Multi-agent City Expansion With Land Use and Transport

Luiz F. S. Eugênio dos Santos[1], Claus Aranha[2], André P. de L. F. de Carvalho[3]
[1]University of Sao Paulo, lf.santos@usp.br
[2]University of Tsukuba, caranha@cs.tsukuba.ac.jp
[3]University of Sao Paulo, andre@icmc.usp.br

Abstract

Simulating the development of cities is interesting from the point of view of understanding human communities, but also brings benefits to city planners. However, understanding the expansion of land use and transportation networks is a known challenge. In this work, we investigate a city development model that combines a rule-based procedural road generation algorithm with multi-agent simulation of land choice and movement through the map using the generated land use and transportation network. The city map is organized in a grid, and an initial land value for each cell is calculated based on its geospatial features. Next, a set of agents are randomly initialized and perform actions on the city map, such as establishing residences, commuting, and trading in the exploration phase. Then, the actions performed by the agents are used as parameters for recalculating land prices and guiding the expansion of the road network, in a network development phase. We evaluate the emergence of geometrical patterns in the road network as well as land use and population distribution in the final map. We also compare maps generated using geographical data from selected locations to their corresponding real world settlements.

Introduction

Cities are dynamic structures with some resemblance to living organisms, with the need for changes to accommodate future demands, such as new roads. Building new roads often needs to attend to geometrical and resources constraints, as well as seeking to provide better accessibility to land otherwise not well integrated to the current transportation network. For the past years, different methods have been proposed to generate realistic road networks, most notably the early works using an adaptation of L-systems (Müller et al., 2006), later further developed upon and used as the basis for other methods of rule-based road network generation (Kelly and McCabe, 2007; Vanegas et al., 2012; Weber et al., 2009), including the well known software *CityEngine* [1].

With the same goal of replicating the geometry of road networks, other approaches were also proposed such as tensor fields (Chen et al., 2008), templates (Sun et al., 2002; Nishida et al., 2016) and, more recently, deep-learning based methods (Chu et al., 2019; Birsak et al., 2022; Fang et al., 2020).

While some of these works demonstrate that specially machine learning-based approaches can produce more realistic and diverse road networks that are tailored to specific input conditions, they mostly lack adaptability in a dynamic scenario where road development influences the behaviour of residents in a given environment and vice-versa.

This aspect was mostly considered in the form of fixed, pre-generated population density heat maps that acts as a global goal for the road expansion in early works (Müller et al., 2006), but later incorporated in a more complete environment, with land use and zoning simulation that further guide how the roads should be expanded (Weber et al., 2009).

Parallel to that, road network generation also plays a role in artificial city generation as a whole. In this domain, agent-based systems in particular often depict the road generation as a natural result that emerges from the agents' actions and patterns (Song and Whitehead, 2019; Lechner et al., 2006). This, however, produces the opposite effect, often resulting in irregular and unpredictable geometries.

In the current work, we seek to integrate a simple rule-based, easily modifiable road network generator with a multi-agent land use and transportation simulation. For that, we repeatedly iterate through several episodes consisting of an *exploration phase* and a *development phase*.

During the exploration phase, agents representing the citizens of the generated city move around the map choosing one plot of land as its residence and performing a daily work-consume-rest routine, where it commutes using the current road network and changes its behaviour based on past experiences to try to attend to its obligation more efficiently. During this phase, iterations are treated as time steps and the speed of the agents in the network influenced by the density of traffic in each street.

In the development phase, land prices for each cell are recalculated taking into account factors such as their variance in elevation and the value given to them by the agents during the exploration phase. This is in turn is converted into a heat

[1]http://www.esri.com/software/cityengine

map and used to guide the expansion of the road network. Other than that, constraints are considered when building new streets, such as elevation variance and intersection with other streets.

We divide the streets in highways and normal roads, setting different rules for them such that while highways guide the development to a certain direction, normal roads simply grant accessibility to all cells, expanding in a grid-like geometry.

We further discuss the two phases of each episode in more detail in the next sections. Following that, used parameters and emerging patterns are presented and, finally, the results discussed and compared to other known methods. All code used in the present work is made publicly available [2], allowing for easily testing with different parameters and modifications.

Background Work

City generation systems can mostly be divided by the aspects of the urban environment they focus on and the approach chosen for generation. While some of the previous work is mostly concerned about generating realistic road shapes and patterns (Beneš et al., 2014; Nishida et al., 2016; Chen et al., 2008; Sun et al., 2002), others focus on the geometry of blocks and buildings (Weber et al., 2009) or the distribution of different land uses through the city (Song and Whitehead, 2019).

Furthermore, the generation approach can be mainly divided in two classes: *top-down* and *bottom-up*. Top-down approaches are more commonly employed, having is most notorious uses in the very first major works on city generation (Müller et al., 2006; Kelly and McCabe, 2007). Its defining factor is the existence of a "central management", generation of the layout under certain rules with no concern for the actions of the possible residents that would inhabit that environment. While this technique mimics well the process of building a planned city, its shortcoming is the difficulty to build naturalistic roads and paths generated from human use.

On the other hand, bottom-up approaches focuses on modelling the behaviour of the residents, such that the structures and roads are a natural result of their actions. This approach in mostly found in agent-based systems, as seen in previous literature (Song and Whitehead, 2019; Lechner et al., 2006). The main problem with such models comes from the tricky nature of setting constraints and rules to the results, that emerge from actions of the agents in this system, making it hard to build patterns that would be trivial with a top-down approach.

Parallel to city generation and on the topic of modeling the urban environment overall, Land Use and Transport Interaction (LUTI) models address the modeling of cities by studying the behavior and relations of the entities that compose them. The movement of people and vehicles and the paths of commuters are of interest to economists and urban planners, and understanding the organization and behavior of populations in cities is crucial to explain and predict urban problems. LUTI models analyze how residents organize themselves in the city, including where they live, work, and access commercial and leisure establishments, as well as travel costs. New approaches based on cellular automata, dynamic systems, and agents have been developed to meet the demand for more detailed models. Multi-agent techniques are particularly interesting for their ability to model the interaction of agents with each other and with the environment while fulfilling objectives in complex scenarios. Cordera et al. (2017); O'Sullivan (2001); Perez et al. (2017); Bouanan et al. (2018)

In this context, the current work seeks to fill the gap between top-down and bottom-up approaches for city generation, drawing inspiration from multi-agent land use and transport simulations to provide an intermediate between the natural behaviour of people in a city and its development guided by a director plan in the form of a rule-based road network generation and expansion through time.

System Description

We initialize the system using as input a height map either randomly generated or extracted from a given location determined by a bounding box in Web Mercator coordinates. Following that, the rectangle determined by the bounds of the map is divided in a square grid composed of cells of 50 meters in side. Bilinear interpolation is used to sample the elevation of the central points of each cell from the height map, fixing $c.elevation$ for each cell $c \in C$.

As cells are arranged in a rectangular grid pattern, we will denote as c_{ij} the cell in line i and column j of the grid. Moreover, for each cell we save the values $c.x$ and $c.y$ as the Web Mercator coordinates of its central point.

With the elevation of the sampled points in the map, we then compute their matrix gradient and set $c.slope$ as the magnitude of the gradient for each cell. Resulting maps are ilustrated in Figure 1

Finally, we set the initial values of the cells as:

$$c.price = e^{c.slope} 3000 \tag{1}$$

After the initialization, we begin to run of the episodes starting from the *exploration phase*, described thoroughly in the following sections.

Development Phase

The goal for the *development phase* is to build new roads that provide accessibility to land otherwise far from road network. For that purpose, we use a rule-based road network generation algorithm adapted from the model described by Kelly and McCabe (2007).

[2] https://github.com/LFRusso/citygen

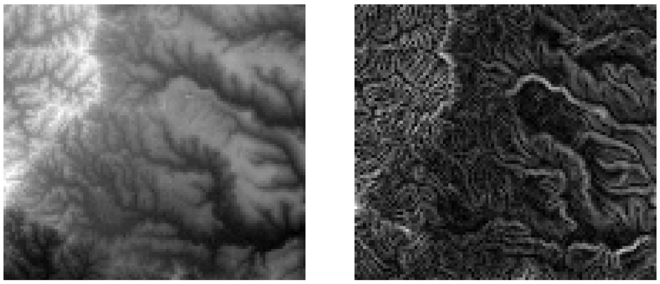

Figure 1: On the left, cells colored based on the elevation of points sampled from the height map and, on the right, the resulting map when colored by the slopes.

As commented before, in most systems that use a similar approach, the *global goals* are generated before the execution of generation algorithm. We introduce the dynamicity of a changing environment by using the values of $c.price$, updated during each episode, to guide the direction of the expansion of the network.

We do that by first taking the matrix P such that each each element $p_{ij} \in P$ is the price of its respective cell in the map. Then, we filter out the cells up to 100 meters from the network, as shown in Equation 2, where $d_N(c_{ij})$ is the distance of cell c_{ij} to is closest road segment.

$$\begin{cases} p_{ij} = c_{ij}.price & \text{if } d_N(c_{ij}) > 100, \\ p_{ij} = 0 & \text{otherwise} \end{cases} \quad (2)$$

Lastly, we smooth the matrix P by applying a Gaussian filter with a kernel of standard deviation $\sigma = 1$.

With the matrix P at hand, we start the road generation algorithm. Much like in similar works, we differentiate roads by *major* and *minor* roads, such that each type has its own generation rules and aims to deal with a different aspect of the transportation system.

Major road segments are generated by expanding the ends of other major roads. Their default segment length is 150 meters and the direction is chosen by taking a sample of 10 equally spaced possible angles in a range of -45^o to 45^o in relation to its parent and selecting the segment that has the greatest average price between its origin and destination points, as seen in Figure 2.

Branching as observed in Figure 3 happens on local maxima, where two major road branches are generated by repeating the process described previously twice for the angle ranges -90^o to -10^o and 10^o to 90^o for each of the two segments.

Minor roads are used improve overall accessibility by connecting more cells to the major roads. They branch off of major road segments perpendicularly and expand in a grid-like shape like shown in Figure 4, branching from other minor road segments on the same way. Minor road segments

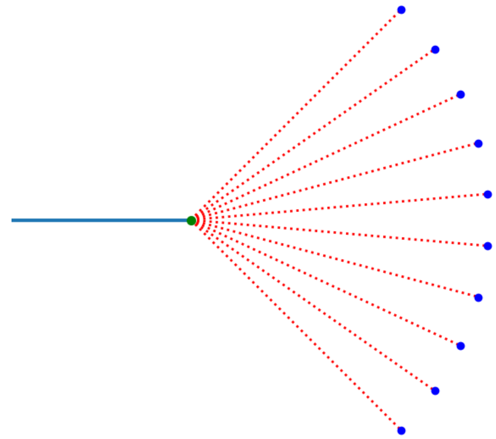

Figure 2: Candidates for next segments on a major road, generated by equally dividing the angle range of -45^o to 45^o in relation to the parent segment.

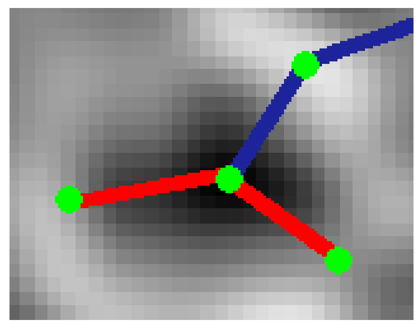

Figure 3: Example of major road branching on a local maxima of the price matrix.

have a default length of 100 meters and have as a condition for their expansion that at least one agent has set its residence in a cell up to 500 meters from the prospected segment.

Other than the previously mentioned generation directives, we further replicate the methods of previous work on rule-based road generation by implementing similar *local constraints* to all roads, aimed to introduce road intersections and guide expansion around some aspects of the terrain such as maximum slope, water and protected areas that are not supposed to be built over. Local constraints are illustrated in Figure 5.

The process of expanding the road network consists of first generating a predefined number of new major road segments. after that, all old nodes (those that were not created by the current expansion of the major roads) are added to a priority queue with priority 0 and checked if they meet the conditions for generating new minor roads. If a minor road is generated, the new node corresponding to the end of the new segment is added to the queue with priority equal to

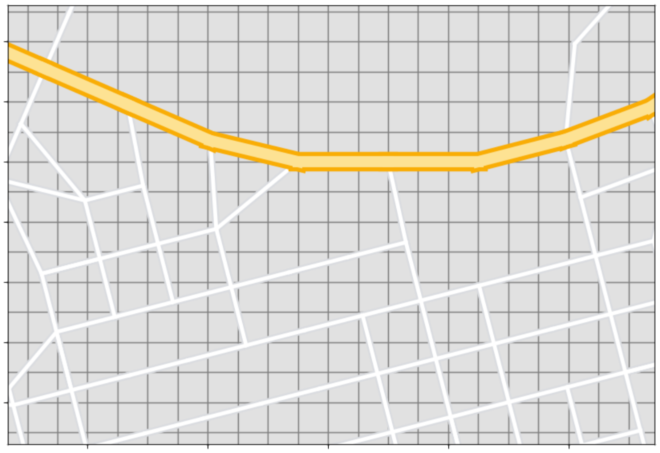

Figure 4: Example of a road network over the cell grid; major roads are represented in yellow while minor roads in white. All roads are expanded according to their own generaion rules and modified by local constraints to fit the network

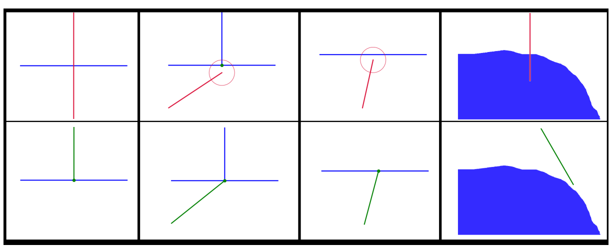

Figure 5: Local constraints used to modify road segments, allowing the creation of new intersections, snapping to existing ones or deviating direction to avoid terrain obstacles. In the figure, images on the top illustrate cases where roads are not allowed to be built and the images directly bellow the modified segment after the local constraints check.

that of the source node of the segment plus one. The whole process is executed until the queue is empty.

Exploration Phase

The exploration phase consists of a multi-agent simulation of the dynamics of trade, transport and land choice within the city. Citizen agents are initialized randomly in the map with a initial score of 5000 points. The closest empty cell to each agent is set as their residence.

After the initialization of the agents, another set of empty cells are chosen to be developed as markets. Markets can have an arbitrary number of *types*, determining the category of good they sell. Finally, each of the new agents have one of the new markets associated to them as their workplace.

Each exploration phase is composed of several *cycles* that represent the days, such that actions taken by the agents consume time and have to be done within the time constraint of the current cycle.

A *cycle* starts with the agents planning the actions that need to be taken to better accomplish the goals for the day. The goals are divided in **work**, **consume** and **rest**:

- The agent has to get out of its residence and commute to its **work**, where it spends 8 hours in the simulation and earns points at the end;
- after leaving work, the agents travels to the closest markets of the selected types (also chosen randomly at the start of the cycle) **consumes**, spending points;
- finally, the agent travels back to its residence and **rests** until the end of the cycle.

At the end of each cycle, all agents are presented with new possible residences to choose from by swapping with their current residence and paying the difference in price.

As mentioned previously in Equation 1, initial cell prices are calculated using exclusively the slope of the heightmap. In subsequent iterations, considering N different types of *commercial* establishments, we set the price of a give cell c as follows:

$$P_c = \frac{7000}{N}\sum_{i}^{N} e^{-T_i(c)} + e^{-c.slope}3000 \qquad (3)$$

In Equation 3, $T_i(c)$ refers to the *expected* travel time between cell c and the closest market of type i. The maximum value assumed by P_c is 10000, in he ideal situation where $c.slope$ and all travel times are equal zero.

On top of the real price of cells, when evaluating a new possible residence, agents also set a subjective score according to the convenience of the cell when considering to their own daily needs. For an agent a, the score given to cell c can be written as:

$$S_{a,c} = P_c + (2 - exp\left(\tau_{a.work}(c)\frac{\ln 2}{1000}\right))1000 \qquad (4)$$

In Equation 4, $a.work$ refers to the cell set as agent a's workplace. $\tau_{a.work}(c)$, in contrast to $T_i(c)$ described in Equation 3, is the *real* time taken by the agent to reach the destination from cell c. While *expected* travel times are inferred exclusively by taking the fastest path in the road network with the constant default speed, the *real* travel time is the actual time experienced by the agent when travelling the roads subjected to the *transport simulation* model, described thoroughly in sequence.

As mentioned before, the road network is composed of a set of road segments and intersections. It can be treated as a bidirectional, weighted graph. Each edge, that represents a road segment, has its weight set as the expected time taken to travel through it. Fixing the default speed for all edges as $v_0 = 50km/h$, we set the weight for edge $e \in E$ with length l_e as $w_e = \frac{l_e}{v_0}$.

However, to encompass the effects of various agents using the same roads, we make the travel speed in a certain segment dependent on the number of agents currently travelling through it at the same time. The travel speed for edge e, currently being crossed by a total of $e.pop$ agents is give by Equation 5.

$$v_e = v_0 \left(\frac{2}{3}\right)^{e.pop-1} \tag{5}$$

During exploration cycles, agents plan their route through their activities of the day using the A^* algorithm to traverse the road network efficiently. After that, the cycle is divided in a total of 36000 iterations, each representing 5 seconds in real time. Each agent moves during the time steps according to the planned route, while subject to the changing speeds as a result of the actions of the rest of the population. Figure 6 illustrates an agent crossing a street segment in those conditions.

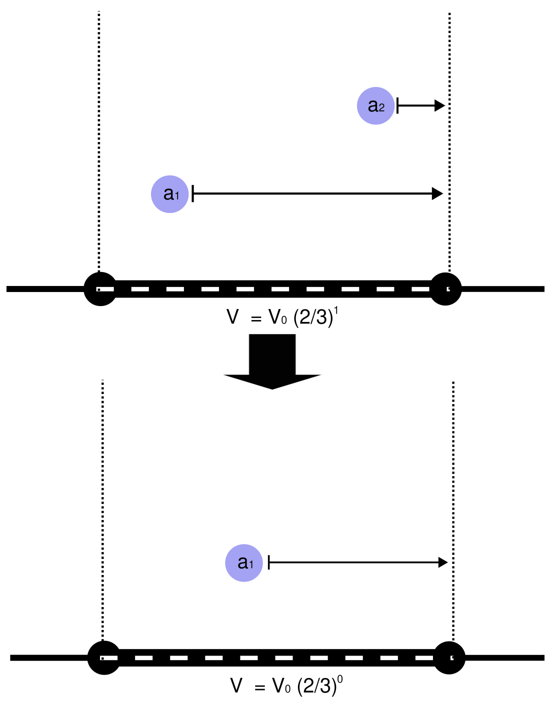

Figure 6: Agents movement through street segment. Positions are updated in every iteration of the cycle based on travelling speed in the current edge, which in turn is calculated based on the number of agents in that segment.

Agents keep track of a shared travel experience graph Γ, initially equal to the road network graph where edges have the default weight $\frac{l_e}{v_0}$. During each cycle, as an agent experiences a travel time τ_e different from what was expected initially while travelling trough edge e as a result of the change in v_e, the weight w_e in Γ is updated as follows:

$$w_e \longleftarrow \frac{w_e + \tau_e}{2} \tag{6}$$

Lastly, we incorporate the market dynamics with the commercial developments of different types through the map. As mentioned, agents work and consume in markets of different types. Each commerce pays its workers and profits from consumers according its type, being the prices and wages manually configured before the execution.

During every cycle, markets loose points when paying for each worker and gain when are visited by agents consuming. After the end of the cycle, if the market looses more money than it gains and the minimum threshold of workers is not exceeded, the worker agent that arrived last is fired and relocated to another commerce in surplus.

All markets are initialized with a total of 5000 points, and if it reached zero are reinitialized in another random location in the map, keeping all previous workers.

The evolution of the system through the cycles of the exploration phase can be seen in Figure 7.

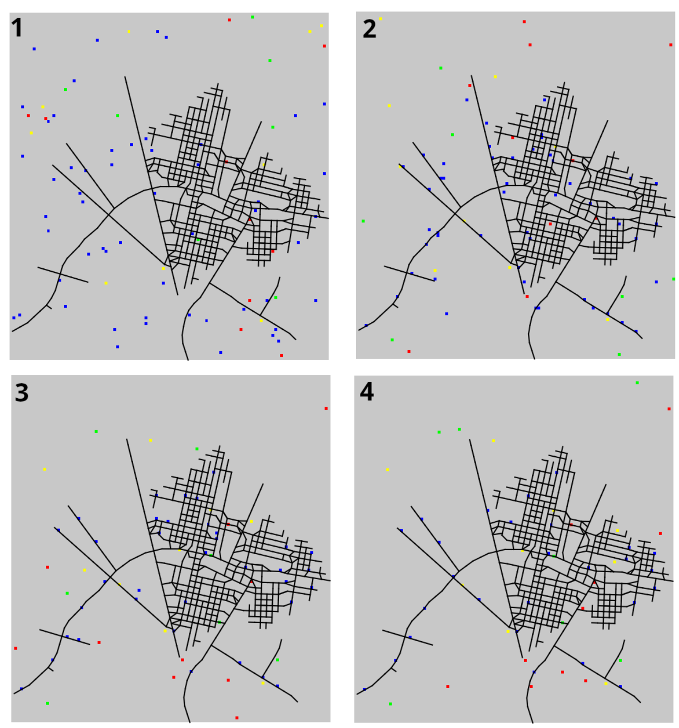

Figure 7: Exploration cycles numbered by corresponding cycle. Agent residences are colored blue while markets of different types are colored yellow, red and green. Agents tend to quickly flock to cells close to the road network due to the higher travel speeds, while markets get relocated randomly until they find regions with more potential buyers, consequently also closer to the network.

Results and Discussions

As it is important to note that all system parameters and conditions are highly configurable, producing with that a variety of possible outcomes, for the sake of visualization we mainly focus on those produced by the default parameters of 3 different market types, having respectively wages of 100, 200 and 300 points for their workers and prices of 20, 30 and 40 points, earned by the market from each visit of an agent in the *consuming* phase.

During the development phase of each episode the road network is expanded to a maximum of 100 sections, while for each exploration phase a total of 50 new agents and 15 new markets are randomly added to the map and their dynamics are simulated for a total of 10 cycles along those previously added. For agent relocation, all cells up to a radius of 250 meters from the current agent's residence are checked as possible candidates for relocation, along with other 5 cells randomly selected from the map for the sake of avoiding local maxima of the scores. Emergence of the results for the set parameters can be seen in Figure 8, outcomes for different elevation maps are displayed in Figure Figure 9.

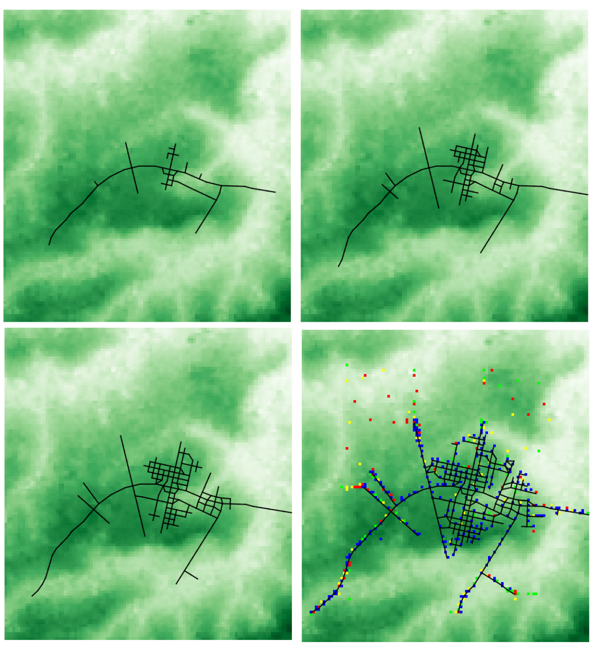

Figure 8: Road network expansion through episodes. Since cell prices are determined by both the variation in elevation of the land and travel times in the map, agents and markets naturally flock to dispersed locations near the network and avoid steep inclines, as observed on the bottom right figure. Moreover, as roads follow land price, the natural shape of the terrain guides the development, as seen on the elevation map.

As highlighted, one of the major advantages of employing a simulation step that incorporates aspects of a bottom-up approach is the possibility to dynamically set goals based on the predicted activities that would take place in the current environment to guide its development. In the present work the defined exploration phase serves this purpose, guiding new developments done in the development phase through the evaluation and devaluation of cell prices according to the agents' action. The evolution of the prices matrix through episodes are illustrated in Figure 10.

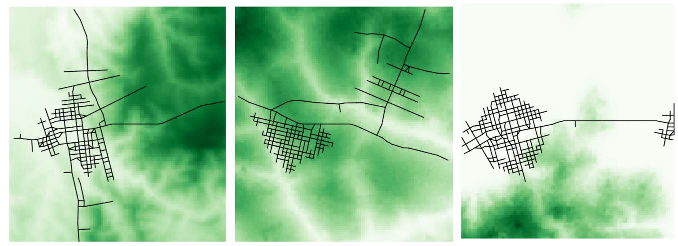

Figure 9: Maps generated on top of different elevation maps. As shown by the images, elevation by itself does not affect land prices and, by consequence, road expansion. The slope, on the other hand, is a determining factor on agent relocation during exploration phase, and as a result new roads are guided away from steep inclines.

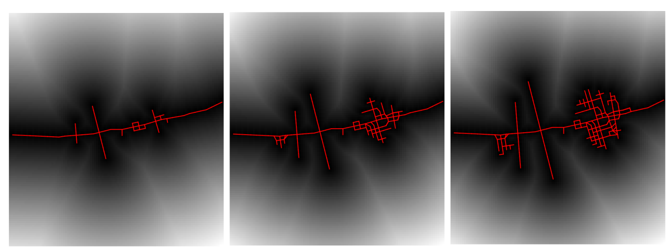

Figure 10: From left to right, evolution of the heatmap relative to the prices of cells in the map. As the road network expands and draws agents near the new segments, cell prices are altered according to the agents' activities. Prices, in turn, guide road expansion to potential valuable locations.

Further evaluations, contrasting population, price distributions and road patterns that emerge from the exploration and development phases to those of real cities built in equivalent geographical locations are currently ongoing, for now limiting the system to its value in the context of procedural generation.

Acknowledgements

This project is funded by the Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), grant 22/02113-6 as well as the Japanese Society for Promotion of Science, grant 22K11918.

References

Beneš, J., Wilkie, A., and Křivánek, J. (2014). Procedural modelling of urban road networks. *Computer Graphics Forum*, 33(6):132–142.

Birsak, M., Kelly, T., Para, W., and Wonka, P. (2022). Large-scale auto-regressive modeling of street networks.

Bouanan, Y., Zerguini, S., and Gaussier, N. (2018). Agent-based modelling of urban land-use development: modelling and simulating households and economic activities location choice. *International Journal of Service and Computing Oriented Manufacturing*, 3:253.

Chen, G., Esch, G., Wonka, P., Mueller, P., and Zhang, E. (2008). Interactive procedural street modeling. *ACM Trans. Graph.*, 27(3):Article 103: 1–10.

Chu, H., Li, D., Acuna, D., Kar, A., Shugrina, M., Wei, X., Liu, M.-Y., Torralba, A., and Fidler, S. (2019). Neural turtle graphics for modeling city road layouts.

Cordera, R., Ibeas, A., Dell'Olio, L., and Alonso, B. (2017). *Land Use-Transport Interaction Models*.

Fang, Z., Yang, T., and Jin, Y. (2020). Deepstreet: A deep learning powered urban street network generation module.

Kelly, G. and McCabe, H. (2007). Citygen: An interactive system for procedural city generation.

Lechner, T., Ren, P., Watson, B., Brozefski, C., and Wilenski, U. (2006). Procedural modeling of urban land use. In *ACM SIGGRAPH 2006 Research Posters*, SIGGRAPH '06, page 135–es, New York, NY, USA. Association for Computing Machinery.

Müller, P., Wonka, P., Hanson, E., Eskuri, N., and Watson, B. (2006). Procedural modeling of cities. pages 139–184.

Nishida, G., Garcia-Dorado, I., and Aliaga, D. G. (2016). Example-driven procedural urban roads. *Comput. Graph. Forum*, 35(6):5–17.

O'Sullivan, D. (2001). Graph-cellular automata: A generalised discrete urban and regional model. *Environment and Planning B: Planning and Design*, 28(5):687–705.

Perez, P., Banos, A., and Pettit, C. (2017). Agent-based modelling for urban planning current limitations and future trends. pages 60–69.

Song, A. and Whitehead, J. (2019). Townsim: Agent-based city evolution for naturalistic road network generation. In *Proceedings of the 14th International Conference on the Foundations of Digital Games*, FDG '19, New York, NY, USA. Association for Computing Machinery.

Sun, J., Yu, X., Baciu, G., and Green, M. (2002). Template-based generation of road networks for virtual city modeling. pages 33–40.

Vanegas, C. A., Garcia-Dorado, I., Aliaga, D. G., Benes, B., and Waddell, P. (2012). Inverse design of urban procedural models. *ACM Trans. Graph.*, 31(6).

Weber, B., Müller, P., Wonka, P., and Gross, M. (2009). Interactive geometric simulation of 4d cities. *Computer Graphics Forum*, 28(2):481–492.