Robots Building Bridges, Not Walls

1st Yiwen Hua
Sibley School of MAE
Cornell University
Ithaca, USA
yh686@cornell.edu

2nd Yawen Deng
Sibley School of MAE
Cornell University
Ithaca, USA
yd298@cornell.edu

3rd Kirstin Petersen
Electrical and Computer Engineering
Cornell University
Ithaca, USA
kirstin@cornell.edu

***Abstract*—The TERMES system is a robot collective capable of constructing 2.5D user-specified structures with specialized bricks. This work extends the original system, by enabling 3D construction without added complexity in the robots. To do this, we introduce an expandable brick which complies with the original TERMES hardware and is inexpensive and fast to fabricate. We further show a decentralized algorithm that permits an arbitrary number of robots to use both original and expandable bricks to build structures with overhangs over convex cavities, i.e. with bridges and roofs. Finally, we discuss a mechanical redesign of the robots towards decreased system cost, fabrication and maintenance time. Although more work is needed to realize construction of large-scale overhangs in practice, our work represents an important step towards construction of complex structures by minimalistic and scalable robot collectives.**

***Index Terms*—robot collectives, automated construction, embodied intelligence, swarm intelligence**

I. Introduction

Interest in automated construction is increasing due in part to new technological enablers, and in part to societal demands stemming from lack of adequate housing, worker safety, and the potential to construct in unprecedented scenarios [1]. Recent progress has largely focused on pre-fabrication, additive manufacturing, and robot-assisted technologies. A more distant goal, is that of autonomous mobile robots collaborating to construct user-specified structures. These collectives may achieve high *efficiency*, because many can work on the same task in parallel, and *robustness*, due to the redundancy of agents [2]. As of yet, this thrust remains largely in research, however, construction by collectives in nature are proof that complex global behaviors can emerge from from local interactions between many individuals [3].

The emphasis in collective robotic construction is on *scalability*; i.e. how to make algorithms that efficiently coordinate large collectives and how to make robust hardware that is easily fabricated and maintained [4]. TERMES, presented in [5], is an example of such a system comprised of 1) an algorithmic framework, where agents coordinate implicitly through observations and modifications of a shared environment, and 2) a robotic implementation that leverages passive mechanical features and co-design to enable comparatively large scale construction of user-specified structures. TERMES was shown to complete 2.5D structures, i.e. multi-layer walls in which upper layers were directly supported by those underneath; here, we extend this work to further permit construction of overhangs such as bridges and roofs. Intuitively, this goal can be accomplished with more advanced sensors and manipulators. Instead, to avoid increased complexity and risk of malfunctions, we introduce slightly enhanced bricks that passively unfold upon deposition. Beyond the practicality of roofed structures, this study is an exercise in minimalism through a holistic design process; exploring how much complexity can be achieved with large collectives of very simple robots. Future research may build on these concepts to expand the system capabilities.

This work was supported by GETTYLAB.

II. Related Work

The idea of shifting functionality from the robots into the material they manipulate has been explored previously, for example through memory enhanced building blocks [6], [7], blocks that direct construction [8], [9], and stimuli-responsive materials [10]. At the extreme end of scale are modular self-reconfigurable robots [11], where the robots themselves become the building material at the expense of significantly increased cost, fabrication, and maintenance. To the best of our knowledge, researchers have yet to explore robotic construction with passively deployable discrete depositions, although similar trends exist in consumer products like pop-up tents and self-inflatable mattresses.

Related literature on multi-robot systems capable of building overhangs is sparse and have only been demonstrated by small teams of closely collaborating robots ranging from beam placement with ground-locked robots [12], [13] and quadcopters [14], to extruded fiber shells [15], and strut structures assembled in weightless environments [16]. In contrast, we focus on multi-element overhangs, robots that climb to reach higher levels of construction, and environmentally-mediated coordination that scales to large collectives and structures.

III. The TERMES System

The TERMES algorithm consist of an off-line compiler and a fixed set of rules to guide the robots [5]. The compiler converts the structure blueprint to a 2D-map with assembly locations, the desired number of bricks at each location, and one-directional motion constraints between locations. This map is given to an arbitrary number of robots, which follow the map and add material as determined by the rule set. Together, these ensure that construction can commence with provable guarantees, despite the limited abilities of the robots to sense and communicate.

Specifically, the TERMES robots are capable of 1) climbing up/down the height of one brick, 2) pick up a brick from a cache, carry it, and deposit it on a level surface, 3) sense other robots and perform obstacle avoidance, 4) sense its location on the structure, the relative height of the location ahead, and if it is ascending or descending a step, 5) keep track of its global location by counting steps from a seed location, and 6) climb off the structure and return to the seed by following the structure perimeter [17]. The goal of the algorithm is therefore to prevent situations that hinder further construction, i.e. placement of bricks to form cliffs/walls that the robots cannot climb or gaps in the structure which cannot be filled [5].

The brick and robots are carefully co-designed (Fig. 1). The robots are approximately the size of a normal brick, weigh ∼850g, and cost $1,8K. They locomote using a wheel-leg combination (whegs, 1), a 1DOF claw for brick grasping (2), and a shelf to rest the bricks during transport (3). To navigate around the structure, the robot uses ultrasonic transceivers (4) which reflect off the rippled edge of the brick (5) independent of their relative angle. The bricks are slightly bigger than the robot footprint, weighing approximately 240g, and costing about $20. They have an indented bowl (6) that keeps the robots turning accurately without advanced sensors, despite the changing curvature of the whegs; notches (7) to make them passively align while climbing, and a filleted handle (8) that permits easy grasping. Robots sense their position using IR sensors and black and white patterns on the brick (9). Finally, bricks stack and align easily via magnets (10) and mechanically inverted features on their horizontal surfaces.

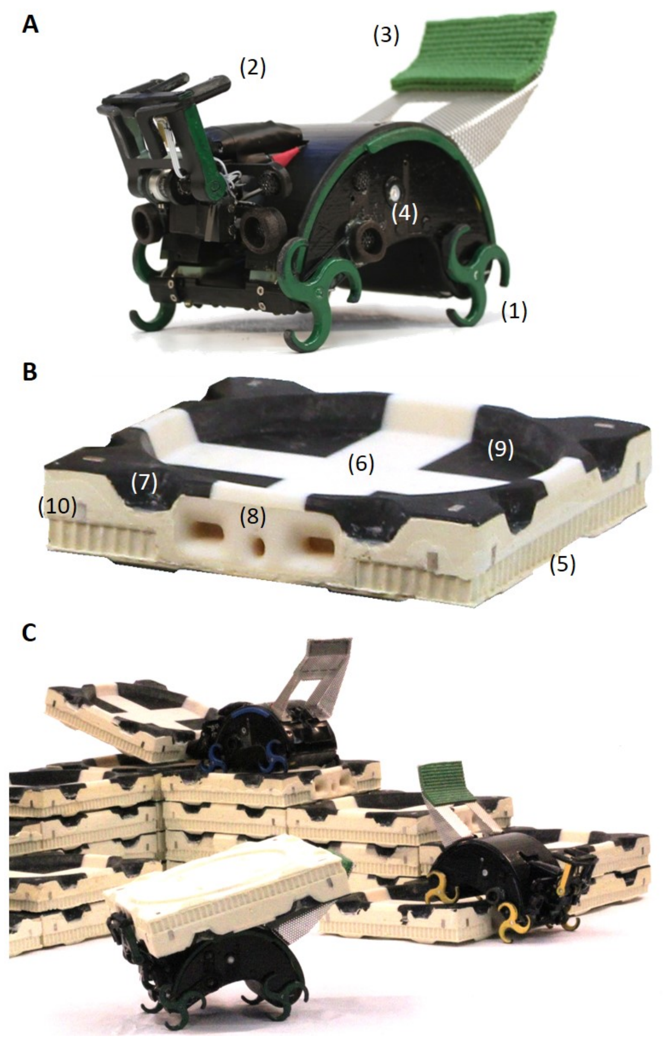

Fig. 1. TERMES robots (A) capable of collectively assembling specialized bricks (B), into user-specified structures in 2.5D (C).

IV. Expandable Bricks

To integrate seamlessly with the original robots and bricks, the expandable bricks must inherit all of the features discussed in Sec. III, i.e. overall geometry and weight, indented features for locomotion, handle for manipulation, and inverted features and magnets for assembly. We omit the rippled border as roof tiles will never be used on the ground. To remain practical for large-scale construction, the expandable bricks must also be relatively inexpensive, simple to fabricate, and reliable.

Our solution consists of two half bricks, held together by a spring hinge and a latch (Fig. 2.A). Each half has indented features and magnets for attachment; the majority of the handle is embedded in the lower half of the brick. The only modification that is required for these bricks to match with the original is a slight translation of the magnets on the planar surface and some small slots on the side to make room for the latch. The expandable bricks fit on top of original and other expanded bricks. Two expanded bricks cannot be placed directly on top of each other in opposite directions (prevented by protruding latch features), however, as such a configuration is replaceable by two normal bricks this issue can be circumvented in a future version of the compiler.

A. Trigger mechanism

The bricks rely on their own weight and magnet bonding strength to deploy the mechanism shown in Fig. 2.B. This consist of 1) a catch which is fixed by a screw to the top half of the brick, 2) a pair of compression springs which push a slider to lock the catch in place, 3) a trigger which is free to slide vertically upon brick placement, and 4) a rotator which translates the triggering force, which is normal to the bottom surface of the brick, into a horizontal movement to open the latch. The top part of the rotator connects with the slider using kite wire (not shown in the figure). Therefore, when the trigger is pushed upward, the rotator rotates clockwise, the slide is pulled into the housing because of the kite wire, and the catch is released. This causes the two halves to open by the rotational torque from torsion spring hinges. To limit harmful dynamic forces acting on the system, the torsion spring is chosen such that it reaches maximum torque upon 120^o rotation; leaving the weight of the upper half of the brick to cause full expansion. The fillet on the slider and catch allow

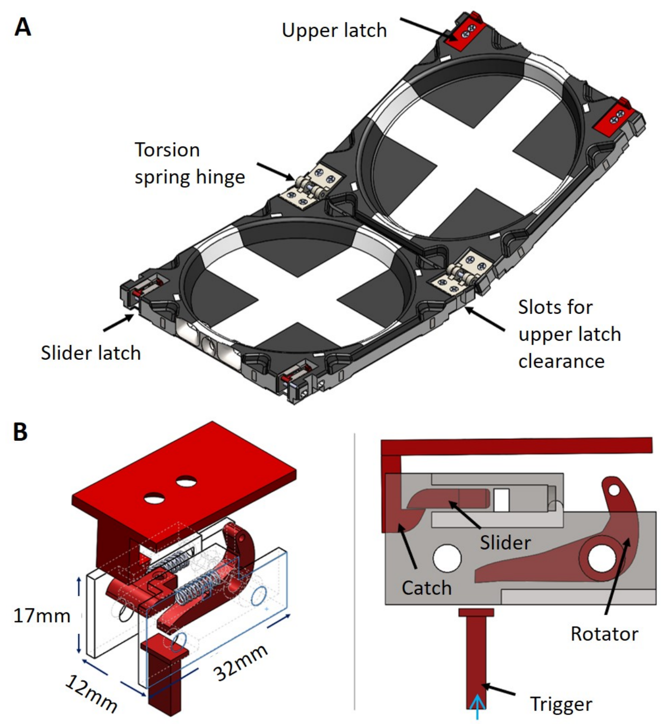

Fig. 2. Expandable brick (A) and trigger mechanism (B).

a press-lock, for an easy manual reset to the unexpanded state.

Because the trigger is designed such that only a vertical force can activate the expansion, the risk of accidental deployment during transport is reduced. We tested ten robot pick-ups, 90^o turns, and placements in a row, with no failures.

B. Fabrication

The expandable bricks consist of rigid urethane foam, 3D printed parts, magnets, and springs. The two halves are cast, similar to before, using a one-step molding process which directly embeds the magnets and handle into the foam (Fig. 3). Notice the ribbon placed underneath the magnets; this ribbon filtrates into the foam as it cures, and hold the magnets securely in place. In the old system, magnets would regularly pop out of the foam; we have yet to see this happen with the ribbon. The decrease in foam volume balances with the weight of the added components and the new bricks average a weight of 240g (similar to before). For ease of access, the latch and spring hinge are assembled separate from the brick, and fastened afterwards with screws. Although the bricks take longer to fabricate, ∼45min compared to 30min in the original bricks, the process can be sped up by molding several in parallel. The new handle, latch, and spring hinge were designed for fabrication in low tolerance PLA 3D printers, which significantly lowers their cost. Although we have doubled the number of magnets; the cost of the new bricks is only ∼$25, $5 more than before. Improvements in price and fabrication time are summarized in Table I.

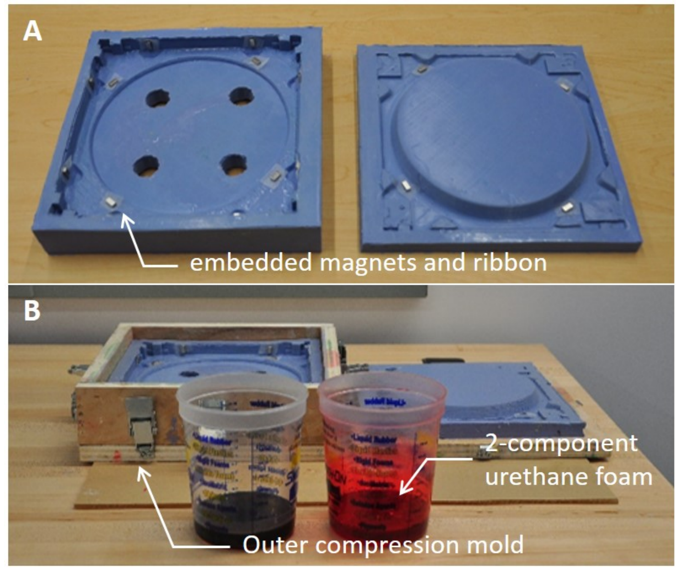

Fig. 3. One-step molding process of expandable bricks, using rubber molds (A) with embedded components and rigid urethane foam (B).

TABLE I
COMPARISON OF ORIGINAL AND NEW BRICKS AND ROBOT MECHANICS.

	Original Brick	**Expandable Brick**	**Original Robot Mechanics**	**New Robot Mechanics**
Price	$20	$25	$1,100	$50
Fab-time	∼30min	∼45min	∼75hrs	∼65hrs

V. ROBOTS

Of the $1,8K cost of a TERMES robot, $1,1K stems from passive mechanical components and chassis material. In a continued effort to decrease the cost, fabrication and maintenance time of these robots we therefore introduce a new mechanical design. Fig. 4.A shows the new robot, results are summarized in Table I. Specifically, we targeted a simplified gear train with less components prone to wear, and a lower tolerance chassis design printed with PLA in a low-end 3D printer (*Lulzbot Taz6*).

The old chassis (Fig. 1.A) consisted of a specialized hull with a lid, printed on an Objet 500 Connex 3D printer. The robot was restricted to fit with the indented bowl in the brick which, in turn, was a leftover from a previous robot design. This constraint led the driving motors to be located off-axis with (3:1) metal gears to transfer rotational torque onto two timing belts which drive the robot through differential steering. These were over-designed with metal pulleys, ball- and nylon bearings. In the new chassis design, removable hip joints (Fig. 4.B) serve several purposes: 1) they permit decreased hull width leading to larger whegs that climb better, 2) they allow motors to be located on axis eliminating the need for gears, and finally 3) they enable easy access to replace motors when they wear down. In the new design, we use motors of similar cost (*Pololu* micro metal gear motors), which are geared three times stronger than before. We further replace the timing pulleys with pulleys of a smaller pitch, which enables replacement by

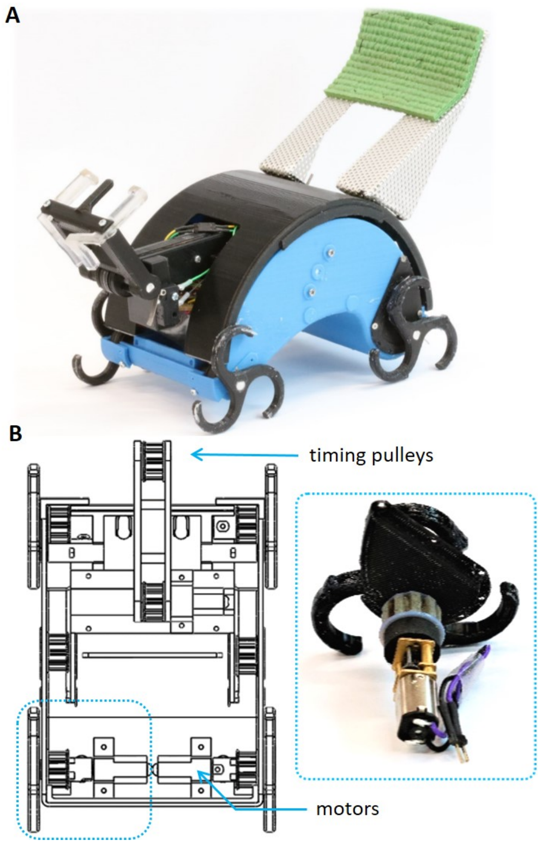

Fig. 4. New mechanical design of the TERMES robot, with low-end 3D printed chasses (A), and simplified gear train (B - top view).

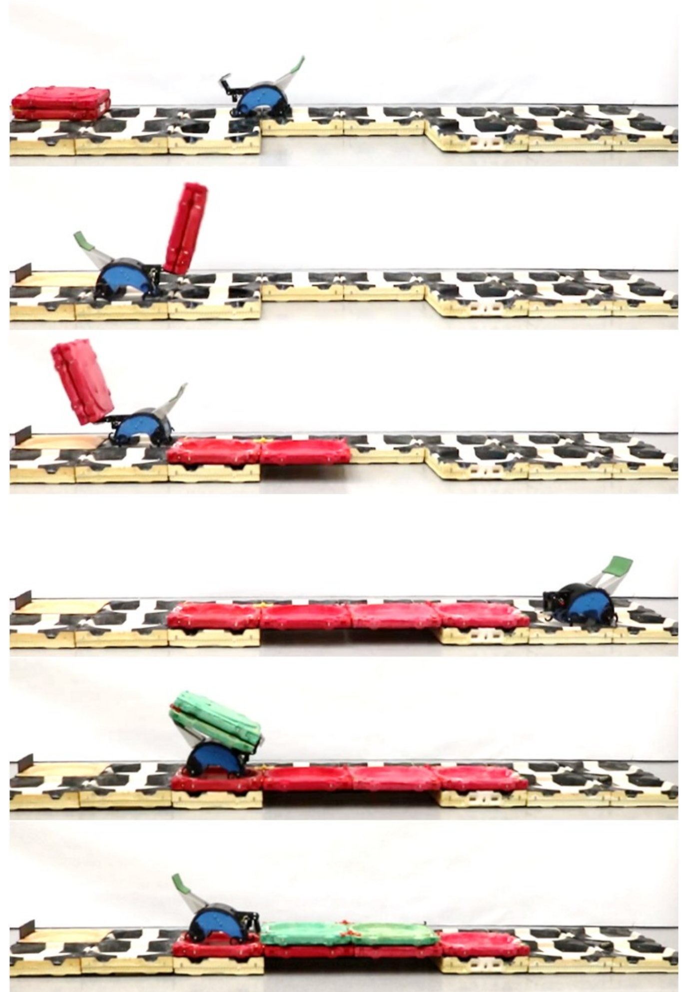

Fig. 5. Snapshots from a video of a remote controlled robot building a bridge on top of a platform of old bricks.

inexpensive 3D printed components. Finally, we increase the thickness of the hull material to permit better axle support, in turn allowing us to omit high accuracy bearings altogether.

Through these efforts we brought the cost of the mechanical components from \$1,1K to \$50. The weakest point in the drive train is the plastic attachment from motor to timing belt (Fig. 4.B, inset). This piece breaks after about 40 brick traversals; we intend to replace this with a metal pulley for a slight cost increase. The old robots took a high effort week to assemble and calibrate (∼75 hrs); the minimized design permits a more 'casual' work week (∼65 hrs).

We added a simple *Arduino Micro* platform with motor drivers and *Bluetooth*, and used a remote control to demonstrate the robot building a small bridge from three expandable bricks on top of regular bricks (Fig. 5 and accompanying video, available at http://cei.ece.cornell.edu/research-2/termes/).

VI. Algorithm

We present an algorithmic framework for collective robotic assembly of structures with pyramid-shaped roofs over convex gaps. This algorithm is closely related to the first TERMES framework, which permitted an arbitrary number of robots to build user-specified structures layer-by-layer [18]. By exploiting the extra functionality of the expandable bricks, this new class of structures does not require any change to the robot platform; we continue to rely on decentralized, locally sensing, unsynchronized, and non-communicating agents.

As before the roof-building algorithm consist of an off-line compiler and an on-board rule set. We present a blueprint of a 2.5D structure with a convex cavity to the compiler, if feasible it returns a maps of the structure in which travel directions over cavities are unlocked consecutively, and the robots then follow this map and add depositions according to their rule set.

A. Off-line Compiler

As presented in [5], the off-line compiler first converts the blueprint into a discrete map of brick depositions, then produces a directed graph between structure locations that designate one-way travel directions. Next, to add one or more roofs, the compiler finds internal and external contours as done in [19], and adds directed paths to the inner contour while complying with directions already assigned. The compiler further instantiates a new 'seed' location, S_n, for every horizontal roof layer. S_n is located in the corner of the cavity which is closer to the structure seed, S. In the opposite corner we place an 'exit' location, E_n. S_n and E_n are the first and last to be

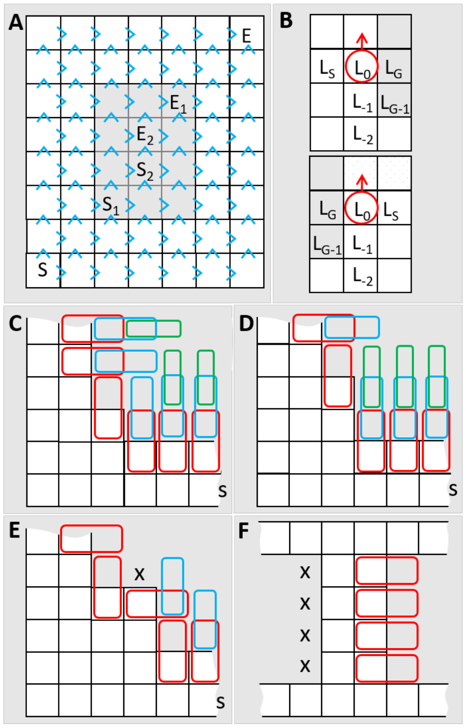

Fig. 6. A) Example of traffic directions generated by the off-line compiler. B) Example annotations to support Alg. 1; the red circle indicates position and orientation of the robot, L_G stands for the location of a gap as sensed by the robot, L_S is where the robot must stand to place a brick over L_0 and L_G. C-D) Examples of concave cavities that can be completed with the algorithms. E-F) Examples of structures that the current system cannot address; i.e. positions marked with an 'x' cannot be filled because $z_s \neq z_0$.

completed, respectively. The process continues until the entire gap has travel directions assigned (Fig. 6.A).

The current compiler automatically builds roofs over all internal cavities, this could modified by having the user specify which cavities should be covered, and which should not.

B. On-board Rule Set

Robots first complete the 2.5D structure using the rule set previously described in [5]. Briefly iterated, this rule set relies on the ability of the robots to sense their absolute location in the structure, $L_0 = \{x_0, y_0, z_0\}$, as well as the relative height of the 'parent' and 'child' sites that immediately lead to and from the last visited location, L_{-1}, respectively (Fig. 6.B). They then decide whether to add material at L_{-1}, if and only if $z_{-1} = z_0$, if the map specifies that a brick is required, and if it does not cause a gap in the structure or a height difference of more than 1 with respect to L_{-1}'s immediate neighbors. Following the compiled map, this prompts the structure to grow as propagating staircases starting from the seed where the brick cache is also located. When more options are available robots choose their path stochastically.

Next, each robot must infer when the 2.5D structure has finished – this can be done by having robots traversing all possible locations in the structure to check their completion; or, more efficiently, the original brick cache can be loaded with the exact number of bricks required to complete this part of the structure. When robots find that the 2.5D structure is complete, they switch to picking up bricks from an expandable brick cache, located next to the normal brick cache, and then unlock the access to the first horizontal layer of the inner contour. Each consecutive horizontal layer is unlocked only when the robots find that the exit brick in that layer has been deposited. The full algorithm is described in Alg. 1; intuitively, the robots traverse over the map, if they find an unfilled gap next to them with all parent sites completed, they move to the neighboring location opposite from the cavity if possible, turn around and add an expandable brick (Alg. 1.line 8). Snapshots from a run of the simulation over an 8 × 8 gap is shown in Fig. 7 and in the accompanying video, available at http://cei.ece.cornell.edu/research-2/termes/. Although we have yet to develop a formal proof, we expect the combination of compiler and rule set is guaranteed to complete because of its close relation to the previously proven set of algorithms.

Algorithm 1 Robot rule set for construction of roofs atop 2.5D buildings. Refer to Fig. 6.B for annotations related to the reference frame of the robot. We use Z and z to denote desired and current height of a location. 'Next' sites are children of L_0 with $|z - z_0| \leq 1$. S_n and E_n denotes seed and exit locations; n the current roof layer.

1: 2.5D structure is completed, n = 0.5.
2: *loop*:
3: move along structure perimeter, and enter at S
4: **if** not holding brick **then**
5:     get expandable brick from cache
6: **while** on structure **do**
7:     move to any 'next' site.
8:     **if** holding expandable brick
      and a gap detected ($z_G = 0$)
      and construction not done ($z_0 \neq Z_0$ **and** $z_G \neq Z_G$)
      and L_S is level with L_0 ($z_S = z_0$)
      and (all parents complete ($z_{G-1} > z_G$)
          or ($z_{G-1} = Z_{G-1}$)
          or at layer seed ($L_G = S_n$)) **then**
9:         move to L_S
10:         place expandable brick at L_0 and L_G
11:     **if** E_n completed ($z_{En} = n$) **then**
12:         unlock next layer ($n = n + 0.5$)
13:     **if** $L_0 = E$ **then**
14:         Leave structure

VII. Limitations and Future Work

Although our current approach shows how the set of structures that can be completed with the TERMES system can be vastly expanded by very simple measures, there are still several limitations to address.

An obvious limitation of the algorithm is that it only works for convex shapes. Our system will be able to patch small concave features (Fig. 6.C-D), however, with larger features the current approach fails (Fig. 6.E). The robots will similarly,

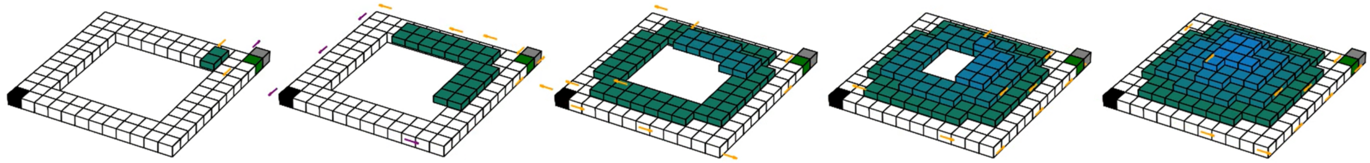

Fig. 7. Simulation of roof construction by ten robots illustrated as purple (unloaded) and yellow (loaded) arrows.

not be able to lay two roofs back to back (Fig. 6.F). If the structure is big enough, the concave issue can be mitigated by simply starting the roof further out than where the cavity starts. More work is also needed to explore if the algorithm can work if the gap perimeter is of different heights. In future work we will investigate automated synthesizers that suggest feasible designs to the user during the design phase.

The robot and material platform implemented in this paper is obviously only proof of concept. Currently, bricks do not attach well enough to permit robots to travel on them. Furthermore, the expandable bricks are added without concern for structural integrity. A long bridge, for example, will have a large moment acting on the anchoring brick. In the future, we wish to address these issues with stronger attachment mechanisms and reactive algorithms that iteratively assesses and adds roof material according to partial orderings inspired by traditional masonry and corbel structures.

As with the previous system, the practical limitation in scalability is the single point of structure entry. We are working on SLAM-based methods to permit robots to enter the structure at any point. Although we would like to further enable higher efficiency here, by letting robots initiate more roof layers simultaneously, this is complicated as it may lead to dead-ends from which robots cannot escape without breaking the traffic rules.

VIII. Conclusion

In this paper, we presented an extension to the original TERMES system, enabling an arbitrary number of TERMES robots to build structures such as roofs and bridges in 3D. We accomplished this, not by more advanced robots, but via addition of inexpensive expandable bricks and a slight modification of the algorithm. We further presented an improved mechanical design of the robot that significantly lowers the cost, as well as fabrication and maintenance time. Although construction systems most likely have much fewer robots than building elements, adding complexity to the robots may also make them significantly more prone to errors. It is worth balancing this need for error tolerance (especially in very large robot collectives where errors propagate in non-intuitive ways), with overall system cost.

Beyond the presence of overhangs being of importance to many types of structures, we view this as an exercise in how to achieve complex design by minimalistic robots. In the future, we expect that many of the same principles can be used to create quickly deployable, and potentially removable scaffolds to permit construction of an even wider range of structures.

References

[1] C. G. Kamel S. Saidi, Thomas Bock, "Robotics in Construction," in *Springer Handbook of Robotics*, S. I. Publishing, Ed., 2016, pp. 1493–1519.

[2] V. Gerling and S. von Mammen, "Robotics for Self-Organised Construction," in *Foundations and Applications of Self* Systems, IEEE International Workshops on*. IEEE, 2016, pp. 162–167.

[3] M. Hansell, *Built by animals: the natural history of animal architecture*. OUP Oxford, 2007.

[4] K. Petersen and R. Nagpal, "Complex Design by Simple Robots," *Architectural Design*, pp. 44–49, 2017.

[5] J. Werfel, K. Petersen, and R. Nagpal, "Designing collective behavior in a termite-inspired robot construction team." *Science*, vol. 343, no. 6172, pp. 754–8, 2014.

[6] J. Werfel and R. Nagpal, "Extended stigmergy in collective construction," *IEEE Intelligent Systems*, vol. 21, no. 2, pp. 20–28, 2006.

[7] J. Wawerla, G. Sukhatme, and M. Mataric, "Collective construction with multiple robots," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, no. October, pp. 2696–2701, 2002.

[8] M. Allwright, N. Bhalla, H. El-faham, A. Antoun, C. Pinciroli, and M. Dorigo, "SRoCS: Leveraging stigmergy on a multi-robot construction platform for unknown environments," in *International Conference on Swarm Intelligence*. Springer, 2014, pp. 158–169.

[9] K. H. Petersen, "Autonomous Construction of Temporary Human Habitats," Master of Science, thesis, University of Southern Denmark, 2008.

[10] S. Poppinga, C. Zollfrank, O. Prucker, J. Rühe, A. Menges, T. Cheng, and T. Speck, "Toward a new generation of smart biomimetic actuators for architecture," *Advanced Materials*, vol. 30, no. 19, p. 1703653, 2018.

[11] M. Yim, W.-M. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, and G. S. Chirikjian, "Modular self-reconfigurable robot systems [grand challenges of robotics]," *Robotics & Automation Magazine, IEEE*, vol. 14, no. 1, pp. 43–52, 2007.

[12] J. Worcester, R. Lakaemper, and M.-y. A. Hsieh, "3-dimensional tiling for distributed assembly by robot teams," in *Experimental Robotics*. Springer, 2013, pp. 143–154.

[13] S. Wismer, G. Hitz, M. Bonani, A. Gribovskiy, and S. Magnenat, "Autonomous construction of a roofed structure: Synthesizing planning and stigmergy on a mobile robot," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 5436–5437.

[14] V. Lindsey, Q., Mellinger, D., Kumar, "Construction of Cubic Structures with Quadrotor Teams," *Robotics: Science and Systems VII*, 2011.

[15] A. Braithwaite, T. Alhinai, M. Haas-Heger, E. McFarlane, and M. Kovač, "Tensile web construction and perching with nano aerial vehicles," in *Robotics Research*. Springer, 2018, pp. 71–88.

[16] B. Jenett and K. Cheung, "BILL-E: Robotic Platform for Locomotion and Manipulation of Lightweight Space Structures," in *25th AIAA/AHS Adaptive Structures Conference*, 2017.

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

[18] J. Werfel, K. Petersen, and R. Nagpal, "Distributed multi-robot algorithms for the TERMES 3D collective construction system," in *In Modular Robotics Workshop, IEEE Intl. Conference on Robots and Systems (IROS).*, 2011.

[19] J. Seo, S. Chae, J. Shim, D. Kim, C. Cheong, and T.-D. Han, "Fast contour-tracing algorithm based on a pixel-following method for image sensors," *Sensors*, vol. 16, no. 3, p. 353, 2016.