

**Žilinská univerzita v Žiline**

**Fakulta riadenia a informatiky**

## **Diplomová práca**

**Študijný program: Informačné systémy**

**Zameranie: Spracovanie dát**

Bc. Marek Drevenák

Určovanie polohy objektov s využitím stereovízie

Object positioning based on stereo vision

Vedúci práce: Ing. Peter Tarábek, PhD., KMMOA, ŽU

Reg. č. 358/2017      Apríl 2018

**Žilina**

ZADANIE TÉMY DIPLOMOVEJ PRÁCE.

Študijný program : Informačné systémy

Zameranie: Spracovanie dát

Meno a priezvisko

Marek Drevenák

Osobné číslo

556208

Názov práce v slovenskom aj anglickom jazyku

Určovanie polohy objektov s využitím stereovízie

Object positioning based on stereo vision

Zadanie úlohy, ciele, pokyny pre vypracovanie

(Ak je málo miesta, použite opačnú stranu)

**Cieľ diplomovej práce:**

Cieľom práce je preskúmanie možností určovania polohy statického objektu snímaného z pohybujúcich sa kamier. Vstupom do systému bude videosekvencia s rozpoznávanými záujmovými objektmi a požadované informácie o kamerách, vrátane GPS. Práca bude skúmať vplyv konfigurácie kamier a parametrov scény (poloha, veľkosť objektov a trajektória pohybu kamier), vrátane samotnej existencie týchto informácií na presnosť určovania polohy objektov. Predpokladá sa vytvorenie simulátora na generovanie datasetov. Simulátor umožní robiť experimenty v kontrolovanom prostredí.

**Obsah:**

1. Analýza súčasného stavu.
2. Výber, prípadne návrh vhodných metód a princípov pre určovanie polohy statických objektov z pohybujúcich sa kamier.
3. Vytvorenie simulátora pre generovanie datasetov v kontrolovanom prostredí.
4. Implementácia metód.
5. Preskúmanie vplyvu konfigurácie kamier a parametrov scény (poloha, veľkosť objektov a trajektória pohybu kamier), vrátane samotnej existencie týchto informácií na presnosť určovania polohy objektov.
6. Záver.

Meno a pracovisko vedúceho DP: Ing. Peter Tarábek, PhD., KMMOA, ŽU

Meno a pracovisko tútora DP:

- 5 FEB. 2013

vedúci katedry  
(dátum a podpis)

## **Pod'akovanie**

Na úvod sa chcem úprimne poďakovať svojim rodičom a známym za podporu počas vypracovávania tejto práce, ale aj počas celého doterajšieho štúdia.

Chcel by som sa poďakovať aj vedúcemu práce Ing. Petrovi Tarábkovi, PhD. za jeho ochotu a rady pri jej vypracovávaní.

Ďakujem Vám

## **Vyhlásenie**

Čestne vyhlasujem, že som diplomovú prácu vypracoval samostatne pod odborným vedením vedúceho práce Ing. Petra Tarábka, PhD. a použil som len literatúru, ktorú uvádzam v zozname.

V Žiline dňa

Marek Drevenák

## *Abstrakt*

DREVENÁK, Marek: Určovanie polohy objektov s využitím stereovízie [Diplomová práca] – Žilinská univerzita v Žiline. Fakulta riadenia a informatiky. – Vedúci práce: Ing. Peter Tarábek, PhD. – Žilina 2018 – 64 s.

Práca skúma určovanie polohy objektu využitím kamier. Analyzuje súčasný stav a základné využívané prístupy akými sú stereo korešpondencia, rekonštrukcia trajektórie svetelných lúčov a triangulácia na základe uhlov, pod ktorými pozorujeme objekt. Vytvára umelý dataset v kontrolovaných podmienkach, implementuje spomenuté metódy a simulátor na automatizované vykonanie experimentov. Výsledky sú analyzované a metódy sú porovnávané vzhľadom na ich presnosť. Okrem toho skúma vplyv trajektórie kamier, ich konfigurácie a parametrov, taktiež parametrov scény a chýb v dátach na presnosť odhadu polohy.

Kľúčové slová: určovanie polohy objektu, stereovízia

## *Abstract*

DREVENÁK, Marek: Object positioning based on stereo vision [Diploma Thesis] – University of Žilina. Faculty of Management, Science and Informatics. – Thesis tutor: Ing. Peter Tarábek, PhD. – Žilina 2018 – 64 p.

Thesis investigates object positioning using cameras. It analyses the state of art, basic principles which are stereo correspondence, light beam trajectory reconstruction and triangulation based on angles, under which we observe the object. It creates artificial dataset in controlled environment, implements mentioned methods and simulator for automatic experimenting. It analyses results and compares methods based on their precision. Furthermore it investigates effect of cameras' trajectories, their configuration and parameters, scene parameters and errors in data on precision of location estimate.

Keywords: object positioning, stereo vision

# Obsah

---

Zoznam obrázkov .....	ii
Zoznam tabuliek .....	iii
Zoznam grafov .....	iii
Použité skratky .....	iv
<b>1 Úvod</b> .....	1
<b>2 Súčasný stav</b> .....	4
2.1 Základné používané prístupy .....	7
2.2 Teoretické východiská .....	11
2.2.1 Proces zachytávania obrazu a geometria obrazu .....	11
2.2.2 Skreslenie spôsobené šošovkami .....	13
2.2.3 Transformácie .....	14
2.2.4 Kalibrácia kamery .....	16
2.2.5 Triangulácia .....	18
2.2.6 Stereo korešpondencia .....	19
2.2.7 Rekonštrukcia trajektórie svetelných lúčov .....	25
2.2.8 Triangulácia založená na pozorovaných uhloch k objektu .....	28
<b>3 Príprava dát</b> .....	29
<b>4 Implementácia simulátora experimentov a metód</b> .....	35
4.1 Simulátor experimentov .....	36
4.2 Podpora vizualizácie a práce s datasetmi .....	39
4.3 Rozšírenie simulátora na úlohu určovania polohy .....	40
<b>5 Experimentálne výsledky</b> .....	43
5.1 Vizualizácia výsledkov .....	43
5.2 Presnosť metód .....	45
5.3 Závislosť medzi vzdialenosťou a presnosťou .....	49
5.4 Vplyv nepresnej GPS polohy, azimutu a <i>bounding boxu</i> na presnosť .....	52
5.5 Vplyv konfigurácie a parametrov kamier na presnosť .....	53
5.6 Ostatné parametre .....	58
<b>6 Zhrnutie a záver</b> .....	59
Zoznam použitej literatúry .....	62
Prílohy .....	64

# Zoznam obrázkov

---

Obrázok 1. Význam horizontálneho posunu v stereo videní .....	8
Obrázok 2. Základné kroky stereo korešpondencie .....	8
Obrázok 3. Ukážka rekonštrukcie svetelného lúča. Pre každú fotografiu je potrebné poznať polohu a natočenie kamery. ....	10
Obrázok 4. Základné kroky rekonštrukcie svetelných lúčov .....	10
Obrázok 5. <i>Pin hole</i> kamerový model .....	12
Obrázok 6. Šošovky. a) Vankúšovité a sudovité skreslenie obrazu – <i>pincushion</i> a <i>barrel distortion</i> b) Systém šošoviek využívaný v kvalitnejších kamerách .....	14
Obrázok 7. Kalibrácia kamery. a) Vizualizácia krokov a najčastejšie výsledky kalibrácie b) Skutočné obrázky vzoru šachovnice .....	17
Obrázok 8. Epipolárna geometria .....	20
Obrázok 9. Rektifikácia fotografií so zobrazenými epipolárnymi priamkami .....	21
Obrázok 10. Hľadanie a párovanie charakteristických bodov. a) Základná technika <i>block matching-u</i> b) Rozšírenie s adaptívnou veľkosťou c) Rozšírenie s hierarchiou veľkostí bloku d) 4D priestor s cenou párovania DSI a vytváranie plochy, ktorá určuje párovanie .....	22
Obrázok 11. Ukážky rôzneho spôsobu vizualizácie disparity mapy .....	23
Obrázok 12. Vzťah medzi hĺbkou a disparitou .....	24
Obrázok 13. Proces <i>forward</i> projekcie .....	25
Obrázok 14. Spätná projekcia a problém nejednoznačnosti. a) Vizualizácia inverzných transformácií pri spätnej projekcii a ich potenciálnych chýb b) Problém vnímania hĺbky .....	28
Obrázok 15. Triangulácia v rovine využitím uhlov a ich získavanie .....	29
Obrázok 16. Nástroj Blender. Postupne zľava: prostredie nástroja, využitie pri CGI vo filmoch, fyzikálne experimenty. ....	31
Obrázok 17. Štruktúra datasetu pre jednu scénu s konkrétnymi parametrami .....	32
Obrázok 18. Štruktúra experimentu a jeho význam .....	36
Obrázok 19. Diagram základných tried v knižnici <i>Simulator</i> .....	37
Obrázok 20. Implementované knižnice .....	38
Obrázok 21. Štruktúra knižnice <i>Triangulation</i> .....	40
Obrázok 22. Model prístupových vrstiev .....	41
Obrázok 23. Vizualná ukážka pracovania jednotlivých metód zobrazená využitím implementovanej knižnice <i>View</i> .....	44
Obrázok 24. Vizualizácia priestoru možných triangulácií pre paralelné kamery .....	57

## Zoznam tabuliek

---

Tabuľka 1. Zoznam modelovaných scén.....	34
Tabuľka 2. Zoznam kamier s parametrami.....	34
Tabuľka 3. Referenčné rozsahy vzdialeností od objektu pre jednotlivé datasety .....	51
Tabuľka 4. Presnosť odhadu vnútorných parametrov procesom kalibrácie .....	55

## Zoznam grafov

---

Graf 1. Porovnanie výsledkov metód zo všetkých datasetov .....	45
Graf 2. Porovnanie výsledkov metód zo všetkých datasetov .....	45
Graf 3. Odstránenie najviac nepresných odhadov - outlierov - pri metóde <i>triangulácia cez uhly</i> . Ukážka na vybraných datasetoch. ....	46
Graf 4. Odstránenie najviac nepresných odhadov - outlierov - pri metóde <i>triangulácia cez uhly</i> . Ukážka len na vybraných datasetoch. ....	47
Graf 5. Výsledky metód pre jednotlivé datasety .....	47
Graf 6. Vplyv rozšírenia 2D výsledku metódy na 3D využitím opísanej heuristiky. Výsledky predstavujú priemer zo všetkých výsledkov metódy <i>triangulácie cez uhly</i> . Metóda bola testovaná s históriou najviac 60 predošlých fotografií. ....	48
Graf 7. Vplyv vzdialenosti od objektu na presnosť. Zohľadnené sú všetky dáta produkované metódami. ....	49
Graf 8. Vplyv vzdialenosti od objektu na presnosť. Metódy sú upravené tak, že pri každom odhade je zohľadnená predošlá história, na ktorú je aplikovaná opísaná metóda filtrovania outlierov .....	50
Graf 9. Porovnanie metód na datasetoch s rovnakým rozsahom vzdialeností od objektu – pod 30 metrov. Pri výpočte bola obmedzená história na 2 sekundy, preto sa výsledky mierne odlišujú od predošlých. ....	51
Graf 10. Vplyv nepresnosti GPS polohy na presnosť metód. Namerané dáta sú prekryté trendom bodkovanou čiarou. Legenda je uvedená v grafe vľavo hore .....	52
Graf 11. Vplyv nepresnosti azimutu na presnosť metód. Namerané dáta sú prekryté trendom bodkovanou čiarou. Legenda je rovnaká ako v predošlom grafe .....	52
Graf 12. Vplyv chyby bounding boxu na presnosť metód. Namerané dáta sú prekryté trendom bodkovanou čiarou .....	53
Graf 13. Chyba projekcie pre kamery použité v datasetoch.....	54
Graf 14. Porovnanie filmovej a smartfónovej kamery .....	55
Graf 15. Chyba triangulácie pre stereo pár filmových kamier s odstupom 15 cm. Azimut určuje uhol, pod ktorým pozorujeme objekt.....	56
Graf 16. Chyba triangulácie pre rôzne konfigurácie kamier: odstup, posun v smere jazdy a vzájomne natočenie. Objekt je vo všetkých prípadoch pozorovaný pod azimutom $0^\circ$ od pomyselných osí kamier .....	58

## Použité skratky

---

2D 3D 4D	Dvojdimenziálny, trojdimenziálny resp. štvordimenziálny
GPS	Globálny lokalizačný systém; z anglického <i>Global Positioning System</i>
CGI	Počítačom vytvorená grafika, fotografie; z anglického <i>Computer Generated Imagery</i>
FPS	Obrazová frekvencia; určuje počet obrázkov zobrazených za sekundu; z anglického <i>frames per second</i>
API	Programové rozhranie aplikácie pre prístup k vnútornej funkcionalite; z anglického <i>application programming interface</i>
SI	Medzinárodná sústava jednotiek; z francúzskeho <i>systeme international</i>
CCD	Súčiastka s nábojovou väzbou, využíva sa vo fotoaparátoch pre záznam obrazovej informácie; z anglického <i>charge-coupled device</i>
CMOS	Súčiastka vo fotoaparátoch pre záznam obrazovej informácie; každý pixel má fotodetektor a zosilňovač; z anglického <i>complementary metal-oxide-semiconductor</i>
RGB, HSI, HSL, HSV, YUV, CMY(K)	Farebné modely, ktoré definujú kódovanie farieb; skratky postupne <i>Red Green Blue, Hue Saturation Intensity, Hue Saturation Lightness, Hue Saturation Value, YUV, Cyan Magenta Yellow (Key)</i>
DSI	Priestor s cenou párovania využívaný pri metódach stereo korešpondencie; anglického <i>disparity space image</i>
NASA	Národná letecká a vesmírna agentúra v USA; z anglického <i>National Aeronautics and Space Administration</i>
XML	Rozšíriteľný značkovací jazyk; z anglického <i>eXtensible Markup Language</i>
JSON	Typ formátu súboru pre prenos dát; z anglického <i>JavaScript Object Notation</i>
Qt	Framework pre vytváranie grafického užívateľského rozhrania, spočiatku vyvíjaný firmou NOKIA v jazyku C++
PNG, JPG	Formáty súborov na uchovanie rastrovej grafiky



# 1 Úvod

---

Za posledné roky urobilo počítačové videnie veľký pokrok. Podarilo sa zdokonaľiť mnohé techniky, ktoré sú úspešne využívané rôznymi komerčnými produktmi. Medzi nimi možno spomenúť napríklad: autonómne taxi vozidlá od spoločnosti Uber, ktoré sú už v prevádzke v Arizone [1], taktiež spoločnosti Volkswagen, Huawei, Samsung, Nvidia, Baidu, Volvo, Fiat, Apple, Intel, BMW, Google a mnohé iné firmy investujú ročne veľké množstvo peňazí do ich vývoja a testovania a majú naplánované v najbližších rokoch uvedenie na trh plne autonómneho komerčného vozidla [2], rozšírená realita a hologramy od Microsoft HoloLens [3], Mars Rover využívajúci kamery na navigáciu a detekciu prekážok [4], rozpoznávanie tváří od Facebooku s rovnakou úspešnosťou ako človek [5], panoramatický pohľad na ulicu na Google mapách a iné. Nie všetky úspešne realizované projekty a nápady sú publikované verejne. Napríklad vojenské autonómne dróny, samonavádzacie strely a iné vojenské a tajné technológie, ktoré využívajú údaje z kamier.

Počítačové videnie má najväčšie uplatnenie v oblastiach ako virtuálna alebo rozšírená realita (vizualizačný softvér a iné), vizuálne asistenčné systémy pre človeka (automobilové, chirurgické asistenčné systémy a iné), rozpoznávanie gest, tváre, textu a vo všeobecnosti objektov, inšpekčné a kontrolné systémy, syntetizácia pohľadov a mnohé iné. Jedny z najpopulárnejších oblastí záujmu sú robotika a autonómne dopravné prostriedky, najmä vozidlá.

Táto oblasť sa rozvíja a vedie k tomu aj skutočnosť, že kamera je relatívne nenákladný senzor, ktorý dokáže poskytnúť veľa informácie. V minulosti boli hlavné aplikácie v oblasti vizuálnej kontroly a robotike. V súčasnosti je skôr dôraz na počítačovú grafiku, virtuálnu realitu a komunikáciu, čo má za následok zvyšujúci sa dopyt po kvalitnejšej priestorovej informácii. Existujúce systémy sú často postavané so špecializovaným hardvérom (ako laserové diaľkomere). To zvyšuje ich cenu. Nové aplikácie majú za cieľ využívať lacné senzory. Takýmito sú napríklad kamery v mobilných zariadeniach. Vyvinuli sa teda techniky na získanie priestorovej informácie, ktorým vystačí len informácia z kamier. [6]

Existuje množstvo netriviálnych problémov, ktoré musia odborníci na počítačové videnie riešiť. Často neexistujú univerzálne a robustné riešenia napriek pokroku, ktorý možno pozorovať. Stretávame sa s tým, že riešenia sú navrhnuté na špecifické varianty problému. Vo všeobecnosti nie sú navrhnuté overené postupy na riešenie určitej úlohy. Tu sú nápomocné vedecké články porovnávajúce rôzne metódy a prístupy. Jednou z výziev počítačového videnia je aj problém priestorovej alebo trojdimenzionálnej rekonštrukcie (ďalej 3D rekonštrukcie).

Ak sa pozrieme bližšie na prácu kamery, možno ju opísať ako transformáciu priestorovej informácie na dvojrozmernú. Fotóny odrazené od povrchu predmetu sú zachytené na svetlocitlivom médiu – elektronickom senzore alebo fotografickom filme. V prípade digitálnej kamery je signál z elektronického senzora prevedený na digitálnu informáciu a uchovaný v tvare digitálneho obrázka. Pri vytváraní obrazu v kamere sa stráca informácia o hĺbke respektíve vzdialenosti. Je to transformácia z troch dimenzií do dvoch (3D na 2D transformácia), pričom neexistuje inverzná transformácia – teda taká, ktorá by vedela k bodu na obrázku jednoznačne priradiť bod v priestore. Toto je dôležité si uvedomiť. A preto je potrebné navrhnuť metódy, ktoré budú riešiť tento problém. Ten je označovaný aj ako určovanie polohy objektu alebo lokalizácia. Zovšeobecnením je 3D rekonštrukcia – poloha a tvar objektu. Tá sa využíva napríklad v robotike na detekciu prekážok, vytváranie máp a navigáciu; v medicíne na vytvorenie priestorového modelu orgánov a buniek; v rozšírenej realite na správne zobrazenie umelého objektu do reálnej scény; v interaktívnych systémoch na zistenie presnej polohy gesta alebo povelu, ako napríklad pri hrách; vo filmovom priemysle na dotvorenie objektov alebo postáv do scény. [7]

Problém 3D rekonštrukcie bol často riešený za použitia údajov z rôznych senzorov ako napríklad infračervené, ultrazvukové a rádiometrické senzory – radar, sonar [8]. Vo všeobecnosti senzory, ktoré nám poskytujú informáciu o vzdialenosti v tvare 2D obrázka, sú označované v anglickej literatúre ako *range cameras*. Môže to byť špeciálne navrhnuté zariadenie využívajúce niektorú zo spomenutých fyzikálnych vlastností. Medzi takéto patrí technika kódovaných clon (z voľného prekladu *coded aperture*), ktorá využíva rôzne vzory pre clonu pre röntgenovú alebo gama kameru. Na základe tieňa, ktorý je vyprodukovaný na optickom senzore pre konkrétnu clonu, dokážeme usúdiť vlastnosti elektromagnetického žiarenia (svetla) a teda aj vzdialenosti. Takéto zariadenia majú využitie napríklad v tomografii. Takýto prístup, teda využitie senzorov

poskytujúcich priestorovú informáciu, sa označuje za aktívny a metódy, ktoré ho využívajú sa označujú za aktívne. Opakom sú pasívne metódy, pri ktorých nevyužívame lasery ani ďalšie podobné zariadenia. Ich vstupom je množina dvojrozmerných obrázkov a výstupom 3D model popisujúci umiestnenie objektu prípadne jeho tvar. Pasívnymi metódami sa bližšie zaoberá fotogrametria – veda zaoberajúca sa meraním priestoru z fotografie (najmä vzdialenosti a uhly).

Existujú 2 základné typy pasívnych metód: môže sa jednať o monokulárne alebo binokulárne metódy. Monokulárne metódy využívajú len obrázky z jednej nemej polohy kamery. Problém sa dá pripodobniť tomu, že si človek jedno oko zakryje a snaží sa uhádnuť vzdialenosť k objektu prípadne jeho veľkosť. Pri riešení sa využívajú najmä tieň a textúra – metódy su anglicky označované ako *Shape-From-Shading* resp. *Shape-From-Texture* – vo všeobecnosti sú veľmi náročné a vyžadujú vyššiu mieru porozumenia scény. Binokulárne alebo tiež nazývané stereo metódy využívajú obrázky z aspoň 2 kamier alebo jednej kamery z aspoň 2 rôznych polôh. Tieto metódy sa snažia pracovať podobným spôsobom ako funguje ľudský vizuálny systém. [7]

Cieľom práce je spracovať videosekvenciu s rozpoznávanými objektmi a odhadnúť ich polohu. A tak ako aj názov práce naznačuje, chceme využívať pasívne stereo metódy 3D rekonštrukcie. Kamera resp. kamery sú pohyblivé a objekt je statický.

Vstupom je videosekvencia reprezentujúca postupnosť obrázkov, ku ktorým sú k dispozícii informácie o kamerách (ich aktuálnych polôh a smerovania, resp. súradníc z Globálneho lokalizačného systému – ďalej GPS – z ktorých je možné získať ich polohu a natočenie) a o rozpoznanom objekte na fotografii. Naším cieľom nie je rozpoznávanie objektu vo fotografii, ale určenie jeho polohy. To môže predstavovať situáciu, že už máme výstup z nejakého detektora a chceme zistiť, kde sa nájdené objekty nachádzajú v priestore. Výhradne na základe týchto informácií a využitím vhodných metód sa pokúsime čo najpresnejšie určiť polohu objektu.

Ďalším z cieľov práce je vytvorenie datasetov v kontrolovanom prostredí. Aby bolo možné takejto dost' abstraktne zadanej úlohe priradiť aj konkrétny význam a nemuseli sme sa rozprávať o pohybe kamery po trajektórii a záujmových bodoch na fotografii, opíšeme úlohu ako:

*„Určte polohu dopravnej značky z pohybujúceho sa vozidla po ceste, v ktorom sú za čelným sklom umiestnené jedna alebo dve kamery.“*

Skúmanie na takomto konkrétnom príklade nemení nič na všeobecnosti riešenia a jeho použití. Statickým objektom bude dopravná značka umiestnená pri ceste a kamery budú opisovať rovnakú trajektóriu pohybu ako vozidlo. K dispozícii pre každý snímok bude GPS poloha kamier. V prípade, že máme k dispozícii GPS polohu vozidla z iného zariadenia a za predpokladu, že sa kamery vzhľadom na vozidlo nepohybujú, je ich možné získať konštantným posunom z GPS vozidla. Ďalším z predpokladov je, že sú k dispozícii všetky vnútorné parametre kamery ako rozmery snímacieho prvku kamery, ohnisková vzdialenosť, zorné pole a iné, ktoré daná metóda potrebuje. Trajektória vozidla, konfigurácia kamier, parametre scény a kamier sú predmetom skúmania.

Futuristická vízia, ktorá nás vedie k hľadaniu riešenia tohto problému je, že raz bude aj počítač schopný vidieť svet priestorovo aspoň tak dobre, ako to dokáže človek. Vedieť odhadnúť hĺbku a vzdialenosť objektov len na základe očí, teda kamier. Rozumieť tvaru a hĺbke priestoru, ktorý vidí. To predstavuje základnú požiadavku pre stroje a robotov, ktoré chceme, aby sa podobali človeku. Či už sa bude jednať o humanoidov alebo dokonca androidov, ktorí sa budú schopní pohybovať po svete a vnímať priestor a objekty v ňom využitím vizuálnych senzorov.

## 2 Súčasný stav

---

V tejto kapitole budú predstavené prostriedky, ktoré sú k dispozícii na riešenie tohto problému v súčasnosti a základné teoretické východiská potrebné na ich pochopenie.

Vo vedeckých článkoch zaoberajúcich sa touto problematikou sú využívané rôzne prístupy ako získať priestorovú informáciu. Existujú špecifické prístupy pre monokulárne kamery pohybujúce sa v priestore, binokulárne kamery alebo konfigurácie s viacerými kamerami, ktoré sa snažia využiť dodatočný pohľad na scénu. Taktiež pre špecifické umiestnenie a smerovanie kamery. Spravidla býva kamera umiestnená tak, aby obraz zachytával záujmovú oblasť priestoru.

Pri automobiloch to je na streche, za čelným sklom alebo v prednej časti vozidla. Ak sa jedná o kamery, ktoré vytvárajú stereo pár, tie sú zvyčajne orientované rovnakým smerom kolmo od spoločnej osi a umiestnené so známou vzdialenosťou medzi nimi. V prípade viacerých kamier a pohľadov nie je zaužívaná špecifická konfigurácia a závisí od konkrétnej úlohy a jej cieľov. Ak je cieľom vytvárať panoramatické zábery spájaním z viacerých fotografií, kamery sú umiestnené tak, že sa ich obraz po dvoch sčasti prekrýva a zachytáva vo výsledku väčší uhol ako zorné pole kamery. Ak je cieľom získať predstavu o tvare objektu, kamery zvyčajne obkolesujú objekt z viacerých uhlov. Taktiež smerovanie kamery môže byť rôzne. Tá môže smerovať dopredu, dopredu s naklonením alebo dokonca nahor. Kamera smerujúca nahor je využívaná napríklad pri úlohách lokalizácie robotov vo vnútorných priestoroch. Navigujú sa pomocou bodov na strope ako rohy, lampy prípadne umelo vytvorené svetelné vzory. Kamera s naklonením smerom nadol sa využíva napríklad pri navigácii vozidiel a detekcii prekážok v blízkosti alebo taktiež v prípade zlého osvetlenia. Vtedy je ku kamere pripevnené aj umelé svetlo. Tu môže vznikáť problém pri veľmi rýchlom pohybe.

Využívanie vizuálnych údajov na vytvorenie modelu priestoru prináša svoje nevýhody: vizuálny vzhľad prostredia sa mení. Mení sa, keď sa kamera pohybuje, ale aj vplyvom ďalších okolností ako sú zmeny v podmienkach osvetlenia, ktoré sú prítomné pri všetkých aplikáciách v skutočnom teréne. Napríklad nedostatočné osvetlenie pri tme, hmle, snežení, prípadne oslepanie z odrazov slnečných lúčov. Prostredie sa môže taktiež zmeniť po tom, čo už bol model priestoru vytvorený. Napríklad tým, že je nejaká jeho časť zakrytá pohybujúcimi sa objektmi alebo vplyvom pohybu kamery samotnej [9]. Nedostatočná kvalita fotografie, objekt vzdialený príliš ďaleko na rozpoznanie a iné. Počítač tak trpí rovnakými obmedzeniami pri vnímaní svojho okolia kamerou ako aj oči človeka. To sú výzvy, ktoré sú kladené na vedecké články, zaoberajúce sa metódami rekonštrukcie 3D štruktúry.

Túto úlohu okrem toho sťažujú nepresnosti v získaných GPS súradniciach a nepresnosti v parametroch kamery (ak sú získavané napríklad z kalibrácie). Musíme rátať s tým, že údaje a podmienky v teréne nebudú dokonalé.

Medzi základné kritériá zohľadňovaných metód patrí presnosť 3D rekonštrukcie. Konkrétne presnosť odhadu polohy nejakého objektu v priestore a ako sa dokáže metóda vysporiadať s nedokonalosťami. Často rôzne štúdie pracujú so špecifickou situáciou, ktorou je potom celá metóda ovplyvnená a obmedzovaná. Vo výsledku sa však vždy porovnáva presnosť odhadu polohy, prípadne hĺbkovej mapy, z ktorej možno túto polohu vypočítať.

Od počiatku využívania počítačového videnia na lokalizáciu a 3D rekonštrukciu v oblasti ako robotika pre navigáciu sa pracovalo s metódami založenými výhradne na obraze. Často kvôli cene iných senzorov. Integrácia GPS informácie na obohatenie a zlepšenie priestorového modelu nie je nový nápad. Využitie aj iných zdrojov na určenie polohy kamery je vec, skúmaná nejedným vedeckým článkom. Pri metódach zaoberajúcich sa vo všeobecnosti 3D rekonštrukciou z obrazu takýto údaj často nie je predpokladom. Môžeme však pri nich uvažovať nad rozšírením. To môže metódu zmeniť na použiteľnú, aj keby sa mohlo spočiatku zdať, že metóda nie je vhodná.

Ak sa pozrieme na prístupy zo študovaných vedeckých článkov a na akých základoch sú postavené, dostávame tieto základné spôsoby určovania polohy: hľadanie spoločných bodov na obrázkoch a vytvorenie hĺbkovej mapy a hľadanie spoločných bodov na obrázkoch a nájdenie priesečníka svetelných lúčov získaných z procesu spätnej projekcie. Samozrejme existujú obmeny týchto spôsobov, ktoré však využívajú spoločné idey. Prvý spôsob sa využíva najmä v prípade, keď sa jedná o stereo kamery. To je častý prípad v takýchto úlohách a dokonca existujú porovnávacíe testy a rebričky na vyhodnocovanie úspešnosti metód využívajúci tento prístup (anglicky *benchmark*). Druhý spôsob je všeobecnejší a je ho možné použiť pre ľubovoľný prípad konfigurácie kamier. Samozrejme každý spôsob má svoje predpoklady, za ktorých je ho možné použiť. Tie budú bližšie vysvetlené v nasledujúcej podkapitole.

Ak by sme sa chceli pozrieť až na základný kameň týchto metód, dospeli by sme ku triangulácii, ktorá je všade prítomná pri úlohách 3D rekonštrukcie. Aký presný význam má tento pojem v počítačovom videní, bude vysvetlené v ďalšom texte. Laicky však možno povedať, že sa jedná o spôsob určenia polohy objektu z dvoch kamier, na ktorých je objekt viditeľný.

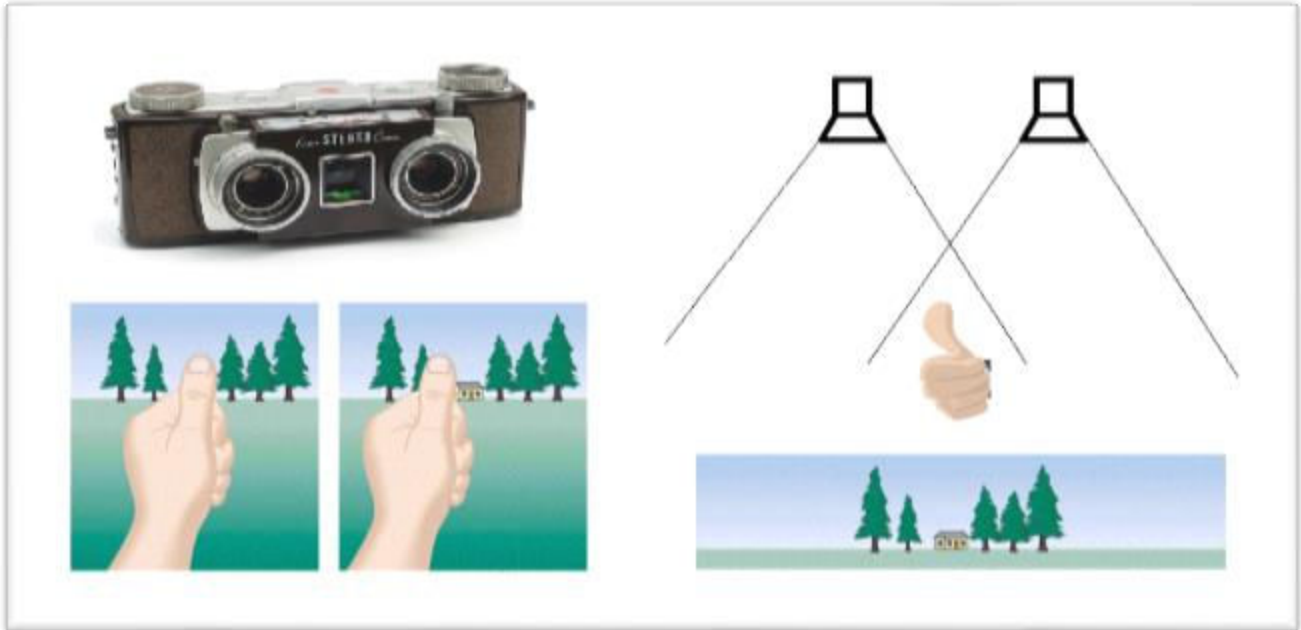
## 2.1 Základné používané prístupy

Ako už bolo spomínané medzi základné myšlienky, ktoré sa spájajú s rekonštrukciou 3D štruktúry sú hľadanie spoločných bodov na obrázkoch a vytvorenie hĺbkovej mapy a hľadanie priesečníka svetelných lúčov získaných z procesu spätnej projekcie.

**Hľadanie stereo korešpondujúcich párov bodov na obrázkoch a vytvorenie hĺbkovej mapy** predstavuje celú kategóriu metód a postupov pri rekonštrukcii. Túto kategóriu je možné označiť aj ako stereo korešpondencia (z anglického *stereo correspondence*). Je to proces odhadnutia 3D modelu scény z dvoch alebo viacerých fotografií (obrázkov) nájdením korešpondujúcich obrazových bodov (ďalej pixelov) a prevedením ich 2D polohy na 3D hĺbku.

Táto kategória vznikla na základe štúdia fungovania vizuálneho systému samotného človeka. Z prvých výskumov sa zistilo, že človek vníma hĺbku rozdielom v pohľade z ľavého a pravého oka (experimenty známe ako *Bela Julesz Random dot experiment* [10]). Ak si človek striedavo zatvorí ľavé a pravé oko, uvidí, že predmet blízko pri ňom skáče relatívne na pozadie. Rovnaký fenomén pozorujeme pri páre fotografií. To, ako veľa sa predmet horizontálne posunie (v oblasti počítačového videnia na to existuje pojem *disparity*) je nepriamo úmerné vzdialenosti od pozorovateľa. Prvé algoritmy boli vyvinuté v oblasti fotogrametrie na automatické vytváranie topografických máp s vrstevnicami z prekrývajúcich sa vzdušných snímok. Bližšie bude tento proces vysvetlený v kapitole 2.2.6 Stereo korešpondencia

Predpokladom pre využitie tejto skupiny metód je vstup z dvoch kamier, ktoré sú umiestnené vedľa seba ako oči človeka, to znamená, že obe zachytávajú spoločnú časť obrazu. Existujú aj rozšírenia takejto geometrie pre tri kamery. Takéto metódy sú náročnejšie na kvalitu fotografie, pretože sa v nej hľadajú korešpondujúce kľúčové body (z anglického *key points*). Často majú problémy pri fotografiách, na ktorých je textúra (inými slovami typ povrchu), ktorá neobsahuje žiadne charakteristické body. Pre prevod *disparity*, teda nameraného horizontálneho posunu, na vzdialenosť musí byť známa vzájomná poloha kamier. Ak sa využíva stereo kamera (má často paralelné zobrazovacie roviny) vzájomná poloha dvoch pohľadov sa líši len posunom.



Obrázok 1. Význam horizontálneho posunu v stereo videní.

Základným výsledkom takýchto metód je získanie hĺbky alebo hĺbkovej mapy. Hĺbkovú mapu počítame v prípade, ak chceme mať predstavu o vzdialenostiach pre celú scénu. Hĺbková mapa predstavuje dvojrozmerný obraz, ktorý pre každý pixel pôvodného obrázku určuje vzdialenosť videného povrchu od pozorovateľa. Tento pojem je analogický s anglickými pojmami ako *Z-buffer* alebo *Z-depth*, ktoré sa využívajú v počítačovej grafike a označujú hĺbku [11].

Základné kroky, ktoré musia byť vykonané metódami, ktoré využívajú túto myšlienku sú nasledovné: rektifikácia obrázkov, nájdenie kľúčových bodov na obrázkoch, nájdenie korešpondujúcich párov, výpočet horizontálneho posunu *disparity*, výpočet hĺbky.



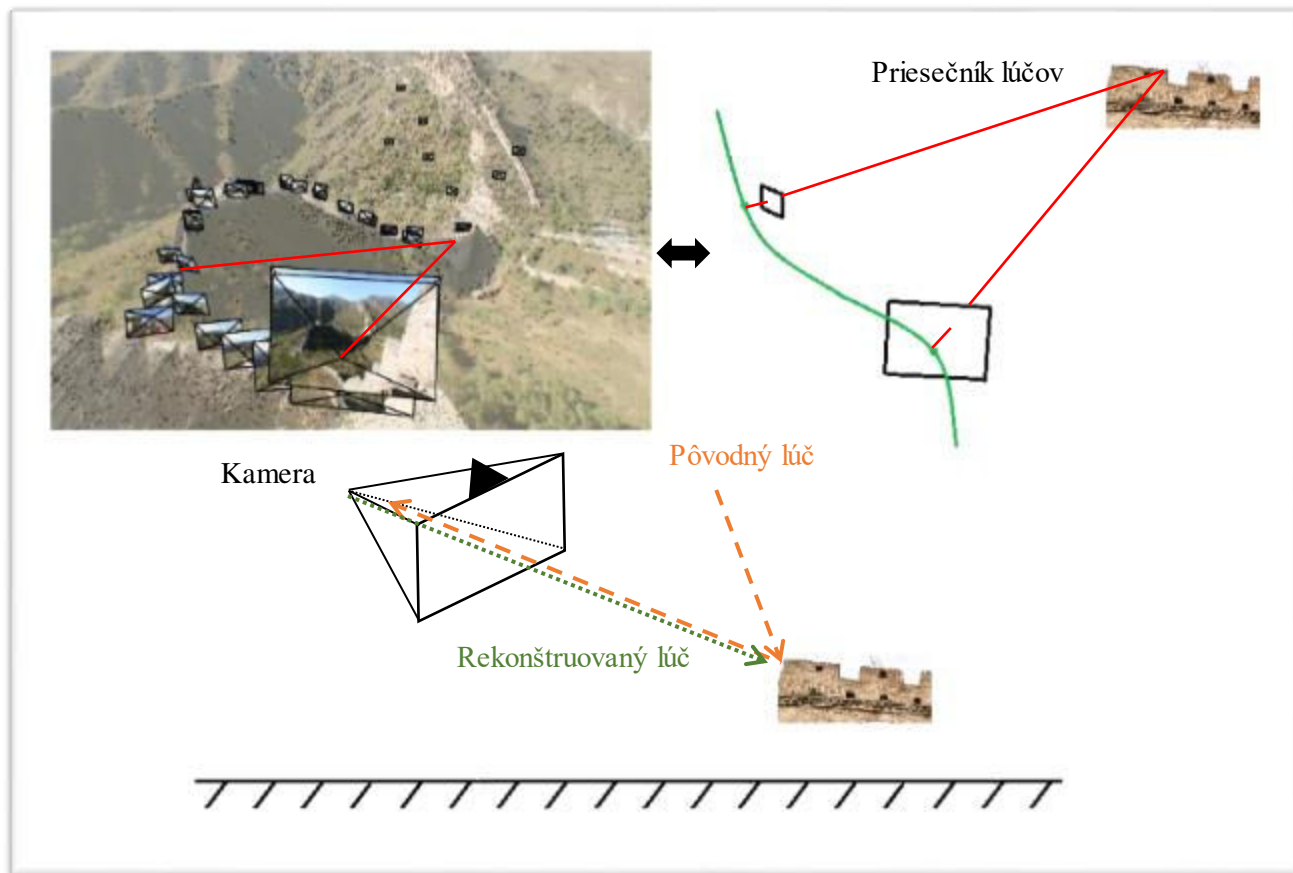
Obrázok 2. Základné kroky stereo korešpondencie



**Hľadanie spoločných bodov na obrázkoch a nájdenie priesečníka svetelných lúčov získaných z procesu spätnej projekcie** predstavuje ďalší zo spôsobov získania priestorovej informácie. Oba prístupy, tak ako aj predošlý, sú založené na triangulácii. Rozdiel je v tom, ako sa trianguluje. Tak ako v predošlom prípade aj tu je cieľom nájsť body na fotografii, ktoré sú spoločné. V tomto prípade sú kamery umiestnené ľubovoľne. Nie nutne vedľa seba. Pri hľadaní kľúčových bodov sa využívajú iné techniky ako pri hľadaní stereo kľúčových bodov. Naším cieľom nie je nájsť *disparitu*, pretože tá by vo všeobecnom prípade nedávala zmysel.

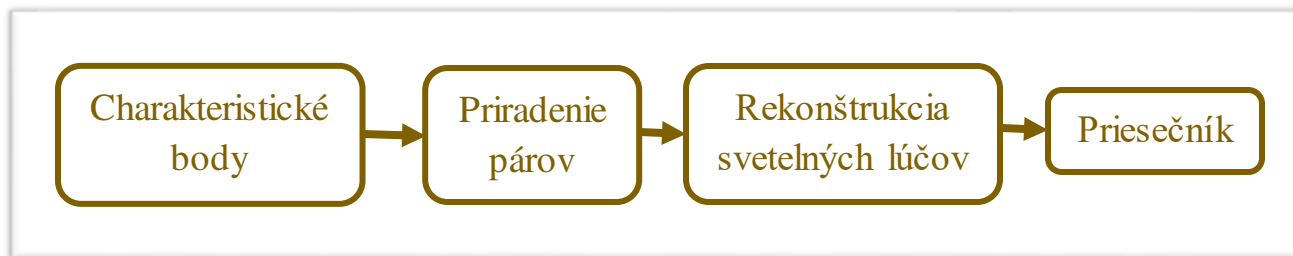
Obraz sa tvorí na zobrazovacom médiu v kamere tak, že naň dopadne svetelný lúč odrazený od povrchu predmetu. Naším cieľom je zrekonštruovať trajektóriu tohto lúča. Využitím matematického modelu kamery dokážeme spätne trasovať, kadiaľ lúč prechádzal. Vychádza pritom z mysleného optického stredu kamery, prechádza šošovkou a prostredím až narazí na povrch predmetu. Jeho ďalšie putovanie je takmer nemožné predpovedať. Nám však táto informácia postačuje na to, aby sme zistili, kde sa nachádza povrch predmetu, od ktorého sa naposledy odrazil. Potrebujeme k tomu ale ešte aspoň jeden lúč odrazený od tohto povrchu. Z fotografií najskôr získame korešpondujúce body. Prvým krokom je získanie charakteristických bodov. Na to sa využívajú rovnaké metódy ako v prípade stereo korešpondencie. Charakteristické body sú také, ktoré sú stabilné pri zmene pohľadu. Cieľom detektora charakteristických bodov je, aby našiel rovnaké body na fotografiách aj keď sú vytvorené z iného uhla [12]. Proces sa odlišuje od predošlej myšlienky stereo korešpondencie v tom, ako sa body párujú. Pri stereo korešpondencii sú obrázky zarovnané a stačí ich prehľadávať horizontálne. V tomto prípade sa využívajú iné postupy. Po nájdení párov sa pre každý bod z páru zrekonštruuje trajektória svetelného lúča a nájde sa priesečník dvoch lúčov v priestore, ktorý určuje polohu objektu.

Predpokladom pre využitie tohto prístupu je známa poloha a natočenie kamier. Poloha kamier sa dá aj odhadnúť metódami anglicky označovanými ako *structure from motion*. Problém je najmä zistenie presného natočenia a rotácie kamier. Sú teda potrebné vonkajšie parametre kamier. Rozšírenie na viacero kamier resp. pohľadov je pomerne jednoduché. Takéto riešenia sú pomerne presné, podliehajú však šumu a nedokonalostiam pri kalibrácii kamery alebo určovaní jej polohy z externého zdroja. Často sa pri praktických aplikáciách stáva, že pri pohybe dôjde k zmene rotácie kamery a tá nie je známa. Vtedy nie je možné tento spôsob využiť.



Obrázok 3. Ukážka rekonštrukcie svetelného lúča. Pre každú fotografiu je potrebné poznať polohu a natočenie kamery.

Základné kroky, ktoré musia byť vykonané: nájdenie charakteristických bodov na obrázkoch, nájdenie korešpondujúcich párov, rekonštrukcia trajektórie svetelných lúčov, nájdenie priesečníka lúčov.



Obrázok 4. Základné kroky rekonštrukcie svetelných lúčov

V oboch prístupoch, ktoré boli opísané, možno nájsť spoločnú časť a tou je detekcia charakteristických bodov na dvoch prípadne viacerých fotografiách a správne priradenie, ktorá časť z jednej fotografie odpovedá inej časti na druhej fotografii. Na to, aby sme mohli zrekonštruovať polohu nejaké predmetu, je táto informácia kľúčová a bez nej nie je možné pokračovať. Toto je minimálna sémantika obrázka, ktorej musí počítač rozumieť pri rekonštrukcii 3D scény.

Zo zadania máme k dispozícii anotáciu (inými slovami určité metadáta), kde sa objekt na obrázkoch nachádza. Ak použijeme túto informáciu a nepožadujeme na výstupe rekonštrukciu celej viditeľnej scény, ale iba tohto objektu, nie je potrebné, aby sme detekovali charakteristické body. Korešpondujúci pár pre objekt je už teda vytvorený anotáciami.

## 2.2 Teoretické východiská

Pre lepšie pochopenie načrtnutých prístupov a pre potreby implementácie je potrebné porozumieť technikám počítačového videnia, ktoré sa budú využívať. Jedným zo základných kameňov je matematický aparát na prácu s digitálnym obrázkom.

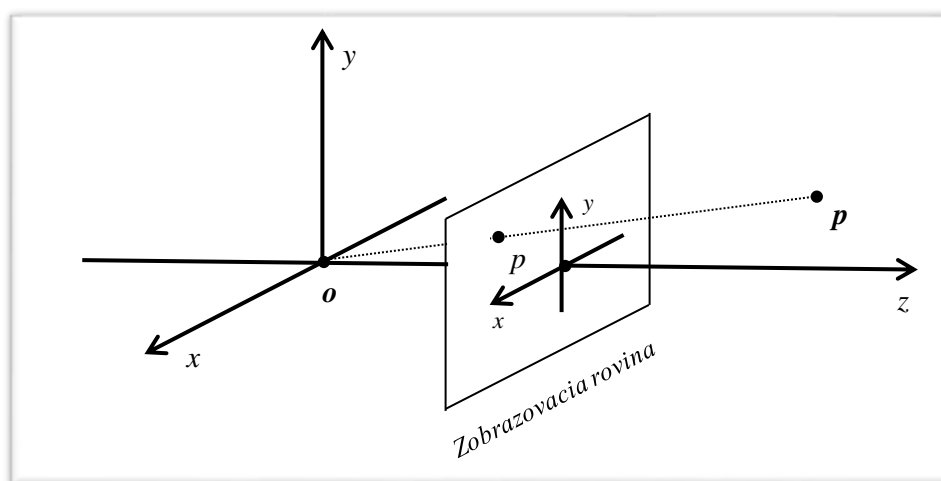
### 2.2.1 Proces zachytávania obrazu a geometria obrazu

Fotografia nemôže existovať bez svetla. Na vytvorenie fotografie musí byť scéna osvetlená aspoň jedným zdrojom svetla. Svetlo sa odráža od povrchu predmetov a prichádza do optiky kamery. Tam zvyčajne prechádza cez šošovky až kým nedosiahne zobrazovacie médium. To obsahuje mriežku s optickými snímacími senzormi, cez ktoré prechádza svetlo. Existujú dva základné typy senzorov označované ako CCD a CMOS. Tie zvyčajne prevedú svetlo na digitálne signály pre tri základné spektrálne farby (vlnové dĺžky): červenú R, zelenú G a modrú B. Jeden obrazový bod je teda tvorený trojicou čísel, ktoré reprezentujú tieto farby. Celú fotografiu tvoria trojice čísel pre každý obrazový bod mriežky. Zvyčajne sa interpretuje ako matica čísel, kde prvok matice predstavuje vektor. Ak zoberieme vždy jednu konkrétnu hodnotu vektora, dostaneme maticu, ktorá sa označuje ako kanál. Ďalšou analogickou interpretáciou je diskretná funkcia dvoch diskretných premenných, ktorej obor hodnôt je priestor vektorov. Po zachytení obrazu môže

nasledovať spracovanie fotografie (takzvaný *image post-processing*) na upravenie jej vlastností a prevod do štandardizovaného formátu. Fotografia je následne uložená v počítači v tvare súboru.

Rastrový obrázok má určité minimálne vlastnosti ako svoje rozmery, rozlíšenie, typ použitého farebného modelu a farebnú hĺbku. Rozlíšenie udáva hustotu obrazových bodov. Pri digitálnych obrázkoch sa často interpretuje ako rozmer, teda počet obrazových bodov. Farebný model určuje, akým spôsobom budú interpretované čísla uložené v matici na vytvorenie výsledného svetla. Najviac používané farebné modely sú RGB, HSI, HSL, YUV, CMY(K). Farebná hĺbka určuje počet bitov na zakódovanie jedného obrazového bodu. Napríklad pre farebný model RGB s 256 odtieňmi je farebná hĺbka 24 bitov.

Geometria kamery a proces vytvorenia digitálneho obrázku sú dôležité pre pochopenie ďalších myšlienok. Formovanie obrazu z trojrozmerného na dvojrozmerný priestor fotografie je z matematického pohľadu chápaný ako premietanie v deskriptívnej geometrii. Geometrické útvary sa zobrazujú na priemetňu. Z anglického slova *projection* sa označuje aj ako projekcia. Existujú viaceré spôsoby takejto transformácie. Nás však zaujíma konkrétne tá, ktorou dokážeme opísať fungovanie kamery. Takáto projekcia sa označuje ako perspektívna a je rovnaká, akú používa ľudské oko, preto je pre nás najintuitívnejšia. Pre perspektívnu projekciu je typické, že rovnobežné priamky (ako napríklad železničné koľaje) sú premietnuté ako priamky, ktoré sa stretávajú

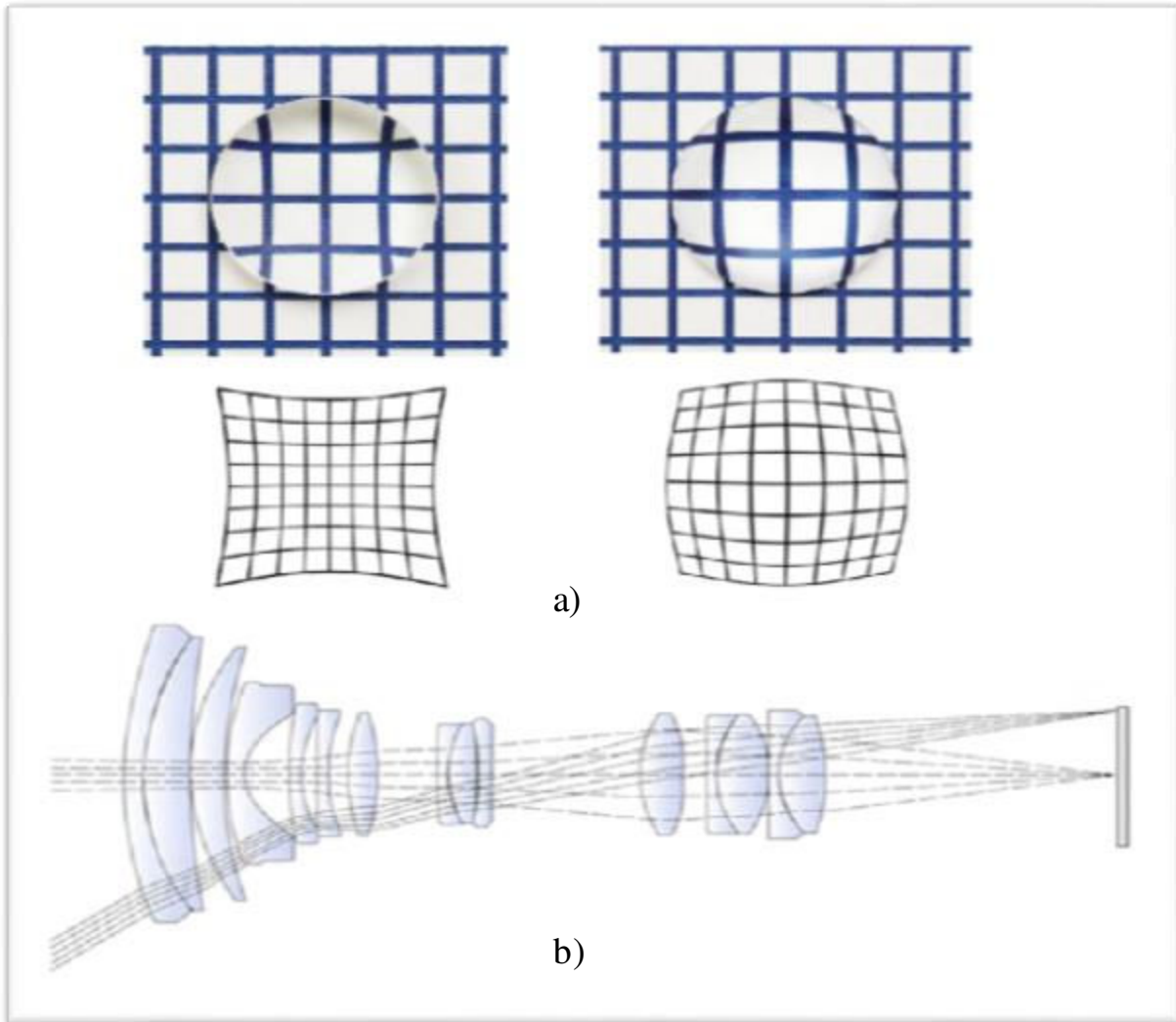


Obrázok 5. Pin hole kamerový model

v jednom bode na horizonte. Tento bod sa označuje ako úbežník, v angličtine označovaný ako *vanishing point*. Geometricky to možno znázorniť modelom kamery, ktorý sa v angličtine označuje ako *pin hole camera model*. Pri projekcii sa stráca jedna dimenzia obsahujúca informáciu o hĺbke.

## 2.2.2 Skreslenie spôsobené šošovkami

Pri reálnych kamerách nie je možné priemetnú postaviť pred šošovku, a preto je obraz zobrazený hore nohami. V súčasných kamerách je už obraz automaticky prevrátený. Prvé kamery fungovali bez šošovky a svetlo prechádzalo len cez úzky priezor. Výsledný obraz nebol ostrý, a preto sa využívajú šošovky, aby prijali viac svetla a koncentrovali ho. Pri štandardnom *pin hole camera* modeli, predpokladáme, že body premietame perspektívnou projekciou. Tá patrí medzi lineárne premietanie a priame čiary premietne na priame. Šošovka však so sebou prináša určité problémy. Nemusí nutne tento predpoklad dodržať. To závisí od použitých šošoviek a ich konfigurácie. Šošovky vo všeobecnosti produkujú chyby ako chromatická, sférická aberácia, koma, skreslenie a astigmatická chyba. Stručne ich možno opísať ako šum alebo nepresnosti v zobrazovaní. Väčšina z týchto nedokonalostí sa dá zredukovať premysleným systémom šošoviek v objektíve pri použití kvalitnej kamery. Pri menej kvalitných kamerách sme obmedzovaní presnosťou ich zobrazovania. Existujú aj techniky počítačového videnia, ktoré sa pokúšajú odstrániť takéto nedokonalosti úpravou zachytenej fotografie. Najčastejšie sa odhaduje skreslenie šošovky (anglicky označované ako *lens distortion*), ktoré najmä pri úlohách fotogrametrie spôsobuje najväčšie škody. Skreslenie spôsobuje najmä obvodové a radiálne zakrivenie, v angličtine označované ako *tangential* a *radial distortion*. Na zmenšenie vplyvu takéhoto zakrivenia sa využívajú matematické modely, ktoré upravia získaný obraz. Na získanie parametrov pre takéto modely sa využívajú rôzne prístupy. Zvyčajne sa odhadnú priamo pri procese kalibrácie kamery (bližšie je k téme kalibrácie uvedené v kapitole 2.2.4 Kalibrácia kamery).



Obrázok 6. Šošovky. a) Vankúšovité a sudovité skreslenie obrazu – *pincushion* a *barrel distortion* b) Systém šošoviek využívaný v kvalitnejších kamerách

### 2.2.3 Transformácie

Pri práci s literatúrou zaoberajúcou sa počítačovým videním sa človek často stretne s vyjadrením súradníc v takzvanom homogénnom tvare. Je to pridanie ďalšej dimenzie k vektoru, pre uľahčenie určitých výpočtov, pre rozšírenie na odpovedajúci počet stupňov voľnosti a najmä preto, aby boli výpočty lineárne. Na prevod do nehomogénneho tvaru treba vektor znormalizovať, čo v tomto prípade znamená predeliť rozšírenou zložkou (v príklade  $k$ ).

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \text{ bod v priestore} \quad \leftrightarrow \quad \tilde{\mathbf{x}} = \begin{bmatrix} kx \\ ky \\ kz \\ k \end{bmatrix} \text{ v homogénnom tvare}$$

Medzi základné priestorové transformácie patrí: translácia, rotácia, podobnosť, afinna transformácia a projekcia.

**Translácia** predstavuje posunutie objektu v priestore. Má 3 stupne voľnosti a zachováva priame a rovnobežné čiary, uhly, dĺžky a orientáciu.

$$\mathbf{x} = [\mathbf{E} \quad \mathbf{T}]_{3 \times 4} \tilde{\mathbf{x}} \quad \text{kde } \mathbf{E} \text{ je jednotková matica a } \mathbf{T} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \text{ so zložkami posunu}$$

**Rotácia** predstavuje zmenu orientácie objektu. Má 3 stupne voľnosti. Zachováva priame a rovnobežné čiary, uhly a dĺžky.

$$\mathbf{x} = [\mathbf{R}]_{3 \times 3} \mathbf{x} \quad \text{s dôležitými vlastnosťami } \mathbf{R}\mathbf{R}^{-1} = \mathbf{R}\mathbf{R}^T = \mathbf{E} \text{ a } |\mathbf{R}| = 1$$

**Rotácia a translácia**, označovaná v angličtine aj *3D rigid body motion* alebo *3D Euclidean transformation*, predstavuje posun objektu a zmenu jeho orientácie. Má 6 stupňov voľnosti. Zachováva priame a rovnobežné čiary, uhly a dĺžky.

$$\mathbf{x} = [\mathbf{R} \quad \mathbf{T}]_{3 \times 4} \tilde{\mathbf{x}}$$

**Podobnosť**, označovaná v angličtine *similarity*, predstavuje rotáciu a transláciu spojenú so zmenou dĺžok. Zachováva priame a rovnobežné čiary a uhly. Má 7 stupňov voľnosti.

$$\mathbf{x} = [s\mathbf{R} \quad \mathbf{T}]_{3 \times 4} \tilde{\mathbf{x}}$$

**Afinna transformácia** zachováva priame a rovnobežné čiary. Má 12 stupňov voľnosti.

$$\mathbf{x} = [\mathbf{A}]_{3 \times 4} \tilde{\mathbf{x}}$$

**Projekcia** zachováva iba priame čiary. Označovaná aj ako 3D perspektívna transformácia alebo 3D homografia. Má 15 stupňov voľnosti.

$$\mathbf{x} \xleftarrow{\text{nehomogénny tvar}} \tilde{\mathbf{x}} = [\mathbf{H}]_{4 \times 4} \tilde{\mathbf{x}}$$

Ak premietame bod medzi dvoma dvojdimenzionálnymi súradnicovými sústavami (rovinami) hovoríme o 2D homografii. Tá má 8 stupňov voľnosti a má mnohé praktické využitia. Využíva sa napríklad pri kalibrácii kamery.

## 2.2.4 Kalibrácia kamery

Pri formovaní obrazu je jedným z krokov vykonanie projekcie. Vo fyzických zariadeniach sa tento proces vykonáva za pomoci objektívu a dômyselného systému šošoviek a zobrazovania. Priestor sa premietne na zobrazovacie médium. Pre nás je dôležité, že z geometrického pohľadu je toto médium rovina. Body z tejto roviny sú namapované do bodov na fotografii. Jedná sa o 2D afínnu transformáciu. Jej parametre sa nazývajú aj vnútorné parametre kamery a sú veľmi dôležité pri ďalšom spracovaní obrazu. Má 6 stupňov voľnosti.

$$\mathbf{K} = \begin{bmatrix} fk_x & s & c_x \\ 0 & fk_y & c_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} fk & s & c_x \\ 0 & afk & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

kde  $s$  je zošikmenie vyjadrené v pixeloch  
 $f$  je ohnisková vzdialenosť  
 $c_x$  a  $c_y$  predstavujú optický stred fotografie vyjadrený v pixeloch  
 $a$  je pomer strán, anglicky *aspect ratio*  
 $k, k_x$  a  $k_y$  slúžia na prevod ohniskovej vzdialenosti do pixelov, určujú počet pixelov na jednotku v priestore (zvyčajne pixelov/milimetre)

Táto matica je často v literatúre uvádzaná tak, že namiesto prvku  $fk$  obsahuje len prvok  $f$ . V takom prípade je však ohnisková vzdialenosť vyjadrená v horizontálnych pixeloch. To vedie k zmätku, preto ohniskovú vzdialenosť vyjadrujeme v tejto práci len v jednotkách priestoru (povedzme v milimetroch). Hodnoty  $k_x$  a  $k_y$  slúžia na konverziu a predstavujú počet pixelov na jednotku priestoru v horizontálnom a vertikálnom smere. Dajú sa získať ako

$$k_x = \frac{w_i}{w_s}$$

$$k_y = \frac{h_i}{h_s}$$

kde  $w_i$  je počet pixelov horizontálne na obrázku  
 $h_i$  je počet pixelov vertikálne na obrázku  
 $w_s$  je šírka zobrazovacej roviny prislúchajúca obrázku v priestorových jednotkách  
 $h_s$  je výška zobrazovacej roviny prislúchajúca obrázku v priestorových jednotkách





presne kolmo na optickú os. V mnohých praktických aplikáciách nám postačuje stanoviť  $s = 0$ ,  
 $c_x = \frac{w_i}{2}$ ,  $c_y = \frac{h_i}{2}$ . [11]

Proces kalibrácie je taký, ktorý sa snaží odhadnúť vnútorné parametre kamery. Obrázok 7 zobrazuje ukážku kalibrácie. Pre čo najpresnejšie výsledky sa kamera kalibruje využitím fotografií so zosnímaným kalibračným vzorom. Takýto vzor má známy tvar a rozmery. Umiestni sa pred kameru a vytvorená fotografia, prípadne viacero fotografií (ak nie je kalibračný vzor trojrozmerný, je potrebné viacero pohľadov na vytvorenie homografie) sa analyzuje za účelom získania vnútorných parametrov. Proces kalibrácie môže byť spojený aj s odhadom parametrov skreslenia spôsobených šošovkami alebo iných parametrov ako sú vonkajšie parametre kamery vzhľadom na známu polohu kalibračného vzoru alebo iného objektu. Pre praktické kalibrovanie kamery možno použiť softvér distribuovaný s knižnicou OpenCV.

## 2.2.5 Triangulácia

Pred tým než sa pustíme do opisu základných spôsobov určovania polohy z kapitoly 2.1 Základné používané prístupy, pozrime sa akú spoločnú myšlienku tieto spôsoby majú. Touto spoločnou abstraktnou myšlienkou je triangulácia, na ktorej sú založené všetky pasívne binokulárne metódy určovania polohy. V anglickej literatúre označovaná ako *triangulation*.

Vo všeobecnosti je to určenie polohy tretieho bodu tak, že sa vytvorí trojuholník resp. nájde sa priesečník dvoch jeho strán. Na trianguláciu, môžeme využiť ľubovoľné vlastnosti obrazu tak, aby dokázali skonštruovať jednoznačne trojuholník. Tento trojuholník musí ležať na rovine, ktorá obsahuje hľadaný bod v priestore. Na definovanie roviny však potrebujeme ešte aspoň dva body také, že spolu s hľadaným bodom neležia na jednej priamke. Zvyčajne sa využíva poloha kamery pri oboch pohľadoch. Ak chceme zistiť len vzdialenosť od objektu, tak polohy kamery ani nepotrebujeme vedieť. Keďže polohu hľadaného bodu ešte nepoznáme, musíme zostaviť taký systém rovníc, to znamená určiť také vlastnosti scény, z ktorých dokážeme nájsť polohu hľadaného bodu. Najintuitívnejšie je využitie polôh kamier a uhlov, pod ktorým hľadaný objekt vidno.

## 2.2.6 Stereo korešpondencia

Cieľom stereo vizuálnych systémov je rekonštrukcia 3D scény založená na *parallaxe*. Tak ako už bolo spomínané (viď kapitolu 2.1), je to jav, pri ktorom sa nám predmety v rôznych vzdialenostiach od pozorovateľa premietnu inak v každom oku. Napríklad pri cestovaní autom sa predmety bližšie pohybujú rýchlejšie ako predmety vzdialené. Predmety na horizonte takmer nemenia svoju polohu. Rovnakú ideu využijeme pri získavaní hĺbkovej mapy.

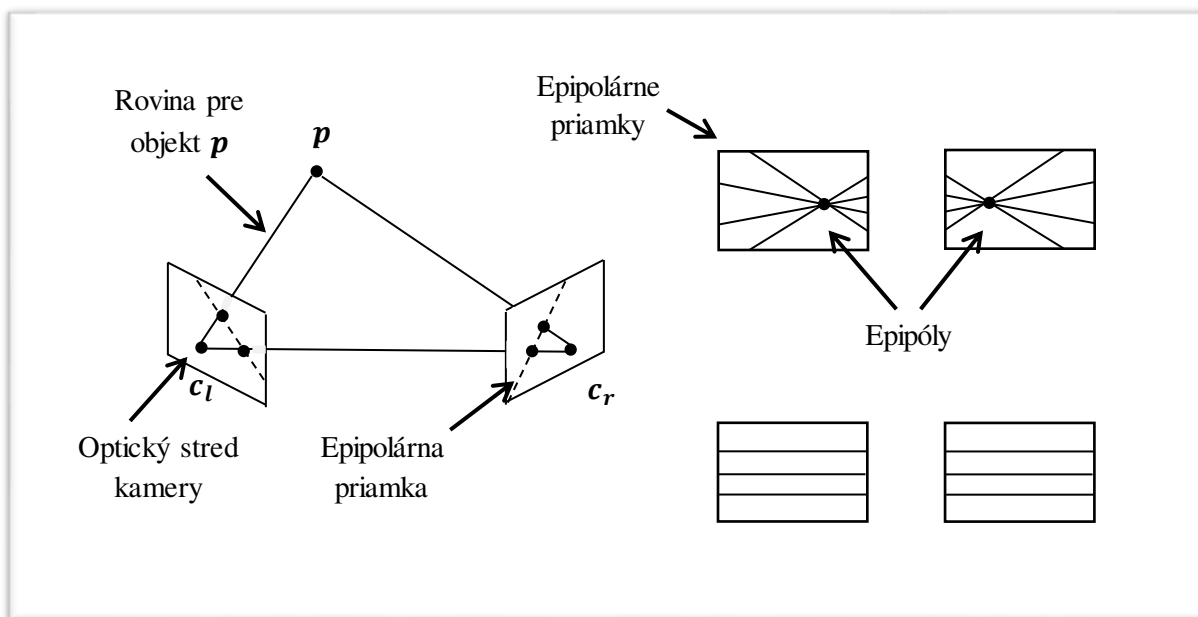
Prvý krok nielen pri stereo korešpondencii, ale aj pri iných prístupoch zaoberajúcich sa 3D rekonštrukciou, by malo byť kalibrovanie kamery. Najmä na zredukovanie skreslenia spôsobeného šošovkami, ale aj na odhadnutie vnútorných parametrov kamery. Pri redukovaní skreslenia sa snažíme aspoň o to, aby boli priame čiary priamymi v obrázku.

Tak ako už bolo vyššie spomínané základné kroky sú nasledovné: rektifikácia obrázkov, nájdenie charakteristických bodov na obrázkoch, nájdenie korešpondujúcich párov, výpočet horizontálneho posunu *disparity* a výpočet hĺbkky.

**Rektifikácia** fotografií znamená ich zarovnanie tak, aby sme pri hľadaní párov charakteristických bodov prehľadávali len jednorozmerný priestor. Stereo systém je geometricky opísaný *epipolárnou geometriou* a aby sme mohli vypočítať *disparitu*, je potrebné, aby boli kamery umiestnené rovnobežne a konfiguráciou sa líšili len posunutím. Ak kamery nie sú takto zarovnané, ale povedzme sú mierne vychýlené, dokážeme stále vytvoriť myšlenú kameru, ktorá bude tento predpoklad spĺňať.

Vytvoríme rovinu, ktorá je definovaná polohami kamier a polohami bodov, odpovedajúcich priemetom priestorového objektu na zobrazovacie roviny jednotlivých kamier (viď obrázok nižšie pre názornejšie vysvetlenie). Priesečník tejto roviny a ľubovoľnej zobrazovacej roviny je priamka, ktorá sa označuje ako *epipolárna priamka*.

Je dôležité si uvedomiť, že epipolárna priamka je závislá na rovine, ktorá je definovaná polohou konkrétneho objektu v priestore. Takže pre každý bod v priestore dostaneme inú rovinu a inú epipolárnu priamku. Samozrejme pre body ležiace na tých istých rovinách dostaneme

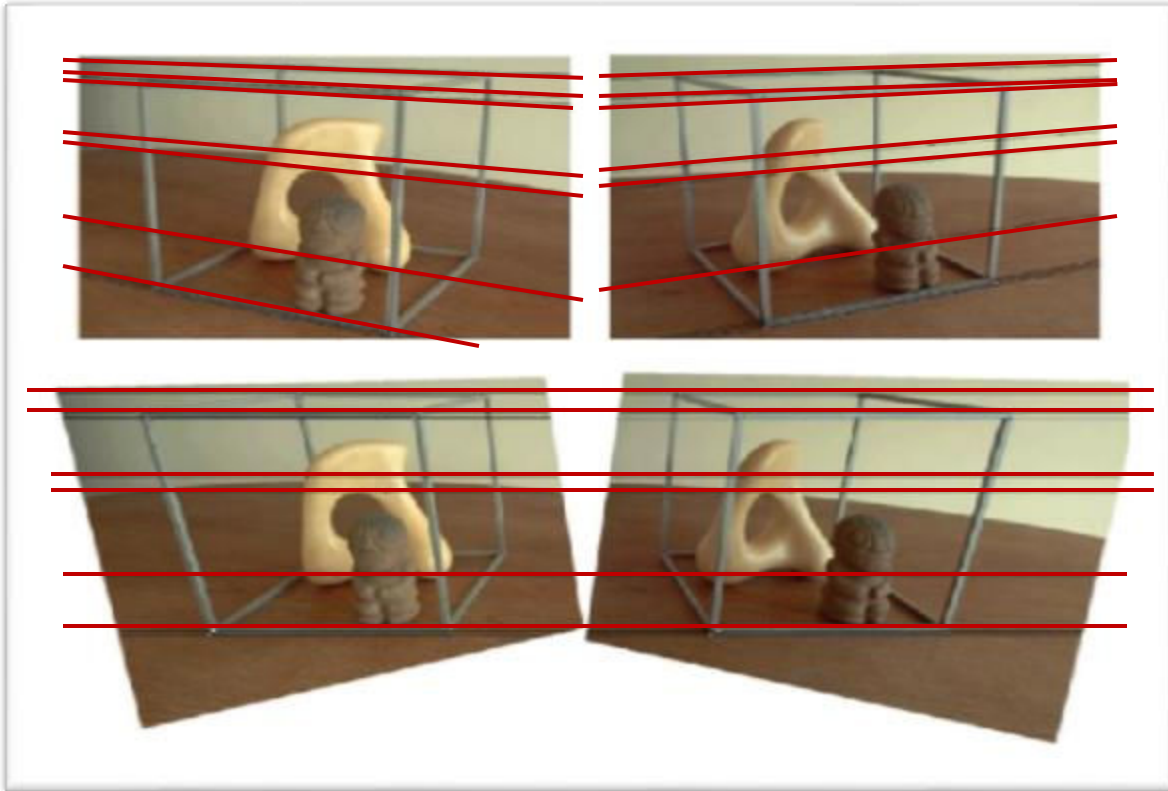


Obrázok 8. Epipolárna geometria

rovnaké epipolárne priamky, to čo chceme zdôrazniť je, že poloha objektu určuje epipolárnu priamku. Ak by sme pre všetky body priestoru našli epipolárne priamky, tie by sa pretínali v jednom bode. Tento bod sa označuje ako *epipól*.

Pre získanie konfigurácie kamier, ako bolo vyššie spomenuté, je našim cieľom z geometrického hľadiska, aby sa epipolárne priamky na zobrazovacej rovine nepretli. V literatúre sa to často interpretuje tak, aby sa epipolárne priamky pretli v nekonečne. Výsledkom teda bude obraz, ktorý sa bude javiť, ako keby bol zosnímaný kamerami, ktoré sú umiestnené rovnobežne vedľa seba. Najdôležitejšia je vlastnosť, že epipolárna priamka v jednom obrázku odpovedá epipolárnej priamke rovnakej roviny na druhom. Takýto tvar sa označuje ako *stereo v kanonickom tvare*. Ak teda budeme prechádzať po riadku v jednom obrázku, mali by prislúchať rovnakým bodom objektu v druhom obrázku, ale z iného uhla.

Na takýchto obrázkoch sa **hľadajú a párujú charakteristické body**. To predstavuje samostatnú výzvu, pretože na predmet sa pozeráme z iného uhla a môže byť čiastočne prekrytý. Metódy možno rozdeliť do dvoch skupín: lokálne a globálne. Lokálne berú do úvahy podmienky, založené na blízkom okolí bodov, ktoré nás zaujímajú. Globálne využívajú body pozdĺž celej



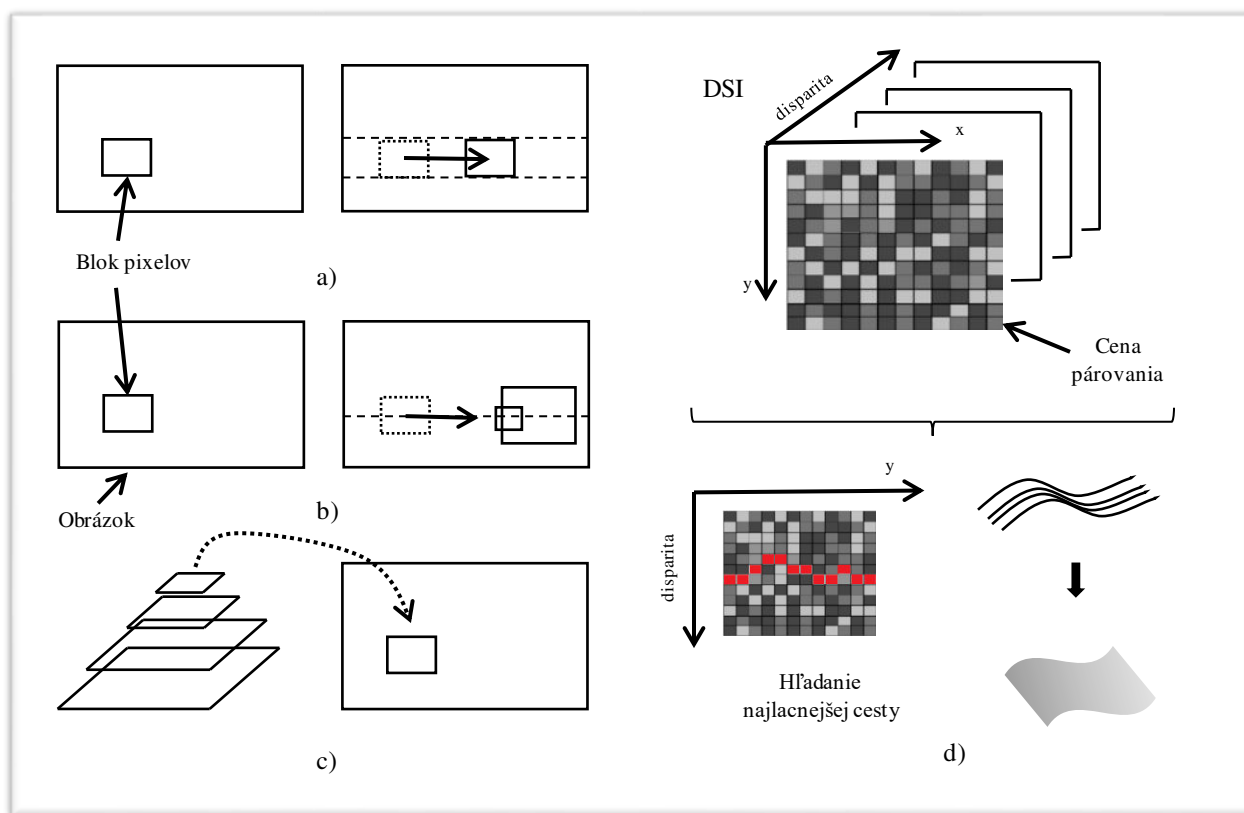
Obrázok 9. Rektifikácia fotografií so zobrazenými epipolárnymi priamkami

epipolárnej priamky alebo obrázka. Príkladom lokálnych metód sú párovanie určitého bloku pixelov, označované ako *block matching*. Príkladom globálnych je vyžitie dynamického programovania alebo rezov v grafe.

*Block matching* algoritmy sú založené na porovnávaní bloku pixelov s regiónom blokov v druhom obrázku. Pri týchto metódach je dôležitá metrika, s ktorou sa bloky porovnávajú. Tieto metódy narážajú na problém určenia veľkosti bloku. Je potrebné určiť kompromis: čím väčší blok tým väčšia spoľahlivosť, že to je ten správny bod a čím menší blok, tým väčšia presnosť lokalizácie bodu. Riešeniami sú adaptívne prispôsobovanie veľkosti blok a hierarchické prístupy. Hierarchické prístupy vykonajú prehládávanie s viacerými veľkosťami blokov a následne analyzujú získané páry na rôznych úrovniach od tých najspoľahlivejších k tým najpresnejším. [14]

Pri globálnych metódach sa často využíva priestor s cenou párovania označovaný ako *disparity space image* DSI. Pre cenu párovania sa využívajú podobné metriky ako pre *block*

*matching*. Pre každú možnú hodnotu disparity vypočítame cenu párovania pre celý obrázok. Pre jednu hodnotu disparity  $d$  dostávame trojrozmerný priestor, ktorý pre pixel  $(x,y)$  určuje cenu párovania s pixelom  $(x,y + d)$  v druhom obrázku. Pre všetky hodnoty dostaneme štvorrozmerný priestor. Optimalizačná úloha je formulovaná ako nájdenie plochy v takomto priestore. Teda pre pixel  $(x,y)$  prehladáme všetky ceny pre všetky hodnoty disparity a vyberieme jednu. Disparitu sa snažíme vyberať tak, aby cena výslednej plochy bola nízka a bola čo „najhladšia“. Techniky založené na dynamickom programovaní rozdeľia problém na menšie podčasti. Pri dynamickom programovaní a pri technikách rezov grafom sa vytvoria trojrozmerné diskretné priestory ale teraz iným spôsobom. Pre každý pixel v riadku a pre všetky hodnoty disparity vypočítame cenu párovania. Takýto diskretný priestor sa dá chápať ako graf, ktorom sú vrcholy všetky diskretné

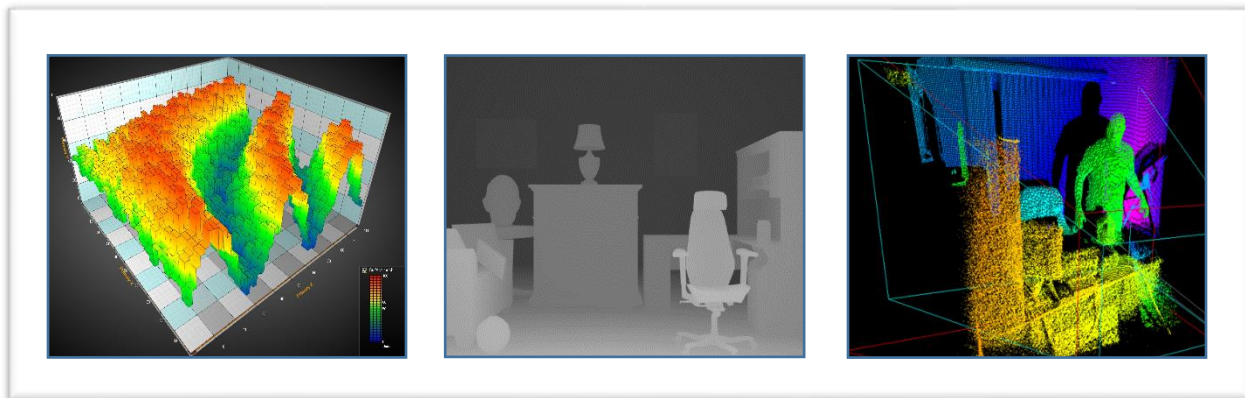


Obrázok 10. Hľadanie a párovanie charakteristických bodov. a) Základná technika *block matching-u* b) Rozšírenie s adaptívnou veľkosťou c) Rozšírenie s hierarchiou veľkostí bloku d) 4D priestor s cenou párovania DSI a vytváranie plochy, ktorá určuje párovanie

hodnoty a existuje 8 hrán pre každý pixel s okolitými pixelmi. V takomto priestore nájdeme cestu s najnižšou cenou takú, že prechádza cez všetky pixely riadku a obsahuje každý pixel práve raz.

V texte pod získavaním a párovaním charakteristických bodov nechápeme striktné využitie detektora na hľadanie charakteristických bodov, ale aj iných detektorov a techník trasovania napríklad založených na extrakcii farby a pod. Vo všeobecnosti je výsledkom nájdenie objektu na oboch obrázkoch.

Po nájdení charakteristických bodov a ich párovaní je cieľom **získanie disparity**. Teda počet pixelov deliacich rovnaký predmet v ľavom a pravom obrázku. Už s takouto informáciou dokážeme vizualizovať hĺbku 3D scény. Nedokážeme však presne povedať vzdialenosť k predmetu. Prečo? Disparita nám určuje len horizontálny posun a na prevod na hĺbku potrebujeme poznať ešte ďalšie parametre. Pri vizualizácii sa využíva fakt, že čím väčšia disparita, tým je objekt bližšie k pozorovateľovi a čím menšia, tým ďalej. Väčšinou sa obrázok s disparitami jednoducho prevedie na čiernobiely prípadne sa určí farebná hĺbková škála.



Obrázok 11. Ukážky rôzneho spôsobu vizualizácie disparity mapy

**Výpočet hĺbky.** Ak chceme získať naozajstnú hĺbku a nie len bezjednotkovú vizualizáciu je potrebný ďalší krok. Dve rovnobežné kamery sa nachádzajú vedľa seba pričom ich poloha sa líši len o transláciu  $B$ . Nech má objekt súradnice  $(X, Y, Z)$ . Priemet objektu na ľavej kamere má súradnice  $(x_l, y)$  a na pravej  $(x_r, y)$ . Disparita  $d = x_l - x_r$ . Využijeme tu základný princíp z počítačovej grafiky: posunutie kamery o  $B$  doprava je to isté, ako keď kamera ostane statická

a priestor posunieme doľava o  $B$ . Dostaneme nové súradnice objektu  $(X - B, Y, Z)$ . Z pin hole kamerového modelu dostávame:

$$\begin{array}{l} \text{Ľavá} \\ \text{kamera} \end{array} \quad \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \xrightarrow{\text{projektované na}} \begin{bmatrix} f \frac{X}{Z} \\ f \frac{Y}{Z} \end{bmatrix}$$

$$\begin{array}{l} \text{Pravá} \\ \text{kamera} \end{array} \quad \begin{bmatrix} X - B \\ Y \\ Z \end{bmatrix} \xrightarrow{\text{projektované na}} \begin{bmatrix} f \frac{X - B}{Z} \\ f \frac{Y}{Z} \end{bmatrix}$$

Pre hĺbku teda platí:

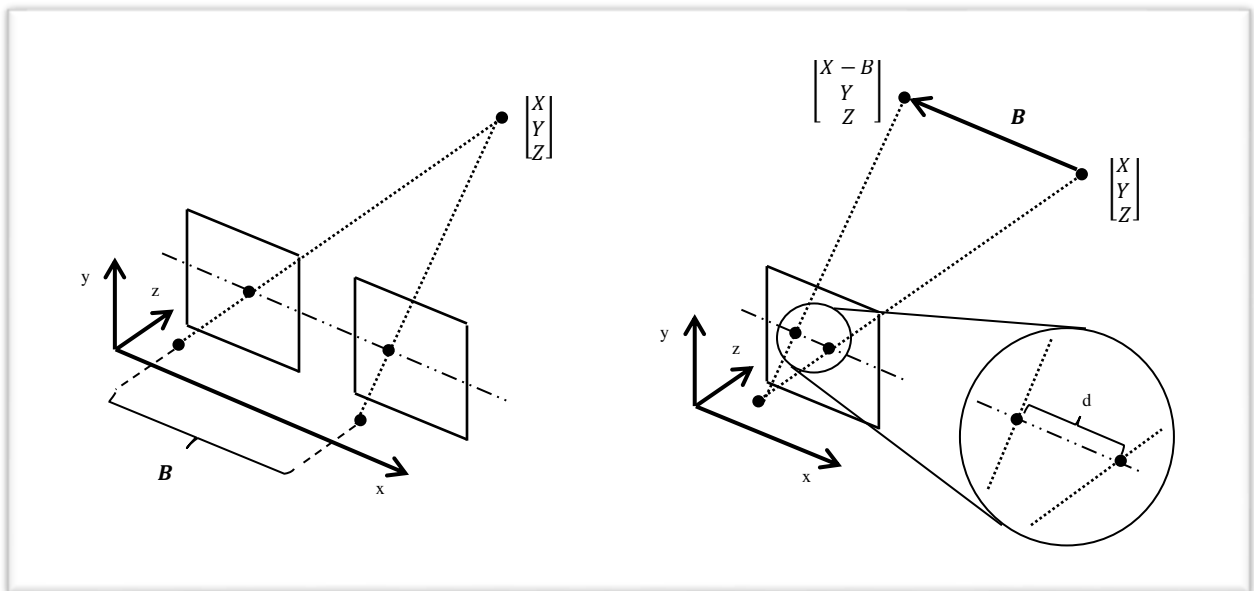
$$d = x_l - x_r = f \frac{X}{Z} - f \frac{X - B}{Z} = f \frac{B}{Z} \Rightarrow Z = f \frac{B}{d}$$

kde  $f$  je ohnisková vzdialenosť kamier

$B$  je translácia medzi kamerami, anglicky označovaná ako *baseline*

$d$  je hodnota disparity

Pre lepšie pochopenie je geometria znázornená na obrázku.

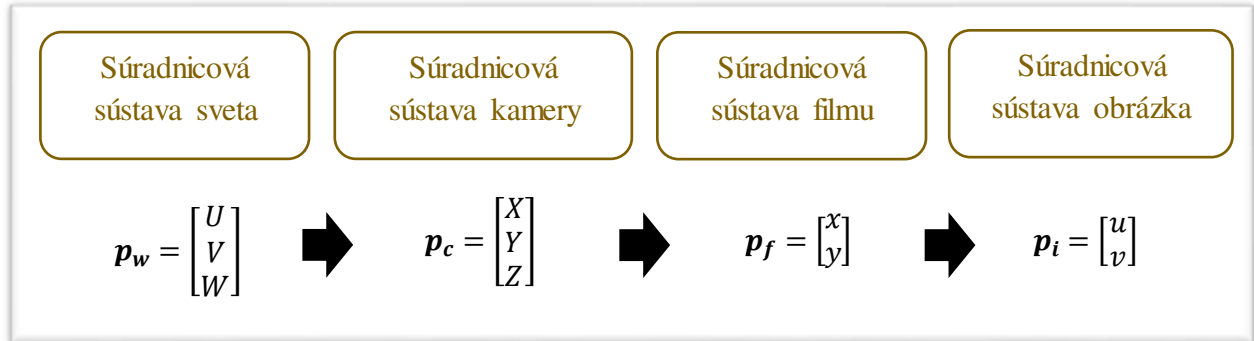


Obrázok 12. Vzťah medzi hĺbkou a disparitou



## 2.2.7 Rekonštrukcia trajektórie svetelných lúčov

Pozrime sa z geometrického pohľadu na proces tvorenia fotografie. Pri procese formovania obrazu sa premieňa medzi 4 súradnicovými sústavami a využívajú sa vyššie spomínané transformácie. Poloha kamery a objektu sú dané súradnicami v priestore s daným počiatkom a smerovaniami 3 základných osí.



Obrázok 13. Proces *forward* projekcie

Prvým krokom je premietnutie objektu do súradnicovej sústavy kamery. Tá má počiatok v optickom strede kamery s osami  $X$  a  $Y$  rovnobežnými so zobrazovacou rovinou a osou  $Z$  kolmou na ňu tak, aby to zodpovedalo kamerovému modelu *pin hole*. Tento krok sa označuje ako transformácia zo súradníc sveta do súradníc kamery, budeme ich preto označovať ako (*súradnicová*) *sústava sveta* a (*súradnicová*) *sústava kamery*. Jedná sa o 3D Euklidovskú transformáciu, teda rotácia a translácia. Parametre tejto transformácie sa označujú ako vonkajšie parametre kamery, v angličtine *extrinsic camera parameters*.

$$\mathbf{p}_c = \mathbf{R}(\mathbf{p}_w - \mathbf{C})$$

kde  $\mathbf{p}_c$  a  $\mathbf{p}_w$  označuje rovnaký fyzický bod (v tomto prípade objekt), vyjadrený v dvoch súradnicových sústavách, porade kamery a sveta <sup>1</sup>

$\mathbf{C}$  predstavuje počiatok sústavy sveta.

<sup>1</sup> V literatúre sa stretne aj s označením  ${}^c\mathbf{p}$  alebo  ${}^w\mathbf{p}$  pre vyjadrenie toho, v akej súradnicovej sústave sú príslušné koordináty

Po roznásobení a v maticovom tvare: <sup>2</sup>

$$\mathbf{p}_c = \mathbf{M}_{ext} \mathbf{p}_w = \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{p}_w$$

Ďalším krokom je prevod zo súradnicovej sústavy kamery na zobrazovaciu rovinu tak, ako to zobrazuje *pin hole* kamerový model. Označuje sa ako (*súradnicová*) *sústava (kamerového) filmu*. Koordináty sú stále vyjadrené v priestorových jednotkách. Túto transformáciu opisuje projekcia. Práve pri tejto transformácii sa stráca priestorová informácia. K projekčnej matici neexistuje inverzná matica, teda z pixelu na fotografii nedokážeme získať pôvodný bod v priestore.

$$\mathbf{p}_f = \mathbf{M}_{proj} \mathbf{p}_c = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{p}_c$$

Pre získanie diskretných hodnôt pixelov zo zobrazovacej roviny, inak povedané zo súradnicovej sústavy filmu, je potrebná ešte jedna transformácia. Tá je opísaná 2D afinnou transformáciou. Jej parametre sa zhodujú s vnútornými parametrami kamery.

$$\mathbf{p}_i = \mathbf{M}_{aff} \mathbf{p}_f = \begin{bmatrix} k & 0 & c_x \\ 0 & ak & c_y \\ 0 & 0 & 1 \end{bmatrix} \mathbf{p}_f$$

Pre získanie nehomogénnych súradníc pixelov treba

$$\mathbf{p}_i = \begin{bmatrix} x \\ y \\ w \end{bmatrix} \xrightarrow{pixel} \begin{bmatrix} \left[ \frac{x}{w} \right] \\ \left[ \frac{y}{w} \right] \end{bmatrix}$$

Tento proces sa anglicky označuje ako *forward projection*. Dá sa zapísať v podobe jednej transformačnej matice (konkrétne projekčnej):

$$\mathbf{p}_i = \underbrace{\mathbf{M}_{aff} \mathbf{M}_{proj}}_{\mathbf{K}} \mathbf{M}_{ext} \mathbf{p}_w$$

---

<sup>2</sup> Treba si uvedomiť, že  $\mathbf{T} = -\mathbf{RC}$

Cieľom pri rekonštrukcii trajektórie svetelného lúča je zistenie odkiaľ lúč do kamery priletel. Svetlo sa šíri priamočiarno (všeobecne známe z fyziky [15]) až kým nenarazí na povrch alebo niečo, čo spôsobí jeho ohyb alebo odraz. Za predpokladu, že sme v priestore naplnenom vzduchom a svetlo sa po poslednom odraze už neohýba, letí priamo do objektívu kamery. Túto trajektóriu môžeme opísať ako priamku v priestore. Potom nám postačujú dva svetelné lúče a ich priamky na nájdenie bodu, z ktorého sa v priestore odrazili. Ak uvažujeme o *pin hole* kamerovom modeli, z ľubovoľného bodu na povrchu predmetu sa do kamery dostane práve jeden svetelný lúč, preto potrebujeme na 2 svetelné lúče odrazené z toho istého bodu aspoň dve kamery.

Ako teda získať rovnicu priestorovej priamky, opisujúcej trajektóriu svetelného lúča? Ideálne by bolo, keby sme mohli nájsť inverznú maticu k projekčnej matici a prenásobiť ňou pixel na obrázku a získať tak priestorový bod. Inverzná matica však neexistuje. Hĺbka objektu sa stratila pri procese projekcie. Preto nedokážeme jednoznačne určiť priestorový bod, ktorý sa pôvodne projektoval – percepčný problém vnímania hĺbky jedným okom/kamerou. Na vyriešenie tohoto problému však máme všetky potrebné údaje ukryté v 2 kamerách. Na definovanie priestorovej priamky nám stačia dva priestorové body na nej ležiace. Jedným z nich je optický stred kamery. Druhý z nich je fyzické miesto na senzore kamery, kam lúč pri prechode kamerou dopadol. Ako tento bod zistiť? Jeho poloha závisí od polohy objektu vo fotografii. Následne ak poznáme parametre kamery, dá sa spätne tento bod zrekonštruovať. Potrebujeme teda zistiť bod na zobrazovacej rovine, ktorý prislúcha nejakému pixelu.

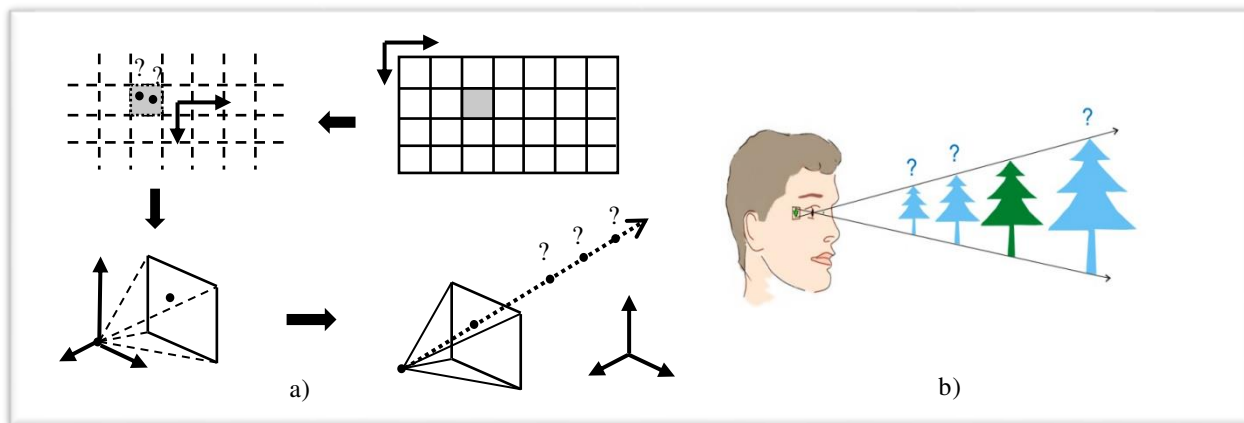
$$\mathbf{p}_i \rightarrow \mathbf{p}_f \xrightarrow{\text{zohľadnenie polohy kamery}} \mathbf{p}_w$$

Z matematického pohľadu

$$\mathbf{M}_{aff}^{-1} \tilde{\mathbf{p}}_i = \mathbf{p}_f = \begin{bmatrix} U \\ V \\ f \end{bmatrix}$$

$$\mathbf{M}_{ext}^{-1} \tilde{\mathbf{p}}_f = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Tento proces sa označuje ako spätná projekcia *backward projection*.



Obrázok 14. Spätná projekcia a problém nejednoznačnosti. a) Vizualizácia inverzných transformácií pri spätnej projekcii a ich potenciálnych chýb b) Problém vnímania hĺbky

Pracujeme samozrejme s diskretizovaným bodom, ktorý prevádzame do trojrozmerného priestoru, čo prináša určité chyby. Jedna oblasť aktívnej časti optického senzora určuje farbu pixelu. Táto oblasť má určité fyzické rozmery a zachytáva viacero lúčov. Rekonštrukciu lúču dokážeme spraviť len na úrovni pixelu a teda za všetky lúče patriace pixelu vyberieme len jedného reprezentanta. Aj malá chyba v odhade polohy bodu na zobrazovacej rovine môže viesť k veľkej chybe v priestorovom odhade.

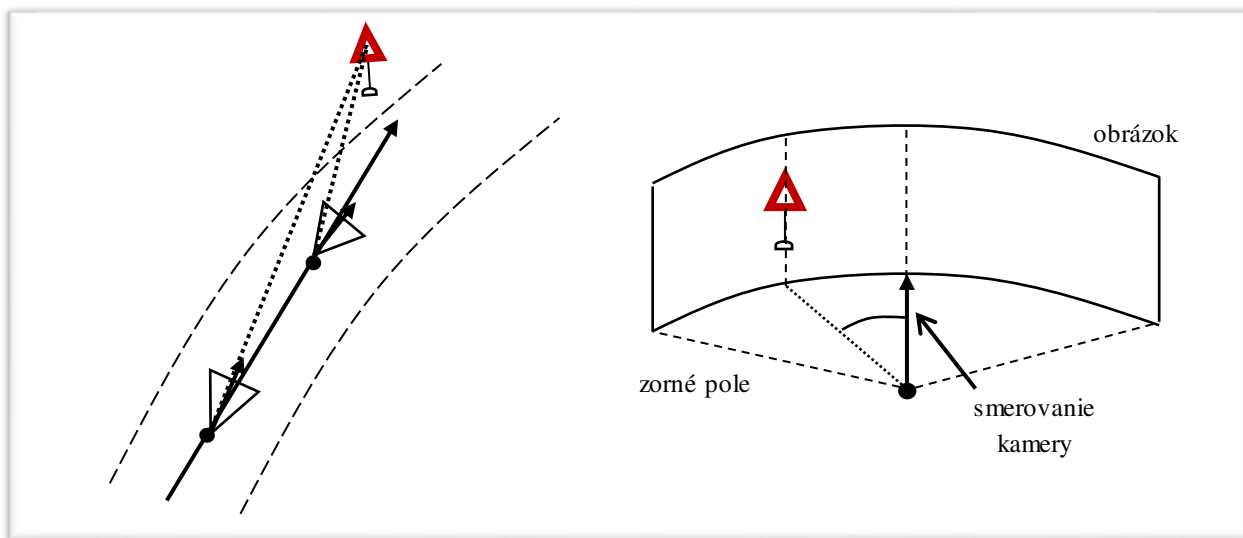
## 2.2.8 Triangulácia založená na pozorovaných uhloch k objektu

Navrhujeme ešte ďalší spôsob určovania polohy, ktorý bude následne implementovaný a experimentálne skúmaný. Ten vznikol od základov počas projektovej výučby a konzultácii k tejto práci.

Ak predpokladáme, že pracujeme s 2 kamerami a získali sme párovanie charakteristických bodov. Za predpokladu, že poznáme polohy kamier, ich smerovanie a zorné pole a kamery sa pohybujú po známej rovine, tak dokážeme určiť polohu objektu na rovine.

Polohy kamier sú premietnuté na mapu. Akoby pohľad z vtácej perspektívy. V tejto rovine budeme hľadať polohu objektu. Na výpočet využijeme trianguláciu tak, že zostrojíme trojuholník. Jeho dva vrcholy sú tvorené polohami kamier premietnuté na rovinu. Pre každú kameru poznáme

uhol, pod ktorým kamera vidí daný objekt. Pri odvodzovaní uhla využijeme zorné pole kamery a jej smerovanie. Uhol je daný horizontálnou polohou objektu na fotografii. Následne využitím sínusovej vety určíme dĺžku ľubovoľnej z neznámych strán, ktorá predstavuje hĺbku. Získanú hĺbku následne interpretujeme a prevedieme ju na presnú polohu objektu. Na to využijeme polohu kamery a jej smerovanie.



Obrázok 15. Triangulácia v rovine využitím uhlov a ich získavanie

Nevýhodou tohoto prístupu je, že požaduje, aby sa kamery pohybovali po rovine a zobrazovacie roviny kamier boli kolmé na túto rovinu.

### 3 Príprava dát

Pri vývoji metód počítačového videnia sa často stretávame s tým, že nevieme odhadnúť, či je nepresnosť spôsobená zlým postupom alebo nedokonalými dátami, s ktorými pracujeme. Pri technikách fotogrametrie, akými sú vo všeobecnosti merania priestoru založené na fotografiách, sú presnosť parametrov kamery a kvalita fotografie kľúčové pri overovaní správnosti, výpočte chyby navrhnutých metód a ich porovnávaní. Pri datasetoch z reálnej kamery vznikajú chyby spôsobené nepresnou kalibráciou, zlým odhadom parametrov, fyzikálnymi nedokonalosťami

šošovky, povahou scény. Takéto datasety potrebujeme okrem toho následne spracovať. Môže sa upraviť samotná fotografia, čo môže spôsobiť vznik určitých chýb. Ak využívame anotačný nástroj, manuálne alebo automatické metainformácie môžu obsahovať chybu. Častou je chyba pri označovaní bodov, ktoré patria objektu napríklad pri vytváraní takzvaného *bounding boxu*. Ani človek často nie je schopný označiť oblasť presne. Pri práci s GPS sú to chyby spôsobené nedostatočným signálom, odrazom signálu od budov a interferenciou s inými signálmi.

V ideálnom prípade by sme mali k dispozícii fotografie neovplyvnené skreslením šošovky s presnými anotáciami a dátami. Dokázali by sme potom lepšie analyzovať ako vzniká chyba, prípadne čo má na jej vzniku najväčší podiel. Ak by sme chceli získať reálne dáta s vysokou presnosťou, potrebovali by sme špecializované zariadenia. Bolo by to značne nákladné a stálo by nás to aj čas potrebný na nakalibrovanie, zozbieranie všetkých dát a ich spracovanie, čo je námet pre samostatný projekt. Zberom dát z kamery na vozidle sa zaoberá napríklad *Karlsruhe Institute of Technology* a *Toyota Technological Institute at Chicago* [16], na ktorom pracuje tím ľudí. Na vozidle je množstvo pomocných senzorov, ktoré sa využívajú na úpravu dát tak, aby dosahovali čo najväčšiu presnosť. Ideálne dáta je nákladné získať vlastnými prostriedkami, ďalšou možnosťou však je využiť dáta, ktoré sú voľne dostupné. Je potrebné preskúmať dáta, ktoré poskytujú rôzne zdroje na internete. Najväčším problémom sa nakoniec zdá byť nájsť dáta, ktoré odpovedajú situácii, ktorú chceme skúmať a poskytujú okrem testovacích dát aj dáta na overenie správnosti výsledku, označované ako *ground truth*.

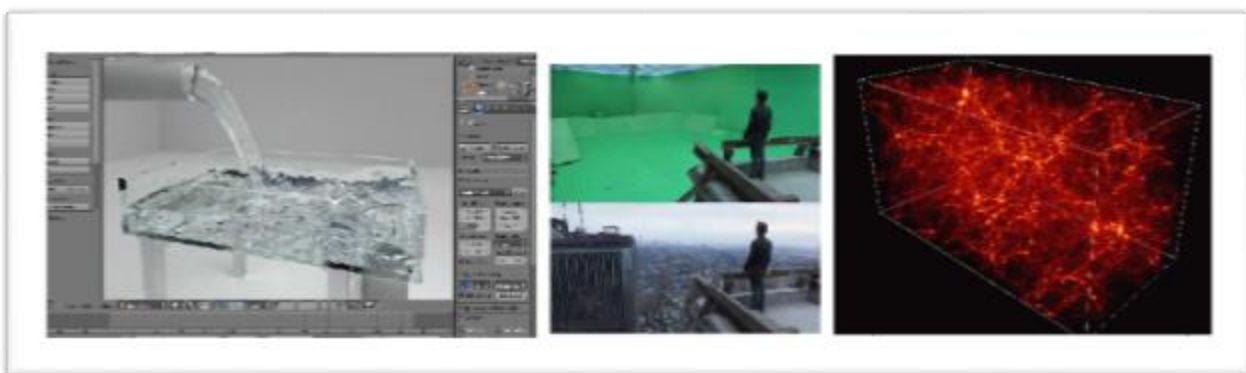
Takéto ideálne dáta však dokážeme vytvoriť. A nielen ideálne dáta, ale aj dáta ovplyvnené konkrétnym typom chyby. To nás motivovalo k vytvoreniu umelého datasetu. Teda vytvoriť kontrolované prostredie, v ktorom dokážeme s jemnosťou ovplyvňovať parametre scény. V takýchto laboratórnych podmienkach vytvoríme scénu, v ktorej abstrahujeme od nerelevantných častí a môžeme sa sústrediť na problém rekonštrukcie 3D scény. Dataset sme označili ako umelý, pretože bude obsahovať dáta, ktoré sú vygenerované a nie dáta z reálnej scény. Samozrejme tieto umelé dáta sa im budú v najväčšej možnej miere podobat'. Každá metóda môže využívať pri rekonštrukcii iný typ dát. Umelý dataset by mal obsahovať všetky potrebné údaje pre konkrétnu metódu. Základnými údajmi sú videosekvencia obrázkov s anotáciou objektu, parametre kamery

ako je rozlíšenie, ohnisková vzdialenosť a ďalšie parametre kamery a scény, poloha a smerovanie kamery a pre potreby overenia správnosti poloha objektu.

Kritéria, ktoré by mal umelý dataset spĺňať: obrázky by mali odpovedať fotografiám z reálnej scény, parametre scény, kamery a anotácie majú dostatočnú presnosť, je možné ovplyvniť dáta určitým typom chyby a mali by existovať scény pre základné situácie, ktoré môžu nastať.

Tak ako bolo v úvode spomenuté, úlohu konkretizujeme na príklade pohybujúceho sa vozidla, na ktorom sú umiestnené kamery. Objekt, ktorého polohu chceme zistiť, je dopravná značka umiestnená pri ceste. Na vytvorenie umelého datasetu je potrebné zvoliť vhodný nástroj na modelovanie scény. Nástroj by mal byť schopný vytvoriť obrázok danej scény so zvolenou konfiguráciou kamier a ich parametrami. Existuje len zopár vhodných nástrojov ako Blender, Autodesk 3ds Max, Modo, K-3D. Naším cieľom nie je porovnávať rôzne modelovacie nástroje. Jeden zo spomenutých nástrojov však prináša aj ďalšie benefity oproti ostatným.

Blender je softvérový nástroj na vytváranie 3D počítačovej grafiky. Je ho možné využiť na celú škálu úloh od filmového priemyslu, vývoja počítačových hier až po simuláciu fyzikálnych experimentov. Práve tým, že tento nástroj obsahuje podporu pre prácu s fyzikálnymi zákonitosťami a poskytuje detailné nastavenia pre scénu, kamery a ich správanie sa, je pre nás vhodným nástrojom na modelovanie. Navyše poskytuje prístup k vlastnému jadrú za použitia Python API, má veľkú komunitu ľudí, rozsiahlu prehľadne spracovanú dokumentáciu a rôzne návody, videotutoriály a diskusné fóra dostupné na internete nielen od členov komunity. Tento



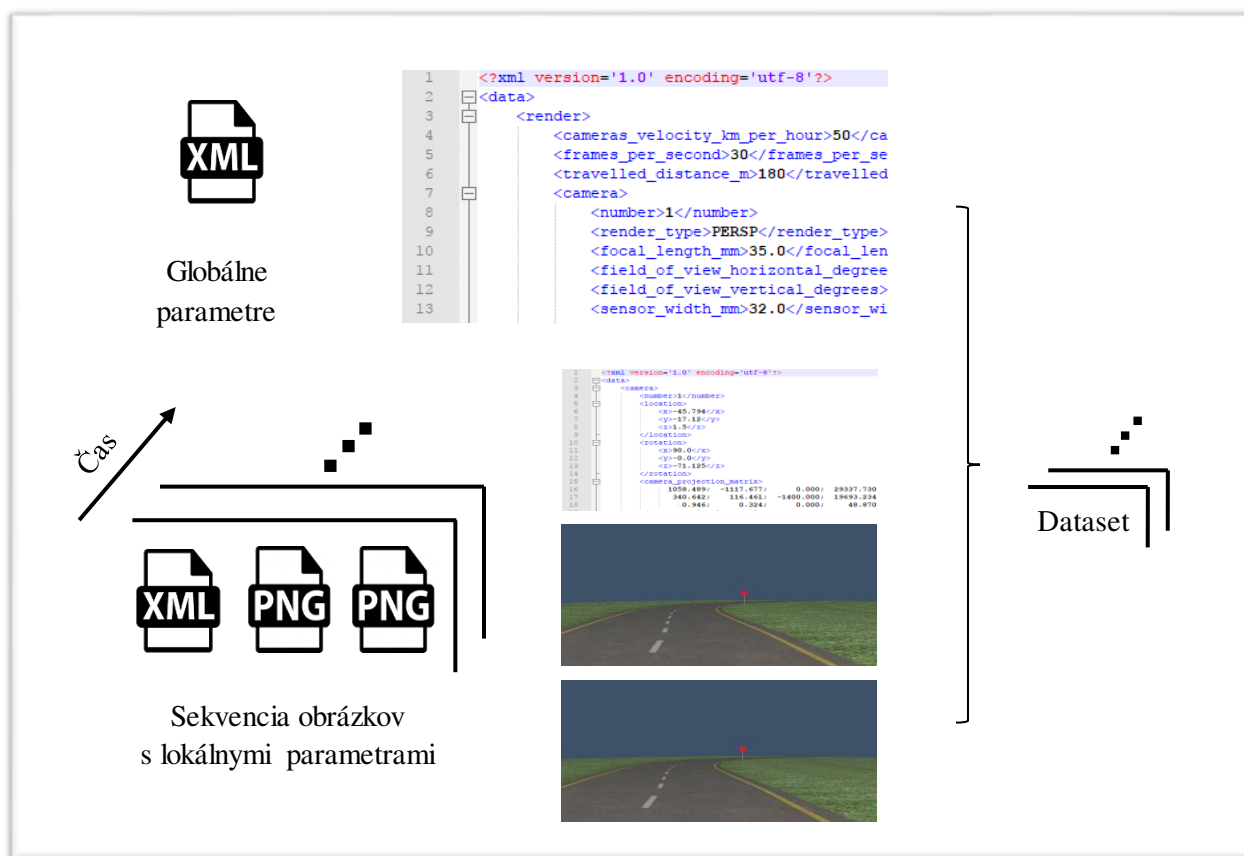
Obrázok 16. Nástroj Blender. Postupne zľava: prostredie nástroja, využitie pri CGI vo filmoch, fyzikálne experimenty.

nástroj je voľný *open-source* projekt. Bol využitý pre tvorbu efektov do filmov ako *Spider-Man 2*, *Captain America* a iné. Dokonca americká vesmírna agentúra NASA využíva tento nástroj.

Prípravu dát možno rozdeliť do 4 krokov: modelovanie scény, generovanie dát, detekcia a oprava anotácie a validácia.

**1. Modelovanie scény.** V prvom kroku je v modelovacom nástroji vytvorená priestorová scéna. Tá obsahuje pohyblivé kamery, cestu a dopravnú značku. Cesta môže mať rôzny sklon a tvar. Kamery sa pohybujú po vopred vytvorenej trajektórii opisujúcej tvar cesty.

**2. Generovanie dát.** Pri generovaní dát sa určí, s akými parametrami majú byť vytvorené dáta. Nastavia sa optické parametre kamier, rýchlosť ich pohybu a dĺžka trajektórie počas ktorej budeme zaznamenávať obraz. Zaznamenané sú globálne údaje pre celú scénu a údaje pre každý



Obrázok 17. Štruktúra datasetu pre jednu scénu s konkrétnymi parametrami







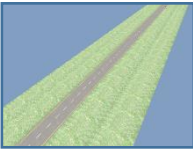

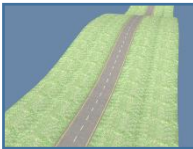

časový okamih videosekvencie. Videosekvencia je uložená v tvare postupnosti obrázkov. Údaje opisujúce scénu a kamery sú v XML súboroch s rovnakým názvom ako názov obrázka, ku ktorému prislúchajú.

**3. Detekcia a oprava anotácie.** Súbor s údajmi obsahuje okrem iného ohraničujúci obdĺžnik, ktorý obsahuje všetky pixely dopravnej značky. Často sa hovorí o *bounding box*-e dopravnej značky. Ak však dopravnú značku nevidno, napríklad je za kopcom alebo vplyvom otáčania vozidla, takýto *bounding box* neexistuje a chceme, aby sa vtedy v XML súbore nenachádzal. Pre jednoduchosť je dopravná značka červenej farby a pri detekcii sa hľadá práve táto farba na obrázku v takzvanom HSV farebnom modeli. Výsledkom detekcie je informácia, či sa na danom obrázku nachádza dopravná značka. Pripomínam, že cieľom práce nie je detekcia objektu, ktorého polohu hľadáme. Takáto anotácia už je vstupom do metód.

Pre účely experimentovania boli vytvorené datasety z viacerých namodelovaných scén. Tieto scény sa snažia pokryť vybrané situácie, v akých sa môže vozidlo na ceste nachádzať. Od priamej až po zakrivenú s rôznym sklonom. Z každej takejto scény sú následne vygenerované datasety s konkrétnymi parametrami scény a kamery. To znamená, že pre každý typ scény a každú sadu parametrov existuje jeden dataset. Zoznam namodelovaných scén so stručnou charakteristikou je v tabuľke Tabuľka 1. Použité typy kamier sú v Tabuľka 2.

**4. Validácia.** Keďže pracujeme s umelo vytvorenými dátami, ako overíme, že sú správne? Dáta v XML súboroch odzrkadľujú konkrétne umiestnenie a natočenie kamery spolu s ďalšími globálnymi údajmi. Tieto údaje prislúchajú ku konkrétnemu časovému okamihu vo videosekvencii. Pre dané údaje potrebujeme overiť, že vytvorený umelý obrázok je správny.

Za predpokladu, že využívame *pin hole* kamerový model, dokážeme z údajov zostrojiť projekčnú maticu a overiť, či sa objekt nachádza na obrázku na pozícii, na ktorej ho očakávame. Na tento účel bol vytvorený nástroj, ktorý rozpozná červenú plochu značky a podľa dát uložených v konfiguračných súboroch overí, či sa objekt nachádzať na obrázku na správnej polohe. Overujeme len správnosť pozície objektu na fotografii nie pozadia alebo cesty. To je dostatočné, pretože pre skúmané metódy je relevantná len poloha objektu. Všetky vygenerované datasety boli overené a sú validné. Fotografia skutočnej scény by obsahovala dopravnú značku na rovnakej

Označenie scény	Stručný opis	Vizuálna ukážka scény	
StraightFlat	Priama plochá cesta		
LeftFlat	Ľavotočivá plochá cesta		
RightFlat	Pravotočivá plochá cesta		
CurvedFlat	Plochá cesta so zákrutami		
StraightConstUp	Priama cesta s konštantným stúpaním		
StraightUp	Priama cesta s nekonštantným stúpaním		
StraightHill	Priama cesta cez kopec – so stúpaním a klesaním		
Hill	Cesta so zákrutami cez kopec – so stúpaním a klesaním		

Tabuľka 1. Zoznam modelovaných scén

Nastavenie kamery			
Typ	Ohnisková vzdialenosť	Veľkosť senzora	Rozlíšenie
Filmová kamera	35 mm	32 x 18 mm	Full HD 1920 x 1080
Smartfón kamera	3,7 mm	4,8 x 3,6 mm	VGA 640 x 480

Tabuľka 2. Zoznam kamier s parametrami

pozícii v obraze – pri rovnakých parametroch scény a kamier. To, čo odlišuje umelú fotografiu od tej skutočnej, sú vlastnosti ako textúra objektov, osvetlenie, prekrývanie predmetov, zaostrenie kamery, skreslenie šošovkami a iné.

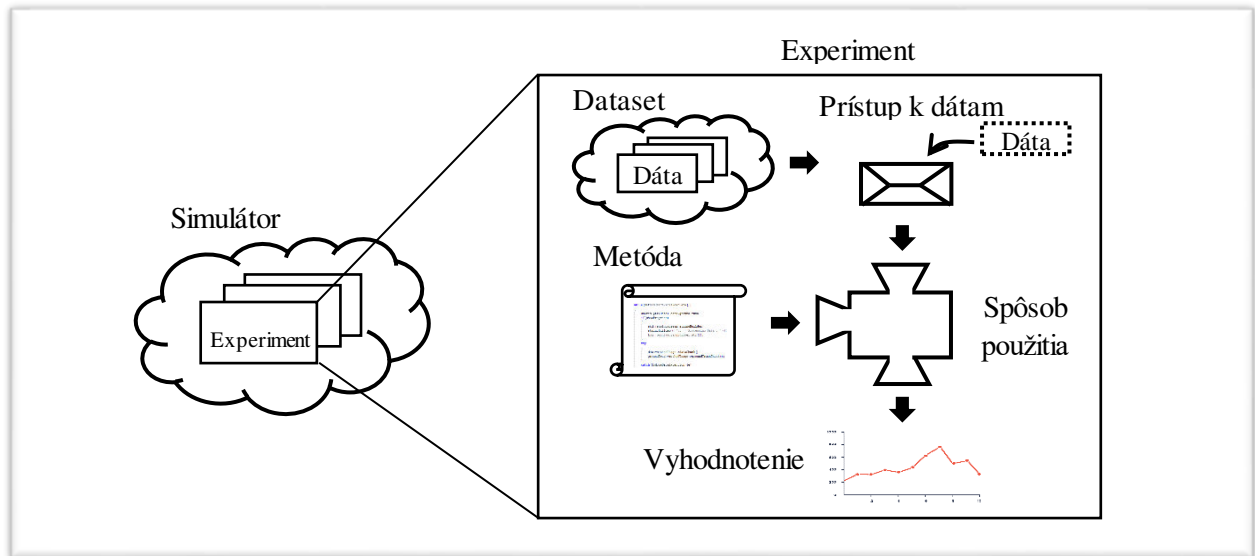
## 4 Implementácia simulátora experimentov a metód

---

Oblasť počítačového videnia, ktorá sa zaoberá rekonštrukciou 3D scény je rozsiahla. Je v nej zaužívané množstvo pojmov, s ktorými sa musí človek oboznámiť. Pri návrhu vlastných metód musí človek experimentovať a sformulovať myšlienku matematicky, aby ju bolo možné implementovať. Pred samotnou implementáciou je potrebné porozumieť problematike a samotnej teórii počítačového videnia. Existujú rôzne knižné a internetové zdroje, návody, diskusné fóra, zverejnené prednášky z iných univerzít a videokurzy, ktoré sú veľmi nápomocné. Často sú dostupné iba v anglickom jazyku. Naším cieľom je implementovať prostredie pre zadefinovanie experimentu a možnosť tento experiment vykonať – nazveme ho simulátor experimentov. Následne pre rôzne spôsoby určovania polohy objektu definovať experiment tak, aby dokázal pracovať s ľubovoľným datasetom.

Nad prichystanými dátami chceme vykonávať automatizované experimenty. Na tento účel implementujeme simulátor experimentov, ktorý umožní jednotným spôsobom pracovať s experimentom a jeho konfiguráciou bez ohľadu na to, akú metódu využívame, s akým typom dát pracujeme a akým spôsobom chceme výsledky vyhodnocovať. Jadro simulátora by malo byť navrhnuté univerzálne, aby ho bolo možné použiť pri ľubovoľných experimentoch z oblasti počítačového videnia. Experiment zapuzdruje dataset, nejaký algoritmus počítačového videnia, spôsob použitia algoritmu na datasete a vyhodnotenie výsledkov algoritmu.

V prípade určovania polohy objektu chceme, aby sa priebeh experimentu čo najviac približoval spracovaniu dát s reálnymi zariadeniami vo vozidle. Metóda má teda v určitom časovom okamihu k dispozícii len dáta od začiatku experimentu po daný moment – nevidí do budúcnosti.



Obrázok 18. Štruktúra experimentu a jeho význam

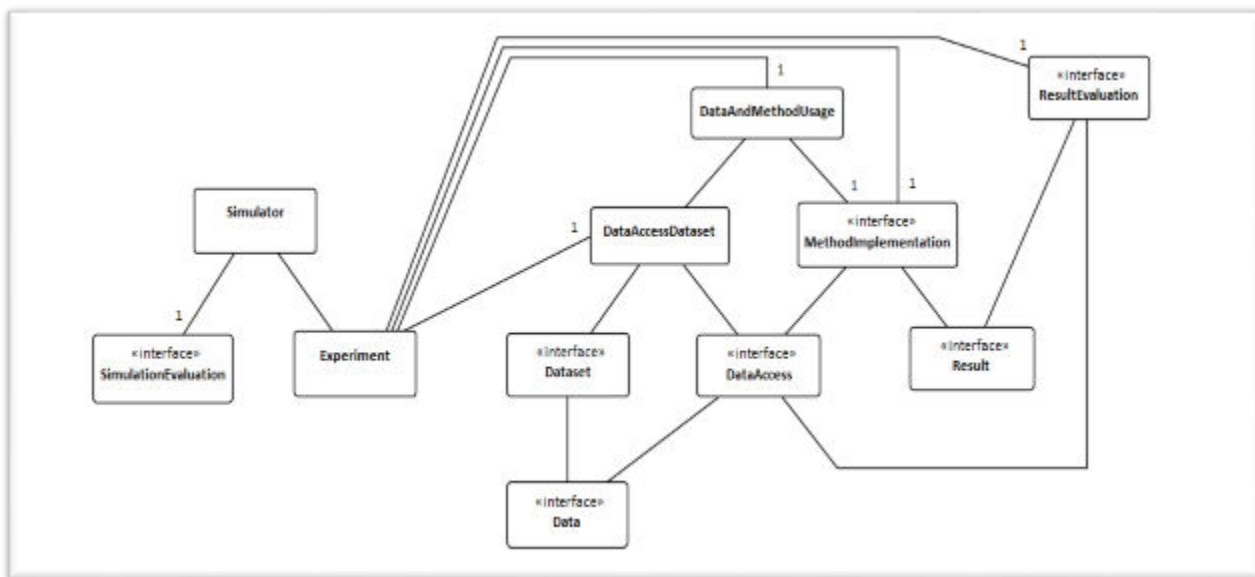
Experiment pre určovanie polohy teda bližšie špecifikujeme tak, že postupne prechádzame dataset a v každom časovom okamihu určíme polohu. Odhadnutá poloha je následne porovnaná s referenčnou dohodnutou metrikou a výsledky sú štatisticky spracované.

## 4.1 Simulátor experimentov

Simulačné jadro má za úlohu poskytnúť rozhrania pre definovanie vlastných experimentov, s ich konfiguráciou, vlastným typom a spôsobom získavania dát, typom prístupu k dátam, krokmi algoritmu, typom výsledku, spôsobom vyhodnocovania a spracovania výsledkov. Vytvára taktiež prostredie na vykonanie takýchto experimentov. Simulačné jadro je vystavané ako samostatná knižnica *Simulator*. Použitie pre konkrétnu úlohu spočíva v tom, že sa vytvorí vlastný typ experimentu, implementuje sa skúmaná metóda, určí sa prístup k dátam a implementuje sa metrika a spôsob vyhodnocovania výsledkov metódy.

Pre využitie a rozšírenie simulátora experimentov pre konkrétny projekt je potrebné porozumieť základnej štruktúre a významu tried – tie sú implementované v jazyku C++. Pre úsporu miesta opíšeme len tie najdôležitejšie.

- *Experiment* – zapuzdruje akcie potrebné na prípravu experimentu, určuje spôsob vykonania nejakého algoritmu nad datasetom a vyhodnotenie jeho výsledkov. Predstavuje automatizovaný experiment, ktorého vykonanie je možné naplánovať. Predstavuje samostatnú jednotku, ktorá si riadi svoje zdroje. Možno ho pripodobniť k programovému vláknu so špeciálnymi vlastnosťami.
- *DataAndMethodUsage* – definuje logiku experimentu a spôsob prepojenia datasetu a metódy. To znamená, aké sú kroky experimentu. Vo všeobecnosti získané dáta z datasetu predpripraví a zavolá nad nimi funkcionality metódy. Výsledok je posunutý ďalej pre potreby vyhodnocovania.
- *DataAccessDataset* – poskytuje dáta z datasetu aplikácii. Často sú pri dátach uložené informácie, s ktorými nechceme, aby metóda pracovala. Môžu to byť pomocné údaje alebo údaje na overenie správnosti výsledku. Prístup je obmedzený iba na údaje, s ktorými môže metóda pracovať. Sú 2 základné prístupy: testovací (dáta pre metódu) a na overenie správnosti (dáta potrebné na vyhodnotenie výsledku).
- *MethodImplementation* – slúži na oddelenie logiky metódy od konkrétnej metódy.



Obrázok 19. Diagram základných tried v knižnici *Simulator*

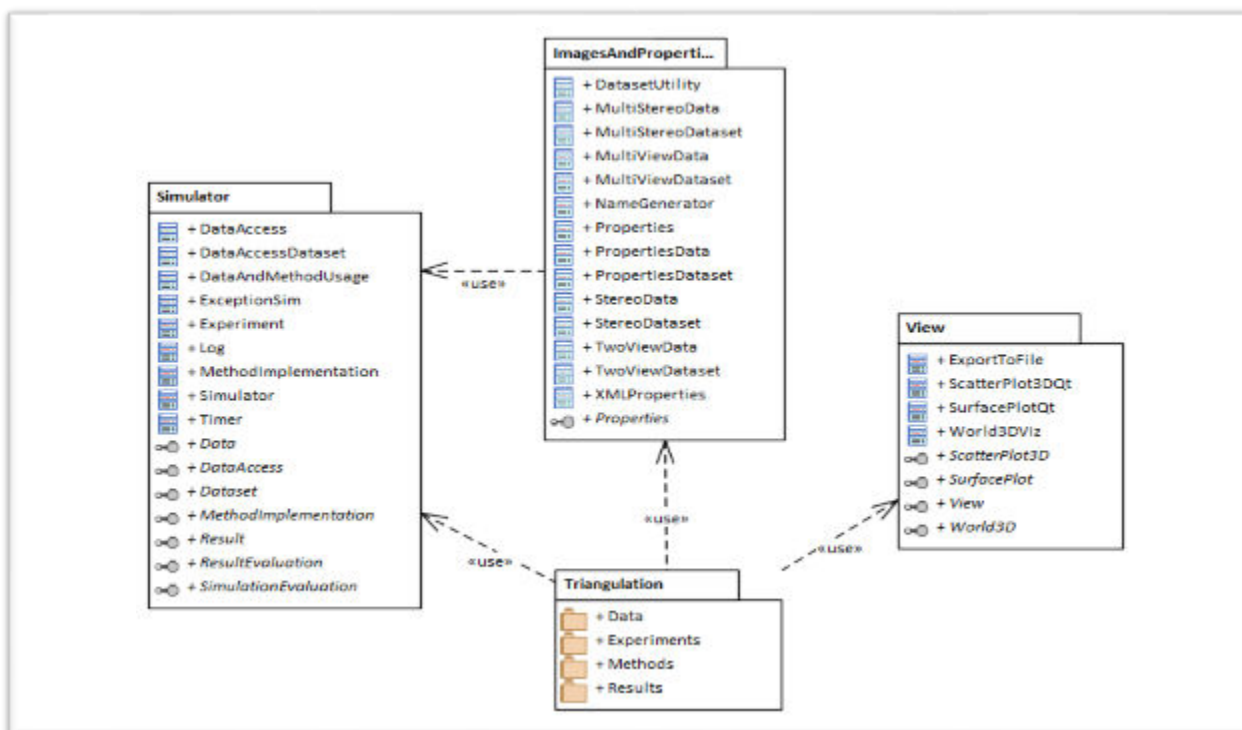
Logika je implementovaná nad rozhraním pre prístup k dátam. Metódy na určovanie polohy budú rozširovať práve toto rozhranie.

Nad touto knižnicou sú vystavané ďalšie pomocné knižnice:

- *ImagesAndPropertiesDataset* – podpora práce s datasetom (aj s datasetom z Obrázok 17). Pričom štruktúra súborov môže byť ľubovoľná. Podporuje viaceré módy prístupu určené podľa rozsahu prístupu k predošlým fotografiám a počtu kamier.
- *Triangulation* – podpora pre úlohy určovania polohy. Obsahuje 4 moduly pre prácu s dátami, algoritmy metód, metriku a vyhodnocovanie výsledkov a podpornú funkcionálnu na vytvorenie experimentov pre simulátor.

Ďalšie implementované knižnice:

- *View* – podpora pre vizualizáciu a export dát. Podporuje vykresľovanie v 3D priestore, vytváranie grafov a export v podobe obrázkov alebo priamych dát.



Obrázok 20. Implementované knižnice

## 4.2 Podpora vizualizácie a práce s datasetmi

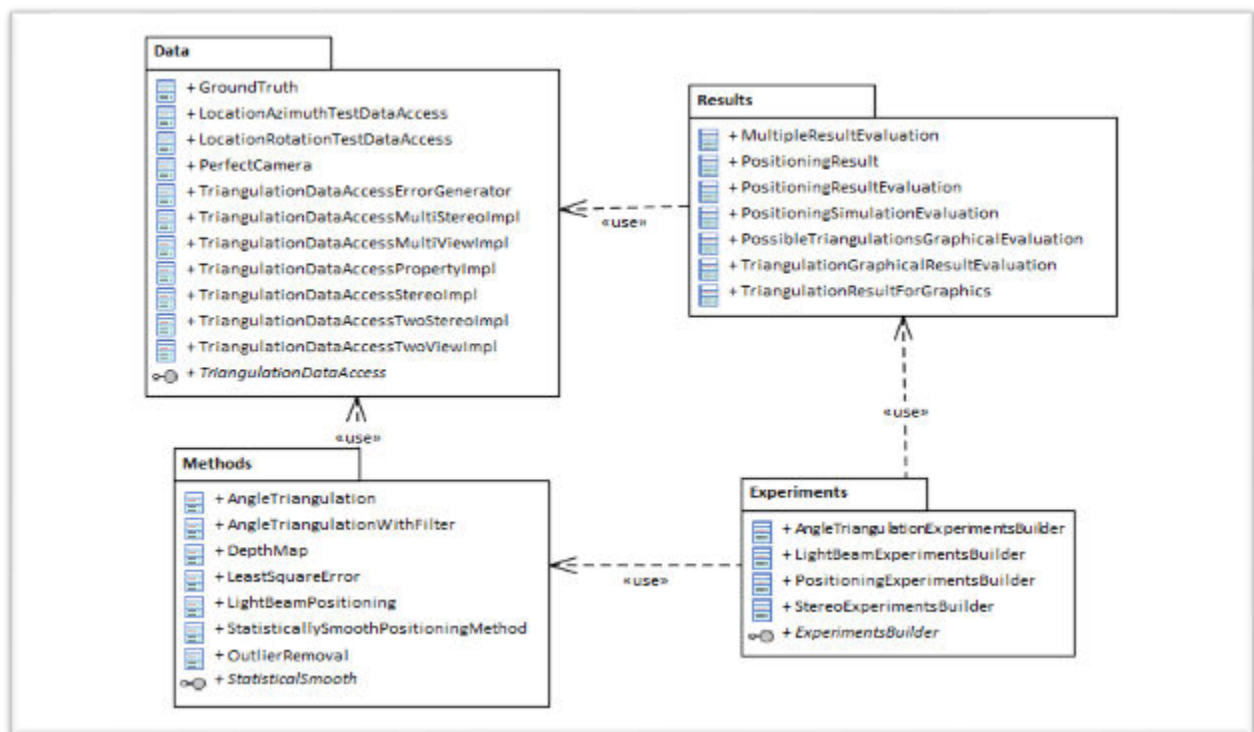
Knižnica **ImagesAndPropertiesDataset**. Oddeluje prácu s datasetom od jadra. Podporuje prácu s datasetom, ktorý je tvorený sekvenciou obrázkov – videosekvenciou – s parametrami. Vizúálna informácia je uložená v súbore v tvare obrázku v ľubovoľnom základnom formáte. Na množinu parametrov prislúchajúcich obrázku sa odkazuje jednoznačným identifikátorom. Je implementovaná podpora pre prácu s XML súborami, v ktorých sú uložené parametre. Identifikátor je v tomto prípade názov XML súboru. Štruktúra je však navrhnutá tak, že využiť iný spôsob uloženia parametrov je taktiež možné, stačí implementovať rozhranie *Properties*. Okrem parametrov pre každý obrázok, teda pre každý časový okamih, je možné definovať aj globálne parametre. V prípade, že využívame XML súbory pre parametre, môžeme tak zredukovať opakovanie duplicitných hodnôt a vo výsledku veľkosť datasetu.

Knižnica **View**. Oddeluje vizualizáciu a export dát. Slúži najmä ako pomocná knižnica pri ladení a práci s výsledkami experimentov, a preto nie je až taká rozsiahla. Pre potreby ladenia bolo nevyhnuté mať k dispozícii 3D priestor pre vykresľovanie grafiky ako napríklad polohy kamier a odhadnutý bod. Pri vyhodnocovaní výsledkov sú zase nápomocné grafy. Štruktúra je navrhnutá cez rozhrania tak, aby nebola aplikácia naviazaná na konkrétnu vizualizačnú knižnicu. Pri 3D vykresľovaní do priestoru sa využíva modul *Viz* z knižnice OpenCV. Pri grafoch sa využíva framework Qt.

Framework Qt slúži na vývoj aplikácií, ktoré dokážu bežať na rôznych softvérových a hardvérových platformách. Framework je pre jazyk C++. Jeho hlavné zameranie je vývoj grafického užívateľského rozhrania GUI. Obsahuje aj mnohé iné moduly na prácu s databázami, parsovanie XML a JSON súborov, manažment vlákien a podporu práce so sieťou. Pri implementácii bol framework využitý pri práci s XML súborami datasetu a ako už bolo spomenuté pri vytváraní aplikačných grafov, kde sa využíva modul pre vizualizáciu dát.

## 4.3 Rozšírenie simulátora na úlohu určovania polohy

Knižnica **Triangulation** rozširuje simulačné jadro pre prácu s úlohou určovania polohy. Implementuje prácu s datasetom s parametrami v XML súbore s konkrétnou štruktúrou. Vytvára prístupové vrstvy k dátam, definuje aký výsledok musia metódy vrátiť, metriku vyhodnocovania výsledkov a ich spracovania. Ďalej implementuje samotné metódy určovania polohy a poskytuje podpornú funkcionálnosť pre jednoduché definovanie experimentov. Je zložená zo 4 modulov: *Data*, *Results*, *Methods*, *Experiments*.



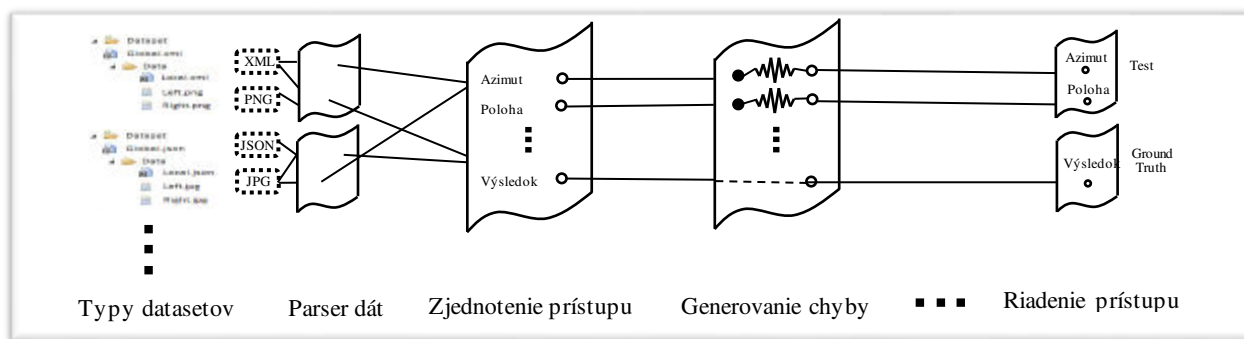
Obrázok 21. Štruktúra knižnice *Triangulation*

Modul **Data**. Má za úlohu implementovať 2 funkcionality: model prístupových vrstiev a prácu s XML súborami datasetu. Existujú viaceré spôsoby prístupu k dátam datasetu a v každom prípade sú dáta poskytnuté z rozhrania *Dataset* iné. Tieto surové dáta sú preto obalené vrstvou, ktorá k nim zjednocuje prístup. Nad touto vrstvou je následne ďalšia vrstva, ktorá určuje k akým dátam možno pristupovať – metóda by nemala mať prístup k výsledku, ktorý môže byť uložený



medzi údajmi v XML súbore. Prístup k dátam je teda riadený formou prístupových vrstiev (Obrázok 22. zobrazuje ukážku). Tie sú žiadané aj z iných dôvodov. Ak chceme napríklad testovať vplyv chyby na dáta, nie je potrebné vygenerovať nový dataset s chybou. Stačí vložiť novú vrstvu a upraviť tok dát z datasetu konkrétnym typom chyby. To je obrovská výhoda, pretože to šetrí čas potrebný na generovanie datasetu a miesto na disku (postačuje jeden dataset pre ľubovoľné množstvo experimentov s chybami).

Modul **Methods**. Hlavným cieľom je implementovať základné metódy určovania polohy objektu (tak ako bolo spomenuté v kapitole 2.1 a detailne opísané v kapitole 2.2) založené na



Obrázok 22. Model prístupových vrstiev

stereo korešpondencii a výpočte hĺbkovej mapy a rekonštrukcii trajektórie svetelného lúča využitím spätnej projekcie. Okrem týchto dvoch metód aj metódu založenú na triangulácii cez uhly (opísanú v kapitole 2.2.8). Bolo potrebné matematicky sformulovať uvedené myšlienky, v čom bola najviac prospešná klasická metóda: pero a papier (a samozrejme knižné a iné zdroje). Metódy bolo potrebné odladiť a overiť správnosť odvodených vzorcov. Základná logika fungovania metód je taká, že vozidlo prechádza po ceste, získava dáta z kamery, pričom v časovom okamihu má k dispozícii aktuálny obrázok, prípadne predošlé obrázky z histórie. Implementované metódy možno rozdeliť na 2 typy: tie, ktoré pracujú len s jedným (prípadne dvoma) časovými okamihmi a nevyužívajú iné predošlé dáta a tie, ktoré využívajú históriu predošlých dát z videosekvencie. Táto história môže byť obmedzená na určitú dĺžku – ak napríklad v praxi využívame zariadenie, ktoré nie je schopné uložiť väčšie množstvo dát. Využitie histórie má najmä opodstatnenie pri

metóde založenej na triangulácii cez uhly, kde sa pokúšame odfiltrovať také situácie, kedy nevznikne trojuholník prípadne je pravdepodobná veľká chyba výpočtu. História môžeme napríklad využiť aj na spresnenie výsledku metódou najmenších štvorcov, ak predpokladáme lokalizáciu rovnakej dopravnej značky.

Pri implementácii je potrebné pracovať okrem iného aj s obrazovou informáciou. Na tento účel je využitá knižnica OpenCV. Podporuje základnú prácu s digitálnou fotografiou ako je načítanie zo súboru uloženom v určitom formáte, upravenie obrazu a mnohé pokročilé metódy predspracovania. Obsahuje množstvo algoritmov z rôznych oblastí počítačového videnia. OpenCV je *open-source* knižnica, jedna z najznámejších a najpoužívanejších v tejto oblasti. Je napísaná v jazyku C a C++ a navrhnutá tak, aby bola výpočtovo efektívna. Vývoj komerčného softvéru touto knižnicou nie je spoplatnený a zdrojové kódy nemusia byť zverejnené. Prichádza s vlastnými dátovými typmi a spôsobom práce, na ktorý si treba zvyknúť. Knižnica je rozsiahla a na jej vývoji pracuje množstvo ľudí. Napriek tomu je dokumentácia k nej pomerne malá a odporúčaný spôsob je skôr siahnuť po komerčných knižných tituloch. Výhodou OpenCV je výpočtová efektívnosť (v porovnaní s inými podobnými knižnicami je výpočtovo najrýchlejšia [17] [18]) a má veľkú komunitu ľudí. Mnohé publikované vedecké články ju využívajú pri implementácii algoritmov. Výborným zdrojom na štúdium sú knižné tituly ako [17] [19] a iné, ktoré predstavujú prácu s knižnicou od základov. Nápomocné sú taktiež diskusné fóra.

Modul **Results**. Definuje prácu s výsledkami metód a ich vyhodnocovanie. Metóda určovania polohy by mala vrátiť výsledok v podobe polohy objektu v scéne. Vyhodnocovanie experimentov má viacero úrovní. Prvá úroveň je vyhodnocovanie výsledku zo skúmanej metódy zvolenou metrikou. Ďalšia úroveň je vyhodnocovanie ako sa metóde darilo na celom datasete, následne na iných datasetoch a v porovnaní s inými metódami.

Modul **Experiments**. Obsahuje podpornú funkcionálnosť pre vytvorenie množiny automatizovaných experimentov. Ručné vytváranie a konfigurácia každého experimentu by bola zdlhávavá, a preto sú vytvorené *builder-y* pre každú z metód.

## 5 Experimentálne výsledky

---

Aké sú výsledky implementovaných metód, ich efektivita a úspešnosť? To je teraz cieľom nášho skúmania. Chceme zistiť, ako sa správajú metódy pri rôznych vstupných podmienkach, aké majú silné a slabé stránky. Pred samotným experimentovaním je vhodné si vytvoriť plán a spôsob akým chceme previesť experimenty. Cieľom je vyhodnotiť presnosť implementovaných metód na určovanie polohy a preskúmať, ako ju ovplyvňujú chyby v dátach a parametre scény. Označme implementované metódy ako metóda 1. *stereo korešpondencia* (viď kapitola 2.2.6), metóda 2. *rekonštrukcia svetelných lúčov* (viď kapitola 2.2.7) a metóda 3. *triangulácia cez uhly* (viď kapitola 2.2.8).

Ako určiť presnosť metódy? V prvom rade musíme definovať metriku na jej výpočet – v tomto prípade jednoducho ako vzdialenosť. Metóda je natoľko presná, nakoľko veľkú chybu vo všeobecnosti produkuje. Napríklad metóda má presnosť 1 meter vtedy, ak je chyba v priemere 1 meter. Definujme teda chybu odhadu ako euklidovskú vzdialenosť medzi 2 bodmi - bod odhadu a bod skutočnej polohy objektu. Jednotka chyby odhadu nech je jeden meter SI.

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} \text{ skutočná (referenčná) poloha objektu} \quad \begin{bmatrix} x \\ y \\ z \end{bmatrix} \text{ odhad}$$

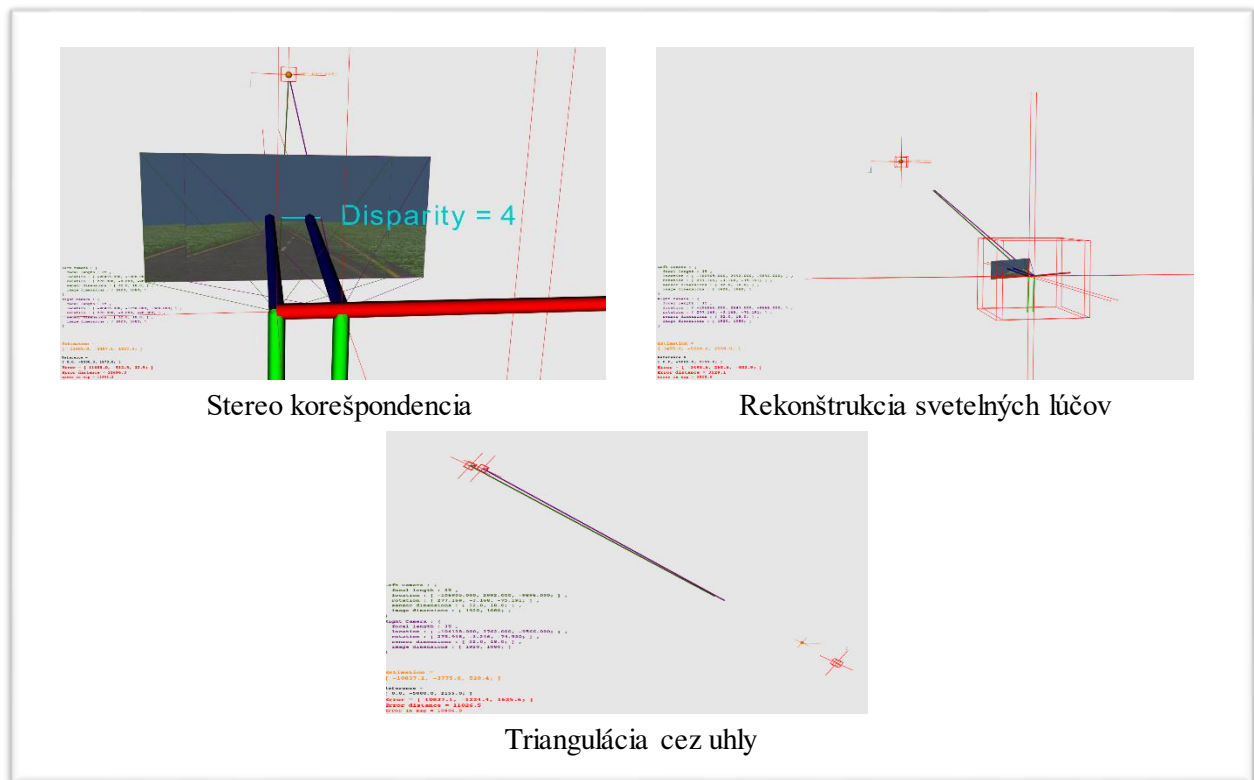
$$\text{chyba odhadu} = \sqrt{(u - x)^2 + (v - y)^2 + (w - z)^2}$$

Skutočná (referenčná) poloha je v našom prípade poloha dopravnej značky v priestore. Metódy 2 a 3 produkujú výsledok v 2D. Ak nie je uvedené inak, je v takom prípade chyba vypočítaná v rovine. Pri experimentoch rozšírenia odhadu do 3D je chyba počítaná v 3D.

### 5.1 Vizualizácia výsledkov

Pri práci s datasetmi počas experimentovania a najmä počas ladenia metód bola veľmi užitočná vizualizačná knižnica. Vďaka nej je možné sa pozrieť až do samotných útrobov a skúmať, ako sa jednotlivé metódy správajú v konkrétnej situácii na ceste. Pri práci na takejto úlohe je totiž

veľmi náročné odladiť metódu len tým, že človek skúma len čísla predstavujúce vektory, uhly a súradnice. Samozrejme aj to je nevyhnutné, ale dramaticky tomu dokážeme pomôcť vizualizáciou. Tá je podobná pri všetkých metódach, aj keď výpočet samotnej polohy môže byť založený na iných princípoch. Vizualizujú sa kamery na 2 rôznych polohách s fotografiou, ktorá udáva ako kamery vidia priestor. Tento priestor je interaktívny a je možné sa v ňom pohybovať a skúmať scénu z rôznych uhlov, ak to je potrebné. Okrem toho je zobrazený odhad a referenčný bod, ktorý je spolu s kamerami obklopený rámom, ktorý určuje 1 metrovú vzdialenosť od bodu, pre jednoduchšie vizuálne pozorovanie. Pre jednotlivé metódy sú taktiež zobrazené relevantné údaje. Chyba odhadu je vypočítaná priamo na obrazovku.



Obrázok 23. Vizuálna ukážka pracovania jednotlivých metód zobrazená využitím implementovanej knižnice View

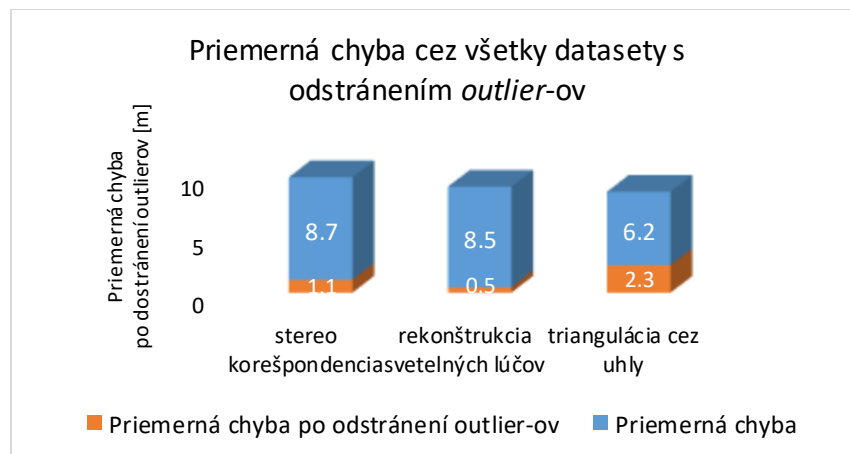
## 5.2 Presnosť metód

V nasledujúcom texte nás bude zaujíma presnosť jednotlivých metód. Zhrnutím všetkých výsledkov vyprodukovaných konkrétnou metódou pre všetky testované datasety dostávame priemernú chybu. Tá je zobrazená v nasledujúcom grafe. Pod všetkými výsledkami sa myslí každý odhad vo všetkých relevantných časových okamihoch videosekvencie pre všetky datasety. To znamená nie len jeden odhad pre napríklad priamu cestu, ale všetky odhady pozdĺž cesty. Zohľadňujú sa nefiltrované výsledky pre všetky vzdialenosti od objektu. Všetky nasledujúce výsledky sú uvádzané pre filmovú kameru, pokiaľ nie je uvedené inak.



Graf 1. Porovnanie výsledkov metód zo všetkých datasetov

Ako už bolo spomenuté, vozidlo sa pohybuje a metóda získava dáta postupne z kamery. Metóda môže využívať pri určovaní polohy predošlé obrázky, čo je definované históriou. Ak nie

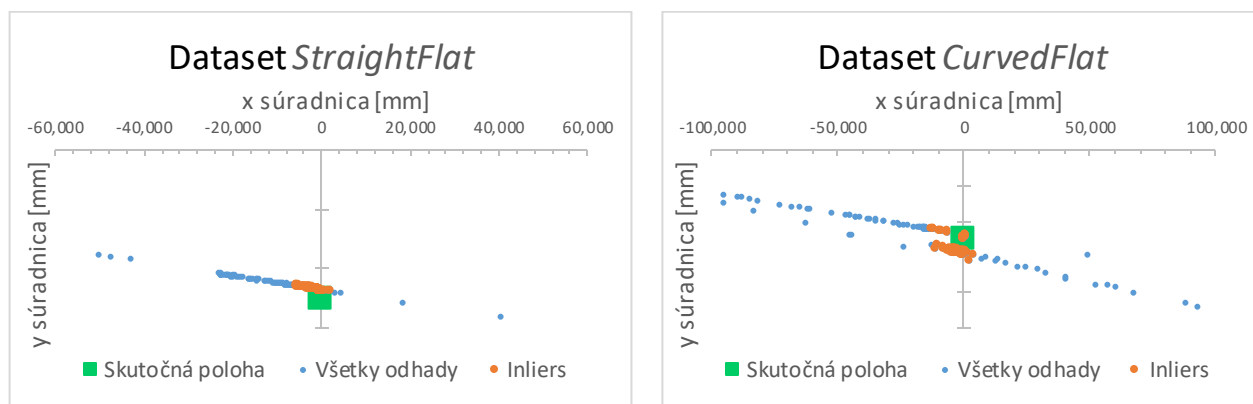


Graf 2. Porovnanie výsledkov metód zo všetkých datasetov

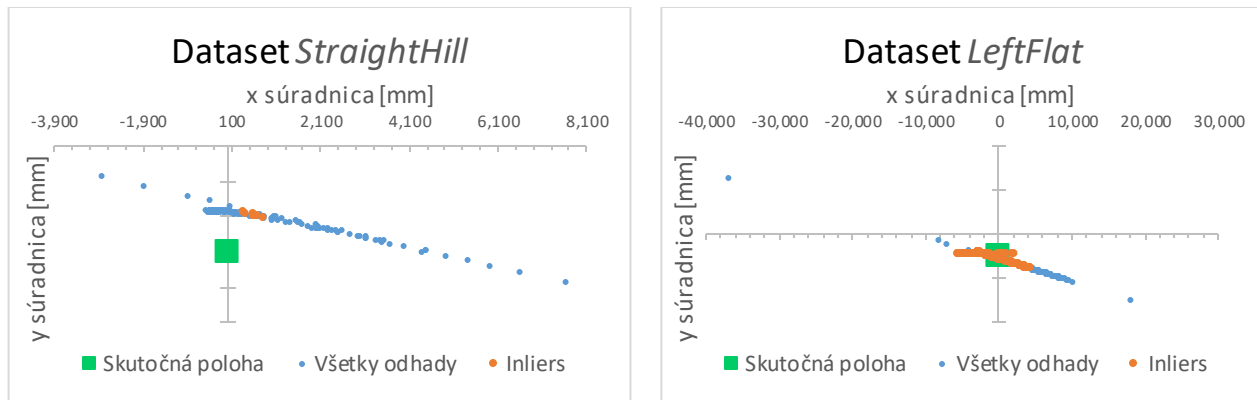
je explicitne uvedené inak, metóda má prístup ku celej histórii. V prípade skúmaných datasetov je cieľom určiť polohu dopravnej značky. Táto značka nemení svoju polohu v priestore, a teda všetky odhady v rámci datasetu sa snažia určiť polohu rovnakého objektu. Odstránením najviac nepresných odhadov a spriemerovaním zvyšných môžeme zlepšiť presnosť metód (bližšie je tento spôsob opísaný v nasledujúcom texte). Takýto prístup možno využiť vždy, keď určujeme polohu statického objektu. Dostávame jeden odhad metódy na polohu dopravnej značky pre jeden dataset. Vplyv na výslednú presnosť je naozaj markantný – vid' Graf 2.

Pod odstránením *outlier-ov* rozumieme odstránenie bodov, ktoré vnášajú do určovania polohy najväčšiu chybu a tie, ktoré majú najväčší rozptyl. Celý problém si môžeme predstaviť tak, že máme množinu bodov, z ktorých odfiltrujeme tie, ktoré sa nachádzajú na okrajoch. Pre každý bod je vypočítaná cena ako vzdialenosť k najvzdialenejšiemu bodu z podmnožiny bodov. Podmnožina bodov obsahuje určité percento najbližších bodov. Následne je zvolená percentuálna hranica pre cenu. Tá rozdelí rozsah cien na *outlier-y* a *inlier-y* (teda okrajové a vnútorné body). To funguje za predpokladu, že väčšina bodov sa nachádza v okolí referenčnej hodnoty a okrajové hodnoty sú zriedkavé. Takýto spôsob odhadu možno použiť pri spracovaní datasetu po dojazde vozidla a vyhodnotením všetkých odhadov získať jeden presnejší. Napríklad pri dodatočnom anotovaní alebo vytváraní máp.

Pre lepšiu predstavu sú uvedené filtrované hodnoty pre jednu metódu a vybranú množinu datasetov.

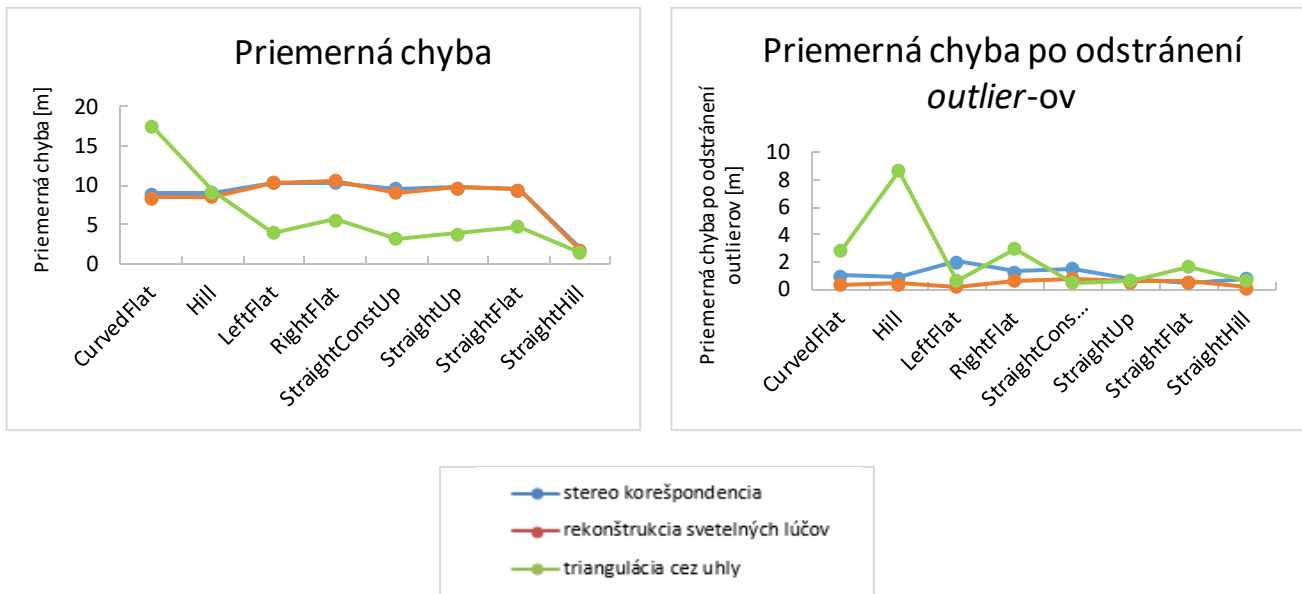


Graf 3. Odstránenie najviac nepresných odhadov - outlierov - pri metóde *triangulácia cez uhly*. Ukážka na vybraných datasetoch.



Graf 4. Odstránenie najviac nepresných odhadov - outlierov - pri metóde *triangulácia cez uhly*. Ukážka len na vybraných datasetoch.

Pre bližšiu analýzu sa pozrieme ako si počínali metódy na jednotlivých datasetoch. Odhady sú rozdelené podľa datasetov.

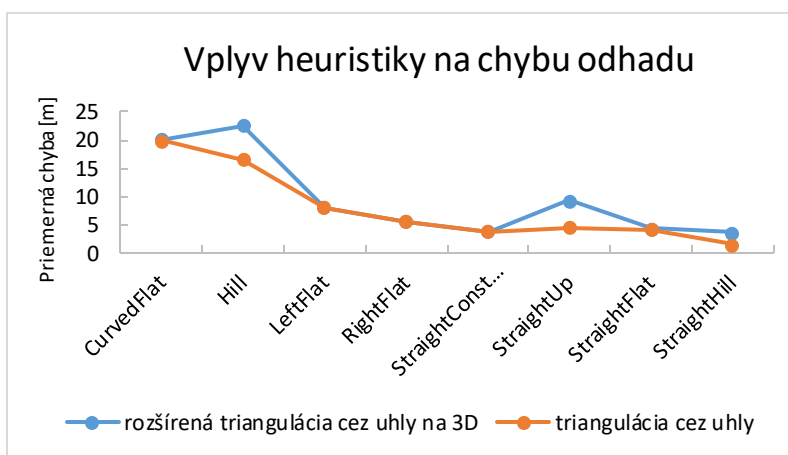


Graf 5. Výsledky metód pre jednotlivé datasety

Metóda založená na stereo korešpondencii dosahuje veľmi podobné výsledky ako rekonštrukcia trajektórie svetelných lúčov. V priemere dosahujú horšie výsledky, po odstránení outlier-ov sú však v priemere úspešnejšie. To naznačuje, že ich výsledky sú kompaktnejšie. Metóda triangulácie cez uhly určite spresnila „jednorazové“ určovanie polohy. Zdá

sa, že najväčší problém jej robí situácia so zakrivenou cestou a nerovnomerné stúpanie a klesanie. Pri poslednom uvedenom datasete sú výsledky zdanlivo veľmi presné, čo je spôsobené tým, že vozidlo vidí značku až pri schádzaní z kopca, a teda odhaduje jeho polohu pri menších vzdialenostiach (bližšie bude táto téma analyzovaná v ďalšom texte). Pri pohybe vozidla na plochej ceste a ceste s konštantným stúpaním dosahujú jednotlivé metódy podobné výsledky. Či teda vozidlo ide do kopca s pravidelným stúpaním neprináša výrazný rozdiel v chybe. Prekvapením je, že dataset *Hill* s rôznymi zákrutami a kopcami, ktorý mal byť najväčšou výzvou, ním bol iba čiastočne. Pri metódach stereo korešpondencie a rekonštrukcie svetelných lúčov sú výsledky pomerne robustné. Napriek tomu pozorovať, že datasety ľavotočivej a pravotočivej zákruty boli o niečo väčšou výzvou ako dataset *Hill*. Zdá sa totiž, že väčšia chyba je pri objektoch, ktoré sa nachádzajú na okrajoch zorného poľa kamery. Pri datasete *Hill* sa objekt viackrát nachádza na okraji, ale nie po takú dlhú dobu ako v prípade datasetov *LeftFlat* a *RightFlat*.

Metóda stereo korešpondencie a triangulácie cez uhly produkujú 2D odhad. V niektorých prípadoch môže byť žiadané získať výsledok, ktorý určuje polohu objektu v priestore. Súradnicu určujúcu v našom prípade výšku sa môžeme pokúsiť odhadnúť. Heuristika na jej odhad je založená na predpoklade, že vozidlo sa pohybuje priamočiarno po rovine zarovnanéj so súradnicovou sústavou. Získaný 2D odhad je v takom prípade bod roviny. Pre ten dokážeme matematicky odvodiť aj jeho tretiu súradnicu. Matematický opis a spôsob výpočtu nie je pre veľkosť odvodu



Graf 6. Vplyv rozšírenia 2D výsledku metódy na 3D využitím opísanej heuristiky. Výsledky predstavujú priemer zo všetkých výsledkov metódy *triangulácie cez uhly*. Metóda bola testovaná s históriou najviac 60 predošlých fotografií.

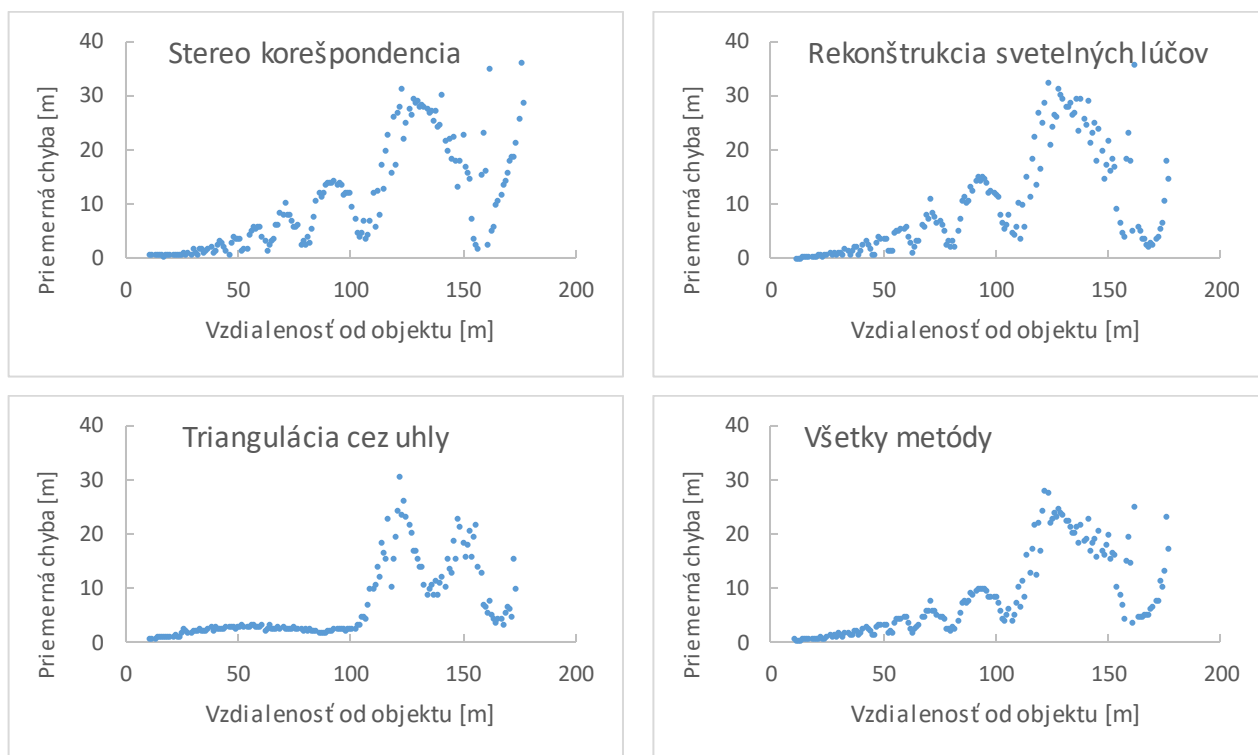


uvádzaný. Heuristické rozšírenie je skúmané na výsledkoch z metódy *triangulácie cez uhly* (Graf 6).

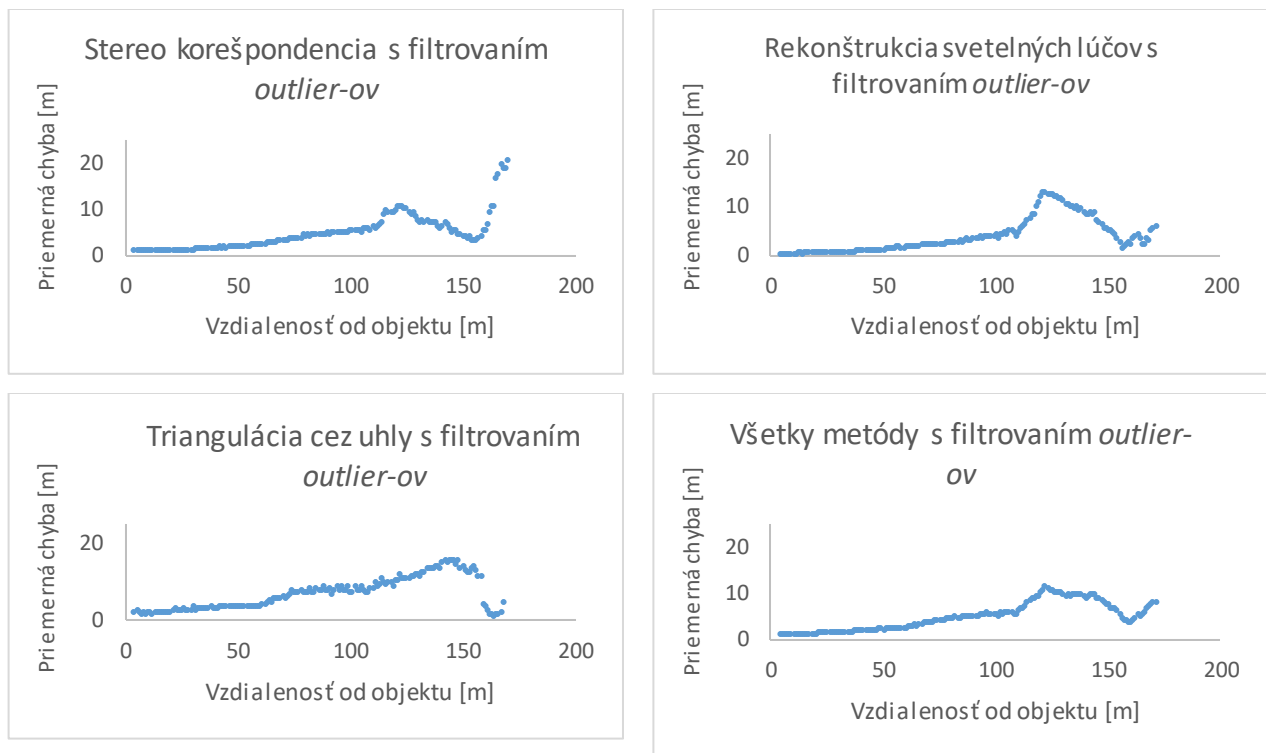
Pri plochej ceste funguje heuristika dokonale, čo je očakávané. Pri nerovnej ceste sú výsledky zhoršené. Cesta zvyčajne nemení výšku radikálne a takéto rozšírenie konkrétne pri úlohe lokalizácie dopravnej značky produkuje relatívne dobré výsledky.

### 5.3 Závislosť medzi vzdialenosťou a presnosťou

Zdá sa, že metódy si počínajú horšie pri väčších vzdialenostiach. Toto tvrdenie je preskúmané nasledujúcim grafom. Zohľadnené sú všetky odhady produkované všetkými metódami. Vzťah nie je ani zďaleka lineárny, možno však povedať, že pri väčších vzdialenostiach je chyba v priemere väčšia. Pokusom o preloženie funkcie bodmi sa najviac približuje mocninová funkcia, čo je pomerne prudký nárast priemernej chyby pri rastúcej vzdialenosti.



Graf 7. Vplyv vzdialenosti od objektu na presnosť. Zohľadnené sú všetky dáta produkované metódami



Graf 8. Vplyv vzdialenosti od objektu na presnosť. Metódy sú upravené tak, že pri každom odhade je zohľadnená predošlá história, na ktorú je aplikovaná opísaná metóda filtrovania outlierov

Všetky datasety majú približne rovnaký rozsah vzdialeností od objektu (Tabuľka 3 obsahuje ich prehľad). Kamera prechádza aj za dopravnú značku. To nie je relevantné a je uvádzaná najmenšia vzdialenosť 0. Keďže odhad produkujú metódy len v prípade, keď je objekt viditeľný, je vhodné zohľadniť aj takýto rozsah vzdialeností. Sú to vzdialenosti, pri ktorých určujeme polohu objektu. Dataset *CurvedFlat* má krátky úsek počas zákruty, kedy značku nevidno. Pri datasete *StraightHill* značku nevidno, keď sa nachádza za kopcom a jedine tento dataset nemá pokryté vzdialenosti väčšie ako približne 70 metrov. Do 70 metrov sú chyby produkované metódami relatívne malé, a preto dosahujú metódy na tomto datasete najlepšie výsledky.

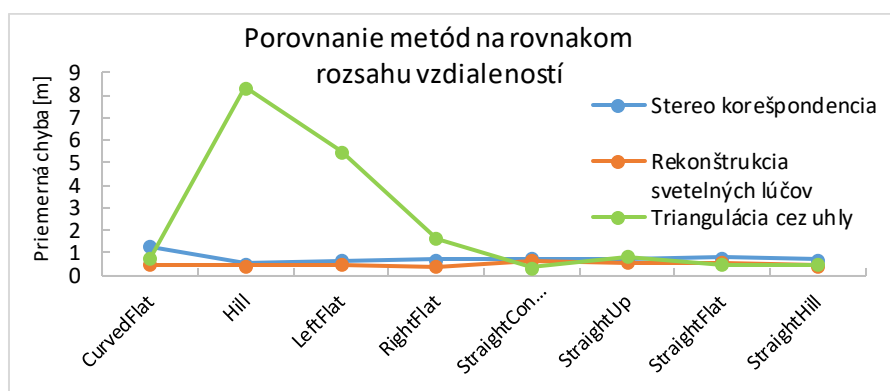
Povedzme, že je našim cieľom preskúmať presnosť na vzdialenosť 30 metrov. Nechceme, aby jeden dataset mal výhodu nad iným alebo ak úloha umožňuje predpokladať takúto maximálnu vzdialenosť objektu od kamery. To môže byť prípad pri určovaní polohy vo vnútorných priestoroch, kedy zvyčajne miestnosť nepresahuje rozmery desiatok metrov. Kludne sme mohli

zvoliť aj inú vzdialenosť napríklad 20, 25 metrov. 30 metrov je povedzme odhad pre určovanie polohy na menšie vzdialenosti.

Referenčné rozsahy vzdialeností od objektu		
Názov datasetu	Rozsah vzdialeností	Rozsah vzdialeností, kedy je možné pozorovať objekt
CurvedFlat	0 – 160 m	13 – 77,105 – 160 m
Hill	0 – 177 m	11 – 177 m
LeftFlat	0 – 174 m	10 – 174 m
RightFlat	0 – 154 m	11 – 154 m
StraightConstUp	0 – 164 m	10 – 164 m
StraightUp	0 – 161 m	10 – 161 m
StraightFlat	0 – 160 m	10 – 160 m
StraightHill	0 – 160 m	10 – 67 m

Tabuľka 3. Referenčné rozsahy vzdialeností od objektu pre jednotlivé datasety

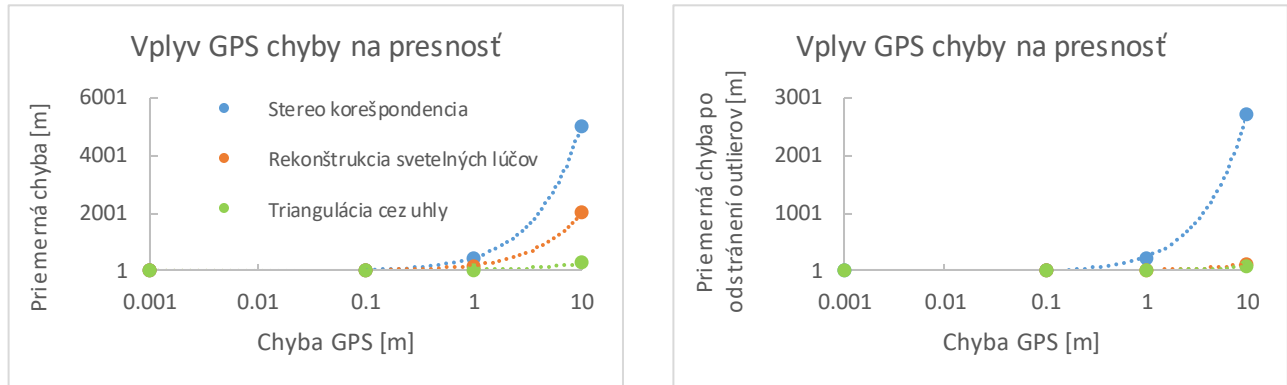
Ak povolíme metóde určiť polohu, len ak sa nenachádza od objektu ďalej ako 30 metrov, dostaneme nasledovné výsledky - Graf 9. Priemerná chyba je u všetkých metód pod 1 meter až na metódu *triangulácie cez uhly*. Tá dosahuje najhoršie výsledky pri datasetoch so zákrutami a nepravidelným stúpaním a klesaním.



Graf 9. Porovnanie metód na datasetoch s rovnakým rozsahom vzdialeností od objektu – pod 30 metrov. Pri výpočte bola obmedzená história na 2 sekundy, preto sa výsledky mierne odlišujú od predošlých.

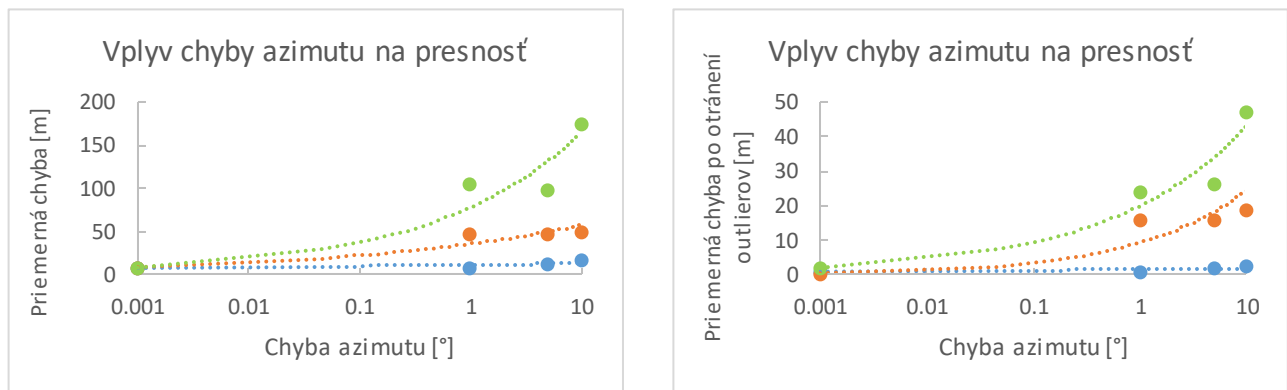
## 5.4 Vplyv nepresnej GPS polohy, azimutu a *bounding boxu* na presnosť

Nepresnosti v polohe kamier, spôsobujú posun v celom matematickom modeli a body pri triangulácii majú iné skutočné polohy. Poloha kamier je určená využitím globálneho polohovacieho systému GPS, ktorý je známy svojou chybovosťou. Okrem nepresnej polohy,



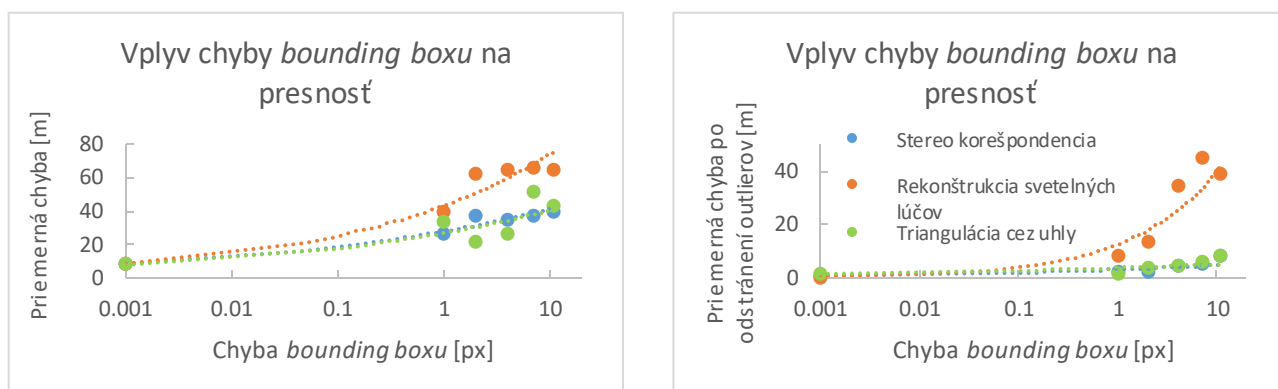
Graf 10. Vplyv nepresnosti GPS polohy na presnosť metód. Namerané dáta sú prekryté trendom bodkovanou čiarou. Legenda je uvedená v grafe vľavo hore

môžeme mať nepresný údaj o natočení kamery, prípadne chybnú anotáciu objektu. Naším cieľom je zistiť ako vplývajú tieto nepresnosti na presnosť metód.



Graf 11. Vplyv nepresnosti azimutu na presnosť metód. Namerané dáta sú prekryté trendom bodkovanou čiarou. Legenda je rovnaká ako v predošlom grafe

Z výsledkov sme zistili, že priemerná chyba rastie lineárne so zväčšujúcou sa chybou polohy GPS. Nie exponenciálne, pretože horizontálna os je logaritmická. Najmenší vplyv má na metódu *triangulácie cez uhly*. Chybný azimut – teda informáciu o tom, kam smeruje kamera – spôsobuje chybu v odhade, ktorá narastá mocninovým radom (rastie pomalšie ako lineárne – odmocnina).



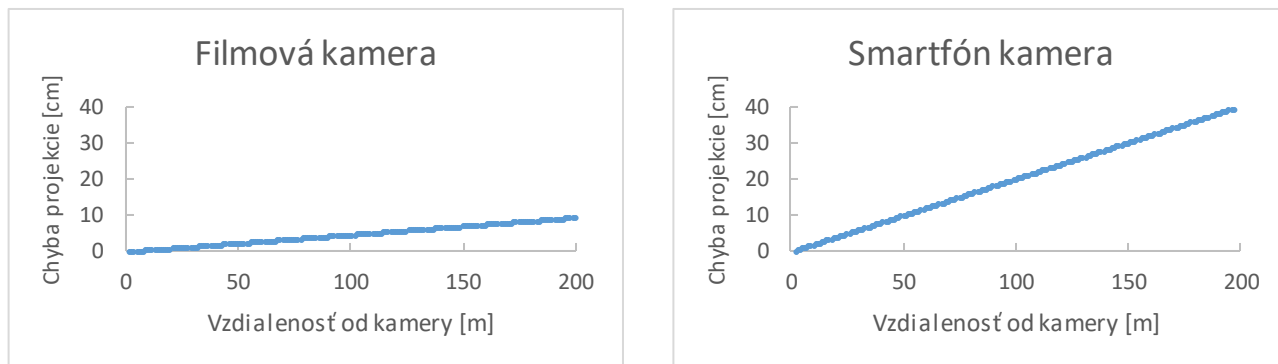
Graf 12. Vplyv chyby bounding boxu na presnosť metód. Namerané dáta sú prekryté trendom bodkovanou čiarou

Priemerná chyba rastie mocninovým radom – pomalšie ako lineárne. Ale už chyba 1 pixel v polohe bounding boxu dokáže zhoršiť určovanie polohy až o 7 metrov. Najväčší vplyv má táto chyba na metódu *rekonštrukcie svetelných lúčov*.

## 5.5 Vplyv konfigurácie a parametrov kamier na presnosť

Pri vytváraní fotografie v kamere sa vytvára obrazový bod tak, že na senzor dopadá viacero svetelných lúčov. Pri väčších vzdialenostiach splynú viaceré priestorovo blízke body, od ktorých sa lúče odrážajú, do jedného. To znamená, že pre určitú vzdialenosť, jednoducho nevieme určiť polohu presnejšie, pretože táto informácia sa stratila pri procese projekcie. Označme ju ako chyba projekcie. Na nasledujúcom grafe je chyba projekcie pre kamery z datasetov. Napríklad pri filmovej kamere zaberá 10 centimetrový objekt vo vzdialenosti 200 metrov jeden pixel.

Zvolenie vhodnej kamery pre konkrétny problém je dôležitý krok pri úlohách počítačového

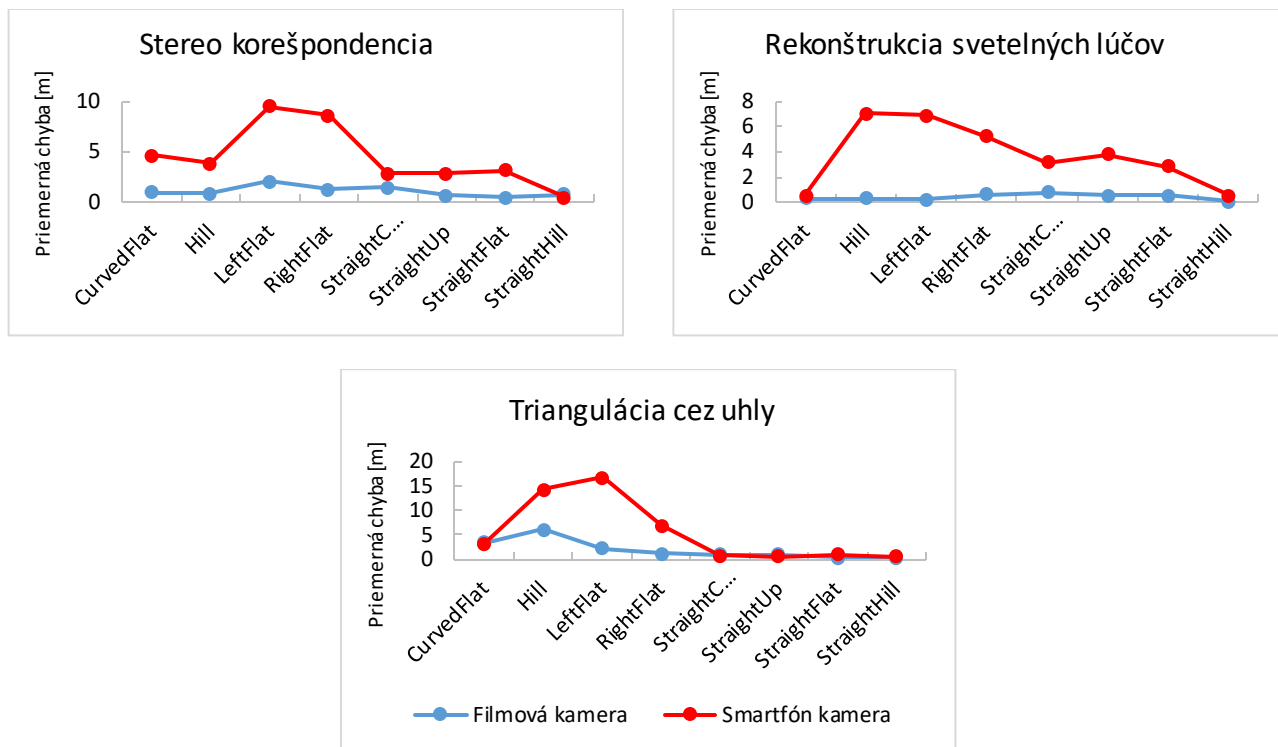


Graf 13. Chyba projekcie pre kamery použité v datasetoch

videnia. Vyhotovenie kamery určuje vizuálne vlastnosti fotografie a najmä spôsob ako sa bude obraz projektovať na senzor. Tieto vlastnosti možno opísať parametrami kamery ako ohnisková vzdialenosť, zorné pole, rozlíšenie fotografie a iné. Tieto parametre sú často vzájomne previazané. Pri generovaní datasetov sme použili 2 sady parametrov, aby sme overili vplyv aj takýchto parametrov na presnosť určovania polohy (viď Graf 14).

Horšia kvalita kamery smartfónu jednoznačne ovplyvnila presnosť skúmaných metód. Menšie rozlíšenie fotografie a menšia ohnisková vzdialenosť zvýšili radikálne chybu pri datasetoch so zákrutami a nepravidelným stúpaním a klesaním. Pri metódach *stereo korešpondencie* a *rekonštrukcie svetelných lúčov* je zhoršenie takmer vo všetkých datasetoch. Metóda *triangulácie cez uhly* sa zhoršila pri troch datasetoch výrazne viac ako iné metódy. V ostatných prípadoch však prekvapivo produkovala podobné chyby ako pri kvalitnejšej filmovej kamere.

Keď nemáme možnosť vybrať vhodnú kameru a pracujeme s nejakou, ktorá je jednoducho k dispozícii, je pri úlohách fotogrametrie dôležité čo najpresnejšie poznať projekciu kamery – teda vnútorné parametre kamery. Tieto údaje môžeme získať od výrobcu alebo popisu produktu. Často však nie sú udávané s dostatočnou presnosťou, prípadne kvôli reklame sú hodnoty skreslené a nie je z nich možné zistiť skutočné. V takom prípade nám neostáva nič iné iba tieto parametre odhadnúť pomocou kalibrácie – tá je odporúčaná aj v prípade, že tieto parametre poznáme.



Graf 14. Porovnanie filmovej a smartfónovej kamery

Na kalibráciu je možné využiť napríklad nástroj z knižnice OpenCV, ktorý z množiny fotografií kalibračného vzoru vypočíta vnútorné parametre kamery. Pozrime sa ako presne dokáže tento nástroj odhadnúť parametre filmovej a smartfón kamery. Na tento účel bolo potrebné namodelovať a vytvoriť nový dataset so scénou, ktorá obsahuje kalibračný vzor.

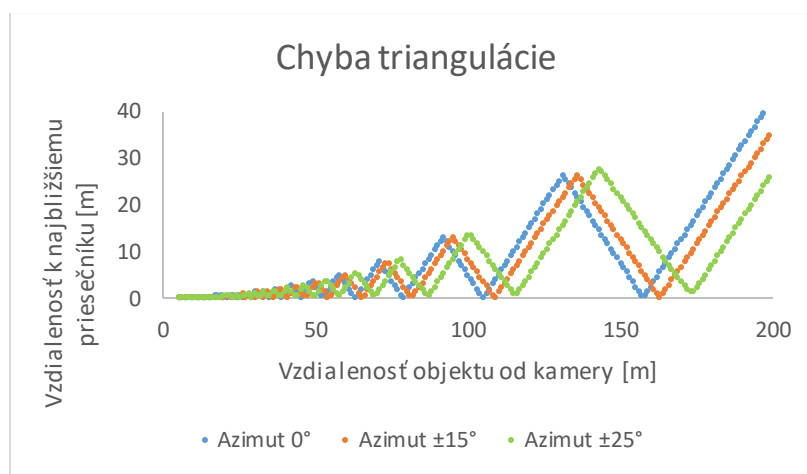
Kalibrácia kamier		
Typ kamery	Odhadnuté vnútorné parametre kamery	Skutočné vnútorné parametre kamery
Filmová kamera	$K_{odhad} = \begin{bmatrix} 2110,9 & 0 & 960 \\ 0 & 2110,9 & 510 \\ 0 & 0 & 1 \end{bmatrix}$	$K_{skutočné} = \begin{bmatrix} 2100 & 0 & 960 \\ 0 & 2100 & 540 \\ 0 & 0 & 1 \end{bmatrix}$
Smartfón kamera	$K_{odhad} = \begin{bmatrix} 494,3 & 0 & 320 \\ 0 & 494,3 & 240 \\ 0 & 0 & 1 \end{bmatrix}$	$K_{skutočné} = \begin{bmatrix} 493,3 & 0 & 320 \\ 0 & 493,3 & 240 \\ 0 & 0 & 1 \end{bmatrix}$

Tabuľka 4. Presnosť odhadu vnútorných parametrov procesom kalibrácie

Z predchádzajúcej tabuľky vidíme, že procesom kalibrácie dokážeme pomerne presne odhadnúť vnútorné parametre kamery. Hodnoty  $k_y, k_x$  majú jednotku pixel na jeden milimeter.

Predpokladajme teraz, že dokážeme navrhnúť najpresnejšiu metódu na určovanie polohy objektu, ktorý na fotografii dokážeme detekovať s presnosťou len na pixel. Aká by bola jej chyba pri nami navrhnutej metrike a datasetoch? Dokážeme to vôbec zistiť? Fotografia samotná neobsahuje nič iné iba množinu bodov. Ak dokážeme zistiť, do ktorej oblasti priestoru a s akou presnosťou sa týmito bodmi dokážeme dostať, získame prehľad o chybe. Z geometrického hľadiska predstavuje túto oblasť množina priesečníkov všetkých priamok vychádzajúcich z optického stredu jednotlivých kamier a prechádzajúcich množinou obrazových bodov. Takýto priestor je diskretný. V tomto prípade sa pod kamerou myslí konkrétny pohľad kamery na konkrétnej polohe. To znamená v prípade stereo kamier dostaneme iný priestor ako v prípade jednej kamery posunutej v smere jazdy pár metrov. Označme vzdialenosť odhadu metódy a najbližšieho bodu tohto priestoru ako chybu triangulácie.

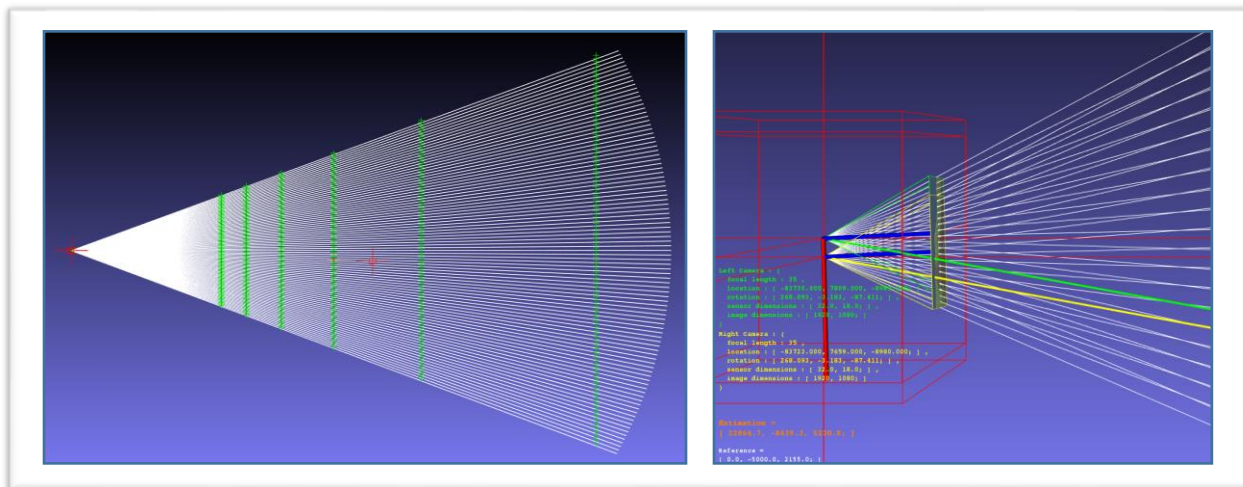
Pozrime sa na prípad stereo kamier z datasetov. Vypočítajme chybu triangulácie v závislosti od vzdialenosti objektu od kamery (viď Graf 15). Tento vzor bolo možné pozorovať už skôr, pri analýze ako vplyva vzdialenosť na presnosť odhadu (viď Graf 7). Záhadou zatiaľ je, prečo chyba nerastie, ale periodicky klesá a stúpa. Pokúsme sa preto vizualizovať priestor



Graf 15. Chyba triangulácie pre stereo pár filmových kamier s odstupom 15 cm. Azimut určuje uhol, pod ktorým pozorujeme objekt



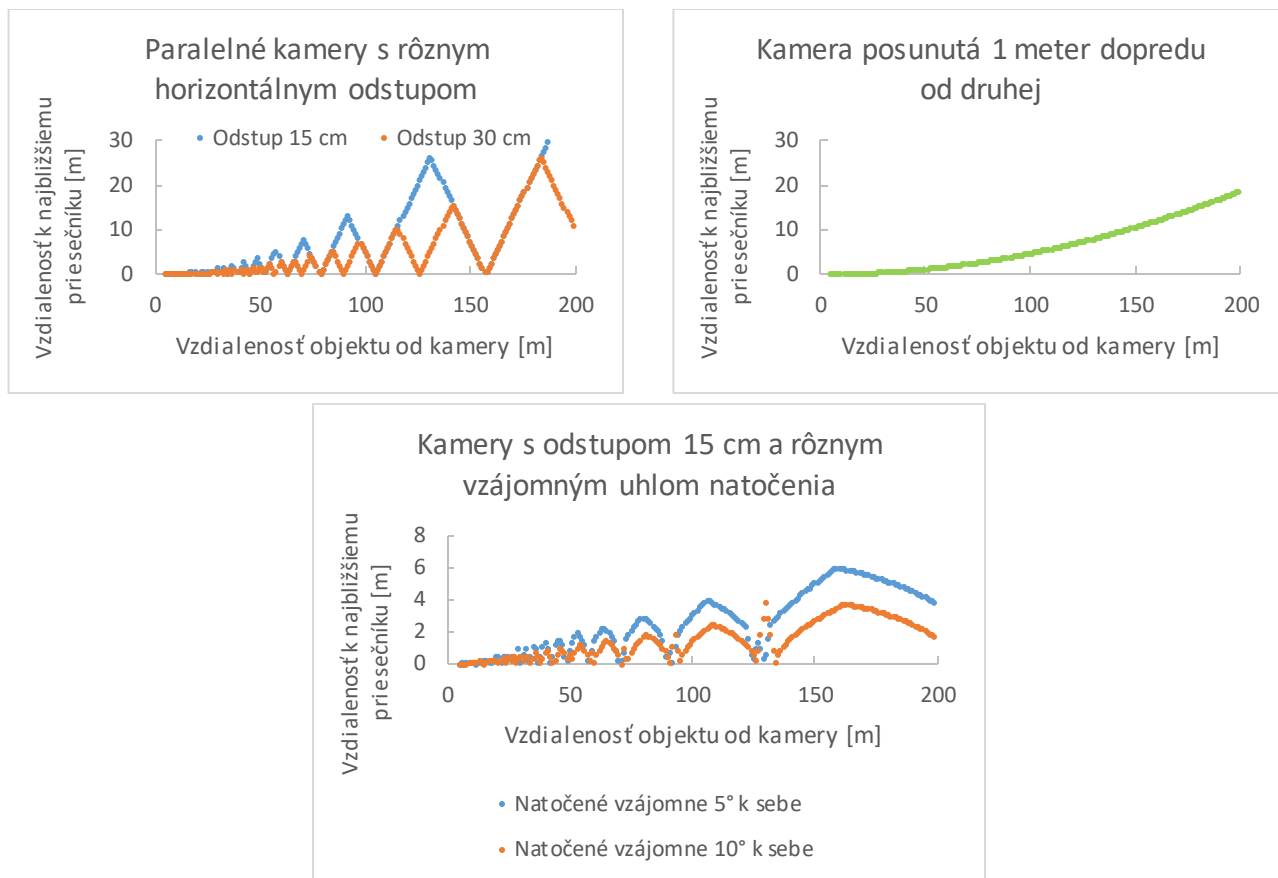
priesečníkov. Pre zjednodušenie vizualizácie zobrazme len priesečníky priamok z jedného rovnakého riadku fotografií. Pokrytie takéhoto priestoru je zobrazené na Obrázok 24. Prekvapivé je zistenie, že priesečníky sa nachádzajú na paralelných priamkach, čo určite nie je ideálne. Žiadané by bolo skôr pokrytie s čo najmenšími prázdnyimi oblasťami.



Obrázok 24. Vizualizácia priestoru možných triangulácií pre paralelné kamery

Ako sa tento priestor zmení, ak zmeníme konfiguráciu kamier ako vzájomné natočenie a poloha? Výpočet takejto chyby je zobrazený na Graf 16. Výrazné zmenšenie chyby pozorujeme, ak kamery nie sú paralelné, ale vzájomne k sebe natočené. Či teda kamery natočiť k sebe a pod akým uhlom je však závislé od konkrétnej úlohy, pretože musíme vyhodnotiť kompromis: pri paralelných kamerách máme najväčšiu vzdialenosť, na ktorú sme schopný objekt vidieť z oboch kamier, pri natočení kamier sa znižuje priestor videný oboma kamerami.

Dôsledkom tohto zistenia je, že žiadna metóda nedokáže určiť odhad polohy s menšou chybou ako je chyba triangulácie. Toto tvrdenie zdanlivo porušuje metóda triangulácie cez uhly (viď Graf 7), ale táto metóda nevyužíva na výpočet stereo pár kamier, a preto chyba triangulácie je iná. Dokonca pri každom odhade sa využívajú iné pohľady, a preto nie je možné jednoducho vypočítať takúto chybu.



Graf 16. Chyba triangulácie pre rôzne konfigurácie kamier: odstup, posun v smere jazdy a vzájomne natočenie. Objekt je vo všetkých prípadoch pozorovaný pod azimutom  $0^\circ$  od pomyselnjej osi kamier

## 5.6 Ostatné parametre

Známa veľkosť objektu je ďalší variant tejto úlohy. Pri úlohe dopravnej značky to môže byť jej rozmer. S takouto informáciou dokonca dokážeme určiť vzdialenosť od objektu z jednej fotografie ak poznáme jej ohniskovú vzdialenosť. Problém následne ostáva previesť túto vzdialenosť na polohu v priestore. Na získanie polohy potrebujeme poznať polohu kamery, jej natočenie a zorné pole kamery. V praxi môžeme mať k dispozícii rozsah pre rozmery objektu a keďže implementované metódy majú rovnaké predpoklady, môžeme tento spôsob použiť na určenie najmenšej a najväčšej vzdialenosti od objektu. To môže pomôcť odstrániť body, ktoré sú príliš nepresné.

Existuje množstvo ďalších vlastností a parametrov scény a kamier, ktoré neboli ešte spomenuté a môžu vplývať na presnosť určovania polohy priamo alebo nepriamo napríklad tak, že ovplyvňujú iné parametre. Pri analýze sme sa však pokúsili identifikovať všetky relevantné parametre. Medzi ďalšie parametre môže patriť rýchlosť kamery, ktorá môže zmeniť vizuálne vlastnosti obrazu – tie sú relevantné najmä pre detektor – a vzdialenosti medzi fotografiami, textúra objektu – dôležitá pri detekcii a iné.

## 6 Zhrnutie a záver

---

Počítačové videnie a jeho využitie a aplikácie v praktických úlohách boli pre mňa veľkou motiváciou pri riešení tohto problému. Taktiež možnosť získať vedomosti a praktické skúsenosti. Je to oblasť vášnivého záujmu vedcov, ktorý neustále posúvajú jej hranice. Vznikajú nové technológie a produkty, o ktorých by mnohí ľudia ani nepovedali, že sú skutočné. Pri štúdiu materiálov som sa naučil mnoho vecí z tejto oblasti. Základy videnia ako takého a techniky spoločné pre všetky jeho podoblasti. Od základov tvorby obrazu, jej geometrie a matematického aparátu na prácu s ňou, významu šošoviek a ich vplyvu, kalibrácii kamery až po prístupy určovania polohy. Pochopenie fungovania systémov na určovanie hĺbky a polohy využitím rôznych prístupov. Taktiež transformácie pri projekcii sú niečo bez čoho počítačová grafika nedokáže existovať a porozumenie základných kameňov je obrovskou výhodou. To nie je všetko, nielen teoreticky ale najmä práca a implementácia navrhnutých postupov, oboznámenie sa s knižnicami a skúmanie výsledkov z reálnych experimentov, toto všetko mi prinieslo obohatenie v podobe vedomostí a skúseností.

Pri vypracovávaní som využíval knižné a internetové zdroje, konzultácie s vedúcim práce a na projektovej výučbe. Nápomocná bola kniha od pána Szeliského (v [13]), ktorá sa už chápe ako štandard, knihy [20] [17] a zverejnené prednášky a študijné materiály z kurzov počítačového videnia z univerzít z Edinburgu [21], Pensylvánie [22] a Floridy [23]. Počas práce som sa taktiež oboznámil s viacerými modulmi knižnice OpenCV a Qt a študoval ich dokumentáciu [24] [25].

Aby sme mali k dispozícii kontrolované prostredie, v ktorom dokážeme s jemnosťou ovplyvňovať všetky jeho parametre, bolo našou motiváciou vytvoriť umelý dataset. Jasne sme definovali kritéria, čo takýto dataset musí spĺňať, a akú by mal mať štruktúru. Bol vybraný vhodný nástroj na modelovanie a *renderovanie* takejto 3D scény. Navrhli sme základnú množinu generovaných datasetov tak, aby pokrývali relevantné situácie. Implementovali sa pomocné nástroje na vytvorenie datasetu v tvare v akom sme požadovali. Tieto nástroje a skripty zahŕňovali: získavanie parametrov zo scény využitím API prístupu do jadra modelovacieho nástroja, ukladanie parametrov v definovanej podobe do XML súborov, ich parsovanie, nástroj na detekciu dopravnej značky pre úpravu anotácii a nástroj na kontrolu validity obrázkov a celého datasetu.

Implementovali sa 3 základné predstavené prístupy určovania polohy založené na stereo korešpondencii, rekonštrukcii trajektórie svetelných lúčov a využitím uhlov, pod ktorými pozorujeme objekt. Okrem samotných metód bolo navrhnuté jadro simulátora experimentov, ktorý má uľahčiť experimentovanie – najmä konfiguráciu metód, spôsob prepojenia metód s rôznymi spôsobmi prístupu k dátam, jednoduchší zber údajov z experimentov a réžie spojenej s prestavovaním programu na iný typ experimentu. Jadro bolo rozšírené na prácu s úlohou určovania polohy s rozšírenou funkcionálnou generovaním chýb, vytváraním množiny experimentov, ladenia, vizualizácie a navrhnutou metrikou vyhodnocovania. Okrem toho boli implementované ďalšie pomocné knižnice na prácu s datasetom, na podporu vykresľovania, plotovania dát a exportu výsledkov.

Implementované metódy boli rozsiahlo experimentálne skúmané a výsledky podrobne analyzované. Vo výsledku sa ukázala metóda rekonštrukcie svetelných lúčov za najpresnejšiu z pohľadu produkovanej chyby a jej rozptylu. Na testovaných datasetoch dokázala odhadnúť polohu objektu v priemere na 50 centimetrov. Jedno zo základných pozorovaní je, že vzdialenosť od objektu zásadne ovplyvňuje presnosť. Nepresnosť v GPS polohe spôsobila výrazne chyby pri metóde stereo korešpondencie a najlepšie takýto typ chyby zvládala metóda triangulácie cez uhly. Možno hovoriť o chybe 3,5 metra na chybu 1 metra GPS okrem metódy stereo korešpondencie, ktorej výsledky sú rádovo horšie. Nepresná hodnota natočenia kamier najviac vplýva na chybu metódy triangulácie cez uhly. Skreslenie o 1° spôsobuje chybu v priemere až o 12 metrov. Ďalším skúmaným typom chyby bola nepresnosť v polohe bounding boxu. Aj posun o 1 pixel v polohe

bounding boxu zhorší presnosť metód v priemere o 3 metre. Skúmaná bola aj presnosť, s akou dokážeme odhadnúť parametre kamery procesom kalibrácie. Ďalšie zistenie je, že konfigurácia kamier taktiež výrazne ovplyvňuje presnosť metód. Odporúčanie je použiť pre bližšie vzdialenosti kamery, ktoré sú vzájomne natočené, určite je však téma z kapitoly 5.5 námet na ďalšie skúmanie. Trajektória pohybu taktiež ovplyvňovala presnosť. Najväčšiu chybovosť sme pozorovali pri cestách so zákrutami a nepravidelným stúpaním a klesaním.

Zo zadania bola cieľom analýza a návrh vhodných princípov na určovanie polohy, vygenerovanie umelých datasetov, vytvorenie simulátora, implementácia metód a ich experimentálne preskúmanie. Celý prístup k problematike som sa snažil poňať profesionálne a zodpovedne.

# Zoznam použitej literatúry

---

- [1] „Self-Driving Ubers,“ 2017. [Online]. Available: <https://www.uber.com/cities/pittsburgh/self-driving-ubers/>. [Cit. 2018].
- [2] „Which companies are making driverless cars?,“ [Online]. Available: <https://www.techworld.com/picture-gallery/data/-companies-working-on-driverless-cars-3641537/>. [Cit. 2018].
- [3] „HoloLens,“ [Online]. Available: <https://www.microsoft.com/en-us/hololens/why-hololens>. [Cit. 2018].
- [4] L. Matthies, M. Maimone, A. Johnson a e. al., „Computer Vision on Mars,“ *International Journal of Computer Vision*, zv. 67, p. 75, 2007.
- [5] „Facebook Creates Software That Matches Faces Almost as Well as You Do,“ 2014. [Online]. Available: <https://www.technologyreview.com/s/525586/facebook-creates-software-that-matches-faces-almost-as-well-as-you-do/>. [Cit. 2018].
- [6] M. Pollefeys, *Visual 3D Modeling from Images*, Chapel Hill, USA: University of North Carolina, 2010.
- [7] „3D reconstruction,“ [Online]. Available: [https://en.wikipedia.org/wiki/3D\\_reconstruction](https://en.wikipedia.org/wiki/3D_reconstruction). [Cit. 2018].
- [8] „Proximity sensors,“ [Online]. Available: [https://en.wikipedia.org/wiki/Proximity\\_sensor](https://en.wikipedia.org/wiki/Proximity_sensor). [Cit. 2018].
- [9] L. Paya, A. Gil a O. Reinoso, „A State-of-the-Art Review on Mapping and Localization of Mobile Robots Using Omnidirectional Vision Sensors,“ *Journal of Sensors*, p. 20, 2017.
- [10] „Random dot stereogram,“ [Online]. Available: [https://en.wikipedia.org/wiki/Random\\_dot\\_stereogram](https://en.wikipedia.org/wiki/Random_dot_stereogram). [Cit. 2018].
- [11] „Depth Map,“ [Online]. Available: [https://en.wikipedia.org/wiki/Depth\\_map](https://en.wikipedia.org/wiki/Depth_map). [Cit. 2018].
- [12] „Feature Detection and matching,“ [Online]. Available: [http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL\\_COPIES/OWENS/LECT11/node5.html](http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT11/node5.html). [Cit. 2018].
- [13] R. Szeliski, *Computer Vision: Algorithms and Applications*, Springer, 2010.

- [14] A. Fusiello, „Stereo Matching: an Overview,“ [Online]. Available: <http://www.diegm.uniud.it/fusiello/teaching/mvg/stereo.pdf>. [Cit. 2018].
- [15] „Theory of Light,“ [Online]. Available: <http://nightlase.com.au/education/optics/light.htm>. [Cit. 2018].
- [16] „The KITTI Vision Benchmark Suite,“ [Online]. Available: <http://www.cvlibs.net/datasets/kitti/>. [Cit. 2018].
- [17] G. Bradski a A. Kaehler, Learning OpenCV: Computer vision with the OpenCV library, O'Reilly Media, Inc., 2008.
- [18] „OpenCV Performance Test,“ [Online]. Available: <http://aishack.in/tutorials/opencv-vs-vxl-vs-lti-performance-test/>. [Cit. 2018].
- [19] D. L. Baggio, Mastering OpenCV with Practical Computer Vision Projects, Packt Publishing Ltd, 2012.
- [20] R. Hartley a A. Zisserman, Multiple View Geometry in computer vision Second Edition, New York: Cambridge University Press, 2003.
- [21] H. Bilen, „Course Computer Graphics,“ University of Edinburgh, 2017. [Online]. Available: <https://www.inf.ed.ac.uk/teaching/courses/cg/index2017.html>. [Cit. 2018].
- [22] R. Collins, „Course Computer Vision I,“ [Online]. Available: <http://www.cse.psu.edu/~rtc12/CSE486/>. [Cit. 2018].
- [23] U. Bagci, „Course Computer Vision,“ 2017. [Online]. Available: <http://www.cs.ucf.edu/~bagci/teaching/computervision17.html>. [Cit. 2018].
- [24] „OpenCV Documentation,“ [Online]. Available: <https://docs.opencv.org/2.4/index.html>. [Cit. 2018].
- [25] „Qt Documentation,“ [Online]. Available: <https://doc.qt.io/qt-5.10/>. [Cit. 2018].

# Prílohy

---

Na priloženom médiu sú k dispozícii:

- zdrojové kódy implementovaných knižníc (číslo prílohy 1)
- ukážky dát z datasetov (číslo prílohy 2)
- experimentálne výsledky (číslo prílohy 3)