# Optimal In-Place Compaction of Sliding Cubes

**Irina Kostitsyna**
TU Eindhoven, The Netherlands

**Tim Ophelders**
Utrecht University, The Netherlands
TU Eindhoven, The Netherlands

**Irene Parada**
Universitat Politècnica de Catalunya, Barcelona, Spain

**Tom Peters**
TU Eindhoven, The Netherlands

**Willem Sonke**
TU Eindhoven, The Netherlands

**Bettina Speckmann**
TU Eindhoven, The Netherlands

## Abstract

The sliding cubes model is a well-established theoretical framework that supports the analysis of reconfiguration algorithms for modular robots consisting of face-connected cubes. This note accompanies a video that explains our in-place algorithm for reconfiguration in the sliding cubes model. Specifically, our algorithm [2] reconfigures any $n$-cube configuration into a compact canonical shape using a number of moves proportional to the sum of coordinates of the input cubes. As is common in the literature, we can then reconfigure between two arbitrary shapes via their canonical configurations. The number of moves performed by our algorithm is asymptotically worst-case optimal and strictly improves upon the current state-of-the-art.

**2012 ACM Subject Classification** Theory of computation → Computational geometry

**Keywords and phrases** Sliding cubes, Reconfiguration algorithm, Modular robots

**Digital Object Identifier** 10.4230/LIPIcs.SoCG.2024.89

**Category** Media Exposition

**Related Version** *Full Version*: `https://arxiv.org/abs/2312.15096`

**Supplementary Material** *Audiovisual*: `https://www.computational-geometry.org/SoCG-videos/socg24video/optimal-in-place-compaction-of-sliding-cubes.mp4`

**Funding** *Tim Ophelders*: partially supported by the Dutch Research Council (NWO) under project no. VI.Veni.212.260.
*Irene Parada*: is a Serra Húnter Fellow. I.P. was supported by grant 2021UPC-MS-67392 funded by the Spanish Ministry of Universities and by the European Union (NextGenerationEU) and partially supported by grant PID2019-104129GB-I00 funded by MICIU/AEI/10.13039/501100011033.

## 1 Introduction

Modular robots consist of a large number of comparatively simple robotic units. These units can attach to and detach from each other, move relative to each other, and in this way form different shapes or configurations. This shape-shifting ability allows modular robots to robustly adapt to previously unknown environments and tasks. In our video, we illustrate an algorithm in the *sliding cubes model*, a well-established theoretical framework that supports the analysis of reconfiguration algorithms for modular robots consisting of face-connected cubes. In this model, a *configuration* $\mathcal{C}$ consists of a set of face-connected cubes. The model allows two types of moves: *slides* and *convex transitions* (see Figure 1). A move that operates on a cube $c$ is valid only if $\mathcal{C} \setminus \{c\}$ is connected.


**Figure 1** Moves in the sliding cube model. Left: Before the moves. Right: After the moves.

Until recently, the most efficient algorithm for the reconfiguration problem in 3D was the algorithm by Abel and Kominers [1], which uses $O(n^3)$ moves to transform any $n$-cube configuration into any other $n$-cube configuration. As is common in the literature, this algorithm reconfigures the input into an intermediate canonical shape. Stock et al. [5] recently announced a worst-case bound of $O(n^2)$ moves for the Abel and Kominers algorithm. Furthermore, their paper presents an in-place reconfiguration algorithm, which runs in time proportional to a measure of the size of the bounding box times the number of cubes. Specifically, their algorithm requires $O(n(wd + h))$ moves in the worst-case, where $w$, $d$, and $h$ are the width, depth, and height of the bounding box, respectively.

In our video, we illustrate our novel in-place algorithm that reconfigures any $n$-cube configuration into a compact canonical shape using a number of moves proportional to the sum of coordinates of the input cubes [2, 3]. This number of moves is asymptotically worst-case optimal. Furthermore, our algorithm directly extends to squares in two dimensions and to hypercube reconfiguration in dimensions higher than three.

**Video.** Reconfiguration algorithms for squares or cubes involve many (literally) moving pieces. As such, text and static images are not sufficient to convey an intuitive sense of their inner workings. Furthermore, checking the correctness of such algorithms on paper only is a challenging task which often fails to detect crucial edge cases. For these reasons, we implemented our algorithm in SquareSlider [4] after extending the tool with 3D capabilities.

Our implementation simply checks for every cube it if allows one of the operations of the algorithm in order (according to their alphabetical name, see the full version of the paper [2]). SquareSlider computes a complete move sequence to compact a starting configuration; this move sequence can be exported. Most animations in our video have been created by importing the move sequences from SquareSlider into the open-source 3D software package Blender.[1]

The video provides a high-level overview of the various operations and definitions used in our algorithm. Furthermore, it explains how these operations together ensure that every configuration can be compacted and it sketches the analysis of the necessary number of moves. The video roughly follows the outline of our algorithm given in Section 2 below.

[1] `https://blender.org`

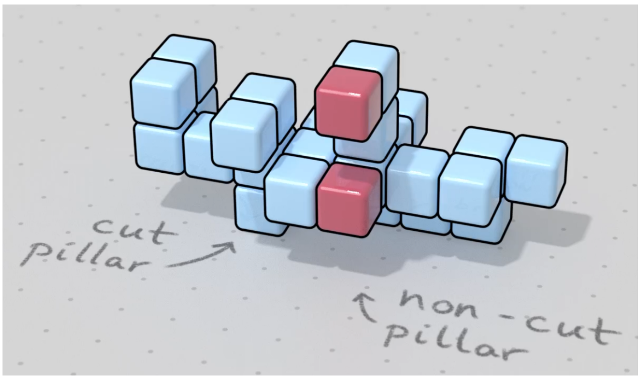


**Figure 2** Example of a cube reducing $Z_{\mathcal{C}}$: the bottom red cube moved down.

## 2 Algorithm

The aim of our algorithm is to compact a configuration $\mathcal{C}$. Call a cube $c = (x, y, z)$ *finished* if the cuboid spanned by the origin and $c$ is completely in $\mathcal{C}$, that is, if $\{0, \ldots, x\} \times \{0, \ldots, y\} \times \{0, \ldots, z\} \subseteq \mathcal{C}$. We call $\mathcal{C}$ finished if all cubes in $\mathcal{C}$ are finished. For a set of cubes $S \subseteq \mathcal{C}$, let $(X_S, Y_S, Z_S)$ denote its coordinate vector sum $\sum_{(x,y,z)\in S}(x, y, z)$. We aim to minimize a *potential function* $\Pi_{\mathcal{C}}$, which is defined in such a way that $\Pi_{\mathcal{C}} = O(X_{\mathcal{C}} + Y_{\mathcal{C}} + Z_{\mathcal{C}})$. We call a configuration $\mathcal{C}$ *non-negative* if every cube $c \in \mathcal{C}$ has non-negative coordinates, and we say that a sequence of $m$ moves is *safe* if the result is a non-negative instance $\mathcal{C}'$, such that $\Pi_{\mathcal{C}'} < \Pi_{\mathcal{C}}$ and $m = O(\Pi_{\mathcal{C}} - \Pi_{\mathcal{C}'})$. This means that the move sequence reduces the potential by at least some constant fraction of $m$ by going from $\mathcal{C}$ to $\mathcal{C}'$. We show that if $\mathcal{C}$ is unfinished, it always admits a safe move sequence.

We define four different operations, that each individually reduce the $z$-coordinates of cubes using at most three moves (see Figure 2). Next, we consider a longer move sequence that moves a complete set of vertically connected cubes to a different $x$- or $y$-coordinate, which we call a *pillar shove* (see Figure 3, left). Lastly, we define an operation that performs any move of $\mathcal{C}$ that reduces the potential, and results in a non-negative instance. For a complete definition of these operations, see the full version of this paper.

**Low and high components.** Suppose the previous operations do not apply. We define *low components* as connected components on the ground layer, and *high components* as connected components in the rest of the configuration (see Figure 3, right). We call a low component $L$ *clear* if $\mathcal{C} \setminus L$ is connected, and $L$ contains no cube $c = (0, 0, 0)$ at the origin. A clear low component $L$ is always connected to a high component via a vertically connected set of cubes, which we call its *clearing pillar*. The full version proves that such a clear low component always exists.

Let $L$ be a clear low component, and assume that $L$ is too small (has too few cubes) to reach the origin. We would like to perform a pillar shove on the clearing pillar $P$. However, it could be that moving $P$ would disconnect $L$. For this specific case we define one last operation. This operation gathers cubes from $L$ towards $P$ to preserve connectivity with the low component while performing a pillar shove on $P$.

This last operation moves cubes that are not part of the clearing pillar. However, the move is still safe, since it is only performed on low components that are too small to reach the origin. This algorithm terminates when no high component remains, or there is at most one high component, which consists entirely of finished cubes.

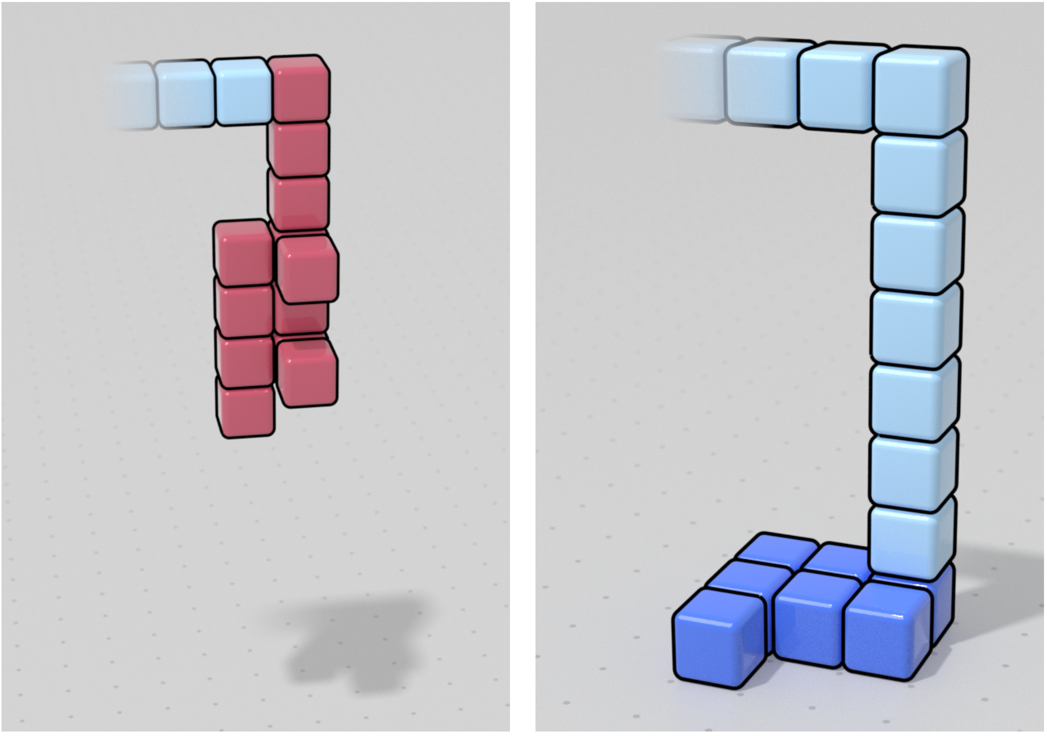

**Figure 3** Left: a pillar shove in progress. Right: a low component (dark blue) connected to a high component (light blue).

All of the operations work not only in 3D, but also in 2D where instead of prioritizing reducing the $z$-coordinate, we prioritize reducing the $y$-coordinate without increasing the $z$-coordinate. Moreover, these moves never move the origin. Therefore, we can now run the exact same moves on the bottom layer in 2D, until the bottom layer is finished. If there is still a high component, it stays connected via the origin. We end up with a finished configuration.

Since each of the operations used in our algorithm is safe, the total number of moves performed is $O(\Pi_{\mathcal{C}}) = O(X_{\mathcal{C}} + Y_{\mathcal{C}} + Z_{\mathcal{C}})$.

## References


1. Zachary Abel and Scott Duke Kominers. Universal reconfiguration of (hyper-)cubic robots. *arXiv e-Prints*, 2011. `arXiv:0802.3414`.
2. Irina Kostitsyna, Tim Ophelders, Irene Parada, Tom Peters, Willem Sonke, and Bettina Speckmann. Optimal in-place compaction of sliding cubes. *arXiv e-Prints*, 2023. `arXiv:2312.15096`.
3. Irina Kostitsyna, Tim Ophelders, Irene Parada, Tom Peters, Willem Sonke, and Bettina Speckmann. Optimal in-place compaction of sliding cubes. In *Proc. 19th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT)*, 2024. To appear.
4. Willem Sonke and Jules Wulms. An interactive framework for reconfiguration in the sliding square model (Media Exposition). In *38th International Symposium on Computational Geometry (SoCG)*, 2022. `doi:10.4230/LIPIcs.SoCG.2022.70`.
5. Frederick Stock, Hugo Akitaya, Matias Korman, Scott Kominers, and Zachary Abel. A universal in-place reconfiguration algorithm for sliding cube-shaped robots in quadratic time. In *Proc. 40th International Symposium on Computational Geometry (SoCG), LIPIcs*, volume 293, pages 1:1–1:14, 2024.