UNIVERSIDAD DE INVESTIGACIÓN DE TECNOLOGÍA EXPERIMENTAL YACHAY

Escuela de Ciencias Matemáticas y Computacionales

TÍTULO: VISUAL ANALYTICS OF TRAFFIC SIMULATION DATA

Trabajo de integración curricular presentado como requisito para la obtención del título de Ingeniero en Tecnologías de la Información

Autor/a:

Almachi Loor Christopher Hernán

Tutor/a:

Ph.D. Armas Andrade Tito Rolando

Co-tutor/a:

Ph.D. Cuenca Pauta Erick Eduardo

Urcuquí, Mayo 2024

Autoría

Yo, **Christopher Hernán Almachi Loor**, con cédula de identidad 0803742873, declaro que las ideas, juicios, valoraciones, interpretaciones, consultas bibliográficas, definiciones y conceptualizaciones expuestas en el presente trabajo; así cómo, los procedimientos y herramientas utilizadas en la investigación, son de absoluta responsabilidad de el/la autor/a del trabajo de integración curricular. Así mismo, me acojo a los reglamentos internos de la Universidad de Investigación de Tecnología Experimental Yachay.

Urcuquí, mayo del 2024.

Firmado electrónicamente por:
CHRISTOPHER HERNAN
ALMACHI LOOR

Christopher Hernán Almachi Loor

CI: 0803742873

Autorización de publicación

Yo, **Christopher Hernán Almachi Loor**, con cédula de identidad 0803742873, cedo a la Universidad de Investigación de Tecnología Experimental Yachay, los derechos de publicación de la presente obra, sin que deba haber un reconocimiento económico por este concepto. Declaro además que el texto del presente trabajo de titulación no podrá ser cedido a ninguna empresa editorial para su publicación u otros fines, sin contar previamente con la autorización escrita de la Universidad.

Asimismo, autorizo a la Universidad que realice la digitalización y publicación de este trabajo de integración curricular en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación

Urcuquí, mayo del 2024.

Firmado electrónicamente por:
CHRISTOPHER HERNAN
ALMACHI LOOR

Christopher Hernán Almachi Loor

CI: 0803742873

Dedication

For my parents Fausto and Sonia, for everything they have done for me. Thank you for being by my side during this journey's good and bad times. Without a doubt, they are my inspiration for sacrifice and hard work. I thank my brothers Fabricio, Lorena, Gabriela, and Valentina for all their advice and support in this university career. Lastly, I also want to dedicate this work to my girlfriend, Marilyn Figueroa. From the moment I chose my thesis topic, she has been there, constantly supporting and motivating me. Her constant encouragement and belief in me have been instrumental in proving that I can achieve my goals.

Christopher Almachi L.

Acknowledgment

My special thanks to my advisors, Rolando Armas and Erick Cuenca, thank you for your guidance, wisdom and great patience throughout this arduous work, without a doubt, your recommendations and suggestions were very useful. You have helped me grow as a student preparing me for the future and I will always be grateful for your mentorship.

To my girlfriend Marilyn Figueroa for her constant and sincere support on this path towards my professional life, thank you very much.

Christopher Almachi L.

Resumen

En el mundo actual, el transporte es una de las necesidades más importantes de la sociedad, especialmente en las zonas urbanas. La gestión y planificación del tráfico son esenciales para garantizar un flujo continuo del tráfico y minimizar la congestión. Además, la visualización de la información desempeña un papel crucial en el análisis visual de datos que proceden principalmente de datos brutos y abstractos. Las simulaciones proporcionan información detallada sobre el flujo del tráfico, las posiciones, la congestión, etc. Sin embargo, la eficacia de los simuladores se ve limitada por la capacidad de los ordenadores, ya que, cuando se enfrentan a conjuntos de datos extensos, su capacidad para soportar simulaciones microscópicas a gran escala disminuye significativamente. En este trabajo, hemos desarrollado una aplicación web de análisis visual denominada TrafficSim-Vis que consta de dos partes: un preprocesamiento y una visualización. La primera parte es un conjunto de pasos para convertir los datos de salida de la Simulación de Transporte Multi-Agente (MATSim) en un formato compatible con frameworks web. La segunda parte se centra en una aplicacion de una sola página a la que se introducirán los datos preprocesados para su visualización. Para ello, la aplicación utiliza frameworks avanzados capaces de manejar grandes cantidades de datos, como Deck.gl. Se verifica el desarrollo y, finalmente, se prueba la aplicacion de visualización con datos de tráfico simulados en Cuenca (Ecuador) y Santiago (Chile). Como resultado, obtenemos una interfaz interactiva que permite una representación atractiva de los agentes; además, los usuarios pueden analizar, interactuar y explorar diferentes aspectos de los datos y obtener información relevante acerca de la simulación.

Palabras Clave:

Datos de tráfico, simulación, visualización, MATSim, aplicación web

Abstract

In today's world, transportation is one of society's most important needs, especially in urban areas. Traffic management and planning are essential to ensure a continuous flow of traffic and minimize congestion. Furthermore, information visualization plays a crucial role in the visual analysis of data that mainly comes from raw and abstract data. Simulations provide detailed information on traffic flow, positions, congestion, etc. However, the effectiveness of simulators is limited by the capacity of computers since when faced with large data sets, their ability to support large-scale microscopic simulations decreases significantly. In this work, we have developed a visual analytics single-page application called TrafficSim-Vis consisting of preprocessing and visualization. The first part is a set of steps to convert the Multi-Agent Transportation Simulation (MATSim) output data into a format compatible with web frameworks. The second part focuses on a single-page application in which preprocessed data will be entered for display. To do this, the application uses advanced frameworks, such as Deck GL, capable of handling large amounts of data. The development is verified, and the visualization tool is finally tested with simulated traffic data in Cuenca (Ecuador) and Santiago (Chile). As a result, we obtain an interactive interface that allows an attractive representation of the agents. In addition, users can analyze, interact, and explore different aspects of the data and obtain relevant information about the simulation.

Keywords:

Traffic Data, simulation, visualization, MATSim, web application

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Traffic congestion is an urban planning problem affecting most cities worldwide. Congestion is generated when urban traffic circulation is disrupted due to increased vehicle density, increasing travel time [15]. Using a similar concept, Afrin *et al.* [16] defines congestion as travel demand that exceeds roadway capacity. Both definitions have transcended over time and include the importance of proper management, giving rise to simulations. Traffic simulations are powerful tools that allow the analysis of traffic behavior in real scenarios and evaluate the impact of traffic management measures. These simulations enable avoiding dangerous traffic situations, optimizing signaling, predicting traffic jams, and planning routes [17] [18].

Several traffic simulation models exist, each with different levels of detail and scale. Macroscopic models are the most comprehensive and usually cover huge areas, while microscopic models are the most detailed and can simulate the movement of individual vehicles [19]. This thesis focuses on the output files of a simulator called Multi-Agent Transport Simulation (MATSim) [20]. MATSim is an open-source framework for implementing large-scale agent-based transport simulations. It allows simulations to be performed in large and small scenarios, identifying the transport agents involved and describing particular characteristics that could impact traffic [20].

Users can define their transportation network and travel demand, and MATSim can simulate traveler behavior and traffic congestion on the specified network. This tool is classified as a microscopic simulation model, as it allows the movement of individual vehicles within a road network by accurately replicating driver behavior [19] [21]. However, when generating large simulation scenarios, the data becomes complex and difficult to analyze. Additionally, the complexity is increased due to the lack of advanced visualization capabilities, the need to incorporate external tools, and the inherent challenges of visually representing large volumes of data [22] [23].

This work presents an effective solution to address the traffic visualization problem by developing a visualization web application using MATSim data. This data goes through preprocessing and parsing techniques to generate the files used. In addition, it integrates the powerful Deck GL library to improve performance in large-scale scenarios, allowing interactive visualizations directly in web browsers without the need for additional frameworks. The solution makes it possible to visualize simulations that usually require computational effort on the part of the computer.

1.2 Problem Statement

Although traffic simulators can help plan the construction of new roads, evaluate the efficiency of transportation systems and optimize road signage, there are still challenges to overcome in the generation of displays [24, 25]. On the one hand, during the review of the state of the art (Chapter 3), a notable shortage of tools developed by other authors for the visualization of simulation data was observed. Furthermore, many of these tools are unavailable or lack support, while in other cases, although they are available on the web, they have unmaintained repositories [8, 13, 26]. It has also been observed that, although some of these tools can work with predefined data, when loading their own data, their operation may be defective or even non-existent [12, 10, 14]. On the other hand, in the case of a viewer dedicated to MATSim such as Via[1] [20], an application that was designed to interpret and represent simulation data. However, unlike MATSim, Via is not open source and is marketed by Simunto, which offers three types of licenses; The free license (valid for six months) only provides basic features and limits viewing to 500 agents. The educational and commercial license (valid for twelve months) claims to have

[1] `https://www.simunto.com/via/`

no functionality restrictions and unlimited agent capacity; However, prices vary depending on the accessories to be used and range between $1,000 and $15,000, respectively [27].

Consequently, these mentioned tools lack an available and functional version that is free. In addition, its ability to use computing resources, facing problems related to visualization due to large volumes of data [10]. Unable to integrate external tools and presenting an unattractive interface, its accessibility is limited to qualified users [22, 23]. Therefore, it conditions transportation management researchers and practitioners to interpret and communicate simulation results effectively. A web application can effectively solve the above problem, allowing MATSim users to view and analyze traffic simulation data using browsers efficiently. With this, we demonstrate the importance of data visualization as a valuable technique for analyzing and understanding simulation. Additionally, developing an interactive interface can help users explore the data and discover patterns and trends that are not evident in the raw data.

1.3 Objectives

1.3.1 General Objective

Create a single-page application designed to visualize traffic data produced by MATSim simulation. Through this objective, the project aims to provide a seamlessly integrated solution focused on visualizing traffic data generated by the simulation, using advanced data processing and web development techniques.

1.3.2 Specific Objectives

- Design an algorithm capable of processing MATSim data into a format capable of web integration.
- Create a single-page application capable of receiving data and generating visualizations. In addition, there is a timer for the corresponding analysis.
- Develop a traffic visualization tool with an interactive user interface. Implemented with HTML, CSS, and JavaScript

1.4 Contribution

This work presents a visual application that addresses the need to achieve visual analysis of complex data generated by traffic simulators such as MATSim. A topic led by Via Simunto, however, its limitations in the free license and high costs per module make it unattainable for scientific groups that do not have financing. Therefore, using information parsing and visualization techniques, a single-page application called TrafficSim-Vis[2] is achieved. The application allows you to view and analyze behavior patterns of simulated vehicles interactively, and can also be used with various MATSim simulations. On the other hand, this work has a conference and article published [28] under the name "Visual Analytic of Traffic Simulation Data: A Review" as part of the 6^{th} Ibero-American Congress, ICSC-Cities 2023, Mexico City and Cuernavaca, Mexico, November 13– 17, 2023. This article presents the state of the art reviewed in this thesis (Section 3), as described by previous work in the development of simulation data visualizations, which served as inspiration for the development of TrafficSim-Vis.

1.5 Document Organization

This work has six main parts. A summary is presented below.

- Chapter 1: Introduction. This chapter introduces the topic and then discusses the problem addressed by the development of this work. The objectives and organization of this paper are presented.
- Chapter 2: Theoretical Framework. This section provides definitions for understanding the work and establishes the theoretical foundations on which it is based.
- Chapter 3: State of the Art. This chapter discusses a variety of work that has had a significant impact on traffic visualization and MATSim. The problems it addressed, as well as its tools and techniques, are examined.
- Chapter 4. Methodology. The methodology used in this work is presented in this

[2] `https://chrisaloor.github.io/TrafficSim-Vis/`

section. The process of creating the web visualization tool and its utilities are presented.

- Chapter 5: Results and Discussion This section presents the findings of this study and discusses various factors involved in creating and applying the web-based traffic visualization tool.

- Chapter 6: Results and Future Work. This section discusses the results and provides an overview of the research that will build on this work in the future.

Chapter 2

Theoretical Framework

The theoretical framework of data visualization will be discussed in this section. Combining technologies such as Deck GL and MapLibre provides a powerful platform for creating interactive and entertaining visualizations. The theoretical framework will serve as a conceptual and technological basis for developing a traffic data visualization and simulation tool. An explanation of why data visualization is crucial for understanding patterns, trends, and relationships in complex data sets will be given. The benefits of interactive visualization tools, which facilitate real-time data exploration and understanding, will be described.

2.1 Traffic Data

The search for good vehicular mobility in large or medium-sized cities has become essential for everyday activities. As cities have developed, so have the distances and times between two places. According to forecasts by the Organisation for Economic Co-operation and Development (OECD), by 2050, the total demand for urban passenger transport will have doubled compared to 2015 [29]. This indicates that traffic congestion on highways is often attributed to a limited perspective on the future development of a city, which implies a significant loss of time and resources for the users of these routes [30]. Therefore, the need arose to develop systems capable of interpreting and managing traffic [31]. However, to achieve this, the collection of genuine traffic data was required, which constitute sets of information collected by monitoring devices installed in vehicles or along roads, such as GPS [4], vehicle sensors [32] or monitoring systems using video and images[5].

Collecting daily traffic data allows identifying driver behavior patterns to detect risky situations and prevent accidents [3]. As time progressed, technological progress enabled the development of robust computational models capable of simulating highly realistic traffic scenarios [23]. Moreover, the need to go further and focus on smart cities with sustainable development has led scientists to turn their attention to traffic simulation data [33].

2.2 Traffic Simulation Data

The constant technological progress has enabled the development of robust computational models that can simulate very realistic traffic scenarios. These simulations range from vast and complex scenarios to small and individual scenarios [23]. Bruce Greenshields [34] was the first person to present a traffic flow model in 1935, focusing his research on the Ohio State Highway. Since then, many scientists and researchers have developed models and simulation tools [21]. Today, simulation-generated data has provided a wealth of insight to researchers, even providing helpful information for city planning. According to Liu [35], a simulation that integrates big data and real-time data, capable of predicting, controlling, and monitoring any vehicle activity, is expected to be achieved in the future.

As shown in Figure 2.1, several traffic simulation models have been created, but their development derives from the basis of the Fundamental Diagram proposed by Greenshields [34]; macroscopic, mesoscopic, and microscopic models. On the one hand, the macroscopic model defines a total road flow characterized by a low level of detail in the transport infrastructure and traffic flow [36]. In other words, its attributes are presented in a generalized form. On the other hand, a microscopic model is characterized by representing individual vehicles' behavior and interactions. It provides a high level of detail in the traffic flow [35, 36]. Finally, the mesoscopic model adopts characteristics of the macro model with its generalizations and the micro model with its vehicle movements.

Most traffic simulation programs opt for the microscopic model since they focus on details that define the behavior of individual drivers. As we use data from a microscopic model in this work, this model is presented in more detail in the next section.

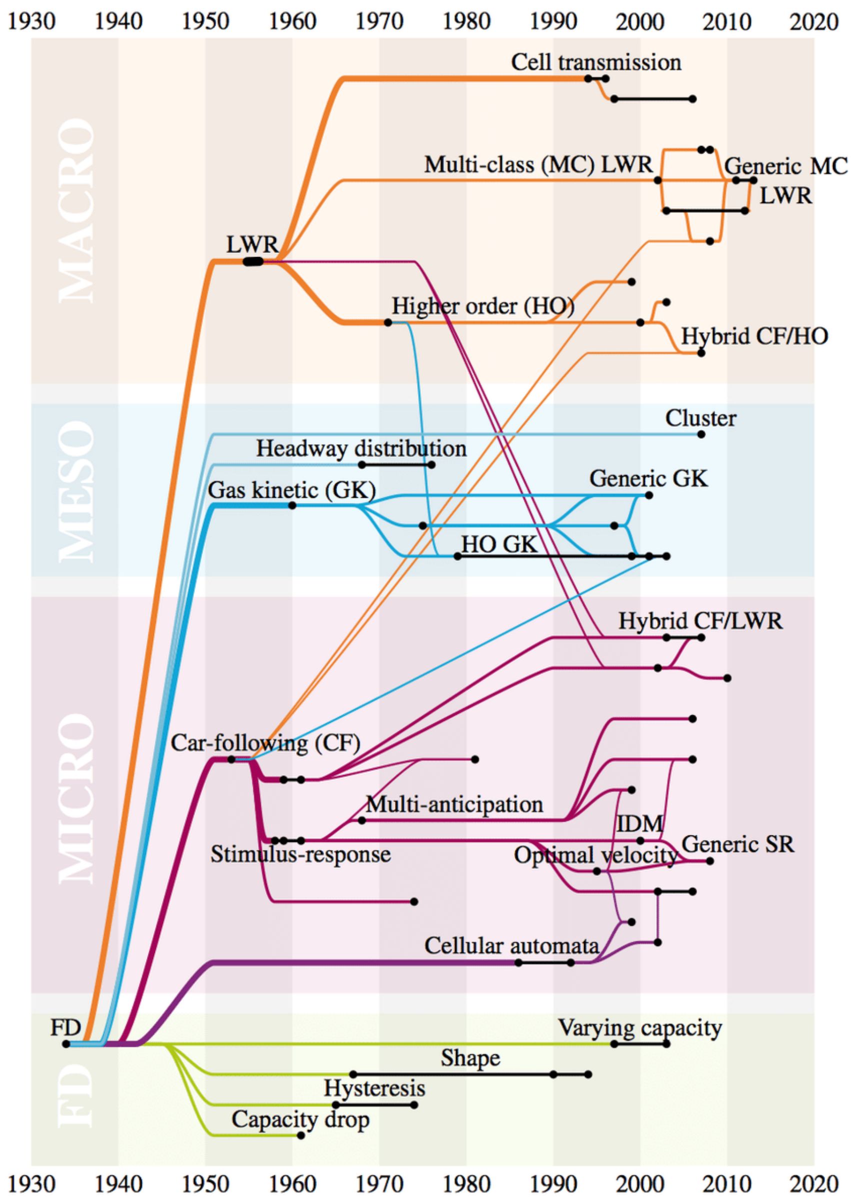


Figure 2.1: Genealogy of traffic flow models. Taken from [1]

2.2.1 Microscopic Simulation

This kind of simulation requires a large amount of data to provide very detailed event results and became popular with the advent of powerful computers capable of performing many calculations. The main objective of microscopic traffic models is to simulate the realistic flow of individual vehicles (agents) through a road network [37], as shown in Figure 2.2. Most microscopic simulators use vehicle tracking, lane changing, and gap acceptance models to simulate vehicle behavior [38].

This means vehicle movements concerning the surrounding traffic must be calculated

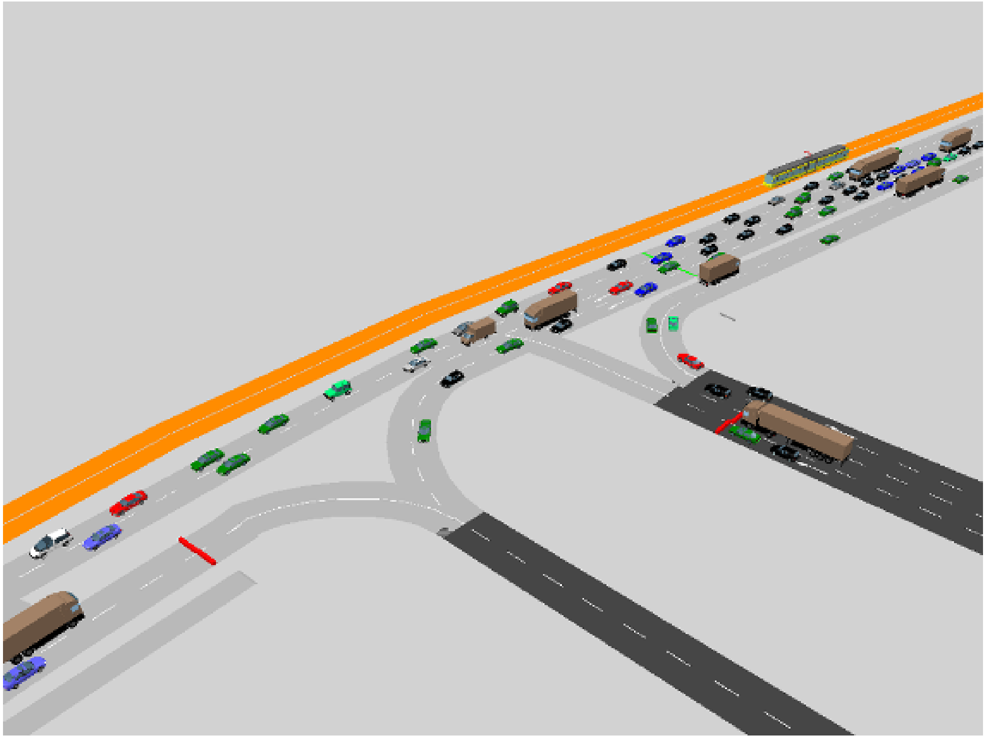


Figure 2.2: Microscopic traffic simulation

at each simulation step. In addition, collect parameters such as flow, density, speed, travel time and delay, long queues, stops, pollution, and fuel consumption. This type of simulation is a powerful and versatile traffic analysis tool [35, 36, 38]. Tools such as Multi-Agent Transport Simulation (MATSim)[1], Simulation of Urban MObility (SUMO)[2], and Multimodal Traffic Simulation Software (VISSIM)[3] employ microscopic simulations. Next, we detail in a better way the simulator that we use for the generation of the data that we preprocess and that will enter into our traffic data visualizer.

2.3 Traffic Data Features

Traffic simulation applications are an invaluable source of various types of data, among which are essential features to achieve distinctive attributes that allow it to reach proper traffic data. To put them in context, in Table 2.1, we have detected five fundamental aspects for the analysis of simulation data: time, space, spatio-temporal, multivariate, and

[1] https://www.matsim.org/
[2] https://eclipse.dev/sumo/
[3] https://www.ptvgroup.com/en/products/ptv-vissim

speed. These data are elements the simulators share and highlight their importance in developing simulations.

Table 2.1: Traffic data features description

Feature	Description
Time (t)	Indicates the change in the course of the data
Space (s)	Location and geographic distribution
Spatio-temporal (x)	Representation of traffic evolution over time and space
Multivariate (m)	Several objects or events moving simultaneously
Velocity (v)	Reflects the speed of objects or events traveling along

2.4 MATSim

MATSim is an open-source tool for simulating transport agents (vehicles) that behave individually. These agents participate in the description of characteristics that can affect traffic in a global way [20]. MATSim starts by assigning a plan for each event, creating an initial demand. The simulation is executed step by step according to the plan; the plan's performance is scored at every moment. Next, the iteration generates a new plan or replanning that allows alternative decisions based on the plan scores. Finally, there is the analysis of the plans that have been executed (see Figure. 2.3).

In other words, agents perform daily activities, such as commuting to work, shopping, or returning home. However, agents must travel due to the distance between the corresponding activity locations. This entails many extra features for the agent, such as travel mode, activity duration, location, and route choice. In this way, the agents describe the traffic of the entire area [39, 22].

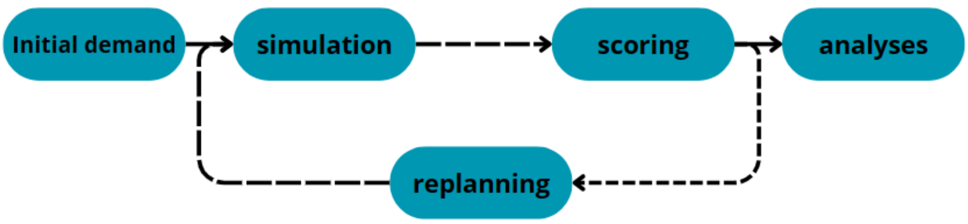


Figure 2.3: Iterative simulation process of MATSim

The simulation data generated by MATSim are complex, challenging to analyze, and caused by the set of configurations previously made. Among the output files generated are counts, events, facilities, households, legs, network, person, plans, and vehicles, among others [20]. However, we concentrate on two files for this work: networks and events. In the following, we will explain what these files contain and why they are essential for the visualization.

2.4.1 Network

The input/output network file (*.xml) describes a graph of nodes and links. In turn, this graph represents the area or city configured for the simulation. This file is important because it contains information about the transportation system's routes, streets, intersections, and connections between nodes [13]. Additionally, it contains geospatial and topological data that describes the location and connectivity of transportation routes, such as nodes, links, link types, capacity, speed limits, and other relevant attributes (see Figure. 2.4). Its use is essential since it defines the infrastructure on which the agents (vehicles) move [20].

```xml
<nodes>
    <node id="1000916573" x="724210.0883466646" y="9680394.5530272" >
    </node>
    <node id="1000916592" x="724280.914322159" y="9680457.326553036" >
    </node>
</nodes>

<links>
    <link id="1005493150004f" from="1162464733" to="1162464562" length="203.6683087"
        freespeed="4.16667" capacity="600.0" permlanes="1.0" oneway="1" modes="car" >
        <attributes>
            <attribute name="origid" class="java.lang.Long">100549315</attribute>
            <attribute name="type" class="java.lang.String">residential</attribute>
        </attributes>
    </link>
</links>
```

Figure 2.4: Network file sample from MATSim.

2.4.2 Events

This output event file (*.xml) stores an ordered succession of events occurring during the simulation. Figure.2.5 shows how each entry represents a specific event with detailed information, such as the time, the agent involved, and the coordinates where it is located [20]. These events describe the different actions of each agent during the simulation and represent in a general way the movements, arrivals at destinations, collisions, changes of direction, traffic, departure of a car, and more [7]. This information leads to the recording of agent behavior, identification of traffic flow problems, congestion assessment, and detailed investigations of transportation system performance. Therefore, the event file is essential for post-simulation analysis and even more so for web visualization; that is the purpose of this work.


```xml
<event time="17955.0" type="entered link" link="489085100014f" vehicle="013"  />
<event time="17956.0" type="left link" link="4062486880013f" vehicle="1810"  />
<event time="17956.0" type="entered link" link="806768630001f" vehicle="1810"  />
<event time="17961.0" type="actend" person="116" link="4000498370006f"
    x="721166.31" y="9680597.76" actType="h4"  />
<event time="17961.0" type="departure" person="116" link="4000498370006f" legMode="car"
    computationalRoutingMode="car"  />
<event time="17961.0" type="PersonEntersVehicle" person="116" vehicle="116"  />
<event time="17961.0" type="vehicle enters traffic" person="116" link="4000498370006f"
    vehicle="116" networkMode="car" relativePosition="1.0"  />
```


Figure 2.5: Event file sample from MATSim.

2.5 Data Visualization

One of the earliest records mentioning data visualization was given by McCormick *et al.* [40], who defined it as computation done through visualization. He also mentions that it converts the symbolic into geometric so that scientists can see what their calculations and simulations look like. It encourages profound and surprising discoveries and enhances the scientific discovery process. Likewise, according to DeFanti *et al.* [41], data visualization is a pervasive area and encompasses many topics of scientific interest. He also emphasizes the remarkable progress that it will make through the years. Therefore, combining computer and human intelligence, achieving data interpretation, and initiating visual analytics is possible [23].

Visual analytics, on the other hand, refers to the process of analyzing data through visual representations. As described by Kohlhammer *et al.* [42], it is a branch that has as its primary objective to handle giant data sets of information by combining them with tools that achieve robust automatic analysis and generating correct human visual analytics.

As [42] goes on to clarify, To be more explicit, visual analytics is the creation of techniques and tools that enable people to:

- Combine data and extract meaning from vast, changing, unclear, and frequently contradicting sources.

- Recognise the anticipated and unearth the surprising.
- Deliver assessments in a timely, understandable, and justifiable manner.
- Effectively convey these assessments so that action can be taken.

To do this, intelligence is introduced into the analytical process through user-driven learning and visualization [43], followed by an exploration process for visual analysis, as illustrated in Figure. 2.6. In the first stage, the raw data can be accessed by preprocessing and obtaining the information. Next, the data is visually transformed into symbols or graphs containing the characteristics of the data. Finally, the visual models are assembled into various visualizations, such as images, infographics, animations, and videos, and mapped to various visual channels. Because the procedures are iterative, the model can be identified and improved upon at each stage until the desired visualization is obtained [23, 42].

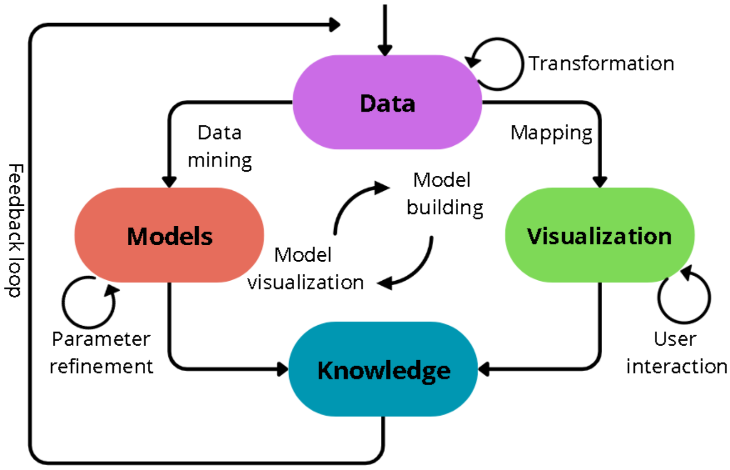


Figure 2.6: Visual analytics process is typified by interactions between data, visualizations, models about the data, and users.

Over time, visualization techniques have become more robust and efficient, allowing users to reveal patterns and trends in data through graphical representations [44]. Also, transform complex data into actionable data intelligently, logically, and understating to optimize the analysis of large amounts of data [45]. This will improve information understanding and decision-making based on concise information [32]. To address the needs of analytical applications that handle large amounts of data, born Deck GL.

2.6 Deck GL

This is an open-source framework focusing on high-performance WebGL-based visualization of large datasets [46]. Its main advantage is allowing users to achieve complete visualizations effortlessly by combining pre-existing layers or using the extensible architecture to create their layers (see Figure. 2.7). It usually represents JSON objects in layers, one on top of the other, adding icons, polygons, and graphics. It also explores and analyzes different visualizations that it can render. In this way, it allows users to obtain meaningful information for various data sets [46, 2].

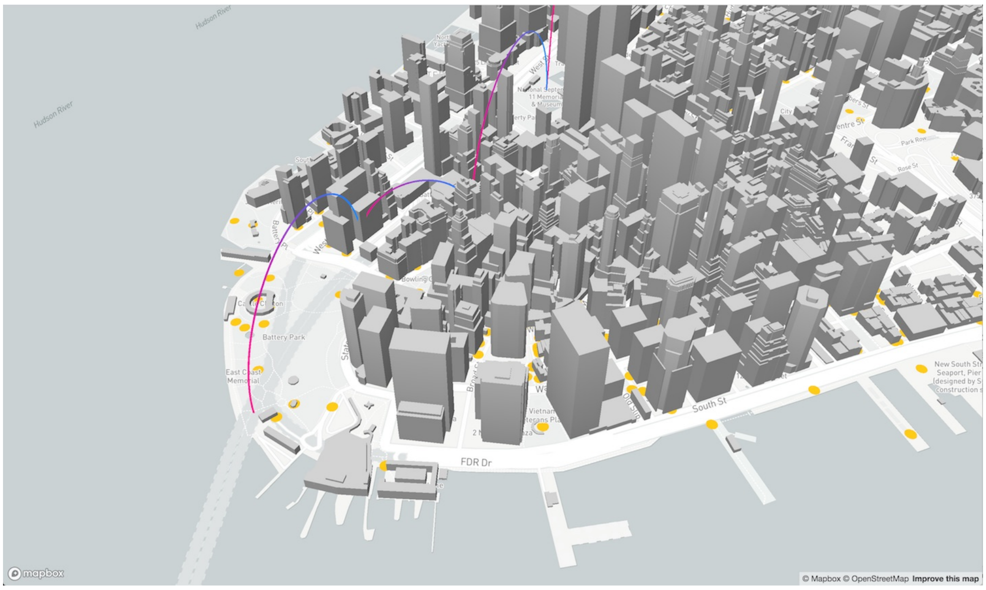


Figure 2.7: Deck.gl usage example [2]

Among the advantages of Deck GL:

1. Enables efficient display and updating of extensive datasets
2. Interactive event management, including filtering, highlighting, and selection
3. Projections of cartography and interaction with key base map providers
4. A list of validated, tried-and-true layers

2.6.1 Layers

Deck GL layers are made to be highly configurable. Adaptable APIs are available for each layer, enabling programmatic control over all aspects of the representation [46]. All basic classes are freely extensible by users to handle particular use cases. There are different layers that we will mention below:

- ArcLayer
- ContourLayer
- GeoJsonLayer
- HeatmapLayer
- HexagonLayer
- LineLayer
- IconLayer
- ScatterplotLayer
- ScenegraphLayer
- ScreenGridLayer
- TerrainLayer
- TextLayer
- TileLayer
- Tile3DLayer
- TripsLayer

In the context of this work, we have pinpointed the GeoJsonLayer and TripsLayer as pivotal elements that significantly align with our objectives. Their distinctive features make them integral to our visualization strategies, and we will now proceed to provide an in-depth exploration of these layers, describing their features and functionalities in detail for enhanced clarity and comprehension [2].

2.6.2 GeoJsonLayer

This layer provides a valuable tool for visualizing geospatial data stored in GeoJSON format, allowing us to efficiently represent polygons, lines, or points [2]. As we can see in Figure 2.8, it is based on JavaScript Object Notation (JSON). It defines several types of FeatureCollection objects and how they are combined to represent data about features using a geographic coordinate reference system, their properties, and their spatial extents [47]. Next, a visualization is presented in Figure 2.9, in which they implement the GeoJsonLayer called Highway Safety in the US using data from the National Highway Traffic Safety Administration, which shows fatal crashes every 1,000 miles on US highways between 1990 and 2015 [2]. The bar representing the frequency of crashes turns red when the number of crashes is higher and green when the crashes are lower.

```
{
    "type": "FeatureCollection",
    "features": [{
        "type": "Feature",
        "geometry": {
            "type": "Point",
            "coordinates": [102.0, 0.5]
        },
        "properties": {
            "prop0": "value0"
        }
    }, {
        "type": "Feature",
        "geometry": {
            "type": "LineString",
            "coordinates": [
                [102.0, 0.0],
                [103.0, 1.0],
                [104.0, 0.0],
                [105.0, 1.0]
            ]
        },
        "properties": {
            "prop0": "value0",
            "prop1": 0.0
        }
    }

   ]
}
```

Figure 2.8: GeoJsonLayer structure format

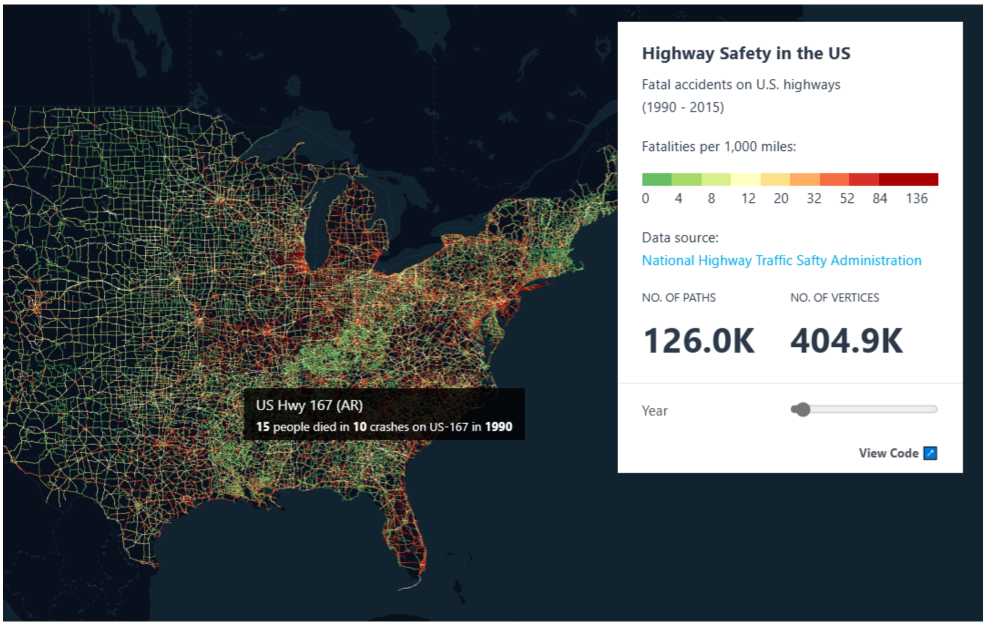


Figure 2.9: GeoJsonLayer usage example: Highway Safety in the US [2]

2.6.3 TripsLayer

This specialized layer is meant to visualize and analyze temporal trajectories or sequences of events over time. For this reason, it is helpful when working with motion data, including vehicle trajectories, GPS traces, and other phenomena that may be mapped to geographic coordinates [2, 46]. For data accessors, see Figure. 2.10, we have path and timestamps, in which the path is an array of navigation points. On the other hand, timestamps save time for each point navigation. Now, Figure. 2.11 shows Green Cab Trips in Manhattan of data from the NYC Taxi and Limousine Commission Trip Records, representing the vehicles in orange. In addition, you can see the GeoJsonLayer from OpenStreetMap data to mount the buildings.

```
[
  {
    "id": 10,
    "path": [
      [-79.00179350001582, -2.89652619999986],
      [-79.00161120001582,-2.895546799999862],
      [-79.00160520001582,-2.895510999999877],
      [-79.00142220001582,-2.894498399999876],
      [-79.00122750001584,-2.893495499999872]
    ],
    "timestamps": [
      20948.0,
      20949.0,
      20950.0,
      20978.0,
      21006.0,
      ]
  },
  {
    "id": 11,
    "path": [
      [-78.99641560001604,-2.892203399999866],
      [-78.99649360001604,-2.893122499999867],
      [-78.99651450001603,-2.893428499999868]
    ],
    "timestamps": [
      24056.0,
      24057.0,
      24062.0,
     ]
  }
]
```

Figure 2.10: TripsLayer structure format

Figure 2.11: TripsLayer usage example: Green Cab Trips in Manhattan of data from the NYC Taxi and Limousine Commission Trip Records [2]

2.6.4 Base Map

Integration of layers in Deck GL functions autonomously from other cartographic elements. However, including a base map is frequently essential for precisely rendering geospatial datasets. There are numerous base map providers offering a wide array of options,

including Mapbox[4], Google Maps[5], Leaflet[6] and MapLibre[7] [2, 46]. For this work, we have selected MapLibre because it fulfills the conditions of being accessible, flexible, and with a great variety of map styles in front of it.

To integrate Deck GL with a rendered base map, we have opted for the technique called **Overlaid** (see figure 2.12). In this approach, the Deck GL canvas is superimposed on the base map as a separate element, providing seamless integration of the geospatial data visualization. This method allows the interactive Deck GL layers to be overlaid on the rest of the map and ensures synchronization between the canvas layers and the rest of the cameras. This synchronization facilitates the coordination of panning and zooming, improving the overall user experience by maintaining spatial consistency between layers.

[4] `https://www.mapbox.com/`
[5] `https://www.google.com/maps/`
[6] `https://leafletjs.com/`
[7] `https://maplibre.org/`

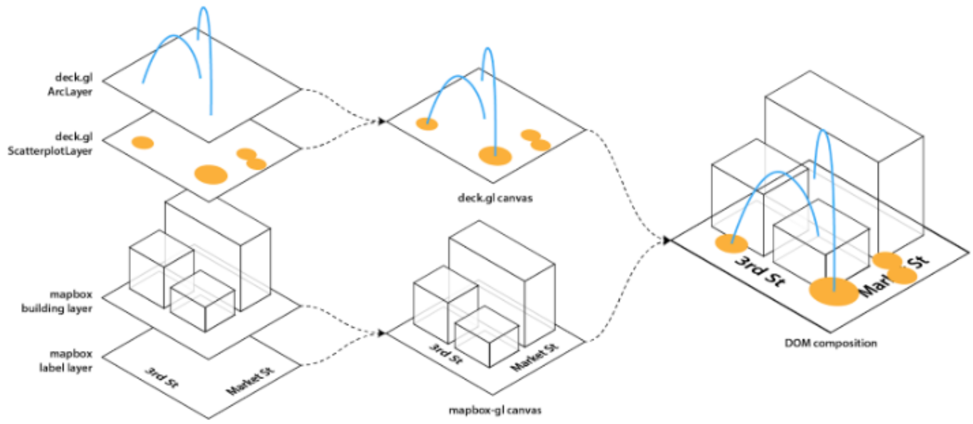


Figure 2.12: Overlaid: integration between deck layers and the base map. [2]

MapLibre

MapLibre GL is an open-source library based on Mapbox GL JS. It offers similar functionality, including rendering interactive vector maps in the web browser using Web GL for fast and fluid performance [48]. This is an excellent choice for this job because it uses high-quality online map display technology without licensing restrictions. Additionally, MapLibre GL supports a wide range of data sources, including OpenStreetMap (OSM). OSM is a collaborative online mapping project that creates and provides free, editable geospatial data worldwide [49]. So, its result makes it ideal for various geospatial data visualization applications.

Chapter 3

State of the Art

This chapter introduces and scrutinizes initial research concerning visualization in traffic simulation. Furthermore, Section 3.1 outlines the methodology used to compile and analyze these studies, culminating in the presentation of a Table 3.1 with the most relevant features (See Section 2.3) taken from the articles for their visual representation 3.1. The structure of these subsections is delineated as follows: Section 3.2 showcases visualizations derived from real scenarios data, while Section 3.3 focuses on visualizations derived from simulated data. Following this, a discussion of pertinent articles and their influence on the research conducted for this work will ensue.

3.1 Methodology of article collection

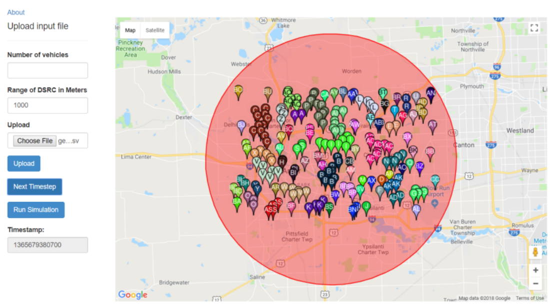
In this work, an extensive exploration was undertaken to pinpoint scholarly papers incorporating actual and simulation-generated traffic scenario data. The inquiry was executed across reputable databases such as Google Scholar, Scopus, and IEEE, employing the Boolean method to enhance search efficiency and precision, a technique acknowledged for its capacity to yield more pertinent results and valuable sources [52]. Then, the keywords used in the method of search were "data traffic", "simulation", and "data visualization" consequently, the search resulted in various articles, of which only those that developed a data visualization system were taken. As a result, a set of 15 related articles was obtained between 2013 and 2023, and they exclusively addressed our search.

In the article selection process for our study, emphasis was placed on incorporating those that delved explicitly into the realm of traffic data visualization. This strategic approach enabled us to concentrate our investigation on research endeavors to achieve a lucid and compelling representation of traffic-related data—an imperative facet in traffic management and decision-making. Consequently, the search encompassed articles utilizing traffic data from real-world scenarios and simulations, focusing on MATSim. This deliberate inclusion facilitated the comparison of diverse implementation approaches. Furthermore, the authors conducted a meticulous examination of the features of traffic data underscored by authors as pivotal in developing visualizations.

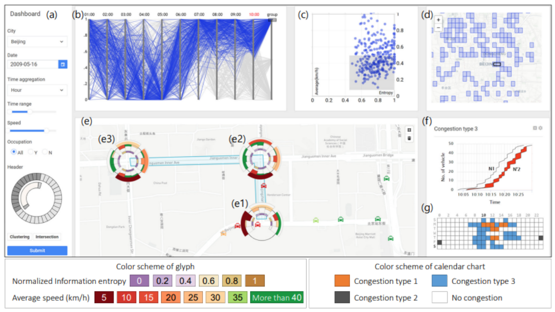
3.2 Traffic visualization of real scenarios

To begin with, let's look at the articles that used real data to generate traffic visualizations. Carter *et al.* [3] present two applications that have been enhanced and developed to augment vehicular networking research. On the one hand, the first application described in the report is the "Multi-hop connectivity simulation," which includes the ability to visualize traffic and mobility of connected vehicles. The application uses the Basic Safety Message (BSM) protocol for vehicle-to-vehicle communication. It allows researchers to simulate different traffic and mobility scenarios to evaluate the performance of vehicular networks. On the other hand, the second application allows any Internet-connected device to remotely monitor a roadway intersection's state over HTTP. It also discusses the importance of vehicular networking research for developing connected and autonomous vehicles. It is noted that research in this field is essential for improving road safety, reducing traffic congestion, and improving transportation efficiency (See Fig 3.1a). Then, in 2019, Pi *et al.* [4] introduced a groundbreaking method for examining the roots of traffic congestion, drawing on traffic flow theory. By employing information theory entropy to pinpoint shifts in vehicle movement and categorizing them into four congestion types using a convolutional neural network, this approach facilitated the identification of congestion arising from factors like traffic signal systems, bottlenecks, and vehicles surpassing road capacity. Additionally, an examination of congestion effects, such as travel delay duration and the count of vehicles undergoing congestion, was conducted by developing a visual analysis system (See Figure 3.1b).

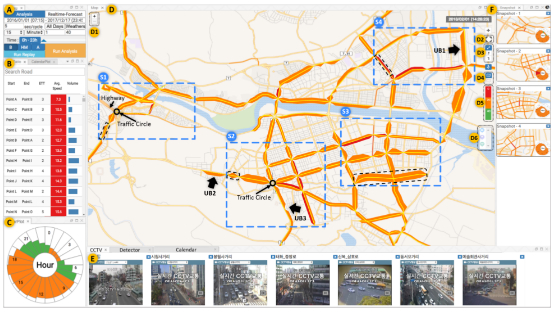
In the same year, Lee *et al.* [5] presented a system that enables users to explore



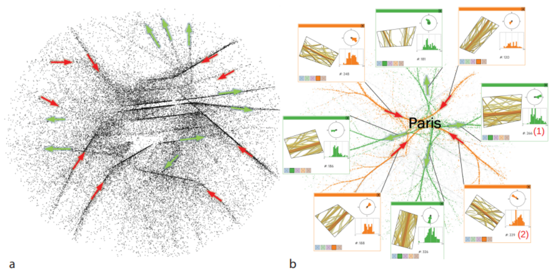
(a)



(b)



(c)



(d)

Figure 3.1: Visual representations crafted by different articles using real traffic data. (a) Displays a simulator interface with a high density of nodes in a given area. [3], (b) Dashboard visual cause analytics system for traffic congestion [4], (c) View of the dashboard with the tracking cameras on an avenue [5], (d) Overview of traffic flows on the most congested streets in the Paris region [6].

and predict traffic congestion based on vehicle detector data. It uses a combination of machine learning models, interactive visualizations, and user feedback to provide insights into traffic patterns and congestion. It also includes details on the system architecture, data preprocessing, machine learning models, and interactive visualizations. Finally, this article provides case studies and domain expert feedback to demonstrate the system's effectiveness (See Figure 3.1c). Crossing from road traffic to seas, Scheepens *et al.* [6] achieves an approach to retrieve information from data sets of moving objects, such as ships and airplanes, using interaction techniques. Several user tasks are identified and described according to the typology of visualization tasks. Visualization, data selection, and analysis are discussed, and use cases are presented to validate the approach. The text also describes how traffic flows can be compared and their dynamics explored. Overall, an approach to visualizing and analyzing data sets of moving objects that may be useful for land and maritime traffic experts is presented (See Figure 3.1d).

Finally, Petrovska *et al.* [26] describes an investigation of vehicle energy consumption on a haul road and how it can be addressed through visual analysis. It uses simulated traffic scenarios to develop a methodology for monitoring and analyzing vehicle driving patterns and estimating their associated energy consumption for acceleration and braking. A web-based visualization dashboard is presented to enable exploration and analysis of vehicle energy consumption patterns in response to spatially and temporally varying traffic signal phases at multiple intersections along a major transportation highway.

3.3 Traffic visualization of simulation scenarios

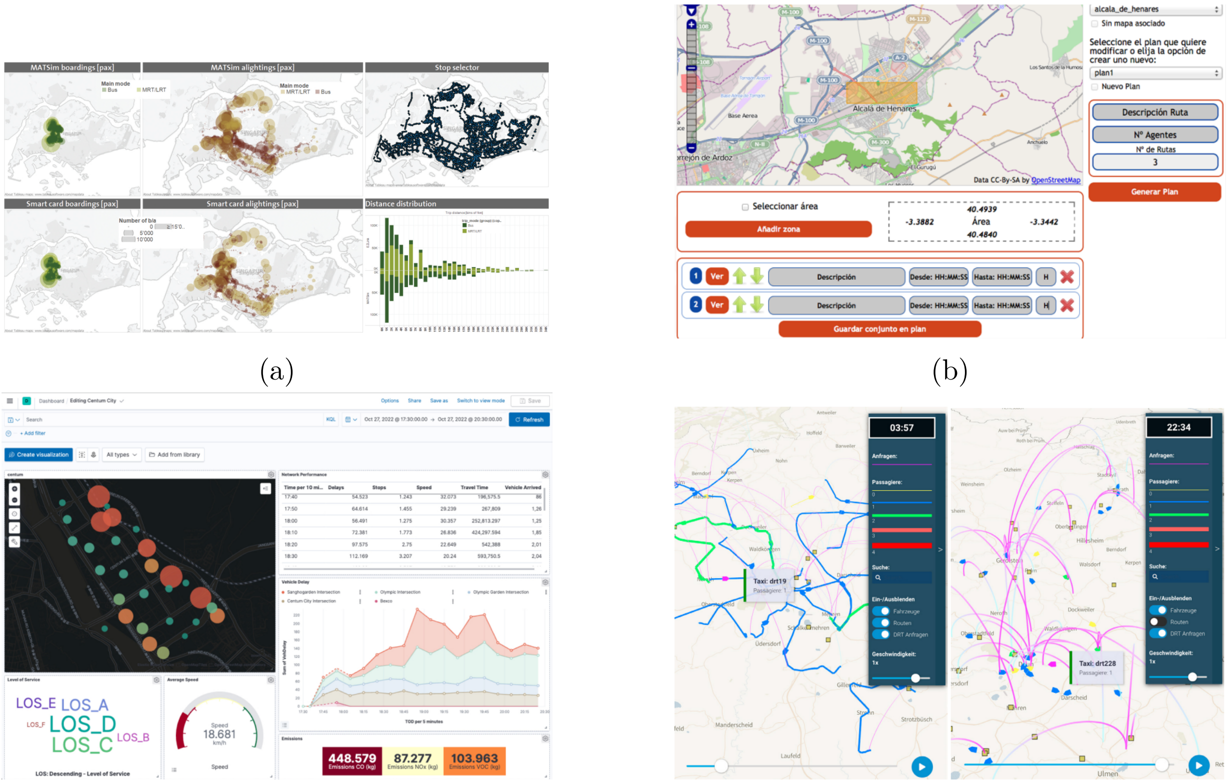
Continuing with the articles, we now look at the articles that used data from traffic simulations to generate visualizations. Xu *et al.* [11] introduces an interactive tool to help transportation planners manage, share, and visualize large amounts of traffic simulation data. The speed control algorithm is incorporated into multiple VISSIM-based simulated scenarios, and its effectiveness can be demonstrated to project collaborators and non-technical participants through easy-to-understand traffic flow visualizations. The application is designed with a modular architecture and a scalable database, making it generalizable and maintainable. It can be easily extended to store and manage outputs produced by traffic simulation models and software tools (See Figure 3.3a). The same author years earlier, Xu *et al.* [50] presented a web-based tool that allows transportation planners to explore and analyze energy consumption patterns resulting from traffic signal phases at multiple intersections along a major transportation corridor. The dashboard enables systematic procedures for estimating the vehicle acceleration and braking energy consumption from user-uploaded traffic simulation outputs. It visualizes the variability of the energy consumption associated with individual vehicle movements, multiple driving directions at an intersection, and signal phase changes at numerous intersections.

Next, Jung *et al.* [9] describes how transportation experts use data visualization to discover patterns and gain valuable information about city traffic. It also mentions how simulation is used to predict the impact of plans on traffic and how simulation results can be presented visually on a dashboard. The authors present a visualization platform called Kibana that allows users to build and share their dashboards in real-time. The platform was developed for Centum in South Korea and is a practical tool for decision-makers to analyze and predict traffic conditions in real time (See Figure 3.2c).

Starting with visualizations that used MATSim simulations. Erath *et al.* [7] introduces an interactive analysis and decision support platform for MATSim simulations. It enables the construction of custom data cubes by constructing SQL queries using graph query construction software. These data cubes are fed into Tableau business analytics software, designed with a relatively programming-agnostic audience in mind. In a simple, well-documented GUI, the user can build dashboards that summarize information and allow interactive aggregation or drill-down through drag-and-drop interaction (See Figure 3.2a).

Another MATSim visualization platform is Charlon *et al.* [10], developed for use by public transportation agencies studying future scenarios of automated and connected shared cab fleets. This platform uses the agent-based transportation microsimulation model to explore scenarios, including the current state and dynamically dispatched fleets with drivers. The platform uses many advanced web technologies, such as VueJS, a popular JavaScript tool for building interactive websites, and the Deck GL visualization library for creating agent-based visualizations. The platform is easy to use and designed to be accessible to non-technical users, such as those who might attend a public outreach meeting (See Figure 3.2d). Piris *et al.* [8] focuses on creating a web application that facilitates the input, output, and interpretation of the data needed to work with the simulator. This application uses different frameworks for the generation of simulation data. We have OSM, canvas, and OpenLayers, among others. Its purpose is to animate the generated data and the data coming from the simulator, providing a high degree of compatibility with any system using a modern web browser. The application has been implemented using JavaScript functions on the client side and Servlet and Java functions on the server side (See Figure 3.2b).

On the other hand, Charlton *et al.* [12] features an entirely web-based platform, which means that no additional software installation is required. It offers a variety of visualizations at both the aggregated and disaggregated levels, including interactive maps and scatter plots. These visualizations can help analyze the results of transportation simulations and identify patterns and trends in the data. It uses a client/server architecture, which means that much of the processing is done on the server while the user interface runs in the user's web browser. The platform is also designed to be highly customizable and extensible (See Figure 3.3b). Miranda and Arruda [13] developed an approach for

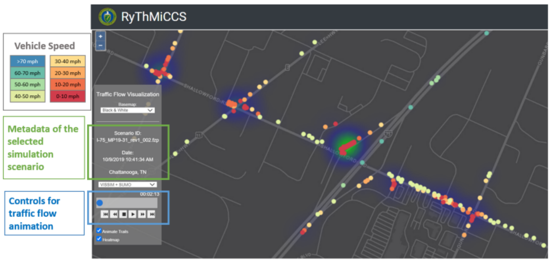


(a) (b) (c) (d)

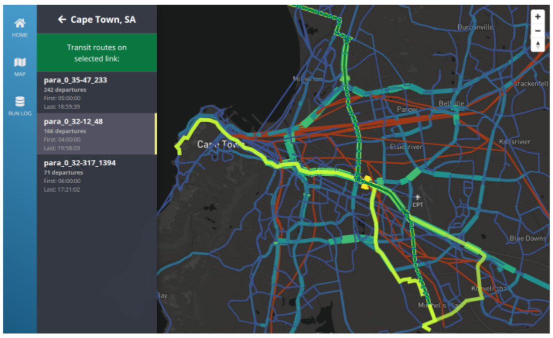
Figure 3.2: Visualizations generated with simulation data. (a) Visualization of a MATSim simulation in Tableau about the number of public transport users [7]. (b) The initial screen of the visualization tool in the description of the plans [8]. (c) Dashboard interface created in elastic [9]. (d) Dynamic response transit system (DRT) animation, vehicles, routes, and destinations [10].

acquiring and handling data about input details within a granular model in developing countries. They further showcased an illustrative scenario within the Brasilia region, the capital of Brazil. The processed information encompassed the road infrastructure and a synthetic population, followed by a subsequent analysis of the area and examination of the traffic data. Demand files are also needed to create a synthetic population. Acquiring information is challenging in developed countries due to the lack of resources and the specificity of the data required (See Figure 3.3c).

Another platform developed by Charlton *et al.* [14] is SimWrapper, an open-source, web-based data platform designed for researchers who want to build interactive dashboards that display the results of their simulations. It produces a wide variety of interactive graphics, maps, and dashboards that are generally useful in the transportation domain. According to feedback in the article, users have found the platform to be flexible and easy to configure via text files, allowing for efficient development and deployment of repeatable and deployable interactive web-based visualizations. However, some users have noted that initial onboarding is too complex, and documentation is always delayed (See Figure 3.3d). Finally, Strippgen [51] presents a data visualizer called OTFVis. According to the authors, this tool was designed to visualize large amounts of data in real time. Therefore, we can say that the main objective was to separate the data source from the data visualization, allowing an easy understanding of one's own data types. Furthermore, the data visualization starts with a writer/reader, who transfers a given data set to the viewer. Therefore, the writer will understand the data format of the simulation system and convert it into a simple format that the viewer can use. The authors mention using 3D graphics hardware to display agents in real time.

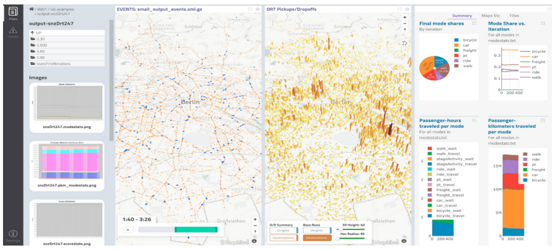


(a)



(b)

(c)



(d)

Figure 3.3: Visualizations generated with simulation data. (a) The user interface includes controls for selecting traffic simulation scenarios and animating traffic flow [11]. (b) Transit routes explorer displays all transit routes and allows the user to see which routes serve in specific links [12]. (c) OTFVis was used to visualize a simulation of morning peak-hour traffic in Brasilia [13]. (d) Web application dashboard where indicated vehicle animation, area data, and basic graphical analysis. [14].

Table 3.1: Summary of the state of art with the relevant features of the systems developed for data visualization (t=time, s=space, x=spatio-temporal, m=multivariate, v=velocity).

Paper	Purpose	Traffic	Features					Data
			t	s	x	m	v	
Carter et al. [3]	Traffic monitoring	Real scenario	✓		✓		✓	Basic Safety Message
Pi et al. [4]	Simulation and prediction		✓		✓		✓	OSM trajectories
Lee et al. [5]	Monitoring and prediction		✓		✓	✓	✓	DSRC data
Scheepens et al. [6]	Trajectories visualization		✓			✓	✓	Vassel trajectory
Petrovska et al. [26]	Traffic analysis and visualization		✓		✓	✓		Google Maps
Xu et al. [11]	Visualization of large-scale traffic	Simulation	✓	✓				VISSIM and SUMO
Xu et al. [50]	Traffic pattern analysis		✓		✓		✓	
Jung et al. [9]	Selection and visualization		✓				✓	
Erath et al. [7]	Trajectory analysis and visualization		✓					MATSim data
Charlton et al. [10]	Visualization taxi routes		✓	✓	✓	✓		
Piris et al. [8]	Generation of simulation of data		✓	✓	✓			
Charlton et al. [12]	Trajectory analysis and visualization		✓	✓	✓	✓		
Miranda et al. [13]	Generation of simulation data		✓	✓	✓			
Charlton et al. [14]	Visualization of large-scale		✓	✓	✓	✓		
Strippgen [51]	Real-time display		✓	✓	✓	✓		

Chapter 4

Methodology

This chapter details the pipeline process carried out to achieve the established objectives. First, a chronological outline for problem-solving is presented. Next, the proposed design addresses the key aspects developed for the traffic data simulation web application. Finally, scenarios and techniques for interpreting the results obtained are proposed.

4.1 Phases of Problem Solving

The pipeline process used to perform this project is depicted in Figure 4.1.

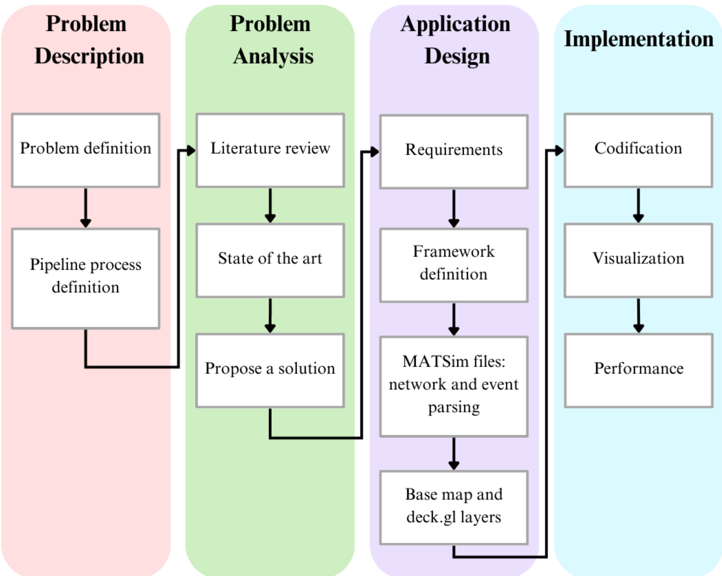


Figure 4.1: Phases of problem solving

4.1.1 Problem Description

In this phase, the integral development of the research is presented in Section 1.2 detailing the central problem to be addressed, identifying the difficulties inherent to the process, and concisely describing how a solution has been proposed. Accordingly, in Chapter 2, we incorporate the fundamental concepts of traffic data, simulation data, microscopic simulation, and MATSim and highlight the relevance of data visualization for this work. We then identify the potential challenges associated with processing large volumes of data, such as those generated by MATSim, and the ability to render them efficiently. Finally, we outline a pipeline process that will be implemented to accomplish this work.

4.1.2 Problem Analysis

Throughout this phase, Chapter 3 presents various articles with essential information to understand how other authors have developed tools to visualize data from simulators. However, those tools are not available; in other cases, although they are available on the web, they present unmaintained repositories. It was also observed that, although some of these tools can work with predefined data, when loading their own data, their operation may be defective or even non-existent. Along with the information provided in the problem description, we have acquired broader knowledge, evidenced by understanding the operation of traffic simulators, as well as data manipulation and visualization. This knowledge was essential to present an accurate and detailed state of the art. Based on this information we have formulated a solution proposal that consists of taking characteristics of the data generated by the MATSim simulator, preprocessing them, and being able to use them for a web visualization.

4.1.3 Application Design

For this phase, we meticulously formulated a set of requirements aligned with the stated objectives described in Section 1.3. Table 4.1 shows the requirements essential to the specific features (see Section 2.3) and capabilities expected of the visualization platform. The identified requirements serve as a comprehensive model, encompassing various facets of the platform to ensure its effectiveness and alignment with the objectives of the work.

Table 4.1: Visualization single-page application requirements

	Requirements	Description
R1	Data Compatibility	Receive data from MATSim, and keep features: time, space, spatio-temporal, multivariate, and velocity
R2	Web library support	Libraries that can efficiently handle large volumes of geospatial data and support temporal analysis
R3	Interactive Visualization	Load data and control simulation time with a timer. In addition, provide information about the data
R4	Open Source	Source code available with freedom to modify, distribute, and improve the software.

To define the framework we would work with, we reviewed different types of libraries with related work exposed until we arrived at Deck GL (see Section 2.6). This library will be the indispensable one to achieve our purpose. In addition, we discussed the MATSim outputs we were interested in studying. On the Deck GL side, it has already provided specific layers (see Section 2.6.1) for network and events data visualizations. However, these required preprocessing to be compatible with the layers.

4.1.4 Implementation

In this phase, the whole implementation is developed; it consists of preprocessing and visualization. The first is an implementation in Python that allows parsing the output files generated in MATSim (see Figure 4.2). Meanwhile, the second is the web application, where the preprocessed files are loaded and, with the help of Deck GL, allows visualization of agents and city roads. In the following, we see the development of both parts.

Part 1: Preprocessing

Throughout this work, we have stressed the importance of analyzing the ”event” and ”network” output files generated by MATSim. Both files present application-specific structures, and they are in compressed format (*.xml.gz). To perform this analysis, it is essential to locate these files inside our workspace with the default names ”output_network.xml.gz” and ”output_event.xml.gz”. The code verifies the presence of both MATSim output files and, once their existence is confirmed, proceeds to perform the parsing steps. The files are converted into data frames with the help of the matsim-python-tools library to facilitate the extraction of the necessary information. Recall that we will obtain two data frames: one called network and one called events. However, within the network data frame, we find two subsets of data: the nodes, which are presented in Table 4.2, and the links, which are detailed in Table 4.3 while Table 4.4 defines the events. Thus, we split the data structure to facilitate the analysis and understanding of the transport network.

Table 4.2: Sub-data frame nodes from network

Name	**Type**	**Description**
node_id	string	Unique node identifier
x	float	x-axis node position
y	float	y-axis node position

Table 4.3: Sub-data frame links from network

Name	**Type**	**Description**
link_id	string	Unique identifier of the link
from_node	string	Identifier of the link start node
to_node	string	Link end node identifier
length	float	Length of the link in specific units
freespeed	float	Free or maximum speed allowed on the link in specific units
capacity	float	Capacity of the link (max number of vehicles that can pass)
permlanes	float	Number of permanent lanes on the link
oneway	int	Indicator of whether the link is one-way (1) or bi-directional (0)
modes	string	Transport modes allowed on the link
origid	string	Original identifier of the link
type	string	Link type (residential, commercial, etc.)

Table 4.4: Event data frame

Name	**Type**	**Description**
time	float	Time at which the event occurred, represented in seconds.
type	string	Type of event that occurred (actend, left link, entered link, etc.)
link	string	Location of the start of the event on the network
vehicle	string	Identification of the vehicle associated with the event

Network

Once we have collected the necessary data in the corresponding data frames, we proceed to generate the network file for subsequent download. In this process, we first convert the x and y coordinates of the nodes to longitude and latitude coordinates in the EPSG 4326 reference system. This is done to avoid possible application integration problems with Deck GL. Subsequently, we perform a mapping of the node identifier data (node_id) to facilitate the efficient extraction of the coordinates of the start (from_node) and end (to_node) nodes of each link. Finally, the result is a GeoJSON file in JSON format containing all the necessary properties, as described in Section 2.6.2. This file meets all the requirements to be successfully uploaded to the corresponding platform.

Event

On the other hand, the event data frame is merged with the links and nodes data frame, identifying key attributes such as "time", "vehicle_id" and "link_id". Through three successive merges, the proper routing of events is established. Finally, this information is transformed into a JSON file structured as TripsJson. This process provides us with two files ready for the next stage, which involves the visualization in the corresponding platform, which we detail as follows.

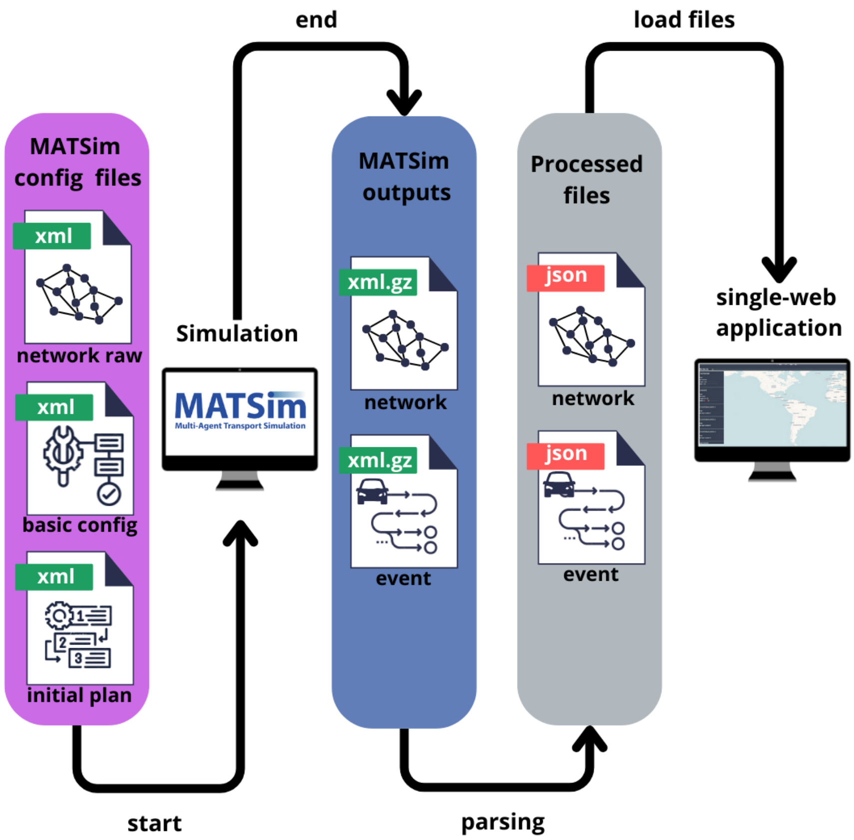


Figure 4.2: Preprocessing data

Part 2: Visualization

The developed single-page application provides a visual representation of traffic simulation data on an interactive map. A set of cutting-edge web technologies, including Deck GL and MapLibre GL, have been employed to create a fluid and feature-rich tool. The application creation process began with importing the various necessary components. Subsequently, we defined key constants and configurations for file loading and status control. Figure 4.3 presents the layout of the layers displayed on the map: base map, GeoJsonLayer and TripsLayer. In addition to the default layers, the application allows dynamic loading of up to three custom layers from JSON files. This functionality gives users the flexibility to incorporate additional custom data into the visualization, further enriching the user experience.

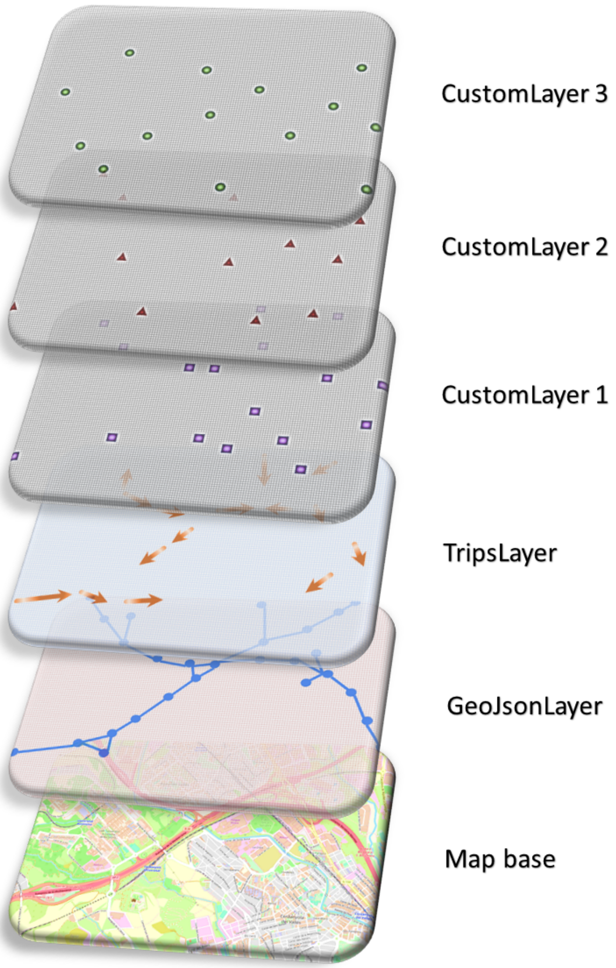


Figure 4.3: Overlapping Layers

4.2 Model Proposal

Based on the requirements outlined in the application design, TrafficSim-Vis is presented, a single-page application for the visualization of MATSim data (see Figure 4.4). As previously mentioned in Table 4.1, this tool involves a preprocessing method to integrate the data into formats compatible with the theoretical framework adopted [R1]. Deck GL, part of said framework, facilitates the efficient management of large volumes of data [R2]. In addition, an interactive interface has been implemented that grants complete control over the displayed data [R3], all using open-source tools [R4].

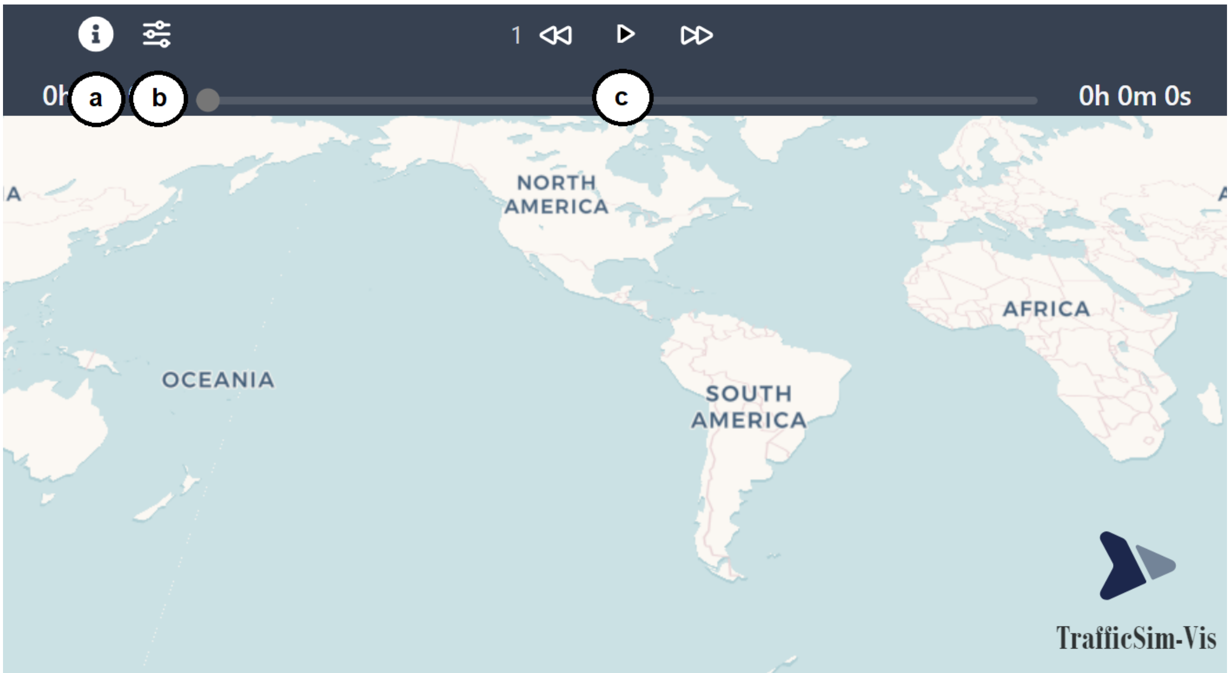


Figure 4.4: TrafficSim-Vis: a) information and specifications of the adjustment elements, (b) layer adjustment tools, (c) simulation playback controls

All visual features implemented in TrafficSim-Vis have been designed to ensure optimal performance and fast response in different browsers. This efficiency is achieved through cutting-edge technologies such as Vite and React. For its part, Vite provides a faster and more efficient development experience for modern web projects, with high framework compatibility allowing developers to achieve good performance in a short time [53]. On the other hand, React, a library created by Facebook and focused on single-page applications, provides the possibility of creating user interfaces with reusable code fragments called components; this makes it very flexible when using other libraries and producing scalable code [54]. By using both in TrafficSim-Vis development, we can dynamically serve the application based on the specific needs of each browser, ensuring a smooth and consistent experience for users and, additionally, providing flexible programming and a wide variety of essential components to create a modern website.

Figure 4.4 shows the first view the TrafficSim-Vis user will perceive. The view shows a sidebar on which there are different buttons previously indicated, and which will be detailed later. Additionally, it shows the base map on which the processed simulation files will be loaded for viewing. However, because no network file is loaded, the initial view is configured by default, being a broad and generalized view of the world.

4.2.1 Information

It was kept in mind that the user would need to have a description of the application and how he could use it. For this reason, an information button was developed, described as (a) in Figure 4.4. This button would be responsible for displaying a modal that provides detailed aspects of the simulation and descriptions of the sliders' characteristics, making it easier for the user to understand the function of each interaction (see Figure 4.5). Once the user has understood these aspects, they are provided with an "Accept" button, which closes the modal and allows the user to continue viewing.

4.2.2 Layer adjustment tools (Sidebar)

The adjustment tools were described in (b) in Figure 4.4. Here, you can view in greater detail the configurations of the different layers that can be integrated into the application. This panel not only offers the ability to view layers but also has sliding functions that can be adjusted for a more impactful and interactive visual representation. It is divided into three sections, covering aspects such as the transportation network, vehicle events, and custom layers. Each of these sections is presented individually below.

Network

This configuration allows adding the parsed network file from the MATSim simulations to a JSON format. Once the file is loaded, the simulation of the corresponding area is automatically zoomed in due to an implementation that detects the first coordinates of

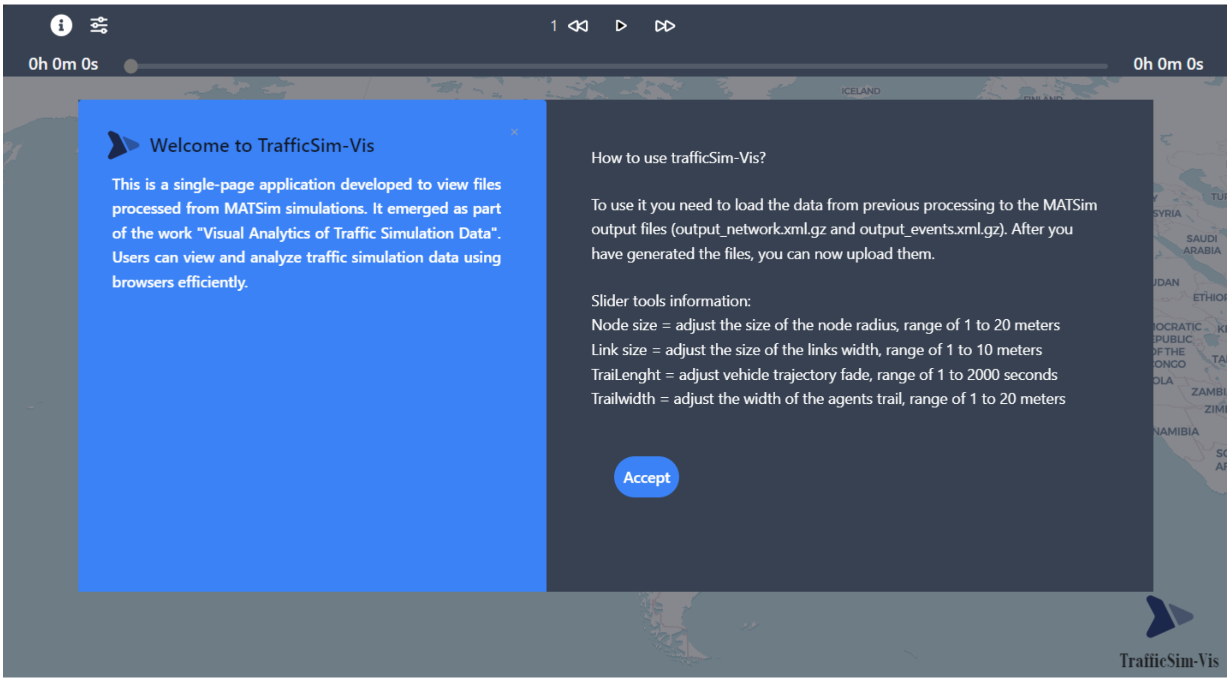
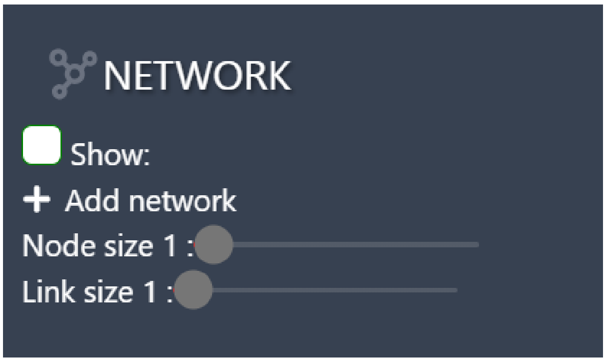


Figure 4.5: TrafficSim-Vis Information

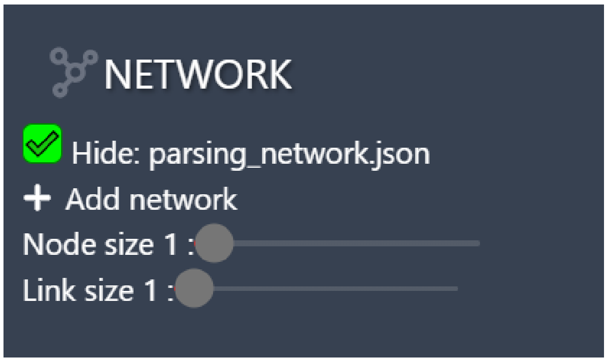
the network and configures the initial view. This is very useful as the user will not have to manually zoom in on the base map to reach the study area. When the user decides to upload a file from the "add network" button, the computer directory will be opened to proceed with the selection of the network file. Once selected, the file name will be displayed next to the "show" option to identify which file has that layer. The green check will indicate whether the layer is being displayed in the display or not (see Figure 4.6). Next, in Figure 4.7, we have the node size slider, which allows us to adjust the size in meters of the node radius, ranging between 1 and 20 times its original size. Additionally, we have a slider for the size of the links, which varies from 1 to 10 meters.

These sliding actions give us a powerful ability to manipulate and customize our network according to our needs, which significantly simplifies the identification and visualization of the most relevant elements within the graph. A brief demonstration of the interaction with the sliders is exhibited in Figure 4.7, where a significant transformation can be seen in the visual representation of the nodes and links, from expanding nodes to highlighting their importance to adjusting the thickness of links to emphasize key connections where events travel.

In addition, tooltips provide us with an invaluable tool to provide detailed informa-

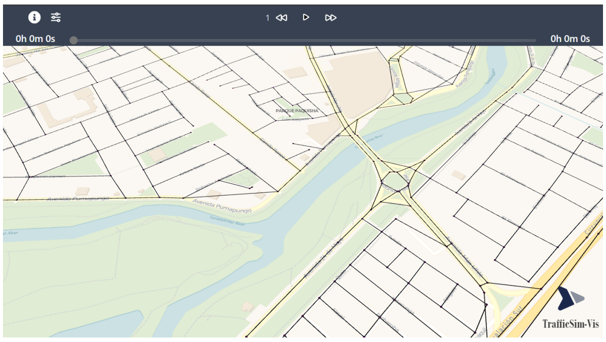


(a) Network: no file upload



(b) Network: with file uploaded

Figure 4.6: Network adjustment tools

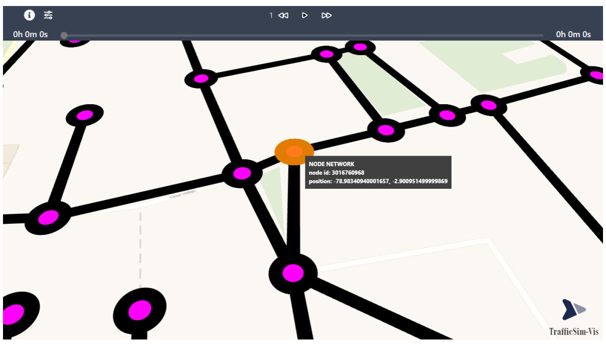


(a) Side nodes and Side links with slider of 1 meter

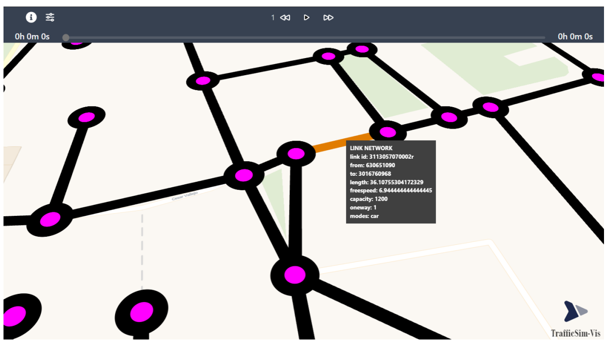
(b) Side nodes and Side links with slider of 5 meters

Figure 4.7: Interaction of adjustment tools

tion to the user. As illustrated in Figure 4.8, when you point to a node or link in the visualization, the selected object will be highlighted to stand out from the other elements. Simultaneously, a tooltip, a small pop-up window, is displayed, revealing crucial information about the specific attributes of that node or link.



(a) Node information



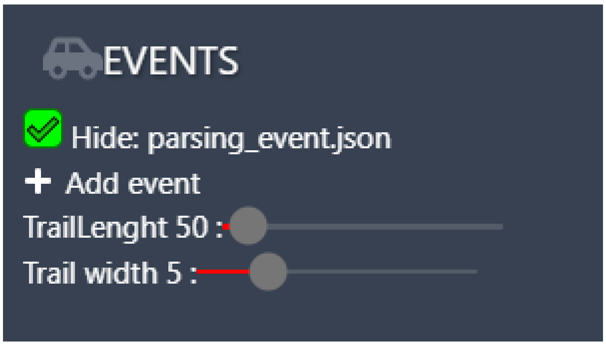
(b) Link information

Figure 4.8: Interactive tooltips information

Events

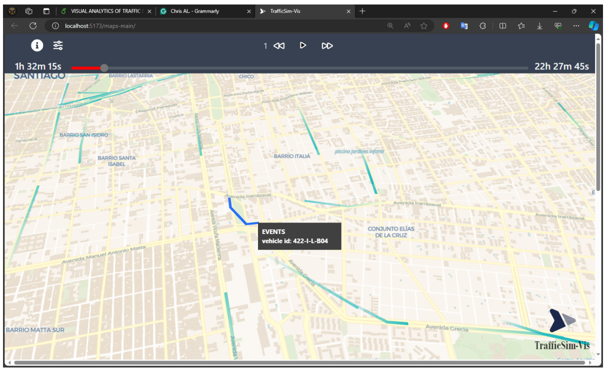
Similar to network, events allow the generated file to be loaded during processing. File loading and positioning are the same as described above (see Figure 4.9). The difference lies in the sliders implemented. We have the trailength slider, which determines the time it takes for a trail to disappear completely. This measurement is expressed in seconds because the TripsLayer layer has already processed the data in this unit of time. In addition, its range goes from 1 to 2000 seconds for fading and is set by default at 50 seconds. On the other hand, the trailwidth slider adjusts the width of the agent's route, which varies from 1 to 20 meters, and which by default is 5 meters (see Figure 4.10). Managing to highlight events over other layers allows the user to interact directly with moving agents and visualize their trajectories effectively.

(a) Events: no file upload

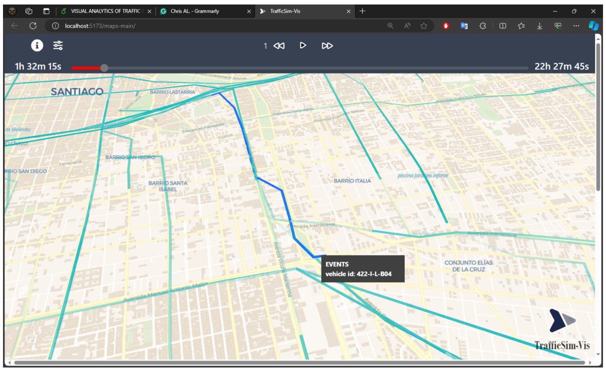


(b) Events: with file uploaded

Figure 4.9: Events adjustment tools



(a) Trailenght and trailwidth with slider in 50 seconds and 5 meters respectively



(b) Trailenght and trailwidth with slider in 113 seconds and 7 meters respectively

Figure 4.10: Events interaction adjustment tools

Custom Layers

This configuration covers the three custom layers of the application, which allow additional representations to be configured, enriching the user experience by offering the possibility of displaying multiple elements. Figure 4.11 show elements that can be presented are points that represent different objects of the real life in transportation, such as power stations, traffic lights, or other objects. It can even support another network on the map. This functionality improves the use of the application by providing greater flexibility and customization in the visualization of simulations.

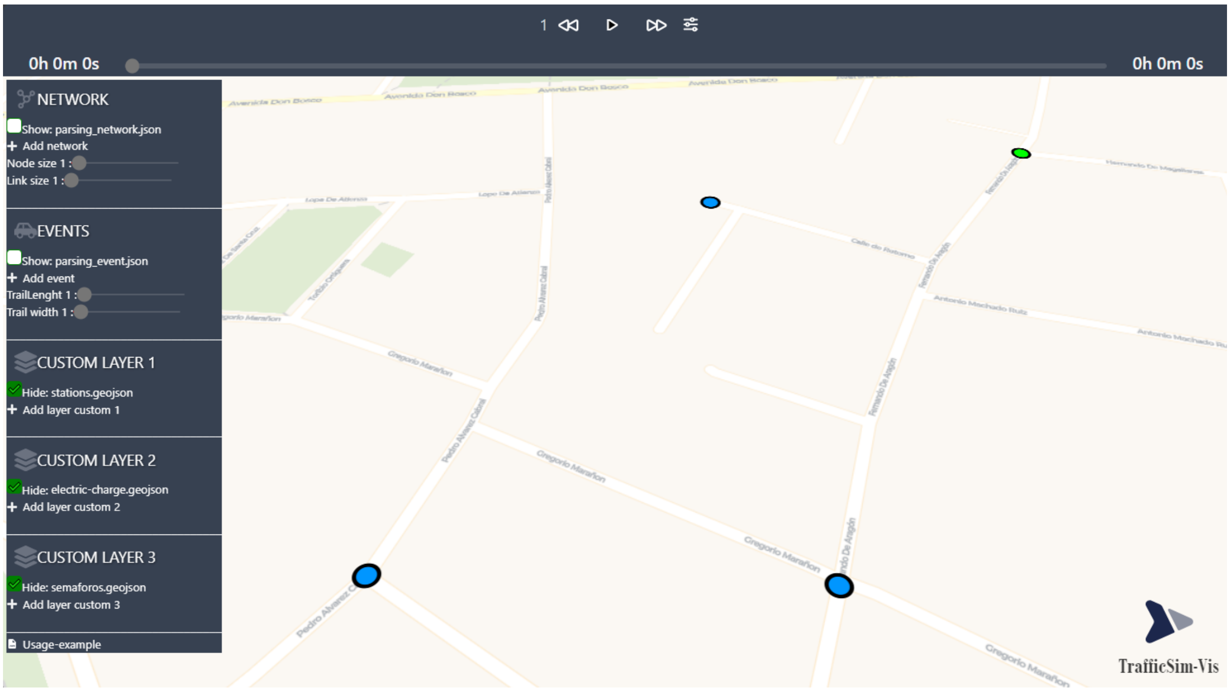
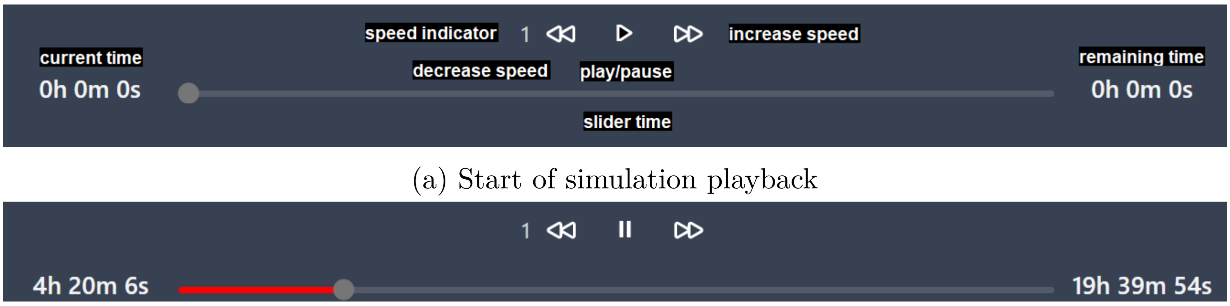


Figure 4.11: Sidebar CustomLayer

4.2.3 Playback Control

As shown in (c) in Figure 4.4, this group of buttons will allow control of the visualization. In other words, it refers to the ability to manage and control the progression of time during the simulation. Figure 4.12a shows the controls in greater detail: a central play/pause button responsible for running and pausing. In addition, two side buttons are responsible for advancing and decreasing the speed of the agents on the visualization and their respective speed indicator on the side. The user can even move in time with the time slider, which allows the evolution of transport conditions to be observed. The duration of the simulations is one day in a 24-hour format; that is, it begins at 00:00 on the left side and ends at 23:59 on the right side.



(b) Simulation running

Figure 4.12: Playback control

Chapter 5

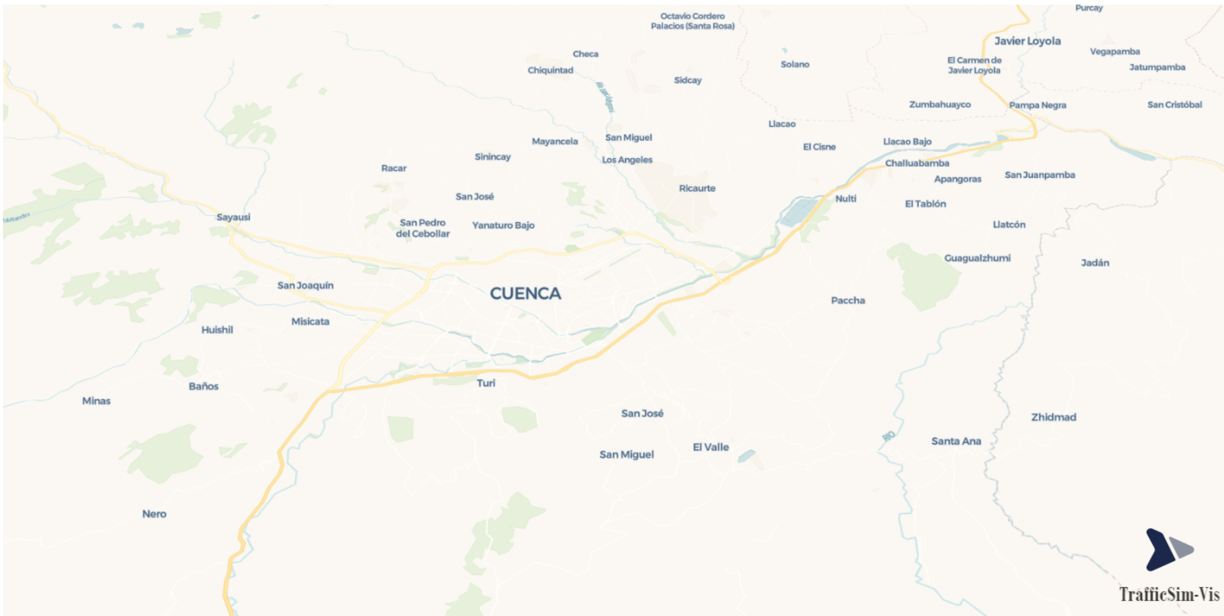
Results and Discussion

This chapter presents the results and discussion of two simulation scenarios using the TrafficSim-Vis implementation. The simulations correspond to Cuenca (Ecuador) and Santiago (Chile). The first one because the data was available and was the beginning of this thesis, while the second one was taken from MATSim repositories to test the visualization capability with larger scenarios. Each visualization was generated after completing the preprocessing process and then loaded into the application for analysis and exploration. In addition, relevant information is addressed by comparing previous works, performance, and various aspects related to visualization.

5.1 Cuenca

The capital of the province of Azuay is located in south-central Ecuador. Over the years, it has achieved the status of the third most influential city in the country with an area of 70.58 km^2 and a population of 361,524 inhabitants [55], both for its growing population and its significant economic impact, founded on commerce, services, and industry [56]. The accelerated growth of the city has generated an unsustainable situation in terms of vehicular and pedestrian traffic [57]. Therefore, several studies are focusing on proposing solutions to the various problems faced by the city, such as the construction of traffic noise predictions [58], polluting emissions [59], traffic signal optimization [60], among others. For this reason, we already have data related to this simulation, which was the basis for starting the visualization tool. Then, after performing the respective processing of the MATSim output files and loading the generated files, the following is obtained.

Figure 5.1 provides a comparison between Cuenca city area with and without network, which is composed of 13,739 nodes and 29,963 links. Thus, we can see how the graph fits the entire surface of the area presented by the base map. It is important to remember that both layers are overlaid to move simultaneously. Offering a clear perspective of the relationship between the graphic and the underlying geography, highlighting the extent and density of the network about the surrounding area.



(a) Cuenca without network

(b) Cuenca with network

Figure 5.1: Cuenca network

In addition, it is a useful resource for understanding the spatial distribution of nodes and links within the broader geographic context.

From a closer plane, in Figure 5.2, we direct our attention towards the center of the city of Cuenca, near the Tomebamba River, to observe how the nodes and links intertwine, adopting positions very similar to those of the base map. However, in Figure 5.3 when we head towards the edges of Cuenca, specifically to the south of the Panamericana Highway, we can notice how the nodes are dispersed, evidencing the limit of the network, which is limited to this particular city. This observation highlights how the network adapts to urban geography, clearly showing the density and distribution of nodes and links in urban areas versus more peripheral regions. Furthermore, in Figure 5.4, the information that TrafficSim-Vis provides us with the tooltips is beneficial because we will be able to notice relevant details seen in the Section 4.1.4 but already from the map. Allowing you to recognize places, positions and their characteristics faster.

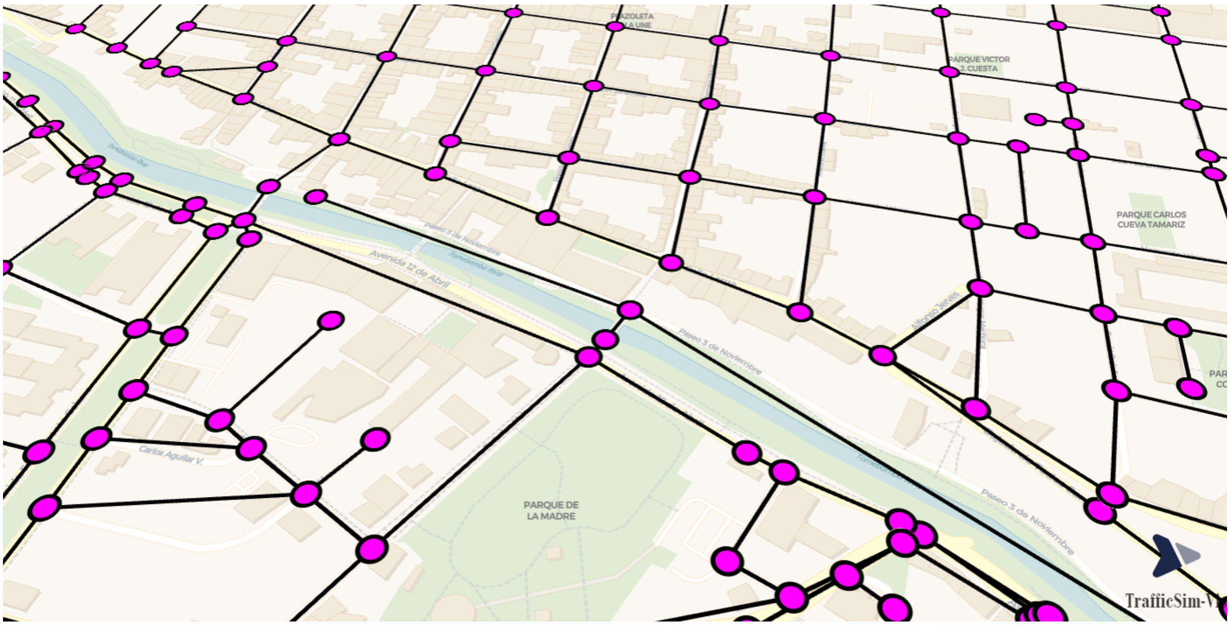


Figure 5.2: Visualization of the network in the center of the city of Cuenca

On the other hand, Figure 5.5 shows the visual representation of a considerable number of events, precisely 15,795 events generated by MATSim. It is essential to remember that each of these events reflects an action taken by a vehicle at a specific time and location during the simulation. With 3,303 vehicles distributed over a full day of simulation, this visualization captures the dynamic complexity of vehicle flow and activity in the simulation.

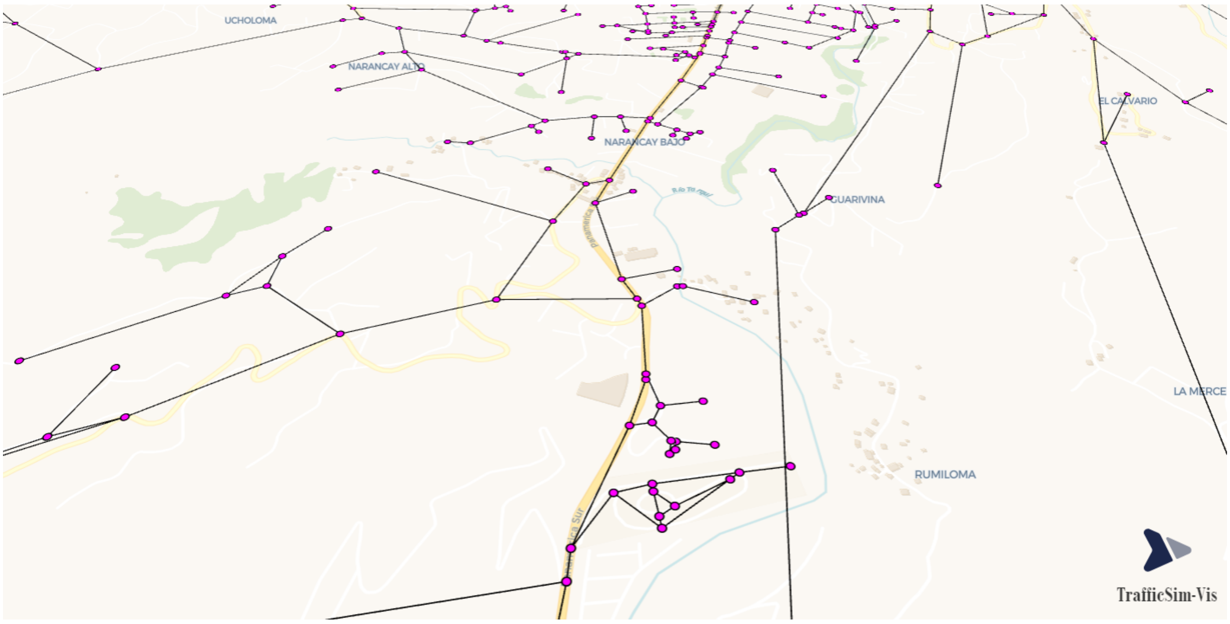
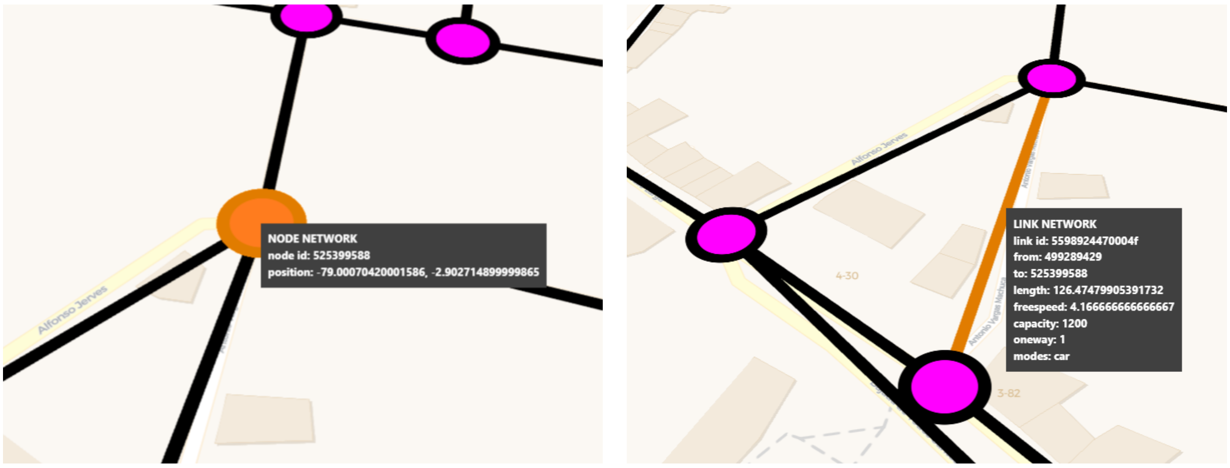


Figure 5.3: Visualization of the network on the edges of the city Cuenca



(a) (b)

Figure 5.4: Information tooltip: (a) Node (b) Link

Furthermore, we can reveal vehicle circulation patterns by adjusting the sliders in Figure 5.6. By identifying the most traveled routes generated by simulation, we obtain valuable information about the busiest roads and predominant traffic flows in the city. At the same time, by selecting a specific vehicle, we have the ability to closely examine its identifier and precisely follow the route it has taken throughout time, providing significant insights into its behavior and trajectory in the urban environment. Additionally, the inclusion of the time slider further expands our analysis capabilities. We can set the time at any desired time, allowing us to study vehicle patterns or specific events at different times of the day.

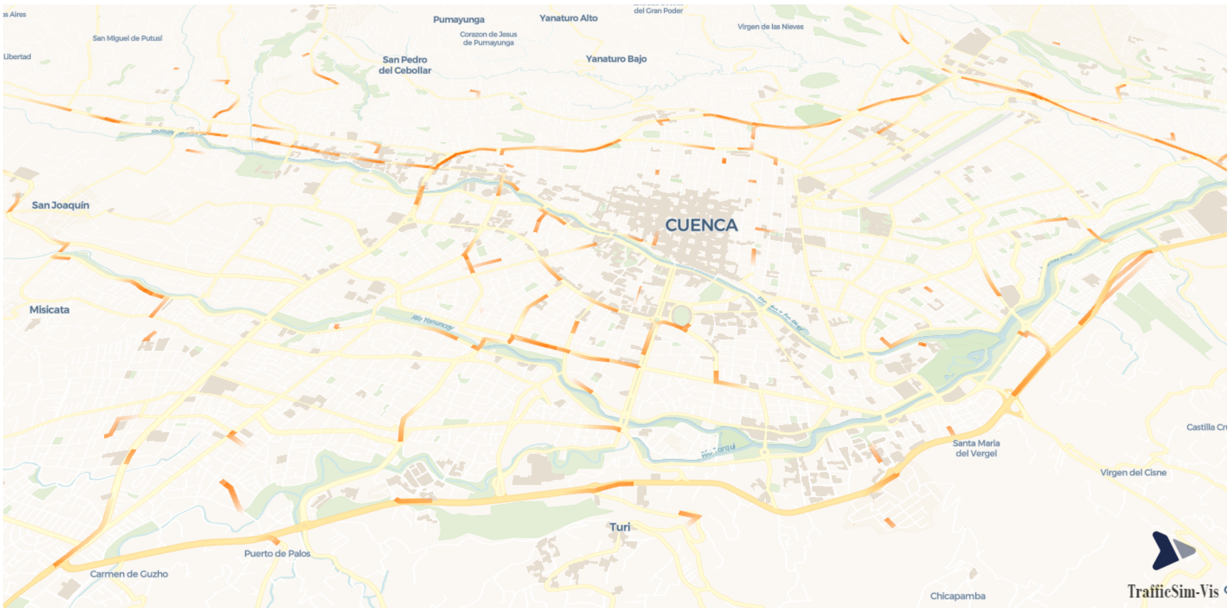


Figure 5.5: Cuenca events

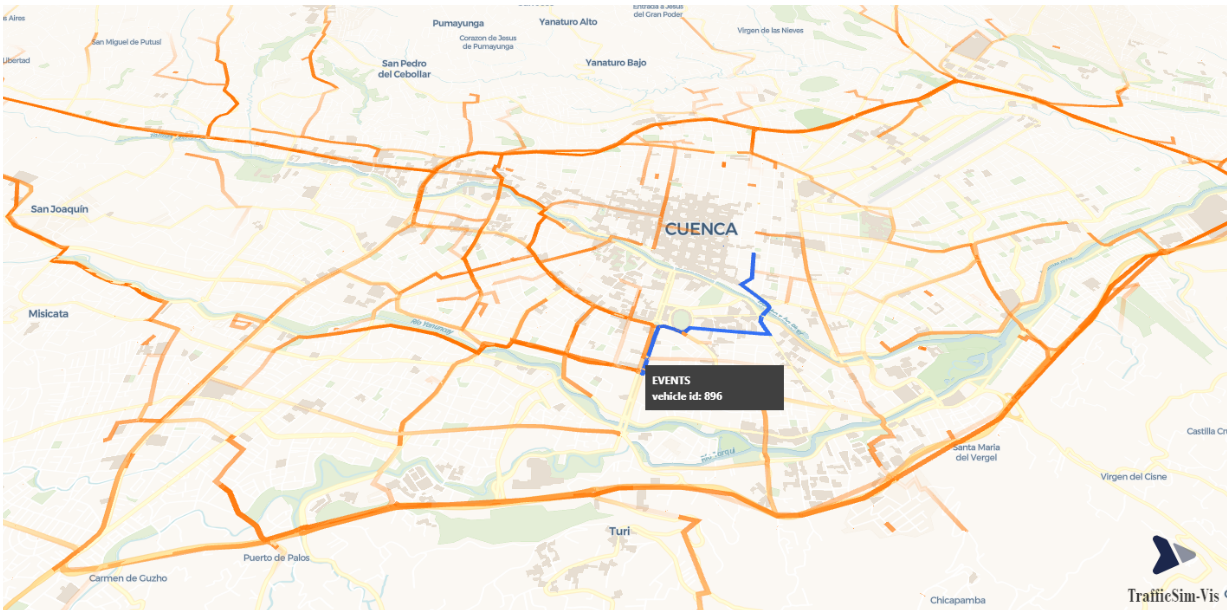


Figure 5.6: Discovering most used routes

In Figure 5.7, a new custom layer called 'Charging Stations' has been configured, which consists of 20 stations identified along with their respective geographical coordinates. This implementation allows identifying possible locations and strategic arrangements for stations in Cuenca, offering a clear and detailed visualization. Additionally, the ability to load up to two additional custom layers further expands the versatility of TrafficSim-Vis.

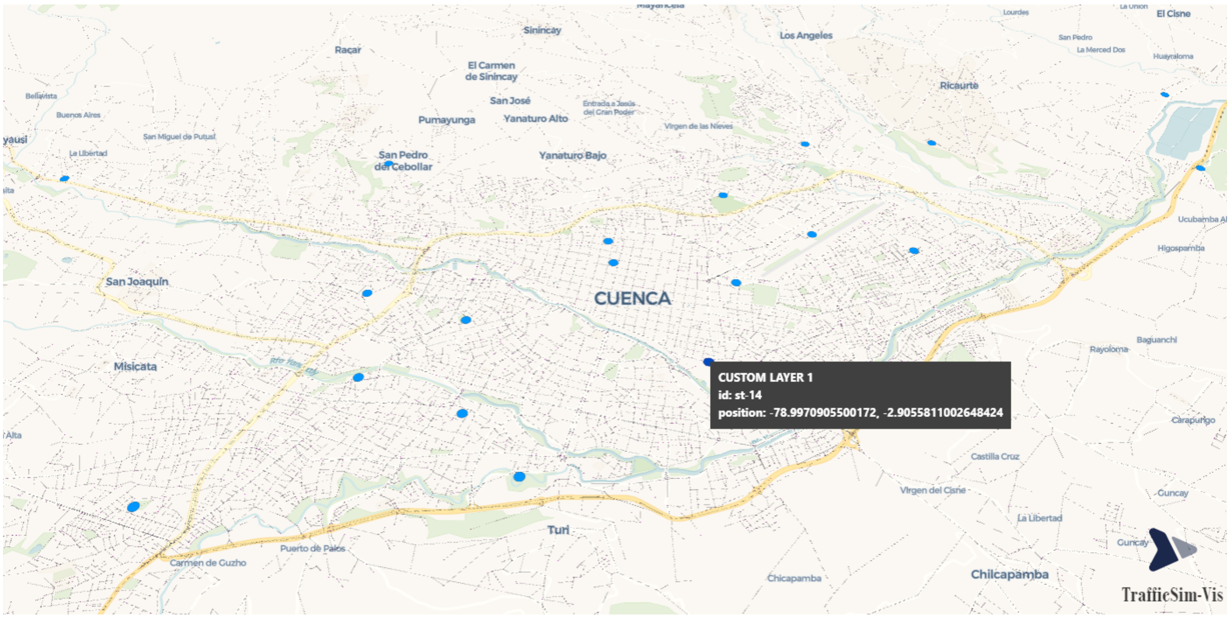


Figure 5.7: Add custom layers in Cuenca

5.2 Santiago

Also known as Santiago de Chile, this city is the capital and main center of Chile. Strategically located in the geographical heart of the country, it covers an area of 641 km^2. According to census carried out by the National Institute of Statistics[1] in 2017, Santiago is home to a population of more than six millions inhabitants, with a notable population density of 8,497 people per square kilometer. As a result, a poorly planned metropolitan city developed, congested, and with severe vehicular mobility problems [61]. So, Kickhofer *et al.* [62] produces a scenario to solve different research problems related to transport policy interventions, accessibility measures, location problems, or business ideas based on mobility. Also, he created several executable versions[2] of the scenario are available in which them has been developed from three primary sources of open data:

- OpenStreetMap road network information
- Public transportation data provided in the Google Transit Feed Specification (GTFS)
- Travel diaries collected in the 2012 Santiago Origin-Destination Survey.

[1] `https://www.ine.gob.cl/estadisticas/sociales/censos-de-poblacion-y-vivienda/censo-de-poblacion-y-vivienda`

[2] `https://svn.vsp.tu-berlin.de/repos/public-svn/matsim/scenarios/countries/cl/santiago/`

The exclusive use of open data sources and the continuous process of gradual improvement make MATSim Santiago one of the most complete and accessible scenarios for researchers around the world [62]. For this reason, it is subject to being tested in TrafficSim-VIs.

The visualization obtained from the network that represents the city of Santiago is shown in Figure 5.8. It consists of a graph of 23,330 nodes and 37,834 links covering the entire city area. Additionally, in Figure 5.9, we present a close-up view of the center of Santiago de Chile to visualize the links and nodes. Which, we can realize that this simulation contains different types of transportation, such as buses, metro trains, colectivos (shared taxis), and private vehicles [62]. These are presented in Figure 5.10, which can be accessed by clicking on the corresponding link to obtain the necessary network information. In addition, the extensive coverage of the network throughout the city of Santiago stands out, extending even to Malepillas, a small town from where simulated vehicles also originate, covering a stretch of up to 70 km to the center of Santiago.

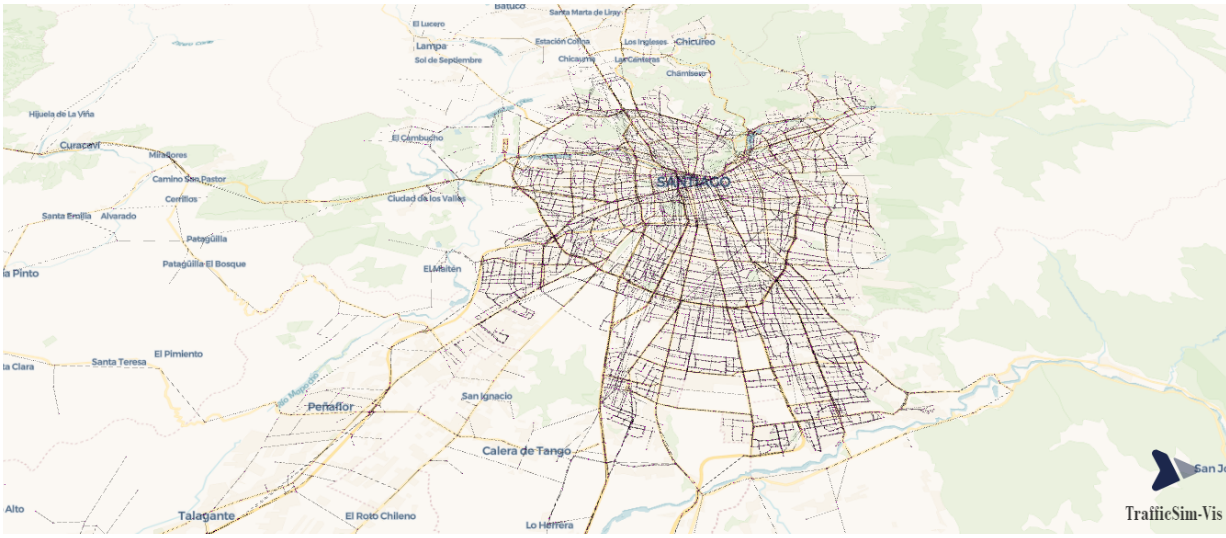
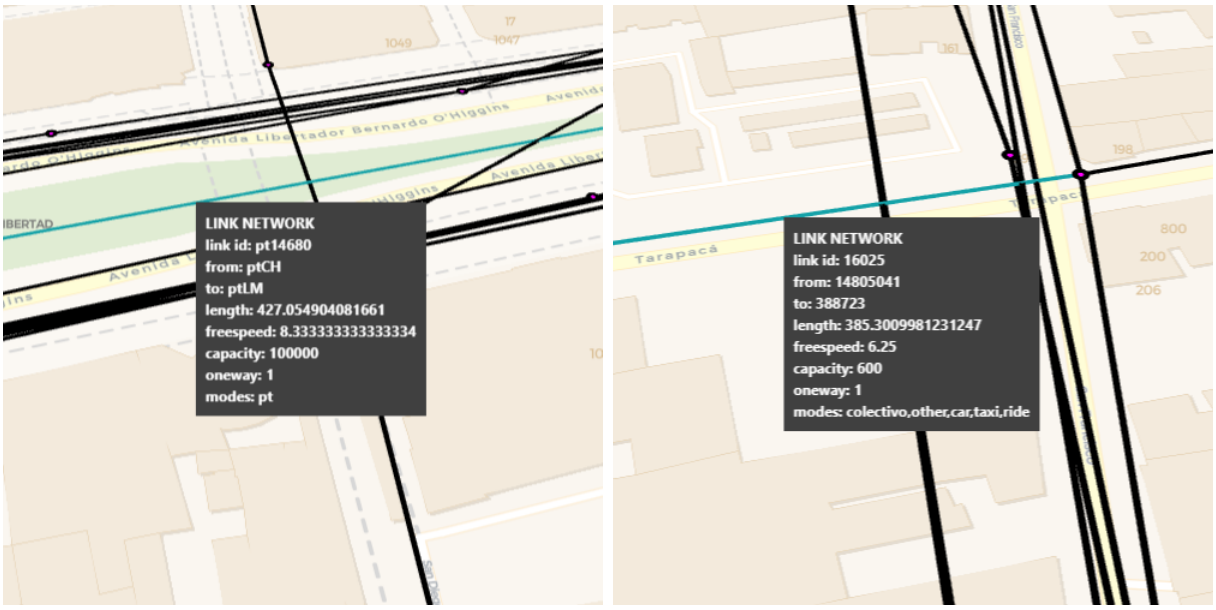


Figure 5.8: Santiago network

On the events side, it consists of 197,309 vehicles distributed in 14,000,000 events spread over one simulation day. However, because this is a very complex and extensive simulation, it has not been possible to load the entire simulation from a single file. Therefore, the event file had to be divided into parts to be analyzed individually. In Figure 5.11, we can see the beginning of the agents' movements throughout the vehicle network in the city.

Figure 5.12 shows different areas of the city where the flow of vehicles is observed at dif-

Figure 5.9: Visualization of the network in the center of the city of Santiago



(a) Public transportation (b) Other vehicles

Figure 5.10: Information on network links

ferent times of the day, providing a detailed view of the dynamics of simulated urban traffic. Furthermore, the dashed lines represent congested vehicular traffic, while the straight lines indicate agents that circulate without obstacles. When you place the cursor over an agent, the information associated with them is displayed, as shown in Figure 5.13. Each vehicle or agent has a unique identifier that allows them to be distinguished, which is essential to understand the complexity of the transportation system in Santiago. This approach makes it possible to identify whether the agent corresponds to one of the different types of

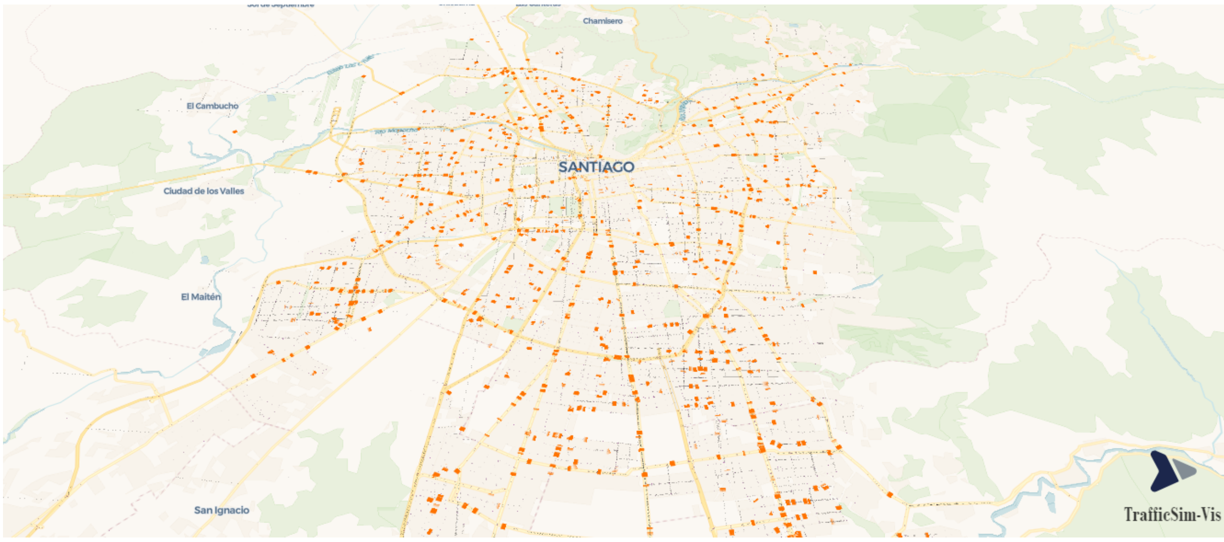
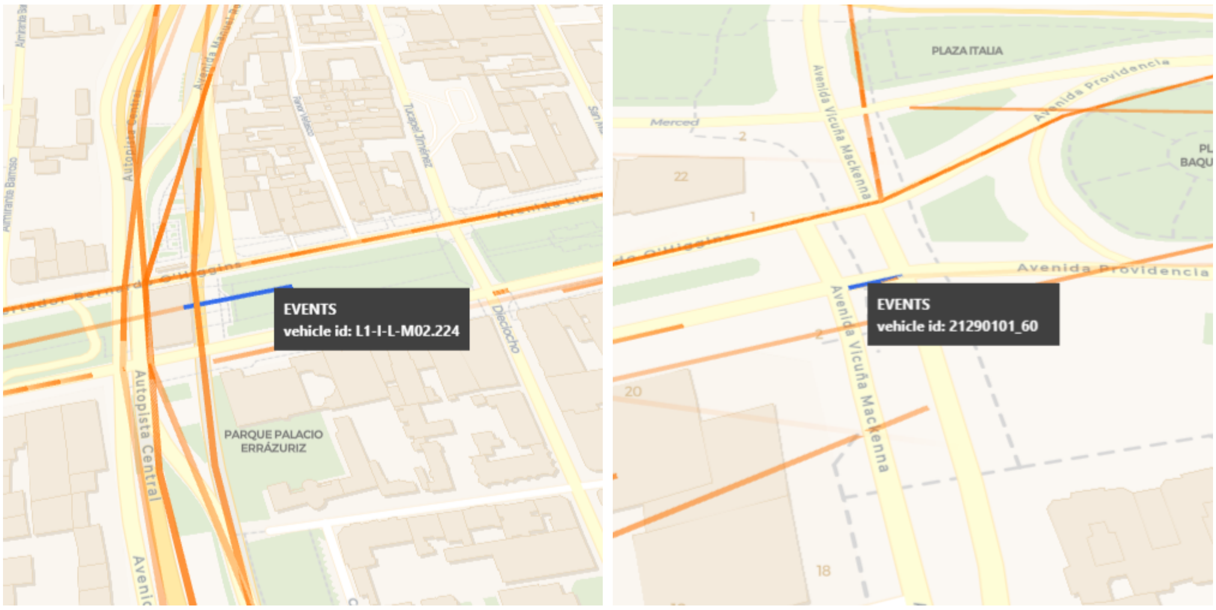


Figure 5.11: Santiago events

transport configured for the simulation, providing valuable details about its behavior and trajectory in real time. Which makes TrafficSim-Vis ideal for visualizing and improving the efficiency of road infrastructure, as well as for designing more effective traffic management strategies in the city, while using a tool available and free to use.

Figure 5.12: Traffic flow in different areas of Santiago



(a) Public Transportation agent (b) Other vehicles agent

Figure 5.13: Information on events

5.3 Performance

As a final step in presenting the results, we carried out an evaluation of the performance of TrafficSim-Vis using the data sets of the cities of Cuenca and Santiago. This process involves a detailed analysis of the application on two different personal computers (PCs), as well as monitoring the use of the processor (CPU), memory (RAM), and graphics processing (GPU), which are detailed in Table 5.1. On the other hand, in Figure 5.14, we can see the information related to the performance of each computer, starting from an initial state of the computer, and see how its use changes with the execution of the different simulation data.

Table 5.1: Features of the PCs on which TrafficSim-Vis was tested.

	CPU	**RAM**	**GPU**
PC 1	Core i5 7200U	8,0 GB	Intel HD Graphics 620
PC 2	Core i5 12450H	16,0 GB	Nvidia GeForce RTX 3050

In this context, PC 1 showed performance conditioned by its characteristics, showing fast loading but medium execution fluidity. Cuenca's visualization remained at a normal level, with acceptable fluency and satisfactory response in memory management. However, despite achieving visualization, a lack of fluidity was perceived at certain moments. In contrast, when running the data corresponding to Santiago, it experienced a greater demand for resources, requiring more significant use of the CPU and RAM and the total capacity of the GPU to achieve optimal response in controls and movements between layers. At times, response to events became slow and tedious, affecting the user experience.

In contrast, the PC 2 exhibited noticeably superior performance, characterized by dynamic and fluid loading and agile response. The visualization of Cuenca was particularly fast, with outstanding fluidity and optimal memory management. On the other hand, when viewing the Santiago data, it demonstrated greater responsiveness, effectively harnessing the potential of the GPU to deliver a more fluid and responsive user experience. Controls and scrolling on layers ran smoothly, providing a more satisfying experience than PC 1 in Santiago's visualization.

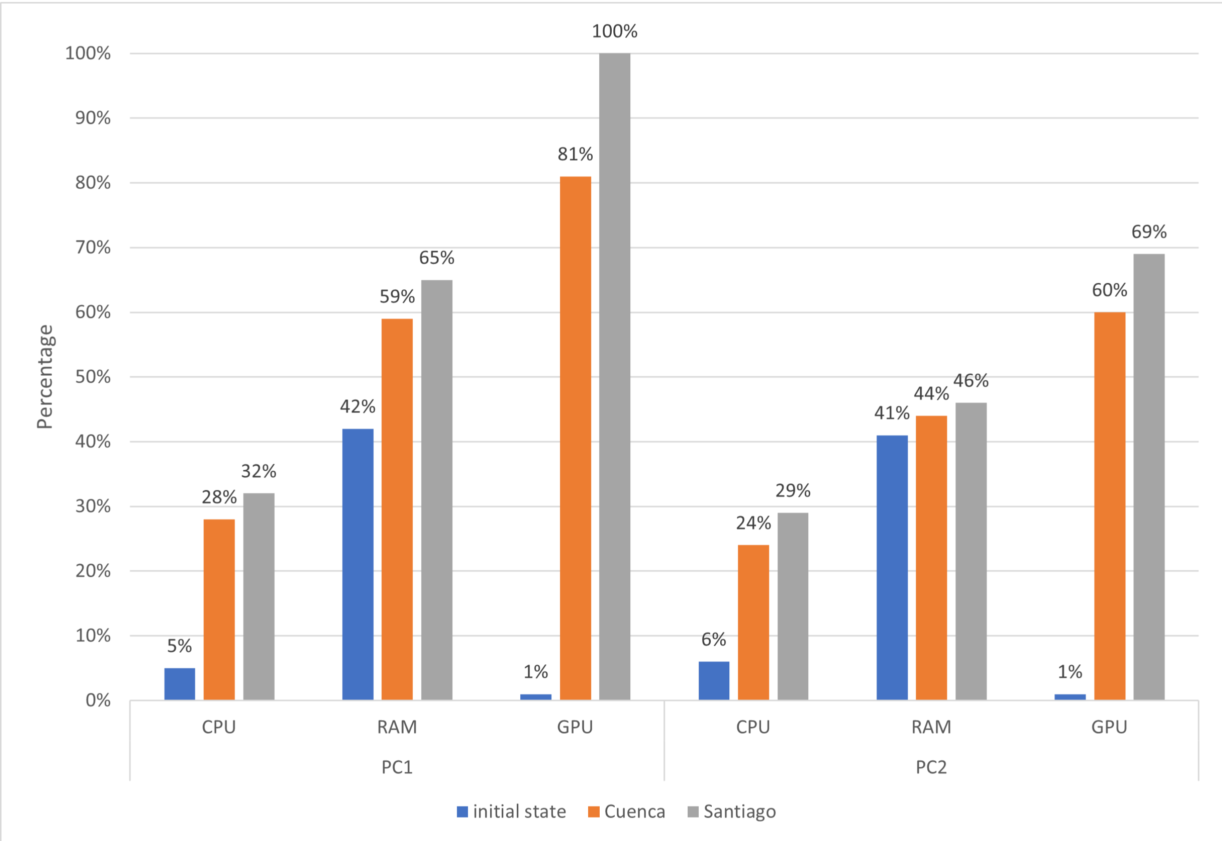


Figure 5.14: Performance on PCs

5.4 Discussion

This section has highlighted the behavior of the application, including its visual representations and interactions, across various data sets. As previously introduced, TrafficSim-Vis focuses on the analysis and visualization of simulation data, objectives that have been achieved at this stage. Given the relevance of MATSim for planning and modeling, we have chosen to work with its results. However, according to Charlton, the amount of data generated by MATSim can be overwhelming, negatively impacting the performance of any application, even for modern hardware, making it difficult to obtain visually satisfying results with detailed data. Charlton also mentions that vehicle flow performance is highly dependent on the performance of the computer the application uses to play the simulations in real-time, with the goal that the browser only has to render the data [12].

As a result, this work follows a similar approach by preprocessing the data generated by MATSim, allowing it only to need to be rendered by the browser. Therefore, the search for a framework capable of handling large volumes of data and efficient in visualization became a priority. After reviewing the available literature, Deck GL was selected as a framework with great potential for our objectives that fulfills its purpose. The first step consisted of developing preprocessing, for which it was necessary to extract all the information necessary for the visualization of the objects. Articles such as Chen [23] and Andrienko [63] provided helpful information for this task. In this sense, network output data and events were extracted. However, these disaggregated files required an integration process with Deck GL. One of the most important processes was the selection of coordinates, as these provide information about the simulation location and allow their transformation to standard coordinates (EPSG:4326) for use with Deck GL. Once this process was completed, the parsed files were downloaded. Following the approach used by Seto [64], efforts were made to reduce the data size as much as possible so that any browser could support it. This was achieved by generating comma-separated files in plain text and JSON formats, thus optimizing file weight and memory usage.

The visualization of the Cuenca and Santiago scenarios resulted in incredibly interactive performances. In the case of the capital of Azuay, the representation of its network was entirely satisfactory, considering that it is a developing city. Solid connections are observed between nodes and links, especially in the central areas of the city (see Figure 5.2). Likewise, the visualization was impressive for Santiago, a metropolitan city with abundant information about its best-known areas. However, in both networks, it is noted that specific nodes and links do not fit properly to the TrafficSim-Vis base map, resulting in straight paths without curves (see Figures 5.3 and 5.4).

This is due to the network generation files used for simulation in MATSim. If sources with data from little-known or outdated cities are used, these results will be obtained. On the other hand, the playback of events was carried out in a fluid and interactive manner. Interactions with buttons, sliders, and modals were executed with great precision. Only in cases of having limited computational resources was a slight impact on the performance of user interactions observed. Tools developed by Charlton [10, 12, 14], Xu [11, 50], and Petrovska [26] also mention that handling large amounts of disaggregated data generates considerable resource usage. Therefore, web applications are still limited by the resources of the computer that runs them.

To conclude this discussion, TrafficSim-Vis has proven to be an exceptional application by offering the ability to render more than 20 thousand vehicles in a simulation, significantly exceeding the 500 vehicles shown by Via in its free version. Its availability online at all times adds great value as an open source tool. In addition, it is only required to bring the simulations format to the format expected by the application to work, so it is not subject only to simulating MATSIm. Through its simple interface, users have the ability to select multiple layers and adjust them according to their needs, allowing them to generate personalized results. The event interaction feature makes it easy to identify the routes followed and provides detailed information about the links used in the trip. Additionally, the ability to adjust playback speed and the ability to add custom layers to represent objects that interact with each other expands the app's capabilities and offers a versatile user experience.

Chapter 6

Conclusions

This chapter presents conclusions about this work, the results, performance, recommendations, and possible future research related to the visualization of traffic simulation data.

6.1 Conclusions

The visualization of traffic simulations is essential for the development and planning of cities. For this reason, developing a single-page web application to visualize a data from MATSim was the purpose of this work. We present a web application called TrafficSim-Vis that allows you to analyze, visualize, and interact with large-scale traffic simulations results. The implementation with a visualization web framework such as Deck GL was essential to manage and represent large volumes of data. The design and implementation of our application was based on different related works from Chapter 3, identifying a unique application approach because we needed it to be fast, smooth to run, and lightweight for computers. This is because the results of the traffic simulations addressed by our application contain large geospatial data sets that need to be processed.

The application successfully achieved the main established objectives:

- A code capable of processing the MATSim simulation outputs and converting them to a file format (JSON) capable of achieving adequate integration with Deck GL web framework was successfully developed. This code is available in the repository.
- It was possible to load the processed files in a simple and independent way. Based on

the state of the art, the files were intended to be easy to load, while also providing quick, modifiable, and real-time visualizations. In addition, to achieve control of the timing of the events together with showing valuable information about the simulation,

- A dynamic and interactive single-page web application capable of generating visualizations was developed, implemented with the latest in HTML, CSS, and JavaScript to achieve innovative interfaces compatible with browsers.

On the other hand, the result obtained from the Cuenca scenario demonstrated a fluid application, with a fairly good initial amount of data that needed to be tested with a larger scenario. For this reason, the Santiago de Chile scenario was fundamental since it tested the visualization capacity of the application with a greater amount of data and demand for TrafficSim-Vis. However, the measured performance shows that the application depends to some extent on the components of the computer running it, because file loading is computer dependent. The application interface (based on related works) is easy to use, accessible and designed for all users. Furthermore, being developed with a responsive design and flexible architecture, the application can be compatible with any browser. On the other hand, being open source, our application can generate low-cost research opportunities by expanding vehicle visualization capabilities, overcoming the limitations of the Via software in its free license and the costs involved in acquiring the complete modules. Other visualization models and tools, such as those seen in related works, are licensed, unsupported, or deprecated.

Finally, our application demonstrated the great potential of web frameworks by combining them with large data sets to generate effective visualizations. The scenarios used guarantee a functional and useful application for researchers and urban planners to use a free access tool. Additionally, continue research that combines the knowledge provided through traffic simulation visualization with the power of web computing, allowing dissemination, comparison, and extended analysis of simulation results through open-source applications.

Appendix .1 shows a guide with the requirements, preprocessing (parsing), and steps to carry out for installing TrafficSim-Vis. All the code is available on the GitHub site at: `https://github.com/ChrisALoor/TrafficSim-Vis`

6.2 Recommendations

In this section, we offer several recommendations based on the challenges and limitations encountered during the course of this work.

- Ensure utilization of updated and consistent MATSim configuration data tailored to the intended simulations.
- Implement an efficient coordinate transformation method to enable swift conversions and minimize memory usage.
- Develop a batch processing mechanism for event visualization, particularly beneficial for resource-intensive simulations with extensive event data.
- Opt for comma-separated values for JSON file generation to minimize the file size.
- Optimize the viewing experience by utilizing computers with dedicated graphics cards to enhance graphical rendering and ensure smooth interaction.
- Employ Node Package Module for streamlined installation of dependencies, simplifying usage, and minimizing setup configurations.

6.3 Future works

In this section, we describe possible implementation approaches that could be taken up for the creation of future work:

- Incorporating more MATSim outputs like "plans.xml" or "vehicles.xml" could facilitate alternative types of analysis, potentially enhancing the quality of results.
- By leveraging parallel processing techniques, the execution time of file processing could be significantly reduced, enhancing overall efficiency.
- Creating an application that seamlessly integrates the processing of MATSim files and subsequent visualization generation could streamline the workflow and enhance user experience.

- Introducing more interactive tools such as dashboard, buttons, filters, sliders, and modals could empower users with greater flexibility for customizing visualizations to meet specific needs.

Bibliography

[1] F. van Wageningen-Kessels, H. van Lint, K. Vuik, and S. Hoogendoorn, "Genealogy of traffic flow models," *EURO Journal on Transportation and Logistics*, vol. 4, no. 4, pp. 445–473, 2015.

[2] Deck.gl. [Online]. Available: https://deck.gl/docs

[3] N. Carter, M. S. Ahmed, J. Hoyos, M. Dale, J. Blevins, N. Hodge, N. Chowdhury, and M. A. Hoque, "Developing mobility and traffic visualization applications for connected vehicles," *arXiv preprint arXiv:1811.11012*, 2018.

[4] M. Pi, H. Yeon, H. Son, and Y. Jang, "Visual cause analytics for traffic congestion," *IEEE transactions on visualization and computer graphics*, vol. 27, no. 3, pp. 2186–2201, 2019.

[5] C. Lee, Y. Kim, S. Jin, D. Kim, R. Maciejewski, D. Ebert, and S. Ko, "A visual analytics system for exploring, monitoring, and forecasting road traffic congestion," *IEEE transactions on visualization and computer graphics*, vol. 26, no. 11, pp. 3133–3146, 2019.

[6] R. Scheepens, C. Hurter, H. Van De Wetering, and J. J. Van Wijk, "Visualization, selection, and analysis of traffic flows," *IEEE transactions on visualization and computer graphics*, vol. 22, no. 1, pp. 379–388, 2015.

[7] A. Erath and P. J. Fourie, "Interactive analysis and decision support with matsim," in *The Multi-Agent Transport Simulation MATSim.* Ubiquity Press, 2016, pp. 253–258.

[8] C. Piris *et al.*, "Estudio y desarrollo de una aplicación web para la generación y tratamiento de datos del simulador matsim," 2013.

[9] J. Jung, T. Oh, I. Kim, and S. Park, "Open-sourced real-time visualization platform for traffic simulation," *Procedia Computer Science*, vol. 220, pp. 243–250, 2023.

[10] W. Charlton, G. Leich, and I. Kaddoura, "Open-source web-based visualizer for dynamic-response shared taxi simulations," *Procedia Computer Science*, vol. 184, pp. 728–733, 2021.

[11] H. Xu, C. Wang, A. Berres, T. LaClair, and J. Sanyal, "Interactive web application for traffic simulation data management and visualization," *Transportation research record*, vol. 2676, no. 1, pp. 274–292, 2022.

[12] B. Charlton and J. Laudan, "Web-based data visualization platform for matsim," *Transportation Research Record*, vol. 2674, no. 10, pp. 124–133, 2020.

[13] D. Miranda and F. Arruda, "Method for simulating a matsim multi-agent activity-based transport model in developing countries," 07 2019.

[14] W. Charlton and B. Sana, "Simwrapper, an open source web-based platform for interactive visualization of microsimulation outputs and transport data," *Procedia Computer Science*, vol. 220, pp. 724–729, 2023.

[15] A. Bull, N. CEPAL *et al.*, *Traffic Congestion: The Problem and how to Deal with it.* ECLAC, 2003.

[16] T. Afrin and N. Yodo, "A survey of road traffic congestion measures towards a sustainable and resilient transportation system," *Sustainability*, vol. 12, no. 11, p. 4660, 2020.

[17] H. Karimi, B. Ghadirifaraz, S. N. Shetab Boushehri, S.-M. Hosseininasab, and N. Rafiei, "Reducing traffic congestion and increasing sustainability in special urban areas through one-way traffic reconfiguration," *Transportation*, pp. 1–24, 2021.

[18] C. Mauro and P. Isreael, "Large scale traffic flow simulation using a distributed system," 2021.

[19] N. T. Ratrout and S. M. Rahman, "A comparative analysis of currently used microscopic and macroscopic traffic simulation software," *The Arabian Journal for Science and Engineering*, vol. 34, no. 1B, pp. 121–133, 2009.

[20] K. W Axhausen, A. Horni, and K. Nagel, *The multi-agent transport simulation MATSim.* Ubiquity Press, 2016.

[21] K. Nagel and F. Marchal, "Computational methods for multi-agent simulations of travel behavior," *Proceedings of International Association for Travel Behavior Research (IATBR), Lucerne, Switzerland*, 2003.

[22] R. A. Waraich, D. Charypar, M. Balmer, and K. W. Axhausen, "Performance improvements for large-scale traffic simulation in matsim," *Computational approaches for urban environments*, pp. 211–233, 2015.

[23] W. Chen, F. Guo, and F.-Y. Wang, "A survey of traffic data visualization," *IEEE Transactions on intelligent transportation systems*, vol. 16, no. 6, pp. 2970–2984, 2015.

[24] P. M. Ejercito, K. G. E. Nebrija, R. P. Feria, and L. L. Lara-Figueroa, "Traffic simulation software review," in *2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA).* IEEE, 2017, pp. 1–4.

[25] M. Saidallah, A. El Fergougui, and A. E. Elalaoui, "A comparative study of urban road traffic simulators," in *MATEC Web of Conferences*, vol. 81. EDP Sciences, 2016, p. 05002.

[26] N. Petrovska and A. Stevanovic, "Traffic congestion analysis visualization tool," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems.* IEEE, 2015, pp. 1489–1494.

[27] Simunto Via. [Online]. Available: https://www.simunto.com/via/licenses/

[28] C. Almachi, R. Armas, and E. Cuenca, "Visual analytic of traffic simulation data: A review," in *Ibero-American Congress of Smart Cities.* Springer, 2023, pp. 48–60.

[29] I. T. Forum, *ITF Transport Outlook 2021*, 2021. [Online]. Available: https://www.oecd-ilibrary.org/content/publication/16826a30-en

[30] M. R. Ullah, K. S. Khattak, Z. H. Khan, M. A. Khan, N. Minallah, and A. N. Khan, “Vehicular traffic simulation software: A systematic comparative analysis,” *Pakistan Journal of Engineering and Technology*, vol. 4, no. 1, pp. 66–78, 2021.

[31] H.-S. Zhang, Y. Zhang, Z.-H. Li, and D.-C. Hu, “Spatial-temporal traffic data analysis based on global data management using mas,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 4, pp. 267–275, 2004.

[32] C. Bachechi, L. Po, and F. Rollo, “Big data analytics and visualization in traffic monitoring,” *Big Data Research*, vol. 27, p. 100292, 2022.

[33] L. Montero, M. P. Linares, O. Serch, and J. Casanovas-Garcia, “A visualization tool based on traffic simulation for the analysis and evaluation of smart city policies, innovative vehicles and mobility concepts,” in *2017 Winter Simulation Conference (WSC)*. IEEE, 2017, pp. 3196–3207.

[34] B. D. Greenshields, J. Bibbins, W. Channing, and H. Miller, “A study of traffic capacity,” in *Highway research board proceedings*, vol. 14, no. 1. Washington, DC, 1935, pp. 448–477.

[35] C.-j. Liu, Z. Liu, Y.-j. Chai, and T.-t. Liu, “Review of virtual traffic simulation and its applications,” *Journal of Advanced Transportation*, vol. 2020, 2020.

[36] S. Dorokhin, A. Artemov, D. Likhachev, A. Novikov, and E. Starkov, “Traffic simulation: an analytical review,” in *IOP Conference Series: Materials Science and Engineering*, vol. 918, no. 1. IOP Publishing, 2020, p. 012058.

[37] E. López-Neri, A. Ramírez-Treviño, and E. López-Mellado, “A modeling framework for urban traffic systems microscopic simulation,” *Simulation Modelling Practice and Theory*, vol. 18, no. 8, pp. 1145–1161, 2010.

[38] J. Barceló, E. Codina, J. Casas, J. L. Ferrer, and D. García, “Microscopic traffic simulation: A tool for the design, analysis and evaluation of intelligent transport systems,” *Journal of intelligent and robotic systems*, vol. 41, pp. 173–203, 2005.

[39] A. Horni, K. Nagel, and K. W. Axhausen, "Introducing matsim," in *The multi-agent transport simulation MATSim.* Ubiquity Press, 2016, pp. 3–7.

[40] B. H. McCormick, "Visualization in scientific computing," *Acm Sigbio Newsletter*, vol. 10, no. 1, pp. 15–21, 1988.

[41] T. DeFanti, M. Brown, and B. McCormick, "Visualization: expanding scientific and engineering research opportunities," *Computer*, vol. 22, no. 8, pp. 12–16, 1989.

[42] J. Kohlhammer, D. Keim, M. Pohl, G. Santucci, and G. Andrienko, "Solving problems with visual analytics," *Procedia Computer Science*, vol. 7, pp. 117–120, 2011.

[43] I. Krak, O. Barmak, and E. Manziuk, "Using visual analytics to develop human and machine-centric models: A review of approaches and proposed information technology," *Computational Intelligence*, vol. 38, no. 3, pp. 921–946, 2022.

[44] K. Wang, M. Liang, Y. Li, J. Liu, and R. W. Liu, "Maritime traffic data visualization: A brief review," in *2019 IEEE 4th International Conference on Big Data Analytics (ICBDA).* IEEE, 2019, pp. 67–72.

[45] H. M. Shakeel, S. Iram, H. Al-Aqrabi, T. Alsboui, and R. Hill, "A comprehensive state-of-the-art survey on data visualization tools: research developments, challenges and future domain specific visualization framework," *IEEE Access*, 2022.

[46] Y. Wang, "Deck. gl: Large-scale web-based visual analytics made easy," *arXiv preprint arXiv:1910.08865*, 2019.

[47] H. Butler, M. Daly, A. Doyle, S. Gillies, S. Hagen, and T. Schaub, "The geojson format," Tech. Rep., 2016.

[48] MapLibre GL JS. [Online]. Available: https://maplibre.org/maplibre-gl-js/docs/

[49] M. Haklay and P. Weber, "Openstreetmap: User-generated street maps," *IEEE Pervasive computing*, vol. 7, no. 4, pp. 12–18, 2008.

[50] H. Xu, A. Berres, C. R. Wang, T. J. LaClair, and J. Sanyal, "Visualizing vehicle acceleration and braking energy at intersections along a major traffic corridor," in

Proceedings of the Twelfth ACM International Conference on Future Energy Systems, 2021, pp. 401–405.

[51] D. Strippgen, "Otfvis: Matsim's open-source visualizer," pp. 225–234, 2016.

[52] P. L. Hammer and S. Rudeanu, *Boolean methods in operations research and related areas*. Springer Science & Business Media, 2012, vol. 7.

[53] Vite js. [Online]. Available: https://vitejs.dev/guide/

[54] React. [Online]. Available: https://es.react.dev/

[55] Instituto Nacional de Estadisticas y Censos, Censo Ecuador 2022. [Online]. Available: https://www.censoecuador.gob.ec/data-y-resultados/

[56] M. E. D. Correa, "Análisis crítico de la planificación urbana de la ciudad de cuenca," *Maskana*, vol. 7, no. 1, pp. 107–122, 2016.

[57] E. A. Hurtado Duarte, R. Romero González, and J. Paucar Camacho, "Tráfico vehicular y peatonal, un indicador de sostenibilidad urbana para la ciudad de cuenca," 2023.

[58] F. M. Armijos Arcos, "Predicción de ruido por tráfico vehicular y elaboración del mapa de ruido utilizando el modelo harmonoise del centro histórico de cuenca," B.S. thesis, 2018.

[59] N. Rivera, C. Mata, J. Lalangui, A. Bermeo, L. Valdez, and J. Morocho, "Análisis de emisiones contaminantes originados por el parque automotor en cuenca," *Revista Ibérica de Sistemas e Tecnologias de Informação*, no. E30, pp. 376–392, 2020.

[60] R. Armas, H. Aguirre, S. Zapotecas-Martínez, and K. Tanaka, "Traffic signal optimization: minimizing travel time and fuel consumption," in *International Conference on Artificial Evolution (Evolution Artificielle)*. Springer, 2015, pp. 29–43.

[61] C. Larraín-Videla, J.-C. Muñoz, and J. Briones, "Gobernanza de transporte en áreas metropolitanas: revisión crítica y análisis para santiago de chile," *EURE (Santiago)*, vol. 48, no. 145, pp. 1–24, 2022.

[62] B. Kickhofer, D. Hosse, K. Turnera, and A. Tirachinic, "Creating an open matsim scenario from open data: The case of santiago de chile," *TU Berlin, Transport System Planning and Transport Telematics: Berlin, Germany*, 2016.

[63] G. Andrienko, N. Andrienko, P. Bak, D. Keim, and S. Wrobel, *Visual analytics of movement*. Springer Science & Business Media, 2013.

[64] T. Seto, Y. Sekimoto, K. Asahi, and T. Endo, "Constructing a digital city on a web-3d platform," in *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Advances in Resilient and Intelligent Cities*. ACM, 2020.

Appendices

TrafficSim-Vis

.1 Installation

This is a guide for the installation, use, and development of TrafficSim-Vis.

.1.1 Requirements

This single-page application uses Node Package Manager (NPM) and was developed using Visual Studio Code.

- Node JS (18.0v >) with NPM (9.0v >)
- Python (3.10v >), install matsim-tools library

.1.2 Preprocess Files

To perform the visualization in TrafficSim-Vis, you must parse the output files generated by MATSim (**output_network.xml.gz** and **output_events.xml.gz**), from the repository, you must enter to parsingfiles and take *parsingFiles_script.py.*

Now, to execute the file, you enter the name of the script, and the coordinates of the simulation that you want to parsing (for example, epsg:4326), after that, the path of the network and events files, respectively.

```
python parsingFiles_script.py epsg:4326 <output_network.xml.g>...

... <output_events.xml.gz>
```

As a result, we will obtain two files that are ready to upload to TrafficSim-Vis. On the other hand, if you want to modify or continue working on developing the code, you need the following.

.1.3 Development Commands

You need to make a copy of the repository and run the following:

```
npm install
```

This command installed all development dependencies.
After that, run:

```
npm run dev
```

This command runs a local server and usually listens on localhost.
Finally, compiles and minifies for production.

```
npm run build
```