Real Time Scanning-Modeling System for Architecture Design and Construction

Ye Zhang[*], Kun Zhang, Kaidi Chen, Zhen Xu

Department of Architecture, Tianjin University, Tianjin, China

Abstract

The disconnection between architectural form and materiality has become an important issue in recent years. Architectural form is mainly decided by the designer, while material data is often treated as an afterthought which doesn't factor in decision-making directly. This study proposes a new, real-time scanning-modeling system for computational design and autonomous robotic construction. By using cameras to scan the raw materials, this system would get related data and build 3D models in real time. These data would be used by a computer to calculate rational outcomes and help a robot make decisions about its construction paths and methods. The result of an application pavilion shows that data of raw materials, architectural design, and robotic construction can be integrated into a digital chain. The method and gain of the material-oriented design approach are discussed and future research on using different source materials is laid out.

1. Introduction

The disconnection between form and material in the architecture field has become a serious problem. With the popularity of 3D modeling software such as SketchUp, Rhino, and Grasshopper, architects are prone to first develop form on a computer screen and only afterwards associate material designations to the predefined geometry model [1]. In improper citing, the digital form is freely rendered with any material bitmap or texture without regard to a material's rational use or practicality [2], and risking that material information cannot be timely and effectively fed back to the design and construction stages.

The separation of material and form or the linear progression from the design intention to materialization has caused some problems in recent decades. Firstly, the process of fabricating source material into a variety of sizes and shapes consumes significant time, while a large quantity of material waste is produced. Secondly, human-designed rules and solutions are prone to be limited by empirical knowledge [3], especially in projects with irregular surfaces and complex geometries. Ignoring source material's properties makes the design and fabrication process more passive, inefficient, imprecise, and expensive. Finally, it hinders expressing building's regional and cultural characteristics as architectural materials with different personalities in different regions are mass produced through the same processing method.

Advances in contemporary digital scanning and bespoke robotic technologies [4] enable material information to be simulated, calculated, and well organized. As a result, it is an alternative approach for architecture. In this study, a new, real-time scanning-modeling system for obtaining material information and incorporating the data into a continuous digital chain of computational design and robotic construction are introduced. As Fig. 1 shows, traditional designs are determined subjectively by the architects; while a new chain with the scanning-modeling system enables an alternative design approach derived objectively by the data of source material.

---

[*] Corresponding author. E-mail address: yaapp2012@gmail.com
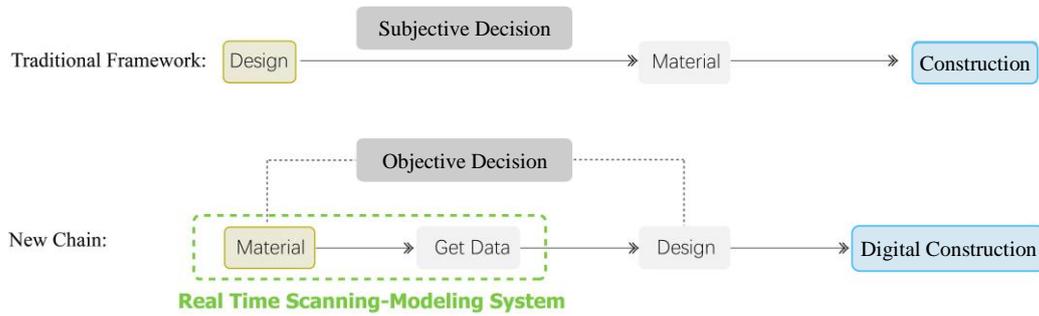
Tel.: +86-15900241936

Fig. 1 Comparison of two modes

The scanning-modeling system is regarded as the sensing end of this digital chain. It reads data of raw materials including their sizes and shapes without cumbersome supporting equipment and builds a visual model in real-time. After collecting and visualizing the data, the calculation end of the chain would generate architectural geometry based on material data, site conditions and structural performance. Finally, at the action end of the chain, a robot arm would be used to fabricate the design. According to the data obtained during the previous stages, the robot would find an assembly location autonomously without having to be programmed in advance. In sum, this study creates a convenient and rapid material information acquisition system that benefits architectural design and digital construction.

2. Literature Review and Innovation Points

This study provides a new scanning-modeling system and examines its application in architectural programs. The research contents include autonomous perception, computational generation, and robotic actuation. Despite a great deal of research on these three topics over the past several decades, there has been very little research intended to integrate them into the field of architecture. Previous studies and new development in measuring and modeling, computational design, and robotic construction are described as follows.

2.1. Measuring and modeling

Table 1 Measuring and modeling methods

	Name	Principle	Speed	Accuracy	Cost	Stability	Limitation
Active Method	Moiré Fringes	Diffraction and interference of light	++	+++	+	+	only available for objects with regular texture
	Time-of-Flight (ToF)	time taken for the wave to bounce back to the emitter	++	--	--	-	Low resolution and easy to be disturbed
	Structured Lighting	Encoding and decoding of light	+	++	+	++	Projection equipment required
	Triangulation	Solve trigonometric equation	+	-	+	+	Data sparseness
Passive Method	Shape From Texture (SFT)	Perspective distortion calculation	+	+	+	-	Surface texture information required
	Shape From Shading (SFS)	Ray calculation	-	-	+	-	Light parameters required
	Multi-View Stereo (MVS）	Parallax principle	--	--	+	+	Image error is big
	Silhouette	Visible shell approximation	+++	--	++	++	Visible shell only

Existing 3D reconstruction methods can be divided into two categories: active and passive. The active methods mainly include Moire Fringe, Time-of-Flight (ToF), and Structured Lighting and Triangulation; while the passive methods mainly include Shape From Texture (SFT), Shape From Shading (SFS), Multi-View Stereo (MVS), and Silhouette. These methods use mathematical and physical knowledge to rebuild objects. Among them, Moire Fringe used the interference of wave to

deduce the shape; SFT, SFS and Structured Lighting mainly use perspective principle; Triangulation, MVS and Silhouette use the methods of deformations of triangle positioning. The characteristics of each method are presented in Table 1 [5-9]. "+" means the performance is strong while "-" means it is weak.

In the field of architecture, the precision of measuring and modeling required is not as strict as other fields. The demands of speed, cost, and stability are higher. Therefore, this study is established on the basis of the silhouette method, which develops to improve the speed and efficiency of getting information on material size, and to make it easier to operate as the computing Moreover, the information communication mechanism are in a visual interface.

2.2. Computational design

Different from design precedents which rely on intuition and experience to solve design problems, computational design encodes design decisions using a computer language; consequently, the design expression is changed from geometry to logic [10]. The computational design is excellent in dealing with complexity, coming up with many options in a short time, and optimizing the solution. Although the computational design is still a relatively young and evolving field [11], some research institutions and researchers have made progress in theoretical development, methodologies, and strategies [12-14].

Based on the methodologies and strategies of computational design, this study presents a pavilion which is generated from material information. The major dimensions of computational design in this study include:

(1) Information transformation: this pavilion makes use of natural branches containing three forks. The length of each fork and the angles between the forks are read by a scanning system and the information is transformed into Grasshopper in real time by a plug-in developed by this research.

(2) Parameter setting: considering the site conditions and the pavilion's functional uses, the parameters are the location of the anchor points, the camber of the surface, the fractal geometry, and the number of forks at a node.

(3) 3D modeling and visualization: the computation of the design and fabrication is conducted in Rhino3D which allows designers to test and simulate in a visual environment.

(4) Processing with data and algorithms: The rules and standards to inform the design direction comprising waste less material, similarity to the target surface, and structural performance.

2.3. Robotic construction

The use of robots in architectural construction can be traced back to the 1980s [15]. At that time, robotic systems were mainly used in Japan, but the efficiency didn't prove satisfactory and hence received fading research attention. However, with the development of digital tools and manufacturing technologies, robotics in architecture has been a burgeoning research topic since 2010. Prior studies have sought to explore and expand the possibilities of robotic fabrication, robotic assembly, and even human-robot interactions on site. However, an important limitation of past research is the material sensing and real-time feedback. This study seeks to fill that gap by emphasizing robotic eye-hand coordination and to break with the typical robot movement pattern which is controlled by humans and not by the robot itself.

3. Real Time Scanning-Modelling System

3.1. Motivation and framework

The primary purpose of the scanning-modeling system is measuring architectural materials on-site automatically and gets their three-dimensional models in real time. The procedure is shown as Fig. 2. For each material component, the images from

different perspectives will be scanned by a camera group. An independent program specially written by this study will capture the effective information of the material image and calculate feature points using an open source computer vision library. In order to ensure the efficiency of operational and data transmission, the program is simplified to transmit data of the quantity of controllable feature points with high information concentration. Then, a pre-programmed plug-in in Rhino/Grasshopper will receive data from the scanning program by memory mapping and analyzing outlines from different perspectives. Finally, a three-dimensional model of the corresponding material component will be generated within 1 second for subsequent design development.
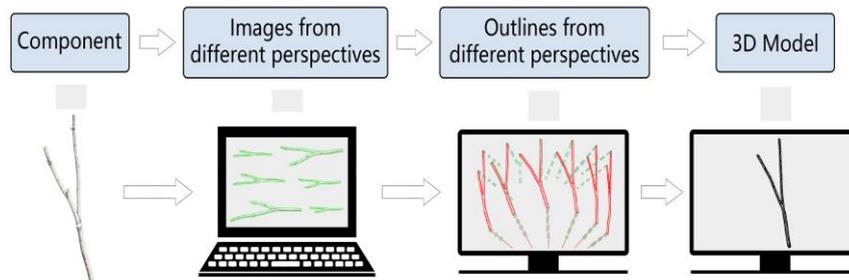

Fig. 2 Scanning-modelling system

3.2. Method

This research takes natural branches as an example material to explain how the scanning-modeling system works. The system extracts the contours of the physical branch and reconstructs it digitally. It has two working paths: a software workflow and a hardware workflow. The software workflow is virtual and programmable, while hardware workflow is realistic and transitive. The data circulate and transmit in the software workflow, participating in complex interactions and transformation operations, before finally outputting a concise digital model of the branch. The hardware workflow is very simple, consisting only of material components, sensors (cameras) and a computer. Six cameras can complete all of the image acquisition work while the computer can realize image recognition, data processing and 3D model reconstruction.

3.3. Hardware workflow

The branches (source material), camera group, and computer constitute the hardware workflow of the system. In the laboratory, a 6m-long custom frame to fix the cameras is built, and the background with black curtains to extract information effectively is covered. The conditions and settings for each hardware element are as follows:

(1) Branches to be tested:

This study used natural branches which had minor branches, leaves and twigs pruned away. Their lengths range from 75 cm to 245 cm and their radius range from 2cm to 10cm.

(2) Number and location of cameras:

The branch's radius is set as R, the length of the frame holding the cameras as L and the camera angle as α. An equation can be calculated as:

$$L = 2R/sin(\alpha/2) \tag{1}$$

The number of pixels in the corresponding direction of the camera is set as n, the resolution value of the reconstructed model is γ. The calculation is based on the 1st formula:

$$\gamma = 2Rcos(\alpha/2)/n = Lsin\alpha/2n \tag{2}$$

The inherent error of this system is not uniform. It is affected by two factors as shown in Fig. 3: one is the angle between the object and the camera, and the other is the normal vector of the surface point with respect to the plane of the camera group. To establish this error, this study sets the angle between two adjacent cameras as δ and the inherent error as e. An equation can be computed as:
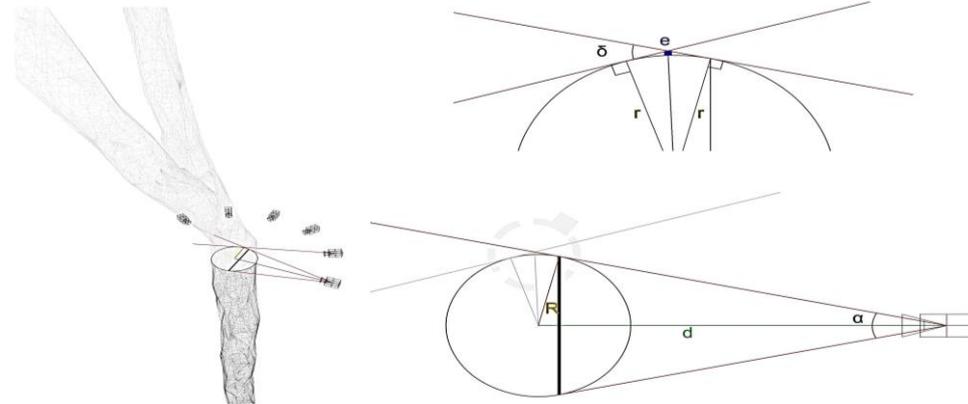
$$e=r[csc(\delta/2)-1] \tag{3}$$

Fig. 3 Calculations

In order for γ> e and considering feasibility, convenience, and accuracy, this research finally takes L = 600cm, α = 0.3π, δ = 0.667π, which means 6 cameras are fixed to a 600 cm-long frame to catch images of the branch.

3.4. Software workflow

The software workflow is the core of this system. It includes three steps, as shown in Fig. 4:

(1)   Generating a description file of calibration cameras.

(2)   An external program which extracts the image of the branch and calculates related data for visual presentation.

(3)   A pre-programmed module in Rhino receives the data and converts it into a digital branch model in real time.

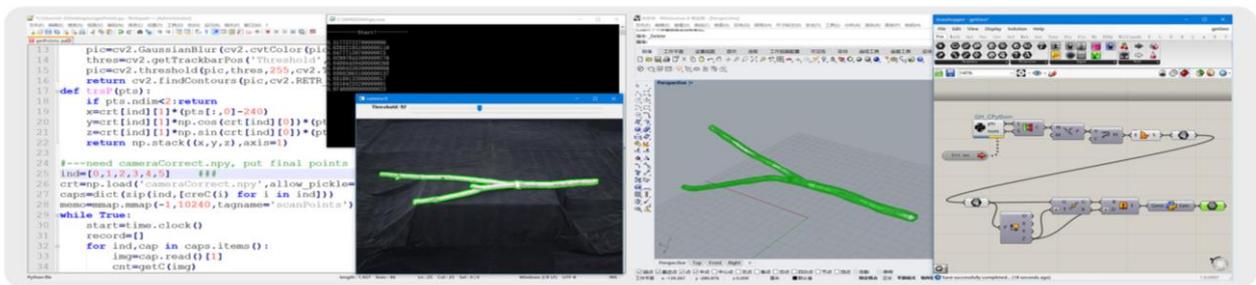The following sections explain the three steps in more detail.

Fig. 4 Software workflow

The first step is to develop a camera calibration file "CameraCorrect.py". The purpose of this step is to determine the mapping relationship between contour pixels and spatial points in the modeling software. The calibration is based on the perspective principle, it does not require a complete camera internal parameter matrix, only a scaling parameter and rotation parameter are taken. Because of the USB broadband limitation, an additional module was added in the Python script in order to read information from multiple cameras simultaneously. There are five commands "z", "x", "f", "s" and "q" in the script: "z" and "x" to change the camera, "f" to refresh, "s" to display the recorded data, "q" to save the data and exit. After the relevant parameters of the calibration tool are entered, an auxiliary script automatically calculates the final calibration parameters and generates this calibration file. This step only runs once in the whole workflow and is not repeated when a branch is scanned.

The second step is to analyze the branch images and generate both a "Points.py" file which contains image information and a "getGeo.gh" file which is used in the modeling software. First, through the Gauss filter process and the binary image process, it is able to extract smooth contours of a branch from different perspectives. Then, the number of pixels selected in the contour is adjusted by a tolerance parameter and the three-dimensional coordinates of all the selected points are calculated. Finally, the three-dimensional array is transferred into Grasshopper through shared memory and the spatial position of the points is inversely mapped in Rhino according to the former calibration file.

The third step in the software workflow is to generate a visual model which runs simultaneously with the second step. A preset program receives array data and reorganizes them into projections from different views in order. By the algorithm of the Boolean intersection, all the information work together to generate a digital branch model representative of the scanned physical branch.

3.5. Results and discussions

The system is characterized by flexibility, reliability, and processing speed. Because of the modular organization of the workflow, the perception of material information is expandable. The researchers can increase the number of cameras to improve the accuracy of the system; in addition, expand the type of sensors to capture and analyze other qualities of the source material. Flexibility is also reflected in the diversity of operating environments and source materials. Although the former branch experiment is carried out in a laboratory, it can be performed in other places including outdoor environments by simply changing the fixing points and corresponding values in the equations. Similarly, the system can scan other building materials (source materials). Additionally, the system has a high tolerance for faults, built-in redundancy; therefore, it is very reliable. Even if some of the sensors are broken, other sensors can still work as usual, so the program can run normally despite the reduced accuracy. Last but not the least, this system runs very fast. Usually, image processing is extremely CPU intensive, but this system basically eliminates this problem with appropriate abstraction and sufficient optimization. Taking the branch project as an example, even in ordinary notebook configuration, a refresh rate of 1 second per time can be achieved. This efficiency not only saves time, but also makes the system more practical to use.

4. Application Example

Using the above-mentioned real time scanning-modelling system, a pavilion is built and the application of the system in new architecture model is shown. This pavilion uses abandoned "Y" shaped branches as source materials. By the algorithms, the computer can make independent decisions and generate a design based on the quantity and characteristics of the existing materials [16]. Then, a robotic arm is used to complete the construction process.

4.1. Design of the pavilion

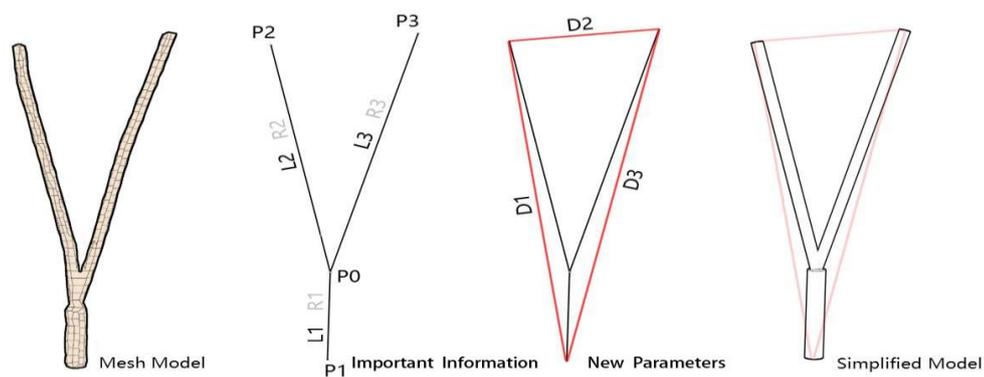### 4.1.1. Selection and organization of the branches

Fig. 5 Information abstraction

With the Scanning-Modelling System, the researchers develop digital models of all wooden branches that can be used for the pavilion. Next, the researchers distill the information which is useful to design and discard data that does not affect the design such as slight unevenness on the branch surface and the bending amplitude of each fork, etc. The useful parameters are three-dimensional coordinates of the four points P0, P1, P2 and P3 on the branch. The length between every two points and the length and thickness of each fork are shown in Fig. 5. Because all tree branches are from the same species of the ash tree, their density is considered the same.
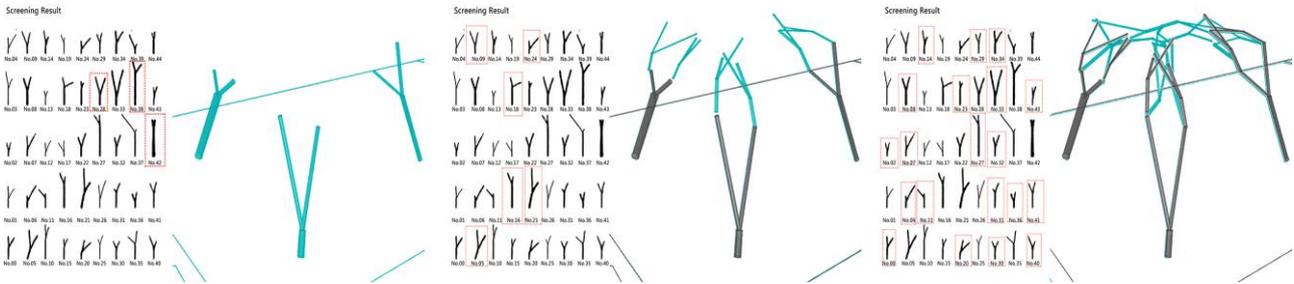

Fig. 6 Computational generation process

According to the environmental conditions and functional requirements, the approximate dimensions of the pavilion are established. The location of the ground points and the geometry logic are seen as a preliminary step, in addition to stipulating that each vertex must have three forks jointing together, as shown in Fig. 6. In the procedure of component selection and organization, the logic is set as follows:

(1) All branches are sorted by volume and the three largest load-bearing branches are selected.

(2) For any two branches B and C which are connected to the same Branch A, if they can intersect, their trajectory of intersection is an ellipse and takes the position with the minimum distance from the central axis of branch A to ensure the overall stability.

(3) If Branch B and Branch C cannot intersect, they are cut shorter to make the intersection possible. According to the information list of all the branches, the computer can quickly calculate the cutting lengths of different options and the branches spatial positions under all possibilities. Corresponding pavilion forms are also simulated and all possibilities are sorted by how much material is wasted.

(4) The top five options are chosen, then "millipedes" and other analysis tools are used to test and optimize their structural performance. Combining the amount of material waste factor, structural properties, and formal aesthetics, the final pavilion design option is selected, as shown in Fig. 7.
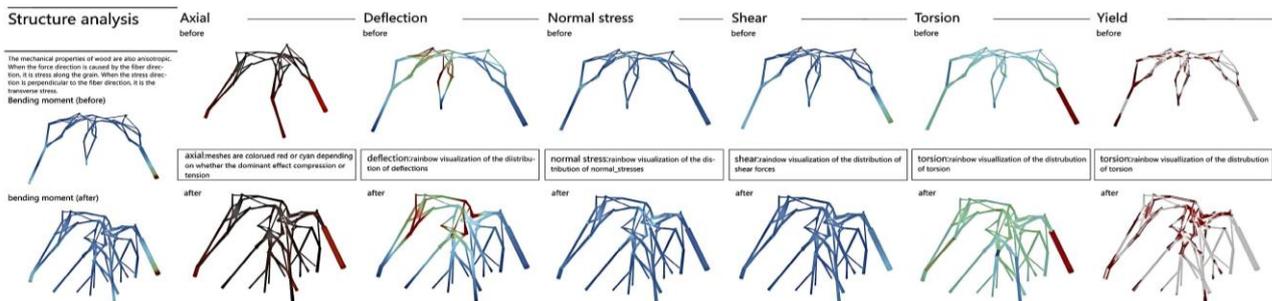

Fig. 7 Structure performance analysis

4.1.2. Detailed design

After the design program is generated and optimized, its details are refined. The first is the node. Every node in this pavilion is different and because all branches have their own angles, the nodes are customized rather than standardized. Traditional rope-wood nodes can adjust the angle and are easy to make; however, they are not strong enough. CNC-customized

tenon-and-mortise nodes are stable, but their fault tolerance ability is weak and the manufacturing takes detailed time. Ultimately, a new type of node with universal joints, three-way pipes and jackscrews are designed. The new node is strong, cheap and easy to make, as shown in Fig. 8.
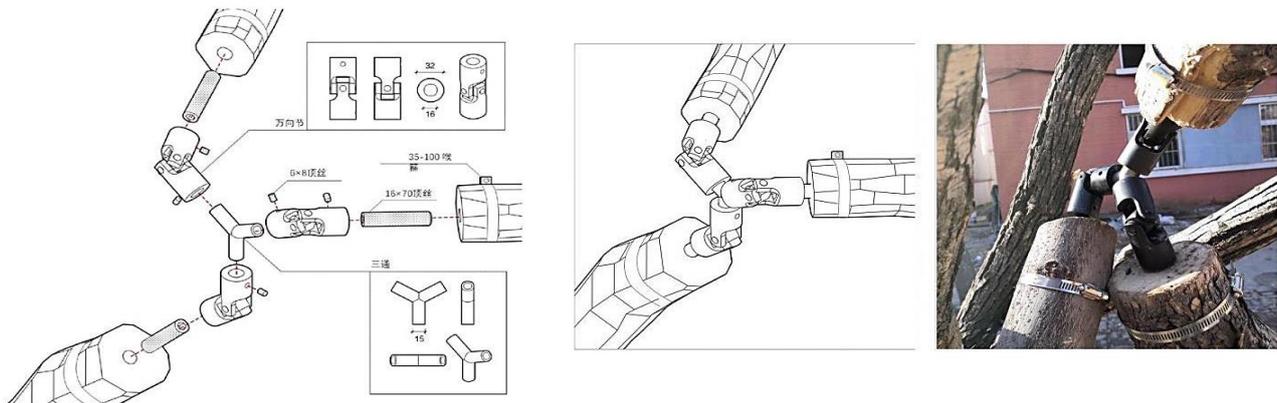

Fig. 8 Joints

For the base, a cast-in-place concrete foundation and embedded bolt method are tried, but this was eliminated due to their insufficient strength and site environmental limitations. Instead, the project uses metal bearing plate and ground pins to establish the base, as shown in Fig. 9.

Fig. 9 Base

4.2. Robotic construction

In the robotic construction stage, the motion of the robot arm and the design of the tool end should be carefully considered. By writing the relevant code in Grasshopper and KukaPrc, the robotic arm can automatically control its grabbing position and motion movement path according to the generated design. The robotic arm determines the final position of each branch quickly and accurately and then fixes the joint by cooperating with human workers or another robot. If there is any problem during the construction process, the data will be fed back into the digital model for real-time adjustment.

In terms of the tool end of the robotic arm, it has to be strong enough to grasp the branch and ensure that it doesn't twist. An electronically controlled claw is first applied. With the AFMotor library in Arduino, the opening and closing state, running speed and rotation direction of the gripper can be controlled in real time. However, claws have limitations: its opening and closing range is only 0-90mm and its maximum clamping force is only 16 Newton. Thus, it is only suitable for grabbing light and small branches.

In order to improve the ability to grab, the pneumatic parallel open and close claw are selected and modified. The size between the parallel plates could be adjusted to match the diameter of the branch. The contact surface of the plates is coated with a special rubber material to increase the coefficient of friction; consequently, the branch wouldn't twist. Features of this pneumatic claw are:

(1) Wide opening and closing range.

(2) Greater grabbling ability.

(3) Suitable for branches with different sizes and weights.

(4) It can be deployed in Grasshopper, so the robot arm can perform the entire step independently (Fig. 10).
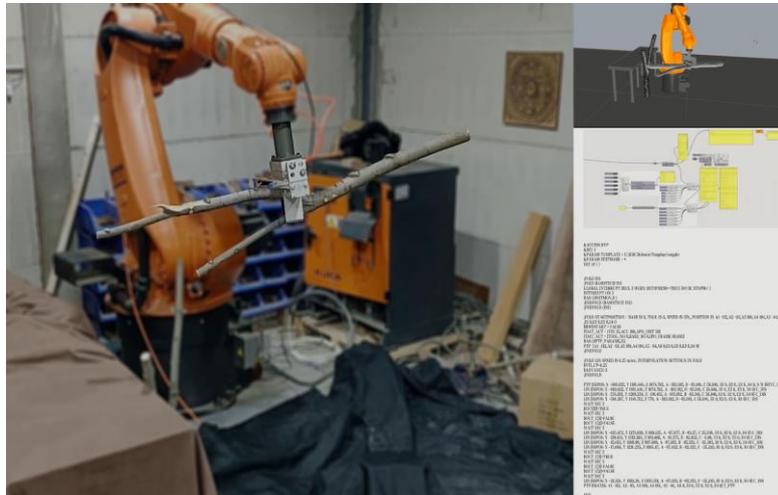

Fig. 10 Robotic construction

5. Conclusion and Future Work

This research succeeds in developing a real time scanning-modelling system to perceive and analyze material information for guiding the architectural design and robotic construction (Figs. 11-12). The material oriented "eye-brain-hand collaboration" chain saves construction materials, greatly reduces the cost of component fabrication, gives unprecedented flexibility to geometry, optimizes structural performance and improves construction work.
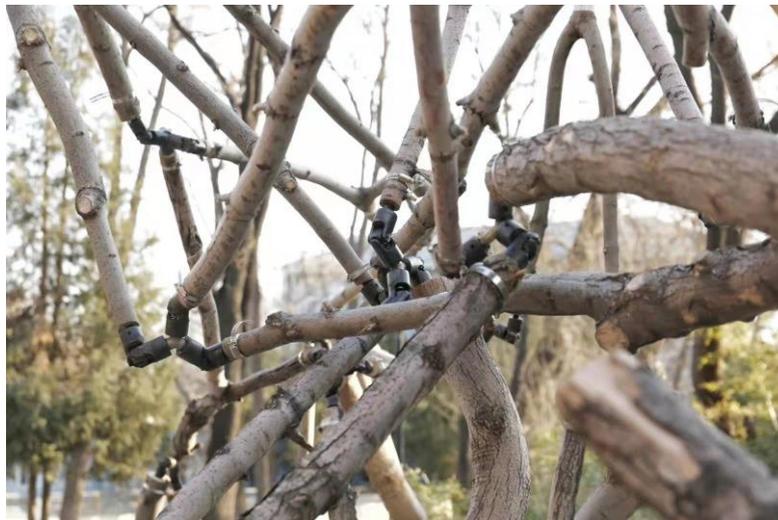

Fig. 11 Pavilion

In this study, there is still some room for improvement. First of all, the environment perception aspect can be enriched. More types of sensors can be installed and the analysis of algorithms can be optimized; therefore, the system can perceive and utilize other kinds of information such as material strength and toughness. Secondly, the computational design part can be further optimized and treated as agent-based simulation and generation. The researchers can develop algorithms that let the agent system provide better outcomes. User interfaces can also be developed. Finally, the experimental pavilion relies exclusively on a Kuka robotic arm. Bespoke robotic arms can be modified and the interoperability of different robotic arms can be developed.

Fig. 12 Pavilion

In future research, the theory and method of this study are expected to be used for other materials in real scale architecture projects. The scripts and scanning-modeling methods can be partially replaced and used for materials with various shapes. For larger scale construction, the locations of sensors could be recalculated using the formula put forward in this paper. The researchers envision that success in digital manufacturing with globally abundant and locally available materials may impact the design approach by considering materials' personalities.

Acknowledgement

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] K. Wu and A. Kilian, "Design natural wood log structures with stochastic assembly and deep learning," Robotic fabrication in architecture, art and design, Springer Press, August 2018, pp. 16-30.

[2] A. Picon, Architecture and the virtual: towards a new materiality, Wissenschaftliche Zeitschrift der Bauhaus-Universität Weimar, 2003.

[3] N. Leach, "Digital cities," Architectural Design, vol. 79, no. 4, pp. 6-13, June 2009.

[4] F. Gramazio, M. Kohler, and J. WillMann, The robotic touch: How robots change architecture: Gramazio&Kohler Research ETH Zurich 2005-2013, 1st ed. Zurich: Park Books, 2014.

[5] F. Chen, G. M. Brown, and M. Song, "Overview of 3-D shape measurement using optical methods," Optical Engineering, vol. 39, no. 1, pp. 10-22, January 2000.

[6] R. Zhu, O. Furxhi, D. Marks, and D. Brady, "Millimeter wave surface and reflectivity estimation based on sparse time of flight measurements," 39th International Conference on Infrared, Millimeter, and Terahertz waves (IRMMW-THz), IEEE Press, September 2014, pp. 1-2.

[7] J. Schlarp, E. Csencsics, and G. Schitter, "Scanning laser triangulation sensor geometry maintaining imaging condition," IFAC-Papers Online, vol. 52, no. 15, pp. 301-306, December 2019.

[8] M. Slembrouck, P. Veelaert, D. Hamme, D. Van Cauwelaert, and W.Philips, "Cell-Based Approach for 3D Reconstruction from Incomplete Silhouettes," International Conference on Advanced Concepts for Intelligent Vision Systems, Springer Press, September 2017, pp. 530-541.

[9] J. S. Franco and E. Boyeb, "Efficient polyhedral modeling from silhouettes," IEEE Transactions on Pattern Analysis and Machine Intelligence, April 2009, pp. 414-427.

[10] H. Hua, "A bi-directional procedural model for architectural design," Computer Graphics Forum, vol. 36, no. 8, pp. 219-231, December 2017.

[11] G. Brugnaro, E. Baharlou, L. Vasey, and A. Menges, "Robotic softness-an adaptive robotic fabrication process for woven structures," Proc. 36th Annual Conference of the Association for Computer Aided Design in Architecture, October 2016, pp. 154-163.

[12] P. Eversmann, F. Gramazio, and M. Kohler, "Robotic prefabrication of timber structures: towards automated large-scale spatial assembly," Construction Robotics, vol. 1, no. 1-4, pp. 49-60, August 2017.

[13] M. Maasoumy and A. Sangiovanni-Vincentelli, "Smart connected buildings design automation: foundations and trends," Foundation and Trends in Electronic Design Automation, vol. 10, no. 1-2, pp. 1-143, March 2016.

[14] R. Snooks and G. Jahn, "Closeness: on the relationship of multi-agent algorithms and robotic fabrication," Robotic fabrication in architecture, art and design, Springer Press, February 2016, pp. 218-229.

[15] J. P. Sousa, C. G. Palop, E. Moreira, A. M. Pinto, J. Lima, P. Costa, et al. "The SPIDERobot: a cable-robot system for on-site construction in architecture," Robotic Fabrication in Architecture, Art and Design, Springer Press, February 2016, pp. 230-239.

[16] R. Rust, D. Jenny, F. Gramazio, and M. Kohler, "Spatial wire cutting: cooperative robotic cutting of non-ruled surface geometries for bespoke building components," Proc. 21st International Conference on Computer-Aided Architectural Design Research in Asia, Southeastern University Press, March-April 2016, pp. 529-538.