Hierarchical Planning for Autonomous Excavator on Material Loading Tasks

Liyang Wang[1], Zhixian Ye[1] and Liangjun Zhang[1]

[1]Baidu Research Institute, Baidu USA, Sunnyvale, California

liyangwang, zhixianye, liangjunzhang@baidu.com

Abstract -

Autonomous excavator develops rapidly in recent years as a result of the shortage of labor and hazardous working environments for operating excavators. We present a novel hierarchical planning system for autonomous excavators. The overall planning system consists of a high-level task planner for task division and base movement planning, and general sub-task planners with motion primitives, which include both arm and base movement. Using the proposed system architecture, we experiment the trench and pile removal tasks in the real world and experiment large-scale material loading tasks in a simulation environment. The results show that the system architecture and planner algorithms are able to generate effective task and motion plans which perform well in autonomous excavation.

Keywords -

Task planning; Autonomous excavator; Planning architecture; Material loading

1 Introduction

Excavators are widely used for many different applications such as moving earth, rock, or other materials. As one of the most versatile heavy equipment, it has a vast market over the world [1]. However, a skillful excavator human operator needs much training [2]. At the same time, many operation sites are located in remote areas with less convenient infrastructures. Moreover, hazardous work environments can impact the health and safety of the human operators on-site [3]. The autonomous excavator has the advantage of addressing these challenges and improving the overall working condition. In recent years, researchers in both academia and industry put more and more effort into developing autonomous excavators [4].

A major challenge for developing autonomous excavators is to design a general planning architecture that is suitable for a wide range of real-world tasks, such as material loading, trench digging, truck loading. In most of the literature, the authors focus on developing key individual components for autonomous excavators, including high-level task planner design, excavation trajectory generation algorithms, and control modules.

As for the high-level task planner design, some research has covered autonomous excavation in task division or base move route planning. In [5], Seo et al. devise a task planner to create optimized work commands for an automated excavator. The key components of the excavation task planner are the modules for work area partitioning and excavation path generation. In [6], Kim et al. present the intelligent navigation strategies, which are essential for an automated earthwork system to execute excavation effectively. In [7], Kim et al. present a complete coverage path planning (CCPP) algorithm by considering the characteristics of earthwork. Also, a path similarity analysis method was proposed to test the effectiveness of the CCPP algorithm.

Other research has contributed to developing the excavator arm motion generation and controller design [8] [9]. In [10], Jud et al. present a planning and control approach for autonomous excavation in which a single dig cycle is defined with respect to the end-effector force-torque trajectory. Compared with the position trajectories for the bucket motion, it has the advantage that it can overcome the limitations of that soil interaction forces, which are dominant and immensely hard to predict or estimate. In [11], Son et al. propose a novel excavation trajectory planning framework for industrial autonomous robotic excavators, which emulates the strategies of human operators to optimize the excavation of different types of soils while also upholding robustness and safety in practice.

Despite these advances, there is less research focusing on the planning architecture that connects the high-level task planner, sub-task planners and motion planning for autonomous excavators. In [12], Elezaby et al. present an event-based finite-state machine strategy model which selects different motion primitives for the wheel loader task cycle. It provides a promising system architecture design, but the motion primitive definition is based on a wheel loader and does not fit the excavator very well. Furthermore, no related sub-task definition exists in the published work and within which the architecture of the task planner is general. Most research mentioned above mostly focuses either on the high-level task planning, or the motion primitive's design. Overall, there is no from top to bottom planning system architecture design for autonomous excavators.

To address the limitations, in this paper, we present a novel hierarchical planning architecture for autonomous excavator systems. We first handle the issue of the vast variety of excavation task types and design a high-level task planner for excavation task division and base move planning. Then we abstract out two types of sub-task planners with good portability and generality, including material removal sub-task planner, and base move sub-task planner. Next we encapsulate the motion primitives and furnish them with suitable controllers for both joints control and base control. Finally, we implement our approach and further validate it in both real-world experiments and dynamic simulation environments with a range of excavators. The results show that the system architecture and planner algorithms are able to generate effective work plans that could be fed into the autonomous excavation system in a general way.

This paper is organized as follows: In Section 2, a novel general planning system architecture is presented. In Section 3, a high-level task planner is proposed in terms of excavation task division and base move planning. We first give our definition of the local task region. Then three common tasks and their planning schemes are introduced in detail. In Section 4, we focus on the implementation of sub-task planners, which include both arm movement and base movement. Their related motion primitives are covered there as well. Both real-world experimental results and simulation result is presented in Section 5. In Section 6, conclusions are drawn, and potential future work is discussed lastly.

2 Planner Architecture

In our approach, we propose a hierarchical planner architecture regarding the general applications. There are two levels of task planners plus one level of motion primitives. From top to bottom, they are high-level task planner layer, sub-task planners layer and motion primitives layer. In most scenarios, the excavator alternates between the motion of its arm to perform excavation operation and the moving of the base to the desired position. Based on this characteristic, our planner separate the material removing and base moving into two planning pipelines.

Generally, there exist a variety of user-defined tasks and it causes the high-level task planner to be highly customized. In our design, we define two types of sub-tasks. And the high-level task planner, being closest to the user interface, only needs to take the user task as input and then divide the whole task into these types of sub-tasks, namely material removal sub-tasks and base move sub-tasks. As a result, the planner plays the role of determining which location the excavator needs to move to and which region of material the excavator needs to dig. For each excavation task, it will determine the sequence of excavator route points first, and at each excavator route point, a sequence of digging regions (sub-task regions) is determined. Once all sub-tasks are done by following the generated sequence, the task assigned by the user is considered finished. For the arm movements task, the high-level task planner assumes the excavator base is stationary, and calculates the sub-task region (the material region) that can be reached by the excavator's arm at that fixed position. For the base movements task, the planner requires the excavator to reach the desired route point with a given heading angle.

Given this high-level task planner and its result of sub-tasks, sub-task planners provide the solution to achieve the short-term goal. It guides the excavator to reach the target route point through waypoints planning. Meanwhile, it helps the excavator complete the sub-region excavation efficiently and accurately at a fixed base position. For simplicity, we name material removal sub-task planner as MRSP, and name base move sub-task planner as BMSP. MRSP receives sub-task region from high-level task planner, then decides the motion primitive and calculates the relevant parameters. BMSP receives the excavator route points from the high-level task planner, and outputs the waypoints between two route points and the relevant constraints.

We abstract the motion primitives as the result of sub-task planner, instead of direct motion control, mainly because there are many repeated and modular operation actions in excavation job, which simplifies the complexity of sub-task planners. Currently, we have designed dig motion primitive, dump motion primitive and base move motion primitive. Besides the task planning results, these primitives will use several external excavation parameters to generate the feasible trajectories. For example, the excavation parameters can be the length of each excavator link, bucket volume, material remove sub-task region size, desired dig depth, and bucket teeth position constraints. The generated trajectories will be the output of the entire planning system, but the input for the controllers to follow.

In summary, we have the high-level task planner to calculate the global stop route points, local task regions, and local task goals of each stop station. And sub-task planners MRSP and BMSP will handle these sub-tasks by calling the encapsulated motion modules, which are the motion primitives we design. The motion primitives will finally generate the trajectories for controllers. The overall architecture of our planner is shown in Fig. 1.

3 High-level Task Planner

The high-level task planner takes the user-defined task as input and divides the task for the sub-task planners. Thus, in this section, we will cover about this division in terms of local task region. Also, for better understanding

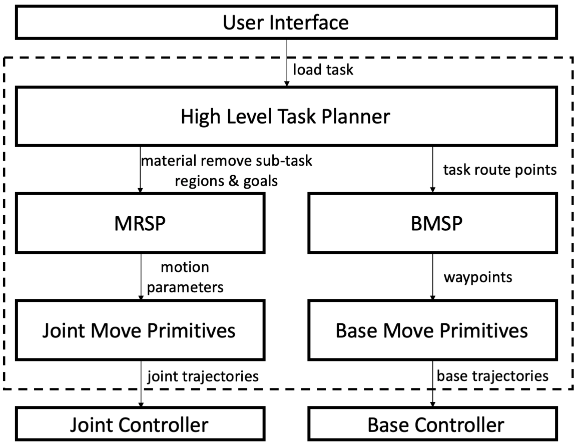

Figure 1. Planner architecture. Components are drawn inside the dashed box.

the use of high-level task planner, three common tasks in real constructions are developed, which are the trench task, pile removal task and large-scale pile removal task.

Here we define two different types of excavation task based on the moving direction and the requirement of the material residue: forward-movement tasks and backward-movement tasks. For forward-movement tasks, the target material is above the surface ground, which means that the goal height of the material removal is the same as that of the surface ground. Only after the material in front of the excavator is removed, can the excavator forward to continue. On the other hand, for backward-movement tasks, the goal height of material is below surface ground, after closer materials are removed, the excavator moves backward to continue. Apparently, the trench task is a backward-movement task, while pile removal and large scale pile removal are forward-movement tasks.

For a given excavator kinematic model, the local safely reachable range is pre-determined. We denote the maximum reachable distance as r_{max}, the minimum reachable distance as r_{min}. To fully cover the global task zone, we set a local task region overlap area, denoted as r_o.

3.1 Local Task Region

Before introducing the three specified common tasks, this section gives the definition of the local task region for MRSP. The task region of MRSP is defined locally in the excavator's base coordinate frame. The local task region is defined using four parameters: 1) area center swing angle α; 2) area angle range or width β; 3) near end distance r_{min}; 4) remote end distance r_{max}. Two types of local task region are provided. The high-level task planner can select sector area or rectangle area as local task region. The parameter β has two definitions. Angle range definition corresponds to sector area, while width definition corresponds to rectangle area. Fig. 2 shows the local task region definition using the parameters.

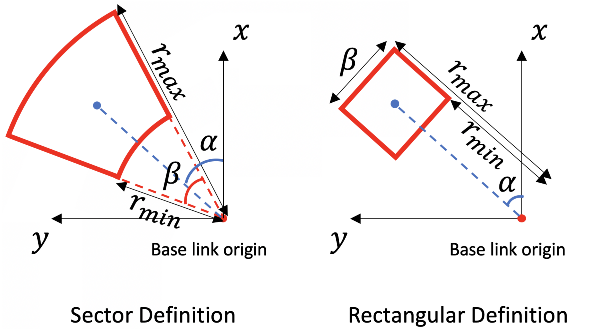

Figure 2. Local task region definition.

There are two points worth noting about the local task region. First, the selection of r_{min} and r_{max} should be based on excavator size to make sure the local task region is reachable by the bucket. Second, during the dig and dump loop, the excavator base pose may have a small change compared to the initial base pose. If the local task region is not updated, the global task zone in the map coordinate frame will change. To solve this problem, MRSP records the task region center in the map coordinate frame initially. Before each dig dump loop, MRSP will check the current excavator base pose in the map coordinate frame, and then adjust the local task region parameters to ensure the local task region does not shift in the global task zone.

3.2 Trench Task

Suppose the desired trench has a length of l, and width of w. The initial route point of the excavator is located along the trench direction, r_{max} meters from the trench beginning. After each sub-task finishes, the excavator base moves backward for the next sub-task. The backward distance $d = r_{max} - r_{min} - r_o$. The total number n of material remove sub-task execution is $n = \lceil l/d \rceil$. To meet the trench length requirement, the near end boundary distance of the last sub-task is $r_{min} + n \cdot d - l$.

In general, the desired trench width w is relatively narrow. We choose rectangle as the sub-task region shape. Fig. 3 shows the high-level planning for trench tasks.

3.3 Pile Removal Task

Pile removal task and trench task share similar definitions and have the same notations to describe the planner. The difference is trench task is a backward-movement task, while the pile removal task is a forward-movement task. Similar to the trench task, suppose the desired pile to remove has a length of l, and width of w. The initial route point of the excavator is located along the pile length

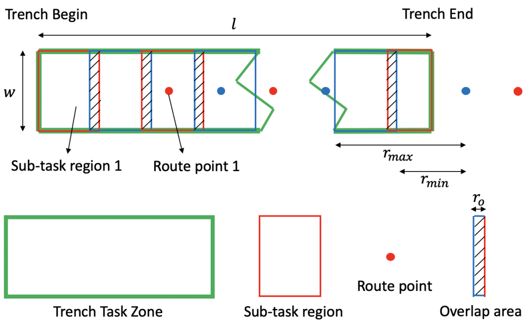

Figure 3. Trench task high level planning.

direction, r_{max} meters back from the pile task zone beginning. After each sub-task finishes, the excavator base moves forward for the next sub-task. The backward distance $d = r_{max} - r_{min} - r_o$. The total number n of material remove sub-task execution is $n = \lceil l/d \rceil$. To meet the pile length requirement, the remote end boundary distance of the last sub-task is $r_{min} + l - (n-1) \cdot d$.

Pile task zone has a relatively wide width w. We choose sector as the sub-task region shape. If the rectangle task region applied, the top left corner or top right corner may not reachable for the excavator. Fig. 4 shows the high-level planning for pile removal task.

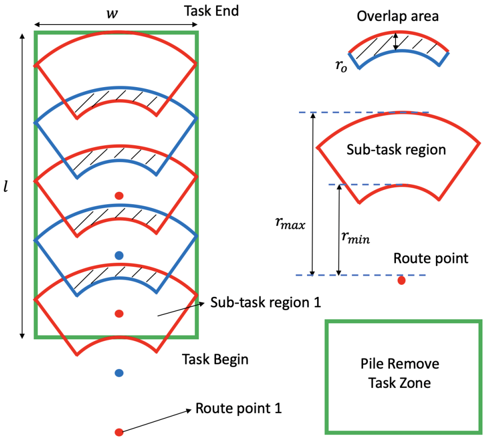

Figure 4. Pile removal task high level planning.

3.4 Large Scale Pile Removal Task

The task zone definition of the large-scale pile removal task is similar to the one of the pile removal task. However, the task zone area is larger than the pile removal task zone that the excavator route path can not be a straight line to finish the task. We denote the maximum width of the pile removal task as q. If the width w of the task zone is larger than q, the task is a large-scale case. Otherwise, the task is considered as pile removal task which described in section 3.3. According to [5], we have:

$$q = 2\sqrt{r_{max}^2 - d^2} \quad (1)$$

Suppose the task zone width w is larger than q, we first separate the task zone into $m = \lceil w/q \rceil$ columns. For each column, we use 180° sector to cover. Consider the limitation of the lidar horizontal field of view in the real application, each 180° sector is further decomposed into a smaller angle range of sectors. The sectors close to each other have an overlap area to secure full coverage. In some applications where the material may be very thick, then multi-layers can be used for sub-task division [5].

Fig. 5 gives an example of large-scale pile remove sub-task regions decomposition. At each route point, the 180° sector is decomposed into 6 parts, and 4 layers are needed to cover whole material. The material-remove sub-tasks sequence of this route point is set to finish the first layer, and sub-region from part 1 to 6. Then move to the next layer until finishing all of it.

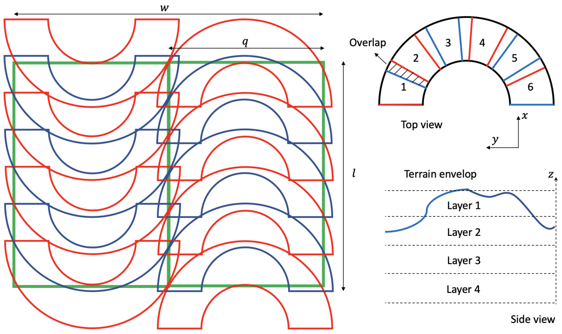

Figure 5. An example of large scale pile remove sub-task regions decomposition.

Between two columns, we design the connection path using a semicircular arc. Since the excavator work area usually has soft ground surfaces, using a small turning radius may cause the excavator to sink into the ground. Fig. 6 shows an example excavator base path planning for the large-scale pile removal task. The initial route point locates at the bottom right corner. The excavator moves straight to the top to finish the first column. Then the excavator moves to the route point marked by red, which represents the route point not associated with a material remove sub task. After that, the excavator follows the semicircular arc as a U-turn to the second column. Finally, the excavator moves straight down to finish the second column.

4 Sub-Task Planners and Motion Primitives

As mentioned in Section 2, we design two sub-tasks planners. Material remove sub-task planner plans dig ma-

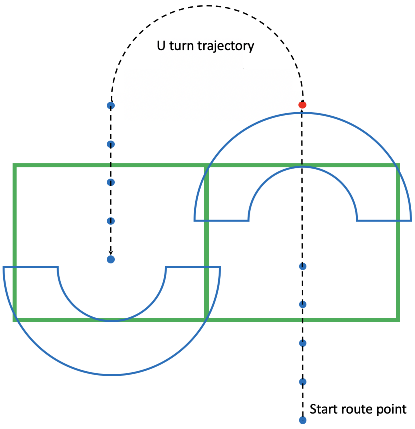

Figure 6. An example of large scale pile remove sub-task base move planning.

terials to reach the desired amount and dump them into given positions. While base move sub-task planner is able to navigate the excavator to reach desired position and heading angle. So in following sub-sections, we will bring up the design details of these sub-task planners. After that, we will cover about the motion primitives that these sub-task planners are based on.

4.1 Material Removal Sub-Task Planner

The input of MRSP contains two parts: (1) sub-task configuration from the high-level task planner; (2) states information needed for online planning. Sub-task configuration includes local task zone, target material height, and dump mode. And we use excavator states (such as excavator joint states and base pose) and environment states (LiDAR point cloud) as our states representation. The output of the MRSP is the motion parameters for our motion primitives.

We have shown the procedures of the material-remove sub-task module in Algorithm 1. MRSP first loads the sub-task configurations. Then it plans the dig-and-dump loop until the material height in the local task zone reaches the target height. The dump position is decided depending on the dump mode.

4.1.1 Point of Attack Planning

MRSP finds the Point of Attack (POA) based on the LiDAR point clouds within the local task zone. In this approach, we determine POA based on the highest point with offset ahead. Fig. 7 shows the method to determine POA.

MRSP first finds the highest point and average height of the local task zone. Then, starting from the base link origin, a straight line that connects the highest point is determined. As shown in the top view, a constant distance offset is added to the line and the (x, y) of POA is determined. As shown in the side view, the z of POA is set equal to the average height.

Algorithm 1 Material remove sub-task algorithm

```
procedure MRSP PROCEDURE
    set sub-task region.
    set sub-task goal.
    set soil dump mode.
loop:
    if goal reach condition meet then
        end loop.
    else
        find point of attack.
        plan dig motion and perform.
        plan dump motion and perform.
end procedure
```

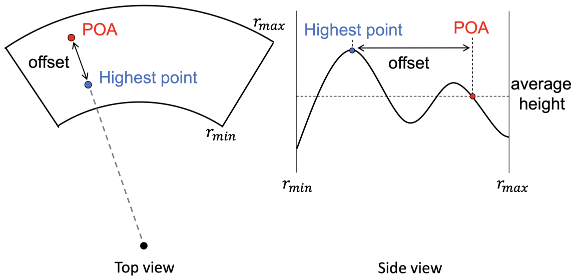

Figure 7. The method to determine POA

4.1.2 Dig Motion Parameters

Dig motion planner first gets POA for the bucket teeth end to contact. Then the dig motion is divided into three steps. First of all, the bucket penetrates into the material with some depth and distance closer to the base origin. Secondly, the bucket drags backward to the base origin to collect materials into the bucket. At last, the bucket closes until the bucket joint to teeth plane is flat, which prevents material leak in the following motion. Based on the discussion above, a dig motion is defined by 7 parameters. We denote the dig motion parameters as $D = [x_b, y_b, z_b, p_d, p_l, d_l, \delta]$, where x_b, y_b, z_b represent the POA in base frame, p_d represents penetration depth, p_l represents penetration length, d_l represents drag length, and δ represents entire dig motion tilt angle with respect to horizon plane. Fig. 8 shows the dig motion parameters definition.

In our approach, these parameters are determined according to the terrain shape to optimize the material volume collected to match the bucket volume. The final dig motion trajectory is further adjusted based on the excavator base pose to handle the case that the roll and pitch angle of the base is not zero.

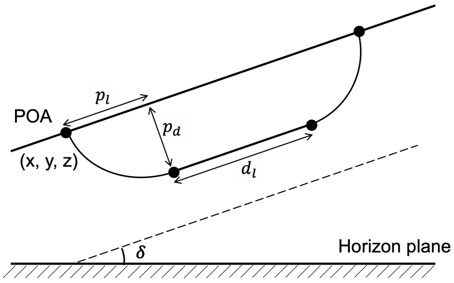

Figure 8. Dig motion parameters definition.

4.1.3 Dump Modes

For universality, MRSP provides three different dump modes. The first mode is using a fixed dump point in the base frame of the excavator. The MRSP dumps to the same point every dig-dump cycles. The second mode is using a floating dump point in the base frame. This floating dump point is passed as the output of high-level task planner for the MRSP, which can be used in the dynamic working environment. The third mode is truck-dump mode. In this mode, the MRSP will subscribe the truck pose through perception module, and calculate the appropriate dump point itself. To make the truck loading of material evenly, MRSP will find the dump spot, where material height is low inside the truck bed, to dump.

4.1.4 End Condition

As for end condition, MRSP is designed to have three stopping conditions in favor of different task excavation strictness requirement, namely "rigid", "regular" and "loose".

We denote the goal height as g, the current highest point in the sub-task zone as h, and the current average height of the sub-task zone as a. For "rigid" mode, the finish triggering condition is $h < g$. For "regular" mode, the triggering condition is $a < g$ and $h < g + b$, where b is a positive value, which represents a height margin of the highest point. For "loose" mode, the finish triggering condition is $a < g$.

4.2 Base Move Sub-Task Planner

The base-move sub-task planner navigates the excavator move to the target pose assigned by the high-level task planner. Similar to MRSP, the input of the base-move subtask planner consists of both the target route point from high-level task planner and the states information required for online planning. The states information includes the excavator base pose and the LiDAR point cloud. While the outputs of the BMSP are waypoints between current route point and next route point as well as relevant walking constraints.

Currently, our BMSP is developed based on the 2-D assumption. So we denote the 2-D target pose as $B = [x_m, y_m, \Theta]$, where (x_m, y_m) is the target position in the map frame, and Θ is the target heading angle in the map frame.

As for global path planning, the Hybrid A* Path Planner algorithm [13] is applied in BMSP. With occupancy map generated by the LiDAR point cloud, current base pose, and target base pose, a smooth collision-free path of the waypoints are generated. In this work, we use the unicycle model[14] as our excavator kinetic model for the MPC controller[15].

4.3 Motion Primitives

Motion primitives are encapsulated from the repeated excavator action, such as digging, dumping and moving base. Therefore, we have designed the dig motion primitive, dump motion primitive and base move motion primitive for our framework.

Joint move primitives, including digging and dumping, require the motion parameters or task configuration as described in section 4.1.2 and 4.1.3. These joint move primitives will calculate the joint trajectories for the PID controllers to follow and finally come up with the joint velocity control commands.

On the other hand, the base move primitive takes the path of waypoints generated by the BMSP as input. And the primitive acts as a MPC controller to generate the base control commands.

For low-level control, both velocity and base control commands will be matched to current inputs of the hydraulic system of excavator using the look-up table.

5 Experiments

We perform trench task, pile removal task in the real excavator, and perform large scale pile removal task in simulation. In the real test, the excavator used is XCMG490DK, which has a max digging radius of $10.47m$, maximum digging height $10.12m$, maximum digging depth $6.93m$, and operating weight $36.6ton$. In the simulation, the excavator used is CAT365CL, which has a max digging radius of $11.44m$, maximum digging height $11.90m$, maximum digging depth $7.17m$, and operating weight $67.0ton$. We used the AGX Dynamics [16] as simulation environment.

5.1 Trench Experiment

The target trench in experiment has a length of $10.0m$, width of $1.5m$, and depth of $2.0m$. Using the architecture and planner presented, the task finished using 20 minutes. Based on the trench size, our algorithm automatically set 5 route points and 5 material remove sub-tasks. Fig. 9 shows the on-site trench experiment. The image on the left is before the trench task started. The image on the right is after the trench task finished.

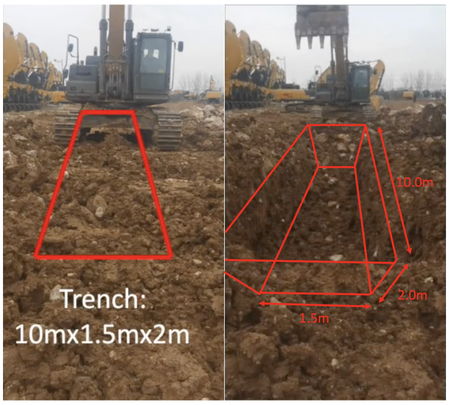

Figure 9. Trench experiment.

5.2 Pile Removal Experiment

The target pile has a length of 8.0m, width of 5.6m, and height of 0.5m. Using our planner, the task finished the task within 16 minutes. Fig. 10 shows the on-site pile removal experiment setup.

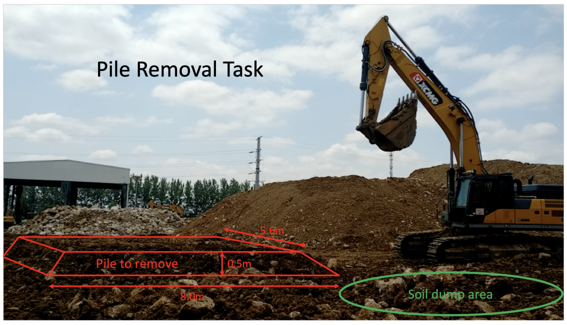

Figure 10. Pile removal experiment setup.

Fig. 11 shows 4 moments during the removal experiment setup, where the base located in different route points. Based on the pile size, our algorithm automatically set 4 route points and 4 material remove sub-tasks.

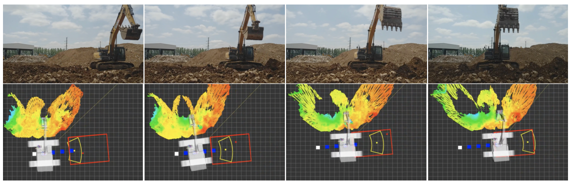

Figure 11. Pile removal experiment process.

5.3 Large Scale Pile Removal Experiment

Large scale pile removal experiment is performed in simulation. The target pile has a length of 36.0m, width of 22.5m, and height of 0.5m. To shorter the simulation time. In the AGX simulator, the pile size is set length equal to 20.0m, width equal to 12.0m, and height remains 0.5m. Fig. 12 shows the simulation results. The figures on the left show the initial simulation configuration, and the figures on the right show the environment when the simulation finished.

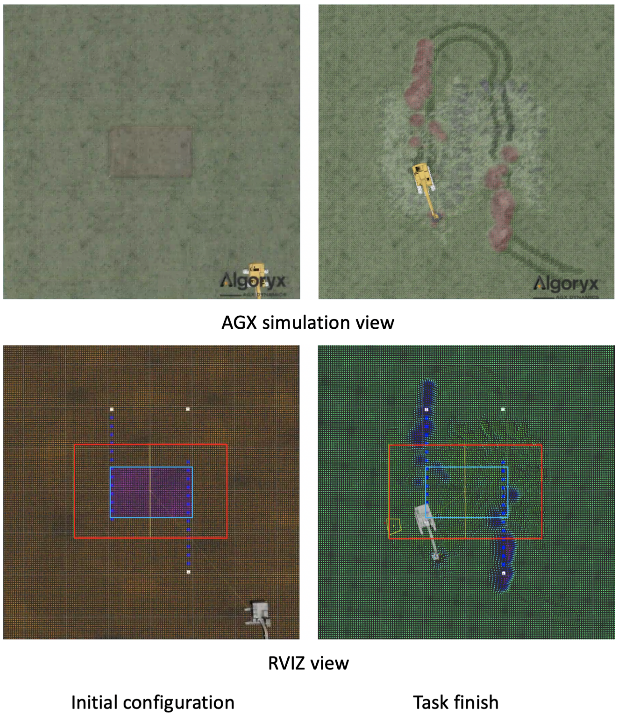

Figure 12. Large scale pile removal experiment using AGX simulator.

Initially, the excavator is located at the bottom right of the environment. It first moves to the first route point planned, and starts manipulation for the first column. Once the first column is finished, it performs a U-turn and then works on the second column. In the RVIZ view, the red box shows the task zone assigned by user, the light blue box shows the actual pile generated in the AGX simulator. The white marker points represent the route point without material-remove sub-task assigned, and blue marker points represent the route point with material remove sub-task assigned. The color shows in RVIZ is base on soil height. In the initial configuration, the color inside the light blue box is very different from the color outside. When the task finishes, the color is uniform which means the task is well done. Since the soil is dumped on the ground directly, there are some small piles locates on the excavator base move trajectory.

6 Conclusion

This paper presents a novel autonomous excavator planning system design. High-level task planning includes task division and base movement planning. General sub-task planners designed with motion primitives are proposed, which take both arm and base movement into consideration. Based on the experiment results, we conclude that using the proposed material-remove sub-task planner and base-move sub-task planner, our method is a general autonomous excavator planning system architecture that can fit different excavation tasks.

Besides excavation tasks, excavators have also been used for tasks such as surfacing the ground and other tasks. The proposed architecture can be extended to new types of tasks in the future. Currently, out base move planning is limited to 2-D. In some working environments, the excavator may be required to drive on a rugged terrain surface. Thus, another future research direction can be 3-D base move planning with consideration of excavator balancing.

References

[1] Grand View Research. Excavator market size, share & trends analysis report by product (crawler, wheeled, mini/compact), by application (construction), by region, competitive landscape, and segment forecasts, 2018 - 2025, July 2018.

[2] VS Velikanov, NV Dyorina, EI Rabina, and T Yu Zalavina. Economic assessment when creating a comfortable operational environment for a mining excavator operator. In *Journal of Physics: Conference Series*, volume 1515, page 042044. IOP Publishing, 2020.

[3] Occupational Safety, Health Administration, et al. Excavation: Hazard recognition in trenching and shoring. *OSHA technical manual*, 1999.

[4] Liangjun Zhang, Jinxin Zhao, Pinxin Long, Liyang Wang, Lingfeng Qian, Feixiang Lu, Xibin Song, and Dinesh Manocha. An autonomous excavator system for material loading tasks. *Science Robotics*, 6(55), 2021. doi:10.1126/scirobotics.abc3164. URL `https://robotics.sciencemag.org/content/6/55/eabc3164`.

[5] Jongwon Seo, Seungsoo Lee, Jeonghwan Kim, and Sung-Keun Kim. Task planner design for an automated excavation system. *Automation in Construction*, 20(7):954–966, 2011.

[6] Sung-Keun Kim, Jongwon Seo, and Jeffrey S Russell. Intelligent navigation strategies for an automated earthwork system. *Automation in Construction*, 21:132–147, 2012.

[7] Jeonghwan Kim, Dong-eun Lee, and Jongwon Seo. Task planning strategy and path similarity analysis for an autonomous excavator. *Automation in Construction*, 112:103108, 2020.

[8] Quang Ha, Miguel Santos, Quang Nguyen, David Rye, and Hugh Durrant-Whyte. Robotic excavation in construction automation. *IEEE Robotics & Automation Magazine*, 9(1):20–28, 2002.

[9] Bin Zhang, Shuang Wang, Yuting Liu, and Huayong Yang. Research on trajectory planning and autodig of hydraulic excavator. *Mathematical Problems in Engineering*, 2017, 2017.

[10] Dominic Jud, Gabriel Hottiger, Philipp Leemann, and Marco Hutter. Planning and control for autonomous excavation. *IEEE Robotics and Automation Letters*, 2(4):2151–2158, 2017.

[11] Bukun Son, ChangU Kim, Changmuk Kim, and Dongjun Lee. Expert-emulating excavation trajectory planning for autonomous robotic industrial excavator. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2656–2662. IEEE, 2020.

[12] Ahmed Adel Elezaby, Mohamed Abdelaziz, and Sabri Cetinkunt. Operator model for construction equipment. In *2008 IEEE/ASME International Conference on Mechtronic and Embedded Systems and Applications*, pages 582–585. IEEE, 2008.

[13] Karl Kurzer. Path planning in unstructured environments : A real-time hybrid a* implementation for fast and deterministic path generation for the kth research concept vehicle. Master's thesis, KTH, Integrated Transport Research Lab, ITRL, 2016.

[14] Alessandro De Luca, Giuseppe Oriolo, and Marilena Vendittelli. Control of wheeled mobile robots: An experimental overview. *Ramsete*, pages 181–226, 2001.

[15] Liyang Wang, Jinxin Zhao, and Liangjun Zhang. Navdog: robotic navigation guide dog via model predictive control and human-robot modeling. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, pages 815–818, 2021.

[16] AGX Dynamics. `https://www.algoryx.se/`, 2021. [Online; accessed 03-June-2021].