Interactive generative rendering from iterative sketch

Xinwei Zhuang
Department of Architecture
University of California, Berkeley
Berkeley, CA
xinwei_zhuang@berkeley.edu

***Abstract*—Architects usually draw sketches in the early design phase to organize and elaborate their initial ideas. These sketches are usually abstract, and hard to interpret, but are useful for the final design decision. This paper provides a scheme for architects and designers to generate preliminary renderings in the early design stage. This study uses conditional generative adversarial networks (cGAN) as the frame, and introduces an updater network. With a simple sketch input, the network will generate a reasonable rendering by the single-image network, and then users are able to refine their sketch based on the generated images, and the network will update the renderings iteratively via the updater network. The dataset is collected from exclusively residential buildings, but the category can be expanded to other types of buildings. Results from single input generation and updated input generation, along with a interactive demo is presented.**

***Index Terms*—cGAN, computer aided design**

I. Introduction

In the early design phase, When a design is request, architects usually make preliminary sketches to visualise their ideas for future development. These sketches are usually abstract, and hard to interpret, but are useful for the final design decision. This study uses conditional generative adversarial networks (cGAN) as the frame, introduces an updater network, to provide an application that with a simple sketch input, it can generate a more developed design with rendering, and users can interactively develop their sketch.

The architectural design, like the urban planning, is a wicked problem. The traditional architectural design requires endless effort in trial and error. Artificial intelligence aided architectural and structural design has been intensively investigated in recent years thanks to the advent of artificial intelligence and big data. However, it encounters two major problems in the architectural field: the insufficient amount of usable data, and the divergent views on AI's creative design ability. This study proposes using generative adversarial network for data generation and generative design. Results will help to push the boundary of AI aided architectural design, that artificial intelligence can prove its ability of abstraction and be a creative design alternative.

The network used in this study is two conditional generative neural networks, the first takes single sketch with the real photo as input, and the second takes the result of the first network, overlaid by updated sketch, with real photo as input. This allows a iterative sketch process to generate rendering with complex sketches, and follows the normal workflow of a sketch process.

II. Related works

A. Generative design

The history of generative design can be traced back to 1900s. A lot of machine learning algorithms, such as generative algorithm, spectral clustering, are applied in architectural design field. However, within this field there are lots of debate on whether generative design eliminate human's effort.

AI aided tasks are mainly confined in the 4 Ds: dangerous, dull, dirty and dumb (Bekey et al., 2008). In these kinds of work, artificial intelligence does prove itself in not only practicability but efficiency. However, it meets diverse opinions when it comes to the question that whether AI can handle creative tasks. Some studies in music, art and computer vision have made some progress, using artificial intelligence to generate brand new products in fields of painting (AARON by Harold Cohen), composing (AIVA by Pierre Barreau), writing (NaNoGenMo by Darius Kazemi), etc. Recently, investigations were also made on architecture, such as improving the interactive design process (Martinez, 2016), and residential floor plan design and optimization (Zheng and Huang, 2018 [9]; Chaillou, 2019; Stanislas, 2019). Most of the studies focused on 2D generation and optimization of building footprints, plan design and ecological performance. Few studies exists on 3D generation until recently (Wu et al, 2016).

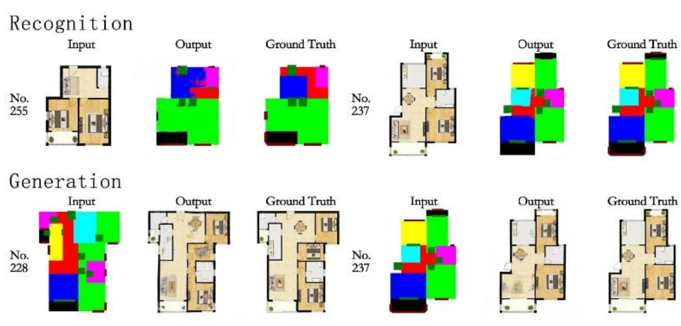


Fig. 1: Apartment floor plan: recognition and generation through generative adversarial network. Zheng and Huang, 2018. [9]

For architectural design, Zheng and Huang (2018) firstly applied Pix2pixHD to recognize and generate apartment floor plans. Experiments for recognition ability is achieved by assigning floor plan design into different color groups, and generation ability is by mapping color map back to floor design. The creative ability of AI is primarily proved, and an insight is given on understanding human design behaviors, such as catching similar characteristics among a group of data. Chaillou (2019) [10] made a step further on generative design through the whole design process. Artificial neural network's design ability from building footprint decision, to floor design, furniture distribution, and to room organization is demonstrated. Residential floor plan style is studied in 4 architectural styles (Baroque, Row House, Victorian Suburban and Manhattan Unit), and the generated floor plans do capture some characteristics of certain styles.

B. Generative model

GAN obtained enormous attention as soon as it was first proposed (Goodfellow, 2014) [8]. Compared with other artificial neural networks, The generative neural network is to train two networks, generator G and discriminator D with a min-max game. Where the discriminator learns to $D(x) = 1$ for real images and $D(x) = 0$ for generated images. Given a set of data, the generator does not calculate the data distribution and probability, but tries to imitate real data distribution without certain formulae; the discriminator tries to distinguish real data from the generated one. The aim is to achieve a Nash equilibrium, a global minimum of the minmax equation.

$$\min_{G} \max_{D}(E_{x\ p_{data}}[logD(x)] + E_{z\ p_z}[log(1-D)(G(z)))])$$

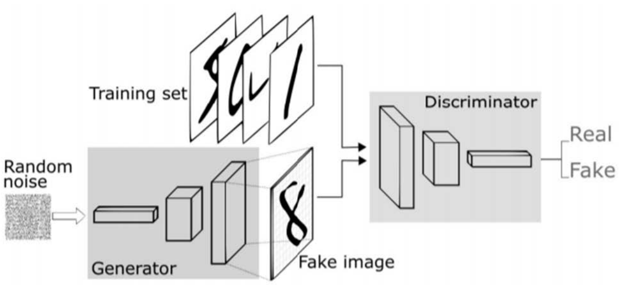


Fig. 2: Demonstration of typical GAN architecture. Radford et al. 2015. [11]

In 2018, Isola et al. proposed conditional generative adversarial network (cGAN) [2]. The conditional generative adversarial network is a subset of generative network. It takes conditional input data to feed both the generator and the deliminator and is capable of image to image translation. Delanoy et al. (2018) [1] proposed a sketch based modelling system that can take translate a sketch into 3D objects. They also introduced an updater in addition to the cGAN, so that user can rotate the generated model and update the object with another sketch to improve the generated mesh.

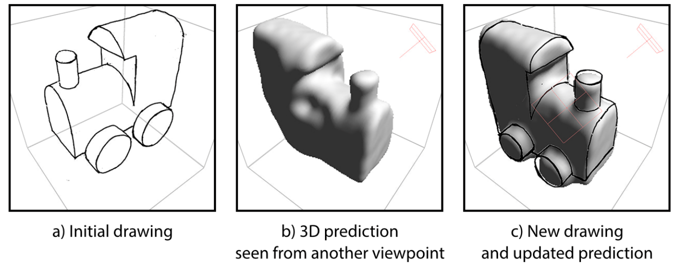


Fig. 3: sketch-based modeling system. Delanoy et al. 2018. [1]

III. Methodology

A. Data preparation

The data is collected from Archdaily. Total of 2740 projects is collected. For each project, multiple images is collected. The total size of the image set is 45776, which is scaled and center-croped to 256x256 pixel. Based on these images, similarity matrix is calculated. The ideal image is a building with a clean background, and interior photos are eliminated from the preprocessing. The images after passing the similarity check is shown as below, the majority of the images are building exterior.

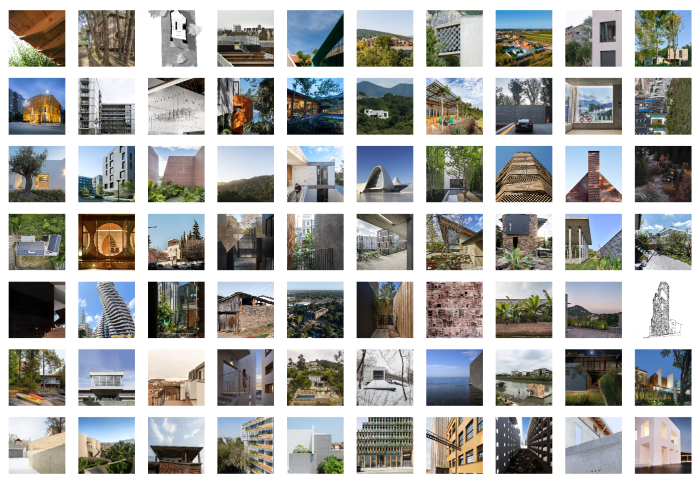


Fig. 4: The architecture of the neural networks.

Then, image contour is calculated. Several contour algorithm is tried. Firstly, the Harris corners is used to generate contour. However, the contour can have noise and do not provide highlight of the architecture itself. Different alpha ratio is used to find the optimal contour for input data. Because of the size of the image (224x224 pixel), the alpha is set to be 2. Similar researches uses image-space contour rendering approach (Delanoy et al. 2017) [1]. Though there is more recent contour generating algorithm such as suggestive contours, proposed by DeCarlo et al. (2003), the architectural sketch usually only with clean lines. And when sketching, the environmental factors can also be considered during the early design phase. Thus the simple Harris Corners can be used in this project. Further investments might includes how to eliminate the background contours. Standard deviation of the Gaussian filter is set to 2 for the first round input, and set to 1 for second round edge input with more detail. The contour image and the original photos are used as paired data which is then feed to the network. Sample data is shown as follows.

Fig. 5: Example of edge extraction. $\alpha = 2$ (left) for the first iteration training, $\alpha = 1$ (middle) for the sequent iterations training, and the real photo (right)

B. cGANs architecture

The network architecture developed in this project is in two parts, both are conditional generative neural networks. The first network takes edge image with $\alpha = 2$ and real photo as paired input. The output of the network is the first iteration of the sketch. This network is pretrained and all the data is processed by this pretrained network to produce intermediate data for the next step. Then the output from the first network is overlapped by a more detailed edge image with $\alpha = 1$, and paired with the same real photo as the input for the second network. Illustration of workflow is as follows.

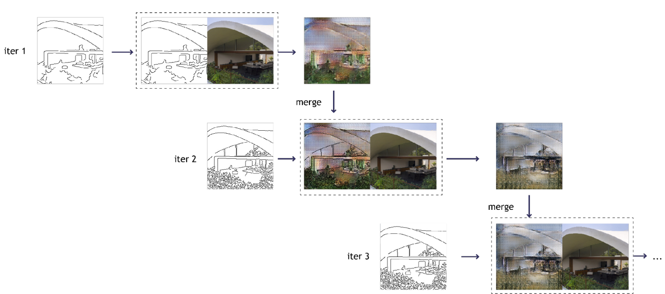


Fig. 6: The architecture of the neural networks.

The architecture of the two networks are U-net encoder-decoder. In both networks, there are 8 up sampling layers and 7 down sampling layers with kernel size 4 and strip size 2. For the down sampling layer, there is a batch normalization and leaky ReLu after each convolution layer, and for the up sampling layers, there is a transposed convolution layer followed by batch normalization, drop out and ReLu. The drop out rate is set to be 50%. The batch size is set to 64. The loss function for the generator is $AdversarialLoss + Lambda * L1Loss$, where lambda is set to 100 according to the recommended parameter value by Dumoulin and Visin (2016) [3]. The discriminator loss is $real_loss + generated_loss$, where the real loss is a sigmoid cross-entropy loss of the real images and an array of ones and the generated loss is a sigmoid cross-entropy loss of the generated images and an array of zeros.

For the up sampling layers, drop-out is used, recommended by Hinton et al. (2012). Firstly, the edge images with α = 2 paired with the original photos are trained 30,000 iteration. This intermediate is then used for the second round training for edge images with α = 1.

IV. Results

A. Single photo input

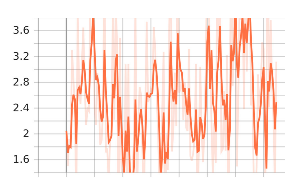
The generated photo from single input (with only a black and white edge diagram as input data) is as follows. The cGAN learned some features such as curved sketch is mostly interpreted as greenery, and straight lines are mostly the facade (in some preference with dark metal modern material). Some shape inside the contours can be interpreted as openings such as windows. The sky and the ground is separated with different colours, and the architecture itself shows a diversity in both colour, material and composition. Results are demonstrated below after 20,000 iteration of training with batch size 64.

Fig. 7: Output from single image training with edge $\alpha = 2$.

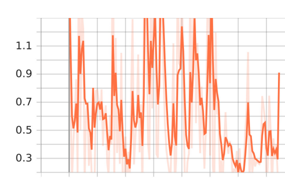
B. Iterative input

The training loss of generator and discriminator is shown as follows. The generator loss is high (2.5) while the discriminator drops below 0.4 after 140 steps.

The iterative process is achieved by inputting the output of the first network overlaid by the edge image with more



(a) generator loss 1



(b) discriminator loss

Fig. 8: Loss

details, such as the window frame and the roof depth. Results are demonstrated below after 10,000 iteration of training with batch size 64.

The result from the second iteration shows extinction in different components of the building. The roof and the window can be easily separated. The results still preserves the ability to separate background (a blue sky), the foreground (the greenery) and the main body (architecture). However, it worth noticing that the diversity in building material seems disappeared, all the architecture converges into a white facade with brown frames.

Fig. 9: Output from single image training with output from iteration 1 overlaid by edges $\alpha = 1$.

V. Discussion

By introducing the updater network, the resolution of the output image is enhanced, and the network can emphasized the edge and separate the blurry output from the single-image iteration.

From the failure cases, some of the rendering produced after the updater network is blurry in the foreground, and the details is lost. This might due to the overcomplexity edge, identifying that a more careful data preparation is required. Another observation is that due to the data imbalance, the majority of the output photos from the first iterations has a 'style' in brick and warm light, whereas the subsequent iterations shows a monotonous appearance with cold metal facade, even though they are trained with the same real photos.

VI. Summary and Future work

In summary, the proposed scheme is able to provide reasonable renderings from sketches, and the generated rendering can be updated if provided more detailed sketch within a second. However, due to the limitation of the scope of the data, generated rendering can present a monotonous modern mental appearance.

There are two ways to improve. The first is to make a larger dataset with more diverse building styles. The architectural style is different across building type, cultural difference, and various environment, it will be useful to make sub field categories for sketch generation, such as training only use residential housing data for residential sketch.

Fig. 10: Successful cases. Result from the single-input iteration (left) and multi-input iterations (right).

Fig. 11: Failure cases. Result from the single-input iteration (left) and multi-input iterations (right).

The second is to introduce multiple outputs from single sketch that user can choose the rendering he/she wants to continue, and use semantic segmentation to segment different components of a image, such as background (sky, greenery, water...), foreground (grass, people), and the architecture itself (facade, glazing, door, column...), and user can choose the area he/she is satisfied with and fix that area, and develop more on the remaining space. Another interesting direction worth investigation is to build a 3D generator from 2D sketch.

Acknowledgment

This project received tremendous help from Prof. Alyosha Efros and Prof. Angjoo Kanazawa, and their wonderful lecture Introduction to Computer Vision and Computational Photography, and their advice for developing this project.

This project is inspired by Delanoy et al. (2018) [1] and Isola et al. (2018) [2]. The code is modified from Isola et al. (2018) [2].

References

[1] Delanoy, J., Aubry, M., Isola, P., Efros, A. A., Bousseau, A. (2018). 3D Sketching using Multi-View Deep Volumetric Prediction. Proceedings of the ACM on Computer Graphics and Interactive Techniques, 1(1), 1–22. https://doi.org/10.1145/3203197

[2] Isola, P., Zhu, J.-Y., Zhou, T., Efros, A. A. (2018). Image-to-Image Translation with Conditional Adversarial Networks. ArXiv:1611.07004 [Cs]. http://arxiv.org/abs/1611.07004

[3] Dumoulin, V., Visin, F. (2016). A guide to convolution arithmetic for deep learning. ArXiv:1603.07285 [Cs, Stat]. http://arxiv.org/abs/1603.07285

[4] Eigen, D., Fergus, R. (2015). Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-Scale Convolutional Architecture. ArXiv:1411.4734 [Cs]. http://arxiv.org/abs/1411.4734

[5] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. ArXiv:1207.0580 [Cs]. http://arxiv.org/abs/1207.0580

[6] Mirza, M., Osindero, S. (2014). Conditional Generative Adversarial Nets. ArXiv:1411.1784 [Cs, Stat]. http://arxiv.org/abs/1411.1784

[7] Sangkloy, P., Burnell, N., Ham, C., Hays, J. (2016). The sketchy database: Learning to retrieve badly drawn bunnies. ACM Transactions on Graphics, 35(4), 119:1-119:12. https://doi.org/10.1145/2897824.2925954

[8] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y. (2014). Generative adversarial nets. In Advances in neural information processing systems (pp. 2672-2680).

[9] Huang, W. and Zheng, H. 2018. Architectural Drawings Recognition and Generation through Machine Learning. Mexico city, ACADIA.

[10] Chaillou, S. (2019). AI + Architecture, Towards a New Approach. Harvard University, 188.

[11] Radford, A., Metz, L., Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. (DCGAN)