

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

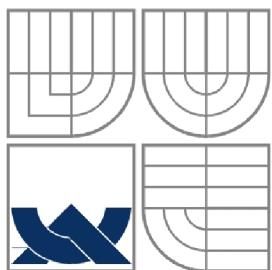
ORIENTACE KAMERY V REÁLNÉM ČASE

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

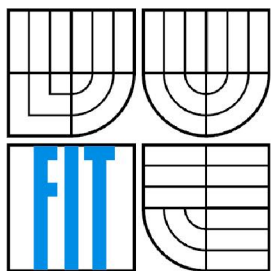
AUTOR PRÁCE
AUTHOR

Bc. Jiří Župka

BRNO 2010



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

ORIENTACE KAMERY V REÁLNÉM ČASE

CAMERA ORIENTATION IN REAL-TIME

SEMESTRÁLNÍ PROJEKT

AUTOR PRÁCE
AUTHOR

Bc. Jiří Župka

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. Vítězslav Beran

BRNO 2010

Abstrakt

Tato práce se zabývá orientací kamery v reálném čase pomocí záběrů z jediné kamery. Offline metody jsou zde popsány a použity jako reference pro srovnání metod pracujících v reálném čase. Metody pracující v reálném čase MonoSlam a PTAM jsou zde popsány a porovnány. Dále jsou v práci nastíněny pokročilé postupy, na kterých je možné nadále pracovat.

Abstract

This work deals with the orientation of the camera in real-time with a single camera. Offline methods are described and used as a reference for comparison of a real-time methods. Methods work in real-time Monocular SLAM and PTAM methods are there described and compared. Further, paper shows hints of advanced methods whereas future work is possible.

Klíčová slova

PTAM, Parallel Tracking and Mapping, scenelib, MonoSlam, Bundle Adjustment, Kalman Filtr, EKF, Extend Kalman Filter, UKF, Unscented Kalman filtr, Epipolární geometrie, epipolar search, epipolar line, quatarion, Lie Group SE(3), RANSAC, Homografie, KLT filtr, Pinhole model, Levenberg-Marquardt

Keywords

PTAM, Parallel Tracking and Mapping, scenelib, MonoSlam, Bundle Adjustment, Kalman Filtr, EKF, Extend Kalman Filter, UKF, Unscented Kalman filtr, Epipolar geometry, epipolar search, epipolar line, quatarion, Lie Group SE(3), RANSAC, Homography, KLT filtr, Pinhole model, Levenberg-Marquardt

Citace

Jiří Župka: Orientace kamery v reálném čase, diplomová práce, Brno, FIT VUT v Brně, 2010

Orientace kamery v reálném čase

Prohlášení

Prohlašuji, že jsem tuto diplomovou vypracoval samostatně pod vedením Ing. Vítězslava Berana. Další informace jsem čerpal z odborných článků publikovaných na internetu. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem vycházel.

.....
Jiří Župka
6.5.2010

Poděkování

Chtěl bych poděkovat Ing. Vítězslavu Beranovi, za vedení a udávání směru této práce. Dále Elišce Klímové za korekturu textu.

© Jiří Župka, 2010.

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

Obsah.....	1
1 Úvod.....	4
2 Základní popis metod SLAM a PTAM.....	6
2.1 Kalman Filtr.....	6
2.2 Rohový detektor Shi and Tomasi.....	9
2.3 KLT Tracker.....	10
2.4 Pinhole model kamery.....	11
2.5 Epipolární geometrie	12
2.6 Bundle Adjustment.....	15
2.7 Výpočet ohniskové vzdálenosti v pixelech.....	16
3 Referenční offline metody.....	17
3.1 Vyhledání význačných oblastí.....	17
3.2 Trasování význačných oblastí	18
3.3 Počítání 3D pohybu kamery.....	18
3.4 Přejít od offline k real-time metodám.....	19
4 MonoSLAM.....	20
4.1 Pravděpodobnostní 3D Mapa.....	20
4.2 Význačné oblasti.....	21
4.3 Inicializace systému.....	22
4.4 Modelování pohybu a jeho predikce.....	23
4.5 Aktivní měření význačných bodů a upravování Mapy.....	24
4.6 Inicializace význačných oblastí.....	25
4.7 Správa Mapy.....	26
4.8 Orientace významných oblastí.....	26
4.9 Souhrn metody.....	27
5 PTAM.....	28
5.1 Mapa.....	28
5.2 Trasování.....	29
5.3 Mapování a inicializace mapy.....	31
5.4 Vytváření mapy prostředí.....	31
5.5 Znovunalezení pozice kamery.....	33
5.6 Souhrn metody.....	34
5.7 Postup kalibrace kamery.....	34
6 Teoretické porovnání metod	36

6.1	Vzájemné porovnání přesnosti metod	37
6.2	Odolnost vůči rozmazání.....	37
6.3	Problém kalibrace.....	38
6.4	Inicializace metod.....	38
6.5	Problém zorného úhlu.....	39
7	Srovnání metod na základě pokusů.....	40
7.1	Postup testování.....	41
7.2	Hardwarové vybavení.....	41
7.3	Softwarové nástroje.....	42
7.4	Vliv nastavení metod	43
7.5	Rozloha snímaného prostředí.....	48
7.6	Vliv snímací soustavy.....	49
7.7	Barevné vlastnosti.....	51
7.8	Pohyblivé a měnící se objekty ve scéně	51
7.9	Vliv náhodných výpadků snímků.....	52
8	Pokročilejší postupy a návrh demo-aplikací.....	54
8.1	Pomocník nevidomých lidí.....	55
8.2	Navigátor autonomních průmyslových vozíků.....	55
8.3	Návrh interiérů.....	56
8.4	Hra v reálném prostředí.....	56
8.5	Prohlížení 3D modelů v prostoru.....	56
9	Závěr.....	57
10	Seznam příloh.....	61
	Příloha 1. Obsah elektronického média.....	62
	Příloha 2. Popis videosekvencí	63

Seznam tabulek

Tabulka 6.1: Porovnání vlastností metod MonoSLAM a PTAM.....	36
Tabulka 7.1: Vlastnosti webových kamer použitých při testování.....	42
Tabulka 7.2: Závislost přesnosti metody PTAM na počtu klíčových snímků.....	44
Tabulka 7.3: Závislost přesnosti metody MonoSLAM na počtu význačných oblastí.....	46
Tabulka 7.4: Tabulka závislostí nepřesnosti kalibrace na přesnosti měření.....	48
Tabulka 7.5: Výsledky měření přesnosti širokoúhlé a normální web kamery.....	50
Tabulka 7.6: Výsledky vlivu rozmazání na přesnost testovaných metod.....	50
Tabulka 7.7: Výsledky testu pohyblivých objektů velikost chyby od referencí metody.....	52
Tabulka 7.8: Vliv velikosti výpadku snímků na velikost chyby metody.....	53

1 Úvod

Problematika orientace kamery v prostoru v reálném čase pouze ze snímaného obrazu není triviální. Skupina metod, která se jí zabývá, se nazývá SLAM (Simultaneous Localization and Mapping). Metody SLAM jsou realizovány rozličným způsobem, například pomocí stereo vidění dvou kamer, kdy vždy máme informaci o hloubce význačných prvků ve scéně. Další skupina využívá jen jednu kameru, ale pro zjišťování hloubky záběru používá laserových či akustických čidel. V této práci budou rozebrány metody používající výhradně informace ze záběru jedné kamery, bez dalších přídavných informací o okolním prostoru. Konkrétně se bude jednat o metody MonoSLAM (Single Camera SLAM) a PTAM (Parallel Tracking and Mapping). Jedná se o dvě různé metody řešící orientaci kamery v prostoru v reálném čase za použití jediné kamery. Principy obou metod budou dopodrobna vysvětleny v následujících kapitolách.

Jako první bude kapitola **Základní popis metod SLAM a PTAM**. Jedná se o nastínění základních principů obou metod, jsou v ní vysvětleny základní metody dnes používané na sestavení metod pro orientaci kamery v prostředí, jako např. Kalman filtr, Epipolární geometrie a Bundle Adjustment. Dále první kapitola obsahuje objasnění dalších pojmů potřebných pro pochopení metod. V následující kapitole jsou popsány **Referenční offline metody** v této práci použité jako referenční metody pro vzájemné srovnání obou metod pracujících v reálném čase a bude zde navíc nastíněna práce offline metod. V kapitole **MonoSLAM** je podrobně popsána funkce metody. V další kapitole je rozebrána metoda **PTAM**. Kapitola **Teoretické porovnání metod MonoSLAM a PTAM** obsahuje porovnání obou metod, jsou v ní rozebírány vlastnosti metod z teoretického hlediska. Také je zde nastínění problémů, se kterými by se mohly metody potýkat. V následující kapitole **Porovnání metod MonoSLAM a PTAM** jsou na základě pokusů metody prakticky porovnány a rovněž uvedeny výsledky porovnání s přidanou úvahou, jak testované metody pracovaly. V předposlední kapitole **Nastínění pokročilejších postupů a demo-aplikací** jsou rozebrány možná budoucí použití metod pro orientaci v prostoru v reálném čase. V kapitole obsahující **přílohy** je popsán obsah média a uvedeny postupy úpravy videosekvencí použitých v kapitole „Porovnání metod MonoSLAM a PTAM“.

V rámci semestrálního projektu byl z velké části vypracován obecný teoretický základ, který byl v rámci diplomové práce ještě rozšířen v kapitole (2). Dále byly převzaty a doplněny teoretické kapitoly zabývající se přímo metodami pracujícími v reálném čase (4) a (5) a kapitola věnující se jejich teoretickým vzájemným porovnáním (6). Poslední kapitolou, jež byla převzata a doplněna v rámci diplomové práce, byly pokročilé postupy nad metodami pracujícími v reálném čase (8). Nově byly přidány kapitoly týkající se offline metod (3) a samotného porovnávání metod (7).

V rámci celé práce budou odkazy literaturu uváděny v []. Odkazy na prvky v samotné práci budou uváděny v (). Bude se jednat hlavně o odkazy na kapitoly, rovnice, obrázky a tabulky.

2 Základní popis metod SLAM a PTAM

V této kapitole budou popsány základní myšlenky, na kterých lokalizace kamery v prostotu funguje. Dále zde budou uvedeny základní a důležité metody pro lepší pochopení problematiky.

Hlavní myšlenka metod pracujících jen se snímky jediné kamery je srovnání dvou po sobě jdoucích snímků. Na srovnávaných snímcích se najdou význačné (invariantní) body, například pomocí metody FAST nebo Shi and Tomasi. Nalezené body z obou obrázků se sesouhlasí například pomocí metod RANSAC (Random Sample Consensus) či metodou KLT (2.3) a na korespondujících bodech se vypočítá např. Homografní matice. Homografní matice, nebo její obdoba, je schopna popsat transformace mezi obrázky, a tím zjistit pohyb kamery. Pro zjištění pozice kamery je, kromě Homografní matice, možné použít i Epipolární geometrii popsanou v kapitole (2.5). Další možná metoda využívá jako základ Kalman Filtr, který je popsán v kapitole (2.1). Kalman Filtr realizuje výpočet pohybu kamery a jako vstupy jsou mu dávány polohy význačných oblastí.

Kalman Filtr a jeho obdoby, jako jsou EKF (Extend Kalman Filter) či UKF (Unscented Kalman Filtr), mohou být použity v real-time verzích SLAM metod používajících jen jednu kameru. Díky Kalman Filtru je možné předpovídat polohu kamery, a tím ušetřit velké množství výpočtů na sesouhlasení dvou různých snímků. Kalman filtr bude popsán v další části dokumentace, která se bude snažit o základní vysvětlení činnosti, jelikož porozumění činnosti Kalman filtru je důležité pro metodu MonoSLAM [1].

V této práci jsou rozebrány a popsány dvě metody zastupující významné proudy v detekci pohybu jedné kamery pracující v reálném čase. Obě metody předpokládají pohyb kamery ve statickém prostoru. První z metod je MonoSLAM, kterou vytvořil Andrew Davison. Je založená na statistickém vyhodnocování polohy a stavu v prostoru s využitím EKF. Druhou metodou je PTAM (Parallel Tracking and Mapping) [2], která je založená na jednoduchém trasování cesty a metodě mapování založenou na Epipolární geometrii. Mapa je následně zpřesňována několikrát náročnější metodou Bundle Adjustment (2.6). Pro ukládání informace do mapy jsou použity klíčové snímky. V následujících kapitolách budou obě metody popsány.

2.1 Kalman Filtr

Kalman filtr je rekurzivní filtr diskretizovaný v časové doméně (neměří kontinuálně, ale jen vzorkuje průběh), který odhaduje stav lineárních dynamických objektů ze série zašumělých měření. Je používán ve velkém množství aplikací, od radaru a počítačového vidění, až po ekonomické problémy.

Kalman filtr modeluje časově diskretizované Markovy řetězce, postavené na lineárních operacích rušených Gausovým šumem (nemodelovanými vlivy). Příkladem může být měření polohy a rychlosti pohybu z hvězdné oblohy popsané v článku [3], kdy měření neprobíhá kontinuálně, ale jednotlivá měření jsou od sebe oddělena určitým časovým úsekem. Stav systému v určitém čase t je reprezentován vektorem reálných čísel např. jak je uvedeno ve vzorci (1).

$$x_t = \begin{pmatrix} p \\ v \end{pmatrix} \quad (1)$$

x je zde vektor reprezentující stav systému, p je poloha objektu a v je například rychlost objektu, jehož stav filtrujeme. Pokud bychom byli schopni kdykoliv naprosto přesně a v dostatečně krátkém čase určit polohu p , aniž by se mezi tím naše poloha změnila, nemuseli bychom Kalman filtr vůbec používat. Problém však nastává v případech, kdy měření polohy není přesné, nebo zjištění aktuální polohy trvá déle, než je doba mezi dvěma měřeními. Tehdy je Kalman filtr ideálním řešením.

Funkce Kalman Filtru

Funkce Kalman filtru se skládá z predikce a následného upřesnění stavu filtrované veličiny. Nový stav je predikován na základě minulého upřesněného stavu. Dále je ještě upřesněn měřením provedeným v čase odpovídajícímu novému stavu, kdy upřesnění respektuje přesnost predikce i provedeného měření. Předpokládá se nepřesnost predikce i měření. Tímto lze dosáhnout přesnějšího výsledku na základě dvou méně přesných vstupů.

První částí je odhad stavu na základě minulého stavu (**Predikce**), která vypočítá přechod z předchozího odhadnutého a upřesněného stavu do následujícího (2).

$$\hat{x}_{t|t-1} = F_t \hat{x}_{t-1|t-1} + B_t u_t \quad (2)$$

Kde $\hat{x}_{t|t-1}$ je odhad stavu v čase t na základě stavu v čase $t-1$ (stav který předcházal před časem t). Funkce F_t je transformační matice mezi stavy. Odhad probíhá na základě minulého upřesněného odhadu stavu $\hat{x}_{t-1|t-1}$. Dále B_t je transformační funkce řízení a u_t je řízení pocházející od uživatele. V tomto případě část vzorce $B_k u_k$ můžeme zanedbat, protože při pohybu kamery v prostoru nemáme žádnou informaci o řízení nositele kamery. V predikční části se počítá i kovariance odhadu jistoty predikce $P_{t|t-1}$ správného odhadu stavu(3).

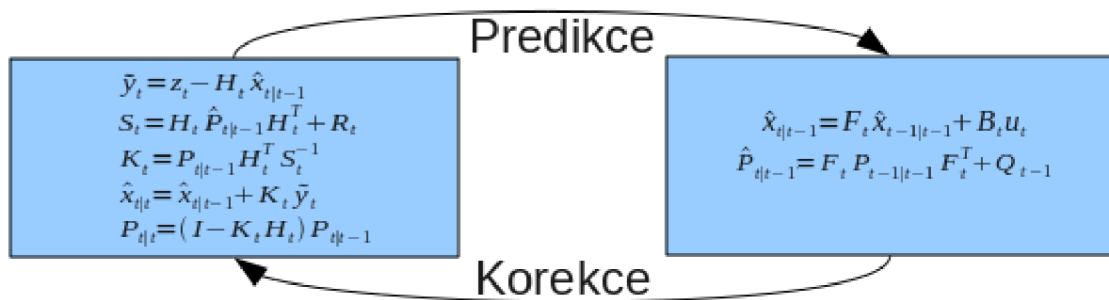
$$\hat{P}_{t|t-1} = F_t P_{t-1|t-1} F_t^T + Q_{t-1} \quad (3)$$

Kde F_t je transformační matice mezi minulým a nynějším odhadem kovariance a Q_{t-1} je kovariance modelující šum v prostředí. Statistický základ potřebný pro pochopení vzorců je výborně popsán v přednášce [4].

Další část filtru realizuje korekci odhadu na základě měření (**Update**). Je popsána těmito rovnicemi (4).

$$\begin{aligned}
 \tilde{y}_t &= z_t - H_t \hat{x}_{t|t-1} \\
 S_t &= H_t \hat{P}_{t|t-1} H_t^T + R_t \\
 K_t &= P_{t|t-1} H_t^T S_t^{-1} \\
 \hat{x}_{t|t} &= \hat{x}_{t|t-1} + K_t \tilde{y}_t \\
 P_{t|t} &= (I - K_t H_t) P_{t|t-1}
 \end{aligned} \tag{4}$$

Kde z_t je hodnota stavu získaná měřením, H_t je transformační matice z prostoru stavů do prostoru měření (tzn. Stav máme popsán ve 3D světě $\mathbb{R}^3 = (x, y, z)$ a měření probíhá ve 2D světě $\mathbb{R}^2 = (u, v)$ v obrázku z kamery), ve kterém je získáváno měření z_t . \tilde{y}_t je pak rozdíl odhadnuté hodnoty a hodnoty naměřené. S_t je takzvaná nejistota predikce v prostoru měření, R_t snímací šum. Například u snímání kamerou je to vzdálenost od středu snímací plochy. S rostoucí vzdáleností od středu roste i nejistota přesnosti snímání. K_t neboli Kalman Gain je váha, se kterou se započítá rozdíl odhadované hodnoty od naměřené do výsledného odhadu stavu $\hat{x}_{t|t}$. Celkový mechanismus Kalman filtru by se dal shrnout následujícím obrázkem (2.1).



Obrázek 2.1: Princip činnosti Kalman filtru.

Z výše popsaného plyne, že pro funkci Kalman filtru není třeba historie všech stavů, ale pro svou činnost potřebuje jen předchozí stav. Z předešlého stavu udělá predikci následujícího. Tím ušetří velké množství výpočtů spojeného s úplným výpočtem stavů z měření. S pomocí měření už jen udělá korekci pro získání větší přesnosti. Princip a příklad je výborně popsán v článku [3].

Odvozené verze Kalman filtru

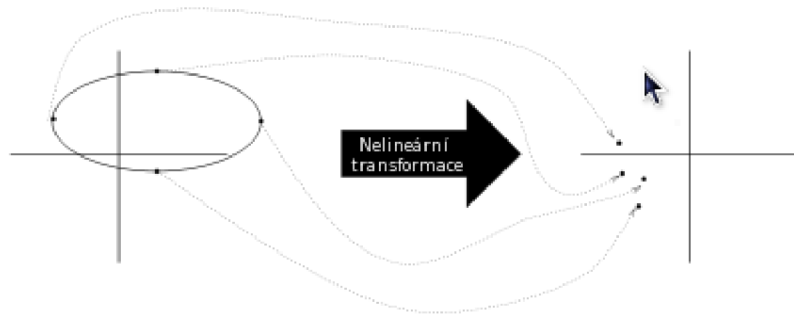
V MonoSLAM metodách není možné použít klasický Kalman filtr, ale používají se z něj odvozené verze jako je EKF a UKF.

EKF (Extend Kalman Filter) je rozšířený Kalman filtr, který je schopný modelovat i procesy, jež není možné modelovat čistě pomocí lineárních funkcí. Tato vlastnost je dosažena díky použití

diferencí. Transformační matice F a H jsou nahrazeny Jacobian Maticí (maticí parciálních derivací), což je ukázáno v (5)

$$\begin{aligned} F_{t-1} &= \left. \frac{\partial f}{\partial x} \right|_{\hat{x}_{t-1|t-1}, u_t} \\ H_t &= \left. \frac{\partial h}{\partial x} \right|_{\hat{x}_{t|t-1}} \end{aligned} \quad (5)$$

Význam tohoto zápisu je aproximace nelineární funkce Taylorovým rozvojem 1. řádu. Dá se říci, že funkce vypočítává tečný povrch k povrchu modelované funkce. To má za následek možnost divergence při velkých změnách. Obdobnou transformací je nutno opatřit i matice modelující šumy prostředí a měření.



Obrázek 2.2: Princip unscented transformace. Obrázek převzat z [5].

UKF (Unscented Kalman filtr) výborně popsán v článku [5]. V této metodě je částečně odstraněna chyba predikce nelineárních funkcí. Základní princip je možné si představit jako modelování pravděpodobnosti více body což je popsáno na obrázku (2.2). Vybrané body nejsou zvoleny náhodně, jako u metod Monte Carlo, ale algoritmicky. Díky tomuto způsobu volby je možné co možná nejlépe popsat dané náhodné rozložení. Body jsou vybírány podle algoritmu popsaného v rovnicích (6). Kde \bar{x} je střední hodnota hodnot x , P_{xx} je aproximována $2n+1$ body,

$$\begin{aligned} X_0 &= \bar{x} & W_0 &= \kappa/(n+\kappa) \\ X_i &= \bar{x} + (\sqrt{(n+\kappa)P_{xx}})_i & W_i &= 1/2(n+\kappa) \\ X_{i+n} &= \bar{x} - (\sqrt{(n+\kappa)P_{xx}})_i & W_{i+n} &= 1/2(n+\kappa) \end{aligned} \quad (6)$$

W_i je váha bodu, $\kappa \in \mathbb{R}$, a $(\sqrt{(n+\kappa)P_{xx}})_i$ je i -tý řádek nebo sloupec z matice $(n+\kappa)P_{xx}$.

2.2 Rohový detektor Shi and Tomasi

Rohový detektor Shi and Tomasi[6], použitý v metodě MonoSLAM, někdy také nazývaný Kanade-Tomasi je z velké části založený na rohovém detektoru Harris[7]. Rohový detektor Harris vychází z rohového detektoru Moravec. Pracuje s výsekem obrazu, který je korelován sám sebou posunutým o

nějaké $\Delta x, \Delta y$. Korelace je navíc váhována Gausiánem, který má počátek ve středu výseku o souřadnicích x, y . Gausián zajistí aby měl střed výseku větší váhu než jeho okolí. Prvotní korelaci je možné najít v článku o detektoru Harris[7]. Po několika úpravách se dospělo ke tvaru popisující auto-korelaci výseku obrazu sama sebou v maticovém tvaru (7)

$$c(x, y) = [\Delta x \ \Delta y] \begin{bmatrix} \sum_w g(x_i, y_i) (I_x(x_i, y_i))^2 & \sum_w g(x_i, y_i) I_y(x_i, y_i) I_x(x_i, y_i) \\ \sum_w g(x_i, y_i) I_y(x_i, y_i) I_x(x_i, y_i) & \sum_w g(x_i, y_i) (I_y(x_i, y_i))^2 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \quad (7)$$

kde W definuje korelovaný výsek obrazu, který indexuje x_i, y_i , $g(x_i, y_i)$ je Gausovo okno se středem v bodě x, y . Dále Δx a Δy jsou posunutí korelovaných výseků obrazů vůči sobě, $I_x(x_i, y_i)$ derivace obrazu v bodě x_i, y_i podle x a $I_y(x_i, y_i)$ derivace obrazu podle y . Celá rovnice se pak dá přepsat do tvaru (8)

$$c(x, y) = [\Delta x \ \Delta y] C(x, y) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}, \quad (8)$$

kde $C(x, y)$ v sobě zahrnuje popis okolí bodu x, y . Matice $C(x, y)$ se dále převede na vlastní čísla λ_1, λ_2 . Vlastní čísla, někdy také nazývané charakteristické kořeny, jsou popsána v článku [8]. Jelikož dostaneme dvě výsledné hodnoty, řešení se rozpadne na čtyři nebo spíše tři případy. V prvním jsou obě hodnoty malé a výsledkem je pravděpodobně rovina, kde je autokorelace malá. V dalším případě, kdy je jedna nebo druhá hodnota výrazně větší, se jedná o detekci hrany. V posledním, kdy jsou obě hodnoty velké, se pak jedná o detekci rohu.

Metoda Shi and Tomasi rohový detektor Harris rozšiřuje o práh, který umožňuje odfiltrovat šum v obraze. Prah je definován pomocí výrazu

$$(\min(\lambda_1, \lambda_2)) > \lambda, \quad (9)$$

kde λ je práh, λ_1 a λ_2 jsou výsledky standardního hranového detektoru Harris.

2.3 KLT Tracker

KLT tracker[6] je algoritmus pro vyhledání korespondujících bodu mezi po sobě jdoucími snímky I a T . Základním předpokladem této metody je nalezení afinní transformace a translace mezi korespondujícími význačnými oblastmi, které jsou navzájem posunuté vlivem jejich pohybu v průběhu času. Hlavní myšlenku celé metody lze popsat jako hledání korespondence dvou oblastí popsaného rovnicí (10)

$$T(Ax + d) = I(x), \quad (10)$$

kde $I(x)$ je intenzita snímků (jasová složka) v bodě $x = [x, y]$, $T(Ax + d)$ je intenzita bodu snímku v T téhož bodu jako ve snímku I , avšak posunutá. Dále pak A je afinní transformace bodů x a d

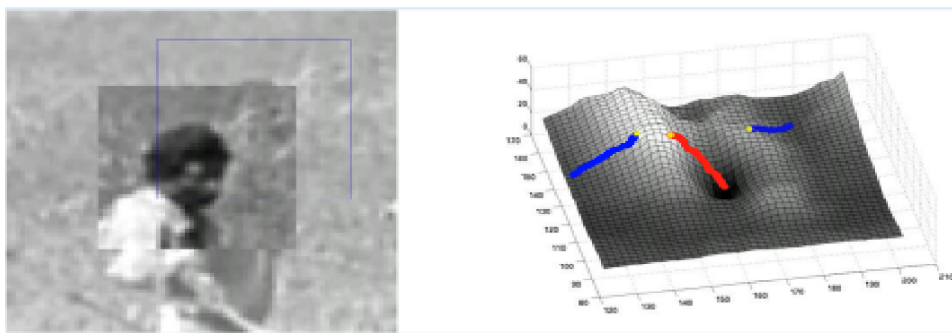
je posunutí bodů (význačných oblastí) vůči sobě. Pro názorné vysvětlení je dobré zbavit se afinní transformace a ukázat funkci trackeru jen jako řešení translace, což je výborně popsáno v této přednášce [9]. Pak se dá rovnice (10) přepsat do tvaru (11).

$$T(\mathbf{x} + \mathbf{d}) = I(\mathbf{x}) \quad (12)$$

Řešení celého problému je pak vyřešeno pomocí **metody nejmenších čtverců** (13)

$$\sum_{\mathbf{x}} [T(\mathbf{x} + \mathbf{d} + \Delta \mathbf{d}) - I(\mathbf{x})]^2, \quad (13)$$

kde $\Delta \mathbf{d}$ je posun ve směru klesajícího gradientu rozdílu obou význačných oblastí. Asi nejlépe je to viditelné na obrázku (2.3). Z něho je také patrný problém konvergence. V případě, kdy se význačné oblasti od sebe příliš vzdálí, nemusí metoda nejmenších čtverců dosáhnout správného výsledku. Je to



Obrázek 2.3: Vysvětluje pohyb hledané význačné oblasti aby byl minimalizován rozdíl oblastí.

Žlutý bod označuje minulou pozici význačné oblasti. Převzato a upraveno z přednášky [9].

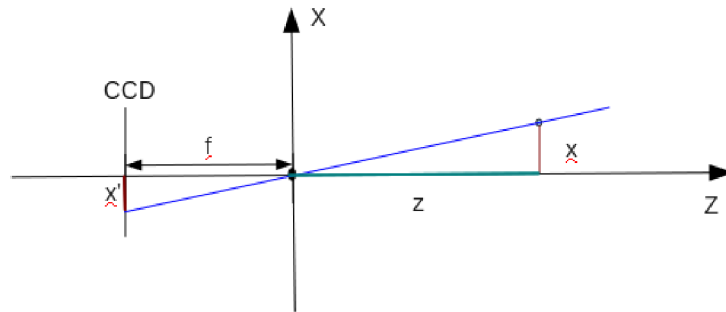
způsobeno aproximací vyhledávací funkce Taylorovým rozvojem, který je oříznutý jen na jeden člen. V případě zajištění kontinuální videosekvence většinou není s konvergencí minimalizační funkce problém.

2.4 Pinhole model kamery

V této podkapitole budou popsány parametry Pinhole modelu kamery. Pro počítání s kamerami je potřeba znát jejich model a parametry. V metodách počítačového vidění se nejčastěji používá Pinhole model. Ten je dostatečně jednoduchý a zároveň schopný popsat dnešní běžně používané kamery. Nejlépe se dá znázornit pomocí obrázku (2.4). Parametry kamery je možno rozdělit na dvě skupiny, vnitřní a vnější. Vnějšími parametry jsou již zmíněná matice R rotace a vektor \vec{t} translace. Vnitřní parametry jsou např. ohnisková vzdálenost, středy zobrazení (x_0, y_0) a rozbíhavost paprsků. Z těchto hodnot je možné poskládat např. projekční matice P definována rovnicí (14)

$$P = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & -af & -bf & x_0 \\ 0 & 0 & -cf & y_0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \vec{t} & R \end{pmatrix} = \begin{pmatrix} 0 & K \\ \vec{t} & R \end{pmatrix} = K \begin{pmatrix} 1 & 0 \\ \vec{t} & R \end{pmatrix}, \quad (14)$$

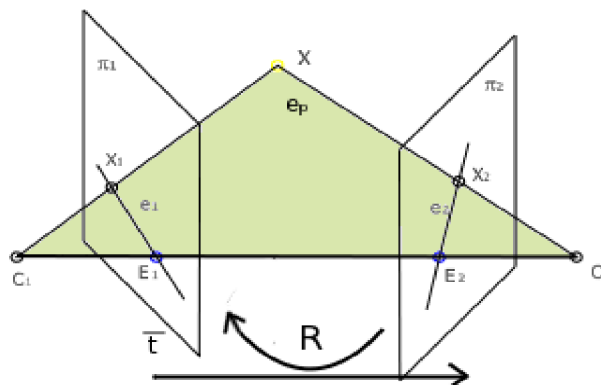
kde f je ohnisková vzdálenost, a, b, c, x_0, y_0 další vnitřní parametry kamery, K je matice 3×3 zahrnující v sobě parametry kalibrace kamery.



Obrázek 2.4: Model Pinhole kamery.

2.5 Epipolární geometrie

V této podkapitole bude popsána Epipolární geometrie [10][11], Fundamentální matice [11] (Fundamental matrix) a Esenciální matice [11] (Essential matrix). Díky Epipolární geometrii je možné vypočítat vzájemnou orientaci dvou snímků a zjišťovat tvar povrchu, který je na obou snímcích zobrazen.



Obrázek 2.5: Základní náznak Epipolární geometrie. Obrázek přebrán z [10] a upraven.

V případě, že je potřeba zjistit vzájemnou polohu dvou snímků určité scény, popřípadě strukturu toho co je na nich zobrazeno, je použití Epipolární geometrie na místě. Základní pojmy týkající se Epipolárního vyhledávání jsou ukázány na obrázku (2.5). Na obrázku jsou vidět π_1 a π_2

průmětny kamer a C_1 a C_2 , což jsou zobrazovací středy kamer (tak je nazývá literatura). Natočení kamer vůči sobě je definováno maticí R 3×3 a vektorem \vec{t} posunutí. U Pinhole modelu kamery jsou středy kamer od průměten ve vzdálenosti rovné ohniskové vzdálenosti f . Body E_1 a E_2 jsou epipoláry, neboli průsečky s průmětnami a čarou (baseline) spojující C_1 a C_2 . Bod X je bod v 3D prostoru, který zobrazují do průmětny π_1 a π_2 průsečky x_1 a x_2 přímkami mezi body C_1 , X a C_2 , X . A nakonec e_p epipolární plocha vymezena epipolárními přímkami e_1 a e_2 . Všechny body uvedené na obrázku musí ležet na epipolární ploše e_p (důležitá podmínka pro řešení Fundamentální a Esenciální matice).

Pro Epipolární geometrii je zvláště významná fundamentální matice F 3×3 prvky, která v sobě obsahuje přepočten bodu, jako např. x_1 ze snímku jedné kamery na epipolární přímce e_1 ve snímku druhé kamery. Vizuální popis je v obrázku (2.5). Po dosazení bodu x_2 do rovnice přímky e_2 musí vyjít nula, protože bod x_2 na přímce e_2 leží. Tento vztah je možné popsat rovnicí (15).

$$x_1^T F x_2 = 0 \quad (15)$$

V tomto bodě se řešení rozpadá na dvě varianty. Jednu, ve které je F známá, a druhou, v níž je F neznámá, ale známe umístění korespondujících bodů (x_1, x_2) . První případ je možno použít pro rekonstrukci povrchu zabraného na obou snímcích. Druhá varianta, kdy se z korespondujících bodů zjišťuje F , je již zajímavější. Nejjednodušší variantou je použití Osmibodového algoritmu (Eight point), popsaného v článku [12]. V článku jsou popsány omezující podmínky, díky kterým není pro řešení soustavy rovnic o 9 neznámých potřeba všech 9 párů korespondujících bodů, ale jen 8. Fundamentální matici je možno použít ještě při řešení vzájemné polohy pomocí 6 a 7 bodů.

Je-li potřeba určit vzájemnou polohu dvou snímků z méně než z 6 bodů, je nutné přejít na Esenciální matici, aby byla zachována rozumná rychlost a přesnost výpočtu. Esenciální matice v sobě nezahrnuje kalibrační parametry kamery. To znamená, že kalibrace K_1, K_2 kamer, které udělaly konkrétní dvojici snímků, musí být známy. Aby toto bylo možné realizovat, je nutné přejít (jak ukazuje článek [10]) na normalizované souřadnice \hat{x} , které v sobě parametry kalibrace již obsahují (16).

$$\hat{x} = K^{-1} x, \quad (16)$$

S použitím normalizovaných souřadnic je pak možné napsat vztah, který musí splňovat výpočet Esenciální matice (17).

$$\hat{x}_1^T E \hat{x}_2 = 0 \quad (17)$$

Esenciální matice je méně obecná než Fundamentální matice, ale díky jejímu použití lze vyřešit vzájemnou polohu dvou snímků jen za pomoci 5 známých dvojic korespondujících bodů za použití algoritmu popsanému v článku [13].

Výpočtem Esenciální matice je získána jen matice, pomocí níž lze ověřit, zda normalizovaný bod z jednoho snímku odpovídá druhému normalizovanému bodu v druhém snímku. Pro účel zjištění pohybu kamery toto nestačí. Je nutné navíc umět Esenciální i Fundamentální matici rozložit. V následujících kapitolách je používána hlavně Esenciální matice, proto bude popsán rozklad hlavně u ní. Epipolární geometrie dokáže určit jen relativní polohu dvou snímků, proto je bez ztráty obecnosti možné použít předpoklad pro projekční matici v níž translace a rotace první kamery takto $P=K[I \ 0]$. Díky tomuto předpokladu se rozklad Esenciální matice změní z rozkladu dvou matic rotace a dvou matic translace na rozklad na jednu matici translace a jednu matici rotace. Nyní je možné Esenciální matici zapsat takto (18)

$$E=SR, \quad (18)$$

kde matice R je matice rotace a matice S je šikmo souměrná matice obsahující informaci o translaci snímků. Dále je předpokládán SVD (Singular value decomposition) rozklad Esenciální matice E ve formě $SVD(E)=Udiag(1,1,0)V^T$. Při ignorování znamének je pak možné, jak ukazuje kniha [14] strana 240, dojít ke dvěma následujícím rozkladům (19)

$$S=UZU^T R=UWV^T \text{ nebo } S=UZU^T R=UW^T V^T, \quad (19)$$

kde (20)

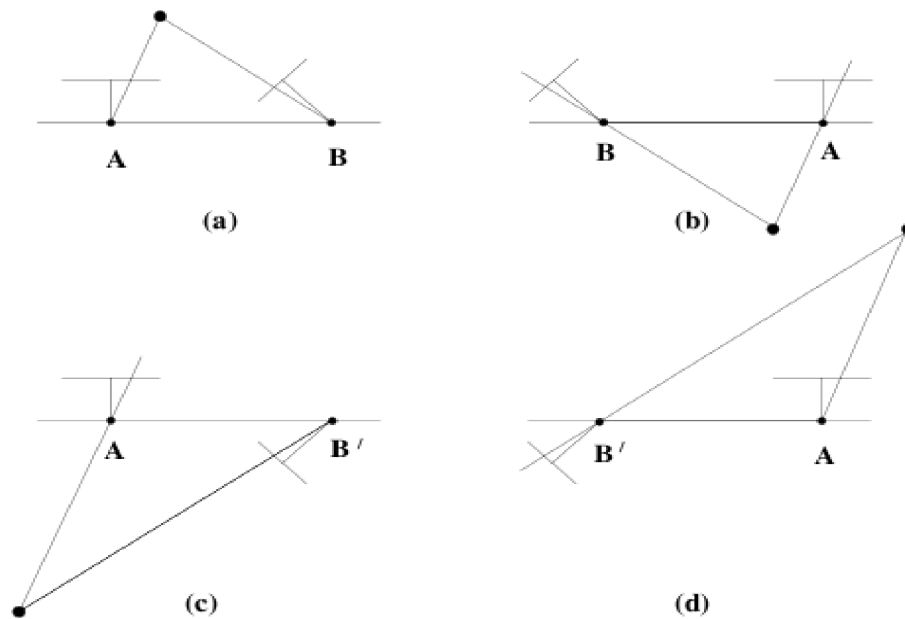
$$W=\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad a \quad Z=\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (20)$$

Jako poslední je nutné získat vektor translace \vec{t} , který je možné nalézt jako třetí sloupec matice U . Bohužel z matic není možné zjistit znaménka \vec{t} ani matice R . Z tohoto důvodu je nutné ověřit čtyři různá řešení. Ta jsou uvedena na obrázku (2.6).

Pro ověření, které ze čtyř řešení je správné, stačí do jedné ze čtyř dvojic projekčních matic P a P' dosadit (21) a ověřit zda oba body leží před objektivem kamery.

$$P_1=K1[I \ 0] \\ P_2=K_2[UWV^T \ u3], \quad K_2[UWV^T \ -u3], \quad K_2[UW^T V^T \ u3], \quad K_2[UW^T V^T \ -u3] \quad (21)$$

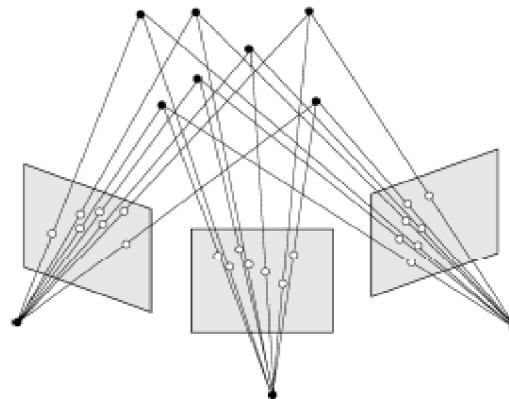
Po vypočítání projekčních matic je možné vypočítat při znalosti vzájemné polohy i strukturu povrchu, který je na obou snímcích zobrazen. Při výpočtu povrchu je nutné předpokládat nepřesnost výsledků způsobenou chybami ve zjišťování polohy význačných oblastí a vlastností zpětného promítání, kdy zobrazený pixel na snímku reprezentuje rozbíhající se kužel v prostoru. Samotný výpočet je možné zjednodušit převedením 3D výpočtu na 2D, a to zanedbáním osy Y , protože výpočet transformace mezi snímky předpokládá polohu obou bodů v jedné epipolární rovině. Jak je tomu na obrázku (2.5).



Obrázek 2.6: Čtyři možná řešení extrakce matice R a vektoru \vec{t} . Převzato z knihy [14].

2.6 Bundle Adjustment

Jedná se o globální metodu hledající parametry snímacích zařízení a objektů ve scéně, která může zpracovávat celou scénu jako celek, tak i po částech. Je založena na vyhledání relativní pozice dvou a více snímků a zároveň 3D pozice význačných oblastí zobrazených na snímcích. Metoda je popsána v článku [15], konkrétně se jedná o metodu Sparse Bundle Adjustment.



Obrázek 2.7: Popis činnosti Bundle Adjustment. Přejato z [15].

Metodu Bundle Adjustment je možné zjednodušeně popsat rovnicí (22)

$$\min_{\hat{P}_j, \hat{X}_i} \sum_{j=1}^m \sum_{i=1}^n d(x_{ij}, P_j X_i)^2, \quad (22)$$

kde x_{ij} je pozice bodu s indexem i ve snímku s indexem j . P_j je projekční matice vnějších i vnitřních parametrů kamery příslušící kameře $P=K[R|t]$ (vysvětlení tohoto zápisu je v kapitole (2.4)), která snímala snímek j . X_i je vektor popisující relativní polohu 3D význačné oblasti, která odpovídá bodu x_{ij} ve snímku. $d(x_1, x_2)$ značí Euklidovu vzdálenost mezi body x_1, x_2 .

Metoda je založena na minimalizaci chyby zpětného zobrazení význačné oblasti do snímku. Což je znázorněno na obrázku (2.6). Zpočátku metoda nemá informaci o poloze snímků ani 3D umístění význačných oblastí. Proto se pro minimalizaci chyby musí optimalizovat jak projekční matice kamer příslušících k jednotlivým snímkům P_j , tak i 3D pozice význačných oblastí na snímcích zobrazených X_i . Řešení Bundle Adjustment vede na nelineární minimalizaci chyby, zároveň je nutné předpokládat chybu měření (šum ve snímcích), a je potřeba zajistit správnou konvergenci. Z těchto důvodů je vhodné pro minimalizaci chyby zvolit nelineární verzi metody nejmenších čtverců Levenberg-Marquardt.

Metoda Levenberg-Marquardt vychází z Gauss-Newton metody. Obě metody jsou do detailů popsány v článku [15]. Minimalizuje chybu $\epsilon = x - \hat{x} = x - f(p)$ mezi x hodnotou naměřenou a \hat{x} hodnotou odhadnutou. Dále $f(p)$ je funkce parametrů p , která popisuje projekci 3D oblastí do snímku kamery. Minimalizace probíhá v několika iteracích. Metoda Levenberg-Marquardt je často pomalejší než Gauss-Newton metoda, ale zato stabilnější. Je to způsobeno přidáním tlumícího parametru μ do Gauss-Newton metody. Celou metodu je pak možné napsat pomocí rovnice (23)

$$J^T J + \mu I \delta_p = J^T \epsilon, \quad (23)$$

kde J je jakobián matice funkce $f(p)$. Definující gradient povrchu funkce $f(p)$. δ_p je krok iterace a I je jednotková matice. Tlumící konstanta μ je měněna v závislosti na úspěšnosti redukce chyby ϵ . Jestliže se iterací dochází k lepším výsledkům, konstanta μ je zmenšována. Naopak v případě zvětšování chyby se konstanta μ zvětšuje a dochází tak ke tlumení rychlosti divergence. A metoda může opět začít konvergovat.

2.7 Výpočet ohniskové vzdálenosti v pixelech

V níže popsaných metodách se často počítá s ohniskovou vzdáleností zadanou v pixelech. Proto je dobré ji umět vypočítat. Ohnisková vzdálenost v pixelech je dána pomocí vzorce (24)

$$FlIP = v_x * Fl / CCD_x, \quad (24)$$

kde v_x je rozměr snímku přes souřadnici x , Fl je ohnisková vzdálenost kamery v mm a CCD_x je rozměr CCD čipu přes souřadnici x .

3 Referenční offline metody

V této kapitole budou popsány využívané offline aplikace a standardní přístupy offline metod, které jsou v této práci použity jen jako reference pro srovnávání přesnosti obou metod pracujících v reálném čase. Offline metody mají vůči metodám pracujících v reálném čase možnost nahlédnout i na snímky, které jsou v budoucnosti právě zpracovávaného snímku. U konkrétních aplikací se nejedná o podrobnosti, protože jsou uzavřené a v dokumentaci neuvádí používané metody, a jen občas je možné určit jakou metodu používají.

Aplikace SynthEyes je vyrobena firmou *Andersson Technologies LLC*. Umožňuje trasovat kameru případně i pohybující se objekty. Trasování je realizováno vícevláknově. V případě složitého pohybu, který automatický režim nezvládne, je možné použít trasování, kdy uživatel dohlíží na to, jak trasovací metoda funguje. Pro urychlení řešení průběhu kamery je zde metodě umožněno přeskakovat skupiny snímků, na kterých se nacházejí redundantní informace. Metoda je schopna během výpočtu korigovat měnící se zoom kamery.

Jako další reference byla použita aplikace voodoo, která je vyvíjena *Laboratorium für Informationstechnologie*. O této aplikaci je již možno získat více informací. Na detekci významných oblastí lze využít skupinu detektorů (Harris, DlineCorner, Susan, Foerstner a SIFT). Pro zjištění korespondence mezi body je možné použít Cross Correlation, KLT Tracking (2.3) a SIFT Matching. O metodě určení pozice kamery není na stránkách výrobce uvedena informace. Pro zpřesnění výsledku metody, která detekuje polohu kamery je použit, stejně jako u metody PTAM, Bundle Adjustment.

Je možné říci, že základní myšlenky offline metod jsou použity v metodách pracujících v reálném čase, ale offline metody mají navíc možnost nahlížet na snímky následující za právě zpracovávaným snímkem. Při offline zpracování videosekvence lze užít náročnější a přesnější algoritmy, a tím zpřesnit celý postup vyhodnocování pohybu kamery.

3.1 Vyhledání význačných oblastí

Pro offline metody připadá v úvahu využití několika přístupů pro vyhledání význačných oblastí. V dřívějších algoritmech se často jednalo o bodové rohové oblasti. Avšak použití jednotlivých význačných bodů s sebou nese problém nalezení správné korespondence mezi body v po sobě jdoucích snímcích. A proto se postupem času začaly používat větší význačné oblasti např. (11x11) pixelu, které je možné výrazně lépe znovu detekovat. Často používaným detektorem význačných oblastí je Shi and Tomasi, popsáný v kapitole (2.2), či metoda FAST[16]. Někdy bývá používán SIFT, nebo popřípadě SUSAN[17].

3.2 Trasování význačných oblastí

V této podkapitole budou popsány metody, které jsou v offline metodách využívány za účelem trasování na-detekovaných význačných oblastí, neboli hledání korespondence mezi význačnými oblastmi dvou po sobě jdoucích snímků. Jinak řečeno význačná oblast nalezená v jednom snímku se nalezne v následujících snímcích. Jak již bylo uvedeno v předchozí podkapitole, je výhodnější používat větší význačné oblasti, které v sobě obsahují více informace o oblasti. Větší množství informací umožňuje spolehlivěji určit, zda jde opravdu o hledanou oblast či nikoliv, a tím předejít zbytečným chybám zavlečených do výpočtu.

Pro samotné trasování se v offline metodách často používá KLT trasovač popsaný v podkapitole (2.3). Díky němu je jednoduché nalézt korespondenci mezi body s velkou přesností. KLT se převážně používá pro videosekvence s těsně navazujícími snímky. Jestliže je předpoklad velkých výpadků snímků, je potřeba použít jiné metody, jako například RANSAC. Ta vyhledává korespondenci mezi body hrubou silou. Při testování kontroluje zda vybrané body splňují některou společnou transformaci. V případě nalezení většinové transformace skončí svou práci. Pro kontrolu transformace se často využívá výpočet matice Homografie, ve kterém se převedou i ostatní význačné oblasti ve snímku a zkontroluje se velikost chyby transformace nalezené metodou RANSAC.

Další možností, která je spíše bližší níže popsaným metodám pracujících v reálném čase, je použití Kalman Filtru. U offline metod [18] se často používá UKF verze Kalmanova filtru popsaná (2.1), nebo jeho mutované verze. UKF je na rozdíl od EKF stabilnější. Kalman filtr, ať jde o jakoukoliv verzi, je určen k filtrování pohybu kamery a předpovídání polohy kamery v dalším snímku. Díky předpovědi pozice kamery je možné předpovídat i polohy význačných oblastí v následujícím snímku, a tím značně urychlit a zpřesnit jejich vyhledávání. O tomto principu je více uvedeno v kapitole (4.4).

3.3 Počítání 3D pohybu kamery

V této podkapitole budou popsány metody pro výpočet 3D transformace mezi dvěma snímky kamery. Výpočet 3D polohy s výpočtem 2D polohy význačných oblastí často velice blízce souvisí. Pokud algoritmus pro výpočet 3D polohy kamery dostane chybnou informaci z metody pro výpočet korespondence 2D významných oblastí, může chybně určit 3D transformaci mezi snímky. Z tohoto důvodu je nutné zajistit co největší přesnost metod pro trasování význačných oblastí. Popřípadě je možné do metod počítajících 3D transformaci dodat funkci pro detekci chybně trasovaných význačných oblastí. Detekce chybného trasování může být realizována pomocí rozdělení trasovaných bodů do různých skupin, nad kterými se realizuje detekce 3D transformace. V případě neshody určité

3D transformace s ostatními, může být vyloučena i s význačnými oblastmi, které patří jen do ní. Tuto variantu přístupu je možné použít jen při dostatku času a význačných oblastí.

Samotný výpočet 3D transformace mezi snímky může být realizován pomocí rozkladu Homografní matice, či rozkladu Esenciální (Fundamentální) matice[19] na jednotlivé složky pohybu kamery. Matice jsou rozkládány na R složku rotace a \vec{t} translační složku. Konkrétně je možné použít pětibodový algoritmus založený na Epipolární geometrii, která je popsána v kapitole (2.5). Jestliže je rekonstrukce pohybu kamery užívána na videosekvenci po sobě jdoucích snímků, je možné použít urychlení v podobě odhadu budoucí pozice kamery. Tento odhad lze realizovat pomocí již zmiňovaného(2.1) Kalman filtru (u offline metod spíš UKF), nebo jiného filtru např. Alfa beta, etc.. .

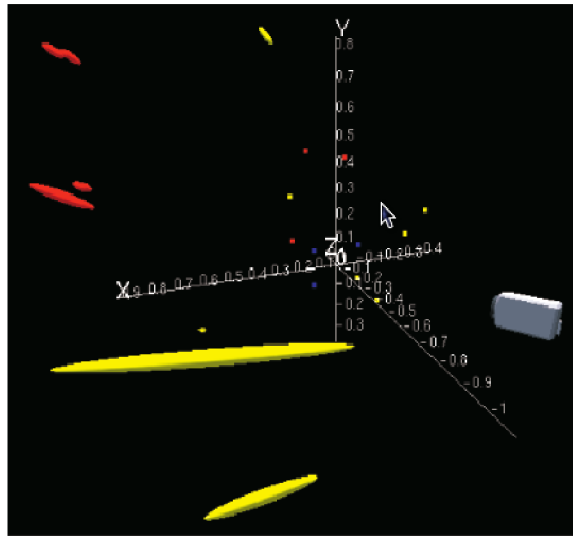
3.4 Přechod od offline k real-time metodám

V této podkapitole budou stručně popsány rozdíly mezi offline a real-time metodami, které umožňují real-time metodám běžet v reálném čase. Real-time metody jsou ve výsledku dostatečně rychlé offline metody, avšak nemohou nahlížet do budoucnosti (na budoucí snímky) za právě zpracovávaný snímek. Také používají význačné oblasti. Trasování je urychleno odhadem pohybu pomocí vhodného filtru (Kalman, alfa beta, decaying velocity). Pozice význačných oblastí jsou předem předpočítány, a pak už je jejich pozice jen mírně upřesněna a dohledána. Po získání přesné pozice korespondujících bodů se odhadnutá pozice použije i pro urychlení dopočítání 3D pozice kamery. Oproti offline metodám si real-time metody nemohou dovolit neefektivně hospodařit s nashromážděnými daty o okolí. Z tohoto důvodu je nutné zabránit přehlcení daty volbou různých metod optimalizace, popsaných v následujících kapitolách.

Přechod od offline metod k real-time metodám není příliš velký technologický skok, spíše se jedná o optimalizaci určitých výpočetně náročných problémů.

4 MonoSLAM

Základní koncept této metody je založen na pravděpodobnostním mapování význačných částí obrazu, reprezentovaných stavu kamery, a všech význačných částí obrazu. V modelu je zahrnuta i nejistota měření a snímání. Inicializace je provedena na začátku a při činnosti průběžně upravována pomocí EKF (Extended Kalman Filtru). V případě nalezení nového význačného bodu, je pravděpodobnostní mapa rozšířena i o nový význačný bod. Protože je nutné pracovat v reálném čase, je nutné z modelu význačné body i mazat. Pro mazání jsou vhodné body, které již nejsou potřeba, nebo je jejich kvalita příliš malá. Základní představu si je možné udělat z obrázku (4.1).



Obrázek 4.1: Stav pravděpodobnostní 3D mapy, ukazující pozici kamery a význačných bodů popsaných elipsami nejistoty, které jsou popsány kovarianční maticí. Převzato z [1].

4.1 Pravděpodobnostní 3D Mapa

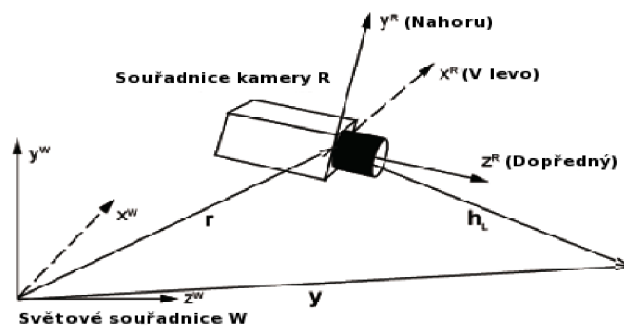
Matematicky je mapa reprezentovaná pomocí stavového vektoru \hat{x} a multikovarianční maticí P . Stavový vektor v sobě obsahuje současný stav (polohu a natočení) kamery i stavy význačných bodů. Mapa je popsána těmito maticemi (25).

$$\hat{x} = \begin{pmatrix} \hat{x}_0 \\ \hat{y}_1 \\ \hat{y}_2 \\ \vdots \end{pmatrix}, P = \begin{bmatrix} P_{xx} & P_{xy_1} & P_{xy_2} & \cdots \\ P_{y_1x} & P_{y_1y_1} & P_{y_1y_2} & \cdots \\ P_{y_2x} & P_{y_2y_1} & P_{y_2y_2} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (25)$$

Kde \hat{x}_o je stav kamery a \hat{y}_i je příznak popisující 3D polohu a vlastnosti význačného bodu. Přesněji je stav kamery \hat{x}_o popsán následujícím vektorem (26)

$$\hat{x}_o = \begin{pmatrix} r^W \\ q^{WR} \\ v^W \\ \omega^R \end{pmatrix} \quad (26)$$

Kde r^W je vektor pozice kamery v souřadnicovém systému světa obr.(4.2), q^{RW} quaternion popisující natočení kamery, v^W vektor rychlosti kamery a ω^R vektor úhlové rychlosti.



Obrázek 4.2: Orientace kamery ve světových souřadnicích. Převzato z [1].

Hlavním důvodem pro tvoření mapy není jen popis prostoru, ale potřeba umožnit pohyb kamery v reálném čase. Proto je mapa zaměřena na ukládání značek s vysokou kvalitou. Aby mapování bylo úspěšné je nutné splnit předpoklad, že mapovaný prostor je mapován jako pevné těleso. To znamená, že mapa nepředpokládá pohyb namapovaných význačných oblastí. I kamera je modelovaná jako pevné těleso. Význačné body v mapě jsou zaznamenávány pomocí elips reprezentujících hranice nejistoty, které jsou při záběru z jiného úhlu upřesňovány. Zjistit relativní polohu vůči nasnímaným bodům není problém, ale zjištění polohy vůči relativnímu světovému souřadnému systému už ano. Nejistota umístění kamery vzhledem ke relativnímu světovému souřadnému systému, je popsána prvky matice P , které jsou mimo diagonálu. Náročnost zpracovávaného algoritmu je $O(\ln^2)$, což znamená, že počet částic může být rámcově okolo 100, v případě práce s kamerou, která snímá na 30Hz. Z tohoto důvodu byl v této metodě zvolen jednoduchý EKF popsáný v (2.1).

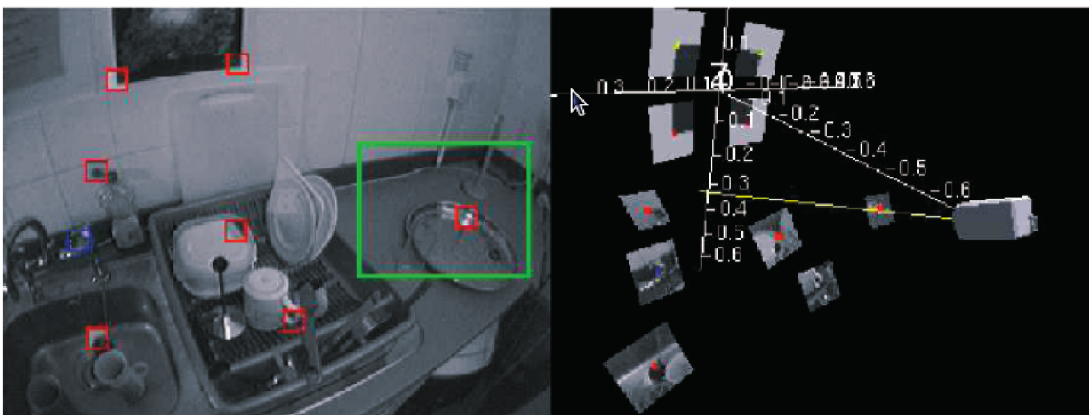
4.2 Význačné oblasti

Na rozdíl od klasických metod, kdy se jako význačné body používaly jen prosté rohy, tato metoda používá podle vzoru Davidsona a Murrya [20] [21] relativně velké oblasti (11x11 pixelu). Velké

oblasti v sobě totiž obsahují mnohem více unikátních informací. Popis oblastí je realizován pomocí vektoru (27)

$$y_{pi} = \begin{pmatrix} r_i^W \\ \hat{h}_i^W \end{pmatrix}, \quad (27)$$

kde r_i^W je pozice v relativních světových souřadnicích a \hat{h}_i^W je jednotkový vektor směrem ke kameře (normálový vektor snímané plošky). Názorná ukázka je zobrazena na obrázku (4.3). Nejprve se ve snímku náhodně určí obdélník, ve kterém se bude význačná oblast hledat. V obrázku (4.3) je tato náhodně zvolená oblast znázorněná zeleným obdélníkem. A dále jsou v oblasti nalezeny význačné oblasti pomocí detekčního operátoru Shi and Tomasi [22]. Při velkém pohybu kamery se význačné oblasti mohou výrazně měnit, a proto je nutné význačné oblasti průběžně nahrazovat novými záběry těchto bodů. V určitých případech je tato metoda výborná, v ostatních může být spíše problematická. Pro lepší lokalizaci oblastí je navíc použita informace o poloze kamery, což zejména urychluje přiřazování odpovídajících oblastí z pravděpodobnostní mapy. Díky uloženým informacím o bodu, jako je poloha r_i^W , orientace \hat{h}_i^W a odhadu polohy kamery \hat{x}_o , je možné udělat projekci význačné oblasti do opravdového snímku, a tam provést jednoduchý Template matching. Při zabírání význačných oblastí s hodně rozdílných úhlů je nutné použít afinní transformaci pro zpracování templatu (vzorů uložených význačných oblastí).

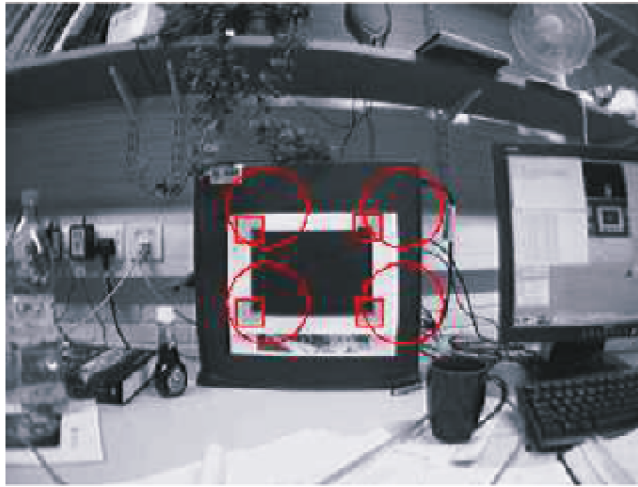


Obrázek 4.3: Ukázka reprezentace význačných bodů v modelu. Převzato z [1].

4.3 Inicializace systému

Ve většině případů metoda při prvním spuštění nemá informaci o struktuře vnějšího světa. Proto to, co uvidí jako první, nastaví jako počátek souřadnicového systému. V případě použití SLAMu s jednou kamerou je inicializace obtížná, proto je v této metodě použita skupina význačných bodů, o

kterých známe jejich vzdálenost a velikost. Nejčastěji se používají 4 význačné body, což popisuje obrázek (4.4).



Obrázek 4.4: Inicializace systému. Převzato z [1].

Čtyři význačné oblasti s přesnými rozestupy jsou voleny z těchto důvodů:

1. V MonoSLAM není možné zjistit vzdálenost význačných oblastí od sebe a od kamery, protože z jednoho snímku nelze přesně vyčíst informaci o hloubce. Použití význačných oblastí navíc umožňuje precizně určit měřítko prostředí.
2. Jestliže máme nějakou známou informaci o prostředí, hned od začátku je možné přistoupit k normálnímu algoritmu.

4.4 Modelování pohybu a jeho predikce

V této metodě je použit EKF (Extended Kalman Filtr), proto se výpočet pohybu skládá ze dvou částí:

1. Predikce pohybu kamery mezi snímky.
2. Zpřesnění odhadu pozice kamery z rozložení význačných bodů.

Protože metoda používá pro vstup jen informace z obrazu, nemáme informaci o činnosti nositele kamery „ušel jsem 1m“. Při předvídání pozice kamery jsou brány v úvahu jen průměrná rychlost a průměrná úhlová rychlost. Proto není příliš vhodné používat tuto metodu pro pohyby s rychlou změnou akcelerace, nebo ve videosekvencích s možností velkého výpadku snímků.

V tomto modelu je pro každý časový krok předpokládána neznámá průměrná akcelerace a^W a úhlové zrychlení α^R což vyjadřuje vzorec (28).

$$n = \begin{pmatrix} V^W \\ \Omega^R \end{pmatrix} = \begin{pmatrix} a^W \Delta t \\ \alpha^R \Delta t \end{pmatrix} \quad (28)$$

Kde V^W je šumová rychlost a Ω^R je šumová úhlová rychlost. Tyto rychlosti mohou vzniknout například při nárazu do nositele kamery, který není plánovaný a započítaný v modelu. Protože se

jedná o Kalman filtr, další důležitou součástí je predikční funkce, díky které je schopen Kalman filtr předpovídat polohu v následujícím kroku. Transformační funkce je popsána ve vzorci (29)

$$f_v = \begin{pmatrix} r_{new}^W \\ q_{new}^{WR} \\ v_{new}^W \\ \omega_{new}^R \end{pmatrix} = \begin{pmatrix} r^W + (v^W + V^W) \Delta t \\ q^{WR} + q((\omega^R + \Omega^R) \Delta t) \\ v^W + V^W \\ \omega^R + \Omega^R \end{pmatrix}. \quad (29)$$

Kde všechny proměnné znamenají to, co ve vzorci (26).

Protože metoda pracuje s EKF filtrem, je nutné vypočítat matici nejistoty odhadu vlivem šumu prostředí Q_v . Šum vypočítáme pomocí vzorce přes Jacobian matici (30)

$$Q_v = \frac{\partial f_v}{\partial n} P_n \frac{\partial f_v^T}{\partial n}, \quad (30)$$

kde P_n je kovarianční matice šumového vektoru n . Pro EKF filtr je nutné spočítat ještě Jacobian matici $\frac{\partial f_v}{\partial x_v}$, která je potřeba pro predikci pozice v následujícím časovém okamžiku (kroku).

V případě, že se P_n zvolí malé, je předpokládán plynulý pohyb, kdy má předpověď polohy velký význam. V případě, že P_n bude nabývat velkých hodnot, budeme předpokládat velké změny rychlostí V^W a Ω^R , a předpovědi tedy nebude dána taková váha.

4.5 Aktivní měření význačných bodů a upravování Mapy

Klíčovou částí této metody je predikce polohy kamery, a tím předpovězení polohy význačných oblastí. Jelikož známe oblast, kde se význačné oblasti nacházejí (velikost oblasti je určena multikovarianční maticí nejistoty P) a o jaké konkrétní částice jde, tato předpověď nám ušetří mnoho výpočetního výkonu. Vyhneme se výpočtu homografní matice a např. metody RANSAC, která se používá pro zjištění korespondence bodů.

V první řadě je učiněn odhad stavu kamery \hat{x}_v a relativní pozice význačných oblastí \hat{y}_i vůči kameře $h_L^R = (x, y, z)$. Toto je popsáno rovnicí (31)

$$h_L^R = R^{RW} (y_i^W - r^W). \quad (31)$$

Kde R^{RW} je matice rotace vytvořená z quaternionu v odhadnutém stavu \hat{x}_v a r^W je pozice kamery.

Dále je nutné udělat zpětnou projekci význačné oblasti do 2D prostoru záběru na souřadnice (u, v) (snímku sejmutého kamerou). Tato perspektivní transformace je popsána rovnicí (32)

$$h_i = \begin{pmatrix} u_i \\ v_i \end{pmatrix} = \begin{pmatrix} u_0 - fk_u \frac{h_{L_x}^R}{h_{L_z}^R} \\ v_0 - fk_v \frac{h_{L_y}^R}{h_{L_z}^R} \end{pmatrix}, \quad (32)$$

kde f je ohnisková vzdálenost, k_u, k_v je $1/\text{pixelsize}$. u_0, v_0 jsou pozice středu snímku. V případě použití širokouhlé kamery je nutné použít další transformaci, která je popsána v článku [1] rovnice 8-10.

Jakobián projekční matice (nebo dvoustupňové projekční matice v případě širokouhlé kamery) umožňuje spočítat nejistotu predikce pozice význačných bodů. Jakobián bere v úvahu jak pozici kamery, tak i pozici samotných částic. Výsledná matice je reprezentována jako symetrická 2×2 inovativní kovarianční matice S_i , jež je popsána vzorcem (33)

$$S_i = \frac{\partial U_i}{\partial x_0} P_{xx} \frac{\partial U_i^T}{\partial x_0} + \frac{\partial U_i}{\partial x_0} P_{xy} \frac{\partial U_i^T}{\partial y_i} + \frac{\partial U_i}{\partial y_i} P_{yx} \frac{\partial U_i^T}{\partial x_0} + \frac{\partial U_i}{\partial y_i} P_{yy} \frac{\partial U_i^T}{\partial y_i} + R, \quad (33)$$

kde R je konstantní šum popisující šum snímání. $U_i = (u_i \ v_i)$ je souřadnice význačné oblasti vzhledem ke středu snímku odvozená od h_i s připočítanou rozbíhavostí paprsku. Velikost R je tím větší, čím větší je vzdálenost od středu kamery. Znalost S_i umožňuje definovat eliptické vyhledávací okno, ve kterém se částice má hledat v reálném obraze. V případě, že v zorném poli je více význačných oblastí, je výhodnější vybírat oblasti, které mají velké S_i . Velké S_i znamená velkou náročnost na vyhledávání, ale zato obsahuje největší množství informace o poloze atd... Teorie o množství informace je popsána v článku [23]. Protože však hledání význačných oblastí s velkou nejistotou je časově náročné, je dobré omezit počet hledaných částic jen na deset až dvanáct.

4.6 Inicializace význačných oblastí

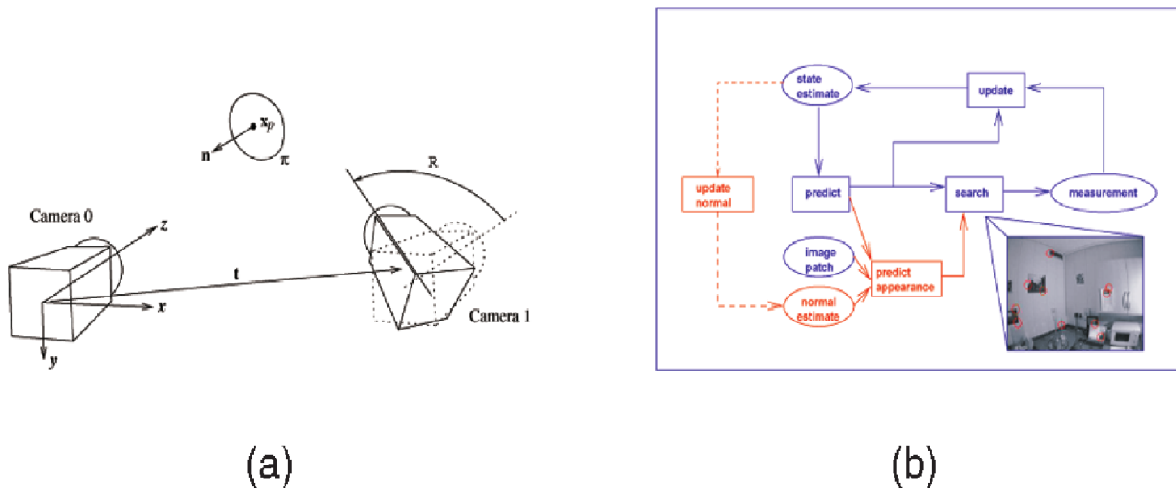
V případě použití jedné kamery není možné z vyhledané význačné oblasti zjistit informaci o hloubce. Proto se oblast nejprve uloží do mapy jako čára, procházející význačnou oblastí s jedním koncem v kameře a druhým koncem v nekonečnu, která se nepoužívá pro měření pozice. A v případě posunu kamery a zabrání význačné oblasti z jiného úhlu se pomocí triangulace (pomocí epipolárních linií) zjistí hloubka, ve které se oblast nachází a vytvoří se z ní standardní oblast v mapě oblastí. Význačná oblast v mapě má pak zápis popsáný rovnicí (27). Vyhledávání opravdové hloubky význačného bodu je jen zobecněním eliptického vyhledávání, kdy se po čáře rozprostrou elipsy a vybere se nejpravděpodobnější hloubka. Což je podrobně popsáno [1] část 3.6.

4.7 Správa Mapy

Správa mapy je jedna z nejdůležitějších částí algoritmu, protože nesmíme připustit přetížení procesoru. V případě, že je v záběru více než 12 význačných oblastí, může dojít k přetížení procesoru, a tím nestíhání snímkové frekvence kamery. V případě širokouhlé kamery je tento problém ještě horší, jelikož je zabírán větší prostor a tím více informací.

Viditelnost částic se počítá relativně vůči otočení a poloze kamery. Nové význačné oblasti se do mapy přidávají jen v případě, že počet viditelných oblastí klesne pod určitou hranici. Význačné oblasti jsou detekovány pomocí *Shi and Tomasi* operátoru, který prozkoumává náhodně umístěné okno o velikosti 80x60 pixelů.

Význačné oblasti jsou mazány za předpokladu, že více jak v 50% případech měření jsou špatné. Toto nastává v případě nalezení např. T-junctions, nebo pohybujících se objektů. Navíc jednou za čas proběhne statická kontrola mapy, která z mapy odstraní nevyhovující body.



Obrázek 4.5: (a) Pohled kamer ze směrů na oblast popsanou normálou (b) cyklus vyhodnocení 3D orientace planární význačné oblasti. Převzato z [1].

4.8 Orientace významných oblastí

Pomocí predikce pozice kamery a normálového vektoru oblasti (ten udává směr, ze kterého byla oblast poprvé detekována), je možné určit orientaci Význačné oblastí vůči kameře obr.(4.5). Díky tomu je možné pomocí homografní matice (34) jednoduše transformovat oblast, tak aby odpovídala korespondující oblasti sejmuté ve snímku z kamery.

$$H = CR[n^T x_p I - t n^T] C^{-1} \quad (34)$$

Kde C je kalibrační matice popisující perspektivní projekci, nebo aproximace perspektivní projekce, R je matice rotace kamery, t translace kamery, n normála k význačné oblasti a x_p pozice kamery.

(I je matice identity 3×3 . Díky tomuto je možné použít template matching vyhledávání v eliptické oblasti.

4.9 Souhrn metody

Metoda je založena na pravděpodobnostním vyhodnocování polohy. Prostor je popsán stavem a pravděpodobnostní mapou rovnice (25) (26), kde jsou uloženy odhady a kovariance umístění významných oblastí i kamery snímající celý prostor. Aby byla metoda dostatečně rychlá, přesná a mohla pracovat s nelineárními daty, používá EKF (extended kalman filtr) pro odhad pohybu kamery. Díky odhadu pohybu kamery může, bez velkých nároků na výpočetní výkon, najít a určit význačné oblasti v zorném poli kamery (viz. vzorce (31)-(33)), a tím si ušetřit velké množství výpočtů s používáním např. RANSAC (metoda pro se-souhlasení odpovídajících si bodů). Po rychlém nalezení a ověření polohy význačných oblastí se pomocí EKF zkoriguje i odhad polohy kamery. Korekce polohy kamery se počítá přímo z odchylky polohy význačných oblastí, přes transformační matici H a K , kde jsou spočítány difference polohy kamery v závislosti na chybě odhadu význačných oblastí. V případě nutnosti se přidávají nové význačné oblasti, nebo se naopak některé oblasti odeberou.

5 PTAM

PTAM neboli *Parallel Tracking and Mapping* je metoda založena na předpokladu, že lze oddělit proces trasování a mapování. Díky tomu je možné mapování provádět po více snímcích (mnoho snímků obsahuje redundantní informaci). To umožňuje na mapování použít jednu z přesnějších a náročnějších metod, než u metody MonoSLAM. Inicializace je prováděna pomocí tzv. Stereo inicializace, při níž se určité místo zabere ze dvou míst. Kvůli těmto rozsáhlým změnám již nelze použít 2D trasování význačných oblastí, ale je nutné použít i trasování s možností vypočítat pozici kamery. Výpočet polohy je založen na jednoduchém filtru pohybu Decay Velocity (pro odhad polohy) a minimalizaci chyby odhadu polohy kamery. Chyba odhadu je vyjádřena jako medián rozdílu odhadované polohy význačných oblastí a jejich reálnou polohou. Chyba odhadu je pak pomocí varianty metody nejmenších čtverců prostřednictvím úpravy projekční matice minimalizována. To metodě trasování umožňuje přesně určit orientaci v 3D prostoru vůči snímkům, které jsou již zaneseny v mapě. Pokud jsou splněny podmínky pro přidání nového klíčového snímku do mapy, jež jsou uvedeny v podkapitole (5.4), je klíčový snímek přidán do mapy. Pro zpřesnění mapování je použita metoda Bundle adjustment [24] [25] [26]. Metoda Bundle adjustment má kubickou výpočetní náročnost. Tato výkonnostní nevýhoda je vyvážena rozdělením mapování a trasování a zavedením tzv. Local Bundle adjustment, kdy výpočet probíhá jen klíčových snímcích z blízkého okolí aktuálního výskytu kamery.

5.1 Mapa

Mapa se sestává z M význačných bodů umístěných v souřadnicovém systému světa W . Bod s pořadovým číslem j je označován jako p_j a je popsán jako (35)

$$p_{jW} = \begin{pmatrix} x_{jw} \\ y_{jw} \\ z_{jw} \\ 1 \end{pmatrix}, \quad (35)$$

ve snímku koordinátu W . Každý bod dále obsahuje normálu na planární významnou oblast n_j a referenci na pixely textury význačné oblasti. Jedná se o obdobu v předchozí metodě.

Mapa dále obsahuje N klíčových snímků. Snímky jsou snímány v různých časech. Každý snímek je asociován se světovým souřadnicovým systémem centrováním podle polohy a natočení kamery, označován jako K_i pro i -tý klíčový snímek. Transformace mezi světovým souřadnicovým systémem a kamerově centralizovaným systémem je E_{k_iW} . Každý klíčový snímek dále obsahuje

čtyřstupňovou snímkovou pyramidu složenou z šedotónových 8bpp snímků. První snímek je 640x480, poslední 80x60 pixelů.

Textura význačných oblastí není uložena samostatně pro každou význačnou oblast, ale je asociována s klíčovým snímkem ve kterém byla nalezena. Význačná oblast má velikost 8x8 pixelů.

5.2 Trasování

Trasovací část metody zpracovává každý snímek, jež přijde z kamery. Zpracování je možno popsat těmito kroky:

1. Je získán snímek z kamery a je odhadnuta předběžná poloha kamery z modelu pohybu *decaying velocity*.
2. Význačné oblasti, které jsou již dříve nalezeny (umístěny v mapě), se zpětně promítnou do sejmutého snímku, aby bylo možné odhadnout pozici snímku.
3. Menší počet bodů (50) v nejhorším rozlišení (z nejnižšího stupně pyramidy) je nalezeno ve snímku.
4. Odhad pozice kamery je upraven podle výsledků bodu 3
5. Větší množství bodů (1000) je zpětně promítnuto a nalezeno v obrázku.
6. Finální pozice kamery je upravena.

Zpracování snímku z kamery probíhá tímto způsobem. Po pořízení obrázku kamerou na rozlišení 640x480 pixelu, se obrázek rozloží do čtyřvrstvé pyramidy. Každý stupeň pyramidy má poloviční rozlišení jak snímek v nižší vrstvě pyramidy. Po zmenšení snímku na poloviční velikost je vhodně rozmazán gausovým filtrem. Na každou vrstvu pyramidy je zavolán FAST-10 [27] pro detekci rohů v každé vrstvě, bez potlačení maxim. Pro předpověď pozice kamery je použit *decaying velocity model*, který je velmi podobný alpha-beta constant velocity modelu. Ale chybí u něj jakékoliv nové měření, odhad zpomalení, nebo zastavení.

Pozice kamery je jedna z nejdůležitějších informací, díky níž lze urychlit většinu složitých výpočtů. Pomocí ní je možné vybrat již mapované body a transformovat na pozici ve snímku. Nejprve je nutné transformovat pozici bodu z oblasti světového souřadného systému do souřadného systému kamery. Tato operace je provedena přes transformační funkci (36)

$$P_{jc} = E_{cw} P_{jw} . \quad (36)$$

Kde E_{cw} je matice reprezentující předpokládanou pozici a rotaci kamery. Jedná se o člena Lie group SE(3) používané v robotice pro popis rotace a translace těles [28]. Projekce transformovaných bodů P_{jc} přes objektiv kamery je popsán rovnicí (37)

$$\begin{pmatrix} u_i \\ v_i \end{pmatrix} = \text{CamProj}(E_{CW} P_{iW}) . \quad (37)$$

Aby bylo možné používat kamery s větším zorným polem (rybí oko), je nutné popsat snímání model kamery co nejpřesněji. Model kamery se dá popsat rovnicí (38), která byla publikována v článku [29]

$$\text{camProj} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} + \begin{bmatrix} f_u & 0 \\ 0 & f_v \end{bmatrix} \frac{1}{r} \begin{pmatrix} x \\ y \\ z \end{pmatrix} . \quad (38)$$

Kde f_u, f_v jsou ohniskové vzdálenosti, r a r' jsou popsány rovnicemi (39) (40)

$$r = \sqrt{\frac{x^2 + y^2}{z^2}} \quad (39)$$

$$r' = \frac{1}{\omega} \arctan(2r \tan(\frac{\omega}{2})) \quad (40)$$

Díky použití Lie Group SE(3) je pak možné použít standardní postup popsaný pro tuto grupu, k minimalizaci chyby reprojekce bodů do snímku. Díky tomu je možno nalézt opravdové polohy kamery. Nalezení projekce je popsáno rovnicí (41)

$$E'_{CW} = M E_{CW} = \exp(\mu) E_{CW} , \quad (41)$$

kde E'_{CW} je opravdová pozice kamery, korigována reprojekcí mapovaných bodů zpět do obrázků a hledáním minimální odchylky mezi korespondujícími body.

Chceme-li **vyhledat význačné oblasti** ve snímku, zabrané kamerou, je nejprve nutné transformovat vybranou oblast. Transformace je realizována pomocí afinní transformace popsané v článku [2] kapitola 5.3. Poté pomocí metody template matching s velikostí porovnávané oblasti 8x8 nalézt místo, kde se částice opravdu ve snímku nachází. Vyhledávací oblast je fixně omezena na okolí předpovězené pozice význačné oblasti. Po vyhledání opravdové pozice bodů ve snímku je na řadě korekce polohy kamery. Update polohy kamery je řešen pomocí iterativní minimalizace odchylky odpovídajících si opravdových bodů a bodů zpětně promítnutých do snímku. Pro řešení tohoto problému je použita funkce Tukey biweight objective popsaná v článku [30]. Maximální počet iterací je omezen na 10, a to kvůli zachování rychlosti s přiměřenou přesností.

Jak již bylo řečeno dříve v kapitole (5.2), je **upřesnění pozice kamery** rozděleno na dva stupně. V prvním případě se pro upřesnění polohy kamery použije jen 50 význačných oblastí, které se vyhledávají ve větším okolí odhadované pozice. Pomocí toho se upřesní pozice kamery a přichází na řadu druhé kolo upřesnění pozice kamery. Ve druhém kole se zpětně promítne 1000 význačných oblastí. Vyhledávání jejich skutečné polohy probíhá v mnohem menších oblastech. Výsledkem druhého kola upřesňování je až subpixelová přesnost vyhledávání.

Aby byla zajištěna robustnost trasování je při každém snímku kontrolována **přesnost trasování**. Pokud je přesnost trasování horší než určitá hranice, tak trasování pokračuje dál, ale není možné přidávat nové klíčové snímky do mapy. Horší kvalita trasování může být způsobena rozmazáním snímku, nebo nesprávným odhadem pozice kamery.

Pokud trasování selže, je v této metodě implementována opravná metoda popsána v článku [31]. Při selhání trasování již není možné použít aktivního hledání (active search), protože již nemáme informaci o poloze ani o směru pohybu. Úloha sledování pohybu se změní na klasifikační metodu. Klasifikátor se snaží nalézt význačné oblasti již mapované v mapě jak je popsáno v podkapitole (5.5). Po nalezení polohy v mapě se pokračuje v trasování jako normálně.

5.3 Mapování a inicializace mapy

V této části je vysvětleno, jak je realizováno mapování význačných oblastí do mapovacího systému. Pro počáteční inicializaci je použita stereo technika. Po tomto kroku může mapování pokračovat v rozšiřování mapy o další klíčové snímky. Celý proces mapování je možné popsat obrázkem (5.1).

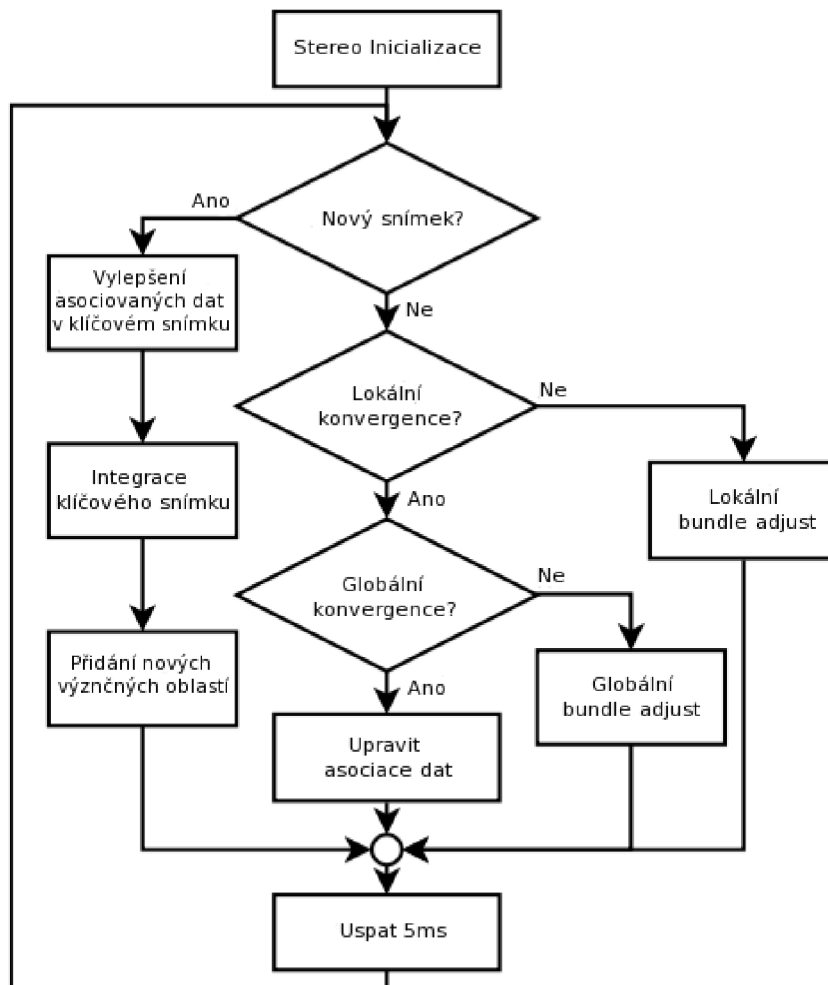
Inicializace mapy je prováděna, když je systém nastartován. Je užit pětibodový stereo algoritmus využívající epipolární geometrii (2.5) a je popsán v článku [13]. Při inicializaci je nutná spolupráce s uživatelem. Uživatel musí začít inicializaci plynulým posunutím kamery o cca. 10 cm, a poté inicializaci dokončit. Začátek a konec inicializace je realizován stiskem na některou klávesu. V průběhu inicializace se vybere okolo 1000 bodů. Mezi body je vyhledávána korespondence pomocí metody RANSAC. Inicializační část může trvat přibližně 3s.

5.4 Vytváření mapy prostředí

U metody PTAM je mapa prostředí složena z **klíčových snímků**. Zpočátku mapa obsahuje jen dva klíčové snímky. Jak se kamera pohybuje jsou přidávány další. Nové klíčové snímky jsou přidány za dodržení těchto pravidel:

1. Kvalita trasování musí být dobrá.
2. Čas mezi snímky musí překročit čas nutný k pořízení dvaceti snímků.
3. Kamera musí být v minimální vzdálenosti od nejbližšího klíčového snímku, který už je v mapě.

Minimální vzdálenost mezi snímky je závislá na průměrné vzdálenosti pozorovaných bodů. Pokud je kamera blízko pozorovaných bodu, je i vzdálenost mezi klíčovými snímky malá, a naopak. Díky tomu je možno předejít ničení mapy při malém pohybu kamery (je zajištěna dobrá triangulace mezi snímky).



Obrázek 5.1: Asynchronní mapovací vlákno. Po inicializaci vlákno pokračuje nikdy nekončící smyčkou. Převzato a upraveno z [2].

Pro každý klíčový snímek je předpokládáno, že trasovací systém určí polohu kamery a polohu všech významných oblastí. Bohužel měřit všechny význačné oblasti není možné, protože je nutné stíhat trasovat každý snímek. Z tohoto důvodu si zbývající body musí dopočítat samo mapování. Mapování tuto činnost provádí pomocí epipolárního vyhledávání (epipolár search). Kdy se k význačné oblasti v jednom snímku hledají body v druhém snímku. Body jsou vyhledávány v blízkém okolí epipolární linie odpovídající bodu z prvního snímku. Množina testovaných bodů okolo epipolární linie je omezena odhadem rozsahu hloubky význačné oblasti. Pro zvýšení rychlosti vyhodnocování jsou dále vyřazeny body, které leží blízko bodů vyhodnocených trasováním. Pro zbylé nové body je nutné dopočítat informaci o hloubce. To je možné jen díky triangulaci s nejbližším klíčovým snímkem. Korespondence mezi klíčovými snímky je řešena pomocí epipolar search.

Protože jsou klíčové snímky z počátku vkládány na základě metody trasování jejich poloha nemusí být vždy úplně přesná. Z toho důvodu je nutné propočítat přesné vzájemné pozice klíčových snímků pomocí metody **Bundle adjustment**. Pokud je kamera na jedné lokalitě a není nutné přidávat nové snímky, je možné propočítávat vzájemnou polohu všech snímků v mapě (Global Bundle adjustment). Pokud je mapa příliš velká (např. 150 snímků), může na dnešních počítačích tato činnost trvat i déle než 10s. To je způsobeno tím, že výpočet Bundle adjustment má kubickou složitost $O(N^3)$. To je nepřijatelné v případě, že kamera zkoumá nový prostor, kde je nutné přidávat velké množství nových klíčových snímků. V tomto případě se přejde na omezenou metodu Local Bundle adjustment, kdy se vybere menší množství snímků. Většinou se vybírají snímky ležící v okolí pozice kamery. Ve vybraných snímcích se najdou body, které jsou viditelné ze všech snímků a pomocí nich se provede Local Bundle adjustment.

V případě, že se kamera nehýbe, nebo se pohybuje v dobře zmapovaném prostředí, je volný čas mapování použit pro korekci starých klíčových snímků a ověření pozice bodů měřených trasovací částí metody. Trasovací část je často nepřesná a může se stát, že body odhadnuté v této části systému jsou špatně změřeny. Proto je těmto bodům dána menší váha při určování pozice. Pokud se bod, získaný trasovací částí systému, použije pro zjištění pozice kamery a selže, je označen jako outliers. Outliers bodu je dána ještě jedna šance. Pokud selže i v dalším případě, je z mapy vyřazen.

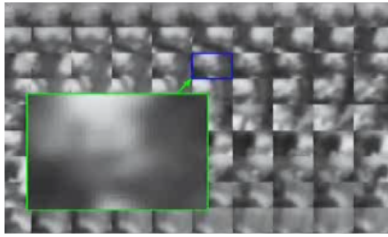
5.5 Znovunalezení pozice kamery

V této části bude popsán princip znovu nalezení pozice kamery po ztrátě aktuální informace o poloze kamery. Znovu nalezení pozice kamery ve scéně je realizováno velice jednoduchou metodou. Používají se snímky s na nejvyššími stupni obrazové pyramidy (s nejmenším rozlišením), které se standardně v metodě počítají a ukládají, což je popsáno v kapitole (5.2). O snímcích v mapě je známá informace o jejich poloze a orientaci. Vyhledávání pozice je realizováno pomocí srovnávání snímku z kamery a všemi ostatními klíčovými snímky uloženými v mapě. Vyhledání snímku v mapě je popsáno následující rovnicí (42)

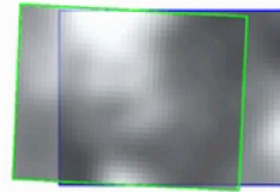
$$index = \underset{index}{\operatorname{minarg}} \left(\sum_{x=0}^{width*height} (c_x - m_{index,x})^2 \right), \quad (42)$$

kde c_x je pixel ze snímku kamery, $m_{index,x}$ bod ze snímku z mapy, který je vybrán pomocí indexu. Funkce vrátí index obrázku, se kterým byl obrázek z kamery nejpodobnější. Jak je na obrázku (5.2) vidět.

Po nalezení nejpodobnějšího snímku se propočítá natočení a posunutí obou snímků vůči sobě aby se co nejpřesněji našla poloha kamery v prostoru. Což popisuje obrázek (5.3). Po nalezení polohy se opět běžným způsobem spustí tracker.



Obrázek 5.2: Nelezení nejpodobnějšího obrázku z Mapy. Převzato z [32].



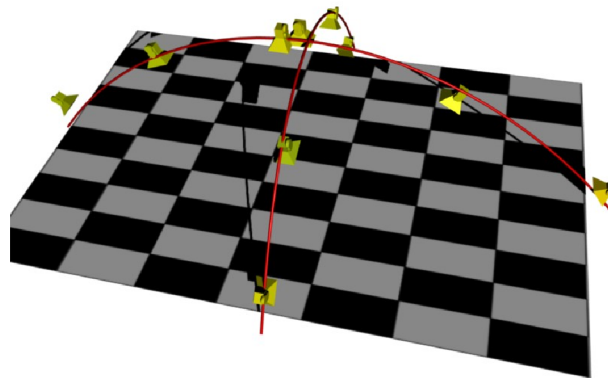
Obrázek 5.3: Nalezení nejlepší shody obou snímků. Převzato z [32].

5.6 Souhrn metody

Metoda PTAM je založena na klíčových snímcích. Jak již zkratka napovídá, trasování a mapování běží paralelně. Tracker určuje okamžitou polohu kamery pro každý snímek a pro tento účel používá jednoduchý filtr Decaying Velocity. Pomocí odhadnuté polohy vybere vhodný klíčový snímek z mapy a udělá korekci odhadnuté polohy. Pokud kamera urazí dostatečnou vzdálenost od posledního klíčového snímku a tracker má informaci o aktuální pozici kamery, mapování si zařadí nový snímek do mapy klíčových snímků. Mapování udržuje informaci o pozici a orientaci klíčových snímků a navíc pomocí metody Bundle Adjustment opravuje a upřesňuje pozice snímků v mapě klíčových snímků. Mapování pracuje ve dvou režimech Local Bundle Adjustment a Global Bundle Adjustment v závislosti na rozsáhlosti vytvořené mapy klíčových snímků. Pokud tracker ztratí informaci o aktuální poloze, přijde na řadu vyhledávání, které výše popsaným způsobem (5.5) opět najde pozici a metoda dále pokračuje v činnosti. Tato metoda těží z použití klíčových snímků, které jí dávají často množství nadbytečných informací, které se dají dále použít k zpřesňování a rozšiřování.

5.7 Postup kalibrace kamery

U metod, které získávají informace o svém okolí ze záběru kamery, je velice důležité správně nastavit či zjistit parametry kamery. Aby bylo možné tyto parametry zjistit, máme dvě možnosti. Buď se před samotným zahájením měření spustí oddělená metoda kalibrace, nebo se parametry kamery zjišťují za běhu metody získávající informace ze svého okolí pomocí vizuální informace. U real-timeových metod je preferována oddělená kalibrace od zbytku metody, aby se metoda nezdržovala zbytečnými operacemi.



Obrázek 5.4: Průběh kalibrace kamery.

U metody PTAM je kalibrátor také oddělen od samotné metody. Při kalibraci se zjišťuje ohnisková vzdálenost osy X, osy Y a rozbíhavost paprsků. Z těchto důvodů je dobré při kalibraci postupovat jak je uvedeno na obrázku (5.4).

6 Teoretické porovnání metod

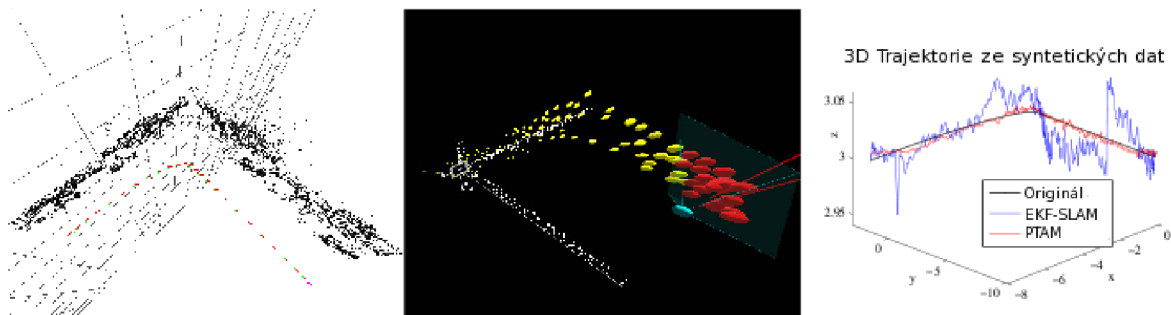
V této kapitole bude uvedeno teoretické porovnání obou metod na základě výše popsaných kapitol. Budou zde nastíněny základní postupy, které metody používají a problémy s nimiž se metody snaží více či méně úspěšně vyrovnat.

	MonoSLAM	PTAM
Základní metoda	Pravděpodobnostní vyjádření pozice kamery i pozice významných bodů	Ukládání klíčových snímků, oddělen proces mapování a trasování
Detekce významných oblastí	Shi and Tomasi	Fast 10, hranice pro mapování obdoba Shi and Tomasi
Metoda předpovědi (active search)	Extended Kalman filter (při velkých nelinearitách problém se stabilitou)	Decaying velocity model (nejzákladnější model, nepřesný)
Metoda upřesnění pozice určené filtrem	Náhled a srovnání s mapovanými body	Náhled a srovnání s mapovanými klíčovými snímky
Metoda zpřesňování modelu	Není chyba se rozšiřuje	Bundle adjustment
Možnost separace na procesy	Ne	Ano
Počet částic v systému	100	5000
Problém s přehlcním systému velkým množstvím bodů.	Ano	Ne
Řešení přehlčení	Vyřazování nepotřebných bodů, a špatných bodů	Vyřazení špatných bodů
Metoda re-lokalizace	Real-Time SLAM Relocalisation [33]	Uveden v kapitole (5.5)
Relokace (nalezení pozice při ztrátě informace o umístění)	Ne (ve SceneLib není, ale je doimplementováno v jiné práci)	Ano (Hlednání nejpodobnějšího snímku v klíčových snímcích)
Detekce uzavření smyčky	Ne (ve s SceneLib, ale v jiných článcích implementováno)	Ne (není implementováno metoda je často natolik přesná, že není potřeba)

Tabulka 6.1: Porovnání vlastností metod MonoSLAM a PTAM.

6.1 Vzájemné porovnání přesnosti metod

Jak je vidět na obrázku (6.1), metoda PTAM je přesnější než metoda MonoSLAM. Metoda MonoSLAM se potýká s problémy, kdy je kamera staticky zaměřena na určité místo zájmu, a kvůli tomu má problém s určením informace o hloubce význačných oblastí. V tomto případě může MonoSLAM chybně určit hloubku význačné oblasti. Metody PTAM se tento problém netýká, protože pro mapování a trasování používá klíčové snímky. Klíčové snímky tvoří paměť, ve které může metoda PTAM najít i body, které v předchozím běhu nenašla.



Obrázek 6.1: V levé části obrázku je mapa bodů nasnímaná metodou PTAM, uprostřed body nasnímané metodou MonoSLAM a napravo porovnání chybového výstupu obou metod. Obrázek přejet z [2].

6.2 Odolnost vůči rozmazání

Pokud se kamera pohybuje rychle, nastávají problémy s rozmazáním snímku. V případě rozmazání snímku není možné úspěšně detekovat význačné oblasti. Metoda PTAM by měla být odolnější vůči rozmazání obrazu vlivem rychlého posunu kamery, než metoda MonoSLAM. Větší odolnost vůči rozmazání by měla být dosažena pyramidovou strukturou uložení klíčových snímků (význačných oblastí). Stupně pyramidy s nižším rozlišením postrádají přesné detaily. Díky použití porovnávání na nižších stupních pyramidy, je možné rozpoznávat význačné oblasti i při jistém stupni rozmazání (jakoby menší detail). Na rozdíl od PTAM, metoda MonoSLAM detekuje význačné oblasti jen na nejdetailnější úrovni obrazových dat, a proto by měla být méně odolná.

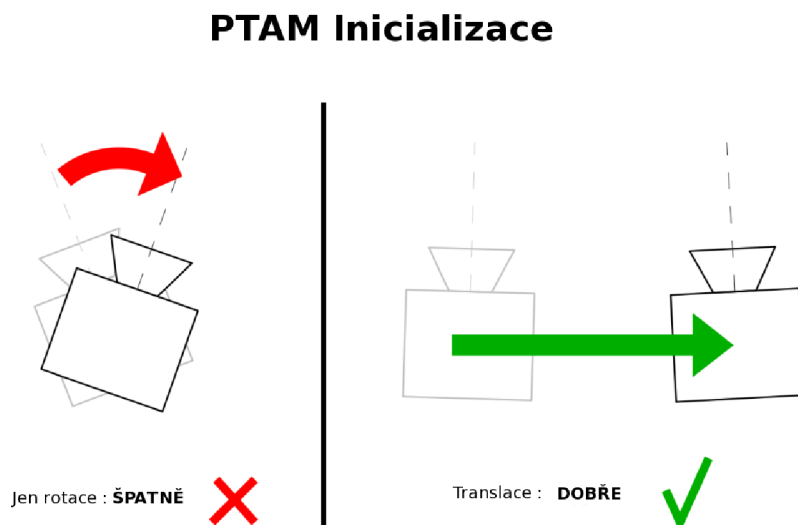
Do metody PTAM byla navíc přidána detekce hran. Bohužel toto rozšíření není volně přístupné, a proto jej není možné v této práci testovat. Bližší informace jsou uvedeny v článku [34].

6.3 Problém kalibrace

Jednou z vážných potíží obou metod je spolupráce s uživatelem. U obou metod je nutné zajistit správnou kalibraci kamery. V případě špatné kalibrace nastává problém s převodem 2D obrazové informace na 3D reprezentaci pomocí bodů a naopak. Pro určení relativní polohy (polohy, ve které si model myslí že je) to nepředstavuje takový problém, až do okamžiku, kdy aplikace potřebuje určit reálnou polohu nebo vzdálenosti. Vliv kalibrace na vlastnosti metody bude otestován níže v části práce zabývající se výsledky experimentů.

6.4 Inicializace metod

Obě metody při prvním spuštění řeší problém s prvotní inicializací polohy. Metoda MonoSLAM potřebuje pro inicializaci nalézt předem známé seskupení význačných oblastí, pomocí kterého si určí relativní polohu. Naproti tomu metoda PTAM tento známý objekt nepotřebuje, ale zato je nutná spolupráce uživatele při prvotní inicializaci. Prvotní inicializace spočívá v tažení kamery pro získání stereo záběru.



Obrázek 6.2: Inicializace metody PTAM. Přejat z [35].

V tomto případě jsou obě metody srovnatelné, avšak pro řadu aplikací je absence známého objektu značnou výhodou. Jedná se například o interaktivní aplikace například hry, které by mohly být pomocí SLAM či PTAM metod realizované. Naopak v případě potřeby přesného určení absolutní polohy kamery je inicializace pomocí pětibodové metody tažení kamerou takřka nepoužitelná. V tomto případě je inicializace pomocí známého seskupení význačných oblastí lepší alternativou.

6.5 Problém zorného úhlu

Pro vizuální určení pozice kamery je nutné získat z prostředí dostatečné množství informace o okolí. Při použití kamer s malým zorným úhlem je tato informace často omezená. V případě, že je metoda používána pro orientaci v malém prostoru mají obě metody tendenci ztrácet informaci o své pozici v prostoru. Podle teoretických předpokladů by metoda PTAM měla tuto situaci zvládat lépe, protože má na-implementovánu metodu znovunalezení své pozice.

7 Srovnání metod na základě pokusů

V této části budou srovnávány obě metody PTAM a MonoSLAM. Metoda MonoSLAM bude testována prostřednictvím implementace pocházející od jejího tvůrce s názvem Scenelib. Taktéž metoda PTAM bude testována ve verzi vytvořenou jejím tvůrcem. Budou zde uvedeny parametry, pomocí nichž budou obě metody porovnávány. Dále zde budou uvedeny vlastnosti, které se budou testovat, postup testování a výsledky provedených testů. Obě metody jsou testovány vůči offline metodám implementovaným v aplikacích Voodoo Camera Tracker [36] nebo SynthEyes.

U konkrétních testů se bude měřit přesnost určení pozice kamery. To znamená souřadnici X,Y a Z. Všechny metody používané v těchto testech vrací relativní souřadnici vůči inicializačnímu místu. Inicializace jsou prováděny různým způsobem, jak je popsáno v předchozích kapitolách. Proto relativní pozice nemají stejnou orientaci ani měřítko. Z toho důvodu je nutné pro srovnání metod nejprve přepočítat relativní souřadnice všech metod do stejného souřadného systému. Porovnávání metody lze pak pomocí přepočítaných souřadnic bez větších problémů srovnat. Pro rozšíření srovnávání je možné přidat i informaci o orientaci (natočení) kamery v prostoru.

Co a za jakých okolností bude testováno je popsáno v následující části.

- **Vliv nastavení vyhodnocující metody**
 - Počet bodů, klíčových snímků, které se používají pro detekci pohybu
- **Vlivy prostředí na přesnost metody**
 - Statický přímočarý pohyb směrem dopředu a bokem
 - Pohyblivé, měnící se objekty ve scéně
 - Barevné vlastnosti
 - Prostředí s dobře detekovatelnými význačnými oblastmi
 - Prostředí s texturními vlastnostmi
 - Rozloha snímaného prostředí (indoor, outdoor)
 - Jednoduché prostředí, kde jsou všechny body viditelné ze všech směrů.
- **Vliv snímací soustavy na přesnost měření**
 - Objektivy
 - Širokoúhlý

- Standardní
- Vliv kalibrace kamery

7.1 Postup testování

Testování probíhá dvěma způsoby. Zaprvé se video k testování nasnímá pomocí neurčité manipulace s kamerou. Například může být kamera nesena rukou bez zvláštního vedení. Zadruhé je kamera uchycena na zařízení, které s kamerou manipuluje specifikovanou rychlostí po přesně určené dráze. Pro vyhodnocení první varianty, kdy je kamera nesena po neurčité dráze, se pro vyhodnocení přesné dráhy kamery použije dosti přesná offline metoda. Výsledek offline metody se použije jako reference pro srovnávání výsledků obou real-time metod MonoSLAM a PTAM.

Postup testování probíhá následujícím způsobem. Video z kamery se uloží jako obrázky ve formátu png, ppm do vybrané složky. Při nahrávání videa musí být zajištěno, aby bylo možné na inicializovat obě metody. Video pro MonoSLAM musí začínat pohledem na inicializační objekt. V tomto případě papír s černým obdélníkem, jehož velikost je definovaná v metodě MonoSLAM a je obsažen v přílohách k této práci. Dále musí být pro metodu PTAM na začátku videa v prvních pár sekundách pomalý tah určitým směrem (vizuálně posun středu kamery ve scéně zhruba o 10-15cm). Poté se spustí skript, který nad videosekvencí spustí příslušné metody rozpoznávání pohybu kamery. Protože výsledky výstupů z metod jsou synchronizovány, spustí se nad nimi metoda, která transformuje výsledky všech metod do odpovídajícího si formátu a určí chybu obou srovnávaných metod.

7.2 Hardwarové vybavení

Všechny testy budou probíhat na kamerách definovaných v tabulce (7.1). Všechny hodnoty jsou převedeny na pixely, protože porovnáváné metody se nastavují pomocí parametrů zadaných v pixelech, popřípadě nějakou ekvivalentní reprezentací. Rozbíhavost, neboli parametr poduškovitého zkreslení, je brána z kalibrační metody z knihovny MRPT, protože obě testované metody používají jiný zjednodušující model kompenzace rozbíhavosti paprsků.

Název	Fire-i unibrain	Logitech Webcam C905	Logitech QuickCam Deluxe for Notebooks
Ohnisková vzdálenost v pixelech přes X	182	347.3	347.3
Ohnisková vzdálenost v pixelech přes Y	182	347.3	347.3
Rozlišení snímaného videa	320x240	320x240	320x240
Frekvence snímků/s	30	15	20
Kvalita snímků	výborná	dobrá	špatná
Typ chyby	Velký polštářkový efekt	Jen menší zorný úhel	Video rozmazané a tmavé
Zorný úhel	80.95°	60.50°	60.50°

Tabulka 7.1: Vlastnosti webových kamer použitých při testování.

7.3 Softwarové nástroje

V této podkapitole budou popsány softwarové nástroje potřebné pro předzpracování videosekvencí, na kterých následně budou testovány obě metody, a nástroje pro zpracování nasbíraných informací z testovaných metod MonoSLAM a PTAM.

Nástroje pro předzpracování videosekvence a zpracování dat z metod budou uvedeny v následujících výčtech, kde budou navíc stručně charakterizovány.

Nástroje pro předzpracování videosekvence

- blur.py
 - Nástroj má za úkol rozmazat videosekvenci jako při rychlém pohybu kamery (Gausiánem simulující dynamický pohyb). Pro zdárné inicializování obou metod je v parametrech zahrnuta i hranice od které se má video pohybovým Gausiánem rozmazat. Dále je možné nastavit i intenzitu rozmazání.
- blurrnd.py
 - Obdobný nástroj jako blur.py, ale rozmazání probíhá náhodnými intenzitami s předdefinovanými rozestupy.
- camrecord.py

- Pomůcka pro nahrávání videa z usb kamery v požadovaném formátu.
- `convert.py`
 - Nástroj pro převod videosekvence mezi formáty spojený s jejich přechíslováním.
- `remover.py`
 - Nástroj určený pro simulaci výpadků snímků. Z videosekvence odstraňuje bloky s předdefinovanou velikostí s minimální předdefinovanou vzdáleností mezi rozestupy.

Nástroje pro zpracování nasbíraných dat z testovaných metod

- `MatrixSyncc`
 - Nástroj má za úkol, po předložení originální stopy naměřené referenční metodou a stopy testované metody, pomocí homografie a metody RANSAC srovnat obě stopy tak, aby si jejich relativní souřadné systémy odpovídaly. Jako parametry jsou nastaveny začátek srovnávání (inicializace metod), délka srovnávání a samotná délka videosekvence.
- `PathDiferent.py`
 - Nástroj pro následné zpracování srovnaných stop pomocí nástroje `MatrixSyncc`. Nástroj spočítá sumu Euklidových vzdáleností mezi jednotlivými prvky testovaných stop a k tomu zobrazí rozdíly bloků stop pro předem definovaný počet bloků videosekvence.

Zmiňované nástroje jsou nahrány na příloženém DVD. Všechny nástroje jsou napsány pro běh v Linuxu, ale po malých úpravách je není problém rozběhnout takřka na jakékoliv platformě. Pro správný chod nástrojů napsaných v programovacím jazyce Python je nutné mít nainstalovány nástroje `ImageMagic`. A pro kompilaci `MatrixSyncc` je potřeba mít nainstalovanou `OpenCV` verze 2.1.

7.4 Vliv nastavení metod

Aby bylo možné provést korektně provádět další testování je nejprve nutné zjistit správné nastavení obou metod. Z tohoto důvodu je tato část testování provedena jako první. V rámci tohoto testu bude ověřena i robustnost obou metod v závislosti na nastavení parametrů kalibrace. Testování robustnosti je důležité pro ostré nasazení, kdy budou metodu používat běžní uživatelé, kteří nemusí mít s kalibrací a nastavováním metod žádné zkušenosti, a proto by mohli dosahovat špatných výsledků.

První část bude zaměřena na testování a nalezení nejlepšího nastavení pro práci metod. Pro metodu PTAM to bude nalezení ideální konstanty upravující počet klíčových snímků. Dále pro metodu MonoSLAM nalezení nejlepšího počtu význačných oblastí. Předpoklad pro toto testování je správně nastavená kalibrace kamery. Postup a teorie o kalibraci kamery je v podkapitole (5.7). Jako reference budou použity výsledky offline metody. Poté se v rámci testované metody nastaví ideální nastavení a bude se zkoumat vliv určité procentuální změny v nastavení kalibrace kamery na přesnost testované metody.

Nejprve k výsledkům metody PTAM, v první části byla testována závislost mezi počtem snímků na plochu testu a přesností metody. Plocha je uváděna v počtu záběrů kamery, ve kterých bude více jak 80 % nové informace. Toto zvláštní měřítko je zavedeno, protože je možné jedním klíčovým snímkem při správné afinní transformaci pokrýt celý prostor, ze kterého je klíčový snímek viditelný.

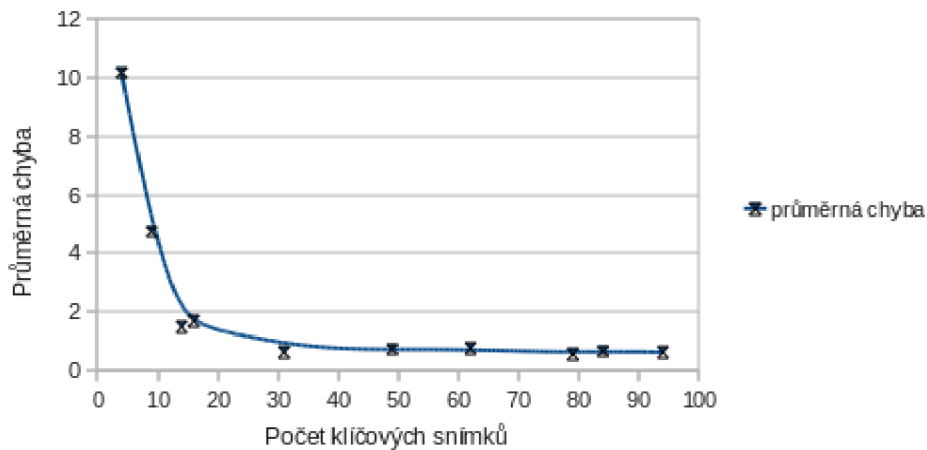
Pro vybrané video pro první testy je toto měřítko asi 4. Jde o video sekvenci *obyvak4*. Výsledky první části testu jsou uvedeny v tabulce (7.2). Počet klíčových snímků ve videosekvenci je měněn pomocí změny parametru určující vzdálenosti mezi klíčovými snímky. Tento parametr bude nazvaný D a udává relativní zkrácení jednotky vzdálenosti mezi klíčovými snímky odvozené od prostoru inicializace metody PTAM.

Parametry		Průměrná chyba	Průměrný koeficient růstu chyby
D	K		
0.01	94	0.61	0.03
0.05	84	0.65	0.03
0.2	79	0.56	0.04
0.5	62	0.74	0.03
1	49	0.65	0.03
2	31	0.62	0.03
3	16	1.69	0.03
4	14	1.47	0.02
5	9	4.74	0.07
5	4	10.17	0.47

Tabulka 7.2: Závislost přesnosti metody PTAM na počtu klíčových snímků.

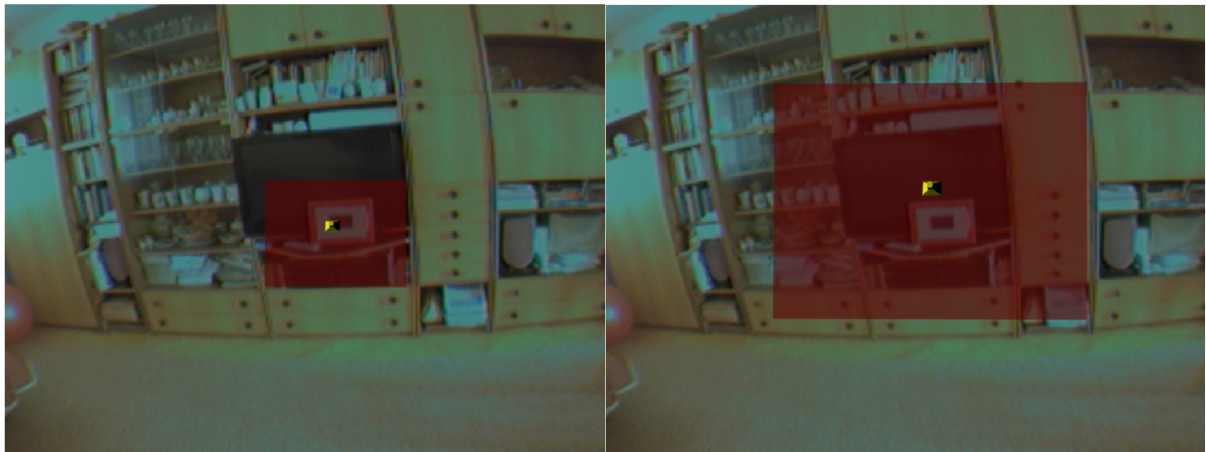
V závislosti na parametru D se mění počet klíčových snímků K , které metoda PTAM začlení do mapy. Průměrná chyba je definována jako průměrná Euklidova vzdálenost mezi referenčním průběhem vypočítaným pomocí offline metody a průběhem metody PTAM.

Závislost počtu klíčových snímků a přesnosti měření polohy je znázorněno v grafu na obrázku (7.1). Jak je z obrázku i tabulky patrné, s narůstajícím počtem klíčových snímků roste i přesnost metody. Na obrázku jsou vidět i body, které nesplňují tendenci křivky chyby a mají menší chybu, než by podle předpokladů měli mít. Tento efekt je způsoben ruční inicializací, kdy různé inicializační informace ovlivňují další chování celé metody.



Obrázek 7.1: Graf závislosti počtu klíčových snímků a přesnosti měření PTAM.

Metoda při inicializaci zjistí relativní rozměr okolí ve kterém se bude kamera pohybovat. Jestliže je metoda na-inicializována příliš blízko nějakého objektu při malém pohybu kamery, metoda předpokládá, že se bude pohybovat v malém prostředí. Neboli se zmenší relativní jednotka



Obrázek 7.2: Prostor inicializace metody PTAM. V levé části obrázku je malý prostor inicializace, kdy při pohybu kamery po prostoru zobrazené scény musí metoda vytvářet nové a nové snímky. Pravá část ukazuje inicializaci, kde je inicializován skoro celý prostor, a proto metoda nemusí vkládat do mapy tolik klíčových snímků. Proto je i přesnější, jelikož používá jako referenci inicializovaný prostor.

vzdálenosti mezi klíčovými snímky. Další parametr, který zapříčinil vybočení několika bodů z řady je i samotný prostor, který metoda při inicializaci zkoumala, jak je popsáno na obrázku (7.6).

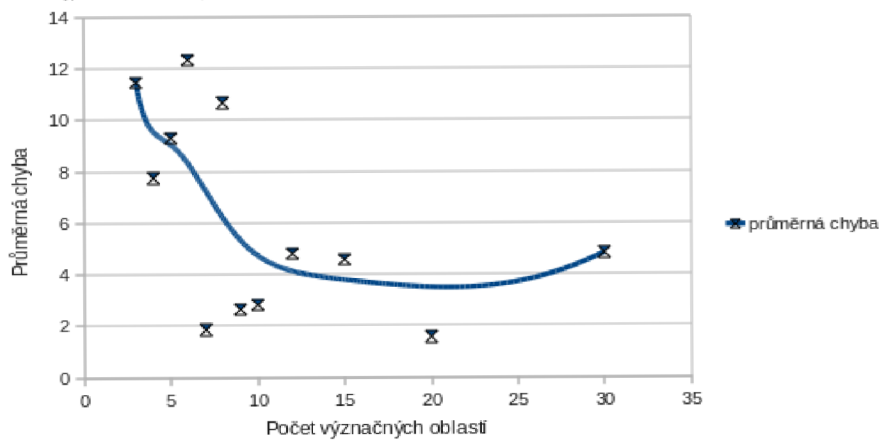
V této pod části testu bude porovnána závislost přesnosti metody MonoSLAM na počtu význačných oblastí, podle kterých se metoda orientuje v prostoru. Výsledky tohoto testu jsou uvedeny v tabulce (7.3). Závislost chyby měření na počtu význačných oblastí je znázorněna v grafu na obrázku (7.3). Je z něj patrné zmenšení chyby měření při zvětšování počtu význačných oblastí. Avšak toto zmenšování chyby naráží na problém nalezení dostatečného množství význačných oblastí. Je sice možné zmenšovat práh pro určení, že se opravdu jedná o význačnou oblast, ale v tom případě metoda začne brát v úvahu i různé stíny a odlesky v obraze.

Počet význačných oblastí	Průměrná chyba	Snímky do totální ztráty orientace ze 1700 snímků
3	11.45	200
4	7.75	1300
5	9.29	300
6	12.33	800
7	1.84	1600
8	10.68	800
9	2.62	1500
12	4.79	800
15	4.55	900
20	1.56	1700
30	4.82	700

Tabulka 7.3: Závislost přesnosti metody MonoSLAM na počtu význačných oblastí.

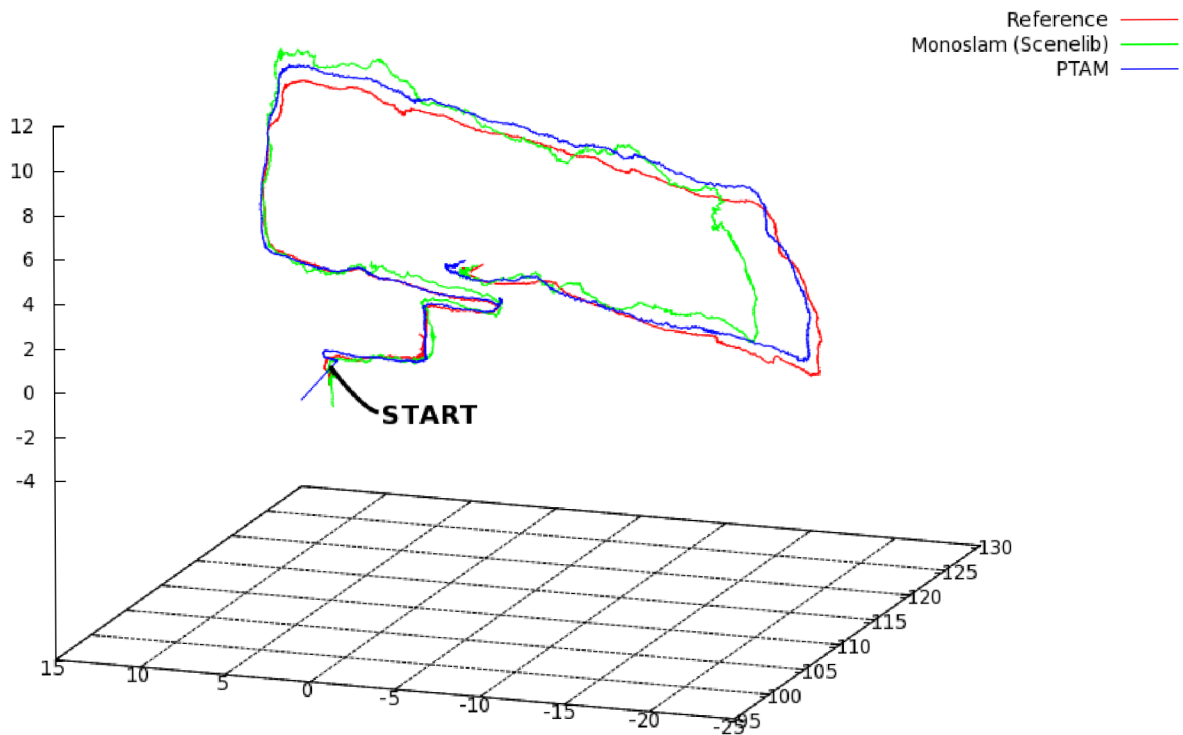
Obrovská chyba přesnosti může být také zapříčiněna samotnou metodou v kombinaci s lesklými objekty, které se na testovací scéně nachází. I přes vědomí tohoto problému byla scéna vybrána pro jednoduchost pohybu kamery a její efektovou komplexnost. Efektová komplexnost znamená, že jsou na scéně vidět odlesky, se kterými by se měla metoda umět vypořádat. Správná detekce špatně určených, nebo pohyblivých bodů je jedna z hlavních vlastností, které musí reálně nasaditelná metoda pro orientaci kamery v prostoru obsahovat. Další velká chyba MonoSLAM v podání Scenelib je absence algoritmu pro opětovné nalezení pozice, při ztrátě informace o poloze kamery.

Dále je z tabulky (7.3) patrné, že ztráta orientace se sice se zvětšujícím počtem bodů zlepšuje, ale u metody MonoSLAM se jedná spíše o náhodný jev, kdy se metoda ztratí. Je to způsobeno vlivem zvolené metody výběru význačných oblastí. Výběr význačných oblastí je popsán v podkapitole (4.2).



Obrázek 7.3: Graf závislosti chyby měření na počtu význačných oblastí.

Výsledek nejlepšího běhu obou metod je znázorněn na obrázku (7.4). Jak je vidět z obrázku i tabulek, metoda PTAM je přesnější a stabilnější.



Obrázek 7.4: Nejlepší průběh metod PTAM a MonoSLAM.

V této podčásti bude porovnávat vliv špatného nastavení metody na přesnost měření informace o poloze kamery. Ideální kalibrace získaná z kalibrační metody bude uměle zhoršována od ideálního nastavení a bude sledován vliv této chyby.

Metoda	Typ přidávané chyby	Chyba měření
PTAM	žádná	0.65
	Ohnisková vzdálenost x -10%	0.76
	Ohnisková vzdálenost x -20%	1.43
	Ohnisková vzdálenost x -30%	2.43
	Rozbíhavost paprsků -5%	0.77
	Rozbíhavost paprsků -10%	5.19
	Rozbíhavost paprsků +10%	1.11
MonoSLAM	žádná	1.56
	Ohnisková vzdálenost x -10%	2.60
	Ohnisková vzdálenost x +10%	1.79
	Ohnisková vzdálenost x +5%	1.95
	Rozbíhavost paprsků -10%	2.32
	Rozbíhavost paprsků +10%	4.91
	Rozbíhavost paprsků +20%	10.51

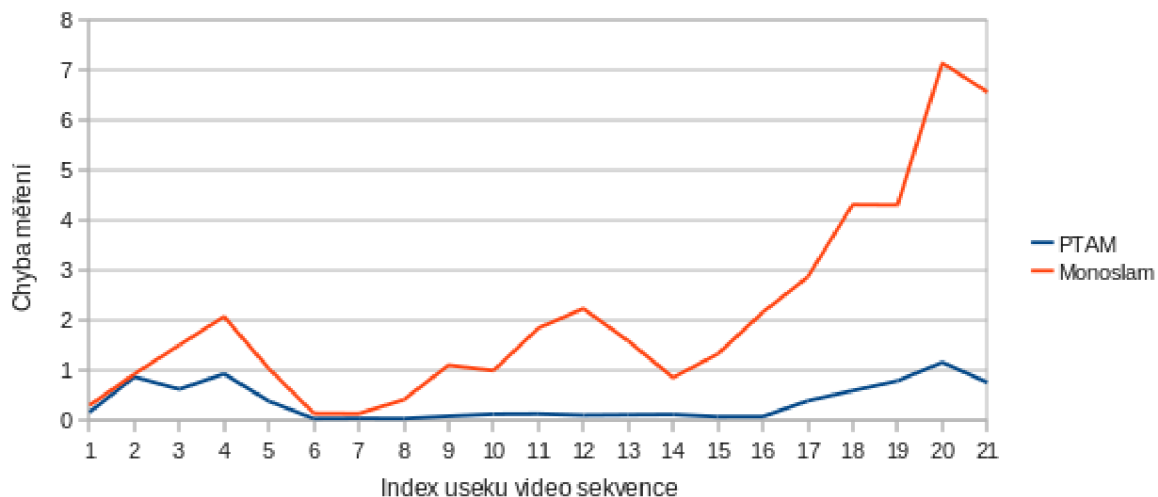
Tabulka 7.4: Tabulka závislosti nepřesnosti kalibrace na přesnosti měření.

Z naměřených údajů je vidět větší robustnost metody PTAM v závislosti na chybě zavedené při kalibraci kamery. Dále je patrná nestabilita metody MonoSLAM, kdy pro 5% chybu v ohniskové vzdálenosti metoda vrací větší chybu měření než u 10% odchylky.

7.5 Rozloha snímaného prostředí

V této části testů nebude zkoumáno problematické a rozsáhlé prostředí, ve kterém se objekty překrývají, poněvadž ani jedna metoda si neklade za cíl tuto problematiku řešit. Bude zde porovnávat vliv kumulace chyby v průběhu výpočtu.

Test bude prováděn na videu s rozlohou asi 10 záběru kamery. Pro testování byla použita videosekvence *dvursize21*. Tyto snímky jsou vedle sebe a jsou spojeny jen jednou hranou. Výsledek průběhu chyby měření je vidět na obrázku (7.5).



Obrázek 7.5: Graf znázorňující průběh kumulace chyby přes průběh video sekvence.

Metoda PTAM podává opět lepší výsledky než metoda MonoSLAM. Metoda PTAM v tomto případě těží z metody Bundle Adjustment, která v průběhu videosekvence upřesňuje polohu klíčových snímků, a tím zajišťuje přesný průběh detekce polohy.

7.6 Vliv snímací soustavy

Pro reálné nasazení metod je nutné předpokládat použití nevhodné kamery. Obzvláště metody, které získávají veškeré informace ze snímku kamery, jsou na toto choulostivé.

V této části budou testovány dvě různé kamery. První kamerou je širokoúhlá s vysokým počtem snímků Fire-i unibrain a druhou standardní webová kamera Logitech Webcam C905. Oběma kamerami byla natočena co nepodobnější video sekvence, jež je zobrazena na obrázku (7.6). Z obrázku je patrné, že širokoúhlá kamera podává metodě daleko více informace o okolním prostoru.



Obrázek 7.6: Vlevo záběr Fire-i unibrain vpravo Logitech Webcam C905. Pro měření byly použity videosekvence *dvurline21* a *dvurline37*

Následující výsledek měření je nutné brát s rezervou, protože měření pochází z různých videosekvencí téhož pohybu. Z naměřených výsledků, uvedených v tabulce, je přesto zřejmá výhoda použití širokoúhlé kamery.

Typ kamery	PTAM chyba	MonoSLAM chyba
Širokoúhlá	0.15	1.23
Běžná web kamera	1.83	4.20

Tabulka 7.5: Výsledky měření přesnosti širokoúhlé a normální web kamery.

Jako další bude testován vliv rozmazání snímku scény. Aby bylo testování co nejpřesnější, je vliv rozmazání testován na jedné video sekvenci. Do vybrané video sekvence *blur329* bylo uměle přidáno pohybové Gausovo rozmazání s různým nastavením σ . Rozmazání bylo nastaveno v závislosti na pozici ve videu. Na začátku videa rozmazání nebylo aplikováno vůbec, aby se obě metody zvládly bez problémů nainicializovat. Po uplynutí času potřebného pro inicializaci metod, je pak dále uplatňováno rozmazání podle potřeby. Buď staticky po celou dobu, anebo náhodně po úsecích videa maximálně do hodnoty největšího povoleného rozmazání. Souhrn výsledků této podčásti tohoto testu je uveden v tabulce (7.6).

Typ pohybového rozmazání	PTAM chyba	MonoSLAM chyba
Bez rozmazání	1.06	1.23
Statické $\sigma=1$	1.17	1.19
Statické $\sigma=2$	1.19	1.26
Statické $\sigma=3$	1.13	1.62
Statické $\sigma=4$	1.60	1.48
Statické $\sigma=5$	1.92	1.19
Statické $\sigma=7$	1.88	1.75
Náhodné do $\sigma=8$	1.30	1.24
Náhodné do $\sigma=12$	2.05	1.17

Tabulka 7.6: Výsledky vlivu rozmazání na přesnost testovaných metod.

Z výsledků měření je vidět větší náchylnost metody PTAM na rozmazání snímků. Největší problém metodě PTAM dělalo náhodné rozmazání s vysokým σ . Naopak metoda MonoSLAM s rozmazáním nemá až takové problémy. Může to být způsobeno extrémní jednoduchostí pohybu, kdy Kalman filtr odvádí výbornou práci, a metodou užitou pro detekci významných oblastí.

7.7 Barevné vlastnosti

V této podkapitole bude testována přesnost či schopnost testovaných metod pracovat při problematických barevných podmínkách. Pro metody pracující s vizuální informací je často problém rozeznat od sebe opakující se (texturní), nebo nevýrazné vzory.

Metody budou testovány na texturní vzor opakující se ve více rozměrech, například vzorovaný koberec, jak je vidět ve videosekvencích *texture*, *texture2*, *texture3* a *texture4*. Dále budou metody testovány na vzor opakující se výhradně v jednom rozměru. Například natažený drát, který vytváří výborné záchytné význačné oblasti, ale při špatném opětovném hledání může význačná oblast zdánlivě klouzat po směru opakování vzoru.

Výsledky metod v tomto testu budou uvedeny ve formě textu, protože bylo testováno několik různých videosekvencí. Byly testovány čtyři videosekvence. První byla zabrána širokoúhlou kamerou, z toho vyplývá menší ostrost výsledného obrazu. Metoda PTAM tuto sekvenci zvládla bez větších potíží. Naopak metoda MonoSLAM měla problémy jak s texturními znaky, tak i se samotným pohybem kamery ve scéně. Druhá videosekvence byl zachycen jednoduchý pohyb dopředu a zpátky, zabíraný pomocí webové kamery Logitech QuickCam Deluxe for Notebooks. Tuto sekvenci zvládla jak metoda PTAM, tak i metoda MonoSLAM. Videosekvenci 3 zabíranou taktéž webovou kamerou obě metody s problémy zvládly. Metoda PTAM vlivem texturního charakteru povrchu posunula počátek o několik centimetrů ve směru pohybu, ale zbytek sekvence zvládla bez problémů. Metoda MonoSLAM neměla problém se začátkem videosekvence, ale ztratila se při rotaci kamery. Videosekvenci číslo čtyři zvládly obě metody bez problémů.

Z výsledků jsou patrné výhody metody MonoSLAM při použití na texturních materiálech. Tyto textury ale musí být od sebe relativně dobře oddělené. Jestliže metoda MonoSLAM narazí na význačný bod vytvořený z hrany knížky nebo drátu, má tendenci jakoby klouzat po této hraně a tím rozbíjet celou mapu.

7.8 Pohyblivé a měnící se objekty ve scéně

V reálném prostředí se objekty ve scéně mění nebo pohybují. Z tohoto důvodu by měly být metody odolné vůči menším změnám ve snímku. Metody by měly být schopny rozeznat body, které jsou důvěryhodné a je možno pomocí nich určovat polohu kamery, a body, které buď nasnímaly omylem, nebo se nepodařilo při jejich začleňování do mapy správně určit jejich polohu. Metody budou testovány na pohybující se stíny a celé objekty. Dále bude proveden test na statické, ale měnící se objekty, např. TV. Testy byly prováděny na videosekvencích *ObyvakTVProblem37* a *Moved*.

Výsledky tohoto pod testu jsou uvedeny v tabulce (7.7). Výsledky statické kamery, která zabírá pohyblivé objekty budou popsány ve formě textu, protože testovací metoda je v tomto případě nevyhovující. V případě obou metod je tento problém uspokojivě vyřešen. V případě 10%-30% měnící se scény metody na změnu téměř nereagují a správně vyhodnocují pohyblivé body jako outliers. Metoda MonoSLAM reaguje na pohyblivé objekty dobře v případě, že je Kalman filtr ustálen. Pokud Kalman filtr předpokládá pohyb může i 10% změna významných bodů, podle kterých se metoda orientuje, způsobit velkou chybu ve výpočtu trajektorie kamery. Při překročení více jak 70% měnící se scény, metoda MonoSLAM ze své podstaty zkolabuje, avšak metoda PTAM se chová, jakoby statická scéna byla pohyblivá, a při pohybu objektu se pohybuje i poloha kamery. Tuto vlastnost metody PTAM je možné využít při uzamčení přidávání dalších snímků do mapy pro trasování objektů [37].

Kamera	Předměty	PTAM	MonoSLAM
Pohyblivá	Měnící se	1.37	5.07
Statická	Pohyblivé	Hodnoceno v textu komentující tabulku	

Tabulka 7.7: Výsledek testu pohyblivých objektů velikost chyby od referencí metody.

Z výsledků je patrné, že metoda PTAM má i v tomto případě přesnější výsledky. Nutno dodat, že přesnější offline metoda SynthEyes měla při výpočtu statické scény s pohyblivými objekty značné problémy, výpočet pohybu kamery pomocí této metody se v jejím případě několikanásobně prodloužil.

7.9 Vliv náhodných výpadků snímků

V této podkapitole budou uvedeny výsledky testů ukazujících vlastnosti metod při výpadcích snímků z videosekvence. Různé oblasti výpadku snímků jsou od sebe vzdáleny o definovanou minimální vzdálenost a jejich délka je maximálně určité velikosti. Aby bylo možné srovnávat jednotlivé sekvence výpadků, jsou vypadlé snímky nahrazeny posledním snímkem před výpadkem. Tento způsob výpadků má jako vedlejší účinek pokažení predikce polohy kamery při dalším snímku. V některých případech může pokažení predikce metodě zlepšit výsledky, naopak při lineárně postupu kamery může výsledky zhoršit. Testy byly prováděny na videosekvenci *obyvak4*.

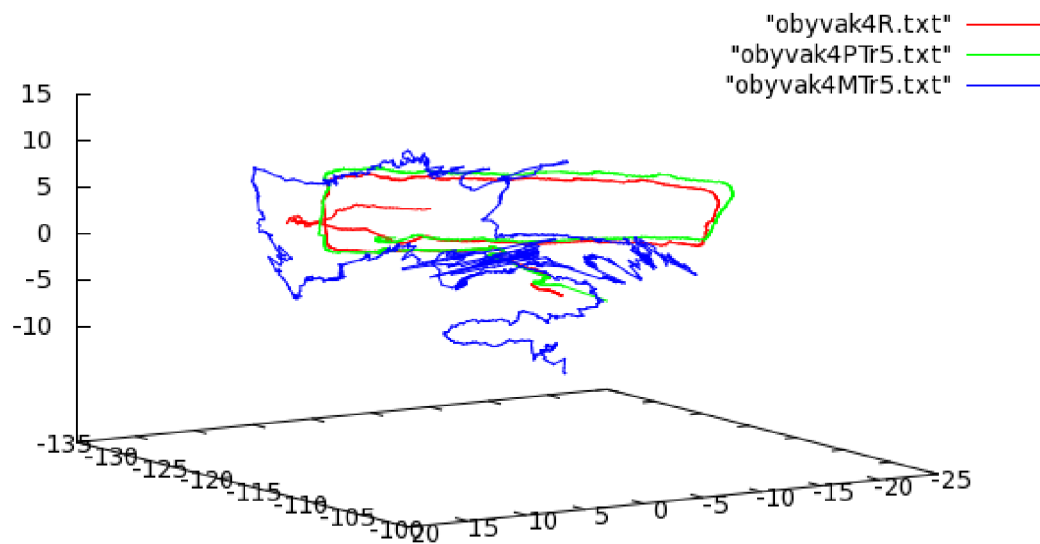
Jak je z tabulky (7.8) patrné, metoda PTAM nemá při rozumné videosekvenci s výpadky snímků žádný velký problém. Problémy nastaly až u 30 vypadlých snímcích v řadě. Navíc metoda PTAM nepotřebovala algoritmus pro znovu-nalezení pozice kamery. Naopak metoda MonoSLAM (Scenelib) nedopadla moc dobře. Při výpadku 5 po sobě jsooucích snímků jen částečně ztrácela přesnost a výsledná trajektorie kamery vypadala podobně jako originál. Jen z míst velké části

výpadků vedla chybným směrem. Při větších výpadcích než 5 snímků se metoda MonoSLAM ztrácela a výsledky v tabulce spíše říkají, jak moc se náhodná trajektorie podobná trajektorii naměřené pomocí přesnější offline metody.

Velikost výpadku	PTAM chyba	MonoSLAM chyba
5	0.598084	7.607187
10	0.629930	25.588301
20	0.567258	14.722448
30	3.436049	10.006367

Tabulka 7.8: Vliv velikosti výpadku snímků na velikost chyby metody.

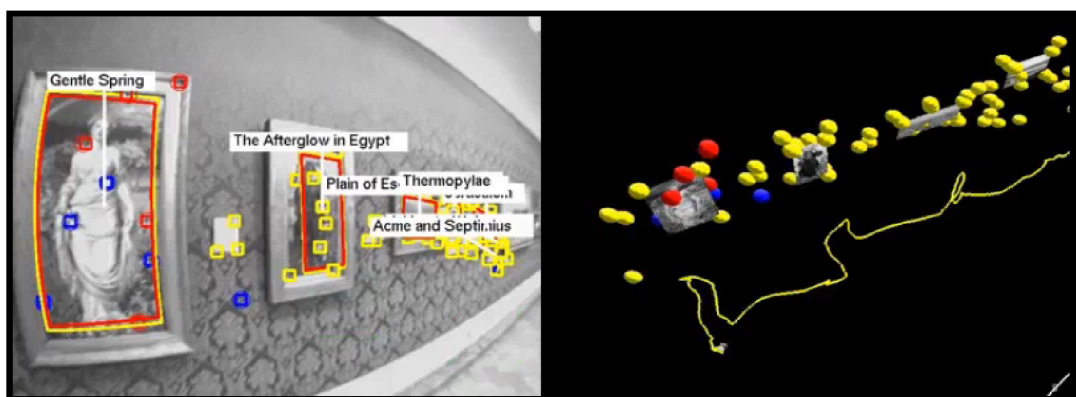
Jak je vidět na obrázku (7.7), metoda MonoSLAM se při výpadcích o velikosti 5 drží částečný tvar trajektorie, ale v místech výpadků má značné problémy.



Obrázek 7.7: Trajektorie drah generovaných metodami při výpadcích délky 5.

8 Pokročilejší postupy a návrh demo-aplikací

Pokročilejšími postupy jsou zde míněny aplikace postavené na metodách SLAM. Ve většině případů jde o rozšíření typu klasifikace objektů, jako je např. detekce obrazů v muzeu. Při klasifikačních metodách se kamera pohybuje v prostoru, kde se mají vyhledávat určité objekty. V průběhu pohybu si systém vytváří model prostředí. Pokud kamera narazí na hledaný objekt, může uložit jeho polohu a texturu do modelu. Poté je například před nalezeným objektem zobrazen text s popisem objektu. Oproti běžné klasifikaci má klasifikace spojená s metodami SLAM výhodu v podobě možnosti ukazovat na objekt, i když jeho přítomnost již není možné detekovat běžnou klasifikací. Příklad je uveden na obrázku (8.1).



Obrázek 8.1: MonoSLAM ve spojení s klasifikací objektů. Přejato z [38].

Dále je možné pomocí metod SLAM realizovat převod geometrie reálného prostředí do reprezentace v počítači. Příklad je uveden na obrázku (8.2).



Obrázek 8.2: Rekonstrukce prostoru na základě metody MonoSLAM. Převzato z [39].

Je také možné rozšířit možnosti herního průmyslu. Na základě metody PTAM již je kupříkladu implementováno demo jisté akční hry. Ta je inspirovaná filmem StarWars, kde Darth Vader střílí mimozemšťany. Zaměřovač Darth Vadera je centrován pomocí středu kamery.

8.1 Pomocník nevidomých lidí

Jednalo by se o aplikaci, která by umožnila nevidomým lidem vnímat vizuální informace ze svého okolí. Pro reálné nasazení by bylo třeba portovat metodu PTAM do mobilního zařízení, jako jsou novodobé smartphony (iphone, android).



Obrázek 8.3: Braillov monitor. Obrázek převzat z [40].

Použití by probíhalo tímto způsobem. Nevidomý by u sebe měl telefon. V případě potíží by spustil (např. na hlasový příkaz) aplikaci. Aplikaci by na-inicializoval, jak je to popsáno u metody PTAM. Mobilním telefonem by se „rozhlédl“ po okolí a přístroj by mu akustickou formou naznačil, zda jsou v blízkosti nějaké překážky (výkop, velký patník, silnice), případně by bylo možné určit i jejich vzdálenost. Pokud by mobilní zařízení mělo i Braillov monitor (8.3), informace by na něm mohla být zobrazena.

Dále by bylo možné aplikaci rozšířit o klasifikační algoritmus, který by umožňoval lokalizovat věci v okolí nevidomého. Nevidomý by si hlasovým příkazem zadal, že chce najít kupříkladu konvičku na čaj. Po prohlédnutí okolí, by systém určil lokalizaci konvičky a informoval nevidomého o její vzdálenosti a poloze.

8.2 Navigátor autonomních průmyslových vozíků

V dnešní době existují autonomní transportní systémy, které umožňují přepravit suť nebo jiný náklad na určené místo bez zásahu člověka. Navigace takových systému by se dala rozšířit o kamerový systém. Po naložení nákladu by se určila velikost nákladu. Vozík by jel po určené trase a při zjištění, že se do určitého prostoru celý náklad nevejde, by zavolal obsluhu, nebo v databázi hledal novou cestu pro dopravení materiálů.

8.3 Návrh interiérů

Uživatel by pomocí aplikace namapoval místnost a po kliknutí na stěnu by se konkrétní stěna překreslila stěnou virtuální. Tímto způsobem by uživatel byl schopen virtuálně vyčistit pokoj a umístit do něj virtuální nábytek.

8.4 Hra v reálném prostředí

Mohlo by se jednat o strategii nebo akční hru. Po spuštění hry by se nainicializovala hrací plocha, aby byl pohyb plynulý a přesný. Ovládání hry by bylo vhodné realizovat pomocí touch screenu. Pohyb po herním prostředí by byl prováděn pohybováním s kamerou. Dokonce by bylo možné, jak ukazuje článek [41], v reálném čase z velkou přesností zrekonstruovat 3D prostor, ve kterém by hra probíhala. Výsledky uvedené ve výše zmiňovaném článku jsou výborné a otevírají PTAM metodám velké spektrum dalšího využití.

8.5 Prohlížení 3D modelů v prostoru

Uživatelem namodelovaný model by byl promítán v prostoru. Uživatel by se kolem něho mohl procházet, nahlížet na něj ze všech stran a případně ho upravovat. Aplikace by byla vhodná pro umělecky zaměřené profese, kdy není vyžadována přespřílišná přesnost, ale je třeba si model co nejlépe představit.

9 Závěr

V rámci této práce byly prostudovány metody dnes používané pro orientaci v prostoru pouze ze snímků jedné kamery. Jednalo se jak o metody pracující v reálném čase, tak o metody pracující nad videosekvencemi v offline režimu. Dále zde byly popsány dnes užívané základní stavební prvky, které byly navíc, v rámci popisu offline metod, zařazeny do kontextu zkoumaných metod realizujících zjišťování orientace kamery v prostoru. Nedílnou součástí této práce bylo i podrobné rozebrání metod pracujících v reálném čase a jejich vzájemné porovnání. To proběhlo jak na bázi teoretického srovnání, tak v rámci praktických pokusů provedených na srovnávaných metodách. Konkrétně se jednalo o zástupce dvou hlavních proudů, a to MonoSLAM (Real-Time Single Camera SLAM) a PTAM (Parallel Tracking and Mapping). V poslední části práce byly navrženy aplikace postavené na metodách orientace kamery v prostoru v reálném čase.

V rámci srovnávání obou metod na základě testu, byla zjištěna kvalitativní převaha metody PTAM. Ta ve většině případů vykazovala větší přesnost a odolnost. Větší odolnosti dosahovala jak v rámci reakce na problematickou videosekvenci, tak ve funkci se špatně nastavenými kalibračními parametry. Jediný test, ve kterém metoda PTAM měla horší výsledky, se týkal rozmazání obrazu vlivem pohybu. Výsledek zjištěný v kapitole „Srovnání metod na základě pokusů“, mohl být ovlivněn stářím poslední dostupné implementace metody MonoSLAM publikované tvůrcem metody.

V průběhu práce jsem narazil na velké množství obtížných témat, které bylo často nezbytné velmi podrobně nastudovat. Jednalo se zejména o matematické problémy, přesahující rámec probíraného učebního materiálu. Implementace obou metod byly na vysoké programátorské úrovni, avšak s metodou MonoSLAM v podání Scenelib byl často problém, kvůli její zastaralé implementaci. Bylo nutné upravovat konfigurační soubory a instalovat starší, již běžně nepoužívaný, software. Aby bylo možné mezi sebou obě testované metody porovnávat, muselo jim být přidáno rozhraní pro práci se stejným daty.

Metody orientace v prostoru pomocí snímků jedné kamery pracující v reálném čase, mají do budoucna velký potenciál. Obzvláště je-li brán v úvahu velký nárůst výkonnosti mobilních zařízení a obrovské úsilí, které je do vývoje nových metod vkládáno. Nad metodami orientace v prostoru je možné realizovat i metody rekonstrukce povrchu, jejichž spojením vzniká mocný nástroj umožňující kooperaci reálného světa a rozšířené reality [41]. V dnešní době jsou ale tyto metody teprve v počátcích a bude potřeba ještě mnoho úsilí, než bude možné jejich reálné nasazení v běžném životě.

Literatura

- [1] A. J. Davidson, I. D. Reid, Nicholas D. Molton a O. Stasse, MonoSLAM: Real-Time Single Camera SLAM, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 2007
- [2] G. Klein, D. Murray, Parallel Tracking and Mapping for Small AR Workspaces, Active Vision Laboratory, 2007
- [3] P. S. Maybeck, Stochastic models, estimation, and control VOLUME 1, ACADEMIC PRESS, 1979
- [4] Doc. Dr. Ing. J. Černocký, Klasifikace a rozpoznávání, Extrakce příznaků, FIT VUT BRNO, 2009
- [5] S. J. Julier, J. K. Uhlmann, A New Extension of the Kalman Filter to Nonlinear Systems, The Robotics Research Group, Department of Engineering Science, The University of Oxford, 1997
- [6] J. Shi a C. Tomasi, Good features to track, Cornell U., 1994
- [7] K. G. Derpanis, The Harris Corner Detector, 2004,
http://www.cse.yorku.ca/~kosta/CompVis_Notes/harris_detector.pdf
- [8] G. Arfken, Mathematical Methods for Physicists, 3rd ed, Orlando, FL: Academic Press, 1985
- [9] T. Svoboda, Kanade-Lucas-Tomasi Tracking (KLT) tracker, 2010,
http://cmp.felk.cvut.cz/cmp/courses/Y33ROV/Y33ROV_ZS20092010/Lectures/Motion/klt.pdf
- [10] Š. Voráčová, Sborník 24. konference Geometrie a počítačové grafiky, Barborka, 2004
- [11] H. Stewénius, Ch. Engels, D. Nistér, Recent Developments on Direct Relative Orientation, ISPRS J. of Photogrammetry and Remote Sensing, 2006
- [12] Z. Zhang, Determining the Epipolar Geometry and its Uncertainty: A Review, Kluwer Academic Publishers, 1997
- [13] H. Stewénius, C. Engels a D. Nistér, Recent developments on direct relative orientation, ISPRS Journal of Photogrammetry and Remote Sensing, 2006
- [14] R. I. Hartley a A. Zisserman, Multiple View Geometry in Computer Vision, 2003
- [15] M. I. A. Lourakis a A. A. Argyros, A Software Package for Generic Sparse Bundle Adjustment, ACM Transactions on Mathematical Software, 2009
- [16] M. Trajtkovic a M. Hedley, Fast corner detection, Image and Vision Computing, 1998
- [17] S. M. Smith a J. M. Brady, SUSAN—A New Approach to Low Level Image Processing, Kluwer Academic Publishers, 1997
- [18] S. Holmes, G. Klein a D. W. Murray, A Square Root Unscented Kalman Filter for visual monoSLAM, Conf on Robotics and Automation, 2008
- [19] M. Sharam, R. Voyles, T. Myers, R. Bodor, O. Masoud, Automatic Euclidean reconstruction for turn-table sequences by indirect epipolar search between pairs of views, sciencedirect, 2005

- [20] A.J. Davison, Real-Time Simultaneous Localisation and Mapping with a Single Camera, Proc. Ninth Int'l Conf. Computer Vision, 2003
- [21] A.J. Davison, D.W. Murray, Mobile Robot Localisation Using Active Vision, Proc. Fifth European Conf. Computer Vision, 1998
- [22] J. Shi and C. Tomasi, Good Features to Track, Proc. 10th Int'l Conf. Computer Vision, 1994
- [23] A.J. Davison, Active Search for Real-Time Vision, Proc 10th Int'l Conf. Computer Vision, 2005
- [24] B. Triggs, P. Mc Lauchlan, R. Hartley a A. Sitzgibbon, Bundle Adjustment - A Modern Synthesis, Springer-Verlag London, UK, 1999
- [25] M. K. Chandraker, Bundle Adjustment Sparse Estimation in Multi-View Geometry, 2004, <http://lear.inrialpes.fr/pubs/2000/TMHF00/Triggs-va99.pdf>
- [26] M. Pollefeys, Bundle Adjustment, 2002, <http://www.cs.unc.edu/~marc/tutorial/node159.html>
- [27] E. Rosten a T. Drummond, Machine learning for high-speed corner detection, In Proc. 9th European Conference on Computer Vision, 2006
- [28] J.M. Selig, Lie Groups and Lie Algebras in Robotic, Nato Advanced Study Institute, 2003
- [29] F. Devernay a O.D. Faugeras, Straight lines have to be straight, Machine Vision and Application, 2001
- [30] V. Patanavijit a S. Jitapunkul, A Tukey's Biweight Bayesian Approach for A Robust Iterative SRR of Image Sequences, IEEE, 2007
- [31] B. Williams, G. Klein a I. Reid, Real-Time SLAM Relocalisation, University of Oxford, UK, 2007
- [32] G. Klein, Keyframe-based SLAM for hand-held Augmented Reality, 2008,
- [33] B. Williams, G. Klein a J. Reid, Real-Time SLAM Relocalisation, University of Oxford, UK, 2007
- [34] G. Klein a D. Murray, Improving the Agility of Keyframe-Based SLAM, Active Vision Laboratory, University of Oxford, UK, 2008
- [35] G. Klein, Georg Klein Home Page, 2009, <http://www.robots.ox.ac.uk/~gk/PTAM/>
- [36] University of Hannover, Voodoo Camera Tracker, 2009, <http://www.digilab.uni-hannover.de/index.html>
- [37] R. Castle, Simultaneous Recognition, Localization and Mapping for Wearable Visual Robots, The University of Oxford, 2009
- [38] R. Castle, Object tracking with monoSLAM: Ashmolean Gallery, 2008, <http://www.youtube.com/watch?v=saUE7JHU3P0>
- [39] R. Castle, G Klein, D. W. Murray, Object tracking with monoSLAM: Ashmolean Gallery, 2009, <http://www.youtube.com/watch?v=saUE7JHU3P0>
- [40] BAUM, BAUM Retec AG, 2010, <http://www.baum.de/cms/en/>

[41] R. A. Newcombe a A. J. Davidson, Live Dense Reconstruction with a Single Moving Camera, University of Oxford, 2010

10 Seznam příloh

Příloha 1. Obsah elektronického média

Příloha 2. Popis videosekvencí

Příloha 1. Obsah elektronického média

Součástí práce je i elektronické médium obsahující následující obsah

/Dokumenty

DP-Zupka.odt práce, dokument v OpenOffice
DP-Zupka.pdf práce, dokument v pdf

/Src

zdrojové kódy práce

/MatrixSyncc

program na synchronizaci 3D datových linií

main.cpp

Makefile

README

dokumentace k překladu

/Metody

PTAM

upravená verze PTAM pro práci s obrázkovou videosekvencí.

MonoSlam

testovaná verze MonoSlamu (Scenelib)

!!!

soubor obsahující nutnou změnu pro úspěšný překlad

README

standardní popis překladu

/Scripts

skripty pro zpracování videa

blur.py

rozmazání videosekvence

blurrnd.py

náhodné rozmazání videosekvence

camrecord.py

nahrání videosekvence z USB kamery

convert.py

konverze videosekvence mezi formáty

pathdiferent.py

porovnání dvou videosekvencí

README

dokumentace k použití skriptů

/VideoSeq

adresář s výchozími videosekvencemi

/Plakat

DP-Zupka.pdf plakát práce

Příloha 2. Popis videosekvencí

V této příloze budou popsány vlastnosti videosekvencí použitých na porovnávání metod. Budou zde uvedeny i postupy, kterými videosekvence procházely při testování.

Pro testování vlivu nastavení metod byla použita videosekvence obyvak4. U videosekvencí nebylo nutné dělat žádné změny. Byly jen použity skripty pro zpracování dat MatrixSyncc a pathdiferent. Ve videosekvenci je obsažen plynulý komplexní pohyb kamery.

Pro testování vlivu rozlohy prostředí nebyla udělána žádná změna ve videu dvurside21. Byly použity skripty pro zpracování dat MatrixSyncc a pathdiferent. Videosekvence obsahuje plynulý pohyb jedním směrem.

Pro testování vlivu snímací soustavy byly použity videosekvence dvurline21 a dvurline37. Jde o podobné videosekvence, které byly zabírány různými kamerami. Pro testování rozmazání scény byla použita videosekvence blur329. Na kterou byly aplikovány pohybové filtry pomocí skriptu blur.py. A to následujícím způsobem:

```
python ./blur.py 1 100
python ./blur.py 2 100
python ./blur.py 3 100
python ./blur.py 4 100
python ./blur.py 5 100
python ./blur.py 6 100
a
python ./blurrnd.py 8 100 50
python ./blurrnd.py 12 100 50
```

Vzniklé videosekvence byly porovnány s referenční pomocí scriptů MatrixSyncc a pathdiferent.

Pro test barevných vlastností byli použity různé videosekvence texture, texture2, texture3, texture4. Na ty nebyl aplikován žádný předpřípravný script. Byly opět použity jen MatrixSyncc a pathdiferent.

Na test „Pohyblivých a měnících se objektů ve scéně“ byly použity videosekvence ObyvakTVProblem37 a Moved. Byly na ně aplikovány jen MatrixSyncc a pathdiferent. Videosekvence ObyvakTVProblem37 a Moved byly vybrány protože obsahují specifický druh pohybu, který byl testován.

Pro test vlivů náhodných výpadků snímku ve videosekvenci bylo pro předzpracování použito scriptů remover.py následujícím způsobem:

```
python ./remover.py 5 100
python ./remover.py 10 100
```

```
python ./remover.py 20 100
```

```
python ./remover.py 30 100
```

Vzniklé videosekvence se nechaly detekovat testovanými metodami a na zpracování výsledku bylo opět použito scriptů MatrixSyncc a pathdiferent. Videosekvence byla vybrána, jelikož byla využita pro kalibraci a obsahuje komplexní pohyb.