



3D building fabrication with geometry and texture coordination via hybrid GAN

Zhenlong Du¹ · Haiyang Shen¹ · Xiaoli Li¹ · Meng Wang¹

Received: 31 December 2019 / Accepted: 17 August 2020
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

Abstract

3D building plays the essential role in digital city construction, city augmented reality and smart urban planning & design. Conventional building construction is accomplished by modeling software which requires significant human intervention. In this paper, a method of 3D building fabrication via Hybrid generative adversarial network (GAN) is proposed, in which a loss function with the introduction of cycle consistency loss and perceptual loss is given, a multi-properties GAN chain is built to create the building with complex architectures. Additionally, a mixed GAN network to generate the geometry and texture coordination is put forward. The discussed method can refine rough architectural models for outputting realistic buildings. Experiments show that generated 3D buildings utilizing the presented method are realistic, with geometry and textural consistency, which improves performance by 20% over traditional methods.

Keywords GAN chain · Hybrid GAN · 3D building generation · multi-properties generation

1 Introduction

Currently, 3D building models are one of the most important components in digital city construction. They play an indispensable role in urban construction & planning and comprehensive city management. 3D modelling construction has been a hot topic in the fields of industrial vision, product prototype design, 3D printing, and building information modeling (BIM). Although CAD software, such as, ProE, 3DSMax, Maya, Unity, Unreal Engine 4 or AutoCAD, has been utilized to make models for several decades, building generation is still labor intensive and time consuming. Deep learning (Wen et al. 2019; LeCun et al. 2015; Lu et al. 2018, 2019; Uemura et al. 2019; Zhao et al. 2018; Xu et al. 2019b, a; Singhal et al. 2019; Kim and Chung 2020)

provides a solution to the issue. Generative adversarial networks (GAN) (Arora et al. 2017; Goodfellow et al. 2014) can learn the distribute from given data and generate new data. It can be applied to generate 3D building models. In this paper, a novel 3D building model generation approach based on hybrid GAN is further investigated.

GAN is based on the principle of a two-player zero-sum game. It involves a generating model G (generator) and a discriminant model D (discriminator). Generator G captures the distribution of sample data, and generates the new data similar to the training data with noise z following a certain distribution (uniform distribution, Gaussian distribution, etc.). Discriminator D is a two-classifier, which estimates the probability of data derived from the training data. In this paper, the loss function composed of the domain classification loss, cycle consistency loss, identity mapping loss and perceptual loss, is presented for StarGAN, which can efficiently fabricate data with diversity and consistency. The two new additions, the identity mapping loss and the perceptual loss, are introduced to the loss function, which improves the quality of 3D building generation.

Individual GAN can only implement one kind of attribute mapping. In order to enable GAN to generate multi-attribute data, this paper combines multiple GANs to form a GAN chain. The output of the previous GAN is used as the input for the next GAN, thus forming a chain that leads to the

✉ Zhenlong Du
duzhl-cad@163.com

Haiyang Shen
ashyhr@163.com

Xiaoli Li
lixl@njtech.edu.cn

Meng Wang
wm97211@163.com

¹ Nanjing Tech University, Nanjing, Jiangsu, China

final results. By combining different mapping GANs into one chain, the discussed approach can generate both 2D textures and 3D building models. Moreover, manipulators can modify the distribution in the middle of the chain to achieve ideal final results. The hybrid GAN chain proposed in this paper can efficiently produce roof, wall and window textures. Additionally, it fabricates realistic textures with contrasts, lighting and window frames, which makes buildings more realistic even though these details are not explicitly required.

Currently, 3D building models cannot be generated directly by GAN. To achieve a 3D building model with consistent texture, this paper presents a hybrid GAN framework which is combined by two kinds of GAN chains, one of which generates texture while the other produces the building model. It guarantees consistency between building models and textures. The texture can be consistently attached to the 3D model to make a complete building.

FrankenGAN (Kelly et al. 2018) generated 3D buildings by combining cVAE-GAN (Bao et al. 2017) and cLR-GAN, and adopted BicycleGAN network chains. It can fabricate building models, but not efficiently enough. Inspired by FrankenGAN, this paper proposes a 3D building generation method with geometry and texture coordination via hybrid GAN. In the paper the proposed 3D building generation method is an improvement on FrankenGAN. These improvements embody that a loss function with new introduction of the identity mapping loss and the perceptual loss, which can ensure details are preserved in the data conversion process; the BicycleGAN used by FrankenGAN is replaced with StarGAN for generating high-quality textures used by building models; multiple GANs are synchronized to generate a consistent pattern distribution in the building production. Experiments show that the method proposed in this paper can generate 3D buildings with reasonable semantics, authenticity, and consistency in styles.

2 Related works

High quality models are usually reconstructed from radar scanned data, and the procedural modeling (Wu et al. 2018; Zhang et al. 2013; Chen et al. 2018) is good choice for 3D building generation. Schwarz and Muller (2015) proposed a novel CGA++ architecture for procedural modeling, which provided a general and integrated solution to procedure modeling. This kind of rule-based procedural modeling is often used to fabricate specific buildings, such as, ancient buildings. Methods (Lin et al. 2011; Bao et al. 2013) are proposed later for further optimizing the procedural modeling. Ilik et al. (2015) proposed a new procedural modeling method which provides the UI for users to model a building facade interactively. The facade is obtained by combining multiple overlapping layers, each of which contains a single linear

grid element described by two simple generator patterns. The design process became more intuitive and the amount of editing for complex layouts decreased significantly.

3D model fabrication is indispensable to 3D printing (Chen et al. 2018). How we fabricate a complicated model meeting requirements, is currently a hotly researched area. Chen et al. (2018) suggested an approach to generating 3D models which achieved the optimal internal elementary blocks and an acceptable residual volume, it could produce reasonable geometry models with the support waste. However, this paper investigates the convenient and efficient 3D building model generation.

Building facade brings on the versatility of appearance. Therefore, how to quickly and easily devise facade has been a hotly researched issue (Wu et al. 2018; Zhang et al. 2013; Chen et al. 2018; Schwarz and Muller 2015; Bao et al. 2013; Ilik et al. 2015; Dang et al. 2014) Some of the previous papers (Dang et al. 2014) suggested an interactive facade design method which analyzed user interaction semi-automatically, first to detect the elements of a facade and their relationship. Variation is applied to these elements and relations, then facade is achieved through optimization, meanwhile unreasonable modification is removed. Moreover, user specification and incremental editing are important for quickly generating the desired building configurations, Vanegas et al. (2012) proposed a framework which combined the procedural urban modeling and the inverse urban modeling, in which the procedural modeling for generating the prototypal building layouts and the inverse modeling progressively modifies the found buildings until a satisfactory design is achieved. In (Vanegas et al. 2012), manipulators unfamiliar with the construction rules of urban modeling there are difficulties producing ideal urban buildings.

3D model generation can be accomplished either by various types of CAD software or by reconstruction from images. The reconstruction is achieved by recovering the shape, material and illumination. Existing 3D model reconstruction methods (Huang et al. 2015; Nash and Williams 2017; Hu et al. 2018a; Wang et al. 2018; Hu et al. 2018b) can produce rather accurate results. Deep learning or GAN can implement the 3D object generation by introducing the object reconstruction procedure. Girdhar et al. (2016) combined the 3D auto encoder with the image encoder, mapped the 3D shape with the image into a common potential space, established a connection between the 3D model and images, and thus reconstructed a 3D object from the given images. Wu et al. (2017) proposed a step-by-step end-to-end network for 3D model generation, which recovered 2.5D sketches and 3D models from a given image.

Wu et al. (2016) also proposed a 3D object generation method from GAN, which employs the nearest volumetric convolution network and GAN for fabricating 3D objects from a probability space. Kelly et al. (2018) proposed a

FrankenGAN neural network to progressively introduce details to the rough model. The given network builds several BicycleGAN (Zhu et al. 2017) chains into one routine to generate semantically consistent geometric, texture and accessories.

Utilization of data features is relevant to the degree of data exploitation. These features include explicit and implicit ones. Conventional methods only make use of explicit features within the data, and there is a limit to what can be learned from the explicit features. Deep learning fully discovers the representation capability of the features, and the knowledge acquired in the process has a strong representative ability. In the paper the wall and roof texture are handled by GAN, and features are exploited by GAN.

3 Automatic 3D building generation with the geometry and texture coordination

3D building generation involves the layout of windows and doors, building types, as well as wall and roof textures. Manifestly, it is a big challenge to generate a 3D building because there are too many components included, and these should all be coordinated with each other. In the paper, hybrid GANs are employed to generate multi-properties concurrently, while GANs chain is exploited for step-by-step generation of complicated textures, while the novel loss function is applied to the used GAN.

3.1 Star generative adversarial networks

Star generative adversarial networks (StarGAN) (Choi et al. 2018) is a generative confrontation network that can generate data in multiple domains within one model. As we know, when there are many areas where data needs to be mapped, a model should be re-trained for each domain. StarGAN networks can implement multiple domain mapping within one unified framework, which apparently improves the efficiency of network training.

StarGAN is composed of one generator and one discriminator, it can implement one-to-many data mapping. StarGAN generator network is shown in Fig. 1. A single generator learns to transfer among multiple property sets (domains). StarGAN accepts data from multiple domains and utilizes only one generator to learn mapping relationships among multiple domains. In conventional GAN, at least three generators are needed to learn the mapping from property P_1, P_2, P_3 and P_4 , while StarGAN employs only one generator to achieve the mutual transformation among domain P_1, P_2, P_3 and P_4 .

In Fig. 1, StarGAN constructs its underlaying network by training a property set $\{P_1, P_2, P_3, P_4\}$, and depending on

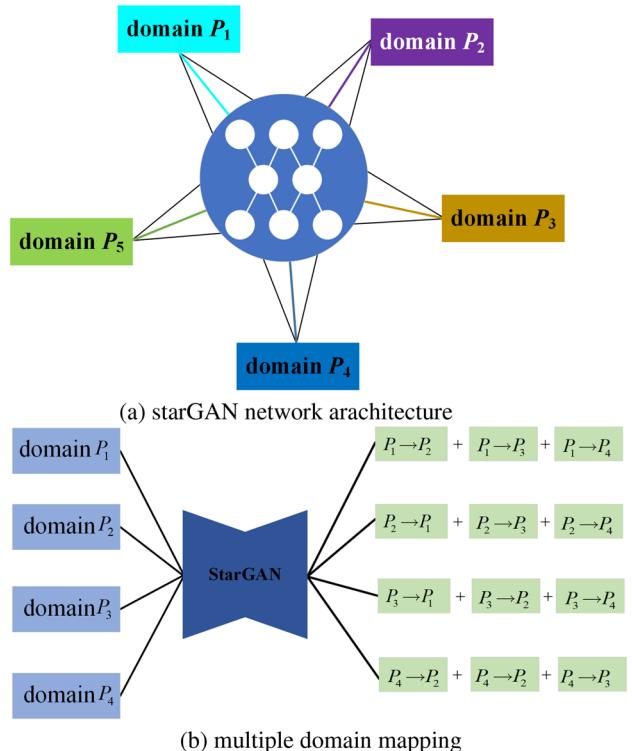


Fig. 1 StarGAN generator network

this network achieves the mapping of multiple properties, for example, $P_1 \rightarrow P_2, P_1 \rightarrow P_3, P_1 \rightarrow P_4, \dots, P_4 \rightarrow P_3$. In the paper, geometrical models and textures are separately served for the base and appearance of 3D buildings, and geometry & texture are different properties, which are handled by two different StarGAN. These combine together to form the algorithmic core.

StarGAN contains two modules, one of which is the generator G and the other is the discriminator D . The target domain information c is added to the input of G , that is, the domain to which the picture is translated, which informs the generation model. In addition to the function of judging whether the picture is real, D also has the ability to judge which category the picture belongs to. This can ensure that the same input image in G produces different effects with different target areas. StarGAN training procedure is shown by Fig. 2.

In the StarGAN training stage, the input data x and the target property label c are input to the generator G . Herein c provides G with the target property, and G attempts to generate a data $\hat{x} = G(x, c)$, which is hard to distinguish from the real data x , and it is sent to the discriminator D together with the real data y . \hat{x} and y are differentiated by D , which learns to distinguish between y and $G(x, c)$ and classifies the generated data to the corresponding category. When D determines that \hat{x} is a spurious datum, \hat{x} and c are again input into

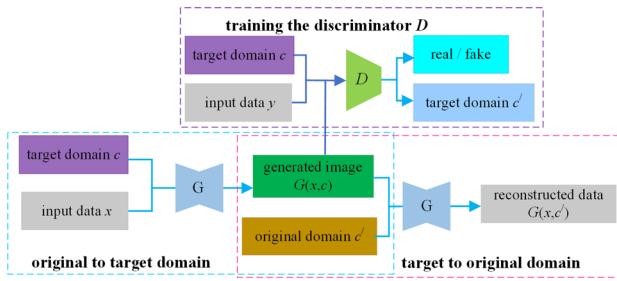


Fig. 2 StarGAN training procedure

D. When the source property label is given, *G* will attempt to reconstruct the original data from the generated data \hat{x} . The reconstruction is to transfer the data from domain c_i to domain c_j . Generally, the content can not be changed very much, and the difference $\Delta c = |c_i - c_j|$ is used to evaluate the property variation. StarGAN training procedure is shown by Fig. 2.

3.1.1 StarGAN loss function

StarGAN trains a single generator *G* to learn the mapping among multiple domains. The input image data *x* and the target domain label *c* are used to fabricate $y \leftarrow G(x, c)$. The auxiliary discriminator (Wang et al. 2018) allows a single discriminator to control multiple properties mapping switch. The discriminator generates a probability distribution $H : x \rightarrow \{H_{src}(x), H_c(x)\}$ on the input data *x* and the domain label *c*.

Confrontation loss StarGAN training is an antagonistic game between *G* and *D*. *G* continuously improves the ability of learning data distribution and data generation, meanwhile *D* improves the discriminating ability on the data authenticity. A dynamic balance between *G* and *D* is achieved when the network reaches a steady state in the process of the confrontations. The confrontational relationship between *G* and *D* is represented as follows:

$$L_{adv} = E_{x \sim p(x)} [\log D_{src}(x)] + E_{x \sim p(x), c \sim p(c)} [\log(1 - D_{src}(G(x, c)))] \quad (1)$$

in which *x* is the input data, *c* is the input target property label, $G(x, c)$ is the generated data, *E* is the mathematical expectation, and $D_{src}(x)$ is the probability distribution of *x* given by *D* under the condition attribute *c*. In the training process, *G* tries to minimize $D_{src}(x)$, while *D* also attempts to maximize $D_{src}(x)$. When a dynamic balance is achieved between *G* and *D*, StarGAN training is accomplished.

Domain classification loss For the discriminator *D*, the real image needs to be classified into the correct category; for the generator *G*, the generated image needs to be classified into the target category. StarGAN converts the input

data *x* and target domain label *c* into an output data *y* and then *y* is classified into a target domain *c* by the assistant classifier. On one hand, the auxiliary classifier optimizes the classification loss of generated data by *G*, and uses the generated data to train in the target domain, as is shown in Eq. 2,

$$L_{cls}^G = E_{x \sim p(x), c \sim p(c)} [-\log D_{cls}(c | G(x, c))] \quad (2)$$

where $D_{cls}(c | G(x, c))$ is the probability distribution of the generated data calculated by *D* on the target domain *c'*. The generator hopes that the generated data can be determined by the discriminator as the target classification *c*, so the goal of the generator is to minimize the loss function L_{cls}^G . On the other hand, the auxiliary classifier optimizes the domain classification loss of the real data via *D*, and trains the real data in the original domain, as is shown in Eq. 3,

$$L_{cls}^D = E_{x \sim p(x), c' \sim p(c')} [-\log D_{cls}(c' | G(x, c))] \quad (3)$$

Where $D_{cls}(c' | G(x, c))$ represents the probability distribution that the discriminator classifies the real samples into the corresponding label category *c'*. The goal of discriminator *D* is to minimize the loss function L_{cls}^D .

Cycle consistency loss In order to ensure that the fabrication data only alters the content relevant to the input property while retaining other input contents to be fixed, the cycle consistency loss is defined by Eq. 4,

$$L_{cyc}^G = E_{x \sim p(x), c \sim p(c), c' \sim p(c')} [\|G(G(x, c), c') - x\|_\rho], \quad (4)$$

where *G* takes the generated data $G(x, c)$ and the original domain label *c'* as the new input to reconstruct the data *x*, and ρ is a positive constant. Here the same generator is used twice. The original image is first converted to the image of the target domain, and then the generated image is reconstructed back to the original image.

3.1.2 Newly introduced loss function terms

Identity mapping loss To maintain the consistency between the input data and the generated one, the identity mapping loss is applied to measure the differences between the input data and the fabricated content as is defined in Eq. 5,

$$L_{id}^G = E_{x \sim p(x), c' \sim p(c)} [\|(G(x, c') - x)\|_\rho], \quad (5)$$

in which the data *x* belonging to the target domain *c'* is taken as the input data again to ensure that it remains unchanged after *G* handling. The difference between *x* and the output $G(x, c')$ is reduced by introducing the expected identity regularization.

Perceptual loss The detail is important to maintain the perceptual sensing. To minimize the loss of details, the

perceptual loss is applied to improve the detail conservation capability as is defined in Eq. 6,

$$L_{pl}^G = \frac{1}{dwh} \|\varphi(G(x, c)) - \varphi(x)\|_2^2, \quad (6)$$

where φ is the feature extraction function, d , w and h separately represent the depth, width and height. The loss function compares the features obtained by the real picture convolution with the features obtained by generating the picture convolution to make the high-level information (content and global structure) close.

Cycle consistency loss, identity mapping loss, perceptual loss and the comprehensive objective function applied by StarGAN are defined on the basis of the domain classification loss as is shown in the follow equation:

$$\begin{aligned} L_D &= -L_{adv} + \lambda_{cls} L_{cls}^D \\ L_G &= L_{adv} + \lambda_{cls} L_{cls}^G + \lambda_{cyc} L_{cyc}^G + \lambda_{id} L_{id}^G + \lambda_{pl} L_{pl}^G, \end{aligned} \quad (7)$$

where L_G and L_D are both the loss function adopted respectively by StarGAN generator G and discriminator D . L_{adv} is the conventional loss function and is reserved in StarGAN. λ_{cls} , λ_{cyc} , λ_{id} and λ_{pl} are the weights to control each loss function of the domain classification loss, cycle consistency loss, the identity mapping loss together with the perceptual loss, respectively.

3.2 Hybrid GAN for multi-properties generation

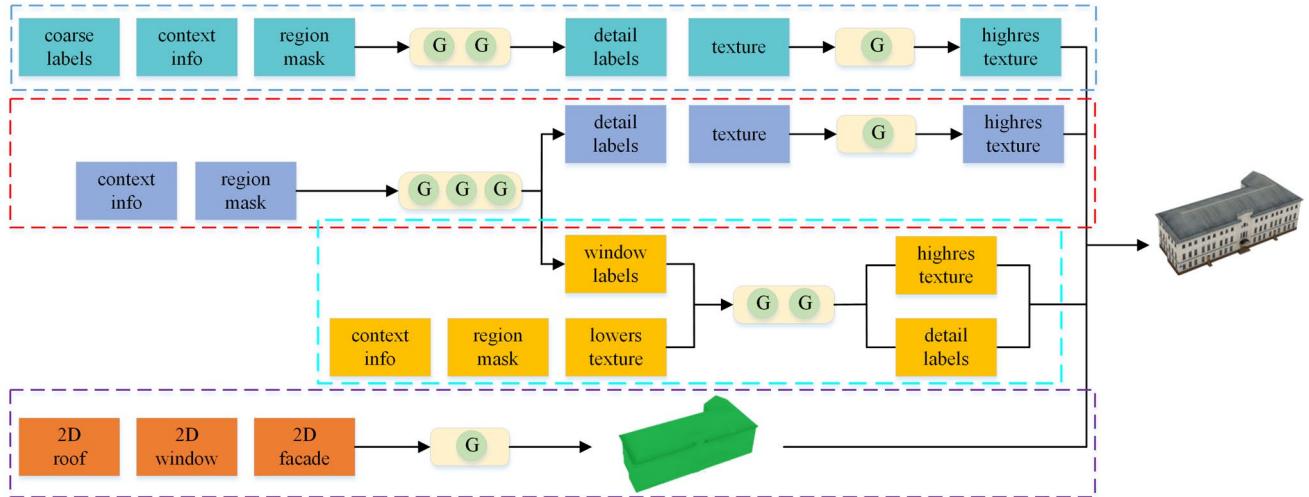
In this paper, the coordination between geometry and texture in 3D building enhances the appearance of realistic buildings. The building texture involves the roof, wall, and windows layout. No one GAN can generate 3D building required elements at one time, because one individual GAN can not

guarantee the consistency in texture size, geometry structure, that two can in alignment. Meanwhile different building styles require different kinds of texture. In the paper, the hybrid GAN chain is used to fabricate the building texture.

Figure 3 shows the hybrid GAN chain texture, six GANs are used together to fabricate a 3D building. Two GAN chains are applied to generate wall and roof textures. Two of the GAN chains further generate high-resolution textures of the walls and the roof. One generates high-definition window textures, and the third one is utilized to create window frames. Additional GANs are applied to create a more realistic 3D building appearance by utilizing lighting and shading. Since the GAN used here involves the generation of 3D data, starGAN is not applied in this instance. However, 3D-recGAN [19] is used for 3D model training and data generation. The resulting textures and geometry are added to the rough architectural model.

As is shown in Fig. 3, hybrid GAN is applied to generate the necessary building texture. The building model with the initial wall and roof defined by edge lines is sent to the hybrid GAN chain. The traditional single GAN has better performance in single property data fabrication, but building generation involves multiple properties and requires them to be consistent and coordinated. In the paper, multiple conventional GANs are combined to deal with the complicated building generation. The GAN hybridization embodies the coordination between single properties. Single property generation is referred to as the wall texture or roof texture generation. The coordination of multiple properties involves the coordination of geometry & texture, the consistency of wall texture and windows layout, the coherence of roof texture and chimney organization.

The significance of using multiple GANs to operate independently in the network is that the intermediate target



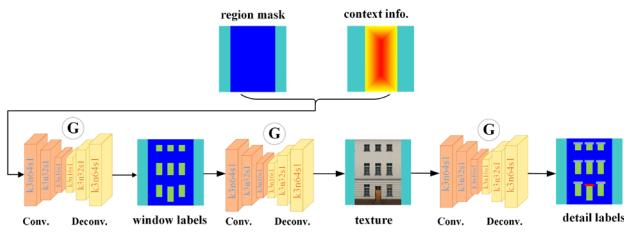


Fig. 4 Wall texture generation procedure

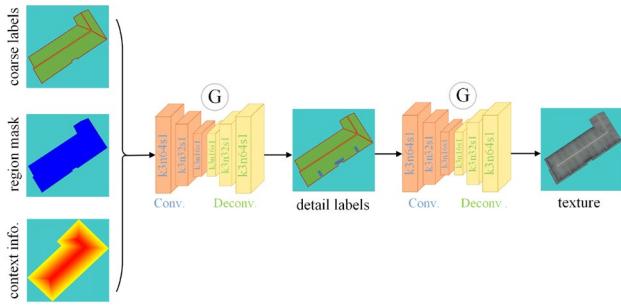


Fig. 5 Roof texture generation procedure

image formed first can provide guidance for subsequent steps. The layout of the intermediate target to get the desired result can be modified, which is more flexible than the end-to-end training method. Secondly, GAN training sometimes becomes unstable. Splitting a large-scale generation network into multiple small-scale generation networks can make training more stable and thus reduce training costs.

3.2.1 GAN chain for progressively fabricating the complex object

Textures and labels are initially generated through a single GAN chain. Each GAN performs a step to generate a transitional image beneficial for the final texture. Take the generation of wall texture for instance, the specific process is shown in Fig. 4.

In which the first GAN generates the window label from the region mask. The second GAN is input with window labels, which converts them into wall textures according to the label position of the windows. The third GAN accepts the generated wall texture by the second GAN, also detects the non-window labels, and generates a more detailed windows label image through regularization.

The procedure of roof texture generation is similar to that of the wall texture. As the Fig. 5 illustration, the labels, region mask as well as the context information are input together to the first GAN. The label image of the roof is signified with ridge lines and valley lines. The region mask describes the valid roof boundary. The context information

indicates the distance from the boundary. While the label image, region mask as well as context information provide the necessary feature, location and boundary to GAN. The first GAN generates the location of chimneys and the leaning roof, and the second GAN generate a rough roof texture. From the roof fabrication procedure, it can be observed that GAN chain progressively generates the roof texture.

The texture directly generated from GAN requires quality improvement. The mixed GAN chains are employed for improving the image resolution. The generated texture is divided into multiple patches to improve the quality. Each patch is enlarged and input into GAN to achieve the super-resolution patch. These patches are then stitched back to construct a super-resolution image. During the composition, the super-resolution image of the window is slightly different from the wall. In the paper, the window is cropped from the generated wall texture, and is scaled up and served for the input data to the GAN. The steps of window generation are similar to ones for the roof.

3.3 3D building generation

Methods of reconstructing 3D shape from images have been extensively investigated, such as Marrnet and ShapeNet. They train a large amount of data in attempt to generate a 3D model based on the perspective transformation. In the paper, a complete convolution network to generate 3D model is proposed, with generator G consisting of five volume convolutional layers, and the convolution kernel size is $4 \times 4 \times 4$, the step size is 2. After each convolution, one batch normalization layer and a *Relu* layer are used, and the last layer is a Sigmoid activation layer. D 's network structure is the same as that of G , the only difference being that the leaky *Relu* activation function is adopted by D while the *Relu* activation function is used by D . There is no pooling layer within the whole network structure.

Each batch updates the state of both G and D during training. For 3D building model generation, the computation cost used by G is longer than one of D . This makes it difficult for G to improve its weight from a much faster D , and results in all samples generated by G being differentiated as high score. In order to maintain the consistency between G and D , an adaptive training method is adopted, for each batch, G is updated only when the accuracy of the last batch is less than 80%. Experiments show that this improvement can lead to better training and achieve enhanced results.

In the paper, 3D building model, its related wall & roof texture, as well as windows, and window sills are blended together for achieving 3D building models. Manifestly, within these elements, some are 3D while others are 2D. The UV maps define the position of each point in the image, and these points are related to the 3D model. In the paper, UV maps together with the geometry generation are employed.

4 Experiments and evaluations

Experiments are implemented on a computer configured with NVIDIA GTX1050ti, 8G RAM and Intel Core i5 7300HQ CPU.

The paper mainly investigates how to efficiently construct a 3D building. State-of-the-art 3D building production methods combine existing parts that are extracted from objects based on the probability or geometric constraints to produce new shapes (Chaudhuri et al. 2011). Based on the initial distribution, GAN has been exploited to generate 1D and 2D data. However, little research has been conducted on 3D data generation via GAN.

Multiple types of data are involved in the paper, such as, texture, windows layout, and building models. These different type of data are used either by individual type or combined. The train texture data is downloaded from the internet, the building appearance data is from the CMP data set, and the building model is from ShapeNet.

Figure 6 demonstrates some 3D buildings generated by the discussed method in the paper. There are 9 groups of experiments, among them the first column shows the building model, the second one gives the building model with window configuration, the third and the fourth ones are two different views of generated buildings. The first three group's results are L-shape buildings, the next three lines show box-shape buildings, and the last three groups demonstrate T-shape buildings. Generated building with three different shapes proves that the given 3D building generation method can fabricate many shapes of building. From the second and the third column it can be observed that the window layout can integrate the wall texture, and the horizontal and vertical spacing of windows are consistent with the real building, moreover reasonable window layout with the appropriate number of doors are both consistent and realistic. Additionally the third and the fourth ones also show the generated roof, which covers the chimneys, spires as well as the slopes.

From Figs. 6 and Fig. 7 it can be observed that the given method by the paper can generate 3D buildings well. Figure 7 demonstrates a 3D building (the 9th row data in Fig. 6) with four different views, and this building is clearly presented for observers to see the wall and roof texture. 3D building in Fig. 7 includes a door, 178 windows, wall texture and other accessories, and these windows involve the large, small and medium tree types. Four views (from the front, side, top and perspective) are presented for fully presenting the generated model, and these figures are highly realistic with clear edges and fine details.

The presented method by the paper can fabricate not only single buildings, but also a group of buildings. These

buildings are situated on the regular or irregular area, and decorated with roads, trees and other accessories, as Fig. 8 illustrates. Figure 8a is composed of the same building generated by the put forward method, and Fig. 8b is constituted by the composite building, and these buildings are arranged symmetrically. Meanwhile, the top two results are buildings arranged on a regular rectangular area, while the bottom two are buildings arranged upon an irregular region. Figure 8 demonstrates that the method discussed in the paper can fabricate the buildings for urban planning.

Figure 9 compares the building generation time consumed by FrankenGAN, VON and ours, the total time used for constructing scenario in Fig. 8a-d. In these four scenarios, the method demonstrated by the paper all achieved the least amount of time compared to the original FrankenGAN and VON.

The distribution of the training data or the generated texture, model are evaluated and compared with Jensen-Shannon divergence (JSD) to calculate their similarity. Since the input data z is randomly sampled from a normal distribution, artifacts such as noise or incomplete portions are inevitably present in the generated data. However, as is shown by Fig. 10, the proposed method performs better than FrankenGAN & VON, and can produce the high-quality 3D buildings.

The quantitative measures including JSD, MMD-CD, MMD-EMD, COV-CD and COV-EMD are employed for evaluating G2L, SAGNet, FrankenGAN, VON and the present method in the paper. Table 1 shows the comparison result, in which the smaller the JSD and MMD the better the 3D building generation. The higher the COV score is, the more different results the present method can produce. Table 1 demonstrates that the proposed method has higher fidelity and diversity in 3D building.

The importance of each loss term to the model is conducted, and Table 2 shows the model performance by removing the loss function term from the GAN model. F-score is evaluated and employed as the harmonic mean of precision and recall, moreover the chamfer distance (CD) and earth movement distance (EMD) are also calculated. The bigger the F-Score the better 3D buildings are generated. The smaller CD and EMD, the better results are given. Through quantitative evaluation results, it is shown that if the same mapping loss and perception loss are deleted, the F-score is significantly reduced, while CD and EMD are increased.

Experiment results show that the proposed method can effectively generate realistic 3D buildings, and comparisons testify that the quality and diversity of generated 3D building by the discussed method are better than the original FrankenGAN and VON.

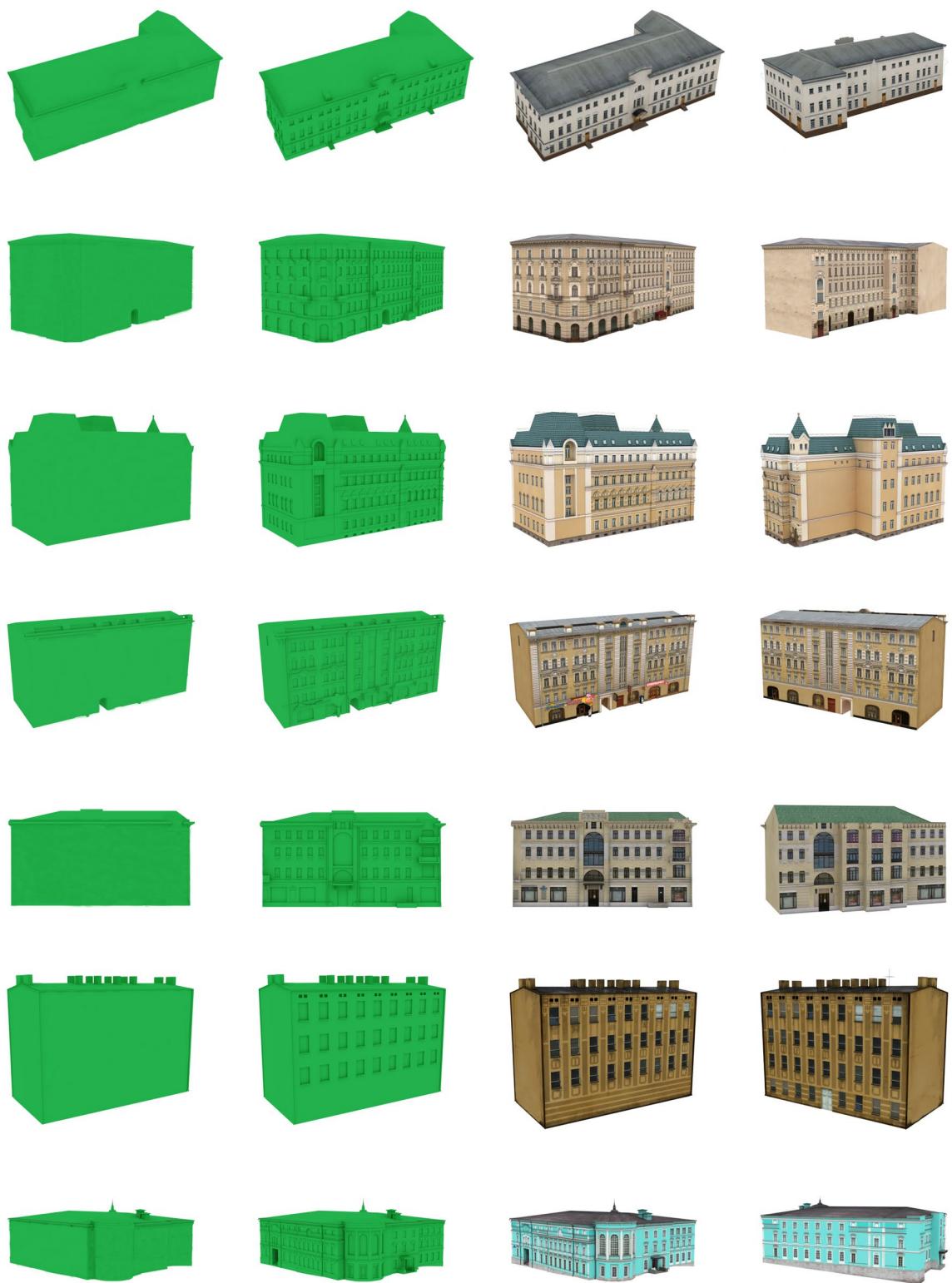


Fig. 6 Generated 3D Building



Fig. 7 Different view to the fabricated 3D building

5 Conclusion

3D building modelling is urgently required in the fields of urban planning, civil engineering, and smart city construction. Inspired by FrankenGAN (Kelly et al. 2018), in the paper an improved 3D building generation method is discussed, which embodies that a novel loss function with introduction of identity mapping loss and the perceptive loss is used for generating the data, the original BicycleGAN is replaced with StarGAN for efficiently implementing the multi-properties data generation.

A 3D building generation method based on improved FrankenGAN is proposed, it includes a novel loss function, multi-properties simultaneous mapping StarGAN, complex

property generation via GAN chains, together with combined GAN for geometry and texture coordination. The paper presents a complicated building generation method for handling multi-properties generation, complex property fabrication, and texture & geometry coordination. The method put forward presents a worthy referenced approach for complicated object generation based on state-of-the-art GAN.

The proposed 3D building generation method can fabricate realistic architecture, which holds the coordinated geometry and texture. Meanwhile the wall texture contains reasonable windows and appropriate layouts, and the building type also covers T-shapes, L-shapes and box-shapes.

The proposed method in the paper only explores the building, in future we will expand the given method for the generation of furniture, and plants.



(a) regular arrangement



(b) symmetrical arrangement



(c) arrangement within triangular area



(d) arrangement within irregular area

Fig. 8 3D Building scenario

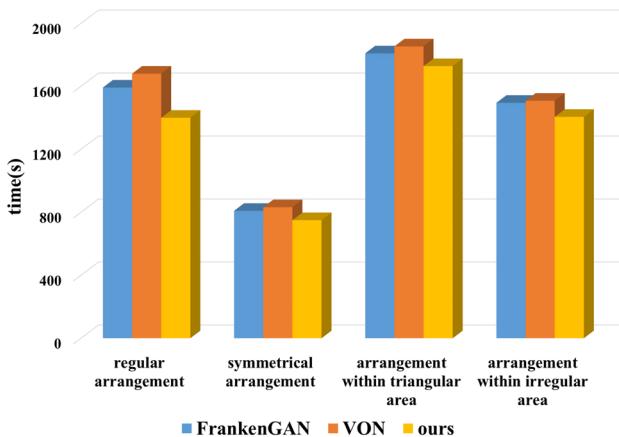


Fig. 9 Comparison of the building generation time

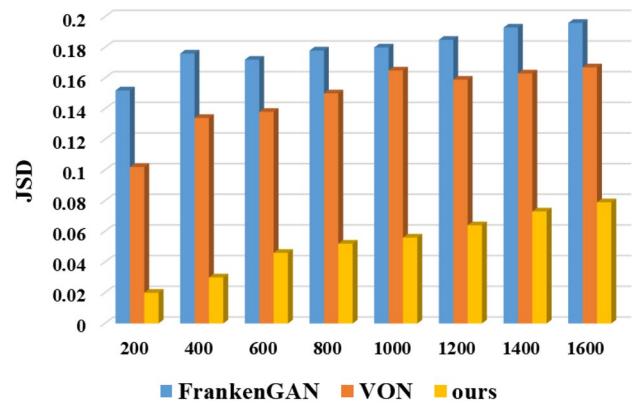


Fig. 10 JSD Comparison between the required data and the generated data

Table 1 Quantitative evaluation among methods

Method	JSD	MMD	MMD	COV	COV
	-CD	-EMD	-CD	-EMD	
G2L	0.0457	0.0034	0.0687	84.6	80.9
SAGNet	0.0478	0.0032	0.0647	77.2	76.4
FrankenGAN	0.0375	0.0029	0.0534	78.5	75.3
VON	0.0349	0.0031	0.0561	77.6	76.5
Ours	0.0297	0.0028	0.0528	88.3	85.7

Table 2 Ablation comparison on different kinds of loss

Category	L_{id}^G	L_{pl}^G	L_{id}^G and L_{pl}^G	Full
F-score	74.276	75.638	70.591	78.495
CD	0.675	0.633	0.732	0.589
EMD	1.583	1.552	1.656	1.398

Acknowledgements This work is supported by the National Natural Science Foundation of China under Grant No. 61672279.

References

- Arora S, Ge R, Liang Y, Ma T, Zhang Y (2017) Generalization and equilibrium in generative adversarial nets. In: Proceedings of Thirty-fourth International Conference on Machine Learning (ICML 2017), PMLR, pp 224–232, <https://doi.org/10.1145/3188745.3232194>
- Bao F, Schwarz M, Wonka P (2013) Procedural facade variations from a single layout. ACM Trans Graph 32(1):1–13. <https://doi.org/10.1145/2421636.2421644> Article No.8
- Bao J, Chen D, Wen F, Li H, Hua G (2017) Cvae-gan: fine-grained image generation through asymmetric training. In: Proceedings of 2017 IEEE International Conference on Computer Vision (ICCV'2017), IEEE press, Venice, Italy, pp 2745–2754, <https://doi.org/10.1109/ICCV.2017.299>
- Chaudhuri S, Kalogerakis E, Guibas L, Koltun V (2011) Probabilistic reasoning for assembly-based 3d modeling. ACM Trans Graph 30(4):1–10. <https://doi.org/10.1145/2010324.1964930> Article No.35
- Chen X, Li H, Fu CW, Zhang H, Cohen-Or D, Chen B (2018) 3d fabrication with universal building blocks and pyramidal shells. ACM Trans Graph 37(6):1–15. <https://doi.org/10.1145/3272127.3275033> Article No.189
- Choi Y, Choi M, Kim M, Ha JW, Kim S, Choo J (2018) Stargan: unified generative adversarial networks for multi-domain image-to-image translation. In: Proceedings of 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2018), IEEE press, Salt Lake City, UT, USA, pp 8789–8797, <https://doi.org/10.1109/cvpr.2018.00916>
- Dang M, Ceylan D, Neubert B, Pauly M (2014) Safe: structure-aware facade editing. Comput Graph Forum 33(2):83–93. <https://doi.org/10.1111/cgf.12313>
- Girdhar R, Fouhey DF, Rodriguez M, Gupta A (2016) Learning a predictable and generative vector representation for objects. In: Proceedings of European conference on computer vision (ECCV'2016), IEEE press, Amsterdam, pp 484–499, https://doi.org/10.1007/978-3-319-46466-4_29
- Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: Advances in Neural Information Processing Systems 27, Curran Associates, Inc., pp 2672–2680, <https://doi.org/10.5555/2969033.2969125>
- Hu R, Wen C, Kaick OV, Chen L, Lin D, Cohen-Or D, Huang H (2018a) Semantic object reconstruction via casual handheld scanning. ACM Trans Graph 37(6):1–12. <https://doi.org/10.1145/3272127.3275024> Article No.219
- Hu R, Yan Z, Zhan J, Kaick OV, Shamir A, Zhang H, Huang H (2018b) Predictive and generative neural networks for object functionality. ACM Trans Graph 37(4):1–13. <https://doi.org/10.1145/3197517.3201287> Article No.151
- Huang H, Kalogerakis E, Marlin B (2015) Analysis and synthesis of 3d shape families via deep-learned generative models of surfaces. Comput Graph 34(5):25–38. <https://doi.org/10.1111/cgf.12694>
- Ilik M, Musialski P, Auzinger T, Wimmer M (2015) Layer-based procedural design of facades. Comput Graph Forum 34(2):205–216. <https://doi.org/10.1111/cgf.12553> Article No.2
- Kelly T, Guerrero P, Steed A, Wonka P, Mitra NJ (2018) Frankengan: guided detail synthesis for building mass-models using style-synchronized gans. ACM Trans Graph 37(6):1–14. <https://doi.org/10.1145/3272127.3275065> Article No.216
- Kim JC, Chung K (2020) Neural-network based adaptive context prediction model for ambient intelligence. J Ambient Intell Humaniz Comput 11:1451–1458. <https://doi.org/10.1007/s12652-018-0972-3>
- LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444. <https://doi.org/10.1038/nature14539>
- Lin J, Cohen-Or D, Zhang H, Liang C, Sharf A, Deussen O, Chen B (2011) Structure-preserving retargeting of irregular 3d architecture. ACM Trans Graph 30(6):1–10. <https://doi.org/10.1145/2070781.2024217> Article No.183
- Lu H, Li Y, Chen M, Kim H, Serikawa S (2018) Brain intelligence: go beyond artificial intelligence. Mobile Netw Appl 23(2):368–375. <https://doi.org/10.1007/s11036-017-0932-8>
- Lu H, Wang D, Li Y, Li J, Li X, Kim H, Serikawa S, Humar I (2019) Comet: a cognitive ocean network. IEEE Wirel Commun 26(3):90–96. <https://doi.org/10.1109/mwc.2019.1800325>
- Nash C, Williams CKI (2017) The shape variational autoencoder: a deep generative model of part-segmented 3d objects. Comput Graph 36(5):1–12. <https://doi.org/10.1111/cgf.13240>
- Schwarz M, Muller P (2015) Advanced procedural modeling of architecture. ACM Trans Graph 34(4):1–12. <https://doi.org/10.1145/2766956> Article No.107
- Singhal S, Passricha V, Sharma P, Aggarwal RK (2019) Multi-level region-of-interest cnns for end to end speech recognition. J Ambient Intell Humaniz Comput 10:4615–4624. <https://doi.org/10.1007/s12652-018-1146-z>
- Uemura T, Nppi JJ, Lu H, Kim H, Tachibana R, Hironaka T, Yoshida H (2019) Ensemble 3d residual network (e3d-resnet) for reduction of false-positive polyp detections in ct colonography. In: Medical Imaging 2019: Computer-Aided Diagnosis, SPIE, CA, USA, pp 276–282, <https://doi.org/10.1117/12.2512173>
- Vanegas CA, Garcia-Dorado I, Aliaga DG, Benes B, Waddell P (2012) Inverse design of urban procedural models. ACM Trans Graph 31(6):1–11. <https://doi.org/10.1145/2366145.2366187> Article No.168
- Wang H, Schor N, Hu R, Huang H, Cohen-Or D, Huang H (2018) Global-to-local generative model for 3d shapes. ACM Trans Graph 37(6):1–10. <https://doi.org/10.1145/3272127.3275025> Article No.214
- Wen Z, Liu D, Liu X, Zhong L, Lv Y, Jia Y (2019) Deep learning based smart radar vision system for object recognition. J Ambient Intell

- Humaniz Comput 10:829–839. <https://doi.org/10.1007/s12652-018-0853-9>
- Wu J, Zhang C, Xue T, Freeman WT, Tenenbaum J (2016) Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In: Proceedings of 30th Conference on Neural Information Processing Systems (NIPS'2016), Curran Associates, Inc., Barcelona, Spain, pp 82–90, <https://doi.org/10.5555/3157096.3157106>
- Wu J, Wang Y, Xue T, Sun X, Freeman WT, Tenenbaum JB (2017) Marnet: 3d shape reconstruction via 2.5d sketches. In: Proceedings of 31th Conference on Neural Information Processing Systems (NIPS'2017), Curran Associates, Inc., Long beach, CA, USA, pp 540–550, <https://doi.org/10.5555/3294771.3294823>
- Wu W, Fan L, Liu L, Wonka P (2018) Miqp-based layout design for building interiors. Comput Graph Forum 37(2):511–521. <https://doi.org/10.1111/cgf.13380>
- Xu X, He L, Lu H, Gao L, Ji Y (2019a) Deep adversarial metric learning for cross-modal retrieval. World Wide Web pp 657–672, <https://doi.org/10.1007/s11280-018-0541-x>
- Xu X, Lu H, Song J, Yang Y, Shen HT, Li X (2019b) Ternary adversarial networks with self-supervision for zero-shot cross-modal retrieval. IEEE Trans Cybern pp 1–14, <https://doi.org/10.1109/tcyb.2019.2928180>
- Zhang H, Xu K, Jiang W, Lin J, Cohen-Or D, Chen B (2013) Layered analysis of irregular facades via symmetry maximization. ACM Trans Graph 32(4):1–13. <https://doi.org/10.1145/2461912.2461923> Article No.121
- Zhao W, Lu H, Wang D (2018) Multisensor image fusion and enhancement in spectral total variation domain. IEEE Trans Multimed 20(4):866–879. <https://doi.org/10.1109/tmm.2017.2760100>
- Zhu JY, Zhang R, Pathak D, Darrell T, Efros AA, Wang O, Shechtman E (2017) Toward multimodal image-to-image translation. In: Proceedings of 31th Conference on Neural Information Processing Systems (NIPS'2017), Curran Associates, Inc., Long beach, CA, USA, pp 465–476, <https://doi.org/10.5555/3294771.3294816>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.