Staging the Self-Assembly Process Using Morphological Information

Navneet Bhalla[1], Peter J. Bentley[2], Peter D. Vize[1] and Christian Jacob[1]

[1]University of Calgary, Calgary, AB, Canada T2N 1N4
[2]University College London, London, UK WC1E 6BT
nbhalla@ucalgary.ca

Abstract

One of the practical challenges facing the creation of self-assembling systems is being able to exploit a limited set of fixed components and their bonding mechanisms. Staging addresses this challenge by dividing the self-assembly process into time intervals, and encodes the construction of a target structure in the staging algorithm itself and not exclusively into the design of components. Previous staging strategies do not consider the interplay between component physical features (morphological information). In this work we use morphological information to stage the self-assembly process, with the benefit of reducing assembly errors and leveraging bonding mechanism with rotational properties. Four experiments are presented, which use heterogeneous, passive, mechanical components that are fabricated using rapid prototyping. Two orbital shaking environments are used to provide energy to the components, and to investigate the role of morphological information with component movement in either two or three spatial dimensions. The experiments demonstrate, as proof-of-concept, that staging enables the self-assembly of more complex morphologies not otherwise possible.

Introduction

Comprehending the principles of self-assembly has been described as one of the important aspects to understanding life (Ingber, 1998). Self-assembly is also considered to being an *enabling technology* for the creation of artificial systems (Pelesko, 2007). Constructing systems with natural characteristics (e.g. self-assembly, self-repair, and parallel construction) as a form of emergent engineering requires an understanding of the interplay between programmability/controllability and self-organisation (Doursat, 2008).

One important challenge when creating artificial self-assembling systems is caused by the use of components that lack the plasticity of biological cells. Using components that cannot differentiate results in self-assembly being constrained to a limited set of fixed components and their bonding mechanisms (Demaine et al., 2008). One strategy to address this challenge is to divide the self-assembly process into stages, referred to as *staged* or *hierarchical* self-assembly. Demaine et al. (2008) formalised the method of staging where components can be added to, or removed from, an environment at various time intervals.

Demaine et al. (2008) demonstrated the benefits of staging theoretically using abstract tiles, where staging the self-assembly process was based on the temporal aspects of conducting laboratory experiments. In contrast, we use physical components, and propose using morphological information as the dividing basis to staging the self-assembly process, inspired by biological development. Here we consider how physical features in a set of heterogeneous, passive, mechanical components can be exploited to reduce potential assembly errors, leverage rotational bonding mechanisms, and create structures with symmetrical/assymerical features. Our staging strategy is consistent with the definition of self-assembly (Whitesides and Gryzbowski, 2002), as a process involving components that can be controlled through their proper design and their environment, and where components can adjust their relative positions.

Staged self-assembly provides the advantage of encoding the construction of a target structure in the staging algorithm itself and not exclusively into the design of the components. For example, a staging algorithm can be used to reintroduce previously used components and bonding mechanisms at later time intervals, prevent the formation of holes, and create more complex morphologies that may not be otherwise possible due to shape conflicts between components.

The following section provides background material to which our staging strategy is built upon. Next, an overview of our approach is provided, including a theoretical model and physical description of the components and environments used. Four experiments follow that demonstrate the creation of self-assembled structures, from a set of components that are divided into two time intervals based on their physical features. Components are fabricated using rapid prototyping, and are placed in one of two orbital shaking environments (on a tray surface or in a jar of fluid). These two environments are used to demonstrate the role of morphological information in terms of component movement spatially in two and three dimensions (2D and 3D). We conclude by summarising how this work provides proof-of-concept evidence for staging the self-assembly process using morphological information.

Background

Biological development utilises explicit stages in its provision of a solution to the construction of multicellular organisms (Wolpert, 1998). The explicit stages in biological development are often irreversible, and cannot be repeated at later stages, such as invagination, gastrulation, and the formation of a body plan. Staged development in nature allows for the creation of more complex phenotypes, which otherwise would not be possible (Wolpert, 1998).

A challenge towards the creation of self-assembling systems is the use of fixed components in contrast to components that can differentiate and communicate (e.g. cells in biological organisms). *DNA nanotechnology* is one example of an application area using fixed components, such as DNA tiles (using interwoven double-stranded DNA to create the body of a tile, and single DNA strands extending from the edges of a tile's body; Winfree et al., 1998). The staged Tile Assembly Model (sTAM) addresses this challenge by incorporating the temporal aspects of conducting laboratory experiments, using DNA tiles for example, into the self-assembly process (Demaine et al., 2008).

The sTAM is an extension to the abstract Tile Assembly Model (aTAM; Winfree, 1998). The aTAM was developed to provide a theoretical framework to investigate the assembly of square tiles (based on DNA tiles) in a square lattice environment. A tile type is defined by the bonding domains on the North, West, South, and East edges of a tile. At least one seed tile must be specified to start the self-assembly process. Tiles cannot be rotated or reflected. There cannot be more than one tile type that can be used at an assembly location in the growing structure. Tile types are in infinite supply, of equal concentration, in the model. All tiles are added to the same environment, *one-pot-mixture*. Tiles can only bond together if the interactions between them meet or exceed the *temperature* parameter. As a result, temperature dictates *co-operative bonding*. The seed tile is first placed in the environment, and additional tiles are added one at a time if the bonding constraints are satisfied.

The sTAM extends the aTAM by dividing the self-assembly process into time intervals. Components can be added to, or removed from, as set of environments, mirroring the laboratory operations of adding/filtering DNA-based components to solutions that can be mixed together. The sTAM has been used to investigate the algorithmic construction of structures, such as a fully connected n×n square ($n \in \mathbb{N}$). The construction of a square is problematic, as assembling tiles must be coordinated to prevent the occurrence of holes. The sTAM has shown an algorithmic efficiency with minimal tile sets and bonding mechanisms (not requiring co-operative bonding, at temperature one) in the construction of such structures. This efficiency is due to staging, and is an advantage over the aTAM itself that relies on co-operative bonding (Rothemund and Winfree, 2000), or other extensions to the aTAM that use either changes in temperature (Kao and Schweller, 2006) or by varying the concentration of tiles (Adleman et al., 2001; Doty, 2009).

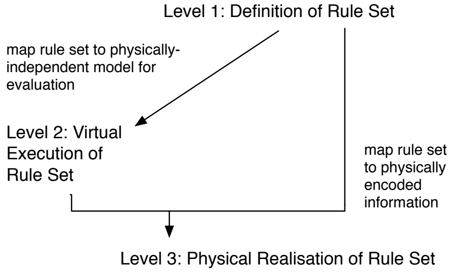

Level 1: Definition of Rule Set
map rule set to physically-independent model for evaluation
Level 2: Virtual Execution of Rule Set
map rule set to physically encoded information
Level 3: Physical Realisation of Rule Set

Figure 1: Three-level approach to self-assembly design.

Situated development is another method investigating staged construction, where artificial evolution was used to evolve the assembly plan of a structure (Rieffel and Pollack, 2005). Based on rapid prototyping, assembly plans were evolved using *permanent* and *temporary* components which were "dropped" in an environment. Temporary components act as scaffolding and can be removed (representing how support material can be removed in rapid prototyping).

In contrast to Demaine et al. (2008) and Rieffel and Pollack (2005), physical examples of staged self-assembly include Wu et al. (2002) where templates were used to self-assemble spherical beads into substructures with specific patterns (e.g. linear, triangular, and hexagonal shapes). As well, He et al. (2008) used three-point start motif tiles to self-assemble tetrahedrons, dodecahedrons, and buckyballs by controlling the motif length and concentration of tiles in a two-step process. Despite this work, there is little (if any) literature that describes the use of morphological information to stage the self-assembly process.

Staging and the Three-Level Approach

The three-level approach provides a high-level description to designing self-assembling systems via physically encoded information (Bhalla et al., 2010). The three levels include: (1) definition of rule set, (2) virtual execution of rule set, and (3) physical realisation of rule set (Fig. 1). Here we extend the three-level approach to incorporate our staging strategy. At level one, a new self-assembly rule is introduced to specify which components are present at a particular time interval. To accommodate this new rule, an extension to a self-assembly model based on the aTAM is provided at level two. Finally, physical features of components that are exploited in our staging experiments is described at level three.

Level One: Definition of Rule Set

A system is described by three categories of self-assembly rules, *component*, *environment*, and *system*, which are in the context of component movement spatially in 2D or 3D.

Component rules specify shape and information. Conceptually similar to DNA tiles, components are either squares

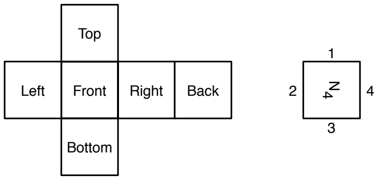


Figure 2: 3D component spatial relationship, and an example of information orientation on a 3D component's face.

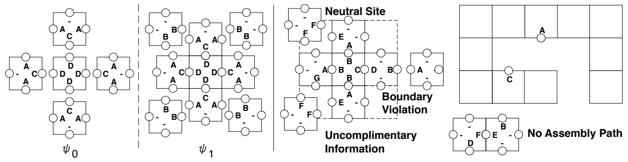


Figure 3: 2DscTAM example, and 2D assembly violations.

(2D) or cubes (3D). Each edge/face of a component serves as an information location (Fig. 2), in either a four-point (Top-Left-Bottom-Right) or six-point arrangement (Top-Left-Bottom-Right-Front-Back). Information is represented by a capital letter (A to H for 2D components, and I to T for 3D components). A subscript (1 to 4) is used with each capital letter (e.g. N_4) to indicate orientation on a 3D component's face. The dash symbol ($-$) represents a neutral site (where no assembly information is present). The spatial relationship of a component's information defines its type.

Environment rules specify environmental conditions such as temperature (ϕ) and *boundary* constraints. An assembly protocol must at least meet the temperature for assembly bonds to occur. The boundary confines components to the environment. Components are permitted to translate and rotate in 2D and 3D systems. In addition, components have rotational information and can be reflected in 3D systems.

System rules specify component type frequency in each time interval (ψ), and two interaction rules (*fits* and *breaks*). Time intervals indicate when components are added to a single environment (e.g. ψ_0; using a subscript 0 to n, where n $\in \mathbb{N}$ and 0 indicates the start of the self-assembly process). If two complementary pieces of information come into contact, (e.g. A fits B), it will cause them to assemble. This rule type is commutative (e.g. if A fits B, then B fits A). Furthermore, fits rules encapsulate component-to-component rotational interactions in 3D systems. A subscript (360, 180, and 90) is used to represent if the faces of complementary 3D components can fit together in four, two, or in one way respectively (e.g. M fits_{180} N). If two assembled pieces of information experience a temperature of two (ϕ_2), then their assembly breaks. The system rules in conjunction with their physical counterparts is provided at the end of this section, *Level Three: Physical Realisation of Rule Set*.

Level Two: Virtual Execution of Rule Set

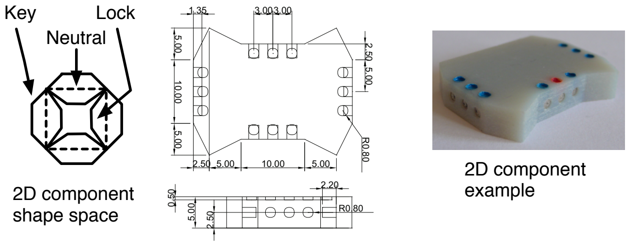
At level two, a self-assembly rule set is mapped to an abstract tile model for computational efficient evaluation, and is used to determine if physical evaluation of a self-assembly rule set is applicable at level three. We extend the concurrent Tile Assembly Model (cTAM; Bhalla et al., 2010) to incorporate staging. In contrast to the aTAM, the cTAM is better suited to the type of self-assembling systems used here by allowing multiple substructures to self-assemble concurrently, not using seed tiles, permitting more than one tile type to be used at an assembly location, and requiring all tiles to be in the same one-pot-mixture environment. The extended cTAM is referred to as the 2D and 3D staged concurrent Tile Assembly Model (2DscTAM and 3DscTAM). Components are permitted to translate and rotate in both the 2DscTAM and the 3DscTAM, but only be reflected in the 3DscTAM.

The input into the 2DscTAM and the 3DscTAM is the number of time intervals, and the multiset of components in each interval (type and frequency). At the start of each time interval, the components corresponding to the current time interval are added to the environment (Fig. 3). A single assembly operation is applied during a time interval, initialised by selecting a single tile/substructure with an open assembly location at random. If no other tile/substructure has an open complementary information location, then the location on the first tile/substructure is labelled *unmatchable*. If there are tiles/substructures with open complementary information locations, all those tiles/substructures are put in an *assembly candidate list*. From the assembly candidate list, tiles/substructures are selected at random until a tile/substructure can be added. If no such tile/substructure can be added, due to an *assembly violation* (Fig. 3), then the location is labelled unmatchable. If a tile/substructure can be added, the open assembly locations on the two tiles/substructres are updated and labelled *match* (all applicable assembly locations, including their rotational properties in the 3D case, must match when adding two substructures). This process repeats until all assembly locations are set to either match or unmatchable. At the end of a time interval, the resulting structures are placed in a single grid environment to determine if boundary violations occur. Before starting the next time interval, all unmatchable information locations are reset. The algorithm repeats, and halts when all time intervals have been completed in sequence.

An added constraint to the 3DscTAM is that substructures (with three or more components) cannot assemble together. This constraint represents observations in preliminary physical experiments conducted by the authors.

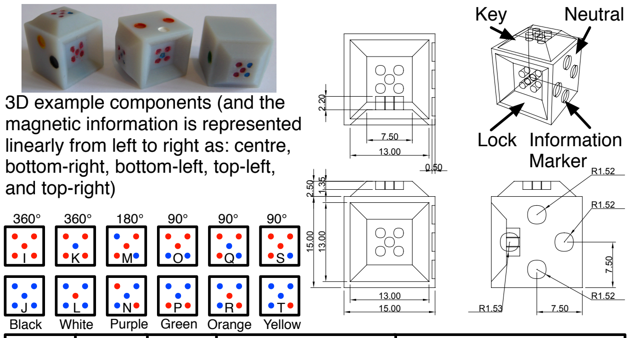
Level Three: Physical Realisation of Rule Set

Components are physically realised using rapid prototyping, at level three. Both 2D and 3D components are defined by their design space (set of physically feasible designs, Fig. 4



Shape	3-bit	Label	*Fits* Rule	*Breaks* Rule
Lock	000	A	A *fits* B → A+B	ϕ_2 *breaks* A+B → A ; B
Lock	110	C	C *fits* D → C+D	ϕ_2 *breaks* C+D → C ; D
Lock	011	E	E *fits* F → E+F	ϕ_2 *breaks* E+F → E ; F
Lock	101	G	G *fits* H → G+H	ϕ_2 *breaks* G+H → G ; H
Key	111	B	B *fits* A → B+A	ϕ_2 *breaks* B+A → B ; A
Key	001	D	D *fits* C → D+C	ϕ_2 *breaks* D+C → D ; C
Key	100	F	F *fits* E → F+E	ϕ_2 *breaks* F+E → F ; E
Key	010	H	H *fits* G → H+G	ϕ_2 *breaks* H+G → H ; G

Figure 4: 2D component specification (construction units in mm), and 2D interaction rules (where red/zero and blue/one represent magnetic south and north respectively, and '→' transition, '+' assembly, and ';' disassembly).


3D example components (and the magnetic information is represented linearly from left to right as: centre, bottom-right, bottom-left, top-left, top-right)

Shape	5-bit	Label	*Fits* Rule	*Breaks* Rule
Lock	00000	I	I *fits*$_{360}$ J → I+J	ϕ_2 *breaks* I+J → I ; J
Lock	10000	K	K *fits*$_{360}$ L → K+L	ϕ_2 *breaks* K+L → K ; L
Lock	01010	M	M *fits*$_{180}$ N → M+N	ϕ_2 *breaks* M+N → M ; N
Lock	10011	P	P *fits*$_{90}$ O → P+O	ϕ_2 *breaks* P+O → P ; O
Lock	00111	R	R *fits*$_{90}$ Q → R+Q	ϕ_2 *breaks* R+Q → R ; Q
Lock	10111	T	T *fits*$_{90}$ S → S+T	ϕ_2 *breaks* T+S → T ; S
Key	11111	J	J *fits*$_{360}$ I → J+I	ϕ_2 *breaks* J+I → J ; I
Key	01111	L	L *fits*$_{360}$ K → L+K	ϕ_2 *breaks* L+K → L ; K
Key	10101	N	N *fits*$_{180}$ M → N+M	ϕ_2 *breaks* N+M → N ; M
Key	01100	O	O *fits*$_{90}$ P → O+P	ϕ_2 *breaks* O+P → O ; P
Key	11000	Q	Q *fits*$_{90}$ R → Q+R	ϕ_2 *breaks* Q+R → Q ; R
Key	01000	S	S *fits*$_{90}$ T → S+T	ϕ_2 *breaks* S+T → S ; T

Figure 5: 3D component specification (construction units in mm), and 3D interaction rules (where red/zero and blue/one represent magnetic south and north respectively, and '→' transition, '+' assembly, and ';' disassembly).

and 5). The design space is a combination of a shape and an assembly protocol space. For both 2D and 3D components, a key-lock-neutral concept defines the shape space. A linear 3-magnetic-bit and a planar 5-magnetic-bit encoding scheme define the assembly protocol space for 2D and 3D components respectively. Magnets are placed within the edges or faces of 2D and 3D components respectively, and are not flush with a component's surface. The result of an air gap allows for adjustable component interactions and selective bonding (Whitesides and Gryzbowski, 2002). Although Miyashita et al. (2009) investigated how component shape and magnetic bonding affects the self-assembly process, they did not consider this morphological information in the context of staged self-assembly.

Here, lock-to-lock interactions can never occur due to their shape. This shape characteristic is influential in assigning 3-magnetic-bit and 5-magnetic-bit encodings to keys and locks. One magnet is placed in each position associated with a key, and two magnets are placed in each position associated with a lock. Strong bonding is ensured for key-to-lock interactions, and weak bonding between key-to-key interactions. The potential occurrence of weak bonding can be reduced with an appropriate physical temperature setting.

The four pairs of complimentary 3-magnetic-bit encodings can be optimally assigned to keys and locks to reduce assembly errors, as any key-to-lock error is at worst a one out of three match. Since this is not above a 50% match, bonding will not occur. Whereas the six pairs of unique complimentary pairs of 5-magnetic-bit encodings (accounting for planar rotation of a component's face) cannot be optimally assigned to keys and locks to reduce assembly errors. In this case, optimal assignment is considered with respect to which encodings are included to construct a target structure. It should be noted that these six encodings encapsulate rotational information for 3D component-to-component interactions, where two pairs encapsulate 360°, one pair encapsulates 180°, and three pairs encapsulate 90° rotational interactions. The 90° encodings have the potential for self-errors between complementary pairs, i.e. a three out of five match. A physical temperature to break three out of five matches, while maintaining five out of five matches, is strived for.

Orbital shakers form the environments for both 2D and 3D components. 2D components are placed on the surface of a tray, and a lid is used to prevent component reflections. 3D components are placed in a jar of mineral oil, to allow components to move freely in 3D space, and prevent oxidation affecting the magnets. The designs for both environments result from earlier experiments conducted by the authors.

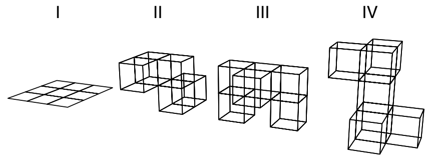


Figure 6: Four target structures for the experiments.

Experiments and Results

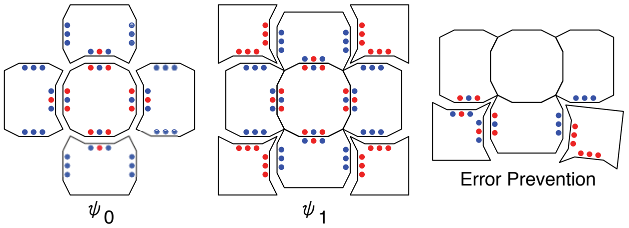
We present four experiments that were conducted to test our approach to staging the self-assembly process using morphological information. The purpose of these experiments is to demonstrate, as proof-of-concept, that staging can enable the self-assembly of closed target structures not otherwise possible. Closed refers to structures with defined boundaries (Whitesides and Gryzbowski, 2002). A target structure was assigned to each experiment (one 2D and three 3D experiments, Fig. 6). Here, the self-assembly process is staged (divided) into two time intervals, where components are only added to a one-pot-mixture environment. Component physical features, such as key and lock shapes and magnetic-bit patterns, are morphological information.

The independent variable is the use of two time intervals. The dependent variable is the resulting self-assembled structures. Enough components are supplied to create one 2D target structure and two 3D structures (due to boundary constraints of the environment). Ten trials are run for each experiment. A virtual trial (level two) is evaluated to being successful if all the potential number of target structures are achieved. A physical trial (level three) is evaluated to being successful if at least one target structure is achieved. The staging strategies and level one rules were designed by the authors. 2D and 3D experimental procedures and results are provided in terms of the three-level approach.

Two-Dimensional System

The staging strategy for creating the 2D 3×3 square target structure is to construct the *centre* and *edges* of the square in the first time interval, and construct the *corners* of the square in the second time interval (Fig. 7). In the first time interval, potential errors between the edge components can be reduced by appropriate selection of 3-magnetic-bit codes and the use of lock shapes to assemble to the centre component. The morphology of the substructure after the first time interval has corner features that can reduce assembly errors with the use of corner components that use only lock assembly shapes. The neutral edges of the corner components effectively block a corner component from assembling to the substructure in an improper orientation (Fig.7).

2D Level One Definition of Rule Set for Experiment Fig.7 provides the component rules. The control group represents components that were not divided into time intervals (non-staged). The experimental group used the same components, but divides them into two time intervals (staged). Interaction rules from Fig. 4 were applicable to both groups.



Target Structure	Staged Component Set
I	ψ_0 {1 x (D,D,D,D), 4 x (B,-,B,C)} ψ_1 {4 x (-,A,A,-)}

Figure 7: Staging strategy for target structure I, and error prevention due to shape and proper 3-magnetic-bit pattern selection (e.g. avoid magnetic repulsion configuration).

2D Level Two Experimental Setup The components from Fig.7 were mapped to an abstract representation for the 2DscTAM. Each component's shape was a unit square. The size of the environment was 10×10 units (as a representation of width×depth, and the ratio between component and environment size). A different random seed was used to initialise the 2DscTAM for each trial.

2D Level Two Experimental Results The staged components successfully created one target structure in each of the ten trials. None of the non-staged components were able to create one target structure. The unsuccessful non-sategd trials either resulted in a set of substructures (due to edge and corner components assembling in incorrect orientations), or the creation of a 3×3 open square. The results at level two were analysed using Fisher's Exact Test (one sided) for binary data (Cox and Snell, 1989). The results are statistically significant with a p-value of 0.

2D Level Three Experimental Setup A level three translation was preformed for both the staged and non-staged components (to observe the physical results of non-staged components). Components were mapped following Fig.7.

An Eden 333 Polyjet rapid prototyping machine was used to fabricate the components from Vero Grey resin. Neodymium (NdFeB) disc magnets (1/16"×1/32", diameter×radius; grade N50) were inserted into the components. Blue/red paint (north/south) marked the magnets.

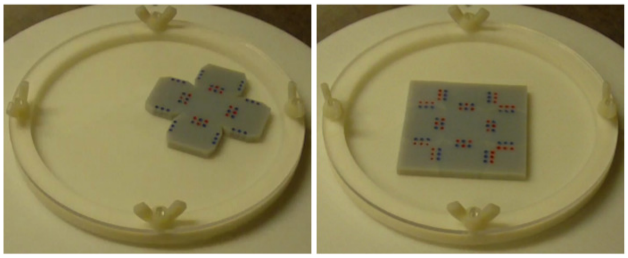
The environment size was mapped in accordance with the base component's size, to specify the dimensions of the circular tray environment. The tray was fabricated using a Dimensions Elite rapid prototyping machine, using ABS plastic (sparse-fill option was used to create a rough surface texture). The outer radius of the tray is 135 mm and the inner radius is 125 mm, while the outer wall height is 9 mm and the inner wall height is 6 mm. The tray was mounted to a Maxi Mix II Vortex Mixer (using a tray mounting bracket, also fabricated using the Dimensions printer). A tray lid was cut using a Trotec Speedy 300 Laser Engraver laser cutting machine, using 2 mm clear acrylic sheet. The tray lid was secured to the tray using polycarbonate screws and wing nuts.

Each physical trial followed seven steps (Bhalla et al., 2010). (1) Set the speed control on the Maxi Mix II Vortex mixer to 1,050 rpm. This speed created an appropriate shaking level (environment temperature) to maintain fits rules, and to mostly break partially matched magnetic codes. (2) Secure the mixer to a table, using a 3” c-clamp and six hex nuts (to help secure the c-clamp to the back of the mixer). (3) Randomly place components on the surface of the tray (trying to ensure that complementary bonding sites on the components are not in-line with each other). (4) Secure the tray lid. (5) Run the mixer for 20 minutes for a non-staged trial, or for two 10 minute intervals for a staged trial. (6) Turn the mixer off. (7) Record the state of the system, observing: the number of target structures created, the number of matching errors (between conflicting physical information, where no fits rule is applicable), and the number of assembly errors (partial attachment between corresponding physical information, where a fits rule is applicable).

2D Level Three Experimental Results The level-three results are provided in Fig. 8, with an example of the end of each time interval of a successful trial. For both component groups, no matching and assembly errors were observed in the ten trials. Only partial structures were observed, and no open 3×3 squares, were observed at the conclusion of the non-staged trials. Using Fisher’s Exact Test, this experiment is statistically significant at the 0.01 level (i.e. there is a 99% certainty the results are not due to chance).

Three-Dimensional Systems

The three 3D target structures have a three component common *core* structure, and vary in the number of *periphery* components (increasing from two, three, and four). The core structure requires two specialised 90° bonds, whereas the perimeter components only require general 360° bonds. As observed by the authors in preliminary 3D experiments, substructures consisting of at least three components are not able to assemble together. Given that the likelihood of general 360° bonds occurring is more likely than specialised 90° bonds, the staging strategy for creating the three 3D target structures is to construct the core substructure in the first time interval, and construct the periphery substructures in the second time interval (Fig. 9). The first time interval leverages the specialised component rotational information. Lock shapes for the 360° bonds are used as part of the morphology of the components in the first time interval, to reduce potential matching errors between specialised and general bonds. Furthermore, the resulting morphologies of the resulting core substructures at the end of the second time interval consist only of neutral and lock shapes, preventing assembly between the core substructures.



Target Structure I - ψ_0

Target Structure I - ψ_1

Target Structure	Group	Successful	Unsuccessful
I	staged	7	3
	non-staged	0	10

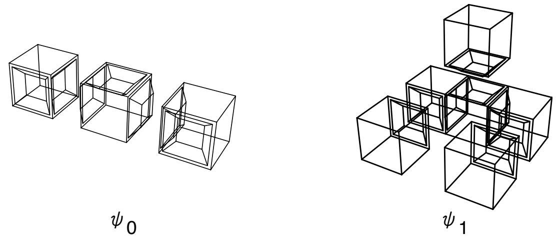
Figure 8: Successful target structure I example trial, and the number of successful/unsuccesful 2D trials.

3D Level One Definition of Rule Set for Experiments The component rules for the 3D experiments is provided in Fig. 9. Control groups and experimental groups represent non-staged and staged (using two time intervals) component sets respectively. The environment temperature was one. The interaction rules from Fig. 5 applied to both groups.

3D Level Two Experimental Setup The components from Fig. 9 were mapped to an abstract representation for the 3DscTAM. A component’s base shape was a unit cube. The size of the environment was 4×4×4 units (representing width×depth×height, and the ratio between component and environment size). A different random seed was used to initialise the 3DscTAM for each trial.

3D Level Two Experimental Results The staged components, for each experiment, successfully created two target structures in each of the ten trials. Whereas, the non-sategd components were not able to create a target structure. As expected, the unsuccessful non-sateged components resulted in substructures consisting of three components (favouring assemblies with 360° bonds) or two components. The results at level two are statistically significant with a p-value of 0 using Fisher’s Exact Test for binary data.

3D Level Three Experimental Setup As with the 2D experiment, a level-three translation was performed for both staged and non-staged components (to observe the physical results of non-staged components). Components were



Target Structure	Staged Component Set
II	ψ_0 {2 x (-,-,O_3,-,O_1,-), 4 x (-,I_1,-,-,P_1,-)} ψ_1 {4 x (J_1,-,-,-,-,-)}
III	ψ_0 {2 x (-,Q_1,-,Q_1,K_1,-), 4 x (-,-,-,K_1,R_1,-)} ψ_1 {6 x (L_1,-,-,-,-,-)}
IV	ψ_0 {2 x (T_1,-,T_4,-,-,-), 4 x (-,-,I_1,I_1,S_1,-)} ψ_1 {8 x (J_1,-,-,-,-,-)}

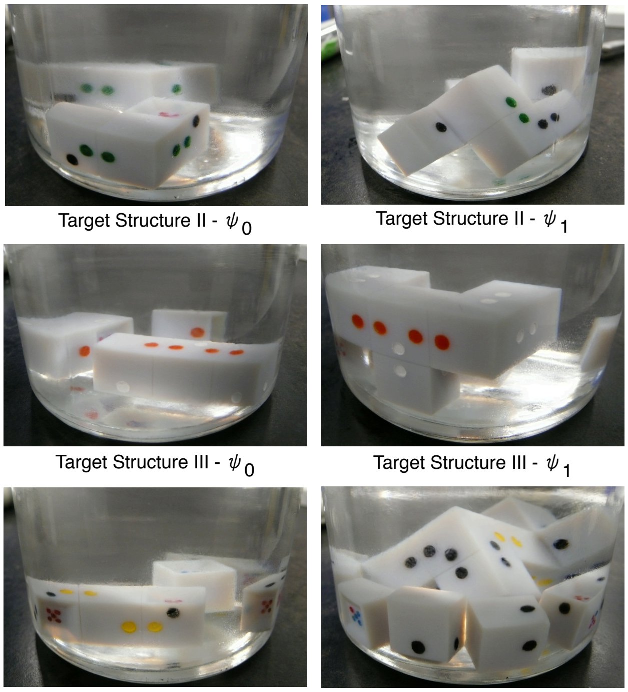
Figure 9: Staging strategy for target structure III (applicable to target structures II and IV).

mapped following Fig. 9, and were fabricated using a similar procedure as the 2D components (with the addition of colour paints to represent rotational information, Fig. 5).

500 mL clear glass, wide-mouth jars with rubber lined lids were used to contain components (91 mm×95 mm; diameter×height). A Trotec Speedy 300 Laser Engraver was used to construct the parts, using 3 mm acrylic sheet, for the jar rack. The rack was assembled using adhesive, screws, and hex nuts. The jar rack was placed on a New Brunswick Scientific Excella E1 Platform Shaker. 325 mL of Rogier Pharma light mineral oil was measured using a graduated cylinder, and poured into the jars (one for each experiment).

Each physical trial followed six steps. (1) Place three jars of mineral oil on the jar rack. (2) Randomly place the components for each experiment into the appropriate jar. (3) Secure the jar lids. (4) Turn the shaker on by setting the speed to 32.5 rpm. (5) Run the shaker for 40 minutes for a non-staged trial, or for two 20 minute intervals for a staged trial. (6) Record the state of each system, observing: the number of target structures created, the number of matching errors, the number of assembly erros, and the number of rotation errors (between complementary components).

3D Level Three Experimental Results The 3D level-three results are provided in Fig. 10, along with examples of the end of each time interval of a successful staged trial. For each experiment, no matching and assembly errors were observed in the ten trials. Rotational errors were observed in each staged experiment (Fig. 11). Using Fisher's Exact Test, the first two 3D experiments are statistically significant at the 0.05 level and the third experiment was statistically significant at the 0.50 level (i.e. there is a 95% and 50% certainty the results are not due to chance). Even though one successful staged trial was observed with the third 3D experiment, we do not consider the result statistically relevant.


Target Structure II - ψ_0
Target Structure II - ψ_1
Target Structure III - ψ_0
Target Structure III - ψ_1
Target Structure IV - ψ_0
Target Structure IV - ψ_1

Target Structure	Group	Successful	Unsuccessful
II	staged	4	6
	non-staged	0	10
III	staged	5	5
	non-staged	0	10
IV	staged	1	9
	non-staged	0	10

Figure 10: Successful target structure II, III, and IV example trials, and the number of successful/unsuccesful 3D trials.

Discussion Four experiments were conducted to demonstrate our morphological information based staging strategy. At level two, all of the staged components sets were able to achieve their respective target structures, whereas none of the non-staged components were able to. All the staged component sets, except for the third 3D experiment, were able to successfully construct their respective target structures at a statistically significant level (with 99% and 95% confidence for the 2D and 3D experiments), at level three.

One physical target structure was achieved in the third 3D experiment, and we observed in the trials a *layering effect* of components/substructures that inhibited the self-assembly of this target structure (IV). As future work, we look to

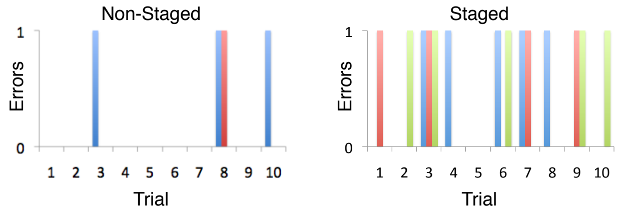


Figure 11: Rotational errors at the end of each 3D trial (target structures II blue, III red, and IV green).

build neutrally buoyant components to address this issue. We are also investigating the use of higher-order magnetic-bit codes, additional magnetic-bit patterns, and new methods for creating a more suitable physical environment temperature to prevent the occurrence of rotational errors.

An implication of staging is on the self-repairing properties of a system. Although we observed the 2D 3×3 square being able to self-repair, this was only within the second stage. Further research into features that allow for, and the understanding of the limits to, self-repair between specific stages is required to continue to further develop our approach. For example, although salamanders undergo development through unique stages, they can regrow lost limbs by repeating earlier developmental stages (Wolpert, 1998).

Nevertheless, we envision our staging strategy being applicable to a variety of applications relying on fixed components, such as the design of nano and microscale structures, circuit design, and DNA computing using self-assembly. Moreover, we envision our staging strategy as an approach to improve the ability of artificial evolution for the creation of more complex physical self-assembling systems.

Conclusions

Staging is an essential part of biological development. In this work we presented a novel approach to staging the self-assembly process using morphological information. This work involved creating two new staged self-assembly analytical tools, the 2DscTAM and the 3DscTAM. Furthermore, this work showed how the interplay between component morphological information (shape and magnetic patterns) can be used to reduce assembly errors and leverage rotational properties by using staging. We presented four proof-of-concept experiments to demonstrate that our staging strategy is a viable method for enabling the self-assembly of more complex morphologies not otherwise possible.

References

Adleman, L., Cheng, Q., Goel, A., and Huang, M.-D. (2001). Running time and program size for self-assembled squares. In Vitter, J. S., Spirakis, P., and Yannakakis, M., editors, *STOC 2001*, pages 740–748, New York, NY. ACM.

Bhalla, N., Bentley, P. J., and Jacob, C. (2010). Evolving physical self-assembling systems in two-dimensions. In Tempesti, G., Tyrrell, A. M., and Miller, J. F., editors, *Proc. of the Int. Conf. on Evolvable Systems*, pages 381–392, Berlin, Germany. Springer-Verlag.

Cox, D. R. and Snell, E. J. (1989). *Analysis of Binary Data*. Chapman and Hall/CRC, Boca Raton, FL, 2nd edition.

Demaine, E. D., Demaine, M. L., Kekete, S. P., Ishaque, M., Rafalin, E., Schweller, R. T., and Souvaine, D. L. (2008). Staged self-assembly: nanomanufacture of arbitrary shapes with o(1) glues. In Garzon, M. and Yan, H., editors, *DNA Computing*, Lecture Notes in Computer Science, pages 1–14. Springer Berlin / Heidelberg.

Doty, D. (2009). Randomized self-assembly for exact shapes. In *FOCS '09*, Washington, DC. IEEE Computer Society.

Doursat, R. (2008). Programmable architectures that are complex and self-organized: from morphogenesis to engineering. In Bullock, S., Noble, J., Watson, R., and Bedau, M. A., editors, *Artificial Life XI*, pages 181–188, Cambridge, MA. MIT Press.

He, Y., Ye, T., Su, M., Zhang, C., Ribbe, A. E., Jiang, W., and Mao, C. (2008). Hierarchical self-assembly of dna into symmetric supramolecular polyhedra. *Nature*, 452:198–201.

Ingber, D. E. (1998). The architecture of life. *Scientific American*, 278(1):48–57.

Kao, M.-Y. and Schweller, R. (2006). Reducing tile complexity for self-assembly through temperature programming. In *Proc. of the 7th Annual ACM-SIAM Symposium on Discrete Algoritms*, pages 571–580, New York, NY. ACM.

Miyashita, S., Nagy, Z., Nelson, B. J., and Pfeifer, R. (2009). The influence of shape on parralel self-assembly. *Entropy*, 11:643–666.

Pelesko, J. A. (2007). *Self Assembly: the Science of Things that Put Themselves Together*. Chapman and Hall/CRC, Boca Raton, FL.

Rieffel, J. and Pollack, J. (2005). Evolutionary fabrication: the emergence of novel assembly methods in artificial ontogenies. In *Proc. of the Genetic and Evolutionary Computation Conference, SEEDS Workshop*, pages 265–272, New York, NY. ACM.

Rothemund, P. W. K. and Winfree, E. (2000). The program-size complexity of self-assembled squares. In Yao, F. and Luks, E., editors, *STOC '01*, pages 459–468, New York, NY. ACM.

Whitesides, G. M. and Gryzbowski, G. (2002). Self-assembly at all scales. *Science*, 295(5564):2418–2421.

Winfree, E. (1998). Simulations of computing by self-assembly. In *DNA Based Computers*, volume IV.

Winfree, E., Liu, F., Wenzier, L., and Seeman, N. (1998). Design and self-assembly of two-dimensional dna crystals. *Nature*, 394(6):539–544.

Wolpert, L. (1998). *The Principles of Development*. Oxford University Press, Oxford, UK.

Wu, H., Thalladi, V. R., Whitesides, S., and Whitesides, G. M. (2002). Using hierarchical self-assembly to form three-dimensional lattices of spheres. *J. Am. Chem. Soc.*, 124:14495–14502.