Autonomous Adaptive Modification of Unstructured Environments

Maira Saboia[1]*, Vivek Thangavelu[1]*, Walker Gosrich[1,2] and Nils Napp[1],

***Abstract*—We present and validate a property-driven autonomous system that modifies its environment to achieve and maintain navigability over a highly irregular 3-dimensional terrain. In our approach we use decision procedures that tie building actions to the terrain model, giving rise to adaptive and robust building behavior. The building algorithm is driven by continuous evaluation and reaction to terrain properties, rather than relying on a structure blueprint. This capability is essential in robotic systems that operate in unstructured outdoor or remote environments, either on their own or as part of a team. We demonstrate the effectiveness of our approach by running a low-cost robot system that can build with compliant bags in a variety of irregular terrains.**

I. Introduction

Many applications of autonomous robots require reliable operation in unstructured environments. For example: disaster response and operation in areas inaccessible to humans, such as extraterrestrial bodies, coincide with uneven terrain and a lack of established infrastructure. We present a system that can autonomously modify such environments in order to provide mobility. It can build adaptive structures over unstructured terrain that effectively incorporate terrain features. Animals provide many examples of this type of construction [1]. They modify their environment to provide protection from weather and predators, to regulate moisture and temperature, to ease traveling along foraging routes, and to provide food storage. The exact shape of the final structures adapts to fit into the pre-existing environment. For example, no two birds nests are exactly alike, but they are a robust solution to their intended function, i.e., keeping eggs warm, hidden, and in place.

In biological construction, there are a variety of ways to encode built structures. Here we focus on strategies that are particularly robust and adaptable: function-driven, iterative construction that relies on stigmergy [2]. Stigmergy is a biological phenomenon that has been adopted by robotics, referring to environmentally mediated communication where information about building actions is encoded in the partially built structure. For example, beavers build in response to the sound of moving water [3]. This cue is directly tied to the function of a dam, which should stop water from moving. The simple linking of build actions to the function of the resulting structure results in adaptive and robust building behavior.

We present a mobility-focused robotic system that autonomously builds structures to enable movement over unstructured terrain. It uses a bio-inspired mechanism of construction that is function-driven; in which the robot builds a structure by ensuring that specific conditions are held. We call these conditions *properties*. Our approach is driven by continuous evaluation and reaction to terrain properties, rather than relying on a structure blueprint. We adapt the reactive building strategy in [4] to identify non-navigable features in a 3D structure, and extend it to incorporate manipulator and motion constraints in the building strategy. The representation of build actions through motion constraints enables robots to choose valid building actions even if the building materials are complex or deformable. In contrast to other autonomous construction systems work, we use simple building materials, that are designed to work well in unstructured terrain and have direct analogs in human emergency construction [5].

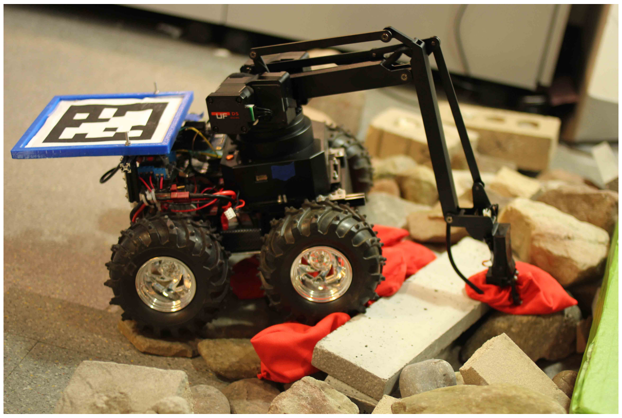

Fig. 1: Low-cost mobile manipulator made from off-the-shelf components that can handle (pickup/drop) compliant bags and navigate over unstructured terrain.

A. Contributions

The contribution of the paper is two-fold. First, we develop a property-driven construction algorithm to autonomously achieve and maintain navigability conditions over irregular 3-dimensional terrain. This approach allows building actions to be synthesized on the fly, resulting in a robust system that can autonomously adapt to a wide variety of inputs, recover from intermittent failures, and reject disturbances in the construction process. The algorithm is novel in its functionality in 3D unstructured terrain. Second, we present a fully functional autonomous robot system that demonstrates the algorithm in an experimental setup that closely resembles real-world applications. The robot uses compliant bags as a building material for the construction and maintenance of a navigable path to a given target location, over a highly irregular terrain.

* Both authors contributed equally to this work.

[1] Department of Computer Science and Engineering, [2] Department of Mechanical and Aerospace Engineering, University at Buffalo, Buffalo, NY 14260, USA

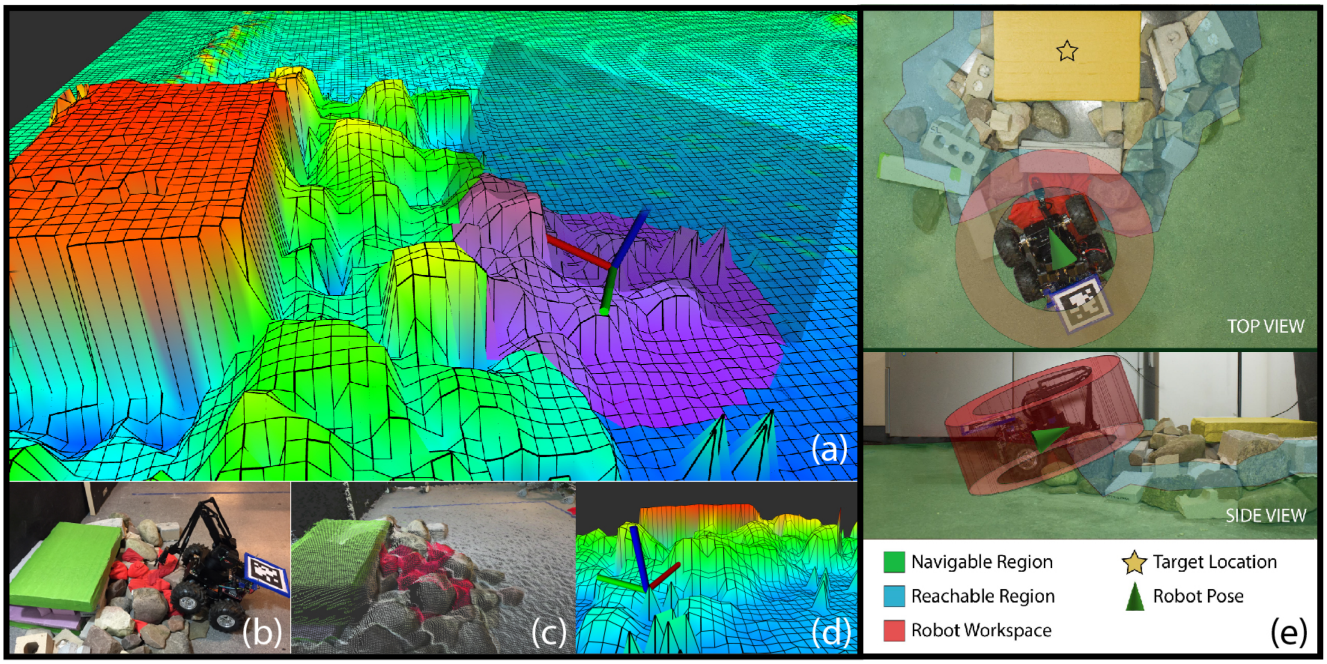

Fig. 2: Navigability parameters depicted in 3D (a-d) and 2D(e) spaces. (a) depicts the 3d height field of the ground plane. The purple region is the robot footprint of diameter d. (b) and (c) depict the RGB and the RGBD images of the environment, and (d) depicts the height field from the robot POV. (e) depicts the navigable and reachable regions.

B. Related Work

Nature-inspired construction systems based on stigmergy have been explored both from an algorithmic perspective [6] and in physical implementation systems [7] since the mid 1990s. The key idea is that the building plan is encoded in rules that agents use when they respond to intermediate build states. These rules in turn define the next intermediate states, which then triggers more rules. This process is thought to govern coordination in swarm constructions systems [8], e.g. termite mounds, where no individual agent has an internal estimate of the structure. Work that uses this approach to enable construction, often focuses on stigmergy as a process to coordinate agents in a distributed system, e.g. [9, 10, 11, 12, 13]. Here we focus on two other properties that many natural construction systems exhibit that stigmergic approaches can provide: adaptability and robustness.

Using this approach is difficult in practice since synthesizing stigmergic rules to build specific structures is challenging, and the contribution of works such as [10] and [12] is to generate rule sets that can build specific shapes. In order to fully exploit the adaptability we also use a functional specification to describe the target structures. This has been explored in a theoretical setting in [14], and is given as an explanation for animal built structures that have direct biological functions [2].

Werfel and Petersen use stigmergy in [10] to build a 3D structure inspired by mound-building termites [11], where a team of identical robots receive a set of low-level rules that collectively produce a specific structure using prefabricated solid bricks. While navigating over the structure, the robots attach bricks in positions that at the same time obey a set of *geometric requirements* and are valid according to the structure plan. Allwright used the same mechanism to conceive the stigmergic block [12]. In this work, the robots ”mark” the blocks by using NFC (Near Field Communication) to toggle LEDs. The patterns left on the blocks can stimulate construction actions [15]. Although they have a purely reactive model of construction, this approach relies on carefully handcrafted rules and specialized construction materials.

Melenbrink shows in [13] the ability to build unsupported cantilevers across a gap using force feedback. By looking at how loads transfer to the ground they can anticipate and possibly prevent tipping over by building a structure for counterbalancing. From the sensor readings, each robot can also determine whether another robot is located nearby. The approach described is presented in simulation and is limited to 2D construction.

In Soleymani's work [16], a self-contained ground robot builds a protective barrier using bags (called compliant pockets). The final shape of the structure is specified via a template and the probability of a certain location to be chosen for deposition is inversely proportional to the *number pockets* in that location. The work focuses on scalability, and demonstrates that their approach work by experiments and extensive simulation. The system is limited to working in structured environments, and the resulting structures cannot be used by the robot. While this approach clearly demonstrates the use of compliant materials, it does not take advantage of their ability to be easily used on irregular terrain.

Similarly Fujisawa's work describes a robot that can modify its environment by depositing foam, which expands and turns rigid. Their work gives a demonstration system, that is able to build structures over regular obstacles, but do not give correctness proofs for their deposition strategy [17].

A complementary approach to enabling mobility in unknown environments is presented in [18]. In this work a high-level task planner coordinates modular robots that plan and execute augmentation tasks. This system is autonomous and adaptive, in that it can generate new plans in response to new environments. However, generating task primitives in unstructured terrain is difficult.

In the work present by Napp in [4], a robot was used to autonomously construct ramps with foam in an unstructured environment. This work focused on a strategy for adaptive ramp building using a reactive algorithm that iteratively fills *non-navigable* gaps and ledges in the ramp structure. However, this work does not take place in a full 3D state space, and does not incorporate manipulation and motion plans.

The rest of the paper is organized as follows: Section II discusses the theoretical background and proposed approach. Section III describes the robot system followed by the various implementation aspects of our work in Section IV. Section V showcases the experiments and the subsequent discussions are presented in Section VI. Section VII concludes the paper.

II. Building Approach

The concept of navigability proposed in [4] provides a method to evaluate whether points in a surface are navigable by a robot, taking into account the robot's kinematic constraints. We extend this concept of navigability to 3D environments and combine it with manipulation and motion constraints to determine allowable and necessary deposition actions, resulting in a structure that the robot can traverse to a final destination. Both the navigability and deposition models are defined using a small number of parameters that are matched to represent the capabilities and constraints of a physical robot.

The construction process follows a simple pipeline: a robot moves toward its destination until it encounters an irregularity in the environment that hinders its progress. Based on its physical capabilities, the direction it is heading and the information of its surroundings, the robot chooses where to modify the structure (by placing bags) to enable its continued navigation. After deposition, these bags are treated as part of the terrain.

In this section we first present the system assumptions along with the notations used throughout the paper. We then introduce the mathematical model of construction, followed by the concepts of *navigability* and *reachability*. After defining the *deposition strategy*, we combine all of the above concepts and present the construction algorithm in a 3D domain.

A. Assumptions and Notations

The robot is a mobile manipulator capable of material pickup and deposition, and moving over non-flat terrain. It is assumed that the manipulator base is mounted parallel to the mobile base. The robot position (or pose) refers to the center of mass of the mobile base. The robot footprint is considered to be a circular region of diameter δ, centered at the robot position. This allows us to abstract the robot orientation and simplify deposition planning.

The robot is deployed at a position that is navigable, and from where it has access to an unlimited supply of building materials. The robot utilizes a global map view of the world for motion planning, while construction-related planning decisions are based on local conditions. As it acquires new information about the environment throughout the building process, it updates its representation of the structure and its surroundings.

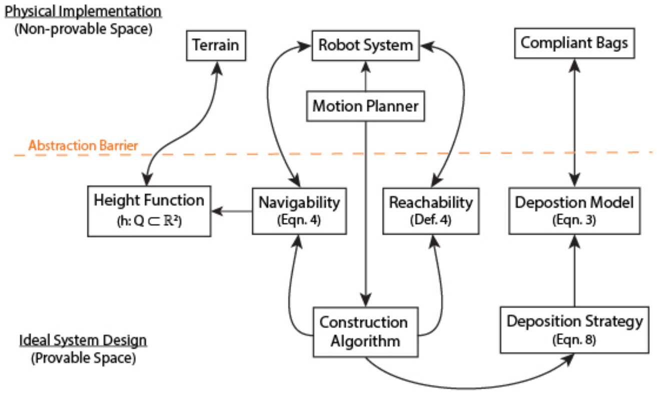

Fig. 3: System Overview showcasing the inter-dependencies among various components. The dotted line separates the ideal system model from the physical implementation. Double-ended arrows depict that the components on either side conform to each other. Single-ended arrows depict the component at the tail of the arrow depends on the component at the arrowhead.

In this paper, bold lowercase letters are used to represent vectors, and their individual components are represented by lowercase letters with subscripts. Physical system parameters are denoted by lowercase Greek letters. All quantities are defined in S.I. units. A robot pose is represented by $\langle \mathbf{p_t}, \mathbf{p_r} \rangle \in SE(3)$, such that $\mathbf{p_t} \equiv \langle p_x, p_y, p_z \rangle \in \mathbb{R}^3$ is the rover position and $\mathbf{p_r} \equiv \langle p_\varphi, p_\theta, p_\psi \rangle \in SO(3)$ is the rover orientation. Here, p_x, p_y, p_z denote the x, y, and z components of the translation and $p_\varphi, p_\theta, p_\psi$ denote roll, pitch and yaw components of the orientation, respectively. $SO(3)$ denotes the special orthogonal group and $SE(3)$ is the special euclidean group in 3D. A stable robot pose is a pose at which the robot is statically stable on the structure surface. The *robot plane* at $\langle \mathbf{p_t}, \mathbf{p_r} \rangle$ is the rectangular plane of reference with an orientation of $\mathbf{p_r}$.

B. Mathematical Model

We base our building implementation atop the model of construction presented in [4], which provides a method to tie the robot's kinematic constraints to an arbitrary structure shape and a concise mathematical way to express a set of poses that the robot can occupy.

Consider an abstract model of the mobile robot as shown in Fig. 4. $\kappa \in \mathbb{R}^+$ is the maximum climbable slope the robot can drive up or down, $\delta \in \mathbb{R}^+$ is the rover body length and $\epsilon \in \mathbb{R}^+$ is the maximum discontinuity that a robot can drive over. These parameters may be derived empirically or analytically for an arbitrary physical robot.

We use continuous functions to effectively model irregular terrain. The building area of the robot is defined by a compact, connected area $Q \subset \mathbb{R}^2$ and is the domain of a bounded, non-negative height function $h : Q \mapsto \mathbb{R}^+$. The graph of $(\mathbf{p}, h(\mathbf{p}))$, $\forall \mathbf{p} \in Q$ describes the structure surface in the building area. The navigability of two points within this area is a property, defined based upon the system and robot

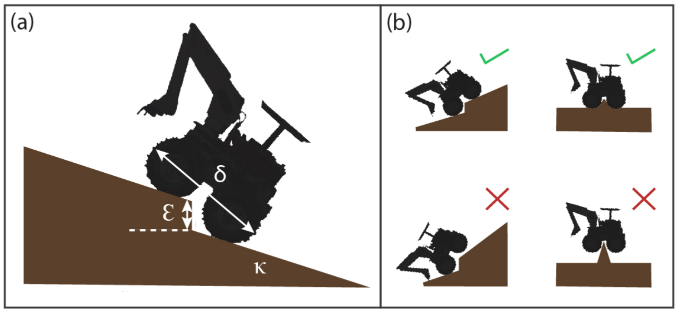

Fig. 4: (a) Depicts the robot navigability parameters in the shaded 2D plane space on a upward slope of maximum steepness κ. (b) Examples of good and bad robot configurations.

parameters. For two points on the structure surface, it is given by:

$$\kappa|\mathbf{p}-\mathbf{q}|+\epsilon \geq |h(\mathbf{p})-h(\mathbf{q})| \text{ s.t } \mathbf{p},\mathbf{q} \in Q, \tag{1}$$

where $|.|$ represents the Euclidean distance between two points. A structure is *navigable* if and only if it is locally (parameter δ) close (parameter ϵ) to K-Lipschitz continuous, and the (function) operator $P_K[h]$ projects any structure to the smallest K-Lipschitz functions that is at least as large as h:

$$P_\kappa[h](\mathbf{p}) = \max_{\mathbf{q}\in Q}\{h(\mathbf{q})-\kappa|\mathbf{q}-\mathbf{p}|\} \tag{2}$$

This operator sets the minimum amount of material needed to be deposited at a certain point to make the structure navigable. It is used in [4] to prove that given an initial structure h_0, a navigable structure can be built on Q after a finite number of depositions, just by adding material, and is bounded above by $P_\kappa[h_0]$.

The top surface of each deposition is modeled by a cone function, which provides an upper bound of how much the environment changes in response to the deposition actions, while the bottom conforms to the structure. Given an apex position $(\boldsymbol{\phi},\sigma) \in Q \times \mathbb{R}^+$ and steepness $\kappa_D \in \mathbb{R}^+$, the deposition function at a point $\mathbf{p} \in Q$ is defined as:

$$f_{(\boldsymbol{\phi},\sigma)}(\mathbf{p}) = \sigma - \kappa_D|\boldsymbol{\phi}-\mathbf{p}| \tag{3}$$

However, physical depositions may not be perfect cones. In fact, as long as the deposition, defined by some arbitrary continuous shape function, is bounded above by a cone with slope greater than κ and maximum deposition height less than ϵ, the proofs presented in [4] guarantee no depositions accidentally make intermediate structures larger than $P_\kappa[h_0]$. Through such bounding cones, the model allows for uncertainty in the exact deposition shape, location and volume.

Further, the robot model conforms to the constraints $(11-13)$ in Section 4.1 of [4], which limit the physical parameters of the robot and deposition to ensure that successful building towards navigation is possible.

C. Navigability and Reachability

In this section, we define what it means for a point to be navigable or reachable by the robot. We derive the orientation limits of valid rover poses in Lemma 1 and a general workspace for the manipulator in Lemma 2. Finally, using the above concepts, we define navigable and reachable regions that are used for the progress of the adaptive building algorithm.

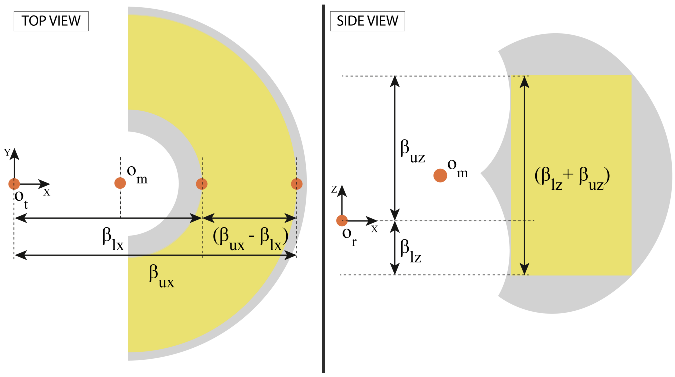

Fig. 5: Manipulator's Workspace in the XY plane (Top view) and YZ plane(Side view) of the robot frame of reference. The gray region depicts the arm's true workspace while the yellow region depicts the reduced region in consideration. β parameters are in the robot frame. In our case, $\beta_{lx} = 0.18$, $\beta_{ux} = 0.34$, $\beta_{ax} = -\pi/2$, $\beta_{bx} = \pi/2$, $\beta_{lz} = -0.12$, $\beta_{uz} = 0.15$.

Definition 1. Point $\mathbf{p} \in Q$ is said to be *navigable* in a structure $(\mathbf{p},h(\mathbf{p}))$, $\forall \mathbf{p} \in Q$, if and only if it satisfies the navigability feature (1) for every two points that are at a distance of at most $\delta/2$ from $\mathbf{p}$, i.e

$$\kappa|\mathbf{r}-\mathbf{q}|+\epsilon \geq |h(\mathbf{r})-h(\mathbf{q})| \;;\; \forall(\mathbf{r},\mathbf{q}) \in B_{\delta/2}(\mathbf{p}) \tag{4}$$

$B_{\delta/2}(\mathbf{p})$ is the set of all points that are at a distance of at most $\delta/2$ from $\mathbf{p} \in Q$ i.e. $B_{\delta/2}(\mathbf{p}) = \{\mathbf{s} \in Q \;;\; |\mathbf{s}-\mathbf{p}| \leq \delta/2\}$.

Lemma 1. *If point $\mathbf{p} \in Q$ is said to be navigable, then the orientation of any stable pose $\langle \mathbf{q_t}, \mathbf{q_r}\rangle$ of the robot at $\mathbf{p}$ is bounded as follows:*

$$\begin{aligned} |q_\varphi| &\leq \tan^{-1}(\kappa) \\ |q_\theta| &\leq \tan^{-1}(\kappa) \end{aligned} \tag{5}$$

Proof: As navigable point $\mathbf{p}$ satisfies condition (4), any stable pose $\langle \mathbf{q_t}, \mathbf{q_r}\rangle$ of the robot at $\mathbf{p}$ must satisfy $|q_\varphi| \leq \tan^{-1}(\kappa)$ and $|q_\theta| \leq \tan^{-1}(\kappa)$. Since navigability is based on the fact that the rover base is modeled as a circular footprint, changing the yaw of the robot will not affect the validity of navigability of $\mathbf{p}$ and thus q_ψ is unbounded. ■

In the following derivations, we make a reasonable assumption that there is no relative rotation between the coordinate frames of the mobile base and the manipulator. Incorporating a relative rotation between the frames would simply require an additional rotational transformation component to be considered. Figure 3(a) and 3(b) depict a generic manipulator's workspace in the robot's frame of reference. We restrict the true workspace of the manipulator, shown in gray, to the regions shown in yellow. The parameters β_{lx} and β_{ux} are the minimum and maximum displacements, respectively, in the XY plane of the robot frame, and β_{lz} and β_{uz} are the minimum and maximum displacements, respectively, in the YZ plane of the robot frame. These parameters are chosen based on the manipulator's specifications and its mounting position on the rover base. This kind of a conservative yet general definition of a manipulator's workspace with respect to the robot pose simplifies the computation of the workspace regions of the robot. The *robot workspace* at a navigable position $\mathbf{p}$ is the set of points in $\mathbb{R}^3$ that can be reached from $\mathbf{p}$ (shown in Fig.5).

Definition 2. The *robot workspace* at a navigable position $\mathbf{o_t}$ is defined by the circular, hollow cylinder of height $(\beta_{uz} - \beta_{lz})$ and radii ranging from β_{lx} to β_{ux}, oriented along an axis $\hat{\mathbf{n}}$ perpendicular to the robot plane. It is defined by the region:

$$\begin{aligned} \beta_{lx} \leq |(\mathbf{q_t} - \mathbf{o_t}) \times (\hat{\mathbf{n}})| \leq \beta_{ux} \\ -\beta_{lz} \leq (\mathbf{q_t} - \mathbf{o_t}) \cdot (\hat{\mathbf{n}}) \leq \beta_{uz}, \end{aligned} \tag{6}$$

where $(\times)$ represents vector cross product and $(\cdot)$ represents vector dot product.

Lemma 2. *All the points in $\mathbb{R}^3$ that the robot can occupy in order to reach a point $\mathbf{b} \in Q$ are bounded from above by $max\{\beta_{ux}, \beta_{uz}\}$.*

Proof: Let $\mathbf{p_r}$ be represented as the unit quaternion of a stable rover pose at point $\mathbf{p}$. Then, the maximum reach in the x and z directions are given by $\mathbf{p'_r} \cdot (\boldsymbol{\beta}_{ux}) \cdot \mathbf{p_r}$ and $\mathbf{p'_r} \cdot (\boldsymbol{\beta}_{uz}) \cdot \mathbf{p_r}$, each upper bounded by the distance $max\{\beta_{ux}, \beta_{uz}\}$, where $(\cdot)$ and $(')$ represents the vector dot product and the inverse of the quaternion vector, respectively. $\boldsymbol{\beta}_{ux}$ and $\boldsymbol{\beta}_{uz}$ denote the vector notations of β_{ux} and β_{uz} , respectively. Inversely, the maximum distance from the destination point $\mathbf{b}$ from where the robot can reach $\mathbf{b}$ is $max\{\beta_{ux}, \beta_{uz}\}$. ■

Definition 3. The *navigable region* N is defined as the connected space such that $N \subseteq Q$, $\mathbf{p_0} \in N$ and $\forall \mathbf{p} \in N$ satisfying condition (4). $\mathbf{p_0}$ is the initial, navigable robot position before construction.

Definition 4. A point $\mathbf{b} \in Q$ is said to be *reachable* by the robot if there exists at least one navigable position $\mathbf{p} \in N$ within a distance $max\{\beta_{ux}, \beta_{uz}\}$ such that condition (6) is satisfied. The *reachable region R* is the set of all points that are *reachable*.

Figure 2(e) depicts the navigable (in green) and reachable regions (in blue).

D. Deposition Strategy

Making a point $\mathbf{q}$ navigable requires modifying it or its surroundings through depositions such that $\mathbf{q}$ satisfies condition (4). According to this condition, the navigability of $\mathbf{q}$ is influenced only by the pairs of points in $B_{\delta/2}(\mathbf{q})$. Hence, we define the *deposition cost* for a pair of non-navigable points $\mathbf{r}, \mathbf{s} \in B_{\delta/2}(\mathbf{q})$ as:

$$C(\mathbf{r}, \mathbf{s}) = |h(\mathbf{r}) - h(\mathbf{s})| \tag{7}$$

Using the cost function, we can choose the deposition position $\mathbf{b} \in Q$ in and around point $\mathbf{q}$, first by identifying the pair of non-navigable points with the maximum height difference and then choosing the one among them with the lowest height i.e. $\forall(\mathbf{r}, \mathbf{s}) \in B_{\delta/2}(\mathbf{q})$ that violates condition (1),

$$\mathbf{b} = \underset{\mathbf{t}}{\operatorname{argmin}}\{h(\mathbf{t}); \mathbf{t} \in \underset{(\mathbf{r},\mathbf{s})}{\operatorname{argmax}}\{C(\mathbf{r}, \mathbf{s})\}\} \tag{8}$$

E. Construction Algorithm

The construction algorithm (Algorithm 1), driven by continuous evaluations and reactions to the terrain properties of the structure, leads to a series of modifications that is dependent on the inaccessible target position $p^* \notin N$ and the deposition strategy of Sec. II-D.

While $\mathbf{p}^*$ remains inaccessible (line 1), the algorithm identifies the closest navigable position $\mathbf{p} \in N$ to $\mathbf{p}^* \notin N$ (line 2) and the point $\mathbf{q}$ in the surroundings of $\mathbf{p}$ that needs to be made navigable (line 3). While $\mathbf{q}$ is not navigable (line 4), the algorithm identifies the deposition position $\mathbf{b}$ (line 5) and subsequently signals the robot to move and deposit at $\mathbf{b}$ (lines 6 and 7). After each deposition iteration (line 1 to line 7), the navigable region N grows or shrinks. The algorithm eventually steers the navigable region N to grow towards the target location until $\mathbf{p}^* \in N$.

Algorithm 1 Adaptive Construction Algorithm. Given a structure h_0 and a target location $\mathbf{p}^*$, the algorithm builds an access structure through a series of depositions based on local sensor readings, in order to obtain a navigable path to $\mathbf{p}^*$.

```
1: while $\mathbf{p}^* \notin N$ do
2:     $\mathbf{p} \leftarrow \operatorname{argmin}_{\mathbf{r}} |\mathbf{p}^* - \mathbf{r}|$ ; $\mathbf{r} \in N$
3:     $\mathbf{q} \leftarrow \operatorname{argmin}_{\mathbf{r}} |\mathbf{p}^* - \mathbf{r}|$ ; $\mathbf{r} \in \{R \wedge$ [line joining $\mathbf{p}^*$ to
               $\mathbf{p}] \wedge$ violates condition(4) $\}$
4:     while $\mathbf{q}$ is not navigable do
5:         $\mathbf{b} \leftarrow \operatorname{argmin}_{\mathbf{t}} \{h(\mathbf{t}); \mathbf{t} \in \operatorname{argmax}_{(\mathbf{r},\mathbf{s})}\{C(\mathbf{r}, \mathbf{s})\}\}$ ;
               $\forall(\mathbf{r}, \mathbf{s}) \in \{B_{\delta/2}(\mathbf{q}) \wedge$ violate condition(1) $\}$
6:         Move to position from where $\mathbf{b}$ is reachable
7:         Deposit at $\mathbf{b}$ in accordance to Eqn. 3
```

The correct behavior of Algorithm 1 is that the resulting structure is navigable after a finite number of depositions subject to the assumptions mentioned in Sec.II-A and Sec. II-B. If the above assumptions are maintained, the correctness of Algorithm 1 is directly derived from the correctness proofs of the local deposition strategy presented in [4]. In terms of correctness, the algorithm presented herein only differs in that it imposes an order to the deposition strategy based on the robot's kinematic constraints in 3D space, and thus does not invalidate the original proof.

The statements about correctness exist only upto this point in the paper. An overview of the provable and non-provable spaces and the inter-dependencies between various system components is showcased in Fig. 3. The ideal system design model conforms to the the various assumptions mentioned in Sec. II-A and Sec. II-B. Any deviation from the ideal system design that may exist in the physical implementation (refer Sec. IV) would lead to non-provable conditions.

III. Robot System

In this section we describe the various functional components of the robot construction system.

A. Building Material

The robot uses compliant bags as the building material. They are simple and inexpensive, and can be filled with sand, soil, rubble and rocks found on site. Additionally, they have favorable handling characteristics and are compliant in their adaptation to placement in uneven and unstructured terrain [5]. The compliant bags used in our experiments are filled with beans and the size of the bags vary slightly, with the mean stretched dimensions [length x width x height] and weight being $[9.3cm, 7.9cm, 0.5cm]$ $(\pm 16.26\%, \pm 20.89\%, \pm 15.32\%)$ and $120g$ $(\pm 20.6\%)$, respectively.

B. Robot Design

The robot used for the construction task (refer Fig. 1) is a low-cost mobile manipulator made from off-the-shelf components, capable of maneuvering over irregular terrain. The robot chassis is made from metal and supports four individual suspension housings for four wheels, each individually driven by a high torque DC motor forming a differential steering system. The individual suspensions and four-wheel drive facilitate stable movement over large irregularities. A PID controller and a rotary encoder are used to maintain closed-loop control for each motor. A low cost 4 DOF arm (uArm manufactured by uFactory [19]) is mounted on the front half of the metal chassis. The arm is modified to use a 1DOF generic gripper to pick up and deposit the complaint bags. The arm and the PID motor controllers are connected to an Intel Joule 570X mounted on the metal chassis, which serves as the on-board computing unit. An AprilTag [20] mounted on top of the robot is used to estimate the pose of the robot in its environment. A Kinect v2 Camera serves as an external observation system to estimate the structure surface.

IV. Implementation Considerations

In this section we present some of the implementation decisions we made with respect to the building strategy.

A. Discretization

In our system, we work on a discretized version of the workspace. In this discrete representation, we assume the construction grid area G (discretized version of Q) is a finite set of $\Delta \times \Delta$ cells $c = (i, j)$ in $\mathbb{N}^2$ parallel to the construction building area. An individual voxel is given by $v = \{c, l\}$ in $\mathbb{N}^3$, where c is the grid cell and l is its discretized height. A given continuous height function $h(q)$ can be represented as an occupancy grid in the voxel space. Each voxel v_k for which $\Delta l_k h(\Delta i_k, \Delta j_k)$ is *occupied* in the discrete representation of h. This discretized view of the world will work for the robot, even if the final structure representation is constrained by voxel resolution. The choice of discretization length should be within the discontinuity limit (ϵ) of the robot's navigability constraints i.e $\Delta < \epsilon$, and Δ should be within the minimum possible dimensions of a single deposition.

B. Bag Deposition Model

In Section II-B, we state that the use of the cone shaped deposition model can guarantee correctness for the construction algorithm. We therefore model our bag deposition by a maximum cone. A total of 40 bags were placed, in varying terrain conditions, to find the distribution for the model. Fig.6(a) and Fig.6(b) depicts the superimposition of the depositions before and after compaction, respectively. Each grid cell is a square of length 1.5cm.

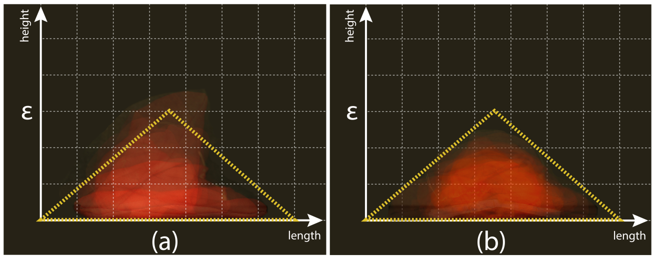

Fig. 6: Bag deposition Model from 40 different bag placements. (a) and (b) depicts the superimposition of the placed bags before and after compaction, respectively. Each grid cell is a square of side 1.5cm. Dotted yellow line depicts the cone model used.

To conform to our construction model, the maximum deposition height must be lesser than or equal to the maximum discontinuity the rover can climb (ϵ). Due to the deformable nature of the bag, the placed bags before compaction have a maximum height of 6cm which is greater than $\epsilon = 4.8$cm in our setup. When the rover moves over these bags or when more bags are placed over them, their maximum height reduces to $4.5cm$, after compaction. A cone with a height of 4.cm (3Δ) and a base diameter of 10.5cm (7Δ) is utilized in our implementation. 68.42% of the depositions before compaction and 94.73% of the depositions after compaction fit into the cone model. Considering the above cone model, we utilize a mechanism to take into account the increased bag heights before compaction. We detect such regions by using the height differences in the structure before and after a placement and treat them with higher thresholds than ϵ and κ, until the bags in those regions are compacted by the placement of another bag or when the rover drives over. The *deposition cost* for a pair of non-navigable points $\mathbf{r}, \mathbf{s} \in B_{\delta/2}(\mathbf{q})$ is modified to take into account the noise in observations by utilizing the average height values at $\mathbf{r}, \mathbf{s}$ over a neighborhood of B_γ.

The statements regarding correctness of the construction algorithm can only be made when the deposited material always conforms to the cone-shaped deposition model. This requirement may be violated when compliant bags are used as a construction material, as shown in Fig.6.

C. Motion Planner

In this section, we describe the motion planning for the arm and the rover base modules of the robot, developed under the ROS [21] framework. The motion planning modules utilize 3D coordinates of the pickup and deposition points of a bag. The entire system is supplemented with modules that translate the various co-ordinate transformations across the robot system.

1) Rover Base: The purpose of this module is to move the rover to a pose from which the destination point is reachable by the arm. A ROS controller was designed for the PID-tuned motor drivers, capable of maintaining constant velocities using the encoder readings. The robot navigation is comprised of a global and a local planner. We use the A* algorithm [22] to get the global path combined with an obstacle detection method from camera sensor readings. A global 2D map of the world is maintained for rover path planning purposes. Given a destination position $\mathbf{b} \in Q$, we find a position $\mathbf{b}^* \in Q$ using

Lemma (2) and Condition(6), from which the rover can reach $\mathbf{b}$. A global path is subsequently determined from the robot's current position to the *rover goal position* $\mathbf{b}^*$.

The motion execution is then carried out by the local motion planner that is based on the Direct Window Approach (DWA)[23] planner in ROS. The DWA controller aims to connect the global path planner to the robot by creating valid kinematic trajectories for the robot to get from a start to a goal location along its way in the global path, correcting trajectories based on external observation. It forward simulates local velocity trajectories and chooses the highest-scoring valid trajectory based on cost evaluations that depend on the distances form the global path, the goal and obstacles in the map.

2) Arm: The manipulation of the arm is controlled using a precise open loop system built into the arm control module, that uses an inverse kinematic model to transcribe the arm goal position to the internal motor angles. Combined with an obstacle detection mechanism from the camera sensor readings, a collision-free path is obtained to move the end-effector of the arm to the pickup or deposition position.

D. Experimental Setup

A global 2D occupancy grid map is maintained for motion planning using the overhead Kinect camera. The depth data is used to get the voxelized representation G of the construction area Q. The construction related decisions are restricted to local views, based on sensor readings from a region around the navigable region of the robot (Fig. 2(e)). As the navigable region changes, new information about the structure is utilized.

Parameters used in the implementation are as follows (in S.I. units): $\Delta = 0.015$, $\kappa = 0.314$, $\delta = 0.30$, $\kappa_D = 0.857$, $\epsilon = 0.048$, $\beta_{lx} = 0.18$, $\beta_{ux} = 0.34$, $\beta_{ax} = -\pi/2$, $\beta_{bx} = \pi/2$, $\beta_{lz} = -0.12$, $\beta_{uz} = 0.15$ and $\gamma = 0.06$.

V. Experiments

We present a series of experiments exhibiting the adaptability and robustness of our approach. The goal of the experiments is to build an access structure that enables the robot to get to an otherwise inaccessible target location, by placing bags in an unstructured and irregular terrain.

Table I depicts the total number of bags used, the number of missed pickups and depositions (due to gripper failure or system inaccuracy), and whether the robot was able to climb the ramp autonomously for each of the experiments. The total missed pickups and depositions made up less than 3% of the total bag placements across all runs. The success of the ramp despite these failures showcases the robustness of the algorithm. Figure 7 depicts some of the experimental runs from start (first column) to finish (third column), demonstrating various initial configurations and how the construction process plans a navigable path through the unstructured terrain. These experiments showcase the adaptability of the construction system. Additionally, a video containing a complete demonstration of the construction process is available.

In experiments 1, a large obstacle prevents the robot from getting close to the target location; the system reacts by building an access path to the large obstacle, and subsequently expanding the structure to the region that was previously inaccessible.

TABLE I: Experiments

Experiment	*Bags Placed*	*Missed Pick-ups/Depositions*	*Ramp Climbable*
1	103	2	Yes
2	106	2	Yes
3	33	0	Yes
4	42	2	Yes
5	65	1	Yes
6	99	1	No
7	49	1	Yes
8	98	3	Yes
9	62	6	Yes
10	170	5	Yes
TOTAL	827	23	–

In Experiment 2, there are no obstacles that prevent the robot from getting close to the target location; the first discontinuities are navigable. Hence, the construction process begins nearest the target location and extends backwards, towards the initial navigable region.

Experiment 3 depicts the algorithm's ability to choose regions where the terrain promotes lesser usage of building material. In experiment 4, the robot first levels the terrain and then gradually increases the slope of the access structure to reach the target position.

Modeling bag depositions as a cone may lead to conditions that violate the completeness proof of the algorithm. In many cases these issues are handled through the techniques mentioned in Sec. IV-B and the overall quality of the structure is not affected. However, if violations occur frequently, the robot may increase the upper bound of the target structure faster than it progresses toward the target. One such example is in experiment 6, when the robot failed to finish a navigable structure. In experiments 7-9, during the construction process, we manually disrupt the structure during the experiment, by adding, removing and/or displacing deposited bags. The construction system then reacts to such disturbances, further demonstrating its adaptability.

Regular repair of the structure is also a requirement; the robot displaces bags and rocks while traversing the structure. Among others, experiment 10 specifically highlights the robustness of the construction system and showcases its maintenance capabilities. The robot is tasked to build an access ramp and once done, climb over it multiple times and repair the structure in the process if required. By constantly assessing the structure surface, the robot decides whether or not it should perform any repair operations between successive climbs.

VI. Discussions and Future Work

By developing our system based on a well-defined model of construction, we are required to assure that the mapping between the abstract theoretical model and the physical execution maintains the necessary assumptions for correctness. One of the main challenges of this work was to assure these conformities, especially because of the approximations used in the bag deposition.

In our future work, we will investigate how we can develop a construction model that may be less restrictive, especially

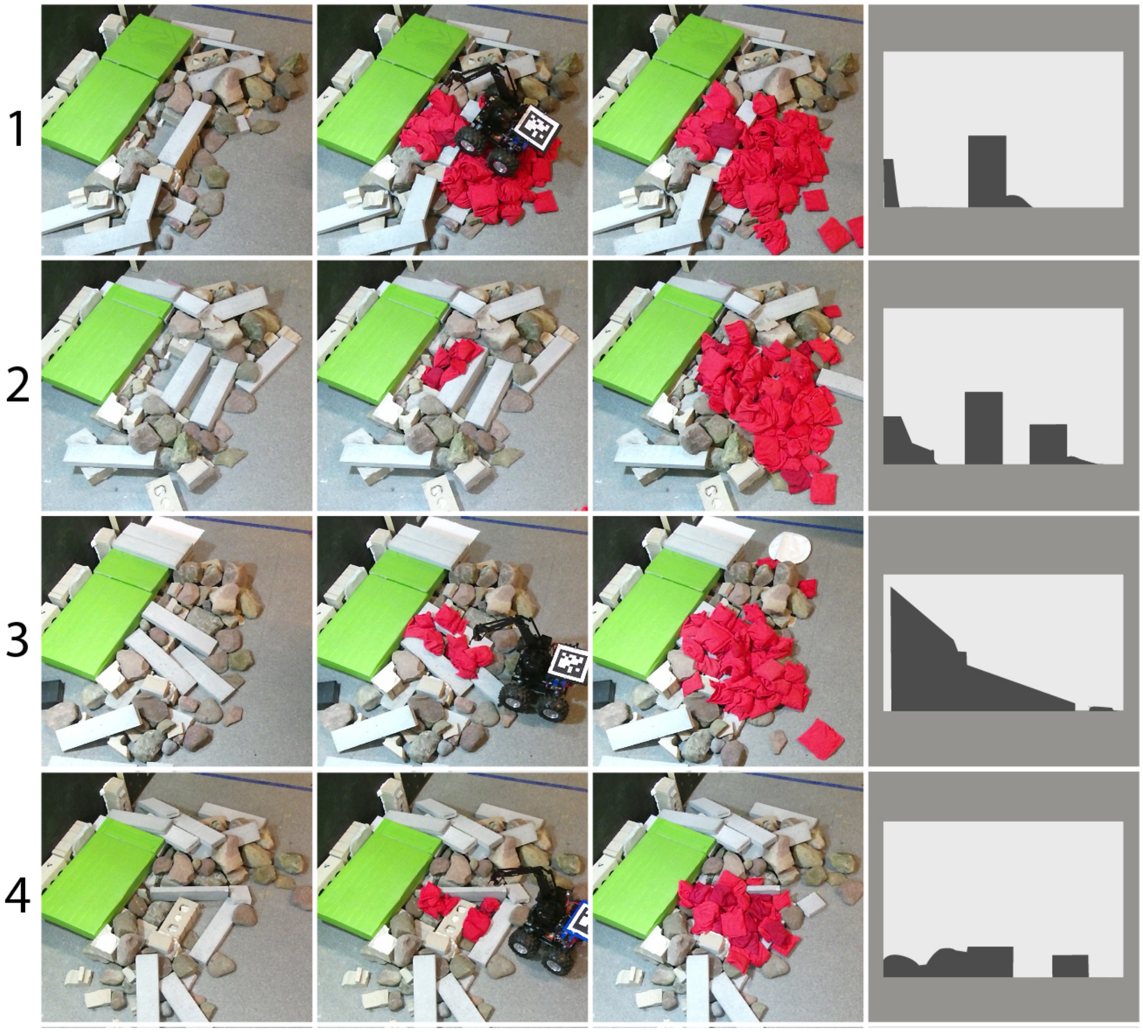

Fig. 7: Experiments showcasing initial setup, intermediate and final structures. Last column of images depict the 2d cross-section of the initial setup (in black) along the center of the green platform of each experiment, as a visual aid to the reader; they do not convey the true nature of irregularities in 3D space.

in relation to the deposition model, and that allows us to use different types of construction materials.

Although we currently use a camera for external observation, the perception module was designed to dynamically estimate the state of the construction with no specific camera position; as along as point clouds for bags, obstacles, and the build platform can be generated, the external system may be easily replaced by a SLAM-based system using appropriate on-board sensors. The intrinsic stigmergic nature of the control system makes our approach ideal for integration into a multi-robot system. At present, the system is capable of approaching a partial-built structure with no prior knowledge, and completing the task autonomously, given a goal position. This demonstrates that, with the addition of a collision-avoidance system, multiple robots could be integrated into the system, each independently planning how best to complete the structure, and in the process collaborating to construct more rapidly.

VII. Conclusion

In this work we present a property-driven, adaptable and robust autonomous system that modifies an unstructured environment to achieve and maintain navigability conditions. We choose an experimental setup and building material that closely resemble real world scenarios, and demonstrate practical effectiveness by using our low-cost robot system in constructing climbable structures with a 90% success rate. The final structure is the result of continuous evaluation of the terrain characteristics and subsequent building actions that are aware of the robot's kinematic and deposition constraints. This is a step toward practical autonomous construction systems in unstructured environments.

Acknowledgment

We are also grateful for the support of the SMART Community of Excellence at the University at Buffalo for supporting early versions of this work to the Science without Borders program (CAPES) for supporting Maira Saboia.

References

[1] M. H. Hansell, *Animal architecture*. Oxford University Press on Demand, 2005.

[2] J. S. Turner, *The Extended Organism: The Physiology of Animal-Built Structures*. Harvard University Press, 2002.

[3] L. Wilsson, *Observations and experiments on the ethology of the European beaver (Castor fiber L.) : a study in the development of phylogenetically adapted behaviour in a highly specialized mammal*. Uppsala: Almqvist och Wiksell, 1971.

[4] N. Napp and R. Nagpal, "Distributed amorphous ramp construction in unstructured environments," *Springer Tracts in Advanced Robotics*, vol. 104, pp. 105–119, 2014.

[5] N. Napp, O. R. Rappoli, J. M. Wu, and R. Nagpal, "Materials and mechanisms for amorphous robotic construction," *IEEE International Conference on Intelligent Robots and Systems*, pp. 4879–4885, 2012.

[6] G. Theraulaz and E. Bonabeau, "Coordination in distributed building," *Science*, vol. 269, no. 5224, p. 686, 1995.

[7] R. Beckers, O. E. Holland, and J.-L. Deneubourg, "From local actions to global tasks: Stigmergy and collective robotics," 1994.

[8] G. Theraulaz and E. Bonabeau, "A brief history of stigmergy," *Artificial life*, vol. 5, no. 2, pp. 97–116, 1999.

[9] Z. Mason, "Programming with stigmergy: Using swarms for construction," in *Proceedings of the Eighth International Conference on Artificial Life*, ser. ICAL 2003. Cambridge, MA, USA: MIT Press, 2003, pp. 371–374. [Online]. Available: http://dl.acm.org/citation.cfm?id=860295.860353

[10] J. Werfel, K. Petersen, and R. Nagpal, "Designing collective behavior in a termite-inspired robot construction team," *Science*, vol. 343, no. 6172, pp. 754–758, 2014.

[11] K. Petersen, R. Nagpal, and J. Werfel, "Termes: An autonomous robotic system for three-dimensional collective construction," *Proc. Robotics: Science & Systems VII*, 2011.

[12] M. Allwright, N. Bhalla, A. Antoun, H El-Faham, C. Pinciroli, and M. Dorigo, "{SRoCS}: Leveraging Stigmergy on a Multi-robot Construction Platform for Unknown Environments," *Swarm Intelligence – Proceedings of ANTS 2014 – Ninth International Conference*, vol. 8667, pp. 157–168, 2014.

[13] N. Melenbrink, P. Michalatos, P. Kassabian, and J. Werfel, "Using local force measurements to guide construction by distributed climbing robots," *In Proceedings of IEEE/RSJ international conference on intelligent robots and systems (IROS)*, 2017.

[14] N. S. Estévez and H. Lipson, "Dynamical blueprints: exploiting levels of system-environment interaction," in *Proceedings of the 9th annual conference on Genetic and evolutionary computation*. ACM, 2007, pp. 238–244.

[15] M. Allwright, N. Bhalla, C. Pinciroli, and M. Dorigo, "Towards autonomous construction using stigmergic blocks," Technical report TR/IRIDIA/2017-003. Brussels, Belgium: IRIDIA, Université Libre de Bruxelles, 2017. URL http://iridia. ulb. ac. be/IridiaTrSeries/index. php, Tech. Rep., 2017.

[16] T. Soleymani, V. Trianni, M. Bonani, F. Mondada, and M. Dorigo, "Bio-inspired construction with mobile robots and compliant pockets," *Robotics and Autonomous Systems*, vol. 74, pp. 340–350, 2015.

[17] R. Fujisawa, N. Nagaya, S. Okazaki, R. Sato, Y. Ikemoto, and S. Dobata, "Active modification of the environment by a robot with construction abilities," *ROBOMECH Journal*, vol. 2, no. 1, p. 9, Apr 2015. [Online]. Available: https://doi.org/10.1186/s40648-015-0030-2

[18] T. Tosun, J. Daudelin, G. Jing, H. Kress-Gazit, M. Campbell, and M. Yim, "Perception-informed autonomous environment augmentation with modular robots," *arXiv preprint arXiv:1710.01840*, 2017.

[19] uFactory, "uArm Swift & uArm Swift Pro Specifications," Tech. Rep., 2017. [Online]. Available: http://download.ufactory.cc/docs/en/uArm-Swift-Specifications-171012.pdf

[20] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 3400–3407.

[21] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[22] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[23] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.