

Vizualizace vývoje dopravního toku v čase

Visualisation of Traffic Flow Evolution Over Time

Pavλίna Smolková

Bakalářská práce

Vedoucí práce: Ing. Kateřina Slaninová, Ph.D

Ostrava, 2023

Zadání bakalářské práce

Student:

Pavλίna Smolková

Studijní program:

B0613A140014 Informatika

Téma:

Vizualizace vývoje dopravního toku v čase
Visualisation of Traffic Flow Evolution Over Time

Jazyk vypracování:

čeština

Zásady pro vypracování:

Student nebo studentka se bude v rámci své práce věnovat vizualizaci vývoje dopravního toku v čase. Informace o dopravě budou získávány z dopravního simulátoru, který pro daný časový interval, a tak zvanou origin-destination matici, provede simulaci průběhu jednotlivých vozidel na mapě. Pohyby jednotlivých vozidel jsou v pravidelných intervalech zaznamenávány. Cílem práce je ze záznamu pohybu aut vygenerovat plynulé video zobrazující dopravní tok z dané simulace. Vizualizační nástroj bude umožňovat také nastavení stylu konečné vizualizace.

Jednotlivé body zadání:

1. Seznámit se s aktuálním stavem dopravního simulátoru vyvíjeného na IT4Innovations, zejména s formátem záznamu pohybu aut.
2. Prozkoumat aktuální možnosti vizualizace dopravních toků.
3. Prozkoumat vhodné knihovny a technologie, provést srovnání a vybrat nejvhodnější kombinaci s ohledem na dostupnost zdrojových kódů. Musí se jednat o open-source knihovny.
4. Implementovat vizualizační nástroj s možností konfigurace konečné vizualizace.

Seznam doporučené odborné literatury:

- [1] Matplotlib: Visualization with Python. [Online]. Available at: <https://matplotlib.org/>
- [2] D3.js JavaScript library. [Online]. Available at: <https://d3js.org/>
- [3] W. Chen, F. Guo and F. -Y. Wang, "A Survey of Traffic Data Visualization," in IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 6, pp. 2970-2984, Dec. 2015, doi: 10.1109/TITS.2015.2436897.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Kateřina Slaninová, Ph.D.**

Datum zadání: 01.09.2022

Datum odevzdání: 30.04.2023

Garant studijního programu: doc. Mgr. Miloš Kudělka, Ph.D.

V IS EDISON zadáno: 07.11.2022 12:25:53

Abstrakt

Tato bakalářská práce se zaměřuje na vývoj vizualizační komponenty, která umožňuje generování snímků pro video zobrazujících data z dopravního simulátoru. Komponenta poskytuje vizualizaci dat pomocí barevného spektra a tří typů rozšíření segmentů dopravní sítě. Kromě toho nabízí konfigurační možnosti dle měřítka mapy a typu dopravní komunikace.

Klíčová slova

vizualizace dat, doprava, vizualizační knihovny

Abstract

This bachelor thesis is focused on the development of a visualization component for generating frames for video displaying data from a traffic simulator. This component provides data visualization using a color spectrum and three types of segments' width styles. It also offers configuration options based on map scale and route type.

Keywords

data visualization, traffic, visualization libraries

Poděkování

Ráda bych poděkovala paní Ing. Kateřině Slaninové, Ph.D. a panu Ing. Martinu Šurkovskému, Ph.D. za jejich odborné vedení, užitečné rady a ochotu při konzultacích v průběhu mé bakalářské práce.

Obsah

Seznam použitých symbolů a zkratk	8
Seznam obrázků	9
Seznam tabulek	10
1 Úvod	11
1.1 Popis simulátoru	11
1.2 Požadavky na aplikaci	13
2 Aktuální možnosti vizualizace	14
2.1 Typy vizualizací	14
2.2 Dopravní informace na mapových portálech	19
3 Srovnání vizualizačních knihoven	21
3.1 Matplotlib	21
3.2 GeoPandas	22
3.3 OSMnx	22
3.4 D3.js	23
3.5 Plotly	23
3.6 Zhodnocení	24
4 Návrh řešení	26
4.1 Vykreslení barvy	26
4.2 Vykreslení rozšiřujících se hran	27
4.3 Plynulý přechod mezi hranami	32
4.4 Adaptace vizualizace dle měřítka mapy	32
5 Implementace	36
5.1 Struktura vykreslovací funkce	36
5.2 Vykreslení kolekce	37

5.3	Styly rozšíření hran	37
5.4	Filtrování segmentů dle stupně přiblížení	38
5.5	Doporučená konfigurace	39
6	Závěr	41
	Literatura	43

Seznam použitých zkratek a symbolů

API	– Application Programming Interface (rozhraní pro programování aplikací)
CSS	– Cascading Style Sheets (kaskádové styly)
ČSN	– české technické normy
DPI	– dots per inch (bod na palec)
HSV	– Hue, Saturation, Value (barevný model odstín-sytost-jas)
HTML	– Hypertext Markup Language
NDIC	– Národní dopravní informační centrum
OSM	– OpenStreetMap
RGB	– Red, Green, Blue (barevný model červená-zelená-modrá)
ŘSD ČR	– Ředitelství silnic a dálnic České Republiky
STC	– Space-Time Cube (časoprostorová kostka)
SVG	– Scalable Vector Graphics (škálovatelná vektorová grafika)

Seznam obrázků

1.1	Segment dopravní sítě rozdělen na úseky dle své geometrie	12
2.1	Jednoduchá bodová mapa (nalevo) a teplotní mapa (napravo) [10]	16
2.2	Pentlogram ČR 2020 [16] a pentlogram křižovatky Praha-Kurnatice [15]	17
4.1	Vykreslení hran s mapováním barev	27
4.2	Porovnání rozšíření přímky definované 50 body (vlevo), 500 body (uprostřed) a 50 body s gradientem (vpravo)	29
4.3	Vykreslení rovnoměrné šířky posunem ve směru osy y (nalevo) a osy x (uprostřed) a dle osy y s gradientem	30
4.4	Výplň dle equidistanty (nalevo) a zobrazení normál (napravo)	31
4.5	Srovnání dvou typů rozšíření bez a se zaoblením hran	33
4.6	Srovnání vykreslení podkladové mapy	34
4.7	Vykreslení mapy v měřítku 1 : 320 000	35
4.8	Vykreslení výřezu mapy v měřítku 1 : 80 000	35
5.1	Srovnání stylů BOXED se zaoblením a EQUIDISTANT bez zaoblení	40

Seznam tabulek

2.1	Frekvence mapových typů ve zkoumaných článcích [5]	15
3.1	Srovnání knihoven	24
5.1	Stupně přiblížení mapy	38
5.2	Souhrn parametrů funkce plot_routes	39

Kapitola 1

Úvod

Data jsou začátkem poznání. Způsob jejich zobrazení je však mnohdy stejně důležitý jako data samotná. Při analýze např. provozu na křižovatce, může množství informací, které se na nás současně valí, vést ke ztrátě přehledu a schopnosti identifikovat vzájemné vztahy mezi sledovanými proměnnými. Na základě správné vizualizace jsme schopni se podívat na celou situaci jiným úhlem pohledu. Tento nový pohled pak může být správným krokem k optimalizaci sledovaného procesu.

Cílem této práce je vytvořit vizualizační nástroj pro vytváření videí zobrazujících průběh simulovaných dopravních situací ve městech. Jedná se o nástroj, který bude snadné použít, a přenést tak numerická data do uživatelsky přívětivějšího formátu vykreslením v podobě síťového grafu. Tento graf zachovává tvar geometrie dopravní sítě, kdy je každá cesta reprezentována hranou grafu, což umožňuje snadnější představu o poloze a typu problematických úseků.

K vývoji vizualizačního nástroje dochází ve spolupráci se Sofíí Michailidu, která se zabývá optimalizační stránkou zpracování dat a vytváření videa z vykreslených snímků. Mým cílem je vytvoření komponenty, kterou tento nástroj využívá. Tato komponenta slouží k vygenerování jednotlivých snímků s důrazem na estetiku, snadné čtení dat a konfiguraci stylu zobrazení.

Text bakalářské práce se skládá z následujících kapitol. Kapitola 2 přináší přehled aktuálních způsobů zobrazování geografických a časově se měnících dat a jejich využití v mapových portálech. V kapitole 3 se přesouvám k porovnání dostupných vizualizačních knihoven a jejich možného využití jako základ pro vyvíjenou komponentu. Následuje kapitola 4, která popisuje různé možnosti řešení požadavků na vykreslování. Způsob zakomponování těchto řešení do výsledné aplikace je dále popsán v kapitole 5. Závěr práce obsahuje zhodnocení dosažených výsledků a srovnání s vytyčenými cíli.

1.1 Popis simulátoru

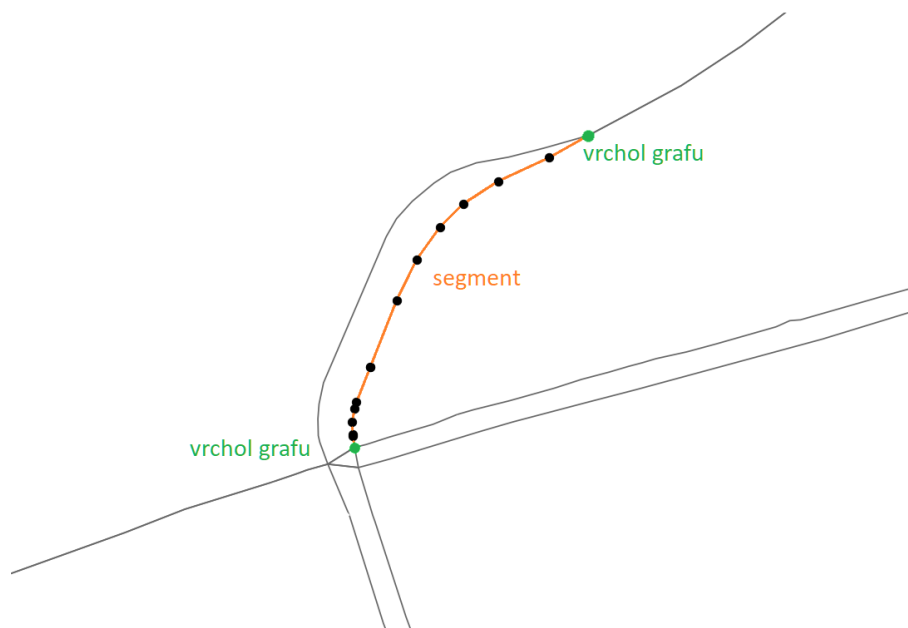
Data, která slouží jako základ vizualizací v mé práci, byla získána ze simulátoru využívaného v rámci projektu EVEREST [1]. Projekt EVEREST cílí na vytvoření uceleného přístupu navrhování výpočtů a komunikace v systému pro vysoce výkonné datové analýzy. K tomuto účelu je využíván datově

řízený návrhový přístup, hardwarově zrychlená umělá inteligence a efektivní monitorování spouštěných úloh. Jednou z případových studií je vytvoření systému pro modelování dopravy v reálném čase pomocí vlastního simulátoru pro chytrý provoz v rámci konceptu smart cities.

Hlavním využitím simulátoru ¹ je optimalizace plynulosti silničního provozu vybrané oblasti. Deterministický charakter simulátoru navíc zajišťuje, že pro stejné nastavení vstupních parametrů je výsledek simulace vždy totožný. Tato vlastnost je velice důležitá z hlediska porovnávání výsledků a jejich opakovatelnosti.

Dopravní síť simulátoru je reprezentována grafem skládajícím se z vrcholů reprezentujících křižovatky a hran reprezentujících dopravní segmenty. Každý segment obsahuje údaje o své geometrii zakřivení, podle které jej lze rozdělit na soustavu rovných úseků. Na obrázku 1.1 můžeme vidět rozdělení segmentu na tyto úseky černými tečkami. Vrcholy grafu dopravní sítě spojené tímto segmentem jsou označeny zeleně.

Při generování simulace dochází v pravidelných časových úsecích k aktualizaci polohy vozidla na základě použitého algoritmu a vytvoření datového zápisu pro každý takovýto pohyb. Zápis obsahuje časový údaj, informace identifikující vozidlo a polohu vozidla na základě dopravního úseku, na kterém se vozidlo nachází [2].



Obrázek 1.1: Segment dopravní sítě rozdělen na úseky dle své geometrie

Vizualizací dat ze simulátoru získáme názornější porovnání dopravní situace při různých nastaveních simulátoru a můžeme ji tak reprodukovat na zjednodušeném modelu odpovídajícím skutečné síti dopravních komunikací.

¹<https://github.com/It4innovations/ruth>

1.2 Požadavky na aplikaci

V této sekci bych ráda blíže popsala požadavky pro vykreslování snímku dopravy pro jeden časový úsek. Tyto požadavky byly úzce diskutovány s vedoucími práce a jsou pro tuto práci klíčové. Jedná se o následující:

1. Vysoká intenzita dopravy vyjádřena rozšířením hrany.
2. Nižší intenzity dopravy vyobrazeny barevně.
3. Plynulé přechody mezi znázorněnými úseky.
4. Flexibilní konfigurace vizualizace.

Hlavní prioritou při vývoji bylo rozšiřování segmentů s velkým počtem aut, protože tato metoda dodává celkové animaci dynamiku. K velmi přeplněným dopravním uzlům takto přivedeme více pozornosti, než by tomu bylo u rozlišení pouze dle barevné škály.

Dopravní síť městských oblastí ale často neumožňuje dostatečný prostor pro rozšíření, které by lineárně zobrazovalo velké rozdíly mezi intenzitami frekventovaných ulic i segmentů s velice malým počtem aut. Pro tyto nižší segmenty jsme se proto rozhodli zvolit barevnou škálu. Toto rozhodnutí umožnilo snížit interval intenzit pro rozšíření a nabídnout tak mnohem příjemnější výsledky.

Jelikož kumulace aut na jednom segmentu není událostí nárazovou, bylo pro nás také důležité, aby přechod z jednoho stádia intenzity na druhé byl proveden plynule. A to jak v rámci jednoho segmentu cesty, tak mezi více segmenty či mezi více snímky animace.

Kapitola 2

Aktuální možnosti vizualizace

Zvolený způsob vizualizace z velké části ovlivňuje typ zobrazovaných dat. Dopravní datasety obvykle obsahují jak informace o poloze, tak i o čase. Jedná se tedy o data vícedimenzionální [3]. Vícedimenzionální data nám dávají možnost brát při analýze v potaz kombinace jednotlivých atributů a ne pouze atributy samotné, což by mohlo vykazovat zavádějící výsledky. Při velkém množství datových záznamů se stává komplexita dat spíše přítěží, jelikož jejich zobrazení je často velice nepřehledné. Využitím agregačních metod [4], můžeme tyto data seskupit a zmenšit, čímž docílíme zpřehlednění výsledné vizualizace. Pro provedení prostorové agregace dojde obvykle k spočítání hustoty dat v rámci segmentu (například reprezentace cesty hranou grafu) či oblasti. Časová agregace zobrazuje změny množství dat v závislosti na dané časové ose. Při kombinaci těchto dvou metod můžeme například zobrazit hustotu v rámci časových intervalů na různých segmentech mapy.

Různé velikosti časových i prostorových oblastí vybraných pro shlukování vedou k velice rozdílným zobrazením. Zvolená agregace pak také ovlivňuje typ vizualizace, který je vhodný s tímto typem agregace použít. Druhým možným způsobem je první zvolit vizualizační metodu a na základě této volby data agregovat. V obou případech je velice důležité brát v úvahu různorodou povahu dat, která se snažíme zobrazit.

2.1 Typy vizualizací

V této podkapitole se zaměříme na způsoby vyobrazení dopravních dat, která jsou častá v rámci vědeckých studií. A. Clarinval a B. Dumas [5] provedli analýzu 146 článků na téma vizualizace dat městské dopravy publikovaných ve vědeckých časopisech a konferenčních sbornících vydaných v období 2008 - 2022. Cílem jejich studie bylo porovnat zvolené zdroje dat, adresované domény, vizualizační techniky a typy cíleného koncového uživatele.

Z dat vyplývá, že 96 % zkoumaných studií použilo metodu geoprostorové vizualizace pomocí mapových souřadnic a 86 % k tomu zvolilo 2D vizualizaci oproti 3D vizualizaci. V tabulce 2.1 můžeme vidět přesné rozložení použití typů vizualizace z hlediska typu vztahu mezi hodnotami a dimenze

vyobrazení. Nejpoužívanější metodou z hlediska typu vztahu hodnot se ukázala vyobrazení kvantitativní. Do této kategorie patří vztahy intervalové a poměrové. Jedná se o data, která můžeme navzájem numericky porovnávat, například počet aut na segmentu nebo počty osob přesouvajících se z jedné oblasti do druhé. Dimenzionální zobrazení byla rozdělena na data definovaná plochou, křivkou či body. Jejich frekvence použití byla rostoucí v tomto pořadí.

Je třeba podotknout, že 115 ze 146 studií cílilo primárně na uživatele s předchozími znalostmi v oblasti mobility. Většina zobrazovaných dat také byla statického charakteru a vizualizace se neměnila na základě nově získaných dat.

Tabulka 2.1: Frekvence mapových typů ve zkoumaných článcích [5]

Typ dat	Bodová	Čárová	Plošná
Nominální	Dot map (34)	Network map (26)	Colored area map (6)
Ordinální	Symbol map (1)	Ordered network map (11)	Ordered colored map (0)
Intervalová nebo poměrová	Graduated symbol map (80)	Flow map (67)	Choropleth map (18)

Charakterizováním typů vizualizace se ve své práci zabýval také W. Chen a kolektiv [3]. Dle jejich návrhu by se daly vizualizace dopravních dat kategorizovat do 4 skupin dle zobrazovaných proměnných. Prvním typem jsou data časová, vyobrazující změny v čase u pozorovaného dopravního subjektu. Dále data prostorová, která popisují umístění subjektu v prostoru. Třetím typem jsou data časoprostorová, která kombinují předchozí dva typy. Poslední kategorií jsou data multivariační, která nezapadají ani do jedné z předchozích kategorií. Do této kategorie patří například data numerická, kategorická či textová. Jelikož data generovaná simulátorem udávají, jak údaje o umístění vozidla v prostoru v případě jednoho snímku, tak změnách v čase při zakomponování snímků do animace, pojďme si nyní představit nejpoužívanější metody u těchto kategorií.

2.1.1 Názvosloví geoprostorové vizualizace

Dle kartografického názvosloví [6] se *mapa, která na podkladu základní mapy, popř. na redukovaném podkladu základní mapy nebo obecně geografické mapy zobrazuje další přírodní, sociálně-ekonomické a technické objekty a jevy a jejich vztahy nazývá tématická mapa*. Tyto mapy se pak dále dělí na kartogramy a kartodiagramy, jejichž bližší vlastnosti popsal například J. Kaňok [7].

Kartogram je mapa zobrazující kvantitativní data přepočtená na jednotku plochy dílčího úseku. Vizualně se zobrazuje pomocí polygonů pro každý dílčí úsek. Polygon je dle své relativní hodnoty vybarven určitou barvou, odstínem či šrafováním. Existuje mnoho typů kartogramů, například homogenní, vztahové, tečkové, čárové nebo prostorové. Je třeba si také dávat pozor při překladu, jelikož v angličtině se termín **cartogram** používá pro tématickou mapu, u které došlo k nafouknutí

či zmenšení dílčích úseků na základě jejich numerické hodnoty. Termín kartogram se překládá jako *choropleth map*, jak je vidět v tabulce 2.1 výše.

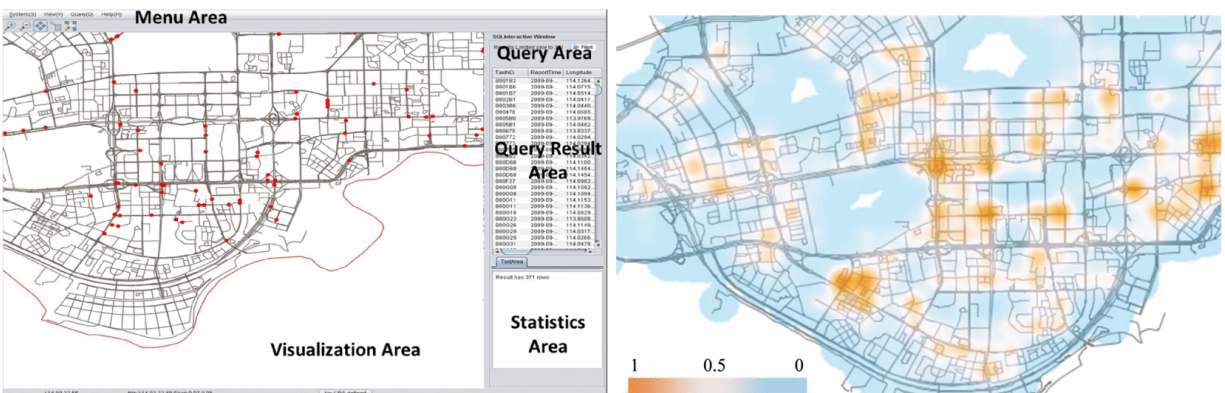
Kartodiagram je mapa vyjadřující absolutní hodnoty pozorovaného jevu. Podobně jako kartografické znaky mohou být diagramy znázorněny body, liniemi nebo plošně. U kartodiagramů J. Kaňok považuje za velmi důležité zvolit vhodnou objektivní stupnici pro data v celé mapě.

2.1.2 Metody zobrazení dat v prostoru

Bodové grafy definují geografické zobrazení, kdy je každá zaznamenaná událost pozorovaná jako umístěný bod v mapě [3]. Tento bod reprezentuje polohu snímaného objektu či jevu. Může se jednat například o lokaci auta či chodce nebo místo dopravní nehody [8].

Výhodou této vizualizační metody je možnost pozorování každého objektu zvláště s velice přesnými souřadnicemi. Tyto souřadnice by měly být ve středu promítaného bodu bez ohledu na jeho velikost. Výjimkou může být situace, kdy se snažíme umístěním zdůraznit polohu objektu (umístěním přístavů blíže k vodní oblasti) nebo pokud by došlo k zakrytí důležitého orientačního bodu (například zálivu či mola) [7].

Při velkém množství může nastat přehlnění grafu, což způsobuje pomalé vykreslování, nepřehlednost a zbytečné překrývání jednotlivých bodů. Tento problém může být vyřešen změnou velikosti bodů, jejich průhledností nebo redistribucí dat [9].



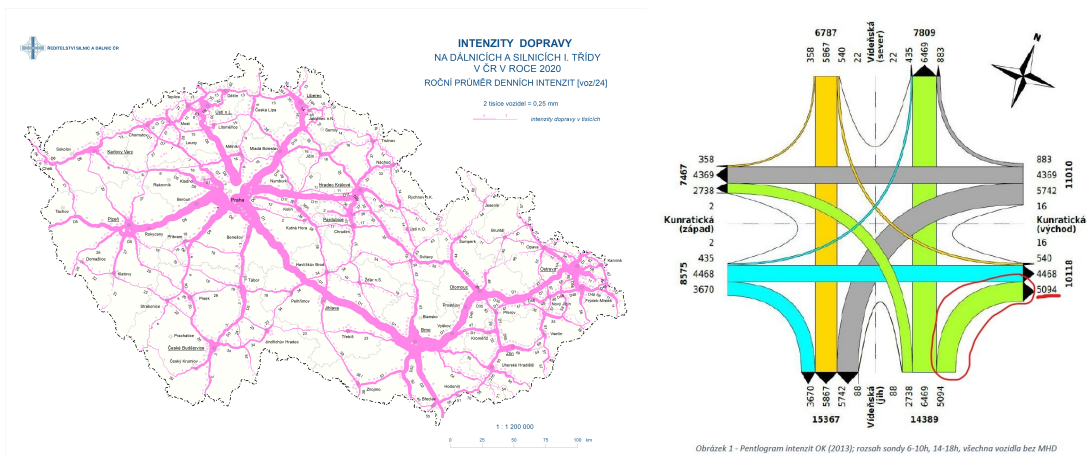
Obrázek 2.1: Jednoduchá bodová mapa (nalevo) a teplotní mapa (napravo) [10]

Mezi bodové mapy patří také **teplotní mapy**. Jejich porovnání je viditelné na obrázku 2.1. Zobrazení vychází ze studie L. Siu a kolektivu [10] zkoumající pohyb 15 000 snímaných vozů taxislužby. U klasické bodové mapy je zaznačena přesná poloha vozidel, u mapy teplotní tuto možnost nemáme. Nabízí nám naopak celkový přehled o četnosti navštívení různých oblastí města. Teplé barvy vyjadřují místa s vysokou intenzitou pozorovaného jevu a postupným přechodem se dostávají do oblastí studené barvy s nízkým výskytem událostí. Pro vytvoření teplotní mapy se využívá například jádrový odhad hustoty, který zajišťuje plynulý přechod mezi jednotlivými oblastmi s rozdílnou hustotou dat [11].

Čárové grafy jsou skvělým řešením pro vyobrazení trajektorií objektů či struktury dopravní sítě. Data jsou u těchto grafů reprezentována liniemi mezi dvěma určujícími body nebo hranami grafu. Pro zvýšení přehlednosti grafu často dochází k sloučení hran určených stejnými body, čímž se zvyšuje hodnota na dané hraně.

U map využívajících trajektorie hrají nejdůležitější roli startovní a cílová destinace. Geometrie čáry, která tyto destinace spojuje, nebývá určena. Pro rychlé a jednoduché zobrazení se proto používá spojení destinací přímkami. Zakřivením hran grafu - mapujícím zakřivení země či hranic kontinentů - lze naopak docílit větší čitelnosti popisků na mapě, rovnoměrného rozptýlení po ploše a zamezit protnutí jednotlivých hran [12].

Významným druhem liniového diagramu je **pentlogram**. Pentlogram je dle ČSN [13] označení pro *grafické vyjádření zjištěných a návrhových intenzit dopravy*. Geometrie dopravní sítě je v tomto případě velmi důležitá, proto nedochází k její úpravě. Tento druh kartodiagramu se často využívá pro zobrazení vnitrostátní hustoty dopravního toku, jako můžeme vidět na obrázku 2.2 (vlevo). Má ale využití i při analýze městské hromadné dopravy či zatížení křižovatek. Intenzita dopravy, měřena jako počet aut na daném segmentu dopravní sítě za určitý čas, je zobrazována pomocí barevného spektra [14] či rozšířením segmentu. Každý segment může být navíc popsán numerickou hodnotou intenzity pro upřesnění. Všechny tyto metody byly použity pro vytvoření pentlogramu na obrázku 2.2 (vpravo) pro vyobrazení dopravní situace na křižovatce z Vídeňské směr Praha-Chodov [15].



Obrázek 2.2: Pentlogram ČR 2020 [16] a pentlogram křižovatky Praha-Kunratice [15]

Plošné grafy jsou poslední variantou zobrazení prostorových dat. Zároveň se jedná o nejméně používanou variantu pro analýzu dat městské dopravy. Tento typ diagramů se více hodí použít, pokud je zkoumaná oblast jednoduše rozdělena do regionů. Těmito regiony mohou být administrativní oblasti nebo například předdefinovaná mřížka. [5].

2.1.3 Metody zobrazení dat v čase

Pokud by cílem naší vizualizace bylo pouze zaznamenávat změny hodnot celkových intenzit za určitou dobu, mohli bychom využít lineární, periodické anebo rozvětvené zaznamenání času. Při těchto zobrazeních je čas ve vizualizaci reprezentován vlastní dimenzí. Tato reprezentace může být posun po ose Y, kruhové zobrazení 24 hodinového cyklu nebo teplotní mapou ve formě kalendáře [3, 14]. Jelikož si ale přejeme zahrnout i geoprostorový charakter dat, doporučuje se využít jednu ze tří níže zmíněných typů vizualizace.

Animace je nejčastějším způsobem zobrazení času v kombinaci s 2D architekturou. [5] Využití pohyblivého výstupu pro zobrazení dat má mnoho výhod. T. Höffler a D. Leutner [17] ve svém průzkumu prošli 26 studií porovnávajících animované a statické vizualizace. Tyto studie se zabývaly zapamatovatelností zobrazené informace. Ve výsledku bylo ve více než 71 % případů zapamatování informací z dynamické vizualizace pomocí videa či animace lepší než u dat zobrazených staticky.

Höffler a Leutner ve své studii také podotkli, že animace se stává účinnou pomůckou pouze v případě, kdy je využita efektivně. Ve studiích, kde byla role animace pouze estetická, byly její výsledky na podobných hodnotách jako tomu bylo u statických obrázků.

Pro vizualizaci vývoje dat v čase je použití animace velice příhodné. Postupnou změnou zobrazovaných dat je získáván pomyslný třetí rozměr a není nutné data příliš shlukovat. Tato metoda je ve studiích také používána v kombinaci s 3D vizualizací [18] či virtuální realitou [19]. Jelikož je časová složka dat vyobrazena dynamicky, je možné využít třetí dimenzi prostoru pro jiný datový atribut, například hloubku či intenzitu.

Space-Time-Cube (STC) je společně s jinými 3D vizualizačními metodami druhým nejčastějším řešením vykreslení času s geodaty [5]. Při této vizualizaci jsou data reprezentována v rámci třírozměrného souřadnicového systému. Dvě ze souřadnic reprezentují prostorovou složku dat a mapují je na podkladovou mapu oblasti. Třetí souřadnice znázorňuje polohu dat na časové ose. Posunem po této ose se v prostoru STC orientujeme v rámci jednotlivých časových rovin. Pokud se stejný objekt nachází ve více časových rovinách, můžeme body znázorňující pohyb objektu spojit a vytvořit tím časovou stopu objektu. Dalším možným způsobem vizualizace je kombinace s barevným mapováním, kdy jsou jednotlivé oblasti zbarvené dle kategorické hodnoty dat [3, 20].

Výhodou využití 3D vizualizace je možnost zobrazení geoprostorové i časové složky dat najednou ve velmi intuitivní formě. Zvláště pro vizualizace jedné trajektorie a její změny v čase je tato metoda velice praktická. Nevýhodou může být nutnost konverze 3D metody pro 2D projekci obrazovek nebo při využití v tištěných materiálech, při této konverzi je zobrazování velkého množství dat často nepřehledné.

Time-flattening je termín pro grafické znázornění času využitím barvy nebo velikosti symbolů. Savitha Baskaran a kol. [21] ve své práci reprezentovali čas jako parametrickou křivku skrz barevné spektrum. Toto spektrum je definované pomocí HSV modelu třemi složkami: odstínem, sytostí barvy a jasem. Pro lineární mapování hodnot se využívá přímky. Křivka ale může mít i tvar kruhu či

spirály. Využití kruhu je doporučeno pro vizualizaci změn v uzavřeném časovém intervalu, například pro průměrnou intenzitu v hodinách dne. Spirálovou metodou zvyšujeme navíc jas s přibývajícím časem, což je vhodné pokud potřebujeme od sebe odlišit různé dny, ale chceme ponechat stejný odstín barvy pro stejnou hodinu v rámci těchto dní.

V praxi se využívá takto získané barvy ke zbarvení bodu na mapě. Pro plynulé přechody mezi jednotlivými datovými body se využívá barevná interpolace ke zbarvení celého segmentu. Použití velikosti symbolů či barvy však není pro zobrazení času příliš obvyklé, jelikož toto vizuální odlišení se používá pro jiné vlastnosti dat, jako je hustota datového toku.

2.2 Dopravní informace na mapových portálech

Mapové portály patří k nejčastějším místům, kde se běžný uživatel setká s vizuální reprezentací dopravní situace. Mezi nejpoužívanější z nich patří Google Maps¹, Naver mapy² a Baidu mapy³. Nejvíce obyvatel České republiky používá stránku Mapy.cz⁴ [22]. V následujících kategoriích se podíváme v jakých aspektech vizualizace dopravy si jsou tyto stránky podobné a kde se naopak liší.

2.2.1 Symboly pro znázornění

Trend ohledně typu znázornění dopravní situace je vcelku jednotný. Tato skutečnost není překvapivá, jelikož cílem portálu je předat informace uživateli co nejefektivněji a to nejlépe způsobem, který již uživatel zná nebo mu připadá intuitivní.

Pro zobrazení dopravní mapy je možné využít tlačítka vyobrazující semafor (Naver, Baidu) nebo volby mezi mapovými vrstvami (Google, Mapy.cz). Po vybrání se uživateli přes klasickou mapu zobrazí čárový graf pro jednotlivé úseky cest. Při oddálení se zobrazují pouze hlavní dopravní úseky v dané oblasti, přiblížením mapy pak dosáhneme podrobnějšího pohledu. U každého úseku jsou vyobrazeny dvě křivky lemující geometrii vozovky. Toto rozložení umožňuje rozdělení intenzity dopravy dle směru, i když při velkém oddálení není zcela geograficky přesné. Šířka čáry je buď konstantní nebo poměrově odpovídá šířce vozovky. Samotné rozlišení stupňů dopravních situací je zbarvením křivek dle barevné škály. K této informaci je na většině mapových portálů možné zobrazit dodatečné vysvětlující symboly ohledně míst dopravních nehod či staveb na komunikaci.

2.2.2 Stupně provozu

Určování stupňů provozu na mapových portálech je obvykle velice vágní. Pro příklad můžeme použít server Google Maps. Legenda na spodním okraji obrazovky zobrazuje 4 barevné pruhy určující škálu od provozu, který je "Rychlý" po provoz "Pomalý", jak je u škály uvedeno. Nápověda portálu [23]

¹<https://www.google.com/maps/>

²<https://map.naver.com/>

³<https://map.baidu.com/>

⁴<https://mapy.cz/>

dále upřesňuje, že označení cest je rozděleno do tří barevných kategorií. Zeleně označené jsou úseky bez zaznamenaného zpoždění v důsledku zpomalení dopravy, oranžová barva se používá pro střední zpomalení a červená pro zásadní zpomalení dopravy. Čím tmavší odstín červené, tím větší dopravní zácpa. S touto škálou se můžeme setkat i na mapách Baidu a Mapy.cz.

Mapy Naver také používají zelenou, žlutou a červenou pro rozlišení provozu. Při najetí myši na škálu ale nabízí i doplňující informace ohledně těchto kategorií. Zelená barva určuje úseky, kde rychlost dopravy neklesla pod rychlost určenou kategorií cesty. Pro silnice v městských částech je tato rychlost přes 30 km/h. U dálnic se hranice posouvá až na rychlost přes 70 km/h. Pokud je rychlost aut na úseku menší než rychlost kategorie cesty, pak je úsek zbarven oranžovou. Při poklesu rychlosti pod polovinu rychlosti kategorie dojde k přebarvení do odstínu červené. Například u městských částí k tomu dochází při rychlosti nižší než 15 km/h.

2.2.3 Systém úrovní NDIC

Národní dopravní informační centrum (NDIC) [24] je specializovaným pracovištěm pro Ředitelství silnic a dálnic České republiky (ŘSD ČR) zajišťující sběr, zpracovávání a distribuci dopravních informací a dat o aktuální dopravní situaci. Tyto informace jsou získávány pomocí telematických systémů - dohledových kamer, detektorů intenzit dopravy, silničního meteorologického systému a dalších. Odtud se prostřednictvím datového distribučního rozhraní data dále šíří k rozhlasovým a televizním stanicím, přeprávcům, telekomunikačním operátorům a ostatním. Dříve zmíněný mapový portál Mapy.cz také získává dopravní informace z NDIC.

Škála intenzit využívaná ŘSD ČR je mnohem specifičtější než je tomu u klasických mapových portálů. Například při vytvoření interaktivní mapy celostátního sčítání dopravy 2020 [16] se drželi následující legendy. Data byla rozdělena do 4 barevných kategorií - zelené, žluté, fialové a červené - dle počtu vozidel za 24 hodin. V každé z těchto kategorií byl navíc využit styl čáry pro větší rozdělení intervalů. Tohoto rozdělení bylo nejvíce potřeba v kategorii červené, kam spadaly všechny úseky s více než 10 000 aut za 24 hodin. Podobný systém byl také implementován na dopravním portálu ŘSD ČR [25] do tří barev (modrá pro stupeň 1-2, žlutá pro stupeň 3 a červená pro 4-5). Tyto stupně jsou závislé na počtu snímaných vozidel na daném segmentu za 24 hodin.

Kapitola 3

Srovnání vizualizačních knihoven

Nyní se zaměříme na knihovny nabízející možnosti vizualizace jak obecných, tak přímo geoprostorových dat. Pro tuto práci jsem zvolila knihovny napsané v jazycích python a javascript.

Javascript patří k hlavním programovacím jazykům používaným v internetových stránkách a umožňuje tak jednoduchou distribuci výsledné vizualizace. Nevýhodou javascriptu je nedostatečná podpora pro rychlé zpracování dat. Pro jeho využití je tedy potřeba předzpracování dat v rámci jiného programovacího jazyka [26].

V pythonu je dostupná infrastruktura pro analýzu dat naimlementovaná v jazyce C. Často využívaná je v tomto odvětví knihovna pandas, která obsahuje mnoho funkcí pro práci s daty. Tato data lze zobrazovat i přímo v rámci pythonu pomocí knihoven zmíněných níže. Pro jednoduché a interaktivní webové zobrazení snímků se často využívá například aplikace Jupyter Notebook [27], umožňující vytváření dokumentů obsahujících jak zdrojový kód popisující manipulaci s daty, tak výsledné tabulky, grafy či diagramy.

3.1 Matplotlib

Matplotlib [28] patří mezi standard python knihoven pro vizualizaci dat. Motivací pro vznik této knihovny byla potřeba ekvivalentu programového prostředí Matlab kompatibilního s knihovnami jazyka python. Matplotlib nabízí 31 různých typů grafů, mezi které patří i základní vykreslování křivek (funkce `plot`), bodů (funkce `scatter`) a polygonů (funkce `bar` nebo `fill_between`). Kromě zobrazování statických vizualizací a jejich exportu knihovna také nabízí generování animací včetně ukládání v různých formátech. Nedílnou součástí jsou i interaktivní vizualizace s využitím tlačítek a posuvníků.

Kromě relativně jednoduchého používání, umožňuje tato knihovna i rozsáhlé možnosti vlastní konfigurace. Výchozí parametry jsou uloženy v globální slovníkové struktuře a dále využívány u vykreslovacích funkcí bez parametrů udávajících styl vizualizace. Mezi parametry, které je možné upravovat, patří například šířka, barva a styl čáry, offset vykreslení, DPI (počet pixelů na délku jed-

noho palce), přiblížení či nastavení popisků grafu. Konfigurace lze změnit také dlouhodobě pomocí konfiguračního souboru nacházejícího se v pracovním adresáři.

Pro jednodušší práci s barvami při vizualizaci lze využít škálu vestavěných barevných palet. Mezi tyto palety patří například palety celého HSV spektra, palety imitující povrch země či palety skládající se pouze z odstínů jedné barvy. Pokud by si uživatel nevybral z nabízených typů palet, nabízí knihovna funkcionalitu pro vytvoření palet vlastních.

3.2 GeoPandas

GeoPandas [29] se označuje jako projekt pro jednoduchou práci s geoprostorovými daty. Jedná se o rozšíření knihovny pandas, používanou pro datovou analýzu. I když knihovna používá vykreslovací funkce knihovny Matplotlib, používání této knihovny pro velká data geoprostorového charakteru je mnohem jednodušší a efektivnější. Funkce pro vykreslení se volá na vytvořeném datovém objektu GeoSeries a soustava os `ax` knihovny Matplotlib se posílá do této funkce jako parametr. GeoSeries může obsahovat data různých datových typů. Jedná se o Polygon, Multipolygon, LineString, MultiLineString, Point a MultiPoint. Funkce na vykreslování grafu proto musí obsahovat funkcionalitu pro vykreslování každého z těchto typů. GeoPandas vytvoří objekty pro vykreslení ve formátu Matplotlib kolekcí. Na konci funkce, po vytvoření všech kolekcí a jejich přidání do snímku, dojde k samotnému překreslení aktuálního snímku. Od roku 2021 byla do knihovny přidána možnost vykreslování pomocí funkce `explore`. Tato funkce zobrazuje data na interaktivní podkladovou mapu skrz knihovnu folium. Folium umožňuje snadný přístup přes python pro vizualizaci interaktivních map. V GeoPandas funkci `explore` dochází k transformaci parametrů pro potřebu funkce folium, vykreslí se podkladová mapa v daném výřezu a data se přidají do vizualizace jako GeoJson objekt.

3.3 OSMnx

OSMnx [30] je balíček pro práci s daty z portálu OpenStreetMap. Funkcionalita OSMnx je postavena na balíčku NetworkX, zabývající se manipulací a dynamikou komplexních sítí, a balíčcích GeoPandas a Matplotlib, zmíněných výše.

Hlavní výhodou pro vykreslování map s tímto balíčkem je zapouzdření komunikace s OpenStreetMap Overpass API. V rámci svého modulu nabízí OSMnx pomocí této API stahování geoprostorových informací o veřejných objektech (jako jsou školy, obchody, parky nebo zastávky) ve formě GeoPandas GeoDataFrame. Dále je také možné využití Overpass API pro získání informací o dopravní síti tvořené ulicemi, cestami nebo i kanály a jejich následném uložení jako síťový graf. Tento způsob konverze umožňuje příjemnější způsob práce s OpenStreetMap daty, možnost využití dat ve formátu grafu a okamžité vykreslování.

OSMnx nabízí dvě základní přístupy k vykreslování. Prvním z nich je nahrazení síťového grafu tabulkou GeoDataFrame a následné použití funkcionality GeoPandas pro zobrazení dat. Kromě

standardních možností konfigurace nabízí knihovna i možnost vykreslení šířky komunikací na základě typu cesty. Druhým způsobem je vykreslování trajektorií v rámci dopravní sítě. Tato možnost umožňuje definovat cestu pomocí NodeID křižovatek, skrz něž daná cesta vede, v případě, že jsou tyto křižovatky přímo spojené hranami grafu. Samotné vykreslení cest zajišťuje balíček Matplotlib dle souřadnic získaných z atributu `geometry` na hraně multigrafu.

3.4 D3.js

D3.js je knihovna pro vizualizaci dat na webu napsaná v jazyce javascript. Hlavním cílem knihovny je datově řízená modifikace objektového modelu dokumentu (DOM) s kompatibilitou s webovými standardy, jako je HTML, SVG a CSS. Díky tomu je možné na grafech vytvořených pomocí D3.js například jednoduše použít vlastní kaskádové styly. Kromě statických vizualizací D3.js nabízí i interaktivní prvky a plynulé přechody při změnách.

Knihovna nenabízí přímo funkce na vytváření jednotlivých typů grafů, jako je tomu například u knihovny Matplotlib. Místo toho umožňuje uživateli vytváření SVG elementů v cyklu na základě typu použitých dat. Například pro vytvoření bodového grafu je třeba definovat vlastní funkci vytvářející SVG elementy pro kruhy, popisky a podobně. K vykreslování grafů se tak často využívá elementu `path` pro vytvoření stejnojmenného SVG elementu. V D3.js na tomto elementu lze nastavit atribut `d` definující funkci k získání vykreslovaných souřadnic z dat. Při geoprostorových vizualizacích lze využít i element `GeoPath`, který bere v potaz styl geografické projekce, a je vhodný pro vykreslování GeoJson objektů [31].

I v ohledu konfigurace vizualizačních parametrů zvolila knihovna D3.js datově řízený přístup. Místo konstantních hodnot může uživatel nastavit hodnoty dynamicky s využitím funkcí. Tohoto přístupu lze využít na změnu nastavení barvy, šířky čáry, výplně útvarů, stylu zakončení čar a pod.

D3.js tak nabízí velký prostor pro vlastní úpravy a nastavení vytvořených grafů společně s nezávislostí na konkrétním aplikačním rámci. Proces vytváření samotných grafů ale vyžaduje velké množství kroků a může být pro začátečníky nepřehledný. Výhodou v tomto ohledu je možnost čerpat z mnoha ukázek kódu a šablon pro často používané typy grafů vytvořených velmi širokou uživatelskou základnou v rámci galerie na serveru Observable [32] či jiných internetových stránkách.

3.5 Plotly

Knihovna Plotly.js [33] také patří mezi vizualizační knihovny jazyka javascript. Z Plotly.js dále vznikly i moduly pro jazyk python, R a další, umožňující využití společně s knihovnou pandas, v rámci Jupyter Notebooku nebo jednoduchých HTML stránek. Plotly.py navíc syntaxí připomíná často používanou knihovnu Matplotlib.

Narozdíl od D3.js, ze kterého Plotly.js vychází, je Plotly.js především knihovna na vytváření grafů. Plotly.js nabízí funkcionalitu pro vykreslení více než 30 různých druhů grafů. Vykreslovací

funkci je pouze potřeba poskytnout položky layout (definující popisky a obecný vzhled grafu) a data (určující mód vykreslení). Nejčastěji se tato data formátují do formy listu slovníků, kde každý slovník obsahuje parametry pro vykreslení jednoho obrazce. Knihovna také obsahuje interaktivní prvky a animace. Velice uživatelsky příjemná je možnost animací nejen pomocí funkce měnící data, ale i plynulé přechody mezi předem definovanými snímky.

Přízpůsobení opět nabízí změny barvy i šířky výplně a ohraničení, dále je k dispozici nastavení různého typu čáry. V celku unikátní vlastností je jednoduché povolení různých druhů interpolace křivek, kdy je uživateli umožněno napojit body jak přímkou, tak různými pravouhlymi uskupeními či plynulou křivkou.

Součástí knihovny je i vykreslování map. Mapy se vykreslují ve vrstvách, které se dělí do 3 základních typů: podkladové mapy, datové vrstvy a vrstvy překrývající. Rastrové i vektorové podkladové mapy jsou zprostředkovány skrz veřejné mapové servery, mezi které patří například OpenStreetMap, Carto nebo Stamen. Pro použití vlastní mapy se pro snazší použití doporučuje využití překrývající vrstvy s nastaveným atributem pro vykreslení pod daty.

3.6 Zhodnocení

Všechny knihovny obsahují základní funkce pro vykreslování bodů, čar či vybarvování plošných útvarů v různých barvách a s různými šířkami. V čem se funkce liší je formát dat zpracovávaných pro tyto vizualizace. Data ze simulátoru jsou evidována pod OSM id uzlů dopravní sítě a pouze knihovna OSMnx pracuje s tímto formátem. Ostatní knihovny lze napojit na OpenStreetMap API, případně využít jiné knihovny pro transformaci na příjemnější formát.

Tabulka 3.1: Srovnání knihoven

	Matplotlib	GeoPandas	OSMnx	D3.js	Plotly
bodové, čárové a plošné útvary	✓	✓	✓	✓	✓
složitější typy vizualizace	✓	ne	ne	✓	✓
interpolovaná šířka čáry	ne	ne	ne	ne	ne
možnost detailní podkladové mapy	ne	✓	✓	ne	✓
vizualizace cest dle OSM id	ne	ne	✓	ne	ne
ukládání snímku	✓	✓	✓	ne	✓
animace a interaktivita	✓	ne	ne	✓	✓
ukládání animace	✓	ne	ne	ne	ne

Způsob vykreslování cest knihovny OSMnx ale probíhá cyklicky pro každou cestu. Mnohem lepším přístupem tento problém řeší knihovna GeoPandas využívající vykreslování všech cest najednou

pomocí kolekcí. Obě tyto knihovny se ale zaměřují spíše na manipulaci s daty než na konfigurace vizualizovaných útvarů a nenabízí mnoho možností v tomto ohledu.

Knihovny D3.js i Plotly umožňují jednoduchou integraci pro web, plynulé přechody mezi šířkami a barvami v rámci animací a rozsáhlou škálu typu grafů. Plotly navíc nabízí integrované ukládání snímků do adresáře a stejně jako GeoPandas a OSMnx vykreslení různých typů podkladových map. Oproti D3.js je také jednodušší na použití. Obě knihovny ale nejsou stavěné pro exportování animací ve formě gifů či videí. Pro vytváření videa by bylo třeba jejich kombinace s jinými nástroji.

Bohužel žádná z knihoven nenabízí funkcionalitu interpolované šířky čáry, u které by docházelo k plynulému přechodu mezi různými šířkami v jednom snímku. Na základě porovnání popsaného v tabulce 3.1 a s ohledem na rychlost vykreslování, kterou měřila ve své práci Sofia Michailidu, jsem proto zvolila pro základ své práce knihovnu Matplotlib. Knihovna nabízí rozsáhlé úpravy vykreslování a přímé ukládání snímků do videoformátu. Pro splnění požadavků komponenty bylo třeba dále zajistit transformaci vstupních dat z OSM id na geosouřadnice a samotnou vizualizaci plynule navazujících segmentů. Znalost toho, jak s knihovnou Matplotlib pracují jiné vizualizační knihovny jako GeoPandas a OSMnx, mi byla při implementaci velice přínosná.

Kapitola 4

Návrh řešení

V této sekci si představíme potencionální řešení jednotlivých požadavků na implementaci vykreslování segmentů. Mapu získáváme skrz OpenStreetMap API v podobě grafu definovaném segmenty, jakožto hranami grafu znázorňující cesty, a vrcholy neboli křižovatkami, které jsou těmito segmenty propojeny. Každý segment má dále informace o své geometrii udané jako list zeměpisných souřadnic. Správným nastavením barvy (vyjadřující rozdíl nižších intenzit dopravy) a šířky segmentu vytvoříme kartodiagram intenzit dopravy.

Při návrhu vizualizace byl kladen důraz na splnění všech požadavků popsaných v kapitole 1.2. Segmenty s nižší intenzitou dopravy budou reprezentovány pouze barevným spektrem a rozšíření bude aplikované pouze u segmentů s intenzitou vyšší. Přechody mezi rozdílnými intenzitami musí být plynulé. Komponenta musí navíc umožnit flexibilní konfiguraci vizualizace.

4.1 Vykreslení barvy

Metody zvážené pro zbarvení segmentu vycházely jak ze způsobu využití barevného značení na mapových portálech, tak z rozdílných metod kombinovaných s kartodiagramy vědeckých studií. Mezi nejčastější barevné přechody patřily jednoznačně zelená-oranžová-červená symbolizující barvy semaforu a modrá-oranžová-červená jako teplotní přechod. Často využívaná byla také žlutá - jako stupeň přechodu k oranžové nebo její kompletní náhrada. Tyto barevné přechody se staly přirozenými kandidáty pro implementaci v rámci této práce. Pro zvolení rozmístění barev bych ráda uvedla následující dva způsoby.

4.1.1 Nastavení barvy na hraně grafu

První z nich vychází z technik užitých na mapových portálech. Jedná se o řešení, kdy je vytvořena stupnice úrovní o 3 až 5 barvách a ke každé barvě je přiřazen interval hodnot intenzity dopravy. Tento způsob vykreslování je podporován většinou vykreslovacích knihoven. Zároveň nabízí jasné spojení mezi barvou a intervalem hodnot a tím i snadné srovnání intenzit mezi dvěma rozdílnými

segmenty. Zvolením metody tří barev ale vznikají velmi silné přechody mezi jednotlivými segmenty. Zároveň nás to tlačí do nastavení jednotné hodnoty intenzity pro celý segment - což odporuje návrhu vizualizace s interpolovanou šířkou hran.

4.1.2 Použití mapování barev

Druhý způsob využívá interpolace dle geometrie hrany grafu. Každá hrana je rozdělena pomocí bodů na kratší rovné úseky. Pokud máme informaci o tom, jak se intenzita na segmentu mění, můžeme pomocí interpolace získat plynule se měnící hodnoty intenzit pro každý úsek segmentu. Tyto hodnoty jsou dále namapovány na RGB hodnoty zvoleného barevného gradientu. Využití barevného mapování společně s gradientem je také častou funkcionalitou. U některých funkcí umožňujících tento typ mapování je uváděno, že dochází k výraznému zpomalení při využití velkého množství dat. U testování dat ze simulátoru v kombinaci s funkcí pro vykreslení čar knihovny Matplotlib jsem toto zpomalení nezaznamenala.



Obrázek 4.1: Vykreslení hran s mapováním barev

4.2 Vykreslení rozšiřujících se hran

Dalším zásadním problémem je vyřešení vizualizace zvětšení šířky hrany pro znázornění dat. Většina knihoven nabízí jednotné nastavení šířky hrany pro všechny křivky, které jsou vykreslovány v jednom kroku. Toto řešení je pro naše požadavky nedostačující. V následující sekci se nejprve blíže podíváme na kritéria, která na implementaci rozšiřujících se hran máme a dále porovnáme způsoby řešení této funkcionality. U každého způsobu navíc posoudíme, jak návrhovým kritériím vyhovuje.

4.2.1 Kritéria

Pro porovnání různých způsobů vykreslení jsme stanovili následující 4 kritéria:

1. Šířka nezávislá na úhlu hrany.
2. Šířka nezávislá na přiblížení mapy.
3. Plynulost rozšiřování.
4. Rychlost vykreslení.

První dvě kritéria ovlivňují čitelnost dat a jejich jednoznačnost. Nemělo by docházet ke zkreslení dat v důsledku formátování grafu. Hrana, která má konstantní hodnotu intenzity po celé své délce, by měla mít i konstantní šířku na všech úsecích své geometrie i přechodech mezi nimi. Šířka vizualizovaná na dvou úsecích se stejnou intenzitou by měla být shodná a neovlivněná geometrií hran definujících tyto úseky.

Šířka by také měla být stejně určena u map zobrazujících malý výřez z mapy, i u map zobrazujících celé velkoměsto či větší oblasti. Rozšíření by mělo být vždy v poměru k ostatním prvkům mapy (například ku vzdálenostem mezi paralelními cestami či velikosti výřezu v rámci zeměpisných souřadnic). Správným zvolením škály hodnot pro tuto šířku zaručíme konzistentní hodnoty a zároveň přehlednost při daném zahuštění dopravní sítě.

Samotné rozšiřování segmentu by nemělo probíhat skokově. Pokud intenzita na začátku segmentu je výrazně rozdílná s intenzitou na konci, měla by tato změna být viditelná pomocí postupně se rozšiřující či zužující vyplněné křivky. Plynulé by mělo být také navazování segmentů. V ideálním případě by přechod mezi segmenty s rozdílnou intenzitou měl být co nejvíce podobný přechodu mezi intenzitami v rámci jednoho segmentu.

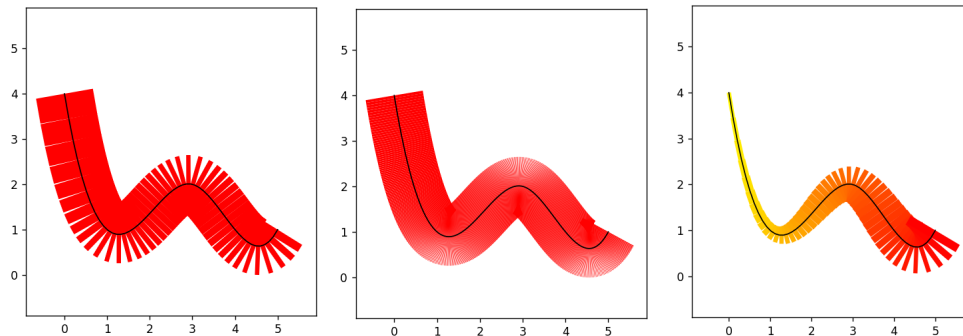
Jelikož výsledné snímky budou využity v rámci videa, je nezbytné zohlednit i rychlost samotného vykreslování. Generování videa vyžaduje uložení tisíců snímků a zrychlení vizualizace každého z nich může výrazně snížit celkovou dobu generování videa. K využití metod umožňujících přesnější či estetičtější výsledky je využito více výpočetního času, naopak využití rychlejšího způsobu vede ke ztrátám u jiných výše zmíněných kritérií. Výsledná volba přístupu tak musí být výsledkem rovnováhy mezi všemi kritérii.

4.2.2 Nastavení šířky na úsecích hrany

Jak bylo zmíněno u řešení barevných přechodů, každý mapový segment je dále rozdělen na menší části určující jeho zakřivení. Tohoto rozdělení se dá využít i pro řešení šířky. Využitím stejné techniky se podle seznamu intenzit v určitých místech nainterpoluje intenzita pro každý z těchto úseků a tato hodnota bude sloužit jako základní parametr pro výpočet šířky segmentu. Výpočet bude pro všechny segmenty a jejich úseky stejný a bude ovlivněn maximální možnou intenzitou, minimálním počtem

aut potřebných pro rozšíření, popřípadě i emulátorem šířky, který pomůže výsledek naškálovat přiměřeně k rozměrům mapy.

Problém s tímto řešením nastává při vyšších hodnotách rozšíření u zaoblených segmentů. Jak je ilustrováno na obrázku 4.2 vlevo, tvoří se na obloucích křivky zubatý okraj. I když černá křivka působí plynule zaobleně, stále se skládá z desítek krátkých okem nezaznamenaných úsečků. Pouhým rozšířením nedojde k vyplnění prázdných míst vzniklých zakřivením. To způsobuje vytvoření zubatého okraje.



Obrázek 4.2: Porovnání rozšíření přímky definované 50 body (vlevo), 500 body (uprostřed) a 50 body s gradientem (vpravo)

Důležitý je také správný způsob vykreslování. Rozdělením segmentů na desítky malých úseků roste mnohonásobně počet tvarů, které je třeba zobrazit. To může vést k značnému zpomalení programu. Pokud jsou jednotlivé úseky spojeny do kolekce a vykresleny najednou, jedná se o nejrychlejší z mých variant řešení. Rozšíření čar reprezentujících úseky segmentu navíc umožňuje jednoduchou integraci s využitím mapování barev.

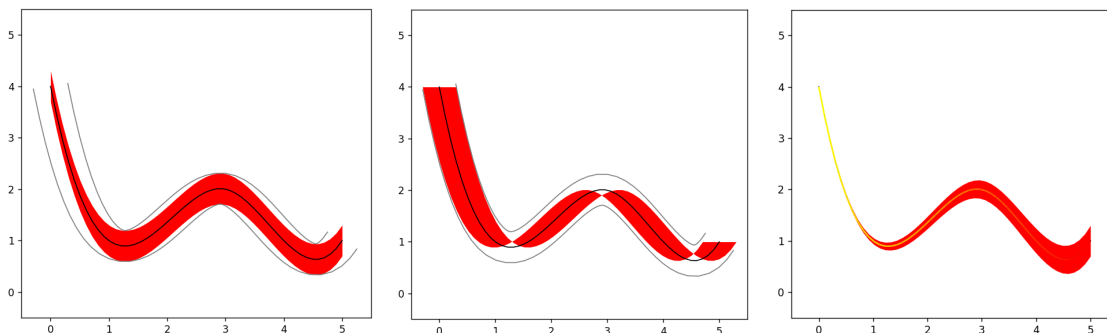
Šířka čáry se u většiny knihoven udává v bodech nebo pixelech. Velikost této jednotky v porovnání s osami grafu záleží na celkové velikosti grafu. Při přiblížení mapy tak zůstává šířka čáry v poměru k velikosti snímku nikoli ke měřítku mapy. Díky tomu je těžké předem odhadnout, na jakou hodnotu je třeba nastavit šířku maximální. V potaz musíme brát nejen přiblížení mapy, ale i rozměry snímku pro nahrávání, což výpočet komplikuje. Navíc toto řešení není vhodné pro vizualizace interaktivní, jelikož v tom případě by se šířka musela měnit při každé změně měřítka mapy.

4.2.3 Vybarvení polygonu mezi dvěma křivkami

Abychom se vyrovnali zadávání šířky v bodech, je možné provést rozšíření čáry vykreslením polygonu. Hranice polygonů se definují dle souřadnic na mapě, velikost polygonu je proto úměrná měřítku mapy a není důležité ji měnit v závislosti na velikosti vykreslovaného snímku. Dá se tak jednodušeji odhadnout hodnota pro rozšíření segmentu.

Jedním ze způsobů definování polygonů je vybarvení prostoru mezi dvěma křivkami. Tento způsob se využívá například u skládaného plošného grafu, kde je každá plocha definovaná dvěma křivkami se stejnou souřadnicí x , ale rozdílnými souřadnicemi y . Můžeme ale vidět i varianty se společnou souřadnicí y a rozdílnými souřadnicemi x , jako je tomu u houslových diagramů. Pro vykreslení polygonu určité šířky je potřeba posunu původní hrany grafu ve směru jedné z os souřadného systému. Takto získáme první okraj polygonu. Druhý sestrojíme posunutím křivky směrem opačným než tomu bylo v předchozím případě.

Sestrojení polygonů tímto způsobem je relativně rychlé, zvláště díky rychlému výpočtu okrajů polygonu. Metoda splňuje jak plynulý přechod mezi intenzitami na segmentu, tak nepřesahuje mimo segment a nepohlcuje geometrii segmentu. Bohužel lze tímto způsobem provést posunutí pouze v rámci směru jedné osy. Díky tomu dochází k modifikaci šířky u úseků rovnoběžných se směrem posunu. Jak můžeme pozorovat na obrázku, 4.3, pokud si zvolíme posun dle osy y , uvidíme, že čím svislejší úsek je, tím méně dojde k jeho rozšíření. A to i v případě, že má být na celém segmentu šířka konstantní. Při posunu dle osy x je tomu stejně u úseků vodorovných. Na obrázku 4.3 je očekávaná šířka zobrazena šedým ohraničením oproti vykreslenému červenému polygonu. Další komplikací je mapování barev na polygon, jelikož rozdělení polygonu na menší úseky by vedlo k výraznému zpomalení programu. Proto je možné vykreslit gradient pouze na křivce samotné a ne na jejím rozšíření.



Obrázek 4.3: Vykreslení rovnoměrné šířky posunem ve směru osy y (nalevo) a osy x (uprostřed) a dle osy y s gradientem

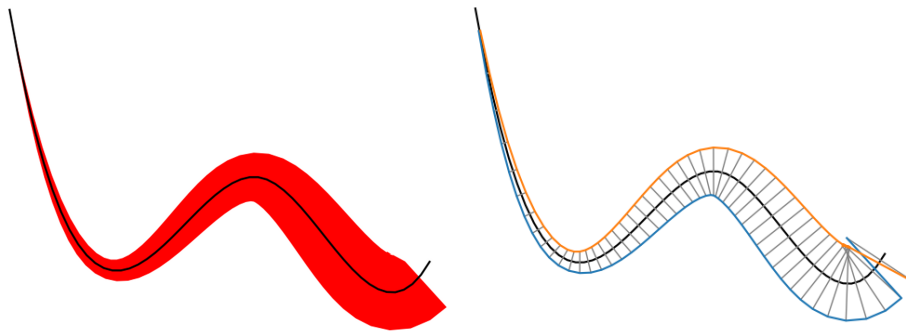
4.2.4 Vybarvení polygonu dle ekvidistanty

Abych se vyvarovala nerovnoměrného rozšíření na základě posunu, rozhodla jsem se u třetí metody definovat okraje polygonu pomocí ekvidistanty. Dle Ottova naučného slovníku [34] je ekvidistanta dané křivky K definována jako geometrické místo bodů, nalézajících se na normálách křivky K a stejně od ní vzdálených. Jedná se tedy o dvojici křivek s daným odsazením neboli vzdáleností od původní křivky.

Jednou z možností řešení pro získání souřadnic těchto odsazených křivek je využití výpočtu pomocí Minkowského sumy, kterou pro tuto funkcionalitu využívá například knihovna Shapely [35] ve funkcích `buffer` a `parallel_offset`. Tyto funkce ale neumožňují změnu velikosti daného offsetu v rámci jedné křivky. Pro výsledek vyhovující požadavkům bude potřeba jiného přístupu.

Ve svém řešení jsem se inspirovala metodou, kterou na svém webu uvedl Nikolay Falaleev [36]. Ten rozdělil původní křivku na sérii úseček. Počet úseček definuje počet výsledných vypočtených bodů ekvidistanty. Každá úsečka je definovaná svým sklonem k , vycházejícím s definice přímky $y = kx + b$. Normála k této úsečky má tedy sklon $-1/k$. Víme-li sklon normály, vzdálenost od úsečky a souřadnice jejího středu, můžeme snadno provést posun středu o daný vektor a dopočítat tak pozici jednoho z bodů ekvidistanty. Čím více úseček na křivce definujeme, tím zvýšíme počet bodů ekvidistanty a její definice bude tedy přesnější. Jelikož segmenty grafu dopravní sítě jsou již předem rozděleny dle své geometrie, rozhodla jsem se využít tohoto dělení.

Protože N. Falaleev pracoval se stejnou vzdáleností pro všechny body polynomiálních funkcí, bylo potřeba pro řešení v mé práci provést několik modifikací. V rámci implementace probíhá výpočet souřadnic bodu po posunutí pro každý bod zvlášť, můžeme proto tuto vzdálenost postupně zvyšovat s každým následujícím úsekem. Dále jelikož hrany našeho grafu nemají vlastnosti funkcí, může dojít u hodně strmých úseků k přetočení stran, posunutí a protnutí původní přímky s ekvidistantou. Tento jev je vidět i na obrázku 4.4. Aby se tomuto chování zabránilo, provádí se před ukončením funkce kontrola protnutí. Pokud by k protnutí došlo, dojde k výměně souřadnic bodů, které protnutí způsobují, za souřadnice posunuté křivky ve vzdálenosti d s opačným znaménkem. Na obrázku 4.4 nalevo také můžeme vidět, že při tomto řešení nedosahuje polygon úplnému okraji křivky. Pro nápravu stačí provést posunutí krajních bodů křivky o stejný vektor, o který se posouval jejich nejbližší střed úseku.



Obrázek 4.4: Výplň dle ekvidistanty (nalevo) a zobrazení normál (napravo)

4.3 Plynulý přechod mezi hranami

V předchozí sekci jsme docílili plynulého přechodu mezi jednotlivými šířkami na jedné hraně. V této sekci se budu zabývat napojením těchto hran na sebe. Pro správné napojení je důležité, aby konce všech hran, které se napojují na společný vrchol, měly stejnou hodnotu intenzity. Pokud ke sjednocení nedojde, budou ve výsledné vizualizaci nevyhnutelné skoky. Jedním z možných řešení může být nastavení hodnoty intenzity na konci segmentu na nulu. Navazování by pak bylo plynulé, ale na každé hraně s rozšířením by docházelo ke vzniku kopcovitých útvarů. Tento typ vizualizace nesprávně naznačuje, že na hraně vždy dochází ke kumulaci aut uprostřed segmentu.

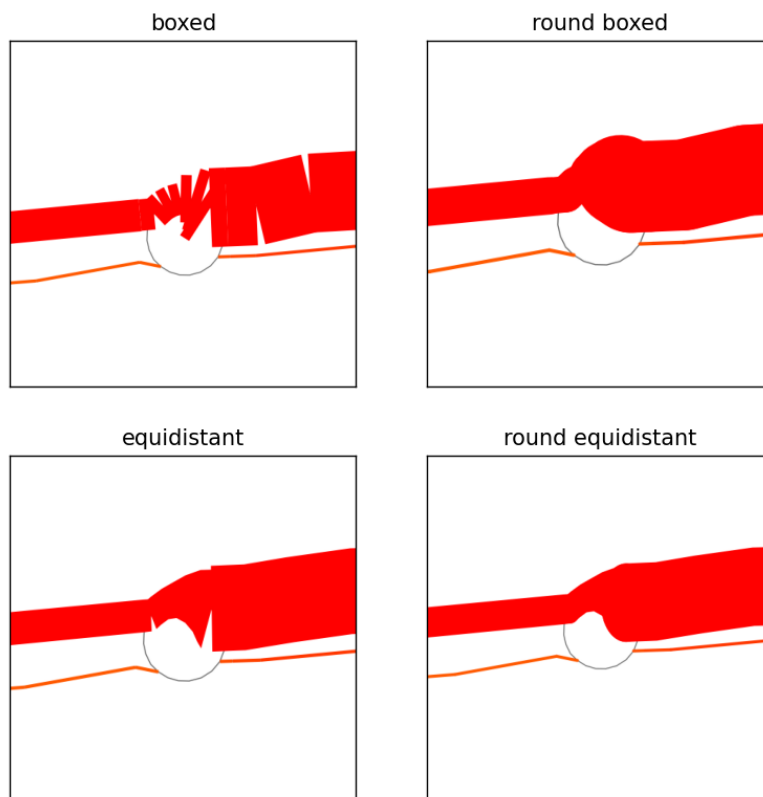
Další možností může být nastavení stejné hodnoty intenzity pro vrchol na základě intenzit segmentů k němu přiléhajících. Díky tomuto způsobu bude přechod mezi dvěma sousedními segmenty stejně plynulý jako přechody mezi úseky segmentu.

Pokud se ale jedná o vrchol znázorňující křižovatku více než dvou cest, začnou vznikat na těchto vrcholech prázdná místa a překrývající rozšíření segmentů. Pro zjednodušení těchto míst napojení je vhodné vykreslení kruhu o poloměru větším než maximální rozšíření sousedních hran. Oba tyto způsoby vytvoří pěkné zakončení segmentu. Knihovna Matplotlib nabízí tuto funkcionalitu v podobě atributu `CapStyle`. Tento atribut umožňuje změnit tvar zakončení vykreslovaných čar. Při testování této funkcionality jsem zjistila, že se jedná i o příhodné řešení pro vyhlazení zubatého okraje, který byl zásadním problémem vykreslování 4.2.2. Pokud rozdělíme pro vykreslení každou hranu na větší množství čar, definovaných vždy dvěma body této čáry, můžeme vykreslit kruh v každém tomto bodě. Jelikož se navíc nejedná o polygon, ale o čáru, lze využít stylu zakončení. Tento proces nezpůsobuje výrazné zpomalení programu a navíc je tato metoda kompatibilní s barevným mapováním vytvářejícím příjemný barevný přechod. Na obrázku 4.5 můžeme vidět srovnání vykreslování bez vyhlazení hran u dvou typů rozšíření. Vykreslení `round boxed` využívá rozšíření pomocí atributu `CapStyle` na jednotlivých vykreslených křivkách, reprezentující části segmentu. Vykreslení `round equidistant` naopak vykresluje kruhové polygony pouze v místech, kde se jednotlivé segmenty napojují.

4.4 Adaptační vizualizace dle měřítka mapy

Při rozšiřování hran je důležité, aby byl na mapě pro rozšíření prostor. Je tedy vhodné navrhnout vykreslování, které svůj vzhled přizpůsobí na základě měřítka zobrazované mapy. Mým prvotním návrhem bylo vyfiltrování cest na základě typu cesty. Při nejvyšším oddálení by se zobrazovaly pouze dálnice a další hlavní dopravní spoje. Při větším přiblížení by se k těmto spojům přidaly například i cesty v obytných oblastech. Tento způsob zobrazení je používán i na většině mapových portálů zmíněných v kapitole 2.2.

Tato metoda se pro naši implementaci ukázala nepříliš vhodná. Jelikož podkladová mapa byla vykreslena ve formě grafu a obsahovala tak všechny cesty - smazáním barevného vykreslování pro cesty

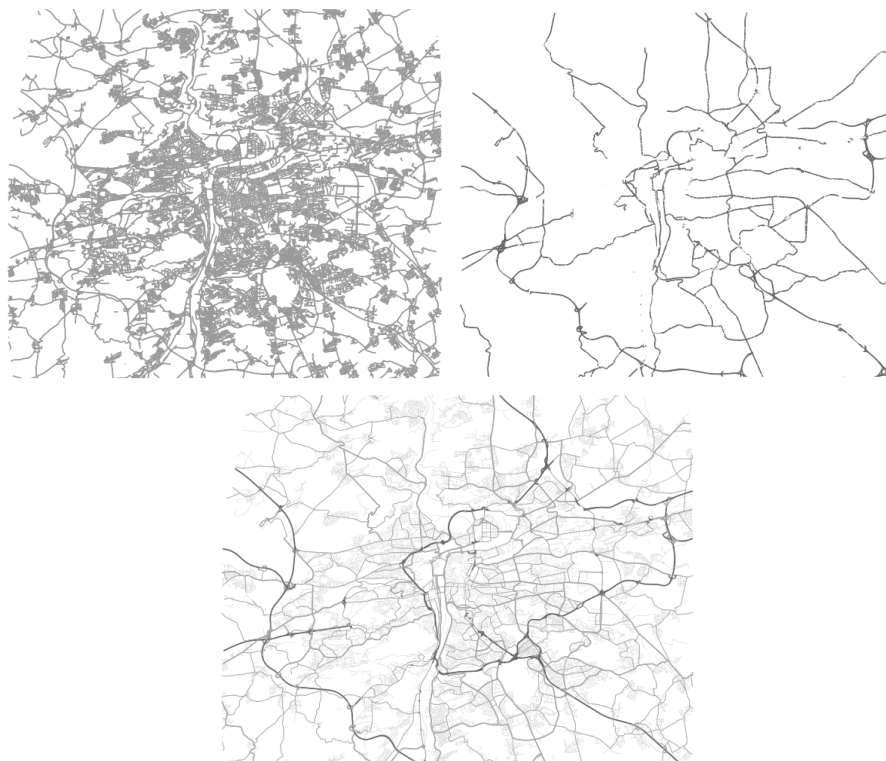


Obrázek 4.5: Srovnání dvou typů rozšíření bez a se zaoblením hran

nižšího řádu bylo pouze docíleno toho, že dopravní síť vypadala méně vytížená a cesty prázdné. Abychom se zbavili tohoto efektu, bylo třeba upravit vykreslování podkladové mapy.

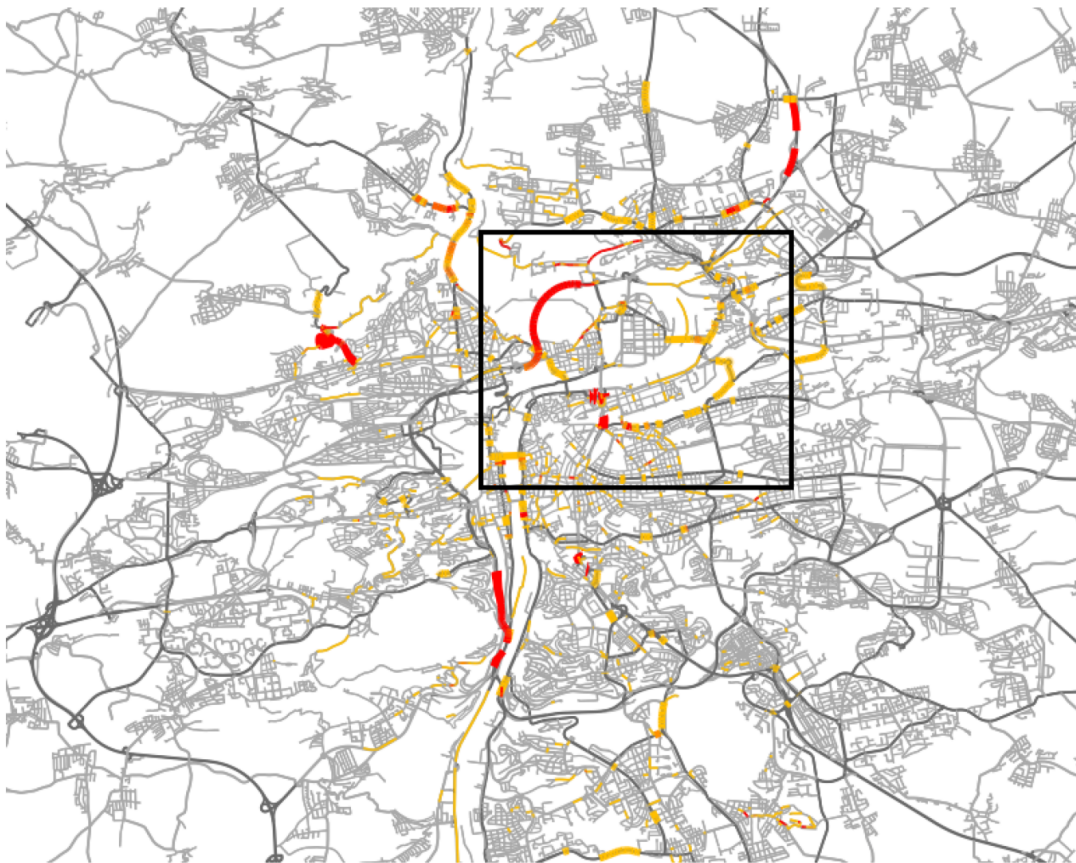
Jednoduchým řešením této problematiky je odstranění hrany grafu dle typu cesty, kterou tato hrana představuje. Tím nám vznikne mnohem méně husté zobrazení dopravní sítě, což umožňuje například zvýšení hodnoty maximálního rozšíření. Srovnání vykreslení podkladové mapy je vidět na obrázku 4.6. Nalevo se nachází původní verze vykreslení grafu dopravní sítě pomocí knihovny OSMnx. Uprostřed můžeme ukázkou toho, jak může vypadat graf po vyfiltrování pouze hlavních dopravních toků. K bližšímu popsání procesu filtrování se věnuji v kapitole 5.4.

Ne vždycky je ale žádoucí cesty nižšího řádu smazat. Smazáním hran z grafu docílíme sice větší přehlednosti, ale přijdeme také o velké množství informací ohledně skutečné členitosti dopravní sítě. Jako mnohem elegantnější řešení se nabízí experimentovat s tloušťkou čar a jejich barvou. Tato metoda je zobrazena na obrázku 4.6 napravo. Využití dvou rozdílných odstínů šedi pro cesty primární a sekundární (ty, které by byly jinak smazány), a tří rozdílných sekundárních šířek čáry dodává podkladové mapě potřebnou strukturu. Mapa tak působí více dynamicky, je přehlednější a navíc přidává výsledné vizualizaci dodatečnou informační hodnotu. Počet využitých odstínů a šířek pak záleží na přiblížení.

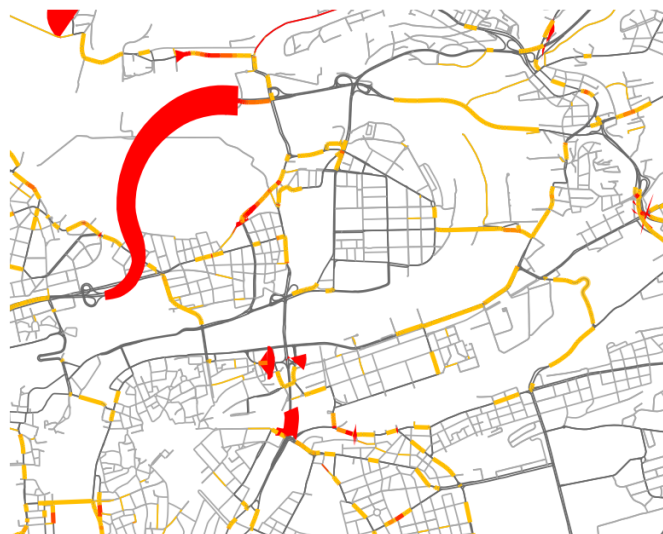


Obrázek 4.6: Srovnání vykreslení podkladové mapy

V posledním kroku je nutné rozhodnout, jakým způsobem se bude vizualizovat intenzita dopravy na sekundárních cestách. Jelikož se tímto dělením snažíme zamezit přehušnění mapy, není vhodné aby se tyto sekundární segmenty vizualizovaly stejně jako ty primární. Kompromisem mezi klasickým zobrazením s rozšířením a naprostou eliminací vykreslování sekundárních segmentů, může být například varianta, kdy se sekundární segmenty vykreslí pouze pomocí gradientu. Tím pak segmenty nezaberou tolik místa a dají prostor důležitým dopravním spojům. Zároveň ale jejich barva určí, zda na těchto úsecích dochází k velké kumulaci aut či nikoli. Jelikož i tento způsob zobrazení by v některých případech nemusel být žádoucí, rozhodla jsem se, že tuto funkcionalitu bude možné ovládat pomocí parametru.



Obrázek 4.7: Vykreslení mapy v měřítku 1 : 320 000



Obrázek 4.8: Vykreslení výřezu mapy v měřítku 1 : 80 000

Kapitola 5

Implementace

Výsledkem mé bakalářské práce je knihovna napsaná v jazyce python založená na metodách knihovny Matplotlib. Po nainstalování knihovny je možné využití funkce `plot_routes`, která přidá do vizualizace vykreslení informační vrstvy tvořené segmenty s určitou intenzitou dopravy. Do knihovny také patří funkce `plot_graph_with_zoom` sloužící pro vykreslení podkladové mapy ve formě síťového grafu na základě jejího měřítka. Tato funkce vychází z funkcionality knihovny OSMnx. Vykreslení jednoho snímku s podkladovou mapou a informační vrstvou ve výsledné vizualizaci trvá přibližně 1.2 sekund na snímek. Funkce `plot_routes` se dá ale použít i s jinými druhy podkladových map, které je možné přes Matplotlib vykreslit. Vytvořenou knihovnu lze také použít s využitím funkce knihovny Matplotlib pro ukádání snímků do videa.

5.1 Struktura vykreslovací funkce

Postup vykreslování vrstvy pro jeden snímek probíhá ve funkci `plot_routes` ve třech základních krocích:

1. Získání geometrie čar a polygonů pro každý segment
2. Vykreslení kolekce barevných čar dle jejich intenzity dopravy
3. Vykreslení kolekce polygonů pro segmenty s rozšířením

Geometrie segmentu se získává z multigrafu, který je parametrem funkce, na základě OSM id krajních bodů segmentu. Tento graf lze sestavit na základě informací z OpenStreetMap a je dostupný ze simulátoru. V tomto kroku se také na základě typu rozšíření generují polygony vyobrazující toto rozšíření či zakončení segmentu. Po ukončení cyklu se geometrie všech segmentů spojí v rámci kolekce, které se nastaví spektrum pro barevné mapování a dále dojde k jejímu vykreslení. Stejným způsobem k tomu dochází v sekci tři pro vykreslování polygonů získaných z první sekce.

Nastavením parametru `plot` na `False` může uživatel docílit přeskočení vykreslovacích sekcí a pouze získat dané kolekce jako návratových hodnot funkce. Uživatel pak může dál tyto kolekce modifikovat, například změnit barevné schéma a podobně. Také má kontrolu nad tím, kdy dojde k vykreslení jednotlivých kolekcí.

5.2 Vykreslení kolekce

Na cestách se dle intenzity dopravy mění jejich barva v rámci spektra metodou popsanou v kapitole 4.1.2. Pro zrychlení vykreslování čar a jednodušší mapování gradientu barev využijeme již dříve zmíněnou strukturu `LineCollection`.

Díky `LineCollection` je možné vykreslit křivky ve formátu $(M, N, 2)$, kdy M udává počet křivek a N počet bodů definujících křivku. Body jsou udávány dvojicí kartézských souřadnic. Tímto způsobem ale lze nastavit pouze M barev, jelikož úseky v rámci jedné křivky musí mít stejnou barvu.

Pro získání více úseků k obarvení je tedy třeba vykreslované křivky transformovat. Listy souřadnic geometrií jednotlivých vyšších segmentů jsou rozděleny na list úseček definovaných svým počátečním a koncovým bodem s dvěma souřadnicemi. Získáme tedy pole velikosti $(P, 2, 2)$, kdy P je násobkem $M*(N-1)$. Měřením jsem vypočetla, že při této operaci dojde k zvýšení počtu vykreslovaných čar na přibližně pětinašobek. Jeden segment tedy obsahuje přibližně pět úseků. Rychlost vykreslování je ovlivněna o přibližně 0.05 s na snímek.

K samotnému mapování bylo vybráno spektrum `autumn_r` z knihovny `Matplotlib`, jelikož nabízí plynulý přechod z žluté po červenou barvu. Tyto barvy jsou často využívány pro zobrazení intenzity dopravy, jak bylo diskutováno v kapitole 2.2. Při zkušebních testech bylo shledáno, že vizualizace rozšířením je vhodná již od hustoty 10 vozidel na segment. Interval barevného mapování tak není dostatečně velký pro větší počet barev. Tento interval barevné škály je možné měnit pomocí parametrů `min_density` a `max_density`.

5.3 Styly rozšíření hran

Pro rozšíření hran jsem se rozhodla ponechat všechny tři navržené styly. Styl `BOXED` odpovídá rozšířením úseků (4.2.2), styl `CALLIGRAPHY` je rozšířením na polygon definovaný posunutím ve směru osy souřadnic (4.2.3) a styl `EQUIDISTANT` vykresluje polygon definovaný ekvidistantní křivkou (4.2.4).

Rozšíření se aplikuje pouze u segmentů s hustotou dopravy vyšší než je udáno v parametru `min_width_density` takovým způsobem, aby segmenty s hustotou stejnou jako je parametr `max_width_density` byly vykresleny nejširším způsobem. Velikost největšího rozšíření je udána parametrem `width_modifier`. Při finalizaci knihovny bylo rozhodnuto, že rozšíření bude udáváno v bodech. Stejně tak tomu je u standardního udávání šířky čáry v knihovně `Matplotlib`.

Směr vykreslení šířky u stylu `CALLIGRAPHY` je udán podle absolutní hodnoty rozdílů prvních a posledních kartézských souřadnic daného segmentu. Jestli je větší rozdíl mezi souřadnicemi y (křivka je spíše svislá) byla použita funkce `fill_betweenx`, která rozšiřuje do stran ve směru osy x.

5.4 Filtrování segmentů dle stupně přiblížení

Jak bylo zmíněno v kapitole 4.4 vizualizace podkladové mapy i samotné informační vrstvy je ovlivněna jejím měřítkem na základě 4 kategorií. Kategorie byly zvoleny na základě kategorií použitých u dopravních map v projektu Floreon+ ¹. Tyto kategorie jsou uvedeny v tabulce 5.1 a jsou k nim přiřazeny typy cest, které je vhodné zobrazit. Typ cesty je převzat z atributu `Highway` ² nastaveným na hraně grafu dopravní sítě získané z OpenStreetMap. Jako maximální měřítko je myšleno nejpodrobnější měřítko, které spadá do daného intervalu úrovně. Například Level 2 je nastaven, pokud měřítko mapy spadá do intervalu 1 : 160 000 až 1 : 319 999.

Tabulka 5.1: Stupně přiblížení mapy

	Maximální měřítko	Typy cest OpenStreetMap
Level 1	1 : 320 000	'motorway', 'motorway_link', 'primary', 'primary_link', 'trunk', 'trunk_link'
Level 2	1 : 160 000	Level 1 + 'secondary', 'secondary_link'
Level 3	1 : 40 000	Level 2 + 'tertiary', 'tertiary_link'
Level 4	1 : 1	Level 3 + 'unclassified', 'residential', 'service', 'living_street', 'road', 'track'

Ve funkci `plot_graph_with_zoom` slouží stupně přiblížení pro výběr zbarvení a šířky cest na podkladové mapě. Uživatel zadá jednu primární hodnotu pro barvu a šířku, spolu s listem barev a šířek sekundárních. Podle získaného stupně přiblížení funkce zvolí, kolik z těchto sekundárních hodnot má vybrat. Popišme si tuto situaci na příkladu.

Pokud systém vyhodnotí, že přiblížení je Level 2, zbarví se primární barvou všechny cesty této kategorie - a to i všechny cesty patřící pod Level 1. Pokud cesta nepatří pod Level 2, zbarví se sekundární barvou. Jedná-li se o cestu spadající pod Level 3, bude se jednat o první z listu sekundárních barev. Cesty spadající pod Level 4 se zbarví druhou hodnotou z tohoto listu. Pokud list nemá dostatečné množství barev, použije se poslední prvek listu nebo list z výchozích parametrů funkce. Stejným procesem se nastavuje i šířka cesty. Pro eliminaci sekundárních cest je možné nastavit sekundární barvu na jednoprvkový list s bílou barvou, či šířku na jednoprvkový list se šířkou nula. Šířka je udávána dle standardů knihovny Matplotlib v bodech.

Pro úpravu zobrazení dle přiblížení ve funkci vykreslující vrstvy intenzit `plot_routes` je třeba nastavit parametr `roadtypes_by_zoom`. Tímto parametrem je možné zastavit vykreslení rozšíření

¹<https://floreon.eu/>

²<https://wiki.openstreetmap.org/wiki/Routing#Highway-type>

pro všechny typy cest, které nepatří pod daný stupeň přiblížení a to s jakýmkoli stylem vykreslování šířky. Barevné mapování je pro tyto segmenty stále k dispozici a jejich vykreslování probíhá společně s primárními segmenty v rámci jedné kolekce. Zapnutí této funkcionality funguje převážně pro zpřehlednění dopravních sítí s výrazně velkým množstvím cest s vysokou intenzitou dopravy.

5.5 Doporučená konfigurace

Funkce pro vykreslování umožňuje použití mnoha konfiguračních parametrů. Souhrn těchto parametrů je popsán v tabulce 5.2. V rámci vývoje komponenty bylo provedeno velké množství testovacích vizualizací z dat ze simulátoru, především vygenerovaných na dopravní síti hlavního města Prahy. Z těchto vizualizací jsem si odnesla poznatky popsané níže.

Název parametru	Popis funkcionality
min/max_density	interval definující mapování barevné škály
min/max_width_density	interval definující mapování rozšíření segmentu
default_linewidth	šířka (v pixelech) nastavená pro segmenty s hustotou provozu rovnou min_density
width_style	styl rozšíření čáry (boxed calligraphy equidistant)
width_modifier	šířka (v pixelech) nastavená pro segmenty s hustotou provozu rovnou max_width_density
round_edges	pravdivostní hodnota udávající, zda jsou přechody mezi segmenty zakulacené pro plynulejší návaznost
roadtypes_by_zoom	pravdivostní hodnota udávající, zda jsou segmenty filtrovány na základě typu cesty a úrovně přiblížení
hidden_lines_width	šířka vykreslení cest, které jsou odfiltrovány na základě úrovně přiblížení (při šířce 0 nedojde k vykreslení hustoty provozu na těchto segmentech)
plot	pravdivostní hodnota udávající, zda segmenty vykreslit nebo je pouze exportovat jako LineCollection/PolygonCollection

Tabulka 5.2: Souhrn parametrů funkce plot_routes

Jelikož použitá dopravní síť je velice členitá, tak i při velkém množství aut má většina segmentů velmi malou intenzitu. Proto je vhodné nastavit barevný přechod pro interval aut mezi 1 až 10 vozidly a od 10 vozidel na úseku segmentu jej začít rozšiřovat.

Nejjednodušší je využití rozšiřování typu BOXED v kombinaci s parametrem round_edges zajišťující plynulé přechody. Jedná se o rychlé vizualizační řešení a umožňuje i gradientový přechod. Pro kvalitnější vizualizaci vystihující přesněji tvar úseku je nejvýhodnější využití stylu EQUIDISTANT. Tento styl je také příhodný, pokud jsou intenzity v rámci jednoho segmentu velice rozdílné. Srovnání těchto stylů je vidět na obrázku 5.1.

Hodnota parametru width_modifier hodně závisí jak na velikosti okna, tak na velikosti a členitosti vykreslované dopravní sítě. Při zobrazení oblasti o velikost asi 20x20 km a standardní velikosti



Obrázek 5.1: Srovnání stylů BOXED se zaoblením a EQUIDISTANT bez zaoblení

Mathplotlib okna 6,4 x 4,8 palců bych doporučovala maximální rozšíření na hodnotě 4. Bez použití `round_edges` je možné tuto hodnotu zvýšit až na 9. Pokud zvýšíme velikost okna například na 14,9 x 7,4 palců pak se u obou typů rozšířit na 10-12 bodů. Pokud bychom pracovali s menším výřezem mapy, například od rozměrech 3x menších tedy přibližně 7 x 7 km, můžeme velikost atributu `width_modifier`, také 3x zvýšit. Pro snadnější manipulaci při přibližování je možné využít funkce knihovny `map_distance_to_points`. Tato funkce převede vzdálenost na mapě na velikost, kterou lze využít pro nastavení šířky segmentu. Jelikož přepočítání bere v úvahu velikost okna a vyobrazované oblasti, stačí nastavit rozšíření na přibližně 1/1000-3/1000 jednotek vzdálenosti na mapě. Návratovou hodnotu funkce lze pak jednoduše nastavit jako vstupní parametr pro vykreslování.

Kapitola 6

Závěr

Tato bakalářská práce se zabývá vizualizací časového vývoje dopravního toku, která je využívána v rámci projektu EVEREST a vytvářena na základě dat z dopravního simulátoru. Cílem práce bylo navrhnout a implementovat komponentu pro generování snímků vizualizace. Spojením těchto snímků vznikne plynulé video zobrazující dopravní tok. Dopravní síť je na snímku reprezentována grafem, jehož vrcholy a hrany odpovídají křižovatkám a dopravním komunikacím.

V rámci práce byla provedena rešerše existujících řešení a nástrojů a seznámení s vizualizačními metodami na mapových portálech. Mezi nejčastější řešení patří vizualizace změn intenzity dopravy pomocí barevných kategorií určujících stav segmentu či pomocí teplotní mapy. Rozšiřování segmentů je využíváno především u vizualizací statických. Na základě srovnání vizualizačních knihoven byla zvolena knihovna Matplotlib, díky možnosti rozsáhlých úprav vykreslování, ukládání snímků a animací a integrace s dalšími nástroji pro zpracování dat a jejich reprezentaci v podobě síťového grafu.

Výsledkem práce je open-source knihovna¹ umožňující vizualizaci intenzit dopravy pomocí barevného spektra a tří typů rozšíření hran. Nejrychlejší je způsob vizualizace nazvaný **BOXED**, při kterém dojde k rozdělení segmentu na více úseků a jejich napojení pomocí zaobleného zakončení čáry. Nejpresnějšího rozšíření dosahuje styl **EQUIDISTANT** vykreslující polygon mezi dvěma křivkami, které jsou definovány svou vzdáleností od křivky segmentu. Plynulosti navazování segmentů je docíleno vykreslením dodatečných kruhů na uzlech grafu.

Knihovna také umožňuje modifikaci podkladové mapy a samotné informační vrstvy na základě měřítko zobrazované mapy a OSM typu dopravní komunikace. Tato funkcionality vizualizaci nejen zpřehledňuje a dodává jí dynamiku, ale nabízí i dodatečnou informační hodnotu.

Využitím funkce knihovny Matplotlib pro ukládání animací umožňuje vytvořená knihovna vygenerovat video dopravního toku z předzpracovaných dat v rychlosti přibližně 1.2 sekund na snímek. Ukázka videa je společně s příkladem pro spuštění dostupná v rámci git repozitáře knihovny.

¹<https://github.com/It4innovations/FlowMapFrame>

Pro rychlejší generování videa je vhodné využít optimalizovanou variantu vyvíjenou ve spolupráci se Sofíí Michailidu.

Možným rozšířením komponenty by mohla být do budoucna automatická konfigurace parametrů pro vizualizaci na základě hustoty dopravní sítě. Tím by se minimalizovala manuální práce s nastavením vizualizace a zároveň by se dosáhlo co nejvěrnějšího zobrazení vývoje dopravního toku. Dále také optimalizované vykreslení pohybu jednotlivých aut či jejich clusterů. Tato funkcionality je již nad rámec zadání této bakalářské práce a bude dokončena v rámci spolupráce s týmem výzkumné laboratoře ADAS na IT4Innovations. V neposlední řadě je v plánu použít návrh této komponenty jako základ vizualizace v platformě Floreon+.

Literatura

1. CONSORTIUM, EVEREST. *Design environment for extreme-scale big data analytics on heterogeneous platforms* [online]. 2020. [cit. 2023-01-04]. Dostupné z: <https://everest-h2020.eu/>.
2. ŠURKOVSKÝ, Martin; BERÁNEK, Jakub; SLANINOVÁ, Kateřina; MARTINOVIČ, Jan; CMAR, Radim. *Traffic Flow Optimization in Smart Cities: Leveraging HPC Resources for Automatic Design Space Exploration* [online]. 2022. [cit. 2023-02-12]. Dostupné z: https://sc22.supercomputing.org/proceedings/tech_poster/tech_poster_pages/rpost168.html.
3. CHEN, Wei; GUO, Fangzhou; WANG, Fei-Yue. A Survey of Traffic Data Visualization. *IEEE Transactions on Intelligent Transportation Systems*. 2015, roč. 16, č. 6, s. 2970–2984. Dostupné z DOI: 10.1109/TITS.2015.2436897.
4. ANDRIENKO, Gennady; ANDRIENKO, Natalia. Spatio-temporal aggregation for visual analysis of movements. In: *2008 IEEE Symposium on Visual Analytics Science and Technology*. 2008, s. 51–58. Dostupné z DOI: 10.1109/VAST.2008.4677356.
5. CLARINVAL, Antoine; DUMAS, Bruno. Intra-City Traffic Data Visualization: A Systematic Literature Review. *IEEE Transactions on Intelligent Transportation Systems*. 2022, roč. 23, č. 7, s. 6298–6315. Dostupné z DOI: 10.1109/TITS.2021.3092036.
6. *Názvosloví v geodézii a kartografii*. 1989. Tech. zpr., Třídící znak 730401. Český normalizační institut.
7. KAŇOK, Jaromír. Kartogram a kartodiagram - stanovení objektivní stupnice. *Sborník ČGS*. 1999, roč. 104, č. 4, s. 268–281. Dostupné také z: https://geografie.cz/media/pdf/geo_1999104040268.pdf.
8. PACK, Michael L.; WONGSUPHASAWAT, Krist; VANDANIKER, Michael; FILIPPOVA, Darya. ICE–visual analytics for transportation incident datasets. In: *2009 IEEE International Conference on Information Reuse & Integration*. 2009, s. 200–205. Dostupné z DOI: 10.1109/IRI.2009.5211551.

9. CHEN, Haidong; CHEN, Wei; MEI, Honghui; LIU, Zhiqi; ZHOU, Kun; CHEN, Weifeng; GU, Wentao; MA, Kwan-Liu. Visual Abstraction and Exploration of Multi-class Scatterplots. *IEEE Transactions on Visualization and Computer Graphics*. 2014, roč. 20, č. 12, s. 1683–1692. Dostupné z DOI: [10.1109/TVCG.2014.2346594](https://doi.org/10.1109/TVCG.2014.2346594).
10. LIU, Siyuan; PU, Jiansu; LUO, Qiong; QU, Huamin; NI, Lionel M.; KRISHNAN, Ramayya. VAIT: A Visual Analytics System for Metropolitan Transportation. *IEEE Transactions on Intelligent Transportation Systems*. 2013, roč. 14, č. 4, s. 1586–1596. Dostupné z DOI: [10.1109/TITS.2013.2263225](https://doi.org/10.1109/TITS.2013.2263225).
11. XIE, Zhixiao; YAN, Jun. Kernel Density Estimation of traffic accidents in a network space. *Computers, Environment and Urban Systems*. 2008, roč. 32, č. 5, s. 396–406. ISSN 0198-9715. Dostupné z DOI: <https://doi.org/10.1016/j.compenvurbsys.2008.05.001>.
12. PHAN, Doantam; XIAO, Ling; YEH, R.; HANRAHAN, P. Flow map layout. In: *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005*. 2005, s. 219–224. Dostupné z DOI: [10.1109/INFVIS.2005.1532150](https://doi.org/10.1109/INFVIS.2005.1532150).
13. ČSN 73 6100-2, *Názvosloví pozemních komunikací - Část 2: Projektování pozemních komunikací*. 2008. Tech. zpr., Třídící znak 736100. Český normalizační institut.
14. WANG, Zuchao; LU, Min; YUAN, Xiaoru; ZHANG, Junping; VAN DE WETERING, Huub. Visual Traffic Jam Analysis Based on Trajectory Data. *IEEE Transactions on Visualization and Computer Graphics*. 2013, roč. 19, č. 12, s. 2159–2168. Dostupné z DOI: [10.1109/TVCG.2013.228](https://doi.org/10.1109/TVCG.2013.228).
15. *Nový bypass okružní křižovatky z Vídeňské na Chodov* [online]. Městská část Praha-Kunratice, 2021. [cit. 2023-02-25]. Dostupné z: <https://www.praha-kunratice.cz/zpravy/5182-novy-bypass-okruzni-krizovatky-z-videnske-na-chodov>.
16. *Prezentace výsledků sčítání dopravy 2020* [online]. Ředitelství silnic a dálnic ČR, 2022. [cit. 2023-01-04]. Dostupné z: https://scitani.rsd.cz/CSD_2020/pages/map/default.aspx.
17. HÖFFLER, Tim N.; LEUTNER, Detlev. Instructional animation versus static pictures: A meta-analysis. *Learning and Instruction*. 2007, roč. 17, č. 6, s. 722–738. ISSN 0959-4752. Dostupné z DOI: <https://doi.org/10.1016/j.learninstruc.2007.09.013>.
18. LI, Yudan; CHATTERJEE, Kasturi; CHEN, Shu-Ching; ZHANG, Keqi. A 3-D Traffic Animation System with Storm Surge Response. In: *2009 11th IEEE International Symposium on Multimedia*. 2009, s. 257–262. Dostupné z DOI: [10.1109/ISM.2009.95](https://doi.org/10.1109/ISM.2009.95).
19. SEWALL, Jason; BERG, Jur van den; LIN, Ming; MANOCHA, Dinesh. Virtualized Traffic: Reconstructing Traffic Flows from Discrete Spatiotemporal Data. *IEEE Transactions on Visualization and Computer Graphics*. 2011, roč. 17, č. 1, s. 26–37. Dostupné z DOI: [10.1109/TVCG.2010.27](https://doi.org/10.1109/TVCG.2010.27).

20. KAPLER, T.; WRIGHT, W. GeoTime information visualization. In: *IEEE Symposium on Information Visualization*. 2004, s. 25–32. Dostupné z DOI: 10.1109/INFVIS.2004.27.
21. BASKARAN, Savitha; FANG, Shiaofer; JIANG, Shenhui. Spatiotemporal visualization of traffic paths using color space time curve. In: *2017 IEEE International Conference on Big Data (Big Data)*. 2017, s. 3398–3405. Dostupné z DOI: 10.1109/BigData.2017.8258325.
22. *Top websites ranking in Czech Republic in March 2023 - Similarweb* [online]. similarweb, 2023. [cit. 2023-04-28]. Dostupné z: <https://www.similarweb.com/top-websites/czech-republic/>.
23. *Use layers to find places, traffic, terrain, biking & transit - Android - Google Maps Help* [online]. support.google.com, 2023. [cit. 2023-01-04]. Dostupné z: <https://support.google.com/maps/answer/3092439>.
24. *Silniční databanka a NDIC* [online]. Ředitelství silnic a dálnic ČR, 2023. [cit. 2023-01-04]. Dostupné z: <https://www.rsd.cz/rsd/silnicni-databanka-a-ndic#zalozka-narodni-dopravni-informacni-centrum-ndic>.
25. *Česká republika – Traffic information* [online]. Ředitelství silnic a dálnic ČR, 2023. [cit. 2023-03-04]. Dostupné z: <https://dopravniinfo.cz/>.
26. DALE, Kyran. *Data Visualization with Python and JavaScript*. "O'Reilly Media, Inc.", 2022-12.
27. GRANGER, Brian E.; PÉREZ, Fernando. Jupyter: Thinking and Storytelling With Code and Data. *Computing in Science & Engineering*. 2021, roč. 23, č. 2, s. 7–14. Dostupné z DOI: 10.1109/MCSE.2021.3059263.
28. HUNTER, J. D. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*. 2007, roč. 9, č. 3, s. 90–95. Dostupné z DOI: 10.1109/MCSE.2007.55.
29. GEOPANDAS DEVELOPERS. *geopandas/geopandas: v0.12.2* [online]. Zenodo, 2022-12. Ver. v0.12.2 [cit. 2023-01-26]. Dostupné z DOI: 10.5281/zenodo.7422493.
30. BOEING, Geoff. OSMNX: New Methods for Acquiring, Constructing, Analyzing, and Visualizing Complex Street Networks. *Computers Environment and Urban Systems*. 2017-07, roč. 65, s. 126–139. Dostupné z DOI: 10.1016/j.compenvurbsys.2017.05.004.
31. ERIC, McGregor; PATRICK, Sly. *Using D3.js* [online]. 2019. [cit. 2023-01-26]. Dostupné z: <http://using-d3js.com>.
32. D3 CONTRIBUTORS. *D3 Gallery* [online]. Observable, Inc., 2022-12. [cit. 2023-02-03]. Dostupné z: <https://observablehq.com/@d3/gallery>.
33. INC., Plotly Technologies. *Collaborative data science* [online]. Montreal, QC: Plotly Technologies Inc., 2015 [cit. 2023-02-03]. Dostupné z: <https://plotly.com/javascript/>.

34. OTTO, J. *Ottův slovník naučný: ilustrovaná encyklopaedie obecných vědomostí*. Praha, 1888. Č. 1. <https://www.digitalniknihovna.cz/nkp/uuid/uuid:204a0f70-05a7-11e5-95ff-5ef3fc9bb22f>.
35. GILLIES, Sean et al. *Shapely: manipulation and analysis of geometric objects* [online]. toblerity.org, 2007. [cit. 2023-03-01]. Dostupné z: <https://github.com/Toblerity/Shapely>.
36. FALALEEV, Nikolay. *Approximate equidistant curve for polynomials* [online]. Computer Vision Lab, 2017-05. [cit. 2023-01-02]. Dostupné z: <https://nikolasent.github.io/mathematics/algorithms/2017/05/02/Approximate-equidistant-curve-for-polynomials.html>.