

A Distributed Reconfiguration Method for 3-D Homogeneous Structure

Eiichi Yoshida Satoshi Murata Haruhisa Kurokawa
Kohji Tomita Shigeru Kokaji

Mechanical Engineering Laboratory, AIST, MITI
1-2 Namiki, Tsukuba-shi, Ibaraki 305 Japan e-mail: eiichi@mel.go.jp

Abstract

A distributed reconfiguration method is proposed for a 3-D reconfigurable machine, composed of many identical mechanical units. The method aims to enable the machine to transform itself into desired structure from an arbitrary initial configuration. The proposed method is implemented in such a way that each unit has identical software, so that any unit can play any role in the system. It is also featured by a stochastic relaxation process, which allows the system to converge to a given target structure by searching for a proper unit motion over many degrees of freedom. Furthermore, the method is extended for the structure to reconfigure itself dynamically according to the environment. The effectiveness of the method is confirmed by computer simulations.

1 Introduction

Recently, a self-reconfigurable structure which consists of many units has been attracting many researchers' attention by its versatility, functional flexibility, and fault-tolerance. Such a system can construct any desired structure without external help, and recover from failure by replacing the faulty units by spare units. The idea is inspired by the fact that living beings are composed of many cells, capable of self-reconfiguration in a decentralized manner.

Possible applications include planetary exploring vehicles or satellite antennae. Transported in a compact folded form, they can expand the structure when working, and repair themselves if some part is damaged.

Keeping this background in mind, we are developing a 3-D self-reconfigurable structure composed of many identical units. We put emphasis on homogeneity of the system to realize such characteristics as self-assembly and self-repair effectively.

Along with the hardware design, development of a method which controls reconfiguration process is another essential issue. The requirements for the method are:

Homogeneity. This is essential for hardware and software, so that any unit can be placed in any part of the system.

Decentralization. The method must work in a distributed way by using local information, preserving the robustness and concurrency of the reconfiguration process, especially in a large-scale system.

Let us briefly review several other studies regarding the control methods of reconfigurable mechanical systems. CEBOT [1] is a dynamically reconfigurable robot with heterogeneous structure. In software aspect, it is rather hierarchical. Chirikjian *et al.* developed a robot that can change its configuration dynamically [2] and uses centralized control like other manually reconfigured robots [3][4]. Kotay *et al.* has recently proposed a novel self-reconfiguring robotic molecule which can construct various 3-D structures [5]. In another approach, a cellular warehouse has recently been studied [6]. Although this system requires human operation for reconfiguration, the software in each cell is identical. Our dynamically reconfigurable fracta system [7]–[9] has homogeneity of both hardware and software.

Despite many studies listed above, distributed reconfiguration method for 3-D structure has not been developed. This paper, therefore, aims to develop a distributed method which enables the 3-D system to metamorphose into desired structure.

We have developed distributed self-reconfiguration methods for a 2-D fracta system [7]–[9]. This, however, cannot be straightforwardly extended to the newly constructed 3-D system, because the latter has far more degrees of freedom (twelve, as opposed to three in 2-D system) and constraints on the motion of units. To cope with these problems, this paper proposes a simple expression of connection description and a reconfiguration method based on a stochastic relaxation method. This method allows the 3-D system to converge to the desired structure in a decentralized manner. The homogeneity of software is also satisfied in its implementation. As a next step, the method is extended to allow dynamic reconfiguration in accordance with sensory inputs obtained from external environment. Reconfiguration method for large-scale system is also under development. Some simulations are conducted to verify the effectiveness of the proposed method.

2 Hardware

The 3-D identical unit should satisfy spatial symmetry. A solution we reached is illustrated in Fig. 1. The unit has six connecting arms in three orthogonal axes which can rotate independently, yielding twelve degrees of freedom. All the degrees of freedom in the six rotation and connec-

tion are controlled by a single DC motor through electromagnetic clutches. Please refer to our previous paper [10] for details of hardware implementation.

For a better understanding of the method, a brief mention of the basic motion is helpful. Transformations of the structure are accomplished by pairwise motion of units. Consider two connected units **X**, **Y** (shaded) on a plane made by 4 units (**A**) as shown in Fig. 2 (a). After releasing the connection to the underlying unit **Z**, unit **Y** moves to another position on the plane when the partner unit **X** rotates around axis **b-b** by 90 degrees (Fig. 2 (b)). By repeating this sort of pairwise motion, various 3-D structures can be configured without external help.

3 Description of Connection

In order to describe target structure, which is a final shape to be formed, we need to express how units are connected to each other in a simple way. In this section, we will classify the connection types of units, and define the reachable positions and movable types.

3.1 Connection types

A unit moves on the orthogonal-cubic lattice by repeating the pairwise motion explained in Section 2. A simple notation is introduced to describe units' connection state.

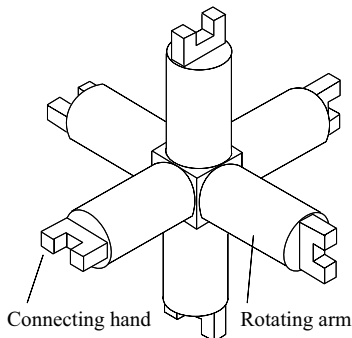


Fig. 1: Schematic view of 3-D unit.

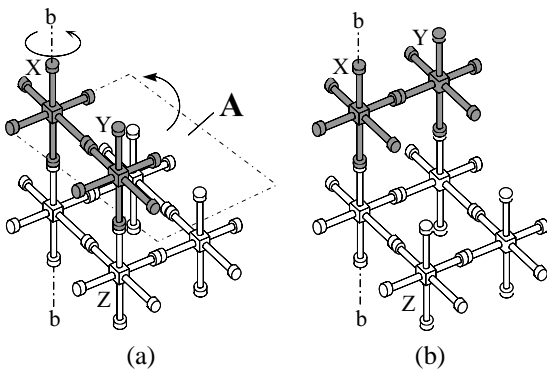


Fig. 2: Typical pairwise motion of 3-D unit. (a) initial configuration. (b) final configuration after pairwise motion of **X** and **Y** (**X** rotated about axis **b-b**).

We classified the connection types of a 3-D unit in one-neighborhood system into nine types as shown in Fig. 3. A unit is described by a sphere with connection arms. Each connection type is expressed by its valence (the number of connecting units) and types. As a result, there are only one or two types for each valence, by neglecting rotational transformations. When there are two types for a valence, we classify them into types 0 and 1 for convenience' sake. For instance, the type with valence 3 and type 0 is written as "C30", where "C" denotes "connection". If there is only one type at the valence, it is just "C1", "C5", or "C6".

The connection types will be used later to describe target structure.

3.2 Reachable Positions

Let us first examine the reachable positions in the simplest case of type C1 (Fig. 4). Assuming that a unit rotates by 90 degrees at one step, a unit with type C1 (shaded unit) has maximum two reachable positions (indicated by \times) which reside in a two-neighborhood system.

When enumerating reachable positions, one should consider the characteristics of the pairwise motion. The motion of passenger unit always requires at least one support unit on the rotating axis of the carrier unit (Fig. 4). If this condition is not satisfied, the corresponding reachable positions should be excluded. In the example in Fig. 2, unit **X** (with type C21) cannot move upward since the pivot unit **Y** has no unit connected in the rotating axis for upward pairwise motion. Reachable positions already occupied by other units are also excluded.

Some other constraints are also imposed upon the unit motion. The motion should be made without carrying and colliding other units, and the connectivity of the whole structure should be maintained after the motion [13].

Valence	1	2	3	4	5	6
Type 0	C1	C20	C30	C40	C5	C6
Type 1		C21	C31	C41		

Fig. 3: Connection types for 3-D units.

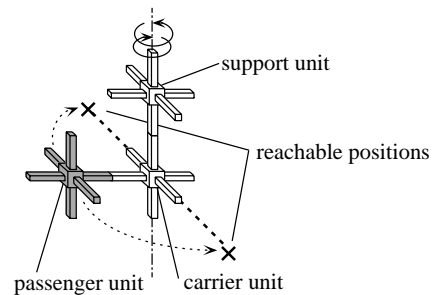


Fig. 4: Reachable positions of C1 type unit.

Reachable positions violating these constraints should be excluded as well. These constraints on 3-D system are one of major difference from 2-D fracta system.

3.3 Movable Types

If units with larger valence (say, more than 4) move, the constraints above become more unlikely to be satisfied (especially connectivity). We thus allow only units with limited connection types to move, which are referred to “movable types.” Types C1, C21, and C31 are selected as movable types here since self-reconfiguration becomes difficult to proceed if it is limited to type C1. The reachable positions for C21 and C31 are easily obtained by superposing those of type C1.

4 Self-reconfiguration Method

This section presents a distributed self-reconfiguration method which allows the 3-D system to transform from arbitrary state into a given target structure.

As mentioned in Section 1, the homogeneity of software should hold, such that any unit should be able to play any role of the target structure. Target structures are expressed by a method suitable for homogeneous system using connection types.

One of major difficulties of developing 3-D reconfiguration method lies in the multiplicity of degrees of freedom in 3-D system. In the 2-D fracta system [7]–[9], when a unit makes a motion, it has only to choose one of two directions, clockwise or counterclockwise. By contrast, a unit might have more than five reachable positions in the newly designed 3-D system. Moreover, the number of reachable positions varies with connective situation of the unit.

Another difficulty is that the method should be implemented in a distributed manner. Namely, each unit should behave according to local information to achieve the global target. This is extremely important for large-scale system.

We introduce a stochastic relaxation process based on simulated annealing to cope with the above difficulties. Together with simple connection description, this process characterizes the 3-D reconfiguration method compared to the formally developed methods for 2-D system [7]–[9]. This method allows the system to converge to given target structure in a distributed fashion.

An alternative solution is a heuristic method like `if[configuration]-then[motion]` rules. It is, however, difficult to list all possible configurations which could be encountered, as noted above and also in another paper [13]. In this case, we are forced to provide ad hoc heuristics for every different target structures. Since we intend to make the method simple and general, we adopted the above connection type notation and a stochastic process.

The method consists of the following three procedures: Each unit...

- (i) tests whether it is movable and, if so, calculates all the reachable positions using the information obtained through local communication.
- (ii) calculates the “difference” D between the current and the target configuration for all reachable positions, and
- (iii) moves with a greater probability in the direction of smaller potential calculated from the difference D .

We call this series of procedures a “control step”. This control step is assumed to be synchronous in all the units. Synchronization is realized by a simple distributed method [11]. We also assume that all the units are connected in the initial configuration.

4.1 Target Description

We describe the target structure using the connection types. The connection state of a unit is written in the form of a “type-list” like

$$\text{type} [\text{type}_1, \dots, \text{type}_n] , \quad (1)$$

(sorted in decreasing order of valence)

where

- n : number of neighbors of the unit,
- type: connection type of the unit,
- type $_j$: connection type of the unit’s j -th neighbor.

Let us take an example of a cube made of eight units (Fig. 5(a)). Each unit has C31 type and is connected to three C31 types, so this situation is written as the type-list C31[C31, C31, C31].

The target structure is expressed as a collection of these type-lists. In this particular case, the target description includes only one type-list as follows:

$$\text{C31} [\text{C31}, \text{C31}, \text{C31}] . \quad (2)$$

For another example shown in Fig. 5(b), the target description is given by two type-lists:

$$\begin{aligned} &\text{C31} [\text{C41}, \text{C31}, \text{C31}] , \\ &\text{C41} [\text{C41}, \text{C41}, \text{C31}, \text{C31}] . \end{aligned} \quad (3)$$

Descriptions can be uniquely determined from the target structure. In general, however, it is a many-to-one mapping; there may be multiple structures corresponding to one

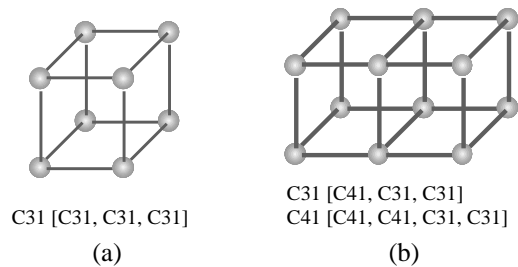


Fig. 5: Description of target structure.

description for many-unit structure. In this paper, we use mainly this simple description to verify the feasibility of 3-D self-reconfiguration method. The problem of description ambiguity will be addressed in future work. As one solution for this problem, a tentative reconfiguration method for large-scale system will be briefly described in Section 7.

4.2 Distributed Reconfiguration using Stochastic Relaxation Method

Each unit must select one of multiple reachable positions based on local information so that the system approaches the target structure.

Calculation of Difference

First, for each reachable position i (including “no motion” as $i = 0$), each unit computes the difference D_i from target structure. Current structure can be acquired in the form of type-list (1) through local communication.

The difference D_i is calculated using (4) based on local information. The value D_i is in fact the difference between the state at reachable position i and the closest target type-list by taking the minimum value. If D_i becomes zero, it means that one of the target type-list is completed at the reachable position i :

$$D_i = \min_{k=1}^M [W \text{diff}(\text{type}^g(k), \text{type}^i) + \sum_{n \in \text{neighbors at point } i} \text{diff}(\text{type}_n^g(k), \text{type}_n^i)] , \quad (4)$$

where

- M : number of target type-lists,
- type^i : connection type of a unit if it moves to point i ,
- type_n^i : connection type of n -th neighbor of a unit if it moves to point i ,
- $\text{type}^g(k)$: connection type of k -th target type-list,
- $\text{type}_n^g(k)$: connection type of n -th neighbor in k -th target type-list.

In (4), $\text{diff}(A, B)$ denotes the difference between type A and type B as follows¹:

$$\text{diff}(A, B) = \begin{cases} 1 - \delta(A, B) & \text{for } v(A) = v(B), \\ a \times |v(A) - v(B)| & \text{for } v(A) \neq v(B). \end{cases} \quad (5)$$

We use the following Kronecker delta-function (since there are only one or two types at each valence):

$$\delta(\text{type}_1, \text{type}_2) = \begin{cases} 1 & \text{for } \text{type}_1 = \text{type}_2 , \\ 0 & \text{otherwise} . \end{cases} \quad (6)$$

¹If the number of neighbors differs in current and goal type-lists, valence “0” is substituted for the blank.

Here $v(\text{type}_1)$ denotes the valence of type_1 , and the parameter a is to weight the difference in valences. In (4), W is a weighting ratio of a unit’s difference to that of neighboring units. The difference is calculated in (4) in more general and simple way than 2-D fracta system.

Stochastic Relaxation Method

Next, the probability P_i that a unit moves to the reachable position i is computed based on Markov Random Field (MRF) [12], by using the difference D_i as a potential function. Besides the convergence to the minimum potential, this method has another advantage — it allows parallel implementation. Thus, MRF is often utilized for parallel image processing like image recognition or noise reduction. It is also considered to be suitable for 3-D reconfiguration of distributed mechanical units. The MRF-based probability is given by:

$$P_i = \frac{1}{Z} \exp\left(-\frac{D_i}{T}\right) , \quad (7)$$

$$Z = \sum_{i \in \text{reachable positions}} \exp\left(-\frac{D_i}{T}\right) ,$$

where the parameter T corresponds to “artificial temperature” in the simulated annealing method. The probability P_i is uniform for each reachable position i with large T value, while P_i for the lowest potential approaches 1 if T is small. T is decreased according to the following formula:

$$T = \frac{C}{\log(1+t)} , \quad (8)$$

where t and C are the number of control steps and constant parameter concerning system size, respectively. By decreasing T gradually in this way, units search uniformly in “searching space” in earlier steps. As time elapses, the units come to make more concentrated motion to the closest point to complete target structure.

The above relaxation computation allows the whole system to converge to the target structure even though there are many alternatives of movement. This method can be implemented in a distributed and parallel way, so each unit can make the calculation using local information only.

5 Simulations

In this section, we will verify the effectiveness of the proposed method by computer simulations. The target structures in Fig. 5 (a), (b) are constructed from given initial configurations in simulations. The parameters are empirically determined as $W = 3.0$, $a = 1.0$ and $C = 10.0$. In the following simulations, only one unit is allowed to move at each step for simplicity².

²This constraint can be relaxed, for example by allowing multiple units’ motion if the they are sufficiently distant from others.

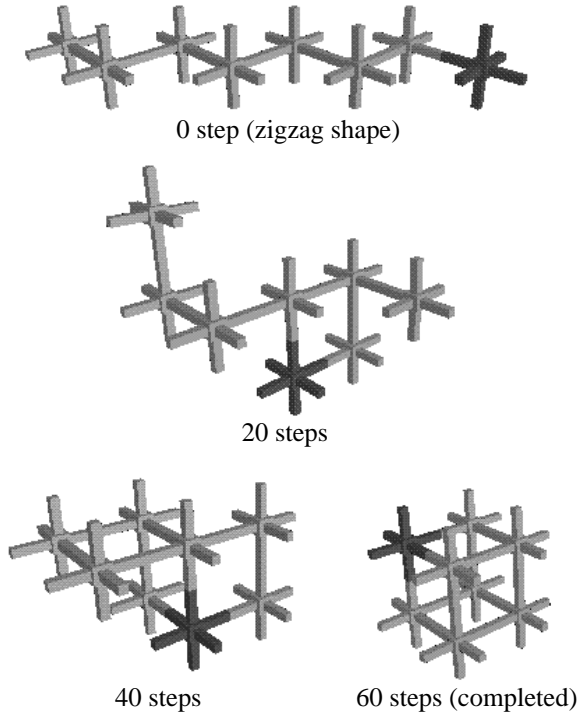


Fig. 6: Self-reconfiguration simulation (8 units).

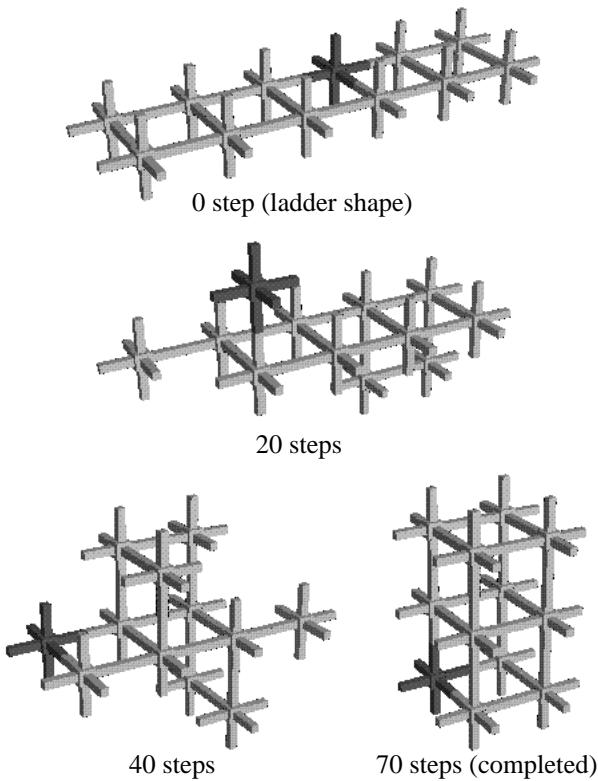


Fig. 7: Self-reconfiguration simulation (12 units).

Figures 6 and 7 show graphical views of successful simulation sequences of 8 and 12 units, starting from zigzag and ladder shapes, respectively. These are initial states where reconfiguration sequences are not trivial.

Here, we picked up the cases in which the self-reconfiguration finished in relatively small steps. These results demonstrate the capacity of distributed search for target configuration.

To verify the effectiveness, a quantitative evaluation is given using the average difference D per unit through 200 simulations. These simulations start from the same initial configurations, but different stochastic relaxation processes are brought about by changing seeds for random number generation. The simulation results are compared to those of random / steepest-descent selection of reachable position instead of MRF used in (7) and (8). In the steepest-descent method, the reachable position which has the minimum difference is always selected.

In Figs. 8 and 9, the average D per unit is plotted in terms of steps. The results obtained from the proposed method (indicated by “proposed”) and random (“random”) / steepest-descent (“steepest”) case are drawn in solid and fine / sparse broken lines respectively. The initial value of D is different in Figs. 8 and 9 because the simulations start from topologically different configurations.

As seen in the graphs, the average D converges to zero as time elapses in both results of the proposed method. This means that the target structures are completed in most of

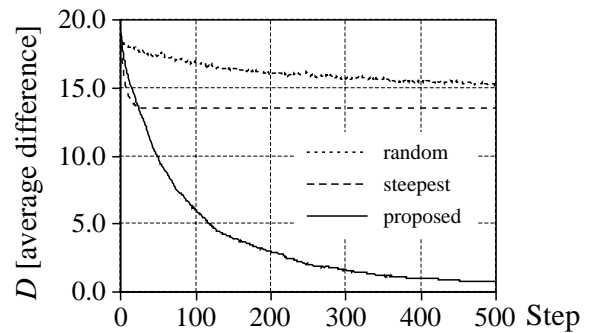


Fig. 8: Difference per unit (8 units).

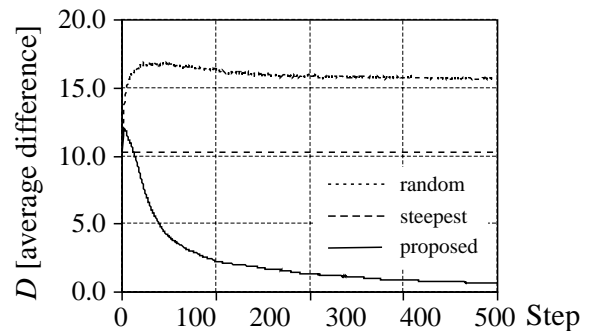


Fig. 9: Difference per unit (12 units).

simulations by this method.

By contrast, the random motion forces D to take large values (> 10) throughout simulation steps; the system hardly converges to desired shape. We can find an interesting phenomenon in Fig. 9, in which D increases from initial state in random case. This is because the initial ladder shape is rather closer to the target than shapes changed by random motion.

As to the steepest-descent motion selection, it shows an early convergence to high D value although it decreases drastically in the very first steps of simulations. This means that the reconfiguration process is easily trapped in a local minima.

It is concluded from these results that the solution (target structure) cannot be reached by just random or steepest-descent motion.

These simulation results, therefore, clearly confirms that the method based on stochastic relaxation works effectively to reconfigure the 3-D system into given target structures. To realize self-assembly of different structures, we have only to change target descriptions owing to its general characteristics, unlike ad hoc heuristics.

This method gives good performance for up to twenty units to construct such structures consisting of cubes as those in Fig. 5. But it still remains to be improved by allowing the structure to reconfigure itself dynamically according to changing environments, and by reducing ambiguity in target description for self-assembly of more complex shapes. An attempt of these improvements will be addressed in the following sections.

6 Dynamic Reconfiguration

One of the next phases of development is dynamic reconfiguration, so that the structure can adapt its configuration when changes take place in the internal state or in the external environment.

The method we have developed so far is “static” one which ends up when the target structure is completed. We are on the way accommodating the method to allow it to change their target structure dynamically according to the present situation (Fig. 10). This is just like living cells in which appropriate regions of chromosome are activated in compliance with their metabolic environments. This section presents the current stage of this extension of the method to dynamic reconfiguration.

An example of the application is a mobile robot which detects some faults and make a temporary repairs, using the structure equipped with optical, acoustic, or chemical sensors, as illustrated in Fig. 11.

As a simple case, we implemented a method for the system to move toward a light source (this can be other kinds of sensory input). This motion is generated by switching its target structure. The system consists of eight units and is

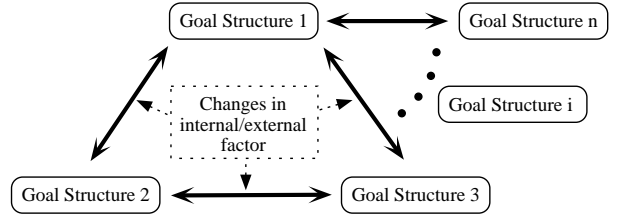


Fig. 10: Dynamic change of target structure according to internal/external factor

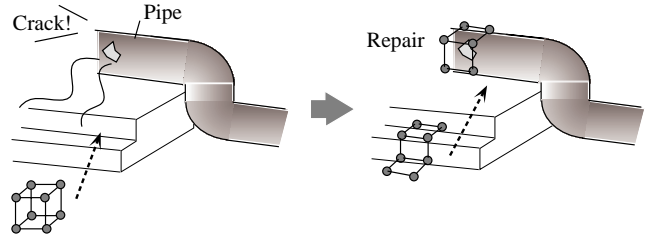


Fig. 11: An application of dynamically reconfiguring structure

given two target structures 1 and 2, cube and ladder configurations respectively (Fig. 12). Each of the units is assumed to have a sensor to detect the direction from which the light comes. The following motion method is executed instead of the reconfiguration method in Section 4. Each unit...

- examines the sensory inputs and runs the self-reconfiguration process (i)–(iii) described in Section 4 with the current target structure,
- switches the target structure if the difference D_0 (no motion) remains lower than a threshold D_{th} during a certain period t_{stable} .

The information about sensory input is transmitted to one-neighborhood when communicating the connection types. Based on the local information, the units assign a higher moving priority to reachable positions approaching the light source such that the whole structure can move toward the light source. The completion of target structure is not strictly required here, as the goal is to generate the directive motion.

We have simulated the proposed motion method to verify its feasibility. In the initial state, the units have cube configuration placed on a plane and the direction of light is along x -axis as shown in Fig. 13(a). We assume that the

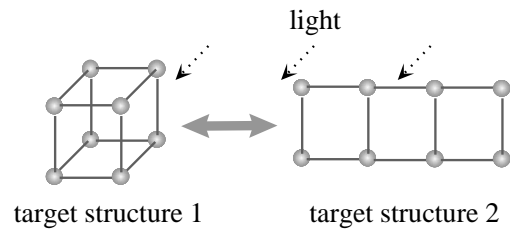
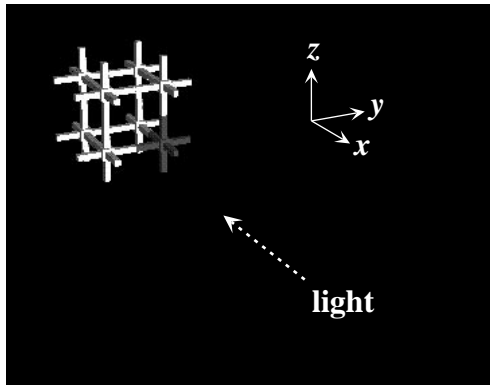
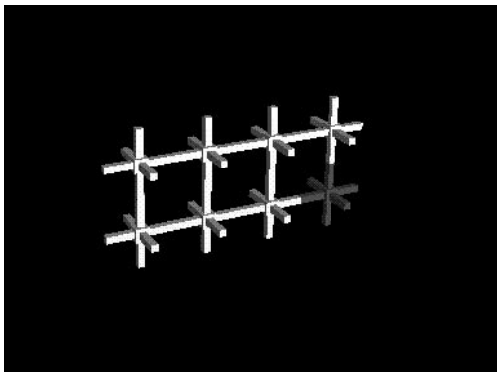


Fig. 12: Two target structures for dynamic reconfiguration

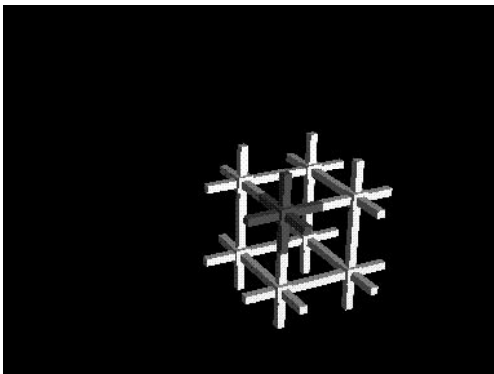
units touching the plane are fixed to it by some connecting mechanism. When switching the target from cube to ladder, the four units whose sensors directly detected the light are set non-movable to assure the motion. The parameters are: $W = 0.5$, $a = 2.0$, $C = 5.0$, $D_{th} = 1.0/8.0$ (for target structure cube/ladder), $t_{stable} = 30$. Figure 13(b), (c) shows a sequence of motion; the group of units makes a consistent motion to the source. The center of mass of the units along x -axis is plotted in terms of step in Fig. 14. We can observe that the whole structure steadily moved by 9.13 [lattice unit length] toward the source in 1000 steps. The



(a) 0 step



(b) 375 steps



(c) 500 steps

Fig. 13: Motion of units toward light source

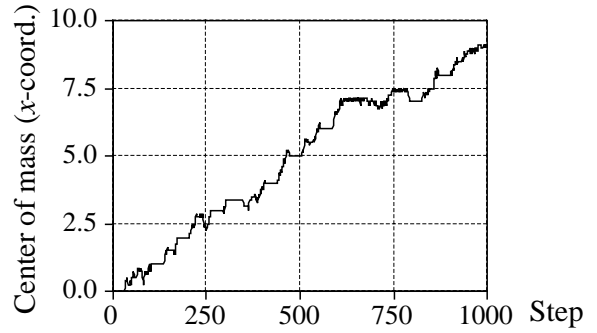


Fig. 14: Motion of units (center of mass, x -coordinate)

velocity is rather slow, but it may be improved by adjusting the parameters of the stochastic process.

The simulation result has demonstrated that dynamic reconfiguration is possible by extending the framework of the distributed method using target structure switching based on sensory information. Although the advantages of decentralization are not fully utilized in this simple example, we would like to emphasize that the basic function of a distributed adaptation process has been verified.

7 Towards Reconfiguration of Large-Scale System

The reconfiguration method based on stochastic process works well for small number of units, say, up to twenty units. However, it becomes less effective for larger system in which ambiguity in description and importance of reconfiguration “sequence” are significant. We are currently developing a method which allows such reconfiguration of many-units based on a developmental scheme like nucleation method in 2-D system [8].

We introduce basic structures which determines the geometrical relationship between “nodes” as shown in Fig. 15. By assigning another sub-structure to each nodes, various complex shapes can be described in a recursive manner as illustrated in Fig. 16. Self-reconfiguration proceeds by assembling first top-level structure, then down to sub-structure, and so forth. This method is expected to realize simple description of large-scale system and concurrent assembly of multiple sub-structures. Figure 17 shows an example of expansion motion from cube to antenna-like flat shape. The details of this method are to be reported in future work.

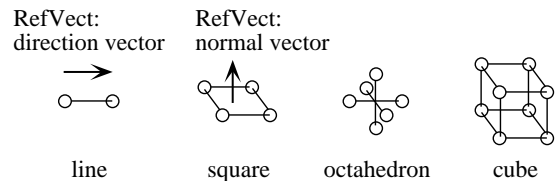


Fig. 15: Primitive types of description

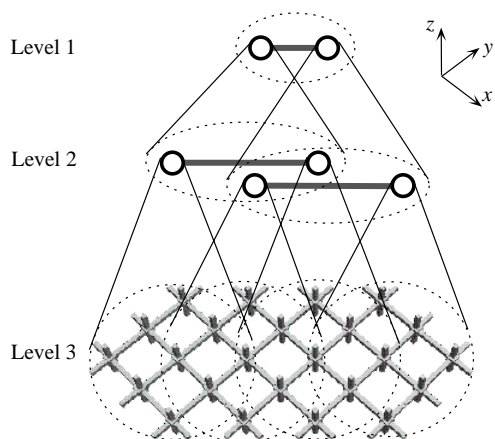


Fig. 16: Recursive description of structure

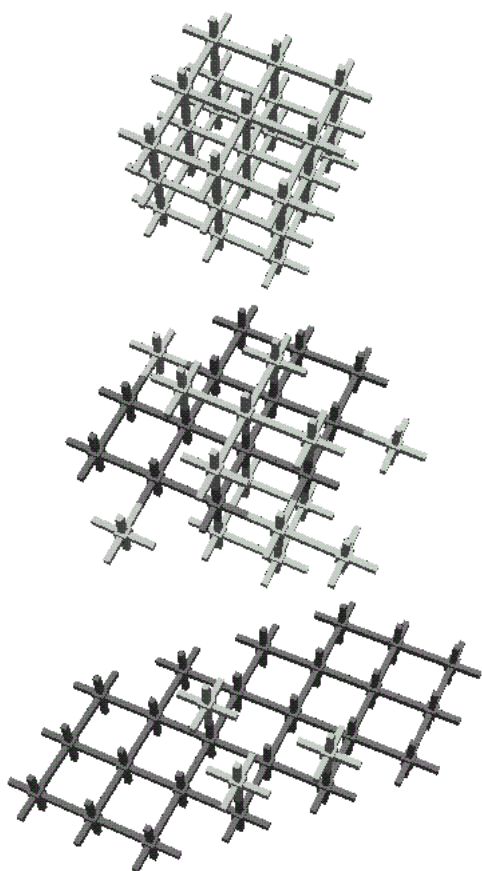


Fig. 17: Self-assembly of many-unit structure

8 Conclusion

This paper presented a distributed method for self-reconfiguration of 3-D structure. Keeping the homogeneity in software aspect for the flexibility and the fault-tolerance, we developed a distributed reconfiguration method.

A stochastic relaxation method was adopted, called MRF, which allows the system to achieve desired target

structure searching over many degrees of freedom. We implemented the method in a distributed way, so that each unit can decide its behavior using local communication.

The effectiveness of the proposed method has been verified by computer simulations. The method effectively brought the system to metamorphose into desired structures.

We are currently on the way to dynamic reconfiguration and self-assembly of large-scale system. Their feasibility was shown by simulations through tentative implementation.

References

- [1] T. Fukuda, et al.: "Dynamically Reconfigurable Robotic Systems," *Proc. Int. Conf. on Robotics and Automation*, 1581–1586, 1988.
- [2] G. Chirikjian, et al.: "Evaluating Efficiency of Self-Reconfiguration in a Class of Modular Robots," *J. of Robotic Systems*, Vol.12, No.5, 317–338, 1996.
- [3] M. Yim: "New Locomotion Gaits," *Proc. Int. Conf. on Robotics and Automation*, 2508–1524, 1994.
- [4] G. J. Hamlin, et al.: "Tetrobot Modular Robotics: Prototype and Experiments," *Proc. Int. Conf. on Intelligent Robots and Systems (IROS'96)*, 390–395, 1996.
- [5] T. Kotay, et al.: "The Self-reconfiguring Robotic Molecule," *Proc. Int. Conf. on Robotics and Automation*, 424–431, 1998.
- [6] T. Sakao, et al.: "The Development of a Cellular Automatic Warehouse" *Proc. Int. Conf. on Intelligent Robots and Systems (IROS'96)*, 324–331, 1996.
- [7] S. Murata, et al.: "Self-assembling Machine," *Proc. Int. Conf. on Robotics and Automation*, 441–448, 1994.
- [8] K. Tomita, et al.: "Reconfiguration Method for a Distributed Mechanical System," *Distributed Autonomous Robotic System 2*, H. Asama, et al., eds., 17–25, Springer, 1996.
- [9] E. Yoshida, et al.: "Distributed Formation Control for a Modular Mechanical System," *Proc. Int. Conf. on Intelligent Robots and Systems (IROS'97)*, 1090–1097, 1997.
- [10] S. Murata, et al.: "A 3-D Self-Reconfigurable Structure," *Proc. Int. Conf. on Robotics and Automation*, 432–439, 1998.
- [11] S. Kokaji, et al.: "Clock Synchronization algorithm for a Distributed Autonomous System," *J. of Robotics and Mechatronics*, Vol.8, No.5, 427–434, 1996.
- [12] S. Geman, et al.: "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE. Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-6, No.6, 721–741, 1984.
- [13] H. Kurokawa, et al.: "A 3-D Self-Reconfigurable Structure and Experiments," *Int. Conf. on Intelligent Robots and Systems (IROS'98)*, in press, 1998.