Creative corbel modeling using evolution principle

Zhang, Yuzhe; Ong, Wayne Chan Chi; Zheng, Jianmin; Lie, Seng-Tjhen

2020

Zhang, Y., Ong, W. C. C., Zheng, J., & Lie, S.-T. (2020). Creative corbel modeling using evolution principle. Proceedings of the 2020 International Conference on Cyberworlds (CW), 9-16. doi:10.1109/CW49994.2020.00010

<https://hdl.handle.net/10356/146240>

<https://doi.org/10.1109/CW49994.2020.00010>

© 2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. The published version is available at: <https://doi.org/10.1109/CW49994.2020.00010>.

Downloaded on 26 Apr 2023 21:10:58 SGT

Creative Corbel Modeling Using Evolution Principle

Yuzhe Zhang*, Wayne Ong Chan Chi*, Jianmin Zheng* and Seng-Tjhen Lie†

*School of Computer Science and Engineering, Nanyang Technological University, Singapore

†School of Civil and Environmental Engineering, Nanyang Technological University, Singapore

Email: zhang.yz@ntu.edu.sg, c170117@e.ntu.edu.sg, asjmzheng@ntu.edu.sg, cstlie@ntu.edu.sg

Abstract—Corbel is a common category of decorative architectural geometry that has clear structure and aesthetic design. This paper presents a method for automatically generating a group of new corbel models from one selected by the user in the dataset. The method consists of offline learning and online generation. The offline learning trains two VAE models (2D CurveVAE and 3D VoxelVAE) for learning the feature representation of corbel parts. The online generation includes a generation algorithm by evolution that evolves to product new generation of models by crossing over and mutating features, and a feature-driven deformation that synthesizes 3D mesh representation of corbel models. By integrating these technical components, we develop a creative corbel modeling tool capable of generating new corbel models that are both “more of the same” and “surprising”, which is demonstrated by experiments.

Keywords—Creative modeling, corbel, evolution principle

I. INTRODUCTION

The field of shape design is fast evolving and expanding its boundaries. Due to rapid development of mathematics, computing, data analytics and machine learning, and availability of a large number of existing 3D models, recent research on geometric modeling is shifting from conventional direct manipulation on points, edges or faces using professional software such as Rhino and 3DSMax towards high level explorative modeling and example-driven synthesis [1].

This paper considers creative modeling for corbel models. A corbel is a typical architectural category and is a structural piece jutting from a wall to carry a superincumbent weight. It could have a plain appearance, or may be elaborately carved with stylized head of humans, animals or imaginary “beasts”. While CAD tools have become ubiquitous to facilitate traditional design and manufacturing, it is time-consuming to design and explore the shapes of corbel models using CAD tools. This thus raises issues of time, cost efficiency, human resource, and, most critically, the creativity essential for designing new models. Note that there are tons of models available in online warehouse, which are well designed in varying shapes and styles. To make use of the existing models, many example-driven synthesis methods start the modeling by offering a set of examples and then generate more instances of the same type guided by some rules extracted or learned from the examples. We aim to develop modeling techniques and a toolkit for creating 3D

corbel models from examples, which provide a fast and cost-effective solution to the design cycle of architectural models.

Our work is inspired by recent progress of generative design that uses artificial techniques to create many design options by simply inputting design specifications [2] and has received great attentions by those forward-looking companies in the world. Generative design provides a way for exploration, inspiration and creativity in modeling. When we say creative corbel modeling, we want the generated models to contain elements of surprise or unexpectedness, which brings inspiration to the designer, and meanwhile to be more of the same as the examples or specifications, particularly retaining architectural structure and geometry style of corbels. These two goals actually conflict with each other, which thus imposes a challenge [3].

We propose a framework to realize creative corbel modeling, which combines geometric modeling with machine learning techniques and example-based modeling. Specifically, we collect a few existing well-designed corbel models that serves as the initial design space, let the user specify one example he/she prefers, and then let evolution based algorithms to generate novel corbel models. To ensure the generated models to meet the two goals mentioned above, we carefully construct high level representations for corbel models, design two Variational Autoencoders (VAE) models for learning the features of the models, and design a generation algorithm involving evolution principle and feature-driven deformation for generating new corbel models. Though we focus on corbel models in this paper, since our algorithm takes triangular meshes as input, it could be easily extended to other models, especially those with similar structures. The major contribution of the paper lies in a new framework for creative corbel modeling and its underlying evolution based generation algorithm.

II. RELATED WORK

This section briefly reviews some relevant work.

A. Example-driven synthesis

Funkhouser et al. first proposed the idea of creating 3D models by assembling parts segmented from shapes in a dataset [1]. Xu et al. extended the idea of part retrieval and (re-)composition into “Fit and diverse” for evolving an entire set of 3D models to obtain generations of fit and diverse new

offsprings [4]. Fit means plausibility that generates chair-like shapes from chairs, while diversity means surprising designs not to be stuck in an elite population. The part exchange is executed via stochastic cross-over. However, machine does not learn any knowledge in this approach. Kalogerakis et al. [5] proposed a probabilistic graphical model, Bayesian network, for part assemblies. The network learns shape “style” and component “style” in latent models, and then new models can be sampled from the learned spaces. Sung et al. proposed “CompletemeMe”, an incremental synthesis to construct a shape one part at a time [6]. A jointly training embedding and retrieval network is constructed. It firstly indexes parts by mapping them to a low-dimensional feature space and secondly maps partial assemblies to appropriate complements. In this way, it suggests each new part to complement the partially constructed shape.

B. Generative modeling

With great success of machine learning, particularly, deep neural networks such as CNNs, VAEs [7] and GAN [2] in computer vision and natural language processing, methods applying deep learning models in geometric modeling have also been proposed for generating 3D shapes. A direct extension of 2D image synthesis is to apply machine learning to 3D voxel grid. In 3D-GAN [8], Wu et al. combined volumetric CNN and GAN to map a 200D latent vector to a 64^3 volume. These voxel based representations usually require huge memory and calculation costs, especially when the volumetric resolution is high. To address this issue, sparse voxel-based methods use octrees to adaptively represent geometry [9].

Various methods have been developed to deal with different types of representation or encoding for 3D shapes. Su et al. [10] projected 3D shapes to multi-view images as input and proposed a novel pooling operation for 3D shape recognition. However, the representation does not contain the full 3D shape information. Qi et al. proposed PointNet [11] and PointNet++ [12] for 3D classification and segmentation, utilizing pooling operations that are order independent. Learning from irregular point clouds is still challenging in order to produce relatively dense and complex geometry. In [13], Chen et al. defined the inside/outside field by taking the sign of its signed distance field for a closed shape. The generative models can be applied to various applications including shape autoencoding, generation, interpolation, completion, and single-view reconstruction. However, the method could not generate structured shapes and strictly applies to closed shapes. As for mesh representations, the shape collection could be encoded as the deformation of a template model [14]. This approach can represent and generate 3D shapes with fine details from the most popular surface meshes as input. However, the use of the template model has the restriction of the same connectivity, which limits the topological and geometric complexity of generated shapes.

With multi-chart representations [15], AtlasNet attempts to overcome the above restriction by generating a shape as a collection of patches, each of which is parameterized to a 2D domain as an atlas.

The fact that man-made shapes are highly structured motivates the structure-aware shape modeling. Some recent works decouple structure and geometry representations so that the structure-aware models process high-level shape abstractions, typically graphs of shape parts and attributes [16], [17], [18], [19]. Li et al. trained independent networks for structure and part geometry, and proposed a generative recursive autoencoder for shape structure based on Recursive Neural Networks (RvNNs) [16]. Wang et al. [17] introduced a global-to-local generative model where GAN is built for structure and conditional AE is augmented to refine in part level. Wu et al. [18] proposed the SAGNet, a structure-aware generative model for 3D shapes, in which the part geometry and structure are jointly learned and fused into a single latent code to intertwine the two types of features for shape modeling. Gao et al. proposed a two-level VAE, in which a PartVAE learns a deformable model of part geometries and a Structured Parts VAE jointly learns part structure and geometry [19].

C. Creativity by evolution

For creative modeling, evolutionary algorithms (EAs), inspired by biological evolution in nature, could mimic producing surprise by the mutation, cross-over, and selection operators. In EAs, we need to encode the contents and design appropriate mutation and cross-over operators to allow the contents to evolve and produce surprises. Controllability is designed by the selection process, where a fitness function determines whether a new creation is allowed to survive to further produce offsprings [3].

Xu et al. [4] proposed the idea of set evolution by combining EA-based stochastic object modeling and a design gallery. It starts with an initial population of 3D objects belonging to the same category, stochastic mutation of object parts and cross-over between objects drive the evolution and produce offspring generations. During the evolution, user preference determines the fitness function for the evolution as selecting shapes from the gallery that are deemed to be fit to breed the next generation. This “fit and diverse” principle makes the set evolution method inspiring and creative.

III. OVERVIEW OF THE PROPOSED METHOD

Our problem can be described as follows: given a set of 3D corbel models represented in common graphics or design format, the user can browse these models and specify one, and then the algorithm automatically generates a group of new corbel models that are “more of the same” (similar) and “surprising” (dissimilar) to the specified one.

Figure 1: Framework of the proposed creative corbel modeling.

To develop such an algorithm, we propose a 3-step framework consisting of model processing, offline learning and online generation by evolution as illustrated in Figure 1.

- At very beginning, we collect corbel models designed by the professional and convert them into triangular mesh representation. Then we decompose the models into three parts: base, main body and decoration. Finally we extract some features for each part. The three parts together with the features form the underlying representation for our modeling task.
- During offline learning, we develop two networks: 2D CurveVAE and 3D VoxelVAE. The 2D CurveVAE model is trained for feature curves of main body, by which we could generate new feature curves by sampling in the latent space. The 3D VoxelVAE model is trained to encode voxelized decoration part into a 64D latent space where similar and dissimilar decoration models could be easily retrieved.
- In online generation stage, by using the dataset and two feature latent spaces created earlier, we encode a corbel model by a gene vector. Then we apply evolution principle to generate a group new models from the one specified by the user, which meets the requirements of fitness and diversity. The final mesh models are synthesized by a feature-driven deformation method.

The details of these processes will be described in the next two sections.

IV. MODEL PROCESSING

Currently we bought 125 corbel models from online 3D model warehouse (www.cgtrader.com) as initial model collection. All these models are triangular meshes in OBJ format, designed by industry designers. To facilitate creative modeling, we extract structure and feature information.

A. Decomposition

According to the characteristics of corbels, we decompose a corbel into three semantically meaningful parts: base, main body and decoration. The base usually has very regular shape and is in touch with the supporting structure such as a wall or a ceiling. The main body represents the overall shape of the corbel. The decoration accounts for the geometric details (see Figure 2). The decomposition can be done manually or by algorithms [6].

Figure 2: A corbel model is decomposed into base (left), main body (middle) and decoration (right) parts.

B. Feature Extraction

For a main body, its front face corresponds to the decoration part and is usually curved. This curved shape represents the overall feature of the main body. Therefore we propose to fit a planar cubic B-spline curve to the profile of this curved shape, as shown in Figure 3. The starting and ending points of the B-spline curve are used as guidelines for decorations to be set on the main body component. Similarly, for the decoration, we can also construct a cubic B-spline curve for the side corresponding to the main body. These B-spline curves serve as the feature curves for the main body and decoration parts, and they will drive the deformation of both parts in the later stage of synthesizing new corbel models.

Figure 3: Feature Extraction for main body: (a) Main body with decoration; (b) Main body with decoration removed; (c) Feature curve of the main body, where red and blue points are starting and ending points for aligning decoration.

For a base part, the feature is captured respectively as the width, height, and depth value of the model, along with a reference point as shown in Figure4. The reference point serves as a pivot point for the main body to align onto the base.

As a result, each corbel model can be decomposed into a base, a main body and a decoration. Each of these parts has corresponding feature. We treat each part and feature as a design element or building block. Thus we assembly them

Figure 4: (a) Features of a base; (b) Feature of a base without top (b); (c) Features of a base without back.

according to their categories which form a design space. Specifically, our design space is composed of the following elements:

- Main body: a set of triangular meshes $\{Mmb_i\}$ and a set of feature curves in B-spline representation $\{Cmb_i\}$
- Decoration: a set of triangular meshes $\{Md_i\}$ and a set of feature curves in B-spline representation $\{Cd_i\}$
- Base: a set of triangular meshes $\{Mb_i\}$ and a set of features in the form of reference points together with width, height and depth values $\{Cb_i\}$.

V. CREATIVE MODELING

This section presents the core techniques for our creative modeling, which consist of machine learning, generation by evolution, and feature-driven deformation.

A. Offline Learning

To facilitate generating new models by evolution principle, we train low dimensional vectors to represent feature curves and decoration. For this purpose, we choose VAE, a generative model that allows machine to learn the representation of data and generate new data instances in good quality. Particularly, in VAE, the input is passed into the encoder layer and encoded as a distribution over the latent space. After encoding is performed, a point from the latent space is sampled from that distribution and passed into the decoding layer where it is decoded for a new instance.

Figure 5: (a) Control points with normal; (b) Control points moved along the normal.

1) *2D CurveVAE*: With the collected corbel models, we extract main body feature curves that are however insufficient for training a good learning model. Hence we perform data augmentation to increase the diversity of main body feature curves. Specifically, for each control point of the B-spline curve that represents a main body feature curve, we

assign the normal of the curve at the point nearest to the control point, as shown in Figure 5a. Then we move the control point along the normal by an offset value from a range between -2 to 2 units (see Figure 5b), which results in the shape change of the feature curve. The augmentation process randomly selects the control points for offsetting. This expansion process is applied to all existing feature curves and eventually we produce 2989 feature curves in the dataset.

The whole set of produced feature curves is then passed into a VAE model. The goal is to obtain the encoding and decoding of the data with minimum information lost. The architecture of our 2D CurveVAE is shown in Figure 6. The input curve is represented as x in $192 \times 2D$ where 192 is the number of points sampled from the B-spline curve and 2 corresponds to 2 dimensions. Let $Enc(\cdot)$ and $Dec(\cdot)$ denote the encoder and decoder of our 2D CurveVAE. $z = Enc(x)$ is a latent encoding vector and $x' = Dec(z)$ is a reconstructed curve. The relationship between the input data x and the latent encoding vector z can be fully defined by prior $p_\theta(z)$, likelihood $p_\theta(x|z)$ and posterior $p_\theta(z|x)$. The estimated posterior $q_\phi(z|x)$ should be very close to the real one $p_\theta(z|x)$. Our 2D CurveVAE tries to minimize the following loss:

$$L_{2DCurveVAE} = \lambda L_{recon} + \mu L_{KL} + L_{Reg} \quad (1)$$

where λ and μ are the weights of loss terms, $L_{recon}(x, x')$ measures the reconstruction error, $L_{KL} = D_{KL}(q_\phi(z|x) \parallel p_\theta(z|x))$ is the Kullback-Leibler divergence to promote Gaussian distribution in the latent space, and L_{Reg} is the regularization term of the network parameters in L_2 norm.

With a trained 2D CurveVAE on the dataset of $n = 2989$ main body feature curves, the latent space vector from the encoder to the decoder that best represents the feature curve of the main body is learned. The encoded representation is a 3D vector where 3 is the number specified as the latent dimensions in the training model to encode the shape of the curve. The Gaussian distribution makes it effective to generate new curves by sampling in the latent space.

2) *3D VoxelVAE*: To compress the representation of decoration, we propose a 3D VoxelVAE, a variant of VAE model, where there is an additional voxel conversion layer before the real dataset is passed into encoder. The meshes of all decorations are first transformed into 3D voxel grids. The dimension of the voxel grid is $128 \times 128 \times 16$, for width, height, and depth, respectively. Each grid is associated with a voxel density. The density is the number of sampled points of the surface mesh that are within the particular voxel space.

With a trained 3D VoxelVAE model using the dataset, the representation value of the decoration is learned. The procedure is similar to 2D CurveVAE, while the output is a 32D vector where 32 is the number specified as the latent dimensions in the training model to encode the feature of the decoration. Within the latent space, similar decorations will

Figure 6: Architecture of 2D CurveVAE.

have a small Euclidean distance, and less similar decorations will have a big Euclidean distance.

B. Online Generation

In the online generation stage, the user specifies one model in the dataset and the algorithm automatically generates a set of new corbel models. The process involves evolution and deformation.

1) *Generation by Evolution*: To realize creative modeling, we propose an evolution algorithm. With all the design elements in the dataset and two VAE models, we can encode each corbel model by a gene vector structure *DNA* which is defined as follows:

$$DNA = \langle Id_{mb}, Id_d, Id_b, fc_{mb}, Par \rangle \quad (2)$$

where Id_{mb} , Id_d and Id_b are indices of main body, decoration and base in the dataset, respectively, fc_{mb} represents the 3D latent vector of the main body feature curve, and Par is the set consisting of base features and starting/ending points for the feature curve.

The outline of the algorithm is given in Algorithm 1. Given a user-selected model M_s , a *DNA* is generated. Then we select the best-fit candidates by *selectParents* operation. By evolution, a set of new gene vectors is produced by cross-over and mutation operations. For each new gene vector, a corbel model is synthesized using the feature-driven deformation that will be described later in this section. The generated corbel models are evaluated by function *EvalPopulation* and only the models in good geometric and physical quality survive. The evolution continues until the number of generated models reaches the target threshold or the user suspends the process.

Assume that the target number of generated models is N . The steps of the algorithm are elaborated below.

- $P_i \leftarrow \text{selectParents}(\mathcal{B}, \mathcal{L}, M_s)$

This function randomly selects $0.2N$ corbel models with their *DNA* vector from $\mathcal{B}$, which will be used with the user-selected model as parents for mutation. In addition, based on the user input M_s , we inquire top $0.5N$ similar and $0.25N$ dissimilar decorations in the latent space of 3D VoxelVAE. We also select $0.6N$ bases from $\mathcal{B}$.

Algorithm 1 Creative corbel model evolution

- Input 1:** initialize population $\mathcal{G}_0 = \{M_i\}_{i=1}^{125}$
Input 2: 3D VoxelVAE latent space $\mathcal{L}$
Input 3: User-selected model M_s
Input 4: target number of generated models N

- 1: Background set $\mathcal{B} \leftarrow \mathcal{G}_0$
 - 2: $i=0$;
 - 3: **while** not termination **do**
 - 4: //select the best-fit individuals for reproduction;
 - 5: $P_i \leftarrow \text{selectParents}(\mathcal{B}, \mathcal{L}, M_s)$
 - 6: //breed new individuals through crossover and mutation operations;
 - 7: $\mathcal{G}_{i+1} \leftarrow \text{Reproduce}(P_i)$
 - 8: //evaluate the individual fitness of new individuals;
 - 9: $\mathcal{G}_{i+1} \leftarrow \text{EvalPopulation}(\mathcal{G}_{i+1})$
 - 10: $\mathcal{B} \leftarrow \mathcal{B} \cup \mathcal{G}_{i+1}$
 - 11: $i=i+1$;
 - 12: **end while**
-

- $\mathcal{G}_{i+1} \leftarrow \text{Reproduce}(P_i)$
The cross-over operation is applied on *DNA* of M_s and selected $0.2N$ corbel models. The mutation operation is then applied by randomly selecting Id_{mb} , Id_d , and Id_b in *DNA* vector and replacing them with the selected candidates. fc_{mb} may also be perturbed in the latent space. The values in *Par* will undergo random minor adjustments.
- $\mathcal{G}_{i+1} \leftarrow \text{EvalPopulation}(\mathcal{G}_{i+1})$
This function is to evaluate the fitness of generated corbel model. The geometric quality is evaluated by checking the existence of self-intersection and the physical quality is evaluated by checking the centroid of side section of main body against the center of the whole model.

2) *Feature-driven Deformation*: What remains now is how to synthesize the 3D corbel model given a gene vector.

Given a new main body feature curve, two deformations are performed with respect to this curve: one for the main body and the other for the decoration. We use direct manipulation of free form deformation (FFD) with sampling points

Figure 7: 3D & 2D views of feature curve driven deformation. (3D views: a,c,e,g; 2D views: b,d,f,h; i: combination of the deformed main body and decoration; j: combination of deformed main body, decoration and base.)

on the feature curves as manipulating points. In Figure 7b, we construct a 2D FFD region shown in the light blue control lattice specified by the bounding box of the side slice of the main body, (X_{min}, Y_{min}) and (X_{max}, Y_{max}) . With uniform cubic B-spline setting and linear initialization, the FFD function is $P(s, t) = \sum_{i=0}^m \sum_{j=0}^n P_{ij} B_i^3(s) B_j^3(t)$, where $P(s, t)$ represents the coordinates of the deformed point with local coordinates (s, t) , and P_{ij} are the control points. The deformation is driven by deforming sampled 100 points $P(s_k, t_k)$ on original feature curve l_{ori} in purple color to Q_k on the target feature curve l_{tgt} in blue color. This is achieved by solving the following minimization problem:

$$\min_{\{P_{ij}\}} \sum_{k=1}^{100} \|P(s_k, t_k) - Q_k\|^2 + \lambda E_L + \beta E_S \quad (3)$$

where $E_L = \sum \|P_{ij} - P_{ij}^{ori}\|^2$ for P_{ij} with $i = 0$ or $j = 0$ is to fix the red control points on top and left lines so that the L shape in the corner part of the model could be kept; $E_S = \sum \|P_{ij} P_{i+1j} - P_{ij}^{ori} P_{i+1j}^{ori}\|^2 + \sum \|P_{ij} P_{ij+1} - P_{ij}^{ori} P_{ij+1}^{ori}\|^2$ is to maintain the edge length between two neighboring control points so that the chance of self-intersection could be reduced; and λ and β are the trade-off parameters. The new positions of control points P_{ij} are the solution of the minimization problem (3). The 3D FFD lattice on the main body is shown in Figure 7a and is adjusted according to the new positions of 2D FFD so as to deform the mesh Mmb_i into the one in Figure 7c.

To deform a decoration, the mesh Md_j and its feature curve Cd_j are first scaled and translated to match the starting and ending points on the feature curve of the main body. The similar direct manipulation FFD method without constraint of E_L is applied to the decoration part as shown in Figures 7e, 7f and Figures 7g, 7h. Figure 7i shows combination of the deformed main body and decoration. Then the base part is simply scaled according to the size of the bounding box of the main body and is then combined with the main body, as shown in Figure 7j.

After all the deformations are performed, all the parts are ready to be combined. The combination of the parts (main body, decoration, and base) is achieved through a union

operation. A new corbel model is eventually generated. In future, the simple union operation will be improved by adapting more sophisticated composition methods such as [20]–[22].

VI. EXPERIMENTS

This section reports the results of experiments conducted to evaluate the proposed techniques.

A. Experiments on VAE models

We first evaluate the 2D CurveVAE model. Figure 8a shows 64 feature curves of main bodies in the original dataset. Figure 8b shows 64 new feature curves generated with feature curve generator learned by VAE. It can be seen that the feature curves from Figure 8b have similar curvature as the feature curves in Figure 8a.

We then evaluate 3D VoxelVAE. The output is a latent feature space L where each decoration is encoded into a 32D vector. The enquiry is made based on the Euclidean distance between vectors in L . Figure 9 shows the results of similar and dissimilar decoration models returned based on the query of a flower shaped decoration, which demonstrates that the compressed low dimensional latent feature space L can represent decorations well.

B. Experiments on Deformation and Parameters

Experimentation was conducted on the deformation of decoration components and also on how the adjustment of the position to mount the decoration to the main body affects the result.

As seen in Figure 10, the decoration mounted on the main body with different ending points can still generate aesthetically pleasing models. However, it is worth pointing out that not all positions are suitable for decorations to be mounted on and further selection is required during the evolution procedure to fine tuning the parameters.

Figure 11 shows some results of combining a decoration with different main bodies using proper parameter setting. Notice that the locations of the starting and ending points on the main body automatically affect the size of the decoration in the output.

Figure 8: Feature curves of main body: (a) original feature curves; (b) feature curves generated by VAE.

Figure 9: Decorations retrieved from the feature space.

Figure 10: Aligning a decoration to a main body with different parameters.

C. Experiments on Evolution

Figure 12 shows a group of models generated by our method according to a user input model. Most of the models generated have the same main body as the input model, but vary in curvature and shape, while about 25% of the models randomly choose other main bodies in the dataset. Most of the generated decoration parts look in a similar style, but have different details or geometry textures. The starting and ending points for the mounting of decorations vary from model to model. The base parts of the models are randomly selected for cross-over. This experiment demonstrates that our developed tool can achieve modeling creativity to a certain extent.

VII. CONCLUSIONS

We have described a creative corbel modeling tool, the underlying techniques of which are geometric modeling and machine learning. In particular, VAE models are developed to learn the latent representation of feature curves and decoration shape, evolution principle is used to develop a

modeling algorithm that evolves to produce new offsprings, and feature-driven deformation is developed to realize the creation of new corbel models. The experiments demonstrate that the developed method and tool are able to produce new models with creativity, and also showcase the success of the use of classic geometric modeling and advanced machine learning techniques in corbel modeling. Our future work will further optimize the algorithmic design and extend the method to other decorative architectural geometry.

ACKNOWLEDGMENT

This work is supported by the Ministry of Education, Singapore, under its MoE Tier-2 Grant (2017-T2-1-076).

REFERENCES

- [1] T. Funkhouser, M. Kazhdan, P. Shilane, P. Min, W. Kiefer, A. Tal, S. Rusinkiewicz, and D. Dobkin, *Modeling by Example*. ACM SIGGRAPH, 2004.
- [2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, *Generative adversarial nets*. NIPS, 2014.
- [3] D. Cohen-Or and H. Zhang, *From inspired modeling to creative modeling*. The Visual Computer, 2016.
- [4] K. Xu, D. Cohen-Or and B. Chen, *Fit and Diverse: Set Evolution for Inspiring 3D Shape Galleries*. ACM Trans. Graph., 2012.
- [5] E. Kalogerakis, S. Chaudhuri, D. Koller, and V. Koltun, *A Probabilistic Model for Component-based Shape Synthesis*. ACM SIGGRAPH, 2012.
- [6] M. Sung, H. Su, V. Kim, S. Chaudhuri and L. Guibas, *ComplementMe: Weakly-Supervised Component Suggestions for 3D Modeling*. ACM Trans. Graph., 2017.
- [7] D. Kingma and M. Welling, *Auto-encoding variational bayes*. arXiv:1312.6114 (2013).
- [8] J. Wu, C. Zhang, T. Xue, W. T. Freeman and J. Tenenbaum, *Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling*. Advances in Neural Information Processing Systems, 2016.

Figure 11: Aligning a decoration to different main bodies with different parameters.

Figure 12: Results of evolution from an input model.

- [9] M. Tatarchenko, A. Dosovitskiy, and T. Brox, *Octree Generating Networks: Efficient Convolutional Architectures for High-resolution 3D Outputs*. ICCV, 2017.
- [10] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, *Multi-view convolutional neural networks for 3D shape recognition*. ICCV, 2015.
- [11] C. Qi, H. Su, K. Mo, and L. Guibas, *PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation*. CVPR, 2017.
- [12] C. Qi, H. Su, K. Mo, and L. Guibas, *PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space*. NIPS, 2017.
- [13] Z. Chen and H. Zhang, *Learning Implicit Fields for Generative Shape Modeling*. CVPR, 2019.
- [14] Q. Tan, L. Gao, Y. Lai, and S. Xia, *Variational Autoencoders for Deforming 3D Mesh Models*. CVPR, 2018.
- [15] T. Groueix, M. Fisher, V. Kim, B. Russell, and M. Aubry, *AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation*. CVPR, 2018.
- [16] J. Li, K. Xu, S. Chaudhuri, E. Yumer, H. Zhang, and L. Guibas, *GRASS: Generative Recursive Autoencoders for Shape Structures*. ACM Trans. Graph., 2017.
- [17] H. Wang, N. Schor, R. Hu, H. Huang, D. Cohen-Or, H. Huang, *Global-to-local generative model for 3D shapes*. ACM Trans. Graph., 2018.
- [18] Z. Wu, X. Wang, D. Lin, D. Lischinski, D. Cohen-Or and H. Huang, *SAGNet: Structure-aware Generative Network for 3D-Shape Modeling*. ACM SIGGRAPH, 2019.
- [19] L. Gao, J. Yang, T. Wu, Y. Yuan, H. Fu, Y. Lai and H. Zhang, *SDM-NET: deep generative network for structured deformable mesh*. ACM Trans. Graph., 2019.
- [20] J. Zheng, G. Wang, and Y. Liang, *Curvature continuity between adjacent rational Bézier patches*. Comput. Aided Geom. Des., vol. 9, no. 5, pp. 321–335, 1992.
- [21] Y. Yu, K. Zhou, D. Xu, X. Shi, H. Bao, B. Guo, and H. Shum, *Mesh editing with Poisson-based gradient field manipulation*. ACM Trans. Graph., vol. 23, no. 3, pp. 644–651, 2004.
- [22] W. Yang, J. Zheng, J. Cai, S. Rahardja, and C. Chen, *Natural and seamless image composition with color control*. IEEE Trans. Image Process, vol. 18, no. 11, pp. 2584–2592, 2009.