

# 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: compositions designed by participants in a workshop. Right bottom: the outcome as an interior screen (2m x 0.9m).

## 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 the irregular properties. In this paper, we take the diversity as playful inputs for design task, and present our participatory game-based design-fabrication platform for customized architectural elements. Taking tree branches as case study of materials with native forms, our online game *BranchConnect* enables users to design 2D networks of branches and realize the design. As the generated design is realizable with ordinal 3-axis CNC milling machines, each connection has a customized unique joinery adapted to the native branch shapes. 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, enabling to integrate them and calculate the performance of the overall structure. 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 resources directly as they are found. Such a direct use can not compete with highly standardized materials and construction system, however, the uniqueness of native shapes is a valuable quality which is lacking in standard materials. As the material is found locally, building and living get much closer, thus people using a facility can easily commit design and fabrication, fostering the sense of belonging to the community []. **TODO: citation** Locally found material easily connects 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 shapes 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 typically the use of native shapes costs more than standardized construction system.

This paper aims to make the above-mentioned qualities of materials with native shapes more accessible by digital technologies. We use locally found branches as materials with native shapes, 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: annual 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 shapes.

The key technical difficulty with materials with natives shapes 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 shapes 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 compositions 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 shapes of woods from orthogonal connections. Physically collected branches are digitally scanned and stored in a cloud database *BranchCollect*, and offline application *Branch Importer* analyzes shapes 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 shapes. We hope our method sheds lights on materials such as waste from demolition of buildings for various design applications.

Our contributions are summarized as follows.

- a workflow enabling to take natural materials with native shapes as design components.
- online platform to collect irregularly shaped materials and serving them to an online game as a design driver.
- fast and robust intersection detection algorithm for fabricating customized non-orthogonal joineries.

## 2 Related Work

3D printers and CNC routers made digital fabrication more accessible, and pre-fabricated customized parts are often used in buildings nowadays [Knaack et al. 2012]. According to Pye, these parts 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” with digital technologies, interactive fabrication enables machines to pick up uncertain happenings and react on it [Willis et al. 2011]. Mueller developed interactive laser

cutting, 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 branches with diverse shapes. While Crowdsourced fabrication project took advantage of humans-in-the-loop in their fabrication system [Lafreniere et al. 2016], our work puts emphasis on crowd-sourced design. As a crowd-sourced design system, Jerry et. al., developed a platform for light users to design trees and plants [Talton et al. 2009]. Our design process is not parametric modeling, also directly linked to physical world. **TODO: ill-logic**

There are few works that take natural materials with native shapes 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]. Getting out from factories, autonomous robotics in construction site is a hot topic among roboticists [Feng et al. 2014]. *In-situ Fabricator* is a system which could be installed in construction site and co-operated with human workers [Dörfler et al. 2016]. Once robot is autonomously localize itself in such an environment, it can build foundational structure for further construction [Napp and Nagpal 2014]. Using locally found objects on-site, such a system can be much simpler.

While these projects demonstrated the capability of digital fabrication processes to handle irregularly shaped materials, design process is still dependent on a designer or architect who has experiences with materials or has access to special software.

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

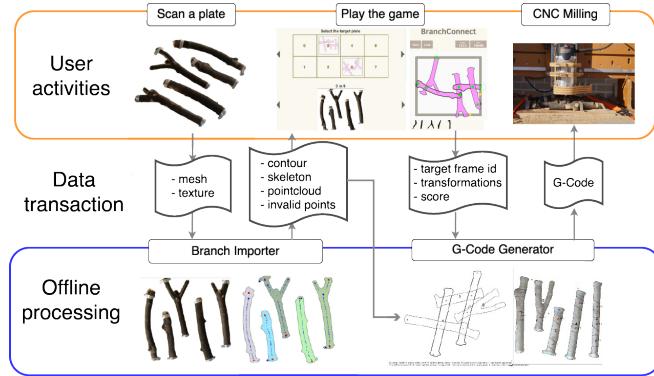
- Including stakeholders in creation of one's environment.
- Experimenting diverse design tastes from multiple point of views.
- Solving complex design tasks with full of diverse solutions.

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 shapes. 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 which are collecting data for solving medical or engineering problems, our game intends to provide a participatory design platform as a solution for social and cultural problems context-aware architectural design with locally found natural materials. **TODO: ambiguous!**

### 183 3 Workflow

184 The overall workflow is Figure 1

185 In this section, we shortly walk through the pipeline in our workflow  
 186 and describe three steps: Digital Model Acquisition, the Game  
 187 System, and Fabrication. Our system takes textured mesh model or  
 188 point cloud with colored vertices. As complete mesh model provides  
 189 more robust result with 3D shapes of branches, we describe  
 190 our process based on mesh model as input (see Figure 2 ).



191 **Figure 2:** A pipeline from model acquisition to fabrication.

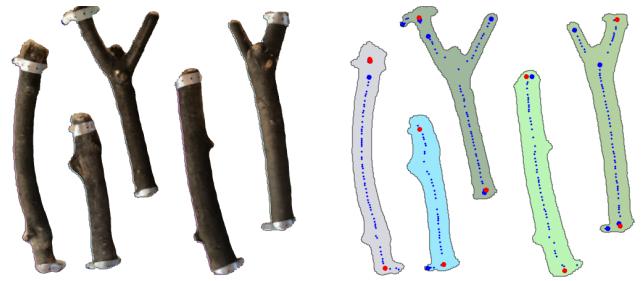
#### 3.1 Digital Model Acquisition

192 There are various methods and software available for scanning 3D  
 193 models and detecting objects. As for the scanning, we describe  
 194 in the Experiment. Taking mesh model with colored texture, our  
 195 *Branch Importer*, integrates necessary functions such as object de-  
 196 tection, skeleton extraction, branch type classification, and fixture  
 197 location detection.

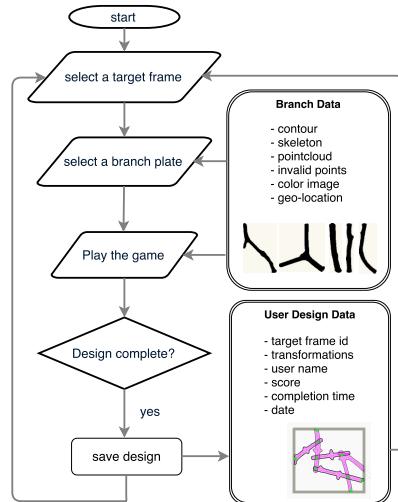
198 After segmenting branches from a plate by height threshold, each  
 199 branch is detected using *findContour* in OpenCV<sup>1</sup>. The obtained  
 200 2D contours are used for extracting skeletons and clustering point  
 201 cloud in the mesh model. Contours are triangulated and skeleton  
 202 points are extracted from middle points on edges of triangles. These  
 203 middle points are compared with binary top view image of the mesh  
 204 model (contours are filled with different colors and the background  
 205 is white). If the point is inside of a contour, the middle point is  
 206 counted in skeleton points. Metal fixture locations are also filtered  
 207 out due as they have bright reflections on original images, however,  
 208 we also double check with simple mouse-clicks to ensure these in-  
 209 valid points. After extracting valid middle points, connectivity of  
 210 skeletons is analyzed by angle of three adjacent skeleton points. If  
 211 the angle stays within a threshold bound, a point is counted in a sub-  
 212 branch, otherwise, new sub-branch is created. Evaluating the num-  
 213 ber of sub-branches, the branch is morphologically classified. Most  
 214 branches are categorized in three shapes: *I*, *V*, and *Y* shapes. *I*  
 215 shape has a straight continuous polyline, *V* has an inflection point,  
 216 and *Y* has a splitting point. The acquired information is stored in a  
 217 cloud database.

#### 218 3.2 BranchConnect: The Game

219 The game objective is to collect valid compositions of branches  
 220 which structurally sounding and possible to fabricate with ordinal  
 221 3 axis CNC milling machines. The workflow of the game is illus-  
 222 trated in Figure 4



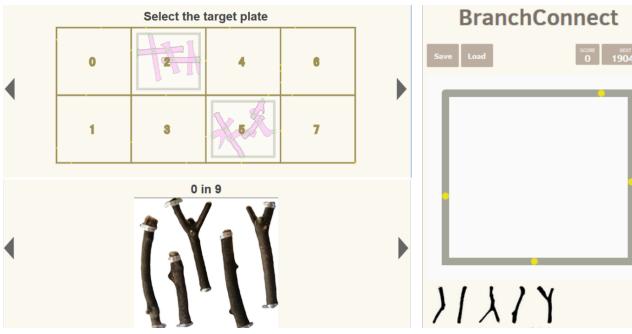
193 **Figure 3:** An overview of *Branch Importer*. Left: a top ortho-view  
 194 image of textured mesh model. Right: Extracted skeletons (blue  
 195 dots). The beginning of skeletons is shown bigger dots. Red dots  
 196 are invalid points.



197 **Figure 4:** The workflow of *BranchConnect*. Branch data and user  
 198 data are stored on cloud database.

219 Firstly a user selects a frame indicating multiple target points to be  
 220 connected, and then selects a set of branches fixed on a plate (The  
 221 left in Figure 5 ). After selecting the target frame and the set of  
 222 branches, the user is guided to the game interface, consisting of the  
 223 frame with the target points, and the set of available branches (The  
 224 right in Figure 5 ). The user picks a branch from the available set  
 225 on the bottom, and then places it in an arbitrary 2D pose through  
 226 basic manipulations such as move, rotate, and mirror. The number  
 227 of available branches differs depending on plates. With the feasible  
 228 diameter of branches (over 2cm) and the plate size (50cm x 50cm),  
 229 the number of available branches is most likely up to six. Within  
 230 the limited number of branches, the user bridges all the target points  
 231 by connecting all the used branches in one group. The game is  
 232 completed when all the target points are connected by branches. For  
 233 higher score, the user can modify the design after the completion.  
 234 After completing the modification, the design is submitted and sent  
 235 to *G-Code Generator*.

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

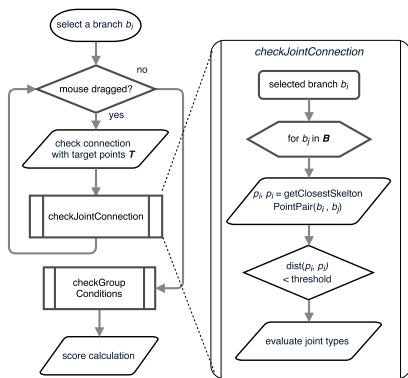


**Figure 5:** Left: the selection interface for target frames (top) and branch panels (bottom). Right: the start interface of the game.

## 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 more limited in 2D and intersection detection only. In broader phase, simplified skeletons are used to find the pair of closest skeleton points between two branches. After finding the pair, skeletons with higher resolutions are used.

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. 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. An overview is illustrated in Figure 6.



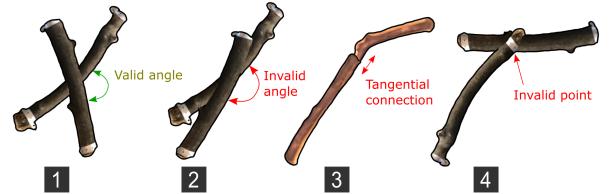
**Figure 6:** Left: an overview of the game system. Right: joint condition checking process.

## Joint Condition

Joint is the essential entity not only in the game but also the entire project including the fabrication process. The closest skeleton point pair is obtained. We accept crossing joints only because they are structurally stable, relatively simple to fabricate, and creates diverse

designs. To ensure the structural performance, we set a connection angle within a range from 45 to 135 degrees. Joints close to metal fixtures are also counted as invalid. Valid and invalid joints are displayed with green and red respectively. The color-code feedback is also given when target points on a frame is connected with a branch. Figure 7 illustrates valid and invalid joint conditions.

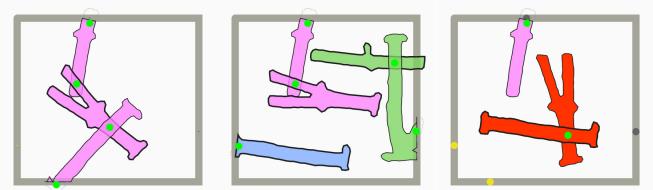
To describe the process, we let  $\mathcal{B}$  denotes a set of all the available branches, and each branch as  $b_i \in \mathcal{B}$ . While the process checks joint condition through all the branches  $\mathcal{B}$ , each detected joint is stored in each branch  $b_i$ , categorized in different conditions such as valid and invalid joints denoted as  $\mathcal{J}_{\text{valid},i}$ ,  $\mathcal{J}_{\text{invalid},i}$  respectively, together with the paired branch id  $b_{\text{paired},j,i}$ . When a branch is connected to one of target point  $t_j \in \mathcal{T}$ , the  $t_j$  is stored in  $p_i$  as  $t_{j,i}$ . **TODO: notaion should be improved!**



**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.

## Group Condition

After checking joint conditions of all the pairs of branches, the system checks the number of groups as well as connectivity with the target points on a frame. If a group is not connected to any target point nor other groups, the group is islanded and structurally invalid. While a user is positioning a branch by dragging or rotating, groups are continuously calculated and indicated by simple color (Figure 8).



**Figure 8:** Left: valid group with two target points connected. Middle: valid but three groups. Right: invalid due to the island situation.

After all the joint conditions are checked, we evaluate group conditions. Through checking the all the branches  $\mathcal{B}$ , the first group  $g_0$  is created and stores  $b_0$ . The other paired branch  $b_{\text{paired},i}$  is used to trace the connection with other compared to the traced through the stored paired branch in each valid joint  $j_i$ . in all groups  $\mathcal{G}$ . The game is completed when the number of  $\mathcal{G}$  is one, and all the target points are connected with the group.

## Score Calculation

We denote the numbers of valid and invalid joints on each branch  $b_i \in \mathcal{B}$  as  $N(\text{valid}_i)$ ,  $N(\text{invalid}_i)$  respectively, the number of groups as  $N(\mathcal{G})$ , the number of islanded groups as  $N(g_{\text{islanded}} \in \mathcal{G})$ ,

**Algorithm 1** Group Condition Update Algorithm

---

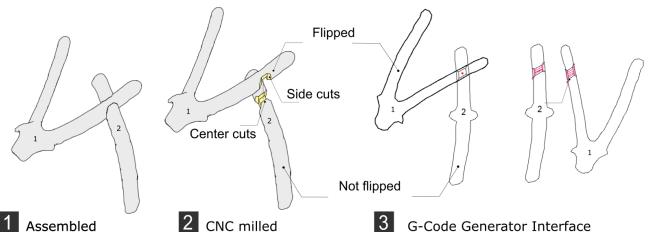
```

1: function UPDATEGROUPS( $\mathcal{B}$ )
2:   Reset all the groups  $\mathcal{G}$ 
3:   Create new group  $g_1$ 
4:    $g_1$  add  $b_1$ 
5:   if  $b_1$  has connected target point  $t_1$  then
6:      $g_1$  set  $t_1$ 
7:    $\mathcal{G}$  add  $g_1$ 
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  $b_{paired,i} \in \mathcal{P}_j$  has  $b_j$  then
13:           $g_j$  add  $b_i$ 
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:           $g_{new}$  add  $b_i$ 
23:           $\mathcal{G}$  add  $g_{new}$ 

```

---

Some fabrication factors such as mirror and invalid points are already considered by *Branch Importer* and the game system of *BranchConnect*. In this section, we describe the process of joinery generation. Each joinery geometry is different but has same topology: two plane surfaces on the sides of branches and one plane top surface. The geometry creates rigid connection with the irregularly shaped sections of branches. 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 side cuts milling paths, which are inwardly offset paths of the original branch contours. **TODO: need to brush up!** The center cuts are paths that are plaining the top surface of the joint. **TODO: describe the cutting height calculation!**



**Figure 10:** An example of intersected pair: 1. an assembled pair of branches. 2. branches after the center and side cuts are milled 3. left: a composition defined by a user right: the original orientations of branches with generated milling paths with red color.

Users can change fabrication parameters such as offset ratio for the side cuts, milling bit diameter, overlapping ratio for defining the center cut depth, feed-rate, moving height and so forth. After confirming the fabrication settings and milling paths, it generates G-Code.

### 4 Experiment

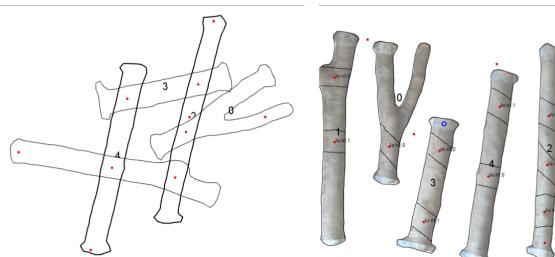
A design and fabrication workshop was organized to examine the validity of our system with a specific design target and a location. We selected a public community house where people in the community share the space and regularly use the facility and participants were selected among them. Participants are four children aged 10 and under (4, 7, 9, and 10 years old) and two parents (Figure 11). We specifically selected children with the age range as non-experts without experiences in computational design or digital fabrication, also for observing the clarity and attractiveness of the game.

The goal for the participants is 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 are asked to follow the entire process from collecting and fixing the branches on a plate, scanning the plate, complete designs by playing the game, and assembly after the CNC milling. Participants were informed about the goal of the workshop, and each process was introduced by experienced tutors.

### 3.3 Fabrication

After a design is selected for fabrication, the validity of the design is further inspected with a high-resolution model. The *G-Code Generator* was developed for fine-tuning the design by checking real-time feedback of updated joineries on branches with scanned orientations (see Figure 9).



**Figure 9:** Interface of G-Code Generator. Users can tweak the design on the left side and immediately see the updated joints and milling paths on the right. **TODO: can this image be integrated with the other one?**



**Figure 11:** An overview of the workshop. 1.the overview of the space. 2.collect branches. 3.cut 4.attach on a plate 5.scan 6.play the game 7.CNC milling.

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 assemble branches after joineries were milled.

## 352 4.1 Setup

### 353 System and Hardware Setup

354 We used two iPad minis with iSense scanners 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 355 well as operating the milling machine. The scan area of iSense is 356 500mm x 500mm, and the milling machine's stroke along z-axis 357 is 70mm, which provide geometric constraints for available branch 358 sizes. *BranchConnect* was hosted at *Heroku* cloud server<sup>2</sup>, and we 359 used *MongoDB*<sup>3</sup> as a cloud database. 360

### 363 Preparation of Branches

364 The participants were asked to collect branches with 2 - 10 cm in 365 diameter. The lower bound of the diameter is for milling process 366 would not destroy them, and the upper bound is for the limit 367 of z-stroke of the CNC router. The collected branches are cut 368 in arbitrary lengths, not longer than 500 mm due to the limit of 369 scanning area. 370

371 As our game system and fabrication process take 3D branch shapes 372 as 2D contours with height map, these constraints work positive for 373 the system automatically filtering out branches with large 3D twists. 374 We ask participants to scan with iPad + iSense and prepare feasible 375 mesh model by themselves. After obtaining mesh models, tutors 376 import models from iPads to a laptop and upload them to database 377 by *Branch Importer*. 378

### 378 User Experiences with the System

379 After models were uploaded to the server, users can see that their 380 plates are added in the selectable branch plates with their names 381 and locations. Users can access to the start page by PC and mobile 382 devices. We prepare both options and let participants choose a 383 device. 384

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.

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

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

## 405 4.2 Results

406 The entire workshop took 4.6 hours to complete the whole process, 407 including introduction, moving, and pauses. Table 4.2 shows durations 408 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
Miscellaneous	1.0	21.7
In total	4.6	100

### 410 Collecting Branches

411 The diameter and length constraints for available branches worked 412 as guidelines for participants rather than restricting finding and cutting 413 arbitrary branches. After cutting branches in certain lengths, 414 participants fixed branches on plates by thin metal plates with screw 415 holes. It was straightforward for them to firmly fix branches so that 416 they are not moved during milling process. These fixture points are 417 counted as invalid points in the game where joinery points can not 418 be generated. The participants built two plates with three and five 419 branches fixed on each plate. 420

### Model Acquisition

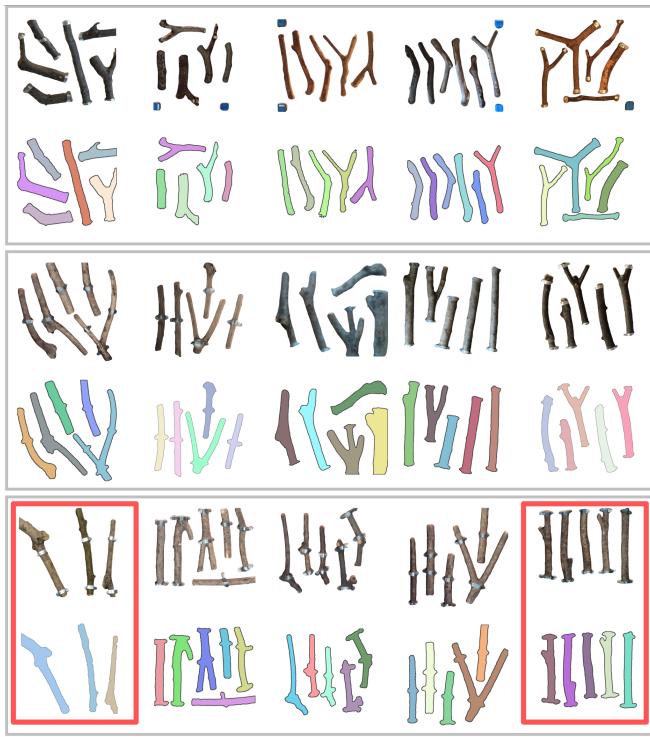
iSense 3D scanners come with an intuitive software for scanning and modifying models. After we gave an instruction, most of participants practiced several scans and successfully scanned models without any problem.

Each scanning and re-touching took 2-3 minutes, and 30 seconds for generating data by *Branch Importer*. Including the prepared panels previously, we scanned 15 plates in total, 75 branches, and 35.3m of total length including sub-branches. **TODO: check the length again!** We got 40 I shaped branches, 19 V shaped branches, and 16 Y shaped branches. The result is shown in Figure 12.

### Design with the Game

Most participants used iPads for navigating pages and playing the game. All the participants understood the goal immediately, however, they had difficulties with mobile touch interface, such as rotation and flipping operation by gestures.

One participant switched to play by a PC for more precise control due to the problem. All the participants chose to develop their own



**Figure 12:** 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.

464 **TODO: check offsetted correct english** After milling was finished  
 465 and when branches were assembled, six pairs of branches were  
 466 loosely connected because the calculated contours were 2-3mm  
 467 eroded than the actual sizes. We avoided this problem by trimming  
 468 branches from 2-5 mm higher than the plate surface. After this  
 469 operation, the rest of connections were tightly connected.  
 470

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

- 474 • deformation of mechanical parts of the CNC router
- 475 • not dense resolution of acquired contours of branches
- 476 • misaligned orientation of the plate compared to the scanned  
 477 model

478 To avoid the misalignments, we modified the *G-Code Generator*  
 479 so that an operator can freely adjust the absolute origin of the gen-  
 480 erated milling paths. The origin was usually set with around the  
 481 center of the plate. After this modification, the misalignment from  
 482 joint center was reduced with 5 mm off at the maximum misalign-  
 483 ment. Branches could absorb 3-5 mm misaligned joint positions  
 484 due to the elasticity of the wooden materials. The misaligned joint  
 485 positions worked as post-tensions, solidifying the structure. We as-  
 486 sume that this is only applicable when an applied bending moment  
 487 and cut surface at a joint is orthogonal or not too much off from  
 488 orthogonal. **TODO: this sentence**

### 489 4.3 Summary

490 The objective of the workshop was to observe the validity of our  
 491 system for non-expert users. We could see that participants aged  
 492 10 and under were capable of following the workflow. Participants  
 493 were encouraged to contribute to the ongoing design with the online  
 494 platform, and successfully provided designs from available branch  
 495 plates.

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

503 A noteworthy fact was that one participant with four years old failed  
 504 to complete the game, but insisted on accepting his design to be fab-  
 505 ricated by the CNC router. We took these requests as their commit-  
 506 ment to the entire workflow, as we observed that all the participants  
 507 insisted on continuing the fabrication process.

## 508 5 Conclusion

### 509 Summary

510 In this paper, we presented a workflow to design and fabricate with  
 511 branches which are not large enough for producing standardized  
 512 building components. Our workflow was validated with case-study  
 513 with lower-aged participants without design and fabrication experi-  
 514 ences. Our online platform to store geometries of scanned branches  
 515 and compositions of branches was reachable by people with vari-  
 516 ous backgrounds. Our light-weight branch joint detection algorithm  
 517 was also validated by running on the browser game, contributed to  
 518 the accessibility of the presented workflow. Together with the ac-  
 519 cessibility, the intuitive interface was simple enough for non-expert  
 520 users. We successfully built a network of branches with rigid joints

438 designs from the scratch, although they had instruction about the  
 439 "continue existing designs".

440 Although there is no time limit in the game system, we set 30 min-  
 441 utes for playing the game, and 8 solutions were given by partici-  
 442 pants. xxx frames were completed per participants and xxx partici-  
 443 pants completed the whole eight target frames. The average score is  
 444 xxx, and average playing duration was xxx to complete each target  
 445 frame.

### 446 Overall Design Consensus **TODO: this section might be re-** 447 **removed**

448 As the target frame selection page can display designs not only  
 449 from one user but also from all the others at once, we could get  
 450 an overview of design options. The designs are displayed as score  
 451 descending order but limited numbers, we could find feasible solu-  
 452 tions with mostly all the target points were bridged. As participants  
 453 were excited by seeing their branches and designs, we took some  
 454 of their solutions and their plates for the fabrication although their  
 455 solutions did not satisfied completion of the game.

### 456 Fabrication

457 We did not have major problems for converting designs to G-Code  
 458 milling paths as we encountered major issues regarding the fabrica-  
 459 tion before the workshop, which are reported in this section.

460 The main problem was the accuracy of acquired contours. We  
 461 observed most of scanned models had occluded regions between  
 462 plates and branches, which create interpolated faces during  
 463 solidifying process, resulting in outwardly offsetted contours.

generated by our joinery milling path generator. Each of joints has customized lapped-joint geometry, which is challenging to design and fabricate.

## Limitations and Future Work

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

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

## Acknowledgements

We appreciate the public house for hosting the CNC router and providing participants for the user study. We thank to the film editor, Shin Yamane for shooting our video clips. We also thank to the developer of an online game 2048, Gabriele Cirulli and his contributors, for sharing source code in Github community.

## References

- BOGUE, R. 2013. 3d printing: the dawn of a new era in manufacturing? *Assembly Automation* 33, 4, 307–311.
- CIMERMAN, B. 2000. Participatory design in architecture: can computers help? In *PDC*, 40–48.
- DÖRFLER, K., SANDY, T., GIFTTHALER, M., GRAMAZIO, F., KOHLER, M., AND BUCHLI, J. 2016. Mobile robotic brick-work. In *Robotic Fabrication in Architecture, Art and Design 2016*. Springer, 204–217.
- FENG, C., XIAO, Y., WILLETT, A., MCGEE, W., AND KAMAT, V. 2014. Towards autonomous robotic in-situ assembly on unstructured construction sites using monocular vision. In *Proceedings of the 31th International Symposium on Automation and Robotics in Construction*, 163–170.
- GREENBERG, B., HITTNER, G., AND PERRY, K., 2010. Smart scrap. Accessed: 2016-09-30.
- HAUERT, S., LO, J. H., NACHUM, O., WARREN, A. D., AND BHATIA, S. N. Crowdsourcing swarm control of nanobots for cancer applications.
- HUBBARD, P. M. 1996. Approximating polyhedra with spheres for time-critical collision detection. *ACM Trans. Graph.* 15, 3 (July), 179–210.
- KHOSHNEVIS, B. 2004. Automated construction by contour craftingrelated robotics and information technologies. *Automation in construction* 13, 1, 5–19.
- KNAACK, U., CHUNG-KLATTE, S., AND HASSELBACH, R. 2012. *Prefabricated systems: Principles of construction*. Walter de Gruyter.
- LAFRENIERE, B., GROSSMAN, T., ANDERSON, F., MATEJKA, J., KERRICK, H., NAGY, D., VASEY, L., ATHERTON, E., BEIRNE, N., COELHO, M. H., ET AL. 2016. Crowdsourced fabrication. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, ACM, 15–28.
- LENNON, M. 2005. Recycling construction and demolition wastes.
- LIMPAECHER, A., FELTMAN, N., TREUILLE, A., AND COHEN, M. 2013. Real-time drawing assistance through crowdsourcing. *ACM Transactions on Graphics (TOG)* 32, 4, 54.
- LUKKA, T. J., TOSSAVAINEN, T., KUJALA, J. V., AND RAIKO, T. 2014. Zenrobotics recycler–robotic sorting using machine learning. In *Proceedings of the International Conference on Sensor-Based Sorting (SBS)*.
- MAIRS, J., 2016. Aa design and make students use a robotic arm to build a woodland barn. Accessed: 2016-09-30.
- MONIER, V., BIGNON, J. C., AND DUCHANOIS, G. 2013. Use of irregular wood components to design non-standard structures. In *Advanced Materials Research*, vol. 671, Trans Tech Publ, 2337–2343.
- MUELLER, S., LOPES, P., AND BAUDISCH, P. 2012. Interactive construction: interactive fabrication of functional mechanical devices. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, ACM, New York, NY, USA, UIST '12, 599–606.
- NAPP, N., AND NAGPAL, R. 2014. Distributed amorphous ramp construction in unstructured environments. *Robotica* 32, 02, 279–290.
- NIELSEN, S. A., AND DANCU, A. 2015. Fusing design and construction as speculative articulations for the built environment.
- OLIVER, P. 1997. *Encyclopedia of vernacular architecture of the world*. Cambridge University Press.
- PYE, D. 1968. *The nature and art of workmanship*. Cambridge UP.
- SCHINDLER, C., TAMKE, M., TABATABAI, A., BEREUTER, M., AND YOSHIDA, H. 2014. Processing branches: Reactivating the performativity of natural wooden form with contemporary information technology. *International Journal of Architectural Computing* 12, 2, 101–115.
- SEIKE, K. 1977. The art of japanese joinery.
- SUJAN, V., DUBOWSKY, S., OHKAMI, Y., ET AL. 2000. Design and implementation of a robot assisted crucible charging system. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 2, IEEE, 1969–1975.
- TALTON, J. O., GIBSON, D., YANG, L., HANRAHAN, P., AND KOLTUN, V. 2009. Exploratory modeling with collaborative design spaces. *ACM Transactions on Graphics-TOG* 28, 5, 167.
- WESTON, R. 2003. *Materials, form and architecture*. Yale University Press.
- WILLIS, K. D., XU, C., WU, K.-J., LEVIN, G., AND GROSS, M. D. 2011. Interactive fabrication: new interfaces for digital fabrication. In *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*, ACM, 69–72.