Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Computer Graphics and Interaction

Management of large-scale traffic sensor data using geospatial context

Master's thesis

Bc. Lucia Semanová

Study Programme: Open Informatics, Master's degree
Field of Study: Human Computer Interaction
Supervisor: Ing. Ladislav Čmolík, Ph.D.

Prague, May 2021

MASTER‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name:	**Semanová Lucia**	Personal ID number:	**452851**
Faculty / Institute:	**Faculty of Electrical Engineering**		
Department / Institute:	**Department of Computer Graphics and Interaction**		
Study program:	**Open Informatics**		
Specialisation:	**Human-Computer Interaction**		

II. Master’s thesis details

Master’s thesis title in English:

Management of large-scale traffic sensor data using geospatial context

Master’s thesis title in Czech:

Správa velkých dat z dopravních senzorů pomocí jejich vizualizace

Guidelines:

Get acquainted with the visualization of geographic data. Analyze existing methods and applications for visualization and management of traffic sensors. Further, analyze and evaluate existing map platforms (Google Maps, OpenStreetMaps, HERE maps, etc.) according to their usability for visualization and management of traffic sensors. Based on the analysis, design and implement a high-fidelity prototype of client side of web application for visualization and management of traffic sensors. Focus mainly on easy localization of traffic sensors based on their spatial coordinates, type, and other attributes. Verify the usability of the prototype with qualitative user test with 6 to 10 users performed with data of at least 100 traffic sensors.

Bibliography / sources:

Tamara Munzner, Visualization Design and Analysis, CRC Press, 2015.
Andrienko G., Andrienko N., Jankowski P., Keim D., Kraak M.-J., MacEachren A.M., and Wrobel S., Geovisual analytics for spatial decision support: Setting the research agenda. International Journal of Geographical Information Science, 21(8), pp. 839-857, 2007.
Keim D., Kohlhammer J., Ellis G., and Mansmann F. (eds.), Mastering the Information Age - Solving Problems with Visual Analytics, Eurographics Association, 2010.

Name and workplace of master’s thesis supervisor:

Ing. Ladislav Čmolík, Ph.D., Department of Computer Graphics and Interaction

Name and workplace of second master’s thesis supervisor or consultant:

Date of master’s thesis assignment: **10.03.2021** Deadline for master's thesis submission: **21.05.2021**

Assignment valid until: **19.02.2023**

Ing. Ladislav Čmolík, Ph.D. Supervisor’s signature	Head of department’s signature	prof. Mgr. Petr Páta, Ph.D. Dean’s signature

III. Assignment receipt

The student acknowledges that the master’s thesis is an individual work. The student must produce her thesis without the assistance of others, with the exception of provided consultations. Within the master’s thesis, the author must state the names of consultants and include a list of references.

.

Date of assignment receipt

Student’s signature

Declaration

I hereby declare I have written this master's thesis independently and quoted all the sources of information used in accordance with methodological instructions on ethical principles for writing an academic thesis. Moreover, I state that this thesis has neither been submitted nor accepted for any other degree.

In Prague, May 21, 2021

....
Bc. Lucia Semanová

Acknowledgement

I would like to thank my supervisor Ing. Ladislav Čmolík, Ph.D., for his great support, guidance, and expertise that helped me during this thesis. Also, I would like to express my sincere thanks to GoodVision Ltd. that gave me an opportunity to work on such an interesting topic and the whole team that educated me in the traffic industry and provided me with valuable advice and suggestions. My greatest thanks go to my family that always supports and believes in me, and to all my friends that helped and encouraged me during my studies.

Abstract

This thesis aims to implement a prototype of a system for the visualization and management of traffic sensors. In order to design the prototype, user research with experienced traffic engineers was performed as well as analysis of existing commercial off-the-shelf solutions. The user research showed that users need to work with time-varying geographical data. Furthermore, the analysis revealed that there is no single solution on the current market that would satisfy all user requirements gathered during the user research. These requirements formed the basis on which low-fidelity design was created, which subsequently aided in implementing a high-fidelity prototype. Traffic analysts are able to use this prototype to leverage geographical context for managing historical data as well as to look for emerging patterns. The usability of the prototype was evaluated through a standard usability testing technique with 9 participants. Few issues with the prototype were identified in the usability testing. However, these can be easily rectified, and overall the prototype provides a comprehensive system for traffic data managers.

Key words: traffic analysis, geographical visualization, location, traffic sensors

Abstrakt

Cílem této práce je implementovat prototyp systému pro vizualizaci a správu dopravních senzorů. Za účelem návrhu prototypu byl proveden uživatelský výzkum se zkušenými dopravními inženýry a analýza stávajících komerčních řešení. Výzkum uživatelů ukázal, že uživatelé musí pracovat s geografickými daty proměnnými v čase. Analýza dále odhalila, že na současném trhu neexistuje jediné řešení, které by uspokojilo všechny požadavky uživatelů shromážděné během průzkumu uživatelů. Tyto požadavky byly základem, pro vytvoření low-fidelity prototypu, který byl následně použít při implementaci high-fidelity prototypu. Dopravní analytici jsou schopni použít tento prototyp a využít geografickou informaci pro správu historických dat a také k hledání nových vzorů v dopravě. Použitelnost prototypu byla vyhodnocena pomocí standardní techniky testování použitelnosti s 9 účastníky. Při testování použitelnosti bylo zjištěno několik problémů. Tyto problémy však lze snadno opravit a prototyp tak celkově poskytuje komplexní systém určený pro dopravní analytiky.

Klíčové slová: dopravní analýza, vizualizace geografických dát, lokace, dopravní senzory

Contents

Chapter 1

Introduction

In the last years, the human population increased to almost 8 billion [1], and it continues to grow, especially in the urban agglomerations and around them. With more than 56% of the world's population living in urban areas, traffic congestion has become an increasingly crucial issue for municipalities [2]. The need for mobility and the volume of traffic infrastructure is increasing and will rise further as the proportion of the urban population is estimated to reach 68% by 2050 [3]. With this rapid change, cities need to be ready with practical traffic solutions. Smart cities are offering long-term visions for managing traffic both by optimizing transportation logistics and reducing congestion.

This growth of urban areas has led to an increased need for mobility and increased public traffic infrastructure volume. As our cities become more and more congested, traffic data analytics is becoming more important. Organizations and local authorities responsible for urban planning and setting up speed limits need to understand how vehicles move through the city and in which places or areas they concentrate more frequently [4]. In order to manage and control traffic flows, the conditions of the road traffic have to be captured [5].

One of the possible solutions to control the traffic is historical traffic data analysis. It can help growing cities to map more efficient road systems and plan better zoning with the help of location intelligence [3]. The location in the traffic analysis, modeling, and planning can be important in many aspects. The traffic data based on location can be used across different industries. For example, traffic count data is used by retailers to aim to place their stores in locations that are frequently visited or are easily accessible. With the predictions of traffic volume and traffic flows, companies in the tourism and hospitality industries can plan their logistics wisely, prepare their product inventory and workforce for busy periods, or foresee any supply chain disruption. Even marketers often work with traffic data analysis for digital out-of-home (DOOH) advertising to display ads, for example, on roadside digital billboards when the traffic is congested. Hence, it can be seen by as many drivers as possible. As mentioned before, local authorities and governments have also used traffic data analysis for strategic road planning, identifying dangerously busy areas by analyzing traffic incidents, or adjusting the legal speed limits where there was a spike in collisions. Analyzing historical traffic data can also give us opportunities for traffic innovations and improvements, to build innovative infrastructure, and plan sophisticated, less congested urban areas [4].

As can be seen, the location of the traffic data is essential to put the data in the correct context. Another significant factor while analyzing historical traffic data is time as well as associated detection of traffic patterns and changes over time. When traffic data are put in the context of time reliable predictions of traffic behavior, traffic engineers can plan the urban traffic in the area more efficiently. In addition, there might be repeated anomalies detected over time which can lead to a potential solution to traffic reorganization, speed limitation, or personalized signal lighting for the area.

1.1 Goal of the thesis

This assignment originated from the need for obtaining geographical information of traffic sensors in the existing tool for traffic data collection and analysis, GoodVision Video Insight. Even though there are nearly unlimited opportunities for using the traffic data in a geospatial context, in this thesis, the focus will be to visualize and manage a large number of traffic sensors and extracted traffic data from video footage that vary over time. The goal of this thesis is to examine a traffic sensor management tool designed to cater to the needs of traffic managers and evaluate its usability through user testing. In order to achieve this, the following needs to be done:

- gathering requirements from users working in traffic industry and based on the collected information creation of personas which mirror typical users (Chapter 2)
- analysis of existing solutions for traffic data analysis and methods for visualizing geographical data as well as comparison of available map platforms (Chapter 2)
- design and implementation of a high-fidelity prototype for the visualization and management of traffic sensors (Chapters 3 and 4)
- verification that the designed prototype is usable by performing usability testing with datasets of 100 or more traffic sensors (Chapter 5)

Chapter 2

Analysis

In this chapter, I will start with user research, propose the target audience, the users' requirements, and determine the type and the attributes of data users need to work with. Then I will continue with identifying the tasks that the visualization should help to solve. When user requirements are prepared, I will analyze existing tools for traffic data analysis and decide whether they meet the requirements and how well they would work with the classified data type. Afterward, I will analyze visualization methods and evaluate them with respect to the data type, types of attributes, and visualization tasks. At the end of this chapter, I will compare three available map platforms and decide which one is the most suitable for the case of this thesis.

2.1 System requirements

In this section, I will collect requirements from the user with the help of quantitative and qualitative user research. I will present the result of the research and summarize the specifications for the system. Based on the requirements, I will propose personas and main scenarios for the system. At the end of this section, I will specify the data that users work with and come up with tasks that will help me choose the correct visualization technique.

2.1.1 User research

In this section, I will present the results of my user research. I have covered both the quantitative and qualitative parts of the research, which I will describe below. Before I start, I will define the difference between quantitative and qualitative research. With quantitative research, we search for facts. It can be usefully thought of as a series of **'what?'** questions, e.g., *What number of traffic sensors do you operate with for traffic survey?*. In contrast, qualitative research is associated with **'why?'** questions that reference its emergence in motivational research [6].

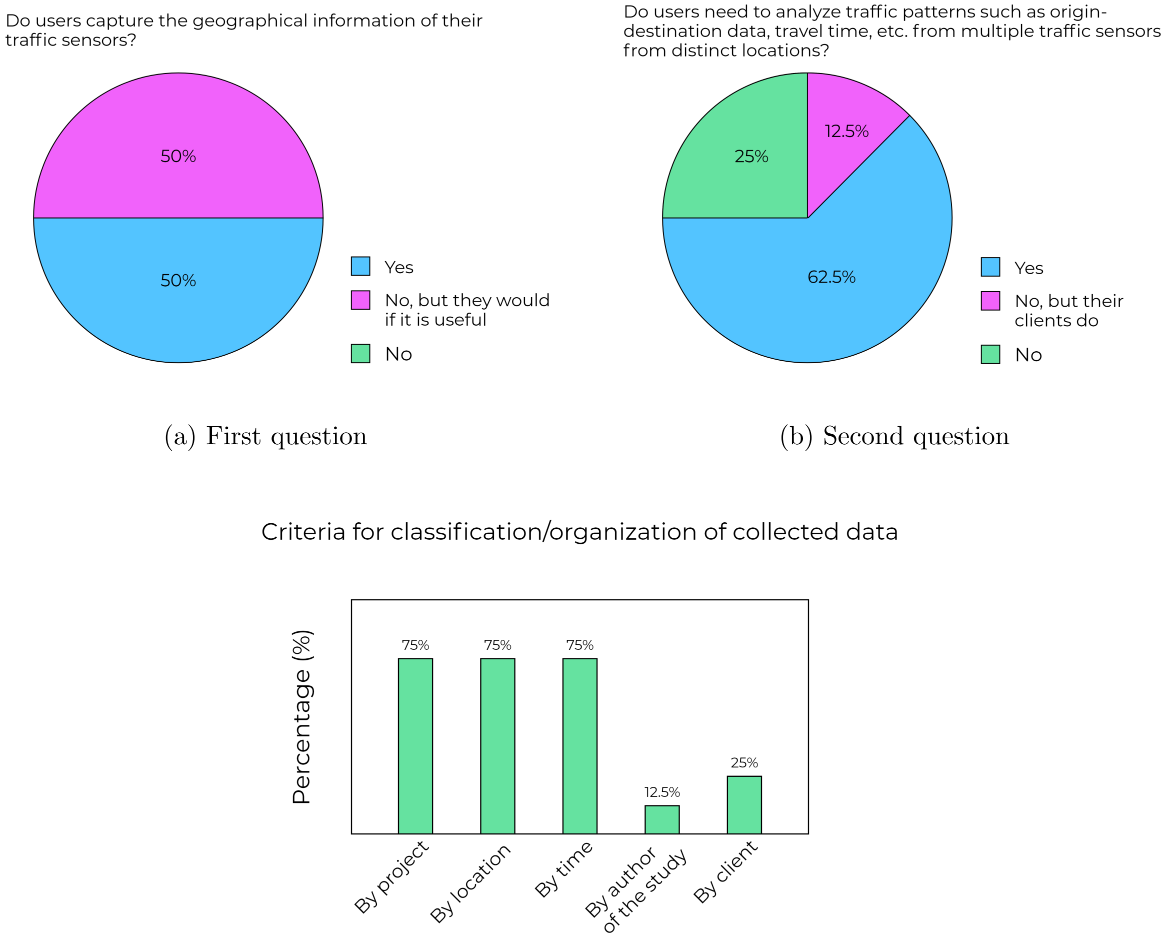
Quantitative research

The quantitative part consisted of the questionnaire, which was sent to GoodVision customers and subscribers, who are already familiar with the product solution. The main goal of the questionnaire was to observe the answers to the three following questions:

1. Are users capable of capturing geographical information of their traffic sensors?

2. Do users need to analyze traffic patterns such as origin-destination data, travel time, etc., from multiple traffic sensors from distinct locations?
3. Based on what criterion do users classify/organize collected data from their traffic surveys?

The questionnaire was composed of various questions that should answer the questions above. Results can be seen in graphs in Figure 2.1.



(a) First question

(b) Second question

(c) Third question

Figure 2.1: Results from quantitative research questionnaire

Even though results from the first question 2.1a shows that only 50% of current customers work with the information related to the location of the traffic sensor, all remaining users are willing to capture this information if they can see the benefit from it. The most frequent reasons why they capture this geo-information about traffic sensors were to get accurate travel time, distance, for speed measurements, or the origin-destination studies and subsequent data mapping and visualization.

The second question was intended to determine if users need to work with traffic data from various traffic sensors simultaneously to obtain general results. As it is shown in Figure 2.1b, 62.5% of the users need to analyze traffic data across multiple traffic sensors.

It is important information as it will be beneficial for the new design to visualize data in a geospatial context from various traffic sensors simultaneously. 12.5% of respondents do not need to do it, but the clients they are doing traffic survey for need to cope with it. That left only 25% of users that consider it acceptable to analyze traffic data just from one traffic sensor at a time.

The last chart 2.1c displays categories that are important for the customers in terms of grouping/organizing their collected data. Up to 75% of respondents chose the project, location, and time as the key factors for classifying their data. It revealed that users work on multiple projects/traffic surveys. This is an interesting finding that will be used later on in Section 2.2.2.

The other way to classify traffic data is by the location that can be sensed as a city, region, motorway from the macro perspective, or an intersection, a roundabout, or a street from the micro perspective. The last major approach to organizing their data is by time. The traffic engineers mostly work with historical traffic data and analyze them. Therefore, time is an important variable that can reveal some repeating patterns in the traffic or, conversely, can detect some anomalies in the traffic.

Besides the results from the main three questions, other observations were made from the questionnaire. The research revealed that 100% of respondents are working on traffic data collections/surveys, and 50% concentrate on transport and traffic modeling. The research showed that most of the users' project contains less than 50 cameras and that 25% do not collect traffic data using traffic sensors.

Qualitative research

As I mentioned earlier, the user research consisted of the qualitative part as well. I managed to contact two traffic engineers who work in the industry for a substantial period, are properly experienced and are aware of new trends in the industry. Neither of the two interviews was taken in person as both respondents are based in the United States. Both interviews gave me a better understanding of the traffic industry, the location data usage and provided me with an overview of the tools that can be used for the traffic analysis. I researched those tools, and a few of them are mentioned in the following sections.

The first interview was with a traffic engineer whose area of expertise is traffic modeling and geometric design of the roadway. I was made aware that the strategy for obtaining traffic data is project-specific. This means that they occasionally use pneumatic tubes for obtaining traffic counts or installing cameras on the roads, but for some projects, they also collect the data from traffic data providers. All the traffic data are provided with geographical information, which allows visualizing the data in specialized platforms, e.g., Vissim [1], Synchro [2]. They are most commonly collecting traffic counts and speed data for their traffic analysis. For very detailed analyses, they are using data such as traffic sensor data, origin-destination data, or vehicle trajectories.

[1] **Vissim** is the tool for traffic engineering and simulation of the traffic flow and impacts, with the possibilities to define multiple what-if scenarios.

[2] **Synchro** is a traffic signal timing software that assists engineers and transportation planners to design, model, optimize, simulate and animate signalized and unsignalized intersections (including roundabouts).

The second interview was with a traffic engineer who is focused on traffic signal timings. The traffic data he used for various projects are mostly obtained from traffic data providers. Subsequently, he uses them in simulation tools for modeling the traffic and diagnosing the problems that could occur. The traffic data that are most notably used in the process are traffic counts, speed data, and traffic density.

Both the quantitative and the qualitative user research showed that the benefits of using geographical information of traffic data could be enormous. However, the value of it depends on the actual project. I collected the types of commonly used traffic data that could be later applied in the new design and get a new perspective on how users work with the provided traffic data. It can lead to redesigning the current Video Insight tool's structure in the future.

2.1.2 Requirements

In this section, I will sum up the needs of the clients that were discovered during the user research:

- Traffic engineers work on multiple traffic surveys simultaneously.
- The number of traffic sensors per survey is usually up to 100.
- The traffic data that are most interesting for the traffic surveyors are traffic counts, traffic speed, and traffic volume.
- There are various types of traffic sensors used for traffic analysis.
- There is a need for analyzing traffic data across multiple traffic sensors within one location (intersection, roundabout, motorway, etc.).
- Regarding traffic surveys that work with historical data, analyzing traffic data with respect to the time is crucial. Therefore time filtering of traffic data is enforced.

2.1.3 Personas

From the results of user research analyzed in Section 2.1.1, I propose three personas that will mirror typical users and try to solve the exposed problems and findings. These personas will be used during the thesis.

The first persona that will be introduced is John. John is 31 years old. He works as a traffic surveyor in a large company. He is in the business for less than 10 years but has worked on many traffic surveys ever since. His job is to create traffic surveys for his clients based on their needs. During the year, he is working on multiple projects simultaneously. He needs to get and process traffic data from approximately 30 cameras per project. One project usually contains cameras across one city from various periods. As John works on multiple projects simultaneously, he prioritizes the client's data organization to enter the traffic data he needs to work with easily. As every client has different requirements from John, he needs to be flexible and able to provide them the results they need. He wants to find a tool for extracting traffic data from video footage where he can easily

organize client's data, obtain or visualize the required traffic data based on the specific time period, as video footage from clients is not always recorded at the same time. John also needs to collect traffic data (e.g., traffic counts) from two, or more, nearly located cameras.
Challenges: The data organization, traffic data visualization filtered by time.

The second persona, Susan, is 37 years old. She works in the industry for 15 years. She is a traffic planner, and together with her team, she won a substantial tender for reorganizing the traffic structure in Boston to increase traffic safety and decrease traffic congestion. Susan and her team will have footage from more than one hundred different cameras across the city. The smaller groups of cameras located near each other need to be analyzed together. The group of cameras is mostly recording the specific intersection or the street/road in the city. Susan's team already recorded the traffic in Boston by traffic cameras. The footage from all cameras is recorded at the same time. Now they are looking for a suitable platform where the traffic data can be obtained from the video recordings. The traffic data they need from the recordings are traffic counts, traffic trajectories, vehicle speed, and traffic volume. In the first part of the project, they need to focus on locations in Boston where the traffic is congested the most. As Susan and her team do not have unlimited time to do a traffic analysis, every team member is responsible for a certain part of the analyses, and they need a tool that allows them to collaborate on the project. Additionally, as a team leader, Susan wants to have all the footage available in one place.
Challenges: How to present a large amount of traffic data to the user.

The last persona is Kevin, who is over 50 years old. He is a traffic modeler focused primarily on signal timing planning. He has excellent experience with many successful projects as he works in the industry for over 30 years. He is already using GoodVision Video Insight for some of his projects. He does not capture the geographical location of his fixed cameras. However, recently he started using drones for recording the traffic scenes, and they provide the geographical location of the footage. Typically, he works on 2-3 projects per year with less than 50 cameras/drones per project. He uses the platform to get traffic data such as traffic counts, speed estimation, and getting traffic reports generated for simulation tools where he can simulate the traffic based on extracted traffic data from his footage. The speed of the vehicles is quite crucial for Kevin, but at the moment, the process of obtaining the vehicle's speed is quite complicated. Furthermore, specific data is not available to Kevin, i.e., the exact distance between the lines he described in the platform.
Challenges: How to integrate traffic sensors with no information about the location (cameras from previous projects)?

2.1.4 Scenarios

Based on the collected requirements from Section 2.1.2, I will formulate the main goals with corresponding scenarios for the system. I have determined two main goals. The first one is related to the navigation in a large number of traffic sensors dependent on geospatial information. This goal is suitable for traffic engineers who work on surveys with many traffic sensors and need to quickly operate with single or multiple traffic sensors. The representative of this group of users that work with the huge number of sensors per project is Susan from Section 2.1.3. In the second goal, traffic patterns over time should be identified in specific locations (intersections, routes, roundabouts, etc.). This goal suits John from Section 2.1.3 as he needs to visualize traffic data with respect to time.

Easily find specific traffic sensor:

- First scenario

  1. The user opens the specific project.
  2. The user uses location search, which navigates him/her in the map.
  3. The user clicks to markers on the map to find the wanted traffic sensor.
  4. The user finds a wanted traffic sensor.
  5. Now, the user can do a desired action with the sensor.

- Second scenario

  1. The user opens the specific project.
  2. The user searches in the list for the name of the specific traffic sensor.
  3. After clicking on the name of the traffic sensor s/he is navigated to the location on the map where the traffic sensor is placed.
  4. Now, the user can do a desired action with the sensor.

- Third scenario

  1. The user opens the specific project.
  2. The user sets the filters to eliminate searched sensors on the map.
  3. From a smaller number of sensors, the user finds the wanted sensor easily.
  4. Now, the user can do a desired action with the sensor.

Get traffic data for a specific location from specific time

- First scenario

  1. The user opens the specific project.
  2. The user switches to the traffic mode.
  3. The user filters the traffic sensors by name.
  4. The user sees traffic timeline across filtered traffic sensors.
  5. The user chooses the specific time that is interesting for him/her.
  6. The user analyzes traffic patterns at a specific time.

- Second scenario

  1. The user opens the specific project.
  2. The user switches to the traffic mode.
  3. The user finds a specific location that is interesting for him/her on the map.
  4. The user selects only the wanted traffic sensors.
  5. The user sees traffic timeline across selected traffic sensors.
  6. The user chooses the specific time that is interesting for him/her.
  7. The user analyzes traffic patterns at a specific time.

2.1.5 Data determination

In this part, I will classify the data and tasks I listed in Section 2.1.2 to be able to choose the suitable visualization techniques better. I will work with the traffic sensor's data. Traffic sensor in the context of this thesis represents some source of video recording's data from fixed cameras, live stream cameras, or drones. Traffic sensor data are extracted from video footage with the help of a trained neural network that can detect and track the traffic objects. Therefore, the better name for traffic sensors in this context would be data sources as traffic sensors can also be other equipment such as, e.g., pneumatic tubes. So from now on, I consider the term **data source** equivalent to the term **traffic sensor**.

Data types

Data types represent what needs to be visualized. I am going to visualize the time-varying geographical data. The geographical data will represent the points of interest, in my case data sources, each with unique **coordinates**. These geographical points have an abstract attribute **name**, two categorical attributes **device type** and **state**, and two ordered time-varying attributes, **traffic volume** (ordinal) and **traffic counts** (quantitative, discrete).

- **Coordinates** - represents the geographical position of a data source
- **Name** - represents the name of the data source, usually contains the name of the road where the data source is located
- **Device type** - represents the type of the traffic sensor with video output (fixed camera, drone)
- **State** - represents what the state of the data source is in the application (empty - without any video footage uploaded, processing - the footage is uploaded but traffic data are being extracted and not available yet, processed - traffic data from the data source are extracted and available to the user, archived - traffic data are old and no longer available, error - something went wrong during the uploading/processing)
- **Traffic count** - represents a count of traffic objects/vehicles extracted from the video footage of the single data source. The traffic count is calculated for every hour of video recording individually.
- **Traffic volume** - is derived from the attribute traffic count and represents the traffic via traffic degrees - high, medium, and low. Traffic volume is also provided for each hour of video footage.

Tasks Types

After describing data types, I know **what** needs to be visualized. Now I need to think about **why** the data should be visualized, for what purpose. To help me realize that, I will propose tasks that the users need to perform. To define these tasks, I will follow Tamara Munzner's task classification [7].

The main task that users need to perform is **identify**. Identifying the characteristics of the target can be understood as, for example, *identify the device type of traffic sensor*, or *identify where the high traffic was.* Moreover, if users have data from various targets, they need to **compare** them. If targets are days in a week, then the task for the action compare could be, for example, *what is the difference between traffic patterns on Wednesday and Friday*. The other possible example is to *compare traffic across multiple locations*.

When I look at the task types from an overview perspective, I see that users need to search for the data sources in the map to perform some other actions. There are multiple reasons why users can search for the specific traffic sensor. Either they know the target but are not aware of the location, so they need to **locate** the sensor, for example, *they need to locate the sensor on a specific motorway or address*. Alternatively, the target and the location can be both known. However, the user wants to **lookup** for some specific information, e.g., *what was the traffic on motorway A27 in England on Wednesday noon*. There is also potential search action if the user does not know the target but knows its characteristic, then s/he needs to **browse** across the data sources to find the sensor with specific attributes.

2.2 Existing products analysis

As a next part of the analysis, I will explore and get familiar with the products that are already on the market, analyze them, find out their weaknesses and strengths to better understand what should be designed.

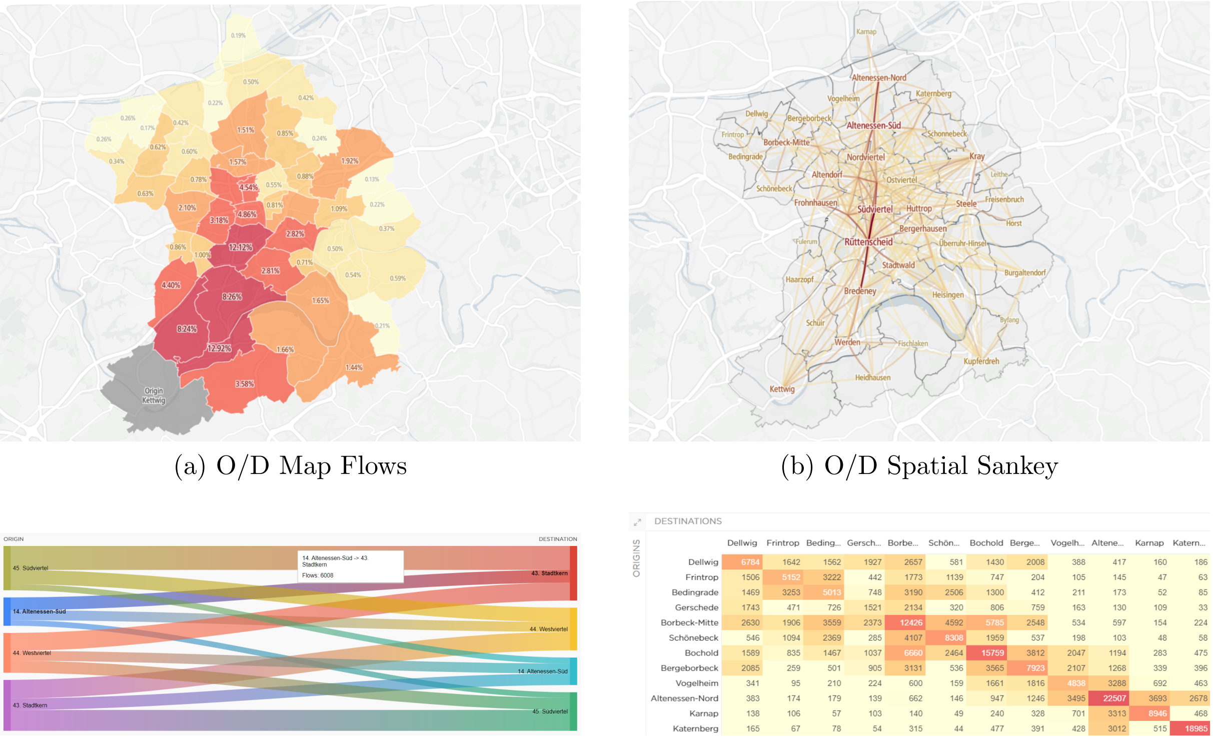
2.2.1 Map management systems

Firstly, I will take a look at map management systems. The platforms I will introduce are specialized in the traffic industry as well. First, I will describe some valuable features from Holland organization TomTom [8] which is a digital mapping and routing company that focuses mainly on car navigation. Then I will present GeoCounts, a product of Transmetric, a company based in Texas, a traffic monitoring, data management, and analysis provider. I will not focus on the traffic analysis while analyzing these products. I will mainly concentrate on how they visualize and manage the data on the map.

TomTom Move [8]

TomTom Move provides multiple solutions, one of them is Origin-Destination (O/D) analysis that is displayed in Figure 2.2. Before I start explaining what particular parts of the figure represent, I will quickly explain what O/D analysis is. As a name tells, O/D represents movement through geographic space, from an origin(O) to a destination(D). This analysis is used to determine travel patterns of traffic what can help to better traffic planning.

As we can see in Figure 2.2, TomTom Move uses multiple visualization methods to represent these origin-destination data. Providing such a huge range of O/D analysis is not an easy task, as TomTom Move needs to collect and combine traffic data across multiple traffic data sources. However, let's take a look at how it deals with representing this information on a map. The first picture 2.2a shows an O/D map representing the regions in Essen, Germany, where the origin is displayed as a gray region and destinations are all other regions in Essen. The region's color is provided by the percentage of the traffic counts that ended their transport in that region. The legend is also provided on the map,



	Dellwig	Frintrop	Beding...	Gersch...	Borbe...	Schön...	Bochold	Berge...	Vogelh...	Altene...	Karnap	Katern...
Dellwig	6784	1642	1562	1927	2657	581	1430	2008	388	417	160	186
Frintrop	1506	5152	3222	442	1773	1139	747	204	105	145	47	63
Bedingrade	1469	3253	5013	748	3190	2506	1300	412	211	173	52	85
Gerschede	1743	471	726	1521	2134	320	806	759	163	130	109	33
Borbeck-Mitte	2630	1906	3559	2373	12426	4592	5785	2548	534	597	154	224
Schönebeck	546	1094	2369	285	4107	8308	1959	537	198	103	48	58
Bochold	1589	835	1467	1037	6660	2464	15759	3812	2047	1194	283	475
Bergeborbeck	2085	259	501	905	3131	536	3565	7923	2107	1268	339	396
Vogelheim	341	95	210	224	600	159	1661	1816	4838	3288	692	463
Altenessen-Nord	383	174	179	139	662	146	947	1246	3495	22507	3693	2678
Karnap	138	106	57	103	140	49	240	328	701	3313	8946	468
Katernberg	165	67	78	54	315	44	477	391	428	3012	515	18985

(a) O/D Map Flows

(b) O/D Spatial Sankey

(c) O/D Sankey diagram

(d) O/D Matrix

Figure 2.2: Origin-Destination Analysis by TomTom Move [8]

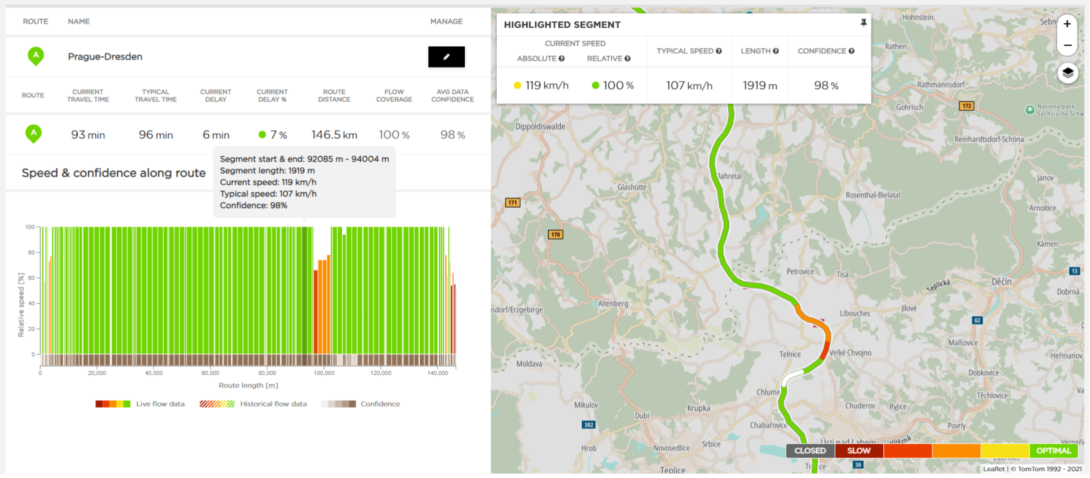
but it is pretty intuitive from my point of view. TomTom Move also offers the possibility to choose the destination region and set the suitable time range. In the second picture 2.2b, the traffic between single regions can be seen. In next pictures 2.2c and 2.2d, the same information is visualized but with different techniques; interactive Sankey diagram (2.2c) and O/D table (2.2d).

As I mentioned at the beginning, TomTom Move is mostly focused on car navigation. Therefore it provides a Route Monitoring service as well. It has a nice user-friendly design, as can be seen in Figure 2.3. It provides live information about traffic on the different road sections in a route that was created beforehand. In this case, route from Prague to Dresden.

Geocounts [9]

Geocounts Traffic, as is shown in Figure 2.4, visualizes continuous counters (green - counters, blue - short term counters, yellow - selected counter) in West Virginia. It provides traffic information as an annual average daily traffic (AADT) over the years, which can be seen in the graph on the left. Geocounts uses World Street Map with Leaflet plugin and GeoJSON layer to visualize these counters. The map contains too many counters, and it is not very easy to navigate between them. What is also not pleasant is the speed of the map. Every operation takes quite a time to render the map after zooming in/out, moving, or choosing a counter. A big number of counters probably causes that.

Geocounts Traffic also enables the visualization of intersections on the map that only makes navigation on the map more complicated. However, after choosing one intersection,



ROUTE	NAME	MANAGE
A	Prague-Dresden	

ROUTE	CURRENT TRAVEL TIME	TYPICAL TRAVEL TIME	CURRENT DELAY	CURRENT DELAY %	ROUTE DISTANCE	FLOW COVERAGE	AVG DATA CONFIDENCE
A	93 min	96 min	6 min	● 7 %	146.5 km	100 %	98 %

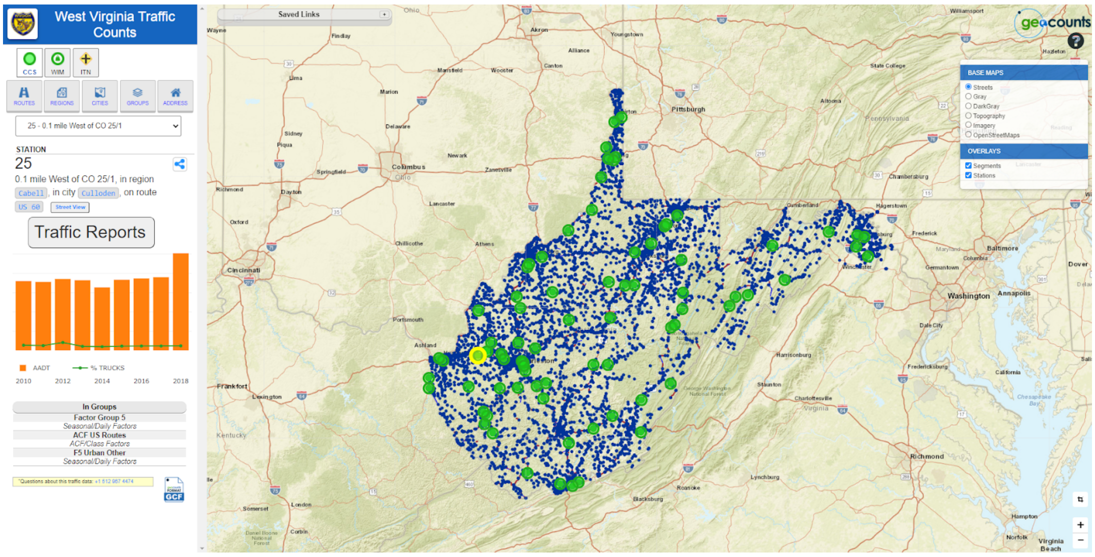
Speed & confidence along route

Segment start & end: 92085 m - 94004 m
Segment length: 1919 m
Current speed: 119 km/h
Typical speed: 107 km/h
Confidence: 98%

HIGHLIGHTED SEGMENT

CURRENT SPEED		TYPICAL SPEED	LENGTH	CONFIDENCE
ABSOLUTE	RELATIVE			
● 119 km/h	● 100 %	107 km/h	1919 m	98 %

Figure 2.3: TomTom Move Route Monitoring [8]



West Virginia Traffic Counts

25 - 0.1 mile West of CO 25/1

STATION
25
0.1 mile West of CO 25/1, in region Cabell, in city Culloden, on route US 60

Traffic Reports

In Groups
Factor Group 5
Seasonal/Daily Factors
ACF US Routes
ACF/Class Factors
F5 Urban Other
Seasonal/Daily Factors

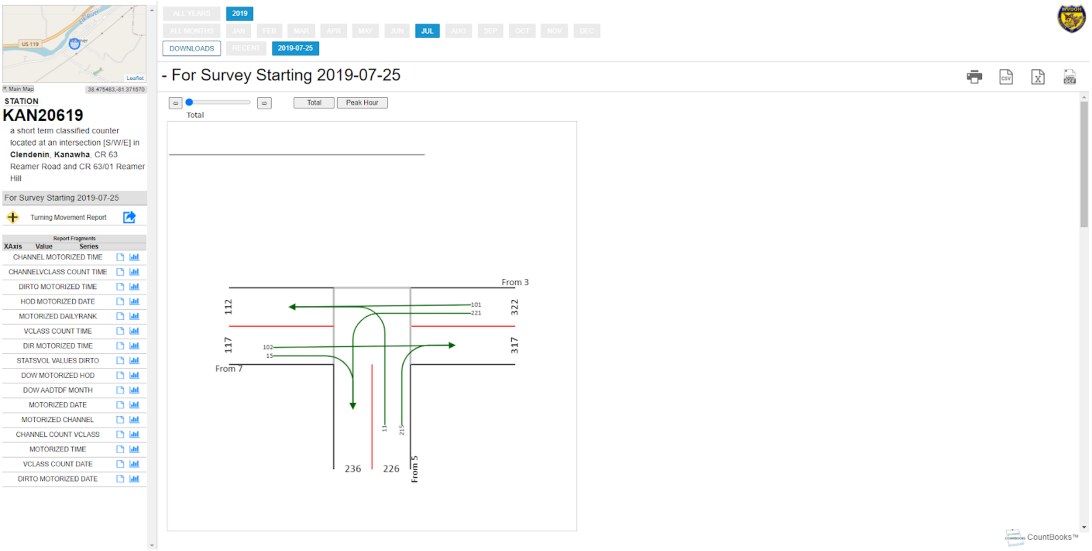
"Questions about this traffic data: +1 512 987 4474

BASE MAPS
● Streets
○ Gray
○ DarkGray
○ Topography
○ Imagery
○ OpenStreetMaps

OVERLAYS
☑ Segments
☑ Stations

Figure 2.4: Geocounts Traffic [9]

it provides an O/D movement report for that intersection, as is displayed in Figure 2.5.



STATION

KAN20619

a short term classified counter located at an intersection [S/W/E] in **Clendenin**, **Kanawha**, CR 63 Reamer Road and CR 63/01 Reamer Hill

For Survey Starting 2019-07-25

Turning Movement Report

Report Fragments

XAxis Value Series
CHANNEL MOTORIZED TIME
CHANNELVCLASS COUNT TIME
DIRTO MOTORIZED TIME
HOD MOTORIZED DATE
MOTORIZED DAILYRANK
VCLASS COUNT TIME
DIR MOTORIZED TIME
STATSVOL VALUES DIRTO
DOW MOTORIZED HOD
DOW AADTDF MONTH
MOTORIZED DATE
MOTORIZED CHANNEL
CHANNEL COUNT VCLASS
MOTORIZED TIME
VCLASS COUNT DATE
DIRTO MOTORIZED DATE

- For Survey Starting 2019-07-25

Figure 2.5: Geocounts Intersection Survey [9]

To summarize Geocounts Traffic, I have to say that the biggest problem I saw is the speed of the map, which can lead to an unpleasant user experience even though the offered features look helpful.

National traffic sensors map systems

Before describing the tools for traffic analysis, I will quickly present some of the national platforms for traffic sensors management that I encountered during the research. The systems from the Czech Republic, United Kingdom, and Australia are shown in Figure 2.6 below.

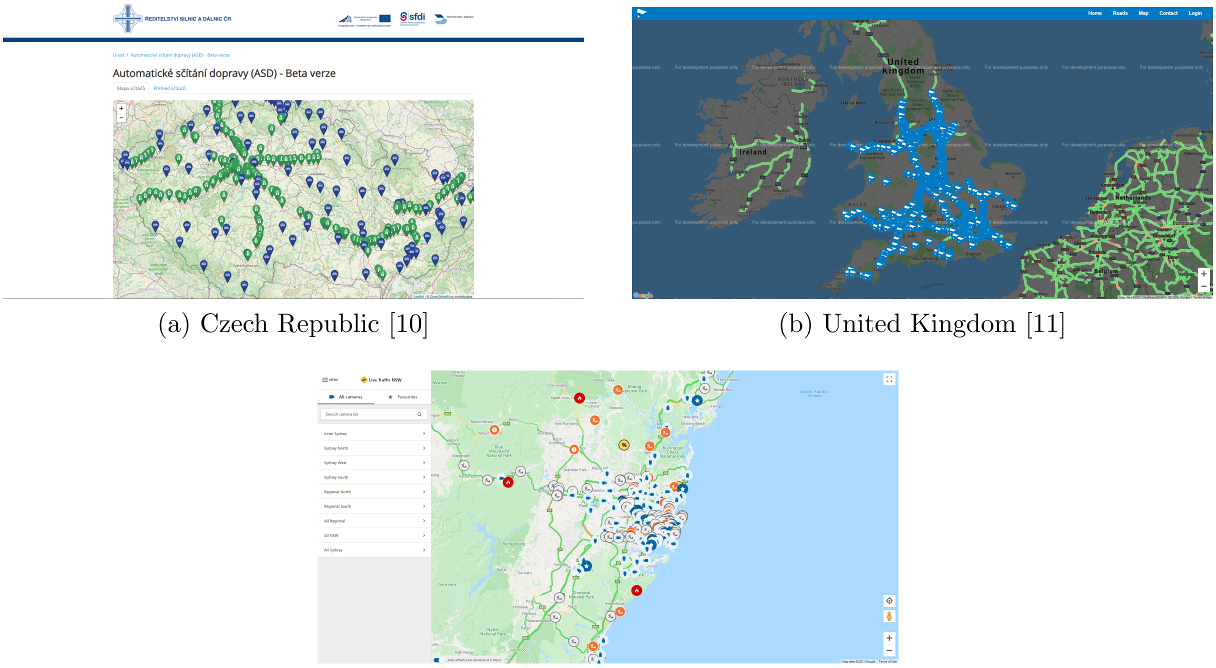
I want to point out that all of these systems visualize the data without any clustering or grouping, which makes the systems overloaded with traffic sensors. That is something that I would like to avoid while designing the thesis' prototype.

2.2.2 Products for traffic data analysis

After the examples of traffic sensors management platforms were described, I will provide examples of available applications for traffic data analysis. Firstly, I will present the solution from Miovision Technologies, which is a Canadian company. After pointing out the available features of this product, I will continue with presenting Czech company RCE Systems and their platform Data From Sky. In the end, I will describe the current solution of the Video Insight platform invented by GoodVision Ltd., based in the United Kingdom.

Miovision DataLink [13]

Miovison, as well as other companies that I will introduce later, has several products. I want to focus on traffic analysis. Therefore I will describe the Miovision DataLink


ŘEDITELSTVÍ SILNIC A DÁLNIC ČR
Automatické sčítání dopravy (ASD) - Beta verze
United Kingdom
Ireland
Live Traffic NSW

(a) Czech Republic [10]

(b) United Kingdom [11]

(c) Australia - Sydney [12]

Figure 2.6: National systems for visualization and management of traffic sensors

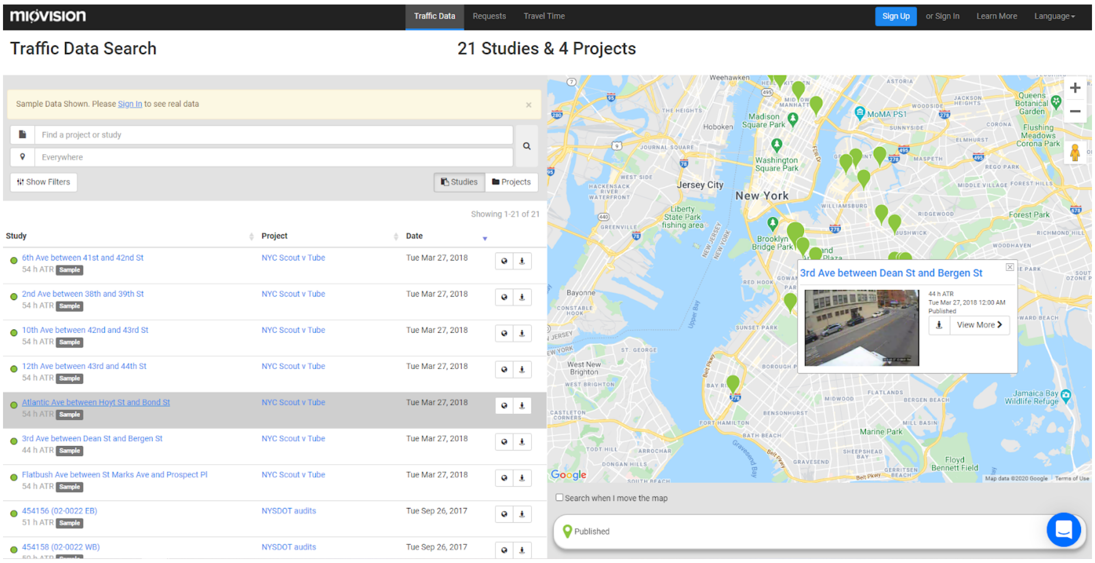
platform, which is designed explicitly for managing traffic projects. The convenience of Miovison DataLink is that with many users, they already have covered many locations by their TrafficLink hardware. Therefore new users can simply request traffic data without the need of own traffic sensors installed.

The process of starting a project with DataLink is following, the user needs to request the data, define specific information about the study type, time of the study, and required classification objects [3]. Then s/he needs to provide a file (excel or KML format) with locations that should be covered. After the user sends the data request and gets over the business process, s/he obtains the traffic data and can analyze them in the application. DataLink already uses the location of the sensors to visualize them on a map for better orientation, as is shown in Figure 2.7. That is something I would like to achieve in this thesis as well.

On the main page, the multiple traffic studies/projects are shown. In DataLink, the project represents a group of studies, and the study represents the traffic data from one location. It can be imagined as traffic data from one traffic sensor. After clicking on the project, a more detailed analysis is displayed; traffic counts and peak hours as is shown in Figure 2.8.

When clicking on the specific study, the detailed traffic count and peak periods are displayed as well. On top of that, video footage or its preview and the origin-destination movement volumes graph are available (see Figure 2.9). Even though the graph is a nice feature, I assume that defining entries/exits is done manually by Miovision, and that is not automated.

[3] **Classification objects** represent the class of the vehicle, e.g., car, motorcycles, bus, etc.



Traffic Data Search

21 Studies & 4 Projects

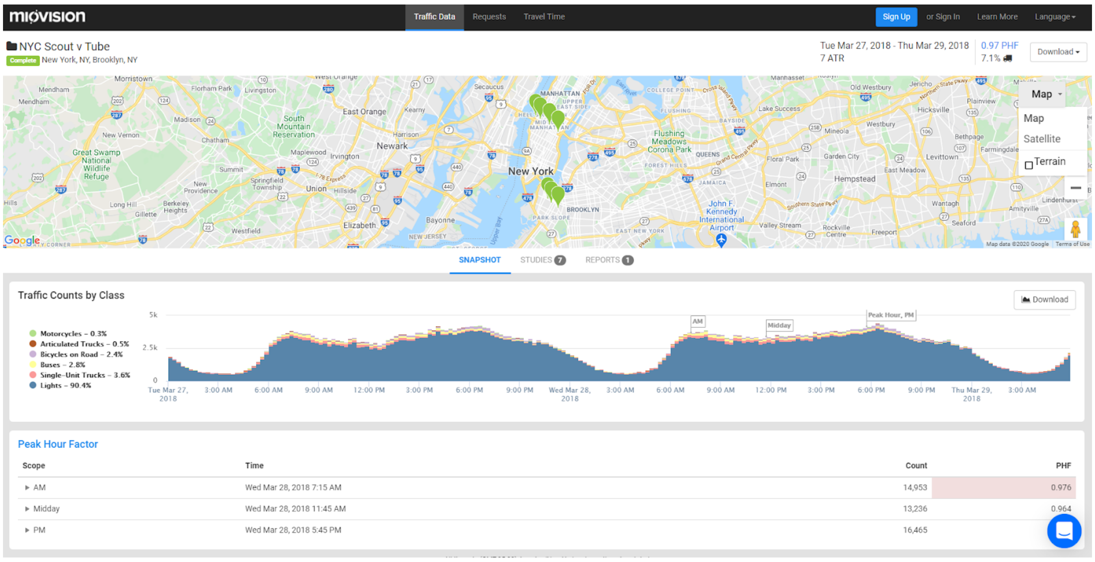
Sample Data Shown. Please Sign In to see real data

Showing 1-21 of 21

Study	Project	Date
6th Ave between 41st and 42nd St 54 h ATR Sample	NYC Scout v Tube	Tue Mar 27, 2018
2nd Ave between 38th and 39th St 54 h ATR Sample	NYC Scout v Tube	Tue Mar 27, 2018
10th Ave between 42nd and 43rd St 54 h ATR Sample	NYC Scout v Tube	Tue Mar 27, 2018
12th Ave between 43rd and 44th St 54 h ATR Sample	NYC Scout v Tube	Tue Mar 27, 2018
Atlantic Ave between Hoyt St and Bond St 54 h ATR Sample	NYC Scout v Tube	Tue Mar 27, 2018
3rd Ave between Dean St and Bergen St 44 h ATR Sample	NYC Scout v Tube	Tue Mar 27, 2018
Flatbush Ave between St Marks Ave and Prospect Pl 54 h ATR Sample	NYC Scout v Tube	Tue Mar 27, 2018
454156 (02-0022 EB) 51 h ATR Sample	NYSDOT audits	Tue Sep 26, 2017
454158 (02-0022 WB) 50 h ATR Sample	NYSDOT audits	Tue Sep 26, 2017

3rd Ave between Dean St and Bergen St
44 h ATR
Tue Mar 27, 2018 12:00 AM
Published

Figure 2.7: Miovision DataLink demo [13]



NYC Scout v Tube
Complete New York, NY, Brooklyn, NY

Tue Mar 27, 2018 - Thu Mar 29, 2018
7 ATR

0.97 PHF
7.1%

Traffic Counts by Class

Peak Hour Factor

Scope	Time	Count	PHF
AM	Wed Mar 28, 2018 7:15 AM	14,953	0.976
Midday	Wed Mar 28, 2018 11:45 AM	13,236	0.964
PM	Wed Mar 28, 2018 5:45 PM	16,465	

Figure 2.8: Miovision DataLink project [13]

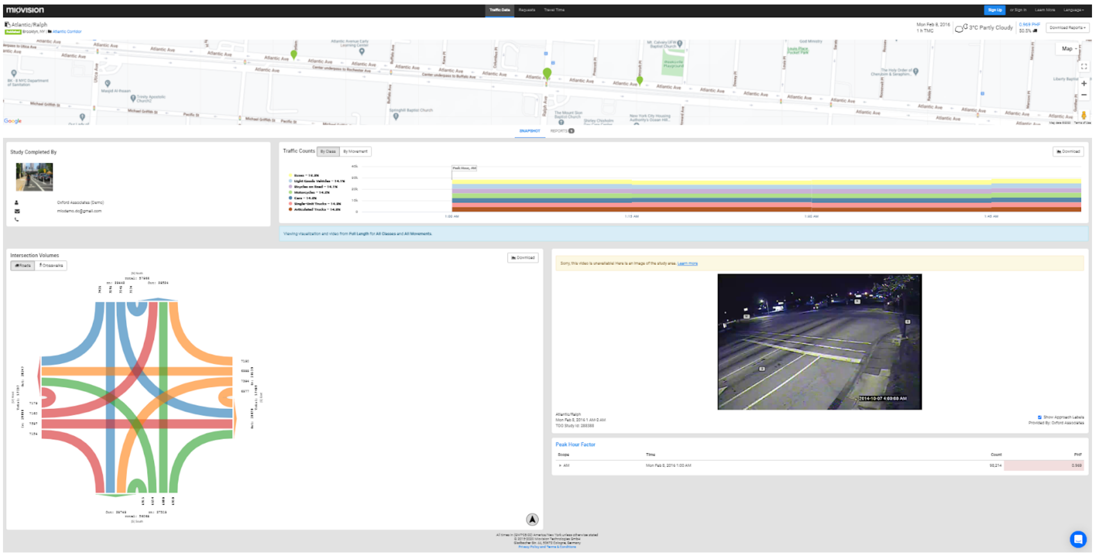
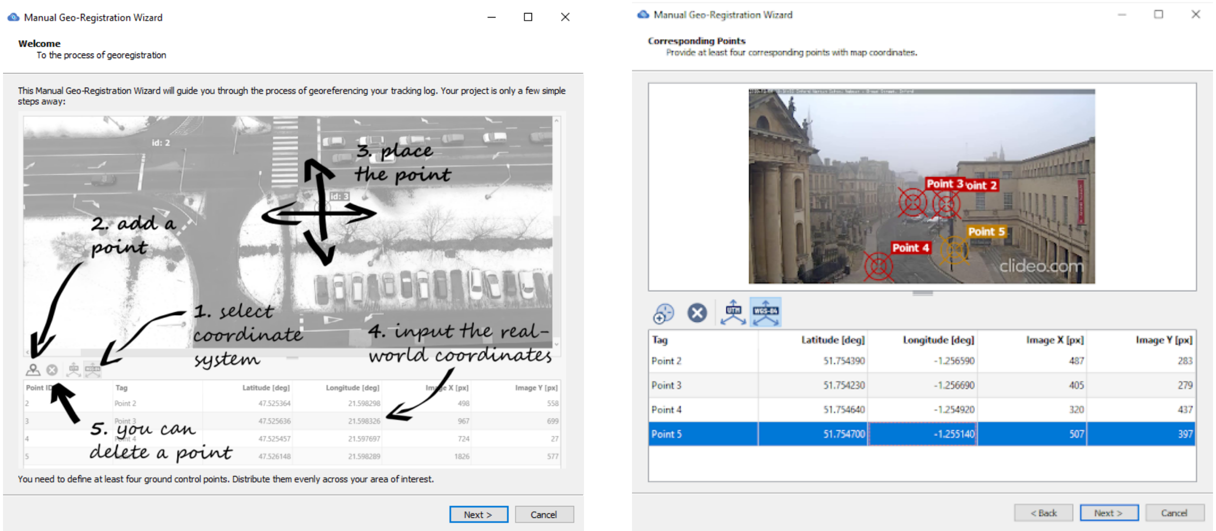


Figure 2.9: Miovision DataLink study [13]

To summarize the Miovision DataLink platform, I would like to point out the features related to map visualization that I consider handy. The main page visualizes the studies with their locations on the map. When hovering over the name of the study or project, the appropriate marker or group of markers highlights on the map. It is a nice detail but not very noticeable as the marker only gets bigger for a while. What makes me wonder is how the users can orient in the map when proceeding with many studies on a small area. Only a few studies can be seen in the demo, so even though some of the markers overlap, it is not hard to navigate between them. I would also like to point out the movement volume diagram, from Figure 2.9, that I believe is a very convenient feature for the users.

DataFromSky TrafficSurvey [14]

The TrafficSurvey platform is one of the products of DataFromSky. It is designed for extracting traffic data for users' traffic surveys. The data are extracted from video footage with the help of artificial intelligence. It allows both the footage from fixed cameras and drones. The web application does the video uploading part. However, for further analysis, the user needs to download their desktop application DataFromSky Viewer. After processing the footage, s/he receives a file with extracted data via email that can be uploaded to the desktop application. For analyzing this platform, I registered here and tried it myself. I used a downloaded 30 minutes long video stream recording from the Youtube channel located on Broad Street in Oxford. While uploading the recording to the platform, I needed to provide the location of the camera. After obtaining extracted data, I downloaded them and then uploaded them to the Viewer application. The TrafficSurvey allows the user to set up the location of the view of his/her footage. The process of this geo-registration is shown in Figure 2.10. The information about the location can be important when there is a need to compute traffic data such as speed or acceleration of the vehicles that cannot be computed without knowing the distances between points. It is a valuable feature, but I have to say that the process of entering the location was painful from the user-experience view. Even with the given manual, it was not very clear, and the fact that I needed to find the coordinates for every point individually in a browser map did not help.



Manual Geo-Registration Wizard

Welcome
To the process of georegistration

This Manual Geo-Registration Wizard will guide you through the process of georeferencing your tracking log. Your project is only a few simple steps away:

id: 2
id: 3
3. place the point
2. add a point
1. select coordinate system
4. input the real-world coordinates
5. you can delete a point

Point ID	Tag	Latitude [deg]	Longitude [deg]	Image X [px]	Image Y [px]
2	Point 2	47.525364	21.598298	498	558
3	Point 3	47.525636	21.598326	967	699
4	Point 4	47.525457	21.597697	724	27
5	[illegible]	47.526148	21.598289	1826	577

You need to define at least four ground control points. Distribute them evenly across your area of interest.

(a) Manual for geo-registration

Manual Geo-Registration Wizard

Corresponding Points
Provide at least four corresponding points with map coordinates.

Tag	Latitude [deg]	Longitude [deg]	Image X [px]	Image Y [px]
Point 2	51.754390	-1.256590	487	283
Point 3	51.754230	-1.256690	405	279
Point 4	51.754640	-1.254920	320	437
Point 5	51.754700	-1.255140	507	397

(b) Providing points

Figure 2.10: DataFromSky TrafficSurvey geo-registration [14]

After extracting the data, a list of trajectories is displayed in the application, and they are also visualized in the video footage if provided. It is also possible to play the video and see different attributes after setting them up. It also supports extracting reports in various formats. The application is hard to use from the beginner's point of view and not very user-friendly. It took me much time to get familiar with it. I am sure that when people learn how to use it, it provides them satisfying traffic analysis.

Overall, the platform providing opportunity for users to set the location of the traffic sensor by themselves is a nice feature. However, without any expertise, it can lead to many possible mistakes that provide incorrect traffic data and result in wrong traffic conclusions from the side of traffic engineers.

GoodVision Video Insight [15]

As the last, I will introduce the GoodVision Video Insight platform for traffic analytic, describe the current state [4] of the tool, and point out the opportunities for enhancement.

GoodVision Video Insight is a tool providing automated traffic data collection and analytics from standard camera recordings, live camera streams, or drone footage. Traffic data are extracted from every video automatically. After this processing, the tool provides a visual presentation of traffic data, the possibility of obtaining traffic data visualizations and analytics based on the specific scenario, and the possibility of exporting all or filtered traffic data from the footage into reports.

Video Insight is composed of two main pages. On the first page is an overview of all

[4] The current state of **Video Insight** platform was analyzed before May 2021, when a newly redesigned version was released

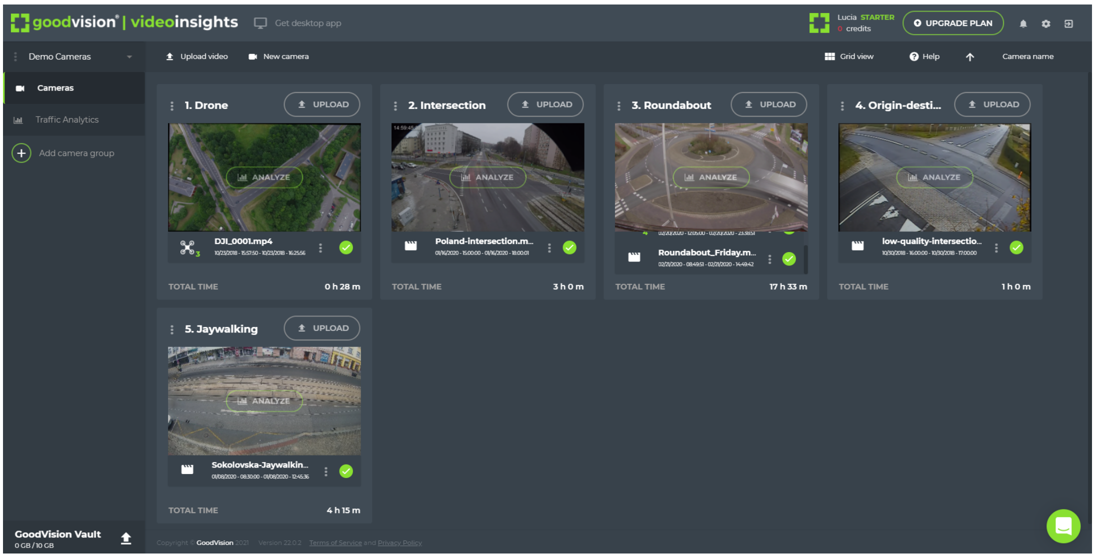


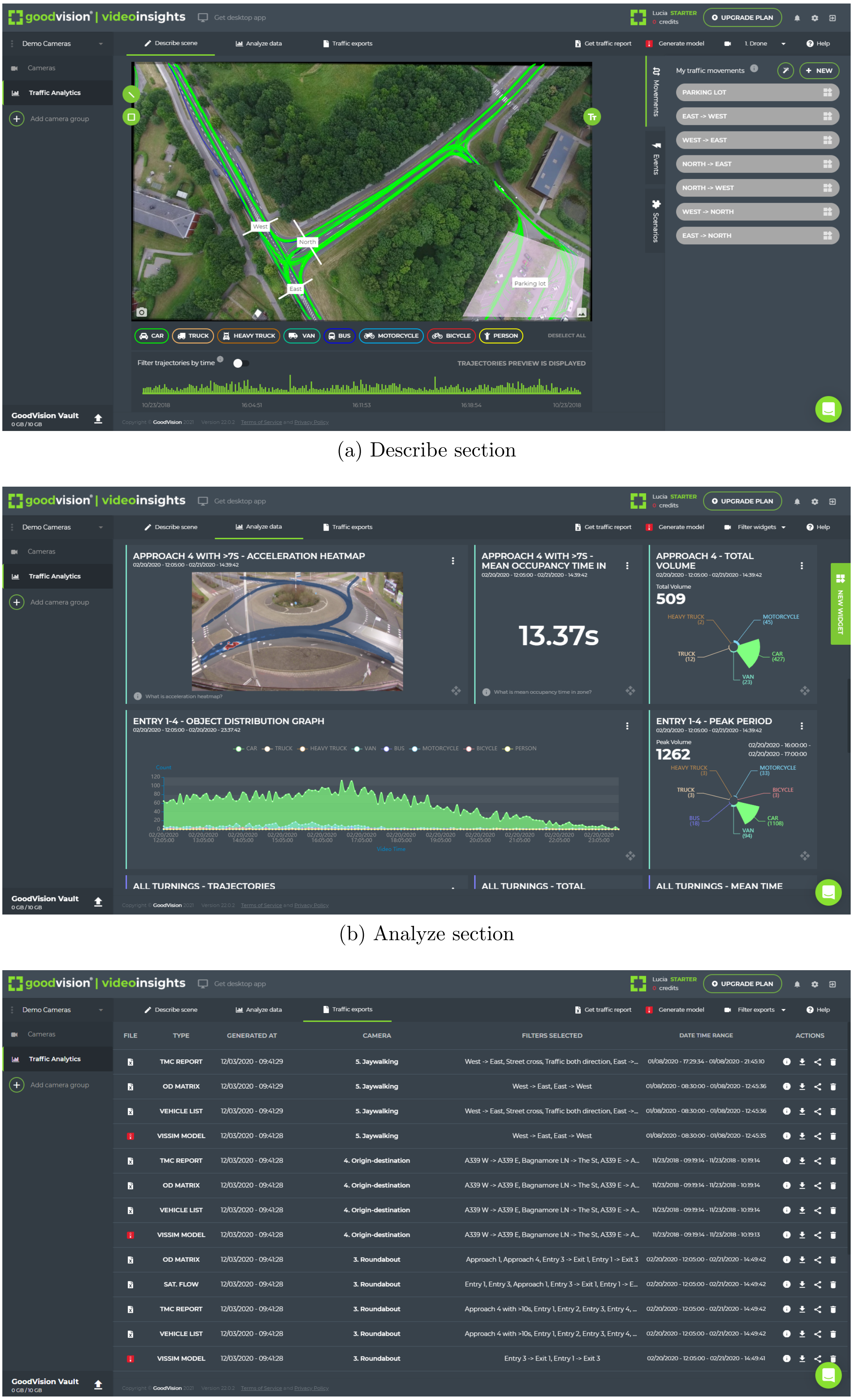
Figure 2.11: Video Insight camera recordings page [15]

cameras (camera represents the subset of videos captured by the same traffic sensor) in a specific camera group, as can be seen in Figure 2.11. The second page, Traffic Analytics, is used for analyzing the processed videos. It has three main sections shown in Figure 2.12.

The first section, Describe scene 2.12a, is designed for users to interactively mark specific parts of the road they need to analyze, e.g., intersection gates, parking lot, etc. When the scene is described, the users can create various traffic widgets, e.g., total volume, that will subsequently appear in Analyze section, shown in Figure 2.12b. The user can analyze the data extracted and visualized from video recordings there. The last part of this page Report section, shown in Figure 2.12c, enables the user to define what should be included in the report. Different types of reports are then presented in this section.

The structure of the GoodVision Video Insight platform is composed of camera groups that can contain cameras. Every camera represents a folder for single-source videos, which means the camera should only include recording from one source of data. The question is, what should the camera group represent? As it was even not clear to the users of this platform, the client's research was done and revealed some interesting findings. The research showed it is not intuitive for the users how they should create these camera groups. They were asked about the way how they want to group their data. The research exposed that the users would like to group/see traffic data representing one intersection, street, roundabouts, etc., together. So basically, it should represent one concrete location. However, the way how they use the platform is totally different. They are using a camera group as a collection of all traffic data from a single survey which contains various locations and many times the big amount of individual sources of data (as mentioned before, this source can be fixed camera, live stream camera, or drone). This organization of the data in the application can lead to an intricate user interface, and data management can become very difficult.

These findings from the client's research led to redesigning the whole approach and structure of the Video Insight platform. This redesign is already deployed at the time of submitting my thesis. Let's take a quick look at how this new design affects the platform.



(a) Describe section

(b) Analyze section

Describe scene | Analyze data | Traffic exports | Get traffic report | Generate model | Filter exports | Help

FILE	TYPE	GENERATED AT	CAMERA	FILTERS SELECTED	DATE TIME RANGE	ACTIONS
	TMC REPORT	12/03/2020 - 09:41:29	5. Jaywalking	West -> East, Street cross, Traffic both direction, East ->...	01/08/2020 - 17:29:34 - 01/08/2020 - 21:45:10	
	OD MATRIX	12/03/2020 - 09:41:29	5. Jaywalking	West -> East, East -> West	01/08/2020 - 08:30:00 - 01/08/2020 - 12:45:36	
	VEHICLE LIST	12/03/2020 - 09:41:29	5. Jaywalking	West -> East, Street cross, Traffic both direction, East ->...	01/08/2020 - 08:30:00 - 01/08/2020 - 12:45:36	
	VISSIM MODEL	12/03/2020 - 09:41:28	5. Jaywalking	West -> East, East -> West	01/08/2020 - 08:30:00 - 01/08/2020 - 12:45:35	
	TMC REPORT	12/03/2020 - 09:41:28	4. Origin-destination	A339 W -> A339 E, Bagnamore LN -> The St, A339 E -> A...	11/23/2018 - 09:19:14 - 11/23/2018 - 10:19:14	
	OD MATRIX	12/03/2020 - 09:41:28	4. Origin-destination	A339 W -> A339 E, Bagnamore LN -> The St, A339 E -> A...	11/23/2018 - 09:19:14 - 11/23/2018 - 10:19:14	
	VEHICLE LIST	12/03/2020 - 09:41:28	4. Origin-destination	A339 W -> A339 E, Bagnamore LN -> The St, A339 E -> A...	11/23/2018 - 09:19:14 - 11/23/2018 - 10:19:14	
	VISSIM MODEL	12/03/2020 - 09:41:28	4. Origin-destination	A339 W -> A339 E, Bagnamore LN -> The St, A339 E -> A...	11/23/2018 - 09:19:14 - 11/23/2018 - 10:19:13	
	OD MATRIX	12/03/2020 - 09:41:28	3. Roundabout	Approach 1, Approach 4, Entry 3 -> Exit 1, Entry 1 -> Exit 3	02/20/2020 - 12:05:00 - 02/21/2020 - 14:49:42	
	SAT. FLOW	12/03/2020 - 09:41:28	3. Roundabout	Entry 1, Entry 3, Approach 1, Entry 3 -> Exit 1, Entry 1 -> E...	02/20/2020 - 12:05:00 - 02/21/2020 - 14:49:42	
	TMC REPORT	12/03/2020 - 09:41:28	3. Roundabout	Approach 4 with >10s, Entry 1, Entry 2, Entry 3, Entry 4, ...	02/20/2020 - 12:05:00 - 02/21/2020 - 14:49:42	
	VEHICLE LIST	12/03/2020 - 09:41:28	3. Roundabout	Approach 4 with >10s, Entry 1, Entry 2, Entry 3, Entry 4, ...	02/20/2020 - 12:05:00 - 02/21/2020 - 14:49:42	
	VISSIM MODEL	12/03/2020 - 09:41:28	3. Roundabout	Entry 3 -> Exit 1, Entry 1 -> Exit 3	02/20/2020 - 12:05:00 - 02/21/2020 - 14:49:41	

(c) Reports section

Figure 2.12: Video Insight Traffic Analytics page [15]

As the research showed, most of the clients work with Video Insight for their traffic surveys. This information forces us to add one more layer to the structure of the platform. Therefore in a new version, there will be possible to create multiple **projects** representing traffic surveys, and for every project, add locations that will include different data sources. Data sources in location can be imagined as multiple cameras in the individual intersection, street, etc. So, **camera** was replaced by **data source** as it did not make sense to call it camera when it represents multiple options such as fixed cameras, live streams, or drones. Moreover, **camera group** became a **location**, as the client's research showed, this is how the users want to group data.

Requirements fulfillment

Even though TomTom, Geocounts, and Miovison DataLink enable visualizing traffic data in a geographical context, they do not provide an option to obtain traffic data from the users' video recordings. They are only able to present traffic data from their data providers. On the other hand, when researching national systems of traffic sensors, they offered only real-time data, which is also not the focus of the target group. The products that enable the processing of the own video footage are DataFromSky TrafficSurvey and GoodVision Video Insight. Regardless of the possibility of DataFromSky TrafficSurvey to set the geoinformation about the traffic sensor's location, it does not provide an option of visualizing the traffic data in that context or even more traffic sensors together. It is also worth mentioning the platform that this thesis should extend, GoodVision Video Insight; it does not keep or work with the geographical information of the traffic sensors yet. Apparently, there does not exist a product that can fulfill all the users' requirements. Therefore there is still a place to design a new system for managing and visualizing traffic data provided from own camera recordings in a geospatial context.

2.3 Visualization analysis

This section will prepare an overview of the visualization and clustering methods used to represent the geographical data. Firstly, I will cover some of the ways how geographic/traffic data are visualized in general. As I should also focus on the large-scale distribution of the data sources, I need to think about how the big amount of the traffic data will be visualized. Therefore, I will also cover clustering techniques that could be suitable for the traffic sensors data. At the end of this section, I will compare available map clustering methods and decide which one is the most suitable for the final product.

2.3.1 Geographical data visualization

Before I present some geovisualization methods, I will define what geospatial data is. Geospatial data describes data that represent objects, features, events, or phenomena on the Earth's surface. It does not matter if it is artificial or natural, if it has a specific location on the globe, it's geospatial. The location may be static in the short-term, e.g., the location of a road, or dynamic, e.g., a moving vehicle or pedestrian. Geospatial data combines location information (usually coordinates on the Earth), attribute information (the characteristics of the object, event, or phenomena concerned), and often also temporal information (the time or life span at which the location and attributes exist) [16].

Geospatial data is most useful when it can be discovered, shared, and used. That is the reason why we want to visualize it. In the context of this thesis, we are talking about the location of traffic sensors that scan the traffic on the roads, thanks to whom the traffic data can be extracted and analyzed afterward.

Geovisualization can be described as "the use of visual geospatial displays to explore data and through that exploration to generate hypotheses, develop problem solutions and construct knowledge" [17]. So what is the reason for the increasing interest in geovisualization over the last 15 years? One of the driving forces is the need to analyze and explore a dramatically increasing amount of geospatial data that are routinely collected these days by many scientific and governmental institutions, private companies, and individuals. It is due to the increasing availability and decreasing cost of technology to acquire store, and process these data [17]. In our context, the examples can be the location of traffic sensors or the location of vehicle trajectories.

There are many methods and techniques in geovisualization, e.g., choropleth maps, cartograms, heat maps, cluster maps, or many others. Instead of defining all the visualization methods, I first need to clarify what data I have, what they represent, and which attributes I want to visualize. For this thesis, I have given geospatial data of traffic sensors that consist of traffic sensor's location coordinates and other attributes described in Section 2.1.5. The detailed information about the data is covered at the beginning of this chapter in Section 2.1.2. In the following part of this section, I will focus on mapping the geospatial data attributes to graphic variables to bring accurate information to the users.

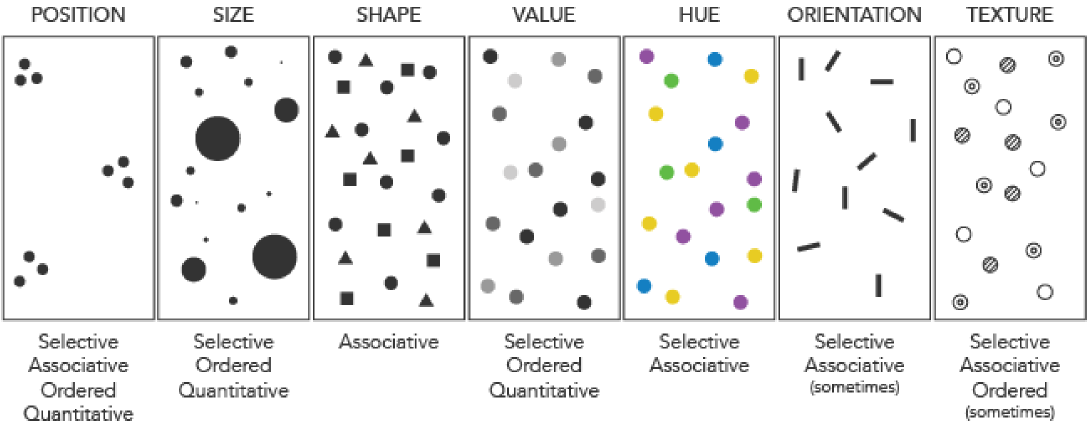


Figure 2.13: Bertin's Visual Variables [18]

Visual variables are the differences in map elements as perceived by the human eye [18]. No matter the type of map, these are the fundamental ways graphic symbols can be distinguished. Jacques Bertin, a French cartographer, and theorist, and a world-renowned authority on the subject of information visualization, proposed an original set of "retinal variables" [19], namely position, size, shape, color value, color hue, orientation, and texture. You can see the graphical variables in Figure 2.13. Other researchers have further expanded the list of variables by color saturation, arrangement, crispness, resolution, and transparency. These variables are means of communicating data to a map reader. Variables can also be combined to represent the same information redundantly, using both size and color value or a combination of different variables. It provides better selectivity and facilitates the judgment of quantitative differences on a map in comparison to using just one variable [17].

2.3.2 Data clustering methods

The primary purpose of this thesis is to design a management system for the traffic sensors that would be easy to use, and to work with the traffic sensors, e.g., searching for specific sensors, would be intuitive and smooth. In Figure 2.14, it can be seen that the density of traffic cameras is enormous. Naively plotting this large number of markers on a map can quickly lead to a poor user experience. It can result in visual overload and slow interaction with a map. It is evident that searching for a specific traffic camera would not be an easy task. Therefore, using data clustering is appropriate. This section will describe an overview of suitable clustering methods that are currently used to solve similar problems.

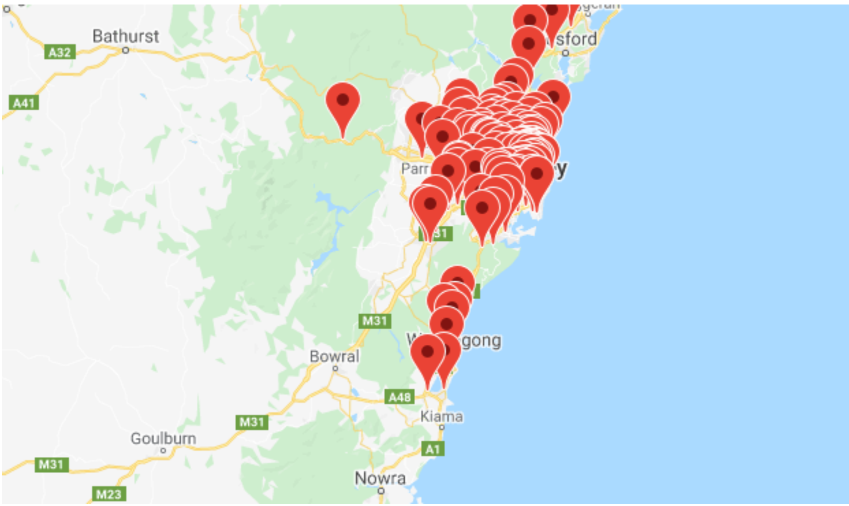


Figure 2.14: Traffic cameras in Sydney [5]

Let's start with a definition of clustering. Clustering is a descriptive task that seeks to identify homogeneous groups of objects based on the values of their attributes, simply explained grouping objects based on their similarity. In a spatial context, the attribute that grouping is based on is the exact location of the spatial objects [20]. Current spatial clustering techniques can be broadly classified into three categories; partitional, hierarchical, and density-based algorithms. In the following part, I will describe one algorithm for every category of clustering techniques I listed above, even though many other clustering algorithms exist.

Partition-based clustering

Probably the most popular partition-based clustering is K-means. It is one of the simplest unsupervised learning algorithms used for clustering. The algorithm partitions n objects into k clusters. K is a desired number of clusters. Each object is then assigned to the closest cluster. Each cluster is then updated based on objects that were assigned to the

[5] Coordinates of traffic cameras are provided by https://opendata.transport.nsw.gov.au/

cluster. This process repeats until no objects are changing anymore. In the end, each object belongs to the cluster with the nearest mean. This algorithm is iterative and aims to minimize a squared error function $f = \sum_{j=1}^{k} \sum_{i=1}^{n} \|x_i^j - c_j\|^2$ where $\|x_i^j - c_j\|^2$ is a chosen distance measured between the object x_i^j and the cluster center c_j [20].

Hierarchical clustering

The representative of hierarchical clustering is a grid-based method. Even though it is not a typical example, it is relevant for this thesis. This algorithm divides the spatial area into rectangular cells, which form a hierarchical structure. It creates a cluster at a particular object and adds objects in the grid cell's bounds to the cluster. It repeats this process until all objects are allocated to the closest grid-based clusters based on the map's zoom level. For each cell, there are two types of parameters, attribute-dependent and attribute-independent parameters. An independent parameter, in this case, is the number of objects in a grid cell, as it can vary based on zoom level. The attribute-dependent parameters are m - mean of all values in the grid cell, s - standard deviation of all values in the grid cell, min - the minimum number of objects in the cell, max - the maximum number of object in the cell and the type of the distribution in the grid cell.

Density-based clustering

Density-based Spatial Clustering of Applications with Noise (DBSCAN) will represent density-based algorithms. This algorithm creates clusters with minimum size and density [20]. The clusters are represented as regions of high density, which are separated by regions of low density [21]. The similarity measure of this method is based on density, as the name of the algorithm tells. Density is defined as a number of objects within a certain distance of each other. The DBSCAN algorithm uses two parameters, *Eps* and $MinPts$, to control the density of the cluster. $Minpts$ indicates the minimum number of points in any cluster. *Eps* indicates a radius. The $Eps - neighborhood$ of the object is then defined as $N_{Eps}(p) = \{q \in D | dist(p,q) \leq Eps\}$. The distance function determines the shape of the neighborhood. The algorithm specifies *core points*, the one in the cluster, and *border points*, that are on the border of the cluster.

DBSCAN starts with an arbitrary object p and retrieves all objects density-reachable from p with respect to *Eps* and $MinPts$. If p is a core point, this procedure yields a cluster. If p is a border point, that means no objects are density-reachable from p, and DBSCAN visits the next object. Two clusters are merged if they are "close" to each other. Basically, the algorithm searches for objects within a distance of *Eps* of each other. If their number exceeds $Minpts$, a cluster is created.

With the knowledge of these clustering techniques, I will compare two of them and decide which one is more suitable for the determined data in the following section.

2.3.3 Visual comparison of clustering methods

After I defined the clustering algorithms, I tried to visualize available traffic cameras [6]. There are 1958 traffic cameras across England shown in Figure 2.15. This visualization is

[6] Coordinates of traffic cameras are provided by https://trafficcameras.uk/

hard to read and orient in. It is also not evident how many cameras are there. Browsing between cameras and find the specific one would definitely not be an easy task for the users.

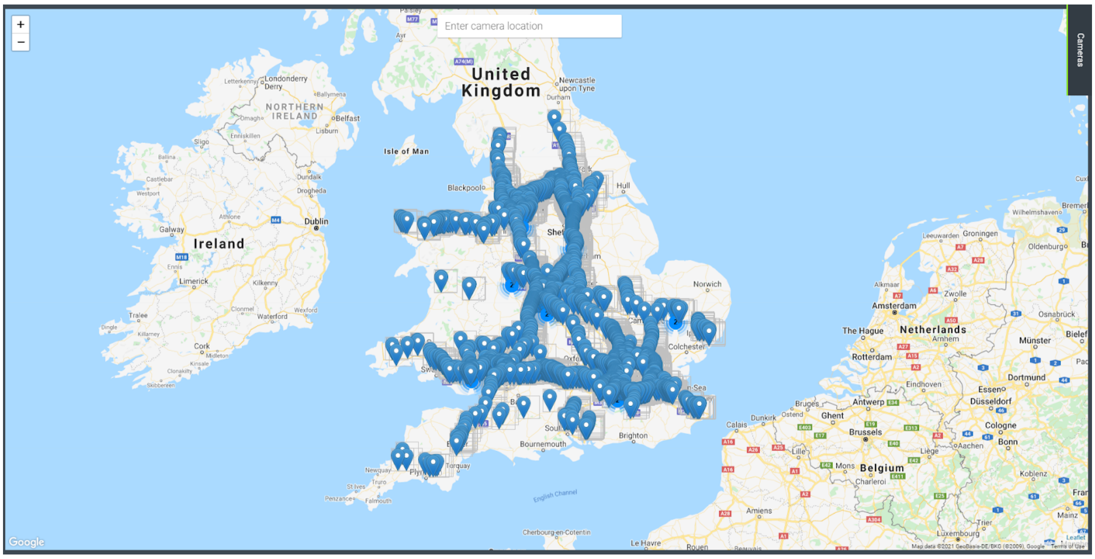
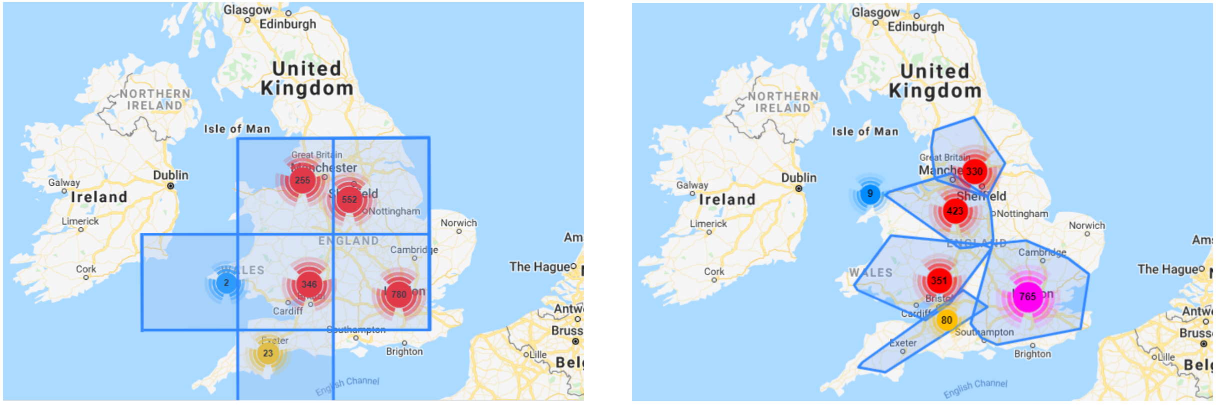


Figure 2.15: Traffic cameras across England

Therefore, I used grid-based and density-based algorithms for this set of data. I tried multiple setups for both of them to achieve the desired result. In the grid-based algorithm, I was changing the size of the grid, and for the density-based algorithm, I was modifying cluster radius. One of the iterations is shown in Figure 2.16. On the left 2.16a, the grid-based clustering is depicted. The grid structure can be observed in this clustering which does not mirror the real-world distribution of the cameras. In the right picture 2.16b, density-based clustering is illustrated. It is clear that the distribution of the cameras is relatively better and that the most significant number of cameras is around London.



(a) Grid-based clustering

(b) Density-based clustering

Figure 2.16: Comparison of clustering algorithms performed on traffic cameras in the United Kingdom

The distribution of density-based clusters is more natural and intuitive. For that reason, the density-based algorithm is going to be used for the purposes of the thesis.

2.4 Maps analysis

For the implementation part of the thesis and the later development of Video Insight, the map platform needs to be integrated into the system. With this intention, I will present three available map platforms that are used worldwide, Google Maps, Here Maps, and Open Street Maps. Firstly I will point out the criteria that will be important for my thesis, then I will compare the platforms, and at the end of this section, I will decide which platform will be used for the implementation.

2.4.1 Platforms comparison

In the following part, I will introduce the main criteria that the platform needs to fulfill and then summarize the features of a specific map platform.

Criteria

1. Does the platform allow visualize multiple custom locations?
2. Does the platform allow clustering of custom locations?
3. Does the platform support rendering of custom data layers [7] on the top of the map?
4. Does the platform allow customizing the map objects (markers, shapes, overlays, etc.)?
5. Does the platform support geocoding [8]?

I will give an overview of the map platforms below to find out if they meet the criteria.

Google Maps

Even though the price is not one of the criteria, it needs to be noted that using Google Maps is for free just if you meet the limit of API calls per month. After the limit is exceeded, you need to pay for it. The pricing and documentation of Maps JavaScript API are available on the official website [22]. Google Maps offer multiple map controls such as zoom in/out, scale, and various map types (roadmap, satellite, terrain, hybrid). It is also possible to design the style of the map on the Google Cloud Platform. The Google Maps support custom markers in .png format and custom layers, such as info popup window, which is displayed after clicking on the marker, also GeoJson [9] data layers, heatmaps with extended Visualization library, and KML [10] layers. The platform also offers various map objects (markers, shapes, symbols, etc.), configurable events, geocoding, and also provides a search service with an integrated Places library supporting auto-complete for

[7] **Data layers** can be understood as objects on the map that are tied to latitude/longitude coordinates, so they move when you drag or zoom the map.

[8] **Geocoding** is the process of transforming a description of a location, such as an address, or a name of a place, to a pair of coordinates, and otherwise.

[9] **GeoJSON** is a standardized format for encoding a variety of geographic data structures. It defines several types of JSON (JavaScript Object Notation) objects and how they are combined to represent data about geographic features, their properties, and their spatial extents [23]

[10] **Keyhole Markup Language** is GIS (Geographic Information System) file format extended from XML (Extensible Markup Language) notation for expressing geographic annotation and visualization within two-dimensional maps and three-dimensional Earth browsers.

addresses. Google maps have also integrated the MarkerClusterer library for clustering markers (grid-based). However, it is also possible to use a different type of clustering using external plugins that are usually adapted to all well-known map platforms.

Criterion number	**Meet the criterion**
1	yes
2	yes
3	yes
4	yes
5	yes

Table 2.1: Google Maps - meeting criteria table

As can be seen in table 2.1, all of the criteria are fulfilled. Also, other exciting features are offered by this map platform.

Here Maps

Here Maps platform is also for free just for the limited API requests per month. The number of these transactions is sufficient for Here Maps as well as for Google Maps. Map controls such as zoom in/out, scale, and map types (standard, satellite, terrain) are also provided. Here Maps platform also supports a possibility to customize markers and gives more opportunities for that, .svg, .png files, and URL address. Density-based clustering is integrated with Here Maps. Similarly, the Here Maps platform allows the addition of custom data layers in GeoJSON format or KML but only for 2D. The configurable events are allowed, and geocoding and search API with autosuggest for addresses are supported. This map platform also fulfills all of the criteria (see table 2.2).

Criterion number	**Meet the criterion**
1	yes
2	yes
3	yes
4	yes
5	yes

Table 2.2: Here Maps - meeting criteria table

Open Street Maps

Like the last map platform, Open Street Maps are going to be introduced. As the name of the platform suggests, it is open-source. Open Street Maps offers the same map controls as the previous platforms with map types - grayscale, streets. Customizing the markers is available and adding GeoJSON data layers is also possible. The map events are predefined, similarly to platforms described above, but without a plugin for event handling is not possible to customize them. Also, other features as marker clustering, geocoding, search, and auto-complete are possible only with plugins. The libraries such as OpenLayers, Leaflet, or Mapbox are commonly used with the Open Street Maps platform for providing a wide range of features.

Criterion number	Meet the criterion
1	yes
2	yes
3	with plugin
4	yes
5	with plugin

Table 2.3: Open Street Maps - meeting criteria table

Open Street Maps platform also fulfills the criteria, but some features are possible only with third-party plugins. The use of extended libraries is not a problem as long as they enable the features you need.

Platform choice

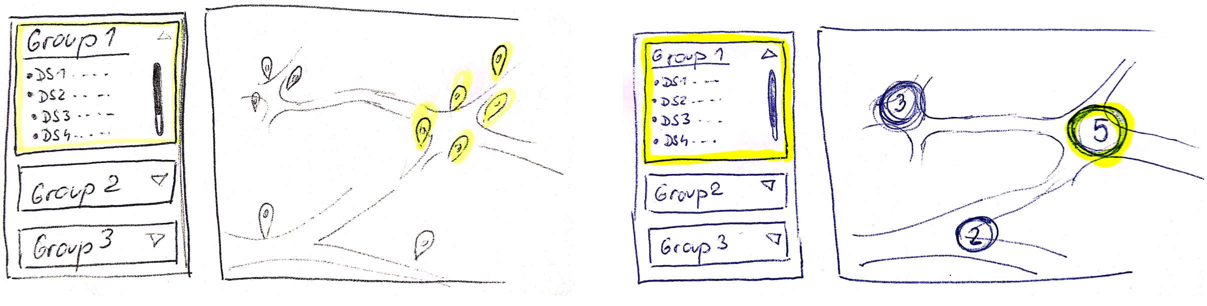
Even though all map platforms fulfilled the criteria and are suitable for the purposes of this thesis, I need to consider that more requirements and features need to be satisfied for future development. Therefore, I chose Google Maps because they provide the broadest range of features from all platforms, even though it is not an open-source solution. This decision was also encouraged by the fact that Google Maps are widely used and familiar to most users. It is also worth mentioning that Google Maps provide the best services with regards to the searching capabilities for certain places. This is highly important for the *locate task* type when locating the specific address, motorway, etc., on the map. The only disadvantage is that Google Maps does not provide the extension for density-based clustering. Thus, there will be a need to use another library combined with Google Maps that support density-based clustering. This library will be Leaflet, and it will be described in detail in Chapter 4.

Chapter 3

Design

In this chapter, I describe the process of designing the prototype. I will focus on tasks that need to be achieved by users. Then I introduce examples of two clustering methods considered for the prototype and choose one of them. I will continue with summarizing the process of designing map markers for the individual data source and clusters based on the principles from Section 2.3.1. Afterward, I present the possible solution for the challenges such as data sources without given information about the location and check if I meet the requirements of the proposed personas from Section 2.1.3. In the end, I offer possible extensions for the Video Insight platform that would be feasible thanks to geographical information but is beyond the scope of this thesis.

Collecting requirements in Section 2.1.2 helped me realize what users need and what should be designed. As mentioned earlier, I need to design a system for visualizing and managing time-varying geographical data. Detailed information about the data and attributes is in Section 2.1.5.



(a) Individual markers

(b) Clusters

Figure 3.1: Highlighted group of data sources on one intersection

In the first stage of design, I needed to consider the structure of the data. A single element, a data source, needs to be visualized on the map. Moreover, it needs to provide other information about itself, such as device type, state, name, traffic volume, and traffic counts. That is one layer of the data. If multiple data sources form a group of data sources, then there is a second structure layer. For this situation, I was thinking about what would be a suitable solution when visualizing multiple groups of data sources together. One way to do it is shown in Figure 3.1a, where there are 3 groups and single data sources are visualized as markers. The data sources that are part of a chosen group are highlighted on the map. Another way to visualize groups of data sources together would be via clusters, as shown in Figure 3.1b. Every cluster represents the group of data sources. The question is, what will happen if the zoom level changes? Will the clusters of the groups become one/divide? If the zooming in will split the clusters, the user can be confused about which traffic sensor marker belongs to which group. Also, if the user zooms out, the markers or clusters should be grouped together. However, what if the markers from two different groups become one cluster? Then the cluster distribution probably will not make sense. Moreover, what if there is a third layer (project), as I mentioned while describing the future version of the Video Insight platform in Section 2.2.2?

To answer these questions, I was forced to think about how users want to work with the data. Do they need to visualize all the data sources together? From the user research, I know that users need to analyze traffic data across multiple data sources from one location, e.g., intersection. It means that there is no need to visualize the individual data sources from different groups together. So in case that there are multiple layers in the application structure, it is not necessary to visualize an individual data source. Instead of visualizing one particular data source, it would be essential to visualize the whole group as one icon with recalculated coordinates to center. In that case, the map will be used just for an overview of the group of traffic sensors. Let's focus on the case where there is just one layer that consists of many data sources in one group. We need to think about how the data sources should be visualized. As I showed in Section 2.3, when visualizing such data on the map, there is a need for clustering, so the map is easy to orient in. The clustering method, density-based technique, was chosen in Section 2.3.3 when compared with the grid-based technique.

In the following section, I will describe the process of design. Summarize the functionalities that should be part of the solution, and offer a design of a low-fidelity prototype that will be implemented.

3.1 System components

In Section 2.1.4, I identified two main goals that the designed system should satisfy. The first is to find specific traffic sensors easily, and the second to get traffic data from specific locations and times. While designing, I will proceed with Shneiderman's visualization mantra [24]: **Overview first, zoom and filter, then details-on-demand**. The overview should summarize the essential information of the data without getting into the minor details and reduce data clutter. It is obvious that using clustering was the right choice to provide an overview about where the data sources are located without overwhelming the users. These clusters should also indicate the number of objects that they cover. How to achieve that will be described in detail in Section 3.2. In that section, I will also describe the visualization of categorical attributes **device type** and **state** of the data source, as well as ordered attribute **traffic volume**.

Let's focus on zoom and filter. Zooming allows users to navigate to the requested area. In the application context, zooming out will extend and merge the clusters of data sources, and zooming in will decay the clusters. On each zoom level, changes should be observable. With filtering, users should be able to visualize and work only with data that they are interested in. The data attributes that users could filter on are **device type** (fixed camera, drone), **state** (empty, processing, processed, archived, error) and **traffic volume** (high, medium, low). Enabling users to set what should be visualized based on these variables would be instructive for them. Similar action to filtering is searching. Users should be able to search for specific data source by **name** attribute, or search for specific **coordinates**, location, city, motorway, street, etc.

The last part of Shneiderman's mantra is details-on-demand. It is essential to show detailed information about the specific data source to the user when requested. Possible information that the users are interested in is usually traffic data. In this thesis, it will be traffic counts data. However, from the view of the Video Insight platform, it could be various traffic data, such as traffic trajectories, object distribution, etc. In other words, all traffic data of the chosen traffic sensor that can be obtained from *Analyze page* of the platform, described in Section 2.2.2. The traffic counts data used for this thesis are time-varying. Therefore the users need to be able to filter the data based on the specific time. A timeline representing traffic counts with a slider that filters time seems to be a solution to enable that feature.

It is also worth mentioning that a correctly developed web application should meet CRUD [1]. Therefore the application should allow users to create a new data source, edit it and delete it as well.

Before I present the design, I will summarize what features should be included in the final prototype.

- Zoom level clustering
- Filtering based on device type
- Filtering based on the state of the data source
- Filtering based on traffic volume
- Searching by data source name
- Searching locations on the map
- Showing traffic data on demand
- Adding a new data source
- Editing existing data source
- Deleting existing data source

When visualizing geographical data, there is an obvious need to represent them on the map. The map will be one of the present components from the overview level. Nevertheless, for better organization and orientation on the page, a list of all data sources would be essential. Therefore, the page should be composed of a data sources list as well. Firstly, I was thinking about a hidden list that appears on demand (after clicking on the tab), as can be seen in Figure 3.2. As showed up later during the design process, there was

[1] **CRUD** stands for Create, Retrieve, Update, Delete, which is a fundamental part of any database or storage based program.

no need for hiding the list of the data sources, as the map was big enough, and click to show the data sources list was just a redundant step for the users as they needed to see it almost all the time. I decided to place the list component on the right side next to the map component. It is shown in Figure 3.3 that displays the final prototype.

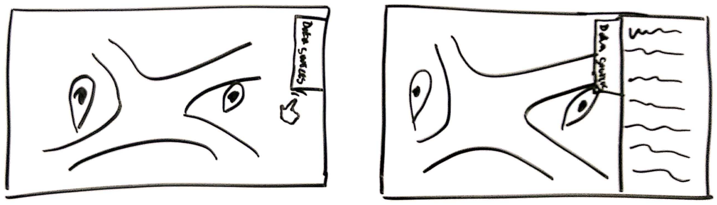


Figure 3.2: Overview level composed from two components - map + hidden list

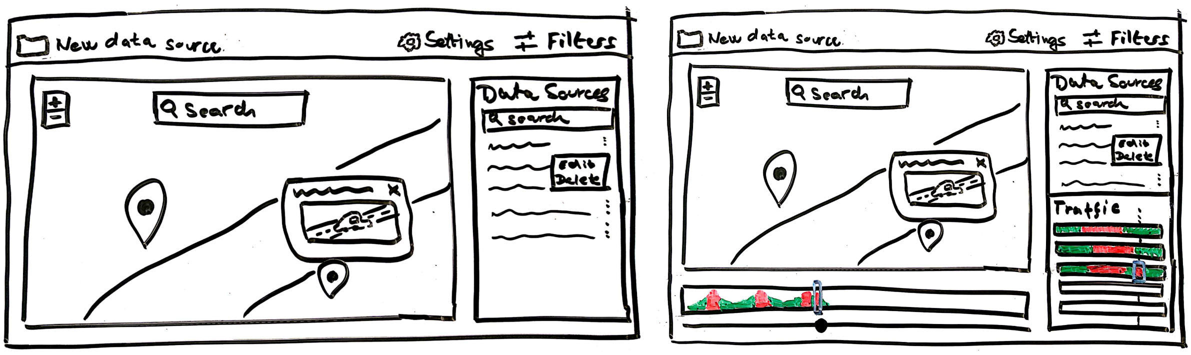
Then there was a need to allow various filtering. It should not interfere with the overview layout much, but still, it should be visible enough. Therefore I decided to use a roll-out button that allows users to set the filters they need. The principle is similar to the tab described earlier. After clicking on the button, filters appear or disappear.

Another action is searching. I defined two types of searching before, searching by name and by location. One is connected to the data sources list and the other to the map. For that reason, I designed two search boxes. One, map-related, will be powered by Google Search, and the other will filter data sources by name.

Before I start explaining how the traffic data, in this case, traffic counts, will be visualized, I clarify which data sources it will include. When talking about traffic data, it is needed to be said that they are available only for data sources whose footage is already processed because only for these data sources are traffic data extracted from the footage and accessible. As a result of that, I decided to create a new mode suitable for data sources that are processed and provide information about the traffic. The users will have an option to select which mode they want to see the map in. An **Overview** or **Traffic** mode. Traffic mode should allow users to filter traffic data by time and detect traffic counts in certain moments. It should be clear how many vehicles passed by in chosen time. It should also be easy for the users to recognize the traffic patterns over time or observe anomalies in the traffic data. The system should also permit to display traffic across all data sources (average traffic counts) or choose individual data source/s.

To better detect traffic patterns, I decided to map traffic counts in the timeline to colors of traffic volume in a specific time to signal users how the traffic is congested. Traffic volume colors will be described later in Section 3.2. Also, there will be two types of timelines. One for visualizing traffic counts and identifying the patterns over the chosen time period (week), shown at the bottom of Figure 3.3b, and another that shows traffic volumes for every hour in a day. Days will be sorted in rows, so the users can observe the similar patterns related to a specific hour as it is shown on the bottom-right in Figure 3.3b.

All of the features mentioned above can be seen in Figure 3.3.



(a) Overview mode

(b) Traffic mode

Figure 3.3: Final prototype design

3.2 Process of designing map markers

This section focuses on categorical attributes of the data source, device type, and state. As mentioned in Section 2.3.1, the most efficient way to visualize categorical attributes is by mapping them on a suitable visual channel.

First, let's concentrate on device types. Two types of devices need to be visualized, fixed camera and drone. When I was thinking about the most appropriate channel for mapping this attribute, the shape seemed like the correct solution. To be more specific, an icon representing the device seemed to be the best choice, as it is very intuitive and cannot be mixed up. The camera and drone icons can be seen in Figure 3.4.

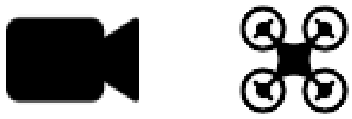
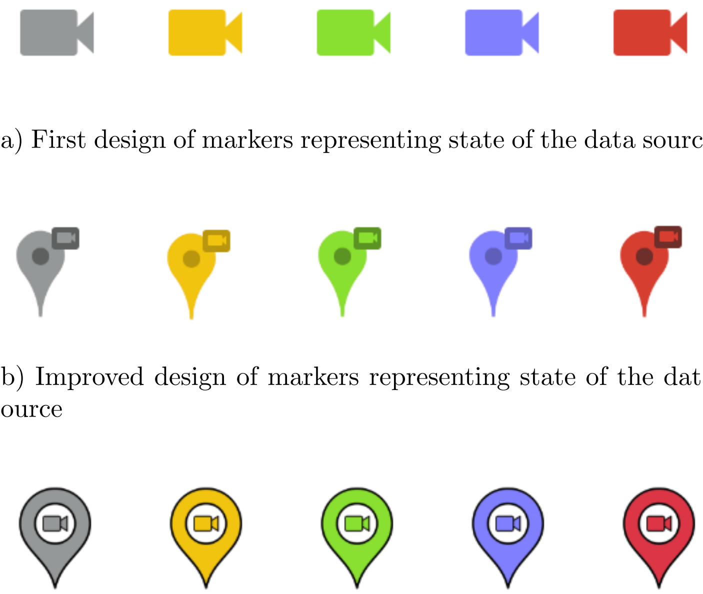


Figure 3.4: Icons representing device type

The following attribute, state, also needs to be mapped to a visual channel. There are 5 possible data sources states - empty, processing, processed, archived, and error. The shape, color, and texture are the channels that would fit this attribute. Other channels are not very relevant in this case. The shape is also not so suitable, as the state represents an abstract condition, not to mention that shape is already used for the device type. The texture would also be an option, but it does not have to be necessarily easy for the user to distinguish between the textures. The color channel would be ideal. In the Video Insight platform, there are already used colors for every state and are straightforward. Error is represented by red, archived by purple, processed data source by green and data source in processing state by yellow. The only state that does not match the original color is an *empty* state. The color representing the empty state in the application is white, but white is not well recognizable on the map. Therefore I changed it to gray. The used color shades can be seen in Figure 3.5.



(a) First design of markers representing state of the data source

(b) Improved design of markers representing state of the data source

(c) Final design of markers representing state of the data source

Figure 3.5: Process of designing markers (only markers with device type *fixed camera* are present)

When I clarified what channels I wanted to map the attributes to, I started designing the markers. The first idea was to visualize just icons of device type with the corresponding color to keep it simple, as shown in Figure 3.5a. That turned out as a bad solution. Even though it is simple, using these markers did not provide clear information about the geographical position. Therefore, I decided to use a marker with a sharp bottom edge, similar to a Google Map marker. I mapped the marker to the matching color and added a device-type icon as a stamp to the marker. The problem with visualizing the precise coordinates was solved, but another complication occurred. The device type was too small and not very recognizable, as can be seen in Figure 3.5b. I continued designing and created the final version of the markers shown in Figure 3.5c. The device icon is now bigger than in the previous version, and it has a border same as a whole marker which helps to greater visibility on the map.

Not only categorical attributes need to be mapped to the visual channel. Traffic volume, which is ordered ordinal attribute, in this case *high*, *medium*, *low*, also needs to be reflected in the map. To understand how to visualize the data, it needs to be realized that attribute traffic volume exists only for data sources whose footage is already processed. Otherwise, traffic data are not available. As mentioned at the beginning of this chapter, there will be two modes: an overview of all data sources and the other for traffic overview of processed data sources. These facts allow us to map traffic volume to color as the state of the data source does not have to be visualized in traffic mode. When searching for applicable colors to represent traffic, I get inspired by a red, yellow, green color palette widespread across the traffic industry. It represents traffic lights, but it is also frequently used to represent the traffic volume, congestion, or speed. The designed icons are shown in Figure 3.6.

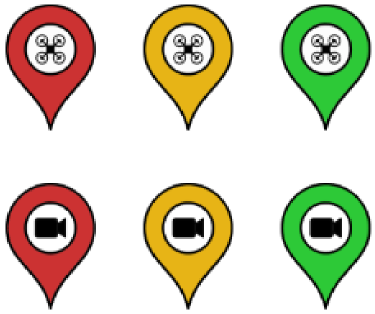
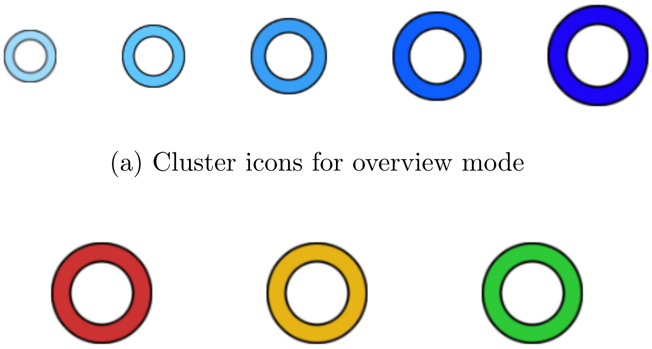


Figure 3.6: Final design of markers representing traffic volume

After designing markers representing data sources, I needed to think about clusters and what information their visualization should provide. The information that clusters provided is how many data sources are grouped in the cluster. That is the variable that I wanted to map. In overview mode, I decided to map this variable to 2 visual channels to enhance this information. The channels I used are size and value. It is also worth mentioning that each cluster also contains the number of the grouped data sources in the middle of the cluster icon. I used a blue color palette for overview clusters as the blue color does not symbolize anything else in the application. Light blue represents the smallest cluster, and with an increasing number of data sources in the cluster, the cluster icon gets bigger and darker, as depicted in Figure 3.7a.



(a) Cluster icons for overview mode

(b) Cluster icons for traffic mode

Figure 3.7: Designed cluster icons (without the number of grouped data sources)

In traffic mode, I decided that it is important to propagate the traffic information into the cluster. Therefore, the average traffic volume from all of the data sources in the cluster is mapped to the cluster icon color. The color channel is used to provide traffic volume in the area. The size channel represents how many objects are in the cluster. The cluster icon is also composed of the precise number of grouped data sources in the middle of the icon. The cluster icons without the numbers are shown in Figure 3.7b.

3.3 Challenges

To be sure that the design fulfills the user requirements, let's review our personas. John needed to filter data by time, which is enabled thanks to traffic mode and filtering. On the other hand, Susan needs to work with a large number of traffic sensors and cope with the traffic data they produce. The data representation on the map is done by clusters which simplify data sources presentation on the map. The design also allows Susan to introduce traffic volume and counts to the client across all or filtered locations. The last persona, Kevin, brings the big issue to the design. How to visualize traffic sensors without location information together with sensors that provide this information? The Video Insight platform offers two views of data sources, grid and table view. My design should be a third view for the data sources used in the future. Therefore one of the possible solutions for this is to encourage the user to use other views in the situation when the position of the sensor is not provided, as can be seen in a top-right sketch in Figure 3.8. There is also a possibility not to offer the map view when there is no data source with provided geographical coordinates. Nevertheless, what about the situation when there are some data sources with location and some of them without it? As design not only visualizes the data sources on the map but also provides a list of them in the right panel, it provides the possibility to list all the data sources, even those without the location, and mark them specifically, so the user is aware of this fact. After clicking on the specific data source in the list, the map should navigate to the specific marker on the map. In case the data source does not have coordinates, the dialog window should appear and inform about this. It is sketched in the bottom parts of Figure 3.8.

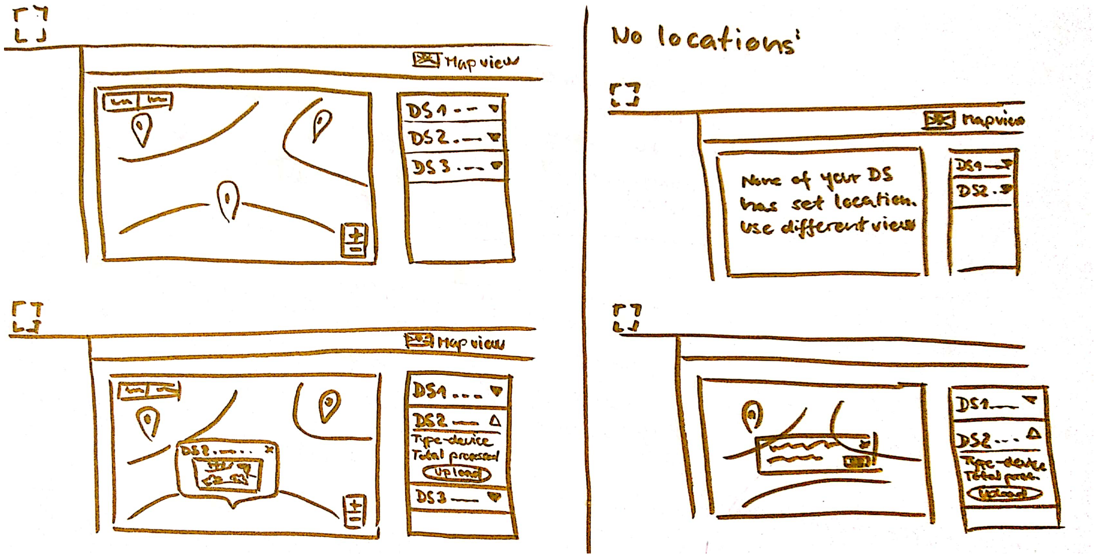
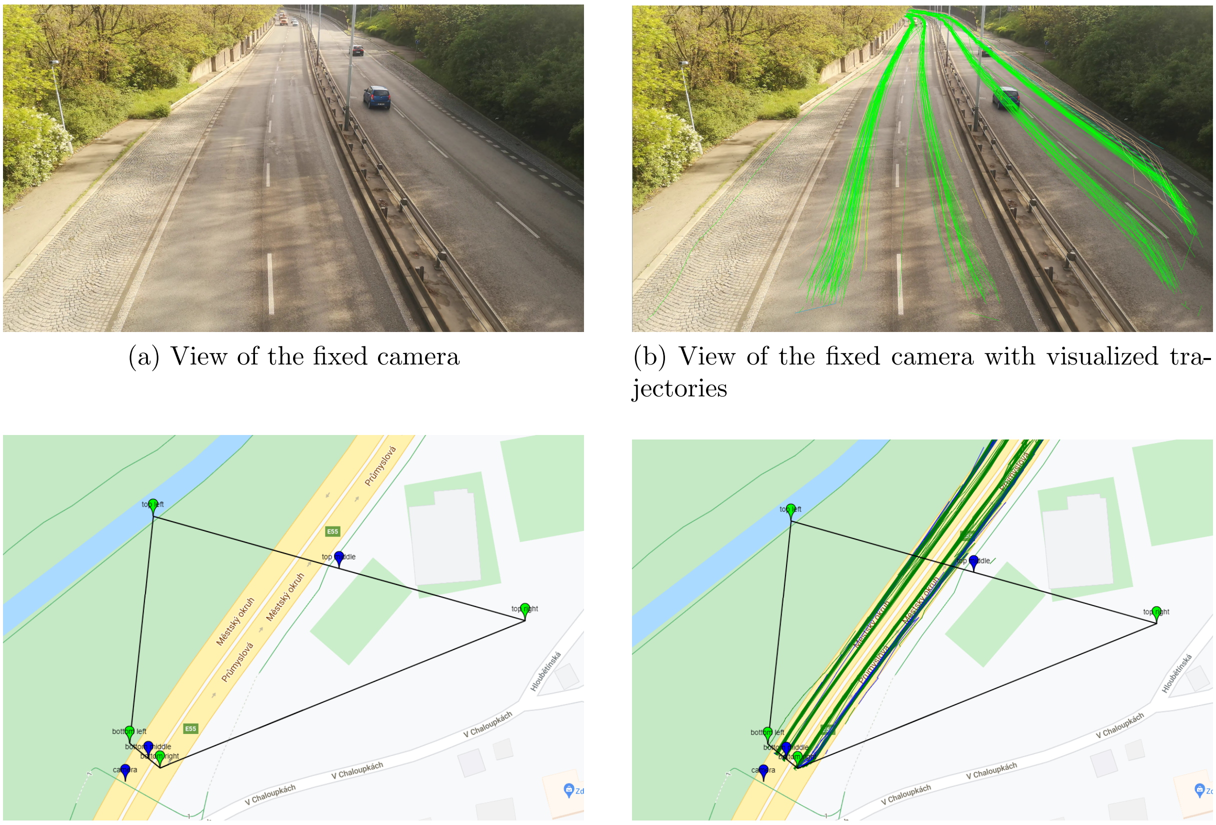


Figure 3.8: Comparison between views with and without provided geo-information of data source

Regardless, Susan's requirements are satisfied, it needs to be said that visualizing the traffic data across the multiple traffic sensors, in this case, traffic counts, represent average traffic counts across selected data sources. Therefore it will not be possible to use these traffic counts for O/D analysis to give her information on how many vehicles exit one traffic sensor and enter another. For enabling such a feature, there is a need to match traffic trajectories of vehicles across multiple data sources. This re-identification of vehicles is a challenging task. One of the possible solutions to doing it is with Automatic Number Plate Recognition(ANPR) which can match a vehicle from one footage to another and track it based on the number plate. It is only possible if the footage has a high resolution and plates are highly visible on the video recording or if the camera supports ANPR.

3.4 Possible expansion of Video Insight

In this section, I present the ideas I observed during the research that, thanks to provided location coordinates, could help to expand the Video Insight platform for traffic analysis but are not the goal of this thesis. As a result of bringing geographical information to the context, interesting computational and visualization tasks can be performed. I introduce just a few of them and describe the process of further application development superficially.



(a) View of the fixed camera

(b) View of the fixed camera with visualized trajectories

(c) Calibrated camera view transformed to the map

(d) Calibrated trajectories transformed to the map

Figure 3.9: Process of geo-mapping

First of all, what geoinformation offers is geo-mapping in a sense to map the traffic data coordinates obtained from video footage into the map coordinates. The way how the extracting traffic data from footage works now can be seen in Figure 3.9a and 3.9b. The video recording is processed, and from the camera view, the trajectories of the vehicles are extracted and subsequently visualized on the scene as shown in Figure 3.9b. Then the user can describe this scene with lines, zones, and movements on the *Describe page* as mentioned in Section 2.2.2. Placing the object into the camera view can be often tricky as the perspective distorts reality. Then the wrong position of the objects on the scene can cause undesirable results. It would be much easier and intuitive for the users to describe the scene directly on the map. However, the process of changing the scene preview into the map is not easy.

It is primarily challenging because it needs to be done manually. To map the scene to the map, at least 4 points need to be provided from both the scene and the map. The more points are provided, the more accurate the calibration of the points can get. Obtaining the result, as shown in Figure 3.9, was preceded by the calibration process. For this process, the latitude and longitude coordinates of the specific points in the map, and the x, y coordinates of these points in the picture, need to be provided. The process of mapping the points for the calibration is shown in Figure 3.10. The mapping takes place in the simple tool I created for the internal use of GoodVision. There are six red markers placed on the map and the scene based on which the calibration was performed later. The blue marker shown in the map represents the position of the camera. It is not part of the tool. It was added just for the demonstration.

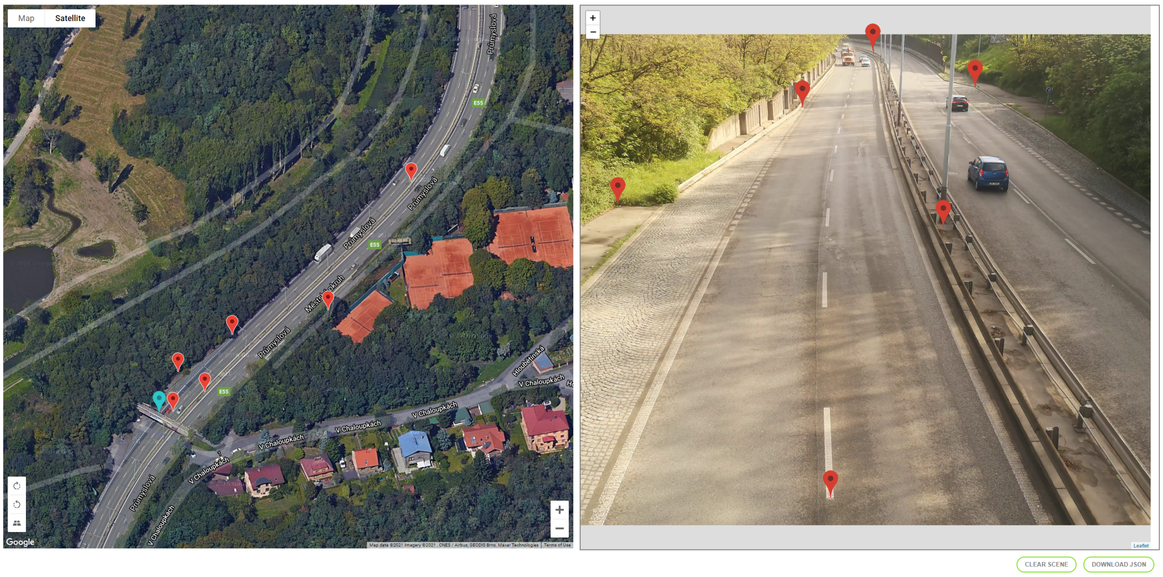


Figure 3.10: Example of mapping points needed for calibration via internal tool

It would be ideal if the users were able to submit the mapping process for their traffic sensors. Then the rest of the process could be done automatically. The problem is that even a small shift of the marker can cause a big change in the final calibration, which can influence the accuracy of the coordinates of the traffic data.

The calibration of the scene and the map being done properly can improve how the speed of the vehicles is calculated at the moment. If users need to know the speed of the vehicles, they need to provide the distance between the single objects in the scene. It can cause troubles for the users if they are not aware of the exact distance. This calibration process could be computed automatically, as the distances between the objects could be calculated from the geographical positions.

When the functionalities mentioned above will be automatized, then the other opportunities can be considered. As I mentioned earlier in this section, describing the scene on the map is easier and more natural than doing it on video footage preview. It could also provide better results if the map is well calibrated. Then the visualization of the vehicles' trajectories could be done differently. For example, not to visualize the individual trajectories like it is done now in the camera preview but group the trajectories of the same object classes (car, truck, but, etc.) and visualize them into one visual channel with respect to the direction of the trajectories (similar to the sketch in Figure 3.11). Also, the traffic counts from O/D analysis could be visualized directly into the map or with the help of a Sankey diagram that preserves the orientation of the roads (similar to one shown in Figure 2.9).

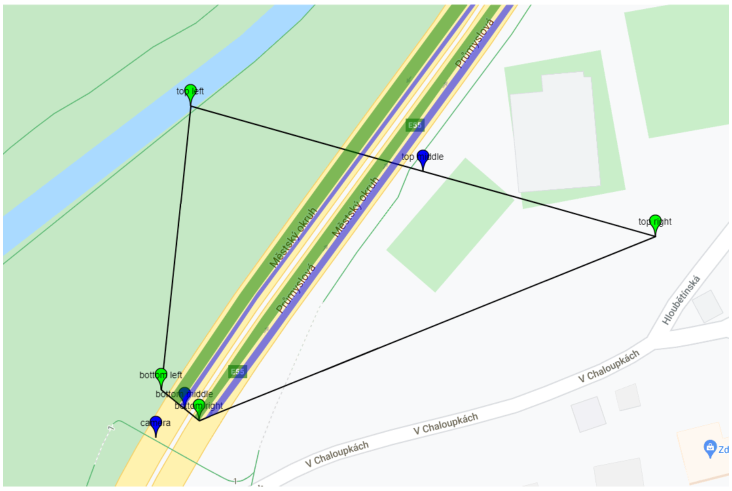


Figure 3.11: Possible visualization of trajectories. Green trajectories represents cars and blue trajectories represents other vehicles.

The O/D analysis is very common and needed in the traffic industry. The current tool provides O/D analysis within one data source. It can be handy when the traffic engineers need to analyze one intersection, roundabout, or different traffic scene. Unfortunately, this feature is limited by traffic data from one data source. Describing the scene across multiple data sources would be a nice and definitely beneficial feature.

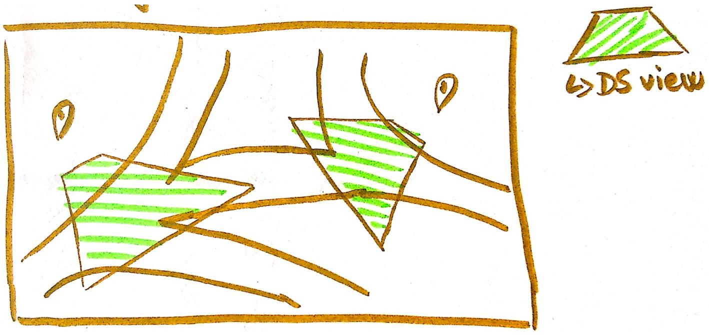


Figure 3.12: Views of two data sources visualized on the map

Figure 3.12 illustrates how the user could recognize where s/he can place the scene objects (lines, zones) to describe the scene. To match the traffic data from different data sources is not an easy task, as mentioned at the end of previous Section 3.3.

Chapter 4

Implementation

In this chapter, I will describe the environment I developed the high-fidelity prototype in. After introducing the environment, I will continue presenting the packages and external libraries applied to the prototype. Then I specify the data used in the prototype for usability testing purposes. Eventually, I will describe the application's structure from the user point of view, introduce the application modes and what functionalities the prototype offers, and finally, I will give a short overview of the repository structure.

4.1 Development environment

This section will describe the environment I implemented the high-fidelity prototype of a client-side web application in. For local development of the web application **Node.js** [25] environment is used and installed via node package manager (**npm**) [26] which is being used for publishing JavaScript-based libraries, applications and packages. I used Node.js version 15.6.0 and version 6.14.11 of npm.

The web application itself is built by the latest version of**Angular** [27] framework - Angular 11. Angular is a component-based framework for building scalable web applications with well-integrated libraries that cover a wide variety of features. It is build on **HTML**, **SCSS** and **TypeScript**. TypeScript is a layer on top of JavaScript language enriched with static typing. HTML is a shortcut for HyperText Markup Language, which is not a programming language but is used for defining the layout of the user interface. SCSS, Sassy Cascading Style Sheet language, is then used for describing the presentation of HTML elements. In contrast with basic CSS, SCSS offers various advantages such as using variables, defining mixins, nesting styles of the components, doing math operations, etc.

For developing the application, I chose integrated development environment (IDE) **Visual Studio Code** [28] which is built on open source. The IDE supports multiple languages and comes with built-in support for TypeScript and Node.js. It also provides numerous plugins and extensions, which makes developing much more effortless. It has integrated commands for version system **Git**. For running the application locally, I used Google Chrome browser, version 90. However, the application should be compatible also with other browsers, Firefox, version 83, Microsoft Edge, version 90, Opera, version 75, and Safari, version 14. The application in Internet Explorer, version 11, is not entirely consistent as a slider for traffic counts is not working and looking as expected.

4.2 Third-party libraries

After summarizing the environment used for development, I will present the third-party libraries used in the high-fidelity prototype. I used several **Angular packages** that helped me with basic functionalities and project structure. For dynamic reacting on asynchronous actions, iterating, composing, filtering streams, or mapping objects to different types, I installed **rxjs** package. For enabling time interval set up for visualizing traffic data, I used **ng-pick-datetime** package. Finally, as I mentioned in Chapter 2, **Google Maps JavaScript API** [22], **Leaflet** [29] and **Leaflet.markercluster** [30] library were used for the implementation. To use Google maps as base maps of the Leaflet plugin, I needed to include one more package **GoogleMutant** [31]. All the packages mentioned above were installed via npm.

4.2.1 Google Maps JavaScript API

As Google Maps JavaScript API is not open-source, there is a need to registering and creating own API key. This key is then provided in the index.html file and used for every API call. There is also possible to provide Google Maps libraries such as Places API, which is used for locating the position on the map. I have created the API key for the development purposes of this thesis, but in the submitted git repository, the key will not be provided. Therefore the search box and map tiles are not going to work correctly without it.

4.2.2 Leaflet library

Even though I decided to use Google Maps as a map platform, I showed in Section 2.3.3 that the most suitable clustering is density-based clustering. This clustering option is not supported by Google Maps API. Therefore, I used the Leaflet.markercluster library, which maintains it. The Leaflet.markercluster plugin is based on the Leaflet library, the leading open-source library with all the needed mapping functions. Even though it implements also search and geocoding, it is not of such high quality as the ones provided by Google Maps. This is the reason why I decided to combine these two plugins to provide an outstanding user experience. Leaflet enables to load map tiles from different map platforms. Unfortunately, it is not possible for Google Maps because their terms of service do not allow to use of undocumented API, e.g., loading map tiles which is essential while integrating the map platform into Leaflet. Therefore, to combine these two libraries, I needed to use the GoogleMutant plugin that legally uses Google Maps API. All of the extended plugins on Leaflet are also open-source.

4.3 Data set applied for prototype

I already mentioned what type of data needs to be visualized in Section 2.1.5. Now I will describe the mocked data set I used while implementing the prototype and later for usability testing. In the first place, I needed coordinates information from at least 100 traffic sensors. The first data set I was trying to visualize consisted of web cameras [1] placed all over Europe. The problem was that they were not traffic cameras and did

[1] https://api.windy.com/api/webcams

not provide an authentic image. Therefore I decided to use different data set to give me coordinates of traffic sensors.

I discovered an open-source API [2] that offers data from 135 traffic cameras. The problem was that the API did not send the response with allowing CORS (cross-origin resource sharing), which is essential for locally running the application. Therefore I decided to call the API via chrome extension API Tester and store the response in one of the .ts files as constant directly in the project. The data set provides information such as coordinates of the traffic camera, the title of the camera, preview image, and direction of the view (north, west, etc.). I consider the information about the direction as interesting information that could be useful for visualizing the camera's view on the map that could be part of the future extensions of the application.

Even though 135 cameras are enough to fulfill the thesis assignment, I decided to use also bigger data set for following usability testing to present the power of the clustering to the users. I used coordinates data [3] from 1958 traffic sensors distributed across England.

The coordinates and name attributes are acquired from the data sets mentioned above. There is still a need to gain the other attributes. All of the sensors from data sets are cameras, but I need the other device types for the system, so I decided to randomly classify the data sources into cameras and drones. The same approach was used for the data source's state attribute, where I randomly generate the value from a set of states (empty, processing, processed, archived, error).

The last attributes that need to be obtained are traffic data, traffic counts, and volume. I could randomly allocate the traffic counts, but then there would not be visible any real patterns in the traffic. Therefore, I decided to use traffic data from the GoodVision database to demonstrate traffic counts to provide a more realistic experience during usability testing. It needs to be said that the counts or volumes do not match the real traffic sensor location. Data were provided from various locations and dates and randomly spread across the mocked data sources in a processed state. The traffic counts are not from a specific date chosen in the application, but the day in the application corresponds to the day in a week that the data were collected. Traffic volume levels were subsequently calculated from traffic counts.

Examples of mocked data set with all attributes mentioned earlier can be seen in every Figure in this chapter.

4.4 Structure of the application

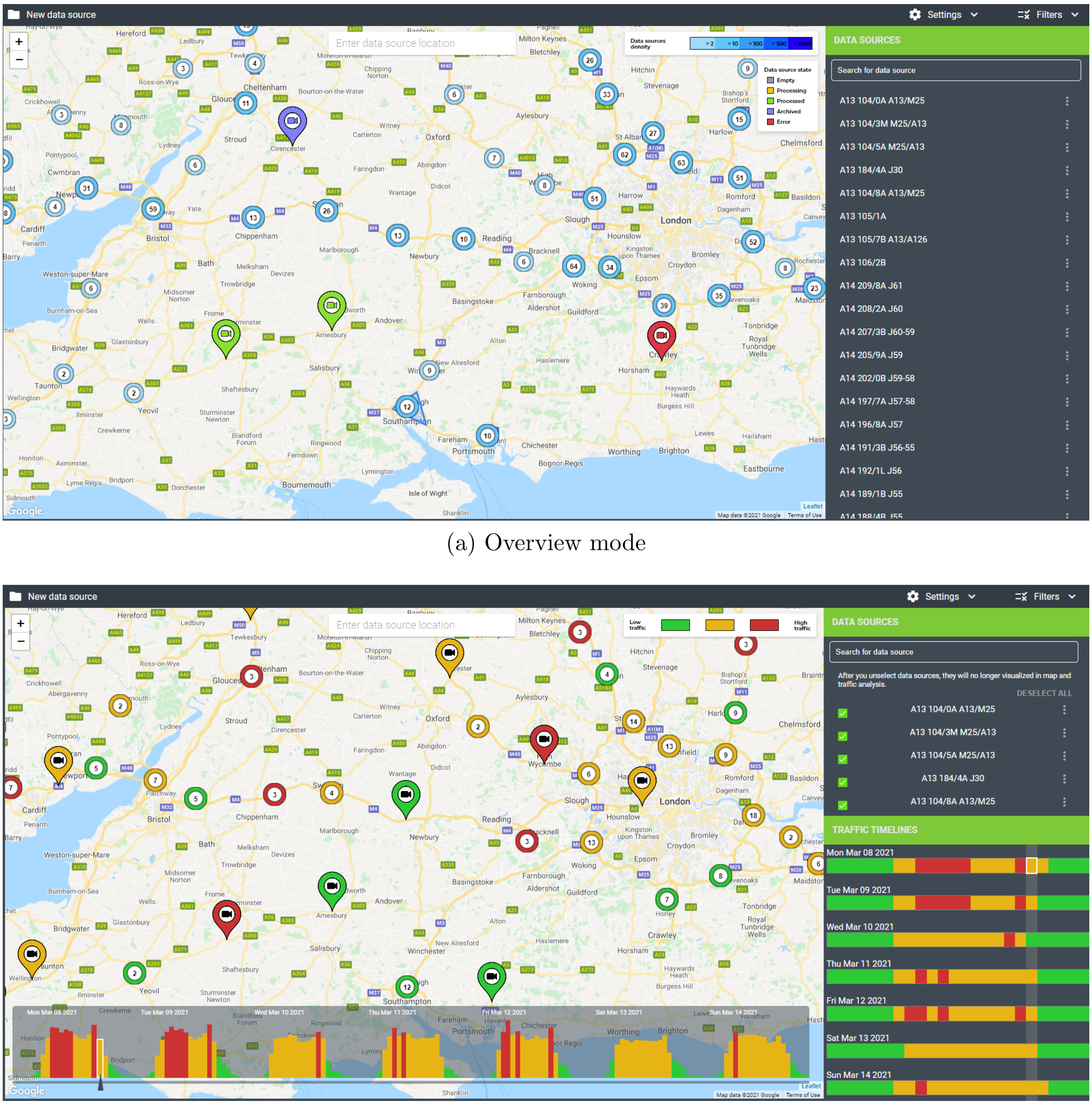
This section will introduce the implemented high-fidelity prototype, describe how it works and what functionalities it offers. At the end of this section, I will quickly present the repository structure of the source code.

Firstly I present the prototype's layout. It consists of three main components, the top

[2] https://api.transport.nsw.gov.au

[3] https://trafficcameras.uk

bar with buttons, the map view, and the list of data sources placed on the right. The layout can be seen in Figure 4.1.

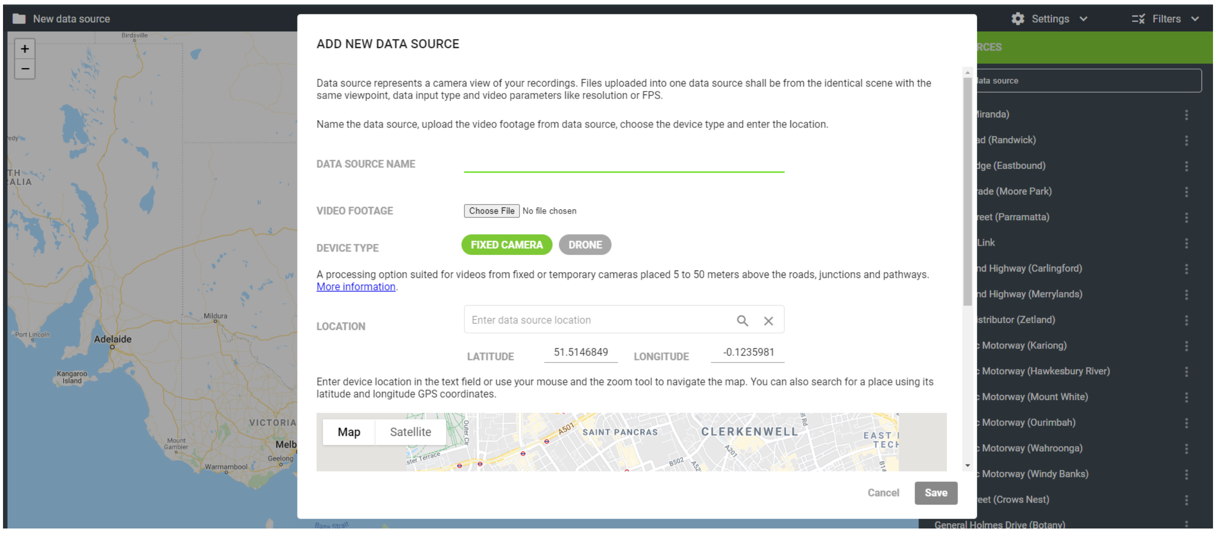


(a) Overview mode

(b) Traffic mode

Figure 4.1: Implemented high-fidelity prototype

The top bar contains three buttons. The first button, *New data source*, placed in the left corner, allows the users to add a new data source to the platform. After clicking on the button, the dialog window opens, shown in Figure 4.2, and the user can provide the information about the data source. S/he needs to fill out the name of the data source, upload the video footage, choose device types and provide the coordinates of where the data source is located. The second button, *Settings*, offers to choose between two modes of the prototype by rolling out the menu with options, Overview mode and Traffic mode. The modes will be described in detail later. The last button *Filters* enables the user to filter across the data sources based on their attributes - device type, data source's state, or traffic volume. It displays the menu where it is possible to select one or multiple filtering options.



ADD NEW DATA SOURCE

Data source represents a camera view of your recordings. Files uploaded into one data source shall be from the identical scene with the same viewpoint, data input type and video parameters like resolution or FPS.

Name the data source, upload the video footage from data source, choose the device type and enter the location.

DATA SOURCE NAME

VIDEO FOOTAGE No file chosen

DEVICE TYPE

A processing option suited for videos from fixed or temporary cameras placed 5 to 50 meters above the roads, junctions and pathways. More information.

LOCATION Enter data source location

LATITUDE 51.5146849 LONGITUDE -0.1235981

Enter device location in the text field or use your mouse and the zoom tool to navigate the map. You can also search for a place using its latitude and longitude GPS coordinates.

Figure 4.2: Dialog window for adding new data source into the platform

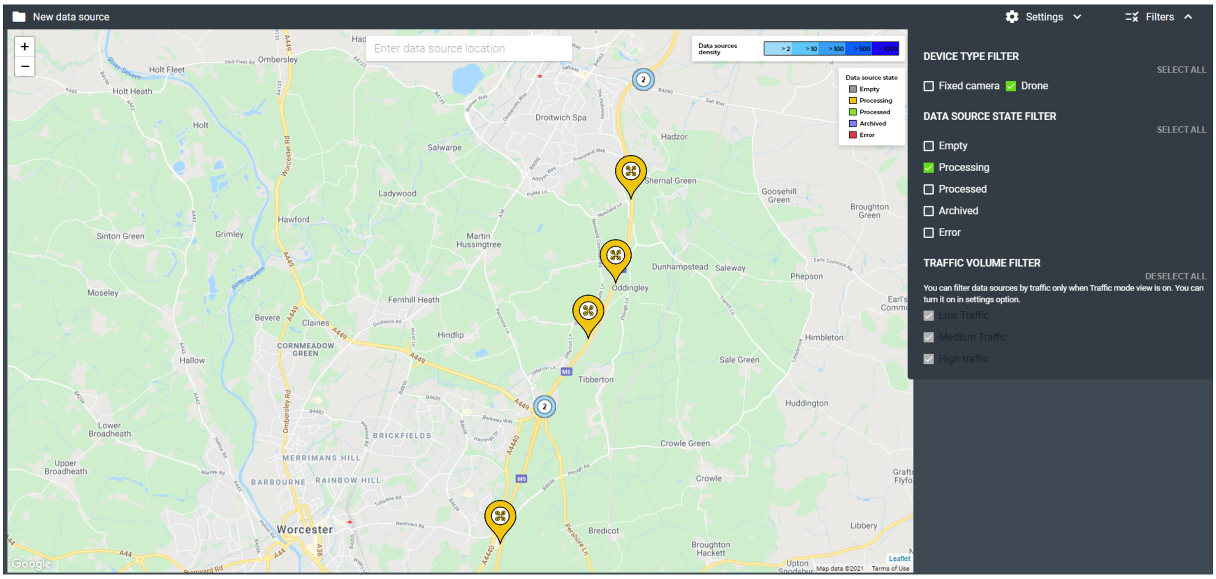
The map view is composed of map background provided by Google Maps, map controls for zooming in and out, which are placed in the top left corner of the map, search box placed in the top center of the map also powered by Google Maps, and legend placed in the top right corner of the map. The search box enables searching for a specific location and also auto-suggests the places. The legend varies on the selected mode of the application, and it is placed in the top-left corner. On the map are also visualized markers representing data sources and cluster markers with the number that represents the number of grouped data sources.

The last component, the data sources' list placed next to the map on the right, contains a search box and a list of all or filtered data sources. The search box enables to filter of data sources based on their name attribute. The list of data sources provides only their names and a menu button on the right, which allows editing or deleting the data source. For *Traffic mode*, the data sources' list is a little bit different, but it will be described in Section 4.4.2.

Finally, I will mention that filtering data sources by all their attributes are promoted to map view. When the user sets filters via the top bar button, only data sources that meet the criteria of the filters will be visualized on the map. The same principle works on filtering via the search box in the data sources' list component.

4.4.1 Overview mode

The *Overview mode* that is shown in Figure 4.1a is designed for the management of traffic sensors. The users can maintain the data sources, perform basic tasks such as creating, editing, and deleting. The user can filter the data sources to check which traffic sensors are processing at the moment or which ones are already processed. S/he can also see if there are the data sources in error state which means that something went wrong while processing the footage. In the *Overview mode*, it is only possible to filter device type and the state of the data source. The traffic volume filtering is disabled in the *Overview mode*. The example of filtering drone traffic sensors whose footage is currently processing is visualized in Figure 4.3.



DEVICE TYPE FILTER

SELECT ALL

□ Fixed camera ☑ Drone

DATA SOURCE STATE FILTER

SELECT ALL

□ Empty
☑ Processing
□ Processed
□ Archived
□ Error

TRAFFIC VOLUME FILTER

DESELECT ALL

You can filter data sources by traffic only when Traffic mode view is on. You can turn it on in settings option.

☑ Low Traffic
☑ Medium Traffic
☑ High traffic

Figure 4.3: Filtering drones whose footage is processing

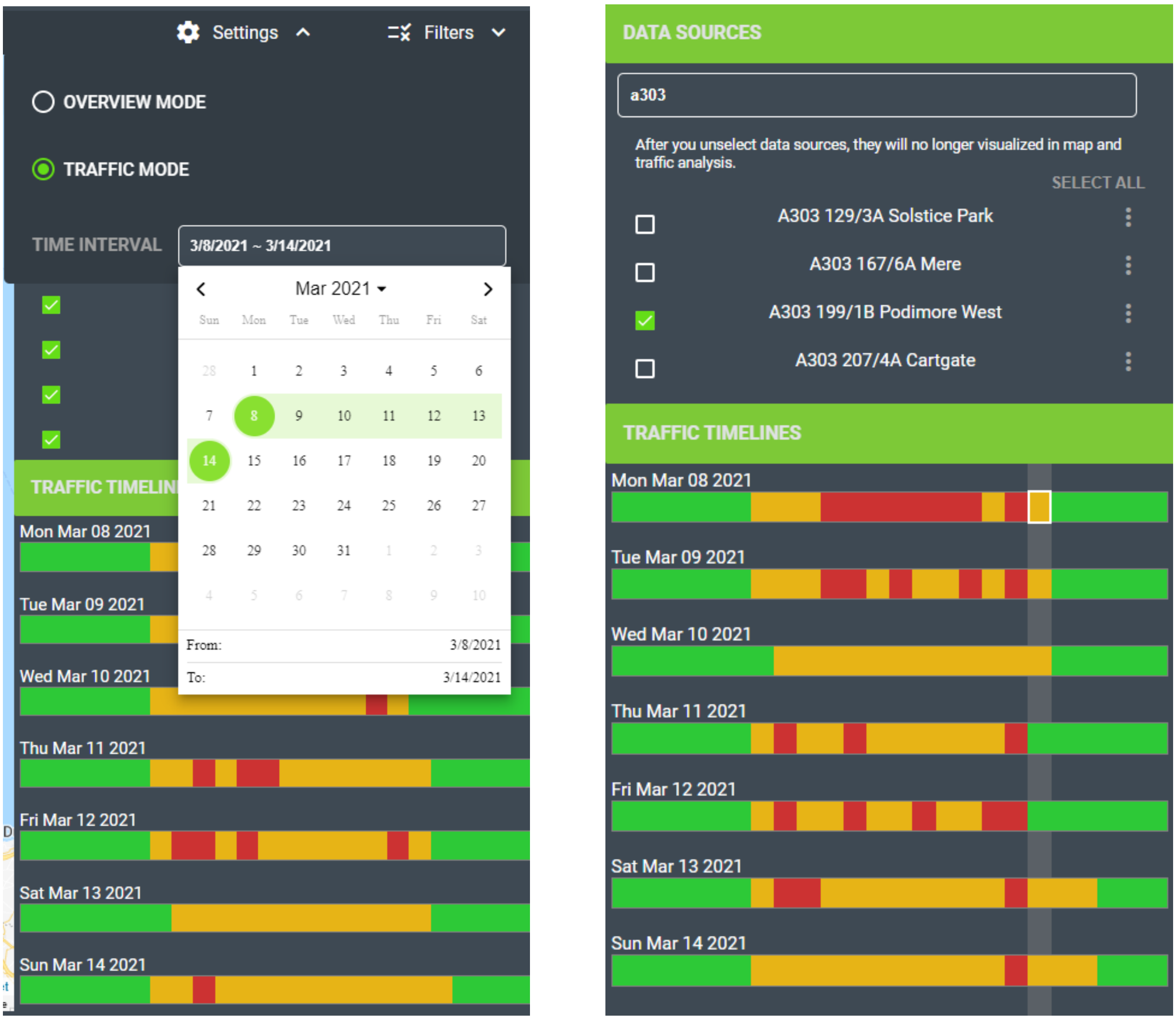
4.4.2 Traffic mode

The *Traffic mode* offers a bit more functionalities. It is essential to say that only processed data sources are visualized in this mode, so the user cannot see data sources in other states. The reason is simple, in *Traffic mode* the users want to see and discover the traffic data, and that is possible only if the video footage is processed and provides these traffic data. In this case, traffic counts and volumes.

In *Traffic mode*, it is also possible to perform basic tasks similarly as in *Overview mode*. It is possible to filter data sources by the device type and traffic volume level. Moreover, as I mentioned, this mode works only with processed data sources; therefore, it is impossible to filter data sources by their state.

When choosing the *Traffic mode*, there is an option to select the time interval for the traffic data, as shown in Figure 4.4a. For the usability testing, there were provided data only from one week. One week is also the max selected interval to visualize right now. When the time interval is selected, only data sources that have some traffic data from this period are visualized, and the average traffic counts across all suitable data sources are visualized in timelines (Figure 4.1b).

Two traffic timelines are present in this mode. One is placed at the bottom of the map with a slider providing average traffic counts from the time interval. The second is placed in the bottom section of the data sources' list. The timeline at the bottom of the map provides information about the traffic counts by the height of the column and traffic volume level by the color of the column. When hovering over the single column, the toolbar



From: 3/8/2021

To: 3/14/2021

After you unselect data sources, they will no longer visualized in map and traffic analysis.

- □ A303 129/3A Solstice Park
- □ A303 167/6A Mere
- ☑ A303 199/1B Podimore West
- □ A303 207/4A Cartgate

(a) Setting time interval for traffic mode

(b) Selecting traffic data for only one data source

Figure 4.4: Implemented high-fidelity prototype - details

shows information about the specific date, hour, and traffic count. When selecting the column, the markers on the map change the color based on the traffic volume in the selected timestamp (selected hour). When selecting the column, the corresponding time slot in the second timeline is selected.

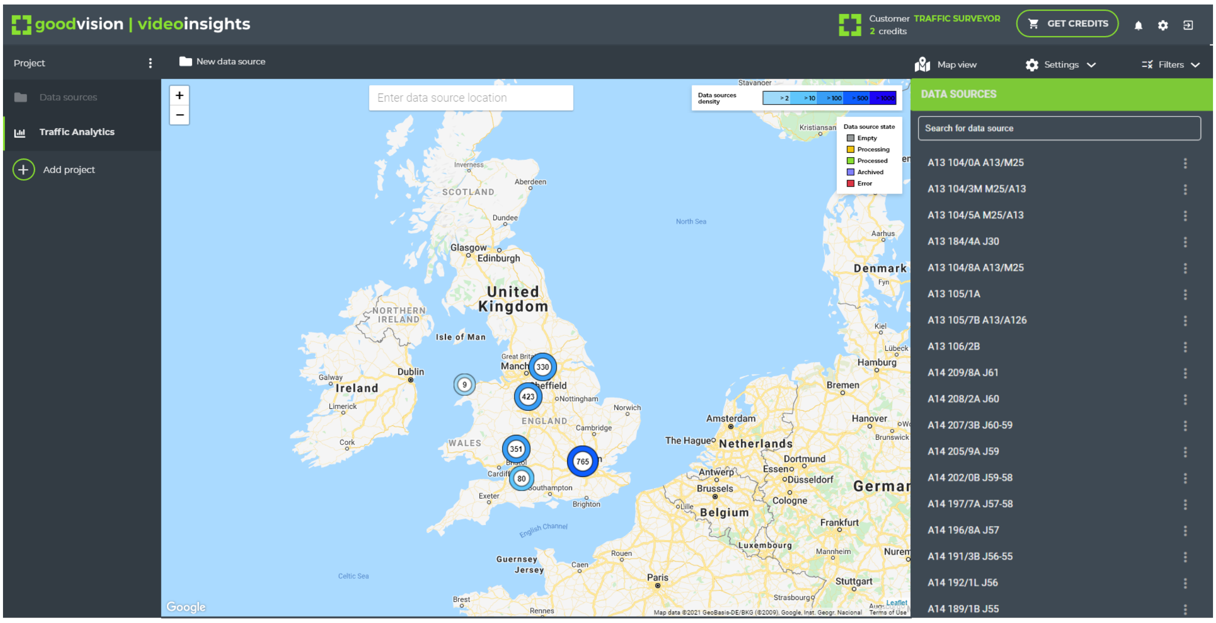
The second timeline visualizes days in a week in rows. Every row represents one day, and it is divided into 24-time slots that are colored based on the traffic volume, the same as in the first timeline. These rows provide average traffic from all of the selected data sources as well. When hovering over the time slot, the specific hour and traffic counts are displayed in the toolbar. This representation of the traffic counts should help the users see the traffic patterns in individual time intervals during the day and compare them with the traffic on other days.

As I mentioned, the timelines show the average traffic across all the data sources. However, what if the user wants to see the traffic only in one location? For better manipulation with data sources, I added the option to choose for which data sources the traffic should be visualized. It is possible via checkboxes near the data source's name, as shown in Figure 4.4b.

4.4.3 Repository structure

The structure of the repository was generated automatically by the Angular framework. All the packages used in this project are defined in the *package.json* and *package-lock.json* files. All SCSS styles are imported in the *styles.scss* file placed in the *src* folder. The primary definition of the user interface is given in the file *index.html.* There are defined the head with scripts and the body. The body is composed of the Angular components placed in *src/app/component* folder. Each component consists of 3 files, HTML defining the component's layout, SCSS defining the visual aspect of the component, and TS defining the logic behind the component. Not all the components are used in the prototype. Some of them were created during the process of implementation but were not applied to the final prototype. The communication between the components is done by services placed in folder *src/app/service.* Additionally, the object definitions, enumerated types, constants, and other entities are defined in the folder *src/app/entity.* In subfolder *consts* also mocked data used for usability testing are defined. Finally, all the icons and pictures used in the prototype are placed in folder *src/assets.*

The environment, technologies, applied data, and all functionalities of the implemented high-fidelity prototype are described above. When I return to Section 2.1.4 where the scenarios of the system were introduced, it can be observed that the high-fidelity prototype allows fulfilling all steps of the proposed scenarios. To determine if it was designed properly, I will execute the usability testing in the next chapter. Before I finish this chapter, I illustrate how the implemented prototype would look after integration to the GoodVision Video Insight platform. It is shown in Figure 4.5.



A13 104/0A A13/M25
A13 104/3M M25/A13
A13 104/5A M25/A13
A13 184/4A J30
A13 104/8A A13/M25
A13 105/1A
A13 105/7B A13/A126
A13 106/2B
A14 209/8A J61
A14 208/2A J60
A14 207/3B J60-59
A14 205/9A J59
A14 202/0B J59-58
A14 197/7A J57-58
A14 196/8A J57
A14 191/3B J56-55
A14 192/1L J56
A14 189/1B J55

Figure 4.5: Integrated high-fidelity prototype into Video Insight platform

Chapter 5

Usability testing

The primary goal of usability testing is to improve the usability of a product. The usability means that the people who use the product can quickly and efficiently accomplish their tasks. The usability testing is executed in a 1:1 session where the participants represent real users and perform the real tasks that the moderator prepared for them. The moderator's role is to record and observe the participant's actions and further analyze the data, diagnose the detected problems and recommend changes to fix them [32].

In this chapter, I will introduce the whole process of the usability testing of the implemented solution. I will present the chosen method for testing, how I performed it and the findings observed during the testing sessions.

5.1 Test preparation

This section describes the quantitative and qualitative testing approach and introduces the method for choosing participants.

5.1.1 Testing methods overview

At the beginning of this chapter, I explained what usability testing is. Now I will introduce two usability testing methods, quantitative and qualitative. The main difference between quantitative and qualitative approaches is the number of participants and the goal of these methods. The goal of the quantitative testing is to evaluate the usability of the existing solution and track the usability over time. It is most likely perform on the complete version of the solution and involves a large number of participants, usually 4 times as many as are needed for qualitative testing [33]. Qualitative testing does not require as many participants. It is suitable to be used during the initial solution evaluation, redesigning, or evaluating the redesigned solution, intending to identify usability issues and find solutions for them [34]. The qualitative study is usually made in iterations when the first iteration with 5 users most likely uncover the 85% of usability problems. These issues can be fixed and redesigned till the next iteration [35].

For my master's thesis, qualitative usability testing is more relevant. I performed it with 9 participants and will discuss the process and the results in the following sections.

5.1.2 Choosing participants

When I sent the questionnaire to GoodVision users and subscribers during the user research, I also allowed them to sign up for future usability testing. I offered this option to people from the qualitative part of the research as well. Both groups are potential users of the designed solution.

ID	Gender	Age	Experience with Video Insight platform
1	female	27	worked with before
2	female	25	no experience
3	male	53	no experience
4	male	27	heard about it
5	female	28	worked with before
6	female	25	worked with before
7	male	35	worked with before
8	male	45	no experience
9	male	25	no experience

Table 5.1: List of participants with specific information

5.2 Testing procedure

In this section, I give an overview of the process of usability testing. I describe how the testing was realized, the condition of the testing, and the whole procedure.

5.2.1 Test's setup

As mentioned in Section 5.1.2, the participants applied for testing during the user research. Because not all the participants are from the Czech Republic and because of the current world's pandemic situation, some of the testing sessions were taken place online. Therefore, there were slightly different conditions for every participant, which is not in line with the recommendation for the usability testing, that the same laboratory should be used [32]. On the other hand, it resembles the real-world usage of the application.

The testing session approximately took 45 minutes per participant. The in-person sessions were conducted in a closed room without any distraction. All participants were given a laptop. The folder with videos needed during the testing and the internet browser with the implemented management system were opened. The printed paper with the tasks was near the laptop. For online sessions, the zipped folder with the videos and the document with the tasks were sent to the user before the testing. I asked the participants to print the document and unzip the folder before the session.

At the beginning of each session, I asked every participant if I can record the session for further analysis. All of them agreed. I also assured them that I am not testing them but the system if it is well designed and intuitive. I asked them to think aloud during the process even though they are not sure if they are doing it right. I told them that the testing comprises two parts, and each has 8 tasks. I informed them that I would ask them about the difficulty of the tasks after each part, and in the end, I will ask them few more questions.

Post-interview questions:

1. Which task was the most problematic for you?
2. Was any of the tasks unnecessary?
3. Were any tasks missing?
4. Which task or functionality of the system did you like the most?
5. Can you imagine using this system for the management of traffic sensors?

5.2.2 Testing scenario

Before the testing, I prepared the test scenario with multiple tasks to perform. Nevertheless, not all the participants were traffic engineers, so I add a background story to the scenario for better empathizing with the role of traffic engineer and the task that needed to be done. The scenario is composed of two parts. Each part consists of 8 tasks. In the first part, the participant should get familiar with the system and try to perform some easy tasks. If the user was unsure about something, s/he could ask for help, and I tried to direct them to the correct solution. This first part was also executed on a smaller amount of data (135 traffic sensors). The second part consisted of more complex tasks performed on the 1958 traffic sensors, and the participants had to perform them on their own.

First part of the scenario

You work as a senior traffic engineer for a big company. Your company wants to apply for a tender to manage traffic across England. As part of a pilot project based in Sydney, you chose to use the system for the management of traffic sensors and monitoring the traffic. The system consists of a map where all available data sources are visualized. Footage of these data sources is processed in the system with the help of artificial intelligence, and traffic counts are accessible for all processed recordings. As you have never worked with the system before, your boss prepared a couple of easy tasks for you and your colleagues to get familiar with the system.

1. Add new data source at coordinates -32.05591, 152.44365 and name it ‘A1 Possum Brush‘.
2. Find a data source in an error state and delete it.
3. Show only data sources which are in processing state or already processed.
4. Show only data sources at M1 Pacific Motorway.
5. Switch to traffic mode.
6. Show only data sources which had on Friday between 4-5 pm High traffic volume.
7. Return filter to show all processed data sources.

8. The timeline shows average traffic across the whole visualized data sources. Show traffic timeline only for one data source and find which day the traffic count was the largest.

Second part of the scenario

Your company just won the tender organized by the UK government to manage the traffic data from almost 2000 traffic sensors across England. Mostly we are speaking of traffic cameras placed on main motorways, but as technologies are innovating, some parts of the motorways are now monitored by drones. Your company started to use the system that you and your colleague are already familiar with. Now it is time to start working on the real tasks. Some of the data sources were added to the system by the previous company a long time ago, so the data sources are already in an archived state. Your boss also uploaded various footage from the second week of March after the tender was won. At the moment, some of the data sources' recordings are already processed, and traffic data are available, but some of the videos are still processing. Now, you are in charge of the system, and there is some work for you to do.

1. Your boss warned you that some of the cameras on the M11 motorway near London are in an error state. That means something unexpected happened to them, and you need to report it to the camera distributor. Find which cameras are needed to be reported.

2. You heard about some rumor about a new traffic survey that already started and using drones to collect the traffic footage. You are curious and want to know how many drones are used for this survey, but you are not acquainted with the city where it takes place. With the help of the management system, try to figure out where (city/motorway) the survey is taking place and how many drones are used for the survey.

3. Right after you found out where the drone survey is taking place, your boss called you to inform you that one drone from the survey is missing from the system. He sent you video footage. You should now add the drone to the system with the coordinates 52.17656, - 2.17467, and the corresponding name is M5 48/5A J6-9.

4. Last week, a new traffic camera was installed in Brighton on Shoreham Bypass(A27) right before the left exit to London Road(A23). Please add a new data source with the name A27 104 A27/A23 and the correct location to the system.

5. In Southampton, 12 cameras are installed. You just found out that the connection between motorways M27 and M3, which allows vehicles to get from West to North East (and in the opposite direction), will be in the reconstruction for next year. It means that all the cameras from this part of the motorway will be uninstalled. You should delete these cameras from the system.

6. Switch to traffic mode.

7. The accident was reported on motorway A13 near London on Wednesday 10th of March during the noon. Can you tell if the motorway was more congested as usual due to the accident when it happened, and for how long it affected the traffic?

8. Your colleagues noticed that the motorway A303 is pretty congested lately. They are thinking about strengthening public transport on this road so the traffic counts will decrease. On the motorway, 4 traffic cameras are placed. Their footage is already processed, and now you need to find out if the traffic is really that congested (the number of vehicles is more than 20 000 per hour*), which part of the highway is congested the most (the name of the most congested camera is enough).

5.3 Findings

In this section, I will introduce observed problems that I found during the usability testing. I present the tasks that caused the most difficulty and also provide possible solutions.

5.3.1 The first part of usability testing

In the first part of usability testing, the tasks were designed to be trivial in order for all participants to gain some familiarity with the platform. For this reason, I was helping the participants if they struggled, which led to completed tasks by all participants successfully. Even though all the users could finish the tasks, I noticed that some of the assignments caused problems or were completed with slight inaccuracy. The following part will describe the complications that occurred in the first, sixth, and seventh assignments. The proposed solutions will be offered at the end of this chapter.

The first task was difficult for almost every participant. They should add a new data source to the system. The complication was that they could not find the button for adding it, placed in the top-left corner of the application. The confusion could be caused by no prior experience and not enough time to get familiar with the system as it was the first task. Further, the button is placed on the left, and all other functionalities are accumulated on the right.

The sixth task was set to join basic filtering of traffic volume with filtering over time. Filter for traffic volume is placed in the filters in contrast with time filtering, which is possible via sliding over the timeline. This has caused a little confusion as the participants expected both filters to be in the same place.

Even though every participant managed to finish the seventh task successfully, they showed some difficulty understanding what should be done. I constructed the task to force them to realize that only processed data sources are visualized in the traffic mode because others do not provide any traffic data. Again, this was mostly an issue for the participants with no prior experience with the Video Insight platform. They did not realize that only processed data sources provide traffic data. It should have been described better before the testing.

5.3.2 The second part of usability testing

During the second part of usability testing, I was not helping the participants with their tasks. I will summarize the observation from the testing. After that, present the deficiencies of the prototype and offer the solutions for them.

In the second task, participants should find a drone survey in the UK. Hence, they should filter the data sources to see only drones and decide where the survey takes place. 38 drones were flying over Worcester, and some of the drones were located elsewhere over England. The correct answer should be that the drone survey took placed in Worcester as there was the most significant number of accumulated drones, but the participants were confused about what should be found. Therefore almost half of the participants answer that the drone survey is all over England. The problem was not filtering the data sources, which was relatively easy for them, but in understanding the task.

The third task was very similar to the first one in the first part of usability testing. It was much easier for the users as they already knew where the *Add new data source* button was placed. Even though the success rate of this task was almost 90%, two of the participants did not select the correct device type of the traffic sensor. This was probably caused by a preselected option of *Fixed camera* in the dialog window. Since it was not a big issue, I will propose the solution shortly here. No device type should be preselected in the dialog window. Then the user has to select the device type every time; otherwise, s/he will not be allowed to create a new data source.

The fourth task was quite problematic, but all of the users managed to accomplish the assignment. The problem was to find the location on the map where the new traffic camera should be installed. I assume the problem was that they were not familiar with the motorways in the UK, so it was not easy for them to find the intersection of two roads. Some of the participants also start finding the position on the map, and because they needed to create a new data source, they were forced to find the location in the dialog window for creating a new data source as well.

The fifth task had the lowest success rate (61.1%) from all of the tasks. It was not understood correctly. The participants usually thought they should remove all the cameras on M27 and M3 motorways, even though the assignment said they should remove only cameras on the specific connection between them. From the post-testing interviews, I discovered that the confusion was based on specifying the number of cameras at the beginning of the task. They assume that all of the mentioned 12 cameras should be deleted. Therefore there was no issue with the system but with an improperly constructed task.

The last task did not cause any issues. I am mentioning it only because I find it interesting that even though each participant accomplished the task accurately, it took them the most time to figure out how to achieve the required result.

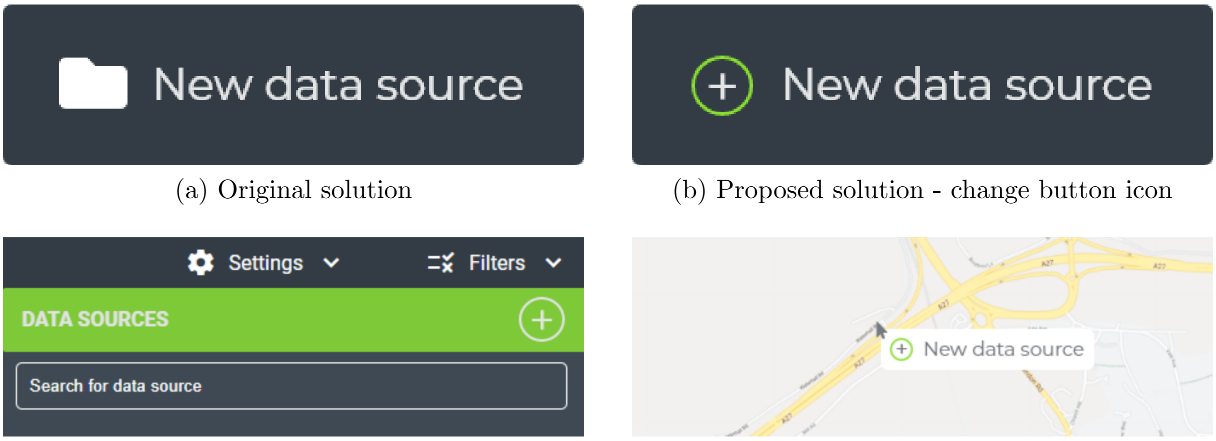
5.3.3 Collected problems and proposed solutions

In this part, I recap the biggest issues found during the usability testing and propose solutions suitable for future development. The problems are listed below.

1. Problem to find *Add new data source* button.
2. It is not intuitive that only the processed traffic sensors are visualized in the traffic mode.
3. Confusion of the colors between modes, e.g., error state (overview mode) with high traffic volume (traffic mode).

4. Layout of the dialog for adding a new data source is not designed correctly.
5. Forgetting to return filters before starting a new task.

Problem to find Add new data source button.
This problem could be solved in different ways. The first option is to change the icon next to the button from folder to plus, as shown in Figure 5.1b that would better signalize the button's function. Another option, showed in Figure 5.1c, is to change the position of the button and move it to the right where most of the components are situated. The last option would be to enable creating a new data source directly on the map. So the users could find the location on the map and then with right-click add new data source as shown in Figure 5.1d.



(a) Original solution

(b) Proposed solution - change button icon

(c) Proposed solution - change position of the button

(d) Proposed solution - right click on the map

Figure 5.1: Original and proposed solutions how to fix not significant button for adding new data source

It is not intuitive that only the processed traffic sensors are visualized in the traffic mode.
One possible solution for this issue could be a popup dialog window that would show up after the user changes the mode to *Traffic*. The dialog window would warn the user that only processed data sources are provided in the *Traffic mode*. The solution is not ideal cause it can cause frustration of the user if it shows every time s/he wants to change the mode. Another way is to put the information directly to the settings where the user chooses the mode. The problem is that users often do not read the additional text.

There is also the possibility to visualize all the markers, even those that are not processed. These not processed markers should be then visualized with low opacity and saturation to signify to users that the data from these markers is not provided in this mode. To be sure which method would solve the problem, it needs to be tested with the users again.

Confusion of the colors between modes, e.g., error state (overview mode) with high traffic volume (traffic mode).
The similarity between the marker icons confused few participants. Even though it was

discovered only in one-third of cases, I assume it could be painful for more users. Therefore I propose the following improvement. When I was designing the markers, I wanted them to be unified. As the usability testing showed, it was not ideal. The markers for different modes should have different representing marker icons, so it is evident that the user is in a different mode. I am offering the possible redesign of the marker icons for an *Overview mode* in Figure 5.2. The new markers for the *Overview mode* do not look the same as the one in the *Traffic mode*. Therefore the user will immediately observe that the mode has changed and that color of the marker in the *Traffic mode* represents the traffic, not the state. The state colors remained the same, but they no longer cover the whole marker. Therefore the difference between the error state and the high traffic is quite recognizable.

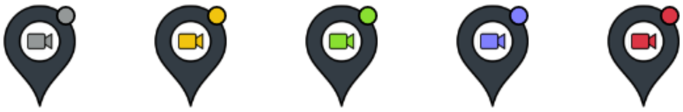


Figure 5.2: Proposed solution - changing marker icons for Overview mode

Layout of the dialog for adding a new data source is not designed correctly.
The dialog window was composed of too much information, so it has to be scrollable. This caused the inputs for coordinates to be not visible initially, and users did use the search box instead of these inputs. After few sessions with the users, I noticed that it caused a problem, and therefore, I fixed the layout of the dialog window. Hence, the inputs for coordinates are now visible immediately. After I changed it, I did not notice any complications in the following sessions.

Forgetting return filters before starting a new task.
During the testing, I observed that many tasks were not finished correctly because participants often forgot to return the selected filters from the previous task. Obviously, I cannot automatically revert filters after every action because the system cannot recognize when the task is completed. Nevertheless, for ease of returning filters, I propose adding a new button to reset the settings. The button could be placed into filters or directly next to the filter button, as it is shown in Figure 5.3, so it will be more visible for the users.

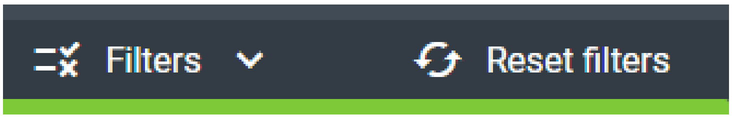
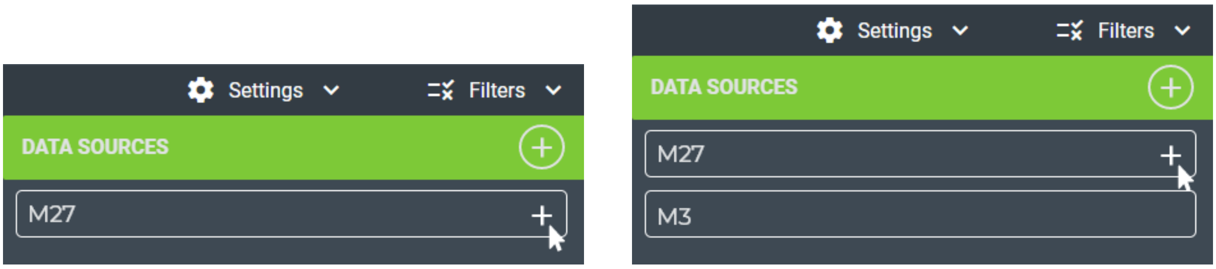


Figure 5.3: Proposed solution - adding reset button

5.3.4 Possible extensions for the future

From the usability testing also some missing features were observed that could be later added to the system. When users set filters, I saw that it would be convenient if the map focused the view right on these filtered data sources. This feature could probably be done by showing only map tiles where markers are present.

Another helpful feature would be to search for more than one motorway by name. For example, when users are looking for the connection between M27 and M3, it would be easier for them to filter both of these names. One option how to reach that would be allowing users to write regular expressions into search, such as and(&), or(||), not(!). However, this could be helpful for experts but maybe too advanced for some of the users. The other possible solution could be to enable adding another search box as visualized in Figure 5.4.



(a) Button for adding new search box

(b) Two search boxes, both searching for different expression

Figure 5.4: Original and proposed solutions how to fix not significant button for adding new data source

The other minor feature that could be helpful for users is providing the counts of displayed data sources, so the user is aware of how many of them are, e.g., processing. Thanks to clustering, the user can already see it, but s/he either has to sum them up or zoom out the map, which is redundant operations. This information could be simply visualized in a component where the data sources are listed.

The last adjustment I observed that could be beneficial is to propagate information about a state/type in the list of data sources in the right-hand panel. Now the list only consists of the name of the data source and actions (edit, delete). Therefore, if the user wants to delete the data source directly from the list, s/he does not have any further information about the data source, what device type, or what state it is in. One possible solution would be adding a device type icon before a name and showing the color of the current state. Proposed solution can be seen in Figure 5.5.

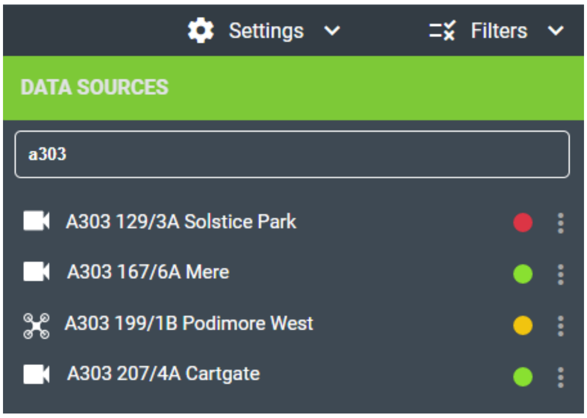


Figure 5.5: Proposed solution - detailed list of data sources

5.4 User requirements fulfillment

The qualitative usability testing was performed with 9 participants. It discovered some minor issues and also brought new ideas to enhance the prototype. Proposed solutions how to improve them are discussed in previous sections. It is important to say that after implementing all these new designs, another iteration of qualitative usability testing needs to be done to ensure that suggested changes helped to solve the problems. After that, the prototype could be integrated into the Video Insight platform.

The usability testing also showed that most of the requirements obtained in Section 2.1.2 were achieved. The prototype is adapted to work with a large number of traffic sensors. It provides the traffic data such as traffic counts and volume, which are essential for traffic analysis. The system also allows extracting traffic data from different traffic sensors that provide video footage from traffic. That enables the analysis of historical traffic data. The prototype allows traffic engineers to work with traffic data across multiple traffic sensors and filter it via time. These features fulfilled the user requirements and therefore covered the gap in the traffic management systems.

Chapter 6

Conclusion

The goal of the thesis was to bring a new effective way to manage and visualize a large number of traffic sensors. To help me achieve that, I started with obtaining user requirements while performing quantitative and qualitative research with engineers who work with the traffic data on a daily basis. When I determined their needs and what traffic data are essential for them, I continued analyzing existing products and applications for managing traffic sensors, visualizing traffic data, and performing traffic analysis. After realizing that the market does not offer a solution that would satisfy all the requirements I collected from the users, I carried on the research. I focused on the visualization techniques that would help me bring the determined data to the users in the best possible way. I also analyzed different map platforms and chose the one which was most suitable for the implementation. When the research was done, I started with summarizing the acquired knowledge to create a sufficient design. There were multiple iterations during the design until I reached the final version. When the design was finally ready, I began implementing a high-fidelity prototype of a client-side web application. Once the implementation was complete, I needed to verify the usability of the application. Therefore, I performed qualitative usability testing with 9 participants. During the testing, I discovered a few deficiencies in the designed system. I fixed some of them between the testing sessions so that they did not influence other participants. For the ones that remained, I proposed solutions that would fix them. Even though a few issues were discovered during the testing, the prototype still meets the user requirements.

As a reminder, this thesis derived from the need to bring the geographical information of traffic sensors into the Video Insight platform. Even though the implemented solution was developed in a separate repository, some features can be later integrated into the platform. The redesign of the Video Insight platform happened during the implementation of this thesis, and one of the parts of this redesign is allowing users to provide the location coordinates via a dialog window similar to the one implemented in the prototype. Bringing the geolocation into the context of traffic data can significantly contribute to the future Research & Development process of the platform. Some of the visionary ideas were already described in Section 3.4 and will be summarized in the next section.

6.1 Future work

As mentioned before, bringing the geographical context into traffic data can deliver numerous opportunities to extend the current platform. Not only from the visualization point of view but also from the computational one. There are many different areas for expansion, such as geo-mapping of the vehicle trajectories (or other traffic data), speed calculation, or origin-destination analysis between multiple data sources, already mentioned in Section 3.4.

To bring it back to the context of this thesis, traffic data with geographical information could offer new levels for visualization tasks. During the thesis, the traffic data was only understood as points without any spatial and directional information. Therefore they were also visualized on the map as markers mapped to specific point coordinates. With automated geo-calibration, it is possible that the traffic data could be mapped to positions of trajectories. In this case, the data could be visualized directly on the map's road system. This could bring a better overview of the traffic from all traffic sensors and further improving the analysis of various traffic patterns.

Bibliography

[1] Worldometer. World population clock. URL https://www.worldometers.info/world-population. Accessed on: 09.01.2021.

[2] David Satterthwaite. An urbanising world. URL https://www.iied.org/urbanising-world. Accessed on: 09.01.2021.

[3] United Nations Department of Economic and Social Affairs. 68% of the world population projected to live in urban areas by 2050, says UN. URL https://www.un.org/development/desa/en/news/population/2018-revision-of-world-urbanization-prospects.html. Accessed on: 09.01.2021.

[4] Datarade. Best traffic data companies, providers and APIs 2020 | datarade. URL https://datarade.ai/data-categories/traffic-data. Accessed on: 09.01.2021.

[5] Andreas Allström, Jaume Barceló, Joakim Ekström, Ellen Grumert, David Gundlegård, and Clas Rydergren. Traffic management for smart cities. In *Designing, Developing, and Facilitating Smart Cities*, pages 211–240. Springer International Publishing. ISBN 978-3-319-44922-7 978-3-319-44924-1. doi: 10.1007/978-3-319-44924-1_11.

[6] Chris Barnham. Quantitative and qualitative research: Perceptual foundations. 57 (6):837–854. ISSN 1470-7853, 2515-2173. doi: 10.2501/IJMR-2015-070.

[7] Tamara Munzner. *Visualization Analysis and Design*. CRC Press, 2015. ISBN 9781498759717.

[8] TomTom Move. O/D Analysis, Version 21.16.1, Route monitoring, Version 2.1.1, 2021. URL https://move.tomtom.com. Accessed on: 27.04.2021.

[9] Geocount. Traffic, 2020. URL https://geocounts.com/traffic/us. Accessed on: 08.11.2020.

[10] RSD, 2021. URL https://mportal.rsd.cz/automaticke-scitani-dopravy. Accessed on: 29.03.2021.

[11] Traffic Cameras UK, 2021. URL https://trafficcameras.uk. Accessed on: 21.01.2021.

[12] Transport for NSW, 2020. URL https://www.livetraffic.com/traffic-cameras. Accessed on: 02.12.2020.

[13] Miovision. DataLink, Version 1.49, 2021. URL https://datalink.miovision.com/demo. Accessed on: 30.03.2021.

[14] RCE Systems. DataFromSky TrafficSurvey, Version 1.3, DataFromSky Viewer, Version 0.5.35, 2021. URL https://ai.datafromsky.com/. Accessed on: 30.03.2021.

[15] GoodVision. Video Insight, Version 23.0.2, 2021. URL https://goodvisionlive.com/goodvision-video-insights. Accessed on: 14.03.2021.

[16] Science Direct. Geospatial data - an overview. URL https://www.sciencedirect.com/topics/computer-science/geospatial-data/pdf. Accessed on: 05.02.2021.

[17] Martin Nöllenburg. Geographic visualization. In *Human-Centered Visualization Environments*, pages 257–294. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-71948-9. doi: 10.1007/978-3-540-71949-6_6. URL http://link.springer.com/10.1007/978-3-540-71949-6_6.

[18] Axis Maps LLC., 2020. URL https://www.axismaps.com. Accessed on: 13.05.2021.

[19] Jacques Bertin. *Semiology of Graphics: Diagrams, Networks, Maps.* Esri Press, 2011. ISBN 978-1-589-48261-6.

[20] Bindiya M Varghese, K. Poulose Jacob, and Avittathur Unnkrishnan. Spatial clustering algorithms. *Asian Journal of Computer Science and Information Technology*, 2014.

[21] Wannes Meert. Clustering maps. Master's thesis, Katholieke Universiteit Leuven, 2006.

[22] Google LLC. Google Maps Javascript API, Version 3.44. URL https://developers.google.com/maps. Accessed on: 06.02.2021.

[23] H. Butler, M. Daly, A. Doyle, S. Gillies, S. Hagen, and T. Schaub. The GeoJSON format. Technical Report RFC7946, RFC Editor, 2016. URL https://www.rfc-editor.org/info/rfc7946. Accessed on: 02.05.2021.

[24] Ben Shneiderman, Catherine Plaisant, Maxine Cohen, Steven Jacobs, and Niklas Elmqvist. *Designing the User Interface: Strategies for Effective Human-Computer Interaction.* Pearson, Boston, 6 edition, 2016. ISBN 978-0-13-438038-4.

[25] OpenJS Foundation. Node.js, Version 15.6.0, 2021. URL https://nodejs.org. Accessed on: 05.02.2021.

[26] npm Inc., Github. Node Package Manager, Version 6.14.11, 2021. URL https://www.npmjs.com. Accessed on: 05.02.2021.

[27] MIT License. Angular, Version 11, 2021. URL https://angular.io. Accessed on: 05.02.2021.

[28] Microsoft. Visual Studio Code, Version 3.4.3, 2021. URL https://code.visualstudio.com.

[29] Vladimir Agafonkin and OpenStreetMap contributors. Laeflet, Version 1.7.1. URL https://leafletjs.com. Accessed on: 24.0.2021.

[30] David Leaver MIT License. Leaflet.markercluster. URL https://github.com/Leaflet/Leaflet.markercluster. Accessed on: 24.0.2021.

[31] Iván Sánchez Ortega. Leaflet.gridlayer.googlemutant, Version v0.10.0. URL `https://gitlab.com/IvanSanchez/Leaflet.GridLayer.GoogleMutant`. Accessed on: 25.02.2021.

[32] Joseph S. Dumas and Janice Redish. *A Practical Guide to Usability Testing.* Intellect Books, 1999. ISBN 978-1-84150-020-1.

[33] Jakob Nielsen. Quantitative studies: How many users to test? *Experience, World Leaders in Research-Based User*, 2006. URL `https://www.nngroup.com/articles/quantitative-studies-how-many-users/`. Accessed on: 25.04.2021.

[34] Jakob Nielsen. Quantitative vs. qualitative usability testing. *Experience, World Leaders in Research-Based User*, 2017. URL `https://www.nngroup.com/articles/quant-vs-qual/`. Accessed on: 25.04.2021.

[35] Jakob Nielsen. Why you only need to test with 5 users. *Experience, World Leaders in Research-Based User*, 2000. URL `https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/`. Accessed on: 26.04.2021.

Appendix A

CD Contents

The attached CD has a following structure:

```
CD
├── Repository
│   └── data-sources-map-view.zip
└── Thesis
    ├── thesis-src.zip
    └── MasterThesis_LuciaSemanova.pdf
```

- **Repository → data-sources-map-view.zip** contains the source code of the high-fidelity prototype of the web-based Angular application. To run the code, Node.js, npm, and Angular CLI are needed to be installed beforehand. Also, the Google Maps API key is not provided, so the prototype will not be fully working. The new API key can be provided in *index.html* file instead of *api_key* in google script. To locally run the application, use *npm run start.*

- **Thesis → thesis-src.zip** contains LaTeX source files for this thesis and figures used within the thesis.

- **Thesis → MasterThesis_LuciaSemanova.pdf** contains a PDF file of this thesis.