Assembly by Intelligent Scaffolding

Erik Komendera, Dustin Reishus, and Nikolaus Correll

erik.komendera@colorado.edu, reishus@colorado.edu, nikolaus.correll@colorado.edu

Department of Computer Science
University of Colorado at Boulder

Technical Report CU-CS 1080-11

April 2011

Assembly by Intelligent Scaffolding

Erik Komendera Dustin Reishus Nikolaus Correll

***Abstract*—We propose a novel class of algorithms for autonomously assembling structures from inert building blocks guided by intelligent scaffolding components. Intelligent scaffold units are equipped with sensing, actuation, computation and communication abilities and facilitate the attachment of inert building blocks to the structure. After attaching an inert building block, the scaffold structure reconfigures to attach the next block until the structure is completed. The proposed algorithms are scale-free and independent of the implementation of the locomotion of building blocks and intelligent scaffolding blocks. For example, movement of building and scaffold blocks can be achieved using manipulating robots or self-assembly in a well-stirred liquid. In a robotic assembly context, the intelligent scaffolds take the role of markers on the structure and allow for reducing the perception and coordination requirements on the robotic team. In this paper, we describe algorithms for converting any desired structure that can be represented as 3D lattice into a finite state machine that is executed by intelligent scaffolding blocks; we prove that all finite structures can be assembled using intelligent scaffolds; and we provide examples of simulations that assemble a square, a fractal structure, and a model of a space station, each using only three intelligent scaffold components.**

I. Introduction

We wish to design distributed algorithms that guide the assembly of structures from *passive* building blocks. We propose to coordinate the assembly of structures using "intelligent scaffolding". Intelligent scaffolds (IS) are computational building blocks that have some sensing, actuation, computation and communication abilities. IS blocks store all the information that are required to assemble a structure and serve as markers on the structure as assembly progresses. IS blocks remain in a contiguous group and always remain attached to the structure. By continuously re-assembling along the surface of the structure being built, arbitrary large-scale structures can be built from inert components with only a few IS blocks (Figure 1).

Autonomous assembly has the potential to revolutionize manufacturing, repair, and construction from the nano- to the macro-scale [29]. Miniaturizing this process is desirable as it allows for parallel assembly of a large number of structures and the assembly of structures in locations that are not accessible by tools that are larger than the actual parts to be assembled. Assembly using intelligent building blocks that coordinate the process has been experimentally demonstrated at the nano-scale using DNA tiles [1, 21, 18], at the meso-scale using intelligent building blocks self-assembling in a liquid [16], on an air table [12], using miniature robots [7], or by disassembly [8] and at the macro-scale using reconfigurable robots [23, 30], using mobile manipulators that are guided by an intelligent structure [27, 28], building blocks equipped

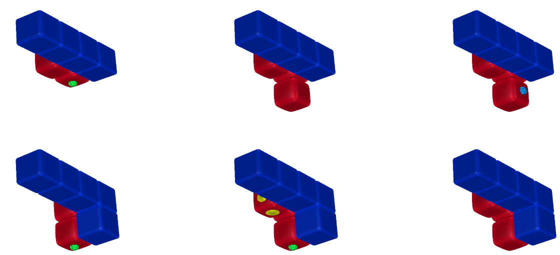

Fig. 1. Intelligent scaffolds (red) assembling a L-shaped structure from passive building blocks (blue), starting from the top, left in clock-wise order. Scaffolding blocks communicate via their faces to attract other scaffolding blocks and building blocks, or to detach from the structure.

with active markers [4], or mixed active/passive truss structures [31]. There is also a large body of theoretical work on (self-)assembly algorithms that might be enabled by future nano-robotic systems [19, 9, 3, 22, 20] or the possibility to embed intelligence into building materials at a large scale and at low cost [28, 10].

In *all* of the above approaches, building blocks equipped with sensing, actuation, computation or communication capabilities need to remain in the final structure, which is not desirable if the structure serves purely structural purposes, e.g., when used as a bridge or habitat. In the IS paradigm, however, the components containing computational capabilities can be re-used after completing the construction and only a few intelligent blocks are required to assemble structures of arbitrary size. Other *autonomous* approaches that allow for (self-)assembly of purely passive structures rely on the application of external fields, e.g., magnetic [14], passive effects [15], or stigmergy [26, 11], where the complexity of the possible structures is limited by the perception capabilities of the assembly agents, and none of these approaches allow for the assembly of arbitrary complex structures.

The IS paradigm is scale-free and independent of the locomotion mechanism of building and IS blocks. For example, IS blocks could guide a team of mobile manipulators to add passive building blocks to a structure, remove an IS block, or attach an IS block to the scaffold structure by signaling using LEDs on their faces (see Figure 1, e.g.). In this case, robot manipulators that move blocks would not need to communicate, would not need the ability to assess construction progress, and could identify where to place the next block by relying on an IS block as a marker using minimal perception abilities. At a microscopic scale, intelligent scaffolds could self-assemble in a well-stirred liquid and selectively bind passive building blocks to their surface, detach, and re-attach elsewhere on the scaffold until the desired structure is completed.

This research has immediate applications in enabling robot manipulators with limited perception capabilities to build large structures using cheap materials in the near future. We believe its real potential, however, lies in providing a theoretical framework for determining the sensing, actuation, computation and communication abilities required by intelligent building blocks on the micro- and milli-scale. Although most of the self-assembly work in this size scale uses only passive effects (capillary, electrical, shape-based, and magnetic) [15], recent results in the field suggest that NEMS and MEMS are possible substrates for the intelligent scaffolding algorithms proposed in this research. For instance [24] demonstrates controlled self-assembly of silver nano-cubes by selective functionalization of their faces, and [25] demonstrates self-assembly of polymer micro components by grafting DNA as binding agent on their faces, leveraging the power of DNA self-assembly for systems three orders of magnitude larger than the DNA itself. Here, temperature can be used to control when certain DNA strands assemble, and temperature-based sequencing of self-assembly using DNA applied to the faces of silicon and gold micro-components was demonstrated in [13]. Similarly, selective heating of micro-sized components for assembly and disassembly using solder (instead of DNA) has been shown in [6]. Also, the emerging field of nano networks makes rapid progress on wireless communication between micro-sized devices; see [2] for a recent survey of the field. Finally, although explicitly not bio-inspired, the intelligent scaffolding paradigm bears resemblance to DNA translation. Here, ribosomes correspond to intelligent scaffolds that execute a sequential program provided by mRNA, and sequentially assemble proteins whose components are delivered to it using tRNA [17].

A. Contribution and Outline

This paper introduces the intelligent scaffold paradigm, presents algorithms that compile arbitrary structures into programs for IS blocks, and demonstrate proofs that show that the IS paradigm can guide the assembly of every finite structure. Definitions are provided in Section II. Section III describes the finite state automaton implementation of intelligent scaffolds. Algorithms that generate assembly sequences for a given structure are described in Section IV, along with proofs of correctness. Algorithms and proofs are validated by simulated assembly of a series of example structures, including a model of the International Space Station, in Section V. Limitations of the approach and future work are discussed in Sections VI and VII.

II. Intelligent Scaffold Definitions

A *structure* is a connected subset of $\mathbb{Z}^3$, denoted $\mathcal{S}$. Without loss of generality, we will assume that $\langle 1,0,0\rangle$ is in $\mathcal{S}$. Positions in a structure are occupied by *blocks*, which are modeled as unit cubes. Structures can be assembled by starting with all positions empty and adding one block at a time in a position adjacent to the growing structure. The blocks do not spontaneously combine with each other;[1] scaffolds are needed to catalyze binding reactions between blocks. A block that binds to a scaffold will also bind to its neighboring blocks and become a part of the structure. More formally, a *structure assembly sequence* to assemble a structure $\mathcal{S}$ is a sequence of structures $\langle \mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_n\rangle$ such that $\mathcal{S}_0 = \varnothing$, $\mathcal{S}_n = \mathcal{S}$, and for all $t \in [n]$, $|\mathcal{S}_t| = t$.

An intelligent scaffold (IS) component is a computational object with faces that are able to connect to and disconnect from blocks or other scaffolds. IS components (sometimes referred to simply as *scaffolds*) direct the assembly of structures by indicating where and when blocks and other scaffolds are to be added to a growing structure. Scaffolds (defined more formally in Section III) are modeled as unit cubes, and work in groups to assemble structures. All scaffolds in a group must be connected to at least one other scaffold in the group and at least one scaffold in the group must be connected to the growing structure.

Scaffold groups move scaffold by scaffold, with individual scaffold components detaching and reattaching later to one of the fixed scaffolds. By a sequence of such actions, we say that the intelligent scaffold group moves along the structure. Scaffolds will also selectively bind to either blocks or other scaffolds, and will selectively detach themselves, allowing for increased control over the assembly process. We define τ to be an upper bound on the maximum size of a scaffold group. A small τ implies that the structure may be cheaper to assemble (requiring fewer computational components), but may take longer to assemble.

More formally, a *scaffold group* is a connected subset of $\mathbb{Z}^3$. Given a structure assembly sequence $\boldsymbol{S} = \langle \mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_m\rangle$, a *scaffold path for* $\boldsymbol{S}$, denoted $\boldsymbol{T}$, is a sequence of scaffold groups $\langle \mathcal{T}_0, \mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_m\rangle$ such that for all $i \in [m]$:

1) $\mathcal{T}_0 = \{\langle 0,0,0\rangle\}$ and
2) $1 \le |\mathcal{T}_i| \le \tau$ and
3) if $i < m$, then $\big||\mathcal{T}_i| - |\mathcal{T}_{i+1}|\big| = 1$ and
4) there exists a non-decreasing function $f : \mathbb{N} \to \mathbb{N}$ such that
   a) $\mathcal{T}_i \cap \mathcal{S}_{f(i)} = \varnothing$ and
   b) $\mathcal{T}_i \cup \mathcal{S}_{f(i)}$ is connected.

For all structures $\mathcal{S}$, an *IS assembly sequence for* $\mathcal{S}$ is a pair $\langle \boldsymbol{S}, \boldsymbol{T}\rangle$, where $\boldsymbol{S}$ is a structure assembly sequence to assemble $\mathcal{S}$ and $\boldsymbol{T} = \langle \mathcal{T}_0, \mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_m\rangle$ is a scaffold path for $\boldsymbol{S}$. The *assembly time* for the IS assembly sequence is m.

III. Intelligent Scaffold Automaton

Intelligent scaffolds to build finite structures (and regular infinite structures) can be implemented as finite state automata. Individual scaffolds may rely on face configuration changes (attachment of an adjacent block or attachment or detachment of an adjacent scaffold) from its neighboring scaffolds to

[1] This assumption can be relaxed if mobile manipulators attach blocks instead of random processes

TABLE I
DEFINITIONS OF SYMBOLS USED THROUGHOUT THE PAPER

Symbol	Meaning
$[k]$	{0,1,2,...,k}
$\mathcal{S}$	Structure (subset of $\mathbb{Z}^3$)
$n = \|\mathcal{S}\|$	Size of structure $\mathcal{S}$
$\boldsymbol{S}$	Structure assembly sequence (sequence of structures)
$\mathcal{T}$	Scaffold group (subset of $\mathbb{Z}^3$)
$\boldsymbol{T}$	Scaffold path (sequence of scaffold groups)
$m =$ length of $\boldsymbol{T}$	Assembly time
$\langle \boldsymbol{S}, \boldsymbol{T} \rangle$	IS assembly sequence
N_f	Face-adjacent region, or von Neumann neighborhood
N_e	Edge-adjacent region
$N_{\mathcal{S}}$	Neighborhood of $\mathcal{S}$

change its own state; thus, each scaffold will pass messages to neighboring scaffolds indicating a change of one of its faces. Messages provide the input for a scaffold to change its internal state: it does this by comparing a received message to a table of transition requirements for the current state. If one such requirement is met, the scaffold component transitions to a new state.

Note that, like many other formal models of self-assembly, the IS paradigm is a restriction of cellular automata. Every IS system can be modeled as a three-dimensional CA. However, the additional restrictions imposed in the IS system provide a better model of physical self-assembling systems than more general CA models.

A. Finite State Machine Model

Each scaffold consists of a state, a set of face configurations, and a transition table. Each face can have one of five configurations: open-to-block (O_b), open-to-scaffold (O_b), bound-to-block (B_b), bound-to-scaffold (B_s), and closed (C_c). The set of configurations is denoted $\mathcal{O}$. Open-to-scaffold can transition to bound-to-scaffold with the attachment of a scaffold; likewise open-to-block can transition to bound-to-block with the attachment of a block. Closed faces facilitate no reactions. An inter-scaffold message is a pair consisting of the state of the scaffold sending the message, and a function mapping a changed face to a new configuration. All face configuration changes results in the scaffold sending a message to all connected scaffolds (and to itself). All inter-scaffold messages must propagate through the entire scaffold group, but in most cases the scaffold group will be small relative to the size of the structure (see Conjecture 1 in Section IV).

Given a structure $\mathcal{S}$, we will define a finite state machine $M = \langle Q, \mathcal{M}, \delta \rangle$ that will run on each scaffold to give an IS assemble sequence $\langle \boldsymbol{S}, \boldsymbol{T} \rangle$. Here, Q is a finite set of *states*, $\mathcal{M}$ is a finite set of *messages*, and $\delta : Q \times \mathcal{M} \to Q$ is a *transition function*. The set Q of states and the transition function δ will be described in Section IV.

There are several notions of "neighborhood" needed to define IS systems: N_f is the face-adjacency region (the 3-dimensional von Neumann neighborhood of the origin), N_e is the edge-adjacency region (the set of positions in the 3×3 cube centered at the origin with distance two from the origin, under the ℓ^1 norm), and $N_{\mathcal{S}}$ is the neighborhood of a structure $\mathcal{S}$. More formally: $N_f = \{\langle 1,0,0\rangle, \langle -1,0,0\rangle, \langle 0,1,0\rangle, \langle 0,-1,0\rangle, \langle 0,0,1\rangle, \langle 0,0,-1\rangle\} = \{+X, -X, +Y, -Y, +Z, -Z\}$. $N_e = \{\langle -1,-1,0\rangle, \langle 1,-1,0\rangle, \langle -1,1,0\rangle, \langle 1,1,0\rangle, \langle -1,0,1\rangle, \langle 1,0,1\rangle, \langle 0,-1,1\rangle, \langle 0,1,1\rangle, \langle -1,0,-1\rangle, \langle 1,0,-1\rangle, \langle 0,-1,-1\rangle, \langle 0,1,-1\rangle\}$. For all structures $\mathcal{S}$, $N_{\mathcal{S}} = \{\boldsymbol{x} \in \mathbb{Z}^3 \setminus \mathcal{S} \mid \exists \boldsymbol{y} \in \mathcal{S}, \boldsymbol{z} \in N_e \cup N_f \wedge \boldsymbol{x} = \boldsymbol{y} + \boldsymbol{z}\}$ is the neighborhood of $\mathcal{S}$.

Let $C_{\text{all}} = \{c \mid c : N_f \to \mathcal{O}\}$ be the set of all functions from N_f to $\mathcal{O}$, each of which is a possible *face configuration* of an intelligent scaffold component. $\mathcal{M} = Q \times C_{\text{all}}$ is the set of all possible messages for a given state machine.

At each time $t \in \mathbb{N}$, positions in $\mathbb{Z}^3$ (referred to as *cells*) can be either empty, occupied by a scaffold, or occupied by a block. At time $t = 0$, there is one *seed scaffold* (canonically at position $\langle 0,0,0\rangle$) and the other cells are empty. Cells can transition from empty to scaffold and back, but the transition to a block is final. The seed state $q_0 \in Q$ is a special state that begins the assembly process. By convention, the seed state cannot be reached by any other state to prevent identical structures from appearing elsewhere in the assembly. The seed scaffold will begin the assembly sequence: new scaffolds and/or structure blocks will attach to the seed scaffold, triggering messages and state transitions. For a fixed scaffold to bind to a free scaffold, the fixed scaffold must have an open-to-scaffold face, but the free scaffold attachment face may be closed.

A newly placed scaffold component will start in a default state. Upon receiving the connection message from its neighbor, a default state will transition to another state. The default state has all six of its faces closed. To detach, a scaffold will transition to the default state. This will cause it to detach and become free, making the cell it was occupying empty. Default and disconnected scaffolds can then be reused. Through progressive attachments, state changes, and detachments, a structure will emerge. The detachment process allows a small number of scaffolds to be used to construct arbitrary finite structures.

More formally, for all $t \in \mathbb{N} \cup \{0\}$, $\mathcal{A}_t : \mathbb{Z}^3 \to \{b, s, \epsilon\}$ is the *assembly space* at time t, where ϵ represents an empty cell, b represents a cell occupied by a block, and s represents a cell occupied by a scaffold. Let $M = \langle Q, \Sigma, \delta \rangle$ be the FSM running on the scaffolds. If $\mathcal{A}_t(\boldsymbol{x}) = s$ for some $t \in \mathbb{N}$ and $\boldsymbol{x} \in \mathbb{Z}^3$, we will denote the state of M on s as s_q and the face configuration of s as s_c. For all structures $\mathcal{S}$, we say that M *assembles* $\mathcal{S}$ iff for all $t \in \mathbb{N} \cup \{0\}$, for all $\boldsymbol{x} \in \mathbb{Z}^3$:

1) $\mathcal{A}_0(\langle 0,0,0\rangle) = s$ in state q_0 and
2) if $\boldsymbol{x} \neq \langle 0,0,0\rangle$ then $\mathcal{A}_0(\boldsymbol{x}) = \epsilon$ and
3) if $\mathcal{A}_t(\boldsymbol{x}) = \epsilon$ and $\mathcal{A}_{t+1}(\boldsymbol{x}) = b$ then there exists $d \in N_f$ such that $\mathcal{A}_t(d(\boldsymbol{x})) = s$, where $s_{c(d^{-1}(\boldsymbol{x}))} = O_b$ and
4) if $\mathcal{A}_t(\boldsymbol{x}) = \epsilon$ and $\mathcal{A}_{t+1}(\boldsymbol{x}) = s$ then there exists $d \in N_f$ such that $\mathcal{A}_t(d(\boldsymbol{x})) = s$, where $s_{c(d^{-1}(\boldsymbol{x}))} = O_s$ and
5) $\mathcal{A}_m^{-1}(b) = \mathcal{S}$

IV. Algorithms

This section will describe the algorithms used to generate an IS assembly sequence for a given structure. We will first describe an algorithm that transforms a given finite structure into an action list (a list of block attachments and scaffold attachments and detachments), then we will describe an algorithm that transforms the action list into a finite state machine that can be run on the scaffold components to assemble the structure.

A. Finite Structure Assembly Planner

The scaffold planner algorithm will attempt to find a sequence of scaffold movements and block placements that construct the structure without violating any of the following rules:

1) All new blocks must be face-adjacent to a block already on the structure.
2) New scaffolds must be attached (face-adjacent) to existent scaffolds.
3) The scaffold group must be contiguous and always have at least one member attached to the structure.
4) No more than a specified number of scaffolds are used.

Pseudocode for the scaffold planner is given in Algorithm 1. The scaffold planner computes an IS assembly sequence by which a set of scaffolds can incrementally assemble a connected structure without a contiguous scaffold group detaching at any step in the process. Following the initial placements of the seed scaffold and seed block in lines 2–5, blocks are added to the structure in an incremental fashion, such that all new blocks are connected to an existent block. This ensures that Rule 1 is followed, and is implemented in lines 7–8 and 15–16 by choosing the next block to be added from the neighborhood of the current structure. If none of the remaining blocks are reachable by the scaffold group, the algorithm backs up to a point where it could have chosen a different assembly sequence and attempts to assemble the structure from that point in lines 9–14. The algorithm finds a scaffold path from the current scaffold group to the next block, subject to Rules 2–4, in lines 17–19. Finally, the action list is updated in lines 20–22.

To prove that all finite structures are assemblable, we will first prove a lemma that all finite structures that do not enclose empty spaces are assemblable:

Lemma 1: For all finite structures $\mathcal{S}$ such that $\mathbb{Z}^3 \setminus \mathcal{S}$ is connected, there exists an IS assembly sequence $\langle \boldsymbol{S}, \boldsymbol{T} \rangle$ for $\mathcal{S}$.

Proof: Let $\mathcal{S}$ be a finite structure such that $\mathbb{Z}^3 \setminus \mathcal{S}$ is connected. Then there exists a structure assembly sequence $\boldsymbol{S} = \langle \mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_n \rangle$ for $\mathcal{S}$ such that for all $i \in [n]$, $\mathbb{Z}^3 \setminus \mathcal{S}_i$ is connected. Note that $\langle \mathcal{T}_0 \rangle$ is a scaffold path for $\langle \mathcal{S}_0 \rangle$. Let $i \in [n-1]$ and assume there exists a scaffold path $\boldsymbol{T}_i = \langle \mathcal{T}_0, \mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_j \rangle$ for $\boldsymbol{S}_i = \langle \mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_i \rangle$. Since $\mathbb{Z}^3 \setminus \mathcal{S}_i$ is connected, there exists a path $P = \langle \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k \rangle$ in $\mathbb{Z}^3 \setminus \mathcal{S}_i$ from $\mathcal{T}_j$ to $\mathcal{S}_{i+1} \setminus \mathcal{S}_i$. Then $\boldsymbol{T}_{i+1} = \langle \mathcal{T}_0, \mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_j, \mathcal{T}_j \cup \{\boldsymbol{x}_1\}, \mathcal{T}_j \cup \{\boldsymbol{x}_1, \boldsymbol{x}_2\}, \ldots \mathcal{T}_j \cup \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{k-1}\} \rangle$ is a scaffold path for $\mathcal{S}_{i+1}$. The lemma follows by induction. ■

Theorem 1: For all finite structures $\mathcal{S}$, there exists an IS assembly sequence $\langle \boldsymbol{S}, \boldsymbol{T} \rangle$ for $\mathcal{S}$.

Proof: Let $\mathcal{S}$ be a finite structure. Let $C = \{C_0, C_1, C_2, \ldots, C_k\}$ be the connected components of $\mathbb{Z}^3 \setminus \mathcal{S}$. Then there exists a minimum spanning tree $U \subset \mathbb{Z}^3$ such that

1) the root of $U \notin \mathcal{S}$ and
2) for all $i \in [k]$ there exists $\boldsymbol{x} \in C_i$ such that $\boldsymbol{x}$ is a leaf of U and
3) $\mathcal{S} \setminus U$ is connected.

Let $\mathcal{S}' = \mathcal{S} \setminus U$. Then $\mathbb{Z}^3 \setminus \mathcal{S}'$ is connected, so $\mathcal{S}'$ is assemblable by Lemma 1. Let $\langle \boldsymbol{S}', \boldsymbol{T}' \rangle$ be an IS assembly sequence for $\mathcal{S}'$. Extend the scaffold path $\boldsymbol{T}'$ to a scaffold path $\boldsymbol{T}''$ such that the last scaffold group of $\boldsymbol{T}''$ is U.

Consider a sequence of positions $\langle \boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_j \rangle$ such that each $\boldsymbol{x}_i$ is in U and the distance from the root to $\boldsymbol{x}_i$ is non-increasing in i. Extend the scaffold path $\boldsymbol{T}''$ to a scaffold path $\boldsymbol{T}$ by removing the position $\boldsymbol{x}_i$ from last scaffold group in turn. Finally, extend the structure assembly sequence $\boldsymbol{S}'$ to a structure assembly sequence $\boldsymbol{S}$ by adding each position $\boldsymbol{x}_i$ in turn if $\boldsymbol{x}_i \in \mathcal{S}$. Then $\langle \boldsymbol{S}, \boldsymbol{T} \rangle$ is an IS assembly sequence for $\mathcal{S}$. ■

Conjecture 1: There exist structures $\mathcal{S}$ such that a group of at least 3 scaffolds is required to assemble $\mathcal{S}$ and all structures can be assembled with at most 3 scaffolds.

We believe that three scaffolds are necessary to assemble some structures in order to keep the scaffold group connected to the growing structure while the group turns corners. For example, Figure 3 shows a structure in which it appears that scaffold groups must turn corners in order to place all the blocks. We believe that three scaffolds are sufficient to assemble all structures because a scaffold group of size three can move along the outside of a structure to get to any necessary position, so long as the group does not completely enclose itself with blocks.

The following two lemmas demonstrate that a scaffold group can access all positions in its moveSpace in such a way that blocks can be placed at any position that is face-adjacent to the growing structure:

Lemma 2: For all structures $\mathcal{S}$, for all scaffold groups $\mathcal{T}$ such that $\mathcal{S} \cap \mathcal{T} = \varnothing$ and $\mathcal{S} \cup \mathcal{T}$ is connected, if moveSpace is the connected component of $N_{\mathcal{S}}$ that contains $\mathcal{T}$ then for all $\boldsymbol{x} \in$ moveSpace, there exists a scaffold path $\langle \mathcal{T}_0, \mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_m \rangle$ such that $\mathcal{T}_0 = \mathcal{T}$ and $\boldsymbol{x} \in \mathcal{T}_m$.

Proof: Let $\mathcal{S}, \mathcal{T}$, moveSpace, and $\boldsymbol{x}$ be as in the statement of the lemma. Since moveSpace is connected, there exists a path $P = \langle \boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_k \rangle$ such that $\boldsymbol{y}_0 \in \mathcal{T} \cap$ moveSpace, $\boldsymbol{y}_k = \boldsymbol{x}$, and for all $i \in [k]$, $\boldsymbol{y}_i \in$ moveSpace. Let $\mathcal{T}_0 = \mathcal{T}$ and for $i \in \{1, 2, \ldots, k\}$, let $\mathcal{T}_i = \mathcal{T}_{i-1} \cup \{\boldsymbol{y}_i\}$. Then $\langle \mathcal{T}_0, \mathcal{T}_1, \ldots, \mathcal{T}_k \rangle$ is a scaffold path such that $\mathcal{T}_0 = \mathcal{T}$ and $\boldsymbol{x} \in \mathcal{T}_m$. ■

Lemma 3: For all structures $\mathcal{S}$, for all $\boldsymbol{x} \in N_{\mathcal{S}}$, for all scaffold groups $\mathcal{T}$ such that $\mathcal{S} \cap \mathcal{T} = \varnothing$ and $\mathcal{S} \cup \mathcal{T}$ is connected, if moveSpace is the connected component of $N_{\mathcal{S}}$ that contains

Algorithm 1 Finite Structure Assembly Planner

def ScaffoldPlanner: structure $\mathcal{S}$, maxScaffolds τ
$\boldsymbol{S} = \langle \mathcal{S}_0, \mathcal{S}_1 \rangle$ where $\mathcal{S}_0 = \varnothing$ and $\mathcal{S}_1 = \{\langle 1,0,0 \rangle\}$.
$\boldsymbol{T} = \langle \mathcal{T}_0 \rangle$ where $\mathcal{T}_0 = \{\langle 0,0,0 \rangle\}$.
$n = 1$, $m = 0$, $m_n = m$
actionList$_n$ $= \langle$addScaffold$(\langle 0,0,0 \rangle)$, addBlock$(\langle 1,0,0 \rangle)\rangle$.
while $\mathcal{S}_n \neq \mathcal{S}$ **do**
  moveSpace$_n$ = connected component of $N_{\mathcal{S}_n}$ containing $\mathcal{T}_m$
  nextBlocks$_n$ $= \mathcal{S} \cap$ moveSpace$_n$
  while nextBlocks$_n$ $= \varnothing$ **do**
    $n = n - 1$, $m = m_n$
    if $n = 1$ **then**
      return; no assembly sequence exists.
    end if
  end while
  Choose closest nextBlock $\in$ nextBlocks$_n$ to $\mathcal{T}_m$.
  nextBlocks$_n$ = nextBlocks$_n$ $\setminus$ {nextBlock}
  nextBlockScaffolds = nextBlock face-adjacency spaces $\cap$ moveSpace$_n$
  Find a path from $\mathcal{T}_m$ to nextBlockScaffolds that follows Lemma 3.
  While moving the scaffold group, increment m for each change in $\mathcal{T}_m$ and maintain $|\mathcal{T}_m| \leq \tau$.
  actionList$_n$ = addScaffold$(x \in \mathbb{Z}^3)$ and/or detachScaffold$(x \in \mathbb{Z}^3)$ for each change in $\mathcal{T}$.
  $\mathcal{S}_{n+1} = \mathcal{S}_{\mathrm{n}} \cap$ {nextBlock}
  Add addBlock$(x \in \mathbb{Z}^3)$ to actionList$_n$
  $n = n + 1$, $m_n = m$
end while
return actionList

$\mathcal{T}$ and $\boldsymbol{x} \in$ moveSpace, then there exists a scaffold path $\langle \mathcal{T}_0, \mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_m \rangle$ such that $\mathcal{T}_0 = \mathcal{T}$ and $\boldsymbol{x} \notin \mathcal{T}_m$ and $\boldsymbol{x} \in \{\boldsymbol{z} \in \text{moveSpace} \mid \boldsymbol{y} \in N_f \wedge \boldsymbol{z} = \boldsymbol{x} + \boldsymbol{y}\}$.

Proof: Let $\mathcal{S}$, $\mathcal{T}$, moveSpace, and $\boldsymbol{x}$ be as in the statement of the lemma. Let $Z = \{\boldsymbol{z} \in \text{moveSpace} \mid \boldsymbol{y} \in \text{moveSpace} \wedge \boldsymbol{z} = \boldsymbol{x} + \boldsymbol{y}\}$. For all $\boldsymbol{z} \in Z$, since moveSpace is connected there exists a path $P_{\boldsymbol{z}}$ from $\mathcal{T}$ to $\boldsymbol{z}$. Consider a shortest such path, P'. Then P' ends at $\boldsymbol{z}' \in Z$ and $\boldsymbol{x} \notin P'$ (otherwise there would be a $\boldsymbol{z}'' \neq \boldsymbol{z}'$ that had a shorter path). By Lemma 2 there exists a scaffold path $\boldsymbol{T}$ that begins with $\mathcal{T}$ at ends with a scaffold group $\mathcal{T}_k$ such that $\boldsymbol{x} \notin \mathcal{T}_k$ but $\boldsymbol{x}$ is adjacent to $\mathcal{T}_k$. ■

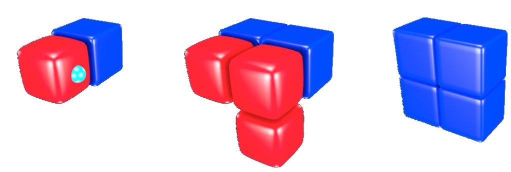

Fig. 2. Three snapshots of the square assembly in progress. The left-most shows the scaffold in state 2 just before a scaffold attaches (the blue circle indicates that the face is open to a scaffold). The middle shows three scaffolds, with the lower right scaffold in state 6 just before a block attaches. The right-most shows the finished assembly after all the scaffolds have detached.

B. Compiler Algorithm

The actionList returned by the Finite Structure Assembly Planner contains sufficient information to create M, the finite state machine. Scaffolds at each position in the actionList represent states, and actions adjacent to that scaffold such as "addBlock" and "addScaffold" translate to O_b or O_s. For a scaffold to detach ("detachScaffold"), the action in actionList immediately before "detachScaffold" is translated to a message $\mu \in \mathcal{M}$ representing that previous action. The transition table for the scaffold and μ would then lead to the default state.

V. Examples

This section presents examples of structures that can be built with intelligent scaffolding, and include a 2×2 lattice as well as the scaffold state machine that was generated by the compiler, a fractal structure that maximizes backtracking, and a model of the International Space Station (ISS) that demonstrates the assembly of structures with many building blocks.

a) Square: A 2×2 square can be created with an 8-state intelligent scaffolding machine, including the default state (labeled 0) and the seed state (labeled 1). Table II describes an intelligent scaffolding state machine that can construct a 2×2 square (Figure 2). In this example, the square is formed on an adjacent plane beneath the scaffold layer, although such restrictions are not necessary. Note that state 1, the seed, cannot be reached by any other state. This restriction means that no inputs can change a scaffold component to the seed state, so it must be done independently. The seed state is assumed to exist at $\langle 0,0,0 \rangle$ as determined by Algorithm 1. The

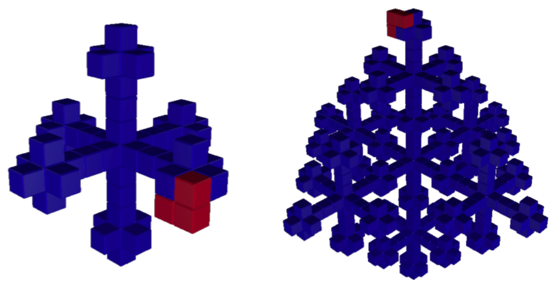

Fig. 3. Instances of a fractal structure that maximizes scaffolding backtracking moves with respect to total number of building blocks.

seed's $-Z$ state is set to open-to-block. When a block connects to that face, the seed transitions to state 2. State 2 opens its $+X$ face to a scaffold. When a default scaffold connects to state 2, the connected face transmits the connection message $\langle 2, \langle\langle 1,0,0\rangle, B_s\rangle\rangle$ to the new scaffold. The new scaffold, starting in state 0, receives the message and transitions to state 3. The rest of the scaffold and block placements proceed similarly. Note that the transition message for states 2, 4, 6, and 7 are same. When state 7 lays the final block, all scaffolds go to state 0 and thus detach, having finished the task. Empty spaces in the state transition table correspond to the state transitioning to itself for a given message.

b) Fractal Snowflakes: If few scaffolds are used, as might be desirable for cost considerations, then to assemble some structures a scaffold group may have to "backtrack" in order to reach all the positions in which blocks need to be placed. By "backtracking", we mean that the scaffold group must move along parts of the structure that have already been completed, without placing any new blocks. Figure 3 shows a fractal structure in which backtracking is necessary if only three IS components are used in the scaffold group. The structure is a three-dimensional, recursive cross, and after assembling one arm of the cross, the scaffold group must move back along the completed are in order to reach the other arms.

c) International Space Station: Finally, we would like to demonstrate that the IS paradigm is able to handle large-scale structures. We obtained a CAD model of the ISS from [5] and converted it into a three-dimensional lattice graph containing 3,782 building blocks. The assembly sequence (actionList length 45,637) was calculated by the Scaffold Planner algorithm for three IS blocks. Construction began at the lower left corner of the model (corresponding to one of the solar panels) and continued outward until the sequence concluded in the opposite corner. Although the algorithm presented in this paper is limited to constructing structures, we believe extending the presented algorithm to handle different kinds of building blocks — that might enable the construction of *functional* structures — is straightforward.

VI. Discussion

The algorithms presented in this paper are scale-free and independent of how building and scaffold blocks locomote. Microscale scaffold blocks might swim in a liquid or might be placed and removed by mobile manipulators. Although the IS paradigm might drastically reduce the coordination and perception requirements of such a manipulation system, this comes at a price: assembling a structure made of n passive building blocks usually requires multiple manipulation steps. Therefore, the IS paradigm becomes less competitive the more capable robots become. However, even if robots are able to assemble structures exclusively from passive building blocks, i.e., without relying on markers on the structure, the IS paradigm could be used to support the assembly of structures that require scaffolds during assembly to overcome gravity.

We have shown that the intelligent scaffold structure can build any finite structure. Depending on the building blocks and locomotion-scheme actually used, there might be additional constraints that prevent a structure from completion. For example, [28] considers blocks that cannot be inserted between two blocks already placed. Also, the workspace of a robotic manipulator might be limited and may not place blocks into arbitrary configurations. Finally, a scaffold structure might also be implemented as a snake or a paper that climbs along the emerging structure by folding, leading to additional constraints and redundant motions that need to be taken into account. We believe that all of these constraints can be modeled in the assembly planner, which is subject to further work.

We believe that three scaffold components are sufficient for assembling any finite structure; in particular we have not found any structure that cannot be assembled by at most three scaffolds. We intend to try to prove or disprove that conjecture in the future. If this conjecture holds, this gives the intelligent scaffolding model an enormous advantage over other self-assembly methods. Not only will arbitrarily large structures be possible with only a few scaffolds, the scaffolds themselves can be reused. This will allow a set of intelligent scaffolds to mass produce structures. If a large number of scaffolds are deployed in a medium with blocks, several structures can be built at once. The blocks themselves only need the requirement that they cannot spontaneously combine unless a scaffold is present to catalyze the reaction.

Although we have not formally analyzed the complexity of the proposed algorithms, all of the structures that we have assembled required state machines roughly on the order of the number of blocks. Considering the computational expressiveness, we believe that the IS paradigm can be reduced to the tile assembly model [20]. The tile assembly model has been shown to be Turing universal, and we conjecture that also the IS paradigm will allow us to implement every possible computation as an assembly process. In turn, we could use this capability to find trade-offs between computation, communication, memory and additional assembly steps.

Some of the tenets of the mathematical model may change to account for realistic computational scenarios such as error-handling and unexpected events. Error-handling in the current algorithm is limited and any single error in the assembly process, including communication errors, could ruin the expected result. We plan to address this issue in future work by extending the scaffold program by error handling routines

TABLE II
2 × 2 SQUARE INTELLIGENT SCAFFOLD STATE MACHINE

State, (Face Configuration)	Message: $\langle q \in Q,\, n_f \in N_f,\, o \in \mathcal{O} \rangle$						
$q \in Q$ ($n_f \in N_f$, $o \in \mathcal{O}$)	$\langle 1, -Z, B_b \rangle$	$\langle 2, +X, B_s \rangle$	$\langle 3, -Z, B_b \rangle$	$\langle 4, -Y, B_s \rangle$	$\langle 5, -Z, B_b \rangle$	$\langle 6, -X, B_s \rangle$	$\langle 7, -Z, B_b \rangle$
0 (C_c)		3		5		7	
1 ($-Z, O_b$)	2						
2 ($+X, O_s$)							0
3 ($-Z, O_b$)			4				
4 ($-Y, O_s$)							0
5 ($-Z, O_b$)					6		
6 ($-X, O_s$)							0
7 ($-Z, O_b$)							0

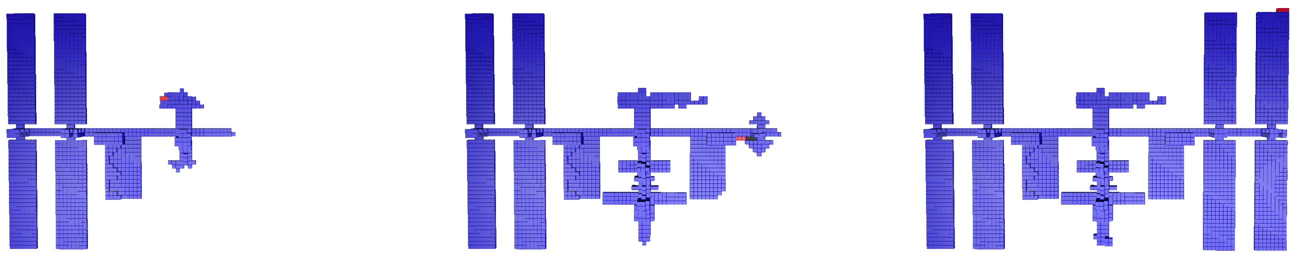

Fig. 4. A 3782-block model of the International Space Station is built incrementally using 3 intelligent scaffold units.

and investigate algorithms that can move a scaffold along a structure to repair it.

The finite state machine model may also change. For example, we might require that a single state have only one open face. Having multiple open faces introduces a race condition if the order of the connections is not predetermined. A poorly designed state machine could take different paths depending on the order that the faces on a single scaffold are bound. Another example is the transition function. Currently, only messages are considered as input.

Finally, we can consider different models of computation in the intelligent scaffolds. For instance, we may allow the IS components to be Turing universal instead of limited to finite state machines.

VII. CONCLUSION AND FUTURE WORK

This paper presents a novel paradigm for autonomous assembly in which assembly is coordinated using intelligent scaffold blocks. Intelligent scaffolding blocks store a representation of the structure to be assembled and keep track of the assembly state. At the macro-scale, where robots assemble a structure, the scaffold coordinates building agents, and simplifies their perception by active signaling, yet allow to construct structures from *passive* blocks. At the micro-scale, where both scaffold and building blocks might move in a stirred liquid, the scaffold selectively binds to itself or free-floating building blocks. We show analytically that intelligent scaffold blocks can assemble any finite structure. We conjecture that every structure can be constructed with at most three scaffolding blocks, and demonstrate the assembly of a structure composed of 3,782 passive blocks using only three intelligent scaffold blocks in simulation. Limitations of the proposed approach are that the complexity of the resulting structure is limited by the memory available on each intelligent scaffold block and that re-arranging the IS blocks requires additional manipulation steps.

In future work, we plan to give our algorithms the ability to compress the resulting FSMs, e.g., by automatically compiling sub-routines for recurrent elements in a structure. We also want to analytically derive an upper bound on the maximum number of assembly steps for any structure, and investigate algorithms that minimize backtracking. Further, we will investigate using the IS paradigm to assemble potentially infinite structures. We will validate the IS paradigm by using it to coordinate the autonomous assembly of 3D structures by a team of mobile manipulators with limited perception capabilities. With respect to a possible implementation of the IS paradigm at the microscale, we will also validate the IS paradigm in 2D on an air-table. Along these lines, we are also interested in formal theory on the minimal sensing, actuation, computation and communication requirements for IS blocks to assemble a specific structure, as well as in exploring trade-offs between sensing, locomotion, computation and communication. Finally, we are interested in investigating algorithms that will allow the intelligent scaffold to split up in sub-teams. By this, IS building blocks could not only work on multiple structures in parallel, but also assemble a single structure in parallel and reduce back-tracking by serving as markers.

REFERENCES

[1] L. Adleman. Molecular computation of solutions to combinatorial problems. *Science*, 266:1021–1024, 1994.

[2] I. Akyildiz and J. Jornet. Electromagnetic wireless nanosensor networks. *Nano Communication Networks*, 1(1):3–19, 2010.

[3] D. Arbuckle and A. Requicha. Self-assembly and self-

repair of arbitrary shapes by a swarm of reactive robots: algorithms and simulations. *Autonomous Robots*, 28(2): 197–211, 2010.

[4] A. Bolger, M. Faulkner, D. Stein, L. White, .S-K. Yun, and Daniela Rus. Experiments in decentralized robot construction with tool delivery and assembly robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.

[5] M. Carbajal. Cad model of the international space station, October 2008. URL `http://www.nasa.gov/multimedia/3d_resources/assets/iss_c2.html`. Last retrieved January 16, 2011.

[6] J. Chung, W. Zheng, T. Hatch, and H. Jacobs. Programmable reconfigurable self-assembly: parallel heterogeneous integration of chip-scale components on planar and nonplanar surfaces. *J. Microelectromech. Syst.*, 15: 457–64, 2006.

[7] W. Evans, G. Mermoud, A. Martinoli, and V. Kumar. Comparing and modeling distributed control strategies for miniature self-assembling robots. In *IEEE International Conference on Robotics and Automation*, pages 1438–1445, Anchorage, AK, 2010.

[8] K. Gilpin, K. Kotay, D. Rus, and I. Vasilescu. Miche: Modular shape formation by self-disassembly. *The International Journal of Robotics Research*, 27(3–4):345–372, 2008.

[9] S. Goldstein, J. Campbell, and T. Mowry. Programmable matter. *Computer*, 38(6):99–101, 2005.

[10] A. Grushin and A. Reggia. Parsimonious rule generation for a nature-inspired approach to self-assembly. *ACM Transactions on Autonomous and Adaptive Systems*, 5 (3), 2010.

[11] O. Holland and C. Melhuish. Stigmergy, self-organisation, and sorting in collective robotics. *Artificial Life*, 5(2):173–202, 1999.

[12] E. Klavins. Programmable self-assembly. *Control Systems Magazine*, 24(4):43–56, August 2007.

[13] T. Kusakabe, T. Tanemura, Y. Higuchi, K. Sugano, T. Tsuchiya, and O. Tabata. DNA mediated sequential self-assembly of nano/micro components. In *Proc. of the Int. Conf. on Micro Electro Mechanical Systems*, pages 1052–1055, Tucson, AZ, 2008.

[14] S. Martel and M. Mohammadi. Using a swarm of self-propelled natural microrobots in the form of flagellated bacteria to perform complex micro-assembly tasks. In *2010 IEEE International Conference on Robotics and Automation (ICRA)*, pages 500–505, May 2010.

[15] M. Mastrangeli, S. Abbasi, C. Varel, C. Van Hoof, J.-P. Celis, and K. Boehringer. Self-assembly from milli- to nanoscales: methods and applications. *Journal of Micromechanics and Microengineering*, 19(8):37pp, 2009.

[16] P.White, V. Zykov, J. Bongard, and H. Lipson. Three dimensional stochastic reconfiguration of modular robots. In *Robotics: Science and Systems*, 2005.

[17] V. Ramakrishnan. Ribosome structure and the mechanism of translation. *Cell*, 108(4):557 – 572, 2002.

[18] D. Reishus, B. Shaw, Y. Brun, N. Chelyapov, and L. Adleman. Self-assembly of DNA double-double crossover complexes into high-density, doubly connected, planar structures. *Journal of American Chemical Society (JACS)*, 127(50):17590–17591, November 2005.

[19] A. Requicha. Nanorobots, NEMS and nanoassembly. *Proceedings of the IEEE, Special Issue on Nanoelectronics and Nanoprocessing*, 91(11):1922–1933, 2003.

[20] P. Rothemund and E. Winfree. The program-size complexity of self-assembled squares. In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing (STOC00)*, pages 459–468, Portland, OR, USA, May 2000.

[21] P. Rothemund, N. Papadakis, and E. Winfree. Algorithmic self-assembly of DNA Sierpinski triangles. *PLoS Biology*, 2(12):e424, 2004.

[22] M. Rubenstein and W. Shen. Automatic scalable size selection for the shape of a distributed robotic collective. In *Int. Conf. on Intelligent Robots and Systems*, Taipei, Taiwan, October 2010.

[23] D. Rus, Z. Butler, K. Kotay, and M. Vona. Self-reconfiguring robots. *Communications of the ACM*, 45 (3):39–45, 2002.

[24] M. Rycenga, J. McLellan, and Y. Xia. Controlling the assembly of silver nanocubes through selective functionalization of their faces. *Advanced Materials*, 20(12): 2416–2420, June 2008.

[25] T. Tanemura, G. Lopez, R. Sato, K. Sugano, T. Tsuchiya, O. Tabata, M. Fujita, and M. Maeda. Sequential and selective self-assembly of micro components by DNA grafted polymer. In *IEEE 22nd International Conference on Micro Electro Mechanical Systems*, pages 184–187, Sorrento, 2009.

[26] G. Theraulaz and E. Bonabeau. Coordination in distributed building. *Science*, 269:686–688, 1995.

[27] J. Werfel and R. Nagpal. Extended stigmergy in collective construction. *IEEE Intelligent Systems*, 21(2):20–28, 2006.

[28] J. Werfel and R. Nagpal. Three-dimensional construction with mobile robots and modular blocks. *International journal of robotics research*, 3-4(27):463–479, 2008.

[29] G. Whitesides and B. Grzybowski. Self-assembly at all scales. *Science*, pages 2418–2421, 2002.

[30] M. Yim, W.-M. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, and G. Chirikjian. Modular self-reconfigurable robot systems—challenges and opportunities for the future. *IEEE Robotics and Automation Magazine*, pages 43–53, March 2007.

[31] S. Yun, Y. Yoon, and D. Rus. Self assembly of modular manipulators with active and passive modules. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems Workshop on Self-Reconfigurable Robots*, San Diego, USA, October 2007.