Available online at www.sciencedirect.com
ScienceDirect
Computers & Graphics 30 (2006) 800–814
COMPUTERS & GRAPHICS
www.elsevier.com/locate/cag

Technical Section

Semantic modeling for ancient architecture of digital heritage

Yong Liu[a,b,*], Congfu Xu[a,b], Zhigeng Pan[a,b], Yunhe Pan[a,b]

[a]State Key Laboratory of CAD & CG, Zhejiang University, P.O. Box 1420, Hangzhou 310027, China
[b]College of Computer Science, Zhejiang University, China

Abstract

In this paper, we propose a system for semantic modeling of southeast China vernacular urban. This system converts the basic modeling components of geometry units such as points, lines, and triangles, etc., into the semantic components such as streets, blocks, patches, and houses, etc. To implement the urban modeling system semantically, an XML based description and DTD [ISO/TC 211/WG 4/PT 19136 Geographic information—Geography Markup Language (GML) https://portal.opengeospatial.org/files/?artifact_id = 7174] based verification technique is used to control the generation process of urban, and this DTD based verification technique can avoid the disadvantage of the grammar system and can ensure the coherent architecture styles of ancient vernacular houses. It has been applied to preserve the digital heritage of southeast China vernacular architecture.
© 2006 Published by Elsevier Ltd.

Keywords: Semantic; Modeling; Vernacular house; Building design; Architecture; Grammars

1. Introduction

Digital heritage is a hot research topic in the past few years. The objective is to find new ways of representing, analyzing, manipulating and managing different kinds of digital cultural objects from different media sources. Nowadays, special attention has been paid to surrogates of fragile physical objects. Modeling of ancient architecture such as buildings is an important step in a digital heritage system.

This paper presents a system for the semantic modeling of vernacular buildings of southeast China, where many ancient towns, streets or blocks are well reserved in the riverside country. The motivation of our work is to build the digital heritage to reserve the architecture of southeast China, such as He Fang Street (Fig. 1.) in Hangzhou (the Capital of South Song dynasty from 1217∼1279, and now capital of Zhejiang province), which was a large commercial street in the past several hundred years; and Xitang, a small beautiful town located at Jiashan country, Zhejiang province, reserved blocks of residential houses of the ancient time; and the same ancient towns, Zhou Zhuang, Tong Li, etc. (all located at Jiangsu province, southeast China).

The basic concept of semantic modeling project is that it promotes the object from the graphical basic units (points, lines, triangles, curves etc.) to semantic elements (for example, in a city, there are streets, districts, houses, etc.). In this way, the users can pay more attention to their special implementation, and those graphical details are encapsulated in the semantic elements.

When modeling ancient architectures in computers, there are many difficulties. One is that the shapes of vernacular houses are much more complex than other modern buildings. The vernacular houses include two different kinds of most important styles: residential houses, and commercial houses. These two kinds of houses are combined to form the ancient cities. And how to arrange the above two types is another difficulty. So the common requirements of the modeling system are: (1) most of the models need to be developed automatically; (2) the buildings and objects in the scene should be able to follow a certain style and be distinguished from each other at several aspects, such as shape and appearance, etc.; (3) the model complexity should be traded off between efficiency and verisimilitude.

*Corresponding author. State Key Laboratory of CAD & CG, Zhejiang University, P.O. Box 1420, Hangzhou 310027, China.
Tel.: +86 571 87952308; fax: +86 571 87951947.
E-mail address: cckaffe@yahoo.com.cn (Y. Liu).

0097-8493/$ - see front matter © 2006 Published by Elsevier Ltd.
doi:10.1016/j.cag.2006.07.008

Fig. 1. Photograph of He Fang Street in Hangzhou, southeast China.

In our semantic modeling project, we divide the urban into components, such as street, house, gate, window, wall and roof, which constitute the semantic portions in urban modeling and are described by XML. To generate and control the urban model automatically, a production-based (similar to the L-system [1]) modeling method is used, which controls the detailed generating process of the vernacular house. For the models generated by those production rules to be random and unpredictable, a DTD based verification technique is implemented to maintain the generated models in a coherent style. As mentioned in the modeling generation, the grammar generation systems have achieved impressive result. It is first implemented in the plant modeling [1–4], and then recently has been used in the large-scale city building [5,6]. The grammar generation systems cause the generated cities lack of authority in urban design, so more attention should be paid on the urban design theory applied in city modeling [7–9].

The fundamental difference between our approach and previous work [6] is that the generation process has been promoted to a higher mankind's semantic layer. That means the input for urban generation is no longer a series of numeric parameters, instead, it has been changed to a segment description language, which can decrease the modeling complexity and modeling workload to those designers significantly.

Main contributions of this paper include

- We increase the modeling level from numeric parameters to language description while decreasing the complexity and workload to designers.
- We improve the rule grammar based system, adding numeric parameters to these rules, which can generate more accurate models.
- We divide the modeling into the XML based semantic components description and DTD based topology and combination verification, which makes the system easy to control.

The rest of the paper is organized as follows: Section 2 discusses some applications of the digital heritage. Section 3 gives the overview of the system architecture, Section 4 describes the semantic elements definitions and Section 5 discusses some semantic control on topology and combination of the southeast vernacular houses modeling system. Then we present some examples of modeling in Section 6. Finally we conclude Section 7 with discussion and directions for future work.

2. Related works for preservation of digital heritage

With the development of computer technology, more and more new technologies have been implemented in the digital heritage. There are three kinds of common computer technologies involved in the digital heritage, they are: digital input method; IBR (IBM) technique; and 3D Modeling technique.

2.1. Digital input method

This method mainly depends on the digital input hardware, such as the digital video, digital camera, three-dimensional scanner, laser scanner and CT scanner, etc. It is mostly used in the statues [10] and solid model heritage reserving. Additionally, the output of laser scanner devices is a very large date set of points, such as the Livio De Luca's work [11], which should be processed by some complex methods and high performance hardware. However, the cost of hardware (especially the scanners) restricts the broad implementation of this method.

2.2. IBR and IBM techniques

IBR [12,13] technique is to process several images to obtain the new images that are different from the original ones. The main usage of IBR technique implemented in digital heritage is the panorama [14], and it can build the whole view by combining multiple images of the heritage. IBM [15,16] is also very useful in digital heritage; it rebuilds the model from the images. While the panorama view only restricts in those input images, these two techniques still cannot offer visitors a real contact with those heritages.

2.3. 3D modeling techniques

The computer modeling techniques have been researched for a long time. Normally there are two categories, manual modeling and automated modeling.

In manual modeling, users adopt commercial modeling software such as AutoCAD, 3D studio max, Maya, etc., to design and render the heritage modeling. Although the manual modeling can generate the most accurate and complicated models (see Fig. 2, the VR scene generated by 3D Studio Max), it is a time-consuming work. A typical manual modeling case is given by ChiuShui Chan, etc. [17], building the model of traditional Chinese architecture. and a higher level of manual modeling can use the VR tools, for example Vega [18], to generate models. And Vega provides a series of interactive tools for building the models and a rendering library implemented by C Language. So we can use Vega's interactive tools to build the models and render those models by Vega's rendering library in our programs.

The automated modeling technique can generate the models with only several parameters, whose detailed generation is controlled by programs. Parish [5,19] introduced a stochastic, parametric L-system to generate the geometry for the buildings, in which, rules were set up to control the transformation, scaling, extrusion and branching of the building geometry. Another approach for architecture auto-modeling divides houses into the basic units, such as roof, wall, window, gate, etc. and combining them to generate new buildings [20,21].

Fig. 2. Walkthrough scene generated by modeling software.

From the above description, most of these modeling techniques are all aimed to those professional modeling experts, who should be familiar with the techniques of computer graphics when they are using these modeling systems. In most of special applications, such as movie production, city planning, etc., the users should consider both the computer graphics techniques and special domain knowledge in practice, and this will restrict the implementation of the modeling systems. Therefore, our semantic project, it will focus on the improvement in the automated grammar modeling system and the description from semantic layer.

3. System architecture

Before presenting the implementation of semantic modeling, let us review the traditional modeling process, see Fig. 3(a). In traditional modeling process, users first form the models in their cerebrum and then exchange the imaged models into basic graphical units (points, lines, curves, etc.). After the time-consuming building process on computers, they can find whether the results are reasonable. If not, the models should be revised by passing the re-modeling process, which is a boring and time-consuming process, and users have to wait patiently when generating and rendering.

The semantic modeling process is depicted in Fig. 3(b). There is a semantic layer before the modeling process, by which users can transform their modeling plans into the semantic description. Only the available plans can be brought forward to the modeling process, which can avoid

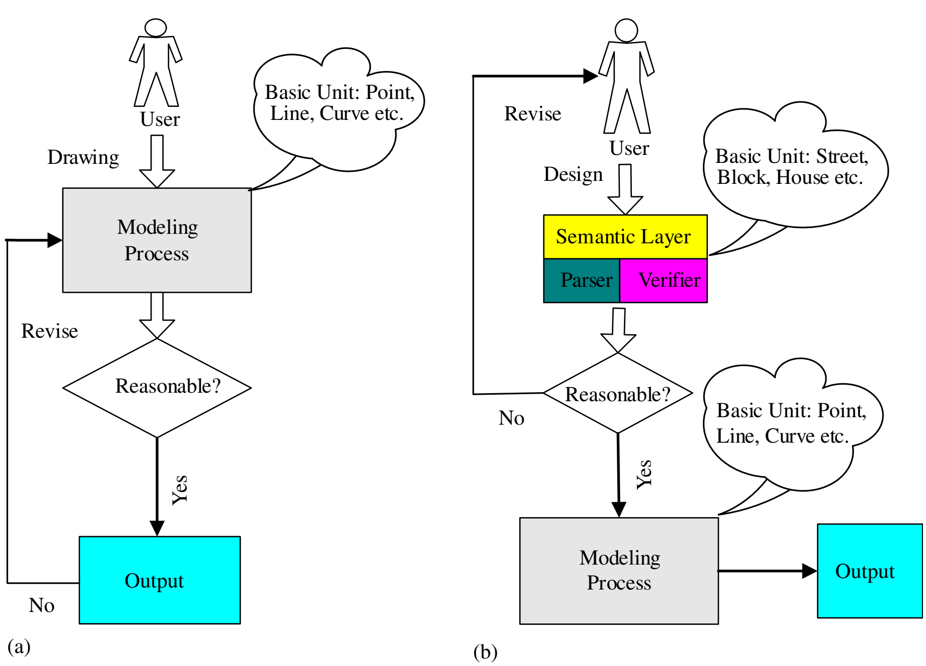


Fig. 3. Comparison between the traditional modeling process and semantic modeling process.

the long and bored re-modeling processes in case of the faulty models.

The semantic layer includes the parser and the verifier. The parser converts the semantic portions into graphical units, and its function is similar to a translator. The verifier checks the availability of the urban models which can be built in the semantic layer, and its function is similar to an auditor.

The verifier of our modeling system is based on the roadmap and the styles of buildings. The verifier works under some common knowledge rules. The first rule is that those houses should be positioned beside the roads and not on them, so the roads could be regarded as the disjunction of the houses. The styles of houses are decided by components and their combination and the styles of components can be dealt with easily, because they are relatively finite. In fact, during the arrangement of these components, it is most difficult to ensure the coherent style of the whole house. In our system, the house is established by the recursive grammar production, and then each house can be individually represented as a sequence of the grammar terms. This way, the style ensuring can be treated as the grammar checking among terms, the verification system only needs to check the sequences generated by the production engine and to match them with the predefined styles.

The detailed system architecture of the semantic modeling process is shown in Fig. 4. In the first step, users should specify the urban map for generation, the semantic architecture components are also specified as the input data. Then the city generator module will match the roadmap with the buildings' block size and patch size automatically and generate the corresponding models in the desired format. The style checking is processed by the city generator with the guide of user specified building style format (DTD), and only the models with right style can be forwarded to the next step for rendering. Before outputting the final model images, the city generator also provides an interactive toolkit (AASMT), see Fig. 17, to adjust the models.

4. Semantic components of vernacular house

The key problem in semantic modeling is the definition of semantic components. Because the vernacular houses in southeast China have similar styles in the components, they can be disassembled into several reusable components, each special house could be regarded as the assembling of these components. In our method, these components are the terms of the recursive productions, which control the houses' generation. Since several components of the vernacular houses, such as the conjunct walls, the roofs and the gate–window walls, have complex constructions and different styles compared with other architectures, the styles of these special components will be introduced in the following.

4.1. Components definition & semantic parser

The components terminologies of vernacular houses are defined in Table 1.

Based on the analysis of the functional components, a hierarchy of the components is built up (Fig. 5). For example, gate, window and wall can be organized as gate–window wall; and gate–window–wall or shop wall, conjunct wall and roof can be organized as house,

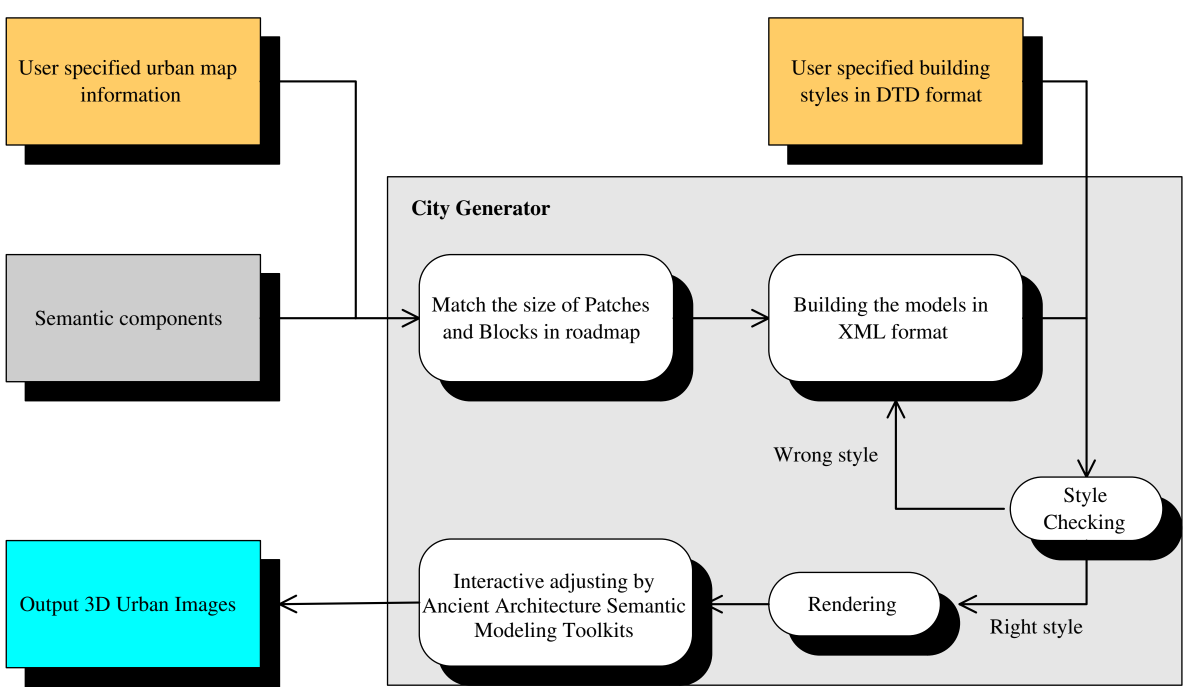


Fig. 4. System implementation of the semantic modeling.

Table 1
Components definition in semantic modeling

Components	Description
Street	Consists of a road and two lines of houses
Zone	Consists of several blocks with the same house type in a certain permutation.
Block	Houses with certain house types separated by roads.
House	Consists of a base, several gates, one roof, more than four walls and several windows.
House type	There are two types, commercial and residential.
House style	The style of houses built in different epoch or region.
Roof	Consists of a lot of shells organized in a certain manner.
Window	Consists of the window frame and texture-based hole.
Gate	Consists of gate frame and the texture-based hole.
Base	A platform on which the house is built.
Texture	Consists of several small texture tiles.
Shell	The smallest element that cannot be divided in a house.
Wall	Including street wall and conjunct wall.
Conjunct wall	It is a wall that shared by two houses.
Street wall	It is a wall adjacent to a street, including gate-window wall and shop wall.
Gate–window wall	It is a street wall of a residential house. (Since it can be divided into wall, gate and window, we can construct a model differing from these three components)
Shop wall	It is a street wall of a commercial house. (Since the shop wall is too complicated to be modeled, we plan to use texture to represent it.)

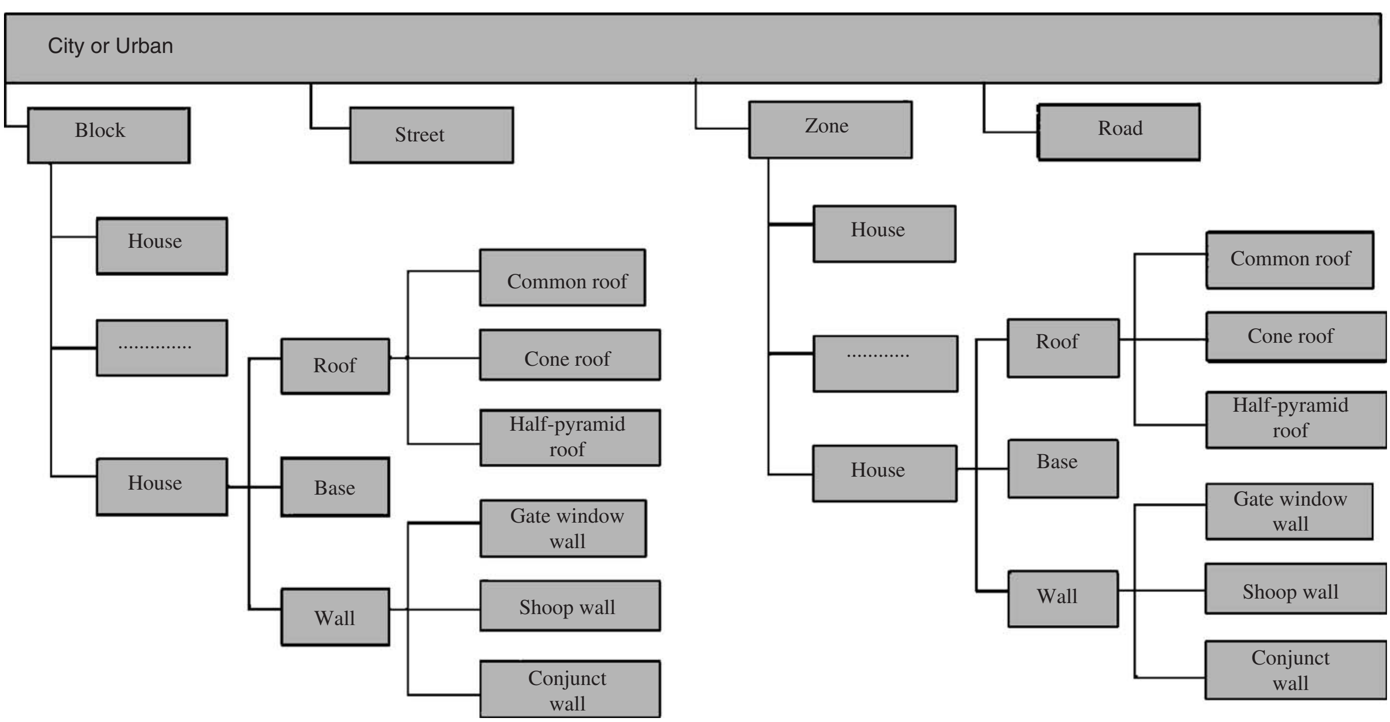


Fig. 5. Hierarchies example of the city components, the city (or urban) can be divided into block, street, zone, road, etc. The house can be divided into three portions: roof, base and wall, each of these portions can be subdivided into some portions.

etc. Fig. 6 gives a sample example of the semantic components in southeast China vernacular house.

4.2. Conjunct walls in vernacular house

The conjunct wall refers to the wall between two houses, which can be used to separate the two houses of different families in the southeast vernacular houses. Conjunct wall consists of two elements: the roof ornament, and the wall body. The roof ornament is a specialty of Chinese traditional architectures, which also exists in all the wall components and looks like a min-roof. It is a simplified common roof, which has no centerline and inclines outside the yard of the vernacular house, this will drain the water outside when raining. So it is used not only as the ornament but also as the simple drain system in vernacular houses. This design is very useful, since there is quite much rain during the whole year of southeast China.

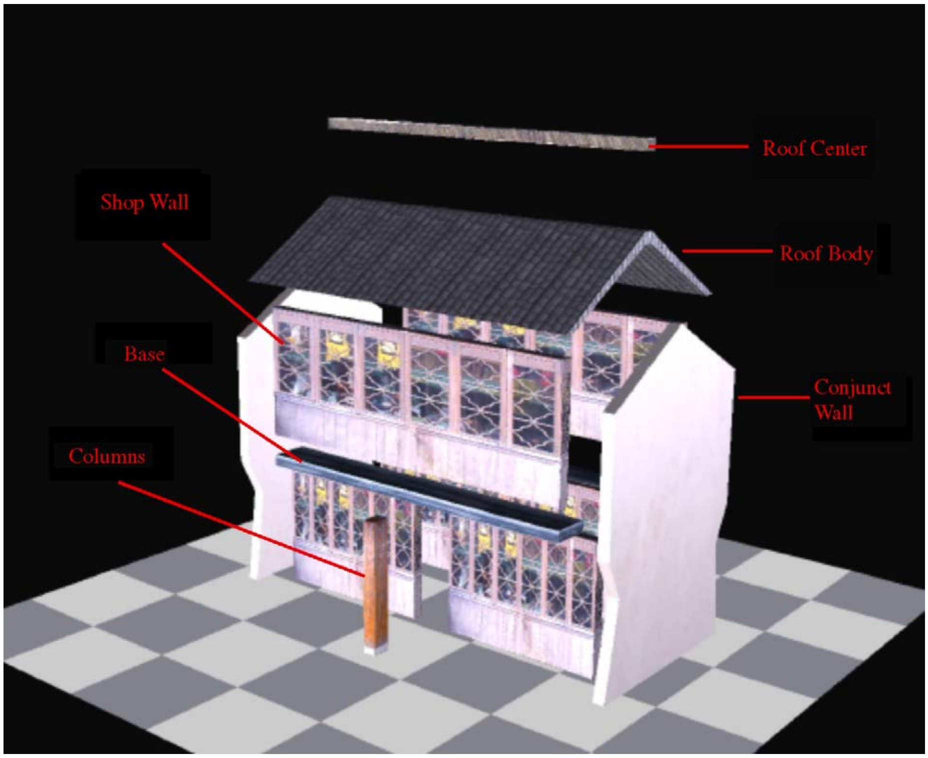


Fig. 6. Semantic description of a simple southeast China house.

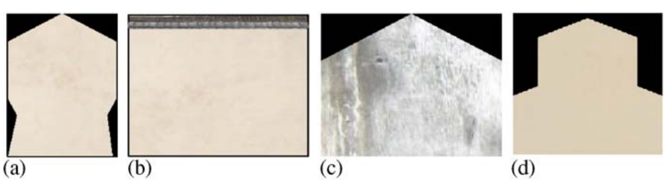


Fig. 7. Four styles of the conjunct walls.

Normally, the geometry of a conjunct wall is irregular as shown in Fig. 7. Fig. 7(b) is the wall with roof ornament.

In our semantic description system, the conjunct wall can be described by several parameters, the length and the width which refers to the minimal container box, the thickness, the geometry (polygon), and the texture, etc. In conjunct wall, the wall ornament is optional based on the position.

4.3. *Roofs in vernacular house*

The roofs of vernacular houses in southeast China may be the most notable symbol of style. The geometry of the roofs is normally inclined downward from the central axes to the outer rims, which can be helpful to the water drain in raining days. There are three types of the roofs in vernacular houses, they are common roof, cone roof and half-pyramid roof. A typical common roof is displayed in Fig. 8. According to it, the parameters of length, width, grade and central axes width of the roof can be described.

Cone roof consists of one center point, several centerlines emitted from the central point and a cone-shaped roof surface. In the implementation view, the central point is a little cylinder, and the centerlines are several cuboids and the cone-shaped roof surface can be constructed by sweeping a rectangle through a circle. So the attributes that a cone roof must have are:

(a) Radius and height of the central point.
(b) Length, width, height and the number of the centerlines.
(c) Length, thickness and joint angle between the cone surface and the perpendicular bisector.
(d) Location and relative height of the center point.

Similar to the cone roof, pyramid roof can also be divided into one central point, several centerlines emitted from the central point and several triangular roof surfaces separated by the centerlines. Especially, the half-pyramid roof appears when a pyramid roof connects with a common roof or a wall. Therefore the pyramid roof is separated by the connecting wall. So the pyramid roof must have the following attributes:

(a) Radius and height of the center point.
(b) Length, width, height and the number of the centerlines.
(c) Joint angle between the centerlines and the perpendicular bisector.
(d) Thickness of the triangular roof surfaces.

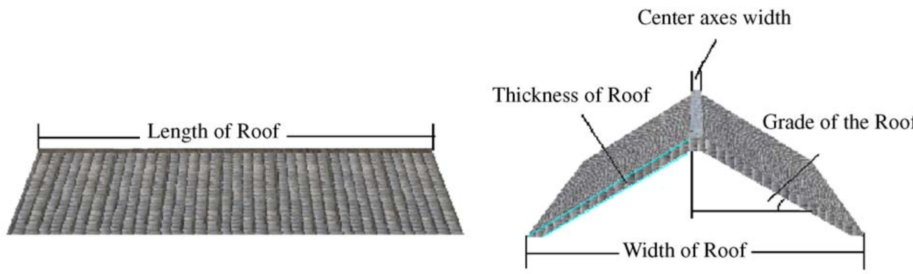


Fig. 8. The common roof of vernacular house in southeast China.

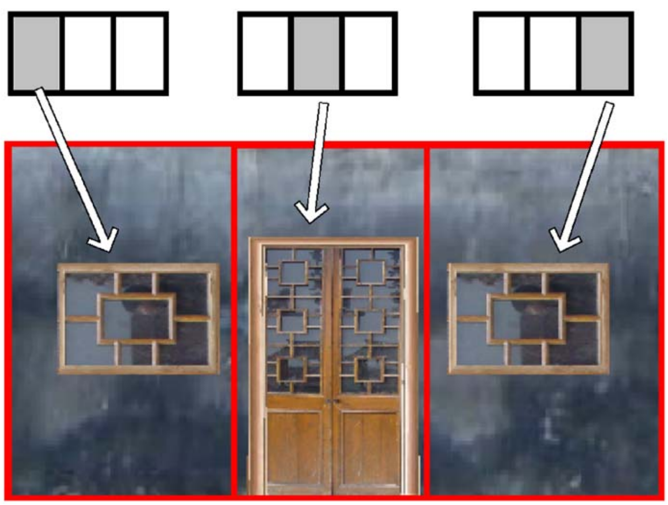


Fig. 9. The grid template for the gate–window wall.

4.4. Gate–Window walls in vernacular house

The gate–window wall refers to the wall that contains the windows and gates. Since the geometry of the gate–window wall is regular, the attributes of the walls, gates and the windows are simple, such as length, width, height, position, texture, etc. However, their combination styles may be varied, how to describe them is quite a challenge. The previous works such as the shape grammar [22–28] and the split grammar [19] have provided a grammar approach; many of them are carried out to generate the architecture automatically using the recursive grammar. Here we adopt a grid template (or style template) technique, which is similar to the split grammar, to maintain the right styles of the gate–window wall in vernacular houses.

The grid template can be used as the verifier in our house generating system [6], it is cell based geometry. Normally, the grid, which corresponds to the gate–window wall, is divided into several cells (Fig. 9), and each cell in the grid contains a window or a wall. Then the combination styles depend on the grid styles, each grid presents a combination style of the gate–window wall. This grid is also called the style template. The details of the template representation will be introduced in Section 5.

5. Semantic control on topology and combination

The above definition of the semantic components only finishes the first step of the semantic modeling; the next important step for semantic modeling is the topology and combination control between those semantic components. In fact, the semantic modeling can be viewed as a language system, in which the semantic components are the basic vocabularies, and the semantic control on topology and combination are language grammars. To assemble the vocabularies randomly cannot present the right meaning, only the assembling with right vocabularies under the grammars can represent the right meaning. So the semantic control on topology and combination is quite important to represent and model the vernacular houses' styles.

The topology control concerns the plane position relationship among houses and walls. The combination control refers to the compounding relationship among all the semantic components defined in previous section.

Obviously the description of grammar is more difficult than that of vocabularies. In our approach, an XML based description and DTD based verification technique have been adopted.

5.1. Semantic control on topology

In our semantic modeling, semantic components are the basic functional units that aim to the urban builders who do not need to grasp those profound techniques of computer graphics. The only thing that they should do is to design the topology of the components. Here, we use an XML [29] based knowledge representation method to depict the topology of the components.

Fig. 10 shows a block unit in our semantic modeling system. By adopting the XML based description method, we can obtain the results as follows:

<Block BlockName = = "Block1">
<House housename = "NorthHouse">
<Layout layouttype = "BorderLayout">
NORTH
</Layout>
… … … … …
… … … … …

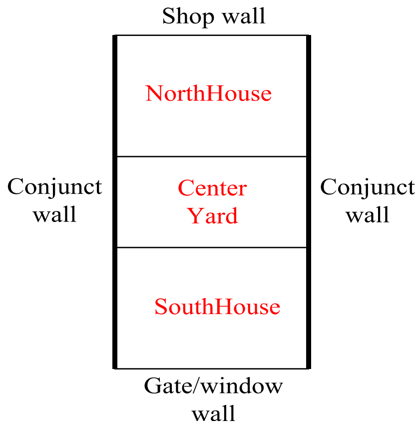


Fig. 10. A typical topology existed in southeast China vernacular house, the unit is normally a rounding architecture; it includes the central yard, a shop wall, gate–window wall, two conjunct walls and two houses.

</House>
<Yard yardname = "CenterYard">
<Layout layouttype = "BorderLayout">
CENTER
</Layout>
… … … … … …
</Yard>
<House housename = "SouthHouse">
<Layout layouttype = "BorderLayout">
SOUTH
</Layout>
… … … … … …
</House>
</Block>

After parsing the above XML code, the modeling system maintains a hierarchical structure of the typical vernacular house, which can be checked by the improved control rules system and then sent to the rendering module for generation. Only when the maintained vernacular house structure parsed from XML based semantic description can meet the requirements of these improved control rules, then the rendering step can be processed. Fig. 11 shows the generated results of the above XML code.

There are six typical topologies of the vernacular house in southeast China, listed in Table 3 (Appendix A). Similarly, they can also be described by the XML styles as well as the example given in Fig. 12. In Table 3, the description of ratio means the corresponding relationship among length of the house, width and length of the topological block. The length and the width in this table refer to the length and width of the whole topological block. The examples in Table 3 are all generated from the XML based description automatically by our ancient architecture semantic modeling toolkits (AASMT) shown in Fig. 17.

5.2. Semantic control on combination

In the modeling process, there should be some rules to ensure the validity of the generated urban buildings. In our previous system [6], we presented a recursive grammar engine to drive the urban modeling system. The terms in the production rules are constituted of those components defined in Section 4.1. In this recursive system, the texture has become one kind of terms that are controlled by the production system. However, this recursive grammar system also has its common weakness, that it is hard to control the generated results (actually we have to design a verification system to ensure and check the generated results in our previous system). So we improve the rule of recursive grammar system, adding the ratio factor to each of these rules. Then when the generation system controls the modeling process, it has to obey both the components combination relationship, which is controlled by the recursive rules, and the components ratio.

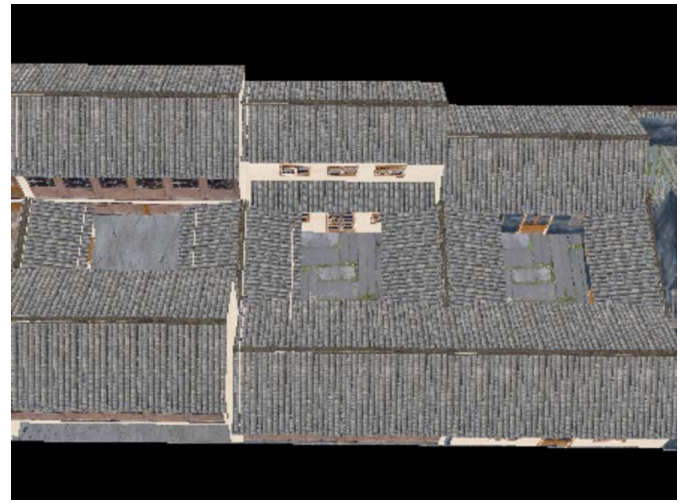


Fig. 11. Typical vernacular house in southeast China.

The improved productions are defined as follows:

*{**CITY** :: (ROAD), (BLOCK);*
[R_{road} (Width, Length, Height, Number)]: [R_{block}(Width, Length, Height, Number)]
}
*{ **BLOCK** :: (PATCH), (BLOCK) | ε;*
[R_{patch} (Width, Length, Height, Number)]: [R_{block}(Width, Length, Height, Number)]
}
*{ **PATCH** :(CONJUNCT_WALLl), (HOUSE), (PATCH) |ε;*
[$R_{conjunct\_wall}$(Width, Length, Height, Number)]: R_{house}(Width, Length, Height, Number)]: [R_{patch}(Width, Length, Height, Number)]
}
*{**HOUSE** :: (BASE), (WALL), (ROOF) ;*
[R_{base}(Width, Length, Height, Number)]: R_{wall}(Width, Length, Height, Number)]: [R_{roof}(Width, Length, Height, Number)]
}

The rule defines both the combination and the ratio relationships. Here the ratio includes four parameters, they are width, length, height, and number, respectively.

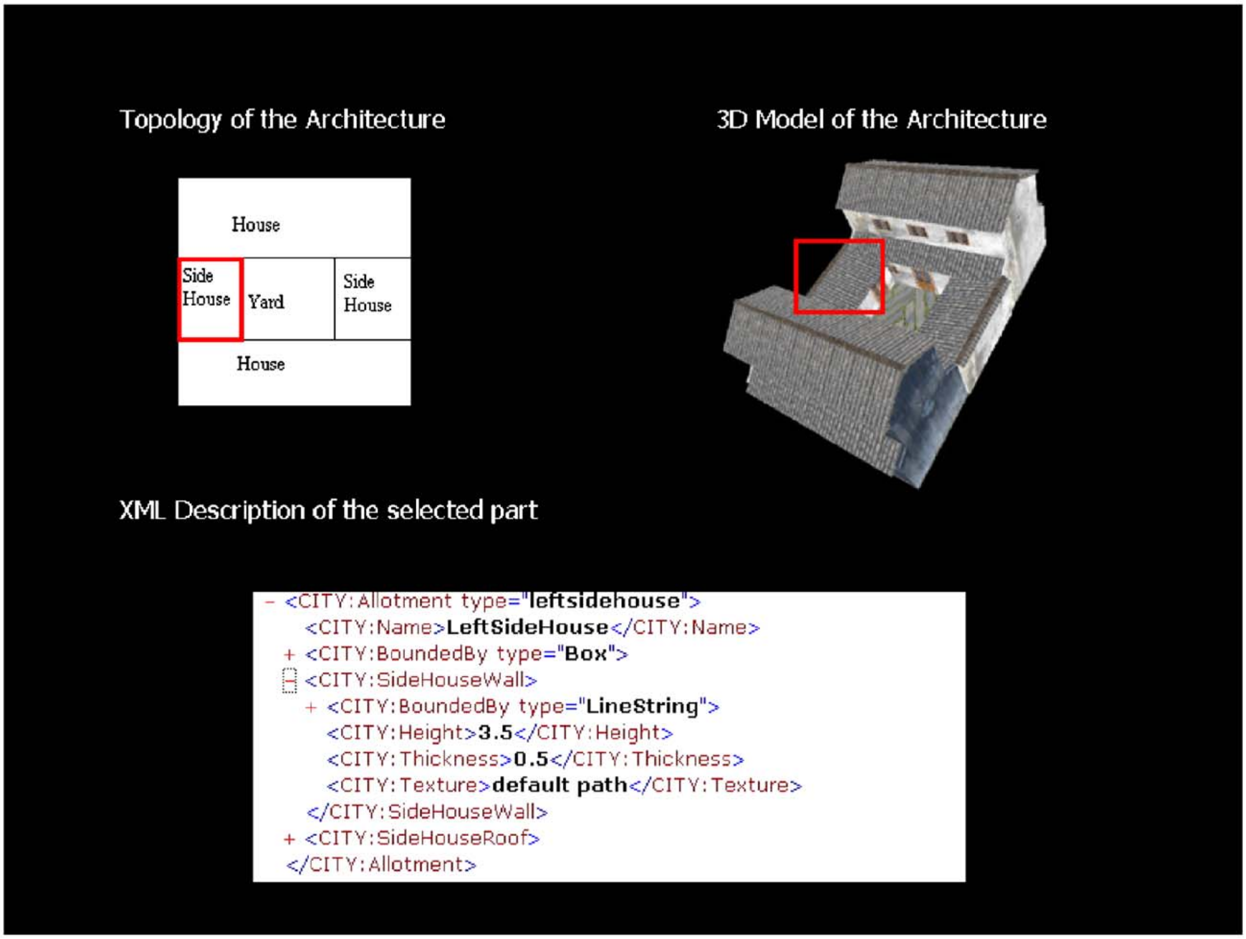


Fig. 12. Topology, description, 3D-Model for the semantic implementations.

The first three parameters give the outline ratio relationship among components, and the *number* parameter describes the numeric ratio among components.

As the rule based recursive systems have achieved many successful implementations such as those in [5,6,19]. It is more suitable for the auto-modeling than verifying whether the architecture style is rational, for the simple constraint rules cannot describe all the styles of complex ancient architectures. On the other hand, once the ancient architecture is illustrated by the rule terms directly, how can we know the style of the architecture obeys the constraint rules in those recursive systems? So a more expressive technique for the semantic modeling is required.

As the topology and the semantic components are all described by the XML, the DTD [30] can be used as the verification protocols. Figs. 13 and 14 give a tree viewed DTD verification grammar for the combination of components. A more detailed DTD file is listed in Appendix B.

In our modeling system, the XML and DTD also implement control rules. The urban models can be multi-instances, but the style of one kind of model is same. So the DTD supported by the XML technique can provide a flexible control policy to regulate the whole style and integrity of models.

Fig. 13. A wrong example generated by the city engine without the control of DTD.

5.3. Roadmap description and control

Another special component in our semantic modeling is the road in southeast China. Normally the spare space

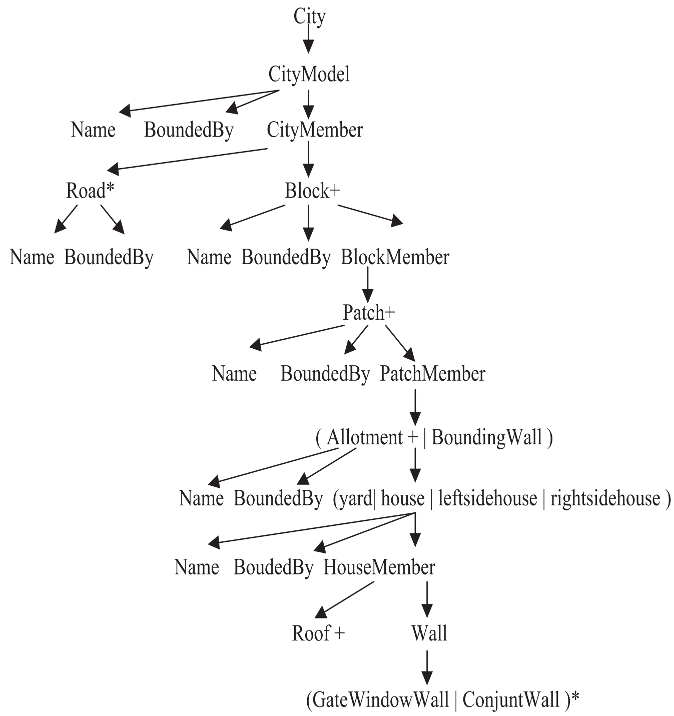


Fig. 14. A tree viewed DTD description for the semantic modeling, here the "BoundedBy" is the minimal contained box of the components, and the "+" means the components can be duplicated one or more, "*" means the components can be duplicated zero or multiple times, "?" means the components can be duplicated zero or one time, and "|" means the components are optional. The arrow means the component at the beginning of the arrow contains the components at its end.

Table 2
The default properties for the road components

Properties	Notes
Road identity	We can differ the road components in the description by using this identity
Road name	A name that the user assigns for the road
Road texture	The rendering texture of the road components
Road geometry information	The position information of the road components

outside the patches and blocks in the semantic components can all be viewed as the roads. However, this condition cannot be always true for all the cases, sometimes, there may be some other additional components that lie on the spare space, and the styles of different roads in patches may be varied. So the roads should be distinguished and marked individually.

The description for the roadmap is also implemented by the XML, and we use the "road" tag to describe the road components, and each road contains four default properties which are listed in Table 2.

The road geometry information is described by four basic geometric elements, i.e. the point, linestring, box and polygon, which are similar to the GML (Geography Markup Language) [31]. So we can process the topology by invoking the GML interfaces.

The DTD definitions for the four elements are given as follows:

1. Point element, the basic unit
<!ELEMENT Point (X, Y, Z)>
<!ELEMENT X (#PCDATA)>
<!ELEMENT Y (#PCDATA)>
<!ELEMENT Z (#PCDATA)>
2. Linestring element
<!ELEMENT LineString (Point,Point)>
3. Box element
<!ELEMENT Box (Point,Point)>
4. Polygon element
<!ELEMENT Polygon (Point+)>

As the roadmap information and the semantic components description language are coherent, we can use a unified description parser. So the parser should be compatible on both the GML and our XML based ancient architecture description language.

6. Experimental results

Here we present some images generated as the output of the semantic urban modeling system (Figs. 15–18). All the images are performed in a real-time system, and the render is implemented in Java3D based on the semantic geometry description files. Fig. 15 demonstrates two different styles of the textures used in vernacular houses. Fig. 16 shows some simple vernacular houses in the right styles. Fig. 17 presents the screenshots of the interactive adjusting toolkit AASMT, which can display both the tree view of the modeling in hierarchical structures (left view) and the visualization of the models (right-up view), then users can adjust the parameters in the tree structures directly, and the visualization of the models will be adjusted as the parameters changing. The interactive adjusting result is shown in Fig. 17(b), and Fig. 17(a) is the model before adjusting. Fig. 18 shows the streets which are consisted of the vernacular houses in southeast China.

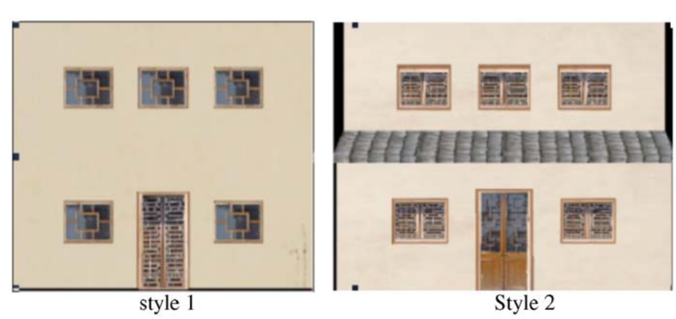


Fig. 15. Two styles of the texture components generated by the production system. Style 1 is a texture generated for planar wall, and Style 2 is the texture generated for two layers' wall with a roof.

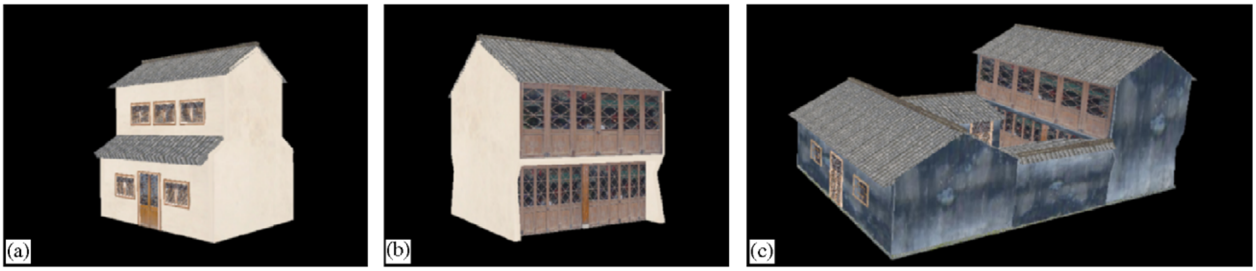


Fig. 16. Two types of houses and block generated by our system. The left (a) is a residential house, and the central (b) is a business house, and the right (c) is a block.

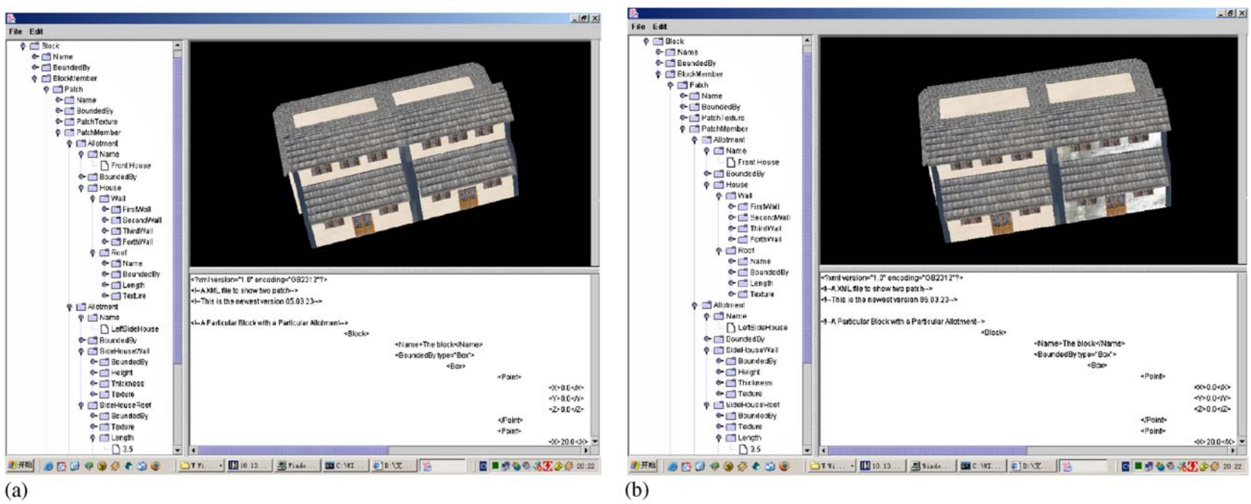


Fig. 17. Ancient architecture semantic modeling toolkits (AASMT), left view is the tree structure of the ancient architecture model, and the right-down view is the XML description for current construction, and the right-up view is the visualization of the architecture models.

Fig. 18. Overview of the vernacular houses' block.

7. Conclusion and future work

This paper discusses the semantic modeling for vernacular houses in southeast Chinese by separating the whole models into several functional components, such as gate, window, wall, and roof, etc. Semantic modeling can be more understandable and controllable, because the modeling process is represented by unified description. The unified description provides more intelligent grammar for the modeling, so users can focus on the urban design and ignore the trivial graphical drawing when generating their models. The corresponding constraint rules are also provided, which can avoid the disadvantages existing in other recursive grammar systems. Our improved production based grammar system can ensure the similar style of the vernacular houses in southeast China.

We envision several possible directions for future work. First, it may focus on the smart distribution of vernacular houses based on different functions of the cities, extending more DTD for different styles of the architecture such as modern architectures, European architectures, etc. Second, the style for multiple houses integrating environment is also an interesting work. Third, the fast and effective methods for corresponding virtual environment generation should be more concerned, which could be used in virtual navigation [32].

Acknowledgements

This project is supported by the Key Project of Natural Science Foundation of China (No. 60533090, No. 60533080, No. 60402010), and it is also partially supported by China 973 Program (No. 2002CB312106).

Appendix A

Table 3.

Table 3
Six typical house topologies in southeast China vernacular house

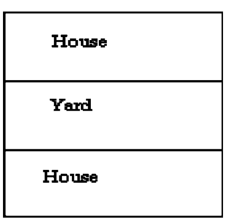
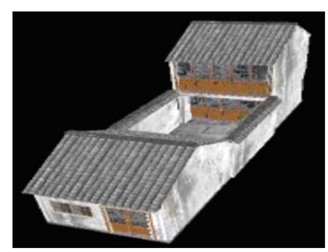
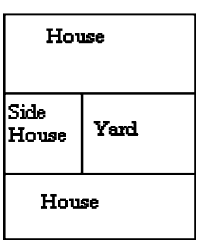
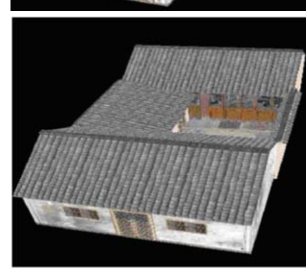
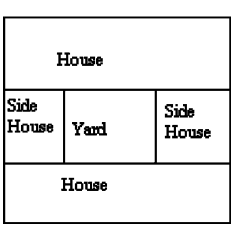
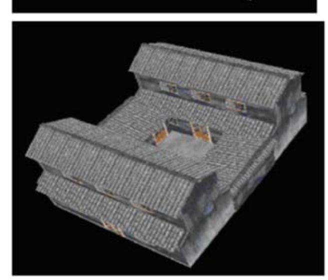
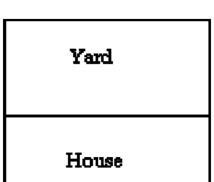
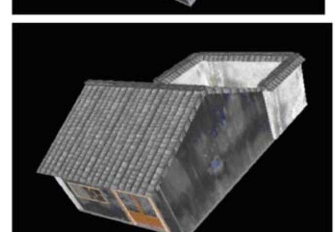
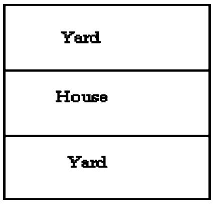
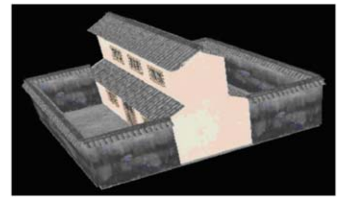
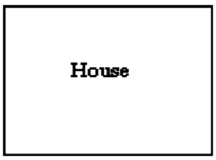
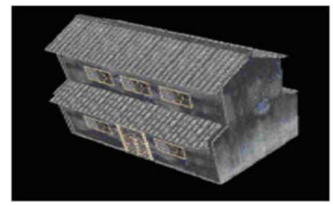
Topology	Description of ratio	Examples
	0.8∗HouseLength < length < 1.5∗HouseLength	
	1.5∗HouseLength < length < 2∗HouseLength	
	2∗HouseLength < length < 2.5∗HouseLength	
	1.3∗HouseLength < width < 2∗HouseLength	

Table 3 (*continued*)

Topology	Description of ratio	Examples
	2∗HouseLength < width < 2.8∗HouseLength Or 0.8∗HouseLength < length < 1.5∗HouseLength	
	0.8∗HouseLength < width < 1.3∗HouseLength	

Appendix B

DTD for the house description:

```
<?xml version = "1.0" encoding = "GB2312"?>
<!ELEMENT CITY:Allotment (CITY:Name, CITY:BoundedBy, CITY:House?, CITY:SideHouseWall?,
CITY:SideHouseRoof?, CITY:Texture?)>
<!ATTLIST CITY:Allotment
    type (house | leftsidehouse | rightsidehouse | yard) #REQUIRED
>
<!ELEMENT CITY:Block (CITY:Name, CITY:BoundedBy, CITY:BlockMember)>
<!ELEMENT CITY:BlockMember (CITY:Patch*)>
<!ELEMENT CITY:BoundedBy (CITY:Box | CITY:LineString | CITY:Polygon)>
<!ATTLIST CITY:BoundedBy
    type (Box | LineString | Polygon) #REQUIRED
>
<!ELEMENT CITY:BoundingWall (#PCDATA)>
<!ELEMENT CITY:Box (CITY:Point,CITY:Point)>
<!ELEMENT CITY:City (CITY:CityModel)>
<!ATTLIST CITY:City
    xmlns:CITY CDATA #REQUIRED
>
<!ELEMENT CITY:CityMember (CITY:Road*, CITY:Block+)>
<!ELEMENT CITY:CityModel (CITY:Name, CITY:BoundedBy, CITY:CityMember)>
<!ELEMENT CITY:FirstFloorWall (CITY:BoundedBy, CITY:Height, CITY:Thickness, CITY:Texture)>
<!ELEMENT CITY:FirstWall (CITY:FirstFloorWall, CITY:SecondFloorWall?, CITY:SecondRoof?)>
<!ATTLIST CITY:FirstWall
    type (GateWindowWall | ShopWall | ConjunctWall ) #REQUIRED
>
<!ELEMENT CITY:ForthWall (CITY:Name, CITY:BoundedBy, CITY:Thickness)>
<!ATTLIST CITY:ForthWall
    type (GateWindowWall | ShopWall | ConjunctWall ) #REQUIRED
>
<!ELEMENT CITY:Height (#PCDATA)>
<!ELEMENT CITY:House (CITY:Wall, CITY:Roof)>
<!ATTLIST CITY:House
    type (residential | commercial) #REQUIRED
    Floor (1 | 2) #REQUIRED
```

```
>
<!ELEMENT CITY:Length (#PCDATA)>
<!ELEMENT CITY:LineString (CITY:Point,CITY:Point)>
<!ELEMENT CITY:Name (#PCDATA)>
<!ELEMENT CITY:Patch (CITY:Name?, CITY:BoundedBy?, CITY:PatchTexture?, CITY:PatchMember?)>
<!ATTLIST CITY:Patch
    type (1|2|3|4|5|6|7) #REQUIRED
>
<!ELEMENT CITY:PatchMember (CITY:Allotment+, CITY:BoundingWall?)>
<!ELEMENT CITY:PatchTexture (#PCDATA)>
<!ELEMENT CITY:Point (CITY:X, CITY:Y, CITY:Z)>
<!ELEMENT CITY:Polygon (CITY:Point+)>
<!ELEMENT CITY:Road (CITY:Name, CITY:BoundedBy)>
<!ELEMENT CITY:Roof (CITY:Name, CITY:BoundedBy, CITY:Length, CITY:Texture)>
<!ATTLIST CITY:Roof
    type CDATA #REQUIRED
>
<!ELEMENT CITY:SecondFloorWall (CITY:BoundedBy, CITY:Height, CITY:Thickness, CITY:Texture)>
<!ELEMENT CITY:SecondRoof (CITY:BoundedBy, CITY:Length, CITY:Thickness, CITY:SlantAngle,
CITY:Texture)>
<!ELEMENT CITY:SecondWall (CITY:Name, CITY:BoundedBy, CITY:Thickness)>
<!ATTLIST CITY:SecondWall
    type (GateWindowWall | ShopWall | ConjunctWall ) #REQUIRED
>
<!ELEMENT CITY:SideHouseRoof (CITY:BoundedBy, CITY:Texture, CITY:Length, CITY:Thickness,
CITY:SlantAngle)>
<!ELEMENT CITY:SideHouseWall (CITY:BoundedBy, CITY:Height, CITY:Thickness, CITY:Texture)>
<!ELEMENT CITY:SlantAngle (#PCDATA)>
<!ELEMENT CITY:Texture (#PCDATA)>
<!ELEMENT CITY:Thickness (#PCDATA)>
<!ELEMENT CITY:ThirdWall (CITY:FirstFloorWall, CITY:SecondFloorWall?, CITY:SecondRoof?)>
<!ATTLIST CITY:ThirdWall
    type (GateWindowWall | ShopWall | ConjunctWall ) #REQUIRED
>
<!ELEMENT CITY:Wall (CITY:FirstWall, CITY:SecondWall, CITY:ThirdWall, CITY:ForthWall)>
<!ELEMENT CITY:X (#PCDATA)>
<!ELEMENT CITY:Y (#PCDATA)>
<!ELEMENT CITY:Z (#PCDATA)>
```

References

[1] Mech, R., Prusinkiewicz, P. Visual models of plants interacting with their environment. In: Proceedings of ACM SIGGRAPH 96, 1996. p. 397-410.

[2] Prusinkiewicz P, Lindenmayer A. The algorithmic beauty of plants. Springer; 1991.

[3] Prusinkiewicz P, James M, Mech R. Synthetic topiary. In: Proceedings of ACM SIGGRAPH 94, 1994. p. 351–8.

[4] Shlakhter I, Rozenoer M, Dorsey J, Teller S. Reconstructing 3D tree models from instrumented photographs. IEEE Computer Graphics and Applications 2001. 53–61.

[5] Parish YIH, Muller P. Procedural modeling of cities, SIGGRAPH'01, 2001. p. 301–8.

[6] Liu Y, Xu C, Pan Y. Fast modeling of vernacular houses of southeast China. In: Proceedings of Fourth International Conference on Virtual Reality and its Application in Industry; 2003.

[7] Yamada G, Kawase K, Hideki K, Hoshino Y. Modern Urban Planning Dictionary, Syokokusya; 1997.

[8] Henderson JV. Urban development, theory, fact and illusion. Oxford University Press; 1988.

[9] Itou S. Urban design and simulation. Kashima Press; 1999.

[10] Marc Levoy, Kari Pulli, Brian Curless, et al. The digital michelangelo project: 3D scanning of large statues. In: Proceedings of ACM SIGGRAPH 2000. p. 131–44.

[11] De Luca L, Véron P, Florenzano M. Semantic-based modelling and representation of patrimony buildings. In: SVE Workshop towards Semantic virtual environments. Switzerland: Villars; 2005. p. 27–36.

[12] Leonard McMillan, Gary Bishop. Plenoptic modeling. An image-based rendering system. In: Proceedings of SIGGRAPH 95. Los Angeles, CA, USA, August 1995. p. 39–46.

[13] Lifeng W, Zhigeng P. Imaged-based rending techniques in virtual reality. Chinese Journal of Image and Graphics 1998;13(12): 1005–9.

[14] Fang X, Yan W, Pan Z. EasyPanorama: a new panorama authoring system. In: Proceedings of fourth international conference on virtual reality and its application in industry; 2003.

[15] Byong Mok Oh, Max Chen, Julie Dorsey, Frédo Durand. Image-based modeling and photo editing. In: Proceedings of SIGGRAPH' 01. Los Angeles, CA, USA, August 2001. p. 433–42.
[16] Yeung SH, Richard S. Construction of panoramic image mosaics with global and local alignment. International Journal of Computer Vision 2000;36(2):101–30.
[17] ChiuShui Chan, Ziyu Tong, Anrong Dang. Virtual reality modeling of traditional Chinese architecture. In: Proceedings of ninth international conference on virtual systems and multimedia. Montreal, October 2003. p. 13–22.
[18] MultiGen-Paradigm Corp. Vega programmer's guide. 2000. 〈http://www.multigen-paradigm.com〉.
[19] Wonka P, Wimmer M. Instant Architecture. ACM Transactions on Graphics 2003;22(4):669–77.
[20] Birch P, Jennings V, Day AM, Arnold DB. Procedural modelling of vernacular housing for virtual heritage environments. EGUK; 2001.
[21] Browne SP, Birch PJ, Jennings VJ, Day AM, Arnold DB. Rapid procedural modeling of architectural structures. In proceedings of VAST, Athens, November 2001. p. 187–96.
[22] Stiny G. Pictorial and formal aspects of shape and shape grammars. Basel: Birkhauser Verlag; 1975.
[23] Stiny G. Introduction to shape and shape grammars. Environment and Planning B 1980;7:349–51.
[24] Knight TW. Designing with Grammars. In: Schmitt GN, editor. Computer-aided architectural design. 1992. p. 33–48.
[25] Knight TW. Color grammars: designing with lines and colors. Environment and Planning B: Planning and Design 1989;16:417–49.
[26] Stiny G. Ice-ray: a note on Chinese lattice designs. Environment and Planning B 1977;4:89–98.
[27] Flemming U. More than the sum of its parts: the grammar of Queen Anne houses. Environment and Planning B 1987;4:323–50.
[28] Stiny G, Mitchell WJ. The Palladian grammar. Environment and Planning B 1978;4:5–18.
[29] W3Cconsortium. ''Extensible Markup Language (XML) 1.0'' Recommendation, 1998. 〈http://www.w3.org/TR/1998/REC-xml-19980210/〉.
[30] 〈http://www.w3schools.com/dtd/default.asp〉.
[31] ISO/TC 211/WG 4/PT 19136 Geographic information—Geography Markup Language (GML) 〈https://portal.opengeospatial.org/files/?artifact_id=7174〉
[32] Steed A. Efficient navigation around complex virtual environments. In: Proceedings of the ACM VRST 1997. p. 173–80.