VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

RUČNÍ SNÍMACÍ ZAŘÍZENÍ 3D OBJEKTŮ

HAND-HELD 3D SCANNER

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **MAREK KUKUČKA**
AUTHOR

VEDOUCÍ PRÁCE **MICHAL HRADIŠ, Ing., Ph.D.**
SUPERVISOR

BRNO 2020

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav počítačové grafiky a multimédií (UPGM) Akademický rok 2019/2020

Zadání bakalářské práce

Student: **Kukučka Marek**
Program: Informační technologie
Název: **Ruční snímací zařízení 3D objektů**
Hand-Held 3D Scanner
Kategorie: Zpracování obrazu

Zadání:

1. Prostudujte základy konvolučních sítí a 3D rekonstrukce objektů ze stereo kamery.
2. Vytvořte si přehled o současných metodách 3D rekonstrukce objektů se zaměřením na digitalizaci dokumentů.
3. Vyberte nebo navrhněte metodu řešící nějakou část rekonstrukce povrchu objektů ze stereo kamery.
4. Nasnímejte sadu dokumentů pro experimenty.
5. Implementujte navrženou metodu a proveďte experimenty nad datovou sadou.
6. Porovnejte dosažené výsledky a diskutujte možnosti budoucího vývoje.
7. Vytvořte stručné video prezentující vaši práci, její cíle a výsledky.

Literatura:

- Dle pokynů školitele.

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce: **Hradiš Michal, Ing., Ph.D.**
Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2019
Datum odevzdání: 31. července 2020
Datum schválení: 1. listopadu 2019

Abstrakt

Cílem této bakálářské práce je implementovat metodu 3D rekonstrukce z dvojic snímků, pořízených pomocí zařízení, které se skládá ze dvou kamer. Byla zvolena metoda stereovize a výsledkem je sparse rekonstrukce snímaného objektu. Prvně je provedena kalibrace kamer a úprava získaných snímků. V další části hledámé korespondující klíčové body ve dvojici snímků. Po získání korespondujících bodů je provedena jejich triangulace. Pro triangulaci z více jak dvou po sobě jdoucích snímků je využíván přepočet projekční matice. V rámci práce je proveden experiment, jehož cílem je otestovat, zdali zrekonstruovaný objekt odpovídá svými rozměry reálnému světu.

Abstract

The goal of this thesis is to implement a method of 3D reconstruction from pairs of images taken using a device that consists of two cameras. The method of stereo vision was chosen and the result is a sparse reconstruction of the scanned object. First, the cameras are calibrated and the acquired images are modified. In the next section, we look for corresponding key points in a pair of images. After obtaining the corresponding points, we an then perform their triangulation. For reconstruction from more then two consecutive images, we use recalculation of projection matrix. In this thesis an experiment is performed, with aim to test whether the reconstructed object corresponds in its dimensions to the real world.

Klíčová slova

zpracování obrazu, 3D rekonstrukce, kalibrace kamer, triangulace, projekční matice, stereovize, sparse rekonstrukce

Keywords

image processing, 3D reconstruction, camera calibration, triangulation, projection matrix, stereo vision, sparse reconstruction

Citace

KUKUČKA, Marek. *Ruční snímací zařízení 3D objektů.* Brno, 2020. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Michal Hradiš, Ing., Ph.D.

Ruční snímací zařízení 3D objektů

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Michala Hradiše. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Marek Kukučka
27. července 2020

Poděkování

Chtěl bych poděkovat panu Ing. Michalu Hradišovi, Ph.D. za odborné vedení mé práce, konzultace a rady, které mi pomohly tuto práci vytvořit.

Obsah

1 Úvod **2**

2 Metody 3D rekonstrukce **3**
2.1 Stereoskopie 4
2.2 Triangulace pomocí laseru 4

3 Práce s kamerami **6**
3.1 Popis použitého přístroje 6
3.1.1 Kamery 6
3.1.2 Objektivy 7
3.2 Využití stereovize 9
3.3 Pořizování snímků 9

4 Dataset **11**

5 Návrh řešení **12**
5.1 Kalibrace 12
5.2 Detekce klíčových bodů 15
5.3 Hledání korespondence bodů 20
5.4 Triangulace 22
5.5 Implementace 27

6 Realizace a experimenty **28**
6.1 Zpracování snímků 28
6.2 Nalezení klíčových bodů a jejich korespondencí v párech snímků 30
6.3 Triangulace 3D bodů 31

7 Diskuze **35**

8 Závěr **36**

Literatura **37**

A Obsah DVD **39**

Kapitola 1

Úvod

V rámci této práce se zabývám metodami 3D rekonstrukce a zpracováním obrazu. Cílem této práce je prozkoumat metody 3D rekonstrukce, zvolit metodu, ke které bude možné využít naše zařízení a vybranou metodu poté implementovat. Vybral jsem si tuto práci, jelikož mě zaujala možnost zabývat se 3D rekonstrukcí.

V kapitole 2 jsou popsány základní metody 3D rekonstrukce. Dále je zde popsáno co může bý výsledkem provedené rekonstrukce a co je jejich cílem.

Kapitola 3 je věnována použitému zařízení. Jsou zde podrobně popsány vlastnosti jednotlivých částí našeho zařízení. V této kapitole je také popsáno jak zařízení používáme pro vytvoření dvojic snímků.

Kapitola 4 popisuje mnou pořízený dataset. Je zde uvedeno rozdělení dat do dvou typů setů a popsána pravidla, podle kterých jsou jednotlivé snímky pojmenovávány.

Kapitola 5 se zabývá několika problémy. První část se zabývá zpracováním získaných snímků. Ze snímků je potřeba odstranit zkreslení a provést jejich rektifikaci. Druhá část se zabývá hledáním klíčových bodů a hledáním jejich vzájemných korespondencí ve dvojici snímků. Třetí část se věnuje triangulaci zpárovaných bodů, tato část nám poskytne 3D pozici bodů pro dvojici bodů.

Kapitola 6 popisuje realizaci projektu. Jsou zde k nalezeni výsledky jednotlivých kroků, které jsem musel podniknout v rámci procesu 3D rekonstrukce. Nachází se tu i experiment, který slouží k ověření, že jsou zachovány poměry mezi rekonstruovanými body a jejich reálnými protějšky.

Kapitola 7 obsahuje diskuzi o možném směru další práce a popis problémů, na které jsem narazil a možnosti jejich případného řešení.

Kapitola 2

Metody 3D rekonstrukce

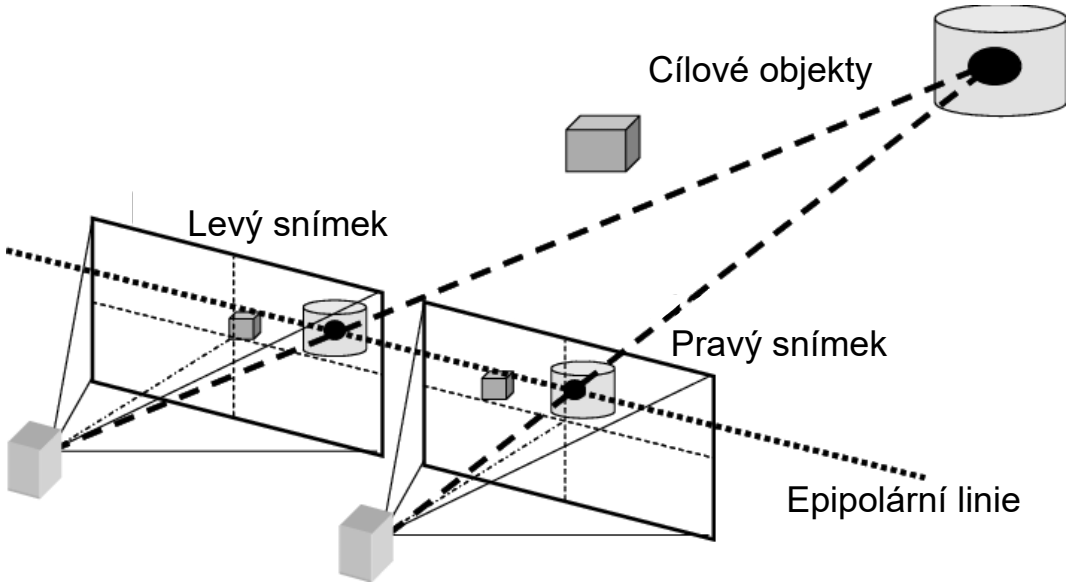
Proces 3D rekonstrukce v oblasti počítačového vidění a počítačové grafiky slouží k zachycení tvaru a vzhledu reálných objektů. Existují aktivní a pasivní metody. Aktivní metody aktivně interferují s rekonstruovaným objektem, buď mechanicky nebo radiometricky pomocí dálkoměrů, za účelem získání hloubkové mapy. Pasivní metody neinterferují s rekonstruovanými objekty, jenom používají senzory aby změřili záření, které objekt vyzařuje nebo odráží, z těchto dat se poté snaží odhadnout 3D strukturu objektu. Tato kapitola obsahuje ukázku několika metod, které se pro 3D rekonstrukci využívají. V této práci bude výsledkem 3D rekonstrukce mračno bodů.

Cílem 3D rekonstrukce poté může být tzv. sparse rekonstrukce nebo dense rekonstrukce. V případě sparse rekonstrukce dochází k odhadu struktury objektu z "řídké" sady znaků a v případě dense rekonstrukce je struktura odhadována z "hustých" oblastí pixelů.

Obrázek 2.1: Na levé části obrázku je výsledek sparse rekonstrukce a na pravé straně je výsledek dense rekonstrukce rekostrukce. [19]

2.1 Stereoskopie

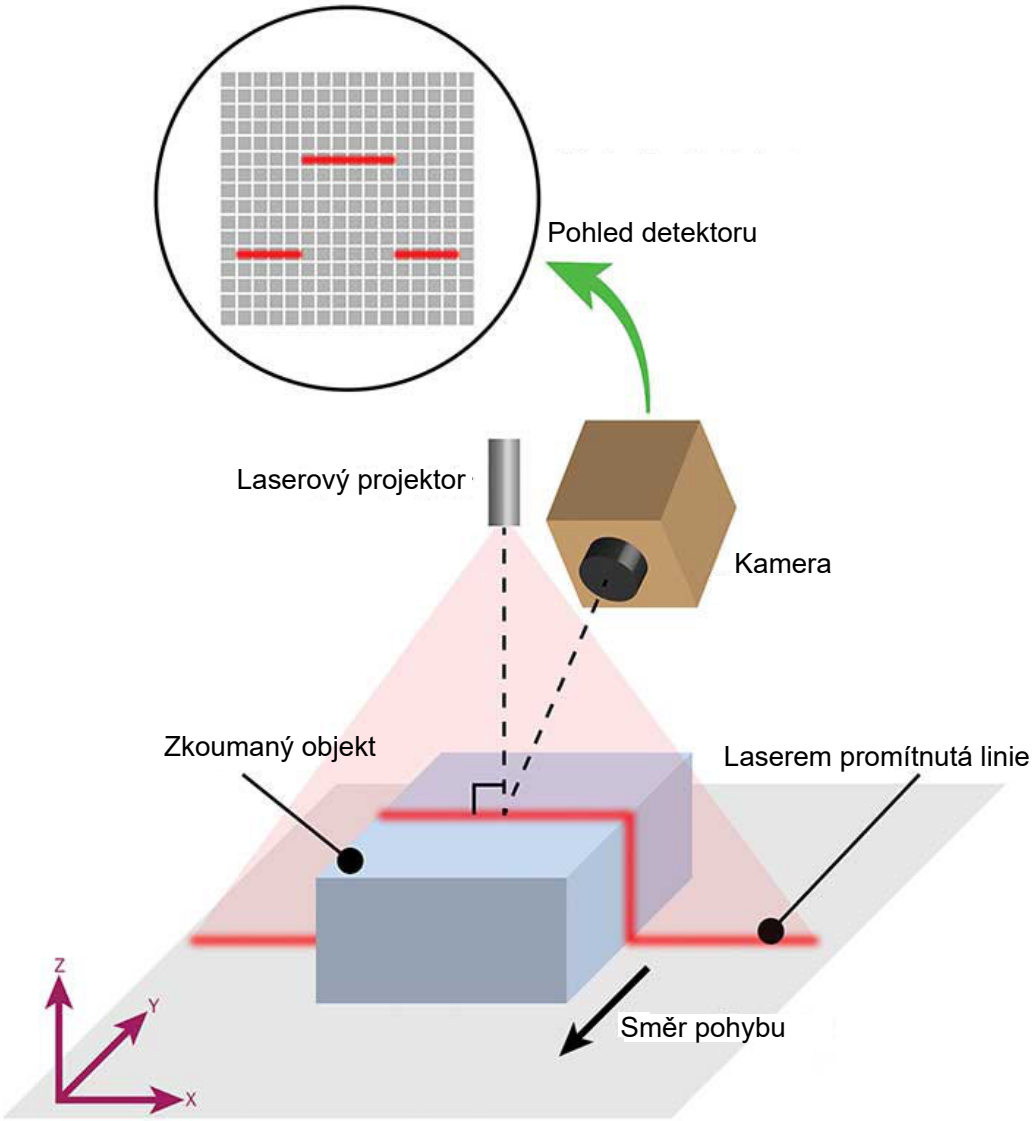
Jedná se o metodu, kterou využívám ve své práci. Tato metoda byla pojmenována sirem Charlesem Wheatstonem. Stereoskopie je též označována jako stereovize. Jedná se o schopnost odvodit informace o 3D struktuře objektu a jeho vzdálenost ve scéně ze dvou a více snímků, které byly pořízeny z různých míst. Stereovize obsahuje proces binokulární fůze a 3D rekonstrukce. V rámci počítačového vidění jako první hledáme korespondence mezi dvěma obrazy, tj. které části obou obrazů jsou projekcemi stejné scény. Po získání korespondence se 3D struktura stanoví pomocí epipolární geometrie. Pomocí odhadu disparity mezi dvěma snímky se poté vyhodnotí výška každého bodu.



Obrázek 2.2: Princip stereovize. [8]

2.2 Triangulace pomocí laseru

Tato metoda funguje na principu promítání paprsku světla na objekt. Pokud má být objekt zachycen kamerou, měl by být detekován pouze jasný bod. Když poté známe přesný úhel promítání paprsku vzhledem k základní linii je možné určit polohu bodu v prostoru. Tato metoda má několik variant. Tyto varianty jsou založeny na mnoha způsobech promítání a detekce světelných paprsků. V případě bodového zdroje musí být celá scéna skenována vertikálně i horizontálně, aby se získala hloubka. Další možností je promítání čáry, při této variantě lze současně získat hloubku všech bodů na dané přímce. Lze tedy použít více metod, každá z nich má své výhody a nevýhody. Mezi faktory, které by měli být brané v úvahu při vybírání metody patří přesnost, přítomnost slepých míst a rychlost skenování scény.



Obrázek 2.3: Promítnutá laserová linie se při pohledu z jiných perspektiv než z perspektivy projektoru jeví zkreslená. Toto zkreslení je kalibrováno a použito k odvození rozměrů zkoumaného objektu. [6]

Kapitola 3

Práce s kamerami

Tato kapitola se věnuje popisu použitého zařízení a principu stereovize, kterého v této práci využívám. Dále je zde popsána práce s kamerami, kterou jsem musel v rámci této práce provést.

3.1 Popis použitého přístroje

K pořízení snímků potřebných k další práci využívám zařízení, které bylo sestaveno panem Ing. Michalem Hradišem, Ph.D. Toto zařízení se skládá ze dvou kamer zasazených v rámu, které jsou osazeny objektivy.

Obrázek 3.1: Použité zařízení.

3.1.1 Kamery

Jedná se o kamery Basler daA1600-60um (CS-Mount) s rozhraním USB 3.0, jsou osazeny senzorem e2V EV76C570 CMOS, který je schopný poskytnout 60 snímků za sekund s rozlišením 2 MP. Podrobná specifikace kamery je k nalezení v tabulce 3.1.

Obrázek 3.2: Kamera Basler daA1600-60um.

Vlastnosti kamery

Prodejce senzoru	e2v
Senzor	EV76C570
Závěrka	Globální závěrka
Maximální kruh obrázku	1/1.8"
Typ senzoru	CMOS
Velikost senzoru	7,2 mm × 5,4 mm
Rozlišení (V x H)	1600 px × 1200 px
Rozlišení snímků	2 MP
Velikost pixelu (V x H)	4,5 m × 4,5 m
Snímkovací frekvence	60 snímků za sekundu
Barva	Monochrome
Rozhraní	USB 3.0
Bitová hloubka	8, 12 bitů
	hardwarový spouštěč
	volný běh
	softwarový spouštěč
Kontrola expozice	programovatelná pomocí API
Digitální vstup	1
Digitální výstup	2
Univerzální vstupní/výstupní pin	2
Požadavky na napájení	1,3 W
Váha	15g
Provozní teplota	0 - 50C

Tabulka 3.1: Vlastnosti kamery

3.1.2 Objektivy

Každá kamera má 5-megapixelový objektiv M1620-MPW2 od firmy computar s ultra nízkým zkreslením. Podrobnější specifikace objektivů je k nalezení v tabulce 3.2.

Vlastnosti objektivu

Obrázek 3.3: Objektiv M1620-MPW2 od firmy computar.

Ohnisková vzdálenost	16 mm
Maximální poměr clony	1:2.0
Provozní rozsah clony	F.2.0 - F16.0
Provozní rozsah zaostření	0,2 mm - Inf.
Ovládání clony	manuální
Ovládání zaostření	manuální
Provozní teplota	-10C - +50C
Zadní ohnisková vzdálenost	14,7 mm
Rozměry	29 mm × 33,53 mm
Váha	53g

Tabulka 3.2: Vlastnosti objektivu

3.2 Využití stereovize

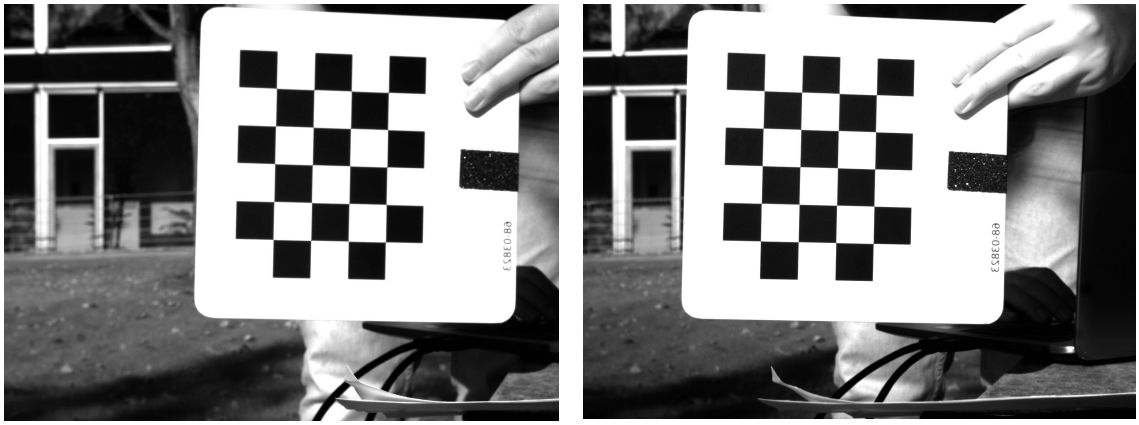
Máme zařízení se dvěma kamerami, které jsou rozděleny základní linií. Tyto dvě kamery poté pořídí snímek ve stejnou dobu a tím získáme dva snímky stejného objektu pořízené ze dvou rozdílných kamer. Tyto dva snímky poté analyzujeme, aby jsme identifikovali stejné pixely v obou dvou snímcích. Jedná se o problém, který je znám jako problém korespondence mezi dvěma kamerami. Například prvky jako jsou hrany mohou být jednoduše nalezeny v jednom snímku a tyto stejné hrany poté můžeme hledat v druhém snímku.

3.3 Pořizování snímků

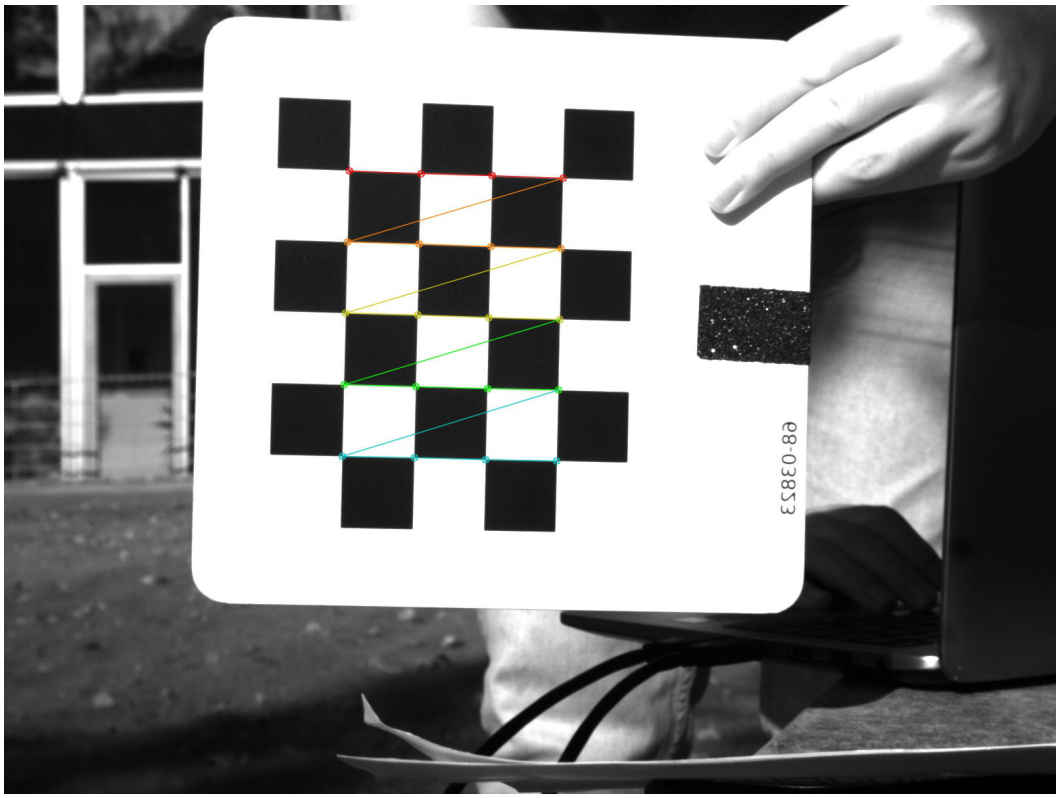
K pořízení snímků pomocí sestrojeného zařízení, jsem vytvořil program pomocí knihovny pypylon, která pochází od firmy Basler. Jedná se o knihovnu pro jazyk Python.

Jsou vytvořeny dvě funkce. První slouží k zachycení kalibračních snímků, které využívám při kalibraci kamery. Kamery aktivně snímají a zobrazují zachycené snímky z obou pohledů. Na těchto snímcích se poté hledá šachovnicový vzor, pokud je na snímku z obou kamer nalezen šachovnicový vzor, může uživatel snímky uložit zmáčknutím mezerníku. To zda jsou šachovnicové vzory detekovány poznáme tak, že dojde k označení rohů šachovnice 3.5. K nalezení a následnému označení šachovnice jsou použity funkce z knihovny OpenCV. Jednotlivé čtverce šachovnicového vzoru, které jsem použil mají rozměry 2,5cm $\times$ 2,5cm. Znát velikost jednotlivých čtverců je důležité pro výpočet vzdálenosti v reálném světě. Druhá funkce umožňuje pořídit dvojice sníků, bez jakéhokoliv omezení. Vuyžívám jí k pořízení snímků rekonstruovaného objektu. Většina snímků byla pořízena ze vzdálenosti 30-40 centimetrů, ale pořídil jsem si data i pro jiné vzdálenosti. Pořizování snímků je synchronizováno softwarovým triggerem. Tímto by mělo být zajištěno, že je dvojice snímků objektu zaznamenána ve stejném čase. Před pořízením snímku se dá nastavit expoziční čas.

Také jsem se pokusil implementovat funkci na pořizování videa. Kamery by měli umožňovat zaznamenávat obraz rychlostí až 60 snímků za sekundu. K této rychlosti se mi ale nepodařilo přiblížit. Má implementace se pohybuje pro záznam kolem 25 až 30 snímků za sekundu a fungovala jenom na operačním systému Windows. Při hledání řešení jsem na narazil na informace, že rychlost snímání může být ovlivněna výběrem expoziční doby a nastavením připojení kamer. Další možné řešení, pro zrychlení získávání snímků je vytvoření hardware triggeru. videa se dá parametrem nastavit délka videa a expoziční čas. Nakonec jsem se rozhodl videa nepoužít, jelikoz při pohybu kamery docházelo k rozmazávání snímků. Rozmázávání jsem se snažil odstranit snížením expozičního času, ale výsledné video poté bylo poměrně špatné kvality. Tento problém by se dal možná odstranit, pokud by se zlepšilo osvětlení snímaného objektu.



Obrázek 3.4: Ukázka páru snímků pořízených použitým zařízením během kalibrace.



Obrázek 3.5: Ukázka nalezení šachovnicového vzoru na snímku. Pokud je vzor nalezen na obou snímcích, může dojít k uložení kalibračního snímku.

Kapitola 4

Dataset

Tato kapitola se zabývá datovou sadou použitou při řešení této práce. Data byla použita při kontrole funkčnosti projektu a při experimentech.

Celý dataset jsem nasnímal sám pomocí mého zařízení. Výsledkem byly dvojice snímků, zachycených ve stejném čase. Popis pořizování snímků je vysvětlen v kapitole 3.

Celkově jsem pořídil stovky snímků. Z těchto všech pořízených snímků jsem poté vytvořil kalibrační a rekonstrukční sety.

Jako první jsem pořizoval kalibrační snímky. Celkem jsem nakonec vytvořil tři kalibrační sety různé kvality. Ty my pomáhaly zjišťovat zda chyby, na které jsem narážel, nemohou být způsobeny špatnou kalibrací. Poté jsem vytvořil 10 setů, které my sloužili při implementaci a testování mého projektu. Tyto sety jsou rozdílné velikosti. Nejmenší obsahuje pouze 3 dvojice snímků a ten největší jich obsahuje 17.

Většina rekonstrukčních setů byla pořízena ze vzdálenosti 35-45cm. V této vzdálenosti bylo pořízeno 8 z 10 setů. Zbylé dva sety byly pořízeny ze vzdálenosti mezi 50-60cm.

Jednotlivé snímky v setu mají striktně dané pojmenování. Jméno snímku se skládá z pozici kamery, “left” nebo “right”, a z čísla reprezentujícího pořadí, ve kterém byl snímek v rámci setu pořízen. Pozice kamery a pořadí jsou od sebe odděleny podtržítkem. Snímky jsou ve formátu JPG. Případná ukázka dvojice z datasetu se vyskytuje v kapitole 3.

K tomuto projektu jsem vytvořil program, který se dá použít k rychlému vytvoření dalších setů.

Kapitola 5

Návrh řešení

V této práci se zabývám metodou stereoskopie a výsledkem mé implementace by měla být sparse rekonstrukce. Tato kapitola popisuje jednotlivé části mého návrhu řešení. První podkapitola se věnuje kalibraci kamery a úpravě získaných snímků. Jedná se o dvě úpravy a to o odstranění zkreslení a rektifikaci. V druhé pokapitole se věnuji získávání bodů zájmu pomocí metod SIFT a SURF. Ve třetí podkapitole se zabývám párováním bodů zájmu mezi dvěma snímky. A poslední podkapitola se věnuje triangulaci jednotlivých bodů, pomocí které získáváme naší řídkou rekonstrukci.

5.1 Kalibrace

Poté co jsme získali sérii kalibračních snímků, bylo potřeba provést jejich zpracování, které nás dovedlo k získání hodnot, potřebných pro další práci. Je potřeba provést kalibrace jednotlivých kamer a poté za pomoci získaných parametrů provést stereo kalibraci.

Kamery a objektivy mají vnitřní zkreslení. Obvykle je zkreslení při náhodném pohledu na obraz mírné. Pro řešení naší úlohy musí být tato chyba opravena.

Dva hlavní druhy zkreslení jsou radiální zkreslení a tangenciální zkreslení. Radiální zkreslení způsobuje, že se přímé čáry objevují zakřivené. Radiální zkreslení se zvětšuje, vzdálenější body jsou od středu obrazu.

Distorzi lze korigovat přesunem pixelů obrázku definovaným kalibrací kamery. Pro zmírnění vlivu radiální distorze se používá například následující korekce podle openCV:

$$x_{corrected} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{5.1}$$

$$y_{corrected} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{5.2}$$

kde k_1, k_2 a k_3 jsou radiální koeficienty deformace a $r^2 = x^2 + y^2$. K tangenciálnímu zkreslení, protože čočka pro snímání obrazu není dokonale rovnoběžně vyrovnána s obrazovou rovinou. Toto zkreslení lze zmírnit použitím korekcí v následujícím tvaru:

$$x_{corrected} = x + [2p_1 xy + p_2(r^2 + 2x^2)] \tag{5.3}$$

$$y_{corrected} = y + [p_1(r^2 + 2y^2) + 2p_2 xy] \tag{5.4}$$

kde p_1 a p_2 jsou tangenciální koeficienty deformace. Koeficienty zkreslení paří mezi vnitřní parametry kamer.

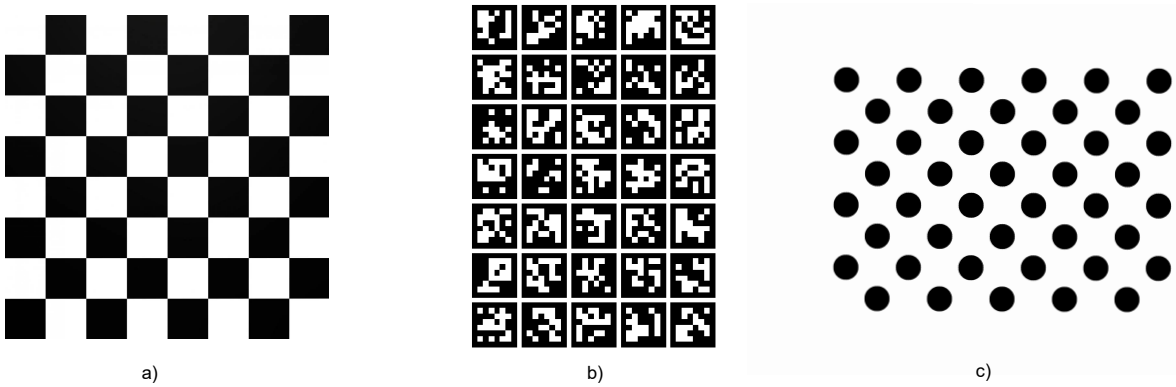
Kalibrace nám slouží k získání vnitřních a vnějších parametrů jednotlivých kamer. Vnitřní parametry jsou specifické pro kameru. Zahrnují informace jako ohnisková vzdálenost (f_x, f_y) a optická centra (c_x, c_y). Ohnisková vzdálenost a optická centra mohou být

použita k vytvoření matice kamery, kterou lze použít k odstranění zkreslení způsobeného čočkami konkrétní kamery. Matice kamery je pro konkrétní kameru jedinečná, takže jakmile je vypočítána, lze ji znovu použít na jiné snímky pořízené stejnou kamerou. Je vyjádřena jako matice 3×3 (5.5).

$$matice\ kamery = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{5.5}$$

Vnější parametry odpovídají vektorům rotace a translace, které převádějí souřadnice 3D bodu do souřadnicového systému. Patří mezi ně také koeficienty zkreslení.

Pro samotnou kalibrace potřebujeme získat odhad vnitřních a vnějších parametrů pro jednotlivé pohledy. Funkce kterou zde používáme je založena na algoritmech Zhanga[22] a Bougueta[5]. Je potřeba specifikovat koordináty 3D bodů objektu a jejich korespondující projekci ve 2D pro každý pohled. Tohoto se dá dosáhnout za použití objektu s nám známou geometrií a lehce detekovatelnými znaky. V této práci je využit kalibrační vzor šachovnice (viz obrazek), který má rozměry 5×4. K detekci šachovnicového vzoru byla použita funkce $findChessboardCorners()$ z knihovny OpenCV. Tato metoda provádí po sobě jdoucí morfologické operace, dokud není identifikováno několik obrysů černé a bílé oblasti a následně jsou z každého obrysu extrahovány čtyři rohy, které obsahují sadu kalibračních bodů. Vzor je rozpoznán pouze pokud jsou identifikovány všechny pravoúhlé čtyřúhelníky. Šachovnicový vzor je pouze jedním z mnoha, které se dají pro kalibraci použít. Další příklady kalibračních vzorů jsou k nalezení na obrázku 5.1.



Obrázek 5.1: Kalibrační vzory a)Šachovnicový b)ArUco c)Bodový

Algoritmus, který je využíván funkcí $calibrateCamera()$ z knihovny OpenCV, provádí následující kroky.

1. Výpočet počátečních vnitřních parametrů.
2. Odhad počáteční pozice kamery, jako by již vnitřní parametry byly známé.
3. Spuštění globálního optimalizačního algoritmu Levenberg-Marquardt, aby se minimalizovala chyba promítání.

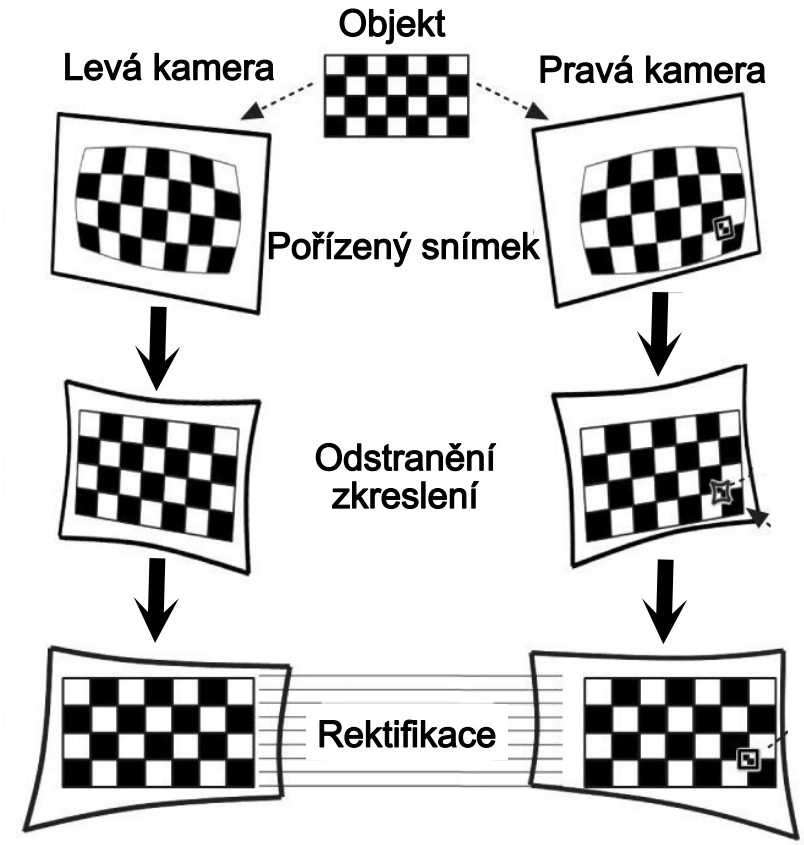
Poté co jsme získali potřebné hodnoty z kalibrace jednotlivých kamer je potřeba odstranit zkreslení a provést rektifikaci.

Dalším krokem je rektifikace. Jedná se o proces transformace používaný k promítání dvou nebo více obrázků do společné obrazové roviny. Tento proces má několik stupňů volnosti a existuje mnoho strategií pro transformaci obrazů do společné roviny. V této práci využíváme algoritmus pro rovinnou rektifikaci [7]. Všechny rektifikované snímky mají následující vlastnosti [12]:

- Všechny epipolární linie jsou rovnoběžné s vodorovnou osou.
- Odpovídající body mají stejné vertikální souřadnice.

Aby bylo možné transformovat původní dvojici obrazů na dvojici rektifikovaných obrazů, je nutné najít projektivní transformaci H. Na H jsou kladeny taková omezení, aby byly splněny výše uvedené vlastnosti. Naší rektifikaci provádíme pomocí rotačních a translačních parametrů, které jsme získali kalibrací našich kamer. Použitý algoritmus má následující kroky:

1. Sestavení rotační matice R_{rect}, která dokáže otočit kameru tak, aby se levý epipól dostal do nekonečna. Epipolární linie budou horizontální.
2. Stejnou rotaci použijeme na pravou kameru.
3. Otočíme pravou kameru o R, což je rotační matice mezi souřadným systémem první a druhé kamery.
4. Pro každý bod levé kamery vypočítáme odpovídající bod v novém stereo rigu.
5. Pro každý bod pravé kamery vypočítáme odpovídající bod v nové stereo rigu.
6. V obou kamerách upravíme měřítko referenčních rámců.



Obrázek 5.2: Ukázka jednotlivých kroků úpravy pořízeného snímku.

5.2 Detekce klíčových bodů

Tato sekce se věnuje detektci klíčových bodů ve snímcích a hledání jejich korespondencí. Pro detekci bodů byly použity algoritmy SIFT[13] a SURF[1]. Principy, na kterých oba algoritmy fungují jsou rozepsány níže.

V prvním kroku je potřeba najít klíčové body. K jejich detekci jsem odzkoušel dva algoritmy. První odzkoušený algoritmus je SIFT (Scale-invariant Feature Transform). SIFT provádí při detekci klíčových bodů následující kroky:

1. Na snímek, ze kterého chceme získat klíčové body použijeme Gaussovské filtry různých stupnic a poté je pořízen rozdíl po sobě následujících Gaussovsky roznamazaných snímků. Klíčové body jsou poté brány jako minima a maxima rozdílu Gaussiánů, které se vyskytují při použití více stupnic.

   Konkrétně rozdíl Gaussianů snímků $D(x, y, \sigma)$ je dán vztahem $D(x, y, \sigma) = L(x, y, k_i\sigma) - L(x, y, k_j\sigma)$, kde $L(x, y, k\sigma)$ je konvoluce původního snímku $I(x, y)$ s Gaussovským rozmazáním $G(x, y, k\sigma)$ s vahou $k\sigma$, příklad, $L(x, y, k\sigma) = G(x, y, k\sigma) * I(x, y)$. Proto snímek rozdílu Gaussiánů mezi vahami $k_i\sigma$ a $k_j\sigma$ je jenom rozdíl Gaussovsky rozmazaných snímků při vahách $k_i\sigma$ a $k_j\sigma$.

Pro scale space extrema detekci pomocí algoritmu SIFT je snímek prvně konvolvován s Gaussovskými rozmazáními při různých stupnicích. Konvolvované snímky jsou seskupeny podle oktávy a je vybrána hodnota k_i, která je vybrána tak, abychom získali pevný počet převrácených obrazů na oktávu. Potom jsou snímky z Gaussovského rozdílu pořízeny ze sousedních Gaussovsky rozmazaných snímků na oktávu. Jakmile jsou snímky Gaussovkých rozmazání získány, jsou klíčové body identifikovány jako lokální minima nebo maxima obrazů Gaussovského rozmazání napříč stupnicemi. To se provádí porovnáním každého pixelu v obrazech Gaussovského rozmazání s jeho osmi sousedy ve stejné váze a devíti odpovídajícími sousedními pixely v každé ze sousedních stupnic. Pokud je hodnota pixelu maximální nebo minimální ze všech porovnávaných pixelů, je vybrána jako kandidát na klíčový bod. Detekce Scale-space extrema vytváří příliš mnoho kandidátů na klíčové body, z nichž některé jsou nestabilní.

2. Dalším krokem v algoritmu je provedení podrobného přizpůsobení blízkým datům pro přesné umístění, stupnici a poměr hlavních zakřivení. Tato informace umožňuje odmítnutí bodů, které mají nízký kontrast nebo jsou špatně lokalizovány podél okraje. Nejprve je pro každý kandidátní klíčový bod použita interpolace blízkých dat k přesnému určení jeho pozice. Přístup, který používá algoritmus SIFT počítá interpolované umístění extrému, což výrazně zlepšuje shodu a stabilitu.[14]

   Provedením interpolace získáme lokaci extrému $\hat{x}$. Pokud je offset $\hat{x}$ v jakékoli dimenzi větší než 0.5, tak to značí, že extrém leží blíž k jinému kandidátnímu klíčovému bodu. V opačném případě se offset přidá do jeho kandidátního klíčového bodu, aby se získal interpolovaný odhad pro umístění extrému. Podobné subpixelové určení umístění scale-space extrema je v implementaci prováděno v reálném čase na základě hybridních pyramid vyvinutých Lindebergem a jeho spolupracovníky.[11]

   Dále musíme zahodit klíčové body s nízkým kontrastem. Abychom se těchto klíčových bodů zbavili vypočteme hodnotu Taylorova rozvoje druhého řádu $D(x)$ s offsetem $\hat{x}$. Pokud je tato hodnota menší než 0.03, kandidátní klíčový bod je zahozen. Jinak je tento bod zachován s finální scale-space lokací $y+\hat{x}$, kde y je původní lokace klíčového bodu.

   Dále je potřeba eliminovat reakci na okraje. Funkce Gaussovského rozdílu bude mít silné reakce podél okrajů, i když kandidátní klíčový bod není odolný vůči malému množství šumu. Abychom zvýšili stabilitu, musíme odstranit klíčové body, které mají špatně určené lokace, ale mají vysoké okrajové odezvy. Pro špatně definované vrcholy ve funkci Gaussovského rozdílu by hlavní zakřivení napříč hranou bylo mnohem větší než hlavní zakřivení podél něj. Nalezení těchto hlavních křivostí je řešením pro vlastní hodnoty Hessianovy matice druhého řádu H 5.6.

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \tag{5.6}$$

   Stopa matice H, kterou je $D_{xx}+D_{yy}$, nám dává sumu dvou vlastních hodnot, zatímco její determinant, kterým je $D_{xx}D_{yy} - D_{xy}^2$, nám poskytuje produkt. Lze ukázat, že se poměr $R = Tr(H)^2/Det(H)$ rovná $(r + 1^2/r)$, které závisí spíše na poměru vlastních čísel než na jejich individuálních hodnotách.

   R je minimální, pokud jsou vlastní hodnoty stejné. Čím vyšší je absolutní rozdíl mezi těmito dvěma vlastními hodnotami, tím vyšší je hodnota R. Z toho vyplývá, že pro

určitý poměr vlastních prahových hodnot r_{th}, pokud je R pro kandidátní klíčové body větší než $(r_{th}+1)^2/r_{th}$, tak je klíčový bod lokalizován a proto odmítnut. Tento krok zpracování pro potlačení odpovědí na hranách je přenos odpovídajícího přístupu v Harrisově operátorovi pro detekci rohu. Rozdíl je v tom, že míra prahování se počítá z Hessiánské matice.

Dále je každému klíčovému bodu přiřazena jedna nebo více orientací na základě lokálních směrů gradientu snímku. Toto je klíčový krok k dosažení invariance k rotaci, protože deskriptor klíčového bodu může být reprezentován relativně k této orientaci, a proto dosáhnout invariance k rotaci snímku. Prvně je pořízen Gaussovsky vyhlazený snímek klíčového bodu $L(x, y, \sigma$ se stupnicí σ , aby mohly všechny výpočty být provedeny bez vlivu stupnice. Pro snímek $L(x, y)$ se stupnicí σ, je velikost přechodu $m(x, y)$ 5.7 a orientace $\theta(x, y)$ 5.8 předpočítána za pomocí rozdílů v pixelech:

$$m(x,y) = \sqrt{(L(x+1,y) - L(x+1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \tag{5.7}$$

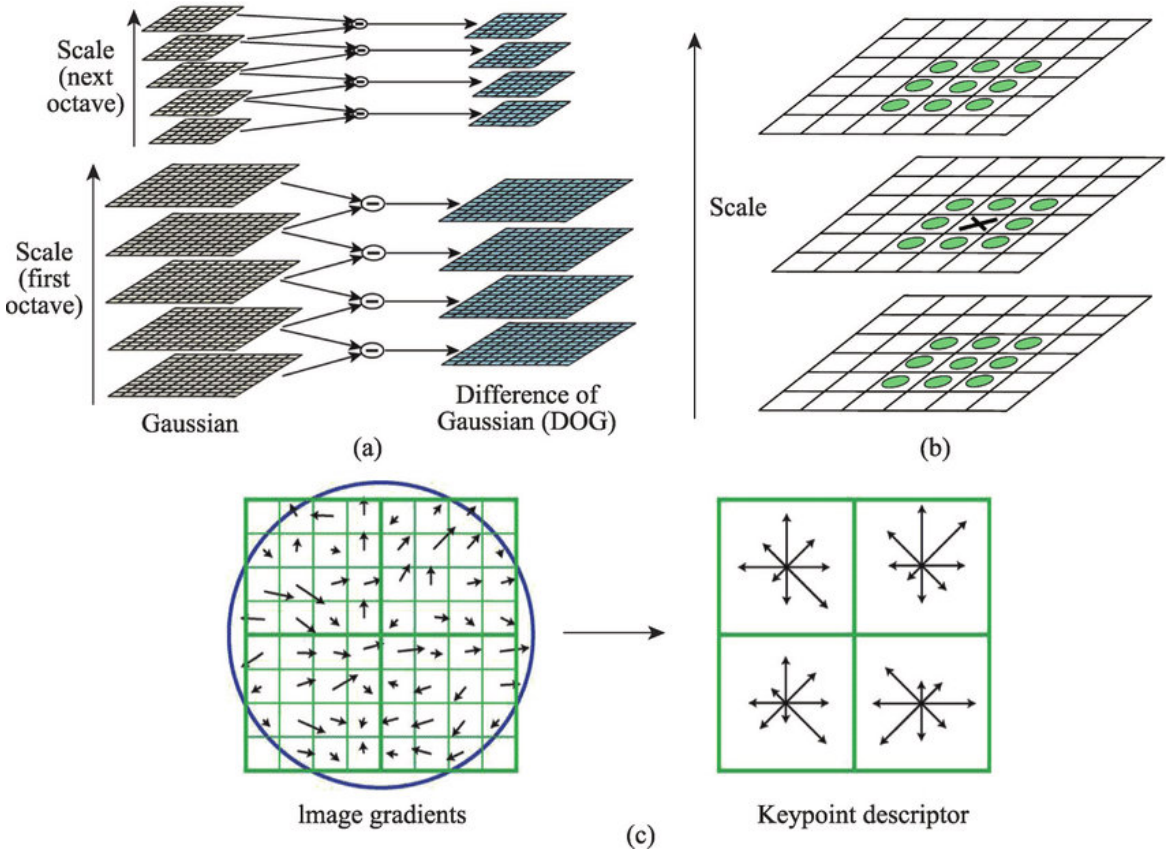
$$\theta(x,y) = atan2(L(x,y+1) - L(x,y-1), L(x+1,y) - L(x-1,y)) \tag{5.8}$$

Výpočet velikosti a směru přechodu se provádí pro každý pixel v sousední oblasti kolem klíčového bodu v Gaussovsky rozmazaném snímku L. Vytvoří se orientační histogram s 36 zásobníky, přičemž každý zásobník pokrývá 10 stupňů. Každý vzorek v sousedním okně přidaný do koše histogramu je vážen svou velikostí gradientu a Gaussovským váženým kruhovým oknem se σ, která je 1,5× větší než stupnice klíčového bodu. Vrcholy v tomto histogramu odpovídají dominantním orientacím. Jakmile je histogram vyplněn, jsou klíčovým bodům přiřazeny orientace odpovídající nejvyššímu vrcholu a místním vrcholům, které jsou v rámci 80% nejvyšších vrcholů. V případě přiřazení více orientací se vytvoří další klíčový bod, který má pro každou další orientaci stejné umístění a stupnici, jako původní klíčový bod.

Předchozí kroky nalezly umístění klíčových bodů v konkrétních stupnicích a přiřadily jim orientace. Nyní chceme vypočítat vektor deskriptoru pro každý klíčový bod tak, aby byl velmi výrazný a částečně invariantní ke zbývajícím variacím. Jako je například osvětlení. Tento krok se provádí na obrázku, jenž je nejblíže stupnicí ke stupnici klíčového bodu. Nejprve se vytvoří sada orientačních histogramů v sousedství 4×4 pixelů s 8 zásobníky. Tyto histogramy jsou počítány z hodnot velikosti a orientace vzorků v oblasti 16×16 kolem klíčového bodu tak, že každý histogram obsahuje vzorky z 4×4 podoblasti původní sousední oblasti. Velikost a orientace gradientu snímku jsou vzorkovány kolem místa, kde je umístěn klíčový bod. Za pomoci stupnice klíčového bodu se vybere úroveň Gaussova rozostření pro snímek. Aby se dosáhlo invariance orientace, jsou souřadnice deskriptoru a orientace gradientu otočeny vzhledem k orientaci klíčového bodu. Velikost je dále vážena Gaussovou funkcí se σ rovnou jedné polovině šířky okna deskriptoru.

Deskriptor se pak stane vektorem všech hodnot těchto histogramů. Protože existuje 16 histogramů, každý s 8 přihrádkami, má vektor 128 prvků. Ačkoli se rozměr deskriptoru jeví jako vysoký, deskriptory s nižším rozměrem než toto nefungují v celé řadě odpovídajících úkolů. [13] Tento vektor je pak normalizován na délku jednotky, aby se zvýšila invariance k afinním změnám v osvětlení. Pro snížení účinků nelineárního osvětlení je použita prahová hodnota 0,2 a vektor je opět normalizován. Proces prahování může zlepšit výsledky porovnávání, i když nejsou přítomny účinky nelineárního

osvětlení. [10] Prahová hodnota 0,2 byla empiricky zvolena a nahrazením pevného prahu jedním systematicky vypočítaným lze výsledky shody zlepšit. [10]



Obrázek 5.3: Konstrukce SIFT deskriptoru [20]. a) Obrazová pyramida. b) Detekce extrémů pro rozdíl Gaussiánů pyramid. (c) Vytvoření deskriptoru klíčového bodu

Druhý algoritmus pro vyhledávání klíčových bodů, který jsem v této práci využil je SURF[1]. Algoritmus SURF poprvé publikovali Herbert Bay, Tinne Tuytelaars a Luc Van Gool a představili je na evropské konferenci o počítačovém vidění v roce 2006. Algoritmus SURF je založen na stejných principech a krocích jako SIFT, ale podrobnosti v každém kroku jsou odlišné. Algoritmus má dvě hlavní části. Těmi jsou detekce klíčových bodů a popis místního sousedství.

1. Prvním krokem je detekce. SURF používá filtry čtvercového tvaru, jako aproximaci Gaussova vyhlazení. Filtrování obrazu čtvercem je mnohem rychlejší, pokud je použit integrální obraz 5.11.

$$S(x,y) = \sum_{i=0}^{x}\sum_{j=0}^{y} I(i,j) \tag{5.9}$$

Součet původního obrázku v obdélníku lze rychle vyhodnotit pomocí integrálního obrazu, který vyžaduje vyhodnocení ve čtyřech rozích obdélníku. SURF používá detektor skvrn založený na Hessianově matici k nalezení klíčových bodů. Determinant

Hessianovy matice se používá jako míra lokální změny kolem bodu a body se vyberou tam, kde je tento determinant maximální. Pokuď máme bod $p = (x, y)$ na snímku I, Hessianova matice $H(p, \sigma)$ v bodě p se stupnicí σ je:

$$H(p, \sigma) = \begin{bmatrix} L_{xx}(p, \sigma) & L_{xy}(p, \sigma) \\ L_{yx}(p, \sigma) & L_{yy}(p, \sigma) \end{bmatrix}, \tag{5.10}$$

kde $L_{xx}(p, \sigma)$, $L_{yx}(p, \sigma)$, $L_{xy}(p, \sigma)$ a $L_{yy}(p, \sigma)$ jsou konvoluce Gaussovského derivátu druhého řádu s obrazem $I = (x, y)$ v bodě p.

Krabicový filtr o velikosti 9×9 je aproximací Gaussiánu se $\sigma = 1, 2$ a představuje nejnižší úroveň pro tzv. blob response maps. Klíčové body lze nalézt v různých stupnicích, částečně proto, že hledání korespondence často vyžaduje srovnávací obrázky, kde jsou vidět v různých stupnicích.

Snímky jsou opakovaně vyhlazovány Gaussovským filtrem a poté jsou vzorkovány, abychom získaly další vyšší úroveň pyramidy. Proto se počítá několik podlaží nebo schodů s různými rozměry masek:

$$\sigma_{approx} = soucasna\ velikost\ filtru\ \times \frac{zkladni\ stupnice\ filtru}{zakladni\ velikost\ filtru} \tag{5.11}$$

Stupnice prostoru je rozdělena do několika oktáv, kde oktáva označuje sérii map odezvy pokrývajících zdvojnásobení stupnice.

V SURF je nejnižší úroveň měřítkového prostoru získána z výstupu filtrů 9×9. Stupnicové prostory v SURF jsou implementovány použitím krabicových filtrů různých velikostí. V souladu s tím je prostor stupnice analyzován zvětšením velikosti filtru spíše než iterativním zmenšením velikosti obrazu.

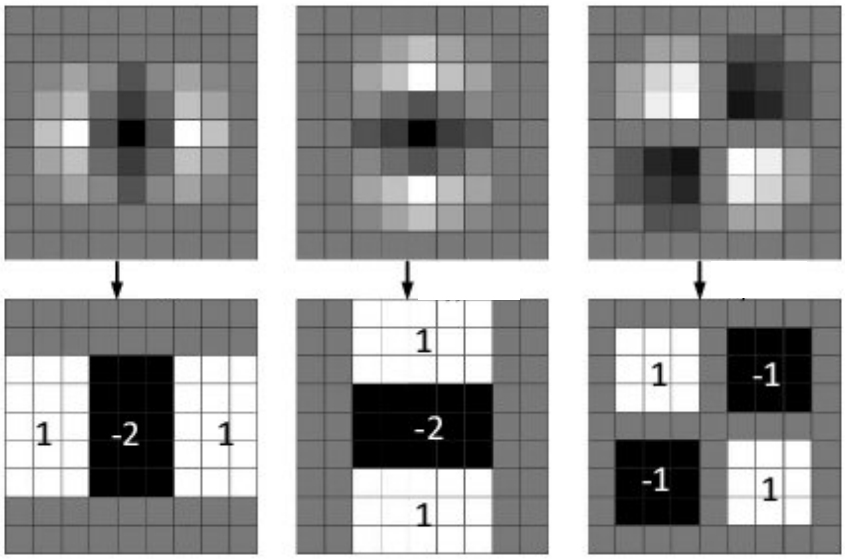
Výstup výše uvedeného filtru 9×9 se považuje za počáteční vrstvu stupnice ve stupnici $s = 1, 2$. Následující vrstvy se získají filtrováním obrazu s postupně většími maskami, s přihlédnutím k diskrétní povaze integrálních obrazů a specifické struktuře filtru. Výsledkem jsou filtry velikosti 9×9, 15×15, 21×21, atd.

Pro lokalizaci klíčových bodů v obraze a přes stupnici se použije ne-maximální potlačení v sousedství $3 \times 3 \times 3$. Maxima determinantu Hessianovy matice jsou pak interpolovány ve stupnici a obrazovém prostoru metodou navrženou Brownem a jeho kolegy. V tomto případě je zvláště důležitá interpolace stupnice prostoru, protože rozdíl v měřítku mezi prvními vrstvami každé oktávy je relativně velký.

2. Druhý krok je věnován desktriptoru, jehož cílem je je poskytnout jedinečný a robustní popis vlastností snímků, např. popisem distribuce intenzity pixelů v sousedství klíčového bodu. Většina deskriptorů je tedy počítána lokálním způsobem, a proto je získán popis pro každé klíčové místo identifikované dříve.

   Dimenze deskriptoru má přímý dopad na jeho výpočetní složitost a přesnost přiřazování bodů. Krátký deskriptor může být odolnější proti změnám vzhledu, ale nemusí nabídnout dostatečnou diskriminaci, a tak může poskytnout příliš mnoho falešných pozitiv.

   První krok spočívá v upevnění reprodukovatelné orientace na základě informací z kruhové oblasti kolem klíčového bodu. Potom vytvoříme čtvercovou oblast zarovnanou s vybranou orientací a z ní extrahujeme deskriptor SURF.



Obrázek 5.4: Laplacián Gaussovy aproximace s boxovými filtry [21]

5.3 Hledání korespondence bodů

Poté co jsme schopni nalézt klíčové body v našich snímcích, musíme být schopni najít korespondující klíčové body v dvojici snímků. Pro tento účel je použita knihovna FLANN(Fast Library for Approximate Nearest Neighbors). [17] Jedná se o knihovnu, která obsahuje kolekci algoritmů optimalizovaných pro rychlé hledání nejbližších sousedů ve velkých datasetech. Z dostupných algoritmů v knihovně FLANN jsem vybral algoritmus pro vyhledávání nejbližších sousedů postavený na kd-stromech [15].

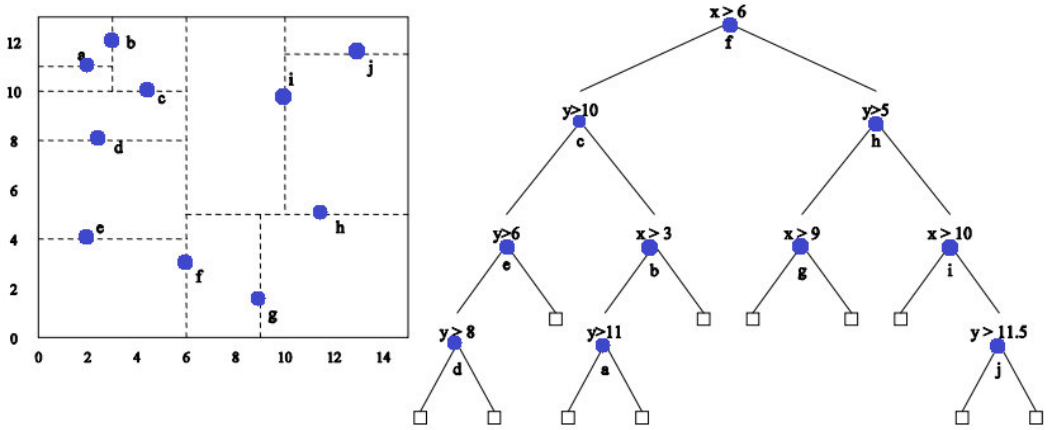
Kd-strom je binární strom, ve kterém je každý listový uzel k-rozměrným bodem. Každý ne-listový uzel lze považovat za implicitně generující rozdělující nadrovinu, která rozděluje prostor na dvě části, známé jako poloprostory. Body vlevo od této nadroviny jsou reprezentovány levým podstromem tohoto uzlu a body vpravo od nadroviny jsou reprezentovány pravým podstromem. Směr nadroviny je vybrán tak, že každý uzel ve stromu je spojen s jednou dimenzí k, s nadrovinou kolmou k ose této dimenze. Pokud je pro konkrétní rozdělení zvolena osa x, všechny body v podstromu s menší hodnotou x, než je uzel, se objeví v levém podstromu a všechny body s větší hodnotou x budou v pravém podstromu. V takovém případě by byla nadrovina nastavena hodnotou bodu x a její normála by byla osa bodu x.[3]

Jelikož existuje mnoho způsobů, jak zvolit dělicí roviny zarovnané s osami, existuje též mnoho různých způsobů, jak konstruovat kd-stromy. Všeobecně uznávaná metoda konstrukce kd-stromu má následující omezení.[4]

- Jak se pohybujeme dolů stromem, cyklujeme osami používanými k výberu dělících rovin.
- Body se vkládají výběrem mediánu bodů vložených do podstromu s ohledem na jejich souřadnice v ose použité k vytvoření dělicí roviny.

Tato metoda vede k získání vyváženého kd-stromu, ve kterém je každý listový uzel přibližně ve stejné vzdálenosti od kořene. Body za mediánem zahrnují pouze ty, které jsou přísně větší než medián. Pro body, které leží na mediánu, je možné definovat funkci, která porovnává body ve všech dimenzích.

Tento algoritmus vytváří invariant. Pro každý uzel jsou všechny uzly v levém podstromu na jedné straně dělicí roviny a všechny uzly v pravém podstromu jsou na druhé straně dělicí roviny. Body, které leží na dělicí rovině, se mohou objevit na jakékoliv straně.



Obrázek 5.5: Vizualizace kd-stromu. [2]

Pokud tedy máme kd-strom, můžeme pro vyhledávání bodů ve stromě použít algoritmus pro vyhledávání nejbližších sousedů. Algoritmus hledání nejbližšího souseda má za cíl najít bod ve stromu, který je nejblíže danému vstupnímu bodu. Toto vyhledávání lze provést efektivně díky vlastnostem stromu, pomocí kterých se rychle odstraní velké části vyhledávacího prostoru. Hledání nejbližšího souseda v kd-stromu probíhá v následujících krocích.

1. Počínaje kořenovým uzlem se algoritmus rekurzivně posune dolů stromem stejným způsobem, jako kdyby měl být hledaný bod vložen. Jde doleva nebo doprava v závislosti na tom, zda je bod menší nebo větší než aktuální uzel v rozdělené dimenze.
2. Jakmile algoritmus dosáhne uzlu, zkontroluje tento uzlový bod a pokud je vzdálenost lepší, je tento uzlový bod uložen jako aktuálně nejlepší.
3. Algoritmus uvolní rekurzi stromu a v každém uzlu provede následující kroky:
   (a) Pokud je aktuální uzel blíž než aktuální nejlepší, stává se aktuálním nejlepším.
   (b) Algoritmus kontroluje, zda na druhé straně dělicí roviny mohou být nějaké body, které jsou blíže hledanému bodu, než ten nejlepší. V koncepci je to provedeno protínáním dělicí nadroviny s hypersférou kolem hledaného bodu, který má poloměr rovný aktuálně nejbližší vzdálenosti. Vzhledem k tomu, že všechny nadroviny jsou zarovnány podle osy, je toto implementováno jako jednoduché srovnání, zda je vzdálenost mezi dělící souřadnicí hledaného bodu a aktuálního uzlu menší, než vzdálenost od hledaného bodu k aktuálnímu nejlepšímu.
      i. Pokud hypersféra protíná rovinu, mohly by existovat bližší body na druhé straně roviny, takže algoritmus musí posunout dolů druhou větev stromu z aktuálního uzlu a hledat bližší body.

ii. Pokud hypersféra neprotíná rozdělující rovinu, algoritmus pokračuje po stromu a celá větev na druhé straně tohoto uzlu je vyloučena.

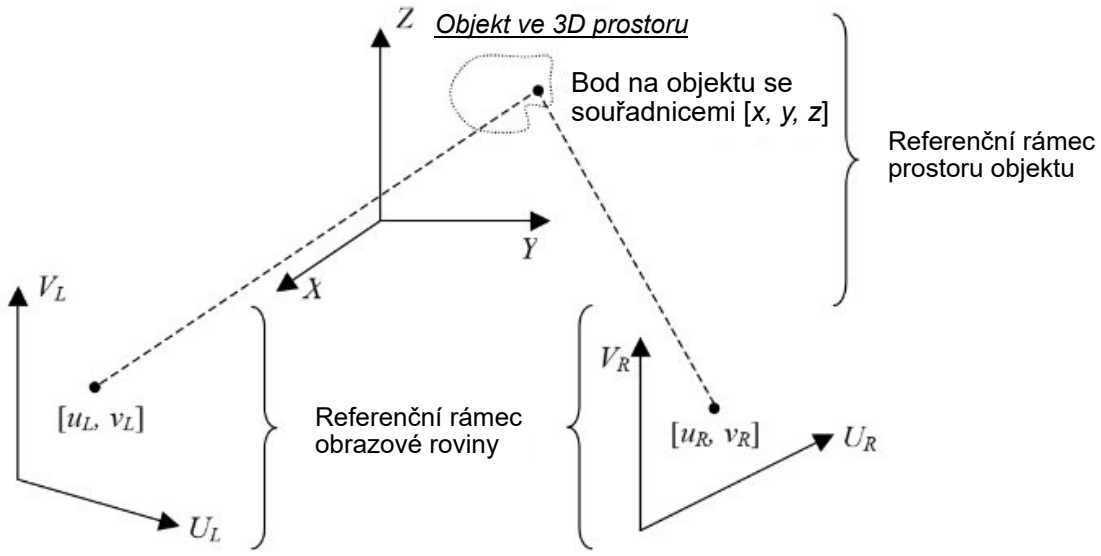
4. Když algoritmus dokončí tento proces pro kořenový uzel, hledání je dokončeno.

5.4 Triangulace

Pro triangulaci bodů využíváme funkci *triangulatePoints*() z knihovny OpenCV. Tato funkce vnitřně používá metodu DLT neboli metodu přímé lineární tranformace[9]. Jedná se o metodu určení trojrozměrného umístění objektu nebo bodů na objektu v prostoru za pomoci dvou pohledů na objekt. Nejprve musíme definovat dva souřadnicové systémy a referenční rámce, ty jsou znázorěny na obrázku 5.6. Velká písmena jsou použita pro označení souřadnicového systému a malá písmena jsou použita k popisu bodů v souřadnicových systémech. Objekt leží v tom, čemu říkáme „referenční rámec prostoru objektu“, který je označován jako souřadnicový systém XYZ.

Ke každému obrazu kamery je přiřazen dvourozměrný referenční rámec. Nazývají se „referenční rámce obrazové roviny“ a označují se pomocí U a R. V DLT budou vždy dva pohledy, které budeme označovat jako „levý“ a „pravý“ pohled. Levé a pravé referenčního rámce obrazové roviny jsou tedy odkazovány na souřadnicové systémy U_LV_L a U_RV_R. Na obrázku 5.6 je jeden referenční rámce pro levou kameru a jeden pro pravou kameru.

Uvažujme bod $[x, y, z]$ umístěný na objektu, jak je znázorněno obrázku 5.6. Tento bod se objeví na levém a pravém obrázku, umístěném podle souřadnic obrázku $[u_L, v_L]$ a $[u_R, v_R]$. Bod $[x, y, z]$ bude mít jednotky délky a $[u_L, v_L]$ a $[u_R, v_R]$ budou mít jednotky v pixelech. Cílem DLT je určit skutečné umístění bodu $[x, y, z]$ na základě u_L, v_L, u_R a v_R. Než toto lze provést, musí být systém kalibrován pomocí bodů známého umístění.



Obrázek 5.6: Referenční rámec prostoru objektu a roviny obrazu a přidružené souřadnicové systémy.

Prvně musíme provést kalibraci, k čemu potřebuje najít matice L a R. Předpokládejme, že známe umístění bodu $[x, y, z]$. Získáme dvojici obrázků, ze které najdeme u_L, v_L, u_R a v_R.

Obrazové body $[u_L, v_L]$ a $[u_R, v_R]$ a bod objektu $[x, y, z]$ mohou být spojeny prostřednictvím řady konstant:

$$u_L = \frac{L_1 x + L_2 y + L_3 z + L_4}{L_9 x + L_{10} y + L_{11} z + 1}, \tag{5.12}$$

$$v_L = \frac{L_5 x + L_6 y + L_7 z + L_8}{L_9 x + L_{10} y + L_{11} z + 1}, \tag{5.13}$$

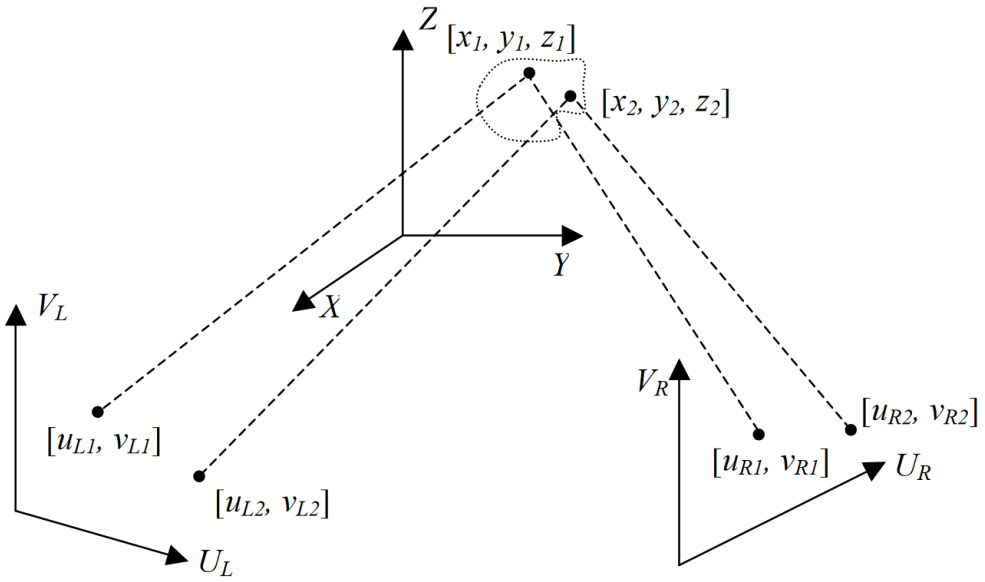
$$u_R = \frac{R_1 x + R_2 y + R_3 z + R_4}{R_9 x + R_{10} y + R_{11} z + 1}, \tag{5.14}$$

$$v_R = \frac{R_5 x + R_6 y + R_7 z + R_8}{R_9 x + R_{10} y + R_{11} z + 1}. \tag{5.15}$$

Z rovnic (5.12) až (5.15) vidíme, že s jedním kalibračním bodem máme sedm známých $(u_L, v_L, u_R, v_R, x, y, z)$, 22 neznámých $(L_1, L_2, L_3, L_4, L_5, L_6, L_7, L_8, L_9, L_{10}, L_{11}, R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8, R_9, R_{10}, R_{11})$ a čtyři rovnice.

K nalezení 22 neznámých potřebujeme alespoň 22 rovnic. To provedeme výběrem více než jednoho kalibračního bodu. Pro každý další kalibrační bod zavedeme čtyři nové rovnice, zatímco konstanty L a R zůstávají stejné. Šest kalibračních bodů poskytne 24 rovnic, takže pro stanovení L a R. musíme získat alespoň šest kalibračních bodů.

Jakmile jsme určili u_L, v_L, u_R, v_R, x, y a z pro alespoň šest bodů, sestavíme je do maticové formy. Abychom to mohli vidět, uvážme dva body, jak je znázorněno na obrázku 5.7.



Obrázek 5.7: Zobrazení dvou kalibračních bodů.

Pro dva body v levém rámečku obrázku jsou rovnice

$$u_{L1} = \frac{L_1 x_1 + L_2 y_1 + L_3 z_1 + L_4}{L_9 x_1 + L_{10} y_1 + L_{11} z_1 + 1}, \tag{5.16}$$

$$v_{L1} = \frac{L_5 x_1 + L_6 y_1 + L_7 z_1 + L_8}{L_9 x_1 + L_{10} y_1 + L_{11} z_1 + 1}, \tag{5.17}$$

$$u_{L1} = \frac{L_1 x_1 + L_2 y_1 + L_3 z_1 + L_4}{L_9 x_1 + L_{10} y_1 + L_{11} z_1 + 1}, \tag{5.18}$$

$$v_{L1} = \frac{L_5 x_1 + L_6 y_1 + L_7 z_1 + L_8}{L_9 x_1 + L_{10} y_1 + L_{11} z_1 + 1}, \tag{5.19}$$

(**??**)

Další rovnice podobné rovnicím (5.16) až (5.19) bude výsledkem pro každý vybraný kalibrační bod. Každý z nich může být uspořádán, jak je ukázáno níže, za použití rovnice (5.16).

$$u_{L1}(L_9 x_1 + L_{10} y_1 + L_{11} z_1 + 1) = L_1 x_1 + L_2 y_1 + L_3 z_1 + L_4, \tag{5.20}$$

$$u_{L1} = L_1 x_1 + L_2 y_1 + L_3 z_1 + L_4 - u_{L1} L_9 x_1 - u_{L1} L_{10} y_1 - u_{L1} L_{11} z_1. \tag{5.21}$$

Podobné rovnice lze získat pro každý $u_{L1} \dots u_{LN}$, $v_{L1} \dots v_{LN}$, $u_{R1} \dots u_{RN}$, $u_{R1} \dots u_{RN}$ a $v_{R1} \dots v_{RN}$, kde N je počet kalibračních bodů. Potřebujeme minimálně 6 bodů ale může jich být víc. Například v_{L1} nám poskytne:

$$v_{L1} = L_5 x_1 + L_6 y_1 + L_7 z_1 + L_8 - v_{L1} L_9 x_1 - v_{L1} L_{10} y_1 - v_{L1} L_{11} z_1. \tag{5.22}$$

Rovnice (5.21) a (5.22) lze sestavit v maticové formě následovně:

$$\begin{bmatrix} x_1 & y_1 & z_1 & 1 & 0 & 0 & 0 & 0 & -u_{L1}x_1 & -u_{L1}y_1 & -u_{L1}z_1 \\ 0 & 0 & 0 & 0 & x_1 & y_1 & z_1 & 1 & -v_{L1}x_1 & -v_{L1}y_1 & -v_{L1}z_1 \end{bmatrix} \begin{bmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \\ L_5 \\ L_6 \\ L_7 \\ L_8 \\ L_9 \\ L_{10} \\ L_{11} \end{bmatrix} = \begin{bmatrix} u_{L1} \\ v_{L1} \end{bmatrix} \tag{5.23}$$

Podobně můžeme přidat do matice, když získáváme až N kalibračních bodů:

$$\underbrace{\begin{bmatrix} x_1 & y_1 & z_1 & 1 & 0 & 0 & 0 & 0 & -u_{L1}x_1 & -u_{L1}y_1 & -u_{L1}z_1 \\ 0 & 0 & 0 & 0 & x_1 & y_1 & z_1 & 1 & -v_{L1}x_1 & -v_{L1}y_1 & -v_{L1}z_1 \\ x_2 & y_2 & z_2 & 1 & 0 & 0 & 0 & 0 & -u_{L2}x_2 & -u_{L2}y_2 & -u_{L2}z_2 \\ 0 & 0 & 0 & 0 & x_2 & y_2 & z_2 & 1 & -v_{L2}x_2 & -v_{L2}y_2 & -v_{L2}z_2 \\ & & & & & \vdots & & & & & \\ x_N & y_N & z_N & 1 & 0 & 0 & 0 & 0 & -u_{LN}x_N & -u_{LN}y_N & -u_{LN}z_N \\ 0 & 0 & 0 & 0 & x_N & y_N & z_N & 1 & -v_{LN}x_N & -v_{LN}y_N & -v_{LN}z_N \end{bmatrix}}_{2N \times 11} \underbrace{\begin{bmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \\ L_5 \\ L_6 \\ L_7 \\ L_8 \\ L_9 \\ L_{10} \\ L_{11} \end{bmatrix}}_{11 \times 1} = \underbrace{\begin{bmatrix} u_{L1} \\ v_{L1} \\ u_{L2} \\ v_{L2} \\ \vdots \\ u_{LN} \\ v_{LN} \end{bmatrix}}_{2N \times 1} \tag{5.24}$$

Hodnoty $L_1 \dots L_{11}$ v rovnici (5.24) jsou jediné neznámé. Můžeme vytvořit podobný maticový systém obsahující u_r, v_r a $R_1 \dots R_{11}$. Levou matici ?? (5.24) označíme jako F_L, matici $L_1 \dots L_{11}$ jako L, a pravou matici ?? jako g_L. Odpovídající matice pro pravý snímek budou F_R, R a g_R. Z čeho vychází, že rovnice (5.24) a její protějšek pro pravý snímek, může být vyjádřen jako

$$F_L L = g_L, \tag{5.25}$$

$$F_R R = g_R. \tag{5.26}$$

Kalibrace se dosáhne řešením pro L a R. Protože F_L a F_R nejsou čtvercové, nemohou být převráceny a L a R musí být místo toho vypočteny metodou nejmenších čtverců. Jednoduchý způsob, jak toho dosáhnout, je použití metody Moore–Penroseovy pseudoinverze. [18][16] Toto je zde zobrazeno pro rovnici (5.25). Prvním krokem je předběžné vynásobení obou stran pomocí F_L^T:

$$F_L^T F_L L = F_L^T g_L. \tag{5.27}$$

Jelikož $F_L^T F_L$ je čtvercová, tak může bý invertovaná. Když produkt předem vynásobíme obě strany $(F_L^T F_L)^{-1}$,

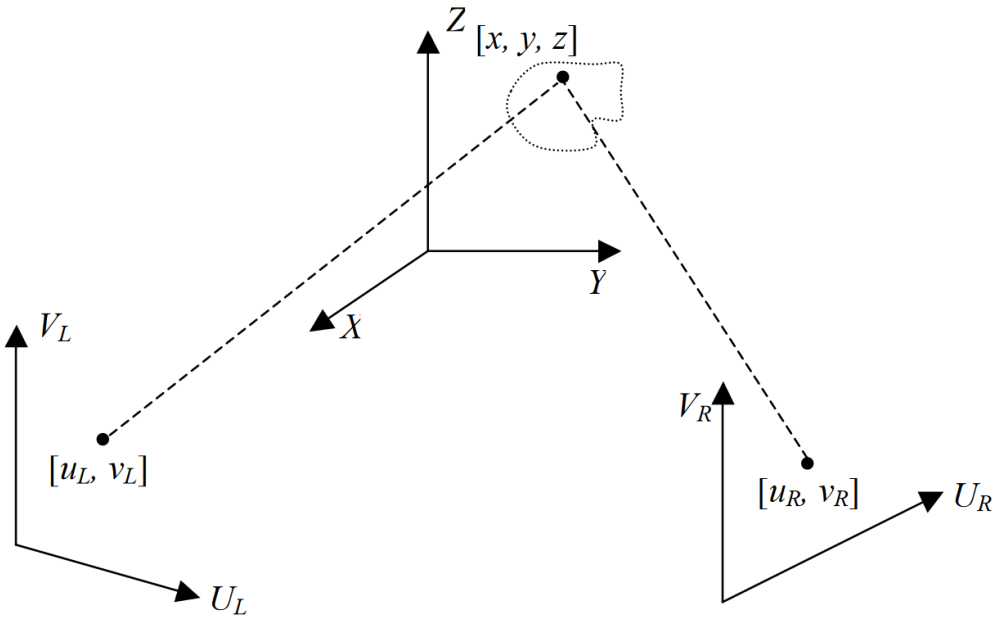
$$(F_L^T F_L)^{-1}(F_L^T F_L)L = (F_L^T F_L)^{-1} F_L^T g_L, \tag{5.28}$$

matice identity je vytvořena na levé straně, takže řešení pro L je získáno jako:

$$L = (F_L^T F_L)^{-1} F_L^T g_L, \tag{5.29}$$

a podobně pro R:

$$R = (F_R^T F_R)^{-1} F_R^T g_R, \tag{5.30}$$



Obrázek 5.8: Zobrazování neznámého bodu.

Jakmilejsme zkalibrovali zobrazovací systém, tim že jsme našli L a R, můžeme nyní najít umístění dalších bodů v kalibrovaném prostoru. To je znázorněno na obr. 3, kde známé jsou u_l, v_L, u_R, v_R, L, R a jediné neznámé jsou x, y a z. Tyto neznámé mohou být nalezeny za použití rovnic (5.12) až (5.15). Máme tedy 3 neznámé a 4 rovnice, uspořádání rovnic a jejich sloučení do maticových nám poskytne

$$\begin{bmatrix} L_1 - L_9u_L & L_2 - L_{10}u_L & L_3 - L_{11}u_L \\ L_5 - L_9u_L & L_6 - L_{10}u_L & L_7 - L_{11}u_L \\ R_1 - R_9u_R & R_2 - R_{10}u_R & R_3 - R_{11}u_R \\ R_1 - R_9u_R & R_6 - R_{10}u_R & R_7 - R_{11}u_R \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} u_L - L_4 \\ v_L - L_8 \\ u_R - R_4 \\ v_R - R_8 \end{bmatrix}. \tag{5.31}$$

Pokud označíme první matici na levé straně jako Q a matici na pravé straně jako q, lze rovnici (5.31) napsat jako:

$$Q \begin{bmatrix} x \\ y \\ z \end{bmatrix} = q,$$

(5.32)

z které můžeme najít souřadnice $[x, y, z]$ pomocí metody Moore–Penroseovy pseudoinverze [18][16]:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = (Q^TQ)^{-1}Q^Tq. \tag{5.33}$$

5.5 Implementace

Program byl vytvořen pomocí jazyka Python a knihovny OpenCV. Knihovnu OpenCV jsem zvolil, jelikož poskytuje velké množství funkcí pro zpracování obrazi.

Kapitola 6

Realizace a experimenty

V této kapitole jsou obsaženy výstupy z jednotlivých částí implementace a experimenty, které byly provedeny.

6.1 Zpracování snímků

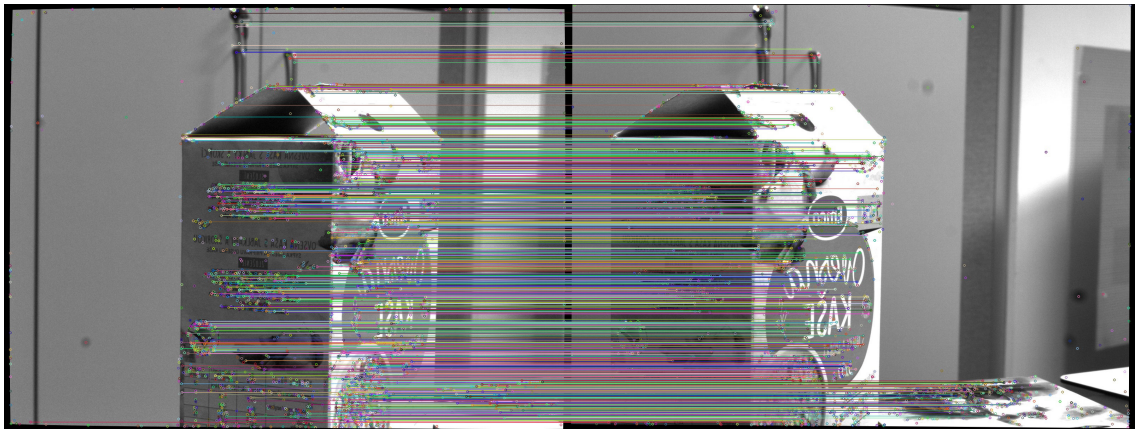
První věc, kterou jsem musel provést bylo zpracování získaných snímků. K tomu bylo potřeba provést kalibraci kamer. Ke kalibraci jsem využil funkce *calibrateCamera*() a *stereoCalibrate*() z knihovny OpenCV. Poté co jsem použil tyto dvě funkce, jsem získal potřebné hodnoty k provedení stereo rektifikace. Tu jsem provedl za použití funkce *stereoRectify*(). Provedení stereo rektifikace mi poté poskytlo všechny potřebné hodnoty pro odstranění zkreslení a následnou rektifikaci dvojic snímků. Jelikož je potřeba odstranit zkreslení z každého získaného snímku bylo potřeba vytvořit mapu. K tomu jsem využil funkci *initUndistortRectifyMap*() z knihovny OpenCV. Tato funkce slouží k výpočtu společné transofrmace pro zkreslení a rektifikaci, jejím výsledkem je poté mapa. Pro každou kameru získáme jednu mapu a tato mapa je poté použita k transformaci jednotlivých snímků. K této transformaci je využita funkce *remap*(). Porovnání původních a upravených snímků je k nalezení na obrázcích níže. Na prvním obrázku je vidět odstranění zkreslení 6.1 a na druhém je za pomoci horizontálních čar ukázáno, že jsou oba snímky promítnuty do jedné roviny 6.2.

Obrázek 6.1: Porovnání snímků před a po odstranění zkreslení a po provedení rektifikace. Horní pár snímků jsou původní snímky. Spodní pár snímků jsou snímky, z kterých bylo odstraněno zkreslení a byla na nich provedena rektifikace.

Obrázek 6.2: Jedná se o stejné snímky jako na 6.1. Jsou zde přidány horizontální linie pro demostraci úspěšné rektifikace.

6.2 Nalezení klíčových bodů a jejich korespondencí v párech snímků

V daleníším kroku jsem se zaměřil na vyhledávání klíčovch bodu a hledání jejich korespondencí. Prvně bylo potřeba detekovat klíčové body. K tomu jsem využil algoritmy SIFT a SURF. Knihovna OpenCV my v tomto poskytla již hotové implementace těchto dvou algoritmů. Ukázka nalezených bodů pomocí algoritmů SIFT a SURF je k nalezení na obrázku 6.4. Poté co jsem našel klíčové body na páru snímků, bylo potřeba najít body, které se objevují na obou snímcích. K tomu jsem využil matcher z knihovny FLANN, příklad je na obrázku 6.3.



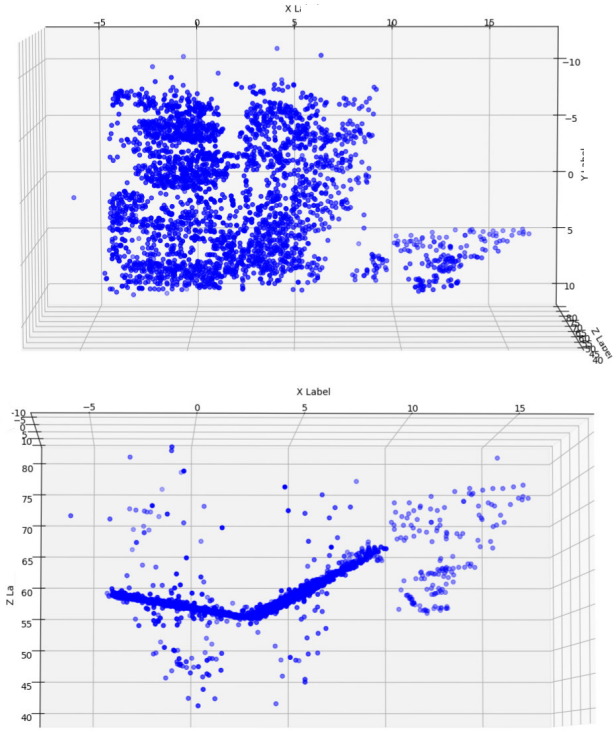
Obrázek 6.3: Výsledek hledání korespondujících bodů mezi dvěma snímky.

Obrázek 6.4: Horní obrázek znázorňuje nalezené klíčové body pomocí metody SIFt a spodní pomocí metody SURF.

6.3 Triangulace 3D bodů

Poté co jsem našel korenspondující dvojice bodů v páru snímků, mohl jsem provést triangulaci. K triangulaci jsem využil funkci *triangulatePoints*() z knihovny OpenCV. V této funkci využívám dvojice bodů, které jsem získal v předchozím kroku a projekční matice

kamer. Tyto projekční matice jsem získal jako jeden z výstupů z funkce $stereoRectify()$. Poté co jsem získal 3D pozice bodů, jsem provedl jejich uložení.



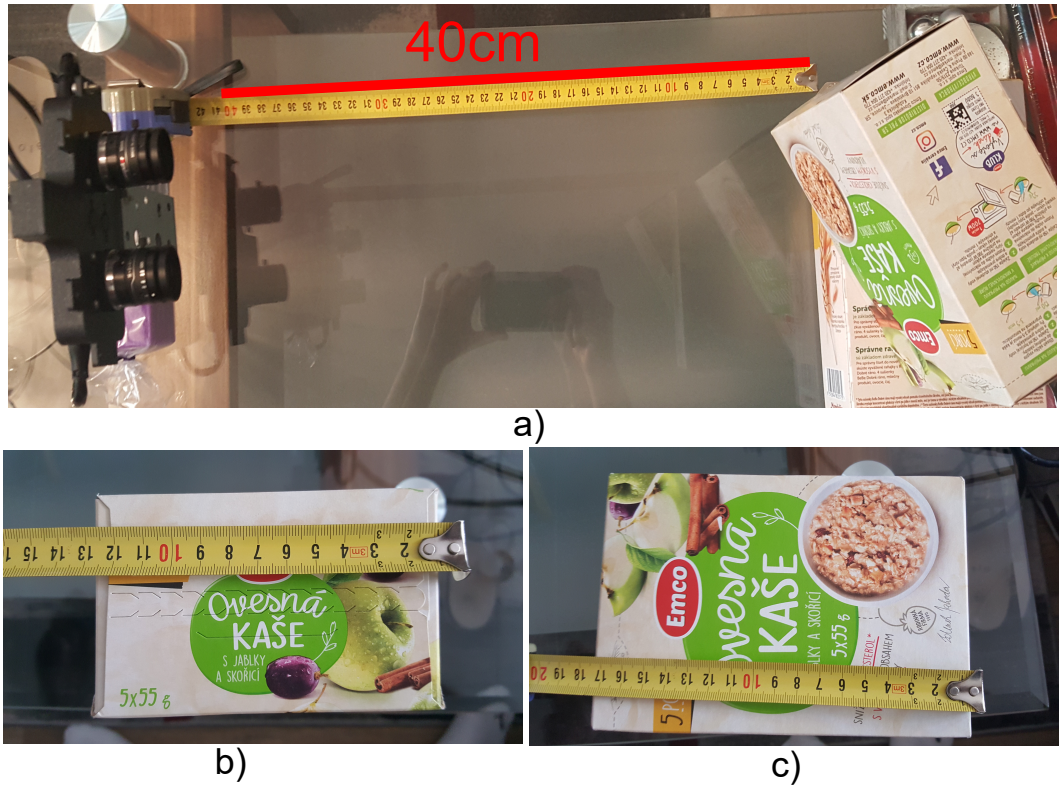
Obrázek 6.5: Jedná se o 3D body získané triangulací z páru snámků. Jedná se o dvojici snímku z obrázku 6.1. Horní obrázek je pohled zepředu a spodní pohled ze shora

Jelikož je mým cílem provádět rekonstrukci z více než dvou snímků, bylo potřeba zajistit aby se všechny body triangulovali do stejného souřadného systému. Abych toho dosáhl musel jsem přepočitávat projekční matici. K tomu jsem potřeboval najít korenspondující 2D body mezi snímkem, který chci použít jako další při triangulaci a snímku, pro který už znám pozici 3D bodů. Poté jsem vytvořil list 2D bodů z nového snímku a jim odpovídající list pozic 3D bodů. Po nalezení těchto sobě odpovídajících bodů jsem použil funkci $solvePnP()$ a z jejího výsledku jsem poté v mé vlastní funkci $calculate_projection_matrix()$ sestavil novou projekční matici.

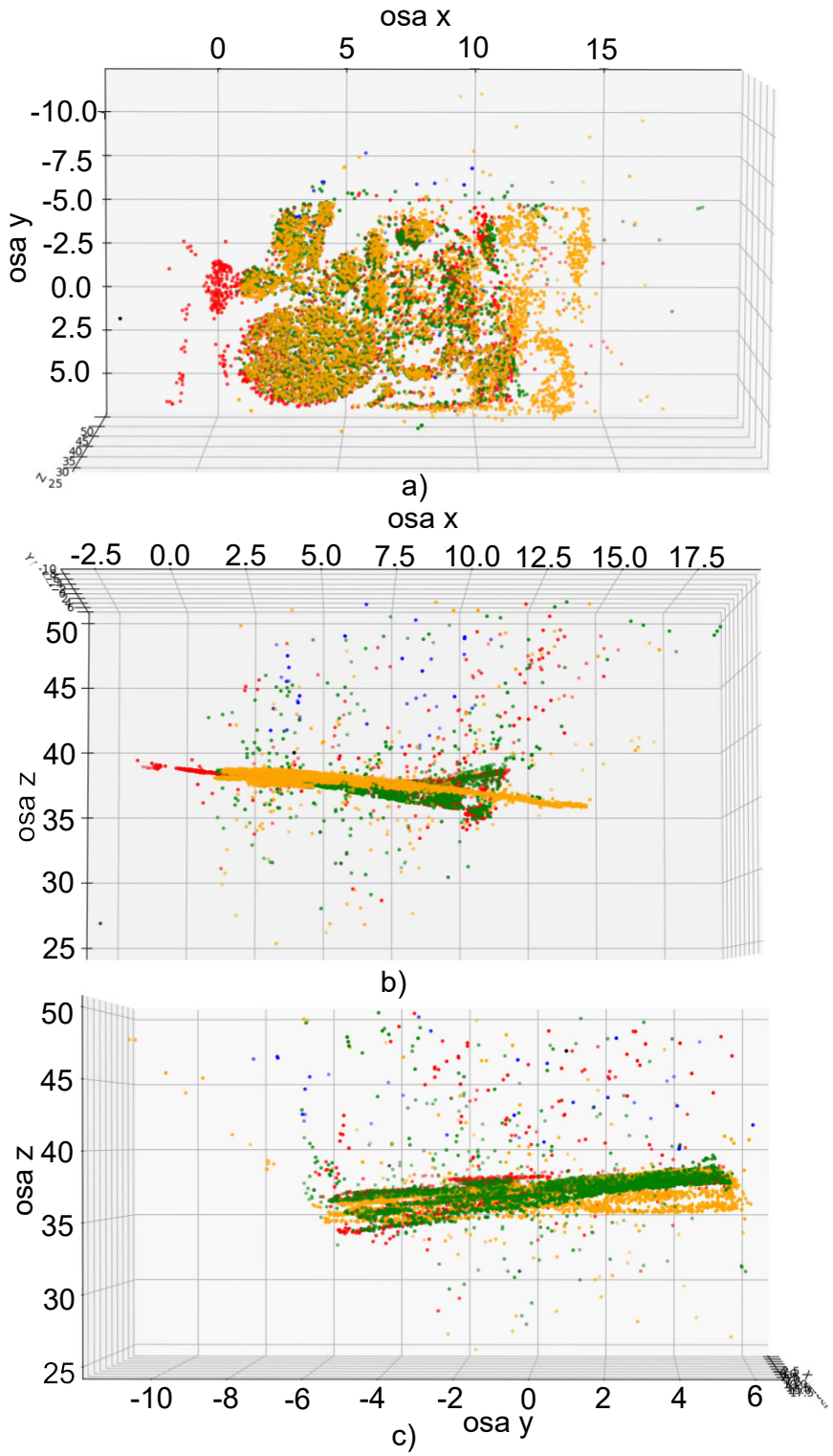
Další věc, kterou jsem chtěl otestovat je zda zrekonstruovaný objekt odpovídá svými rozměry reálnému světu. K tomu jsem si přichystal testovací prostředí. Během pořizování snímků do datasetu jsem v jednom případě změřil výšku a šířku objektu a jeho vzdálenost od zařízení 6.6. Z výsledků je na první pohle zřejmé, že se velikost zrekonstruovaného objektu se blízce přibližuje reálné velikosti, jelikož osy by měly být v jednotkách centimetrů 6.7. V rámci tohoto experimentu jsem provedl sparse rekonstrukci pomocí 6 snímků. Během mých experimentů se mi nezdařilo sestrojit rekonstrukce z více jak 8 snímků. Problémy nastávají při přepočítávání projekční matice. K tomuto problému nejspíše dochází kvůli kumulaci chyb, vycházejících z kvality kalibrace a nedostatečnému filtrování chybných hodnot.

Pokusil jsem se též provést zpětnou projekci 3D bodů do 2D roviny. Při provádění zpětné projekce jsem vytvořil průměrnou odchylku od původního bodu pro souřadnici x i y. Z

výsledku vyplynulo, že odchylka od původního bodu roste s každou novou triangulací. V momentě kdy chyba dosáhne vyšší hodnoty, začne triangulace selhávat. Růst této odchylky bude nejspíše způsoben nepřesnostmi při přepočtu promítací matice.



Obrázek 6.6: a) Ukázka rozložení a vzdálenosti mezi zařízením a snímaným objektem. b) Měření šířky objektu. Šířka vychází přibližně na 12cm. c) Měření výšky objektu. Výška vychází přibližně na 17cm.



Obrázek 6.7: Výsledek sparse rekonstrukce ze 6 snímků. Osy jsou uvedeny v jednotkách cm. a) Pohled zepředu. b) Pohled ze shora. c) Pohled zdola.

Kapitola 7

Diskuze

Dalším krokem v rozšiřování této práce, by mohla bý implementace dense rekonstrukce. Při psaní práce jsem narazil na metodu optical flow, která se dá také využít pro řešení problému korespondence. Tato metoda se jeví, že by mohla být pro tento projekt vhodnější, než metoda mnou použitá. Bylo by též potřeba rozšířit schopnost odfiltrovat extrémní případy. Mojí doměnkou je, že nedostatečná schopnost mého programu odfiltrovat extrémní případy,

V této práci jsem narazil na dva hlavní problémy. Prvním problém se vázal k pořizování datasetu. Původně jsem se snažil pořídit video rekonstruovaného objektu, ale narážel jsem na problém s rozmazáváním snímků, které bylo způsebené pohybem zařízení. Rozmazání se my povedlo odstranit jen za cenu snížení expozičního času. Při snížení exponzičního času ale utrpěla kvalita snímků. Zlepšení kvality pořizovaných snímků, by se nejspíše dalo dosáhnout rozšířením zařízení o nějaký zdroj osvětlení.

Druhým problémem, na který jsem narazil, byly špatné přepočty projekční matice. Můj odhad je, že tento problém může být způsoben kvalitou kalibrace a nedostatečnou schopností mého programu odfiltrovat extrémní případy.

Kapitola 8

Závěr

Tato práce se zabývala 3D rekonstrukcí objektů pomocí zařízení, které je popsáné v kapitole 3. Pro 3D rekonstrukci jsem si zvolil metodu stereoskopie a výsledkem je sparse rekonstrukce. V první části práce jsem musel jsem vytvořit program, který by mi umožnil pořizovat snímky. K vyvoření programi jsem použil knihovnu pypylon od výrobce kamery. Podařilo se mi vytvořit program, který umožňuje získávat kalibrační snímky a také snímky objektu, který chceme rekonstruovat. Pokusil jsem se též tento program rozšířit o možnost záznamu videíí, ale narazil jsem zde na problémy, které jsou podrobněji popsány v kapitole 3.

V druhé části práce jsem se věnoval zpracování získaných snímků. Mým cílem zde bylo odstranit zkreslení a poté být schopen rektifikovat dvojice snímků. V rámci toho jsem musel provést kalibraci používaných kamer. K této kalibraci jsem využil funkce z knihovny OpenCV a získal jsem potřebné hodnoty. Díky získaným hodnotám se mi podařilo úspěšně odstranit zkreslení a také provádět rektifikaci dvojic snímků.

V třetí části jsem řešil hledání korespondujících bodů, mezi dvěma snímky. Prvním krokem bylo najít klíčové body na jednotlivých snímcích. K tomu jsem prvně využil algoritmus SIFT. Později jsem se dozvěděl, že algoritmus SURF by měl být pro moje řešení lepší. Poté co jsem oba algoritmy odzkoušel, rozhodl jsem se přidat možnost použít i algoritmus SURF. Z mého experimentování my vyšlo, že SURF byl schopný najít větší počet klíčových bodů 6.4. Poté co jsem našel klíčové body, bylo potřeba najít jejich korespondence. K tomu jsem zvolil knihovnu FLANN, která my poskytla algoritmus pro vyhledávání nejbližších sousedů postavený na kd-stromech.

Když jsem získal korespondující body, mohl jsem přejít k finálnímu kroku. V tomto kroku jsem provedl triangulaci bodů a tím jsem získal pozice těchto bodů ve 3D prostoru. Prvně jsem provedl triangulaci pouze z jedné dvojice snímků. Abych mohl pomocí triangulace do stejného souřádného systému, již získaných bodů, získat 3D z dalších snímků, musel jsem přepočítat projekční matici. Poté jsem provedl experiment, kterým jsem chtěl ověřit, zda má implementace správně odhaduje vzdálenosti. Tento experiment se poměrně vydařil. Při rekonstrukci z více než dvou snímků jsem poté narazil na problém s přepočtem projekční matice. Tento problém se mi vyřešit nepodařilo, takže jsem zůstal omezený v tom, z jakého maximálního počtu snímků mohu rekonstrukci provést.

Literatura

[1] Bay, H., Ess, A., Tuytelaars, T. a Gool, L. V. Speeded-Up Robust Features (SURF). *Comput. Vis. Image Underst.* 2008, sv. 110, s. 346–359.

[2] Ben Halima, N. a Hosam, O. Bag of Words Based Surveillance System Using Support Vector Machines. *International Journal of Security and Its Applications.* Duben 2016, sv. 10, s. 331–346. DOI: 10.14257/ijsia.2016.10.4.30.

[3] Bentley, J. L. Multidimensional binary search trees used for associative searching. *Commun. ACM.* 1975, sv. 18, s. 509–517.

[4] Berg, M., Cheong, O., Kreveld, M. a Overmars, M. Orthogonal Range Searching. In:. Leden 2008. DOI: 10.1007/978-3-540-77974-2_5.

[5] Bouguet, J.-Y. Camera calibration toolbox for matlab. In:. 2001.

[6] Callen, D. *Configuring a 3D Triangulation Vision System* [online]. 2017 [cit. 2020-20-07]. Dostupné z: https://www.photonics.com/Articles/Configuring_a_3D_Triangulation_Vision_System/a62061.

[7] Fusiello, A., Trucco, E. a Verri, A. A Compact Algorithm for Rectification of Stereo Pairs. Říjen 2000, sv. 12. DOI: 10.1007/s001380050120.

[8] Hariyama, M., Yokoyama, N. a Kameyama, M. Design of a Trinocular-Stereo-Vision VLSI Processor Based on Optimal Scheduling. *IEICE Transactions on Electronics.* Duben 2008, 91-C, s. 479–486. DOI: 10.1093/ietele/e91-c.4.479.

[9] Hartley, R. a Zisserman, A. Multiple view geometry in computer vision. In:. Cambridge university pres, Leden 2003, s. 312.

[10] Kirchner, M. Automatic Thresholding of SIFT Descriptors. In:. Září 2016, s. 291–295. DOI: 10.1109/ICIP.2016.7532365.

[11] Lindeberg, T. a Bretzner, L. Real-time scale selection in hybrid multi-scale representations. In:. Srpen 2003, sv. 2695, s. 148–163. DOI: 10.1007/3-540-44935-3_11.

[12] Loop, C. a Zhang, Z. Computing rectifying homographies for stereo vision. In:. únor 1999, sv. 1, s. –131 Vol. 1. DOI: 10.1109/CVPR.1999.786928. ISBN 0-7695-0149-4.

[13] Lowe, D. Object Recognition from Local Scale-Invariant Features. *Proceedings of the IEEE International Conference on Computer Vision.* Leden 2001, sv. 2.

[14] Lowe, D. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*. Listopad 2004, sv. 60, s. 91–110. DOI: 10.1023/B%3AVISI.0000029664.99615.94.

[15] Moore, A. W. *An intoductory tutorial on kd-trees* [online]. 1991 [cit. 2020-14-07]. Dostupné z: https://www.ri.cmu.edu/pub_files/pub1/moore_andrew_1991_1/moore_andrew_1991_1.pdf.

[16] Moore, E. H. On the reciprocal of the general algebraic matrix. *Bulletin of the American Mathematical Society*. 1920, sv. 18, s. 509–517.

[17] Muja, M. FLANN -Fast Library for Approximate Nearest Neighbors User Manual. Červenec 2020.

[18] Penrose, R. A generalized inverse for matrices. *Mathematical Proceedings of the Cambridge Philosophical Society*. Cambridge University Press. 1955, sv. 51, č. 3, s. 406–413. DOI: 10.1017/S0305004100030401.

[19] Simões, F., Almeida, M. W. S., Pinheiro, M., Anjos, R. D., Santos, A. et al. Challenges in 3D Reconstruction from Images for Difficult Large-Scale Objects: A Study on the Modeling of Electrical Substations. *2012 14th Symposium on Virtual and Augmented Reality*. 2012, s. 74–83.

[20] Wei, L., Zhou, Z., Lang, C. a Zhou, Y. A survey on image and video stitching. *Virtual Reality Intelligent Hardware*. Únor 2019, sv. 1, s. 47. DOI: 10.3724/SP.J.2096-5796.2018.0008.

[21] Yang, G., Li, D., Ru, G., Cao, J. a Jin, W. Body Height Estimation System Based on Binocular Vision. *International Journal of Online Engineering (iJOE)*. Duben 2018, sv. 14, s. 177. DOI: 10.3991/ijoe.v14i04.8400.

[22] Zhang, Z. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2000, sv. 22, č. 11, s. 1330–1334.

Příloha A

Obsah DVD

Obsah přiloženého DVD:

- **data/** - adresář datasetu
- **source/** - adresář se zdrojovými kódy
- **text/** - adresář s textovou částí práce
- **video/** - adresář s videem