IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 27, 2017, accepted September 14, 2017, date of publication September 20, 2017, date of current version October 25, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2754858

Rapid Modeling of Chinese Huizhou Traditional Vernacular Houses

SHANG-LIN LI, LIN LI, MING-WEI CAO, LI CAO, WEI JIA, (Member, IEEE), AND XIAO-PING LIU
School of Computer and Information, Hefei University of Technology, Hefei 230009, China

Corresponding author: Xiao-Ping Liu (lxp@hfut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61370167, Grant 61673157, and Grant 61602146, in part by the Natural Science Foundation of Anhui Province for Youths under Grant JZ2015AKZR0664, and in part by the National Key Research and Development Plan under Grant 2016YFC0800100.

ABSTRACT In this paper, we propose an automatic modeling method based on constructive grammar for building Chinese Huizhou traditional vernacular houses, which are famous and important examples of Chinese traditional architecture. The proposed method includes two steps: layout generation and feature generation. In the layout generation step, layout guidelines based on 2-D geometric rules are used to evaluate the layout result. In the feature generation step, a probabilistic network is used to generate the features for a single house while feature conflict penalty functions are used to generate the reasonable features for multiple houses. Both the layout generation and feature generation steps use the Metropolis–Hastings algorithm to search for the optimal solution. To increase the computational efficiency of the proposed system, the two steps are performed on a graphics processing unit device using a Monte Carlo sampler. Moreover, a dynamic probability strategy is used to perform the layout perturbation. The proposed method is easy to implement and extend. The experimental results show that the proposed method is practical and efficient. Non-expert users can rapidly generate reasonable Huizhou vernacular houses via simple and intuitive interactions with the system.

INDEX TERMS Rapid modeling, procedural modeling, building modeling, stochastic optimization, Huizhou traditional vernacular House.

I. INTRODUCTION

In recent years, interactive 3D graphics and virtual reality technologies have rapidly developed and the demand for urban planning and simulations as well as digital heritage conservation has increased. As a result, the rapid modeling of buildings (RMB) has received significant attention by academics and industry leaders. RMB is the automatic or interactive modeling of buildings using computer graphics technologies. The buildings to be modeled include virtual buildings as well as actual historical and urban city buildings.

In RMB, historical buildings are a special case that can be used for different applications, such as digital heritage conservation, historic monument reconstruction, and virtual tourism. However, modeling historical buildings is a complex task. In certain applications, the user may only be concerned with the outer geometric surface of the building. In other applications, the constructive rules, building component details and interior structural information underlying the building geometry should also be modeled. To describe the details of historical buildings, the building elements (or components) are parameterized and assembled to create or form an entire building, a process known as parametric modeling and building information modeling (BIM). In 2009, historical building information modeling (HBIM) [1] was proposed, which focuses on the emergence of historical architectural pattern books that define architectural rules and details. This technique provides the opportunity to develop a new method of modeling historical buildings [2].

A variety of RMB techniques are available, and they can be classified into two groups: rule-based modeling methods and data-based modeling methods. Rule-based modeling methods use predefined rules and grammar to control the modeling procedure. The most well-known methods are procedural modeling methods [3]–[8], which have become the mainstream RMB methods because of their successful applications [6]. Rule-based modeling methods provide a cost-effective solution to modeling all types of buildings, including historical buildings; however, the majority of these methods require the user to have a high level of technical knowledge to modify the rules and grammar of the model. Data-based modeling methods model or reconstruct buildings using various types of data. Based on the data type used,

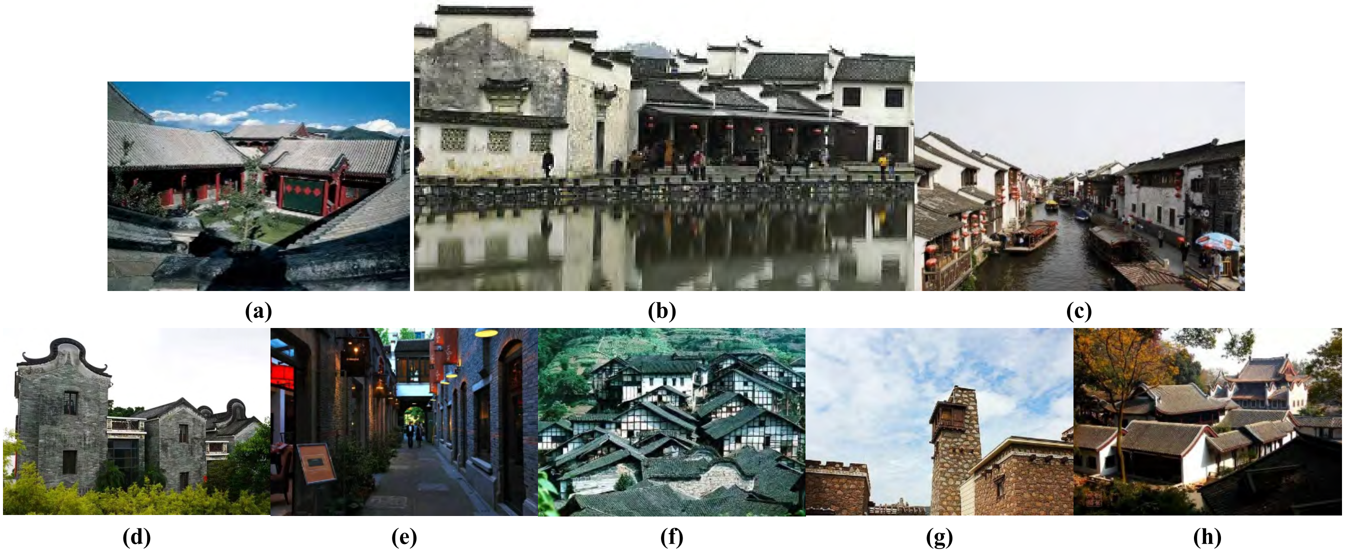


FIGURE 1. Eight major genres of traditional architecture in China: (a) Beijing courtyard house, (b) Huizhou architecture, (c) vernacular houses of south east China, (d) Lingnan architect, (e) Shanghai architecture, (f) Western Sichuan vernacular house, (g) Qionglong architecture, (h) Academy building.

data-based modeling methods can be divided into image-based modeling methods [9]–[13] and LiDAR-based modeling methods [14]–[17]. Data-based modeling methods are user friendly and provide intuitive, human-machine interactions or an automatic reconstruction pipeline. However, the modeling results are highly dependent on the data entered. Because image data and LiDAR data contain only the outer surface geometric information of buildings, they do not easily represent the architectural rules and details of historical buildings, which are important features modeled by HBIM.

China might have more traditional buildings than any other country in the world because of its large geographic size, multiplicity of nationalities, and long history. There are approximately 600 types of vernacular houses in China according to the statistics collected by the Ministry of Housing and Urban-Rural Construction of the People's Republic of China in 2014. Chinese traditional architecture can be divided into eight genres: the Beijing courtyard house, Huizhou architecture, the vernacular houses of southeast China, Lingnan architecture, Shanghai architecture, Western Sichuan vernacular houses, Qionglong architecture, and Academy buildings. Examples of the genres are shown in Fig. 1. The locations of these genres in China are shown in Fig. 2.

Huizhou architecture is one of the most important genres of traditional Chinese architecture and the most valuable genre to traditional Chinese culture because of its unique architectural styles and outstanding architectural achievements. Huizhou architecture is related to the architectural genres in the Anhui, Zhejiang and Jiangxi provinces and their surrounding areas as shown in Fig. 2. Huizhou architecture reached its peak during the Ming and Qing Dynasties of China. A typical Huizhou architecture has white walls, black tiles and Matou walls as shown in Fig. 3. Clansman residents with different social statuses, such as

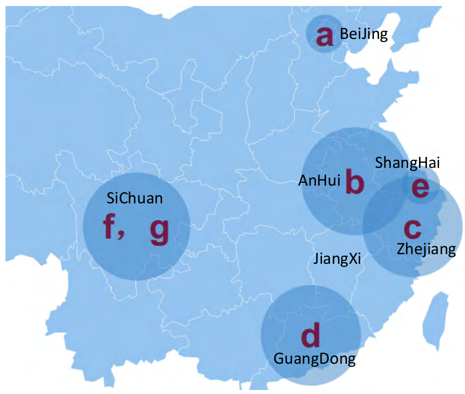


FIGURE 2. Locations of the eight genres of Chinese architecture in China.

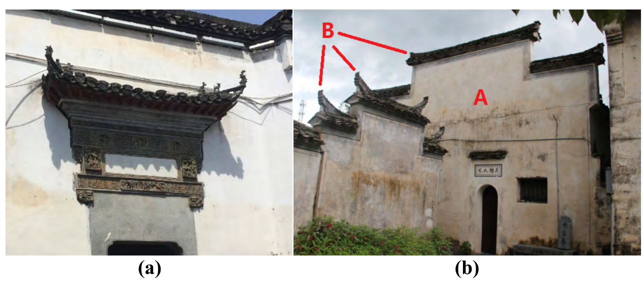


FIGURE 3. Typical decorations of Huizhou architecture: (a) gatehouse; (b) white wall, which is indicated by A, and Mar tau wall (top component of the white wall, also known as horse-head wall), which is indicated by B.

officials, businessmen or Confucian scholars, lived together under a strict patriarchal clan hierarchy and each house had a unique decorative style. Fig. 4 shows several examples of Huizhou traditional vernacular houses.

Most of the building components of Huizhou architecture

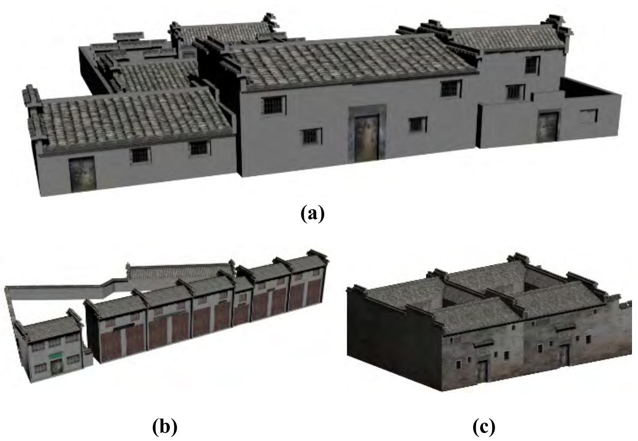


FIGURE 4. **Different types of Huizhou architecture vernacular houses: (a) Confucianism; (b) businessmen; and (c) officials.**

are wood, including the beams, columns, purlins, doors, etc. As a result, the original Huizhou architecture rarely endures over long-term periods. The Chinese government is becoming increasingly interested in the digital protection and reproduction of Huizhou architecture. Unsurprisingly, RMB is the most important and effective technology for this task.

Several professional modeling software systems are available to model Huizhou vernacular houses, such as Autodesk 3ds Max, Maya or Google SketchUp. However, such methods are tedious and require the user to have well-developed artistic skills. Moreover, the quality of the results relies on the user's ability.

To improve the modeling efficiency, researchers have proposed several methods of automating the modeling task and increasing the computational efficiency. However, in the case of Huizhou architecture modeling, current methods have the following limitations: 1) The majority of the procedural modeling methods are applicable only to urban city buildings (e.g., modern skyscrapers) or require that the user edit the grammar; 2) image-based modeling methods rely heavily on the user's interaction or require the user to have specific skills and applicable background knowledge, e.g., hand-drawn skill; and 3) machine learning methods require many training samples or considerable pre-processing.

In this paper, we propose a novel method for the rapid modeling of Huizhou traditional vernacular houses. Using our method, reasonable Huizhou vernacular house groups can be generated rapidly by users without prior training, a background in architecture design or modeling skills. Moreover, the proposed method is not limited to modeling Huizhou vernacular houses but can be extended to other types of historical buildings.

The main contributions of this work are summarized as follows:

1. A novel end-to-end pipeline for rapid modeling and layout of Huizhou vernacular houses is proposed. Particularly, the proposed system is very suitable for those users without any architecture or modeling background.
2. Layout guidelines based on 2D geometry rules are incorporated in an evaluation function to guide the layout generation. Metropolis-Hastings algorithm and sequential annealing method are used to find the best layout solution.
3. The layout results and conflict penalty functions are used to generate and evaluate the house features. The similar optimization framework mentioned in the layout generation step is used here to find the best feature results.
4. The layout and generated features are implemented with graphics processing unit (GPU) to parallelize and accelerate the code to meet the needs of real-time interactions with the system, and perturbations are optimized using a dynamic probability strategy with a Softmax function.

The remainder of this paper is organized as follows: in Section II, we briefly review several modeling methods; in Section III, we provide an overview of the proposed method; in Sections IV and V, we present the layout generation and feature generation of vernacular houses; in Section VI, we provide implementation and modeling results to demonstrate the proposed method; and in Section VII, we discuss the study limitations and future research topics.

II. RELATED WORK

Object modeling and object layout are the primary tasks of scene modeling [18]. In this section, we present recent academic work on these two tasks.

A. RAPID MODELING OF BUILDINGS

Of the available methods for RMB, procedural modeling techniques are effective methods of rapidly generating buildings and urban cities [3]. Based on the L-system, Parish and Müller [8] proposed the CityEngine system to generate a roadmap and urban area. Wonka *et al.* [19] proposed split grammars to design and generate buildings and urban cities. Duarte [20] introduced a discursive grammar for building Siza's houses in Malagueira that consisted of a programming grammar and a designing grammar. Müller *et al.* [4] proposed a CGA (computer graphic architecture) shape based on shape grammar to model buildings that was successfully used to recreate the archaeological site of Pompeii. Schwarz and Müller [7] extended the above method to CGA++, which allowed coordination across multiple shapes using an event mechanism. In recent work, Edelsbrunner *et al.* [21] extended shape grammar to cylindrical and spherical coordinate systems for generating structures with round geometries. Hua [22] proposed a bi-directional procedural model to incorporate geometric-topological reasoning.

With regard to modeling Chinese ancient architecture (CAA), Liu *et al.* [23] was the first to propose a constructive grammar for CAA. Additionally, Liu *et al.* [24]–[27] applied an ontology-based approach to the modeling process of ancient Chinese architecture and semantic units.

In addition, they built a knowledge library of southeast Chinese vernacular houses (see Fig. 1(**c**)), which included information provided by architectural domain experts. Huang and Tai [28] developed a user-friendly interactive system to build Chinese tings. Using this system, users with a limited Chinese architecture background can complete the modeling process by operating control-points via user guidance and interaction clues.

Of the data-based modeling methods, image-based modeling is an effective method to provide a user-friendly and convenient human-machine interaction. "Sketching reality" [29] is a method for converting a free-hand building sketch into a realistic-looking texture-mapped 2.5D model. Using the structural symmetry of architecture, Jiang *et al.* [12] proposed a single image-based modeling method in which the user must mark all the architectural structures explicitly via computer interactions. Hou *et al.* [30] presented a modeling framework to model Chinese architectures via elevation drawing. Additionally, Hou *et al.* [31] integrated the single-view architectural modeling method with the procedural modeling method by using a novel algorithm to adaptively segment repeated curved stripes of Chinese architecture.

Recently, machine learning methods have been widely used for RMB. For example, Merrell *et al.* [32] used a probability network to describe the high-level requirements of architecture. Based on a probability network, Fan and Wonka [33] used simulated annealing to generate the roof of a house and a genetic algorithm to generate the facade features of the house. Nishida *et al.* [34] introduced deep learning techniques into RMB. For each object type of a building's components, a convolutional neural network (CNN) was trained to recognize the sketched drawing and then generate the parameters.

B. 2D LAYOUT METHOD

The layout problem, also known as the spatial allocation problem, is an important aspect of modeling multiple objects, and it can be treated as a 2D plane placement problem. The problem has an exponential complexity and is difficult to solve directly. A heuristic search method [35] is commonly used to identify an optimal solution to the layout problem. Another important method is stochastic optimization, such as simulated annealing [36] and genetic algorithms [37]. These classical algorithms are widely used to solve small-scale layout problems, especially those involving indoor scenes. Merrell *et al.* [32] used a probability network to guide the layout generation of a single house. Camozzato *et al.* [38] proposed an automated generation method that generates 2D models of building interiors from hand-drawn building sketches, and the outline and openings of the buildings were extracted via image processing.

To solve large-scale building layout problems. Guerrero *et al.* [39] proposed a learning model based on probability and statistics. The system can generate complex scenes using examples from the user by learning and propagating shape placements in 2D polygonal scenes, and it requires only limited user interaction. To ensure that the generated layouts meet both accessibility and aesthetic criteria for arbitrary shapes, Peng *et al.* [40] developed a tiling domain method with a set of deformable templates.

Furniture arrangement is another common 2D layout problem. Yu *et al.* [41] proposed a system to learn the hierarchical and spatial relationship information from examples of indoor scenes and then used a Metropolis-Hastings search to determine the final placement of components. Merrell *et al.* [42] incorporated the guidelines for furniture layout as terms in a density function using a hardware-accelerated Monte Carlo sampler to search for the optimal solutions.

In addition, a classical 2D layout problem is tag cloud generation. An important problem of tag cloud generation is overlap removal and relative positioning among 2D objects [43]–[46]. Additionally, such algorithms can be used in building layout modeling.

III. OVERVIEW

In our method, Huizhou vernacular houses are automatically generated by inputting the initial frontage data of the houses into the system. An overview of our method is illustrated in Fig. 5, and the pipeline is outlined as follows.

A. INPUT DATA PRE-PROCESSING

The initial frontage data of houses and road data are preprocessed. The data can be obtained by user interactions with the system or by image processing. In this stage, users can specify the features they do not want to change, which is useful when modeling a specific target.

B. LAYOUT GENERATION

The layouts of vernacular houses are optimized. A hardware-accelerated Monte Carlo sampler method is used to minimize the energy function, which is incorporated by several layout guidelines based on 2D geometry rules. The layout results are used in the next stage.

C. POST PROCESSING

The spatial information among the neighboring houses, such as adjacency relationships, is extracted from the layout results generated in the previous stage. These results will be calculated using the CPU and then used in the next stage.

D. FEATURE GENERATION

The primary task is to generate the features of the vernacular houses. For each vernacular house, a probabilistic network is used to generate rational features. Moreover, penalty functions are used to avoid feature conflicts that might occur among multiple vernacular houses. The optimization method for the penalty function is similar to that of stage 2.

E. HOUSE GENERATION

The layout and feature results from the above steps are described using a Huizhou architecture constructive

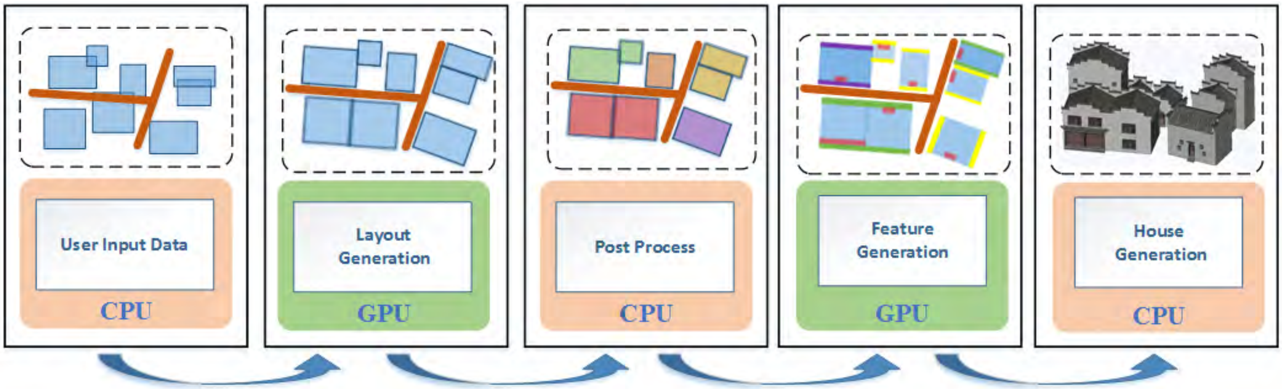


FIGURE 5. Overview of our method. First, the user inputs the initial layout data, including the position and rotation information of the houses and roads. Then, the layout information is calculated in the GPU stage using a hardware-accelerated Monte Carlo sampler. In the post process stage, the adjacency information of the houses is extracted from the layout data. The features of the houses are then generated using the data from the previous stage to avoid feature semantic conflicts. Finally, the houses are generated in the last CPU stage.

grammar. All the vernacular houses are generated automatically by executing the grammar.

IV. LAYOUT GENERATION

A. LAYOUT REPRESENTATION

The primary task of this step is to generate a plausible layout by inputting into the system the initial frontage data of the layout of vernacular houses. The initial layouts of the houses can be generated by user interactive modeling or obtained via image processing. For example, Canny edge detection can be used to extract house and road profiles or boundaries. The input data should be preprocessed to create a standardized representation.

The house layout problem can be treated as a 2D geometric problem. Each road can be described by several continuous line segments, and each segment can be assigned a width. In our method, a house is defined as a bounding box and can be calculated using the following formula:

$$\begin{pmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ 1 & 1 & 1 & 1 \end{pmatrix} = \mathbf{M}\left(\begin{pmatrix} x & x & x & x \\ y & y & y & y \\ 1 & 1 & 1 & 1 \end{pmatrix} + \begin{pmatrix} -0.5w & 0.5w & 0.5w & -0.5w \\ -0.5d & -0.5d & 0.5d & 0.5d \\ 0 & 0 & 0 & 0 \end{pmatrix}\right), \quad (1)$$

where x, y is the center of the box, w defines the width, d defines the depth, and θ defines the rotation angle. Other definitions of the parameters in (1) are illustrated in Fig. 6. Matrix $\mathbf{M}$ is composed of a translation and rotation matrix as follows:

$$\mathbf{M} = \begin{pmatrix} 1 & 0 & x \\ 0 & 1 & y \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & -x \\ 0 & 1 & -y \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta & x - x\cos\theta + y\sin\theta \\ \sin\theta & \cos\theta & y - x\sin\theta - y\cos\theta \\ 0 & 0 & 1 \end{pmatrix}. \quad (2)$$

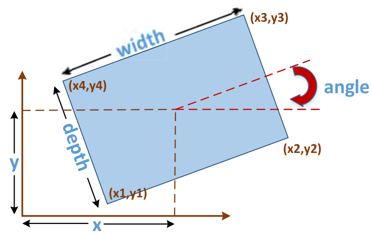


FIGURE 6. Description of a vernacular house.

B. LAYOUT GUIDING RULES

We designed a set of geometric rules to evaluate the layout result. Each rule is defined as a continuous function F, and all the rules are combined into an energy function Γ_{layout}. The energy function is defined as follows:

$$\Gamma_{layout} = \sum_{i=1}^{n} \omega_i F_i, \quad (3)$$

where the function F represents different rules and ω is the weight of each function. The resulting layout optimization is equivalent to the optimization of the energy function.

1) OVERLAP RULE

Overlap removal is the most important of the layout guiding rules. Fortunately, this rule is straightforward. Because our goal is to remove all overlaps, we must summarize all the overlapped area. The energy function for the overlap rule can be defined as follows:

$$F_{overlap} = \omega_{overlap} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} PIA(a_i, a_j), \quad (4)$$

where $\omega_{overlap}$ is a penalty factor andPIA is a function used to calculate the intersection area of two convex polygons (houses a_i and a_j). Fig. 7 shows an example of 3 overlapping houses.

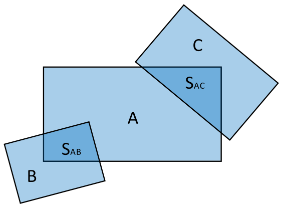


FIGURE 7. Intersection area of houses.

2) DISTANCE RULE

Each section of the road is described by a line segment R. Here, r represents the number of road sections and w_j represents each road width. The distance rule is used to control the distance between the road centerline and the house. The function can be defined as follows:

$$F_{road} = \omega_{road} \sum_{i=1}^{n} \sum_{j=1}^{r} f_{road} \cdot (\min_{k=1 \to m_i} DIST(p_{i,k}, R_j), d_{i,j}, k_{i,j}, w_j), \quad (5)$$

where m_i is the number of vertices of the bounding box polygon i, $DIST$ is a function used to calculate the distance from each polygon vertex to the road R_j, and $d_{i,j}$ is the shortest distance from the center of each house to the road. $k_{i,j}$ is an indicator, when the house intersects with the road, $k_{i,j} = 1$; otherwise, $k_{i,j} = 0$. f_{road} is a function that forces the distance from the road to the house to be approximately equal to w_j.

$$f_{road}(x, d, k, w) = \begin{cases} w/2 + r - d, & k = 1; \\ |x - w/2|, & k = 0,\ x \le w; \\ w/2 + \log(1 + x - w), & k = 0, x > w; \end{cases} \quad (6)$$

This function indicates that when the distance between the house and road is too great, a gravitational force is observed; otherwise, a repulsive force is observed. Moreover, this function can be related to the overlap between the house and the road for extreme values. To ensure function continuity, the function is defined as a piecewise function as depicted in Fig. 8.

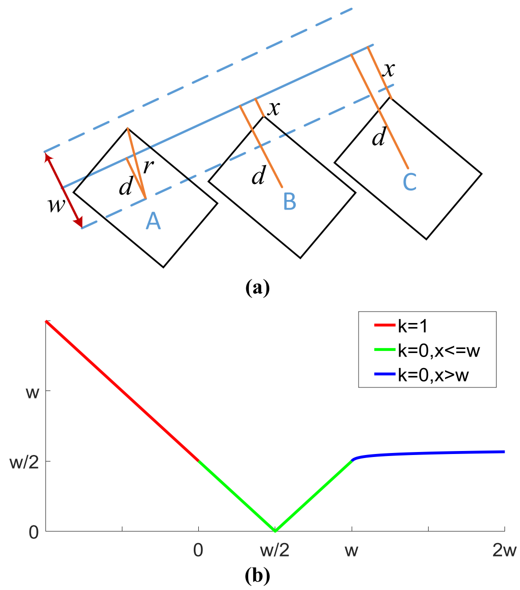


FIGURE 8. Distance rule for the house and road. (a) Different cases of the relationship between the house and road. (b) Function used to control the distance. The different functions in the piecewise function are represented by three colors.

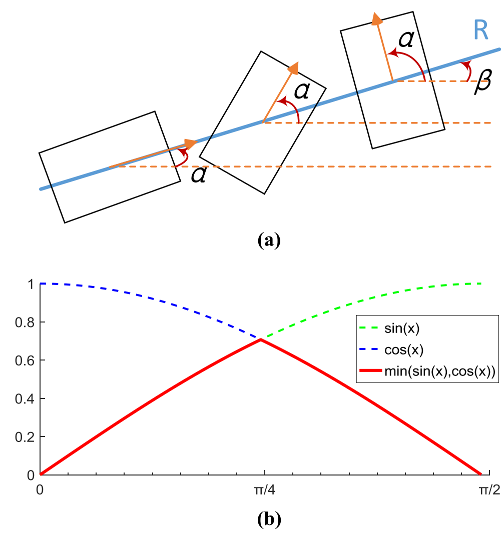


FIGURE 9. Rotation rule of the house and road. (a) Angle between the house and road. (b) Angle between the house and road is controlled by the function. The red line shows the final function, which forces the house to be parallel or perpendicular to the road.

3) ROTATION RULE

Usually, houses and roads should be parallel or vertical, and other conditions should be punished. The rotation rule is used to control the angle between the house and the road. Here, vector $\vec{\alpha}$ represents the house orientation and vector $\vec{\beta}$ represents the road orientation. The rotation rule is defined as follows:

$$F_{rotate}(\vec{\alpha}, \vec{\beta}) = \omega_{rotate} f_{rotate}(\min_{k=0 \to 2}(\left| \text{acos}(\frac{\vec{\alpha} \cdot \vec{\beta}}{\|\vec{\alpha}\| \cdot \|\vec{\beta}\|}) - \frac{\pi}{2}k \right|)), \quad (7)$$

where the ω_{rotate} is a weight factor, f_{rotate} is the rotation rule function which is depicted in Fig. 9. The vectors are limited to the first and fourth quadrants. The range of $acos$ is $[0, \pi]$; therefore, the angle between the house and the road can be calculated by the circulation decline $\pi/2$.

Another optional rule is to control the angle between neighboring houses. This rule is similar to the rotation rule between the house and the road; however, the weight of the function is different.

4) OPTIONAL RULES

In addition, several optional rules can be used to create varied layout results. These rules include but are not limited to the following.

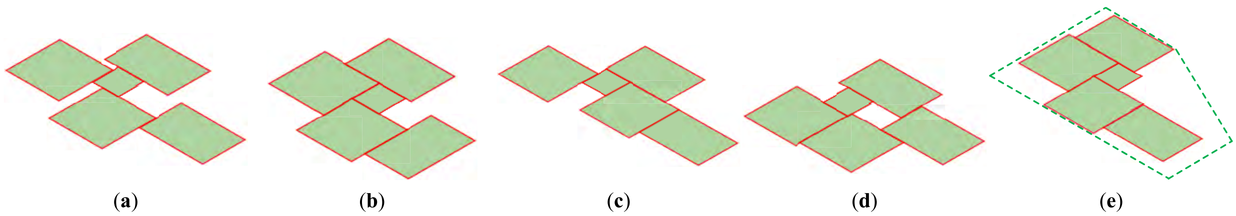


FIGURE 10. Layout results using different rules. (a) Original layout. (b) Minimum convex hull area rule only. (c) Minimum of the sum of neighbor distance rule only. (d) Combination strategy of the convex hull and neighbor distance rule. (e) Combination strategy of the neighbor distance and boundary constraint rule.

a: BOUNDARY CONSTRAINTS

This rule is used to ensure that all vernacular houses stay inside the user-specified boundary. The implementation of this rule is the same as that for the overlap rule.

b: SUM OF NEIGHBOR DISTANCE

Minimizing this value arranges the vernacular houses close together, and it can ensure that no ''holes'' appear in the final layout.

c: CONVEX HULL AREA

All of the vertices of vernacular house bounding boxes compose a convex hull. Minimizing the area of the convex hull is another method of arranging the houses close together.

Different layouts can be generated by using different combinations of rules or by using different parameter weights that are either set empirically or depending on the model application. Fig. 10 shows the results from applying different rules in the system.

C. PERTURBATION OF LAYOUT

Although directly optimizing the energy function is difficult, stochastic optimization represents a suitable method of resolving this problem. We use a Markov chain Monte Carlo sampler [47] to search the energy functions for an optimized solution. The Metropolis-Hastings algorithm [48] is used to explore the density function of each parameter of the houses.

A layout configuration representing the state of the system, can be defined as $\phi = \{x_i, y_i, \theta_i | i = 1, \ldots, n\}$, where x_i, y_i, θ_i define the position and rotation parameters of n houses. In each iteration of temperature, we ''move'' the current layout configuration ϕ to get a new layout configuration ϕ^*, and accept it with a Metropolis-Hastings acceptance probability:

$$\alpha(\phi^*|\phi) = \min\left(1, \frac{f(\phi^*)}{f(\phi)}\right), \tag{8}$$

where f is a Boltzmann-like objective function

$$f(\phi) = \exp(-\beta\ C(\phi)), \tag{9}$$

where β is a constant inversely proportional to the temperature and C is the energy function defined in (3). Now we get:

$$\alpha(\phi^*|\phi) = \min\left(1, \exp(\beta\ (C(\phi) - C(\phi^*))\right). \tag{10}$$

The system anneals with decreasing temperature. Note that the energy may increase when accepting a ''bad'' move, which enables the method to escape from local minima.

In each move step, we choose a single house at random and perturb the parameters with a given probability as follows.

1) *Position*
   A Gaussian term $\mathcal{N}(0, \sigma_p^2)$ is added to the center coordinate x, y of a house to change its position. The position perturbation will affect the overlap and distance relationships among the houses and roads.
2) *Rotation*
   A Gaussian term $\mathcal{N}(0, \sigma_\theta^2)$ is added to the parameter θ of the house to change its orientation. Rotation perturbation will directly affect the rotation energy function and all the position and rotation calculations.

The values of the standard deviations σ_p and σ_θ are set empirically. If these values are too small, then many iterations will be required for the function to converge. If these values are too large, then finding an optimal solution might be impossible. In addition, the perturbation of parameter w, d is optional depending on whether the user assumes that the size of the house is fixed.

D. DYNAMIC PROBABILITY STRATEGY

The classical algorithm employed in [42] uses an equal probability to choose the proposal perturbation move. However, we use a dynamic probability strategy. Because rotation also affects the house position, the rotation-related component of the energy function should converge first. We use the following two strategies to accelerate the convergence of the energy function.

1) Priority strategy
   This strategy ensures that each rotation perturbation has a higher influence factor on the energy function than on the position perturbation. That is, rotation has a higher priority than position, which is achieved by increasing the $\omega_{overlap}$ parameter in formula (7).
2) Dynamic probability
   Because the rotation energy converges faster than the position energy, the probabilities of rotation perturbation and position perturbation should change dynamically. For example, if the rotation energy has reached its optimal value, then it is not necessary to set a probability value for the rotation perturbation.

We use $f^*(x)$ to define the maximum rate of energy change within a window of size k as follows:

$$f^*(x) = \frac{\underset{i\in[i-k,i]}{Max}\ x_i - \underset{i\in[i-k,i]}{Min}\ x_i}{\underset{i\in[i-k,i]}{Min}\ x_i}. \tag{11}$$

Now, we can define E_p for the position-relative energy function and E_r for the rotation-relative energy function. The probability of position perturbation in the $i+1$ iteration can be calculated using the following formula:

$$P^{i+1}_{position} = \frac{f^*(\mathrm{E_p}) + 1}{f^*(\mathrm{E_p}) + f^*(\mathrm{E_r}) + 2}. \tag{12}$$

The probability of rotation perturbation is $1 - P^{i+1}_{position}$. Notice that we use the Laplace smoothing method to increase the robustness of the algorithm. We have found that better results are achieved using the Softmax function as follows:

$$P^{i+1}_{position} = \frac{e^{f^*(E_p)}}{e^{f^*(E_p)} + e^{f^*(E_r)}}. \tag{13}$$

The Softmax function is much more robust, and it can make the probabilities more diffuse in position and rotation perturbation.

When $i < k$, we use an equal probability strategy to choose the possible proposal perturbation move. When $i \geq k$, we use the aforementioned dynamic probability.

V. FEATURE GENERATION

In this stage, the layout results are used to generate the house features. These features are extracted from historical architectural pattern books, which define the architectural rules and details.

A. CONSTRUCTIVE GRAMMAR OF THE HUIZHOU VERNACULAR HOUSE

The structure of the Huizhou traditional vernacular house can be determined from the current literature on ancient Chinese wooden architecture [49]–[51] and the features of Huizhou architecture [52], [53]. A Huizhou vernacular house can be reconstructed via a decomposition and combination of its components. A typical Huizhou traditional vernacular house structure [54] is shown in Fig. 11. The wall, Matou wall, door, window, gatehouse and door step are the most representative components of a Huizhou traditional vernacular house. Additionally, these features are the key elements for feature generation.

The procedure to combine components is controlled by grammar. Our Huizhou architecture constructive grammar (HACG) is based on the constructive grammar (CG) [23]. However, we make the following improvements to the CG.

1) OBJECT ORIENTED DESIGN

We use an object and inheritance mechanism to design the HACG, which is different to the template language used in [54]. The parameters of the components are treated as properties while the connection rules between components are

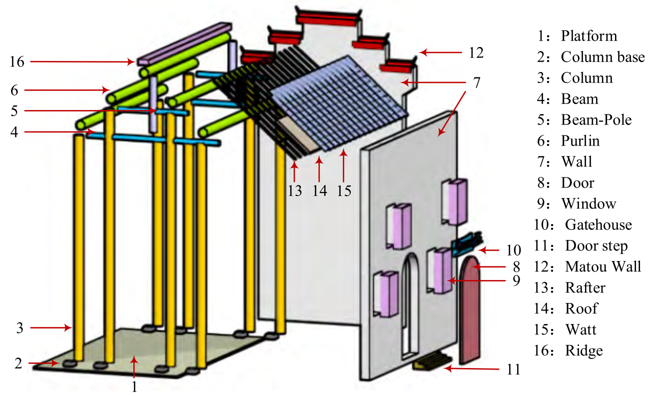


FIGURE 11. Structure of a Huizhou vernacular house.

designed as interfaces. Table 1 lists the primary properties and interfaces of the components of a Huizhou vernacular house. Each instantiated entity of a component can communicate with other instantiated entities of a component to coordinate their characteristics. This communication mechanism was inspired by the event mechanism of CGA++.

2) MULTI-LEVEL PARAMETERS

Because of the large number of building components, the user must control many parameters to model a house. To reduce user workload, we designed a multi-level parameter system in which the user is required to control only a few high parameters while the other parameters are calculated automatically by our system. The high parameters are controlled by the feature probabilistic network, which will be discussed in the next section.

B. FEATURE PROBABILISTIC NETWORK

"Hui-style features" are the most important characteristics of Huizhou vernacular houses, and they have specific semantics. For example, the Matou wall, gatehouse and other components vary widely depending on the identity of the house owner. For example, for houses with higher floors and costs, the Matou wall will have more layers. In addition, the layers of the Matou wall, the type of the house wall and the identity of the house owner will influence the type of Matou. To ensure that the feature generation conforms to the appropriate semantics, we use a probability network in [54] to describe these relationships. An example of a probability network for a Huizhou vernacular house is shown in Fig. 12. In this example, each node takes a discrete value to represent a feature. For a continuous value, discretization or the Gaussian mixture model (GMM) are also appropriate. We generate this network from background knowledge and the famous Huizhou villages Xidi Hongcun from southern Anhui Province. The primary features of the components of a Huizhou vernacular house and their directly related probabilistic network nodes are listed in Table 1. The features that users cannot access directly are not listed.

For a single vernacular house, all features can be generated using Gibbs sampling by the probabilistic network or by

TABLE 1. Main parameters of the components of a Huizhou vernacular house.

Component	Parameters	
	Property parameters	Connection rules
Platform	Width, Height, Thickness, Ratio	Rule (ColumnBase, DoorStep, Wall)
Column base	Style, Width, Height, TopInRadius, BottomInRadius, OutRadius	Rule (Column)
Column	Style, Width, Height, Length, Radius	Rule (Beam, Purlin)
Beam	Style, Width, Height, Length, Radius	Rule (BeamPole, Purlin)
Beam-pole	Width, Height, Length, Radius	Rule (Purlin)
Purlin	Width, Height, Length	Rule (Rafter)
Wall	Style, Thickness, sideWallLength, maxSideWallLength, eavesHeight, rooftopHeight, NumberOfPlies, Ratio	Rule (Door, Window, Gatehouse, DoorStep, MatouWall)
Door	Style, Width, Height, BorderWidth, ArcRadius	Rule (DoorStep, Gatehouse, Window)
Window	Style, Width, Height, BorderWidth	Rule (DoorStep)
Gatehouse	Style, Width, Length, Height	
Door step	Style, NumberOfStep, Width, Length, Thickness, FaceLength, NeckInLength, NeckInWidth	
Matou Wall	Style, Width, Height, BaseLength, UpLength, UpHeight, Tail	Rule (Roof)
Rafter	Width, Height, Length	Rule (Watt, Ridge)
Roof	Width, Height, Length	Rule (Ridge)
Watt	Width, Height, Length, Thickness	
Ridge	Width, Height, Length	

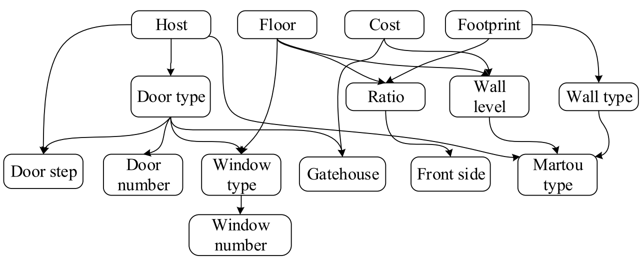


FIGURE 12. Probabilistic network of a Huizhou vernacular house. Each node uses a discrete value to represent a feature.

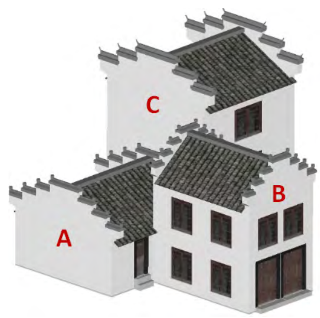


FIGURE 13. Example of feature conflicts among multiple houses. The door position, Mar tau wall level and house style are unreasonable because of the feature conflicts with neighboring houses.

user interactions using a recommended system [54], [55]. However, for multiple houses, we must resolve the feature conflicts among different houses in each Huizhou vernacular house groups, which will be discussed in next section.

C. FEATURE GUILD RULES

The feature conflicts usually occur among adjacent vernacular houses because the adjacent houses must coordinate. Moreover, these houses usually belong to a single family and should have consistent characteristics.

For example, consider the 3 houses in Fig. 13. The door of house A should not be on the right side of the house because it would be blocked by house B. Additionally, house A is shorter than houses B and C; therefore, its Matou wall should have fewer layers than houses B and C. In addition, the three houses should be included in a single family. However, the features of house A show that the owner of the house is a farmer while the features of houses B and C show that owners are businesspeople. All of these cases should be punished in the feature generation.

All the features that should be considered are listed in Table 2. We can observe that the majority of these features are controlled by the probabilistic network when no conflict occurs. When a conflict occurs, we must extract the spatial information from the layout result in the post process.

TABLE 2. Parameters used in the generation of a Model of a Huizhou vernacular house.

Parameter	Domain	Related Probabilistic Network Node
X	$\mathbb{R}$	n/a
Y	$\mathbb{R}$	n/a
Width	$\{300 \le Width \le 2000, Width \in \mathbb{R}]$	Footprint
Depth	$\{300 \le Depth \le 2000, Depth \in \mathbb{R}]$	Footprint
Angle	$\{-90 \le Angle \le 90, Angle \in \mathbb{R}]$	n/a
Host	$\{1 = 'confucianism', 2 = 'businessmen', 3 = 'officials'\}$	Host
Cost	$\{1 = 'poor', 2 = 'medium', 3 = 'rich'\}$	Cost
Floor	$\{1, 2, 3\}$	Floor
DoorPosition	$\{1 = 'front', 2 = 'back', 3 = 'left', 4 = 'right'\}$	n/a
WindowPosition	$\{1 = 'front', 2 = 'back', 3 = 'left', 4 = 'right'\}$	n/a
MatouPosition	$\{1 = 'front / back', 2 = 'left / right'\}$	MatouPosition
MatouLevel	$\{1, 2, 3, 4, 5\}$	MatouLevel
WallType	$\{1 = 'symmetrical', 2 = 'step'\}$	WallType
MatouType	$\{1 = 'normal', 2 = 'sit\ bucket', 3 = 'tail'\}$	MatouType
GateHouseType	$\{1 = 'none', 2 = 'simple', 3 = 'whole'\}$	GateHouse
DoorNumber	$\{1 = 'none', 2 = 'single', 3 = 'double'\}$	DoorNumber
WindowNumber	$\{1 = 'none', 2 = 'few', 3 = 'more'\}$	WindowNumber
StepNumber	$\{0, 1, 2, 3, 4, 5\}$	DoorStep

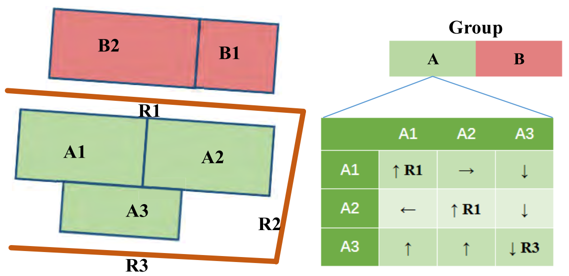


FIGURE 14. Neighbor information of the house groups. In this example, A2 is on the right of A1, which means that the "right" tag is marked in the row of A1 and column of A2. Meanwhile, the nearest road to A1 is R1, which is recorded in the diagonal of the matrix.

The most important information is the adjacency information. We use a disjoint-set data structure to quickly divide the houses into familial groups. For each familial group, we use a matrix to store the adjacency information of the houses. The adjacency information could be a neighbor's direction or distance depending on the application of the model. In addition, the diagonal elements of the matrix store the nearest road from each house. See Fig. 14 for details.

Using this adjacency information, feature conflicts can be quickly detected without extensive geometry calculations. The feature conflict rules can be specified by several penalty functions. The framework used in the layout stage is used to search for the best combination of features via the GPU.

D. PERTURBATION OF FEATURES

As with layout perturbation, in each iteration, we choose a single house at random and perturb a random feature. Because the feature values are discrete, 2 methods are used to change the feature value.

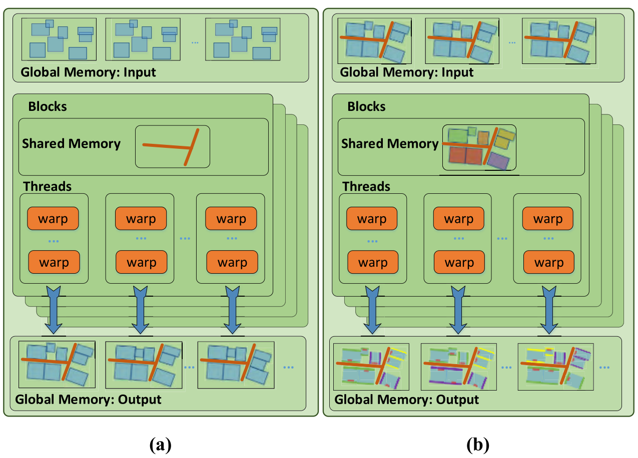
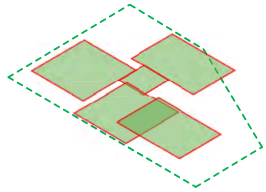
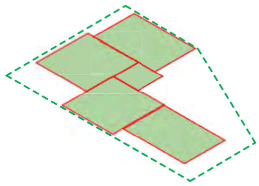
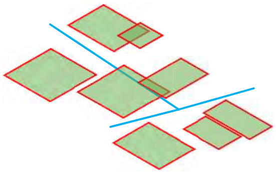
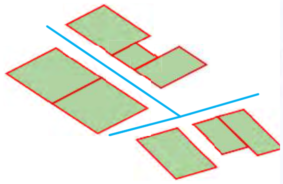
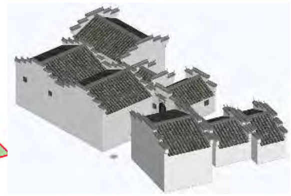
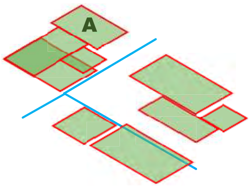
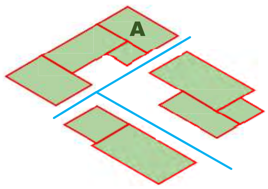
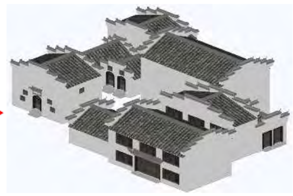
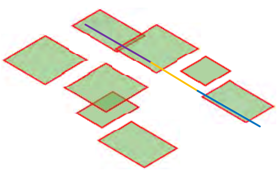
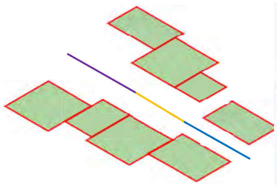
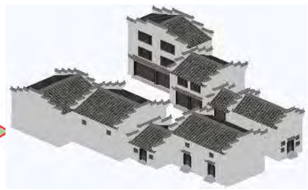
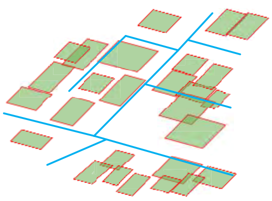
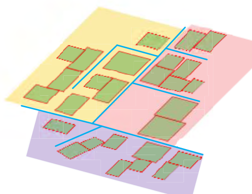
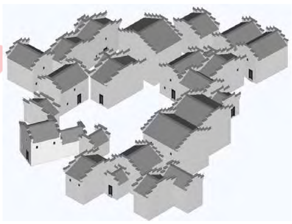


FIGURE 15. CUDA framework of the layout generation and feature generation implementation. (a) Layout stage, in which the shared memory stores the additional public information, such as roads. (b) Feature generation stage, in which the shared memory stores the neighbor matrix information. The global memory is used to store the input and output data. Each thread in a block will generate a result, and the best result among these will be the final result.

1) **Increment and modulus**
   In this method, an increment is added to the feature parameter using a modulo operation to ensure that the result is in the domain of the feature.
2) **Change randomly**
   In this method, a value is randomly selected within the definition domain of the feature.

TABLE 3. Modeling results and performance.

Input	Layout generation result	Feature generation result	Computational cost (10^{th} average) Generation	Time(s)	Iteration	Energy
			Layout (traditional Metropolis-Hastings)	1.36	175	3176
			Layout (Dynamic strategy)	1.31	166	2657
			Feature	0.19	24	0
			Layout (traditional Metropolis-Hastings)	1.23	239	5174
			Layout (Dynamic strategy)	1.07	197	4862
			Feature	0.23	39	0
			Layout (traditional Metropolis-Hastings)	1.53	241	8667
			Layout (Dynamic strategy)	1.15	165	7042
			Feature	0.31	58	0
			Layout (traditional Metropolis-Hastings)	1.37	241	5365
			Layout (Dynamic strategy)	1.12	237	5012
			Feature	0.25	45	0
			Layout (traditional Metropolis-Hastings)	1.36 1.58 1.25	241 241 238	7253 8174 6527
			Layout (Dynamic strategy)	1.14 1.18 1.09	175 192 168	6584 7348 5846
			Feature	0.22 0.29 0.21	39 48 34	0 0 0

Because of the use of adjacency tables, we are able to avoid extensive geometric calculations. Using the Metropolis-Hastings algorithm in the feature generation stage is faster than using layout generation.

In addition, advanced users can specify the specific features of particular houses. These features will be marked as constants; consequently, the perturbation process will skip these features during the feature generation stage.

VI. RESULTS

A. IMPLEMENTATION

To meet the real-time requirements of user interactions with the system, we use CUDA to implement the GPU parallel acceleration framework. Each thread is treated as a single Markov random process. We run a Metropolis-Hastings algorithm to determine the minimum energy value. After all threads have completed their tasks, we choose the smallest energy value of all threads as the final result. The framework is depicted in Fig. 15.

We implemented the Monte Carlo sampler and performed all experiments on a PC with a quad-core CPU running at 3.60 GHz with 8.0 GB of memory and an NVIDIA GTX 760 graphics card. The layout generation step is a time-consuming stage that includes intensive geometric operations; therefore, we set the number of blocks to 6 and the number of threads to 32. During the feature generation stage, the number of blocks and threads can be set to larger values. The max iteration times of temperature are limited to 241, and in each temperature iteration we do the perturbation 30 times.

B. MODELING RESULTS

Using our system, the modeling process is rather simple. The user must specify only the draft road and the initial frontage data of the houses. Then, the work is performed

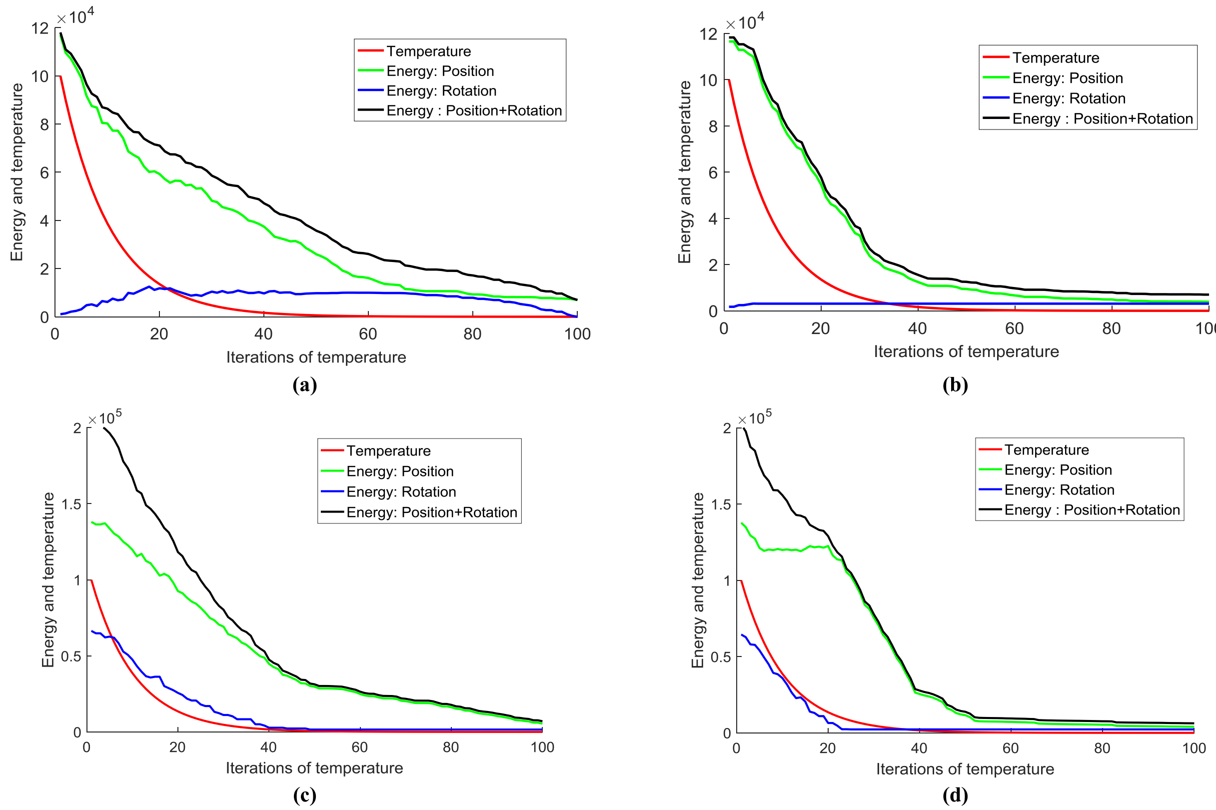


FIGURE 16. Results of the temperature and energy functions in first 100 iterations. The data in row 1-2 are obtained from the modeling results in Fig. 16, row 1-2. (a, c) Equal probability is used to select the possible proposal perturbation move, and (b, d) dynamic probability strategy is used.

automatically by the system. Table 3 shows the results of several modeling outputs generated by the proposed system. The first row shows an example with a boundary constraint, and the second row shows an example with roads. In the third row, the user specifies an officer style house (marked by A), and its neighbors are also represented by an officer style. In the fourth row, the results show an example, in which 3 sections of roads have different widths. The last row shows an example of 27 houses, and it is divided into 3 regions with different colors. The first column of Table 3 shows the original data input. The second column shows the optimized layout results. The third column shows the feature generation results. The last four columns give the performance data of both layout generation and feature generation. We compare the proposed dynamic probability layout model with traditional Metropolis-Hastings algorithm in [41] and [42] to evaluate the computational efficiency. The time column gives the computational time (second) of each model. The iteration column gives the temperature iteration times. Note that iteration times less than 241 means that the system has found a minimum energy value, therefore, the iterative process can be terminated prematurely. The energy column gives the minimal energy value in the end of the iteration. The data shows that our dynamic probability strategy model is able to find better layout results with shorter time and less iterations.

Both the layout and feature generations take approximately 1 or 2 seconds, which can meet the real-time interaction requirements. However, generating a relatively large vernacular house village using our layout generation method might require a considerable amount of time. A better option is to divide the scene into multiple regions and generate each region individually. The last row in Table 3 shows an example with 3 sub-regions.

Fig. 16 shows the changes in the minimum energy value after each temperature iteration. The red line indicates the change in temperature, and the green line indicates the energy of position and the blue line indicates the energy of rotation. Notably, the energy value of the rotation converges faster than the energy value of the position because of its higher perturbation priority. Fig.16 (a) and (b) show that when there is no need to change the rotation for each house, the traditional Metropolis-Hastings has poor performance because it always gives half the chance to the rotation perturbation. Our method has better performance on suppression of unneeded rotation perturbation. Fig. 16(c) and (d) show examples consider both the position and rotation. The energy value approaches the optimal value at approximately the 100th iteration when using the equal probability strategy while it approaches the optimal value at approximately the 60th iteration when using the dynamic probability strategy.

To generate a target model, the user must specify the desired features according to the target picture. In this case, the feature generation step can be omitted. Fig. 17 (a)-(c) shows the result of modeling using the target picture from

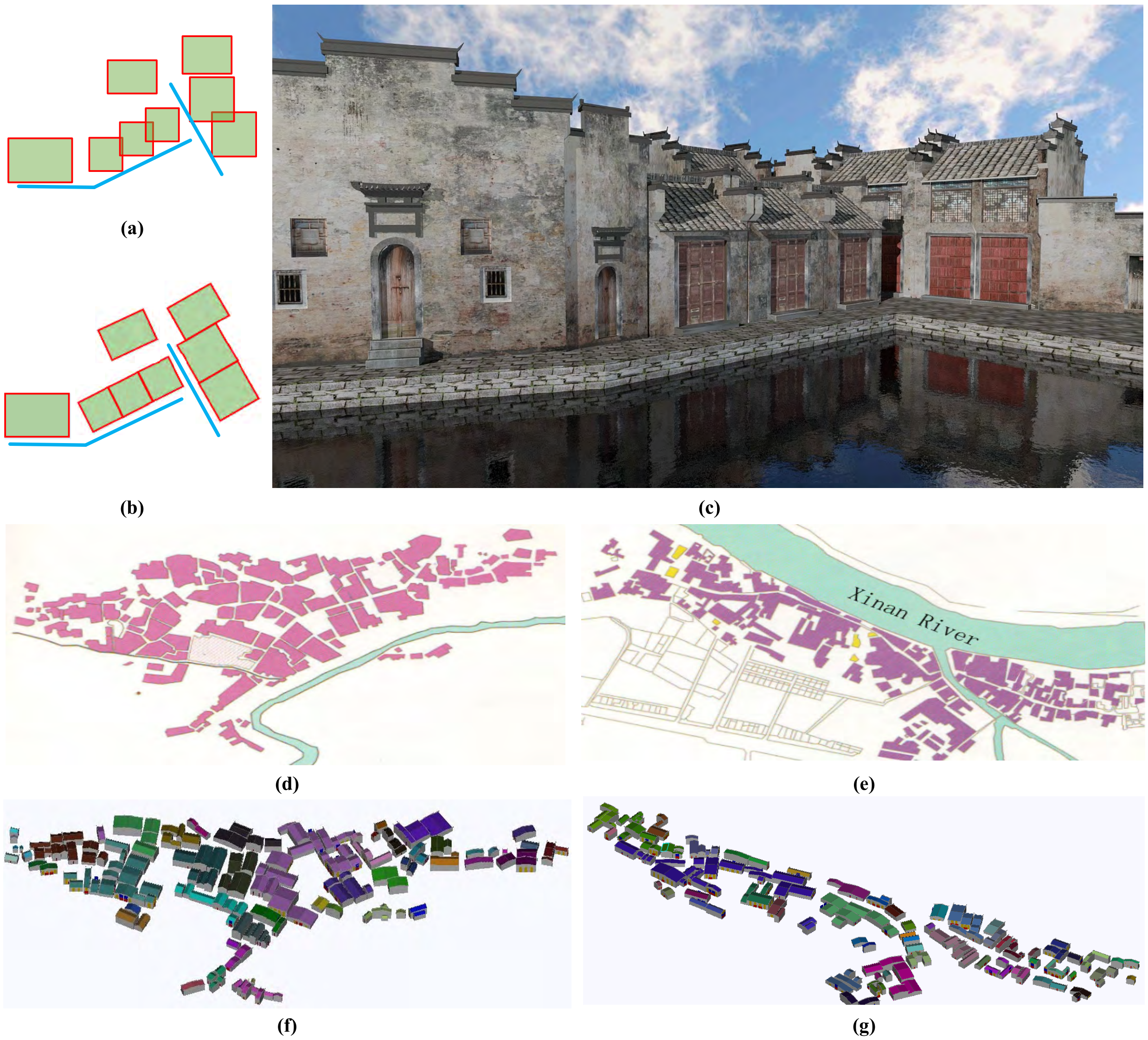


FIGURE 17. Examples of modeling with a target picture using our method. (a) Draft layout input from the Xidi Hongcun ancient village by user interaction. There are 8 houses and 3 section of roads. (b) Layout generation result. (c) Final result, which is rendered using a realistic style. (d) Sketch map for the Zhanqi cun ancient village in southern Anhui. (e) Sketch map for the Wan'an cun ancient village in southern Anhui. (f) Subdivision and generation result of (d), which contains 153 vernacular houses. (g) Subdivision and generation result of (e), which contains 161 vernacular houses. Note that the different vernacular house groups are marked by different colors.

Fig. 1(b). Fig. 17 (d)-(g) shows two large examples of Huizhou ancient traditional villages, which were generated by subdividing the large scenes into multiple vernacular house groups. The sketch map for the ancient villages was adopted from [53]. More than one hundred vernacular houses are included in these two villages, and the model can be generated in several minutes.

C. COMPARISON TO PRIOR WORK

Prior work on RBM has implemented various the applications and technology. It is common knowledge that the modeling result generated by procedural modeling method is typically assessed by qualitatively evaluating or by conducting a user study [56]. We list the characteristics of the previously used methods in Table 4 and compare these methods based on the following factors.

Constructive rule: This factor determines whether the modeling method has included the details of the building construction rules, which play an important role in HBIM.

Multiple coordination: This factor indicates whether the modeling method can coordinate multiple model features automatically.

User interface: This factor indicates how users control the modeling process using the modeling system.

Derivation and variation: These factors indicate the ability to generate models via derivation and variation.

The majority of the shape grammar based methods can generate models via derivation and variation. However, the user must operate the grammar directly (e.g., by editing the script). The majority of the image-based modeling

TABLE 4. Comparison of our system with other systems.

Modeling system	Method	Application	Constructive rule	User interface	Multiple coordination	derivation and variation
Wonka et al. [19]	split grammar	general architecture	N	rules (L-system)	N	high
Liu et al. [23]	constructive grammar	Chinese historical building	Y	rules(XML)	N	high
Liu et al. [27]	semantic modeling	Chinese vernacular house of Southeast	Y	Rules (hash-table)	N	high
Müller et al. [4]	shape grammar	archaeological site of Pompeii	N	rules(CGA)	Y	high
Jiang et al. [12]	single image modeling	Chinese historical building	Y	user-drawn strokes.	N	low
Hou et al. [30]	drawing-based procedural modeling	Chinese historical building	N	user-drawn segments	N	low
Huang et al. [28]	procedural modeling	Chinese ting	Y	control-points	N	high
Schwarz et al. [7]	shape grammar	general architecture	N	rules (CGA++)	Y	high
Hou et al. [31]	single image modeling	Chinese historical building	Y	user-drawn segments	N	low
Li et al. [54]	procedural modeling probabilistic network	Single Chinese Huizhou architecture	Y	recommend system	N	high
Our system	Huizhou architecture constructive grammar	Chinese Huizhou traditional vernacular houses	Y	initial footprints	Y	high

methods use constructive rules to segment the building components. However, generating models via derivation and variation using these modeling methods is difficult. A traditional construction grammar is a suitable method of generating a Chinese historical building model. However, such methods must consider the coordination among multiple houses. Our approach combines the advantages of the aforementioned methods.

VII. CONCLUSION

This paper presented an end-to-end approach for the automatic generation of visually plausible Huizhou vernacular house layouts and features using simple layout data input by the user. Our procedural modeling system is based on the improved constructive grammar [23]. We have introduced a GPU-accelerated MCMC optimization framework to generate the layouts and features for multiple houses, while the 2D geometry layout rules and conflict penalty functions are used to evaluate the results of layout and features. When dealing with a single house, we used the probability network in [54] to generate its features to avoid performing unnecessary calculations. The proposed dynamic probability strategy layout model is able to find better layout results with shorter time and less iterations comparing to the traditional Metropolis-Hastings layout model [41], [42]. Compared to other RBM system, our system has the advantages in user interface, multiple object coordination and derivation.

In summary, the proposed method is versatile and expansible. The pipeline of the proposed method can be easily extended to other Chinese historical architecture.

However, the proposed method has several limitations. The first limitation is that the layout result depends on the initial layout input. As a result, if several unrealistic inputs are included, such as a house inside another house, the system might be unable to place the inside house outside of the outside house. The second limitation is the lack of validation of the model with a physical layout, since the layout rules, feature conflict functions and the parameters used in the stochastic optimization are designed empirically. The last limitation occurs in the feature generation step. Although feature conflicts between adjacent vernacular houses can be resolved, other complex conflicts cannot be resolved using neighbor information.

In our future work, we will expand our model to include additional types of Huizhou architecture, such as Huizhou government offices, memorial archways and ancestral halls, with the goal of generating a more realistic model of a Huizhou village.

REFERENCES

[1] M. Murphy, E. McGovern, and S. Pavia, "Historic Building Information Modelling (HBIM)," *Struct. Survey*, vol. 27, no. 4, pp. 311–327, 2009.

[2] M. Murphy, E. McGovern, and S. Pavia, "Historic Building Information Modelling—Adding intelligence to laser and image based surveys of european classical architecture," *ISPRS J. Photogram. Remote Sens.*, vol. 76, pp. 89–102, Feb. 2013.

[3] R. M. Smelik, T. Tutenel, R. Bidarra, and B. Benes, "A survey on procedural modelling for virtual worlds," *Comput. Graph. Forum*, vol. 33, no. 6, pp. 31–50, 2014.

[4] P. Müller, P. Wonka, S. Haegler, A. Ulmer, and L. Van Gool, "Procedural modeling of buildings," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 614–623, 2006.

[5] M. Lipp, P. Wonka, and M. Wimmer, "Interactive visual editing of grammars for procedural architecture," *ACM Trans. Graph.*, vol. 27, no. 3, 2008, Art. no. 102.

[6] B. Watson, P. Müller, O. Veryovka, A. Fuller, P. Wonka, and C. Sexton, "Procedural urban modeling in practice," *IEEE Comput. Graph. Appl.*, vol. 28, no. 3, pp. 18–26, May/Jun. 2008.

[7] M. Schwarz and P. Müller, "Advanced procedural modeling of architecture," *ACM Trans. Graph.*, vol. 34, no. 4, 2015, Art. no. 107.

[8] Y. I. Parish and P. Müller, "Procedural modeling of cities," in *Proc. 28th Annu. Conf. Comput. Graph. Interact. Techn.*, 2001, pp. 301–308.

[9] P. E. Debevec, C. J. Taylor, and J. Malik, "Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach," in *Proc. 23rd Annu. Conf. Comput. Graph. Interact. Techn.*, 1996, pp. 11–20.

[10] B. M. Oh, M. Chen, J. Dorsey, and F. Durand, "Image-based modeling and photo editing," in *Proc. 28th Annu. Conf. Comput. Graph. Interact. Techn.*, 2001, pp. 433–442.

[11] A. R. Dick, P. H. Torr, and R. Cipolla, "Image-based modeling and photo editing," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 111–134, 2004.

[12] N. Jiang, P. Tan, and L.-F. Cheong, "Symmetric architecture modeling with a single image," *ACM Trans. Graph.*, vol. 28, no. 5, 2009, Art. no. 113.

[13] S. Agarwal *et al.*, "Building rome in a day," *Commun. ACM*, vol. 54, no. 10, pp. 105–112, Oct. 2011.

[14] E. Kwak, "Automatic 3D building model generation by integrating LiDAR and aerial images using a hybrid approach," Ph.D. dissertation, Dept. Geomatics Eng., Univ. Calgary, Calgary, Alberta, 2013.

[15] F. Rottensteiner, "Automatic generation of high-quality building models from LiDAR data," *IEEE Comput. Graph. Appl.*, vol. 23, no. 6, pp. 42–50, Nov. 2003.

[16] H. Lin *et al.*, "Semantic decomposition and reconstruction of residential scenes from LiDAR data," *ACM Trans. Graph.*, vol. 32, no. 4, 2013, Art. no. 66.

[17] H. Arefi, "From LiDAR point clouds to 3D building models," Ph.D. dissertation, Dept. Inst. Appl. Comput. Sci., Univ. Federal Armed Forces Munich, Munich, Bavaria, 2009.

[18] M. Shinya and M. C. Forgue, "Laying out objects with geometric and physical constraints," *Vis. Comput.*, vol. 11, no. 4, pp. 188–201, 1995.

[19] P. Wonka, M. Wimmer, F. Sillion, and W. Ribarsky, "Instant architecture," *ACM Trans. Graph.*, vol. 22, no. 4, pp. 669–677, 2003.

[20] J. P. Duarte, "A discursive grammar for customizing mass housing: The case of Siza's houses at Malagueira," *Autom. Construction*, vol. 14, no. 2, pp. 265–275, 2005.

[21] J. Edelsbrunner, S. Havemann, A. Sourin, and D. W. Fellner, "Procedural modeling of architecture with round geometry," *Comput. Graph.*, vol. 64, pp. 14–25, May 2017.

[22] H. Hua, "A bi-directional procedural model for architectural design," *Comput. Graph. Forum*, to be published. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1111/cgf.13074/full, doi: 10.1111/cgf.13074.

[23] H. Liu, Q. Wang, W. Hua, D. Zhou, and H. Bao, "Building Chinese ancient architectures in seconds," in *Proc. Int. Conf. Comput. Sci.*, 2005, pp. 248–255.

[24] Y. Liu, C. Xu, Z. Pan, and Y. Pan, "Semantic modeling project: Building vernacular house of southeast China," in *Proc. ACM SIGGRAPH Int. Conf. Virtual Reality Continuum Appl. Industry (VRCAI)*, 2004, pp. 412–418.

[25] Y. Liu, C. Xu, Q. Zhang, and Y. Pan, "Ontology based semantic modeling for Chinese ancient architectures," in *Proc. Int. Conf. Artif. Intell.*, 2006, vol. 21. no. 2, p. 1808.

[26] Y. Liu, C. Xu, Q. Zhang, and Y. Pan, "The smart architect: Scalable ontology-based modeling of ancient Chinese architectures," *IEEE Intell. Syst.*, vol. 23, no. 1, pp. 49–56, Jan./Feb. 2008.

[27] Y. Liu, C. Xu, Z. Pan, and Y. Pan, "Semantic modeling for ancient architecture of digital heritage," *Comput. Graph.*, vol. 30, no. 5, pp. 800–814, 2006.

[28] C.-Y. Huang and W.-K. Tai, "Ting tools: Interactive and procedural modeling of Chinese ting," *Vis. Comput.*, vol. 29, no. 12, pp. 1303–1318, 2013.

[29] X. Chen, S. B. Kang, Y.-Q. Xu, J. Dorsey, and H.-Y. Shum, "Sketching reality: Realistic interpretation of architectural designs," *ACM Trans. Graph.*, vol. 27, no. 2, 2008, Art. no. 11.

[30] F. Hou, Q. Yue, and Q. Hong, "Drawing-based procedural modeling of Chinese architectures," *IEEE Trans. Vis. Comput. Graphics*, vol. 18, no. 1, pp. 30–42, Jan. 2012.

[31] F. Hou, H. Qin, and Y. Qi, "Procedure-based component and architecture modeling from a single image," *Vis. Comput.*, vol. 32, no. 2, pp. 151–166, 2016.

[32] P. Merrell, E. Schkufza, and V. Koltun, "Computer-generated residential building layouts," *ACM Trans. Graph.*, vol. 29, no. 6, 2010, Art. no. 181.

[33] L. Fan and P. Wonka, "A probabilistic model for exteriors of residential buildings," *ACM Trans. Graph.*, vol. 35, no. 5, pp. 1–13, 2016.

[34] G. Nishida, I. Garcia-Dorado, D. G. Aliaga, B. Benes, and A. Bousseau, "Interactive sketching of urban procedural models," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 1–11, 2016.

[35] M. Larive, O. L. Roux, and V. Gaildrat, "Using meta-heuristics for constraint-based 3D objects layout," in *Proc. 7th Int. Conf. Comput. Graph. Artif. Intell.*, 2004, pp. 12–13.

[36] J. Michalek, R. Choudhary, and P. Papalambros, "Architectural layout design optimization," *Eng. Optim.*, vol. 34, no. 5, pp. 461–484, 2002.

[37] R. W. Flack and B. J. Ross, "Evolution of architectural floor plans," in *Proc. Eur. Conf. Appl. Evol. Comput.*, 2011, pp. 313–322.

[38] D. Camozzato, L. Dihl, I. Silveira, F. Marson, and S. R. Musse, "Procedural floor plan generation from building sketches," *Vis. Comput.*, vol. 31, nos. 6–8, pp. 753–763, 2015.

[39] P. Guerrero, S. Jeschke, M. Wimmer, and P. Wonka, "Learning shape placements by example," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 1–13, 2015.

[40] C.-H. Peng, Y.-L. Yang, and P. Wonka, "Computing layouts with deformable templates," *ACM Trans. Graph.*, vol. 33, no. 4, 2014, Art. no. 99.

[41] L.-F. Yu, S.-K. Yeung, C.-K. Tang, D. Terzopoulos, T. F. Chan, and S. J. Osher, "Make it home: Automatic optimization of furniture arrangement," *ACM Trans. Graph.*, vol. 30, no. 4, 2011, Art. no. 86.

[42] P. Merrell, E. Schkufza, Z. Li, M. Agrawala, and V. Koltun, "Interactive furniture layout using interior design guidelines," *ACM Trans. Graph.*, vol. 30, no. 4, 2011, Art. no. 87.

[43] M.-T. Chi, S.-S. Lin, S.-Y. Chen, C.-H. Lin, and T.-Y. Lee, "Morphable word clouds for time-varying text data visualization," *IEEE Trans. Vis. Comput. Graphics*, vol. 21, no. 12, pp. 1415–1426, Dec. 2015.

[44] K. Koh, B. Lee, B. Kim, and J. Seo, "ManiWordle: Providing flexible control over Wordle," *IEEE Trans. Vis. Comput. Graphics*, vol. 16, no. 6, pp. 1190–1197, Nov./Dec. 2010.

[45] B. Lee, N. H. Riche, A. K. Karlson, and S. Carpendale, "SparkClouds: Visualizing trends in tag clouds," *IEEE Trans. Vis. Comput. Graphics*, vol. 16, no. 6, pp. 1182–1189, Nov./Dec. 2010.

[46] H. Strobelt, M. Spicker, A. Stoffel, D. Keim, and O. Deussen, "Rolled-out wordles: A heuristic method for overlap removal of 2D data representatives," *Comput. Graph. Forum*, vol. 31, no. 3pt3, pp. 1135–1144, 2012. [Online]. Available: http://onlinelibrary.wiley.com/wol1/doi/10.1111/j.1467-8659.2012.03106.x/full

[47] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, Apr. 1970.

[48] S. Chib and E. Greenberg, "Understanding the metropolis–Hastings algorithm," *Amer. Stat.*, vol. 49, no. 4, pp. 327–335, 1995.

[49] L. Bai and J. Wang, *The Structure of the Official Building in Ching Dynasty*, (In Chinese), Beijing, China: Beijing Univ. Technol., 2000.

[50] S. Liang, *Ying Zao Fa Shi Zhu Shi[The Annotated Yingzao Fashi]*, (In Chinese). Beijing, China: Zhongguo Jianzhu Gongye, 1983.

[51] G. Pan and J. He, *Construction Interpretation*, (In Chinese). Nanjing, China: Southeast Univ., 2005.

[52] Y. Zhu, *Huizhou Architecture*, (In Chinese). Hefei, China: Anhui people's Publishing, 2005.

[53] R. Liu and N. Jin, *HuiZhou Traditional Architecture Features Illustrated*. Beijing, China: Construction Industry of China, 2015.

[54] S.-L. Li, L. Li, M.-W. Cao, and X.-P. Liu, "Construction-oriented rapid modeling method on Huizhou architectures," (In Chinese) *J. Softw.*, vol. 27, no. 10, pp. 2542–2556, 2016.

[55] S. Chaudhuri, E. Kalogerakis, L. Guibas, and V. Koltun, "Probabilistic reasoning for assembly-based 3D modeling," *ACM Trans. Graph.*, vol. 30, no. 4, 2011, Art. no. 35.

[56] J. Beneš, T. Kelly, F. Děchtěrenko, J. Křivánek, and P. Müller, "On realism of architectural procedural models," *Comput. Graph. Forum*, vol. 36, no. 2, pp. 225–234, 2017.

SHANG-LIN LI received the B.S. degree in software engineering from Chinese Geoscience University, Wuhan, China, in 2009, and the Master's degree in computer application technology from the Hefei University of Technology, Hefei, China, in 2014, where he is currently pursuing the Ph.D. degree with the School of Computer and Information. His research interests include computer graphics and modeling.

LIN LI received the Master's degree in computer application technology and the Ph.D. degree in computer application technology from the Hefei University of Technology, Hefei, China, in 2014 and 2016, respectively. She is currently an Assistant Professor with the School of Computer and Information, Hefei University of Technology. Her research interests include computer graphics and computer animation.

MING-WEI CAO received the Master's degree in computer software and theory from the Hefei University of Technology, Hefei, China, in 2014, where he is currently pursuing the Ph.D. degree with the School of Computer and Information. His research interests include 3-D reconstruction, computer vision, and machine learning.

LI CAO received the Master's degree in computer application technology from the Hefei University of Technology, Hefei, China, in 2007, and the Ph.D. degree from the University of Hong Kong in 2012. He is currently an Assistant Professor with the School of Computer and Information, Hefei University of Technology. His research interests include mesh generation and geometric processing.

WEI JIA (M'12) received the B.Sc. degree in informatics from Central China Normal University, Wuhan, China, in 1998, the M.Sc. degree in computer science from the Hefei University of Technology, Hefei, China, in 2004, and the Ph.D. degree in pattern recognition and intelligence system from the University of Science and Technology of China, Hefei, in 2008. He was a Research Assistant and an Associate Professor with the Hefei Institutes of Physical Science, Chinese Academy of Science, from 2008 to 2016. He is currently an Research Associate Professor with the School of Computer and Information, Hefei University of Technology. His research interests include computer vision, biometrics, pattern recognition, image processing, and machine learning.

XIAO-PING LIU received the Master's and Ph.D. degrees in computer science from the Hefei University of Technology, Hefei, China, respectively. He is currently a Professor with the School of Computer and Information, Hefei University of Technology. His research interests include 3-D reconstruction, computer animation, and cooperative computing.