

Participatory Design and Fabrication Platform with Non-standard 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. 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 since primitive shelters, however, the use of them in modern buildings is limited, mainly due to the irregular properties of non-standard materials. In this paper, we take the diversity as playful inputs for design task, and present our game based design and fabrication platform for non-experts. Taking tree branches as case study of non-standard found materials, our online game *BranchConnect* enables users to design 2D networks of branches and realize the design with parametrically generated joineries. As the generated design is realizable with ordinal 3-axis CNC milling machines, each connection has a customized unique joinery adapted to the diverse branch shapes. The scoring system of the game guides users to design structurally sounding solutions with given branches. Together with low-cost mobile scanning devices, everyone 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 non-experts, and let them collect branches in a nearby forest, and design/fabricate a 2D fence by integrating multiple design solutions.

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-ACCD-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 characterized by 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].

Taking a look on traditional construction, or even fast back-warding to primitive shelters and huts, locally found resources were naturally used in these buildings. The idea of standard in buildings was not invented in the modern times but has been extensively applied as pyramids were built with standardized building blocks. The use of diverse non-standard materials can not compete with highly standardized materials and construction system, however, the uniqueness from their diverse shapes is the quality which standard materials can not serve. 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, however, special skills are typically required to handle these materials, mainly due to the irregular properties.

This paper aims to make the above-mentioned qualities of non-standard materials more accessible with digital technologies. We use locally found branches as non-standard resource found almost everywhere not only in country-side but also in urban environment such as parks and along streets. Most of these branches are annually pruned, and then chipped and burned with some cost. The size of branches (from 5 -30 cm in diameter) is too small for furniture or other structural applications. The diverse shape of branches pro-

vide adequate technical challenges in design and fabrication. Using high-precision but low-cost scanning devices and personal digital fabrication machines, even non-standard material properties can be analyzed and controlled.

The key technical difficulty with non-standard materials 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 non-standard materials 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. In this way, our method would be applicable to other kinds of non-standard materials. 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 experiment to design and fabricate an architectural element out of irregularly shaped branches with our workflow with humans-in-the-loop. 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. 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 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. The game system and developed import/export applications are currently limited to branches, however, the principle of human-in-the-loop for designing with non-standard materials is applicable to any other non-standard materials. 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 non-standard materials as design components.
- online platform to collect irregularly shaped materials and serving them to an online game as a design driver.
- an algorithm generating 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]. These parts are processed from highly standardized material, thus its digital fabrication process is “workmanship with certainty”; a batch process of reading Gcode 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 geometry, our work takes the diverse native branch shapes.

There are few works that take irregular shape of natural resources as aesthetic characteristics. Schindler and his colleagues used digitally scanned wood branches and used them for furniture and interior design elements [Schindler et al. 2013]. 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 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 non-standard resources, the design process is still dependent on a designer or architect who has experience with non-standard materials or has access to software which can compute the structural capability. 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, multiple designers can involve in design process, which could lower the cost of design fee with non-standard materials. Even non-trained lay people might be able to provide novel designs. For example, *Nano-Doc* took gamification as an approach to solve complicated problem. Firstly, it trains players on basic rules how nano-particles swarm through cancerous cells, and let them find novel treatments for real configuration of tumor cell [Hauert et al.].

3 Workflow

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

3.1 Digital Model Acquisition

There are various methods and software available for scanning 3D models and detecting objects. As for the scanning, we describe

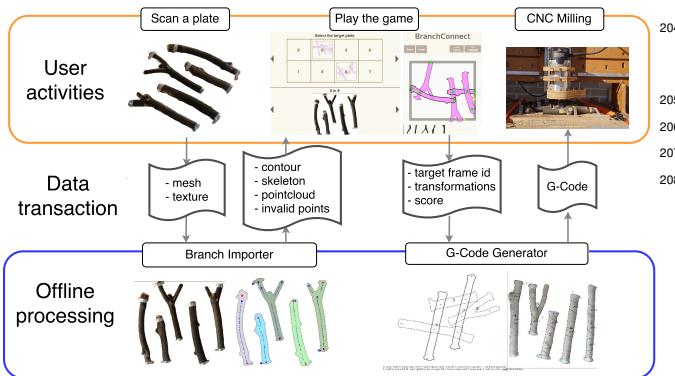


Figure 2: A pipeline from model acquisition to fabrication.

in the Experiment. Taking mesh model with colored texture, our *Branch Importer*, integrates necessary functions such as object detection, skeleton extraction, branch type classification, and fixture location detection.

After segmenting branches from a plate by height threshold, each branch is detected using *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 binary top view image of the mesh model (contours are filled with different colors and the background is white). If the point is inside of a contour, the middle point is counted in skeleton points. Metal fixture locations are also filtered out due as they have bright reflections on original images, however, we also double check with simple mouse-clicks to ensure these invalid points. After extracting valid middle points, connectivity of skeletons is analyzed by angle of three adjacent skeleton points. If the angle stays within a threshold bound, a point is counted in a sub-branch, otherwise, new sub-branch is created. Evaluating the number of sub-branches, the branch is morphologically classified. Most branches are categorized in three shapes: *I*, *V*, and *Y* shapes. *I* shape has a straight continuous polyline, *V* has an inflection point, and *Y* has a splitting point. The acquired information is stored in a cloud database.

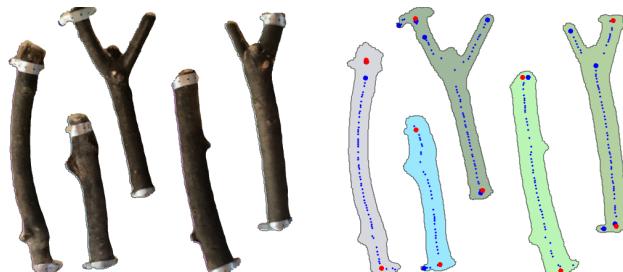


Figure 3: An overview of Branch Importer. Left: a top ortho-view image of textured mesh model. Right: Extracted skeletons (blue dots). The beginning of skeletons is shown bigger dots. Red dots are invalid points.

3.2 BranchConnect: The Game

The game objective is to collect valid compositions of branches which structurally sounding and possible to fabricate with ordinal 3 axis CNC milling machines. The workflow of the game is illustrated in Figure 4

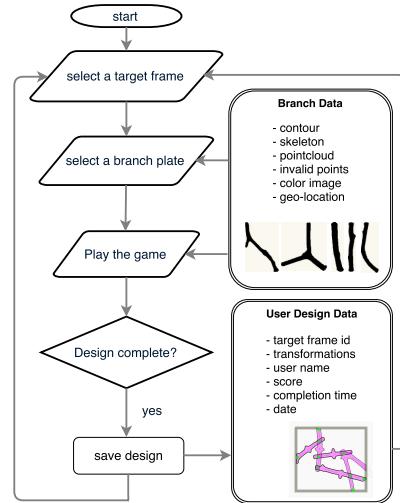


Figure 4: The workflow of BranchConnect. Branch data and user design data are stored on cloud database.

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 5). 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 5). 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 mirror. 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 by branches. For higher score, the user can modify the design after the completion. After completing the modification, the design is submitted and sent to *G-Code Generator*.

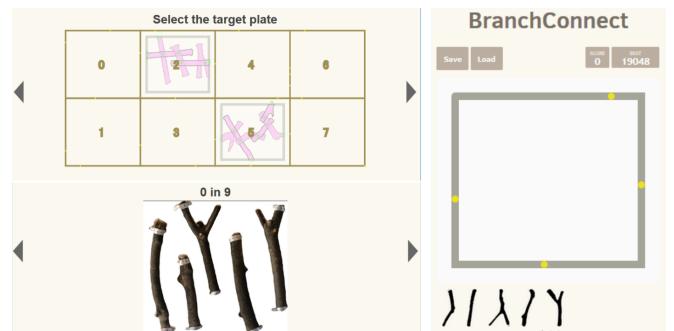


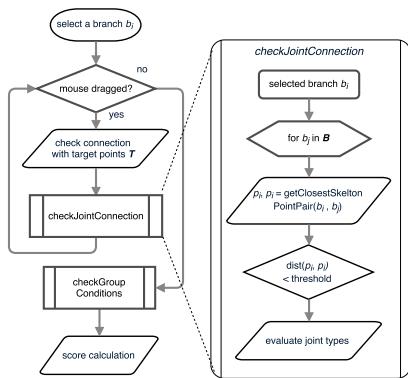
Figure 5: Left: the selection interface for target frames (top) and branch panels (bottom). Right: the start interface of the game.

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

226 **System Overview**

227 There are many collision detection libraries available, however, our
 228 game needs to detect intersected branch pairs, thus surface contact
 229 detection is overkill for our application. Also most branches come
 230 with free-form concave shapes, thus further geometric preparation
 231 such as convex decomposition is necessary for using these libraries.
 232 For fast and robust intersection detection, our system extensively
 233 uses skeletons of branches. Hubbard and Philip developed colli-
 234 sion detection by representing object with hierarchical 3D spheres
 235 aligned on skeletons [Hubbard 1996]. Our system takes similar
 236 approach but more limited in 2D and intersection detection only. In
 237 broader phase, simplified skeletons are used to find the pair of clos-
 238 est skeleton points between two branches. After finding the pair,
 239 skeletons with higher resolutions are used.

240 A joint is created when an intersecting pair is detected, and the
 241 pair forms a group. The group is used for evaluating connections
 242 between target points. The game is completed once all the target
 243 points are connected by a group of branches. The conditions of
 244 joint and group are indicated with simple color-code. Once the user
 245 finishes positioning, score is updated with weighted sum of param-
 246 eters. Together with the color-code, the score update guides the user
 247 to form a valid design. An overview is illustrated in Figure 6.

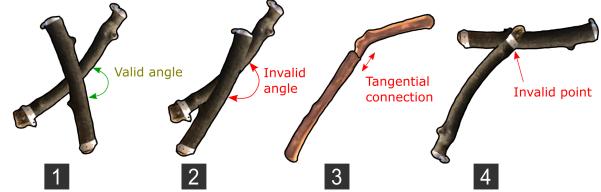


248 **Figure 6:** Left: an overview of the game system. Right: joint condi-
 249 tion checking process.

250 **Joint Condition**

251 Joint is the essential entity not only in the game but also the entire
 252 project including the fabrication process. The closest skeleton point
 253 pair is obtained We accept crossing joints only because they are
 254 structurally stable, relatively simple to fabricate, and creates diverse
 255 designs. To ensure the structural performance, we set a connection
 256 angle within a range from 45 to 135 degrees. Joints close to metal
 257 fixtures are also counted as invalid. Valid and invalid joints are
 258 displayed with green and red respectively. The color-code feedback
 259 is also given when target points on a frame is connected with a
 260 branch. Figure 7 illustrates valid and invalid joint conditions.

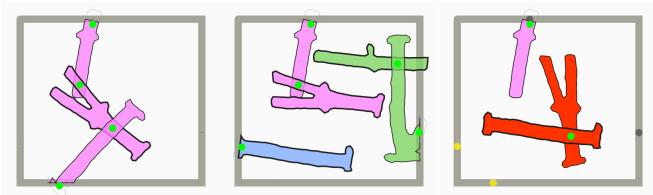
261 To describe the process, we let \mathcal{B} denotes a set of all the available
 262 branches, and each branch as $b_i \in \mathcal{B}$. While the process checks
 263 joint condition through all the branches \mathcal{B} , each detected joint is
 264 stored in each branch b_i , categorized in different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



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

272 **Group Condition**

273 After checking joint conditions of all the pairs of branches, the sys-
 274 tem checks the number of groups as well as connectivity with the
 275 target points on a frame. If a group is not connected to any target
 276 point nor other groups, the group is islanded and structurally in-
 277 valid. While a user is positioning a branch by dragging or rotating,
 278 groups are continuously calculated and indicated by simple color
 279 (Figure 8).



280 **Figure 8:** Left: valid group with two target points connected. Mid-
 281 dle: valid but three groups. Right: invalid due to the island situ-
 282 ation.

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

290 **Score Calculation**

291 We denote the numbers of valid and invalid joints on each branch
 292 $b_i \in \mathcal{B}$ as $N(valid_i)$, $N(invalid_i)$ respectively, the number of
 293 groups as $N(\mathcal{G})$, the number of islanded groups as $N(g_{islanded} \in \mathcal{G})$,
 294 the number of bridged target points as $N(t_{bridged,i}) \in \mathcal{T}$. The score
 295 is weighted sum of these joint and group conditions (see Eq 1).

296 **TODO: notaion should be improved!**

$$\begin{aligned}
 Score = & w_1 \sum_1^{N(\mathcal{B})} N(valid_i) + w_2 \sum_1^{N(\mathcal{B})} N(invalid_i) \\
 & + w_3 \sum_1^{N(\mathcal{G})} g_{islanded} + w_4 \sum_1^{N(\mathcal{T})} t_{bridged} \\
 \text{s.t. } & w_j \geq 0 \forall j \in 1, \dots, 4
 \end{aligned} \tag{1}$$

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  $g_{new}$ 

```

309
310
311
312 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!**

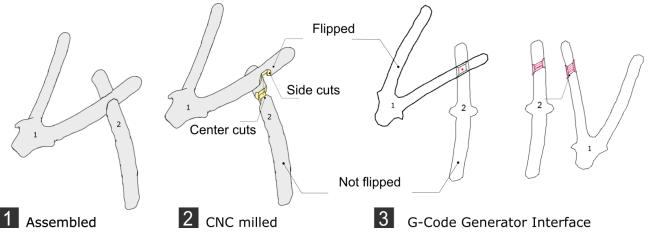


Figure 10: An example of intersected pair: 1. an assembled pair of branches. 2. branches after the center and side cuts are milled left: a composition defined by a user right: the original orientations of branches with generated milling paths with red color.

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

4 Experiment

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

327 The goal for the participants is to contribute to an ongoing design
328 and fabrication process of screen wall (2m by 0.9m) consisting of
329 8 rectangles. Six frames were already designed and built, thus the
330 rest two frames were prioritized for them to design and fabricate in
331 this workshop.

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

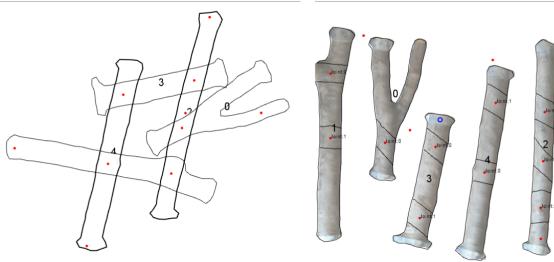


Figure 9: Interface of G-Code Generator. Users can tweak the design on the left side and immediately see the updated joints and milling paths on the right. **TODO:** can this image be integrated with the other one?

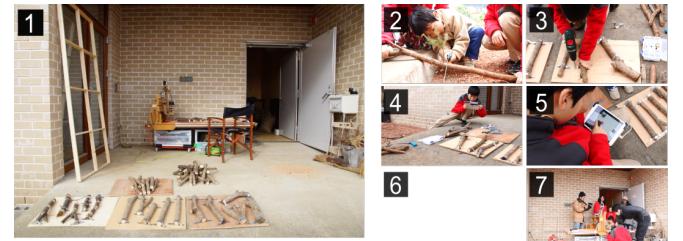


Figure 11: An overview of the workshop. 1.the overview of the space. 2.cutting 3.preparation 4.scanning 5.retouching 6.playing the game 7.CNC milling.

295 Some fabrication factors such as mirror and invalid points are al-
296 ready considered by *Branch Importer* and the game system of
297 *BranchConnect*. In this section, we describe the process of joinery
298 generation. Each joinery geometry is different but has same topo-
299 logy: two plane surfaces on the sides of branches and one plane top
300 surface. The geometry creates rigid connection with the irregularly
301 shaped sections of branches. Similar to the joint searching pro-
302 cess with skeletons, the *G-Code Generator* searches a set of four
303 closest points from high-resolution contours, expecting that every
304 intersected contour has four curves. After finding the four closest
305 points, it trims two curves from each contour of branch. (two from
306 intersecting branch and two from intersected branch) at each joint.
307 The trimmed contours are transformed to the original scanned ori-
308 entation and used for generating milling paths. Two curves from

338 4.1 Setup

339 System and Hardware Setup

340 We used two iPad minis with iSense scanners attached for scanning branches, and a 3 axis CNC milling router with a 6 mm diameter
 341 milling bit. We used a laptop PC (Lenovo w240 with intel core i7) for running *Branch Importer* and *G-Code generator*, as
 342 well as operating the milling machine. The scan area of iSense is
 343 500mm x 500mm, and the milling machine's stroke along z-axis
 344 is 70mm, which provide geometric constraints for available branch
 345 sizes. *BranchConnect* was hosted at *Heroku* cloud server², and we
 346 used *MongoDB*³ as a cloud database.

349 Preparation of Branches

350 The participants were asked to collect branches with 2 - 10 cm in
 351 diameter. The lower bound of the diameter is for milling process
 352 would not destroy them, and the upper bound is for the limit of
 353 z-stroke of the CNC router. The collected branches are cut
 354 in arbitrary lengths, not longer than 500 mm due to the limit of
 355 scanning area.

356
 357 As our game system and fabrication process take 3D branch shapes
 358 as 2D contours with height map, these constraints work positive for
 359 the system automatically filtering out branches with large 3D twists.
 360 We ask participants to scan with iPad + iSense and prepare feasible
 361 mesh model by themselves. After obtaining mesh models, tutors
 362 import models from iPads to a laptop and upload them to database
 363 by *Branch Importer*.

364 User Experiences with the System

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

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

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

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/>

391 4.2 Results

392 The entire workshop took 4.6 hours to complete the whole process,
 393 including introduction, moving, and pauses. Table 4.2 shows dura-
 394 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
Miscellaneous	1.0	21.7
In total	4.6	100

396 Collecting Branches

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

406 Model Acquisition

407 iSense 3D scanners come with an intuitive software for scanning
 408 and modifying models. After we gave an instruction, most of par-
 409 ticipants practiced several scans and successfully scanned models
 410 without any problem.

411 Each scanning and re-touching took 2-3 minutes, and 30 seconds
 412 for generating data by *Branch Importer*. Including the prepared
 413 panels previously, we scanned 15 plates in total, 75 branches, and
 414 35.3m of total length including sub-branches. **TODO: check the**
 415 **length again!** We got 40 I shaped branches, 19 V shaped branches,
 416 and 16 Y shaped branches. The result is shown in Figure 12.

417 Design with the Game

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

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

426 Although there is no time limit in the game system, we set 30 min-
 427 utes for playing the game, and 8 solutions were given by partici-
 428 pants. xxx frames were completed per participants and xxx partici-
 429 pants completed the whole eight target frames. The average score is
 430 xxx, and average playing duration was xxx to complete each target
 431 frame.

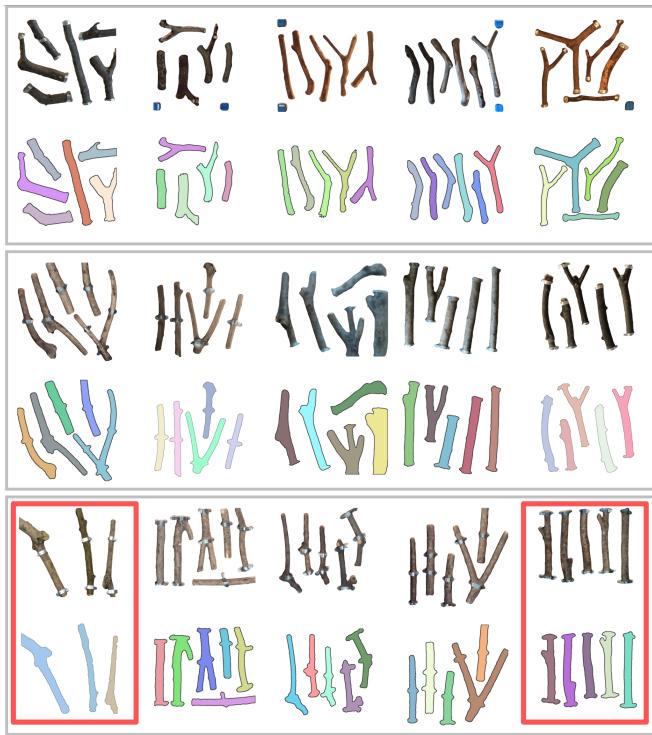


Figure 12: An overview of all the 15 scanned plates for the workshop. Top raw of each set shows ortho-top views of scanned mesh models, and the bottom raw is the recognized branches assigned random colors. The red-lined rectangles indicate the plates built by participants in the workshop.

Overall Design Consensus **TODO: this section might be removed**

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

Fabrication

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

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

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

as reasons such as,

- 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

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

4.3 Summary

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

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

A noteworthy fact was that one participant with four years old failed to complete the game, but insisted on accepting his design to be fabricated by the CNC router. We took these requests as their commitment to the entire workflow, as we observed that all the participants insisted on continuing the fabrication process.

5 Conclusion

Summary

In this paper, we presented a workflow to design and fabricate with branches which are not large enough for producing standardized building components. Our workflow was validated with case-study with lower-aged participants without design and fabrication experiences. Our online platform to store geometries of scanned branches and compositions of branches was reachable by people with various backgrounds. Our light-weight branch joint detection algorithm was also validated by running on the browser game, contributed to the accessibility of the presented workflow. Together with the accessibility, the intuitive interface was simple enough for non-expert users. We successfully built a network of branches with rigid joints 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 crafting-related 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.
- LAMBERT, M., AND KENNEDY, P. 2012. Using artificial intelligence to build with unprocessed rock. In *Key Engineering Materials*, vol. 517, Trans Tech Publ, 939–945.
- LENNON, M. 2005. Recycling construction and demolition wastes.
- LERER, A., GROSS, S., AND FERGUS, R. 2016. Learning physical intuition of block towers by example. *arXiv preprint arXiv:1603.01312*.
- 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.
- SANDY, T., GIFTTHALER, M., DÖRFLER, K., KOHLER, M., AND BUCHLI, J. 2016. Autonomous repositioning and localization of an in situ fabricator.
- SCHINDLER, C., TAMKE, M., TABATABAI, A., AND BEREUTER, M. 2013. Serial branches: Strategies to activate the performative capacity of naturally grown wooden form with contemporary information technology. In *eCAADe 2013: Computation and Performance—Proceedings of the 31st International Conference on Education and research in Computer Aided Architectural Design in Europe, Delft, The Netherlands, September 18–20, 2013*, Faculty of Architecture, Delft University of Technology; eCAADe (Education and research in Computer Aided Architectural Design in Europe).
- 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.
- 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.

625 YOSHIDA, H., IGARASHI, T., OBUCHI, Y., TAKAMI, Y., SATO,
626 J., ARAKI, M., MIKI, M., NAGATA, K., SAKAI, K., AND
627 IGARASHI, S. 2015. Architecture-scale human-assisted addi-
628 tive manufacturing. *ACM Trans. Graph.* 34, 4 (July), 88:1–88:8.