A ROS2-based 3D Block Printer System for Additive Manufacturing with Non-Empirical Stability Analysis

Pedro Saman Diogenes N CESARINO
Graduate School of Engineering Science
Yokohama National University
Yokohama, Japan
saman-pedro-tw@ynu.jp

Yusuke MAEDA
Faculty of Engineering
Yokohama National University
Yokohama, Japan
maeda@ynu.ac.jp

***Abstract*—The presented paper introduces a new 3D block printer system designed for the safe assembly of toy blocks using a robotic manipulator. The system involves converting a CAD object into a block representation and defining an assembly plan while considering mechanical stability. If an unsafe assembly is detected, the system employs various stabilizing strategies. The assembly process is demonstrated using a simulated environment with a Denso Wave Cobotta robot. The robotic control system is developed using ROS2 Humble and MoveIt2 for motion planning. The system's efficiency is showcased by assembling different 3D objects based on the generated simulation output.**

***Index Terms*—Additive manufacturing, block printing, automatic assembly, industrial robot**

I. Introduction

Additive Manufacturing (AM) is a technique that generates a desired shape from a 3D model by depositing material incrementally instead of removing it like conventional milling and drilling machines. The popular 3D printers are a good example of the potential of this method. Being cost-efficient, fast and able to create highly customizable designs utilizing diverse materials, AM is utilized in many areas like toy fabrication, biomedicine, aerospace engineering, and civil construction [1]. Such flexibility and accessibility enabled its use in many places that otherwise would not be able to create such items like schools, homes and laboratories.

There are many AM methods of depositing material like: stereolithography apparatus (SLA), inkjet printing, binder printing, fused deposition modeling, laminate object, etc. [2]. Assembling objects with prefabricated building blocks can also be seen as AM. Several studies have tackled the construction and optimization of LEGO models ([3], [4], [5]). However, these studies focused on the generation of the sculptures and did not consider their automated assembly.

Many groups have studied robotic 3D printing with digital materials. Hiller and Lipson [6] analyzed "voxel printing" and showed that voxel printing has some advantages, including perfect repeatability and the use of multiple materials. Sekijima et al. [7] proposed a digital material named a "Kelvin Block" that can be connected to each other by magnetic force. Suzuki et al. [8] presented an automatic assembly system using "Dynablock", their original magnetic block that can assemble multiple blocks simultaneously. Gilday et al. [9] proposed a technique to automatically assemble, disassemble, and reassemble LEGO block models. Their approach can optimize block operation in terms of time, number of blocks, and the probability of success. However, it does not consider mechanical stability of block sculptures during operation.

Using the block assembly approach to AM has some limitations. Mainly regarding the resolution of the assembled shapes, where the minimum size of a block imposes a limit in how many details can be reproduced. Also, this method has difficulty dealing with high mechanical properties like moving parts due to the bonding strength between building blocks.

On the other hand, the system has the following characteristics that are not usually available in other conventional 3D printers: (i) possibility to construct structures composed of multiples materials; (ii) flexibility to print in multiple colors, without adhesive, during assembly time; (iii) having a final product that can be later disassembled to be reused. It can potentially have other advantages like higher assembly precision and repeatability due to the usage of a standardized material and its self-alignment effect. Moreover, by using functional blocks, it is possible to fabricate shapes with function (e.g., electrical circuits [10]).

Previously, we developed a 3D block printer system in which a robot assembles LEGO-like toy blocks to create brick sculptures from desired CAD models [11]. This system can transform a 3D object into block model and define an assembly plan that aims to ensure the assembly safety, that is, every block insertion should not break the already assembled structure, using some empirical statements. These empirical rules were defined using experience in building block structures. As an example, the most basic rule is a block must have at least one contact point with a block in the layer below, that is, it cannot be floating.

If deemed unsafe, the system can deal with the unassemblable shapes by performing three different approaches. Those are: changing the block insertion order, inserting support blocks and dividing the structure into subassemblies. If a

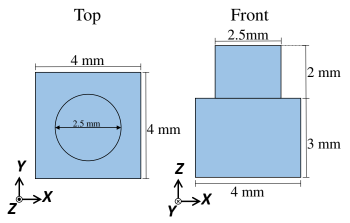
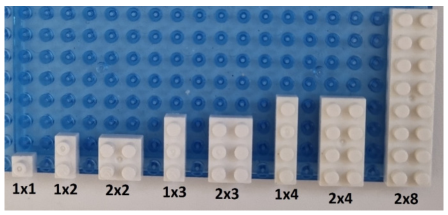
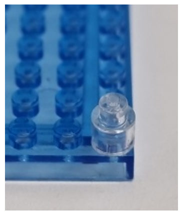


Fig. 1: Nanoblock unit size.



(a) Available block sizes.



(b) Support block.

Fig. 2: Different types of Nanoblock.

safe assembly plan is found, the robotic manipulator joint trajectories are calculated using an architecture based on ROS and using MoveIt as online motion planner. Additional functions that adapt the simulated control signals to be used in the physical robot were implemented to execute the printing in real life. To improve the system reliability, a first study on how to substitute the empirical stability rules to a mechanical analysis approach was done in [12] and then improved and expanded in [13].

In this paper we present the following improvements made to the 3D block printer. First, the empirical judge was completely substituted by one that calculates the structure static balance of forces and torques during and after every block insertion. Also, all three stabilizing strategies had to be adapted to work with the new mechanical analysis when the algorithm detects an unsafe block. These two changes were made as it is a more generic approach that should satisfactorily be able to verify stability independent of new block type addition or system modifications, while the empirical approach would need to be changed constantly to take into consideration such changes.

Then, to integrate the new nonempirical assembly planner approach into our 3D block printer, a new robot control system had to be created. Aiming for longer support and use of the newest tools, this system was developed using Robot Operating System 2 (ROS2) to standardize the functions to the robot. It was also used MoveIt2 as the motion planning software with all its convenient functionalities like collision-avoidance and Cartesian trajectories. Finally, we achieved a complete 3D block printer that performs the assembly of a block model in a simulated environment following the safe assembly plan generated.

II. Building Blocks

In our 3D block printer, we use the "nanoblock", a product from Kawada Co. Ltd., as a building block. It is a toy brick with structure as in Fig. 1. The smallest block is 4mm wide, 4mm deep, 3mm high, and has a 2mm high cylindrical stud with a 2.5mm diameter. All blocks have a cavity space in their bottom where another block stud can be inserted to snap both blocks together. To simplify the denotation, a block is regarded as the number of studs columns $\times$ rows. Regardless of its size in X and Y axis, all blocks have the same height, denoted as one unit. Fig. 2a shows some of the available block types and their denominations.

All nonsquare blocks have pairs of transposed equivalents. For example, a 2×1 block is the rotation by 90° of a 1×2 and it is distinguished during assembly time. Although blocks placed in the same layer only share normal contact forces between themselves, when a block is inserted on top of other block, or the base, a friction force appear between the lower block stud and the upper block hole and, that way, holding the structure together.

The system can handle any block type, but the ones without symmetry in the X and Y axis; it is not possible to use, for example, L or T shaped blocks. Only the support blocks (Fig. 2b explained in section IV-B) have cylindrical shape and are transparent in color. As assembly is performed by a robot, each block is inserted from top to bottom and layer-by-layer labeled from 1 (bottom) to n (top). Also, assembly needs to be performed on top of a base with many studs that makes a connection with the bottom part of the first layer blocks giving more stability during assembly when compared with a plain surface.

III. From CAD Model to Block Model

The starting point of the 3D printer is to choose the desired CAD model in a stereolithography (STL) format and transform it into a set of voxels. This is currently done using a free Web service [14] that decomposes the object into cubic parts. Next, it is necessary to search for the best set of blocks that fills exactly the voxel model shape. This is executed as follows:

1) Initialize a list of layers K sorted from 1 to n.
2) Dequeue the first element k from K.
3) Initialize a list of possible blocks L and sort it in descending order of total number of studs.
4) Dequeue the first element l of L.
5) If l does not fit in the remaining voxels of layer k, go back to step 4. Else fix block l in this position, mark those voxels as complete.
6) If there are noncompleted voxels in k, go back to step 3. Else increment k by 1.
7) If $k > n$, finish. Else go back to step 2.

To increase the probability of placing a block making connection with a higher number of other blocks in the adjacent layers, proceed with the rules below in step 3 for all pairs of transposed blocks (e.g., 8×2 and 2×8) in L.

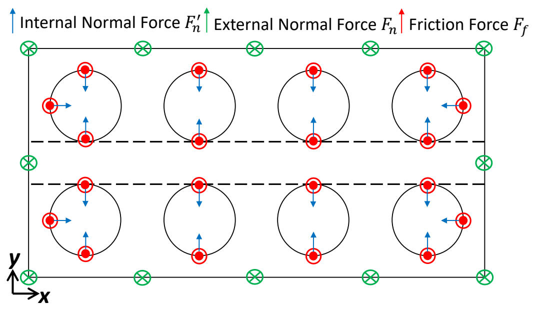


Fig. 3: Force model of a 4x2 block.

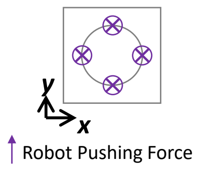


Fig. 4: Pushing force while inserting a 1×1 block.

- If k is even, place the block with more studs in the X axis before its transposed.
- If k is odd, place the block with more studs in the Y axis before its transposed.

That way, every layer will be filled giving priority to bigger blocks and with alternating preference to block orientation minimizing the required number of blocks in a layer and potentially increasing the connectivity between blocks with the lower and upper layers. Finally, the color is given manually to the generated block model. However, it is possible to create a program to automatically assign color to each block given the original 3D CAD model [15].

IV. Assembly Planning

Now that we have a block model, it is necessary to generate an assembly plan. The assembly planner is tasked to decide a block insertion order that guarantees that a robot can safely build the structure. It also needs to be able to handle the cases where stability cannot be guaranteed.

A. Mechanical Analysis

For every block in the input model, the mechanical analysis gets the current block size and a list of all blocks that are in contact with it. Then it determines all forces location and its limit value. Fig. 3 shows an example of a 4×2 block and all forces that could potentially appear in the Z axis. Fig. 4 illustrates how the robot insertion force appears in a 1×1 block. All blocks that are $x \times 2$ $(x > 2)$ and its transposed equivalents have a flexible partition in its inside part alongside the longer dimension that are weaker than the outside structure. This weaker part is showing as dashed lines in Fig. 3.

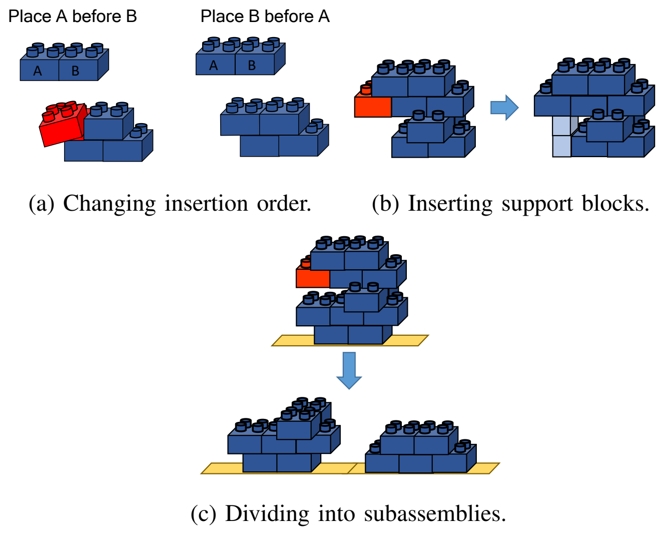
It would be hard to calculate the exact value of every force in complex structures with many blocks. To tackle this issue, a linear programming problem (LPP) was developed in [13] and is defined as in (1) to search for force distribution that maximizes the minimum "capacity" (C_m) [4]. This metric is defined as the difference between the maximum possible and the resultant friction force calculated for each block connection. A block connection is defined for each block stud that is snapped together with another block cavity. Positive capacity value means that the connection can still handle forces without breaking. Negative capacity value means that force is overflowing in that connection. Although theoretically a $C_m > 0$ would guarantee the safe insertion, we adopt a conservative approach and defined the minimum acceptable value of 20% of the maximum friction force measured for one connection to take into consideration modeling and measurement errors.

$$\underset{C_M, C_i, f_{f_i}, \boldsymbol{F}, \boldsymbol{F}_k, a_k}{\text{maximize}} \quad C_M = \min_i C_i \tag{1}$$
$$\text{subject to} \begin{cases} C_M \leq C_i = T - f_{f_i} & (i = 1, 2, \cdots, m) \\ \boldsymbol{W}_j \boldsymbol{A} \boldsymbol{F} = \boldsymbol{Q}_j & (j = 1, 2, \cdots, B) \\ \boldsymbol{F} = (\boldsymbol{F}_1^T, \cdots, \boldsymbol{F}_n^T)^T & \\ \boldsymbol{F}_k = \begin{cases} a_k \boldsymbol{t}_k & (k \in \mathcal{F}_f) \\ a_k \boldsymbol{n}_k & (k \in \mathcal{F}_n) \end{cases} & (k = 1, 2, \ldots, n) \\ a_k \geq 0 & (k = 1, 2, \ldots, n) \\ a_k \leq \boldsymbol{T}_{x\times 2} \quad (k \in \mathcal{F}'_{n,x\times 2}) & (k = 1, 2, \ldots, n) \\ f_{f_i} = \sum_{k \in \mathcal{F}_{f_i}} a_k & (i = 1, 2, \ldots, m), \end{cases}$$

where C_i is the i-th connection capacity; T is the maximum friction force a connection can have before snapping out; f_{f_i} is the i-th connection resultant friction force; m is the number of connections; $\boldsymbol{W}_j \boldsymbol{A}$ and $\boldsymbol{Q}_j$ are the j-th block wrench matrix and applied external wrench, respectively; B is the number of blocks; $\boldsymbol{F}$ is the force vector with every force in the model and n is the total number of forces; a_k is the k-th force magnitude; $\boldsymbol{t}_k$ and $\boldsymbol{n}_k$ are the unit tangential and normal forces of the k-th force, respectively; $\mathcal{F}_f$ and $\mathcal{F}_n$ are the sets of indexes of friction and normal forces, respectively; $\boldsymbol{T}_{x\times 2}$ is the maximum force possible in the contact points that are in the flexible parts of the block, while $\mathcal{F}'_{n,x\times 2}$ is the set of indexes of such forces; Finally, $\mathcal{F}_{f_i}$ the subset of friction forces from $\mathcal{F}_f$ that are applied in the i-th connection.

B. Dealing with Unassemblable Cases

During the planning, if structure stability cannot be guaranteed due to the LPP not being able to find a solution, the algorithm will try three different approaches to make it assemblable. The first would be changing block insertion order inside the current layer. Sometimes a simple change in the order can make an unassemblable shape assemblable. If yet not possible, the planner will try to insert support blocks below the blocks that are identified as unstable. That way, new connections between the block and the base will give more support and thus increase the structure stability. Another option is to divide the assembly into two or more subassemblies placing the unstable block directly in contact with the base.



(a) Changing insertion order. (b) Inserting support blocks.

(c) Dividing into subassemblies.

Fig. 5: Stabilizing strategies.

Priority is always given to the change order strategy, as it does not modify the original block model, then insertion of support blocks, as they are transparent and usually easy to remove. If the number of support blocks reaches an arbitrary threshold limit, the block model is divided into subassemblies. Fig. 5 shows all strategies in practice. In Fig. 5a, if block B is inserted first, the internal forces between the blocks are enough to hold the structure when block A is inserted. Fig. 5b and 5c illustrate both strategies solving the same unassemblable red block.

V. Robotic Control System

A robotic control system is necessary to execute the assembly plan generated using the mechanical analysis described in section IV. Our previous system [11] was developed using ROS Kinetic in an Ubuntu 16.04 machine. As many new features had to be integrated, a new robotic control system was developed from the ground. Although it has a large set of useful libraries to build robotic systems, ROS1 has shown some limitations mainly when dealing with delivering data over unreliable links and the lack of built-in security methods [16]. As ROS2 is developed and improved, it will become increasingly interesting to substitute the old platform with the newer version.

With that in mind, we decided to change the old platform, as both the ROS Kinetic and Ubuntu 16.04 does not receive any more support, to the ROS2 Humble Hawksbill running in an Ubuntu 22.04 (Jammy Jellyfish) as it those were the newest and with longer end of life date at the time. As the robotic manipulation platform, we used MoveIt2 Humble. MoveIt2 is a framework with many capabilities like motion planning, 3D perception, collision checking, inverse kinematics, control, manipulation, etc. [17]. It is also possible to use its visualization tool RViz2 to simulate the robot and its environment.

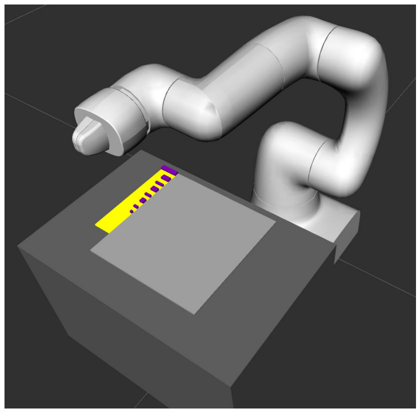


Fig. 6: Pick and place setup.

A. Robot Simulation

A simulated environment was created to test and verify the assembly process before implementing it using real robots. One of the advantages of using ROS and MoveIt to develop the system is the easiness of changing the utilized robot, the gripper or claws of the end effector and modifying the assembly setup. The robot utilized was the Denso Wave Cobotta (Fig. 6). Using RViz as interface, it is possible to visualize the MoveIt trajectory solution for every assembly step.

After spawning the desired robot, we can: verify the robot workspace and joint movements, create manually the robot start and final positions to see the movement, create Cartesian paths for the tool, and create paths that avoid all collision with the environment objects and with the robot links itself. By creating a ROS node to populate the simulation with many objects, it is possible to create the desired manipulation setup and all the blocks to be assembled by the robot.

B. Manipulation Task

To create an automatic and precise assembly process the development of a ROS node is required that implements the manipulation task for every block in the assembly plan. To do this, we used the Manipulation Task Constructor framework to create multiple interdependent subtasks making it flexible, transparent and simple to improve with new functionalities. The manipulation task starts with an initialization phase. It is as follows:

1) Initialize the robot in RViz2.
2) Initialize the assembly setup in RViz2.
3) Initialize every block available in the block feeder with a dummy color.
4) Read the assembly plan.
5) Open the gripper claws.

The assembly plan generated in the previous development dictates only the assembly order, position, orientation and color of each block. How it is going to be picked from the block feeder and how it is going to be placed at the requested position is up to the manipulation task algorithm. The only other requirement is that blocks must be inserted from top to bottom in the Z axis. Therefore, after the initialization phase, the following procedure is executed for every block in the assembly plan:

- Position the end effector above the block in the parts feeder.
- Close gripper claws.
- Move the robot end effector to a position above the insertion position facing the Z axis negative direction.
- Move the end effector in a straight line until the correct height position.
- Open the gripper claws.
- Retreat the robot from the structure.

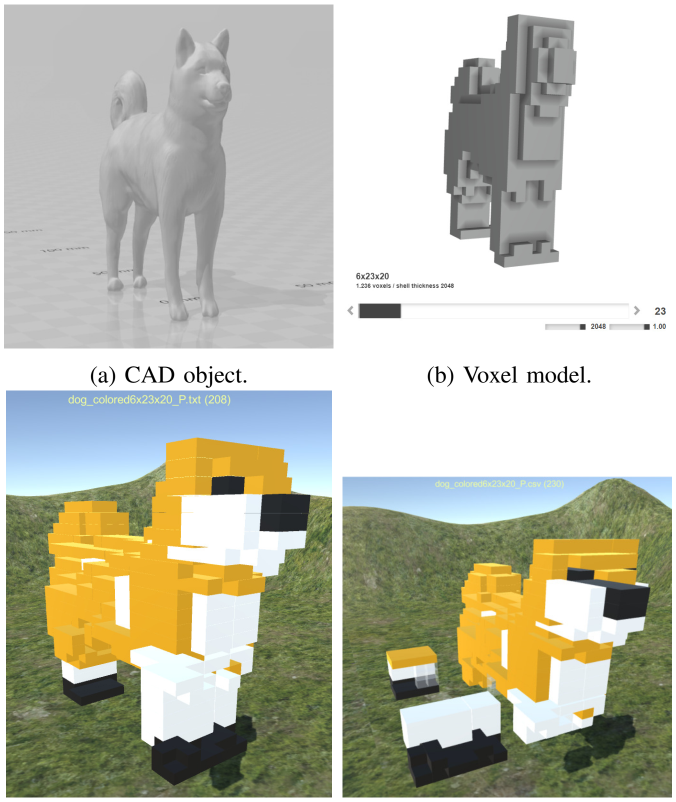
VI. Experiments and Discussion

To test the system, a simple working setup was created as in Fig. 6. There we have, the Cobotta robot, equipped with its default gripper, in white, the assembly table in dark gray, the assembly base in light gray, a block feeder in yellow where all block types are displayed and the available blocks in purple. For simplicity, only one block of each type is displayed with a dummy color. When a block in the feeder is picked up its color is corrected and, once the block placing is finished, a new dummy block is spawned in the open space. As this simulation does not have any dynamic phenomena, the blocks are created without its studs or cavities.

Demonstration of the system can be seen in Fig. 7 and 8, where the CAD models utilized (Fig. 7a and 8a) were downloaded in [18] and [19]. The correspondent voxels and initial block models are in Fig. 7b and 7c and in Fig. 8b and 8c. Fig. 7d and 8d show the visualization of the safe assembly plan generated. In the dog experiment it was not possible to assemble as it was due to its large stomach without contact points, so the planner decided to divide the assembly into two subassemblies and a few support blocks. The star model did not need to be divided and it was possible to assemble by using some support blocks in its lower part. Finally, we can visualize the blocks assembled by the robot in RViz2 in Fig. 9a and 9b.

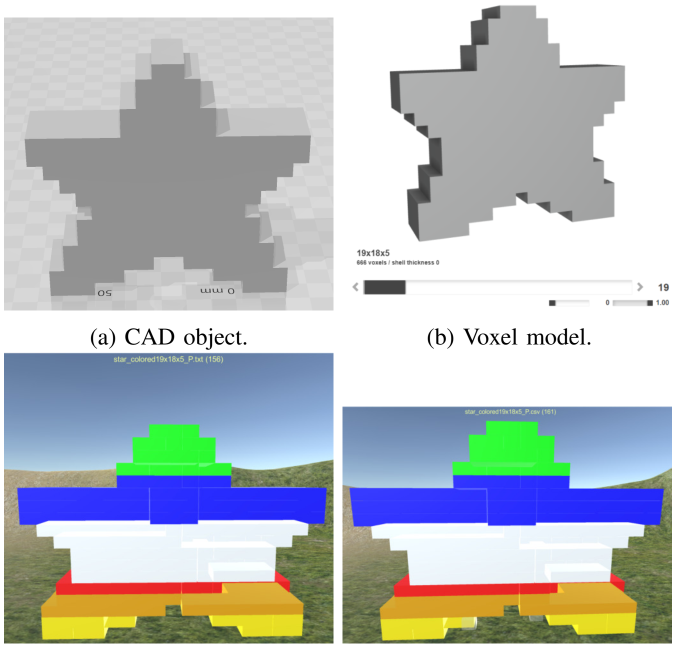
The system works well in simulating the assembly process while creating joint trajectories that avoid collision while placing every block in its correct position. Because the trajectory planning algorithm tries to find many different paths to solve a task, no block failed to be placed. The simulation was also capable of faithfully representing block shape and color including the support block. That way we were able to perform autonomous block assembly with a wider variety of building blocks than our previous implementation [11] using mechanical analysis as stability judge.

Table I compares the assembly planning results between the new system and the previous one. As a result of being able to use a wider variety of blocks size, both the dog and star examples reduced the total number of blocks by 50% and 38%, respectively. Regarding the number of support blocks and subassemblies, there is still room for improvement. The number of each of these two parameters will change drastically


6x23x20
1,236 voxels / shell thickness 2048
23
2048
1.00
dog_colored6x23x20_P.txt (208)
dog_colored6x23x20_P.csv (230)

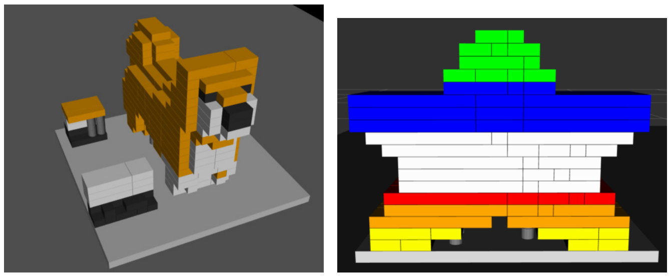
(a) CAD object. (b) Voxel model. (c) Initial block model. (d) Safe assembly plan.

Fig. 7: Dog model block printing.


50
0 mm
19x18x5
666 voxels / shell thickness 0
19
0
1.00
star_colored19x18x5_P.txt (156)
star_colored19x18x5_P.csv (161)

(a) CAD object. (b) Voxel model. (c) Initial block model. (d) Safe assembly plan.

Fig. 8: Star model block printing.



(a) Dog assembly simulation. (b) Star assembly simulation.

Fig. 9: Simulation output.

TABLE I: Assembly results for the Dog and Star block model

CAD Model	Dog		Star	
System Block Type	Previous	New	Previous	New
1x1	54	8	0	13
1x2	126	31	116	8
2x2	234	14	128	0
1x3	Not Available	22	Not Available	10
2x3	Not Available	5	Not Available	8
1x4	Not Available	61	Not Available	51
2x4	Not Available	36	Not Available	34
2x8	Not Available	30	Not Available	32
Support Blocks	24	22	14	5
Total number	507	229	258	161
Subassemblies	3	2	2	0

by modifying the arbitrary lower limit for capacity in a block connection, 20% in this experiments, and the parameter that determines that a subassembly is necessary, maximum of 15 support blocks per layer or a support column of 10 units.

A smaller number of small block types, and consequently the total number of blocks, has a direct impact in the assembly time making it much faster and more versatile. Bigger blocks also impact the stability of the structure as they can hold together more blocks and because of the possibility of having more area below them without contact points with other blocks and yet be assemblable, what would be impossible for smaller blocks.

VII. Conclusion

Our previous 3D block printer system [11] was limited in block types, due to relying on empirical rules to judge block insertion stability. With a new mechanical analysis approach to judge stability [13] bigger and more versatile blocks could be used. It was necessary to integrate the new judge into the 3D printer system and create a new robot control environment to calculate trajectory and then execute the assembly process in a simulated environment. These changes allowed our system to create assembly plans more reliable due to the mechanical stability calculation and faster due to the smaller number of total blocks needed.

In the presented study we achieved a novel system that autonomously assembles block models using a physics-based calculation approach to judge each assembly step stability in a simulated environment. The 3D printing steps are taking a 3D model input, transforms it into a block model, creates the safe assembly plan using mechanical analysis to judge safety, deals with unassemblable shapes by changing block order, inserting support blocks or dividing the assembly process, and, finally, autonomously execute the assembly plan using an industrial robot in simulated environment. As future work, we aim to create the real assembly setup, adapt the robot controllers to use the real robot, and improve the stabilizing strategies to search for better solutions. We also consider adding to the block printer post processing methods to increase the resolution of the assembled block models.

References

[1] Ngo, T. D., Kashani, A., Imbalzano, G., Nguyen, K. T., and Hui, D, “Additive manufacturing (3D printing): A review of materials, methods, applications and challenges,” Composites Part B: Engineering, vol. 143, pp. 172–196, 2018.

[2] Bhushan, B., and Caspers, M., “An overview of additive manufacturing (3D printing) for microfabrication,” Microsystem Technologies, vol. 23, pp. 1117–1124, 2017.

[3] Kim, J. W., Kang, K. K., and Lee, J. H., “Survey on automated LEGO assembly construction,” n, in Proceedings of WSCG 2014 Conference on Computer Graphics, Visualization and Computer Vision, pp. 89–96, 2014.

[4] Luo, S.J., Yue, Y., Huang, C.K., Chung, Y.H., Imai, S., Nishita, T. and Chen, B.Y., “Legolization: Optimizing LEGO designs,” ACM Trans. on Graphics, vol. 34(6), pp. 1–12, 2015.

[5] Hong, J. Y., Way, D. L., Shih, Z. C., Tai, W. K., and Chang, C. C., “Inner engraving for the creation of a balanced lego sculpture,” The Visual Computer, vol. 32, pp. 569–578, 2016.

[6] Hiller, J. D., and Lipson, H., “Fully recyclable multi-material printing,” International Solid Freeform Fabrication Symposium, pp. 98–106, 2009.

[7] Sekijima, K., and Tanaka, H., “Reconfigurable three-dimensional prototype system using digital materials” ACM SIGGRAPH 2015 Posters, pp. 1–1, 2015.

[8] Suzuki, R., Yamaoka, J., Leithinger, D., Yeh, T., Gross, M. D., Kawahara, Y., and Kakehi, Y., “Dynablock: Dynamic 3d printing for instant and reconstructable shape formation,” In Proceedings of the 31st annual ACM symposium on user interface software and technology, pp. 99–111, 2018.

[9] Gilday, K., Hughes, J., and Iida, F., “Achieving flexible assembly using autonomous robotic systems,” IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 6105–6110, 2018.

[10] MacCurdy, R., McNicoll, A. and Lipson, H., “Bitblox: Printable digital materials for electromechanical machines,” The International Journal of Robotics Research, vol. 33-10, pp. 1342–1360, 2014.

[11] Kohama, M., Sugimoto, C., Nakano, O., and Maeda, Y, “Robotic additive manufacturing with toy blocks,” IISE Transactions, vol. 53-3, pp. 273–284, 2020.

[12] Maeda, Y., Kohama, M., and Sugimoto, C., “3D Block Printing: Additive Manufacturing by Assembly,” IEEE/RSJ IROS 2019 Workshop on the current limits and potentials of autonomous assembly, 2019.

[13] Cesarino, P., and Maeda, Y., “A Force-based Assemblability Analysis for a Robotic 3D Block Printer,” Proc. of 40th Annual Conference of the Robotics Society of Japan, 1J1-01, 2022.

[14] Arjan, W. Convert your 3D model or image into voxels in your browser. Available at https://drububu.com/miscellaneous/voxelizer (accessed April 20, 2023).

[15] Kozaki, T., Tedenuma, H. and Maekawa, T., “Automatic generation of LEGO building instructions from multiple photographic images of real objects,” Computer-aided Design, vol. 70, pp. 13–22, 2016.

[16] Macenski, S., Foote, T., Gerkey, B., Lalancette, C. and Woodall, W., “Robot Operating System 2: Design, architecture, and uses in the wild,” Science Robotics, vol. 7, 2022.

[17] PickNik Robotics. MoveIt Humble. Available at https://moveit.picknik.ai/humble/index.html (accessed April 23, 2023).

[18] Thingiverse Dog by YahooJAPAN. Available at https://www.thingiverse.com/thing:182130 (accessed April 25, 2023).

[19] Thingiverse Pixel tree topper star by knape. Available at https://www.thingiverse.com/thing:194864 (accessed April 25 , 2023).