The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

A Cuboctahedron Module for a Reconfigurable Robot

Shuguang Li *Student Member, IEEE*, Jianping Yuan, Franz Nigl, and Hod Lipson *Member, IEEE*

Abstract—**We present a concept for a modular robot with a quasi-regular polyhedron based on a cuboctahedron element. Lattice-type modular robots can adapt their morphology by reconfiguring to various shapes. While regular polyhedrons provide the bases of many promising 3D lattice elements, few modular robots have shapes with more than six regular faces. The conceptual design and prototypes of cuboctahedron elements are presented in this paper. To account for the various connecting configurations between robotic modules, we propose a directed graph with three parameters to represent the morphology of such a modular robotic system.**

I. Introduction

MODULAR robotic systems are comprised of a set of simple building blocks that are connected by physical interfaces or docking ports. The optimal way to connect these modules depends on the specific task and the possible topologies. The design of the individual modules plays a critical part in determining the performance of the entire system: By replacing, adding, and removing modules, the modular robotic system can change its morphology to adapt to the new tasks and environment [1]-[3].

One of key challenges in modular robotics research is to design robotic modules with a large variety of reconfiguration capabilities, while maintaining simplicity at same time. Much of the current work in this field is focused on homogeneous systems, in which all the modules are identical, and the system is either of chain-type or lattice-type [1][4]. A lattice-type robot usually has more available connecting faces than chain-type robots and can create more complex morphologies. Many researchers have developed cube or sphere-like 3-dimensional robots with one degree of rotational freedom [5][6]. The more regular connecting faces the robotic module has, the more topological possibilities the whole system will have, and therefore regular polyhedrons are promising candidates for the module's shape. Due to fabrication difficulties and geometrical complexities, however, few modules have more than six connecting faces.

This work was supported in part by the U.S. National Science Foundation's Office of Emerging Frontiers in Research and Innovation, grant #0735953 and the Chinese Scholarship Council.

Shuguang Li is a Ph.D. student of Northwestern Polytechnical University, Xi'an, Shaanxi 710072 P.R. China and a visiting student at Cornell University, Ithaca, NY 14850 USA (email: SL968@cornell.edu)

Jianping Yuan is professor with Northwestern Polytechnical University, Xi'an, Shaanxi 710072 P.R. China (e-mail: jyuan@nwpu.edu.cn).

Franz Nigl is researcher with the Cornell Computational Synthesis Laboratory, Cornell University, Ithaca, NY 14853, USA (e-mail: fn44@cornell.edu).

Hod Lipson is associate professor with the Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 14853, USA (e-mail: hod.lipson@cornell.edu).

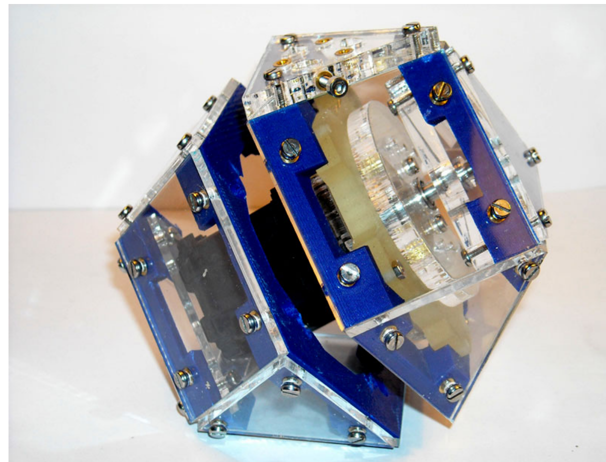


Fig. 1. The cuboctahedron module

In this study, we propose quasi-regular polyhedrons, called a cuboctahedron, as the basic shape for a robotic module. This polyhedron has 14 regular faces, and each module has one rotational degree freedom (DOF) as shown in Fig.1. Such a cuboctahedron module could significantly diversify the morphologies of a reconfigurable system, albeit with the addition of increased complexity in representation and planning.

A good methodology for describing the morphology of connected robots should be efficient in terms of configuration recognition, reconfiguration planning, and behavior design. It should also provide opportunities for applying intelligent search algorithms to solve various planning problems. In this paper, we also show a directed graph with three parameters to represent the modular robotic system's morphologies.

This paper is organized as follows: Section II reviews the related work on the configurations of modular robots and their representation methodologies. The conceptual design and the prototype of the cuboctahedron robot are presented in Section III, and details on the geometrical properties and the graph based morphological representation method are described in Section IV. Finally, we outline a discussion with the conclusion and future work in Section V.

II. Background

A. Chain-type and Lattice-type Modular Robots

Many modular reconfigurable robotic systems consist of homogeneous unit modules. According to geometrical arrangement of their units, these systems can be easily divided into two basic categories: chain-type and lattice-type systems.

All modules of chain-type systems are always connected together as a string or tree topology, therefore, the general architecture of the chain or tree is serial. This type has been the primary design choice in early modular robotic systems like Polybot [7] and Conro [8].

Lattice-type systems, on the other hand, can be built up by connecting several modules in a 2- or 3-dimensional space with various topologies. The potential configurations in these systems are determined by the units' geometrical properties and their connectors. The cube and sphere are the most popular lattices in such systems, for example I-Cube [9], Telecube [10], Crystal [11], Molecule [12] and Slimebot [13].

Usually, lattice-type systems have more potential connectors on each module than chain-type system, thus they can explore more configurations using the same number of units. Recently, some modular robotic systems were designed with the abilities to show both chain-type and lattice-type arrangements. These systems are hybrid-type systems, such as Atron [6], Molecubes [5], M-Tran [14], Superbot [15] and Roomboot [16].

B. Morphological Representation of Modular Robotic System

To describe a system composed of modules, previous studies have proposed different solutions of both direct and indirect representations.

The simplest way to represent a morphology or configuration is to use a group of parameters. Chocron and Bidaud introduced a method for task based design of modular robotic systems using Genetic Algorithms (GA); they used a single binary string to encode the manipulator's topology and configuration, such as the relative orientation of joint axis, joint type and link length [17]. The Snakebot was proposed with a simultaneous use of a GA for optimizing the sensor morphology and genetic programming (GP) for developing the motion patterns. The morphological parameters are encoded as a linear chromosome [18]. Zykov *et al.* employed a series of code pairs with variable length to represent the Molecube's morphology in the evolutionary search process for autonomous self-reproduction [5].

For more complex situations, the configuration matrix has been used extensively. Chen and Burdick represented a modular robot as an assembly incidence matrix and applied a GA to optimize the configuration for specific tasks [19]. Similarly, the incidence matrix has been used for describing the modules and their interactions in the evolutionary design of a locomotion system [20]. Park *et al.* introduced the adjacency matrix for configuration recognition on a modular robotic system [21]. For the heterogeneous Odin robot [27], an interconnection matrix, connector matrix and link matrix were built to present the physical structure of the robot and the module functionalities in the configuration.

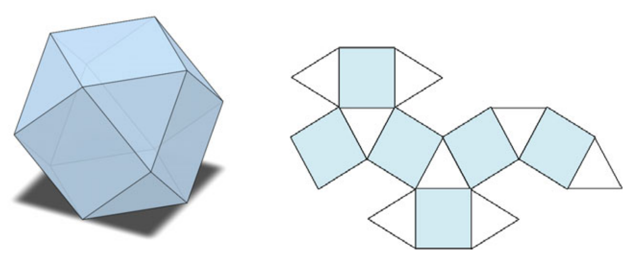
Graph based methods tend to be the most effective solutions to describe large and complex morphologies in a direct way. Sims used a directed graph representation to describe the creatures in simulation, nodes in the graph are represented body segments, and neural controllers are also embedded within these nodes [22]. Funes and Pollack implemented tree representations of 2D and 3D Lego structures, where each node on the tree represents a brick and has a group of parameters to indicate the bricks and their descendants. Based on this representation, a GP method was used to evolve the target configuration [23]. A dynamic and distributed reconfiguration planning algorithm for chain-type self-reconfigurable robots was developed by Hou and Shen, they used a modular graph to represent the SuperBot's configuration, where each node is a module and the edges indicate the physical connection between modules [24].

In terms of indirect methods, the grammar-like method can provide a compact representation for complicated and repeated structures, thus it can be used explore complex morphologies with easy grammars. Hornby and Pollack described a method that could evolve the morphology and neural controller of a three-dimensional robot, which uses Lindenmayer systems (L-systems) as the generative representation for the robots [25]. The artificial ontogeny was proposed by Bongard and Pfeifer [26].The agent is represented by genetic regulatory networks, and it can evolve in simulated ontogenetic process.

III. Cuboctahedron Robot

A. Geometry of Cuboctahedron

The cuboctahedron (Fig.2a), a quasi-regular polyhedron, belongs to the Archimedean solids. It is made from six equilateral square faces and eight equilateral triangular faces; two triangles and two squares meet at one vertex. One cuboctahedron has 12 such identical vertices in total and 24 identical edges where each triangle abuts a square. The name cuboctahedron comes from "cube" and "octahedron," and a cuboctahedron can be built by slicing suitable sections off the vertices of either a cube or an octahedron[28][29]. The main geometrical properties of the cuboctahedron are shown in Table I.



(a) (b)

Fig. 2. A cuboctahedron solid (a) and its geometrical net (b).

TABLE I

GEOMETRICAL PROPERTIES OF CUBOCTAHEDRON

Vertices	Edges	Triangle faces	Square faces	Volume	Area
12	24	8	6	$\frac{5}{3}\sqrt{2}a^3$	$(6+2\sqrt{3})a^2$

In the items of Volume and Area, *a* represents the length of edge

B. *Implementation of Hardware*

According to the geometry of cuboctahedron, we tried to maximize the usability of a modular lattice robot by cutting a cuboctahedron into two equal halves through one of four regular hexagon planes, thus, each half has three equilateral square faces and four equilateral triangular faces. Each half is oriented with an equilateral hexagonal face and an equilateral triangular face as bottom and top respectively; this solid is well known as the triangular cupola in geometry, which is one of the Johnson solids.

We then designed a simple rotational mechanism to enable the cuboctahedron robot to get one rotational DOF between its two halves. A CAD model is shown in Fig.3.

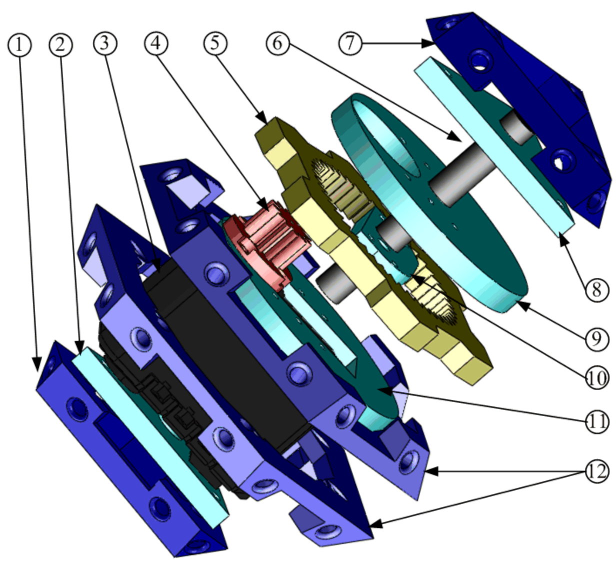
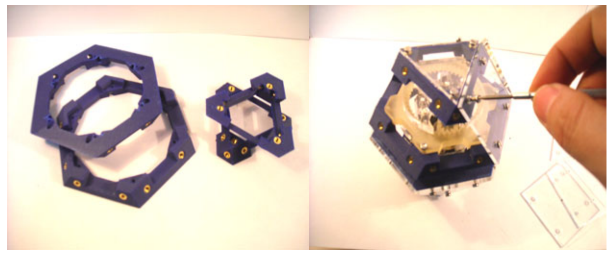


Fig. 3. Exploded view of the robot's CAD model. Where 1, 7 and 12 are frames; 2 is a motor holder; 3 is a servo motor; 4 and 5 are external and internal gears respectively; 6 is the rotation axle; 8 is the holder of the axle; 9 and 11 are gear locaters; 10 serves as an axle locater and holder.

In order to make fabrication easier, and to reduce the weight of the robot, we introduced a novel structure for the cuboctahedron robot, which reduces the robot body to two pairs of top-bottom frames and a group of triangular and square sheets. These top frames and bottom frames have a triangular shape and a hexagonal shape respectively, and include small holes on for affixing the outer plates. The edge length of our cuboctahedron robot is about 50 mm.

Based on the CAD model, we then built all frames in ABS plastic using a 3D printer (Dimension SST 768, Stratasys Inc). The cover sheets were cut from 0.125 inch thick acrylic sheets on a laser cutter (Epilog Helix 24, Epilog Laser Inc). We then employed brass inserts (M2.5 with internal thread) in the small holes on the ABS frames (Fig.4 (a)) after heating those inserts; thus all the outer plates could then be manually fixed onto the frames by small screws (Fig.4 (b)). The assembled robot has a transparent appearance and weighs approximately about 300 g. More importantly, it is quite easy to repair, assemble, and replace the components. The main design parameters are listed in Table II.



(a) (b)

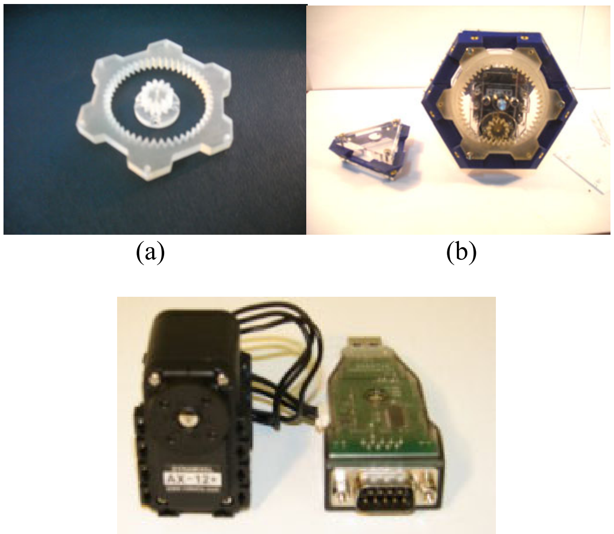
Fig. 4. Frame prototypes (a-b)

TABLE II

PHYSICAL PARAMETERS OF ROBOT

Size	Length of edge is 5cm
Weight	300g
DOF	1 rotational
Torque	6.5 N.m

As shown in the CAD model and prototype, this robot has a single rotational DOF. It is driven by a small DC servo motor through two gears. We defined the half of robot with the motor as the base half, and the other half is called the rotation half. The gear system in each robot consists of two spur gears. The smaller external gear has 12 teeth and is connected to the output shaft of actuator. The bigger internal gear has 48 teeth and is fixed to the hexagonal bottom frame of the other half (rotation half) of robot. This gearbox system has a ratio of 1:4, and can therefore significantly reduce the actuator's rotational speed and increase the torque. The maximum torque it can provide is about 6.5 Nm. We fabricated these gears by from the FullCure720 transparent acrylic-based photopolymer material on an Objet EDEN 260V 3D printer. The printed gears are shown in Fig.5 (a). They have enough strength for driving the robot in the required torque range. As for the motor, it is mounted to a triangular top frame of the base half. To keep the rotation axis stable, we installed a high-strength aluminum alloy shaft with a pair of miniature high-precision ball bearings at the center of the rotation half. Furthermore, self-locking retaining ring springs and some acrylic blocks were implemented into the driving mechanism as the locater and holder.



(a) (b)

(c)

Fig. 5. Driving system: printed gears (a), gearbox (b), and motor with controller (c).

C. Control and Communications

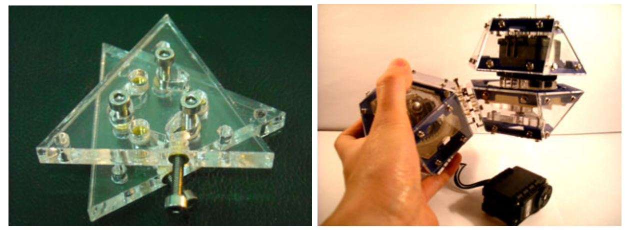
We employ the Dynamixel AX-12+ servo motor as the actuator for our robot. It can be programmed either for continues running mode or for servo operation. Moreover, the feedback of torque, angular position, angular speed, temperature and voltage can be obtained from the motors. The USB2Dynamixel (Robotis) device is used to directly drive the actuators by connecting to a USB port on a PC. Our robot's control program was written in C++ using IDE of Microsoft Visual Studio 2005, and run through the USB2dynamixel controller to all motors. We connect all modules in a daisy chain manner by cables, and the communication is based on a 1Mbps half-duplex asynchronous serial bus on TTL level, so each individual servo can be addressed individually, or a broadcast can be sent to all servos at once. The motor's power is provided through an AC-DC transformer at nine volts.

D. Genderless Docking Port:

A simple genderless docking port was developed for our robots. In this preliminary design, we have focused on the morphology of the modular robotic system; therefore, this docking port is just a simple mechanism for mechanical connection, without functional of power or signal transmission. Instead, the signal and power are transmitted by the external cables.

The docking port is a classic "pin-hole" style connection mechanism with the same number of pins and holes. These are equally arranged around the center of the docking plane on each port. We designed our docking hole as a guiding slot, so the pins can be located to the correct position by reversely rotating two ports by a small angle. A screw was also installed to the vertical direction of one hole as the locker of the docking device.

We fabricated two kinds of docking ports by cutting 0.25 inch thick acrylic sheets on the laser cutter. The triangular shaped port has 3 pairs of pin-holes and the square port has 4 pairs. Cap-head screws are used as the pins and locker. This replaceable device enables us to connect all robots just with simple, manual operations.



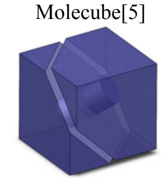
(a) (b)

Fig. 6. Triangular docking port (a) and two connected robots (b).

IV. REPRESENTATION OF MORPHOLOGY

A. Geometrical properties of robots

TABLEIII
COMPARISON BETWEEN MOLECUBE AND CUBOCTAHEDRON

	Molecube[5]	Cuboctahedron
		
Lattice type	Cube	Cuboctahedron
No. of connectable face	6	14
No. of reachable lattice location	3	7
No. of DOF	1	1
Planar manipulator	No	Yes

As the prototypes above show, the cuboctahedron robot provides both simple construction as well as a large variety of reconfiguration possibilities. Our previous Molecube robot exhibited a novel long diagonal of a cubic grid as the rotational axis, which enabled a single module to perform out-of-plane reconfiguration and to have three reachable lattice locations. Thus, each cube had three possible swivel states, and four possible orientations for the swivel axis. Here, the new modular robot extends the capabilities to a higher level. Every single robot has 14 connectable faces with triangular or square shape, and seven body locations can be reached on one robot. This means that each robot has six swivel states plus one fixed state on the top triangular face. Due to the different shape of connectable faces, our robot has nine potential relative orientations to the swivel axis. A comparison between the Molecube and our cuboctahedron robot can be seen in TableIII.

These capabilities allow plenty of opportunity for exploring flexible and diverse morphological reconfiguration in a modular robotic system. For example, two adjacent robots are shown in Fig. 7(a). The base module is fixed to the ground by a square face, and a second module is connected it by a square face also. They are initially arranged as a horizontal chain (left), but after swiveling the base module 120° in clockwise direction, these two robots configure as a vertical line to the ground (middle). Finally, another horizontal chain located in the vertical direction of the initial one is achieved by rotating the base module 120°, again in the clockwise direction. The final configuration is shown at the right. In terms of docking orientation, connecting through two square or two triangular faces could supply us with up to nine different relative orientations between the two robots' axes. For example, two robots connected by one set of square faces is shown in Fig.7 (b), their axes have potentially four geometrical relations in three ways: parallel (about 35° angle to ground), intersection (about 70°), and in two planes with an angle of roughly 80°. All of these angles were measured from the CAD model.

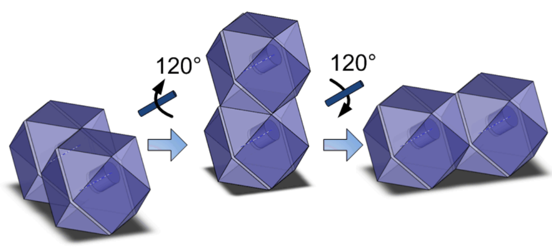
A pair of important rotational properties of the cuboctahedron robot should be noted. Firstly, in one cycle, if the two halves are rotated one sixth of a turn from the initial state with respect to each other, then the Johnson solid is obtained, which is well known as the triangular orthobicupola. However, if the two halves are rotated one third of one turn in either the clockwise or counter-clockwise direction, then the robot has the same geometry as it initially did. According to these geometric properties, we define the standard initial state of a single robot as a cuboctahedron, which is described in the previous section.

B. *Representation of morphology*

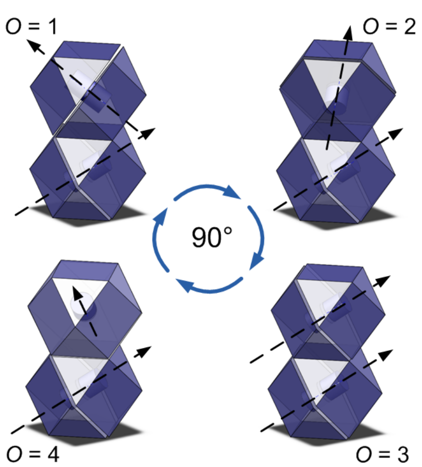
Our robot exhibits diverse connecting topologies, and also added complexity in configuring a scaled modular robotic system. A standard method for representing the robots' morphology is required for extensive applications. Previous works suggest several solutions for this issue. In this section, we propose a graph–based, direct morphological representation for the connected multiple robot system. The morphology can be described by simply connected nodes, where each node indicates a module, namely a cuboctahedron robot. The connecting position and orientation of each robot can be expressed entirely by a three parameter vector C_n:

$$C_n = (H, F, O)$$

where n is the module's ID. Module 1 is the starting module for describing the morphology. It can be defined by any terminal module in the robot system. H denotes which of the halves from the n-1 module (parent) the current module n is connected to, and it has two possible values: B and R, representing the base half and rotation half respectively (Fig. 8(a)). The concepts of base half and rotation half are different from the previous description of a single module's mechanical design. Here, the base half is connected to ground or to a module closer to the ground. The rotation half is the remaining half, which can be more easily rotated.

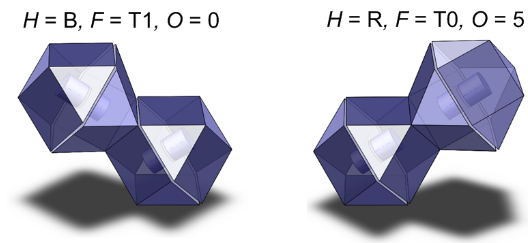


(a)

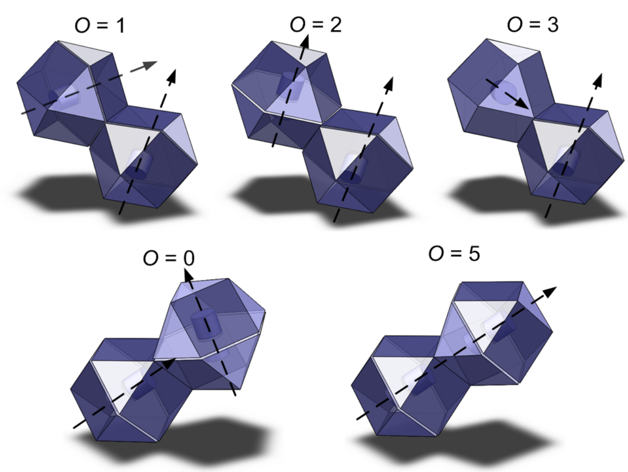


(b)

Fig. 7. An example of reconfiguration (a), and four relative orientations by docking through different square faces (b).



(a)



(b)

Fig. 8. Two possible halves for connection (a), and five relative orientations by docking through different triangular faces (b).

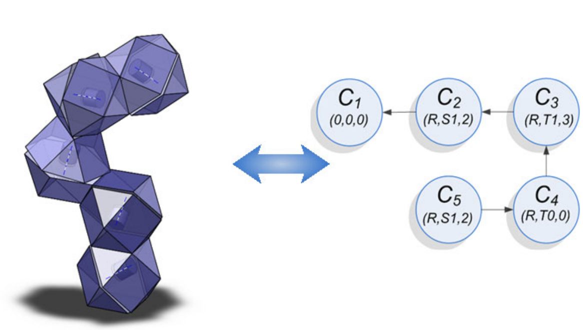
The connecting faces are indicated by *F*, which is relative to the parent module and current module. *F* has seven potential values from the set of *{T0; T1; T2; T3; S1; S2; S3}*, where *T* means the docking face is a triangular face and *S* represents the square shape docking face. The numbers following *T* and *S* are the ID of all docking faces with the same shape on one half of the parent module. The ID sequence is in the clockwise direction around the rotational axis of the parent module. Note that *T0* specifies the triangular face, which is perpendicular to the module's axis.

O represents the relative docking orientation to the parent module's axis. As we know, if two modules are docking through a triangular face, they have three relative orientations. We define the first orientation as the case that two modules have the axes in a cross manner, and the second one as the next orientation, which is obtained by rotating the current module from the first orientation with 120° in the clock-wise direction around the planar center of the docking face. Similarly, connecting through the square face provides four different orientations. We define these orientations in the same way but with a 90° rotation between the two orientations. We use numbers 1, 2, 3 and 4 to denote orientations for the 1st to 4th respectively. Due to the sole orientation of the *T0* face, we set 0 as the default orientation value. We use 5 to represent the docking orientation that puts the axis of two on a same line. This special case occurs only when two *T0* faces from two modules are docking together; finally, we could give a value set of *{0;1;2;3;4;5}* to *O* for representing all cases. The examples are shown in Fig.7 (b) and Fig.8 (b).

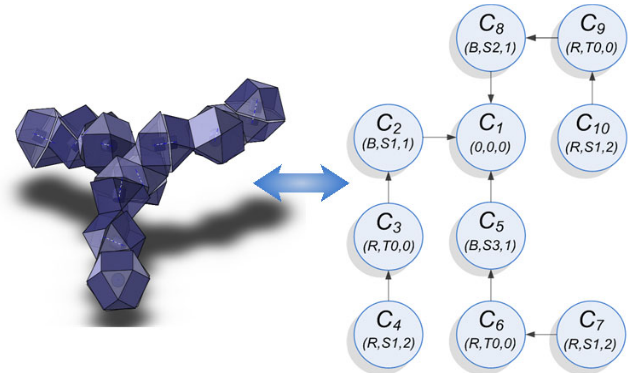
It should be noted that the first module commonly connects to a base or plays a central hub role in the system, so the elements of C_1 will be initialized as all 0.

To demonstrate the proposed representation on morphology, we built two robot systems with multiple modules. The first one is a five DOF open serial chain arm and another is a tripedal robot. Their morphologies and morphological graphs are shown in Fig.9.

The robot shown in Fig. 9 (a) has five identical modules (left), providing five DOF to the arm. We believe that it is possible to assemble an arm to meet a given requirement, such that the angle between an end-effecter and the ground (or any axis), is a reachable space and a set of points. As for the graph representation (right), we chose the module connected to ground as the first module of graph, so the $C_1 = (0, 0, 0)$. Each node with three parameters can uniquely specify a module with its connection position and orientation, and the arrow-line indicates the "connecting to" relations from the current module to the parent. Additionally, we designed a robot model with three legs and made from ten modules. The module located at the center is the body, and each leg has three modules connected in a chain to the body module. This morphology is shown in Fig.9 (b) (left). The central module is initialized as the base for the graph representation.



(a)



(b)

Fig. 9. Two robotic systems and their morphological representations: a robotic arm (a) and a tripedal robot (b).

V. Conclusion and Future Work

In this paper we designed a new, small modular robot based on a cuboctahedron body shape. Each robot has one degree of rotational freedom between two halves, each of which has the shape of a triangular cupola. Prototypes were fabricated according to the design, including the motor, gears, and mechanical connectors. This kind of modular robot has multiple connectable faces and several docking orientations, thus it significantly extends the solution space of possible morphologies. According to the robot's geometrical properties, we proposed a graph based representation method for describing the robot's morphology. Two examples were presented and a directed graph with three parameters was found that can simply and accurately specify the connecting topologies between modules. This representation has the potential to be applied as genotype in evolutionary design.

The work presented here is the initial mechanical stage of design, and many of continuations of this research are planned. Physical tests of the structural stability with actuation torque limits will be pursued in next step. A docking mechanism with integrated power and communication will also be developed for more autonomous applications. Moreover, we plan to design target morphologies by evolutionary search, so a physical simulator based on the Open Dynamics Engine (ODE) [30] is currently being developed. Additional work will include integrating a controller model to the morphological graph, and applying an evolutionary computation method (GP, e.g.) to evolve the robot's body and brain simultaneously within the simulation environment. With this capability, the modular robotic system could exhibit high level behaviors by simultaneously changing its morphology and controller.

ACKNOWLEDGMENT

We thank members of the Computational Synthesis Lab for useful discussions and especially John Amend for help editing this paper.

REFERENCES

[1] M. Yim, et al, "Modular Self-Reconfigurable Robot Systems," IEEE Robotics and Automation Magazine, vol.14, no.1, pp.43-52, 2007

[2] V. Zykov, E. Mytilinaios, B. Adams, and H. Lipson, "Self-reproducing machines," Nature, vol. 435, no.7038, pp. 163-164, 2005.

[3] K. Lee and G. S. Chirikjian, "Robotic Self-Replication: a descriptive framework and a physical demonstration from low complexity parts," IEEE Robotics and Automation Magazine, vol.14, no.4, pp.34-43, 2007

[4] N. Brener, F. B. Amar, and P. Bidaud, "From Crystals to Lattice Robots," In Proceedings on the IEEE International Conference on Robotics and Automation (ICRA), pp.3514-3519, 2008.

[5] V. Zykov, E. Mytilinaios, M. Desnoyer, and H. Lipson, "Evolved and Designed Self-Reproducing Modular Robotics," IEEE Transactions on Robotics, vol. 23, no. 2, 2007.

[6] E. H. Østergaard, K. Kassow, R. Beck, and H. H. Lund, "Design of the ATRON lattice-based self-reconfigurable robot," Autonomous Robots, vol.21, no.2, pp.165–183, 2006

[7] M. Yim, D. Duff, K. Roufas, "PolyBot: a Modular Reconfigurable Robot", In Proceedings on the IEEE International Conference on Robotics and Automation (ICRA), pp.514-520, 2000.

[8] A. Castano, A. Behar, and P. Will, "The conro modules for reconfigurable robots," IEEE/ASME Transactions on Mechatronics, vol. 7, no.4, pp. 403-409, 2002.

[9] C. Unsal, , H. Kiliccote, and P.K.Khosla, "I (CES)-Cubes: A Modular self-reconfigurable bipartite robotic system," In Proceedings of SPIE Sensor Fusion and Decentralized Control in Robotic Systems II, vol. 3839, pp. 258–269, SPIE. 1999.

[10] J.W. Suh, S.B. Homans, and M. Yim, "Telecubes: mechanical design of a module for self-reconfigurable robotics", In Proceedings on the IEEE International Conference on Robotics and Automation (ICRA), pp.4095-4101, 2002.

[11] R. Fitch, D. Rus, and M. Vona, "A basis for self-repair robots using self-reconguring crystal modules," Intelligent Autonomous Systems, vol. 6, pp. 903-910, 2000.

[12] K. Kotay and D. Rus, "Locomotion versatility through self-reconfiguration," Robotics and Autonomous Systems, vol.26, no.2, pp.17-32, 1999.

[13] M. Shimizu, T. Kato, M. Lungarella and A.Ishiguro, "Adaptive Reconfiguration of a Modular Robot through Heterogeneous Inter-Module Connections," In Proceedings on the IEEE International Conference on Robotics and Automation (ICRA), pp.3527-3532, 2008.

[14] S. Murata, et al, "M-TRAN: Self-reconfigurable modular robotic system," IEEE/ASME Transactions on Mechatronics, vol. 7, no.4, pp. 431-441, 2002.

[15] B. Salemi, M. Moll, and W.-M. Shen, "SUPERBOT: A deployable, multi-functional, and modular self-reconfigurable robotic system," In Proceedings of the IEEE/IRSJ International Conference on Intelligent Robots and Systems(IROS), pp. 3636-3641, 2006.

[16] A. Sproewitz, A. Billard, P. Dillenbourg and A. J. Ijspeert, "Roombots-Mechanical Design of Self-Reconfiguring Modular Robots for Adaptive Furniture," In Proceedings on the IEEE International Conference on Robotics and Automation (ICRA), pp.4259-4264, 2009.

[17] O. Chocron and P. Bidaud, "Genetic design of 3D modular manipulators" In Proceedings on the IEEE International Conference on Robotics and Automation (ICRA), pp.223-228, 1997.

[18] I. Tanev and K. Shimohara, "Co-evolution of active sensing and locomotion gaits of simulated snake-like robot," In Proceedings of the Conference on Genetic and Evolutionary Computation(GECCO'09), pp.257-264, 2009.

[19] I. M. Chen and J. W. Burdick, "Determining task optimal modular robot assembly configurations," In Proceedings on the IEEE International Conference on Robotics and Automation (ICRA), pp.132-137, 1995.

[20] O. Chocron and P. Bidaud, "Evolutionary algorithm for global design of locomotion systems" In Proceedings of the IEEE/IRSJ International Conference on Intelligent Robots and Systems(IROS), pp. 1573-1578, 1999.

[21] M. Park, S. Chitta, A. Teichman and M. Yim, "Automatic Configuration Recognition Methods in Modular Robots," The International Journal of Robotics Research, vol. 27, no. 3-4, pp.403-421, 2008.

[22] K. Sims. "Evolving 3D morphology and behavior by competition", In Proceedings of Artificial Life IV, pp.28-39. MIT Press, 1994.

[23] P. Funes and J. Pollack, "Evolutionary body building: Adaptive physical designs for robots," Artificial Life, vol.4, no.4, pp.337-357, 1998.

[24] F. Hou and W.-M. Shen, "Distributed, Dynamic, and Autonomous Reconfiguration Planning for Chain-Type Self-Reconfigurable Robots," In Proceedings on the IEEE International Conference on Robotics and Automation (ICRA), pp.3135-3140, 2008.

[25] G. S. Hornby and J. B. Pollack, "Creating high-level components with a generative representation for body-brain evolution," Artificial Life, vol.8, no.3, pp. 223–246, 2002.

[26] J. C. Bongard and R. Pfeifer, "Repeated structure and dissociation of genotypic and phenotypic complexity in artificial ontogeny," In Proceedings of the Conference on Genetic and Evolutionary Computation (GECCO'01), pp.829-836, 2001.

[27] A. Lyder, H.G. Petersen, and K. Stoy, "Representation and Shape Estimation of Odin, a Parallel Under-actuated Modular Robot", In Proceedings of the IEEE/IRSJ International Conference on Intelligent Robots and Systems(IROS), pp. 5275-5280, 2009

[28] M. J. Wenninger, Polyhedral Models. Cambridge University Press, 1971.

[29] Cuboctahedron, http://en.wikipedia.org/wiki/Cuboctahedron

[30] R. Smith, *Open dynamics engine*, 2005 Available: http://www.ode.org