Generation of Building Placements in OpenStreetMap

Eduardo Hauck[1], Aber Mitsuteru[1], and Claus Aranha[1]

University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki 305-8577, Japan

Abstract. The OpenStreetMap project provides open-access geographic data that can be used for many applications. However, due to its collaborative nature, missing building footprint information is a common problem in several areas. In this work, we look at the problem of how to suggest virtual building placement to complete the geographic information missing from urban areas in OpenStreetMap. We propose a system for the generation of synthetic data with two main mechanisms. First, the system employs an evolutionary search with the MAP-Elites algorithm to generate a diverse range of building placements, grouping the solutions according to their features. Second, the building placements with the highest fitness in each feature group are tested against a convolutional neural network classifier that was trained to identify map images of a user-selected city (against other cities). The placement that receives the best rank by the neural network is selected as the output of the system. Our experiments show that the system generates believable building placements for locations with partially missing building footprint, although the evolutionary search is not able to find candidates in the whole range of selected features. Used in combination with simulation models, the system can become a valuable tool for policy evaluation in urban planning.

Keywords: Urban Modeling · Evolutionary Algorithm · Procedural Content Generation · Building Placements.

1 Introduction

The use of real geographic and population data has become a common trend in the study of urban areas to understand and predict urban phenomena. From agent-based simulation, where social systems are represented and simulated to observe emergent phenomena, to urban test-beds, where a replicable urban area is generated for testing specific applications such as self-driving cars, the use of geographic and population data has proven to be beneficial to urban-related research. Understanding the underlying properties of a city can be fundamental to guide the policies and decisions made by local governments, having a positive effect on its citizens' lives.

A recurring problem faced by researchers in these areas is that of collecting data. The traditional way of collecting geographic and population data was to

(a) (b)

Fig. 1: An example of the proposed system in action. The left image (a) shows an area from OSM with missing building footprints. The right image (b), our system is used to fill out the area.

contact and obtain the data directly from their sources, such as local governments or corresponding census institutions. This made it difficult to carry large scale studies involving multiple cities or even multiple countries. The rise of the internet and of the open-access data provided an alternative solution to this problem via volunteered geographic information (Goodchild, 2007). The OpenStreetMap (OSM) project [1], for example, is the largest collaborative effort to provide open access to geodata and currently contains information from a considerable number of cities around the world. However, this data is often non-uniform, requiring time-consuming preprocessing, and is frequently a victim of vandalization [9].

In particular, an issue with geographic data from regions outside the big metropolises is the frequent missing data for buildings. Roads and transportation networks can be considered the fundamental part of the map as they provide the means to come and go to the location being mapped. Therefore, they are often one of the first information about an area to be added to OSM. On the other hand, building data is laborious to map and, apart from points of interest, such as touristic attractions or shops, have little application for the average user. Consequently, areas outside of central business districts are often overlooked by mappers and present little to no building data.

However, building data can be important for a number of research applications such as traffic routing, regulation compliance monitoring, and natural disaster response management. Because the road networks are already available for many areas, we can generate building placements that mimic the building footprints of areas with incomplete data as well as to explore different ways in which cities could be reconfigured or remodeled. In turn, this would allow urban planners to experiment with different policies in simulation to help decision-making in tasks such as urban remodeling or controlling of urban sprawl.

[1] https://www.openstreetmap.org/

This paper presents a system for generating building placements for areas with incomplete building footprints in OSM. The system uses two mechanisms, one for generation of a diverse set of different building placement configurations, and another for the evaluation of the most fit individual in each range of the set. An example output of the system is presented in Figure 1.

The generation part of the system employs the MAP-Elites algorithm [8] to find different configurations of building placements and group them according to their features. The main property of Map-Elites is to search for high-quality solutions covering the full range of the feature space. This property can be ideal for this problem, as there are virtually an infinite number of valid ways in which the building placements could be reconfigured.

The evaluation part uses a Dense Convolutional Neural Network [4] classifier trained in areas that present the characteristics desired for the generation. The classifier evaluates the individuals with the highest fitness in each group in the range of the feature space and selects the highest scoring one for output. The generated individual is outputted in a ready-to-be-used OSM file format.

We evaluate the proposed system in Sumida City in the Greater Tokyo Area and show that despite a limited overall diversity in the candidate pool, the system can be used to complete areas with partial missing building footprints. We envision this system being used for a number of applications ranging from video games to social simulations and policy evaluation.

2 Related Works

Despite the increased availability of geographic data enabled by open-access projects such as the OpenStreetMap, the lack of data for specific regions or even data vandalizing are still a common problem [9]. The use of synthetic urban model data has been suggested as a viable way to perform a number of urban-related research that is often slowed down by the difficulty in obtaining such data [5]. While some applications do require real urban data from a particular area, using standardized synthetic urban data that contains the desired properties of the target area is often enough to answer research questions.

In this regard, academic research in city generators, traditionally limited to the area of computer graphics or video game development, are seeing their use in urban modeling for tackling problems related to natural disaster response management and regulatory compliance monitoring. City generation has been explored from a number of different perspectives. In computer graphics, there is the challenge of modeling and rendering 3D urban spaces, while in urban modeling, there is the challenge of simulating and visualizing the dynamics of urban areas [11]. Researchers have already explored generation systems that can create road networks [10], parcels [12], land uses [6], buildings layouts [7] to name a few.

Vanegas et al. [11] suggest a general pipeline for modeling urban spaces, as shown in Figure 2. The pipeline breaks down the urban modeling task into at least 3 high-level components that cities are traditionally built from: roads, blocks (or parcels), and buildings. Researchers have attempted to tackle one or more steps of this pipeline, and they present a survey of urban modeling works, of which we highlight a few that are relevant for our work in building placement generation.

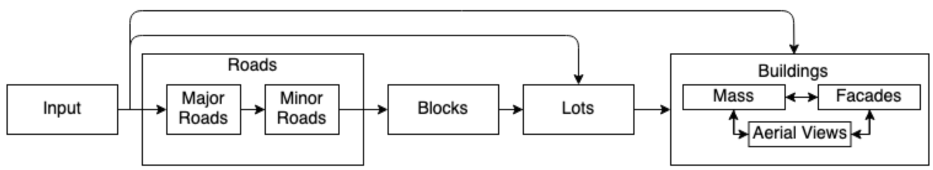


Fig. 2: Urban Modeling Pipeline as suggested by Vanegas et al. in Modelling the Appearance and Behaviour of Urban Spaces [11].

Nishida et al. [10] propose an example-based road network generation where a user-selected area is scanned and the general shapes of the street network are identified. These shapes are then used to generate a new road network in a different area, combined with a procedural approach to generate streets where the identified shapes can not be applied (such as rivers or lakes). Although the work provided a flexible and fast framework to generate a road network, it suffers from a trade-off: if too much example-based generation is used, the output resembles a simple copy of its input; if too much procedural-based generation is used, the output presents little resemblance to the input.

Vanegas et al. [12] propose a method for procedural generation of parcels in urban modeling. This method can be used in combination with procedural road generators to achieve a realistic-looking road network, including the residential streets that divide the parcels. The authors present two different algorithms for partitioning an area enclosed by roads into a parcel. The first one is given by a straight skeleton, which gives an internal structure for simple polygons by partitioning it in a tree-like fashion into monotone polygons. The second one is given by a recursive splitting of the block's polygonal area given by the oriented bounding boxes. The first algorithm always provides street access to its parcels, which can be a desired feature depending on the city. The use of only one of the algorithms or both of them at different levels is allowed. We draw inspiration from their second partitioning algorithm, as will shown in the next section.

Despite efforts into generating realistic buildings with different architecture types, the challenge of where and how to populate buildings in a given map that already contains road and building information is still relatively unexplored in the literature. To our knowledge, no work in the literature so far has looked into the problem of completing already existing geographic data available from a data-set. In our work, we attempt to generate building placements for urban areas that can can be integrated with already existing data in OSM.

3 Proposed System

3.1 Overview

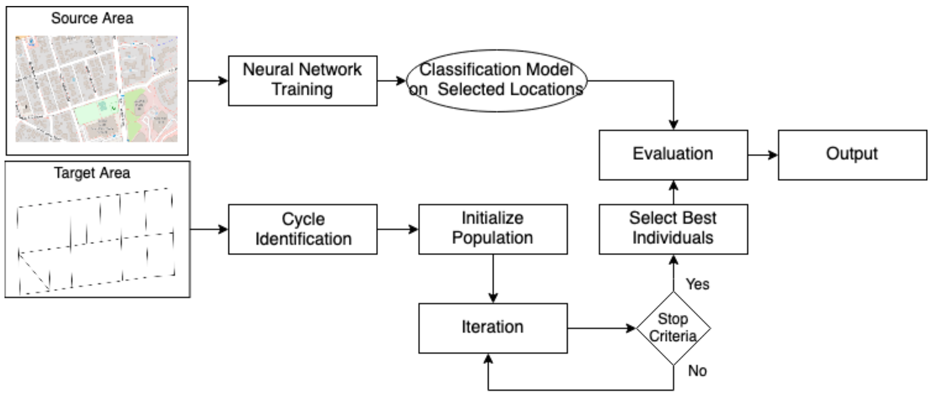


Fig. 3: Overview of the proposed system.

We propose a system that generates building placements for an area in OSM with an incomplete building footprint. We model a city or a given map as a collection of parcels (cycles in the road network graph) and assign numbers that correspond to the number of building placements inside each parcel. Different configurations of building placements are evolved for a fixed number of iterations. After the search is concluded, a convolutional neural network classifier model is used to evaluate the similarity between the area with generated building placements and a user-selected area.

An overview of the system is given by Figure 3. The system receives as input a source area and a target area. The source area correspond to OSM data that presents the desired building placements characteristics. The target area corresponds to the cycles in the road network of an area in OSM with incomplete building footprints. A convolutional neural network classifier is trained with image data from the source area and will be later used to evaluate the generated data for the target area. The cycles with and without building footprints in the target area area identified and used their information is used to represent building placement candidates in an evolutionary process. The system generates a pool of possible building placements for the parcels as the initial population and separates them into groups according to their building density characteristics. The initial population go through a fixed number of iterations, at which point the best individuals for each group are chosen to be evaluated by the classification model. The individual with highest accuracy is selected as output.

It is worth noting that, when considering the evaluation of the generated output, even when objective functions or clear constraints are imposed, it is difficult to determine what is a good result for the building placements generated. In our work, we try to make sure that the output is consistent with the existing data by employing a neural network classifier, but ultimately, deciding on the best output is up to the user. This is why we focus on the variety of outputs that the system can produce in our experiments, rather than to try and find the optimal building placement for the selected location. Accordingly, the images presented in this paper are results that we found reasonable, interesting or simply unexpected from our experiments.

3.2 Processing input

The first step of the system is loading the data from the source and target areas. The data from the source area is comprised of the OSM data file containing the whole information from the selected location, while the data from the target area is comprised of the chordless cycles of the road network in the area the building placements are to be generated. These cycles can be seen as the parcels formed by the streets in the road network.

The data from the source area is loaded, rendered at a fixed zoom level and saved as an image to be used during the training process of the classification model. Details on the classification model are on section 3.3. The data from the target area is loaded, and for each cycle in the road network, the centroid of the cycle area and its building densities are computed. A minimum spanning tree from the centroids is also computed to obtain closest neighbors for each parcel. A candidate output of building placements is then represented as a vector of numbers where each index represents a parcel and each value represents the number of buildings to be generated on that parcel. Details on the algorithm used to generate buildings from these values are on section 3.4.

3.3 Classification Model

The system employs a classification model to ensure that the generated output presents the building placement characteristics of a location specified by the user. The classification is done by a Dense Convolutional Neural Network [3] (DenseNet) classifier. A DenseNet is a type of convolutional neural network that simplifies the connectivity pattern between layers by connecting every layer directly with each other. They have been shown to provide state-of-the-art results in image classification tasks. The DenseNet-121 implementation provided by the Keras library is employed [2].

The positive and negative examples selected by the user are rendered at a fixed zoom level and split into 128x128 pixels sub-images to be used as input for the neural network during training. After the generation process, the fittest individual in each range of the feature space is rendered and evaluated by the classifier. The individual with the highest score in the evaluation process is selected for the output.

[2] https://keras.io/api/applications/densenet/

Automatically identifying the areas with significant missing building footprint is not a trivial task as it would generally require manual checking of each location of the map either in person or via satellite imagery. Although works specifically targeting the measuring of completeness of building footprints in OSM exist [2][1], for the purposes of this work, we assume that both the areas for training (low missing data) and generation (high missing data) are given by the user.

3.4 Building Placement Generation

Given a parcel and a number of building placements, buildings are generated following the method proposed for procedurally generating parcels in [12]. We apply a recursive partitioning algorithm following the geometry of the parcel, as demonstrated in Figure 4. The partitioning works as follows.

At each step of the algorithm, the oriented bounding box that encloses the current polygon (cycle of the graph) with minimum area is computed. Next, the middle point of the largest side of the box is identified. The polygon is partitioned by the line that intersects the middle point, orthogonal to the smallest side of the box. This process is repeated recursively until the number of desired building placements is achieved. Buildings in the OSM format are subsequently generated by shrinking the geometry of the resulting partition in 50%.

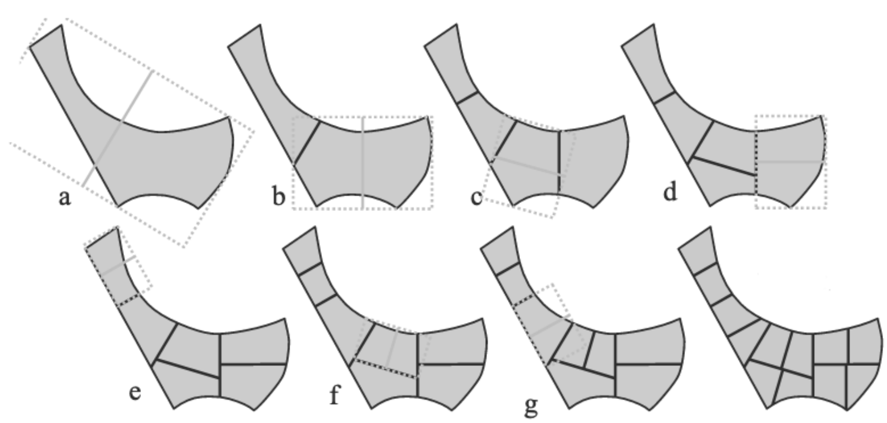


Fig. 4: Recursive oriented bounding box partitioning algorithm, modified from Procedural Generation of parcels in urban modeling by Vanegas et al. in [12]. At each step, the minimum oriented bounding box of the current polygon is identified and the polygon is partitioned in two from the larger side of the box. The process is repeated for the two generated sub-polygons.

3.5 Evolutionary Process

The system employs an evolutionary process inspired by the MAP-Elites algorithm [8] to generate a diverse set of building placement candidates before selecting one for output. Our candidate solution is modeled as a vector of numbers where each position in the vector represents a chordless cycle in the road network and each value in the vector represents the number of building placements for that parcel.

Two features are considered to characterize an individual. The first is simply the total number of building placements of the individual. The second is the way these placements are distributed among the parcels. For that end, the system uses a density error metric that returns higher values for candidates that present abrupt changes in the densities of two neighboring parcels. The system then prioritizes configurations that look natural by minimizing the divergence of building density in the surrounding areas of a parcel. It follows assumption that building densities tend to be high in areas in the business district centers and that this density slowly decreases towards suburban areas. While this may not be representative of every city existent, it gives the system a good approximation for the generated content to look natural.

A steady-state genetic algorithm is used to find different configurations of building placements that minimize the density error metric while trying to maximize the difference between the candidates. The steps for the process are broken down as follows:

- Population Initialization: random candidates are generated so that the range between minimum and maximum total densities is fully covered. This range is divided evenly and individuals are allocated to the groups according to their total building densities.
- Iteration: each candidate produces an offspring via mutation. If the child's building density is changed, the child is placed in the appropriate partition for that density. Otherwise, the child replace the parent in the current partition if its density error is better than the parent.
- Population downsizing: to ensure that a population of diverse individuals is obtained, remove candidates from the same partition that are similar enough from one another. A similarity metric and a fixed threshold are employed.

The second and third steps described above are repeated for a fixed number of times. After this process is finished, the candidates with lowest density error for each group in the feature space are selected to be evaluated by the classifier. The candidate scoring the highest accuracy in the classifier is selected to be outputted as an OSM file. Additionally, all the candidates can be presented to the user so he can make the final decision about the best output.

3.6 Similarity metric

To ensure that the population of a given group in the feature space contains a diverse enough configuration of building placements, a similarity metric is used to discard similar candidates. The main purpose of the similarity metric is to ensure that in a given group, only distinct enough candidates are saved. This similarity is used in the population downsizing, after every round of iteration, to keep the population diverse but prevent it from growing uncontrollable. We propose two similarity metric functions, one based on density orders and other based on density range.

- Density order similarity: The densities of every parcel of two individuals are computed and ordered in ascending order. The similarity between the two individuals is given by the percentage of parcels taking the same index in the ordered vector.
- Density range similarity: The densities of every parcel of two individuals are computed. The similarity between the two individuals is given by the percentage of parcels in with a parcel with lower density has density within 80% of the parcel with the same index in the other individual.

4 Experiments & Discussion

4.1 Experiment Setup

To evaluate our system, we selected the Greater Tokyo Area (GTA) as the area to perform our experiments. Despite being the densest metropolitan area in the world, large patches of urban areas have missing building footprints. The fact that it is also located in a region prone to natural disasters (such as earthquakes, tsunamis and typhoons) is another motivator, as exploring building placement configurations on a disaster simulation model may provide valuable information to local governments and urban planners.

We selected an area of approximately $200.000m^2$ in Sumida City, a special ward located to the east of the GTA. Figure 5 shows the area selected for the experiment. It can be observed that building data around the more popular commercial streets have been mapped, however, the majority of the building data around it is missing.

4.2 Similarity metric evaluation

First, we performed a preliminary test of the two proposed similarity metrics to evaluate which one would be the most appropriate for the system. It is important for the system to maintain both high quality individuals as well as a wide range of individuals with different building configurations.

In order to evaluate the similarity metrics, we run the system using the two proposed similarity metrics at each time, and compared how the total population as well as the density error changed during the iterations. We set as a threshold for an individual to be discarded if its building placement configuration had a similarity above 0.35 with another individual. In this case, the individual with higher error would be discarded.

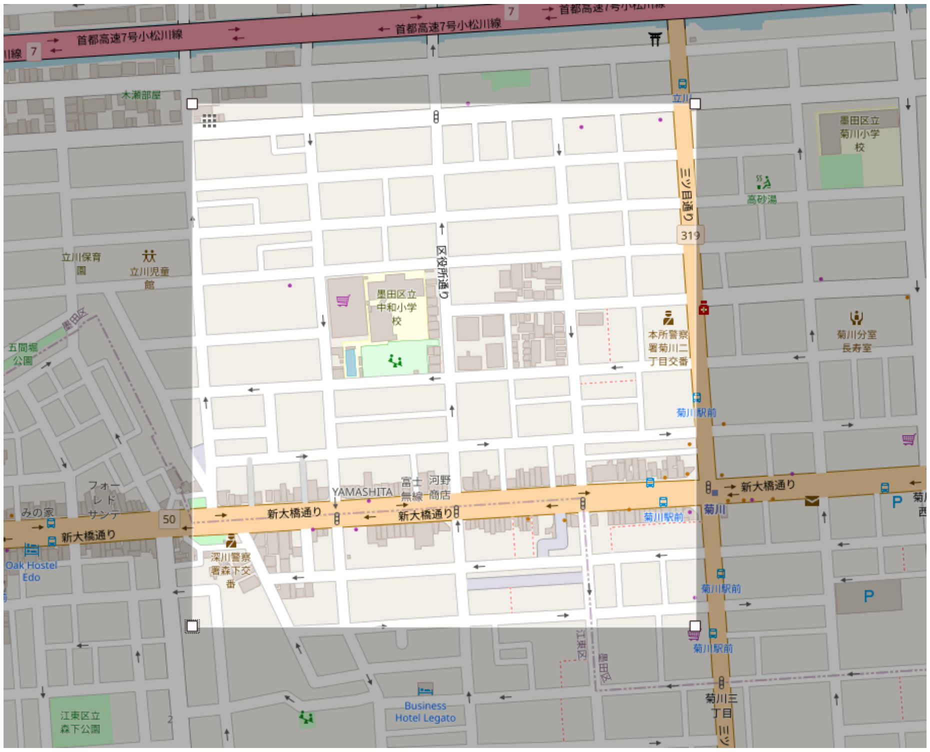
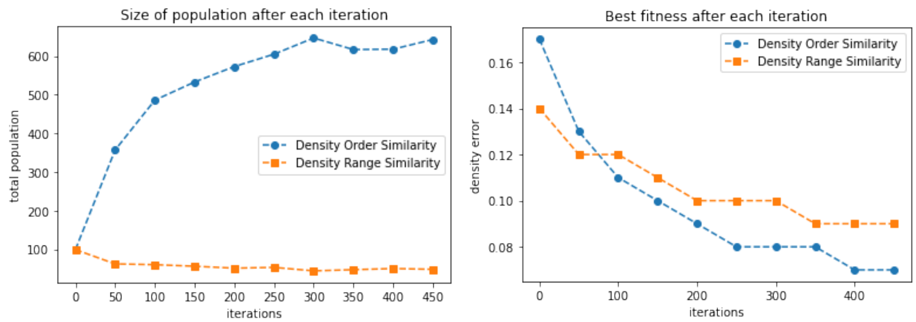


Fig. 5: The patch of Sumida City in the Greater Tokyo Area selected in OpenStreetMap used in our experiments.

Figure 6 shows the results for this experiment. In Figure 6a we can notice that the size of populations grows drastically for the density order similarity, while it mostly stabilizes after a short decrease for the density range similarity while in Figure 6b we can see that density order similarity had a better overall error after the same amount of iterations.

The abrupt increase in the population for the density order similarity metric can be explained by the fact the density order possibility space increases factorially. That is, given an area with n parcels, we have $n!$ different ways of computing density orderings. While this may be feasible for small sizes, in the case of our experiment location (38 parcels for an individual) the population may grow uncontrollably. And while the error metric obtained with the density order similarity was small at the end of the iterations, splitting the budget in terms of evaluations instead of iterations showed no noticeable difference between the two similarity metrics in terms of density error.



(a) (b)

Fig. 6: How the population and density error changed through iterations.

4.3 Building placements evaluation

From the preliminary experiment, we adopted the density range similarity metric and rerun the experiment as follows. We set as the maximum number of buildings for the area as 165 and used an initial population of 100 individuals. The range between the two features (0 to maximum buildings and 0 to maximum error) is evenly partitioned, and each individual is assigned a group according to their feature values. The density errors are normalized between 0 and 1 for the initial population, and we discard individuals with error greater than 1 during the evolutionary process.

Each individual generate an offspring for 500 number of iterations via a mutation of its chromosome. If the offspring's density or error metric fall outside the range of the parent, then it is placed in the appropriate group. If they are the same as the parent and the fitness is better, than the child replace the parent in the group, otherwise the child is discarded. The mutation rate is set to 0.1 and a mutation is given by an increase or decrease in the number of buildings of a certain parcel. The maximum value for this mutation is given proportionally to the current total number of buildings to ensure that candidates in all ranges of building density continue to be generated throughout the process.

After the 500 iterations, the individuals with best fitness for each group are selected to be rendered and tested against the neural network model and the highest rating is outputted. We also perform a qualitative analysis of all the best individuals for each group.

The code for the system as well as the code used for the experiment is available on our GitHub repository[3].

Results Figure 7 shows the resulting population after 500 iterations over the initial population and Figure 8 shows the rendered output of the highest scoring candidate in the classifier. The resulting population presents a larger number

[3] blind

of candidates for higher building densities, while for lower densities mostly one candidate per cell remained after the iterations. Additionally, the accuracy values from the classifier were for the most part the same throughout the grid with no noticeable difference in the accuracy values for any of the two axes in the grid.

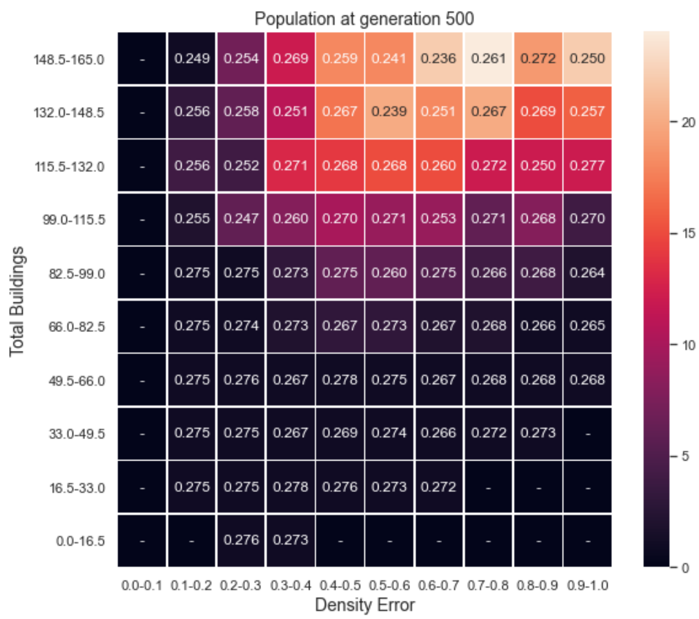


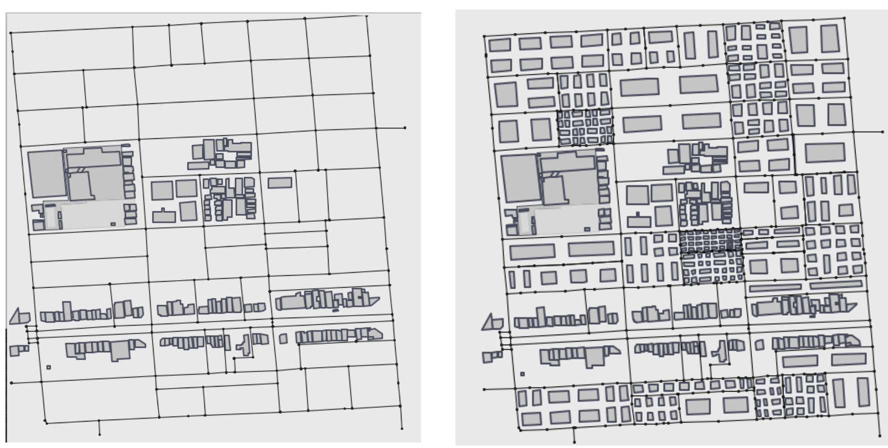
Fig. 7: Resulting population after 500 iterations. Each cell shows the highest accuracy obtained by a candidate of that cell in the classifier while the color indicates the total number of candidates remaining in the cell.

The resulting grid highlights a few problems. The first problem is about the coverage of feature space. The second problem is related to the the classifier evaluation.

The system was not able to fully cover the range of possible candidates, with some nearly empty cells for lower building densities. There is at least two reasons for this. First, the steady-state evolution of candidates with proportional mutational of the parcels was not enough to create candidates with different enough similarity scores between each other. Second, the fixed value of 0.35, obtained from the preliminary experiment, may be too low for candidates in the lower density range. Adopting a proportional similarity threshold may be able to alleviate this issue.

We expected that the classifier would be able to better tell apart different generated individuals. Since the GTA is a highly dense metropolis, candidates with higher density (towards the top of the grid) would score better in the evaluation process. However, the values were approximately the same for the whole range. It is possible that our procedural approach generate building footprints with geometries that fail to blend with the existing organic building placements of the area, resulting in lower accuracy values in the classifier. A solution to this problem would be using an example-based generation, where the geometry of existing buildings are using in the generated building placements.

While it can be argued that the procedural approach generates unnatural-looking building placements, a qualitative analysis of the sample results shows that, when the missing building footprints is not very large, the generated building placements can blend better with the existing data, such as the one shown previously in Figure 1b. Moreover, for a number of tasks in which the number and placement of the buildings (regardless of their geometries) is important, the system can still provide valuable information with the building placement generation.



(a) Original area in Sumida City

(b) Same area after generation

Fig. 8: Rendered output of the individual with highest accuracy according to the classifier.

5 Conclusion

We propose a system for generation of synthetic data for OpenStreetMap that finds building placements for locations with incomplete building footprints. The system employs an evolutionary search with a MAP-Elites algorithm to find different possible building placement configurations for a user-selected area. The candidate solutions are evolved and grouped according to their features in the feature space. Once the evolutionary process is over, a Dense Convolutional Neural Network classifier is employed to select the generated candidate with the highest similarity to a user-selected source area.

With the selected parameters for the experiment, the system was not able to find candidates in the full range of the features tested, but it did find and optimize at least one candidate for most cells. While the output may look unnatural for building placement generation in larger empty areas, the result can be convincing enough if there is a balanced mix of parcels with and without missing building footprints. The system can be a valuable tool to allow research in fields where building footprints are essential, such as in disaster response simulations.

5.1 Limitations & Future Work

The parameters used for the evolutionary process were not able to get a large set of candidates for the full range of features we tested for. Using other evolutionary algorithms (e.g. generational instead of steady-state) as well as adopting a similarity threshold according to the features of the individual may help alleviating this problem.

We currently employ a procedural technique to generate the building data, and although the geometry of each building generated is believable, when repeated through an extended area it loses the expressiveness of a real city. This property may also affect the accuracy of the classifier. Using already existing building geometries for the building placements can both increase the spontaneity of the output as well as allowing for better optimizing against a target urban area.

As future research opportunities, our system currently requires the user to select an area for building placement. However, given that large areas have missing building footprints in OSM, an interesting research direction would be to explore models that can identify the likelihood of an area to contain missing data information and automatically generate building placements for it. This would allow the system to be run in large scale areas, exploring different building configurations for whole cities, states or even whole countries.

Additionally, it is worth noticing that we are not currently considering any area-specific restrictions for the building placements. It would be valuable to explore how to integrate factors such as local regulations and zoning laws into the generation. This may be an important avenue to explore in order to encourage the use of this research by urban planners and local governments.

References

1. Brovelli, M.A., Zamboni, G.: A new method for the assessment of spatial accuracy and completeness of openstreetmap building footprints. ISPRS International Journal of Geo-Information **7**(8), 289 (2018)
2. Hecht, R., Kunze, C., Hahmann, S.: Measuring completeness of building footprints in openstreetmap over space and time. ISPRS International Journal of Geo-Information **2**(4), 1066–1091 (2013)
3. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4700–4708 (2017)

4. Huang, S.C., Coulom, R., Lin, S.S.: Monte-carlo simulation balancing in practice. In: International Conference on Computers and Games. pp. 81–92. Springer (2010)
5. Kim, J.S., Kavak, H., Crooks, A.: Procedural city generation beyond game development. SIGSPATIAL Special **10**(2), 34–41 (2018)
6. Lechner, T., Ren, P., Watson, B., Brozefski, C., Wilenski, U.: Procedural modeling of urban land use. In: ACM SIGGRAPH 2006 Research posters, pp. 135–es (2006)
7. Merrell, P., Schkufza, E., Koltun, V.: Computer-generated residential building layouts. In: ACM SIGGRAPH Asia 2010 papers, pp. 1–12 (2010)
8. Mouret, J.B., Clune, J.: Illuminating search spaces by mapping elites. arXiv preprint arXiv:1504.04909 (2015)
9. Neis, P., Goetz, M., Zipf, A.: Towards automatic vandalism detection in openstreetmap. ISPRS International Journal of Geo-Information **1**(3), 315–332 (2012)
10. Nishida, G., Garcia-Dorado, I., Aliaga, D.G.: Example-driven procedural urban roads. In: Computer Graphics Forum. vol. 35, pp. 5–17. Wiley Online Library (2016)
11. Vanegas, C.A., Aliaga, D.G., Wonka, P., Müller, P., Waddell, P., Watson, B.: Modelling the appearance and behaviour of urban spaces. In: Computer Graphics Forum. vol. 29, pp. 25–42. Wiley Online Library (2010)
12. Vanegas, C.A., Kelly, T., Weber, B., Halatsch, J., Aliaga, D.G., Müller, P.: Procedural generation of parcels in urban modeling. In: Computer graphics forum. vol. 31, pp. 681–690. Wiley Online Library (2012)