Robotic Construction of Arbitrary Shapes with Amorphous Materials

Nils Napp and Radhika Nagpal
{nnapp@wyss, rad@eecs}.harvard.edu
Harvard University
Cambridge MA, USA

***Abstract*— We present a locally reactive algorithm to construct arbitrary shapes with *amorphous materials*. The goal is to provide methods for robust robotic construction in unstructured, cluttered terrain, where deliberative approaches with pre-fabricated construction elements are difficult to apply. Amorphous materials provide a simple way to interface with existing obstacles, as well as irregularly shaped previous depositions. The local reactive nature of these algorithms allows robots to recover from disturbances, operate in dynamic environments, and provides a way to work with scalable robot teams.**

I. Introduction

A. Motivation

Robots are well suited to perform tasks that put people in harm's way or need to be performed faster and more consistently that humans can. The long-term goal of this project is to enable robots to do the type of construction that is particularly useful in emergency situations, where severe time constraints and hazardous, poorly prepared construction environments are the norm, for example building support structures, levies, or access ramps. The focus of this paper is to enable one or more mobile robots to reliably build approximations to pre-specified structures with amorphous building materials. The target shape is assumed to be much larger than the robots, so they need to navigate and move on top of the partially completed structure. Amorphous materials allow pre-specified structures to be built on or around irregularly shaped obstacles—something that is difficult to achieve with deliberative approaches and pre-fabricated construction elements. In contrast to most of the related work on robotic construction and assembly, that focusses on building with discrete lattice-like elements, we use a continuous problem formulation and exploit the additional mathematical tools that come with it.

Our algorithmic approach relies on reactive robot behavior, which means robots use current local information to make decisions as opposed to following a fixed construction plan or maintaining a world model to execute such a plan. This approach provides feedback during the construction process, which allows us to work with amorphous materials that deform after deposition and operate in poorly characterized and dynamic environments. In previous work we demonstrated a locally reactive algorithm to adaptively build ramps over arbitrary unknown terrain. Here we extend the work on ramp building by approximating arbitrary target structures as a series of ramps. We present two flavors of the shape building algorithm: first, an algorithm where robots have global positioning, and second an algorithm where robots can only locally sense terrain. With global positioning, the underlying ramp building algorithm guarantees access to active sites throughout the construction process and its adaptive nature enables the final shape to be built over irregular terrain. Without global positioning, we design a compiler that takes an arbitrary goal structure, g, and generates *markers* in the environment such that when robots react to this marked up environment they build the desired shape, Fig. 1.

We envision these type of construction algorithms and robots to be useful in preparing hazardous cluttered sites with loose rubble for people or other robots, either by building a stabilizing layer over loose material or providing support structures and roughly level surfaces for more accurate types of construction. By exploring the tradeoff between locomotion and construction capabilities of a robot against the allowable approximation error and construction speed, this theoretical work can serve as a guide when designing such systems. The contribution of this paper is to provide a compiler for arbitrary target structures into an set of environmental markings that reproduce a target structure to within a pre-specified accuracy when robots respond to it with a known locally reactive behavior. In addition, the key technical contribution is a lower bound for terminating structures of the ramp building [11] in terms of $f-$continuity. This bound allows us to prove compiler works and provides more tools for reasoning about amorphous construction in general.

Section II sets up the mathematical notation provides needed results. Procedures for building arbitrary structures are described in Sec. III, where Sec. III-A summarizes previous work on building ramps, and Sec. III-C presents our main result. Simulations and a detailed error analysis of the resulting procedure are given in Sec. IV.

B. Related Work

Since construction is generally difficult, dirty, and often dangerous work there has been much interest in automating it. This brief literature survey mostly focuses on related algorithmic problems instead of mechanism design or low-level control problems.

Previous work on autonomous construction often focuses on the case where robots (or building blocks) have good estimates of their global position and all share the same consistent target shape [2][7][12][16][19] or compiled local

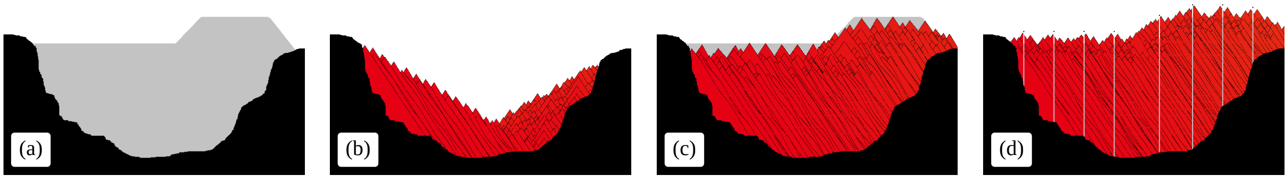

Fig. 1. (a) Irregular unnavigable terrain (black) and a dam-like target structure (gray). (b) Ramp building executed over terrain with no knowledge of target structure. Note that material is only added to move past steep features, i.e. to make terrain navigable. (c) Shape building with global knowledge. In addition to ramp building, robots make a deposition whenever they within the target shape and more than ϵ from the shape's boundary. (d) Shape built by executing ramp building in an marked up environment. Robots do not know about the final target shape, but respond to markers compiled according to Alg. 1 in Sec. III-C to approximate the target shape.

rules to create it [3][6]. Some approaches either abstract away motion constraints [16][21], or are developed for physical systems that do not have complicated motion constraints that restrict acceptable intermediate structures [2, pp130–151][7]. Approaches based on additive manufacturing, e.g. [4] also fall into this category of algorithms. This classification based on specification approach can be further broken down into how much information about the global state individual robots have during runtime. Some algorithms assume global knowledge of the state of the construction progress [2], [7] while others use only local observations of the construction state [12], [6], [16], [3]. The latter approaches often exploit the local nature to gain speedup though parallelism [12], [18], [21] and fault tolerance through reactive behaviors [6], [20]. When we assume that robots have global position estimates, the presented amorphous construction neatly falls into this line of research. Robots share the same coordinate system and goal, but have only local knowledge of the building progress and use local rules to satisfy global motion constraints.

The problem of specifying a target structure in a system with known growth dynamics is special case of programmed *self-assembly*, in which a system of interacting components is programmed/designed in such a way that the interaction dynamics result in a desired final shape. The extensive prior work in this area typically assumes that individual pieces do not have good estimates of their global position or any global knowledge, such as the number of interacting components, size of the assembly volume, assembly state, etc. Instead the global behavior is encoded by specifying purely local behaviors [5], [9]. This general approach has been applied to a wide variety of systems, ranging from DNA [13], micro-machined components where interactions are controlled by carefully engineering energy landscapes [1], to robots, where local interactions can take advantage of the considerable storage and computational power of micro-processors [14]. The idea of using feedback on the whole self-assembling system has also been explored by a number of groups. The interactions are still local, but external global knowledge of the assembly state is used to tune them [8][10][15]. The proposed markup procedure in Sec. III-C most closely resembles patterned self-assembly or seeded growth [17], where global knowledge is used to compile environmental markings that encode the target shape. Purely reactive (but carefully designed) robot behavior represents the growth dynamics.

II. The Amorphous Construction Model

The following notation is similar to our earlier work and included both to achieve a self-contained presentation and to concisely state the pervious results. This sections describes a state and deposition model for amorphous construction, a model for structures which robots can navigate, and a global projection operator on structures that is used for proving correctness for building strategies in the next section.

A. State and Deposition Model

We model the *construction area* Q as a convex, compact, and finite subset of $\mathbb{R}$ (or $\mathbb{R}^2$) and the domain of a bounded, non-negative height function $h : Q \to \mathbb{R}^+$ which describes a *structure*. Robots move on structures and modify them. While building specific structures, the *goal structure* is denoted by g. Given a goal structure and initial structure h, the *support* of g is the set of points where the goal structure differs from the initial environment, and is denoted by $S = \{x \in Q \mid h(x) < g(x)\}$, which we assume is a connected subset of Q. Given two structures h and g, g is said to *dominate* h when $h \leq g \equiv h(x) \leq g(x)\ \forall x \in Q$. In general, all relational symbols for functions should be interpreted pointwise. Function spaces are denoted by scripted letters. For example, let $\mathcal{Q}$ be the space of real-valued, bounded functions on Q, and $\mathcal{Q}^+ \subset \mathcal{Q}$ the subset of non-negative ones. Function application to points and operator applications to functions are denoted by $(\cdot)$ and $[\cdot]$ respectively.

Robots can deposit amorphous construction material and control its volume and position. The free (top) surface of each deposition is modeled by a parameterized *shape function* $d \in \mathcal{Q}$ while the bottom conforms to the structure. As a simple, yet sufficiently general, family of shape functions we use cones. Given an apex-position pair $(\phi, \sigma) \in Q \times \mathbb{R}^+$ and steepness $k_d \in \mathbb{R}^+$ let

$$d_{(\phi,\sigma)}(x) = \sigma - k_d|\phi - x|. \quad (1)$$

The deposition operator $D : \mathcal{Q} \times \mathcal{Q}^+ \to \mathcal{Q}^+$ models interactions of depositions with the environment, here simply covering it as construction with materials like mud, expanding foam, or sand would. Given a structure $h \in \mathcal{Q}^+$ with

$h(\phi) < \sigma$, the new structure after deposition $d_{(\phi,\sigma)}$ is given by

$$D[d_{(\phi,\sigma)}, h](x) = \max\{(d(x), h(x)\}. \quad (2)$$

Given an initial structure $h_0 \in \mathcal{Q}^+$ a structure is built by a sequence of depositions characterized by their shape parameters $(\phi_1, \sigma_1), (\phi_2, \sigma_2), (\phi_3, \sigma_3), ...$ The height function h_n after n depositions is defined recursively by

$$h_n(x) = D[d_{(\phi_n,\sigma_n)}, h_{n-1}](x). \quad (3)$$

After the n-th deposition, the local reactive rules of each robot direct it to move on h_n and possibly make a deposition resulting in a new structure h_{n+1}. For example, in the case where robots have global knowledge, they deposit whenever they are inside the goal shape.

B. Navigable Structures and f-continuity

Building ramps requires a concise description of navigable structures. This section recasts the definition from [11][Eqn.(5)], where the primary aim was tractability, as type of continuity condition ($f-$continuity).

We use three parameters to describe robot specific motion constraints: $K \in \mathbb{R}^+$, to model the maximum steepness robots can drive up or down, $\epsilon \in \mathbb{R}^+$, to model the largest discontinuity robots can freely move past, and $d \in \mathbb{R}^+$, to limit the amount of discontinuity in a small area, such as the robot length. A structure is called *navigable* if and only if it is locally (parameter r) close (parameter ϵ) to $K-$Lipschitz, i.e. $\forall x, y \in Q$ and $|x - y| \le r$:

$$|h(x) - h(y)| \le \epsilon + K|x - y|. \quad (4)$$

We show here how to recast this definition as a continuity constraint where a single function, Eqn. (8 which depends the three navigability parameters, is used to weigh the distance between two points.

A function $h \in \mathcal{Q}$ is called f-continuous iff

$$\forall x, y \in Q \qquad |h(x) - h(y)| \le f(|x - y|), \quad (5)$$

where $f : \mathbb{R}^+ \to \mathbb{R}^+$ is a monotone function that is zero at zero and sub-additive i.e.

$$\begin{aligned} f(0) &= 0 \\ x \le y &\Rightarrow f(x) \le f(y) \quad \forall x, y, \in \mathbb{R}^+, \\ f(x + y) &\le f(x) + f(y) \quad \forall x, y \in \mathbb{R}^+. \end{aligned} \quad (6)$$

For example, when $k(x) = Kx$ then a function is $f-$continuous with function k (written as k-continuous) iff it is $K-$Lipschitz, see Fig. 2(a) for example functions.

To reason about global guarantees of our local algorithms, we define the projection operator P_f in Eqn.(7), which maps any structure to the *closest* f-continuous function that can be built by only adding material. At point $x \in Q$, P_f takes the maximum value of any needed additions so all other points fulfill condition (5),

$$P_f[h](x) = \max_{y \in Q}\{h(y) - f(|x - y|)\}. \quad (7)$$

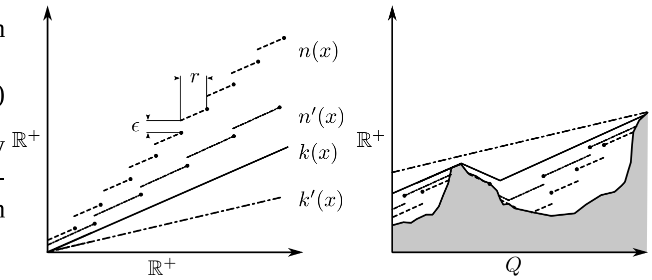

Fig. 2. (a) Example functions for $f-$continuity. The functions k and k' describe Lipschitz continuity with different parameters K. The functions n and n' are navigability functions with different parameters r and ϵ. (b) Different f-continuous projections of the same a sample function (filled grey) using using k, k', n and n' for specifying continuity.

Theorem 1: The operator $P_f : \mathcal{Q} \mapsto \mathcal{Q}$ in Eqn. (7) has the following properties:

1) $P_f[h](x)$ is the smallest $f-$continuous function that dominates h.
2) $f \le f' \Rightarrow P_f[h] \ge P_{f'}[h]$ (i.e. steeper $f \Rightarrow$ add less)

See Fig. 2(b) for examples and Sec. VI for proof.

Theorem 2: The function

$$n(x) = \left\lceil \frac{x}{r} \right\rceil \epsilon + Kx \quad (8)$$

fulfills the conditions in Eqn. (6) and $n-$continuous functions are exactly the set of navigable ones.

Proof: (n-continuous $\Rightarrow$ navigable) For a given n continuous function h, restricting the definition of n-continuity in Eqn. (5) to point pairs x, x' s.t. $|x - x'| \le r$ results in the navigability condition $|h(x) - h(x')| \le \epsilon + K|x - x'|$ (navigable $\Rightarrow$ n-continuous) Given an arbitrary point pair $x, x' \in Q$, the line is also in Q by assumption. Along this line let x_0, x_1 ... x_N be $N + 1$ points spaced long the line with $x_0 = x$ and $x_N = x'$, where the first N points are spaced r apart and the last pair possibly less, $|x_N - x_{N-1}| \le r$. For each pair x_{i-1}, x_i by navigability $|h(x_i) - h(x_{i-1})| \le \epsilon + |x_i - x_{i-1}|$. There are $N = \left\lceil \frac{|x-x'|}{r} \right\rceil$ such point pairs and summing the incremental differences implies n-continuity

$$\begin{aligned} & K|x - x'| + \left\lceil \frac{|x - x'|}{r} \right\rceil \epsilon \\ = \quad & K \sum_{i=1}^{N} |x_i - x_{i-1}| + N * \epsilon \ge \sum_{i=1}^{N} |h(x_i) - h(x_{i-1})| \\ \ge \quad & |h(x_0) - h(x_N)| = |h(x) - h(x')| \end{aligned} \quad (9)$$

■

Navigability is defined for structures and checked for point pairs. Using the equivalence between n-continuity and navigability allows a direct definition of navigable points in a structure. A point $x \in Q$ in structure h is called navigable iff $P_n[h](x) = h(x)$.

III. Building Arbitrary Structures

This section describes increasingly complex examples of building structures: (1) a summary of pervious results on building ramps that make arbitrary terrain naviagable; (2) a strategy for building a goal structure, g, on arbitrary, potentially unnavigable terrain, when robots can estimate their global position in a consistent reference frame, i.e. have GPS; and (3) a compilation procedure for producing markers that allow robots to build structures to within a pre-specified error ϵ_T using only local knowledge. As presented, all algorithms in this section require $Q \subset \mathbb{R}$ to get an ordering of points in Q. One simple way of applying these algorithms to $Q \subset \mathbb{R}^2$ is to fix a path and directly use the results. Alternatively, these results could be extended to higher dimensions by ordering points via a different scheme, for example, the estimated effort it takes to make them navigable or other procedures to create efficient implementations.

A. Building Ramps Adaptively

Our previous work on adaptive ramp building [11] guarantees the construction of a final structure structure, h_*, over initial terrain, h_0, that is navigable everywhere between a starting point, x_0, and a goal position x_*. One or more robots can build ramps using only local knowledge of the current terrain and the heading direction toward x_*.

The ramp building algorithms works by maintaining a navigable area around x_0, called *accessible region*, and extending it toward x_* until the goal is reached, Fig. 3(a). Specifically, a robot repeats the following sequence of operations:

1) Move toward goal until reached or finding non-navigable feature, i.e. a point pair $|x - y| < r$ with $|h(x) - h(y)| > K|x - y| + \epsilon$.
2) Deposit on lower point of non-navigable point pair (minimum deposition height ϵ and maximum height given by robot geometry).
3) Backup by $2r$.

As depositions made in response to non-navigable features might themselves be non-navigable features, backing up guarantees that robots maintain the accessible region that extends from x_0. This region might temporarily shrink, e.g. when a robot is moving uphill, encounters an obstacle, and makes a deposition that is too steep to move past. By backing up and repeating the procedure new depositions are checked for navagability.

How the navigability parameters K, ϵ, and r relate to a robot's geometry is illustrated in Fig. 3(b). We assume that they can be chosen conservatively for a given robot. In fact, since they influence the built structure one might choose these values to fulfill both robot motion constraints and produce desirable results. For example, one could create a smoother structure by choosing an artificially low value of ϵ. The following considerations place limits on these parameters:

Deposition resolution	$< \epsilon <$	Motion constraints.
0	$\leq K <$	Deposition paramteres (e.g. K_d) or motion constraints.
Robot length	$< r <$	Sensing range.

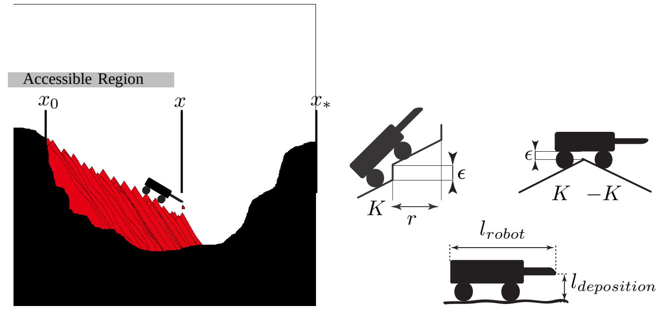

Fig. 3. (a) Schematic of adaptive ramp building. Cone like depositions are shown in red. The accessible region reaches form the smallest point in the domain to the current position x. When ramp building is complete (Fig. 1b) the whole domain is navigable. (b) Schematic of robot specific geometric limits on navigability parameters. The lengths $l_{deposition}$ and l_{robot} place limits on the maximum deposition height and r.

Within these ranges the parameters can be chosen freely and different combinations will result in different final structures: all navigable according to Eqn. (4).

Theorem 3: Setting $x_0 = \min(Q) + 2r$ and $x_* = \max(Q)$ and running the ramp building algorithm on an initial structure h_0 where points up to the initial position, $x \leq x_0$, are navigable results in a final structure h_* with upper and lower bounds given by

$$P_n[h_0] \leq h_* \leq P_k[h_0]. \tag{10}$$

Proof: The choice of x_0 and x_* imply that h_* is navigable everywhere. The upper bound P_k is proved in [11, Sec. 4.1]. Theorem 2 implies that $P_n[h_0]$ is navigable, and Thm. 1.1 implies it is the smallest navigable function that dominates h_0. Since $h_0 \leq h_*$, $P_n[h_0]$ is a lower bound for h_*. ■

Note that [11] assumes a continuous h_0 in order to ensure progress—a restriction we would like to relax to allow discontinuities at markers in the next section.

Lemma 4: Adaptive ramp building works over discontinuous structures with a countable number discontinuities of the from $h_0(x) + \sum_i \alpha_i \delta(x - x_i)$, where $\delta(0) = 1$, $\alpha_i > 0$, and $h_0 \in \mathcal{Q}^+$ is continuous.

Proof: Since they have measure zero and finite height, this type of discontinuity does not change any of the volume computations in the progress proofs. The restriction $\alpha_i > 0$ ensures that the discontinuous points x_i are only deposition locations (lower point in non-navigable pair) if its neighborhood (responsible for deposition volume) ensures progress [11, Sec. 3.1]. ■

B. Building With Global Positioning

With global positioning, the problem of building an arbitrary goal structure $g(x)$ is easy to solve. In addition to ramp building robots execute the following behavior: they make a deposition (of a maximum height $g(x) + \frac{\epsilon}{2}$) whenever they are in the support, S, of g and

$$h_n(x) < g(x) - \frac{\epsilon}{2}. \tag{11}$$

For these depositions, they follow the same sizing and backing up strategy as in adaptive ramp building. Following

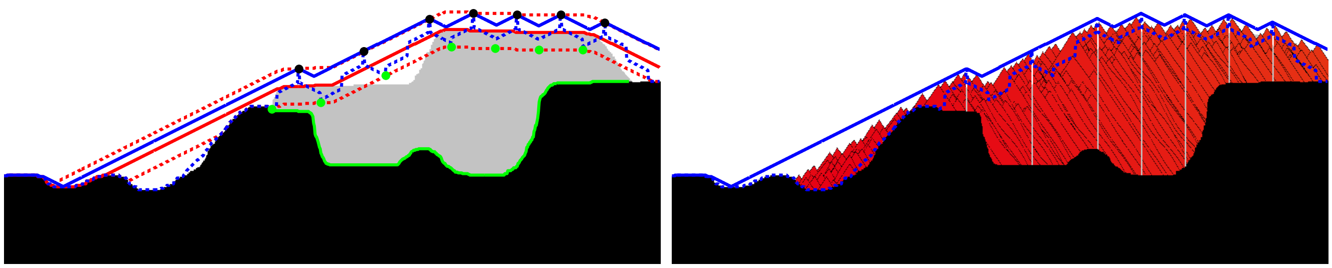

Fig. 4. (a) Diagram of compilation procedure and error bounds. The original environment is shown in black, the structure to be built build, g, in grey, and the markers as black dots. The K-Lipschitz approximation of the shape is shown in solid red and the $\pm\epsilon_T$ bounds of acceptable final structures, u and l, as dashed red lines. The upper and lower bounds P_k and P_n induced by the marker are shows as in blue lines. The compilation procedure results in markers that have a height equal to the upper $+\epsilon_T$ bound and are spaced so the lower bounds induced by each marker intersect at points (shown in green) on the $-\epsilon_T$ bound of the structure. Markers are only placed on the support, shown in green, of the grey structure, i.e. points in the original environment on which the structure should be built. (b) Final structure built in response to markers.

this strategy essentially treats the interior of g as an obstacle and the ramp building algorithm adaptively builds a ramp over it.

Robots might need to additional material to ensure that the structures they are moving on are navigable. Where and how much they add is determined by $u(x)$ and $l(x)$, upper and lower approximations of g that take navigability into consideration. They are defined in terms of, potentially non-navigable, auxiliary functions g_u and g_l:

$$g_u(x) = \begin{cases} g(x) + \epsilon_T & , x \in S \\ h_0(x) & , \text{otherwise} \end{cases}$$

$$u(x) = P_k[g_u](x) \tag{12}$$

$$g_l(x) = \begin{cases} \max(g(x) - \epsilon_T, h_0) & , x \in S \\ h_0(x) & , \text{otherwise} \end{cases}$$

$$l(x) = P_k[g_l](x). \tag{13}$$

In terms of the navigable upper and lower bounds this strategy results in the following structure.

Theorem 5: When robots execute the ramp building algorithm and additionally deposit on points $x \in S$ when condition Eqn. (11) with the maximum deposition height constraint $g(x) + \frac{\epsilon}{2}$ as outlined above, the resulting structure $h_* \leq u \ \forall x \in Q$ and $l \leq h_*$ for all navigable points in S with $\epsilon_T = \frac{\epsilon}{2}$.

Proof: The deposition height is limited by g_u on navigable points of g. Since any additional depositions to ensure navigability the are made by the ramp building algorithm, the upper bound from Thm. 3 is $P_k[g_u] = u$. Robots will deposit on points in S with $h_n \leq g(x) - \frac{\epsilon}{2} \leq l$ and can make these depositions without violating the upper bound since the maximum deposition height is as least ϵ. Where g is navigable, i.e. points that are not covered by a ramp, robots will keep adding to the target structure until the $l \leq h_n$. ■

C. Building Without Global Positioning

Without global positioning individual robots do not know where they are with respect to the target shape. Local sensing allows them to assess the navigability of their immediate surroundings. Yet, while this restriction limits the ability of individual robots to make decisions about where to build, dropping the need of sharing in a consistent global reference frame also makes this approach interesting to study in theory and robust to position and progress uncertainty in practice.

Specifying Structures: The particular strategy we pursue here is to design a set of N discrete markings m of the form

$$m(x) = \sum_{i=1}^{N} \alpha_i \delta(x - x_i) \tag{14}$$

so that ramp building on $h_0 + m$ results in the final target shape g to within a pre-specified error ϵ_T. We tackle this problem by designing a compilation procedure that takes an arbitrary initial environment $h_0(x)$, goal shape $g(x)$, and error ϵ_T to produce initial markings m on h_0 s.t. the system dynamics of ramp building have a steady state $h_*(x)$ with $|h_*(x) - g(x)| \leq \epsilon_T$ on S, subject to g being navigable.

Compiling Markers

Each marker above a certain height makes a structure non-navigable, so that robots will build a ramp in response. Given the upper bound u and lower bound l for acceptable final structures the compilation procedure in Alg. 1 generates m, of the form Eqn. (14), such that ramp building on $h_0 + m$, result is a navigable structure that is bound between l and u for every point in the support S, and bounded above by u outside of the support.

Marker placements are computed iteratively by choosing positions such that when their height is $h_0(x) + m(x) = u(x)$ then the lower bounds of the resulting ramps intersect on l, Fig. 1(a). The points, where the lower bounds intersect, are called crossing points $c_i \in S$. The compilation algorithm

Alg. 1 starts with a crossing point $c_0 = \min(S)$ and alternates between computing the next larger marker location x_1, which in turn determines the next crossing point c_1, where $c_0 < x_1 < c_1 \ldots$.

Algorithm 1 Compiling Markers.

1: $c_0 = \min(S)$
2: $N = 0$
3: **while** $c_i < \max(S)$ **do**
4: $\quad x_i = \max\{x \in S \mid |u(x) - n(|x - c_i|) > l(c_i)\}$
5: $\quad c_{i+1} = \max\{c \in S \mid |u(x_i) - n(|x_i - c|) > l(c)\}$
6: $\quad$ increment N
7: **end while**
8: $m(x) = \sum_{i=1}^{N} u(x_i) - h_0(x_i)\delta(x - x_i)$.

Theorem 6: Given an initial structure h_0 and goal structure $g \geq h_0$ and target error $\epsilon_T > \epsilon$, the compilation procedure in Alg. 1 terminates and generates a final marking m such that executing adaptive ramp building on $h_0 + m$ results in a final structure h_* with the property that $|h_* - P_k[g]| \leq \epsilon_T$ on the support of g and $h_* \leq u$ outside the support.

Proof: This proof proceeds in three steps. First the global upper bound, then the lower bound on S, and finally termination.

The upper bound u is k-continuous. By construction $h_0 + m \leq u$ and the upper bound for the final structure h_* of adaptive ramp building on $h_0 + m$ is $P_k[h_0 + m]$. Since $P_k[h_0 + m]$ is the smallest k-continuous function that dominates $h_0 + m$, u must be at least as large and $h_* < u$ globally. Since $u - P_k[g] \leq \epsilon_T$ the other limit to ensure that $|h_* - P_k[g]| \leq \epsilon_T$ on S is a lower bound for h_* on S.

We prover the lower bound by induction and show that the marker between two crossing points c_i and c_{i+1} induces the lower bound on h_* between the two points. Since adding markers can only increase the size of h_* adding more markers can only make the bound tighter. Assume that $|h_*(x) - P_k[g](x)| \leq \epsilon_T$ for all $x \leq c_i$. By Line 4 in Alg. 1, the smallest n-continuous function built in response to the marker at x_i is larger than l at c_i and the points between c_i and x_i. By Line 5 the next c_{i+1} is close enough that at the smallest $n-$continouse function built in response to the marker at x_i at lest as larger as l at c_{i+1} and the points in between x_i and c_{i+1}.

Since the magnitude of the derivative of l and u are both bounded by K, the minimum separation between c_i and x_i is at least $\frac{\epsilon_T - \frac{\epsilon}{2}}{K}$. Thus each pass through the loop ensures progress. Termination requires that ϵ_T is at least $\frac{\epsilon}{2}$, otherwise $c_i = x_i = c_{i+1}$ and iterations through the loop (Line 3–6) do not result in progress. ■

Once the compiler has used global information to compute maker height and locations, adaptive ramp building can build the structure by locally responding to them, see Fig. 4(b).

IV. Simulation Experiments and Parameter Sensitivity Analysis

This section focuses on the quantifying different types of errors when building arbitrary structures. Given a target shape g this approach to designing markers has three distinct sources of error in the final structure

1) If the goal shape g is not K-Lipschitz it is approximated by $P_k[g]$.
2) The difference between $|P_k[h_0 + m] - P_k[g]|$ which result from approximating the shape by a finite number of markers.
3) The uncertainty in the exact shape of h_* of where the ramp building algorithm will stop, i.e. the bounds $P_n[h_0 + m] \leq h_* \leq P_K[h_0 + m]$.

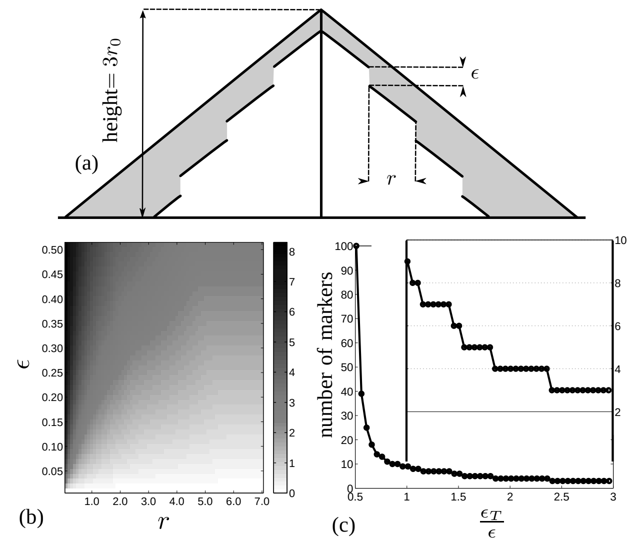

Fig. 5. Adaptive Ramp Building. (a) Schematic of test geometry for quantifying error (grey area). All quantities are normalized to some r_0, so this geometry represents building a ramp over a marker that is three times a high as the robot is long.(b) Error volume of sample structure as a function of varying r, and ϵ and a constant $K = 0.5$. Since all variables are normalized to r_0, a value of $r = 2$ represents increasing the characteristic length of a robot by 2 ,either by physically increasing the robot size or adjusting the sensing range. The error is given in r_0^2. (c) Number of markers produced by Alg. 1 on the target structure in Fig. 4(a) as a function of changing ϵ_T and keeping all other parameters the same. The inset shows that as ϵ_T grows only a small number of markers are needed, and the same number of markers can produce several different levels of accuracy. For example, $\epsilon_T = 2.3\epsilon$ and $\epsilon_T = 1.9\epsilon$ both require 4 markers.

The first type of error is due to a robot's motion constraints. If robots cannot move over steep features, then the the final shape cannot have any. This error can be reduced by using robots with navigability constraints that have a large value of K. This type of error is bounded by $|g - P_k[g]|$.

The second type of error is related to marker density. Since it is controlled by the compilation parameter ϵ_T, the (L_∞) error itself is easy to quantify, $\pm\epsilon_T$ from $P_k[g]$. Figure 5(c) shows the number of markers needed to compile the target structure in Fig 4(a) as function of ϵ_T where all other parameters are constant. Since the navigability function n has a discontinuity, choosing the smallest ϵ_T for a given number of markers makes sense. For some values the error can be decreased essentially for free (no additional markers). This type of error (L_1) is bounded by $\epsilon_T|S|$, i.e. target epsilon times the size of the support. From the proof of Thm. 6, an upper bound for the number of markers is given by the minimum speparation between markers and the diameter of S, i.e. for a given target error and parameter combination the number of needed markers is bound by $K * \frac{|S|}{\epsilon_T - \frac{\epsilon}{2}}$. And the procedure does indeed produce an excessive amount of markes when ϵ_T approaches the lower theoretical limit $\frac{\epsilon}{2}$, see Fig. 5(c).

The third type of error depends on the navigability parameters K, r, and ϵ. We use the reference geometry in Fig. 5(a) to evaluate the tradeoff of different parameter combinations. The reproduction fidelity should (and does) depend on how much information each robot has, i.e. how local the algorithm is, and how much uncertainty each amorphous deposition has. On one hand, if the construction material is very sloppy and does not allow for accurate depositions (large ϵ) the construction material limits fidelity, and more information does not help. On the other hand, if the view of the world is too local, i.e. robots cannot see much of their environment (small r), uncertainties propagate and limit fidelity, and more accurate materials do not help, Fig. 5(b).

The different error sources can be controlled in ways and this section gives guidance in how to trade off different design parameters. For example, whether it is better to design a robot that is better at climbing or to focus on a better deposition mechanism depends both on the structure and the acceptable reproduction error ϵ_T.

V. Conclusion

We present algorithms for robotic construction with amorphous building materials that allows one or more robots to build approximations to arbitrary shapes and provide an error analysis for the various sources of approximation error. Amorphous building materials allow robots to build structures over unknown irregular terrain, and these methods are designed to enable robotic construction in disaster areas where such environments are the norm. The presented approach utilizes previous work on adaptive ramp building as a base behavior to provide access to building sites during the construction process. Complicated structures can be approximated by a series of ramps. By using single markers to encode ramps, we provide a compilation procedure that can efficiently encode approximations to an arbitrary target structure with a relatively small number of markers. This approach allows robots without global positioning to build structures by locally reacting to markers.

In addition to creating a physical implementation, we plan to extend this work by looking for more physically realistic markers. One promising option is to mark up environment by choosing the location according this algorithm and encoding the desired height in some other way, for example, the marker color or a digital tag embedded the building material that robots can read and modify. The next theoretical step that would make both the presented work and the underlying ramp building algorithm more feasibly to implement on a robot is would be to get a better characterization of allowable shape functions for depositions. Rather then designing the amorphous deposition mechanism around a mathematical abstraction that works, it would be much better to look for good deposition mechanisms and then adjust the theory accordingly.

Acknowledgemetns

We are grateful to the Wyss Institute for Biologically Inspired Engineering which funded this work.

References

[1] Mila Boncheva and George M. Whitesides. Making things by self-assembly. *MRS Bulletin*, 30:736–742, 10 2005.

[2] Gramazio & Kohler et Raffaello d'Andrea. *Flight Assembled Architecture*. HYX Architecture, Contemporary Art, Expos Review, Digital Cultures, Philosophy, 2013.

[3] Alexander Grushin and James A. Reggia. Automated design of distributed control rules for the self-assembly of prespecified artificial structures. *Robotics and Autonomous Systems*, 56:334 – 359, 2008/04/30/ 2008.

[4] B. Khoshnevis. Automated construction by contour crafting related robotics and information technologies. *Journal of Automation in Construction Special Issue: The best of ISARC 2002*, 13:5–19, 2004.

[5] E. Klavins, R. Ghrist, and D. Lipsky. A grammatical approach to self-organizing robotic systems. *Automatic Control, IEEE Transactions on*, 51(6):949–962, 2006.

[6] K. Kotay and D. Rus. Generic distributed assembly and repair algorithms for self-reconfiguring robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, Sendai, Japan, 2004.

[7] Quentin Lindsey, Daniel Mellinger, and Vijay Kumar. Construction of cubic structures with quadrotor teams. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2011.

[8] Gregory Mermoud, Massimo Mastrangeli, U. Upadhyay, and A. Martinoli. Real-time automated modeling and control of self-assembling systems. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 4266–4273, 2012.

[9] R. Nagpal, F. Zambonelli, E.G. Sirer, H. Chaouchi, and M. Smirnov. Interdisciplinary research: roles for self-organization. *Intelligent Systems, IEEE*, 21(2):50–58, 2006.

[10] Nils Napp, Samuel Burden, and Eric Klavins. Setpoint regulation for stochastically interacting robots. *Autonomous Robots*, 30(1):57–71, 2011.

[11] Nils Napp and Radhika Nagpal. Distributed amorphous ramp construction in unstructured environments. In *Distributed Autonomous Robotic Systems (DARS12)*, 2012.

[12] Kirstin Petersen, Radhika Nagpal, and Justin Werfel. Termes: An autonomous robotic system for three-dimensional collective construction. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2011.

[13] Paul W. K Rothemund, Nick Papadakis, and Erik Winfree. Algorithmic self-assembly of dna sierpinski triangles. *PLoS Biol*, 2(12):e424, 12 2004.

[14] M. Rubenstein and W. Shen. Automatic scalable size selection for the shape of a distributed robotic collective. In *2010 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, Taipei Taiwan*, 2010.

[15] Michael T Tolley and Hod Lipson. On-line assembly planning for stochastically reconfigurable systems. *The International Journal of Robotics Research*, 2011.

[16] Justin Werfel and Radhika Nagpal. Three-dimensional construction with mobile robots and modular blocks. *Int. J. Rob. Res.*, 27:463–479, March 2008.

[17] George M. Whitesides and Bartosz Grzybowski. Self-assembly at all scales. *Science*, 295(5564):2418–2421, 2002.

[18] James Worcester, Joshua Rogoff, and M. Ani Hsieh. Constrained task partitioning for distributed assembly. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 4790–4796, 2011.

[19] S. Yun, D. A. Hjelle, H. Lipson, and D. Rus. Planning the reconfiguration of grounded truss structures with truss climbing robots that carry truss elements. In *Proc. of IEEE/RSJ IEEE International Conference on Robotics and Automation*, Kobe, Japan, May 2009.

[20] Seung-kook Yun and D. Rus. Adaptation to robot failures and shape change in decentralized construction. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2451–2458, 2010.

[21] Seung-kook Yun, Mac Schwager, and Daniela Rus. Coordinating construction of truss structures using distributed equal-mass partitioning. In Cdric Pradalier, Roland Siegwart, and Gerhard Hirzinger, editors, *Robotics Research*, volume 70 of *Springer Tracts in Advanced Robotics*, pages 607–623. Springer Berlin Heidelberg, 2011.

VI. Proofs

Proof: 1 (Smallest dominating $f-$continuous function): The proof proceeds in three steps, first we show that $P_f[h]$ is f-continuous, second that it is dominating, and third that it is the smallest such function.

To prove f-continuity, assume to the contrary that $P_f[h](x)$ is not $f-$continuous. Then $\exists\ x, x' \in Q$ s.t. $|P_f[h](x) - P_f[h](x')| > f(|x - x'|)$. Without loss of generality, assume that $P_f[h](x') < P_f[h](x)$. Dropping the absolute value and rearranging terms gives:

$$P_f[h](x) - f(|x - x'|) \quad > \quad P_f[h](x')$$

or

$$\begin{aligned} & \max_{y \in Q}\{h(y) - f(|x - y|\} - f(|x - x'|) \\ > \quad & \max_{y' \in Q}\{h(y') - f(|x' - y'|)\}. \end{aligned} \tag{15}$$

Choosing, the possibly sub-optimal value, $y' = y$ for the smaller term yields

$$h(y) - f(|x - y|) - f(|x - x'|) > h(y) - f(|x' - y|).$$

Sub-additivity and monotonicity of f lead to the following contradiction

$$\begin{aligned} & f(|x' - y|) > f(|x - y|) + f(|x - x'|) \\ \geq \quad & f(|x - y| + |x - x'|) \geq f(|x' - y|). \end{aligned}$$

Thus, $P_f[h](x)$ is $f-$continuous.

To show that $P_f[h]$ dominates h, note that in the definition in Eqn. 7 the maximization is over all elements $y \in Q$, since $x \in Q$ and $f(0) = 0$, the value of the new function $P_f[h](x)$ at x is at least as large as $h(x)$.

Finally, to show $P_f[h]$ is the smallest dominating $f-$continuous function, let $g, h \in \mathcal{Q}^+$ be two functions where $g \geq h$ and g is $f-$continuous. Assume to the contrary that $\exists x \in Q$ s.t. $P_f[h](x) > g(x)$. By the definition of $P_f[h]$ $\exists x' \in Q$ s.t.

$$h(x') - f(|x' - x|) > g(x).$$

Since $g \geq h$,

$$g(x) < h(x') - f(|x' - x|) \leq g(x') - f(|x' - x|)$$

leads to a contradiction about the $f-$continuity of g

$$f(|x' - x|) < g(x') - g(x) \leq |g(x') - g(x)|.$$

Thus $P_f[h]$ is the smallest $f-$continuous function that dominates h.

■

Proof: 2 (Preserved order of f): Given two functions $f, f' \in \mathbb{R}^+ \to \mathbb{R}^+$ that fulfill condition (6) where $f \leq f'$ and an arbitrary structure $h \in \mathcal{Q}^+$, then $P_{f'}[h] \leq P_f[h]$. The proof follows directly from the definition Eqn. (7), since every point in $P_{f'}[h]$ is smaller (subtracts more) than $P_f[h]$.

■