

Received 22 August 2023; revised 3 May 2024 and 27 September 2024; accepted 3 October 2024. Date of publication 24 October 2024;  
date of current version 22 November 2024.

This article was recommended by Executive Editor Sanjiv Singh.

Digital Object Identifier 10.1109/TFR.2024.3485037

# Toward Autonomous Excavation Planning

LORENZO TERENZI<sup>1</sup> AND MARCO HUTTER<sup>1</sup> (Member, IEEE)

Robotics Systems Lab, ETH Zürich, 8092 Zürich, Switzerland

CORRESPONDING AUTHOR: LORENZO TERENZI (lterenzi@ethz.ch)

This work was supported by the NCCR Digital Fabrication and Robotics, the SNF under Project 188596.

(Regular Article)

**ABSTRACT** Excavation plans are essential in construction projects, dictating the dirt disposal strategy and excavation sequence based on the final geometry and machinery available. While most construction processes rely heavily on coarse sequence planning and local execution planning driven by human expertise and intuition, fully automated planning tools are notably absent from the industry. This article introduces a fully autonomous excavation planning system. Initially, the site is mapped, followed by user selection of the desired excavation geometry. The system then invokes a global planner to determine the sequence of poses for the excavator, ensuring complete site coverage. For each pose, a local excavation planner decides how to move the soil around the machine, and a digging planner subsequently dictates the sequence of digging trajectories to complete a patch. We showcased our system by autonomously excavating the largest pit documented so far, achieving an average digging cycle time of roughly 30 s.

**INDEX TERMS** Building automation, construction (buildings), mobile robots.

## I. INTRODUCTION

THE construction industry plays a vital role in building infrastructure for essential human needs, such as shelter and transportation. However, it faces persistent challenges, including stagnant productivity growth, increasing labor costs [24], and worsening labor shortages [25]. Furthermore, construction sites pose significant safety risks, resulting in high rates of worker injuries and fatalities [3].

Automation, which has revolutionized the manufacturing sector, could provide a solution to these issues in the construction industry. In particular, the automation of earth-work tasks using excavators holds great promise due to their repetitive nature and the precision required. While the complexity of these tasks has historically presented obstacles, recent research has revealed promising opportunities for developing fully autonomous excavation systems. Controlled environments involving trenching or bulk excavation, such as excavating foundations for buildings (e.g., raft or strip foundations), large-scale mining operations, and cut-and-fill processes necessary for road construction, could benefit the most from these advancements.

In recent years, there have been significant advances in automating earthmoving tasks, such as trenching [4], [15], [20], loading materials onto trucks [33], [38], and constructing embankments [15]. However, these methods often focus on specific geometries or limited site sizes, limiting

their applicability to general excavation scenarios. While large-scale excavations have been attempted using coverage routines [6], [19], they rarely address the challenges of local soil rearrangement and heavily rely on heuristics [4], [15], [17]. Consequently, the excavation planning problem for general geometries and bulk excavation remains an open challenge in the field of autonomous earthmoving.

Excavation planning is a complex, multilevel task that involves careful consideration of both global and local factors. The global plan focuses on the overall strategy of the excavation, dictating the sequence in which the excavator moves across the site. This includes efficiently navigating obstacles, avoiding trapping the excavator, and ensuring that soil disposal does not hinder subsequent movements. Such a plan requires careful long-term planning as it is more complex than standard graph traversal problems such as the traveling salesman problem (TSP), due to the need for possibly revisiting zones multiple times and the interdependencies between different excavation zones.

The local plan complements the global strategy by managing the immediate Earth movements around the excavator's current position. It involves determining the precise excavation methods needed to achieve the desired terrain geometry while efficiently transporting soil to designated disposal locations. The local strategy must be adaptable to unexpected challenges such as spillages or structural collapses and aims



**FIGURE 1.** (a) Terrain before the excavation, the excavation area is indicated in violet. (b) Excavated pit, the sides and bottom are highlighted in violet.

to optimize the excavation process to meet design specifications with minimal adjustments.

Both planning levels are deeply interconnected; errors in the initial global planning can have cascading effects, complicating local execution and potentially halting progress. Despite the complexities, the literature provides limited guidance on integrating these global and local aspects effectively. This underscores the need for advanced research in excavation planning to develop robust strategies that ensure both efficient navigation and precise Earth manipulation, ultimately facilitating smooth and continuous operation throughout the excavation site.

This work presents a holistic solution for automating bulk excavation in uncluttered environments, as illustrated in Fig. 1. We treat global excavation planning as a coverage problem [19], dividing the workspace into coverable cells. A method is developed to find the optimal sequence of excavator base poses that allows the entire workspace to be excavated while satisfying soil disposal constraints. The order of cells is determined using graph navigation algorithms, and dynamic programming optimizes the entry and exit points of cells, which are covered using simple zig-zag navigation primitives. The local earthwork planning problem involves efficiently redistributing soil around the excavator's base before moving to a new pose. The local planner selects digging and dumping locations based on the current and desired geometry and proximity to the designated soil disposal area. Bayesian optimization is used to determine the initial condition of parametrized digging trajectories. The navigation stack employs a rapidly exploring random tree (RRT) planner and a pure pursuit controller to minimize travel distance and maximize clearance from obstacles and hazardous excavated areas. An overview of the system and its components is shown in Fig. 2.

## II. RELATED WORK

We build on prior work in research and field deployment of autonomous excavators, global earthworks planning, local earthworks planning, and trajectory planning for digging.

*Advancements in Excavator Automation:* There has been significant progress in the automation of excavators in recent

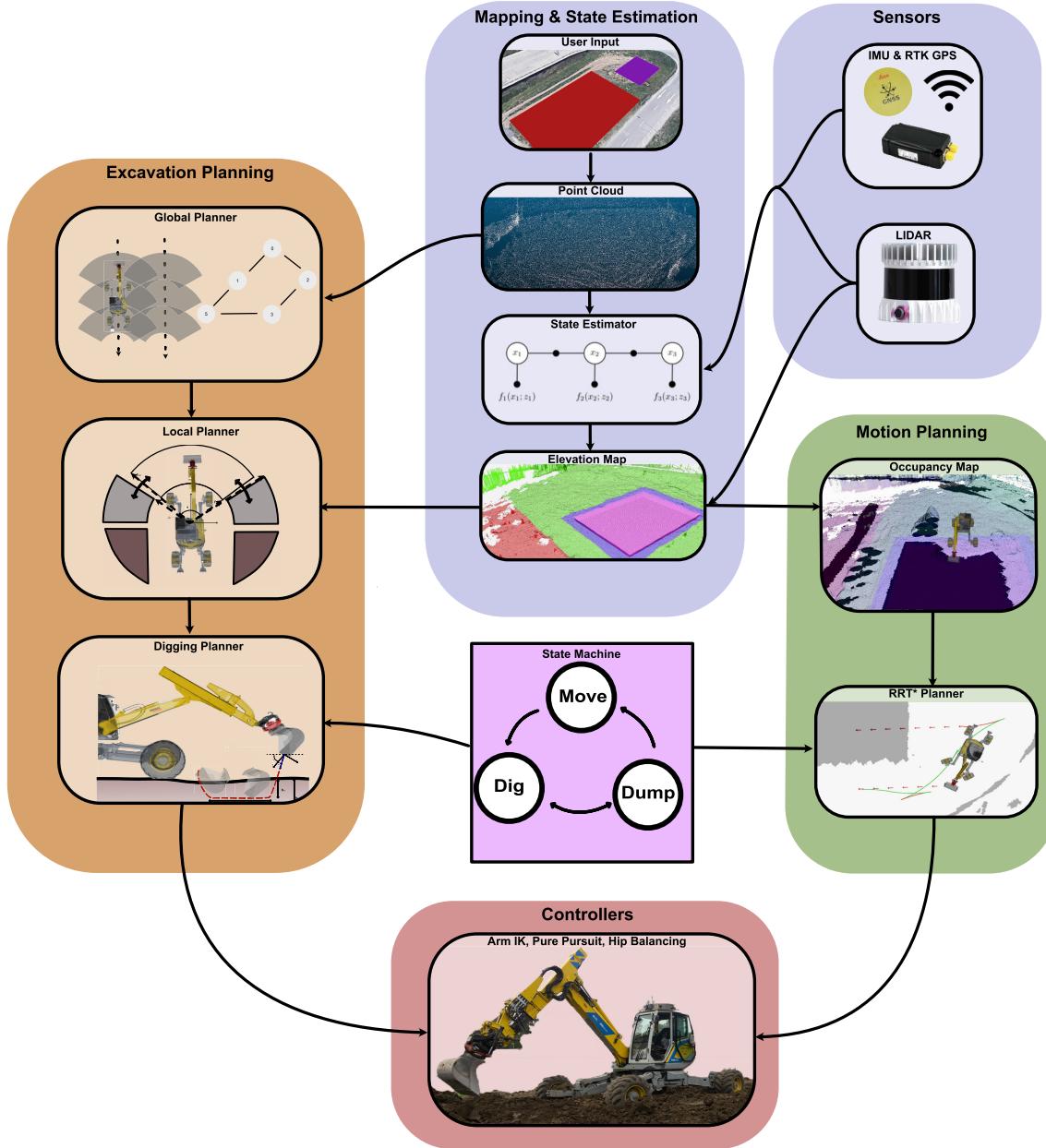
years. Early efforts, such as LUCIE [2], focused on performing dig cycles without environmental perception. More recent work has incorporated range sensors [4], [33] to enable autonomous digging in varied terrain profiles and the loading of soil onto trucks. The use of state machines and kinematic motion primitives has also been explored for autonomous trenching [9], with the addition of light detection and ranging (LiDAR)-based elevation mapping and velocity and force control for the arm in [17]. Built Robotics has recently developed an automation kit for trenching tasks [20].

Other recent research has focused on more complex excavation tasks, such as creating embankments [15]. Our work aims to extend this progress by addressing excavation tasks of different geometries and constraints on dirt disposal and navigation.

*Global Earthworks Planning:* The global earthworks planning problem involves devising a set of poses for the excavator during the excavation and determining a sequence of digging and dirt disposal areas. This problem has been addressed in a variety of ways in prior research.

In one approach, Woo et al. [36] employed a discretized workspace, using a reinforcement learning (RL) agent to generate a value function over the grid. The cell with the highest estimated value is selected for digging. However, this approach overlooks important aspects of dirt handling and fails to ensure the reachability of the subsequent dig cell. Consequently, such models may not effectively adapt when environmental conditions or the excavation site layout changes after initial digging, akin to the dynamic challenges found in games such as Sokoban where each move alters the state of the playing board.

On the other hand, Groll et al. [9] and Jud et al. [15] developed specialized navigation trajectories and soil disposal strategies for autonomous excavation of an embankment and trench, respectively. Although these systems successfully completed their specific tasks, the methods are not generalizable to excavation of arbitrary geometries. Furthermore, the approaches lack robustness to handle unexpected obstacles or changes in terrain that may arise in dynamic environments.



**FIGURE 2.** System architecture diagram. Subsystems are grouped in modules indicated in different colors: sensors (blue), mapping and state estimation (violet), excavation planning (orange), motion planning (green), and controllers (red). The state machine in the middle of the diagram triggers the execution of the different modules. The arrows represent the flow of information from one subsystem to another.

Global earthwork planning can also be viewed as a coverage problem, where the objective is to find a sequence of base poses that allow the arm to reach and excavate the entire area. Coverage path planning (CPP) algorithms have been widely used in diverse fields such as agriculture, house cleaning robots, underwater exploration, and mapping via unmanned aerial vehicles [8]. Commonly, these algorithms presuppose the space to be covered that is known beforehand and remain static, which is not the case in dynamic and changing excavation sites.

Coverage algorithms such as wavefront and spanning tree coverage prove efficient when considering discretized space. However, these can generate trajectories involving backtracking (impossible during digging due to the alteration of the terrain) and possibly complex paths, thereby complicating dirt handling. In continuous spaces, the initial step of a CPP algorithm often involves dividing the space into coverable zones using simple motion subroutines such as a standard zigzag trajectory. For this purpose, decomposition methods, such as Morse [1] and Boustrphedon [6], can be employed.

Yet, these methods may fail to adapt to on-the-fly changes in the environment, such as those caused by ongoing excavation activities.

Kim et al. [19] proposed a CPP algorithm for navigating multiple earthwork systems operating simultaneously in a known environment. This approach utilized a Morse function to partition the space and solved the TSP to compute the sequence of cells to be processed. This method, however, has several limitations, including reliance on edge adjacency to establish cell connectivity, a large set of navigation patterns, and no consideration for dirt handling.

Cao et al. [5] and Xie et al. [37] also tackled the TSP-CPP problem using unmanned autonomous vehicles for 3-D scanning. They used dynamic programming and simple subroutines for coverage. These methods assume that the environment does not change, allow revisiting observed areas, have a predefined cell connectivity structure, and have relaxed cell adjacency conditions. In contrast, the environment changes during excavation, and navigation is no longer possible in excavated areas. The cell connectivity also changes because excavating can block access to parts of the workspace. In addition, cell connectivity is more restricted, as moving between cells can only happen outside the excavation site or through corner adjacency.

*Local Earthworks Planning:* The challenge of local earthworks planning has been largely overlooked in the literature. The solution is relatively straightforward for specific tasks such as trenching, with the soil being dumped beside the trench [9]. In other works, the problem is simplified by assuming that the dirt is immediately disposed of or dumped into a truck [4].

Prior approaches to this problem have used a semicircular digging workspace in front of the excavator. This concept has been reflected in various works, such as [28] and [30].

*Digging Planners and Controllers:* Digging planners are responsible for planning the sequence of digging trajectories to complete a task within a given digging zone. The problem is typically divided into two subproblems: selecting the attack point (where the digging trajectory begins on the excavation surface) and designing a 2-D digging trajectory planner/controller to execute the trajectory given the attack point and digging plane.

Singh and Cannon [30] use expert operator heuristics to divide the digging zone into sections along the radial and tangential directions relative to the excavator base, which are dug sequentially until a certain precision is reached. However, these approaches can force the 2-D digging planner to repeatedly dig low-volume scoops until the required precision in a subsection is reached. They may also make the plan brittle to wall collapses or soil spillage during the excavation of future adjacent sections or during dirt transport to the dump zone. In contrast, Zhang et al. [38] suggest using a data-driven approach to select the attack point of the digging trajectory by learning from human operator preferences.

Son et al. [32] use dynamic motion primitives to efficiently learn from expert data, with a modulation module to adapt

the trajectory to different soil types. Lee et al. [21] use model predictive control (MPC) to plan digging trajectories in simulation. However, whether the system can bridge the sim-to-real gap without accurate soil modeling and a model of the machine's dynamics is uncertain. Egli et al. [7] successfully demonstrate the deployment of a soil-adaptive digging controller trained with RL in simulation only.

Singh and Simmons [31] introduce a methodology that resembles ours that focuses on optimizing individual digging actions, defined by parameters such as digging angle, entry height, and digging distance, using geometric and physical constraints to design trajectories. Their approach, primarily validated in 2-D simulations via simulated annealing, aims to simplify autonomous excavation planning, though its effectiveness in real-world, dynamic conditions and with complex soil mechanics has not been thoroughly tested. In contrast, our method employs the Bayesian optimization to determine optimal digging trajectories, maximizing soil extraction. Our planner also explicitly rolls out trajectories using the digging controller directly, making it adaptable to various conditions and controllers. We delegate the task of handling force constraints to the digging controller.

## A. CONTRIBUTIONS

This article presents a novel and comprehensive approach for autonomously excavating large sites using a single excavator. Demonstrated on a legged Menzi Muck M545 excavator, our framework offers a solution to specify the excavation area, constraints on navigation, and dirt disposal in georeferenced coordinates. It introduces a global workspace planner that calculates the required base poses, guaranteeing that they are collision-free and achievable. In contrast to existing methods, we propose using cell corner adjacency, as these corners represent local subroutines' start or endpoints, to facilitate easier planning.

The local workspace planner in our approach is tasked with deciding soil redistribution around the excavator without base movement. Here, our system improves upon existing designs by creating five adaptable zones in the local workspace. These zones are dynamic in size based on the required excavation geometry and serve as areas for digging and dumping soil.

The digging trajectory planner, another integral part of our system, determines the precise excavation point and trajectory. In a significant improvement over previous methods, we employ the Bayesian optimization to identify the free parameters of a digging trajectory that greedily maximizes scooped soil volume.

Our system was put to the test by digging an approximately rectangular pit, measuring  $11.6 \times 15.6 \times 1$  m, which required planning intermediate dirt dumping sites. The successful completion of this complex task, involving hundreds of load cycles and moving about 300 ton of material within half a work day, demonstrates the system's capabilities and reliability in real-world excavation scenarios.

Notably, our system holds the distinction of being the first of its kind capable of autonomously performing diverse excavation tasks such as digging trenches, pits, and handling more complex projects with dirt disposal and navigation constraints. Its development marks a significant step forward in excavation automation, moving beyond the limitations of previous systems that were constrained to simple 2-D tasks or required extensive human intervention. It is also one of the few systems to demonstrate reliable autonomous digging for hundreds of load cycles [4] and the movement of approximately 300 ton of material in a half work day.

This article's key contributions are given as follows.

- 1) A global workspace planner that decomposes the workspace using the Boustrophedon algorithm, finds the related quotient graph that encodes the connectivity of the workspace, computes the minimum branching tree of the graph, and finds the visiting sequence of the nodes with a postorder traversal to ensure the connectivity of yet-to-be-excavated areas. The planner also takes into account dumping constraints and the presence of obstacles when using dynamic programming to choose the start point and endpoint of each cell.
- 2) A local workspace planner that determines how to move dirt around the excavator without moving the base, considering user input and constraints on dirt disposal. The planner selects dig areas based on the discrepancy between the current soil geometry and the desired one and dump areas based on the distance to a user-designated dumping zone.
- 3) A digging trajectory planner that aims at reaching the target geometry in a dig area by using Bayesian optimization to find the free parameters of a digging trajectory, which greedily maximizes scooped soil volume.
- 4) A safe and robust navigation system based on a sampling planner and pure pursuit controller that allows the excavator to navigate to the next base pose while avoiding obstacles and maintaining a safe distance from the excavation site.
- 5) An experimental validation and quantitative results that set a new benchmark for autonomous excavation performance.
- 6) The introduction of a novel dataset comprising authentic building silhouettes and urban crop layouts, the development of a software program for generating realistic excavation shapes through procedural methods, and a benchmark to assess the effectiveness of the excavation planning system. All associated code and resources are open-sourced at the digbench repository.<sup>1</sup>

### III. METHOD

**HEAP:** It is an M545 Menzi Muck 12-ton legged excavator that has been adapted for autonomous forestry operations [11], rock wall construction [13], and digging tasks

<sup>1</sup>[git@github.com:leggedrobotics/digbench.git](https://github.com/leggedrobotics/digbench.git)

[7], [15]. The chassis is equipped with servo valves and pressure sensors that enable the deployment of a force-based chassis balancing controller [10], which significantly aids navigation over challenging terrain. The system is equipped with an Ouster OS-0 128 LiDAR for mapping, localization, and online site mapping during operations. A global positioning system (GPS) receiver with real-time kinematic (RTK) correction and an inertial measurement unit (IMU) is mounted on the cabin. IMUs are also mounted on each arm link to estimate the kinematic position of the shovel. An encoder measures the cabin's orientation relative to the base. For more details on the hardware setup, see [16].

**System Overview:** The deployed system integrates mapping, localization, and various planning components to enable autonomous excavation. It incorporates a user interface for excavation planning, global earthwork planning, local excavation planning, a digging planner, and a navigation sampling-based planner, as depicted in Fig. 2. To assess the effectiveness of the excavation planning system. The planning modules can use different digging controllers, ranging from straightforward kinematic controllers to advanced RL strategies.

### A. MAPPING AND STATE ESTIMATION

#### 1) MAPPING

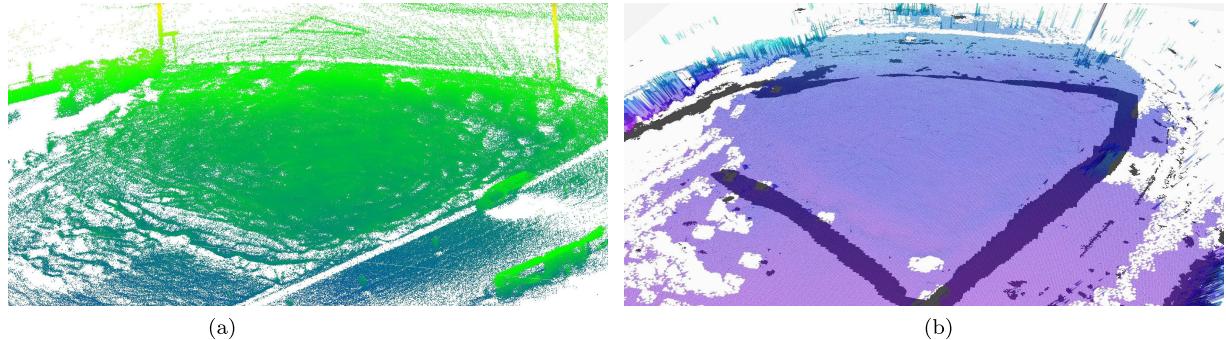
To plan excavation tasks effectively, the system requires a precise construction site map. This map, often represented as an elevation map, outlines the excavation area, target depth, dump zones, and any obstacles. In addition, it can be used to differentiate between the original terrain and newly excavated soil.

To produce an initial map for global planning, we manually traversed the excavation site and collected point clouds. These were then registered in a georeferenced frame with a voxel size of 0.05 m using Open3D simultaneous localization and mapping (SLAM) [12]. This iterative closest point (ICP)-based SLAM system identifies loop closures through local map matching. We registered the map by integrating GPS readings with ICP registration. We used the Earth-centered, Earth-fixed (ECEF) coordinate system for mapping since excavation plans typically define dig points using ECEF or its derivatives. Alternatively, this map can also be acquired using traditional surveying methods [22] or other robotic technologies such as drones [35].

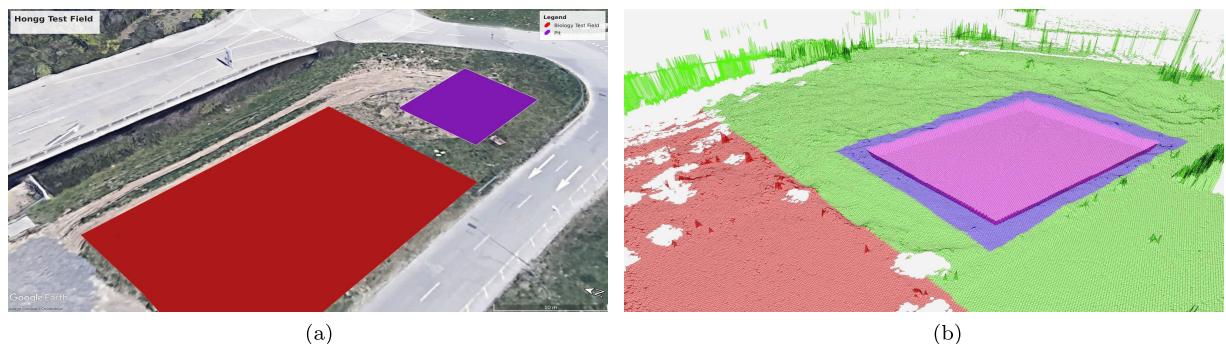
We then transformed the point cloud map into a 2.5-D elevation map at a resolution of 0.1 m using the grid map library [11], as depicted in Fig. 3. This map representation is consistently used throughout this article. We applied a hole-filling filter to the map to counteract minor occlusions from the LiDAR's field of view due to soil irregularities.

#### 2) USER INPUT

The system requires several input layers in the form of grid maps. These layers detail the excavation area, the desired depth, the locations of obstacles, and the designated dump sites. Each layer functions as a mapping  $f(x, y) \rightarrow z$ , where



**FIGURE 3.** (a) Point cloud map generated with Open3D SLAM of the excavation site. (b) Elevation map (in blue) and offline traversability map (black nontraversable) generated from the point cloud map. The traversability has been manually modified to include the fence, which is not visible via the LiDAR.



**FIGURE 4.** (a) Users use Google Earth Pro to delineate various layers. No-go zones appear in red, dirt dumping restrictions are marked in red, and excavation zones are indicated in violet. Excavation depths are assigned to these polygons. (b) Target elevation is color-coded according to the excavation mask layer. Red indicates no-dumping zones, green shows allowed dump sites, blue marks the excavation area boundary, and violet represents the excavation area.

$(x, y)$  represents a position on the map and  $z$  is the value of the layer at that position. In the context of an elevation map,  $z$  corresponds to the terrain's elevation.

Google Earth Pro serves as the tool for defining these layers. It allows users to create georeferenced shapes, allocate attributes, and specify elevation at each vertex. It also provides a 3-D and top-down view of the excavation site. The user-defined layers are merged to produce a unified grid map, which the excavation planning system uses (see Fig. 4). The integrated map comprises the following layers.

- 1) *Elevation*: This is sourced from a point cloud map, representing the terrain's height.
- 2) *Target Elevation*: Defined in Google Earth Pro, this layer marks the intended excavation depth. Users can set this as absolute values or relative to the existing ground level.
- 3) *Excavation Mask*: A layer with integer values showing both the excavation and dump sites.
- 4) *Occupancy*: A binary layer detailing terrain traversability, the presence of obstacles, and off-limit areas.

### 3) STATE ESTIMATION

To estimate the robot's state, we use a graph-based multisensor fusion approach [26]. The state estimator fuses

IMUs, encoders, RTK-GPS, and LiDAR measurements. The graph-based fusion of the LOAM-based LiDAR odometry, as described in CompSLAM [18], together with the RTK-GPS measurements allows the system to be robust to connectivity and GPS outages. A robust state estimator is essential for the machine to operate reliably for several hours.

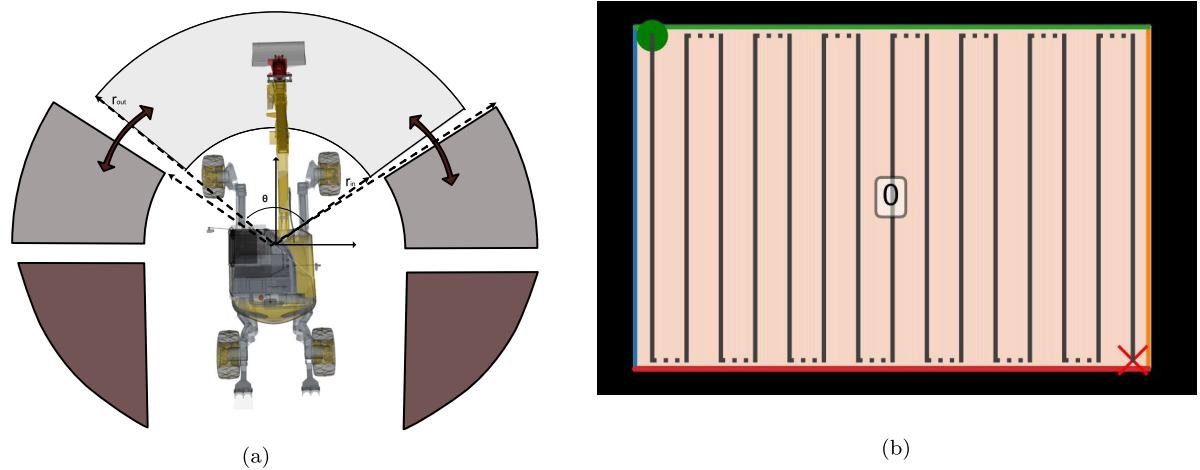
## B. GLOBAL EXCAVATION PLANNER

The global excavation planner determines the sequence of poses for the excavator base to ensure that the entire excavation area is within arm's reach. The planner design is grounded on the local excavation geometry shown in Fig. 5(a). The local digging geometry, illustrated in Fig. 5(b), defines the excavator's accessible digging area based on its current base pose.

### 1) OVERVIEW

The global planner accepts an excavation map from the user, encoding target dig and dump zones and obstacles. The proposed method is well suited for excavation planning due to several key features.

- 1) It decomposes the dig space into coverable cells using Boustrophedon decomposition, which adapts well to the changing environment during excavation. Unlike



**FIGURE 5.** (a) Local excavation geometry, showing the five planning areas.  $\theta$  is the workspace angle and has a value of 1.9 rad, and  $r_{in}$  and  $r_{out}$  are the digging area inner and outer radius equal to 7.0 and 4.5 m, respectively. The inner and outer radii of the front-left and front-right areas are 7.5 and 3.5 m, respectively. (b) Coverage paths are generated for a rectangular workspace. The green dot represents the starting point of the excavation, and the red cross is the endpoint. The dotted lines show the quickest way to join two excavation lines.

previous methods that rely on fixed cell connectivity, this approach updates the cell structure as excavation progresses.

- 2) The directed quotient graph generated over the cells captures the restricted movement constraints in excavation, where revisiting dug cells is not allowed. This is in contrast to classical coverage methods designed for static environments.
- 3) Finding a minimum branching spanning tree minimizes intercell movement, to avoid unnecessary travel through the global workspace. The post-order traversal maintains access to remaining cells while ensuring nontraversal of dug cells.
- 4) Dynamic programming efficiently identifies the optimal cell corner sequence that minimizes travel distance, taking into account the changing cell connectivity during excavation.
- 5) The outer optimization loop selects the coverage orientation that balances key excavation objectives, such as maximizing covered area, minimizing local workspaces, aligning with the main excavation direction, and minimizing path length. This allows adapting the plan to different excavation scenarios such as trenching or bulk excavation.

The system includes an outer and inner optimization loop to efficiently generate excavation plans tailored to the unique constraints and objectives of the earthmoving process.

## 2) BOUSTROPHEDON DECOMPOSITION AND QUOTIENT GRAPH

The first step in the process is to decompose the target space using the Boustrophedon decomposition [6] based on a specific coverage orientation. During this decomposition, the space is segmented into different cells as a slice passes

over the workspace, altering its connectivity. Fig. 6(a) and (d) provides the examples of such decomposition, demonstrating a vertical coverage orientation. Following the decomposition, the cells' connectivity is represented through a graph, where each node symbolizes one of a cell's four extreme points. The adjacency matrix is then constructed using an equivalence relationship between two vertices  $v_1$  and  $v_2$ , as defined in the following equation:

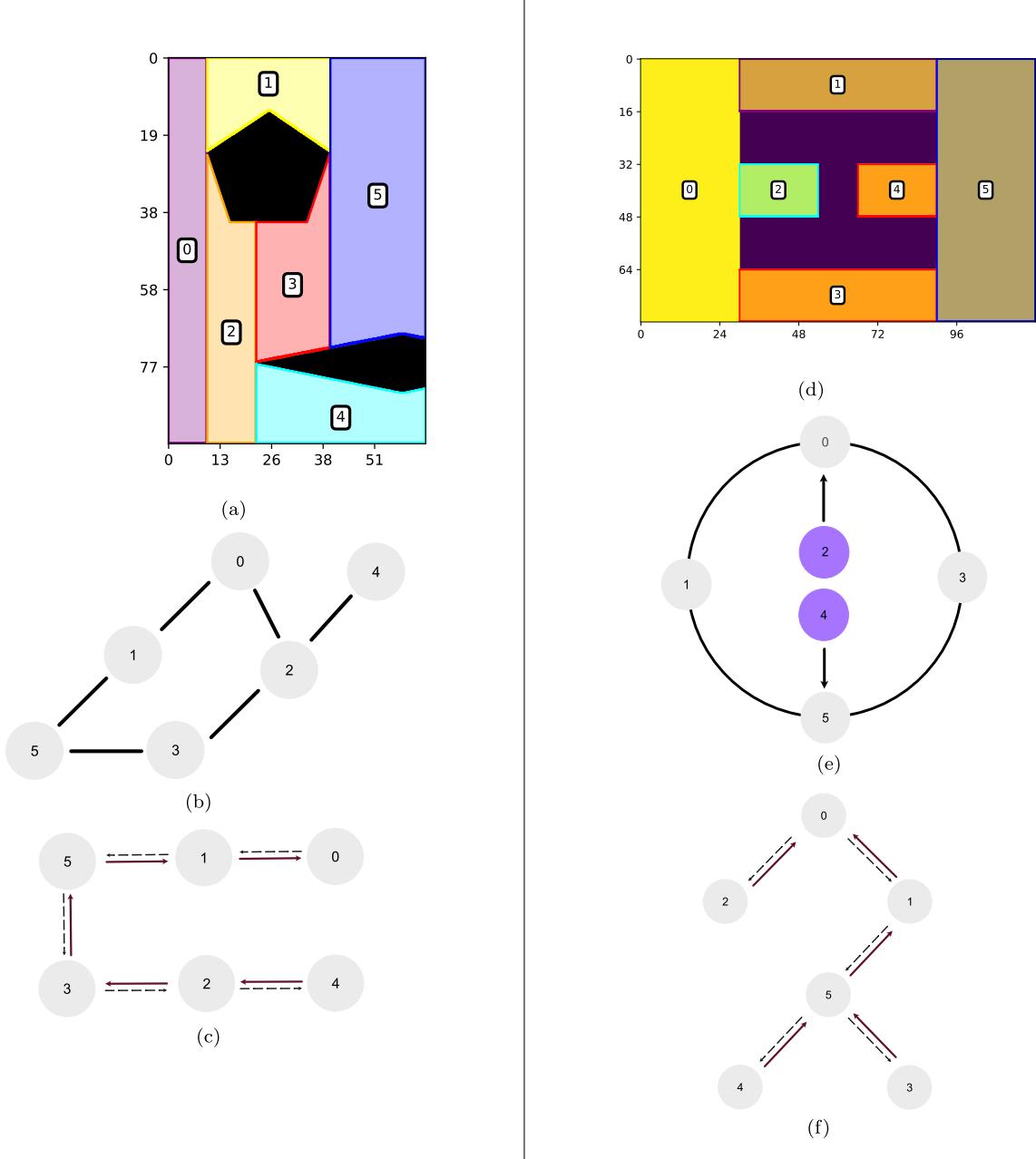
$$v_1 \sim v_2 \iff \exists c \in C, v_1 \in c, v_2 \in c \quad (1)$$

where  $C$  signifies the set of cells. Subsequently, the quotient graph is generated by collapsing all equivalent vertices, with the edges connecting cells sharing at least one common vertex [see Fig. 6(b)]. The graph is bidirectional in scenarios involving only convex obstacles, except when the obstacles align with the sweeping slice's direction. Conversely, the graph is directed when one vertex shares corners with another, but not vice versa (e.g., when the obstacle is concave, the sweeping slice passes through its concave part), as illustrated in Fig. 6(e).

## 3) SPANNING TREE AND TOPOLOGICAL SORTING

The excavation problem's structure presents practical challenges that prevent a direct formulation as a TSP or its variants. In this context, multiple cell visits are allowed and might be necessary, but a dug cell becomes unavailable for navigation. Furthermore, starting and ending at the same cell is not a requirement.

We address this problem by constructing a spanning tree of the graph, using post-order traversal to ensure full tree traversal. This strategy enables the excavator to dig the entire area, revisiting undug cells and minimizing driving time by reducing the number of branches in the tree. A tree with no vertex with a branching degree greater than one allows continuous digging. The problem, known as the minimum branch

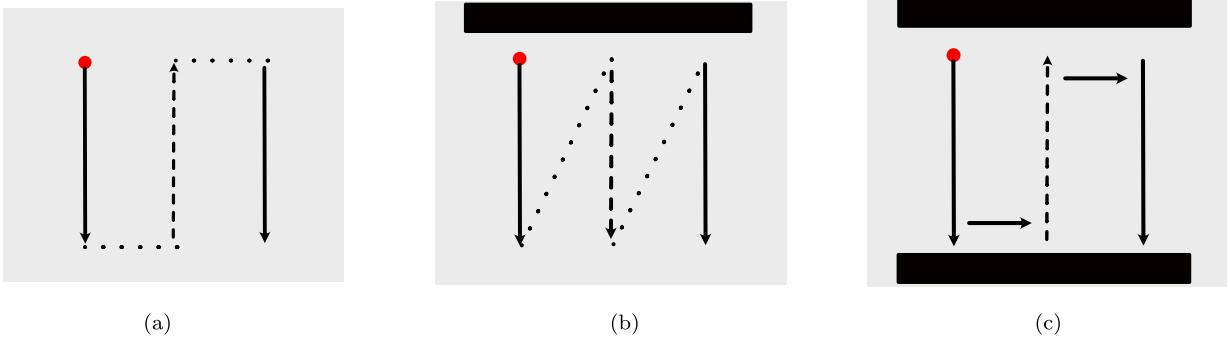


**FIGURE 6.** Workspace decomposition and cell visitation workflow, colors have been added just for easy visualization. Top: two examples of Boustrophedon decomposition. The space has been subdivided into different cells, each marked with a different color. Center: corresponding quotient graph to the two examples. For the left column case, the graph is undirected since all cells share one with at least another cell in one corner; for the right column, no cell shares a corner with cells 2 and 4. Therefore, these two cells must be excavated before cells 0 and 5. The simply connected components of the graph are indicated with a different color. Bottom: resulting in minimum branching spanning trees for the above graphs. A dashed edge indicates that the machine moved to the cell, while a solid edge indicates that the machine excavated the cells. (a) Example with convex obstacle. (b) Quotient graph (convex). (c) Minimum branching spanning tree (convex). (d) Example with concave obstacle. (e) Quotient graph (concave). (f) Minimum branching spanning tree (concave).

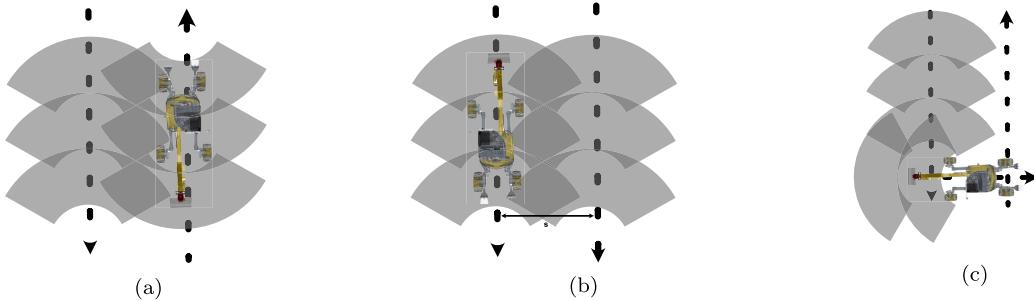
vertices problem (MBVP), is efficiently solvable via dynamic programming for both directed and undirected graphs [29]. After generating the spanning tree, the sequence of cells to visit is determined through post-order traversal, with the results depicted in Fig. 6(c) and (f).

#### 4) DYNAMIC PROGRAMMING FOR LOCAL COVERAGE

With the sequence of cells determined, the problem now focuses on selecting optimal corners to visit and the associated coverage subroutine. Three potential subroutines connect the vertices of a cell (Fig. 7), creating a series of spaced



**FIGURE 7.** Three possible subroutines used to cover a cell. (a) Alternating lane directions with obstacles lying only one side. (b) Two consecutive lanes having the same direction. (c) Coverage with obstacles on both sides present.



**FIGURE 8.** Coverage geometry, where the excavator always moves backward, corresponding to the subroutines shown in Fig. 7. The worst case alignment of the local workspace is used to determine the lane spacing. (a) Alternating travel direction. (b) Same travel direction. (c) Coverage with obstacles.

base poses (Fig. 8). The distance covered during each routine is the sum of distances between successive base poses, with an extra cost term accounting for the turns, approximating the additional distance needed for maneuvering. Should the robot's footprint collide with any obstacle, the path length is set to infinity; otherwise, distances in undug areas are approximated as the Euclidean distance between points.

The problem of finding the optimal sequence of corners to visit can also be efficiently solved using dynamic programming. The complexity of this approach is  $O(CV^2)$ , where  $C$  represents the number of corners and  $V$  represents the number of subroutines available.

We use the following recurrence relation shown in the following equation to solve the problem using dynamic programming:

$$D_{i,c_k} = \min_{c_n, c_j, l} (D_{i-1, c_n} + d_{i,l}(c_n, c_j) + d_o(c_j, c_k)) \quad (2)$$

where  $D_{i,c_k}$  represents the minimum distance required to reach corner  $c_k$  in cell  $i$ . The term  $d_{i,l}(c_n, c_j)$  denotes the length of the coverage path between corners  $c_n$  and  $c_j$  in cell  $i$ . The coverage path can have a flipped lane in the last position, controlled by the binary variable  $l$ . Finally,  $d_o(c_j, c_k)$  signifies the length of the path between corners  $c_j$  of cell  $i$  and  $c_k$  in cell  $i+1$ .

Fig. 9 shows the solutions to the global coverage problem for the cases we have considered.

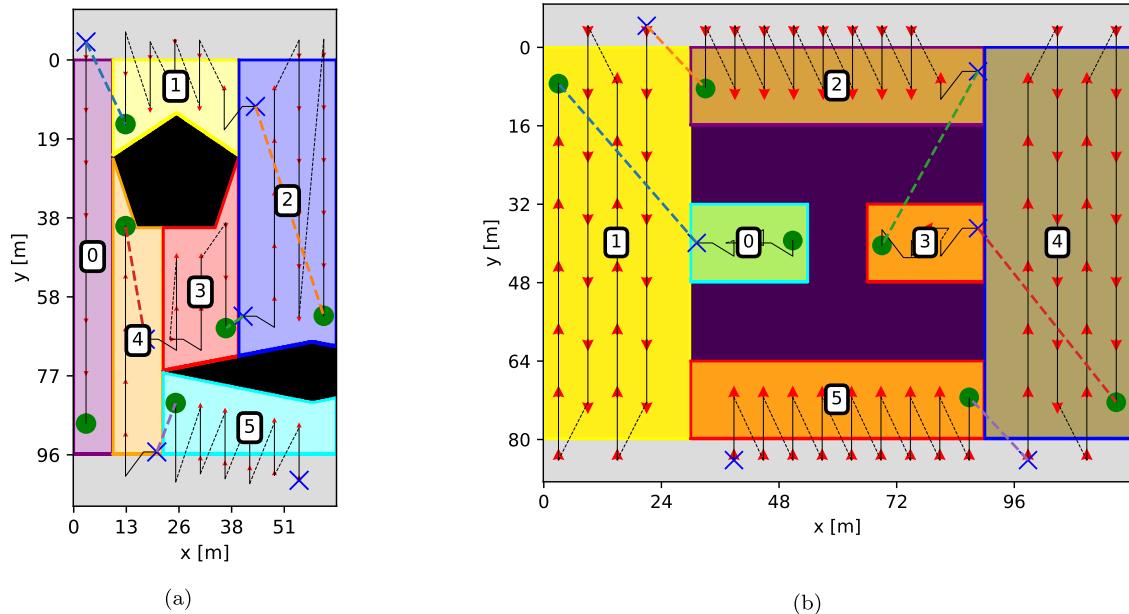
## 5) OUTER OPTIMIZATION

The excavation layer is first segmented into locally connected areas. A global planning problem is then formulated and solved for each area individually. To create a comprehensive excavation plan, paths determined for these local areas are concatenated, optimized as a TSP. The orientation of each connected area significantly influences the plan, affecting factors such as the number of cells involved, available workspaces, and path feasibility.

The optimal orientation for each excavation area is determined by solving a nonlinear optimization problem, which is represented as follows:

$$\theta = \arg \min J(\theta) = c_{ax}(\theta - \phi) + c_p L_p + c_n N_w - c_a A_c \quad (3)$$

where  $\theta$  is the orientation angle to be optimized. The function  $J(\theta)$  includes terms accounting for various aspects of the excavation plan. The term  $c_{ax}(\theta - \phi)$ , where  $c_{ax}$  is a coefficient, represents the importance of alignment with the main axis  $\phi$  of the excavation object, penalizing orientations that deviate from  $\phi$ . The term  $c_p L_p$  adds cost based on the path length  $L_p$  necessary to cover the excavation area, with  $c_p$  modulating this influence. The term  $c_n N_w$ , where  $c_n$  is a coefficient, adds cost based on the number of workspaces  $N_w$  required, as more workspaces can increase operational complexity. Finally, the term  $-c_a A_c$ , where  $c_a$  is a positive



**FIGURE 9.** Coverage paths for the scenarios illustrated in Fig. 3. (a) Excavation plan for convex obstacles (or not dig zones), with each cell's excavation starting at the green dot and ending at the red cross. (b) Solution for a concave obstacle, following the same start and end conventions. Solid lines represent excavation paths, while dotted lines indicate the forward orientation of the base during navigation to the next path segment. Cells are re-indexed to match the coverage sequence.

coefficient, rewards higher coverage fractions  $A_c$  of the area, promoting more efficient space usage.

Depending on the type of excavation, such as trenching or bulk excavation, these coefficients are adjusted to emphasize different aspects of the plan. For trenching, emphasis is placed on alignment with the trench's main direction through  $c_{ax}$ , while for bulk excavation, where alignment might be less critical, more emphasis might be placed on minimizing the number of workspaces or maximizing the area coverage.

## 6) CONSIDERATIONS ON DUMPING CONSTRAINTS

Dumping constraints significantly impact the global planning process, particularly for small excavations where a single machine handles the work. In larger excavation sites, additional equipment, such as skid steers, backhoes, tracks, and cranes, are employed to efficiently remove soil from the site, allowing the excavator to focus on digging. In such cases, the excavator can simply dump the soil on the side of the lane that does not obstruct its path, and other machines will remove it.

However, when dumping constraints are present, the global planner may need to adapt the coverage line by extending it beyond the digging workspace, if necessary, to reach a permanent dumping location. As the excavator moves to the next lane, which must be cleared of loose soil for navigation, the local workspaces of the previous lane become unreachable. Consequently, finding a permanent dumping location on the previous lane is a prerequisite to move on to the next one. If no suitable location is found, the plan is considered infeasible and assigned an infinite cost. The outer optimization loop must then explore alternative starting corners or utilize a different coverage orientation to find a feasible solution.

The presence of dumping constraints can lead to various scenarios that affect the global plan, such as temporary dumping in the current lane, longer excavation paths, and potentially unfeasible plans. Section V-D provides practical examples that demonstrate the effects of different dumping constraints on the planner's performance and the resulting excavation strategies.

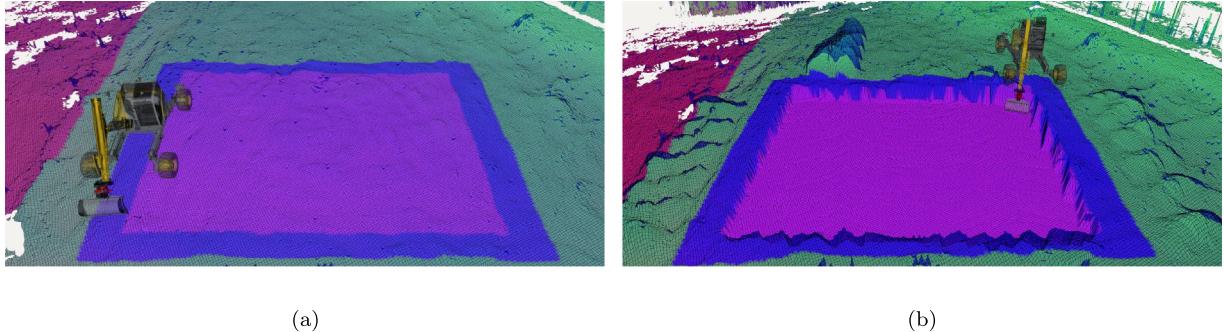
### C. VOLUMETRIC SIMULATION OF EARTH MOVEMENT

This section describes our method to simulate soil movement for system simulation. We represent the soil as a simplified geometric solid volume, excluding dynamics, as our primary focus is excavation planning.

To mimic digging, we track the shovel's path, removing soil without considering the forces on the shovel's edge. Mass conservation is maintained by calculating the volume of soil removed with each scoop and adding an equivalent volume to the dumping location.

Our simulation models the falling soil particles and calculates their ground distribution. This is achieved by dividing the space into “bucket slices,” assuming that particles in each slice fall according to a normal distribution independent of one another. The mean of this distribution is the slice’s center, and the standard deviation is half the bucket’s width. In addition, we assume that the dumping location is relatively flat, preventing lateral soil sliding post-dumping.

The resulting ground height distribution is described by (4) and (5), where variables are defined as follows— $\sigma_x$  and  $\sigma_y$  are half the shovel dimensions in the body frame along the  $x$ -axis and  $y$ -axis,  $\psi$  is the shovel's heading angle in the world frame,  $N$  is the number of discretized elements in the bucket, and  $V_b$



**FIGURE 10.** Elevation maps generated while excavating in simulation. (a) Elevation map at the beginning of the excavation. (b) Elevation map after the excavation is finished.

is the bucket's volume:

$$h(x) = \frac{V_b}{N} \sum_{i=0}^N \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu_i)^T \Sigma^{-1} (x - \mu_i)\right) \quad (4)$$

$$\Sigma = \begin{bmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{bmatrix} \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix}. \quad (5)$$

In Fig. 10, two snapshots of a pit's excavation in our simulated test field are shown. The model's main goal was to expedite the system's overall development through simulation, not to create a realistic representation of soil transport. Validating the model using real-world data is challenging since factors, such as water content, soil type, and compaction, are not considered.

#### D. LOCAL EXCAVATION PLANNER

The local excavation planner arranges soil around the excavator's base to fulfill the excavation task. Fig. 5(a) illustrates the local excavation geometry, which is divided into five areas.

- 1) *Front*: An unexcavated digging area.
- 2) *Front Left and Front Right*: Areas for digging or dumping previously dug soil.
- 3) *Back Left and Back Right*: Areas for dumping soil excavated from other regions.

The excavator's rear is kept clear for navigation, enabling operation in constrained spaces. The front areas' sizes can vary with the task.

The planner uses the excavation mask layer in the elevation map, containing constraints and target geometry information.

#### 1) EXCAVATION MASK

The updated excavation mask layer guides the choice of dig and dump zones based on user input. It includes the following.

- 1) *Dig Area (Violet)*: Identified by a discrepancy between target and current elevations.
- 2) *Permanent Dump Area (Green)*: For permanent disposal of excavated material.
- 3) *Neutral Area (Light Blue)*: An area for digging and temporary material dumping, eventually moving to the permanent dump area.

4) *No-Go Area (Red)*: User-defined, no dig or dump area.

5) *Boundary Area (Dark Blue)*: Equivalent to a no-go area at the excavation zone's edge, preventing soil spillage.

The neutral area combines user-defined areas, the convex hull of potential future base footprints, and unreachable dump areas from the current machine configuration. It ensures a continuous path to future target poses by keeping material away from this region.

#### 2) DIG AND DUMP ZONE SELECTION

The local planner selects the dig and dump zones by first checking the front zone and then lateral front zones based on the criteria.

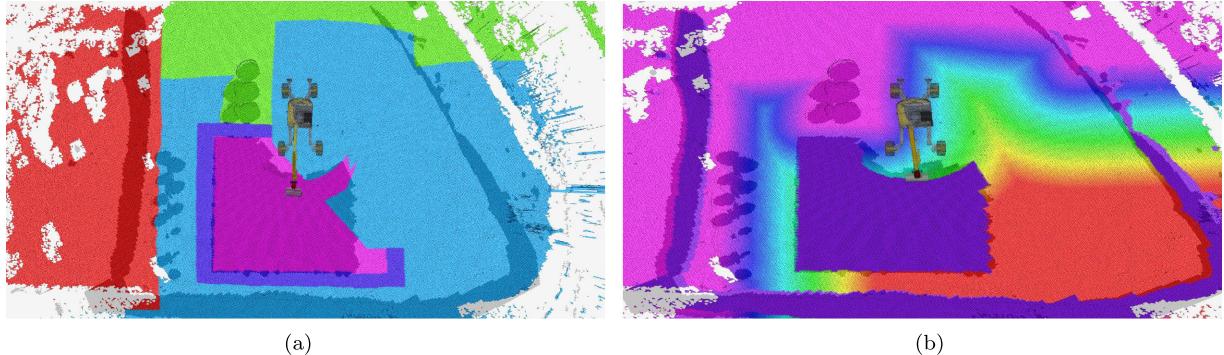
- 1) Less than 10% of elevation map cells deviate more than 0.1 m from the target elevation.
- 2) Remaining volume to excavate is less than 8% of the total volume.

The front zone's target elevation uses the “desired elevation” layer, while the front-left and front-right zones use the “original elevation” layer. Thresholds for these conditions were determined experimentally for a good balance between accuracy and speed.

The dumping zone is selected from one of four options: front left, front right, back left, or back right. The selection is based on the lowest dumping cost, calculated as follows:

$$C_D = \frac{1}{N} \sum_{i=1}^N SDF_{dump}(x_i) + \alpha ||x_{dig} - x_{dump}||^2 \quad (6)$$

where  $C_D$  represents the dumping cost,  $N$  is the number of cells within the selected zone, and  $SDF_{dump}(x_i)$  is the signed distance function for each cell  $i$  in the zone relative to the nearest permanent dumping area. This function measures the global planning cost and incorporates dumping constraints. The coordinates for the digging and dumping locations are denoted by  $x_{dig}$  and  $x_{dump}$ , respectively. The weighting factor  $\alpha$  is set to 4.0, which scales the influence of the distance between the digging and dumping locations on the overall cost. The term  $||x_{dig} - x_{dump}||^2$  calculates the squared Euclidean distance between these locations, aiding in the prioritization of closer dumping zones when several of have



**FIGURE 11.** Elevation maps of the terrain during the digging process in simulation. (a) Colored based on the excavation mask following the color scheme defined above. (b) Colored based on the dumping cost, with red indicating the highest cost and violet indicating the lowest cost.

similar SDF<sub>dump</sub> costs. The dumping cost at an intermediate stage of pit excavation is shown in Fig. 11. This formulation aims to minimize transport time and number of moves before final disposal.

A permanent dump area is accessible if within the excavator's reach or further along the path and does not obstruct future targets. The local excavation geometry, shown in Fig. 5(a), and the global planner's strategy contribute to this. Several dirt handling scenarios in the single-cell excavation are demonstrated in Fig. 12.

The dig and dump selection continues until all zones are excavated, and then, the machine transitions to the refinement phase, involving soil grading.

During refinement, the excavator sweeps from left to right in a grading motion to smooth the terrain. The arm's back-and-forth movement levels any unevenness to achieve the desired slope and surface smoothness.

#### E. ARM TRAJECTORY PLANNER

The arm trajectory planner governs all arm trajectories, such as digging, dumping, and grading motions. It remains agnostic to the specific trajectories and focuses on optimizing the trajectories' parameters or initial conditions. The digging system optimizes trajectories to scoop the maximum volume in the target zone. This work employs parameterized trajectories with inverse kinematics, but the planner is extendable to other low-level controllers.

#### F. PARAMETRIZED DIGGING TRAJECTORIES

The digging trajectory is divided into penetration, dragging, and closing. Penetration starts at ground level at an initial position,  $x_0$ , with the shovel edge moving along the shovel's  $x$ -axis, as shown in Fig. 13, until the target depth is reached. This occurs when the target elevation or maximum depth to the surface is attained. The attitude angle,  $\gamma$ , is independent of the soil profile and varies linearly across the workspace, with values summarized in Table 1.

In the dragging phase, the bucket moves radially along the boom, with its angle varying linearly between  $\gamma_{\min}$  and  $\gamma_{\max}$ . The dragging motion stops when the bucket is full,

a self-collision risk is detected, or it exits the diggable zone. To increase the average volume per scoop, the scooped volume is calculated by integrating the shovel depth, including areas outside the target workspace but still part of the next workspace. As excavation progresses, scoops diminish in size, ensuring maximum soil removal.

During the closing phase, the bucket begins to close, moving radially and vertically until fully closed and at a target height above the soil.

*Loose Soil Trajectories:* Scooping loose soil is challenging as it lacks support from undisturbed soil and can easily displace and accumulate near the machine. The same parametrization is used to scoop loose soil, with different initial attitude angles and radial distance during the closing phase (as shown in Table 1). The initial attitude angle is reduced, and the bucket only moves vertically up during closing, thus reducing the radial forces on the soil, allowing scooping without dragging and spillage.

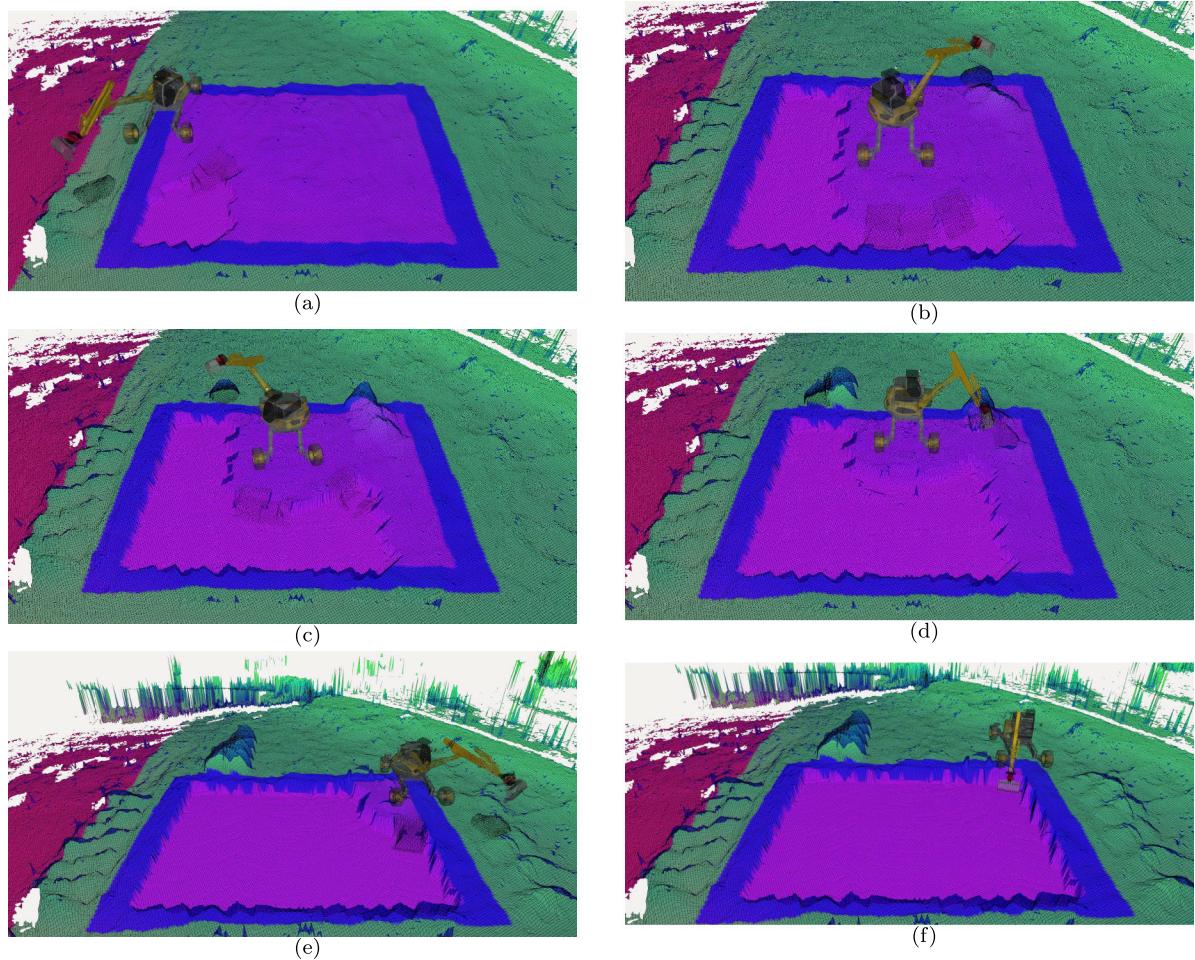
#### 1) DIGGING PLANNER

The digging planner selects the parameters for the digging trajectory to maximize the expected scooped soil volume in the target digging zone. We use the Bayesian optimization as the trajectory parameter space is low-dimensional, we cannot estimate gradients of the objective function, and the objective function is costly to evaluate since the whole trajectory must be simulated. This is particularly important if using a different subsystem generates digging trajectories requiring more computation time.

In our case, the optimization problem [shown in (7)] is reduced to finding the 2-D coordinates of the initial position for the digging trajectory

$$\begin{aligned} & \max_{(\mathbf{r}, \theta)} V_w(\tau(r, \theta)) \\ & \text{subject to } r_{\text{in}} < r < r_{\text{out}} \\ & \quad \theta_{\min} < \theta < \theta_{\max}. \end{aligned} \quad (7)$$

The optimization objective is to maximize the scooped soil volume  $V_w(\tau(r, \theta))$  in the workspace over a circular sector defined by radii  $r_{\text{in}}$  and  $r_{\text{out}}$  and angular limits  $\theta_{\min}$  and  $\theta_{\max}$ .



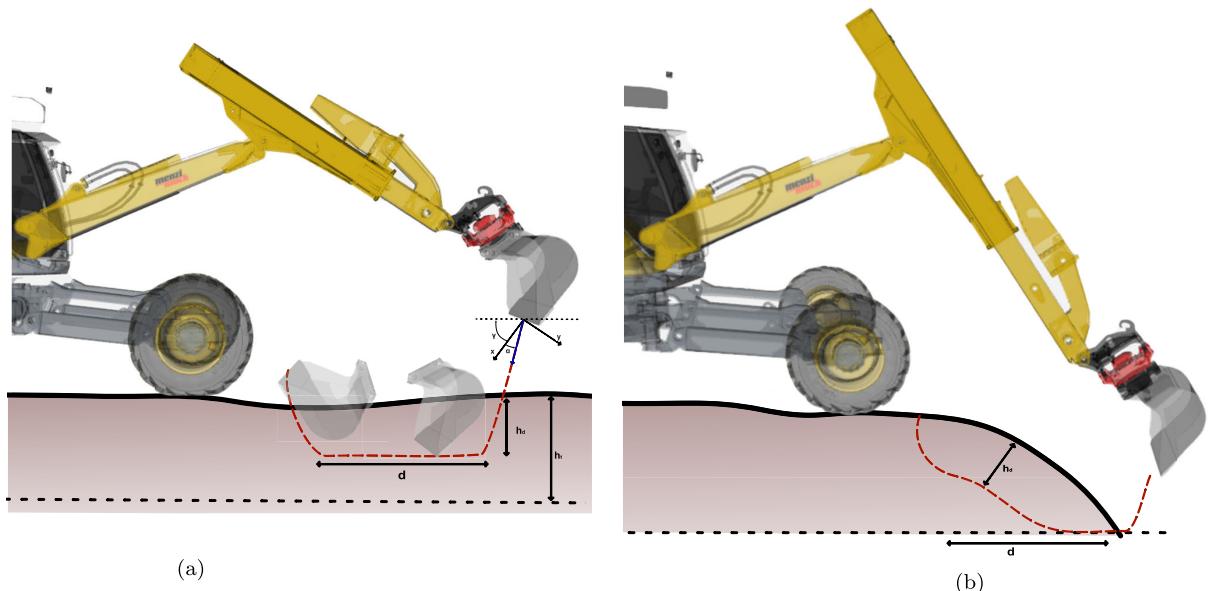
**FIGURE 12.** Excavation mask layer and zone dumping costs for different cases throughout the excavation of a pit in simulation. (a) Excavation mask layer. (b) Soil is moved from the front dig area to the front-left dump area. The back-left area has a higher cost because it is further away from the point where the excavator has dug. (c) First lane is completed, and soil is moved from the front to the back-right zone. The left zones are all inactive as they overlap with already dug areas. The front-right zone has a higher dumping cost since it is further away from the reachable dump zone in the back. (d) Dirt is moved from the front and front-right zone to the back-right zone. (e) Permanent dump zone is unreachable, and the dirt is moved from the front and front-left zone to the back-right zone. (f) First two lanes are completed, and multiple permanent dump zones are reachable. The dirt is moved from the front to the front-right or back-left zone, depending on which zone is closer to the digging point.

**TABLE 1.** Trajectory parameters.

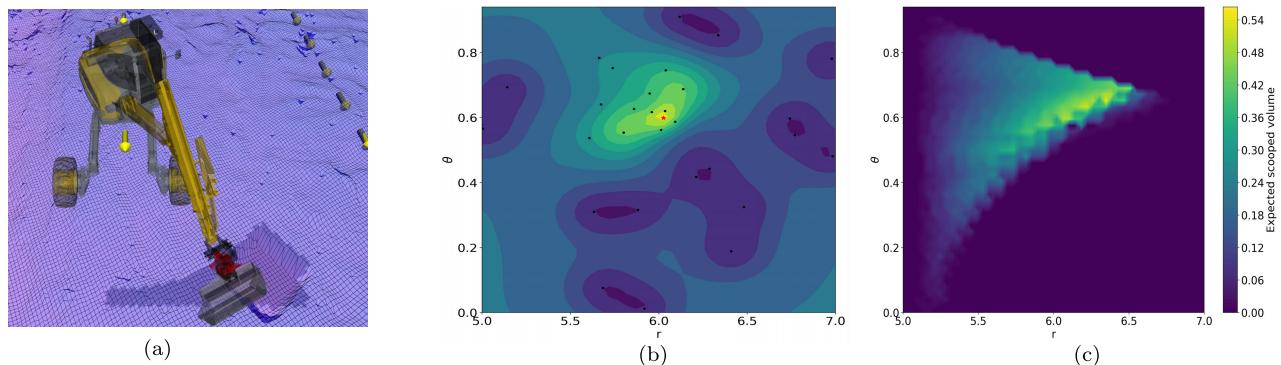
Parameter	Value	Description
$\gamma_{\min}$	0.5 rad	Minimum attitude angle
$\gamma_{\max}$	1.5 rad	Maximum attitude angle
$\gamma_{\max-dirt}$	0.9 rad	Maximum attitude angle for loose soil
$d_{\max}$	1.5 m	Maximum dragging distance
$h_{\max}$	0.3 m	Maximum depth
$V_b$	0.6 m <sup>3</sup>	Bucket volume
$V_{\max}$	0.8 m <sup>3</sup>	Maximum volume
$h_c$	0.5 m	Height change in the closing motion
$v_d$	0.5 m/s	Digging velocity

We solve this optimization problem using the Bayesian optimization with Gaussian processes, expected improvement as the acquisition function, and a custom sampler. The

initial points are sampled uniformly in Cartesian space, some Gaussian noise is added, and then, the points are transformed to polar coordinates. Since we sample multiple times the



**FIGURE 13.** Two examples of digging trajectories on different terrain profiles. (a) Typical on undug flat soil. (b) Common while digging a pit or trench.



**FIGURE 14.** (a) Excavator digging at the first workspace of the pit, indicated in a darker color. The workspace is both constrained radially and tangentially by the edges of the pit. (b) Estimated optimization landscape of the expected scooped volume function via a Gaussian process. The dots in the left plot are the points sampled by the optimizer with our custom sampler. (c) Optimization landscape of the expected scooped volume function created with full search over the elevation map.

same space, the added noise allows the system to explore the dig area better. The Gaussian noise has a standard deviation of one-fourth of the spacing between points. We query the optimizer with 20 randomly selected values and allow ten iterations of refinement for a total of 30 calls to the objective function. The optimizer has been tuned to yield results with an average error of  $0.06 \text{ m}^3$ , approximately 10% of the scooped volume, from the actual maximum. The grid search optimizer, by comparison, needs 150 samples to achieve the same precision, scaling exponentially if the number of tunable parameters increases. The real and estimated optimization landscapes are compared in Fig. 14.

## 2) REFINEMENT TRAJECTORIES

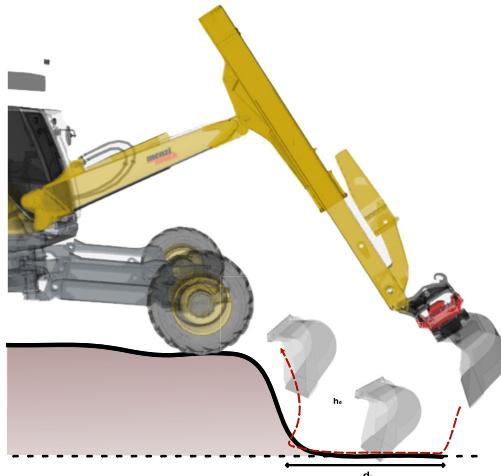
Inspired by expert operators aiming to achieve a final grade of the soil surface, we design refinement trajectories to remove

minor imperfections and spillage left from digging. The front digging zone is expanded by 10% in the radial and tangential directions, and any accumulated soil closer to the machine base than the inner radius of the current digging zone is removed in the next workspace. We use only radially inward motions to minimize the risk of pushing soil out of reach.

During arm movement, the shovel edge height and attitude angle are maintained as specified, and the cabin is rotated by an angle equal to the angular dimension of the shovel edge, as shown in Fig. 15.

## 3) IN-AIR MOTION TRAJECTORIES

The arm trajectories for reaching the digging location and dumping soil are created using Hermite splines with a step velocity profile. This approach enables circular arm movement around the base, with the plan computed in cylindrical



**FIGURE 15.** Refinement trajectory.

coordinates by interpolating between the bucket's initial and final positions. The orientation is fixed for trajectories leading the bucket to the digging trajectory's starting point. In contrast, the bucket is kept closed during the dumping trajectory until the final six seconds when it begins to open. The ground is kept at a minimum distance of 0.3 m to prevent collisions, as determined by querying the elevation map at various points along the shovel edge. The hierarchical inverse kinematic controller presented in [14] tracks the plans.

The dump point is selected by applying the shovel filter (a 2-D projection of the bucket with unit weight) to the dumping zone. This process involves convolving the shovel filter with the height map of the dumping area and conducting a grid search over permissible locations to identify the cell with the minimum dumping cost, as defined in the following equation:

$$C_{\text{dump}}(x, y) = \alpha_d \sum_{i=1}^n \sum_{j=1}^n S_{ij} h_{\text{dirt}}(x + \Delta s, y + \Delta s) - \beta_{bd} x_{bd} - \gamma_{bd} |y_{bd}| \quad (8)$$

where  $h_{\text{dirt}}(x, y)$  represents the height of the dirt at each potential dumping location, reflecting how the dirt piles up. The shovel filter  $S$ , consisting of the bucket's 2-D projection, is convolved with the height data to integrate the dirt's distribution across the selected zone, scaled by the grid's resolution  $\Delta s$ . This convolution effectively assesses how well a potential dump point can accommodate the dirt based on its existing pile height.

The coefficients  $\alpha_d = 1.0$ ,  $\beta_d = 0.1$ , and  $\gamma_d = 0.05$  govern the weights of the respective terms in the equation. The first term, weighted by  $\alpha_d$ , aggregates the adjusted heights across the shovel filter, encouraging the selection of areas with lower dirt heights for even spreading. The second term,  $-\beta_{bd} x_{bd}$ , and the third term,  $-\gamma_{bd} |y_{bd}|$ , involve the coordinates  $x_{bd}$  and  $y_{bd}$  of the dump point relative to the excavator's base. These terms penalize dump points that are

closer to the base, thus promoting dumping further away. This reduces the likelihood of dirt accumulation near the operational base, optimizing workspace usage.

#### IV. NAVIGATION PLANNING

The motion planning module produces safe and efficient paths for an excavator during excavation tasks, utilizing information from an occupancy map and motion constraints.

##### A. OCCUPANCY MAP

The motion planning module employs an occupancy map, a 2-D representation of the terrain's traversability, to validate sampled states and compute path costs. The map is formed by merging information from the following:

- 1) an offline traversability map considering slopes and steps;
- 2) user-defined nonaccessible zones and manually identified obstacles invisible to LiDAR, such as wire fences;
- 3) online-generated maps marking dug areas and soil piles as nontraversable;
- 4) dug areas marked as nontraversable to avoid soil deformation and surface quality degradation.

Fig. 16 depicts an operational example in our test area.

##### B. PLANNER

We utilize RRT\* [34] implemented in the OMPL with a state space defined by Reeds–Shepp curves in SE2 for local motion planning. This choice accommodates the changing nature of dug areas and soil piles.

The planner produces a sequence of poses, accounting for terrain traversability through an occupancy map comprising offline maps, inaccessible areas, and real-time updates. Excavator clearance is also considered, as loading soil too close to the edge may cause collapses, damaging the machine or affecting the grade. Falling risks are evaluated as well.

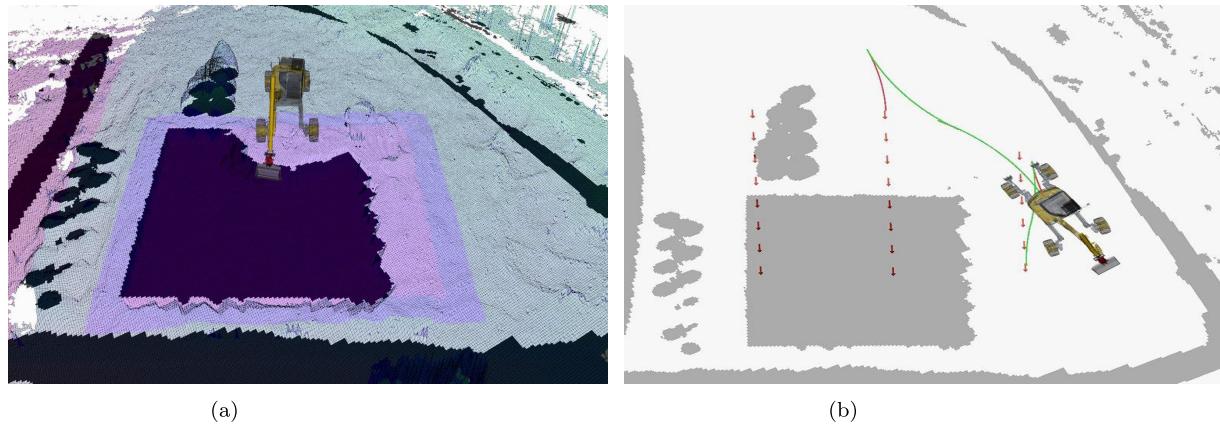
The validity of states is assessed through the occupancy map, and costs are calculated using the following:

$$C_n(s) = \alpha_n \cdot d(s_1, s_2) + \frac{1}{M} \sum_{j=0}^M \beta_n \cdot \exp(-\gamma_n \cdot \text{sdf}_{\text{dug}}(f_{s2j})) \quad (9)$$

where  $d(s_1, s_2)$  is the distance between the current and previous base position and  $\text{sdf}_{\text{dug}}(f_{s2j})$  is the SDF value for the footprint  $f_{s2j}$  and is the number of elevation map cells inside the footprint of the robot. The parameters  $\alpha_n$ ,  $\beta_n$ , and  $\gamma_n$  are set to 10.0, 5.0, and 0.7, respectively.

The full pose is obtained by constraining height relative to soil and maintaining gravity alignment [10]. Plans are tracked using a pure pursuit controller [11], with safety measures for deviation from target paths due to slippery or muddy terrain. Fig. 6(b) provides an example of a generated plan.

The planner's variable time ranges from 0.5 to 5 s for 0.5–60-m plans, utilizing the RRT\* algorithm with up to ten trials. Success was achieved 78.7% of the time in a simulated scenario, and the failure probability is less than  $3e^{-7}$ .



**FIGURE 16.** RRT\* uses an occupancy map to check the feasibility of the path. Both images are generated in simulation. (a) Occupancy map is generated by integrating the latest elevation map information. Note how the pit and dumping piles are marked as untraversable. (b) Generated plan is shown in green (forward moving) and red (backward moving). The excavator is able to navigate around the pit and reach the last lane.

### C. STATE MACHINE

The control state machine for the excavator robot is organized as follows.

- 1) *Initialize Workspace*: It initializes the workspace by defining the local digging geometry.
- 2) *Main Digging Loop*: While the workspace is not complete, the robot repeatedly goes through the following states.
  - a) *Check Workspace*: It determines if the workspace is complete and finds the next dig and dump point using the local planner.
  - b) *Find Dig Point*: It uses the digging planner to locate the starting point of the scoop and moves the arm to that position.
  - c) *Dig*: It executes the digging motion.
  - d) *Dump*: It identifies the dump point, moves to it, and discards the soil.
- 3) *Find Path Plan*: It locates a path plan to the subsequent base pose using the RRT\* planner.
- 4) *Driving*: It propels the excavator to the designated base pose utilizing the pure pursuit controller.

*Note:* The control state machine also handles various corner cases separately to ensure robust operation.

## V. EXPERIMENTS

In this section, we provide an overview of the experimental evaluation conducted to assess the performance of our excavation planner. We employ two distinct sets of experiments: simulation-based tests focusing on the global planner's performance in a realistic excavation scenario and real-world deployment to evaluate the full system's capabilities during the excavation of a prototypical building foundation.

This section discusses the tests carried out to evaluate the performance of our excavation planner. We conducted two types of tests: 1) simulations that assess the global planner's efficiency in virtual digging environments and 2) real-world

tests that gauge the system's overall ability during the excavation of a typical building foundation.

### A. GLOBAL PLANNER EXPERIMENTS

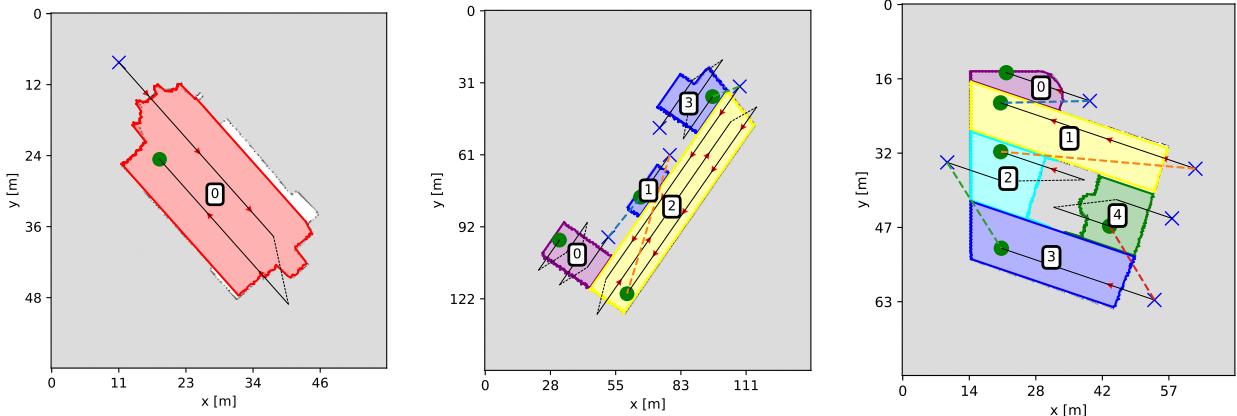
This section discusses the tests carried out to evaluate the performance of our excavation planner. We conducted two types of tests: simulations that assess the global planner's efficiency in virtual digging environments and real-world tests that gauge the system's overall ability during the excavation of a typical building foundation.

These building maps vary in size, from 20 to 100 m on each side. The random crops are even bigger, from 100 to 1000 m per side. From these data, we generated five types of excavation tasks.

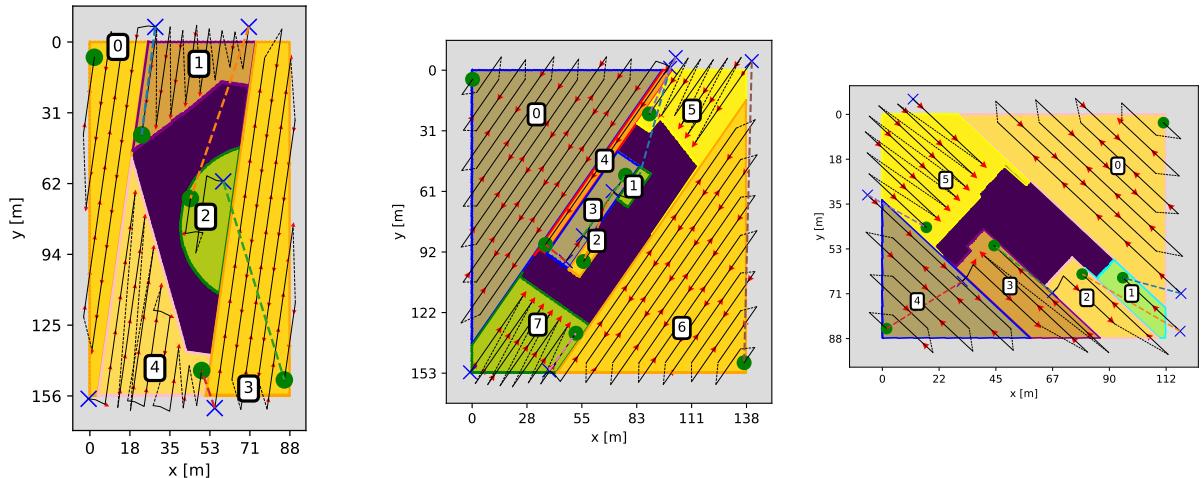
- 1) *Foundations*: Digging the shape of a building with no obstacles.
- 2) *Exterior Foundations*: Digging the inverse shape of a building's foundation and treating the buildings as obstacles.
- 3) *Exterior Foundations Traversable*: Similar to above, but here, the building shapes can be crossed or passed through.
- 4) *Crops*: Digging that involves multiple building shapes on one map.
- 5) *Exterior Crops*: Digging the insides of streets and parks, with buildings acting as obstacles.

For the first three tasks, we assumed dirt that could be placed anywhere outside the dig area. For the last two tasks, due to their complex nature, we did not consider how to manage the excavated dirt. Instead, we presumed dirt that could be moved off-site from any location using other methods or machines.

We assessed the planner's performance based on four criteria: successful plan rate, path efficiency, workspace efficiency, and digging coverage. A plan is “unsuccessful” if the



**FIGURE 17.** Three solved samples from the “Foundations” dataset.



**FIGURE 18.** Three solved samples from the “Exterior Foundations” dataset.

planner cannot process the digging area’s shape or cannot find a digging solution at all.

The *path efficiency* indicates how straight and short the system’s movement paths are. It is found by adding up the straight-line distances between all digging positions and then dividing by the digging area’s size

$$S_p = \sum_{i=0}^{N-1} \frac{(x_{B_{i+1}} - x_{B_i})}{\sqrt{A_d}}. \quad (10)$$

The *workspace efficiency* measures how many local working areas are necessary for the planning problem. It is determined by

$$S_w = \frac{N_w \cdot A_w}{A_d}. \quad (11)$$

where  $A_w = (1/2)\pi R_{\max}^2$  represents the reference workspace area. The *digging coverage* reveals what percentage of the digging area the plan covers.

The solved excavation plans for the five different datasets can be seen in Figs. 17–21.

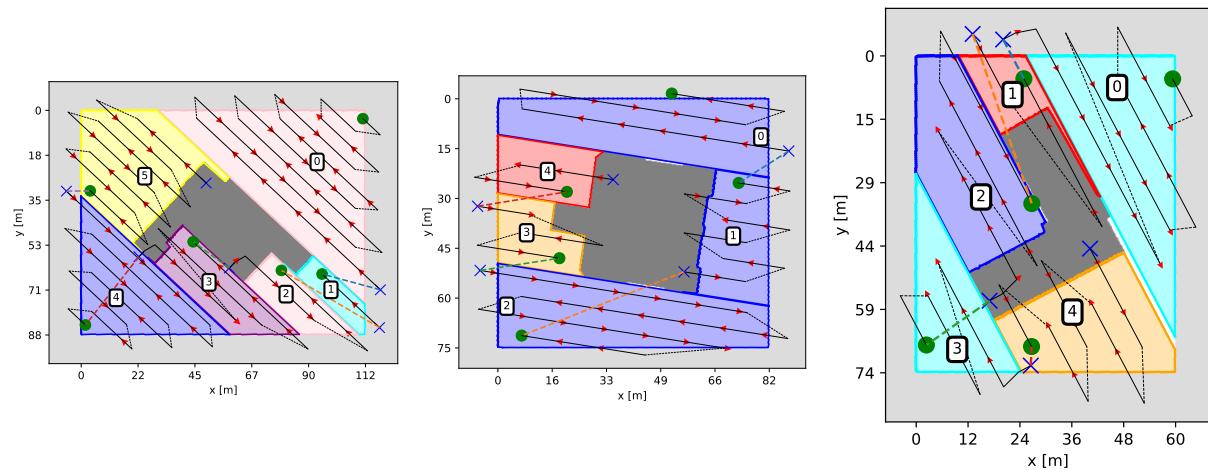
Tables 2 and 3 show the benchmark scores for the global planner without and with coverage orientation optimization, respectively. When not optimizing the coverage angle, it was aligned with the main axis of the excavation area, which already provided a stronger baseline compared to taking the original orientation of the excavation map into account.

## B. DEPLOYMENT EXPERIMENTS

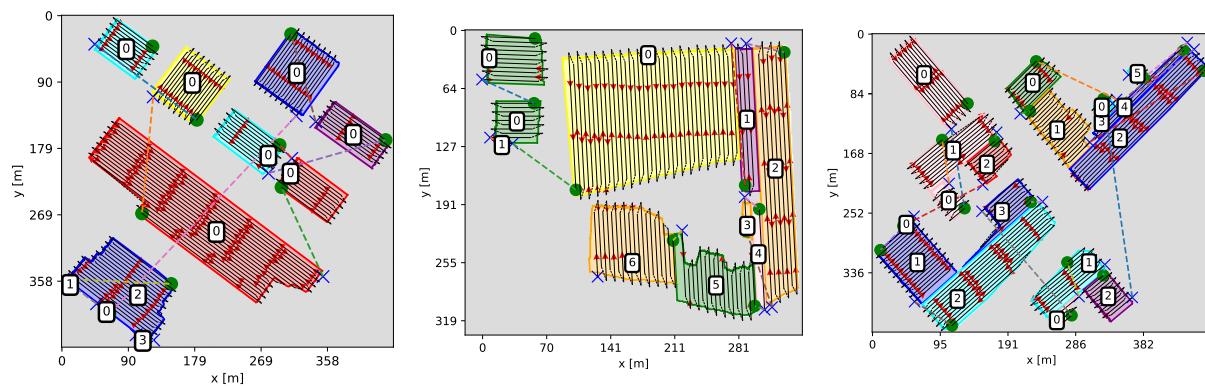
The experiment assessed the performance of our autonomous excavator in digging a pit of dimensions  $15.6 \times 11.5$  m and a target depth of approximately 1 m.

The same target height was set for the entire excavation area to maintain a uniform pit floor. This task was chosen as it is typical for excavator operators and requires multiple passes to relocate the dug soil without using a dump truck or wheel loader.

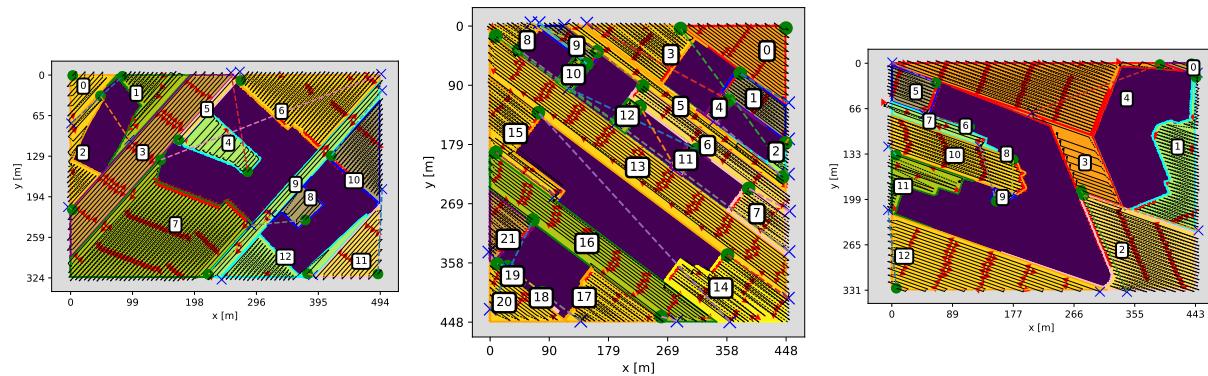
The solution is depicted in Fig. 22. Yellow arrows represent base poses, and the excavation mask color scheme is consistent with Fig. 12. Only corners 1 and 2 of the four considered starting points yield feasible solutions due to spatial



**FIGURE 19.** Three solved samples from the “Exterior Foundations Traversable” dataset.



**FIGURE 20.** Three solved samples from the “Crops” dataset.



**FIGURE 21.** Three solved samples from the “Exterior Crops” dataset.

constraints and a manually marked fence, which prevent full pit coverage beginning from the other two corners.

Fig. 4 displays the test field’s map, created using Open3DSlam, and the user input layers. The machine was operational for 4 h and 25 min to complete the pit, with a time breakdown in Table 4. Fig. 23 illustrates the digging

stages in the test field, previously shown in simulation in Fig. 12. Videos of the excavation stages are available in the supplementary materials.<sup>2,3</sup>

<sup>2</sup>[https://youtu.be/bWw4RRqz\\_dM](https://youtu.be/bWw4RRqz_dM)

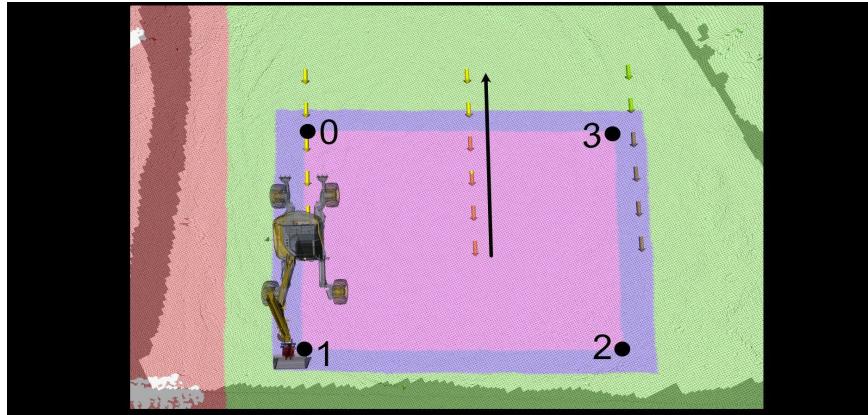
<sup>3</sup><https://youtube.com/playlist?list=PLpIdy9k5iMC9hMnRszYzi8OUj-WYN6UQ>

**TABLE 2.** Benchmark scores for the global planner without coverage angle optimization.

Dataset (size)	$S_p$	$S_w$	Coverage Fraction
Foundations (838)	3.23 (2.52)	14.20 (4.09)	0.944 (0.087)
Exterior Foundations (838)	9.96 (4.60)	15.03 (1.78)	0.952 (0.115)
Exterior Foundations Traversable (838)	9.78 (4.24)	15.35 (1.77)	0.960 (0.104)
Crops (100)	68.12 (22.57)	27.95 (3.29)	0.912 (0.163)
Exterior Crops (100)	105.23 (45.16)	26.66 (3.51)	0.782 (0.293)

**TABLE 3.** Benchmark scores for the global planner with coverage angle optimization. The coverage direction is aligned with the major axis of the excavation area.

Dataset (size)	$S_p$	$S_w$	Coverage Fraction
Foundations (838)	5.08 (2.51)	19.89 (4.38)	0.982 (0.052)
Exterior Foundations (838)	9.87 (5.40)	13.66 (2.11)	0.972 (0.072)
Exterior Foundations Traversable (838)	9.64 (4.42)	13.91 (2.14)	0.978 (0.041)
Crops (100)	73.79 (19.75)	29.09 (3.20)	0.991 (0.028)
Exterior Crops (100)	130.08 (41.67)	26.45 (3.41)	0.982 (0.039)


**FIGURE 22.** Solution to the excavation task. The yellow arrows correspond to the base poses, and the color scheme for the excavation mask is the same as used in Fig. 12.

**TABLE 4.** Breakdown of operations' time by state of the excavator.

State	Mean Duration (std)	Total Duration
Initialize Workspace	0.45 (0.18) s	0h 0m
Check Workspace	0.29 (0.13) s	0h 2m
Dig	14.13 (3.38) s	1h 51m
Arm to Dig Point	8.04 (3.64) s	1h 8m
Arm to Dump Point and Dump Soil	9.91 (2.30) s	1h 1m
Find Path Plan	9.38 (38.20) s	0h 7m
Driving	34.07 (37.07) s	0h 12m
Retract Arm	3.96 (1.07) s	0h 1m
<b>Total</b>	-	4h 25m

Evaluating the efficiency involves two metrics: cycle time and volume of soil excavated per time unit. The recorded cycle time—the time needed to dig, dispose of soil, and reposition the arm for subsequent digging—is 32.08 (5.47) s. However, directly comparing the system's efficiency to that

of a proficient human operator can be challenging. This complexity arises from the fact that the cycle time is influenced by a multitude of variables, including but not limited to the size and type of excavator used, the properties of the soil, and the employed method of unloading [23].



**FIGURE 23.** Snapshots of the pit excavation, each image shows a different earthworks planning strategy. (a) Terrain before the excavation. (b) Dumping on the left side. (c) Soil is moved from the front of the dig area to the back-left area, which is still inside the excavation area but is closer to its boundary. (d) Moving dirt from front-left to back-left zone. (e) End of the second lane, soil is accumulated behind the first excavated lane. (f) Dirt is moved from the front and front-right zone to the back-right zone, which is now reachable. (g) Dumping on the right side. (h) Final result of the excavation.

Hence, while drawing a direct comparison between the system and the human operator is tempting, the multitude of influencing factors suggests a nuanced evaluation is necessary.

Table 5 details efficiency metrics based on excavation area and modes. The overall times include driving, digging, and refinement. The total volume is determined by comparing the initial map with the map after excavation in each workspace. The volume excavated in the front area represents the amount of undug dirt in the pit, whereas the volume moved in front-side regions corresponds to piles of dirt that need further relocation.

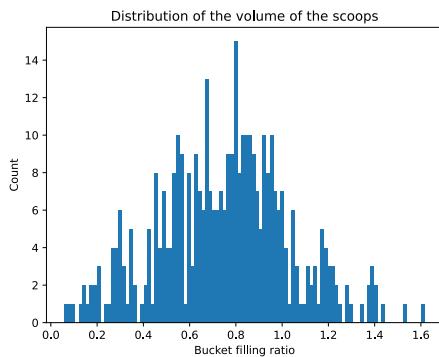
To better understand the scooping efficiency, we analyzed the distribution of soil volume dug for each scoop. A total of 380 scoops were executed during the operation, excluding bucket motions used for refining the dig area. The distribution of scoop volume, shown in Fig. 24, appears to follow a Gaussian distribution with a mean of  $0.45 \text{ m}^3$  and a standard deviation of 0.16. The bucket volume is approximately  $0.6 \text{ m}^3$ , but it can be overfilled for more efficient digging. Lower volume scoops are often used to remove the last remaining soil from the dig area or result from constraints on digging geometry, such as being close to the edges of the pit with the bucket.

**TABLE 5.** Breakdown of efficiency metrics.

Dig Area	Duration [h]	Mean Scoop Vol. [m <sup>3</sup> ]	Vol. [m <sup>3</sup> ]	Dig Eff. [m <sup>3</sup> /h]
Front	3h 20m	0.43	142.47	42.70
Front-Left	0h 17m	0.55	16.18	54.72
Front-Right	0h 0m	-	-	-
Refinement	0h 24m	-	-	-
Overall including Loose Dirt	4h 25m	-	158.77	35.95
Overall	4h 25m	-	142.47	32.35

**TABLE 6.** Digging precision statistics in centimeters.

Condition	Mean Level	Abs. Err.
Before ref.	-3.5	9.7
After ref.	-1.3	7.2


**FIGURE 24.** Distribution of the scoops' volume expressed as a ratio of the bucket volume (0.6 m<sup>3</sup>).

The summary of the excavation precision is outlined in Table 1 (formerly referred to as Table 6). The mean level metric represents the average height difference to the desired elevation. Significant inconsistencies in the flatness of the pit's bottom are evident from the values presented. Before refinement, the average absolute error measures around 10 cm. However, through refinement, this error was successfully reduced by 3 cm.

Several factors hinder finer precision, such as unidirectional grading, sensor noise from the LiDAR system, and the GPS's height covariance. Moreover, imprecise tracking from using a singular set of low-level PID gains to manage arm velocity causes potential errors in varied arm configurations during digging and grading.

### C. DIFFERENT EXCAVATION GEOMETRIES

The system generates plans for various excavation geometries, such as carving the letters H, T, and E on the ground. To protect the edges of the letters, a nonpermanent dump area surrounds each letter. Fig. 25 displays the snapshots of the excavation process using the same map as in Fig. 12.

### D. EFFECTS OF DIFFERENT EXCAVATION CONSTRAINTS

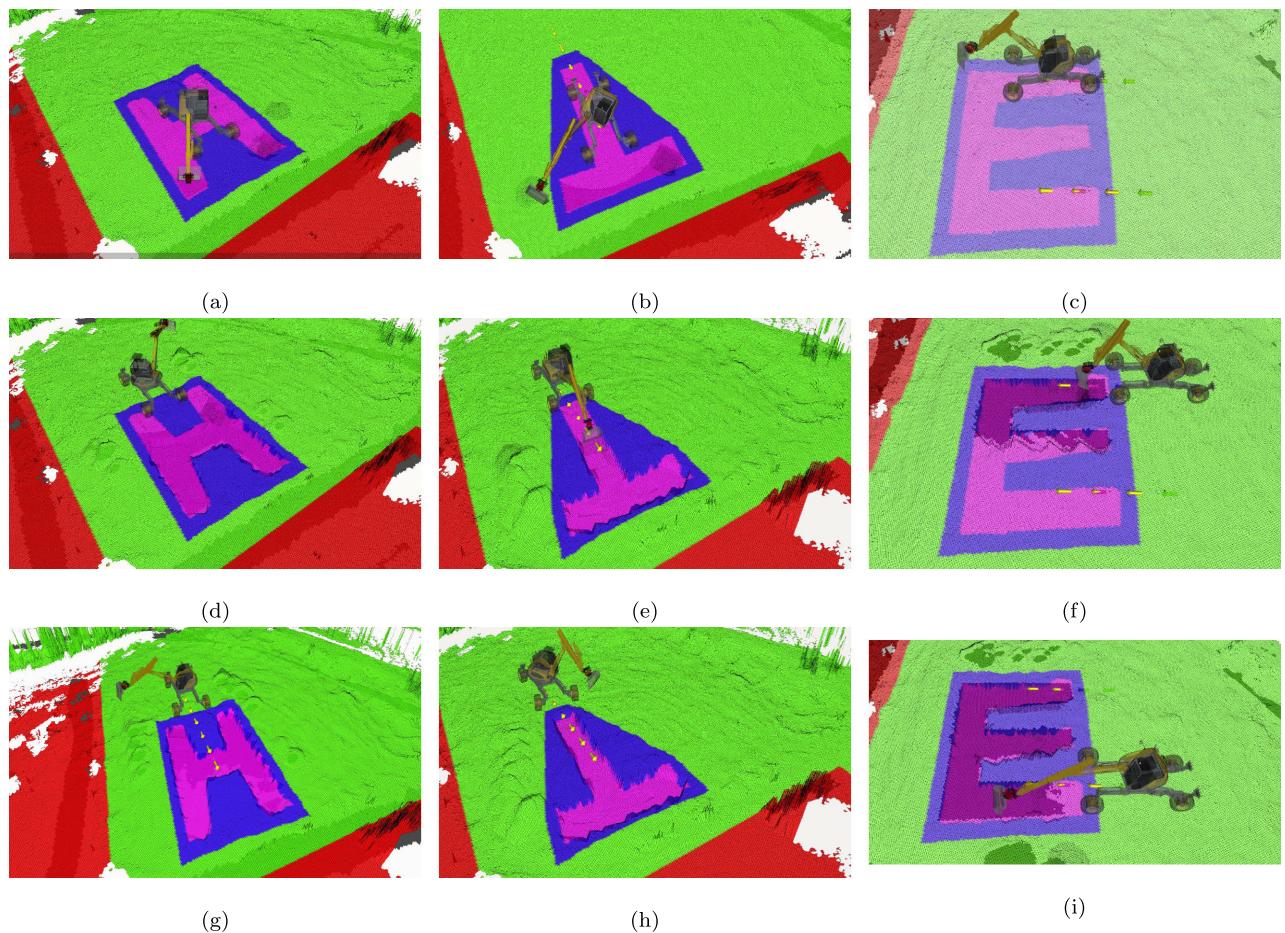
This section provides an overview of different excavation scenarios under varied dumping constraints, shown in the video supplement. In the first scenario, the back of the first lane is designated as a no-dump area. Two feasible solutions exist, starting at either corner 1 or 2 (see Fig. 12). The solution starting at corner 2 is selected due to its lower cost and efficient excavation pathway (see Fig. 26). In the second scenario, permanent dump areas behind all lanes are disallowed, making the excavation paths from both corners longer. Temporary dumping in the back of the first lane precedes the transfer of soil to permanent dump areas [see Fig. 27(a) and (b)]. In the third scenario, areas behind the lanes are assigned as nondumping zones. Without any accessible dump areas, the excavator ultimately becomes trapped, illustrating the limitations of the current planning method [see Fig. 27(c) and (d)]. Finally, in the fourth scenario, the side of the first lane is off-limits for dumping. The excavator is forced to temporarily dump the soil inside the pit, making the corner 2 solution infeasible. The corner 1 solution is preferred, and subsequent soil handling follows standard practices [see Fig. 27(f) and (e)].

### E. EXPERIMENTAL INSIGHTS

This section highlights the important lessons learned during the preparation and execution of the experimental campaign that were not obvious beforehand.

#### 1) ENSURE ROBUST NAVIGATION

Having a robust and accurate navigation stack is critical for the success of the experiment. The navigation stack should ensure precise positioning of the base while simultaneously ensuring the safety of the machine against potential falls in the excavation site or collisions. It is visually challenging to distinguish between excavated and unexcavated areas or soil piles due to the uniformity of color and texture. Employing a depth sensor, such as LiDAR, is essential, although it still does not allow for accurate estimation of soil height directly in front of the machine while excavating or for precise estimation of local workspace edges. Proprioception and explicit knowledge of workspace dimensions are needed to temporarily mark these areas off for navigation, especially maintaining the correct lateral clearance from the previous workspace when initiating a new lane. Softer soils



**FIGURE 25.** Snapshots of the letters' excavation process in simulation, the target depth is 0.7 m. Outside of no dumping close to the letters' edges, no further dumping constraint has been added. (a), (d), and (g) H letter. (b), (e), and (h) T letter. (c), (f), and (i) E letter.

require larger clearances due to the increased risk of wall collapse.

## 2) MANAGE WORKSPACE EDGES CAREFULLY

Handling workspace edges effectively requires aligning the base orientation and positioning along the edge. This alignment allows the machine to drive backward while excavating and maintain high excavation quality along the edge. Accurate positioning and heading are necessary to excavate the edges appropriately. Introduce recovery behaviors to correct the machine's base pose in cases of misalignment, which often occurs due to wheel slippage and difficulties in tracking a path moving over loose soil.

## 3) BACKUP ENVIRONMENT STATE REGULARLY

It is critical, both during development and deployment, to regularly back up the state of the robot and the environment. Without saving the state (robot state and current map), it becomes challenging to resume operations at the same point or to replicate a failed case after an interruption. Sensor noise can invalidate a previously considered finished workspace, and LiDAR occlusions can obscure

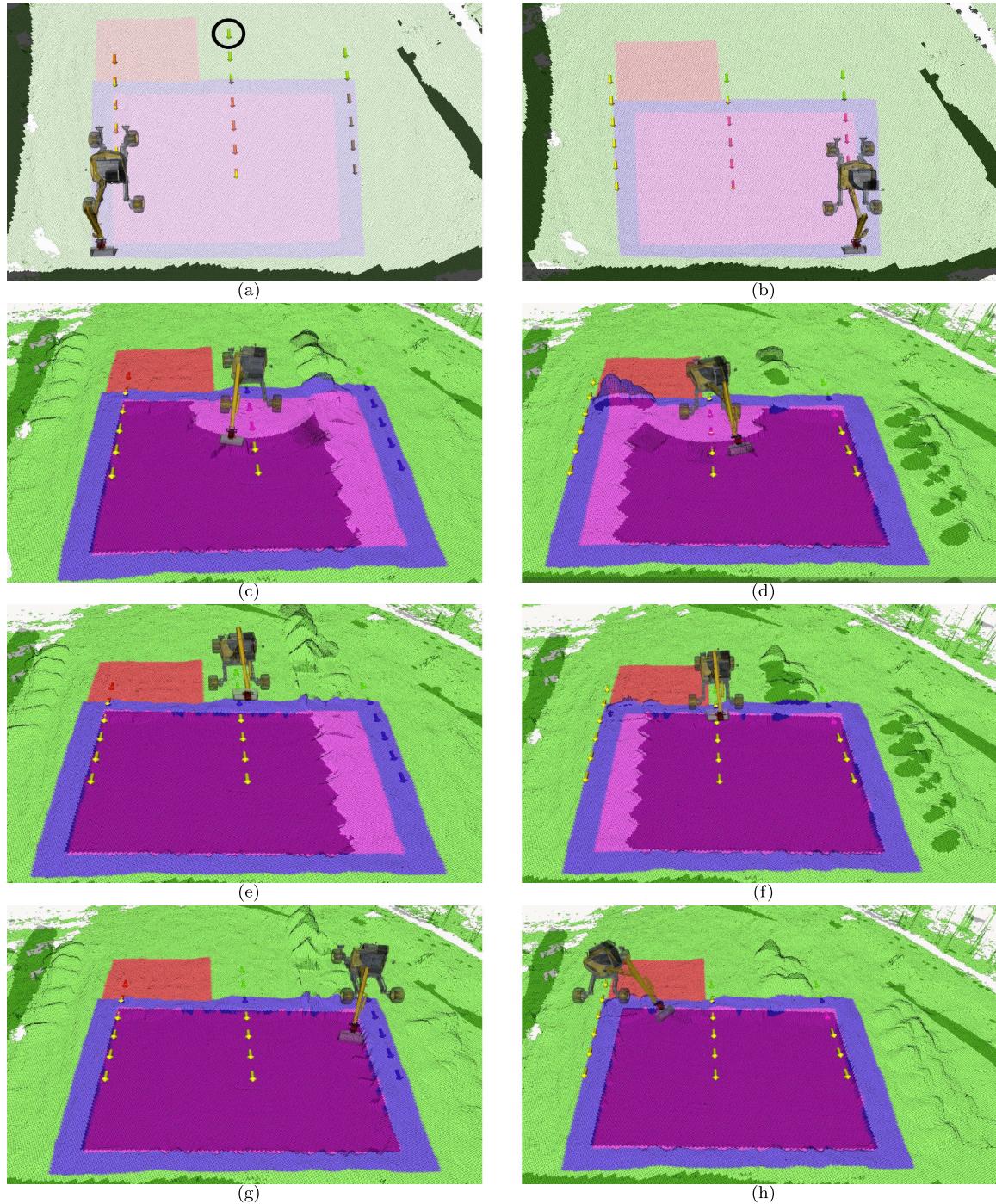
whether the current workspace is completed, leading to numerous empty scoops until the map is updated via proprioception. For navigation, properly marking off excavated parts to define terrain traversability is also complex due to occlusions.

## 4) DEFINE CLEAR STOPPING CRITERIA FOR DIGGING

Deciding when to stop excavating a workspace and move on to the next one is a key challenge for the digging planner. The primary parameter used for this decision is the fraction of soil left to excavate in the current workspace. However, accurately estimating the remaining soil volume is difficult due to errors in state estimation and noise in depth sensor measurements. Employing a low-noise, high-density LiDAR, such as one with 128 rays, can assist, but some errors are inevitable, leading to diminishing returns in enhancing accuracy for stopping criteria.

## 5) ENSURE STATE ESTIMATION OPERATES OFFLINE

Achieving a high level of robustness with minimal interventions is important. The state estimator should not rely solely on global information, such as RTK-GPS, which we found



**FIGURE 26.** Two feasible simulated plans side by side. (a), (c), (e), and (g) Plan that starts at corner 1. The extra base pose necessary to complete the workspace is circled. (b), (d), (f), and (h) Plan starting at corner 2.

unreliable with frequent outages. Depending on local information, such as LiDAR scans or visual features, is essential to prevent state estimator drift when GPS covariance spikes.

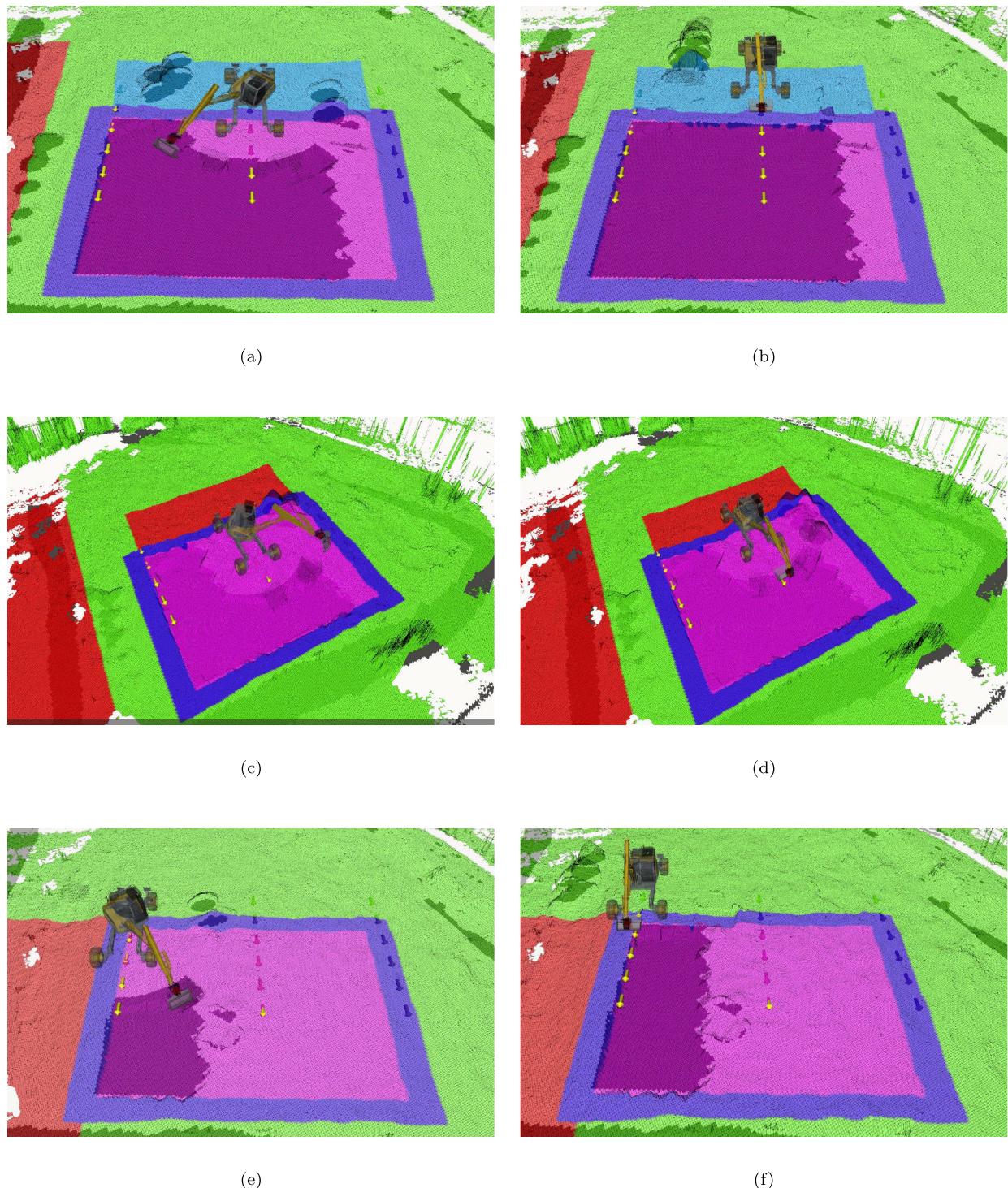
#### 6) FLATTEN THE WORKSPACE FOR STABILITY

Ensuring a flat platform from which to dig is important for operational stability. Our chassis balancing controller aids in keeping the base gravity-aligned by adjusting the leg joints

of our legged excavator. For a tracked excavator, include a subroutine in the state machine or behavior tree to ensure a flat workspace.

#### 7) CAREFULLY DEFINE LOCAL WORKSPACE BOUNDARIES

Handling loose soil differs significantly from direct excavation. It necessitates a different digging planner and tuning



**FIGURE 27.** Excavation scenarios. (a) Soil from the central lane is dumped in the light blue area (temporary dump location). (b) Excavating the middle lane, the soil is starting to accumulate at the pit's edge. (c) Soil has been removed to a permanent dump location. (d) Moment where the local planner failed. There is no space available to dump soil inside the pit, and the dumping area outside the red area is too far away. (e) Simulation of the excavation of a pit with the no-dump area adjacent to the left side of the pit. The soil from the first lane is dumped inside the pit. (f) Soil from the first lane is dumped in a permanent dump area.

the boundaries of the workspace to prevent soil spillage over or adjacent to the machine, complicating navigation to the next waypoint set by the global planner. Avoid throwing soil beside the machine as it becomes unreachable.

#### 8) SIMPLIFY THE STATE MACHINE FOR ENHANCED ROBUSTNESS AND SIMPLICITY

We recommend simplifying the state machine as much as possible to improve system reliability and reduce the complexity

of transition logic, which can be a source of bugs. Instead of implementing subroutines that handle edge cases, such as scooping again without dumping if the bucket is not full, focus on enhancing the digging and arm controllers. Prioritize system reliability and simplicity over performance enhancement at this stage. Making the system functional should be the first goal before optimizing processes.

## VI. CONCLUSION AND DISCUSSION

This study presents a fully autonomous system for excavation planning and execution. Using a 12-ton excavator, we demonstrate the ability of the system to dig a pit measuring  $15.6 \times 11.5 \times 1$  m in a total of 4 h and 25 min. The system has an average cycle time of about 32 s and can move  $42.7 \text{ dm}^3$  of soil per hour, with a final grade error of 7.2 cm on average.

The global planner tackles the excavation problem by finding a set of base poses that allow the excavator to dig through the excavation area. Using boustrophedon decomposition, it first decomposes the excavation site into cells with simple navigation patterns. It then uses a tree search algorithm to find the optimal order of the cells and minimize the total travel cost, ensuring the feasibility of the excavation. The plan is then completed with dynamic programming to determine the start and end points of the excavator's path for each cell. The local excavation planner determines how to move the soil around the excavator for each base pose. It targets moving the dirt from a reachable digging area into an allowed dumping area. The digging planner uses a Bayesian optimizer to choose the parameters of digging trajectories that maximize the scooped volume in the target workspace. The navigation between base poses is executed by an RRT\* sampling-based planner, minimizing travel costs. We utilize Google Earth Pro for target geometries, though professional architecture software is a potential future improvement.

The current global planner and simple zigzag subroutines restrict the range of possible plans. Future work could integrate a system for navigation and local excavation planning, potentially employing RL. Challenges include creating an abstract yet practical environment for sim to real transfer.

One limitation of the current digging planner is that it uses a greedy optimization approach, which can result in trajectories that do not minimize the number of scoops required to complete the local workspace. This limitation can be addressed by using a nongreedy optimization approach. In addition, simulation speed advances could enable the training of model-free RL agents that do excavation planning based on elevation map data [27]. Instead of going end to end, another possibility is to use a lower level policy, such as the one described in [7], and train higher level policy that conditions it to dig on target excavation locations. These approaches could lead to more efficient and effective excavation planning in the future.

Overall, addressing these limitations could lead to a more effective and versatile excavation system in the future.

## REFERENCES

- [1] E. U. Acar, H. Choset, A. A. Rizzi, P. N. Atkar, and D. Hull, "Morse decompositions for coverage tasks," *Int. J. Robot. Res.*, vol. 21, no. 4, pp. 331–344, Apr. 2002.
- [2] D. A. Bradley and D. W. Seward, "Developing real-time autonomous excavation—The LUCIE story," in *Proc. 34th IEEE Conf. Decis. Control*, vol. 3, Aug. 1995, pp. 3028–3033.
- [3] United States Bureau of Labor Statistics, "The economics daily: A look at workplace deaths, injuries, and illnesses on workers' memorial day," U.S. Dept. Labor, Washington, DC, USA, Apr. 2022. Accessed: Feb. 12, 2023. [Online]. Available: <https://www.bls.gov/opub/ted/2022/a-look-at-workplace-deaths-injuries-and-illnesses-on-workers-memorial-day.htm>
- [4] H. Cannon, "Extended earthmoving with an autonomous excavator," School Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU-RI-TR-99-10, May 1999.
- [5] C. Cao, J. Zhang, M. Travers, and H. Choset, "Hierarchical coverage path planning in complex 3D environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Paris, France, May 2020, pp. 3206–3212.
- [6] H. Choset, "Coverage of known spaces: The boustrophedon cellular decomposition," *Auto. Robots.*, vol. 9, no. 3, pp. 247–253, 2000.
- [7] P. Egli, D. Gaschen, S. Kerscher, D. Jud, and M. Hutter, "Soil-adaptive excavation using reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 9778–9785, Oct. 2022.
- [8] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robot. Auto. Syst.*, vol. 61, no. 12, pp. 1258–1276, Dec. 2013.
- [9] T. Groll, S. Hemer, T. Ropertz, and K. Berns, "Autonomous trenching with hierarchically organized primitives," *Autom. Construct.*, vol. 98, pp. 214–224, Feb. 2019.
- [10] M. Hutter et al., "Force control for active chassis balancing," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 2, pp. 613–622, Apr. 2017.
- [11] E. Jelavic, D. Jud, P. Egli, and M. Hutter, "Robotic precision harvesting: Mapping, localization, planning and control for a legged tree harvester," *Field Robot.*, vol. 2, no. 1, pp. 1386–1431, Mar. 2022.
- [12] E. Jelavic, J. Nubert, and M. Hutter, "Open3D SLAM: Point cloud based mapping and localization for education," in *Proc. Robotic Perception Mapping, Emerg. Techn., ICRA Workshop*, 2022, p. 24.
- [13] R. L. Johns et al., "Autonomous dry stone," *Construct. Robot.*, vol. 4, nos. 3–4, pp. 127–140, Dec. 2020.
- [14] D. Jud, G. Hottiger, P. Leemann, and M. Hutter, "Planning and control for autonomous excavation," *IEEE Robot. Autom. Lett.*, vol. 2, no. 4, pp. 2151–2158, Oct. 2017.
- [15] D. Jud, I. Hurkxkens, C. Girot, and M. Hutter, "Robotic embankment," *Construct. Robot.*, vol. 5, no. 2, pp. 101–113, Jun. 2021.
- [16] D. Jud et al., "HEAP—The autonomous walking excavator," *Autom. Construct.*, vol. 129, Sep. 2021, Art. no. 103783.
- [17] D. Jud, P. Leemann, S. Kerscher, and M. Hutter, "Autonomous free-form trenching using a walking excavator," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 3208–3215, Oct. 2019.
- [18] S. Khattak, H. Nguyen, F. Mascarich, T. Dang, and K. Alexis, "Complementary multi-modal sensor fusion for resilient robot pose estimation in subterranean environments," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Sep. 2020, pp. 1024–1029.
- [19] S.-K. Kim, J. Seo, and J. S. Russell, "Intelligent navigation strategies for an automated earthwork system," *Autom. Construct.*, vol. 21, pp. 132–147, Jan. 2012.
- [20] G. Kumparak. (2022). *Built Robotics Raises Another \$64M To Make Construction Equipment Autonomous*. [Online]. Available: <https://techcrunch.com/2022/04/01/built-robotics-raises-another-64m-to-make-construction-equipment-autonomous/>
- [21] D. Lee, I. Jang, J. Byun, H. Seo, and H. J. Kim, "Real-time motion planning of a hydraulic excavator using trajectory optimization and model predictive control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 2135–2142.
- [22] Leica Geosystems. (2023). *Leica Geosystems—Laser Scanners*. [Online]. Available: <https://leica-geosystems.com/products/laser-scanners>
- [23] O. Litvin and Y. Litvin, "Evaluation of effect of the excavator cycle duration on its productivity," in *Proc. E3S Web Conf.*, vol. 174, 2020, p. 1010.
- [24] McKinsey. (2016). *Beating the Low-productivity Trap: How To Transform Construction Operations*. [Online]. Available: <https://www.mckinsey.com/capabilities/operations/our-insights/beating-the-low-productivity-trap-how-to-transform-construction-operations>

- [25] McKinsey. (2022). *Addressing the U.S. Construction Labor Shortage*. [Online]. Available: <https://www.mckinsey.com/industries/public-and-social-sector/our-insights/will-a-labor-crunch-derail-plans-to-upgrade-us-infrastructure>
- [26] J. Nubert, S. Khattak, and M. Hutter, “Graph-based multi-sensor fusion for consistent localization of autonomous construction robots,” in *Proc. Int. Conf. Robot. Autom. (ICRA)*, Philadelphia, PA, USA, May 2022, pp. 10048–10054.
- [27] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, “Learning to walk in minutes using massively parallel deep reinforcement learning,” in *Proc. 5th Conf. Robot Learn.*, 2022, pp. 91–100.
- [28] J. Seo, S. Lee, J. Kim, and S.-K. Kim, “Task planner design for an automated excavation system,” *Autom. Construct.*, vol. 20, no. 7, pp. 954–966, Nov. 2011.
- [29] S. Silvestri, G. Laporte, and R. Cerulli, “A branch-and-cut algorithm for the minimum branch vertices spanning tree problem,” *Comput. Oper. Res.*, vol. 81, pp. 322–332, May 2017.
- [30] S. Singh and H. Cannon, “Multi-resolution planning for earthmoving,” in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 1, Aug. 1998, pp. 121–126.
- [31] S. Singh and R. Simmons, “Task planning for robotic excavation,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 2, Aug. 1992, pp. 1284–1291.
- [32] B. Son, C. Kim, C. Kim, and D. Lee, “Expert-emulating excavation trajectory planning for autonomous robotic industrial excavator,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, NV, USA, Oct. 2020, pp. 2656–2662.
- [33] A. Stentz, J. Bares, S. Singh, and P. Rowe, “A robotic excavator for autonomous truck loading,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Systems. Innov. Theory, Pract. Appl.*, vol. 3, Victoria, BC, Canada, Aug. 1998, pp. 1885–1893.
- [34] I. A. Sucan, M. Moll, and L. E. Kavraki, “The open motion planning library,” *IEEE Robot. Autom. Mag.*, vol. 19, no. 4, pp. 72–82, Dec. 2012.
- [35] Wingtra. (2023). *Wingtra—Mapping Drone WingtraOne: All Features*. [Online]. Available: <https://wingtra.com/mapping-drone-wingtraone/all-features/>
- [36] S. Woo, J. Yeon, M. Ji, I.-C. Moon, and J. Park, “Deep reinforcement learning with fully convolutional neural network to solve an earthwork scheduling problem,” in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2018, pp. 4236–4242.
- [37] J. Xie, L. R. G. Carrillo, and L. Jin, “An integrated traveling salesman and coverage path planning problem for unmanned aircraft systems,” *IEEE Control Syst. Lett.*, vol. 3, no. 1, pp. 67–72, Jan. 2019.
- [38] L. Zhang et al., “An autonomous excavator system for material loading tasks,” *Sci. Robot.*, vol. 6, no. 55, Jun. 2021, Art. no. eabc3164.

• • •