

České vysoké učení technické v Praze  
Fakulta elektrotechnická

katedra počítačů

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Filip Bouška**

Studijní program: Otevřená informatika (bakalářský)  
Obor: Softwarové systémy

Název tématu: **Validace simulátoru davu**

Pokyny pro vypracování:

Seznamte se s existujícím simulátorem davu AgentCrowd. Po konzultaci s vedoucím práce naměřte reálné parametry vybrané lokality. Tuto lokalitu namodelujte v simulátoru a porovnejte simulované výsledky s reálným chováním davu. Zmapujte dostupné metody porovnání podobnosti trajektorií, vybranou metodu implementujte a změřte chování simulovaného davu pro různá nastavení parametrů simulátoru.

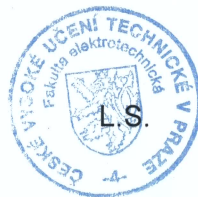
Seznam odborné literatury:

Ondřej Hrstka, Jan Drchal, Vojtěch Micka, Michal Štepanovský, Zdeněk Buk, Miroslav Čepěk: Traffic Flow Modeling Project Report v.1, N62909-12-1-7097

Vedoucí: Ing. Zdeněk Buk, Ph.D.

Platnost zadání: do konce letního semestru 2014/2015

  
doc. Ing. Filip Železný, Ph.D.  
vedoucí katedry

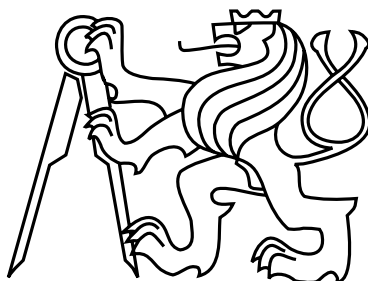


  
prof. Ing. Pavel Ripka, CSc.  
děkan

V Praze dne 25. 2. 2014



České vysoké učení technické v Praze  
Fakulta elektrotechnická  
Katedra počítačů



Bakalářská práce

## **Validace simulátoru davu**

*Filip Bouška*

Vedoucí práce: Ing. Zdeněk Buk, Ph.D.

Studijní program: Otevřená informatika, prezenční, Bakalářský

Obor: Softwarové systémy

5. ledna 2015



## Poděkování

Tímto bych rád poděkoval vedoucímu této bakalářské práce, Ing. Zdeňku Bukovi, Ph.D., za cenné rady, připomínky, za trpělivost a podporu. Dále bych chtěl poděkovat všem, kteří mě při tvorbě této práce jako i po dobu celého studia podporovali a motivovali k další činnosti.



## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 5. 1. 2015

.....





# Abstract

Nowadays, computer crowd simulation has become increasingly more necessary. They are used for urban planning, building or location security testing, as well as in computer games or movies. It is important for simulations to be as accurate as possible and to correspond with real situations. For many simulators, their validation is not sufficient enough or it is missing entirely. In this thesis, I deal with the verification of the proper functionality of the simulator AgentCrowd. To verify my findings, I created two scenarios. The first scenario focuses on examining specific agent's trajectories. The second scenario substantiates the simulator from a general and statistical point of view. Both of the two scenarios are divided into several sub-experiments. Through these experiments, I ascertained the differences between the simulator and real situations.

# Abstrakt

Počítačové simulace davu lidí jsou v dnešní době stále více potřebné. Využívají se při urbanistickém plánování, při simulacích testování bezpečnosti objektů nebo lokací a dále například v počítačových hrách nebo filmech. Důležité je, aby simulace byly co nejpřesnější a aby co možná nejvíce odpovídaly skutečným situacím. U mnoha simulátorů však není jejich validace příliš obsáhlá nebo chybí úplně. V této práci jsem se zabýval ověřením správné funkčnosti simulátoru AgentCrowd. Pro jeho validaci jsem provedl dva scénáře. První je zaměřen na zkoumání konkrétních trajektorií agentů. Druhý ověřuje simulátor z obecnějšího a statistického hlediska. Každý z obou scénářů je rozdělen na několik dílčích experimentů. Pomocí těchto experimentů jsem zjišťoval odlišnosti simulátoru od reálných situací.



# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
1.1	Simulace davu . . . . .	1
1.2	Popis problému . . . . .	2
1.3	State of the Art . . . . .	2
<b>2</b>	<b>Analýza a návrh řešení</b>	<b>5</b>
2.1	Úvod . . . . .	5
2.2	Popis simulátoru . . . . .	5
2.2.1	Nastavení simulátoru . . . . .	5
2.2.2	Druhy agentů . . . . .	6
2.3	Scénář 1 . . . . .	7
2.4	Scénář 2 . . . . .	7
<b>3</b>	<b>Realizace</b>	<b>11</b>
3.1	Scénář 1 . . . . .	11
3.1.1	Trajektorie . . . . .	12
3.1.2	Heat maps . . . . .	14
3.2	Scénář 2 . . . . .	14
3.2.1	Cestovní časy . . . . .	17
3.2.2	Časy opuštění scénáře . . . . .	20
3.2.3	Hodnocení průzkumem . . . . .	20
<b>4</b>	<b>Výsledky</b>	<b>23</b>
4.1	Scénář 1 . . . . .	23
4.1.1	Trajektorie . . . . .	23
4.1.2	Heat maps . . . . .	23
4.2	Scénář 2 . . . . .	24
4.2.1	Cestovní časy . . . . .	24
4.2.2	Časy opuštění scénáře . . . . .	25
4.2.3	Hodnocení průzkumem . . . . .	25
<b>5</b>	<b>Závěr</b>	<b>29</b>
5.1	Shrnutí . . . . .	29
5.2	Možná rozšíření . . . . .	29
	<b>Literatura</b>	<b>31</b>

**A Obsah CD**

**33**

# Seznam obrázků

2.1	Potenciálová mapa . . . . .	6
2.2	Mapa scénáře 1 . . . . .	8
2.3	Mapa scénáře 2 . . . . .	9
3.1	Detekce hýbajících se osob . . . . .	11
3.2	Detekce hýbajících se osob . . . . .	11
3.3	Referenční snímek pro první scénář . . . . .	12
3.4	Ukázka trajektorie . . . . .	13
3.5	Vybrané trajektorie z videa . . . . .	13
3.6	Vzdálenosti trajektorií z videa a trajektorií pro daný typ agenta. Čím menší sloupec, tím menší je vzdálenostní rozdíl mezi měřenými trajektoriemi. . . . .	15
3.7	Trajektorie video . . . . .	15
3.8	Trajektorie simulátor . . . . .	15
3.9	Počty agentů, kteří vstoupili do mapy scénáře pro každou ze sedmi entry zón . . . . .	16
3.10	Počty agentů, kteří odešli z mapy scénáře pro každou ze sedmi exit zón . . . . .	16
3.11	Entry/exit zóny scénáře 2 . . . . .	16
3.12	Ukázka simulace scénáře 2 . . . . .	17
3.13	Mapa scénáře 1 - trajektorie na nichž byly měřeny časy . . . . .	18
3.14	Diagram vyjadřující počet agentů podle velikosti rozdílů naměřených a simulovaných časů . . . . .	20
3.15	Body vyjadřující, kdy lidé (dolní body) a agenti (horní body) opustili mapu scénáře . . . . .	20
4.1	Trajektorie video . . . . .	24
4.2	Trajektorie simulátor . . . . .	24
4.3	Příklad neobvyklé trajektorie. Tato trajektorie vybočuje z průměrných trajektorií pro danou dvojici entry/exit zón. . . . .	25
4.4	Heat map vygenerovaná trajektoriemi z videa. Čím světlejší místo, tím více lidí jím prošlo. . . . .	26
4.5	Heat map vygenerovaná trajektoriemi ze simulátoru. Čím světlejší místo, tím více lidí jím prošlo. . . . .	26
4.6	Obrázek zobrazuje, v čem se liší heat mapa generovaná trajektoriemi z videa a heat mapa generovaná simulátorem. Pokud by byly obě stejné, viděli bychom pouze černý obrázek. Bílá barva značí jejich rozdíl. . . . .	26

- 4.7 Rozdíly mediánů časů naměřených v simulátoru a referenčních časů. Čím blíže je hodnota nulové ose tím menší je rozdíl oproti referenčnímu času a tím pádem daná konfigurace více odpovídá skutečnosti. . . . . 27
- 4.8 Časy naměřené v simulátoru pro všechny konfigurační soubory a pro každou trasu. Pro porovnání je u každé trasy nejvíce vlevo zobrazen referenční čas. . . 27
- 4.9 Časy odpovídají, diagram je ve tvaru Gaussovy křivky. Na pravém obrázku jsou body vyjadřující, kdy lidé (dolní body) a agenti (horní body) opustili mapu scénáře. . . . . 27
- 4.10 Časy neodpovídají a časové rozdíly jsou větší. Na pravém obrázku jsou body vyjadřující, kdy lidé (dolní body) a agenti (horní body) opustili mapu scénáře. 28

# Seznam tabulek

3.1	Referenční vzdálenosti trajektorií získaných z videa. Čím menší je číslo, tím více jsou si měřené trajektorie podobné. . . . .	14
3.2	Průměrná vzdálenost trajektorií. Čím menší je číslo, tím více jsou si měřené trajektorie podobné. . . . .	14
3.3	Referenční časy . . . . .	18
3.4	Rozdíly časů . . . . .	19
4.1	Všech pět trajektorií z videa jsem navzájem porovnal abych získal referenční vzdálenosti. Čím menší je číslo, tím více jsou si měřené trajektorie podobné. . . . .	23
4.2	Trajektorii od každého druhu agenta jsem změřil se všemi trajektoriemi z videa. Zde jsou naměřené vzdálenosti. Čím menší je číslo, tím více jsou si měřené trajektorie podobné. . . . .	24
4.3	Průměrné hodnocení, které udělili účastníci v provedeném průzkumu. Vidíme, že nejlepšího hodnocení dosáhl agent typu <i>PotentialAwareAgent</i> . . . . .	25
A.1	Obsah CD . . . . .	33





# Kapitola 1

## Úvod

### 1.1 Simulace davu

Na úvod bych nejdříve vysvětlil, co to vlastně simulace davu je. Obecně je počítačová simulace pokusem o vymodelování reálného světa či hypotetické situace za pomoci počítače tak, aby bylo možné studovat tento systém a vysledovat jak funguje. Simulace davu je proces simulace pohybu velkého množství subjektů. Subjekty - také nazývané agenti - získávají umělou inteligenci, na základě níž se rozhoduje o jejich chování. Tato umělá inteligence může být založena na jedné nebo více funkcích, jako je zrak, sluch, emoce, úroveň agresivity, množství energie nebo další [9]. V každém simulátoru se používá jiný přístup k získání umělé inteligence agentů. Každý agent má svůj cíl - úkol, kterého se snaží dosáhnout - splnit. Při vykonávání svého úkolu potkává další agenty, se kterými navzájem interaguje.

V dnešní době vzniká významný počet aplikací, které dokáží simulovat pohyb velkého množství agentů. Simulace davu lidí se využívá v mnoha odvětvích, jako jsou kinematografie, počítačové hry, bezpečnostní plánování, či plánování prostoru. Ve všech těchto případech je snaha o to, aby simulace byla co možná nejvíce odpovídající realitě. Pro všechny zmíněné kategorie jsou důležité rozdílné aspekty simulace. Například při bezpečnostním plánování je důraz na reálnost velmi zásadní. Vezmeme-li odvětví kinematografie, hodí se simulátory davů při tvorbě 3D grafiky v davových scénách, ve kterých vystupují až tisíce lidí. Například ve válečných filmech jsou záběry na armády pochodující po bitevním poli a natáčení scén s lidmi by bylo velmi finančně i časově náročné a je tedy výhodné scénu vytvořit pomocí simulace. Jako příklad bezpečnostní simulace můžu uvést aplikace, které zjišťují bezpečnostní problémy dané lokace jako průběh evakuace lidí při požáru [4]. Při evakuaci jde o životy lidí a důraz na provedení simulace, co nejvíce odpovídající skutečnosti, je zde velký. Další využití simulátoru může sloužit k návrhu lokace tak, aby se v ní lidé mohli efektivně pohybovat. Při architektonickém návrhu nákupního centra, vlakového nádraží nebo stanice metra se musí počítat s tím, že se v navrhované lokaci bude pohybovat velké množství lidí a je důležité, aby se zde lidé mohli rychle pohybovat a navzájem se neomezovali. Simulátory se též používají při tzv. urban planning neboli plánování města [3]. V další řadě se algoritmy z těchto simulátorů mohou použít v počítačových hrách, například pro simulaci chování vojska v historické strategii nebo pro simulaci městského ruchu v akční střílečce [2, 5]. V poslední době je vyvíjen čím dál větší důraz na realitu hry. Ať už ve vizuálním smyslu, jakožto kvalita

grafiky, tak i ve smyslu chování umělé inteligence. Rychlost simulace je zde velmi důležitá, protože se mění podle chování hráče.

## 1.2 Popis problému

Pro všechna tato využití je potřebné, aby výstup simulátoru byl co nejvíce podobný opravdové situaci v reálném světě. Jelikož chování člověka nelze přesně přenést na počítačový algoritmus, není lehké tyto simulátory nastavit. Nemůžeme jasně definovat, jaké chování člověka je reálné a normální. Dav lidí se chová jinak při běžné chůzi po ulici nebo při evakuaci, požáru či při demonstracích. Velké množství simulátorů je tedy uděláno ve smyslu, jak by lidé teoreticky měli chodit. Tento přístup ovšem nezaručuje opravdu validní simulaci, pokud není ověřen na reálných datech. Například evakuace lidí je velmi často simulovaný scénář, ale v reálném světě tato situace nenastává příliš často, a jestliže nastane, tak je velmi složité získat potřebná data pro následovné porovnávání. To je důvod, proč u většiny simulátorů davů neexistuje jejich validace.

V této práci se budu zabývat ověřením, zda funkčnost simulátoru AgentCrowd odpovídá realitě, neboli jeho validací pro následující použití. K ověření simulátoru jsem navrhl dva scénáře. První scénář zkoumá simulaci na úrovni porovnávání trajektorií na rozdíl od druhého, který pojímá větší oblast a zaměřuje se na porovnání simulátoru a reality z globálnějšího pohledu. Pro tyto scénáře jsem se pokusil najít metriky, pomocí kterých je možno měřit parametry simulace. Tyto parametry jsem následně porovnával s daty získanými z natočených videí. Tato porovnání jsem prováděl pro různá nastavení simulátoru.

## 1.3 State of the Art

Momentálně používané metriky validace simulace jsou převážně statistiky jako maximální cestovní čas, zastoupení pomalých chodců nebo pro více specifické simulátory evakuace např. počet lidí evakuovaných v určitém čase, hustota distribuce agentů přes sadu polygonálních oblastí [1, 6]. Kromě statistického pohledu se k validaci přistupuje také srovnáním trajektorií. Např. článek [10] zkoumá šest různých metod pro porovnávání trajektorií. Zin bere trajektorii agenta generovanou simulátorem a porovnává ji s trajektorií člověka v reálné situaci natočeného na videu [6]. Toto porovnání provádí tak, že určuje vzdálenost reálné a simulované trajektorie. Následně se tuto vzdálenost snaží minimalizovat změnou sady parametrů v algoritmu simulátoru. Pro každou sadu parametrů se nejprve snažili simulovat všechny osoby ve stejnou dobu, ale zjistili, že generované trajektorie jsou zcela odlišné od trajektorií získaných z analýzy reálného videa. Jako důvod uvádí, že, v případě pokud jde mnoho jedinců současně, dochází k zvětšování rozdílů v simulovaných a reálných trajektoriích, což je známo jako efekt motýlího křídla a výsledkem je úplně jiný scénář. Validací simulace se dále zabývá Lerner [7]. Tvrdí, že měření „look-and-feel“, tedy měření pouze na základně vizuálního vjemu, může z větší vzdálenosti vypadat přirozeně, i když při bližším zkoumání přirozeně nevypadá. Lerner pozoruje agenty z bližšího pohledu. Zkoumá trajektorie a atributy, které popisují stav agenta. Porovnává agenty ze simulace s lidmi natočenými na videu a každému agentovi přiděluje hodnocení. Toto hodnocení odpovídá procentům, jak moc agentovy parametry odpovídají parametrům lidí z videa. Další zajímavý přístup zvolil Pelechano

[8]. Provádí čtyři různé pokusy. V každém pokusu nechá dobrovolníka, který používá sadu pro virtuální realitu, aby se zúčastnil určitého scénáře, jako např. evakuace. Scénář je vizualizován a zaplněn simulovanými agenty. Při každém pokusu je algoritmus pro simulaci agentů jiný. Porovnává tzv. přístupy „social force“, „rule based“, „cellular automata“ a „HiDAC“. Ve své práci však nehodnotí tyto 4 přístupy, ale zaměřuje se na pocity dobrovolníků, kteří se nacházeli v simulaci. Tento přístup by se však mohl dále rozvinout a využít i pro validaci samotných simulátorů.



## Kapitola 2

# Analýza a návrh řešení

### 2.1 Úvod

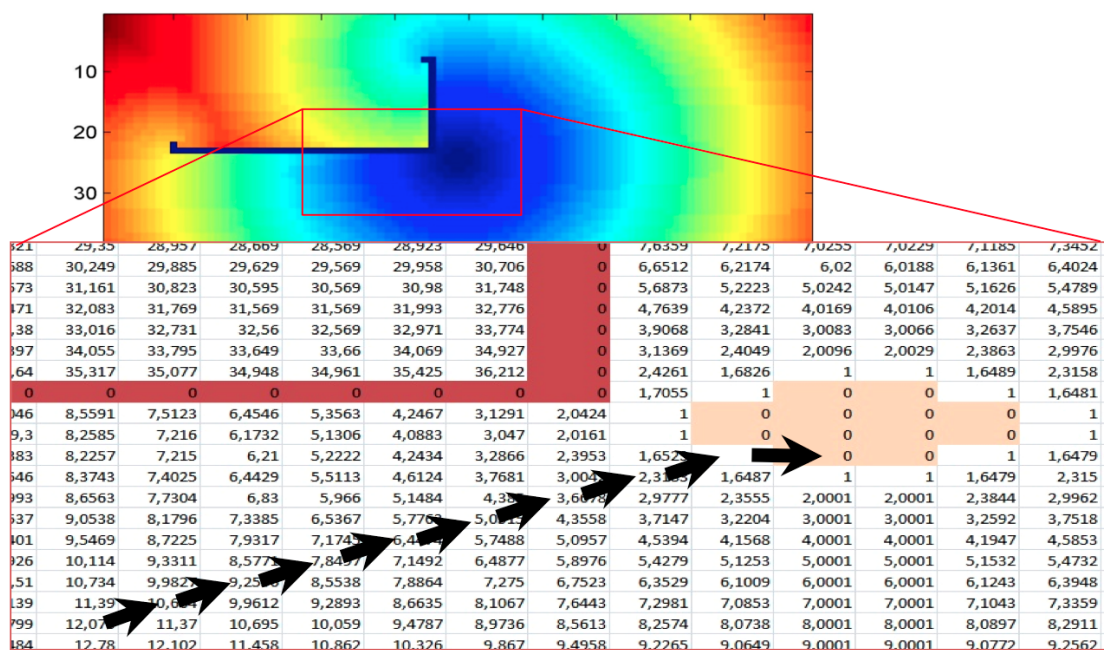
Nebudu se zde do detailu zabývat tím, jak funguje simulátor AgentCrowd. Budu zkoumat, jak již nastavený simulátor funguje a budu porovnávat naměřená data z videí s datovým výstupem ze simulátoru. K ověření simulátoru AgentCrowd bych chtěl přistupovat takto: Provedu dva různé scénáře. Každý scénář bude k validaci přistupovat jiným způsobem. První scénář bude simulovaná data zkoumat na bázi trajektorií a druhý bude data vyhodnocovat statisticky. Lokace pro oba scénáře se budou lišit, jelikož při každém scénáři budou důležité odlišné aspekty naměřených dat. Jelikož nelze předpokládat, že by se povedlo v simulátoru nasimulovat chování jednotlivých lidí úplně do detailu, rozhodl jsem se tyto výjimky částečně zanedbat. Například člověk, který telefonuje a pouze chodí do kruhu, bude ve videu zaznamenán, ale v simulátoru není možné takovéto chování nastavit či simulovat. Nastavení simulátoru je tedy nutno zjednodušit, ale pouze v té míře abychom nezacházeli do příliš modelové situace.

### 2.2 Popis simulátoru

Dříve, než více popíšu oba zmíněné scénáře, bych popsal, jaké nastavení lze v simulátoru AgentCrowd provádět. Simulátor je navržen tak, že se agenti pohybují po tzv. voxelech nebo též buňkách. Každý tento voxel má danou velikost. Mapa je tedy tvořena maticí čísel, kde každé číslo představuje určitý voxel. Algoritmus rozhodování agentů, kam mají jít, je v simulátoru AgentCrowd řešen pomocí dvojrozměrného pole s buňkami. Jedná se o tzv. potenciálově zaměřený přístup [4]. Každá tato buňka obsahuje informaci, zda je volná nebo již obsazená některým agentem. Dále obsahuje informaci o potenciálu, který určuje míru výhodnosti proto, aby na ni agent přešel. Tato výhodnost se vypočítává z pozice buňky vůči exit zóně a překážek, které stojí po cestě do ní (obrázek 2.1). Výstupem simulátoru je soubor, který obsahuje souřadnice všech trajektorií.

#### 2.2.1 Nastavení simulátoru

Zde jsou popsány soubory, ve kterých se provádí nastavení simulátoru pro prováděný scénář.



Obrázek 2.1: Potenciálová mapa

config.conf - Jedná se o hlavní konfigurační soubor, který obsahuje parametry simulace. Nastavují se zde parametry pro různé druhy agentů. Mezi ně např. patří rychlost agenta. Můžeme zde nastavit, kolik agentů se má vygenerovat za minutu a také definovat dvojice tzv. entry/exit zón pro daného agenta. Mezi obecné parametry simulace patří rozměr jedné buňky a počet vygenerovaných agentů k ukončení simulace. Dále je zde uveden počet řádků a počet sloupců matice scénáře.

obstacles.csv - Soubor udávající mapu scénáře. Jedná se o obdélníkovou matici o rozměrech nastavených v souboru config.conf. Pro každou hodnotu se po načtení souboru rozhodne, jaký druh buňky se zvolí. Buňka může být neprůchozí, tedy překážka. Nebo může být zvoleno z uvedených druhů povrchu. Ke každému druhu se váže jeho plánovací rychlostní koeficient. To znamená, že pokud jedna cesta vede přes trávu a je o trochu kratší než jiná cesta po chodníku, nemusí být tato kratší cesta pro agenta rychlejší.

EntryZoneV2.csv a ExitZoneV2.csv - Soubor udávající pozici dané entry nebo exit zóny. Stejně jako v případě obstacles.csv, jedná se o matici stejné velikosti, ale nyní je matice tvořena samými nulami a pouze v místě chtěné entry/exit zóny jsou místo nul jedničky.

## 2.2.2 Druhy agentů

V obou scénářích chci provádět několik simulací, kde v každé budu používat jiný druh agenta. Simulátor má tyto typy agentů: *AgentWithSocialForce* (social), *PotentialAwareAgent* (potential), *RandomizedPotentialAwareAgent* (randomized) a *CrowdPersonAgent* (crowd). Každý z těchto agentů používá jiný algoritmus k rozhodování, kam udělá další krok. Ve zkratce

popíši, jak se od sebe liší. *CrowdPersonAgent* se rozhoduje absolutně náhodně, *Randomized-PotentialAwareAgent* se pohybuje také náhodně, ale je již ovlivňován potenciálovou mapou a udržuje si přibližný směr pohybu. *PotentialAwareAgent* a *AgentWithSocialForce* jsou již sofistikovanější agenti a druhý zmíněný se od prvního liším tím, že je kromě potenciálů zároveň ovlivňován tzv. sociálními silami. To znamená, že je ovlivňován i ostatními agenty a při rozhodování, kam provést další krok bere v potaz i trajektorie ostatních agentů.

## 2.3 Scénář 1

Při rozmyšlení, jak nejlépe validovat funkčnost simulátoru AgentCrowd vyvstává několik problémů. Jak k validaci přistoupit respektive jakou metriku pro validaci zvolit. První úvaha byla, že bych natáčel nějaký prostor z výšky a poté z videa získal trajektorie osob, které prošly záběrem. Ty bych následně porovnal s výstupními trajektoriemi ze simulace provedené v simulátoru. Jelikož bych měl zaznamenány body všech trajektorií, mohl bych porovnat dvě trajektorie z reality i ze simulátoru. Hodnocením pro tento test by bylo porovnání těchto trajektorií vykreslených v programu Mathematica. Další možností by bylo zobrazit mapu lokace jako tzv. heat-map. Předpokládám, že výsledkem by měly být dva obrázky s podobnou heatmapou. Tyto heat mapy mohou odečíst a získat tak vizualizaci, která by znázorňovala rozdíl mezi oběma heat mapami. S touto možností však přichází technický problém. Nejsem schopný natáčet oblast o velikosti ulice či náměstí tak, aby pohled byl co nejvíce z výšky. Lokaci bych se pokoušel snímat z co největší výšky, aby docházelo k co nejmenšímu zkreslení. Snímaná oblast by byla tedy menší než pro druhý scénář, ale na videu by byla zaznamenaná celá. To by umožnilo přesnější zkoumání pohybu lidí i agentů. Video musí být natáčeno ze stativu, tak aby mohlo být analyzováno. Technicky možné je natočit většinou interiér nějaké větší stavby (nákupní dům). Při prvním seznámení s provedenou prací na validaci AgentCrowd jsem získal již natočené video, ve kterém byl natočen prostor před vchody do škol FA a FIT. Toto video však nebylo vhodné pro mé zamýšlené potřeby, jelikož snímaný prostor nebyl příliš členitý. Rozmýšlel jsem tedy, kde bych mohl natočit video vlastní. Ze všech možných lokací jsem zvolil atrium v Národní Technické knihovně (NTK). Je zde možnost natáčet z velké výšky - odpovídající výšce 5 pater. Natáčeno bude atrium v prostorách NTK viz obrázek 2.2. Toto „náměstí“ má dobře členitý profil. Lidé vcházejí betonovým schodištěm a poté jdou nejčastěji do výtahu nebo pokračují nahoru po schodech. Zároveň bude zaručen dostatečný počet lidí, kteří projdou touto lokací. Jejich počet však nebude příliš velký, tak aby následná analýza dat nebyla tolik rozsáhlá. Natočené video rozdělím na jednotlivé snímky a ty následně analyzuji pomocí programu Mathematica. Ten na každém snímku najde pohybující se osoby a zaznamenává jejich trajektorie. Takto získaná data můžeme poté porovnat s výstupními daty ze simulátoru. Metrikou využitou při porovnávání trajektorií se budu dále zabývat v kapitole 3.1

## 2.4 Scénář 2

Dále jsem uvažoval druhý scénář. V tomto scénáři zvolím lokaci větší a jako metriku zvolím jiné parametry než jsou trajektorie lidí jako v první úvaze. K analýze budu přistupovat z globálnějšího statistického pohledu. Zde jsem se rozhodl, jako lokaci zvolit ulici Technická v



Obrázek 2.2: Mapa scénáře 1

Dejvicích, kde se nachází vchody do škol VŠCHT a ČVUT. Tato lokace byla zvolena, jelikož se v ní nachází chodníky v obou směrech ulice, podél kterých je z jedné strany trávník a z druhé silnice, na které jsou zaparkována auta. Předpokládá se, že lidé chodí nejčastěji po chodníku a při vzájemných interakcích se navzájem vyhýbají a obcházejí. V krajních případech, při velké skupině, lidé sejdou z chodníku do trávy nebo do silnice tak, aby se vyhnuli bez kolize. Nastavení simulátoru provedu podle dat, která získám z natočených videí. Chtěl bych zde provést tato měření. Jako první změřím, jak dlouho trvá dojít ze stanoveného bodu do jiného. Následně provedu simulace s různým nastavením. Z výstupních souborů zjistím časy, které odpovídají simulovaným agentům pro stejné body. Tyto časy porovnam a určím, zda jsou podobné. Jako druhé změřím časy, ve kterých lidé nebo agenti opustí mapu scénáře. Časy porovnam a graficky znázorním. Z důvodů tohoto měření budu muset nastavit simulátor co nejpřesněji tak, aby co nejvíce odpovídal situaci v Technické ulici. Třetí přístup by byl pomocí formuláře, který rozešlu několika participantům. V tomto formuláři by byl zobrazen záznam se simulacemi provedených s různým nastavením. Participant by každou simulaci ohodnotil podle toho, jak moc by mu přišla realistická.

Pro potřeby této simulace bude nutné konfigurovat pět různých entry/exit zón. Entry zóna je zóna, ve které se objevují agenti a vstupují do modelovaného scénáře. Exit zóna je cílová zóna, do které agent směřuje po zrození v dané entry zóně.

V tomto scénáři natočím tyto zóny (čísla v závorce odpovídají obrázku 2.3):

- vstup do Technické ulice směrem od Národní technické knihovny. (1)
- vstup do Technické ulice směrem z metra Dejvická (2)
- vstup do Fakulty elektrotechnické (3)



- vstup do Fakulty strojní (4)
- vstup do Vysoké školy chemicko-technologické (5)



Obrázek 2.3: Mapa scénáře 2

Aby bylo měření co nejpřesnější, musejí se všechny zóny snímat ve stejný čas. Z videí získám počty lidí, kteří vstoupili a kteří odešli do nebo z dané entry/exit zóny. Vystává zde však problém, že by se ve zvolené lokaci již před začátkem záznamu nacházeli lidé. Zóna byla tedy před začátkem měření zaplněna lidmi. Simulátor Agent Crowd však po puštění simulace začíná za prázdného stavu. Toto vyřeším tak, že nechám simulátor nějakou dobu běžet a zpracovávat budu data generovaná až po uplynutí této doby.

Pro přesnější nastavení jsem musel do simulátoru dodělat některé vlastnosti. Jelikož z videí získám přesné časy, kdy lidé vstoupí do scénáře, mohu toho využít a nastavit stejné časy i pro simulaci. Simulátor však umožňuje nastavit pouze průměrný počet agentů, kteří se mají v dané zóně vygenerovat. Hodilo by se tedy doimplementovat vlastnost, která by při spouštění simulace načítala data o vstupních časech pro každé entry zónu. V simulaci by se tedy agenti generovaly ve stejný čas jako v natočených videích.

Simulátor AgentCrowd umožňuje pro každou dvojici entry/exit zón v konfiguračních souborech nastavit kolik průměrně agentů se má v dané entry zóně zrodit za jednu minutu. Jedná se tedy o náhodné rozložení četností. V první simulaci jsem tedy nastavil konfigurační data tak, aby průměrné hodnoty zrozených agentů odpovídaly skutečným naměřeným datům. Jelikož pro jednu entry zónu může existovat pouze její související exit zóna, ale člověk který vstoupil v jedné entry zóně může směřovat do jiné exit zóny než jiný člověk ze stejné entry zóny. Musel jsem tedy každou entry zónu rozdělit a přiřadit ke každé odlišnou exit zónu. Jelikož videa mapují pouze malou oblast entry zón, není z nich zřejmé, kam který člověk dojde. Nevíme tedy přesné parametry pro spuštění simulace. Počty agentů pro danou dvojici entry/exit zón tedy musím nastavit podle dat získaných z analýzy videí. Z nich získám četnosti agentů, kteří vstoupili nebo odešli ze scénáře. Podle těchto četností budu

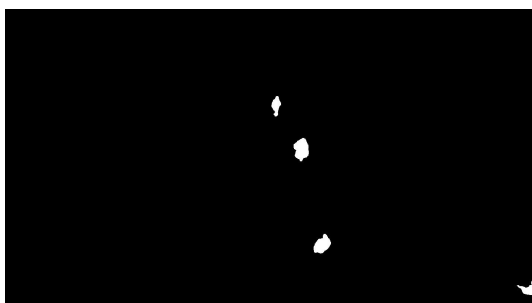
moci nastavit kolik agentů z jedné entry zóny má jít do konkrétní exit zóny. Nepodaří se mi zaručit úplně stejné podmínky, jako byly ve skutečnosti, ale při provedení více simulací by to nemělo mít zásadní vliv. Za účelem snazšího nastavování těchto časů vytvořím program pro generaci konfiguračních souborů.

## Kapitola 3

# Realizace

### 3.1 Scénář 1

V tomto scénáři jsem se rozhodl natočit video zachycující atrium v Národní technické knihovně. Natočené video jsem zpracovával následovně. Nejdříve bylo nutné video rozsekát na jednotlivé snímky, jelikož budu analýzu provádět v programu Mathematica a budu zkoumat video snímek po snímku. Za ideální počet snímků za vteřinu jsem zvolil dva. Video tedy rozsekám na snímky po půl vteřině. Z videa dlouhého necelých 30 minut mám tedy 3470 snímků. Zvolil jsem také referenční snímek (3.3). Tento referenční snímek neobsahuje žádné pohybující se osoby. Bude sloužit k porovnávání s ostatními snímky. Při detekci pohybujících se osob jsem postupoval následovně. Jelikož bylo video natáčeno ze stativu, mohl jsem odečíst zkoumaný snímek od referenčního a tím získat rozdíl, tedy například pohybující se osobu, která na referenčním snímku není. Aplikací různých funkcí, které Mathematica nabízí jsem získal výsledný snímek. Na snímku jsou zobrazeny všechny hýbající se osoby jako bílé objekty (obrázek 3.1). Pomocí dalších funkcí jsem získal souřadnice těchto objektů. Mám tedy souřadnice všech pohybujících se lidí pro daný snímek (obrázek 3.2). Tyto souřadnice jsem ukládal do souboru. Zároveň jsem k souřadnicím udávajícím polohu přidal údaj o čase - číslo analyzovaného snímku. Tento postup jsem aplikoval na všechny snímky. Výsledkem je 3470 souborů (jeden soubor pro jeden snímek). Každý z těchto souborů tedy obsahuje souřadnice a časový údaj všech osob, které se na snímku vyskytují.



Obrázek 3.1: Detekce hýbajících se osob



Obrázek 3.2: Detekce hýbajících se osob

Měl jsem připravená data pro všechny snímky a mohl jsem postoupit dále. Z těchto

souřadnic jsem tvořil konkrétní trajektorie. Při tom jsem postupoval takto. Vzal jsem data z prvního snímku a pro každou dvojici souřadnic (pro každé člověka) jsem hledal dvojici souřadnic z následujícího souboru, která je nejbližší dvojici první. Zároveň musí splňovat podmínku, že jsou od sebe vzdálené pouze do určité vzdálenosti. Dále rozdíl časů nesmí přesáhnout určitou hodnotu. To nám zaručí přesněji tvořené trajektorie. Kdybychom neměli časový údaj, mohla by se trajektorie utvořit špatně. Když osoba vyjde jednou exit zónou a za půl minuty vstoupí stejnou exit zónou osoba jiná. Jelikož rozdíl jejich pozic byl velmi malý, pokračovala by trajektorie prvního člověka tak, že by se spojila s trajektorií člověka druhého. Jejich časový rozdíl je však velký takže se místo spojení vytvoří trajektorie nová. Tento postup skrývá nevýhodu v tom, že při velkém počtu lidí ve snímku se nemusí trajektorie tvořit úplně přesně tak, jak vypadala ve skutečnosti. Pro účely validace však plně stačí, jelikož pro první část měření potřebuji jen několik konkrétních trajektorií a v druhé části naopak průměruji všechny trajektorie dohromady. Všechny výsledné grafy a obrázky jsou v kapitole 4.1.1.



Obrázek 3.3: Referenční snímek pro první scénář

### 3.1.1 Trajektorie

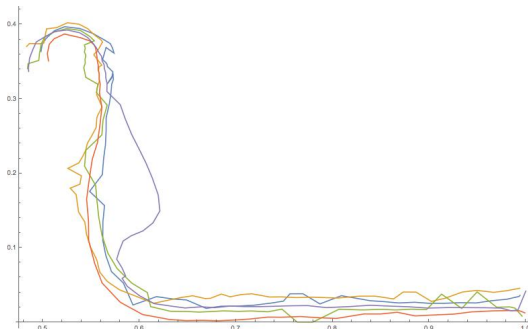
Při hledání správného přístupu k porovnávání trajektorií jsem použil metodu z [10]. Tento článek porovnává 6 různých způsobů měření podobnosti trajektorií. Jako jednu z nejvhodnějších metoda vyhodnocuje měření pomocí Euklidovské vzdálenosti (3.1). Pro své měření jsem tedy zvolil tuto metodu. Metoda měří vzdálenost dvou odpovídajících bodů z obou porovnávaných trajektorií. Toto měření se provede pro  $N$  bodů a všechny výsledky se sečtou. Následně se součet vydělí celkovým počtem bodů, takže získáme průměrnou vzdálenost obou trajektorií.

$$D(A, B) = \frac{1}{N} \sum_{n=1}^N \sqrt{(a_n^x - b_n^x)^2 + (a_n^y - b_n^y)^2} \quad (3.1)$$

Rozměry trajektorií zaznamenaných z videa a trajektorií ze simulátoru jsou odlišné. Musel jsem je tedy převést na stejné měřítko. Po převedení jsem získal trajektorie, jejichž souřadnice x i y se pohybují v rozmezí 0 až 1, kde 0 odpovídá nejmenší hodnotě z obou souřadnic a 1 hodnotě největší. Takže vlastně hodnota 1 je rozdíl x-ové největší a nejmenší souřadnice. Při přepočtu odpovídá hodnota 1 vzdálenosti přibližně 12 metrům - záleží na konkrétní trajektorii. Každá z trajektorií má odlišný počet bodů, kterými je definována. Při každém měření jsem body trajektorie interpoloval a získal funkci, která tuto trajektorii popisuje. Interpolaci jsem provedl pro souřadnice x i pro souřadnice y. Pomocí těchto funkcí jsem mohl získat odpovídající body z obou souřadnic a to i takové body, které nejsou přímo ve výčtu bodů, jež definují trajektorii. Počet bodů, u kterých jsem měřil vzdálenosti, odpovídal délce kratší z trajektorií. V budoucí práci by bylo možné měření provést s větším počtem bodů a tím získat přesnější údaje. Kromě rozdílů v poloze bodů jsem chtěl také měřit jejich rozdíl časový. Část prostoru, kudy lidé v NTK chodili, tvoří schody a lidé po nich chodí s jinou rychlostí než chodí po rovině. Porovnávání s trajektoriemi ze simulátoru, kde schody vymodelovat nelze, tedy nebylo možné. Mohl jsem vzít pouze ty části trajektorie, kde nejsou schody. Jelikož však schody tvoří podstatnou část trajektorie, jak je vidět na obrázku 3.4, rozhodl jsem se měřit vzdálenosti pouze prostorové.



Obrázek 3.4: Ukázka trajektorie



Obrázek 3.5: Vybrané trajektorie z videa

Nyní mohu přejít k samotnému měření. Postupoval jsem následovně: Ze všech trajektorií, která jsem získal analýzou videa, jsem jich pět vybral. Tyto trajektorie znázorňuje obrázek 3.5. Abych získal referenční hodnoty, změřil jsem vzdálenosti mezi těmito trajektoriemi. Trajektorii jsem poměřil s ostatními čtyřmi trajektoriemi. Tyto vzdálenosti naleznete v tabulce 3.1. Jsou to tedy průměrné vzdálenosti mezi dvěma porovnávanými trajektoriemi po celou jejich délku. Hodnoty jsou v jednotkové vzdálenosti, na kterou byli převedeny. Hodnota se může na první pohled zdát malá, ale pokud ji převede na metry tak hodnota 0,04 odpovídá přibližně 45 centimetrům.

Poté jsem postupně provedl simulace pro všechny druhy agentů. *CrowdPersonAgent* jsem z tohoto měření vyloučil. Simulátor zaznamená pouze trajektorie agentů, kteří dojdou do cíle a při nastavení tohoto druhu nedošel žádný agent. Provedl jsem tedy tři simulace. Ze všech tří jsem zvolil pět odlišných trajektorií. Každou trajektorii jsem poté porovnával s pěti

	video 1	video 2	video 3	video 4	video 5
video 1	-	0,0350224	0,0195657	0,0211143	0,0320279
video 2	0,0350224	-	0,0411644	0,0417831	0,0325165
video 3	0,0195657	0,0411644	-	0,0207646	0,0335506
video 4	0,0211143	0,0417831	0,0207646	-	0,0355586
video 5	0,0320279	0,0325165	0,0335506	0,0355586	-

Tabulka 3.1: Referenční vzdálenosti trajektorií získaných z videa. Čím menší je číslo, tím více jsou si měřené trajektorie podobné.

trajektoriemi z videa. Naměřené hodnoty jsou v tabulce 3.2. Graficky znázorněná vzdálenost každé z pěti trajektorií z videa a trajektorie pro daný typ agenta 3.6.

	video	social	potential	randomized
$\phi$	0,03130681	0,042220556	0,053055984	0,09091776

Tabulka 3.2: Průměrná vzdálenost trajektorií. Čím menší je číslo, tím více jsou si měřené trajektorie podobné.

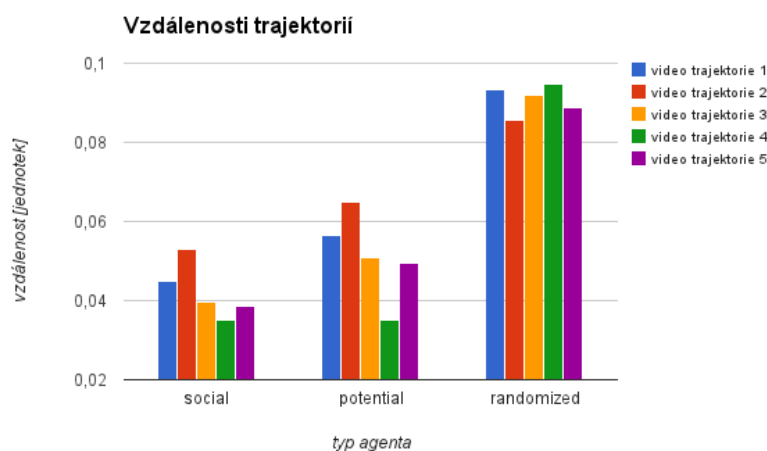
### 3.1.2 Heat maps

Pro analýzu jsem používal stejná data jako při měření výše, kde jsem si ze všech trajektorií vybíral pouze několik konkrétních. Zde jsem nejdříve ze všech zaznamenaných trajektorií odstranil ty, které by ovlivňovaly výsledek a nebylo je možné nasimulovat. Jedná se například o lidi sedící na lavičce, či lidé postávající na místě a pracující na terminálech. Výsledné trajektorie jsou znázorněny na obrázku 3.7.

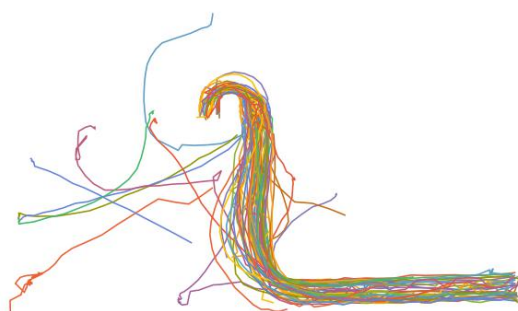
Jak si můžeme všimnout, tak trajektorie na obrázku 3.8 působí kostrbatým dojmem. To je způsobeno tím, že simulátor AgentCrowd není zamýšlen na simulace takto malých prostorů. Jeho primární užití je plánování oblastí o rozměrech až dvacetkrát větších, než je mnou prováděný scénář. Jak již bylo zmíněno, technická realizace scénáře s větší plochou však nebyla možná. Výsledné heat mapy naleznete v kapitole 4.1.2.

## 3.2 Scénář 2

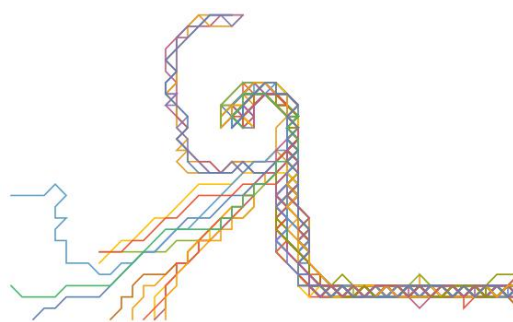
Problémem v prvním scénáři je, že pokud bychom chtěli souvisle sledovat celou mapu scénáře potřebovali bychom ji natáčet z veliké výšky, což nebylo technicky proveditelné. Natáčeli se tedy pouze lokace vstupu resp. výstupu lidí z mapované lokace. Natáčení proběhlo 17.12. 2013 od 16:20 až 16:30. Lokace zmíněné v 2.4 jsem tedy současně natáčel po dobu deseti minut. Získaná videa jsem si přehrál a pro každou zónu jsem počítal, kolik lidí v ní vejde 3.9 a kolik lidí v ní odejde 3.10. Pro přesnější nastavení simulátoru jsem zóny NTK a metro rozdělil na dvě samostatné entry/exit zóny. NTK horní a metro horní odpovídají chodníku, který je na straně, kde se nachází vstupy do škol FS a FEL. NTK dolní a metro dolní



Obrázek 3.6: Vzdařlenosti trajektorií z videa a trajektorií pro daný typ agenta. Čím menší sloupec, tím menší je vzdálenostní rozdíl mezi měřenými trajektoriemi.



Obrázek 3.7: Trajektorie video



Obrázek 3.8: Trajektorie simulátor

odpovídají chodníku, kde se nachází vstup do školy VŠCHT (obrázek 3.11). Dále jsem pro každého člověka zaznamenal přesný čas, kdy do scénáře vstoupí. Tyto časy pro každou entry zónu naleznete na příloženém CD ve složce *Data*.

Abych mohl v simulátoru nastavit přesné vstupní časy pro každého agenta tak, aby se zrodil ve stejném čase jako v reálně natočené situaci, musel jsem v simulátoru doimplementovat některé vlastnosti. Nově se může pro každou entry zónu přidat konfigurační soubor obsahující časy, ve kterých se mají agenti zrodit. Ke každé této časové hodnotě je přiřazeno číslo určující množství agentů, kteří se mají zrodit. Například takto: 03:24 3 - v třetí minutě a dvacáté čtvrté vteřině se mají vygenerovat 3 agenti. Jedná se o textový soubor a formát jeho názvu souboru musí odpovídat názvu entry zóny, pro kterou je určen a musí mít prefix *times\_*. Například tedy *times\_7\_V2.txt* bude určovat časy pro entry zónu s názvem *7\_V2*. Aby simulátor časy načetl, musí se do konfiguračního souboru přidat parametr *timing = 1*.

Dále jsem v simulátoru zavedl rozlišení buněk podle povrchu. V tomto scénáři jsem použil tyto typy povrchů: chodník (0), silnice (1), tráva (2) a překážka (3). Každému z těchto

FEL	48
FS	35
VŠCHT	57
NTK horní	104
NTK dolní	88
metro horní	56
metro dolní	69
celkem	457

Obrázek 3.9: Počty agentů, kteří vstoupili do mapy scénáře pro každou ze sedmi entry zón

FEL	24
FS	27
VŠCHT	28
NTK horní	56
NTK dolní	60
metro horní	139
metro dolní	121
celkem	455

Obrázek 3.10: Počty agentů, kteří odešli z mapy scénáře pro každou ze sedmi exit zón



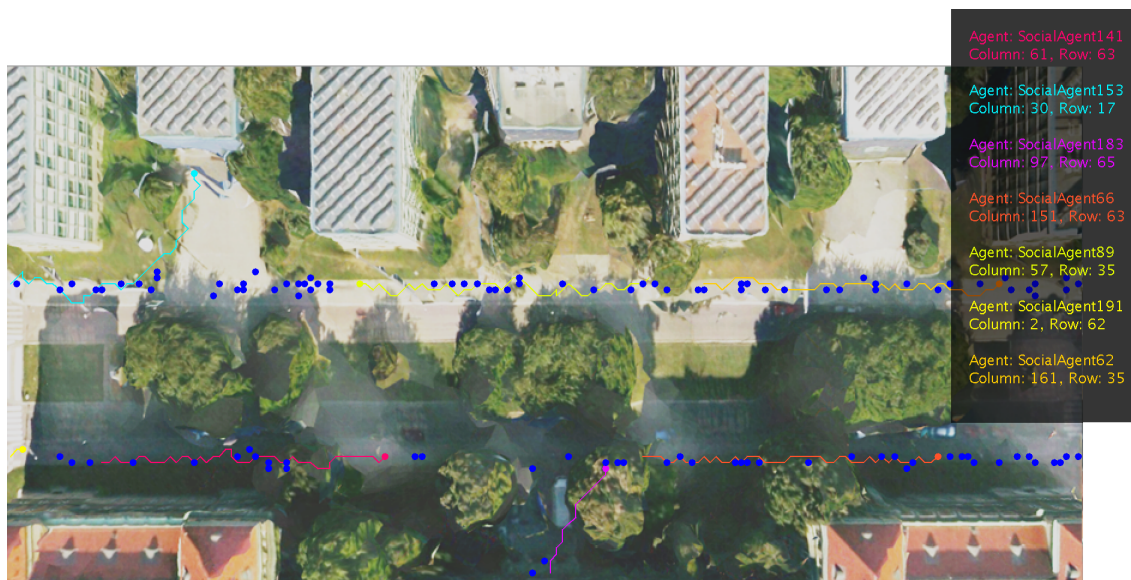
Obrázek 3.11: Entry/exit zóny scénáře 2

povrchů odpovídá jiná číselnice znázorněna v závorce za typem povrchu. Tyto hodnoty se nastavují v souboru *obstacles.csv*. Takovéto rozdělení povrchů slouží při rozhodování agenta, jakou trasu si pro svou chůzi vybere. Povrchy mají rozdílný koeficient, který určuje rychlost povrchu. Například toto číslo je pro chodník větší než číslo pro trávu, takže pokud jde agent z bodu A do bodu B a trasa po chodníku je delší než trasa po trávě, záleží na nastavení koeficientů pro to, jakou trasu si agent vybere. Koeficienty se nastavují v souboru *config.conf*. Toto počítání rychlostí a potenciálů je vlastně přenesení lidského uvažování do algoritmu. Když se člověk rozhoduje, zda jít delší cestou po chodníku nebo přejít kratší cestou po trávníku, záleží například, je-li tráva mokrá, je-li v trávníku bahno apod. O přesné hodnotě tohoto koeficientu nemá smysl uvažovat, jelikož může být pro každého člověka nastaven jinak. Koeficient jsem tedy experimentálně zvolil a následovným sledováním simulací jsem jej zpřesňoval.

Jakožto první výstup simulátoru pro mě byla vizualizace simulace. Pohledem jsem kon-



troloval, zda simulátor správně řídí agenty a nedochází k jejich nesmyslnému chování. Tato metoda, ačkoliv se jeví jako neměřitelná, mi stačila pro hrubé nastavení. Kontroloval jsem, zda agenti nezbytně nechodí mimo chodník (v trávě). Souvisle s tím jsem sledoval, jak se agenti vyhýbají, když se vedle sebe nevejdou na chodník. Dále jestli agenti volí nejrychlejší cestu ke svému cíli.

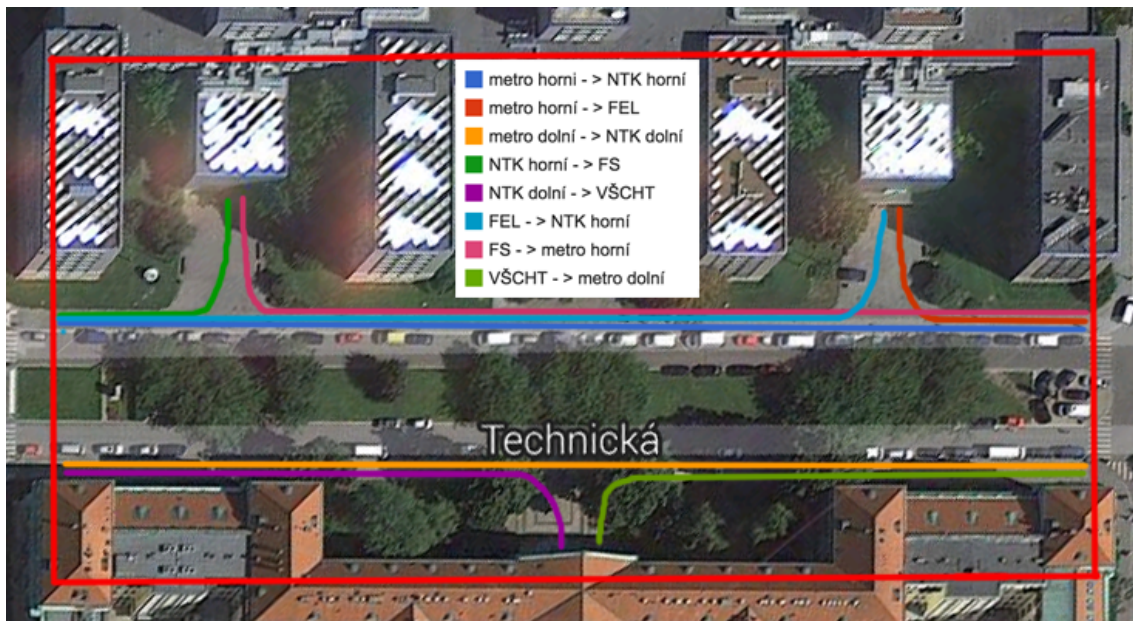


Obrázek 3.12: Ukázka simulace scénáře 2

### 3.2.1 Cestovní časy

V této sekci jsem měřil časy, za které ušli simulovaní agenti dané trasy. Celkem jsem měření prováděl na osmi trasách. Jak vypadají trajektorie měřených tras, vidíte na obrázku 3.13. Nejdříve jsem pěšky prošel každou z těchto osmi tras a pomocí stopky jsem změřil, za jakou dobu ji projdu. Při chůzi jsem se snažil jít stejně rychle, jako chodila většina lidí na ulici, tak aby naměřené časy co nejvíce odpovídaly průměrné době potřebné k průchodu dané trasy. Kromě stopky jsem též používal mobilní aplikaci, která pomocí GPS zaznamenávala ušlou vzdálenost a rovněž čas. Výsledné hodnoty jsem zprůměroval a zaznamenal jako referenční časy (tabulka 3.3).

Následně jsem prováděl simulace s různým nastavením simulátoru. Nyní vysvětlím, co vše jsem při simulacích nastavoval. První a nejdůležitější parametr v tomto měření bylo nastavení druhu agenta, který si při simulaci použije viz 2.2.2. Další z nastavovaných veličin byly rychlosti chůze agentů. V konfiguračním souboru jsem nastavil rychlost chůze agenta na hodnotu 3,75 mil/h, to odpovídá rychlosti 6 km/h. Tato rychlost odpovídá průměrné rychlosti, kterou jsem procházel dané trasy. Dále jsem nastavil odchylku rychlosti na hodnotu 1,4 mil/h (2,24 km/h). Tato hodnota umožňuje, že každý z agentů může chodit buď rychleji nebo pomaleji v rozmezí od 2,35 mil/h (3,76 km/h) do 5,15 mil/h (8,24 km/h). Což nám zaručuje podmínky více odpovídající skutečnému světu, kde též všichni lidé nechodí



Obrázek 3.13: Mapa scénáře 1 - trajektorie na nichž byly měřeny časy

metro horní → NTK horní	141 s
metro horní → FEL	39 s
metro dolní → NTK dolní	139 s
NTK horní → FS	40 s
NTK dolní → VŠCHT	80 s
FEL → NTK horní	120 s
FS → metro horní	124 s
VŠCHT → metro dolní	84 s

Tabulka 3.3: Referenční časy

konstantní rychlostí, ale svou rychlost chůze mění v závislosti na podmínkách v jakých se momentálně vyskytují.

Jak jsem zmiňoval výše, z videí nebylo možné určit, kam jaký člověk jde. Nemohl jsem tedy pro každého agenta, který vstoupí do scénáře, určit kam má jít, tak aby to odpovídalo skutečné situaci. Vycházel jsem tedy ze statistických údajů. Pro každou entry zónu jsem počítal rozložení agentů, kteří vstupují v této entry zóně. Uvedu na příkladu zóny FEL. Touto zónou vstoupilo po čas nahrávání videa (10 minut) celkem 48 lidí. Ze zóny FEL poté může člověk jít buď do zóny NTK horní nebo metro horní. Ostatní čtyři entry zóny jsem zanedbal, jelikož pravděpodobnost, že by do nich člověk šel, není příliš vysoká. Zónou NTK horní odešlo celkem 56 lidí a zónou metro horní odešlo celkem 139 lidí. Z těchto čísel jsem určil poměr s jakou pravděpodobností půjde člověk z lokace FEL do lokace NTK horní a s jakou pravděpodobností půjde do lokace metro horní. Poměry jsou:  $\text{NTK horní} - \frac{56}{(56+139)} = 0,28718$

a metro horní -  $\frac{139}{(56+139)} = 0,71282$ . To znamená, že člověk, který vstoupí v zóně FEL, půjde s pravděpodobností přibližně 29 procent do lokace NTK horní a s pravděpodobností 71 procent do lokace metro horní. S konkrétními čísly to tedy vychází  $0,28718 * 48 = 13,78464$  a  $0,71282 * 48 = 34,21536$ . Do lokace NTK horní půjde 14 lidí a do lokace Metro horní 34 lidí. Což odpovídá součtu 48 lidí, kteří vyjdou z lokace FEL. Všechny tyto poměry jsou zaznamenány v souboru *results* ve složce *Data* na příloženém CD.

Pro potřeby snazšího tvoření časových konfiguračních souborů jsem vytvořil aplikaci, která načte soubory pro každou entry zónu obsahující časy, kdy lidé vstupují do scénáře (tyto časy jsem získal z natočených videí). Aplikace následně tyto časy rozdělí podle stejného principu, který je popsán výše. Aplikace vygeneruje náhodné číslo v rozmezí 0 až 1. Pro příklad zóny FEL platí, že pokud je vygenerované číslo menší než 0,28718, přidělí se čas do souboru pro dvojici entry/exit zón FEL a NTK horní a pokud je větší než 0,28718 tak se čas přidělí do souboru pro dvojici FEL a metro horní. Vytvoří se tedy 16 souborů pro každou dvojici entry/exit zón, které jsem definoval.

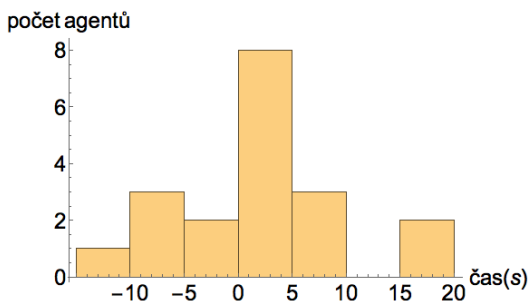
Pro analýzu dat získaných ze simulací jsem si vytvořil program. Ten načte výstupní soubor *agentlog.m* a vypíše časy všech agentů pro zadanou dvojici entry/exit zón. Celkově jsem provedl deset simulací. První čtyři simulace jsem provedl pro každý ze čtyř druhů agentů. Simulace při nastavení agenta typu *CrowdPersonAgent*, stejně jako ve scénáři 1, nevygenerovala žádná data, jelikož žádný agent nešel do svého cíle. Tento druh agenta jsem dále již nevyužíval. Při využití typu *RandomizedPotentialAwareAgent* byly časy diametrálně odlišné od referenčních a rozhodl jsem se ho také dále nevyužívat. Dále jsem provedl po třech simulace pro zbývající druhy agentů. Pomocí mého programu jsem získal časy pro všechny měřené trasy. Tyto časy jsou v souboru *results* ve složce *Data* na příloženém CD. Rozdíly časů získaných ze simulace a časů referenčních jsou v tabulce 3.4. Tabulka znázorňuje zprůměrované časy v sekundách všech osmi trajektorií. Čím jsou hodnoty v tabulce blíže nule, tím více odpovídají referenčním časům. Kladná hodnota znamená, že simulovaným agentům daného typu trvalo ujít danou trajektorii déle naopak záporná hodnota znamená, že agenti byli rychlejší a prošli trasu za menší čas.

potential config 1	-4,4 s
potential config 2	-5,9 s
potential config 3	-2,8 s
potential config 4	-1,4 s
social config 1	46,5 s
social config 2	47,9 s
social config 3	47,2 s
social config 4	47,1 s
randomized config 1	157,4 s

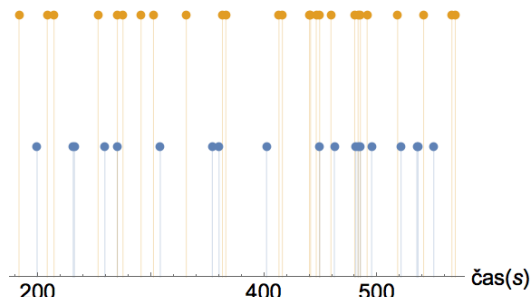
Tabulka 3.4: Rozdíly časů

### 3.2.2 Časy opuštění scénáře

Jako další experiment jsem porovnával časy, kdy lidé opouští scénář, s časy, kdy jej opouští agenti. Z videí jsem si zaznamenal časy, kdy lidé opustí danou zónou scénáře. Takto jsem zaznamenal časy pro zóny FEL, FS a VŠCHT. Tyto časy jsem poté porovnával s časy, ve kterých opustili scénář agenti. Porovnání probíhalo takto. Časy z videa jsem musel ořezat, jelikož po spuštění simulace je scénář prázdný a než přijde první agent do své exit zóny uplyne určitý časový úsek. Z časů z videa jsem tedy bral pouze časy, které byly až po tomto časovém úseku. Každému času z videa jsem přiřadil nejbližší z časů ze simulátoru. Výsledek měl být graf ve formě histogramu. Tento histogram by znázorňoval distribuci rozdílů časů mezi videem a simulátorem. Předpokládal jsem, že pokud budou časy odpovídat výsledné histogramy budou mít Gaussovo rozdělení. Jak se však při simulacích ukázalo Gaussovo rozdělení měly i simulace, kde byl nastavený typ agenta *RandomizedPotentialAwareAgent*. Tento agent se pohybuje náhodně a je zřejmé, že realitě neodpovídá. Toto je způsobeno tím, že při velkém počtu porovnávaných časů se vždy najde odpovídající čas, který není tolik vzdálený. Další problém je, že při tvorbě časových konfiguračních souborů se cílová trajektorie vybírá náhodně dle zvoleného poměru. Není však zaručeno pořadí stejné jako bylo natočeno na video. Tím pádem je velmi těžké najít odpovídající časové údaje pro video a pro simulátor. Ukázka výsledného diagramu je na obrázku 3.14. Na obrázku 3.15 zase vidíme časovou osu, na které body znázorňují čas, kdy lidé a agenti opustili mapu scénáře. Tento experiment se tedy pro moji konfiguraci nehodil. Kdyby se ovšem podařilo simulovanou lokaci natočit celou, mohli bychom získat trajektorie všech lidí a věděli bychom, kam který člověk jde.



Obrázek 3.14: Diagram vyjadřující počet agentů podle velikosti rozdílů naměřených a simulovaných časů



Obrázek 3.15: Body vyjadřující, kdy lidé (dolní body) a agenti (horní body) opustili mapu scénáře

### 3.2.3 Hodnocení průzkumem

Jako poslední měřítko jsem zvolil průzkum formou dotazníku (naleznete jej na příloženém CD). V dotazníku umístěném na internetu jsem participanty seznámil se simulovanou lokací a poté zhlédli video, na kterém byly zaznamenány čtyři simulace. Každá z těchto simulací měla nastavený jiný druh agenta. Úkolem poté bylo přidělit všem simulacím body od 1 do 10 podle toho, jak moc si dotyčný participant myslí, že simulace odpovídá skutečnosti. Čím

více bodů tím více simulace odpovídá skutečnosti. Jedná se pouze o subjektivní hodnocení, ale i tak bych ho považoval za platné. Výsledky dotazníku jsou v kapitole [4.2.3](#)



# Kapitola 4

## Výsledky

### 4.1 Scénář 1

#### 4.1.1 Trajektorie

Z naměřených vzdáleností v tabulkách 4.2 a 4.1 je vidět, že trajektorie videa mají dle předpokladů nejmenší průměrnou vzdálenost. Průměrná vzdálenost trajektorií při nastavení *RandomizedPotentialAwareAgent* je na druhou stranu, dle předpokladu, největší. Hodnoty pro agenty typu *AgentWithSocialForce* a *RandomizedPotentialAwareAgent* se pohybují mezi tím. Můžeme vidět, že první zmíněný typ má přibližně o 25 % přesnější trajektorie než druhý zmíněný.

	video 1	video 2	video 3	video 4	video 5
video 1	-	0,0350224	0,0195657	0,0211143	0,0320279
video 2	0,0350224	-	0,0411644	0,0417831	0,0325165
video 3	0,0195657	0,0411644	-	0,0207646	0,0335506
video 4	0,0211143	0,0417831	0,0207646	-	0,0355586
video 5	0,0320279	0,0325165	0,0335506	0,0355586	-

Tabulka 4.1: Všech pět trajektorií z videa jsem navzájem porovnal abych získal referenční vzdálenosti. Čím menší je číslo, tím více jsou si měřené trajektorie podobné.

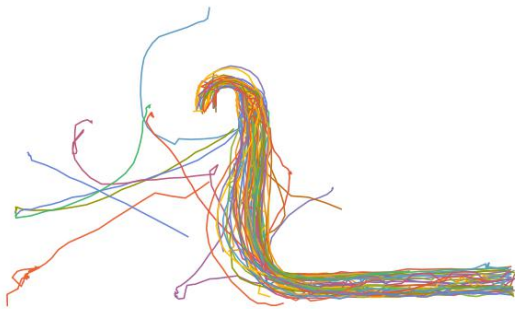
#### 4.1.2 Heat maps

Na obrázcích 4.4 a 4.5 vidíme heat mapy pro video i pro simulátor. Jejich rozdíl nalezneme na obrázku 4.6. Jak vidíme, rozdíl tvoří především okraje okolo krajních trajektorií. Dále jsou vidět výrazné bílé skvrny. Jelikož v jejich místě dochází ke křížení trajektorií lidí. Jak si můžeme všimnout v případě označené trajektorie na obrázku 4.3. Tato trajektorie se liší od ostatních trajektorií. Jedná se o jednu ze situací, které nelze dobře namodelovat. Jedinec měl stejnou entry/exit zónu jako všichni ostatní a většinu času také sledoval stejnou trajektorii, z nenadání ale vybočil a opět se vrátil zpět. Člověk kterému trajektorie patří, šel pravděpodobně k výtahu a poté si to rozmyslela šel po schodech. Kdybychom chtěli všechny

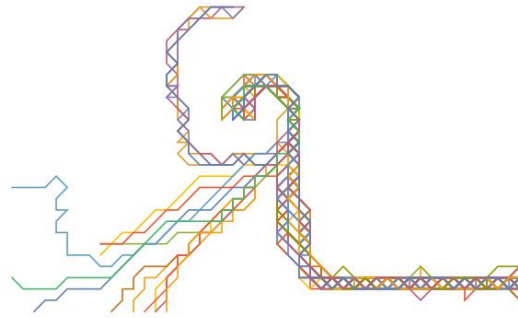
	social	potential	randomized
video 1	0,0450545	0,0563964	0,0933763
video 2	0,0528556	0,0648727	0,0857351
video 3	0,0396466	0,0508545	0,0919295
video 4	0,0350754	0,0438572	0,0947640
video 5	0,0384706	0,0492992	0,0887839
$\emptyset$	0,0422206	0,0530560	0,0909178

Tabulka 4.2: Trajektorii od každého druhu agenta jsem změřil se všemi trajektoriemi z videa. Zde jsou naměřené vzdálenosti. Čím menší je číslo, tím více jsou si měřené trajektorie podobné.

tyto neobvyklé situace nasimulovat tak, aby generované trajektorie byly stejně různorodé jako trajektorie z reálného světa, museli bychom simulátor velmi složitě nastavovat. Heat mapa ze simulátoru má tedy pravidelnější strukturu a vidíme, že je tvořena pouze trajektoriemi o několika tvarech.



Obrázek 4.1: Trajektorie video



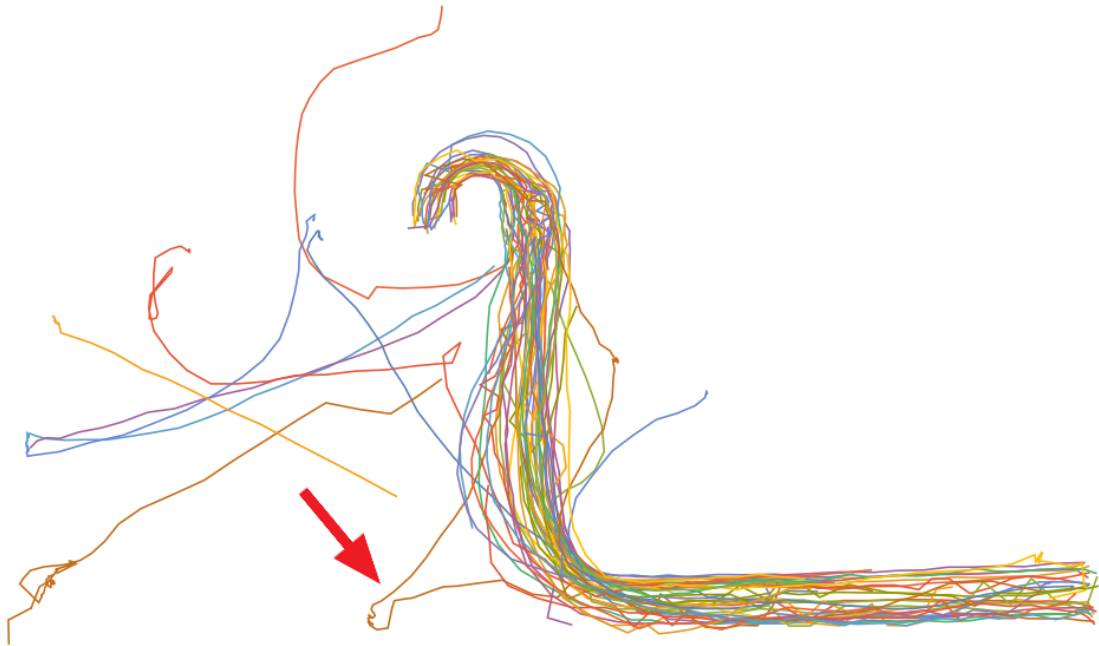
Obrázek 4.2: Trajektorie simulátor

## 4.2 Scénář 2

### 4.2.1 Cestovní časy

Rychlost chůze agentů byla nastavena vždy stejně a na obrázku 4.7 vidíme, že nejpřesnějších časů dosahovali agenti typu *PotentialAwareAgent*. Agentům typu *AgentWithSocialForce* trvalo projít danou trasu déle než jsou referenční časy naměřené v reálném světě. Pro názornost je zde uveden i typ *RandomizedPotentialAwareAgent*, který dosahuje absolutně nepřesných časů. Když jsem prováděl měření, zkoušel jsem zároveň měnit i rychlost nastavené chůze a odchylku od této rychlosti. Při nastavení typu *PotentialAwareAgent* se rychlost agentů měnila a zároveň bylo vidět, že agenti chodí odlišnými rychlostmi. Pro *AgentWithSocialForce* se však rychlost neměnila a agenti dosahovali stále stejných časů. Také všichni agenti chodili stejnou rychlostí.





Obrázek 4.3: Příklad neobvyklé trajektorie. Tato trajektorie vybočuje z průměrných trajektorií pro danou dvojici entry/exit zón.

#### 4.2.2 Časy opuštění scénáře

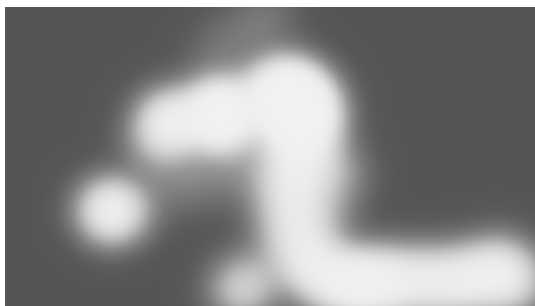
Při tomto měření bylo očekáváno, že diagramy naměřených časů budou mít tvar Gaussovy křivky pokud budou odpovídat časům naměřených v realitě. Problémy s tímto měřením jsou zmíněny v kapitole 3.2.2. Zde uvádím ukázky diagramů. Na obrázku 4.9 vidíme příklad, kde časy odpovídají. Obrázek 4.10 znázorňuje diagram pro nepřesné časy. Vidíme na něm, že

#### 4.2.3 Hodnocení průzkumem

Respondenti udělovali od jednoho do deseti bodů. Jeden bod znamená nejmenší věrohodnost a deset bodů znamená naprostá shoda s realitou. Výsledky jsou v tabulce 4.3. Všechny získané údaje jsou v na přiloženém CD ve složce *Data*.

CrowdPersonAgent	1,00
RandomizedPotentialAwareAgent	2,80
PotentialAwareAgent	8,15
AgentWithSocialForce	7,50

Tabulka 4.3: Průměrné hodnocení, které udělili participantů v provedeném průzkumu. Vidíme, že nejlepšího hodnocení dosáhl agent typu *PotentialAwareAgent*.



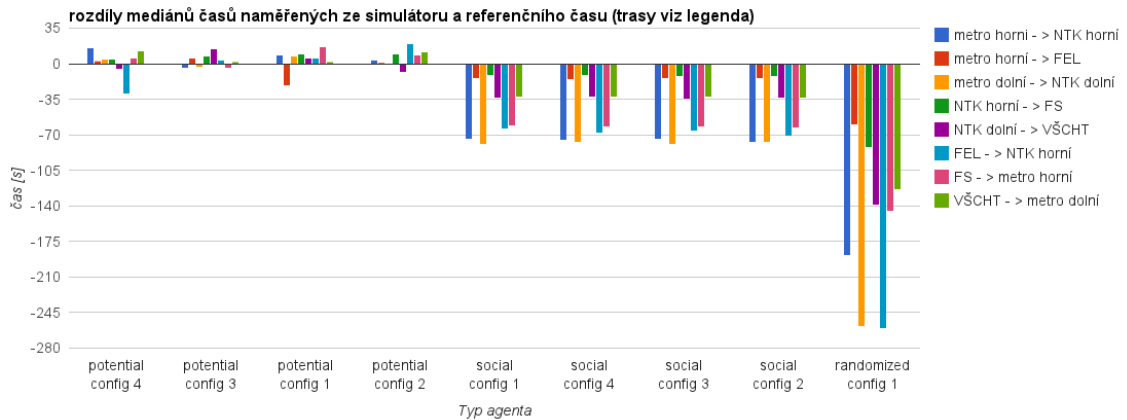
Obrázek 4.4: Heat map vygenerovaná trajektoriemi z videa. Čím světlejší místo, tím více lidí jím prošlo.



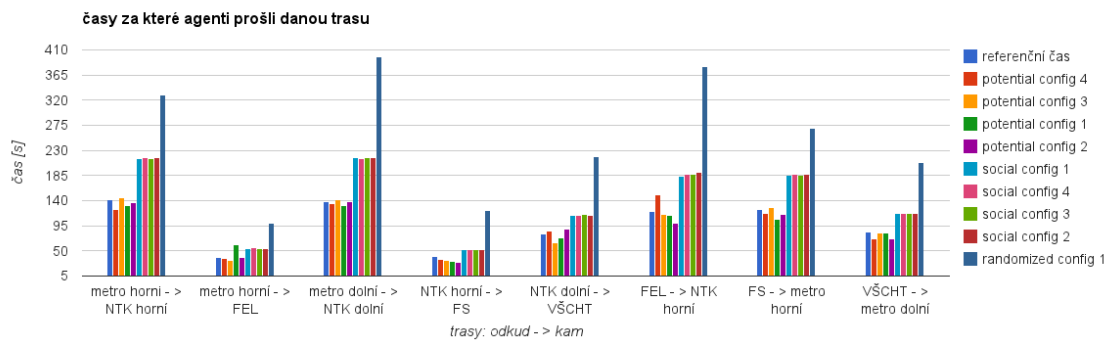
Obrázek 4.5: Heat map vygenerovaná trajektoriemi ze simulátoru. Čím světlejší místo, tím více lidí jím prošlo.



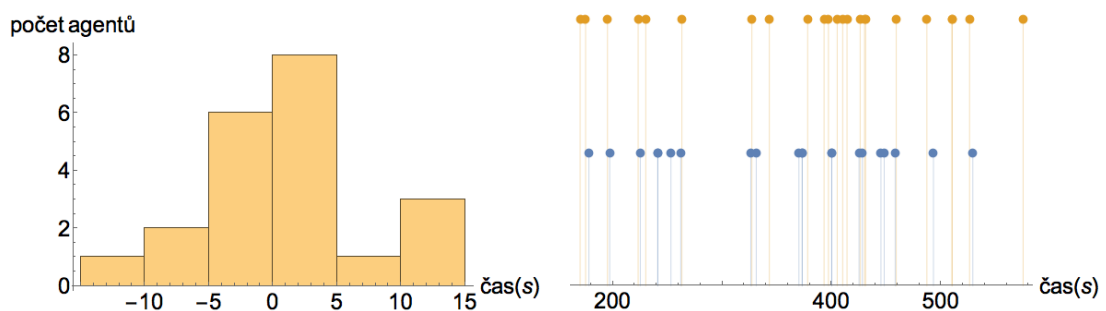
Obrázek 4.6: Obrázek zobrazuje, v čem se liší heat mapa generovaná trajektoriemi z videa a heat mapa generovaná simulátorem. Pokud by byly obě stejné, viděli bychom pouze černý obrázek. Bílá barva značí jejich rozdíl.



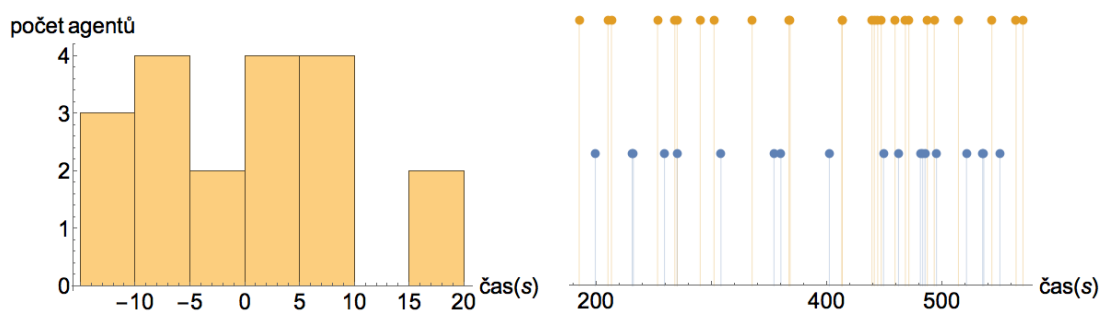
Obrázek 4.7: Rozdíly mediánů časů naměřených v simulátoru a referenčních časů. Čím blíže je hodnota nulové ose tím menší je rozdíl oproti referenčnímu času a tím pádem daná konfigurace více odpovídá skutečnosti.



Obrázek 4.8: Časy naměřené v simulátoru pro všechny konfigurační soubory a pro každou trasu. Pro porovnání je u každé trasy nejvíce vlevo zobrazen referenční čas.



Obrázek 4.9: Časy odpovídají, diagram je ve tvaru Gaussovy křivky. Na pravém obrázku jsou body vyjadřující, kdy lidé (dolní body) a agenti (horní body) opustili mapu scénáře.



Obrázek 4.10: Časy neodpovídají a časové rozdíly jsou větší. Na pravém obrázku jsou body vyjadřující, kdy lidé (dolní body) a agenti (horní body) opustili mapu scénáře.

# Kapitola 5

## Závěr

### 5.1 Shrnutí

V této práci jsem realizoval dva scénáře. V prvním scénáři jsem natočil video v prostorách Národní technické knihovny. Z analýzy videa jsem získal trajektorie lidí. Poté jsem změřil vzdálenosti mezi trajektoriemi z videa a ze simulátoru. Výsledná data ukazují, jak přesné trajektorie generují různé druhy agentů. Pomocí heat map jsem zobrazil rozdíly mezi všemi trajektoriemi z globálnějšího pohledu. Ve druhém scénáři jsem k problému přistoupil jiným způsobem. Zvolil jsem větší lokaci - část ulice Technická v Praze. Natáčel jsem všechny vstupy do této ulice. Mezi ně patří i vstupy do škol FEL, FS a VŠCHT. Videá mi umožnila získat data pro přesné nastavení simulací. V simulátoru jsem implementoval možnost přidat soubor s časy, ve kterých se mají vygenerovat agenti. Doplnil jsem také vlastnost umožňující zvolení typu povrchu v mapě simulace. Podařilo se mi změřit tato data: Porovnání časů, za které agenti a lidé prošli dané trasy. Dále jsem změřil rozdíly časů, ve kterých lidé odcházeli z lokace, a časů, kdy lokaci opustili agenti. Jako poslední měřítko jsem provedl výzkum pomocí dotazníku, ve kterém byl participanty zhodnocen každý druh agenta. Na základě měření vyšlo, že nejpřesnější trajektorie generuje typ *AgentWithSocialForce*. Nejmenší odchylky od měřených cestovních časů má typ *PotentialAwareAgent*. Tento typ získal také nejlepší hodnocení v dotazníku.

### 5.2 Možná rozšíření

Námětů možné budoucí práce je několik.

- V prvním scénáři je připraveno prostředí pro možnou budoucí práci. Postupem, jakým jsem v tomto scénáři postupoval, lze analyzovat jakékoliv video a můžeme tedy navrhnout scénář s jiným prostředím. Pro získání rozsáhlejších dat s delšími a členitějšími trajektoriemi by bylo dobré natočit prostor o větší ploše s vyšším počtem lidí. Následné porovnávání trajektorií by už bylo stejné, jako v mém scénáři.
- Všechna porovnávání jsem dělal po částech a různými způsoby. Bylo tedy časové náročné všechna měření provést a zaznamenat jejich výsledky. Další možností rozšíření by

mohla být implementace programu, který by zjednodušoval analýzu výstupního souboru ze simulátoru. Zobrazoval by například všechny trajektorie, jejich délku a čas, za který agent trajektorii prošel. Tento program by se mohl použít i pro jiné simulátory a zjednodušil by práci a orientaci v simulacích.

- Aby se mohly přesněji provádět simulace i v menších lokacích mohl by se v simulátoru přepracovat způsob pohybu agentů. Místo pohybu po buňkách (cells) by se pohybovali podle absolutních souřadnic. Vznikaly by tak přirozenější trajektorie bez skoků.

# Literatura

- [1] B. Banerjee and L. Kraemer. Validation of agent based crowd egress simulation. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 1 - Volume 1*, AAMAS '10, pages 1551–1552, Richland, SC, 2010. International Foundation for Autonomous Agents and Multiagent Systems.
- [2] F. Dignum. Agents for games and simulations ii. *Lecture Notes in Computer Science*, 6525, 2011.
- [3] N. Farenc, R. Boulic, and D. Thalmann. An informed environment dedicated to the simulation of virtual humans in urban context. In *Computer Graphics Forum*, volume 18, pages 309–318. Wiley Online Library, 1999.
- [4] R.-Y. Guo, H.-J. Huang, and S. Wong. Route choice in pedestrian evacuation under conditions of good and zero visibility: Experimental and simulation results. *Transportation Research Part B: Methodological*, 46(6):669 – 686, 2012.
- [5] S. Jain and C. R. McLean. Integrated simulation and gaming architecture for incident management training. In *Simulation Conference, 2005 Proceedings of the Winter*, pages 10–pp. IEEE, 2005.
- [6] Z. Jin and B. Bhanu. Optimizing crowd simulation based on real video data. pages 3186–3190, Sept 2013.
- [7] A. Lerner, Y. Chrysanthou, A. Shamir, and D. Cohen-Or. Data driven evaluation of crowds. In A. Egges, R. Geraerts, and M. Overmars, editors, *Motion in Games*, volume 5884 of *Lecture Notes in Computer Science*, pages 75–83. Springer Berlin Heidelberg, 2009.
- [8] N. Pelechano, C. Stocker, J. Allbeck, and N. Badler. Being a part of the crowd: Towards validating vr crowds using presence. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '08, pages 136–142, Richland, SC, 2008. International Foundation for Autonomous Agents and Multiagent Systems.
- [9] Wikipedia. Crowd simulation — wikipedia, the free encyclopedia, 2014. [Online; accessed 1-January-2015].
- [10] Z. Zhang, K. Huang, and T. Tan. Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 3, pages 1135–1138, 2006.





# Příloha A

## Obsah CD

V tabulce [A.1](#) najdeme popis všech složek a souborů na přiloženém CD.

<b>Název složky</b>	<b>Popis</b>
Text	složka s výslednou bakalářskou prací v PDF
Aplikace	složka s rozšířenou aplikací AgentCrowd a s aplikacemi vytvořenými pro analýzu
Data	složka obsahující výsledné tabulky s naměřenými hodnotami
Mathematica notebooks	složka se všemi Mathematica notebooks, které byly použity při analýze
Videa	všechna natočená videa

Tabulka A.1: Obsah CD