decades, intelligent robots for fruit harvesting have been actively developed to bridge the increasing gap between feeding a fast-growing population and limited labor resources. Moreover, the recent international travel restrictions due to the COVID-19 pandemic in 2019-2021 have exacerbated the limited labor resources issue, leading to the unavailability of seasonal migrant workers [1]. A robotic harvester can significantly boost productivity by reducing labor and production expenses, increasing yield and quality, and improving environmental management [2]. However, current harvesting robots are limited in their capabilities in motion planning, designed for specific plant structures. This limitation is due to the target crops' unstructured and dynamic nature and obstructions within their working environment [3], [4]. Environment obstructions, such as branches and leaves, reduce the performance efficiency of harvesting robotic manipulators. In addition, they are likely to collide with those obstacles when performing harvesting tasks. Therefore, establishing a benchmarking approach to accurately evaluate a collision-free motion planning algorithm for a given task is crucial to increasing the performance of robotic harvesting manipulators.

This study aimed to perform a benchmarking of different motion planning algorithms based on real-time perception, i.e., 3D occupancy grid mapping for a robotic harvester of a palm-like tree application. In general, palm-like trees, such as oil palm, dates, and coconut trees, have unique morphological characteristics that challenge motion planning algorithms differently from other crops. The benchmarking of motion planning algorithms based on real-time 3D occupancy mapping should achieve reliable outcomes due to the actual characteristics of the working environment, which is mimicked from the real working environment into the benchmarking scene and constrains the motion planning algorithms. In this work, four motion planning algorithms from the OMPL library, namely, RRTConnect, BiTRRT, BFMT, and FMT, were benchmarked using ROS and MoveIt platform. The process outlined in this study contributes significantly to performing the motion planning benchmarking based on realistic environmental conditions. Furthermore, it can help adopt a high-performing motion planning algorithm and effectively execute such harvesting tasks in actual works.

II. Related Work

In this section, we discuss the role of various benchmarking platforms in evaluating motion planning algorithms for different applications in the literature. In addition, a brief overview and application of 3D occupancy grid mapping in robotic real-time perception will also be presented.

A. Motion Planning Benchmarking

Motion planning is a fundamental topic in robotics that deals with finding an optimal path that satisfies a target specification subject to constraints [5]. The issue of "which planner to choose" could be hard to answer, given the wide range of applications that robotic manipulators are used for [6]. During the last few years, several works have compared and analyzed the motion planning algorithms via benchmarking for different applications during the last few years. Iversen and Ellekilde [7] presented a benchmark for a set of motion planning algorithms based on three different scenarios for bin-picking applications. Despite longer planning time, the algorithms integrated with optimization outperformed due to faster execution. Morgan et. al [8] proposed three different robot benchmarking protocols, namely the Modified Box and Block Test (BBT), Targeted-BBT, and Standard-BBT, for assessing various aspects of the system separately and the results compared with human performance. Chatzilygeroudis et al. [9] represented a new benchmarking protocol to evaluate algorithms for bi-manual robotic manipulation of semi-deformable objects. Therefore, the work makes the benchmark accessible to various related fields, from adaptive control and motion planning to learning the tasks through trial-and-error learning. Jedrzejczyk et al. [10] have investigated a tomato harvesting application and suggested a benchmark of optimally configured motion planners available within Robot Operating System (ROS) and MoveIt platforms. The results indicated a comparison of efficiency and repeatability of particular planners for a planning scene imitating conditions in a greenhouse or similar pick-and-place tasks. Magalhães et al. [11] suggested benchmarking path planner algorithms from Open Manipulator Planning Library (OPML) for tree pruning tasks. Thus, the results demonstrated good agreement for the BiTRRT algorithm compared with other algorithms from the OMPL library, such as BKPIECE, LBKPIECE, SBL, and others. Despite the numerous studies that benchmarked motion planning algorithms for robotic manipulators in industrial applications, few works focused on agricultural applications. Accordingly, this paper studied benchmarking motion planning algorithms for a palm-like tree harvesting application.

B. 3D Occupancy Grid Mapping

Robotic perception is understood as a system that endows the robot with the ability to perceive, comprehend, and reason about the surrounding environment. In addition, robotic perception is crucial for a robot to make decisions, plan, and operate in real-world environments, through numerous functionalities and operations ranging from accupancy grid mapping to object detection algorithms [12]. Fryc et al. [13] proposed a robust multi-stage pipeline for efficient, collision-free brick picking given the pose of a target object. In this case, Octomap represented the realistic simulated environment as inputs to generate a set of motion plans for the robot. Terasawa et al. [14] presented a novel framework that combines a sampling-based planner and deep learning for faster motion planning, focusing on heuristics. For this purpose, the HM-TS-RRT algorithm obtained a heuristic map of the environment information from the Octomap for motion planning and generating collision-free paths within a reasonable time. Hence, the results based on HM-TS-RRT outperform the existing planners, especially in terms of the average planning time with smaller variance. Gai et al. developed a vision-based system for under-canopy navigation of agricultural robotic vehicles using a Time-of-Flight (ToF) camera [15]. A novel algorithm was used to detect parallel crop rows from occupancy grids taken under crop canopies. Therefore, the proposed system was able to map the crop rows with mean absolute errors (MAE) of 3.4 cm and 3.6 cm in corn and sorghum fields, where are provided lateral positioning data with MAE of 5.0 cm and 4.2 cm owing to the position in corn and sorghum crop rows. Chao and Chen [16] proposed a framework of visual perception, scenario mapping, and fruit modeling for robotic harvesters in orchard environments. The scenario mapping module applied the Octomap to represent the multiple classes of objects within the environment. The experiment results were shown that the localization and pose estimation of fruits, which are obtained at 0.955 and 0.923 values for the accuracy of visual perception and modeling algorithm. However, not many studies use real-time 3D occupancy grid mapping for benchmarking scenes. In this work, we create a real-time realistic benchmarking scene based on sensor information. This approach could help the evaluation of motion planning algorithms more accurately and obtain more reliable results.

III. Methodology

The assessment of motion planning algorithms is commonly performed using benchmarking platforms for robotic manipulators [16]. However, the outcomes can be reliable when the scene is more to reality [17]. Thus, developing methods for replicating real-work environment specifications in benchmarking scenes is crucial, e.g., the “sensed representation” of test scenes which are built from sensor information. This study established a lab-setup environment as benchmarking scene modeled by converting point cloud data from a stereo-depth camera into a 3D occupancy grid map using the Octomap algorithm [18]. The experiment was then conducted to benchmark the motion planning algorithms for the three motion problems with different levels of complexity based on the oil palmtree harvesting task. The process mentioned above, including generating the 3D occupancy grid map and the benchmarking task, was implemented sequentially. All the system components worked together simultaneously in real-time. Thus the experimental results were obtained for performance assessment of the motion planning algorithms. The overall methodology procedure for this study is indicated in Fig. 1.

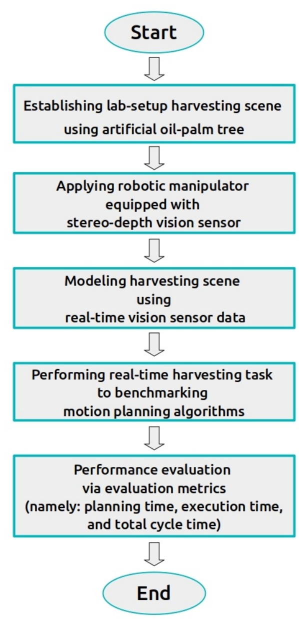


Fig. 1. Overall Methodology Procedure.

The methodology procedure of this work was performed through experimental work, as shown in Fig. 1. First, an artificial oil palm tree was utilized to establish an oil palm fruit harvesting scene. Furthermore, a robotic manipulator equipped with a stereo-depth camera (Intel RealSense D435) was used to conduct the harvesting task. The harvesting task was performed using four motion planning algorithms from the OMPL library, namely RRTConnect, BiTRRT, BFMT, and FMT. Therefore, the experimental findings evaluated the motion planning algorithms' performance via evaluation metrics, namely planning time, execution time, and total time.

A. Experiment Setup Environment

A laboratory-based experimental setup was established using an artificial palm-like tree and a four DoF robot manipulator to mimic the characteristic of a real oil palm tree harvesting environment, as shown in Fig. 2. In addition, a stereo-depth vision sensor was mounted on the robot manipulator's end-effector to model the working environment in the benchmarking scene.

(a) (b) (c)

Fig. 2. (a) Laboratory-Scale Experimental Setup with a Robot Arm Attached to a Camera and Artificial Tree (b) Real Oil Palm Tree (c) Artificial Tree.

B. Robot Arm Kinematics

This work used an open-hardware robotic manipulator, model OpenManipulator made by Robotis, as shown in Fig. 3, to develop a generic and low-cost harvesting platform [19]. This robot platform allows the users to optimize its morphology, modify the length of the links, or design the robot for their specific purposes [20]. Furthermore, the robotic manipulator has four DoF, which, based on previous studies, met the minimum requirement in terms of degree of freedom for a palm-like tree harvesting application [21].

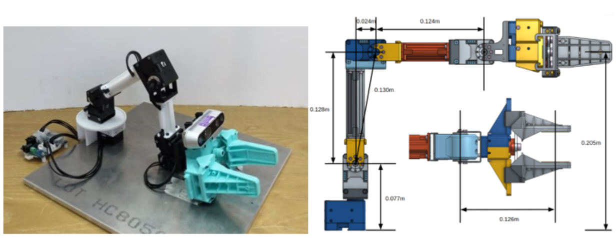


Fig. 3. OpenManipulator Robot Arm from Robotis.

Meanwhile, Fig. 4 illustrates the kinematics model of the robot manipulator, which was utilized to obtain the Denavit-Hartenberg parameters convention [22], as depicted in Table I. Besides, the robot manipulator was also modeled within Robot Operating System (ROS) platform using a domain-specific modeling language called Unified Robotic Description Format (URDF). The robot model was stored in a URDF file to represent the properties of the robot in the ROS Visualization (RVIZ) platform [18]. The URDF file format is based on XML language, which allows for encoding the robot's components such as links, joints, shapes, and physical appearance, as shown in Fig. 5.

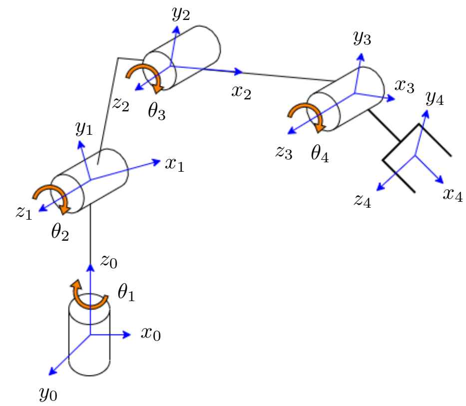


Fig. 4. Kinematic Model of Robotis OpenManipulator Robot Manipulator.

TABLE I. DH PARAMETERS FOR ROBOTIS OPENMANIPULATOR.

Joint	θ_i (°)	d_i (m)	a_i (m)	α_i (°)
1	θ_1	L_1	0	90
2	$\theta_2 + (90° - \theta_0)$	0	L_6	0
3	$\theta_3 - (90° - \theta_0)$	0	L_4	0
4	θ_4	0	L_5	90

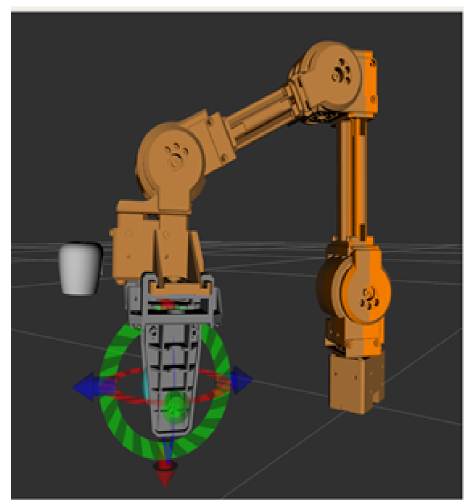


Fig. 5. Robotis Open-Manipulator Model in RVIZ Software based on its URDF File.

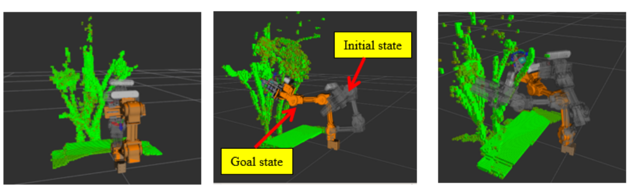
C. Modeling the Test Scene into a 3D Occupancy Grid Map

This work used an Intel Realsense D435i stereo-depth sensor, integrated with a point cloud library (PCL), to generate point cloud data [23] to replicate a real-work environment for the test scene. The camera is equipped with two left/right image sensors, OmniVision OV2740, which can produce full-high-definition (FHD) at 60 frames per second (fps). In addition, the Octomap library was implemented to convert the point cloud data of the lab-setup environment into a 3D occupancy grid map in real-time. Fig. 6 depicts a sample of a 3D occupancy grid map of the scene, including the tree leaves, which was generated using the Octomap algorithm.

Fig. 6. A Sample of a 3D Occupancy Grid Map Including the Robot Manipulator and Artificial Oil Palm Tree.

D. Generating the Harvesting Motion Planning Problems

In this benchmarking, three motion problems with different levels of complexity in terms of collision avoidance were defined based on a real oil palm harvesting task to measure the performance of the motion planning algorithms. Each of the motion problems contains a pair of initial and goal states. The initial robot arm's configuration and position are identical for all three motion problems. The gray-colored robot arm represents the initial state, while the orange-colored robot arm indicates the goal state, as shown in Fig. 7a, 7b, and 7c. The use of the gripper was beyond the scope of this work. Therefore, it is not included in these motion planning problems.



(a) (b) (c)

Fig. 7. Three Different Test Scenarios in this Work.

1) Right-Side Fruit Bunch Problem: The initial state of this problem is located in a less constrained space than the goal state, as illustrated in Fig. 8. The goal state is a stretched robot arm's configuration with the robot arm's end-effector approaching the right side of the fruit bunch from the bottom. Furthermore, the goal state represents the 'ready to cut' pose of the fruit bunch for the robot arm.

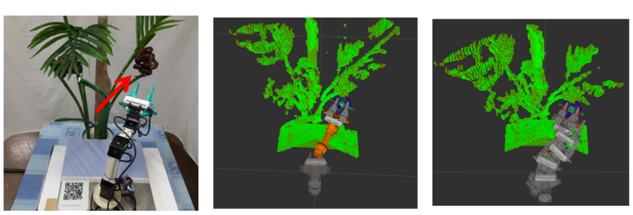


Fig. 8. Right Side of the Fruit Bunch Problem.

2) Left-Side Fruit Bunch Problem: The goal state of this problem is situated in a more constrained space than the previous motion problem due to the narrow passage that the robot arm should pass through to reach the target, as shown in Fig. 9. As in the previous motion problem, The goal state is a stretched robot arm's configuration with the robot arm's end-effector approaching the right side of the fruit bunch from the bottom. In addition, the goal state represents the 'ready to cut' pose of the fruit bunch for the robot arm.

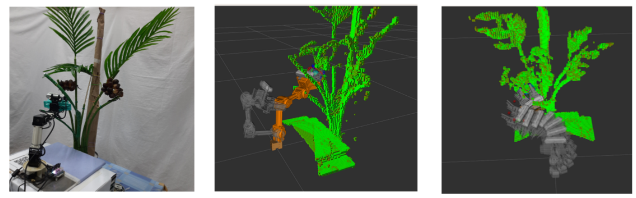


Fig. 9. Left Side of the Fruit Bunch Problem.

3) Left-Side Fruit Bunch to Right-Side Fruit Bunch Problem: This motion problem defines the left side of the fruit bunch as the initial state and the right side of the fruit bunch as the goal state, as shown in Fig. 10. Furthermore, the motion path needs to be created from a highly constrained space, traveling through a narrow passage and finally reaching another highly constrained space, as illustrated in Fig. 10. Thus, a higher level of complexity in terms of obstacle avoidance is provided by this motion problem than the previous motion problems.

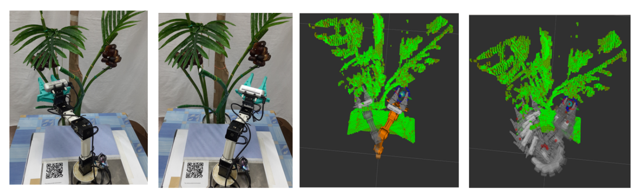


Fig. 10. Left Side to the Right Side of the Fruit Bunch Problem

E. Performance Evaluation

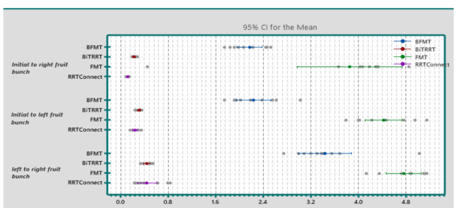
The experiment was performed within the ROS MoveIt platform on a computer with Ubuntu 18.04.6 LTS Operating System, Intel® Core^{TM} i7-9750H CPU, 16 GB Memory, Nvidia GeForce RTX 2060, and ROS Melodic distribution. The ROS MoveIt platform integrates a 3D occupancy grid map of the working environment with motion planning algorithms to generate feasible paths and solve motion problems for a given task.

In order to evaluate the performance of the motion planning algorithms for solving and executing the motion planning problems, three metrics, namely planning time, execution time, and total cycle time, were defined. All the metrics were analyzed individually to demonstrate the higher performance algorithm within each of them. Planning time is considered when it takes for a motion planner to find a viable path for a given motion problem. Furthermore, The time it takes for the robot arm to move from its initial state to its goal state for a given motion path is known as execution time. Finally, the addition of planning and execution time is considered total cycle time.

The motion planning algorithms, RRTConnect, BiTRRT, BFMT, and FMT, were implemented to solve and conduct the motion problems. For each specific problem, the problem was solved and executed 10 times by each of the algorithms for 40 runs per problem. Given the three motion problems, 120 runs were performed to conduct the benchmarking task. Furthermore, each algorithm was given 5 seconds time-out to solve the respective motion problem since such harvesting applications need to be conducted quickly in real practice. The successful cycles only were used in the analysis.

IV. Results and Discussion

Fig. 11 illustrates the results of the planning time for the three motion planning problems solved by the algorithms mentioned above. In the first problem, the right-side fruit bunch, RRTConnect achieves the shortest planning time with the lowest mean of 0.11 s and also the lowest standard deviation of 0.0215 s. In contrast, FMT had the worst performance among the others. This outcome indicates that for straight forward motion planning problems, with the minimum need for curvature to avoid colliding with obstacles, RRTConnect algorithm can be considered a fast algorithm for creating motion paths.



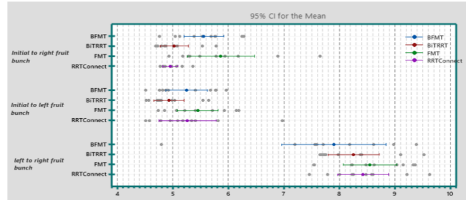
Mean	StDev	Minimum	Maximum
2.1646	0.2924	1.7465	2.5245
0.2234	0.0322	0.1838	0.2799
3.846	1.230	0.450	4.846
0.11691	0.02150	0.08145	0.14862
2.227	0.404	1.748	3.022
0.3091	0.0368	0.2417	0.3571
4.418	0.435	3.782	5.140
0.2414	0.0568	0.1524	0.3470
3.433	0.622	2.738	5.019
0.4356	0.0712	0.3314	0.5356
4.758	0.413	4.127	5.147
0.4387	0.2257	0.2325	0.8259

Fig. 11. Planning Time for the Three Planning Motion Planning Problems.

In the second motion planning problem, the left-side fruit bunch, more complexity in terms of collision avoidance was provided by the motion problem to the algorithms. In this problem, RRTConnect outperformed the other algorithms again (slightly faster than BiTRRT) with the lowest mean of 0.2414 s and the lowest standard deviation of 0.0568 s. Meanwhile, in the third motion planning problem, from the left-side fruit bunch to the right-side fruit bunch, RRTConnect had the shortest mean. In contrast, BiTRRT, which had a slightly higher mean, demonstrated a lower standard deviation than RRTConnect. Again, the FMT had the longest computation time among all.

Meanwhile, Fig. 12 represents the execution time of motion planning generated by the algorithms within the three motion problems. For example, in the first problem, from the initial position to the right-side fruit bunch, RRTConnect achieved the fastest execution time with the lowest mean of 4.9455 s and the lowest standard deviation of 0.1932 s. Meanwhile, FMT was the slowest, with the highest mean and highest standard deviation with more outliers than other algorithms.

In the second problem, from the initial position to the left-side fruit bunch position, RRTConnect achieved the lowest mean of 5.257 s while obtaining the highest standard deviation

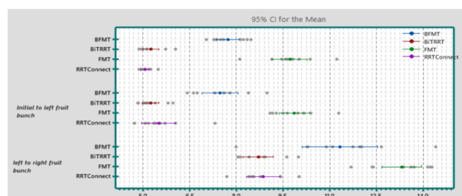


Mean	StDev	Minimum	Maximum
5.552	0.495	4.759	6.271
5.020	0.358	4.689	5.775
5.856	0.850	4.927	7.645
4.9455	0.1932	4.7601	5.3549
5.242	0.529	4.501	5.960
4.928	0.382	4.521	5.636
5.441	0.521	4.737	6.179
5.257	0.718	4.503	6.965
7.893	1.313	4.787	9.382
8.253	0.641	7.645	9.523
8.547	0.678	7.527	9.362
8.419	0.642	7.451	9.622

Fig. 12. Execution Time for the Three Planning Motion Planning Problems.

of 0.718 s. In contrast, FMT had the highest mean of 5.441 s and the most significant number of outliers. In the third problem, from the left side to the right side fruit bunch, the most complicated motion within this benchmarking work was provided to the motion planning algorithms. The lowest mean of 7.893 s and the highest standard deviation of 1.313 s was achieved by the BFMT. In contrast, FMT obtained the highest mean of 8.547 s but a lower standard deviation for the motions execution time.

Furthermore, Fig. 13 shows the results for the total cycle time, including motion planning times and the respective motion execution times. In the first problem, the right-side fruit brunch, RRTConnect achieved the highest performance in motion planning time and the respective execution time with the mean of 5.062 s and standard deviation of 0.1889 s. In contrast, the lowest performance was obtained by FMT with a mean of 9.702) and a standard deviation of 0.803 s.



Mean	StDev	Minimum	Maximum
7.717	0.519	7.025	8.459
5.244	0.367	4.873	6.044
9.702	0.803	8.095	11.213
5.0624	0.1889	4.8670	5.4816
7.469	0.796	6.408	8.982
5.237	0.374	4.834	5.956
9.859	0.676	9.080	11.278
4.998	1.803	0.340	7.312
11.326	1.696	7.980	14.401
8.689	0.683	8.050	9.992
13.305	0.890	11.670	14.271
8.858	0.740	7.684	10.004

Fig. 13. Total Cycle Time for the Three Planning Motion Planning Problems.

In the second problem, despite the lowest mean of 4.998 s for RRTConnect, its standard deviation is significantly higher than the second-lowest mean, which BiTRRT achieved. To this extent, BiTRRT can be considered a more consistent algorithm for such motion problems. In the third problem, where a more complicated motion problem than the first and the second one was provided, BiTRRT achieved the highest performance with the lowest mean and standard deviation of 8.689 s and 0.683 s, respectively. In contrast, FMT demonstrated the lowest performance concerning its mean of 13.305 s and a number of outliers.

Due to the smaller size of the palm-like artificial tree, which was used in our experimental setup, compared to actual palm-like trees, the Octomap resolution was increased. Thus, the generated 3D occupancy map contains all the necessary details of the artificial tree. However, the increase in resolution would result in high computational cost leading to a rise in the planning and execution time accordingly. On the other hand, with regard to the much larger size of real palm-like trees, which would result in less computational cost for generating a 3D occupancy map with lower resolution, a considerable decrease in motion planning time and execution time is expected for real palm-like tree harvesting applications.

V. Conclusion

In this study, a benchmarking of OMPL motion planning algorithms for a robotic harvester of a palm-like tree application was performed using the ROS and MoveIt platform. An experimental harvesting application setup was established using an artificial palm tree and a four DoF robotic manipulator equipped with a stereo-depth sensor. A 3D occupancy map was constructed using the Octomap algorithm to replicate the features and characteristics of the working environment based on the point cloud data produced by the stereo-depth sensor, which is imported into the benchmarking scene. The benchmarking was then performed within three harvesting scenarios, each including a motion planning problem with different levels of complexity, all as real-time experimental work. The motion planning performance was studied by defining three evaluation metrics: planning time, execution time, and total cycle time. RRTConnect demonstrated the highest performance in the first harvesting scenario according to the outcomes. However, for the second and third scenarios, BiTRRT outperformed the other algorithms. The work presented in this study can be extended to include methods for optimally configuring the OMPL motion planning algorithms based on the features and characteristics of a palm-like tree harvesting application to achieve the highest performance for a given motion planning algorithm.

Acknowledgment

This work was financially supported by Universiti Kebangsaan Malaysia (grant no. GUP-2019-018 and GGPM-2019-051).

References

[1] H. Zhou, X. Wang, and C. Chen, "Intelligent robots for fruit harvesting: Recent developments and future challenges background," 2021, Available online: https://doi.org/10.21203/rs.3.rs-497056/v1 [accessed on 1 June 2022].

[2] D. Sepulveda, R. Fernandez, E. Navas, M. Armada, and P. Gonzalez-De-Santos, "Robotic aubergine harvesting using dual-arm manipulation," IEEE Access, vol. 8, pp. 121889–121904, 2020.

[3] L. van Herck, P. Kurtser, L. Wittemans, and Y. Edan, "Crop design for improved robotic harvesting: A case study of sweet pepper harvesting," Biosyst. Eng., vol. 192, pp. 294–308, 2020.

[4] C. W. Bac, E. J. van Henten, J. Hemming, and Y. Edan, "Harvesting Robots for High-value Crops: State-of-the-art Review and Challenges Ahead," J. F. Robot., vol. 31, no. 6, pp. 888–911, 2014.

[5] M. Moll, I. A. Sucan, and L. E. Kavraki, "Benchmarking Motion Planning Algorithms: An Extensible Infrastructure for Analysis and Visualization," IEEE Robot. Autom. Mag., vol. 22, no. 3, pp. 96–102, 2015.

[6] D. Youakim and P. Ridao, "Motion planning survey for autonomous mobile manipulators underwater manipulator case study," Rob. Auton. Syst., vol. 107, pp. 20–44, 2018.

[7] T. F. Iversen and L. P. Ellekilde, "Benchmarking motion planning algorithms for bin-picking applications," Ind. Rob., vol. 44, no. 2, pp. 189–197, 2017.

[8] A. S. Morgan, K. Hang, W. G. Bircher, F. M. Alladkani, A. Gandhi, B. Calli, and A. M. Dollar, "Benchmarking cluttered robot pick-and-place manipulation with the box and blocks test," IEEE Robot. Autom. Lett., vol. 5, no. 2, pp. 454–461, 2020.

[9] K. Chatzilygeroudis, B. Fichera, I. Lauzana, F. Bu, K. Yao, F. Khadivar, and A. Billard, "Benchmark for bimanual robotic manipulation of semi-deformable objects," IEEE Robot. Autom. Lett., vol. 5, no. 2, pp. 2443–2450, 2020.

[10] F. Jedrzejczyk, J. Bajer, G. Gawdzik, J. Glowka, and A. Sprońska, "Benchmark and analysis of path planning algorithms of 'ROS MoveIt!' for pick and place task in tomato harvesting," Adv. Intell. Syst. Comput., vol. 1390, pp. 272–284, 2021.

[11] S. A. Magalhães, F. N. dos Santos, R. C. Martins, L. F. Rocha, and J. Brito, "Path planning algorithms benchmarking for grapevines pruning and monitoring," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 11805 LNAI, pp. 295–306, 2019.

[12] C. Premebida, R. Ambrus, and Z.-C. Marton, "Intelligent robotic perception systems," Appl. Mob. Robot., 2019.

[13] S. Fryc, L. Liu, and T. Vidal-Calleja, "Robust pipeline for mobile brick picking," Australas. Conf. Robot. Autom. ACRA, 2019.

[14] R. Terasawa, Y. Ariki, T. Narihira, T. Tsuboi, and K. Nagasaka, "3D-CNN based heuristic guided task-space planner for faster motion planning," in Proceedings of the IEEE Int. Conf. Robot. Autom., pp. 9548–9554, 2020.

[15] J. Gai, L. Xiang, and L. Tang, "Using a depth camera for crop row detection and mapping for under-canopy navigation of agricultural robotic vehicle," Comput. Electron. Agric., vol. 188, no. June, p. 106301, 2021.

[16] H. Kang and C. Chen, "Visual perception and modelling in unstructured orchard for apple harvesting robots," pp. 1-23, 2019, Available online: http://arxiv.org/abs/1912.12555 [accessed on 1 June 2022].

[17] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," Auton. Robots, vol. 34, no. 3, pp. 189–206, 2013.

[18] X. Ling, Y. Zhao, L. Gong, C. Liu, and T. Wang, "Dual-arm cooperation and implementing for robotic harvesting tomato using binocular vision," Rob. Auton. Syst., vol. 114, pp. 134–143, 2019.

[19] M. A. M. Adzeman, M. H. M. Zaman, M. F. Nasir, M. F. Ibrahim, and S. M. Mustaza, "Kinematic modeling of a low cost 4 DOF robot arm system," Int. J. Emerg. Trends Eng. Res., vol. 8, no. 10, pp. 6828-6834, 2020.

[20] M. H. M. Zaman, M. F. Ibrahim, and A. M. Moubark, "Dimensional optimization of 4-DOF robot manipulator using artificial bee colony algorithm," in Proceedings of the Int. Conf. Electr. Eng. Informatics, pp. 5-8, 2021.

[21] A. A. Aljanobi, S. A. Al-Hamed, and S. A. Al-Suhaibani, "A setup of mobile robotic unit for fruit harvesting', in Proceedings of the 19th Int. Work. Robot. Alpe-Adria-Danube Reg., pp. 105–108, 2010.

[22] H. Z. Ting, M. H. M. Zaman, M. F. Ibrahim, and A. M. Moubark, "Kinematic analysis for trajectory planning of open-source 4-DoF robot arm," Int. J. Adv. Comput. Sci. Appl., vol. 12, no. 6, pp. 769-777, 2021.

[23] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in Proceedings of the IEEE Int. Conf. Robot. Autom., pp. 1-4, 2011.