AG2417
2022-10-28

Dynamic Web Map of Noise Data and Traffic Flow

Yu Gao
Alba Lunner
Pelle Rauch
Lola Voignier-Simon

TRAFFIC NOISE EXPLORER

A DYNAMIC WEB MAP APPLICATION

1. Abstract

This report addresses the process and final result of the development of a dynamic web application displaying a noise and traffic simulation. Noise pollution is an environmental stressor which affects citizens' health (*European Environment Agency*, 2020), this makes methods for measuring, quantifying and visualizing noise pollution important to develop necessary policies and means of control to lower the noise in cities.

The developed web application is suitable for any screen size and demonstrates various possibilities of visualizing noise distribution from traffic. Some simulation choices include filtering of noise from just one specific vehicle or entire traffic, different sized buffers around chosen vehicles and reducing noise from the vehicle. The user can also change car type to electric car and choose the detail level of the noise visualization.

The application presents a prototype, or proof of concept, with a number of simulation options. Improvements include the ability to quantify and sum noise from one or several vehicles and more efficient rendering.

2. Background

Road traffic is the biggest source of noise pollution in Europe. Noise pollution is affecting the quality of life in many cities and has negative effects on citizens' health. According to the World's Health Organization, noise pollution from road traffic is the second most harmful environmental stressor in Europe (*European Environment Agency*, 2020). The ability to monitor and analyze noise pollution correlated to traffic levels could contribute to deeper knowledge of the effects of noise pollution and offer solutions on how to lower it.

The application can be used as a tool by individuals who want to observe the traffic level and noise pollution to understand where noise pollution is most prominent. This could affect personal decisions such as where to buy an apartment or where to go for a walk. The web application can also be used by organizations to analyze the connection between road traffic and noise levels.

3. Requirement analysis and specification

3.1 Requirements analysis

For this particular project the key stakeholders would be the students developing the web application. The focus would be to learn about software development connected to geospatial analyses in order to pass a course.

In a further perspective, the key stakeholders of the web application are the funding organizations of the GEOMETRIC project and the researchers looking for a pedagogical way of displaying the simulation of gathered data to organizations and private users.

To capture the requirements of the web application, the project group participated in a presentation on the GEOMETRIC projects' development and end goals. These end goals include quantifying the noise pollution from individual vehicles on a route and the sum of the noise from a company's fleet. Quantifying this could contribute to fees proportionate to the amount of noise a company produces in a city.

The scope of this project could not fulfill all end goals, but could provide a proof of concept. This project will result in a prototype with potential to further build functionalities which can help to further specify requirements of the application.

3.2 Use case

To specify the requirements and visualize the end product, a use case was constructed. Following use case illustrates the importance of users' possibility to independently filter data and the need for relevant parameters.

System: Traffic and noise visualization web application

Primary actor: Employee in Stockholm municipality

Scenario: An employee at the environmental department wants to check the level of noise disturbance of a company's fleet in order to give them a proportionate fee. The employee checks the web application and filters the traffic depending on the parameters of the company in question. The noise disturbance is adjusted according to the percentage of electric vehicles, total number of vehicles in the company fleet and vehicle routes through the city. The level of noise disturbance produced by the company is decided and a fee is set.

3.3 Requirements specification

3.3.1 Functional requirements

Users need to be able to choose between values of the parameters. Users should be able to choose a type of vehicle, and visualize different types of vehicles simultaneously. Characteristics such as speed and acceleration should also be visualized. Noise level could be set to display noise from the entire traffic or just one vehicle. The users should also be able to adjust the noise level for a specific vehicle.

The application needs to be able to process chosen values and adjust the simulation accordingly. The application needs to store input in a database. The map should be easy to understand and use.

In further development stages the users should be able input their own values instead of predefined values.

3.3.2 Technical requirements

The rendering and data processing in the application needs to be fast enough for the application to be usable, similar to simulating in real-time.

4. Choice of Design

The chosen design of the application consists of a database for storage, a backend solution with node.js and a frontend solution based on HTML, CSS and JavaScript, see figure x below.

Node.js was used to execute the program on a web server during the development phase to continuously test the functionality and layout of the components. The application was deployed on an KTH-server since it is accessible to all.

PostgreSQL was used as the database system to store all data. It was used with the PostGIS extension which adds support for spatial data, it was useful since both the noise data and the traffic data was geographically bound.

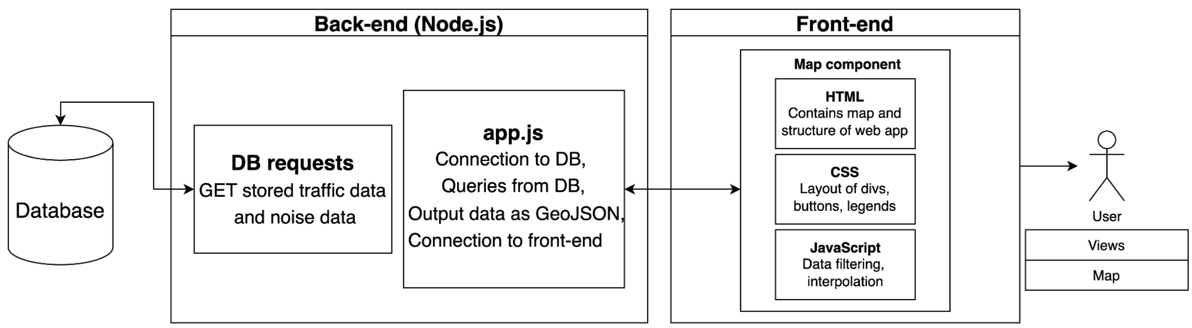

Figure x. Diagram of the web applications architecture

HTML was used in the frontend to define a structure of the website. Modal window with instructions is set to show up at page load, then the page with header, map and footer is shown. The modal window shows the logo and a small set of instructions on how to use the web application. The header contains all filtering choices the user has. The footer has some background information about the web application. The map takes up the most space since it’s displaying the simulations.

CSS defined the layout of the HTML elements. It was used to determine the amount of space on different screens each element would have. JavaScript was used to program the behavior of the application and to manipulate the data fetched from the database.

Bootstrap was implemented because of its automatic grid system that resizes the HTML elements, allowing us to easily create a mobile-friendly application that was functional on every screen size. For its many mapping features and the fact that it is suitable for mobile use, Leaflet was used to create the web map. Turf.js was used to make the geospatial analyses.

5. Data

The data used is provided and managed by the GEOMETRIC project (GEO-based Multi-layer Environmental Modeling of Urban Traffic).

5.1 Noise data

The noise data consists of five minutes noise levels in an area in Södermalm, Stockholm. It was divided into 1 second timesteps and consisted of receiver id, timestep and noise level in decibel A.

5.2 Traffic data

The provided traffic data consists of five minutes of simulated traffic by the simulation software Eclipse SUMO. The data used in the project includes time data, acceleration, id, lane, angle, slope, position, speed, vehicle type and latitude and longitude coordinates.

6. Tests

Unit testing was done continuously during the development to ensure each unit's functionality and compatibility with the main code, so called white-box testing.

Each unit was integrated into the web application immediately after a successful unit test. For integration, black-box testing was implemented by controlling that the requirement specifications were fulfilled after by the integrated units. Both the unit tests and integration tests consisted of launching the web application on the web server and testing that the added unit worked, and all other units still worked as intended.

The web application was tested on four different operating systems and on one browser, chrome.

A user acceptance test has not been carried out, and could be an important next step to ensure the satisfaction of the users and collect modification suggestions.

7. Deployment

Deploying the application is done by utilizing a KTH-server. When the server is active/started/up and running, users can access the application by entering following address: http://130.237.64.7:3000/home

8. Reflection

A few solutions for user interface were tested but didn’t make it through the integration testing. A bootstrap-vue template was downloaded and launched but it turned out to be too time-consuming to integrate with the map element, so this solution was abandoned after a number of tries. Similarly, a few nice-looking solutions in CSS were tested but didn’t integrate well with the map element and looked strange on small screens. This resulted in building a simple, functional, mobile-friendly CSS layout from scratch.

To control the simulation, the first solution was to create a slider, but it didn’t work very well for the first few tries. In order to move on with the project it was decided to settle for a play button and a pause button instead.

Some solutions improved the application significantly. The query of the noise was incredibly slow at first, so a GIST index was applied in the database to the location column. This improved the query time by 100 times.

9. Demonstration

The first thing the user faces is the modal window with the logo and user instructions describing which possibilities the user has, this is demonstrated in figure x below.

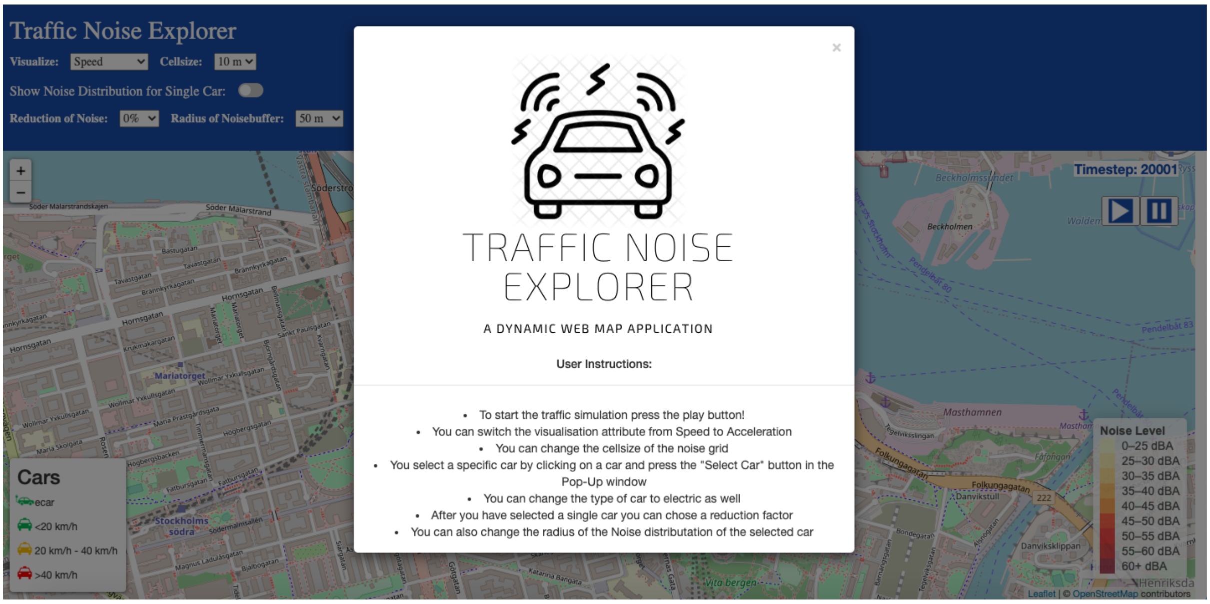

Figure 2. Modal window of web application at page load

Secondly, the user views a header with filtering choices and the map with play buttons to start the simulation and pause button to pause the simulation, see figure x and x below. Above these buttons, a time step counter shows what time step is currency displaying on the map. The header contains choices of whether to visualize the cars based on speed or acceleration. It is also possible to choose grid size to see a more detailed noise distribution. Furthermore, the user has the choice of displaying the noise for a specific car. The chosen car can have different sized buffers to display a number of predefined areas of noise. These areas can be set to have a certain percentage of the original noise to simulate lower noise levels.

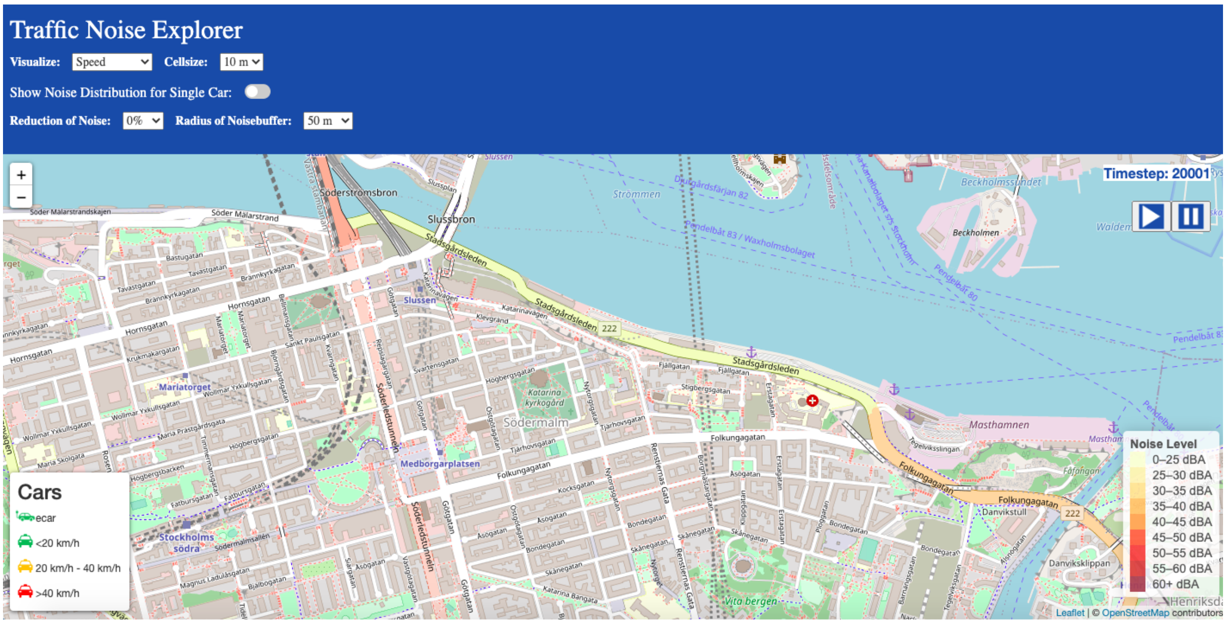

Figure 3. Web application on a medium sized computer screen

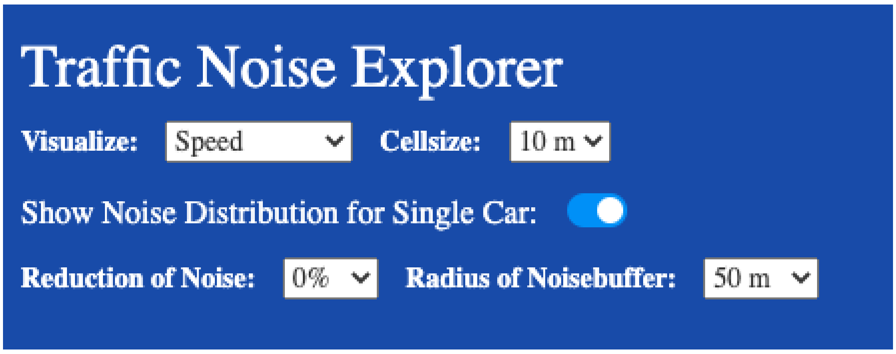

Figure 4. Choices to apply to the simulation

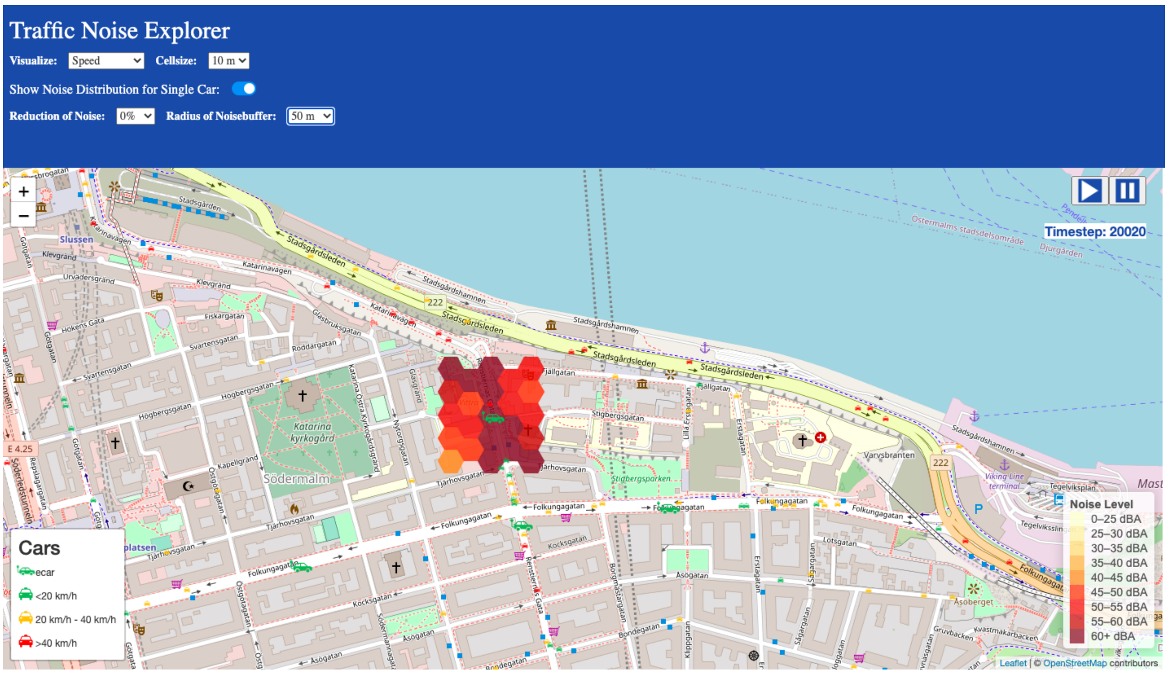

Figure 5. Web application with simulation of noise from a selected car with a buffer of 50 meters

The application viewed on a mobile device is demonstrated in figure x below. The buttons are enlarged to make them clickable and comfortable to use on smaller screens as well. In figure x the information pop-up is exemplified. The information consists of vehicle ID, current speed in meters per second, acceleration in $\frac{m}{s^2}$ and vehicle type. In the popup the user can choose a car to display its noise and change vehicle type to electric car.

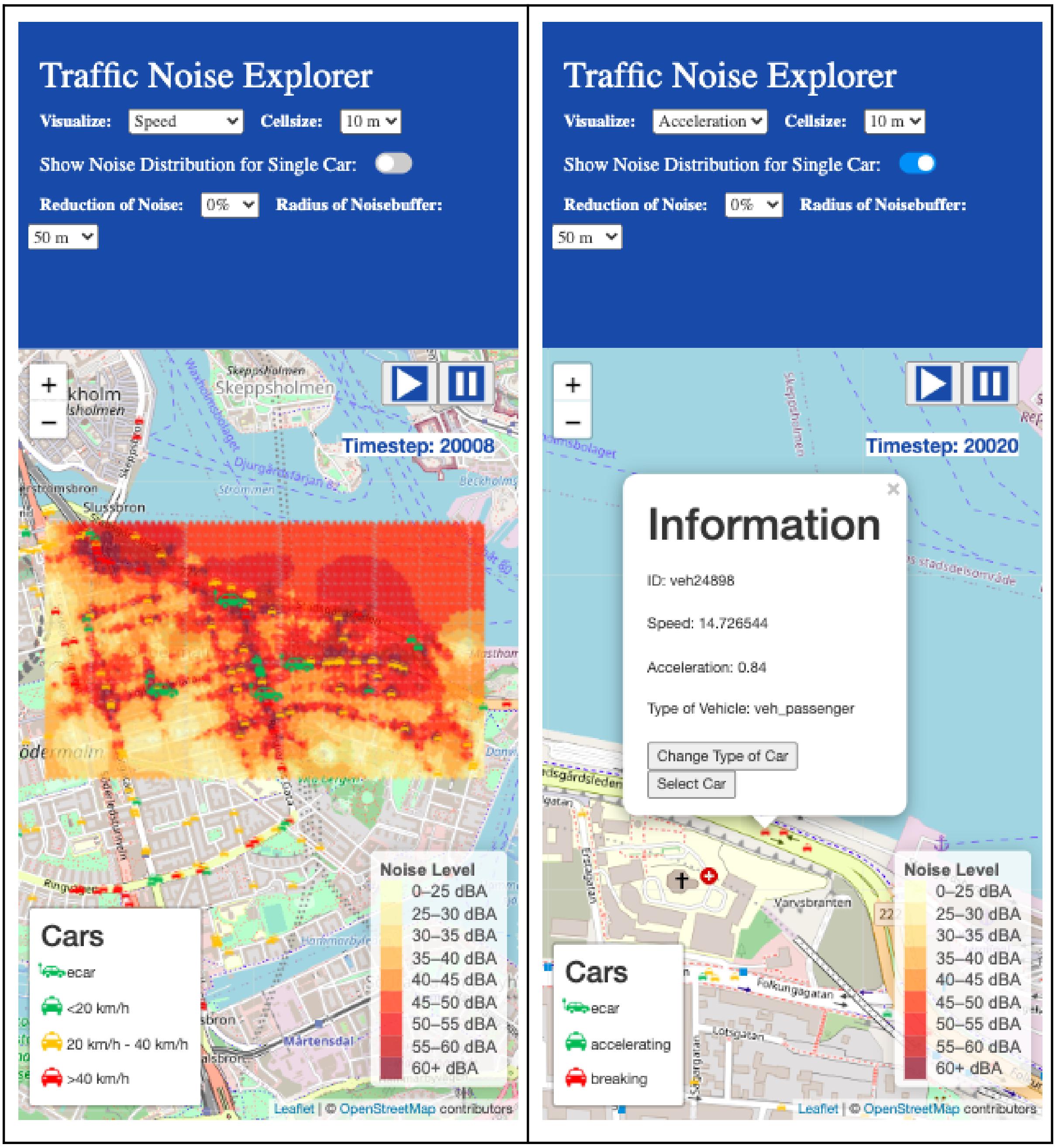

Figure 6. Web application on a mobile device (IPhone XR in this case), the left picture shows the simulation and the right picture displays the information pop-up

10. Teamwork

The team members had different responsibilities during the project. Yu did a lot of work on the interpolation. Pelle worked on the database connection and the implementation of the GeoJSON returns in the appliction. Alba worked on the layout and reporting and Lola did lots of reporting. The whole group did continuous testing throughout the entire project.

11. Limitations and Improvements

An improvement of the app would be to have more effective rendering. Other improvements include the ability to put any value into a filter instead of predefined values. Additional relevant parameters could be added, such as quantifying the noise from one or several vehicles to improve the relevancy to the GEOMETRIC project. Further developments can aim to implement noise routing so that, for instance, trucks can be navigated in such a way that they cause as little additional noise as possible.

The limitations of the web application include .. time and skills probably

12. Conclusions

In conclusion, a solid first prototype of a noise and traffic web application has been developed. The application could definitely benefit from some improvements, but it is a good beginning of a proof of concept.

13. Sources

European Environmental Agency. *Road traffic remains the biggest source of noise pollution in Europe.* 2020. Available: https://www.eea.europa.eu/highlights/road-traffic-remains-biggest-source [Fetched 2022-10-6]