Hiding Sliding Cubes: Why Reconfiguring Modular Robots Is Not Easy

Tillmann Miltzow
Utrecht University, The Netherlands
t.miltzow@uu.nl

Irene Parada
TU Eindhoven, The Netherlands
i.m.de.parada.munoz@tue.nl

Willem Sonke
TU Eindhoven, The Netherlands
w.m.sonke@tue.nl

Bettina Speckmann
TU Eindhoven, The Netherlands
b.speckmann@tue.nl

Jules Wulms
TU Eindhoven, The Netherlands
j.j.h.m.wulms@tue.nl

Abstract

Face-connected configurations of cubes are a common model for modular robots in three dimensions. In this abstract and the accompanying video we study reconfigurations of such modular robots using so-called sliding moves. Using sliding moves, it is always possible to reconfigure one face-connected configuration of n cubes into any other, while keeping the robot connected at all stages of the reconfiguration. For certain configurations $\Omega(n^2)$ sliding moves are necessary. In contrast, the best current upper bound is $O(n^3)$. It has been conjectured that there is always a cube on the outside of any face-connected configuration of cubes which can be moved without breaking connectivity. The existence of such a cube would immediately imply a straight-forward $O(n^2)$ reconfiguration algorithm. However, we present a configuration of cubes such that no cube on the outside can move without breaking connectivity. In other words, we show that this particular avenue towards an $O(n^2)$ reconfiguration algorithm for face-connected cubes is blocked.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Sliding cubes, Reconfiguration, Modular robots

Digital Object Identifier 10.4230/LIPIcs.SoCG.2020.78

Category Media Exposition

Supplementary Material The code used, along with a list of coordinates of the cubes in the construction, can be found at `https://github.com/tue-aga/cubes-checker`.

Funding This work was initiated at the 5th Workshop on Applied Geometric Algorithms (AGA 2020).
Tillmann Miltzow: Supported by the Netherlands Organisation for Scientific Research (NWO) under project no. 016.Veni.192.250.
Bettina Speckmann: Partially supported by the Netherlands Organisation for Scientific Research (NWO) under project no. 639.023.208.
Jules Wulms: Supported by the Netherlands eScience Center (NLeSC) under project no. 027.015.G02.

Acknowledgements We thank Tim Ophelders for his help with the computational verification of the properties of our configuration.

1 Introduction

Modular robots consist of several identical modules that can assemble themselves into various configurations. Often such modules live on a lattice and can move only relative to each other. One of the main algorithmic challenges in this context is to find efficient ways to reconfigure one configuration into any other. This topic has attracted significant attention in the computational geometry community [1, 2, 3, 4, 5, 7]; yet many fundamental questions remain open.

One frequently studied paradigm of modular robots is the so-called *sliding cube model*. Here, a *configuration* C is a face-connected set of cubes on the cubic grid. The model allows two types of moves: *slides* and *convex transitions* (see Figure 1). A cube c is *movable*, that is, c is allowed to perform a move, if and only if removing c from the configuration C leaves $C \setminus c$ face-connected.

slide   convex transition

Figure 1 The two types of moves in the sliding cube model.

To reconfigure a vertical pillar of n cubes into a horizontal line, $\Omega(n^2)$ moves are necessary, since that is the sum of the n (grid) distances to any horizontal plane through one of the cubes. Furthermore, Abel and Kominers [1] showed that $O(n^3)$ slides and convex transitions suffice to transform any configuration into a strip. Since both moves are reversible, this implies that any two configurations of n cubes can be reconfigured into each other with $O(n^3)$ moves. It has been conjectured that there is a simple way to close this gap, using so-called outer cubes. Specifically, we say that the *outside* of a configuration C is the unbounded maximal set of face-connected grid cells not in C. A cube c in a configuration C is an *outer cube* of C if it is face-connected to the outside. Furthermore, a *void* is a maximal face-connected set of grid cells neither in C, nor on the outside. If any configuration of cubes contains at least two movable outer cubes, then it is straight-forward to construct a horizontal line of cubes using $O(n^2)$ moves. In particular, we can iteratively construct the horizontal line to the right of a rightmost cube in the original configuration. If we are not done, we can find one movable outer cube c not yet part of the horizontal line. Then c can slide to the right of the horizontal line in $O(n)$ moves (we can actually find the shortest route by DFS). If the configuration does not contain voids, then there are indeed always two movable outer cubes. This fact about outer cubes has been used as the basis of reconfiguration algorithms [6, 8]. However, in this paper and in the associated video we show that there are configurations of cubes where no outer cube is movable.

2 The configuration

Consider the face-adjacency graph of the cubes. This graph has a spanning tree where all leaves correspond to cubes on the boundary of the configuration (either face-adjacent to the outside or to a void) [7]. Any tree with at least two vertices has at least two leaves. Hence, every configuration with more than one cube contains at least two movable cubes. It follows that the face-adjacency graph of configurations with the fewest movable cubes is a path. We

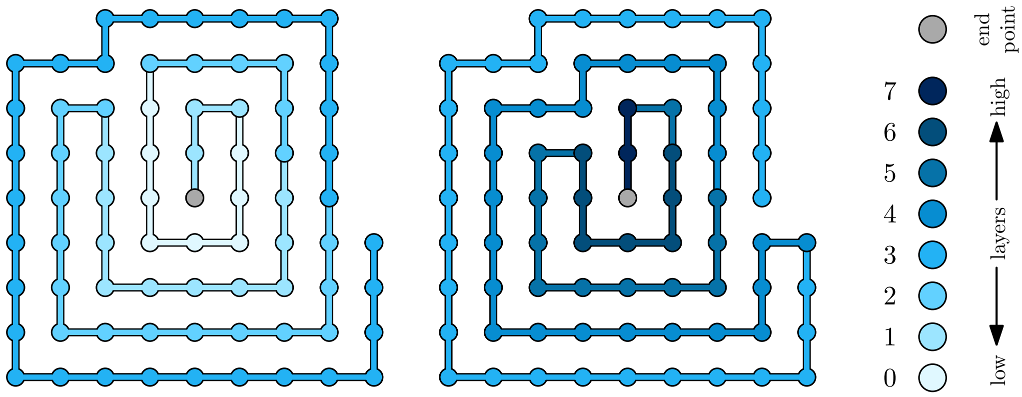

Figure 2 Adjacency graph of a basket, in top-down view: bottom *(left)* and lid *(right)*. The colors correspond to layers. Some positions contain two or three cubes stacked on top of each other. Note that layer 3 is drawn in both halves.

are hence basing our construction on such configurations. Intuitively, we are building two rope baskets out of the path of cubes. In each basket we are creating a void, in which we are hiding one endpoint of the path. The other endpoint becomes the top of the lid of the basket. Finally, we connect the two lids via a path.

We build each basket layer by layer (see Figure 2 (left) and Figure 3 (left)). This is a delicate construction, as we have to enclose a void while not creating cycles in the face-adjacency graph. Starting at an endpoint e of the path, on Layer 2, we first loop around e on Layers 1 and 0 in such a way that we trap e locally on the inside. The remainder of the construction closes this void, so that e is actually hidden inside it. To do so we build a lid for the basket, as shown in Figure 2 (right) and Figure 3 (right), ending with an endpoint e' on the top of the lid. All that remains now is to connect the top cubes of each basket via a path of cubes that does not introduce cycles (see Figure 4).

We computationally verified the correctness of our basket construction. The code used, along with a list of coordinates of the cubes in the construction, can be found at `https://github.com/tue-aga/cubes-checker`.

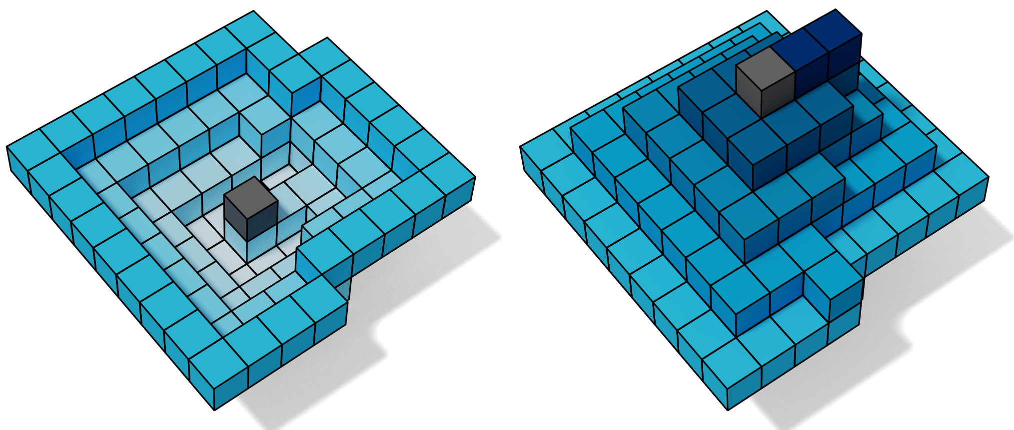

Figure 3 3D rendering of a basket: bottom only *(left)*, and entire basket *(right)*.

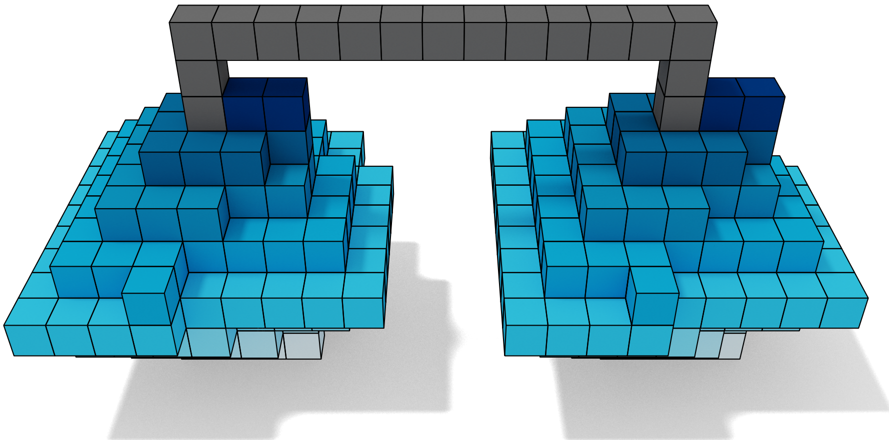

Figure 4 3D rendering of the complete construction.

3 The video

Our construction is inherently three dimensional and two-dimensional illustrations such as Figures 2 and 3 can convey only a schematic view of one of the baskets. To make our result more accessible and to stimulate interest in this fascinating class of modular robots, we hence produced a video using 3D animation in Blender. Our video first introduces modular robots and the sliding cube model, and then proceeds to build our configuration step-by-step.

4 Future work

The sliding cubes model extends to dimensions higher than three. Hence, concerning our specific construction, one can wonder if a similar example can be constructed in four or higher dimensions. Furthermore, we need two voids to trap both endpoints of our path. Is there an example that uses only one void? An obvious open question on the algorithmic side is the existence of an algorithm that reconfigures two configurations of n cubes in $O(n^2)$ time. Furthermore, the question arises if there are input sensitive algorithms to reconfigure cubes. That is, given two configurations C_1 and C_2, can we find the minimum number k of moves to reconfigure C_1 into C_2?

References

1 Zachary Abel and Scott Duke Kominers. Universal reconfiguration of (hyper-)cubic robots. *ArXiv e-Prints*, 2011. `arXiv:0802.3414v3`.

2 Hugo A. Akitaya, Esther M. Arkin, Mirela Damian, Erik D. Demaine, Vida Dujmovic, Robin Flatland, Matias Korman, Belen Palop, Irene Parada, André van Renssen, and Vera Sacristán. Universal reconfiguration of facet-connected modular robots by pivots: The $O(1)$ Musketeers. In *27th Annual European Symposium on Algorithms (ESA 2019)*, volume 144 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 3:1–3:14, 2019. `doi:10.4230/LIPIcs.ESA.2019.3`.

3 Greg Aloupis, Sébastien Collette, Mirela Damian, Erik D. Demaine, Robin Flatland, Stefan Langerman, Joseph O'Rourke, Val Pinciu, Suneeta Ramaswami, Vera Sacristán, and Stefanie Wuhrer. Efficient constant-velocity reconfiguration of crystalline robots. *Robotica*, 29(1):59–71, 2011. `doi:10.1017/S026357471000072X`.

4 Greg Aloupis, Sébastien Collette, Mirela Damian, Erik D. Demaine, Robin Flatland, Stefan Langerman, Joseph O'Rourke, Suneeta Ramaswami, Vera Sacristán, and Stefanie Wuhrer. Linear reconfiguration of cube-style modular robots. *Computational Geometry*, 42(6):652–663, 2009. doi:10.1016/j.comgeo.2008.11.003.

5 Adrian Dumitrescu and János Pach. Pushing squares around. *Graphs and Combinatorics*, 22:37–50, 2006. doi:10.1007/s00373-005-0640-1.

6 Robert Fitch, Zack Butler, and Daniela Rus. Reconfiguration planning for heterogeneous self-reconfiguring robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and System (IROS 2003)*, volume 3, pages 2460–2467, 2003. doi:10.1109/IROS.2003.1249239.

7 Ferran Hurtado, Enrique Molina, Suneeta Ramaswami, and Vera Sacristán. Distributed reconfiguration of 2D lattice-based modular robotic systems. *Autonomous Robots*, 38:383–413, 2015. doi:10.1007/s10514-015-9421-8.

8 Daniela Rus and Marsette Vona. Crystalline robots: self-reconfiguration with compressible unit modules. *Autonomous Robots*, 10:107–124, 2001. doi:10.1023/A:1026504804984.