

Participatory Layout Design-Fabrication with Native Forms of Natural Materials

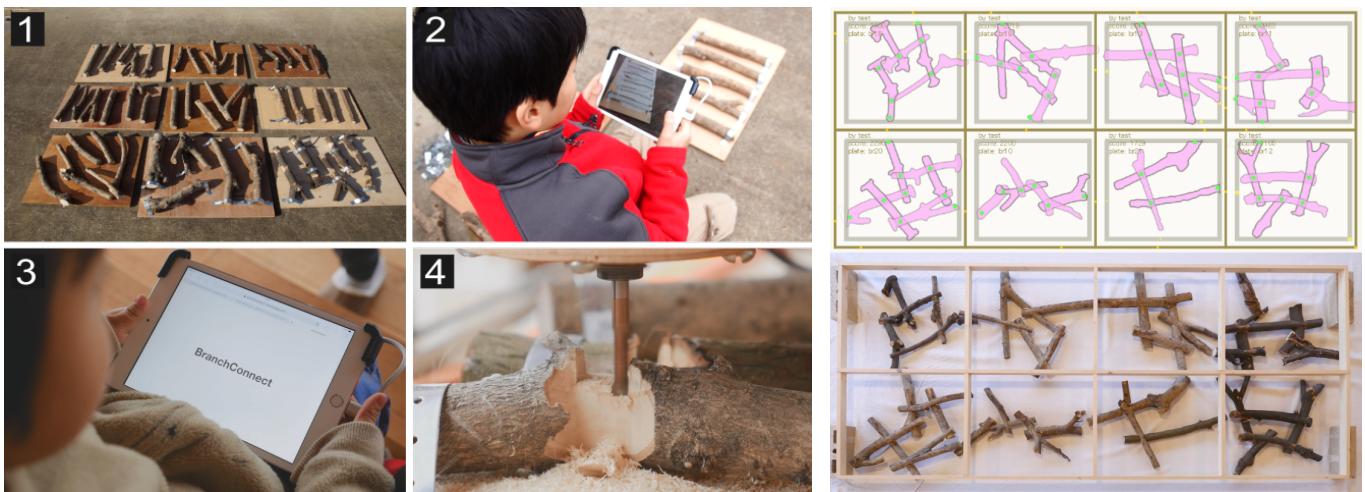


Figure 1: Left: an overview of the workflow: 1. fix branches on plates. 2. scan the plates and upload the model. 3. play the game with scanned branches. 4. fabricate joineries by a CNC router. Right top: branch layouts partially designed by participants in a workshop. Right bottom: the fabricated 2D fence (2m x 0.9m). Each pair of branches is connected with rigid lapped joinery.

Abstract

Diverse natural materials such as stones and woods have been used as architectural elements, preserving their native forms since primitive shelters, however, the use of them in modern buildings is limited due to their irregular properties. In this paper, we take the diversity as playful inputs for design task, and present our game-based design-fabrication platform for customized architectural elements. Taking tree branches as a material with native forms, our online game *BranchConnect* enables users to design 2D networks of branches and fabricate it by a CNC router. The game considers fabrication constraints such as limitations with ordinal 3-axis CNC routers. Each connection has a customized unique joinery adapted to native forms of branches. The scoring system of the game guides users to design structurally sound solutions with given branches. Together with low-cost mobile scanning devices, users with diverse contexts can contribute to design and fabrication process not only as a game user, but also from collecting branches around their physical environments and uploading them to our online platform. For validating our process, we conducted a workshop with end-users (children and their parents). They collected branches in a nearby forest and contributed to design/fabricate a 2D fence with our system.

Keywords: radiosity, global illumination, constant time

Concepts: •Computing methodologies → Image manipulation; Computational photography;

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s). © 2016 Copyright held by the owner/author(s).

SIGGRAPH 2016 Posters, July 24-28, 2016, Anaheim, CA

ISBN: 978-1-4503-ABCD-E/16/07

DOI: <http://doi.acm.org/10.1145/9999997.9999999>

1 Introduction

Modern buildings are characterized by its uniformity; built upon the same principle of construction system, consisting of standardized building component and its assembly process. The standardized construction system is favored because of its efficiency in design and production. Each component satisfies specified structural performance, thus the assembled resulting structure can be analyzed systematically. On the other hand, excessive standardization converged buildings into similar materials and details, resulting in the detached design from built environment. Reacting on the issue, designers and architects actively use local materials not only as inspirational sources, but also as a catalyst of their design to the local context [Oliver 1997].

Since primitive shelters and huts, traditional construction has used locally found natural materials directly as they are found [Weston 2003]. Such a direct use can not compete with highly standardized materials and construction system, however, the uniqueness of native forms is a valuable quality which is lacking in standardized materials. As the material is locally obtained, building and living get much closer, thus people using the building can easily commit design and fabrication, fostering the sense of belonging to the community []. **TODO: citation** Locally obtained materials can easily connect design and the context of built environment in this way, but their irregular properties limit the use of them in modern buildings. Traditionally, craftsman has taken care irregular natural materials varying their native forms and dynamically design the global design by considering individual material properties [Pye 1968]. Such a task is difficult to be automated and these skills are developed through years of training, thus the use of native forms typically costs more than standardized construction system.

This paper aims to make the above-mentioned qualities of materials with native forms more accessible by digital technologies. We use locally obtained branches which can be found almost everywhere not only in country-side but also in urban environment such as parks and along streets. Public service takes care of these branches: an-

nual pruning, storing, and chipping or burning with some costs. The size of branches (from 5 -30 cm in diameter) is too small for furniture or other structural applications. It is a challenge for digital design and fabrication to utilize the diverse branches in meaningful ways. High-precision but low-cost scanning devices and personal digital fabrication machines make it possible to analyze and control natural materials with diverse native forms.

The key technical difficulty of materials with natives forms is design. Optimization is a straightforward approach: minimize an energy function which integrates structural and fabrication costs. This approach, however, is limited to particular design scenario with specific materials. Furthermore, the concept of optimum solution is well suited to goals such as efficiency or low-cost, but these goals are not the qualities materials with natives forms can compete with standardized construction systems. Instead, we take humans in our scan-design-fabricate workflow not only to solve the design problem, but also to provide an opportunity for people to participate in the workflow. Traditionally, in case of constructing public and symbolic buildings, such as church, people in a community took initiatives from fund raising to design, or even in construction process [].

TODo: citation

In this paper, we report our case study to design and fabricate an architectural element out of irregularly shaped branches, using our humans-in-the-loop system. We developed an online platform where users can post branches found at hand, and design with them through the game *BranchConnect*. The game system itself helps users to design valid branch layouts and enable them to fabricate customized joineries to connect them together without screws and adhesives. The design of our joinery extends the traditional orthogonal lap-joints to various angles within a range, freeing the diverse forms of woods from orthogonal connections. Physically collected branches are digitally scanned and stored in a cloud database *BranchCollect*, and offline application *Branch Importer* analyzes forms of branches and upload to the database. The simple visual feedback and scoring system of the game guide users to valid solutions, which are further inspected by an offline application *G-Code Generator* for CNC milling process. The game system and developed import/export applications are currently limited to branches, however, the principle of human-in-the-loop for design is applicable for other kinds of materials with diverse irregular forms. We hope our method sheds lights on materials such as waste from demolition of buildings for various design applications.

In summary, our contributions are

- a workflow enabling to take natural materials with native forms as design components.
- an online game-based approach to participatory design-fabrication.
- an algorithm to generate customized non-orthogonal joineries which are fabricatable by a CNC router.

2 Related Work

3D printers and CNC routers made digital fabrication more accessible, and pre-fabricated customized building components are often used in buildings nowadays [Knaack et al. 2012]. According to the theory by Pye [1968], these components are processed from highly standardized material, thus its digital fabrication process is “workmanship with certainty”; a batch process of reading G-Code and strict execution of the code. On the other hand, as “workmanship with risks” in digital fabrication, interactive fabrication enables machines to pick up uncertain happenings and react on it [Willis et al. 2011]. Mueller and her colleagues developed interactive laser cut-

ting, taking user inputs and recognizing placed objects in a fabrication scene [Mueller et al. 2012]. While their system interprets objects as simple platonic geometry, our work takes the native forms of branches, however, our work does not support interactivity in fabrication process. Crowdsourced Fabrication project took advantage of humans-in-the-loop in their fabrication system [Lafreniere et al. 2016]. On the other hand, our work puts emphasis on crowdsourced design as a socially networked fabrication. As a crowdsourced design system, Jerry et. al., developed a platform for light users to design trees and plants [Talton et al. 2009]. Our work also developed online collaborative design platform but directly linked to the real-world.

There are few works that take natural materials with native forms as design components. Schindler and his colleagues used digitally scanned wood branches and used them for furniture and interior design elements [Schindler et al. 2014]. Monier and colleagues virtually generated irregularly shaped branch-like components and explored designs of large scale structure [Monier et al. 2013]. Using larger shaped forked tree trunks, *Wood Barn* project designed and fabricated custom joineries to construct a truss-like structure[Mairs 2016]. *Smart Scrap* project digitally measured lime stone leftover slates from a quarry and digitally generated assembly pattern of slates [Greenberg et al. 2010]. In industry, recognition of irregularly shaped objects is essential for waste management. *ZenRobotics* developed a system that sorts construction and demolition waste by picking objects on a conveyor belt using robotic hands [Lukka et al. 2014]. For factory automation purpose, there is a system that recognizes irregularly shaped objects and sort them into a container [Sujan et al. 2000]. While these projects demonstrated the capability of digital fabrication processes to handle irregularly shaped materials, design process with native natural materials is still dependent on experts.

Cimerman discussed architectural design practices that took computer-mediated participatory (architectural) design [Cimerman 2000]. He mentioned three motivations of digital participatory design:

- Including stakeholders in creation of one's environment.
- Experimenting diverse design tastes from multiple point of views.
- Solving complex design tasks with full of diverse solutions.

Opening database of available local materials, people with various backgrounds can involve in design process, which could lower the cost of design fee with natural materials with native forms. For example, *Nano-Doc* took gamification approach to search valid nanoparticle designs against tumors out of infinite design space [Hauert et al.]. *DrawAFriend* has developed an online game to collect big-data for drawing applications which assist humans with auto-stroke assistance [Limpaecher et al. 2013]. While these works developed games for collecting valuable data for solving medical or engineering problems, our game is served as a collaborative design platform, aiming to solve the socio-cultural issues in modern buildings: 1.generic design and 2.detached context.

3 Workflow

As illustrated in the left of Figure 1, our workflow starts from physically collecting branches. The collected branches are uploaded to cloud database by *Branch Importer*, and served to the online game-based design application *BranchConnect*. The game system uses skeletons for its joint detection process working on browsers and accessible from laptop computers and mobile devices. As in the right of 1, users can explore a global design with multiple branch

layouts. Once the global design is fixed, designed layouts are further inspected by *G-Code Generator*, which generates customized joineries for CNC milling. After finishing the milling process, users physically assemble branches and complete the fabrication process. The pipeline of the workflow is illustrated in the Figure 2. In this section, we introduce two steps in the pipeline: Digital Model Acquisition and Fabrication. As for the game system, please refer Section 4.

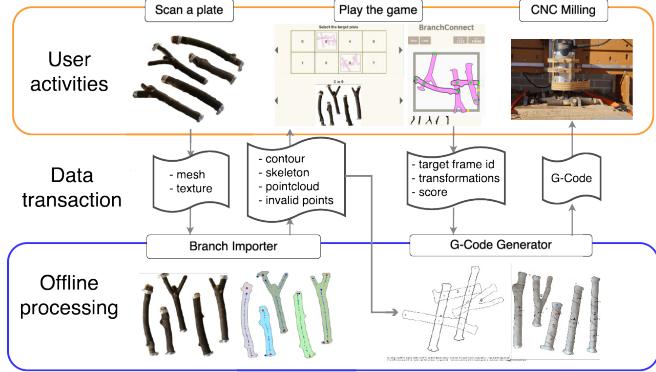


Figure 2: A pipeline from model acquisition to fabrication.

3.1 Digital Model Acquisition

Our system takes textured mesh model or point cloud with colored vertices. As complete mesh model provides more robust results with 3D shapes of branches, we describe our process based on mesh model as an input. There are various methods and software available for scanning 3D models. As for scanning setup, we describe in the Section 5. Taking mesh model with colored texture, our *Branch Importer* provides functions such as object detection, skeleton extraction, branch type classification, and fixture point setting.

The scanned result is a mesh model representing branches with a fixed plate. The system first identifies branches by applying simple height threshold, and then applies contour detection (we currently use *findContour* in OpenCV¹). The obtained 2D contours are used for extracting skeletons and clustering point cloud in the mesh model. Contours are triangulated and skeleton points are extracted from middle points on edges of triangles. These middle points are compared with top view image from OpenCV. If the point is inside of a contour, the middle point is counted as a valid point. After extracting valid middle points, the connectivity of skeletons is analyzed. In case grafting branch is detected, a new skeleton sub-branch is added. The result is shown in Figure 3. Metal fixture locations are confirmed by simple mouse-clicks and set as invalid points. The acquired information is stored in a cloud database.

3.2 Fabrication

After a design is selected for fabrication, the validity of the design is further inspected by a high-resolution model. The *G-Code Generator* displays joineries and milling paths on scanned orientations. In case joineries are invalid with the high-resolution model, a layout can be easily modified with simple mouse inputs (see Figure 4). Users can also change milling parameters such as offset ratio for the side cuts, milling bit diameter, depth of joineries, cutting speed, moving height and so forth. After confirming the fabrication settings and milling paths, it generates G-Code.

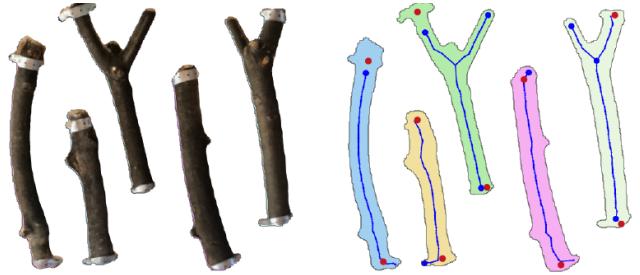


Figure 3: An interface of *Branch Importer*. Left: a top ortho-view image of textured mesh model. Right: Extracted skeletons are shown with blue dots. The beginning of skeletons is shown bigger dots, and the red dots are invalid points defined by a user.

Some fabrication factors such as invalid points due to metal fixtures and flipped (further described in Section 4.2) are already considered by *Branch Importer* and the game system respectively. In this section, we describe the process of joinery generation. Each joinery's geometry is parametrically modeled with two planar surfaces on the sides of branches (side cuts) and one planar top surface (center cuts) (see Figure 5.2). Side cuts have wedged corners for smooth assembly process. The geometry creates rigid joints with irregularly shaped sections of branches.

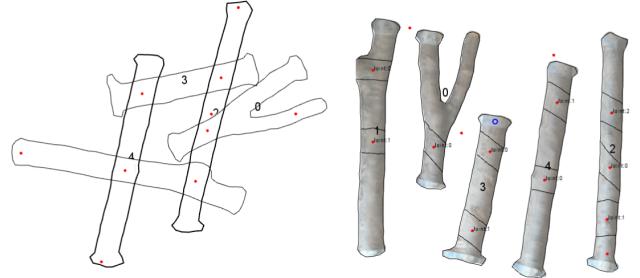


Figure 4: Interface of *G-Code Generator*. Left: a layout defined by a user. The calculated joineries are displayed. Right: an original orientation of a scanned plate with joineries.

Similar to the joint searching process with skeletons, the *G-Code Generator* searches a set of four closest points from high-resolution contours, expecting that every intersected contour has four curves. After finding the four closest points, it trims two curves from each contour of branch. (two from intersecting branch and two from intersected branch) at each joint. The trimmed contours are transformed to the original scanned orientation and used for generating milling paths. Two curves from an intersected branch are used for generating side cuts milling paths, which are inwardly offset paths of the original branch contours. Height of center cuts is half of branch diameter. The diameter as the joint is calculated with the contour. We compare this value with actual height from the stored point cloud. In case of under-cuts, the system detects different values and adjust the center cuts according to the recalculated diameter.

¹Open Source Computer Vision Library: <http://opencv.org/>

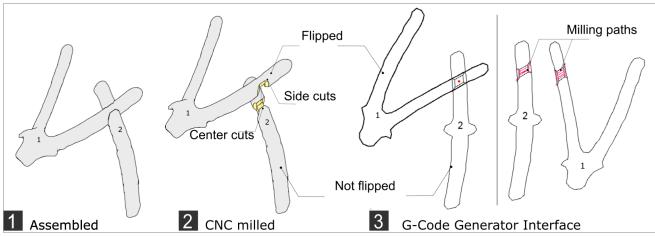


Figure 5: An example of intersected pair: 1. an assembled pair of branches. 2. branches after the center and side cuts are milled 3. left: a layout defined by a user right: the original orientations of branches with generated milling paths with red color.

4.1 System Overview

There are many collision detection libraries available, however, our game needs to detect intersected branch pairs, thus surface contact detection is overkill for our application. Also most branches come with free-form concave shapes, thus further geometric preparation such as convex decomposition is necessary for using these libraries. For fast and robust intersection detection, our system extensively uses skeletons of branches.

Hubbard and Philip developed collision detection by representing object with hierarchical 3D spheres aligned on skeletons [Hubbard 1996]. Our system takes similar approach but limited in 2D, but more focused on searching fabricatable joints. In the game, simplified skeletons are used to find the pair of closest skeleton points between two branches. When a branch is selected, it is counted as an active, and the system searches the closest skeleton point from skeletons of other branches. More precise joint calculation with higher resolution is further described in the next section.

A joint is created when an intersecting pair is detected, and the pair forms a group. The group is used for evaluating connections between target points (*Bridged*). The game is completed once all the target points are connected by a group of branches. The conditions of joint and group are indicated with simple color-code. Once the user finishes positioning, score is updated with weighted sum of parameters. Together with the color-code, the score update guides the user to form a valid design.

4.2 Joint Condition

Joint is the essential entity not only in the game but also in the fabrication process of customized lapped joineries.

Importantly, each pair of branches must have one flipped branch for fabrication constraint. We describe this further in a later section (see Section 3.2).

Figure 7 illustrates valid and invalid joint conditions.

Our joint only takes crossed pair (see Figure 7.1) because they are structurally stable, relatively simple to fabricate, and creates diverse designs. Tangential connections are counted as invalid as fabrication of tangential joinery is challenging with small branches (see Figure 7.3).

The online game is accessible by laptops and mobile touch devices, and many users can play at the same time. The objective of the game is to collect valid layouts of branches which are fabricatable with 3 axis CNC milling machines. By analyzing the connectivity of branches and target points, the game checks structural feasibility of a given layout. Similar to our game, the guidance system during furniture design inspected connectivity, durability, and stability [Umetani et al. 2012]. Unlike their work, our game puts emphasis on *fabricability*, as well as geometric *connectivity*, and does not calculate structural performance of each joint. Instead, we restrict valid layout space by limiting valid joint and conditions. We also assume that every fabricated joint works as a rigid joint, thus single connection is stable to hold a pair of branches.

Firstly a user selects a frame indicating multiple target points to be connected, and then selects a set of branches fixed on a plate (The left in Figure 6). After selecting the target frame and the set of branches, the user is guided to the game interface, consisting of the frame with the target points, and the set of available branches (The right in Figure 6). The user picks a branch from the available set on the bottom, and then places it in an arbitrary 2D pose through basic manipulations such as move, rotate, and flip. The number of available branches differs depending on plates. With the feasible diameter of branches (over 2cm) and the plate size (50cm x 50cm), the number of available branches is most likely up to six. Within the limited number of branches, the user bridges all the target points by connecting all the used branches in one group. The game is completed when all the target points are connected. For higher score, the user can modify the design after the completion, and save it to the database.

Joint is the essential entity not only in the game but also in the fabrication process of customized lapped joineries.

Importantly, each pair of branches must have one flipped branch for fabrication constraint. We describe this further in a later section (see Section 3.2).

Figure 7 illustrates valid and invalid joint conditions.

Our joint only takes crossed pair (see Figure 7.1) because they are structurally stable, relatively simple to fabricate, and creates diverse designs. Tangential connections are counted as invalid as fabrication of tangential joinery is challenging with small branches (see Figure 7.3).

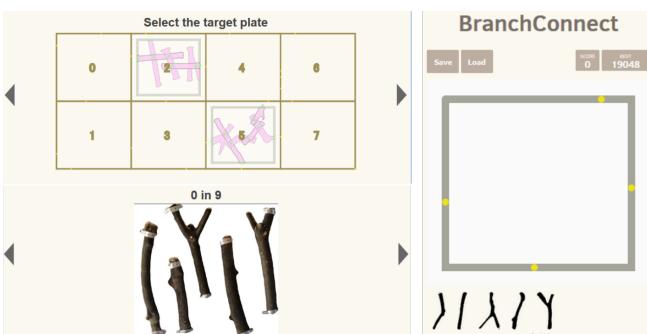


Figure 6: Left: the selection interface for target frames (top) and branch panels (bottom). Right: the start interface of the game.

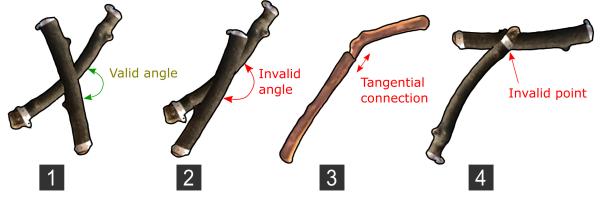


Figure 7: Joint conditions. 1.valid joint. 2.invalid for violating the angle. 3.invalid tangential connection. 4.invalid for connecting on a fixture point.

A valid joint's angle stays within a fixed range (see Figure 7.1 and 2). Joints close to metal fixtures are also counted as invalid (see Figure 7.4). Valid and invalid joints are displayed with green and red respectively.

To describe the process, we let \mathcal{B} denotes a set of all the available branches, and each branch as $b_i \in \mathcal{B}$. While the process checks joint condition through all the branches \mathcal{B} , each detected joint is

322 stored in each branch b_i , categorized in different conditions such as
 323 valid and invalid joints denoted as $j_{\text{valid},j,i} \in \mathcal{J}_{\text{valid},i}$ and $j_{\text{invalid},j,i} \in$
 324 $\mathcal{J}_{\text{invalid},i}$ respectively. When a branch b_i is connected to one of target
 325 point $t_j \in \mathcal{T}$, the t_j is stored in b_i .

326 A flowchart of the game system with joint and group conditions is
 327 illustrated in Figure 8.

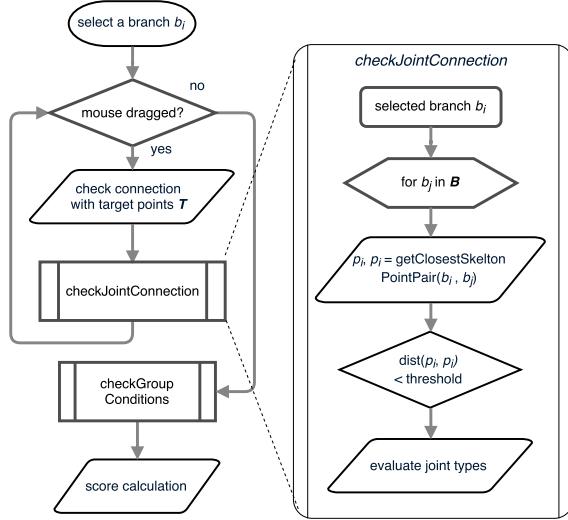


Figure 8: Left: an overview of the game system with 1. joint check and 2. group check and 3. score calculation. This process is iteratively executed while a user is exploring layout by dragging a branch. The joint check process is further illustrated in the right, and group condition check is described in Algorithm 1.

4.3 Group Condition

328 After checking joint conditions of all the pairs of branches, the sys-
 329 tem checks the number of groups as well as its connection with the
 330 target points on a frame. If a group is not connected to any target
 331 point nor other groups, the group is *Islanded* and structurally in-
 332 valid. While a user is positioning a branch by dragging or rotating,
 333 groups are continuously calculated and indicated by simple color
 334 (Figure 9).
 335

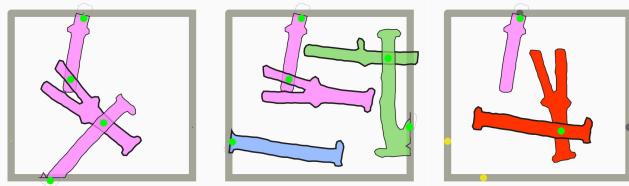


Figure 9: Left: valid group with two target points connected. Middle: valid but three groups. Right: invalid due to the Islanded situation.

336 After all the joint conditions are checked, we evaluate group con-
 337 ditions. Through checking the all the branches B , the first group
 338 g_0 is created and stored as b_0 . When a branch is connected to a
 339 target point, the graphics of the point changes, and the branch and
 340 its belonging group's color also changes. When a group bridges
 341 a pair of target points, a special score is added and displayed in
 342 a pop-up square, also the graphics of target point's changes. The
 343 branch connected to the target point is trimmed at the target point,
 344 and the trimmed length is subtracted from the score. The game is

345 completed when the number of \mathcal{G} is one, and all the target points
 346 are connected with the group. The algorithm which checks group
 347 conditions is described in 1.

Algorithm 1 Group Condition Update Algorithm

```

1: function UPDATEGROUPS( $\mathcal{B}$ )
2:   Reset all the groups  $\mathcal{G}$ 
3:   Create new group  $g_0$ 
4:    $b_0$  is added to  $g_0$ 
5:   if  $b_0$  has connected target point  $t_i \in \mathcal{T}$  then
6:      $g_0$  sets  $t_i$ 
7:    $g_0$  is added to  $\mathcal{G}$ 
8:   for each branch  $b_i$  in  $\mathcal{B}$  do
9:      $GroupConnection \leftarrow false$ 
10:    for each group  $g_j$  in  $\mathcal{G}$  do
11:      for each branch  $b_j$  in  $g_j$  do
12:        if  $b_{\text{paired},i} \in \mathcal{P}_j$  has  $b_j$  then
13:           $b_i$  is added to  $g_j$ 
14:           $GroupConnection \leftarrow true$ 
15:          if ( $b_i$  has  $t_i$ ) and ( $g_j$  has  $t_j$ ) then
16:            Set  $g_j$  as Bridged
17:          if  $g_j$  has no  $t_j$  then
18:            Set  $g_j$  as Islanded
19:            break
20:        if  $GroupConnection$  is false then
21:          create new group  $g_{new}$ 
22:           $b_i$  is added to  $g_{new}$ 
23:           $g_{new}$  is added to  $\mathcal{G}$ 
  
```

4.4 Score Calculation

348 In the score calculation processes, following entities forms the
 349 score: the numbers of valid and invalid joints on each branch,
 350 the number of groups as $N(\mathcal{G})$, the number of islanded groups
 351 as $N(g_{\text{islanded}} \in \mathcal{G})$, the number of bridged target points as
 352 $N(t_{\text{bridged},i}) \in \mathcal{T}$). The trimmed lengths of branches which are
 353 connected with target points are denoted as $\text{trimmed}(t_j, b_i)$. The
 354 score is weighted sum of these joint and group conditions, denoted
 355 in Equation (1). The weights 'w1...w5' are non-negative weight
 356 coefficients pre-adjusted in advance by authors.
 357

$$\begin{aligned}
 Score = & w_1 \sum_1^{N(\mathcal{B})} \sum_1^{N(\mathcal{J}_{\text{valid},i})} j_{\text{valid},j,i} + w_2 \sum_1^{N(\mathcal{B})} \sum_1^{N(\mathcal{J}_{\text{invalid},i})} j_{\text{invalid},j,i} \\
 & + w_3 \sum_1^{N(\mathcal{G})} g_{\text{islanded}} + w_4 \sum_1^{N(\mathcal{T})} t_{\text{bridged}} \\
 & + w_5 \sum_1^{N(\mathcal{T})} \text{trimmed}(t_j, b_i) \\
 \text{s.t. } w_j & \geq 0 \forall j \in 1, \dots, 5
 \end{aligned} \tag{1}$$

5 Case Study

359 A design and fabrication workshop was organized to examine the
 360 feasibility of our system with a specific design target and a location.
 361 We selected a public community house where people in the com-
 362 munity share the space and regularly use the facility. Participants
 363 of the workshop were selected among them, who were four chil-
 364 dren aged 10 and under (4, 7, 9, and 10 years old) and two parents

(Figure 10). We specifically selected children with this range of age as non-experts without experiences in computational design or digital fabrication, also for observing the clarity and attractiveness of the game. The entire workshop was filmed and documented in a short clip. Please refer the supplementary video material for the documentation.

The goal for the participants was to contribute to an ongoing design and fabrication process of screen wall (2m by 0.9m) consisting of 8 rectangles. Six frames were already designed and built, thus the rest two frames were prioritized for them to design and fabricate in this workshop.

Participants were informed about the goal of the workshop, and each process was introduced by experienced tutors. The processes were from collecting and fixing the branches on a plate, scanning the plate, complete designs by playing the game, and assembly after CNC milling.



Figure 10: An overview of the workshop. 1.the overview of the space. 2.collect branches. 3.cut in certain lengths 4.attach on a plate 5.scan the plate 6.play the game 7.CNC milling.

5.1 Setup

System and Hardware Setup

We used two iPad-minis with iSense depth cameras attached for scanning branches, and a 3 axis CNC milling router with a 6 mm diameter milling bit. We used a laptop PC (Lenovo w240 with intel core i7) for running *Branch Importer* and *G-Code generator*, as well as operating the milling machine. The scan area of iSense camera is 500mm x 500mm, and the milling machine's stroke length along z-axis is 70mm, which provide geometric constraints for available branch sizes. *BranchConnect* was hosted at *Heroku* cloud server², and we used *MongoDB*³ as a cloud database.

Preparation of Branches

The participants were asked to collect branches with 2 - 10 cm in diameter. The lower bound of the diameter was set due to the milling bit size, and the upper bound was set for the limited length of z-stroke of the CNC router. The collected branches were cut in arbitrary lengths, not longer than 500 mm due to the limit of scanning area.

As our game system and fabrication process take 3D branch shapes as 2D contours (with limited use of point cloud), these constraints work positive for the system which automatically filter out branches with large 3D twists. We asked participants to scan with iPad + iSense and prepare feasible mesh model by themselves. After obtaining mesh models, tutors imported models from iPads to a laptop and upload them to database by *Branch Importer*.

²Heroku is a platform as a service (PaaS) that enables developers to build, run, and operate applications entirely in the cloud. <https://www.heroku.com/>

³MongoDB is a free and open-source cross-platform document-oriented database program. <https://www.mongodb.com/>

User Experiences with the System

After models are uploaded to the server, users can see that their plates are added in the selectable branch plates with their names and locations. Users can access to the start page by PC and mobile devices. We prepare both options and let participants choose a device.

A user is firstly directed to a start page and asked to submit a user name. Secondly, the user is navigated to target frame selection page, and asked to pick one out of eight frames. Each frame has different target points. The interface also shows the completed branch organizations within each target frame. If there are multiple designs, three designs with highest scores are displayed. The user can change the currently displayed design by clicking within each frame and choose either starting their design from scratch, or select the design and improve it.

After selecting a target frame, the user goes to branch selection page, displaying 15 plates when the workshop was held. In this page, they see the plates made by themselves on the page, as well as their names on the plate. The user can select the same plate for designing other target frames. By clicking a displayed branch plate, the user is navigated to the game interface. After completing to bridge all the target points, the design is automatically uploaded to the database, but the user can continue to design. Tutors and participants select design solution for each target frame, and an experienced tutor operates *G-Code Generator* as well as the CNC router. Participants were asked to assembly branches after joineries were milled.

5.2 Results

The entire workshop took 4.6 hours to complete the whole process, including introduction, moving, and pauses. Table 5.2 shows durations of each task.

Task	Duration (hour)	Fraction (%)
Introduction	0.3	6.5
Collecting branches	0.6	13.0
Preparing plates	0.8	17.4
Preparing models	0.3	6.5
Uploading models	0.2	4.3
Designing by the game	0.5	10.8
Inspecting models	0.2	4.3
CNC milling	0.5	10.8
Assembling	0.2	4.3
Moving, pauses	1.0	21.7
In total	4.6	100

Collecting Branches

The diameter and length constraints for available branches worked as guidelines for participants rather than restricting finding and cutting arbitrary branches. After cutting branches in certain lengths, participants fixed branches on plates by thin metal plates with screw holes. It was straightforward for them to firmly fix branches so that they are not moved during milling process. These fixture points are counted as invalid points in the game where joinery points can not be generated. The participants built two plates with three and five branches fixed on each plate.

Model Acquisition

iSense camera comes with an intuitive software for scanning and modifying models. After we gave an instruction, most of participants practiced several scans and successfully scanned models

453 without any problem.

454 Each scanning and re-touching took 2-3 minutes, and 30 seconds
 455 for generating data by *Branch Importer*. Including the prepared
 456 panels previously, we scanned 15 plates in total, 75 branches, and
 457 35.3m of total length including sub-branches. **TODO: check the
 458 length again!** We got 40 *I* shaped branches, 19 *V* shaped branches,
 459 and 16 *Y* shaped branches. The result is shown in Figure 11.

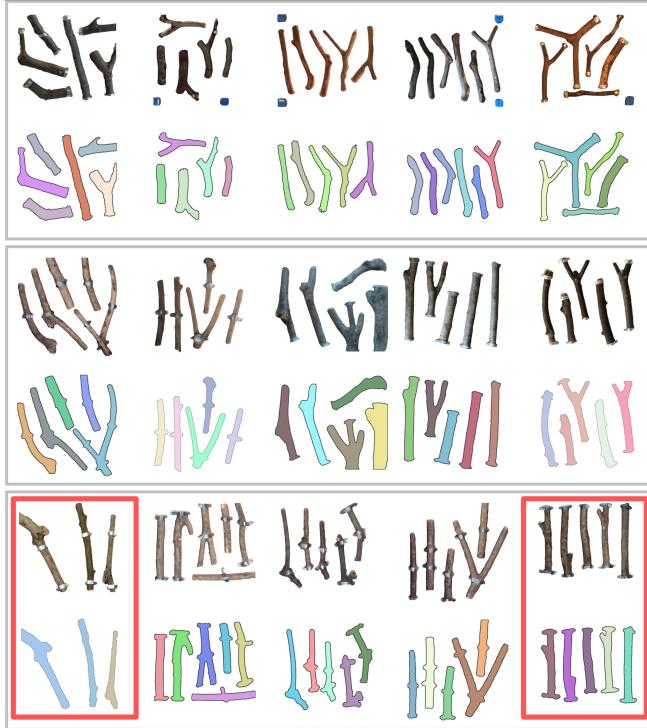


Figure 11: An overview of all the 15 scanned plates for the workshop. Top raw of each set shows ortho-top views of scanned mesh models, and the bottom raw is the recognized branches assigned random colors. The red-lined rectangles indicate the plates built by participants in the workshop.

460 Design with the Game

461 Most participants used iPads for navigating pages and playing the
 462 game. All the participants understood the goal immediately, how-
 463 ever, they had difficulties with mobile touch interface, such as rota-
 464 tion and flipping operation by gestures.

465 One participant switched to play by a PC for more precise control
 466 due to the problem. All the participants chose to develop their own
 467 designs from the scratch, although they had instruction about the
 468 "continue existing designs".

469 Although there is no time limit in the game system, we set 30 min-
 470 utes for playing the game, and 8 solutions were given by partici-
 471 pants. xxx frames were completed per participants and xxx partici-
 472 pants completed the whole eight target frames. The average score is
 473 xxx, and average playing duration was xxx to complete each target
 474 frame.

475 Overall Design Consensus **TODO: this section might be re-** **476 moved**

477 As the target frame selection page can display designs not only
 478 from one user but also from all the others at once, we could get

479 an overview of design options. The designs are displayed as score
 480 descending order but limited numbers, we could find feasible solu-
 481 tions with mostly all the target points were bridged. As participants
 482 were excited by seeing their branches and designs, we took some
 483 of their solutions and their plates for the fabrication although their
 484 solutions did not satisfied completion of the game.

485 Fabrication

486 We did not have major problems for converting designs to G-Code
 487 milling paths as we encountered major issues regarding the fabri-
 488 cation before the workshop, which are reported in this section.

489 The main problem was the accuracy of acquired contours. We
 490 observed most of scanned models had occluded regions between
 491 plates and branches, which create interpolated faces during
 492 solidifying process, resulting in outwardly offsetted contours.
TODO: check offsetted correct english After milling was finished
 493 and when branches were assembled, six pairs of branches were
 494 loosely connected because the calculated contours were 2-3mm
 495 eroded than the actual sizes. We avoided this problem by trimming
 496 branches from 2-5 mm higher than the plate surface. After this
 497 operation, the rest of connections were tightly connected.
 498

500 We also observed that many milling paths were 5-10 mm off from
 501 the center of planned joints. Multiple reasons could be considered
 502 as reasons such as,

- 503 • deformation of mechanical parts of the CNC router
- 504 • not dense resolution of acquired contours of branches
- 505 • misaligned orientation of the plate compared to the scanned
 506 model

507 To avoid the misalignments, we modified the *G-Code Generator*
 508 so that an operator can freely adjust the absolute origin of the gen-
 509 erated milling paths. The origin was usually set with around the
 510 center of the plate. After this modification, the misalignment from
 511 joint center was reduced with 5 mm off at the maximum misalign-
 512 ment. Branches could absorb 3-5 mm misaligned joint positions
 513 due to the elasticity of the wooden materials. The misaligned joint
 514 positions worked as post-tensions, solidifying the structure. We
 515 assume that this is only applicable when an applied bending moment
 516 and cut surface at a joint is orthogonal or not too much off from
 517 orthogonal. **TODO: this sentence**

518 5.3 Summary

519 The objective of the workshop was to observe the validity of our
 520 system for non-expert users. We could see that participants aged
 521 10 and under were capable of following the workflow. Participants
 522 were encouraged to contribute to the ongoing design with the online
 523 platform, and successfully provided designs from available branch
 524 plates.

525 We had several requests from participants regarding the game in-
 526 terface but also related to the workshop organization. Several par-
 527 ticipants requested to allow multiple branch plates for designing a
 528 frame, or even remove the target frame and let them freely design
 529 with branches. Also a participant who gave up with iPad operations
 530 requested additional buttons for mobile touch interface to keep an
 531 active branch selected.

532 A noteworthy fact was that one participant with four years old failed
 533 to complete the game, but insisted on accepting his design to be fab-
 534 ricated by the CNC router. We took these requests as their commit-

535 ment to the entire workflow, as we observed that all the participants
 536 insisted on continuing the fabrication process.

537 6 Conclusion

538 Summary

539 In this paper, we presented a workflow to design and fabricate with
 540 branches which are not large enough for producing standardized
 541 building components. Our workflow was validated with case-study
 542 with lower-aged participants without design and fabrication experi-
 543 ences. Our online platform to store geometries of scanned branches
 544 and compositions of branches was reachable by people with vari-
 545 ous backgrounds. Our light-weight branch joint detection algorithm
 546 was also validated by running on the browser game, contributed to
 547 the accessibility of the presented workflow. Together with the ac-
 548 cessibility, the intuitive interface was simple enough for non-expert
 549 users. We successfully built a network of branches with rigid joints
 550 generated by our joinery milling path generator. Each of joints has
 551 customized lapped-joint geometry, which is challenging to design
 552 and fabricate.

553 Limitations and Future Work

554 Our system is limited to handle 3D shaped branches as 2D with
 555 height information only used for segmentation and G-Code genera-
 556 tion. Our workflow requires branches to be fixed on a plate, which
 557 takes the longest duration in the workflow except for moving and
 558 pause in between tasks. Despite of successfully fabricated non-
 559 orthogonal joineries, we did not complete the attaching branches to
 560 target frames, as we prioritized to validate branch-branch joineries.

561 Our design workflow fully relies on human's ability for obtaining
 562 design solutions. As the problem of limited number of branches
 563 and fixed target points to be connected, the system could assist hu-
 564 mans to reach to structurally sounding solutions with less efforts.
 565 Collected data from game could be further analyzed for extracting
 566 meaningful data for the development of assisting algorithm. Our
 567 joint and group detection, and milling path algorithms are not ap-
 568 plicable to other purposes at this moment. It is valuable to com-
 569 pare the performance of our joint and group detection system to
 570 other collision detection library. In our workflow, the pipeline is
 571 not seamlessly connected: experienced operators need to take care
 572 of retouching mesh model, Branch Importer, and G-Code generator,
 573 and operating a CNC router.

574 Acknowledgements

575 We appreciate the public house for hosting the CNC router and pro-
 576 viding participants for the user study. We thank to the film editor,
 577 Shin Yamane for shooting our video clips. We also thank to the
 578 developer of an online game 2048, Gabriele Cirulli and his contrib-
 579 utors, for sharing source code in Github community.

580 References

- 581 CIMERMAN, B. 2000. Participatory design in architecture: can
 582 computers help? In *PDC*, 40–48.
- 583 GREENBERG, B., HITTNER, G., AND PERRY, K., 2010. Smart
 584 scrap. Accessed: 2016-09-30.
- 585 HAUERT, S., LO, J. H., NACHUM, O., WARREN, A. D., AND
 586 BHATIA, S. N. Crowdsourcing swarm control of nanobots for
 587 cancer applications.

588 HUBBARD, P. M. 1996. Approximating polyhedra with spheres
 589 for time-critical collision detection. *ACM Trans. Graph.* 15, 3
 590 (July), 179–210.

591 KHOSHNEVIS, B. 2004. Automated construction by contour craft-
 592 inglelated robotics and information technologies. *Automation in
 593 construction* 13, 1, 5–19.

594 KNAACK, U., CHUNG-KLATTE, S., AND HASSELBACH, R. 2012.
 595 *Prefabricated systems: Principles of construction*. Walter de
 596 Gruyter.

597 LAFRENIERE, B., GROSSMAN, T., ANDERSON, F., MATEJKA,
 598 J., KERRICK, H., NAGY, D., VASEY, L., ATHERTON, E.,
 599 BEIRNE, N., COELHO, M. H., ET AL. 2016. Crowd sourced
 600 fabrication. In *Proceedings of the 29th Annual Symposium on
 601 User Interface Software and Technology*, ACM, 15–28.

602 LIMPAECHER, A., FELTMAN, N., TREUILLE, A., AND COHEN,
 603 M. 2013. Real-time drawing assistance through crowdsourcing.
 604 *ACM Transactions on Graphics (TOG)* 32, 4, 54.

605 LUKKA, T. J., TOSSAVAINEN, T., KUJALA, J. V., AND RAIKO, T.
 606 2014. Zenrobotics recycler—robotic sorting using machine learn-
 607 ing. In *Proceedings of the International Conference on Sensor-
 608 Based Sorting (SBS)*.

609 MAIRS, J., 2016. AA design and make students use a robotic arm
 610 to build a woodland barn. Accessed: 2016-09-30.

611 MONIER, V., BIGNON, J. C., AND DUCHANOIS, G. 2013. Use of
 612 irregular wood components to design non-standard structures. In
 613 *Advanced Materials Research*, vol. 671, Trans Tech Publ, 2337–
 614 2343.

615 MUELLER, S., LOPES, P., AND BAUDISCH, P. 2012. Interac-
 616 tive construction: interactive fabrication of functional mechani-
 617 cal devices. In *Proceedings of the 25th annual ACM symposium
 618 on User interface software and technology*, ACM, New York,
 619 NY, USA, UIST '12, 599–606.

620 OLIVER, P. 1997. *Encyclopedia of vernacular architecture of the
 621 world*. Cambridge University Press.

622 PYE, D. 1968. *The nature and art of workmanship*. Cambridge
 623 UP.

624 SCHINDLER, C., TAMKE, M., TABATABAI, A., BEREUTER, M.,
 625 AND YOSHIDA, H. 2014. Processing branches: Reactivating
 626 the performativity of natural wooden form with contemporary
 627 information technology. *International Journal of Architectural
 628 Computing* 12, 2, 101–115.

629 SUJAN, V., DUBOWSKY, S., OHKAMI, Y., ET AL. 2000. Design
 630 and implementation of a robot assisted crucible charging system.
 631 In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE
 632 International Conference on*, vol. 2, IEEE, 1969–1975.

633 TALTON, J. O., GIBSON, D., YANG, L., HANRAHAN, P., AND
 634 KOLTUN, V. 2009. Exploratory modeling with collaborative
 635 design spaces. *ACM Transactions on Graphics-TOG* 28, 5, 167.

636 UMETANI, N., IGARASHI, T., AND MITRA, N. J. 2012. Guided
 637 exploration of physically valid shapes for furniture design. *ACM
 638 Trans. Graph.* 31, 4, 86–1.

639 WESTON, R. 2003. *Materials, form and architecture*. Yale Uni-
 640 versity Press.

641 WILLIS, K. D., XU, C., WU, K.-J., LEVIN, G., AND GROSS,
 642 M. D. 2011. Interactive fabrication: new interfaces for digital
 643 fabrication. In *Proceedings of the fifth international conference
 644 on Tangible, embedded, and embodied interaction*, ACM, 69–72.