

ZÁPADOČESKÁ UNIVERZITA V PLZNI

FAKULTA APLIKOVANÝCH VĚD

KATEDRA GEOMATIKY

DIPLOMOVÁ PRÁCE

**Určení polohy mobilního telefonu v interiéru budovy
zvolenou technikou**

ZADÁNÍ DIPLOMOVÉ PRÁCE
(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Hana KUBÍČKOVÁ**
Osobní číslo: **A16N0158P**
Studijní program: **N3602 Geomatika**
Studijní obor: **Geomatika**
Název tématu: **Určení polohy mobilního telefonu v interiéru budovy zvolenou technikou**
Zadávací katedra: **Katedra geomatiky**

Z á s a d y p r o v y p r a c o v á n í :

1. Rešerše obecných přístupů k navigaci v interiérech a prozkoumání jejich potenciálu.
2. Přehled jednotlivých přístupů k navigaci v interiéru pomocí obrazového záznamu.
3. Návrh postupu pro určení polohy mobilního telefonu v interiéru budovy (včetně vymezení nutných podmínek).

Rozsah grafických prací: dle potřeby
Rozsah kvalifikační práce: cca 45 stran
Forma zpracování diplomové práce: tištěná/elektronická
Seznam odborné literatury:

- Koyuncu, H., & Yang, S. H. (2010). A survey of indoor positioning and object locating systems. IJCSNS International Journal of Computer Science and Network Security, 10(5), 121-128.
- Yihong Wu, Fulin Tang, Heping Li: Image Based Camera Localization: an Overview. Visual Computing for Industry. University of Chinese Academy of Sciences, 2018.
- Ozdenizci, B., Coskun, V., & Ok, K. (2015). NFC internal: An indoor navigation system. Sensors, 15(4), 7571-7595.

Vedoucí diplomové práce: Ing. Karel Jedlička, PhD.
Katedra geomatiky

Datum zadání diplomové práce: 31. října 2018
Termín odevzdání diplomové práce: 17. května 2019



Doc. Dr. Ing. Vlasta Radová
děkanka



Doc. Ing. Václav Čada, CSc.
vedoucí katedry

V Plzni dne 31. října 2018

Abstrakt

Předkládaná diplomová práce je zaměřena na určení polohy mobilního telefonu v interiéru budovy. V první části práce jsou popsány obecné přístupy pro polohování v interiéru budovy a prozkoumán jejich potenciál. Dále je práce podrobněji zaměřena na přehled přístupů k polohování v interiéru pomocí obrazového záznamu. Na základě získaných poznatků je v poslední části práce představeno navržené řešení pro určení polohy mobilního telefonu v interiéru budovy a to včetně vymezení nutných podmínek. Návrh řešení byl realizován za využití knihovny OpenCV.

Klíčová slova

vnitřní polohovací systém, polohování na základě obrazu, počítačové vidění, SIFT, detekce významných prvků, popis významných prvků, korespondence, kamera mobilního telefonu, Structure from motion, PnP problém, projekční matice, epipolární geometrie, OpenCV

Abstract

The presented thesis is focused on determining the position of a mobile phone in the interior of a building. The first part describes general approaches for positioning in the interior of the building and explores their potential. Furthermore, the work is focused in more detail on the overview of approaches to positioning in the interior using captured image. Based on the acquired knowledge, the last part of the thesis introduces the proposed solution for determining the position of a mobile phone in the interior of a building, including the definition of necessary conditions. The solution was implemented using the OpenCV library.

Key words

Indoor positioning system, image-based positioning system, computer vision, SIFT, feature detection, feature description, correspondence, cellphone camera, Structure from motion, PnP problem, projection matrix, epipolar geometry, OpenCV

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracovala samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této diplomové práce.

Dále prohlašuji, že veškerý software, použitý při řešení této diplomové práce, je legální.

.....

V Plzni dne 16.5.2019

Hana Kubičková

Poděkování

Na tomto místě bych ráda poděkovala svému vedoucímu diplomové práce Ing. Karlovi Jedličkovi, PhD. za odborné vedení, cenné informace, věcné připomínky a čas, které mi v průběhu zpracování práce věnoval. Dále děkuji Ing. Radku Fialovi, Ph.D. za poskytnutí mnoha cenných rad, konzultací a entusiasmů. Poděkování patří také Ing. Ivanu Gruberovi z katedry kybernetiky, který mi s ochotou poskytl cenné informace. Nakonec bych ráda poděkovala své rodině za podporu v průběhu celého studia.

Obsah

Seznam symbolů a zkratk	10
Seznam obrázků.....	12
Seznam tabulek.....	13
Seznam grafů	13
Seznam výpisů.....	13
Úvod	15
1 Rešerše přístupů k navigaci v interiérech.....	17
1.1 GNSS	18
1.2 Inerciální navigace	18
1.3 Optické polohování.....	20
1.4 Navigace v interiérech s posílením infrastruktury	20
1.4.1 Pseudosatelity	21
1.4.2 Technologie rádiových vln	21
1.4.3 Technologie infračerveného záření	24
1.4.4 Technologie ultrazvukových vln	25
1.5 Porovnání polohovacích technik.....	26
2 Teoretické základy určení polohy ze snímku a algoritmy počítačového vidění	28
2.1 Souřadnicové systémy	28
2.2 Geometrie jednoho pohledu.....	30
2.2.1 Matematický model kamery	31
2.2.2 Projekční matice	31
2.2.3 Odhad projekční matice.....	32
2.3 Geometrie dvou pohledů.....	33
2.3.1 Fundamentální matice.....	34
2.3.2 Odhad fundamentální matice.....	35
2.3.3 Esenciální matice	36
2.3.4 Odhad projekčních matic z esenciální matice	37
2.4 Algoritmy související s vyhledáváním snímkových korespondencí.....	40
2.4.1 Detekční algoritmy	40
2.4.2 Určení korespondencí mezi snímky	44
2.4.3 Filtrace nepravých korespondencí	45
3 Přehled přístupů k polohování pomocí obrazového záznamu.....	47

3.1	Polohování ve známém prostředí.....	47
3.1.1	Lineární PnP problém ($n \geq 6$).....	48
3.1.2	EPnP problém.....	48
3.1.3	Nelineární PnP problém ($3 \leq n < 6$).....	49
3.2	Polohování v neznámém prostředí.....	49
3.2.1	SLAM.....	49
3.2.2	SFM.....	50
3.3	Rešerše publikací využívající pro určení polohy v interiéru kameru mobilního telefonu.....	51
4	Navržené řešení pro určení polohy v interiéru prostřednictvím jednoho snímku.....	55
4.1	Příprava databáze.....	57
4.2	Použité programové vybavení.....	59
4.3	Navržené řešení.....	60
4.3.1	Načtení dat.....	60
4.3.2	Výběr korespondujícího snímku z databáze.....	60
4.3.3	Odhad esenciální matice.....	63
4.3.4	Odhad projekční matice C1.....	66
4.3.5	Odhad měřítka.....	67
4.3.6	Odhad projekční matice C2.....	72
4.4	Hodnocení výsledků navrženého řešení.....	73
	Diskuse.....	82
	Závěr.....	84
	Zdroje.....	86
	Přílohy.....	i
	Zdrojový kód.....	i
	Tabulky.....	viii
	Obsah přiloženého CD.....	x

Seznam symbolů a zkratek

2D	2 – dimenzionální
3D	3 – dimenzionální
AoA	Angle of Arrival
BRIEF	Binary Robust Independent Elementary Features
BSD	Berkeley Software Distribution
CPU	Central Processing Unit
EPnP	Efficient Perspective n Point
GLONASS	Globalnaja navigacionnaja sputnikovaja sistéma
GNSS	Global Navigation Satellite System
GPU	Graphics Processing Unit
IEEE	Institute of Electrical and Electronics Engineers
IMU	Inertial Measurement Unit
JPEG	Joint Photographic Experts Group
LCD	Liquid Crystal Display
LED	Light – Emitting Diode
MAC	Media Access Control
MAPSAC	Maximum A Posterior Sample Consensus
MEMS	Micro Electro Mechanical Systems
MLESAC	Maximum Likelihood Estimate Sample Consensus
NAVSTAR GPS	Navigation Signal Timing and Ranging Global Positioning System
OCR	Optical Character Recognition
ORB	Oriented Fast and Rotated Brief
PCA – SIFT	Principal Component Analysis - Scale Invariant Feature Transform
PnP	Perspective n Point
PRN	Pseudorandom Noise
QR	Quick Response
RAM	Random Access Memory
RANSAC	Random Sample Consensus
RFID	Radio Frequency Identification
RGB	Red Green Blue
RGB – D	Red Green Blue - Depth
RSS	Received Signal Strength
SIFT	Scale Invariant Feature Transform
SFM	Structure From Motion
SLAM	Simultaneous Localization and Mapping
SSID	Service Set Identifier
SURF	Speed Up Robust Feature
SVD	Singular Value Decomposition
TDoA	Time Difference of Arrival

ToA	Time of Arrival
UWB	Ultra Wide Band
VÚGTK	Výzkumný ústav geodetický topografický a kartografický
XML	eXtensible Markup Language

Seznam obrázků

Obr. 1 Rozdělení vnitřních polohovacích systémů.....	17
Obr. 2 Lokalizace na základě buněk.....	23
Obr. 3 Porovnání signálu Wi-Fi/Bluetooth a UWB	24
Obr. 4 Šíření signálu RFID a ultrazvukového signálu.....	25
Obr. 5 Zvolená technika pro určení polohy uživatele v interiéru budovy.....	27
Obr. 6 Znázornění souřadnicových systémů	29
Obr. 7 Model dírkové kamery	30
Obr. 8 Geometrie korespondujících bodů.....	34
Obr. 9 Čtyři možná řešení vycházející z SVD rozkladu esenciální matice.....	39
Obr. 10 Princip vyhledávání scale-faktoru	41
Obr. 11 Princip SIFT deskriptoru	42
Obr. 12 Pincip FAST detektoru.....	43
Obr. 13 Nalezené korespondence při změně barevné intenzity snímku.....	44
Obr. 14 Rozdělení metod výpočtu polohy ze snímku	47
Obr. 15 Znázornění významných prvků scény.....	52
Obr. 16 Systém OCRAPOSE	53
Obr. 17 Schéma navrženého řešení	56
Obr. 18 Půdorys interiéru budovy a zobrazení počátku objektového souřadnicového systému	57
Obr. 19 Struktura vytvořené databáze	58
Obr. 20 Struktura zápisu 2D – 3D korespondence vlíčovacího bodu v XML dokumentu..	58
Obr. 21 Rozložení vlíčovacích bodů ve snímku uloženém v databázi.....	59
Obr. 22 Výběr nejvíce odpovídajícího databázového snímku vůči vstupnímu snímku	61
Obr. 23 Ukázka nalezených korespondencí mezi databázovým snímek (vlevo) a vstupním snímek (vpravo).....	62
Obr. 24 Znázornění epipolárních přímek vedených významnými body ve snímku.....	64
Obr. 25 Znázornění snímkových korespondencí, ze kterých byla vypočtena transformační matice homografie	69
Obr. 26 Transformované snímkové souřadnice obrazů vlíčovacích bodů ve vstupním snímku	70
Obr. 27 Testované scény - scéna č. 1 (vlevo), scéna č. 2 (vpravo)	74
Obr. 28 Ukázka snímku IMG_4955	78

Seznam tabulek

Tab. 1 Porovnání polohovacích technik	26
Tab. 2 Výsledky kalibrace v programu Agisoft Lens.....	65
Tab. 3 Výsledky testu určení polohy na základě navrženého řešení	74
Tab. 4 Vliv počtu vyhledaných snímkových korespondencí na střední souřadnicovou chybu	78
Tab. 5 Výsledky testování snímku IMG_4955.....	79
Tab. 6 Výsledky testování vstupních snímků oproti sadě 5 databázových snímků	80
Tab. 7 Podrobné výsledky testu určení polohy na základě navrženého řešení – sada 14 snímků uložených v databázi.....	viii
Tab. 8 Podrobné výsledky testování vstupních snímků oproti sadě 5 databázových snímků	ix
Tab. 9 Podrobné výsledky testování snímku IMG_4955	ix

Seznam grafů

Graf 1 Výsledky testu navrženého řešení	76
Graf 2: Vliv délky základny na střední souřadnicovou chybu určení polohy	77
Graf 3 Vliv počtu vyhledaných korespondencí na střední souřadnicovou chybu určení polohy	79
Graf 4 Procentuální vyjádření získaných přesností určení polohy pro dvě sady s rozdílným počtem snímků.....	81

Seznam výpisů

Výpis 1 Zjištění snímku s maximálním počtem shod	63
Výpis 2 Odhad esenciální matice z fundamentální matice.....	65
Výpis 3 Ukázka funkce recoverPose()	65
Výpis 4 Převod souřadnic z XML souboru do datové struktury pole	66
Výpis 5 Odhad projekční matice	67
Výpis 6 Výpočet projekčního centra kamery	67
Výpis 7 Převod vektoru translace do objektového souřadnicového systému.....	68

Výpis 8 Transformace snímkových souřadnic obrazů vlíčovacích bodů snímku z databáze do vstupního snímku.....	71
Výpis 9 Projekční matice kamer sestavené pro účel triangulace snímkových souřadnic obrazů vlíčovacích bodů.....	71

Úvod

V dnešní době umožňuje funkci určování polohy mnoho moderních elektronických zařízení, zejména pak to, které jsme zvyklí používat každý den - mobilní telefon. S rychle rostoucím technologickým vývojem mobilních telefonů a oblasti polohových služeb se samozřejmostí staly mobilní telefony s lokalizací zajištěnou prostřednictvím globálního navigačního systému. Postupem času začala stoupat potřeba lokalizovat tato zařízení také uvnitř budov. Hlavním cílem bylo usnadnit činnosti, jako jsou umístění pacientů do nemocnice, hledání pracovníků ve velkých kancelářských budovách nebo rychlá orientace v rozsáhlých nákupních centrech. Přijímání GNSS signálu je uvnitř budov velmi problematické, protože není umožněna přímá viditelnost GNSS signálu mezi satelity a přijímači uvnitř budovy. Přímá viditelnost je nutná pro zpracování signálu, a tak není možné využít tuto techniku samu o sobě pro účely polohování v interiérech. Důležitým požadavkem pro navigaci v interiéru je navíc vyšší přesnost, než jakou vyžaduje použití v exteriéru. Pokud by chyby v poloze přesahovaly několik metrů, uživatel by nebyl schopný rozeznat například jednotlivá podlaží nebo místnosti mezi sebou a služba by tak poskytovala informace o místech, která jsou od skutečné polohy vzdálená. Z těchto důvodů začala být problematika týkající se polohování v interiéru budov řešena pomocí jiných technik - nejčastěji jejich kombinací, a společně s nimi začaly být vyvíjeny nové algoritmy.

Vzhledem k tomu, že v současné době existuje mnoho přístupů, jakými lze určit polohu v interiéru budovy, je jedním z cílů této diplomové práce představit, kategorizovat a prokoumat potenciál stávajících technik polohování v interiéru budovy. Dalším cílem práce je sestavení přehledu jednotlivých přístupů k polohování v interiéru pomocí obrazového záznamu. Posledním bodem zadání této diplomové práce je návrh řešení pro určení polohy mobilního telefonu v interiéru budovy na základě získaných poznatků.

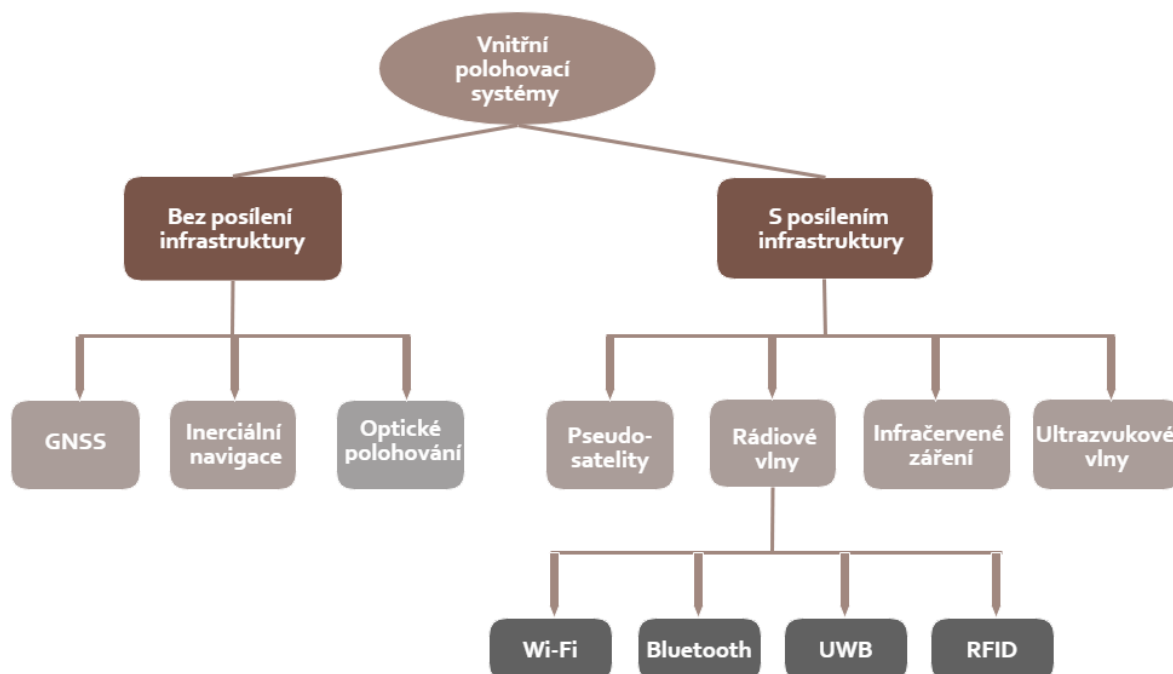
Text je rozdělen na 3 hlavní části. V první kapitole jsou kategorizovány, popsány a porovnány stávající techniky určení polohy v interiéru budovy. Druhá kapitola je věnována technice polohování pomocí obrazového záznamu, ve které jsou představeny principy výpočtu polohy z obrazového záznamu a algoritmy počítačového vidění, na jejichž základu tato technika stojí. Následuje kapitola, která zakončuje teoretickou část práce rešerší publikací a článků, které se zabývají určením polohy uživatele prostřednictvím kamery mobilního telefonu. Praktická část práce je uvedena čtvrtou kapitolou, která dává prostor pro návrh vlastního řešení určení polohy pomocí obrazového záznamu pořízeného kamerou

mobilního telefonu a hodnocení výsledků navrženého řešení. Praktická část práce je zakončena diskusí, ve které jsou shrnuta omezení navrženého řešení.

1 Rešerše přístupů k navigaci v interiérech

Navigace obecně je dle Ruppel et al. (2009) proces sledování a ovládání pohybu objektu z původního místa do cíle. Na cestě z místa A do cílového místa B poskytují navigační systémy čtení a aktualizaci polohy a vedou uživatele pomocí srozumitelných vizuálních, slyšitelných nebo hmatatelných prostředků. Pokud dojde k jakémukoliv vybočení mimo trasu, je vyžadováno přesměrování nebo oprava stávající polohy. Znalost polohy nazývaná jako polohování, je tak nezbytnou podmínkou přesné navigace.

Stávající techniky využívané pro polohování v interiéru budovy mohou být děleny podle jejich přesnosti a nároků na infrastrukturu. První skupina systémů je založena na stávající infrastruktuře, kam řadíme GNSS, inerciální senzory a optické polohování. Přesnost těchto technik je sice limitována, ale v praxi bývají tyto techniky posilovány jinou infrastrukturou, jako jsou pseudosatelity, Wi-Fi, Bluetooth nebo RFID, které řadíme ke druhé skupině systémů, kterou obecně nazýváme navigace s posílenou infrastrukturou (viz Obr. 1). V následujících kapitolách budou popsány jednotlivé techniky polohování v interiéru, a to od vysvětlení jejich principů až po zhodnocení jejich přesnosti.



Obr. 1 Rozdělení vnitřních polohovacích systémů

1.1 GNSS

Globální navigační družicový systém (Global Navigation Satellite System) je souhrnný termín užívaný k obecnému označení globálních družicových systémů, jimiž jsou zejména NAVSTAR GPS (Navigation Signal Timing and Ranging Global Positioning System), GLONASS, Galileo nebo BeiDou. (Slovník VÚGTK, 2018)

Na problematiku polohování v interiéru pomocí GNSS je nahlíženo v mnoha člancích z různých úhlů pohledu. Jsou to nejen otázky vlivu použitého stavebního materiálu na propustnost GNSS signálu (například článek autorů Yi, Li & Gu, 2012), ale především snahy zpřesnit získanou polohu. Zpřesňování polohy je realizováno buď využitím GNSS přijímačů s vysokou citlivostí nebo doplněním této technologie jiným způsobem navigace.

O využití přijímačů s vysokou citlivostí pojednává například článek autorů Schon & Bielenberg (2008). V tomto příspěvku je porovnáván výkon GNSS přijímače s vysokou citlivostí a geodetického přijímače při jejich použití ve vnitřním testovacím prostředí. Testování ukázalo, že pro geodetické přijímače je celková dostupnost signálu ve vnitřním prostředí velmi slabá (okolo 7%), oproti tomu přijímače s vysokou citlivostí byly schopny zachytit 73% z teoreticky možného počtu přijatých signálů. V tomto případě jde o vysokou toleranci šumu v signálu, který ovšem zvyšuje riziko chyby v určení polohy. Na základě měřených dat byla vypočtena poloha pro každou epochu a použitím sofistikovaných analytických postupů byla získána poloha objektu s přibližně 2m chybou v každé souřadnici. (Schon & Bielenberg, 2008) Co se týče zpřesňování polohy získané pomocí GNSS doplněním dalších technik polohování, jedná se především o implementaci inerciální navigace (viz kapitola 1.2) nebo využití pseudosatelitů a bezdrátových systémů jako jsou Wi-Fi, Bluetooth, RFID (radiofrekvenční identifikace) nebo UWB (Ultra-Wide Band). Podrobněji budou tyto způsoby polohování popsány v kapitole, která je soustředěna na navigování v interiérech s posílenou infrastrukturou (viz kapitola 1.4).

1.2 Inerciální navigace

Inerciální navigace je další navigační technikou, jejíž infrastrukturu není potřeba dále posilovat. Systém inerciální navigace je tvořen pohybovými senzory (akcelerometry), snímači otáčení (gyroskopy) a případně magnetickými senzory (kompasy), pomocí nichž dochází ke kontinuálnímu výpočtu aktuální relativní polohy, orientace a rychlosti pohybujícího se objektu vůči známému výchozímu bodu. Jednotky inerciálních měření

(IMU) se typicky skládají ze 3 navzájem kolmých akcelerometrů, 3 navzájem kolmých gyroskopů a počítače, který data zpracovává. S technologickým rozvojem a nástupem chytrých mobilních telefonů začal být kladen důraz na miniaturizaci senzorů inerciální navigace. Toho bylo dosaženo implementací nízkonákladových MEMS (Micro Electro Mechanical Systems) senzorů, jejichž použití skýtá své výhody i nevýhody. Nízká velikost, váha, spotřeba energie a cena těchto senzorů je vykoupena nižší přesností než jakou poskytují optické gyroskopy. (Woodman, 2007; Weston & Titterton, 2000)

Vzhledem k tomu, že je výpočet relativní polohy objektu vztažen ke známému počátečnímu bodu, setkáváme se u inerciální navigace se dvěma zásadními nedostatky. Prvním z nich je skutečnost, že přesnost polohy závisí na počáteční poloze. Malé chyby při měření zrychlení a úhlové rychlosti se hromadí do větších chyb v rychlosti a přispívají tak větším odchylkám v poloze, s čímž souvisí druhý problém. Tedy, že chyba polohy roste úměrně s časem od opuštění výchozího bodu, protože je nová pozice odhadována z předchozí odhadnuté polohy, naměřeného zrychlení a úhlové rychlosti. Z těchto důvodů bývá získaná poloha pravidelně opravována vstupem jiné technologie polohování. (Woodman, 2007)

Inerciální navigace se obvykle využívá k doplnění dalších navigačních systémů, čímž je dosaženo vyšší míry přesnosti než použití jakéhokoliv samostatného systému. Často bývá doplňkem GNSS systému (GNSS-IMU) nebo je kombinována s bezdrátovými systémy. Práce Niedermeier et al. (2010) řeší implementaci GNSS s inerciálním navigačním systémem, konkrétně spojením přijímačů s vysokou citlivostí a MEMS IMU. Prostřednictvím tohoto výzkumu byla demonstrována schopnost ve sledovacím režimu souvisle sjednotit přijímané satelitní signály do 2 s pro dynamický pohyb objektu a do 10 s při statickém stavu objektu, což umožňuje stanovit polohu po většinu času pohybování se interiérem budovy.

V souvislosti s používáním nízkonákladových inerciálních senzorů byla dále publikována práce autorů Evennou & Marx (2006). Ke korekci driftu, který oslabuje výsledky vypočtené polohy inerciální navigace, byl využit bezdrátový systém Wi-Fi. Takto navržená architektura navigačního systému byla nakonec porovnávána se samostatně použitým Wi-Fi a inerciálním polohovacím systémem. Výzkum ukázal, že spojením těchto dvou technologií můžeme v určení polohy dosáhnout přesnosti, která činí až 1 m.

1.3 Optické polohování

Přístupy využívající k určení polohy obrazovou informaci řadíme do oblasti počítačového vidění, jehož snahou je co nejdokonaleji napodobit reálné vidění člověka počítačem. Počítačové vidění je úzce spjato s fotogrammetrií, neboť střetem zájmu obou oblastí je zkoumání teorie a aplikace centrální projekce kamery. Jejich společným cílem je jinými slovy získání prostorových informací ze snímku nebo videozáznamu pořízeného kamerou. S tím dále souvisí kalibrace kamery, 3D projekce nebo tvorba 3D modelů. (Hartley & Mundy, 1993) Pro účely optického polohování lze využít různé typy kamer, jako jsou kamery mobilního telefonu, všesměrové kamery nebo trojrozměrné kamery. Jejich výkon se však liší v závislosti na množství informací, které lze extrahovat ze snímku. Optické polohovací systémy mohou být rozděleny do dvou kategorií. První z nich je systém ego-motion, které pro odhad polohy kamery využívají pohyb kamery vzhledem k statické scéně, a statických systémů, které pro odhad polohy lokalizují pohyblivé objekty v obrazech. (Alarifi et al., 2016)

Základním úkolem této techniky je určení polohy a rotace ve 3D z 2D snímků. Typicky je pořízeno několik snímků z jedné pozice kamery nebo několik snímků z více pozic jedné kamery. Informace o hloubce monokulárních snímků lze získat využitím pohybu kamery. V tomto přístupu, známém jako syntetické stereo vidění, je scéna pozorována postupně z různých míst stejným fotoaparátém a hloubku obrazu lze odhadnout způsobem podobným přístupu stereo vidění. Nicméně, základna mezi sekvenčními obrazy musí být určena doplňkovou technikou, protože nemůže být určena ze samotných snímků. Pokud využíváme stereo kamery se známou základnou, lze měřítko stanovit ze stereoskopických snímků. Nevýhodou je, že výkon stereo-kamerového systému je ovlivněn délkou základny. Příliš krátká základna způsobí, že geometrie není příznivá pro protínání vpřed, a proto není použitelná pro malá ruční zařízení. Alternativou tedy může být přímé měření vzdálenosti pomocí laserových snímačů nebo kamer s širokoúhlým záběrem. (Mautz & Tilch, 2011)

1.4 Navigace v interiérech s posílením infrastruktury

V této kapitole se budeme zabývat navigacemi s posílenou infrastrukturou, což je takový způsob navigace, kdy je pro určení polohy uživatele použito hardwarové vybavení v podobě vysílačů signálu. Existuje mnoho technologií, kterými je možné posílit stávající infrastrukturu, rozdíl mezi nimi je především v jejich architektuře, charakteru vysílaného

signálu, přesnosti a spolehlivosti. Nejčastěji jsou tyto technologie navigování rozděleny do 3 hlavních kategorií. Jsou jimi systémy založené na infračerveném záření, systémy založené na rádiových vlnách a systémy založené na ultrazvukových vlnách. V následujících podkapitolách budou stručně vysvětleny nejčastěji používané techniky posílení infrastruktury, a to Wi-Fi, Bluetooth, radiofrekvenční identifikace (RFID), ultrazvuk, infra-záření a UWB systémy. (Gu et al., 2009)

1.4.1 Pseudosatelity

Jedním z řešení, jak získat polohu v interiéru budovy, je využití tzv. pseudosatelitů (též pseudolitů). Pseudosatelity jsou pozemní rádiové vysílače, které byly vyvinuty jako pozemní alternativa GNSS s rozsáhlou oblastí pokrytí (až 50 km). Takové doplnění družicových navigačních systémů umožňuje rozšířit oblast pokrytí signálem i do vnitřního prostředí umístěním několika pseudosatelitů do rohů budovy, které simulují satelitní konstelaci. Každý pseudosatelit generuje a vysílá signály podobné GNSS signálu. Generované signály a GNSS signály mají sice stejnou strukturu signálu, ale liší se nosnou frekvencí, PRN (Pseudorandom Noise) kódem nebo navigačními údaji. (Rapinski et al., 2012) Aby bylo možné přijímat pseudosatelitní signál, je potřeba modifikovat běžné GNSS přijímače. Jak se vyhnout modifikaci uživatelského terminálu jen řešeno v článku autorů Xu et al. (2015). Jeho hlavní myšlenkou je přijímání skutečného GPS signálu, jeho opakování a předání vnitřním vysílacím anténám. Přenášené signály mohou být zpracovány generálním přijímačem, a proto není vyžadována modifikace hardwaru na přijímači.

1.4.2 Technologie rádiových vln

Technologie rádiových vln je v oblasti polohování v interiérech výhodná zejména proto, že rádiové vlny mohou snadno pronikat překážkami v podobě zdí, nábytku nebo lidí. Z tohoto důvodu mají systémy založené na rádiových vlnách větší oblast pokrytí a vyžadují menší hardwarové vybavení než ostatní systémy. Tuto technologii lze rozdělit na technologii s využitím úzkého pásma, kam patří Wi-Fi, Bluetooth a RFID a širokopásmovou, kam řadíme UWB systémy. (Xiao et al., 2011)

Wi-Fi

Wi-Fi (nebo také Wi-fi, WiFi, Wifi, wi-fi, wifi) je v informatice označení několika standardů IEEE 802.11 popisujících bezdrátovou komunikaci v počítačových sítích (též *Wireless LAN*,

WLAN). Samotný název Wi-Fi vytvořila *Wireless Ethernet Compatibility Alliance*. Tato technologie využívá tzv. "bezlicenčního frekvenčního pásma". (Bianchi, 2000)

System určování polohy pomocí Wi-Fi je založen na měření intenzity přijatého signálu a na tzv. metodě "otisku prstů". Metoda otisku prstů porovnává signál měřených Wi-Fi v neznámých pozicích s předem naměřenými signály Wi-Fi ve známých pozicích, které jsou uloženy do referenční databáze. Typickými parametry, které jsou užitečné pro lokalizaci Wi-Fi hotspotu nebo bezdrátového přístupového bodu jsou SSID a MAC adresa přístupového bodu. Přesnost této metody je závislá na počtu zaznamenaných poloh přístupových bodů do databáze. Kromě poloh přístupových bodů je potřeba vložit do databáze také propagační parametry, jimiž jsou exponent cesty, odchylka šumu, a rádiová mapa, ve které je pro každý přístupový bod zaznamenána intenzita signálů okolních přístupových bodů. (Eisa et al., 2013) Postupy spojené s vytvořením takové databáze jsou obvykle velmi časově náročné a pracné. Takto vytvořená databáze navíc není dynamická, což při lokalizaci uživatele v novém prostředí znamená, že je nepoužitelná. (Zhuang et al., 2015)

Současné vědecké práce se tak zabývají způsoby, jak tento problém s vytvářením databáze odstranit. Zhuang et al. (2015) ve své práci zavádí prototypový systém založený na crowdsourcingu, který automaticky generuje databázi za využití načtených dat během běžného používání mobilního telefonu. Pro odhad polohy přístupových bodů Wi-Fi a parametrů propagace je použit adjustment algoritmus. Dosavadní experimenty ukázaly, že průměrná odchylka v určení polohy uživatele činí několik metrů. (Gu et al., 2009)

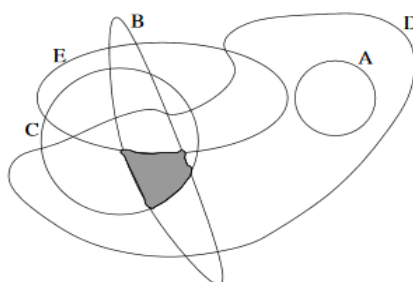
Bluetooth

Technologie Bluetooth je v informatice definována standardem IEEE 802.15.1 a byla vyvinuta pro bezdrátovou komunikaci mezi dvěma a více elektronickými zařízeními.

Lokalizace zařízení pomocí Bluetooth je další technologií, která je stejně jako Wi-Fi technologie závislá na měření intenzity přijatého signálu, které vysílají Bluetooth majáky. Nejprve je odvozena vzdálenost majáku od zařízení a poté je vypočtena poloha na základě znalosti principu trilaterace (nebo multilaterace). Vzhledem k tomu, že je ve vnitřním prostředí šíření signálu ztíženo různorodostí stavebních materiálů, nábytkem a dalším vybavením a rozsah šířeného signálu je pro Bluetooth typicky malý, nelze garantovat, že pokles signálu musí nutně znamenat prodloužení vzdálenosti. Technologie Bluetooth je

navíc charakteristická drahými přijímači a přesnost určení polohy závisí na množství použitých přijímačů. (Ozdenizci et al., 2015)

Začaly být tedy řešeny jiné způsoby výpočtu polohy zařízení, při kterých není nutné spoléhat na intenzitu přijatého signálu. Jedním z řešení tohoto problému bylo vytvoření referenční databáze úrovní signálu a zhotovení rádiové mapy stejně jako u Wi-fi. Dalším způsobem se pak zabýval Chawathe (2009), který tento problém ve své práci řeší využitím jen takových majáků, u kterých je zaznamenána přímá viditelnost a svůj přístup nazývá lokalizace na základě buněk (viz Obr. 2).

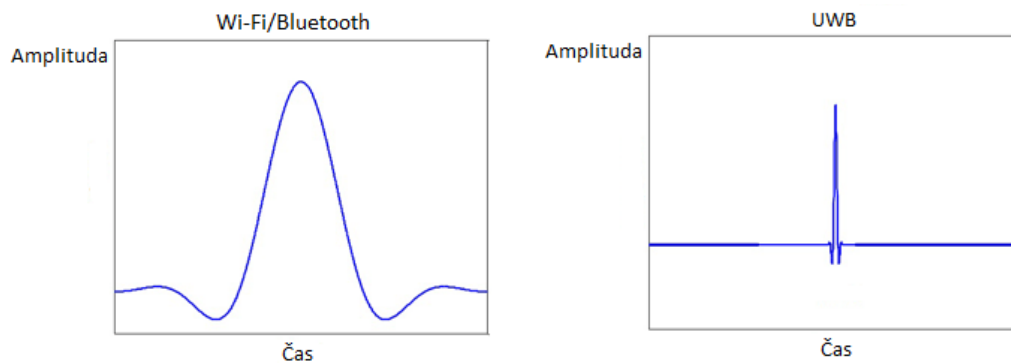


Obr. 2 Lokalizace na základě buněk: Mobilní zařízení, které je v dosahu majáků B, C a D a nikoliv v dosahu A a E, musí být lokalizováno ve vyznačené oblasti (převzato z Chawathe, 2009)

Ultra-wideband systémy

Vnitřní polohovací systémy založené na technologii Ultra-wideband (UWB) využívají velmi krátkých širokopásmových signálů. Výhodou je, že krátké širokopásmové signály umožňují snadné měření času mezi začátkem a koncem přijímání signálu. Oproti měření intenzity přijímaného signálu, které je využíváno u Wi-Fi a Bluetooth, může být na základě těchto impulsů změřena vzdálenost mezi vysílačem a přijímačem přesně. Další výhodou oproti dvěma zmíněným technikám je skutečnost, že širokopásmové signály velmi efektivně odolávají rušení v podobě bílého šumu nebo rušení způsobeného šířením odražených signálů. Technika UWB je díky tomu charakteristická vysokou přesností měření, která činí 5-10 cm. (Connell, 2015)

I UWB systémy však mají svá omezení. Polohovací systém založený na technologii UWB trpí nízkým výkonem a častými výpadky signálu. (Ozdenizci et al., 2015) Vysoká přesnost navíc vyvažuje vysokou cenu tohoto systému. (Connell, 2015)



Obr. 3 Porovnání signálu Wi-Fi/Bluetooth a UWB (převzato a upraveno z Connell, 2015)

Radio frekvenční identifikace

Radiofrekvenční identifikace (RFID) využívá k automatické identifikaci a sledování značek připojených k objektům elektromagnetické pole. Hlavními součástmi RFID systémů jsou čtečka spojená s anténou, která dotazuje více značek najednou. Princip této metody je následující: značky se aktivují dotazováním a poté zasílají odpověď prostřednictvím jedinečného identifikačního řetězce zpět do čtečky. (Bekkali et al., 2007) Technologie RFID je rozdělena do dvou kategorií, a to pasivní RFID a aktivní RFID. U pasivní RFID je sledovaná značka přijímač signálu. Je charakteristická malou velikostí a nízkou cenou značek, ale oblast pokrytí signálem je také malá. Oproti tomu u aktivní RFID je oblast pokrytí signálem vyšší, a to díky využití vysílačů představující značky. Cena těchto značek je však mnohem vyšší než u pasivního přístupu. (Gu et al., 2009)

RFID byla ve spojitosti s navigováním v interiéru budovy studována z mnoha úhlů pohledu. Zkoumány byly především přístupy k umístění značek, typům použitých značek a druhům snímaných systémů. Některé z přístupů využívají informace o přijaté síle signálu, zatímco jiné používají RFID informace jako optimalizaci dat získaných z akcelerometru. (Saab & Nakad, 2011)

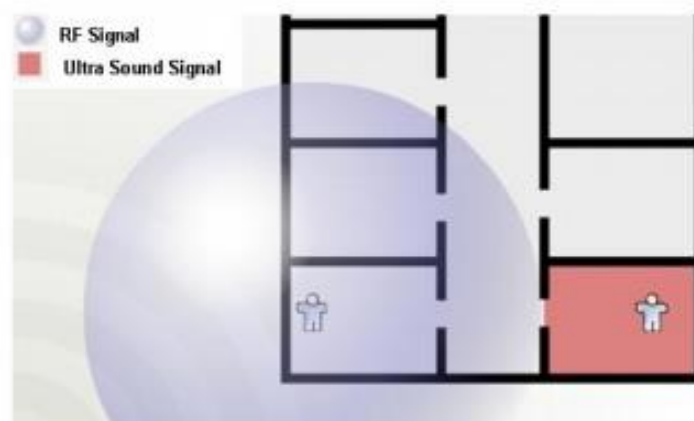
1.4.3 Technologie infračerveného záření

Infračervené záření je elektromagnetické vlnění s vlnovou délkou v rozmezí 760 nm a 1 mm. Polohovací systémy založené na infračerveném záření se řadí mezi nejpoužívanější systémy k určení polohy využívající bezdrátové technologie připojení. Vysílání infračerveného záření zajišťují LED diody. Většina polohovacích systémů založených na infračerveném záření využívají přímou viditelnost mezi vysílačem (diodou) a přijímačem bez rušení od silných světelných zdrojů. Co se týče výhod, které přináší tento

způsob polohování, patří sem zejména malé provedení vysílačů a velice přesné určení polohy – několik milimetrů. Mezi nevýhody pak řadíme určitá omezení v určení polohy, která zapříčiňuje sluneční světlo a interference od fluorescenčního světla. Tento problém lze vyřešit použitím optických a elektronických filtrů, které odstraní rušení světelných zdrojů, což ale vede k nárůstu ceny tohoto způsobu polohování. (Gu et al., 2009) Vedle toho je tato technika nákladná na hardwarové vybavení a náročná na údržbu. (Al Nuaimi & Kamel, 2011)

1.4.4 Technologie ultrazvukových vln

Ultrazvuk je akustické vlnění, jehož frekvence je nad hranicí slyšitelného zvuku (přibližně 20 kHz) a vyznačuje se pomalou rychlostí šíření a zanedbatelným pohlčováním kapalnými a pevnými látkami. To jsou výhody, které daly vzniknout myšlence navigování v interiéru budovy pomocí ultrazvukových vln. Pro výpočet vzdálenosti mezi vysílačem a přijímačem signálu se využívá doba letu signálu, což vyžaduje správnou časovou synchronizaci síťových uzlů. Přesnost dosažená prostřednictvím této techniky navigování nepřesahuje více než několik centimetrů, přestože je ovlivněna odraženými ultrazvukovými signály a dalšími zdroji hluku. (Medina et al., 2013; Gu et al., 2009) Nevýhodou tohoto způsobu polohování je požadavek na velký počet přijímačů, což je spojeno s vyšší cenou. (Koyuncu, & Yang, 2010) Z tohoto důvodu bývají ultrazvukové vysílače doplňovány radio-frekvenčními vysílači, čímž dochází k nárůstu oblasti pokrytí signálem (viz Obr. 4).



Obr. 4 Šíření signálu RFID a ultrazvukového signálu (převzato z Gu et al., 2009)

1.5 Porovnání polohovacích technik

V níže uvedené tabulce jsou porovnány stávající techniky polohování v interiéru z hlediska přesnosti, kterou tyto techniky poskytují, a jejich pořizovací ceny. V tabulce je také uvedena metoda měření, která byla pro přehlednost vyjádřena zkratkami:

- TDoA (Time Difference of Arrival) – časový rozdíl přijetí signálu
- ToA (Time of Arrival) – čas přijetí signálu
- AoA (Angle of Arrival) – úhel přijatého signálu
- RSS (Received Signal Strength) – síla přijatého signálu

Uvedená tabulka vychází z článku autorů Alarifi et al. (2016).

Tab. 1 Porovnání polohovacích technik

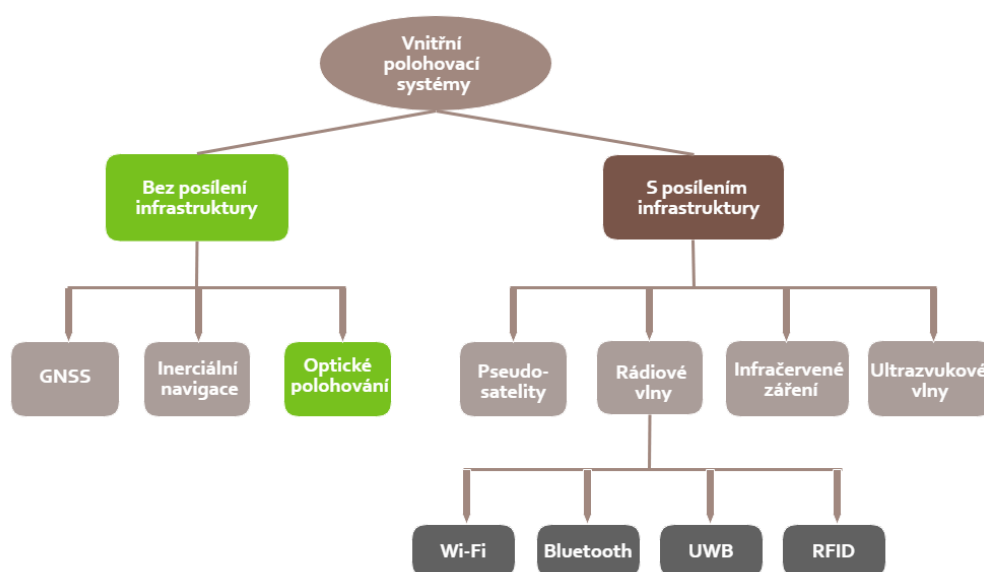
Technika	Metoda měření	Poznámka	Přesnost	Cena
<i>GNSS</i>	TDoA	Přijímače s vysokou citlivostí	2 m	vysoká
<i>Inerciální navigace</i>	Změna zrychlení	Doplnění další technikou	1 m	vysoká
<i>Optické polohování</i>	Rozpoznávání obrazu	-	30 cm – 1 m	nízká
<i>Pseudosatelity</i>	RSS	-	Subcentimetrová ¹	vysoká
<i>Wi-Fi</i>	RSS	-	2 – 4 m	střední
<i>Bluetooth</i>	RSS	-	Subdecimetrová ²	vysoká
<i>UWB</i>	ToA/TDoA	-	Subdecimetrová	vysoká
<i>RFID</i>	RSS	Pasivní	2 - 4 m	nízká
	RSS	Aktivní	1m	vysoká
<i>Infračervené záření</i>	AoA	-	Subcentimetrová	vysoká
<i>Ultrazvukové vlny</i>	ToA	-	Subdecimetrová	vysoká

Pro efektivní navigování v interiéru je důležitá nejen vyšší přesnost určení polohy uživatele, ale také jednoduchost jejího určení. To souvisí s nízkými pořizovacími náklady, minimální

¹ Edge & Jobs (2001)

² Bluetooth SIG, Inc. (2019)

údržbou, nízkými náklady na údržbu a minimálním využitím nové infrastruktury. Vezmeme-li v potaz tyto požadavky, pak bychom měli opomenout techniky s posílenou infrastrukturou, neboť pořizovací náklady a náklady na údržbu jsou pro rozsáhlé interiéry vyšší. Využití těchto technik má význam ve skladových areálech a malých uzavřených prostorech, kde lze relativně levně infrastrukturu posílit, a to včetně speciálního hardware a software pro klienta, výměnnou za vyšší přesnost. Pro individuální navigaci ve volně přístupných interiérech jako jsou nemocnice, letiště nebo obchodní centra nelze počítat s jiným klientem, než běžným mobilním telefonem. V souvislosti s využitím mobilního telefonu bychom se proto měli soustředit na takové techniky, které pro určení polohy nevyužívají žádnou další infrastrukturu. GNSS a inerciální navigace však samy o sobě neposkytují dobré výsledky, a proto bývají často doplňovány dalšími technikami určování polohy, čímž roste jejich cena a požadavky na hardware a software. Poslední technikou, která nevyužívá pro polohování v interiéru další infrastrukturu, je technika optického polohování. Tato technika využívá pro určení polohy kameru, kterou je v dnešní době vybaven každý mobilní telefon. Optické polohování je tak charakteristické nízkou pořizovací cenou a navíc dle uvedených zdrojů poskytuje velice dobré výsledky v určení polohy uživatele. Z tohoto důvodu jsme techniku optického polohování shledali pro polohování prostřednictvím mobilního telefonu velmi efektivní a rozhodli jsme se naši pozornost v následujících kapitolách zaměřit na principy, na základě kterých dochází k výpočtu polohy ze snímku a dále pak algoritmům počítačového vidění, které jsou úzce spjaty s touto problematikou.



Obr. 5 Zvolená technika pro určení polohy uživatele v interiéru budovy

2 Teoretické základy určení polohy ze snímku a algoritmy počítačového vidění

Polohování na základě pořízeného obrazového záznamu je velice obsáhlé téma a v současné době existuje pro tento způsob určování polohy několik odlišných přístupů. Z tohoto důvodu je tato kapitola věnována principům výpočtu polohy ze snímku a algoritmům počítačového vidění, které usnadňují řešení těchto úloh.

2.1 Souřadnicové systémy

Před tím, než přistoupíme k výkladu principů určení polohy prostřednictvím snímku, je nejprve nutné definovat několik souřadnicových systémů. Zde je potřeba dodat, že jsou všechny níže popsané souřadnicové systémy pravoúhlé a pravotočivé.

- *Objektový (geodetický) 3D souřadnicový systém (x_i, y_i, z_i)* : Objektové souřadnice vyjadřují skutečný tvar, velikost a polohu předmětů.
- *Souřadnicový systém kamery (X_c, Y_c, Z_c)* : Jedná se o souřadnicový systém vztažený ke kameře, jehož počátek se nachází v projekčním centru kamery C . Vztah mezi objektovým souřadnicovým systémem lze popsat pomocí translace t a rotace R .
Poloha bodu X_i v objektovém souřadnicovém systému popsána vektorem $X_i = [x_i, y_i, z_i]^T$. Chceme-li tentýž bod vyjádřit v souřadnicovém systému kamery, pak je nutné provést rotaci a posun tohoto bodu:

$$X_c = \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = R [X_i - t], \quad (1)$$

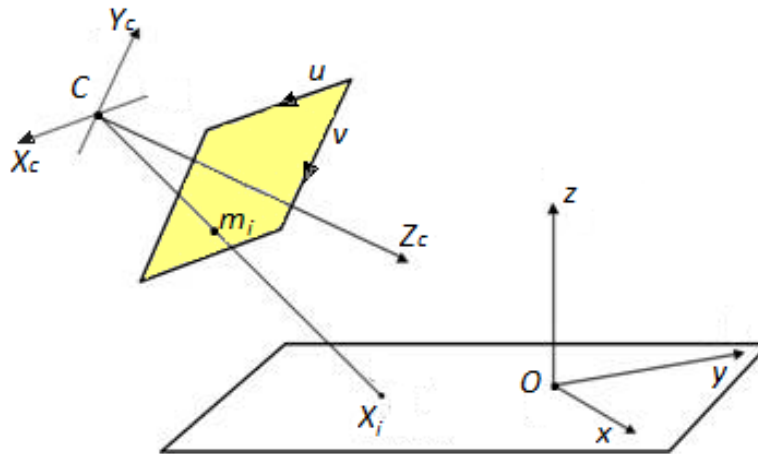
kde R je matice rotace kamery a t je vektor posunu (translace) bodu X_i .

- *Souřadnicový systém snímku (u_i, v_i)* . Počátek tohoto souřadnicového systému leží v pravém horním rohu snímku a jeho osy u a v jsou rovnoběžné s osami souřadnicového systému kamery. (Schindler, 2014)

Bod X_c je promítnut do roviny snímku v bodě m_i :

$$m_i = \left[-f \frac{X_c}{Z_c}, -f \frac{Y_c}{Z_c}, f\right]^T, \quad (2)$$

kde f je ohnisková vzdálenost kamery. (Sonka et al., 2007)



Obr. 6 Znáornění souřadnicových systémů

Perspektivní projekce použitá pro promítání bodu X_i do bodu m_i v rovině snímku je nejčastěji vyjádřena v maticovém tvaru (viz kapitola 2.2). Z tohoto důvodu nejsou pro výpočet využívány kartézské souřadnice, ale homogenní souřadnice, jejichž výhoda spočívá především v jednotnosti maticové reprezentace.

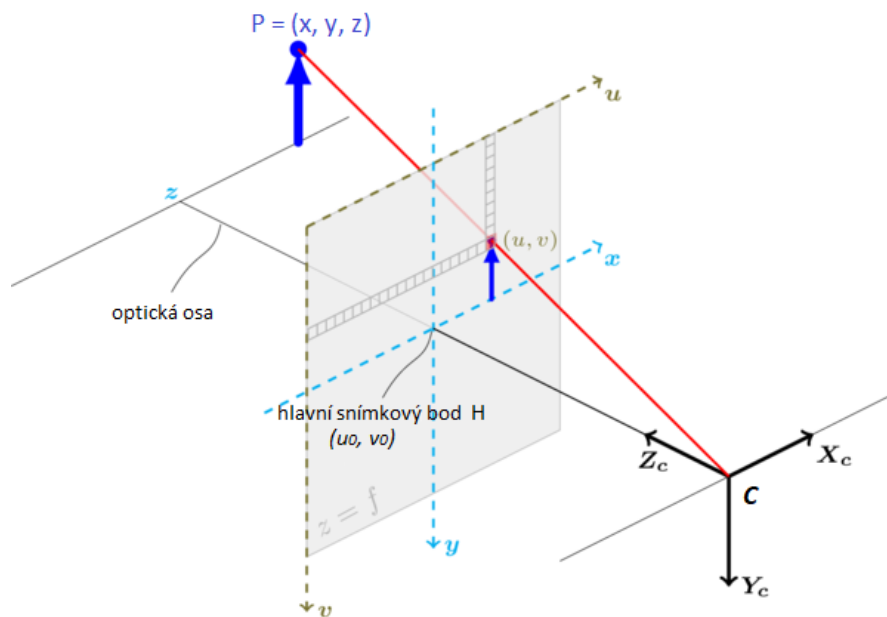
Zápis homogenních souřadnic libovolného bodu X_i s objektovými kartézskými souřadnicemi (X, Y, Z) je uspořádaná čtveřice (x, y, z, w) , pokud platí:

$$X = \frac{x}{w}, Y = \frac{y}{w}, Z = \frac{z}{w}, w \neq 0 \quad (3)$$

Čtvrtou souřadnicí w je označována váha bodu X . Pro jednoduchost výpočtů je váha bodu zvolena jako $w = 1$. 3D prostorové homogenní souřadnice jsou tedy zapsány pomocí vektoru $X_i = [x_i, y_i, z_i, 1]^T$ a obdobně pak snímkové homogenní souřadnice $m_i = [u_i, v_i, 1]^T$.

2.2 Geometrie jednoho pohledu

Základním modelem, který tvoří obraz v kameře, je perspektivní projekce popisující strukturu obrazu pomocí tzv. dírkového modelu kamery. Běžně používané objektivy kamer se snaží perspektivní projekci co nejvíce přiblížit, a proto je uvažována tzv. dírková kamera (pinhole camera), která je v geometrickém souladu s tímto modelem zobrazení. Princip dírkové kamery spočívá v tom, že každému bodu snímané scény odpovídá bod na snímku. Všechny paprsky, které spojují bod a jeho obraz, se protínají v projekčním centru C (viz Obr. 7). (Pavelka, 2003) Tato skutečnost je pro stanovení polohy projekčního centra kamery (polohy uživatele) z obrazového záznamu klíčová z toho důvodu, že pro její určení stačí znát několik korespondencí mezi objektovými a snímkovými souřadnicemi vřícovacích bodů (viz kapitola 2.2.2). Vřícovací bod je ve slovníku VÚGTK definován jako bod sloužící pro geometrickou transformaci nejčastěji obrazových dat do jiného souřadnicového systému.



Obr. 7 Model dírkové kamery (převzato a upraveno z OpenCV Documentation, 2014)

Přestože se běžné objektivy kamer snaží co nejvíce přiblížit modelu dírkové kamery, jejich reálné provedení se od této idealizace podstatně liší a je potřeba stanovit prvky vnitřní a vnější orientace, které kameru jednoznačně určují. (Pavelka, 2003) Mezi prvky vnitřní orientace patří konstanta komory f , poloha hlavního snímkového bodu $H = (u_0, v_0)$ a průběh radiální distorze. Polohu kamery popisují prvky vnější orientace. Jedná se o prostorové souřadnice středu vstupní pupily $C = (X_0, Y_0, Z_0)$, neboli projekčního centra

kamery, a tři nezávislé úhly rotace kamery ω , φ , κ . Parametr ω vyjadřuje rotaci kolem osy x , parametr φ rotaci kolem osy y a parametr κ rotaci kolem osy z .

2.2.1 Matematický model kamery

Model kamery tvoří lineární transformaci 3D projektivního prostoru do 2D projektivního prostoru. Mějme 3D objektové homogenní souřadnice bodu $\mathbf{X}_i = [x_i, y_i, z_i, 1]^T$ a snímkové homogenní souřadnice $\mathbf{m}_i = [u_i, v_i, 1]^T$ obrazu tohoto bodu. Obraz \mathbf{m}_i bodu \mathbf{X}_i vznikne v místě, kde paprsek procházející bodem \mathbf{X}_i a projekčním centrem \mathbf{C} protne rovinu snímku. Z toho vyplývá, že body \mathbf{X}_i , \mathbf{C} a \mathbf{m}_i musí být kolineární. Následující text byl převzat a upraven z Hartley & Zisserman (2003), Sonka et al.(2007).

Vztah mezi bodem \mathbf{X}_i a \mathbf{m}_i neboli model kamery je vyjádřen následovně:

$$s\mathbf{m}_i = \mathbf{K}[\mathbf{R}|\mathbf{t}]\mathbf{X}_i, \quad (4)$$

kde s je měřítko, \mathbf{R} je matice rotace kamery vůči objektovému souřadnicovému systému, \mathbf{t} je vektor translace kamery vzhledem k počátku objektového souřadnicového systému a \mathbf{K} je kalibrační matice kamery o velikosti 3×3 , která je vyjádřena následujícím způsobem:

$$\mathbf{K} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 1 & 1 \end{bmatrix},$$

kde α , β jsou měřítka os x , y ; parametr γ popisuje zkosení těchto os a u_0 , v_0 jsou souřadnice hlavního snímkového bodu (středu snímku).

Vztah (4) lze rozepsat následovně:

$$s \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}. \quad (5)$$

2.2.2 Projekční matice

Pokud jsou zvoleny 3D objektové souřadnice jako homogenní $\mathbf{X}_i = [x_i, y_i, z_i, 1]^T$, je možné zavést perspektivní projekci pomocí tzv. projekční matice \mathbf{P} o velikosti 3×4 , což převádí tento problém projekce na lineární.

Projekční matice je vyjádřena následujícím způsobem:

$$\mathbf{P} = \mathbf{K} [\mathbf{R}|\mathbf{t}] = \mathbf{K}\mathbf{R}[\mathbf{I}, -\mathbf{C}] \quad (6)$$

V této matici jsou ukryty prvky vnější i vnitřní orientace kamery. Kalibrační matice \mathbf{K} obsahuje prvky vnitřní orientace a matice $[\mathbf{R}|\mathbf{t}]$ popisuje pohyb kamery kolem statické scény nebo naopak pohyblivého objektu před statickou kamerou. Matice \mathbf{I} je jednotková matice a matice \mathbf{C} udává polohu středu promítání kamery.

Projekční matici lze odhadnout dvěma způsoby a to buď prostřednictvím známé scény, nebo neznámé scény. V následující podkapitole se budeme věnovat odhadu projekční matice ze známé scény. Pro odhad projekční matice z neznámé scény je potřeba využít minimálně dvou snímků. Tomuto případu bude věnována kapitola 2.3, ve které bude objasněna geometrie dvou pohledů.

2.2.3 Odhad projekční matice

Mějme známé homogenní 3D objektové souřadnice vřícovacího bodu $\mathbf{X}_i = [x_i, y_i, z_i, 1]$ jemu odpovídající homogenní 2D snímkové souřadnice $\mathbf{m}_i = [u_i, v_i, 1]$, pak lze jejich vztah vyjádřit následovně:

$$\mathbf{m}_i = \alpha_i^{-1} \mathbf{P} \mathbf{X}_i \quad (7)$$

Pokud vyjádříme součin kalibrační matice a rotační matice $\mathbf{Q} = \mathbf{K}\mathbf{R}$ a vektory této matice jako $\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3$ pak lze vztah (7) vyjádřit následovně:

$$\alpha_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \alpha_i^{-1} \begin{bmatrix} \mathbf{q}_1^T & q_{14} \\ \mathbf{q}_2^T & q_{24} \\ \mathbf{q}_3^T & q_{34} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}, \quad (8)$$

kde α_i je měřítko a $\alpha_i \neq 0$.

Na základě výše uvedené rovnosti obdržíme pro každou dvojici 3D objektových souřadnic vřícovacích bodů $\mathbf{X}_i = [x_i, y_i, z_i, 1]$ a jim odpovídajících 2D snímkových souřadnic $\mathbf{m}_i = [u_i, v_i, 1]$ dvě rovnice:

$$\begin{aligned} (\mathbf{q}_3^T \mathbf{X}_i + q_{34}) u_i &= \mathbf{q}_1^T \mathbf{X}_i + q_{14} \\ (\mathbf{q}_3^T \mathbf{X}_i + q_{34}) v_i &= \mathbf{q}_2^T \mathbf{X}_i + q_{24} \end{aligned} \quad (9)$$

Tyto rovnice obsahují celkem 12 neznámých ($P = 3 \times 4 = 12$), a proto je pro výpočet projekční matice potřeba znát 3D objektové souřadnice nejméně šesti vlíčovacích bodů, které navíc nesmí být v objektovém prostoru koplanární.

Tím dostáváme následující vztah:

$$\begin{bmatrix} \mathbf{X}_1^T & 1 & \mathbf{0}^T & 0 & -u_1 X_1^T & -u_1 \\ \mathbf{0}^T & 0 & \mathbf{X}_1^T & 1 & -v_1 X_1^T & -v_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{X}_k^T & 1 & \mathbf{0}^T & 0 & -u_k X_k^T & -u_k \\ \mathbf{0}^T & 0 & \mathbf{X}_k^T & 1 & -v_k X_k^T & -v_k \end{bmatrix} \cdot \begin{bmatrix} \mathbf{q}_1 \\ q_{14} \\ \mathbf{q}_2 \\ q_{24} \\ \mathbf{q}_3 \\ q_{34} \end{bmatrix} = 0 \quad (10)$$

$$\mathbf{A}\mathbf{q} = 0,$$

kde $\mathbf{A} \sim 2k \times 12$, $\mathbf{q} \sim 12 \times 1$, $k \geq 6$

Polohu a rotaci kamery lze vyjádřit pomocí následujícího vztahu:

$$\mathbf{Q} = \mathbf{K}\mathbf{R}$$

$$0 = \mathbf{P}\mathbf{C} = [\mathbf{Q}, \mathbf{t}] \begin{bmatrix} \mathbf{C} \\ 1 \end{bmatrix} = \mathbf{Q}\mathbf{C} + \mathbf{t}$$

$$\Rightarrow \mathbf{C} = -\mathbf{Q}^{-1}\mathbf{t}$$

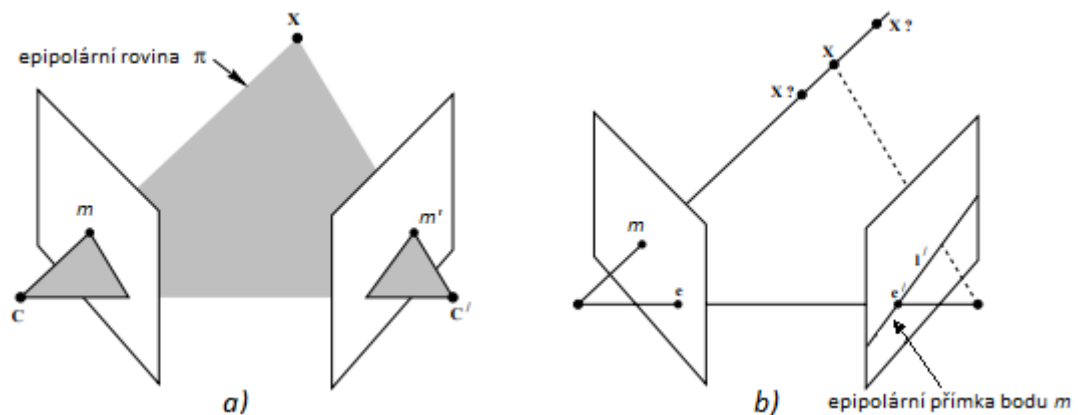
Pro separaci kalibrační matice \mathbf{K} a rotační matice \mathbf{R} je možné použít například \mathbf{QR} dekompozici matice, která ji rozkládá na horní trojúhelníkovou matici \mathbf{K} a ortogonální matici \mathbf{R} .

2.3 Geometrie dvou pohledů

Předchozí kapitola byla věnována geometrii jednoho snímku, jejíž základ je postaven na znalosti korespondencí snímkových souřadnic \mathbf{m} s objektovými 3D souřadnicemi téhož bodu \mathbf{X} ve 3D prostoru. Nyní předpokládejme, že bod \mathbf{X} ve 3D prostoru je zobrazený prostřednictvím dvou snímků, které byly pořízeny dvěma kamerami s různými polohami projekčních center. Na snímku, který byl pořízený první kamerou s projekčním centrem \mathbf{C} , je zobrazený 3D bod \mathbf{X} prostřednictvím snímkových souřadnic \mathbf{m} a na druhém snímku pořízeným kamerou s projekčním centrem \mathbf{C}' jako \mathbf{m}' . Chceme-li určit geometrický vztah mezi korespondujícími snímkovými souřadnicemi \mathbf{m} a \mathbf{m}' , případně projekčních center \mathbf{C} a \mathbf{C}' , je potřeba využít znalosti principů epipolární geometrie, což je ve zkratce projektivní

geometrie mezi dvěma snímky. Základní charakteristikou epipolární geometrie je její nezávislost na struktuře scény. Je závislá pouze na vnitřních parametrech kamery a jejich relativní orientaci.

Na Obr. 8 je zachycen základní princip epipolární geometrie. Proložení snímkových souřadnic bodu \mathbf{m} a \mathbf{m}' , bodu \mathbf{X} 3D prostoru a projekčních center kamer \mathbf{C} a \mathbf{C}' společnou rovinou π , dostaneme epipolární rovinu. Přímka, která je definována projekčními centry kamer \mathbf{C} a \mathbf{C}' se nazývá základna. V místě, kde dochází k průniku rovin obou snímků a základny, vznikají epipóly \mathbf{e} a \mathbf{e}' . Sestrojíme-li průnik epipolární roviny a obrazových rovin, pak obdržíme epipolární přímky \mathbf{l} a \mathbf{l}' . Odtud vyplývá výhoda, že pokud chceme obdržet korespondující obraz bodu \mathbf{X} ve druhé obrazové rovině, pak jeho vyhledávání můžeme omezit pouze na danou epipolární přímku a nemusíme tak vyhledáváním pokrýt celou obrazovou rovinu. Následující text byl převzat a upraven z publikace (Hartley & Zisserman, 2003).



Obr. 8 Geometrie korespondujících bodů: a) zobrazení epipolární roviny π , b) 3D souřadnice bodu \mathbf{X} , který je promítnut do obrazové roviny prostřednictvím snímkových souřadnic \mathbf{m} , musí ležet na přímce, která je definovaná projekčním centrem \mathbf{C} a snímkovými souřadnicemi \mathbf{m} - převzato a upraveno z (Hartley & Zisserman, 2003)

2.3.1 Fundamentální matice

Algebraickou reprezentací epipolární geometrie je tzv. fundamentální matice \mathbf{F} o rozměru 3×3 charakteristická hodností 2, jejímž úkolem je popis přechodu obrazu \mathbf{m} bodu \mathbf{X} z prvního snímku na druhý snímek prostřednictvím epipolární přímky \mathbf{l}' :

$$\mathbf{l}' = \mathbf{Fm} \quad (11)$$

Pokud je 3D bod X zobrazen na snímku pořízeného kamerou s projekčním centrem C pomocí snímkových souřadnic m a jemu korespondující bod na snímku pořízeného kamerou s projekčním centrem C' jako m' , pak lze vztah těchto bodů vyjádřit jako:

$$m'^T F m = 0 \quad (12)$$

Fundamentální matici je možné kromě snímkových korespondencí vypočítat prostřednictvím projekčních matic kamer P a P' :

$$F = [e'] \times P' P^+, \quad (13)$$

kde P^+ je pseudo-inverzní matice matice P a $e' = P' C$.

K výpočtu fundamentální matice je vyžadována pouze znalost korespondencí snímkových souřadnic bodů promítnutých v obou snímcích a není potřeba nutně znát vnitřní parametry kamer ani jejich relativní polohu. Pokud však chceme získat relativní polohu jednoho z projekčních center C a C' , je potřeba navíc znát vnitřní kalibrační prvky obou kamer a polohu projekčního centra jedné z kamer.

2.3.2 Odhad fundamentální matice

K odhadu fundamentální matice lze přistupovat dvěma rozdílnými způsoby, a to buď díky znalosti projekčních matic kamer P a P' za využití vztahu (13) nebo prostřednictvím získaných snímkových korespondencí m a m' , kdy neznáme vnitřní a vnější kalibrační prvky kamer.

Co se týče druhého případu, jsou metody výpočtu fundamentální matice rozděleny dle počtu získaných korespondencí mezi obrazovými rovinami. Jedná se o sedmibodový algoritmus a osmibodový algoritmus.

Sedmibodový algoritmus

Již z názvu tohoto algoritmu je patrné, že pro odhad fundamentální matice je potřeba sedm korespondujících snímkových souřadnic m a m' . Fundamentální matice je charakteristická sedmi stupni volnosti a sedmibodový algoritmus je tak nejjednodušší možnou variantou jejího odhadu. Mezi výhody tohoto algoritmu nepochybně patří nízký počet korespondencí, které jsou potřeba pro odhad fundamentální matice. Nevýhodou je však skutečnost, že generuje jedno až tři možná řešení. Navíc tento algoritmus nelze použít v případě nadbytečného počtu korespondencí, kdy je jejich počet $n > 7$. (Armangué & Salvi, 2003)

Osmibodový algoritmus

Pokud máme k dispozici počet korespondujících bodů $n \geq 8$, je pro odhad fundamentální matice možné aplikovat osmibodový algoritmus. Jeho výhoda oproti sedmibodovému algoritmu spočívá v jednoznačnosti řešení a možnosti využití redundantních korespondencí, které minimalizují chybu v odhadu. (Armangué & Salvi, 2003)

Uvažujme $n \geq 8$ korespondujících homogenních snímkových souřadnic $\mathbf{m}_i = [u_i, v_i, 1]^T$ a $\mathbf{m}'_i = [u'_i, v'_i, 1]^T$, pak po úpravě rovnice (12) dostaneme:

$$\begin{pmatrix} u_1 u'_1 & u_1 v'_1 & u_1 & v'_1 u'_1 & v_1 v'_1 & v_1 & u'_1 & v'_1 & 1 \\ u_2 u'_2 & u_2 v'_2 & u_2 & v'_2 u'_2 & v_2 v'_2 & v_2 & u'_2 & v'_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_n u'_n & u_n v'_n & u_n & v'_n u'_n & v_n v'_n & v_n & u'_n & v'_n & 1 \end{pmatrix} \begin{pmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ \vdots \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{pmatrix} = 0 \quad (14)$$

Vztah (14) lze zjednodušeně zapsat jako:

$$\mathbf{A}\mathbf{f} = 0 \quad (15)$$

Řešení rovnice (14) lze získat prostřednictvím lineárních metod pouze v případě, že má matice \mathbf{A} hodnost 8. Tento případ ale nenastává příliš často, a proto je nutné pro dosažení výsledku využít např. metodu nejmenších čtverců nebo RANSAC algoritmus popsany v kapitole 2.4.3. Počet korespondujících bodů je přímo úměrný počtu rovnic soustavy, a proto je výpočet takto rozsáhlé soustavy rovnic velice náročný. Navíc obecná řešení nalezená těmito metodami nespĺňují hodnost 2, kterou musí splňovat fundamentální matice. Nejčastěji se pro konečný odhad fundamentální matice používá SVD rozklad (Singular Value Decomposition), který bude dále představen v souvislosti s odhadem projekčních matic kamer z esenciální matice.

2.3.3 Esenciální matice

Esenciální matice \mathbf{E} je speciálním případem fundamentální matice ve smyslu využití normalizovaných snímkových souřadnic $\hat{\mathbf{m}}$ a $\hat{\mathbf{m}}'$, pro které platí vztah:

$$\hat{\mathbf{m}} = \mathbf{K}^{-1}\mathbf{m} \quad (16)$$

Historicky byla esenciální matice představena dříve než fundamentální matice a fundamentální matice by tak měla být považována za zobecnění esenciální matice, kdy pro její určení není potřeba znát vnitřní kalibrační prvky kamer. Vztah mezi korespondujícími normalizovanými snímkovými souřadnicemi $\hat{\mathbf{m}}$ a $\hat{\mathbf{m}}'$ je tak prostřednictvím esenciální matice vyjádřen velice podobně, jako v případě fundamentální matice:

$$\hat{\mathbf{m}}'^T \mathbf{E} \hat{\mathbf{m}} = 0 \quad (17)$$

Vztah mezi esenciální maticí a fundamentální maticí lze po dosazení vztahu (16) do vztahu vyjádřit jako:

$$\mathbf{E} = \mathbf{K}'^T \mathbf{F} \mathbf{K} \quad (18)$$

Nyní předpokládejme dvojici projekčních matic \mathbf{P} a \mathbf{P}' kamer s projekčními centry \mathbf{C} a \mathbf{C}' . Pokud projekční centrum \mathbf{C} první kamery leží v počátku souřadnicového systému 3D prostoru, pak lze projekční matici této kamery vyjádřit jako $\mathbf{P} = \mathbf{K}[\mathbf{I}|\mathbf{0}]$ a projekční matici druhé kamery jako $\mathbf{P}' = \mathbf{K}'[\mathbf{R}|\mathbf{t}]$. Esenciální matici lze vyjádřit následovně:

$$\mathbf{E} = [\mathbf{t}]_{\times} \mathbf{R} \quad (19)$$

Pokud projekční matici první kamery označíme jako \mathbf{P}_1 a projekční matici druhé kamery jako \mathbf{P}_2 , pak lze tyto projekční matice vyjádřit způsobem:

$$\mathbf{P}_1 = \mathbf{K}_1 \mathbf{R}_1 [\mathbf{I} | -\mathbf{C}_1] \quad (20)$$

$$\mathbf{P}_2 = \mathbf{K}_2 \mathbf{R} \mathbf{R}_1 [\mathbf{I} | -(\mathbf{C}_1 + \mathbf{t})]$$

Abychom však získali vektor translace \mathbf{t} , který je definován až na měřítko, a matici rotace \mathbf{R} , které společně popisují vztah mezi projekčními centry obou kamer, je nutné provést SVD rozklad matice \mathbf{E} , který je popsán v následující kapitole.

2.3.4 Odhad projekčních matic z esenciální matice

Odhadu projekčních matic kamer, které zachycují téže scénu ze dvou různých úhlů pohledu, lze docílit v případě, že známe informaci o jejich relativní poloze (máme k dispozici vektor translace \mathbf{t} a matici rotace \mathbf{R}). Obě informace jsou obsaženy v esenciální matici \mathbf{E} . Nejjednodušším způsobem, kterým lze separovat dané informace o posunu a rotaci mezi kamerami je již zmíněný SVD rozklad.

Prostřednictvím SVD rozkladu získáme matice \mathbf{U} , \mathbf{S} , \mathbf{V} , díky kterým lze esenciální matici zapsat následujícím způsobem:

$$\mathbf{E} = \mathbf{USV}^T, \quad (21)$$

kde \mathbf{U} a \mathbf{V} jsou ortogonální matice o rozměru 3×3 a \mathbf{S} je diagonální matice, která vypadá následovně:

$$\mathbf{S} = \begin{pmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Diagonální prvky s obsažené v matici \mathbf{S} jsou singulární hodnoty matice \mathbf{E} , které dle podmínky $\det \mathbf{E} = 0$, musí sestávat ze dvou identických hodnot a jedné nulové hodnoty.

Pokud navíc definujeme ortogonální matici \mathbf{W} ve tvaru:

$$\mathbf{W} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

můžeme podle následujícího vztahu odhadnout dvě možná řešení pro matici rotace \mathbf{R} :

$$\mathbf{R} = \mathbf{UWV}^T \text{ a } \mathbf{R} = \mathbf{UW}^T\mathbf{V}^T$$

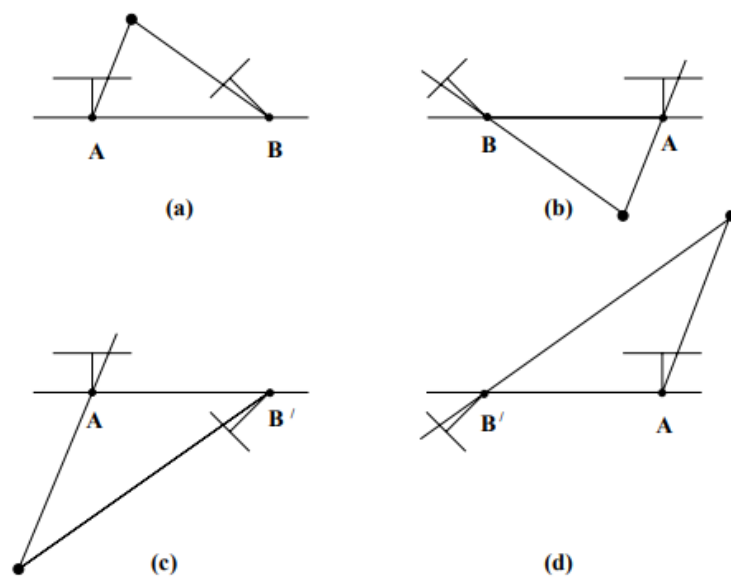
Vektor translace \mathbf{t} je definován třetím sloupcem matice \mathbf{U} :

$$\mathbf{t} = \mathbf{U}(0,0,1)^T = \pm \mathbf{u}_3$$

Vzhledem ke dvěma možným řešením pro matici rotace \mathbf{R} a dvěma řešením pro vektor translace \mathbf{t} , který je definovaný až na měřítko, získáváme jejich kombinací čtyři možná řešení pro výslednou projekční matici \mathbf{P}' :

$$\begin{aligned} \mathbf{P}' &= [\mathbf{UWV}^T | + \mathbf{u}_3], \mathbf{P}' = [\mathbf{UWV}^T | - \mathbf{u}_3], \mathbf{P}' = [\mathbf{UW}^T\mathbf{V}^T | + \mathbf{u}_3], \\ & \mathbf{P}' = [\mathbf{UW}^T\mathbf{V}^T | - \mathbf{u}_3] \end{aligned} \quad (22)$$

Správné řešení z těchto čtyř možných je takové, kdy libovolný 3D bod snímané scény leží před oběma kamerami zároveň (viz Obr. 9). Abychom zjistili, zda daný 3D bod \mathbf{X} leží před oběma kamerami zároveň, je nutné použít triangulaci tohoto bodu ze snímkových souřadnic \mathbf{m} a \mathbf{m}' .



Obr. 9 Čtyři možná řešení vycházející z SVD rozkladu esenciální matice – pouze řešení a) je správné (převzato z Hartley & Zisserman, 2003)

2.4 Algoritmy související s vyhledáváním snímkových korespondencí

Při výpočtu fundamentální a esenciální matice je využíváno korespondencí snímkových souřadnic, na základě kterých je možné dávat snímky do souvislostí. Aby bylo možné nelézt korespondence mezi snímky, je potřeba nejprve vyhledat ve snímku významné body, čímž je myšleno takové místo ve snímku, které je dostatečně odlišitelné od svého okolí a je rozpoznatelné z více směrů a vzdáleností. Významné body lze rozdělit do dvou kategorií, a to podle způsobu jejich vzniku a podle detekční metody, na základě které jsou body vyhledávány. Do první kategorie řadíme přirozené významné body, což jsou takové body, které se přirozeně vyskytují ve scéně. Oproti tomu syntetické významné body, což mohou být například výrazně zbarvené geometrické útvary, se ve scéně objevují díky zásahu člověka. (Mian et al., 2008)

Významné body jsou vyhledávány ve snímku tzv. *feature detektory*, které zároveň zaznamenají jejich polohu v pixelech. Aby byly možné dávat tyto prvky do dalších souvislostí, je důležité je pomocí vhodného algoritmu popsat – tzv. *feature deskriptor*. Feature deskriptor kóduje informace do řady čísel a působí jako druh číselného „otisku prstu“, prostřednictvím kterého lze od sebe jednoznačně rozlišit jednotlivé významné prvky ve scéně. V ideálním případě by měla být tato informace invariantní v rámci transformace obrazu, což by mělo zajistit vyhledání stejných bodů, i přestože byl snímek nějakým způsobem transformován (např. změna měřítka, zkosení nebo pootočení). Dále je žádoucí, aby byl znovu vyhledán tentýž významný bod i přes působení fotometrických změn, což je například změna intenzity světla nebo jasů. (Grauman & Leibe, 2011)

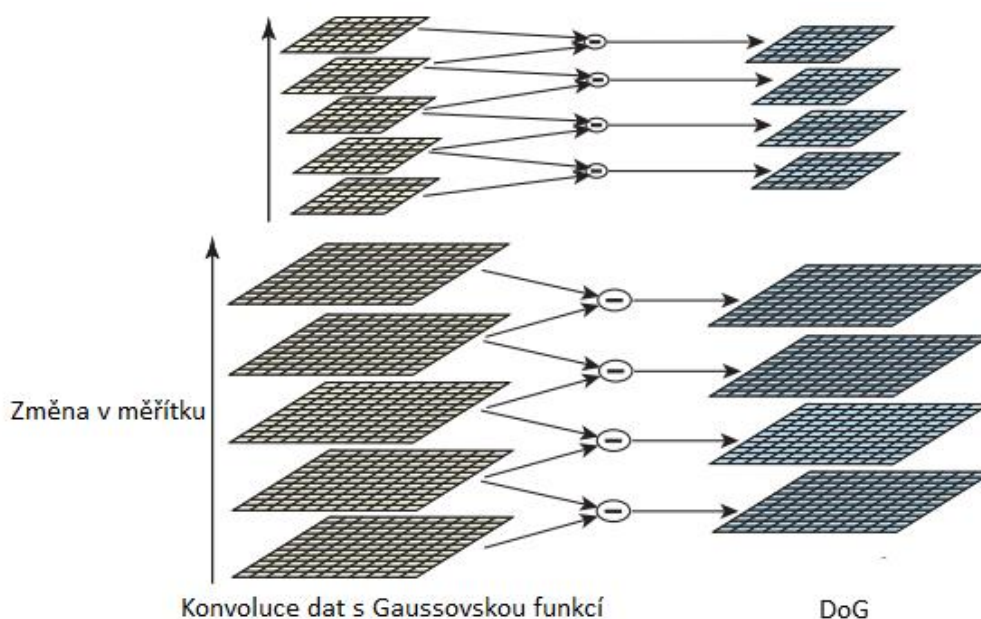
2.4.1 Detekční algoritmy

Pro automatizaci procesu detekce významných prvků ve snímcích, které mohou být dále užitečné mj. pro výpočet polohy kamery, bylo vyvinuto několik algoritmů. Nejznámějšími z nich jsou SIFT (Scale Invariant Feature Transform), SURF (Speed up Robust Feature) a ORB (Oriented Fast and Rotated Brief). SIFT algoritmus vynalezl v roce 2004 Lowe a o několik let později vznikl rozšířením tohoto algoritmu algoritmus SURF, který se vyznačuje nižší výpočetní náročností než SIFT. Nejmladší z těchto tří algoritmů je algoritmus ORB, který je alternativou pro oba zmíněné algoritmy.

SIFT

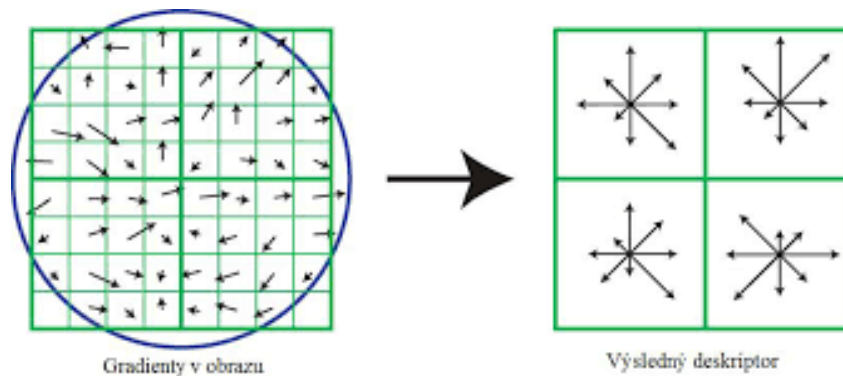
SIFT neboli Scale Invariant Feature Transform je metoda, která extrahuje významné invariantní body nezávisle na měřítku vstupních dat, což umožňuje spolehlivé porovnávání snímků téže scény pořízených z odlišných míst. Výpočetní náklady na extrahování významných bodů jsou minimalizovány přístupem tzv. kaskádového filtrování, kdy jsou výpočetně náročnější operace aplikovány jen v místech, která projdou počátečním testem. Princip SIFT algoritmu spočívá v následujících krocích:

- 1) *Nalezení kandidátů na významné body*: První etapou pro detekci významných bodů je identifikace míst a měřítek (tzv. *scale – space*), které mohou být opakovaně použity při změnách pohledu na téže scénu. To je implementováno prostřednictvím tzv. DoG (Difference of Gaussian) filtru. Nejprve je potřeba provést konvoluci obrazových dat s Gaussovskou funkcí s různým nastavením rozptylu a po vzájemném odečtení jednotlivých rozostřených obrazů je získáno měřítko obrazu. Nakonec jsou detekována lokální minima a maxima díky porovnávání každého pixelu s jeho osmi sousedy v daném měřítku.



Obr. 10 Princip vyhledávání scale-faktoru (převzato a upraveno z Lowe, 2004)

- 2) *Filtrace chybně vybraných bodů:* Jakmile jsou nalezeni kandidáti na významné body na základě porovnávání pixelů s jejich sousedy, následuje filtrace bodů, které mají nízký kontrast nebo jsou chybně lokalizovány podél hran.
- 3) *Přřazení orientace bodu:* Oblast významného bodu je následně rozdělena na menší segmenty, ve kterých jsou spočítány jednotlivé lokální orientace kontrastu, ze které je následně vypočtena hlavní orientace.
- 4) *Popis významného bodu:* Poslední fází tohoto algoritmu je přiřazení deskriptoru každému významnému bodu. K tomu dochází na základě vypočtené hlavní orientace kontrastu (viz Obr. 11). Z toho vyplývá, že jsou deskriptory gradienty vypočtené v okolí významných bodů. (Lowe, 2004)



Obr. 11 Princip SIFT deskriptoru (převzato a upraveno z Lowe, 2004)

SURF

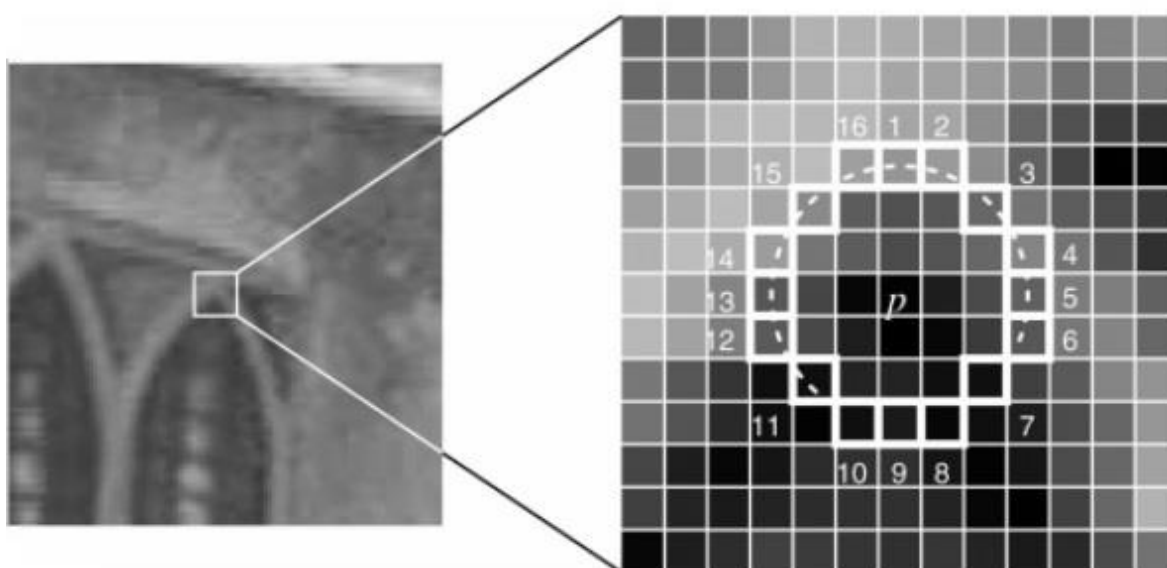
SURF algoritmus byl vyvinut tak, aby snížil výpočetní náročnost vyhledávání významných bodů ve snímku a urychlil tak proces lokalizace v reálném čase. Na rozdíl od metody SIFT je při použití SURF algoritmu požadováno předzpracování snímků, které spočívá v nahrazení snímku integrálním obrazem. Z integrálního obrazu je následně možné získat údaj o intenzitě oblasti snímku v konstantním čase s potřebou znalosti krajních bodů významné oblasti. K detekci významných bodů ve snímku je využit detektor, který je založený na výpočtu determinantu Hessovy matice. Popis nalezených významných bodů zajišťují tzv. Haar wavelets operátory, které jsou aplikovány na jeho okolí. (Bay et al., 2008)

ORB

Tento algoritmus je spojením algoritmu FAST detektoru významných prvků a BRIEF (Binary Robust Independent Elementary Features) deskriptoru nalezených významných prvků. (Rublee et al., 2011)

Princip FAST detektoru je následující:

- 1) Nejprve je vybrán pixel p , který by se mohl potenciálně stát významným bodem, a I_p je intenzita tohoto pixelu.
- 2) Dále je zvolena prahová hodnota t a je definováno okolí vybraného pixelu, jehož obvod tvoří 16 pixelů (viz Obr. 12).
- 3) Pixel p je významným bodem, pokud existuje množina n souvislých pixelů v definovaném okolí, které jsou jasnější než $I_p - t$ nebo tmavší než $I_p + t$
- 4) Posledním krokem je tzv. high – speed test, který vezme pixely označené na příkladu na Obr. 12 čísly 1, 5, 9, 13 a v případě, že se jedná o významný bod, pak alespoň tři z těchto pixelů odpovídají podmínce uvedené v bodu 3. (Rosten et al., 2010)



Obr. 12 Princip FAST detektoru (převzato z Rosten et al., 2010)

BRIEF je binárním popisem oblasti významného bodu vytvořený ze sady testů binární intenzity. Pokud uvažujeme okolí významného bodu, které bylo zbaveno šumu, pak je binární test τ definován takto:

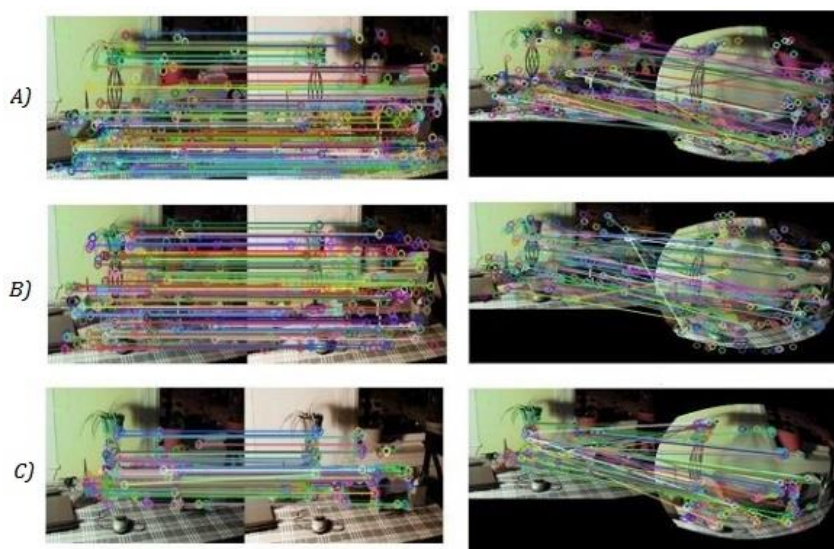
$$\tau(I; p, q) := \begin{cases} 1 & : I(p) < I(q) \\ 0 & : I(p) \geq I(q) \end{cases}$$

kde $I(p)$ je intenzita okolí významného bodu p . Významný bod je nakonec popsán vektorem, který vzešel z n binárních testů:

$$f_n(I) := \sum_{1 \leq i \leq n} 2^{i-1} \tau(I; p_i, q_i).$$

Teprve poté, co je oblast významného bodu popsána binárním řetězcem o velikosti 128, 256 nebo 512 bitů, je možné vyhledat korespondenci daného bodu na dalším snímku. (Calonder et al., 2010)

Karami et al (2017) ve svém článku porovnávali výkonnost výše uvedených algoritmů při jejich aplikování na transformované nebo zkrácené snímky. Na základě jejich výzkumu bylo prokázáno, že ORB je nejrychlejším algoritmem ze tří testovaných, ale jeho nevýhodou je, že ve většině případů detekuje významné body pouze uprostřed snímku. Naopak výpočetní rychlost SIFT algoritmu není tak dobrá jako u ORB nebo SURF algoritmu, ale vykazuje nejlepší výsledky pro většinu scén a detekuje významné body napříč celým snímkem (viz Obr. 13)



Obr. 13 Nalezené korespondence při změně barevné intenzity snímku (vlevo) a při soudkovitém zkrácení snímku (vlevo) A) SIFT, B) SURF, C) ORB (převzato z Karami et al., 2017)

2.4.2 Určení korespondencí mezi snímky

Poté, co jsou ve snímcích detekovány a popsány významné body, přichází na řadu vyhledávání korespondujících významných bodů napříč snímky, což je výpočetně nejvíce nákladná část mnoha algoritmů počítačového vidění. Tato problematika je také označována

jako nalezení nejbližšího souseda. Korespondenci dvou významných bodů lze určit na základě dvou odpovídajících si deskriptorů, které jsou reprezentovány jako vektory v multidimenzionálním prostoru. Od korespondenčních algoritmů je očekávána automatizovanost, schopnost vyhledávat pouze pravé korespondence a nezávislost na působení fotometrických změn a transformací snímku.

2.4.3 Filtrace nepravých korespondencí

Přestože jsou stávající algoritmy pro vyhledávání a popis významných prvků ve snímku navrženy tak, aby byly odolné vůči fotometrickým změnám a transformacím snímku, není pravidlem, že jsou těžké body na vstupním snímku vždy detekovány a popsány stejně, čímž dochází k přiřazení nepravých korespondencí, které je nutné odstranit. Je to dáno především tím, že jsou korespondence vyhledávány mezi snímky, které byly pořízeny více kamerami s odlišnými projekčními centry. Vhodným doplněním deskriptorů detekčních metod je algoritmus RANSAC (Random Sample Consensus), kterým je možné zpřesnit odhady korespondencí významných bodů ve dvou snímcích, případně odstranit odlehlé hodnoty.

RANSAC

Jedním z nejvýznamnějších algoritmů, který byl primárně vyvinut pro řešení problému určování polohy, byl publikován Fischerem a Bollesem v roce 1981. Algoritmus RANSAC je iterativní metoda pro odhad parametrů matematického modelu ze souboru pozorovaných dat, která obsahují odlehlé hodnoty. Účelem algoritmu je odstranění vlivu odlehlých hodnot na hodnoty odhadů, a proto lze algoritmus RANSAC interpretovat také jako metodu detekce odlehlých hodnot. Již z názvu tohoto algoritmu je zřejmé, že pro výpočet matematického modelu je náhodně vybrán z celkového počtu zadaných hodnot pouze vorek n hodnot, který je dále rozšiřován konzistentními body z dané sady hodnot. (Derpanis, 2010) Z tohoto důvodu je pro odhadnutí co nejpřesnějšího matematického modelu velice důležité mít k dispozici co nejvyšší počet přesně stanovených hodnot. Hodnoty, které odpovídají odhadnutému matematickému modelu, se také nazývají „inliers“ a odlehlé hodnoty „outliers“.

Princip RANSAC algoritmu spočívá v pěti základních krocích:

- 1) Náhodný výběr podmnožiny dat ze vstupní sady dat, která je označena jako hypotetiční inliers.
- 2) Odhadnutí matematického modelu na základě hypotetických inliers.
- 3) Stanovení, kolik bodů z celkového počtu bodů odpovídá předdefinované toleranci. Data, která se vyznačují nižší odchylkou, než je stanovená tolerance, jsou označeny jako inliers a stávají se tak součástí konsensu. V opačném případě se jedná o outliers.
- 4) Pokud bylo v rámci souboru konsensů klasifikováno dostatečné množství dat, které je dáno předem stanovenou prahovou hodnotou T , model je považován za schválený.
- 5) V opačném případě jsou kroky 1 – 4 opakovány do určeného počtu iterací N .

Počet iterací N musí být stanoven dostatečně vysoko, aby byla zajištěna pravděpodobnost p (obvykle nastavena na hodnotu 0,99), že nejméně jedna sada náhodných prvků neobsahuje odlehlou hodnotu. Pokud označíme w^n jako pravděpodobnost, že všechny hodnoty obsažené v sadě jsou inlier hodnoty, a pravděpodobnost výběru nejméně jedné outlier hodnoty jako $1 - w^n$, pak je počet potřebných iterací vyjádřen následujícím vztahem:

$$N = \frac{\log(1 - p)}{\log(1 - w^n)}$$

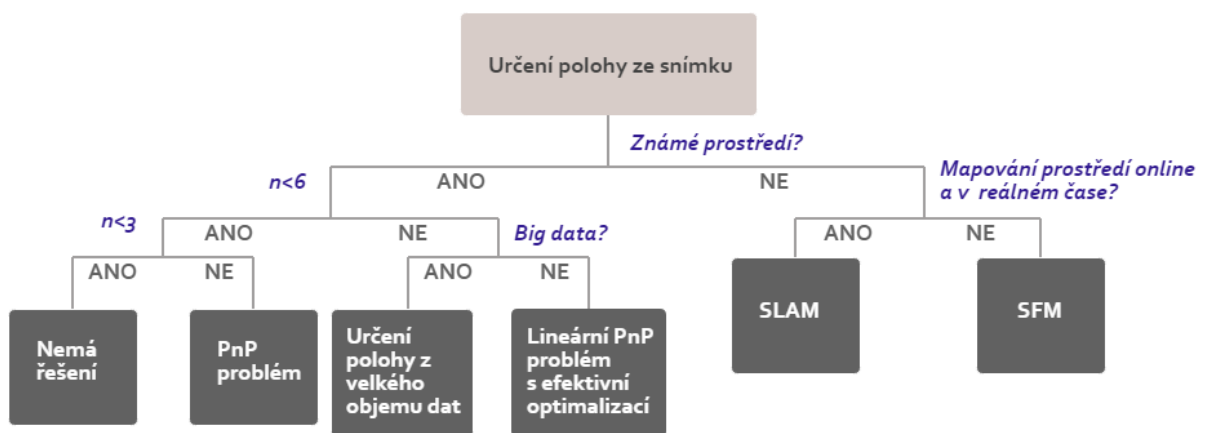
Pro algoritmus RANSAC bylo vyvinuto několik rozšíření jako je například MLESAC (Maximum Likelihood Estimate Sample Consensus) více viz (Torr & Zisserman, 2010) nebo MAPSAC (Maximum A Posterior Sample Consensus) více viz (Wang & Quan, 2008)

3 Přehled přístupů k polohování pomocí obrazového záznamu

Na základě toho, zda je prostředí předem známé či nikoliv, rozdělujeme v oblasti počítačového vidění metody výpočtu polohy ze snímku na dvě hlavní kategorie:

- polohování pomocí obrazového záznamu ve známém prostředí
- polohování pomocí obrazového záznamu v neznámém prostředí

Polohování ve známém prostředí znamená, že již před samotným pořizováním snímku, prostřednictvím kterého je určena poloha kamery, bylo provedeno mapování prostředí, což z pohledu fotogrammetrie znamená, že máme zaměřeny 3D souřadnice vlíčovacích bodů. (Wu & Li, 2018) Dáme – li tyto informace do souvislosti s předchozím výkladem, pak je pro výpočet polohy ve známém prostředí klíčová projekční matice, kterou lze určit prostřednictvím jednoho snímku. Oproti tomu na problém výpočtu polohy z neznámého prostředí je potřeba nahlížet z pohledu epipolární geometrie. (Civera et al., 2010)



Obr. 14 Rozdělení metod výpočtu polohy ze snímku (převzato a upraveno z Wu & Li, 2018)

3.1 Polohování ve známém prostředí

Jestliže máme k dispozici sadu 3D objektových souřadnic vlíčovacích bodů a zároveň víme, kam se tyto body promítnou ve snímku, mluvíme o známé scéně. To znamená, že máme k dispozici sadu korespondencí 3D objektových souřadnic vlíčovacích bodů a 2D snímkových souřadnic obrazů těchto bodů. V oblasti počítačového vidění je odhad pozice

kamery z n 2D – 3D bodových korespondencí základním a dobře popsáním problémem. Nejobecněji lze tuto úlohu řešit tak, že nejsou předem známy vnitřní kalibrační prvky kamery. Odhadu polohy (určení prvků vnější orientace) je možné dosáhnout na základě znalosti minimálně šesti 2D – 3D korespondencí vlíčovacích bodů za použití algoritmu přímé lineární transformace. Přímá lineární transformace ale nedokáže podchytit lokální deformace ve vzniklém digitálním obraze. (Obr & Koska) Existuje proto několik zjednodušení tohoto problému, které se proměňují v seznam rozličných algoritmů. Nejběžnějším zjednodušením algoritmu přímé lineární transformace je zjištění vnitřních kalibračních parametrů kamery před tím, než je odhadována samotná pozice kamery. Úloha, která řeší určování polohy kamery ze známých vnitřních kalibračních parametrů kamery, je v počítačovém vidění nazývána jako PnP problém (Perspective- n -point). (Penate – Sanchez et al., 2013)

Pokud se zaměříme na počet bodových korespondencí n , které jsou potřebné pro získání polohy projekčního centra kamery, pak lze PnP problém rozdělit do dvou základních skupin. První skupina metod využívá pro určení polohy $3 \leq n < 6$ bodů a druhá skupina pak $n \geq 6$ bodů. (Wu & Li, 2018)

3.1.1 Lineární PnP problém ($n \geq 6$)

PnP problém slouží k odhadu pozice kalibrované kamery z množiny n 3D souřadnic bodů a jim odpovídajících 2D snímkových souřadnic. Řešení problému PnP má původ v kalibraci kamery ze známé scény, kterou provádíme ve dvou krocích. Nejprve je potřeba odhadnout projekční matici P , prostřednictvím které lze následně odhadnout vnější a vnitřní parametry kamery. Projekční matice je tak pro řešení PnP problému klíčová.

V souvislosti s PnP problémem, kdy je počet bodů $n \geq 6$, jsou dosavadní studie zaměřeny na dva hlavní aspekty. Prvním aspektem je efektivní optimalizace výpočtu polohy kamery z relativně nízkého počtu bodů n a druhým pak dosažení rychlého odhadování polohy kamery z rozsáhlého objemu dat (big data). Co se týče prvního aspektu, Lepetit et al. (2009) poskytl přesné řešení PnP problému, které nazval $EPnP$ (Efficient PnP).

3.1.2 $EPnP$ problém

Jedná se o neiterativní metodu, která řeší PnP problém obecněji a pro výpočet polohy kamery využívá počet vlíčovacích bodů $n \geq 3$. Tato metoda je založená na představě, že každý z n bodů může být vyjádřen jako vážený součet čtyř virtuálních vlíčovacích bodů.

Souřadnice vlíčovacích bodů jsou nejprve řešeny jako neznámé a teprve po jejich výpočtu je vyřešena konečná pozice fotoaparátu. Aby bylo možné EPnP vyřešit, je stejně jako u PnP problému nutné předem znát vnitřní kalibrační prvky fotoaparátu. (Lepetit et al., 2009)

3.1.3 Nelineární PnP problém ($3 \leq n < 6$)

Posledním typ PnP problému spočívá v jeho nelineárním řešení, kdy je $n = 3, 4$ nebo 5 . Fischer & Bolles (1971) problém formulovali následovně:

„Vzhledem k daným relativním prostorovým polohám n kontrolních bodů a vzhledem k známým úhlům, které mezi sebou svírají každé dvojice kontrolních bodů s dalším bodem nazvaným jako centrum perspektivy (CP), najděte délky segmentů linií, které spojují CP s každým z kontrolních bodů.“

Jestliže zvolíme minimální počet bodů $n = 3$, pak se jedná o P3P problém, který lze vyřešit třemi korespondencemi snímkových souřadnic bodů a jim náležejících 3D objektových souřadnic. Nevýhodou tohoto přístupu je, že s pouhými třemi body přináší P3P problém až 4 možná řešení pro rotaci a posun kamery. V praxi se pro odstranění nejednoznačnosti využívá čtvrtá případně pátá korespondence (P4P/P5P problém), prostřednictvím kterých je vybráno nejlepší řešení z těchto čtyř možných. (Gao et al., 2003) Matematické principy výpočtu tohoto problému popisuje například (Zhang et al., 2005).

3.2 Polohování v neznámém prostředí

Nyní se věnujme případu, kdy předem neznáme prostředí, ve kterém se pohybujeme. Do této kategorie řadíme dvě hlavní metody počítačového vidění, které se od sebe liší v závislosti na tom, zda mapujeme neznámé prostředí online a v reálném čase, či nikoli. Pokud ano, jedná se o SLAM (Simultaneous Localization and Mapping). V opačném případě mluvíme metodě SFM (Structure From Motion), kdy k rekonstrukci prostředí dochází až poté, co je pořízeno dostatečné množství snímků statické scény z několika rozdílných míst. Společnou podstatou obou metod je detekování významných bodů ve snímcích, na základě kterých lze určit souvislosti mezi snímky pořízenými z rozdílných pozorovacích míst mapované scény.

3.2.1 SLAM

Úloha simultánní lokalizace a mapování propojuje úlohu určení polohy a mapování prostředí, ve kterém se pohybujeme. Podstatné při tom je, že k oběma úkonům dochází současně v reálném čase.

S ohledem na rozdílné přístupy k mapování prostředí dělíme SLAM do 4 kategorií:

- *Geometrický (metrický) SLAM* – úkolem tohoto přístupu je co nejpřesněji vypočítat 3D mapu prostřednictvím přesně definovaných matematických vztahů.
- *Learning SLAM* – u tohoto přístupu je před samotnou lokalizací a mapováním nutné nejprve vytvořit množinu dat, na základě které se platforma učí poznávat mapované prostředí.
- *Topologický SLAM* – tento přístup nevyžaduje přesné výpočty 3D mapy, ale reprezentuje prostředí prostřednictvím známých vztahů nebo topologie.
- *Marker SLAM* – k polohování a mapování prostředí je u tohoto typu SLAMu přístupováno na základě znalosti značek umístěných v mapovaném prostředí. (Wu & Li, 2018)

V současné době je s rozvojem oblasti strojového učení nejvíce diskutován přístup Learning SLAM, jehož výkonnost závisí na použité datové sadě a v důsledku toho má nízkou generalizační schopnost. Learning SLAM tedy není tak flexibilní jako 3D metrický SLAM a výsledná mapa prostředí není příliš přesná. Z tohoto důvodu zůstává v popředí, co se týče využitelnosti v praxi, metrický SLAM. Studií zabývajících se samotným topologickým SLAMem je naopak čím dál tím méně, protože se postupem času vyvinul v metrický SLAM. U Marker SLAMu mluvíme vzhledem k rozpoznávání známých značek v mapovaném prostředí o polo známém prostředí. Právě způsob reprezentace značek je hlavním tématem článků, které se tímto druhem SLAMu zabývají. (Wu & Li, 2018)

3.2.2 SFM

SFM se v poslední době stala velice populární metodou pro rekonstrukci 3D scény. Během procedury SFM je výpočet polohy kamery pouhým mezikrokem, který slouží pro obnovení trojrozměrné scény ze sady projektivních měření, které jsou reprezentovány jako sbírka dvojrozměrných obrazových sekvencí. (Wu & Li, 2018) Přestože je výpočet polohy a orientace kamery pouhou součástí celku metody SFM, patří tato metoda mezi přístupy, které využívají řešení polohy kamery a měla by být v této kapitole zmíněna.

SFM zahrnuje celkem tři fáze:

- Extrahování důležitých prvků ze snímků (body, linie apod.) a porovnání prvků napříč všemi snímky. Díky tomuto kroku jsou získány souvislosti (korespondence) mezi jednotlivými snímky.

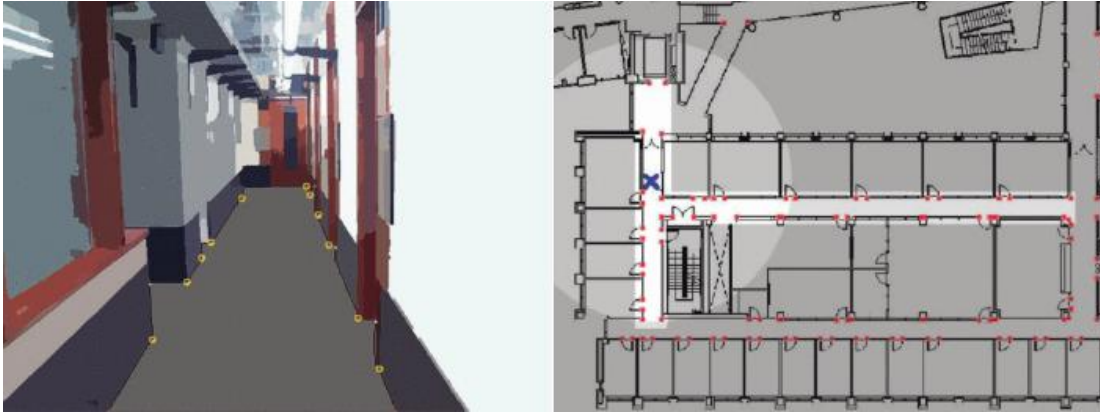
- Odhad pohybu kamery díky znalosti relativní polohy dvou snímků, která byla určena na základě korespondencí mezi jednotlivými snímky.
- Obnovení 3D struktury za využití odhadovaných pohybů kamery. (Özyeşil et al., 2017)

Pro řešení úlohy SFM existují dva rozdílné přístupy, přírůstková SFM a globální SFM. V přírůstkové SFM jsou do kolekce snímků postupně přidávány snímky s již vyřešenými polohami kamer, kdežto u globální SFM jsou pozice kamer řešeny najednou.

3.3 Rešerše publikací využívající pro určení polohy v interiéru kameru mobilního telefonu

Nejstarším přístupem tohoto způsobu polohování, je využití QR kódu, který se v souvislosti s určením polohy stává zdrojem dat obsahující navigační mapu nebo textové informace vztahující se k poloze. Nahradíme-li QR kódy snímky, pak mluvíme o metodě, která je analogická metodě tzv. otisku prstu. (Namiot, 2015) Tato metoda spočívá v odesílání pořízených snímků na webový server, kde dochází k porovnávání nově pořízeného snímku vůči databázi snímků mapující interiér budovy. Autoři Ravi et al. (2006) tvrdí, že přesnost v učení polohy tímto přístupem je 1m s více než 80% pravděpodobností.

Jeden z prvních přístupů využívající snímek získaný kamerou mobilního telefonu představili Hile et al. (2008) pro polohování v prostorách chodeb, kde se díky přítomnosti opakujících se prvků (rohy, přechody mezi podlahou a stěnou, dveře) vyskytuje mnoho přirozených značek. Namísto přímého vyhledávání značek autoři využili metodu segmentace obrazu, díky které byla nalezena celá plocha podlahy, ze které byly vysledovány její hrany (viz Obr. 15). Jejich přístup dále spočíval v nalezení těchto prvků v databázi půdorysu chodby, ve které jsou uloženy všechny důležité hrany, a následném výpočtu polohy uživatele díky získaným korespondencím mezi pořízeným snímekem a databází. Na základě navrženého postupu dosáhli chyby v určení polohy pohybující se okolo 0,30 m.



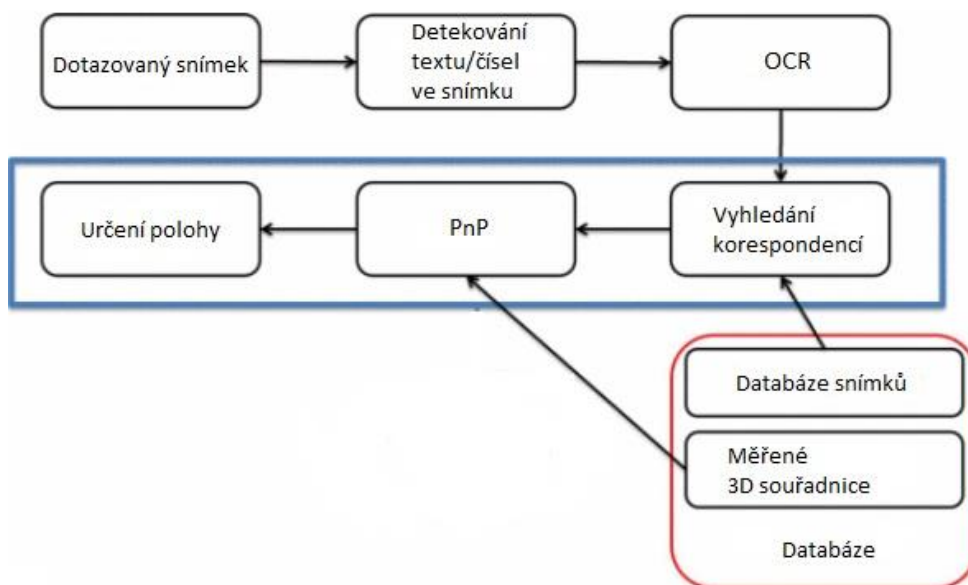
Obr. 15 Znáznornění významných prvků scény dle Hile et al. (2009)

Většina prací, které se zabývají problematikou polohování mobilních telefonů na základě porovnání obrazu, využívá SIFT algoritmus, který dokáže rozpoznávat charakteristické body obrazové funkce při změně měřítka, šumu a osvětlení. (Lowe, 1999) Příkladem takovéto práce může být například práce Liang et al. (2015), kde snahou autorů článku bylo dosáhnout přesnosti v poloze menší než 1 m za využití snímků pořízených mobilním telefonem. S využitím SIFT vlastností snímků bylo požadované přesnosti v poloze docíleno u více než 55% pořízených snímků. Obdobný přístup využili autoři Werner et al. (2011) s tím rozdílem, že pro vyhledávání korespondencí mezi snímky použili SURF algoritmus a dosáhli tak přesnosti v určení polohy 0,68 m.

Kawaji et al. (2010) pojednávají o navigování v prostředí muzea na základě všesměrně pořízených panoramatických snímků v intervalu 2 m, které tvoří obrazovou mapu. Jejich hlavním cílem je nalezení pozice uživatele s co nejvyšší přesností a za co nejkratší možnou dobu. Autoři k vyřešení tohoto problému využívají pro detekci významných prvků ve snímcích algoritmus PCA-SIFT, který extrahuje důležité prvky ve snímcích uložených v databázi a uživatelem pořízeném snímku. Na základě počtu extrahovaných prvků je vybrán nejlepší snímek z databáze a přiřazena uživateli poloha, která je převzata od snímku v databázi (metoda otisku prstů). Výsledky studie ukázaly, že výše uvedeným postupem lze odhadnout pozici za 2,2 s s 90% přesností.

Sadeghi et al. (2015) navrhli polohovací systém OCRAPOSE, který je rovněž založený na rozpoznávání významných prvků ve snímku a následném porovnávání nově pořízeného snímku vůči snímkům uložených v databázi. Jejich přístup k určení výsledné polohy uživatele se však od výše uvedených přístupů liší. Poloha projekčního centra kamery, kterou byl pořízený nový snímek, je vypočtena prostřednictvím PnP problému. Autoři umístili do

místností tabulky s textovými nebo číselnými informacemi, čímž implementovali metodu OCR (Optical Character Recognition). Korespondence jsou tak mezi snímky prostřednictvím číselných nebo textových charakteristik vyhledávány prostřednictvím SIFT algoritmu přesněji. 3D souřadnice rohů těchto tabulek jsou použity jako vstupní hodnoty pro výpočet PnP problému. Výzkum prokázal střední chybu v určení polohy menší než 0,50 m.



Obr. 16 Systém OCRAPOSE (převzato a upraveno Sadeghi et al., 2015)

Obdobný přístup zvolili autoři Deretey et al. (2015), kteří představili metodu, ve které pro určení polohy uživatele využívají jednu kalibrovanou monokulární kameru, jejíž poloha není závislá na znalosti polohy kamery z předchozího výpočtu. Rozdíl oproti předchozímu řešení je v tom, že nově pořízené snímky jsou porovnávány vůči předem vytvořenému 3D modelu mapovaného prostředí, který poskytuje 3D souřadnice pro výpočet EPnP problému. Přesnost techniky byla testována porovnáním získaných poloh s referenčními polohami, které byly obdrženy RGB-D SLAMem. Přestože navržené řešení vyžaduje tvorbu 3D modelu v offline fázi, jedná se o velice slibnou techniku, neboť standardní odchylka v každé souřadnici nepřesahovala hodnotu vyšší než 1 m.

Vývoj RGB-D čidel, které dokážou zachytit současně RGB snímek a informaci o hloubce, vedl k doplnění SLAMu o tato čidla a vznikl tak nový přístup nazvaný RGB-D SLAM. Tento přístup navíc Wang et al. (2014) doplnili o PnP řešení pro usnadnění lokalizačního procesu. Někteří, např. Williams et al. (2007), využívají PnP problém společně se SLAMem pro relokalizaci snímačů v případě, že dojde ke ztrátě polohy získané z předešlého snímání.

Další přístup založený na rozpoznávání obrazu zvolili Van Opendenbosh et al. (2014). Pro dosažení maximální odchylky 1 m v určení polohy mobilního telefonu byly pořízeny snímky v každé buňce mřížky o rozměrech 1x1 m a z každého místa bylo navíc pořízeno 16 snímků z odlišných pozorovacích úhlů. Tato strategie ale nemá dobré uplatnění v rozsáhlých prostorách, protože vysoký počet snímků v datové sadě má za následek vysoké výpočetní náklady a zvýšenou paměťovou náročnost na mobilní telefony a webové servery.

4 Navržené řešení pro určení polohy v interiéru prostřednictvím jednoho snímku

Důležitými požadavky při navrhování řešení této diplomové práce jsou využití kamery mobilního telefonu a automatizovanost celého procesu. To v praxi znamená, že by měl být uživatel schopný určit svoji polohu z jednoho snímku pořízeného kamerou mobilního telefonu bez nutnosti dalšího zásahu – například označování vřícovacích bodů ve snímku. Určování polohy prostřednictvím jednoho snímku, u kterého neznáme 2D – 3D korespondence vřícovacích bodů, však není možné. V nejjednodušším případě proto musí stát vedle uživatelem pořízeného snímku databáze snímků obsahující informaci o poloze projekčních center kamer, ze kterých byly snímky pořízeny. Tuto polohu je po prohledání databáze a nalezení nejvíce podobného snímku možné přiřadit uživateli. Uvědomme si však, že pokud bychom chtěli přiřadit uživateli polohu pouze na základě podobnosti vstupního snímku a snímku uloženého v databázi, musela by být databáze složena z velkého počtu snímků, jejichž prohledávání by mělo velký dopad na výpočetní náročnost navrženého řešení. Navíc by jako v případě Van Opdenbosh et al. (2014) přesnost určení polohy závisela na velikosti segmentů pomyslné sítě, ve které by probíhalo pořizování snímku do databáze.

Chceme-li tedy vytvořit databázi s co nejmenším počtem snímků a zároveň dosáhnout co nejvyšší možné přesnosti, musíme se na získanou dvojici vstupního a databázového snímku dívat z jiného úhlu pohledu. Poté, co obdržíme dvojici korespondujících snímků, můžeme polohu uživatele hledat prostřednictvím dvou snímků, které byly pořízeny dvěma kamerami s rozdílnými projekčními centry. Navržené řešení pro určení polohy uživatele v interiéru budovy je tedy založeno především na znalostech principů epipolární geometrie a PnP problému a je velice podobné řešení, které pro výpočet polohy kamer využívá metoda SFM.

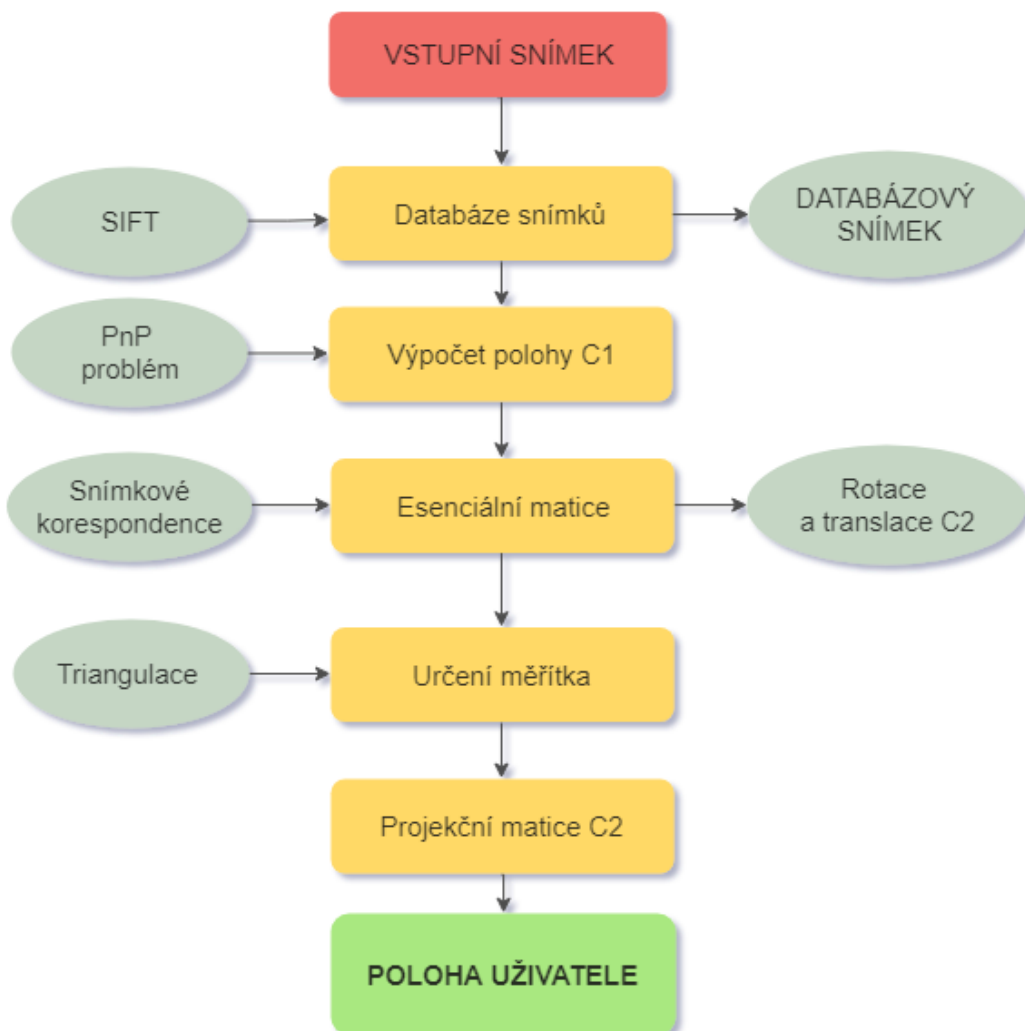
Podstata navrženého řešení je shrnuta v bodech níže a vizualizována na Obr. 17:

- Příprava podkladů pro tvorbu databáze - zaměření 3D souřadnic vřícovacích bodů a pořízení snímků interiéru
- Tvorba databáze snímků a databáze 2D – 3D korespondencí vřícovacích bodů
- Výběr korespondujícího snímku z databáze prostřednictvím SIFT algoritmu
- Výpočet projekčního centra kamery C_1^* , kterou byl pořízený snímek z databáze
- Výpočet esenciální matice z 2D korespondencí významných bodů, které byly obdrženy SIFT algoritmem

- Výpočet matice rotace a vektoru translace kamery C_2^{**} , kterou byl pořízený vstupní snímek
- Určení měřítka prostřednictvím triangulace vlíčovacích bodů ze snímkových souřadnic
- Výběr správného řešení projekční matice kamery C_2 a určení polohy uživatele

*pro jednoduchost je kamera, kterou byly pořízeny snímky do databáze, označena jako C_1

** pro jednoduchost je kamera, kterou byl pořízen vstupní snímek (uživatelé pořízený snímek), označena jako C_2



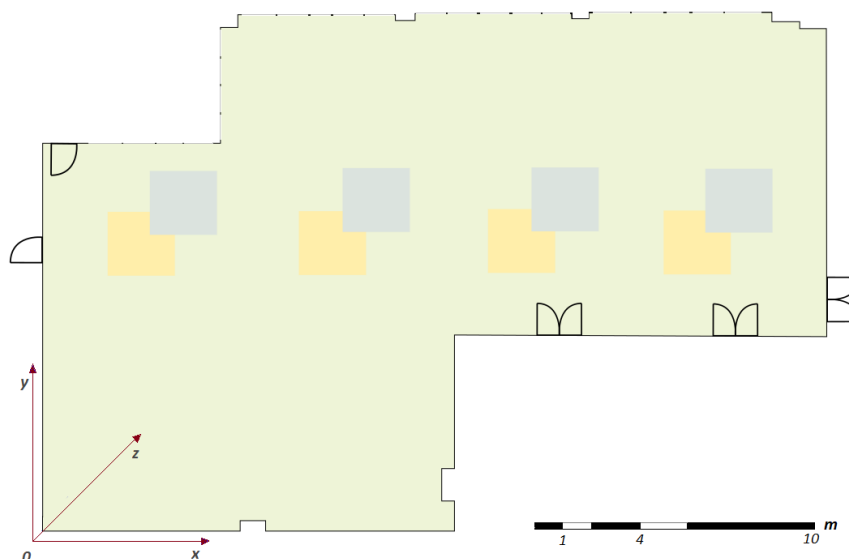
Obr. 17 Schéma navrženého řešení

Jednotlivé kroky řešení budou podrobně popsány v následujících kapitolách.

4.1 Příprava databáze

Klasická definice databáze uvádí, že databáze je systém souborů s pevnou strukturou záznamů, které jsou mezi sebou navzájem propojeny pomocí klíčů. V širším smyslu jsou součástí databáze i softwarové prostředky (systém řízení báze dat), které umožňují manipulaci s uloženými daty a přístup k nim. V našem případě je ale pod pojmem databáze myšlený adresář, do kterého budou uloženy snímky interiéru a náležející XML soubory daných snímků, což znamená, že se nejedná o databázi v pravém slova smyslu.

Před tím, než mohlo být přistoupeno k tvorbě samotné databáze, bylo nutné provést snímkování interiéru a zaměření 3D souřadnic vhodných vlíčovacích bodů. Vybraným interiérem, ve kterém bylo testováno navržené řešení určení polohy uživatele, jsou dostatečně rozsáhlé kancelářské prostory (viz Obr. 18).



Obr. 18 Půdorys interiéru budovy a zobrazení počátku objektového souřadnicového systému (modré a oranžové čtverce značí texturu podlahy)

Vlíčovací body, které budou později sloužit pro výpočet projekčního centra kamery C_1 , byly voleny tak, aby byly neměnné a dobře signalizované (rohy oken, dveří, místnosti, příp. vzor na podlaze). Zaměření 3D souřadnic vlíčovacích bodů bylo vzhledem k jednoduchému a pravidelnému členění prostoru provedeno metodou konstrukčních oměrných. Objektový souřadnicový systém byl zvolený jako pravotočivý místní souřadnicový systém s počátkem v jednom z rohů místnosti (viz Obr. 18). Snímkování interiéru budovy bylo provedeno kamerou mobilního telefonu iPhone SE, kterou bylo pořizeno celkem 60 snímků ve formátu JPEG s rozlišením 4032×3024 pixelů popisujících prostředí.

Účelem vytvořené databáze snímků je získání referenční polohy kamery, od které se bude dále odvíjet výpočet polohy uživatele. V předchozích kapitolách bylo zmíněno, že pro výpočet projekční matice a následné určení polohy kamery prostřednictvím jednoho snímku, je potřeba znát minimálně 6 2D – 3D korespondencí vlíčovacích bodů. Proto vedle snímků, které by měly dostatečně přesně charakterizovat interiér budovy, obsahuje vytvořená databáze také XML dokumenty snímků (viz Obr. 20), ve kterých je pro každý snímek uloženo minimálně 6 korespondencí snímkových a objektových souřadnic vlíčovacích bodů nacházejících se na daném snímku. Při výběru vlíčovacích bodů ve snímcích bylo nutné dbát také na to, aby vybrané vlíčovací body neležely pouze v jedné rovině (nebyly v reálném světě koplanární) a aby byly na snímku rozloženy co nejrovnoměrněji. Počátek souřadnicového systému snímku je v levém horním rohu snímku, kladná osa x vede vpravo a k ní kolmo dolů kladná osa y .



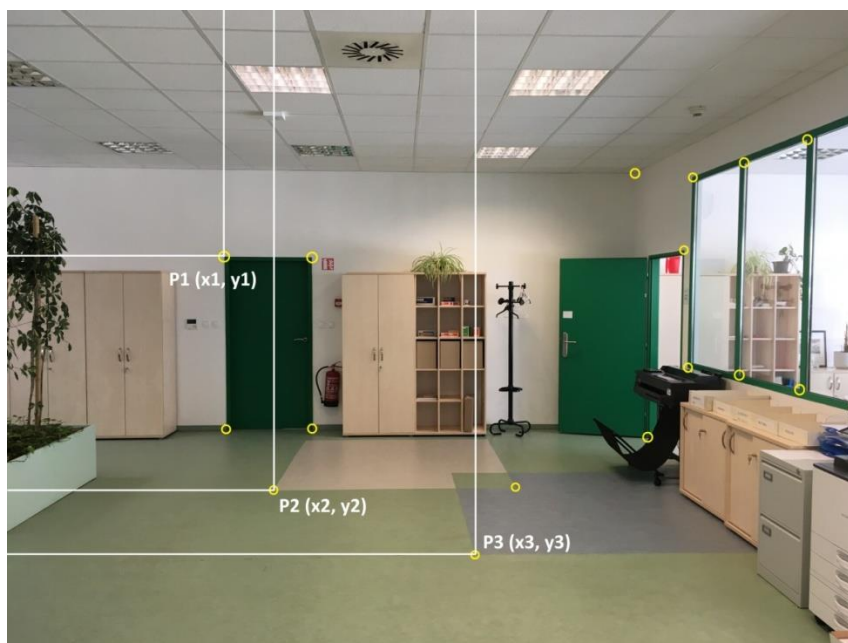
Obr. 19 Struktura vytvořené databáze

```

<data>
<keypoint>
  <name>dvere</name>
  <id>77</id>
  <image_coor_x>565</image_coor_x>
  <image_coor_y>195</image_coor_y>
  <world_coor_x>23.49</world_coor_x>
  <world_coor_y>10.80</world_coor_y>
  <world_coor_z>2.99</world_coor_z>
</keypoint>

```

Obr. 20 Struktura zápisu 2D – 3D korespondence vlíčovacího bodu v XML dokumentu



Obr. 21 Rozložení vlíčovacích bodů ve snímku uloženém v databázi

4.2 Použité programové vybavení

- **Agisoft Lens**

Agisoft Lens od společnosti Agisoft LLC je bezplatný software pro automatickou kalibraci fotoaparátu, který používá jako kalibrační pole LCD displej s šachovnicovým vzorem. Software poskytuje odhad úplné kalibrační matice kamery, včetně nelineárních koeficientů zkreslení. Odhadnuté kalibrační parametry fotoaparátu mohou být uloženy ve formátu XML, který lze následně použít jiným softwarem vyžadujícím přesné kalibrační parametry kamery. (Agisoft LLC, 2018; Agisoft LLC, 2011b)

- **Jupyter Notebook**

Jupyter Notebook je otevřená webová aplikace, která umožňuje vytvářet a sdílet dokumenty obsahující živý kód, rovnice, vizualizace a narativní text. Jedná se o webovou verzi python konzole, která umožňuje psát příkazy a kontrolovat výstup. Přestože je Jupyter napsaný v jazyce Python, podporuje i jiné jazyky jako je Julia a R. (Jupyter Notebook, 2019)

- **OpenCV**

OpenCV (Open Source Computer Vision Library) je knihovna zaměřená především na počítačové vidění a zpracování obrazu v reálném čase. Je vydávána pod licencí

BSD, a proto je zdarma pro akademické i komerční využití. Disponuje rozhraním C++, Python a Java a podporuje Windows, Linux, Mac OS, iOS a Android. Knihovna OpenCV byla navržena pro výpočetní efektivitu a je silně zaměřena na aplikace v reálném čase. OpenCV byla přijata po celém světě a má více než 47 tisíc uživatelů. Našla své využití v interaktivním umění, hledání min, sešívání map nebo v pokročilé robotice. (OpenCV Library, 2018)

- **Anaconda**

Anaconda je bezplatná open-source distribuce programovacích jazyků Python a R pro vědecké výpočty. Jejím cílem je zjednodušení správy balíčků a jejich vývoje. Verze balíčků jsou spravovány systémem správy balíčků Conda. (Anaconda, Inc., 2019)

4.3 Navržené řešení

Návrh řešení pro určení polohy uživatele v interiéru budovy byl realizován v prostředí Jupyter Notebook v jazyce Python za využití knihovny OpenCV poskytující mnoho funkcí, které lze pro zpracování této diplomové práce využít. V následujících podkapitolách budou detailně popsány jednotlivé kroky navrženého řešení ve spojitosti s vytvořeným zdrojovým kódem.

4.3.1 Načtení dat

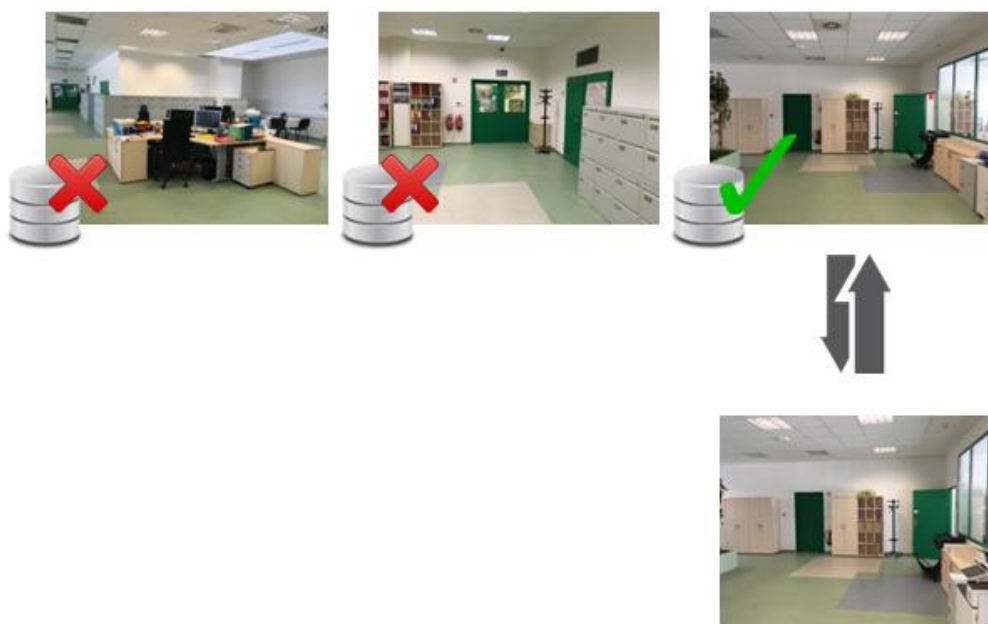
Prvním krokem je definování cest ke složkám, ve kterých jsou umístěny „databázové“ snímky (databaseImage) a snímek pořízený uživatelem, který si pro jednoduchost označíme jako vstupní snímek (inputImage).

Databázové snímky jsou umístěny v adresáři „data“ společně s XML soubory každého snímku a vstupní snímek je uložen v adresáři s názvem „vstup“. XML soubory s informacemi o polohách vlíčovacích bodů v objektovém a snímkovém souřadnicovém systému nesou stejný název jako příslušný snímek v databázi, čímž je umožněno jeho snadné vyhledání. Načtení snímků do aplikace Jupyter je zprostředkováno díky funkci `imread()`.

4.3.2 Výběr korespondujícího snímku z databáze

Po definování cest ke snímkům můžeme přistoupit k výběru takového snímku z databáze, který se nejvíce podobá vstupnímu snímku. K tomuto účelu jsme vytvořili funkci `getBestMatch()`, do které vstupují cesty k adresářům, ve kterých jsou uloženy databázové

snímky a cesta k adresáři vstupního snímku. Abychom takovýto snímek mohli nalézt, musíme postupně projít všechny snímky z databáze, vyhledat významné body ve snímcích a na základě vyhledaných významných bodů přiřadit korespondence mezi databázovými snímky a vstupním snímek. Korespondujícím databázovým snímek je snímek s nejvyšším počtem vyhledaných korespondencí.



Obr. 22 Výběr nejvíce odpovídajícího databázového snímku vůči vstupnímu snímku

Výběr vhodného detekčního algoritmu byl pro zpracování této diplomové práce stěžejní, neboť vyhledané významné body nebudou plnit pouze funkci vyhledání nejvíce odpovídajícího snímků z databáze, ale budou hrát také důležitou roli při výpočtu fundamentální matice, na základě které bude vypočtena esenciální matice.

Knihovna OpenCV nabízí pro detekci a popis významných bodů ve snímcích algoritmy SIFT, SURF a ORB, jejichž srovnání provedl Karami et al. (2017). Přestože je výpočetní rychlost použitých algoritmů neméně důležitým faktorem při procesu určování polohy, je pro nás kvůli následnému výpočtu esenciální matice ze získaných snímkových korespondencí podstatnější rovnoměrné a spolehlivé vyhledávání významných bodů ve snímku. Z tohoto důvodu byl pro vyhledávání významných bodů vybrán algoritmus SIFT, který je v knihovně OpenCV implementován pomocí funkce `detectAndCompute()`.

Poté, co jsou ve snímcích detekovány a popsány významné body, je dalším krokem nalezení korespondencí významných bodů mezi vstupním snímek a snímky uloženými v databázi.

Knihovna OpenCV nabízí pro vyhledávání korespondencí dvě strategie, Brute-Force Matcher a FLANN Matcher. Tím méně složitým je Brute-Force Matcher, jehož podstata spočívá ve spojení deskriptoru významného bodu v prvním snímku se všemi deskriptory ve druhém snímku a následnému výpočtu všech vzdáleností. Korespondujícím bodem je pak bod s nejkratší vypočtenou vzdáleností. Knihovna FLANN (Fast Library for Approximate Nearest Neighbours) obsahuje kolekci algoritmů, které jsou optimalizované pro rychlé vyhledávání nejbližších sousedů ve velkých datových souborech a navíc pracuje mnohem rychleji než Brute-Force Matcher. Z tohoto důvodu byl pro přiřazení snímkových korespondencí použit FLANN Matcher, konkrétně funkce `DescriptorMatcher::knnMatch()`, která vrací k – nejlepší shody prostřednictvím algoritmu k – nejbližšího souseda. Více viz (Muja & Lowe, 2014). Ukázka vyhledaných korespondencí mezi snímek z databáze a vstupním snímek jen na Obr. 23.



Obr. 23 Ukázka nalezených korespondencí mezi databázovým snímek (vlevo) a vstupním snímek (vpravo)

Vyhledaný počet korespondencí mezi vstupním snímek a databázovými snímky je ukládán do proměnné `maxGoodCount`, která je součástí pole `imageInfo` obsahující kromě počtu vyhledaných korespondencí také cestu ke konkrétnímu snímek z databáze a snímkové souřadnice vyhledaných významných bodů. Ukládání významných bodů do proměnných je důležité z toho důvodu, že budou dále využity při odhadu fundamentální matice. Jednotlivá pole `imageInfo` jsou ukládány do pole `imagesInfo`, kterým funkce `getBestMatch()` iteruje, dokud nenalezne snímek s nejvyšším počtem snímkových korespondencí `maxImageInfo`.

```

maxGoodCount = 0
maxGoodCountPosition = 0

i = 0
for imageInfo in imagesInfo:
    if imageInfo[1] > maxGoodCount:
        maxGoodCount = imageInfo[1]
        maxGoodCountPosition = i
    i += 1
maxImageInfo = imagesInfo[maxGoodCountPosition]
return maxImageInfo

```

Výpis 1 Zjištění snímku s maximálním počtem shod

4.3.3 Odhad esenciální matice

Po vyhledání nejlepší shody mezi databází a vstupním snímkem a nalezení korespondencí mezi těmito snímky, můžeme přistoupit k výpočtu fundamentální matice, ze které bude následně odhadnuta esenciální matice. Z Obr. 23 je zřejmé, že vyhledané korespondence mezi snímkem z databáze a vstupním snímkem nejsou vždy pravé. Aby tyto nepravé korespondence neovlivnily výsledek odhadu fundamentální matice, je při jejím odhadu využit algoritmus RANSAC, který tyto nepravé korespondence filtruje. Pro odhad fundamentální matice nabízí knihovna OpenCV funkci `findFundamentalMat()`, do které vstupují snímkové souřadnice významných bodů vstupního snímku a snímku z databáze a zvolená metoda odhadu fundamentální matice. V našem případě byla zvolena metoda `FM_RANSAC`, u které je nutné dále definovat parametr maximální vzdálenosti od bodu k epipolární linii v pixelech. Tato vzdálenost určuje, kdy je bod považován za odlehlý a nebude tak použit pro konečný odhad fundamentální matice. Parametr lze nastavit na hodnoty 1 – 3 v závislosti na přesnosti lokalizace významného bodu, rozlišení obrazu a šumu v obraze.



Obr. 24 Znáornění epipolárních přímek vedených významnými body ve snímku

Známe – li hodnoty odhadu fundamentální matice, přichází na řadu odhad esenciální matice, jejíž dekompozicí získáme matici rotace a směrový vektor translace popisující vztah mezi polohami projekčního centra kamery, kterou byl pořízený snímek z databáze, a kamery, kterou byl pořízený vstupní snímek. Ze vzorce (18) je zřejmé, že je nutné znát vnitřní kalibrační prvky kamer, kterými byly snímky pořízeny. Vzhledem k tomu, že jsme pro snímkování interiéru budovy použili kameru mobilního telefonu, u které nejsou předem známy vnitřní kalibrační prvky, muselo následovat jejich určení prostřednictvím procesu kalibrace. Pro tyto účely byl vybrán software Agisoft Lens, který využívá při kalibraci fotoaparátu model dírkové kamery. Celkem bylo pořízeno 24 snímků šachovnicového kalibračního pole, které je promítáno na LCD displej počítače. Po načtení pořízených snímků do programu byly vybrány požadované odhadované parametry. Jsou jimi horizontální a vertikální konstanta kamery f_x a f_y , souřadnice hlavního snímkového bodu c_x a c_y a koeficienty radiální a tangenciální distorze K_1, K_2, K_3 a P_1, P_2 .

Odhadnuté vnitřní kalibrační prvky kamery použitého mobilního telefonu byly uloženy do proměnných `calibration_matrix` a `distCoeff` a jsou uvedeny v Tab. 2.

Tab. 2 Výsledky kalibrace v programu Agisoft Lens

	Odhadnutá hodnota	Směrodatná odchylka
$f_x [px]$	3538,13	1,55
$f_y [px]$	3538,13	1,55
$c_x [px]$	2011,41	0,71
$c_y [px]$	1503,78	0,63
$K_1 [px^{-2}]$	0,06614	0,00173
$K_2 [px^{-4}]$	-0,07107	0,00905
$K_3 [px^{-6}]$	-0,10745	0,01435
$P_1 [px^{-1}]$	-0,000100	0,000055
$P_2 [px^{-1}]$	-0,000370	0,000068

Nyní lze přistoupit k odhadu esenciální matice, který je v kódu vyjádřen následovně:

```
E =
np.dot(np.dot(np.transpose(calibration_matrix),F_est),calibration_matrix)
```

Výpis 2 Odhad esenciální matice z fundamentální matice

Obdržíme – li hodnoty odhadu esenciální matice, můžeme provést její dekompozici, čímž získáme dvě možná řešení pro rotační matici a dvě řešení pro vektor translace mezi projekčními centry kamer C_1 a C_2 . Abychom obdrželi pouze jedno řešení pro rotační matici a vektor translace, použili jsme funkci `recoverPose()`, která je k tomu určena. Vstupy této funkce jsou odhadnutá esenciální matice a vyhledané korespondence mezi snímkem z databáze a vstupním snímkem:

```
R_est, t_est = cv2.recoverPose(E, bestImageInfo[2],bestImageInfo[3])
```

Výpis 3 Ukázka funkce `recoverPose()`

Funkce spočívá ve dvou krocích. V prvním provede dekompozici esenciální matice pomocí funkce `decomposeEssentialMat()`. Kombinací dvou řešení pro matici rotace a vektor translace získáme 4 možná řešení (viz kapitola 2.3.4) pro projekční matici kamery, kterou byl pořízený vstupní snímek. Ve druhém kroku jsou tato řešení testována prostřednictvím

triangulace snímkových korespondencí. V případě, že mají triangulované body kladnou hloubku, tzn., že leží před oběma kamerami, pak je toto řešení považováno za správné. Výstupem funkce `recoverPose()` je tedy správné řešení pro matici rotace R_{est} , která popisuje natočení kamery C_2 vůči kameře C_1 a směrový vektor translace t_{est} , který udává směr vektoru posunu projekčního centra kamery C_2 vzhledem k projekčnímu centru kamery C_1 .

4.3.4 Odhad projekční matice C_1

Přestože známe natočení a směr translace kamery C_2 vůči kameře C_1 , nemůžeme určit polohu jejího projekčního centra (polohu uživatele), dokud nebudeme znát polohu projekčního centra kamery C_1 v objektovém souřadnicovém systému, resp. projekční matici této kamery. Pro výpočet rotace a vektoru translace kamery C_1 využijeme funkci `solvePnP()`, do které vstupují snímkové souřadnice vřícovacích bodů a korespondující objektové souřadnice těchto bodů. Potřebné korespondence jsou uloženy v XML souboru náležejícího snímku z databáze, které je nutné převést do datové struktury pole. K tomuto účelu byla vytvořena funkce `getCoordsFromXML()`, jejímž vstupem je cesta k danému snímku. Funkce parsuje XML soubor a ukládá snímkové souřadnice vřícovacích bodů do pole s názvem `imageCoords` a objektové souřadnice do pole `worldCoords`.

```
def getCoordsFromXML(path):
    imageCoords = []
    worldCoords = []
    xmlPath =
path.lower().replace('.jpg', '.xml').replace('.jpeg', '.xml');
    xmlTree = ET.parse(xmlPath)
    xmlRoot = xmlTree.getroot().findall('keypoint')
    a = 0
    for child in xmlRoot:
        xmlX = int(child.find('image_coor_x').text)
        xmlY = int(child.find('image_coor_y').text)
        imageCoords.append([xmlX, xmlY])
        xmlWorldX = float(child.find('world_coor_x').text)
        xmlWorldY = float(child.find('world_coor_y').text)
        xmlWorldZ = float(child.find('world_coor_z').text)
        worldCoords.append([xmlWorldX, xmlWorldY, xmlWorldZ])
    return (imageCoords, worldCoords)
```

Výpis 4 Převod souřadnic z XML souboru do datové struktury pole

Výstupem funkce `solvePnP()` je matice rotace kamery C_1 `rvec` a vektor translace `tvec` v souřadnicovém systému kamery. Abychom získali polohu projekčního centra kamery C_1 v objektovém souřadnicovém systému, musíme nejprve určit projekční matici této kamery prostřednictvím funkce `getProjectionMatrix()`. Vstupy této funkce jsou matice `rvec` a vektor `tvec` a výstupy projekční matice kamery `projection_matrix` a matice rotace kamery `rot` udávající natočení kamery vůči objektovému souřadnicovému systému.

```
def getProjectionMatrix(PnP):
    rvec = PnP[1]
    tvec = PnP[2]
    rot = cv2.Rodrigues(rvec)[0]
    Q_t_in=np.concatenate((rot, tvec), axis=1)
    projection_matrix = np.dot(calibration_matrix,Q_t_in)

    return (projection_matrix, rot)
```

Výpis 5 Odhad projekční matice

Obdržíme – li projekční matici kamery C_1 , můžeme zavolat funkci `getPosition()`, která vrací polohu projekčního centra kamery v objektovém souřadnicovém systému.

```
def getPosition(P):
    rotation = P[:3,:3]
    translation = P[:,3]
    position = np.dot((np.linalg.inv(rotation)), translation)

    decomposition_P = cv2.decomposeProjectionMatrix(P)
    rotation_inv = np.linalg.inv(decomposition_P[1])

    direction = np.dot(rotation_inv, [[0],[0],[1]])
    direction = direction.T
    return (position, direction)
```

Výpis 6 Výpočet projekčního centra kamery

4.3.5 Odhad měřítka

Nyní víme, jakým způsobem je v objektovém souřadnicovém systému umístěna kamera, kterou byl pořízený snímek z databáze. Rovněž jsme na základě výpočtu esenciální matice a její následné dekompozice zjistili, jakým způsobem je vůči této kameře orientovaná

kamera, kterou byl získaný vstupní snímek. Po transformaci první kamery do objektového souřadnicového systému však musíme do tohoto souřadnicového systému transformovat také vektor translace kamery vstupního snímku t_est :

```
translation_obj_est = np.dot(np.linalg.inv(rot_db), t_est)
```

Výpis 7 Převod vektoru translace do objektového souřadnicového systému

Ačkoli nyní známe správný směr vektoru translace kamery C_2 vzhledem ke kameře C_1 , narážíme na velice zásadní problém. Ten spočívá v tom, že je odhadnutý směrový vektor jednotkový a neznáme jeho skutečnou velikost. Jinými slovy víme, jakým směrem je od kamery C_1 , která pořídila databázový snímek, umístěna kamera C_2 , kterou byl pořízený vstupní snímek, ale neznáme správnou velikost pomyslné základny mezi oběma kamerami. Abychom tedy zjistili skutečnou velikost odhadnutého směrového vektoru translace, musíme určit měřítko.

Jediným možným způsobem, jakým lze odhadnout měřítko směrového vektoru translace, je triangulace dvou vlíčovacích bodů prostřednictvím náležejících snímkových souřadnic obrazů těchto bodů. Pomocí získaného páru souřadnic je možné zjistit vzdálenost triangulovaných vlíčovacích bodů v souřadnicovém systému kamer, tedy před určením měřítka. Tuto vzdálenost označme jako d_{img} . Dále lze z objektových souřadnic vlíčovacích bodů jednoduše vypočítat jejich skutečnou vzdálenost d_{obj} v objektovém souřadnicovém systému. Nakonec můžeme výsledné měřítko určit pomocí vztahu:

$$s = \frac{d_{obj}}{d_{img}}$$

Před samotným provedením triangulace dvou vlíčovacích bodů je nutné nejprve získat snímkové souřadnice obrazů vybraných vlíčovacích bodů ve snímku. Co se týče snímku uloženého v databázi, snímkové souřadnice obrazů vlíčovacích bodů můžeme jednoduše extrahovat z náležejícího XML souboru snímku. Problém však nastává v případě snímku pořízeného uživatelem, u kterého bez manuálního označení vlíčovacích bodů ve snímku nevíme, kam se tyto body ve snímku promítnou.

Pro automatizované získání snímkových souřadnic obrazů bodů byla odhadnuta transformační matice homografie ze snímkových korespondencí vyhledaných SIFT

algoritmem. Na základě znalosti transformačního vztahu mezi vstupním snímkem a snímkem uloženým v databázi je možné následně promítnout obraz bodu ve snímku uloženém v databázi do vstupního snímku. Nicméně, dostatečně přesné transformace snímkových souřadnic dosáhneme pouze ve dvou případech. V prvním případě je matice homografie vypočtena z roviny a v této rovině se také nachází bod, který je transformován. Ve druhém případě je vypočtena ze snímků, které byly pořízeny kamerou s totožným projekčním centrem, což je princip, na základě kterého jsou pořizovány a následně spojovány panoramatické snímky. Vzhledem k tomu, že jsou naše snímky pořízeny kamerami s odlišnými projekčními centry, musíme zajistit, aby byly pro výpočet měřítka translace vybrány takové vlíčovací body, které leží v rovině, ze které byla vypočtena matice homografie (viz Obr. 25 a Obr. 26).

Výpočet transformační matice homografie je umožněn v knihovně OpenCV prostřednictvím funkce `findHomography()`, do které v našem případě vstupují proměnné `bestImageInfo[2]` a `bestImageInfo[3]` jakožto souřadnice snímkových korespondencí.



Obr. 25 Znáznornění snímkových korespondencí, ze kterých byla vypočtena transformační matice homografie



Obr. 26 Transformované snímkové souřadnice obrazů vřícovacích bodů ve vstupním snímku

Pro transformaci snímkových souřadnic obrazů vřícovacích bodů ze snímku z databáze do vstupního snímku byla vytvořena funkce `transformPoints()`, do které vstupují vybrané snímkové souřadnice obrazů vřícovacích bodů. Výběr vřícovacích bodů a jim náležejících snímkových souřadnic zajišťuje funkce `getGroupedCoordsFromXML()`, která prochází XML soubor snímku z databáze a volí pro transformaci pouze takové souřadnice, které leží v rovině, ze které byla vypočtena matice homografie. Konečnou transformaci snímkových souřadnic obrazů vřícovacích bodů do vstupního snímku pak zajišťuje funkce knihovny OpenCV `perspectiveTransform()`, do které vstupují snímkové souřadnice obrazů vřícovacích bodů na snímku z databáze a odhadnutá matice homografie o rozměru 3×3 .

```

def transformPoints(bestImageInfo):
    result = []
    coords = getGroupedCoordsFromXML(bestImageInfo[0])
    transformacni_matice = cv2.findHomography(bestImageInfo[2],
bestImageInfo[3], cv2.RANSAC,5.0)[0]
    for coord in coords[2][0]:
        P= np.array(coord,dtype=np.float32)
        P = np.array([[P]])
        transformace_bodu =
cv2.perspectiveTransform(P,transformacni_matice)
        transformace_bodu = np.array(transformace_bodu, dtype=np.int32)
        result.append(transformace_bodu)
    return result

```

Výpis 8 Transformace snímkových souřadnic obrazů vřícovacích bodů snímku z databáze do vstupního snímku

Když známe souřadnice obrazů vřícovacích bodů ve vstupním snímku, můžeme přistoupit k triangulaci vřícovacích bodů ze získaných snímkových souřadnic pomocí funkce `triangulatePoints()`. Vstupními parametry funkce jsou projekční matice kamery, kterou byl pořízen snímek z databáze P_1 , projekční matice kamery, kterou byl pořízen vstupní snímek P_2 a snímkové souřadnice obrazů vřícovacích bodů na vstupním snímku a snímku z databáze. Projekční matice kamery P_2 (`proj_mat_in`), kterou byl pořízený vstupní snímek, známe pouze v souřadnicovém systému kamery C_1 . Z tohoto důvodu byla matice P_1 (`proj_mat_db`) sestavena tak, aby bylo projekční centrum kamery C_1 v počátku objektového souřadnicového systému.

```

proj_cam_db =
np.dot(calibration_matrix,np.dot(rot_db,np.concatenate((I,C),axis=1)))

proj_cam_in =
np.dot(np.dot(np.dot(calibration_matrix,R_est),rot_db),np.concatenate(
(I,np.add(C,translation_obj_est)), axis=1))

```

Výpis 9 Projekční matice kamer sestavené pro účel triangulace snímkových souřadnic obrazů vřícovacích bodů

Poté, co obdržíme souřadnice triangulovaných vřícovacích bodů v souřadnicovém systému kamer, můžeme určit měřítko vektoru translace kamery C_2 prostřednictvím podílu

vzdálenosti vlíčovacích bodů v objektovém souřadnicovém systému a vzdálenosti vlíčovacích bodů v souřadnicovém systému kamer.

4.3.6 Odhad projekční matice C_2

Nakonec přichází na řadu odhad projekční matice kamery, kterou byl pořízený vstupní snímek, a s ní určení polohy uživatele. Odhad této projekční matice vychází ze vztahu (20) uvedeného v kapitole 2.3.3.

Do výpočtu tedy vstupuje kalibrační matice kamery C_2 , která je v tomto případě shodná s kalibrační maticí kamery C_1 , neboť byly oba snímky pořízeny téže kamerou. Dále pak rotační matice R_{est} popisující natočení kamery C_2 vzhledem ke kameře C_1 , rotační matice kamery C_1 (rot_{db}), poloha projekčního centra kamery C_1 a vektor translace mezi oběma kamerami t_{est} vynásobený vypočteným měřítkem.

Po odhadu projekční matice je posledním krokem zavolání funkce `getPosition()`, prostřednictvím které obdržíme polohu projekčního centra C_2 neboli polohu uživatele v interiéru.

4.4 Hodnocení výsledků navrženého řešení

Tato kapitola je věnována výsledkům navrženého řešení, na základě kterých je dále zhodnocena přesnost navrženého řešení. Aby bylo možné určit přesnost navrženého řešení, bylo nutné zjistit skutečnou polohu projekčních center kamer, ze kterých byly pořízeny všechny testované vstupní snímky. Nejprve byly pro účely testování využity jako vstupní snímky snímky uložené v databázi, u kterých je možné jednoduše určit polohy projekčních center kamer díky uloženým 2D – 3D korespondencím vlíčovacích bodů uložených v XML souboru za využití PnP problému. Přesnost určení polohy prostřednictvím PnP problému byla testována pořízením snímků z několika známých stanovisek. Výsledky testu prokázaly střední souřadnicovou chybu nepřesahující hodnotu vyšší než 0,05 m. Z tohoto důvodu byly takto odhadnuté souřadnice polohy považovány za správné. Testované snímky byly pořízeny ve stejný den a při stejném osvětlení (rozsvícená světla) jako snímky uložené v databázi. Aby tedy byla ověřena nezávislost řešení na fotometrických změnách, bylo pořízeno několik testovacích snímků při nižší intenzitě světla (zhasnutá světla), u kterých byly rovněž stanoveny polohy projekčních center, ze kterých byly snímky pořízeny. Jedná se o snímky IMG_5071, IMG_5072, IMG_5074, IMG_5082, IMG_5084 a IMG_5086. Zde je nutné dodat, že snímky byly pořízeny za ideálních podmínek, tzn. bez pohybu osob ve snímané scéně, beze změny snímané scény a stejnou kamerou mobilního telefonu, kterou byly pořízeny snímky do databáze.

Pro určení přesnosti navrženého řešení byla vypočtena střední souřadnicová chyba určení polohy dle Vyhlášky č. 26/2007 Sb., která uvádí následující vztah:

$$m_{xy} = \sqrt{\frac{1}{2}(m_x^2 + m_y^2)}, \quad (23)$$

kde m_x je střední chyba určení souřadnice x a m_y je střední chyba určení souřadnice y vypočtené ze vztahu:

$$m_x = \sqrt{\frac{\Delta x \Delta x}{n}}, \quad m_y = \sqrt{\frac{\Delta y \Delta y}{n}}, \quad (24)$$

kde Δx a Δy jsou ouřadnicové rozdíly vypočtené podle následujících vzorců a n je počet měření.

$$\Delta x = x_{PnP} - x_{est}, \Delta y = y_{PnP} - y_{est}, \quad (25)$$

kde x_{PnP} a y_{PnP} jsou souřadnice polohy stanovené z PnP problému a x_{est} a y_{est} jsou souřadnice polohy získané navrženým řešením.

Vzhledem k časové náročnosti tvorby databáze, zejména pak vytvoření XML souborů 2D – 3D korespondencí vlíčovacích bodů, bylo provedeno testování pouze dvou scén (viz Obr. 27). Celkem bylo pro první scénu provedeno testování 12 vstupních snímků, Tyto snímky byly testovány oproti sadě 14 snímků uložených v databázi zachycující téže nebo velice podobnou scénu. Pro druhou scénu bylo provedeno testování 5 vstupních snímků oproti databázi s 9 snímky.

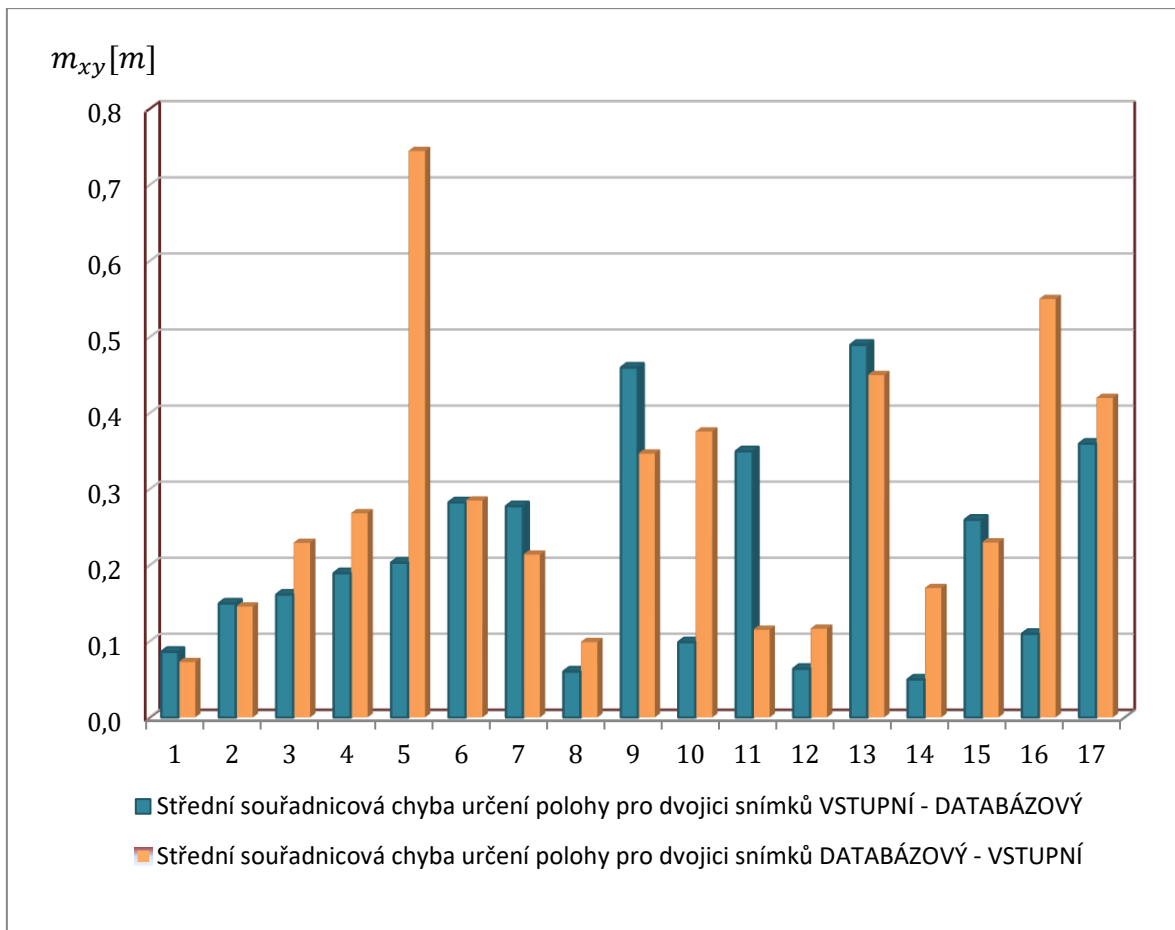
Po vyhledání korespondujícího databázového snímku a následném výpočtu polohy vstupního snímku (polohy uživatele) byla navíc získaná dvojice snímků vyměněna. To znamená, že byl vyhledaný databázový snímek použit jako vstupní a vstupní snímek jako databázový. Předpokladem bylo, že pro oba páry snímků získáme vlivem vyhledaných snímkových korespondencí velice podobný výsledek pro hodnotu střední souřadnicové chyby určení polohy. Přestože byla ve většině případů naše teorie potvrzena, byly zaznamenány případy, kdy se střední souřadnicové chyby určení polohy značně lišily (viz Tab. 3).



Obr. 27 Testované scény - scéna č. 1 (vlevo), scéna č. 2 (vpravo)

Tab. 3 Výsledky testu určení polohy na základě navrženého řešení (podrobné výsledky jsou uvedeny v příloze)

	Vstupní snímek	DB snímek	Počet korespondencí	Poloha projekčního centra DB snímku			Poloha projekčního centra vstupního snímku			Poloha uživatele			Střední souřadnicová chyba m_{xy} [m]
				X[m]	Y[m]	Z [m]	X [m]	Y [m]	Z [m]	X [m]	Y [m]	Z [m]	
1	I_4898	I_4962	2710	16,90	8,44	1,59	18,41	9,75	1,67	18,31	9,68	1,63	0,09
	I_4962	I_4898	2470	18,41	9,75	1,67	16,90	8,44	1,59	16,95	8,35	1,66	0,07
2	I_4297	I_4898	2621	18,41	9,75	1,67	16,14	8,72	1,64	16,21	8,92	1,77	0,15
	I_4898	I_4297	2638	16,14	8,72	1,64	18,41	9,75	1,67	18,25	9,62	1,64	0,15
3	I_4745	I_4743	2414	15,55	10,34	1,65	17,29	8,57	1,60	17,20	8,78	1,60	0,16
	I_4743	I_4745	2579	17,29	8,57	1,60	15,55	10,34	1,65	15,82	10,16	1,56	0,23
4	I_4954	I_4958	1769	12,40	8,42	1,63	12,64	9,66	1,59	12,76	9,90	1,57	0,19
	I_4958	I_4954	1502	12,64	9,66	1,59	12,40	8,42	1,63	12,52	8,78	1,60	0,27
5	I_4909	I_4908	2532	9,75	9,30	1,62	9,76	9,20	1,61	9,49	9,10	1,64	0,20
	I_4908	I_4909	2868	9,76	9,20	1,61	9,75	9,30	1,62	9,67	10,35	1,62	0,74
6	I_4956	I_4955	2435	12,49	9,68	1,65	12,23	8,44	1,69	12,47	8,12	1,63	0,28
	I_4955	I_4956	2516	12,23	8,44	1,69	12,49	9,68	1,65	12,33	10,05	1,70	0,29
7	I_4955	I_4954	3065	12,64	9,66	1,59	12,49	9,67	1,64	12,83	9,80	1,58	0,26
	I_4954	I_4955	2762	12,49	9,67	1,64	12,64	9,66	1,59	12,34	9,62	1,63	0,21
8	I_4961	I_4962	2852	16,90	8,44	1,60	15,17	8,39	1,60	15,25	8,36	1,60	0,06
	I_4962	I_4961	2835	15,17	8,39	1,60	16,90	8,44	1,60	16,76	8,44	1,60	0,10
9	I_5071	I_4956	1790	12,25	8,44	1,68	11,92	8,06	1,82	12,41	7,64	1,63	0,46
	I_4956	I_5071	1933	11,92	8,06	1,82	12,23	8,44	1,69	11,92	8,06	1,82	0,35
10	I_5072	I_4908	1637	9,75	9,30	1,62	9,85	9,21	1,60	9,71	9,21	1,60	0,10
	I_4908	I_5072	1739	9,85	9,21	1,60	9,75	9,30	1,62	9,86	8,78	1,60	0,38
11	I_5074	I_4743	2674	15,55	10,34	1,65	15,82	9,23	1,58	15,47	9,58	1,63	0,35
	I_4743	I_5074	2669	15,82	9,23	1,58	15,55	10,34	1,65	15,52	10,18	1,56	0,12
12	I_5082	I_4909	1577	9,76	9,20	1,61	9,72	9,22	1,59	9,81	9,21	1,60	0,06
	I_4909	I_5082	1473	9,72	9,22	1,59	9,76	9,20	1,61	9,60	9,24	1,75	0,12
13	I_4740	I_4742	1156	7,62	8,16	1,59	8,67	8,70	1,62	8,79	9,38	1,63	0,49
	I_4742	I_4740	881	8,67	8,70	1,62	7,62	8,16	1,59	7,93	7,61	1,62	0,45
14	I_5084	I_4913	1338	8,43	8,08	1,65	8,57	7,71	1,5	8,61	7,66	1,50	0,05
	I_4913	I_5084	1186	8,57	7,71	1,5	8,43	8,08	1,65	8,49	8,32	1,63	0,17
15	I_5086	I_4944	702	6,72	9,65	1,61	6,58	10,00	1,67	6,71	9,65	1,62	0,26
	I_4944	I_5086	718	6,58	10,00	1,67	6,72	9,65	1,61	6,74	9,98	1,67	0,23
16	I_4914	I_4913	2332	8,42	8,08	1,64	8,45	8,01	1,66	8,38	8,15	1,65	0,11
	I_4913	I_4914	2101	8,45	8,01	1,66	8,42	8,08	1,64	8,36	7,31	1,72	0,55
17	I_4944	I_4913	893	8,42	8,08	1,64	6,72	9,65	1,61	6,90	9,17	1,59	0,36
	I_4913	I_4944	1030	6,72	9,65	1,61	8,42	8,08	1,64	8,06	8,56	1,64	0,42

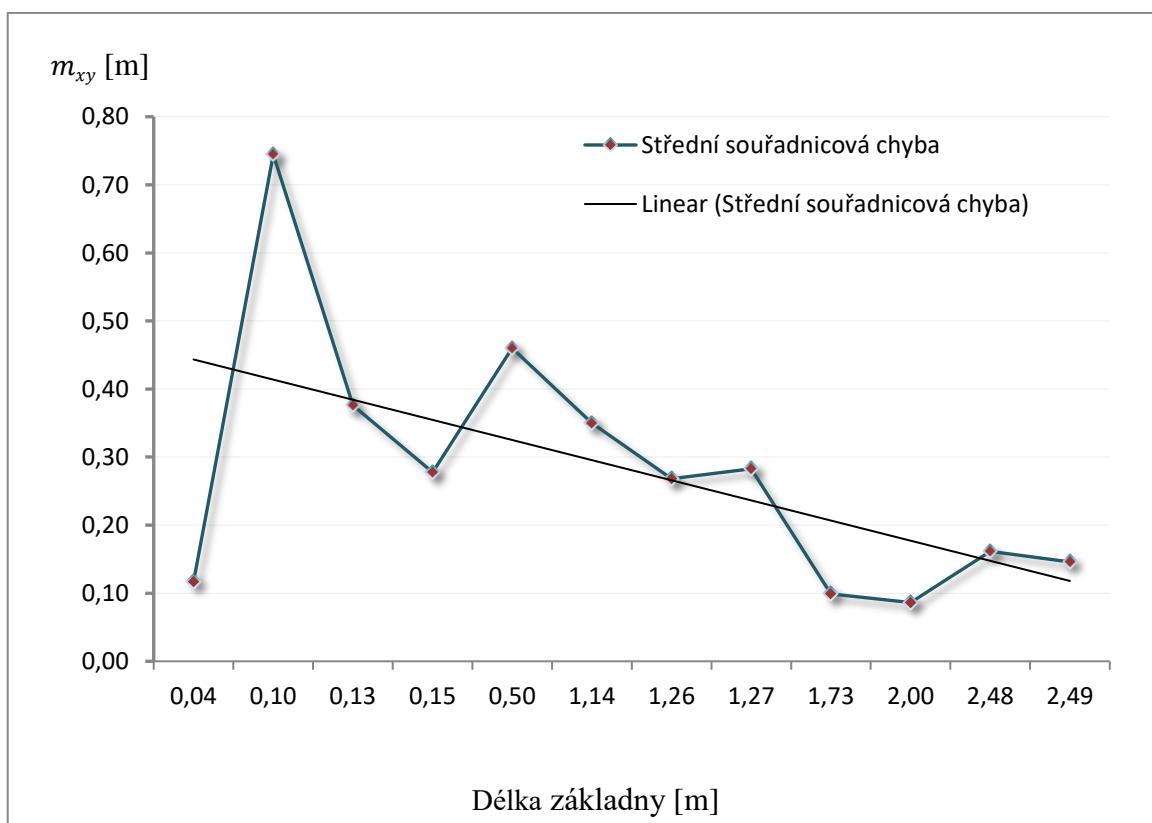


Graf 1 Výsledky testu navrženého řešení

Z výše uvedené tabulky a grafu je patrné, že se jedná o dvojice snímků s čísly 5 a 16. Pokud se zaměříme na dvojici s číslem 5, pak byl snímek IMG_4909 byl nejprve testován jako vstupní a na základě počtu 2532 vyhledaných korespondencí mu byl přiřazen snímek IMG_4908 uložený v databázi. Poloha projekčního centra kamery, kterou byl pořízený vstupní snímek IMG_4909, byla určena se střední souřadnicovou chybou 0,08 m, což lze považovat za výborný výsledek. Jakmile však byly snímky vyměněny, což znamená, že byl snímek IMG_4908 použit jako vstupní, obdrželi jsme překvapivý výsledek. I přes vyšší počet nalezených korespondencí (2868), jsme obdrželi hodnotu pro střední souřadnicovou chybu určení polohy 0,74 m. Vzhledem k tomu, že je tato chyba dána především velikostí odchylky v souřadnici Y, je pravděpodobné, že byl chybně určen vektor translace mezi projekčními centry kamer, kterými byly tyto snímky pořízeny. Chybné stanovení vektoru translace jsme zaznamenali zejména v případech, kdy délka základny mezi projekčními centry kamer, kterými byly pořízeny dvojice databázového snímku a vstupního snímku, byla natolik krátká, že projekční centra kamer byla téměř totožná (viz Přílohy). V případě dvojice snímku č. 5 byla základna mezi projekčními centry kamer dlouhá 0,10 m a u dvojice č. 16

pouhých 0,08 m. Minimální rozdíly ve stanovené přesnosti naopak můžeme zaznamenat u dvojic č. 1, 2 a 6, u kterých byla základna mezi projekčními centry kamer delší než 1,50 m.

Výsledky uvedené v příloze dokazují, že příliš krátká základna mezi projekčními centry kamer, kterými byly pořízeny vstupní snímky a snímky uložené v databázi, nepříznivě ovlivňuje orientaci směrového vektoru translace, který jsme obdrželi z odhadu esenciální matice. Nicméně, ve většině případů byly tyto chybné odhady kompenzovány odhadem měřítka a chyby v určení polohy tak nedosahovaly vysokých hodnot. Příímka proložená hodnotami středních souřadnicových chyb v níže uvedeném grafu potvrzuje, že v testované sadě snímků klesá s delší základnou střední souřadnicová chyba v určení polohy uživatele. Lze předpokládat, že tento trend je obecný, nebyl ale dostatečně statisticky testován.



Graf 2 Vliv délky základny na střední souřadnicovou chybu určení polohy

Pokud se zaměříme na počet vyhledaných korespondencí a střední souřadnicovou chybu v určení polohy pro jednotlivé snímkové dvojice (viz Tab. 3) zjistíme, že v mnoha případech je střední souřadnicová chyba určení polohy vyšší i přes vyšší počet vyhledaných snímkových korespondencí (viz Tab. 4).

Tab. 4 Vliv počtu vyhledaných snímkových korespondencí na střední souřadnicovou chybu

Pár	Vstupní snímek	DB snímek	Počet korespondencí	Střední souřadnicová chyba m_{xy} [m]
1	IMG_4898	IMG_4962	2710	0,09
	IMG_4962	IMG_4898	2470	0,07
3	IMG_4745	IMG_4743	2414	0,16
	IMG_4743	IMG_4745	2579	0,23
5	IMG_4909	IMG_4908	2532	0,20
	IMG_4908	IMG_4909	2868	0,74
7	IMG_4955	IMG_4954	3065	0,26
	IMG_4954	IMG_4955	2762	0,21
10	IMG_5073	IMG_4908	1637	0,10
	IMG_4908	IMG_5073	1739	0,38

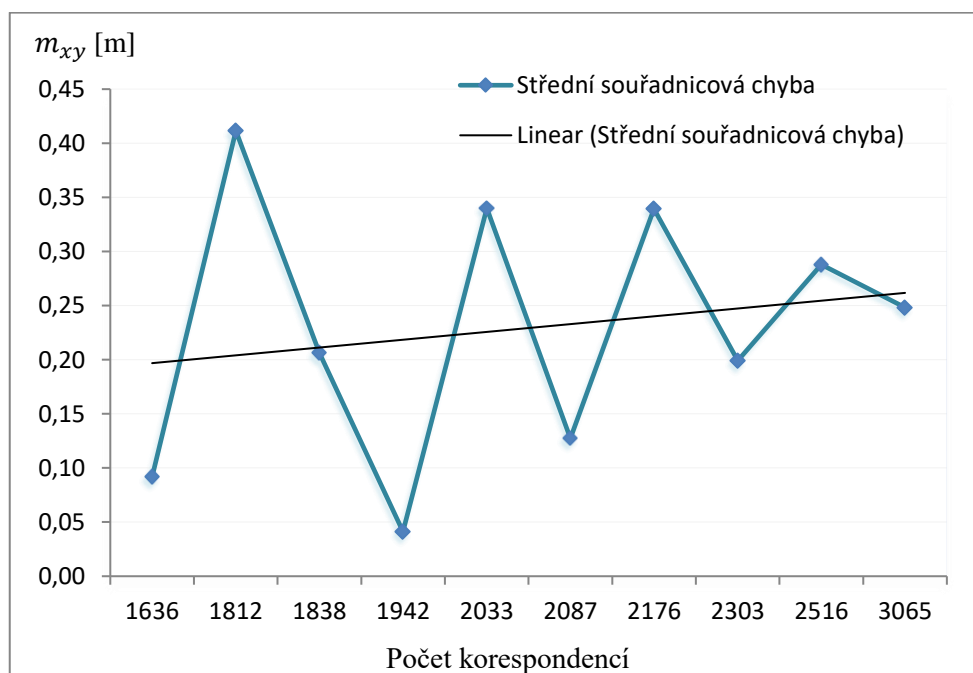
Skutečnost, že vyšší počet vyhledaných korespondencí nemusí vždy zaručit vyšší přesnost v určení polohy, vedla k provedení dalšího testu navrženého řešení. Ten spočívá v určení polohy vstupního snímku vůči snímkům zachycující tutéž scénu při klesajícím počtu snímkových korespondencí. Tento test byl proveden pro snímek IMG_4955 zobrazený na obrázku níže a získané výsledky jsou uvedeny v Tab. 5.



Obr. 28 Ukázka snímku IMG_4955

Tab. 5 Výsledky testování snímku IMG_4955

Vstupní snímek IMG_4955											
DB snímek	Počet korespondencí	Poloha projekčního centra DB snímku			Poloha projekčního centra vstupního snímku			Poloha uživatele			Střední souřadnicová chyba m_{xy} [m]
		X[m]	Y[m]	Z[m]	X [m]	Y [m]	Z [m]	X [m]	Y [m]	Z [m]	
4954	3065	12,64	9,66	1,59	12,50	9,68	1,65	12,83	9,80	1,58	0,25
4956	2516	12,23	8,44	1,69	12,50	9,68	1,65	12,33	10,05	1,70	0,29
4961	2303	15,16	8,39	1,62	12,50	9,68	1,65	12,75	9,55	1,66	0,20
4909	2176	9,76	9,20	1,61	12,50	9,68	1,65	12,11	9,40	1,55	0,34
4894	2087	16,71	8,12	1,66	12,50	9,68	1,65	12,60	9,53	1,66	0,13
4743	2033	15,55	10,34	1,65	12,50	9,68	1,65	12,98	9,71	1,71	0,34
4297	1942	16,15	8,72	1,64	12,50	9,68	1,65	12,53	9,73	1,78	0,04
4962	1838	16,91	8,44	1,60	12,50	9,68	1,65	12,75	9,53	1,69	0,21
4745	1812	17,30	8,57	1,60	12,50	9,68	1,65	13,05	9,49	1,60	0,41
4895	1636	16,71	8,14	1,62	12,50	9,68	1,65	12,63	9,68	1,66	0,09



Graf 3 Vliv počtu vyhledaných korespondencí na střední souřadnicovou chybu určení polohy

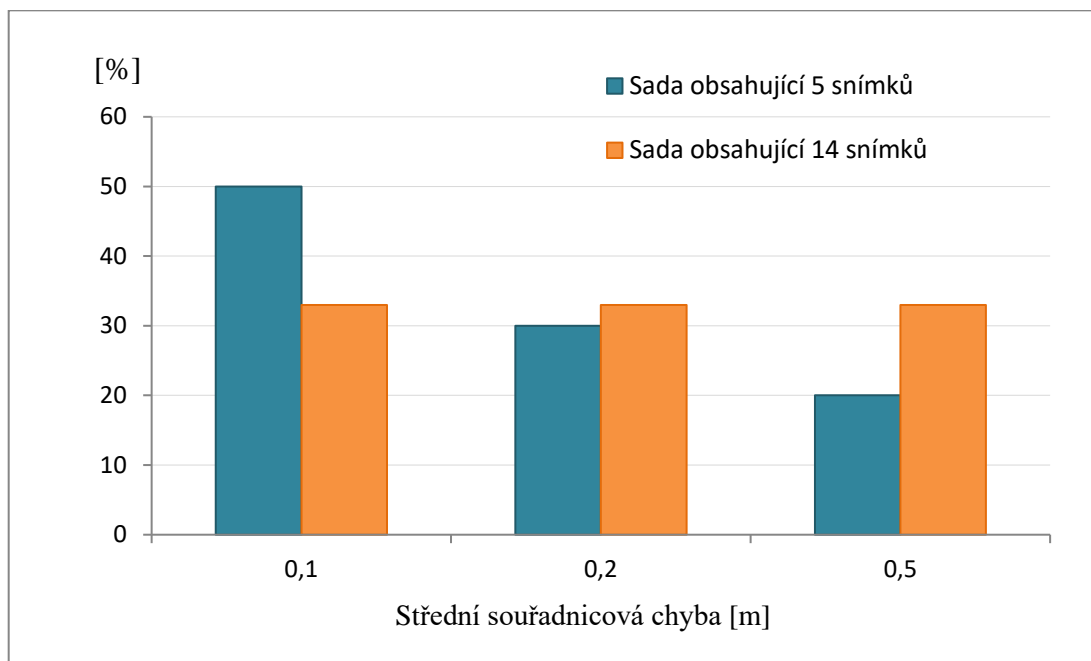
Z výše uvedených výsledků je patrné, že počet vyhledaných snímkových korespondencí nemá vliv na výslednou přesnost určení polohy, pokud se jedná o tutéž snímanou scénu. To potvrzuje skutečnost, že jsme obdrželi přesnost určení polohy 0,09 m pro snímek s nejnižším počtem vyhledaných snímkových korespondencí, což je ve srovnání se snímkem s nejvyšším počtem vyhledaných snímkových korespondencí lepší výsledek. Na základě tohoto zjištění byla databáze snímků scény č. 1 omezena na 5 snímků z původních 14 a znovu bylo provedeno testování vstupních snímků.

Tab. 6 Výsledky testování vstupních snímků oproti sadě 5 databázových snímků

Vstupní snímek	DB snímek	Počet korespondencí	Poloha projekčního centra vstupního snímku			Poloha uživatele			Střední souřadnicová chyba m_{xy} [m]
			X [m]	Y [m]	Z [m]	X[m]	Y[m]	Z[m]	
IMG_4908	IMG_4956	2486	9,75	9,30	1,68	9,75	9,20	1,71	0,07
IMG_4961	IMG_4297	2761	15,17	8,38	1,62	15,22	8,33	1,63	0,05
IMG_5074	IMG_4743	2674	15,83	9,23	1,58	15,47	9,58	1,63	0,36
IMG_5082	IMG_4956	1490	9,73	9,23	1,59	9,69	9,10	1,67	0,10
IMG_5071	IMG_4956	1790	11,92	8,06	1,82	12,41	7,64	1,63	0,46
IMG_5073	IMG_4908	1637	9,85	9,21	1,60	9,71	9,21	1,60	0,10
IMG_4898	IMG_4297	2638	18,42	9,76	1,67	18,25	9,62	1,64	0,16
IMG_4962	IMG_4297	2889	16,90	8,44	1,60	16,90	8,61	1,56	0,12
IMG_4954	IMG_4958	1769	12,65	9,66	1,60	12,76	9,91	1,57	0,19
IMG_4745	IMG_4297	2688	17,30	8,57	1,60	17,27	8,58	1,64	0,02
IMG_4895	IMG_4297	2244	16,72	8,14	1,62	16,73	8,01	1,64	0,09

Výsledky testu ukazují velice dobré výsledky, protože poloha projekčního centra vstupního snímku byla u více než 50% testovaných snímků určena se střední souřadnicovou chybou do 0,10 m, u téměř 30% snímků se střední souřadnicovou chybou do 0,20 m a u zbylých 20% snímků se střední souřadnicovou chybou do 0,50 m. Tento test navíc prokázal, že menší počet snímků zachycující jednu scénu vede k lepším výsledkům, než v případě testování oproti sadě 14 snímků, kdy jsme u 33% testovaných snímků získali chybu v určení polohy do 0,10 m, u 33% snímků chybu v určení polohy do 0,20 m a u zbylých 33% snímků pak chyba v určení polohy nepřesahovala 0,50 m.

Navíc z těchto výsledků vyplývá velice zásadní poznatek. Čím nižší bude počet uložených snímků v databázi, které popisují stejnou scénu interiéru, tím vyšší je pravděpodobnost přesného určení polohy. Jinými slovy se při nízkém počtu uložených snímků v databázi snižuje pravděpodobnost, že dojde k přiřazení snímku s podobnou polohou projekčního centra kamery, kterou byl pořízen snímek z databáze. Současně je ale potřeba říci, že počet snímků uložených v databázi nesmí podkročit limitní hranici, kdy by určení polohy bylo pro nedostatek snímků popisující charakter interiéru nemožné.



Graf 4 Procentuální vyjádření získaných přesností určení polohy pro dvě sady s rozdílným počtem snímků

Diskuse

V předešlých kapitolách bylo představeno navržené řešení určení polohy uživatele v interiéru budovy s využitím kamery mobilního telefonu a poté následovalo zhodnocení navrženého řešení. Navržené řešení však má svá omezení, která je potřeba v této kapitole zmínit.

Prvním z nich je stanovení přesnosti určené polohy. Abychom provedli ověření konceptu navrženého řešení, bylo provedeno hodnocení přesnosti na základě porovnání odhadnuté polohy vůči poloze, která byla získána na základě znalosti PnP problému, který lze vyřešit prostřednictvím 2D – 3D korespondencí vlíčovacích bodů. Při reálném využití by však takovéto stanovení přesnosti nebylo možné, neboť by navržené řešení ztratilo základní vlastnost, kterou je automatizovanost. Pokud by chtěl uživatel ověřit přesnost odhadnuté polohy, možným řešením by mohlo být pořízení více snímků z jednoho místa na více stran a následné porovnání získaných poloh.

Dále byl při pořizování vstupních testovacích snímků využit stejný mobilní telefon, kterým byly pořízeny snímky do databáze. Z toho vyplývá, že jsme již měli k dispozici vnitřní kalibrační parametry kamery tohoto mobilního telefonu zjištěné pomocí software Agisoft Lens. V reálném nasazení lze očekávat, že se vnitřní kalibrační parametry kamer mobilních telefonů liší, a proto by musel uživatel po instalaci hypotetické aplikace zjistit nejprve kalibrační parametry kamery svého mobilního telefonu. Knihovna OpenCV nabízí pro tento účel funkci `calibrateCamera()`, která po nasnímání kalibračního pole a vložení objektových a snímkových souřadnic vrátí vnitřní a vnější kalibrační parametry kamery. Znamenalo by to však, že by se toto kalibrační pole muselo nacházet v interiéru dané budovy.

Dále bylo zjištěno, že nejvíce snímkových korespondencí je SIFT algoritmem vyhledáváno v místech, kdy se ve snímané scéně nachází pravidelné vzory, obrazy nebo nápisy. Ačkoli byla provedeným testem stejné scény prokázána neexistence přímé úměry mezi počtem vyhledaných snímkových korespondencí a střední souřadnicovou chybou v určení polohy, mohl by nastat případ, kdy bude snímaná scéna z pohledu zpracování obrazu nezajímavá (např. bílé zdi). Tato scéna by proto neobsahovala dostatečně kontrastní významné body ve snímku, což by přineslo velice nízký nebo dokonce žádný počet vyhledaných korespondencí a určení polohy by tak nebylo umožněno. Z tohoto důvodu by bylo vhodné doplnit takto nekонтastní prostředí o obrazy nebo textová pole stejně jako bylo zmíněno v článku autorů Sadehgi et al. (2015).

Velkým úskalím tohoto řešení je výpočetní náročnost SIFT algoritmu. Doba výpočtu hledání významných prvků v jednom snímku trvá přibližně 3 minuty při využití hardware s následujícími parametry – 1.8 GHz CPU, 2 GB GPU, 6 GB RAM. Při vyhledávání nejvíce korespondujícího snímku z databáze o sadě 10 snímků byla zaznamenána doba výpočtu 30 minut. Přestože bylo v průběhu testování navrženého řešení zjištěno, že pro určení relativně přesné polohy uživatele není potřeba příliš velká sada snímků uložených v databázi, reálné využití by z hlediska takovéto výpočetní náročnosti na hardware nebylo možné ani s nízkým počtem snímků v databázi. Výpočetní náročnost navrženého řešení by mohla být snížena tím, že by byly významné prvky ve snímcích uložených v databázi vyhledány předem a jejich snímkové souřadnice by se staly součástí vytvořené databáze. K vyhledávání významných prvků prostřednictvím SIFT algoritmu by tak v průběhu určování polohy docházelo pouze u uživatelem pořízeného snímku. Dalším řešením snížení výpočetní náročnosti navrženého řešení by mohla být hierarchizace snímků. To znamená, že by byly jednotlivé databázové snímky hodnoceny v závislosti na četnosti jejich výběru z databáze. Snímky s nejvyšším počtem hodnocení by následně byly při výběru nejlepšího snímku z databáze procházeny jako první. Navíc, v reálném nasazení by se jednalo o klient-server architekturu, která odděluje klienta (aplikaci s grafickým uživatelským rozhraním) a server, kteří spolu komunikují přes počítačovou síť. Požadavek na určení polohy by byl tedy zasílán prostřednictvím aplikace na webový server vybavený výkonnějším hardware.

Co se týče vzhledu interiéru budovy, je dále potřeba zmínit, že s postupem času může docházet k jeho změnám. Některé vybavení může být přesunuto, odstraněno nebo naopak přidáno. Takovéto změny prostředí by nepochybně ovlivnily vyhledávání korespondencí mezi snímky z databáze a vstupním snímkem. V důsledku toho by tak docházelo k nepřesnému nebo dokonce nemožnému určení polohy. Databáze snímků popisující interiér budovy by proto při reálném použití musela být aktualizována.

Závěr

Cílem této diplomové práce byl návrh postupu určení polohy mobilního telefonu v interiéru budovy. Před tím, než bylo přistoupeno k samotnému návrhu postupu určení polohy mobilního telefonu v interiéru budovy, byly nejprve prozkoumány obecné přístupy k navigaci. V současné době existuje mnoho technik umožňující určení polohy uživatele v interiéru budovy, které jsou děleny podle jejich přesnosti a nároků na infrastrukturu. První skupina systémů je založena na stávající infrastruktuře, kam řadíme GNSS, inerciální senzory a optické polohování. Druhá skupina systémů je charakteristická posílením stávající infrastruktury prostřednictvím pseudosatelitů, Wi-Fi, Bluetooth, RFID, senzorů infračerveného záření nebo UWB systémy. První kapitola této diplomové práce podrobně popisuje výše uvedené techniky polohování prostřednictvím rešerše, která je založena na průzkumu stávajících článků a publikací věnujících se této problematice. Kapitola je zakončena porovnáním stávajících technik polohování na základě přesnosti, kterou tyto techniky poskytují, a ceny jejich pořízení.

V souvislosti s návrhem vlastního řešení, které využívá pro určení polohy mobilní telefon, byla naše pozornost zaměřena na techniky, které pro určení polohy nevyužívají další infrastrukturu. GNSS a inerciální navigace samy o sobě neposkytují dobré výsledky, a proto bývají často doplňovány dalšími technikami určování polohy, čímž roste jejich cena a požadavky na hardware a software. Poslední technikou, která nevyužívá pro polohování v interiéru další infrastrukturu, je technika optického polohování. Tato technika využívá pro určení polohy kameru, kterou je v dnešní době vybaven každý mobilní telefon. Ve druhé kapitole byly popsány principy výpočtu polohy z obrazového záznamu a dále pak algoritmy počítačového vidění, které usnadňují řešení těchto úloh. Následovala kapitola, ve které byly kategorizovány způsoby výpočtu polohy z obrazového záznamu, a nakonec byla provedena rešerše publikací využívající pro určení polohy kameru mobilního telefonu.

Poslední kapitola byla věnována vlastnímu návrhu určení polohy mobilního telefonu v interiéru budovy. Navržené řešení kombinuje princip epipolární geometrie, jakožto geometrie dvou pohledů kamer s odlišnými projekčními centry, a PnP problém, který umožňuje výpočet polohy kamery z n 2D – 3D korespondencí vlíčovacích bodů ve snímku. Z průzkumu stávajících publikací, které se věnují využití kamery mobilního telefonu pro určení polohy, vyplývá, že takovýto přístup dosud nebyl použit. Pro návrh vlastního řešení byla využita knihovna OpenCV. Testování navrženého řešení bylo uskutečněno za ideálních

podmínek, tzn. bez pohybu osob ve snímané scéně, beze změny snímané scény a stejnou kamerou mobilního telefonu, kterou byly pořízeny snímky do databáze. Celkem byly provedeny dva testy, při kterých byly pořízené vstupní snímky testovány oproti dvěma sadám o rozdílném počtu snímků uložených v databázi. Střední souřadnicové chyby určení polohy nepřesahovaly u obou testů hodnotu vyšší než je 0,50 m. Přesto lze ale konstatovat, že lepších výsledků bylo dosaženo v případě, kdy sada snímků uložených v databázi obsahovala nižší počet snímků.

Navržené řešení má svá omezení, která dávají prostor k dalšímu rozšíření předložené diplomové práce. Mezi tato omezení patří stanovení přesnosti určené polohy při reálném použití, kalibrace kamery mobilního telefonu před zahájením polohování a snížení výpočetní náročnosti navrženého řešení.

Zdroje

- Agisoft LLC. (2018) Agisoft Lens. Version 0.4.1. Petrohrad. Dostupné z: <https://download.cnet.com/Agisoft-Lens-64-bit/3000-6677_4-75573964.html>
- Agisoft LLC. (2011b) Agisoft Lens User Manual: Version 0.4.1. Petrohrad. Dostupné z: <<http://downloads.agisoft.ru/lens/doc/en/lens.pdf>>
- Al Nuaimi, K., & Kamel, H. (2011, April). A survey of indoor positioning systems and algorithms. In *2011 international conference on innovations in information technology* (pp. 185-190). IEEE. Dostupné z: <<https://ieeexplore.ieee.org/abstract/document/5893813/>>
- Alarifi, A., Al-Salman, A., Alsaleh, M., Alnafessah, A., Al-Hadhrami, S., Al-Ammar, M., & Al-Khalifa, H. (2016). Ultra wideband indoor positioning technologies: Analysis and recent advances. *Sensors*, *16*(5), 707. Dostupné z: <<https://www.mdpi.com/1424-8220/16/5/707>>
- Anaconda, Inc. (2019) Anaconda Python/R Distribution Open Source. Python 3.7 Version. Dostupné z: <<https://www.anaconda.com/distribution/>>
- Armangué, X., & Salvi, J. (2003). Overall view regarding fundamental matrix estimation. *Image and vision computing*, *21*(2), 205-220. Dostupné z: <<https://www.sciencedirect.com/science/article/pii/S0262885602001543>>
- Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-up robust features (SURF). *Computer vision and image understanding*, *110*(3), 346-359. Dostupné z: <<https://www.sciencedirect.com/science/article/pii/S1077314207001555>>
- Bekkali, A., Sanson, H., & Matsumoto, M. (2007, October). RFID indoor positioning based on probabilistic RFID map and Kalman filtering. In *Third IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2007)* (pp.21-21).IEEE. Dostupné z: <<https://ieeexplore.ieee.org/abstract/document/4390815>>
- Bianchi, G. (2000). Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on selected areas in communications*, *18*(3), 535-547. Dostupné z: <<https://ieeexplore.ieee.org/abstract/document/840210>>
- Bluetooth SIG, Inc. (2019). Enhancing Bluetooth Location Services with Direction Finding. Dostupné z: <https://3pl46c46ctx02p7rzdsvsg21-wpengine.netdna-ssl.com/wp-content/uploads/2019/03/1901_Enhancing-Bluetooth-Location-Service_FINAL.pdf>
- Calonder, M., Lepetit, V., Strecha, C., & Fua, P. (2010, September). Brief: Binary robust independent elementary features. In *European conference on computer vision* (pp. 778-792). Springer, Berlin, Heidelberg. Dostupné z: <https://link.springer.com/chapter/10.1007/978-3-642-15561-1_56>
- Civera, J., Grasa, O. G., Davison, A. J., & Montiel, J. M. M. (2010). 1-Point RANSAC for extended Kalman filtering: Application to real-time structure from motion and visual odometry. *Journal of Field Robotics*, *27*(5), 609-631. Dostupné z: <<https://onlinelibrary.wiley.com/doi/full/10.1002/rob.20345>>

Connell, C. (2015). What's the difference between measuring location by UWB Wi-Fi and bluetooth?. *Electron. Des. Mag., Penton Media, New York City, NY, USA, Tech. Rep.* Dostupné z: <<https://www.electronicdesign.com/datasheet/what-s-difference-between-measuring-location-uwbi-fi-and-bluetooth-pdf-download>>

Deretey, E., Ahmed, M. T., Marshall, J. A., & Greenspan, M. (2015, October). Visual indoor positioning with a single camera using PnP. In *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)* (pp. 1-9). IEEE. Dostupné z: <<https://ieeexplore.ieee.org/abstract/document/7346756>>

Derpanis, K. G. (2010). Overview of the RANSAC Algorithm. *Image Rochester NY*, 4(1), 2-3. Dostupné z: <<http://rmozone.com/snapshots/2015/07/cdg-room-refs/ransac.pdf>>

Edge, L., & Jobs, G. (2001). Centimeter-accuracy indoor navigation using GPS-like pseudolites. Dostupné z: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.365.872&rep=rep1&type=pdf>>

Eisa, S., Peixoto, J., Meneses, F., & Moreira, A. (2013, October). Removing useless APs and fingerprints from WiFi indoor positioning radio maps. In *International Conference on Indoor Positioning and Indoor Navigation* (pp. 1-7). IEEE. Dostupné z: <<https://ieeexplore.ieee.org/abstract/document/6817919>>

Evennou, F., & Marx, F. (2006). Advanced integration of WiFi and inertial navigation systems for indoor mobile positioning. *Eurasip journal on applied signal processing*, 2006, 164-164. Dostupné z: <<https://dl.acm.org/citation.cfm?id=1288407>>

Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 381-395. Dostupné z: <<https://dl.acm.org/citation.cfm?id=358692>>

Gao, X. S., Hou, X. R., Tang, J., & Cheng, H. F. (2003). Complete solution classification for the perspective-three-point problem. *IEEE transactions on pattern analysis and machine intelligence*, 25(8), 930-943. Dostupné z: <<https://ieeexplore.ieee.org/abstract/document/1217599>>

Grauman, K., & Leibe, B. (2011). Visual object recognition. *Synthesis lectures on artificial intelligence and machine learning*, 5(2), 1-181. Dostupné z: <<https://www.morganclaypool.com/doi/abs/10.2200/S00332ED1V01Y201103AIM011>>

Gu, Y., Lo, A., & Niemegeers, I. (2009). A survey of indoor positioning systems for wireless personal networks. *IEEE Communications Surveys & Tutorials*, 11 (1), 2009. Dostupné z: <<https://www.csd.uoc.gr/~hy439/papers/gu2009pdf.pdf>>

Hartley, R. I., & Mundy, J. L. (1993, September). Relationship between photogrammetry and computer vision. In *Integrating photogrammetric techniques with scene analysis and machine vision* (Vol. 1944, pp. 92-106). International Society for Optics and Photonics. Dostupné z: <<https://www.spiedigitallibrary.org/conference-proceedings-of-spie/1944/0000/Relationship-between-photogrammetry-and-computer-vision/10.1117/12.155818.short>>

Hartley, R., & Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge university press. Dostupné z:

<[http://cvrs.whu.edu.cn/downloads/ebooks/Multiple%20View%20Geometry%20in%20Computer%20Vision%20\(Second%20Edition\).pdf](http://cvrs.whu.edu.cn/downloads/ebooks/Multiple%20View%20Geometry%20in%20Computer%20Vision%20(Second%20Edition).pdf)>

Hile, H., & Borriello, G. (2008). Positioning and orientation in indoor environments using camera phones. *IEEE Computer Graphics and Applications*, 28(4), 32-39. Dostupné z: <<https://ieeexplore.ieee.org/abstract/document/4557953>>

Chawathe, S. S. (2009, October). Low-latency indoor localization using bluetooth beacons. In *2009 12th International IEEE Conference on Intelligent Transportation Systems* (pp. 1-7). IEEE. Dostupné z: <<https://ieeexplore.ieee.org/abstract/document/5309711>>

Jupyter Notebook (2019). Project Jupyter. Dostupné z: < <https://jupyter.org/install>

Karami, E., Prasad, S., & Shehata, M. (2017). Image matching using SIFT, SURF, BRIEF and ORB: performance comparison for distorted images. *arXiv preprint arXiv:1710.02726*. Dostupné z: < <https://arxiv.org/abs/1710.02726>>

Kawaji, H., Hatada, K., Yamasaki, T., & Aizawa, K. (2010, October). Image-based indoor positioning system: fast image matching using omnidirectional panoramic images. In *Proceedings of the 1st ACM international workshop on Multimodal pervasive video analysis* (pp. 1-4). ACM. Dostupné z: <<https://dl.acm.org/citation.cfm?id=1878041>>

Koyuncu, H., & Yang, S. H. (2010). A survey of indoor positioning and object locating systems. *IJCSNS International Journal of Computer Science and Network Security*, 10(5), 121-128. Dostupné z: <https://s3.amazonaws.com/academia.edu.documents/33814983/A_Survey_of_Indoor_Positioning_and_Object_Locating_Systems.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1557403469&Signature=tbo92VNMtPYo%2BsTpvkPCyjdutUU%3D&response-content-disposition=inline%3B>

Lepetit, V., Moreno-Noguer, F., & Fua, P. (2009). Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81(2), 155. Dostupné z: <<https://link.springer.com/article/10.1007/s11263-008-0152-6>>

Liang, J. Z., Corso, N., Turner, E., & Zakhor, A. (2015). Image-based positioning of mobile devices in indoor environments. In *Multimodal Location Estimation of Videos and Images* (pp. 85-99). Springer, Cham. Dostupné z: <https://link.springer.com/chapter/10.1007/978-3-319-09861-6_5>

Lowe, D. G. (1999, September). Object recognition from local scale-invariant features. In *iccv* (p. 1150). Ieee. Dostupné z: <<http://new.csd.uwo.ca/Courses/CS9840a/PossibleStudentPapers/iccv99.pdf>>

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2), 91-110. Dostupné z: <<https://link.springer.com/article/10.1023/B:VISI.0000029664.99615.94>>

- Mautz, R., & Tilch, S. (2011, September). Survey of optical indoor positioning systems. In *2011 international conference on indoor positioning and indoor navigation* (pp. 1-7). IEEE. Dostupné z: <<https://ieeexplore.ieee.org/abstract/document/6071925>>
- Medina, C., Segura, J., & De la Torre, A. (2013). Ultrasound indoor positioning system based on a low-power wireless sensor network providing sub-centimeter accuracy. *Sensors*, *13*(3), 3501-3526. Dostupné z: <<https://www.mdpi.com/1424-8220/13/3/3501>>
- Mian, A. S., Bennamoun, M., & Owens, R. (2008). Keypoint detection and local feature matching for textured 3D face recognition. *International Journal of Computer Vision*, *79*(1), 1-12. Dostupné z: <<https://link.springer.com/article/10.1007/s11263-007-0085-5>>
- Muja, M., & Lowe, D. G. (2014). Scalable nearest neighbor algorithms for high dimensional data. *IEEE transactions on pattern analysis and machine intelligence*, *36*(11), 2227-2240. Dostupné z: <<https://ieeexplore.ieee.org/abstract/document/6809191>>
- Namiot, D. (2015). On indoor positioning. *International Journal of Open Information Technologies*, *3*(3), 23-26. Dostupné z: <<http://injoit.ru/index.php/j1/article/view/183/140>>
- Niedermeier, H., Eissfeller, B., Winkel, J., Pany, T., Riedl, B., Wörz, T., ... & Jimenez-Banos, D. (2010, September). DINGPOS: High sensitivity GNSS platform for deep indoor scenarios. In *2010 International Conference on Indoor Positioning and Indoor Navigation* (pp. 1-10). IEEE. Dostupné z: <<https://ieeexplore.ieee.org/abstract/document/5647542>>
- Obr, V., & Koska, B. Dvoukroková metoda kalibrace digitální kamery s využitím nelineárních transformací. Dostupné z: <http://k154.fsv.cvut.cz/~koska/publikace/soubory/metody_odstraneni_distorzi.pdf>
- OpenCV Library (2018). Version 3.4.2. Santa Clara: Intel Corporation, Willow Garage, Itseez, Dostupné z: <<https://opencv.org/releases/page/2/>>
- OpenCV Documentation (2014) Version 2.4.13.7. Opencv dev team. Dostupné z: <<https://docs.opencv.org/2.4.13.7/>>
- Ozdenizci, B., Coskun, V., & Ok, K. (2015). NFC internal: An indoor navigation system. *Sensors*, *15*(4), 7571-7595. Dostupné z: <<https://www.mdpi.com/1424-8220/15/4/7571>>
- Özyeşil, O., Voroninski, V., Basri, R., & Singer, A. (2017). A survey of structure from motion*. *Acta Numerica*, *26*, 305-364. Dostupné z: <<https://www.cambridge.org/core/journals/acta-numerica/article/survey-of-structure-from-motion/C4B2E7BB10BC2C11AF71BC80B584D378>>
- Pavelka, Karel. Fotogrammetrie. 1. vyd. Plzeň: Západočeská univerzita, 2003, 247 s. ISBN 80-7082-972-9.
- Penate-Sanchez, A., Andrade-Cetto, J., & Moreno-Noguer, F. (2013). Exhaustive linearization for robust camera pose and focal length estimation. *IEEE transactions on pattern analysis and machine intelligence*, *35*(10), 2387-2400. Dostupné z: <<https://ieeexplore.ieee.org/abstract/document/6425380>>

- Rapinski, J., Cellmer, S., & Rzepecka, Z. (2012). Modified GPS/pseudolite navigation message. *The journal of navigation*, 65(4), 711-716. Dostupné z: <<https://www.cambridge.org/core/journals/journal-of-navigation/article/modified-gpspseudolite-navigation-message/9D5193030D5D6AD9B0ACA8094F6E00F3>>
- Ravi, N., Shankar, P., Frankel, A., Elgammal, A., & Iftode, L. (2005, August). Indoor localization using camera phones. In *Seventh IEEE Workshop on Mobile Computing Systems & Applications (WMCSA'06 Supplement)* (pp. 1-7). IEEE. Dostupné z: <<https://ieeexplore.ieee.org/abstract/document/4625206>>
- Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. Dostupné z: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.370.4395&rep=rep1&type=pdf>>
- Rosten, E., Porter, R., & Drummond, T. (2010). Faster and better: A machine learning approach to corner detection. *IEEE transactions on pattern analysis and machine intelligence*, 32(1), 105-119. Dostupné z: <<https://ieeexplore.ieee.org/abstract/document/4674368>>
- Ruppel, P.; Gschwandtner, F. Spontaneous and Privacy-Friendly Mobile Indoor Routing and Navigation. In *Proceedings of the Second Workshop on Services, Platforms, Innovations and Research for New Infrastructures in Telecommunications*, Lübeck, Germany, 29 September–2 October 2009; pp. 2574–2583. Dostupné z: <<http://cs.emis.de/LNI/Proceedings/Proceedings154/gi-proc-154-225.pdf>>
- Saab, S. S., & Nakad, Z. S. (2011). A standalone RFID indoor positioning system using passive tags. *IEEE Transactions on Industrial Electronics*, 58(5), 1961-1970. Dostupné z: <<https://ieeexplore.ieee.org/abstract/document/5504205>>
- Sadeghi, H., Valaee, S., & Shirani, S. (2015, April). Ocrapose: An indoor positioning system using smartphone/tablet cameras and OCR-aided stereo feature matching. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 1473-1477). IEEE. Dostupné z: <<https://ieeexplore.ieee.org/abstract/document/7178215>>
- Schindler, K. (2014). Mathematical Foundations of Photogrammetry. *Handbook of Geomatics*, 1-14. Dostupné z: <https://link.springer.com/content/pdf/10.1007/978-3-642-27793-1_63-2.pdf>
- Schon, S., & Bielenberg, O. (2008, March). On the capability of high sensitivity GPS for precise indoor positioning. In *2008 5th Workshop on Positioning, Navigation and Communication* (pp. 121-127). IEEE. Dostupné z: <<https://ieeexplore.ieee.org/abstract/document/4510365>>
- Sonka, M., Hlavac, V. & Boyle, R., (2007) *Image Processing, Analysis and Machine Vision*. Thomson, 3rd edition, ISBN 978-0-495-08252
- Torr, P. H., & Zisserman, A. (2000). MLESAC: A new robust estimator with application to estimating image geometry. *Computer vision and image understanding*, 78(1), 138-156. Dostupné z: <<https://www.sciencedirect.com/science/article/pii/S1077314299908329>>

Van Opendenbosch, D., Schroth, G., Huitl, R., Hilsenbeck, S., Garcea, A., & Steinbach, E. (2014, October). Camera-based indoor positioning using scalable streaming of compressed binary image signatures. In *2014 IEEE International Conference on Image Processing (ICIP)* (pp. 2804-2808). IEEE. Dostupné z: <<https://ieeexplore.ieee.org/abstract/document/7025567>>

VÚGTK (2019). *Terminologický slovník zeměměřičství a katastru nemovitostí* [online], 2005-2019 Dostupné z: <<https://www.vugtk.cz/slovník/hledej.php>>

Vyhláška č. 26/2007 Sb., kterou se provádí zákon č. 265/1992 Sb., o zápisech vlastnických a jiných věcných práv k nemovitostem, ve znění pozdějších předpisů, a zákon č. 344/1992 Sb., o katastru nemovitostí České republiky (katastrální zákon), ve znění pozdějších předpisů, (katastrální vyhláška)

Wang, Z., & Quan, Y. (2008, December). An improved method for feature point matching in 3d reconstruction. In *2008 International Symposium on Information Science and Engineering* (Vol. 1, pp. 159-162). IEEE. Dostupné z: <<https://ieeexplore.ieee.org/abstract/document/4732191>>

Wang, Y., Zhang, Q., & Zhou, Y. (2014, June). RGB-D mapping for indoor environment. In *2014 9th IEEE Conference on Industrial Electronics and Applications* (pp. 1888-1892). IEEE. Dostupné z: <<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6931476>>

Werner, M., Kessel, M., & Marouane, C. (2011, September). Indoor positioning using smartphone camera. In *2011 International Conference on Indoor Positioning and Indoor Navigation* (pp. 1-6). IEEE. Dostupné z: <<https://ieeexplore.ieee.org/abstract/document/6071954>>

Weston, J. L., & Titterton, D. H. (2000). Modern inertial navigation technology and its application. *Electronics & Communication Engineering Journal*, 12(2), 49-64. Dostupné z: <<https://ieeexplore.ieee.org/abstract/document/844485>>

Woodman, O. J. (2007). An introduction to inertial navigation. University of Cambridge. *Computer Laboratory, Tech. Rep. UCAMCL-TR-696*, 14(2007), 15. Dostupné z: <<https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-696.html>>

Wu, Y., Tang, F., & Li, H. (2018). Image-based camera localization: an overview. *Visual Computing for Industry, Biomedicine, and Art*, 1(1), 8. Dostupné z: <<https://vciba.springeropen.com/articles/10.1186/s42492-018-0008-z>>

Xiao, J., Liu, Z., Yang, Y., Liu, D., & Han, X. (2011, June). Comparison and analysis of indoor wireless positioning techniques. In *2011 International conference on computer science and service system (CSSS)* (pp. 293-296). IEEE. Dostupné z: <<https://ieeexplore.ieee.org/abstract/document/5972088>>

Xu, R., Chen, W., Xu, Y., & Ji, S. (2015). A new indoor positioning system architecture using GPS signals. *Sensors*, 15(5), 10074-10087. Dostupné z: <<https://www.mdpi.com/1424-8220/15/5/10074>>

Yi, T. H., Li, H. N., & Gu, M. (2012). Effect of different construction materials on propagation of GPS monitoring signals. *Measurement*, 45(5), 1126-1139. Dostupné z: <<https://www.sciencedirect.com/science/article/pii/S0263224112000425>>

Zhang, C. X., & Hu, Z. Y. (2005). A general sufficient condition of four positive solutions of the P3P problem. *Journal of Computer Science and Technology*, 20(6), 836-842. Dostupné z: <<https://link.springer.com/article/10.1007/s11390-005-0836-0>>

Zhuang, Y., Li, Y., Lan, H., Syed, Z., & El-Sheimy, N. (2015). Smartphone-based WiFi access point localisation and propagation parameter estimation using crowdsourcing. *Electronics Letters*, 51(17), 1380-1382. Dostupné z: <<https://digital-library.theiet.org/content/journals/10.1049/el.2015.1724>>

Přílohy

Zdrojový kód

```
import cv2
import glob
import numpy as np
import xml.etree.ElementTree as ET
from matplotlib import pyplot as plt

# -----
#                               definování funkcí
# -----
# -----funkce parsuje XML a vrací snímkové a objektové souřadnice-----
# -----

def getCoordsFromXML(path):
    imageCoords = []
    worldCoords = []

    #parsování xml
    xmlPath = path.lower().replace('.jpg', '.xml').replace('.jpeg',
'.xml');
    xmlTree = ET.parse(xmlPath)
    xmlRoot = xmlTree.getroot().findall('keypoint')

    #iterace vlíčovacími body v xml
    a = 0
    for child in xmlRoot:
        xmlX = int(child.find('image_coor_x').text)
        xmlY = int(child.find('image_coor_y').text)
        imageCoords.append([xmlX,xmlY])
        xmlWorldX = float(child.find('world_coor_x').text)
        xmlWorldY = float(child.find('world_coor_y').text)
        xmlWorldZ = float(child.find('world_coor_z').text)
        worldCoords.append([xmlWorldX,xmlWorldY,xmlWorldZ])
    return (imageCoords, worldCoords)

def getGroupedCoordsFromXML(path):
    resultXZImagePairs = []
    resultYZImagePairs = []
    resultXZWorldPairs = []
    resultYZWorldPairs = []
    resultXZ = []
    resultYZ = []
    #parsování xml
    xmlPath = path.lower().replace('.jpg', '.xml').replace('.jpeg',
'.xml');
    xmlTree = ET.parse(xmlPath)
    xmlRoot = xmlTree.getroot().findall('keypoint')
    #iterace vlíčovacími body v xml
    for child in xmlRoot:
        xmlX_child = int(child.find('image_coor_x').text)
```

```

xmlY_child = int(child.find('image_coor_y').text)
xmlWorldX_child = float(child.find('world_coor_x').text)
xmlWorldY_child = float(child.find('world_coor_y').text)
xmlWorldZ_child = float(child.find('world_coor_z').text)
for child2 in xmlRoot:
    xmlX_child2 = int(child2.find('image_coor_x').text)
    xmlY_child2 = int(child2.find('image_coor_y').text)
    xmlWorldX_child2 = float(child2.find('world_coor_x').text)
    xmlWorldY_child2 = float(child2.find('world_coor_y').text)
    xmlWorldZ_child2 = float(child2.find('world_coor_z').text)
    distY = xmlWorldY_child2 - xmlWorldY_child
    distX = xmlWorldX_child2 - xmlWorldX_child
    if child.find('id')!=child2.find('id'):
        if xmlWorldX_child == xmlWorldX_child2 and
xmlWorldZ_child == xmlWorldZ_child2 and xmlWorldZ_child != 0 and
xmlWorldZ_child2 != 0 and distY <= 5 and distY > 0:

resultXZ.append([[xmlX_child,xmlY_child],[xmlWorldX_child,xmlWorldY_chi
ld,xmlWorldZ_child]],[xmlX_child2,xmlY_child2],[xmlWorldX_child2,xmlWor
ldY_child2,xmlWorldZ_child2]])
        if len(resultXZImagePairs) < 1:

resultXZImagePairs.append([[xmlX_child,xmlY_child],[xmlX_child2,xmlY_chi
ld2]])

        if len(resultXZWorldPairs) < 1:

resultXZWorldPairs.append([[xmlWorldX_child,xmlWorldY_child,xmlWorldZ_ch
ild],[xmlWorldX_child2,xmlWorldY_child2,xmlWorldZ_child2]])
        if xmlWorldY_child == xmlWorldY_child2 and
xmlWorldZ_child == xmlWorldZ_child2 and xmlWorldZ_child != 0 and
xmlWorldZ_child2 != 0 and distX <= 5 and distX > 0:

resultYZ.append([[xmlX_child,xmlY_child],[xmlWorldX_child,xmlWorldY_chi
ld,xmlWorldZ_child]],[xmlX_child2,xmlY_child2],[xmlWorldX_child2,xmlWor
ldY_child2,xmlWorldZ_child2]])
        if len(resultYZImagePairs) < 1:

resultYZImagePairs.append([[xmlX_child,xmlY_child],[xmlX_child2,xmlY_chi
ld2]])

        if len(resultYZWorldPairs) < 1:

resultYZWorldPairs.append([[xmlWorldX_child,xmlWorldY_child,xmlWorldZ_ch
ild],[xmlWorldX_child2,xmlWorldY_child2,xmlWorldZ_child2]])

    return (resultXZ, resultYZ, resultXZImagePairs, resultXZWorldPairs,
resultYZImagePairs, resultYZWorldPairs)

# -----
# -----funkce vrací nejpodobnější snímek z DB-----
# -----
def getBestMatch(databaseImagePaths, inputImage):
    #[0] = path, [1] = matchesCount, [2] = pts1, [3] = pts2
    imagesInfo = []
    #iterace jednotlivými obrázky

```

```

for path in databaseImagePaths:

    imageInfo = [None]*4
    imageInfo[0] = path;
    databaseImage = cv2.imread(path,0)
    # vyhledání korespondencí mezi databázovým a vstupním snímkem
    sift = cv2.xfeatures2d.SIFT_create()
    # find the keypoints and descriptors with SIFT
    inputImageKeypoints, inputDes =
sift.detectAndCompute(inputImage,None)
    imageKeypoints, imgDes =
sift.detectAndCompute(databaseImage,None)
    # FLANN parameters
    FLANN_INDEX_KDTREE = 0
    index_params = dict(algorithm = FLANN_INDEX_KDTREE, trees = 5)
    search_params = dict(checks=50)
    flann = cv2.FlannBasedMatcher(index_params,search_params)
    matches = flann.knnMatch(inputDes, imgDes, k=2)
    good = []
    pts1 = []
    pts2 = []
    # ratio test as per Lowe's paper
    for i,(m,n) in enumerate(matches):
        if m.distance < 0.8*n.distance:
            good.append(m)
            pts1.append(imageKeypoints[m.trainIdx].pt)
            pts2.append(inputImageKeypoints[m.queryIdx].pt)
    imageInfo[1] = len(good);
    pts1 = np.array(pts1,dtype=np.float32)
    pts2 = np.array(pts2,dtype=np.float32)
    print(imageInfo[0])
    print(imageInfo[1])
    imageInfo[2] = pts1
    imageInfo[3] = pts2
    #append image info
    imagesInfo.append(imageInfo)
#zjištění obrázku s maximálním množstvím shod
maxGoodCount = 0
maxGoodCountPosition = 0
i = 0
for imageInfo in imagesInfo:
    if imageInfo[1] > maxGoodCount:
        maxGoodCount = imageInfo[1]
        maxGoodCountPosition = i
    i += 1
maxImageInfo = imagesInfo[maxGoodCountPosition]
return maxImageInfo

# -----
# -----funkce vrací transformované body pomocí homografie-----
# -----
def transformPoints(bestImageInfo):
    result = []
    coords = getGroupedCoordsFromXML(bestImageInfo[0])

```

```

    transformacni_matice = cv2.findHomography(bestImageInfo[2],
bestImageInfo[3], cv2.RANSAC,5.0)[0]
    for coord in coords[2][0]:
        P= np.array(coord,dtype=np.float32)
        P = np.array([[P]])
        transformace_bodu =
cv2.perspectiveTransform(P,transformacni_matice)
        transformace_bodu = np.array(transformace_bodu, dtype=np.int32)
        result.append(transformace_bodu)
    return result
# -----
# -----funkce vrací polohu kamery a směr optické osy-----
# -----
def getPosition(P):
    rotation = P[:3,:3]
    translation = P[:,3]
    position = np.dot(-(np.linalg.inv(rotation)), translation)
    decomposition_P = cv2.decomposeProjectionMatrix(P)
    rotation_inv = np.linalg.inv(decomposition_P[1])
    direction = np.dot(rotation_inv,[[0],[0],[1]])
    direction = direction.T
    return (position, direction)
# -----
# ----funkce vrací projekční a rotační matici z řešení PnP problému----
# -----
def getProjectionMatrix(PnP):
    rvec = PnP[1]
    tvec = PnP[2]
    rot = cv2.Rodrigues(rvec)[0]
    Q_t_in=np.concatenate((rot, tvec), axis=1)
    projection_matrix = np.dot(calibration_matrix,Q_t_in)

    return (projection_matrix, rot)
# -----
#                               Cesty ke snímkům z databáze
# -----
databaseImagePaths = glob.glob("C:\\Users\\hanka\\Documents\\Python
projects\\DP\\data\\*.jpg")
# načtení vstupního snímku
inputImagePath = 'C:\\Users\\hanka\\Documents\\Python
projects\\DP\\vstup\\IMG_4913.jpg'
img_in= cv2.imread(inputImagePath, 0)
# -----
#                               získání snímku s nejvyšší shodou
# -----
bestImageInfo = getBestMatch(databaseImagePaths,img_in)
img_best_db = cv2.imread(bestImageInfo[0],0)
img_best_path = bestImageInfo[0]
print(img_best_path)
print(bestImageInfo[1])
# -----
#                               definice proměnných
# -----
width = img_best_db.shape[1]

```

```

height = img_best_db.shape[0]
size = (width,height)
distCoeff = np.zeros((5,1),np.float64)
k1 = 0.06614;
k2 = -0.07107;
k3 = -0.10745;
p1 = -0.000100;
p2 = -0.000370;

distCoeff[0,0] = k1;
distCoeff[1,0] = k2;
distCoeff[2,0] = p1;
distCoeff[3,0] = p2;
distCoeff[4,0] = k3;
calibration_matrix = np.eye(3,dtype=np.float32)
calibration_matrix[0,2] = width/2.0 # define center x
calibration_matrix[1,2] = height/2.0 # define center y
calibration_matrix[0,0] = 3538.13 # define focal length x
calibration_matrix[1,1] = 3538.13 # define focal length y
W = [[0,-1,0],[1,0,0],[0,0,1]]
I = np.array([[1,0,0],[0,1,0],[0,0,1]])
C = np.array([0,0,0])[np.newaxis,:].T

# -----
# výpočet projekční matice DB snímku z 2D-3D souřadnic
# -----
db_best_coords = getCoordsFromXML(img_best_path)
db_img_points = db_best_coords[0]
db_obj_points = db_best_coords[1]
db_obj_points = np.array(db_obj_points,dtype=np.float32)
db_img_points = np.array(db_img_points,dtype=np.float32)
PnP_db = cv2.solvePnP(db_obj_points, db_img_points, calibration_matrix,
distCoeff)
projection_mat_db = getProjectionMatrix(PnP_db)[0]
print ('-----projekční matice C1-----')
print(projection_mat_db)
print ('-----poloha C1-----')
polohaC1 = getPosition(projection_mat_db)[0]
print(polohaC1)
print(getPosition(projection_mat_db)[1])
print ('-----rotační matice C1-----')
rot_db = getProjectionMatrix(PnP_db)[1]
print(rot_db)
# -----
# výpočet projekční matice vstupního snímku z 2D-3D souřadnic
# -----
input_coords = getCoordsFromXML(inputImagePath)
in_img_points = input_coords[0]
in_obj_points = input_coords[1]
in_obj_points = np.array(in_obj_points,dtype=np.float32)
in_img_points = np.array(in_img_points,dtype=np.float32)
PnP_in = cv2.solvePnP(in_obj_points, in_img_points, calibration_matrix,
distCoeff)
projection_mat_in = getProjectionMatrix(PnP_in)[0]

```



```

print ('-----projekční matice C2-----')
print(projection_mat_in)
print ('-----poloha C2 -----')
polohaC2 = getPosition(projection_mat_in)[0]
print(polohaC2)
print(getPosition(projection_mat_in)[1])
print ('-----rotační matice C2-----')
rot_in = getProjectionMatrix(PnP_in)[1]
print(rot_in)
# -----
#           tranlace mezi C1 a C2 získaná z projekčních matic kamer
# -----
translation_obj = np.subtract(polohaC2,polohaC1)
print ('-----translace mezi C1 a C2-----')
print(translation_obj)

# -----
#           výpočet fundamentální matice
# -----
F_est, mask =
cv2.findFundamentalMat(bestImageInfo[2],bestImageInfo[3],cv2.FM_RANSAC,3
,0.99)

# -----
#           výpočet esenciální matice a dekompozice
# -----
E =
np.dot(np.dot(np.transpose(calibration_matrix),F_est),calibration_matrix
)
points, R_est, t_est, mask = cv2.recoverPose(E,
bestImageInfo[2],bestImageInfo[3])
print ('-----rotace mezi C1 a C2 odhadovaná-----')
print(R_est)
print ('-----translace mezi C1 a C2 odhadovaná-----')
print(t_est)
# -----
#   translace odhadnutého smeroveho vektoru mezi C1 a C2 v souřadnicovém
#           systému C1
# -----
translation_obj_est = np.dot(np.linalg.inv(np.dot(R_est,rot_db)),t_est)
print ('-----translace mezi C1 a C2 odhadovaná v SS C1-----')
print(translation_obj_est)
# -----
#           výpočet měřítka triangulací
# -----
imageCoords_db = getGroupedCoordsFromXML(img_best_path)[2][0]
imageCoords_input = transformPoints(bestImageInfo)

objectCoords = np.array(getGroupedCoordsFromXML(img_best_path)[3][0],
dtype = np.float32)

proj_mat_db =
np.dot(calibration_matrix,np.dot(rot_db,np.concatenate((I,C),axis=1)))

```

```

proj_mat_in =
np.dot(np.dot(np.dot(calibration_matrix,R_est),rot_db),np.concatenate((I
,np.add(C,translation_obj_est)), axis=1))
triangulateP1_1 =
cv2.triangulatePoints(proj_mat_db,proj_mat_in,np.array(imageCoords_db[0]
,dtype=np.float32),np.array(imageCoords_input[0][0][0],dtype =
np.float32))
triangulateP1_2 =
cv2.triangulatePoints(proj_mat_db,proj_mat_in,np.array(imageCoords_db[1]
,dtype=np.float32),np.array(imageCoords_input[1][0][0],dtype =
np.float32))
triangulateP1_1 = np.divide(triangulateP1_1[:3],triangulateP1_1[3])
triangulateP1_2 = np.divide(triangulateP1_2[:3],triangulateP1_2[3])
imageObjectDistance = np.subtract(triangulateP1_1,triangulateP1_2)
imageObjectDistance = np.linalg.norm(imageObjectDistance)
worldObjectDistance = np.subtract(objectCoords[0],objectCoords[1])
worldObjectDistance = np.linalg.norm(worldObjectDistance)
scale = np.divide(worldObjectDistance, imageObjectDistance)
print('-----odhadnuté měřítko-----')
print(scale)

# -----
#                               odhad projekční matice P2 kamery C2
# -----
translace_obj_est_final = np.dot(scale,translation_obj_est)
C_plus_translation = np.add(polohaC1,translace_obj_est_final.T)
print ('-----poloha_estimated-----')
projekcni_matice_est =
np.dot(np.dot(np.dot(calibration_matrix,R_est),rot_db),np.concatenate((I
,-C_plus_translation.T), axis=1))
print(getPosition(projekcni_matice_est))
print(projekcni_matice_est)

```

Tabulky

Tab. 7 Podrobné výsledky testu určení polohy na základě navrženého řešení – sada 14 snímků uložených v databázi

Vstupní snímek	DB snímek	Počet korespondencí	Poloha projekčního centra DB snímku			Poloha projekčního centra vstupního snímku z PnP problému			Poloha uživatele			Souřadnicové rozdíly		Střední souřadnicová chyba m_{xy} [m]	Translace mezi C1 a C2			Odhadnutý směrový vektor translace			Odhadnuté měřítko	Délka základny [m]
			X [m]	Y [m]	Z [m]	X [m]	Y [m]	Z [m]	X [m]	Y [m]	Z [m]	Δx [m]	Δy [m]		X [m]	Y [m]	Z [m]	X [m]	Y [m]	Z [m]		
IMG_4898	IMG_4962	2710	16,90	8,44	1,59	18,41	9,75	1,67	18,31	9,68	1,63	0,10	0,07	0,09	1,510	1,310	0,080	0,750	0,661	0,011	1,874	2,00
IMG_4962	IMG_4898	2470	18,41	9,75	1,67	16,90	8,44	1,59	16,95	8,35	1,66	-0,05	0,09	0,07	-1,510	-1,310	-0,080	-0,722	-0,691	-0,019	2,02	
IMG_4297	IMG_4898	2621	18,41	9,75	1,67	16,14	8,72	1,64	16,21	8,92	1,77	-0,07	-0,20	0,15	-2,270	-1,030	-0,030	-0,934	-0,354	0,040	2,363	2,49
IMG_4898	IMG_4297	2638	16,14	8,72	1,64	18,41	9,75	1,67	18,25	9,62	1,64	0,16	0,13	0,15	2,270	1,030	0,030	0,919	0,394	-0,001	2,288	
IMG_4745	IMG_4743	2414	15,55	10,34	1,65	17,29	8,57	1,60	17,20	8,78	1,60	0,09	-0,21	0,16	1,740	-1,770	-0,050	0,726	-0,678	0,001	2,265	2,48
IMG_4743	IMG_4745	2579	17,29	8,57	1,60	15,55	10,34	1,65	15,82	10,16	1,56	-0,27	0,18	0,23	-1,740	1,770	0,050	-0,681	0,731	-0,016	2,175	
IMG_4954	IMG_4958	1769	12,40	8,42	1,63	12,64	9,66	1,59	12,76	9,90	1,57	-0,12	-0,24	0,19	0,240	1,240	-0,040	0,234	0,971	-0,020	1,525	1,26
IMG_4958	IMG_4954	1502	12,64	9,66	1,59	12,40	8,42	1,63	12,52	8,78	1,60	-0,12	-0,36	0,27	-0,240	-1,240	0,040	-0,135	-0,990	-0,002	0,884	
IMG_4909	IMG_4908	2532	9,75	9,30	1,62	9,76	9,20	1,61	9,49	9,10	1,64	0,27	0,10	0,20	0,010	-0,100	-0,010	-0,790	-0,605	0,089	0,328	0,10
IMG_4908	IMG_4909	2868	9,76	9,20	1,61	9,75	9,30	1,62	9,67	10,35	1,62	0,08	-1,05	0,74	-0,010	0,100	0,010	0,088	-0,996	-0,005	1,115	
IMG_4956	IMG_4955	2435	12,49	9,68	1,65	12,23	8,44	1,69	12,47	8,12	1,63	-0,24	0,32	0,28	-0,260	-1,240	0,040	-0,016	-0,999	-0,001	1,550	1,27
IMG_4955	IMG_4956	2516	12,23	8,44	1,69	12,49	9,68	1,65	12,33	10,05	1,70	0,16	-0,37	0,29	0,260	1,240	-0,040	0,052	0,998	0,013	1,609	
IMG_4955	IMG_4954	3065	12,64	9,66	1,59	12,49	9,67	1,64	12,83	9,80	1,58	-0,34	-0,13	0,26	-0,150	0,010	0,050	0,781	0,621	-0,058	0,238	0,15
IMG_4954	IMG_4955	2762	12,49	9,67	1,64	12,64	9,66	1,59	12,34	9,62	1,63	0,30	0,04	0,21	0,150	-0,010	-0,050	-0,938	-0,332	0,088	0,167	
IMG_4961	IMG_4962	2852	16,90	8,44	1,60	15,17	8,39	1,60	15,25	8,36	1,60	-0,08	0,03	0,06	-1,730	-0,050	0,000	-0,998	-0,049	0,005	1,653	1,73
IMG_4962	IMG_4961	2835	15,17	8,39	1,60	16,90	8,44	1,60	16,76	8,44	1,60	0,14	0,00	0,10	1,730	0,050	0,000	0,999	0,034	-0,012	1,597	
IMG_5071	IMG_4956	1790	12,25	8,44	1,68	11,92	8,06	1,82	12,41	7,64	1,63	-0,49	0,42	0,46	-0,330	-0,380	0,140	0,201	-0,997	0,061	0,821	0,50
IMG_4956	IMG_5071	1933	11,92	8,06	1,82	12,23	8,44	1,69	11,92	8,06	1,82	0,31	0,38	0,35	0,310	0,380	-0,130	0,514	-0,856	-0,045	0,001	
IMG_5072	IMG_4908	1637	9,75	9,30	1,62	9,85	9,21	1,60	9,71	9,21	1,60	0,14	0,00	0,10	0,100	-0,090	-0,020	-0,475	-0,860	-0,182	0,119	0,135
IMG_4908	IMG_5072	1739	9,85	9,21	1,60	9,75	9,30	1,62	9,86	8,78	1,60	-0,11	0,52	0,38	-0,100	0,090	0,020	0,009	-0,999	-0,003	0,431	
IMG_5074	IMG_4743	2674	15,55	10,34	1,65	15,82	9,23	1,58	15,47	9,58	1,63	0,35	-0,35	0,35	0,270	-1,110	-0,070	-0,107	-0,993	-0,301	0,758	1,14
IMG_4743	IMG_5074	2669	15,82	9,23	1,58	15,55	10,34	1,65	15,52	10,18	1,56	0,03	0,16	0,12	-0,270	1,110	0,070	-0,312	0,949	-0,016	1,003	
IMG_5082	IMG_4909	1577	9,76	9,20	1,61	9,72	9,22	1,59	9,81	9,21	1,60	-0,09	0,01	0,06	-0,040	0,020	-0,020	0,898	0,388	-0,205	0,028	0,05
IMG_4909	IMG_5082	1473	9,72	9,22	1,59	9,76	9,20	1,61	9,60	9,24	1,75	0,16	-0,04	0,12	0,040	-0,020	0,020	-0,573	0,094	0,813	0,208	
IMG_4740	IMG_4742	1156	7,62	8,16	1,59	8,67	8,70	1,62	8,79	9,38	1,63	-0,12	-0,68	0,49	1,050	0,540	0,030	0,693	0,720	-0,030	1,697	1,18
IMG_4742	IMG_4740	881	8,67	8,70	1,62	7,62	8,16	1,59	7,93	7,61	1,62	-0,31	0,55	0,45	-1,050	-0,540	-0,030	-0,555	-0,831	-0,002	1,304	
IMG_5084	IMG_4913	1338	8,43	8,08	1,65	8,57	7,71	1,5	8,61	7,66	1,50	-0,04	0,05	0,05	0,140	-0,370	-0,150	0,388	-0,869	-0,306	0,482	0,40
IMG_4913	IMG_5084	1186	8,57	7,71	1,5	8,43	8,08	1,65	8,49	8,32	1,63	-0,06	-0,24	0,17	-0,140	0,370	0,150	-0,124	0,972	0,194	0,629	
IMG_5086	IMG_4944	702	6,72	9,65	1,61	6,58	10,00	1,67	6,71	9,65	1,62	-0,13	0,35	0,26	-0,140	0,350	0,060	-0,935	0,348	-0,060	0,006	0,38
IMG_4944	IMG_5086	718	6,58	10,00	1,67	6,72	9,65	1,61	6,74	9,98	1,67	-0,02	-0,33	0,23	0,140	-0,350	-0,060	0,984	-0,174	0,011	0,167	
IMG_4914	IMG_4913	2332	8,42	8,08	1,64	8,45	8,01	1,66	8,38	8,15	1,65	0,07	-0,14	0,11	0,030	-0,070	0,020	-0,511	0,858	0,043	0,082	0,08
IMG_4913	IMG_4914	2101	8,45	8,01	1,66	8,42	8,08	1,64	8,36	7,31	1,72	0,06	0,77	0,55	-0,030	0,070	-0,020	-0,123	-0,943	0,127	0,729	
IMG_4944	IMG_4913	893	8,42	8,08	1,64	6,72	9,65	1,61	6,90	9,17	1,59	-0,18	0,48	0,36	-1,700	1,570	-0,030	-0,812	0,582	-0,033	1,869	2,31
IMG_4913	IMG_4944	1030	6,72	9,65	1,61	8,42	8,08	1,64	8,06	8,56	1,64	0,36	-0,48	0,42	1,700	-1,570	0,030	0,778	-0,627	0,012	1,732	

Tab. 8 Podrobné výsledky testování vstupních snímků oproti sadě 5 databázových snímků

Vstupní snímek	DB snímek	Počet korespondencí	Poloha projekčního centra DB snímku			Poloha projekčního centra vstupního snímku			Poloha uživatele			Souřadnicové rozdíly		Střední souřadnicová chyba m_{xy} [m]	Translace C1 a C2			Odhadnutý směrový vektor translace			Odhadnuté měřítko	Délka základny [m]
			X [m]	Y [m]	Z [m]	X [m]	Y [m]	Z [m]	X [m]	Y [m]	Z [m]	Δx [m]	Δy [m]		X [m]	Y [m]	Z [m]	X [m]	Y [m]	Z [m]		
IMG_4908	IMG_4956	2486	12,25	8,44	1,68	9,75	9,30	1,68	9,75	9,20	1,71	0,00	0,10	0,07	-2,500	0,860	0,000	-0,957	0,288	0,012	2,602	2,64
IMG_4961	IMG_4297	2761	16,15	8,72	1,65	15,17	8,38	1,62	15,22	8,33	1,63	-0,05	0,05	0,05	-0,980	-0,340	-0,030	-0,924	-0,380	-0,013	1,002	1,04
IMG_5074	IMG_4743	2674	15,55	10,34	1,65	15,83	9,23	1,58	15,47	9,58	1,63	0,36	-0,35	0,36	0,280	-1,110	-0,070	-0,107	-0,994	-0,030	0,758	1,14
IMG_5082	IMG_4956	1490	12,25	8,44	1,68	9,73	9,23	1,59	9,69	9,10	1,67	0,04	0,13	0,10	-2,520	0,790	-0,090	-0,968	0,248	-0,006	2,641	2,64
IMG_5071	IMG_4956	1790	12,25	8,44	1,68	11,92	8,06	1,82	12,41	7,64	1,63	-0,49	0,42	0,46	-0,330	-0,380	0,140	0,201	-0,997	0,061	0,821	0,50
IMG_5073	IMG_4908	1637	9,75	9,30	1,62	9,85	9,21	1,60	9,71	9,21	1,60	0,14	0,00	0,10	0,100	-0,090	-0,020	-0,475	-0,860	-0,182	0,119	0,13
IMG_4898	IMG_4297	2638	16,15	8,72	1,64	18,42	9,76	1,67	18,25	9,62	1,64	0,17	0,14	0,16	2,270	1,040	0,030	0,919	0,394	-0,001	2,288	2,50
IMG_4962	IMG_4297	2889	16,15	8,72	1,64	16,90	8,44	1,60	16,90	8,61	1,56	0	-0,17	0,12	0,750	-0,280	-0,040	0,982	-0,147	-0,111	0,766	0,80
IMG_4954	IMG_4958	1769	12,40	8,42	1,60	12,65	9,66	1,60	12,76	9,91	1,57	-0,11	-0,25	0,19	0,250	1,240	0,000	0,234	0,971	-0,020	1,525	1,26
IMG_4745	IMG_4297	2688	16,15	8,72	1,64	17,30	8,57	1,60	17,27	8,58	1,64	0,03	-0,01	0,02	1,150	-0,150	-0,040	0,991	-0,129	-0,008	1,124	1,16
IMG_4895	IMG_4297	2244	16,15	8,72	1,64	16,72	8,14	1,62	16,73	8,01	1,64	-0,01	0,13	0,09	0,570	-0,580	-0,020	0,630	-0,776	-0,003	0,916	0,81

Tab. 9 Podrobné výsledky testování snímku IMG_4955

Vstupní snímek IMG_4955																					
DB snímek	Počet korespondencí	Poloha projekčního centra DB snímku			Poloha projekčního centra vstupního snímku			Poloha uživatele			Souřadnicové rozdíly		Střední souřadnicová chyba m_{xy} [m]	Translace C1 a C2			Odhadnutý směrový vektor translace			Odhadnuté měřítko	
		X [m]	Y [m]	Z [m]	X [m]	Y [m]	Z [m]	X [m]	Y [m]	Z [m]	Δx [m]	Δy [m]		X [m]	Y [m]	Z [m]	X [m]	Y [m]	Z [m]		
4954	3065	12,64	9,66	1,59	12,50	9,68	1,65	12,83	9,80	1,58	-0,33	-0,12	0,25	-0,14	0,02	0,06	0,782	0,621	-0,058	0,238	
4956	2516	12,23	8,44	1,69	12,50	9,68	1,65	12,33	10,05	1,70	0,17	-0,37	0,29	0,27	1,24	-0,04	0,052	0,998	0,134	1,609	
4961	2303	15,16	8,39	1,62	12,50	9,68	1,65	12,75	9,55	1,66	-0,25	0,13	0,20	-2,66	1,29	0,03	-0,900	0,435	0,013	2,685	
4909	2176	9,76	9,20	1,61	12,50	9,68	1,65	12,11	9,40	1,55	0,39	0,28	0,34	2,74	0,48	0,04	0,996	0,084	-0,025	2,353	
4894	2087	16,71	8,12	1,66	12,50	9,68	1,65	12,60	9,53	1,66	-0,1	0,15	0,13	-4,21	1,56	-0,01	-0,945	0,326	0,001	4,345	
4743	2033	15,55	10,34	1,65	12,50	9,68	1,65	12,98	9,71	1,71	-0,48	-0,03	0,34	-3,05	-0,66	0,00	-0,971	-0,236	0,023	2,659	
4297	1942	16,15	8,72	1,64	12,50	9,68	1,65	12,53	9,73	1,78	-0,03	-0,05	0,04	-3,65	0,96	0,01	-0,962	0,270	0,035	3,758	
4962	1838	16,91	8,44	1,60	12,50	9,68	1,65	12,75	9,53	1,69	-0,25	0,15	0,21	-4,41	1,24	0,05	-0,967	0,254	0,021	4,297	
4745	1812	17,30	8,57	1,60	12,50	9,68	1,65	13,05	9,49	1,60	-0,55	0,19	0,41	-4,80	1,11	0,05	-0,977	-0,211	0,001	4,346	
4895	1636	16,71	8,14	1,62	12,50	9,68	1,65	12,63	9,68	1,66	-0,13	0	0,09	-4,21	1,54	0,03	-0,935	0,354	0,009	4,369	

Obsah přiloženého CD

Kubickova_DP.pdf – text diplomové práce

Kubickova_DP_prakticka_cast.py – zdrojový kód praktické části práce

data/data_scena1_a – soubor obsahující 14 databázových snímků testované scény č. 1 s příslušnými xml soubory

data/data_scena1_b – soubor obsahující 5 databázových snímků testované scény č. 1 s příslušnými xml soubory

data/data_scena2 – soubor obsahující 9 databázových snímků testované scény č. 2 s příslušnými xml soubory

vstup/ - složka obsahující testované vstupní snímky