

# The Pattern of Life Human Mobility Simulation (Demo Paper)

Hossein Amiri

Emory University, USA

[hossein.amiri@emory.edu](mailto:hossein.amiri@emory.edu)

Joon-Seok Kim

Oak Ridge National Laboratory, USA

[kimj1@ornl.gov](mailto:kimj1@ornl.gov)

Dieter Pfoser

George Mason University, USA

[dpfoser@gmu.edu](mailto:dpfoser@gmu.edu)

Will Kohn

Emory University, USA

[will.kohn@emory.edu](mailto:will.kohn@emory.edu)

Hamdi Kavak

George Mason University, USA

[hkavak@gmu.edu](mailto:hkavak@gmu.edu)

Carola Wenk

Tulane University, USA

[cwenk@tulane.edu](mailto:cwenk@tulane.edu)

Shiyang Ruan

George Mason University, USA

[sruan@gmu.edu](mailto:sruan@gmu.edu)

Andrew Crooks

University at Buffalo, USA

[atcrooks@buffalo.edu](mailto:atcrooks@buffalo.edu)

Andreas Züflle

Emory University, USA

[azufle@emory.edu](mailto:azufle@emory.edu)

## ABSTRACT

We demonstrate the Patterns of Life Simulation to create realistic simulations of human mobility in a city. This simulation has recently been used to generate massive amounts of trajectory and check-in data. Our demonstration focuses on using the simulation twofold: (1) using the graphical user interface (GUI), and (2) running the simulation headless by disabling the GUI for faster data generation. We further demonstrate how the Patterns of Life simulation can be used to simulate any region on Earth by using publicly available data from OpenStreetMap. Finally, we also demonstrate recent improvements to the scalability of the simulation allows to simulate up to 100,000 individual agents for years of simulation time. During our demonstration, as well as offline using our guides on GitHub, participants will learn: (1) The theories of human behavior driving the Patterns of Life simulation, (2) how to simulate to generate massive amounts of synthetic yet realistic trajectory data, (3) running the simulation for a region of interest chosen by participants using OSM data, (4) learn the scalability of the simulation and understand the properties of generated data, and (5) manage thousands of parallel simulation instances running concurrently.

## CCS CONCEPTS

- Information systems → Geographic information systems ; Location based services .

## KEYWORDS

Pattern of Life, Simulation, Trajectory, Dataset, Customization

## ACM Reference Format:

Hossein Amiri, Will Kohn, Shiyang Ruan, Joon-Seok Kim, Hamdi Kavak, Andrew Crooks, Dieter Pfoser, Carola Wenk, and Andreas Züflle. 2024. The Pattern of Life Human Mobility Simulation (Demo Paper). In *Proceedings of (SIGSPATIAL '24)*. ACM, New York, NY, USA, 4 pages. <https://doi.org/XXXXXX.XXXXXXXX>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

SIGSPATIAL '24, October 29 – November 1, 2024, Atlanta, GA

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-XXXX-X/18/06...\$15.00

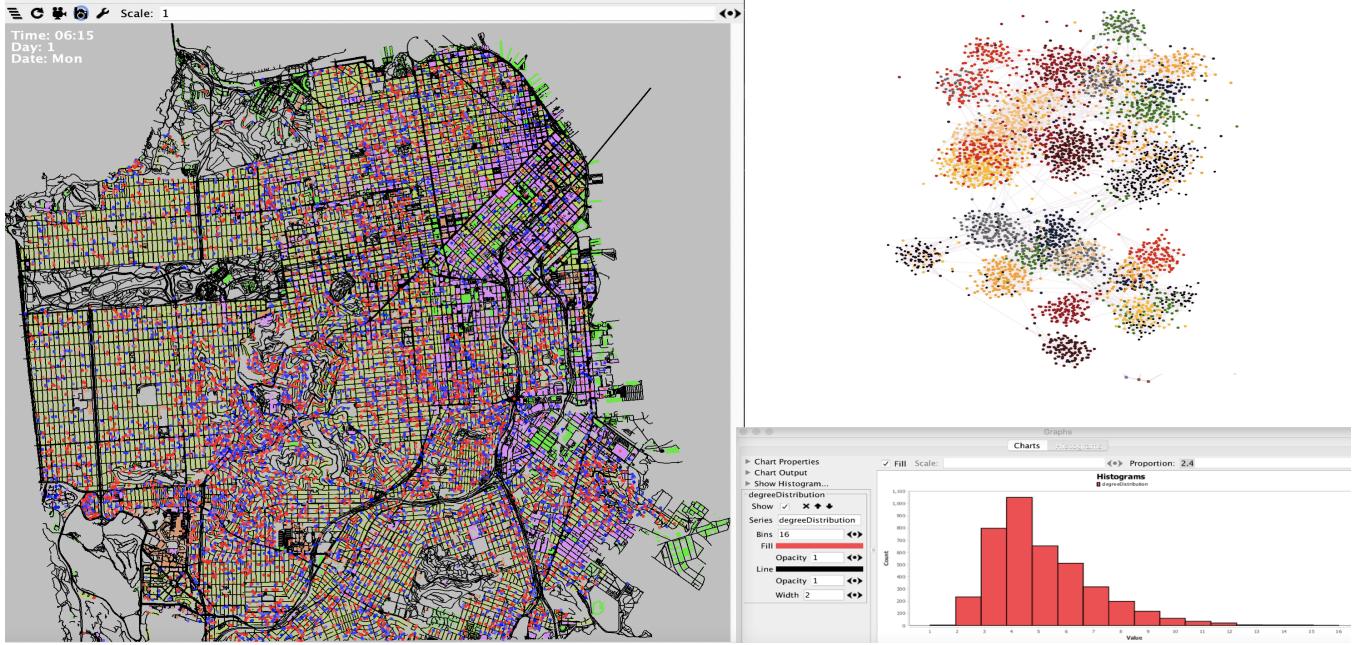
<https://doi.org/XXXXXX.XXXXXXXX>

## 1 INTRODUCTION

Trajectory data [2] captures frequently measured location sequences of objects such as humans [18], vehicles [16], and animals[4]. Trajectory data enables a deeper analysis of human behaviors [15, 19], their mobility [12], enhancing infectious disease modeling [8], and facilitating urban mobility studies [20]. Additionally, it is instrumental in traffic analysis [5], the detection of anomalies [17] and urban planning [6]. Despite the plethora of applications for analyzing trajectory data, research is hindered by a lack of available trajectory data sets. The largest and most commonly used trajectory dataset of individual human trajectories is GeoLife [18] which captures only 182 individual human users where most of these users are captured only for a few hours or even minutes [7].

To fill this gap and enable research on large trajectory datasets, our team has developed the Patterns of Life Simulation [21] which was created in Java leveraging the MASON agent-based modeling toolkit [9]. This simulation models the Maslowian [11] needs of agents (i.e., people) which drive agents' actions and behaviors: Agents need to go home to find shelter overnight; Agents need to go to work to make money (i.e., financial balance); Agents need to go to restaurants to eat; and Agents need to go to recreational sites to meet friends and sustain their social network. Many more facets of the logic driving the behavior of agents in the Patterns of Life simulation can be found in [21]. Figure 1 shows a screenshot of the graphical user interface (GUI) used to show the movement of agents across a study region and their social network. We used the Patterns of Life simulation to generate large sets of check-in data [7] and large sets of trajectory data [2]. However, these datasets pertain only to a small set of pre-selected study regions. This demonstration will show how the Patterns of Life Simulation can be adapted by users for their own purpose and their own study region. Specifically, this demonstration will explain how to:

- Run the Patterns of Life Simulation with the GUI to visually explore the mobility patterns of a region
- Run the Patterns of Life Simulation headless (without GUI) for large-scale data generation
- Adapt the simulation to any region in the world using OpenStreetMap [13] data
- Showcase recent scalability improvements allowing to simulate hundreds of thousands of agents
- Demonstrate a framework that supports concurrently running thousands of instances of the simulation.



**Figure 1:** A screenshot of the graphical user interface of the Patterns of Life Simulation. The GUI shows the map and the movements of agents on the left side and the social network of agents and their statistical properties on the right side.

## 2 DEMONSTRATION

The demonstration will be organized as follows: Section 2.1 provides an overview of the Patterns of Life Simulation using GUI and Section 2.2 demonstrates how to use the simulation for data generation with the GUI. Then Section 2.3 describes how OpenStreetMap data can be used to change the simulation to any region of interest in the World. Section 2.4 details the changes we have made to increase the scalability of the simulation and Section 2.5 demonstrates a framework to run many simulations in parallel very large-scale trajectory data generation. In this demonstration, we utilized what we call the “Vanilla” simulation as a baseline and is located in the following GitHub repository: <https://github.com/gmuggs/pol>. All the improvements, optimization, and documentation to reproduce this demonstration are found at <https://github.com/onspatial/generate-mobility-dataset>

### 2.1 Patterns of Life (POL) Demonstration: GUI

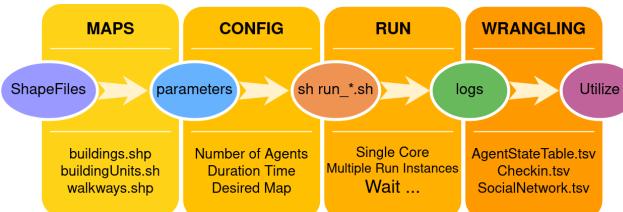
For the demonstration to be presented at SIGSPATIAL '24, we will run the Patterns of Life Simulation live to generate trajectory data. We have prepared several maps, including those of Atlanta, USA; Beijing, China; Berlin, Germany; and San Francisco, USA. By default, we will use the Atlanta map using 1000 agents for this demonstration. We choose such a small number of agents to avoid waiting times during the demonstration for simulation initialization (we will come back to run time of the simulation in Section 2.4). First, we will demonstrate a single run of the simulation using the GUI to explain how the simulation works, how agents move across the map, and how they interact with each other to form social networks as seen in Figure 1. While the simulation is running, we will give a brief overview of the theories of social behavior guiding agents’ behavior including Maslow’s Hierarchy of Needs [11] and the Theory of Planned Behavior [1]. For the live-audience at SIGSPATIAL '24

we will include details of these theories on a poster that will be exhibited next to our demonstrator. We will also briefly explain how the social network (on the right of the GUI shown in Figure 1) is formed and evolves. The interested reader of this paper may find these details in our simulation description [21]. After simulating for about one minute (wall-clock time), about ten days of simulation time will have been generated. We will show the generated log files including trajectories, check-ins, and time-series of agent states.

### 2.2 POL Demonstration: Headless

To use the Patterns of Life Simulation for large-scale data generation without having the GUI as a computational bottleneck, we also show how the simulation can be run headless. The overall process to run the simulation and generate a single trajectory dataset is outlined in Figure 2. After generating the maps and configuring the simulation parameters, the simulation is started using a bash script using the pre-compiled jar file of the Patterns of Life simulation. The simulation generates the logs in the specified directory. The logs contain information about the agents’ movements, interactions, social network, check-in, and other data that can be specified for specific applications or disabled for efficiency.

Using a Bash script on Linux, we will demonstrate a use case that involves running a single simulation instance with predefined parameters to showcase the data generation process for that instance. Furthermore, we will explore the creation and customization of log files during this stage, using both the simulation source code and the provided data processing scripts. To demonstrate the data visually, our poster will also show a simulated data set such as Figure 3 which shows a downsampled version of generated trajectories data in San Francisco. We will illustrate how to alter the simulation’s source code to integrate custom features and subsequently compile this modified code into a JAR file.



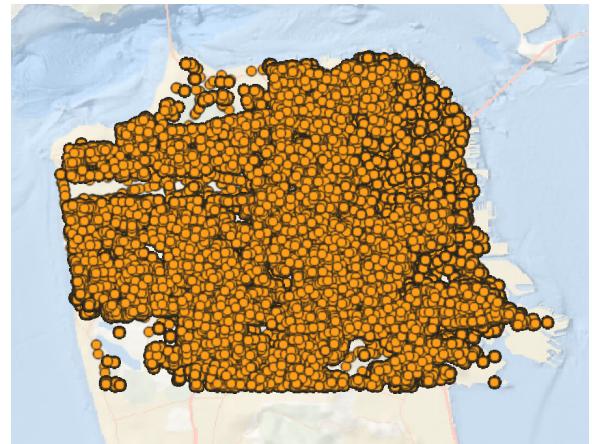
**Figure 2: Steps to generate the one trajectory dataset.**

### 2.3 Adapting POL to Simulate any Region in the World

Then, we will ask people in the audience which region of interest, anywhere in the world, they would like to simulate. We will obtain map data for this region from OpenStreetMap (OSM), transform the data into the format required by the simulation, run the simulation on this new region, and explain how the simulation uses information about building footprints as well as information on residential and commercial areas to initialize the simulation. Specifically, to generate a new map for the simulation, we follow the steps outlined in Figure 4 to create the spatial layers required for the simulation environment. Specifically, the simulation relies on three distinct shapefiles: `buildings.shp`, `buildingUnits.shp`, and `walkways.shp`. The `buildings.shp` contains the footprints of buildings and information on their usage as residential or commercial. While OSM does not include building usage information for every building, it has been shown that building usage can be accurately predicted from other building features [3]. This file is created using the Overpass API for OSM and then processed in QGIS to filter and assign attributes that define building type. The `buildingUnits.shp` file is then generated from the `buildings.shp`, by creating smaller individual units (apartments, shows) within a building, thereby facilitating more granular simulations of movement and interaction within the building premises. In the same way, the `walkways.shp` file is created to map out the transportation network for agents to move between buildings. Once these files are generated and placed in the corresponding folders, the Patterns of Life Simulation is started with the new study region. For readers unable to attend the demonstration in-person, we document instructions how OSM data can be quickly obtained and used in the simulation in our GitHub repository at <https://github.com/onspatial/generate-mobility-dataset/blob/main/documentation/maps.md>

### 2.4 Large-Scale Data Generation

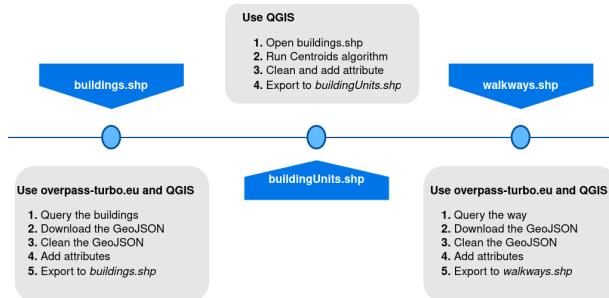
Initial profiling of the original Patterns of Life Simulation, facilitated by VisualVM [14], enabled the identification of critical performance bottlenecks. A key finding was the importance of initial parameter settings, such as the agent's starting financial balance. During early simulation stages, agents predominantly struggled with paying their rent. This caused agents to frequently change jobs and home locations in an attempt to satisfy their Financial Need (i.e., earn money). This behavior created a substantial computational bottleneck. In the updated version used for this demonstration, we limited the agent's ability to re-evaluate their home and job to only once-per-day rather than every five-minute tick. In addition, we changed



**Figure 3: Example Dataset Visualization**

agents' decision-making to use Euclidean distance to find the nearest places of interest rather than network distance. While this is a change to the agents' logic, we argue that this change is realistic, as humans do not run a Dijkstra single-source shortest-path search to find nearby places to visit, but rather use an approximation based on the human's mental map of the area [10]. These modifications, coupled with general code cleanup and optimization of the logging system, markedly improved simulation performance. We also addressed compile-time warnings and disabled journal record updates at each simulation step to streamline execution. Additionally, we enhanced output readability, for instance, by converting the time display from milliseconds to a more comprehensible format. These adjustments not only alleviated the identified bottlenecks but also improved the overall functionality and reliability of the simulation.

We conducted a comparative analysis of the initialization and simulation time between the original ("Vanilla") and the improved ("Improved") version of the Patterns of Life Simulation. The results are presented in Table 1. The new version is significantly faster than the Vanilla version in both initialization and simulation time. For instance, when simulating 1000 agents, the "improved" version is 3.8 times faster in initialization and 8.1 times faster in simulation time compared to the "vanilla" version. The performance improvement is more pronounced as the number of agents increases. For example, when simulating 150,000 agents, it was not reasonable to run the Vanilla version due to the long initialization and simulation time. In contrast, the "Improved" version can handle this number of agents with a reasonable time frame. An example of generated data can be found in [2] we used the framework to generate a dataset with various number of agents and maps for different regions. Since we won't have time to regenerate these very large datasets at SIGSPATIAL, we will instead explore already-generated datasets which include billions of check-ins, trillions of social links, and tens of trillions of trajectory points. We will note that these numbers are many order of magnitude larger than any existing datasets. The reader can find more details in our Data & Resource Paper [2].



**Figure 4: Steps to generate a new map for the simulation.**

Time	Initialization		Simulation		
	Agents	Vanilla	Improved	Vanilla	Improved
1000	33.22 s	8.66 s	10.3 min	1.27 min	
5000	4.61 min	48.83 s	3.69 hour	5.68 min	
10,000	13.87 min	3.27 min	15.8 hour	14.73 min	
15,000	24.12 min	8.74 min	35.45 hour	32.56 min	
100,000	DNF	4.37 hour	DNF	9.0 hour	
150,000	DNF	9.65 hour	DNF	19.29 hour	

**Table 1: Comparative Analysis of Initialization and Simulation Time for "Vanilla" and "Improved" Versions Over a Ten-Day Period**

## 2.5 Parallelization of POL

Since the simulation uses one single core, other simulation instances can be run in parallel to generate more data. We developed tools to run simulations in parallel, parameterize the simulations for specific use cases, and process logs to format output data in a customized format. The parallelization framework can operate using either fork-join or job-queue approaches. In the fork-join model, all processes must wait for the slowest one, whereas the job-queue model does not wait and ensures maximum parallelism is achieved. The fork-join approach is suitable when the next generation in a parallel world requires information from its predecessors, such as in genetic algorithms. Conversely, the job-queue approach is beneficial for studying varying parameter data without dependencies. Upon completion of the simulations, data processing techniques can be employed to concatenate and pre-process the logs for more in-depth studies, such as plotting them or run machine learning algorithms on the output. The source code for the enhanced simulation, parallelization framework, and data-processing tools is available online at <https://github.com/onspatial/generate-mobility-dataset>.

We will demonstrate how to set up a large number of simulations, allow the algorithms to run all simulations in parallel, and generate data. Additionally, we will discuss how to calibrate the parameters at each step to ensure the data are as expected and properly calibrated for the research.

## 3 CONCLUSION

Human mobility data is a crucial component in various research fields, including urban planning, epidemiology, and transportation. The availability of large-scale human mobility data has enabled researchers to develop models and tools to analyze and predict human behavior. However, generating such data is challenging due to

the complexity of human mobility patterns and the need for large-scale data to develop accurate models. In this paper, we present a framework for generating large-scale human mobility data using an existing agent-based simulations. We demonstrate how to optimize the simulation code, generate new maps, run the simulation, and process the logs to generate the final data. We also show how to run multiple simulations in parallel to generate large-scale data efficiently. We present the results of our framework, including the performance improvement of the optimized simulation code and the generated data. The framework is flexible and can be adapted to different research needs by customizing the simulation parameters and data processing steps. The source code for the enhanced simulation, parallelization framework, and data-processing tools is available online at <https://github.com/onspatial/generate-mobility-dataset>.

## REFERENCES

- [1] Icek Ajzen. 1991. The theory of planned behavior. *Organizational behavior and human decision processes* 50, 2 (1991), 179–211.
- [2] Hossein Amiri et al. 2023. Massive Trajectory Data Based on Patterns of Life. In *SIGSPATIAL '23*, 1–4.
- [3] Kuldip Singh Atwal, Taylor Anderson, Dieter Pfoser, and Andreas Züfle. 2022. Predicting building types using OpenStreetMap. *Scientific Reports* 12, 1 (2022), 19976.
- [4] Clément Calenge, Stéphane Dray, and Manuela Royer-Carenzi. 2009. The concept of animals' trajectories from a data analysis perspective. *Ecological informatics* 4, 1 (2009), 34–41.
- [5] Wenqiang Chen, Tao Wang, et al. 2022. Lane-based Distance-Velocity model for evaluating pedestrian-vehicle interaction at non-signalized locations. *Accident Analysis & Prevention* 176 (2022), 106810.
- [6] Sibren Isaacman, Richard Becker, et al. 2012. Human mobility modeling at metropolitan scales. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*. 239–252.
- [7] Joon-Seok Kim, Hyunjee Jin, et al. 2020. Location-based social network data generation based on patterns of life. In *MDM*. IEEE, 158–167.
- [8] Will Kohn, Hossein Amiri, and Andreas Züfle. 2023. EPIPOL: An Epidemiological Patterns of Life Simulation (Demonstration Paper). In *SIGSPATIAL SpatialEpi Workshop*. 13–16.
- [9] Sean Luke, Robert Simon, Andrew Crooks, Haoliang Wang, Ermo Wei, David Freelan, Carmine Spagnuolo, Vittorio Scarano, Gennaro Cordasco, and Claudio Cioffi-Revilla. 2018. The MASON Simulation Toolkit: Past, Present, and Future. In *Proceedings of the 19th International Workshop on Multi-Agent-Based Simulation*, Paul Davidsson and Harko Verhagen (Eds.). Stockholm, Sweden, 75–87.
- [10] EJ Manley, JD Addison, and T Cheng. 2015. Shortest path or anchor-based route choice: a large-scale empirical analysis of minicab routing in London. *Journal of transport geography* 43 (2015), 123–139.
- [11] Abraham H Maslow. 1943. A theory of human motivation. *Psychological review* 50, 4 (1943), 370.
- [12] Mohamed Mokbel, Mahmoud Sakr, Li Xiong, Züfle, et al. 2024. Mobility Data Science: Perspectives and Challenges. *ACM Transactions on Spatial Algorithms and Systems* (2024).
- [13] OpenStreetMap contributors. 2023. Overpass turbo. <https://overpass-turbo.eu/>.
- [14] Oracle. 2017. VisualVM. <https://github.com/oracle/visualvm> Accessed on 2024-05-28.
- [15] Eran Toch, Boaz Lerner, Eyal Ben-Zion, and Irad Ben-Gal. 2019. Analyzing large-scale human mobility data: survey of machine learning methods and applications. *Knowledge and Information Systems* 58 (2019), 501–523.
- [16] Jing Yuan, Yu Zheng, Xing Xie, and Guangzhong Sun. 2011. T-drive: Enhancing driving directions with taxi drivers' intelligence. *IEEE Transactions on Knowledge and Data Engineering* 25, 1 (2011), 220–232.
- [17] Zheng Zhang, Hossein Amiri, et al. 2023. Large language models for spatial trajectory patterns mining. *arXiv preprint arXiv:2310.04942* (2023).
- [18] Yu Zheng, Xing Xie, Wei-Ying Ma, et al. 2010. GeoLife: A collaborative social networking service among user, location and trajectory. *IEEE Data Eng. Bull.* 33, 2 (2010), 32–39.
- [19] Feng Zhu, Cheng Chang, Zhiheng Li, Boqi Li, and Li Li. 2024. A generic optimization-based enhancement method for trajectory data: Two plus one. *Accident Analysis & Prevention* 200 (2024), 107532.
- [20] Yuanshao Zhu et al. 2024. SynMob: Creating High-Fidelity Synthetic GPS Trajectory Dataset for Urban Mobility Analysis. *Advances in Neural Information Processing Systems* 36 (2024).
- [21] Andreas Züfle et al. 2023. Urban life: a model of people and places. *Computational and Mathematical Organization Theory* 29, 1 (2023), 20–51.