UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR IT-SICHERHEIT

Precision of city simulations using publicly available city data

Präzision von Städtesimulationen anhand von öffentlichen Städtedaten

Bachelorarbeit

im Rahmen des Studiengangs
IT-Sicherheit
der Universität zu Lübeck

vorgelegt von
Kevin Richter

ausgegeben und betreut von
Prof. Dr. Esfandiar Mohammadi

mit Unterstützung von
André Gode, MACH AG
Prof. Dr. Moreen Heine

Lübeck, den 10. März 2021

Abstract

Smart city is an increasingly important concept when it comes to city design, however, planning and upgrading to this very concept remains a challenge with many uncertainties involved. To minimize such uncertainties it is important to meticulously predict how changes to a city-structure will influence the life of its population. One way to achieve that would be through the use of machine learning, however when the topic of training these machine learning models comes with heavy data privacy concerns. A way to circumvent this problem would be the use of synthetically created real-world data. In this work, the goal is to evaluate various city simulation frameworks regarding their potential for generating synthetic data. The choice fell on the city-builder Cities Skylines. It was then determined how to efficiently create a model of a real-world city in the space of the game and evaluate the simulation with publicly available real-world data. It was found that Cities Skylines holds the potential for generating valid synthetic data due to its relative out of the box accuracy to real-world trends but also customizability of the simulation through the use of third-party game modifications.

Zusammenfassung

Smart City bezeichnet ein Konzept des Städtedesigns, das zunehmend an Relevanz gewinnt, jedoch gestaltet sich die Planung und Durchführung von Änderungen in Richtung dieses Konzepts oftmals als eine Herausforderungen die von zahlreichen Unsicherheiten untermauert wird. Um diese Unsicherheiten zu minimieren ist es wichtig genau zu prognostizieren wie sich Veränderungen der Stadtstruktur auf das Leben der Einwohner auswirken kann. Eine Möglichkeit dies zu erreichen wäre durch die Nutzung von Machine-Learning, jedoch ist das Unterfangen solche Machine-Learning Modelle zu trainieren gespickt Bedenken bezüglich Datenschutzes. Um dieses Problem zu umgehen wäre es möglich synthetisch generierte Datensätze zu nutzen. Das Ziel dieser Arbeit ist es verschiedene Stadtsimulationsframeworks hinsichtlich ihres Potentials für die Erstellung von künstlichen Daten auszuwerten. Die Wahl des Frameworks fiel auf den city-builder Cities Skylines. Anschließend wurde erörtert wie man ein Model einer real existierenden Stadt am effizientesten innerhalb der Spielumgebung kreiert und zusätzlich die Simulation mit öffentlich zugänglichen Datensätzen verifiziert. Die Erkenntnis wurde getroffen, dass Cities Skylines großes Potential für die Erstellung von gültigen, synthetischen Daten-

sätzen innehält, einerseits durch die relative Ähnlichkeit zu realen Trends aber auch durch die Anpassbarkeit der Simulation durch third-party Spielmodifikationen.

Erklärung

Ich versichere an Eides statt, die vorliegende Arbeit selbstständig und nur unter Benutzung der angegebenen Hilfsmittel angefertigt zu haben.

Lübeck, 10. März 2021

Contents

1 Introduction and Motivation

We live in a society where people are continuously drawn towards living in large cities, thus it is of the essence to make sure that a city is not outgrown by its population and new ways to keep the inner operations of large cities going need to be found and implemented. That is however an increasingly more challenging task since in a modern city a wide variety of ongoing processes are at play constantly. The concept of smart city aims to realize a process of modernization and obtain maximum possible efficiency.

The problem with modernizing a city to smart city standards however is that there is a lot of unpredictability associated with the procedure. For example, if parts of a street are renewed and thus closed off for an extended time period it is hard to predict how the traffic flow of the city will change during that time which could lead to a rise in noise- and air pollution levels in other parts that would be deemed unacceptable. Of course, it would be unwise to simply go into modernisation blind let fate decide whether it will turn out well, thus a way to predict a city's behaviour upon such sudden changes must be found.

One such way would be to train machine-learning models with the data of a city's inhabitants to make accurate predictions concerning sudden changes, however, this approach comes with heavy concerns in regards to data privacy since there exist attacks against machine learning models that isolate the training data such as in [SSSS17]. Another approach would be using software designed to emulate the behaviour of cities and plan changes according to a simulation, such a simulation could also be used to avoid the aforementioned privacy leakage by providing data without having to use personal data.

Such software has been existing even in a more casual manner since 1989 when SimCity[Inc89] was the first video-game to manifest the city-building game genre. City-building games have evolved a lot since then and it has become worth thinking about employing such city builders which are originally designed for recreational purposes as a basis for smart city planning. However, there remains the problem of choosing a simulation framework that is most suited for the specific use case of planning smart city. While there have been cases of avid users trying to recreate their home-towns in simulations such as the aforementioned SimCity or other games such as the popular and fairly recent Cities Skylines it remains to be seen how well these emulated cityscapes perform when it comes to approximating real-world behaviour.

Contribution

In this work first steps are made to create a model of the city of Lübeck in Cities Skylines. Publicly available, manually curated and pre-processed data was imported to the game framework to create such a model. Then an evaluation of this first iteration of a simulated Lübeck took place. It was found that while the simulated traffic showed promise the predicted pollution only loosely correlated with data gathered from actual air quality measurement stations. Several future directions concerning how to extend Cities Skylines into a serious simulation were found with regards to the games extensive customizability through the use of third party mods.

Outline of this thesis

In the first part of this thesis, several city simulation frameworks are being examined and reviewed concerning their suitableness for this project. Afterwards, key parts that helped with setting up the simulation are explained. Then the move into the setup phase is made, where the methods, problems and solutions of said problems that came when creating the simulation are explained. Furthermore, the Experiment section covers both the collection of validation data as well as the evaluation of this project's findings. In the final section, an outlook is given towards what future pieces of work could be able to achieve with regards to either fine-tuning existing methods or creating new ones to extend the functionality of this framework and simulation.

2 Background Info

2.1 Searching for a good simulation tool

Throughout this project, a plethora of city simulators was examined when trying to find the one that would be most suited for this project. The simulation that should be used in the experiment had to fulfil a certain set of criteria, first of all, it should cover a variety of smart city-related aspects such as waste management, traffic, health and energy management to only mention a select few. Furthermore, the simulation needed to allow a kind of ”free-form” mode because already existing cities needed to be modelled as accurately as possible without any limitations significantly hindering the process. Finally, the availability of tools to help with the construction of a real-world city in the space of a simulation was of relevance for efficient conduction of experiments. The simulations that did not make the cut are listed here and the reason why those particular simulations were not ultimately chosen will be given.

First Open Source Micropolis[Sim] a self-proclaimed spiritual successor to SimCity based on the original SimCity published in 1989 was considered. While it stays true to the original game the grade of gamification of this simulation is too much to be considered for a proper scientific experiment concerning the simulation of smart city aspects. The game places a lot of value on being able to create your own cities from the ground up as opposed to making the conversion of a real-world city into the simulation feasible.

The next simulation that was considered is OpenSC2k[Ope]. In concept, it seemed like a promising project that could have been used for the experiments, however during the time of looking for a suitable simulation the repository information stated that the software was not in a usable state, the uncertainty of whether this would change in a timely fashion quickly disqualified it for use in this project.

Another framework that seemed very promising for smart city projects was SCSSimulator as part of [SBKM] by Santana et al. This simulation aimed at correctly displaying smart city relevant aspects, however, it was only an early prototype before it was discontinued and reworked into a much more specified software, thus making it unsuitable for the intent of this project.

InterSCSimulator[Int] is that more specified software that derives from the previously mentioned SCSSimulator. It mainly focuses on observations and examination regarding traffic. Despite being very accurate just one aspect of smart city is simply not enough to

be considered in this project thus we deemed InterSCSimulator to not provide enough for these specific purposes.
3d.city[3dC] was another consideration, it serves as a 3D version of the previously mentioned Open Source Micropolis. It features pretty much the same amount of potential and limitations and based on its predecessor also being deemed a framework unfit for this project the choice was made to not consider this one any further.
During the search for a suitable simulation framework, one with the name of citybound[aep] was found, a very ambitious project aiming to eventually simulate the machinations of a city down to every last citizen. Unfortunately due to the project being as high-reaching as it is, it is still very much an early prototype and not in a usable state to approximate and anticipate smart city developments.
The last software that was examined other than Cities Skylines was Smart City Plan[Gam20]. A software that aims to simulate the creation and administration of a modern smart city. While sounding very promising it quickly became very apparent that this software aimed more at being a game than being a simulation due to bad scaleability and very simplified systems ultimately rendering it unusable for these experiments.

2.2 Cities Skylines

2.2.1 Cities Skylines

The choice of our framework fell to Cities Skylines developed by Colossal Order[Ltd15] and published by Paradox Interactive in 2015. It features a variety of observable features that are relevant to smart city planning, it has also received praise for its realistic portrayal of traffic flow and could serve as a great basis to observe the impact of structural changes on a city. It also features an API that allows changes to be made to several calculations and behaviours present in the simulation framework. This API is also often referred to as the modding API.

2.2.2 Mods in Cities Skylines

Modification, often abbreviated as modding, has a long history in the space of videogames. Players have often modified games to more to their liking, being that in- or decreasing difficulty, making visual changes or even making them more realistic. Mods have a long history of bringing games to their limits and sometimes even break them. The idea is now that the existing simulation framework of Cities Skylines can be modified in a way that it accurately applies to real-world data if it does not already. Then further changes to the simulation can be made to predict the behaviour of the city when reacting to those changes. Furthermore, mods can be used to make the process of creating a simulated city in the likeness of a real-world city significantly easier without sacrificing accuracy. Additionally, the process of running the simulation can be simplified by using mods that mitigate unrealistic simulation behaviour originally meant to serve as a gamified challenge to a city builder game such as the absurd frequency of fires or the fact that electricity cannot be imported.

2.2.3 Geo Skylines

The mod used in this experiment is Geo Skylines developed by Pinos et al. throughout [PVP20], it aims to recreate cities accurately through the use of available real-world data via sets provided through public services such as OpenStreetMaps. With those datasets, a relatively realistic depiction of Lübeck has been achieved throughout this project. The process of transferring data is described in the "HowTo" document that comes with the download of the mod.

2.2.4 Validation Data

The data used for later on validating the accuracy of the simulation was obtained through publicly available sources. Concerning traffic an attempt was made to collect traffic incidents through the Bing-Maps API[bin], however, due to the global COVID-19 pandemic the general trends had been heavily impacted due to generally less traffic taking place, thus a month of collecting traffic incident data yielded barely any usable data, so instead the traffic layer of our simulation was compared to the traffic layer provided by the Google-Maps[goo] service. To collect data regarding air pollution the website openSenseMap[OSM] was used which provided data from two sensors stationed in Lübeck. For this experiment, the arithmetic hourly means for PM2.5 and PM10 measurements over one month were used.

3 Related Works

Juraschek et al.[JHT17] showed that city simulations can be used in principle to observe aspects of urban production in city districts. They attempted to create a rudimentary model of the urban district of Braunschweig in various similar seeming simulation frameworks. In their findings, they note that through the use of modifications it can be possible to modify the City Skylines software in a way that benefits the purpose of scientific data gathering. However, they only allude to the fact that Cities Skylines proves to be a potential bearing framework. In this work, the goal is to take these findings one step further and verify potential accuracies of a simulation as well as observe on a larger scale than one urban district.

Olzweski et al.[OCSW20] in their 2020 work managed to create a service that would prove more accurate than geo skylines, the mod used in this experiment; however, their work only functions with geographical data located in Poland. Nevertheless, it is a good indicator for what needs to be achieved in the long run with this and future projects as well as the potential Cities Skylines holds as a framework for serious simulations.

In [Sot10] Sootama discussed how mods can be applied to existing games to expand the possibilities of them. In the work it is also mentioned how mods can be used to employ more realism to a game and how much research can go into a mod. However the focus of that particular work lies in examining the people who create mods rather than observing modifications themselves.

Haahtela et al. [HVK+15] examine how Cities Skylines can be used as an educational tool and verifies that it features well modelled functioning city systems. They also mention the fact that modification of the game could be a viable tool for educating upon the topic of city planning. In this work, the goal is to verify this notion with regards to researching purposes concerning smart city.

In [Ber16] Bereitschaft argues that while city building games currently do not reflect the real world accurately the thrive off of active communities. It is however not discussed in depth how these communities can help improve unrealistic behaviour of a game due to the way it was created by the developers.

During [Ear20] Earle is employing fractal neural networks to optimize certain aspects of the game Micropolis which is based on the original SimCity. While this is not particularly interesting for this experiment it shows that such methods might be used in the future when it comes to planning out certain aspects of an existing city-scape by using a simulation.

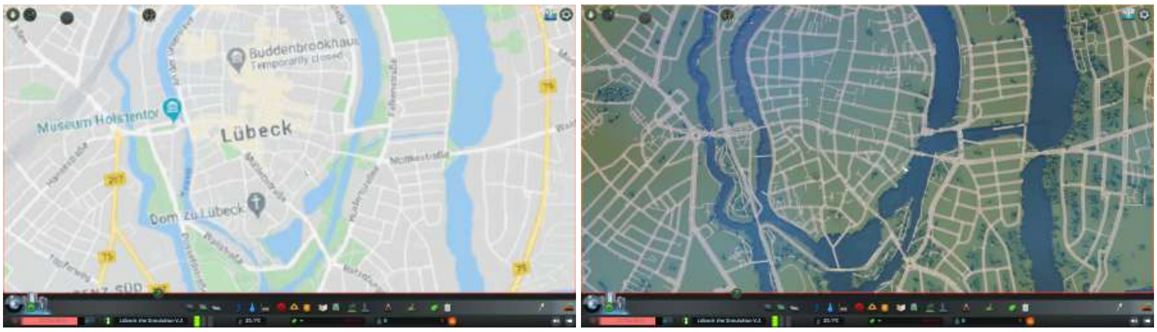
4 Setup

4.1 Mods used to set up the simulation

A variety of mods was also used to simplify the process of setting up and running the simulation by mitigating problems that would arise due to the gamification of the simulation-framework. Table 4.1 features a list of mods used in this experiment and briefly explains their functionality.

Table 4.1: Modlist

Unlimited Money/Soil/Oil and Ore	Bundled in with the game, provide unlimited resources
Unlock all	Also native to the game, circumvents progression needed to unlock streets, buildings and zones.
25 Spaces	Unlocks the full map, in a normal game only parts of the complete playable map can be unlocked
Unlimited Outside Connections/Any Road Outside Connections	Allows for free-form street management during terrain creation
Harmony 2.04/Prefab Hook	Dependencies for other mods, not having any effect by themselves
Remove need for Pipes/Power Lines	Removes infrastructure of water and power
No Fires/No Abandonment/Pollution Remover	Various mods used to remove unrealistic behaviour or mechanics
Infinite Resources	Provides Industrial Zones with infinite resources for production
Traffic Manager: President Edition	changes behaviour of traffic
Geo Skylines	converts real world data into in-game objects
Image Overlay	allows images in the game files to be overlaid into the game



(a) The overlaid picture.

(b) The same spot without the overlay active.

Figure 4.1: Mods such as Image Overlay help with the manual post processing during setup.

The "Unlimited"-Mod-Series and Unlock all was used to allow creating a city without needing to accommodate for limited resources present in normal play. Unlimited Outside Connections, Any Road Outside Connections and Remove need for Pipes/Power Lines was used to simplify the infrastructure of the simulation to get it ready for evaluation faster. 25 Spaces was employed to make sure that the whole possible scope of the simulation could be observed. No Fires/No Abandonment/Pollution Remover were all used to remove unrealistic game behaviour, this was to prevent scenarios that would simply not happen like that in the real world like building flammability being too high on average, crime rates being ridiculously high and spontaneous building abandonments. Geo Skylines was already explained earlier and Traffic Manager: President Edition is later used as part of the experiment. Prefab Hook and Harmony 2.04 are solely used for dependency purposes. Image Overlay was used to directly review the transferred data by overlaying an image of the area from a service like OpenStreetMaps over the simulation shown in Figure 4.1.

4.2 City creation and manual post processing

The previously mentioned Geo Skylines mod is bundled in with an in-depth guide on how to transfer a city into the Cities Skylines framework, however, even there the last step of the procedure is described as "manual post-processing". The mod and its methods of transferring real-world data to an in-game environment are far from perfect; so ironing out the quirks remains a manual as well as rather tedious process. In the way it currently operates the mod ignores waterways when placing down roads leading to streets being submerged when there should have been bridges placed or bridges are attempted to be placed but are not set quite right as shown in 4.2. This was only fixed by manually delet-

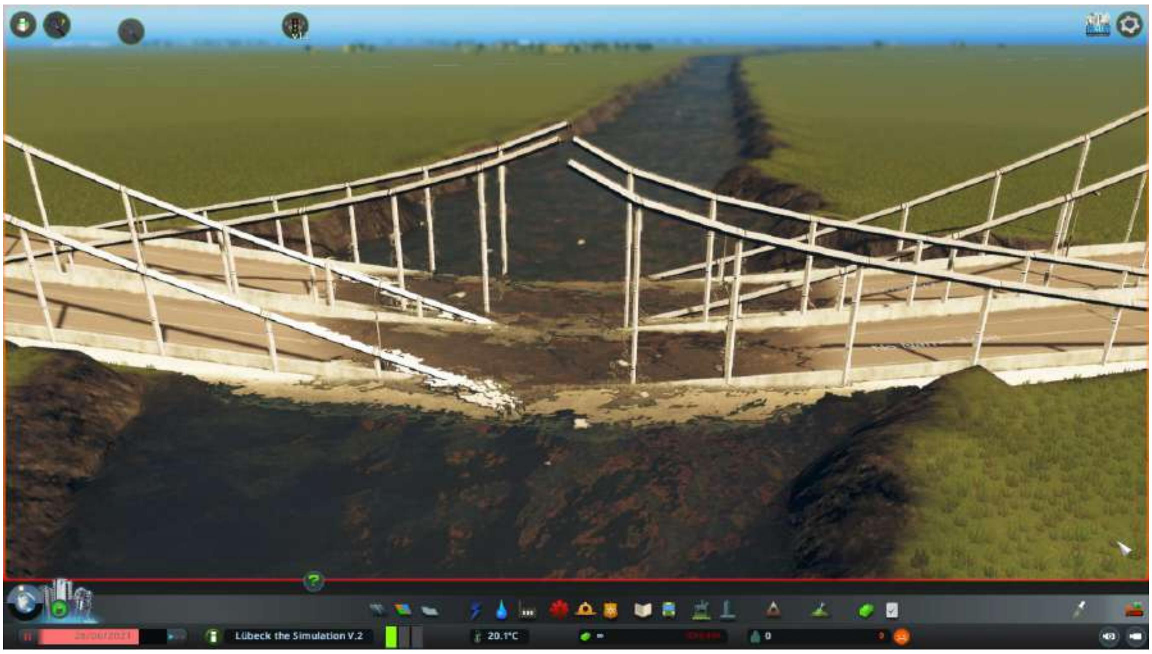
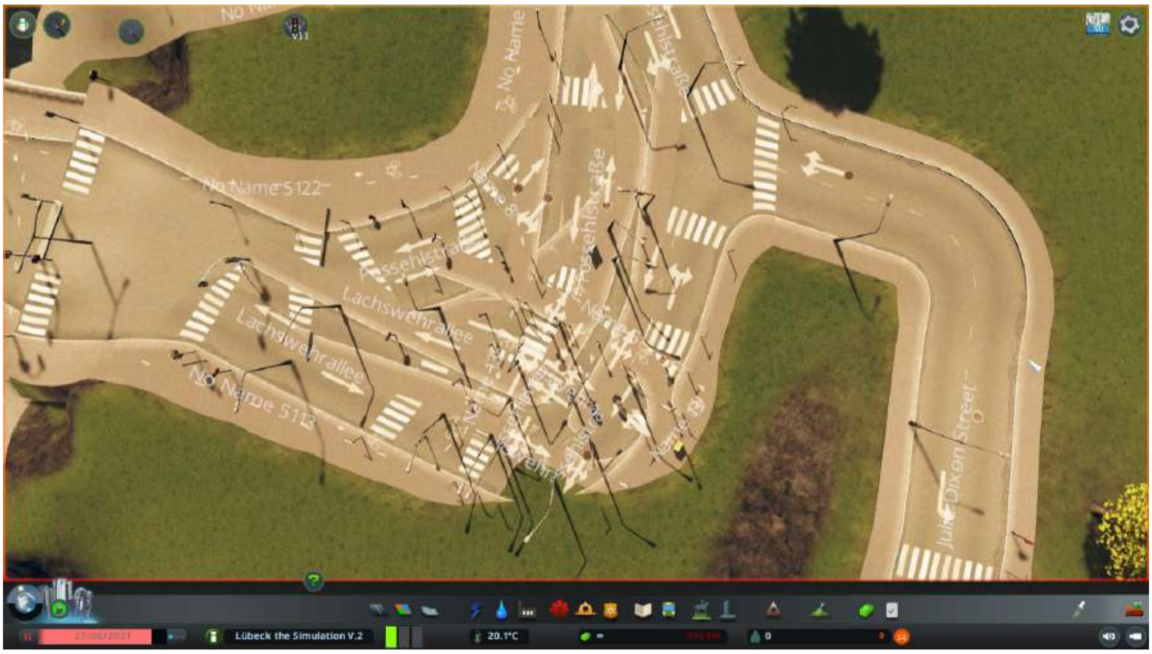
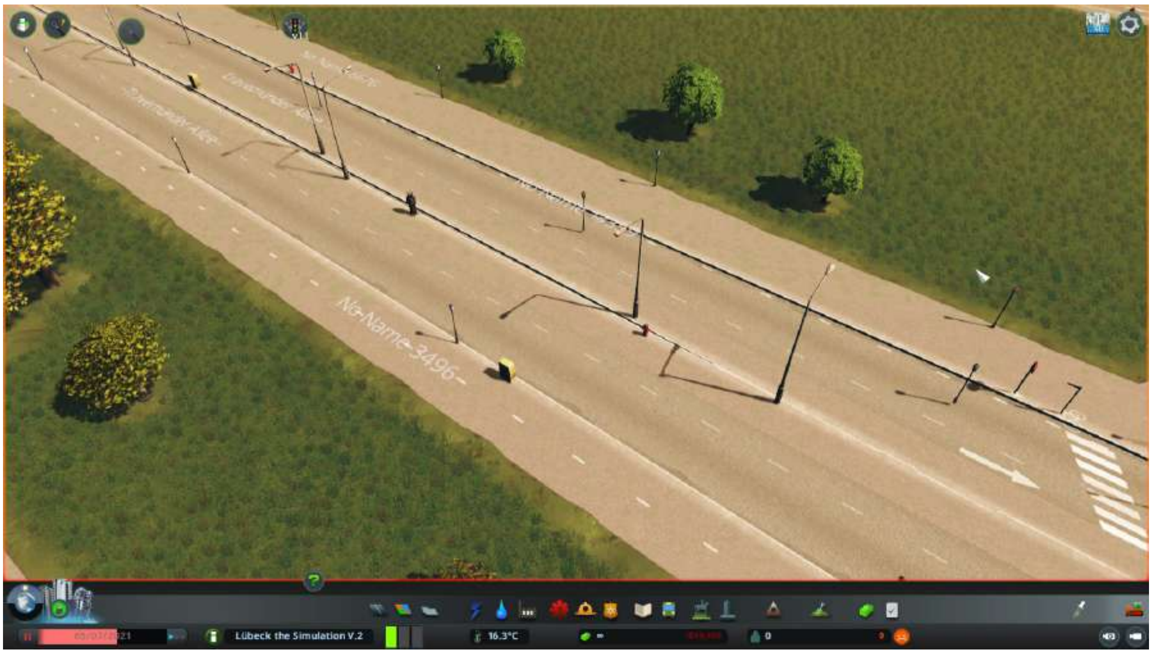


Figure 4.2: This bridge did not exactly succeed at its job.

ing and re-placing the roads which is not that much of a problem in itself but leaves room for human inaccuracies and mistakes. It is also incredibly important to map the real-world categories to appropriate in-game counterparts, finding approximations for very specific street-types is already challenging enough, however one must also pay attention to the limits set by the game. The mod sets down game objects which are derived from real-world data in one instance and thus can ignore the usual limitations set by the game. For example, while the initial placement of the road data might take place the limit could very well already have been reached with that amount of game objects preventing any more to be placed and thus greatly hindering the process of correcting wrong roads. Additionally, when handling large quantities of data the mod does take a long time to finish processing all data at hand. In this project the import of road data took roughly over 12 hours on average, this must be kept in mind when creating a city and planning to conduct an evaluation. Another interesting anomaly was the fact that through the mod streets were able to be placed in a way that would not be possible in normal play. For example, numerous sections would simply be deemed ”too short” or too close to another road, however, because the mod places down all roads simultaneously these limitations do not apply which results in rather interesting street formations that look like they should not work but ultimately do like in Figure 4.3a or Figure 4.3b. To further improve the accuracy of the simulation we had to modify some of the data at hand. Normally the amount of lanes on a given street gets lost during the conversion from real-world data to part of the simulation.



(a) This looks like it should not work but somehow it does.



(b) Normally these two streets would be deemed ”too close” to each other, but because the mod sets them down in one instantaneous moment this limitation does not apply. If one was to delete one of these streets however they would not be able to place them down again.

To circumvent this and achieve more realism the data set is checked for all lane entries with a number greater than 3 then the number of lanes is appended to the road category entry. After that, the file used to map real-world data onto simulated streets needs to be adjusted by adding the modified names to it.

5 Experiment

5.1 Collecting validation data

For the sake of confirming the validity of the simulated synthetic data, publicly available data was collected through map services like Google-Maps and Bing-Maps when regarding traffic data and OpenSenseMap concerning Pollution data. However, it is important to keep in mind that at the point of writing this and throughout the experiment the world was subject to the global COVID-19 pandemic. Thus it may be possible that the real world data does not correctly reflect the long term trends of both pollution and traffic. To gather this specific traffic data the Google-Maps view had to be centred on the city of choice, in this case, Lübeck, then it was possible to view the traffic layer with the option on the left-hand side which is labelled ”Light/Moderate/Heavy traffic in this area”. From that point onwards one can choose to observe live traffic or the typical trends during a given time, in this experiment the choice fell to the latter option. When trying to collect Bing-Maps data, the approach was to use the API. For one month every ten minutes from 6 AM to 6 PM the API would be called for congestion traffic events to hopefully approximate certain trends, however, the data collected over that time-frame proved to not be sufficient enough to provide valuable insight. Concerning pollution data it was not as much of a challenge to obtain usable data, openSenseMap provides the option to download data collected by individual senseBoxes with a flexible, customizable time-frame and various options for curating the data pre-download.

5.2 Evaluating the simulation

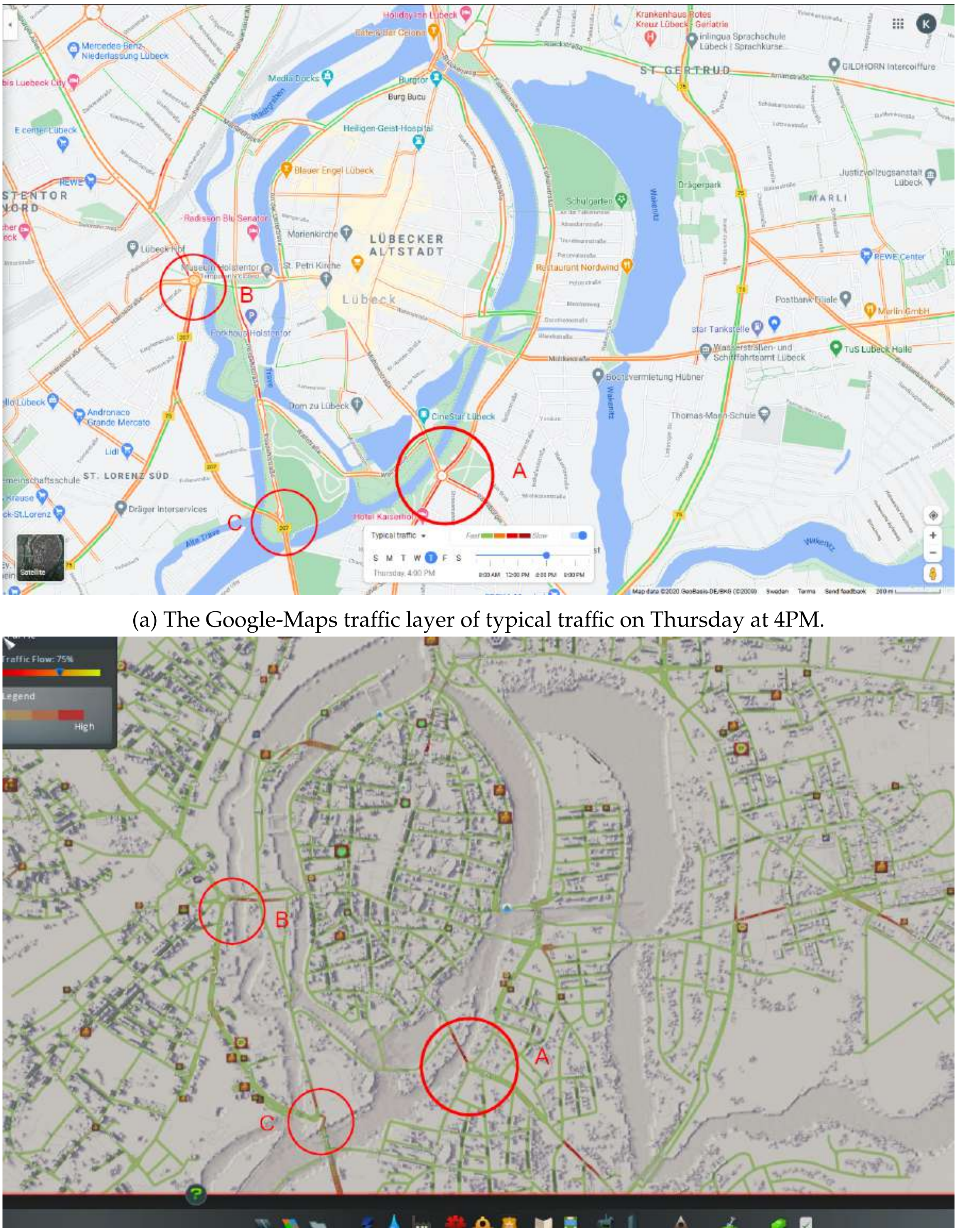
One of the goals of this thesis is to evaluate whether the present simulation is accurate. So to gather an initial idea on how well the simulation performs with regards to smart city relevant aspects it is of the essence to evaluate simulated data by comparing it to its real-world equivalent. It is worth noting however that Cities Skylines first and foremost is still a video-game and primarily serves the purpose of entertainment. Thus it does not provide a native way to extract detailed data samples of various city machinations such as traffic and pollution. Because implementing an API that serves this purpose was out of scope for this project the alternative way of manually collecting and evaluating data samples has been taken throughout this experiment.

5.2.1 The general simulation

Before taking an in-depth look at how the simulation performs when observing specific aspects relevant to smart city planning it is important to look at the simulation as a whole on a surface level. It quickly becomes apparent that the simulation does a well enough job when it comes to emulating the visuals of the city that was chosen for this experiment. Thanks to the GeoSkylines mod it was made sure that the simulation resembles the target city of Lübeck to a certain degree, this manifests itself by the fact that the central districts of the city are mainly used for commercial purposes by featuring a variety of places of purchase. Through manual post-processing, it was also possible to achieve a simulated population of ≈ 220000 citizens which roughly lines up with Lübecks own ≈ 216000.

5.2.2 Traffic Data

The simulation displays the flow of traffic in a colour-coded manner which is similar to the way it is displayed in the Google-Maps service thus resulting in two easily comparable measurements. To evaluate the simulation data from Google-Maps in the form of screen-captures showing the typical traffic during certain times were collected. The times chosen were 8 AM, 12 PM, 4 PM and 8 PM on every day of the week. The simulation data was collected similarly using screen-captures, 19 samples have been taken all roughly 2 simulated weeks or about 7 real-time minutes apart. When comparing the samples of both real-world and simulated traffic like in Figure 5.1 it becomes apparent that certain trends seem to match up, the simulation seems to particularly mirror the flow of traffic during weekdays at 8 AM and 4 PM which are usually traffic-heavy times.



(a) The Google-Maps traffic layer of typical traffic on Thursday at 4PM.

(b) A sample of the simulated traffic layer.

Figure 5.1: A comparison of the Google-Maps layer and the simulated traffic layer showing that certain high traffic trends line up.

5.2.3 Pollution

Unfortunately, not all aspects that were observed proved to be viable observable criteria out of the box. The simulation framework does not feature measurement for air pollution such as in the form of fine particles, there is a measure however that seemed comparable at a surface level, noise pollution. The way noise pollution is displayed in the simulation is via the means of something akin to a heat-map that features a 9 step gradient of colour to signalise the intensity of it. The real-world data was collected through the opensensemap service. For the sake of comparison, the hourly average of the data-points for one month was used, afterwards, a division of the measurements into 9 steps according to the in-game gradient took place using the maximum hourly average as the highest value of said gradient.

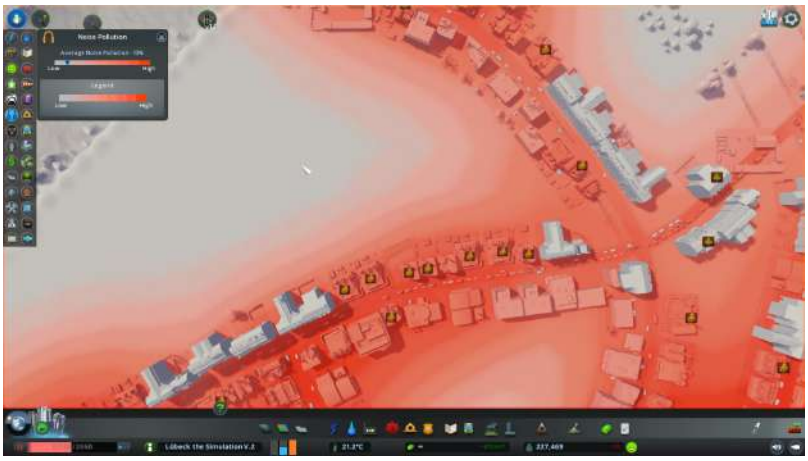


Figure 5.2: Concerning PM2.5(fine particles) with a max value of $\approx 4.58\mu/m^3$ the gradient would be divided into $\approx 0.51\mu/m^3$ per step. The mean is $\approx 0.65\mu/m^3$ thus the sensor-box would be located in a colour of the very first gradient step. This however is not the case. For PM10 with a max value of $\approx 14.75\mu/m^3$ the gradient is divided into $\approx 1.64\mu/m^3$ per segment. With a mean of $\approx 0.95\mu/m^3$ this aspect can thus be deemed accurate.

If the simulation was accurate then the calculated average of all hourly averages should fall into a step on the gradient akin to the colour of the region the sensor is in. The unfortunate truth however is that while one sensor with the location shown in 5.2 was accurate concerning PM10 measurements as shown in 5.4 were both median and mean fall into the projected range, it failed when observing PM2.5 measurements shown in 5.3 where only the median falls into the range of simulated data, however, the majority of the belly is still part of the range. The other sensor, location is shown in 5.5, was deemed to be inaccurate for both measurements as shown in 5.6 and 5.7, in these cases both median and mean fell out of the simulated range, it is also worth noting that both plots feature two bellies which indicate that the measurements are a lot more varied. This is a strong indicator that the simulation might work better when the observed aspect does not feature a lot of variances.

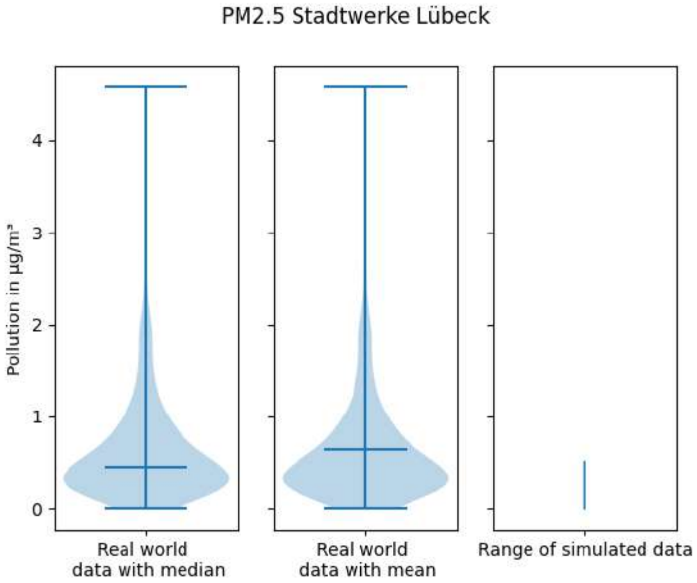


Figure 5.3: A violin plot showing all data samples with the violins themselves representing the frequency of values occurring with an extra line symbolising the valid range for PM2.5, the vertical lines representing the maximum value at the top and the minimum value at the bottom. The lines in between those represent median and mean respectively. If both median and mean were to fall into the area represented by the ”range of simulated data” we could assume the approximation to be somewhat accurate to real world data. While this is not the case the ”belly” of the violin still falls into the range.

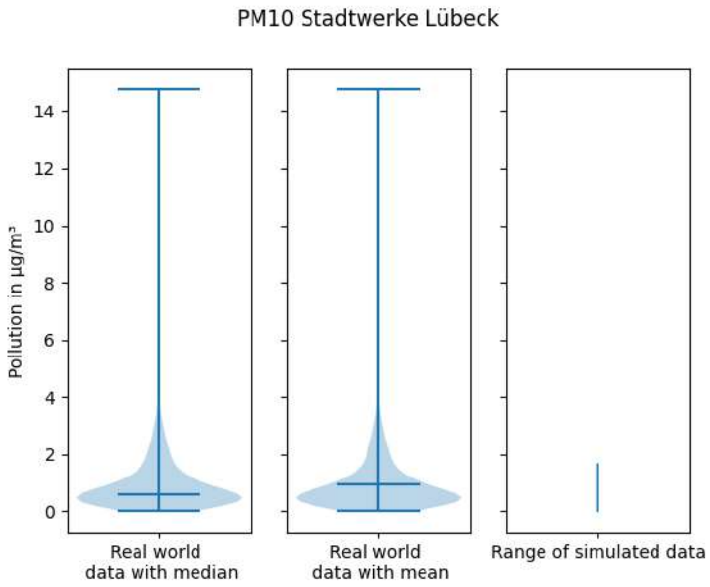


Figure 5.4: Same method for PM10. This time the belly is a lot flatter signalising that there is a lot more outliers, however both median and mean fall into the range of simulated data thus verifying the supposed accuracy of this particular use-case.

However, these inaccuracies most likely stem from the fact that noise is not a good approximation of air pollution since industrial zones might produce a lot of noise but not a lot of fine particle emission depending on the kind of industry. So when compared to cars the simulation rightfully deems industry to be a lot louder and cars to be quieter on average, applying that to our approximation would mean that cars seem a lot cleaner concerning air pollution than they are in real life. A test run with mods was not conducted, however, meaning that there is a possibility that with the help of such modifications a more accurate approximation of air pollution could be achieved.

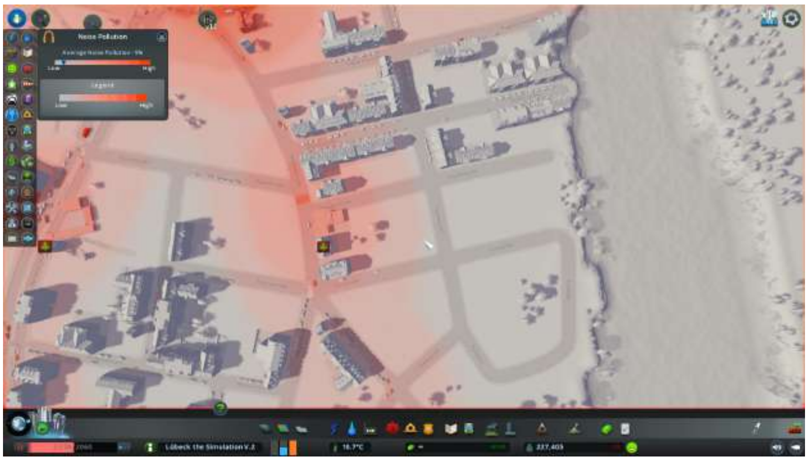


Figure 5.5: With regards to PM2.5: with a max value of $43.5\mu/m^3$ the gradient would be divided into $\approx 4.9\mu/m^3$ per step, thus with a mean of $10.2\mu/m^3$ the sensor-box would be located in a colour of the third gradient step. For PM10: with a max value of $113.48\mu/m^3$ the gradient is to be divided into steps of $\approx 12.61\mu/m^3$. So with a mean of $19.33\mu/m^3$ the location should fall into a colour of the second gradient, , however the cursor, showing the position of the sensor, is in an area of the lowest possible value, thus proving both approximations to be inaccurate.

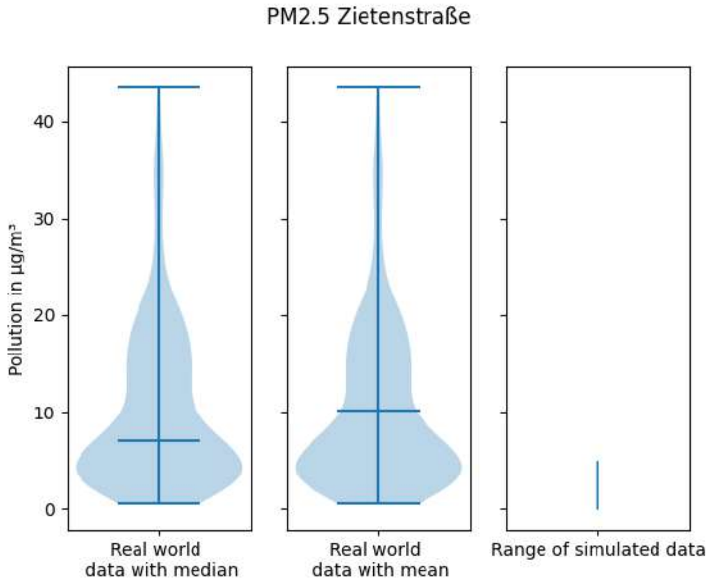


Figure 5.6: This particular violin plot features two bellies signifying a wider distribution of values rather than all values falling into a particularly small range. Both mean and median also fall out of the simulated data-range which unfortunately means that the simulated data is to be deemed inaccurate.

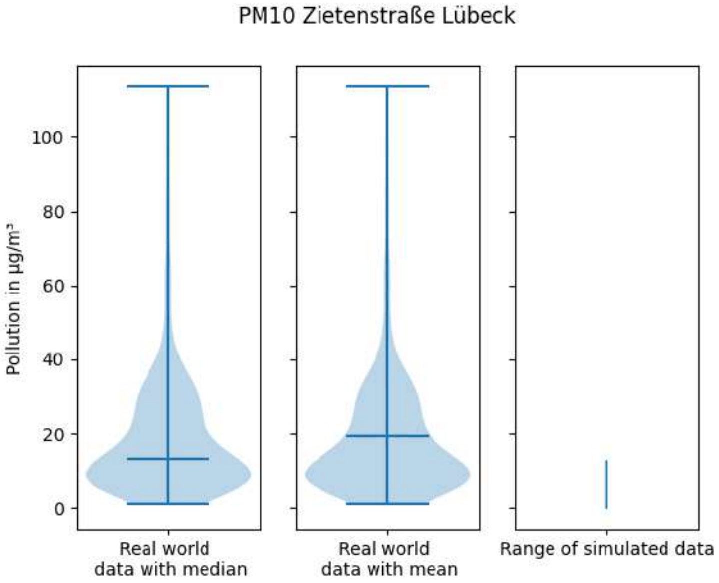
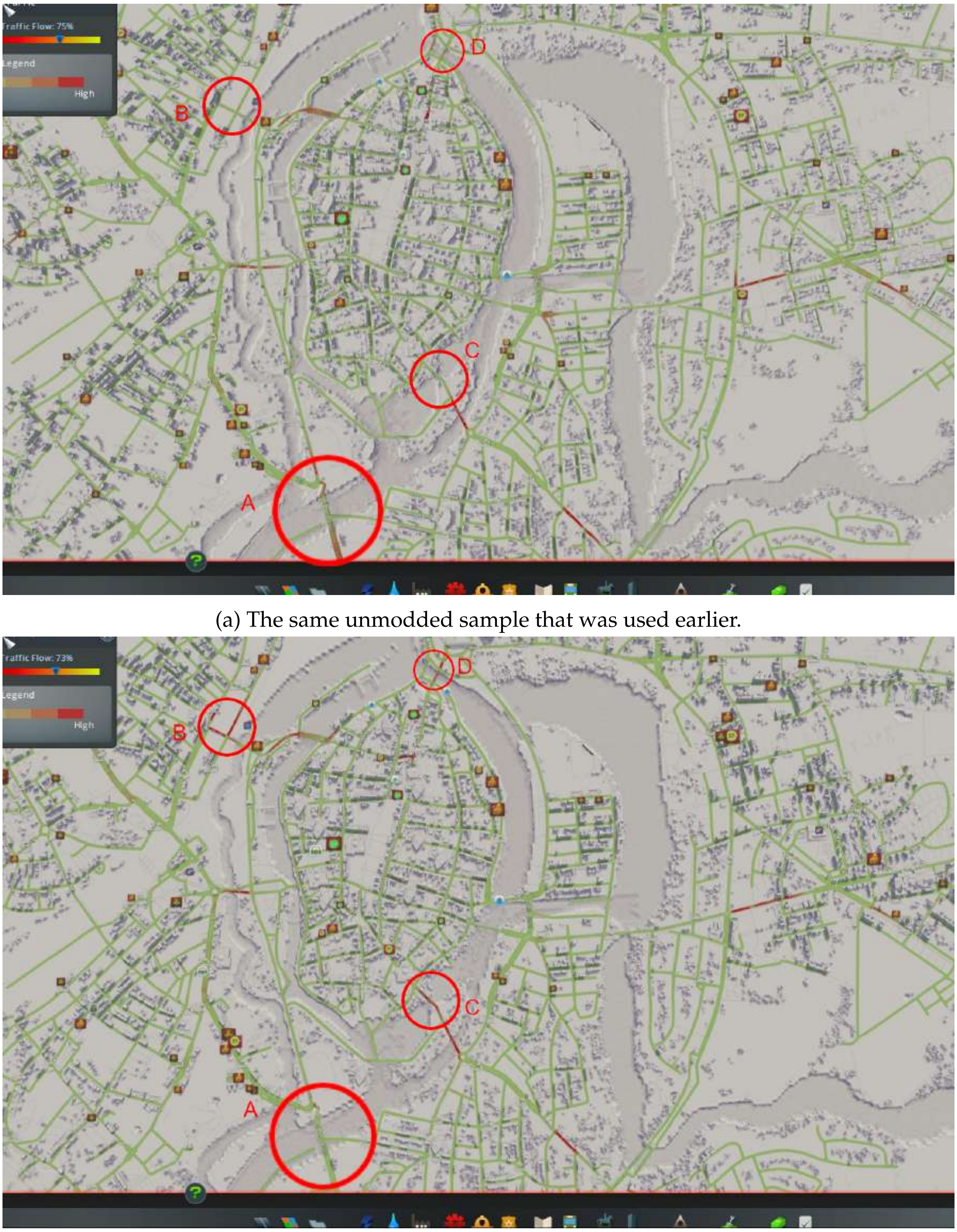


Figure 5.7: A similar pattern could be seen in the plot concerning PM10 data. This time only the median falls into the simulated data range thus showing that for this measure the simulation is at the very least somewhat accurate.

5.3 Modding

Modifying the game using the Traffic Manager: President Edition Mod, which advertises improving the traffic ai the results that were achieved thereafter slightly differed. However, a sample of 12 screen-captures reveals that the trends previously discovered roughly stay the same, nonetheless at some points traffic flow showed improvements over the vanilla (unmodded) simulation. While this might not have necessarily helped improve the realism of the simulation it shows that certain aspects of the game can be modified to achieve differing results thus allowing fine-tuning of the simulation to precisely emulate the behaviours of a real-world city. In Figure 5.8 the sample used to highlight trends akin to real-world data is shown in comparison to a now modded sample. Some significant differences for both traffic hotspots and other more general regions are highlighted thus signifying the impact mods can have on the behaviour of simulated traffic.

These were just some first attempts at trying to relate the simulated data to the real-world equivalents. There is still a plethora of observable data aspects that can be examined such as waste management, power usage and water availability.



(a) The same unmodded sample that was used earlier.

(b) A modded sample using Traffic Manager:President Edition.

Figure 5.8: A comparison between an unmodded and modded sample highlighting some key differences. The points chosen for this comparison featured are rather major points of traffic, this is to highlight the fact that mods can have a major impact on the simulated behaviour of certain aspects.

6 Future Work

Despite the promising results, the experiment showed some problems with this methodic. First of all, there were difficulties when it came to the fact that at the end of the day City Skylines is and was designed as a game. No matter how good a simulation it is out of the box it's bound to be flawed by virtue of the simulation trying to provide a challenging yet fun experience to its players. Furthermore, the current simulation is more akin to an early prototype. There are significant discrepancies when it comes to a direct in-depth comparison between the simulation and its real-world counterpart. Furthermore, such abstractions need to be made for the sake of simplicity since the game in its core needs to be runnable on commercially available PCs. An example of such simplifications is the way the game emulates regions of a city by dividing them into zones, those zones are low- and high-density variations of residential and commercial zones, as well as industrial and office building zones. A lot of the game flow is based on these zones so it becomes important to try and correctly approximate the city to be projected into such zones. For that reason, a properly curated dataset must be provided and the mapping has to be as accurate as possible to accommodate for problems that might arise such as to little or too much living space being present in the final simulation. For the experiment, I choose to over-approximate, as to not run into the issue of having to constantly upscale the simulation, since preparing too many zones and then stopping their development proved to be easier than under-approximating and having to retroactively adjust. Another rather curious problem that arose was the fact that after some time standing waters just seemed to dry out, this did not impact the results in any way and could be ignored for the sake of this experiment however this particular oddity seems to be something that is still very much worth noting.

For future research, it would be wise to keep these inherent inaccuracies in mind and try to improve upon them by developing or using mods that influence the simulation in a way that will lead to more realistic behaviour. The dataset that is used for transferring geographical data into the in-game environment can also be improved upon by further curation. This curation could take the form of either specifying the types of roads and buildings that need to be present in the modelled city or just generally giving more thought about the approximation that has to take place when transferring a real-world city into an in-game environment. Furthermore, the game also allows the modification and creation of assets that can be used to model certain aspects and services not currently

present. It will be important to determine what aspect is of note for what city and to adjust and develop mods specifically for edge-cases.
Despite its flaws and slight inaccuracies, this simulation serves as a solid base for the future. A workable model of Lübeck now exists and can be modified using more accurate data as well as continue to be observed regarding smart city relevant aspects by employing and developing mods for this specific purpose. This solution however still needs real-world data for the sake of comparison. The goal is to create a simulation that will accurately show the effect changes in favour of smart city have on an existing populated environment. For this purpose, Cities Skylines does not provide what is needed yet. However, it becomes apparent that the framework has potential regardless and it is exciting to see where this potential may carry the framework in the future to come.

7 Conclusion

It has become very apparent that smart city will play a big part in modern planning of habitual space. With that vision on the simulation of such spaces becomes immensely important for economical and ecological reasons. With this work, the first steps towards a viable city approximation have been made. It was shown that while still lacking in some departments, Cities Skylines provides a solid basis for the development of future simulations. The game features a plethora of observable aspects relevant to outlining smart city plans. These aspects might not always be exactly true to life but with the addition of third-party modification support, the game allows its users to influence the inner machinations in a way that would make it suitable for these purposes. Furthermore, mod support allows for the simplification of creating a working simulation, by allowing the use of publicly available real-world data to be used for a true to life image of an existing city.
All these factors play into the fact that there is a lot of potential for the future and that through further work on this basis better simulations and thus more accurate synthetic data can be achieved which will help designing smart city infrastructure without violating the data privacy of citizens.

References

[3dC] 3d.city. `https://lo-th.github.io/3d.city/`. Accessed: 2021-03-10.

[aep] aeplay. citybound. `https://aeplay.org/citybound`. Accessed: 2021-03-10.

[Ber16] Bradley Bereitschaft. Gods of the City? Reflecting on City Building Games as an Early Introduction to Urban Systems. *Journal of Geography*, 115(2):51–60, March 2016.

[bin] Bing maps dev center. `https://www.bingmapsportal.com/`. Accessed: 2021-03-10.

[Ear20] Sam Earle. Using Fractal Neural Networks to Play SimCity 1 and Conway's Game of Life at Variable Scales. *arXiv:2002.03896 [cs, stat]*, January 2020. arXiv: 2002.03896.

[Gam20] Ambiera Games. Smart city plan. `https://www.ambieragames.com/smartcityplan/index.html`, 2020. Accessed: 2021-03-10.

[goo] Google maps. `https://www.google.com/maps`. Accessed: 2021-03-10.

[HVK+15] Pyry Haahtela, Tuomas Vuorinen, Anni Kontturi, Henrik Silfvast, Mika Väisänen, and Jaakko Onali. Gamification of Education: Cities Skylines as an educational tool for real estate and land use planning studies. page 14, 2015.

[Inc89] Maxis Software Inc. Simcity. `https://archive.org/details/msdos_SimCity_1989`, 1989. Accessed: 2021-03-10.

[Int] Interscsimulator. `https://github.com/ezambomsantana/smart_city_model`. Accessed: 2021-03-10.

[JHT17] Max Juraschek, Christoph Herrmann, and Sebastian Thiede. Utilizing Gaming Technology for Simulation of Urban Production. *Procedia CIRP*, 61:469–474, 2017.

[Ltd15] Colossal Order Ltd. Cities skylines. `https://www.citiesskylines.com/`, 2015. Accessed: 2021-03-10.

[OCSW20] Robert Olszewski, Mateusz Cegiełka, Urszula Szczepankowska, and Jacek Wesołowski. Developing a Serious Game That Supports the Resolution of Social and Ecological Problems in the Toolset Environment of Cities: Skylines. *IJGI*, 9(2):118, February 2020.

[Ope] Opensc2k. `https://github.com/pdeva/OpenSC2K`. Accessed: 2021-03-10.

[OSM] opensensemap. `https://opensensemap.org/`. Accessed: 2021-03-10.

[PVP20] Jan Pinos, Vit Vozenilek, and Ondrej Pavlis. Automatic Geodata Processing Methods for Real-World City Visualizations in Cities: Skylines. *IJGI*, 9(1):17, January 2020.

[SBKM] Eduardo Felipe Zambom Santana, Daniel Macedo Bastista, Fabio Kon, and Dejan S Milojicic. SCSimulator: An Open Source, Scalable Smart City Simulator. page 9.

[Sim] SimHacker. micropolis. `https://github.com/SimHacker/micropolis`. Accessed: 2021-03-10.

[Sot10] Olli Sotamaa. When the Game Is Not Enough: Motivations and Practices Among Computer Game Modding Culture. *Games and Culture*, 5(3):239–255, July 2010.

[SSSS17] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership Inference Attacks Against Machine Learning Models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18, San Jose, CA, USA, May 2017. IEEE.