Computers & Graphics 36 (2012) 178–184

Contents lists available at SciVerse ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Novel Applications of VR

Improving procedural modeling with semantics in digital architectural heritage

Liu Yong [a], Zhang Mingmin [b,*], Jiang Yunliang [c], Zhao Haiying [d]

[a] Institute of Cyber Systems and Control, Zhejiang University, 310027 Hangzhou, China
[b] College of Computer Science, Zhejiang University, 310027 Hangzhou, China
[c] School of Information and Engineering, Huzhou Teachers College, 31300 Huzhou, China
[d] College of Computer Science and Technology, Xinjiang Normal University, Urumqi 830054, China

ARTICLE INFO

Article history:
Received 30 April 2011
Received in revised form
5 December 2011
Accepted 11 January 2012
Available online 28 January 2012

Keywords:
Procedural modeling
Digital architectural heritage
Semantic ontology
Rule reduction

ABSTRACT

We first introduce three challenges in the procedural modeling of digital architectural heritages and then present a general framework, which integrates several machine intelligence and semantic techniques, e.g., the ontology design approach, pattern mining, auto-annotating and rule reduction, to improve the procedural methods in architectural modeling. Several evaluations and experiments are also presented. The experimental results illustrate the improvements following our approach.

1. Introduction

The aim of digital heritage is to construct or record cultural objects accurately with digital techniques and then to manipulate, represent and express these cultures with proper multimedia methods. Among these cultural objects, ancient architecture is an important heritage. Because it requires a huge cost to construct architectural heritages individually and manually due to their various styles and complex textures, many researchers are focusing on the problem of automatically modeling architectures. Because the architectural heritages that need to be digitally rebuilt may have been destroyed or may not have existed, only a few ancient literatures have recorded the facades of architecture heritages; 3D scanning methods [1,2] or image-based modeling [3–5] methods are not applicable in such cases of modeling architectural heritages. Under this condition, the procedural modeling method may be an optimal approach because the styles of architectural heritages are naturally clustered by their region and historical age. The procedural modeling approach can generate a large number of architectures with similar styles and facades very quickly, and then, artists may easily make further revisions to construct accurate digital architectural heritages.

The procedural method has been approved as a useful approach in urban modeling. It employs procedural rules, which are controlled by a specific grammar, the basic units in architecture (or urban architecture), to randomly generate the models by combining the basic units with the procedural rules. With the proper grammar and units, the procedural method could produce large numbers of architectures with similar styles very quickly.

However, there are several weaknesses; e.g., it is hard to obtain the procedural rules, and there is a low success rate for architectural models as well as a lack of annotation for digital architectural heritages, which constrain its further implementation in digital heritages. In this paper, we address the weaknesses of the current procedural method in modeling digital architectural heritages and attempt to integrate several semantic techniques into the current procedural modeling methods. In our approach, we first investigate the deep connotations of those drawbacks and then propose a general procedural modeling framework with semantic techniques. Finally, we introduce a real case of ancient Chinese architecture modeling with this semantic procedural framework. To evaluate the improvement of the semantic procedural modeling processing, a measurable variable is defined, and the corresponding experimental results are compelling and encouraging.

2. Related works and challenges of procedural modeling in digital architectural heritages

The procedural method is a hot topic in architectural automodeling due to its simple parameters and fast generating speed.

* Corresponding author.
E-mail addresses: cckaffe@yahoo.com.cn (L. Yong), zmm@cad.zju.edu.cn (Z. Mingmin).

0097-8493/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cag.2012.01.003

It first occurred in the shape grammar [6] that was implemented in architectural analysis and new style construction. Then, a L-system based grammar was introduced in procedural urban modeling [7], which used recursive grammar to divide the patches and to place the buildings. In addition, a new grammar named split grammar was designed by Wonka [8] that generated the house model via transforming, scaling, extruding and splitting the basic geometrical units. Additionally, these two above techniques have been integrated into one interactive urban modeling system [9] and have achieved impressive success in the virtual heritage project of the ancient city of Pompeii. Another popular procedural method [10,11] employed the basic architectural components on the semantic level, such as windows, walls and gates, and generated models by combining these semantic components. The other technique that should be mentioned is modeling with the control of natural language [12,13]. However, these systems are limited in two aspects: it is hard to guarantee the coherence of logic in natural languages, and ambiguities inherent in languages make it difficult to describe some target scenes. Recently, a new approach that implements the existing specific styles on the new instances of architecture via interaction has been introduced by Daniel [14], which rudimentarily learns the architectural style from the existing models and reproduces similar architectures; thus, interesting and possible solution are provided to extract the rules in procedural modeling.

Although procedural methods have achieved impressive success in architectural auto-modeling, these methods still suffer from several problems (or challenges).

1. The first challenge is *how to enhance the capability of procedural methods in complex architectures*. The procedural modeling task becomes difficult when the style of architecture is intricate, which means a greater complexity of the basic units and a greater obscurity of the procedural rules and grammars. Building those complex architectures with procedural methods is quite a heavy burden for designers. Extracting the procedural rules for these complex architecture styles is quite difficult, and they are not reusable for different architecture styles, which means that designers may have to redesign the procedural rule systems for different architecture styles.

2. The second challenge is *how to auto-annotate the generated models properly* when using procedural methods. In digital heritage, rather than only generating the models of heritages, we should add some cultural explanations for these models or parts of these models to present the culture to the public. The traditional procedural modeling method cannot support these notes, and annotating the generated models manually will require a large amount of labor. Therefore, how to auto-annotate the models in the procedural modeling method may be the second challenge. There are also many works that attempt to implement semantic techniques into the digital heritage applications [15–17]. One of the most useful techniques in digital heritage is semantic annotation, which employs ontology to extract structured knowledge from the natural language description [15]. It may reduce the manual work for the general case; however, for a digital architectural heritage case with a special historical (or regional) background, the method may fail due to the absence of concept libraries within the corresponding special scopes, e.g., we cannot annotate the heritages of the Chinese Sui Dynasty with the concept libraries of Ancient European heritages. In particular, for the current procedural modeling approach, extracting the structured semantic components from the generated models and annotating them are impossible.

3. The third challenge is *how to boost the efficiency of the procedural methods*. The efficiency of the procedural methods does not refer to the temporal performance; it represents the success (generating desired models) ratio of the procedural methods. When using the procedural methods to generate models, only a few of them are desirable, and we will obtain many more bio-products that are of the incorrect combination, incorrect topology or incorrect style models. This is caused by the redundancy of the procedural rules, which are obtained manually and subjectively.

The above three challenges are concerned with the shareability, scalability and reliability of the procedural methods. In the next section, we will introduce intelligent semantic techniques for procedural modeling and attempt to achieve the requirements of those challenges.

3. Procedural modeling solutions with semantic

Here, we first analyze the challenges in procedural modeling methods from the viewpoint of semantic and machine intelligence and then present our solutions with detailed intelligent semantic techniques.

3.1. Intelligent semantic viewpoint on the challenges

Machine intelligence has received great success in computer graphics [18] and visualization [19]. Additionally, procedural architectural modeling can also be regarded as a similar problem [20]. In digital architectural heritages, the aim of procedural modeling is to generate architectures with a specific style, which can be described by an ontology concept from a semantic viewpoint, consisting of a knowledge library (i.e., the procedural rules), an entity set in the specific style of architecture (i.e., the basic geometrical units in the procedural methods) and the transferring function among the knowledge library, entity set and model instances [20,21].

Therefore, the first challenge can be described by how to design the ontology corresponding to the complex architectures. The advanced features of ontology will provide more powerful tools for designers. With the description of ontology, designers can also use some graph mining techniques [22] to find the hidden topology and combination patterns for complex architectures. The rules can be shared among different styles with common ontological parents. The second challenge will be represented by auto-generating the annotations from the ontology definition. The third challenge is concerned with both the procedural algorithms and the selection of procedural rules when generating models. Either irrelevant procedural rules or improper algorithms will lead to the low efficiency of the modeling system.

From the viewpoint of ontology, designing entities is easier in practice. A simple intuition design may refer to the natural component categories of architecture, e.g., the roof, window and wall. The three challenges are all concerned with the “goodness” of the design for the knowledge library. To achieve the “good” knowledge library, we should reduce the redundancy in the procedural rules, which are introduced subjectively.

Because all of these procedural rules are manually extracted by architects, most of them are empirical and imprecise. The knowledge library may be highly redundant, which is a major drawback, especially when we generate a single style of architecture, such as ancient Chinese styles or ancient Indian styles. The redundancy in the knowledge library can be summarized as follows:

- The rules in the knowledge library will describe several styles of architecture, so it is redundant to a specific architecture style. It is called *other-style redundancy* [21].
- Some general rules should be shared by multiple architecture styles, i.e., the gate and wall combination rules of ancient Chinese architecture may be the same as that of ancient Indian architecture. This redundancy is called *share redundancy* [21].
- Some incorrect rules may be introduced due to mistakes made by the architects. If these rules are included in the generation

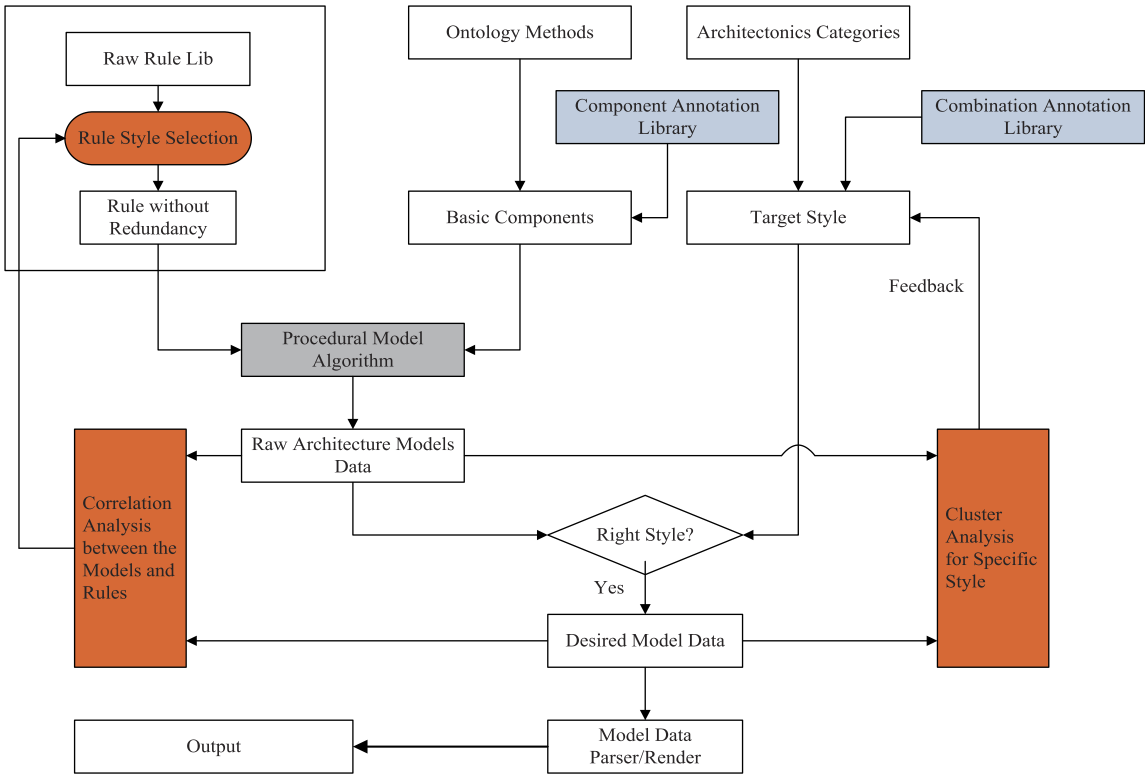


Fig. 1. Our solution for a semantic procedural architecture modeling system.

of the architecture, they will affect the generated results. This redundancy is called *incorrect-rule redundancy* [21].

- There may be more than one rule that refers the same combination or topology relationship of a specific style. Obviously, redundancy occurs due to knowledge repetition. This redundancy is called *repeating redundancy* [21].

The drawbacks of redundancy are obvious.

- The incorrect rules in a knowledge library will produce unreasonable architectures, e.g., the gate may lie upon the window or the roof may lie below the pillar. This is the concern of the third challenge.
- Because the knowledge library contains rules for multiple styles that share the same rules, "un-unified" styles and unexpected structures will be generated, which means that the architectures may be hybrid styles. For example, some architectures may contain an Indian roof and a Chinese mainframe.
- The redundancy and incorrect rules will decrease the performance of the modeling system. Users need to generate a large number of instances and only obtain a few satisfactory results.

3.2. Details of our approach

Here, we discuss a new approach for procedural modeling systems to overcome the limitations of the challenges. In our approach, we use ontology to improve the design of complex architectures, to find the hidden pattern for the complex architecture style by implementing data mining methods, to share the rules for different styles, to auto-generate annotations for architectures, and to reduce the redundancy of the procedural rules by feature selection methods. The framework of the solution is shown in Fig. 1.

3.2.1. Ontology design

The challenges of procedural methods require simplifying the design task for complex architectures. The ontology technique has provided many useful tools and protocols in the concept representation, and it particularly adapts to distinguish similar conceptions. In our approach, we first design the basic hierarchical architecture components to construct the entities set V in ontology; the basic units are all described in a semantic XML format [11], and all of the characteristics and relationships among the components are regarded as the knowledge library R in ontology. A sample is given as follows:

V = {'*roof*', '*wall*', '*shopwall*', *conjunctwall*',
'*floor*', '*column*', '*house*', '*base*'}
R = {
house :: *base*|*wall*|*roof*;
wall :: *shopwall*|*conjunctwall*;
wall :: *shopwall*|*column*|*shopwall*|*conjunctwall*;
}

where V is the semantic component set, and R is the knowledge library.

Then, the generation of models can be regarded as a function $F_R(V)$ that employs rules from R and basic components from V to generate the architectural models. If the generated model w belongs to the desired architectural heritage style, we denote it as $w \in W$, where W is the instance set in ontology. Obviously, W refers to all of the models generated by $F_R(V)$ (procedural method) that belong to the correct heritage style, and it is an infinite set.

Therefore, our solution in complex architectural procedural methods design does not obtain the procedural rules and basic units.[1] On the contrary, we design them along the hierarchical conceptions of the architectonics category and approach our target

[1] In our system, the basic units are the actual architecture components in architectonics [20].

from the parental concept and similar concepts from the ontology operations, such as knowledge reusing and overlapping [23].

3.2.2. Pattern mining

In our procedural modeling system [11], architecture models are stored in a XML format that combines both the combination and topology information. As the XML organizes the data with structural tree labels, we use some cluster-based algorithms [24] to discover the frequent combining pattern from both the raw model data and the verified data, which is shown in Fig. 1. There are two pattern mining methods, manual structure mining and auto-structure mining. Manual structure mining begins from the pre-defined combining pattern and filters the patterns with high frequencies, and auto-structure mining does not use the pre-defined pattern templates but enumerates several possible templates and then searches for frequent patterns with the enumerated templates.

3.2.3. Automatic annotations

There are two kinds of annotations for architecture heritages in our approach. One includes the annotations for single components, which present the cultural background or its features; the other one contains the annotations for combination patterns, which present the description of specific spatial combinations of the basic units. In our approach, the annotations for components are based on the ontology, which present possible aliases for the components and some simple annotations. When generating models in procedural modeling, the aliases and simple annotations, along with the used components, are also added to the models. However, this method needs a designer to add the annotations to the corresponding components manually; therefore, we employ a components annotation library, which contains the explanation of the heritages and can be generated from encyclopedias or online Wiki sites. Then, the components described by ontology can match their corresponding annotations from that library, which is similar to Tukka's method [17].

The combination annotations generated in our approach are based on the combination annotation library, which is presented by the designer and consists of the following pattern rules:

$[[c_1]Tc_2]L[c_3c_6c_7]R[[c_4]Bc_5]D \rightarrow \text{Description}$

All of the geometric objects in our shape grammar are within a boundary box shown in Fig. 2, and we also define six spatial relationships for the combinations, which are "Top(T) of the object", "Down(D) of the object", "Left(L) of the object", "Right(R) of the object", "Front(F) of the object" and "Back(B) of the object". With

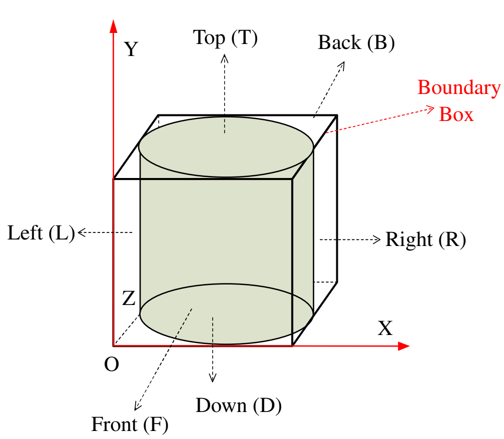


Fig. 2. The representations of boundary box and coordinate for a spatial object used in our approach, and this figure also shows the six spatial control terms in our shape grammar.

these spatial control terms, experts could easily present their own domain knowledge on the spatial combinations of the scenes. For example, the roof of a Southeast Chinese house may be constituted with two components: a roof center (c_1) and a roof body (c_2). The roof center lies on the top of the roof body; therefore, we can express the roof as the following:

$\text{Roof} = [c_1]Tc_2$

In the above example, '$c_3c_6c_7$' means that the three objects are placed in alignment with the X-axis.

3.2.4. Rule reducing

As the redundancy problem impair procedural modeling, it should be removed when modeling. The rule reduction consists of two steps, one is designing the measure, which could evaluate the performance of the rules that are used. It also represents the correlation between the model data and the rules that are used. The other step is the rule selection algorithm, which should find the optimal rule set quickly based on the predesigned measure.

The correlation measure, which is the most important factor in rule reduction, could introduce many machine learning methods, such as correlation analysis, information theory, granule and margin theory. Here, in our approach, we use a granular-based correlation measure.

In addition to representing the goodness of the current rule set, a good correlation measure will also benefit the selection algorithms greatly because most computation costs in selection algorithms are spent on correlation computation.

In our solution, for the modeling rules reduction, a roughness-based correlative measurement function is used [25,21].

$\underline{P}$ represents the rules in P that can be certain with respect to the desired architecture style and is calculated as follows:

$$\underline{P} = \left\{ \bigcup r \,|\, \forall w \text{ where } w \in W,\ r \notin K,\ w = F_{K\cup\{r\}}(V) \text{ and } r \in P \right\} \tag{1}$$

Here, P is a subset of the whole knowledge library U, and V and W are the entity set and instance set of ontology, respectively.

Similarly, $\overline{P}$ represents the rules in P that can be possible with respect to the desired architecture style, which is calculated as follows:

$$\overline{P} = \left\{ \bigcup r \,|\, \exists w \text{ where } w \in W,\ r \notin K,\ w = F_{K\cup\{r\}}(V) \text{ and } r \in P \right\} \tag{2}$$

Here, $F_{K\cup\{r\}}(V)$ refers to the process of utilizing the rule set $K \cup \{r\}$ to combine the basic architecture components V into an architecture heritage as shown in Fig. 3.

The optimal procedural rule set can be calculated as follows:

$$P{*} = \arg\max \left(\frac{|\overline{U-P}| - |\underline{U-P}|}{|\overline{U-P}|} - \frac{|\overline{P}| - |\underline{P}|}{|\overline{P}|} \right)$$

In our implementation, the K is normally initialized as several basic architecture modeling rules, which is necessary for the desired style architecture generation. Then, for testing each rule r in P for whether the $w = F_{K\cup\{r\}}(V)$ is the correct style ($w \in W$), see Fig. 3. Similarly, $\underline{U-P}$ and $\overline{U-P}$ can also be defined by testing the rule r in the rule set $U-P$ (U is the whole knowledge library) for whether the $w = F_{K\cup\{r\}}(V)$ is a correct style model.[2]

Because the search space is incredible large (it is 2^N when there are N rules), we should add some heuristic strategies or present some approximate substitutes in our selection algorithms.[3]

[2] $\forall w \in W$ can be determined by enumerating a large number N of instances; we believe that the negative condition will occur when N is large enough.

[3] More details please refer our previous publications in [23,21].

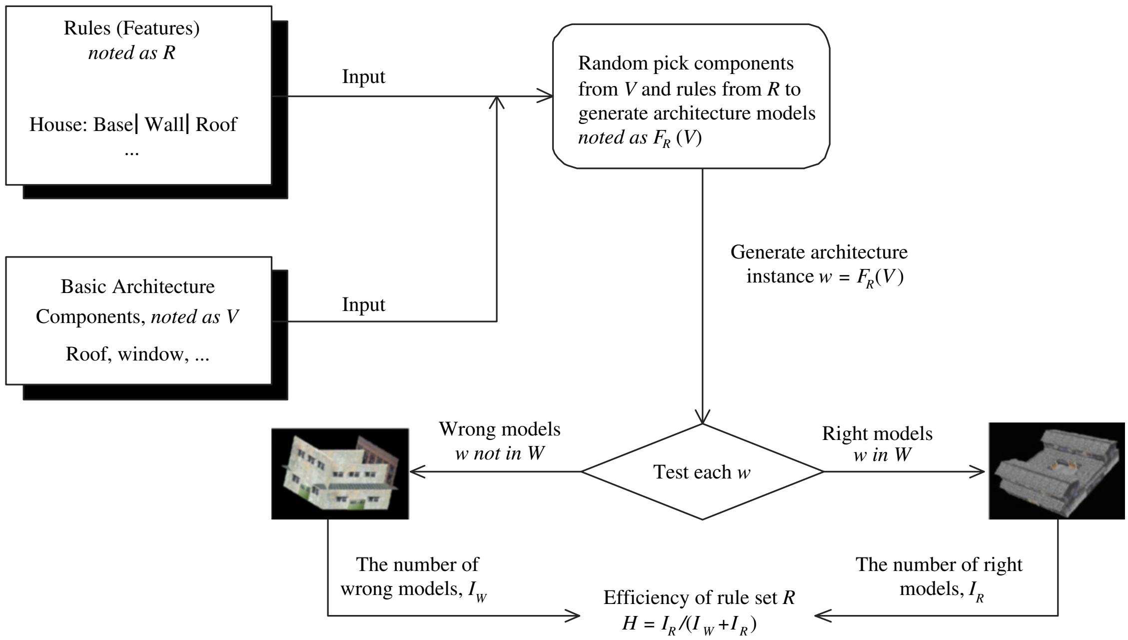


Fig. 3. Implementation of our architecture modeling system [21].

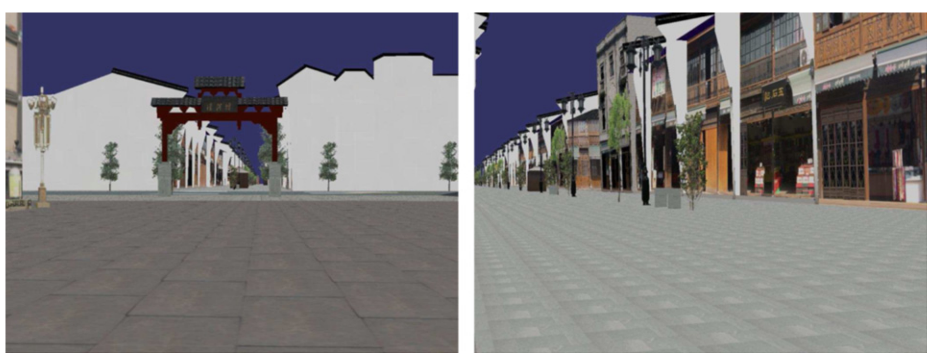


Fig. 4. Experimental results of the virtual heritage of Hefang Street, Hangzhou, China.

4. Evaluations and results

In our virtual heritage project, building a large number of houses manually is a heavy burden, so we considered using semantic procedural methods in our task. We first used the procedural model system, which could generate a semi-style and semi-structure in ancient Chinese architecture and then adjust the details by our interactive toolkits [11]. We aimed to build Hefang Street, which is shown in Fig. 4, in Hangzhou, which used to be the commercial downtown of Hangzhou (it was the Capital of the South Song dynasty from 1217 to 1279 and is now the capital of Zhejiang province). We also used the semantic procedural modeling method to construct the virtual Jing-Hang Grand Canal both with 3D models and historical annotations [26]; the results are shown in Fig. 5. The Jing-Hang Grand Canal is the longest ancient man-made canal in the world.

We also implemented intelligent and semantic techniques to reduce the rule set and to find the optimal procedural rule subset for ancient Southeast Chinese architecture models. Our modeling system [11,23], which aims to automatically generate a similar 3D style and similar structural ancient Chinese architecture models, works as shown in Fig. 3. The modeling has been aided by an ontology technology to unify the notion definitions [20]. The 3D models are generated by randomly assembling the basic house components under a series of first order logic grammar rules. As the rules are concluded subjectively, there may be much redundancy in the entire rule set, which will lead to a low efficiency ratio H (for the definition, please refer to Fig. 3), and thus, the number of satisfying models is rather low. In our modeling system, the rules applied during generation are similar to the features in classification; they need to be filtered with the proper method, e.g., the method in Section 3.2.4. Using a good rule set will generate architecture models as accurately as possible. Here, the modeling problem is especially sensitive to the rules that are used; less control rules will decrease the styles of the models, even though they can have a very high efficiency ratio. Therefore, rule selection should maintain the integrity of the subset with respect to the modeling cases.

4.1. User study evaluation

In this section, we describe a user study comparing our semantic procedural modeling approach with the CityEngine [27], which is a commercial procedural modeling toolkit based on the works of Mueller et al. [9,7,28].

To evaluate the effectiveness of our approach, we invited 10 graduate students from different backgrounds, all of whom were familiar with computers but none of whom had previous experience with professional 3D modeling. We spent 30 min training each participant with our semantic procedural modeling grammars and

Fig. 5. Experimental results of Jing-Hang Grand Canal generated by the semantic procedural approach.

another 30 min for the CGA grammars of CityEngine [27,9],[4] in a randomized order. Then, the participant was shown a sequence of photographs of ancient Southeast Chinese architecture and ancient urban blocks in random order. The participants were required to design the procedural grammar system to produce architectures and blocks with the same style as those in the photographs, with a maximum of 30 min spent on each task. Half of the participants began with CityEngine, and half began with our semantic approach. We then used each grammar system designed by the participants to generate 20 instances of ancient Chinese architecture heritages; thus, each participant produced 40 instances, 20 by the CGA grammar in the CityEngine and 20 by our semantic procedural approach. After that, we invited five heritage experts to divide these instances into “correct models” and “incorrect models” depending on whether their styles belonged to the ancient Southeast Chinese. We used $I^C_{W_i}$ and $I^S_{W_i}$ to denote the incorrect model number generated with CityEngine and with our semantic procedural approach, respectively, for the ith participant. The corresponding correct model number generated with CityEngine and our approach is denoted as $I^C_{R_i}$ and $I^S_{R_i}$, respectively. For the judgment of each heritage expert, we calculated the efficiency ratios (efficiency of the rule set) of CityEngine and of the semantic approach as follows:

$$H^C_i = I^C_{R_i}/(I^C_{W_i}+I^C_{R_i}), \quad H^S_i = I^S_{R_i}/(I^S_{W_i}+I^S_{R_i}) \tag{3}$$

The average efficiency ratios of the five experts for each participant are given in Table 1.[5]

Table 1
User study results for the average efficiency ratio of five experts.

Participant	CityEngine (%)	Semantic (%)
1	29	40
2	40	55
3	50	55
4	25	65
5	75	90
6	47	75
7	63	80
8	15	32
9	16	29
10	28	44
Average	38.8	56.0

We believe that the comparison of the efficiency ratios in Table 1 illustrates that the grammar system designed with our approach was consistently more efficient than the grammar system conducted by CityEngine. Using a t-test analysis, the effect of the grammar system on the model's efficiency ratio was statistically significant with higher than 99% confidence.

In the second phase of our user study, the five experts would discuss and rank the instances in each group[6] and then chose the best instance from every group of 20 instances. Then, we obtained the two-model pairs from each designer; one model was generated by CityEngine, and the other model was generated by the semantic procedural approach.

In the third phase, another 10 graduate students who had not participated in the previous phase were asked to compare the two-model pairs designed by the same student based on their quality and coherency of style and make a decision on which one of the two-model pairs was better. The models are rendered by third-party commercial software and could not be distinguished as to which procedural modeling approach was used to create them. We call these 10 graduate students reviewers and call the previous 10 graduate students users. Before the reviewers began their judgments, we spent 10 min to familiarize them with the style of ancient Southeast Chinese architectural heritages.

The comparison of the reviewers' results is provided in Table 2. In the third phase, a direct comparison of the ranks of the models made with different participants can be misleading because the designs of highly skilled users tend to outrank those of less skilled users, regardless of the approach that was used. It is more appropriate to analyze how each participant fared when using our approach compared with using CityEngine. To quantify the extent of this preference, we calculated that for all of the pairs, meaning the best model[7] generated using our approach and the best model[8] generated using CityEngine by the same participant, more reviewers preferred the models generated using our approach is better than those generated by CityEngine.

We also calculated the probability of receiving a positive judgment in the third phase, which resulted in a high probability of 91% with a confidence interval of [0.8376, 0.9521] and a confidence level of 95%.

4.2. Discussion

In this section, we will discuss some of the differences between these two procedural modeling approaches that could account for the difference in quality of the models produced in our user study.

The first difference is that the rule system may be established for different design patterns. The rules in our semantic procedural modeling method are naturally categorized by their concept domains,

[4] The tutorial is available in the “Help” section of CityEngine.

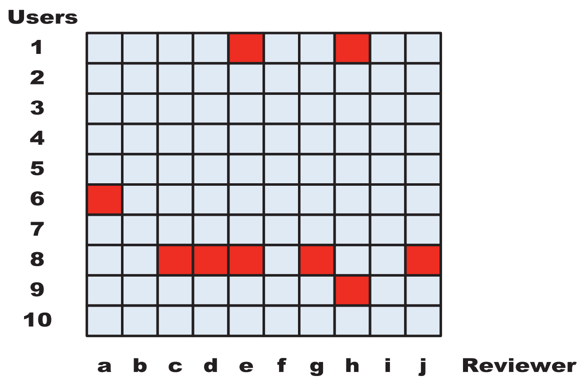
[5] Here, we did not present the standard deviation of each efficiency ratio because there are fewer differences among the five experts in the decision between incorrect or correct.

[6] Here, each group refers to the 20 instances generated by either CityEngine or the semantic procedural approach of each designer; the experts will discuss and come to a consensus about which one is the best model in each group.

[7] It was chosen by five experts in the second phase.

[8] It was chosen by the five experts in the second phase.

Table 2
User study results: each row is the comparison result of a different reviewer (a–j) of the two-model pairs generated by the study participants. The gray boxes indicate positive judgments, and the red boxes indicate negative judgments.



which may be regarded as a "top-down" design pattern. The rules in CityEngine are parallel or non-hierarchical, which may increase the cost to manage them and sometimes even increase the redundancies in the rule system and lead to low performances in the efficiency ratio.

Another difference is with the basic elements that each procedural modeling approach used. CityEngine uses basic geometrical objects, such as area and volume, while our approach uses the semantic components incorporating ontology techniques. We found that users were more immediately comfortable with the grammar for basic geometrical objects in the case of architecture modeling, such as structured modern architectures. However, we also found that using the semantic components may generate superior results when constructing unstructured complex architectures, e.g., ancient Chinese architectural heritages, because of the following: in the semantic procedural approach, the components are hierarchical in natural semantic cognition, which may help designers to be more focused on the design of spatial combinations and coherent styles; the semantic procedural approach enables users to attach additional ontology descriptions with both FOL (first order logic) and DL (Description logic) rules, which can provide more accurate descriptions of the targets; and the semantic components in our approach could support automatic annotations by instinct.

5. Conclusion

In this paper, we addressed three challenges of the procedural modeling method in digital architectural heritages and presented a general framework to construct large-scale, similar 3D models of digital architectural heritages and culture annotations. We also evaluated our approach via experiments and sample cases.

We are now focusing on increasing the procedural modeling system's scalability and ease of use. Another issue we are currently addressing is how to discover more useful hidden patterns using machine learning algorithms and data mining techniques.

Acknowledgments

This research is supported by The National Natural Science Foundation Project of China (60803053, 61173123, 60970076, 61173124, 60863010), "Information Processing and Automation Technology" Zhejiang Province Key Subject Open Fund (201100803), Natural Science Foundation Project of Zhejiang Province (Y1100070) and Special Program for the National Basic Research Program of China (Pre-973 Program, Grant no. 2010CB334709).

References

[1] Früh C, Zakhor A. 3D model generation for cities using aerial photographs and ground level laser scans. In: CVPR, vol. 2, 2001. p. 31–8.
[2] Addison AC, Gaiani M. Virtualized architectural heritage: new tools and techniques. IEEE MultiMedia 2000;7:26–31.
[3] Debevec PE, Taylor CJ, Malik J. Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. In: Proceedings of the 23rd annual conference on computer graphics and interactive techniques, SIGGRAPH '96. New York, NY, USA: ACM; 1996. p. 11–20.
[4] Poli D. Reality-based 3D city models from aerial and satellite data. Geoinformatics 2006;9(3):8–11.
[5] Antone ME, Teller SJ. Automatic recovery of relative camera rotations for urban scenes. In: IEEE conference on computer vision and pattern recognition; 2000. p. 2282–9.
[6] Knight TW. Designing with grammars. In: Schmitt GN, editor. Computer-aided architectural design. Weisbaden: Verlag Viewag; 1992. p. 33–48.
[7] Parish YIH, Müller P. Procedural modeling of cities. In: Proceedings of the 28th annual conference on computer graphics. Los Angeles, CA, USA: ACM; 2001. p. 301–8.
[8] Wonka P, Wimmer M, Sillion FX, Ribarsky W. Instant architecture. ACM Trans Graph 2003;22(3):669–77.
[9] Müller P, Wonka P, Haegler S, Ulmer A, Gool LJV. Procedural modeling of buildings. ACM Trans Graph 2006;25(3):614–23.
[10] Birch PJ, Browne SP, Jennings VJ, Day AM, Arnold DB. Rapid procedural-modelling of architectural structures. In: Proceedings of the 2001 conference on virtual reality, archeology, and cultural heritage, Glyfada, Greece, November 28–30, 2001. pp. 187–96.
[11] Liu Y, Xu C, Pan Z, Pan Y. Semantic modeling for ancient architecture of digital heritage. Comput Graph 2006;30(5):800–14.
[12] Johansson R, Berglund A, Danielsson M, Nugues P. Automatic text-to-scene conversion in the traffic accident domain. In: Proceedings of the 19th international joint conference on artificial intelligence. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.; 2005. p. 1073–8.
[13] Coyne R, Sproat R. Wordseye: an automatic text-to-scene conversion system. In: SIGGRAPH; 2001. p. 487–96.
[14] Aliaga DG, Rosen PA, Bekins DR. Style grammars for interactive visualization of architecture. IEEE Trans Vis Comput Graph 2007;13(4):786–97.
[15] Tzouveli PK, Simou N, Stamou GB, Kollias SD. Semantic classification of byzantine icons. IEEE Intell Syst 2009;24(2):35–43.
[16] Isaac A, Wang S, Zinn C, Matthezing H, van der Meij L, Schlobach S. Evaluating thesaurus alignments for semantic interoperability in the library domain. IEEE Intell Syst 2009;24(2):76–86.
[17] Ruotsalo T, Aroyo L, Schreiber G. Knowledge-based linguistic annotation of digital cultural heritage collections. IEEE Intell Syst 2009;24(2):64–75.
[18] Hertzmann A. Machine learning for computer graphics: a manifesto and tutorial. In: Pacific conference on computer graphics and applications; 2003. p. 22–6.
[19] Liu Ma K. Machine learning to boost the next generation of visualization technology. IEEE Comput Graph Appl 2007;27(5):6–9.
[20] Liu Y, Xu C, Zhang Q, Pan Y. Ontology based semantic modeling for chinese ancient architectures. In: Proceedings of the 21st national conference on artificial intelligence (AAAI'06), the 18th conference on innovative applications of artificial intelligence (IAAI'06), Boston, MA, 2006.
[21] Liu Y, Jiang Y, Huang L. Modeling complex architectures based on granular computing on ontology. IEEE Trans Fuzzy Syst 2010;18(3):585–98.
[22] Chakrabarti D, Faloutsos C. Graph mining: laws, generators, and algorithms. ACM Comput Surv 2006;38(1):2.
[23] Liu Y, Xu C, Zhang Q, Pan Y. The smart architect: scalable ontology-based modeling of ancient chinese architectures. IEEE Intell Syst 2008;23(1):49–56.
[24] Li G, Feng J, Wang J, Zhou L. Incremental sequence-based frequent query pattern mining from xml queries. Data Min Knowl Discov 2009;18:472–516.
[25] Modrzejewski M. Feature selection using rough sets theory. In: Machine learning: ECML-93, european conference on machine learning, Vienna, Austria; April 5–7, 1993. p. 213–26.
[26] Chen W, Zhang M, Pan Z, Liu G, Shen H, Chen S, Liu Y. Animations, games, and virtual reality for the Jing-Hang grand canal. IEEE Comput Graph Appl 2010;30(3):84–8.
[27] The Cityengine, Procedural Inc. 〈http://www.procedural.com/〉.
[28] Müller P, Zeng G, Wonka P, Gool LV. Image-based procedural modeling of facades. ACM Trans Graph 2007;26(3):85.