3D Staged Tile Self-Assembly

Arne Schmidt[1]

1 **Department of Computer Science, TU Braunschweig, Germany.**
aschmidt@ibr.cs.tu-bs.de

Abstract

In the staged self-assembly model, building blocks, called tiles, are added to bins in stages. This results in multiple subassemblies attaching to each other at each stage. Previous work by Demaine et al. show that any polyomino can be created within $O(\log^2 n)$ stages. We improve this to $O(\log n)$ stages and also show that for any three-dimensional monotone polycube there is a staged self-assembly system with $O(\log n)$ stages, $O(1)$ glue types and a scale factor of five constructing this shape.

1 Introduction

In 1998, Erik Winfree [11] introduced the abstract Tile Self-Assembly Model (aTAM) in which unrotatable building blocks (called Wang tiles [10]) with specific *glue types* on their sides successively attach to a given tile, called *seed*. Each glue type has a strength and a tile can attach to an existing assembly through a matching glue type if the sum of corresponding glue types is at least a threshold value τ, called *temperature*.

Ten years later, Demaine et al. [3] introduced the staged Tile Self-Assembly Model (sTAM). Here, not only tiles but whole subassemblies can stick to each other rather than only attaching to a seed. In the sTAM, the assembly process is split up into several phases, called *stages*. In each stage, multiple subassemblies are created independently in various *bins*. The results can then be mixed together in a next stage and unwanted assemblies can be filtered out. Two subassemblies attach to each other if the sum of strengths of all matching glue types along the common boundary exceeds the temperature.

Demaine et al. [4] showed that, in contrast to aTAM, where a line can only be built with $\Omega(N)$ glue types, any shape can be built within $O(\log^2 n)$ stages using $O(1)$ glue types and $O(k)$ bins in sTAM. Here, N is the number of tiles of the shape, n is the side length of a smallest enclosing square, and k is the number of corners of the shape.

1.1 Our Contribution

We prove that there are staged self-assembly systems using $O(\log n)$ stages, $O(1)$ glue types, an $O(1)$ scale factor, temperature $\tau = 1$, and full connectivity to assemble (i) any two-dimensional shape using $O(k)$ bins, and (ii) any monotone three-dimensional shape using $O(n)$ bins.

1.2 Related Work

In the past few years, many models have been developed on top of the aTAM. Demaine et al. [3] introduced the two-handed assembly model (2HAM) in which two partial assemblies can bind to each other; from this model they developed the sTAM. Padilla et al. [5] introduced a hierarchical system based on signal passing, i.e., glues are inactive until activated by a signal.

Stages are also considered in the literature: Reif [7] uses a step-wise model for parallel computing, Park et al. [6] assemble DNA lattices with a hierarchical assembly technique and Somei et al. [9] use a step-wise assembly of DNA tiles. However, non of these works consider complexity aspects. In a recent paper, Chen and Doty [2] study the effects of parallelism in

35th European Workshop on Computational Geometry, Utrecht, The Netherlands, March 18–20, 2019.
This is an extended abstract of a presentation given at EuroCG'19. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear eventually in more final form at a conference with formal proceedings and/or in a journal.

hierarchical systems. In contrast to these works, Chalk et al. [1] consider the optimal stage complexity for a fixed number of bins and glue types in the sTAM. They show that for b bins and t tile types $O(\frac{K(S)-t^2-bt}{b^2} + \frac{\log\log b}{\log t})$ stages are sufficient, where $K(S)$ is the Kolmogorov complexity of a shape S. However, their technique uses a large scale factor; we use a scale factor of only five.

2 Definitions

Due to space constraints we omit the formal definition of an sTAM, which can be found in [3]. **Polyominoes.** A *polyomino* P is a union of unit squares, called *pixels*, joined edge to edge (see Fig. 1). W.l.o.g., we assume that the pixels are centered at points from $\mathbb{Z}^2$. A pixel is a *boundary pixel* if at least one of the eight (axis parallel or diagonal) neighbor positions is not occupied by a pixel in P. A boundary pixel p is a *corner* pixel of P if there are no two neighbor pixels p' and p'' that are collinear with p and that are boundary pixels. The *corner set* of a polyomino is the set of all corner pixels along all boundaries.

We define k as the cardinality of the corner set, N as the number of pixels of P, and n the side length of a bounding square. We consider the following metrics:

Stage Complexity: The number of stages needed to assemble the shape.

Bin Complexity: The maximum number of different bins used at the same time.

Glue Complexity: The number of different glue types.

Scale Factor: A scale factor c replaces each tile of a polyomino by a $c \times c$ supertile.

Full Connectivity: A polyomino is called *fully connected* if and only if there is a matching glue type between each pair of adjacent tiles.

Polycubes and the corresponding metrics are defined analogously.

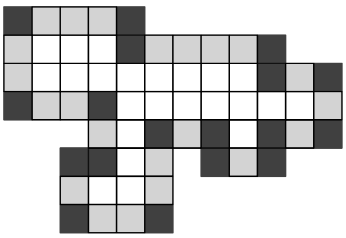

Figure 1 A polyomino with $N = 63$ tiles, its $k = 18$ corner pixels (dark gray), and its ordinary boundary pixels (light gray). A smallest square bounding the polyomino has side length $n = 12$.

3 Two Dimensions

We describe a staged tile self-assembly system that constructs an arbitrary two-dimensional shape within $O(\log n)$ stages using only $O(1)$ glue types and an appropriate scale factor. This is an improvement of a factor of $O(\log n)$ compared to Demaine et al. [4]. However, the construction is similar.

Consider a polyomino P^5, which is the result of scaling a polyomino P by five. We define the *backbone* similar to the backbone in [4]: First, we add all boundary pixels of P^5 to the backbone. Furthermore, we add *tunnels* starting at the leftmost bottommost boundary pixel p belonging to a hole. The tunnel extends two lines to the left; one at p and one two rows above p. We stop extending as soon as we reach another boundary pixel. Having a tunnel we remove the boundary pixels enclosed by the tunnel (see Fig. 2 left). Note that special cases may occur when connecting a tunnel to the boundary (see Fig. 2 right). This construction gives a single cycle visiting all holes, which is easy to construct.

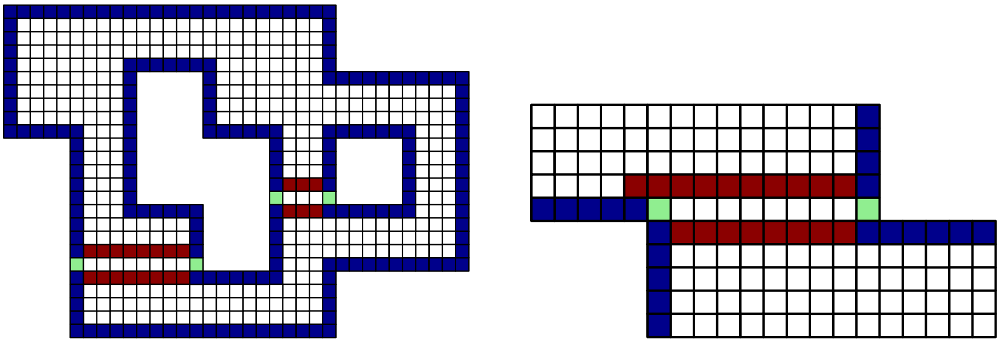

Figure 2 Left: Example of a backbone (blue and red). Blue pixels denote boundary pixels, red pixel represent tunnels. The green pixel get removed from the backbone. Right: A special case that may occur during construction of the backbone: The tunnel (red) reaches a corner. One side of the tunnel connects to the first pixel after the corner.

▶ **Theorem 1.** *The backbone B_P of a polyomino P is constructible within $O(\log n)$ stages using $O(k)$ bins and three glue types at temperature $\tau = 1$.*

Proof. Each tile is connected to exactly two other tiles. Let p_0 be any tile of B_P and p_1, p_2 its two neighbors. Also, let a be the glue type between p_1 and p_0, and b the glue type between p_2 and p_0. We remove p_0 from B_P, obtaining a path-like structure whose ends have glue types a and b. Now, decompose this path by cutting it at a corner tile along an edge such that the number of corner tiles in both substructures are the same. The glue type between the two tiles where we made the cut will be the third glue type c. Again, we have path-like structures left, thus we can repeat the procedure. At some point we have only straight line paths left. These can be assembled with three glue types and $O(1)$ bins within $O(\log n)$ stages at temperature $\tau = 1$ (see [3]). As there are at most $O(k)$ different lines, $O(k)$ bins will be sufficient. ◀

▶ **Theorem 2.** *A polyomino P can be constructed within $O(\log n)$ stages using $O(k)$ bins, $O(1)$ glue types with full connectivity, and a scale factor five at temperature $\tau = 1$.*

Proof. The construction of an arbitrary polyomino proceeds in three steps: (1) Construct the backbone, (2) fill up missing boundary tiles and finally, (3) fill up all other tiles.

While constructing the backbone B_P we can ensure that each side of a tile in B_P pointing to the inner side of the polyomino gets a glue type corresponding to the chart in Fig. 3. For example, if the east side of a boundary tile points to the inner side, the glue type is $\{0, 1\}$, $\{5, 6\}$, $\{10, 11\}$, $\{15, 16\}$ or $\{20, 21\}$ depending on the position within the scaled tile (see Fig. 3). If a boundary tile in B_P is adjacent to a missing boundary tile (green tile in Fig. 2), then we set the glue type as shown in Fig. 4, increasing the number of glue types by seven.

After constructing the backbone with all glues needed, we add the missing boundary tiles in a single stage. In another stage, we can now fill up the whole polyomino with the tiles shown in Fig. 3. Because any side facing to a hole or to the outer face has no glue, no tile can be placed to a position not belonging to the polyomino.

We conclude: For phase (2) and (3) we need two stages and $O(1)$ glue types. Thus, in total we have $O(\log n)$ stages, $O(1)$ glue types, scale factor five, $O(k)$ bins, and full connectivity at temperature $\tau = 1$. ◀

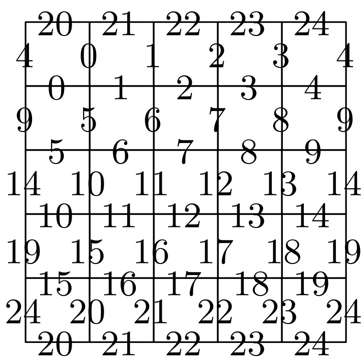

Figure 3 Glue types for tiles in a scaled tile; these are 25 different types.

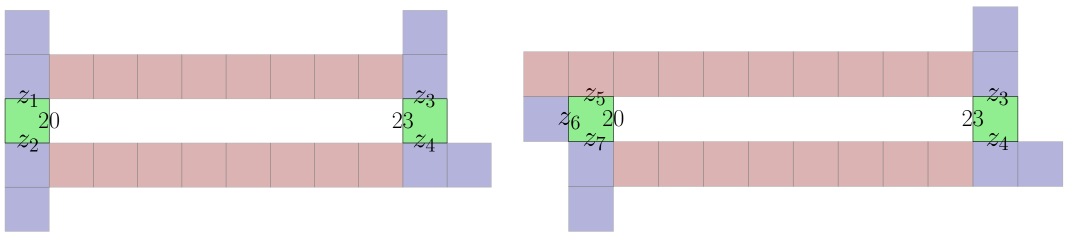

Figure 4 Glue types for tiles on the boundary missing in B_P.

4 Monotone, Three-Dimensional Shapes

Most assembly systems use the fact that the third dimension can be used to combine two partial assemblies. However, this cannot be easily done while constructing three-dimensional shapes, unless we can make use of a fourth dimension. Therefore, we need staged assembly systems that only combine partial three-dimensional assemblies for which we have an obstacle-free path. Previous work [8] considers assembly of polyominoes with straight free paths. In this paper we only consider a special case of three-dimensional shapes: z-monotone polycubes.

▶ **Definition 3.** A polycube P is z-monotone if the intersection of any x-y-plane with P results in a two-dimensional connected polyomino. We call such an intersection a *layer* of P. See Fig. 5 for an example.

The idea to construct three-dimensional monotone shapes is to construct *slices* that are scaled by a factor of five, having *plugs* and *sockets* (similar to tabs and pockets for two-dimensional shapes in [3]). Consider a polycube P scaled by a factor of five resulting in P^5. We enumerate each layer of P^5 bottom to top modulo five. For each block of five layers,

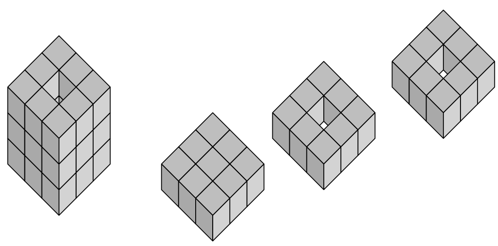

Figure 5 Left: A three-dimensional z-monotone polycube. Right: Three layers of P.

we create two polycubes as follows: P_1 gets layers 0 and 1 completely, and additionally the boundary tiles of layer 2, which form a socket (brown in Fig. 6). If there is a layer -1 below layer 0, then we form a plug with the tiles from layer -1. Let $L_{\text{-}1}$ be the tiles in layer -1 having a neighbor in layer 0. We add all tiles of $L_{\text{-}1}$ to P_1 except its boundary tiles (turquoise in Fig. 6).

The second polycube, P_2, gets layer 3 and 4 completely, and additionally all tiles of layer 2 except its boundary tiles, which form a plug (turquoise in Fig. 6). If there is a layer above layer 4, namely layer 5, then we form a socket. Let L_4 be the set of tiles of layer 4 having a neighbor in layer 5. Then we remove all tiles except the boundary pixels of L_4 from P_2 (brown in Fig. 6).

Performing this procedure for each block of five layers results in a decomposition into slices (an example is given in Fig. 6). Before we start describing how to assemble the slices, we explain how the slices can be used to get the final shape (note that this construction is similar to a rectangle decomposition in 2D [4]). We cut between two slices that split the polycube in half. We give both, the plug and the socket, a set of glue types, say G_1 (see Fig. 7 left). At the next cut we use the set of glue types G_2. Let G_3 be the third set of glues. At each recursion step, any subpolycube $\tilde{P}$ uses two of the three sets. We cut between two slices in $\tilde{P}$ and use the third set of glues for the new plug and the socket. This gives us a decomposition tree with logarithmic height and linear width, thus we need $O(\log n)$ stages and $O(n)$ bins.

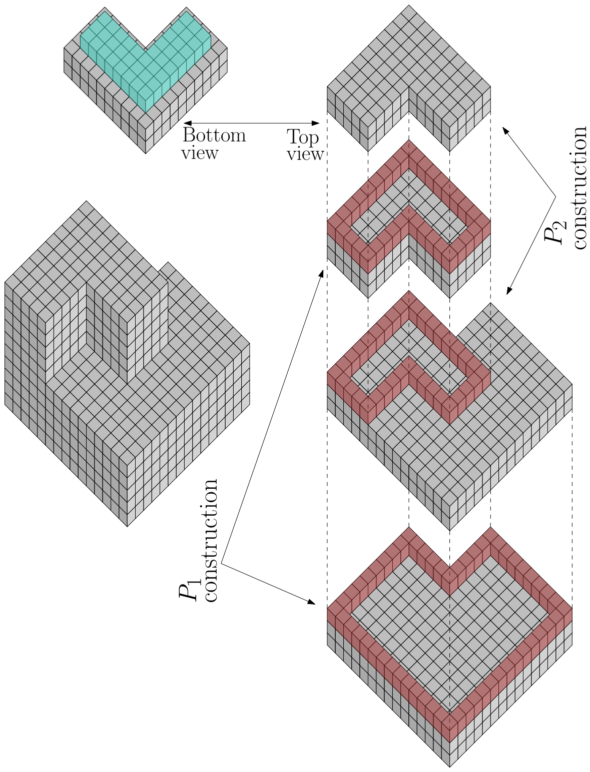

Figure 6 Decomposition of a polyomino (left) into slices (right). Slices have sockets (brown tiles) on the top and plugs on the bottom side (turquoise tiles).

▶ **Lemma 4.** *A slice can be constructed within $O(\log n)$ stages using $O(1)$ glue types, $O(k)$ bins with full connectivity at temperature $\tau = 1$.*

Proof. For each slice we have at least one layer with all tiles, one socket on the top and a plug on the bottom. Consider the tiles T_p we removed to obtain the plug, i.e., the tile of the socket of

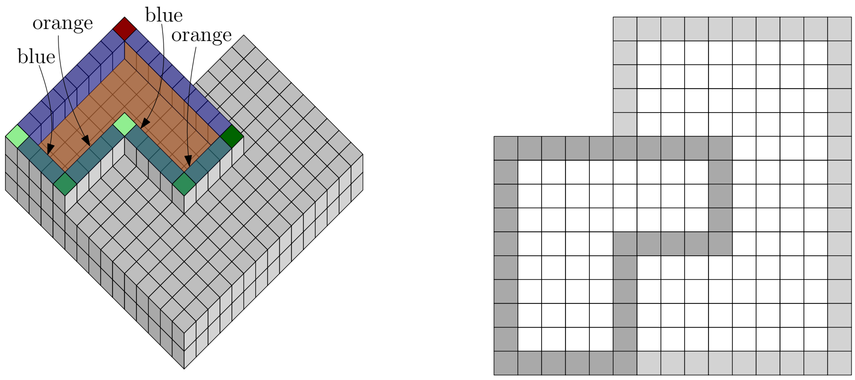

Figure 7 A slice with $O(1)$ glue types for the socket and its 2D projection.

the slice below the current slice. Furthermore, let T_s be the set of tiles of the socket. If we look at a 2D projection of T_p, T_s and a complete layer P_L, i.e., we ignore the z-direction, then at least one of T_p or T_s is the same as the boundary tiles T_L of P_L.

W.l.o.g., let T_s be unequal to T_L. Due to space constraints we only consider the case where P_L has no holes and T_s and T_L have common tiles (in other cases we can make use of tunnels like in Section 3). Let $\tilde{P}$ be the polyomino obtained by projecting T_s, T_p and T_L onto the x-y-plane (see Fig. 7 right). The tiles of $\tilde{P}$ can be divided into three types of tiles: (i) tiles only belonging to T_L, (ii) tiles only belonging to T_s, and (iii) tiles belonging to both, T_L and T_s. We can decompose $\tilde{P}$ by cutting off type (i) and type (ii) tiles with $O(\log n)$ cuts first, such that only lines with tiles of type (i), (ii) or (iii) remain. These lines can be built within $O(\log n)$ stages with correct glue types on their sides.

We can now fill up the slice with tiles that have glues on the top and bottom encoding (1) the glue type of the socket above, (2) the glue type of the plug below the current pixel, and (3) the distance to the top/bottom of the slice. Because there are only a constant number of such combinations possible, also a constant number of glue types will be sufficient. It remains to differentiate between tiles that have the socket above and those that do not have the socket above. We use 25 glue types to fill up the area, that is enclosed by type (ii) and (iii) tiles, as shown in Section 3. Another 25 glue types can be used to fill up the remaining area. In total these are $O(1)$ glue types. ◀

▶ **Theorem 5.** *A z-monotone polycube can be constructed within $O(\log n)$ stages using $O(1)$ glue types, $O(n)$ bins with full connectivity, and a scale factor five at temperature $\tau = 1$.*

5 Conclusion and Future Work

In this paper we started the first investigation of three-dimensional shapes within the staged self-assembly model. We showed that monotone three-dimensional shapes can be assembled within $O(\log n)$ stages, $O(1)$ glue types, $O(n)$ bins using full connectivity and a scale factor of 5. Future work could consider to look at upper and lower bounds for the stage complexity. Is it still possible to assemble an arbitrary polycube within $O(\log n)$ stages and only $O(1)$ glue types?

References

1 C. Chalk, E. Martinez, R. Schweller, L. Vega, A. Winslow, and T. Wylie. Optimal staged self-assembly of general shapes. *Algorithmica*, pages 1–27, 2016.

2 H.-L. Chen and D. Doty. Parallelism and time in hierarchical self-assembly. *SIAM Journal on Computing*, 46(2):661–709, 2017.

3 E. D. Demaine, M. L. Demaine, S. P. Fekete, M. Ishaque, E. Rafalin, R. T. Schweller, and D. L. Souvaine. Staged self-assembly: nanomanufacture of arbitrary shapes with O(1) glues. *Natural Computing*, 7(3):347–370, 2008.

4 E. D. Demaine, S. P. Fekete, C. Scheffer, and A. Schmidt. New geometric algorithms for fully connected staged self-assembly. *Theoretical Computer Science*, 671:4–18, 2017.

5 J. E. Padilla, W. Liu, and N. C. Seeman. Hierarchical self assembly of patterns from the robinson tilings: DNA tile design in an enhanced tile assembly model. *Natural Computing*, 11(2):323–338, 2012.

6 S. H. Park, C. Pistol, S. J. Ahn, J. H. Reif, A. R. Lebeck, C. Dwyer, and T. H. LaBean. Finite-size, fully addressable DNA tile lattices formed by hierarchical assembly procedures. *Angewandte Chemie*, 118(5):749–753, 2006.

7 J. H. Reif. Local parallel biomolecular computation. In *DNA-Based Computers*, volume 3, pages 217–254, 1999.

8 A. Schmidt, S. Manzoor, L. Huang, A. T. Becker, and S. P. Fekete. Efficient Parallel Self-Assembly Under Uniform Control Inputs. *IEEE Robotics and Automation Letters*, 3(4):3521–3528, 2018.

9 K. Somei, S. Kaneda, T. Fujii, and S. Murata. A microfluidic device for DNA tile self-assembly. In *DNA Computing (DNA 11)*, pages 325–335. 2006.

10 H. Wang. Proving theorems by pattern recognition—II. *Bell system technical journal*, 40(1):1–41, 1961.

11 E. Winfree. *Algorithmic self-assembly of DNA*. PhD thesis, California Institute of Technology, 1998.