

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

REKONSTRUKCE 3D SCÉNY Z OBRAZOVÝCH DAT

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

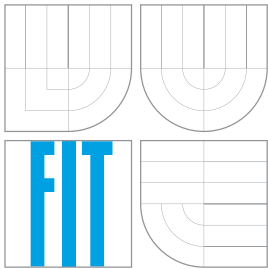
AUTHOR

Bc. ONDŘEJ AMBROŽ

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

REKONSTRUKCE 3D SCÉNY Z OBRAZOVÝCH DAT

3D SCENE RECONSTRUCTION FROM IMAGES

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. ONDŘEJ AMBROŽ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. MICHAL ŠPANĚL

BRNO 2010

Abstrakt

V práci jsou popsány již existující systémy rekonstrukce scény a uvedeny teoretické základy nutné při rekonstrukci scény z obrazových dat. Je navržen systém pro rekonstrukci scény z videosekvence, který je dále implementován a hodnoceny jeho výsledky s možností další práce. Jsou využity a popsány knihovny OpenCV, ARToolKit a SIFT.

Abstract

Existing systems of scene reconstruction and theoretical basics necessary for scene reconstruction from images data are described in this work. System of scene reconstruction from video was designed and implemented. Its results were analyzed and possible future work was proposed. OpenCV, ArtToolKit and SIFT libraries which were used in this project are also described.

Klíčová slova

3D rekonstrukce, epipolární geometrie, parametry kamery, fundamentální matice, významné body, OpenCV.

Keywords

3D reconstruction, epipolar geometry, camera parameters, fundamental matrix, feature points, OpenCV.

Citace

Ondřej Ambrož: Rekonstrukce 3D scény z obrazových dat, diplomová práce, Brno, FIT VUT v Brně, 2010

Rekonstrukce 3D scény z obrazových dat

Prohlášení

Prohlašuji, že jsem tuto práci vypracoval samostatně pod vedením pana Ing. Michala Španěla

.....

Ondřej Ambrož
25. května 2010

Poděkování

Děkuji vedoucímu práce panu Ing. Michalu Španělovi za cenné rady a připomínky při vytváření práce.

© Ondřej Ambrož, 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

1	Úvod	5
2	Existující řešení 3D rekonstrukce a mapování pohybu kamery	6
2.1	3D rekonstrukce scény z několika pohledů	6
2.2	Systém 3D rekonstrukce scény pomocí otáčející se základny	8
2.3	MonoSlam	9
2.4	PTAM - Parallel Tracking and Mapping	11
3	Významné body	14
3.1	Harrisův detektor	14
3.2	SIFT	15
3.3	SURF	17
3.4	Korespondence významných bodů	18
4	Kamera	19
4.1	Perspektivní projekce	19
4.2	Vnitřní parametry kamery	20
4.3	Vnější parametry kamery	21
4.4	Projekční matice	21
4.5	Výpočet projekční matice	21
5	Rekonstrukce 3D bodu	23
5.1	Fundamentální matice a epipolární geometrie	23
5.2	Lineární triangulační metoda	24
5.3	SVD	25
6	Návrh řešení	26
6.1	Výpočet vnitřních parametrů kamery	26
6.2	Získání snímků	27
6.3	Nalezení významných bodů v obraze	28
6.4	Vyhledávání korespondencí	28
6.5	Odstranění falešných korespondencí	29
6.6	Výpočet pozice kamery na základě markeru	29
6.7	Výpočet pozice kamery z již dříve rekonstruovaných bodů	29
6.8	Výpočet nových 3D bodů	30
6.9	Přepočet existujících 3D bodů	31
6.10	Výpočet normály každého bodu	31
6.11	Rekonstrukce povrchu metodou Poisson	31

7 Implementace	33
7.1 ARToolKit	33
7.2 OpenCV	35
7.3 SIFT	36
7.4 Kalibrace kamery	37
7.5 Nastavení parametrů programu, inicializace vstupu videa	39
7.6 Vyhledání významných bodů SIFT, SURF	39
7.7 Vyhledání korespondencí	40
7.8 Určení pozice kamery pomocí markeru	41
7.9 Určení pozice kamery z již rekonstruovaných bodů	41
7.10 Výpočet nových bodů z korespondencí	42
7.11 Přepočítání již rekonstruovaných bodů	43
7.12 Výpočet normál rekonstruovaných bodů	43
8 Výsledky	44
8.1 Test umělé scény	44
8.2 Test reálné scény ze souboru	46
8.3 Test reálné scény z kamery	47
8.4 Zhodnocení výsledků testů a implementace	49
8.5 Možnost budoucího vývoje	50
9 Závěr	51
A Obsah CD	55
B Porovnání bodů s umělou scénou	56
B.1 Okolí markeru	56
B.2 Přední stěna	56
B.3 Vrchní stěna	56
B.4 Celá scéna	58

Seznam obrázků

2.1	Poisson rekonstrukce povrchu [18].	7
2.2	Původní snímek budovy (a) ve srovnání s 3D rekonstrukcí (b) [18].	7
2.3	(a) Konfigurace systému platformy. (b) Systém platformy [25].	8
2.4	2 snímky rekonstruované scény[25].	9
2.5	Vypočítané mračno bodů [25].	9
2.6	Zrekonstruovaná síť trojúhelníků s texturou [25].	10
2.7	Ukázka programu monoSLAM [9].	11
2.8	Ukázka rozšířené reality. Darth Vaderova laserová zbraň zaměřuje pomocí optického centra kamery. [13].	12
2.9	Mapování. Po inicializaci se smyčka nekonečně opakuje a příležitostně získá nový snímek ze sledovacího systému. [13].	13
3.1	Znázornění problému vyhledávání významných bodů pomocí měřítkově závislého Harrisova detektoru[8].	15
3.2	Konstrukce scale-space pomocí DoG [8].	16
3.3	Ukázka vyhledání významných bodů pomocí SIFTu	17
3.4	Zleva doprava: Gaussova druhá parciální derivace ve směru y a xy a aproximace pomocí čtvercové oblasti [5].	17
4.1	Schéma odvození perspektivní projekce [14].	20
4.2	Transformace z obrazových souřadnic do pixelu[24].	21
5.1	Epipolární geometrie dvou pohledů [7].	23
5.2	Nepřesnost rekonstrukce. Stínovaná oblast v každém z příkladů ilustruje nepřesnost, která závisí na úhlu mezi kamerami. Bod je čím dál méně přesně vypočítán, čím více jsou směry kamery rovnoběžné [10].	24
6.1	Zjednodušený návrh řešení.	27
6.2	Významné body nalezené v obraze. Zeleně označené korespondence s předchozími snímky, červeně korespondence již rekonstruovaných bodů.	30
6.3	Část rekonstruovaných bodů v prostoru. Body používané pro výpočet projekční matice jsou žluté a světle modré.	30
6.4	Vlevo jsou zobrazeny orientace bodů směřující ke kameře. Vpravo pak vypočtené normály bodů.	32
7.1	Implementační diagram.	34
7.2	Ukázka možnosti použití knihovny ARToolKit [1].	35
7.3	Vyhledání objektu ve scéně pomocí knihovny SIFT[4].	37
7.4	Označování bodů při kalibraci.	38

7.5	Potvrzení průniků úseček.	38
7.6	Označování linií na kalibračním vzoru.	38
7.7	Vyhledání významných bodů. Vlevo metodou SIFT, vpravo metodou SURF	40
7.8	Vyhledání a označení pozice markeru ve snímku. Snímek je nejprve prahován a pak v něm nalezen marker.	42
8.1	Program Blender s vytvářenou jednoduchou scénou pro rekonstrukci.	45
8.2	Ukázka vyhledaných bodů metodou SIFT.	45
8.3	Dva pohledy na rekonstruované body. Je vidět pravý úhel mezi dvěma stěnami.	45
8.4	Porovnání scény s rekonstruovanými body v Blenderu.	46
8.5	Snímek testu reálné scény ze souboru s vyznačenými body metodou SIFT. .	46
8.6	Vlevo: Náhled na rekonstruované body celé scény při využití metody SIFT. Vpravo: Zadní pohled na rekonstruované body válcovité krabičky nacházející se vlevo od markeru, společně s body na povrchu jedné z knížek.	47
8.7	Náhled na rekonstruované body celé scény v testu reálné scény ze souboru při využití metody SURF	47
8.8	Vyhledané významné body z pohledu na zadní stranu krabičky.	48
8.9	Boční pohled na rekonstruovanou krabičku. Modré body jsou rekonstruované při znalosti markeru. Červené body jsou rekonstruované z vypočítané pozice kamery.	48
8.10	Dva pohledy na zrekonstruovaný povrch krabičky metodou Poisson v pro- gramu MashLab.	49
B.1	Histogram bodů okolí markeru od roviny markeru.	57
B.2	Histogram bodů okolí přední stěny od roviny přední stěny.	57
B.3	Histogram bodů okolí vrchní stěny od roviny vrchní stěny.	58
B.4	Celkový histogram vzdáleností celé scény.	59

Kapitola 1

Úvod

3D rekonstrukce objektů a scény dosáhla během posledních let velkého rozvoje. Jejimi postupy se začíná zabývat stále více lidí. Se snížením výpočetního času na nových procesorech se již dají algoritmy používat i v reálném čase. Větší zájem je tak i ve virtuální a hlavně rozšířené realitě, která získává čím dál větší oblibu a kde se právě možnosti 3D rekonstrukce dají dobře uplatnit. 3D začíná být skloňováno i u neodborné veřejnosti při nástupu 3D televizorů, které podle prognóz mají zažít v letošním roce velký rozmach. Dále u natáčení filmů ve 3D je využíváno stereokamer pro získání hloubky snímané scény. Tyto jednotlivé příklady vedou k zájmu i o samotnou 3D rekonstrukci a získávání pozice kamer ve scéně, která je základem přesné rekonstrukce. Projekty jako MonoSlam nebo PTAM, kde první se zaměřuje na automatickou lokalizaci robota v prostoru, a to i pro rozsáhlé scény, ovšem s menší přesností, a druhá určující přesnější lokalizaci využitelnou pro již zmiňovanou rozšířenou realitu. Zakládají dobrou možnost rekonstrukce celé snímané scény i jednotlivých objektů. Metody MonoSlam a PTAM jsou stále předmětem vývoje a každým rokem se zlepšují a zpřesňují.

Tato práce se zabývá a snaží vysvětlit základní postup geometrické rekonstrukce statické scény, která je snímána kamerou. Rekonstrukce scény je počítána z videa a nebo přímo z připojené kamery. Kamera je zkalibrovaná a její počáteční pozice míří v prostoru na předem určený a známý marker. Parametry kamery je tak možno pro první snímky získat velice přesně a dále z již rekonstruovaných bodů získávat pozici kamery, i když ta již marker nesnímá.

Ve druhé kapitole se zaměřují na již existující možnosti řešení 3D rekonstrukce scény a metody určující pozici kamery. Třetí kapitola se věnuje detektoru SIFT, SURF, Harrisově detektoru a korespondencím významných bodů. Čtvrtá kapitola se zabývá parametry kamery a možností jejich vypočítání z bodů ve scéně. V páté kapitole je pak uveden základ teorie pro 3D rekonstrukci ze dvou a více bodů. V kapitole návrh řešení jsou představeny jednotlivé části programu a jejich zamýšlená funkčnost jako celku. V následující kapitole je popsána samotná implementace programu podle navrhnutého řešení. Kapitola výsledky pak na několika testech představuje funkčnost vzniklého systému a jsou shrnuty jeho klady a zápory a možnost pokračování práce.

Kapitola 2

Existující řešení 3D rekonstrukce a mapování pohybu kamery

V kapitole jsou popsány dvě již existující řešení 3D rekonstrukce, následované popisem metod sledování pozice kamery ve scéně.

2.1 3D rekonstrukce scény z několika pohledů

Společnost Goodwill Company vyvinula systém pro 3D rekonstrukci scény z několika snímků [18], přičemž počítá s nasazením na automaticky řízených strojích.

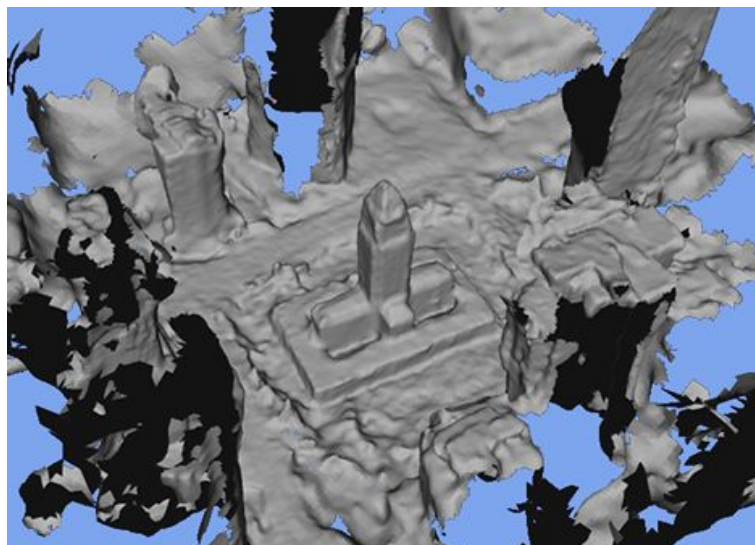
Základem systému je zkalibrovaná kamera, kterou je pořízeno velké množství snímků objektu z různých pozic. V každém snímku jsou pak vyhledávány význačné body pomocí SIFT detektoru, spojujány s korespondujícími body v dalších snímcích a pomocí korespondencí pak počítány vnější parametry kamery. Tyto parametry jsou použity pro výpočet fundamentálních a esenciálních matic. Po nalezení těchto matic dochází k výpočtům bodů v prostoru. Pomocí metody Poisson¹ rekonstrukce je získán povrch 3D objektu a nakonec je na něj nanášena textura.

Celý proces se dá tedy shrnout do následujících částí:

- Vytvoření modelu kamery a kalibrace.
- Detekce význačných bodů v obraze metodou SIFT.
- Vyhledání korespondujících bodů.
- Získání vnějších parametrů kamery.
- Vypočítání počátečního mraku bodů.
- Rekonstrukce povrchu metodou Poisson, obr. 2.1.
- Otexturování modelu, obr. 2.2.

Pro model kamery je využita standardní dírková kamera s projekční maticí. Kalibrace vnitřních parametrů kamery je získána z nejméně deseti snímků černobílé šachovnice snímané z několika různých uhlů. Stejně jsou vypočítány i radiální a tangenciální deformace.

¹Více o metodě Poisson lze nelézt česky v bakalářské práci [21] nebo přímo v originále [12]



Obrázek 2.1: Poisson rekonstrukce povrchu [18].



Obrázek 2.2: Původní snímek budovy (a) ve srovnání s 3D rekonstrukcí (b) [18].

Detekce význačných bodů metodu SIFT zajišťuje invariantní body vůči změnám měřítka, rotacím a osvětlení. Při detekci jsou vytvářeny deskriptory o rozsahu 128 hodnot pro každý bod.

Deskriptory bodů jsou použity pro vyhledání korespondujících bodů v předchozích snímcích pomocí euklidovské vzdálenosti a vyhledávání nejbližšího souseda v kd-stromech. Pokud nejbližší a druhý nejbližší bod mají poměr vzdálenosti menší než 0.8 jedná se s velkou pravděpodobností o korespondující bod. Korespondence jsou dále ověřovány na základě epipolární geometrie.

Vypočítána je fundamentální matice 8-bodovým algoritmem a z ní je získána esenciální matice. Ta je použita pro výpočet vnějších parametrů kamery a základního mraku bodů. Výsledek je pak pře počítán pomocí metody svazku paprsků, kdy jsou zpřesněny všechny stavy kamery a získané 3D body.

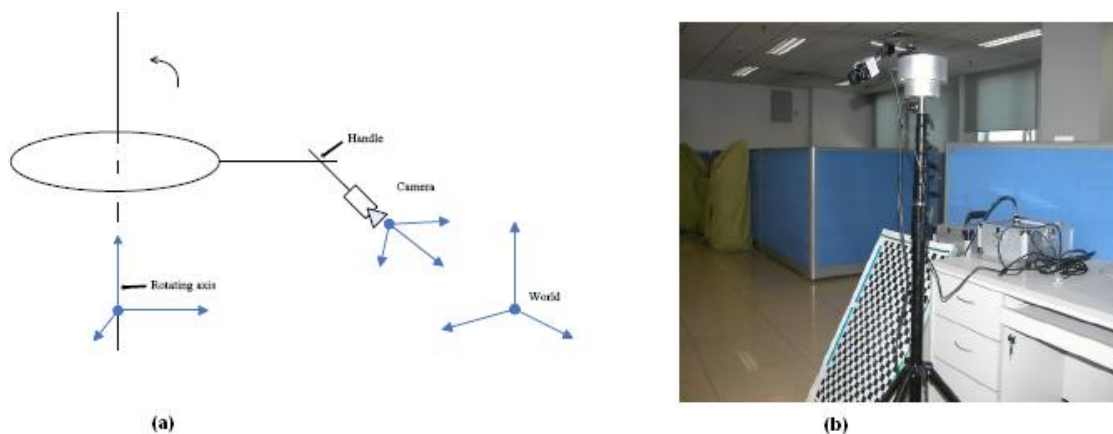
Rekonstrukce povrchu z mraku bodů je následně rozdělena na několik částí. V první jsou využity techniky iterativně rostoucího povrchu a PMVS („Patch-base Multi-view Stereo“). Tyto metody vypočtou orientace a normály bodů. Poté následuje rekonstrukce povrchu metodou Poisson. Nakonec je výsledný model otexturován.

Při porovnávání modelu a výsledné rekonstrukce došli autoři k závěru, že jejich metoda má chybovost pod jedno procento.

2.2 Systém 3D rekonstrukce scény pomocí otáčející se základny

Zhang [25] se ve své práci zabývá možností 3D rekonstrukce scény nasnímané kamerou z otáčející se základny.

Jak je vidět na obrázku 2.3, systém se skládá z kamery, otáčející se základny a kontrolní jednotky. Kamera je přichycena na základně na pohyblivém ramenu, které umožňuje změnit úhel pohledu do scény. Otáčející se základna souvisle mění rovinné umístění kamery ve scéně a úhel změny polohy kamery lze získat z kontrolní jednotky.



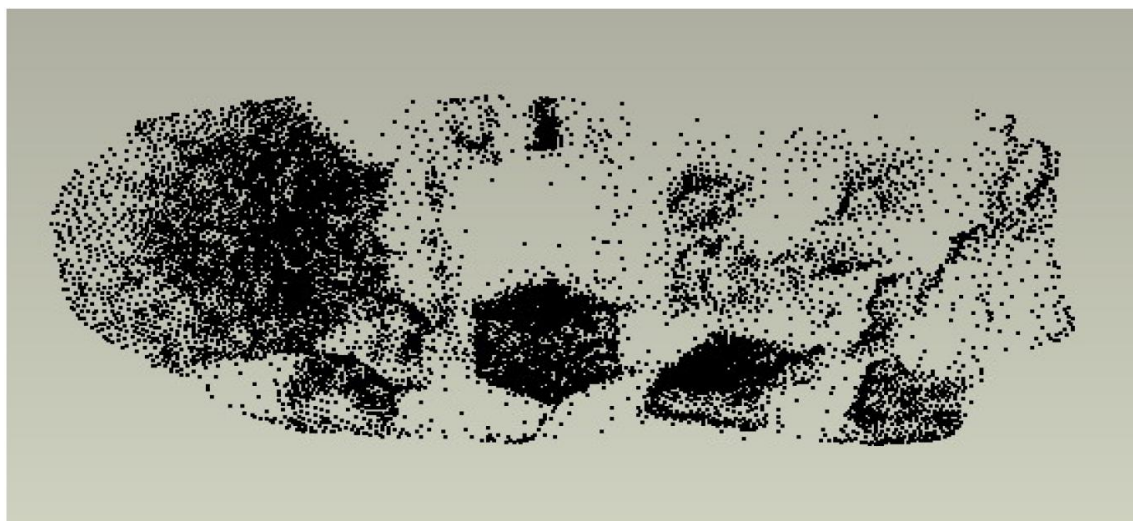
Obrázek 2.3: (a) Konfigurace systému platformy. (b) Systém platformy [25].

Model kamery je dírková kamera s vnitřními a vnějšími parametry. V prvním kroku je kamera kalibrována pomocí černobílé šachovnice, kdy jsou zjištěny i radiální a tangenciální deformace. Tento kalibrační vzor je umístěn na podlahu a je zvolen jako rovina x-y světového souřadného systému. To znamená, že celý vzor leží v $Z = 0$.

Po úspěšné kalibraci a nastavení systému se přistoupí k samotné rekonstrukci, kdy scéna je snímána vždy po předem nastaveném úhlu pootočení základny kamery obr. 2.4. Při znalosti pootočení a správném nastavení souřadných systému a vztahů mezi nimi je provedena rekonstrukce pomocí triangulační metody. Kdy jednotlivé korespondující body byly získány metodou SIFT a porovnáváním deskriptorů těchto bodů. Tak vznikne mračno bodů, jak je vidět na obr. 2.5, které je ještě zpřesněné metodou svazků paprsků. Výsledkem je pak částečně texturovaná síť trojúhelníků viz. obr. 2.6.



Obrázek 2.4: 2 snímky rekonstruované scény[25].



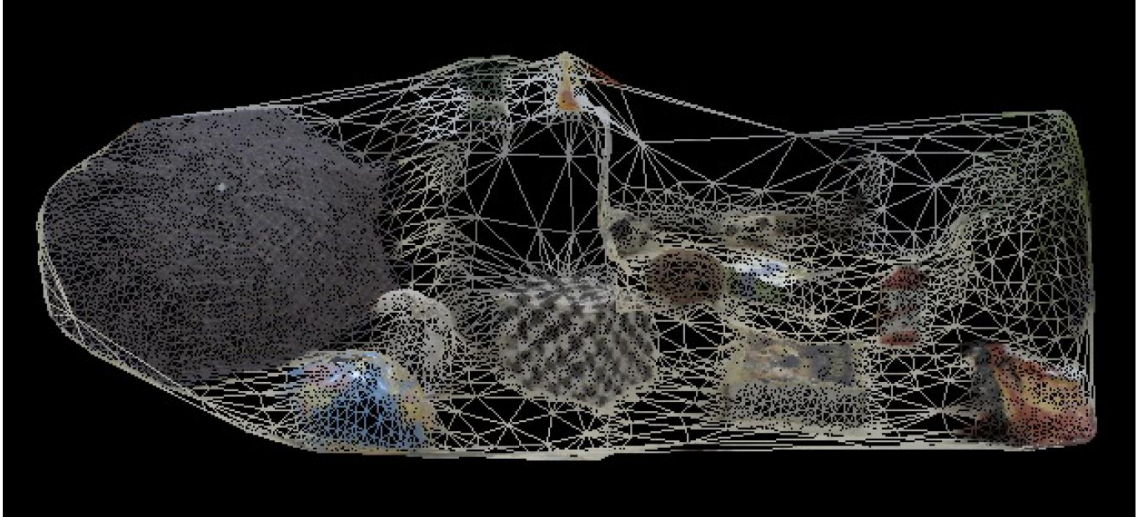
Obrázek 2.5: Vypočítané mračno bodů [25].

Autoři sami přiznávají, že jimi navržený systém není zatím moc přesný, ale je plně automatický a v průběhu výpočtu nepotřebuje žádný zásah uživatele. Také funguje pro poměrně velké scény. Cílem je sestavit plně automatický systém pro rekonstrukci místa činu v kriminalistice.

2.3 MonoSlam

Jedná se o algoritmus pracující v reálném čase, který vypočítává (rekonstruuje) 3D trajektorii kamery pohybující se neznámou scénou [9].

Klíčovým konceptem přístupu je pravděpodobnostní mapa založená na významných bodech. Mapa je inicializována při spuštění systému a přetrvává až do konce výpočtu. Zároveň se postupem času doplňuje a zvětšuje dynamicky a je upravována pomocí *rozšířeného kalmanova filtru* (angl. *Extended Kalman Filter*). Pravděpodobná pozice kamery a významné



Obrázek 2.6: Zrekonstruovaná síť trojúhelníků s texturou [25].

body jsou přidávány během pohybu a sledování scény. Pokud jsou nové významné body nalezeny, mapa je zvětšena o nové stavy a některé body, pokud je to nezbytné, mohou být i mazány.

Matematicky je mapa prezentována stavovým vektorem x a kovarianční maticí P

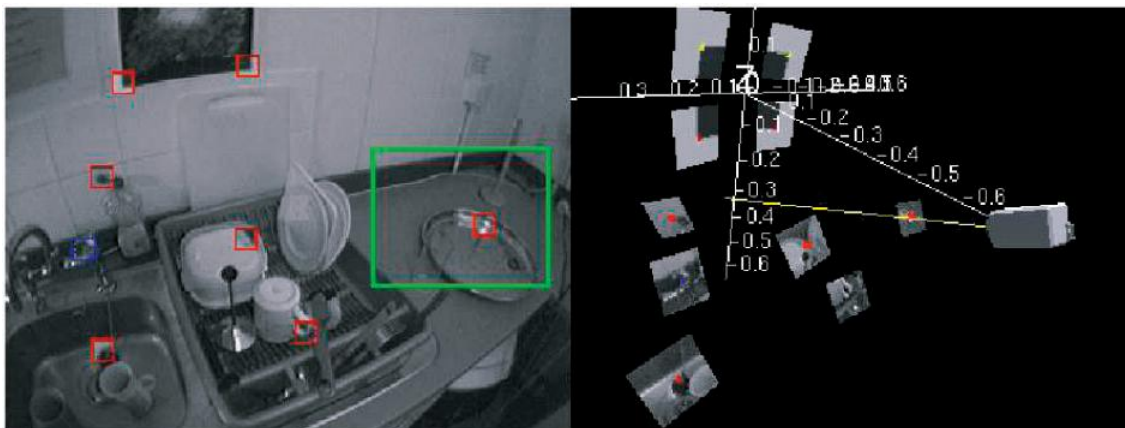
$$x = \begin{pmatrix} x_v \\ y_1 \\ y_2 \\ \vdots \end{pmatrix}, P = \begin{bmatrix} P_{xx} & P_{xy_1} & P_{xy_2} & \dots \\ P_{y_1x} & P_{y_1y_1} & P_{y_1y_2} & \dots \\ P_{y_2x} & P_{y_2y_1} & P_{y_2y_2} & \dots \\ \vdots & \vdots & \vdots & \dots \end{bmatrix} \quad (2.1)$$

Explicitně stavový vektor kamery x_v se skládá z 3D pozice vektoru r^W , orientovaného quaternionu q^{RW} , rychlostního vektoru v^W a úhlové rychlosti ω^R relativní souřadnému systému W a snímku kamery R :

$$x_v = \begin{pmatrix} r^W \\ q^{WR} \\ v^W \\ \omega^R \end{pmatrix} \quad (2.2)$$

Úlohou mapy je v tomto případě umožnit lokalizaci v reálném čase a neobsahovat úplný popis scény, proto se autoři zaměřili na získání kvalitních významných bodů. Relativně velké oblasti obrázku jsou brány jako významné body, díky tomu jsou dlouhodobě znovu použitelné. Tyto oblasti obrázku jsou vyhledávány automaticky ve snímcích kamery. Cílem je možnost nelézt tyto oblasti znovu i v jiných snímcích a to i při rychlých pohybech kamery. Pro lokalizaci používají strategii vytvořenou Shi a Tomasi [19].

Pro inicializaci kamery je použit známý kalibrační tvar, který je před kameru postaven. Díky tomuto vzoru je poskytnuto několik významných bodů (většinou 4)[9]. Použitím vzoru dojde k možnosti vypočítat měřítko, které je důležité pro další běh algoritmu. Navíc odpadá nutnost získávání prvních významných bodů pro kalibraci a ihned je možno odhadovat pohyb kamery obr.2.7.



Obrázek 2.7: Ukázka programu monoSLAM [9]

Metoda MonoSlam je původně zamýšlena pro vytváření mapy robota a zjišťování polohy v neznámé scéně, kde pracuje výborně i pro velice rozsáhlé scény s možností návratu na již prozkoumaná místa. Autoři použili tuto metodu i pro rozšířenou realitu, kdy do mapované scény místnosti umístili několik virtuálních políček a ty byly následně relativně přesně zobrazovány do snímaného videa.

2.4 PTAM - Parallel Tracking and Mapping

Paralelní sledování a mapování pro malé prostory rozšířené reality [13] je práce George Kleina a Davida Murrayho z Oxfordské univerzity.

Cílem metody je sestavování mapy prostředí na základě zjišťování pozice a sledování zkalibrované kamery v předem neznámém prostředí bez jakýchkoliv známých objektů nebo inicializujících předmětů. Po sestavení scény je možné do ní vkládat virtuální objekty, jak je možno vidět na obrázku 2.8.

Metoda lze shrnout následujícími body:

- Sledování a mapování je rozděleno a běží ve dvou paralelních vláknech.
- Mapování je založeno na klíčových snímcích, které jsou zpracovány dávkovou technikou.
- Mapa je hustě inicializována ze stereo páru pomocí 5-bodového algoritmu.
- Nové body jsou vkládány na základě epipolárního vyhledávání.

Mapa obsahuje kolekci význačných bodů ve světovém souřadném systému. Každý význačný bod reprezentuje lokálně rovinnou malou plochu ve scéně. Všechny body také obsahují normálu a referenci na původní pixel v obraze.

Mapa dále obsahuje N klíčových snímků, které jsou zachyceny kamerou v různém čase. Klíčový snímek má přiřazen snímek scény a uchovává 4-úrovňovou pyramidu v odstínech šedi, kde úroveň 0 obsahuje celý snímek 640x480 a v dalších úrovních je podvzorkován až do úrovně tři o rozměrech 80x60 pixelů.

Sledovací systém získává snímky přímo z kamery a v reálném čase vypočítává pozici kamery vůči sestavené mapě. Tento systém má několik kroků:

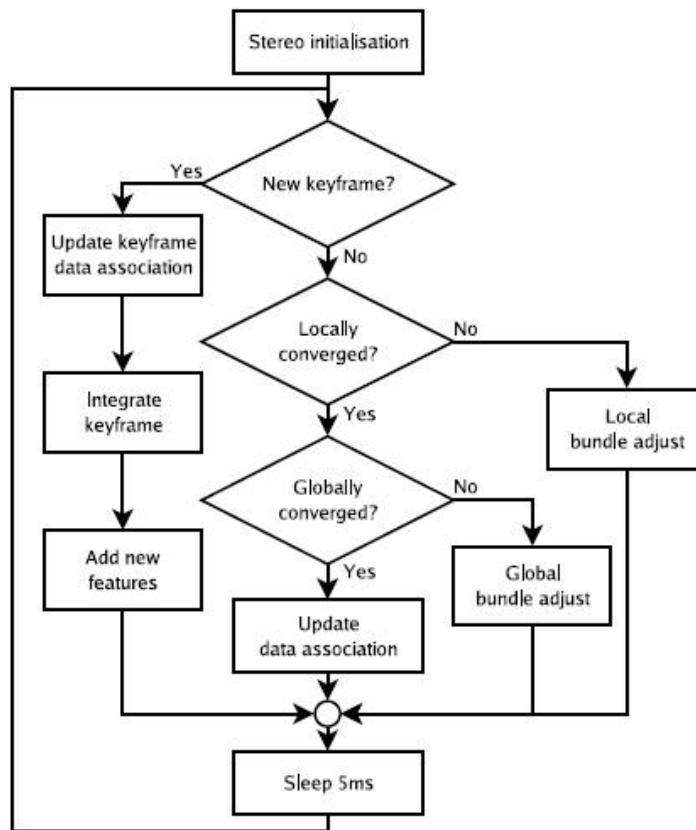


Obrázek 2.8: Ukázka rozšířené reality. Darth Vaderova laserová zbraň zaměřuje pomocí optického centra kamery. [13]

- Nový snímek je získán z kamery a předběžná pozice je získána z modelového pohybu kamery.
- Body z mapy jsou zobrazeny do snímku pomocí předběžné pozice kamery.
- Malá část bodů je vyhledána ve snímku.
- Pozice je upřesněna využitím vyhledaných bodů.
- Velký počet bodů je na základě zpřesněné pozice zobrazen do snímku a vyhledán.
- Finální pozice kamery je vypočítána ze všech dříve nalezených bodů.

Mapování se sestává ze dvou rozdílných částí: První je inicializace mapy na základě stereo techniky, kdy ze dvou vzdálených snímků kamery je vytvořena základní mapa pro další běh algoritmu. Poté je mapa průběžně doplňována a rozšiřována, jak jsou nové klíčové snímky vkládány sledovacím systémem. Operace mapovacího stádia jsou na obr. 2.9

Tato metoda je velice přesná pro získávání pozice kamery v malé scéně. Pro větší scény je nevhodná. S dalším průzkumem prostředí narůstá počet klíčových snímků a čas výpočtu se zvyšuje, čímž metoda přestává stíhat pracovat v reálném čase. Navržena je pro zobrazování rozšířené reality v malé neměnné scéně, například na pracovním stole. Jak autoři sami



Obrázek 2.9: Mapování. Po inicializaci se smyčka nekonečně opakuje a příležitostně získá nový snímek ze sledovacího systému. [13]

uvádějí, přesnost získávání pozice kamery je mnohem vyšší, než v případě MonoSlam a také se lépe vyrovnává s rychlejšími změnami pozice a rotace kamery ve všech směrech.

Kapitola 3

Významné body

Významný bod (angl. feature point nebo interest point) - jedná se o bod, resp. body, které nás v obrázku zajímají a jsou pro jeho další zpracování důležité. Např. se jedná o rohy, nebo body, které se dají znovu dobře vyhledat. To je jejich základní vlastnost, možnost znovu vyhledat ten samý bod i v jiném obrázku, kde je scéna jinak pootočená.

Významný bod má následující charakteristiky[22][8]:

- Přesnou, matematicky dobře podloženou definici.
- Přesně definovanou pozici v obrazovém prostoru.
- Lokální strukturu v obraze kolem významného bodu bohatou na informace vhodné pro pozdější zpracování vizuálním systémem.
- Je stabilní z hlediska působení lokálních a globálních deformací v obrazové doméně tak, aby byl bod opět nalezen s vysokým stupněm opakovatelnosti.
- Je vhodné, aby obsahoval atribut měřítka pro možnost vypočítat ho z reálné změny obrázku stejně jako s měnícím se měřítkem.

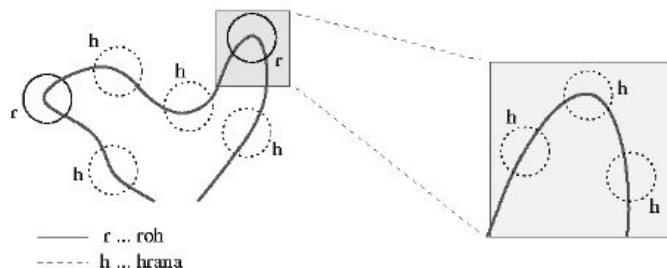
Detektorů významných bodů je poměrně hodně. Mezi nejznámější patří Harrisův detektor a SIFT¹.

3.1 Harrisův detektor

Jedná se o detektor významných bodů, který vyhledává rohy a hrany. Byl vytvořen v roce 1988 Chrisem Harrisem a Mikem Stephensem. Vychází z Moravcova operátoru, ale liší se od něj v používání takzvané lokální autokorelační funkce[17]. Použitím této funkce dosahuje Harrisův detektor výborných výsledků. Je nezávislý na rotaci a posunu a odolný vůči šumu. Nedokáže si však poradit se změnou měřítka. Jak je vidět na obrázku 3.1. Ten znázorňuje problém při vyhledávání významných bodů. Jsou zkoumány body znázorněné křivky. Kolečka představují tzv. kruhová okénka, pomocí kterých se vyhodnocuje zakřivení v okolí bodu. Pomocí nepřerušovaných kroužků se znázorňují body vyhodnocené jako významné. Křivka z těchto okének vystupuje pod výrazně jiným směrem, než vstupuje. Všechny ostatní body křivky jsou vyhodnoceny jako hrany. V pravé části je část stejné křivky, ale je znázorněna ve větším měřítku. Je na ní použit opět Harrisův detektor se stejným okénkem

¹ SIFT - Scale-invariant feature transform, autorem je David Lowe

jako v prvním případě. Díky většímu měřítku už se ale směr křivky uvnitř okénka nikde nezmění natolik, aby byl bod vyhodnocen jako roh. Body vyhledané pomocí měřítkově závislých metod tedy nelze použít pro porovnávání struktur s různými měřítky, neboť se po změně měřítka v obrázku vyhledají jiné struktury[8].



Obrázek 3.1: Znázornění problému vyhledávání významných bodů pomocí měřítkově závislého Harrisova detektoru[8].

3.2 SIFT

SIFT - Scale-invariant feature transform, jehož autorem je David G. Lowe, je jedním z nej-používanějších detektorů významných bodů v současnosti. Zmínka je o něm i ve většině prací zabývajících se počítačovým viděním. Je popsán v článku [15]. Zde uvedu jen základní vlastnosti a popis. Každý významný bod, který se pomocí SIFTu najde, dostane přiřazen tzv. deskriptor. Jedná se o vektor 128 hodnot celých čísel. Deskriptory jsou invariantní vůči následujícím operacím s obrazem[8]:

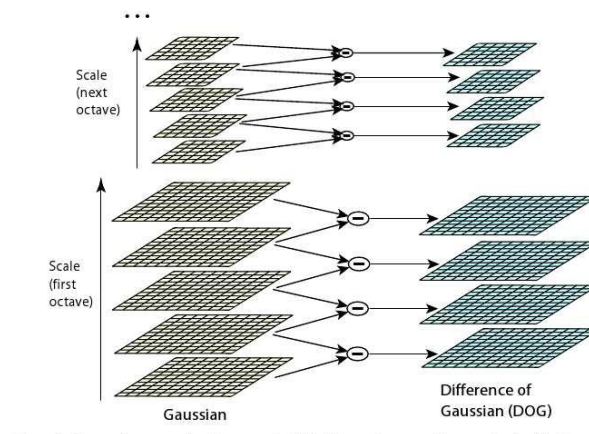
- Zvětšení, zmenšení - tedy změna měřítka.
- Posun, 2D rotace.
- Změna kontrastu, jasu, osvětlení, šum.
- Částečně invariantní vůči prostorové změně úhlu pohledu.

Proces získávání významných bodů z analyzovaného obrázku lze rozdělit do několika fází[8]:

1. Sestrojení scale-space a vyhledání lokálních extrémů:

V první fázi výpočtu se musí zkontrolovat měřítkově nezávislá reprezentace obrázku, tzv. scale-space. V něm se dále vyhledávají přes všechna měřítka lokální extrémů. Pro zvýšení výkonnosti je pro generování scale-space použita místo Laplacianu Gaussovy funkce (LoG) jeho aproximace pomocí rozdílu Gaussových funkcí (DoG). Rozdíl Gaussových funkcí je získán odečtením dvou na sobě nezávislých Gaussových funkcí o různém měřítku. Scale-space se tedy generuje konvolucí s obyčejným Gaussem a měřítkově nezávislá forma se získá později rozdílem po sobě jdoucích vrstev viz. obr. 3.2.

Možné významné body leží v lokálních extrémech scale-space. Vyhledání probíhá na všech vrstvách kromě krajních. Každý bod je porovnán se svými sousedy a to i ze dvou okolních vrstev. U nalezených extrémů se zmapují jejich souřadnice a postupují do další fáze.



Obrázek 3.2: Konstrukce scale-space pomocí DoG [8].

2. Přesnější lokalizace klíčových bodů:

Body získané v první fázi se podrobí podrobnějšímu zkoumání. Upřesnění bodu je provedeno pomocí aproximace funkce popisující jeho okolí, jak v souřadnicích, tak v měřítku. Pokud extrém leží blíže sousednímu bodu, je tento bod za něj vyměněn a aproximace probíhá znovu.

3. Odstranění hranových bodů:

Použitím DoG reprezentace vytváří extrémy i na hranách. Body jsou však v tomto případě špatně lokalizované a je třeba je odstranit. Využívá se druhé derivace. Body kolem hran budou mít velkou hlavní křivost kolmo na hranu a malou podél hrany. Hlavní křivost v okolí bodu lze vyjádřit pomocí vlastních čísel. Pokud bod bude ležet v sousedství hrany, potom bude jedno z vlastních čísel výrazně větší, než druhé[8].

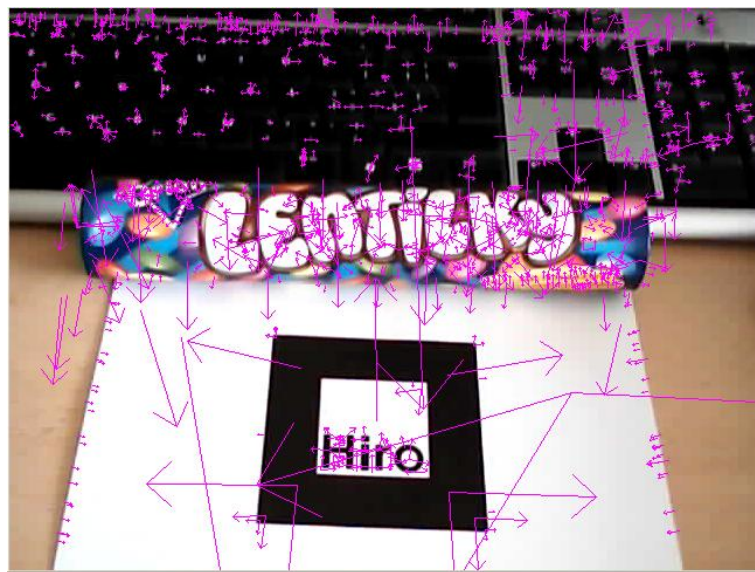
4. Přiřazení orientací:

Každému významnému bodu je přiřazena dominantní orientace na základě orientací gradientů² v okolí bodu. Dominantní orientace je získána z histogramu orientací již zmíněných gradientů jako vrcholek tohoto histogramu. Pokud jsou detekovány další vrcholky s hodnotou více jak 80% hodnoty maximální, tak pro každý takový vrcholek je vyroben nový významný bod se stejným umístěním bodu prvního, ale s jinou orientací. Tento postup je potřebný pro zjištění nezávislosti na rotaci.

5. Tvorba deskriptorů:

Deskriptor obsahuje informace o vzhledu obrazového okolí vypočítaného významného bodu. Okolí musí být tedy popisováno způsobem odolným proti světelným a geometrickým změnám. Pro popis okolí jsou použity již v předchozím kroku vypočítané gradienty. Na základě orientací gradientů v okolí bodu je pro každý významný bod vypočten deskriptor. Pomocí něj jsou potom porovnávány významné body mezi analyzovanými snímky.

²Gradient je diferenciální operátor, jehož výsledkem je vektorové pole vyjadřující směr a velikost změny skalárního pole.

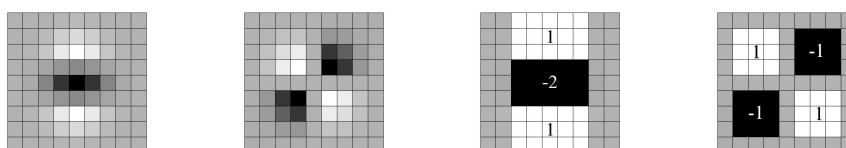


Obrázek 3.3: Ukázka vyhledání významných bodů pomocí SIFTu

3.3 SURF

SURF - Speeded Up Robust Features je název metody vyhledávající korespondující body mezi dvěma snímky popsané v práci autora Baye [5]. Autor se kromě schopnosti vyhledávání významných bodů invariantních k posunu, rotaci a změnám měřítka zaměřuje, jak je zřejmé z názvu, na rychlost vyhledávání. Metoda z části vychází z metody SIFT popsané v předchozí kapitole a autoři ji použili jako inspiraci hlavně pro výpočet deskriptorů.

Detektor je postaven na aproximaci Hessian matice. Scale-space je generováno přímo pomocí Hessianu. Jak autoři uvádějí, pracuje nad integrálním obrazem, takže tuto metodu nazvali Fast-Hessian detektor. Integrální obraz je použit hlavně z důvodu urychlení celého výpočtu. Umožňuje totiž rychlý výpočet obdélníkových konvolučních filtrů. Hodnota integrálního obrazu $I_{\Sigma}(x)$ na pozici $x = (x, y)$ představuje sumu hodnot všech pixelů vstupního obrazu I v obdélníkové oblasti určené bodem x a počátkem. S vypočítaným integrálním obrazem stačí jen čtyři součty pro výpočet sumy jakékoliv svislé obdélníkové oblasti [5].



Obrázek 3.4: Zleva doprava: Gaussova druhá parciální derivace ve směru y a xy a aproximace pomocí čtvercové oblasti [5].

Dále jsou použity čtvercové oblasti rozměru 9x9 viz. obr. 3.4, které aproximují Gaussovu druhou derivaci a reprezentují nejmenší měřítko. Odezvy filtrů jsou dále normalizovány s ohledem na velikost čtvercové oblasti. To zaručuje konstantní Frobeniovu normu pro jakoukoliv velikost filtru.

Narozdíl od vytváření scale-space jako pyramidy obrazu, kdy je každá vrstva vyhlazována Gaussovou funkcí a podvzorkována pro získání vyšší vrstvy pyramidy, není u této metody nutný iterativní postup, díky integrálnímu obrazu a obdélníkovým filtrům. Je tedy možno každou vrstvu pyramidy vytvářet nezávisle na té předchozí z původní vrstvy.

Lokalizace bodů probíhá vyhledáním lokálního maxima stejně jako u metody SIFT, kdy je každý bod scale-space porovnáván s 26 okolními sousedy a pokud je největší, jedná se o významný bod.

Pro zajištění nezávislosti metody na rotaci obrázku je nalezeným významným bodům přiřazena orientace. Postup je opět podobný jako u metody SIFT, ale ze získaných orientací kolem významného bodu je brán v úvahu pouze nejdominantnější směr a ostatní se zanedbávají. Dominantní směr se určuje na kruhovém okolí významného bodu o daném poloměru[8].

Deskriptory u metody SURF jsou podobné jako u metody SIFT, opět se jedná o popis okolí bodu pomocí gradientů. Rozsah popisu bodu je tentokrát 64 hodnot, rozšířený popis má však stejně hodnot jako metoda SIFT, tedy 128. Konstrukce deskriptoru probíhá ze čtvercové oblasti okolí bodu, která je natočena podle orientace bodu. Rozsah popisu deskriptoru bodu má 64 hodnot a byl zvolen opět kvůli rychlosti při vyhledávání korespondencí.

3.4 Korespondence významných bodů

Jak bylo řečeno v předchozím textu, odpovídající si body na dalších obrázcích mají stejné (nebo velmi podobné) deskriptory. Pokud tedy hledáme podobné body, porovnáváme deskriptory jednotlivých bodů pomocí Euklidovské vzdálenosti. Dva body s nejmenší vzdáleností by si pak měly odpovídat a mělo by se jednat o korespondence. V praxi se však používá mírně odlišný výpočet a to takový, že se najdou pro jeden bod dva odpovídající body s nejmenší Euklidovskou vzdáleností. Pokud je tato vzdálenost v poměru menším jak 0.7 jedná se u nejmenší vzdálenosti s velkou pravděpodobností o korespondenci. Hodnota 0.7 byla zjištěna testováním jako hodnota s nejlepšími výsledky. Pokud vezmeme v potaz velký počet vyhledaných bodů ve snímku a všechny tyto body se musejí porovnat se všemi ve druhém snímku, ve kterém se hledají korespondence, využívá se pro vyhledávání algoritmus nejbližšího souseda pomocí KD-stromů. Tímto by mělo dojít ke snížení výpočetní náročnosti.

Kapitola 4

Kamera

V kapitole je nejprve uvedena perspektivní projekce následovaná popisem vnitřních a vnějších parametrů kamery. Dále představena projekční matice kamery. Poslední částí je popis výpočtu projekční matice.

4.1 Perspektivní projekce

Perspektivní projekce je nelineární neafinní projekce, která zobrazuje vrcholy promítaných objektů prostřednictvím paprsků protínajících se v jednom bodě, ve středu projekce. Střed projekce je zároveň většinou místem pozice pozorovatele. Velikost průmětů objektů je nepřímouměrně závislá na jejich vzdálenosti od průmětny. Čím je objekt blíže u průmětny, tím je jeho obraz větší a naopak. Rovnoběžnost promítaných hran není v této projekci zachována[14].

Perspektivní projekce odpovídá našemu lidskému vidění reálného světa. Proto se nejčastěji používá pro realističtější zobrazení ve virtuální realitě, architektuře, počítačových 3D hrách, atd.

Perspektivní projekci můžeme vyjádřit prostřednictvím transformace s maticí P_{per} (viz. vztah 4.4 a 4.5). Její odvození, vycházející z parametrického vyjádření promítacího paprsku, najdeme ve vztazích 4.1 - 4.3 (viz. obr. 4.1).

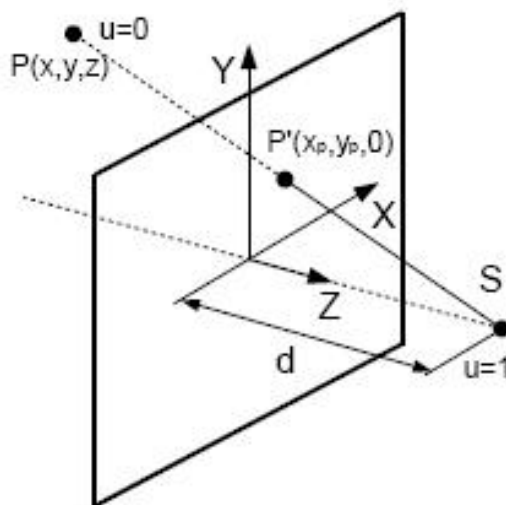
$$x = x - xu, \quad y = y - yu, \quad z = z - (z + d)u \quad (4.1)$$

$$u \in \langle 0, 1 \rangle, \quad u = 0 \rightarrow (x, y, z), \quad u = 1 \rightarrow (0, 0, d) \quad (4.2)$$

$$z_P = 0 \rightarrow u = \frac{z}{z + d}, \quad x_P = x \frac{1}{1 + \frac{z}{d}}, \quad y_P = y \frac{1}{1 + \frac{z}{d}} \quad (4.3)$$

$$(x_P, y_P, z_P, w_P)^T = P_{per} \cdot (x, y, z, w)^T, \quad w_P = 1 + \frac{z}{d} \quad (4.4)$$

$$P_{per} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{d_z} & 1 \end{bmatrix} \quad (4.5)$$



Obrázek 4.1: Schéma odvození perspektivní projekce [14]

4.2 Vnitřní parametry kamery

Jedná se o parametry, které jsou nutné pro převod ze souřadnic obrázku do souřadnic pixelů obr. 4.2. Parametry charakterizují optický, geometrický a číselný model kamery. Matice vnitřních parametrů má rozměr 3x3:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (4.6)$$

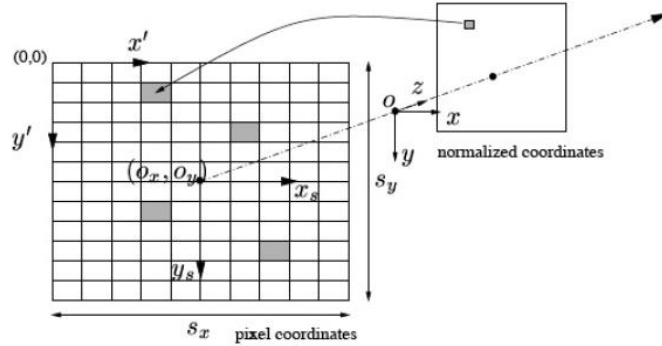
Matice K se nazývá *matice vnitřních parametrů* (*angl. intrinsic camera matrix*), kde c_x a c_y jsou souřadnice základního bodu (většinou se jedná o střed snímku), f_x a f_y jsou ohniskové vzdálenosti uváděné v jednotkách vycházejících z pixelů. Matice vnitřních parametrů kamery je nezávislá na scéně a jako taková může být používána od jejího výpočtu až do změny ohniskové vzdálenosti. [11].

Dále lze f_x a f_y vypočítat také pomocí vzorců:

$$f_x = \text{ohnisková vzdálenost}[mm]/(\text{šířka pixelu}[um]/10^3)$$

$$f_y = \text{ohnisková vzdálenost}[mm]/(\text{výška pixelu}[um]/10^3)$$

Pokud je matice vnitřních parametrů známa, je možné provést přesnou rekonstrukci. Tedy získat přesnou polohu 3D objektu a ne pouze jeho podobnost. Tím je myšleno, že je možné zrekonstruovat jeho přesné rozměry.



Obrázek 4.2: Transformace z obrazových souřadnic do pixelu[24]

4.3 Vnější parametry kamery

Vnější parametry kamery popisují vztah mezi světovým souřadným systémem a souřadným systémem kamery. Jedná se o rotační matici R a vektor posunutí t .

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (4.7)$$

$$t = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} \quad (4.8)$$

4.4 Projekční matice

Vztah mezi bodem M ve 3D a jeho projekcí do bodu m je popsán rovnicí.

$$m = K[Rt]M, \quad (4.9)$$

kde K je matice vnitřních parametrů kamery, (R, t) jsou vnější parametry kamery: rotační matice R a vektor posunu t , které převádějí světový souřadný systém do souřadného systému kamery[11].

4.5 Výpočet projekční matice

V této podkapitole je rozebrán výpočet projekční matice ze známých bodů v prostoru a jejich odpovídajících bodů v obraze. Výpočet vychází z práce Carlose Viala [20].

Máme-li zadané správně určené 3D body v prostoru, lze souřadnice 3D bodů společně s jejich korespondencemi ve snímku použít pro výpočet projekční matice. Jsou-li souřadnice bodů v prostoru homogenní, vzhledem k souřadnému systému scény v milimetrech, a body obrazu homogenní, určené v pixelech, vztahené k jednomu rohu obrazu, lze zapsat vztah mezi těmito dvěma množinami bodů projekční maticí M .

$$q = M \cdot p \quad (4.10)$$

$$\begin{bmatrix} U \\ V \\ S \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ T \end{bmatrix} \quad (4.11)$$

Pro větší přehlednost je zavedena notace, kdy projekční matice je vyjádřena jako:

$$M = \begin{bmatrix} m_1 & m_{14} \\ m_2 & m_{24} \\ m_3 & m_{34} \end{bmatrix} = \begin{bmatrix} m^1 \\ m^2 \\ m^3 \end{bmatrix} \quad (4.12)$$

kde m^1, m^2, m^3 jsou řádky matice M a m_1, m_2, m_3 jsou 1×3 řádkové vektory projekční matice M bez posledního členu. Vztah mezi souřadnicemi bodů a členy projekční matice je znám jako *přímá lineární transformace (angl. Direct Linear Transformation DLT)*[20]. 3D bod p s odpovídajícím bodem q ve snímku má vztah $q - M \cdot p$. Z toho plyne, že vektor q a výsledek $M \cdot p$ jsou identické. Jelikož jsou uváděny v homogenních souřadnicích, tato podmínka je spojená se směrem orientace vektoru, ale ne s jeho modulem. Pro zaručení platnosti podmínky musí být jejich vektorový součin roven nule.

$$q \times Mp = 0 \quad (4.13)$$

Vezmeme-li v potaz, že třetí řádek je kombinací předchozích dvou, existují jen dvě lineární nezávislé rovnice pro každý pár bodů. Vznikne tak následující výraz:

$$\begin{bmatrix} 0 & -Sp^T & Vp^T \\ Sp^T & 0 & -Up^T \end{bmatrix} \cdot \begin{bmatrix} m^{1T} \\ m^{2T} \\ m^{3T} \end{bmatrix} = 0 \quad (4.14)$$

Tento výraz se dá zapsat jako $A \cdot m = 0$, kde vektor m obsahuje členy projekční matice sestavené do sloupcového vektoru. A je pak matice složená ze souřadnic bodů v obraze a ve scéně, které jsou známy. Pro získání soustavy nezávislých lineárních rovnic by body scény neměly ležet v jedné rovině.

Jedním ze způsobů řešení rovnic $A \cdot m = 0$ je převod rovnice 4.14 na nehomogenní rovnici. Toho lze dosáhnout předpokladem, že $m_{34} = 1$. Jelikož m_{34} je translace kamery v souřadné ose Z , nestává se, že by byla rovna nule. Obecně, pokud je $m_{34} = 1$ výraz 4.14 se převede do formy $A' \cdot m' = b$, kde A' je matice A bez posledního sloupce a b je právě poslední sloupec A . m' je sloupcový vektor jen o jedenácti prvcích projekční matice, protože předpokladem je, že dvanáctý prvek je roven jedné. Pro výpočet jedenácti prvků matice je využita metoda nejmenších čtverců. Prvních jedenáct prvků matice je vypočteno jako:

$$m' = (A'^T \cdot A')^{-1} A'^T \cdot b \quad (4.15)$$

Poslední, tedy dvanáctý prvek matice je roven 1.

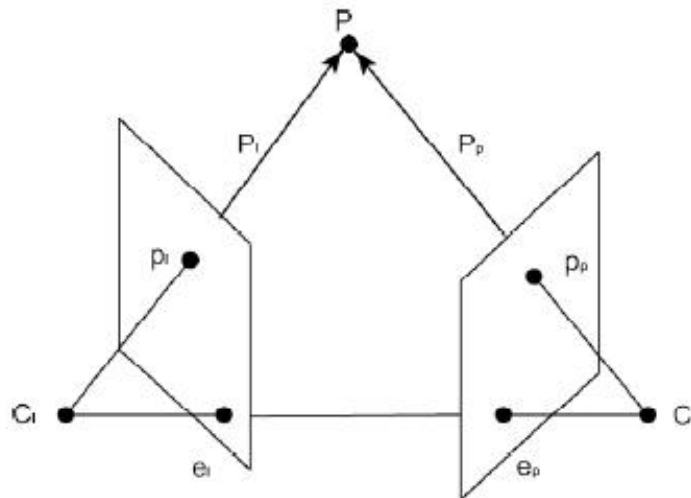
Kapitola 5

Rekonstrukce 3D bodu

Kapitola představuje základy nutné pro výpočet bodů z korespondencí. Nejprve je uvedena teorie fundamentální matice, následovaná popisem triangulační metody.

5.1 Fundamentální matice a epipolární geometrie

Epipolární geometrie popisuje základní geometrický vztah mezi dvěma perspektivními kamerami nezávisle na pozorované scéně. V následující části popíšeme jen základní vztahy mezi dvěma kamerami.



Obrázek 5.1: Epipolární geometrie dvou pohledů [7].

Na obr. 5.1 vidíme dvě kamery určené středy promítání C_i , C_p a příslušnými projektivními rovinami, bod v prostoru P a jeho projekce na levé a pravé průmětně p_p , p_i [7].

Epipolární geometrie je algebraicky reprezentovaná esenciální maticí E a z ní vycházející fundamentální maticí F . Esenciální matice je definována vztahem:

$$P_p^T E P_i = 0, \quad (5.1)$$

kde P_p, P_l jsou zápisy bodu P v souřadných systémech levé a pravé kamery. Pokud známe transformaci mezi souřadnými systémy kamer:

$$P_p = R(P_l - T) \quad (5.2)$$

můžeme matici E vyjádřit vztahem:

$$E = SR, \quad (5.3)$$

kde R je matice rotace viz. 5.2 a

$$S = \begin{pmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{pmatrix}, \quad (5.4)$$

kde T_x, T_y, T_z jsou prvky vektoru T viz. 5.2. Fundamentální matice je definována vztahem:

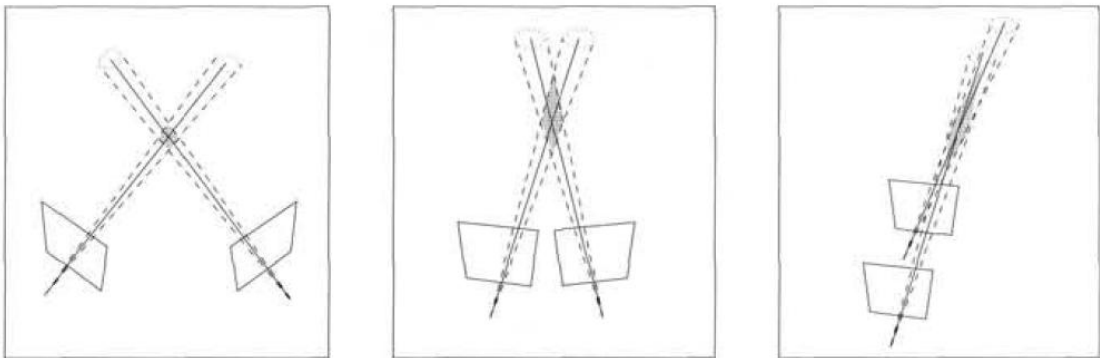
$$p_p^T F p_l = 0, \quad (5.5)$$

kde p_p, p_l jsou projekce bodu P na průmětny pravé a levé kamery v pixelových souřadnicích[7].

5.2 Lineární triangulační metoda

V této kapitole popíšeme lineární triangulační metodu pro výpočet bodu v prostoru. Tuto metodu využívám v programové části.

Lineární triangulační metoda se používá pro výpočet 3D bodu v prostoru ze dvou navzájem korespondujících bodů ve dvou snímcích scény při znalosti projekčních matic obou snímků. Z obrázku 5.2 plyne i úskalí metody, kdy je možný nepřesný nebo špatný výpočet, pokud je posun kamery mezi dvěma snímky malý a nebo při pohybu kamery pouze dopředu či dozadu.



Obrázek 5.2: Nepřesnost rekonstrukce. Stínovaná oblast v každém z příkladů ilustruje nepřesnost, která závisí na úhlu mezi kamerami. Bod je čím dál méně přesně vypočítán, čím více jsou směry kamery rovnoběžné [10].

Navzájem korespondujícím bodům x a x' ze dvou obrazů odpovídají rovnice $x = PX$, $x' = P'X$, kde P a P' jsou projekční matice dvou kamer. Tyto rovnice mohou být zkombinovány do nové rovnice $AX = 0$, jenž je lineární rovnicí pro X .

Jako první je odstraněn homogenní měřítkový faktor pomocí vektorového součinu daných tří rovnic pro každý bod obrázku, z čehož dvě jsou lineárně nezávislé. Například pro první obrázek, $x \times (PX) = 0$ rozepsáním vzniká:

$$x(p^{3T}X) - (p^{1T}X) = 0 \quad (5.6)$$

$$y(p^{3T}X) - (p^{2T}X) = 0 \quad (5.7)$$

$$x(p^{2T}X) - y(p^{1T}X) = 0, \quad (5.8)$$

kde p^{iT} jsou řádky matice P . Tyto rovnice jsou lineárními komponenty bodu X . Rovnice je poté ve formě

$$AX = 0 \quad (5.9)$$

kde A lze sestavit jako:

$$A = \begin{bmatrix} xp^{3T} - p^{1T} \\ yp^{3T} - p^{2T} \\ x'p'^{3T} - p'^{1T} \\ y'p'^{3T} - p'^{2T} \end{bmatrix} \quad (5.10)$$

a z každého snímku náleží A dvě rovnice, celkově tedy čtyři rovnice pro čtyři neznámé.

Rovnici 5.2 lze řešit dvěma způsoby. Prvním je homogenní řešení pomocí SVD, pak $A = UWV^T$ a řešením je poslední sloupec matice V^T .

Druhé řešení je nehomogenní, kdy předpokládáme $X = (X, Y, Z, 1)^T$ a ze soustavy rovnic se stává soustava přeúčená, která se vyřeší metodou nejmenších čtverců. Tento postup je nevhodný pro body, kde se poslední souřadnice v základním řešení blíží nule a nahrazení za 1 může způsobovat nepřesný výpočet[10].

5.3 SVD

Singulární dekompozice matice (angl. Singular Value Decomposition), jedná se o rozklad obdélníkové reálné nebo komplexní matice, který nalézá uplatnění například ve zpracování signálu a statistice. Použijí ji k řešení lineárních rovnic. Další použití podle [23] je např.:

1. výpočet pseudoinverzní matice
2. řešení homogenních lineárních rovnic
3. minimalizace pomocí metody nejmenších čtverců
4. aproximace matic

Kapitola 6

Návrh řešení

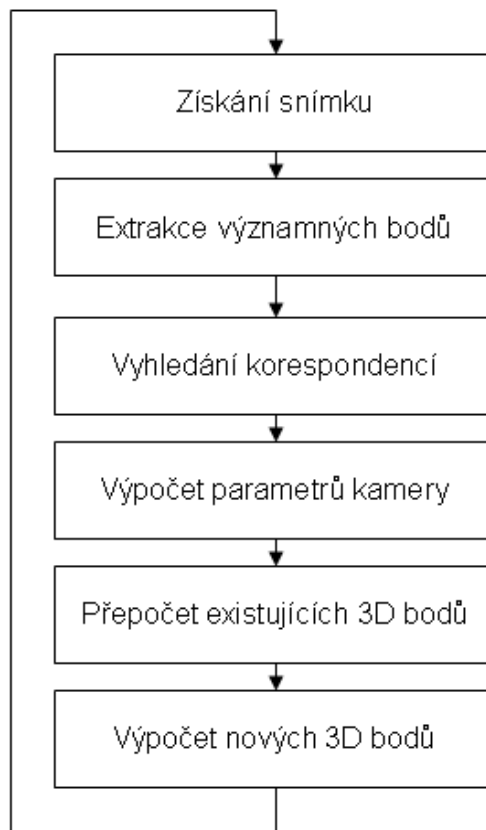
V této kapitole je popsán návrh řešení vycházející z kombinací již existujících přístupů k problematice 3D rekonstrukce. Výsledný program řeší rekonstrukci scény nebo její části z video sekvence a výsledkem je zrekonstruovaný povrch. Návrh se dá shrnout do následujících bodů a jeho zjednodušující schéma je na obrázku 6.1:

- Výpočet vnitřních parametrů kamery.
- Získání jednotlivých snímků z kamery nebo načítání ze souboru.
- Nalezení významných bodů v obraze ve snímcích pomocí různých detektorů Harris, SIFT, SURF.
- Výpočet korespondencí bodů mezi snímky scény.
- Využití epipolární geometrie na odstranění falešných korespondencí.
- Výpočet pozice kamery pomocí již zrekonstruovaných bodů, nebo na základě zadaného markeru.
- Rekonstrukce 3D bodů triangulační metodou.
- Vyřazení chybně spočítaných bodů zpětnou projekcí.
- Výpočet normály každého bodu.
- Rekonstrukce povrchu metodou Poisson.

6.1 Výpočet vnitřních parametrů kamery

Ještě před samotným přistoupením k rekonstrukci je nutné mít zkalibrovanou kameru, která bude použita pro 3D rekonstrukci snímané scény. Správně zkalibrovaná kamera je předpokladem pro geometricky dobrý výsledek rekonstrukce scény. Při snímání scény je počítáno se zafixovanými vnitřními parametry kamery, tedy že se nebude měnit zaostření a zůstane stabilní pro celou dobu rekonstrukce.

Výpočet bude proveden externím programem, ještě před samotným spuštěním rekonstrukce. Navrhovaný program bude předpokládat soubor s vnitřními parametry kamery na vstupu jako jeden z parametrů.



Obrázek 6.1: Zjednodušený návrh řešení.

6.2 Získání snímků

Snímky budou získávány buď z kamery nebo z video souboru. Zpracovává se vždy jeden snímek za druhým, přičemž se pokaždé přeskočí určitý počet snímků, kvůli urychlení výpočtů a z důvodu malé změny scény mezi jednotlivými za sebou jdoucími snímky. Bude možno měnit počet snímků, které se mají z rekonstrukce vynechat. Možnost měnit počet vynechaných snímků před dalším zpracovávaným snímkem povede ke schopnosti upravit rekonstrukci podle aktuálně zpracovávaného videa. Pokud je pohyb videokamery rychlý a dochází tedy k rychlé změně umístění kamery ve scéně, bude potřeba snížit počet vynechaných snímků. Naopak pokud je pohyb kamery pomalý a nedochází k rychlé změně pozice kamery ve scéně, bude se moci nastavit větší počet neaktuálních snímků, a tím snížit náročnost výpočtu.

Pro získávání snímků je počítáno s využitím funkcí některé knihovny ARtoolkit, OpenCV nebo některých dalších.

6.3 Nalezení významných bodů v obraze

Nalezení významných bodů ve snímku bude jednou ze stěžejních částí navrhovaného systému. Na správné lokalizaci významných bodů, tak aby byly stabilní i v dalších snímcích a znovu vyhledatelné podle použitého deskriptoru, závisí přesnost rekonstrukce. Musí být získáno velké množství bodů, aby bylo možno získat i velké procento bodů korespondujících v předchozích a následujících snímcích.

Mezi často používaný detektor významných bodů v pracích zabývajících se tematikou 3D rekonstrukce patří SIFT kap.3.2. Body nalezené tímto postupem jsou stabilní pro rotace i změny měřítka. Díky robustnímu deskriptoru okolí významného bodu není složité nalézt korespondence i na ostatních snímcích. Nepočítá se s implementací celého algoritmu, ale využití již existujících knihoven pro OpenCV postavených na algoritmu a postupech popsaných autorem SIFTu D. Lowem.

Dalším detektorem významných bodů použitým v navrhovaném systému bude SURF viz. kap. 3.3. Tento detektor je podobný již zmiňovanému SIFT, je invariantní vůči rotacím a změnám měřítka. SURF dosahuje podobných výsledků jako SIFT, ale v kratším čase.

V systému bude možnost vyhledávat body jednou z vybraných možností SIFT nebo SURF. Nepočítá se však s vyhledáváním významných bodů oběma metodami zároveň.

V případě úspěšné implementace celého systému a dostatku času bude další možností využít jednoduššího vyhledávače významných bodů, například Harrisova detektoru kap. 3.1, který vyhledává rohy ve zpracovávaném obraze. Popisem okolí bodu pak budou samotné pixely nalézající se v těsné blízkosti. Vyhledávání bodů proběhne v každém snímku podle pohybu kamery až do jejich posunu mimo zorné pole kamery. Bude tak možno rekonstruovat rohy a kraje objektů ve scéně a dosahovat vizuálně lepší rekonstrukce s využitím i při rekonstrukci povrchu.

6.4 Vyhledávání korespondencí

Pro vyhledání korespondujících významných bodů budou využity deskriptory těchto bodů vypočítané již při lokalizaci metodami SIFT nebo SURF. Rozsahy deskriptorů obou metod budou stejné velikosti, která činí 128 hodnot. Tím se docílí jednoduššího postupu záměny SIFT za SURF v implementaci výsledného systému.

Korespondence bodů bude získávána tak, že každý významný bod právě zpracovávaného snímku bude porovnán se všemi body ze snímku předcházejícího. Jako korespondující body budou označeny ty, které mají minimální euklidovskou vzdálenost svých deskriptorů. Zároveň poměr mezi minimální a druhou nejmenší vzdáleností je menší, než zadaná konstanta. Tato konstanta bude zvolena na základě výsledků při tvorbě systému. Tímto postupem vzniknou dvě množiny navzájem si odpovídajících bodů pro dva snímky. Po tomto kroku mohly však vzniknout některé falešné korespondence, které je nutné odstranit.

Korespondence významných bodů nebudou vyhledávány jen oproti předchozímu snímku, ale také proti již zrekonstruovaným bodům. Toto vyhledávání je prováděno kvůli získání korespondencí bodů v obraze s body v prostoru, díky čemuž může být v následujících krocích zjištěna poloha kamery v prostoru a projekční matice. Dále vyhledáním korespondencí bodů ve snímku a prostoru dojde k přepočtu umístění bodu v prostoru a jeho zpřesnění.

6.5 Odstranění falešných korespondencí

Odstranění falešných korespondencí bude nutné ke správné rekonstrukci, jinak by docházelo k chybám ve výpočtech. Stejně tak je odstranění chybných korespondencí důležité pro správnou funkci zjišťování pozice kamery a projekční matice.

Při zjišťování správnosti korespondencí bude využité epipolární geometrie, kdy korespondující bod musí ležet na epipolární přímce vypočítané na základě fundamentální matice a bodu z prvního snímku kap. 5.1.

Fundamentální matice bude vypočítána 8-bodovým algoritmem s využitím metody RANSAC, která by měla zaručit správný výsledek pro většinu bodů. Po odfiltrování chybných korespondencí zůstanou jen stabilní korespondence využitelné pro rekonstrukci scény a výpočet vnějších parametrů kamery.

6.6 Výpočet pozice kamery na základě markeru

K výpočtu pozice kamery na základě markeru nepotřebujeme předem znát žádné vypočítané body v prostoru a stačí nalézt ve snímku kamery známý vzor, podle kterého se již pozice kamery v prostoru vypočítá.

Vzor bude ve scéně jeden, přičemž by měl být pro rekonstrukci scény využíván jen v počátečních fázích snímání videa, kdy ještě nebyly nalezeny a zrekonstruovány žádné nebo málo bodů v prostoru pro možnost výpočtu umístění kamery přímo. Pro zjištění pozice kamery na základě známého vzoru ve scéně bude využito již existujících řešení a markeru některé z knihoven OpenCV nebo ARtoolkit.

6.7 Výpočet pozice kamery z již dříve zrekonstruovaných bodů

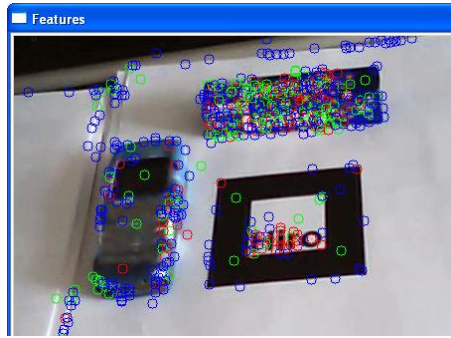
Jak je uvedeno v kap. 4.5 lze zjišťovat pozici kamery na základě již zrekonstruovaných bodů v prostoru, které jsou znovu lokalizovány v nově zpracovávaném snímku kamery. Toho bude využito i v navrhovaném systému rekonstrukce 3D scény.

Ve zpracovávaném snímku budou vyhledány významné body, které byly již jednou rekonstruovány. Navíc se musí jednat o body, které mají pozici v prostoru správně vypočítanou. Pokud by byly zvoleny špatné body, docházelo by i k chybě výpočtu kamery, která by mohla postupně začít narůstat.

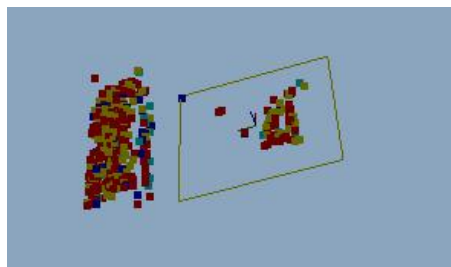
Z tohoto důvodu budou po každém vypočtení pozice kamery nalezené významné body v daném snímku, které odpovídají již některým zrekonstruovaným bodům prověřeny pro jejich přesnost a pokud odpovídají, budou označeny jako vhodné pro výpočet pozice kamery v dalším cyklu programu. Na obr. 6.2 je možné je vidět jako červeně označené významné body, na obr. 6.3 jako žluté a světle modré.

Jestliže bude nalezeno dostatečně, nejméně však 6 bodů v prostoru, kterým odpovídají body v právě zkoumaném snímku podle $x = P * X$, kde x jsou souřadnice bodu ve snímku, X jsou souřadnice bodu v prostoru a nakonec P je neznámá projekční matice. Lze poté sestavit soustavu lineárních rovnic a řešit například metodou nejmenších čtverců jak je uvedeno v práci [20] a popsáno v kap. 4.5.

Výsledkem bude jednorozměrné pole jedenácti prvků, které lze jednoduše rozdělit do projekční matice 3×4 a poslední prvek je roven 1.



Obrázek 6.2: Významné body nalezené v obraze. Zeleně označené korespondence s předchozími snímky, červeně korespondence již rekonstruovaných bodů.



Obrázek 6.3: Část rekonstruovaných bodů v prostoru. Body používané pro výpočet projekční matice jsou žluté a světle modré.

6.8 Výpočet nových 3D bodů

Další stěžejní částí návrhu je samotná rekonstrukce jednotlivých bodů, tedy výpočet pozice nalezeného významného bodu snímku v prostoru. V systému bude využita triangulační metoda, která potřebuje pro rekonstrukci bodu dva korespondující body ve dvou snímcích. Tyto snímky, ve kterých se korespondující body nacházejí, by měly mít v prostoru od sebe větší vzdálenost, jak je uvedeno v kap. 5.2, aby výsledný vypočtený bod byl co nejpřesnější.

Při výpočtu pomocí triangulační metody budou lineární rovnice řešeny pomocí SVD.

Dále pro každý takto rekonstruovaný bod bude vypočtena i orientace tohoto bodu. Orientace bude určena jako směrový jednotkový vektor od bodu v prostoru k pozici kamery, vše ve světovém souřadném systému. Nebude se jednat o normálu bodu, ale jen o předběžnou orientaci pro další výpočty.

Při triangulační metodě bude jistě docházet k chybám ve výpočtu, které mohou být způsobeny chybnou korespondencí, nebo nepřesnostmi v určení polohy počítaného bodu v prostoru, například i špatným výpočtem projekční matice v předcházejícím bodu návrhu.

Spočítané 3D je tedy nutné ještě ověřit zpětnou projekcí, jestli nedošlo k již zmíněným chybným výpočtům bodu. Bod bude pomocí projekčních matic promítnut ze 3D prostoru zpět do snímků a srovnány odchylky oproti původnímu vyhledaným významným bodům. Pokud je odchylka v daném intervalu, bude bod zařazen do konstruovaného mračna, jinak bude vyřazen.

6.9 Přepoččet existujících 3D bodů

Při nalezení korespondencí na již zrekonstruované body bude docházet ke zpřesňování pozice bodu v prostoru.

Výsledný přepočítaný bod lze řešit znovu pomocí triangulační metody ze dvou bodů a výsledný 3D bod pak počítat jako průměr předchozích vypočtených hodnot. Jedná se o nejjednodušší řešení, ale docházelo by k nepřesnostem při špatně určené pozici kamery a nebylo by jisté, kdy body vyřazovat jako chybně zrekonstruované.

Dalším možným řešením je přeúčtená soustava lineárních rovnic korespondujících bodů, která je řešena metodou nejmenších čtverců.

Ta provede přepoččet bodu na základě přeúčtené matice lineárních rovnic kap. 5.2. Do matice jsou však zahrnuty všechny korespondující body náležící bodu v prostoru.

Výsledný 3D bod bude dále testován zpětnou korespondencí, pokud přidáním nové korespondence dojde u zpětné projekce k překročení povolené meze odchylky, bude celá struktura podrobena novým přepočtům. Pokud se chyba zpětné projekce vejde do odchylky zachová se nově přepočítaný bod jako správný.

V případě chyby dojde k vypočítání zpětné projekce pro všechny body ve struktuře zrekonstruovaného bodu. Největší chyba bude vyhledána a tento bod ze struktury odstraněn jako chybná korespondence a 3D bod bude znovu vypočítán ze zbývajících bodů.

Test zpětnou projekcí se použije ještě jednou pro upravený bod a pokud bude jakýkoliv bod ze struktury označen jako chybný, bude odstraněna raději celá struktura a 3D bod tak zcela smazán.

6.10 Výpočet normály každého bodu

Předpokládaným výstupem celého navrhovaného systému má být zrekonstruovaný povrch snímané scény. V případě použití metody Poisson pro rekonstrukci povrchu, požaduje tato metoda na vstupu orientované body.

Prvotní orientace bodu bude určena pomocí pozic kamer snímků ve kterých byl daný bod pozorován. Vektory od bodu směrem ke kamerám budou dále zprůměrovány, aby byl nalezen výsledný směr předběžné orientace bodu.

Pro výpočet normály bodu bude využito několika nejbližších bodů v okolí. Proložení plochy těmito body bude provedeno metodou nejmenších čtverců a z plochy pak určena normála.

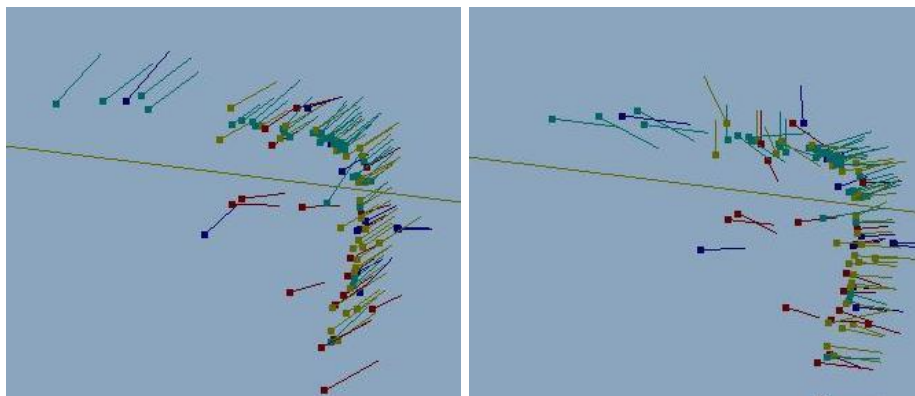
U vypočtené normály bude nutné ještě zjistit její správný směr. Vypočte se úhel podle vzorce 6.1, který svírá normála s prvotní orientací zjišťovanou při rekonstrukci a přepočtu bodu.

$$\cos\varphi = \frac{\vec{u} \cdot \vec{v}}{|\vec{u}| \cdot |\vec{v}|} \quad (6.1)$$

Pokud bude úhel v rozmezí $0^\circ - 180^\circ$ má normála správný směr, jinak bude normála otočena. Obrázek 6.4 ukazuje zrekonstruované body s orientacemi a poté s vypočtenými normálami.

6.11 Rekonstrukce povrchu metodou Poisson

Nakonec po zpracování všech snímků z kamery nebo souboru bude použito zjištěné mračno bodů pro rekonstrukci samotného povrchu kamerou sledované scény. Pro tuto rekonstrukci



Obrázek 6.4: Vlevo jsou zobrazeny orientace bodů směřující ke kameře. Vpravo pak vypočtené normály bodů.

je počítáno s použitím metody Poisson[12], která jako vstupní parametry má orientované body a výstupem jsou trojúhelníky zrekonstruovaného povrchu.

Metodu Poisson je možno implementovat do systému pomocí již existující knihovny nebo využít existujícího programu. Ten jako vstupní soubor předpokládá již zmíněné orientované body a výstupem je pak soubor ve formátu `.ply` obsahující zrekonstruovaný povrch scény.

V první fázi bude využito již existujícího programu. Samotná implementace by se použila až v případě dobrých výsledků.

Kapitola 7

Implementace

Tato kapitola popisuje celou implementaci navrhnutého programu a použitých knihoven. Nejprve jsou uvedeny tři základní knihovny, které byly použity při samotné implementaci programu. Jedná se o velice stručné nastínění schopností knihoven, bez většího popisu jejich funkcí a možností použití. Následuje postup kalibrace kamery pomocí externího programu a hlavní obsah pak tvoří podkapitoly věnující se implementaci jednotlivých částí, tak jak byly uvedeny v návrhu řešení rozšířeném o další prvky. Základní představu implementace a chod programu shrnuje diagram obr. 7.1.

Program byl napsán v jazyce C/C++ s využitím knihoven uvedených v následujících kapitolách a vyvíjen na platformě Microsoft Windows v prostředí Microsoft Visual Studio 2008 a důraz byl kladen i na možnost přenositelnosti na ostatní platformy.

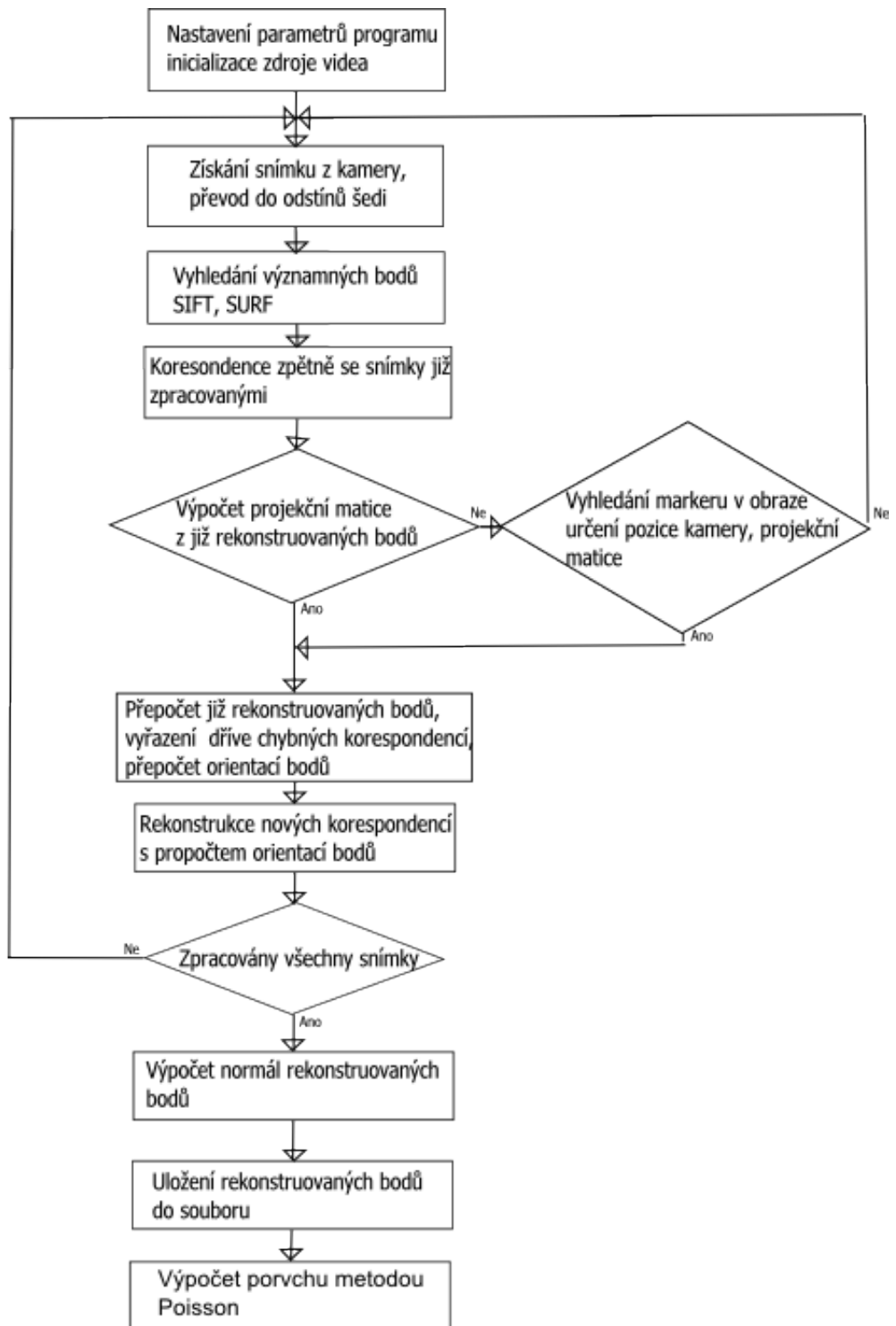
7.1 ARToolKit

ARToolKit je softwarová knihovna pro vytváření aplikací rozšířené reality. Tyto aplikace dodávají do obrazu reálného světa virtuální objekty a to způsobem, že překrývají původní obraz snímáný kamerou. Ukázka je vidět na obrázku 7.2, kde je možno vidět trojrozměrný virtuální objekt dodaný do scény na reálný podklad. Uživatel pozoruje scénu pomocí speciálních brýlí (náhlavního displeje), které má nasazeny. Když uživatel otáčí s podkladem, na kterém je virtuální objekt umístěn, otáčí se stejně i ten a uživatel má pak pocit, že je objekt s podkladem pevně svázán.

Jedním z největších úskalí při vývoji aplikací rozšířené reality je problém sledování orientace pohledu uživatele a jeho pohybu v prostoru. Pro správnou funkci vykreslování virtuálních objektů do reálné scény je potřeba znát přesnou polohu uživatele a směr, kam se dívá v reálném světě.

ARToolKit využívá algoritmy počítačového vidění pro řešení již zmíněného problému lokalizace uživatele. Video knihovny ARToolKit vypočítávají přesnou pozici kamery a orientaci relativně k markeru (podkladu) v reálném čase. To umožňuje jednoduchý vývoj širokého pole aplikací rozšířené reality. Některé z důležitých výhod knihovny jsou [1]:

- Sledování pozice jedné kamery a její orientace v prostoru.
- Kód pro vyhledávání, který využívá jednoduchých černých čtverců.
- Možnost využít jakéhokoli čtvercového markeru.
- Jednoduchou kalibraci kamery.



Obrázek 7.1: Implementační diagram.



Obrázek 7.2: Ukázka možnosti použití knihovny ARToolkit [1].

Knihovna ARToolkit je distribuována s kompletními zdrojovými kódy a distribuce jsou pro tyto platformy SGI IRIX, Linux, MacOS a Windows. Tím je zajištěna i podmínka přenositelného kódu vytvořené výsledné aplikace. Využitím ARToolkit se zabývá například i diplomová práce „Parametry systému pro rozšířenou virtuální realitu“ [16].

7.2 OpenCV

OpenCV je open source knihovna počítačového vidění napsaná v C/C++ a distribuovaná pro systémy Linux, Windows a Mac OS X[6]. OpenCV je hojně používaná knihovna při vytváření programů zaměřených na počítačové vidění, zpracování obrazu, kalibrace kamer, rekonstrukce objektů, vyhledávání objektů v obraze atd.

OpenCV lze rozdělit do tří základních částí [2]:

- Cxcore

- Cv
- Highgui

Cxcore zajišťuje základní funkcionalitu knihovny. Obsahuje základní datové struktury, dynamické datové struktury, operace nad těmito strukturami. Základní funkce vykreslování do rastru.

Cv obsahuje zpracování obrazu a funkce počítačového vidění. Tedy stěžejní funkce OpenCV. Jedná se o filtrování obrazu, geometrické transformace, histogramy, vyhledávání význačných bodů, analýzu pohybu, sledování objektu a vyhledávání objektu a v neposlední řadě funkce pro kalibraci kamery a 3D rekonstrukci.

Poslední část highgui pak slouží k vytváření základního uživatelského rozhraní a hlavně ke čtení a zápisu snímků a videa.

Jednoduchá ukázka načtení snímku ze souboru, základní konverze do odstínu šedi a zobrazení na monitoru pak vypadá takto:

```
//Vytvoření okna, do kterého se bude zobrazovat.
cvNamedWindow("Okno");

//Otevření a načtení obrázku ze souboru
IplImage *img = cvLoadImage("obrazek.jpg");

//Vytvoření datové struktury, kde bude uložen převedený snímek.
IplImage *gray=cvCreateImage(cvSize(img->width,img->height),IPL_DEPTH_8U,1);

//Převedení původního snímku do odstínu šedi.
cvConvertImage(img,gray);

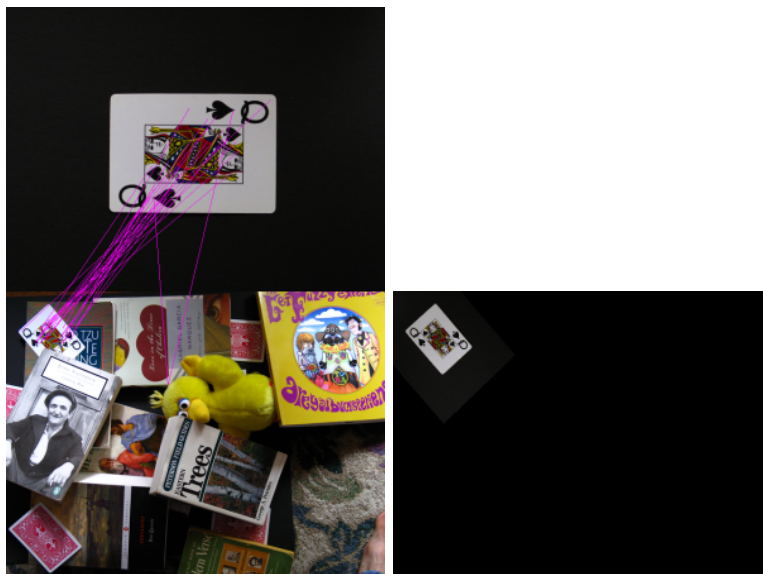
//Zobrazení snímku v okně.
cvShowImage("Okno",img);
```

7.3 SIFT

Poslední knihovnou, jejíž implementace je použita v navrhnutém systému, je knihovna SIFT Feature Detector autor je Rob Hess[4]. Jedná se o balík zdrojových souborů a přeložených binárních souborů, které slouží pro vyhledání významných bodů metodou SIFT. Binární soubory jsou ukázkové aplikace vytvoření korespondencí mezi zadanými dvěma snímky, například objektu jako samostatného snímku, a pak toho samého ve scéně, kdy je provedena lokalizace a označení objektu právě ve scéně.

Z této knihovny je v navrhnutém systému využívána pouze funkce na vyhledání významných bodů ve scéně, která má název `sift_features`. Funkce je předán 8b snímek převedený do odstínu šedi, funkce v něm vyhledá významné body a v parametru vrací strukturu typu `feature`.

Jediná závislost knihoven je na OpenCV kap.7.2, tudíž se opět jedná o přenositelný kód na jiné platformy.



Obrázek 7.3: Vyhledání objektu ve scéně pomocí knihovny SIFT[4].

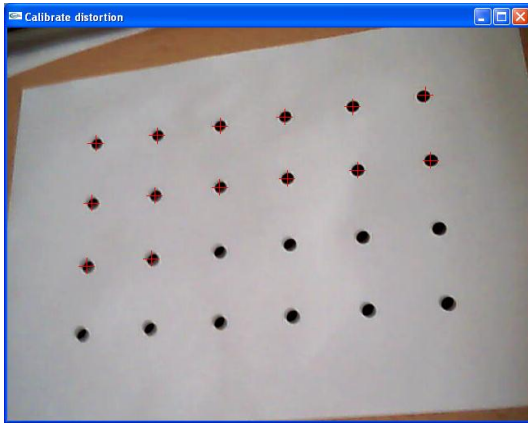
7.4 Kalibrace kamery

Navrhnutý systém počítá s již dopředu zkalibrovanou kamerou, kdy jsou vnitřní parametry kamery a koeficienty zkreslení předány jako jeden z parametrů výslednému programu pro rekonstrukci scény. Kalibrace není přímo implementována, ale používá se program z balíku ARToolkit.

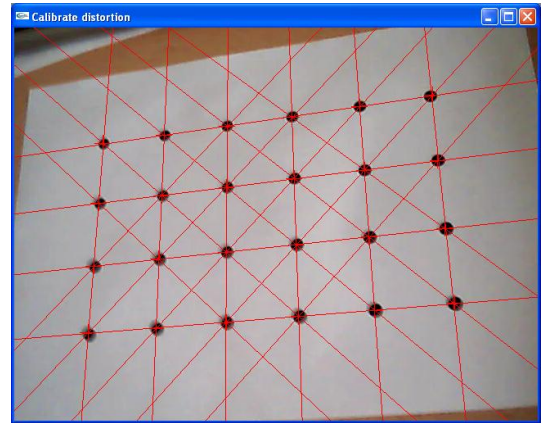
Je možno využít dvou postupů pro získání vnitřních parametrů kamery. 1. využít dvou programů `calib_dist` a `calib_cparam`. Programy jsou spuštěny postupně za sebou, protože druhý potřebuje na vstupu hodnoty získané z prvního.

`Calib_dist` používá pro kalibraci vzor skládající se ze 4x6 teček, jak je vidět na obrázku 7.4. Po spuštění programu je uživatel vyzván, aby zadal vzdálenost mezi tečkami na kalibračním vzoru. Poté se objeví video z připojené kamery a uživatel musí provádět označování teček na kalibračním vzoru, kdy kalibrační vzor může snímat pod libovolným úhlem, ale tak, aby byl vždy celý vzor viditelný. Vždy když je celý vzor viditelný zachytí uživatel pomocí myši snímek a v něm postupně označuje černé tečky pomocí myši obr. 7.4, přičemž záleží na pořadí. První tečka je v kalibračním vzoru v levém horním rohu, poslední pak v pravém dolním a postupuje se po řádcích. Proces zachytávání a označování vzoru by měl proběhnout 5-10krát pod různým úhlem a vzdáleností. Jakmile je označen dostatečný počet snímků a uživatel se rozhodne pro ukončení kalibrace jsou ještě jednou zobrazeny postupně všechny snímky i s propojenými úsečkami mezi body obr.7.5 a je nutné každý zobrazený snímek potvrdit pro jeho správnost. Výstupem programu je pak střed promítání a zkreslení čoček kamery. Výstup je zapsán na příkazový řádek a je nutný jako vstup následujícího programu.

`Calib_cparam` program určí ohniskovou vzdálenost kamery a další parametry. Po spuštění požaduje zadat výsledky kalibrace z předchozího programu. Navíc se mu zadá vzdálenost jednotlivých čtverců kalibračního vzoru, který se skládá ze sedmi vertikálních a devíti horizontálních úseček. Dalším ze zadávaných parametrů je počet měření, který bude proveden a vzdálenost posunu kamery každého kroku měření od kalibračního vzoru. Opět jako

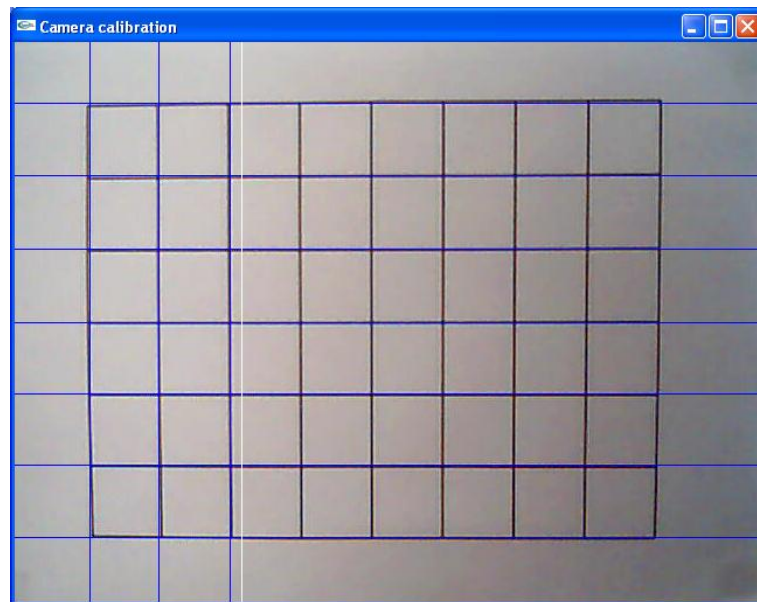


Obrázek 7.4: Označování bodů při kalibraci.



Obrázek 7.5: Potvrzení průniků úseček.

u prvního programu se objeví video. Zde je však nutné nastavit kameru přímo před snímání kalibrační vzor tak, aby mířila kolmo a nejlépe na střed snímku. Poté uživatel pomocí myši snímek uloží a začne na něm označovat horizontální a vertikální linie kalibračního vzoru obr. 7.6. Horizontální úsečky se označují shora dolů a vertikální zleva doprava. Jakmile je označování dokončeno, uživatel musí posunout kameru nebo vzor o přesně určenou vzdálenost, podle toho co zadal na začátku programu a celý proces opakovat. Celý proces je opakován pětkrát a po dokončení je uživatel vyzván, aby zadal jméno souboru do kterého se mají parametry uložit. Tento soubor pak bude vstupem výsledného implementovaného systému rekonstrukce scény.



Obrázek 7.6: Označování linií na kalibračním vzoru.

7.5 Nastavení parametrů programu, inicializace vstupu videa

Jelikož video z každé kamery může mít různou snímkovací frekvenci, velikost snímků atd., je nutné mít možnost nastavovat co nejvíce parametrů programu pro optimální rekonstrukci. Program nezpracovává všechny snímky videa a jsou vybírány jen po určitém čase. Kdyby byl zpracováván každý snímek, prodlužoval by se čas běhu programu, dále by měly vyhledané korespondující významné body ve dvou snímcích mezi sebou malou vzdálenost a docházelo by k chybě při rekonstrukci. Parametr je v programu pod názvem `skippedFrames`.

Dále je nutné nastavovat kolik snímků zpětně bude prohledáváno na korespondence. Tento parametr se opět odvíjí od snímkovací frekvence a již dříve nastaveného počtu vynechávaných snímků, kdy čím více snímků se vynechává, tím více by se jich pak mělo zpětně procházet pro vyhledávání korespondencí. Parametr `backFrames`.

Posledním parametr týkající se vynechávání a procházení již zpracovaných snímků na korespondence významných bodů je `farFrames`, který udává, po kolika snímcích zpětně se má otestovat další snímek na korespondence významných bodů. Tento postup je vhodný zejména v případech, kdy kamera byla chvíli mimo již známý prostor a vrátila se zpět. Jsou tak znovu vyhledány již známé body z předchozích rekonstrukcí.

Při nastavení `skippedFrames 10 backFrames 5 farFrames 5` je zpracováván každý desátý snímek příchozí z kamery, na korespondence je prohledáno posledních 5 zpracovaných snímků a pak každý pátý.

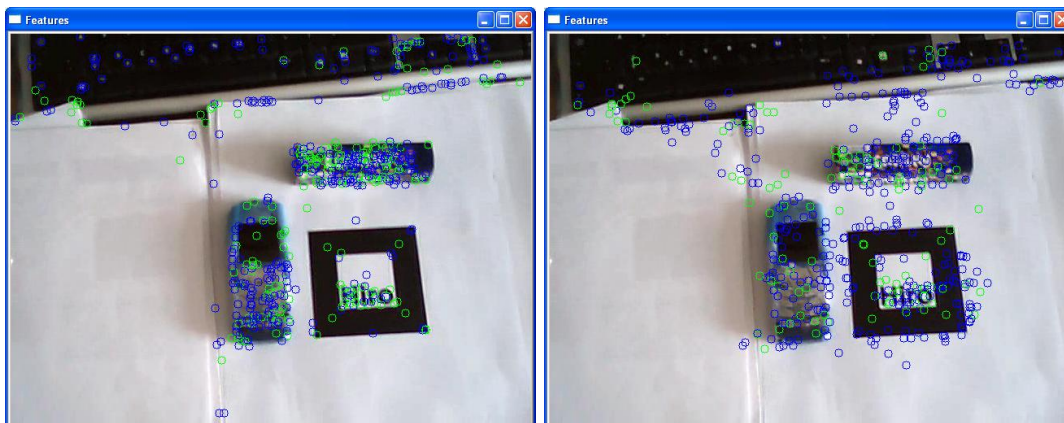
Inicializace videa je nutná pro určení vstupního proudu do programu, jestli se bude jednat o vstup přímo z kamery a nebo ze souboru. Pro přijímání videa je využito funkcí OpenCV. Načítání ze souboru je následně provedeno funkcí `cvCaptureFromFile()` a přímo z kamery pak `cvCaptureFromCAM(0)`, kdy je použita první přítomná kamera instalovaná a připojená v systému.

Pro obě dvě popsané inicializace je jednotné získávání snímků funkcí `cvQueryFrame()`. A dále se dá pracovat jednotně s načteným snímkem nezávisle na vstupním zdroji. Podmínkou pro funkčnost je dále znalost rozměrů vstupního snímku. Ty se určují ze zadaných parametrů při spuštění programu. Rozměry by šlo získávat až při načítání snímků, ale knihovna ARToolKit je potřebuje pro inicializaci vnitřních parametrů kamery ještě před samotným zpracováním prvního snímku.

7.6 Vyhledání významných bodů SIFT, SURF

Vyhledávání bodů je rozděleno na dvě možnosti, první je využití SIFT a druhá SURF. Metoda SIFT je implementována v knihovně popsané v kap. 7.3. SIFTy jsou vyhledány v obraze převedeném do odstínu šedi funkcí `sift_features()`, která v parametru vrací vyhledané významné body ve strukturách typu `feature` obsahující vždy základní vlastnosti vyhledaného bodu (pozici bodu ve snímku, deskriptor). Možností nastavení parametrů vyhledání bodů metodou SIFT je 7, kde každý ovlivňuje způsob a prostor ve kterém jsou body extrahovány. Parametry zůstaly nastaveny na původní hodnoty, které dávají dobré výsledky a jsou přednastaveny již v základu.

SIFTy jsou extrahovány podle postupu popsaném Lowem v práci [15], kdy dochází k sestavení scale-space a vyhledání lokálních extrémů následované přesnější lokalizací klíčových bodů, přiřazení orientací a tvorbě deskriptorů o rozsahu 128 hodnot. Pro každý takto nalezený bod je vytvořena struktura popisující daný bod.



Obrázek 7.7: Vyhledání významných bodů. Vlevo metodou SIFT, vpravo metodou SURF

Další implementovanou metodou pro extrakci významných bodů je metoda SURF napsaná podle [5], kterou obsahuje nová verze knihovny OpenCV 2.0. Funkce má názvem `cvExtractSURF()` vstupní obraz opět požaduje v stupních šedi a základními parametry, které ovlivňují extrakci bodů a je možné je měnit, jsou parametry vytvářející scale-space obrazu.

Výstupem funkce jsou dvě sekvence typu `CvSeq`, jedna obsahující nalezené významné body a druhá obsahující jejich deskriptory. Jelikož by byla implementace dvou struktur pro práci s body složitá, byla vytvořena funkce `surf_features`, která výstupní struktury extrakce SURFů převádí do struktury `features` stejně jako předešlé SIFTy. Jsou přehrány pozice bodů a celé deskriptory. Jelikož i deskriptory u metody SURF jsou voleny na délku 128 hodnot program dále nerozlišuje, se kterým typem bodů pracuje.

Typ metody extrakce bodů se volí v parametru programu při jeho spuštění a v průběhu provádění programu ho již nelze měnit. Obě dvě metody je možno vidět na obrázku 7.7, kde metoda SIFT je vlevo a metoda SURF vpravo.

7.7 Vyhledání korespondencí

Vyhledávání korespondencí probíhá mezi aktuálně zpracovávaným snímkem získaným z kamery nebo souboru, tedy významnými body v něm nalezenými, se zvolenými snímky, významnými body extrahovanými z předešlých snímků, podle parametrů popsaných v kap. 7.5. Samotná implementace funkce `match()` vyhledává korespondence vždy mezi dvěma snímky, jedním je vždy aktuální a druhým některý z již zmíněných. Funkci jsou předány struktury významných bodů obou snímků, do kterých jsou ve výsledku označeny zpětné korespondence pomocí ukazatelů.

Funkce porovnává každý významný bod aktuálního snímku se všemi významnými body druhého zadaného snímku. Pro každou dvojici bodů je vypočtena Euklidovská vzdálenost jejich deskriptorů. Dvě nejmenší vzdálenosti jsou vybrány a porovnány, pokud je jejich poměr menší, než zadaná konstanta, jedná se s největší pravděpodobností o korespondující body. Takto jsou sestaveny dvě pole představující možné korespondence mezi dvěma snímky. Tyto dvě pole jsou dále zpracovány pro vyřazení chybných korespondencí pomocí epipolární geometrie.

Vyřazení chybných korespondencí se provádí pomocí funkce implementované v OpenCV

`cvFindFundamentalMat()`. Funkce pracuje se zmíněnými dvěma poli možných korespondencí a s využitím metody RANSAC vypočítá nejvíce pravděpodobnou fundamentální matici, která pro správné korespondence splňuje rovnici $p_1^T F p_2 = 0$. Jelikož souřadnice bodů nejsou nalezeny přesně a ani fundamentální matice není zcela přesná, rovnice se většinou nerovná 0. Korespondující bod se hledá v okolí epipolární přímky kap. 5.1. Spočítá se vzdálenost korespondujícího bodu od přímky a pokud je vzdálenost menší, než zvolená mez, je bod považován za korespondující.

Funkce `cvFindFundamentalMat()` má právě i parametry pro zadání hodnoty meze, kdy je bod považován ještě za korespondující a nacházející se na epipolární přímce. V parametru `status` se pak vrací v poli hodnoty 0 nebo 1 podle toho, jestli body splnily kritérium a jedná se o korespondence či ne.

Po vyhledání korespondencí jsou upraveny struktury nově nalezených bodů tak, aby obsahovaly hodnoty potřebné pro jejich další zpracování v následující funkcích. Jsou do nich uloženy ukazatele na struktury právě nalezených korespondencí a naopak ze struktur nalezených korespondencí jsou získána data, zda se nejedná o již rekonstruované body v prostoru.

7.8 Určení pozice kamery pomocí markeru

Hlavní částí programu je správné nalezení a určení pozice kamery v prostoru. Se znalostí pozice kamery a vnitřních parametrů se již dají rekonstruovat body v prostoru z nalezených korespondencí významných bodů ve snímcích.

Pro určování pozice kamery bez znalosti scény se jevílo vhodné využít funkcí balíku `ARToolKit`. Tato sada knihoven se používá pro vytváření jednoduchých aplikací rozšířené reality. Má dobré vlastnosti při určování pozice kamery ve scéně pomocí markeru ve scéně umístěného.

`ARToolKit` vyhledává marker za pomoci prahování obrazu a vyhledávání vzoru v obraze. Marker je nalezen funkcí `arDetectMarker()`. Té se předá přednastavená hodnota prahu a hlavně obraz jako ukazatel na pole typu `ARUInt8`, které obsahuje všechny tři složky barvy plus alfa kanál. Bylo tedy nutné převádět pro každý snímek z přijímaného typu struktury v `OpenCV` do typu v `ARToolKit`. Dále pro detekci markeru musejí být známy vnitřní parametry kamery a uloženy v globální proměnné `cparam` typu `ARparam`. Jednotlivé kroky nalezení markeru jsou na obrázku 7.8 získaném z aplikace `simpleLite` z balíku `ARToolKit`.

Vlastní pozice kamery je získávána funkcí `arGetTransMat()`, která v parametru vrací matici 3×4 složenou z rotace a translace $[R|T]$. Jedná se o pozici a orientaci kamery vztahované vůči sledovanému markeru. Pozice kamery v souřadném systému markeru se pak získá inverzí nalezené matice, kde pozice je poslední sloupec této matice.

Vypočítaná pozice kamery je nakonec uložena do struktury `siftFeaturesArray`, aby bylo možné ji používat dále pro rekonstrukci bodů v prostoru, a také je nutné ji znát při zpracovávání dalších snímků. Výsledná projekční matice je pak získána vynásobením matice vnitřních parametrů kamery a matice rotace a translace kamery $P = K[R|T]$.

7.9 Určení pozice kamery z již rekonstruovaných bodů

Jedním z cílů, kterých práce chtěla dosáhnout byl výpočet pozice kamery bez znalosti markeru z již rekonstruovaných významných bodů.



Obrázek 7.8: Vyhledání a označení pozice markeru ve snímku. Snímek je nejprve prahován a pak v něm nalezen marker.

Jak je v návrhu napsáno, je nutné nejprve označit body, které mají v prostoru již správně určenou pozici. Označování probíhá ve funkci `avgReprojectErr()`, která prochází všechny významné body ve struktuře 3D bodu a počítá chybu zpětné projekce. Pokud je chyba menší než zadaná hodnota, je bod označen jako vhodný pro výpočet pozice kamery.

Pokud je ve snímku nalezeno alespoň deset bodů vhodných pro výpočet pozice kamery, je zavolána funkce `calcExternParametr()`, která vypočítá celou projekční matici metodou nejmenších čtverců a matice je poté získána pozice kamery. Nalezených bodů by stačilo šest, ale pro přesnější projekční matici je vhodných více bodů.

7.10 Výpočet nových bodů z korespondencí

Při správně vypočítané projekční matici a znalosti korespondencí bodů aktuálně zpracovávaného snímku s body z předcházejících snímků jsou vypočítány body v prostoru.

Implementace je ve funkci `reconstruct`, která ze zadaných dvou korespondujících bodů ze dvou snímků a projekčních matic kamer obou snímků sestaví matici A . Tato matice je vyřešena rozkladem SVD pomocí funkce v OpenCV `cvSVD`. Výsledkem je čtyřprvkový vektor, který vydělený posledním jeho členem, dává homogenní souřadnice rekonstruovaného bodu v prostoru.

Jelikož výpočet není zcela přesný a mohlo dojít chybě při určování korespondencí, je nutné ověřit rekonstruovaný bod zpětnou projekcí. Zpětná chyba projekce pro jeden bod je implementována ve funkci `reprojectErrX()`.

Pokud je chyba menší než zadaná hodnota je bod uložen do struktury `point3d` i s odkazy na použité korespondující body a naopak těm je uložen odkaz do struktury rekonstruovaného bodu. Také je uložena orientace bodu v prostoru jako rozdíl pozice kamery a vypočítaného bodu normalizovaná funkcí `normalize()`.

7.11 Přepočítání již rekonstruovaných bodů

U již existujících bodů je nutné provádět korekci pozice prostoru, protože ne vždy je výpočet z předchozí kapitoly správný a někdy nepomůže ani zpětná projekce k vyřazení chybně rekonstruovaných bodů.

Přepočítání je prováděno ve funkci `upgrade3d()`. Každý významný bod, který byl v obraze nalezen a má korespondenci s již rekonstruovaným bodem, je přidán do struktury korespondujícího bodu a jsou provedeny přepočty bodu v prostoru funkcí `recalculatePoint2()`. Je-li vrácenou hodnotou některý z indexů korespondujících bodů, je tento bod vyřazen, jinak dojde k vyřazení celého 3D bodu a následně ke zmazení všech odkazů na tento bod v prostoru a také všech odkazů v korespondujících bodech.

7.12 Výpočet normál rekonstruovaných bodů

Jednou z částí návrhu je použití metody Poisson pro rekonstrukci povrchu scény z rekonstruovaných bodů. Tato metoda není v tomto programu implementována a využívá se externího programu nebo aplikace `MeshLab`, která tuto metodu také obsahuje.

Metoda však pro svůj správný běh potřebuje znát normály rekonstruovaných bodů. Výpočet probíhá na základě vyhledání bodů v okolí, sestavení roviny z vyhledaných bodů a výpočet normály.

Popsaný postup je implementován funkcí `countSaveNormal()`. Funkce vyhledá pro každý rekonstruovaný bod zadaný počet bodů v jeho okolí. Metodou nejmenších čtverců vypočte plochu těchto bodů a nakonec i normálu výsledné plochy jejíž směr upraví podle předběžných orientací uložených u 3D bodu.

Po výpočtu normál je možné uložit všechny rekonstruované body do souboru typu `.xyz`, kde jsou na jednom řádku za sebou vypsány souřadnice bodu následované normálou bodu.

Kapitola 8

Výsledky

Kapitola uvádí výsledky při testování programu nejprve na uměle vytvořené scéně z programu Blender a poté na scénách pořízených webkamerou. Jedná se o malé scény s jedním nebo více objekty.

Implementovaný program se skládá ze tří oken:

- Okno kamery.
- Okno vyhledaných významných bodů.
- 3D náhled na rekonstruované body.

Ovládá se pomocí klávesnice a myši. Parametry mu jsou předány z příkazové řádky při spuštění.

8.1 Test umělé scény

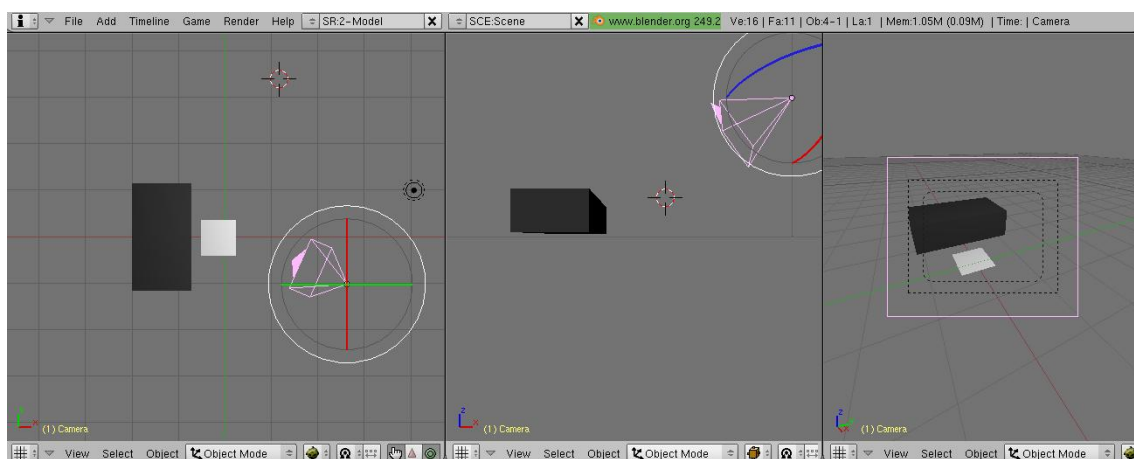
Tato podkapitola uvádí výsledky z testovací scény vytvořené v programu Blender. Ve scéně se nachází jeden objekt tvaru kvádrů s nanesou texturou pro lepší vyhledávání významných bodů. Druhým objektem je pak samotný marker viz. obr. 8.1. Kamera se pohybuje mezi dvěma vyznačenými pozicemi a stále míří do scény na pozici markeru a objektu.

Vyhledané významné body jednoho snímku jsou zobrazeny na obr. 8.2. Body jsou vyhledány hlavně na černých liniích textury. V případě neotexturovaného objektu, by bylo nalezeno minimum významných bodů.

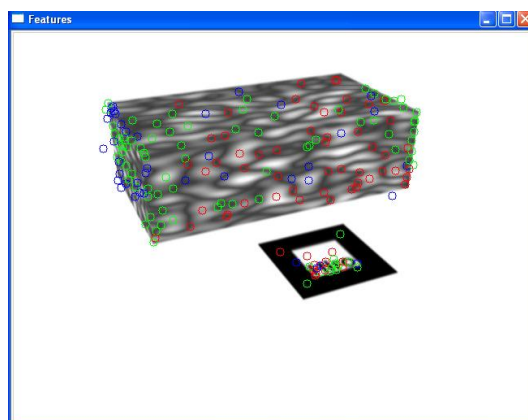
Obrázek 8.3 ukazuje dva pohledy na zrekonstruované body v okně 3D náhledu. Je vidět, že přední a horní stěny tvořené rekonstruovanými body jsou na sebe kolmé.

Poslední obrázek 8.4 této podkapitoly porovnává scénu s rekonstruovanými body. Podle porovnání není chyba rekonstrukce u výsledných bodů velká.

Jak je vidět z příslušných obrázků dopadla rekonstrukce uměle vytvořené scény poměrně dobře. Mezi rekonstruovanými body se nenachází velký počet chybně určených bodů a nebylo tak potřeba body ani filtrovat, či odstraňovat. Otestovány byly metody SIFT i SURF pro vyhledání bodů a dávaly velice podobné výsledky. Rekonstruovaných bodů není však mnoho a proto nebylo možné správně určit normály a provést rekonstrukci povrchu. Porovnání bodů s původní scénou je v příloze B.



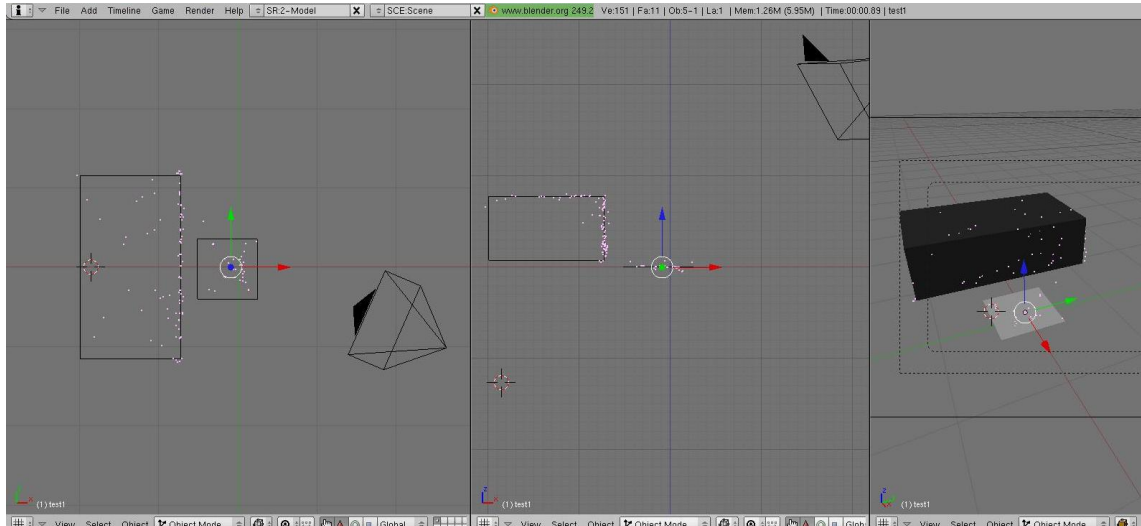
Obrázek 8.1: Program Blender s vytvářenou jednoduchou scénou pro rekonstrukci.



Obrázek 8.2: Ukázka vyhledaných bodů metodou SIFT.



Obrázek 8.3: Dva pohledy na rekonstruované body. Je vidět pravý úhel mezi dvěma stěnami.



Obrázek 8.4: Porovnání scény s rekonstruovanými body v Blenderu.



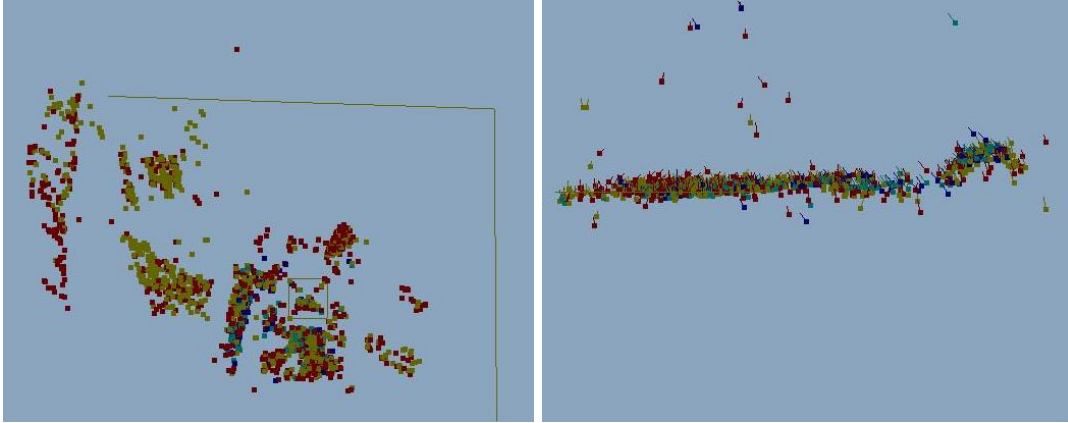
Obrázek 8.5: Snímek testu reálné scény ze souboru s vyznačenými body metodou SIFT.

8.2 Test reálné scény ze souboru

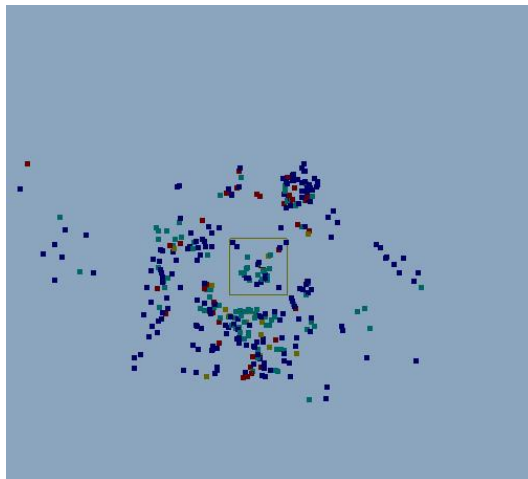
Ve scéně se vyskytuje několik objektů, jedná se o knihy a různé objekty, jak je vidět na obrázku 8.5, kde jsou vyznačeny i významné body. Videosekvence má dobu trvání 36 sekund a většinu času míří kamera na marker. Na konci videosekvence se pohybuje i mimo něj. Zpracováván byl každý 10tý snímek a korespondence vyhledávány v pěti posledních snímcích.

Náhled 3D rekonstruovaných bodů při vyhledávání metodou SIFT celé scény a vybraných objektů je možno vidět na obrázku 8.6. Rekonstruovaných bodů je mnoho a objevuje se i množství chybných rekonstrukcí. Je tady nutné část bodů filtrovat. Například filtrováním podle počtu významných bodů obsažených ve struktuře 3D bodu.

V každém snímku bylo nalezeno přibližně 2500 významných bodů metodou SIFT. Z toho důvodu trval výpočet korespondencí s předcházejícími snímky dlouho. Použitím metody SURF bylo nacházeno okolo 1200 bodů na snímek. Zde se ukázala hlavní přednost SURF



Obrázek 8.6: Vlevo: Náhled na rekonstruované body celé scény při využití metody SIFT. Vpravo: Zadní pohled na rekonstruované body válcovité krabičky nacházející se vlevo od markeru, společně s body na povrchu jedné z knížek.



Obrázek 8.7: Náhled na rekonstruované body celé scény v testu reálné scény ze souboru při využití metody SURF

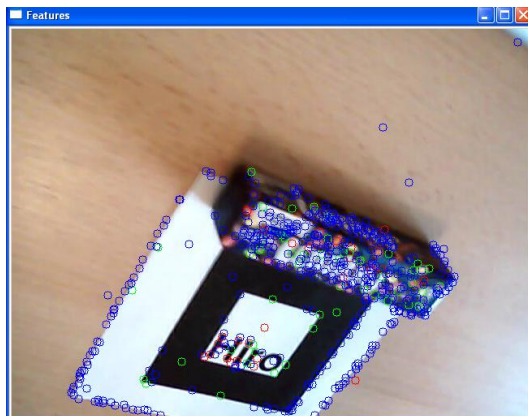
a to v rychlosti vyhledávání bodů. Pokud by bylo použito i deskriptoru o rozsahu 64 hodnot místo použitých 128, byl čas trvání výpočtu opět rychlejší. Jak je však vidět na obrázku 8.7 je rekonstruovaných bodů mnohem méně než metodou SIFT.

Výpočet normál a pokus o rekonstrukci povrchu se nepovedl. Vypočítané normály měly špatný směr a z rekonstrukce povrchu pak vyšel model, který neodpovídá rekonstruované scéně.

8.3 Test reálné scény z kamery

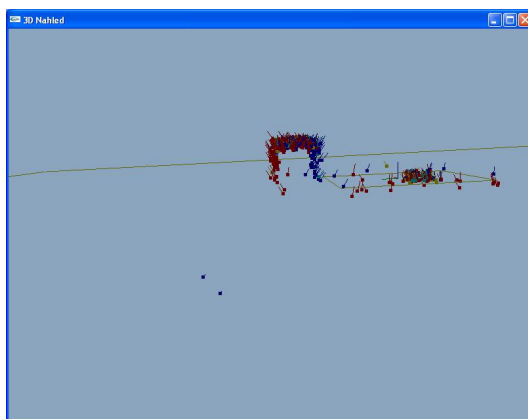
Další z testovaných scén je scéna rekonstruovaná v „reálném“ čase. Nový snímek je z kamery získán až po zpracování předchozího snímku. V průběhu zpracování nejsou žádné další snímky z kamery přijímány.

Jedná se o scénu s krabičkou umístěnou vedle markeru obr. 8.8. Kamera se nejprve pohybuje v přední části. Po získání dostatečného množství rekonstruovaných bodů se kamera posouvá a sleduje zadní stranu krabičky, ta je také rekonstruována. Tím je demonstrována rekonstrukce i bez vyskytujícího se markeru v zorném poli kamery.



Obrázek 8.8: Vyhledané významné body z pohledu na zadní stranu krabičky.

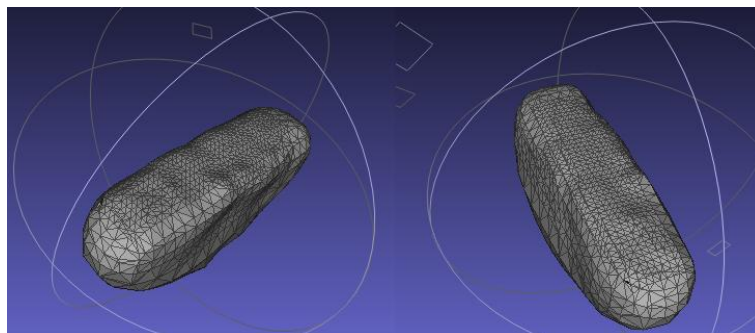
Obrázek 8.9 zobrazuje rekonstruovanou krabičku z bočního pohledu. Je vidět tvar krabičky s přední, zadní a horní stěnou.



Obrázek 8.9: Boční pohled na rekonstruovanou krabičku. Modré body jsou rekonstruované při znalosti markeru. Červené body jsou rekonstruované z vypočítané pozice kamery.

Díky dostatečnému počtu rekonstruovaných bodů i po odfiltrování základních rekonstrukcí, je možno vypočítat relativně dobře normály mračna bodů. Výsledek metody Poisson je pak vidět na obrázku 8.10.

Rekonstrukce při získávání snímků přímo z kamery dopadla dobře i pro zadní část krabičky, kde body byly počítány bez znalosti markeru. Výsledného modelu bylo docíleno po 350 snímcích, kdy byl zpracováván každý 10tý snímek a pro korespondence procházeno vždy posledních 5 snímků. Jako u jednoho z mála testů byl získán i relativně přesný povrch z mračna bodů. Pro tento test bylo využito opět vyhledávání významných bodů metodou SIFT. Vyhledané korespondence významných bodů získaných metodou SURF se nedařilo



Obrázek 8.10: Dva pohledy na zrekonstruovaný povrch krabičky metodou Poisson v programu MashLab.

správně zrekonstruovat a pokud k rekonstrukci došlo, byly při dalším korespondenci většinou vyřazeny. Dále i při správné rekonstrukci se nedařilo body označit jako vhodné pro výpočet pozice kamery.

8.4 Zhodnocení výsledků testů a implementace

Z předešlých kapitol lze zjistit, že implementace rekonstrukce dává poměrně dobré výsledky hlavně při vyhledávání významných bodů metodou SIFT. Daří se zrekonstruovat správně tvary těles vyskytujících se ve scéně. Rekonstruovaných bodů je nalezeno v každém zpracovávaném snímku dostatečné množství, korespondencí s předešlými snímky je také dost, ale samotný výpočet rekonstrukce pak většinu vyhledaných párů vyřadí. Zlepšení nastává až po výpočtu několika bodů, které jsou zároveň označeny jako vhodné pro získávání pozice kamery a kamera je tak počítána přímo.

Zároveň však dochází k výpočtu bodů, které nejsou na povrchu těles a přesto jsou považovány za správně zrekonstruované. Z toho důvodu se osvědčilo vyřazování již zrekonstruovaných bodů, které jsou nalezeny i v dalších snímcích, ale přesahují mez chyby zpětné projekce a to i po přepočtu.

Dále je u zrekonstruovaných bodů počítáno, z kolika významných bodů byly určeny. Lze tak jednoduše filtrovat, které body by měly mít přesnější pozici. V reálných scénách má velké množství bodů po prvotní inicializaci ze dvou korespondencí nepřesně určenou pozici.

Občas se také stane, že vypočítaná pozice kamery je nepřesná. Pokud pak dojde k rekonstrukci bodů a ty nejsou vyřazeny jako chybné, zůstávají až do konce výpočtu v mračnu bodů. Většinou jsou však takto vypočítané body po čase znovu nalezeny a vyřazeny jako nepřesné.

Hlavním problémem, který při rekonstrukci nastává, je rekonstrukce povrchu metodou Poisson. Výpočet normál bodů v mračnu pouze z nejbližších sousedů je naivní metodou a dochází k mnoha nepřesnostem. Z tohoto důvodu, tak ani metoda rekonstrukce povrchu nedává dobré výsledky. Jediné částečně uspokojivé výsledky byly v testu s kamerou, kdy bylo zrekonstruováno velké množství bodů na malém prostoru a propočteno normál, tak dopadl lépe. Dá se říci, že navržený systém s tímto výpočtem normál je pro metodu Poisson zcela nevhodný a pro uspokojivé výsledky by bylo nutné implementovat sofistikovanější algoritmus výpočtu normál bodů.

8.5 Možnost budoucího vývoje

Jak již bylo napsáno, metoda rekonstrukce povrchu nefunguje správně a pro svoji funkčnost potřebuje mít spočítané normály každého vstupního bodu. Chtělo by tedy upravit nebo zcela předělat výpočet normál. Jako možné řešení se jeví výpočet s využitím algoritmu RANSAC, jak je uvedeno v práci [21].

Dále po implementaci výpočtu normál, by bylo vhodné implementovat metodu Poisson přímo do programu. Knihovna v C++ již existuje a popsána je v [3]. Zařazení knihovny do programu by tedy nemělo představovat větší problém.

Dříve zmiňované občasné problémy s lokalizací kamery by šly řešit například pomocí zavedení dynamického výpočtu pozice kamery z předchozích pozic a rychlosti jejího posunu. Takto je řešena lokalizace kamery jak v MonoSLAM, tak v PTAM.

Nabízí se také možnost, řešit lokalizaci kamery přesně, jak je uvedeno v práci věnující se metodě PTAM [13]. Rekonstrukce velkého množství bodů by pak probíhala až po ukončení získávání videa, kdy by se celá videosekvence prošla znovu a nyní by byly body vyhledávány metodou SIFT a rekonstruovány s využitím již známých pozic kamery.

Kapitola 9

Závěr

V rámci práce byly prostudovány metody pro vyhledávání významných bodů SIFT a SURF, jež byly použity i v implementovaném systému. Z porovnání obou metod vyšla v implementaci lépe metoda SIFT, u které jsou korespondence stabilnější. Metoda SURF je však rychlejší a při vyhledávání významných bodů dává podobné výsledky.

Také byly nalezeny a popsány některé již existující metody 3D rekonstrukce a výpočtu pozic kamery. Pro lokalizaci kamery je nejpřesnější metoda PTAM, která by mohla být použita pro další pokračování práce. Metoda, bez jakéhokoliv markeru nebo předchozí znalosti scény, dokáže určovat přesnou pozici kamery v prostoru.

Byl navrhnut systém vycházející z již existujících řešení a kombinující několik různých knihoven do jednoho systému a také představena jednoduchá metoda získávání pozice kamery z již rekonstruovaných bodů.

Navrhnutá aplikace byla implementována a dává dobré výsledky při získávání pozice kamery a rekonstrukci bodů ve scéně. Aplikace zpracovává videosekvenci přímo z kamery nebo videosouboru a provádí rekonstrukci bodů v prostoru z vyhledaných významných bodů a jejich korespondencí v jednotlivých snímcích.

Bylo provedeno několik testů pro demonstraci funkčnosti aplikace. Jednalo se o test uměle vytvořené scény, kde výsledné mračno bodů bylo zobrazeno do modelu scény pro ukázkou přesnosti rekonstrukce. Další dva testy s reálnou scénou přinesly uspokojivé výsledky při lokalizaci kamery mimo známý marker.

Problémem je implementace naivního výpočtu normál z okolí rekonstruovaného bodu. Nepřesný výpočet normál má za následek nefunkčnost posledního bodu návrhu, tedy rekonstrukce povrchu externím programem.

Možným pokračováním je úprava výpočtu normál a implementace metody rekonstrukce povrchu z mračna bodů přímo do programu. Dále se nabízí možnost, upravit lokalizaci kamery na základě postupů v metodě PTAM. Samotnou rekonstrukci pak provádět, až po určení pozice kamery ve všech snímcích.

Literatura

- [1] ARToolKit. [online], [cit. 2010-05-10].
URL <<http://www.hitl.washington.edu/artoolkit/>>
- [2] OpenCV 2.0 C Reference. [online], [cit. 2010-05-05].
URL <<http://opencv.willowgarage.com/documentation/index.html>>
- [3] Poisson Surface Reconstruction (Version 2). [online], [cit. 2010-05-22].
URL <<http://www.cs.jhu.edu/~misha/Code/PoissonRecon/>>
- [4] SIFT Feature Detector. [online], [cit. 2010-05-10].
URL <<http://web.engr.oregonstate.edu/~hess/>>
- [5] BAY, H.; TUYTELAARS, T.; GOOL, L. V.: SURF: Speeded Up Robust Features. In *Computer Vision - ECCV 2006*, July 2006, s. 404–417, doi:10.1007/11744023.
- [6] BRADSKY, G.; KAEHLER, A.: *Learning OpenCV*. O'Reilly, 2008, ISBN 978-0-596-51613-0.
- [7] BYSTRICKÝ, V.: Určení pozice bodu v prostoru pomocí dvou kamer. Technická zpráva, Západočeská univerzita v Plzni, Fakulta aplikovaných věd, 2007.
- [8] BÍLEK, P.: *Významné body v obraze: detekce, korespondence a lokalizace ve 3D*. Bakalářská práce, FEL ČVUT, 2007.
- [9] DAVISON, A.; REID, I.; MOLTON, N.; aj.: MonoSLAM: Real-Time Single Camera SLAM. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, ročník 29, č. 6, June 2007: s. 1052–1067, ISSN 0162-8828, doi:10.1109/TPAMI.2007.1049.
- [10] HARTLEY, R.; ZISSERMAN, A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press, druhé vydání, 2003.
- [11] Intel: *OpenCV Reference Manuals*.
- [12] KAZHDAN, M.; BOLITHO, M.; HOPPE, H.: Poisson Surface Reconstruction. In *Eurographics Symposium on Geometry Processing*, 2006.
- [13] KELIN, G.; MURRAY, D.: Parallel Tracking and Mapping for Small AR Workspaces. In *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, Nara, Japan, November 2007.
- [14] KRŠEK, P.: Základy počítačové grafiky IZG, Studijní opora, dostupno pro zapsané v předmětu IZG na FIT VUT v Brně.

- [15] LOWE, D. G.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, ročník 60, 2 2004: s. 99–110.
- [16] MATĚNA, L.: *Parametry systému pro rozšířenou virtuální realitu*. Diplomová práce, FI MU, 2007.
- [17] PARKS, D. H.: Harris/Plessey Operator. [online], [cit. 2010-01-01].
URL <<http://www.cim.mcgill.ca/~dparks/CornerDetector/mainHarris.htm>>
- [18] RESSENDIZ, F. G.: 3D Scene Reconstruction from Multiple Views. [online], [cit. 2010-01-03].
URL <<http://www.thegoodwillcompany.co.uk/3Dfromfmv.htm>>
- [19] SHI, J.; TOMASI, C.: Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, Jun 1994, s. 593–600, doi:10.1109/CVPR.1994.323794.
- [20] VIALA, C.; Sanchez SALMERON, A.: Performance evaluation of linear camera calibration techniques. In *Automation Congress, 2004. Proceedings. World*, ročník 18, 28 2004, s. 49 – 54.
- [21] VÁVRA, R.: *Rekonstrukce ploškové reprezentace z mračna bodu*. Bakalářská práce, FEL ČVUT, 2008.
- [22] Wikipedia: Interest point detection. [online], [cit. 2010-01-03].
URL <http://en.wikipedia.org/wiki/Interest_point_detection>
- [23] Wikipedia: Singular value decomposition. [online], [cit. 2010-01-03].
URL <http://en.wikipedia.org/wiki/Singular_value_decomposition>
- [24] YANG, A. Y.: Image Formation and Camera Models. [online], [cit. 2008-04-29].
URL <<http://inst.eecs.berkeley.edu/~ee225b/sp07/lectures/lec12.pdf>>
- [25] ZHANG, F.; SHI, L.; XU, Z.; aj.: A 3D Reconstruction System of Indoor Scenes with Rotating Platform. In *Computer Science and Computational Technology, 2008. ISCST '08. International Symposium on*, ročník 2, dec. 2008, s. 554 –558, doi:10.1109/ISCST.2008.322.

Seznam použitých zkratek

2D Two-dimensional

3D Three-dimensional

MonoSLAM Single Camera Simultaneous Localisation and Mapping

PTAM Parallel Tracking and Mapping

RANSAC Random Sample Consensus

SIFT Scale-Invariant Feature Transform

SURF Speeded Up Robust Features

SVD Singular Value Decomposition

Dodatek A

Obsah CD

Struktura složek a popis jejich obsahu.

- ARToolKit - zkomprimovaný soubor knihovny ARToolKit.
- DP_latex - soubory diplomové práce v latexu.
- DP_pdf - soubor v pdf formátu.
- MeshLab - instalační soubor program MashLab, kterým se dají prohlížet výstupní soubory programu a mající implementovanou metodu Poisson.
- OpenCV - instalační soubor knihovny OpenCV 2.0.
- Plakat - plakát diplomové práce.
- SIFT - zkomprimovaný soubor knihovny SIFT.
- Testy - předpřipravené testy, přeložené binární soubory a data nutná pro spuštění testů.
- Zdrojove_kody - zdrojové kódy výsledného programu.

Dodatek B

Porovnání bodů s umělou scénou

Body z výsledku testovací scény (celkem 135) byly rozděleny do tří částí:

- Okolí markeru (34 bodů).
- Přední stěna objektu (70 bodů).
- Vrchní stěna objektu (28 bodů).
- Neporovnávané body (3 body).

Body byly označovány v programu MeshLab. Tři nevybrané body nejspíše ležely na boční stěně nebo nebyly při výběru označeny.

Všechny výpočty byly provedeny v prostředí Matlab. Marker má rozměry 8x8cm v programu uvedeny jako 80x80mm. Všechny následující výsledky jsou tedy v milimetrech. Byla vypočítána průměrná vzdálenost bodu od zvolené roviny a každé části vytvořen histogram vzdáleností bodu od roviny v hodnotách 0 až 10 s krokem 0,2. Každý sloupec histogramu tedy ukazuje, kolik bodů má danou vzdálenost od předpokládané roviny.

B.1 Okolí markeru

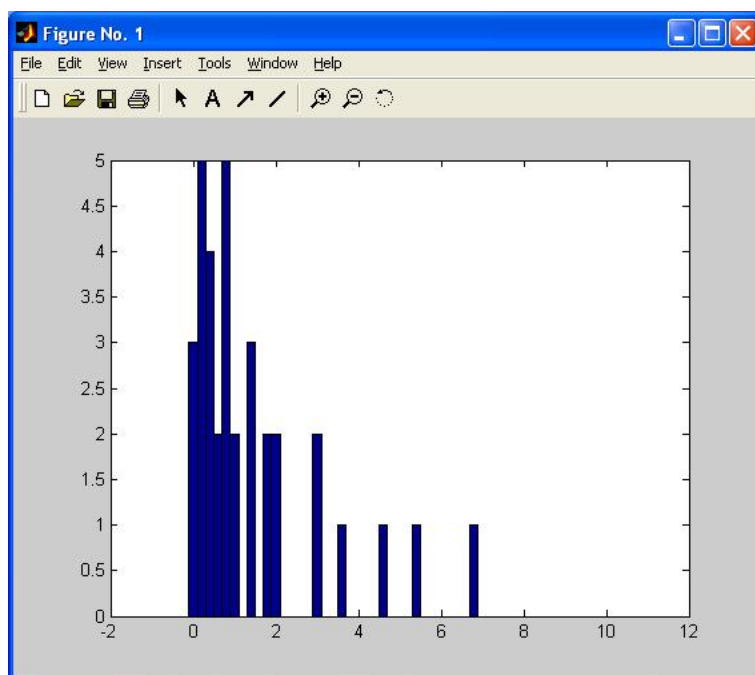
Rovina, ke které byly body porovnávány pro okolí markeru byla zvolena rovina Z , jelikož střed markeru leží na souřadnicích $[0, 0, 0]$. 34 bodů mělo průměrnou chybu 1,4162. Sestavený histogram je na obr. [B.1](#).

B.2 Přední stěna

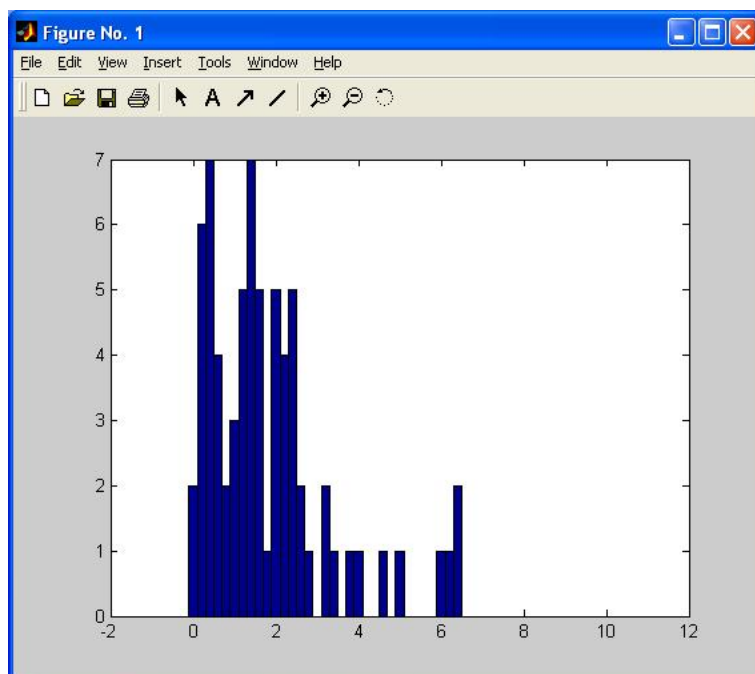
Rovina byla vypočítána ze tří bodů přední stěny objektu, získanými v programu Blender: $[-119.28, 62.16, 81.48]$, $[-119.280, 62.160, 7.875]$, $[123.375, 62.160, 81.480]$ Průměrná vzdálenost od plochy byla 1,829. Sestavený histogram je na obr. [B.2](#).

B.3 Vrchní stěna

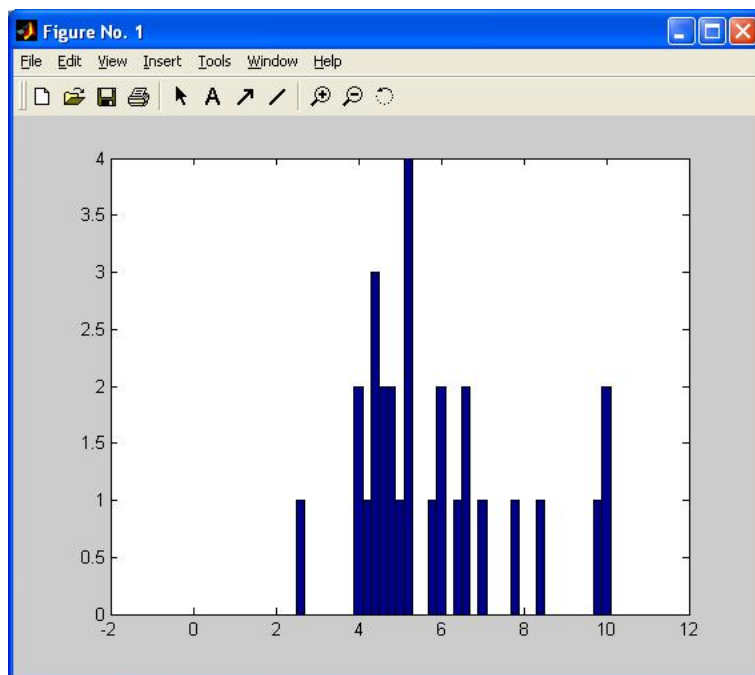
Rovina byla vypočítána ze tří bodů vrchní stěny objektu, získanými v programu Blender: $[-119.28, 62.16, 81.48]$, $[123.375, 195.405, 81.48]$, $[123.375, 62.160, 81.480]$ Průměrná vzdálenost od plochy byla 6,1084. Sestavený histogram je na obr. [B.3](#). Jelikož byla vzdálenost bodů, větší než u předchozích výpočtů, zkusil jsem posunout rovinu o 4 a průměrná chyba



Obrázek B.1: Histogram bodů okolí markeru od roviny markeru.



Obrázek B.2: Histogram bodů okolí přední stěny od roviny přední stěny.

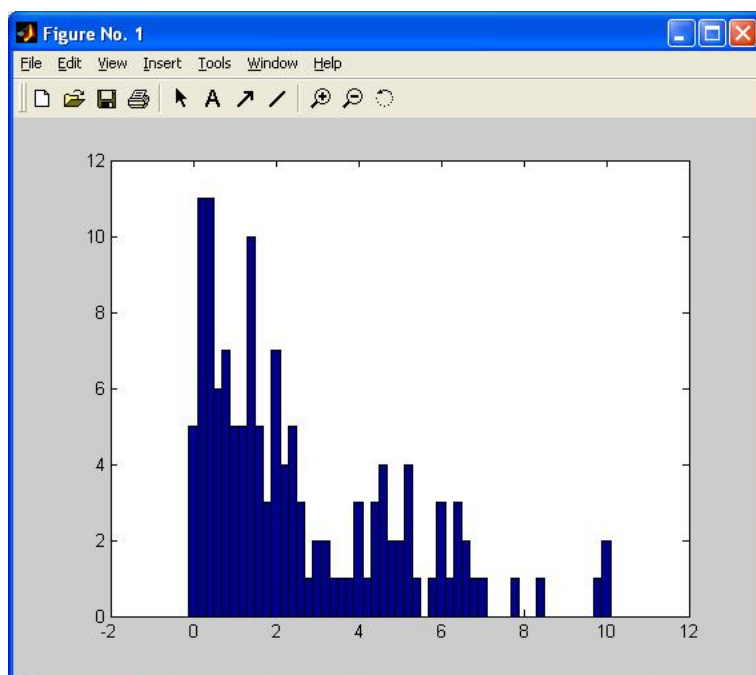


Obrázek B.3: Histogram bodů okolí vrchní stěny od roviny vrchní stěny.

se upravila na 2,2123. Dále je však počítáno s původní hodnotou. Chyba mohla vzniknout při zvětšování a posouvání objektu v programu Blender, kdy tato se nepromítla do chyby přední stěny.

B.4 Celá scéna

Všechny vzdálenosti bodů byly spojeny do jednoho celku a vypočítána průměrná vzdálenost, která byla 2,6261. A sestaven výsledný histogram [B.4](#).



Obrázek B.4: Celkový histogram vzdáleností celé scény.