Materials Today: Proceedings xxx (xxxx) xxx

Contents lists available at ScienceDirect

Materials Today: Proceedings

journal homepage: www.elsevier.com/locate/matpr

Interlocking assemblies: Applications and methods

Peng Song

Singapore University of Technology and Design, 487372 Singapore, Singapore

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Interlocking assembly
Structural stability
Computational design
Digital fabrication

ABSTRACT

Interlocking assemblies are assemblies where all the parts, except a key part, are immobilized relative to one another, preventing the whole assembly from falling apart under external forces. Starting from the key part, interlocking assemblies can be repeatedly disassembled and re-assembled, facilitating reuse of the parts. Due to advantages in structural stability and disassemblability, interlocking assemblies have been used in a variety of applications, including puzzles, 3D printing, furniture, and architecture. However, designing interlocking assemblies is a highly challenging task because of the global nature of the interlocking property. To address this challenge, a number of computational methods and tools have been developed in the last decade, mainly by the computer graphics community, for designing and fabricating personalized interlocking assemblies with various geometric forms. The design problem is typically formulated as a shape decomposition, joint planning, or 3D tiling problem, according to the user provided input. In this paper, we review these problem formulations, classify the state-of-the-art computational design methods, and propose possible directions for future research.

Selection and peer-review under responsibility of the scientific committee of The International Conference on Additive Manufacturing for a Better World.

1. Introduction

3D assemblies refer to objects that combine multiple component parts into a structure with a specific form and/or functionality. Connection mechanisms are usually required to prevent the parts from moving relative to one another and make the assembly structurally stable for practical use. However, these connection mechanisms can be irreversible (e.g., glue), impair the structural integrity of parts (e.g., nails), or degrade the external appearance of the assembly (e.g., clamps).

Rather than relying on additional explicit connectors, component parts in an *interlocking assembly* are connected into a structurally stable assembly purely based on the geometric arrangement of the parts. This intriguing property facilitates repeated assembly and disassembly, and significantly simplifies the correct alignment of parts during construction. Consequently, interlocking assemblies have been used in a variety of applications, including puzzles [1], 3D printing [2], furniture [3], and architecture [4].

In an interlocking assembly, parts need to follow certain orders to be assembled into the target shape. Once assembled, there is only one movable part, called the key, while all other parts as well as any subset of parts are immobilized relative to one another [5]; see Fig. 1(a). According to the static-kinematic duality, an interlocking assembly is in equilibrium under arbitrary external forces when the key is held by other means [6]. Compared with non-interlocking assemblies, interlocking assemblies are structurally stable under unpredictable external forces yet enforce higher complexity on the parts geometry and their joining.

The main challenge of designing interlocking assemblies is to ensure two conflicting properties simultaneously: structural stability and disassemblability. This is because structural stability requires strict joining to restrict relative movement among parts yet disassemblability demands at least one collision-free plan to separate the parts; see Figure (b-d) for three non-interlocking examples. To address the challenge, a number of computational methods and tools have been developed in the last decade, mainly by the computer graphics community, for designing and fabricating personalized interlocking assemblies with various geometric forms [7]. The design problem is typically formulated as a shape decomposition, joint planning, or 3D tiling problem, according to the user provided input.

In this paper, we review applications and computational design methods of interlocking assemblies. In Section 2, we enumerate and introduce a variety of applications in which interlocking assemblies have demonstrated their unique strengths in structural stability and disassemblability. In Section 3, we review formulations of interlocking assembly design problem and then classify

https://doi.org/10.1016/j.matpr.2022.08.548
2214-7853/
Selection and peer-review under responsibility of the scientific committee of The International Conference on Additive Manufacturing for a Better World.

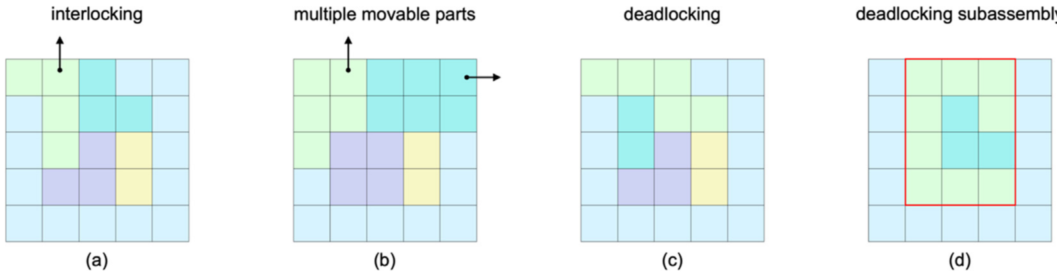

Fig. 1. (a) A 2D interlocking assembly where the green part is the key and the black arrow is the key's moving direction. (b-c) Three non-interlocking examples: an assembly with multiple movable parts, an assembly that is deadlocking (i.e., non-disassemblable), an assembly with a deadlocking subassembly (see the red rectangle). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the state-of-the-art computational methods to address the problem. We conclude with a discussion of limitations of existing approaches and some thoughts on future research directions.

2. Applications of interlocking assemblies

Interlocking is a general methodology to connect parts purely based on the parts' geometric arrangement, without relying on any external connectors such as nails and screws. Whenever external connectors cannot be used or are not preferred to be used, interlocking is an alternative to connect parts for making an assembly that is structurally stable and disassemblable. Here we enumerate and introduce several applications of interlocking assemblies:

Puzzles. Geometric puzzles are challenging games that require one to assemble the puzzle pieces into a target 3D shape [1]. Among the various families of geometric puzzles, interlocking puzzles are particularly intriguing and elegant since they become structurally stable once all puzzle pieces are put together; see Fig. 2(a). A few interesting subclasses of interlocking puzzles have been studied and designed in the literature:

- A *burr puzzle* is an interlocking puzzle consisting of notched sticks [8]. The interlocking mechanism of burr puzzles are centralized in a small region in the middle of the puzzle.
- A *recursive interlocking puzzle* is a puzzle that remains interlocking after the sequential removal of the puzzle pieces [5]. One feature of these puzzles is that they have a unique sequence to (dis)assemble the puzzle pieces.
- A *high-level interlocking puzzle* is a puzzle with a high-level-of-difficulty in terms of disassembly [9]. The level of difficulty is generally assessed by the number of moves required to remove the first subassembly from the puzzle.
- A *dissection puzzle* requires assembling a common set of pieces into multiple (usually-two) distinct forms. A dissection puzzle can be made steady by designing the puzzle pieces in a way that they are interlocking in each geometric form [10].

Multi-way joints. A woodworking joint is typically used for connecting a pair of parts to restrict their relative movement. A woodworking joint can be extended to connect more than two parts, called a multi-way joint [11]. The interlocking concept can be applied to a multi-way joint to enhance their ability of connecting parts. Fig. 2(b) shows an example multi-way joint used in a chair frame structure [12], where the three elongated parts are interlocking based on the joint integrated at one end of each part.

3D printing. 3D printing can easily manufacture shapes with geometric complexity that is unattainable by traditional techniques. However, it still has limitations such as limited printing volume and long printing time. Assembly-based 3D printing is a strategy to address these limitations by representing a target shape with a set of simpler and/or smaller parts. To ensure that the 3D printed parts can be assembled and form a structurally stable assembly, integral joints can be planed and constructed on the parts to make them interlocking. Fig. 2(c) shows an example, where a 6-part shell assembly is interlocking based on the joints constructed at the boundary of each part [2].

Laser cutting. Compared with 3D printers, laser cutters can only cut 2D polygonal shapes out of a flat plank. Despite this limitation, laser cutting has many advantages in terms of fabrication like high precision, high speed, fewer constraints on the fabrication space. To fabricate a 3D shape, assembly-based laser cutting aims to approximate the 3D shape with a set of 2D planar parts made by laser cutting. Integral joints (e.g., mortise and tenon joints, halved joints) can be planned and constructed on the laser-cut parts to make the resulting assembly interlocking. Fig. 2(d) shows an interlocking laser-cut assembly used as a supporting base for cost-effective fabrication of a large statue [13].

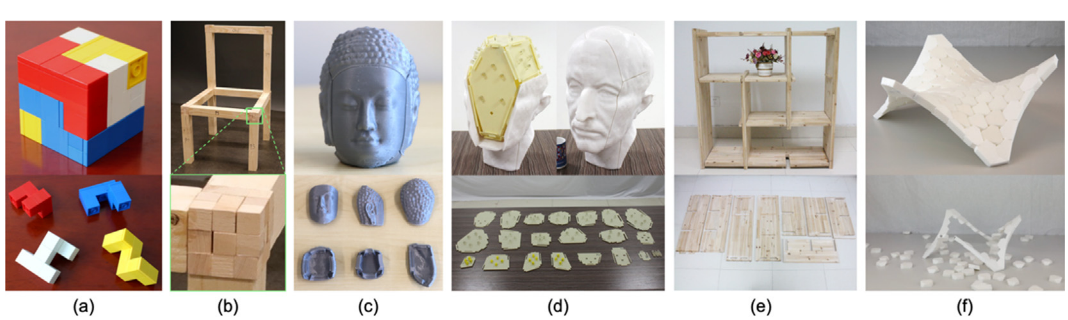

Fig. 2. Applications of interlocking assemblies: (a) puzzle, (b) multi-way joint, (c) 3D printing a statue, (d) laser-cut support structure, (e) furniture, and (f) architecture. The assembly is shown on the top and the component parts are shown at the bottom.

Furniture. Furniture typically consists of elongated and planar parts that are connected together by glue, nails, hinges, screws, or other means that do not encourage disassembly and reassembly. An alternative approach is to use an interlocking mechanism and design interlocking furniture [3], where the component parts tightly interlock with one another by a network of joints; see Fig. 2(e) for an example.

Architecture. Discrete architecture is an architectural paradigm that utilizes the combinatorial organization logic of a discrete set of architectural parts to accelerate the physical assembly of architecture and to reduce the cost and material wastage [14]. By making the architectural parts interlocking, the physical assembly process is possible to be significantly simplified as a set of pick-and-place operations, by avoiding the complex procedure of installing fasteners like screws and minimizing the usage of scaffold that makes the incomplete assembly in equilibrium [15,16]. Hence, interlocking architecture has great potential to be automatically constructed with robots [17]. Fig. 2(f) shows an interlocking free-form architecture composed of a set of rigid blocks and a large boundary part [6].

3. Design of interlocking assemblies

Designing an interlocking assembly is extremely hard for humans, even for skilled professional. For example, as mentioned in [1], designing an interlocking puzzle requires hours, or even days, of mental work. Only in the late 1970 s, Cutler [18] proposed to use computers to exhaustively try and discover new possible configurations, thus introducing a large number of six-piece interlocking structures. Nevertheless, due to the combinatorial complexity, his computer program took almost three years to loop over the tens of billions of possible configurations for a variable cubical volume of less than 4^3 voxels.

The recent advances in digital fabrication motivates the need of designing personalized interlocking assemblies, e.g., with user-specified 3D geometric form and number of parts. To this end, a number of computational methods and tools have been developed in the last decade, mainly by the computer graphics community, to enable general users to design interlocking assemblies conveniently and efficiently. We review different ways to formulate the interlocking assembly design problem in Section 3.1 and classify the state-of-the-art computational design methods in Section 3.2.

3.1. Problem formulation

Computational design of interlocking assemblies is typically formulated as shape decomposition, joint planning, or 3D tiling problem, according to the user provided input; see Fig. 3. In particular, the first two formulations are a top-down problem, meaning that the assembly's whole shape is given and the goal is to generate the parts or joints. The last formulation is a bottom-up problem, meaning that parts with known geometry are provided and the goal is to generate a layout of the parts (or their instances) to make the resulting assembly interlocking.

Shape decomposition. When the user-specified input is a target shape, computational design of interlocking assemblies can be formulated as a shape decomposition problem; see Fig. 3(a). The user can also specify the desired number of parts in the assembly. The goal of the shape decomposition is that all the decomposed parts interlock with one another. In addition, the geometry of each decomposed part should be connected to ensure fabricablility. This formulation has been used for designing interlocking puzzles [5,8,9,10] and solid structures for 3D printing [19].

Joint planning. When the user-specified input is a set of pre-segmented parts without joints, designing interlocking assemblies can be formulated as a joint planning problem; see Fig. 3(b). The goal is to plan and construct a set of predefined joints (e.g., mortise and tenon joint) or variable joints with optimized geometry (e.g., multi-way joint) among the initial parts to make them interlocking. This formulation has been used for designing interlocking frame structures such as furniture [3,12,20] or interlocking shell structures for 3D printing [2].

3D Tiling. When the user-specified input is a set of parts with known geometry, designing interlocking assemblies can be formulated as a 3D tiling problem; see Fig. 3(c). The goal is to find a layout of the parts (or their instances) to make the resulting assembly interlocking, similar to building Lego assemblies. One good example is a set of voxel-like interlocking blocks proposed by Zhang and Balkcom [21,22]. They connect instances of these blocks layer-by-layer into various interlocking voxelized shapes. Another example is the SL block proposed by Shih [23], which is an octocube consisting of an *S*-shaped and an *L*-shaped tetracubes attaching to each other along sides. Reinforcement learning can be utilized to find a layout of the SL blocks to approximate a given shape [24].

3.2. Computational methods

A number of computational methods have been proposed for addressing the above formulated interlocking assembly design problem, in particular, shape decomposition and joint planning. We classify these methods into the following categories:

Retargeting. This method creates interlocking assemblies by retargeting an existing interlocking configuration within a user-specified target shape. This method has been used for addressing the above shape decomposition problem. Xin et al. [8] developed a retargeting approach to create burr puzzles by replicating and connecting multiple instances of an existing six-piece interlocking burr structure within a given target shape. The retargeting method is also possible to address the joint planning problem. One limitation of this method is that it has less flexibility to design interlocking assemblies with complex shapes since it reuses an existing interlocking configuration instead of creating a new one that can better adapt to the given shape.

Randomized search. When the design space is not that large, a randomized search can be used to find an interlocking configura-

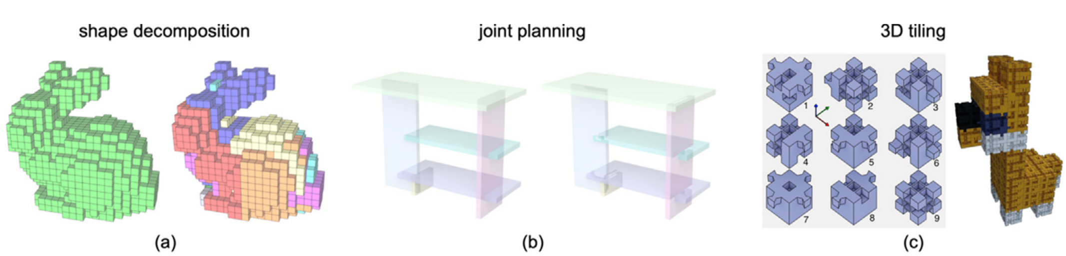

Fig. 3. Formulating the interlocking assembly design problem in three different ways: (a) shape decomposition [5], (b) joint planning [3], and (c) 3D tiling [21]. For each formulation, the user input is shown on the left while the design output is shown on the right.

tion. This method has been used to address the joint planning problem. Yao et al. [2] designed interlocking shell assemblies for 3D printing by using a randomized search with pruning to generate candidate joint configurations and verifying their global interlocking by using physically based simulation. The randomized search can be inefficient to address the shape decomposition problem due to the large and possibly continuous design space.

Divide-and-conquer. A more efficient method to address the interlocking assembly design problem is based on divide-and-conquer. The key idea is to construct Local Interlocking Groups (LIGs), which are a subset of connected parts that are locked by a specific key in the group, and then enforce dependency among these LIGs. The advantage of this method is that the resulting assemblies are guaranteed to be interlocking. Yet, the limitation is that the explored search space is restricted to a small subset of all possible interlocking configurations. Song et al. [5] first proposed this method to address the shape decomposition problem for designing recursive interlocking puzzles. Given a voxelized 3D shape, their method iteratively extracts pieces while enforcing a local interlocking condition among every-three consecutive pieces. This method was later extended to handle smooth non-voxelized shapes for 3D printing solid shapes [19]. This method also can be used to address the joint planning problem. Fu et al. [3] focused on plate structures such as furniture that have been initially partitioned into parts, and computed an interlocking joint configuration following the LIG-based approach, where each LIG has only 3 or 4 parts and thus the joint configuration in each LIG can be searched exhaustively. This method has been extended to interlock 2D laser-cut parts into a convex polyhedron [13] and to design reconfigurable furniture with multi-key interlocking [20].

DBG-based method. To address the limitation of the divide-and-conquer method, Wang et al. [12] represented parts blocking relations with a family of base Directional Blocking Graphs (DBGs) and leveraged efficient graph analysis tools to test and compute an interlocking arrangement of the parts. Compared with the divide-and-conquer method, the strength of this DBG-based method is twofold: 1) allowing exploring the full search space of interlocking configurations; and 2) supporting a wide range of geometric forms of resulting assemblies. Gilibert et al. [25] extended the DBG-based method with a Markov process and turtle graphics to design 2D interlocking assemblies that can be assembled for any prescribed combination of translations and rotations.

Shape optimization. Shape optimization method makes a given assembly with pre-segmented parts interlocking by optimizing the continuous shape of each part or joint. Given an initial assembly with planar contacts, Wang et al. [26] made the assembly interlocking by transforming each planar contact into a curved-contact joint using shape optimization, during which a motion-based equilibrium method is proposed to ensure interlocking.

4. Discussion and future work

The state of the art in computational design of interlocking assemblies has evolved rapidly over the last decade. A number of computational methods and tools have been developed for designing various kinds of interlocking assemblies for different usage scenarios. To simplify the design problem, these methods have made several assumptions. First, they assume the fabrication material is rigid. By this, design of interlocking assemblies can be formulated as a geometric modeling and optimization problem, with minimal consideration of material behavior (e.g., friction). Very few research works study interlocking assemblies with deformable parts [27,28]. Second, existing methods generally assume monotone and linear (dis)assembly plans [29] to simplify the problem of checking (dis)assemblability. Supporting more complex disassembly plans such as non-monotone [30] or non-coherent [31] plans would be an interesting future work. Third, existing methods ignore tolerance as well as its impact on the resulting assembly at the design stage, leading to a discrepancy between the virtual design and fabricated assembly in terms of structural stability. Future research effort can be devoted to tolerance analysis [32,33] of interlocking assemblies as well as designing interlocking assemblies whole structural stability is minimally affected by the tolerance. By investigating interlocking assemblies with some of these assumptions relaxed or removed, we will not only expand the scope of interlocking assemblies, but also be able to design interlocking assemblies with more predictable behavior, making these assemblies more ready to be used in real applications.

Data availability

No data was used for the research described in the article.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the SUTD Start-up Research Grant (Award Number: SRG ISTD 2019 148).

References

[1] Stewart T. Coffin, Geometric Puzzle Design, A. K. Peters, 2006.
[2] M. Yao, Z. Chen, X.u. Weiwei, H. Wang, Modeling, Evaluation and Optimization of Interlocking Shell Pieces, Comput. Graph. Forum (Pacific Graph.) 36 (7) (2017) 1–13, https://doi.org/10.1111/cgf.13267.
[3] Chi-Wing Fu, Peng Song, Xiaoqi Yan, Lee Wei Yang, Pradeep Kumar Jayaraman, and Daniel Cohen-Or. Computational Interlocking Furniture Assembly. ACM Transactions on Graphics (SIGGRAPH), Article No. 91, 2015. https://dl.acm.org/doi/10.1145/2766892.
[4] B. Deepak, Sustainable Dry Interlocking Block Masonry Construction, In 15th International Brick and Block Masonry Conference, 2012.
[5] Peng Song, Chi-Wing Fu, and Daniel Cohen-Or. Recursive Interlocking Puzzles. ACM Transactions on Graphics (SIGGRAPH Asia), 31(6), Article No. 128, 2012. https://dl.acm.org/doi/10.1145/2366145.2366147.
[6] Ziqi Wang, Peng Song, Florin Isvoranu, and Mark Pauly. Design and Structural Optimization of Topological Interlocking Assemblies. ACM Transactions on Graphics (SIGGRAPH Asia), 38(6), Article No. 193, 2019. https://dl.acm.org/doi/10.1145/3355089.3356489.
[7] Z. Wang, P. Song, M. Pauly, State of the Art on Computational Design of Assemblies with Rigid Parts, Computer Graphics Forum (Eurographics) 40 (2) (2021) 633–657, https://doi.org/10.1111/cgf.142660.
[8] Shi-Qing Xin, Chi-Fu Lai, Chi-Wing Fu, Tien-Tsin Wong, Ying He, and Daniel Cohen-Or. Making Burr Puzzles from 3D Models. ACM Transactions on Graphics (SIGGRAPH), 30(4), Article No. 97, 2011. https://dl.acm.org/doi/10.1145/2010324.1964992.
[9] Rulin Chen, Ziqi Wang, Peng Song, and Bernd Bickel. Computational Design of High-level Interlocking Puzzles. ACM Transactions on Graphics (SIGGRAPH), 41(4), 2022.
[10] K. Tang, P. Song, X. Wang, B. Deng, F.u. Chi-Wing, L. Liu, Computational Design of Steady 3D Dissection Puzzles, Computer Graphics Forum (Eurographics) 38 (2) (2019) 291–303, https://doi.org/10.1111/cgf.13638.
[11] M. Larsson, H. Yoshida, N. Umetani, T. Igarashi, Tsugite: Interactive Design and Fabrication of Wood Joints, in: In ProceedIngs of the 33rd Annual ACM Symposium on User Interface Software and Technology (UIST), 2020, pp. 317–327.
[12] Ziqi Wang, Peng Song, and Mark Pauly. DESIA: A General Framework for Designing Interlocking Assemblies. ACM Transactions on Graphics (SIGGRAPH Asia), 37(6), Article No. 191, 2018. https://dl.acm.org/doi/10.1145/3272127.3275034.
[13] Peng Song, Bailin Deng, Ziqi Wang, Zhichao Dong, Wei Li, Chi-Wing Fu, and Ligang Liu. CofiFab: Coarse-to-Fine Fabrication of Large 3D Objects. ACM Transactions on Graphics (SIGGRAPH), 35(4), Article No. 45, 2016. https://dl.acm.org/doi/10.1145/2897824.2925876.
[14] G. Retsin, Discrete Architecture in the Age of Automation, Architectural Design 89 (2) (2019) 6–13.

[15] Mario Deuss, Daniele Panozzo, Emily Whiting, Yang Liu, Philippe Block, Olga Sorkine-Hornung, and Mark Pauly. Assembling Self-Supporting Structures. ACM Transactions on Graphics (SIGGRAPH Asia), 33(6), Article No. 214, 2014. https://dl.acm.org/doi/10.1145/2661229.2661266.
[16] Gene Ting-Chun Kao, Axel Kórner, Daniel Sonntag, Long Nguyen, Achim Menges, and Jan Knippers. Assembly-aware Design of Masonry Shell Structures: A Computational Approach. In Proceedings of the International Association for Shell and Spatial Structures Symposium, 2017.
[17] Yu-Chou Chiang, Henriette Bier, and Sina Mostafavi. Design to Robotic Assembly: An Exploration in Stacking. Frontiers in Digital Humanities, 5, Article No. 23, 2018.
[18] B. Cutler, A Computer Analysis of All 6-piece Burrs, Self published (1994).
[19] Peng Song, Zhongqi Fu, Ligang Liu, and Chi-Wing Fu. Printing 3D Objects with Interlocking Parts. Computer Aided Geometric Design (GMP), 35-36, 137-148, 2015. https://www.sciencedirect.com/science/article/pii/S0167839615000436.
[20] Peng Song, Chi-Wing Fu, Yueming Jin, Hongfei Xu, Ligang Liu, Pheng-Ann Heng, and Daniel Cohen-Or. Reconfigurable Interlocking Furniture. ACM Transactions on Graphics (SIGGRAPH Asia), 36(6), Article No. 174, 2017. https://dl.acm.org/doi/10.1145/3130800.3130803.
[21] Yinan Zhang and Devin Balkcom. Interlocking Structure Assembly with Voxels. In IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, 2173–2180, 2016.
[22] Yinan Zhang, Yotto Koga, and Devin Balkcom. Interlocking Block Assembly with Robots. IEEE Transactions on Automation Science and Engineering, 18(3), 902-916, 2021. https://ieeexplore.ieee.org/document/9410285.
[23] Shen-Guan Shih. The Art and Mathematics of Self-Interlocking SL Blocks. In Proc. Bridges Conference. 107-114, 2018.
[24] Bastian Wibranek, Yuxi Liu, Niklas Funk, Boris Belousov, Jan Peters, and Oliver Tessmann. Reinforcement Learning for Sequential Assembly of SL-Blocks: Self-interlocking Combinatorial Design Based on Machine Learning. In Proc. the 39th eCAADe Conference, vol. 1, 27-36, 2021.
[25] Pierre Gilibert, Romain Mesnil, and Olivier Baverel. Rule-based Generative Design of Translational and Rotational Interlocking Assemblies. Automation in Construction. 135, Article No. 104142, 2022. https://www.sciencedirect.com/science/article/pii/S0926580522000152.
[26] Ziqi Wang, Peng Song, and Mark Pauly. MOCCA: Modeling and Optimizing Cone-joints for Complex Assemblies. ACM Transactions on Graphics (SIGGRAPH), 40(4), Article No. 181, 2021. https://dl.acm.org/doi/10.1145/3450626.3459680.
[27] Mélina Skouras, Stelian Coros, Eitan Grinspun, and Bernhard Thomaszewski. Interactive Surface Design with Interlocking Elements. ACM Transactions on Graphics (SIGGRAPH Asia), 34(6), Article No. 224, 2015. https://dl.acm.org/doi/10.1145/2816795.2818128.
[28] Vinayak R. Krishnamurthy, Ergun Akleman, Sai Ganesh Subramanian, Matthew Ebert, Jiaqi Cui, Chia-an Fu, and Courtney Starrett. Geometrically Interlocking Space-Filling Tiling Based on Fabric Weaves. IEEE Transactions on Visualization and Computer Graphics, preprint, 2020. https://www.computer.org/csdl/journal/tg/5555/01/09376630/1rSN0xMbjfa.
[29] Somayé Ghandi and Ellips Masehian. Review and Taxonomies of Assembly and Disassembly Path Planning Problems and Approaches. Computer-Aided Design, vol. 67-68, 58-86, 2015. https://www.sciencedirect.com/science/article/pii/S0010448515000664.
[30] Ellips Masehian and Somayé Ghandi. Assembly Sequence and Path Planning for Monotone and Nonmonotone Assemblies with Rigid and Flexible Parts. Robotics and Computer-Integrated Manufacturing, 72, Article No. 102180, 2021. https://www.sciencedirect.com/science/article/pii/S0736584521000636.
[31] Xinya Zhang, Robert Belfer, Paul G. Kry, and Etienne Vouga. C-Space Tunnel Discovery for Puzzle Path Planning. ACM Transactions on Graphics (SIGGRAPH), 39(4), Article No. 104, 2020. https://dl.acm.org/doi/abs/10.1145/3386569.3392468.
[32] Y. Cao, T. Liu, J. Yang, A Comprehensive Review of Tolerance Analysis Models, The International Journal of Advanced Manufacturing Technology 97 (2018) 3055–3085, https://doi.org/10.1007/s00170-018-1920-2.
[33] Samuel Lensgraf, Karim Itani, Yinan Zhang, Zezhou Sun, Yijia Wu, Alberto Quattrini Li, Bo Zhu, Emily Whiting, Weifu Wang, and Devin Balkcom. PuzzleFlex: Kinematic Motion of Chains with Loose Joints. IEEE International Conference on Robotics and Automation (ICRA), 6730-6737, 2020.