

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 (Computer Numerical Control) router. Right top: branch layouts designed by multiple users. Red arrows indicate layouts by participants in the workshop. Right bottom: the fabricated 2D fence (2000 mm × 900 mm). 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. 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: annual pruning, storing, and chipping or burning with some costs. The size of branches (from 50 -300 mm 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.

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-Code¹ 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 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 interacts with the native forms of irregularly shpaed branches. 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 crowd-sourced 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 such as generic design and 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, which works on browsers

¹G-Code is the generic name for a control language for CNC machines.

on laptop computers or mobile touch 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.

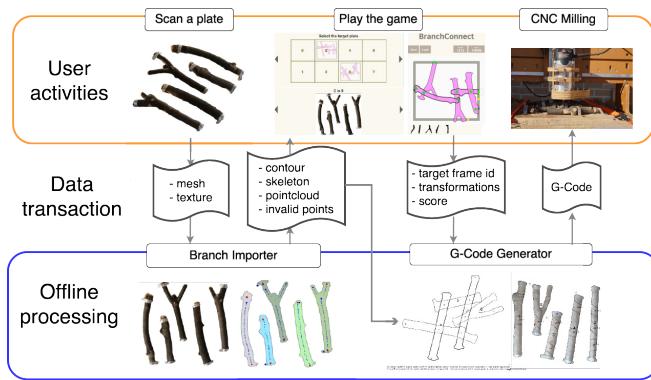


Figure 2: A pipeline from model acquisition to fabrication.

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. 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. 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.

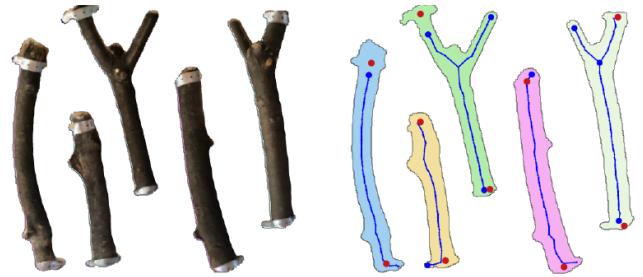


Figure 3: An interface of *Branch Importer*. Left: a top ortho-view image of textured mesh model. Right: Extracted skeletons are shown with blue dots. The beginning of skeletons is shown bigger dots, and the red dots are invalid points defined by a user.

Some fabrication factors such as invalid points due to metal fixtures and flipped (further described in Section 4.1.1) 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 cut) (see Figure 5.2). Side cuts have wedged corners for smooth assembly process. The geometry creates rigid joints with irregularly shaped sections of branches.

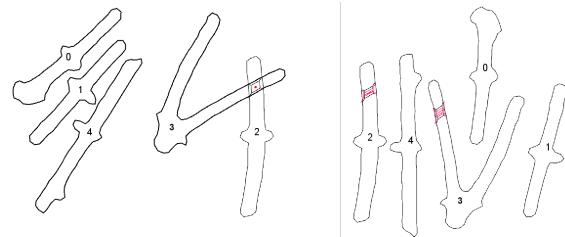


Figure 4: The interface of *G-Code Generator*. Left: a layout defined by a user. Right: the original orientations of branches with generated milling paths with red color.

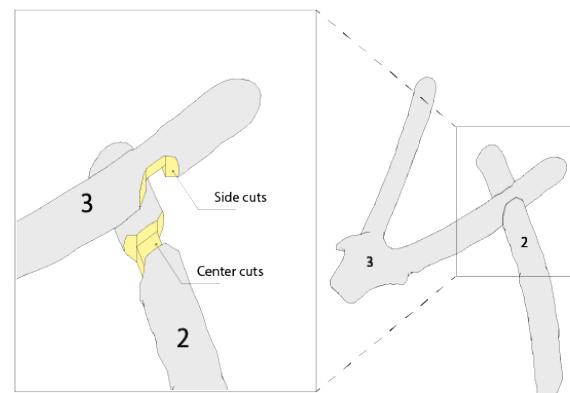


Figure 5: An example of joinery. Left: branches after the center and side cuts are milled. Right: an assembled pair of branches.

Similar to the joint searching process with skeletons (see Section 4.1.1), the *G-Code Generator* searches a set of four closest points from high-resolution contours, expecting that every intersected con-

tour has four curves. After finding the four closest points, it trims two curves from each contour of branch. 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.

4 BranchConnect: The Game

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 feasibility of a given layout. Similar to our game, the work *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 limit valid layout space by selected joint conditions, and group conditions. We also assume that every fabricated joint works as a rigid joint, thus single connection is counted as stable to hold a pair of branches.

Describing user experience, firstly a user selects a target 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 at the bottom (the right in Figure 6). The user picks a branch from the available set on the bottom, and drag&drop it to the inside of the target frame. By selecting and dragging a branch, the user searches a good 2D pose through geometric manipulations such as move, rotate, and horizontal flip (or mirror). 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 conditions of joint and group are indicated with simple color-code. Together with the color-code, the score update guides the user to form a valid design. Within the limited number of branches, the goal is to bridge all the target points by connecting all the used branches in one group. For higher scores, the user can keep modifying the design, and save it to the database.

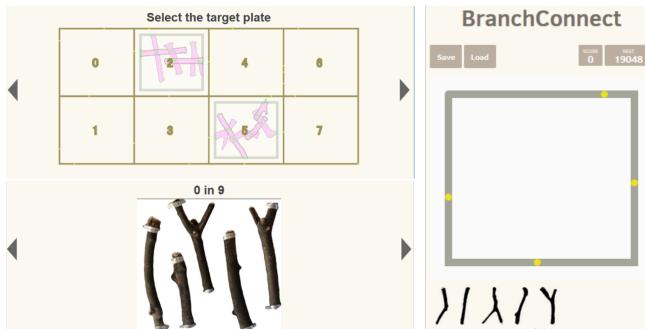


Figure 6: Left: the selection interface for target frames (top) and branch panels (bottom). Right: the start interface of the game.

4.1 The Game System

There are many physics simulation libraries for game, however, our game needs to detect intersected branch pairs, thus collision detection with physics engines is overkill for our browser game. Also, branches have 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 game extensively uses down-sampled skeletons of branches.

Hubbard and Philip developed collision detection by representing an object with hierarchical 3D spheres aligned on a skeleton [Hubbard 1996]. Our game takes similar approach but limited in 2D, but more focused on searching fabricatable joints. In the game, down-sampled skeletons are used to find the pair of closest skeleton points between two branches. When a branch is selected, the system searches the closest skeleton point of the selected branch with other skeletons of available branches. More precise joint calculation with high-resolution contours is further described in Section 3.2.

4.1.1 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 (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). 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.

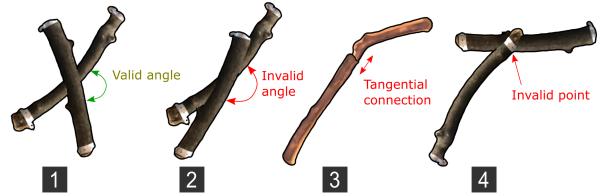


Figure 7: Joint conditions. 1. valid joint. 2. invalid for violating the angle. 3. invalid tangential connection. 4. invalid for connecting on a fixture point.

To describe the joint update process, let each branch b_i be a member of a set of the branches \mathcal{B} dropped inside of the target frame by a user. The user freely choose $\mathcal{B} \in \mathcal{B}_{\sqrt{\square}}$, where $\mathcal{B}_{\sqrt{\square}}$ is the branches fixed on a selected plate, denoted as $\mathcal{B}_{\checkmark\Box}$. We accept one joint with a pair of branches, but a branch can have multiple joints with other branches. The process starts from the selected branch b_i and updates joint conditions of the selected branch with paired branches. When an intersected pair is detected, it stores j -th joint $j_{i,j}$ in b_i with joint conditions (see Figure 7). After evaluation, as in Figure 7, we have joint labeled as valid or invalid. When a branch b_i is connected to one of target points $t_j \in \mathcal{T}$, the target point t_j is stored in b_i . Note that we also take one target point for each branch. This process as well as group condition update process are illustrated in Figure 8.

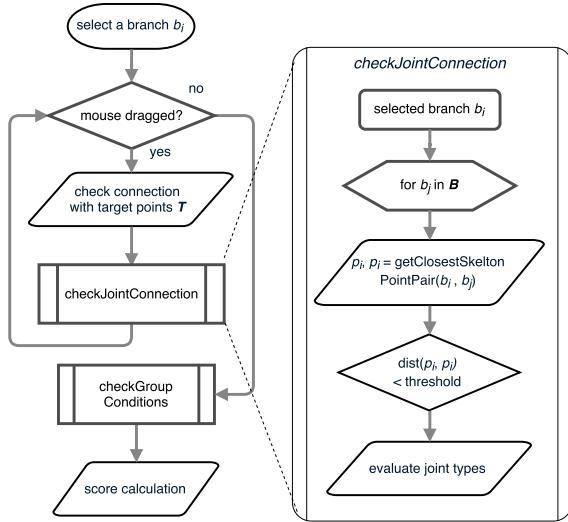


Figure 8: Left: an overview of the game system with 1. joint update 2. group update and 3. score calculation. This process is iteratively executed while a user is exploring layout by dragging a branch. The joint update process is further illustrated in the right, and group condition update is described in Algorithm 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  $b_{paired,i} \in \mathcal{P}_j$  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  $\mathcal{G}$ 

```

325 **4.1.2 Group Condition**

326 After updating joint conditions of all the paired of branches, the
327 system updates the number of groups as well as its connection with
328 the target points on a frame. If a group is not connected to any tar-
329 get point nor other groups, the group is *Islanded* and structurally
330 invalid. While a user is positioning a branch by dragging or ro-
331 tating, groups are continuously calculated and indicated by simple
332 color (Figure 9).

345 **4.1.3 Score Calculation**

346 We calculate the score with weighted sum of following entities: the
347 numbers of valid and invalid joints on each branch, the number of
348 groups as $N(\mathcal{G})$, the number of islanded groups as $N(g_{islanded} \in \mathcal{G})$,
349 the number of bridged target points as $N(t_{bridged,i}) \in \mathcal{T}$. The
350 trimmed lengths of branches which are connected with target points
351 are denoted as $trimmed(t_j, b_i)$. The score is weighted sum of these
352 joint and group conditions, denoted in Equation (1). The weights
353 $w_1 \dots w_5$ are non-negative weight coefficients pre-adjusted in
354 advance by authors.

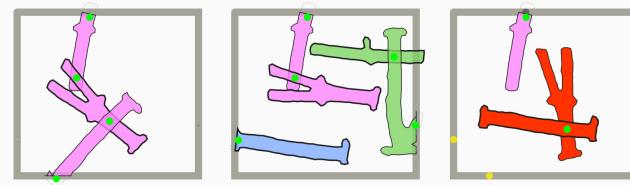


Figure 9: Left: valid group with two target points connected. Middle: valid but three groups. Right: invalid due to the *Islanded* situa-
tion.

355 After all the joint conditions are updated, we evaluate group con-
356 ditions. Iterating $b_i \in \mathcal{B}$, the first group g_0 is created and stored as
357 b_0 . When a branch is connected to a target point, the graphics of the
358 point changes, and the branch and its belonging group's color also
359 changes. When a group bridges a pair of target points, a special
360 score is added and displayed in a pop-up square, also the graphics
361 of target point's changes. The branch connected to the target point
362 is trimmed at the target point, and the trimmed length is subtracted
363 from the score. The game is completed when the number of \mathcal{G} is
364 one, and all the target points are connected with the group. The al-
365 gorithm which checks group conditions is described in Algorithm
366 1.

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

355 **5 Case Study**

356 A design and fabrication workshop was organized to examine the
357 feasibility of our system with a specific design target and a loca-
358 tion. We selected a public community house where people in the
359 community share the space and regularly use the facility. Partici-
360 pants of the workshop were selected among them, who were four
361 children (aged 4, 7, 9, and 10) and two parents (Figure 10). We
362 specifically selected children with this range of age as non-experts
363 without experiences in computational design or digital fabrication,
364 also for observing the clarity and attractiveness of the game.

365 The entire workshop was filmed and summarized in the supplemen-
366 tary video material. Prior to the workshop, we have adjusted the
367 weights in Equation 1 manually to ensure that the feasibility of lay-
368 outs is correlating to the scores (see Figure ??). Through adjusting

369 the game setting, we have built six target frames.

370 The goal for the participants was to contribute to an ongoing design
 371 and fabrication process of screen wall ($2000\text{ mm} \times 900\text{ mm}$) consisting of eight rectangles ($500\text{ mm} \times 450\text{ mm}$). Six frames were
 372 already designed and built, thus the rest two frames were prioritized
 373 for them to design and fabricate in this workshop.

375 Participants were informed about the goal of the workshop, and
 376 each process was introduced by experienced tutors. The processes
 377 were from collecting and fixing the branches on a plate, scanning
 378 the plate, complete designs by playing the game, and assembly after
 379 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.

380 5.1 Preparations and User Experiences

381 System and Hardware

382 We used two iPad minis with iSense depth cameras attached for
 383 scanning branches, and a 3 axis CNC milling router with a 6 mm
 384 diameter milling bit. We used a laptop PC for running *Branch Im-*
385 porter and *G-Code generator*, as well as operating the milling ma-
 386 chine. The scan area of iSense camera is $500\text{ mm} \times 500\text{ mm}$, and
 387 the milling machine's stroke length along z-axis is 70 mm , which
 388 provide geometric constraints for available branch sizes. *Branch-*
389 Connect was hosted at *Heroku* cloud server ², and we used *MongoDB*
390 ³ as a cloud database.

391 Preparations

392 The participants were asked to collect branches with $20 - 100\text{ mm}$
 393 in diameter. The lower bound was for the milling bit size, and the
 394 upper bound was for the limited length of z-stroke of the CNC
 395 router. The collected branches were cut in arbitrary lengths, not
 396 longer than 500 mm due to the limit of scanning area. As our game
 397 system and fabrication process take 3D branch shapes as 2D con-
 398 tours (with limited use of point cloud), these constraints were ben-
 399 efitial for the system by filtering out branches with large 3D twists.
 400 The diameter and length constraints worked as guidelines for partic-
 401 ipants rather than restricting finding and cutting arbitrary branches.
 402 The number of available branches per plate was different depending
 403 on branch sizes. Within the feasible diameter of branches and the
 404 plate size, the number of available branches was most up to six (see
 405 Figure 11).

406 After cutting branches in certain lengths, participants fixed
 407 branches on plates by thin metal fixtures with screw holes. After an
 408 instruction, participants successfully fixed branches by themselves.
 409 They built two plates with three and five branches fixed on each
 410 plate.

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/>

411 After fixing branches on plates, we asked participants to scan them
 412 prepare feasible mesh model on iSense application running on iPad
 413 mini. Thanks to the intuitive interface of iSense, participants prac-
 414 ticed several scans and successfully scanned models without prob-
 415 lem. After obtaining mesh models, tutors imported models from
 416 iPads to a laptop and uploaded them to the database by *Branch Im-*
 417 *porter*.

418 User Experiences

419 As for more general user experiences with the game, see Section
 420 4 as well as the video material. In this section, we describe more
 421 specific user experiences and feedback from participants.

422 All participants used iPads for navigating pages and playing the
 423 game. They had difficulties with mobile touch interface, such as ro-
 424 tation and flipping operation by gestures. We had several requests
 425 from participants regarding the game interface but also related to
 426 the workshop organization. Several participants requested to al-
 427 low multiple branch plates for designing a frame, or even remove
 428 the target frame and let them freely design with branches. Also a
 429 participant who gave up the game with iPad requested additional
 430 buttons for mobile touch interface, such as to keep an active branch
 431 selected. The participant with four years old failed to complete the
 432 game. He insisted on accepting his design to be fabricated. Simil-
 433 arily, a branch plate made by a participant had only three branches,
 434 which was not enough to fulfill bridging target points, however, the
 435 participant insisted on accepting it in the selection.

436 Global Design Consensus and Fabrication

437 As the target frame selection page could display all layout designs,
 438 we could get an overview of design options. The layout designs
 439 were displayed as score descending order with limited numbers
 440 (three top highest scores for each target frame), we could find fea-
 441 sible layout designs easily with mostly all the target points were
 442 bridged. As participants were excited by seeing their branches and
 443 designs, we took two invalid layout designs and one plate which
 444 did not have enough branches for the global design. After selecting
 445 layouts, an experienced tutor operates *G-Code Generator* as well as
 446 the CNC router. Participants were asked to assemble branches after
 447 joineries were milled.

448 5.2 Results

449 The entire workshop took 4.6 hours to complete the whole process,
 450 including introduction, moving, and pauses. Table 5.2 shows dura-
 451 tions 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 |

453 Model Acquisition

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. We got 59 branches
 458 with a single skeleton, 16 branches with multiple skeletons for
 459 grafting. The result is shown in Figure 11.

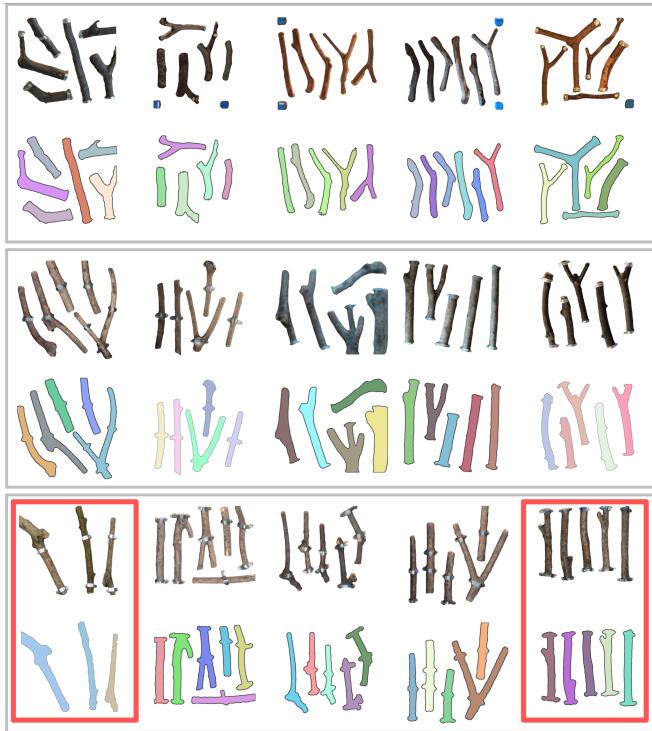


Figure 11: An overview of all the 15 scanned plates for the construction of the fence. Top raw of each set shows ortho-top views of scanned mesh models, and the bottom raw is the recognized branches with randomly assigned colors. The red-lined rectangles indicate the plates built by participants in the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

483 the center of planned joints. Multiple reasons could be considered
 484 as reasons such as,
 485

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

489 To avoid the misalignments, we modified the *G-Code Generator*
 490 so that an operator can freely adjust the absolute origin of the gen-
 491 erated milling paths. The origin was usually set with around the
 492 center of the plate. After this modification, the misalignment from
 493 joint center was reduced with 5 mm off at the maximum misalign-
 494 ment. Branches could absorb 3-5 mm misaligned joint positions
 495 due to the elasticity of branches, and solidifying the structure with
 496 residual stresses from misalignments.

6 Conclusion

498 In this paper, we presented a workflow to design and fabricate with
 499 branches with their native forms, which are not large enough for
 500 producing standardized building components. Our workflow was
 501 validated by the case study with lower-aged participants without
 502 design and fabrication experiences.

503 Our online platform with stored scanned branches is accessible and
 504 multiple users can submit design layouts and explore a global de-
 505 sign. Our branch joint detection and group condition update al-
 506 gorithms are running on the browser game which can be accessed
 507 from laptops and mobile devices, contributing to the accessibility
 508 of the presented workflow. Together with the accessibility, the intui-
 509 tive interface was simple for non-expert users, validated by the case
 510 study. We successfully built a network of branches with rigid joints
 511 generated by our joinery milling path generator. Each of joints has
 512 customized lapped-joint geometry, which extends design possibili-
 513 ties of branches or woods with their native forms.

514 Our workflow touches many developed research areas such
 515 as skeleton extraction, structural optimization, object detec-
 516 tion/recognition, and data-driven design-fabrication. Focusing on
 517 the use of native forms, each step of our workflow has potential
 518 to contribute to each area with the use of native forms of natural
 519 materials. Also, our workflow was developed based on the
 520 participation of users, thus the entire process is not necessarily
 521 automated, however, some tasks could be improved to assist users.

523 The skeleton extraction could take incomplete point-set directly
 524 from original tree branches before they are cut in length. With
 525 data-driven approach, the system could distinguish trees and which
 526 part of tree the branch from. With morphological analysis, the
 527 system could suggest users where to cut branches to achieve user-
 528 defined target design. Structural and geometrical validity/invalidity
 529 of obtained materials could be analyzed. Our workflow requires
 530 branches to be fixed on a plate, which takes the longest duration
 531 in the workflow except for in-between tasks such as moving and
 532 pausing. Using a robotic manipulator with a gripper, the attaching
 533 process could be skipped.

534 Our game system is limited in 2D, whereas original branch forms
 535 have rich 3D geometry with textures. In our case, these informa-
 536 tion was used in limited ways such as in skeleton extractions and G-
 537 Code generation. Despite of successfully fabricated non-orthogonal
 538 joineries, we did not complete the attaching branches to the target
 539 frames, as we prioritized to validate branch-branch joineries. Our
 540 layout design process is fully dependent on users with limited feed-

Design with the Game

461 One participant switched to play by a PC for more precise con-
 462 trol due to the problem. Interestingly, all the participants chose to
 463 develop their own designs from the scratch, although they had in-
 464 struction about the "continue existing designs".

465 We set 30 minutes for playing the game, and eight layout designs
 466 were given by participants. Two frames were completed per partici-
 467 pants and two participants completed the whole eight target frames.
 468 The average score was xxx, and average playing duration was xxx
 469 to complete each target frame. **TODO: recheck the numbers by**
 470 **server**

Fabrication

472 We observed most of scanned models had occluded regions be-
 473 tween plates and branches, which create interpolated faces during
 474 solidifying process, resulting in outwardly offsetted contours.
TODO: check offsetted correct english After milling was finished
 475 and when branches were assembled, six pairs of branches were
 476 loosely connected because the calculated contours were 2-3 mm
 477 eroded than the actual sizes. We avoided this problem by trimming
 478 branches from 2-5 mm higher than the plate surface. After this
 479 operation, the rest of connections were tightly connected.

481
 482 We also observed that many milling paths were 5-10 mm off from

541 back during design process. The game can provide suggestive feed-
 542 back with structural analysis of each joint and entire structure.

543 Our joint and group detection algorithms are limited with materials
 544 with skeletons, and our joinery generator is limited to branches.
 545 Both steps use down-sampled or high-resolution point sets. It is
 546 valuable to validate the approach by comparing with other available
 547 methods such as collision detections or joint detection with down-
 548 sampled model by interpolation.

549 Finally, our game-based design could be applied to different pur-
 550 poses, not only for participatory layout design but also for collect-
 551 ing data of user behaviors during design. Also application to other
 552 kinds of materials could be investigated.

553 Acknowledgments

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

559 References

560 CIMERMAN, B. 2000. Participatory design in architecture: can
 561 computers help? In *PDC*, 40–48.

562 GREENBERG, B., HITTNER, G., AND PERRY, K., 2010. Smart
 563 scrap. Accessed: 2016-09-30.

564 HAUERT, S., LO, J. H., NACHUM, O., WARREN, A. D., AND
 565 BHATIA, S. N. Crowdsourcing swarm control of nanobots for
 566 cancer applications.

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

570 KHOSHNEVIS, B. 2004. Automated construction by contour craft-
 571 ingrelated robotics and information technologies. *Automation in
 572 construction* 13, 1, 5–19.

573 KNAACK, U., CHUNG-KLATTE, S., AND HASSELBACH, R. 2012.
 574 *Prefabricated systems: Principles of construction*. Walter de
 575 Gruyter.

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

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

584 LUKKA, T. J., TOSSAVAINEN, T., KUJALA, J. V., AND RAIKO, T.
 585 2014. Zenrobotics recycler-robotic sorting using machine learn-
 586 ing. In *Proceedings of the International Conference on Sensor-
 587 Based Sorting (SBS)*.

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

590 MONIER, V., BIGNON, J. C., AND DUCHANOIS, G. 2013. Use of
 591 irregular wood components to design non-standard structures. In
 592 *Advanced Materials Research*, vol. 671, Trans Tech Publ, 2337–
 593 2343.

594 MUELLER, S., LOPES, P., AND BAUDISCH, P. 2012. Interac-
 595 tive construction: interactive fabrication of functional mechani-
 596 cal devices. In *Proceedings of the 25th annual ACM symposium
 597 on User interface software and technology*, ACM, New York,
 598 NY, USA, UIST ’12, 599–606.

599 OLIVER, P. 1997. *Encyclopedia of vernacular architecture of the
 600 world*. Cambridge University Press.

601 PYE, D. 1968. *The nature and art of workmanship*. Cambridge
 602 UP.

603 SCHINDLER, C., TAMKE, M., TABATABAI, A., BEREUTER, M.,
 604 AND YOSHIDA, H. 2014. Processing branches: Reactivating
 605 the performativity of natural wooden form with contemporary
 606 information technology. *International Journal of Architectural
 607 Computing* 12, 2, 101–115.

608 SUJAN, V., DUBOWSKY, S., OHKAMI, Y., ET AL. 2000. Design
 609 and implementation of a robot assisted crucible charging system.
 610 In *Robotics and Automation, 2000. Proceedings. ICRA’00. IEEE
 611 International Conference on*, vol. 2, IEEE, 1969–1975.

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

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

618 WESTON, R. 2003. *Materials, form and architecture*. Yale Uni-
 619 versity Press.

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