Construction of Cubic Structures with Quadrotor Teams

Quentin Lindsey, Daniel Mellinger, Vijay Kumar
Mechanical Engineering and Applied Mathematics
University of Pennsylvania
Philadelphia, PA 19104
Email: quentinl,dmel,kumar@seas.upenn.edu

Abstract— **We propose and investigate a system in which teams of quadrotor helicopters assemble 2.5-D structures from simple structural nodes and members equipped with magnets. The structures, called Special Cubic Structures (SCS), are a class of 2.5-D truss-like structures free of overhangs and holes. Grippers attached to the bottom of each quadrotor enable them to pick up, transport, and assemble the structural elements. The design of the nodes and members imposes constraints on assembly which are incorporated into the design of the algorithms used for assembly. We show that any SCS can be built using only the feasible assembly modes for individual structural elements and present simulation and experimental results with a team of quadrotors performing automated assembly. The paper includes a theoretical analysis of the SCS construction algorithm, the rationale for the design of the structural nodes, members and quadrotor gripper, a description of the quadrotor control methods for part pickup, transport and assembly, and an empirical analysis of system performance.**

I. Introduction

There is a small and growing class of applications in which robots are used in assembly and construction of structures [11]. These applications usually require extremely structured and expensive environments, which is typical in industrial automation used by automotive, electronics and packaging industries. This paradigm relies on a static environment in which absolute positions and orientations of parts and fixtures remain unchanged so that industrial robots can be programmed to pick, place and assemble with relatively simple and extremely reliable position and hybrid controllers with very little adaptation or planning [8].

There are many assembly/construction applications in which the environment is less structured, *e. g.* ship building or aircraft assembly industry. These applications require a fixed-position layout where resources are brought to the product. While robots may be involved in such tasks as making long continuous welds along the hull of the ships or in the manufacture of prefabricated sections, human workers are closely involved in operating machinery used for assembly or installation of components.

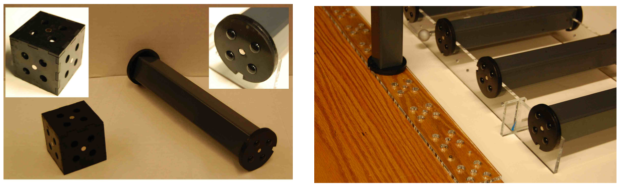
Rotorcraft are used in construction work, especially for aerial lifting or transport to hard-to-access sites including downtown skyscrapers, mountainous terrain and oil rigs, or tasks requiring assembly of tall towers. However, aerial vehicles are operated manually even though recent work [4], [9], [16] suggests that robotic helicopters can outperform even the most skilled human pilots in many applications.

In this paper we will explore the assembly of three-dimensional structures similar to those involved in construction of scaffolds, tower cranes, skyscrapers, and high-voltage towers using autonomous aerial robots. As in any manufacturing application, it is necessary to employ basic design for assembly principles [5] and to ensure that the parts must be matched to the robots and end-effectors that are assembling them. We consider part designs suitable for robotic assembly [7] and design truss-like elements that can easily snap together and end-effector that are suitable for aerial grasping, transport and assembly. We show that a team of aerial robots (quadrotors) are able to construct structures automatically from simple structural nodes and members using carefully-designed assembly modes compatible with the part design. The paper includes a theoretical analysis of the construction algorithm, a description of the quadrotor control methods for part pickup, transport and assembly, and an empirical analysis of system performance with multiple quadrotors.

While the work on assembly by aerial robots is quite limited, this paper builds on extensive robotics literature in the areas of robotic assembly [20], robotic grasping [18], autonomous helicopters [6], [21] and modular robotics [7].

In the most relevant work [7], the authors create a system in which stationary robotic manipulators are used to build truss-like 3-D structures from parts that are fed to them. Instead of climbing the structure to add additional parts, each 2-D level is automatically built and elevated to a higher level. A benefit of this is that the system can be transplanted to any environment without reconfiguration.

Recent work has examined using local rules and stigmergy to build 2-D structures with groups of robots. Robots in this work encircle the current structure placing blocks at locations determined exclusively from current state of the structure. With block capabilities, speed and robustness of construction can be increased while the capabilities of the robots can be reduced [23], [24]. Purely stochastic systems also use local rules derived from interaction rates to dictate how parts are received [17] or whether parts are assembled or disassembled [13]. While these systems offer robustness to failure due to environmental or mechanical causes, they only provide probabilistic guarantees, which are often unsatisfying.



(a) A node (left) and a member (right) with close up views (b) Part bins containing parts

Fig. 1. Parts used to construct Special Cubic Structures

Our work is related to the two bodies of work described above but differs in several important ways. First, our approach to construction is deterministic. The structure blueprint is automatically translated into a deterministic plan for assembly of simple cubic structures. Second, and perhaps more importantly, all the assembly is performed by aerial robots, which introduces constraints unique to the coordination of aerial robots, aerial grasping and assembly with low-complexity grippers.

II. Construction

A. *Part Geometry*

In this work the basic units of construction are nodes and members as shown in Fig. 1(a). The nodes and members used in this work were inspired by the work of [7]. Each node is a small cube that can attach to six members, each of which is a rectangular prism. A single node attached to a single member constitutes a module (Fig. 4(a)).

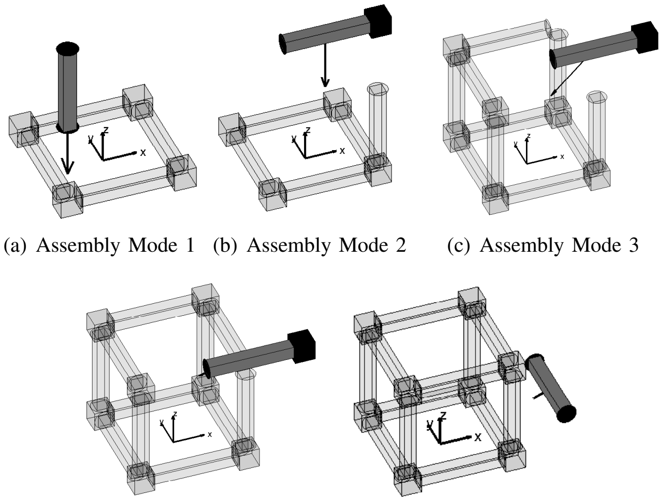
B. *Special Cubic Structures*

We consider 2.5-D tower-like structures consisting of strata of identical cubes with two constraints. First, we do not allow overhangs since this requires cantilevered members to support more weight than is possible at each joint. Second, because of assembly constraints, we further require that each layer of cubes of has no holes. We will call such structures Special Cubic Structures (SCS).

The members are placed onto the SCS as horizontal members (beams), vertical members (columns) or as modules using the five assembly modes shown in Fig. 2. Assembly mode 1 (M1) is used to place the vertical columns required for each layer. Once the columns are placed, assembly modes $2-5$ (M2-M5) are used to construct squares in a 2-D stratum as shown in Fig. 3. Due to the design of the hardware, some assembly modes are not possible. For example, it is not possible to assemble a horizontal member between two nodes that are already assembled to columns. The algorithm for constructing SCSs takes such constraints into account and is described next.

C. *Algorithm for constructing SCSs*

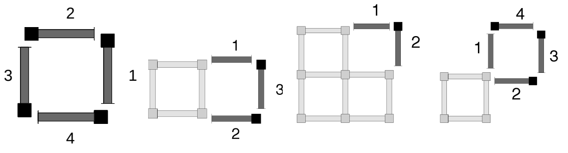
We now describe our *Wavefront Raster* (WFR) algorithm for building each 2-D stratum for any SCS. The algorithm is called the Wavefront Raster algorithm because we first mark all the squares that expand the *wavefront* by 1-hop and then build them in the order of a *raster* scan, from left to right and top to bottom as shown in the pseudocode in Algorithm 1.



(a) Assembly Mode 1 (b) Assembly Mode 2 (c) Assembly Mode 3

(d) Assembly Mode 4 (e) Assembly Mode 5

Fig. 2. The five assembly modes used to construct a stratum in a SCS.



(a) Assembly Primitive 1 (b) Assembly Primitive 2 (c) Assembly Primitive 3 (d) Assembly Primitive 4

Fig. 3. Assembly primitives used to complete squares in a partially-built stratum given that some squares have already been assembled and order of part assembly.

The Special Cubic Structure (SCS) construction algorithm couples the WFR algorithm with the necessary column placements to build any SCS.

D. *Algorithm Properties*

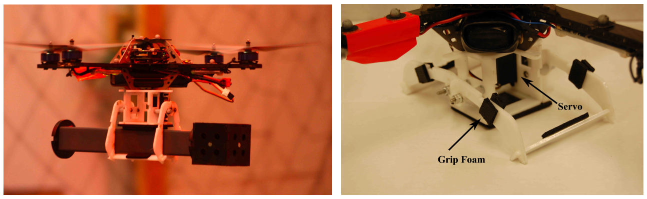
Here we point out several properties of the algorithms used for building structures. We note that the WFR algorithm builds and completes 2-D squares one at a time, which adds stability to the structure by limiting the number of partial squares to one with the stratum. This property also implies that parts are added in a serial manner. In general it is not possible to assemble parts at different points in the structure concurrently using these algorithms. However, if for any stratum (and for subsequent strata in the 2.5-D SCS), the structure has disconnected regions and the distance between them is such that the quadrotors can concurrently assemble parts without interference, the assembly plan can be split into separate

Algorithm 1 Wavefront Raster (WFR) Algorithm

```
build any square in the 2-D region
while not finished do
    mark squares immediately connected to already built region
    for (leftmost column) to (rightmost column) do
        build marked squares in column from bottom to top
```

Algorithm 2 Special Cubic Structure (SCS) Construction Algorithm

```
build first 2-D stratum with WFR algorithm
while not finished do
    place columns required for next 2-D stratum
    build all groups of connected regions in next 2-D stratum with WFR algorithm
```



(a) Quadrotor with Part (b) Gripper

Fig. 4. (a) A Hummingbird quadrotor carrying a module. (b) A single degree of freedom gripper made of acrylic actuated by a servo motor with a layer of foam to facilitate grasping.

feasible plans for each disconnected region.

Here we state some theoretical properties of the algorithms and show the WFR and SCS algorithm can build SCSs.

Lemma 1. *At any step of the WFR algorithm, the partially-constructed stratum is connected.*

Proof. Every placed square is a neighbor to an already placed square so the built region must always remain connected. □

Lemma 2. *The WFR algorithm for a SCS can be implemented using the four building primitives, P1-P4 in Fig. 3.*

Proof. See appendix. □

Lemma 3. *The WFR algorithm can build any* 2*-D connected region.*

Proof. Since the region is connected and finite, any unbuilt square is always n edges away (in the graph representation of the squares in a stratum) from an already built square and n is finite. So the unbuilt square will be built after n or less loops in Algorithm 1. This is true for all unbuilt squares. Additionally, Lemmas 2 guarantees that all squares can be built using the available assembly primitives. □

Theorem 1. *The SCS construction algorithm can realize any SCS.*

Proof. All columns can be placed with assembly mode 1 and all SCS strata can be build using the WFR algorithm. □

III. Experimental Infrastructure

A. Robots

We use the Hummingbird quadrotors sold by Ascending Technologies, GmbH [1] for experimental validation of the basic concepts. The quadrotor is approximately 55 cm in diameter, weighs approximately 500 g including battery, while providing approximately 20 minutes of operation with no payload. The maximum payload is around 500 grams.

Each robot is equipped with a gripper (see Fig 4(b)) specially designed for the parts used for the SCSs. The single-degree of freedom gripper consists of a pair of fingers driven by a simple slot mechanism powered by a Hitec HS-82MG servo, which has 2.8 kg-cm of torque and a mass of 19 g. The gripper is fabricated from acrylic and a layer of foam tape adds to the coefficient of friction for a more stable grasp.

B. Parts and bins

Stated in II-A, the basic units of construction in this work are nodes and members. Each face of a node has four circular slots with complementary protrusions at the two ends of each member to provide features for assembly. Magnets are embedded at the center of each face to allow for a snap fit connection. The key differences from [7] are a reduction in the number of magnets and the mass of the parts to allow for transport by quadrotors and for robust assembly. Each node has a mass of 60 g and each member has a mass of 119 g so that the largest payload (a module) has a mass of 179 g.

Parts are stored in bins before assembly as shown in Fig. 1(b). Columns are stored in a bin consisting of a horizontal plate designed with two rows of holes that accommodate a face of the column. In the center of each set of holes is a small magnet, which is strong enough to support the columns vertically in the downwash of the quadrotors but is weak enough that the quadrotor can easily lift the column from the bin. Horizontal members (or beams) and modules are stored with their longitudinal axis in a horizontal position in a different bin with evenly spaced notches. The notches are spaced far enough apart so that the gripper when fully opened will not interfere with another part. Further, they are elevated from the surface such that when a quadrotor lands upon the part, the gripper can close around the member.

C. State estimation

In our work we rely on the VICON motion tracking system [3] for state estimation for the aerial vehicles, and for estimating the position and orientation of the part bins and base of the desired SCS to be built. The VICON system provides position feedback at 150 Hz with marker position accuracy on the order of a millimeter. The workspace of the tracking system is 6.7 m × 4.4 m × 4.0m. Since it is impractical to place VICON markers on every part, each bin is designed to be a pallet and parts are stacked so that the position and orientation of a part with respect to the bin is known. While quadrotors use state feedback for assembling individual parts, there is no direct estimate of the state of the SCS. Since the initial position and orientation of the base of the SCS are known, the quadrotors keep track of the number and types of parts that have been assembled and are able to infer the state of the SCS at any point.

IV. Control for Robotic Assembly

This section will briefly describe the two levels of the control for each quadrotor. Low level controllers described in Sec. IV-A are used to execute three maneuvers. First, the quadrotor can hover at any specified position. Second, the quadrotor can execute a specified trajectory between any two desired points. Third, the quadrotor can apply open-loop yaw moments to test successful assembly of a part.

At a higher level, multiple quadrotors are coordinated using a finite state automaton to perform the assembly of a specified SCS efficiently and safely (Sec. IV-B) like in Fig. 5.

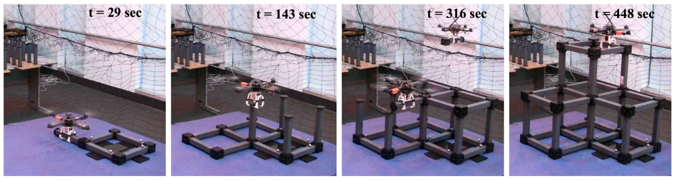


Fig. 5. Intermediate snapshots of a pyramid-like SCS being built by three quadrotors

A. Quadrotor Control

The quadrotor controller is illustrated in Fig. 6. Because the quadrotors operate at near hover conditions, we use controllers derived from the linearized equations of motion defined in [15] where the roll and pitch angles, ϕ and θ, are proportional to accelerations in x and y. An inner loop controls the attitude of the robot similar to the approach used in other work [6], [10], [12], [15]. An outer position control loop prescribes the desired roll and pitch angles required to achieve the desired accelerations.

Let $\mathbf{r}_T(t)$ and $\psi_T(t)$ be the trajectory and yaw angle we are trying to track. The command accelerations in the ith direction, $\ddot{r}_i^{des}$, are calculated from PID feedback of the position error, $r_{i,T} - r_i$. Note that the desired velocities and accelerations are given by $\dot{r}_{i,T} = \ddot{r}_{i,T} = 0$ for hover. Here the integral control terms constantly adapt to the changing mass and center of mass of the system due to the changing payload. The desired roll and pitch angles are then calculated from the first two components of the desired acceleration while $\psi^{des} = \psi_T$ is specified for each task.

The four desired motor speeds, ω_i^{des}, are calculated from four rotor speed differentials $(\Delta\omega_\theta, \Delta\omega_\phi, \Delta\omega_\psi, \Delta\omega_F)$ and the nominal rotor speed required to hover, ω_h, through a constant linear transformation:

$$\begin{bmatrix} \omega_1^{des} \\ \omega_2^{des} \\ \omega_3^{des} \\ \omega_4^{des} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 & 1 \\ 1 & 1 & 0 & -1 \\ 1 & 0 & 1 & 1 \\ 1 & -1 & 0 & -1 \end{bmatrix} \begin{bmatrix} \omega_h + \Delta\omega_F \\ \Delta\omega_\phi \\ \Delta\omega_\theta \\ \Delta\omega_\psi \end{bmatrix}. \tag{1}$$

The attitude control block generates motor speed differentials $(\Delta\omega_\theta, \Delta\omega_\phi, \Delta\omega_\psi)$ according to PD control on the Euler angles and the angular velocities. The fourth motor speed differential, $\Delta\omega_F$, is derived from the desired acceleration in the z-direction.

As in [15], the high-level position control loop runs on a control computer that receives the quadrotor pose estimates from VICON. Interprocess communication on the control computer is handled by ROS [19] and a ROS-MATLAB bridge [2]. The control computer sends inputs to the ARM7 processor on the quadrotors via ZIGBEE at a fixed rate of 100 Hz, which runs the low-level attitude control loop and computes the desired motor speeds.

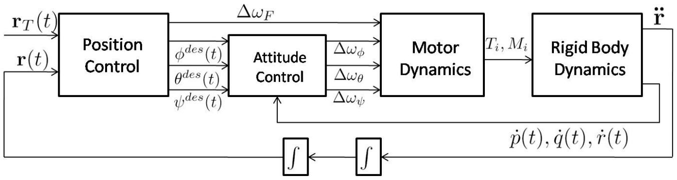


Fig. 6. Control Loops for position and attitude control

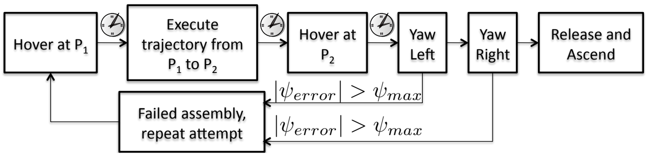

Hover at P1
Execute trajectory from P1 to P2
Hover at P2
Yaw Left
Yaw Right
Release and Ascend
$|\psi_{error}| > \psi_{max}$
Failed assembly, repeat attempt
$|\psi_{error}| > \psi_{max}$

Fig. 7. Composition of (a) hover controller; (b) trajectory controller; and (c) yaw controller for assembling a part to a partially-completed SCS.

B. Finite State Automaton

In this section we will describe the Finite State Automaton that coordinates the concurrent action of multiple quadrotors to enable multiple quadrotor experiments.

States in the FSA: We use a FSA with five states as shown in Fig. 8. We require that only one quadrotor is retrieving parts from the part bins and that only one quadrotor is assembling parts to the SCS. `Waiting_on_Bin` and `Waiting_for_Assembly` have FIFO (First In, First Out) queues, where the quadrotors hover in place until the part bins or the SCS become available. At the conclusion of the experiment, each quadrotor transitions to the `Finish` state.

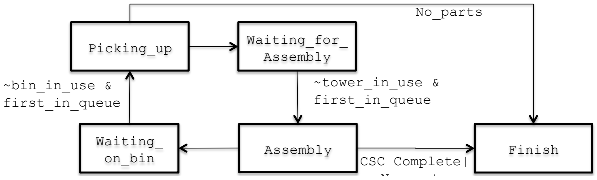


Fig. 8. The finite state automaton for picking up and assembling parts using multiple quadrotors.

To avoid collisions between the quadrotors, we design the layout for assembly accordingly. The part bins, the SCS, the hover positions for `Waiting_on_Bin` and `Waiting_for_Assembly` are located around the perimeter of a loop such that no two paths taken by the quadrotors are close to each other at any given time. Furthermore, we add delays between state transitions to ensure that a quadrotor serving the part bins or assembling the structure has sufficient time to leave the area before another quadrotor enters that same area.

Picking Up Parts: Columns are stored vertically in one bin while horizontal members and modules are stored horizontally in a different bin. To pick up columns, the quadrotor approaches and hovers in place above the specified column. It subsequently descends to a height such that when the grippers are closed, the face of the column is supported by the gripper from below. The quadrotor then ascends and hovers in place for a specified time interval. The integral term in the control loop allows the robot to adapt to the load. During this period the commanded thrust required to compensate for the load is used to determine if the quadrotor has grasped the part successfully. If the commanded thrust does not exceed the nominal thrust required for hover with the weight of the expected payload, the robot knows the grasp was unsuccessful and tries to grasp the part again. On repeated failures, the robot abandons that column and moves to the next available column.

Horizontal members and modules are grasped by having the fingers close around the longitudinal axis. The quadrotor hovers over the specified part and descends slowly until it is approximately 1 cm above the part. The quadrotor then cuts its thrust while controlling to a zero pitch and roll angle to "land" on the member. By landing on the member and grasping, we can ensure that the grippers will fully close around the part. The robot then increases thrust to ascend and hover in place. Again, adaptation through the integral control term allows the quadrotor to determine whether a part has been picked up successfully and try again if necessary.

Assembly: The strategy for placing parts on the structure is illustrated in Fig. 7. For each assembly mode, the quadrotor first hovers in place at a point P_1 (the location of each part in Fig. 2). It then descends in a straight-line trajectory to P_2, the desired position of the part on the structure, along the direction shown in Fig. 2. Then the quadrotor hovers in place at P_2 for a fixed time until the magnets snap into place. We have found that the slight fluctuations about the position setpoint inherent in the position control and the influence of the magnet are normally enough to cause the part to snap into place.

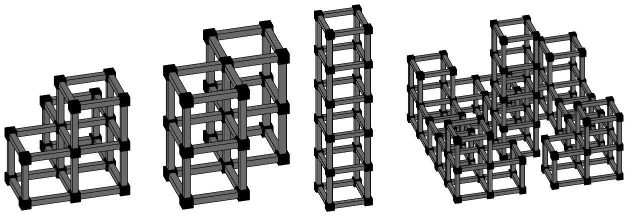
Although this technique is quite robust, we use a simple error detection method to determine if the assembly was successful. After the fixed time of hovering in place at P_2, the quadrotor executes an open-loop yaw moment. If the yaw angle, ψ, of quadrotor results in a large error, $\psi_{error} = |\psi_{des} - \psi| > \psi_{max}$, which is indicative of a unsuccessful assembly, the quadrotor ascends to P_1 to retry. If the error is small, it means the part has been fixed in place and the robot infers successful assembly and proceeds to the next task.

V. System Evaluation

A. Empirical Evaluation and Assessment

Here we present experimental results for six trials, two for each of three representative SCSs, in Fig. 9 and Table I. Snapshots of the assembly process for a pyramid-like SCS are shown in Fig. 5. Trials for a fourth structure (the castle) were investigated in simulation because of the limitation on the number of parts available and the battery life of the quadrotors.

Our simulation models the grasping, transport and assembly process. While attempts to assemble parts may fail in experiments, there are no failures in the assembly process in the simulation . The number of repeated assembly attempts (shown in Table I) explains the discrepancy between the time to completion in simulation versus experimentation. Throughout all experimental trials there were two unsuccessful assemblies where the quadrotor determined it had assembled a part correctly but the part was actually pinned in an incorrect position.



(a) Pyramid (b) Wall (c) Tower (d) Castle

Fig. 9. Representative Special Cubic Structures (SCS) tested in simulation and experiments.

	Pyramid	Wall	Tower	Castle
Number of Parts	32	34	40	192
Successful Assemblies (Trial 1/2)	32,32	33,34	40,39	
Time (Trial 1/2)	449.6 450.7	486.6 486.2	588.2 587.3	
Column retries (Trial 1/2)	5,5	3,1	8,3	
Beam retries (Trial 1/2)	4,5	2,2	5,1	
Time (in simulation)	443.6	480.4	581.9	2642.0

TABLE I

Time to completion (in seconds) and success rate for SCS in Figure 9 for two experimental trials and in simulation.

B. Process Variation and Assembly Tolerance

In robotic assembly, it is necessary to ensure that the process variation (errors in position and orientation of each part and the position of the partially assembled structure) is smaller than the admissible tolerance required for pick up or for assembly. We empirically evaluated the allowed process variation for our assembly process with several trials for relative positioning errors of up to ±5 cm and orientation errors of up to 30°. Figure 10 shows the probability of successful assembly of a part as a function of the error in the position and orientation of the structure during the five assembly modes depicted in Figure 2. It can readily be seen that if the position errors are within ±3 cm in the horizontal plane, ±1 cm in the vertical plane, and ±5 deg in orientation, the probability of successful assembly is almost 1.0 in all assembly modes. Larger position and orientation error can be accommodated by repeated attempts as shown in Figure 7.

A similar set of experiments showed that the system is even more robust in picking up parts as shown in Table II. Near 100% success rates are achieved for much larger position and orientation errors than for assembly. Note that since grasping columns is invariant to rotations about the yaw axis, that entry in the column is missing.

C. Effect of Number of Quadrotors

For the experimental setup described in Section IV, increasing the number of quadrotors quickly leads to diminishing returns. Figure 11 shows the completion time and distribution

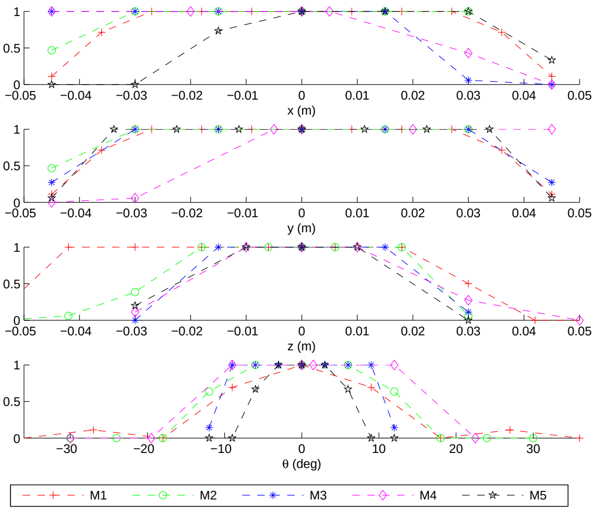


Fig. 10. Empirical evaluation of the success rates of placing parts for each assembly mode depicted in Fig. 2

Type	x(cm)	y(cm)	z(cm)	θ(deg)
Vertical	±4.23	±3.81	$[-2.54, 2.03]$	—
Horizontal	±2.54	±4.23	—	±25

TABLE II

EMPIRICAL EVALUATION OF THE NEAR 100% SUCCESS RATE FOR VERTICAL AND HORIZONTAL PARTS. FOR HORIZONTAL PARTS, THE y-AXIS IS ALONG THE LENGTH OF THE PART.

of time spent in different states for a representative SCS consisting of two horizontal cubes requiring 16 part placements. For experiments with 3 or more quadrotors, the completion time remains a constant. This behavior is a direct consequence of our decision to have only one robot serve the part bins and one robot perform assembly on the structure at a given time.

VI. DISCUSSION

Main contribution of the paper: The construction algorithm and control schemes outlined in Secs. II and IV have yielded a reliable system that can construct any Special Cubic Structure (SCS). This work is unique because it represents, to our knowledge, the first autonomous aerial robot construction system, and demonstrates a proof-of-concept system that can correctly and reliably construct a SCS for the construction of towers, scaffolds, trusses and commercial buildings. Of course, a number of issues will arise when we scale up to build real-world structures which we address below.

Localization without motion capture systems: Although we depend extensively on VICON, we have demonstrated that this system can perform in less than ideal situations with errors in positioning and orienting. In Fig 10 and Table II, we showed the effect of position and orientation errors on part assembly and part pickup. Clearly our present system guarantees errors that are smaller than the maximum allowable errors. With the current state of the art laser and vision based onboard localization systems boasting accuracies of better than 3 cm in position and 1° in yaw [21], it is possible to perform the individual tasks discussed here without the motion capture system.

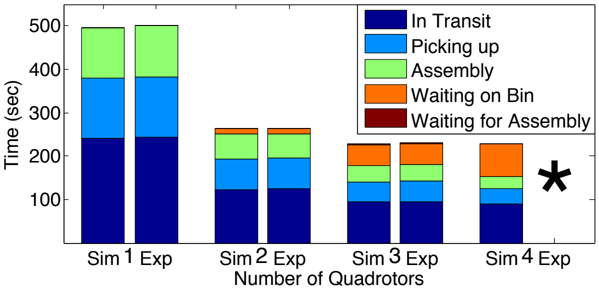


Fig. 11. Time spent on different states (Fig. 8) during the construction process.

Power: The scale of the structures that we can currently build is mainly limited by battery life. While an unloaded quadrotor hovering in place can stay aloft for approximately 20 mins, a quadrotor loaded intermittently with the parts has a maximum flight time of 10 mins. To mitigate this effect, the batteries would need to be changed during the experiment. Certainly, this can be solved by integrating charging stations such as those described in [22] so that the longer assembly tasks can be completed, but at the cost of increased system complexity.

Parallelization of assembly tasks: The current system implements assembly plans in a serial fashion. From Fig. 11, we can see that the benefit of multiple quadrotors quickly saturates. The algorithms presented here ensure that the SCS can be built using primitives that can be implemented with our subset of feasible assembly modes. In order to realize additional benefits with multiple quadrotors, we must solve two problems. First, we must develop new strategies of creating feasible assembly plans that can place parts without introducing intermediate structures that result in a deadlock condition (*i. e.*, without resulting in states that preclude further assembly). Second, the system must allow several quadrotors to retrieve parts simultaneously from multiple supply bins and would need to incorporate more sophisticated path planners to resolve conflicts and to avoid collisions. Both these problems are future research directions.

Heterogeneity: In this work we were limited with the constraints of an aerial robotic team. By integrating ground robots into this system, we can potentially build subassemblies on the ground and have quadrotors cooperatively lift and place them in a manner similar to the cooperative lifting described in [14].

Part design and fasteners: While the snap-fit parts improve the ease and reliability of assembly, real structures will require stronger joints. This may be done using smarter parts with built in fasteners that include anchor screws and toggle bolts that are automatically deployed on assembly or by attaching fasteners manually, as is currently done in automated building construction. Smarter parts can also be used to facilitate better fault detection by recognizing their neighbors and the state of the partially built structure and communicating with the robots during the assembly process.

Summary: In conclusion, construction of structures with aerial robots represents an exciting area of research with many potential applications. This paper represents the first step in this direction with a proof-of-concept, provably correct approach to assembling Special Cubic Structures from parts with embedded magnets. As our discussion above illustrates, there are many ways to extend this work with a plethora of new research challenges for the robotics community.

Acknowledgments

The authors would like to acknowledge Yash Mulgaonkar for help assembling the parts.

References

[1] Ascending Technologies, GmbH. http://www.asctec.de.

[2] ROS-Matlab Bridge. http://github.com/nmichael/ipc-bridge.

[3] Vicon Motion Systems, Inc. http://www.vicon.com.

[4] M. Bernard and K. Kondak. Generic slung load transportation system using small size helicopters. In *In Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3258–3264, 2009.

[5] G. Boothroyd and W. Knight. Design for assembly. *Spectrum, IEEE*, 30(9):53 –55, September 1993.

[6] S. Bouabdallah. *Design and Control of Quadrotors with Applications to Autonomous Flying*. PhD thesis, Ecole Polytechnique Federale de Lausanne, Lausanne, Switzerland, February 2007.

[7] K.C. Galloway, R. Jois, and M. Yim. Factory floor: A robotically reconfigurable construction platform. pages 2467 –2472, may. 2010.

[8] Mikell P. Groover. *Automation, Production Systems, and Computer-Integrated Manufacturing*. Prentice Hall Press, Upper Saddle River, NJ, USA, 3rd edition, 2007.

[9] J. F. Henderson, J. Potjewyd, and B. Ireland. The dynamics of an airborne towed target system with active control. In *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, volume 213, 1999.

[10] Gabriel M. Hoffmann, Haomiao Huang, Steven L. Wasl, and Er Claire J. Tomlin. Quadrotor helicopter flight dynamics and control: Theory and experiment. In *In Proc. of the AIAA Guidance, Navigation, and Control Conference*, 2007.

[11] Hanjong Joo, ChiSu Son, Kyunghun Kim, Kyunghwan Kim, and Jaejun Kim. A study on the advantages on high-rise building construction which the application of construction robots take (iccas 2007). In *Control, Automation and Systems, 2007. ICCAS '07. International Conference on*, pages 1933 –1936, 2007.

[12] S. Lupashin, A. Schollig, M. Sherback, and R. D'Andrea. A simple learning strategy for high-speed quadrocopter multi-flips. pages 1642–1648, Anchorage, AK, May 2010.

[13] Loic Matthey, Spring Berman, and Vijay Kumar. Stochastic strategies for a swarm robotic assembly system. In *ICRA*, pages 1953–1958, 2009.

[14] Daniel Mellinger, Michael Shomin, Nathan Michael, and Vijay Kumar. Cooperative grasping and transport using multiple quadrotors. In *International Symposium on Distributed Autonomous Systems*, Lusanne, Switzerland, November 2010.

[15] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar. The GRASP multiple micro UAV testbed. *IEEE Robotics and Automation Magazine*, 17(3):56 –65, 2010.

[16] Nathan Michael, Jonathan Fink, and Vijay Kumar. Cooperative manipulation and transportation with aerial robots. *Autonomous Robot*, 30:73–86, 2011.

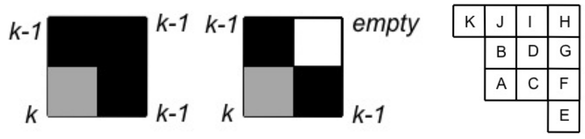
[17] N. Napp and E. Klavins. Robust by composition: Programs for multi-robot systems. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2459 –2466, May 2010.

[18] Paul Pounds and Aaron Dollar. Hovering stability of helicopters with elastic constraints. In *ASME Dynamic Systems and Control Conference*, 2010.

[19] Morgan Quigley, Ken Conley, Brian P. Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*, 2009.

[20] Arthur C. Sanderson, Luiz Homem de Mello, and Hui Zhang. Assembly sequence planning. *AI Magazine*, 11(1):62–81, 1990.

[21] S. Shen, N. Michael, and V. Kumar. Autonomous multi-floor indoor navigation with a computationally constrained MAV. In *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, Shanghai, China, May 2011. To Appear.



(a) (b)

Fig. 12. (a) Inward Pointed L-shape (left) and Inward Pointless L-shape (right) and (b) square placement reference figure.

[22] Mario Valenti, Daniel Dale, Jonathan P. How, Daniela Pucci De Farias, and John Vian. Mission health management for 24/7 persistent surveillance operations. *American Institute of Aeronautics and Astronautics*, 2007.

[23] Justin Werfel and Yaneer Bar-yam. Distributed construction by mobile robots with enhanced building blocks. In *International Conference on Robotics and Automation*. IEEE Press, 2006.

[24] Justin Werfel, Donal INgber, and Radhika Nagpal. Collective construction of environmentally-adaptive structures. In *International Conference on Intelligent Robots and Systems*, 2007.

Appendix: Proof of Lemma 2

In order to prove Lemma 2 we will consider constructing an individual square during the WFR algorithm in space 0 during wave k as shown in Fig. 14(a). There are 8 immediate neighbors of space 0 so there are 256 possible situations that can occur. We will show that many of the 256 possible situations are impossible and that the only ones remaining can be constructed using the construction primitives P1-P4 shown in Fig. 3.

We introduce *Inward L-shapes* as illustrated in Fig. 12(a) and define them with the aid of Fig. 12(b). We define an *Inward Pointed L-shape* as a situation where square A was built during wave k and squares B, C, and D were built during wave $k-1$. We define an *Inward Pointless L-shape* as a situation where square A was built during wave k and squares B and C were built during wave $k-1$ and space D is to be left empty. Note that *Inward L-shapes* can be rotated versions (90°, 180°, and 270°) of the ones shown in Fig. 12(a). We first show that *Inward L-shapes* imply locally disconnected regions of squares during a previous construction wave.

Lemma 4. *For SCSs the Inward Pointless L-shape shown in Fig. 12(a) implies that spaces E, F, G, I, J, and K must contain a set of disconnected squares built during wave $k-2$.*

Proof. No squares were built in E, F, G, I, J, or K prior to wave $k-2$ because then B or C would have been built before wave $k-1$. The fact that square B (C) was built during wave $k-1$ and A during wave k implies that one or more squares were built in spaces I, J, and K (E, F, and G) during wave $k-2$. Spaces G and I cannot both contain squares because this would imply a hole in the original structure at D. Therefore, E, F, G, I, J, and K must contain a set of disconnected squares built during wave $k-2$. □

Lemma 5. *For SCSs the Inward Pointed L-shape shown in Fig. 12(a) implies that for some wave number w there exists a set of squares of the shape of the region E-K that are disconnected and filled during wave w.*

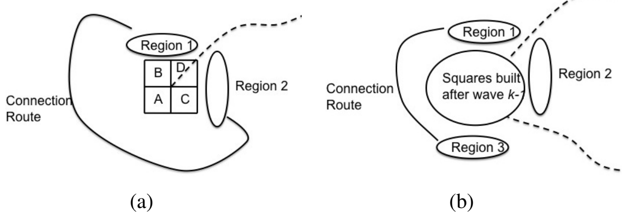


Fig. 13. (a) Region 1 and 2 cannot connect through some curve illustrated by the dotted line. (b) Region 2 is disconnected from both Region 1 and 3.

Proof. We can use the same logic as in the proof of Lemma 4 to show that if G and I are not both occupied then E-K are disconnected and $w = k - 2$. If G and I are occupied and H is not then Lemma 4 tells us that during wave $k-3$ we must have a set of nodes shaped like E-K that are disconnected, meaning $w = k - 3$. The only other possibility is that G, H, and I are all filled. In this situation we again have an *Inward Pointed L-shape*. So, if we repeat the logic we eventually result in a disconnected region shaped like E-K or a region shaped like G-I with all squares filled. However, we cannot continue to have three squares built at every wave because the WFR algorithm must start at wave 1 with a single built square. Therefore, for some wave number, w, we must have a region shaped like $E - K$ that is disconnected. □

We now show that *Inward L-shapes* imply globally disconnected regions.

Lemma 6. *For SCSs the WFR algorithm cannot result in the Inward Pointless L-shape.*

Proof. From Lemma 4 we know that spaces E, F, G, I, J, K contain two disconnected regions. As shown in Fig 13(a), we can draw a curve outward from square A that must separate the two regions of squares (Region 1 and 2). If they are completely disconnected this contradicts Lemma 1 so they must be connected in some way along the *Connection Route*. The Connection Route implies at least three rotated *Inward L-shapes* must exist. However, theses *Inward L-shapes* must also eventually result in locally disconnected regions like that shown in Fig. 13(a). Consider any one of the *Inward L-shapes*, it must eventually separate Regions 2 and 3 as shown in Fig. 13(b). Region 2 is then completely disconnected from region 1 and 3 which contradicts Lemma 1. Therefore, the WFR algorithm cannot result in an *Inward Pointless L-shape*. □

Lemma 7. *For SCSs the WFR algorithm cannot result in the Inward Pointed L-shape.*

Proof. According to Lemma 5 we must eventually result in two disconnected regions in a shape like E-K. Then the same logic used to prove Lemma 6 can be used to prove this Lemma. □

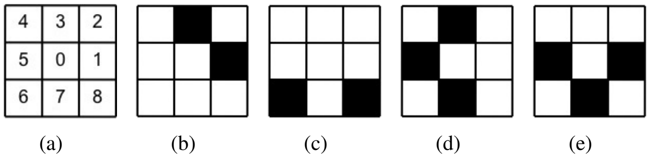


Fig. 14. Neighbor reference figure (a) and examples of situations that cannot occur (b-e)

Now we consider building a square in space 0 during wave k as shown in Fig. 14(a) and the 256 possible situations that can occur. Here we note that any of the spaces 4, 5, 6, and 7 that are filled at the time of construction may have been filled during wave k or $k-1$ and any of them not filled at this point will never be filled. Blanks in spaces 1, 2, 3, or 8 may or may not be filled during wave k. Any squares built in spaces 1, 2, 3, or 8 must have been built during wave $k - 1$.

Lemma 8. *At no point during the WFR algorithm for a SCS can a hole exist in the constructed structure.*

Proof. Since the original structure does not contain holes, any hole in the structure must contain spaces that are to be filled. Any hole that is to be filled must contain a *Inward L-shape* in the top right corner. Since *Inward L-shapes* are impossible the WFR algorithm cannot result in a structure with holes. □

We can now eliminate all of the impossible situations and prove Lemma 2.

Proof of Lemma 2: We first eliminate any situation that contains squares at both spaces 1 and 3 like shown in Fig. 14(b). These squares must have been filled during wave $k-1$ meaning these situations are *Inward L-shapes* which are impossible according to Lemmas 6 and 7. This eliminates 64 situations.

We next consider situations where prior to construction we have (at least) two disconnected regions in spaces $1-8$ like the example shown in Figure 14(c). These may be connected outside of the spaces $1 - 8$ but if this is true then building square 0 results in a hole in the structure. This contradicts Lemma 8 and we can dismiss 115 more situations as impossible.

The next situation to consider is one with squares built in spaces 3, 5, and 7 and not at 1, see Fig. 14(d) as an example. Square 3 must have been built during wave $k - 1$. If Square 5 was built during wave $k - 1$ then we would have an *Inward L-shape* so it must have been built during wave k. Square 7 must have been built during wave k or wave $k - 1$. If it was built at wave k then one or more of the spaces directly below 6, 7, and 8 must have been filled during wave $k - 1$. Either way square 3 must be connected to squares at the bottom. If they are connected around space 1 then the structure contains a hole, which is impossible. If they are connected around space 5 at least two *Inward L-shapes* are required which is impossible due to Lemmas 6 and 7. We can then dismiss 16 more situations.

The final situation to consider is one with squares in spaces 1, 5, and 7 and not in 3, see Fig. 14(e) as an example. The same logic used above can be used to show that blocks on the left must somehow be connected to blocks on the right which implies that this situation is impossible. We can then dismiss 16 more situations.

The remaining 45 situations can all be built using the constructions primitives P1-P4. □