

CityFiction – Scenarios for Densification

Henrik Malm and Petra Jennings

FOJAB

Malmö, Sweden

{henrik.malm,petra.jennings}@fojab.se

ABSTRACT

The aim of the research presented in this paper is to find new ways of understanding and visualising relationships between different types of city data, and thereby support decision making regarding city growth and densification based on desires and values. To this end, a new software tool called CityFiction has been developed. This analysis tool is intended to work as a laboratory for understanding city data, and it will be shown that by interactively weighting, or prioritizing, different measures on the available input data, scenarios for the future development of a city can be efficiently and pedagogically explored. By using the city of Malmö, Sweden, as a pilot study, the tool has been applied to test the placement of e.g. new parks and train stations, and it has aided in the understanding of how the available city data interacts, and possibly conflicts, in the search for a sustainable development of the city.

Author Keywords

Data exploration; data visualisation; city analysis software; city growth; urban development; GIS.

1 INTRODUCTION

The amount of data that we collectively are creating and storing in the world today is doubling every second year, according to a recent analysis by IDC [1]. This enormous wealth of information can be of tremendous value, but can also be overwhelming when it comes to quickly parsing out relevant information and understanding how different data sources correlates, connects or conflicts.

With the increase in the implementation of Smart Cities and Smart Buildings, the rapid rate of data growth is especially true for our urban environments [2]. Here, data collected by sensors and by online devices is used to increase operational efficiency in the city, share information to the public and to improve the quality of government services and citizen welfare. Relating to this is the large increase in mapping of data and the availability of this to the public, like through well-known Google Maps [3] or by more advanced initiatives, such as computer vision driven dense city information mapping [4], or wiki-based open source initiatives [5]. However, most data collected and stored about

our cities is still not openly available and can be both difficult and expensive to obtain, although there exist initiatives to change this [6].

In the current work, the focus has been on city data that can be related to the fitness of an area for potential future urban development. This can be pure geometrical GIS data on the location of e.g. buildings, parks, roads and waterways, but also the density of inhabitants, the plot ratio and the location of different programs and services, such as offices, retail, schools, health centers etc. Sensor data on noise, pollution, traffic etc. is also of primary interest. At least for correlation, economic and social data, such as real estate prices or crime rates can also be relevant. The developed tool is fundamentally open to many different types of data, and can be adjusted to local scenarios and questions. Its big benefit is the capacity to efficiently and interactively find out in which way any kind of input data affects the results. The only requirement on the data is that there is a reasonable way of combining the different data sources to a single fitness value, which describes the suitability of urban development at a specific spatial location, according to the user's preferences.

2 BACKGROUND

The current work was initiated in collaboration with the city of Malmö in Sweden, regarding a visionary city development study on the city. Malmö is Sweden's third largest city and has during the last two to three decades gone through a rapid transition from an industrial harbour city to a city focusing more on education and culture. The city is evolving rapidly and the current project was specifically set up to study how Malmö could grow from its current approx. 300.000 inhabitants to 500.000 during the next few decades, preferably without too much development outside the outer beltway. This poses a major challenge, which calls for inventive approaches to city development and densification.

To this end, we have developed a new type of city data analysis software, which could aid the city planners in exploring different scenarios for developing the city, in order to host almost double its current number of inhabitants.

2.1 Related Work

The approach first researched, for the task at hand, was to create a computer simulation algorithm for city growth. This area has been studied extensively, with pioneering work on mathematical and computational modelling of land-use and urban growth already appearing in the late fifties and through the sixties and seventies [7, 8, 9, 10]. Many authors have since then continued this work, where especially approaches based on Cellular Automata (CA) has been a very active research direction. Influential work along these lines has been published by e.g. Clarke [11, 12] and Batty [13], with the latter also looking extensively at agent-based models and networks for city modelling [14].

Several authors have, along with their theoretical work, also developed and released software packages for land-use simulation, city growth and urban planning such as UrbanSim [15, 16], UrbanFootprint [17], FUTURES [18] and LanduseSim [19]. UrbanSim and UrbanFootprint are comprehensive city planning tools based on large mapped databases (currently mainly focused on the US). UrbanSim is mostly aimed for the real estate market and lets the user evaluate different transportation and land use policies. It also includes procedural city modelling. UrbanFootprint has advanced ways of filtering, visualising and analysing large amounts of available city data, where the analyses includes reports on e.g. transit accessibility or energy use for a set of plots given by the filtering. The user can also choose new programmes, such as parks, high-rise residential etc, for the selected plots and evaluate how this change affects the

analyses. In contrast to these tools, CityFiction is working with general areas instead of sets of plots, and is aiming mostly at larger masterplanning tasks and visions. Specifically, it works by finding suitable areas for new development when the user has chosen a certain weighting (or prioritisation) of different measures on the available data, and shows how new urban intervention affects these results. The weighting could for example be set based on what the city regards as its common values for a good residential area.

In LanduseSim, functions for the fitness for developing different programs at a specific spatial location is approximated, before a CA-based city growth simulation is run, however, it is not clearly documented how this is done in practice. In the current paper, the focus is on the creation and analysis of these kinds of fitness functions. By using the software presented here, we can input a large amount of data, make measurements on the data and then interactively design different fitness functions for different programs (e.g. residential or commercial). By the laboratory style of the program, users can easily understand how the data interacts to create different fitness landscapes. After finding reasonable functions for each studied program, a city growth algorithm could possibly be applied in order to see how the land use develop and change over time. However, this later stage is not the focus of the current paper. Both the creation of the fitness function and the succeeding program distribution simulation has been previously studied by the current authors, but then in the setting of program distribution in a single building [20].



Figure 1. The Graphical User Interface of CityFiction, with the Data Control Interface to the left and the Map Viewport to the right.

Another interesting, and related, area in urban analysis is the field of spatial statistics. Statistics of global autocorrelation, such as Moran's I index, or measures of local spatial association, such as Getis and Ord's $G_i^*(d)$ or Anselin's LISA (Local Indicators of Spatial Association) has been extensively used to analyse urban spatial patterns, c.f. [21,22,23]. These statistics could for example be used to together with CityFiction in order to measure the amount of clustering created by our weighted fitness function, or to compare the output of the fitness analysis with the spatial patterns of the actual current distribution of building programs and land prices. However, these approaches to spatial analysis lies outside the scope of the current paper.

3 THE TOOL

In this section, the developed tool will be presented in detail. The tool is a stand-alone executable program, however, currently there is a pre-processing and configuration step required before the program is run on a new set of data. The current section will start with a brief description of this first step, and then give a full overview of the developed software.

3.1 Pre-processing

The tool works with rasterised map data as input, i.e. pixelated bitmap images. For the calculations to work properly, all data therefore needs to be translated to this format before the program is run. This means that if e.g. GIS (Geographic Information System) data is used, the vector-based format first needs to be rasterised, which can be done within standard GIS packages such as QGIS [24] or ArcGIS [25]. Furthermore, all resulting images needs to be registered to match in scale and location, and have the same resolution and format. Currently, this pre-processing step has been implemented conveniently by a Python-script in QGIS which preforms the rasterization and sets the right format of the resulting images.

The next step is to tell the software to load the correct data as input, and where to find it. This is done by writing a simple configuration file, as a standard text file, which lists the location of each of the input images in question. For each set of input data listed in the file, a few default settings should also be given. These settings will be explained in more detail

below, when going through the workings of the actual program. In order to control the visualisation of the data in the viewport of the program, there is also a possibility to set a color for each input data entry in the configuration file.

3.2 Interface Overview

When CityFiction is started up, the user is met by a graphical interface as illustrated in Figure 1, which shows the program run on map data of the city of Malmö. The two main parts of this window is the Data Control Interface to the left, and the Map Viewport to the right, the latter filling up the main part of the window. In the top left of the Map Viewport, there is a pie chart giving some more detail on the current analysis results. For further information on this diagram, see Section 3.5 below.

3.3 Data and Controls

As mentioned earlier, the aim of the program is to help the user create a fitness function that shows the suitability of e.g. residential development in different areas of the city. This two-dimensional fitness function is shown in the Map Viewport as shades of brown, where a lighter shade means a higher fitness, c.f. Figure 2 where the map information is removed and only the 2D fitness function is shown. At the top of the Data Control Interface, the 2D fitness function $F(x, y)$ is interactively constructed by the user. Let

$$d_i(x, y), \quad i = 1, \dots, N$$

be N different 2D data sets loaded into the program. Let then $M(d)$ be a measurement function on the data d . A simple and common example of a measurement function is the minimum distance function, let us call this $M_D(d)$. Let the data $d_i(x, y)$ be a binary image marking the location of a specific geographic area, such as the sea, or the location of point objects, such as train stations. $M_D(d_i(x, y))$ is then constructed by calculating the smallest distance in every point (x, y) to this area or set of points. This value is then inverted and normalised to a value between 0 and 1, so that M_D has the maximum value of 1 closest to the area or point in question. The fitness function F is then the weighted sum of a number of these measurements on the data.

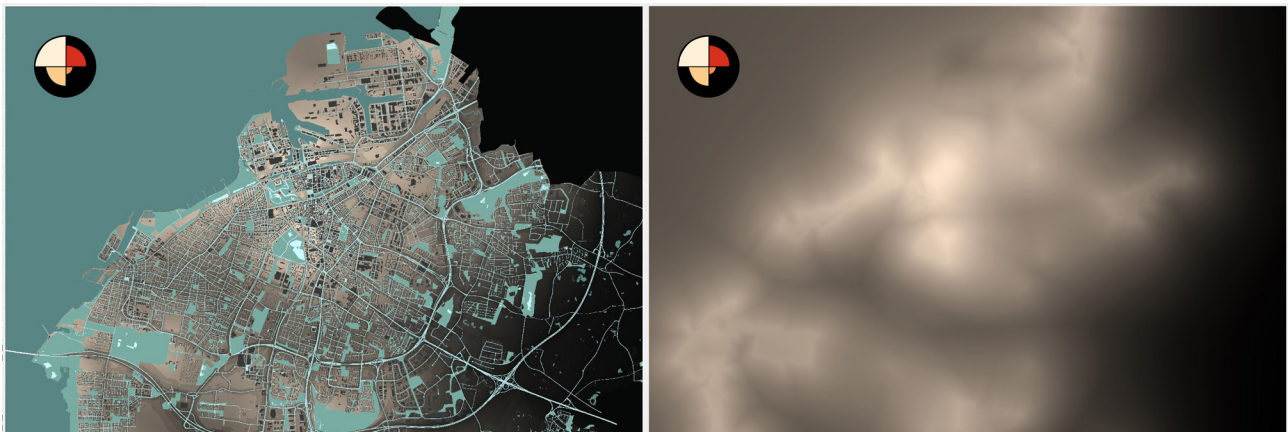


Figure 2. If the map data showed in the left panel is removed, we can view only the underlying fitness function, as in the right panel.

Let

$$w_i(x, y), \quad i = 1, \dots, N$$

be N different weights, where $0 < w_i < 1$. Then, the fitness function is constructed as

$$F(x, y) = \sum_{i=0}^N w_i M(d_i(x, y)) .$$

In the Data Control Interface, the loaded input data in the current execution of CityFiction is listed consecutively, see Figure 3. Each set of loaded data is also marked with a unique color. In the section for each data item i , the corresponding weight w_i is set by a slider called *Weight*, or by the spin box next to it. In the current version of the program, there is also a slider called *Fall-Off* for each loaded data set. If the minimum distance measure is used as measurement function (M_D), the *Fall-Off* slider controls an exponential transformation of this measurement, so that it can be set how quickly the distance measure should fall off to zero. The higher the value, the quicker the fall-off. This is, for example, useful if we want the distance to the sea to affect the fitness function up to approx. a distance of 2km, while a smaller park only affects the fitness up to a distance of, say, 300m. The default value of the sliders in this section is set in configuration text file, as mentioned earlier. Future developments of CityFiction could also include an easy possibility to add a sharp distance threshold, i.e. so that the fitness value is 1 (normalized) up to the threshold and 0 at larger distances. Another possibility is a feature to let the user set the fall-off function more generally, via a manipulating a graph function in the GUI.

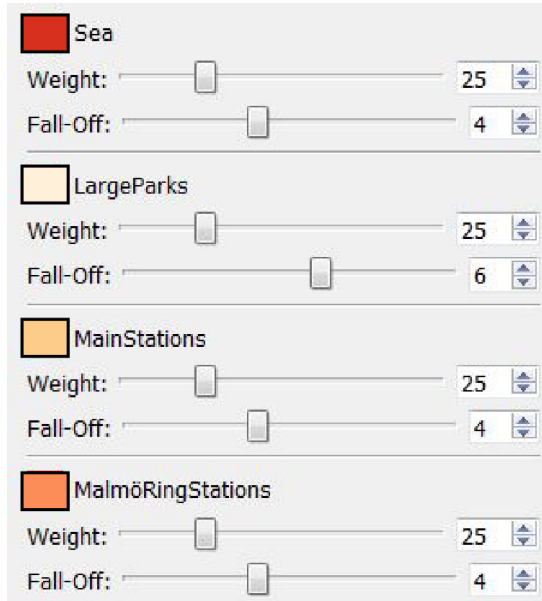


Figure 3. The area of the Data Control Interface where the weights are set in order to construct the fitness function.

By pressing the button *Show Highest Fitness Areas*, further down on the Data Control Interface, the areas with the highest fitness according to the weight settings, are marked with a white field. By changing the value of the slider *Area Size*, we can set how many square kilometers that are marked in white, c.f. Figure 4. As can be noticed, the white area is partially transparent, and this transparency value actually has a quite important meaning. If we consider the complete white field as totally buildable, this area would be totally opaque white. However, this field is also covering areas representing e.g. water, buildings, parks and streets. For maximum flexibility, it is possible to set the probability of development, also in these areas, to larger than 0. For example, while densifying the city there is a large possibility that some of the new development will be add-ons to, or redevelopment of, existing buildings. Therefore, it is reasonable that the possibility of development at the locations of the current building stock is larger than 0. The sea might, for example, seem like a less obvious area for development. However, reclaimed land has historically been used for development to a quite large degree in sea-side cities, so it not totally unreasonable to also consider water area as an area possible for development, to some degree.

The degree to which areas such as the sea or the location of current buildings should be considered for development is set using the sliders located below the list of sliders for the fitness function. The geographic area data listed here is also loaded when the program is started up, and should therefore also be listed in the configuration text file. By setting the slider value called *Development Probability* for each data item listed here, the possibility of building in the area represented by this data is given a value between 0 and 1. This also controls the transparency with which this data item is shown in the white field representing the highest fitness values on the map.

Among the data listed in this section of the Data Control Interface, we could also include non-binary data, as we have done in our current tests by including noise measures in the city. This data should be normalized to be in the range 0 to 1 and then affects the development probability multiplicatively, so that there is a smaller probability of development in a high noise area. The distance measures presented in the beginning of this section is, however, only applicable to binary data.

Below the *Development Probability* slider for each of the data items above, there is a slider named *Smallest Distance*. This is used in order to set a “buffer area” around the areas covered by these data items. For example, it might not be possible to build a new building closer than 8 meters to an existing building, unless it is considered as an add-on. Further, it might not be reasonable to build a new building closer than 10m to the shoreline. To this end, the *Smallest Distance* slider can be set, in order to offset these listed areas with a distance value.



Figure 4. The white field showing the area with highest fitness. When the area is increased the field grows and morphs into new shapes.



Figure 5. A new area is drawn in green as a polygon in the Map Viewport. This area is then added to the loaded park data which directly changes the fitness function and the white field showing the area with highest fitness.

3.4 Drawing and Adding New Areas

One feature of CityFiction is that areas or point data can be added interactively to existing loaded data in the program, and the effect of this addition can be seen directly on the fitness function. Let's say that a particular area in the city has quite a low fitness for residential development since there is no larger park situated close enough to this area. By then drawing a new park in the Map Viewport we can directly see how this park affects the weighted fitness for building in this area, cf. Figure 5. The drawing is performed by clicking a number of times on the right mouse button in different locations, in order to place the corners of a polygon. If a clicked location is close enough to the starting point, the polygon is automatically closed and filled. In a multiple choice menu in the Data Control Interface we can then choose the loaded data type (parks, water etc) which the area should be added to. By then clicking the button *Add Area*, the area is added to the chosen data type. The fitness function is then directly updated in the Map Viewport, incl. the white field indicating the area with highest fitness value.

3.5 The Pie Chart

The brown shading in the *Map Viewport* shows one value for each location, which represents the weighted fitness. However, in many cases it is of interest to see the specific

composition of the fitness in a particular location. We might, for example, be interested in seeing how large part of the fitness value in this location is related to closeness to parks and how much is related to the closeness to a train station. This is achieved by the pie chart in the *Map Viewport*, c.f. Figure 6. This diagram has as many circle sectors as the number of data items loaded for the fitness calculation. If the number of data items used increase, the pie chart might get a bit too cluttered to read. Currently, there is a limit on 10 sectors in the chart, and if the calculations include more data than this, only the 10 measures with the highest weights are shown in the chart. Further, the circle sectors are coloured according to the colour markers next to the data items listed in the *Data Control Interface*, cf. Figure 3 and Figure 6. When hovering with the mouse over a location on the map, the radius of the sectors is changing according to the value of the different data measures. That is, if the location is very close to the sea, the radius of the sector relating to the sea data is almost at its maximum in the diagram, while if we are very far from any station, the radius of the sector relating to the station data is very small.

The current weighting of the different data measures can also be read in the diagram. If we, for example, increase the *Weight* slider for the sea data, the angle for the sector representing this data is increased accordingly. In order to

see this change in the pie chart properly while changing the slider, we first need to lock the pie chart for a specific location. This is done by clicking the left mouse button at that location in the *Map Viewport*. A red dot will then appear on the map and the pie chart will be locked to this location until the left mouse button is clicked again, releasing the diagram, cf. Figure 6. In future versions of CityFiction, the possibility of marking a larger area, calculating the mean values in this area and then plotting these mean values in the pie chart will also be implemented.

3.6 A Word on the Implementation

As mentioned, CityFiction is a stand-alone executable program, currently only running on the Window operating system. The program is written in C++, using the Qt SDK library to create the user interface, c.f. [26], and the OpenCV library for the image processing, c.f. [27]. OpenCV is a very powerful resource, which for example made it possible to calculate distance measures in real-time through its Distance Transform filters.



Figure 6. The Pie chart is here locked at the location of the red dot in the map. This point is located in a park and also relatively close to a train station, as the diagram shows.

4 DISCUSSION AND FUTURE WORK

We believe that the analysis software presented in this paper, already in its current state, represents a powerful tool for understanding city data and how it interacts. By changing the weights, or priorities, of the different sets of loaded input data, and by hovering over different locations on the map with the mouse, we can quickly identify locations that are well-suited for development. In contrast, we can also understand what factors are missing at a location for it to be suitable, according to our preferences. Changing the sliders, and simultaneously seeing the white field, which marks the highest fitness area, morph from one shape and location to another is a very pedagogical experience. Also, the ability to draw in new data “on the fly” makes it possible to further test different scenarios for city development, for example to test the location of a new train station or a new park. Generally, we have discovered how important real-time feedback is in

this type of tool and how educational the above mentioned interactivity is. Increasing the resolution and adding more sets of data to the calculation, of course, affects the interactivity, with the complexity scaling to the power of two with the resolution and the linearly with the number of mapped items in the data set. However, running on a mid-range laptop, interactivity has been shown to be working well with a raster resolution of 1600x1200 pixels and up to 10 different data maps in the calculation.

Interestingly, by prioritising the different data measures in a way that feels relevant for residential development, we can in the current project see a rather big correlation with the actual future development plans for Malmö. However, it is of course also interesting when the data interact in ways that wasn’t expected, and when the tool makes us realise aspects of the data that was not obvious from the beginning.

Generally, our discussions with the Malmö planning office has been very beneficial in developing CityFiction. We envision the tool to function as a platform for discussion among different stakeholders in the city’s future (citizens, politicians, planners, etc.) about desires and values and how the relate to city development and densification. This could lead to agreed and common values from where reasonable weighting of the data and measures used by CityFiction can be set as default and the resulting suggested areas for development, based on these values, could easily be visualised. The users could then also quickly find out how a new intervention, such as a new large park, affects these results.

A strength of the software is, however, that these common values can be changed, and that the result of this change is directly visualised by the program. Working with another city, maybe in another country, might e.g. also lead to a quite different set of common values.

4.1 Future Work

There are many ways in which this tool could be further developed. For example, future work could include the ability to in parallel make different settings for a number of different programs, i.e. residential, commercial, offices etc, and overlay the best fitness areas for each program simultaneously in the viewport. This would require that the *Data Control Interface* is made more compact, with different pages for different program setting etc. A more compact interface would also make it easier to load in a larger number of data sets than is currently used.

The usability of the software would also be greatly improved with a tighter association to available GIS software packages. Ideally, the program would be developed as a plug-in to e.g. ArcGIS or QGIS [24,25]. The selection of data to be used in the CityFiction analysis could then easily be done in the main program’s layer window, while the rasterisation would be performed inherently by the GIS package’s internal rasterisation functions.

Another future development of the software could include more intelligence in the analysis results by e.g. generating relational impacts. Greater development in an area might, for example, lead to over-crowding of transit and under-development lead to higher risk of crime. Identifying these kind of negative impacts might impart significant knowledge to the user, if this addition is based on well-documented research findings.

By connecting CityFiction to the research on city growth algorithms presented in Section 2.1, the software could also be used as a first step to set up relevant fitness functions, before a city growth (or land-use) algorithm is run using these fitnesses. The applied growth algorithm could e.g. be based on Cellular Automata, where different programs would compete about the available developable areas, where this competition would be affected by the fitness settings.

Another aspect that needs some more study is the question on how to best approximate how large area needs to be developed in order to reach a certain number of new inhabitants. As detailed in Section 3.3, the size of the white field marking the area with the highest fitness for development is currently set using the *Area Size* slider. But how many new inhabitants could actually fit in this area? Currently, this is approximated by setting a maximum density, say 10000 persons per square kilometer, using the *Max Density* slider in the Data Control Interface. This density will then be true for the opaque white areas, while the density in the transparent areas are multiplied with their transparency value between 0 and 1, i.e. with their probability value for further development. This means that if the probability value for development at the location of current buildings is set to 0.2, it is approximated that on average 2000 persons per square kilometre could be added by redeveloping, or adding on to, the current building stock. A more sophisticated way of approximating the potential for increase of number of inhabitants is probably needed, based on area regulations on building height, current availability of infrastructure and services etc. Further, in order to properly study how much new development that could actually be done in a certain area, we would really need to zoom in on this area and work on a different, more detailed, scale. CityFiction could then be used to roughly map out relevant target areas for development, which then could be studied further by a more detailed tool. This second tool would then need to work in 3D in order to study possible massings and by making analysis on daylighting etc. However, this stretches far outside the topic of this paper.

5 CONCLUSION

In this paper, a new city data analysis tool called CityFiction has been presented in detail. The tool has been developed in order to aid in decision making regarding city growth and densification. By prioritising between different measures on the available input data, scenarios for the future development of a city can be efficiently and pedagogically explored and visualised. The user can also add new data interactively to

the analysis by drawing in the viewport. In this way, the effect of adding a new train station in a specific location, or adding a new park, can analysed “on the fly”. Using the tool for a visionary city development project for the city of Malmö in Sweden, we have found it to give very reasonable outputs and be very educational in helping to understand how different data sets interacts in order to create a measure on the suitability for development and densification in the city.

ACKNOWLEDGMENTS

The authors would like to thank the City Planning Office at Malmö City Council for great consultation, an inspiring target project and for the GIS data used to test the developed application. We would also like to thank FOJAB for financing the research and for great discussions with the colleagues at the office regarding the project, not at least with David Kiss. Finally, the authors would like to thank Jens Jul Christensen for contributing substantially in the early stages of this project.

REFERENCES

1. Digital Universe of Opportunities: Rich Data and the Increasing Value of the Internet of Things. *EMC Digital Universe with Research & Analysis by IDC*. <https://www.emc.com/leadership/digital-universe/2014iview/index.htm>. As of 21 Nov 2018.
2. Woetzel, J, et.al. Smart Cities: Digital Solutions for a more Livable Future. *McKinsey Global Institute*. McKinsey&Company. June 2018.
3. Google Maps, Goggle Inc., <https://www.google.com/maps/>. As of 21 Nov 2018.
4. Mapillary, Mapillary AB, <https://www.mapillary.com>. As of 21 Nov 2018.
5. OpenStreetMap, <http://www.openstreetmaps.org/>. As of 21 Nov 2018.
6. Berends, J, et.al. Analytical Report 6: Open Data in Cities 2, *European Data Portal*, June 2017.
7. Lowry, I.S. A Model of Metropolis. Report RM-4035-RC. RAND Corporation, Santa Monica, CA, USA, 1964.
8. Tobler, W. A Computer Movie Simulating Urban Growth in the Detroit Region, *Economic Geography* 46, 2 (1970), 234-240.
9. Batty, M. Modelling Cities as Dynamical System, *Nature* 231, 5303 (1971), 425.
10. Batty, M. Recent developments in Land-use Modelling: A Review of British Research. *Urban Studies* 9, 2 (1972), 151-177.
11. Clarke, K.C., et.al, A Self-Modifying Cellular Automaton Model of Historical Urbanization in the San Fransisco Bay Area, *Environmental and Planning B*, 24 (1997)

12. Chaudhuri G., Clarke, K.C, The SLEUTH Land Use Change Model: A Review, *Int. J. Environmental Resources Research* 1, 1 (2013).
13. Batty, M., Cities and Complexity: Understanding Cities with Cellular Automata, Agent-Based Models, and Fractals. MIT Press, Boston, USA, 2005.
14. Batty, M., A New Science of Cities, MIT Press, Boston, USA, 2013.
15. Teerarojanarat, S., et.al., Urban Growth Simulation with Urban Sim, *Proc. FOSS/GRASS Users Conf.*, Bangkok, Thailand, Sept 2004.
16. UrbanSim, <http://www.urbansim.com>, As of 24 Nov 2018.
17. UrbanFootprint, <http://urbanfootprint.com>, As of 6 Feb 2019.
18. Petrasova, A., et.al., Open Source Approach to Urban Growth Simulation, *Int. Arch. Photogrammetry, Remote Sensing and Spatial Information Sciences*, Prague, Czech Republic, July 2016.
19. LanduseSim, <http://www.landusesim.com>, As of 24 Nov 2018.
20. Malm, H. and Nicholas, P, Re-use and Adaptation of Existing Building Structures using Monte Carlo Simulation, *Proc. Building Simulation and Optimization 2014*, London, UK.
21. Pàez, A, and Scott, DM, Spatial Statistics for Urban Analysis: A Review of Techniques with Examples, *GeoJournal* 61, (2004), 53-67.
22. Getis, A, and Ord, JK, The Analysis of Spatial Association by the Use of Distance Statistics, *Geographical Analysis* 25, (1993), 276.
23. Anselin, L, Local Indicators of Spatial Association – LISA, *Research paper 9331*, Regional Research Institute, West Virginia University, Morgantown, 1995.
24. QGIS, <https://www.qgis.org>, As of 26 Nov 2018.
25. ArcGIS, <https://www.arcgis.com>, As of 26 Nov 2018.
26. Qt, <https://www.qt.io>, As of 25 Nov 2018.
27. OpenCV, <https://opencv.org>, As of 25 Nov 2018.