Volume xx (200y), Number z, pp. 1–15

Interactive Iconized Grammar-based Pailou Modeling

Zhong-Qi Cai [1] and Ying-Sheng Luo [1] and Yu-Chi Lai [1] [†] and Chih-Shiang Chan [2] and Wen-Kai Tai [1]

[1] CSIE Department, National Taiwan University of Science and Technology, Taiwan
[2] CSIE Department, National Taohua University, Taiwan

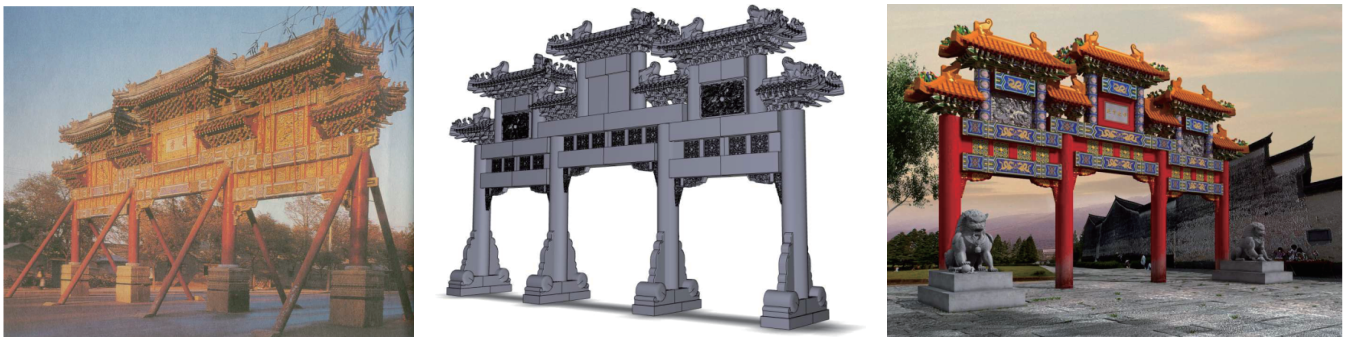

Figure 1: *Our Pailou modeling system can easily let novices imitate an existing Pailou (left) to construct its equivalent 3D representation (middle). While using textured and carved structural components, the constructed Pailou becomes vivid and colorful to populate a virtual world (right).*

Abstract

Pailous are representative Chinese architectures used to express commemoration, but their geometric structures and semantic construction rules are too complex to fast and intuitively model them with traditional modeling tools. Therefore, this work proposes an intuitive modeling system for stylized creation and manipulation for novices. Our system encapsulates their structural components as icons and semantic layouts as topological graphs while users can create and manipulate icons with topological recommendations. Our interpreter can automatically and immediately transform a graph to its corresponding model with provided components using our parametric L-system grammars derived from architectural rules. While using it to imitate existing representative Pailous and design imagined ones, the constructed results can achieve desired visual complexities. When comparing to an existing 3D model tool, Maya, in modeling a Pailou and Toukung, our system is effective and simple without requiring remembering and understanding those complex rules.

CCS Concepts

•***Computing methodologies*** → ***Graphics systems and interfaces; Shape modeling;***

1. Introduction

Pailous, also called Paifangs, are made of painted and ornamented wood or carved stone with glazed roofs. They, acting similarly as Arc de Triomphe in Paris, are important architectures in East Asia standing in front of mausoleums, temples, bridges, thoroughfares, and parks. In order to express commemoration and blessing, they usually have decorated boards of carved moral inscriptions written by calligraphers [Man92]. However, their structural components, topological layouts, and construction rules are generally complex, and it would require architectural expertise and 3D modeling skills to create stylized Pailous in order to populate an East Asian world. In other words, it is impossible for novices to build their own Pailous in a short period of time. Therefore, this work aims at simplifying the modeling process by having an interactive system to design and build Pailous of visually correct structural components and topological layouts.

There are several possible modeling mechanisms for building different Pailous. 3D modeling systems, such as Maya and 3ds Max, let users build all structural components and topological layouts

† Corresponding author:cheeryuchi@gmail.com

from scratch, but it requires knowledge of all architectural details and takes hours to build a Pailou. Grammar-based methods, such as L-system grammars [MPB05], shape grammars [MVW*06, KPK10], and split grammars [WWSR03, MWH*06], are alternatives to procedurally create various models. However, to design a realistic Pailou with them would require the familiarity of both architectural semantics and production grammars. As a result, because Pailous consist of independent structural components topologically linked together, our system encapsulates structural components using icons and encodes their topologies using layout graphs while users can directly select plausible icons based on derived semantic examination for intuitiveness and interactivity. Teoh [Teo09] designs procedural descriptions for many ancient East Asian architectures and provides a library of various examples. However, appending a new style of Pailous is still hard due to the need of architectural expertise and his procedural descriptions while storey-based representations complicate modeling topologically complex structures, such as Toukungs. Additionally, there are also systems focusing on procedurally modeling Chinese Tings [HT13], Chinese Palaces [Liu05], and Southern vernacular houses [YCZY04], which carry the same spirit as ours, but their derived rules are specific and cannot directly be used for Pailous. To the best of our knowledge, our system is the first attempting to interactively, intuitively, and procedurally design and model various Pailous.

After implementation, we can imitate realistic Pailous shot in collected pictures and design imagined stylized ones within a very short period of time. Furthermore, in order to understand its effectiveness, we designed a user study of modeling a Toukung and Pailou using our framework against a 3D modeling system, Maya. Generally, the participants, who does not have any architectural knowledge, can easily create and design their own in a very short period. As a result, it shows the effectiveness and intuitiveness of our framework. We make the following **contributions** in the paper. We have identified an interesting application of intuitive Pailou design for novices. The core is a graph-based interactive system, whose nodes geometrically encapsulate structural components, and edges semantically represent topological connections among components, while users select plausible icons with topology-recommended examination. Simultaneously, our system can directly and procedurally interpret the component graph using our parametric L-system grammars derived from the architectural semantics to construct its corresponding model with provided components. The rest of the paper is organized as follows. Section 2 overviews related works. Section 3 summarizes Pailou's structural components and grammars along with their categorization. Section 4 describe our iconized graph-based Pailou modeling system along with our derived geometric and semantic grammars. Section 5 demonstrates several digital Pailous built with our interactive system along with a comparison user study. Section 6 concludes this work and gives several future research directions.

2. Related Work

Architectural modeling has long been the research topics for decades in Computer graphics, Architectures, and heritage digitization. However, due to the length limitation, we only focus on directly related research, and interesting readers can refer to a recent survey [STBB14] for more details. Traditionally, commercial 3D systems, such as Maya and 3D Max, provide user interfaces to create structural components and transform them to design semantically and topologically correct Pailous from scratch. However, the modelers need architectural expertise and 3D modeling skills, and the process generally takes hours to days. We overcome the limitations by letting designers select legitimate component icons based on our derived construction regulations.

Another category is grammar-based procedural modeling. Prusinkiewicz *et al.* [PL90, PJM94, PMKL01] apply L-system rewriting and turtle interpretation to describe various branching plants while research [PM01, MPB05] extend the concept to build various hierarchical architectures. They are compact and can build various models with a few derived parameters, and Pailous are symmetric and grow from the central entrance upwardly and laterally with branching structures similar to plants. Therefore, we derive L-system representations for Pailous and Toukungs according to structural and semantic analysis for efficiency, flexibility, controllability, and portability. However, there are a few application issues. They require modeling expertise to design corresponding descriptive grammars. Complex and counter-intuitive rewriting and interpretation mechanisms make it hard to modify an existing one even with the help of an expert. Furthermore, the effect of parameter adjustment is generally complex and less intuitive. Therefore, we design an interactive modeling system with plausible component selection enforced with our derive semantic rules to construct a component graph. Our interpreter can immediately transform the graph to its corresponding L-system string and 3D representation along with provided components. Shape grammars [Dua02, Hav05, MVW*06, KPK10] use shape rules and generation engines to model various buildings. Split grammars [WWSR03, MWH*06, SM15] extend shape grammars to decompose a shape into a set of smaller shapes while providing mechanisms for automatic attribute adjustment and selection within provided grammars. Müller *et al.* [MVW*06] apply shape grammars for a set of collected ancient Mayan examples. Writing grammar sets is hard for designers inexperienced in programming because it requires both familiarity of descriptive grammars and architectural rules. This makes it hard for novices. On the contrary, we extract their structural components as component icons and topological regulations as a topological graph to describe a specific Pailou while users can directly select plausible icons enforced with our derived semantic examiner to design their own Pailous. Although there are research using graph-based grammars [GK07] or developing visual editing tools for grammar-based methods [LWW08, Pat12], requiring familiarity of architectural rules is still hard for novices. Therefore, we derive a construction-enforced examiner to let user incrementally append plausible components for intuitiveness and easiness.

per. There are research [BWS10, ZXJ*13] automatically decomposing an input model based on high-level design objectivity, such as symmetry, into a procedural representation. They can easily assemble the input with a few tuning parameters, but they generally lack good mechanics to do fine-grained control over the result. Additionally, the input generally comes from pictures, point clouds, and models, and this limits it to create non-existing, novel, or non-

modeling architectures. Therefore, we develop an icon-graph-based and construction-rule-enforced interactive modeling system with greater freedom. Similarly, image-based methods [JTC09, NBA18] can use user indication and deep learning respectively to formulate architectural grammars. They have similar limitations as the above methods. Furthermore, it is hard to reconstruct fine 3D details such as tiles, friezes, and balustrades, and Pailous' structural and topological rules are complex and hard to encode. Therefore, we design a simple modeling mechanic of icon selection. Data-driven learning-based algorithms [YCZY04, LYH10, RMBG*13, RJT18] automatically define grammars from a large set of examples. Generally, these algorithms aim at generating 3D models in similar types and developing proper descriptive grammars. However, while the provided examples are not thorough, they cannot find descriptive mechanisms for missed examples. For examples, Corinthian capitals in Greek temples and Toukungs of ancient East Asian roofs are important decorations, but they might miss from all examples, these methods cannot find their descriptions. Additionally, although they can easily assemble similar buildings of provided examples, they generally lack fine-grained control over the result in order to achieve users' imagination. We manually determine a thorough set of structural components as icons, and derive topological layout rules as a graph while users can construct the graph with plausible icon selection enforced and ensured by our grammar examiner. At the same time, they can also manipulate icon-embedded parameters for fine-grained control. While importing different pre-built models and incorporating with other procedural modeling methods, our system can easily incorporate fine features, such as carved patterns, into procedural construction. Additionally, there are also inversion research [WOD09, TLL*11, RMGH15] deriving parameters based on the structural stability and other goals to create desired buildings using specific procedural modeling mechanisms. Although they can automatically estimate a set of proper parameters, users must provide proper objective functions and building descriptions which require architectural expertise. Our system directly removes the burden with an icon-based modeling system with enforcement of architectural rules and parametric manipulation.

There are also research [CKX*08, NGDA*16] using sketches to model architectures and adjust certain features of the provided architectures. Although they are simple and intuitive, sketching architectures for users without training is difficult. Their success generally depends on successfully matching features in the collected database while Pailous' complex topological knowledge is hard to encode with simple features. Furthermore, sketching generally lacking of precision [OSSJ09] for proper and precise modeling makes it hard to create and manipulate fine-detailed components. In order to achieve simpleness and intuitiveness, we let users directly select semantically correct components represented with icons with immediate feedback.

Teoh [Teo09] aims at developing a procedural representation based on the unit of storeys to describe as many ancient east Asian architectures as possible along with a set of exemplar buildings. They also provide a parametric representation for four major roof types and a lego-like representation for bracket sets. However, this method decomposes an architecture into a set of storeys for various components. Representing a Pailou, which generally has long pillars and lintels with extra decoration components attached or interconnected with each other, requires a large number of storeys. This becomes even worse while we extend entrances vertically and laterally and decorate the main structure with more appending components. Furthermore, it still requires structural and topological knowledge of Pailous and bracket sets, i.e.Toukungs, for designing a new style. All make it hard for novice. There are also research aiming at provide better procedural control and modeling abilities for special types of Asian architectures including Chinese Tings [HT13], Chinese Palaces [Liu05], and Southern vernacular houses [YCZY04], but their rules are specific for their goals and cannot directly be used for Pailous. This work aims at providing intuitive and comprehensive modeling user interface to design various Pailous.

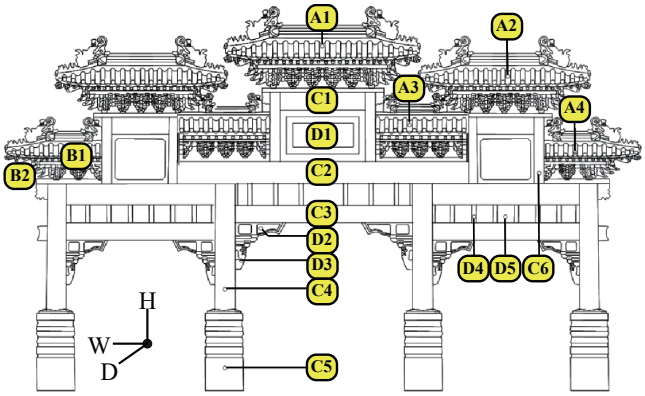

Figure 2: *This illustrates the structural components including roofs (A), Toukungs (B), bodies (C), and decorations (D). Based on its location and relationship to other structures, roofs can be further categorized into the main roof (A1), next roofs (A2), clipped roofs (A3), and side roofs (A4). Toukungs locate among roofs and bodies based on their relative location to roofs can be categorized as middle Toukungs (B1) and side Toukungs (B2). Bodies can be further categorized into single lintels (C1), big lintels (C2), small lintels (C3), pillars (C4), pillar bases (C5), and suspended pillars (C6). Others are decoration including inscribed boards (D1), Quetis (D2), Yundans (D3), booklet pillars (D4), and flower boards (D5).*

3. Architectural Background in Pailou

Pailous are diverse and complex, but they still follow a set of construction rules [Man92]. They generally consist of three major components from ground to top: bodies, Toukungs, and roofs as shown in Fig. 2. Bodies consist of two elements, pillars and lintels. Fundamentally, a lintel connects two pillars to form a fundamental entrance in the shape of Π, and the same Π structure can extend upwardly by stacking one above another, while it can also extend laterally by aligning one next to another with a sharing pillar. Pillars, whose shape is cylindrical or cuboidal, and surface has paintings or carving, structurally ensure the standing and hold up roofs and Toukungs. Their number are traditionally even and determine Pailou's total width while two neighboring pillars form an entrance where the central one is the widest. Furthermore, there are two types, main and suspended pillars, based on whether they touch

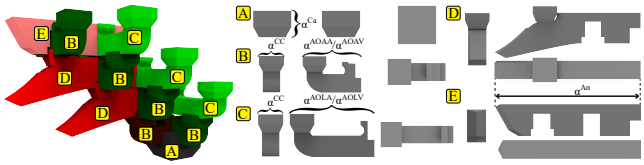

Figure 3: *The left is a 3-level middle Toukung consisting of the cap (A), axial-oval arms (AO) (B), axial-long arms (AL) (C), Angs (D), and noses (E). The right shows the top, front, and side views of an exemplar set along with measurements of cap's thickness, connection's width, Ang's length, perpendicular AO's length, parallel AO's length, perpendicular AL's length, and parallel AL's length where perpendicular denotes the arm is perpendicular to the roof ridge and parallel denotes the arm parallel to the roof ridge. The measurements are expressed as ratios to the pillar height denoted as α^{Ca}, α^{CC}, α^{An}, α^{AOAA}, α^{AOAV}, α^{AOLA}, and α^{AOLV} detailed in Section 4.4.*

the ground. There are usually pillar bases at the bottom of the pillar to strengthen pillars' support with fantastic carving. The main function of lintels is to support Toukungs and other structures. There are three types: big, small, and single lintels according to their functionalities: big lintels, which are above the pillar tops or between two pillars, support Toukungs and upper structures; small lintels, which are between pillars and under big lintels, connect with lower structures such as Quetis or act as optional decorations; single lintels appear along with suspended pillars to support Toukungs only. Generally, there are optionally decorated components including inscribed boards, flower boards, and booklet pillars. Inscribed boards are made of wood or stone for inscription of commemoration or decorated carving patterns and generally stay in the middle of the entrances. Additionally, intricate Pailous generally append a Queti and a Yundan at the corner of a lintel and a pillar where Yundan is always below a Queti. Optionally, architects can append roofs above the lintels while they can also add intricate painted or carved bracket sets, Toukungs, between a roof and a lintel.

Roofs consist of numerous tiles and ridges accompanying with stone-made animals decorated along the ridge [HT13] while there are optionally Toukungs under them. We take their structural roles of Pailous into consideration as main, next, clipped, and side roofs: the main roof is the central top one with the largest size; next roofs are next to the main Roof one or another next roof; clipped roofs are those between two suspended pillars of respective entrances, and they generally do not have eaves corners; side roofs are at both ends with only half of the roof structure.

Toukungs, usually made of wood with fine carvings and colorful paintings to enrich appearance, are under roofs to act as supporting structures. As illustrated in Fig. 3, they generally consists of five elements, caps, axial-oval arms, axial-long arms, Angs, and noses, interlocking to form a stable structures for supporting extended roofs. They are two types, side and middle Toukungs, based on their location relative to the supported roof as B1 and B2 shown in Fig. 2: side Toukungs appear at the corner to hold the eaves corners while middle Toukungs are between side Toukungs.

4. Iconized Graph-based Interactive Framework with Procedural Representation

Each entrance is symmetric, and thus we can generate the left half and mirror it to the right. While starting from the center of the central entrance, we can hierarchically connect all components as a component graph, whose nodes are component icons, and edges are topological connections among components. As shown in Fig. 4, we decompose and analyze existing Pailous to determine a set of structural components. At the same time, we collect and model corresponding 3D models for various structural components. While studying their semantic rules [Man92], a parental component in the graph has a fixed set of plausible connecting components. For example, a pillar can allow the connection from a lintel but not from an inscribed board. We express all connection relationship as a relation matrix detailed in Section 4.1. Similarly, while starting from the cap, Toukungs are branch-like using repeated interconnection rules. Both structures are suitable for parametric L-system representations. In order to procedurally construct Toukungs in different levels and various Pailous, we derive different sets of L-system representations detailed in Section 4.2 and 4.3.

In order to intuitively and simply create a Pailou in a few minutes for novices, we design an icon-based framework to interactively construct the component graph as shown in Fig. 6. Users start from a basic entrance of a lintel with two pillars. They can interactively identify an existing component as the active, and our system shows all legitimate components based on the corresponding row in the relation matrix. They can incrementally select the desired components, such as flower boards, by clicking the icon to add them into the component graph with default parameters. Additionally, users can also manipulate component-based parameters, such as the number of levels for a Toukung, to finely adjust their appearance. Currently, while changing happens, our system runs collision detection to avoid penetration among components. Furthermore, users can drag icons to have a better layout for examination. At the same time, our system analyzes the graph to create its layout chart and form L-system strings. Finally, it interprets the strings to create the corresponding 3D Pailou and Toukungs with the provided 3D components for immediate feedback. Generally, architects the lateral structures into three parts: main, repeating, and side where the main constructs the central entrance, the repeating constructs multiple middle entrances, and the side constructs the outmost entrance as shown in Fig. 7. Although the main entrance may have different structures, the lateral growth usually has the same layouts. Therefore, we provide a grouping mechanic to group an existing entrance set as a pre-built component graph and duplicate them to reduce the labor to append them forward and laterally

4.1. Relation Matrix

As shown in Fig. 8, this work uses a relation matrix to represent the legitimate linking relation between all possible pairs of components where a relation matrix is a $N^C \times N^C$ matrix. Originally, there are 7 structural components. However, roofs may optionally have eave corners, and thus, we have two different roof types: clipped and eaves roofs. Correspondingly, there are two different Toukung types: middle and side Toukungs. Therefore, our current implementation has 10 components. Each row is a structural component

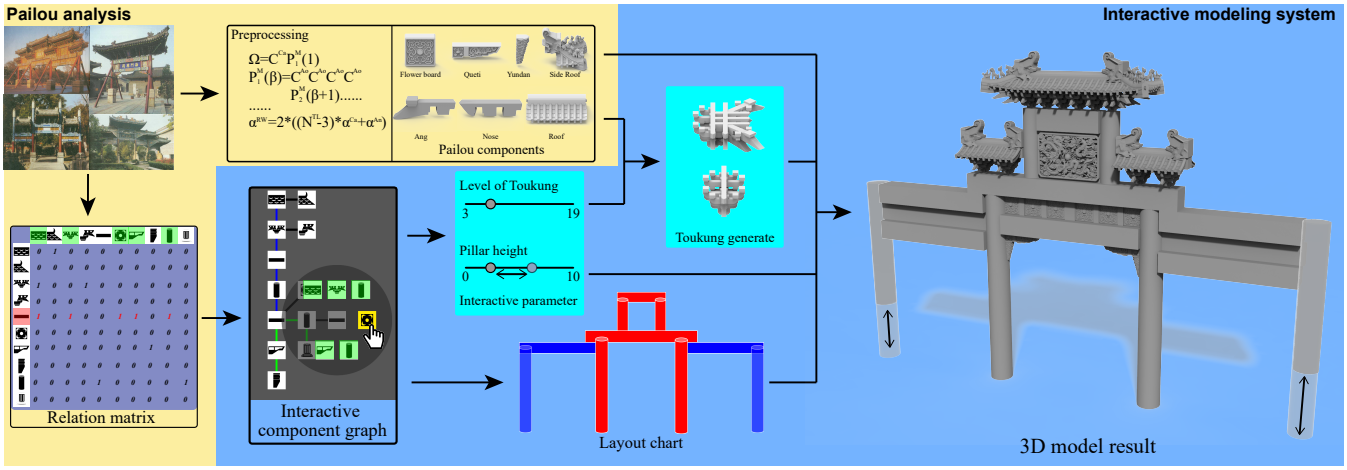

Figure 4: *Preprocessing analyzes structural and topological rules to determine a set of structural components along with their 3D models, create a relation matrix of componential interconnection relationship, and derive L-system grammars for Pailous and Toukungs. During the run time, a user can interactively and incrementally add plausible component icons into a component graph based on the relation matrix and adjust component-based parameters while our system can immediately interpret the graph along with parameters to the corresponding Toukungs and Pailous.*

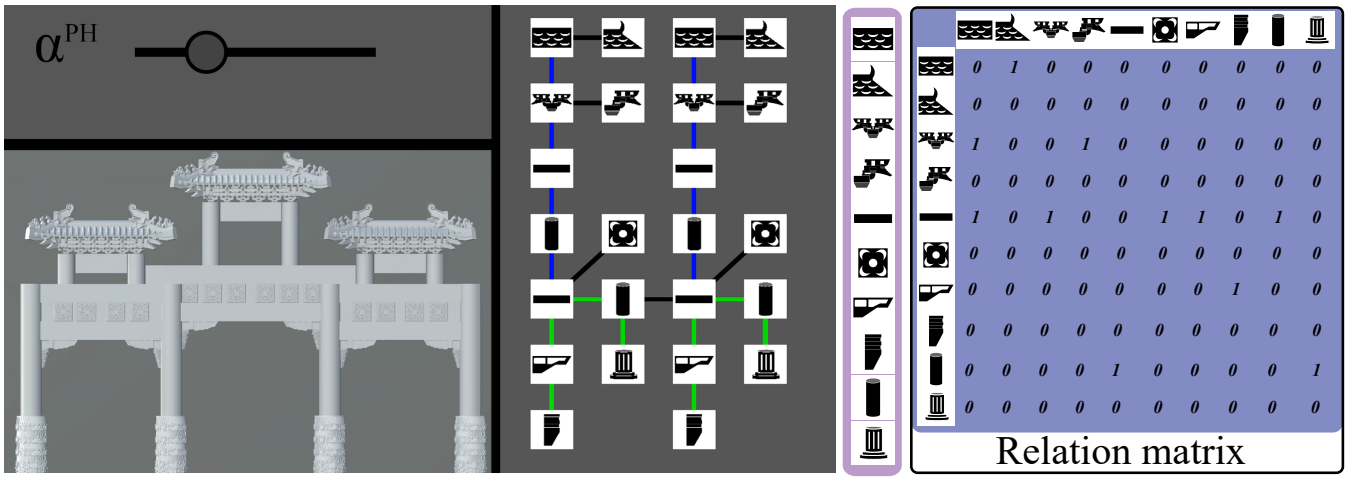

Figure 5: *Our iconized interface mainly consists of two elements: a graph (A) to describe the topological relationship among components and components icons (B) along with the relation matrix (C). From top to bottom are the clipped roof, the eaves roof, the middle Toukung, the side Toukung, the lintel, the pillar, the Queti, the flower board, and the Yundan.*

identified with its corresponding icon. While the boolean value at a column of the row indicates whether the column component can legitimately connect to itself. For example, 1 at the $(5,3)$-th element means that Toukungs can legitimately link to a lintel.

4.2. Pailou Layout Chart Construction and Topological L-system Formulation

When starting from the bottom cap of Toukungs and the middle of the central entrance, their topologies are similar to branching trees. L-system is a string rewriting mechanism based on designed production grammars [PL90] which can be expressed as an ordered triplet $\mathscr{G} = \langle \mathscr{V}, \Omega, \mathscr{P} \rangle$ where $\mathscr{V}$ is a set of alphabets, Ω is the axiom, i.e., a word, and $\mathscr{P}$ is a set of production grammars. Table 1 provides the interpretation of those commonly used symbols in our representation while we also incorporate with parameters to extend its modeling abilities. Research generally use it to model branching structures, and therefore, we derive our own L-system grammars to represent both structures with the following considerations. 1) During production, we can easily modify scale ratios and use them to define the frame of Toukungs and Pailous. 2) Production applies in parallel to generate a string whose terminal symbols contain positional and geometric information. As a result, we formulate the relationship among components with derived parameters and pro-

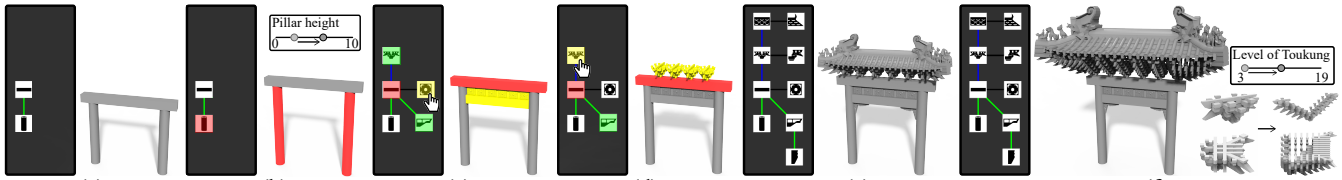

Figure 6: *This shows an exemplar construction procedural. A user can start from a basic Pailou (a) and set the pillar as active component (marked in red) and the flower boards (marked in yellow) while our system show all legitimate components including Toukungs, flower boards, and Quetis (marked in green) (b). The process can repeat until we have the final result (d)(e). At the same time, the user also changes the level of Toukungs from* 3 *to* 7 *(f).*

Symbol	Interpretation
F	Move forward and draw a line.
F(d)	Move forward by distance d and draw a line with d units.
f	Move forward without drawing a line.
f(d)	Move forward by distance d without drawing a line.
$+$	Turn left by default.
$+(\theta)$	Turn left by angle θ.
$-$	Turn right by default.
$-(\theta)$	Turn right by angle θ.
$\wedge$	Pitch up by default.
$\wedge(\theta)$	Pitch up by angle θ.
&	Pitch down by default.
$\&(\theta)$	Pitch down by angle θ.
$<$	Roll left by default.
$<(\theta)$	Roll left by angle θ.
$>$	Roll right by default.
$>(\theta)$	Roll right by angle θ.
|	Turn 180 degrees. This is equivalent to $+(180)$ or $-(180)$.
%	Roll 180 degrees. This is equivalent to $<(180)$ or $>(180)$.
$	Pitch 180 degrees. This is equivalent to $\wedge(180)$ or $\&(180)$.
[	Start a branch. Push the current state of the turtle onto a stack.
]	Complete a branch. Pop a state form the stack.
$\sim X$	Incorporate a predefined surface identified by symbol X.
$\sim X(s)$	Incorporate a predefined surface identified by symbol X, scaled by s, at the turtle's current location and orientation.

Table 1: *This lists those commonly used L-system symbols for their corresponding Turtle interpretation.*

cedurally model them with the parametric L-system for efficient, controllable, flexible, and portable modeling abilities in the following paragraphs.

Architects stack Π-shaped entrances upwardly and align them laterally as shown in Fig. 7. Then, they append decorative components, such as flower boards and Quetis, at the corresponding lintels and pillars while requiring auxiliary components to support appending ones, such as an extra lintel is added to support appending flower boards. As shown in Fig. 9(a), we first decompose the component graph into consisting entrances, expressed as a **layout chart**, which records the upward and lateral hierarchy of entrances represented as a set of colored line segments for its lintel and pillars, in the following steps. We start at the central lintel-pillar combination. At each combination, while there is an extra pillar connected to its lintel, we add an upward entrance. Similarly, while there is an extra lintel connected to its pillar, we append a lateral entrance. We set the traversing combination at the next available upward and lateral ones, and repeat the traversing until we reach the top end of all branches.

After determining all upward and lateral entrances, we can procedurally write out the parametric L-system string and build the corresponding Pailou as shown in Fig. 9(b), (c), and (d). Our current implementation builds the clipped roofs, eaves roofs, middle Toukungs, and side Toukungs on the fly. They are represented them with $R^C(\mathbf{p})$, $R^E(\mathbf{p})$, $T^M(\mathbf{p})$ and $T^S(\mathbf{p})$ where $\mathbf{p}$ represents their corresponding parameters. Pillars, lintels, inscribed boards, pillar bases, Quetis, Yundans, booklet pillars, and flower boards are loaded as pre-modelled 3D models. They are represented with $M^P(i,\mathbf{p})$, $M^L(i,\mathbf{p})$, $M^{IB}(i,\mathbf{p})$, $M^{PB}(i,\mathbf{p})$, $M^Q(i,\mathbf{p})$, $M^Y(i,\mathbf{p})$, $M^{BP}(i,\mathbf{p})$, and $M^{FL}(i,\mathbf{p})$ respectively where i indicates the i-th 3D

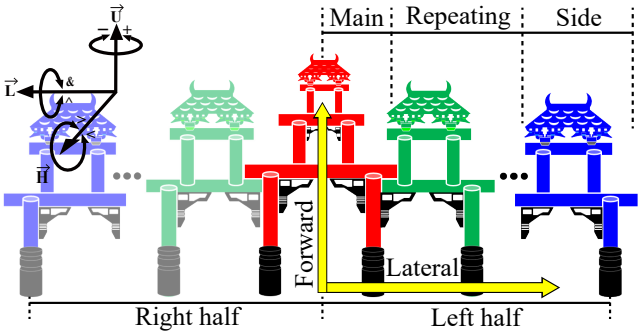

Figure 7: *This shows that Pailou are symmetric, and its topological layout can generally divide into the upward and lateral directions where the upward increases its height by adding entrances on the top and the lateral increases its width by adding entrances next to the current one.*

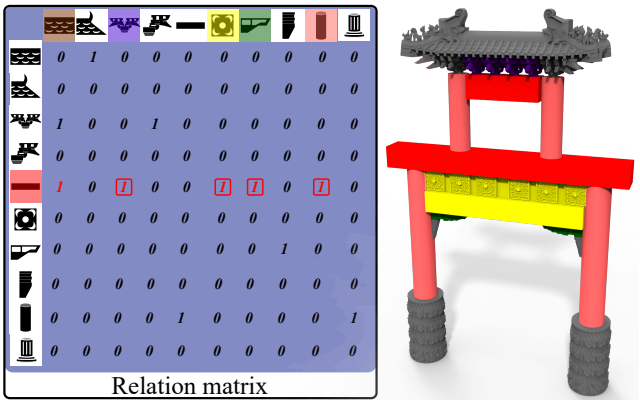

Figure 8: *This shows an exemplar use of the relation matrix while the active component marked in red is the lintel listed in the 5-th row. All components plausibly connected to the lintel have their corresponding column of a value as 1 while Toukungs marked in blue, flower boards marked in green, pillars marked in purple, and Quetis marked in yellow are those actually selected in the right example.*

model. Combining both procedural and geometric representations, which are imported and constructed as a whole, allows us to compute the desired position, orientation, and scale along with the corresponding geometric model. For example, we can import a model for the pillar base and generate it according to the turtle interpretation while we use the roof modeling tool proposed by Huang *et al.* [HT13] to procedurally design roofs and generate them as a whole according to the turtle interpretation. We first initiate the L-system string from the central entrance, G1, as shown in Fig. 9(b) as $\Omega = P^P_{11}$. Then, we first add the string segments for the pillar and lintel of $C^P(i,\mathbf{p},\mathbf{x},\mathbf{r})$ and $C^L(i,\mathbf{p},\mathbf{x},\mathbf{r})$ where $C^X = T(\mathbf{x})R(\mathbf{r})M^X$, $T(\mathbf{x})$ indicates the translation of $\mathbf{x}$, $R(\mathbf{r})$ indicates the rotation of $\mathbf{r}$, and M^X is the production string for the object, such as M^p for a pillar. For simplification, the following neglects i, $\mathbf{p}$, $\mathbf{x}$, and $\mathbf{r}$ from the denotation. At the same time, we also add decoration components of a Queti, flower boards, booklet pillars, and a Yudan. This point also initiates G12 and G21 entrances, P^P_{12} and P^P_{21}. Therefore, $P^P_{11} = C^P C^L C^Q C^Y C^{FB} C^{FB} C^{FB} C^{BP} C^{BP} C^{BP} P^P_{12} P^P_{21}$. In the next iteration, P^P_{12} produces the pillar, lintel, Toukungs, and others while P^P_{21} produces the entrance and its appending structures. By repeating the process, we can generate the entire corresponding L-system string. In other words, our system first inserts the central main entrance and its decorative components. Then, we append the above and neighboring entrances. The process repeats until there is no entrance above or next to the current active one. Fig. 10 provides the complete L-system production process for this example.

4.3. Procedural Toukung Construction

When carefully observing Toukungs, we can decompose them into three parts, capping, repeating, and supporting, as shown in Fig. 11. The capping part consists of the cap, which produces the base to start the construction, along with a level of bracket sets as shown in Fig. 11. The repeating consists of levels of bracket sets along with Angs to extend the structure outward. The supporting consists of a set of brackets and Noses to support the roof. Side Toukungs need to support the eaves corners where they must elongate one side asymmetrically while the inner half has similar structures as those in middle Toukungs. Accordingly, we derive two different parametric L-system grammars for the middle and side parts respectively. Our current implementation generates caps, axial-oval bracket arms, axial-long bracket arms, Angs, and noses using pre-modelled 3D models while representing them as $M^{Ca}(i,\mathbf{p})$, $M^{AO}(i,\mathbf{p})$, $M^{AL}(i,\mathbf{p})$, $M^{An}(i,\mathbf{p})$, and $M^{No}(i,\mathbf{p})$. We first initiate the **middle** part at the cap as shown in the bottom of Fig. 12(a). In order to simplify our discussion, we use left-right to denote the direction from left to right according to the view of Fig. 12 while using out-in to denote the direction from the far side of the view to the near side. The cap also links to left-right and in-out axial-oval sets denoted as P^M_1, and thus, $\Omega = C^{Ca}P^M_1(1)$. In the second iteration, we generate all four axial-oval arms, C^{AO}. Each left-right arm links with an in-out axial-oval set and an Ang/nose depending on whether it is the last level, P^M_2, while each in-out arm connects to an in-out axial-long arm, P^M_5. Thus, $P^M_1(\beta) = C^{AO}C^{AO}C^{AO}C^{AO}P^M_2(\beta+1)P^M_2(\beta+1)P^M_5P^M_5$. In the third iteration, $P^M_5 = C^{AL}$ generates an axial-long arm. P^M_2 generates an in-out axial-oval set and an Ang/nose while the Ang links to an in-out axial-oval set and an Ang/Nose depending on whether it is the last level, P^M_3, in order to support the roof. We express P^M_2 and P^M_3 as follows.

$$P^M_2(\beta) = \begin{cases} C^{AO}C^{AO}C^{An}P^M_3(\beta+1)P^M_5P^M_5 & if \beta < N^{TL}-1 \\ P^M_4 & otherwise \end{cases} \quad (1)$$

$$P^M_3(\beta) = \begin{cases} C^{AO}C^{AO}C^{An}P^M_5P^M_5P^M_3(\beta+1) & if \beta < N^{TL}-1 \\ C^{AO}C^{AO}C^{An}P^M_5P^M_5P^M_4 & otherwise \end{cases} \quad (2)$$

And $P^M_4 = C^{AO}C^{AO}C^{No}$. The process repeats until reaching the destined level. Fig. 13 provides the complete L-system production grammars.

We first initiate the **side** part at the cap as shown in the top of Fig. 12(a). In order to simplify our discussion, we use the side arm to denote the arm extending in the eaves' direction and the in, our, left, and right arms to denote the arm respectively in the far, near, left, and right sides of the viewer. The cap also links

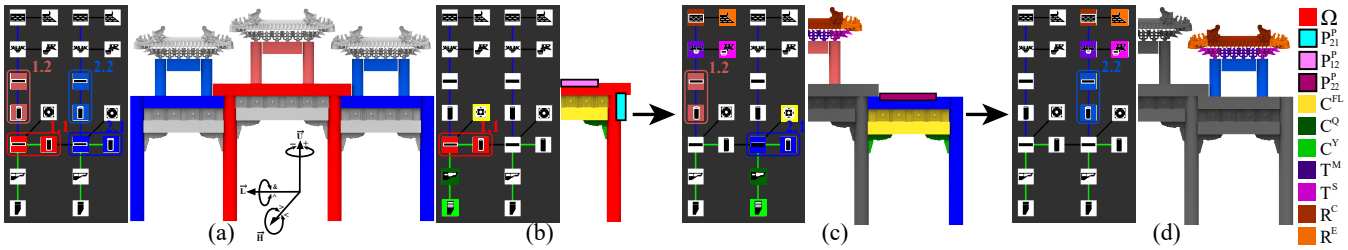

Figure 9: *(a) shows that we start traversing from the central entrance (G11) upwardly to the above entrance (G12) and laterally to the neighboring entrance (G21). Recursively, we can find the one (G22) above the G2 entrance. From (b) to (d) demonstrates our L-system string formulation, we iteratively insert the string segments for P^P_{11}, P^P_{12}, P^P_{21}, and P^P_{22}. Each P^P appends string segments for its corresponding pillar, lintel, and decorative components.*

to left-right and in-out axial-oval sets and denoted as P^S_1. Additionally, in order to support the eaves corner, there is an extra arm rotating 45 degrees around the y axis. Therefore, $\Omega = C^{Ca}P^S_1(1)$. In the second iteration, we generate all five axial-oval arms, C^{AO}. The in arm connects with an in-out axial-oval set and an Ang/nose depending on whether it is the last level, P^M_2. The out arm links with an axial-oval arm, P^S_6, and an Ang/nose depending on whether it is the last level, P^S_2. The left arm connects to an Ang/nose depending on whether it is the last level, P^S_2. The right arm links with an axial-long arm, P^S_8. The side arm connects with three Angs/noses depending on whether it is the last level, P^S_4. Thus, $P^S_1(\beta) = C^{AO}C^{AO}C^{AO}C^{AO}C^{AO}P^M_2(\beta+1)P^S_2(\beta+1)P^S_6(\beta+1)P^S_2(\beta+1)P^S_8P^S_4$. In the third iteration, P^M_2 propagates and generates as the one described in Eq. 1. P^S_2 generates an Ang/nose while the Ang connects to an Ang/nose depending on whether its the last level, P^S_9.

$$P^S_2(\beta) = \begin{cases} C^{An}P^S_9 & if\beta < N^{TL}-1 \\ P^S_5 & otherwise \end{cases} \quad (3)$$

$$P^S_9(\beta) = \begin{cases} C^{An}P^S_9 & if\beta < N^{TL}-1 \\ P^S_5 & otherwise \end{cases} \quad (4)$$

P^S_4 generates three Angs/noses, while the in and left arms each connect to another Ang/Nose depending on whether it is the last level, P^S_4, and the side connects to three Angs/noses depending whether it is the last level, P^S_4. Therefore,

$$P^S_4(\beta) = \begin{cases} C^{An}C^{An}C^{An}P^S_9P^S_9P^S_4 & if\beta < N^{TL}-1 \\ P^S_5P^S_5P^S_5 & otherwise \end{cases} \quad (5)$$

$P^S_5 = C^{No}$ generates a nose. P^S_6 generates an axial-oval arm, C^{AO}, and the arm links to an axial-oval arm, P^S_6, and an Ang/nose depending on whether it is the last level, P^S_2.

$$P^S_6(\beta) = \begin{cases} C^{AO}C^{An}P^S_6 & if\beta < N^{TL}-1 \\ C^{AO}P^S_5 & otherwise \end{cases} \quad (6)$$

$P^S_8 = C^{AL}$ generates an axial-long arm. The process repeats until reaching the destined level. Fig. 14 provides the complete L-system production grammars.

4.4. User-Controlled parameters

In addition to structural components and topological layouts, parameters are also important to vary their appearance. Since our system connects all components in an axis-aligned manner, we can use an axis-aligned bounding box (AABB) for manipulation. Fig. 9(a) has the coordinate and rotation L-system symbols for our operation while Fig. 2 has the coordinate of width, height and depth. Furthermore, we create each pre-modelled model, whose front facing the minus depth direction, center at the origin, and three AABB dimensions are 1, respectively. While plugging into our procedural process, the modeling dimensions of AABBs are expressed in terms of the pillar height of their corresponding entrance, H^P_{ij}, where i and j is the index of the entrance, and the pillar height of the central entrance is a user specific parameter, ω. In other words, these dimensions may vary from one entrance to another depending on their pillar height. Generally, architects uniformly scale those extended upward and laterally with α^{Up}_{ij} and α^{Side}_{ij} embedded with the pillar icon. For example, $H^P_{12} = \alpha^{Up}_{11}H^P_{11}$, and $H^P_{21} = \alpha^{Side}_{11}H^P_{11}$. Since traditional pillars are cylinders, the scale ratios for their width and depth are the same and modified simultaneously using α^{PW}_{ij}. The cross section of a lintel is generally a square, and the ratios are also modified simultaneously using α^{LH}_{ij}. The roof dimension is related to the size of middle and side Toukungs under it, and details are given in next paragraphs. We can compute the center of each component according to its belonging entrance along with the corresponding pillars and lintels. For example, the center of the right pillar of the central entrance is equal to $(-W^E - 0.5W^P, 0.5H^P, 0.5W^P)$.

Similarly, we can also express the dimension of each Toukung element as the ratio to the pillar height. Architects may have axis-oval and axis-long arms of two different lengths for the parallel and perpendicular direction of the roof ridge. Therefore, we have two different control ratios for each arm as shown in the right of *Fig.* 3. There are six different control parameters: the width ratio of the connecting point and the length ratio of an Ang, an axis-oval arm perpendicular to the roof ridge, an axis-oval arm parallel to the roof ridge, an axis-long arm perpendicular to the roof ridge, and an axis-oval arm parallel to the roof ridge to the pillar height denoted as α^{CC}, α^{An}, α^{AOAA}, α^{AOAV}, α^{AOLA}, and α^{AOLV}. As shown in Fig. 15, we can calculate roof's width and depth ratios to the pillar height as $\alpha^{RW} = \alpha^{LW} + 2((N^{TL}-3)\alpha^{Ca} + \alpha^{An})$

#define α_{ij}^{XW}, α_{ij}^{XH}, α_{ij}^{XD}, α_{ij}^{Up}, α_{ij}^{Side}
#include M^{PB}, M^{P}, M^{L}, M^{FB}, M^{Q}, M^{Y}, T^{M}, T^{S}, R^{C}, R^{E}

Ω : $[P_{11}^{P}(\alpha_{ij}^{PH})]|P_{11}^{P}(\alpha_{ij}^{PH})$

$P_{ij}^{P}(\alpha)$: $i == 1 and j == 1$
$[+f(\alpha * \alpha_{ij}^{EW})$
$[\sim M^{PB}(\alpha * \alpha_{ij}^{PB} * dec[i,j,7]) \sim M^{p}(\alpha * \alpha_{ij}^{PW})]$
$[f(\alpha)[$
$|\sim M^{L}(\alpha * \alpha_{ij}^{EW})$
$f(-\alpha_{ij}^{LH} * dec[i,j,6]) \sim M^{FB}(\alpha * \alpha_{ij}^{EW} * dec[i,j,6])$
$f(-\alpha_{ij}^{LH} * dec[i,j,6]) \sim M^{L}(\alpha * \alpha_{ij}^{EW} * dec[i,j,6])$
$f(-\alpha_{ij}^{LH} * dec[i,j,4]) \sim M^{Q}(\alpha * \alpha_{ij}^{EW} * dec[i,j,4])$
$f(-\alpha_{ij}^{LH} * dec[i,j,5]) \sim M^{Y}(\alpha * \alpha_{ij}^{EW} * dec[i,j,5])$
$P_{i+1j}^{P}(\alpha * \alpha_{ij}^{Side})]]$
$[f(\alpha)P_{ij+1}^{P}(\alpha * \alpha_{ij}^{Up})]$

$P_{ij}^{P}(\alpha)$: $i! = 1 and j == 1$
$[|\sim M^{L}(\alpha * \alpha_{ij}^{EW} * 2)$
$[\ f(-\alpha_{ij}^{LH} * dec[i,j,6]) \sim M^{FB}(\alpha * \alpha_{ij}^{EW} * dec[i,j,6])$
$f(-\alpha_{ij}^{LH} * dec[i,j,6]) \sim M^{L}(\alpha * \alpha_{ij}^{EW} * dec[i,j,6])$
$f(-\alpha_{ij}^{LH} * dec[i,j,4]) \sim M^{Q}(\alpha * \alpha_{ij}^{EW} * dec[i,j,4])$
$f(-\alpha_{ij}^{LH} * dec[i,j,5]) \sim M^{Y}(\alpha * \alpha_{ij}^{EW} * dec[i,j,5])]$
$+f(\alpha * \alpha_{ij}^{EW})[P_{ij+1}^{P}(\alpha * \alpha_{ij}^{Up})|P_{ij+1}^{P}(\alpha * \alpha_{ij}^{Up})]$
$+f(\alpha * \alpha_{ij}^{EW})[\$ \sim M^{P}(\alpha * \alpha_{ij}^{PW})]P_{i+1j}^{P}(\alpha * \alpha_{ij}^{Side})$
$[\ f(-\alpha_{ij}^{LH} * dec[i,j,6]) \sim M^{FB}(\alpha * \alpha_{ij}^{EW} * dec[i,j,6])$
$f(-\alpha_{ij}^{LH} * dec[i,j,6]) \sim M^{L}(\alpha * \alpha_{ij}^{EW} * dec[i,j,6])$
$f(-\alpha_{ij}^{LH} * dec[i,j,4]) \sim M^{Q}(\alpha * \alpha_{ij}^{EW} * dec[i,j,4])$
$f(-\alpha_{ij}^{LH} * dec[i,j,5]) \sim M^{Y}(\alpha * \alpha_{ij}^{EW} * dec[i,j,5])]$
$]$

$P_{ij}^{P}(\alpha)$: $j! = 1$
$[f(\alpha)$
$-\sim M^{L}(\alpha * \alpha_{ij}^{EW})$
$f(\alpha_{ij}^{LH}) \sim T^{M}(\alpha * \alpha_{ij}^{EW} * \alpha_{ij}^{LW} * dec[i,j,0])$
$[f(\alpha * \alpha_{ij}^{EW}) \sim T^{S}(dec[i,j,1])]$
$[f(\alpha_{ij}^{TKH}) \sim R^{C}(\alpha * \alpha_{ij}^{EW} * \alpha_{ij}^{LW} * dec[i,j,2])$
$[+f(\alpha * \alpha_{ij}^{EW}) \sim R^{E}(dec[i,j,3])]]$
$P_{ij+1}^{P}(\alpha * \alpha_{ij}^{Up})]$
$[+f(\alpha * \alpha_{ij}^{EW}) \sim M^{P}(\alpha * \alpha_{ij}^{PW})]$

Figure 10: *This shows our derived parametric L-system production grammars for the exemplar Pailou. α_{ij}^{XW}, α_{ij}^{XH}, α_{ij}^{XD}, α_{ij}^{Up}, and α_{ij}^{Side} are user-specified parameters for each entrance where α_{ij}^{XW}, α_{ij}^{XH}, and α_{ij}^{XD} represents width, height, and depth scale ratio for the X component for the entrance ij, and X are all components for the entrance. M^{PB}, M^{P}, M^{L}, M^{FB}, M^{Q}, M^{Y}, T^{M}, T^{S}, R^{C}, R^{E} are the primitive of the pillar base, pillar, lintel, flower board, Queti, Yudan, middle Toukung, side Toukung, clipped roof, and eaves roof respectively.*

and $\alpha^{RD} = 2((N^{TL}-3)\alpha^{Ca} + \alpha^{An})$, respectively. Users can choose these parameters arbitrarily, but traditionally, there are only a few harmonic sets for various Pailous [Man92]. Due to the length limit, we show several possibly harmonic sets in our supplemental web†. This work uniformly sets 7/61, 10/12, 6/7, 1/1, 1/2, 4/5, 1/10, 3/2, 9/2, 33/20, 12/5, 37/20, and 67/20 as default for α^{PW}, α^{EW}, α^{LH}, α^{LW}, α^{Up}, α^{Side}, α^{Ca}, α^{CC}, α^{An}, α^{AOAA}, α^{AOAV}, α^{AOLA}, and α^{AOLV} of all entrances.

† web site: http://graphics.csie.ntust.edu.tw/pub/PailouModeling/

5. Results

Users can easily use our system to design and construct Pailous within a very short period of time. In order to easily denote different styles of Pailous, we categorize them based on its main topological and structural components. First, we can use the number of pillars and entrances in the first lateral level and the total number of roofs denoted as, denoted as N^{P}-N^{E}-N^{R}, which can briefly abstract Pailou's topologies, to denote their style. Traditionally, N^{P} is usually an even number and larger than two; N^{E} is equal to $N^{P}-1$; N^{R} is one of the following, $\{0, N^{E}, 2N^{E}+1, 3N^{E}\}$. However, our system is so flexible to design Pailous which do not follow the traditional rules. We can further separate the above into pillar and ridge

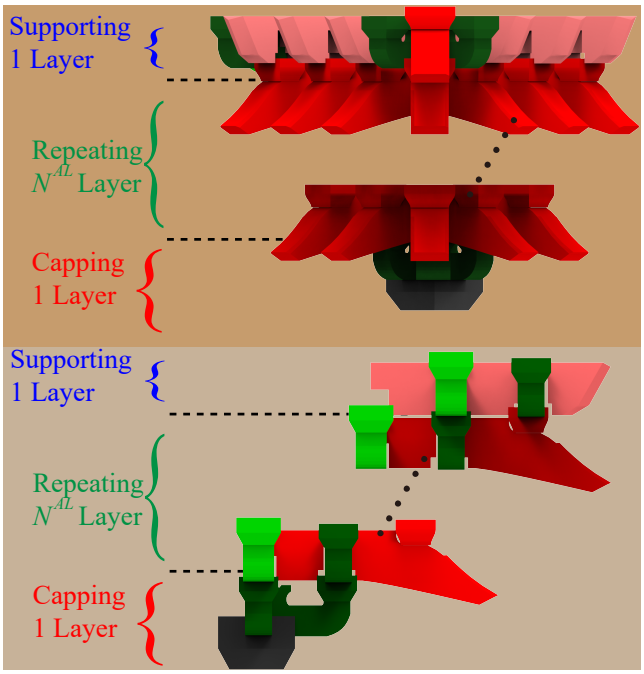

Figure 11: *Generally, we can procedurally decompose a middle Toukung (Bottom) and a side Toukung (Top) into three parts: capping, repeating, and supporting where the capping initiates the construction, the repeating repeats the fundamental structures to increase the level and height, and the supporting directly contacts and supports the roof.*

types based on the existence of eaves corners, B^E. Roofs of the pillar type have no eaves corner while their pillar tops are higher than the ridge of the main roof. The ridge type generally has roofs with eaves corners and Toukungs. Our system can easily construct various Toukungs and Pailous. Additionally, we have also conducted a usability test of modeling a desired Toukung and Pailou to evaluate its usefulness. Due to length limitation, we list relative parameters and detailed statistics in the supplemental web[†].

5.1. Toukung Modeling

Toukungs are important Pailou components. Users can easily select the number of levels, and our system can easily generate the corresponding 3D Toukungs with the provided elements using our L-system Toukung grammars as shown in Fig. 16. Generally, the process is within a second for different levels, even when the number is large.

5.2. Pailou Modeling

Users can easily select provided 3D structural components of the Toukungs, inscribed boards, lintels, flower boards, booklet pillars, Quetis, Yundans, Chuihuas and capital stones to build various Pailous as shown in Fig. 17, 18, and 19. At the same time, we also measure the time spent on building each of them and list the time along with each constructed result. It generally takes us

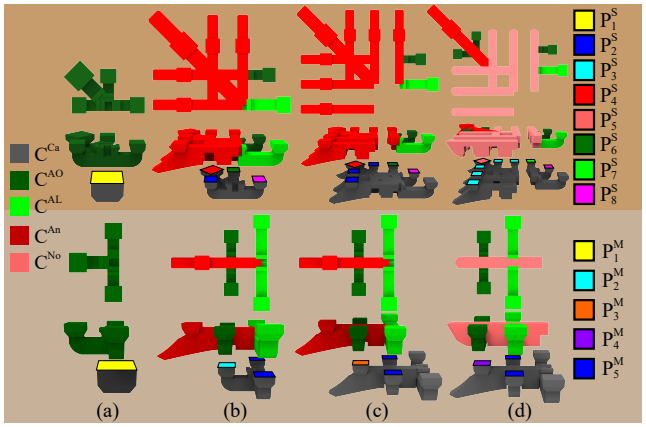

Figure 12: *From (a) to (d) demonstrates our L-system string formulation in different iterations for a 4-level middle and side Toukungs while the color of a connecting point marks a production rule such as yellow marks the production rule of P_1^S and P_1^M for the middle and side Toukungs.*

Model	Polygons	Modeling time (s)	File size (KB)
3-level Side	6.71e3	1	9.58e2
3-level Middle	4.33e3	1	8.58e2
5-level Side	1.75e4	1	2.35e3
5-level Middle	8.04e3	1	1.60e3
7-level Side	3.13e4	1	4.15e3
7-level Middle	1.22e4	1	2.35e3
18-level Side	1.87e5	1	2.49e4
18-level Middle	3.37e4	1	6.49e3

Table 2: *This shows the modeling statistics of the constructed Toukungs including the number of polygons, the time required to model, and the file size while using the default parameters.*

about a minute to build a simple Pailou and about 2 minutes to construct a moderate one while to design 6-5-15 takes us about 5 minutes which is the longest one. Similarly, it would take us about a hour to build a simple one with Maya shape transformation with the same provided component set. Since Pailous are commonly seen architectures in China, we imiate several collected examples as shown in Fig. 17 and 18. Additionally, our system can easily design imagined Pailous as shown in Fig. 19. These Pailous are intriguing. Especially, those Pailous contain multiple layers of roofs such as the 1-4-4, and 3-4-4 ridge-typed Pailous. After construction, we can use them to populate scenes in 3D Max while using VRay for rendering as shown in Fig. 1.

5.3. User Study

In order to understand the performance of our system, we have conducted a user study to evaluate its efficiency enhancement while comparing to a modeling tool, Maya. Since our framework aims for novices to interactively and intuitively design and construct various Pailous, we first invited 30 amateur participants for our studies. They are students and workers from National Taiwan University of

#define N^{TL}, α^{Ca}, α^{CC}, α^{AOAA}, α^{AOLA}, α^{AOAV}, α^{An}
#include M^{An}, M^{Ca}, M^{No}, M^{AO}, M^{AL}

$$
\begin{array}{lll}
\Omega & : & \sim M^{Ca}(\alpha^{Ca})f(\alpha^{Ca})P_1^M(1) \\
P_1^M(\beta) & : & [+\sim M^{AO}(\alpha^{AOAA})f(\alpha^{Ca})P_5^M][-\sim M^{AO}(\alpha^{AOAA})f(\alpha^{Ca})P_5^M] \\
 & & [\sim M^{AO}(\alpha^{AOAV})f(\alpha^{Ca})P_2^M(\beta+1)][|\sim M^{AO}(\alpha^{AOAV})f(\alpha^{Ca})P_2^M(\beta+1)] \\
P_2^M(\beta) & : \quad \beta < N^{TL}-1 & [+\sim M^{AO}(\alpha^{AOAA})f(\alpha^{Ca})P_5^M][-\sim M^{AO}(\alpha^{AOAA})f(\alpha^{Ca})P_5^M] \\
 & & [\sim M^{An}(\alpha^{An})f(\alpha^{Ca})\&f(\alpha^{CC})\wedge P_3^M(\beta+1)] \\
P_2^M(\beta) & : \quad else & \sim M^{No}(\alpha^{An})\&f(\alpha^{CC})[+\sim M^{AO}(\alpha^{AOAA})][-\sim M^{AO}(\alpha^{AOAA})] \\
P_3^M(\beta) & : \quad \beta < N^{TL}-1 & [+\sim M^{AO}(\alpha^{AOAA})f(\alpha^{Ca})P_5^M][-\sim M^{AO}(\alpha^{AOAA})f(\alpha^{Ca})P_5^M] \\
 & & [\sim M^{An}(\alpha^{An})f(\alpha^{Ca})\&f(\alpha^{CC})\wedge P_3^M(\beta+1)] \\
P_3^M(\beta) & : \quad else & [+\sim M^{AO}(\alpha^{AOAA})f(\alpha^{Ca})P_5^M][-\sim M^{AO}(\alpha^{AOAA})f(\alpha^{Ca})P_5^M] \\
 & & [\sim M^{An}(\alpha^{An})f(\alpha^{Ca})P_4^M] \\
P_4^M & : & \sim M^{No}(\alpha^{An})\&f(\alpha^{CC})[+\sim M^{AO}(\alpha^{AOAA})][-\sim M^{AO}(\alpha^{AOAA})] \\
P_5^M & : & \sim M^{AL}(\alpha^{AOLA})
\end{array}
$$

Figure 13: *This shows our derived parametric L-system production grammars for the middle Toukungs. N^{TL} is the number of production iterations and we set it to be 5 by default. α^{Ca}, α^{CC}, α^{AOAA}, α^{AOLA}, α^{AOAV}, α^{AOLV}, and α^{An} are user-specified parameters. M^{An}, M^{Ca}, M^{No}, M^{AO}, and M^{AL} are the primitive of Ang, Cap, Nose, axial-oval arm and axial-long arm respectively.*

#define N^{TL}, α^{Ca}, α^{CC}, α^{AOAA}, α^{AOLA}, α^{AOAV}, α^{AOLV}, α^{An}
#include M^{An}, M^{Ca}, M^{No}, M^{AO}, M^{AL}

$$
\begin{array}{lll}
\Omega & : & \sim M^{Ca}(\alpha^{Ca})f(\alpha^{Ca})P_1^S(1) \\
P_1^S(\beta) & : & [+\sim M^{AO}(\alpha^{AOAA})f(\alpha^{Ca})P_2^S][-\sim M^{AO}(\alpha^{AOAA})f(\alpha^{Ca})P_8^S] \\
 & & [\sim M^{AO}(\alpha^{AOAV})f(\alpha^{Ca})P_6^S(\beta+1)][|\sim M^{AO}(\alpha^{AOAV})f(\alpha^{Ca})P_6^S(\beta+1)] \\
 & & [+(45)\sim M^{AO}(\alpha^{AOAV})f(\alpha^{Ca})P_4^S(\beta+1)][|-(45)\sim M^{AO}(\alpha^{AOAV})f(\alpha^{Ca})P_4^S(\beta+1)] \\
P_2^S(\beta) & : \quad \beta < N^{TL}-1 & [\sim M^{An}(\alpha^{An})f(\alpha^{Ca})\&f(\alpha^{CC})\wedge P_2^S(\beta+1)] \\
P_2^S(\beta) & : \quad else & [\sim M^{An}(\alpha^{An})f(\alpha^{Ca})P_3^S] \\
P_3^S & : & \sim M^{No}(\alpha^{An}) \\
P_4^S(\beta) & : \quad \beta < N^{TL}-1 & [+(45)\sim M^{An}(\alpha^{An})f(\alpha^{Ca})\&f(\alpha^{CC})\wedge P_2^S(\beta+1)][-(45)\sim M^{An}(\alpha^{An})f(\alpha^{Ca})\&f(\alpha^{CC})\wedge P_2^S(\beta+1)] \\
 & & [\sim M^{An}(\alpha^{An})f(\alpha^{Ca})\&f(\alpha^{CC})\wedge P_4^S(\beta+1)] \\
P_4^S(\beta) & : \quad else & [+(45)\sim M^{An}(\alpha^{An})f(\alpha^{Ca})P_3^S][-(45)\sim M^{An}(\alpha^{An})f(\alpha^{Ca})P_3^S] \\
 & & [\sim M^{An}(\alpha^{An})f(\alpha^{Ca})\&f(\alpha^{CC})\wedge P_5^S] \\
P_5^S & : & \sim M^{An}(\alpha^{An})[+(45)\sim M^{AO}(\alpha^{AOAA})][-(45)\sim M^{AO}(\alpha^{AOAV})] \\
P_6^S(\beta) & : \quad \beta < N^{TL}-1 & [\sim M^{An}(\alpha^{An})f(\alpha^{Ca})\&f(\alpha^{CC})\wedge P_6^S(\beta+1)][-\sim M^{AO}(\alpha^{AOAA})f(\alpha^{Ca})P_8^S] \\
P_6^S(\beta) & : \quad else & [\sim M^{An}(\alpha^{An})f(\alpha^{Ca})P_7^S][-\sim M^{AO}(\alpha^{AOAA})f(\alpha^{Ca})P_8^S] \\
P_7^S & : & \sim M^{No}(\alpha^{An})\&f(\alpha^{CC})-\sim M^{AO}(\alpha^{AOAA}) \\
P_8^S & : & \sim M^{AL}(\alpha^{AOLA})
\end{array}
$$

Figure 14: *This shows our derived parametric L-system production grammars for the side Toukungs. N^{TL} is the number of production iterations and we set it to be 5 by default. α^{Ca}, α^{CC}, α^{AOAA}, α^{AOLA}, α^{AOAV}, α^{AOLV}, and α^{An} are user-specified parameters. M^{An}, M^{Ca}, M^{No}, M^{AO}, and M^{AL} are the primitive of Ang, Cap, Nose, axial-oval arm and axial-long arm respectively.*

Science and Technology. They have normal or corrected normal vision. Their ages range from 21 to 28 years old with a mean of 22.75, and the participants comprised 8 females and 22 males. All are voluntary. In addition, all of them have at least one years of experiences using Maya and are familiar with the tools provided by Maya. The study was conducted under a computer with Nvidia GTX 760, Intel i7 3820 and 8 GB main memory and the specialist study was under a laptop with Nvidia GTX 740, Intel i5 4200U and 8 GB main memory. All studies were with an AOC i2757fm monitor of a resolution of 1920×1080, brightness of 250 cd/m^2 and refresh rate of 60 Hz. Our framework aims at intuitively design Toukungs and various Pailous. Therefore, we chose a 3-level Middle-Toukung and a simple Pailou as shown in Fig. 20(a) and (c). While providing a set of structural Pailou components, each participant respectively used Maya shape manipulation to assemble them and used our icon-based framework with the same set to create the desired 3D models. The user study was conducted as follows: 1) Participants were asked to have a five-minute training section in order to get familiar with our framework. 2) The instructor gave instructions including the order of the tools and the procedural de-

Model	Polygons	Modeling time (s)	File size (KB)	ω	α_{ij}^{PW}	α_{ij}^{EW}	α_{ij}^{LH}	α_{ij}^{LW}	α_{ij}^{Up}	α_{ij}^{Side}
2-1-2 (Fig.17-1)	2.23e5	45	4.59e3	9	7/61	10/12	6/7	1	0.8	-
6-5-5 (Fig.17-2)	1.15e6	140	1.50e5	9	7/61	10/12	6/7	1	0.8	1
4-3-3 (Fig.17-3)	8.94e5	92	1.52e4	9	7/61	10/12	6/7	1	0.8	1
2-1-3 (Fig.17-4)	6.88e5	87	1.38e4	9	7/61	10/12	6/7	1	0.8	0.8
2-1-3 (Fig.18-1)	1.07e6	70	2.11e4	9	7/61	20/12	6/7	1	0.3	-
4-3-7 (Fig.18-2)	2.11e6	150	4.28e4	9	7/61	10/12	6/7	1.1	0.3	0.8
6-5-11 (Fig.18-3)	1.84e6	200	3.72e4	9	7/61	10/12	6/7	1.1	0.3	1
2-1-1 (Fig.18-4)	6.86e5	39	1.36e4	9	7/61	10/12	6/7	1	-	-
4-3-3 (Fig.18-5)	8.71e5	80	3.36e4	9	7/61	10/12	6/7	1	-	1
6-5-5 (Fig.18-6)	2.48e6	123	4.93e4	9	7/61	10/12	6/7	1	-	1
2-1-4 (Fig.19-1)	7.74e5	110	1.54e4	9	7/61	10/12	6/7	1	0.3	0.4
4-3-4 (Fig.19-2)	9.15e5	106	1.82e4	9	7/61	10/12	6/7	1	0.3	1
4-3-4 (Fig.19-3)	1.23e6	113	2.45e4	9	7/61	10/12	6/7	1	0.3	0.4
6-5-15 (Fig.19-4)	2.51e6	297	5.00e4	9	7/61	10/12	6/7	1.1	0.3	1
8-7-7 (Fig.19-5)	1.78e6	182	3.52e4	10	7/61	10/12	6/7	1	-	1

Table 3: *This shows the modeling statistics of the constructed model including the number of polygons, the time required to model, and the file size along with its important modeling parameters. We have applied the same entrance-based scale ratios to all entrances.*

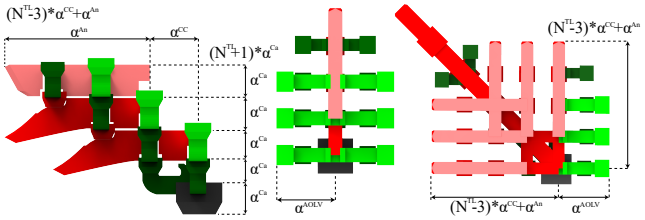

Figure 15: *In the left, each level contributes α^{Ca} in height, and thus, it is total height is $(N^{TL}+1)\alpha^{Ca}$ as shown in the left while each extra level after 3 contributes α^{CC} in depth, and thus, it is total depth is $(N^{TL}-3)\alpha^{CC}+\alpha^{An}$. The width is α^{AOLV}. Similarly, the side extension is $(N^{TL}-3)\alpha^{CC}+\alpha^{An}$.*

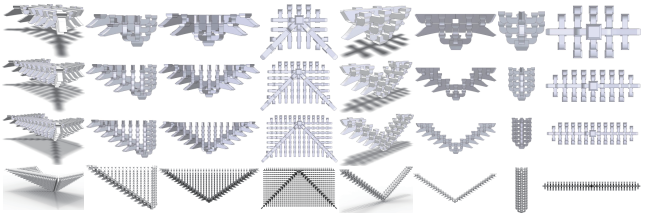

Figure 16: *This shows the construction results of the 3-, 5-, 7-level, and 18-level Toukungs. The left are side Toukungs and the right are middle Toukungs. From left to right in each set are the perspective, front, side, and bottom views of the corresponding Toukung.*

tails. 3) Participants were asked to create the destined 3D model with the first chosen tool. 4) Participants were asked to create the destined 3D model with the second chosen tool. 5) Repeat 3 and 4 until finishing assembling the 3D Toukungs and Pailou. The order of each model was predetermined, and the order of the tools was chosen in a counter-balance manner to avoid the order bias. During the study, we recorded how much time each participant spent on building each 3D model. Fig. 20(b) and (d) show the means and standard deviations of the operation time for Toukung and Pailou. The study aims at evaluating whether our tool can improve the

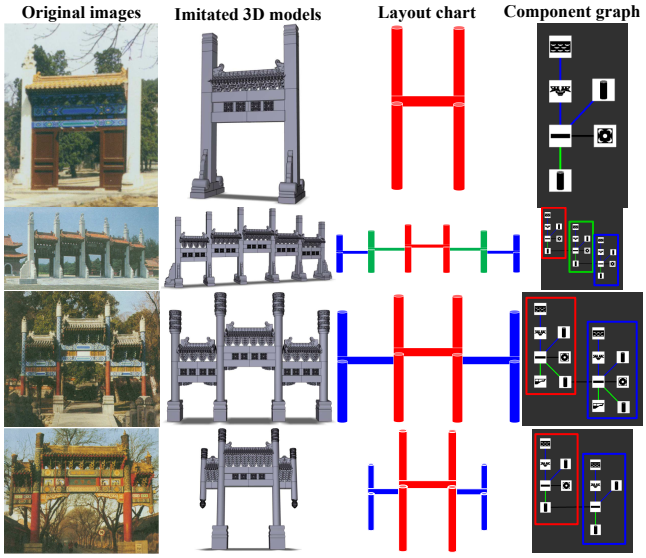

Figure 17: *Users can easily imitate existing 2-1-2, 6-5-5, 4-3-3, and 2-1-3 pillar-typed Pailous in the first column by selecting component icons to build their representative component graphs in the fourth column while our system extracts their structural layout charts in the third column to interpret its L-system grammars for their equivalent 3D models in the second column. Furthermore, it takes us 45, 140, 92, and 87 s to build these four with our system, respectively.*

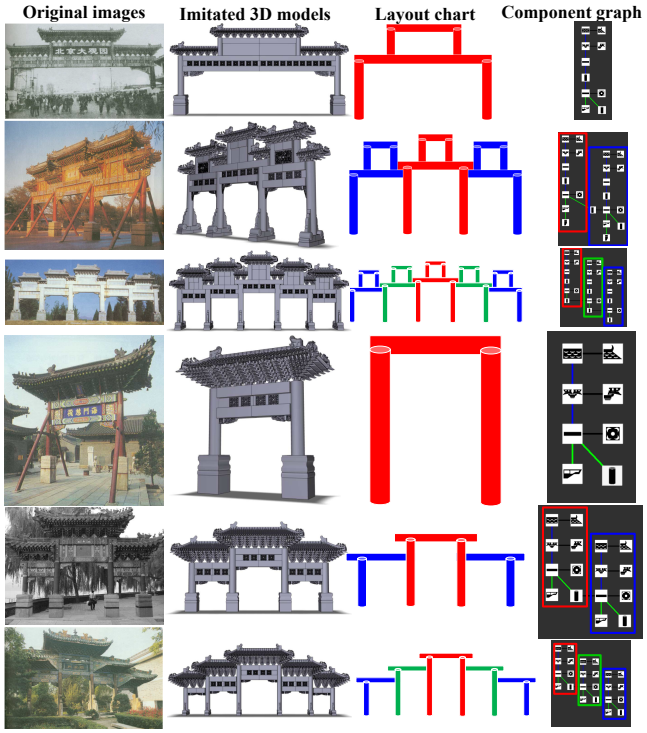

Figure 18: *Users can easily imitate existing* 2-1-3, 4-3-7, 6-5-11, 2-1-1, 4-3-3, *and* 6-5-5 *ridge-typed Pailous in the first column by selecting component icons to build their representative component graphs in the fourth column while our system extracts their structural layout charts in the third column to interpret its L-system grammars for their equivalent 3D models in the second column. Furthermore, it takes us* 70, 150, 200, 39, 80, *and* 123 *s to build these six with our system, respectively.*

	Time(s.)	
	Mean	Std.
Toukung(Maya)	699.6	299.9
Toukung(Ours)	4.05	1.64
Pailou(Maya)	1953	725.9
Pailou(Ours)	281.1	130.1

Table 4: *This shows the means and standard deviations of the time in the unit of seconds to build the desired Toukung and Pailou in the usefulness study of the proposed tool.*

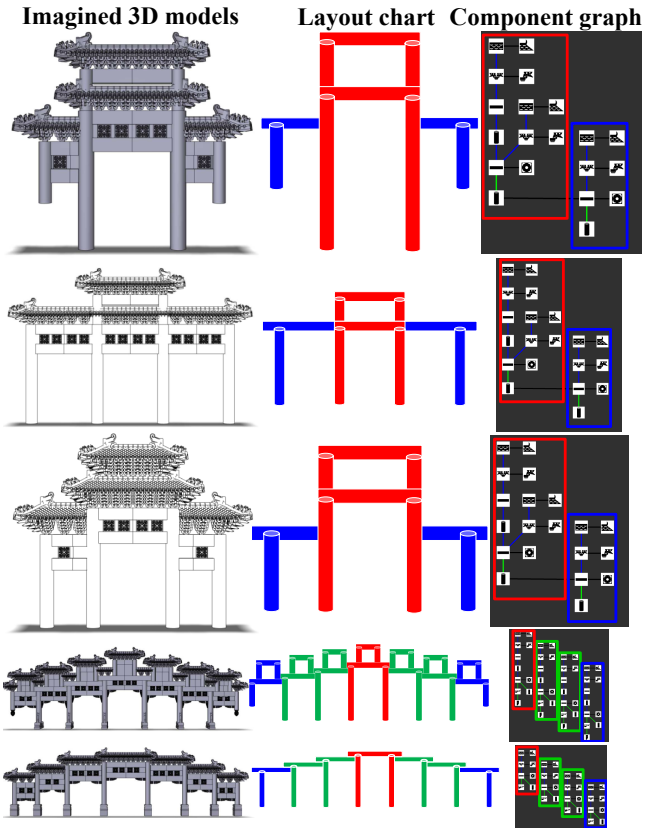

Figure 19: *Users can easily design their own* 2-1-4, 4-3-4, 4-3-4, 6-5-15, *and* 8-7-7 *ridge-typed Pailous by selecting component icons to build their representative component graphs in the third column while our system extracts their structural layout charts in the second column to interpret its L-system grammars for their equivalent 3D models in the first column. Furthermore, it takes us* 110, 106, 113, 297, *and* 182 *s to build these five with our system, respectively.*

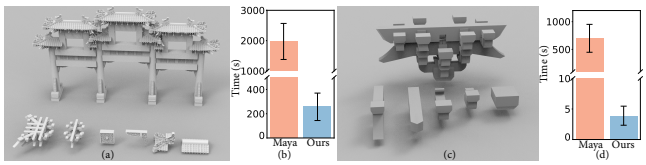

Figure 20: *(a) and (c) show the Toukung and Pailou used for our user study respectively while The top in both cases are the desired results, and the bottom are the provided 3D components. (b) and (d) show the means and standard deviations of the time in the unit of seconds to build the desired Toukung and Pailou in the usefulness study of the proposed tool.*

modeling efficiency. Differences among the participants who have different genders and ages which are called nuisance variables may make a significant contribution to error variances and thereby affect the judgement. Therefore, we used a randomized block factorial design (RBFD) [Kir82] to employ a blocking procedure to assign the levels of nuisance variations randomly to the experimental units

for distribution of known and unsuspected variation sources among the units over the entire experiment in order to avoid the affection of just one or a limited number of affected factors and compute their $F_{1,116} = 406.5$ ($p < 10^{-5}$) for spending time according to Eq.9.4-1 of the book written by Kirk [Kir82]. Therefore, the tool type is a significant factor for the modeling time. Based on the averaging operation time, the proposed tool does enhance the modeling efficiency.

6. Conclusion

This work designs an interactive icon-graph-based modeling system derived from Pailous' structural and topological analysis. Users can easily and intuitively create and manipulate a desired Pailou model by selecting component icons of a few selected parameters with topological recommendations to form a topological layout graph while our system can automatically interpret the graph based on our derived L-system grammars to construct 3D geometries. However, our system is not without limitations, and there are a few future research directions. First, currently, we focus on 1D ground-extension Pailous which act as line segments which are similar to walls because they are widely available and frequently seen. However, 2D ground-extension Pailous extend in another dimension to be rectangular for extra interior spaces for sheltering. They provide more design varieties by adding an extra constructed propagation direction, depth, while comparing to our designed algorithm. This introduces extra structural components and construction and interaction rules. We would like to append our graph and parametric L-system grammars for designing them. Second, currently, our interactive system only focuses on their topologies while using provided structural components. Since these components generally vary and have important impacts on visual appearance, we would like to extend our collected set of various 3D components for more varieties. Additionally, these main structural and subsidiary components, such as mystical animal patterns, generally vary, and users may want to design their own. Therefore, we would like to derive different design mechanisms for them. Third, our system currently interprets the icon graph as derived L-system grammars, but other procedural modeling methods such shape grammars should be able to achieve the same effect. It would be interesting to extend our system for other procedural modeling methods. Finally, fantastic paintings and fine carvings generally exist on the different componential surfaces, but they are currently smooth and plain to have dull shading. It would be interesting to have a painting and carving design mechanism for visually varieties.

References

[BWS10] BOKELOH M., WAND M., SEIDEL H.-P.: A connection between partial symmetry and inverse procedural modeling. *ACM Transactions on Graphics 29*, 4 (2010), 104:1–104:10. 2

[CKX*08] CHEN X., KANG S. B., XU Y.-Q., DORSEY J., SHUM H.-Y.: Sketching reality: Realistic interpretation of architectural designs. *ACM Trans. Graph. 27*, 2 (2008), 11:1–11:15. 3

[Dua02] DUARTE J.: *Malagueira Grammar - Towards a Tool for Customizing Alvaro Siza's Mass Houses at Malagueira*. PhD thesis, MIT School of Architecture and Planning, 2002. 2

[GK07] GANSTER B., KLEIN R.: An integrated framework for procedural modeling. In *Proceedings of the 23rd Spring Conference on Computer Graphics* (2007), SCCG '07, pp. 123–130. 2

[Hav05] HAVEMANN S.: *Generative Mesh Modeling*. PhD thesis, Informatik-Institute, 2005. 2

[HT13] HUANG C.-Y., TAI W.-K.: Ting tools: interactive and procedural modeling of chinese ting. *The Visual Computer 29*, 12 (2013), 1303–1318. 2, 3, 4, 7

[JTC09] JIANG N., TAN P., CHEONG L.-F.: Symmetric architecture modeling with a single image. *ACM Trans. Graph. 28*, 5 (2009), 113:1–113:8. 3

[Kir82] KIRK R.: *Experimental Design: Procedures for the Behaviors Sciences*, second ed. Brooks/Cole Publishing Company, 1982. 13, 14

[KPK10] KRECKLAU L., PAVIC D., KOBBELT L.: Generalized use of non-terminal symbols for procedural modeling. *Computer Graphics Forum 29*, 8 (2010), 2291–2303. 2

[Liu05] LIU T.-X.: *Procedural Chinese Palace Modeling*. Master's thesis, Dong Hua University, Taiwan, 2005. 2, 3

[LWW08] LIPP M., WONKA P., WIMMER M.: Interactive visual editing of grammars for procedural architecture. *ACM Trans. Graph. 27*, 3 (2008), 102:1–102:10. 2

[LYH10] LIU Y., Y. J., HUANG L.: Modeling complex architectures based on granular computing on ontology. *IEEE Trans. on Fuzzy Systems 18*, 3 (2010), 585–598. 3

[Man92] MAN Y.-N.: *Chinese Architecture: Paifang*. Jing Xiu Publishing Company, Taipei City, Taiwan, 1992. 1, 3, 4, 9

[MPB05] MARVIE J.-E., PERRET J., BOUATOUCH K.: The fl-system: A functional l-system for procedural geometric modeling. *The Vis. Comput. 21*, 5 (2005), 329–339. 2

[MVW*06] MÜLLER P., VEREENOOGHE T., WONKA P., PAAP I., VAN GOOL L.: Procedural 3d reconstruction of puuc buildings in xkipché. In *Proceedings of the 7th International Conference on Virtual Reality, Archaeology and Intelligent Cultural Heritage* (2006), VAST'06, pp. 139–146. 2

[MWH*06] MÜLLER P., WONKA P., HAEGLER S., ULMER A., VAN GOOL L.: Procedural modeling of buildings. *ACM Trans. Graph. 25*, 3 (2006), 614–623. 2

[NBA18] NISHIDA G., BOUSSEAU A., ALIAGA D. G.: Procedural modeling of a building from a single image. *Comput. Graph. Forum 37* (2018), 415–429. 3

[NGDA*16] NISHIDA G., GARCIA-DORADO I., ALIAGA D. G., BENES B., BOUSSEAU A.: Interactive sketching of urban procedural models. *ACM Trans. Graph. 35*, 4 (2016), 130:1–130:11. 3

[OSSJ09] OLSENA L., SAMAVATIA F. F., SOUSAA M. C., JORGE J. A.: Sketch-based modeling: A survey. computers & graphics. *Computers & Graphics 33*, 1 (2009), 85–103. 3

[Pat12] PATOW G.: User-friendly graph editing for procedural modeling of buildings. *IEEE Comput. Graph. Appl. 32*, 2 (2012), 66–75. 2

[PJM94] PRUSINKIEWICZ P., JAMES M., MĚCH R.: Synthetic topiary. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques* (1994), SIGGRAPH '94, pp. 351–358. 2

[PL90] PRUSINKIEWICZ P., LINDENMAYER A.: *The Algorithmic Beauty of Plants*. Springer-Verlag, Heidelberg, Berlin, Germany, 1990. 2, 5

[PM01] PARISH Y. I. H., MÜLLER P.: Procedural modeling of cities. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques* (2001), SIGGRAPH '01, pp. 301–308. 2

[PMKL01] PRUSINKIEWICZ P., MÜNDERMANN L., KARWOWSKI R., LANE B.: The use of positional information in the modeling of plants. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques* (2001), SIGGRAPH '01, pp. 289–300. 2

[RJT18] RITCHIE D., JOBALIA S., THOMAS A.: Example-based authoring of procedural modeling programs with structural and continuous variability. *Computer Graphics Forum 37*, 2 (2018), 401–413. 3

[RMBG*13] RUIZ-MONTIEL M., BONED J., GAVILANES J., JIMÉNEZ E., MANDOW L., PÉREZ-DE-LA-CRUZ J.: Design with shape grammars and reinforcement learning. *Adv. Eng. Inform. 27*, 2 (2013), 230–245. 3

[RMGH15] RITCHIE D., MILDENHALL B., GOODMAN N. D., HANRAHAN P.: Controlling procedural modeling programs with stochastically-ordered sequential monte carlo. *ACM Trans. Graph. 34*, 4 (2015), 105:1–105:11. 3

[SM15] SCHWARZ M., MÜLLER P.: Advanced procedural modeling of architecture. *ACM Trans. Graph. 34*, 4 (2015), 107:1–107:12. 2

[STBB14] SMELIK R. M., TUTENEL T., BIDARRA R., BENES B.: A survey on procedural modelling for virtual worlds. *Comput. Graph. Forum 33*, 6 (2014), 31–50. 2

[Teo09] TEOH S. T.: Generalized descriptions for the procedural modeling of ancient east asian buildings. In *Proceedings of the Fifth Eurographics Conference on Computational Aesthetics in Graphics, Visualization and Imaging* (2009), Computational Aesthetics'09, pp. 17–24. 2, 3

[TLL*11] TALTON J. O., LOU Y., LESSER S., DUKE J., MĚCH R., KOLTUN V.: Metropolis procedural modeling. *ACM Trans. Graph. 30*, 2 (2011), 11:1–11:14. 3

[WOD09] WHITING E., OCHSENDORF J., DURAND F.: Procedural modeling of structurally-sound masonry buildings. *ACM Trans. Graph. 28*, 5 (2009), 112:1–112:9. 3

[WWSR03] WONKA P., WIMMER M., SILLION F., RIBARSKY W.: Instant architecture. *ACM Trans. Graph. 22*, 3 (2003), 669–677. 2

[YCZY04] YONG L., CONGFU X., ZHIGENG P., YUNHE P.: Semantic modeling project: Building vernacular house of southeast china. In *Proceedings of the 2004 ACM SIGGRAPH International Conference on Virtual Reality Continuum and Its Applications in Industry* (2004), VRCAI '04, pp. 412–418. 2, 3

[ZXJ*13] ZHANG H., XU K., JIANG W., LIN J., COHEN-OR D., CHEN B.: Layered analysis of irregular facades via symmetry maximization. *ACM Trans. Graph. 32*, 4 (2013), 121:1–121:13. 2