GENERATIVE DESIGN OF URBAN FABRICS USING DEEP LEARNING

JINMO RHEE[1] and PEDRO VELOSO[2]
[1,2]*Computational Design Laboratory, CRAIDL, Carnegie Mellon University*
[1,2]*{jinmor|pveloso}@andrew.cmu.edu*

Abstract. This paper describes the Urban Structure Synthesizer (USS), a research prototype based on deep learning that generates diagrams of morphologically consistent urban fabrics from context-rich urban datasets. This work is part of a larger research on computational analysis of the relationship between urban context and morphology. USS relies on a data collection method that extracts GIS data and converts it to diagrams with context information (Rhee et al., 2019). The resulting dataset with context-rich diagrams is used to train a Wasserstein GAN (WGAN) model, which learns how to synthesize novel urban fabric diagrams with the morphological and contextual qualities present in the dataset. The model is also trained with a random vector in the input, which is later used to enable parametric control and variation for the urban fabric diagram. Finally, the resulting diagrams are translated to 3D geometric entities using computer vision techniques and geometric modeling. The diagrams generated by USS suggest that a learning-based method can be an alternative to methods that rely on experts to build rule sets or parametric models to grasp the morphological qualities of the urban fabric.

Keywords. Deep Learning; Urban Fabric; Generative Design; Artificial Intelligence; Urban Morphology.

1. Introduction

Urban fabric is a key concept in urban design that consists of the configuration of streets, parcels, lots, and buildings (Oliveira, 2016, p. 8). It operates in the intersection of urban and architectural scale and can reflect important aspects of social phenomena and cultural practices. Given the importance of the configuration of the urban fabric in its inhabitants' lives, understanding, categorizing, and designing them has been a long-lasting challenge for design researchers.

In the past decades, there have been several studies on the computational synthesis of urban fabrics using different computational methods, such as multi-agent system (Biao et al., 2008), diffusion-limited aggregation (Koenig, 2011), rule-based system (Pellitteri et al., 2010), L-systems (Parish & Müller, n.d.), etc. While these computational methods can generate good results, they

PROJECTIONS, Proceedings of the 26th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA) 2021, Volume 1, 31-40.

are limited to (1) modes of operation of existing generative models and (2) the designers' capacity to understand and encode the desired morphological qualities and contextual adaptations in principles, rules, and parameter calibration of the models.

In this research, we present Urban Structure Synthesizer (USS), a learning-based model to synthesize urban fabric diagrams that alleviate the reliance on knowledge explicitly embedded in generative models. Unlike rule-based or agent-based system, which depends on experts to create rules for generation, USS learns the morphological features of urban fabrics from data.

Firstly, we use a method to convert site data from GIS into diagrammatic images with contextual information introduced in the previous research (Rhee et al., 2019). After curating the site data with the desired patterns of urban forms and features, we use the resulting context-rich dataset to:

- train an analytical model that organizes the data into clusters with similar morphological types (Rhee et al., 2019).
- train a WGAN model to synthesize site-responsive diagrams of urban fabric based on the morphological qualities of the sites in the dataset (Figure 1).

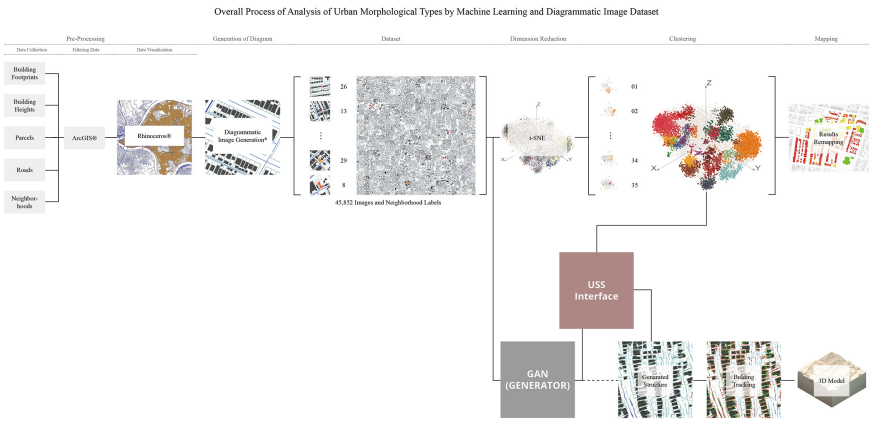

Figure 1. Two different processes: analysis (top) and generation (bottom) of urban fabrics using different machine learning models. USS interface connects the analytical and generative model.

2. Learning

2.1. DATA

USS employs a diagrammatic image dataset (DID), a data synthesis technique for raster images with a diagrammatic representation of two-dimensional building form and its neighboring urban contexts. Diagrammatic image representation in the dataset "offers two advantages in machine learning: low level of image noise

and support for custom selection of morphological information".(Rhee et al., 2019, p. 345).

Currently, the most popular way to create images of urban information is by using satellite and map images. Both methods can add unnecessary information or noise to the images. Therefore, a method such as image segmentation is typically employed to remove the noise of the representation for datasets based on satellite images. In contrast, diagrammatic image representation is synthesized from GIS information selected by the user, which preemptively removes or even excludes image noise from the dataset.

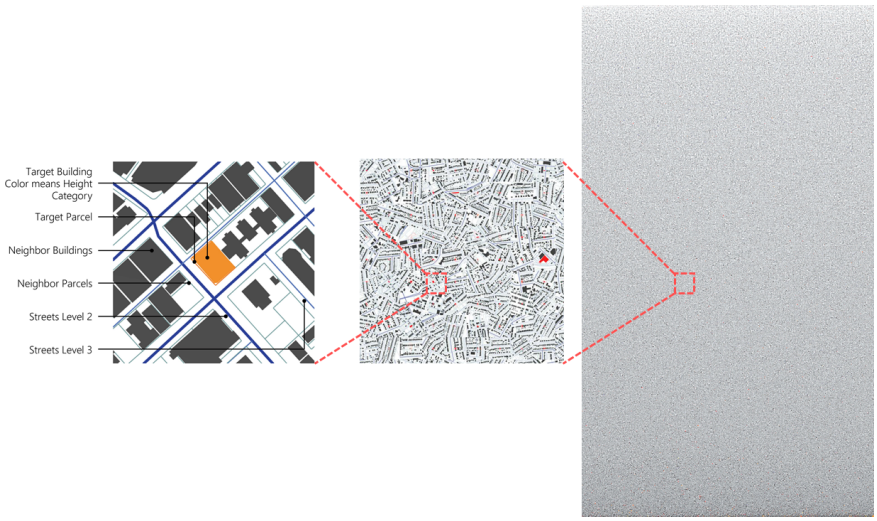

Figure 2. Composition of DID-PGH dataset in three different level.

Taking advantage of these qualities, we use DID to create a dataset of Pittsburgh, USA, which we will referreferred to as DID-PGH. DID-PGH comprises images with a target building placed on the center, and their neighboring building footprints, street network, and the parcel shapes around. The height of center building is represented through the color of the solid inside the polygon of the footprint. The more reddish color indicates that the building is lower, while the more yellowish color indicates that the building is higher. Each image is 512 x 512 pixels, and the dataset has a total of 45,852 images (see previous research for more information, Rhee et al., 2019).

2.2. GENERATIVE MODEL AND TRAIN

The selection of the generator for USS was based on an initial test of three different generative deep learning models for urban fabric synthesis: Variation Autoencoder (VAE), Generative Adversarial Networks (GAN), and Wasserstein Generative Adversarial Networks (WGAN). We compared the resulted images from each model after training all models with the context-rich dataset in the same training epochs. VAE resulted in the blurriest images. The images from GAN was

sharper than VAE's one, but it hardly captures the morphological features form the context-rich dataset. With more stable training, the images created by WGAN tend to be sharper than the images generated by the simple GAN.

USS specifically uses Wasserstein Generative Adversarial Networks - Gradient Penalty (WGAN-GP) (Gulrajani et al., 2017) as the model to generate new urban fabric diagrams based on the captured patterns of contextual and morphological features of the dataset. The overall structure of this model is similar to other Generative Adversarial Networks (GAN) model, except for some changes in the components of the optimization, such as the error function and the gradient penalty term. It has two networks: a generator and a critic. During training, the generator creates images conditioned on random input vectors, and the critic provides a value signal indicating how real each input image is. In the initial step of training, the generator synthesizes random images, and the critic provides random signals. The key to GAN lies in how we alternate the training of the two networks so that the generator becomes more adept at fooling the critic and the critic at identifying which observations are fake (Foster & Safari, 2019, p. 100).

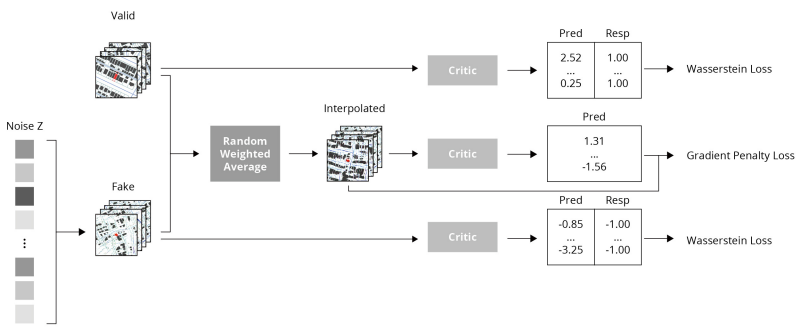

Figure 3. The structure of WGAN-GP model for training DID-PGH dataset.

In this research, the model generates a new urban fabric by learning patterns of contextual features of a given building. The generator of WGAN-GP creates an urban fabric image using a noise vector of size 100 as the input parameters.

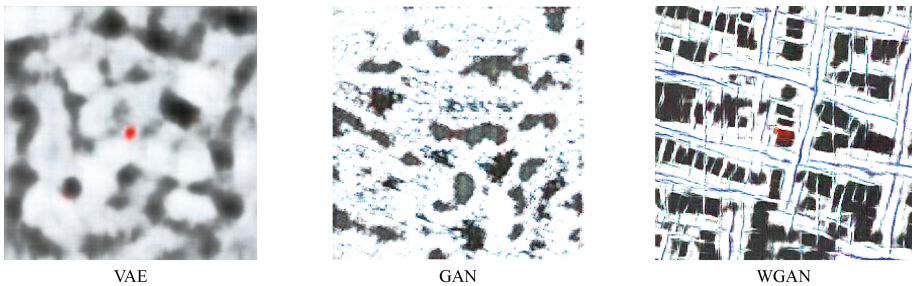

Figure 4. Comparison of images generated by different deep learning models.

We trained this model for about 10000 batches (15 epochs) with 64 for batch size. The model's optimizer is 'Adam' (Kingma & Ba, 2017), and the learning rate is 2.0E-4. The model has no dropout layers, and both discriminator and generator use Leaky Rectified Linear Unit (Leaky ReLU) as the activation function. This model was trained on a computer with the following specifications: 'Intel(R) Core (TM) i7-8700k @ 3.70GHz', 64GB memory, and two GTX-1080ti graphic cards. It took almost 37 hours to train the model. The losses reduced significantly until 3000 batches, , and after that, the changes were subtle.

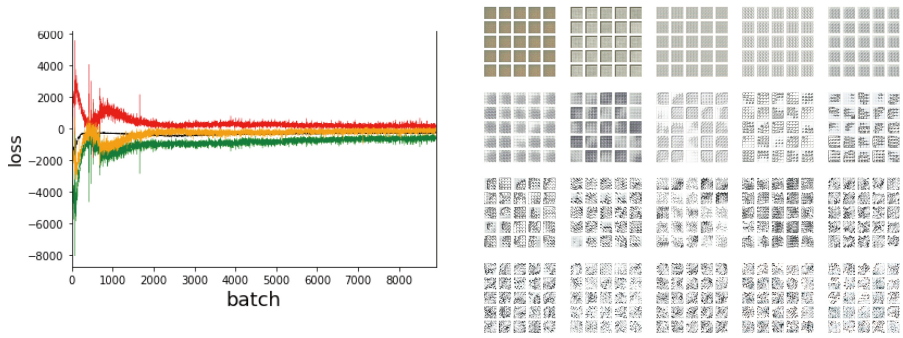

Figure 5. Changes of loss and image quality during the training process.

3. Design Implementation

3.1. USER INTERFACE

To generate urban fabric configurations, USS has an interface that relies on three steps: loading the trained model in the back-end, communication between front-end and back-end, and object tracking system.

When the USS is launched, the trained model is loaded in the back-end of the interface. Users can see an image of the urban fabric, sampled sliders, and buttons in the front-end fo the interface. The sliders represent the values of the noise Z vector for image synthesis. After training, each value of the vector can capture a design variation consistent with the design space of the original dataset. From the users' perspective, these sliders can be used to control morphological variations of the synthetic urban fabric in real-time. Although the original size of the vector is 100, we sampled 31 values that are represented as sliders in the interface. After manually changing the values of the sliders, the users can click on the 'GENERATE' button to pass the updated noise vector to the generator, which returns an updated synthetic image.

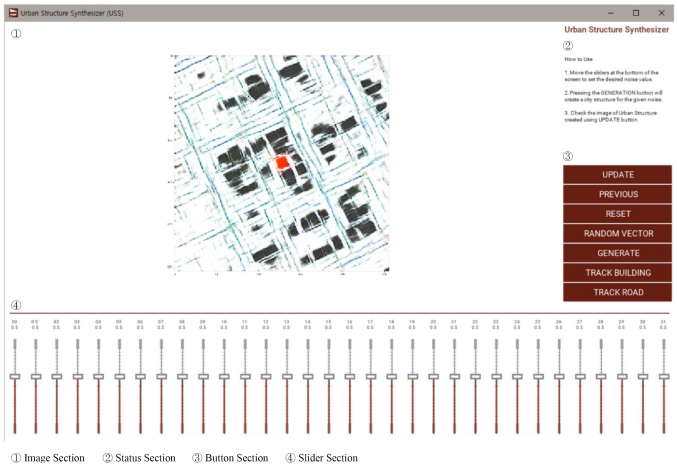

Figure 6. Interface of USS.

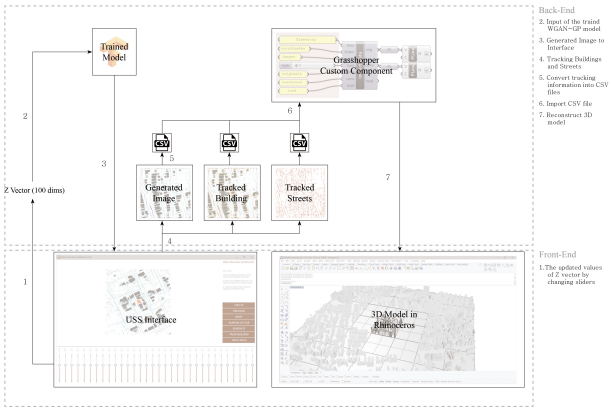

Figure 7. USS user interface system configuration.

To store the results, USS extracts the polylines of buildings and using OpenCV - a library for real-time computer vision. Each detected and tracked 2D information is reconstructed into 3D information using Rhinoceros®, the popular modeling tool in architectural design, and Grasshopper, a graphical algorithm editor tightly integrated with Rhinoceros.

Through custom components in Grasshopper, USS can communicate with Grasshopper in real-time and users can import the tracked object's outlines and its bounding boxes. After importing, users can see automatically reconstructed a 3D urban fabric based on the synthesized image from USS. USS tracks the color of the center building,converts the color value into height value according to the preset of color range, and saves this value so that users can import this height

information in Grasshopper to build a 3D model.

3.2. A DESIGN EXPERIMENT WITH USS

In order to test USS, we propose a speculative design exercise in Pittsburgh. The design uses USS to insert urban elements based on the morphological qualities of downtown Pittsburgh into the typical low-rise residential area of Shadyside. This exercise delves into the potential of how USS can be used for speculating about conflicts of scale and morphology.

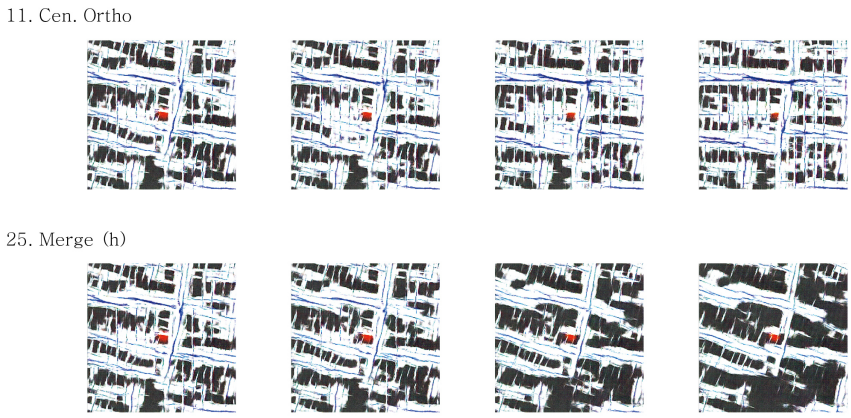

Figure 8. The changes in urban fabric images according to the 25th and 11th feature slider values.

Firstly, a urban fabric grid is applied to the site. The tiles of the grid have the same range and size as the image to be synthesized by USS. Then, part of the existing urban fabric is removed from several grid tiles to secure the place for new tiles.

Figure 9. Urban fabric design process using USS.

Secondly, the downtown fabric tiles are customized, produced, and displayed in real-time based on slider values of the interface. Each slider captures different functional features. For example, the 11th slider influences the orthogonal arrangement of buildings and streets. The 25th slider controls the merging of buildings in the bottom center of the tile.

Designing an urban tile with sliders in USS consists of two main steps: global and local setting. The global setting loads a synthesized image of the desired urban

fabric through presets. The slider values will be changed according to the presets. The local setting contains adjusting the preset sliders for fine-tuning the global setting.

While the user changes sliders, USS uses the analytical model to show the types of urban fabrics that the currently generated image is most similar to. The analytical model becomes a visual guide that indicates how users can modify the global and local settings to design the intended fabric. When users finished modifying the sliders and press to confirm button, USS places the reconstructed 3D model of the fabric on the site. Repeating these adjusting and confirming process eight times, we have generated eight downtown-like urban fabric tiles.

USS is used to transplant the synthesized downtown urban fabric to the residential area of Shadyside. The downtown tiles are placed without rigid borders between the existing and newly proposed fabrics. Therefore, the existing residential grid was replaced by the downtown business grid, which meant that the proposed fabrics had to be re-aligned with the residential fabrics. Through this design process, a downtown-like organization was created in the given area.

3.3. DESIGN EVALUATION

The speculative design presented in the previous section resulted in a heterogeneous fabric at the center of the urban residential site in terms of building sizes, shapes, and the axis of the grid with the gradual changes at the periphery. The buildings in the proposed urban fabric are larger and higher than the buildings in the previous fabric. The grid axes of the previous fabric are almost orthogonal and form rectangular grid patterns. On the other hand, the proposed fabric has diagonal axes that form triangular grid patterns. This resulting urban collage exposes the contrast between the different morphologies with an intricate geometric model generated in real-time.

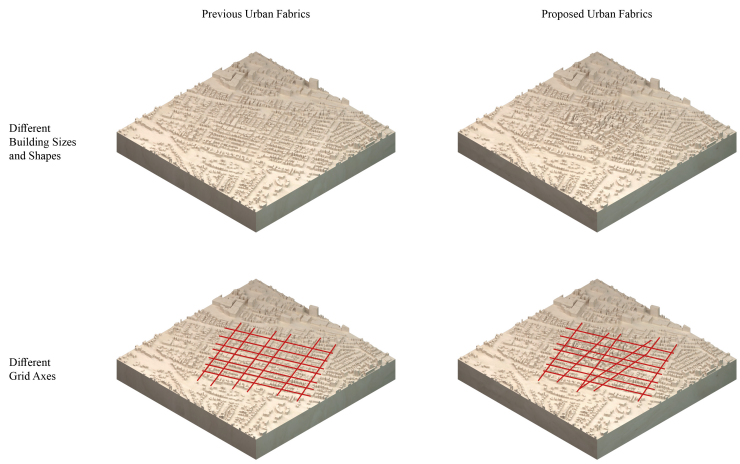

Figure 10. Comparison of the previous and proposed urban fabrics.

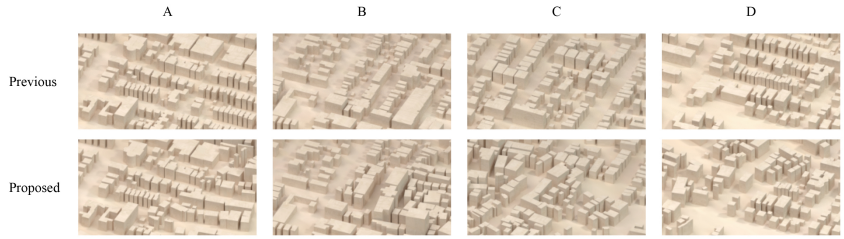

Figure 11. Different form of urban space in the previous and proposed urban fabrics.

Figure 11 shows the difference between the previous residential urban fabrics and the proposed business urban fabrics in detail. In cases A and B, the proposed fabric designed through USS has different size of buildings, clusters, and blocks, but keeps the existing street system and its axes. Cases C and D show that the streets in a new direction were created by utilizing characteristics of the existing fabrics. In the areas where different fabrics conflict, the periphery of the fabrics became indistinct by gradually changing the size of the buildings and directions of streets.

Our experiment shows the potential of USS to support a new urban design approach based on the investigation of urban fabric designs that are morphologically consistent. For example, like the experiment above, designers could create another downtown variation with multi-core structure (instead of uni-core) by preserving a the global downtown setting and exploring local solutions with the slides. Instead of explicitly modeling streets, building blocks, urban amenities, and public space, urban designers or planners can interact with real-time variations of fabric morphology on the site for design ideation.

4. Conclusion

USS illustrates the potential of using deep learning model in the generation of urban fabric, which enables designers not only to get access to the statistical features of datasets but also to produce novel designs from these features.

Through deep learning, it was possible to design a new urban fabric by extracting complex patterns from the existing fabrics and synthesizing the patterns according to a dataset curated by the designer. The deep learning-based design system is an alternative to generative systems that require human expertise to access and set generative rules and parameters. Learning-based systems can handle larger amounts of and more complex features than hand-craft rule-based and parametric systems. For example, in the design example of this study, the user did not define the rules of how to generate and position the new urban fabric. The deep learning model synthesized a new fabric by learning the pattern of the numerous existing fabrics. The learned function contains transformations that are potentially more complex and adaptive than rules explicitly defined by experts. Models based on more intricate rules can reveal design patterns and knowledge embedded in the built environment and, therefore, can support designers to devise

new design realms.

One of the main limitations of USS is the lack of control over the results. By relying on a random vector, the control of the variations of the result is not easy to grasp by the users. In order to extend USS to real-design applications, it is necessary to explore useful and reliable human-computer interactions and control mechanisms. There are some interesting alternatives from the generative modeling perspective, such as using paired datasets for a conditional generation. Considering that the dataset used in USS is already pre-processed, it is straightforward to derive custom diagrammatic controls such as the axes, masses (Zheng, 2018), connections in the border, etc. From the human-computer interaction, it is necessary to establish systematic studies with the users to identify adequate interaction modes with design variations.

References

Biao, L., Rong, L., Kai, X., Chang, L. and Qin, G.: 2008, A GENERATIVE TOOL BASE ON MULTI-AGENT SYSTEM, *Proceedings of the CAADRIA 2008*.

Foster, D.: 2019, *Generative Deep Learning : Teaching Machines to Paint, Write, Compose, and Play*, O'Reilly Media.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V. and Courville, A.C.: 2017, Improved Training of Wasserstein GANs, *Advances in Neural Information Processing Systems 30*, 5767–5777.

Kingma, D.P. and Ba, J.: 2017, Adam: A Method for Stochastic Optimization, *ArXiv:1412.6980*.

Koenig, R.: 2011, Generating urban structures: A method for urban planning supported by multi-agent systems and cellular automata, *Przestrzeń i Forma (space & FORM)*, **16**, 353-376.

Oliveira, V.: 2016, *Urban Morphology: An Introduction to the Study of the Physical Form of Cities*, Springer International Publishing.

Parish, Y.I.H. and Müller, P.: 2001, Procedural Modeling of Cities, *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, New York, 301–308.

Pellitteri, G., Lattuca, R. and Conti, G.: 2010, A Generative Design System to Interactively Explore Different Urban Scenarios, *Proceedings of the eCAADe2010*.

Rhee, J., Cardoso Llach, D. and Krishnamurti, R.: 2019, Context-rich Urban Analysis Using Machine Learning: A case study in Pittsburgh, PA, *Proceedings of the 37th eCAADe and 23rd SIGraDi*.

Zheng, H.: 2018, Drawing with Bots: Human-computer Collaborative Drawing Experiments, *Proceedings of CAADRIA 2018*.