DupRobo: Interactive Robotic Autocompletion of Physical Block-based Repetitive Structure

Taizhou Chen[1], Yi-Shiun Wu[2], and Kening Zhu[1][0000−0001−6740−4921]

[1] School of Creative Media, City University of Hong Kong, Hong Kong
taizhou.chen@my.cityu.edu.hk, keninzhu@cityu.edu.hk
[2] Department of Micro Engineering, EPFL, Switzerland
alan830908@gmail.com

Abstract. In this paper, we present DupRobo, an interactive robotic platform for tangible block-based design and construction. DupRobo supported user-customizable exemplar, repetition control, and tangible autocompletion, through the computer-vision and the robotic techniques. With DupRobo, we aim to reduce users' workload in repetitive block-based construction, yet preserve the direct manipulatability and the intuitiveness in tangible model design, such as product design and architecture design. Through a user study with 12 participants, we found that DupRobo significantly reduced participants' perceived physical demand, overall efforts, and frustration in the process of block-based structure design and construction, compared to the situation without DupRobo. In addition, the participants rated DupRobo as easy to learn and use.

Keywords: Block assembly; Robotics; Autocompletion; Tangible user interface; Programming by demonstration; Tangible programming.

1 Introduction

Assembly blocks (e.g. LEGO) have been widely applied in various creative areas, such as product design [17] and architecture design [27]. Different from sketching which is thought of as 2D visual design thinking [6], physical block building, with its emphasis on assembly and manipulation, ought to be considered 3D physical design thinking, a more tangible, interactive way of exploring design [27].

However, it is still confusing and tedious for non-experienced user to construct large physical models through the piece-by-piece block-building process [28]. It is even more difficult to create new models from scratch. Several researches have been done to tackle this problem, via the automatic generation of block-building instruction [18, 21, 35] and automatic construction of digital design with robots [9, 10, 19, 26]. However, automatic instruction generation still requires users to build the structure by themselves. Although robotic automatic construction could reduce users' workload on physical building, most of them required the software-based modelling process, in which the complex modelling procedure disconnected users from the material investigation of the actual artefact being designed [15], leading to the lack of creative artistry or craftsmanship [5]. In addition, designing 3D models with CAD software requires additional expertise to organize and structure operations [16]. Research further indicated that interacting with a 3D model virtually can be far less intuitive than actually making the physical model [12, 37], and tangible interfaces supported better problem-solving process than the software interfaces did [24,41]. In addition, tangible artefacts and interfaces have demonstrated their benefits to design iteration [4], digital fabrication [39], digital entertainment [38,40], and medical service [29].

Many large complicated structures often involve many repetitive parts. For example, the Parthenon model contains multiple similar pillar structure, and the Great Wall model consist of many embattlements. Meanwhile, it is observable that highly creative patterns can be generated through controlling the repetition of a primitive exemplar. The exemplar repetition, also knowns as autocompletion, has been widely supported in many 2D/3D design software [13, 23, 33, 34]. Considering the physical block-building process, there is a need to reduce the manual workload in repetitive physical block-based construction through physical autocompletion yet preserve the direct manipulability and the intuitiveness in tangible modelling.

In this paper, we present DupRobo (Fig. 1), an interactive robotic platform for tangible block-based design and construction. Inspired by the drawing autocompletion, DupRobo supported physical exemplar creation, repetition control, and physical autocompletion. The setup of DupRobo consists of a KinectV2 sensor above the workbench platform to track the user input (i.e. the exemplar building and the marker-based commands), a robotic arm for automatic construction, and a block inventory management mechanism. Our user study with 12 participants showed that DupRobo significantly reduced participants' perceived physical demand, overall efforts, and frustration in the process of block-based structure design and construction, compared to the situation without DupRobo. In addition, the participants rated DupRobo as easy to learn and use, and they commented that DupRobo could assist them to achieve their desired structures.

2 Related Work

DupRobo was inspired by the existing works on the 2D drawing autocompletion, the block-assembly tracking and generation, and the robot assistant system.

2.1 Software-based Autocompletion: 2D & 3D

The research on automatic repetitions of visual patterns has been greatly advanced in recent data-driven methods [13, 33]. These works impose a list of sequential orders with the user-defined exemplar to be cloned to the desired output region through various gestural commands, such as brushes. Similar autocompletion techniques were recently applied on 3D surface sculpting [23], which support brushing and stippling for modeling repeated part such as tentacles, hairs, and repeated texture on meshes. Although the autocompleted 3D virtual surface can be physicalized through 3D printing, it still requires users to edit the 3D model in graphical user interfaces which could be less direct or intuitive than the tangible user interface. Moreover, the autocompletion concept was also used on animation autocompletion, Xing et al. [34] introduced a method that simplify the process of creating frame-by-frame animation through manual sketches. It allows

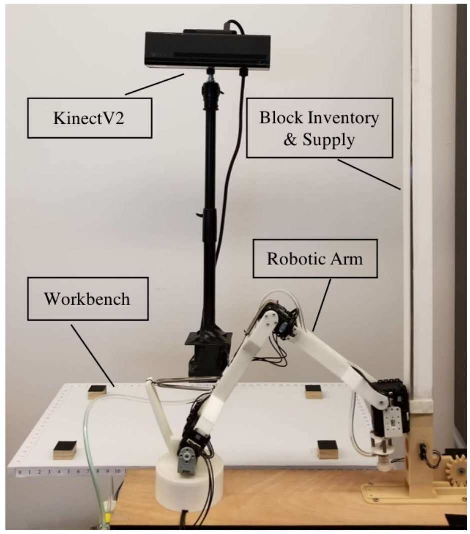

Fig. 1. System setup of DupRobo.

users to define character animation such as motion and trajectory easily by adding some command-based strokes. Taking one step further, we implement the concept of autocompletion into a fully tangible interface, with DupRobo supporting the physical block-based modeling autocompletion with tangible commands.

2.2 Block-Assembly Tracking

There have been several researches on tracking the procedure of block assembly. Miller et al. [20] and Van de Leemput at al. [30] both developed Kinect-based systems to track the construction of LEGO models. Their methods assumed that the physical model always stays with the base on the table, to reduce the complexity of tracking. Gupta et al. [7] presented Duplotrack, a real-time system to track the assembly process of Duplo blocks in 6 degrees of freedom. They used one Kinect sensor, thus requiring the physical model to be continually rotated to conduct 360-degree scanning. Hsieh et al. [11] introduced RFIBricks, a system that use ultrahigh frequency radio-frequency identification for block-based structure recognition. This method requires addition components, such as RFID tags on each block, thus increasing the cost and the size of the block. To make a low-cost system as well as taking consideration of the block size, DupRobo adopted the Kinect-based tracking method which is similar to [20, 30], for both block-structure tracking and command-marker recognition.

2.3 Block-based Structure Generation & Construction

Besides block-assembly tracking, several researches were conducted on block-based structure generation and automatic construction. Kim et al. [14] presented a thorough literature review on automated LEGO assembly construction. Luo et al. [18] introduced Legolization, consisting of a force-based analytic algorithm and a layout refinement algorithm that allows automatic generation of a LEGO brick layout from a given 3D model. Mueller et al. [21] developed faBrickation, a rapid-prototyping method that combined hand-made LEGO structures and 3D-printed parts for functional objects. More recently, Yun et al. [35] developed a Legorization framework that produces a LEGO model through voxelization from user-specified 3D mesh model.

To automatically build block-based structures, Sekijima et al. [26] developed a reconfigurable 3D prototyping system that that can automatically assemble the octahedron-shape blocks with embedded magnetic joints. Hiller and Lipson [9] introduced a method that 3D-printed a model in voxel with advantages of perfect repeatability and supporting multiple materials. Hiller developed a “VoxJet" printer using spherical voxels [10]. Maeda et al. [19] have developed a 3D-block printing system that allows reconstructing 3D CAD models into physical block-based structures. Their system can automatically convert a 3D model to a block structure that consisting of primitive LEGO blocks, then trigger the robot for automatic assembly. Although these systems support automatic block building with robotic technology, they still requires users’ software-based model design, which require extensive learning and practice. Research have proved that tangible modelling interfaces helped users achieve better learning and task performance than pure software interfaces did [24]. DupRobo leverages these advantages of tangible interfaces, offering an intuitive physical prototyping environment.

2.4 Robotic Assistant

DupRobo is also strongly inspired by the recent development in interacting with robotic assistants for industrial and in-home purpose. Zhao at al. [36] utilized AR markers to control house-keeping robots. Orendt et al. [22] validated the intuitiveness and robustness of a One-Shot programming-by-demonstration robotic system. More recently, Sefidgar et al. [25] developed a set of physical blocks with visual markers for robot programming in a pick-and-place task context. Their studies proved the high intuitiveness and learnability of situated tangible programming for robotic assistants. Thus, the similar interaction techniques were adopted in DupRobo. Wang et al. [32] proposed a framework for describing human-robot collaboration. Van den Bergh et al. [31] developed a low-cost robot setup for collaborative personal fabrication activities in Fab Labs and Makerspaces. Smithwick et al. [27] envisioned an intelligent robotic platform that assists architects in tangible prototype design. DupRobo was directly motivated by this vision and stepped further with tangible control interface for the robot. More importantly, DupRobo distinguished itself by offering tangible modelling and control interfaces which could be more intuitive than the graphical user interfaces.

3 DupRobo

DupRobo is an interactive robotic platform aiming to support tangible block-based repetitive structure design and construction. It allows users to design and construct complicated repeated assembly block architecture in a few simple steps from scratch. There are three key features in DupRobo:

Firstly, DupRobo supports tangible design with physical blocks and command markers, motivated by the proven benefit of tangible interfaces on problem solving and creative design processes [24]. Many repeated block-based structures can be represented by: a single element repeats according to certain rule. Thus, we decoded repetitive block structures into two parts: the exemplar and the repeat rules. We designed a set of tangible marker blocks that define specific rules for repetition in an intuitive way, allowing users to easily explore and create their own repetitive block structures by arranging and combining the placement of different markers.

Secondly, DupRobo provides an integrated workbench-type environment by connecting the virtual and physical world with our preview system. To reduce the gap between the virtual and physical world during the design process, every user input (i.e. exemplar construction and marker placement) will be captured, calculated, and showed on the screen in real time. Our preview system also supports navigation operation, allowing users to preview and understand the current state of the generated structure from different perspectives in 3D space.

Lastly, DupRobo emphasizes on human-robot collaboration. We aim to use robotic arms to assist human for constructing complicated and tedious block structure. Human-robot collaboration is defined as: a sequence of interdependent actions in a Human-Robot interaction setting towards a shared goal [3]. DupRobo builds physical blocks based on the user's exemplar and physical commands. Therefore, users and DupRobo collaborate and contribute to the complete design and construction pipeline.

3.1 Hardware

Workbench The workbench consists of a wood platform and a block inventory. The wood platform (60cm $\times$ 30cm) was drilled by the laser cutter with 18 rows by 37 columns, in total 666 holes with the diameter of 3mm and the depth of 2mm to embed the neodymium magnets. The distance between two neighboring magnets was 15mm. The platform was attached on a sliding mechanism with a 200-step stepping motor for the vertical movement. Every 200 steps movement of the stepping motor moves the platform 5 mm vertically.

The block inventory/supply mechanism was consisted with a 3D printed gear structure with an embedded servo motor (Model No.: MG-996R). A switch was attached on the storing compartment to detect the level of blocks. Once the number of the blocks is low, the LED will light up as an indication to notify the user. When there is no block left, a loud buzz will warn the user and pause the process until the user refill the blocks.

Building Blocks As shown in Fig. 2, the 3D-printed block sizes 28mm by 28mm and 15mm in height. We designed the size of the block slightly smaller than the gap between two adjacent magnets on the platform, a smooth placement by robotic arm. There were 4 neodymium magnets square-distribute embedded in both the top and bottom side of the block.

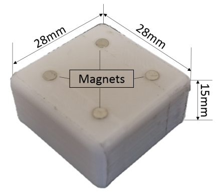

Fig. 2. DupRobo building block.

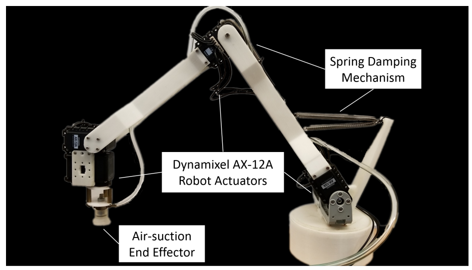

Fig. 3. DupRobo robotic arm.

Robotic Arm The robotic arm (Fig. 3) consists of six Dynamixel AX-12A Robot Actuators, an air-suction cap as the end effector, and the 3D printed frames with springs as the damping system to reduce the load of and ensure the stable movement of the actuators. To grip a block, the system turns on the air pump and the valve by the relay-controlled switch circuit. To build a block, the microcontroller receives the 3D coordinate of the to-build block, and calculate the real-time movement of each actuator based on the inverse kinematics and a real-time PID control algorithm. The air pump creates a vacuum state of the suction cap and enable it to hold the block. The valve will release air into the suction to release the block once reaching the targeted coordinate.

Command Makers We designed three marker-base commands: *Anchor* marker, *Up* marker, and *Go* marker, as shown in Fig. 4.

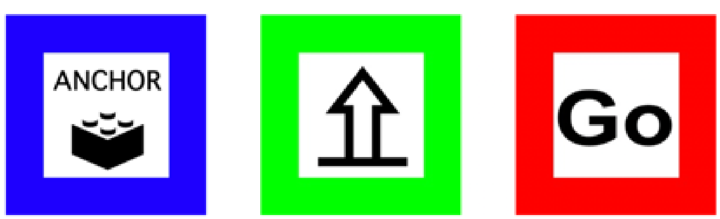

Fig. 4. (left) Anchor marker; (middle) Up marker; (right) Go marker.

The *Anchor* marker is the main command in DuoRobo. With the *Anchor* marker, DupRobo will generate the structure that lines up between the last exemplar-building position and the marker position using the exemplar (Fig. 5(b)). If one or more *Anchor* markers already exist on the platform, the newly placed *Anchor* marker will start lining at the end point of the last *Anchor* marker's operation and end the line at the current *Anchor* marker (Fig. 5(d)).

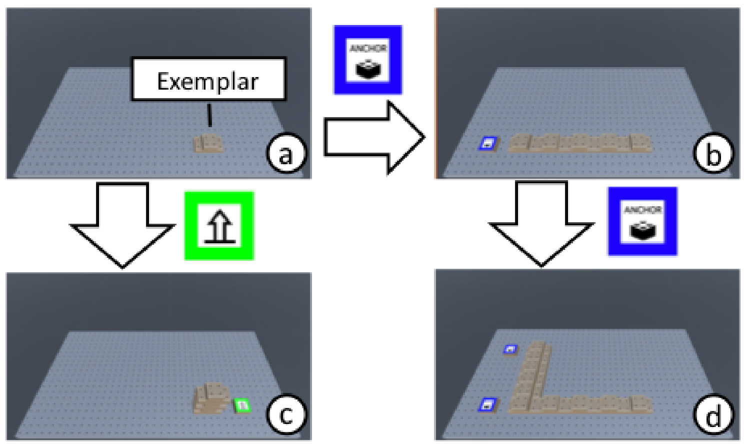

Fig. 5. (a) Workbench without any marker, (b) Workbench with the first *Anchor* marker, (c) Workbench with the first *Up* marker, (d) Workbench with more *Anchor* markers.

The *Up* marker will repeat the exemplar toward the up direction twice (Fig. 5(c)). It is also possible to set the repeat time as a controllable parameter, which means that we can have an additional marker associate with the *Up* marker that control the repeat time. In the current prototype as the proof of concept, we set the repeat time with a constant value of two, due to the limited building space.

Combination of the *Anchor* marker and the *Up* marker provides diversified outputs. If an *Up* marker was set after one or more *Anchor* marker (Fig. 6(b)), the exemplar will be extended linearly upward from the start position to the last *Anchor* marker (Fig. 6(c)), instead of building upward perpendicularly. On the other hand, if the anchor marker was set after the *Up* marker (Fig. 6(d)), then the up operation will be conducted first and the result structure will be set as repeated element before the *Anchor* marker's operation using the new exemplar (Fig. 6(e)).

3.2 Software

The software of DupRobo consists of two main parts: block/marker-placement recognition, and repetition generatione/preview.

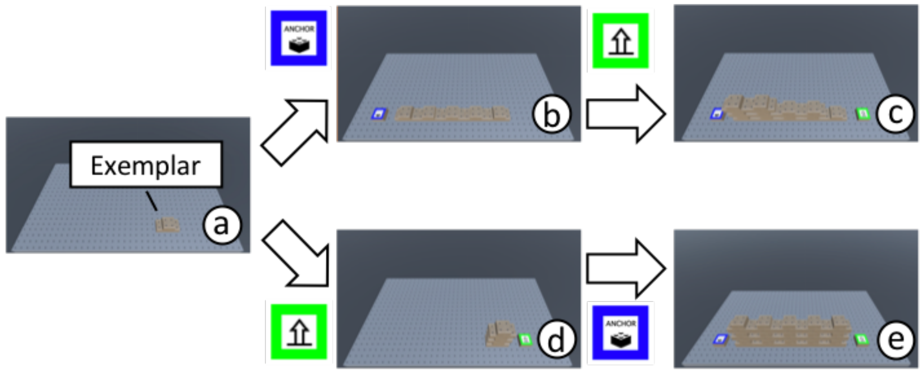

Fig. 6. (a) Workbench without any marker, (b) Workbench with the first *Anchor* marker, (c) Workbench with the *Up* marker after the *Anchor* marker, (d) Workbench with the first *Up* marker, (e) Workbench with the *Anchor* marker after the *Up* marker.

Block/Marker-Placement Recognition The recognition part was developed using C++ with the Kinect library and the OpenCV library. We used the depth frame for the block-placement tracking, and the RGB frame from the KinectV2 for the marker detection. Since the relative position between Kinect and the workbench platform are stationary (40cm vertically), and the possible positions for the block placement are fixed with the embedded magnets' positions on the platform, we mapped each possible position to the 3D coordinate in the depth frame. Therefore, we can track each block's position on the platform according to the 2D coordinate in the depth frame. We can also get the height of that particular position from the depth information. To increase the tracking accuracy, we utilize four sets of 2D coordinates to track one block. Once detecting the simultaneous depth changes (i.e. the change of grayscale color in the depth frame) at two diagonal points, the system triggered the event of block-placement detection. Fig. 7 illustrates an example of block-placement detection for two blocks at different height levels ((X_i, Y_j) denotes the coordinate of the block, and (p_i^x, p_j^y) denotes the coordinate of the to-scan pixel).

For marker detection, the system obtained the marker mask by calculating the L2 distance between the pre-defined RGB value of a particular marker and each pixel in the current RGB frame. The system then performed the rectangle-detection process on the marker mask to retrieve the center of the detected rectangle as the marker's position respectively.

Repetition Generation & Preview Given the structure and the position of the exemplar, and the type and the position of the marker, the preview program calculated the positions of to-build blocks, which is done by a vector-base algorithm. For the *Anchor* marker, we adopted a recursive process, *anchorRepetition*, as described in Fig. 8. For the *Up* marker, the preview part naively calculated the positions of the to-built blocks towards the up direction.

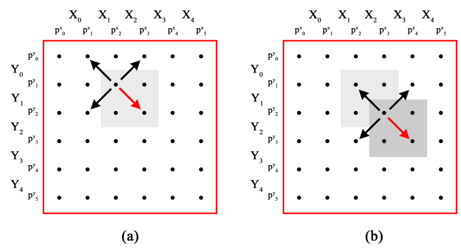

Fig. 7. (a) If depth change was detected at point (p_2^x, p_1^y), then scan through (p_1^x, p_0^y), (p_3^x, p_0^y), (p_1^x, p_2^y), (p_3^x, p_2^y), as indicated by the black arrows. Another detection of depth change was found at (p_3^x, p_2^y), as indicated by the red arrow, thus it is selected as a block placement at layer 1. (b) If there is another depth change at (p_3^x, p_2^y), we conduct the same diagonal-point-scanning process to detect the block at (X_3, Y_2) at layer 2.

4 Walkthrough

4.1 Step 1: Ideation and Exemplar Building

Users firstly decide what they want to build with DupRobo before considering the shape of the exemplar and the repetitive structure. For better recognition performance as well as for distinguishing users' input blocks, we use wooden blocks for creating the exemplar.

Use case: A master student on architecture design, Alan, would like to build a close area with blocks that look like the wall of an ancient castle. He decided to use 3 blocks to create a reversed T-shape structure as the exemplar. As he built the exemplar by placing blocks on the workbench, DupRobo software executed the recognition process after each block placement, and rendered the blocks in the preview (Fig. 9).

4.2 Step 2: Marker Placement

After creating the exemplar, users need to place markers on the workbench for generating the to-build repetitive structure. For every camera frame, DupRobo software will detect whether there are markers on the platform or not. Once detected a particular marker, the system will generate and render the repetitive structure in the virtual environment for preview.

Use case: Alan then started to place marker on the workbench. Since he would like to add some height for the wall structure, he first put an *Up* marker on the platform. From the preview system he saw the exemplar was repeated perpendicularly twice (Fig. 10(a)). He then put an *Anchor* marker on the right of the exemplar, as he wanted the expand the exemplar to the right. From the preview system, he saw what he expected (Fig. 10(b)). He then put the second *Anchor* marker in front of the first one (Fig. 10(c)), and lastly put the third *Anchor* marker on the left of the second (Fig. 10(d)). He saw the structure repeating the exemplar to the second and the third *Anchor* marker correctly on the preview screen.

Algorithm: anchorRepetition

anchorRepetition (*E, M*)
Input: A list *E* of blocks of the exemplar, marker *M*
Output: A list *R* of repeated blocks

get Distance *D* from *E.position()* to *M.position()*
get Direction *d* from *E.position()* to *M.position()*

loop through the position *p* for each potential placement of the repeated exemplar *E'*, given the *E'.boundingBox()* is adjacent to *E.boundingBox()*; and select position *p'* whose direction to E has the smallest angle to *d*.

get Distance *d'* from *p'* to *M.position()*

if ($d' < D$)
    instantiate R_{next} at position *p'*
    anchorRepetition (R_{next}, *M*)
else
    return *R*

Fig. 8. Psuedocode of repetition generation for the *Anchor* marker.

Fig. 9. Physical exemplar construction.

4.3 Step 3: Correction and Fine-tuning

DupRobo contains a buildability checking process for each marker detection, to ensure that the physical autocompletion process could be accomplished. In the current prototype, we prepared in total 64 3D-printed blocks with neodymium magnets. DupRobo software will calculate the number of blocks needed to build the repetitive structure. While there needs more than the current total number of blocks (i.e. 64) for building, the exceeding part will be rendered in a semi-transparent manner in the preview window. If there is a collision detected between the newly generated block and the previous block, the newly generated block would be rendered in red for warning.

Use case: After placing the third anchor marker, Alan noticed that some blocks in the preview system are semi-transparent (highlighted in the red box in Fig. 10(d)), meaning that the current set of blocks is not enough for DupRobo to finish the physical autocompletion. He did not want this happen so he decided to reset the marker. He then removed the *Up* marker from the workbench (Fig. 11(a)), and placed the fourth *Anchor* marker to close the shape (Fig. 11(b)). However, he noticed that subsequently there are some blocks become red on the preview window, indicating collision. This was caused

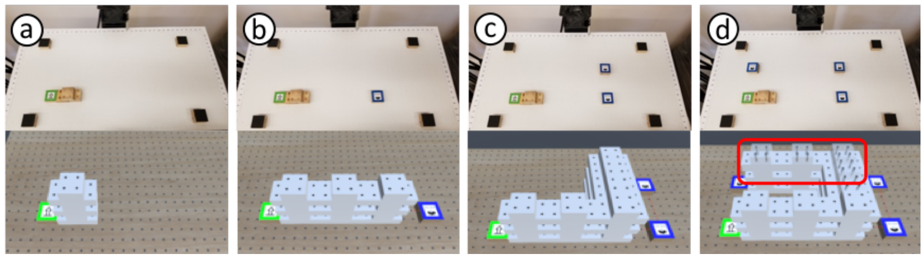

Fig. 10. (a) Place an *Up* marker, (b) - (d) Place the first, second, and third *Anchor* marker.

by the wrong position of the fourth marker. Therefore, he adjusted the position of the fourth marker to form a perfect close area (Fig. 11(c)). Lastly, he placed a up marker to generate a stair-like wall structure (Fig. 11(d)).

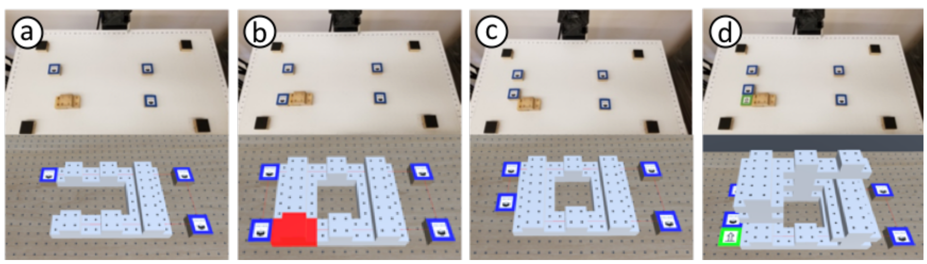

Fig. 11. (a) Correction by removing the *Up* marker, (b) Place the fourth *Anchor* marker and collision was detected, (c) Adjust the placement of the fourth *Anchor* marker, (d) Lastly, place the *Up* marker.

4.4 Step 4: Physical Autocompletion

Users can trigger the autocompletion process by placing the *Go* marker. DupRobo software will check the buildability again before triggering the robotic arm. In order to prevent from physical collision during the physical construction process, the architecture will be constructed from left to right meanwhile from nearby to faraway.

Use case: Alan was satisfied with the generated structure. He put the go marker to trigger the physical autocompletion process. Finally he got a will-built structure as what he saw in the preview system (Fig. 12).

5 Workshop Study

To evaluate the usefulness and usability of DupRobo, we conducted 10 individual workshops with an emphasis on the following questions:

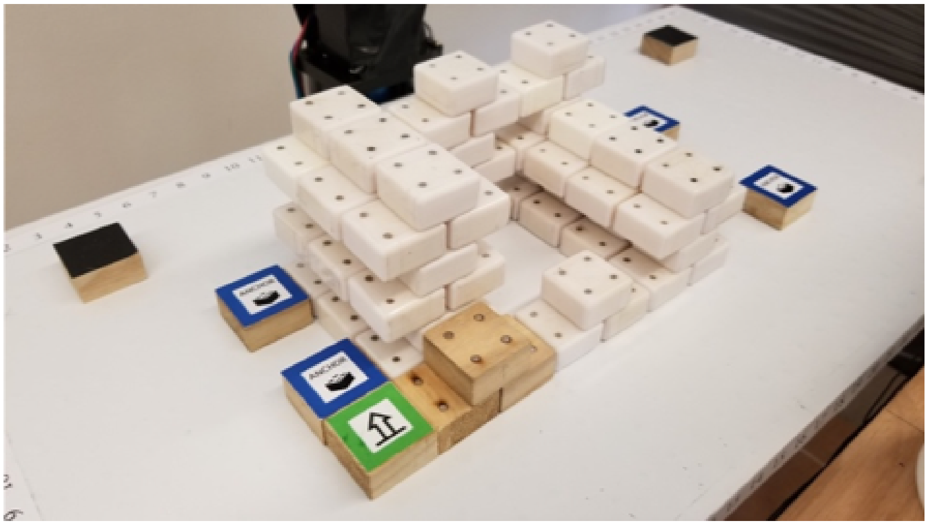

Fig. 12. Final product constructed by DupRobo.

- How do users evaluate the intuitiveness and learnability of DupRobo?
- Do users find DupRobo useful and engaging?
- How does DupRobo reduce the workload of repetitive block building?

We have adopted the evaluation strategy and procedure followed by other creative systems [2, 39, 42] for our evaluation.

5.1 Participants

Our workshops had a total 12 participants (one per workshop) consisting of five males and five females with ages ranging from 22 to 33 years (M=27.3, SD=4.63). The workshops were held in a research laboratory with a dimension of 10m × 7m. Prior to conducting the workshops, we recorded the information on each participant's professional background (five on computer graphics & rendering, three on interface design, and two on product design).

5.2 Apparatus

Each participant worked with a DupRobo system, which consisted of hardware (the workbench, the robotic arm, the 3D-printed blocks with embedded magnets, and the marker-based command blocks) and a software interface for result preview installed in a Dell Optiplex 990DT desktop PC.

5.3 Procedure

The workshop was conducted in four sessions:

1. Introduction. (10 minutes) The workshop facilitator gave a brief introduction of DupRobo, and showed a few examples that can be made using DupRobo, through powerpoint slides.

2. Guided Task. (15-20 minutes) After the briefing, the participant was given a printed tutorial on how to make a repetitive structure that can be auto-completed by DupRobo. The participant were asked to recreate this example to familiarize themselves with DupRobo. The activity involved creating the exemplar from scratch, placing the command markers, and triggering the robotic arm for auto-completion.

3. Free Task. (30-40 minutes) The participant was asked to explore their creativity and imagination by creating a new repetitive structure. This session aimed to provide us insights on how DupRobo allows users to explore their creativity. In addition, once satisfied with his/her design, the participant needed to construct one physical model of the design by him/herself, besides triggering the robotic arm. The participant constructed his/her design manually while DupRobo was building. This was to compare users' workload on repetitive structure building with and without DupRobo.

4. Demo. (10 minutes) After make his/her own structure, the participants was asked to present their design to the facilitator and explain the design rationale.

The workshop process was video recorded with the participants' consent. After the workshop, the participant answered a user-experience questionnaire on their impressions of the system (1 - strongly disagree, 5 - strongly agree). In addition, the participant was asked to two copies of a modified version (1 - very low, 7 - very high) of NASA TLX questionnaire [8] on their perceived workload with and without DupRobo.

5.4 Results

User Evaluation of DupRobo's Intuitiveness and Learnability Table 1 shows the detailed ratings of the user-experience questionnaire from all the participants. Participants found DupRobo's marker-based command interface intuitive and easy to learn. One participant reported that it was *"easy to get used to the system after the guided task"*. Other participants commented: *"It was interesting to see the robot following my commands"*, that DupRobo can be easily and quickly understood, and that *"it was like having a robot as a building assistant"*. The results of the questionnaire showed that intuitiveness earned a score of 4.25/5 while the learnability scored 4.67/5.

All the participants were able to finish the guided task within the allotted time of 20 minutes. In the 40-minute free task, the participants were able to come up with different structures and implement them through the autocompletion by DupRobo. They were allowed to ask questions when they faced difficulties, but very few did. There were, at most, two questions asked during each of the 12 workshops, which suggested that the toolkit was self-explanatory.

Nonetheless, the questions posed did help us to identify minor usability problems of the interface. Examples were *"Can I use the structure created by the robot as a new exemplar?"*, *"Can I change my design while the robot is building?"*, *"Can the system give any suggestion on resolving the collision?"*, and *"Can I have blocks in different shapes, such as circle and triangle? It seems I only have cube-shape."*, indicating that the reuse function, the recommendation function, and the variety of blocks can be improved in our current system.

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	Mean	SD
It is easy to learn to use this toolkit.	5	4	5	5	5	5	4	5	4	5	4	5	4.67	0.49
It is easy to use this toolkit to create the physical structure that I want.	4	5	4	4	4	4	5	5	4	4	4	4	4.25	0.45
It is easy to use the software to design and plan the physical structure.	4	4	4	5	5	4	4	5	4	4	3	5	4.25	0.62
This toolkit is useful in creating physical block structures.	5	4	4	5	4	4	3	5	4	4	4	5	4.25	0.62
Making physical block structures with this toolkit is fun.	5	4	4	5	4	5	4	4	2	5	4	5	4.25	0.87
I enjoy creating physical block structures using this toolkit.	5	4	4	5	4	4	4	5	2	5	4	5	4.25	0.87
I became creative in creating physical block structures with this toolkit.	4	3	5	4	4	4	3	5	4	4	4	4	4.00	0.60
I became productive in creating physical block structure with this toolkit.	5	4	4	5	5	5	4	4	5	4	5	4	4.5	0.52
I would recommend it to my friends.	5	3	4	5	5	3	3	5	4	4	4	5	4.17	0.83

Table 1. Participants' ratings of the user-experience questionnaire. P#: Participant ID.

Fig. 13. The repetitive structures created by the workshop participants.

User Opinions on DupRobo's Usefulness and Capacity to be Engaging Workshop participants unanimously agreed that the toolkit is useful (4.25/5) and that it can be employed in the following: architecture design; product design; furniture design; and pure entertainment. In terms of the creative process, we observed that one user made a few drafts outside of the board before he put the exemplar and the marker on the platform. Two users commented that they can mentally visualize the end results based on the markers. The rest of the users mentioned that they didn?t have a clear idea on what they could build at the beginning, and the markers and the preview function assisted them to explore different ideas through “try and error”. They tended to refer to the visualization on the screen, to decide the next step of design. The statement of “I became creative in creating physical block structures with this toolkit." was averagely rated 4/5.

Overall enjoyment scored 4.25/5. It was observed that enjoyment increased when the participants were allowed to be creative in the free task. They were excited by the opportunity to create repetitive structures with DupRobo. Fig. 13 shows the examples of the products created by the participants during the free tasks. Participants generally liked the toolkit, and most of them agreed (4.17/5) that they wanted to recommend it to their friends.

Workload Reduction and Productivity Support The average total NASA-TLX score with DupRobo is 17.67/42 (SD = 4.94), and the average total score without DupRobo is 26.25/42 (SD = 8.24). Table 2 shows the detailed ratings of the NASA-TLX questionnaire from all the participants. Wilcoxon Signed Ranks Test showed that the participants rated significantly less workload ($Z = 2.675$, $p < 0.01$) with DupRobo than without DupRobo. For the individual items of NASA TLX questionnaire, Wilcoxon Signed Ranks Test showed that the participants rated significantly less physical demand ($Z = 2.715$, $p < 0.05$), overall effort ($Z = 2.537$, $p < 0.05$), and frustration ($Z = 2.132$, $p < 0.05$) with DupRobo than the process of building by themselves without DupRobo. One participants explicitly commented that *"It reduces tedious work"*. Furthermore, the participants rated 4.5/5 on the statement "I became productive with DupRobo", although the building speed of the robotic arm was generally slower than the speed of the participants themselves.

		P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	Mean	SD
Mental Demand	With DupRobo	5	4	2	3	2	2	4	2	3	6	5	2	3.33	1.43
	Without DupRobo	6	3	1	5	6	5	3	7	5	4	3	6	4.50	1.73
Physical Demand	With DupRobo	2	1	2	3	2	1	2	2	1	2	2	1	1.75	0.62
	Without DupRobo	5	1	1	5	5	5	6	7	7	5	6	5	4.83	1.95
Temporal Demand	With DupRobo	6	1	1	3	5	1	2	2	1	6	4	2	2.83	1.95
	Without DupRobo	5	1	1	5	3	1	3	7	6	5	4	2	3.58	2.07
Effort	With DupRobo	4	1	2	3	3	2	2	2	3	3	3	1	2.42	0.90
	Without DupRobo	6	3	1	5	4	5	3	7	5	5	4	4	4.33	1.56
Performance	With DupRobo	6	3	6	4	3	6	3	7	6	4	5	5	4.83	1.40
	Without DupRobo	6	3	6	4	5	5	5	4	4	4	5	6	4.75	0.97
Frustration Level	With DupRobo	5	1	2	3	2	2	4	2	2	3	3	1	2.50	1.17
	Without DupRobo	7	1	1	5	3	2	6	7	6	5	3	5	4.25	2.18
Total Score	With DupRobo	28	11	15	19	17	14	17	17	16	24	22	12	17.67	4.94
	Without DupRobo	35	12	11	29	26	23	26	39	33	28	25	28	26.25	8.24

Table 2. Participants' ratings of the NASA-TLX questionnaire. P#: Participant ID.

6 Limitation & Future Work

Despite the capabilities we have shown, the current prototype of DupRobo could be improved in several aspects.

6.1 Variety of Command Markers

While the three markers in our current prototype could cover most basics repetitive block-based structures, there is a need for more variation of marker-based commands to support more complicated structures. For example, we currently defined the *Up* operation with a default parameter of two as the proof of concept, due to the limited space for construction. We could have another marker associate with the Up marker to control the time of vertical repetition.

In our current prototype, the orientation of the exemplar repetition was fixed according to the user input. It is possible to design another command marker, in the future work, that control the direction of the autocompleted part, to support more diversities for the final product. In addition, we would include the grouping operation for the future version, which allow users to group the current repeated structure as an exemplar for the new repetition.

Furthermore, in our current setting, the possible paths were mostly straight lines. To this end, we will include more marker for constructing curved paths in our future work. Curves can be easily represented using quadratic function or Bessel function. Therefore, we could include more markers (e.g. quadratic marker, Bessel marker) for generating curved structures.

As a systematic step of the future work, we will conduct a series of participatory design sessions with users to co-design new marker-based commands for DupRobo.

6.2 Speed of Autocompletion

Some workshop participants commented on the slow building speed of the robotic autocompletion process. It currently took approximately 25s for the robotic arm to grip and build one block on the workbench. The speed of the robotic arm was fine-tuned to ensure the stability of griping and building. Another reason is that the servo motor we used is not powerful enough to stably support the robotic arm, especially the base joint in which even a small fluctuation caused a large shake on the end effector. Furthermore, the 3D-printed robotic arm limited the range of movements and the capability of building large structures. While the DIY mechanical implementation was used in the current prototype as a proof of concept, we will incorporate high-quality industrial robotic arms in the future.

6.3 Recognition Distortion

In the current prototype, wed use a top-down-facing KinectV2 sensor for both exemplar tracking and marker recognition. Our current algorithm would suffer from the problem of image distortion, especially when the exemplar was set on the edge of the platform. Thus, we limited the area of the workbench for a better recognition result. Therefore, we will tackle this problem by improving the current algorithm with the method of lens distortion rectification [1], to restore the Kinect images.

6.4 Multi-Robot Collaboration

While DupRobo is inspired by Smithwick et al.'s vision of robotic design assistant [27], we further envision that the robotic assistant in tangible creative process could duplicate the designer's initial design, and perform iterations of the initial design, enabling rapid prototyping of different design configuration. In addition, the robotic assistant can experiment and construct complex possibilities that are difficult for manual efforts , and further provide new design suggestions to the designer. As the future work, we will investigate coordinating multiple robots with different shapes and functionalities to construct more complex shapes.

7 Conclusion

In this paper, we present DupRobo, an interactive robotic platform for tangible block-based design and construction. DupRobo supported user-customisable exemplar, repetition control with command markers, and physical autocompletion by the robotic arm. A user study with 12 participants showed that the participants rated DupRobo as easy to learn and use. More importantly, DupRobo significantly reduced participants' perceived physical demand, overall effort, and frustration in the process of block-based repetitive structure design and construction, while being compared to the situation without DupRobo. **While this system is currently limited by the variety of the command marker and the speed of the autocompletion process, it can be considered as the first attempt to explore building repeated assembly blocks structure using physical marker-based command. The current system can be easily expanded with more functionalities.** With DupRobo, we envision that the robotic assistant could provide tangible creative design support by duplicating, iterating, and experimenting complicated structure design with minimum input from the human designers, yet preserving the craftsmanship, the materiality, and the intuitiveness of physical artefact design.

8 Acknowledgments

This research was partially supported by grant from the Centre for Applied Computing and Interactive Media (ACIM) of School of Creative Media, the Strategic Research Grants (Project No. 7005021 & 7005172), the Teaching Development Grant (Project No. 6000623 & 6000639), the Applied Research Fund (Project No. 9667189), City University of Hong Kong, and grants from the Research Grants Council (Project No. CityU 21200216) and the Environment and Conservation Fund (Project No. EECA1956) of the Hong Kong Special Administrative Region, China.

References

1. Burak Benligiray and Cihan Topal. 2015. Lens distortion rectification using triangulation based interpolation. In *International Symposium on Visual Computing*. Springer, 35–44.
2. Leah Buechley, Mike Eisenberg, Jaime Catchen, and Ali Crockett. 2008. The LilyPad Arduino: using computational textiles to investigate engagement, aesthetics, and diversity in computer science education. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 423–432.
3. Judith Bütepage and Danica Kragic. 2017. Human-Robot Collaboration: From Psychology to Social Robotics. *arXiv preprint arXiv:1705.10146* (2017).
4. Alexandru Dancu, Catherine Hedler, Stig Anton Nielsen, Hanna Frank, Zhu Kening, Axel Pelling, Adviye Ayça Ünlüer, Christian Carlsson, Max Witt, and Morten Fjeld. 2015. Emergent interfaces: Constructive assembly of identical units. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 451–460.
5. Christopher Blake Evernden. 2014. *Digital imperfections: analog processes in 21st century cinema*. Ph.D. Dissertation. Lethbridge, Alta.: University of Lethbridge, Dept. of New Media.
6. Gabriela Goldschmidt. 1994. On visual design thinking: the vis kids of architecture. *Design studies* 15, 2 (1994), 158–174.

7. Ankit Gupta, Dieter Fox, Brian Curless, and Michael Cohen. 2012. DuploTrack: a real-time system for authoring and guiding duplo block assembly. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. ACM, 389–402.
8. Sandra G Hart and Lowell E Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Advances in psychology*. Vol. 52. Elsevier, 139–183.
9. Jonathan Hiller and Hod Lipson. 2009a. Design and analysis of digital materials for physical 3D voxel printing. *Rapid Prototyping Journal* 15, 2 (2009), 137–149.
10. Jonathan D Hiller and Hod Lipson. 2009b. Fully recyclable multi-material printing. In *Solid Freeform Fabrication Proceedings*. Citeseer, 98–106.
11. Meng-Ju Hsieh, Rong-Hao Liang, Da-Yuan Huang, Jheng-You Ke, and Bing-Yu Chen. 2018. RFIBricks: Interactive Building Blocks Based on RFID. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 189.
12. Hiroshi Ishii and Brygg Ullmer. 1997. Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*. ACM, 234–241.
13. Rubaiat Habib Kazi, Takeo Igarashi, Shengdong Zhao, and Richard Davis. 2012. Vignette: interactive texture design and manipulation with freeform gestures for pen-and-ink illustration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1727–1736.
14. Jae Woo Kim, Kyung Kyu Kang, and Ji Hyoung Lee. 2014. Survey on automated LEGO assembly construction. (2014).
15. Evelina Kourteva and Dermott Mc Meel. 2017. Entropy: Unpacking the form through post digital making. *The Design Journal* 20, sup1 (2017), S172–S183.
16. Danny Leen, Raf Ramakers, and Kris Luyten. 2017. StrutModeling: A Low-Fidelity Construction Kit to Iteratively Model, Test, and Adapt 3D Objects. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. ACM, 471–479.
17. Daniel M Lofaro, Tony Truong Giang Le, and Paul Oh. 2009. Mechatronics education: from paper design to product prototype Using LEGO NXT Parts. In *FIRA RoboWorld Congress*. Springer, 232–239.
18. Sheng-Jie Luo, Yonghao Yue, Chun-Kai Huang, Yu-Huan Chung, Sei Imai, Tomoyuki Nishita, and Bing-Yu Chen. 2015. Legolization: optimizing LEGO designs. *ACM Transactions on Graphics (TOG)* 34, 6 (2015), 222.
19. Yusuke Maeda, Ojiro Nakano, Takashi Maekawa, and Shoji Maruo. 2016. From CAD models to toy brick sculptures: A 3D block printer. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2167–2172.
20. Andrew Miller, Brandyn White, Emiko Charbonneau, Zach Kanzler, and Joseph J LaViola Jr. 2012. Interactive 3D model acquisition and tracking of building block structures. *IEEE transactions on visualization and computer graphics* 18, 4 (2012), 651–659.
21. Stefanie Mueller, Tobias Mohr, Kerstin Guenther, Johannes Frohnhofen, and Patrick Baudisch. 2014. faBrickation: fast 3D printing of functional objects by integrating construction kit building blocks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 3827–3834.
22. Eric M Orendt, Myriel Fichtner, and Dominik Henrich. 2016. Robot programming by non-experts: Intuitiveness and robustness of one-shot robot programming. In *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 192–199.
23. Mengqi Peng, Jun Xing, and Li-Yi Wei. 2017. Autocomplete 3D Sculpting. *arXiv preprint arXiv:1703.10405* (2017).

24. Bertrand Schneider, Patrick Jermann, Guillaume Zufferey, and Pierre Dillenbourg. 2011. Benefits of a tangible interface for collaborative learning and interaction. *IEEE Transactions on Learning Technologies* 4, 3 (2011), 222–232.
25. Yasaman S Sefidgar, Prerna Agarwal, and Maya Cakmak. 2017. Situated tangible robot programming. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 473–482.
26. Keita Sekijima and Hiroya Tanaka. 2015. Reconfigurable three-dimensional prototype system using digital materials. In *ACM SIGGRAPH 2015 Posters*. ACM, 89.
27. Daniel Smithwick, David Kirsh, and Larry Sass. 2017. Designerly Pick and Place: Coding Physical Model Making to Inform Material-Based Robotic Interaction. In *Design Computing and Cognition'16*. Springer, 419–436.
28. John Strobel. 2010. *All the better to see you with: A comparison of approaches to delivering instructions for Lego construction tasks*. Ph.D. Dissertation. Bowling Green State University.
29. Shu-Chiao Tsai, Hooman Samani, Yu-Wei Kao, Kening Zhu, and Brian Jalaian. 2018. Design and Development of Interactive Intelligent Medical Agent. In *2018 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*. IEEE, 210–215.
30. Sil Van de Leemput and Martijn Van Otterlo. 2013. DUPLOG: Probabilistic logical interpretation of Duplo assemblies from 3D Vision. (2013).
31. Jan Van den Bergh, Bram van Deurzen, Tom Veuskens, Raf Ramakers, and Kris Luyten. 2018. Towards Tool-Support for Robot-Assisted Product Creation in Fab Labs. In *International Conference on Human-Centred Software Engineering*. Springer, 219–230.
32. Xi Vincent Wang, Zsolt Kemény, József Váncza, and Lihui Wang. 2017. Human–robot collaborative assembly in cyber-physical production: Classification framework and implementation. *CIRP annals* 66, 1 (2017), 5–8.
33. Jun Xing, Hsiang-Ting Chen, and Li-Yi Wei. 2014. Autocomplete painting repetitions. *ACM Transactions on Graphics (TOG)* 33, 6 (2014), 172.
34. Jun Xing, Li-Yi Wei, Takaaki Shiratori, and Koji Yatani. 2015. Autocomplete hand-drawn animations. *ACM Transactions on Graphics (TOG)* 34, 6 (2015), 169.
35. Grim Yun, Cheolseong Park, Heekyung Yang, and Kyungha Min. 2017. Legorization with multi-height bricks from silhouette-fitted voxelization. In *Proceedings of the Computer Graphics International Conference*. ACM, 40.
36. Shengdong Zhao, Koichi Nakamura, Kentaro Ishii, and Takeo Igarashi. 2009. Magic cards: a paper tag interface for implicit robot control. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 173–182.
37. Kening Zhu. 2012. A framework for interactive paper-craft system. In *CHI'12 Extended Abstracts on Human Factors in Computing Systems*. ACM, 1411–1416.
38. Kening Zhu, Taizhou Chen, Feng Han, and Yi-Shiun Wu. 2019. HapTwist: creating interactive haptic proxies in virtual reality using low-cost twistable artefacts. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, 693.
39. Kening Zhu, Alexandru Dancu, and Shengdong Shen Zhao. 2016. Fuseprint: A DIY 2.5 D printing technique embracing everyday artifacts. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*. ACM, 146–157.
40. Kening Zhu, Owen Noel Newton Fernando, Adrian David Cheok, Mark Fiala, and Theam Wei Yang. 2010. Origami recognition system using natural feature tracking. In *2010 IEEE International Symposium on Mixed and Augmented Reality*. IEEE, 289–290.
41. Kening Zhu, Xiaojuan Ma, Gary Ka Wai Wong, and John Man Ho Huen. 2016. How different input and output modalities support coding as a problem-solving process for children. In *Proceedings of the The 15th International Conference on Interaction Design and Children*. ACM, 238–245.

42. Kening Zhu and Shengdong Zhao. 2013. AutoGami: a low-cost rapid prototyping toolkit for automated movable paper craft. In *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 661–670.