Building Chinese Ancient Architectures in Seconds

Hua Liu, Qing Wang, Wei Hua , Dong Zhou, and Hujun Bao

State Key Lab of CAD&CG, Zhejiang University
{sun_day, qwang, huawei, zddong, bao}@cad.zju.edu.cn

Abstract. Chinese Ancient Architecture (CAA) has a remarkable feature that its construction follows a set of rigorous constraints. Based on this property, we present a procedural approach for fast CAA modeling. Knowledge of the architecture construction is formalized as a set of rules and the primary features of buildings are parameterized to control the modeling process. By this approach, it is efficient and convenient to generate regular architectures or a class of architectures with a similar style. A system is implemented based on this idea. More than one hundred different buildings have been generated by a rule set comprising of about 120 rules.

1 Introduction

Chinese Ancient Architecture (CAA) is well known for its gorgeous appearance and contraption. However, it is really tedious to model it with mainstream modeling software, since specific knowledge of CAA construction is required. More over, the construction of CAA is very complex, as illustrated in Figure1.

By exploring the intrinsic discipline of CAA, we present a procedural approach for fast modeling, which significantly simplifies the traditional modeling process. The knowledge of the architecture construction is formalized in a set of rules. The building process consists of iteratively interpretation of rules and user-specified style parameters. Using our system, even novices with little specific knowledge can also generate different kinds of CAA by simply setting a few control parameters. Figure 7 shows a typical scene which is generated in less than two seconds.

1.1 The Feature of Chinese Ancient Architectures

Chinese ancient architecture has a remarkable feature that both the architecture itself and its layout follow a set of rigorous construction constraints. Although different architectures with different grades, races, regions or eras, have their own features, from the view of structure, they follow a whole set of construction constraints which were illuminated in the book *Yinzaofashi*[13, 14].

As the fact that Chinese ancient architecture follows a set of rigorous construction constraints, we could use a set of modeling rules to formalize the construction constraints. Given some initial constraints, CAA models can be generated rapidly by procedural modeling approach based on these rules. Furthermore, using this modeling method, modeler needs little architecture knowledge.

1.2 Related Work

Our approach for modeling architectures is one of the procedural modeling techniques. L-System or parameter L-System, which is mainly used in plant modeling [1, 2, 3], is a predominant one among them. In [4], an extended L-System was used to generate streets of a city. Population density, height map and water map of a city were used as the input parameters that influenced the generation of the streets.

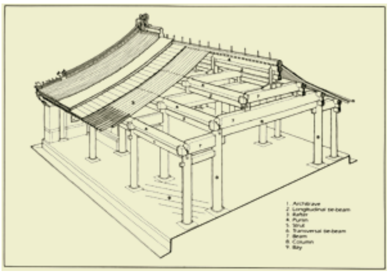

Fig. 1. The picture of an Chinese Ancient Architectures

Shape grammar or *parameterized shape grammar* [5], was a powerful analysis and design tool for architectures. Using this tool, modeling rules can be abstracted from the existing buildings [6, 7] and new architectures can be designed by a set of modeling rules [8, 9]. However, the rules written by the shape grammar are directly based on 2D/3D shapes. It is difficult to identify and represent the shape rules by computer. Selecting and deducing rules only can be executed by architects. As a consequence, it is not suitable for automatic modeling of architectures.

Split grammar [10] was a subset of the shape grammar. It describes the rules of shape splitting, which is used to produce a class of models. *Control grammar* was introduced to distribute the attributes of the modeling rules and selecting the next executed rule by matching attributes. The modeling process starts from a bounded simple shape, and then splits it recursively to form a complex shape with many details progressively.

Another technique in architecture modeling is Image-Based Modeling (IBM). The realistic models can be recovered from photographs [11, 12]. However, these methods need human interactions to identify the pairs of corresponding points. It decreases the modeling efficiency in practice. This method can be used to rebuild architectures, but is not designed to create new models without photographic input data.

1.3 Our Approach

The idea of our approach is from the shape grammar. As it is hard to describe the shape grammar by computer completely, we try to define a subset of it, which is named constructive grammar.

Constructive grammar (CG) is different from the split grammar (SG) in some aspects. The main difference between them is the modeling process. The SG starts from a bounded simple shape, and then splits it recursively to form a complex shape with many details, while the CG starts from an unbounded space and produces an architecture model by combining some kinds of basic shapes progressively. The process of the CG is like building toy bricks which we played in our childhood. Additionally, the SG emphasizes designing surface detail of architecture, while the CG focuses on building the whole structure of the architecture, including its exterior and interior geometry information. As illustrated in Figure 5, the result models built by our approach include not only the exterior geometry information, but also the interior structure information.

2 Constructive Grammar

Constructive grammar introduced in this paper is a specialized form of the shape grammar [4], which is designed to procedurally model buildings. Its utility stems from the fact that the restrictions of architectures have been carefully chosen so as to make automatic rule selection possible. The atomic object manipulated by the grammar is the component.

2.1 Constructive Grammar

- **Definition 1:** A *primitive* is a parameterized basic shape.
  Buildings always consist of a set of basic shapes, such as cubes, cylinders, prisms and so on. These basic shapes are called primitives. The geometry of a primitive is parameterized, which is used to represent the geometry of the component.
- **Definition 2:** A *component* is a basic building block of an architecture model. It is represented by a quad-tuple $< n, p, A, s >$, where n is the component name, which indicates its semantic meaning; p is a primitive; A is a finite set of attributes and s describes the state of component.
  The name n distinguishes the components from each other according to their functions. The primitive p indicates the geometry of the component. The state s is an enumerative variable that records the information of the component during a modeling process, which helps to control the modeling process.
- **Definition 3:** A *component list* is a set of components with a sequence, where the sequence indicates the building order of these components.
  A simplest component list is composed of only one component.
- **Definition 4:** *Constructive Grammar* is defined by $< M, F, R, S, \$ >$,where

  1. M is a finite set of components;
  2. F is a finite set of restrictions;
  3. R is a finite set of rules in the form $\alpha \rightarrow \beta$, where α is a component and β is a component list;
  4. S is a finite set of occupied space of a component list;
  5. $\$$ is an initial space in S;

The vocabulary of the constructive grammar is the set $V = \{t(m) \mid m \in M, t \in T\}$, where T is the set of transform e.g. rotation, translation and reflection. The restrictions in F are abstracted from a class of buildings and formalized as some Boolean functions.

R is a set of rules. Here, the rules are formalized as the productions in the extension L-System. All the rules are classified into two classes: *combining rules* and *conversion rules*, based on their functions. Combining rules are used to add new components and keep the existing ones, while conversion rules are used to replace the existing components with new ones. As illustrated in Figure 2, after a conversion rule C3 is used, a PILLARBOX is converted into a PILLAR. However, a combining rule B2 is used to create a PILLARBOX on the existing BASEBOX.

Each element s in S is the occupied space of a component list. It is a specific component, which is used to locate the corresponding component list in 3D space. In our approach, the bounding box of a component list is used to represent s and the geometry of s is a cube primitive. \$ is specified by user, determining the initial space which an architecture model occupied.

2.2 A Simple Example

An example of modeling a pillar, which is illustrated in Figure 2, is given in this section and the corresponding rules are shown in appendix A.

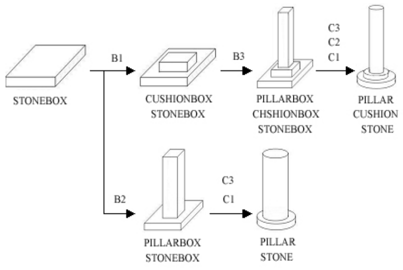

Fig. 2. The modeling process of a pillar. The strings above the arrow lines are the rule numbers of the executed rules

The symbol \$ determines the initial size and position of the model. The tokens BASEBOX, CUSHIONBOX and PILLARBOX, which belong to the set S, represent the space occupied by the corresponding component BASE, CUSHION and PILLAR. Rules B1, B2 and B3 are the combining rules to build the connectivity of components. Meanwhile, the rules C1, C2 and C3 are conversion rules to determine the actual form of the model.

The restrictions STATE == BEGIN and STATE == FINISHED both use the state information STATE of components as their parameters. The STATE of component BASEBOX indicates whether the BASEBOX has a PILLARBOX or CUSHIONBOX on it. The rules B1 and B2 will be selected and executed only when the STATE of component BASEBOX equals to BEGIN, which indicates that the BASEBOX has no PILLAEBOX or CUSHIONBOX on it. In this example, we only use the modeling state of components to indicate the restrictions. In other situations, position and direction of the components are also applied in restrictions.

The expression $1.0 - x$ in rule B1 and the value 0.4 in rule B1 are the probability of the rule for the probability comparison. x is a variable randomized between 0.0 and 1.0 during the modeling process.

3 A Modeling System for Chinese Ancient Architectures

In this section, we will show you how to build Chinese Ancient Architectures with the constructive grammar.

3.1 Abstract Modeling Rules

3.1.1 Local Pattern

The styles of CAAs are mainly showed in three aspects: Wall、Pillar and Roof. Each of them has its own pattern, which represents the basic structure variation in CAAs.

- Pillar pattern: To decide the building's pillar layout. As illustrated in Figure 3, it has four different patterns;
- Wall pattern: To decide the relations between walls and pillars of a building. As showed in Figure 4, three patterns are included;
- Roof pattern: To decide the building's roof style. CAAs have mainly six roof patterns and we adopted three of them in our system.

In our system, corresponding to each pattern, a subset of rules are created to describe it.

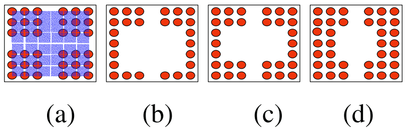

Fig. 3. It shows four pillar patterns, in which dot means pillar and box means the building body's room. In (a), the rectangle stands for ***KAIJIAN***. In this example the *AKR* is 6 and *AKC* is also 6

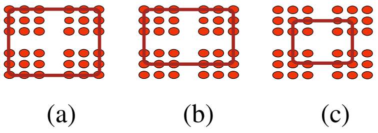

Fig. 4. It shows three wall patterns, in which the pillar pattern is as same as the one shown in figure 3(a). Dot means pillar, while line means wall

3.1.2 KAIJIAN

KAIJIAN is a professional glossary in CAAs. It can be regarded as the unit space in a building. In Figure 3, the area of the small rectangles are KAIJIANs. *The Amount of KAIJIAN in Row*(AKR) and *the Amount of KAIJIAN in Column* (AKC) decide the scale of the building.

3.1.3 The Generation of Rules

In our system, we took AKR, AKC and three local patterns as control parameters. Each parameter could choose three to five values. Using these control parameters, more than 100 different building models could be generated.

The constructive rules were created according to [13, 14]. The architecture elements such as door, pillar, baluster and so on, were all treated as components in the constructive grammar. And all the parameters of them were described as attributes. The bounding box of the component was written in rules as the element in the set S. For generating layout of buildings, the arrangement of buildings was also considered.

Totally, about 10 primitives, 30 components and 30 attributes were defined to create the rule set. It consists of about 120 modeling rules, in which 110 rules are for modeling buildings and the rest are for creating buildings' layout.

3.2 Implementation

The interpreting process of our system is similar to the extension L-system. The rules are interpreted by a string parser and a visualization parser. All the rules are managed by a hash-table, which accelerates rules selection. In the hash-table, the name of the component is used as the key to search rules. The rules with the same key are further organized to a list sorted by the rule's probability. The symmetric character of Chinese Ancient Architecture is exploited to accelerate the modeling process.

Textures are assigned to the corresponding components based on their semantics meanings. One look-up table is used to store the relations between textures and the corresponding components. The name of the component acts as the search key for finding the correct texture in texture mapping. In texture coordinates computing, the relation between the size of texture and the dimension of the corresponding component is considered.

3.3 Experimental Results

Our system was tested on a Windows platform with AMD AthlonXP 1800+ CPU,ATI Radeon 9100 graphics card and 512MB memory.

There were five control parameters in our test. By changing the value or the combination of these control parameters, more than one hundred kinds of constructions could be obtained. Additionally, architecture models of different styles could be generated by applying different textures. The Figures[5,6,7] show some models generated by our system (All the models were rendered in 3D studio MAX) .

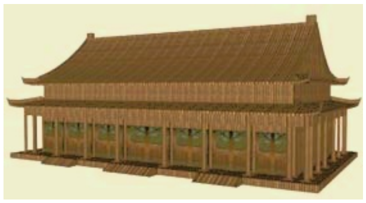
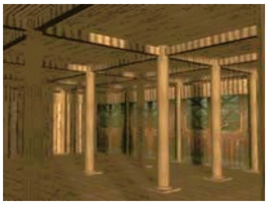

Fig. 5. Bamboo house. The control parameters of the model are 7AKR & 5AKC and the generating time is 0.381s. Left: the outside of the bamboo house; Right: the inside of the bamboo house

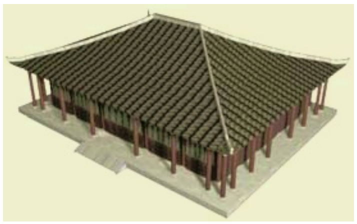
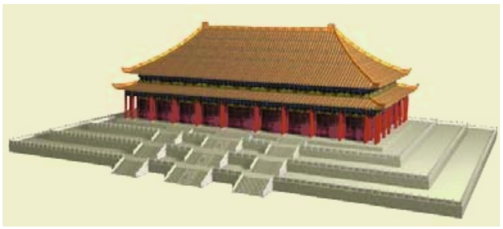

Fig. 6. Left: Civilian house. The control parameters of the model are 5AKR&5AKC and the generating time is 0.24s ; Right: Palace. The control parameters of the model are 9AKR & 5AKC and the generating time of it is 0.972s

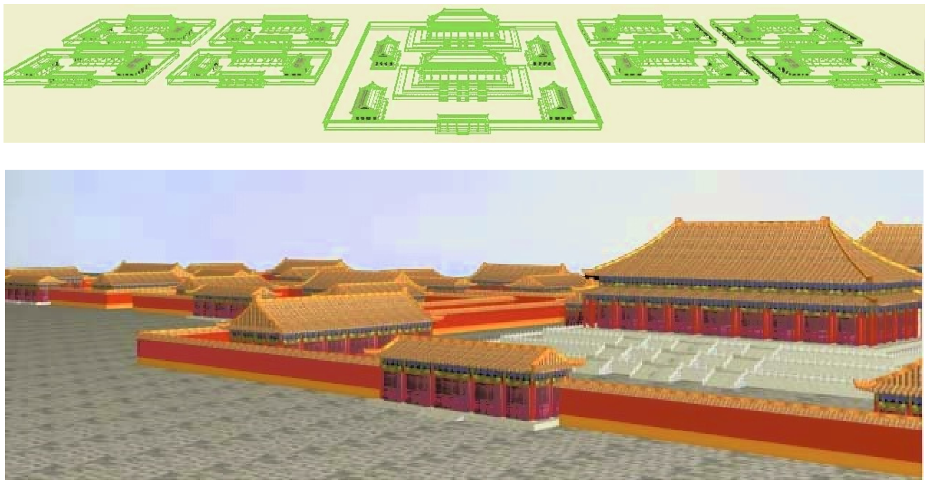

Fig. 7. The image shows a corner of the Forbidden City built by our system. The generating time of it is 1.932s

4 Conclusion and Future Work

In this paper, a rapid procedural modeling approach to build CAAs is represented, which is based on the constructive grammar. This approach is suitable for modeling regular architectures or a class of architectures with some common features, as well as for generating layouts which consist of buildings with a similar style. More than One hundred buildings were generated by about 110 modeling rules.

The most important step of our approach is to abstract the semantic rules from architectures. In order to do that, we must analyze the features of buildings and formalize them into semantic rules. Sometimes, it is not intuitive enough to acquire the rules. In the future, we will explore a method to abstract rules conveniently or even automatically by machine learning techniques. Additionally, a texture synthesis approach driven by semantic rules is considered to be integrated into the current system in the future.

Acknowledgements

We acknowledge Prof. Andrew I-kang Li, in the Department of Architecture of the Chinese University of Hongkang, who has offered us many references about architectures. We also acknowledge the contributions from Mr. Huang xiang, in the Architecture Department of Zhejiang University. This work is supported in part by National 973 Program of China (2002CB312102), National Natural Science Foundation of China (Grant No. 60203014, 60021201).

References

1. Przemyslaw.P, James.M and Měch.R.: Synthetic topiary. In Proceedings of ACM SIGGRAPH 1994, ACM Press, A.Glassner, Ed, pp. 351-358.
2. Przemyslaw.P, Lars Mündermann, Radoslaw Karwowski, and Brendan Lane: The use of positional information in the modeling of plants. In Proceedings of ACM SIGGRAPH 2001, ACM Press, pp. 289-300.
3. P. Prusinkiewicz and A. Lindenmayer: The algorithmic beauty of plants. Springer-Verlag, NewYork,Inc. 1990. New York,USA.
4. Yoav I.H.Parish and Pascal Müller: Procedural modeling of cities. In Proceedings of ACM SIGGRAPH 2001, ACM Press, E.Fiume,Ed, pp. 301-308.
5. G.Stiny: Introduction to shape and shape grammars. Environment and Planning B 7 (1980), Pion Ltd, London, England. pp. 349- 351.
6. G.Stiny: Ice-ray: a note on Chinese lattice designs. Environment and Planning B 4 (1977), Pion Ltd, London, England. pp.89-98.
7. G.Stiny and W. J. Mitchell: The Palladian grammar. Environment and Planning B 5 (1978), Pion Ltd, London, England. pp.5-18.
8. T.W.Knight: Designing with Grammars. In CAAD futures'91 Ed G. N. Schmitt, Computer-Aided Architectural Design (Verlag Viewag, Weisbaden, 1992), pp.33-48.
9. T.W.Knight: Color grammars: designing with lines and colors. Environment and Planning B: Planning and Design 16 (1989), Pion Ltd, London, England. pp.417-449.
10. Peter Wonka, Michael Wimmer, Francois Sillion, and William Ribarsky: Instant Architecture. ACM Transactions on Graphics, ACM Press, volume 22. number 3. pp. 669-677. July 2003
11. Paul E. Debevec, Camillo J. Taylor and Jitendra Malik: Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In Proceedings of ACM SIGGRAPH 96, ACM Press, H. Rushmeier, Ed., pp.11-20.
12. Paul Debevec, Yizhou Yu and George Borshukov: Efficient View-Dependent Image-Based Rendering with Projective Texture-Mapping. Proceedings of 9th Eurographics Workshop on Rendering, Vienna, Austria, June 1998, pp.105-116. UC Berkeley Technical Report, UCB//CSD-98-1003
13. Liang Sicheng: Yingzao fashi zhushi [The annotated Yingzao fashi]. Beijing:Zhongguo jianzhu gongye,1983.
14. Liang Sicheng: Liang Sicheng quan ji. Chapter 4,6,7, Beijing: Zhongguo jianzhu gongye,2001.