Jingwen Wang, Wenjun Liu, Gene Ting-Chun Kao, Ioanna Mitropoulou, Francesco Ranaudo, Philippe Block, Benjamin Dillenburger

Multi-Robotic Assembly of Discrete Shell Structures

Abstract: Discrete shell structures are renowned for their material efficiency and elegance. However, they might require a fair amount of falsework as temporary supports during construction, which could partially offset the gains in material efficiency. This research is an investigation of the use of multi-robotic assembly for the autonomous construction of discrete shells. We present a design-to-fabrication workflow (Fig. 1) that combines structural design, construction sequence assessment, reachability evaluation, and robotic motion planning. We demonstrate the potential of the proposed workflow through a simple and a more complex example. This work contributes to the research field of assembly-aware design and multi-robotic manufacturing to improve assembly efficiency and reduce waste.

Keywords: computational design of stable discrete assemblies, robotic path planning & motion planning, assembly-aware design, design-to-fabrication workflow

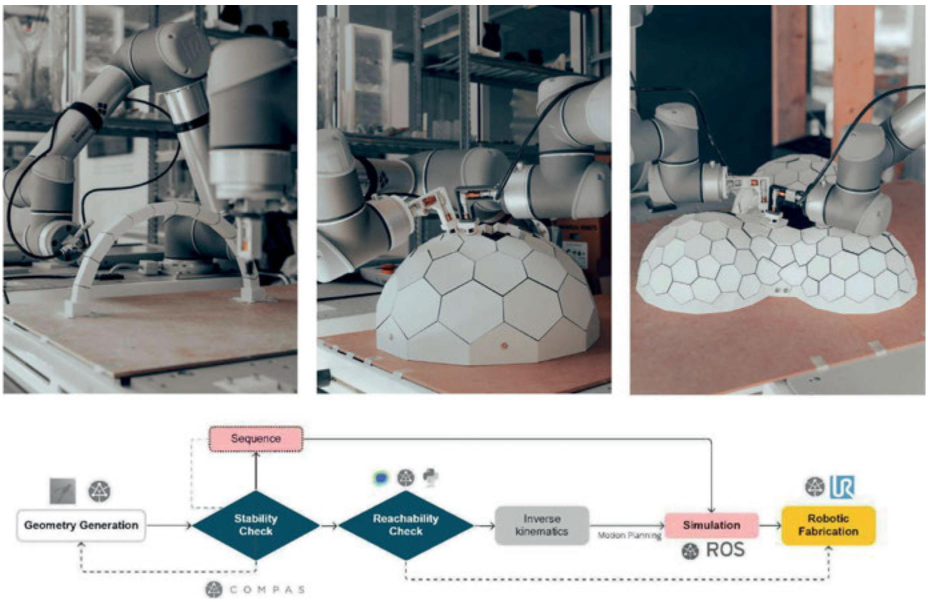

Fig. 1: A fabrication-driven workflow for robotic assembly of discrete shell structures.

https://doi.org/10.1515/9783111162683-020

1 Introduction

Shell structures are common in architecture due to their efficiency and ability to span long distances with minimal material. However, their complex geometry often requires extensive formwork, which can be labor-intensive, wasteful, and costly (Rippmann 2016). Discrete shells consist of distinct elements like blocks or bricks. Compared to continuous shells, discrete shells offer faster construction, and longer lifespan (Rippmann 2016) and can be dismantled and reassembled again. Specifically, this paper focuses on dry-jointed discrete shells, where the elements are not grouted or interlocked together but interact through unilateral (compression) contact. The Armadillo Vault serves as an example (Block et al. 2016).

Although prefabrication of discrete shell components is possible, complex falsework is often needed during assembly. Traditionally, these supports are provided as props and beams that are pre-installed on-site, registered to the correct heights and locations, and removed after decentering the shell. Techniques have been developed to reduce the need for temporary supports, such as (Ponce and Melendez 2015; Deuss et al. 2014). Such strategies, however, impose extensive geometric constraints on the final form. With the increasing use of automation in construction and advancements in computational tools for stability evaluation (see Sec. 2.2), new construction methods are now possible. Specifically, the capacity of robotic arms to move, precisely position, and hold elements in space make them particularly suitable for the fabrication of discrete assemblies.

In this paper, we outline a design-to-fabrication workflow for the autonomous assembly of dry-jointed discrete shell structures, which combines the structural design of the shell with the design of the robotic system. The remainder of this paper is structured as follows: Sec. 2.1 shows the current state-of-the-art regarding the robotic fabrication of discrete assembly; in Sec. 2.2 the main computational tools for the design of discrete shell structures are presented; in Sec. 3, the design-to-fabrication workflow is explained in detail; Design examples and results are presented in Sec. 4, and contribution and discussion are in Sec. 5.

2 State of the art

2.1 Robotic assembly of discrete structures

Fabrication systems with a single robotic arm can easily stack elements to create vertical structures such as walls and columns (Piškorec et al. 2019). However, arching and vaulted structures require more supports to ensure stability during construction. These additional supports can be provided, for example, by temporary scaffolding or by additional robotic arms. This research focuses on the latter strategy in order to remove the need for traditional falsework.

Two robotic arms assembling discrete vaulted structures were previously explored by Wu and Kilian et al. (2020) to erect a compression-only arch, and by Parascho et al. (2020) for the fabrication of a discrete vault with mortar joints. These studies demonstrate multi-robotic system's potential for scaffold-free assembly. Nevertheless, the success of the construction process is highly constrained by the capacity of the robots to access a certain space without colliding with each other, with other obstacles, or with the structure itself. Therefore, an evaluation of the reachability of the robotic system, i. e. the number of Inverse Kinematics (IK) solutions in which a robot can access a given frame in space without collisions, needs to be assessed during the design stage. Specifically, a reachability map (Makhal and Goins 2018) can be constructed to visually show the reachability of a robot for a discrete set of input frames corresponding to possible positions and orientations of the robot's end effector. However, this definition of reachability and of the associated map only provides information regarding the workspace, without specifically targeting the assembly task. This issue is further explained in Sec. 3, Step 3, where we introduce DARM, a novel way of assessing the reachability of discrete assemblies.

2.2 Parameters affecting the stability of discrete shells

Besides reachability, stability is also a crucial factor to consider during fabrication. For constructing a discrete shell assembly, two types of stability should be considered. The global stability of the assembly when finished, and its stability at each construction step. Key factors affecting the stability include:

- *Geometry of the shell:* because of the dry-jointed assumption, the geometry of the shell should follow compression-only shapes. Numerous tools are available for this task, such as RhinoVault (Rippmann 2016).
- *Thickness of the shell:* this depends on the loads acting on the structure and their distribution. Here, we assume a uniformly distributed load. We also refer to the Heyman hypotheses for masonry structures (Heyman 1995) and assume that the strength of the materials is much higher than the stresses imposed by the loads. More details regarding the determination of the correct thickness of the shell are provided in Sec. 3, Step 2.
- *Discretization:* the tessellation, the shape and inclination of the interfaces for dry-jointed discrete shells greatly affect the stability of a discrete shell, mainly during construction. However, searching for the optimal construction sequence is a challenging combinatorial problem to solve (Beyeler et al. 2015) beyond the scope of this paper. Engineering judgment can be used to select a correct assembly strategy, as shown in Kao et al. (2017).

– *Interface friction:* this parameter depends on the material chosen. Higher friction coefficients help the stability of the structure during assembly.

2.3 Stability evaluation of discrete shells

Computational tools are available to evaluate the stability of discrete shell structures both during assembly and after completion. Rigid-block Equilibrium (RBE) Whiting et al. (2009) enables fast and stable 3D discrete element equilibrium analyses, but falls short when dealing with complex interfaces and friction. Kao et al. (2022) addressed this by introducing Coupled Rigid-Block Analysis (CRA) method, which more robustly combines equilibrium and kinematics in a penalty formulation.

3 Proposed workflow

We proposed our own design-to-fabrication workflow for the autonomous assembly of dry-jointed discrete shell structures that integrates both the structural design of the shell and the robotic fabrication (Fig. 2).

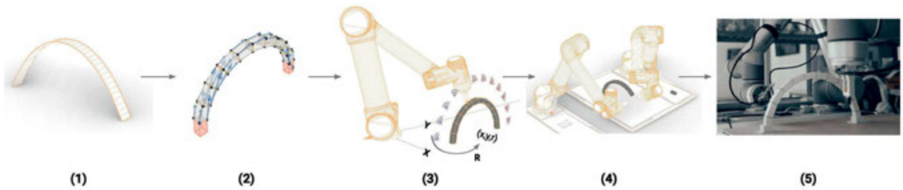

Fig. 2: Proposed workflow: (1) discrete assembly design, (2) stability evaluation, (3) robotic system design, (4) motion planning, (5) fabrication.

Step 1: Discrete assembly design

The first step of the workflow is to parametrically design the shell such that its stability can be evaluated later. The parameterization can be done with any of the tools mentioned in Sec. 2.2, and ultimately depends on the preferences of the designer.

Step 2: Stability evaluation

In this research, we propose the use of CRA and its open-source implementation (Kao 2022) to evaluate the stability.

Step 2a: Global stability. The global stability of the shell is assessed both under vertical and horizontal loads. In particular, horizontal loads are simulated by virtually tilting the structure and measuring the tilting angle: the higher the tilting angle, the more stable the structure is under horizontal actions (wind/earthquakes). The designer can determine the thickness of the shell using the information from the critical tilting angle analysis.

Step 2b: Stability during construction Once the global behavior has been analyzed, a stability analysis is conducted to determine if the assumed construction sequence is valid. For each step of the assembly sequence, CRA is used to assess if the structure is stable and where additional supports are needed. Since our system is composed of two robots, any solution that requires more than one support (the other robotic arm) or to add more than two blocks at the same time must be discarded. If a stable solution cannot be found for the given assembly sequence, the designer can change the sequence, the tessellation or the thickness of the shell. It is important to mention that, for the scope of this research, the assembly sequence is based on engineering intuition and only evaluated a posteriori (see Sec. 2).

Step 3: Robotic system design

Once the structure has been designed and the assembly sequence is validated, the robotic system can be designed, i. e. it is possible to determine the relative positions of the robots and structure. To facilitate the path planning process, we present a variation of the reachability map called *discrete assembly reachability map* (DARM). Unlike the standard reachability map (Section 2), DARM evaluates the reachability of each block of the assembly and provides combined scores that evaluate the feasibility of the entire assembly. It is worth mentioning that the results of DARM depend on the end-effector design and the type of robotic arms used.

Given an assembly in position $P_j(x, y, r)$, with x, y the location on a 2D cartesian plane and r the orientation of the assembly, consisting of n blocks , there are $K_{i,j}$ potential approaching frames for each block i and position j, where i means the index of the block and j means the index of the position. With approaching frames (Fig. 3) we refer to the potential location and orientation of the robot end-effector when placing the block i to its assembled location. The following two scores can be computed to evaluate the reachability for a given position P_j:

- The *ConfigScore* (C_j) is the number of IK solution types that can reach at least one of the approaching frames for each block of the assembly. For a 6-axis robotic arm, the number of IK solution types is 8. C_j is a measure of the capability of the robot to complete the structure for each type of IK solution. If the robot can use the same configuration to construct the structure, the fabrication process will be smooth and faster.
- The *FrameScore* (F_j) is the sum of robot configurations that can reach all the approaching frames (independently from the type).

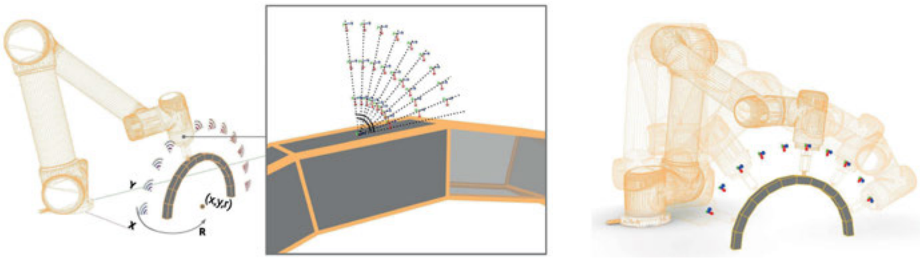

Fig. 3: Left: Position and orientation of the structure $P_j(x, y, r)$. Middle: Potential approaching frames of a block. Right: Smooth fabrication if the robot can finish the assembly with one type of configuration.

Both C_j and F_j are a measure of the redundancy of the system. As we are still not considering the collisions of the robot with the structure and with the other robot, higher scores provide a higher chance that a solution will be found later in the more accurate motion planning process. The DARM algorithm consists of the following steps:

1. For each robot:
   (a) Uniformly sample the workspace in j positions P_j of the assembly.
   (b) Calculate the $K_{i,j}$ potential approaching frames for each block for each sample position.
   (c) Find all valid robot configurations for all approaching frames. In this step, only the robot self-collisions and the collisions with the floor are considered.
   (d) Calculate the ConfigScore (C_j) and FrameScore (F_j) for each position P_j.
   (e) Select the best position as the one with the highest C_j. If there is more than one selected position, then pick the one that has the highest F_j.

   In d), it is possible to only calculate the F_j for the positions with the highest C_j for saving computational power.
2. Fix the relative location of the two robots by moving one of the two such that the best positions (location and rotation) of the structure for both coincide.
3. Use the IK solution types found for each robot (1c) as starting configurations for the motion planning algorithm (further described in Step 4a).

To better illustrate the algorithm, we can consider a simple arch assembly (Fig. 4) to be assembled using one 6-axis UR5 robot with 8 IK solution types. The assembly consists of $n = 10$ movable blocks and two supports (12 elements in total). For each block, we generate $K = 51$ approaching frames. At a sample location $P_0(x_0, y_0, r_0)$, for one of the two robots we have:

– $C_0 = 2$, as the robot can reach at least one approaching frame of each block of $P(x_0, y_0, r_0)$ with two different IK solution types. This was computed using compas_fab analytical solver (Rust et al. 2018) with Pybullet for collision checking (Coumans and Bai 2023).

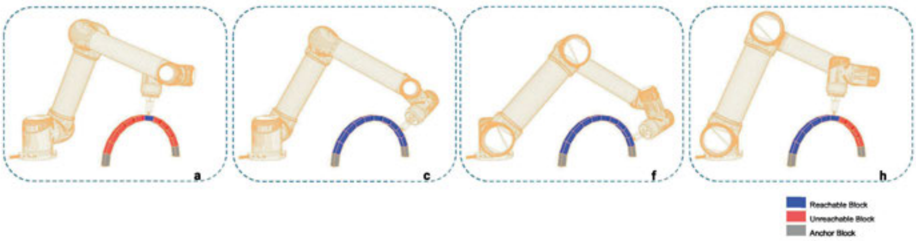

Fig. 4: Reachability of the robot at different IK solution types. Here only IK Types (c) and (f) can reach at least one approaching frame of each block. As a result, the ConfigScore is 2.

– $F_0 = 1377$. This was calculated by removing from the maximum FrameScore $F_0^{\max} = 51 \cdot 10 \cdot 8 = 4080$, the unfeasible solutions due to positions out of reach, self-collision and collisions with the floor.

By calculating the scores for different assembly positions, a score map can be built (Fig. 5) to visualize the preferable location (highest F_j among the ones with highest C_j), highlighted with a red circle in the figure.

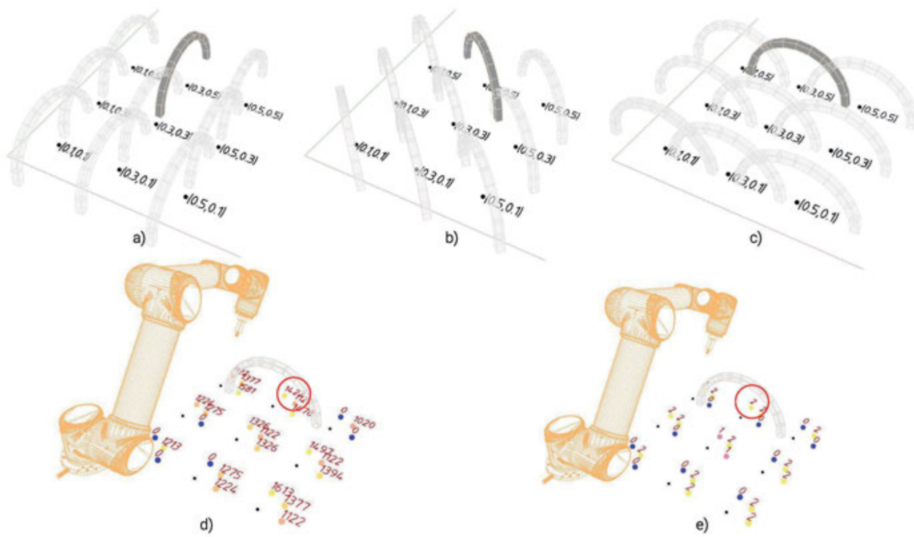

Fig. 5: Use of DARM to design the location and orientation of an arch a)-c) Sample locations and orientations of the arch $P_j(x, y, r)$. Build score maps, d) F_j FrameScore map and e) C_j ConfigScore map. The P_j at red circle is the best location.

Step 4: Motion planning
The motion planning consists of the following parts.

Step 4a: Trajectory design. The robot motion trajectory (Fig. 6) is designed to have (a) a start configuration, (b) a linear pick trajectory to pick up the block from the pick-up station, (c) a free space trajectory to transport the block to approaching frames, (d) a linear place trajectory from approaching frames to the final location, and (h)–(i) a free return trajectory to move back to the start position. The critical trajectory to be planned here is the free space trajectory (c), which covers a large distance and has more possibilities for collisions.

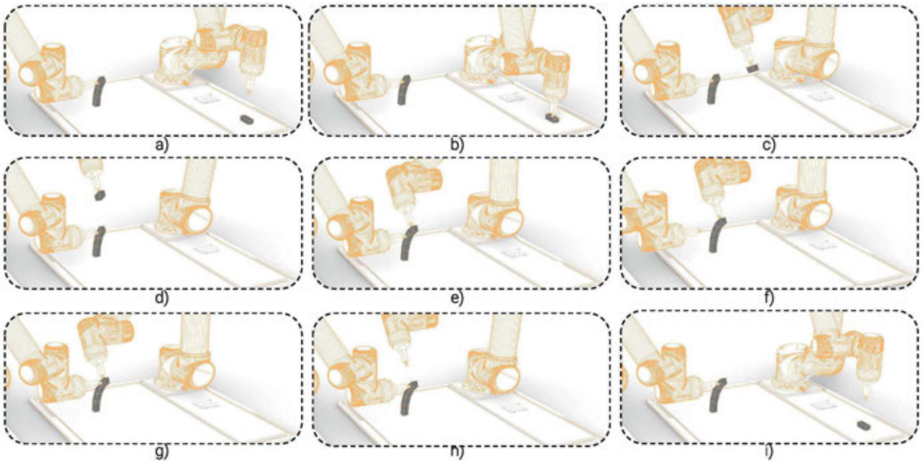

Fig. 6: Robotic motion trajectory: (a) start configuration, (b) linear pick trajectory, (c) free space trajectory, (d) linear place trajectory, (e) place to position, (f)–(g) holder switch holding position, (h)–(i) free return trajectory.

The free space trajectory (c) is planned using the Rapidly-exploring Random Tree (RRT) (LaValle et al. 2001) algorithm, implemented through Open Motion Planning Library, considering all collisions that can occur: self-collisions, collisions between robots, collisions with the table, with the already-built structure. The algorithm is initialized with a starting configuration that significantly affects performance and resulting trajectory; if it is too far from the target configuration, the algorithm may need to explore a large portion of the configuration space before finding a feasible path.

As explained in Step 3, the start configuration for the free space trajectory is derived using DARM. For the selected position P, one or more robot IK solution types are feasible for the entire assembly. We use the configurations from one of these solution types as the starting configuration of the free motion at the pick-up station. Since DARM can be calculated without considering the full range of collisions that can occur, it is possible that in this step a previously feasible configuration is now ruled out. In our experience, this reuse of information from the reachability map largely reduces the computation time of the motion planning and aids in finding good solutions with as few changes in IK solution types as possible.

Step 4b: Collaboration strategy. Next, different roles are assigned to each robot, namely either placer or holder: The placer picks up a block from the pick-up station and places it in the assembly. The holder holds the structure in a certain location so that it remains stable while the assembly process continues.

As a result, the following collaboration strategies are possible. *Asymmetric roles:* one robot is always the *placer* and the other is always the *holder.* Since the two robots have different roles, the process is relatively uncomplicated, but more traveling paths are required, increasing the fabrication time. *Symmetric roles:* the two robots are both *placers* and *holders;* they switch roles after the placement of each block. The planning process is more complicated as the robots cover a larger space with their motion, and their paths intersect more often, but the overall fabrication time is reduced. The impact of the collaboration strategy used will be further discussed in the results (Sec. 4).

Step 5: Fabrication

Once the stability of the structure is assessed and the robotic system and the fabrication planning are defined, the fabrication process can begin.

4 Applications

4.1 Dome

In this section, we apply the proposed workflow to a relatively simple dome assembly (Fig. 7) with a circular base of radius 15 cm.

The geometry of the dome (Step 1) was designed by the form-finding process of Kangaroo2-Grasshopper. The blocks of the assemblies were 3D-printed. To accurately pick blocks of different shapes and sizes, we added a notch connection between the pick-up station and the block (Fig. 9, left (c)). We attach sandpaper (Fig. 9, left (d)) to the contact surface to increase the friction coefficient to 0.8. A 0.5 friction coefficient was conservatively assumed for the analyses in CRA.

The global stability check (Sec. 3, Step 2a) is shown in Fig. 7, right (f). The tilt angle analysis shows limit thickness for several values of the tilt angle. We assumed, for this example, a 40° tilt angle which gave us the thickness of our geometry.

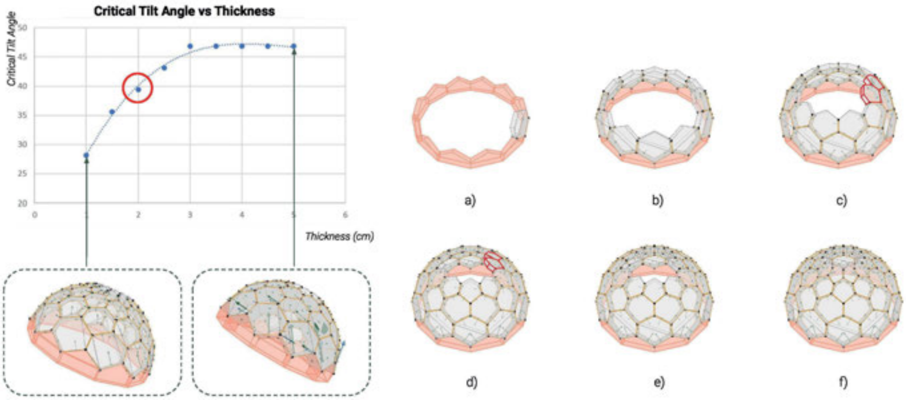

Fig. 7: Left: Critical tilt angle analysis of the dome. Right: Construction sequence analysis of the dome (red base blocks – anchors in CRA analysis, marked red blocks – unstable blocks during construction).

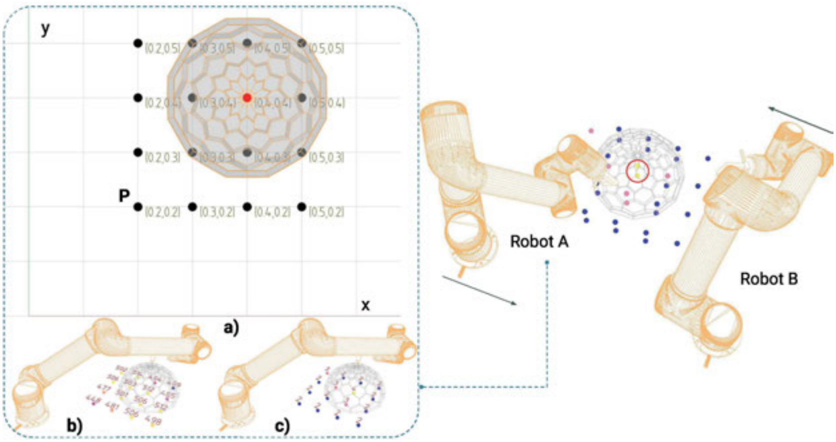

Fig. 8: Left: (a) Sampling of the assembly locations $P(x, y)$ for RobotA, Calculate; (b) frameScore and configScore for each assembly location. Right: DARM for positioning the two robots so that the best positions coincide. The red circle represents the ideal location to have max ConfigSore and FrameScore.

The discretization of the dome was chosen to have a layer-by-layer construction sequence, where each horizontal layer is stable when completed. Figure 7 shows the results obtained using CRA for several steps of the construction sequence (Sec. 3, Step 2b). It is possible to note that in Fig. 7, right (d) two blocks must be placed together with two robots, or one robot needs to hold one of the two while the other robot places the next one. This feedback was used to design the motion plan.

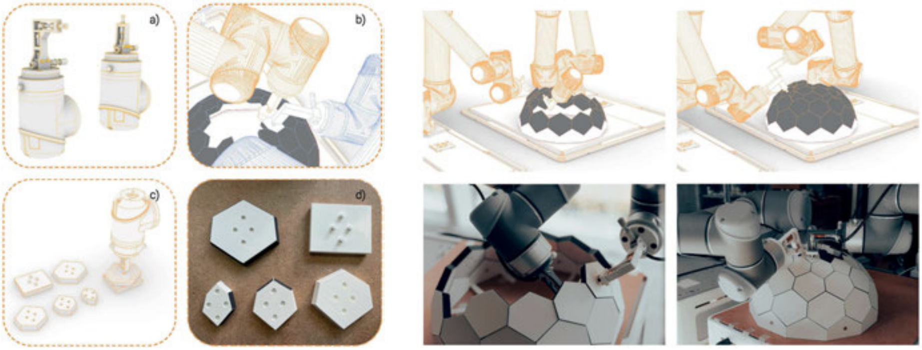

Fig. 9: Left: (a) L-shape and straight end-effectors, (b) end-effectors comparison, (c) notch with pick Station, (d) sandpaper. Right: Asymmetric roles fabrication for the dome and symmetric roles fabrication for the dome.

The setup for this example consisted of two UR5 robots with suction end-effectors; straight and L-shape (Fig. 9, left (a)). The straight end-effector helps reduce the collision between the robot and already constructed structures, while the L-shape end-effector helps reduce the collision of the two robots with each other (Fig. 9, left (b)).

We used DARM (Fig. 8) to locate the position of the structure and robots (Sec. 3, Step 3). Since the dome is axisymmetrical, the rotation is not relevant. Thus we only sampled the workspace for positions in X and Y axis $P_j(x, y)$.

The proposed trajectory design and collaboration strategy from Sec. 3, Step 4 is applied to the motion planning and fabrication of a dome. By using the IK solutions type from DARM as the initial configuration, the motion planning time is significantly reduced. For the given geometry, an asymmetric-roles fabrication is challenging, as the blocks must be held from the intrados side (Fig. 9, right). Thus, we used the asymmetric-roles fabrication only for the lower layers, and symmetric-roles fabrication for the top two layers (Sec. 3, Step 4b).

4.2 Tri-dome

We successfully extended the workflow to a more complex geometry, the tri-dome shown in Fig. 10, left, consisting of three arches (in blue) and three domes (orange). The stability checks are presented in Fig. 10, DARM analysis is shown in Fig. 11 and some steps of motion planning and fabrication process are shown in Fig. 12.

Fig. 10: Left: Geometry of the tri-dome. Right: Construction sequence of a tri-dome.

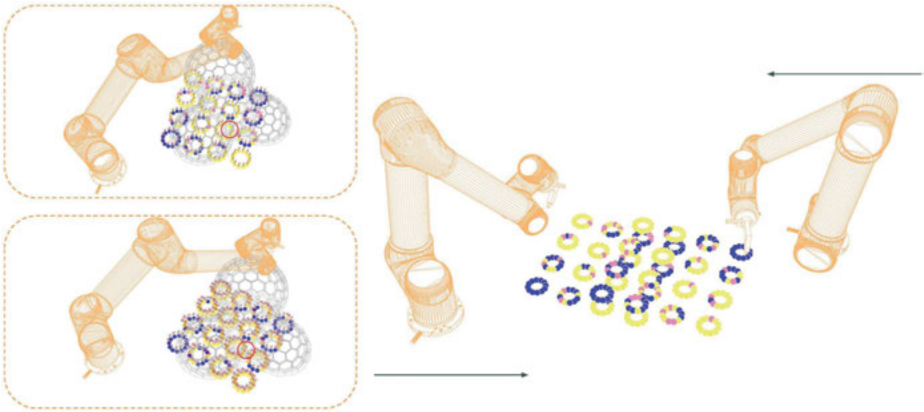

Fig. 11: DARM of the tri-dome. Left: FrameScore map and ConfigScore map. Right: DARM for positioning the two robots so that the best positions coincide.

5 Contribution and discussion

In this paper, we propose a design-to-fabrication workflow for discrete shell structures that, thanks to the use of a dual-robotic system, removes the need for additional falsework. We showed that this workflow can be applied to complex geometries. Our contributions can be summarized in the following points:

- A novel workflow to analyze and fabricate stable discrete assemblies for autonomous construction, with stability check, reachability check, and motion planning.
- A discrete assembly reachability map (DARM) that combines assembly information and robotic reachability information.

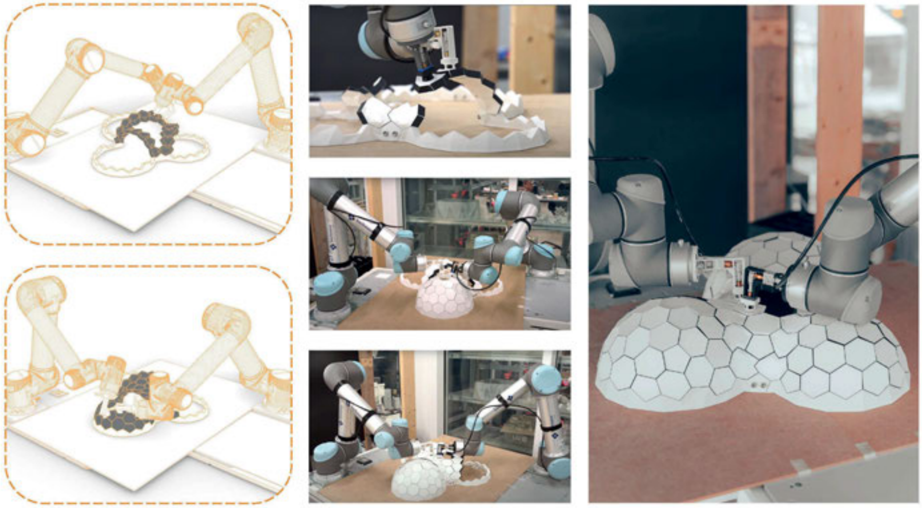

Fig. 12: Motion planning and fabrication of then tri-dome.

The following items are worth to be mentioned:

– *Scaling-up:* although the workflow is generalizable, the current applications are limited by the scale and payload of the chosen robots. For applications in architecture, the payload of the robots should consistently increase.
– *Design space:* The complexity of the structures in this research does not fully represent the possibility of the robotic fabrication strategy. Future works should apply the proposed workflow to more complex geometries and include additional robots.
– *Accuracy:* the inaccuracy of our current workflow comes from imperfections during the 3D printing of the blocks, unevenness of the base, tolerance of the calibration process, and impact forces from the suction end-effectors when initiating or releasing contact with the blocks. This problem could be reduced by implementing force sensors and visual feedback in the workflow.

This research is a contribution to the field of assembly-aware design and robotic path planning. The resulting workflow demonstrates how to embed stability and robotic reachability constraints into the design of spanning structures. Implemented in the early stages of a design, such workflow drastically reduces the need for falsework, thus improving productivity and reducing waste.

References

Beyeler, L., J.-C. Bazin, and E. Whiting. 2015. A Graph-Based Approach for Discovery of Stable Deconstruction Sequences. In *Advances in Architectural Geometry 2014,* Eds. P. Block, J. Knippers, N. J. Mitra, and W. Wang, 145–57. Springer International Publishing.

Block, P., T. Van Mele, M. Rippmann, M. DeJong, J. Ochsendorf, M. Escobedo, and D. Escobedo. 2016. Armadillo Vault – An Extreme Discrete Stone Shell. *Detail* 10 (October): 940–42.

Coumans, E., and Y. Bai. 2023. PyBullet, a Python Module for Physics Simulation for Games, Robotics and Machine Learning. https://github.com/bulletphysics/bullet3

Deuss, M., D. Panozzo, E. Whiting, Y. Liu, P. Block, O. Sorkine-Hornung, and M. Pauly. 2014. Assembling Self-Supporting Structures. In *ACM Transactions on Graphics.* Vol. 33. Association for Computing Machinery. DOI: 10.1145/2661229.2661266

Heyman, J. 1995. *The Stone Skeleton: Structural Engineering of Masonry Architecture.* Cambridge University Press. DOI: 10.1017/CBO9781107050310

Kao, G. T.-C. 2022. COMPAS CRA: Coupled Rigid-Block Analysis (CRA) for the COMPAS Framework. 2022. DOI: 10.5281/zenodo.7043135

Kao, G. T.-C., A. Iannuzzo, B. Thomaszewski, S. Coros, T. Van Mele, and P. Block. 2022. Coupled Rigid-Block Analysis: Stability-Aware Design of Complex Discrete-Element Assemblies. *CAD Computer Aided Design* 146 (May). DOI: 10.1016/j.cad.2022.103216

Kao, G. T.-C., A. Körner, D. Sonntag, L. Nguyen, A. Menges, and J. Knippers. 2017. Assembly-Aware Design of Masonry Shell Structures: A Computational Approach. In *International Association for Shell and Spatial Structures.* https://www.researchgate.net/publication/320188621

LaValle, S. M., and Jr. J. J. Kuffner. 2001. Randomized Kinodynamic Planning. *The International Journal of Robotics Research* 20 (5): 378–400. DOI: 10.1177/02783640122067453

Makhal, A. and A. K. Goins. 2018. Reuleaux: Robot Base Placement by Reachability Analysis. *Second IEEE International Conference on Robotic Computing (IRC),* 137–42. http://arxiv.org/abs/1710.01328.

Parascho, S., I. Xi Han, A. Beghini, M. Miki, S. Walker, E. P. G. Bruun, and S. Adriaenssens. 2020. LightVault: A Design and Robotic Fabrication Method for Complex Masonry Structures. In *Advances in Architectural Geometry,* 350–75.

Piškorec, L., D. Jenny, S. Parascho, H. Mayer, F. Gramazio, and M. Kohler. 2019. The Brick Labyrinth. In *Robotic Fabrication in Architecture, Art and Design 2018,* Eds. J. Willmann, P. Block, M. Hutter, K. Byrne, and T. Schork, 489–500. Springer International Publishing.

Ponce, A. R., and R. R. Melendez. 2015. Curves of Clay: Mexican Brick Vaults and Domes. In *Architecture and Mathematics from Antiquity to the Future: Volume I: Antiquity to the 1500s,* Eds. K. Williams and M. J. Ostwald, 309–24. Springer International Publishing. DOI: 10.1007/978-3-319-00137-1_21

Rippmann, M. 2016. Funicular Shell Design: Geometric Approaches to Form Finding and Fabrication of Discrete Funicular Structures. ETH Zurich, Department of Architecture. DOI: 10.3929/ethz-a-010656780

Rust, R., G. Casas, S. Parascho, D. Jenny, K. Dörfler, M. Helmreich, A. Gandia et al. 2018. COMPAS-FAB: Robotic Fabrication Package for the COMPAS Framework. https://github.com/compas-dev/compas_fab/.

Whiting, E., J. Ochsendorf, and F. Durand. 2009. Procedural Modeling of Structurally-Sound Masonry Buildings. *ACM Transactions on Graphics* 28 (5): 1–9. DOI: 10.1145/1618452.1618458

Wu, K., and A. Kilian. 2020. Designing Compression-Only Arch Structures Using Robotic Equilibrium Assembly. In *Impact: Design With All Senses,* 608–22. Springer International Publishing. https://doi.org/10.1007/978-3-030-29829-6_47.