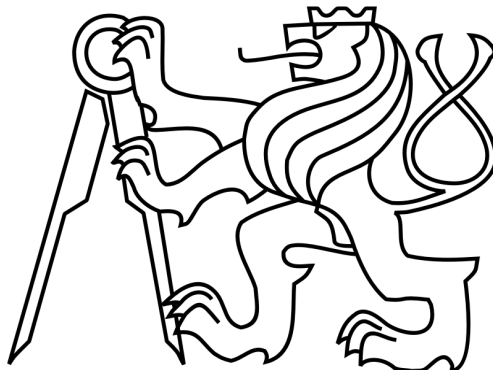


České vysoké učení technické v Praze
Fakulta elektrotechniká

**Trojdimenzionální rekonstrukce scény,
která se postupně proměňuje**

Agáta Pokorná

Bakalářská práce



vedoucí práce: Ing. Tomáš Pajdla, Ph.D.

Praha, květen 2012

Trojdimenzionální rekonstrukce scény, která se postupně proměňuje

V závislosti na vlastnostech vstupních dat bylo pro detekci změn ve 2D snímcích v minulosti navrženo několik algoritmů. Vybrané z nich tato bakalářská práce v teoretické části popisuje. Dále navrhuje optimalizaci rekonstrukce 3D modelu prostředí s pohybujícími se objekty z dat vybraných z všeobecně dostupného videozáznamu. Jádrem navrženého řešení je nalezení parametrů, při nichž pohyblivé objekty nejsou zrekonstruovány nebo jsou co nejvíce minimalizovány. Druhá část bakalářské práce se zabývá detekcí pohybu a oddělením pohyblivého objektu na vlastních experimentálních datech.

3D reconstruction of a gradually changing scene

There are several algorithms dealing with detection of scene changes which at first sight differ on type of input data. Some of them are described in the theoretical part of this thesis. Next task solves optimization of 3D scene reconstruction containing moving objects. Input data are selected from widely known TV series. The goal of proposed solution is to find parameters which minimize or don't reconstruct moving objects. The second part of this thesis deals with motion detection and separate a moving object using our experimental data.

Prohlášení autora práce

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

V Praze dne 25.5.2012


podpis

Poděkování

Chtěla bych poděkovat vedoucímu mé bakalářské práce Tomáši Pajdlovi za množství podnětných nápadů a neustálou vstřícnost. Také bych chtěla poděkovat Kataríně Pokorné a Tomáši Bilkovi za jejich všestrannou podporu při studiu a psaní této práce.

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: Ing.arch. Agáta Pokorná

Studijní program: Elektrotechnika a informatika (bakalářský), strukturovaný

Obor: Kybernetika a měření

Název tématu: Trojdimenzionální rekonstrukce scény, která se postupně proměňuje

Pokyny pro vypracování:

1. Prostudujte literaturu o trojdimenzionální rekonstrukci měnící se scény.
2. Na příkladu scény, kterou vytvoříte, experimentálně studujte, jak ovlivňuje postupná změna scény její rekonstrukci.
3. Formulujte problém rekonstrukce scény, která se pomalu proměňuje.
4. Experimentálně ověřte, že formulace má smysl na předchozím příkladu scény a na dalším příkladu z jiného zdroje.

Seznam odborné literatury:

- [1] Hartley, R. and Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University, Cambridge, 2nd edition, 2003.
- [2] Leoputra, W. S.; Tan, T.; Venkatesh, S.: A Unified 2D-3D Video Scene Change Detection Framework for Mobile Camera Platforms. 11th International Conference on Control Automation Robotics & Vision (ICARCV), 2010.
- [3] Ozden, K. E.; Cornelis, K.; van Gool, L.: A Unified 2D-3D Video Scene Change Detection Framework for Mobile Camera Platforms. Proceedings of the IEEE 12th Signal Processing and Communications Applications Conference, 2004.
- [4] Imre, E.; Guillemaut, J. Y.; Hilton, A.: Calibration of Nodal and Free-Moving Cameras in Dynamic Scenes for Post-Production. In Proc. International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT 2011), pages 260–267, 2011.

Vedoucí bakalářské práce: Ing. Tomáš Pajdla, Ph.D.

Platnost zadání: do konce letního semestru 2011/2012


prof. Ing. Vladimír Mařík, DrSc.
vedoucí katedry




prof. Ing. Pavel Ripka, CSc.
děkan

Obsah

1.	Úvod	1
1.1.	Cíle bakalářské práce	1
1.2.	Struktura bakalářské práce	1
2.	Definice problému	2
3.	Stav problematiky ve světě	4
3.1.	Detekce změn ve scéně při snímání mobilní kamerou	4
3.2.	Segmentace, modelování a porovnávání videozáznamů zachycující několik pohybujících se objektů	5
3.3.	Kompozice obrazu pro náhodně se objevující objekty ve scéně	5
3.4.	Modeling the world	6
3.4.1.	Pseudokód algoritmu	7
4.	Vstupní 2D data	9
4.1.	Vstupní data pro optimalizaci rekonstrukce z videozáznamu seriálu Friends	9
4.2.	Tvorba vstupních dat pro analýzu pohybu	10
5.	Algoritmus optimalizace rekonstrukce	11
5.1.	Hrubý popis algoritmu	11
5.2.	Analýza vstupních dat	11
5.2.1.	Histogramy stabilních bodů	11
5.2.2.	Kontrola výskytu stabilních bodů na obrázcích	12
5.3.	Metody výběru stabilních bodů	13
5.3.1.	Výběr bodů s nejvíce korespondencemi	13
5.3.2.	Výběr bodů v sekvenci	13
6.	Získané výsledky optimalizačního algoritmu	14
6.1.	Zvolené parametry	14
6.2.	Grafické výstupy jednotlivých výsledků optimalizací	15
7.	Algoritmus analýzy experimentálních dat	17
7.1.	Princip lokace pohybu	17
7.2.	Příprava vstupních dat	18
7.3.	Zpracování dat a ověření experimentu	18
8.	Závěr	21
	Seznam použitých zdrojů	22

Přílohy.....	23
A Dokumentace k programu ProgOptimization	24
B Dokumentace k programu analýzy experimentálních dat	27
C Příklady spouštěcích souborů.....	29
D Použitý software.....	30
E Obsah CD	30

Seznam obrázků

Obrázek 1 : souřadné soustavy při hledání pozice kamery (převzato z [1]).....	2
Obrázek 2 : homografie (převzato z [1], upraveno)	3
Obrázek 3 : ukázka výsledné detekce nestabilních objektů (převzato z [2]).....	4
Obrázek 4 : geometrická interpretace rektifikace patche (převzato z [3])	5
Obrázek 5 : výsledky rekonstrukce na datech z filmového záznamu (převzato z [3]).....	5
Obrázek 6 : zleva doprava: vstupní obrázek, plocha vstupního obrázku, ke kterému nebyly nalezeny odpovídající části v databázi a další tři fotografie z databáze, které byly použity pro získání podobných částí (převzato z [4])	6
Obrázek 7 : (zleva doprava) jeden ze vstupních obrázků, 3D model, connectivity graph.....	7
Obrázek 8 : příklad vstupních dat - seriál Friends	9
Obrázek 9 : příklad vstupních dat - experiment	10
Obrázek 10 : výstup funkce TractLenHist, nastavený parametr MinTrackLen 65	12
Obrázek 11 : vykreslení vybraného vstupního obrázku se zákresem stabilních bodů(•), vlevo pro MinTrackLen 60, vpravo pro MinTrackLen 65	12
Obrázek 12 : vlevo: 3D vizualizace s nastavením MinTrackLen 60, vpravo: 3D vizualizace s nastavením MinTrackLen 65. Šipka ukazuje na vybrané body(•) na pohyblivém objektu, které jsou pro optimalizaci nežádoucí.	13
Obrázek 13 : rekonstrukce final_2263 po optimalizaci z nadhledu	15
Obrázek 14 : nejlepší výsledek rekonstrukce bez optimalizace (work_1945).....	15
Obrázek 15 : dataset 757 / work_2249 / final_2263.....	16
Obrázek 16 : dataset 758 / work_2250 / final_2275.....	16
Obrázek 17 : dataset 700 / work_2034 / final_2271.....	16
Obrázek 18 : princip výběru tracků zachycující pohyblivý objekt	17
Obrázek 19 : rekonstrukce scény ze společného datasetu, šipka ukazuje na posunutý objekt, který se zrekonstruoval ve dvou různých pozicích.....	18
Obrázek 20 : vizualizace rozdělení bodů do skupin podle náležitosti k sekvenci.....	18
Obrázek 21 : vyjmuté body, které nesousedí s více než 10% bodů z množiny „společné“:.....	19
Obrázek 22 : výřezy okolí feature pointu (tj. +) nezrekonstruovaného tracku s chybou $\leq 3\text{px}$	19
Obrázek 23 : výřezy okolí feature pointu (tj. +) nezrekonstruovaného tracku s chybou $\leq 3\text{px}$	20
Obrázek 24 : uživatelské rozhraní programu ProgOptimizace	24
Obrázek 25 : : uživatelské rozhraní programu ProgAnalyze.....	27

Seznam tabulek

Tabulka 1 : výsledky optimalizace v přehledu	14
Tabulka 2 : počty výskytu tracků	19

1. Úvod

Jednou z úloh počítačového vidění je porozumět obsahu digitálního obrazu, což je například sledování či rozpoznání objektů nebo detekce pohybu v obraze. Právě detekcí a odstraněním pohybujících se objektů ze scény se zabývá tato práce. Přestože na dané téma vzniklo mnoho studií, zatím žádná neposkytuje dostatečné řešení.

Jedním z přístupů k rekonstrukci 3D scény je vytvoření modelu z 2D vstupních dat zachycující vybranou scénu z několika různých úhlů. Typickým omezením naší práce je, že neznáme parametry kamer. Proto jsou veškeré uvedené rekonstrukce vytvářeny metodou nalezení vzájemně si odpovídajících bodů na jednotlivých snímcích.

Tato rekonstrukce 3D scény je v současnosti založena na několika fázích. Zkráceně se dají popsat jako hledání významných korespondencí ve vstupních datech, výpočet vztahu mezi obrazy a postupné nalezení 3D souřadnic bodů. Následuje rekonstrukce povrchů zachycených objektů. Známé metody realizace tohoto postupu se značně liší v závislosti na požadovaném výsledku.

1.1. Cíle bakalářské práce

Hlavním cílem této bakalářské práce je formulovat problém rekonstrukce scény, která se pomalu proměňuje a následně formulaci ověřit na datech. Proto je potřeba provést analýzu scény s pohyblivým objektem na známých datech a pokusit se nalézt způsob, jak odfiltrovat pohybující se objekty z výsledné rekonstrukce scény. Pro dosažení tohoto hlavního cíle bylo dále specifikováno několik vedlejších úkolů:

1. Prostudovat dostupné metody zabývající se detekcí pohybu. Problém detekce změny ve scéně je řešen již několika různými způsoby.
2. Vytvořit vlastní experimentální scénu, analyzovat ji a ověřit na ní formulaci problému rekonstrukce pohybu ve scéně.
3. Vytvořit program, který by dovedl analyzovat a optimalizovat data obecné scény tak, aby byly odfiltrovány pohybující se objekty.

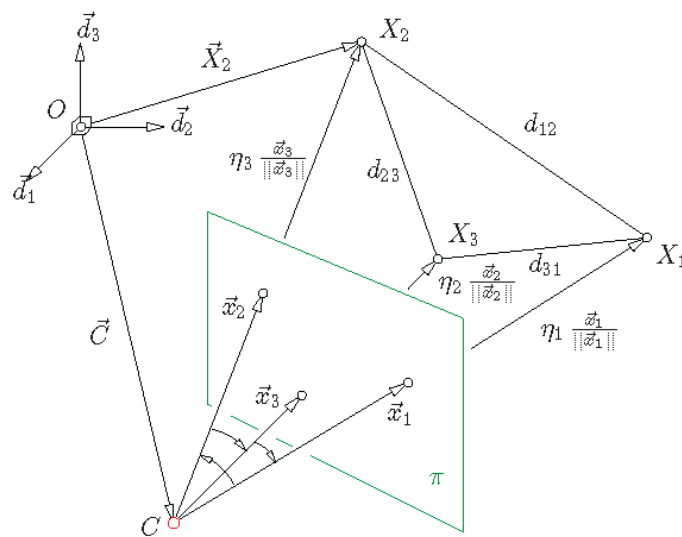
1.2. Struktura bakalářské práce

V kapitole 2 je podrobně definován problém, kterým se tato práce zabývá. Kapitola 3 seznamuje s několika úspěšnými přístupy k detekci pohybu ve scéně. Seznámení je provedeno popsáním principů, použitých důležitých algoritmů a s ukázkami dosažených výsledků. Kapitola 4 popisuje vstupní data, včetně jejich získání a charakteristik. Kapitola 5 se zabývá algoritmem analyzující a optimalizující rekonstrukci nad obecnými daty. V následující kapitole 6 jsou uvedeny grafické výsledky optimalizačního algoritmu. Kapitola 7 popisuje analýzu pohybujícího se objektu ve statické scéně spolu s ověřením předpokládaných vlastností pohybujících se objektů ve scéně na experimentálních datech. Poslední kapitola 8 shrnuje všechny dosažené výsledky a navrhuje další vývoj.

2. Definice problému

Základní úlohou při hledání 3D souřadnic bodů z 2D obrazů je nalezení parametrů kamery (tj. fotoaparátu nebo pozice videokamery), která body snímala. Vstupní informací jsou 3D souřadnice bodů $[x,y,z]$ a jim odpovídající 2D souřadnice $[u,v]$ ve vybraném obrazu. Pokud známe kalibraci kamery (parametry K,R,f (měřítko, rotaci, ohniskovou vzdálenost)) střed kamery lze nalézt ze 3 bodů - jejich známých souřadnic ve 3D a odpovídajících souřadnic ve 2D prostoru, viz. obr. 1. Následující principy a výpočetní postupy jsou převzaty z [1].

Obrázek 1 : souřadné soustavy při hledání pozice kamery (převzato z [1])



Pokud však kalibraci nemáme možnost zjistit, je postup nalezení pozice kamery odlišný. Předpokládejme, že známe 3D souřadnice bodů a jejich odpovídající průměty ve 2D. Pak hledáme projekční matici P splňující vztah:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = Q \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad \text{kde} \quad Q = \varepsilon P$$

Pro kameru s neznámou kalibrací je podmínkou znát alespoň 6 bodů a jejich souřadnice. To zdůvodňuje následující rovnice, kterou obdržíme úpravou předchozího vztahu. Pokud chceme najít její řešení, musí obsahovat alespoň 11 lineárně nezávislých řádků. Každý bod přidá do rovnice 2 řádky.

$$\begin{bmatrix} x & y & z & 1 & 0 & 0 & 0 & 0 & -ux & -uy & -uz & -u \\ 0 & 0 & 0 & 0 & x & y & z & 1 & -vx & -vy & -vz & -v \end{bmatrix} q = 0 \quad \text{kde} \quad Q = \begin{bmatrix} q_1^T \\ q_2^T \\ q_3^T \end{bmatrix}$$

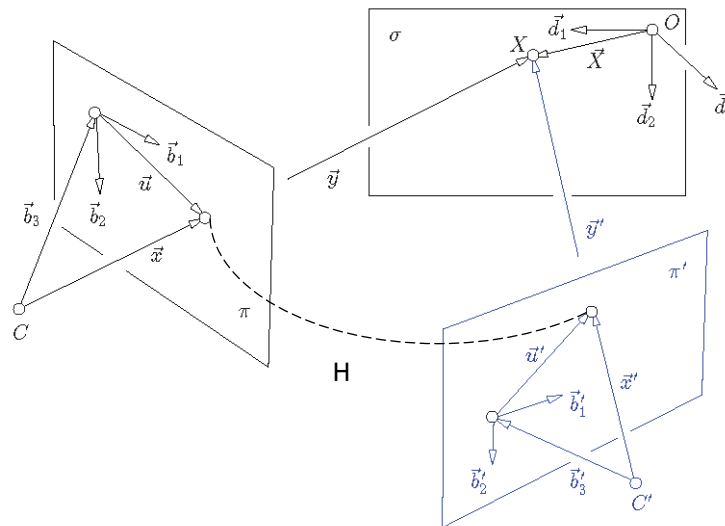
$$M q = 0$$

Nalezení matice P umožňuje získat chybějící parametry: měřítko, rotaci, ohniskovou vzdálenost a polohu středu kamery.

$$P = \left[\frac{1}{f} KR \quad \left| \quad -\frac{1}{f} KR \vec{C}_s \right. \right]$$

Jestliže jsou vstupem pouze 2D obrazy a nikoliv informace o souřadnicích bodů a kalibraci kamery jako v našem případě, pak projekční matici musíme najít jiným způsobem. Pro řešení jsme použili SFM algoritmus, který využívá znalosti homografie. Homografie je transformace mezi dvojicí bodů z různých obrazů zobrazující tentýž 3D bod, viz. obr. 2.

Obrázek 2 : homografie (převzato z [1], upraveno)



Zjednodušený postup SFM algoritmu se dá popsat následujícími kroky: nalezení význačných bodů (tzv. feature points), nalezení korespondencí význačných bodů (tzv. tracky), nalezení homografie (která má největší podporu) pro iniciační pár obrázků a připojování dalších obrázků homografií k tomuto páru. Tracky, jejichž jednotlivé korespondence po aplikování homografie vykazují chybu nad stanovený práh nejsou zrekonstruovány.

SFM má i omezení, se kterými bylo cílem se vypořádat. Tracky se mohou nadetekovat i na pohybujících se nebo měnících se objektech. Algoritmus nezrekonstruuje rychle se pohybující objekty např. chodící osoby. Pokud však zachytí postavu, která sedí např. v 6 a více snímcích, je velmi pravděpodobné, že se zrekonstruuje minimálně její torzo. Pokud je ve scéně objekt, který po určitém čase změní svoji polohu - např. židle, je opět pravděpodobné, že ji SFM algoritmus zrekonstruuje v obou jejích pozicích. Je to dáno tím, že SFM algoritmus hledá významné korespondence přes všechny vstupní data. Vybraný bod zrekonstruuje, pokud najde jeho dostatečnou podporu přes více obrázků (nikoliv všechny).

3. Stav problematiky ve světě

3.1. Detekce změn ve scéně při snímání mobilní kamerou

Tento algoritmus se zabývá detekcí změn ve scéně, která byla nasnímána kamerou, připevněnou k automobilu jedoucímu ulicemi města. Většina scény je stabilní - statická - tzv. pozadí. Pohybující nebo měnící se objekty jsou označovány jako popředí a patří sem například chodci či jedoucí auta. Princip algoritmu je převzat z [2].

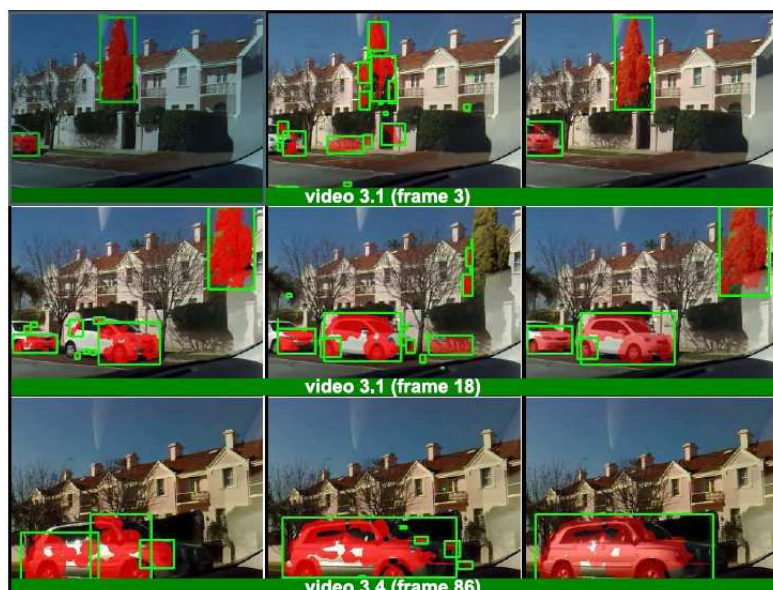
Základní části algoritmu lze rozložit na následující kroky: Po nasnímání dat se scéna rekonstruuje pomocí SFM algoritmu, získaná 3D data se zarovnají podle času, detekuje se změna ve 3D rekonstrukci přes prostorovou konzistenci a nakonec se objekty rozřadí do různých rychlostních tříd.

Výsledek SFM algoritmu je řídký oblak 3D bodů vztažených ke geometrii kamery. Výsledné rekonstrukce jsou časově seřazené podle času nasnímání. Vyniklé clusterly jsou pak mezi sebou porovnávány pomocí algoritmu SNN (Shared Nearest Neighbour), aby se získaly stejné korespondence bodů z časově různých rekonstrukcí. Dva body si odpovídají, pokud splňují následující kritéria: bod musí patřit do skupiny K-nejbližších sousedů korespondujícího bodu a obráceně, jejich vzdálenost musí ležet pod určitým prahem a nakonec oba body musí sdílet minimální počet podobných nejbližších sousedů.

Body, které nemají korespondenci jsou body popředí a šumu. V dalším kroku je šum odstraněn následovně: za šum jsou považovány body, jejichž hustota vůči ostatním bodům bez korespondence nesplňuje stanovený práh. Ve výsledku získáme body popředí.

Na závěr se rozdělí body do skupin podle rychlosti. Aby bylo možno určit rychlost objektu, musí se vypočítat rychlost snímající kamery z její pozice z SFM rekonstrukce a z časových údajů pořízení záběrů.

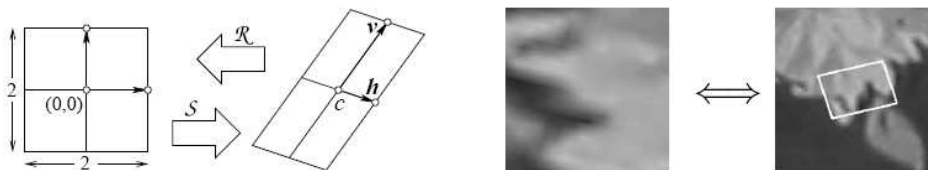
Obrázek 3 : ukázka výsledné detekce nestabilních objektů (převzato z [2])



3.2. Segmentace, modelování a porovnávání videozáznamů zachycující několik pohybujících se objektů

Základem uvedeného přístupu je úvaha, že všechny objekty se dají substituovat skupinou malých ploch - tzv. patches. Patche jsou vyhledané afinně - kovariantní regiony na povrchu tělesa. U každého regionu se určí tvar, měřítko a orientace a pomocí KLT trackeru se najde poloha dané patche v každém obrázku. Princip algoritmu je převzat z [3].

Obrázek 4 : geometrická interpretace rektifikace patche (převzato z [3])



Postup se skládá z několika kroků: nejprve se extrahují jednotlivá patches - trackují se přes vícero obrázků, následuje segmentace tracků a vytvoření 3D modelu objektů ve scéně. Hlavní část metody iteruje mezi 2 kroky - pomocí RANSAC algoritmu se vybere hlavní pohyb a ponechají se body ostatních (odlišně rychlých) objektů. To se opakuje tak dlouho, dokud množina zbylých bodů nemůže spolehlivě vyčlenit další objekt.

Obrázek 5 : výsledky rekonstrukce na datech z filmového záznamu (převzato z [3])



3.3. Kompozice obrazu pro náhodně se objevující objekty ve scéně

Tento přístup používá jako základ neuspořádanou databázi předem pořízených fotografií zachycující statickou scénu vybraného interiéru. Smyslem práce je nadetekovat ve vstupní fotografii nový objekt. Celý algoritmus lze popsat následovně: najdou se obrázky zachycující stejnou část scény a transformují se do geometrie vstupního obrázku. Následně je z jejich částí vytvořena kompozice odpovídající vstupnímu obrázku. Plochy vstupního obrázku, pro které není nalezená žádná odpovídající část jsou považovány za nové objekty scény. Princip algoritmu je převzat z [4].

Nepředpokládá se, že databáze bude obsahovat stejnou fotografii scény, která je dána na vstupu. Pomocí kombinace nástrojů HESAFF a SIFT se v databázi vyhledají ty nejpodobnější. Poté se transformují do geometrie vstupního obrázku a vyberou se z nich oblasti, které se shodují se vstupním obrázkem.

Nejpodstatnější omezení, se kterým se uvedený algoritmus potýká, je špatné rozpoznání odpovídajících si částí při rozdílném osvětlení prostorů a závislost na bohaté databázi. Pro uvedenou práci byla použita databáze interiéru jednoho podlaží budovy pořízená robotem čítající kolem 9000 snímků.

Vzhledem k tomu, že při rekonstrukci scény námi vybraného seriálu není možnost získat dostačující databázi statických částí scény, nebyla tato metoda podrobněji studována.

Obrázek 6 : zleva doprava: vstupní obrázek, plocha vstupního obrázku, ke kterému nebyly nalezeny odpovídající části v databázi a další tři fotografie z databáze, které byly použity pro získání podobných částí (převzato z [4])



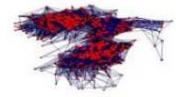
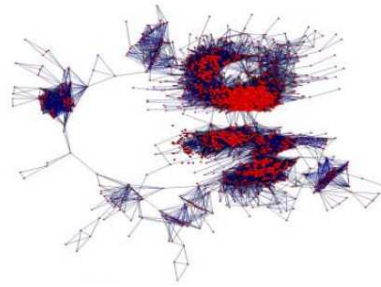
3.4. Modeling the world

Tato metoda se ukázala jako nejpřínosnější. Její kostru tvoří několik základních kroků, které nám umožnily více poznat strukturu SFM algoritmu. Typické pro tuto metodu je, že nevytváří 3D model. Jejím úkolem je organizace obrazů a jejich prostorová lokace. Vstupními daty jsou výsledky vyhledávání zvoleného hesla v internetové galerii. Princip algoritmu je převzat z [5].

Hlavním přínosem této práce bylo, že nám pomohla podrobněji porozumět fungování SFM algoritmu. Pro lepší přehled je v následující sekci napsána hlavní část algoritmu v pseudokódu.

Nejdříve se na všech vstupních obrázcích nadetekují významné korespondence (SIFTy). Následně se pro každý pár hledají vzájemně si odpovídající body. Dále se iteruje nad 8 body a pomocí RANSACu se počítá matice zobrazení. Zde jsou nastaveny prahy, které odstraňují nevyhovující data. V následujícím kroku se matche (tzn. nalezené korespondence) zorganizují do tracků. Vytvoří se connectivity graph (viz. obr. 7), který prezentuje vazby - společné korespondence mezi jednotlivými obrazy. V další části (odpovídá 3.4.1. FIND CAMERAS) se naleznou kamery a 3D lokace snímků.

Obrázek 7 : (zleva doprava) jeden ze vstupních obrázků, 3D model, connectivity graph



3.4.1. Pseudokód algoritmu

FIND MATCHES & TRACKS

```

for (all pictures) {
    pictures i, j
    create kd tree of j
    find feature_i in j - take 2 NN(d1,d2) from kd_tree_j
    for (all features_i)
        if (d1/d2 < 0,6) num_ofMatches_ij+=1
    end
}
for (every 2 pictures) {
    if (num_ofMatches_ij > 16) {
        RANSAC (8 points){
            return matrix F
        }
        if ( delta (match i ->F-> j) < (0,6%* max. dim.) ) {
            delete match
        }
        minimalization of delta (LM algorithm (F) )

        if (removing_matches < 20 ) {
            delete all matches
        }
    }
}
for ( all remaining pictures) {
    transitive closure {
        create tracks
    }
    create image connectivity graph
}

```


FIND CAMERAS

```
first {
    initial pair = select ( matches > 100 & lowest H_matrix (treshold = 0,4 % max.dim.) )
    5 points_algorithm -> R,T -> bundle adjustment -> X (3D coord.),R,T
}
for (all cameras) {
    add {
        select (camera A with max. M matches  $\subset$  X && camera B with 0,75M )
        if (M < 20){
            return;
        }
    }
}
for added cameras {
    DLT RANSAC (t = 0,4%max.dim.) -> K,R,T
    bundler adjustment (new cameras) clamp treshold
    bundler adjustment (new cameras' points)
    delete tracks where (min. 1 points with high reprojection error)
    max. appical angle > 2°
}
for all connected cameras {
    bundle adjustment(all cameras)
}
}
```

4. Vstupní 2D data

4.1. Vstupní data pro optimalizaci rekonstrukce z videozáznamu seriálu Friends

Jako vstupní data pro algoritmus odstraňující ze scény pohybující se objekty jsme vybrali video ze seriálu Friends. Pro tuto volbu jsme se rozhodli, protože se jedná o sitcom - odehrává se ve studiu za podobných světelných podmínek a na omezeném počtu scén, je velmi populární, což vylučuje možnost vstupní data záměrně přizpůsobit našim optimalizačním algoritmům a nakonec seriál má několik sérií - proto je velká pravděpodobnost, že se najde dostatek vhodných záběrů k rekonstrukci.

Na začátku byly vybrány díly, jejichž prostředí (statické zařízení bytu) se nejméně mění a zároveň jsou to díly sousední. U takovýchto dílů se dá předpokládat, že stejná významná korespondence bude nalezena ve všech dílech. Díly jsme rozdělili na jednotlivé záběry (tzv. shoty) pomocí programu M-Player. Pro potřeby optimalizace byly jednotlivé shoty přejmenovány následovně: *číslo série_číslo dílu_číslo shotu v dílu*.

Jednotlivé shoty bylo zapotřebí vybrat ručně a podle určitých kritérií:

1. shot musí být co nejméně zaostřen - čím víc je obraz přesný, tím lépe se scéna zrekonstruuje
2. jeho většina obsahuje statickou scénu - nikoliv osoby nacházející se ve scéně - kvůli úspoře výpočetního výkonu, takovéto shoty by se do rekonstrukce většinou nepřipojily
3. scéna je dostatečně osvětlena (nikoliv např. několika bodovými zdroji světla (imitace večera, noci); tento bod reflektuje omezení SFM algoritmu)

Maximální velikost záběrů z Friends, kterou šlo získat, byla 1280x720 px. SFM algoritmus pracuje s konstantní velikostí vstupních dat, proto si po nahrání datasetu (souboru všech vstupních fotografií nebo shotů z videa) obrázky zvětší nebo zmenší. Prioritní je tudíž ostrost detailu.

Obrázek 8 : příklad vstupních dat - seriál Friends



4.2. Tvorba vstupních dat pro analýzu pohybu

Vzhledem k nižší kvalitě dat předchozího videozáznamu jsme pro další práci zvolili data, která více vyhovují podrobnější analýze. Jedná se o sérii digitálních fotek, které zachycují scénu složenou z geometrických objektů. Ve scéně je pouze jeden objekt, který se pohybuje. Fotografie byly nafoceny ve fotoateliéru při konstantním převážně rozptýleném světle a byly pořízeny digitálními fotoaparáty Nikon se shodným nastavením.

Scéna je zachycena ve dvou datasetech, které zobrazují scénu a pohybující se objekt ve dvou různých polohách (prostý posun o 2 cm). Vzhledem k tomu, že umíme odlišit do kterého datasetu libovolná fotografie patří, mohli jsme tuto znalost použít pro optimalizaci a následnou identifikaci pohybujícího se objektu.

Obrázek 9 : příklad vstupních dat - experiment



5. Algoritmus optimalizace rekonstrukce

5.1. Hrubý popis algoritmu

Algoritmus lze primárně rozdělit na část analýzy a optimalizace. V prvním kroku necháme zrekonstruovat všechna vstupní data s několika odhady parametrů scény, které se napíší do spouštěcího souboru. Výsledek první rekonstrukce se analyzuje a následně se vygeneruje nový spouštěcí soubor s optimalizovanými parametry. Výsledkem druhého spuštění je již optimalizovaná rekonstrukce. Program se jmenuje „ProgOptimization“ a obsahuje veškerou potřebnou funkcionalitu prezentovanou v této kapitole. Dokumentace k vytvořenému programu se nachází v příloze A.

Pro přehlednost je algoritmus rozepsán do následujících bodů:

- 1) výběr vhodných vstupních dat (vytvoření datasetu)
- 2) zpracování dat pomocí CMP SfM Web Service
- 3) první rekonstrukce**
- 4) vygenerování vybraných dat (stabilních bodů) do formátu sbav (optimalizace)
- 5) zpracování dat spuštěním sbav na CMP SfM Web Service
- 6) výsledná optimalizovaná rekonstrukce**

5.2. Analýza vstupních dat

Nejdříve se dataset posoudí odhadem. Na jeho základě se sestaví spouštěcí soubor. Příklad takového souboru je v příloze C, spolu s příkladem spouštěcího souboru sbav z bodu 5) (podkapitoly 5.1.).

Pokud po zpracování dat získáme první rekonstrukci, jejíž bodový model přibližně odpovídá zrekonstruované scéně, zahájíme podrobnější analýzu modelu za použití funkcí zmiňovaného programu „ProgOptimization“. Pro níže uvedené obrázky byla použita data k datasetu 757.

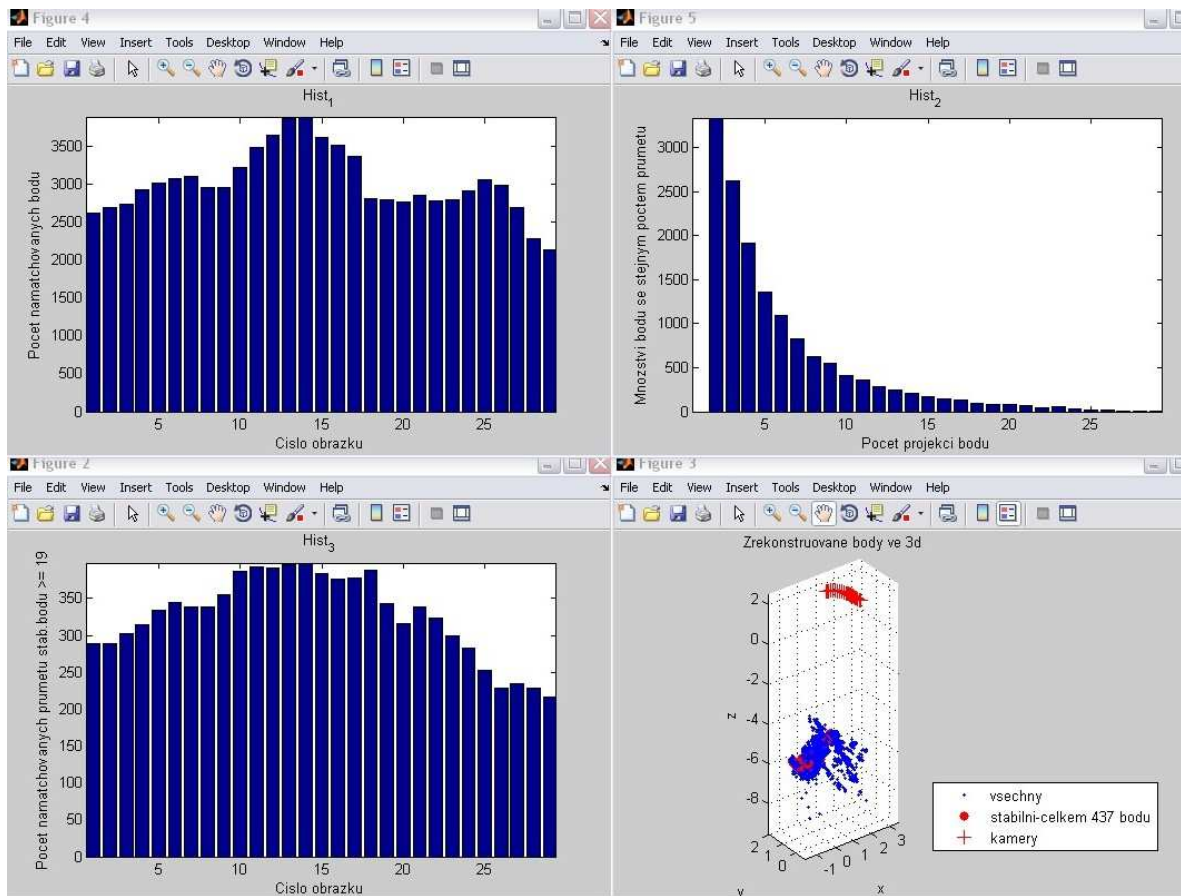
5.2.1. Histogramy stabilních bodů

Funkcionalitu tvorby histogramů zajišťuje výběr volby TrackLenHist. Výsledek ukazuje: počet zrekonstruovaných bodů, což odpovídá úspěšnosti připojení obrázku (viz. obr.10 - Hist_1), celkovou četnost skupin bodů se stejnou korespondencí (viz. obr.10 - Hist_2), počet stabilních bodů v obrázcích (viz. obr.10 - Hist_3) a nakonec vizualizace stabilních bodů ve 3D prostoru (viz. obr.10 - Zrekonstruované body ve 3D).

Při prvním spuštění této funkce je vhodné se zaměřit na histogram 1. Zde se projeví, zda je dataset vhodný k rekonstrukci. Jestliže je, měl by mít obrázky s relativně podobným množstvím zrekonstruovaných bodů. Stejně kritérium platí i pro histogram 3, k jehož vytvoření zadáme parametr určující výběr stabilních bodů (MinTrackLen), který vybere pouze body zobrazující se do min. MinTrackLen % z celkového počtu vstupních obrazů.

3D vizualizace je vhodná pro okamžitou kontrolu, zda se stabilní body nevybraly nevhodně - např. na dlouho stojících postavách. Umožňuje tak nalezení optimální hranice, kdy se vyberou body na stabilních objektech.

Obrázek 10 : výstup funkce TractLenHist, nastavený parametr MinTrackLen 65



5.2.2. Kontrola výskytu stabilních bodů na obrázcích

Pro ještě přesnější analýzu výskytu stabilních bodů byla naprogramována funkce ShowPict. Ta umožňuje snadnou orientaci ve scéně a vybraných stabilních bodech. Výstupem je uživatelem určený obrázek se zakreslením všech bodů a odlišením bodů stabilních (viz. obr. 11).

Obrázek 11 : vykreslení vybraného vstupního obrázku se zákresem stabilních bodů(•), vlevo pro MinTrackLen 60, vpravo pro MinTrackLen 65



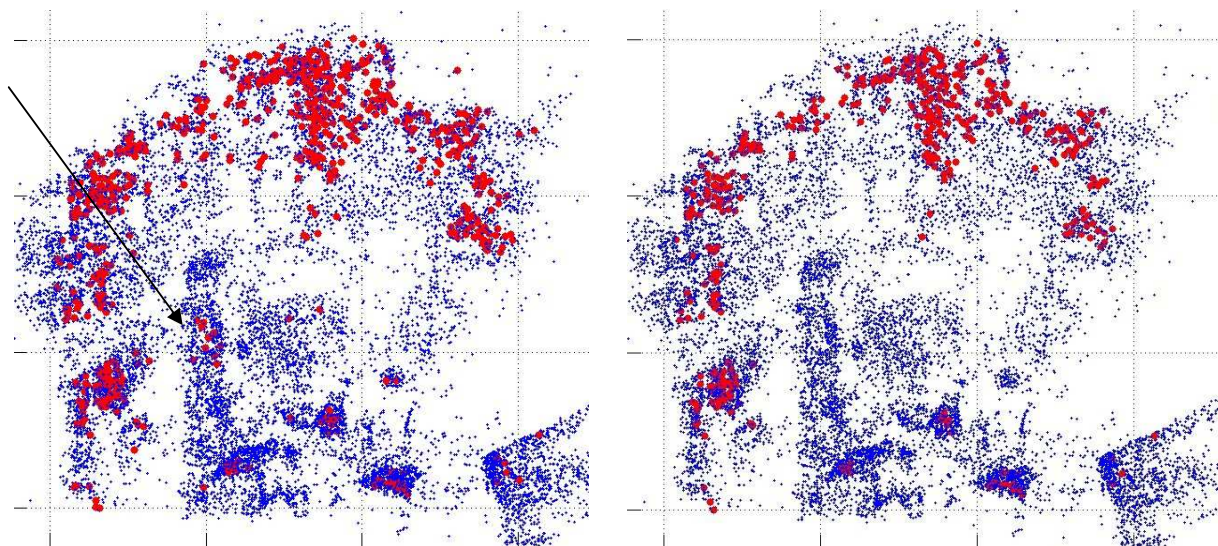
5.3. Metody výběru stabilních bodů

5.3.1. Výběr bodů s nejvíce korespondencemi

Lze předpokládat, že body, které se namatchovaly ve většině vstupních obrázků (uvažujeme-li obrázky zachycující jednu scénu) budou body na stabilních objektech. Proto je vybereme a použijeme ve spouštěcím souboru. Při následné rekonstrukci jsou stabilní body referenční - kamery odpovídající těmto bodům vykazují menší chyby vůči skutečnému stavu.

Metoda vyžaduje stanovení hranice MinTrackLen uživatelem. V úvahu se musí vzít celkové množství obrázků a body vybrané daným parametrem překontrolovat. Ukázka kontroly je na obr. 12.

Obrázek 12 : vlevo: 3D vizualizace s nastavením MinTrackLen 60, vpravo: 3D vizualizace s nastavením MinTrackLen 65. Šipka ukazuje na vybrané body (•) na pohyblivém objektu, které jsou pro optimalizaci nežádoucí.



5.3.2. Výběr bodů v sekvenci

Jiný způsob výběru stabilních bodů je stanovení délky sekvence po sobě časově následujících vstupních obrázků (shotů videa), ve kterých se bod musí bez vynechání vyskytovat, aby byl uznán jako stabilní. Délku sekvence se osvědčilo zadávat jako interval. Podstatou výběru je úvaha, že pokud se bod nadetekuje v sekvenci, pak je velmi pravděpodobné, že bude určen správně a kamery určené z těchto bodů budou mít malou odchylku od skutečnosti.

Uvedenou funkčnost zajišťuje v našem programu funkce Sequence. Do parametrů Min a Max se zadá dolní a horní mez intervalu počtu obrázků v sekvenci. Pro optickou kontrolu se výsledek v závěru vizualizuje obdobně jako na obr.12.

6. Získané výsledky optimalizačního algoritmu

6.1. Zvolené parametry

Data pro optimalizaci byla vybírána dvěma hlavními způsoby:

- 1) funkcí TrackLenHist a zadáním parametru MinTackLen, kdy se určily body objevující se v MinTackLen % obrázků z jejich celkového počtu.

Tento parametr je ovlivněn celkovým počtem obrázků. Proto pro velké datasety je lepší ho volit nižší a upravovat ho podle 3D vizualizace.

- 2) funkcí Sequence a zadáním parametrů Min a Max, které určily body objevující se za sebou bez vynechání v minimálně Min a maximálně Max za sebou jdoucích obrázcích (shotech)

Nastavení Min a Max se nejvíce osvědčilo mezi 8 až 10 shoty, nezávisle na počtu shotů v datasetu.

Na různě rozsáhlých datasetech stejné scény bylo vyzkoušeno několik způsobů jejich optimalizace - viz. tab. 1. Získané výsledky jsou ohodnoceny následující stupnicí:

- 1 výrazné zlepšení výsledné rekonstrukce
- 1- výrazné zlepšení s drobnými nedostatky oproti 1
- 2 patrné zlepšení s nedostatky
- 3 patrné zlepšení s výraznými nedostatky
- 4 patrné drobné zlepšení
- výsledek se nelišil od výsledku bez optimalizace

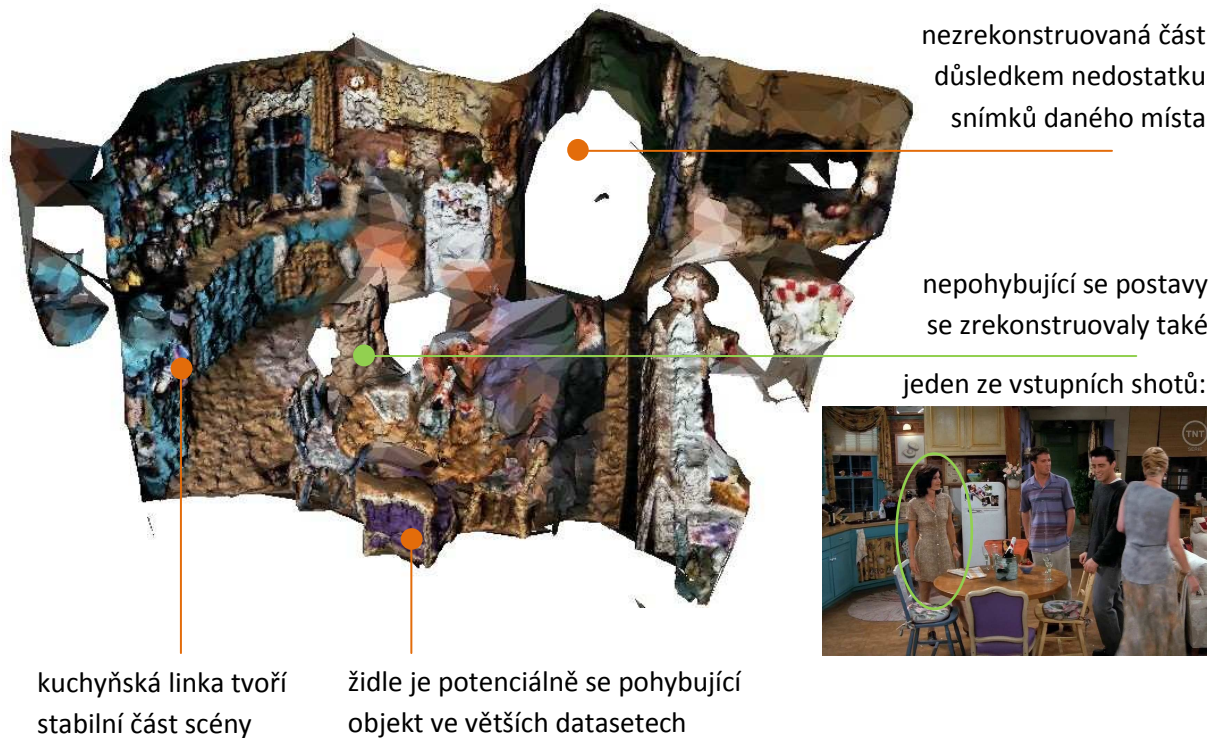
Tabulka 1 : výsledky optimalizace v přehledu

Č. data- setu:	Počet sekvencí:	Série:	Díly:	Cel- kem obr:	job bez úpravy: ONEBUTTON	Počet namatch. bodů	% nejpoč. / min-max	Počet stab. bodů	SBAV s vybranými parametry	job multireconv s parametry ze sbav	Ostatní nastavení	Hodno- cení:
757	1	2	1	29	work_2249	14 tis	80%	149	sbav_2252	final_2254		2
							80%	- -	sbav_2266	-		-
							90%	33	sbav_2255	final_2256		-
							90%	- -	sbav_2257	final_2259		-
							65%	437	sbav_2260	final_2261		-
							6-8	715	sbav_2262	final_2263		1
							6-8	- -	- -	final_2276	simplify<3>	1
							6-8	- -	- -	final_2277	simplify<8>	1
							8-10	346	sbav_2283	final_2284		2
							80 + 6-8	14	sbav_2264	final_2265		-
758	6	2	1	101	work_2250	43 tis	40%	111	sbav_2267	final_2269		4
							8-12	1157	sbav_2272	final_2273		2
							8-10	807	sbav_2274	final_2275		1
							8-10	- -	- -	final_2289	consistent<3>	2
							8-10	- -	- -	final_2304	consistent<5>	-
							6-8	2623	sbav_2278	final_2279		3
							8-10	671	sbav_2281	final_2282	work bez surf	1-
700	sekv+solo	2	1	197	work_2034	44 tis	8-10	929	sbav_2270	final_2271		1
							80 + 8-10	1025	sbav_2305	final_2306		2
759	sekv+solo	2	1,2	310	work_2251	104 tis	8-10	1105	sbav_2285	final_2286		-

6.2. Grafické výstupy jednotlivých výsledků optimalizací

Pro lepší orientaci je na obr.13 zachycena rekonstruovaná scéna z nadhledu a s vysvětlivkami. Následující scény odpovídají různě početným vstupním datasetům, viz. tab.1. Čísla složek a datasetů odpovídají jejich číslům na CMP SfM Web Service. Popis obrázků 15 až 17 odpovídá masce: č. obrázku : označení datasetu / složka vstupní rekonstrukce/složka optimalizované rekonstrukce

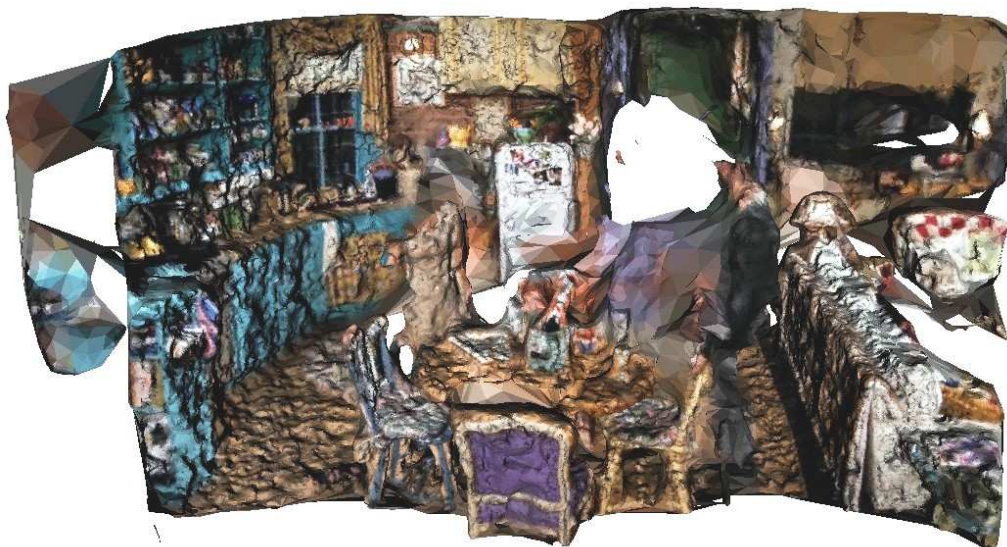
Obrázek 13 : rekonstrukce final_2263 po optimalizaci z nadhledu



Obrázek 14 : nejlepší výsledek rekonstrukce bez optimalizace (work_1945)



Obrázek 15 : dataset 757 / work_2249 / final_2263



Obrázek 16 : dataset 758 / work_2250 / final_2275



Obrázek 17 : dataset 700 / work_2034 / final_2271



7. Algoritmus analýzy experimentálních dat

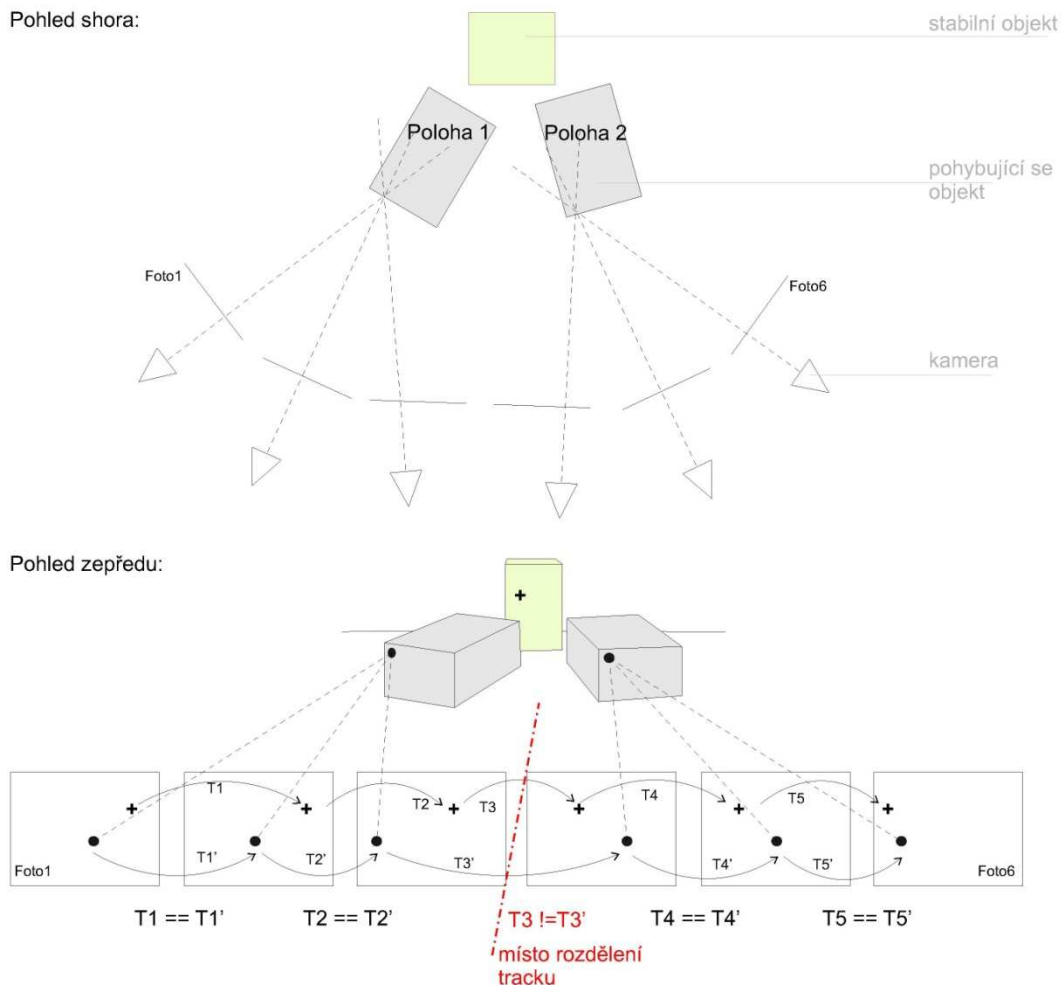
Vzhledem k nedostatečné kvalitě dat na scéně z videozáznamu jsme se rozhodli vytvořit si vlastní experimentální scénu a provést na ní další analýzu vlastností pohybujících se objektů. Pro tyto účely byl vytvořen program „ProgAnalyze“ umožňující vizualizaci jednotlivých výstupů. Dokumentace k vytvořenému programu se nachází v Příloze B.

7.1. Princip lokace pohybu

Pro geometrickou detekci pohybu ve scéně jsme se rozhodli použít následující princip: Nejdříve načteme všechny nalezené tracky. Tracky jsou vytvářeny na základě hledání podobnosti feature pointů v obrázcích. Zrekonstruují se tracky, které tvoří největší konzistentní množinu. V tomto případě to je stabilní pozadí scény a rozdělené tracky pohybujícího se objektu. Předpokládáme, že tracky pohybujícího se objektu byly rozděleny v místě, kde byla velká chyba převládající homografie a proto bychom je měli najít v nezrekonstruovaných trackech v délce přes obě sekvence. Viz. obr. 18.

V algoritmu tento výběr provedeme tak, že vyjmeeme nezrekonstruované tracky, jejichž body patří do množiny pohyblivých bodů (oddělených funkcí Analýze).

Obrázek 18 : princip výběru tracků zachycující pohyblivý objekt



7.2. Příprava vstupních dat

Vstupní dataset obsahující dvě sekvence s různou pozicí jednoho objektu jsme zrekonstruovali pomocí CMP SfM Web Service. Na výsledku se objevil pohyblivý objekt v obou polohách, viz. obr. 19.

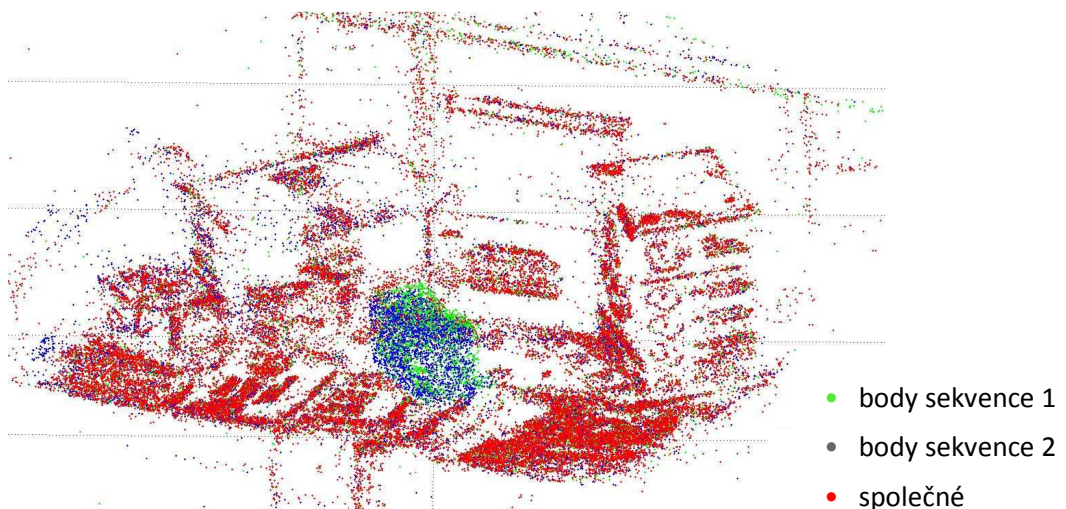
Obrázek 19 : rekonstrukce scény ze společného datasetu, šipka ukazuje na posunutý objekt, který se zrekonstruoval ve dvou různých pozicích



7.3. Zpracování dat a ověření experimentu

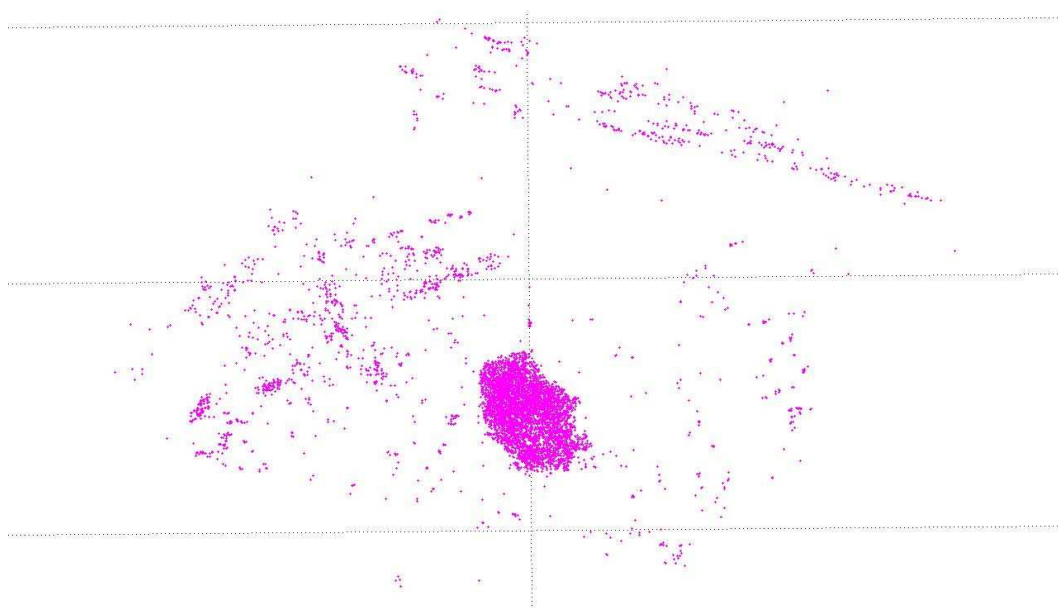
Protože známe obrázky patřící do jednotlivých sekvencí, využili jsme tuto znalost a vizualizovali rozdělení do 3D bodového modelu, viz. obr. 20. Body, které se úspěšně namatchovaly v sekvenci 1 i 2 lze považovat za body stabilní. Úvahou vychází, že body, které v naší scéně patří výhradně do jedné množiny, budou body pohybujícího se objektu. SfM algoritmus nenajde tyto body v druhé sekvenci, protože se objekt posunul a geometrie jejich kamery tak nepodporuje geometrii kamery většiny bodů.

Obrázek 20 : vizualizace rozdělení bodů do skupin podle náležitosti k sekvenci



Na obrázku 20 je většina bodů skupiny „společné“ nakreslena třemi barvami. Abychom mohli s body pohybujícího se objektu dále pracovat, museli jsme je z množiny výpočetně oddělit. To zjednodušeně znamená oddělit body, které nejsou překresleny červenou barvou. K tomu účelu jsme použili Delaunayovu triangulaci, kdy se každý 3D bod spojí hranou s vybranými sousedy. Z bodů spojených triangulací se vybírají ty, které nejsou spojeny s více než 10% body patřícími do množiny „společné“. Výsledek je na obr. 21.

Obrázek 21 : vyjmuté body, které nesousedí s více než 10% bodů z množiny „společné“:



Dalším krokem bylo ruční dohledání nezrekonstruovaných tracků, které obsahují body z vyjmuté množiny, obrázky z obou sekvencí a feature points s minimální odchylkou od skutečnosti ($\leq 3\text{px}$). Tyto tracky se nám dle předpokladu podařilo najít, viz. obr. 22 a 23. Počet jejich výskytu je uveden v tab.2.

Tabulka 2 : počty výskytu tracků

počet zrekonstr.tracků	56 867
počet nezrekonstr. tracků	24 648
počet tracků s body z množiny „společné“	1 916
počet nezrek. tracků pohyb. se objektu s odchylkou $\leq 3\text{px}$	8

Obrázek 22 : výřezy okolí feature pointu (tj. +) nezrekonstruovaného tracku s chybou $\leq 3\text{px}$



Obrázek 23 : výřezy okolí feature pointu (tj. +) nezrekonstruovaného tracku s chybou $\leq 3\text{px}$

Obr 1 feature 4714



Obr 4 feature 4863



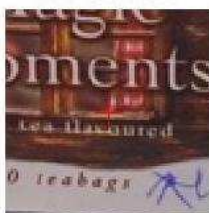
Obr 5 feature 4970



Obr 6 feature 6218



Obr 7 feature 6084



Obr 20 feature 5071



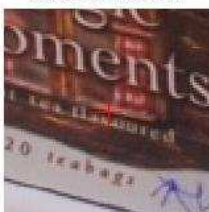
Obr 21 feature 4834



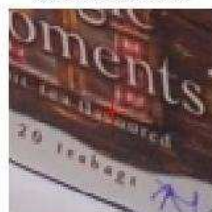
Obr 22 feature 5544



Obr 24 feature 5751



Obr 25 feature 6278



8. Závěr

Z vlastností samotného SFM algoritmu nelze snadno odstranit ze scény objekty, které se přemísťují pomaleji. Pokud je objekt stabilní na např. 6 a více snímcích, pak má dostatečnou podporu a algoritmus ho zrekonstruuje. Pokud se posune na jiné místo zachycené na další sérii snímků, je zrekonstruován i na nové pozici. Vznikne tak ve scéně duplicitní rekonstrukce daného objektu.

Z výsledků rekonstrukce obecného videozáznamu je zřejmé, že ke zlepšení došlo po opětovném spuštění rekonstrukce s optimalizovanými parametry, tj. spuštěním s body, které jsme označili za stabilní. Účinnější se při hledání stabilních bodů ukázal přístup hledající sekvence bodů určité délky oproti přístupu minimálního počtu korespondencí bodu. Stojící postavy, které patří mezi nestabilní objekty se přesto nepodařilo zcela odstranit.

Předpokládáme, že potřebného výsledku by se dalo dosáhnout, pokud bychom měli k dispozici více vstupních dat zachycující stabilní část scény a nastavili dostatečně vysoké spodní hranice minimálního počtu shotů, ve kterých se stabilní bod musí vyskytovat. Zlepšení za použití sekvenčního přístupu by se dalo dosáhnout za předpokladu ručního odstranění sekvencí, na kterých se pomalu pohybující objekty objevují.

Následující analýzu jsme prováděli na námi vytvořené scéně. Objekt zachycený ve dvou různých pozicích se zrekonstruoval dvakrát. Další zkoumání potvrdilo předpoklad, že specifickou znalostí snímku, ve kterém se objekt přemísť, dokážeme pohyb detekovat. Je tak možné vyčlenit body pohybujícího se objektu. Přesvědčili jsme se, že SFM algoritmus najde tracky na pohybujícím se objektu v obou pozicích, ale protože geometrie kamer z těchto tracků nepodporují geometrii kamer převládající skupiny, jsou tyto tracky zamítnuty.

Dalším postupem by mohl být experiment s různým nastavením snímku, kdy došlo k pohybu objektu. Kdybychom vyjmuli body patřící do výhradně první nebo druhé sekvence a na zbytku bodů znovu spustili rekonstrukci, pak může být zajímavé analyzovat počet nezrekonstruovaných tracků. Pokud by se objevila závislost na snímku posunu a počtu nezrekonstruovaných tracků, pak by zřejmě bylo možné určit snímek, od kterého se objekt nachází na nové pozici a tudíž i oddělit body pohybujícího se objektu bez znalosti snímku, na kterém k posunu došlo.

Seznam použitých zdrojů

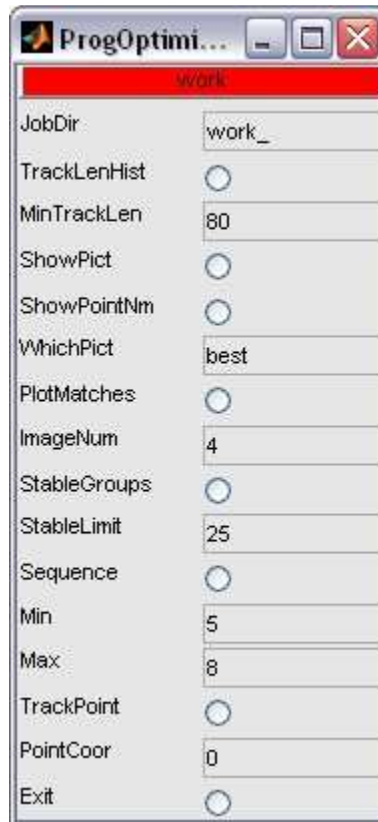
- [1] Pajdla Tomáš, *Geometry of Computer Vision*, CMP Internal Material. Unpublished.2011
- [2] Leoputra Wilson S., Tele Tan a Svetha Venkatesh, *A Unified 2D-3D Video Scene Change Detection Framework for Mobile Camera Platforms*, [online]. 2010, s. 2147-2153, Dostupné z: <http://espace.library.curtin.edu.au/R/>
- [3] Rothganger Fred, Svetlana Lazebnik, Cordelia Schmid a Jean Ponce, *Segmenting, Modeling, and Matching Video Clips Containing Multiple Moving Objects*, [online]. 2004, s. 1-16. Dostupné z: <http://www.di.ens.fr/willow/pdfs/pami06.pdf>.
- [4] Hongwen Kang, Alexei A. Efros, Martial Hebert a Takeo Kanade, *Image Composition for Object Pop-out*, 2nd International IEEE Workshop on 3D Representation for Recognition, [online]. 2009, Dostupné z: http://hwkang.com/files/ICOP/ICOP_3dRR.pdf
- [5] Noah Snavely , Steven M. Seitz a Richard Szeliski, *Modeling the World from Internet Photo Collections*, International Journal of Computer Vision, [online]. 2008, Dostupné z: http://phototour.cs.washington.edu/ModelingTheWorld_ijcv07.pdf

Přílohy

A Dokumentace k programu ProgOptimization

Tento algoritmus programu je umístěn na CD ve složce ProgOptimization.

Obrázek 24 : uživatelské rozhraní programu ProgOptimization



- tlačítko „work“
 - po vybrání libovolného radiobuttonu je pro vykonání vybraného příkazu potřeba kliknout na toto aktivační tlačítko
- JobDir :
 - vstup: název složky, která obsahuje data první rekonstrukce
 - složka se musí nacházet na stejné úrovni jako složka ProgOptimization
 - tato složka musí obsahovat:
 - bundle.out
 - camnames.txt
 - shoty z nichž se rekonstruovalo uložené ve složce „dataset“
 - list_tmp.txt
 - obrázky volně v základní složce jobu a ve složce dataset
- TractLenHist:
 - vstup: hodnota parametru MinTrackLen
 - výstup: histogram výskytu namatchovaných bodů v jednotlivých obrázcích
histogram četnosti bodů se stejným počtem korespondencí
histogram počtu stabilních bodů v jednotlivých obrázcích
3D zobrazení rekonstrukce s odlišením stabilních bodů

- výstupní data, která uloží pro další funkce:

data pro job typu sbav - do složky, která byla zadána v JobDir

data pro zobrazení bodů na vybraném obrázku - funkce ShowPict

- MinTackLen:
 - vstup: hodnota v procentech (1-100)
 - zadaná hodnota znamená, v kolika obrázcích (vypočteno, 100% odpovídá všem obrázkům v datasetu) minimálně musí mít bod korespondencí, aby byl považován za stabilní. Pro početné algoritmy je vhodné volit tuto hodnotu nižší.
 - tato hodnota určuje hranici \geq (stabilní body), nad kterou funkce TractLenHist uloží body pro uložení do formátu sbav, pokud je =0, pak data pro sbav neukládá
- ShowPict:
 - před spuštěním: při zadání jiné JobDir nutno nejdříve spustit TrackLenHist - loaduje uložená data z této funkce (paměťová úspora)
 - výstup:
 - zobrazí tabulku, v níž vypíše množství stabilních bodů v jednotlivých obrázcích
 - vykreslí body ve 3D, kde zvýrazní stabilní body příslušející vybranému obrázku = odpovídá hodnotě parametru WhichPict
 - zobrazí obrázek s pořadovým číslem WhichPict, vykreslí do něj všechny feature points, které se v něm nacházejí a označí ty, které jsou stabilní
 - pevně zadaná velikost obrázku
- ShowPointNm:
 - označíme v případě, že chceme vykreslit k bodům do obrázku WhichPict i jejich pořadová čísla
 - obrázek se pak nepřehledný, tato funkce se zapíná v případě registrace různých scén a potřeby zjistit čísla bodů, na které se scény budou „kotvit“
- WhichPict :
 - říká, který obrázek má zobrazit po spuštění funkce ShowPict a vykreslit do něj stabilní body - lze volit best nebo libovolně pořadové číslo obrázku
- PlotMatches:
 - funkce, která byla převzata z archivu Center for Machine Perception
 - vykreslí všechny nadetekované body obrázku ImageNum, včetně velikosti okolí
 - velikost okolí je znázorněna kruhem kolem bodu, čím větší kruh, tím je okolí bodu více jedinečné a proto vhodné pro rekonstrukci
 - pro tuto funkci je potřeba stáhnout kompletní výsledek rekonstrukce
- ImageNum:
 - funkce, která byla převzata z archivu Center for Machine Perception
 - udává pořadové číslo obrázku (od 1), který chceme zobrazit a vykreslit na něm body pomocí fce PlotMatches.

- StableGroups:
 - tabulka: řádek: č.obr., sloupec: č.nejstabilnějších bodů, které se v obr. vyskytují
 - pracuje se vytvořenými daty patřící k danému job, pokud byl alespoň jednou spuštěn

- StableLimit:
 - počet sloupců tabulky funkce StableGroups

- Sequence:
 - vytvoří pole, kde pořadí řádku odpovídá pořadí bodu
 - je poli jsou uloženy počtu obrázků, které jsou v sekvenci pro daný bod
 - uloží pouze points.txt pro sbav, cameras.txt zůstávají stejné jako v TrackLen
 - sekvence stejného čísla (8,8) lze zadat

- Min:
 - minimální počet obrázků včetně - v sekvenci hledané funkcí Sequence
 - celé číslo, nikoliv procento !

- Max :
 - maximální počet obrázků včetně - v sekvenci hledané funkcí Sequence
 - celé číslo, nikoliv procento !

- TrackPoint:
 - spuštění této funkce musí předcházet spuštění Sequence pro daný job
 - při vstupním parametru PointCoor vypíše tabulku, kde řádek odpovídá pořadovému číslu 3D bodu a sloupec odpovídá počtu sekvencí, ve kterých se nachází

- PointCoor
 - parametr „řádek, sloupec“ tabulky zobrazené funkcí TrackPoint

- Exit: po vybrání tohoto buttonu a kliknutí na tlačítko „work“ je program ukončen

B Dokumentace k programu analýzy experimentálních dat

Tento algoritmus programu je umístěn na CD ve složce ProgAnalyze.

Obrázek 25 : : uživatelské rozhraní programu ProgAnalyze



- tlačítko work
 - po vybrání libovolného radiobuttonu je pro vykonání vybraného příkazu potřeba kliknout na toto aktivační tlačítko
- JobDir :
 - vstup: název složky, která obsahuje data po prvotní zpracování dat pomocí CMP SfM Web Service
 - složka se musí nacházet na stejné úrovni jako ProgAnalyze
 - tato složka musí obsahovat:
 - bundle.out
- Analyze:
 - načte data, rozdělí zrekonstruované body do třech skupin a výsledek zobrazí, skupiny:
 - na body zrekonstruované ze sekvence 1 (fotoaparátem Cam_1)
 - na body zrekonstruované ze sekvence 2 (fotoaparátem Cam_2)
 - na body, které se zrekonstruovaly do obou kamer
 - body, které byly nalezeny v obou kamerách jsou vyjmuty a zobrazeny v samostatném okně
- Cam_1, Cam_2 :
 - do obou parametrů se zadá první a poslední fotografie patřící dané sekvenci (názvy fotografií musí být zvoleny takové, aby při seřazení podle názvu nebyly promíchány fotografie různých sekvencí)

- Perc_AllCam:
 - udává kolik procent z celkového počtu sousedů jednotlivého bodu, může

- OneFeature
 - zobrazí histogram obrázků s počtem nezrekonstruovaných bodů v každém obrázku
 - na příkazovou řádku uživatel napíše číslo vybraného obrázku
 - zobrazí se obrázek se zakreslením všech feature pointů a jejich čísel
 - na příkazovou řádku uživatel napíše číslo vybraného feature pointu
 - výstupem jsou vykreslené okolí feature pointu v nezrekonstruovaném tracku, do kterého daný feature point patří

- Pic_Fea
 - parametr určuje jaký obrázek a číslo feature pointu se má vykreslit funkcí OneFeature

- OneTrack
 - zobrazí vybraný nezrekonstruovaných track dle pořadového čísla vepsaného do příkazové řádky nebo uvedeného do parametru WhichTrack
 - tabulka všech nezrekonstruovaných tracků se vypíše na příkazovou řádku programu Matlab
 - v případě vyplnění parametru WhichTrack se zobrazí zadaný track

- WhichTrack
 - parametr jímž uživatel určuje, který track chce vizualizovat funkcí OneTrack

- Track_ofIntersec
 - zobrazí první track z nezrekonstruovaných tracků, které obsahují body z množiny „společné“, po stisknutí libovolného tlačítka zobrazuje další tracky v pořadí
 - pokud je vyplněn parametr StartFrom, pak jako první zobrazí track s pořadovým číslem StartFrom

- StartFrom
 - říká, jaké je první pořadové číslo tracku funkce Track_ofIntersec

- Manual_Show
 - zobrazuje napevno zadaná pole našlých tracků, které byly zamítnuty, ale odpovídají skutečnosti pro job2430

- Exit: po vybrání tohoto buttonu a kliknutí na tlačítko work je program ukončen.

C Příklady spouštěcích souborů

Spouštěcí soubor pro dataset 757 ke spuštění první rekonstrukce:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE xmljob
SYSTEM "http://ptak.felk.cvut.cz/sfmservice/DTD/onebuttonv2.dtd">
<xmljob dataset_id="757">
  <onebuttonv2>
    <simplify/>
    <video/>
    <calibration file="">
      <flength>1500</flength>
    </calibration>
  </onebuttonv2>
</xmljob>
```

Spouštěcí soubor pro dataset 757 ke spuštění první rekonstrukce:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE xmljob SYSTEM "http://ptak.felk.cvut.cz/sfmservice/DTD/sbav16.dtd">
<xmljob dataset_id="757">
  <sbav16>
    <cameras>cameras.txt</cameras>
    <points>points.txt</points>
  </sbav16>
</xmljob>
```

D Použitý software

Matlab R2010b

M-Player 1.0 rc2

SfM algoritmus na CMP SfM Web Service

E Obsah CD

Zdrojové kódy:

ProgOptimization ... zdrojové kódy programu optimalizace obecné scény, spouští se otevřením souboru *user_interface.m*

ProgAnalyze ... zdrojové kódy programu optimalizace obecné scény, spouští se otevřením souboru *user_interface.m*

Datové složky:

work_2249 ... výsledek první rekonstrukce datasetu 757

work_2250 ... výsledek první rekonstrukce datasetu 758

work_2034 ... výsledek první rekonstrukce datasetu 700

final_2263 ... výsledek rekonstrukce po optimalizaci z datasetu 757

final_2275 ... výsledek rekonstrukce po optimalizaci z datasetu 758

final_2271 ... výsledek rekonstrukce po optimalizaci z datasetu 700

job2430 ... výsledek první rekonstrukce pro experimentální data

bakalarska_prace.pdf