

# Teoretická simulácia dopravy lietajúcich áut v Prahe v programovacom jazyku Python

Peter Basár

České Vysoké Učení Technické Fakulta Elektrotechnická



## Úvod

Táto práca je podmienená otázkou možnej budúcnosti dopravy, kde vo vzdušnom priestore mesta premávajú lietajúce autá. Táto práca avšak nereprezentuje žiadnu predpoveď budúcnosti ale slúži len ako hrubý odhad dopravy lietajúcich áut ako aj použiteľný zdroj a informácie ako takú jednoduchú simuláciu vytvoriť.

Zamerali sme sa na tieto otázky:

Koľko lietajúcich áut by preletelo vzdušným priestorom Prahy počas dňa ak by sme počet lietajúcich áut určovali z počtu registrovaných osobných automobilov v Prahe, a predpokladali by sme kyvadlovú dopravu z okrajov Prahy do najhustejšie zaľudnených častí a naspäť.

Zvolili sme si 1 bod v Prahe (FEL ČVUT DEJVICE), kde skúmame celkový a priemerný počet preletov áut za hodinu, minútu a počas celého dňa v určenej radiálnej vzdialenosti 150 metrov.

V prvej časti tejto práce si rozoberieme problematiku pojmu lietajúcich áut.

V druhej časti si opíšeme dáta, ktoré sme použili v simulácii ako aj extrakciu a prenesenie získaných dát do formy, s ktorou môžeme pracovať.

V tretej časti si rozoberieme simuláciu naprogramovanú v Python-e<sup>[1]</sup> za pomoci knižníc OpenCV<sup>[2]</sup> a NumPy<sup>[3]</sup>.

V poslednej štvrti časti si roznalýzujeme výsledky, ktoré sme zo simulácie získali.

## 1. Problematika lietajúcich áut

Pojem lietajúce autá v tejto práci nie je vyhradený pre autá, ktoré dokážu lietať v zmysle, že dokážu jazdiť na ceste a lietať vo vzduchu. Pojem lietajúce autá v tejto práci zastrešujú vzdušné vozidlo, ktoré dokáže svojho majiteľa premiestniť z bodu A do bodu B. Presnejší pojem by bola osobná helikoptéra bez (nie nutne) ľudského pilota.

Táto práca je čiastočne reakciou na snahy niektorých globálnych spoločností vytvoriť sieť osobných helikoptér vo veľkých mestách formou taxislužby.

Takéto zavedenie nesie so sebou veľa výziev, ktoré musia tieto spoločnosti prekonať. Jedným z nich je aj hlučnosť daných vozidiel a táto práca na túto problematiku jemne nadväzuje.

Keďže je známe, že domáce drony sú hlučné zariadenia. Hlučnosť vozidla, ktorého výkon a hmotnosť bude výrazne vyššia sa bez pochyby, za súčasných technológií, prejaví aj na množstve vyprodukovaného hluku. Samozrejme, bojovať s hlukom sa dá viacerými spôsobmi ale táto práca je viac motivovaná tým, že prelet lietajúceho vozidla by bol hlučný zo vzdialenosti 150 metrov.

Táto práca poskytuje teda určité informácie na predstavu toho ako by vypadalo vzdušné prostredie, v ktorom premáva určitý počet lietajúcich vozidiel.

## 2. Dáta

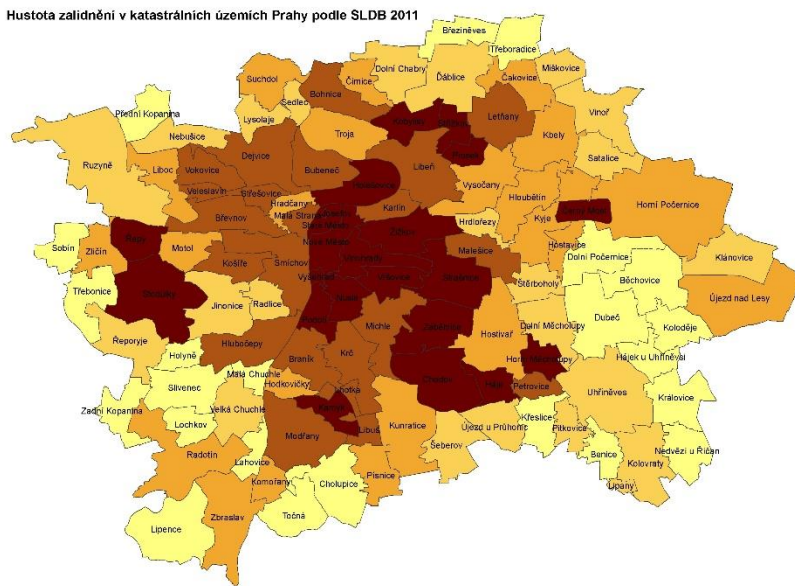
### a. Použité dáta

Na to aby sme sa so simuláciou dokázali aspoň trochu priblížiť realite potrebujeme čo najrelevantnejšie dáta a informácie ohľadom dopravy v Prahe.

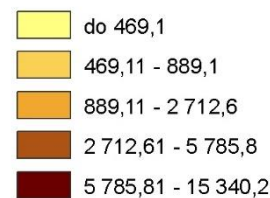
Najoptimálnejšie dáta ktoré by sme mohli mať by boli také, ktoré by hovorili o tom ako funguje kyvadlová doprava v Prahe a to tak, že by sme mali informácie o začiatku a konci trasy pre každé auto, ktoré jazdí Prahou.

Jedným zo spôsobov ako by sme mohli takéto dáta získať by bola kooperácia s mobilnými operátormi, ktoré takéto dáta ukladajú alebo s „big-data“ spoločnosťami, ktoré takéto dáta využívajú pre svoje produkty. Z dôvodu nedostatku času a malej relevantnosti tejto práce sme sa rozhodli pre menej relevantnejšie dáta a to konkrétne hustota zaľudnenia v Prahe podľa katastrálnych území<sup>[4]</sup>.

Hustota zaľudnenia v katastrálnych územích Prahy podľa SLDB 2011



hustota zaľudnenia  
(obvykle bydlíci obyvatelia na 1 km<sup>2</sup>)



Zdroj: ČSÚ, SLDB 2011

**Obrázok č.1** Zdroj: Český statistický úřad, Sčítání lidu, domů a bytů - Hlavní město Praha - analýza výsledků – 2011, Hustota zaľudnenia v katastrálnych územích Prahy podľa SLDB 2011<sup>[4]</sup>

Porovnaní dle rozlohy, počtu obyvatel a stupňů motorizace a automobilizace		
	Praha	ČR
Rozloha (km <sup>2</sup> )	496	78 870
Počet obyvatel (mil.)	1,295	10,610
Počet motorových vozidel (tis.)	1 059	7 551
z toho osobní automobily (tis.)	845	5 573
Stupeň motorizace		
motorových vozidel na 1 000 obyvatel	818	712
počet obyvatel na 1 motorové vozidlo	1,2	1,4
Stupeň automobilizace		
osobních automobilů na 1 000 obyvatel	652	525
počet obyvatel na 1 osobní automobil	1,5	1,9

**Obrázok č.2** Zdroj: Technická správa komunikací hl. m. Prahy, a. s., Ročenka dopravy 2017 str:7<sup>[5]</sup>

Ako už je v úvode uvedené, počet lietajúcich áut budeme určovať z celkového počtu registrovaných osobných automobilov v Prahe<sup>[5]</sup>. Zo stránky Technickej správy komunikácií hl. m. Prahy sme zistili, že v Prahe sa nachádzalo (2017) 652 automobilov na 1 000 obyvateľov.

Na to aby sme spočítali celkový odhad počtu osobných automobilov v Prahe potrebujeme celkový počet obyvateľov Prahy. Tieto údaje sme získali zo stránky Českého statistického úradu<sup>[6]</sup>. Počet obyvateľov v Prahe v roku 2017 bol 1 294 513, na výpočty sme použili číslo 1 300 000.

Celkový počet osobných automobilov v Prahe (2017) nám teda vyšiel 847 600.

Dáta, ktoré nám pomohli v popise aktivity osobných automobilov počas dňa boli taktiež

z ročenky Technickej správy komunikácií hl. m. Prahy<sup>[7]</sup>. Konkrétne sa jedná o denné variácie dopravy v Prahe počas pracovného dňa. Tieto dáta sme ale využili iba na relatívne porovnanie aktivity dopravy rána oproti obedu.

Predpokladáme, že sa v Prahe priemerne počas dňa všetky osobné automobily nevyužívajú, a preto sme si určili, že z celkového počtu registrovaných automobilov v Prahe sa v našej simulácii využije 80% (imaginárna premenná).

Pracovali sme taktiež s premennou, ktorá percentuálne určuje celkový počet lietajúcich áut v Prahe z počtu osobných automobilov. Simulovali sme dopravu pre 1%,5%,10%,50%,100%. Výsledný počet lietajúcich áut bol vypočítaný nasledovne:

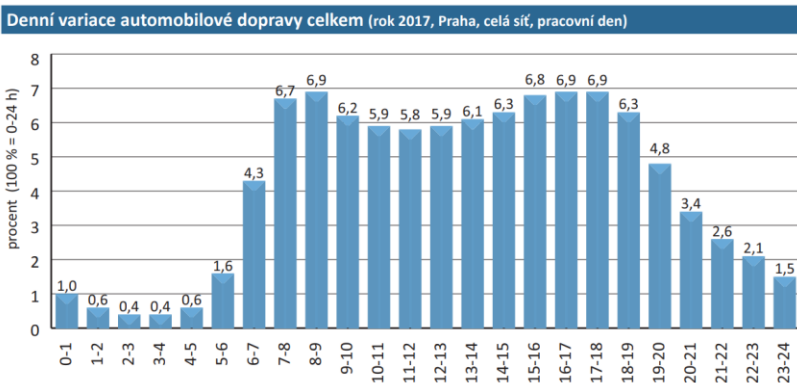
Cl: Celkový počet lietajúcich automobilov  
Pos: Počet osobných automobilov (847 600)  
Pcv: Percento využitých (80%)  
Pcl: Percento lietajúcich {1%,5%,10%,50%,100%}

$$Cl = Pos * Pcv * Pcl$$

napr.

$$6\,780 = 847\,600 * 0.8 * 0.01$$

Celkovú kyvadlovú dopravu v Prahe, ktorú chceme simulovať sme si rozdelili na 3 časti: ráno (6:00-11:00), obed (11:00-15:00), večer (15:00-20:00). Ráno a večer pracuje s rovnakým počtom a to celkovým počtom lietajúcich automobilov a obed pracuje s časťou celkového počtom lietajúcich áut, pričom berieme do úvahy, že na obed sa v Prahe využije 87% (fiktívna premenná) z počtu lietajúcich áut, ktoré sa využijú počas rána a večera. A teda



**Obrázok č.3** Zdroj: Technická správa komunikací hl. m. Prahy, a. s., Ročenka dopravy 2017 str:14<sup>[7]</sup>

Clo: Celkový počet lietajúcich automobilov obed  
Pos: Počet osobných automobilov (847 600)  
Pcv: Percento využitých (80%)  
Pcl: Percento lietajúcich {1%,5%,10%,50%,100%}  
Pco:Percento lietajúcich áut počas obeda (87%)

$$Clo = Pos * Pcv * Pcl * Pco$$

napr.

$$5\,899 = 847\,600 * 0.8 * 0.01 * 0.87$$

Na to aby sme mali informácie ohľadom priemerného počtu áut v našom okolí, musíme si určiť referenčný bod (FEL ČVUT DEJVICE), v okolí ktorého budeme lietajúce autá počítať a vzdialenosť od referenčného bodu, ktorá bude určovať oblasť, v ktorej sa bude počítať, keď do nej vozidlo vstúpi. Za referenčný bod sme si určili elektrotechnickú fakultu ČVUT. Za vzdialenosť sme si určili 150 metrov.

## **b. Extrahovanie dát z obrázka pre hustotu zaľudnenia**

Na to aby sme dokázali reprezentovať údaje z farebne vyznačeného obrázka pre hustotu zaľudnenia potrebujeme istým spôsobom z obrázka získať číselné údaje. Vieme, že pixely sú na obrázkoch najčastejšie reprezentované pomocou trojice čísel v rozpätí 0-255, ktoré popisujú RGB spektrum. Na to aby sme dokázali tieto údaje interpretovať si musíme vytvoriť program, ktorý, v našom prípade, prehliada každý pixel obrázka a snaží sa prísť na to do aká hustota danej farbe zodpovedá (legenda na obrázku č.1).

Ak sa na daný obrázok ale pozrieme môžeme si všimnúť, že sa v obrázku nachádza veľa rušivých elementov ako názvy území a hranice medzi územiami.

Aby sme sa zbavili týchto nedostatkov tak sme názvy a hranice z obrázku odstránili [obrázok č.4 vrch] a na zjednodušenie detekovania farieb sme si zvolili čierno-biele spektrum [obrázok č.4 stred]. Teda piatim farbám z legendy sme priradili 5 farieb na čiernobielym spektre. Aby sme rozlíšili pozadie priradili sme mu odlišnú farbu [obrázok č.4 spodok].

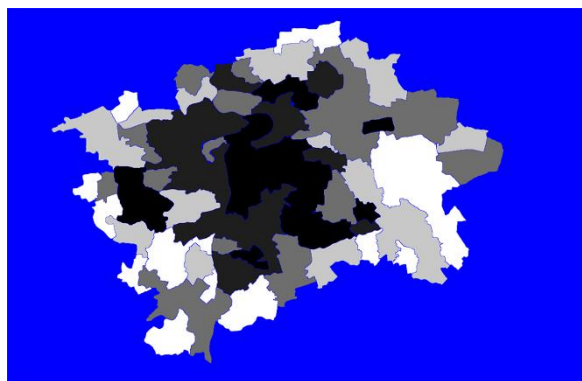
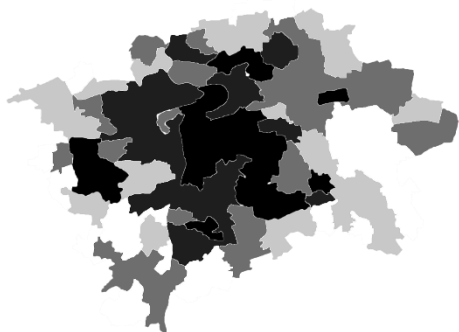
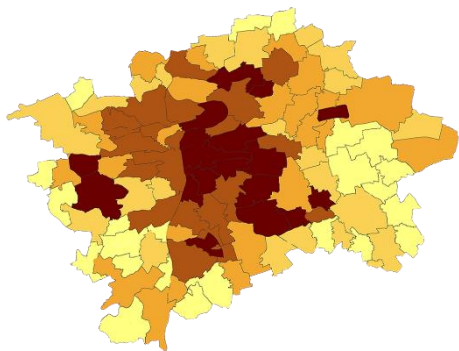
Aby sme si dokázali následnú simuláciu krajšie vizualizovať. Tak sme si prichystali k upravenému obrázku aj štylizovaný obrázok, ktorý bol získaný a zostrihaný z častí máp za pomoci softvéru Styling Wizard<sup>[8]</sup>.



**Obrázok č.3** Mapa zostrihaná z kúskov máp pomocou softvéru Styling Wizard<sup>[8]</sup>

Dimenzie obrázku mapy a obrázku hustoty zaľudnenia boli jemne odlišné, a tak sme obrázok s hustotou zaľudnenia jemne pootočili a zmenšili/zväčšili aby sa zhodoval s obrázkom mapy.

Treba si dať ale pozor na to, aby boli obrázky, ktoré budeme nahrávať do programu



**Obrázok č.4** Zdroj: Český statistický úřad, Sčítání lidu, domů a bytů - Hlavní město Praha - analýza výsledků – 2011, Hustota zalidnění v katastrálních územích Prahy podle SLDB 2011<sup>[4]</sup>

\*upravený obrázok č.1

v rovnakom rozlíšení, keďže budeme interpretovať informácie a potom ich vykresľovať po pixeloch.

Následne si daný upravený obrázok načítame v programovacom jazyku Python za pomoci knižnice OpenCV a začneme ho interpretovať.

Na to nám ale treba algoritmus:

Upravený obrázok sme sa rozhodli interpretovať tak, že si ho rozdelíme na sériu maličkých štvorčekov, konkrétne s rozmermi 5x5 pixelov.

Následne kontrolujeme každý pixel v danom štvorčeku a hľadáme najčastejšiu farbu, aká sa tam vyskytuje.

Farbu, ktorá sa tam najčastejšie vyskytuje porovnáme s našou čierno-bielou legendou hustoty obyvateľstva.

Informácie o ľavom hornom pixeli/ pravom dolnom pixeli (tieto pixely ukladáme preto, aby sme z daných informácií dokázali identifikovať hranice štvorčekov) a väčšinovej hustoty štvorčeka si uložíme do poľa.

Keď prejdeme všetkými malými štvorčkami v obrázku, celé pole, v ktorom sú uložené informácie o štvorčkoch, uložíme do súboru. (V našom prípade sme dáta ukladali do súboru boxes.csv)

Pre ilustráciu, štvorčky s hustotou vypadali nasledovne [obrázok č. 5].

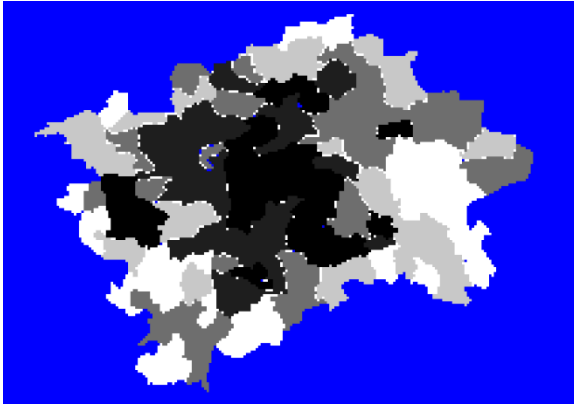
Celý algoritmus na načítanie obrázku a ukladanie dát je možný nájsť na Githube<sup>[9]</sup> v súbore *readSaveDataPicture.py*

### 3. Simulácia

Simulácia, ktorú sme naprogramovali funguje na nasledovnom princípe:

Simuluje každú sekundu tým spôsobom, že sekunda je v simulácii braná ako premenná, pričom simulácia začína o 6:00 (21 600. sekunda dňa) ráno a končí o 20:00 (72 000. sekunda dňa) večer.





**Obrázok č.5** Zdroj: Český statistický úřad, Sčítání lidu, domů a bytů - Hlavní město Praha - analýza výsledků – 2011, Hustota zalidnění v katastrálních územích Prahy podle SLDB 2011<sup>[4]</sup>

\*upravený obrázok č.3 dole

Algoritmus na simuláciu času je možný nájsť na Githube<sup>[9]</sup> v súbore *simulationCore.py*

Naša simulácia pracuje s akousi hustotou lietajúcich áut  $x$  za minútu. Táto hodnota sa počíta ako podiel celkového počtu lietajúcich áut, ktoré majú za dané časové obdobie preletieť (ráno, obed, večer) a dĺžka daného časového obdobia v sekundách.

A teda, každú minútu pridáva  $x$  automobilov do priestoru. Algoritmus na výpočet hustoty automobilov a dĺžky časových období je možné nájsť na Githube<sup>[9]</sup> v súbore *simulateTraffic.py*

Pridávanie automobilov funguje na tomto princípe. Z nahratých dát o štvorčekoch vieme, aký štvorček reprezentuje akú hustotu. Ráno predpokladáme smer dopravy z okrajov Prahy do centra. Ráno je pravdepodobnosť výberu začiatku trasy z krajných častí Prahy (časti s menšou hustotou) väčšia ako z vnútorných častí Prahy (časti s väčšou hustotou), a taktiež pravdepodobnosť, že si vyberieme koniec trasy je väčšia v strede Prahy (väčšia pravdepodobnosť že autá pôjdu z krajov do stredu). Na obed je pravdepodobnosť výberu začiatku a konca trasy rovnaká na celej Prahe. Večer sa využijú pravdepodobnosti z rána len sa prehodia (väčšia pravdepodobnosť že autá pôjdu z centra von). Toto nie je úplne pravda ale potrebovali sme určitým spôsobom simulovať kyvadlovú dopravu. Podmienka výberu je taktiež minimálna dĺžka trasy. Túto dĺžku sme určili ako 4 800 metrov čo je na našej mape ekvivalent 150 pixelov. Teda, ak vzdialenosť začiatku a konca bude menej ako 150 pixelov, daný výber pokladáme za neplatný a opakujeme ho, kým nenájdeme vhodnú náhradu. Algoritmus na pridávanie automobilov je možné nájsť na Githube<sup>[9]</sup> v súbore *simulateTraffic.py*, algoritmus na pravdepodobnostné vyberanie začiatku a konca trasy na základe časového obdobia v súbore *simulationCore.py*

Pridané vozidlá sa v našej simulácii pohybujú po „vzdušnej čiare“/priamke, ktorej začiatočný a koncový bod je určený začiatočným a koncovým bodom daného lietajúceho auta. Rýchlosť lietajúceho auta sme simulovali v pixeloch/sekundu.

V našom prípade sme použili rýchlosť 0,7 pixela za sekundu, čo po prepočítaní koresponduje s rýchlosťou 22.4 metra za sekundu. Táto rýchlosť je síce malá ale nemá veľký vplyv na simuláciu. Ide iba o kozmetickú úpravu v našej simulácii. Ak by sme túto rýchlosť zdvojnásobili, takmer nič by sa v našej simulácii nezmenilo okrem vizuálnej stránky. Algoritmus na simuláciu pohybu lietajúcich áut je možné nájsť na Githube<sup>[9]</sup> v súbore `simulateTraffic.py`

Počas behu našej simulácie neustále zaznamenávame dáta o lietajúcich vozidlách. Konkrétne si zaznamenávame začiatok trasy, koniec trasy lietajúcich vozidiel a čas kedy boli pridané. Tieto dáta si ukladáme do 3 súborov podľa časového obdobia kedy boli vytvorené. Z jedného behu simulácie teda máme dáta z rána, obeda aj večera. Algoritmus na ukladanie dát je možné nájsť na Githube<sup>[9]</sup> v súbore `simulateTraffic.py`

Video zo simulácie je dostupné na YouTube<sup>[10]</sup>.

#### 4. Spracovanie

Dáta, ktoré sme zo simulácie získali boli: informácie o začiatku a o konci cesty vozidla ako aj čas v sekundách kedy sa tato cesta uskutočnila.

Z bodov o začiatku a konci cesty sme spravili priamku.

Súradnice referenčného, od ktorého sme počítali vzdialenosť vozidiel sme si určili pomocou mapy a na našom obrázku mal teda súradnice  $(x, y) = (460, 285)$ .

Pomocou týchto súradníc a súradníc vozidiel sme jednoducho vytvorili rovnicu vzdialenosti bodu od priamky.

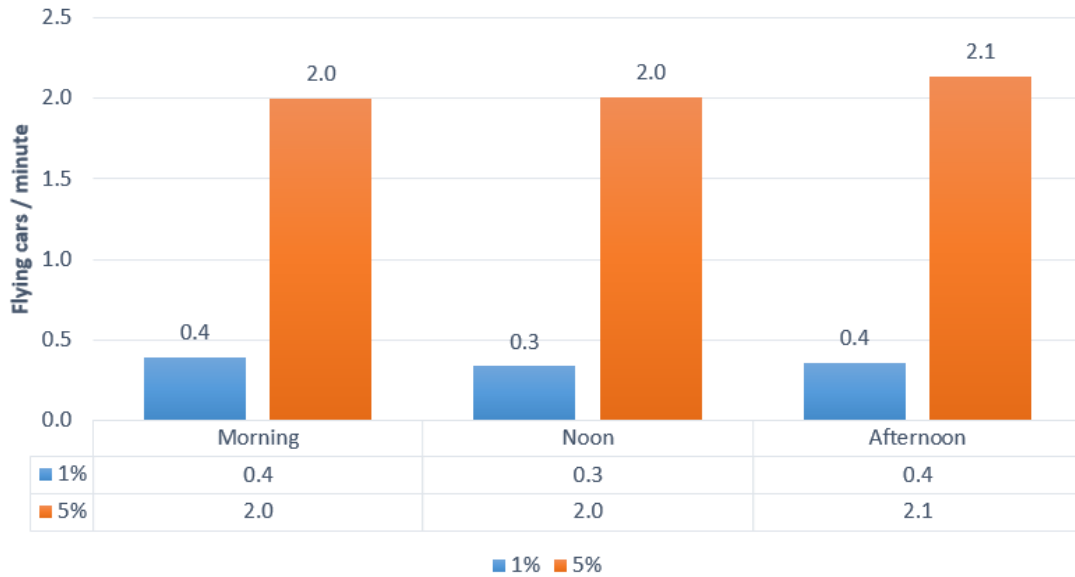
Všetky vozidlá, ktoré následne boli v kratšej vzdialenosti ako 150 metrov, čo na našich obrázkoch korešponduje s pixelovou vzdialenosťou 4.6875 pixela, sme spočítali a výsledok sme pre danú časť dna zapísali.

Následne sme počty vozidiel delili počtom hodín (dĺžky danej časti dňa v hodinách) aby sme získali priemerný počet vozidiel za hodinu. Taktiež sme počty delili minútami aby sme získali priemerný počet vozidiel za minútu.

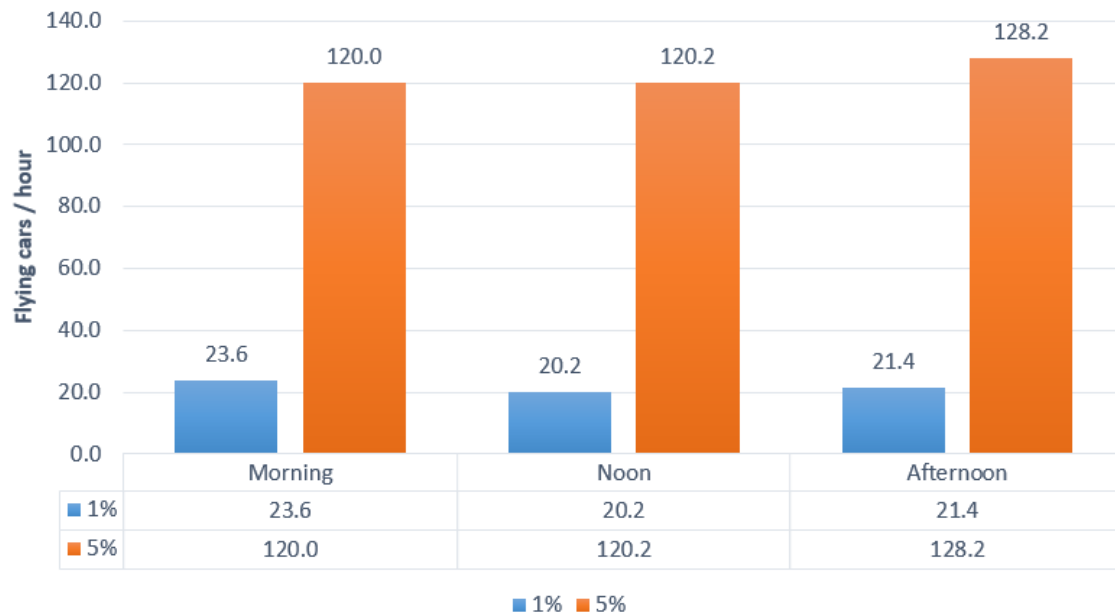
Výsledné diagramy s dátami vypadajú takto:

Všetky diagramy a dáta, ktoré sme získali sú verejne dostupné na Githube<sup>[9]</sup> v priečinku `gathered_data`.

### Flying cars / minute in 150 m radius [1%, 5%]

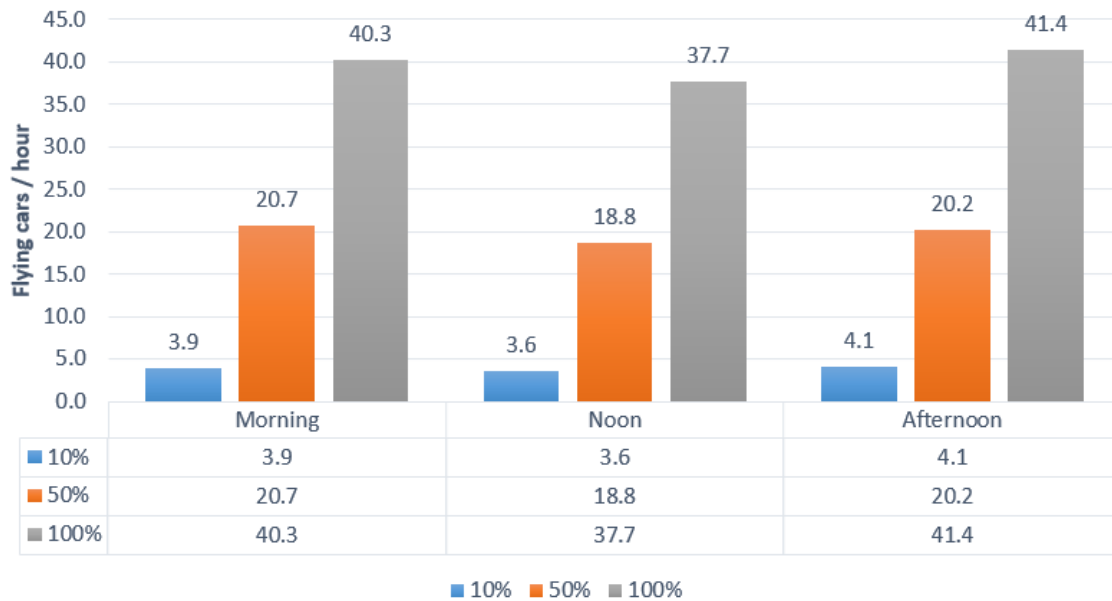


### Flying cars / hour in 150 m radius [1%, 5%]

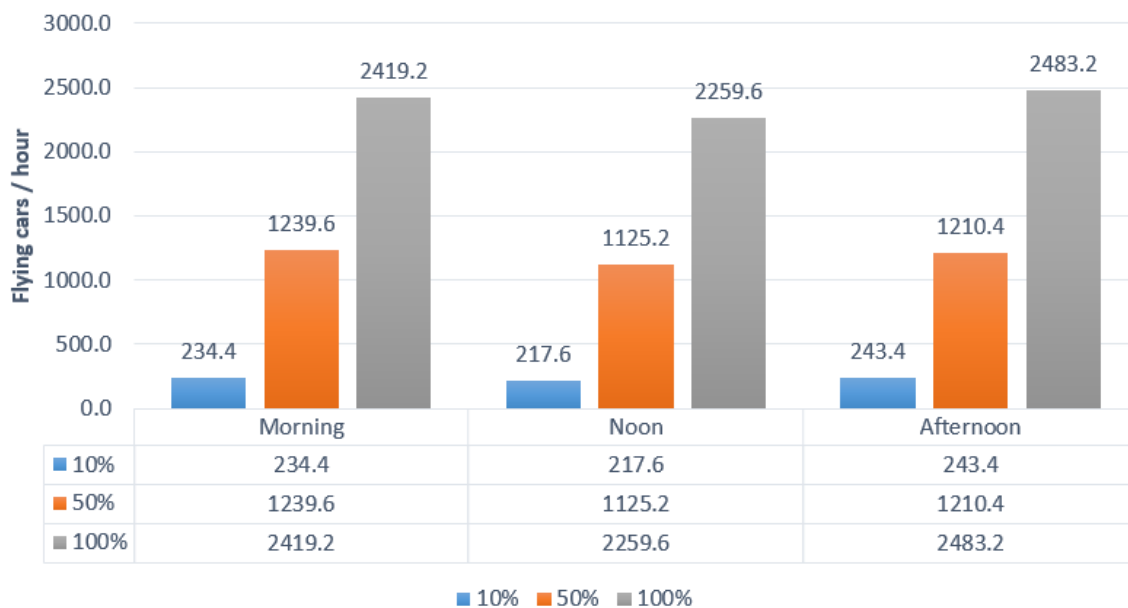


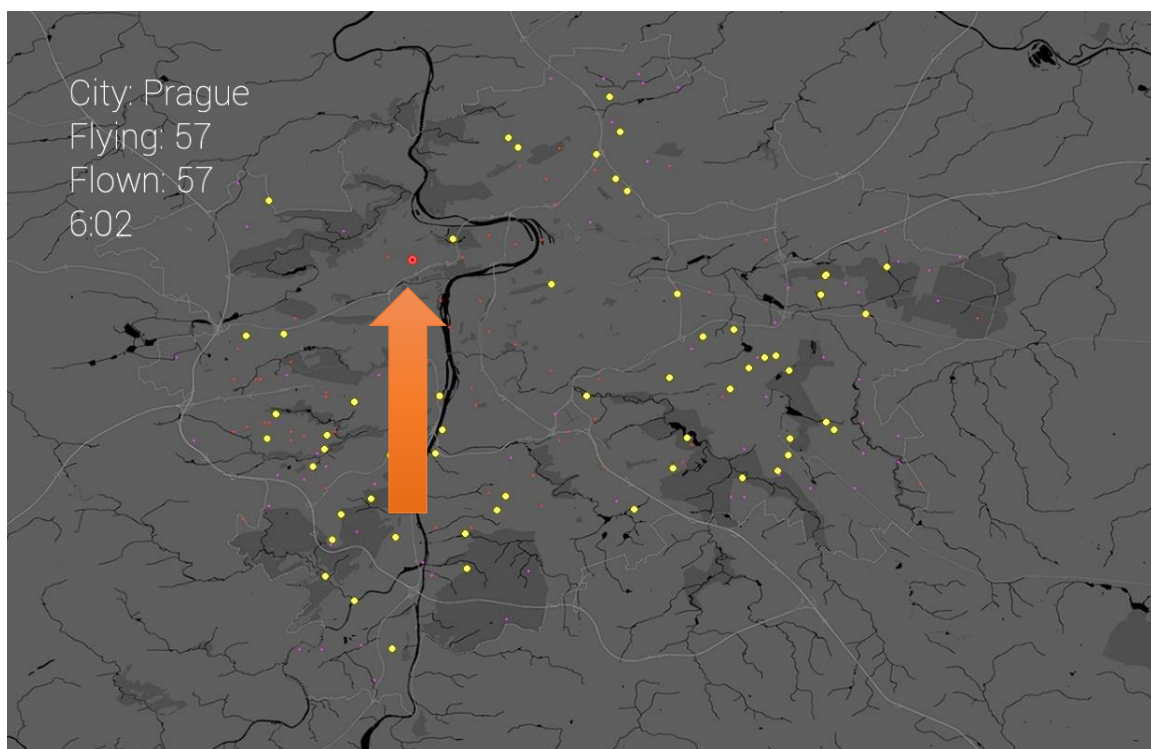
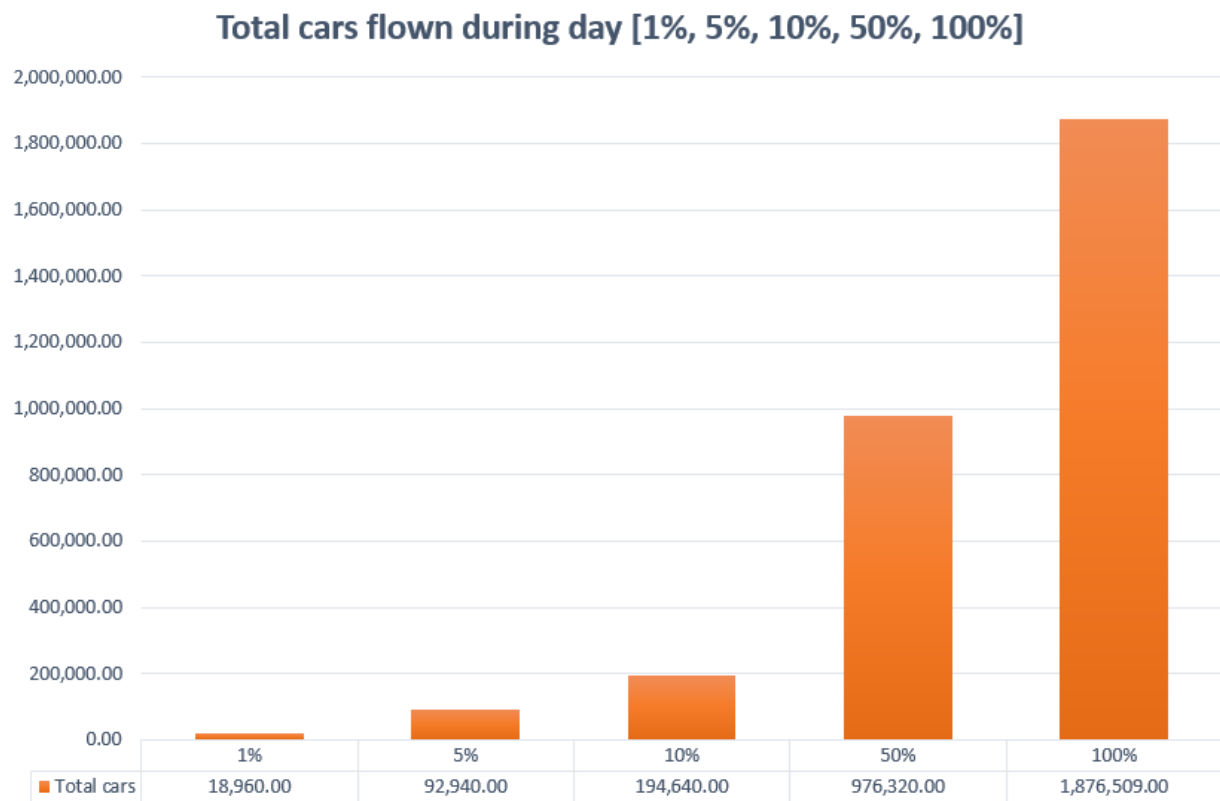


### Flying cars / minute in 150 m radius [10%, 50%, 100%]



### Flying cars / hour in 150 m radius [10%, 50%, 100%]





**Obrázok č.4** Zdroj: Snímok z videa<sup>[10]</sup> simulácie, na ktorom je vyznačený referenčný bod FEL ČVUT, Dejvice

Zo získaných dát môžeme vyčítať, že priemerný počet preletov vo vzdialenosti 150 metrov v okolí FEL ČVUT DEJVICE by bol pri jednom percentu lietajúcich áut približne 23,6 ráno a 21,4 večer. Čo je približne 0,4 lietajúcich vozidiel za minútu a teda každé 3 minúty by v danom okolí preletelo lietajúce vozidlo.

Lietajúce autá by takýmto spôsobom dokázali prispievať k zvukovému znečisteniu mesta. Ale ak by dané lietajúce vozidlá lietali vo väčších výškach a robili všetko preto, aby vydávali čo najmenej zvuku, poprípade by lietali po určených vzdušných trasách a počet týchto vozidiel sa držal pod určitým limitom, je predstaviteľné aby niečo takéto mohlo teoreticky fungovať. Najväčšie iniciatívy v tejto oblasti majú firmy, ktoré sa snažia tieto lietajúce vozidlá presadzovať vo forme lietajúcich taxíkov a údajne majú prvé oficiálne testy začať v roku 2020.

Nakoniec je to ale samotné mesto a ľudia, ktorí rozhodnú o budúcnosti tejto formy dopravy, ktorá má svoje výhody aj nevýhody.

Záver tejto práce tvoria vytvorené dáta, vizualizácia simulácie a kód poskytnutý na vytvorenie podobnej simulácie.

Dáta si už môže každý interpretovať podľa svojej potreby.

## REFERENCIE

1. VAN ROSSUM, Guido, DRAKE Fred, Python Reference Manual, PythonLabs, Virginia, USA, 2001. Available at <http://www.python.org>
2. Bradski G. The OpenCV Library. Dr Dobb's Journal of Software Tools. 2000.
3. OLIPHANT, Travis, Numpy: A guide to NumPy, USA: Trelgol Publishing 2006.
4. Český statistický úřad, Sčítání lidu, domů a bytů - Hlavní město Praha - analýza výsledků – 2011. Dostupné na [https://www.czso.cz/csu/czso/104135-13-n-k3018\\_2013-20](https://www.czso.cz/csu/czso/104135-13-n-k3018_2013-20)
5. Technická správa komunikací hl. m. Prahy, a. s., Ročenka dopravy 2017 str:7, Dostupné na <http://www.tsk-praha.cz/wps/portal/root/nabidka-sluzeb/rocenky>
6. Český statistický úřad, Krajská správa ČSÚ v hl. m. Praze, Obyvatelstvo, Dostupné na <https://www.czso.cz/csu/xa/obyvatelstvo-xa>
7. Technická správa komunikací hl. m. Prahy, a. s., Ročenka dopravy 2017 str:14, Dostupné na <http://www.tsk-praha.cz/wps/portal/root/nabidka-sluzeb/rocenky>
8. Google Inc., [software] <https://mapstyle.withgoogle.com>
9. Repozitár: Flying-cars-simulation-in-Prague-with-Python, <https://github.com/PeterKillerio/Flying-cars-simulation-in-Prague-with-Python>
10. Dočasné video simulácie dostupné na YouTube pod názvom Flying cars in Prague [unedited]