ArchIndex : A Web-based and Data-driven Retrieval System for City Blocks

Yichen Mo, Baizhou Zhang and Biao Li

Y. Mo (Corresponding author) • B. Zhang • B. Li
Institute of Architectural Algorithms and Applications, Southeast University, Nanjing, China
Email: moyichen@seu.edu.cn

B. Zhang
Email: zhangbaizhou@seu.edu.com

B. Li
Email: jz_studio@seu.edu.cn

1. Introduction

The adoption of AI and machine learning techniques provide a profound and long-lasting impact on the design process. The description and representation of the city are being radically transformed. However, current studies lack easy-to-use research tools [1]. Recently, analyzing large datasets in the browser is possible. Morphocode [2] visualizes the structure of Manhattan's urban fabric with real-time interactive tools. MIT Senseable City Lab [3] develops and deploys tools to learn about cities.

The increase in digital architectural data also inspired research on retrieval and indexing system. Benjamin [4] first introduces an information retrieval system for parcels, which allows searching for the shape of the building plots. Xu et. al. [5] build a 3-dimensional city case database of Rome including road, building, and functional information for the renewal of the area around the Roma Termini Railway Station.

This study introduces a retrieval system called ArchIndex. Automatically built from data of arbitrary cities, ArchIndex provides an interface that allows users to interactively modify and search for a list of similar city blocks in a web browser (Figure 1). Users can first get a general impression of the urban fabric of the city, then retrieve and explore similar city blocks with a unique block id for existing solutions in the city. The search results are marked on the city map, which also generates an index to the real world. With the retrieval system, architects can optimize the design of future buildings and urban renewal considering the coherence with the urban context.

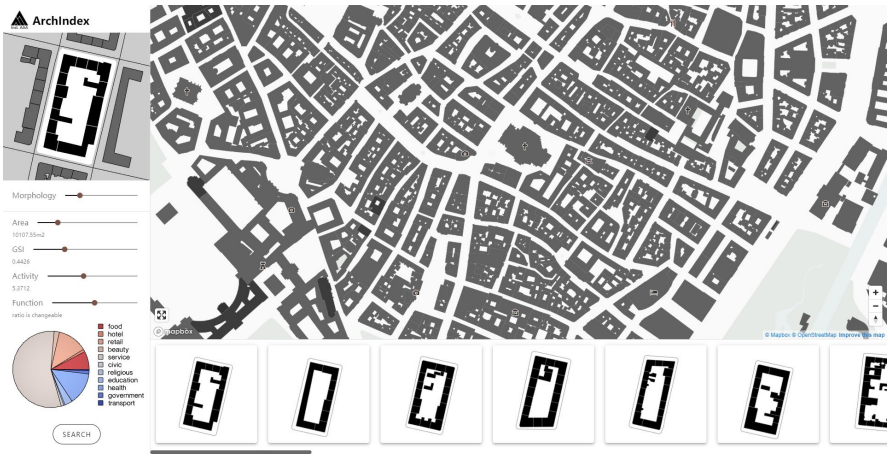

Figure 1. Interface for retrieval system. Left: input image and attributes. Bottom: retrieved result. data © OpenStreetMap

2. Method

2.1 Data Acquisition and Set Up

The city data is collected from OpenStreetMap(OSM), including buildings, roads, GPS tracks, and POIs. OSM, as a wiki-style project, has high-quality vector geographical data updated by volunteers each day. The data acquisition pipeline automatically generates a city-scale spatial database from web APIs. This paper uses the map of the city Vienna, more than 8000 valid city blocks extracted from the main networks of the city.

ArchIndex uses city block as a basic unit for a retrieval system (Figure 2). Within the block unit, the author proposes 5 properties as indicators for the retrieval system. *Area* and *GSI* are pre-calculated from the block and building polygon; *Morphology*, *Activity,* and *Function* are described in detail in the next subsection. After preprocessing, the value of the 5 indicators is also stored in the database table. In the application, the geometry and other information can be acquired from the database by authorized users.

Figure 2. City block as a basic unit of index, 3D city data ©CityJSON

2.2 Deep Feature Extraction

Feature extraction has a long history in the computer vision field with discussions about using manual extraction or machine intelligence. The implementation of *Area* and *GSI* in this study is also a comparison between manual and automated methods.

In recent years, deeper convolutional neural networks have changed the landscape of computer vision, especially in image recognition and segmentation [6]. The use of a pre-trained model can quickly generate the feature maps from image datasets.

Each city block is presented by the block boundary and inside buildings as a 500x500 raster image. After validation of block boundary including checking the area and built areas of the block, the number of valid block images is 6662. The author experiments the pre-trained Inception v3 model[7], the pre-trained CapsNet[8] with the initial weight of 99.31% accuracy on MNIST datasets[9], and an Autoencoder[10]. The choice of the model is an optional parameter considering the rapid evolution of deep learning models. The extractor is also the most replaceable module in the application (Figure 3).

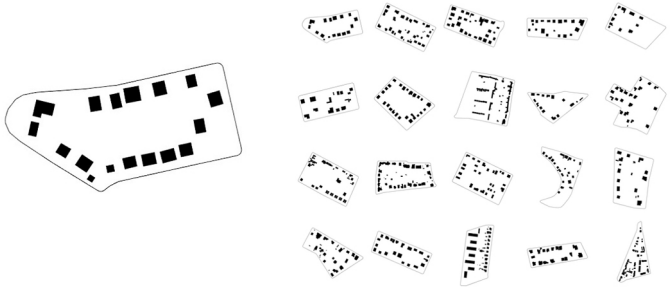

Figure 3. Dataset image and trial nearest neighbor search result

2.3 Trajectory Density

The trajectory data come from the GPS-equipped devices, which can be regarded as mobile sensors probing the activity of an area. For each block, selected trajectories intersect with the axis-aligned bounding box of the block polygon. The trajectories are evenly divided by 20 meters (Figure 4). The quantitative indicator of *Activity* is represented by:

$$Activity = \frac{N}{A} \tag{1}$$

Where N is the number of divided points, A is the block area.

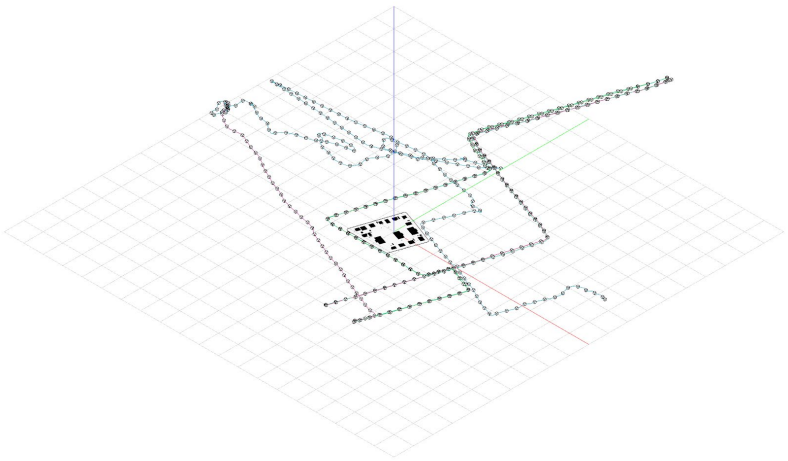

Figure 4. Trajectory density as activity indicator

2.4 Functional Diversity

POI data is pre-classified into 13 categories to describe the function type of each position. Functional diversity is the number of functional points to the total number within a 5-minute walking circle, about 400 meters (Figure 5). Therefore, the length of the resulting vector is 13 and the sum of the values is 1.

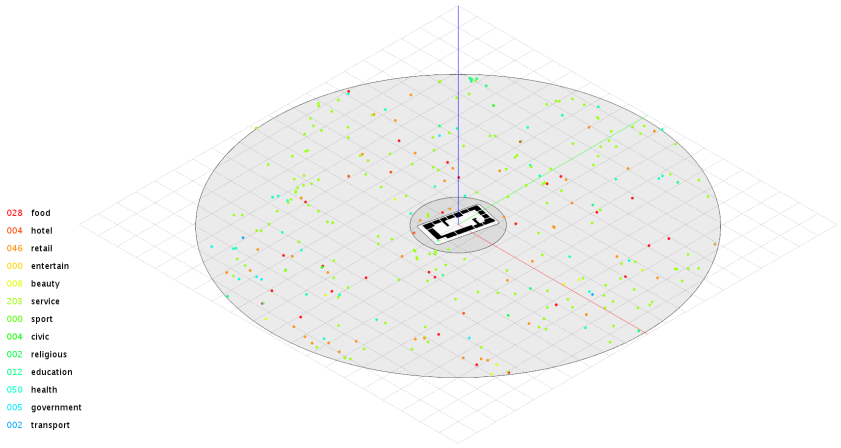

Figure 5. Functional diversity is a 13-length vector indicator

2.5 Euclidean Distance

The similarity measure between samples uses Euclidean distance with all the indicators normalized to [0, 1]. Moreover, the distance function includes a user-specified coefficient to describe the weight of each indicator.

3. Application

This paper provides an exploratory design tool at the early stage of the architectural and urban design process. In this interactive retrieval system, users can directly explore archetypes in the city. ArchIndex also allows users to edit the building footprints, functional diversity or adjust the coefficients for retrieving, offering feedback to the user's design.

3.1 Interactive Retrieval System

The retrieval system is public as a website, which uses a WebGL frontend. The backend including Python and Java for feature processing and spatial database connection respectively. The glue that connects the different ends of the browser and server is the WebSocket protocol and the JSON data format.

The retrieval procedure starts with a block ID as input. The server will soon return the information of the city block and similar sample results. ArchIndex hides the underlying mechanism (Figure 6) and provides a user-friendly interface.

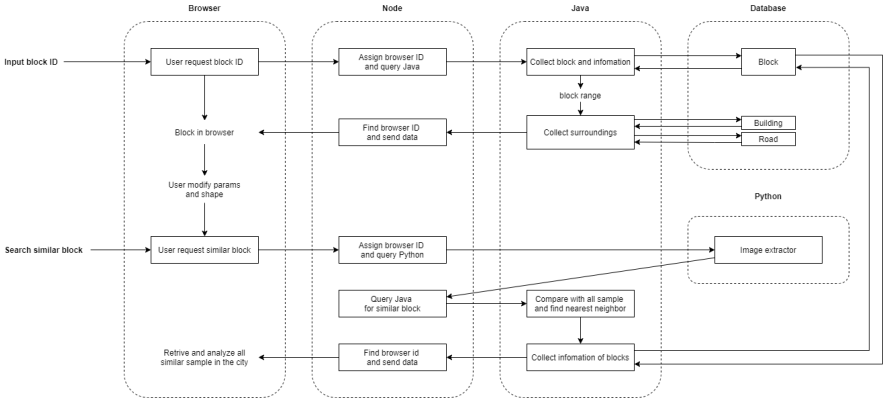

Figure 6. Overview framework and data pipeline for the retrieval system

3.2 Sample Query

For a sample query, users only need to set the block *Morphology*, *Area*, *GSI*, *Activity*, and *Function* information, then press the search button. The blocks in the result panel use tooltip to show the numerical information and can mark the block location on the map by click on the resulting image. Figure 7,8 shows the results of sample queries under different search conditions.

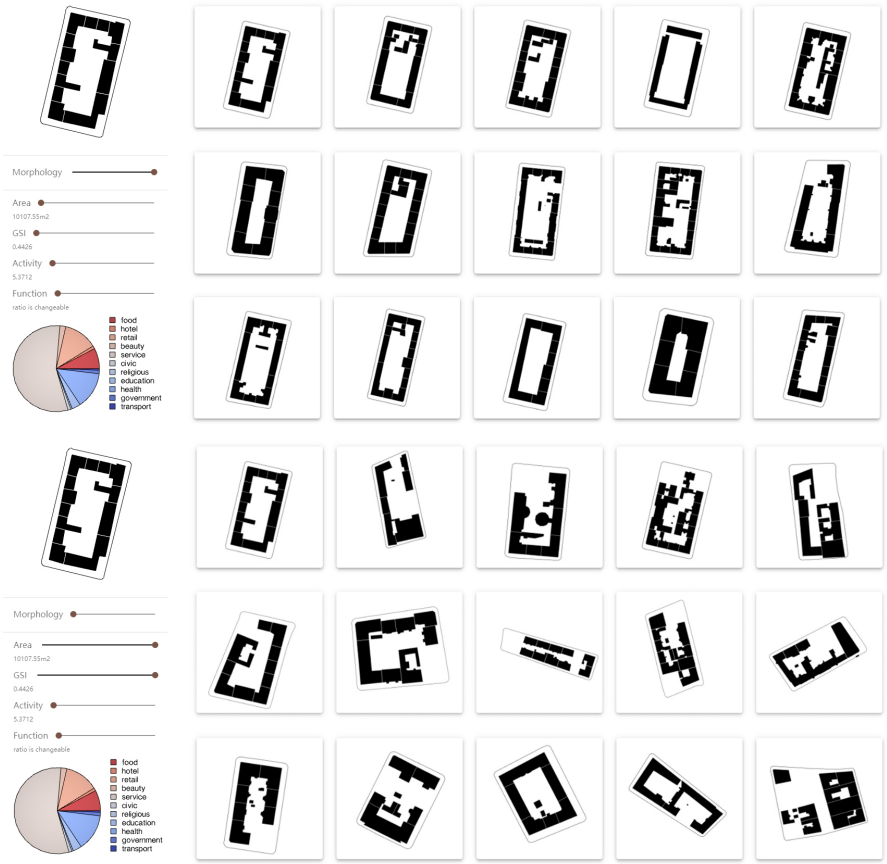

Figure 7. Use ArchIndex as a comparison tool to contrast the manual and automated morphological descriptions

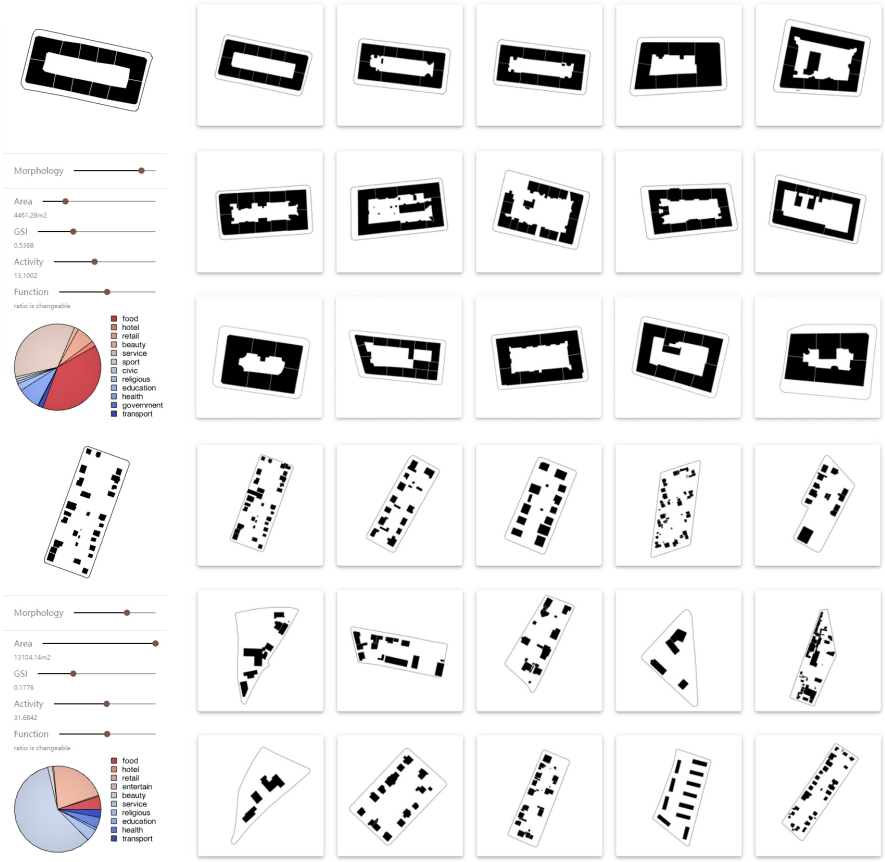

Figure 8. Use ArchIndex explore archetypes of the city

4. Discussion

Before the design stage, ArchIndex allows architects to emphasize existing urban forms. Re-evaluating the value of historical urban fabric forms is beneficial to rediscovering the significance of urban context and improving the identity of the city. The methodology and tool chain of this paper also structure the comparison mechanism of different descriptive indicators of city blocks. Moreover, the program can quickly set up and build an application for any other cities. Further research could concentrate on finding numerical representations for abstract concepts such as spatial relationships and graph structure.

References

[1] Boeing, G. (2017). OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. Computers, Environment and Urban Systems, 65, 126-139.

[2] MORPHOCODE. (2020). Home - MORPHOCODE. https://morphocode.com/. Accessed 1 January 2021.

[3] MIT Senseable City Lab. (2020). MIT Senseable City Lab. https://senseable.mit.edu/. Accessed 1 January 2021.

[4] Dillenburger, B. (2010). Space Index: A retrieval-system for building-plots. *Future Cities: proceedings of the 28th Conference on Education in Computer Aided Architectural Design in Europe, September 15-18, 2010, Zurich, Switzerland, ETH Zurich* : vdf Hochschulverlag AG and der ETH Zürich.

[5] Xu, J., & Li, B. (2019). Application of Case-Based Methods and Information Technology in Urban Design-The Renewal Design of the urban region around Roma Railway Station. *Intelligent & Informed - Proceedings of the 24th CAADRIA Conference*, Wellington, New Zealand, 625-634.

[6] K. He, X. Zhang, S. Ren, & J. Sun. (2016). Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 770–778). doi:10.1109/CVPR.2016.90.

[7] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2818-2826).

[8] Sabour, S., Frosst, N., & Hinton, G. E. (2017, December). Dynamic routing between capsules. In Proceedings of the 31st International Conference on Neural Information Processing Systems (pp. 3859-3869).

[9] GitHub. (2020). Cedrickchee/capsule-net-pytorch. http://github.com/cedrickchee/capsule-net-python. Accessed 1 April 2021.

[10] Moosavi, V. (2017). Urban morphology meets deep learning: Exploring urban forms in one million cities, town and villages across the planet. *arXiv e-prints*, arXiv:1709.02939.