

INTEGRATING BUILDING FOOTPRINT PREDICTION AND BUILDING MASSING

An experiment in Pittsburgh

JINMO RHEE¹, PEDRO VELOSO² and
RAMESH KRISHNAMURTI³
^{1,2,3}*Carnegie Mellon University*
^{1,2,3}*{jinmor|pveloso|ramesh}@andrew.cmu.edu*

Abstract. We present a novel method for generating building geometry using deep learning techniques based on contextual geometry in urban context and explore its potential to support building massing. For contextual geometry, we opted to investigate the building footprint, a main interface between urban and architectural forms. For training, we collected GIS data of building footprints and geometries of parcels from Pittsburgh and created a large dataset of Diagrammatic Image Dataset (DID). We employed a modified version of a VGG neural network to model the relationship between (c) a diagrammatic image of a building parcel and context without the footprint, and (q) a quadrilateral representing the original footprint. The option for simple geometrical output enables direct integration with custom design workflows because it obviates image processing and increases training speed. After training the neural network with a curated dataset, we explore a generative workflow for building massing that integrates contextual and programmatic data. As trained model can suggest a contextual boundary for a new site, we used Massigner (Rhee and Chung 2019) to recommend massing alternatives based on the subtraction of voids inside the contextual boundary that satisfy design constraints and programmatic requirements. This new method suggests the potential that learning-based method can be an alternative of rule-based design methods to grasp the complex relationships between design elements.

Keywords. Deep Learning; Prediction; Building Footprint; Massing; Generative Design.

1. Introduction

Urban context is a fundamental aspect of architectural design and has become more crucial in urban architecture in that it contains complex relationships of various urban elements. It has been explicitly used as the source for form generation in different architectural movements, such as traditional design, critical regionalism or even in contemporary practices based on diagrams. However, the integration of urban context information and design synthesis is still secondary

in generative design research. For instance, in space planning, CAAD has traditionally employed search and optimization algorithms to explore design alternatives based on internal building requirements, such as areas and adjacencies (Mitchell 1977; Liggett 2000). Building parcels and site geometry are generally treated as constraints for design generation.

Alternatively, agent and rule-based systems have been used to describe the generative logic of a certain design style and also the morphological qualities of existing buildings. Examples that explore the urban context include shape grammars (Ena 2018), L-systems (Parish and Müller 2001), custom urban systems (Hillier and Hanson 1984) and agent-based models such as cellular automata and diffusion-limited aggregation (Koenig 2011). Such systems tend to be harder to integrate with design synthesis because they require an expert to set the design rules or to tune the parameters of existing models.

With current accessibility to abundant urban data, novel approaches can be used to bridge this gap between contextual information and generative logic, without recourse to an expert. For example, machine learning addresses models that can improve in tasks such as regression, classification, clustering, dimensionality reduction and even generation based on data. With deep neural networks, machine learning can address large datasets and provide useful models for design synthesis.

In order to explore the potential of deep learning for space planning, we developed an experiment to capture geometric information from an urban context (represented by a dataset) and translated this into the design of a building on a new site. More specifically, we investigated the analysis and prediction of building footprint, one of the main interfaces between urban and architectural forms. This problem has been addressed in recent research, for example, Chaillou (2019), who trained a GAN on a database of Boston's buildings to generate an image of a building footprint based on the image of a parcel. He further developed a pipeline to partition rooms, define openings and furnish the spaces.

In contrast to Chaillou who creates a GAN-based pipeline from parcel to interior spaces, we focus on the relationship between a larger urban context and the building footprint, with a simpler model. Figure 1 shows the overall process of predicting and generating building footprints using a deep learning model on Pittsburgh in Pennsylvania. We collected Geographic Information System (GIS) data on building footprints and parcels in Pittsburgh and created a large custom dataset that synthesizes the morphological relationships between the target building footprint and its neighbor conditions as contextual information using a diagrammatic representation, Diagrammatic Image Dataset (DID). Next, we formulated the problem as a supervised learning task (regression). A modified version of VGG neural network models the relationship between (c) the diagrammatic image of a building parcel and context without footprint, and (q) a quadrilateral representing the original footprint. The option to use a compact geometric representation for the output simplified the training process, precluding the need for further image processing steps, and provided the required input format for Massigner (Rhee, Cardoso Llach, and Krishnamurti 2019), which is used for the integration between contextual information and architectural massing. The

basic idea is that contextual information is not used to define a final building volume, but as a guide for further generative exploration based on the internal requirements of the building.

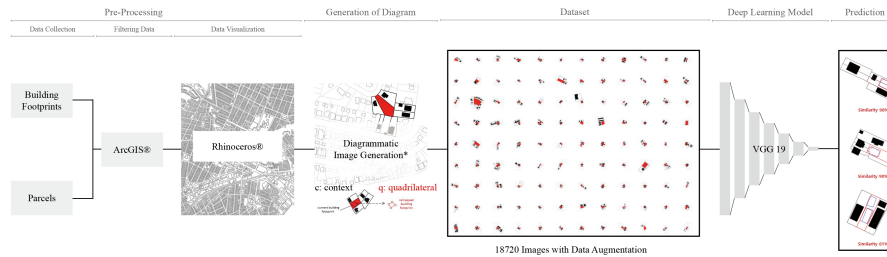


Figure 1. Overall Process of Generating Building Footprints Using Deep Learning and Diagrammatic Image Dataset of Building Occupancy in Pittsburgh, PA.

2. Data, Model, and Learning

2.1. OCCUPANCY MODEL

The diagrammatic image dataset (DID, provisionally patented, developed by Pedro Veloso and Jinmo Rhee) contains images with fixed colors and graphical elements to represent an urban context. The advantage of this type of representation is that designers can choose which aspects of the context to emphasize, thereby reducing the amount of noise from the original data. The image size is 512×512 (px). To get a proper range of contextual information to be included in a diagrammatic image, owing to the variety in the shapes of the buildings, the square root (14m) of the average area of all building footprints ($202 \square$) was set as the average radius for a target building. Assuming that three neighboring buildings are included on one side of the target building, half of the range of context information is set as 98m and eventually one side of the image range is set as 196m (Rhee, Cardoso Llach, and Krishnamurti 2019).

Based on the DID format, we developed an ‘Occupancy Model’ (see Figure 2), which shows the morphological relationship between the target building and its neighbor conditions. The occupancy model approximates the target building to a quadrilateral, stores its normalized coordinates as a vector, and contains the parcel shape of the target parcel, footprints of neighboring buildings and parcels as a single image (see Figure 3). At the center of a diagrammatic image with an occupancy model, there is an empty parcel filled with a solid color which indicates target parcel information. Around the target parcel, there are geometries representing information on adjacent buildings and parcels. The footprint information for the target building is a vector with the coordinates of the quadrilateral $[x_0, y_0, x_1, y_1, x_2, y_2, x_3, y_3]$, tagged to the diagrammatic images as the label.

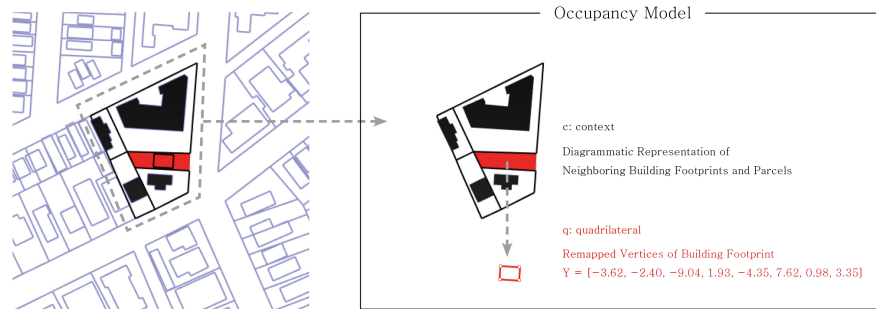


Figure 2. Concept of Occupancy Model.



Figure 3. Using the Context image (c) to Approximate the Quadrilateral of the Target Building Footprint (q).

2.2. GENERATION FOR DID FOR SHADYSIDE (PITTSBURGH, PENNSYLVANIA, USA)

We used the GIS data provided by Allegheny County, Pennsylvania, for information on building footprints and geometries of parcels: ‘2017 Allegheny County - Building Footprints’ and ‘2017 Allegheny County - Parcels’. After importing .shp files in ArcGIS, we set an area covering about 1.5km around Pittsburgh at an intersection near where several neighborhoods meet: Shadyside, East Liberty, Friendship, Bloomfield, Garfield, Highland Park History District, and Larimer. This site includes 7,598 buildings and 8,459 parcels. After converting building footprints and parcel geometries into .dwg files, Rhinoceros and Grasshopper were used to import these files and generate the DID for the occupancy model. For diagrammatic images, the process requires setting a diagram drawing style using line weights and colors for the geometry. The style can be customized. In this research, target parcels are colored solid red, neighboring buildings colored solid black, and neighboring parcels are represented by 2px wide black lines.

The geometrical information for the footprint of a target building is the label for its diagrammatic image. Each footprint is represented by a vector with the coordinates of its vertices. As the vector size depends on the number of vertices of the footprint, the size of the vector representing the original footprint varies on its shape (see Figure 2). The learning process requires a standardized vector format of a constant length. Figure 3 illustrates this need for a constant length using geometry approximation. The top polyline has four points, which can be represented by a vector of length 8. Another, the bottom polyline has six points, which can be represented by a vector of length 12. The first vector can obviously be represented by a vector of length 12 with four trailing 0's. However, should the footprint contain much more points represented by say a vector of length 100, the data becomes highly distorted (e.g., there are 92 trailing 0's). Therefore, approximating the geometry to a quadrilateral gives a vector that has constant length, and this contributes to more successful learning results.

To approximate the geometry to a quadrilateral, we tested seven different algorithms: the four longest distance, largest area, largest overlapping area, most similar to a rectangle-shape, most similar variance, smallest variance, and a Delaunay triangulation. We chose the Delaunay triangulation because it creates a quadrilateral successfully even in the case of concave geometry (see Figure 4).

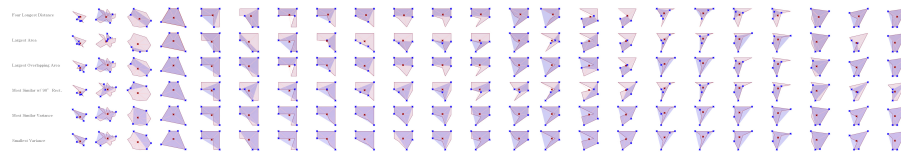


Figure 4. Seven Different Methods to Approximate Footprints into Quadrangles.

In order to validate the dataset for the occupancy model, we checked several conditions:

We first filtered empty target parcel cases. If the target parcel is empty, there is no target building footprint information to be converted to a label. In the same way, cases where there are no neighboring buildings near a target building were excluded. Moreover, we checked the collision cases between geometries: target parcel and neighbor buildings, target parcel and target buildings, target buildings and the window geometry, target parcel and the window geometry, neighbor buildings and the window geometry, and neighbor parcels and the window geometry. If there are any collisions, we excluded the case from the dataset. Lastly, cases where the relative size of the target building (the area of target building / the area of the target parcel $\leq e$, $e \approx 0.15$) is very small, such as temporary storage in yard, were also excluded.

After filtering invalid cases, the total number of the diagrammatic images is 2,080. This number of images is not enough to train the deep learning model for a regression problem and may occur overfitting. Therefore, in order to improve learning accuracy and reduce overfitting, we augmented the dataset by rotating the sample images, since we can assume that the relation between context and

footprint in Pittsburgh is weakly correlated with orientation. The original building images and labels are rotated 3 degrees apart 8 times counterclockwise. In total, 18,720 images derived from the 2,080 original images by this image augmentation method. Figure 5 shows the part of DID for Shadyside.

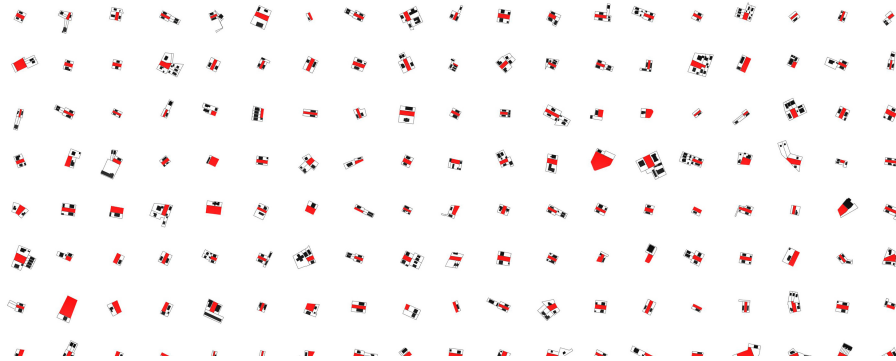


Figure 5. Part of DID for Shadyside.

2.3. MODEL AND LEARNING

Visual Geometry Group 19 (VGG 19, Simonyan and Zisserman 2014), a deep-neural network, was implemented to train the two-dimensional geometrical relationship between a building and its adjacent buildings along with their parcels for predicting building footprints. Considering that a parcel includes a house building and a garage building, the size of its y-value is 16 representing two quadrilaterals. Each quadrilateral has 4 points and each point has 2 coordinates. Therefore, we tweaked VGG19 to receive y-values of the dataset with tensor shape (18720, 16). Also, we added two dropout layers to prevent overfitting cases in learning. We compiled the model with an SGD (Stochastic Gradient Descent) optimizer, MSE (Mean Squared Error) loss function, and 0.00005 learning rate. Batch size is 64 for 800 epochs for learning. 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 41 hours to train the data. With this trained model, un-figured images were given to predict the building footprints by considering the given surrounding buildings and parcels.

3. Results and Design Implementation by Building Massing

3.1. LEARNING RESULT AND PREDICTION

In final learning, the model shows a maximum of 94.47% training accuracy and 93.14% validation accuracy (see Figure 6). We remap the predicted footprints on the target parcel and compared the similarity (%) with the original target building by considering the overlapping area and difference of vertices positions. Similarity is expressed through the average of the differences between an actual point (x, y) and predicted point $(\bar{x}, \bar{y})$ (1). The best and worst predicted footprints

had respectively 98% and 81% similarity with the cases with one original target building. However, when there is more than one target building in a parcel, it shows only one building footprint with 42% similarity (see Figure 7).

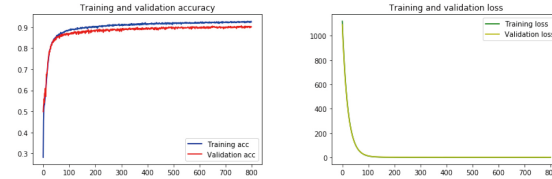


Figure 6. Learning Results, Accuracy and Loss Accuracy and Loss of Training and Validation Dataset.

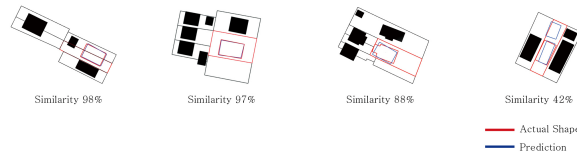


Figure 7. Prediction Result and Similarity to Actual Shapes.

$$S(Similarity) = \left(1 - \frac{1}{n} \sum_{i=0}^n \sqrt{(x_i - \bar{x}_i)^2 + (y_i - \bar{y}_i)^2} \right) \cdot 100, n = 3 \quad (1)$$

By visualizing the filters and the feature map of the trained model, we tracked how and what the model learned from the dataset. A filter-applied feature map to input images by layer is one way of visualizing the convolutional neural network. Generally, a convolutional neural network is assumed to be a ‘black box’ and it is hard to provide a reason for a specific decision. However, this visualization can help users of the network have a level of insight and understanding of the internal process of convolutional neural networks.

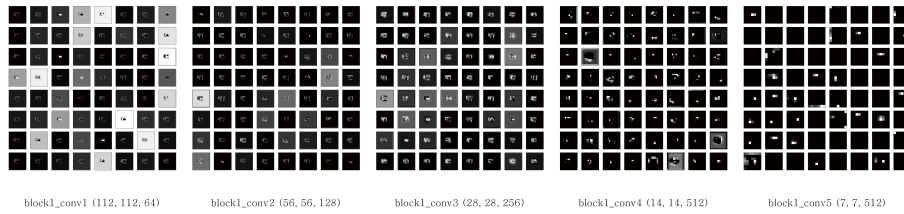


Figure 8. The Result of Filter-Applied Feature Map to a Sample Image by Each Convolutional Max-Pooling Layer.

Figure 8 is the result of filter-applied feature map to a sample image by each convolutional max-pooling layer. The brighter pixel indicates larger weights, and the darker pixel indicates lighter weights. The feature map in the first convolutional max-pooling layer of VGG19 shows the almost same details as the original images. The deeper the convolution layer, the more condensed the information shows. The feature map in the last convolutional max-pooling layer of VGG19 illustrates the pixel-scale white squares. It can mean the model abstracted well the feature of the dataset and trained well to grasp the generalized pattern of the given dataset.

3.2. DESIGN IMPLEMENTATION BY BUILDING MASSING

Prediction of quadrilateral can be decoded as two different ways for building massing: parametric and morphing. (see Figure 9). The parametric approach reconstructs the quadrilateral to the polyline shape by specifying the parameter t on one side of a sloped quadrilateral. If the quadrilateral projections are accurate enough and the original building footprint has parallel walls, its shape can be exactly reconstructed. The middle of the Figure 9 shows the various example masses from parametric decoding method. This method allows architects to grasp the sense of scale, size, and composition of existing mass. If the site does not have existing buildings or their shape is unknown, this decoding method can be used as a restoration tool by providing an inferred shape domain based on their contexts.

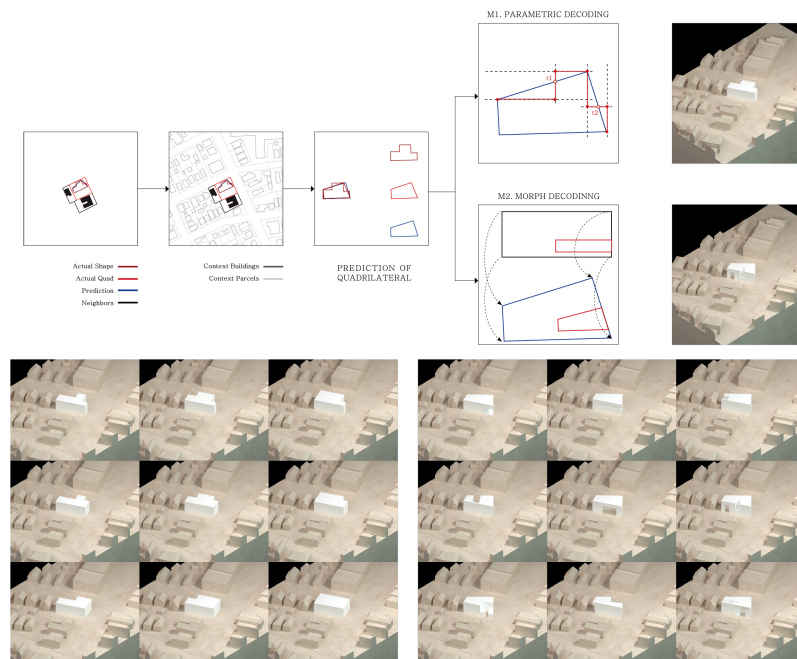


Figure 9. Two Different Methods to Decode Footprint Prediction (Top), Example of Parametric Decoding (Bottom-Left), and Example of Morphing Decoding (Bottom-Right).

The morphing approach is to translate this prediction as a new shape of building footprint or a maximum envelope. In this case, the prediction plays role to provide the size and location of the building within a given site parcel. Due to restriction of building footprints to quadrilaterals, box morphing method can be applied to the prediction shapes for generating new massing.

We used a genetic algorithm-based massing automation tool for housing (Rhee and Chung 2019) to generate an archetype of morphing shape. This tool subtracts voids from a maximum volume and maximizes the usage of within a given conditions, such as building regulations or codes. After generating the optimized volumes from the automation tool, the site volume can be generated by extruding the prediction of quadrilateralized building footprints. Then the optimized volume is morphed to the site volume. The right images of the Figure 9 show the examples of various morphed shapes.

4. Conclusion

In this paper we presented an initial experiment of a novel application of deep learning where we use a simple learning model and geometrical representations to integrate the contextual information with design synthesis. It successfully illustrates how generative systems can extrapolate the dependency of computational synthesis on internal building factors, such as spatial adjacencies, opening location, heat radiation optimization, to incorporate external qualities that are barely captured with the metrics and requirements.

Additionally, in contrast to the conventional hypothetical-deductive logic of the conventional CAAD methods, our approach promotes an alternative inductive approach - i.e. it supports the generalization of the knowledge acquired from data to novel cases with a function approximator (see Cardon, Cointet, and Mazières 2018 for this distinction). Our model learns a complex function that maps the relation between a certain notion of context (in our case, a diagram of the urban site) and the desired footprint based on a dataset.

In contrast to the direct optimization of a parametric model, our approach enables not only the reconstruction of cases from the dataset but also the generalization of the synthesis for previously unknown sites. Unlike rule-based systems, it does not require an expert to create a grammar or to tune a certain model. Rather than generating a few deterministic rules based on the accessibility or bias of the information of certain experts, patterns from existing data are employed to discover the generative rules. The designer does not have to be an expert in shape grammars or urban morphology. She only must curate a certain dataset representing the context, so the model can learn the desired relationship between context and form.

Finally, thorough and continuous research in various research aspects is still required to understand the broader and deeper applicability of contextual learning in generative design, such as space planning and building massing. Some of the open questions are: How to curate a design dataset?, What aspects of the context can be embedded in DID?, What types of geometry can be learned with simple regression models?, How to incorporate other representations in our method to

address other aspects of the context, such as in geometric learning?

References

- Cardon, D., Jean-Philippe, C. and Antoine, M.: 2018, Neurons Spike Back: The Invention of Inductive Machines and the Artificial Intelligence Controversy, *Réseaux, Machines Prédictives*, **5**(211), 173-220.
- Chaillou, S.: 2019, “Architecture & Style.” Medium.” . Available from <<https://towardsdatascience.com/architecture-style-ded3a2c3998f>> (accessed 30th June 2019).
- Ena, V.: 2018, De-Coding Rio de Janeiro’s Favelas: Shape Grammar Application as a Contribution to the Debate over the Regularisation of Favelas. The Case of Parque Royal., *Computing for a Better Tomorrow - Proceedings of the 36th ECAADe Conference*, Lodz, **2**:429–438.
- Hillier, B. and Julienne, H.: 1984, *The Social Logic of Space*, Cambridge Core.
- 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.
- Liggett, R.S.: 2000, Automated Facilities Layout: Past, Present and Future, *Automation in Construction*, **9**, 197-215.
- Mitchell, W.J.: 1977, *Computer-Aided Architectural Design*, Van Nostrand Reinhold Company.
- Parish, Y.I.H. and Pascal, M.: 2001, Procedural Modeling of Cities, *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, New York, 301–308.
- Rhee, J. and Chung, J.: 2019, A Study of Automation of Housing Design Method Using Artificial Intelligence, *Annual Conference in Architectural Institute of Korea*, Daejeon, **39**:181–84.
- 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 Conference*, Porto, **3**:343–52.
- Simonyan, K. and Andrew, Z.: 2015, Very Deep Convolutional Networks for Large-Scale Image Recognition, *International Conference on Learning Representations 2015*.