

TECHNICKÁ UNIVERZITA V LIBERCI

FAKULTA MECHATRONIKY A MEZIOBOROVÝCH INŽENÝRSKÝCH STUDIÍ



DISERTAČNÍ PRÁCE

VYUŽITÍ POČÍTAČOVÉHO VIDĚNÍ V ROBOTICE

TECHNICKÁ UNIVERZITA V LIBERCI

FAKULTA MECHATRONIKY A MEZIOBOROVÝCH INŽENÝRSKÝCH STUDIÍ

Studijní program: P 2612 Elektrotechnika a informatika

Studijní obor: 2612V045 Technická kybernetika

VYUŽITÍ POČÍTAČOVÉHO VIDĚNÍ V ROBOTICE

DISERTAČNÍ PRÁCE

Autor: Ing. Jakub Štilec

Vedoucí práce: Doc. Václav Záda, CSc.

V Liberci 28.4.2008

Abstrakt

V mnoha případech musí být průmyslový i mobilní robot schopen vnímat a rozpoznat prostředí, ve kterém se nachází. Důležitým tématem v této oblasti jsou metody stereo vidění. Jedná se o proces konstrukce trojrozměrného modelu scény, založený na zpracování dvou a více obrazů této scény, nasnímaných z různých míst. Hlavním problémem je vytvoření mapy disparity (hloubkové mapy obrazu). Mnoho výpočetně rychlých metod není schopno zpracovat scény s nedostatečným, nebo žádným výskytem textury. Pro tento typ scén a také pro scény s překrývajícími s objekty, nebo odlesky dochází ke vzniku velkého počtu chyb. Tato práce se zaměřuje na komplexní proces stereo vidění. Je zde prezentována metoda automatické kalibrace kamer. Dále uvádíme implementaci metody stereovidění rozšířené o přidání filtru výsledné mapy disparity. Takto upravená metoda je schopna zpracovávat scény s nedostatkem textury a snížit množství výskytu chyb ve výstupní mapě disparity. Navíc bylo realizováno urychlení celé metody, pomocí generování počátečního odhadu mapy disparity, založeném na detekci uměle přidaných vizuálních značek do scény. V závěru práce je předvedena úprava použité metody stereovidění pro distribuovaný výpočet.

Abstract

In many cases, the industrial and mobile robots must be able to sense and recognize their environment. An important research topic in the field is stereo vision. Stereo vision is the process of constructing a three dimensional model of a scene through processing of two or more planar images of this scene. The main problem is the construction of a disparity map, because it is computationally intensive. Next problem is quality of output disparity map. Many of computationally relatively fast methods aren't able to process scenes with a lack of texture. Those produce big number of faults for these types of scenes and also for scenes with occlusion objects. This thesis is focused on the whole process of stereo vision. A method for automatic camera calibration is presented. We implemented method of stereo matching of rectified images and extended it by adding post processing of output disparity map. Extended method is able to process scenes with a lack of texture and decrease number of faults in disparity map. Additionally computational time was decreased by use of initial estimation based on detection of artificial visual targets in the scene. Finally we proposed a method for parallelization of stereo matching algorithm.

Obsah

Seznam obrázků.....	viii
Seznam tabulek	x
1. Úvod.....	1
2. Detekce efektoru ramene robotu	4
2.1. Požadavky na kalibraci kamery	5
2.2. Detekce polohy efektoru ramene robotu.....	6
2.3. Vlastní realizace	7
2.4. Experimentální výsledky	10
3. Model kamery a kalibrace	12
3.1. Úvod	12
3.2. Model kamery	16
3.2.1. Pozice a orientace kamery.....	20
3.2.2. Perspektivní projekce	20
3.2.3. Poruchy objektivu	21
3.2.4. Obraz v počítači	25
3.3. Kalibrační metody	26
3.4.1. Hallova metoda.....	27
3.4.2. Faugerasova metoda	29
3.4.3. Faugerasova metoda s radiálním zkreslením	34
3.4. Určení přesnosti	36

3.5.1.	Měření v trojrozměrném prostoru	37
3.5.2.	Měření ve dvourozměrném prostoru.....	41
3.5.	Experimentální výsledky	41
4.	Prostorová rekonstrukce	45
4.1.	Úvod	45
4.2.	Princip triangulace	47
4.3.	Epipolární geometrie	49
3.5.1.	Výpočet polohy optických středů kamer.....	51
3.5.2.	Výpočet polohy epipólů.....	51
3.5.3.	Urychlení výpočtu směrového vektoru epipolární přímky.....	52
3.5.4.	Rektifikace	53
5.	Hlubkové vnímání	55
5.1.	Omezení množiny bodů pro párování	58
5.2.	Klasifikace metod stereovidění.....	59
5.3.	Metody založené na hledání význačných míst	60
5.3.1.	Korelační stereo metody	60
5.3.2.	Iterativní stereo metody.....	61
5.3.3.	Konstruktivní stereo metody.....	61
5.4.	Symbolické párování.....	62
5.4.1.	Metody založené na porovnávání intensity	62
5.4.2.	Metoda založená na fázovém posunutí FFT	63
6.	Realizace metody stereo vidění.....	64

6.1.	Úvod	64
6.2.	Princip metody stereovidění.....	66
6.2.1.	Hodnotící funkce.....	66
6.2.2.	Vyjádření odlišnosti pixelů	67
6.3.	Možnosti vylepšení metody	68
6.3.1.	Omezení usnadňujících systematické a efektivní vyhledávání	68
6.3.2.	Omezení zlepšující funkci algoritmu na netexturovaných regionech	68
6.4.	Propagace informací mezi řádky.....	70
6.5.	Generování počátečního odhadu	71
6.5.1.	Vizuální značky.....	71
6.5.2.	Kategorizace vizuálních značek	72
6.5.3.	Využití vizuálních značek	73
6.6.	Experimentální výsledky	74
6.6.1.	Příklad detekce jednoduché scény	75
6.6.2.	Příklad detekce složené nepřekrývající se scény	77
6.6.3.	Příklad detekce složené překrývající se scény	79
6.6.4.	Příklad detekce kruhové plochy	81
6.6.5.	Příklad detekce válcové plochy	83
6.6.6.	Příklad detekce komplikované scény	85
6.6.7.	Příklad generování počátečního odhadu pomocí vizuálních značek.....	87
6.7.	Zhodnocení výsledků	88
7.	Distribuované počítačové vidění	91

7.1.	Návrh.....	91
7.2.	Technologie.....	94
7.2.1.	Přímá komunikace TCP/IP	94
7.2.2.	Binární protokol.....	95
7.2.3.	Textový protokol.....	97
7.2.4.	XML protokol	97
7.3.	Web Services.....	99
7.3.1.	Specifikace	100
7.4.	Rychlost komunikace	101
7.5.	Návrh webové služby na straně serveru.....	102
7.6.	Zhodnocení výsledků	105
8.	Závěr	107
9.	Literatura	111
10.	Vlastní publikace.....	117

Seznam obrázků

Obr. 2.1: Efektor ramene robotu.....	4
Obr. 2.2: Kalibrační terčik	5
Obr. 2.3: Nasnímaná scéna s vypnutým a zapnutým LED indikátorem	8
Obr. 2.4: Detekované pozice LED indikátoru.....	9
Obr. 3.1: Modelování kamery krok č. 1	18
Obr. 3.2: Modelování kamery krok č. 2	18
Obr. 3.3: Modelování kamery krok č. 3	19
Obr. 3.4: Modelování kamery krok č. 4	19
Obr. 3.5: Radiální zkreslení	23
Obr. 3.6: Ukázka efektu radiálního zkreslení.....	23
Obr. 3.7: Tangenciální zkreslení	24
Obr. 3.8: Ukázka efektu tangenciálního zkreslení	24
Obr. 3.9: Princip měření přesnosti	37
Obr. 3.10: Princip měření přesnosti	38
Obr. 3.11: Princip měření přesnosti	40
Obr. 3.12: Princip měření přesnosti	40
Obr. 4.1: Systém prostorového vidění robotu	46
Obr. 4.2: Snímání scény pomocí dvou kamer	47
Obr. 4.3: Příklad epipolární přímky	50
Obr. 6.1: Příklad occlusion objektů	65
Obr. 6.2: Hlavní kategorie vizuálních značek.....	73
Obr. 6.3: Dvojice rektifikovaných obrazů	75
Obr. 6.4: Výsledná mapa disparity	75
Obr. 6.5: Průběh jasové funkce na párových řádcích rektifikovaných obrazů	76
Obr. 6.6: Dvojice rektifikovaných obrazů	77

Obr. 6.7: Výsledná mapa disparity	77
Obr. 6.8: Průběh jasové funkce na párových řádcích rektifikovaných obrazů	78
Obr. 6.9: Dvojice rektifikovaných obrazů	79
Obr. 6.10: Výsledná mapa disparity	79
Obr. 6.11: Průběh jasové funkce na párových řádcích rektifikovaných obrazů	80
Obr. 6.12: Dvojice rektifikovaných obrazů	81
Obr. 6.13: Výsledná mapa disparity	81
Obr. 6.14: Průběh jasové funkce na párových řádcích rektifikovaných obrazů	82
Obr. 6.15: Dvojice rektifikovaných obrazů	83
Obr. 6.16: Výsledná mapa disparity	83
Obr. 6.17: Průběh jasové funkce na párových řádcích rektifikovaných obrazů	84
Obr. 6.18: Dvojice rektifikovaných obrazů	85
Obr. 6.19: Výsledná mapa disparity	85
Obr. 6.20: Průběh jasové funkce na párových řádcích rektifikovaných obrazů	86
Obr. 6.21: Dvojice rektifikovaných obrazů	87
Obr. 6.22: Výsledná mapa disparity	88
Obr. 7.1: Schéma architektury klient-server	92
Obr. 7.2: Schéma distribuce metody stereo vidění	93
Obr. 7.3: Proces metody stereo vidění	94
Obr. 7.4: Příklad datové struktury přenášeného obrazu	96
Obr. 7.5: Příklad dvou různých datových struktur přenášeného obrazu	97
Obr. 7.6: Schéma kombinace řízeného a neřízeného kódu	102
Obr. 7.7: Schéma distribuce metody stereo vidění	104

Seznam tabulek

Tab. 2.1: Doba analýzy jednoho snímku v závislosti na jeho rozlišení	10
Tab. 2.2: Časy snímání dvojic trojrozměrných a dvourozměrných bodů pro kalibraci	11
Tab. 3.1: Výpočetní rychlost kalibračních metod	42
Tab. 3.2: Porovnání trojrozměrné přesnosti	43
Tab. 3.3: Porovnání dvourozměrné přesnosti v nezkresleném obrazu.....	43
Tab. 3.4: Porovnání dvourozměrné přesnosti ve zkresleném obrazu.....	43
Tab. 6.1: Tabulka přehledu výsledků generování mapy disparity	88
Tab. 6.2: Tabulka porovnání výpočetních časů v závislosti na rozlišení bez použití počátečního odhadu pomocí visuálních značek.....	90
Tab. 6.3: Tabulka porovnání výpočetních časů v závislosti na rozlišení bez použití počátečního odhadu pomocí visuálních značek.....	90
Tab. 7.1: Typy příkazů použitých v distribuovaném vidění	103
Tab. 7.2: Typy odpovědí na příkaz použitých v distribuovaném vidění	104
Tab. 7.3: Tabulka výpočetních časů na samostatných počítačích	105

1. Úvod

Pokud by se měly vyplnit předpovědi významných kapacit v oblasti robotiky, měly by naše domovy být na hranici velkého přelomu v oblasti automatizace a robotizace. Inteligentní roboty by se již brzy měly objevit v běžném životě v širokém množství různých forem. Budou schopny bezpečného pohybu v prostředí, ve kterém se zorientují zejména pomocí vizuální informace získané pomocí kamer. Naopak jejich vlastníci s nimi budou schopni komunikovat za použití přirozených metod komunikace, jako jsou gesta a řeč.

Vývoj autonomního robotu vyžaduje pokročilé znalosti z mnoha oblastí vědy. Jedná se o mechaniku, automatické řízení, mikroelektroniku, elektrotechniku, senzorové systémy, softwarové inženýrství a metody umělé inteligence. K dosažení požadovaného cíle lze naštěstí požadavky kladené na vývoj robotu rozdělit do několika menších množin, které mohou být řešeny samostatně.

Jednou z těchto samostatných oblastí je zpracování vizuální informace. Na základě jejího rozpoznávání a zpracování je možné naplánovat trajektorie pohybu v prostoru s překážky, manipulaci s objekty a v neposlední řadě lze také využít vizuální zpětné vazby. Tato schopnost je zásadní pro široké spektrum aplikací od průmyslových robotů, mobilních robotů až po inteligentní spotřebiče. Přitom je jediným požadovaným vstupem obraz z kamer, kde přes vysokou cenu průmyslových kamer, cena běžných počítačových kamer klesá a při použití vhodné metody zpracování, je jejich výstup postačující. Téma trojrozměrného počítačového vidění je velice aktuální a v této oblasti stále probíhá intenzivní vývoj a stále vznikají nové metody řešení. Přes všechnu snahu, ale současný stav poznání stále neumožňuje masivní rozšíření autonomních systémů v běžném životě. Právě z tohoto důvodu je trojrozměrné počítačové vidění obsahem této práce.

Z důvodu, že jednotlivé části procesu stereovidění jsou v dostupné literatuře uváděny samostatně a na izolovaných případech, je cílem této práce, popsat celý proces jako jediný celek, realizovaný pro konkrétní aplikaci s robotickým ramenem firmy ABB. Celý proces lze rozdělit do následujících částí:

- rozpoznání bodů ve scéně pro následnou kalibraci
- kalibrace kamer
- prostorová rekonstrukce, epipolární geometrie a rektifikace obrazů
- sestavení mapy disparity
- distribuovaný výpočet

Rozpoznání bodů ve scéně pro následnou kalibraci je proces specifický pro konkrétní aplikaci, proto je cílem této práce nalézt vhodnou, rychlou a zároveň jednoduchou metodu jeho řešení, založenou na použití aktivně řízených LED ve snímané scéně. Popis tohoto řešení lze nalézt v kapitole 2.

Kalibrace kamer je podstatným krokem pro jakékoliv další zpracování obrazu. Jedná se o oblast diskutovanou v mnoha vědeckých publikacích. Co zde, ale chybí je přehled některých metod vhodných pro použití v oblasti robotiky spolu s jejich porovnáním na konkrétním případě. Na tento problém se budeme snažit odpovědět v kapitole 3.

Přesto, že je epipolární geometrie velice dobře popsána, je důležité ji zmínit. Nebudeme se pouze zaměřovat na epipolární geometrii samotnou, ale zejména na její důsledky nutné pro prostorovou rekonstrukci a rektifikace obrazu. Více lze nalézt v kapitole 4.

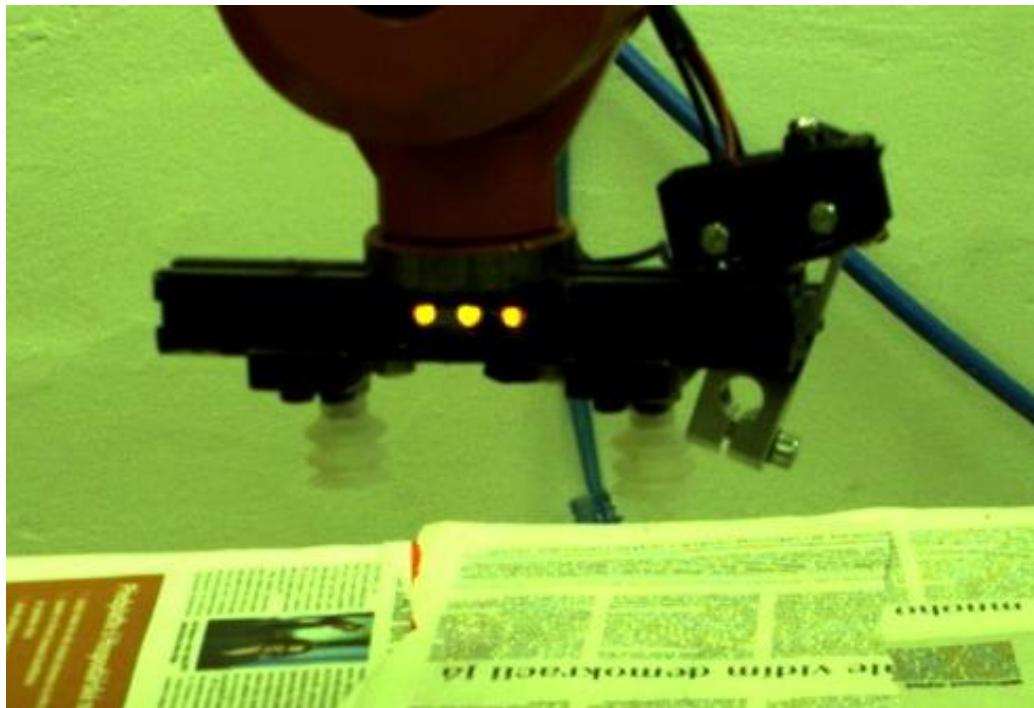
Oblast problematiky hloubkového vnímání je komplexní a stále se rozvíjející. Proto je v kapitole 5 uveden přehled existujících přístupů, jejich vlastností a možnosti použití omezení pro urychlení uvedených metod.

Získané rektifikovaných obrazy jsou vhodným vstupem pro metodu stereo párování a sestavení výsledné mapy disparity. Cílem této práce je implementovat existující metodu založenou na dynamickém programování a zjednodušeném korelováním okolí pixelů. Dále tuto metodu rozšířit o filtr výstupní mapy disparity. Tento filtr bude schopen, na základě speciálních vlastností vstupní dvojice obrazů a mapy disparity, odfiltrovat chyby vzniklé v průběhu výpočtu a doplnit informaci z míst s nízkým, nebo žádným výskytem textury. Součástí práce také bude ověření funkce metody na několika rozdílných typech scény. Další zajímavou možností, kterou se tato práce bude zabývat, je generování počátečního odhadu pro metodu stereo-párování, založeném na detekci pozice a orientace vizuálních značek umístěných ve scéně. Tohoto počátečního odhadu bude použito pro urychlení metody stereo-párování. Více informací lze nalézt v kapitole 6.

Vzhledem k vysokým výpočetním časům metody stereovidění je nezbytné výpočet distribuovat. Tento problém zde bude řešen poněkud netradičním způsobem a to kombinací řízeného jazyku C# na platformě .NET, který řeší komunikaci a neřízeného jazyku C++ na platformě Win32, který zajišťuje výpočet. Podrobný popis navrženého řešení lze nalézt v kapitole 7.

2. Detekce efektoru ramene robotu

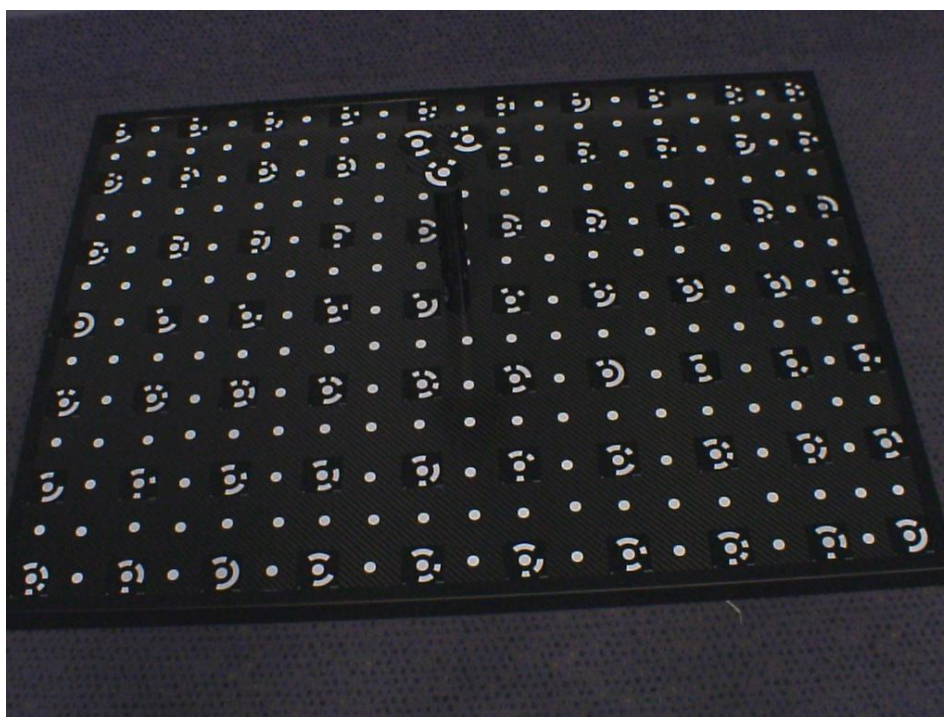
Úloha detekce efektoru ramene robotu v nasnímaném obrazu je v oblasti robotiky velice významnou. Lze jí využít např. pro kalibraci kamer, vizuální zpětnou vazbu, ale nabízejí se i další možnosti využití. Problémem zůstává volba vhodné metody pro samotnou detekci. Je možné využít metod rozpoznávání obrazu, které jsou výpočetně náročné, nebo použití přidaných aktivních či pasivních vizuálních prvků. V této kapitole se budeme zabývat možnostmi detekce polohy efektoru ramene robotu pro kalibraci kamer. Zhodnotíme možnosti použití aktivních i pasivních metod a navrhne vlastní metodu kalibrace kamery, s použitím aktivních vizuálních značek, vhodnou pro použití v oblasti průmyslových ramen robotů. V závěru kapitoly lze nalézt experimentální ověření představené metody spolu se zhodnocením výpočetní náročnosti. Příklad snímku ramene efektoru opatřeného LED naleznete na Obr. 2.1.



Obr. 2.1: Efektor ramene robotu

2.1. Požadavky na kalibraci kamery

Pro kalibraci kamery se běžně využívají kalibrační terčíky. Příklad takového terčíku naleznete na Obr. 2.2. Problémem je ovšem fakt, že tyto terčíky musí být vyrobeny ze speciálních materiálů, aby splňovaly definovanou přesnost ve specifikovaném teplotním rozsahu. Právě z tohoto důvodu jsou velice drahé. Dalším problémem je nutnost speciálního uskladnění. Není možné připustit situaci, kdy by byl kalibrační terčík umístěn přímo na výrobní lince. Mohlo by dojít k jeho zničení či deformaci.



Obr. 2.2: Kalibrační terčík

Vzhledem k faktu, že robotická ramena dosahují velmi vysokých přesností, přibližně desetiny až setiny milimetru, nabízí se správná myšlenka použít přímo robotického ramena pro úlohu kalibraci kamer.

Dnešní moderní řídicí systémy robotů umožňují přímou navigaci v kartézském souřadném systému. Není tedy nutné řešit úlohu inverzní kinematiky robotu a jsme přímo schopni navést efektor robotu do požadované polohy v kartézském souřadném systému, bez nutnosti výpočtu úhlů natočení jeho jednotlivých členů, jelikož tuto funkci provede řídicí systém ramene robotu. Následně musíme vhodným způsobem detekovat polohu jeho efektoru v obrazech ze všech použitých kamer. Výhodou tohoto přístupu je fakt, že jednotlivé kamery budou kalibrovány vzhledem k relativnímu systému souřadnic daného ramena robotu. Získaný tvar vstupních informací následně velice usnadní výpočetní operace v prostoru ramene robotu. Další zajímavou možností je také odměřování vzájemné pozice efektorů více robotů a to v případě, že ramena robotů leží ve společném zorném úhlu jednotlivých kamer.

Shrňme požadavky na parametry nutné pro kalibraci kamery:

1. Trojrozměrná poloha bodu zájmu v prostoru ramene robotu. Poloha musí být vyjádřena v kartézském souřadném systému s počátkem vztaženým vzhledem k ramenu robotu.
2. Dvourozměrná poloha bodu zájmu v diskrétních souřadnicích v obraze nasnímaném danou kamerou.

2.2. Detekce polohy efektoru ramene robotu

Pro detekci efektoru ramene robotu je nutné nalézt vhodný způsob, který musí splňovat následující kritéria:

1. Odolnost vůči chybám: Metoda musí být schopna pracovat za nepříznivých podmínek. Případně pokud selže, musí být toto selhání detekováno.

2. Nízká výpočetní náročnost: Samozřejmě by bylo možné použít metod detekce tvaru efektoru ramene robotu ve scéně. Tento postup by však byl výpočetně náročný a pro velké množství aplikací absolutně nepoužitelný.
3. Odolnost vůči změnám světelných podmínek: Velice důležitý faktor pro nasazení metody. Pokud si např. uvědomíme výrobní linku, pak jiné světelné podmínky jsou při ranní směně a jiné při noční směně se zapnutým umělým osvětlením.

Jak již bylo zmíněno lze pro detekci efektoru použít metod rozpoznávání objektů. Tyto metody jsou však velice pomalé a vykazují značné množství chyb. Právě z tohoto důvodu je vhodné na efektor ramene robotu připevnit umělé navigační prvky. Počet a rozmístění těchto prvků je samozřejmě určen požadavky použité kalibrační metody. S rostoucím počtem prvků ovšem také stoupá odolnost detekčních metod vůči chybám.

Dělení navigačních prvků:

1. Pasivní prvky: Nesou stálou informační hodnotu. Mohou být určeny snadno detekovatelným vzorem, snižují tedy nároky na aplikační logiku. Ovšem mají své nevýhody a to pokud jsou umístěny v prostředí s nekontrolovanými podmínkami, kde dochází k výkyvům intenzity osvětlení, odlesků, nebo překrytí jinými objekty (occlusion objekty).
2. Aktivní prvky: Řeší problém nepříznivých podmínek ve snímané scéně. Přinášejí ovšem větší náročnost na zpracování, vlastní realizaci a aplikační logiku. Svým působením mohou v některých případech negativně ovlivnit snímanou scénu.

2.3. Vlastní realizace

Pro realizaci vhodné metody, která by splňovala výše zmíněná kritéria, bylo nutné připevnit na efektor ramene robotu speciální přípravek se světelnými diodami. Tento přípravek

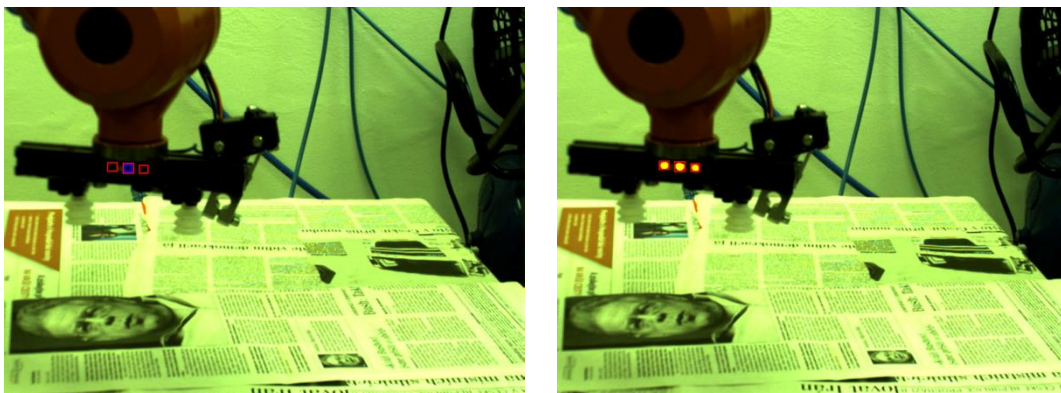
obsahuje tři diody, které jsou propojené s řídicím systémem robotu. Detekce polohy přípravku pak probíhá následujícím způsobem:

1. Navedení ramene robotu na pozici: Robot je naveden na souřadnici v kartézském souřadném systému (relativním vůči danému robotu). Systém čeká na ustálení v dané poloze.
2. Rozsvícení LED: Po ustálení v definované poloze dojde k rozsvícení LED umístěných na přípravku.
3. Uložení obrazové informace: Následně je uložena obrazová informace ze všech kamer vztažených k danému robotu.
4. Zhasnutí LED: Po skončení procesu sejmutí obrazové informace jsou LED vypnuty.
5. Opětovné uložení obrazové informace: Následně je znovu uložena obrazová informace ze všech kamer vztažených k danému robotu.



Obr. 2.3: Nasnímaná scéna s vypnutým a zapnutým LED indikátorem

Postup operací naleznete na Obr. 2.3. Díky jednotlivým krokům dostáváme z každé kamery obraz s rozsvícenými a zhasnutými LED a jsme tedy schopni získat rozdílovou informaci, která umožní detekci polohy pixelů jednotlivých LED.



Obr. 2.4: Detekované pozice LED indikátoru

Pro spolehlivou detekci poloh jednotlivých LED ovšem potřebujeme následující postup:

1. Určení rozdílové informace obou obrazů: Provedením operace rozdílu obrazů s rozsvícenými a zhasnutými LED získáváme přímo pixely LED. Bohužel v reálném prostředí dochází ke vzniku vibrací, šumu a odleskům, které způsobují poruchy.
2. Prahování podle použité barvy LED: Jedná se o první krok k odstranění chybných informací v důsledku poruch zmíněných v prvním bodě.
3. Provedení operací uzavření a otevření: Tato operace usnadní následující operaci segmentace. Dojde k odstranění pixelů vzniklých v důsledku šumu.
4. Detekce hran. Zjednoduší výpočetní náročnost následující operace segmentace.
5. Segmentace jednotlivých LED elementů: Segmentace kontinuálních elementů.
6. Eliminace chybných objektů: Probíhá podle velikosti segmentovaných objektů, využívá se předpokladu, že LED jsou stejně velké. V průběhu operace jsou vyřazeny objekty, jejichž obsah je nižší, nebo vyšší, nežli definované rozmezí.
7. Test počtu detekovaných LED: Na efektoru ramene robotu je připevněn konečný počet elementů. Libovolná odchylka tohoto počtu signalizuje selhání metody.
8. Test geometrického uspořádání LED: Kontroluje, zda elementy splňují definované geometrické uspořádání a zachovávají vůči sobě poměrnou vzdálenost. V našem případě musí jednotlivé elementy ležet na úsečce a mezi sebou mít poměrnou

vzdálenost 1:1:2, vyplývající z konstrukce přípravku umístěného na efektoru ramene robotu.

Po provedení jednotlivých kroků již dostáváme informaci o pozici jednotlivých LED (viz Obr. 2.4).

2.4. Experimentální výsledky

Výpočetní náročnost procesu detekce LED na efektoru ramene robotu je velice nízká. Tabulku časů výpočtu v závislosti na rozlišení snímků naleznete v Tab. 2.1.

Rozlišení snímku [pixel x pixel]	Doba výpočtu [ms]
320x240	114
640x480	525
800x600	894
1024x768	2213
1280x960	3125

Tab. 2.1: Doba analýzy jednoho snímku v závislosti na jeho rozlišení

Vzhledem k automatizaci celého procesu snímání bodů pro kalibraci zvolených kamer je časová náročnost přímo úměrná počtu snímaných bodů, jak je patrné z následující tabulky. Fixní částí celé doby snímání je počáteční nastavení ramene robotu. Dlouhé časy jsou způsobeny přemísťováním ramene robotu do požadované polohy. Celkovou dobu trvání snímání jednotlivých bodů pro kalibraci kamery je možné nalézt v Tab. 2.2. Jedná se o čas snímání trojrozměrné polohy efektoru ramene robotu a diskrétní dvourozměrné polohy efektoru ve snímaném obraze kalibrované kamery.

Počet bodů pro kalibraci	Doba zpracování [s]
8	120
25	360
100	1100
268	2820

Tab. 2.2: Časy snímání dvojic trojrozměrných a dvourozměrných bodů pro kalibraci

3. Model kamery a kalibrace

Kalibrace kamery je podstatným krokem pro jakékoliv další zpracování obrazu. V posledních letech bylo publikováno mnoho studií týkajících se metod kalibrace. Přesto je však stále obtížné nalézt vhodnou metodu pro řešení daného problému a nelézt porovnání této metody vůči ostatním metodám. Tento problém vychází z nedostatku standardizovaných konvencí a existence mnoha rozdílných metod. Tato kapitola proto prezentuje přehled kalibračních metod vhodných pro použití v oblasti robotiky s hlavním záměrem na jejich experimentálním ověření na konkrétní robotické úloze v reálném prostředí. V závěru kapitoly lze nalézt srovnání rychlosti a přesnosti uvedených metod kalibrace.

3.1. Úvod

Kalibrace kamery je prvním krokem k počítačovému vidění. Přestože některé metody stereo vidění používají informace z nekalibrovaných kamer [18], je kalibrace důležitým krokem ve všech oblastech vyžadujících metrickou informaci. Použití precizně kalibrovaných kamer umožňuje měření vzdáleností v reálné scéně na základě projekcí z jednotlivých kamer na dvojrozměrnou plochu jejich CCD [19] a [20]. Několik aplikací těchto metod je:

1. **Hloubková rekonstrukce:** Každý bod v obrázku definuje optický paprsek procházející jednak tímto bodem a dále ohniskovým bodem kamery ve směru do scény. Použití více jak jednoho pohledu na nehybnou scénu (pořízeného např. pomocí stereoskopického systému, jedné přesně popsané pohybující se kamery, nebo strukturovaného zdroje světla) umožňuje nalézt průsečík zmíněných optických parsků a provést tak výpočet trojrozměrné pozice v metrickém souřadném systému [21], [22] a [23].
2. **Vizuální inspekce:** Po provedení hloubkové rekonstrukce lze rekonstruovaný model porovnat s existujícím (v systému uloženým) modelem. Toto se provádí např. z důvodů

detekování nedokonalostí při výrobě, jako jsou prohnutí, rýhy a jiná poškození v oblasti kontroly kvality. Počítačem řízená inspekce umožňuje automatickou kontrolu a inspekci kvality na rozdíl od subjektivního, nepřesného a pomalého pohledu člověka. Navíc je odstraněn faktor subjektivního pohledu více lidí na více směnných provozech. Tento způsob řešení často vede na použití statistických metod [24].

3. **Lokalizace objektů:** Nalezená hloubková informace usnadňuje odlišení jednotlivých objektů ve scéně. Následně je pak možné snadno relativní pozice mezi těmito objekty. Tato možnost přináší mnoho oblastí použití, jako je např. aplikace automatické montáže na výrobních linkách, detekce překážek v pracovním prostoru robotu a navigace robotu na základě takto získaných dat [25], [26] a [27].
4. **Lokalizace kamery:** Pokud je kamera připojena na rameno robotu, nebo na platformu mobilního robotu, pak je možné pozici a orientaci kamery vypočítat na základě lokalizace známých značek ve scéně. Pokud jsou takto získaná data ukládána a průběžně analyzována, je možné mapovat okolí robota a vyhledat v něm aktuální pozici i orientaci celého robotu. Této informace lze s výhodou využít v kontrole, návrhu, optimalizaci a plánování pohybu robotu [28], [29] a [30].

Kalibrace kamery je rozdělena do dvou kroků. Prvním z těchto kroků je vytvoření modelu kamery. Tento krok jde ruku v ruce s nalezením matematického modelu fyzického i optického chování senzorů a jeho určení pomocí množiny parametrů. Druhý krok vychází z použití iteračních metod sloužících k estimaci hodnot těchto parametrů. Rozlišujeme dva druhy parametrů, které musí být zvažovány:

1. **Vnitřní parametry:** Modelují vnitřní geometrii a optické charakteristiky obrazového senzoru. Tyto parametry určují způsob, jakým je světlo promítáno čočkou na plochu obrazového senzoru.

2. **Vnější parametry:** Určují polohu a orientaci kamery s ohledem na souřadný systém. Také poskytují metrické informace s ohledem na uživatelsky definovaný lokální souřadný systém na místo souřadného systému kamery.

Technika kalibrace kamery může být klasifikována ve vztahu k několika rozdílným kritériím:

1. Lineární vs. nelineární kalibrace kamery (často rozdělena s ohledem na modelování distorze objektivu) [31].
2. Vnitřní vs. vnější kalibrace kamery. Vnitřní kalibrace kamery se týká získávání fyzických a optických parametrů kamery [32], vnější kalibrace kamery se pak zaměřuje na měření pozice a orientace kamery ve scéně [33] a [34].
3. Implicitní [35] versus explicitní [36] kalibrace kamery. Implicitní kalibrace je proces kalibrace kamery bez explicitního výpočtu fyzických parametrů. Přestože výsledky mohou být použity pro trojrozměrné měření a výpočet souřadnic v obraze, je tento postup je nepoužitelný pro modelování kamery, protože získané parametry nekorrespondují s fyzickými parametry [37].

Následující rozdílné přístupy mohou být používány jednotlivými kalibračními metodami k estimaci parametrů modelu kamery.

1. **Nelineární optimalizační techniky:** Kalibrační techniky se stávají nelineárními, pokud je jakýkoliv druh vad objektivů součástí modelu kamery. V tomto případě jsou parametry kamery získávány iterativním procesem, který minimalizuje stanovenou hodnotící funkci. Minimalizovaná funkce je ve většině případů definována jako vzdálenost mezi body v obraze a modelované projekce daného bodu získané během iterování. Výhodou iterativního přístupu je, že téměř každý model může být tímto způsobem kalibrován. Přesnost určení parametrů stoupá spolu se zvětšujícím se počtem iterací a řešení konverguje k reálným hodnotám parametrů. Přes veškeré

výhody této techniky, je zde však požadavek na velice dobrý počáteční odhad, který garantuje konvergenci k řešení [42].

2. **Lineární techniky:** Vypočítávají přímo transformační matici. Tyto techniky využívají metody nejmenších čtverců k získání transformační matice, která vyjadřuje relaci mezi hodnotami trojrozměrné pozice bodu v prostoru scény s jejich dvourozměrnými pozicemi projekce kamery. Výhodou tohoto postupu, je jednoduchost matematického modelu. Stinnou stránkou je ovšem nemožnost vypořádání se s poruchami objektivů a modelování a zohlednění těchto poruch. Tento problém samozřejmě silně snižuje přesnost celého systému. Někdy je také velice obtížné extrahovat parametry z matice, protože je použito metod implicitní kalibrace. Některé reference týkající lineární kalibrace je možné nalézt v pracích [35], [38] a [31].
3. **Technika rozdělení do dvou kroků:** Tyto techniky používají metod lineární optimalizace k výpočtu některých parametrů a v druhém kroku jsou zbývající parametry vypočítány iterativně. Tato technika umožňuje rychlou kalibraci, dobrý počáteční odhad, snížení počtu iterací a možnost modelování poruch objektivů. Konvergence je totiž zaručena v prvním kroku metody. Tento typ metod spojuje výhody všech zmíněných metod. Více lze nalézt v pracích [39], [43] a [44].

Tato kapitola slouží jako popis některých nejvíce používaných metod kalibrace. První metodu kalibrace kamery popsal Hall v roce 1982 a jednalo se o implicitní lineární metodu kalibrace kamer. Dochází k výpočtu matice rozměru 3×4 , kde tato matice vyjadřuje vztah mezi trojrozměrnými pozicemi bodů ve scéně a jejich dvourozměrnými projekcemi do plochy kamery.

Následující práce Faugerase zveřejněná v roce 1986 se zabývala extrakcí fyzických parametrů kamery ze zmíněné transformační techniky. Další metody již byly založeny na nelineárních metodách explicitní kalibrace kamery, obsahující model chování poruchy objektivu. První

z nich je jednoduchým rozšířením Faugerasovy lineární metody o radiální zkreslení. Nejvíce používanou metodou je metoda Tsai založena na dvou fázové technice, modelující pouze radiální zkreslení [39]. Kompletní model kamery byl představen Wengem v roce 1992. Jeho model zahrnoval tři rozdílné typy zkreslení objektivu [43].

Nezapomeňme, že jedním ze základních problémů detailního pochopení kalibračních techniky je nedostatek standardizací matematických výrazů a používání rozdílných souřadných systémů. Oba problémy pak způsobují velké obtíže v porovnávání vhodnosti metod pro specifické účely a využití. Právě z tohoto důvodu je část této práce věnována popisu metod a jejich prezentaci s jednotným zápisem. Veškeré zmíněné techniky jsou zde popsány spolu s uvedením jejich dvourozměrné a trojrozměrné přesností. Lze zde nalézt také krátké pojednání o metodách výpočtu přesnosti metod kalibrace se zaměřením na jednotné porovnání možností rozdílných kalibračních metod.

Tato kapitola je rozdělena následovně. První část obsahuje informace o modelování kamery a názorném postupu, jak je model kamery postupně složen ze sekvence geometrickým transformací. Následující sekce popisuje rozdílné techniky kalibrace kamery, které slouží k estimaci parametrů modelu kamery. Na závěr je uvedeno několik metod výpočtu přesnosti a závěrečné zhodnocení.

3.2. Model kamery

Model kamery je matematickou formalizací, která aproximuje chování libovolného fyzického zařízení pro snímání obrazu, použitím souboru matematických operací. Proces modelování kamery je založen na aproximaci vnitřní geometrie kamery spolu s její pozicí a orientací ve scéně. Existuje několik typů modelů kamery, lišících se podle požadované přesnosti a zohlednění vad objektivu. Nejjednodušší model kamery je založen na lineárních transformacích, který ale nemodeluje zkreslení. Na druhé straně jsou nelineární modely,

které dostatečně přesně simulují zkreslení objektivu. Tyto modely jsou výhodné zejména v aplikacích, kde je nutná vyšší přesnost výpočtu. Jejich výpočet je ovšem náročnější.

Nejjednodušší model kamery je již zmíněný model, který prezentoval Hall [35]. Úkolem je nalezení lineárního vztahu mezi 3D body ve scéně a jejich 2D obrazy na projekční ploše CCD v kameře. Tento vztah je vyjádřen jako transformační matice (3.1):

$$\begin{pmatrix} s^I X_d \\ s^I Y_d \\ s \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \end{pmatrix} \begin{pmatrix} {}^w X_w \\ {}^w Y_w \\ {}^w Z_w \\ 1 \end{pmatrix} \quad (3.1)$$

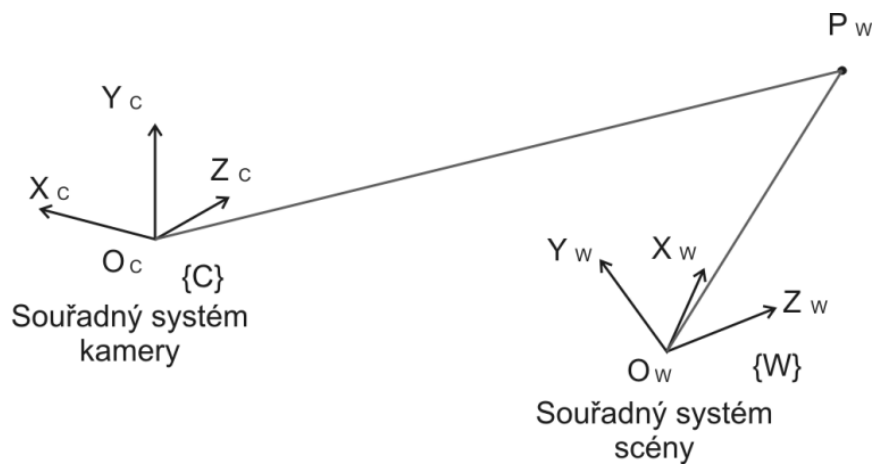
Pak trojrozměrný bod ve scéně P_w popsáný pomocí metrického souřadného systému (tedy ${}^w P_w$) je možné transformovat prostřednictvím Hallovy transformační matice do dvojrozměrného bodu P_d v pixelech (s ohledem na souřadný systém kamery), viz (3.2).

$${}^I P_d = ({}^I X_d, {}^I Y_d) \quad (3.2)$$

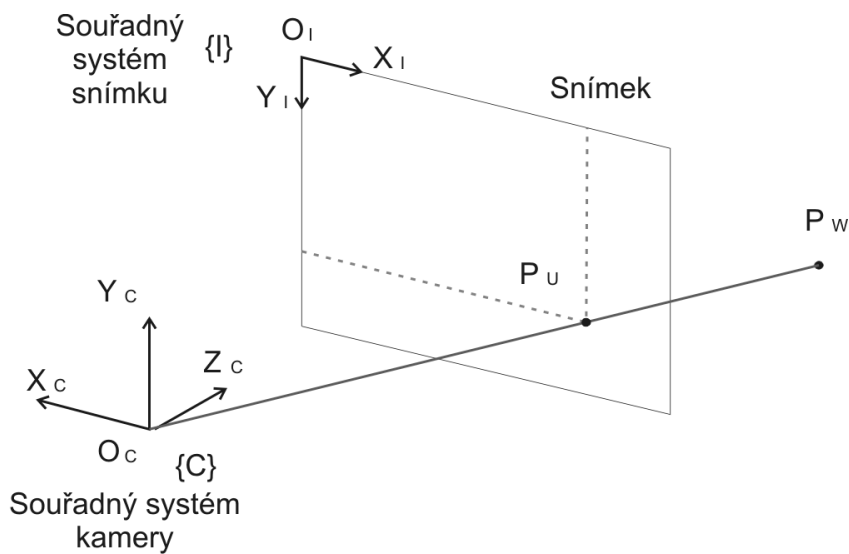
Modelování kamery je běžně rozděleno do následujících čtyř kroků:

1. Prvním krokem je převod bodu ${}^w P_w$ ze souřadného systému scény do souřadného systému kamery ${}^c P_w$.
2. Dalším krokem je transformace bodu ${}^c P_w$ na projekční plochu kamery a získání bodu ${}^c P_u$ prostřednictvím projektivní transformace.
3. Třetím krokem je modelování vad objektivu. Bod ${}^c P_u$ je pak transformován do reálné projekce ${}^c P_d$ (který může kolidovat s bodem získaným kamerou).

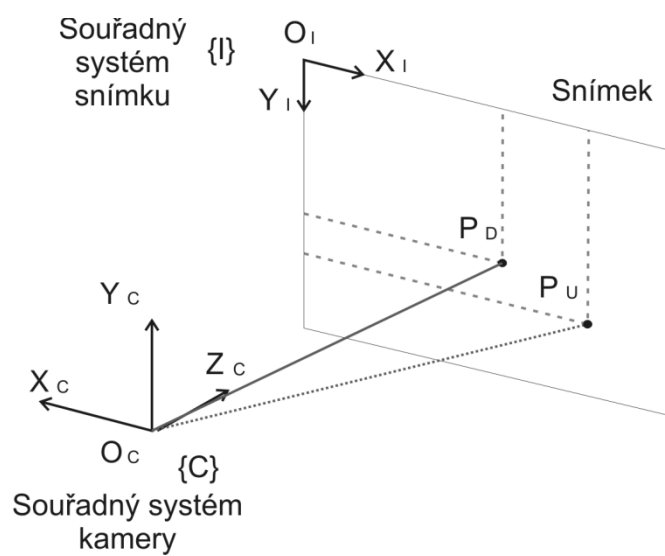
4. Poslední krokem je transformace z metrického souřadného systému do diskrétního souřadného systému kamery, které specifikuje polohu jednotlivých pixelů a získáváme tak souřadnice ${}^I P_d$.



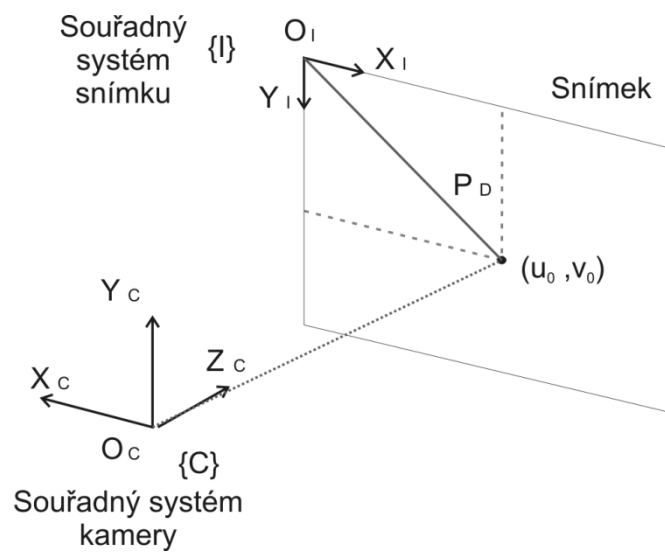
Obr. 3.1: Modelování kamery krok č. 1



Obr. 3.2: Modelování kamery krok č. 2



Obr. 3.3: Modelování kamery krok č. 3



Obr. 3.4: Modelování kamery krok č. 4

3.2.1. Pozice a orientace kamery

Transformace souřadného systému scény do souřadného systému kamery je ve všech případech použitých modelů kamer stejná. Transformace je modelována za použití vektoru posunu a matice rotace, jak je ukázáno v následující rovnici (3.3).

$$\begin{pmatrix} {}^cX_w \\ {}^cY_w \\ {}^cZ_w \end{pmatrix} = {}^cR_w \begin{pmatrix} {}^wX_w \\ {}^wY_w \\ {}^wZ_w \end{pmatrix} + {}^cT_w \quad (3.3)$$

Vezměme bod wP_w vztahený k souřadnému systému scény a aplikuje rovnici (3.2). Pak získáváme bod cP_w v souřadném systému kamery. Je nutné upozornit, že matice cR_w popisuje orientaci souřadného systému scény $\{W\}$ vůči ose souřadného systému kamery $\{C\}$ a tedy cT_w vyjadřuje pozici počátku souřadného systému scény vzhledem k $\{C\}$.

3.2.2. Perspektivní projekce

Uvažme, že optický senzor může být modelován jako štěrbinová kamera [19]. Pak můžeme říci, že plocha obrazu umístěná ve vzdálenosti f od optického centra kamery O_c je paralelní s rovinou definovanou osami souřadného systému X_c a Y_c . Navíc uvažme bod objektu cP_w vztaheného k souřadnému systému kamery. Pokud je tento bod promítnut přes ohniskový bod O_c , pak optický paprsek protíná plochu obrazu ve 2D souřadnicích obrazového bodu cP_u . Tento vztah vyjadřuje následující rovnice (3.4):

$${}^cX_u = f \frac{{}^cX_w}{{}^cZ_w}, \quad {}^cY_u = f \frac{{}^cY_w}{{}^cZ_w} \quad (3.4)$$

Veškeré dále zmíněné modely kamery řeší tento problém perspektivní transformace stejným způsobem.

3.2.3. Poruchy objektivu

Třetí krok je založený na modelování zkreslení objektivu. Následují rovnice (3.5) přibližují způsob výpočtu a to transformaci polohy nezkresleného bodu cP_u (tedy polohy bodu promítnutého pomocí ideální kamery bez jakýchkoliv vad) do pozice zkresleného bodu cP_d . Parametry δ_x a δ_y reprezentují dané zkreslení.

$${}^cX_u = {}^cX_d + \delta_x, \quad {}^cY_u = {}^cY_d + \delta_y \quad (3.5)$$

Model kamery Faugeras-Toscani nezahrnuje zkreslení objektivu, proto jsou body cP_u a cP_d totožné. V tomto případě jsou i koeficienty δ_x a δ_y (3.6).

$$\delta_x = 0, \quad \delta_y = 0 \quad (3.6)$$

Zmíněný model však může být vylepšen modelováním radiálního zkreslení. Model kamery, který představil Tsai [39], uvažuje pouze radiální zkreslení, viz (3.7). Tento typ zkreslení je z velké míry způsoben radiálním zakřivením čočky. Schéma lze nalézt na Obr 3.5.

$$\delta_x = \delta_{xr}, \quad \delta_y = \delta_{yr} \quad (3.7)$$

Porucha způsobená radiálním zkreslením může být modelována pomocí výrazu (3.9), kde k_1 reprezentuje jako první člen rozvoje pro vyjádření radiálního zkreslení. Bylo dokázáno, že první člen řady k_1 je dostačující k modelování radiálního zkreslení ve většině možných aplikací [39].

$$\delta_{xr} = k_1 {}^cX_d ({}^cX_d^2 + {}^cY_d^2), \quad \delta_{yr} = k_1 {}^cY_d ({}^cX_d^2 + {}^cY_d^2) \quad (3.8)$$

Wengův model uvažuje následující tři typy zkreslení: radiální zkreslení, mimostředné zkreslení a hranolové rozptýlení. Celkové zkreslení je pak vyjádřeno jako suma všech tří jednotlivých zkreslení (3.9).

$$\delta_x = \delta_{xr} + \delta_{xd} + \delta_{xp}, \quad \delta_y = \delta_{yr} + \delta_{yd} + \delta_{yp} \quad (3.9)$$

Weng ovšem předložil model distorze objektivu na základě polohy nezkresleného bodu ${}^cP_u = ({}^cX_u, {}^cY_u)$, na místo zkresleného bodu ${}^cP_d = ({}^cX_d, {}^cY_d)$. Přestože oba přístupy přicházejí v úvahu, musí být tento fakt zohledněn v procesu kalibrace, protože kalibrační parametry jsou rozdílné. V tomto případě jsou rovnice (3.2) substituovány následujícím výrazem (3.10):

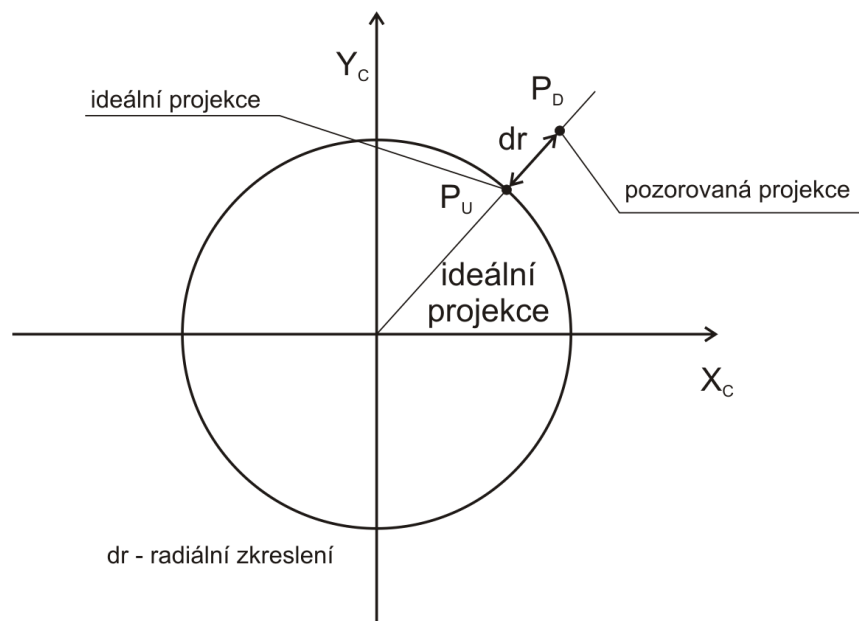
$${}^cX_d = {}^cX_u + \delta_x, \quad {}^cY_d = {}^cY_u + \delta_y \quad (3.10)$$

Radiální distorze je modelována, stejným způsobem jako je způsob popsáný Tsaiem, ovšem Weng uvažuje polohy nezkreslených bodů.

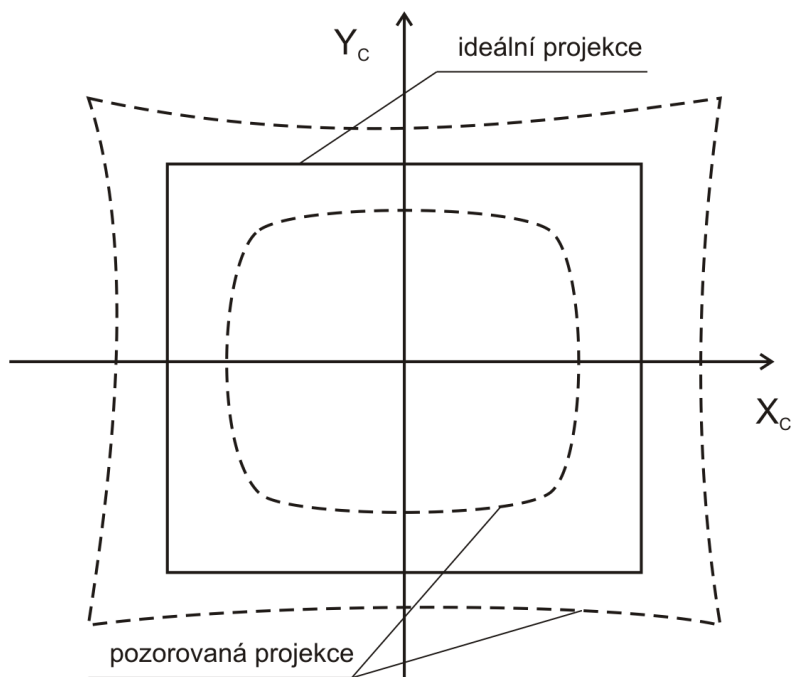
$$\delta_{xr} = k_1 {}^cX_u ({}^cX_u^2 + {}^cY_u^2), \quad \delta_{yr} = k_1 {}^cY_u ({}^cX_u^2 + {}^cY_u^2) \quad (3.11)$$

Mimostředné zkreslení vzniká z důvodu, že optické centrum čoček není vhodně vyrovnáno s centrem CCD čipu kamery. Tento typ zkreslení přináší radiální a tangenciální zkreslení, které je možné pospat pomocí následujících rovnic (3.12).

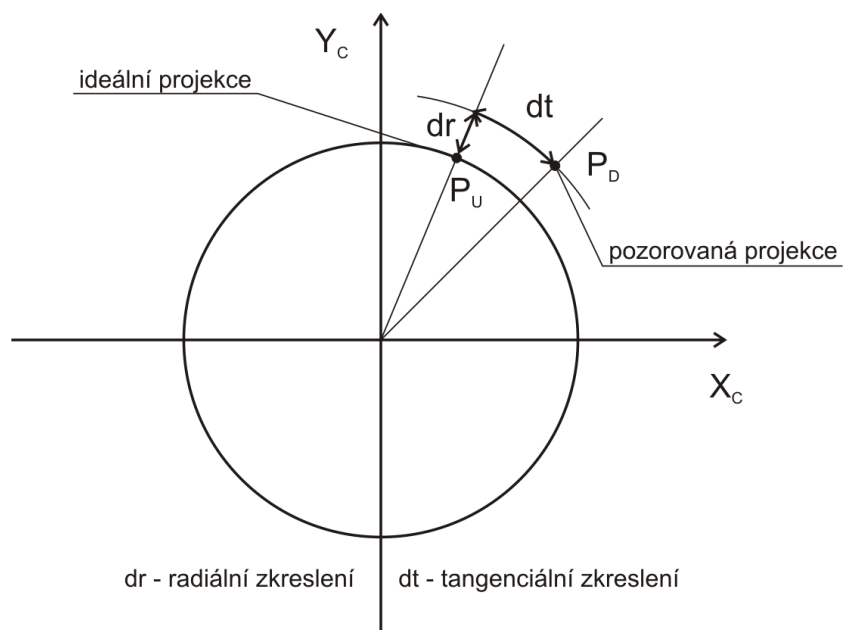
$$\begin{aligned} \delta_{xd} &= p_1 (3 {}^cX_u^2 + {}^cY_u^2) + 2p_2 {}^cX_u {}^cY_u \\ \delta_{yd} &= 2p_1 {}^cX_u {}^cY_u + p_2 ({}^cX_u^2 + 3 {}^cY_u^2) \end{aligned} \quad (3.12)$$



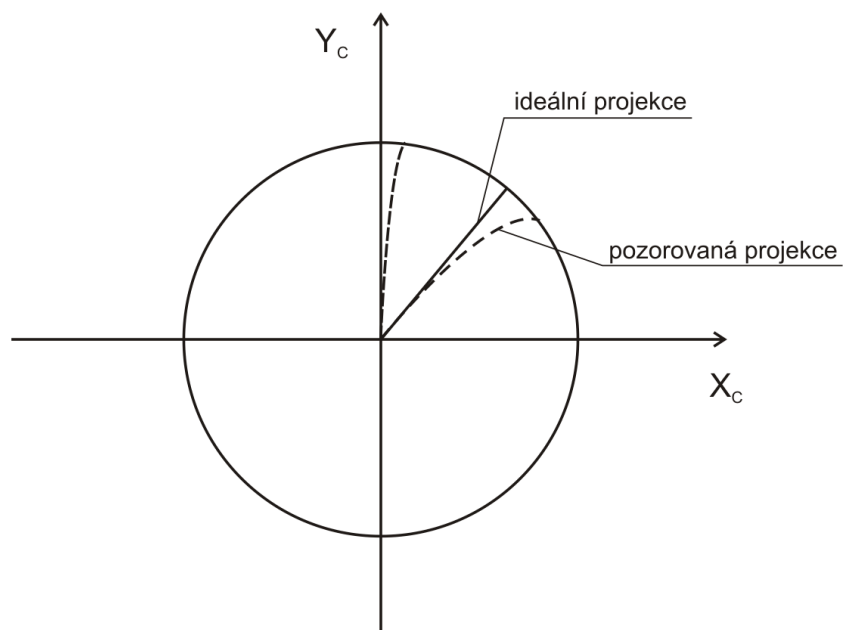
Obr. 3.5: Radiální zkreslení



Obr. 3.6: Ukázka efektu radiálního zkreslení



Obr. 3.7: Tangenciální zkreslení



Obr. 3.8: Ukázka efektu tangenciálního zkreslení

Hranolové zkreslení vzniká v důsledku poruch objektivu a výroby stejně jako sestavení kamery. Tento typ zkreslení může být modelován přidáním tenkého hranolu do optického systému, způsobujícího radiální a tangenciální zakřivení. Toto zakřivení lze modelovat následovně (3.13):

$$\delta_{xp} = s_1({}^cX_u^2 + {}^cY_u^2), \quad \delta_{yp} = s_2({}^cX_u^2 + {}^cY_u^2) \quad (3.13)$$

Využitím rovnic (3.9), (3.13), (3.15) a použitím následujících substitucí:

$$\begin{aligned} g_1 &= s_1 + p_1, \\ g_2 &= s_2 + p_2, \\ g_3 &= 2p_1, \\ g_4 &= 2p_2 \end{aligned} \quad (3.14)$$

získáváme následující rovnice (3.15) reprezentující celkové zkreslení kamery:

$$\begin{aligned} \delta_x &= (g_1 + g_3) {}^cX_u^2 + g_4 {}^cX_u {}^cY_u + g_1 {}^cY_u^2 + k_1 {}^cX_u ({}^cX_u^2 + {}^cY_u^2) \\ \delta_y &= g_2 {}^cX_u^2 + g_3 {}^cX_u {}^cY_u + (g_2 + g_4) {}^cY_u^2 + k_1 {}^cY_u ({}^cX_u^2 + {}^cY_u^2) \end{aligned} \quad (3.15)$$

3.2.4. Obraz v počítači

Konečným krokem je vyjádření polohy bodu cP_d vzhledem k počítačovému obrázku v pixelech $\{I\}$. Tato transformace souřadných systémů může být provedena dvěma rozdílnými způsoby v závislosti na použitém modelu kamery.

Model kamery Faugeras-Toscani, Faugeras-Toscani se zkreslením a model podle Wenga používá pro tuto transformaci následující rovnice:

$$\begin{aligned} {}^I X_d &= -k_u {}^cX_d + u_0 \\ {}^I Y_d &= -k_v {}^cY_d + v_0 \end{aligned} \quad (3.16)$$

Kde (k_u, k_v) jsou parametry, které transformují měřítko jednotlivých bodů z metrického souřadného systému kamery do diskrétních souřadnic počítačového obrázku (v pixelech).

Parametry (u_0, v_0) určují posunutí obou souřadných systémů projekci ohniskového bodu do plochy obrazu v pixelech, tedy význačný bod. Jeho pozice je použita k definování posunutí mezi souřadnými systémy.

Tsaiho model kamery zavádí jiné rovnice, které provádějí stejnou transformaci. Tyto rovnice jsou následující:

$$\begin{aligned} {}^I X_d &= -s_x d'_x{}^{-1} {}^C X_d + u_0 \\ {}^I Y_d &= -d_y{}^{-1} {}^C Y_d + v_0 \end{aligned} \quad (3.17)$$

Kde (u_0, v_0) jsou komponenty význačného bodu v pixelech, s_x je měřítko obrazu.

$$d'_x = d_x \frac{N_{cx}}{N_{fx}} \quad (3.18)$$

Kde d_x je vzdálenost středů navazujících sensorových prvků ve směru osy x, d_y je vzdálenost středů navazujících sensorových prvků ve směru osy y, N_{cx} je počet sensorových prvků ve směru osy x a N_{fx} je počet pixelů v obrazové řádce vzorkované počítačem.

3.3. Kalibrační metody

Kalibrační metody závisí na matematickém modelu, který byl použit k aproximaci chování kamery. Lineární modely, jako je např. Hallův, nebo Faugeras-Toscani používají metodu nejmenších čtverců k získání parametrů modelu. Nelineární metody kalibrace jako je Faugeras-Toscani se zohledněním zkreslení, Tsai a Weng používají techniku skládající se ze dvou kroků. V prvním kroku je určena lineární aproximace s ohledem na získání počátečního odhadu a dále je ve druhém kroku spuštěn iterativní algoritmus, který je využit k optimalizaci parametrů. V této části jsou jednotlivé kalibrační metody vysvětleny, uvedeny jejich rovnice a popsán kalibrační algoritmus.

3.4.1. Hallova metoda

Metoda kalibrace založené na Hallově modelu vychází z následujícího vzorce:

$$\begin{aligned} {}^I X_u &= \frac{A_{11} {}^W X_w + A_{12} {}^W Y_w + A_{13} {}^W Z_w + A_{14}}{A_{31} {}^W X_w + A_{32} {}^W Y_w + A_{33} {}^W Z_w + A_{34}} \\ {}^I Y_u &= \frac{A_{21} {}^W X_w + A_{22} {}^W Y_w + A_{23} {}^W Z_w + A_{24}}{A_{31} {}^W X_w + A_{32} {}^W Y_w + A_{33} {}^W Z_w + A_{34}} \end{aligned} \quad (3.19)$$

Úpravou proměnných, je možné získat následující výraz:

$$\begin{aligned} 0 &= A_{11} {}^W X_w - A_{31} {}^I X_u {}^W X_w + A_{12} {}^W Y_w - A_{32} {}^I X_u {}^W Y_w \\ &\quad + A_{13} {}^W Z_w - A_{33} {}^I Y_u {}^W Z_w + A_{14} - A_{34} {}^I Y_u \\ 0 &= A_{21} {}^W X_w - A_{31} {}^I Y_u {}^W X_w + A_{22} {}^W Y_w - A_{32} {}^I Y_u {}^W Y_w \\ &\quad + A_{23} {}^W Z_w - A_{33} {}^I Y_u {}^W Z_w + A_{24} - A_{34} {}^I Y_u \end{aligned} \quad (3.20)$$

Neznámé A_{ij} jsou vyjádřeny vektorem A o dvanácti parametrech. Získáváme následující výraz:

$$QA = 0 \quad (3.21)$$

Kde A je vektor o 12 neznámých z předchozí rovnice, Q je matice $2n \times 12$, kde n je počet párů bodů, použitých ke kalibraci kamery. Pár bodů je vytvořen z trojrozměrného bodu vyjádřeným v souřadném systému scény $\{W\}$ a jeho dvourozměrné projekce na snímací plochu kamery vyjádřenou v pixelech v souřadném systému $\{I\}$.

$$A = (A_{11} \ A_{12} \ A_{13} \ A_{14} \ A_{21} \ A_{22} \ A_{23} \ A_{24} \ A_{31} \ A_{32} \ A_{33} \ A_{34})^T \quad (3.22)$$

Každý takovýto pár bodů přidává do matice Q následující dvě řádky:

$$Q_{2i-1} = \begin{pmatrix} {}^wX_{ui} \\ {}^wY_{ui} \\ {}^wZ_{ui} \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ - {}^I X_{ui} {}^wX_{wi} \\ - {}^I X_{ui} {}^wY_{wi} \\ - {}^I X_{ui} {}^wZ_{wi} \\ - {}^I X_{ui} \end{pmatrix}^T, \quad Q_{2i} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ {}^wX_{ui} \\ {}^wY_{ui} \\ {}^wZ_{ui} \\ 1 \\ - {}^I Y_{ui} {}^wX_{wi} \\ - {}^I Y_{ui} {}^wY_{wi} \\ - {}^I Y_{ui} {}^wZ_{wi} \\ - {}^I Y_{ui} \end{pmatrix}^T \quad (3.23)$$

Uvažujme dále, že trojrozměrné polohy množiny n kalibračních bodů a jejich korespondujících dvourozměrných projekcí v obraze je známa (n musí být větší nebo rovno 6). Dále uvažujme beze ztráty obecnosti, že parametr $A_{34} = 1$. Všechny elementy vektoru A mohou být určeny použitím lineární metody nejmenších čtverců, tedy dostáváme:

$$Q' A' = B' \quad (3.24)$$

$$A' = (A_{11} \ A_{12} \ A_{13} \ A_{14} \ A_{21} \ A_{22} \ A_{23} \ A_{24} \ A_{31} \ A_{32} \ A_{33})^T \quad (3.25)$$

$$Q'_{2i-1} = \begin{pmatrix} {}^wX_{ui} \\ {}^wY_{ui} \\ {}^wZ_{ui} \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ - {}^I X_{ui} {}^wX_{wi} \\ - {}^I X_{ui} {}^wY_{wi} \\ - {}^I X_{ui} {}^wZ_{wi} \end{pmatrix}^T, \quad Q'_{2i} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ {}^wX_{ui} \\ {}^wY_{ui} \\ {}^wZ_{ui} \\ 1 \\ - {}^I Y_{ui} {}^wX_{wi} \\ - {}^I Y_{ui} {}^wY_{wi} \\ - {}^I Y_{ui} {}^wZ_{wi} \end{pmatrix}^T \quad (3.26)$$

$$B'_{2i-1} = {}^I X_{ui}, \quad B'_{2i} = {}^I Y_{ui} \quad (3.27)$$

Následně je vektor neznámých A vypočítán pomocí aplikace výpočtu pseudoinverzní matice ukázané v následující rovnici:

$$A' = (Q'^T Q')^{-1} Q'^T B' \quad (3.28)$$

3.4.2. Faugerasova metoda

Z důvodu potřeby kalibrovat kompletní model kamery předložené Faugerasem a Toscanim je nutné kombinovat již zmíněné rovnice (3.3), (3.5), (3.6), (3.16) a získat tak následující vztah:

$$\begin{aligned} {}^I X_u &= -k_u f \frac{r_{11} {}^W X_w + r_{12} {}^W Y_w + r_{13} {}^W Z_w + t_x}{r_{31} {}^W X_w + r_{32} {}^W Y_w + r_{33} {}^W Z_w + t_z} + u_0 \\ {}^I Y_u &= -k_v f \frac{r_{21} {}^W X_w + r_{22} {}^W Y_w + r_{23} {}^W Z_w + t_y}{r_{31} {}^W X_w + r_{32} {}^W Y_w + r_{33} {}^W Z_w + t_z} + v_0 \end{aligned} \quad (3.29)$$

Nezapomeňme, že zmíněné rovnice mohou být vyjádřeny v metrické formě, tedy následujícím způsobem:

$$\begin{pmatrix} s^I X_d \\ s^I Y_d \\ s \end{pmatrix} = \begin{pmatrix} \alpha_u & 0 & u_o & 0 \\ 0 & \alpha_v & v_o & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} {}^W X_w \\ {}^W Y_w \\ {}^W Z_w \\ 1 \end{pmatrix}, \quad (3.30)$$

kde $\alpha_u = -fk_u$ a $\alpha_v = -fk_v$. Transformační matici A lze pak přepsat následujícím způsobem:

$$\begin{pmatrix} s^I X_d \\ s^I Y_d \\ s \end{pmatrix} = A \begin{pmatrix} {}^W X_w \\ {}^W Y_w \\ {}^W Z_w \\ 1 \end{pmatrix} \quad (3.31)$$

$$A = \begin{pmatrix} \alpha_u r_1 + u_0 r_3 & \alpha_u t_x + u_0 t_z \\ \alpha_v r_1 + v_0 r_3 & \alpha_v t_y + v_0 t_z \\ r_3 & t_z \end{pmatrix} \quad (3.32)$$

Parametry kamery mohou být získány ze symbolické matice A porovnáním s numerickou maticí získanou kalibrací kamery Halloovou technikou. Orientace vektorů r_i musí být ortogonální a také skalární součin mezi dvěma vektory musí splňovat následující požadavky ortonormality:

$$\begin{aligned} r_i r_j^T &= 0 \quad i \neq j \\ r_i r_j^T &= 1 \quad i = j \end{aligned} \quad (3.33)$$

S využitím těchto vlastností lze ze vztahu (3.32) určit čtyři vnitřní parametry ($\alpha_u, \alpha_v, u_0, v_0$) a šest vnějších parametrů ($r_1, r_2, r_3, t_x, t_y, t_z$) následujícím způsobem:

$$\begin{aligned} u_0 &= A_1 A_3^T & v_0 &= A_2 A_3^T \\ \alpha_u &= -(A_1 A_1^T - u_0^2)^{1/2} & \alpha_v &= -(A_2 A_2^T - v_0^2)^{1/2} \\ r_1 &= \frac{1}{\alpha_u} (A_1 - u_0 A_3) & t_x &= \frac{1}{\alpha_u} (A_{14} - u_0 A_{34}) \\ r_2 &= \frac{1}{\alpha_v} (A_2 - v_0 A_3) & t_y &= \frac{1}{\alpha_v} (A_{24} - v_0 A_{34}) \\ r_3 &= A_3 & t_z &= A_{34} \end{aligned} \quad (3.34)$$

Kde lze matici A vyjádřit následujícím způsobem:

$$A = \begin{pmatrix} A_1 & A_{14} \\ A_2 & A_{24} \\ A_3 & A_{34} \end{pmatrix} \quad (3.35)$$

Ovšem před samotnou estimací parametrů kamery musí být matice A vypočítána. Faugeras představil novou variantu metody estimace matice A odvozenou z Hallovy metody. Z tohoto důvodu byl výraz převeden do následujícího tvaru:

$$\begin{aligned} A_1^w P_w + A_{14} - {}^I X_u (A_3^w P_w + A_{34}) &= 0 \\ A_2^w P_w + A_{24} - {}^I Y_u (A_3^w P_w + A_{34}) &= 0 \end{aligned} \quad (3.36)$$

Následně jsou vyjádřeny neznámé proměnné:

$$\begin{aligned} {}^I X_u &= \frac{A_1}{A_{34}} {}^w P_w + \frac{A_{14}}{A_{34}} - \frac{A_3}{A_{34}} {}^w P_w {}^I X_u \\ {}^I Y_u &= \frac{A_2}{A_{34}} {}^w P_w + \frac{A_{24}}{A_{34}} - \frac{A_3}{A_{34}} {}^w P_w {}^I Y_u \end{aligned} \quad (3.37)$$

Uvažujeme množinu pěti parametrů $X = (T_1, T_2, T_3, C_1, C_2)^T$, kde $T_1 = \frac{A_1}{A_{34}}$, $T_2 = \frac{A_3}{A_{34}}$, $T_3 = \frac{A_2}{A_{34}}$, $C_1 = \frac{A_{14}}{A_{34}}$ a $C_2 = \frac{A_{24}}{A_{34}}$.

$$\begin{aligned} {}^I X_u &= T_1 {}^w P_w + C_1 - T_2 {}^w P_w {}^I X_u \\ {}^I Y_u &= T_3 {}^w P_w + C_2 - T_2 {}^w P_w {}^I Y_u \end{aligned} \quad (3.38)$$

Hodnota vektoru X je získána pomocí lineární metody nejmenších čtverců:

$$B = QX, \quad (3.39)$$

kde

$$Q = \begin{pmatrix} {}^w P_{wi}^T & - {}^I X_{ui}^T {}^w P_{wi}^T & 0_{1 \times 3} & \dots & 1 & 0 \\ 0_{1 \times 3} & - {}^I Y_{ui}^T {}^w P_{wi}^T & {}^w P_{wi}^T & \dots & 0 & 1 \\ & & & \dots & & \end{pmatrix}, B = \begin{pmatrix} \dots \\ {}^I X_{ui} \\ {}^I Y_{ui} \\ \dots \end{pmatrix} \quad (3.40)$$

Pak je vektor X vypočítán pomocí předchozí rovnice.

$$X = (Q^T Q)^{-1} Q^T B \quad (3.41)$$

Nyní jsou parametry kamery získány z vektoru X pomocí rovnic (3.44):

$$\begin{aligned} T_1 &= \frac{r_3}{t_z} u_0 + \frac{r_1}{t_z} \alpha_u & C_1 &= u_0 + \frac{t_x}{t_z} \alpha_u \\ T_2 &= \frac{r_3}{t_z} & & \\ T_3 &= \frac{r_3}{t_z} v_0 + \frac{r_2}{t_z} \alpha_v & C_2 &= v_0 + \frac{t_y}{t_z} \alpha_v \end{aligned} \quad (3.42)$$

Uvažujeme, že norma tří vektorů orientace r_i je rovna jednotce. S použitím předchozích vztahů, je možné určit parametr t_z . Uvažujeme-li navíc, že $r_3 = 1$, pak

$$t_z = \frac{1}{\|T_2\|} \quad (3.43)$$

Zbytek parametrů může být získán za použití vlastností skalárního a vektorového součinu mezi vektory, které jsou:

$$\begin{aligned} v_1 v_2 &= \|v_1\| \|v_2\| \cos \alpha \\ \|v_1 \times v_2\| &= \|v_1\| \|v_2\| \sin \alpha \end{aligned} \quad (3.44)$$

Dále platí, že:

$$\begin{aligned} r_i r_j^T &= 0 & i &\neq j & \|r_i \times r_j\| &= 1 & i &\neq j \\ r_i r_j^T &= 1 & i &= j & \|r_i \times r_j\| &= 0 & i &= j \end{aligned} \quad (3.45)$$

Vnitřní parametry mohou být získány následujícím způsobem:

$$\begin{aligned} u_0 &= \frac{T_1 T_2^T}{\|T_2\|^2} & v_0 &= \frac{T_1 T_3^T}{\|T_2\|^2} \\ \alpha_u &= -\frac{\|T_1^T \times T_2^T\|}{\|T_2\|^2} & \alpha_v &= -\frac{\|T_2^T \times T_3^T\|}{\|T_2\|^2} \end{aligned} \quad (3.46)$$

Navíc vnější parametry, které reprezentují orientaci kamery, jsou následující:

$$\begin{aligned} r_1 &= -\frac{\|T_2\|}{\|T_1^T \times T_2^T\|} \left(T_1 - \frac{T_1 T_2^T}{\|T_2\|^2} T_2 \right) \\ r_2 &= -\frac{\|T_2\|}{\|T_2^T \times T_3^T\|} \left(T_3 - \frac{T_2 T_3^T}{\|T_2\|^2} T_2 \right) \\ r_3 &= \frac{\|T_2\|}{\|T_2\|} \end{aligned} \quad (3.47)$$

Vnější parametry, které reprezentují pozici kamery, lze vyjádřit následujícím způsobem:

$$\begin{aligned} t_x &= -\frac{\|T_2\|}{\|T_1^T \times T_2^T\|} \left(C_1 - \frac{T_1 T_2^T}{\|T_2\|^2} \right) \\ t_y &= -\frac{\|T_2\|}{\|T_2^T \times T_3^T\|} \left(C_2 - \frac{T_2 T_3^T}{\|T_2\|^2} \right) \\ t_z &= \frac{1}{\|T_2\|} \end{aligned} \quad (3.48)$$

Použitím vektorů r_i v rovnici (3.49) dostáváme přímo matici vyjadřující rotaci ${}^C R_w$. Tři úhly orientace kamery α, β, γ mohou být určeny z rovnice rovnosti symbolické matice ${}^C R_w$ a její numerické reprezentace získané kalibrací. Všechny parametry lineárního modelu Faugerase jsou nyní získány. Tyto parametry vyjadřují závislost mezi pozicí trojrozměrných bodů ve scéně a jejich dvourozměrných projekcí. Tento model je samozřejmě více přesný v případě, že je zahrnuto radiální zkreslení, jak bude zmíněno v následující kapitole.

3.4.3. Faugerasova metoda s radiálním zkreslením

V případě horší kvality obrazu již samotná Faugerasova metoda není dostačující. Může být, ale snadno modifikována tak, aby zahrnovala radiální zkreslení. Jednotlivé výrazy se pak stanou nelineárními a lineární metoda nejmenších čtverců musí být nahrazena iterativním algoritmem.

Kombinací rovnic (3.3), (3.4), (3.5) a (3.7) dostáváme:

$$\begin{aligned} {}^cX_d + {}^cX_d k_1 r^2 &= f \frac{r_{11}^w X_w + r_{12}^w Y_w + r_{13}^w Z_w + t_x}{r_{31}^w X_w + r_{32}^w Y_w + r_{33}^w Z_w + t_z} \\ {}^lY_d + {}^cY_d k_1 r^2 &= f \frac{r_{21}^w X_w + r_{22}^w Y_w + r_{23}^w Z_w + t_y}{r_{31}^w X_w + r_{32}^w Y_w + r_{33}^w Z_w + t_z} \\ r &= \sqrt{{}^cX_d^2 + {}^cY_d^2} \end{aligned} \quad (3.49)$$

Navíc je nutné využít rovnice (3.16) pro transformaci z metrického souřadného systému do diskrétního systému v pixelech. Pak rovnice (3.49) definuje vektor neznámých, který je možný vypočítat za použití iteračních metod (např. Newtonova metoda).

$$x = (\alpha, \beta, \gamma, t_x, t_y, t_z, k_u, k_v, u_0, v_0, k_1)^T \quad (3.50)$$

Pro příklad Newtonova metoda minimalizuje následující výraz:

$$G(x_k) \approx G(x_{k-1}) + J\Delta x_k, \quad (3.51)$$

kde x je vektor neznámých, $G(x)$ je minimalizovaná funkce, $G(x_k)$ je hodnota blízkého řešení a J interpretuje jakobián funkce $G(x)$. Pro nalezení řešení je nezbytné, aby se hodnota funkce $G(x_k)$ blížila nule.

$$G(x_k) = 0 \quad (3.52)$$

Jedním z hlavních problémů iterativních algoritmů je konvergence k řešení, která je ovlivněna počátečním odhadem. Počáteční odhad však může být snadno získán kalibrováním pomocí lineární metody Faugere bez radiálního zkreslení, kde předpokládáme hodnotu $k_1 = 0$. Pro každou iteraci je nezbytné vypočítat hodnotu Δx_k pro získání nové hodnoty x .

$$J\Delta x_k = -G(x_{k-1}) \quad (3.53)$$

Pak úpravou rovnic (3.49) a (3.16) jsou funkce $U(x)$ a $V(x)$ ve tvaru:

$$\begin{aligned} U(x) &= f \frac{r_{11}^w X_w + r_{12}^w Y_w + r_{13}^w Z_w + t_x}{r_{31}^w X_w + r_{32}^w Y_w + r_{33}^w Z_w + t_z} - \frac{({}^I X_d - u_0)}{-k_u} \\ &\quad - k_1 \left(\left(\frac{({}^I X_d - u_0)}{-k_u} \right)^2 + \left(\frac{({}^I Y_d - v_0)}{-k_v} \right)^2 \right) \frac{({}^I Y_d - v_0)}{-k_v} \\ V(x) &= f \frac{r_{21}^w X_w + r_{22}^w Y_w + r_{23}^w Z_w + t_y}{r_{31}^w X_w + r_{32}^w Y_w + r_{33}^w Z_w + t_z} - \frac{({}^I Y_d - v_0)}{-k_v} \\ &\quad - k_1 \left(\left(\frac{({}^I X_d - u_0)}{-k_u} \right)^2 + \left(\frac{({}^I Y_d - v_0)}{-k_v} \right)^2 \right) \frac{({}^I X_d - u_0)}{-k_u} \end{aligned} \quad (3.54)$$

Abychom nyní mohli nalézt řešení, je nezbytné aplikovat rovnice (3.56) na n bodů kalibrace. Pro nalezení řešení musíme aplikovat výraz (3.53), jehož použití vyžaduje symbolický zápis funkce $G(x_k)$ a její matici parciálních derivací jako prvky matice J . Jak je ukázáno v následujících rovnicích:

$$G(x_{k-1}) = \begin{pmatrix} U_1(x_{k-1}) \\ V_1(x_{k-1}) \\ \vdots \\ V_n(x_{k-1}) \end{pmatrix} \quad (3.55)$$

$$J = \begin{pmatrix} \frac{\partial U_1(x_{k-1})}{\partial \alpha} & \frac{\partial U_1(x_{k-1})}{\partial \beta} & \dots & \frac{\partial U_1(x_{k-1})}{\partial k_1} \\ \frac{\partial V_1(x_{k-1})}{\partial \alpha} & \frac{\partial V_1(x_{k-1})}{\partial \beta} & \dots & \frac{\partial V_1(x_{k-1})}{\partial k_1} \\ \vdots & \dots & \ddots & \vdots \\ \frac{\partial V_n(x_{k-1})}{\partial \alpha} & \frac{\partial V_n(x_{k-1})}{\partial \beta} & \dots & \frac{\partial V_n(x_{k-1})}{\partial k_1} \end{pmatrix} \quad (3.56)$$

Nakonec získáváme parametry daného modelu pomocí použití pseudoinverzní transformace rovnice (3.57) v každé iteraci. Čím více je vykonáno iterací, tím lepší přesnosti je dosaženo, samozřejmě po určitém počtu iterací se již řešení prakticky nemění.

$$\begin{aligned} \Delta x_k &= -(J^T J)^{-1} J^T G(x_{k-1}) \\ x_k &= x_{k-1} + \Delta x_k \end{aligned} \quad (3.57)$$

3.4. Určení přesnosti

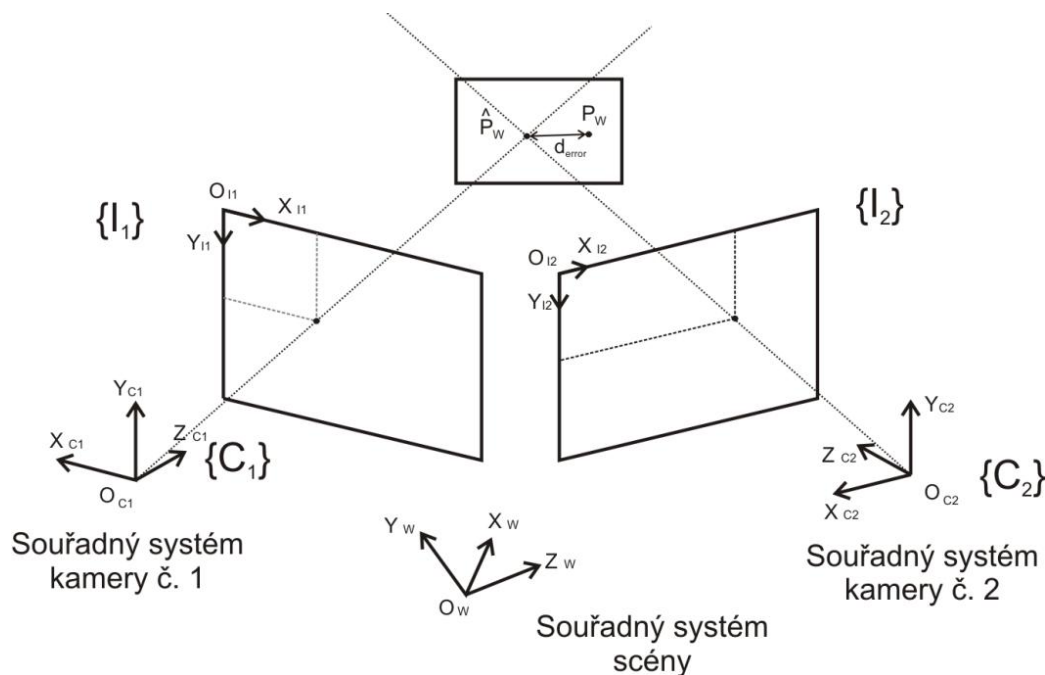
Metody použité k výpočtu přesnosti kalibrace kamery mohou být klasifikovány do dvou skupin.

1. První skupina je založena na analyzování odchylek mezi reálnými pozicemi trojrozměrných bodů ve scéně vzhledem k trojrozměrné pozici vypočítané z dvourozměrné projekce bodu ve snímaném obrazu.
2. Druhá skupina metod porovnává reálnou (nasnímanou) pozici pixelu ve dvourozměrném obrázku vzhledem k přepočítané pozici trojrozměrného bodu ve scéně na obrazovou plochu obrázku.

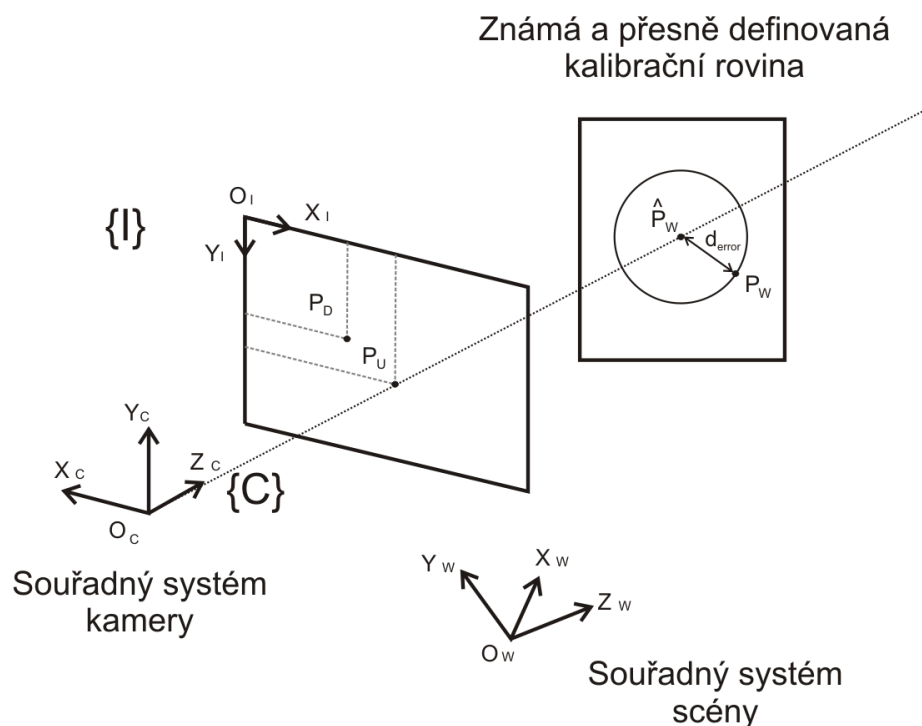
3.5.1. Měření v trojrozměrném prostoru

V následujícím textu jsou vysvětleny a popsány některé nejvíce používané metody měření přesnosti:

1. **Trojrozměrná pozice získaná pomocí stereo triangulace:** V prvním kroku této metody jsou pořízeny dva obrázky scény, která obsahuje množinu trojrozměrných testovacích bodů, jejichž pozice jsou dobře známy. V druhém kroku, jsou vypočítány trojrozměrné souřadnice stejných bodů z jejich projekcí a za použití kalibračních parametrů daných kamer. Tato pozice je určena jako průsečík optických paprsků obou kamer. Na závěr je porovnávána odchylka mezi reálnými a vypočítanými body. Přesnost záleží na kvalitě kalibrace obou kamer a použitém matematickém modelu kalibrace. Princip tohoto výpočtu lze nalézt na Obr. 3.9.



Obr. 3.9: Princip měření přesnosti



Obr. 3.10: Princip měření přesnosti

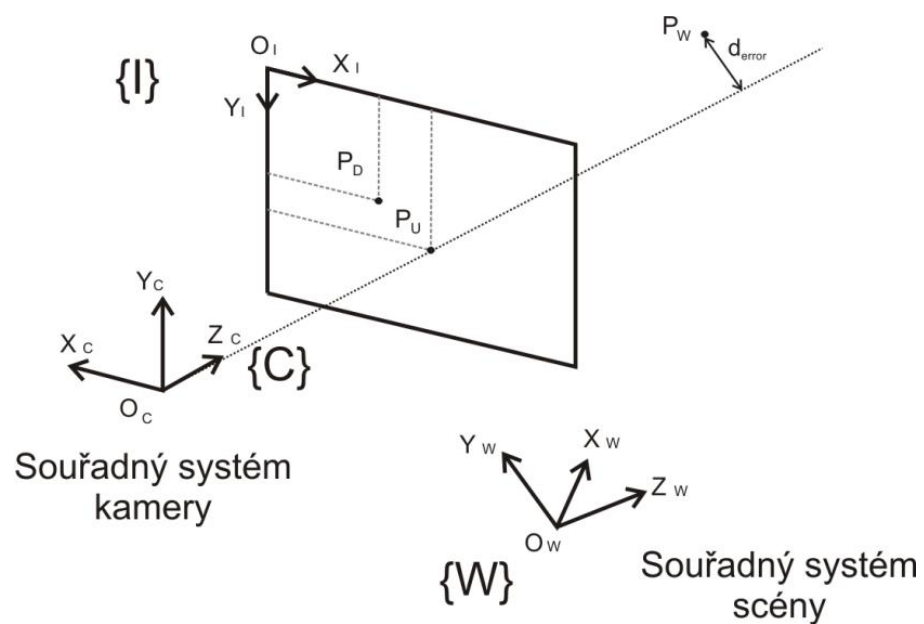
2. **Rádus nejednoznačnosti v kalibrační ploše.** Nejprve určíme množinu trojrozměrných testovacích bodů, které leží na stejné testovací ploše a jejichž pozice v souřadném systému scény jsou dobře známy. Dále je pro každý bod v dvourozměrném obraze scény použit kalibrační model kamery pro výpočet optického paprsku mezi ohniskovým bodem kamery a bodem v obraze. Zkoumáním optického paprsku a jeho průsečíku s testovací plochou detekujeme bod průsečíku. Vzdálenost mezi reálnou trojrozměrnou polohou testovacího bodu a vypočteným průsečíkem definuje poloměr nejednoznačnosti okolo tohoto bodu. Princip metody lze nalézt na následujícím obrázku Obr. 3.10.
3. **Vzdálenost s ohledem na optický paprsek.** Tato metoda je zobecněním předchozích metod. Odchylna měření je dána jako vzdálenost mezi reálnou trojrozměrnou pozicí testovacích bodů a optickým paprskem generovaným jeho projekcí. Výhodou je, na

rozdíl od metody „trojrozměrná pozice získané pomocí stereo triangulace“, požadavek na data pouze z jedné kamery. Na rozdíl od metody „radius nejednoznačnosti v kalibrační ploše“ není zapotřebí jen bodů z jedné plochy. Příklad metody lze nalézt na Obr. 3.11.

4. **NSCE** (normalizovaná chyba stereo kalibrace) [43]. Množina pixelů v obrázku je pomocí kalibračního modelu kamery transformována zpět do scény, tak že každý zpětně promítnutý pixel pokrývá určitou plochu ve scéně. Tato plocha pak určuje rozlišení metody pro danou vzdálenost. Orientace plochy je kolmá k optické ose. Příklad lze nalézt na Obr. 3.12.

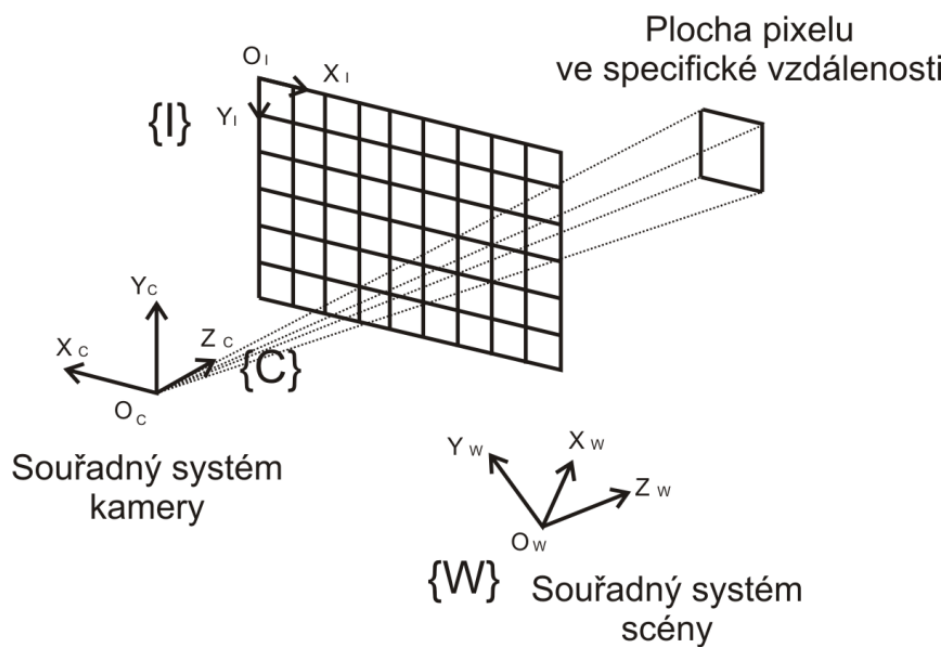
Nechť je dáno n zkoumaných bodů. Hloubka plochy i -tého trojrozměrného bodu je rovna ${}^cZ_{wi}$, dále řádek a sloupec ohniskové vzdálenosti je α_u a α_v . Zpětná projekce pixelu na tuto plochu je pak obdélník o velikost $a \times b$. Nechť jsou skutečné souřadnice i -tého trojrozměrného bodu definovány jako uspořádaná trojice $({}^cX_{wi}, {}^cY_{wi}, {}^cZ_{wi})$ v souřadném systému kamery a nechť souřadnice získány zpětnou projekcí pixelu a jeho průsečíku s povrchovou plochou jsou také reprezentovány jako uspořádaná trojice $({}^c\hat{X}_{wi}, {}^c\hat{Y}_{wi}, {}^c\hat{Z}_{wi})$ v souřadném systému kamery. Normalizovaná chyba stereo kalibrace (Normalized Stereo Calibration Error) NSCE je pak definována jako:

$$NSCE = \frac{1}{n} \sum_{i=1}^n \left[\frac{({}^c\hat{X}_{wi} - {}^cX_{wi})^2 + ({}^c\hat{Y}_{wi} - {}^cY_{wi})^2}{\frac{{}^c\hat{Z}_{wi}^2 (\alpha_u^{-2} + \alpha_v^{-2})}{12}} \right]^{1/2} \quad (3.58)$$



Obr. 3.11: Princip měření přesnosti

Obrazová plocha



Obr. 3.12: Princip měření přesnosti

3.5.2. Měření ve dvourozměrném prostoru

Následující sekce popíše dvě možnosti výpočtu přesnosti v ploše obrazu v jednotkách pixelů:

1. **Přesnost ze zkreslených souřadnic v obrazu:** Nejprve připravíme obraz obsahující množinu testovacích bodů se známou trojrozměrnou polohou. Následně určíme jejich dvourozměrné pozice, jako projekci získanou modelem kamery do plochy obrazu a to s ohledem na zkreslení. Přesnost metody je pak získána jako měření odchylky mezi reálnými dvourozměrnými body (získanými pomocí metod rozpoznávání obrazu) a body vypočtenými (získanými pomocí projekce modelem kamery).
2. **Přesnost z nezkreslených souřadnic v obrazu:** Nejprve připravíme obraz obsahující množinu testovacích bodů se známou trojrozměrnou polohou. Následně určíme jejich dvourozměrné pozice, jako lineární projekci modelem kamery do plochy obrazu bez ohledu na zkreslení čočky. Pokračuje se rozpoznáním reálných dvourozměrných pozic bodů v obrazu a odstraněním zkreslení, pomocí použití modelu kamery a tedy získání množiny bodů s nezkreslenými souřadnicemi. Přesnost je pak získána jako měření odchylky mezi body vypočtenými (získanými pomocí projekce modelem kamery) a reálnými nezkreslenými dvourozměrnými body (získanými pomocí metod rozpoznávání obrazu a odstraněním zkreslení).

3.5. Experimentální výsledky

Experimentální výsledky jsou prezentovány v této sekci ze dvou hlavních důvodů:

1. Prvním z nich je porovnání jednotlivých metod kalibrace kamer z hlediska přesnosti a výpočetní náročnosti.

2. Druhým důvodem je aplikace kalibračních metod na reálný případ z oblasti robotiky. Použití reálných dat je velice důležitým faktorem, jelikož výpočetní metody nejsou ovlivněny pouze chybou modelu kamery a numerickou metodou použitou pro minimalizaci. Je nutné si uvědomit, že v reálné aplikaci je nutné počítat s chybou způsobenou nepřesnou detekcí trojrozměrné polohy bodu ve scéně a také nepřesnou detekcí dvourozměrné polohy bodu v nasnímaném obraze.

Vlastní realizace automatizované metody pro robustní detekci polohy efektoru ramene robotu byla představen v předešlé kapitole. Použitím dat, získaných touto jsme nyní schopni ověřit funkčnost jednotlivých metod kalibrace kamery a ověřit jejich rychlost a přesnost.

Je nutné upozornit, že výpočet byl provádění na notebooku Asus A6Ja s procesorem Intel Core Duo T2400 1,83GHz, 1GB paměti RAM a operačním systémem Windows Vista. Jak již bylo zmíněno veškerý kód byl řešen jako řízený kód v jazyce C#, implementovaný pro platformu. NET.

V následující tabulce Tab. 3.1 naleznete srovnání rychlostí jednotlivých kalibračních metod. Jedná se o dobu výpočtu kalibračních parametrů pro jednu kameru. Doba výpočtu je vyjádřena v milisekundách a je zaokrouhlena na celá čísla.

Z výsledků je jasně patrné, že lineární metody (Hallova a Faugerasova) jsou velice rychlé, zatímco iterativní metody jsou v důsledku nutnosti iterování k řešení pomalejší.

Metoda\Počet bodů	8	100	268
Hallova	1ms	1ms	2ms
Faugerasova	1ms	1ms	2ms
Faugerasova se zkreslením	1ms	13ms	21ms

Tab. 3.1: Výpočetní rychlost kalibračních metod

V následujících tabulkách lze nalézt vypočítané hodnoty přesností jednotlivých metod pro 268 párů trojrozměrných bodů ve scéně a jejich dvourozměrných projekcí v obraze.

Tab. 3.2 ukazuje vzdálenost vypočteného optického paprsku od reálné polohy trojrozměrného bodu v pracovním prostoru ramene robotu viz kapitola 3.5.1.1. Tabulka obsahuje průměrnou hodnotu chyby trojrozměrné pozice v milimetrech, její směrodatnou odchylku σ a maximum. Dále obsahuje vypočtenou hodnotu NSCE viz kapitola 3.5.1.4.

Metoda	Průměr [mm]	σ [mm]	Max [mm]	NSCE
Hallova	0.3775	0.4753	3.0459	-
Faugerasova	0.3731	0.4956	3.1423	0.9162
Faugerasova se zkreslením	0.3742	0.4852	2.9881	0.9171

Tab. 3.2: Porovnání trojrozměrné přesnosti

Metoda	Průměr [px]	σ [px]	Max [px]
Hallova	0.3774	0.4661	2.9731
Faugerasova	0.3741	0.4861	3.0683
Faugerasova se zkreslením	0.3743	0.4759	2.9192

Tab. 3.3: Porovnání dvourozměrné přesnosti v nezkresleném obraze

Metoda	Průměr [px]	σ [px]	Max [px]
Hallova	0.3774	0.4660	2.9730
Faugerasova	0.3740	0.4861	3.0683
Faugerasova se zkreslením	0.3742	0.4758	2.9174

Tab. 3.4: Porovnání dvourozměrné přesnosti ve zkresleném obraze

Tab. 3.3 a Tab. 3.4 obsahuje přesnosti měření pozic obrazových bodů ve zkeslených i nezkreslených souřadnicích viz kapitola 3.5.2. Tabulka obsahuje průměrnou hodnotu chyby dvourozměrné pozice v obraze v pixelech, její směrodatnou odchylku σ a maximum.

Z výsledku je patrné, že průměrné hodnoty chyb jsou poměrně nízké a metodu kalibrace kamery lze použít pro širokou škálu úloh. Dále jsme vyhodnotili, že metoda generování dvojic reálných trojrozměrných bodů ve scéně a jejich dvourozměrných projekcí v nasnímaných obrazech je velice přesná. Navíc zachovává vysokou rychlost výpočtu a je schopná fungce i za velmi špatných světelných podmínek.

4. Prostorová rekonstrukce

Prostorová rekonstrukce slouží k získání trojrozměrné informace scény snímané jednou pohybující se kamerou, nebo statickou strukturou složenou minimálně ze dvou kamer. Vzájemný vztah mezi rozdílnými pohledy na scénu lze popsat epipolární geometrie, která je blíže popsána v této kapitole.

4.1. Úvod

Jedním z možných přístupů prostorové rekonstrukce je použití kalibrace kamery, zmíněné v kapitole 3. V tomto případě je identifikován model kamery, který modeluje vztah mezi reálnými trojrozměrnými pozicemi bodů ve scéně a jejich dvourozměrnými projekcemi snímanými pomocí kamer.

Model kamery může být použit k následujícím operacím:

1. Výpočet polohy dvourozměrné projekce z trojrozměrné polohy reálného bodu ve scéně.
2. Výpočet trojrozměrného optického paprsku procházejícího známou polohou jeho dvourozměrné projekce.

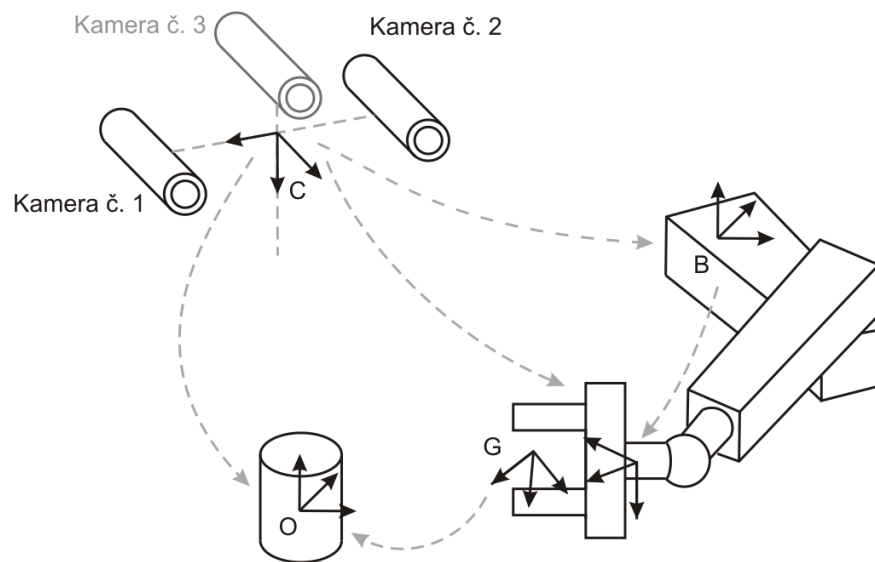
Ke zjištění prostorové informace o snímané scéně je ovšem zapotřebí použít nejméně dvou optických paprsků. S jejich pomocí je pak možné určit trojrozměrnou informaci pomocí metody triangulace.

Množina paprsků může být získána pomocí:

1. Jedné pohybující se kamery, nesené např. na ramenu robotu, či na mobilní robotické plošině. Její poloha a orientace musí být přesně známa.

2. Systému binokulárního vidění sestaveného z fixní struktury dvou kamer, resp. ze tří (systém trinokulárního vidění) a více kamer. Příklad systému binokulárního, resp. trinokulárního (třetí kamera je zobrazena odlišnou barvou) systému vidění lze nalézt na Obr. 4.1. Systém je složen ze dvou, resp. tří šterbinových kamer, s jejichž pomocí je snímána scéna okolí ramene.

Je nutné upozornit, že v aktivních systémech mohou být optické a geometrické charakteristiky kamer dynamicky měněny a to v důsledku změn scény, pohybu ve scéně a změny vlastností kamer.



Obr. 4.1: Systém prostorového vidění robotu

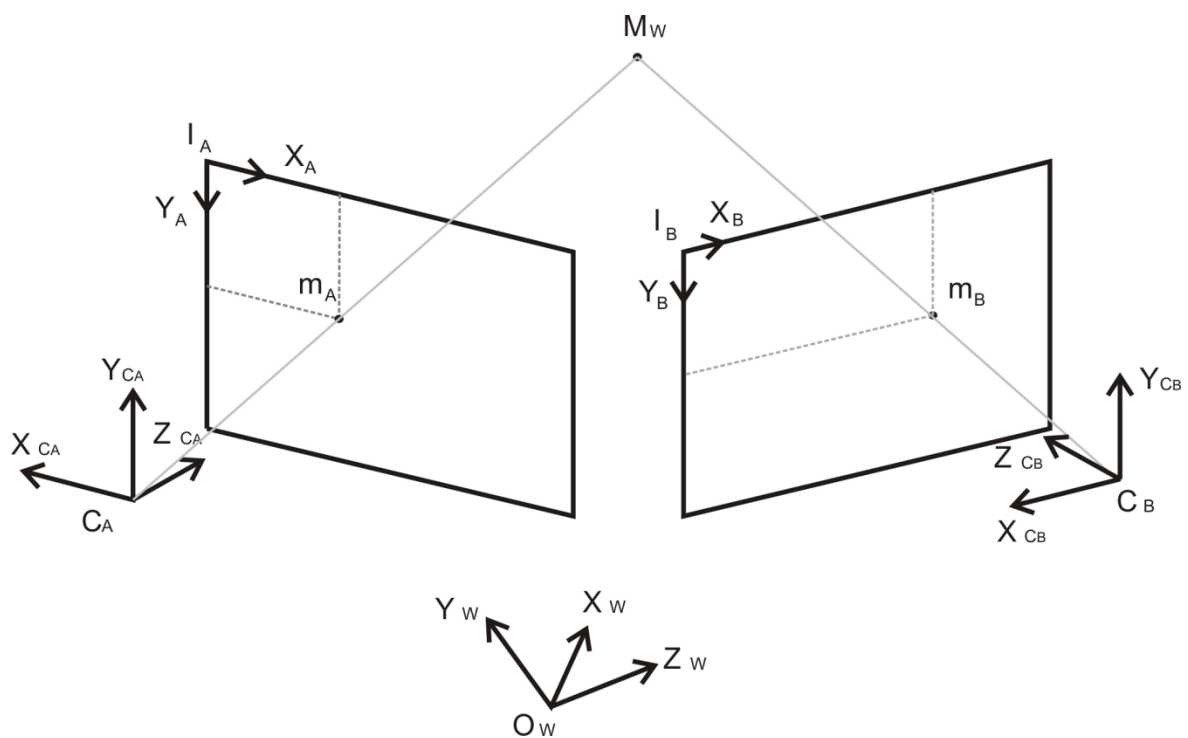
Vzhledem k fixní pozici jednotlivých kamer systému stereo vidění je možné modelovat jejich vzájemný vztah. Geometrie popisující tento vztah se nazývá epipolární geometrie. Poprvé byla publikována Longuet-Higginsem v roce 1981. Od této doby bylo vyvinuto velké množství výrazných vylepšení. Rozsáhlý počet prací se věnuje tematikou samo se kalibrující a nekalibrované systémy. Pro příklad, v roce 1992 publikoval Faugeras krátkou zpráva o samo

se kalibrujícím systémem, které byly následně použity k určení parametrů kamery pro epipolární geometrii [19].

V této kapitole bude stručně popsán matematický aparát převzatý z [58], [59] a odvozeny základní vztahy pro jeho praktickou aplikaci. Matematický aparát je zjednodušený pro konkrétní případ, kdy kamery snímající scénu jsou zaměřeny na shodný objekt a jsou vůči sobě natočeny (viz Obr. 4.2).

4.2. Princip triangulace

Princip triangulace je založen na výpočtu trojrozměrné pozice bodu ve scéně z průsečíku nejméně dvou optických paprsků. Schéma lze nalézt na Obr. 4.2.



Obr. 4.2: Snímání scény pomocí dvou kamer

V praxi ovšem kamera nefunguje jako ideální štěrbinu a to z důvodu zkreslení objektivu. Také jsou zde chyby způsobené digitalizací obrazu. Problém je způsoben tím, že se optické paprsky neprotínají v jednom bodě.

Proces triangulace musí být upraven tak, aby našel bod, který leží uprostřed nejkratší spojnice obou optických paprsků. Zvažme dvourozměrný obrazový bod m_A a korespondující bodu m_B je v druhém obrázku. Oba body m_A a m_B jsou obrazy stejného trojrozměrného bodu scény M . Matematické modely obou kamer umožňují nalezení polohy bodů m_A a m_B s respektem na souřadnicový systém kamery. Také jsme schopni určit trojrozměrný směrový vektor optického paprsku jako funkci polohy obrazového bodu $m_A = ({}^I X_A, {}^I Y_A, 1)^T$, $m_B = ({}^I X_B, {}^I Y_B, 1)^T$:

$$\begin{aligned}\delta_A(m_A) &= N_A \cdot m_A \\ \delta_B(m_B) &= N_B \cdot m_B\end{aligned}\tag{4.1}$$

Optický paprsek je pak definován jako přímka procházející optickým centrem kamery a se směrem odpovídajícím směrovému vektoru získaným podle vztahu (4.1).

Úkolem je nalezení nejkratší spojnice obou paprsků. Jedná se o úsečku definovanou body P_A a P_B , kde polohy těchto krajních bodů jsou:

$$\begin{aligned}P_A &= C_A + \alpha \cdot \delta_A(m_A) \\ P_B &= C_B + \beta \cdot \delta_B(m_B)\end{aligned}\tag{4.2}$$

Hodnoty parametrů α , β jsou z intervalu $\langle 0; \infty \rangle$ a lze je nalézt řešením soustavy rovnic:

$$\begin{aligned}(P_A - P_B) \times \delta_A(m_A) &= 0 \\ (P_A - P_B) \times \delta_B(m_B) &= 0\end{aligned}\tag{4.3}$$

Po dosazení dostáváme:

$$\begin{aligned}
(C_A - C_B + \alpha \cdot \delta_A(m_A) - \alpha \cdot \delta_B(m_B)) \times \delta_A(m_A) &= 0 \\
(C_A - C_B + \alpha \cdot \delta_A(m_A) - \beta \cdot \delta_B(m_B)) \times \delta_B(m_B) &= 0
\end{aligned} \tag{4.4}$$

Po dalších úpravách, které zde neuvádíme, dostáváme řešení ve tvaru:

$$\begin{aligned}
\alpha &= \frac{d_1 d_2 - d_3 - d_4}{d_5 d_4 - d_2 d_2} \\
\beta &= \frac{d_1 + d_2 \alpha}{d_4}
\end{aligned} \tag{4.5}$$

Hodnoty proměnných d_1, d_2, d_3, d_4, d_5 :

$$\begin{aligned}
d_1 &= (C_A - C_B) \cdot [\delta_B(m_B)]^T \\
d_2 &= \delta_A(m_A) \cdot [\delta_B(m_B)]^T \\
d_3 &= (C_A - C_B) \cdot [\delta_A(m_A)]^T \\
d_4 &= \delta_B(m_B) \cdot [\delta_B(m_B)]^T \\
d_5 &= \delta_A(m_A) \cdot [\delta_A(m_A)]^T
\end{aligned} \tag{4.6}$$

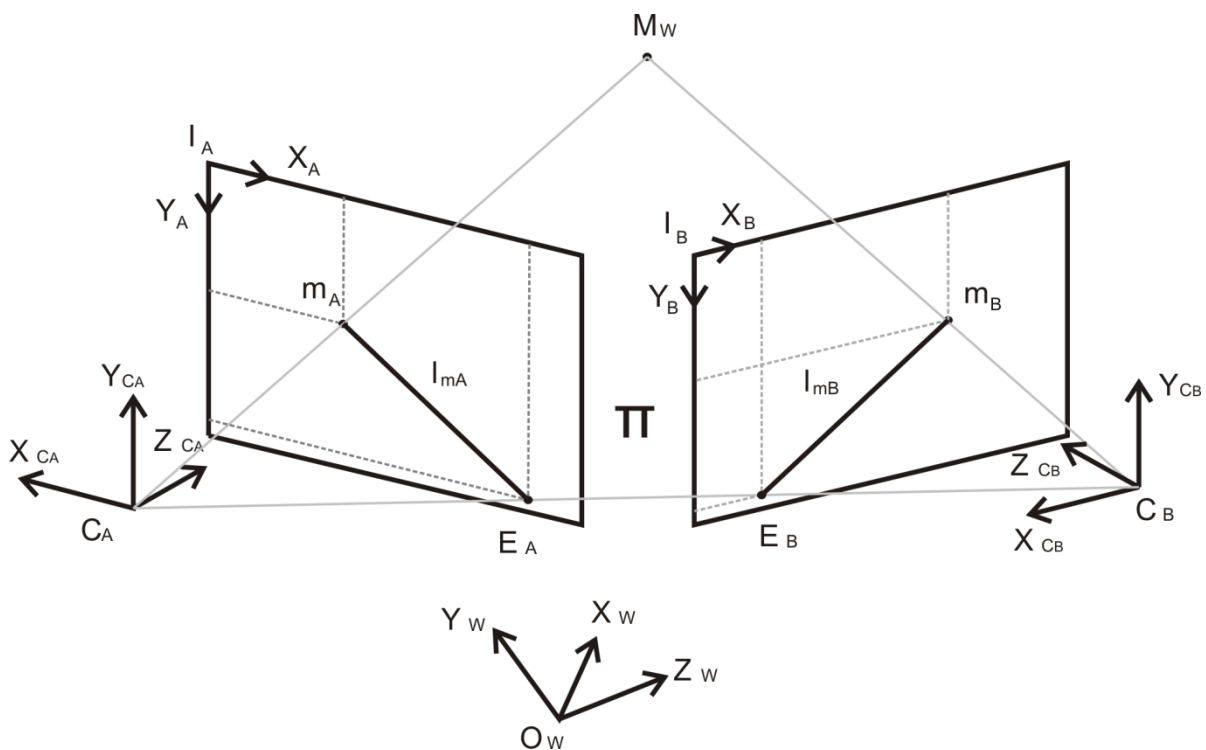
Po vyjádření polohy bodů P_A a P_B lze snadno vyjádřit polohu triangulovaného bodu P jako bod uprostřed nejkratší spojnice optických paprsků:

$$P = \frac{P_A + P_B}{2} \tag{4.7}$$

4.3. Epipolární geometrie

Epipolární geometrie definuje vztah mezi dvěma kamerami. Zvažme trojrozměrný bod ve scéně $M = ({}^W X, {}^W Y, {}^W Z, 1)^T$ vyjádřen vzhledem k souřadnému systému scény $\{W\}$ a jeho dvourozměrné projekce $m_A = ({}^I X_A, {}^I Y_A, 1)^T$ a $m_B = ({}^I X_B, {}^I Y_B, 1)^T$ v pixelech na snímcích získaných pomocí dvou statických kamer. Tři body m_A, m_B a M určují plochu Π , která protíná

obě obrazové plochy v epipolárních přímkách l_{mA} a l_{mB} , nebo naopak. Stejná plocha ovšem může být vypočítána použitím ohniskových bodů C_A a C_B a jedné dvourozměrné projekce m_A , nebo m_B . Tento princip redukuje problém korespondence v celé ploše obrazu na vyhledávání podél jednoduché úsečky, tedy části epipolární přímky. Schematický příklad epipolární přímky a epipólu lze nalézt na Obr. 4.3.



Obr. 4.3: Příklad epipolární přímky

Nechť jsou známy matematické modely dvou použitých kamer (viz kapitola 3), vyjádřené jako matice A, B rozměru 4×4 .

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{pmatrix} \quad (4.8)$$

$$B = \begin{pmatrix} B_{11} & B_{12} & B_{13} & B_{14} \\ B_{21} & B_{22} & B_{23} & B_{24} \\ B_{31} & B_{32} & B_{33} & B_{34} \\ B_{41} & B_{42} & B_{43} & B_{44} \end{pmatrix}$$

3.5.1. Výpočet polohy optických středů kamer

Nejprve je nutné určit optické středy obou kamer. Pro zjednodušení výpočtu nejprve vyjádříme matici T_A a vektor U_A z matice A první kamery.

$$T_A = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} \quad (4.9)$$

$$U_A = \begin{pmatrix} -A_{14} \\ -A_{24} \\ -A_{34} \end{pmatrix}$$

Analogicky stejným způsobem vyjádříme matici T_B a vektor U_B z matice B druhé kamery.

Optické středy obou kamer C_A a C_B pak vypočteme podle vztahu:

$$C_A = T_A^{-1} \cdot U_A$$

$$C_B = T_B^{-1} \cdot U_B \quad (4.10)$$

3.5.2. Výpočet polohy epipólů

Pokud optické středy vyjádříme jako:

$$C_A = \begin{pmatrix} c_{Ax} \\ c_{Ay} \\ c_{Az} \end{pmatrix}, \quad C_B = \begin{pmatrix} c_{Bx} \\ c_{By} \\ c_{Bz} \end{pmatrix}, \quad (4.11)$$

Ize snadno určit polohu epipólů E_A , E_B obou kamer následujícím způsobem:

$$E_A = A \begin{pmatrix} c_{Bx} \\ c_{By} \\ c_{Bz} \\ 1 \end{pmatrix}, \quad E_B = B \begin{pmatrix} c_{Ax} \\ c_{Ay} \\ c_{Az} \\ 1 \end{pmatrix}, \quad (4.12)$$

kde

$$E_A = \begin{pmatrix} E_{Ax} \\ E_{Ay} \\ E_{Az} \end{pmatrix}, \quad E_B = \begin{pmatrix} E_{Bx} \\ E_{By} \\ E_{Bz} \end{pmatrix} \quad (4.13)$$

3.5.3. Urychlení výpočtu směrového vektoru epipolární přímky

Pro usnadnění výpočtu vektoru optického paprsku z dvourozměrné projekce je nutné zavést vektory t_{A1} , t_{A2} , t_{A3} jako řádky matice A:

$$t_{A1} = \begin{pmatrix} A_{11} \\ A_{12} \\ A_{13} \end{pmatrix}, \quad t_{A2} = \begin{pmatrix} A_{21} \\ A_{22} \\ A_{23} \end{pmatrix}, \quad t_{A3} = \begin{pmatrix} A_{31} \\ A_{32} \\ A_{33} \end{pmatrix} \quad (4.14)$$

Dále vyjádříme vektorové součiny n_{A1} , n_{A2} , n_{A3} následujícím způsobem:

$$\begin{aligned} n_{A1} &= t_{A2} \times t_{A3} \\ n_{A2} &= t_{A3} \times t_{A1} \\ n_{A3} &= t_{A1} \times t_{A2} \end{aligned} \quad (4.15)$$

Sestavíme matici vektorových součinů N_A rozměru 3×3 tak že jednotlivé sloupce matice po řadě doplníme prvky vektorů n_{A1} , n_{A2} , n_{A3} :

$$N_A = \begin{pmatrix} n_{A1}^T & n_{A2}^T & n_{A3}^T \end{pmatrix} \quad (4.16)$$

Zavedeme matici S_B rozměru 2×3 sestavenou z prvků vektoru E_B následujícím způsobem:

$$S_B = \begin{pmatrix} E_{Bz} & 0 & -E_{Bx} \\ 0 & E_{Bz} & -E_{By} \end{pmatrix} \quad (4.17)$$

Dále vypočteme matici M_{BA} :

$$M_{BA} = S_B \cdot (T_B \cdot N_A) \quad (4.18)$$

Pomocí této matice jsme okamžitě schopni pro libovolný bod $m_B = ({}^I X_B, {}^I Y_B, 1)^T$ ve snímku pořízeném kamerou (popsanou maticí B) určit dvourozměrný vektor směru epipolární přímky, jako funkci λ_b polohy bodu m_B :

$$\lambda_b(m_b) = M_{BA} \cdot m_b \quad (4.19)$$

Analogickým způsobem lze odvodit matici M_{AB} . Pomocí které pro libovolný bod m_A ve snímku pořízeném kamerou (popsanou maticí A) určit dvourozměrný vektor směru epipolární přímky, jako funkci λ_a polohy bodu m_a :

$$\lambda_a(m_a) = M_{AB} \cdot m_a \quad (4.20)$$

3.5.4. Rektifikace

Jak již bylo uvedeno, pokud vyhledáváme podél epipolární přímky, dojde k výraznému urychlení výpočtu. Prohledávaná plocha je tak redukována na úsečku. Pro urychlení výpočtu je výhodnější dvojici zpracovávaných obrazů transformovat takovým způsobem, aby odpovídající si párové pixely ležely v obou obrazech na stejných řádcích.

Rektifikace je tedy proces, při kterém se obraz transformuje takovým způsobem, že epipolární přímky jsou v obou obrazech zarovnány horizontálně.

Pro rychlý výpočet je výhodně vyjádřit vektory c_1, c_2, c_3 :

$$\begin{aligned}
c_2 &= C_A \times C_B \\
c_1 &= c_2 \times C_B \\
c_3 &= (C_A - C_B) \times c_2
\end{aligned} \tag{4.21}$$

S jejich pomocí pak vyjádříme matici optických středů C rozměru 3×3 tak, že doplníme její řádky po řadě obsahem vektorů c_1, c_2, c_3 :

$$C = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} \tag{4.22}$$

Dále s použitím vztahu (4.17) sestavíme matice Q_1 a Q_2 :

$$\begin{aligned}
Q_1 &= C \cdot N_A \\
Q_2 &= C \cdot N_B
\end{aligned} \tag{4.23}$$

Nyní lze provést rektifikaci všech bodů obrazu podle následujícího vztahu:

$$m_A'^* = Q_1 \cdot m_A, \tag{4.24}$$

kde $m_A = ({}^IX_A, {}^IY_A, 1)^T$ je bod zdrojového obrazu ve snímku pořízeném první kamerou (popsanou maticí A) a $m_A'^* = ({}^{I'}X_A, {}^{I'}Y_A, {}^{I'}W_A)^T$ je odpovídající bod rektifikovaného obrazu, který je nutné normalizovat:

$$m_A' = \begin{pmatrix} {}^{I'}X_A / {}^{I'}W_A \\ {}^{I'}Y_A / {}^{I'}W_A \\ 1 \end{pmatrix}, \tag{4.25}$$

Transformaci jednotlivých bodů druhého obrazu lze provést analogickým způsobem.

5. Hlubkové vnímání

Hlubkové vnímání je jedním z nejdůležitějších procesů rozpoznávání obrazu u člověka a vyšších obratlovců. Zejména proto, že umožňuje vysoce přesné pohyby v prostoru a vizuální zpětnou vazbu.

Metody hloubkového vnímání:

1. **Zaostřování:** Pro danou ohniskovou vzdálenost se předměty v určité vzdálenosti jeví jako ostré a předměty vzdálenější či bližší se jeví jako rozmazané. Monotónní závislost mezi ohniskovou vzdáleností a ostrostí objektů v závislosti na jejich vzdálenosti nabízí možnost využití v oblasti prostorového vidění.
2. **Hlubková informace z pohybu:** Jedná se o detekci prostorové informace z pohybu. Pro příklad zmiňme např. pozorování z vlaku. Pohyb vzdálených objektů je velice pomalý, na rozdíl od pohybu blízkých předmětů, který se jeví jako velice rychlý. Tento efekt je pro určení hloubkové informace velice dobře použitelný. V případě, že je rychlost objektu známa, je možné určit jeho vzdálenost.
3. **Kognitivní metody:** Ze znalosti velikosti určitého typu objektu jsme schopni jeho pozorování, určit, jak daleko se nachází. V rámci pozorování dochází k perspektivní projekci a tedy odpovídajícímu zmenšování vzdálených objektů. Pokud známe parametry perspektivní projekce a typickou velikost objektu, jsme schopni určit jeho vzdálenost.
4. **Stereovidění:** Jedná se o nejvýznamnější metodu, která umožňuje vnímat hloubkovou informaci pozorované scény. Měření vzdálenosti je založené na současném vnímání scény ze dvou či více rozdílných pohledů. Vzdálené objekty se v obou pohled liší jen velmi málo, na rozdíl od objektů blízkých, jejichž posunutí je velmi výrazné.

Každý z těchto přístupů ovšem má ale své stinné stránky a omezení. Hloubka detekovaná pomocí metody zaostřování nás nutí použít mobilní optický systém s omezenou hloubkou pohledu. Hloubková informace získaná z pohybu nás zase omezuje pouze na konstantní pohyby a ze své podstaty způsobuje slepý bod přímo před kamerou, který je nutné kompenzovat. Kognitivní způsob na druhou stranu vyžaduje schopnost učení se a rozpoznávání obrazu, který je výpočetně velmi náročný. Dalším negativním faktorem je, že objekty se ve scéně často překrývají, což celou situaci komplikuje. Na závěr je zde stereovidění vyžadující alespoň dva zdroje obrazu a vyznačující se vysokou výpočetní náročností.

Vzhledem k problémům jednotlivých přístupů se v přírodě vždy vyskytují kombinace jednotlivých metod. Pro zajímavost zmiňme příklady vyskytující se v přírodě. Např. žáby využívají zaostřování a stereovidění, holubi využívají stereovidění a určení hloubkové informace z pohybu a člověk využívá stereovidění spolu s kognitivními schopnostmi.

Souhrn jednotlivých vlastností lze nalézt v následující tabulce.

Metoda	Výhody	Nevýhody
Ohnisková vzdálenost	Jeden zdroj obrazu	Nutná optika s omezenou hloubkou pohledu
Hloubka z pohybu	Jeden zdroj obrazu	Nutná znalost rychlosti, slepý úhel
Kognitivní schopnosti	Jeden zdroj obrazu	Nutná schopnost učení a rozpoznávání, nízká robustnost
Stereoskopie	Robustnost, statická metoda	Dva zdroje obrazu, náročný výpočet

Tabulka 5.1

Jak již bylo řečeno, nejdůležitější metodou hloubkového rozpoznávání je metoda stereovidění, a proto je na ní zbytek této kapitoly zaměřen. Tato kapitola pokrývá základní aspekty stereovidění.

Metody stereovidění jsou založeny na vyhledávání shodných elementů ve zdrojových obrazech (proces stereo korespondence). Použití přímé intenzity jednotlivých pixelů obrazu pro vyhledávání shod není dobrým řešením, jelikož rozdíly nasnímané intenzity mezi použitými kamerami jsou poměrně značné. Tyto rozdíly jsou způsobeny drobnými odlišnostmi použitých CCD čipů, které nelze vyrobit absolutně shodné, dále světelnými podmínkami a šumem v obrazu. Rozdíly intenzity pak způsobují značné problémy při procesu párování pixelů v jednotlivých obrazech. Pro zajímavost uveďme, že Julesz [51] prováděl experimenty s náhodnými tečkovanými stereogramy a prokázal, že lidé jsou schopni rozpoznat hloubkovou informaci i přes problém s rozdíly intenzity v obou pohledech.

Proces stereo korespondence je velmi obtížný. Pokud je ve scéně obsaženo N bodů, které je nutné detekovat, pak musí být uvažováno N^2 možných shod. Mnoho těchto bodů je možné vyloučit vzhledem ke geometrickým omezením, ale i přes použité omezení je stále možné nalézt více než jedno správné řešení. Úloha je navíc komplikována existencí tzv. occlusion objektů. To jsou objekty ve scéně, které se promítají pouze na jednu z kamer, protože díky jinému úhlu pohledu ostatních kamer jsou tyto objekty částečně skryty za ostatními objekty ve scéně. Samozřejmě může dojít i ke skrytí daného objektu šumem vzniklým během snímání.

Naneštěstí přes veškeré snahy docílit v této oblasti pokroku, nebyla zatím oblast stereovidění dostatečně a jednoznačně popsána a standardizována. Tento fakt vede k výskytu velkého počtu metod výpočtu a stále se objevující nové metody. Tato kapitola se snaží popsat co největší počet v současnosti používaných metod.

V posledních dvaceti letech se obor robotiky zabývá studiem *existujících řešení* v přírodě a jejich realizací v technické praxi. V této kapitole si popíšeme fakta základního principu stereoskopie u lidí a zvířat. Jak již bylo popsáno dříve, jsou znalosti z této oblasti značně neucelené, a proto nemůže být v oblasti robotiky poskytnut kompletní model a definice

stereo vidění. Pro lepší pochopení problematiky je důležité zmínit některé aspekty vyskytující se v přírodě a využít tyto aspekty v návrhu jednotlivých robotických systémů.

Stereovidění je mechanismus, pomocí kterého je získána hloubková informace jednotlivých bodů v obrazu a to i navzdory nejednoznačnostem, které mohou nastat.

Umělé i přírodní stereoskopické systémy mají samozřejmě svá omezení a to zejména v oblasti rozpoznávané hloubky. Tato omezení vyplývají jednak z geometrie stereoskopického systému, dále pak z uspořádání snímacích senzorů a způsobu zpracování obrazové informace.

5.1. Omezení množiny bodů pro párování

Vzhledem k faktu, že metody stereovidění jsou výpočetně velice náročné, je nutné stanovit jistá omezení, oklešťující velikost prohledávaného prostoru možných párových bodů. Z jednotlivých možných omezení jmenujme následující:

1. **Epipolární omezení:** Bylo již zmíněno v předchozích kapitolách.
2. **Omezení podobnosti:** Silné a jednoznačné omezení, založené na předpokladu, kde pouze shodné vlastnosti mohou být spárovány mezi obrázky. Např. vzestupná hrana může být spárována pouze se vzestupnou hranou, jelikož směr hrany je vzhledem k použité projekci invariantní.
3. **Omezení kontinuity:** Objekty ve scéně jsou spojitě, proto se i disparita v hloubkové mapě musí měnit pouze spojitě, nikoli skokem. Tedy samozřejmě s výjimkou hranic objektů. Problémem ovšem je, že nejčastěji vyhledávané objekty jsou hrany, které tvoří hranice objektů. Tento fakt způsobuje omezení tohoto přístupu pouze na některé metody stereovidění.
4. **Omezení jednoznačnosti:** Každý význačný prvek v jednotlivých obrázcích může být přiřazen právě jednomu objektu v párovém obrázku. Problém samozřejmě nastává

s odlesky a occlusion objekty, které se vykytují pouze v obraze jedné z kamer a nemají párový ekvivalent.

5. **Omezení pořadí:** Toto omezení lze použít v systémech s rovnoběžnými optickými osami. Body ležící na levé straně od dané pozice, budou od této pozice ležet na levé straně i v druhém obraze.
6. **Omezení symetrie:** Vychází z následujícího předpokladu. Pokud se určitý objekt v levém obraze jeví jako shodný s prvkem v pravém obraze, pak toto pravidlo musí platit i naopak, tedy algoritmus musí být symetrický.

Zmíněné definice omezení nebyly nikdy formálně standardizovány, ale jsou široce používané v oblastech stereovidění. Prvky obrazu, které nesplňují tato kritéria, jsou vyloučeny z dalšího výpočtu, nebo jsou vyhodnocovány až v pokročilých stádiích výpočtu algoritmu.

5.2. Klasifikace metod stereovidění

Z pohledu implementace lze stereovidění rozdělit na dvě části:

1. Vyhledávání význačných částí.
2. Hledání stereo korespondence těchto částí.

Struktura lidského mozku naznačuje, že i lidé zpracovávají obrazovou informaci podobným způsobem. Existence mnoha zpětných vazeb mezi vyhledávání význačných částí a hledání stereo korespondence těchto částí však naznačuje, že se nejedná o striktně oddělené procesy, ale o procesy vzájemně propojené. Přes tento fakt, jsou procesy v robotice realizovány odděleně a samostatně.

5.3. Metody založené na hledání význačných míst

Tyto metody následují pravidla dekompozice celého procesu. Nejprve se extrahují projektivní invarianty, tedy oblasti, které jsou v obou obrazech neměnné.

Existuje velmi mnoho způsobů, jak vyhledávat objekty v obraze. Nejvíce používanou metodou je detekce hran, které je definováno jako strmá změna intensity v dané oblasti. Zmiňme např. Sobelův filtr, Canny-Derliche, Laplancián Gausián (LOG), Gábor Longos atd.

S ohledem na typ filtru a přítomnosti prahování mohou být vyhledávané vlastnosti numerické (binární, skalární, nebo vektory), nebo symbolické. V prvním případě mohou být algoritmy klasifikovány na korelační stereo metody, iterativní stereo metody a konstruktivní stereo metody. Bližší popis lze nalézt v následujících kapitolách.

5.3.1. Korelační stereo metody

Korespondence mezi dvěma vlastnostmi může být vypočtena numericky. Jednotlivé vlastnosti jsou korelovány a vypočtený koeficient je tím větší, čím jsou si vlastnosti podobnější. Samozřejmě tento přístup vyžaduje prověření všech možností a nalezení globálního, nikoli lokálního maxima.

Korelační stereo algoritmy jsou výhodné zejména z důvodu jejich jednoduchosti a neiterativního přístupu k výpočtu. Na druhou stranu, ale přinášejí velké množství chyb ve vypočtené hloubkové mapě a tyto chyby musí být minimalizovány prostřednictvím použití některých dříve zmíněných omezení a také pomocí následného zpracování vhodným filtrem. Dalším možností vylepšení řešení je hledat shody ve dvourozměrných výřezech na místo pouhých epipolárních čar. Tento způsob ovšem zpomaluje celý výpočet.

5.3.2. Iterativní stereo metody

Iterativní metody stereovidění kopírují metody korelační. Jako počáteční odhad je použit odhad vypočtený pomocí operace korelace mezi dvěma vstupními obrázky. V tomto případě je použito metod hledání globálního maxima. Vhodným postupem je iterativní optimalizační algoritmus, který v průběhu jednotlivých iterací vylučuje lokální maxima a iteruje k maximu globálnímu. Za tímto účelem je možné použít mnoho již existujících optimalizačních metod.

Optimalizační algoritmy také v některých případech úspěšně používají metod umělé inteligence. Zmiňme například Hopfieldovy neuronové sítě [52], simulované metody žíhání [53], buněčné neuronové sítě [54] a další.

Iterativní a kooperativní stereo algoritmy produkují velice dobré výsledky, ale jejich nevýhodou jsou dlouhé doby výpočtu a pomalá konvergence k řešení.

5.3.3. Konstruktivní stereo metody

Některé metody stereovidění jsou založeny na jiné optimalizační metodě a to metodě dynamického programování. Počáteční spárování je v tomto případě odhadnuto a to typicky aplikováním omezení pořadí na hranice obrázku a výsledné řešení je pak konstruováno iterováním a přidáváním možných shod v obrazech do předešlých řešení. Takto na místo počátku výpočtu se všemi možnými kombinacemi, jsou eliminovány špatné možnosti. Algoritmy této metody začínají s omezeným počátečním odhadem, který je v jednotlivých krocích doplňován a vylepšován. Výhodou metody je dosažení stabilního řešení a toto řešení je vždy blízkého k řešení optimálnímu.

5.4. Symbolické párování.

Jedná se o speciální případ metod vyhledávajících význačná místa v obraze. Kde tyto metody nepracují s nízko úrovněnými objekty, ale s objekty vysokoúrovněnými, jako jsou úsečky, jejich spojení a prohnutí. Protože je obtížné tyto vlastnosti jednoznačně identifikovat, byly pojmenovány zmíněným způsobem.

5.4.1. Metody založené na porovnávání intensity

Tyto metody pracují s přímým párováním založeným na intensitě jednotlivých obrazových bodů. Samozřejmě tento druh párování pixelu po pixelu produkuje extrémně velké množství špatných výsledků, a proto je nutné přidat doplňující informace. Kontextuální informace ve formě sousedících pixelů je použita k poskytnutí dodatečné informace, která slouží k diskriminaci vadných pixelů. Protože tyto metody mohou pracovat s dvourozměrným okolím bodu, jsou často přiřazeny do kategorie metod založených na detekci oblasti.

Metody tohoto druhu mohou být interpretovány jako částečný případ korelačních metod stereo algoritmů, kde je zkoumaná hodnota určena jako váhová funkce intenzity pixelů na okolí zkoumaného bodu. Stejně jako u korelačních metod je určen vektor vzdálenosti, který vyjadřuje podobnost okolí jednotlivých pixelů. Nejčastěji se využívá absolutní vzdálenosti, dále kvadrát vzdálenosti a také se velmi často používána Hammingova vzdálenost. Samozřejmě je možné použít i některé již zmíněné omezení.

Hlavní výhodou je produkce kompletní hloubkové mapy. Navíc za pomoci použití interpolačních technik je možné dosáhnout přesnosti výpočtu hloubkové mapy se sub-pixelovou přesností. Samozřejmě použití těchto přídavných metod prodlužuje dobu výpočtu.

Metody založené na hodnotě intensity jednotlivých pixelů mají také své nevýhody. Nejsou odolné vůči rozdílům snímání jednotlivých kamer. Tato nevýhoda omezuje použití metody na

specifické omezené podmínky a použití kvalitního hardware. Také studie týkající se rozpoznávání obrazu člověkem ukazují, že člověk těchto metod nevyužívá. Nicméně jsou tyto metody nejlépe realizovatelné a nejméně náročné na prostředky pro výpočet.

5.4.2. Metoda založená na fázovém posunutí FFT

Pokud nebereme v úvahu occlusion objekty, které jsou překryty jinými objekty, je pravý obraz posunutou a šumem ovlivněnou verzí levého obrazu. Posunutí pak určuje hloubku, po které pátráme. Zmíněný typ metod určuje disparitu založenou na teorému posunu Fourierovy transformace. Tento teorém je založen na faktu, že posunutí v prostoru, či času produkuje fázové posunutí ve frekvenční oblasti [55].

Metody založené na fázovém posunutí FFT produkují přímo hloubkovou mapu a to jako fázový rozdíl ve frekvenčním spektru levého a pravého obrazu. Přesto mají rozdílně vzdálené objekty rozdílná posunutí v obraze. Posuv tedy není konstantní přes celý obraz, a proto Fourierovu transformaci bez další úpravy není možné použít. Proto je nutné použít Gáborovu funkci [56], okénkovou Fourierovu transformaci, waveletovou transformaci, nebo Walshovy filtry [58].

Výhodou této metody je produkce stabilního řešení, neiterativní přístup a hloubková mapa celého obrázku, jelikož okénko Fourierovy transformace může být spočítáno pro libovolnou část obrazu. Dále tato metoda přináší možnost výpočtu se sub-pixelovou přesností, vzhledem k faktu, že posunutí fáze Fourierova obrazu není nutně omezeno na celá čísla. Na druhou stran jsou však tyto metody velice výpočetně náročné. Je nutné zmínit, že tato metoda je inspirovaná zpracováním obrazu člověkem, jelikož byly nalezeny souvislosti, že neurony v oblasti vidění zpracovávají signály obdobným způsobem.

6. Realizace metody stereo vidění

V této kapitole lze nalézt popis základního principu stereovidění. Dále je zde popsán způsob autorovy realizace metody stereovidění vycházející z dynamického programování, založené na metodě popsané v [57]. Vzhledem k faktu, že při výpočtu mapy disparity dochází k velkému množství chyb, byla použita metoda upravena přidáním filtru na výsledné mapě disparity, který úspěšně minimalizuje některé druhy vzniklých chyb. Popis, návrh a realizaci zmíněného filtru lze nalézt v této kapitole. Metoda stereovidění byla také experimentálně rozšířena o použití počátečního odhadu získaného na základě detekce vizuálních značek ve snímané scéně.

6.1. Úvod

Úkolem metody stereovidění je ze vstupních obrazů vytvořit mapu disparity. Jedná se o matici D rozměru výšky a šířky vstupních obrazů v pixelech, kde hodnota prvku $D_{x_A y_A}$ odpovídá vzdálenosti pixelu o poloze (x_A, y_A) v prvním obraze od jeho párového pixelu v obraze druhém.

Jako vstup použité metody stereovidění uvažujeme dvojici rektifikovaných obrazů (viz kapitola 4). Principem metody je tedy nalezení shodných pixelů v obou obrazech a následném určení mapy disparity. Vzhledem k faktu, že vstupní obrazy jsou rektifikovány, je vyhledávání omezeno pouze na řádky. Jedná se tedy o operaci, jejímž vstupem je pouze shodný řádek levého a pravého obrazu (prohledávaný prostor tak je výrazně zúžen).

Pixely z jednoho z řádku prvního obrazu jsou explicitně spárovány s pixely shodného řádku v druhém obraze. Tato korespondence je vyjádřena jako sekvence shod S , kde je každá shoda vyjádřena jako uspořádaná dvojice pixelů (x_A, x_B) . V tomto případě x_A vyjadřuje polohu bodu na aktuálně prohledávaném řádku v prvním obraze a x_B analogicky vyjadřuje polohu bodu na

shodném řádku v druhém obraze. Jedná se o souřadnici sloupce, ve kterém se pixel nachází. Jelikož se jedná o operaci prováděnou pouze na řádcích obrazu, nemá smysl informaci o čísle řádku uvažovat. Shoda dvou pixelů je vyjádřena jako maximální podobnost intenzity pixelů v obou obrazech vyjádřených pomocí funkcí $I_A(x_A)$ a $I_B(x_B)$.

Je nutné počítat s výskytem nepárových pixelů. Tyto pixely vznikají v důsledku překrytí dalšími objekty ve scéně, přítomnosti šumu v obraze, nebo v důsledku odrazů. Pro daný pixel v jednom z obrazů tedy neexistuje párový pixel v obraze druhém. Pro sekvenci těchto pixelů budeme pro zjednodušení používat označení occlusion objekty. Extrémní případ takovéto sekvence je uveden na Obr. 6.1. Lze zde nalézt pohledy na stejnou scénu ze dvou kamer, jejichž poloha i orientace je velice rozdílná. Všimněte si, že jeden z vizuálních terčů je celkově skrytý.



Obr. 6.1: Příklad occlusion objektů

Hodnota disparity $\delta(x_A)$ pixelu x_A je definována jako rozdíl souřadnic x_A a x_B odpovídajících pozicím párových pixelů v obou obrazech:

$$\delta(x_A) = x_A - x_B \quad (6.1)$$

Hodnotu disparity occlusion objektů (sekvence nepárových pixelů) není možné určit. Proto jí je přiřazena hodnota nejbližšího sousedícího objektu, kde je ještě možné ji vyjádřit.

6.2. Princip metody stereovidění

V této části naleznete stručný popis principu použité metody stereovidění. Metoda byla převzata z [58]. Součástí popisu jsou také důvody, pro které byla právě tato metody vybrána pro použití v oblasti robotiky.

6.2.1. Hodnotící funkce

Spolu se zavedením sekvence shod pixelů S je nutné zavést způsob, jakým je ohodnocena kvalita této sekvence. Zavedeme tedy hodnotící funkci, která vyjadřuje jak daná sekvence shod pixelů S popisuje pravou korespondenci jejích jednotlivých pixelů. Hodnotící funkce $\gamma(S)$ sekvence shod pixelů S je definována konstantní penaltou pro každý occlusion objekt, konstantní odměnou pro každou shodu a také sumou rozdílů porovnávaných pixelů. Hodnotící funkce má tedy následující tvar:

$$\gamma(S) = P_O\mu_O - P_S\mu_S + \sum_{i=1}^{P_S} d(x_{Ai}, x_{Bi}) \quad (6.2)$$

Kde μ_O je konstantní penalta pro occlusion objekt, μ_S je konstantní odměna pro shodu, $d(x_{Ai}, x_{Bi})$ vyjadřuje rozdíl mezi hodnotou intensity dvou pixelů nacházejících se na definovaných souřadnicích x_{Ai} a x_{Bi} . Konstanta P_O definuje počet occlusion objektů a konstanta P_S definuje počet shod v sekvenci S .

Takto navržená hodnotící funkce preferuje oblasti s konstantní disparitou. Pokud je to jen možné, je každému objektu přiřazena konstantní hodnota disparity, přesto že se disparita objektu mění (jako například v případě válcových ploch). Tuto vlastnost však ospravedlňuje dostatečná schopnost rekonstrukce scény. Dále usnadňuje přesnější výpočet hloubkové informace, protože vyzdvihuje změnu hodnoty disparity na hranicích objektů. Navíc zjednodušení hodnotící funkce utváří hodnotící funkci jednoduchou na pochopení, implementaci i zejména na výpočet.

Konstanta μ_O je interpretována jako množství indicií, nezbytných k definování velikosti změny hodnoty disparity na hranicích objektů (myšleno vzhledem k definici shody). Zatímco konstanta odměny za shodu μ_S je definována jako maximální velikost rozdílů pixelů, která je běžně předpokládána mezi dvěma navazujícími pixely. Dohromady tyto konstanty vystupují jako celková penalizace za occlusion objekty.

6.2.2. Vyjádření odlišnosti pixelů

Výraz $d(x_{Ai}, x_{Bi})$ hodnotí, jak rozdílné jsou intenzity pixelů $I_A(x_{Ai})$ a $I_B(x_{Bi})$ v levém a pravém obrazu. Tato odlišnost však nemůže být vzata jako pouhý rozdíl jasů, protože postup digitalizace obrazů pomocí kamer způsobuje drobné odlišnosti v hodnotách intenzit $I_A(x_{Ai})$ a $I_B(x_{Bi})$ a to i přes použití přibližně shodných a vysoce kvalitních zařízení. Typicky je tento problém řešen použitím okénkové metody prohledávání, nebo za pomoci použití sub-pixelové přesnosti, kde nejde o hodnotu samotného pixelu, ale v úvahu se bere průběh jasové funkce na okolí daného pixelu. Problém však zůstává, že tyto způsoby řešení přináší velkou výpočetní náročnost, které si bohužel v oblasti robotiky nemůžeme dovolit.

Právě důvod rychlého výpočtu, který je v oblasti robotiky nutně vyžadován, nás donutil vybrat takovou metodu stereovidění, která využívá lineární interpolace jasové funkce na okolí dvou pixelů ke změření míry jejich odlišnosti. Tento způsob je odolný vůči rozdílům intenzity

obou obrazů vzniklých v průběhu digitalizace a je stále výpočetně nenáročný. Více informací lze nalézt v [57].

6.3. Možnosti vylepšení metody

Pro zvýšení rychlosti výpočtu a kvality výstupních dat se zaměřujeme na následující dvě kategorie možných omezení:

1. Omezení usnadňujících systematické a efektivní vyhledávání.
2. Omezení zlepšující funkci algoritmu na netexturovaných regionech.

V následujících podkapitolách budou obě kategorie omezení blíže popsány.

6.3.1. Omezení usnadňujících systematické a efektivní vyhledávání

Jak již bylo uvedeno na začátku kapitoly, je s výhodou použito epipolárního omezení, které omezuje prohledávaný prostor na vyhledávání podél shodných řádků v obou obrazech.

Dále je, stejně jako u mnoha jiných algoritmů stereovidění, nutné přidat omezení prohledávané oblasti. Jedná se o omezení disparity pouze na interval $0 \leq \delta \leq \delta_{MAX}$, kde δ_{MAX} vyjadřuje maximální možnou disparitu.

6.3.2. Omezení zlepšující funkci algoritmu na netexturovaných regionech

Vzhledem k nejednoznačnosti v oblastech s nedostatkem textury, vyžaduje použitý algoritmus přítomnost textury v celém obrázku. Samozřejmě by bylo ideální každou

nasnímanou scénu uměle doplnit o dostatečně texturované povrchy. Toto řešení je však velice komplikované a bohužel ho není možné provést vždy. Jednou z možností jak tento problém řešit je nalezení takových vlastností vstupních obrazů, které umožní doplnit informace o disparitě v oblastech s nedostatečnou, nebo žádnou texturou.

Podle velikosti změny hodnoty intenzity lze oblasti obrazu rozdělit do dvou následujících kategorií:

1. **Oblasti se skokovou změnou intenzity:** Jedná se zejména o hrany objektů. V těchto oblastech lze předpokládat skokovou změnu hodnoty disparity, vzhledem k přechodům mezi objekty.
2. **Oblasti se spojitou změnou intenzity:** Lze předpokládat, že v oblastech se spojitou změnou intenzity dochází také ke spojitě změně disparity.

Dále jsme schopni každému prvku matice disparity přiřadit hodnotu, která vyjadřuje jeho věrohodnost. Přiřazujeme následující dvě hodnoty:

1. **Vysoká věrohodnost vypočtené disparity:** Je přiřazena pixelům, v jejichž okolí se hodnota disparity nemění skokem. Vycházíme z předpokladu, že snímaná scéna je spojitá, tedy i vypočtená disparita se musí měnit spojitě.
2. **Nízká věrohodnost vypočtené disparity:** Je přiřazena pixelům, jejichž disparita vykazuje skokové odchylky vůči prvkům v blízkém okolí.

Využitím těchto vlastností jsme schopni potlačit chyby a doplnit informaci o disparitě v místech, které použitý algoritmus nebyl schopen určit (jako jsou např. occlusion objekty a nedostatečně texturované plochy).

6.4. Propagace informací mezi řádky

Jak bylo naznačeno v kapitole 6.3 je nutné rozšířit použitý algoritmus takovým způsobem, aby nezpracovával pouze informaci ze samostatných řádků, ale uvažoval i jejich okolí. Možným řešením je rozšíření jednorozměrné hodnotící funkce na funkci dvourozměrnou. Bohužel však takovéto rozšíření způsobuje výrazné zpomalení celého výpočtu.

Z tohoto důvodu jsme se rozhodli použít odlišného přístupu, který nezpůsobí tak výrazné zpomalení výpočtu. Jedná zpravování výsledné mapy disparity pomocí filtru. S jeho pomocí lze přenést hodnoty disparity z oblastí, kde je schopen algoritmus určit její hodnotu s vysokou věrohodností do oblastí, kde mohlo dojít k výskytu chyb. Tento způsob řešení je velice rychlý a navyšuje celkovou dobu výpočtu pouze minimálně.

Jak již bylo naznačeno v kapitole 6.3.2, je možné každému pixelu jednoduše přiřadit úroveň věrohodnosti vypočtené hodnoty disparity. Tato hodnota je ovlivněna dvěma faktory:

1. Velikostí změny intenzity na okolí daného pixelu.
2. Počtem pixelů, u kterých se hodnota disparity nemění skokem. Uvažujeme pixely ve sloupci nad, nebo pod daným pixelem. Sloupce jsou uvažovány z důvodu funkce algoritmu po samostatných řádcích a snahou rozšířit algoritmus z výpočtu po řádcích na celou oblast obrazu. Pokud si vzájemně odděleně vypočtené řádky přibližně odpovídají, pak je možné usuzovat, že byly vypočteny korektně a věrohodnost vypočtené disparity je vysoká.

Hodnoty disparity pixelů s vysokou věrohodností jsou promítnuty ve svém sloupci, tak že nahrazují hodnoty disparity pixelů s nízkou věrohodností vypočtené disparity, ke kterým dorazí, dokud nedojde ke skokové změně intenzity (způsobené např. přítomností hrany).

Potom co je tento filtr uplatněn na sloupcích, je analogickým způsobem proveden na řádcích již upravené mapy disparity. Tento krok zaručuje vyhlazení mapy disparity a odstranění chyb v podobě izolovaných výčnělků.

6.5. Generování počátečního odhadu

Jak již jsme nastínili, jsou metody stereo vidění velice závislé na hlavním omezení a tím je maximální a minimální velikost disparity. V případě velkých hodnot rozdílu maximální a minimální disparity, výpočetní časy extrémně narůstají. Tento fakt komplikuje situaci zejména v případech, kdy nemůžeme, nebo nechceme umístit snímací kamery blízko sebe. Dále je problémem i omezený prostor výpočtu mapy disparity, což v některých případech může být nedostačující. Pokud ovšem vezmeme v úvahu, že jsme schopni nějakým způsobem ovlivnit snímanou scénu ramene robotu, a podmínky v tomto systému přizpůsobit, jsme schopni nalézt řešení zmíněných problému. Řešením je přidání jednoznačně rozpoznatelných značek do snímané scény, s jejichž pomocí je možné okamžitě rozpoznat jejich trojrozměrnou polohu a orientaci a na tomto základě pak generovat počáteční odhad pro metodu stereovidění.

6.5.1. Vizuální značky

Počítačové vidění je schopno jednoznačně určit přesnou polohu definované značky ve scéně. Výhodou tohoto postupu je nízká výpočetní náročnost, tedy schopnost provést danou operaci v reálném čase. Další zřejmou výhodou je odolnost celého systému, kdy algoritmy počítačového vidění jsou schopny danou značku, nebo terčik vyhledat i za velice špatných světelných podmínek.

Metody detekce vizuálních značek byly vytvořeny zejména z důvodů využití v robotice, ale také např. v „*augmented reality*“ atd. Jejich primární úlohou je eliminovat počty špatně detekovaných terčiků a také eliminovat množství záměn jednotlivých terčiků ve scéně.

Terčíky jsou pak většinou použity ve větším množství, aby se tak zajistilo, že pokud se bude snímací kamera pohybovat, bude vždy některý z terčků viditelný. S rostoucím počtem terčků se také zvyšuje přesnost celého systému.

6.5.2. Kategorizace vizuálních značek

V zásadě rozlišujeme dvě hlavní kategorie terčků (viz Obr. 6.2):

1. vytištěné dvourozměrné vzory
2. prostorově orientované útvary

Zmiňme kritéria, kterými lze vizuální terčíky definovat:

1. **Chybná detekce přítomnosti:** Vykazuje míru chyby detekce terčiku, pokud se ve scéně vůbec nevyskytuje.
2. **Možnost záměny jednotlivých terčků:** Vyjadřuje pravděpodobnost záměny dvou terčků.
3. **Možnost nenalezení terčiku:** Vyjadřuje pravděpodobnost, že terčik vyskytující se ve scéně, nebude vůbec nalezen.
4. **Minimální velikost terčiku:** Vyjadřuje minimální velikost terčiku v pixelech, která zajistí jeho detekci.
5. **Možnost variací:** Definujeme maximální počet jednoznačně odlišitelných terčiku ve snímané scéně.
6. **Rychlost výpočtu:** Definuje výpočetní náročnost metody, která je nutná k rozpoznání použitého typu terčiku. Důležitým faktorem je detekce značek v reálném čase.
7. **Odolnost vůči změnám světelných podmínek:** Definuje omezení světelných podmínek, za kterých je terčik ještě možné identifikovat.
8. **Odolnost vůči překrytí:** Definuje, zda metoda je schopna terčik identifikovat i v případě jeho částečného překrytí jiným objektem.



Obr. 6.2: Hlavní kategorie vizuálních značek

Při tvorbě vizuální značky jsou striktně stanoveny její fyzické parametry. V případě vytisknuté značky jde o její přesnou velikost a strukturu, v případě prostorového terčíku, pak jde o jeho prostorovou orientaci a vzdálenosti jednotlivých elementů.

Díky těmto informacím jsme schopni určit trojrozměrnou polohu a orientaci již s použitím jediné kamery. S použitím více kamer se zvyšuje přesnost metody. Kromě trojrozměrné polohy a orientace je také možné detekovat jedinečný identifikátor obsažený ve vizuálním terčíku a jednoznačně ho ve scéně rozpoznat.

6.5.3. Využití vizuálních značek

Pokud ve snímané scéně použijeme vizuální značku, jsme schopni okamžitě identifikovat její trojrozměrnou polohu a orientaci ve scéně. Pro použití v robotice lze tak snadno v reálném čase rozpoznat objekty, jejichž vyhledání by jinak bylo pomocí metod počítačového vidění zdlouhavé a komplikované.

Další možností, která se nabízí, je vytvoření počátečního odhadu pro urychlení a zefektivnění běhu metod stereo vidění. Do scény umístíme na vhodné rovinné plochy určitý počet vizuálních značek. V každém bodě tyto značky identifikujeme a určíme jejich trojrozměrnou polohu, orientaci a měřítko. S pomocí takto zjištěných údajů lze vytvořit počáteční odhad

disparity obrazu, kde v oblastech výskytu vizuálních značek dojde k doplnění známých hodnot disparity. Vzhledem k faktu, že hodnota disparity se na většině míst mění spojitě, lze hodnotu disparity doplnit i do okolí známých poloh vizuálních terčů. Metoda tak výrazně urychluje výpočet a to zejména ve scénách s velkou plochou označených rovinných ploch.

Je nutné upozornit, že se tato práce nezabývá řešením problému detekce vizuálních značek ve snímané scéně. Pro jejich detekci bylo použito vývojové SDK Unifeye firmy Metaio.

6.6. Experimentální výsledky

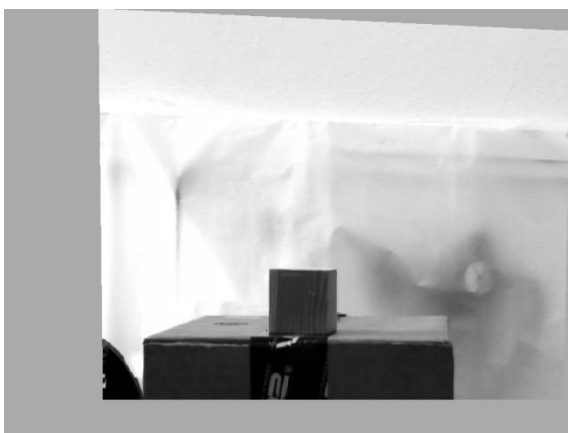
V této části lze nalézt výstupy metody stereoidění pro několik rozdílných typů úloh. Součástí každé úlohy je vždy dvojice vstupních rektifikovaných obrazů, průběhy jasové funkce $I(x)$ na dvou vybraných párových řádcích rektifikovaných obrazů a výsledná mapa disparity. Na konci kapitoly lze nalézt případ, který jako svůj počáteční odhad používá odhad založený na základě detekce vizuálních značek spolu se srovnáním rychlosti výpočtu oproti klasické metodě bez počátečního odhadu.

Veškeré dvojice obrázků, které byly zpracovány v následujících podkapitolách, měly rozlišení 800x600 pixelů a následně byly rektifikovány do obrázků o rozměru 650x488 pixelů. Maximální hodnota disparity jako vstup algoritmu byla nastavena na poměrně vysokou hodnotu a to 70. Experimentálně byly zvoleny následující hodnoty parametrů metody stereoidění: penalizace za occlusion objekty $\mu_O = 40$ a konstanta odměny za shodu $\mu_S = 13$. Pro tyto parametry vykazovala použitá metoda nejlepší výsledky.

Je nutné upozornit, že výpočet byl prováděn na notebooku Asus A6Ja s procesorem Intel Core Duo T2400 1,83GHz, 1GB paměti RAM a operačním systémem Windows Vista. Kód byl z části řešen v řízeném jazyce C#, implementovaný pro platformu .NET, kde výpočetně náročné části byly řešeny v neřízeném jazyce C++ jako DLL knihovna pro platformu Win32.

6.6.1. Příklad detekce jednoduché scény

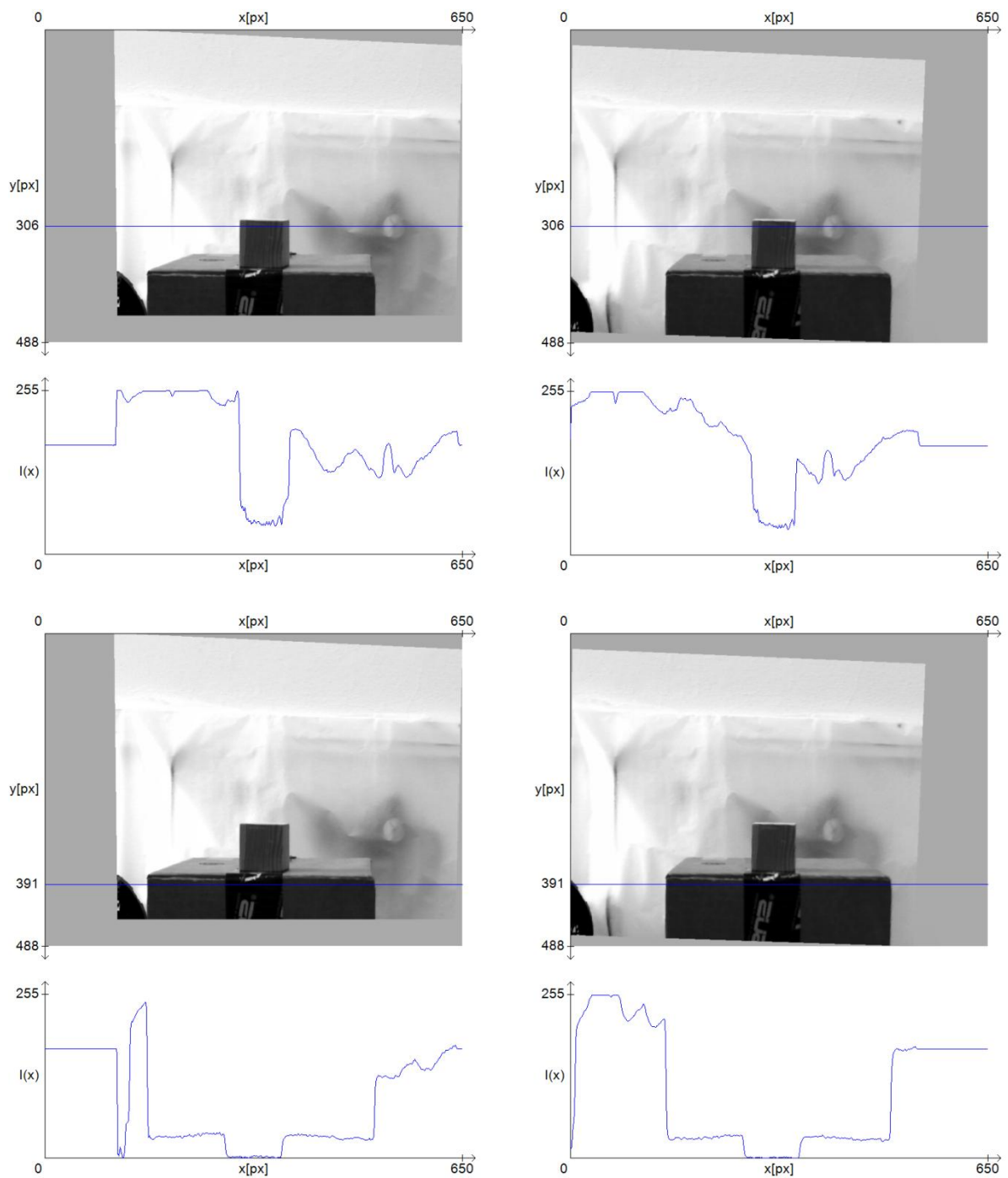
Na Obr. 6.3 lze nalézt dvojici rektifikovaných obrázků. Obr. 6.4 obsahuje výslednou mapu disparity, kde světlejší barva značí vyšší hodnotu disparity a odpovídající vzdálenosti objektu. Na obrázku Obr. 6.5 pak průběhy jasové funkce na párových řádcích rektifikovaných obrazů. Z průběhu jasových funkcí je patrné, že se pozadí obou obrazů poměrně liší, přesto algoritmus vyprodukoval velice kvalitní mapu disparity. Doba výpočtu trvala 24.7s.



Obr. 6.3: Dvojice rektifikovaných obrazů



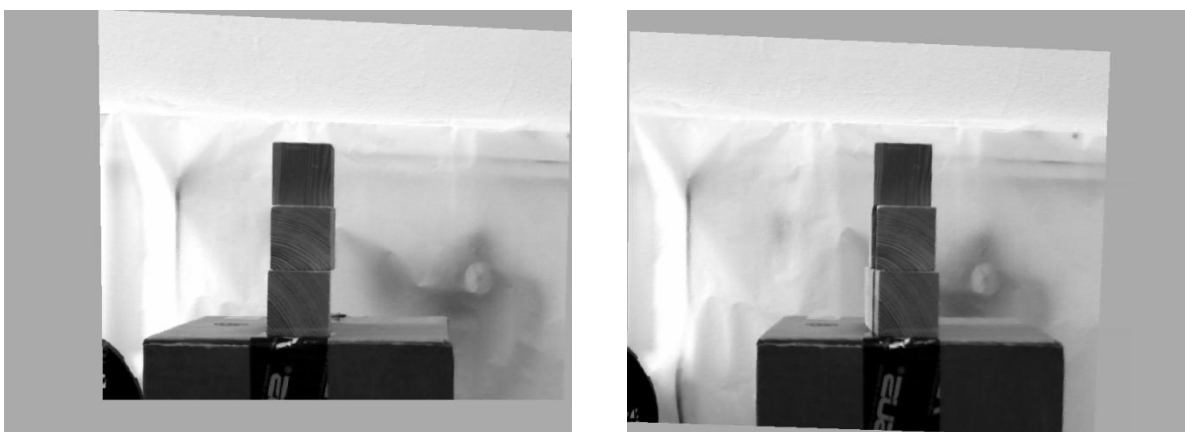
Obr. 6.4: Výsledná mapa disparity



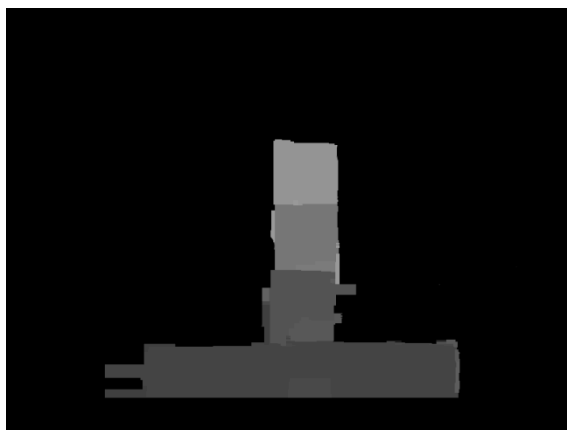
Obr. 6.5: Průběh jasové funkce na párových řádcích rektifikovaných obrazů

6.6.2. Příklad detekce složené nepřekrývající se scény

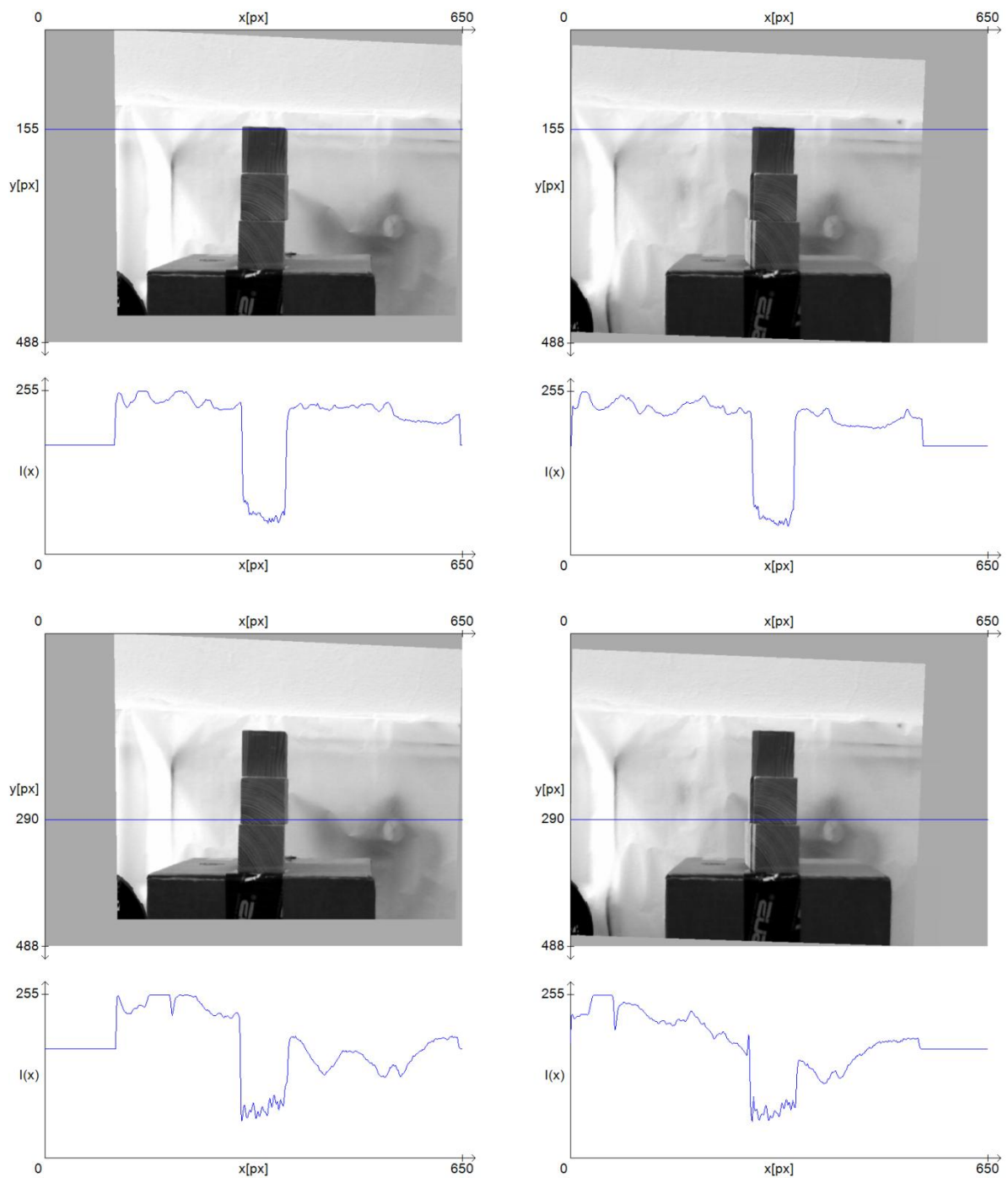
Na následujícím obrázku Obr. 6.6 lze nalézt dvojici rektifikovaných obrazů. Dále na Obr. 6.7 výslednou mapu disparity, kde světlejší barva značí vyšší hodnotu disparity a odpovídající vzdálenosti objektu a na Obr. 6.8 pak průběhy jasové funkce na vybraných párových řádcích rektifikovaných obrazů. Doba výpočtu trvala 22.3s.



Obr. 6.6: Dvojice rektifikovaných obrazů



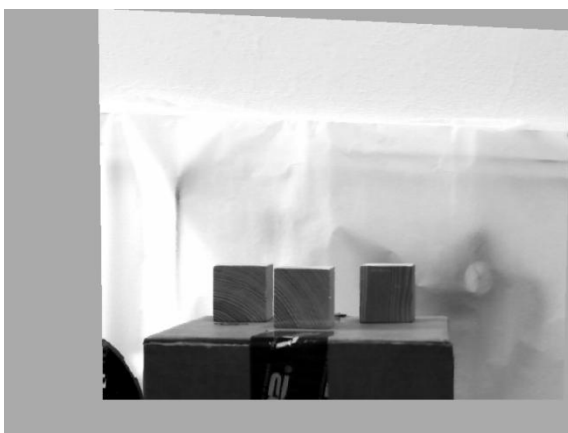
Obr. 6.7: Výsledná mapa disparity



Obr. 6.8: Průběh jasové funkce na párových řádcích rektifikovaných obrazů

6.6.3. Příklad detekce složené překrývající se scény

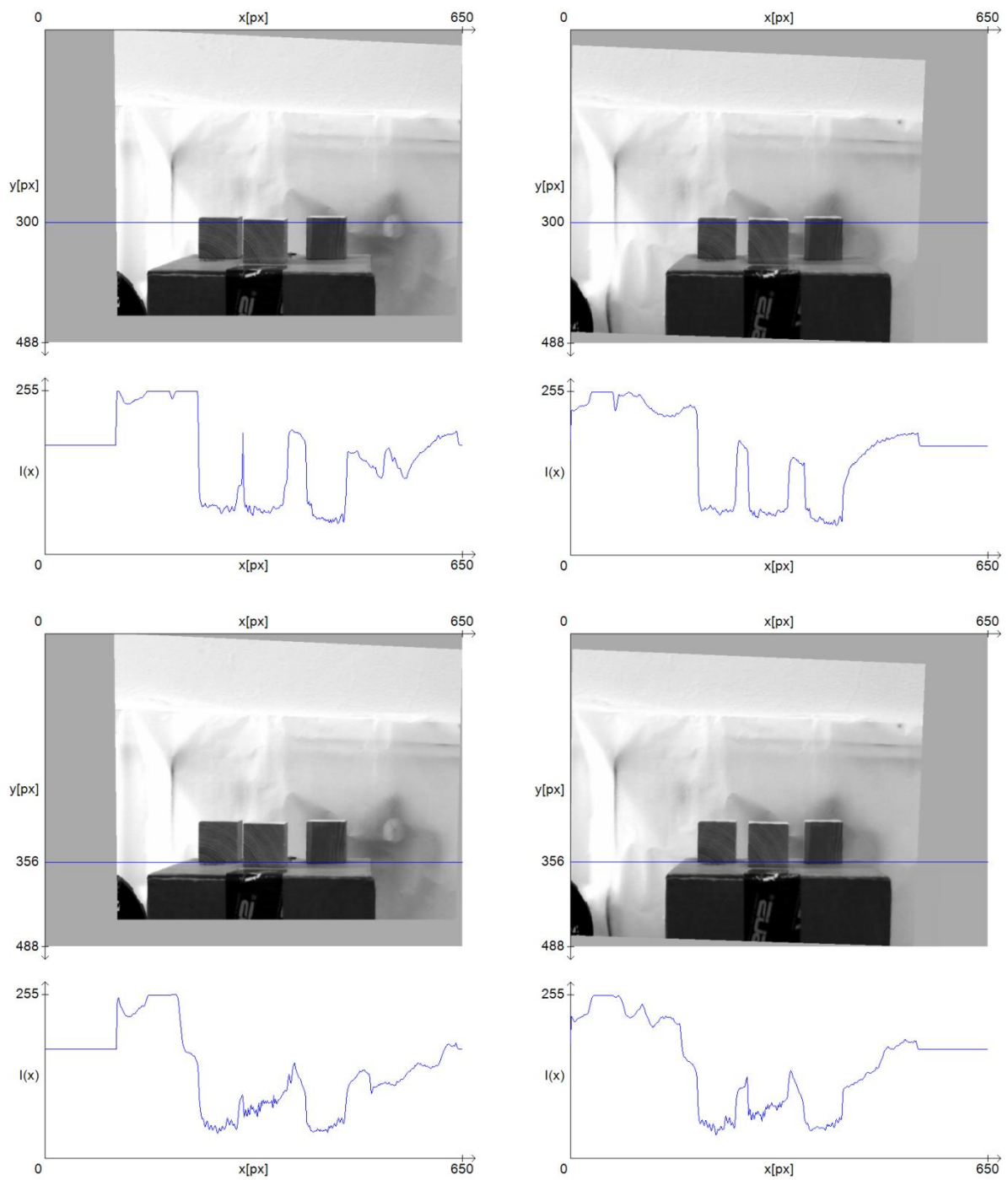
Na Obr. 6.9 lze nalézt dvojici rektifikovaných obrázků. Dále na Obr. 6.10 výslednou mapu disparity, kde světlejší barva značí vyšší hodnotu disparity a odpovídající vzdálenosti objektu. Na Obr. 6.11 jsou obsaženy průběhy jasové funkce na párových řádcích 300 a 356 rektifikovaných obrazů. Z výsledků je patrná, že i přes problém překrývající se objektů byl algoritmus schopný nalézt poměrně dobré řešení. Doba výpočtu trvala 20.5s.



Obr. 6.9: Dvojice rektifikovaných obrazů



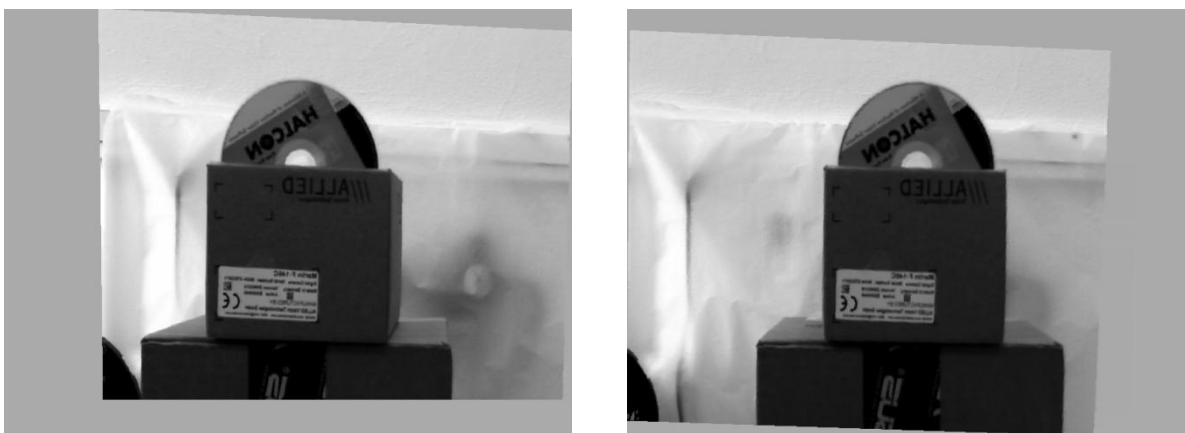
Obr. 6.10: Výsledná mapa disparity



Obr. 6.11: Průběh jasové funkce na párových řádcích rektifikovaných obrazů

6.6.4. Příklad detekce kruhové plochy

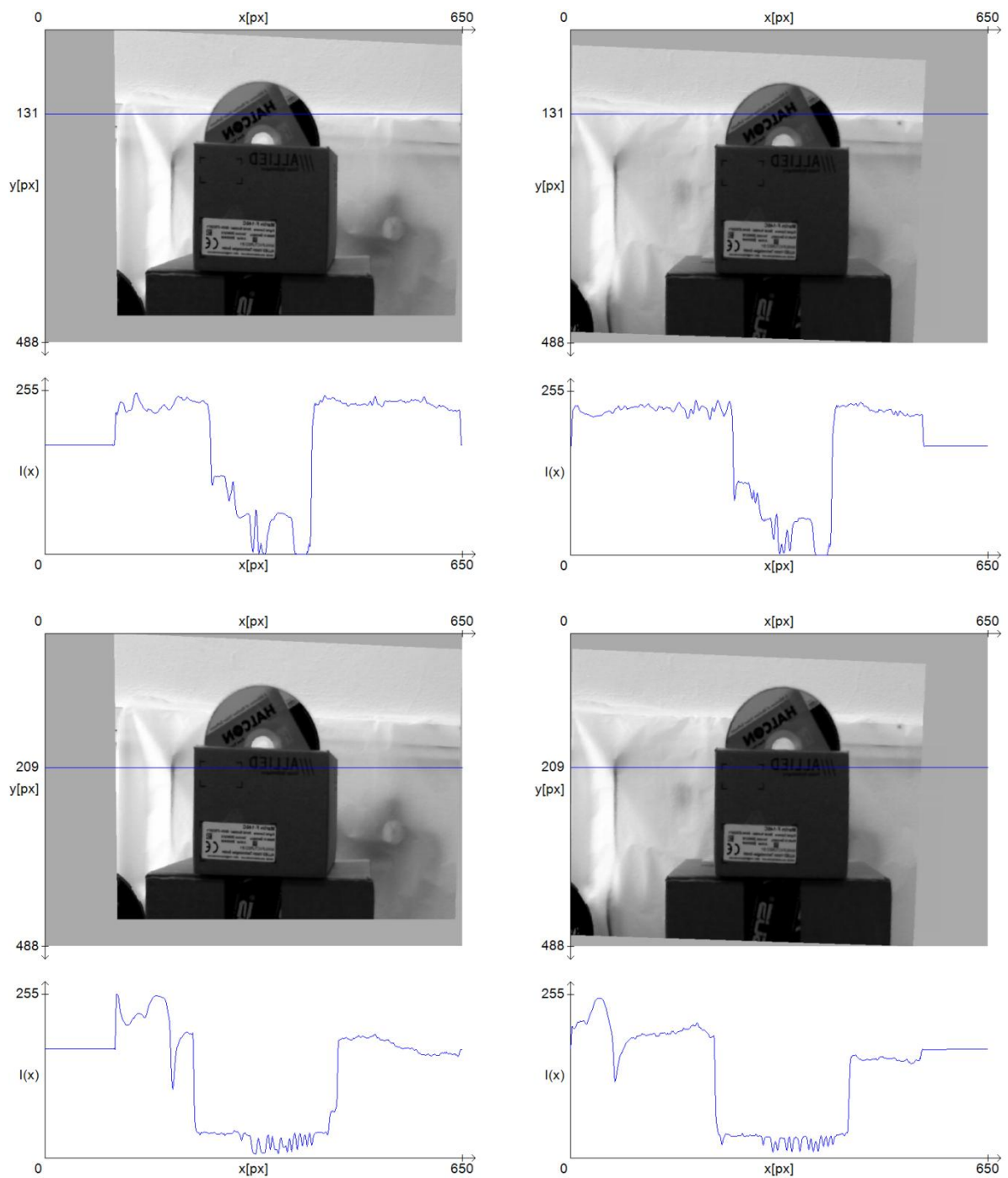
Na následujícím obrázku Obr. 6.12 lze nalézt dvojici rektifikovaných obrázků. Dále na Obr. 6.13 výslednou mapu disparity, kde světlejší barva značí vyšší hodnotu disparity a odpovídající vzdálenosti objektu. Na Obr. 6.14 pak průběhy jasové funkce na vybraných párových řádcích rektifikovaných obrazů. Lze zjistit, že i přes nedostatečně texturovanou scénu byl algoritmus schopen korektně doplnit mapu disparity. Doba výpočtu trvala 20.7s.



Obr. 6.12: Dvojice rektifikovaných obrazů



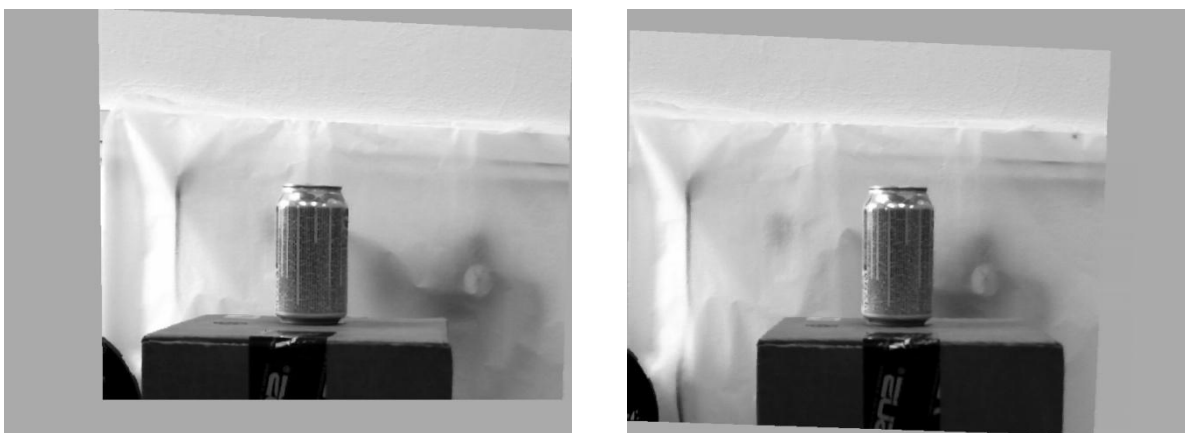
Obr. 6.13: Výsledná mapa disparity



Obr. 6.14: Průběh jasové funkce na párových řádcích rektifikovaných obrazů

6.6.5. Příklad detekce válcové plochy

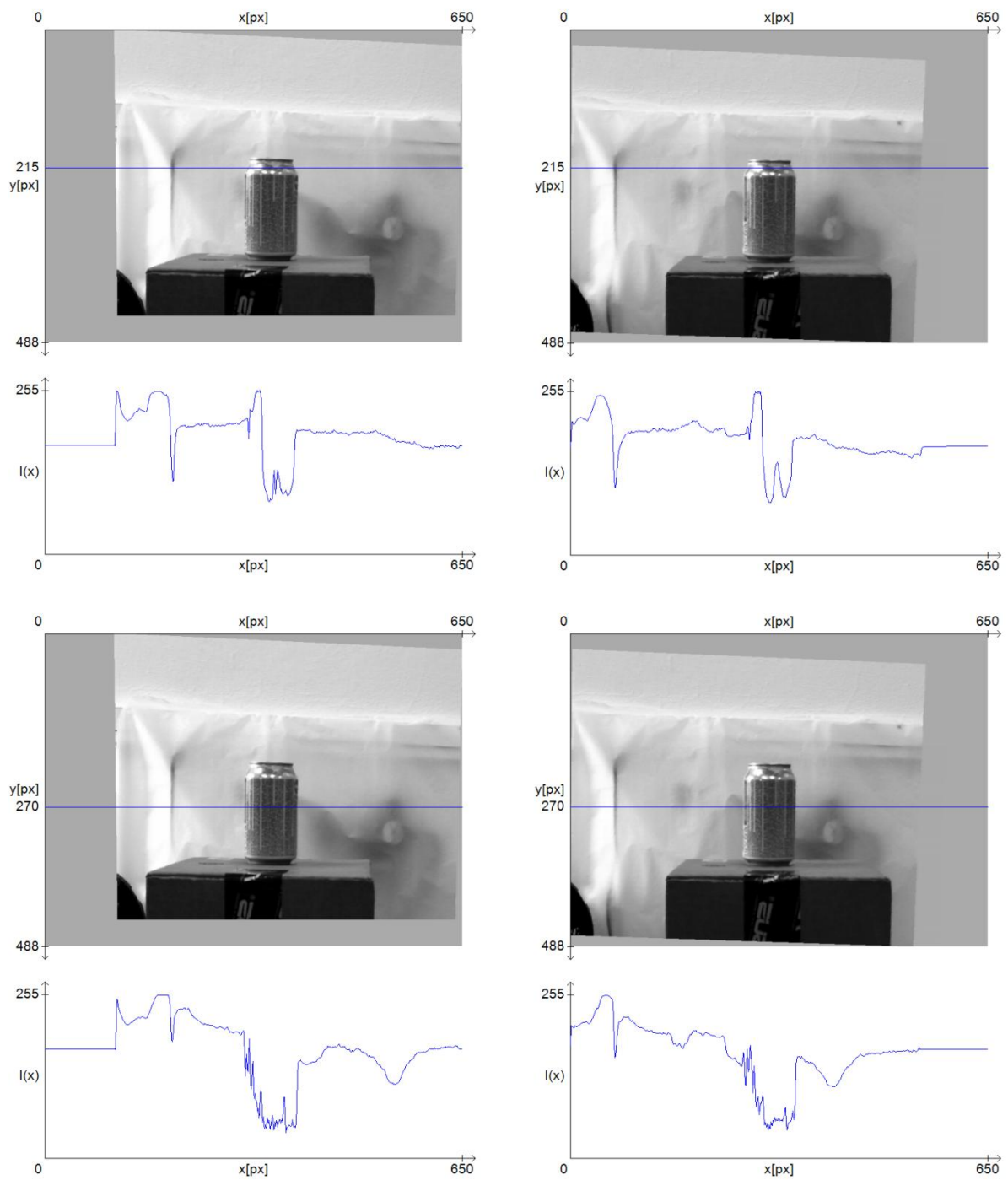
Na následujícím obrázku Obr. 6.15 lze nalézt dvojici rektifikovaných obrázků. Dále na Obr. 6.16 výslednou mapu disparity a na Obr. 6.17 pak průběhy jasové funkce na vybraných párových řádcích rektifikovaných obrázků. Z průběhu jasové funkce je možné zjistit odlišnosti způsobené odlesky na ploše plechovky. Z mapy disparity lze rozeznat dříve zmíněný charakter metody, kde je snaha celému objektu přiřadit jednu hodnotu disparity. Výpočtu trval 21.4s.



Obr. 6.15: Dvojice rektifikovaných obrázků



Obr. 6.16: Výsledná mapa disparity



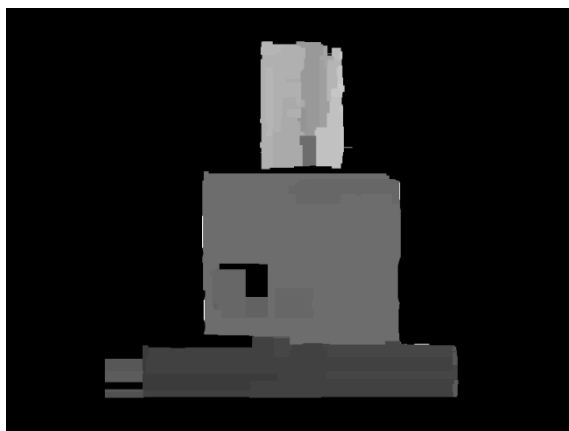
Obr. 6.17: Průběh jasové funkce na párových řádcích rektifikovaných obrazů

6.6.6. Příklad detekce komplikované scény

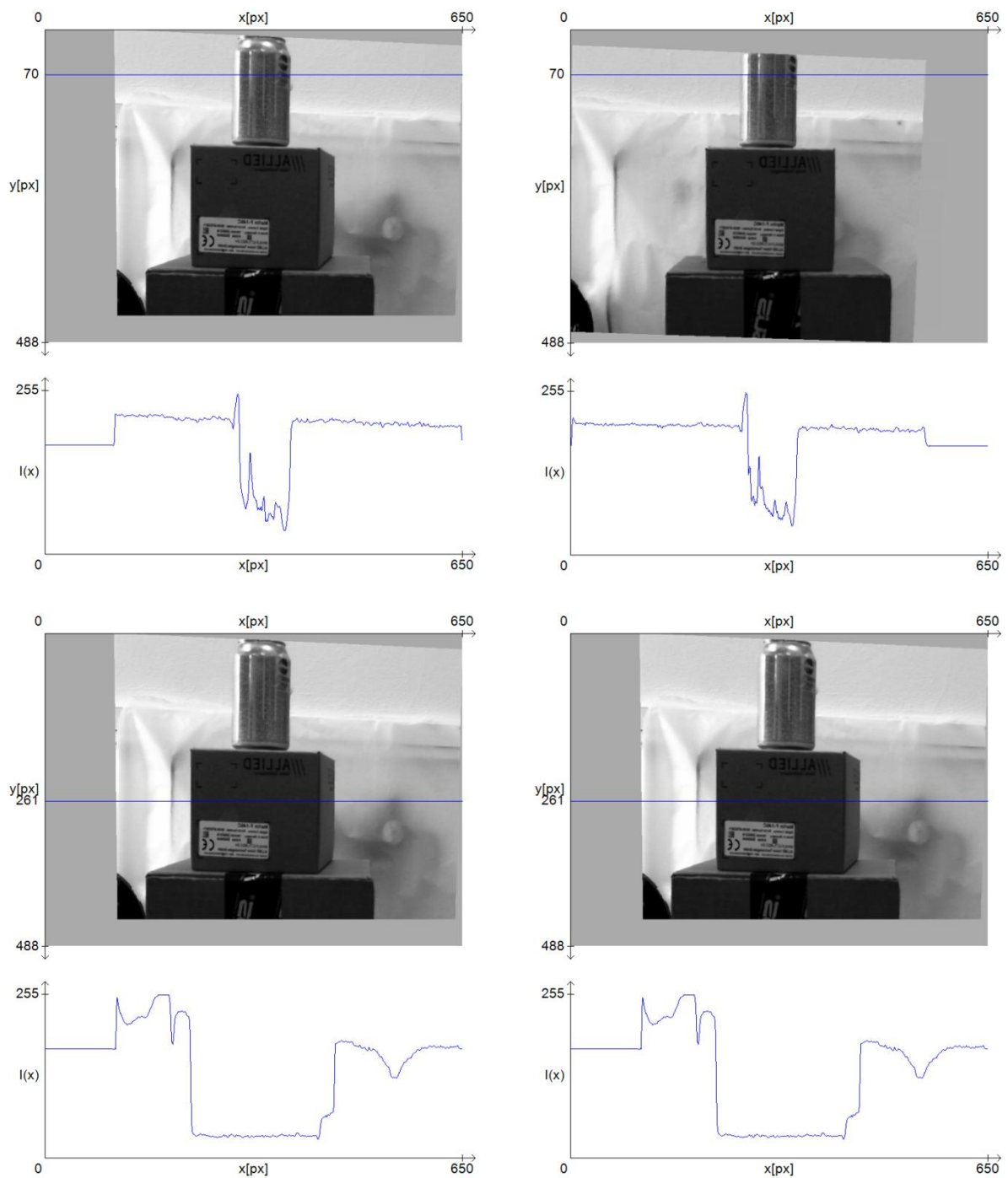
Na následujícím obrázku Obr. 6.18 lze nalézt dvojici rektifikovaných obrázků. Dále na Obr. 6.19 výslednou mapu disparity, kde světlejší barva značí vyšší hodnotu disparity a odpovídající vzdálenosti objektu. Na Obr. 6.20 pak průběhy jasové funkce na vybraných párových řádcích rektifikovaných obrazů. V tomto případě se příliš neprojevily negativní odlesky a algoritmus vyprodukoval poměrně kvalitní výstup. Doba výpočtu trvala 22.5s.



Obr. 6.18: Dvojice rektifikovaných obrazů



Obr. 6.19: Výsledná mapa disparity

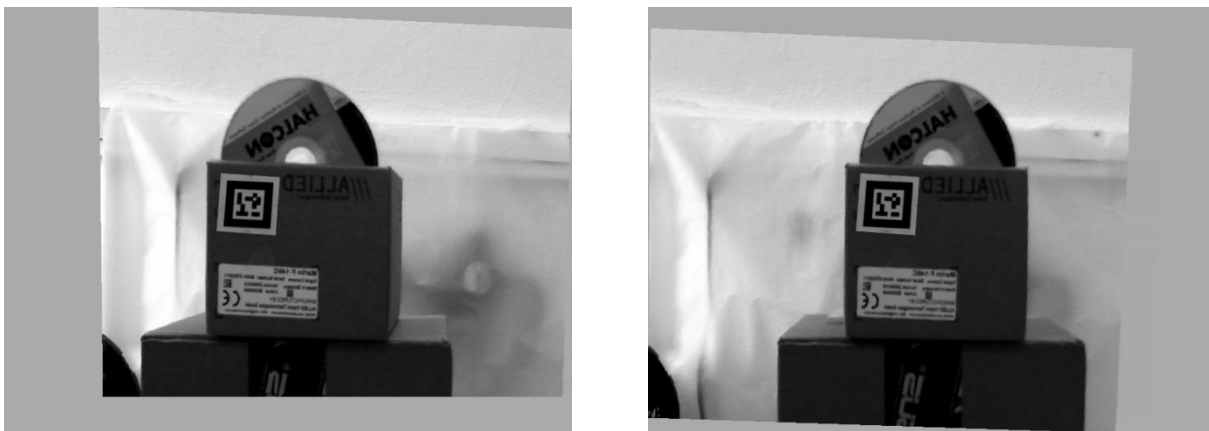


Obr. 6.20: Průběh jasové funkce na párových řádcích rektifikovaných obrazů

6.6.7. Příklad generování počátečního odhadu pomocí vizuálních značek.

Jedná se o naprosto stejný případ, jako v části 6.6.4. Scéna se liší pouze použitím vizuálních jedné vizuální značky. Na následujícím obrázku Obr. 6.21 lze nalézt dvojici rektifikovaných obrazů, na Obr. 6.22 pak výslednou mapu disparity. Pokud výsledky porovnáte s předchozím případem, nedošlo k téměř žádnému rozdílu, doba výpočtu však trvala pouhých 16.2s na rozdíl od 20.7s bez použití počátečního odhadu generovaného pomocí vizuálních značek.

Je jasně patrné, že použití tohoto postupu zvyšuje výpočetní rychlost celého algoritmu, ale na druhou stranu klade nároky na úpravu scény, která nemusí být vždy možná. Samotné řešení detekce vizuálních značek nebylo součástí této práce, pro detekci jsme využili vývojové SDK Unifeye firmy Metaio, které umožňuje detekci pozice, orientace a identifikátoru vizuálních značek přímo v reálném čase. Zopakujme, že značnou výhodou je okamžitá znalost trojrozměrné polohy vizuální značky ve scéně, která může být využita v celé řadě robotických aplikací.



Obr. 6.21: Dvojice rektifikovaných obrazů



Obr. 6.22: Výsledná mapa disparity

6.7. Zhodnocení výsledků

Přehled výsledků zpracování všech zmíněných typů úloh lze nalézt v Tab. 6.1.

Typ scény (kapitola)	Rozlišení obrazu	Doba výpočtu [s]
jednoduchá scéna (6.6.1)	800x600	24.7
složená nepřekrývající se scéna (6.6.2)	800x600	20.5
složená překrývající se scéna (6.6.3)	800x600	22.3
kruhová plocha (6.6.4)	800x600	20.7
kruhové plocha s počátečním odhadem (6.6.7)	800x600	16.2
detekce válcové plochy (6.6.5)	800x600	21.4
detekce komplikované scény (6.6.6)	800x600	22.5

Tab. 6.1: Tabulka přehledu výsledků generování mapy disparity

Realizovaná metoda byla schopná podat v prostředí ramene robotu velice kvalitní výsledky. Chyby, vzniklé v průběhu výpočtu, byly z velké části odstraněny následnou aplikací navrženého filtru na vypočítané mapě disparity (viz kapitoly 6.3, 6.4). Algoritmus byl schopen

korektně zpracovat nejen kruhové a válcové objekty, ale byl schopen zpracovat i scénu, ve které došlo ke vzájemnému překrývání jednotlivých objektů (occlusion objekty). Ke kvalitnímu výstupu metody došlo i v případě nedostatečně texturované scény, jako byl např. případ „kartonové krabice“.

V případě, kdy bylo použito generování počátečního odhadu pomocí jedné vizuální značky, došlo k urychlení výpočtu z 20.7s na 16.2s. Z tohoto výsledku lze předpokládat, že v případě použití větší počtu vizuálních značek dojde k výraznějšímu zkrácení doby výpočtu. Tato varianta, ale nebyla vyzkoušena. Metoda vizuálních značek je také schopna napravit chyby u problematických objektů, které jinak způsobují vysoké množství chyb a to jednoduchým překrytím problematických částí plochou vizuálního terčíku. Problémem ovšem zůstává, že tento přístup nelze aplikovat vždy.

Časy výpočtu se mohou zdát poměrně vysoké, je ale potřeba si uvědomit, že maximální hodnota disparity jako vstup algoritmu byla nastavena na vysokou hodnotu 70. Samozřejmě s klesající hodnotou maximální disparity klesá i výpočetní náročnost algoritmu, ovšem na úkor rozlišení metody.

V tabulce Tab. 6.2 lze nalézt výpočetní časy metody vzhledem k rozlišení vstupních párů obrázků. Jedná se o typ scény kruhová plocha (viz kapitola 6.6.4) bez použití počátečního odhadu. Upozorníme, že popsané rozlišení uvažuje velikosti vstupních obrazů, na kterých ještě nebyla provedena rektifikace.

V tabulce Tab. 6.2 lze nalézt výpočetní časy metody vzhledem k rozlišení vstupních párů obrázků. Jedná se o typ scény kruhová plocha (kapitola 6.6.7) s použitím počátečního odhadu. Z výsledků je patrné, že při použití počátečního odhadu pomocí vizuálních značek, se nejvíce projeví časová úspora u vstupních obrazů s vysokým rozlišením, zatímco u nižších rozlišení, se změna projeví pouze minimálně.

Rozlišení [px]	Doba výpočtu [s]
800x600	20.7
640x480	14.2
320x240	3.4
160x120	0.9

Tab. 6.2: Tabulka porovnání výpočetních časů v závislosti na rozlišení bez použití počátečního odhadu pomocí vizuálních značek

Rozlišení [px]	Doba výpočtu [s]
800x600	16.2
640x480	12.1
320x240	2.9
160x120	0.8

Tab. 6.3: Tabulka porovnání výpočetních časů v závislosti na rozlišení bez použití počátečního odhadu pomocí vizuálních značek

7. Distribuované počítačové vidění

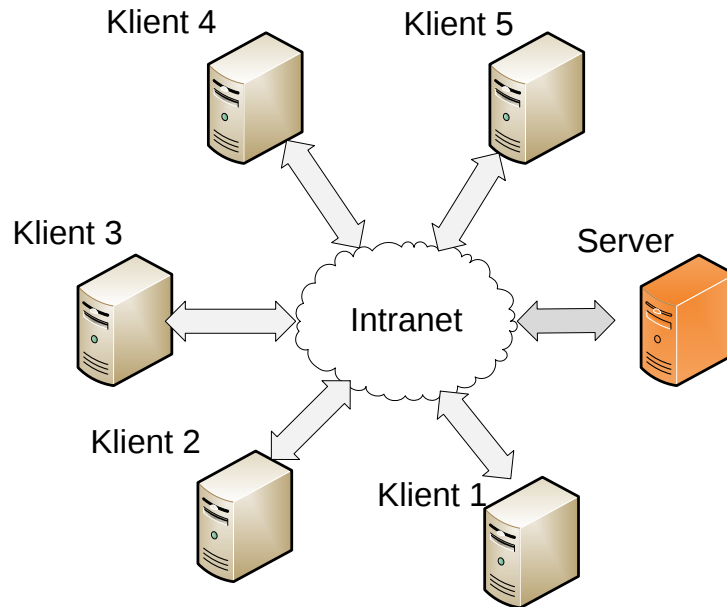
Velkým problémem metod dvourozměrného i trojrozměrného počítačového vidění je jeho výpočetní náročnost. Aby bylo dosaženo co nejrychlejší výpočtu, je nutné použít metod paralelního výpočtu. Toto může být provedeno buď pomocí výpočetních klastrů, které jsou velice drahé, nebo pomocí sítě běžných počítačů, které se v posledních letech výrazně zlevnily. Z tohoto důvodu se zdá být vhodným řešením zřetěžit výkon běžných počítačů a s jejich pomocí vytvořit distribuovaný systém počítačového vidění.

Na distribuovaný systém počítačového vidění jsou kladeny následující požadavky:

- architektura klient-server
- zabezpečená komunikace
- možnost identifikace serveru
- možnost identifikace služeb serveru
- maximální rozšiřitelnost
- stabilita

7.1. Návrh

Distribuovaný systém uvažujeme jako architekturu typu klient-server (viz Obr. 7.1). Ovšem na rozdíl od přístupu, kdy serverová část provádí veškeré výpočty a klient pouze odebírá výsledky jeho služeb, je v tomto případě serverová část určena pouze pro plánování úloh a výpočty jsou řešeny pomocí klientských částí. S rostoucím počtem klientů se tedy zvyšuje výkon celého systému.



Obr. 7.1: Schéma architektury klient-server

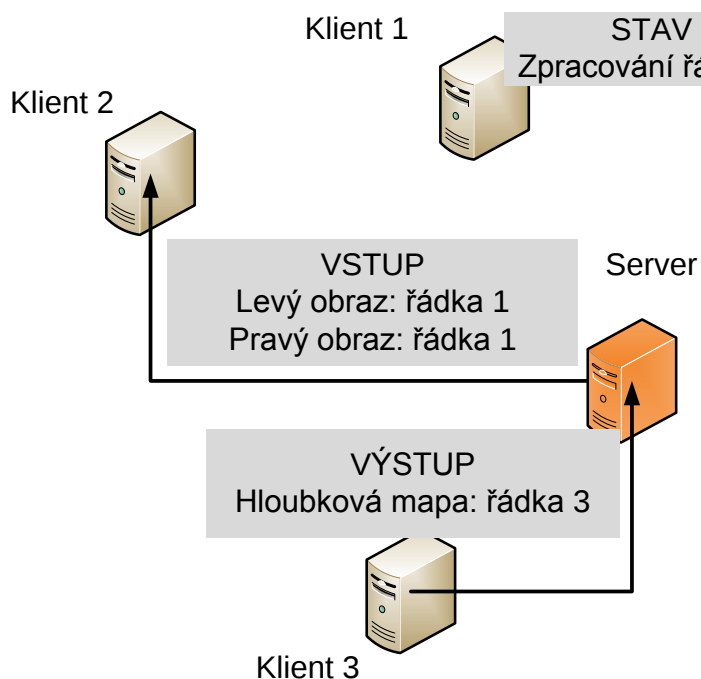
Pokud lze vhodným způsobem rozdělit celý proces výpočtu metody stereoidění na nezávislé celky, pak je také možné s rostoucím počtem použitých klientských počítačů zvyšovat výkon systému. Samozřejmě s omezením, maximálního počtu klientských počítačů, kdy již není možné další rozdělení činností realizovat.

Nejprve je nutné vzít v úvahu postup, jakým pracují metody stereoidění. Upozorníme, že popis procesu, který je zde uveden, odpovídá metodě stereoidění realizované v rámci této práce a nejedná se o obecný popis celého procesu. Schéma lze nalézt na Obr. 7.3. Jeho jednotlivé části procesu jsou:

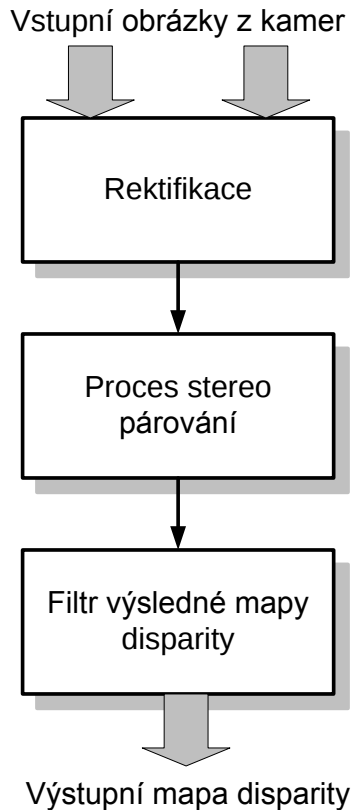
1. **Rektifikace:** Proces rektifikace (viz kapitola 3.5.4) je velice rychlý a není nutné ho distribuovat. Vstupem procesu rektifikace jsou nasnímané obrazy z obou kamer, kalibrační parametry jednotlivých kamery a informace o epipolární geometrii. Výstupem je dvojice rektifikovaných obrazů.

2. **Stereo párování:** Vstupem této metody jsou odpovídající si párové řádky z obou rektifikovaných obrazů. Výstupem metody jsou řádky mapy disparity. Proces je izolovaný pouze na vstupní řádky, které je možné zpracovávat absolutně samostatně. Není tedy problém ho distribuovat na klientské počítače. Serverová část aplikace pak jednotlivé výstupní řádky mapy disparity sestavuje do celkové výsledné mapy disparity.
3. **Filtr výsledné mapy disparity:** Tento filtr má dvě části. Nejprve se výsledná mapa disparity filtruje podél sloupců a následně podél řádků. Tento filtr by bylo možné distribuovat podél samostatných řádků a sloupců. Doba zpracování filtru je ovšem velice krátká, proto jsme se jeho distribucí v rámci této práce nezabývali.

Schéma činnosti distribuované metody počítačového vidění lze nalézt na Obr. 7.2.



Obr. 7.2: Schéma distribuce metody stereo vidění



Obr. 7.3: Proces metody stereo vidění

7.2. Technologie

Pro řešení výše zmíněných požadavků lze využít celé řady technologií. V této kapitole si je krátce popíšeme, ukážeme metodiku použití a zhodnotíme klady a zápory jejich využití s hlavním důrazem na rychlost a bezpečnost.

7.2.1. Přímá komunikace TCP/IP

Jedná se o socketovou komunikaci prostřednictvím protokolu TCP/IP. Serverová část naslouchá na definovaném portu a zpracovává klientské požadavky. Výhodou je vysoká

rychlost a to z důvodu využití vlastních jednoduchých protokolů. Tento způsob je však komplikovaný z hlediska implementace, kdy je nutné ručně napsat veškerý komunikační kód aplikace a navrhnout vlastní komunikační protokoly. Tento způsob přináší vysokou efektivitu komunikace, ale na druhou stranu otevírá možnost vytvoření bezpečnostních mezer, kterých se vývojář může nevědomě dopustit.

Pro tento typ komunikace je nutné zavést vhodný komunikační protokol. V zásadě lze použít následujících možností:

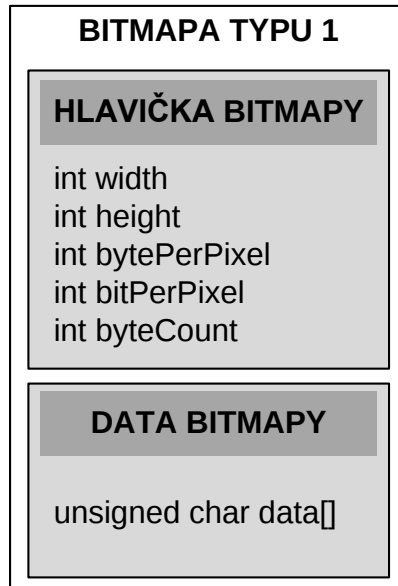
1. binární protokol
2. textový protokol
3. textový protokol založený na XML

V následujících podkapitolách budou jednotlivé možnosti komunikačních protokolů blíže popsány.

7.2.2. Binární protokol

Binární protokol je nejefektivnější vzhledem k objemu přenášených dat a rychlosti jejich zpracování. Přenášejí se pouze binární hlavičky a dále pak již pouze binární data. Nevýhodou binárního protokolu je nemožnost ho jednoduše dekódovat, což přináší nutnost úpravy klientské i serverové části aplikace, při jakékoliv změně formátu protokolu. Implementace musí mít striktně definovanou strukturu a velikost hlaviček a všech přenášených dat.

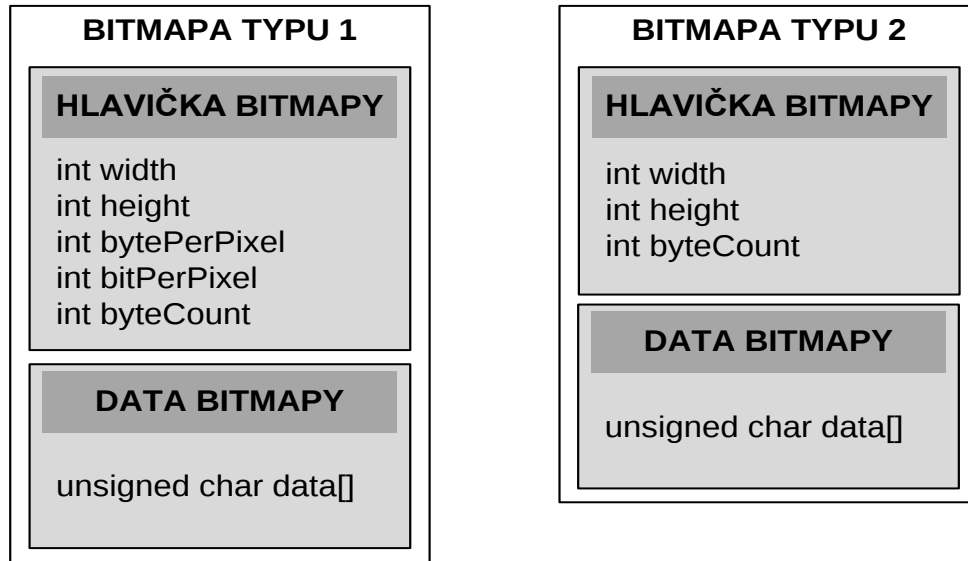
Na následujícím příkladu si ukážeme nevýhodu tohoto přístupu. Představte si, že ze serveru chceme odeslat obrázek. Provedeme to takovým způsobem, že klientovi odešleme následující datovou strukturu (viz Obr. 7.4).



Obr. 7.4: Příklad datové struktury přenášeného obrazu

Tento přístup by sám o sobě neznamenal žádný problém, ale co se stane, pokud budeme v některých případech chtít odeslat obrázek jiného formátu. Například dva formáty na následujícím obrázku Obr. 7.5. Toto již nemůžeme provést přímo, protože klient musí vždy vědět přesnou délku a formát zpracovávané datové struktury. Jedinou možností je zavést přídatnou strukturu, která tuto informaci bude přenášet.

Tento přístup by ovšem způsobil nefunkčnost starých verzí klientů po zavedení nové verze některých klientech, nebo serveru, kteří tento specifický typ dat vyžadují (z důvodu rozšíření funkcionality). Přesto, že starý typ klientů nový typ dat nevyžaduje, nebudou jejich aplikace fungovat. Z důvodu opravy tohoto případu je pak nutné přepsat velkou část aplikační logiky serveru. Právě toto je důvod pro použití rozšiřitelných formátů, ovšem za ztrátu části výkonu aplikace.



Obr. 7.5: Příklad dvou různých datových struktur přenášeného obrazu

7.2.3. Textový protokol

Textový protokol odstraňuje nutnost znalosti přesného formátu a velikosti datových struktur. Přináší sice nadbytečná data, ale umožňuje přenos dat v „člověkem čitelné podobě“. Tento způsob výrazně snižuje dobu proniknutí vývojáře do funkce již existujícího systému. Stále však zůstává několik problému a to je řešení struktury, hierarchie, typu a validace přenášených data. Z tohoto důvodu je v současné době velice rozšířená komunikace založená na bázi XML formátu.

7.2.4. XML protokol

XML (Extensible Markup Language) je obecný značkovací jazyk. Jeho hlavním účelem je usnadnění sdílení dat mezi širokým množstvím rozdílných informačních systémů a to zejména v síti internet.

Jedná se o zjednodušenou podmnožinu SGML (Standard Generalized Markup Language) a je navržen takovým způsobem, aby byl „čitelný člověkem“. XML je doporučeným standardem konsorcia W3C (World Wide Web Consortium), jedná se o otevřený formát, na který se nevztahují žádné poplatky. Doporučení W3C specifikují lexikální gramatiku i požadavky na zpracování XML dokumentů.

Schéma XML dokumentu XSD (XML Schema Definition) definuje syntaktická pravidla pomocí množiny omezení. Typicky omezuje elementy a atributy a hierarchii povoleného obsahu. Například element čas může obsahovat pouze jeden element pojmenovaný hodina, jeden element minuta a pouze jeden element sekunda. Navíc je možné definovat, že element hodina a element minuta musí být obsažen vždy, ale element sekunda je již volitelný.

Dokument, který odpovídá danému schématu je validním dokumentem. Takovýmto způsobem lze na straně serveru ověřovat požadavky a ty nevyhovující (nevalidní) okamžitě ignorovat.

Struktura formátu XML a stanovená striktní pravidla pro jeho formátování umožňují softwarovým vývojářům přenechat jeho zpracování standardním nástrojům, které jsou nyní součástí takřka každého programovacího jazyku. Vývojář se tak musí starat o zpracování dat, ale pouze o nakládání s nimi a to na poměrně vysoké úrovni abstrakce.

Aplikace předchozího příkladu pomocí protokolu XML by vypadala následovně:

```
<messageBody>
  <bitmap1 width="100" height="100" bytePerPixel="1"
    bitPerPixel="10">
    <!--Binární data zakódovaná pomocí Base64 -->
  </bitmap1>
</messageBody>
```

Pokud je nutné přenášet jiný typ obrázku, jednoduše zaměníme element obsažený uvnitř elementu „*messageBody*“. Příklad naleznete zde:

```
< messageBody >
  <bitmap2 width="100" height="100" bytePerPixel="1"
    bitPerPixel="10">
    <!--Binární data zakódovaná pomocí Base64 -->
  </bitmap2>
</ messageBody >
```

Na první pohled lze poznat, že provedená změna neovlivní žádným způsobem fungování starších typů aplikací, neznámá data nezpůsobí vznik kritické chyby, pouze dojde k detekci neznámého elementu (porovnejte s kapitolou 7.2.2). Toto je velice výhodně pro další rozšiřitelnost aplikace.

7.3. Web Services

Konsorcium W3C definuje webové služby (Web Services) jako softwarový systém navržený k usnadnění spolupráce výpočetních stanic přes datovou síť. Webové služby jsou často používány pouze jako webové API, které může být používáno vzdáleně přes datovou síť a vykonáváno na vzdálené výpočetní stanici, která danou webovou službu hostuje.

Definice webových služeb podle konsorcia W3C zahrnuje mnoho rozdílných typů. Běžně se používá komunikace typu klient server prostřednictvím XML zpráv, dodržujících standard SOAP. Důležitou vlastností webových služeb je schopnost nabídnout jednotný způsob poskytnutí dostupných služeb hostovaných na daném výpočetním stroji klientovi. Tento popis je definován ve formátu WSDL (Web Service Definition Language). Tímto způsobem je

umožněno automatické generování zdrojových kódů pro komunikace s webovou službou ve vývojových prostředích pro Javu, C# atd.

7.3.1. Specifikace

Bohužel díky velké variabilitě webových služeb neexistuje kompletní specifikace pro jejich definici. Na místo toho existuje několik hlavních specifikací, které se liší podle použitého typu technologie:

1. **SOAP:** Založeno na formátu XML, rozšiřitelný formát obálek zpráv ve spojení s podkladovým protokolem. Primárními protokoly jsou HTTP a HTTPS.
2. **WSDL:** Založeno na formátu XML. Umožňuje popsat interface webové služby. Typicky je tato vlastnost používána pro automatické generování kódu pro komunikaci s webovou službou. Dalším dobrým použitím je dynamické volání webových služeb.
3. **UDDI:** Protokol pro zveřejňování meta dat o webových službách. Umožňuje aplikacím najít webovou službu a to buď v době návrhu, nebo přímo za běhu aplikace.

Největší množství hlavních specifikací je specifikováno konsorciem W3C, včetně XMLP, SOAP a WSDL. Specifikace UDDI (Universal Description, Discovery and Integration) pochází z OASIS.

Proč tedy využívat webové služby pro distribuovaný systém počítačového vidění, když můžeme využít přímé komunikace prostřednictvím protokolu TCP/IP a použít i komunikace ve formátu XML? Opověď na tuto otázku je jednak WSDL, sloužící pro definici webových služeb. Dále pak přídatné specifikace webových služeb a tou je zejména zaměření na bezpečnost přenášených dat.

Zabezpečení komunikace definuje jak kódovat přenášená XML data, jak tyto data elektronicky podepsat přímo v SOAP obálce a zabezpečit tak neoprávněné změny v průběhu přenášení signálu, což slouží jako alternativa k používání HTTPS zabezpečených kanálů.

Podstata bezpečnosti přenosu data je v současné době stále důležitější. Distribuovaný systém počítačového vidění nasazený na výrobní lince může využívat výpočetního počítačů umístěných ve vzdálené budově. Samotná výrobní linka i budova s výpočetní technikou jsou zabezpečeny. Problém je pak v samotném přenosu dat mezi dvěma body, který může být narušen a zneužit. Proto je jako součást kvalitního distribuovaného systému počítačového vidění nezbytné řešení zabezpečení komunikace, které je však bohužel stále opomíjené.

Použití rozšířených vlastností webových služeb WSE (Web Services Enhancements) výrazně usnadňuje již hotová implementace na platformách, jako je .NET a Java. Velkou výhodou je profesionalita implementace, na které se podílí mnoho celosvětově významných firem a samozřejmě i bezpečnostních odborníků. Pokud bychom takovýto kód vytvořili sami, nevyhnuli bychom se množství nevědomých chyby, zapříčiňujících „*bezpečnostní díry*“ v systému. Toto je hlavní a zásadní důvod pro použití webových služeb.

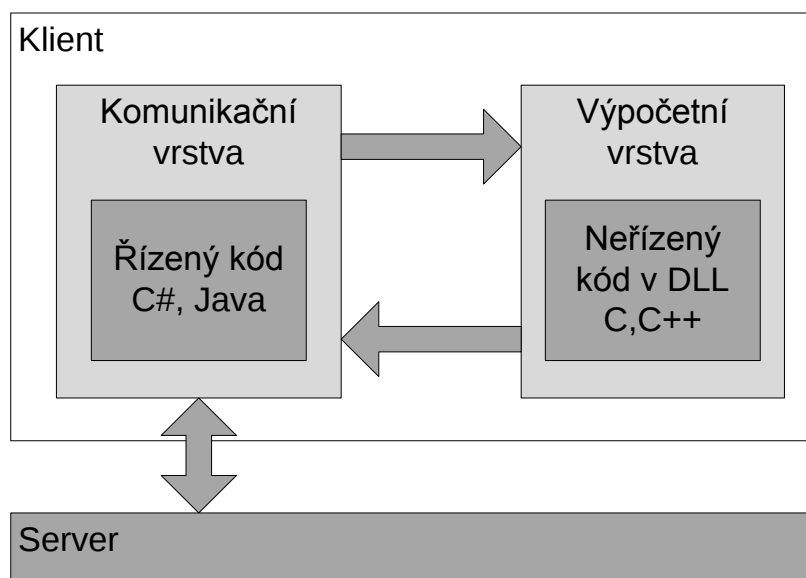
Pokud aplikaci budeme vyvíjet na platformě .NET, pak mohou být webové služby hostovány např. Microsoft IIS serveru. V případě programovacího jazyka Java je možné vybrat z celé škály webových serverů, jmenujme např. Apache Tomcat s rozšířením pro webové služby jménem Axis. Samozřejmě by bylo možné implementovat celý systém vlastním způsobem, to by ovšem bylo velice komplikované.

7.4. Rychlost komunikace

Problémem ovšem zůstává, že silnou podporu webových služeb ze strany již existujících knihoven nalezneme převážně pouze v řízených programovacích jazycích, které výkonnostně nedosahují výkonu kódu implementovaného v neřízeném jazyce C, nebo C++.

Jak ale tento problém vyřešit? Jedním z možných řešení, je použití kombinace řízeného a neřízeného kódu. Tedy část aplikace zajišťující komunikaci implementovat v řízeném kódu.

Metodu stereo párování, a následného filtru mapy disparity řešit jako DLL knihovnu v jazyce C++. To řešení se zdá být velice efektivním. Schéma lze nalézt na Obr. 7.6.



Obr. 7.6: Schéma kombinace řízeného a neřízeného kódu

7.5. Návrh webové služby na straně serveru

Nejprve je nutné vhodně rozvrhnout, která data se budou mezi klientem a serverem vyměňovat. Shrňme zde nyní situaci podle požadavků na straně klienta:

1. Server musí mít uloženu informaci o počtu připojených klientů. Tato informace bude použita pro optimální rozplánování úloh. Musí být tedy zavedeny dvě metody pro přihlášení a odhlášení klientských stanic. Je nutné vzít v úvahu, že na straně klienta může dojít k fatálnímu selhání. Proto je nutné zavést kontrolu časů odpovědí a automaticky nefunkční klienty odpojit a jejich úlohy přeplánovat na ostatní klienty. Musíme si také uvědomit, že klient může vykonávat mnoho různých typů výpočtů: Rektifikace, stereo párování a filtrace mapy disparity. Proto musí klient, jak součást

přihlašovacího procesu, předat serveru i seznam svých schopností. Tedy seznam metod, které je schopen poskytnout a zpracovat.

2. Po přihlášení do systému je již klient schopen zpracovávat výpočet. V tomto okamžiku musí existovat mechanismus, kterým klient požádá server o přidělení balíčku dat pro zpracování. Pokud na serveru data pro zpracování nejsou, je klient na definovaný čas uspán. V opačném případě klient získá balíček dat, který jednoznačně definuje proces, specifickou metodu procesu, vstupní data pro výpočet a jedinečný identifikátor klientské úlohy.
3. Po dokončení výpočtu na straně klienta musí existovat mechanismus pro odeslání dat zpět na server. Každý proces má definovaný výstupní formát dat. Tyto data jsou pak na server odeslány spolu s jedinečným identifikátorem klientské úlohy (nutné pro spárování s požadavkem).
4. Klient nutně musí být ošetřen pro případ vzniku výjimky. Také je nezbytné monitorovat fatální pád aplikace a zajistit automatický restart služby.

Shrnutí příkazů lze nalézt v Tab. 7.1.

Typ příkazu
Login
Logout
GetTask
ReturnTaskResult

Tab. 7.1: Typy příkazů použitých v distribuovaném vidění

Navíc je nutné definovat základní balíček přenášených dat (viz Tab. 7.2):

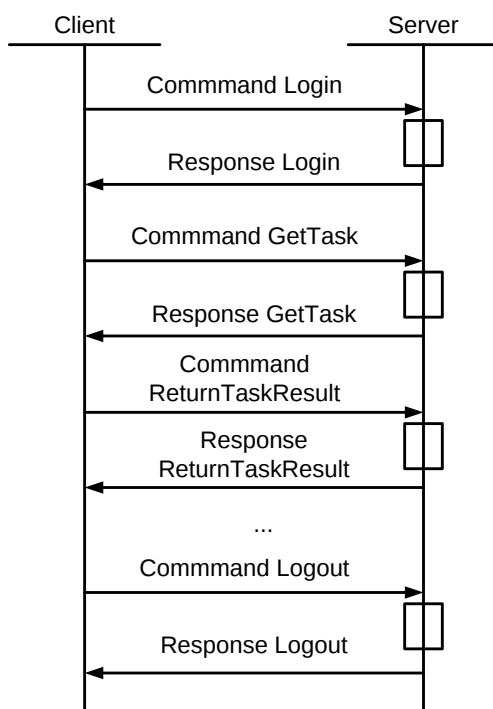
1. Libovolné požadavky klienta a serveru se přenášejí jako balíček dat „*Command*“. Hlavním parametrem je identifikátor příkazu a také data příkazu. Reakcí serveru na přijetí tohoto balíčku je odpověď typu „*Response*“, nebo „*Error*“.

2. Odpověď serveru typu „*Response*“ značí úspěšně zpracovaný příkaz a v těle jsou umístěny data odpovědi.
3. Odpověď serveru typu „*Error*“ značí chybu vzniklou v průběhu zpracování příkazu. V těle toho balíčku se nacházejí data, která chybu specifikují.

Balíček dat
Command
Response
Error

Tab. 7.2: Typy odpovědí na příkaz použitých v distribuovaném vidění

Proces komunikace klientu se serverem lze nalézt v UML diagramu na Obr. 7.7.



Obr. 7.7: Schéma distribuce metody stereo vidění

7.6. Zhodnocení výsledků

Na závěr kapitoly o distribuovaném počítačovém vidění uvedme rychlosti, s jakými data zpracovávala dvojice počítačů. Upozorníme, že metodu počátečního odhadu pomocí vizuálních značek jsme z časového důvodu neupravovali pro použití v distribuovaném výpočtu.

Pro výpočet byly použity dva shodné notebooky Asus A6Ja s procesorem Intel Core Duo T2400 1,83GHz, 1GB paměti RAM a operačním systémem Windows Vista. Použité notebooky obsahují dvou jádrové procesory, proto je možné efektivně spustit na každém z nich dva klienty a celkově tak simulovat distribuovaný výpočet na čtyřech jednoprosorových počítačích. Serverová aplikace byla nainstalována na jednom ze dvou notebooků společně s aplikací klientskou.

V tabulce Tab. 7.3 lze nalézt výpočetní časy distribuované metody stereovidění vzhledem k počtu použitých výpočetních klientů. Jedná se o typ scény kruhová plocha (kapitola 6.6.4) bez použití počátečního odhadu, kde parametry použité pro výpočet jsou stejné jako v případě úlohy z kapitoly 6.6.4. Rozlišení použitých vstupních obrazů před rektifikací je 800x600.

Počet použitých výpočetních klientů	Doba výpočtu [s]
1	21.2
2	11.3
3	7.8
4	6.1

Tab. 7.3: Tabulka výpočetních časů na samostatných počítačích

Z výsledků je patrné, že metoda byla navržena takovým způsobem, kde s rostoucím počtem výpočetních strojů dochází k přibližně lineárnímu nárůstu výpočetního výkonu. Při

dostatečném počtu výpočetních strojů lze tedy dosáhnout výpočetní rychlosti řádově sekund, která je již v oblasti robotiky poměrně dobře použitelná. Výpočet je teoreticky možné distribuovat až na počet výpočetních klientů odpovídající počtu řádků ve vstupních obrazech. V případě nutnosti rychlejšího výpočtu by kromě použití většího počtu klientů bylo nutné zavést distribuovaný výpočet rektifikace vstupních obrazů a také koncového filtru výstupní mapy disparity.

V případě použití nižšího rozlišení vstupních obrazů 320x240 bylo při použití čtyř výpočetních klientů dosaženo rychlostí blížících se jedné sekundě pro zpracování jedné dvojice vstupních obrazů.

8. Závěr

Závěrem krátce shrneme obsah a výsledky jednotlivých kapitol. Připomeňme, že dílčí experimentální závěry lze nalézt v kapitolách 2.4, 3.5, 6.6 a 7.6.

Kapitola 2 popisovala realizaci rychlé a zároveň jednoduché automatizované metody detekce polohy efektoru ramene robotu pro konkrétní aplikaci s ramenem robotu společnosti ABB. Realizovaná metoda umožňuje snadné a automatizované nasnímání množiny párů trojrozměrných pozic bodů ve scéně a jejich dvourozměrných pozic v obraze. Získaná data lze použít pro následnou kalibraci jednotlivých kamer. Použitá metoda je založena na detekci polohy aktivních LED umístěných na efektoru ramene robotu, kde trojrozměrná poloha LED je dobře známa a detekuje se pouze poloha jednotlivých LED v nasnímaném obraze. Ověřili jsme vysokou odolnost vůči chybám způsobených nedokonalostí nasnímaného obrazu a díky proměnlivým světelným podmínkám.

Kalibrace kamer je podstatným krokem pro jakékoliv další zpracování obrazu. Jedná se o oblast diskutovanou v mnoha vědeckých publikacích. Co zde, ale chybí je přehled některých metod vhodných pro použití v oblasti robotiky spolu s jejich porovnáním na konkrétním případě. Právě z tohoto důvodu byly v kapitole 3 realizovány tři metody kalibrace kamery a to metoda Hallova, Faugerasova a Faugerasova se zkreslením. Jednotlivé metody byly popsány spolu s použitým matematickým aparátem, tak aby je bylo možné znovu snadno implementovat. Na závěr byla ověřena a porovnána přesnost všech zmíněných metod na reálných datech z robotické scény.

Kapitola 4 popisuje možnosti hloubkové rekonstrukce a epipolární geometrii. Nezaměřujeme se pouze na epipolární geometrii samotnou, ale zejména na její důsledky nutné pro prostorovou rekonstrukci a rektifikaci nasnímaných obrazů. V rámci této kapitoly byla

vytvořena vlastní implementace metody rektifikace, která je důležitým předpokladem pro následující metodu stereo párování.

Oblast problematiky hloubkového vnímání je komplexní a stále se rozvíjející, proto kapitola 5 obsahuje shrnutí existujících metod, jejich vlastností a možnosti použití omezení pro urychlení uvedených metod. Součástí textu je množství referencí, které se dají dobře použít pro další výzkum v oblasti stereovidění.

Získané rektifikovaných obrazy jsou vhodným vstupem pro metodu stereo párování a sestavení výsledné mapy disparity. V rámci této práce byla v kapitole 6 úspěšně implementována metoda stereo párování, založená na dynamickém programování a zjednodušeném korelování na okolí pixelů. Metoda byla dále vylepšena o filtr výstupní mapy disparity, tak aby došlo k odstranění chyb a doplnění informací v místech obrazu s nedostatečnou, nebo žádnou texturou. Použitý filtr byl navržen takovým způsobem, aby nebyl výpočetně příliš náročný. Součástí řešení je také experimentální ověření funkce metody na několika rozdílných typech scény. Z výsledků je patrné, že metoda byla schopna zpracovat komplexní scény a produkovat kvalitní mapu disparity. Dalším bodem, kterým jsme se zabývali, je generování počátečního odhadu pro metodu stereo-párování, založeném na detekci pozice a orientace vizuálních značek umístěných ve scéně ramene robotu. Jedná se o specifický případ, který si v námi uvažovaném případě scény ramene robotu můžeme dovolit. Počátečního odhadu bylo úspěšně využito pro urychlení metody stereo-párování.

Kapitola 7 je odpovědí na vysokou výpočetní náročnost metod počítačového vidění a zavádí vlastní řešení distribuovaného výpočtu. Tento problém je řešen poněkud netradičním způsobem. Jedná se o kombinaci řízeného jazyka C# na platformě .NET, který řeší veškeré otázky komunikace a neřízeného jazyka C++ na platformě Win32 , který zajišťuje výpočet (jako DLL knihovna volaná z .NET). V této kapitole jsou diskutovány rozdílné možnosti přístupu k řešení a záměrem je snaha o použití moderních technologií. Metoda byla navržena

takovým způsobem, kde s rostoucím počtem výpočetních strojů dochází k přibližně lineárnímu nárůstu výpočetního výkonu.

Práce se od ostatních prací v oboru liší v následujících bodech:

- Použití jednoduchého filtru výstupní mapy disparity, který výrazně nezvyšuje celkový výpočetní čas, ovšem zlepšuje kvalitu výstupních dat jednoduché a výpočetně nenáročné metody stereo-vidění.
- Úprava metody pro zpracování vysokých hodnot disparity, včetně experimentálního ověření. Ostatní metody v tomto případě zpracovávají vstupní obrazy s velmi omezujícím limitem na rozsah disparity.
- Návrh počátečního odhadu pomocí detekce vizuálních značek ve snímané scéně. Pro detekci bylo použito metod z oblasti augmented reality. Použitý postup je v oblasti stereo-vidění nový a v ostatních pracích nevyzkoušený.
- Pojetí celého procesu stereo-vidění robotů jako propojeného celku. Ostatní práce se zabývají vždy jenom izolovanou oblastí.

Tyto výsledky poskytují dobrý základ pro další výzkum v oblasti počítačového vidění, kde již není nutné pracně procházet množství existujících publikací a získat tak dostatečné množství informací nutných pro realizaci celého komplexního procesu stereovidění. Práce dále zavádí rozšíření metody stereovidění o generování počátečního odhadu pomocí detekce vizuálních značek ve scéně, které může být zajímavým řešením pro průmyslové aplikace. Polohu a orientaci vizuálního terčíku je možné detekovat v reálném čase a doba výpočtu mapy disparity na okolí vizuálního terčíku je minimální. Práce dále přináší nový pohled na distribuovaný výpočet, ve kterém se snaží poskytnout řešení pomocí moderních technologií.

Pro budoucí pokračování je zajímavým řešením implementace námi navržené metody pro GPU moderních grafických karet [3], která by výrazným způsobem urychlila celkovou dobu výpočtu. Další možností je úprava metody pro použití nejenom s robotickým ramenem, ale také s mobilními roboty. Použití vizuálních terčů nabízí také další možnost rozvoje, kde by docházelo k trojrozměrné rekonstrukci objektů s umístěným terčem a jejich uložení jejich geometrie do databáze. V dalším zpracování by pak již bylo možné okamžitě načíst hloubkovou informaci celého objektu a dopočítat pouze zbytek snímané scény,

Závěrem je možné říci, že se povedlo úspěšně realizovat jednotlivé metody počítačového vidění a sestavit je do funkčního celku použitelného v rámci počítačového vidění robotu. Tyto metody jsou ve většině případů prezentovány jako oddělené celky a je obtížné nalézt zdroj informací, který by je rozumným způsobem zastřešoval. Tato práce problematickou situaci řeší a přináší všechny nezbytné informace nutné k realizaci vidění robotu a předkládá celkové a funkční řešení, navíc zavádí několik zajímavých inovací oproti stávajícím metodám počítačového vidění. Celý proces stereovidění je navíc řešen pomocí vlastní implementace a kromě vývojového SDK Unifeye firmy Metaio (detekce vizuálních značek ve scéně) není použito žádných jiných SDK ani jiných nástrojů.

9. Literatura

- [1] J. Kostková, J. Čech, R. Šára. Feasibility Boundary in Dense and Semi-Dense Stereo Matching. CVPR 2007 IEEE Conference. 2007.
- [2] V. Kolmogorov, A. Criminisi, A. Blake, G. Cross, C. Rother. Bi-layer segmentation of binocular stereo video, US Proc. IEEE Computer Vision and Pattern Recognition, 2005.
- [3] R. Yang, L. Wang, G. Welch, M. Pollefeys. Stereovision on GPU. Workshop on Edge Computing Using New Commodity Architectures, 2006.
- [4] A. Blake, P. H.S. Torr, I. Cox, A. Criminisi. Estimating uncertainty in dense stereo disparity maps, Cambridge, 2003.
- [5] J. Kostková, R. Šára. Stratified Dense Matching for Stereopsis in Complex Scenes. British Machine Vision Conference. 2003.
- [6] H. Ishikawa , D. Geiger. Occlusions, discontinuities, and epipolar lines in stereo. ECCV. 1998.
- [7] Wesley E. Snyder, Hairong Qi. Cambridge Machine Vision, University Press. 2004. ISBN 052183046X
- [8] Jorge L C Sanz. Advances in Machine Vision. Springer, 1989. ISBN 0387968229
- [9] Michael Jenkin, Laurence Hartus. Spatial Vision in Humans and Robots. Cambridge University Press, 1994. ISBN 0521430712
- [10] Reinhard Klette, Shmuel Peleg, Gerald Sommer. Robot Vision. Springer, 2001. ISBN 3540416943
- [11] J. Pauli. Learning-Based Robot Vision. Springer, 2001. ISBN 3540421084
- [12] M. Hebert. Active and passive range sensing for robotics. IEEE International Conference on Robotics and Automation, 2000.

- [13] Nicholas Ayache. Artificial Vision for Mobile Robots. Cambridge, 1991. ISBN 0262011257
- [14] E. Trucco and R. B. Fisher. Acquisition of consistent range data using local calibration. IEEE International Conference on Robotics and Automation, 1994.
- [15] R. A. Brooks. Intelligence without representation. Artificial Intelligence Journal, 1991.
- [16] M. A. Rodrigues, Y. F. Li, M. H. Lee, J. J. Rowland, and C. King. Robotic grasping of complex objects without full geometrical knowledge of shape. IEEE International Conference on Robotics and Automation, 1995.
- [17] A. Hoover, G. Jean-Baptiste, X. Jiang, P. J. Flynn, H. Bunke, D. B. Goldgof, K. Bowyer, D. W. Eggert, A. Fitzgibbon, and R. B. Fisher. An experimental comparison of range image segmentation algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1996.
- [18] R. I. Hartley. Euclidean Reconstruction from Uncalibrated Views. Second European Workshop on Applications of Invariance in Computer Vision, 1993.
- [19] O. D. Faugeras. Three-dimensional computer vision. The MIT Press, Cambridge, Massachusetts, 1993.
- [20] R. M. Haralick and L. G. Shapiro. Computer and robot vision. Addison-Wesley Publishing Company, 1992.
- [21] R. Ahlers and J. Lu. Stereoscopic Vision - An Application Oriented Overview. SPIE - Optics, Illumination, and Image Sensing for Machine Vision IV, 1989.
- [22] J. Batlle, E. Mouaddib and J. Salvi. A Survey: Recent Progress in Coded Structured Light as a Technique to Solve the Correspondence Problem. Pattern Recognition, 1989.
- [23] R. A. Jarvis. A Perspective on Range Finding Techniques for Computer Vision. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1983.
- [24] T. S. Newman. A Survey of Automated Visual Inspection. Image Understanding, 1995.

- [25] A. Broggi. Vision-Based Driving Assistance in Vehicles of the Future. IEEE Intelligent Systems, 1998.
- [26] A. Casals. Sensor devices and systems for robotics. Springer-Verlag. NATO ASI Series, Berlin Heidelberg, 1989.
- [27] L. Charbonnier and A. Fournier. Heading Guidance and Obstacles Localization for an Indoor Mobile Robot. IEEE International Conference on Advanced Robotics, 1995.
- [28] D. Khadraoui, G. Motyl, P. Martinet, J. Gallice and F. Chaumette. Visual Servoing in Robotics Scheme Using a Camera/Laser-Stripe Sensor. IEEE International Journal on Robotics and Automation, 1996.
- [29] R. K. Lenz and R. Y. Tsai. Calibrating a Cartesian Robot with Eye-on-Hand Configuration independent of Eye-to-Hand Relationship. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1989.
- [30] M. Li. Camera Calibration of a Head-Eye System for Active Vision. European Conference on Computer Vision, 1994.
- [31] M. Ito. Robot Vision Modelling - Camera Modelling and Camera Calibration. Advanced Robotics, 1991.
- [32] R. K. Lenz and R. Y. Tsai. Techniques for Calibration of the Scale Factor and Image Center for High Accuracy 3D Machine Vision Metrology. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1988.
- [33] Y. Liu, T. S. Huang and O. D. Faugeras. Determination of Camera Location from 2-D to 3-D Line and Point Correspondences. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1990.
- [34] C. C.Wang. Extrinsic Calibration of a Vision Sensor Mounted on a Robot. IEEE Int. Journal on Robotics and Automation, 1992.

- [35] E. L. Hall, J. B. K. Tio, C. A. McPherson and F. A. Sadjadi. Measuring Curved Surfaces for Robot Vision. Computer Journal, 1982.
- [36] J. Batista, H. Araújo and A. T. de Almeida. Iterative Multistep Explicit Camera Calibration. IEEE Transactions on Robotics and Automation, 1999.
- [37] G.-Q. Wei and S. De Ma. Implicit and Explicit Camera Calibration: Theory and Experiments. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1994.
- [38] O. D. Faugeras and G. Toscani. The Calibration Problem for Stereo. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1986.
- [39] R. Y. Tsai. A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses. IEEE International Journal on Robotics and Automation, 1987.
- [40] Z. Hong and J. Yang. An Algorithm for Camera Calibration Using a Three-Dimensional Reference Point. Pattern Recognition, 1993.
- [41] S. Kamata, R. O. Eason, M. Tsuji and E. Kawaguchi. A Camera Calibration Using 4 Points Targets. International Conference on Pattern Recognition, 1992.
- [42] C. C. Slama, C. Theurer and S. W. Henriksen. Manual of photogrammetry. American Society of Photogrammetry, 1992.
- [43] J. Weng, N. Ahuja and T. S. Huang. Matching Two Prespective Views. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1992.
- [44] G.-Q. Wei and S. De Ma. Implicit and Explicit Camera Calibration: Theory and Experiments. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1994.
- [45] J. Salvi, J. Batlle and E. M. Mouaddib. A Robust-Coded Pattern Projection for Dynamic 3D Scene Measurement. Pattern Recognition Letters, 1998.

- [46] R. Deriche, Z. Zhang, Q.-T. Luong and O. Faugeras. Robust Recovery of the Epipolar Geometry for an Uncalibrated Stereo Ring. European Conference on Computer Vision, 1994.
- [47] M. Bober, N. Greorgis and J. Kittler. On Accurate and Robust Estimation of Fundamental Matrix. Computer Vision and Image Understanding, 1998.
- [48] P. H. S. Torr and A. Zisserman. MLESAC: A New Robust Estimator with Application to Estimating Image Geometry. Computer Vision and Image Understanding, 2000.
- [49] B. Julesz. Foundations of Cyclopean Perception. University of Chicago Press, 1971.
- [50] Y. Ruichek and J.-G. Postaire. A neural matching algorithm for 3-D reconstruction from stereo pairs of linear images. Patter Recognition Letters, 1996.
- [51] Thomas Babu, B. Yegnanarayana, and Sukhendu Das. Stereo-correspondence using Gabor logons and neural networks. In Proceedings of the International Conference on Image Processing, 1995.
- [52] A. Zanela and S. Taraglio. Sensing the third dimension in stereo vision systems: a cellular neural networks approach. Engineering Applications of Artificial Intelligence, 1998.
- [53] A.V. Oppenheim and R.W. Schaefer. Discrete-Time Signal Processing. Prentice Hall, 1989.
- [54] W. Von Seelen et al. A Neural Architecture for Autonomous Visually Guided Robots- results of the NAMOS Project, 1995.
- [55] M. Adjouadi, F. Candocia, and Riley J. Exploiting Walsh-based attributes to stereo vision. IEEE Transactions on Signal Processing, 1996.
- [56] D. G. Jones and J. Malik. Computational framework for determining stereo correspondence from a set of linear spatial filters. Image and Vision Computing, 1992.
- [57] S. Birchfield and C. Tomasi. Depth discontinuities by pixel-to-pixel stereo. Technical Report TAN-CSTR- 96-1573, Stanford University, 1996.

- [58] H. S. M. Coxeter. Projective Geometry. University of Toronto, 1974.
- [59] Z. Zhang and G. Xu. Epipolar Geometry in Stereo, Motion and Object Recognition. Kluwer Academic Publishers, 1996.

10. Vlastní publikace

Štilec J.: DIRECTX FOR ROBOT VISION. In: 4th International PhD Conference on Mechatronical Engineering 2006. Plzeň, 2006. pp. 95-96. ISBN 80-7043-486-4.

Štilec J., Holada M., Kopetschke I., Pirkel P., Pelc M., Matela L., Horčíčka J.: THE PROTOTYPE OF HUMAN - ROBOT INTERACTIVE VOICE CONTROL SYSTEM. In: Proc. of the Fourth International Conference on Informatics in Control, Automation and Robotics (ICINCO 2007), Angers, France, 2007. vol. RA-1, pp. 307-310. ISBN: 978-972-8865-83-2

Štilec J., Holada M., Pelc M., Kopetschke I., Pirkel P., Matela L., Horčíčka J.: VOICE INTERACTIVE CONTROL SYSTEM FOR ROBOTS WITH DISTRIBUTED COMPONENTS. In: ECMS 2007 & Doctoral School (EDSYS, GEET). Liberec, 2007. pp. 160-164. ISBN 978-80-7372-218-0.

Štilec J.: PARALLEL STEREO VISION FOR ABB ROBOTS. In: ECMS 2007 & Doctoral School (EDSYS, GEET). Liberec, 2007. pp. 170-174. ISBN 978-80-7372-218-0.