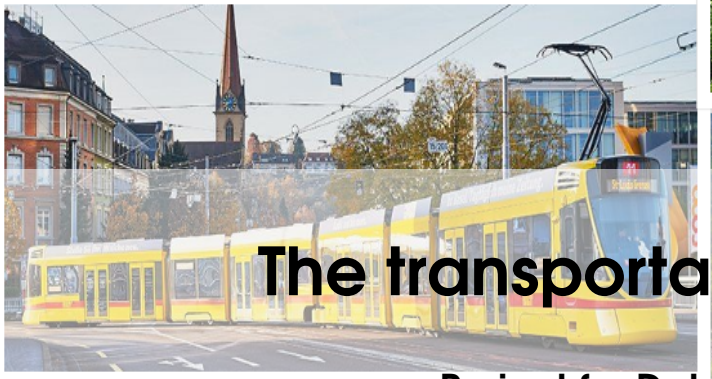



The transportation in Switzerland

Project for Data Visualisation course



Students:

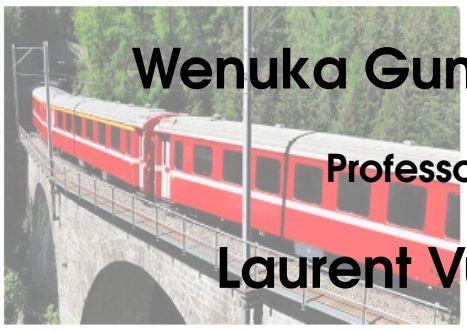
Mariam Hakobyan

Tianchi Yu

Wenuka Gunarathna

Professor:

Laurent Vuillon




PROJECT FOR COM-480 DATA VISUALIZATION COURSE, EPFL

Github - `https://github.com/com-480-data-visualization/com-480-project-story-tellers`
Web page - `https://threestorytellers.github.io/`

Spring 2020, COVID-19 time

Contents

1. Introduction

1.1 Overview

The public transportation system of Switzerland is playing a big role in mobility system. People use the transportation system for different purposes: work commute, sport, travel, outdoor or mountain activities, so on. This is where the **motivation** of the project comes from: "Visualize the time and space dynamic data of transportation system of Switzerland and find interesting insights".

The **inspiration** of the map comes from the ocean-ship map visualization by *Klin digital* team. We have found several life tracking applications for the transportation system, but more specifically the top view of the running vehicles of different types in delta time has not yet implemented for Switzerland to our knowledge. To that end, we have visualized a Dynamic flow Map for moving vehicles in time and graphical interactive figures to get detailed view of the system. The **overview** of the visualization is a process of interactive map in different map modes and graphical figures for statistical to answer research questions. The **target audience** of the project is planned to include people who are interested in transportation system, network visualization, Switzerland and who are just curious about things in world. Will require some knowledge for the user to interact with the map functionalities and to understand the statistical figures.

1.2 Data

We have used **Open Data Platform of Swiss Public Transport** for almost all data we have processed in the visualisation.

- 2020 Timetable data [1] - data obtained initially and used for graphs in the website.
- GTFS format data [2] - public transportation feed of Switzerland used into Interactive Map visualization
- GEO Json of Switzerland map [4] - for map drawing map boundaries,
- Station information [3] - Overall spatial information of big stations.

While processing the timetable data we have obtained 473233 schedules in a day out of which 78% share belongs to bus schedules Figure 1.1.

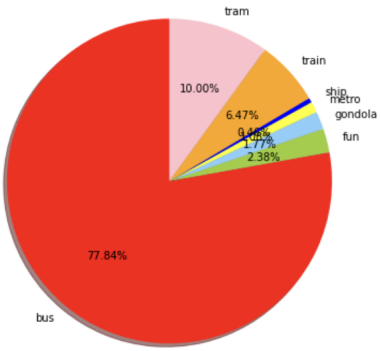


Figure 1.1: Transportation type distribution in schedules

The high percentage of bus share in day schedule is because of bus frequencies and bus distribution in almost all regions. Because of data size limitation we have excluded the data for bus vehicles and stations from the map visualizations.

1.3 Data for Interactive Map

Since the GTFS text files did not had shapes.txt files, we have applied the *Precise OpenStreetMap (OSM)* map matching to public transport schedules API to get the exact trajectory of the route from point to point depending on what the the mode of transportation [6]. New *shape.txt* file has been generated during conversion to characterize the real trajectory of a trip.

Once the completed text data was there we use the Github repositery [11] to convert the text data to geojson files containing map related data and other trip related data to Sqlite format.

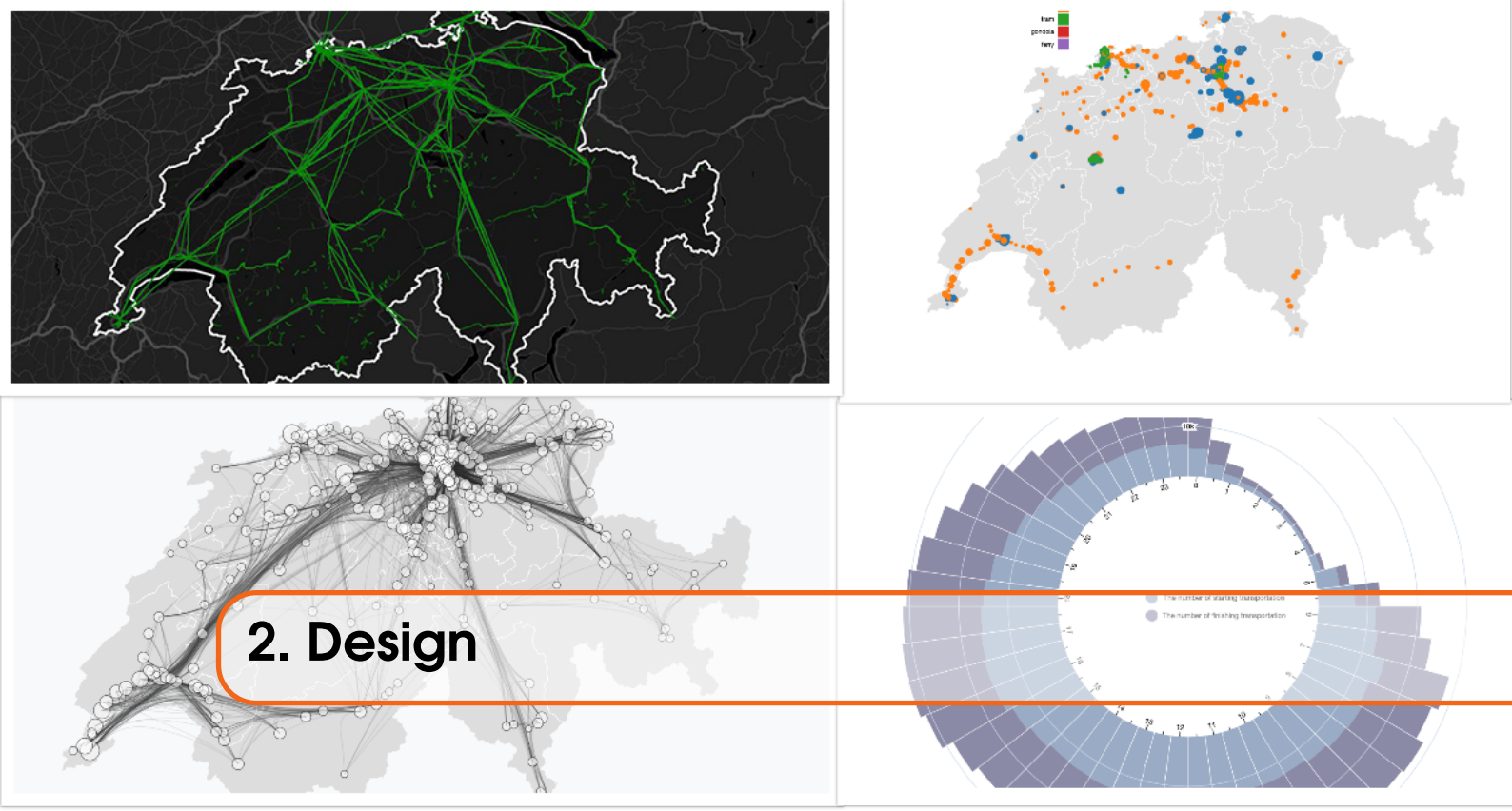
However the Sqlite database was around 2GB, which was impossible to load along with the web site, therefore we had to move that entire database to Mysql and hosted it on a third party hosting provider. The $gtfs\_shapes.geojson$ which was containing shapes corresponding to each trip was also bigger than the maximum size allowed our server, *github.io* limitation we had to rethink about implementing it. When analyzing the data, around 97% data was due to busses and trains. Therefore due to server limitations we were force to add less detailed shapes for trains (straight lines in between stations) and also to exclude busses in full. Further compression was done using a python script finally achieving a geojson file of 6.7MB for shapes which was a 99% reduction in size when compared to original file.

1.4 Data preparation for d3 js graphs

From the start and end time of the trips of timetable data we have extracted the number of Initiating and Terminating transports in different time frequency: {hourly, half hourly, quarter hourly}. This data later on reflects in the Radial Bar chart in "hour" frequency.

Using the Station information data we could extract the top 300 train stations in Switzerland in terms of number of passenger boarding. Then we extracted the direct connections for the 300 stations and the number of trains passing through each connections in a day. This information is used in Edge bundling graph.

We have also extracted the number of entering transportation in any type of stations(except bus type, because of aforementioned reason) in time, which later on reflects in the Time interactive map graph. We have used python to extract data for d3 js graphs, and saved into the $/github_web/data$ folder.



2. Design

2.1 Interactive Map

The one of the main milestones of the project was to create an interactive flow-map for the transportation system, which will depict the flow of the vehicles in time and space.

We have started the interactive map by refering to the Github project [5] which was using the Google API for constructing the map. On top of the map we have incorporated the layers [12] with Geojson and Json data files, to draw the the points as moving vehicles in delta time, points as stations with coordinates, lines as traffic roads. We have also implemented functionalities for the map to contain information about stations: Geo-positions and operating transport types, traffic routes defined by the trajectory of the trips. This functionalities are available to user in the form of the Figure 2.1 UI.

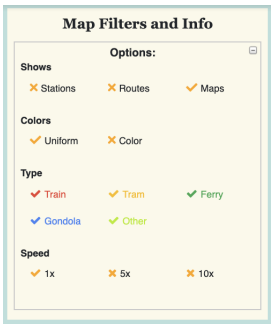


Figure 2.1: Map filters and Info

The most challenging task of the interactive map was to show the most important data within the server limitations. We had to spent a significant time creating an API and also to reduce the existing data files.

Several filters and functionalities were implemented on the interactive map. In the filter section, filters were added to show/hide stations, routes and also the map of Switzerland. Add a color

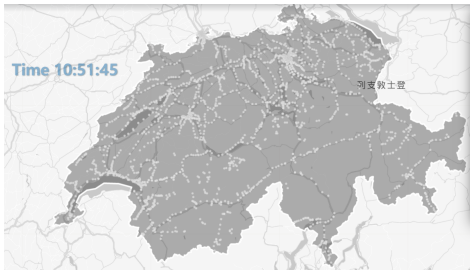


Figure 2.2: Uniformly colored Stations

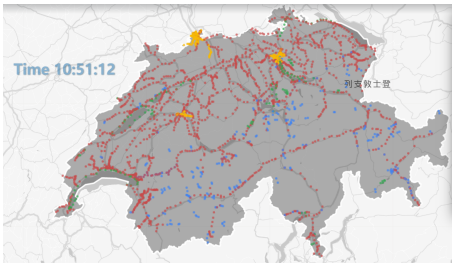


Figure 2.3: Multi coloured Stations

scheme to each type of stations were was also added along with a filtering process for each type of vehicle. The speed set button was designed to animate vehicles at different speeds while the time bar was used to navigate the user to any preferred time in the current date.

Vehicles Control

The most important idea for us is achieving some kinds of simulation of traffic movements by the actual schedule of vehicles on the map. One of the most interesting functionality is activating and controlling vehicles.

Vehicle simulation: We constructed the line-Pools and picked the coordinates in every route as the positions of vehicles. On the map, with the time changing, there is a nice simulation for the traffic moving. With different icons, the moving of different types of vehicles is clear and easy to be understood.

Vehicle info: We also want to show the detailed information of vehicles. So, users can focus on each vehicle on the map. When users click every nodes of vehicles, the basic information and the schedule of that vehicle will be posted on the screen, where users can check every stations and arrive or depart time.

Vehicle following: Besides the overall display effect, it will be really nice if users can follow every vehicle on the map by the *follow* button in the block diagram on the right after one vehicle being chosen. The map will focus on the movement of single vehicle. Of course, we can stop the following and back to the general simulation.

Vehicle path: Until now, the path of vehicles is still not visual-able, for each vehicle, we add an option of showing or hiding the single route at that moment. By this, the whole information about the vehicle is properly show out.

Stations & Routes

After vehicles, the interesting thing is to describe the locations of stations.

Stations showing: To avoid the confusion of large number of points on the map, the initial stats for the stations showing is turned off, which means the stations layer is hidden. With the button of stations, we can show the overall layout of all stations in the data set to the map.

What do we want to show is the distribution of the stations in Switzerland, with this figure, it is easy to see the traffic intensity. But without classification, we can not get more information from that. So, we involved different colors into the stations.

Stations classification: Show the stations in different colors according to different types.

To make the picture more visual-able and distinct, we chose the color palette of Google brands. which you can see in Figure 2.3

Stations filter: With the classification, the filer option is another functionality by which the user can easily research deeper about different types of stations.

Route showing: Having the nodes, the “trees” are ready to come out. We want to take the traffic flows as the “blood vessels” and take the vehicles as the “flowing blood”, by which the whole picture of the entire Swiss transportation network can be vividly displayed.

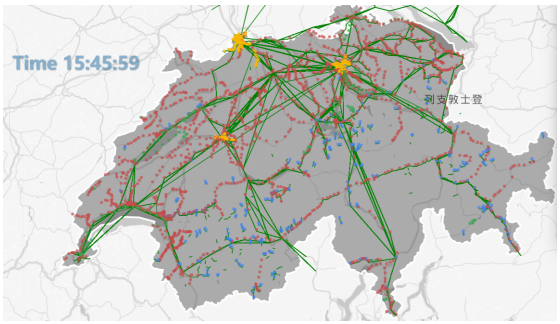


Figure 2.4: traffic roads

Time Functionality

There is one more important functionality about the timer. If all simulation is fixed, it will be unacceptable for the users waiting for several hours to watch the simulation of vehicles later in that day. Then designing a timer would be necessary.

Time showing: Initially, it should start from the real time when users come into the website.

Speed: Because the schedules of trains are usually long, we need speedup the velocity to watch the movement fluently. We set three levers there, users can do researches on the 1x, 5x and 10x speedup stats.

Time control: It is not enough, if we can access any time in one day, it would be much convenient and humane. Then, we add a time bar at the bottom as a entry for every 10 minutes in one day.

2.2 Radial bar chart

The research question around this graph initially was posed as: *"What is the "day" routine of transportation schedule?"*. We were keen to see the night and day shifts and the pick time of the transportation schedules. Initially, we planned the graph to be Bar chart in flat axis, but the fact that we were dealing with time periodic data where 23 : 55 is very close to 00 : 00 we came up with the *Radial stacked bar chart* representation of the number of initiating and terminating transports in each **half hour**. Figure 2.5.

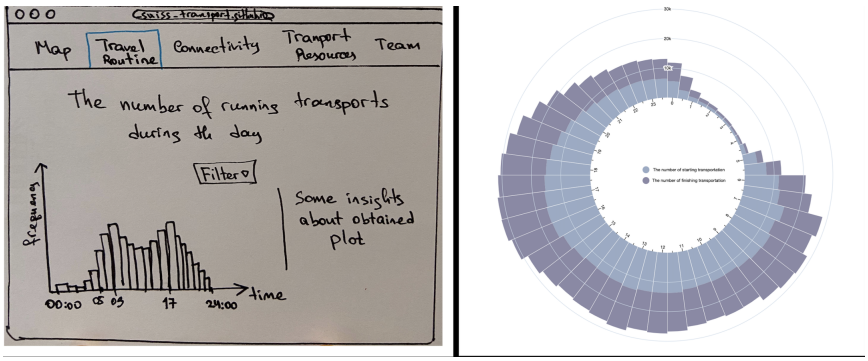


Figure 2.5: Routine graph: Sketch and Realization

We treat the **time** as discrete categorical variable to plot the Radial Stacked Bar Chart. The reason of visualizing also the number of terminating transportation is to see the transformation

from pick to non-pick time and visa versa. The used can hover around the graph and get the number of transportation for each class and for specific hour.

2.3 Map bundling

We were interested to see visualization that depicts: "How connected are the big train stations in different parts of the country?". The term connected here means direct trip from a stop to stop. Initially, we had planned to represent this information as a force graph where nodes would be the stations, colored according to canton group and the edges will be simulated by physical forces characterizing the frequency of trains for connected nodes.The implementation of the graph came out not very visually appealing because we had large number of nodes and edges which created visual clutter. We had to choose the number of nodes and threshold the edge power in a way that the network would be connected so that the user would have the best experience to see all the nodes. But in that case we may appear to be biased in data selection.

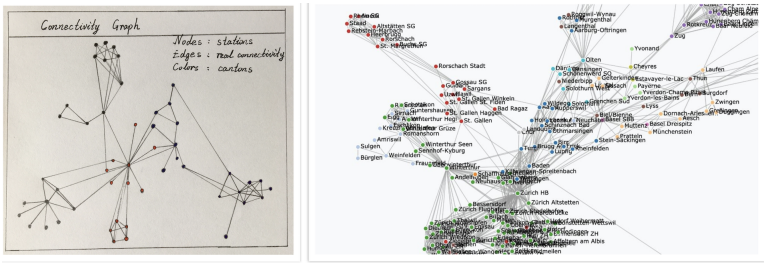


Figure 2.6: Force graph: Sketch and Realization

In the process of brainstorming solutions to the challenge of visualizing the connectivity graph, we realized the fact that stations had GEO spatial information {latitude, longitude } which can not be forced in the force graph as the locations od nodes are defined by physical force, but can be used to position the nodes properly. The edge bundling was considered as a solution, and the example of Flight Paths Edge Bundling of US airports appeared to be starting point for map, data and interaction [8]. We use Swiss boundaries topojson, stations and trips data processed and located in $/github_web/data$ folder for defining the graph and edge weights. The advantage of edge bundling visualization on the map is that the stations are fixed in the projected Geo locations and there is no need for hierarchy. The source code uses d3 geo Voronoi-arc to draw the diagrams and edge bundling. The result of visualization is the following Figure 2.7.

Figure 2.7: Edge bundling on Map

The user can over around the station and see the most powerful connections of the nodes/stations.

As well as the tool tip will give the station name and canton information of the node/station.

2.4 Capacity

We were curious about the amount of transport resources by type in each canton initially. So the stacked bar chart was the initial plan for the visualization where the classes would be the cantons and the groups the transportation type. While visualizing this bar chart we have noticed that the groups are unevenly distributed(trains are dominating) and the graph wasn't visually appealing. Figure2.8

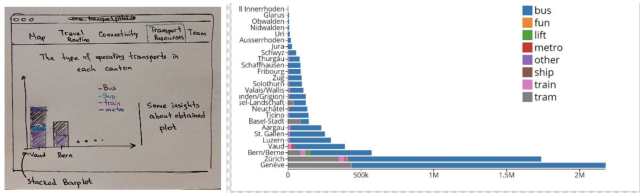


Figure 2.8: Initial Sketch of stacked bar chart and Realization

The solution to this challenge again came in the form of transforming the visualization into space/map and time domain. The key information to the user is the coloring pallet of different types of the stations and the circle radius characterizing the amount of transportation in specific time period. To give more flexibility to the user we have added the Timeline bar where the user can choose the specific time period and length for seeing the amount of incoming vehicles in each station. The user can hover around the stations and get name and count information with the help of tool tip.Figure 2.9

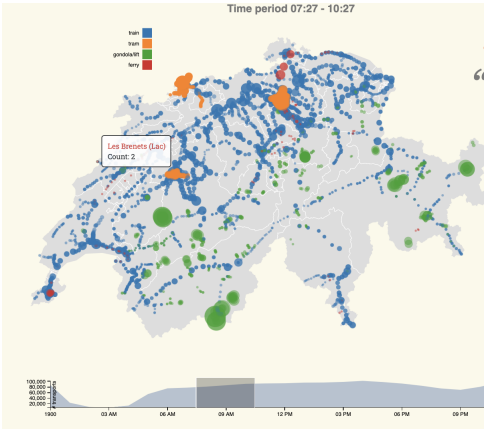


Figure 2.9: Initial Sketch of stacked bar chart and Realization

3. Terminus

3.1 Project success evaluation

The have implemented almost all the steps that we have defined in the the beginning of the project for visual data story of transportation system. We have put a lot emphasizes on having informative and user friendly visualizations. The only challenge that we haven't thought about in the beginning is the size of the data and lack of web hosting resources in terms of memory. We have built the visualization framework which was tested to be working with small amount of data, but we couldn't host all the data we have planned to. Thus, we had to filter out several routes and trips from the data to reduce load of the memory. We have evaluated that we don't lose a lot information by removing the bus routes because the bus schedule are quite uniform in time and space(across country) so there was no diversity to depict with bus transportation modes. After filtering the data we again end up having files which exceed the max limitation of github. Thus, we put very small sample of the data in the working web and to see all the vehicles movement, we would ask to run the web locally and download the files from this link to **githubio_web.api.gtfs_data** file.

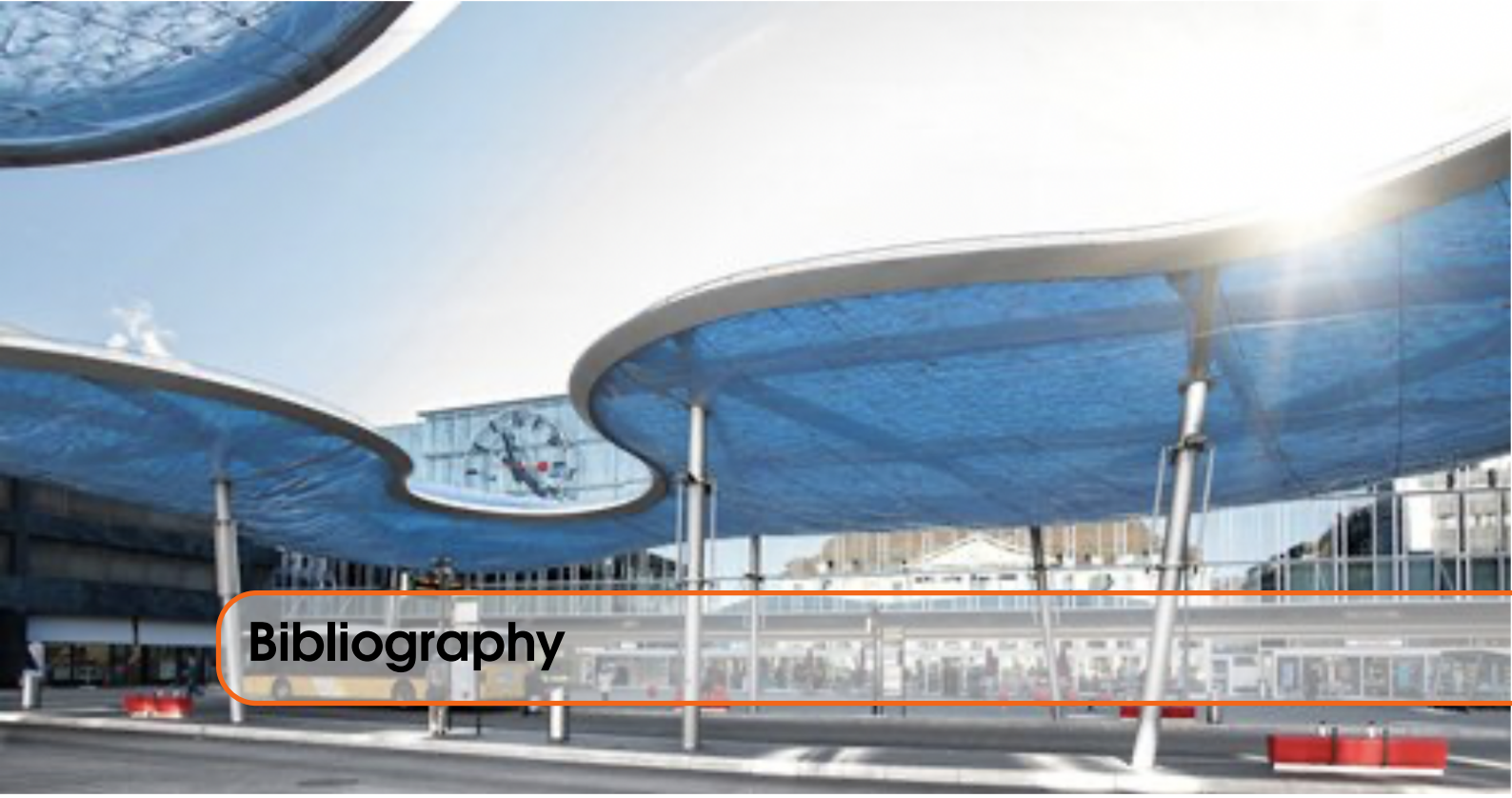
3.2 Task distribution and Peer assessment

We quite managed to implement initially defined things without even one face to face meeting because of COVID19 confinement. The tasks of the project are composed of the following parts:

- Data processing - Initially Mariam and Tianchi processed data and found insights, Wenuka worked on google map API. GTFS data was pre-processed by Mariam while Wenuka converted them to Geojson and SQL format and compressed.
- Data hosting - Wenuka worked to put the large datasets into a remote server and connect to main web on github.io.
- Interactive Map visualization - Ideas have been discussed all togethor
  - Backend - Api to read time table was mainly done by Wenuka
  - Frontend - UI parts and interactive web design was mainly done by Tianchi also with some help from Mariam and Wenuka
- The d3.js graphs - Ideas have been discussed all together. Routine graph, Connectivity graph

and Capacity graph was done by Mariam while Tianchi and Wenuka were ready to help whenever needed.

- Overall design - Mostly Wenuka, Tianchi and Mariam have also given a hand,
- Process book - all togethor,
- Screencast - Mariam



Bibliography

REF

[1] Timetable data - https://opentransportdata.swiss/en/dataset/timetable-2020-hrdf

[2] GTFS data - https://opentransportdata.swiss/en/dataset/timetable-2020-gtfs/resource/559278ea-fd2a-4d93-a462-f9c6d7b69dae

[3] Passengers boarding and alighting for train stations- https://opentransportdata.swiss/en/dataset/einundaus

[4] GEO json data for Switzerland bounderies - https://github.com/interactivethings/swiss-maps

[5] Transit map Api - https://github.com/vasile/transit-map

[6] Pfaedle - https://github.com/ad-freiburg/pfaedle

[7] Radial Stacked Bar Chart - https://bl.ocks.org/KoGor/9f3932d3c7154f8ab3ea2078b2aca113

[8] Flight Paths Edge Bundling - https://bl.ocks.org/sjengle/2e58e83685f6d854aa40c7bc546aeb24

[9] Interactive map in d3 js - https://bl.ocks.org/domhorvath/dd850c5e97d4022fbf0f11611a0cf528

[10] Google GTFS reference - https://developers.google.com/transit/gtfs/reference

[11] GTFS vizualisation from GTFS text files - https://github.com/vasile/GTFS-viz

[12] Google Layer building - https://developers.google.com/maps/documentation/javascript/layers

4. Script for Screencast

Welcome to the screencast of the visualization project of Swiss Public transportation system. The aim of the project is to Visualize the space and time dynamic data of one of the key pillars in Swiss mobility and find interesting insights.

We were interested to visualize the flow of the vehicles in delta time on the map, the transportation day routine, station connectivity and stations capacity.

The main pillar of the visualization is this interactive map, which shows the different types of vehicles moving in time and according to schedule. The mode of the Vihecles are train, ferry, gondola/lift, tram. While hovering around the vehicle we can also get information about the trip, destination and schedule. The User has a chance to change and interact with a data by exploring the Options section in the Map filters part, We can choose what we want to see on the map, for example stations and in case of Opting color option in the second part we see the type of stations, and the user can filter out what to see and not to see, The user can also choose to see the routes which have been generated by the Open street maps and GTFS schedule data to get proper trajectory mapping. And most importantly the user can Speed up the time by 10 second per second maximum and see the relatively fast movement of the vehicles. In time bar we can also change time to see the vehicles movements in different time of the day.