Copyright © 2019 Tech Science Press CMC, vol.58, no.2, pp.307-318, 2019

Improve Computer Visualization of Architecture Based on the Bayesian Network

Tao Shen[1, *], Yukari Nagai[1] and Chan Gao[2]

Abstract: Computer visualization has marvelous effects when it is applied in various fields, especially in architectural design. As an emerging force in the innovation industry, architects and design agencies have already demonstrated the value of architectural visual products in actual application projects. Based on the digital image technology, virtual presentation of future scenes simulates architecture design, architectural renderings and multimedia videos. Therefore, it can help design agencies transform the theoretical design concept into a lively and realistic visual which can provide the audience with a clearer understanding of the engineering and construction projects. However, it is challenging for designers to produce satisfactory renderings due to the frequent fault data during rendering. In this paper, we use the 3Ds MAX as the operating platform and we present an algorithm based on the Bayesian network to construct a vector representation of the fault data. On this basis, a case study of 3D Max' application has also been presented.

Keywords: 3Ds MAX, architectural visualization, application, geometric reconstruction, Bayes.

1 Introduction

In the field of traditional architectural design, designers only use a few sketches to convey to the audience, which means the audience basically cannot witness the design process, except some communication with a few professional terms, such as cuboids, equilateral triangles, trapezoids and so on. With the birth of new media, audiences are not satisfied with the form of expressing the deducing results through traditional data and texts [Stephygraph and Arunkumar (2016)]. Therefore, the characters, symbols and mathematical equation need to be represented in visual images so that human thinking modes can quickly understand the space establishment [Kurup, Sullivan, Hannagan et al. (2017)]. With the increasingly powerful computing speed and massive storage space of computers, people can operate multi-dimensional simulation technology more quickly and easily, since the design process of computer data can be established in short order, making many abstract calculation processes visual forms through the visual image. The

[1] Knowledge Science, Japan Advanced Institute of Science and Technology, Nomi City, Ishikawa 923-1292, Japan.

[2] Architecture Department, Huzhou University, Huzhou, Zhejiang, 313000, China.

[*] Corresponding Author: Tao Shen. Email: shentao@jaist.ac.jp.

results can also be predicted in advance that it will be more easily for people to modify their design and also for audiences to understand designers' ideas.

Computers have become a new media tool between architecture and the natural environment. Currently, the computer simulation technology of the building environment has been applied to many architectural designs. The development of computer technology has realized the visual design. Based on the 3Ds MAX operating platform and the examples concerning the production process of indoor and outdoor renderings and roaming animations, we investigated the application of 3Ds MAX in architecture visual design.

In 1990, Autodesk Software Development Corporation launched the first animation design software 3D Studio, also known as 3Ds Max or MAX, which is a 3D animation rendering and production software developed by the company based on the personal computer system. It consists of modeling, animation, material setting and rendering scheme with excellent function and applicability of human-computer interaction [Dalia, Yingyue, Daniel R et al. (2015)], and many customers around the world now use it.

3Ds MAX has been used as the 3D software since the appearance of architectural visual design. As the leading software in 3D production, it has significant functions, and it is mainly used for architecture renderings and landscape tours production in the architecture visualization process. However, it is difficult for designers to produce satisfactory renderings due to the frequent fault data [Ghebrebrhan, Aranda, Walsh et al. (2017)]. In this investigation, we improve this situation by presenting an algorithm based on the Bayesian network to construct a vector representation of the fault data.

2 Bayesian architecture visualization model based on geometric reconstruction

The algorithm is summarized as the followings: (1) through the geometric reconstruction process of the Bayesian network [Song, Li, Gu et al. (2016)], the redistribution and reimplementation [Arunkumar, Kumar and Venkataraman (2016)] of the information contained in the network model can be realized, as well as the centralization [Arunkumar, Kumar and Venkataraman (2016)] of distributed information. Thus this algorithm enables a few model nodes to contain most of the fault data information; (2) the fault signal can generally be described with the multivariate form of the Gaussian function to construct a vector representation of the fault data with sparse features. It can be assumed that in the subnet area, there is a right-angle projection problem, which is $\boldsymbol{s}_{l_e} = \boldsymbol{U}_{l_e}^T \boldsymbol{v}_{l_e}$, then the vector representation can be expressed as follows:

$$\boldsymbol{s}_{i_{l_e}} = \left[\boldsymbol{s}_{i_{l_k}}, \boldsymbol{0}_{n_{l_e}-k} \right], \boldsymbol{s}_{i_{l_k}} = \left[s_{i_{l_1}}, s_{i_{l_2}}, \cdots, s_{i_{l_k}} \right] \tag{1}$$

In the vector representation, the number of zero elements is $n_{l_e} - k$. Meanwhile, the number of non-zero elements is *k*, $k < n_{l_e}$ and the parameter $\boldsymbol{s}_{i_{l_k}}$ is a non-zero block in the fault data, whose size is *k*. Therefore, a Gaussian distribution may be used to construct a multivariate representation model specifically:

$$\begin{cases} p(\boldsymbol{s}_{i_{l_k}}) \sim \mathcal{N}(0, \boldsymbol{C}_{i_0}) \\ \boldsymbol{C}_{i_0} = \gamma_i \Sigma_i \\ i = \{x, y, z\} \end{cases} \tag{2}$$

In Eq. (2), γ_i is the scalar function in the visual model of the architecture function and $\Sigma_i \in \mathcal{R}^{k\times k}$ is the positive definite parameter matrix of the visual model of the building function. Using the settled Bayesian rule, we set the noise interference in the fault data as $\boldsymbol{w}_{q_i}$; the vector dimension is $M\times 1$, which satisfies the data feature of independent and identical distribution in the random process and the form of $\boldsymbol{w}_{q_i} \sim \mathcal{N}\left(0,\sigma_{w_i}\boldsymbol{I}_M\right)$. Then we can get the following theorem form:

Theorem 1: The posterior density vectors that are presented in visual data parameters $\boldsymbol{s}_{i_{l_k}}$ of architecture function also satisfies Gaussian distribution features. Its Gaussian matrix is $p\left(\boldsymbol{s}_{i_{l_k}} \mid \boldsymbol{y}_{qi};\sigma_{w_i},\gamma_i\Sigma_i\right)\sim\mathcal{N}\left(\mu_{s_{i_l}},\boldsymbol{C}_i\right)$, whose mean and variance of make up a matrix in the following forms:

$$\begin{cases}\mu_{s_{i_l}}=\boldsymbol{C}_{i_0}\boldsymbol{A}_{u_{l_k}}^T(\boldsymbol{A}_{u_{l_k}}\boldsymbol{C}_{i_0}\boldsymbol{A}_{u_{l_k}}^T+\sigma_{w_i}\boldsymbol{I}_M)^{-1}y_{q_i}\\ \boldsymbol{C}_i=\boldsymbol{C}_{i_0}-\boldsymbol{C}_{i_0}\boldsymbol{A}_{u_{l_k}}^T(\boldsymbol{A}_{u_{l_k}}\boldsymbol{C}_{i_0}\boldsymbol{A}_{u_{l_k}}^T+\sigma_{w_i}\boldsymbol{I}_M)^{-1}\boldsymbol{A}_{u_{l_k}}\boldsymbol{C}_{i_0}\end{cases} \tag{3}$$

Among them, the dimension of the matrix $\boldsymbol{A}_{u_{l_k}}=\boldsymbol{A}_{u_{l(:,1:k)}}$ is $M\times k$, which means intercepting the first *k* columns of data in the matrix.

Prove: Assume that the calculation equation of the probability density in the observation matrix $\boldsymbol{A}_{u_{l_k}}$ and $\boldsymbol{s}_{i_{l_k}}$ is:

$$\begin{aligned}&p(\boldsymbol{y}_{q_i}\left|\boldsymbol{A}_{u_{l_k}},\boldsymbol{s}_{i_{l_k}}\right.)=\prod_{n=1}^{M}p(y_{q_{i_n}}\mid(\boldsymbol{A}_{u_{l_k}})_{(:,n)},\boldsymbol{s}_{i_{l_k}})\\ &=\frac{\exp(-\frac{1}{2\sigma_{w_i}}\left\|\boldsymbol{y}_{q_i}-\boldsymbol{A}_{u_{l_k}}\boldsymbol{s}_{i_{l_k}}\right\|_2^2)}{(2\pi\sigma_{w_i})^{m/2}}\sim\mathcal{N}(\boldsymbol{A}_{u_{l_k}}\boldsymbol{s}_{i_{l_k}},\sigma_{w_i}\boldsymbol{I}_M)\end{aligned} \tag{4}$$

Based on the Bayesian rules [Arunkumar, Jayalalitha, Dinesh et al. (2012)], the posterior distribution calculation model of the visualization model of architecture function can be obtained:

$$p(\boldsymbol{s}_{i_{l_k}}\mid\boldsymbol{y}_{q_i},\boldsymbol{A}_{u_{l_k}})=\frac{p(\boldsymbol{y}_{q_i}\mid\boldsymbol{A}_{u_{l_k}},\boldsymbol{s}_{i_{l_k}})p(\boldsymbol{s}_{i_{l_k}})}{p(\boldsymbol{y}_{q_i}\mid\boldsymbol{A}_{u_{l_k}})} \tag{5}$$

The Fourier marginal likelihood constant used in the equation is mainly used to normalize the posterior distribution of the architectural function visualization model. The calculation form can be expressed as follows:

$$p(\boldsymbol{y}_{q_i}\mid\boldsymbol{A}_{u_{l_k}})=\int p(\boldsymbol{y}_{q_i}\mid\boldsymbol{A}_{u_{l_k}},\boldsymbol{s}_{i_{l_k}})p(\boldsymbol{s}_{i_{l_k}})d\boldsymbol{s}_{i_{l_k}} \tag{6}$$

Substituting Eq. (6) into the posterior probability parameter in Eq. (5), the improved equations are:

$$\begin{aligned}&p(\boldsymbol{s}_{i_{l_k}}\mid\boldsymbol{y}_{q_i},\boldsymbol{A}_{u_{l_k}})\propto\exp\\ &\times(-\frac{(\boldsymbol{s}_{i_{l_k}}-\mu_{s_{i_l}})^T(\sigma_{wi}^{-1}\boldsymbol{A}_{u_{l_k}}^T\boldsymbol{A}_{u_{l_k}}+\boldsymbol{C}_{i_0}^{-1})(\boldsymbol{s}_{i_{l_k}}-\mu_{s_{i_l}})}{2})\end{aligned} \tag{7}$$

$$\mu_{s_{il}} = \sigma_{w_i}^{-1}(\sigma_{w_i}^{-1}\boldsymbol{A}_{u_{lk}}^{T}\boldsymbol{A}_{u_{lk}} + \boldsymbol{C}_{i_0}^{-1})^{-1}\boldsymbol{A}_{u_{lk}}\boldsymbol{y}_{q_i} \tag{8}$$

Thus, the calculation equation of the posterior distribution parameter of the architecture function visual model in Gaussian function is:

$$p(\boldsymbol{s}_{i_{lk}} \mid \boldsymbol{y}_{q_i}, \boldsymbol{A}_{u_{lk}}) \sim N\left(\frac{1}{\sigma_{w_i}}\boldsymbol{C}_i\boldsymbol{A}_{u_{lk}}\boldsymbol{y}_{q_i}, \boldsymbol{C}_i\right) \tag{9}$$

In the equation,

$$\begin{aligned} \boldsymbol{C}_i &= (\sigma_{w_i}^{-1}\boldsymbol{A}_{u_{lk}}^{T}\boldsymbol{A}_{u_{lk}} + \boldsymbol{C}_{i_0}^{-1})^{-1} \\ &= \boldsymbol{C}_{i_0} - \boldsymbol{C}_{i_0}\boldsymbol{A}_{u_{lk}}^{T}(\sigma_{wi}\boldsymbol{I}_M + \boldsymbol{A}_{u_{lk}}\boldsymbol{C}_{i_0}\boldsymbol{A}_{u_{lk}}^{T})^{-1}\boldsymbol{A}_{u_{lk}}\boldsymbol{C}_{i_0} \end{aligned} \tag{10}$$

For the parameters σ_{w_i} and $\boldsymbol{C}_{i_0}$ used in the architecture function visual model, the maximum value $p(\boldsymbol{s}_{i_{lk}} \mid \boldsymbol{y}_{q_i}, \boldsymbol{A}_{u_{lk}})$ estimated with posterior probability can be calculated as:

$$\begin{aligned} \hat{\mu}_{s_{il}} &= \left(\sigma_{w_i}\boldsymbol{C}_{i_0}^{-1} + \boldsymbol{A}_{u_{lk}}^{T}\boldsymbol{A}_{u_{lk}}\right)^{-1}\boldsymbol{A}_{u_{lk}}^{T}\boldsymbol{y}_{q_i} \\ &= \boldsymbol{C}_{i_0}\boldsymbol{A}_{u_{lk}}^{T}(\boldsymbol{A}_{u_{lk}}\boldsymbol{C}_{i_0}\boldsymbol{A}_{u_{lk}}^{T} + \sigma_{w_i}\boldsymbol{I}_M)^{-1}y_{q_i} \end{aligned} \tag{11}$$

Thus, for the parameters $\sigma_{w_i}, \gamma_i, \Sigma_i$ in the architecture function visual model, the fault model in the subnet expanded in the form of:

$$\boldsymbol{v}_{l_e} = \boldsymbol{U}_{l_e}\begin{bmatrix} \mu_{s_{xl}} & \mu_{s_{yl}} & \mu_{s_{zl}} \\ \boldsymbol{0}_{n_{l_e}-k} & \boldsymbol{0}_{n_{l_e}-k} & \boldsymbol{0}_{n_{l_e}-k} \end{bmatrix} \tag{12}$$

For the parameters $\sigma_{w_i}, \gamma_i, \Sigma_i$ in the architecture function visual model, the method of optimizing the expect extremism is used to obtain the calculation result $p\left(\boldsymbol{y}_{q_i}; \left(\boldsymbol{s}_{i_{lk}} \mid \boldsymbol{y}_{qi}; \sigma_{w_i}, \gamma_i, \Sigma_i\right)\right)$. This calculation shows that the i can be converted into x, y or z in model calculation. This method of processing can achieve the largest reduction of $-\log\left(p\left(\boldsymbol{y}_{q_i};\theta\right)\right)$, and the likelihood boundary of the architecture function visual data is calculated as follows:

$$\mathcal{L}(\theta; \boldsymbol{y}_{q_i}) = (\boldsymbol{y}_{q_i})^T \sum\nolimits_{y_i}^{-1} \boldsymbol{y}_{q_i} + 1og\left|\Sigma_{y_i}\right| \tag{13}$$

In the equation, $\Sigma_{y_i} = \sigma_w\boldsymbol{I}_M + \boldsymbol{A}_{u_{lk}}\boldsymbol{C}_{i_0}\boldsymbol{A}_{u_{lk}}^{T}$. The following theorem schemes can characterize the iterative calculation of the parameters in the architecture function visual model:

Theorem 2: The maximum likelihood estimation is used for conducting the learning process of the parameter $\sigma_{w_i}, \gamma_i, \Sigma_i$ in the architecture function visual model according to the following rules:

$$\sigma_{w_i} = \frac{\left\|\boldsymbol{y}_{q_i} - \boldsymbol{A}_{u_{lk}}\boldsymbol{s}_{i_{lk}}\right\|_2^2 + \sigma_{w_i}\left[k - Tr(\boldsymbol{C}_i\boldsymbol{C}_{i_0}^{-1})\right]}{M} \tag{14}$$

$$\gamma_i = \frac{Tr(\sum\nolimits_i^{-1}(\boldsymbol{C}_i + \mu_{s_{il}}(\mu_{s_{il}})^T))}{k} \tag{15}$$

$$\gamma_i = \frac{Tr(\sum_i^{-1}(\boldsymbol{C}_i + \mu_{s_{i_l}}(\mu_{s_{i_l}})^T))}{k} \tag{16}$$

Prove: Maximum Likelihood Estimation algorithm mainly includes two main processes during the execution: E-estimation and M-estimation.

(1) E-estimation: use the expectation function to calculate the mean of the likelihood function of the architecture function visual model:

$$\begin{aligned} Q(\theta \mid \theta^{(t)}) &= E_{\boldsymbol{s}_{i_{l_k}} \left| y_{qi} ; \theta^{(t)} \right.} \left[1ogp(\boldsymbol{y}_{q_i}, \boldsymbol{s}_{i_{l_k}} ; \theta^{(t)}) \right] \\ &\quad + E_{\boldsymbol{s}_{i_{l_k}} \left| y_{qi} ; \theta^{(t)} \right.} [1ogp(\boldsymbol{s}_{i_{l_k}} ; \gamma_i, \Sigma_i)] \end{aligned} \tag{17}$$

$$\theta^{(t+1)} := \arg\max_{\theta} Q(\theta \mid \theta^{(t)}) \tag{18}$$

(2) M-estimation: use the optimization calculation process to select $\theta = (\sigma_{w_i}, \gamma_i, \Sigma_i)$ to maximize the value of the following model:

$$\theta^{(t+1)} := \arg\max_{\theta} Q(\theta \mid \theta^{(t)}) \tag{19}$$

Through the parameter simplification process, a simplified model of the parameter σ_{w_i} in the model of Eq. (18) can be obtained in the form:

$$Q(\sigma_{w_i} \mid \theta^{(t)}) \propto E_{\boldsymbol{s}_{i_{l_k}} \left| y_{qi} ; \theta^{(t)} \right.} \left[1ogp(\boldsymbol{y}_{q_i}, \boldsymbol{s}_{i_{l_k}} ; \sigma_{w_i}) \right] \tag{20}$$

We conduct the derivation process on the parameter item $Q(\sigma_{w_i} \mid \theta^{(t)})$ and set its value to 0 to get the following model form:

$$\sigma_{w_i} = \frac{\left\| \boldsymbol{y}_{q_i} - \boldsymbol{A}_{u_{l_k}} \boldsymbol{s}_{i_{l_k}} \right\|_2^2 + \sigma_{w_i} \left[k - Tr(\boldsymbol{C}_i \boldsymbol{C}_{i_0}^{-1}) \right]}{M} \tag{21}$$

In the diagnostic model, we also take into account the value of the parameter γ_i, Σ_i, Eq. (20) can be simplified as follow:

$$Q(\gamma_i, \Sigma_i \mid \theta^{(t)}) \propto E_{\boldsymbol{s}_{i_{l_k}} \left| y_{qi} ; \theta^{(t)} \right.} [1ogp(\boldsymbol{s}_{i_{l_k}} ; \gamma_i, \Sigma_i)] \tag{22}$$

Simultaneously, we establish the following relation:

$$1ogp(\boldsymbol{s}_{i_{l_k}} ; \gamma_i, \Sigma_i) \propto -\frac{k}{2} 1og \left| \gamma_i, \Sigma_i \right| - \frac{\boldsymbol{s}_{i_{l_k}}^T \left(\gamma_i \Sigma_i \right)^{-1} \boldsymbol{s}_{i_{l_k}}}{2} \tag{23}$$

Therefore, the equivalent form of Eq. (21) is:

$$\begin{aligned} Q(\gamma_i, \Sigma_i \mid \theta^{(t)}) &\propto -\frac{k}{2} 1og \left| \gamma_i \right| - \frac{k}{2} 1og \left| \Sigma_i \right| \\ &\quad - \frac{Tr\left(\left(\gamma_i \Sigma_i \right)^{-1} \left(\boldsymbol{C}_i + \mu_{s_{i_l}} \mu_{s_{i_l}}^T \right) \right)}{2} \end{aligned} \tag{24}$$

In the equation, the parameter item $\mu_{s_{i_l}}$ and $\boldsymbol{C}_i$ can obtain the estimation result based on the calculation process shown in Eq. (11). Through the partial derivative process, the training calculation form for the parameter item γ_i can be obtained as:

$$\gamma_i = \frac{Tr\left(\Sigma_i^{-1}\left(\boldsymbol{C}_i + \mu_{s_{il}} \mu_{s_{il}}^T\right)\right)}{k} \tag{25}$$

Similarly, it can be calculated that the form $\Sigma_i = \frac{\boldsymbol{C}_i + \mu_{s_{il}} (\mu_{s_{il}})^T}{\gamma_i}$ is satisfied.

In addition, in order to improve the diagnostic performance of the architecture function visual model, the improvement measure taken is to define the valuing form of Σ_i, adopting the model structure with symmetrical form, with the composing elements $(\Sigma_i)(m,l) = r_i^{|m-l|}$ and $\forall m,l \in [1,2,\cdots,k]$ of the matrix and then construct a model of non-zero elements in the matrix, adopting the calculation model of first-order autoregressive model. The calculation form of r_i is:

$$r_i = sign(m_1/m_0)\min\{|m_1/m_0|, 0.99\} \tag{26}$$

In Eq. (25), the parameter term m_0 is the mean of the diagonal elements of the matrix, and the parameter term m_1 is the mean of the diagonal elements of Σ_i. Algorithm 1 shows the details about how to improve the algorithm.

Algorithm 1 Reconstruction of the fault geometry

```
Input: Coding method $\boldsymbol{A}$ 、$\boldsymbol{U}_{l_e}$, the number of non-zero blocks of matrix $k$, fault sample $\boldsymbol{y}_q$;
Output: Parameter estimation of architecture visual model $x, y, z \in \mathcal{R}^{n_{l_e} \times 1}$.
For $l=1,2,\cdots,L$ do
  For $m=1,2,\cdots,K$ do
    Mean $\mu_{s_{xl}}$ 、$\mu_{s_{yl}}$ and $\mu_{s_{zl}}$ calculated:
    $\mu_{s_{il}} = \boldsymbol{C}_{i_0} \boldsymbol{A}_{u_{l_k}}^T (\boldsymbol{A}_{u_{l_k}} \boldsymbol{C}_{i_0} \boldsymbol{A}_{u_{l_k}}^T + \sigma_{w_i} \boldsymbol{I}_M)^{-1} y_{q_i}$
    Variance $\boldsymbol{C}_x$ 、$\boldsymbol{C}_y$ and $\boldsymbol{C}_z$ calculation:
    $\boldsymbol{C}_i = \boldsymbol{C}_{i_0} - \boldsymbol{C}_{i_0} \boldsymbol{A}_{u_{l_k}}^T (\boldsymbol{A}_{u_{l_k}} \boldsymbol{C}_{i_0} \boldsymbol{A}_{u_{l_k}}^T + \sigma_{w_i} \boldsymbol{I}_M)^{-1} \boldsymbol{A}_{u_{l_k}} \boldsymbol{C}_{i_0}$
    Model parameter $\sigma_{w_i}, \gamma_i, \Sigma_i$ updated:
    $\sigma_{w_i} = \frac{\left\| \boldsymbol{y}_{q_i} - \boldsymbol{A}_{u_{l_k}} \boldsymbol{s}_{i_{l_k}} \right\|_2^2 + \sigma_{w_i} \left[ k - Tr(\boldsymbol{C}_i \boldsymbol{C}_{i_0}^{-1}) \right]}{M}$
    $\gamma_i = \frac{Tr(\sum_i^{-1}(\boldsymbol{C}_i + \mu_{s_{il}} (\mu_{s_{il}})^T))}{k}$
    $\Sigma_i = \frac{\boldsymbol{C}_i + \mu_{s_{il}} (\mu_{s_{il}})^T}{\gamma_i}$
    Parameter $\gamma_i$ recalculates:
```

Algorithm 1 Reconstruction of the fault geometry

```
$r_i = sign(m_1/m_0)\min\{|m_1/m_0|, 0.99\}$
    Model parameter matrix $\Sigma_i$ re-estimated:
$\Sigma_i = Toeplitz\{[1, r, \cdots, r^{k-1}]\}$
  Endfor
Sub-grid coordinate conversion of architecture function visual model $l$:
$\boldsymbol{v}_{l_e} = \boldsymbol{U}_{l_e}\begin{bmatrix} \mu_{s_{xl}} & \mu_{s_{yl}} & \mu_{s_{zl}} \\ \mathbf{0}_{n_{l_e}-k} & \mathbf{0}_{n_{l_e}-k} & \mathbf{0}_{n_{l_e}-k} \end{bmatrix}$
For the descending point in the architectural function visualization model, the operation $\cup_{j=1}^{b} N(i_{l_j}) \backslash I_l$ is performed and $i_{l_j}$ is the boundary position of the sub-network 1 of the architectural function visualization model.
Endfor
```

In the model construction of the Bayesian geometric reconstruction for the architecture function visual model selected above, there are two loop processes nested, so the computational complexity of the algorithm can be analyzed based on these two nested loops. The computational complexity of the previous cycle is 1: *L*, and that of the latter cycle is 1: *K*. Therefore, the computational complexity of the outer cycle is $\mathcal{O}(4L \times K)$, the computational complexity of the inner cycle is $\mathcal{O}(L)$, and the overall computational complexity is $\mathcal{O}(4L \times K + L) = \mathcal{O}(n^2)$.

3 Bayesian architectural visual design

From the process of abstraction to figuration, the design concept is more intuitive and persuasive. Visualized products for architectural design can also be considered as a form of media, which has the characteristics of outstanding mass communication. Architectural design visualization products, no matter architectural renderings and real estate or film and television animation, are all targeted at specific audiences, enhancing the effectiveness and accuracy of information through disseminating certain information and using different visual means, and thus achieve the expected results [Yingyue, Qi, William et al. (2016)].

The architectural visual design mainly includes these following contents: indoor 2.2.

3.1 Indoor and outdoor rendering

The indoor rendering is a kind of form that an interior designer describe abstract thinking with the 3Ds MAX production software for creating an interior model and with Photoshop software for post-processing, transforming the abstract thinking language into the visualized language. It gives people a clearer understanding of the design concepts as it truly reproduces the designer’s innovation through the faithful representation of various elements such as color, texture, shape, structure, etc. [Stephygraph, Arunkumar and Venkatraman (2015)].

The outdoor rendering is mostly similar to the indoor renderings, except that the outdoor renderings have more details such as the background elements of the blue sky, white clouds, trees, pedestrians, roads, etc., but these factors all can be post-processed by Adobe Photoshop software. In 3Ds MAX software, designers need to complete the building. However, it should be noted that there are no fixed equations, no matter whether it is an indoor or outdoor rendering, the processing methods need to be adjusted according to the actual situation and need to be changed randomly. Therefore, designers can produce satisfactory renderings.

3.2 Architectural roaming animation

The architectural roaming animation, also known as real estate animation, uses the 3Ds MAX software to create a virtual building environment on the computer, with the location, building appearance, interior decoration, landscape and supporting facilities, then designers import virtual figures and animals into sequence images rendered by 3Ds MAX software. 3Ds MAX software is also used to create animation effects such as wind, rain, thunder and lightning with its particular flow system, presenting every scene dynamically in the building roaming animation [Jennifer, Yingyue and Kathryn (2015)]. The post-processing uses the perspectives, close-ranges, close-ups, panoramas, etc. to express the depth of the animation, so that the audience can browse from any angle at any spot.

3.3 Virtual reality

Virtual Reality was developed by U.S. programmer Lannir in 1989 to use computer technology to generate a realistic 3D visual, auditory, tactile or olfactory sensory world, allowing users to take advantage of natural skills and some devices that make browses and interact investigation to the objective virtual world from their own point of view. It compares virtual reality with human senses [Fernandes, Gurupur, Sunder et al. (2017)].

Virtual reality basically includes three elements. The first element is immersion, also called sensation, which refers to the degree of reality that users feel as a protagonist in the simulated environment. The second element is interaction, which refers that users can communicate interactively with the virtual reality scene, therefore the virtual reality must percept what the user knows, feels, and thinks. The third element is imagination; this element can be sensory or psychological. The sense of presence and participation here is judged based on the user's personal psychology [Jennifer, Yingyue and Kathryn (2015)].

Virtual reality can fulfill people's dreams that cannot be accomplished in the real world. It can not only reproduce the real existing environment but also can create the virtually non-existent.

4 Case study: architectural indoor rendering

Now we report a case study where the algorithm based on the Bayesian network was used to avoid the fault data during rendering with 3Ds Max. The rendering result demonstrates the effectiveness of this algorithm.

4.1 3Ds MAX in producing an indoor rendering

In-line equations/expressions are embedded into the paragraphs of the text. For example, $E = mc^2$. In-line equations or expressions should not be numbered and should use the same/similar font and size as the main text.

4.2 Display style

(1) Making the main contour model

In the modeling of architectural renderings, we first made the main contour model as shown in Fig. 1.

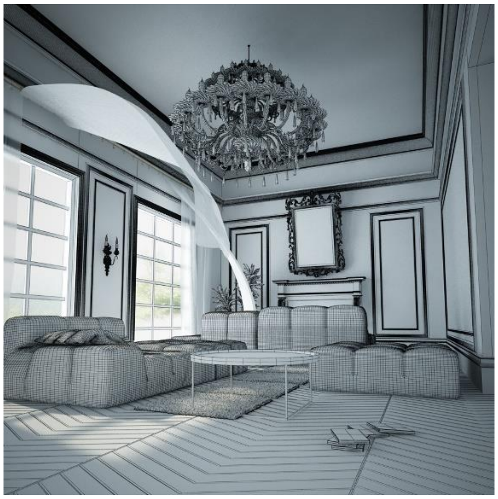


Figure 1: The main contour model

(2) Attaching the texture

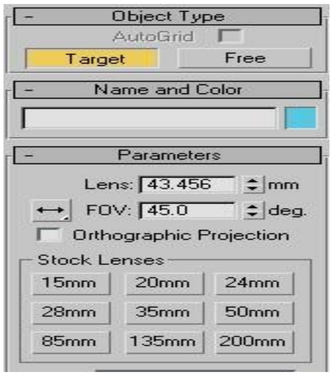
Because the case is an indoor rendering, a V-Ray plug-in needs to be installed to the produced model.

(3) Light setting

The effect of the material of the interior architecture is represented by light. In the 3Ds MAX, the lighting settings should not be random. On the contrary, the lighting plan for the entire indoor scene should be done first.

(4) Creating camera

Before rendering, the camera should be created and adjusted through the height of the camera determination, rendering angle observation, rendering parameters adjusting, and then the image can be outputted and saved, as shown in Fig. 2.



Lens: 43.456 mm

FOV: 45.0 deg.

□ Orthographic Projection

Figure 2: Rendering parameters setting

(5) Post-processing

Post-processing of interior renderings is generally performed in Adobe Photoshop software, as shown in Fig. 3.

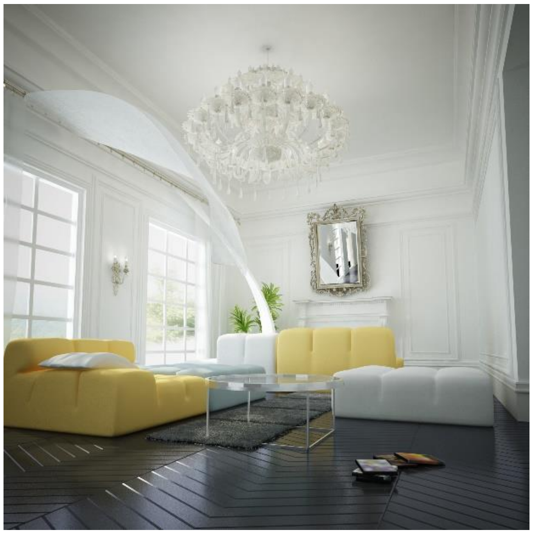


Figure 3: Final computer visualization

The connotations and cultures represented by different styles of indoor effect pictures are various, so when expressing interior design, the choice of style should be based on different cultural connotations. In the production of such an indoor rendering, it is necessary to understand the structure of the house and design according to the floor plan of the house first. Besides, it is also essential to get familiar with the various modeling tools in 3Ds Max. At last, expecting results can be achieved just according to creativity.

5 Conclusions and prospect

3Ds MAX three-dimensional software has been used since the beginning of architectural visual design. As the major software in 3D production, it has greatly important functions.

The fineness of the architectural model and the fluency of the architectural animation all depend on 3Ds MAX. Because of Bayesian 3Ds MAX running through the entire process of building visual design, the operation of this software is complex to some extent. What's more, knowledge of other subjects is also incorporated with it, which is not easy to master completely. In short, because of the difficulties mentioned above and the limited ability of us, this paper still needs to be further improved in many aspects. However, due to the recognition of the practical value of Bayesian 3Ds MAX applied to architectural visual design, we tried our best to meet the challenge and attempted to complete the research. We hope to start a more professional discussion that could provoke more profound mature research in the future.

References

Arunkumar, N.; Jayalalitha, S.; Dinesh, S.; Venugopal, A.; Sekar, D. (2012): Sample entropy based ayurvedic pulse diagnosis for diabetics. *International Conference on Advances in Engineering, Science and Management*, pp. 61-62.

Arunkumar, N.; Kumar, K. R.; Venkataraman, V. (2016): Automatic detection of epileptic seizures using new entropy measures. *Journal of Medical Imaging and Health Informatics*, vol. 6, no. 3, pp. 724-730.

Abdelhamid, D. S.; Zhang, Y.; Lewis, D. R.; Moghe, P. V.; Welsh, W. J. et al. (2015): Tartaric acid-based amphiphilic macromolecules with ether linkages exhibit enhanced repression of oxidized low density lipoprotein uptake. *Biomaterials*, vol. 53, pp. 32-39.

Fernandes, S. L.; Gurupur, V. P.; Sunder, N. R.; Arunkumar, N.; Kadry, S. (2017): A novel nonintrusive decision support approach for heart rate measurement. *Pattern Recognition Letters*.

Ghebrebrhan, M.; Aranda, F.; Walsh, G.; Ziegler, D.; Giardini, S. et al. (2017): Textile frequency selective surface. *IEEE Microwave and Wireless Components Letters*, vol. 27, no. 11, pp. 989-991.

Han, J. W.; Zhang, Y.; Uhrich, K. E. (2015): Amphiphilic macromolecule self-assembled monolayers suppress smooth muscle cell proliferation. *Bioconjugate Chemistry*, vol. 26, no. 7, pp. 1359-1369.

Kurup, P.; Sullivan, C.; Hannagan, R.; Yu, S.; Azimi, H. et al. (2017): Review of technologies for characterization of heavy metal contaminants. *Indian Geotechnical Journal*, vol. 47, no. 4, pp. 421-436.

Pan, W. P.; Chen, Z. Y.; Chen, S. Z.; Feng, Z. Y. (2013): Automatic clustering of social tag using community detection. *Applied Mathematics & Information Sciences*, vol. 7, no. 2, pp, 675-681.

Song, Y.; Li, N.; Gu, J.; Fu, S.; Peng, Z. et al. (2016): β-Hydroxybutyrate induces bovine hepatocyte apoptosis via an ROS-p38 signaling pathway. *Journal of Dairy Science*, vol. 99, no. 11, pp. 9184-9198.

Stephygraph, L. R.; Arunkumar, N.; Venkatraman, V. (2015): Wireless mobile robot control through human machine interface using brain signals. International conference on smart technologies and management for computing, communication. *Controls, Energy and Materials*, pp, 596-603.

Stephygraph, L. R.; Arunkumar, N. (2016): Brain-actuated wireless mobile robot control through an adaptive human-machine interface. *Advances in Intelligent Systems and Computing*, vol. 397, pp. 537-549.

Zhang, Y. Y.; Li, Q.; William, J. W.; Prabhas, V. M.; Moghe, P. V. et al. (2016): Micellar and structural stability of nanoscale amphiphilic polymers: implications for anti-atherosclerotic bioactivity. *Biomaterials*, vol. 84, pp, 230-240.