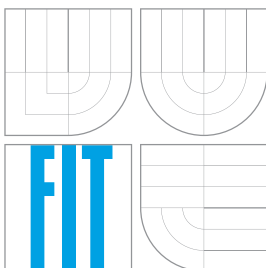


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

REKONSTRUKCE 3D MODELU PROSTŘEDÍ A LOKALIZACE KAMERY

3D MODEL RECONSTRUCTION AND CAMERA LOCALIZATION

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. TOMÁŠ VAHALÍK

VEDOUcí PRÁCE

SUPERVISOR

Ing. VÍTEZSLAV BERAN, Ph.D.

BRNO 2014

Abstrakt

Tato práce se zabývá rekonstrukcí 3D modelu prostředí ze sady fotografií a následnou lokalizací kamery pozorovatele. Popisuje základní principy a techniky využívané pro tvorbu modelů prostředí a také pro následný odhad pozice kamery ze sady 2D bodů kamery do 3D bodů modelu. Dále zkoumá vliv parametrů na kvalitu rekonstrukce a možnosti lokalizace. Porovnává kvalitu deskriptorů při tvorbě modelu a na základě lokalizace umožňuje realizovat rozšířenou realitu.

Abstract

This thesis focuses on reconstruction of 3D environment model from a set of photographs followed by camera localization. It describes basic principles and techniques used to create environmental models and techniques for camera pose estimation from 2D camera points to 3D model points. It also examines the influence of parameters on the quality of reconstruction and the possibilities of localization. It compares the quality of the descriptors in the process of creation of the model and based on localization it allows to implement augmented reality.

Klíčová slova

Lokalizace kamery, model prostředí, 3D rekonstrukce, klíčové body, rozšířená realita, sledování pohybu kamery

Keywords

Camera localization, environment model, 3D reconstruction, augmented reality, key-points, camera tracking

Citace

Tomáš Vahalík: Rekonstrukce 3D modelu prostředí
a lokalizace kamery, diplomová práce, Brno, FIT VUT v Brně, 2014

Rekonstrukce 3D modelu prostředí a lokalizace kamery

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Vítězslava Berana, Ph.D..

.....
Tomáš Vahalík
27. května 2014

Poděkování

Rád bych poděkoval svému vedoucímu, Ing. Vítězslavu Beranovi, Ph.D., za trpělivou pomoc, cenné rady a připomínky při řešení této diplomové práce. Dále také rodině za podporu po celou dobu mých studií a v neposlední řadě přítelkyni Markétě, která mi dodávala motivaci k dokončení této práce.

© Tomáš Vahalík, 2014.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	3
2	Zpracování obrazu	4
2.1	Klíčové body	4
2.2	SIFT	5
2.3	SURF	7
2.4	Hledání korespondencí mezi klíčovými body	8
3	Registrace a lokalizace	11
3.1	Kalibrace kamery	11
3.2	Registrace obrazu	12
3.3	3D rekonstrukce scény	13
3.4	Lokalizace kamery	15
3.5	3D datové modely	16
4	Návrh systému	18
4.1	Specifikace cíle	18
4.2	Podobné systémy	18
4.3	Návrh lokalizátoru	19
4.4	Návrh modelu	20
4.5	Návrh systému pro trackování pohybu kamery	21
4.6	Propojení lokalizace a sledování pohybu kamery	21
4.7	Budoucí možná rozšíření	23
5	Model	24
5.1	Tvorba datové sady	24
5.2	Tvorba modelu	24
5.3	Nástroje	25
5.4	Experimenty	27
5.5	Validace tvorby modelů	28
6	Sledování pohybu kamery	33
6.1	Realizace trackeru	33
6.2	Reference mezi klíčovými body	34
6.3	Experimenty	35
6.4	Reference GFTT a KP	35
6.5	Stabilita systému	36
6.6	Rychlost systému	36

7	Lokalizátor kamery	37
7.1	Realizace	37
7.2	Realizace lokalizátoru	38
7.3	Experimenty	41
7.4	Kvalita lokalizace	41
7.5	Rychlost systému	45
8	Závěr	46

Kapitola 1

Úvod

Rozšířená realita, jakožto nástroj dokreslování informací do obrazu pozorovaného světa, je stále poměrně mladým odvětvím informačních technologií, avšak velmi rychle se rozmáhajícím. Problémem při dokreslování informací o pozorované scéně může být úspěšné nalezení potřebných korespondencí, o nichž je potřeba podat potřebné informace. Pouze na základě vizuálních korespondencí může systém selhávat v případech, kdy je obrazová kvalita nedostatečná nebo je potřeba pokrýt větší prostor. Jedním řešením může být využití markerů – vizuálních značek, které určují vykreslované informace o dané scéně, zároveň také slouží pro udržení korespondencí. Bohužel ne všude lze takovýchto značek využít. Proto je potřeba nalézt jiný přístup, jenž nevyužívá vizuálních značek.

Metody, které využívají pouze video vstup bez přidání značek, jsou většinou úzce spojeny s robotikou a autonomní navigací robotů ve známém či neznámém prostředí. Mezi tyto přístupy patří například SLAM (*Simultaneous localization and mapping*) nebo PTAM (*Parallel tracking and mapping*). Robot pohybující se v neznámém prostředí tento prostor zkoumá (mapuje) a zároveň aktuálně zmapovaného prostředí využívá pro lokalizaci. Díky tomu může nezávisle na člověku daný prostor prozkoumat a zjistit, že je na místě, na kterém již byl.

Přístupy z robotiky mohou být inspirací pro navržení metody pro lokalizaci kamery v prostoru pro potřeby rozšířené reality v rámci určitého prostředí. Výhodou tohoto přístupu oproti prvnímu zmíněnému, kdy se využívá lokalizace pomocí vizuálních značek, je to, že není potřeba prostředí nijak modifikovat pro potřeby lokalizace. Tento postup může být vhodný zejména v prostorách, kde označování míst vizuálními značkami není možné, např. různé památky.

Proces tvorby takové rozšířené reality bude mírně složitější než s jednoduchým použitím markerů, ovšem s využitím vhodných nástrojů bude nabízet mnohem širší možnosti využití ve větších prostorách.

Tímto bylo nastíněno možné využití této práce, která si klade za cíl lokalizaci v rámci modelu prostředí. Nejdříve budou popsány metody hledání korespondencí a tvorba modelu. Dále budou vysvětleny principy lokalizace ve 3D prostoru s pomocí 2D bodů. V další kapitole jsou shrnuty nástroje a techniky využívané v průběhu práce a následuje popis návrhu samotného systému. Následně jsou popsány jednotlivé části navrženého systému z pohledu realizace a provedených experimentů.

Kapitola 2

Zpracování obrazu

V této kapitole budou popsány techniky z oblasti počítačového vidění, které se používají ve spojení nejen s rozšířenou realitou. Počítačové vidění je disciplína v informačních technologiích, zabývající se získáváním informací, následnou analýzou a zpracováním obrazových dat z různých zdrojů vizuálních informací, jako například webových kamer, fotoaparátů či jiných zařízení. Jedná se zejména o techniky týkající se klíčových bodů, které jsou velmi důležité při hledání korespondencí mezi obrazy, což je nedílná součást také rekonstrukce prostředí.

2.1 Klíčové body

Klíčové body můžeme chápat jako význačná místa v obraze – například rohy, hrany, průsečíky přímek. Co budeme považovat za význačný příznak obrazu je závislé na dané úloze, kterou potřebujeme řešit. Pro vyhledávání takových významných bodů v obraze se využívají takzvané detektory. U význačných bodů můžeme hodnotit jejich kvalitu, která závisí zejména na zvoleném detektoru. Jak lze hodnotit kvalitu význačného bodu? Opět je zde hlavní otázkou, jaké priority pro detekci máme, od toho lze odvodit různé důležité vlastnosti[17]:

- Robustnost – náchylnost na šum v obraze, rozmazání, diskretizaci a podobně.
- Rozlišitelnost – jednoznačné matematické vyjádření, které zajistí porovnatelnost v databázi příznaků.
- Invariantnost vůči natočení, zkosení a jiným transformacím.
- Výkonnost – zda-li je detekce dostatečně rychlá.

Jak již bylo zmíněno výše, záleží zejména na dané úloze, která má určité nároky – například algoritmy pro sledování změn pracující v reálném čase využívají zejména detektory rohů, které jsou významné pro velký výkon, avšak za cenu nižší robustnosti. Na straně druhé mohou být algoritmy, které vyžadují zejména robustnost a invariantnost vůči fotometrickým či geometrickým transformacím. Zde se jedná zejména o vyhledávání korespondencí mezi různými obrazy. Příklad lze vidět na obrázku 2.4 kde zelené přímky spojují nalezené korespondence mezi detekovanými klíčovými body. Mezi nejznámější metody pro získání a popis klíčových bodů, které jsou velmi robustní, patří SIFT a SURF.

2.2 SIFT

SIFT je zkratkou anglického *Scale-Invariant Feature Transform*, metoda poprvé představená Davidem Lowe v roce 2004. Popis technik vychází z [13]. SIFT je, jak již z názvu vyplývá, invariantní vůči změně měřítka, také ale vůči rotaci nebo změnám světelných podmínek. Dále je také odolný vůči šumu. Díky těmto vlastnostem spadá mezi nejrobustnější metody pro získání a popis klíčových bodů. Tato robustnost je ale vykoupena vysokou výpočetní náročností.

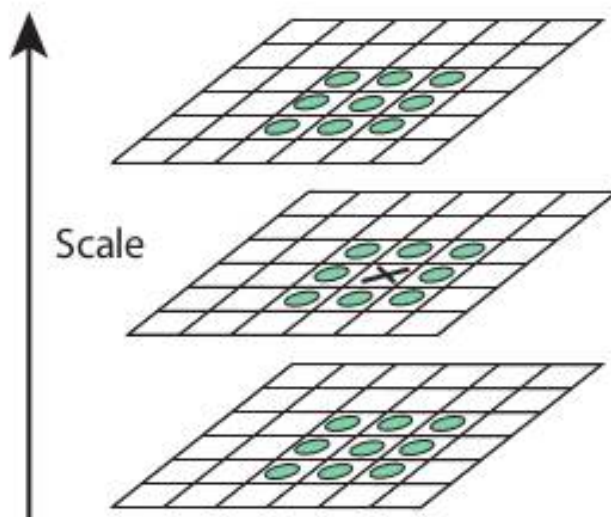
Každý významný bod je popsán deskriptorem, který je reprezentován 128-rozměrným vektorem celých kladných čísel.

Detekce klíčových bodů

Prvním krokem pro detekci klíčových bodů je využito dvojí aplikace Gaussova filtru s různým σ , čímž aproximujeme získání druhé derivace daného obrazu. Aproximace je používána zejména kvůli výkonnosti. Tento proces je aplikován pro různé oktávy obrazu v Gaussovské pyramidě. Změnou měřítka o oktávu se myslí zdvojnásobení velikosti jádra daného filtru, jehož efekt je přibližně ekvivalentní snížení rozlišení obrazu na polovinu. Počet takových oktáv je dán vztahem $\log_2(\min(w, h))$, kde w je šířka obrazu a h je jeho výška – čímž dosáhneme vyhledávání klíčových bodů pro všechny možné velikosti. Tím je zajištěna invariance vůči měřítku.

Tímto procesem vzniknou tzv. rozdílové obrazy (DoG – *difference of gaussians*) ve kterých se dále hledají maxima a minima, jež mohou být považována za potenciální klíčové body.

Tato maxima a minima jsou vybrána a následuje porovnání hodnoty pixelu s osmi okolím a následně také s devíti pixely v dalším měřítku. Pokud bude hodnota pixelu minimem nebo maximem v porovnání se všemi pixely, pak lze říct, že daný klíčový bod je nejlépe reprezentován v daném měřítku, jak lze vidět na obrázku 2.1.



Obrázek 2.1: Křížek znázorňuje lokální minimum/maximum v porovnání s ostatními měřítky. ²

Popis klíčových bodů

Takto jsou nalezeny klíčové body, pro které je potřeba následně přiřadit orientaci. V tomto kroku se zajišťuje invariantnost vůči rotaci. Nejprve je potřeba určit orientaci daného klíčového bodu a to se udělá tak, že se vezme okolí daného klíčového bodu a v závislosti na daném měřítku se vypočítá hodnota gradientu.

Dále se vypočte samotný deskriptor, vezme se 16×16 okolí kolem klíčového bodu, které se rozdělí na 16 pod-bloků o velikosti 4×4 . Pro každý pod-blok se utvoří 8 bin orientovaný histogram - čímž vznikne požadovaný 128 bin hodnotový vektor.

Také se provádí další kroky, které ještě zvyšují robustnost vůči jasovým změnám v obrazech (změnám osvětlení) a tak podobně.



Obrázek 2.2: Vizualizace detekovaných klíčových bodů s pomocí SIFT metody. Velikost kruhu znázorňuje měřítko, orientace je znázorněna spojením středu s okrajem.⁴

Jakmile jsou detekované body popsány, lze je vizualizovat. Taková vizualizace poté může názorně ukázat v jakém měřítku byl daný klíčový bod nalezen a jaká je jeho orientace. Takto vizualizované klíčové body lze vidět na obrázku 2.2.

²<http://docs.opencv.org/>

⁴<http://docs.opencv.org/>

2.3 SURF

SURF neboli *Speeded-Up Robust Features* je metoda využívána pro získání a popis klíčových bodů vycházející z metody SIFT, která byla příliš pomalá. Následující informace vycházejí z [2] a [3].

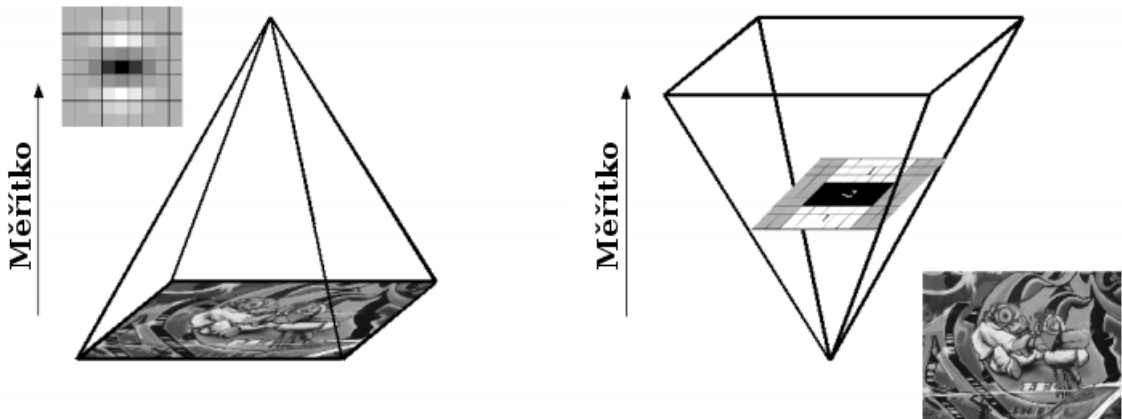
Detekce klíčových bodů

Detektor klíčových bodů je založen na Hessově matici, která pro bod $x = (x, y)$ v obraze I je $\mathcal{H}(x, \sigma)$ v x s měřítkem σ definována takovýmto předpisem:

$$\mathcal{H}(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix} \quad (2.1)$$

kde $L_{xx}(x, \sigma)$ je konvolucí Gaussových derivací druhého řádu s obrazem I v bodě x , podobně je tomu tak i pro $L_{xy}(x, \sigma)$ a $L_{yy}(x, \sigma)$.

Oproti metodě SIFT, která aproximuje pomocí dvojího Gaussovského rozmazání, využívá SURF detektor pro aproximaci tzv. box filtry, které dokáží ještě rychleji aproximovat deriváty Gaussových derivací druhého řádu s využitím integrálního obrazu. Díky tomu není potřeba sestavovat obrazovou pyramidu postupným zmenšováním obrazu, ale naopak lze takový filtr postupně zvětšovat a vždy aplikovat přímo na původní obraz, jak je znázorněno na 2.3.



Obrázek 2.3: Namísto redukce velikosti obrazu, jak lze vidět na levém obrázku, lze díky integrálnímu obrazu zvyšovat měřítko filtru (vpravo). [2]

Popis klíčových bodů

Podobně jako je tomu u metody SIFT, tak také u této metody je nejprve potřeba určit orientaci, kterou lze získat z informací o oblasti kolem daného klíčového bodu. Dále je vytvořena čtvercová oblast orientovaná daným směrem a z této oblasti se vypočte deskriptor pro daný klíčový bod. Tento deskriptor lze získat měřením změny lokální intenzity ve vertikálním směru d_y a samozřejmě ve směru horizontálním d_x . Daná velikost okolí, která se analyzuje, je násobkem měřítka, pro daný klíčový bod, konkrétně 20σ . Tato oblast se dále

rozdělí na bloky 4×4 . Pro každý takovýto pod-blok se vypočítá odezva Haarovy vlnkové funkce (zjednodušeně půjde o odezvu d_x v horizontálním směru a d_y ve směru vertikálním) pro 5×5 rovnoměrně rozmístěných bodů. Tyto odezvy jsou následně sečteny přes všechny pod-oblasti, čímž se vytvoří první sada hodnot pro vektor příznaků. Dále pro zajištění polaridy změny intenzity se extrahuje absolutní hodnota součtu jednotlivých odezev.

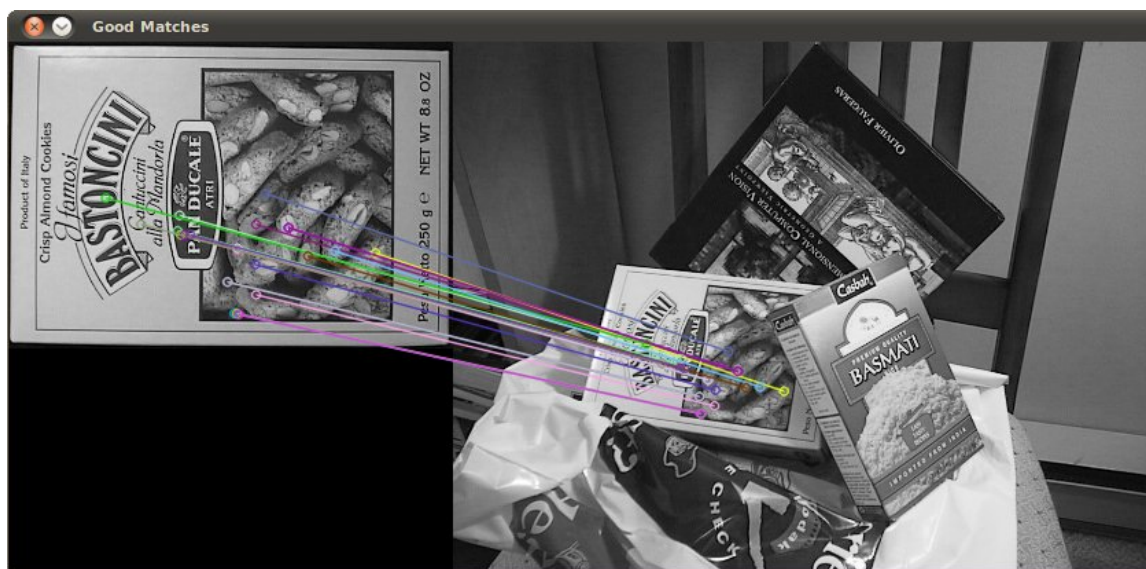
$$v = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|) \quad (2.2)$$

Konkatenací přes všechny 4×4 pod-oblasti dostaneme vektor o délce 64.

2.4 Hledání korespondencí mezi klíčovými body

Dalším nezbytným krokem, jakmile máme získány a popsány klíčové body z různých obrazů, je nalezení jejich korespondencí. Existuje mnoho metod pro hledání korespondencí mezi význačnými body. Výběr metody závisí zejména na aktuální úloze a použití těchto korespondencí.

Příklad nalezených korespondencí mezi jednotlivými body lze nalézt na obrázku 2.4.



Obrázek 2.4: Nalezené shody mezi klíčovými body.⁵

Metody SURF a SIFT, jak bylo vysvětleno výše, mají deskriptory navržené tak, aby je šlo jednoduše porovnávat na základě jejich **Euklidovské vzdálenosti**. Čím nižší tato vzdálenost je, tím podobnější tyto body jsou.

Vyhledávání hrubou silou

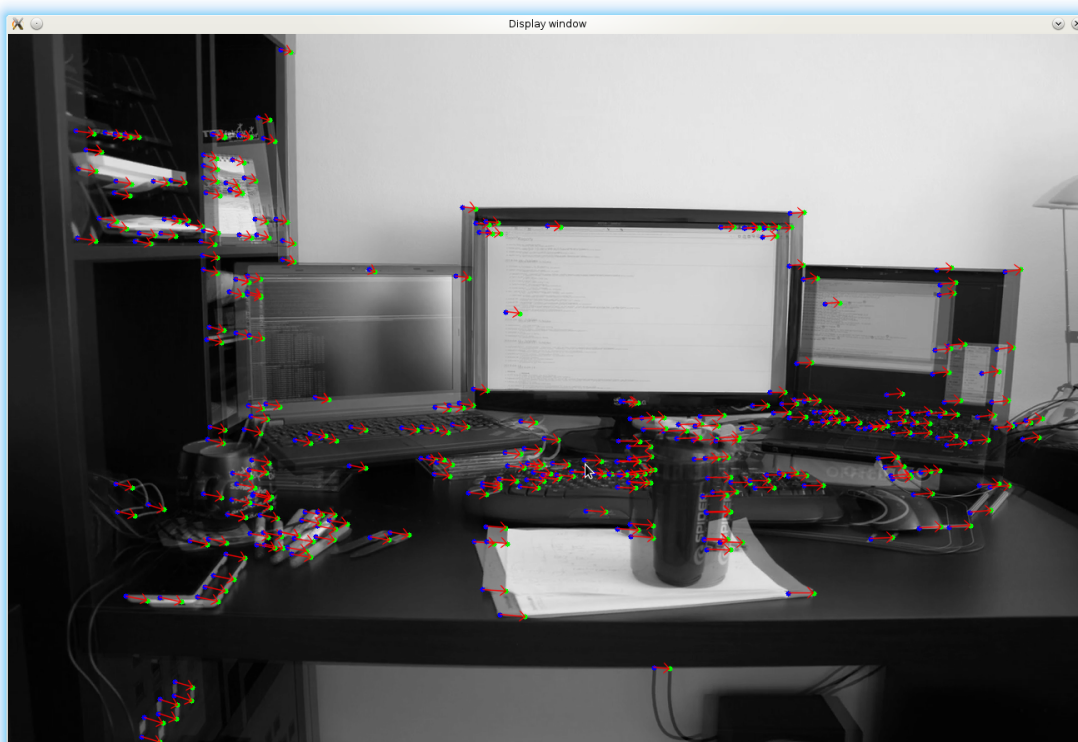
Nejtriviálnějším přístupem pro vyhledávání korespondencí je jejich vzájemné porovnávání. Pro každý význačný bod (deskriptor) ze vstupní sady deskriptorů se hledá vždy jeden nejbližší v celé sadě druhé.

⁵<http://docs.opencv.org/>

Následně se je potřeba nastavit prahovou hodnotu pro danou vzdálenost (jejich maximální možnou vzdálenost) a na základě této vzdálenosti brát pouze takové body z druhého obrazu, jejichž vzdálenost je menší než hodnota daného prahu.

Vyhledávání nejbližšího souseda

Dalším přístupem je hledání nejbližšího souseda – *nearest neighbour* – v okolí klíčového bodu. Nejprve se zpracují vstupní data a z nich se utvoří *vyhledávací strom*. Následně při vyhledávání korespondencí je využito této stromové struktury pro hledání vhodných párů na základě jejich vzdálenosti. I zde pro dosažení lepších výsledků, je možno stanovit prahovou hodnotu, na základě které budou nalezené korespondence dále otestovány a která bude určovat maximální vzdálenost nejbližšího souseda.



Obrázek 2.5: Sledování klíčových bodů.

Sledování klíčových bodů

Další možnou variantou, jak zjistit korespondující si klíčové body, je jejich sledování, nebo-li *feature tracking*. S tímto také úzce souvisí výpočet optického toku, což může být další přístup pro sledování pohybu kamery. Hlavní rozdíl mezi sledováním klíčových bodů a výpočtem optického toku také tkví v tom, že výpočet optického toku probíhá pro každý pixel a tyto operace lze provést na celém snímku [9].

Tato metoda je založena na tom, že nejprve detekujeme klíčové body, které mohou být přesně sledovány za pomoci některé z metod lokálního vyhledávání. Často používaný přístup pro lokální vyhledávání je např. za pomoci nejmenších čtverců. Tato metoda je

vhodná více pro hledání korespondencí mezi po-sobě jdoucími snímky, jelikož v těchto případech nedochází k velkému posuvu a tudíž nebudou klíčové body v následujícím snímku příliš vzdáleny od pozice ve snímku původním, jak lze vidět na obrázku 2.5 – modrý bod znázorňuje původní pozici v obraze, šipka ukazuje kam se bod přesunul a zelený bod je novou pozicí v následujícím obrázku. Tento obrázek je pro názornou ilustraci složen ze dvou po sobě jdoucích snímků z kamery, kde pozice bodů odpovídají reálným pozicím v daných obrazech.

Kapitola 3

Registrace a lokalizace

V této kapitole budou představeny techniky vhodné pro registraci obrazů kamery a také principy využívané pro lokalizaci ve 3D prostoru. Tyto techniky a principy jsou klíčové pro správnou rekonstrukci modelu prostředí a následnou lokalizaci. Jsou vysvětleny klíčové vlastnosti při tvorbě 3D struktury na základě 2D korespondujících si bodů ve více obrazech.

3.1 Kalibrace kamery

Kalibrovaná kamera je klíčem ke správnému určení vztahu mezi reálným světem a výslednou fotografií či videem tvořeným právě touto kamerou.

Kalibrace kamery je proces výpočtu parametrů modelu dírkové kamery aproximující model kamery, která pořizuje obrázky nebo video, které je nějakým způsobem zkreslené. Jelikož se ve fotoaparátech a kamerách většinou používá objektiv, založený na sadě čoček, ať už spojek nebo rozptylek, proto je velmi nákladné vytvořit takové kamery či fotoaparáty, které mají minimální zkreslení.

Aby popis kamery byl co nejbližší realitě, byl zaveden pojem *vnitřní parametry* kamery – *intrinstické parametry* [14]. Ty umožňují přepočítat polohu bodu ze systému metrického na polohu bodu v pixelech, kde navíc bude počátek soustavy souřadnic v levém horním rohu obrazu. Transformace souřadného systému obrázku (v metrickém systému) na pixelové souřadnice může být proveden jak je ilustrováno na 3.1.

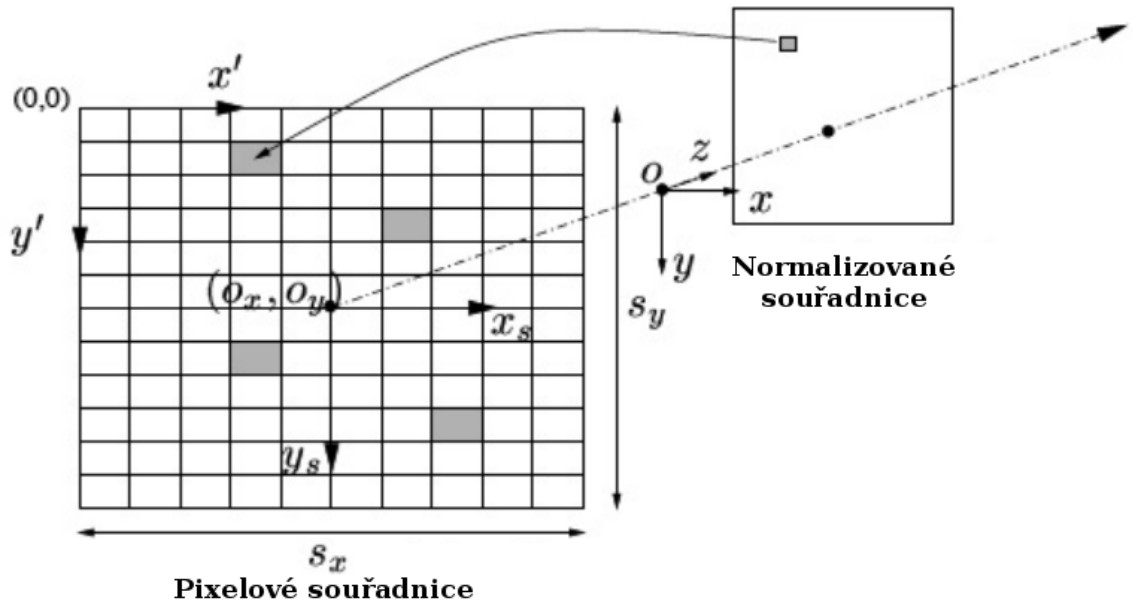
Intrinstické parametry získáme tak [23], že nejprve přepočítáme souřadnice z metrického systému do polohy udávané v pixelech s pomocí následující matice.

$$\begin{bmatrix} x_s \\ y_s \end{bmatrix} = \begin{bmatrix} s_x & \alpha \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (3.1)$$

kde hodnoty s_x a s_y závisí na velikosti pixelů v metrické soustavě. Dále hodnota α udává zkosení pixelu na čipu, které se u dnešních moderních čipů velmi blíží k nule, proto ji lze pro zjednodušení zanedbat.

Dále je nutné posunutí počátku souřadné soustavy ze středu obrázku do levého horního rohu, což se provede tak, že k hodnotám x_s a y_s připočteme hodnoty o_x a o_y , které nám udávají vzdálenost hlavního bodu od počátku soustavy v pixelech, jak lze vidět na následujícím maticovém tvaru:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (3.2)$$



Obrázek 3.1: Transformace z obrazových souřadnic do pixelových souřadnic. [22]

Z tohoto vztahu lze vyvodit, že matice vnitřních parametrů bude obsahovat čtyři parametry (v případě že zanedbáme zkosení pixelu na čipu) a bude v následujícím tvaru:

$$A = \begin{bmatrix} s_x & \alpha & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

Extrinsické parametry, neboli *vnější parametry*, jsou parametry udávané R, T které určují transformaci souřadného systému ze 3D souřadnic světa do 3D souřadnic kamery. Definují nám pozici středu kamery a směr kamery v souřadném systému světa. [23]

Z těchto předpokladů lze odvodit projekční transformaci mezi souřadnicemi 3D bodu v prostoru a homogenními souřadnicemi 2D bodu obrazu vyjádřeného v pixelech.

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = A \begin{bmatrix} R & T \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad (3.4)$$

3.2 Registrace obrazu

Registrace obrazu spadá mezi techniky využívané v počítačovém vidění, kdy jde o proces spojování nebo hledání překryvu dvou či více obrazů jedné scény, které byly utvořeny v různou dobu, z různých pozorovacích úhlu nebo dokonce s pomocí různých pořizovacích zařízení.

Prvním krokem pro registraci obrazu je detekce význačných bodů. Lze použít některou z metod popsanych v 2.1. Dále je potřeba nalézt korespondence mezi těmito body, například jak bylo popsáno v 2.4. Jakmile jsou nalezeny korespondence, je potřeba spočíst odhad transformačního modelu. Transformační model lze vyjádřit pomocí parametrů mapovací

funkce mezi původním a novým snímkem, kdy požadované parametry lze vypočítat na základě získaných korespondencí mezi jednotlivými rysy.

Mezi takové mapovací funkce patří kupříkladu homografie.

Homografie je projektivní transformace mezi dvěma perspektivami, ve kterých jsou přímky mapovány na přímky. Homografii si lze představit jako mapování bodů z jedné plochy na druhou $x_i \leftrightarrow x'_i$.

Tento vztah lze vyjádřit jako

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \quad (3.5)$$

Pro výpočet homografie je potřeba znát alespoň 4 korespondující body, reálně se ovšem využívá co největší množství korespondenčních bodů, aby se dosáhlo co největší robustnosti.

RANSAC

Algoritmus RANSAC, neboli *RANdom SAmple Consensus* [5], česky *shoda náhodných vzorků*, je iterativní metoda pro získání (odhad) parametrů různých matematických modelů, hojně využívána v oblasti počítačového vidění. Spadá do takzvané kategorie robustních estimátorů. S metodou RANSAC úzce souvisí zejména dva důležité pojmy:

- **Inlier** – označuje vzorek dat, který náleží danému modelu,
- **outlier** – označuje vzorek dat, který nenáleží do daného modelu.

Z toho lze usuzovat, že RANSAC je využíván v modelech, který se skládá ze vzorku dat, který obsahuje i data taková, která do tohoto modelu nepatří.

Metoda RANSAC pracuje následovně:

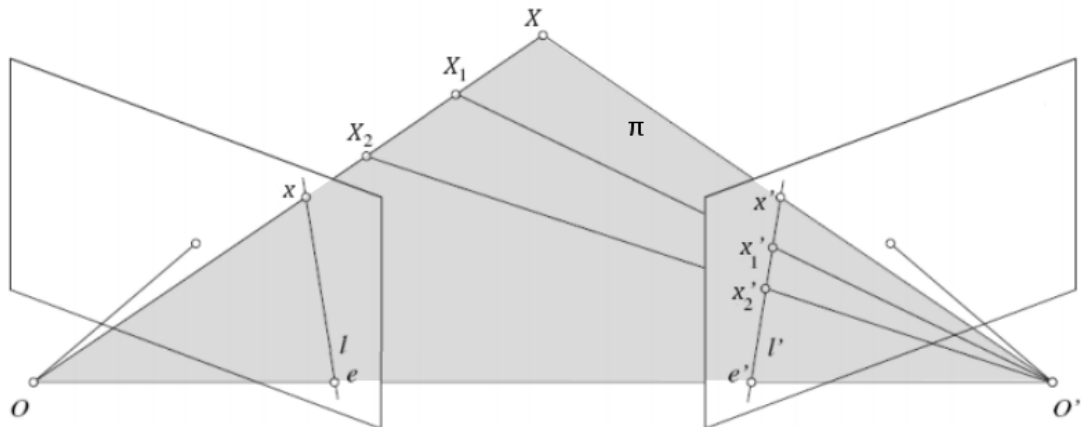
1. Ze vstupních dat vyber náhodný vzorek.
2. Vypočti parametry modelu.
3. Ověř platnost vypočteného modelu na všech datech.
4. Pokud klesne pravděpodobnost nalezení lepšího modelu pod stanovený limit, tak skončí, jinak jdi na bod 1.

Náhodně vybraný vzorek bývá určen minimální velikostí, aby postačil pro získání parametru modelu.

Pro více informací o této metodě, lze nahlédnout do [5], odkud také bylo čerpáno.

3.3 3D rekonstrukce scény

V této části budou nastíněny principy podstatné pro 3D rekonstrukci scény. 3D rekonstrukcí scény rozumíme proces tvorby 3D modelu z obrazových dat snímaných kamerou či fotoaparátem.



Obrázek 3.2: Mapování $x \rightarrow l'$. [15] .

Epipolární geometrie

Epipolární geometrie mezi dvěma pohledy popisuje geometrii průsečíků mezi obrazovými rovinami a rovinami které mají jako osu bázi (báze je přímka spojující středy kamer) [8]. Představme si scénu pozorovanou dvěma kamerami, ve které leží bod X . Na první kameře se nám bod X promítne do bodu x a na druhé kameře jako bod x' , jak lze vidět na obrázku 3.2. Rovina která je tvořena spojením středů promítání O , O' a bodem X (samozřejmě také body x a x' leží na této rovině) je nazývána epipolární rovina (π).

Vezmeme-li průsečík průmětny a báze (přímky procházející středy projekcí), získáme body e a e' .

Průsečík epipolární roviny s průmětnou se nazývá epipolára. V obrázku znázorněna jako přímka l' .

Fundamentální matice

Na základě předchozích poznatků o epipolární geometrii se podívejme na její matematické vyjádření. Fundamentální matice je právě algebraickým vyjádřením epipolární geometrie [8].

Fundamentální matice nám určuje mapování $x \rightarrow l'$, viz. obrázek 3.2. Fundamentální matice má sedm stupňů volnosti.

$$F = \begin{bmatrix} a & b & \alpha a + \beta b \\ c & d & \alpha c + \beta d \\ e & f & \alpha e + \beta f \end{bmatrix} \quad (3.6)$$

Dále platí následující vztahy:

$$\begin{aligned} Fx &= l' \\ F^T x' &= l \\ x'^T Fx &= 0 \end{aligned} \quad (3.7)$$

S fundamentální maticí úzce souvisí také matice esenciální E . Jedná se o matici fundamentální, která má normalizované souřadnice.

3.4 Lokalizace kamery

V této sekci je popsána metoda lokalizace kamery v prostoru. Lokalizací je myšlen odhad pozice v rámci určitého prostoru, tedy získání 3D pozice ze sady 2D bodů.

Nejjednodušším způsobem nalezení pozice kamery je využití sady lineárních rovnic, kde závislost 2D pozice bodu na matici kamery je dána vztahem: [18]

$$x_i = \frac{p_{00}X_i + p_{01}Y_i + p_{02}Z_i + p_{03}}{p_{20}X_i + p_{21}Y_i + p_{22}Z_i + p_{23}} \quad (3.8)$$

$$y_i = \frac{p_{10}X_i + p_{11}Y_i + p_{12}Z_i + p_{13}}{p_{20}X_i + p_{21}Y_i + p_{22}Z_i + p_{23}} \quad (3.9)$$

kde (x_i, y_i) je měřená 2D pozice klíčového bodu a (X_i, Y_i, Z_i) jsou známé pozice ve 3D daného klíčového bodu.

Výpočet je možno provést pomocí algoritmu *direct linear transform* (DLT), neboli přímé lineární transformace [8].

Aby bylo možno vypočítat 12 neznámých v P , je potřeba mít alespoň 6 známých korespondenčních bodů mezi 2D a 3D lokacemi. Výpočet P lze provést například dle [20].

Jakmile je získáno P , je možné dále získat rigidní transformaci (takovou transformaci, která zachovává vzdálenosti mezi jednotlivými body) (R, t) a to s pomocí vnitřních parametrů kamery K . Tento vztah je dán předpisem:

$$P = K [R|t] \quad (3.10)$$

Lze předpokládat, že kalibrační matice vnitřních parametrů je známa předem a vstupní obraz, který je analyzován je zkalibrován (bez zkreslení), pak postačí pro určení polohy minimálně 3 body. Algoritmy využívané pro řešení tohoto problému jsou známy jako lineární PnP algoritmy. Základním zjištěním je, že vizuální úhel mezi párem bodů x_i a x_j ve 2D prostoru musí být shodné s úhlem mezi korespondujícími body p_i a p_j ve 3D prostoru [8].

Vezme-li se množina korespondujících si 2D a 3D bodů $\{(\hat{x}_i, p_i)\}$, kde $\hat{x}_i$ vyjadřuje jednotkové směry získané transformací 2D obrazových bodů x_i do jednotkové normy 3D směru $\hat{x}_i$ pomocí inverzní kalibrační matice K ,

$$\hat{x}_i = N(K^{-1}x_i) = \frac{K^{-1}x_i}{\|K^{-1}x_i\|} \quad (3.11)$$

Neznámé jsou vzdálenosti d_i z počátku kamery c ke 3D bodu p_i , kde $p_i = d_i\hat{x}_i + c$. Ilustrováno v obrázku 3.3.

Dále s pomocí Kosinova pravidla pro trojúhelník $\Delta(c, p_i, p_j)$ platí

$$f_{ij}(d_i, d_j) = d_i^2 + d_j^2 - 2d_id_jc_{ij} - d_{ij}^2 = 0 \quad (3.12)$$

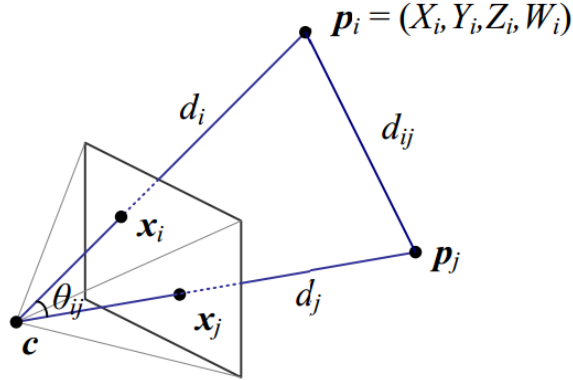
kde

$$c_{ij} = \cos \theta_{ij} = \hat{x}_i \cdot \hat{x}_j \quad (3.13)$$

a

$$d_{ij}^2 = \|p_i - p_j\|^2. \quad (3.14)$$

Jestliže je známých 5 a více korespondencí, lze vytvořit $\frac{(n-1)(n-2)}{2}$ trojic, pro získání lineárního odhadu s využitím metody SVD (*Singular Value Decomposition* – singulární dekompozice matice, více v [12]).



Obrázek 3.3: Odhad pozice pomocí přímé lineární transformace s pomocí měření vizuálních úhlů a vzdáleností mezi dvojicemi klíčových bodů. [18]

Jakmile bude zvyždřený odhad d_i vzdáleností, pak lze vygenerovat 3D strukturu tvořenou škálovaným bodem ve směru $d_i \hat{x}_i$, kterýžto může být zarovnán s 3D mračnem $\{p_i\}$ s využitím absolutní orientace, pro dosažení požadovaného odhadu polohy.

3.5 3D datové modely

V této podkapitole budou představeny datové modely. Datové modely jsou základem pro uchování dat vytvořených při 3D rekonstrukci a vhodně zvolená reprezentace může napomoci budoucímu využití daného modelu.

Typy modelů

V této části jsou popsány základní nepoužívané typy 3D modelů. Informace byly čerpány z [7], [11] a [19].

3D síť spadá mezi velmi jednoduché geometrické reprezentace, které popisují objekt jako sadu vrcholů a hran, které dohromady tvoří polygony ve 3D prostoru. Libovolná síť může obsahovat libovolné polygony. Regulární síť se skládá pouze z polygonů stejného typu. Jedna z nejhojněji využívaných sítí v počítačové grafice, je síť trojúhelníková, která, jak již její název napovídá, je kompletně tvořena z trojúhelníků. Záleží na hustotě a velikosti daných trojúhelníků jsme schopni modelovat různou úroveň detailu, od velmi hrubých modelů, po modely velmi detailní. Ukázka takového modelu je na ??.

Mračna bodů lze považovat za množinu data-bodů v určitém souřadném systému. Ve trojrozměrném systému jsou tyto body většinou definovány za pomoci souřadnic X, Y, Z a často jsou určeny k reprezentaci vnějšího povrchu nějakého objektu. Mračna bodů lze tvořit například s pomocí speciálních 3D skenerů, které automaticky získávají velké množství bodů na povrchu objektů v okolí skeneru (například *Velodyne Lidar*¹). Takto získané body se využívají k mnoha účelům, jako je například tvorba 3D CAD modelů pro výrobu, zajišťování kvality výroby, dále také pro různé vizualizace, animace a podobně.

Octree je zkrácený název pro oktalový strom. Tato struktura je využívána zejména v robotice. Kořen stromu obsahuje celou uloženou scénu a ta se postupně dělí na osm

¹<http://velodynelidar.com/lidar/lidar.aspx>

stejně velkých podčástí se společným bodem ve středu dané scény. Každou z těchto částí lze reprezentovat pomocí uzlu. Tyto vzniklé podprostory se dále dělí obdobným způsobem. Jakmile je v prostoru objekt, ukládá se do listu, který je potomkem daného uzlu, do kterého spadá. Takovéto dělení probíhá tak dlouho, dokud se v prostoru nachází nějaké objekty nebo může být stanovena jiná podmínka, která dělení ukončí. Takovou podmínkou může být třeba výška stromu nebo maximální počet objektů na uzlu.

Třidimenzionální objekt může být reprezentován jako $2^n \times 2^n \times 2^n$ třírozměrné pole pro určité celé číslo n .

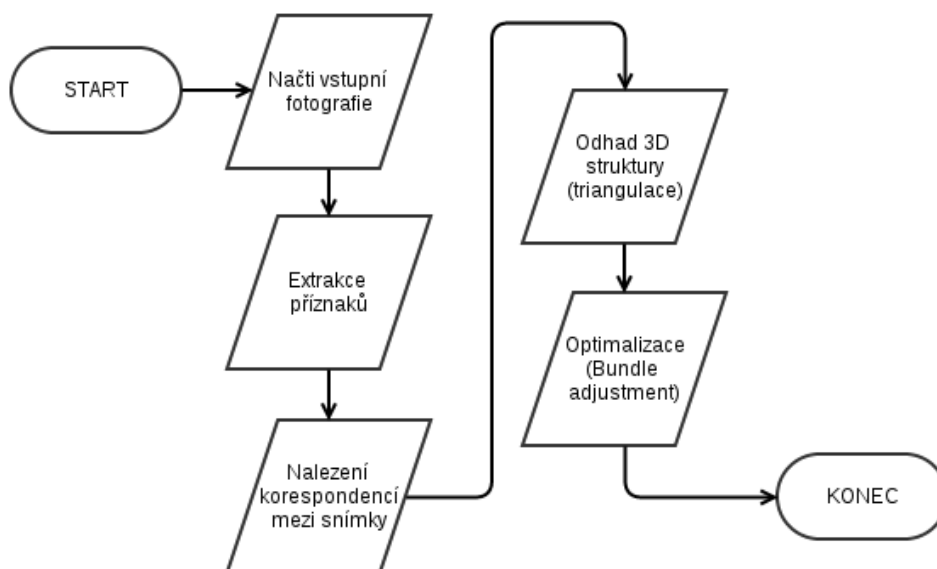
Structure from motion

Structure from motion, neboli struktura z pohybu, je řada zobrazovacích technik, které se vztahují k odhadování 3D struktury ze sady fotografií, které mohou být ve spojení se signály o lokalitě pohybu. Tyto principy jsou studovány zejména v oblasti počítačového vidění, ale také v oblasti robotiky.

Ve spojení s viděním biologickým, odpovídá *SfM* jevu, kterým jsou obdařeni lidé a mnohé další živé bytosti - a to takovým, že jsme schopni obnovit 3D strukturu z projekce 2D pohybu na sítnici, který je dán pohybem objektu nebo celé scény [8].

Pro člověka je přirozené shromáždit velké množství informací o trojrozměrné struktuře okolního světa a to pouhým pohybem v daném prostoru. Když se pohybujeme, předměty kolem nás se také hýbou, tato informace je získána z obrázků vnímaných v průběhu času [16]. Hledání struktury z pohybu představuje shodný problém jako hledání struktury ze stereo-vize. V obou případech je potřeba nalézt korespondence mezi rekonstruovaným 3D objektem a vstupními obrazy. Pro zjištění daných korespondencí se hojně využívá sledování význačných bodů v obraze a s pomocí jejich trajektorií se rekonstruuje jejich 3D pozice a také pohyb kamery.

Průběh SfM lze rozdělit do kroků zobrazených na obrázku 3.4.



Obrázek 3.4: Structure from motion pipeline znázorňující jednotlivé kroky.[1]

Kapitola 4

Návrh systému

Před samotným návrhem programové části práce je důležité důkladně stanovit cíl práce od kterého se bude odvozovat požadovaná funkcionalita a podmínky na provoz. Toto následně umožní práci rozdělit do dílčích částí které mohou být detailně popsány a přehledně rozděleny do více kapitol.

4.1 Specifikace cíle

Cílem této práce je lokalizace kamery v rámci modelu prostředí která bude dále využitelná například pro realizaci rozšířené reality. Možné využití bylo nastíněno v kapitole 1 – Úvod. Ovšem než bude možné samotnou kameru lokalizovat, je nutné vytvořit model daného prostředí. Samotná tvorba modelu pokládá několik zásadních otázek – jak má model vypadat, jaké budou požadované vlastnosti, či jakým způsobem bude reprezentován z pohledu datových struktur. Ovšem tyto otázky nelze obecně zodpovědět bez znalosti lokalizačního systému a jeho požadavků na data obsažená v modelu.

Nejprve bude vhodné navrhnout systém pro lokalizaci kamery v rámci předem známého prostoru a na základě poznatků o tomto systému poté navrhnout strukturu pro uchování daného rekonstruovaného modelu prostředí.

Na základě informací o lokalizované kameře je dále nutné tyto informace nějakým způsobem využít pro realizaci rozšířené reality. Pro rozšířenou realitu je klíčové, aby systém pracoval v reálném čase, proto bude vhodné využít nějakého systému pro sledování pohybu kamery, jelikož tyto přístupy pracují velmi dobře v reálném čase, díky vyhledávání blízkých korespondencí mezi po-sobě jdoucími snímky, jak bylo popsáno v 2.4.

Jak lze vidět, celý systém lze rozložit do několika dílčích částí:

1. Rekonstruovaný 3D model prostředí.
2. Lokalizátor kamery.
3. Tracker pohybu kamery.
4. Realizátor rozšířené reality.

4.2 Podobné systémy

Při návrhu je vhodné vzít v potaz již navržené fungující systémy, které se zabývají podobnými problémy. Mezi tyto systémy lze zařadit především systémy, které se zabývají

mapováním prostředí na robotických platformách a zejména souběžnou lokalizací v rámci mapovaného prostoru.

Prvním z takovým přístupem je metoda **SLAM**¹ – *simultánní lokalizace a mapování* (volně přeloženo z *Simultaneous localization and mapping*). Jak již název napovídá, jedná se o techniku, kdy dochází snímání prostoru okolo robota, pomocí různých senzorů a následně se na základě těchto informací sestavuje mapa (model) daného prostředí. Výsledky ze senzorů jsou následně algoritmicky zpracovány pro potřeby lokalizace. Této techniky lze využít i pro potřeby zpřesňování již sestavené mapy. Tento přístup pro účely mé práce není příliš vhodný, zejména proto, že pro sestavování mapy se využívá různých senzorů a ne pouze kamery – existuje modifikace MonoSLAM², která počítá pouze s takovýmto vstupem, ale stále je zde primární využití pro procházení a autonomní navigaci v rámci neznámých scén.

Dalším přístupem je metoda **PTAM**³. – *Paralelní trackování a mapování* (volně přeloženo z *Parallel tracking and mapping*). Zde je rozdíl oproti metodě SLAM v tom, že se předpokládá využití právě jedné kamery jako vstupu ze které se využívají veškeré informace. Jak je popsáno v [10], tak vhodné využití těchto principů je pro rozšířenou realitu v rámci menších neznámých prostor. Dalším hlavním rozdílem je oddělení mapování a lokalizace do dvou separátních částí. Ideální je využívat více-jádrové procesory, kde každé jádro může obsluhovat právě jednu činnost. S touto metodou lze zobrazovat rozšířenou realitu v reálném čase. Tento přístup může posloužit jako dobrý zdroj inspirace pro mou práci. Ač se zdá, že se jedná o stejné zaměření, není tomu tak, hlavní a nejpodstatnější rozdíl je v tom, že tato práce si dává za cíl lokalizaci kamery v rámci známého modelu prostředí, kdežto PTAM počítá právě s tím, že prostředí je předem neznámé a nezmapované.

Na metodě PTAM jde vidět výhodnost rozložení dílčích částí problému do paralelního zpracování ve více vláknech. Sice s tímto roste i náročnost synchronizace dat a řešení jiných problémů vycházejících z vícevláknového programování, ovšem tato cena je vykoupena vyšší výpočetní výkonností.

4.3 Návrh lokalizátoru

S pomocí technik popsaných v 3.4 je dále třeba navrhnout systém pro lokalizaci kamery. Tento systém bude obstarávat robustní lokalizaci na základě reference vstupního snímku s modelem prostředí. Protože model prostředí ještě není navržen, je vhodné nejprve zanalyzovat požadavky na tento model z pohledu systému pro lokalizaci.

Lokalizační část se bude skládat z několika dílčích kroků:

1. Popis obrazu klíčovými body.
2. Nalezení 2D-3D referencí mezi vstupním snímkem a modelem.
3. Výpočet matice kamery pro daný vstupní snímek.

Prvním krokem je popis obrazu klíčovými body. Techniky pro popis jsou popsány v kapitole 2.1. Tohoto poznatku bude vhodné dále využít při návrhu struktury modelu prostředí. Dále je potřeba nalézt 2D-3D reference. Tímto krokem se rozumí nalezení vhodných

¹ <http://www.openslam.org/>

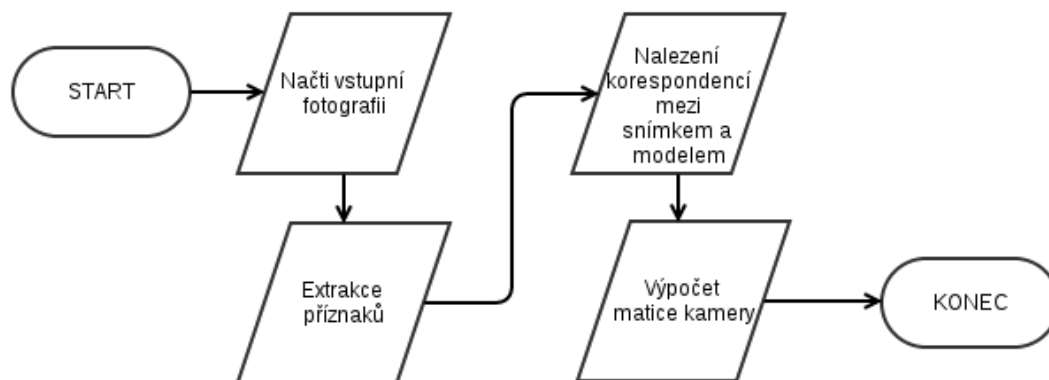
² http://www.robots.ox.ac.uk/~lav/Papers/davison_etal_pami2007/davison_etal_pami2007.pdf

³ <http://www.robots.ox.ac.uk/~gk/publications/KleinMurray2007ISMAR.pdf>

korespondenci mezi klíčovými body ve vstupním obraze a klíčovými body, které popisují jednotlivé body ve 3D modelu.

Nejjednodušší možností jak tyto reference získat, je využití vstupních fotografií, ze kterých byl tvořen model a nalézt *nejbližší* snímek který odpovídá tomuto pohledu a na základě podobnosti těchto pohledů nalézt reference z 2D do 3D. Tento přístup není ale příliš vhodný, jelikož by bylo potřeba porovnat velké množství obrazů navzájem. Optimálnějším přístupem by mohlo být vyhledávání korespondencí pomocí vyhledávacího stromu. Zjednodušeně by šlo o prohledávání podprostorů a hledání nejbližšího souseda, tak jak bylo nastíněno v kapitole 2.4.

Dále je potřeba na základě těchto referencí vypočíst matici kamery, který je blíže popsán v kapitole 3.4. Vzhledem k tomu, že výpočet kamery probíhá na základě mnoha vzorků, je možné, že některé vzorky do tohoto modelu nepatří. Tyto nenáležící body by bylo vhodné eliminovat před samotným výpočtem nebo lze využít metody RANSAC, která je popsána v kapitole 3.2. Celá struktura lokalizace kamery je názorně vidět na obrázku 4.1.



Obrázek 4.1: Diagram znázorňuje lokalizace kamery v rámci předem rekonstruovaného modelu prostředí.

4.4 Návrh modelu

Jak bylo popsáno v předchozí kapitole, tak pro potřeby lokalizace je vhodné využít klíčových bodů, respektive jejich deskriptorů, pro nalezení 2D-3D referencí mezi vstupním obrazem a modelem rekonstruovaného prostředí. Tohoto předpokladu bylo využito při návrhu modelu. Aby model zastával svou funkci, je potřeba aby každý bod měl svou jednoznačnou pozici v prostoru $X \times Y \times Z$.

Další důležitou součástí každého bodu je jeho popis získaný z jednotlivých fotografií při rekonstrukci. Tento popis je tvořen vektorem vektorů příznaků – zjednodušeně se jedná o jednotlivé deskriptory nalezených korespondujících si bodů při rekonstrukci.

V případě že model není potřeba využívat pro vizualizace rekonstruovaných scén

Z těchto poznatků vychází, že ideálním typem 3D modelu pro využití pro lokalizaci bude upravené mračno bodů. Více o tomto datovém typu modelu lze nalézt v kapitole 3.5.

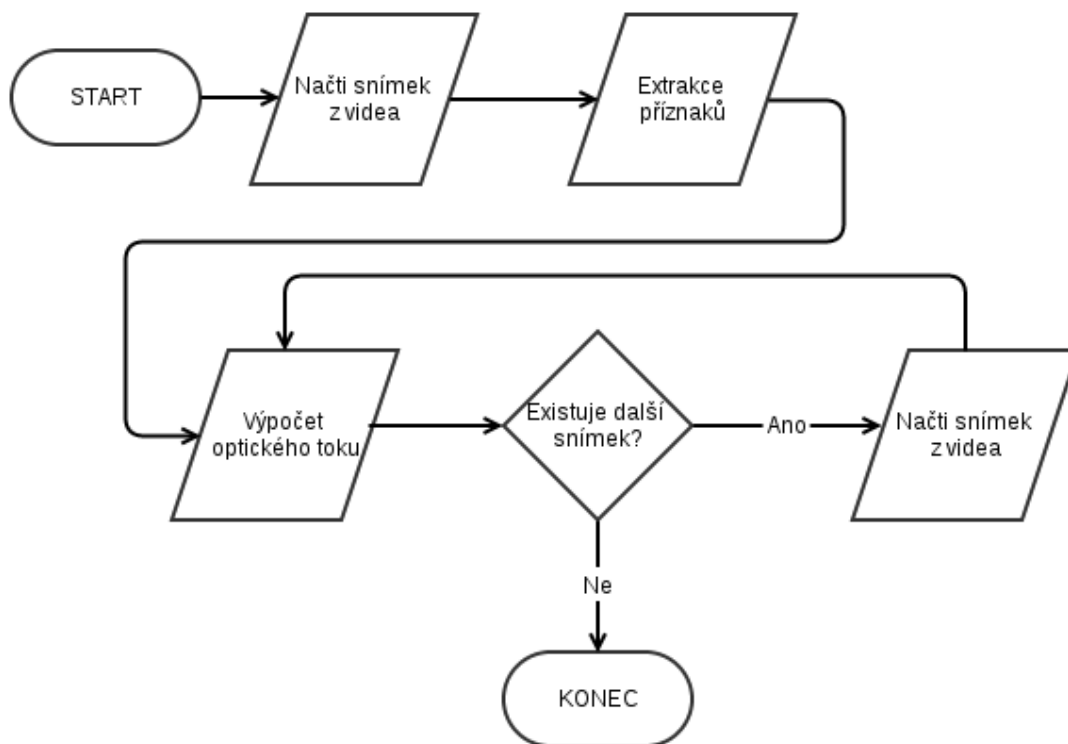
4.5 Návrh systému pro trackování pohybu kamery

V této kapitole budou diskutovány možnosti urychlení s pomocí systému pro sledování pohybu kamery. Jak již bylo zmíněno v kapitole 4.2, bylo by vhodné využít oddělení jednotlivých částí do pod-celků, které budou moci fungovat paralelně. Část lokalizace, tak jak byla navržena, stále není vhodná pro vyhledávání pozice kamery v reálném čase. Systémy, které spolehlivě pracují v reálném čase jsou systémy popsané v kapitole 2.4.

Tyto systémy jsou postaveny na tom, že primárně sledují změny mezi jednotlivými snímky a tudíž lze rychle nalézt korespondence mezi nimi. Tyto korespondence se vyhledávají v rámci jakýchsi okének, která určují maximální vzdálenost od prvního bodu v jednom obraze kde se hledá shodný bod v obraze druhém.

Pro rychlou detekci význačných bodů se využívá zejména hranových a rohových detektorů.

Na schématu 4.2 lze návrh sledovacího systému.



Obrázek 4.2: Diagram znázorňuje tracking klíčových bodů.

4.6 Propojení lokalizace a sledování pohybu kamery

Dalším podstatným krokem je spojení předem navržených částí v jeden celek. Dle inspirace metodou PTAM, byl systém navržen do více ucelených částí, které se starají o jim význačnou činnost. Ovšem samotné propojení těchto částí by nebylo dostatečné, jelikož je potřeba přepočítávat matici kamery pro každý vstupní snímek. Aktuálně navržený tracker pohybu kamery dokáže sledovat jednotlivé vstupní body, avšak nemá reference do 3D

modelu k těmto bodům.

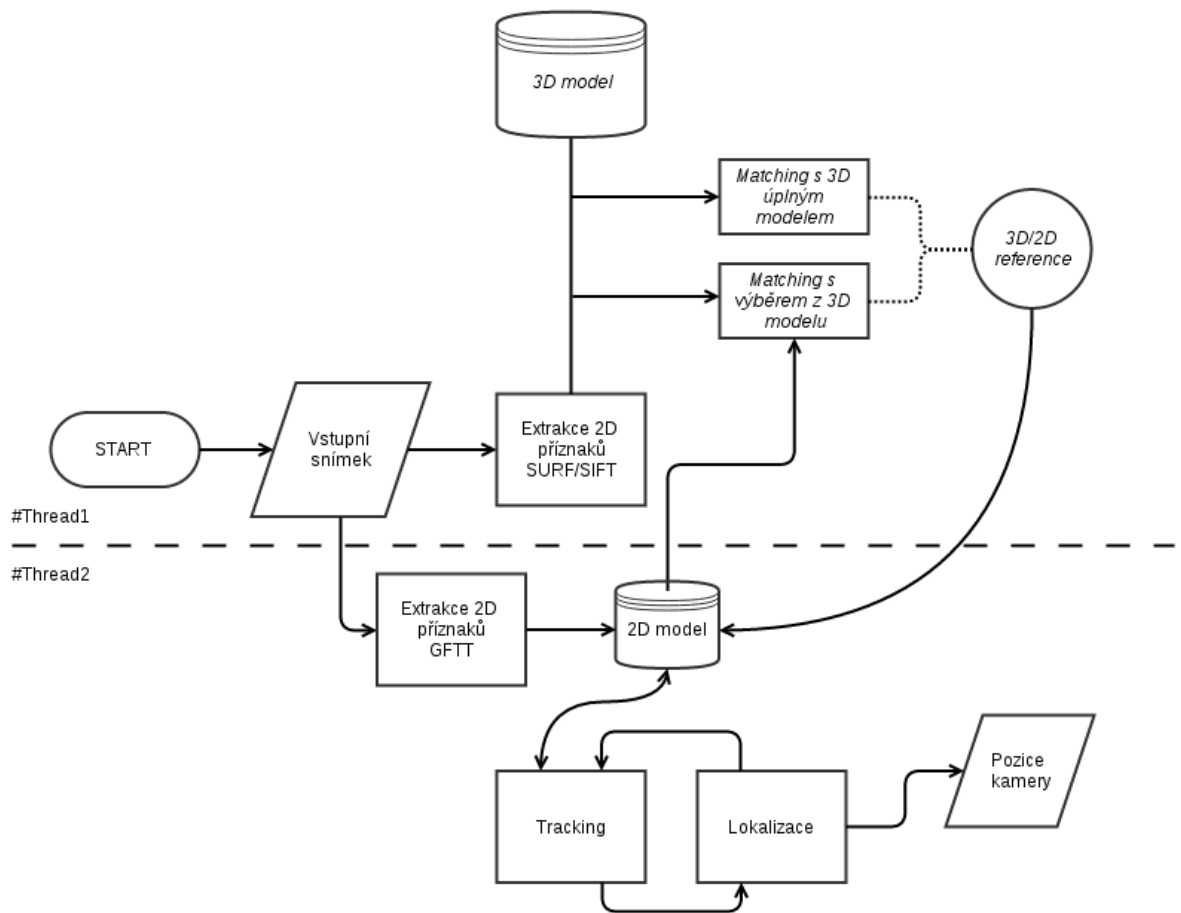
Pro tyto účely je potřeba navrhnout metodu, která toto propojení zajistí nebo způsob jakým tento problém vyřešit.

S přihlédnutím k navrženému modelu a lokalizátoru je potřeba vycházet z informací, které jsou k dispozici. Lokalizátor pracuje se sadou klíčových bodů, ke kterým našel reference do 3D modelu prostředí. Ideálním případem by bylo sledovat tyto body a na základě změny jejich pozic poté přepočítávat matici kamery.

Bohužel klíčové body SIFT nebo SURF nejsou na sledování mezi jednotlivými snímky příliš vhodné kvůli své výpočetní náročnosti. Výhodné bude nalézt nejbližší korespondující klíčové body k bodům z lokalizace, které následně lze sledovat v průběhu dalších snímků.

Nabízí se řešení, kdy pro každý klíčový bod z lokalizátoru je potřeba nalézt vhodný klíčový bod pro sledování optického toku. Tímto lze získat potřebné reference mezi rekonstruovaným modelem a vstupním obraze v reálném čase.

Navrhované spojení lokalizátoru a trackovacího systému je vyobrazeno na 4.3, kde lze vidět i navrhované rozdělení na výpočet do více vláken.



Obrázek 4.3: Schéma návrhu celého systému.

První vlákno by poskytovalo robustnější lokalizaci na základě vyhledání korespondencí přímo s modelem. Také by zajišťovalo doplňování nových 2D-3D referencí ke kterým by se posléze našly referenční klíčové body pro sledování.

První vlákno tedy zastává důležitou roli kvůli inicializaci lokalizace a následnému doplňování bodů pro sledování. Každý lokalizační cyklus by bylo vhodné využít pro aktuální zpřesnění matice kamery.

4.7 Budoucí možná rozšíření

Jedním z možných rozšíření, které by zajistili optimalizaci průběhu lokalizace kamery. Toho by šlo docílit pokud bychom aktuální lokalizovanou kameru využili k omezení bodů, které poskytuje 3D model pro vyhledávání. Vytvořilo by se tím pomyslné prostorové okno, ve kterém by bylo potřeba vyhledávat korespondenční body a zejména u větších modelů by mohlo dojít k výraznému urychlení lokalizace.

Kapitola 5

Model

Tato kapitola se zabývá modelem prostředí, což je podstatná část navrženého systému, jelikož se jedná o klíčový prvek pro lokalizaci v daném prostředí. Budou zde popsány principy pro tvorbu datových sad vhodných pro tvorbu modelů, dále nástroje které lze využít pro realizaci této tvorby a v neposlední řadě také návrh experimentů pro vyhodnocení kvality rekonstrukce a zhodnocení výsledků. Návrh ideální datové struktury je možné nalézt v kapitole 4.4.

5.1 Tvorba datové sady

V této části popíšu nezbytné kroky pro tvorbu datové sady modelu. Jedná se o praktické zkušenosti nabrané při tvorbě datových sad v průběhu tvorby této práce. Za celou dobu bylo pořízeno na 30 datových sad tvořených od 10 po 90 snímků.

Počet snímků v datové sadě odpovídá velikosti daného prostředí, které je potřeba modelovat. Pro malý prostor – například pracovní stůl – vystačí menší počet fotografií. Je vhodné využít vrstvení fotografií – kdy pro každou fotografovanou pozici utvořím více snímku, čímž vznikne překrytí fotografií a zvýší to přesnost modelu. V závislosti na využitém softwaru a potřebném počtu bodů je potřeba zvolit vhodný počet těchto vrstev. Tento počet bude diskutován v části s experimenty.

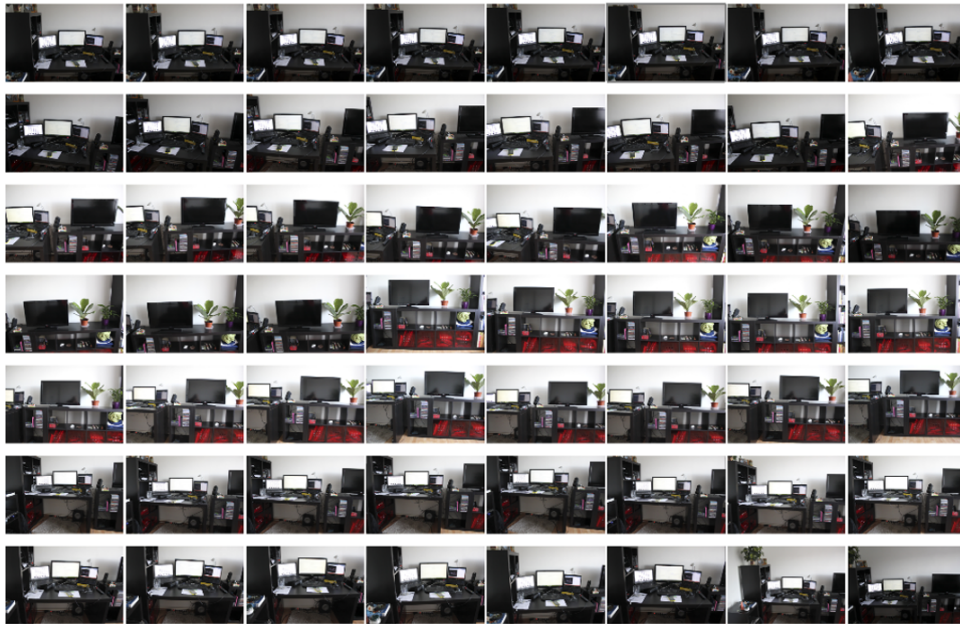
Ukázková sada fotografií pro tvorbu testovacího modelu, je vidět na obrázku 5.1.

Pokrytí daného prostoru je výhodné mít rovnoměrné, čímž se opět dá dosáhnout lepší kvality výsledku. Dále lze brát v potaz vzdálenost jednotlivých fotografií od fotografovaného objektu – čímž lze dosáhnout symbolické změny rozlišení px/m . Navíc kombinace detailnějších fotografií s celkovými pohledy celkově mohou zvětšit detailnost daného modelu.

5.2 Tvorba modelu

Jakmile jsou vytvořeny datové sady, následuje vygenerování klíčových bodů včetně deskriptorů pro každou vstupní fotografii. Pomocí těchto klíčových bodů lze následně provést vyhledání korespondencí mezi jednotlivými obrázky. Z těchto korespondencí je možné odhadnout 3D strukturu korespondencí mezi jednotlivými body vzhledem k umístění kamer. Tato základní struktura se tvoří například pomocí triangulace. Jakmile je sestaven základní model, následuje jeho optimalizace s využitím metody *bundle adjustment*.

Výsledkem těchto kroků je množina bodů, resp. mračno bodů v prostoru, které odpovídají detekovaným korespondujícím si význačným bodům a také pozice kamer, jak lze vidět



Obrázek 5.1: Výběr ze sady fotek využitých pro tvorbu jednoho z modelů.

na obrázku 5.2.

V tuto chvíli je dostupný tzv. řídký model, není úplně vhodný pro vizualizaci, ovšem skládá se zejména z klíčových bodů a jejich souřadnic v prostoru, což je vhodné. Dále je dobré mít stále i informaci o tom, jaký klíčový bod je v jakém zdrojovém obrázku, v případě potřeby, pokud bychom zjistili, že pohled do modelu se kryje zejména s body z určité fotografie, tak lze využít tuto fotografii na přesnější dohledání jednotlivých korespondencí mezi klíčovými body – svým způsobem je možnost docílit určité úrovně detailu, z nižší úrovně v modelu k vyšší úrovni ve zdrojových fotografiích.

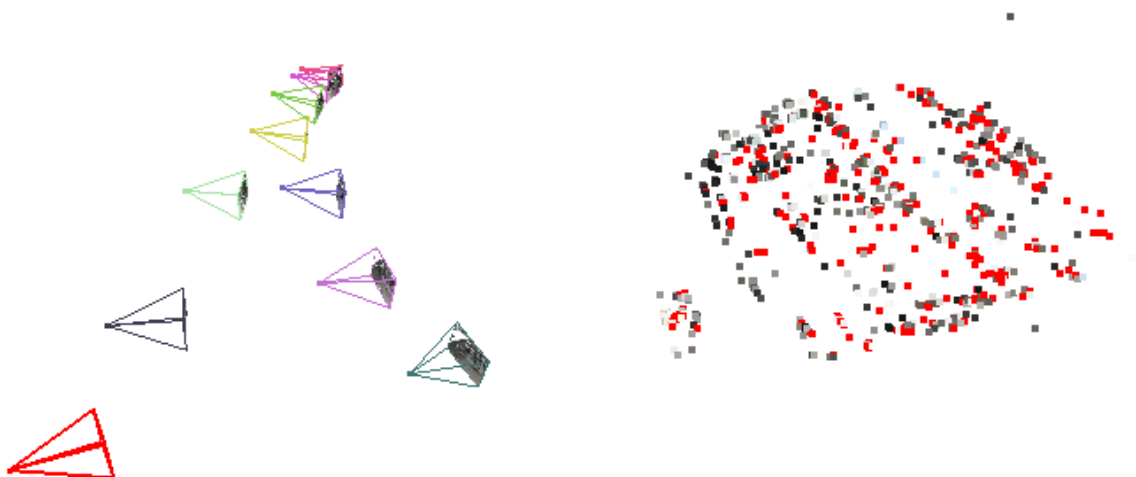
Dále je možné provést tzv. *dense reconstruction* (*hustá rekonstrukce*) – čímž jsme schopni vytvořit hustý model daného prostředí či objektu, jak lze vidět na obrázku 5.3.

5.3 Nástroje

V této sekci budou představeny nástroje a techniky, které byly využívány při tvorbě této práce. Jedná se zejména o aplikace, které jsou určeny pro tvorbu modelů s pomocí techniky SfM.

VisualSFM

VisualSFM je desktopová aplikace určená pro 3D rekonstrukci s využitím *structure from motion* (SfM). Rekonstrukční systém v této aplikaci zahrnuje více různých projektů, které jsou pro SfM potřebné. Mezi tyto projekty patří například *SiftGPU* – knihovnu pro hledání klíčových bodů SIFT s urychlováním výpočtu na grafické kartě nebo *Multicore Bundle Adjustment*, který dokáže využívat více-jádrových procesorů pro své výpočty nebo *Incremental Structure from motion* [21].



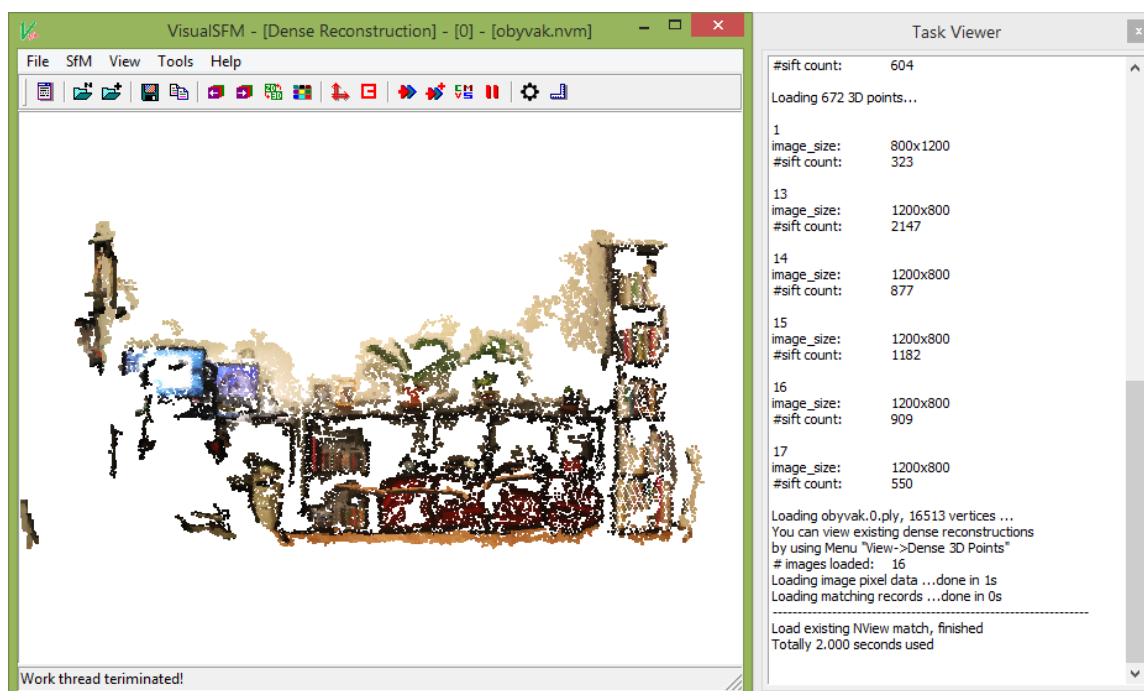
Obrázek 5.2: Hranoly reprezentují pozice kamer a směr pohledu, body reprezentují jednotlivé význačné body. Červeně označená kamera koresponduje s červeně vyznačenými body.



Obrázek 5.3: Hustý model, vytvořený rekonstrukcí modelu řídkého.

Tento nástroj poskytuje přehledné GUI na tvorbu modelu ze vstupních fotografií, avšak formát dat není vhodný pro další zpracování. Také nelze zvolit jiný detektor klíčových bodů, což znemožňuje testování jednotlivých detektorů a jejich vlivu na kvalitu při tvorbě modelu.

Tento software je vhodný pro tvorbu 3D modelů ze vstupních fotografií, poskytuje kva-



Obrázek 5.4: Náhled uživatelského rozhraní aplikace VisualSFM

litní výstup a využívá knihoven, které mají možnost hardwarové akcelerace.

Exploring Structure from Motion using OpenCV

V rámci publikace [4] lze nalézt příklad, který popisuje tvorbu modelu tvořeného mračenem bodů pomocí metody SfM¹.

Jelikož pro svou práci potřebuji nástroj pro tvorbu modelů, který bude poskytovat rozhraní pro nastavení parametrů různých částí systému, je přiložená aplikace vhodná z několika důvodů. Mohu dopsat rozhraní pro export potřebných dat pro následné využití v další části práce. Dále mohu experimentovat s nastavením jednotlivých částí tvorby modelu a vyhodnocovat vliv těchto parametrů na výsledný model.

5.4 Experimenty

V této části bude popis testů a experimentů provedených při tvorbě modelu. Testy byly spouštěny na notebooku s procesorem *Intel Core i5-4200M*, který má dvě jádra a čtyři vlákna s maximální frekvencí 2.5GHz a 4GB operační paměti. Nejprve provedu analýzu přesnosti tvorby modelu, také rychlost, počet nalezených bodů a průměrnou projektivní chybu. Při vývoji pro potřeby testování vzniklo mnoho datových sad, pro adekvátní reprezentaci různých scén byly provedeny testy na šesti sadách. Rozdělení jednotlivých sad bylo bráno dle těchto kritérií:

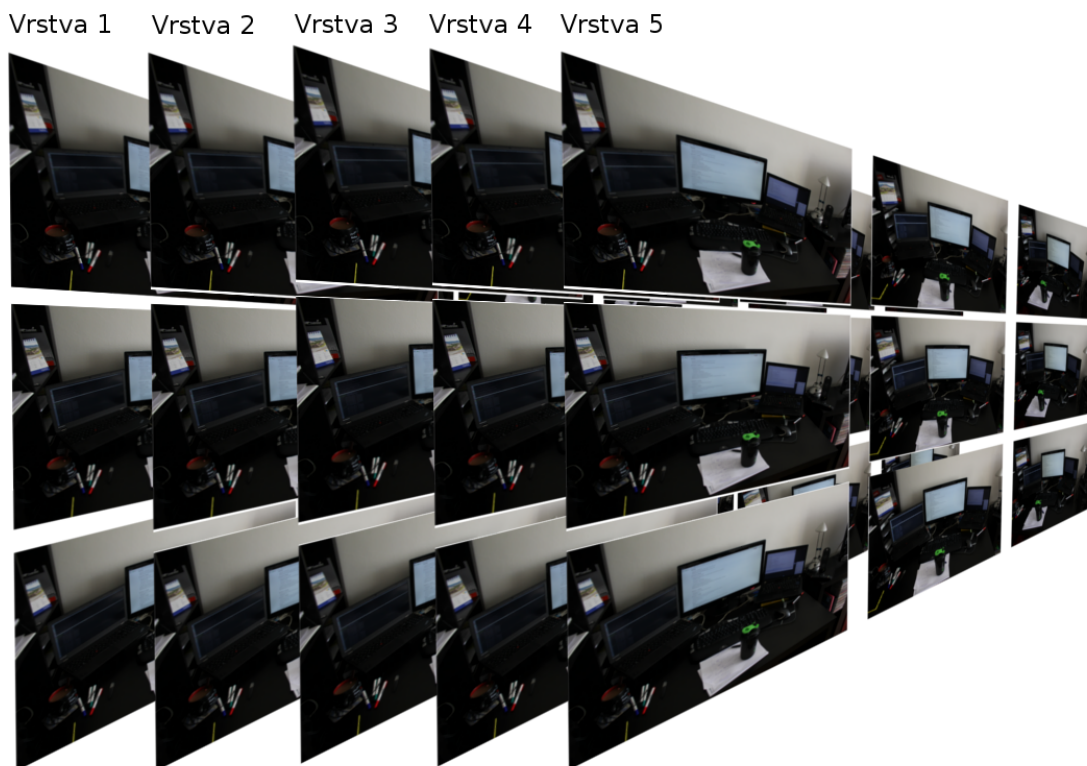
1. **Planární** – fotografie rozmístěné na rovné zdi.
2. **Prostorový** – běžný prostor otevřenějšího charakteru.

¹Structure from Motion – struktura z pohybu

3. **Vizuálně zaplněný.**
4. **Vizuálně prázdný.**
5. **Reálně zaplněný** prostor, nemusí ale vizuálně působit jako plný (čisté plochy atp.).
6. **Reálně prázdný** prostor.

5.5 Validace tvorby modelů

Pro analýzu přesnosti tvorby modelu bylo potřeba vytvořit speciální sadu fotografií, na které lze provádět testy přesnosti tvorby. Tato sada je významná tím, že daná scéna je pokryta více vrstvami fotografií tak, že pozice kamer jsou v každém kroku zhruba stejné. V tomto případě je tedy pokrytí modelu velmi dobré a proto lze očekávat malou chybu a velký počet stabilních bodů.



Obrázek 5.5: Grafické znázornění rozložení fotografií při tvorbě testovacího modelu.

Jakmile jsou provedeny testy formou tvorby modelu ze vstupních dat, kdy každá vrstva má 9 fotografií a takových vrstev je dohromady 10, takovým způsobem, že model je nejprve utvořen ze všech vstupních fotografií a poté budou po vrstvách fotografie odstraňovány, pak by při dostatečném pokrytí rekonstruované scény nemělo docházet k velkým rozdílům v chybě ani počtu stabilních bodů.

Na obrázku 5.5 lze vidět grafické znázornění takové tvorby modelu.

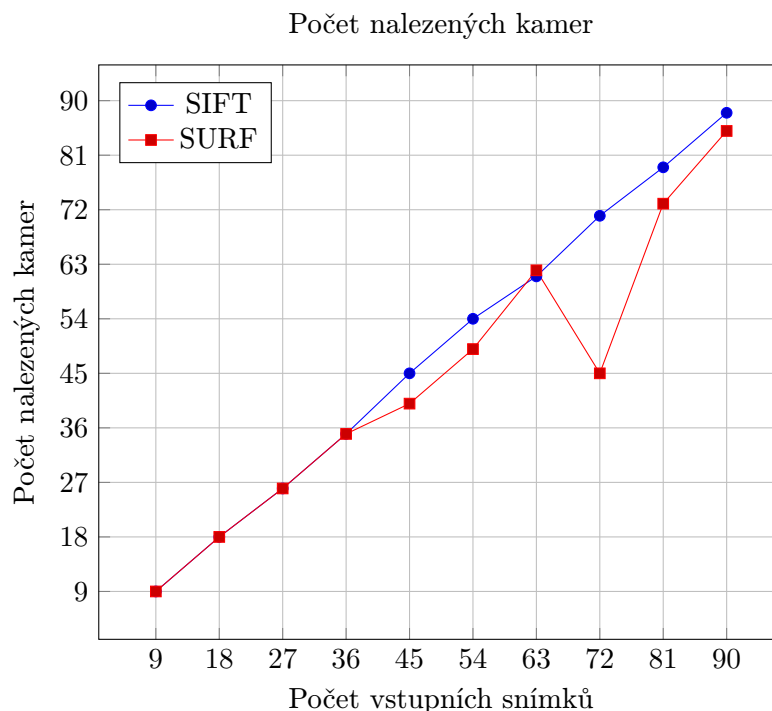
Testovací datová sada čítá 6 modelů, tak jak byly popsány v kapitole 4.4. Každý model obsahuje 10 vrstev po 9 fotografiích, celkem tedy 90 fotografií na model.

Pro přehlednost nebudou uvedeny grafy pro všechny modely, ale pouze ty, které byl buďto něčím významné nebo zajímavé.

Kvalita vstupních fotografií

Prvním zkoumaným parametrem byla kvalita vyfocených fotografií z pohledu jejich vhodnosti pro tvorbu modelu. V případě že pohled některé z fotografií byl mimo snímané prostředí (odchylka byla příliš velká), je možné že nebylo nalezeno dostatečné množství párů pro rekonstrukci dané kamery. V případě že dochází k neúspěšnému nalezení více kamer po sobě, software pro rekonstrukci modelu ukončí svou činnost.

Pro ilustraci tohoto stavu lze na grafu níže vidět, že takový případ nastal při použití SURF detektoru, kdy pro 72 vstupních fotografií bylo nalezeno pouze 45 kamer.



Následující tabulka shrnuje úspěšnost rekonstrukce ve všech modelech s jednotlivými detektory/deskriptory a shrnuje procentuální úspěšnost tvorby modelu z pohledu lokalizace všech vstupních kamer.

Detektor/deskriptor	Očekávaný počet kamer	Počet nalezených kamer	%
SIFT	2970	2916	98
SURF	2970	2328	78

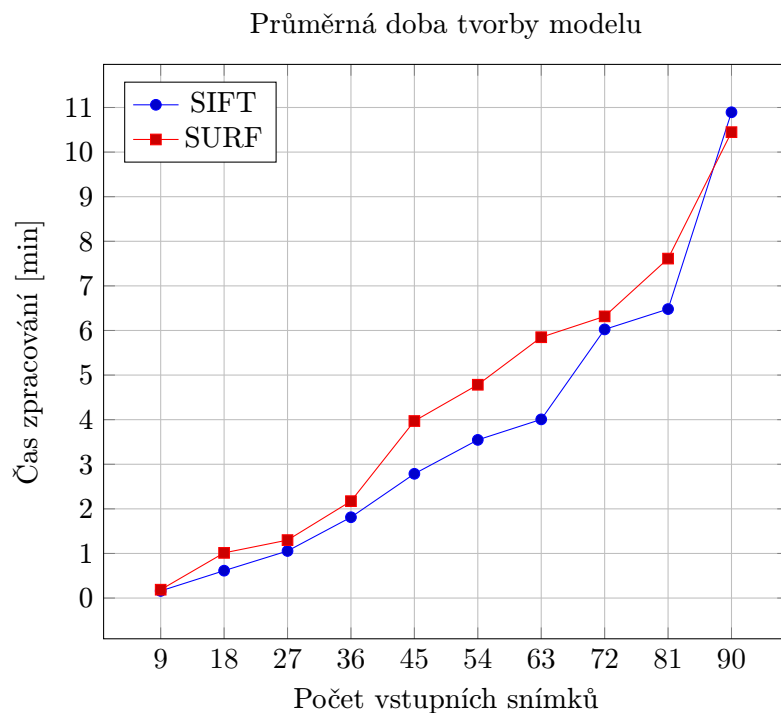
Na základě prezentovaných dat, lze říct, že rekonstrukce s detektorem SURF je náchylná na chybné snímky. Celkový rozdíl v přesnosti 20% je velmi mnoho. Chyba s detektorem SIFT se projevila, ale průměrná úspěšnost přesáhla 98%, což je velmi dobrý a očekávaný výsledek.

Využití metody SURF lze doporučit pro scény s vysokým výskytem vizuálních informací, tam podává srovnatelné výsledky s metodou SIFT.

Rychlost zpracování

Dalším testovacím faktorem byla doba zpracování jednotlivých modelů v závislosti na použité metodě. Předpokladem je výkonnostní výhoda metody SURF nad metodou SIFT.

Následující graf ukazuje průměrný časový průběh při tvorbě modelů v závislosti na počtu vstupních fotografií.



Výsledky délky zpracování jsou překvapující, jak lze vidět na grafu výše. Doba zpracování se SIFT detektorem byla v průměru kratší než se SURF detektorem. Předpokladem byla vyšší výkonnost detektoru SURF, jelikož, jak bylo zmíněno v kapitole 2.3, jedná se o metodu vycházející z principu SIFT, avšak má navrženy optimalizace pro vyšší výkonnost.

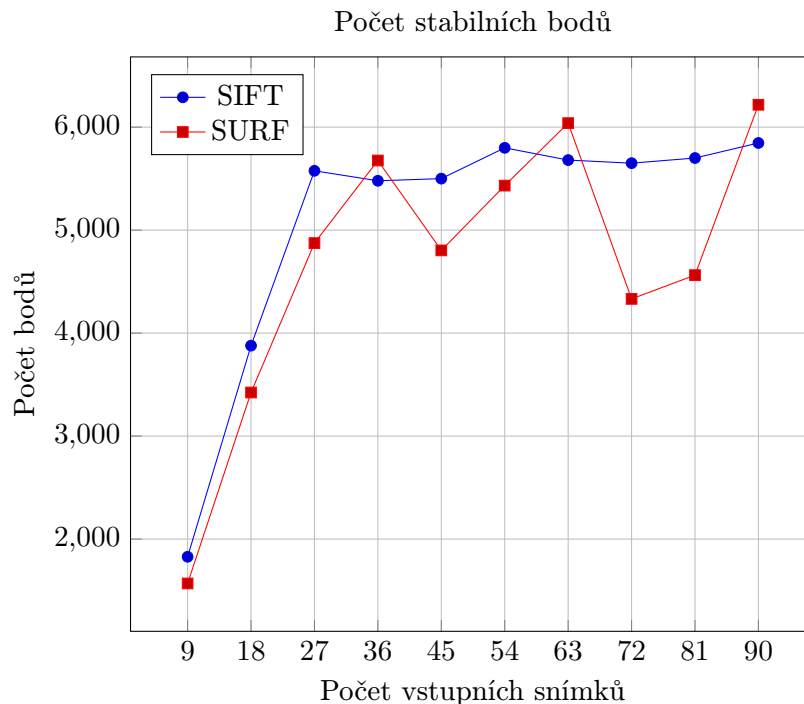
Tento průběh je zřejmě způsoben základním nastavením detektorů. Další možností je problém v rámci implementace jednotlivých modulů v knihovně OpenCV.

Námětem pro budoucí rozšíření může být právě otestování různých verzí knihovny OpenCV.

Počet stabilních bodů v modelu

Následující test popisuje rozložení počtu velmi stabilních bodů v rámci jednotlivých modelů. Za stabilní body z hlediska testování jsou považovány body s nízkou projekční chybou, které jsou viditelné z většího množství kamer.

Předpokladem je postupný nárůst počtu bodů s přibývajícím počtem překrývajících se vrstev. Od určitého počtu vrstev by se počet stabilních bodů měl ustálit okolo určité hodnoty.



Z grafu, který zobrazuje průměrný počet nalezených bodů ve všech modelech v závislosti na počtu vstupních souborů lze nalézt předpokládaný přechod – stabilizaci – daného modelu.

Detektor SIFT tento předpoklad přibližně splnil. Dostatečné stability lze dosáhnout v průměru od tří vrstev, počet stabilních bodů se pohyboval v rozpětí 5400 – 5800 bodů.

Detektor SURF vykazoval jistou míru nespolehlivosti už při detekci mezi snímky, také rozdíly v počtu stabilních bodů jsou značné.

Jednotlivé modely měly samozřejmě celkový počet bodů vyšší než je vyobrazen zde, jak ale bylo uvedeno v úvodu tohoto testu, jedná se o výběr z mračna bodů.

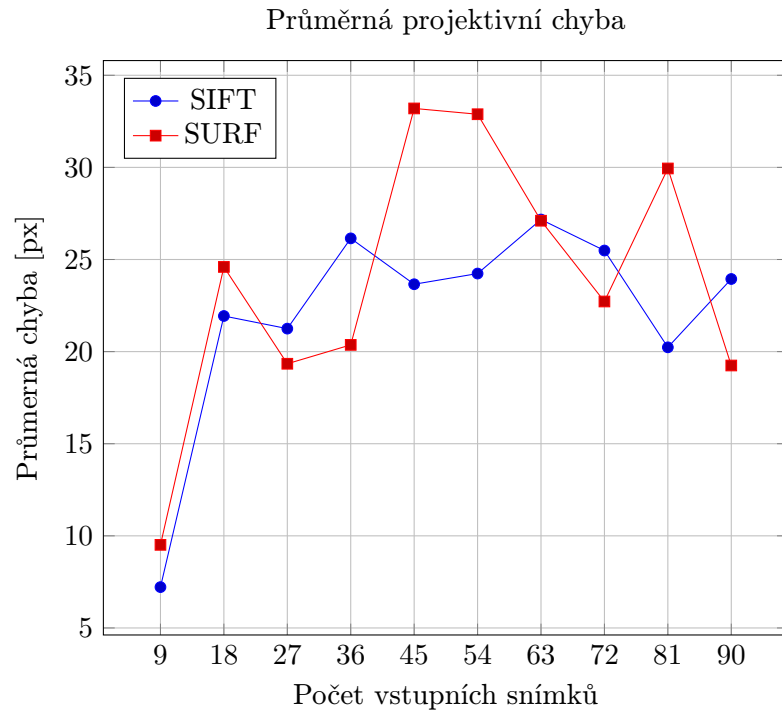
Průměrná chyba modelu

Dále byla zjišťována průměrná projekční chyba vytvořeného modelu. Tato chyba vykazuje míru přesnosti daného modelu.

Výpočet této chyby probíhá zprůměrováním projekční chyby všech nalezených bodů.

Tato chyba má za následek nepřesnost následné lokalizace v rámci modelu.

Předpokladem je s vyšším počtem vrstev klesající chyba díky většímu překrytí jednotlivých částí snímaného modelu a od určitého počtu ustálení chyby.



Dle grafu lze vidět, že předpoklad s klesající chybou byl mylný. Velmi nízká chyba při tvorbě modelu z malého množství fotografií vyšla nad očekávání dobře. Toto lze přikládat metodě tvorby modelu, kde se používá metod optimalizace a chyba se minimalizuje. Jelikož tyto modely měly poměrně málo stabilních bodů, tak jejich optimalizace proběhla úspěšně.

Tuto nestabilitu je možno přičíst k zvolenému systému pro rekonstrukci scény. Nejedná se o profesionální aplikaci, ale spíše o jednodušší verzi vhodnou pro výukové účely.

Zde se nabízí možnost pro **budoucí rozšíření** této práce o metody zpřesnění tvorby modelu. Jednou možností může být využití jiných externích nástrojů, jsou lépe odladěny. Jako další možnost se nabízí využití senzorů v kombinaci s vizuálním snímáním pomocí kamery, takže by mohl být model vytvořen s velmi vysokou přesností. Toto ovšem vyžaduje speciální hardware s jehož pomocí lze snímat i hloubková data, případně měřit reálnou vzdálenost od daného zařízení.

Kapitola 6

Sledování pohybu kamery

Tato kapitola pojednává o trackovacím systému. Tento systém byl navržen s ohledem na sledování pohybu kamery v reálném čase.

6.1 Realizace trackeru

Při implementaci trackovacího systému jsem opět využil modulárního návrhu, některé moduly byly využity jak pro tvorbu lokalizační části, tak pro trackovací systém. Při návrhu jsem vycházel z životního cyklu při sledování klíčových bodů. Tato činnost lze popsat následujícími kroky:

1. **Inicializace.**
2. **Načtení snímku z kamery/videoa ke zpracování**, vyhledání GoodFeaturesToTrack.
3. **Výpočet optického toku** mezi předchozím snímkem a aktuálním, pokud předchozí snímek neexistuje, jdi na následující krok.
4. **Vyměň aktuální klíčové body** a snímek z kamery s předchozími.
5. **Pokud je klíčových bodů dostatek**, pak jdi na 3, pokud ne jdi na 2.
6. **Ukončení činnosti**, uvolnění zdrojů.

Načtení snímku a hledání klíčových bodů

V této části se využívá shodný modul, který byl popsán v 7.2. Jak již bylo zmíněno, při inicializaci modulu lze jednoduše zvolit typ *detektoru* i *deskriptoru* klíčových bodů. V tomto případě se využívá detekce GFTT¹. Pro detekci *GFTT* se využívá *Harrisův* rohový detektor nebo se vypočítává vlastní číslo gradientní matice pro detekci rohů, s tou výjimkou, že vyhledané rohy musí splňovat určité nároky na kvalitu.

Pro potřeby trackeru je vhodné také tyto nalezené rohy zpřesnit na mezi-pixelovou pozici. K tomu se využívá funkce z OpenCV `cornerSubPix()`².

¹Good Features To Track

²http://docs.opencv.org/modules/imgproc/doc/feature_detection.html#cornersubpix

Výpočet optického toku

Pro výpočet optického toku se využívá algoritmus *Lucas-Kanade* implementován v OpenCV. Jedná se o funkci `calcOpticalFlowPyrLK()`. V tomto případě se jedná o upravenou implementaci, která je iterativní a pracuje s obrazovými pyramidami.

Výpočet optického toku nám zajišťuje vyhledání nových pozic nalezených rohů (klíčových bodů) v aktuálním snímku obrazu díky znalosti předchozího snímku. Více informací o sledování klíčových bodů lze nalézt v [2.4](#).



Obrázek 6.1: Ztráta klíčových otáčením kamery.

Dostatek klíčových bodů

Pro dostatečně přesné sledování je potřeba stanovit hranici počtu bodů, abychom mohli považovat sledování za dostatečně přesné. Na druhou stranu vysoké množství bodů bude zbytečně výkonnostně náročné. Experimentální cestou jsem stanovil spodní hranici počtu sledovaných bodů na 50 a maximální počet vyhledávaných na 200.

V průběhu sledování dochází ke ztrátám korespondencí, zejména kvůli tomu, že se kamera otočí a klíčové body *vypadnou* ze sledovaného obrazu, viz. obrázek [6.1](#).

6.2 Reference mezi klíčovými body

Reference klíčových bodů jsou důležitou součástí systému, ty budou zajišťovat správné provázání mezi jednotlivými klíčovými body. Zásadním problémem zde bylo správné nastavení párování jednotlivých bodů. Proto bylo potřeba analyzovat do jaké vzdálenosti je možné nalézt dostatečný počet korespondujících si klíčových bodů z různých detektorů. Je-li pro lokalizaci využívám zejména detektor SIFT a SURF, bral jsem tyto jako referenční. K nim je potřeba nalézt vhodné kandidátní body detektorem *GFTT*.

6.3 Experimenty

V této části budou popsány experimenty provedené se systémem pro sledování pohybu klíčových bodů. Půjde zejména o rychlost detekce bodů a výpočtu optického toku. Tyto testy mají prokázat vhodnost využití výpočtu optického toku pro sledování pohybu kamery v rámci zpracování videa v reálném čase.

Tyto experimenty byly provedeny na samostatném trackovacím systému, který stojí základem tomu využitému v rámci celé aplikace, čímž se dá určit následné chování v rámci celého systému.

6.4 Reference GFTT a KP

K tomu jsem implementoval jednoduchou aplikaci, s pomocí které jsem toto mohl vyhodnotit. Tato testovací aplikace má následující strukturu:

1. **Inicializace.**
2. **Načtení vstupního snímku** a převedení do šedotónového provedení.
3. **Popis snímku *GFTT*.**
4. **Popis snímku *SIFT* a *SURF*.**
5. **Vyhledání nejbližších bodů** do předem stanovené vzdálenosti pro každý detektor.
6. **Ukončení činnosti**, uvolnění zdrojů.

Následující tabulka znázorňuje dobu pro vyhledání a popis klíčových bodů.

Detektor	čas
GoodFeaturesToTrack	0.23314 s
SURF	3.35196 s
SIFT	4.79236 s

Následující tabulka ukazuje rozložení provázanosti jednotlivých detektorů na *GFTT*. Test byl proveden na fotce s vysokým rozlišením.

	SURF		SIFT	
Celkem nalezeno KB	6216		6693	
Max. vzdálenost	počet	průměrná chyba [px]	počet	průměrná chyba [px]
1px	25	0.627	60	0.627
2px	94	1.078	322	1.209
3px	243	1.604	844	1.658
4px	523	2.180	1678	2.084
5px	985	2.754	2823	2.481

Z výsledků lze vidět, že klíčové body SIFT detektoru jsou rozmístěny pro potřeby párování s GFTT lépe než body SURF. V závislosti na vstupním obraze postačí pro SIFT detektor vyhledávat páry do vzdálenosti 2px kdežto pro dosažení srovnatelných výsledků je potřeba pro SURF detektor vyhledávat do vzdálenosti alespoň 3px. Z výpočetního hlediska se nejedná o podstatný rozdíl, avšak se vzdáleností také roste průměrná chyba, což je nežádoucí.

6.5 Stabilita systému

Při testování na jednotlivých vstupních scénách vše probíhalo velmi dobře a výsledky byly dodatečně stabilní. Realizace výpočtu optického toku v rámci knihovny OpenCV nabízí velmi dobře optimalizovaný *Lucas-Kanade* algoritmus, jak bylo popsáno v realizaci.

6.6 Rychlost systému

Rychlost				
	Počet měření	min [s]	max [s]	avg [s]
Vyhledávní GFTT	500	0.018	0.112	0.06
Výpočet optického toku	500	0.035	0.018	0.024

Měřené výsledky prokazují, že zvolená metoda pro výpočet optického toku a vyhledávání klíčových bodů GFTT byla správná. Naměřené výsledky pochází z měření ve více scénách z video-vstupu s rozlišením $1980px \times 1020px$.

Kapitola 7

Lokalizátor kamery

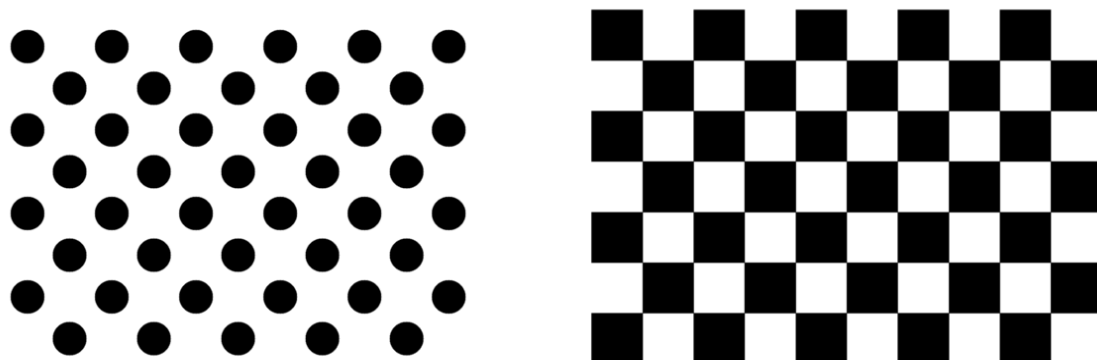
V této kapitole lze nalézt informace o návrhu a realizaci lokalizátoru. Wide baseline matching – neboli lokalizace vůči celému modelu daného prostředí. Jedná se o velmi podstatnou část navrženého systému. Více o návrhu lze nalézt v kapitole [4.3](#).

Nejprve bude popsána realizace, která zahrnuje i kalibraci kamery. Dále jsou popsány experimenty a jejich vyhodnocení.

7.1 Realizace

V této kapitole rozeberu realizaci samotného systému. Návrh i implementace byla provedena s využitím objektového přístupu. Systém je implementován v jazyce C++, pro podporu algoritmů počítačového vidění využívám knihovny OpenCV ve verzi 2.4, která nabízí širokou škálu funkcí a operací s obrazy na nižší úrovni abstrakce.

Kalibrace kamery



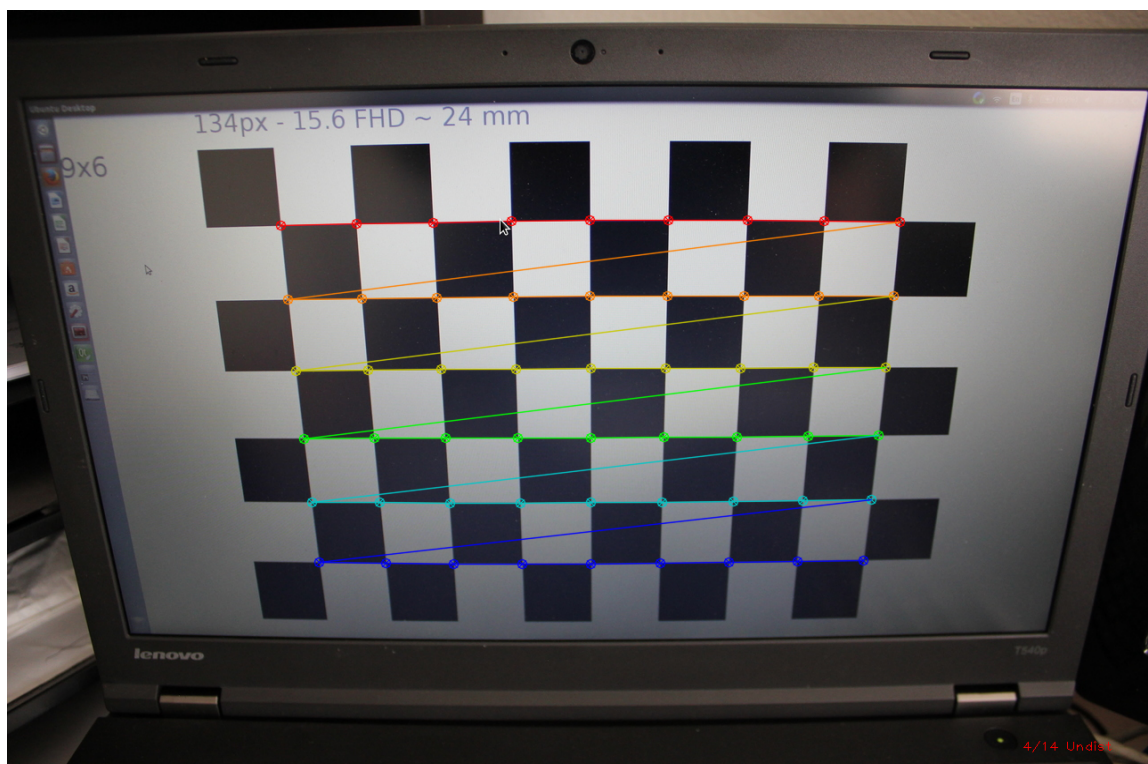
Obrázek 7.1: Dva nejčastěji používané vzory pro kalibraci kamery. Vlevo vzor s asymetricky rozloženými kruhy, vpravo šachovnicový vzor.

Navržený systém nevyžaduje vstup z kalibrované kamery, ale potřebuje mít informace o kalibrační matici předem známé. Více informací o kalibraci kamery naleznete v sekci [3.1](#). Vzhledem k tomu je nutné kalibraci kamery provést před samotným spuštěním a tato data systému poskytnout ve formě konfiguračního souboru. Pro tyto účely využívám vzorového

příkladu kalibrování kamery s knihovnou OpenCV [6]. Tato utilita přijímá za vstup konfigurační soubor, který určuje parametry kalibrace. Takové parametry mohou být například:

- kalibrační vzor, viz. obrázek 7.1,
- počet vstupních souborů,
- velikost obrazce v jednotkách reálného souřadného systému,
- seznam vstupních souborů.

Na základě těchto konfiguračních informací započne hledání kalibračního vzoru ve vstupních obrazcích, jak lze vidět na obrázku 7.2 a na základě nalezených pozic se vypočítá kalibrační matice a matice zkreslení. Na základě kalibrační matice se ještě vypočítá průměrná projektivní chyba - ta se spočítá tak, že se udělá projekce bodu objektu na obrazový bod a na základě rozdílu vzdálenosti mezi takto vypočteným bodem a bodem získaným pomocí vyhledávání kalibračního vzoru můžeme zjistit odchylku v pixelech. Takto se provede výpočet pro všechny vstupní obrazy a následným zprůměrováním získáme průměrnou projektivní chybu pro danou kalibrační matici. Tato chyba by se měla co nejvíce blížit k 0.



Obrázek 7.2: Nalezený kalibrační obrazec ve tvaru šachovnice.

7.2 Realizace lokalizátoru

Jádro systému lokalizátoru bylo navrženo modulárně s ohledem na budoucí rozšířitelnost. Při návrhu jsem vycházel zejména z životního cyklu lokalizace kamery v rámci modelu, kdy lze jeho činnost popsat v několika krocích:

1. **Inicializace.**
2. **Načtení modelu prostředí a kalibrace kamery** a následné zpracování modelu.
3. **Načtení snímku z kamery/video ke zpracování**, popis obrazu klíčovými body a výpočet jejich deskriptorů.
4. **Nalezení 2D-3D referencí** mezi vstupním snímkem a modelem.
5. **Výpočet matice kamery** pomocí algoritmu RANSAC.
6. **Činnost se opakuje od bodu 2.**, pokud existuje další vstupní snímek.
7. **Ukončení činnosti**, uvolnění zdrojů.

Načtení modelu prostředí a kalibrace kamery

Model prostředí je do aplikace předáván formou konfiguračního souboru ve formátu YAML¹ nebo XML², jelikož knihovna OpenCV nabízí rozhraní pro ukládání a načítání datových struktur do konfiguračních souborů. Toto rozhraní je zajištěno třídou `FileStorage`³. Tento konfigurační soubor nese veškeré informace potřebné pro práci s modelem. Mezi tato data patří:

- Mračno bodů - pozice v prostoru, viditelnost bodu z různých vstupních obrazů.
- Klíčové body vstupních obrazů a k nim deskriptory.
- Cesta k původním souborům a jejich názvy.
- Matice kamer vypočtených při tvorbě modelu.
- Kalibrační matice kamery vstupních souborů.

Veškerou práci s konfiguračním souborem tvoří třída `myLoader`. Jako parametr konstruktoru přijímá cestu ke konfiguračnímu souboru. Dále tato třída také zajišťuje tvorbu stromové struktury deskriptorů ze vstupního mračna. Tato stromová struktura dále slouží jako vyhledávací strom při hledání korespondujících bodů.

Načtení snímku a hledání klíčových bodů

Vstupní parametr určuje, zda-li lokalizátor využívá vstupního video souboru nebo bude přijímat vstup z vestavěné či připojené kamery. Zpracování snímků probíhá bez vynechávání, tudíž dokud neproběhne celá smyčka výpočtu, další snímek se nenačítá.

Pro načítání snímku se využívá knihovny OpenCV, která podporuje celou řadu vstupních formátů a umožňuje variabilní nastavení také vstupního streamu z připojené kamery.

Vyhledávání klíčových bodů obstarává třída `myDetector`, která dává možnost nastavení různých parametrů. Pro stabilní vyhledávání korespondencí mezi snímky se zpravidla využívá SIFT, popsáný v kapitole 2.2 nebo SURF, popsáný v kapitole 2.3. Knihovna OpenCV nabízí velkou škálu detektorů klíčových bodů, také jako jejich deskriptorů. Pro tyto účely

¹ <http://cs.wikipedia.org/wiki/YAML>

² <http://cs.wikipedia.org/wiki/XML>

³ http://docs.opencv.org/modules/core/doc/xml_yaml_persistence.html

je vhodné využít rozhraní `FeatureDetector`, respektive `DescriptorExtractor`. Toto rozhraní dává možnost jednoduše měnit typ detektoru a deskriptoru bez nutnosti dělat složité zásahy v kódu.

Mezi podporované *detektory* patří:

- `SiftFeatureDetector` - hledání SIFT bodů.
- `SurfFeatureDetector` - hledání SURF bodů.
- `ORB` - hledání bodů ORB, neboli Oriented Brief.
- `GoodFeaturesToTrackDetector` - hledání rohů v obraze, které splňují určité požadavky na kvalitu. Vhodné pro tracking.
- `FastFeatureDetector` - hledání FAST bodů. Vhodné pro tracking.?

Některé podporované *deskriptory* využitelné v OpenCV jsou:

- `SiftDescriptorExtractor` - popis klíčových bodů deskriptory SIFT.
- `SurfDescriptorExtractor` - popis klíčových bodů deskriptory SURF.
- `ORB` - popis klíčových bodů deskriptory ORB.
- `FREAK` - Fast Retina Keypoint - popis klíčových bodů inspirovaný lidským viděním.

Nalezení 2D-3D referencí

Při načítání konfiguračního souboru se také tvoří struktura pro vyhledávání korespondencí mezi vstupním snímkem a modelem. Pro vyhledání korespondencí se využívá knihovny FLANN⁴, která je implementována v rámci OpenCV. Tato knihovna nabízí možnost využití rychlého hledání nejbližších sousedů včetně systému pro automatickou volbu algoritmu a optimálních parametrů v závislosti na datové sadě.

Jakmile jsou nalezeny korespondující si deskriptory, přichází na řadu vyhledání 2D-3D referencí. Sada deskriptorů ze vstupního obrazu popisuje určité klíčové body, takže se sestaví vektor ukazatelů do vstupního obrazu (2D souřadnice) a k nim se utvoří vektor stejné velikosti ze sady deskriptorů ze sady vyhledávané v modelu. Tyto deskriptory spadají k některému bodu v prostoru. Každý bod v prostoru je popsán nejméně dvěma deskriptory.

Toto zajišťuje třída `myDetector`, která obsahuje metody pro detekci, popis a vyhledávání klíčových bodů.

Výpočet matice kamery

Samotná lokalizace je zapouzdřena do třídy `myLocalisation`. Pro výpočet matice kamery je potřeba sada 2D bodů a k nim korespondující body ve 3D prostoru. Na základě těchto korespondencí lze poté vypočítat translační a rotační vektor (t, r) dané kamery.

Pro výpočet využívám PnP algoritmu založeném na RANSACu, což zajišťuje dobrou odolnost vůči *outlierům*. Bližší informace jsou diskutovány v kapitole 3.4.

Z těchto vektorů lze poté jednoduše vytvořit matici pro danou kameru, která je ve formátu 4×3 .

⁴ <http://www.cs.ubc.ca/research/flann/>

Rotační vektor se s pomocí funkce **Rodrigues** transformuje na rotační matici R 3×3 , která je základem pro matici kamery. Čtvrtý chybějící sloupec matice je tvořen translačním vektorem. Takže výsledná hledaná matice kamery bude ve tvaru:

$$cam = \begin{bmatrix} R(0,0) & R(0,1) & R(0,2) & t(0) \\ R(1,0) & R(1,1) & R(1,2) & t(1) \\ R(2,0) & R(2,1) & R(2,2) & t(2) \end{bmatrix} = [R \mid t] \quad (7.1)$$

7.3 Experimenty

V této části budou popsány realizované experimenty, které budou zkoumat kvalitu lokalizace v rámci daného modelu. Dále také rychlost prováděné lokalizace a ale také na navrženém systému pro lokalizaci. Testy lokalizace a celkové aplikace byly prováděny na notebooku s procesorem *Intel Core 2 Duo E6670*, který poskytuje výpočetní výkon dvou jader s maximální frekvencí 2.2GHz a disponuje 8GB operační paměti.

Další testy budou na trackovací systém, v jakých podmínkách bude sledování stabilní a dostatečně robustní. Další testy ověří kvalitu lokalizace v jednotlivých modelech. Vyhodnocovat budu rychlost a projektivní chybu. K těmto účelům byla vytvořena aplikace, která náhodně vybírá fotografie ze kterých bude vytvořen referenční model a fotografie ve kterých se bude provádět lokalizace.

7.4 Kvalita lokalizace

V této části budou představeny naměřené výsledky přesnosti lokalizace. Pro účely testování byl sestaven testovací skript, který ve vytvořených datových sadách nejprve tvořil referenční modely, které se skládaly ze všech dostupných fotografií. Následně se náhodně některé fotografie při tvorbě modelu vyloučily a využily se pro vyhodnocení přesnosti. Přesnost byla analyzována pomocí projektivní chyby.

Dále byla vyhodnocena náročnost jednotlivých kroků kroků při lokalizaci.

V následující tabulce lze vidět hodnoty naměřené při testování přesnosti lokalizace a úspěšnosti nalezení pozice dané kamery v prostoru.

Velikost modelu udává počet fotografií ze kterých daný model byl celkem utvořen. V rámci těchto testů se tvořil referenční model na lokalizaci ze všech fotografií kromě jedné a tato se lokalizovala vůči modelu, toto proběhlo pro každou fotografii z datové sady.

Jednotlivé modely odpovídají představeným typům modelů v sekci 5.4:

1. **Planární** - fotky rozmístěné na rovné zdi.
2. **Prostorový** - běžný prostor otevřenějšího charakteru.
3. **Vizuálně zaplněný**.
4. **Vizuálně prázdný**.
5. **Reálně zaplněný** prostor, nemusí ale vizuálně působit jako plný (čisté plochy atp.).
6. **Reálně prázdný** prostor.

Model 1

Tento model je tvořen sadou fotografií na rovné ploše. Obsahuje velké množství vizuálních rysů proto lze předpokládat vysokou přesnost lokalizace a minimum neúspěchu.

Velikost modelu	Průměrná projekční chyba [px]	Úspěšnost lokalizace
90	4.24	82/90 [91%]
81	4.37	72/81 [89%]
72	5.16	65/72 [90%]
63	4.89	50/63 [79%]
54	5.01	45/54 [83%]
45	5.64	36/45 [80%]
36	5.79	29/36 [81%]
27	5.15	20/27 [74%]
18	5.63	13/18 [72%]
9	6.13	6/9 [67%]

Jak lze vidět v tabulce, tak předpoklad byl správný. Projekční chyba pohybující se mezi 4 a 6 px je dle očekávání dobrá. Daný pokles úspěšných lokalizací s klesající velikostí modelu byla předpokládána, jelikož jak již bylo ukázáno v experimentech s kvalitou tvořených modelů, tak u modelů s malým počtem fotografií – bez překrytí je sice nízká projekční chyba, ovšem je výrazně nižší počet bodů, ze kterých je model tvořen, tudíž dochází k tomu, že není nalezen dostatečný počet referencí mezi fotografií a 3D modelem.

Model 2

Tento model je tvořen objekty umístěnými uprostřed místnosti, takže neobsahuje výraznější omezení stěnami. Tento model byl snímán ze všech směrů, objekty jsou si vizuálně podobné, proto lze předpokládat vyšší chybovost než u modelu předchozího. Lze také předpokládat nižší míru úspěšnosti lokalizace (více vstupních obrazů bez nalezeného odhadu pozice kamery), zejména u menších modelů.

Velikost modelu	Průměrná projekční chyba [px]	Úspěšnost lokalizace
90	5.16	81/90 [90%]
81	5.38	70/81 [87%]
72	4.99	60/72 [83%]
63	5.21	49/63 [78%]
54	6.05	41/54 [76%]
45	6.54	32/45 [70%]
36	6.29	24/36 [66%]
27	6.47	16/27 [60%]
18	7.18	8/18 [45%]
9	7.22	3/9 [37%]

Jak lze vidět v tabulce, tak průměrná chyba nalezených kamer byla o něco málo vyšší než v předchozím případě. Poměr úspěšně lokalizovaných kamer k těm, které lokalizovány nebyly

byl u větších modelů velmi dobrý, dle předpokladu ovšem u modelů tvořených menším počtem fotografií i úspěšnost správného odhadu pozice kamery klesla. Toto je zapříčiněno malým počtem stabilních. bodů

Model 3

Tento model je tvořen zejména z knih, které na sobě mají mnoho obrázků a textů. Jsou v prostoru a vizuální informace v něm obsažená je velmi bohatá. Neobsahuje téměř žádné volné plochy. Lze předpokládat vysokou přesnost lokalizace a minimum neúspěchu.

Velikost modelu	Průměrná projekční chyba [px]	Úspěšnost lokalizace
90	4.17	83/90 [92%]
81	4.64	71/81 [88%]
72	4.84	61/72 [85%]
63	4.32	52/63 [82%]
54	4.51	43/54 [79%]
45	5.17	35/45 [78%]
36	4.22	27/36 [74%]
27	5.09	20/27 [75%]
18	5.69	13/18 [72%]
9	6.01	6/9 [67%]

Jak lze vidět v tabulce, tak předpoklad byl správný. Kvalita lokalizace byl relativně shodná s prvním modelem – oba tyto modely jsou bohaté na vizuální informace, což dopomáhá kvalitnímu nalezení korespondencí. Celková úspěšnost nalezení byla taktéž velmi podobná modelu č.1.

Model 4

Tento model je tvořen zejména nevýraznými krabičkami bez obrázků či textů. Tento model je velmi prostý na vizuální informace, obsahuje mnoho volných jednobarevných ploch. Předpokladem je, že úspěšnost bude nižší než v předchozích případech. Chyba nalezených by měla být relativně stejná jako v předchozích případech.

Velikost modelu	Průměrná projekční chyba [px]	Úspěšnost lokalizace
90	5.27	77/90 [85%]
81	4.97	66/81 [82%]
72	5.37	56/72 [78%]
63	5.74	47/63 [74%]
54	5.89	36/54 [66%]
45	6.02	28/45 [62%]
36	5.92	19/36 [54%]
27	5.45	11/27 [42%]
18	6.15	6/18 [33%]
9	4.52	2/9 [19%]

Jak lze vidět v tabulce, tak předpoklad byl správný. Chyba se pohybuje stále do 7px. Relativní počet nalezených kamer vůči nenalezeným klesl. U malých modelů byla úspěšnost velmi malá. Tuto nízkou úspěšnost lze vysvětlit nízkou hustotou vizuálních informací na základě kterých se tvoří klíčové body potřebné pro nalezení referencí mezi snímkem a modelem.

Model 5

Tento model je tvořen různorodými objekty, jedná se o běžné, relativně zaplněné prostředí – pracovní stůl s okolím. Předpokladem je, že úspěšnost nalezení bude poměrně dobrá a chyba průměrně shodná s předchozími případy.

Velikost modelu	Průměrná projekční chyba [px]	Úspěšnost lokalizace
90	5.13	82/90 [91%]
81	4.88	70/81 [86%]
72	5.00	59/72 [82%]
63	4.78	50/63 [80%]
54	4.55	43/54 [79%]
45	5.35	35/45 [77%]
36	5.32	27/36 [75%]
27	5.15	20/27 [74%]
18	5.21	13/18 [72%]
9	5.98	6/9 [67%]

Jak lze vidět v tabulce, tak předpoklad byl poměrně správný. Zaplněný prostor poskytuje dostatečné množství vizuálních informací pomocí kterých lze poté úspěšně vyhledávat korespondence. Výsledky jsou velmi podobné prvnímu modelu.

Model 6

Tento model je tvořen běžným prostředím, které je ovšem takřka prázdné. Předpokládám výsledky podobné s modelem č.4, jelikož oba tyto modely nedisponují velkým množstvím vizuálních informací.

Velikost modelu	Průměrná projekční chyba [px]	Úspěšnost lokalizace
90	5.34	77/90 [85%]
81	4.13	67/81 [83%]
72	5.68	57/72 [89%]
63	5.59	48/63 [76%]
54	4.57	41/54 [75%]
45	6.71	35/45 [77%]
36	6.23	27/36 [75%]
27	5.32	20/27 [74%]

18	4.43	12/18 [67%]
9	6.39	6/9 [65%]

Jak lze vidět v tabulce, tak předpoklad byl správný. Projekční chyba pohybující se mezi 4 a 6 px je dle očekávání dobrá. Daný pokles úspěšných lokalizací s klesající velikostí modelu byla předpokládána, jelikož jak již bylo ukázáno v experimentech s kvalitou tvořených modelů, tak u modelů s malým počtem fotografií – bez překrytí je sice nízká projekční chyba, ovšem je výrazně nižší počet bodů, ze kterých je model tvořen.

7.5 Rychlost systému

V této části bude analyzována rychlost navrženého systému. Jedná se o data naměřená při spouštění předchozích testů stability systému. Následující tabulka ukazuje, jaké je poměrné rozložení rychlosti dílčích částí systému.

Činnost	min [s]	max [s]	průměr [s]
Detekce klíčových bodů a popis deskriptory	1.24	4.26	2.85
Vyhledání 2D-3D referencí	0.088	0.941	0.736
Nalezení matice kamery	0.094	0.145	0.113
Celkem	1.617	6.752	3.332

Naměřené hodnoty ukazují, že nejslabším článkem systému je vyhledávání a popis klíčových bodů. Toto je zapříčiněno zejména velikostí vstupního obrazu. Bylo by možné využít menšího rozlišení, bylo by to ale na úkor přesnosti. Vzhledem k tomu, že lokalizační metoda je využita pro inicializaci pozice kamery a následně pro upřesňování polohy kamery vůči modelu, delší doba zpracování není problémem. Pro využití v real-time aplikacích lze tohoto systému využít v kombinaci se systémem pro sledování klíčových bodů, jak bylo navrženo v kapitole 4.

Kapitola 8

Závěr

Cílem této práce bylo navrhnout systém pro lokalizaci kamery v rámci 3D modelu prostředí rekonstruovaného ze sady fotografií. Následně takto lokalizovanou kameru využít pro realizaci rozšířené reality. Dále navržený systém implementovat, provést testy přesnosti a stability v různých scénách.

Teoretická část se zabývá vyhledáváním a popisem klíčových bodů, jelikož je to podstatná část potřebná jak pro rekonstrukci scény, tak pro následné možnosti lokalizace. Dále se věnuje metodám využívaným pro rekonstrukci 3D modelu scény, tak jako metodami pro lokalizaci kamery.

Další důležitá kapitola obsahuje návrh dílčích částí systému. Zde jsou diskutovány již realizované podobné systémy.

První rozebranou částí je model. V této kapitole je popsána tvorba datových sad na základě nabraných zkušeností v průběhu práce, která dále pojednává o nástrojích využitých při práci a rozebírá experimenty provedené při tvorbě různých modelů a diskutuje dosažené výsledky.

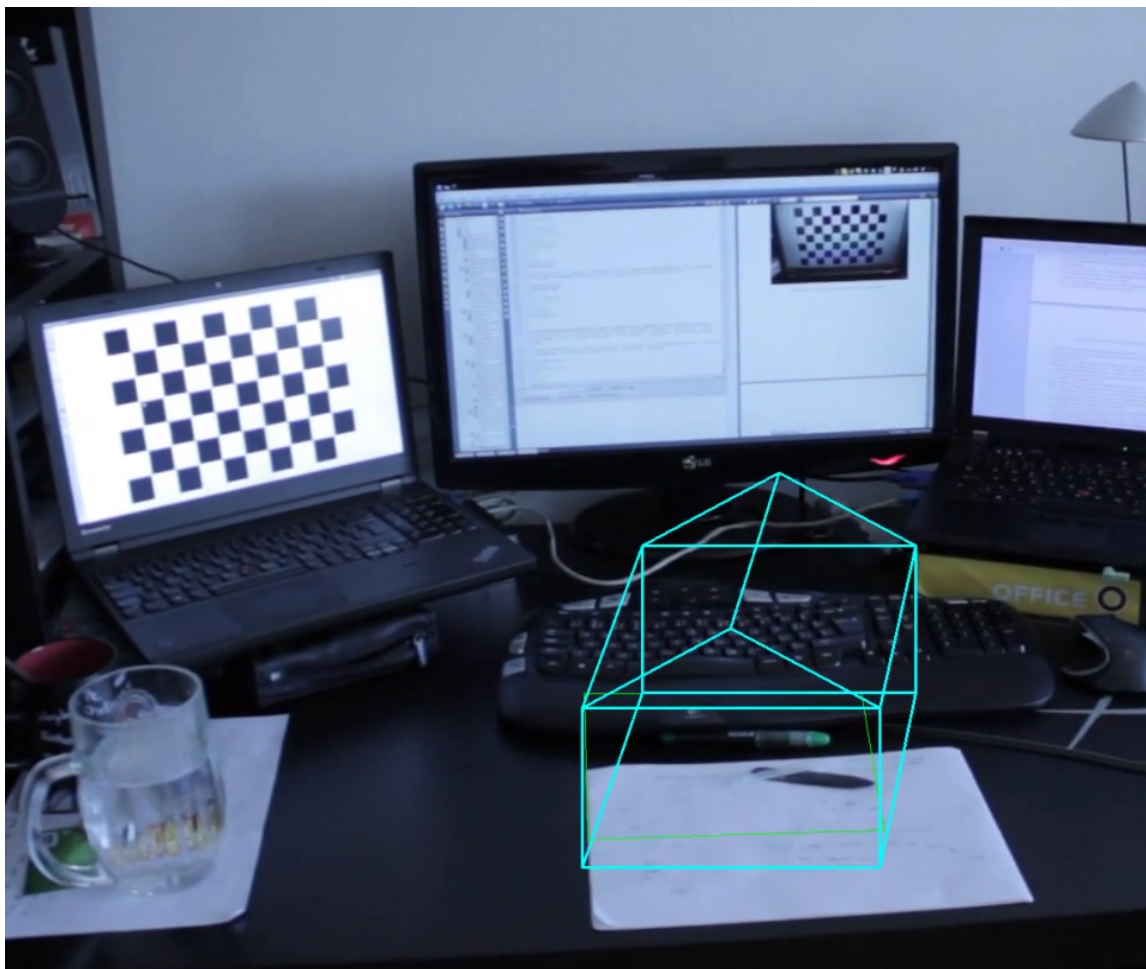
Dále je popsána část systému zabývající se sledováním pohybu kamery v reálném čase pomocí sledování optických toků. V této kapitole je popsána nejen realizace, ale i provedené experimenty a diskutovány jejich výsledky.

Následující kapitola popisuje realizaci lokalizační části systému a experimenty provedené pro zjištění přesnosti a rychlosti navrženého řešení.

Jak je zřejmé, práce je rozdělena do několika dílčích částí, které jsou detailně popsány. Realizovaný systém využívá pro inicializaci k odhadu pozice kamery navržených metod, které umožňují aby byl systém dostatečně robustní. Toto má za následek, že je systém pomalý, takže nevhodný pro použití v aplikaci která vyžaduje práci v reálném čase.

Aby bylo možné realizovat rozšířenou realitu, bylo nutné využít metod pro sledování pohybu kamery v reálném čase. Pro tyto potřeby byla navržena metoda, která zajišťuje propojení přesné lokalizace s následným sledováním klíčových bodů pro rychlý odhad změny pozice kamery. Inspirace pochází z metody PTAM, kde je systém také rozdělen do více vláken a každé zajišťuje chod jedné části systému. V případě této práce jedno vlákno obsluhuje lokalizaci, která udává iniciální pozici kamery a dále poskytuje body, jenž mají vazbu z obrazu (2D) na model prostředí (3D). Tyto informace využívá trackovací systém a díky výpočetní nenáročnosti výpočtu optického toku zajišťuje bezproblémový chod i na video-vstupech s vysokým rozlišením a zároveň zachovává vysokou míru přesnosti.

V jednotlivých kapitolách jsou prezentována i možná budoucí rozšíření – například optimalizace vyhledávání v rámci omezené části modelu na základě předchozí znalosti pozice kamery.



Obrázek 8.1: Výsledkem práce je demo aplikace realizující základní rozšířenou realitu pracující v reálném čase.

Výsledky experimentů tvorby modelu ukazují důvod, proč se stále v metodách zabývajících se rekonstrukcí prostředí využívá zejména metody hledání a popisu klíčových bodů SIFT a ne metody SURF, která by dle autorů měla poskytovat dostatek robustnosti a lepší výkonost oproti metodě SIFT. Z experimentů vyplynulo, že tomu tak není a metoda SURF není příliš vhodnou variantou pro potřeby rekonstrukce prostředí.

Realizovaný systém by pro produkční nasazení vyžadoval ještě jisté úpravy, týkající se zejména implementace z pohledu synchronizace dat mezi vlákny. Na základě dosažených výsledků lze navrženou metodu hodnotit jako použitelnou pro potřeby robustní lokalizace v rámci předem vytvořeného modelu, tak jako ve spojení se sledováním optického toku nabízí dostatek výkonu pro realizaci rozšířené reality v reálném čase. Ukázku demo aplikace lze vidět na obrázku 8.1.

Literatura

- [1] Baillard, C.; Zisserman, A.: Automatic Reconstruction of Piecewise Planar Models from Multiple Views. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1999, s. 559–565.
- [2] Bay, H.; Ess, A.; Tuytelaars, T.; aj.: Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, ročník 110, č. 3, 2008: s. 346 – 359, ISSN 1077-3142.
URL <http://www.sciencedirect.com/science/article/pii/S1077314207001555>
- [3] Bradski, G. R.; Kaehler, A.: *Learning OpenCV - computer vision with the OpenCV library: software that sees*. O'Reilly, 2008, ISBN 978-0-596-51613-0, I-XVII, 1-555 s.
- [4] Emami, S.; Ievgen, K.; Mahmood, N.: *Mastering OpenCV with Practical Computer Vision Projects*. Packt Publishing, Limited, 2012, ISBN 9781849517829.
URL <http://books.google.cz/books?id=GXewmAEACAAJ>
- [5] Fischler, M. A.; Bolles, R. C.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, ročník 24, č. 6, 1981: s. 381–395.
- [6] Gábor, B.: Camera calibration With OpenCV. 2014, [cit. 2014-05-11].
URL http://docs.opencv.org/doc/tutorials/calib3d/camera_calibration/camera_calibration.html
- [7] Görig, J.: *Model dynamické scény pro robota*. Diplomová práce, FIT VUT, Brno, 2012.
- [8] Hartley, R.; Zisserman, A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press, druhé vydání, 2003, ISBN 0-521-54051-8.
- [9] Hoiem, D.: Feature Tracking and Optical Flow, Computer Vision, University of Illinois. 2012, [cit. 2014-05-05].
URL <http://courses.engr.illinois.edu/cs543/sp2012/>
- [10] Klein, G.; Murray, D.: Parallel Tracking and Mapping for Small AR Workspaces. In *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, Nara, Japan, November 2007.
- [11] Lin, H.-Y.: *Computer Vision Techniques for Complete 3D Model Reconstruction*. Dizertační práce, State University of New York, 2002.
- [12] Long Quan, Z. L.: Linear N-Point Camera Pose Determination. In *IEEE Transactions on pattern analysis and machine intelligence*, ročník 21, July 1999, s. 774–780.

- [13] Lowe, D. G.: Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vision*, ročník 60, č. 2, Listopad 2004: s. 91–110, ISSN 0920-5691.
URL <http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94>
- [14] Ma, Y.; Soatto, S.; Kosecka, J.; aj.: *An Invitation to 3-D Vision: From Images to Geometric Models*. SpringerVerlag, 2003, ISBN 0387008934.
- [15] Markéta Dubská, B. P.: Přehled pojmů výpočetní geometrie. 2013, [cit. 2014-04-17].
URL <http://medusa.fit.vutbr.cz/stereo/doc/vge.pdf>
- [16] Shapiro, L.; Stockman, G.: *Computer vision*. Prentice Hall, 2001, ISBN 9780130307965.
URL <http://books.google.cz/books?id=FftDAQAAIAAJ>
- [17] Stockman, G.; Shapiro, L. G.: *Computer Vision*. Upper Saddle River, NJ, USA: Prentice Hall PTR, první vydání, 2001, ISBN 0130307963.
- [18] Szeliski, R.: *Computer Vision: Algorithms and Applications*. New York, NY, USA: Springer-Verlag New York, Inc., první vydání, 2010, ISBN 1848829345, 9781848829343.
- [19] Tobler, R. F.; Maierhofer, S.: A Mesh Data Structure for Rendering and Subdivision. 2006.
- [20] Viala, C.; Sanchez Salmeron, A.: Performance evaluation of linear camera calibration techniques. In *Automation Congress, 2004. Proceedings. World*, ročník 18, June 2004, s. 49–54.
- [21] Wu, C.: VisualSFM : A Visual Structure from Motion System. 2013, [cit. 2014-01-09].
URL <http://ccwu.me/vsfm/>
- [22] Yang, A. Y.: Image Formation and Camera Models. 2007, [cit. 2013-12-15].
URL <http://inst.eecs.berkeley.edu/~ee225b/sp07/lectures/lec12.pdf>
- [23] Zhang, Z.: A flexible new technique for camera calibration. 2003, [cit. 2013-12-16].
URL <http://research.microsoft.com/en-us/um/people/zhang/Papers/TR98-71.pdf>