

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra kybernetiky

DIPLOMOVÁ PRÁCE

Plzeň, 2013

Bc. Lukáš Bureš

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Lukáš BUREŠ**
Osobní číslo: **A11N0048P**
Studijní program: **N3918 Aplikované vědy a informatika**
Studijní obor: **Kybernetika a řídicí technika**
Název tématu: **Sledování pohybu rukou ve 3D pomocí dvou kamer**
Zadávací katedra: **Katedra kybernetiky**

Z á s a d y p r o v y p r a c o v á n í :

1. Seznamte se s metodami kalibrace kamer, sledování objektů ve videozáznamu a 3D rekonstrukce.
2. Navrhněte metodiku snímání gest.
3. Implementujte a srovnajte metody sledování baravně odlišeného objektu ve videozáznamu.
4. Implementujte algoritmus 3D rekonstrukce trajektorií sledovaných objektů.

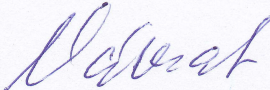


Rozsah grafických prací: **dle potřeby**
Rozsah pracovní zprávy: **35-50 stránek A4**
Forma zpracování diplomové práce: **tištěná**
Seznam odborné literatury:

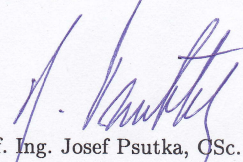
[1] Sonka M., Hlavac V., Boyle R.: **Image Processing, Analysis, and Machine Vision, Third Edition, Thomson, 2008.**

Vedoucí diplomové práce: **Ing. Marek Hruží**
Katedra kybernetiky

Datum zadání diplomové práce: **24. září 2012**
Termín odevzdání diplomové práce: **17. května 2013**


Doc. Ing. František Vávra, CSc.
děkan




Prof. Ing. Josef Psutka, CSc.
vedoucí katedry

V Plzni dne 24. září 2012

PROHLÁŠENÍ

Předkládám tímto k posouzení a obhajobě diplomovou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Plzni dne 16. května 2013

.....
Bc. Lukáš Bureš

PODĚKOVÁNÍ

Chtěl bych vyjádřit velké poděkování panu Ing. Markovi Hrúzovi za cenné připomínky, odborné rady, podněty k zamyšlení a dlouhodobé odborné vedení, za které jsem mu velmi vděčen. Dále bych chtěl poděkovat mé rodině za podporu při studiu a v neposlední řadě všem, kteří mě jakkoliv podpořili při psaní diplomové práce nebo samotném studiu na vysoké škole.

Anotace

Diplomová práce v úvodu popisuje teorii počítačového vidění, konkrétně model dírkové kamery a 3D projekci. Dále práce shrnuje teorii stereo kalibrace, epipolární geometrie a 3D rekonstrukce. Popisuje metodu odečítání pozadí a prezentuje hlavní metody pro sledování objektů. Praktická část této diplomové práce je implementována v programovacím jazyce C++ za využití knihovny OpenCV. Je rozdělena na systém stereo kalibrace, odečítání pozadí a systém pro sledování a 3D rekonstrukci objektu. V závěru prezentuje získané výsledky a navrhuje další možná vylepšení.

Klíčová slova : C++, OpenCV, stereo kalibrace, odečítání pozadí, sledování, 3D rekonstrukce, mapa disparit, vyhledávací tabulka, 3D projekce

Annotation

This diploma thesis describes the theory of computer vision in the introduction specifically the pinhole camera model and 3D projection. The thesis also summarizes the theory of stereo calibration, epipolar geometry and 3D reconstruction. It describes a method for background subtraction and it presents main methods for tracking objects. The practical part of this thesis is implemented in the programming language C++ using OpenCV library. It is divided into a system of stereo calibration, background subtraction and system of tracking and 3D reconstruction. In conclusion, it presents the obtained results and suggests another possible improvements.

Keywords : C++, OpenCV, stereo calibration, background subtraction, tracking, 3D reconstruction, disparity map, lookup table, 3D projection

Obsah

Obsah	I
Seznam obrázků	IV
Seznam tabulek	VII
1 Úvod	1
1.1 Popis problému	1
1.2 Současný stav řešeného problému	2
2 Počítačové 3D vidění a geometrie	4
2.1 Body a přímky v projekčním prostoru	5
3 Model kamery	6
3.1 Model dírkové kamery	6
4 3D vidění a projektivní geometrie	11
4.1 Kalibrace jedné kamery	11
4.1.1 Kalibrace kamery ze známé scény	12
4.1.2 Další metody kalibrace	14
4.2 Dvě kamery, stereo vidění	14
4.2.1 Epipolární geometrie	15
4.2.2 Fundamentální matice	16
4.2.3 Osmibodový algoritmus	17
4.3 Rektifikace	18
4.4 3D rekonstrukce	21
5 Metoda odečítání pozadí	22
5.1 Obecný přístup	22

5.1.1	Základní přístup	23
5.1.2	Filtrace mediánem	23
5.1.3	Gaussovské rozložení	23
5.1.4	Gaussovská směs (GMM)	24
5.1.5	Kernel Density Estimace (KDE)	25
5.1.6	Modelování histogramem	25
5.2	Barevné prostory	26
6	Sledování	27
6.1	Optický tok	27
6.2	Lucas-Kanade	28
6.3	Kontinuální adaptivní mean-shift	30
6.3.1	Mean-shift	30
6.3.2	Camshift	31
6.4	Kalmanův filtr	33
6.5	Particle filtr	35
7	Systém stereo kalibrace	38
7.1	Použitý hardware a software	38
7.2	Kalibrační vzor	40
7.3	Popis implementace systému stereo kalibrace	41
7.4	Možnosti kalibrace	44
8	Systém odečítání pozadí	45
8.1	Úvod a definice vyhledávací tabulky	45
8.2	Popis implementace systému odečítání pozadí	46
8.2.1	Trénování LUT pozadí	46
8.2.2	Trénování LUT popředí	46
8.2.3	Dilatace LUT	50
8.3	Experiment volby barevného prostoru	51
8.3.1	Uvažované barevné prostory	51
8.3.2	Nastavení a popis experimentu	52
8.3.3	Výsledky	54
8.3.4	Vyhodnocení výsledků a výběr barevného prostoru	59
8.4	Vizualizace segmentace pomocí LUT	59

8.5	Vizualizace LUT	60
9	Systém sledování a 3D rekonstrukce	62
9.1	Popis implementace systému sledování a 3D rekonstrukce	62
9.1.1	Výpočet a vylepšení reprezentujícího 3D bodu	65
9.2	Vizualizace 3D trajektorií	70
10	Další možná rozšíření a vylepšení	72
11	Závěr a zhodnocení výsledků	73
A	Uživatelská příručka k systému stereo kalibrace	76
B	Uživatelská příručka k systému odečítání pozadí	77
C	Uživatelská příručka k systému sledování a 3D rekonstrukce	79
	Literatura	I

Seznam obrázků

2.1	Projekční perspektiva rovnoběžných přímek.	5
3.1	Geometrie lineární perspektivy kamery.	7
3.2	Geometrie podobnosti trojúhelníků.	8
4.1	Geometrie systému se dvěma kamerami.	15
4.2	Rektifikovaná konfigurace dvou pohledů na scénu, kde epipolární přímky jsou orientované rovnoběžně a epipóly leží v nekonečnu.	19
4.3	Rektifikovaná dvojice obrázků, ve kterých si jednotlivé řádky odpovídají.	19
4.4	Jednoduchá stereo geometrie v rektifikované konfiguraci kamer.	20
7.1	Typ použitých běžně dostupných webových kamer značky Trust.	39
7.2	Nejmenší možný kalibrační vzor obsahující pouze 4 vnitřní rohy.	40
7.3	Použitý kalibrační vzor o velikosti 6×9 vnitřních rohů.	40
7.4	Náhled dvojice kamer s vizualizací nalezených vnitřních rohů pro následnou stereo kalibraci.	42
7.5	Rektifikovaná dvojice obrázků.	43
8.1	Vizualizace binární masky popředí.	48
8.2	Vizualizace binární masky pozadí.	48
8.3	Vizualizace výsledných barevných bodů, které jsou natrénovány do LUT pravého popředí.	49
8.4	Použitý strukturní element s počátečním bodem ve středu.	50
8.5	V levé části je zobrazen pohled na obrazovou scénu a trénovaný barevný obrazec. V pravé části se nachází odpovídající ruční segmentace červeného trénovacího obrazce.	53
8.6	Detail ruční segmentace.	53
8.7	Detail RGB segmentace.	53

8.8	Detail YC_bC_r segmentace.	54
8.9	Detail YUV segmentace.	54
8.10	Detail CIE-Lab segmentace.	54
8.11	Detail CIE-Luv segmentace.	54
8.12	Detekované počty bodů v ručně označených snímcích.	55
8.13	Detekované počty bodů v RGB snímcích.	55
8.14	Detekované počty bodů v YC_bC_r snímcích.	55
8.15	Detekované počty bodů v YUV snímcích.	55
8.16	Detekované počty bodů v CIE-Lab snímcích.	55
8.17	Detekované počty bodů v CIE-Luv snímcích.	55
8.18	Chyba RGB.	57
8.19	Chyba YC_bC_r	58
8.20	Chyba YUV.	58
8.21	Chyba CIE-Lab.	58
8.22	Chyba CIE-Luv.	58
8.23	Levá část představuje originální snímek z kamery a pravá polovina výslednou segmentaci pomocí dilatované LUT.	59
8.24	Detail segmentací, v levé polovině segmentace pomocí LUT před di- latací, v pravé polovině po dilataci.	60
8.25	Vizualizace možného výsledku natrénované RGB barevné LUT.	61
8.26	Vizualizace možného výsledku natrénované RGB barevné LUT po aplikování dvou iterací dilatace.	61
9.1	Vizualizace jednotlivých ohraničujících rámečků.	63
9.2	Snímek určený k 3D rekonstrukci a výpočtu reprezentujícího bodu trajektorie.	66
9.3	Levá část obsahuje mapu disparit a pravá binární masku odpovídající snímku na Obr. 9.2.	66
9.4	Izometrický pohled na 3D rekonstruované body.	67
9.5	Přední pohled na 3D rekonstruované body.	67
9.6	Přední pohled, body pro výpočet reprezentujícího 3D bodu trajektorie.	68
9.7	Data objektu společně s chybně rekonstruovanými body.	69
9.8	Oříznutá data objektu.	69
9.9	Trajektorie pohybu ve směru osy z	70
9.10	Trajektorie písmene V v rovině yz	70

SEZNAM OBRÁZKŮ

9.11 Trajektorie kruhových pohybů v rovině xz	71
9.12 Trajektorie kruhového pohybu v rovině xy	71

Seznam tabulek

8.1	Počty bodů a absolutní procentuální rozdíl od ručně označené segmentace.	56
8.2	Hodnoty součtů velikostí absolutních chyb v pixelech přes všechny snímky.	57
8.3	Střední hodnoty relativních chyb a jejich procentuální hodnoty. . . .	58
8.4	Počty pixelů v jednotlivých LUT.	60
9.1	Souřadnice centroidů.	68
9.2	Souřadnice centroidů pro modelový příklad.	69

Kapitola 1

Úvod

Člověk používá k orientaci v prostoru z velké části svůj zrakový smysl, proto se díky páru očí dokáže velmi precizně a bez velkých potíží pohybovat v prostoru. Nečiní mu problémy odhadovat vzdálenost, vykonávat pohyb, provádět interakci, představovat si pohyby a trajektorie objektů v prostoru. Pokud tyto věci, přirozené pro člověka, jsou aplikovány na stroje, konkrétně výpočetní techniku a počítače, tak se jedná o velmi složité a komplexní problémy. Počítačové vidění v současné době disponuje řadou nástrojů a přístupů pro získávání odpovídajících informací z obrazových dat. Tyto přístupy nejsou zdaleka tak přesné, intuitivní a rychlé jako v případě lidí.

V mé diplomové práci využiji nástroje poskytované počítačovým viděním a pokusím se přiblížit lidskému vnímání prostoru, konkrétně 3D reprezentaci trajektorií.

1.1 Popis problému

Diplomová práce je vyústěním semestrálních projektů absolvovaných během dvou let studia. Cílem bylo vytvořit sadu programů umožňujících rekonstrukci 3D trajektorie pohybu barevného objektu v prostoru pomocí dvou kamer. Tento relativně složitý problém lze rozdělit na několik dílčích, které jsou popsány níže.

- Nejprve je nutné zvolit použité kamery a jejich prostorové umístění. Definovat podmínky kalibrace, zvolit kalibrační vzor a vhodné vstupní informace. Určit relevantní výstupy použitelné následujícími systémy. Zvolit vhodnou strukturu uložení výstupů. Učinit návrh a implementaci systému pro stereo kalibraci kamer. Zhodnotit a ověřit správnost kalibrace kamer.
- Další část řeší automatické trénování barevného modelu objektu. Je nutné zvolit tvar, členitost a barvu objektu. Datovou a barevnou reprezentaci modelu objektu. Dále zvolit vhodný přístup. Navrhnout a implementovat systém automatického trénování barevného modelu objektu, vizualizovat a ověřit jeho správnost. Navrhnout způsob vylepšení barevného modelu a uložit v odpovídající datové reprezentaci.

- Zvolit, popřípadě navrhnout vhodnou metodu sledování barevného objektu na základě výstupů z předešlých systémů.
- Závěrečná část je tvořena návrhem systému pro 3D rekonstrukci obrazových dat sledovaného objektu. Je nutno využít výstupů z předešlých systémů, informace o poloze a 2D tvaru objektu. Provést rekonstrukci dat do 3D souřadnic. Vizualizovat 3D data. Navrhnout metody pro reprezentaci 3D objektu odpovídajícím jednobodovým zastoupením. Uložit, reprezentovat a vizualizovat výsledné trajektorie pohybu. Zhodnotit úspěšnost 3D rekonstrukce.

1.2 Současný stav řešeného problému

Počítačové vidění [1] je relativně nový, dynamický a rychle se rozvíjející vědní obor, zabývající se oblastmi řízení procesů, navigace, modelování objektů a prostředí, interakcí, automatickým ověřováním, 3D rekonstrukcí scény, detekcí událostí a mnohým dalším. Jedná se tedy o velmi širokou oblast, ve které vzniklo mnoho obecných přístupů pro řešení dílčích problémů. Některé ze základních přístupů lze nalézt například v [2].

Pro počítačové vidění je velmi důležitá kalibrace kamery. V současné době jsou již zaběhlé přístupy, které využívají kalibrační vzor [3], obecněji popsáno v [2]. Dále je možné využít alternativní metody kalibrace více kamer pomocí laserového ukazátka [4].

V současné době je trénování modelů barevných objektů stále řešená úloha. Obecné přístupy byly prezentovány v [2], [5] a [6]. Úloha byla zkoumána ze stochastického hlediska v [7] a dále pomocí shlukování barev s využitím metody K-means [8].

V oblasti sledování objektů bylo prezentováno mnoho metod, které využívají rozdílné přístupy a specializují se na konkrétní problémy. Proto nelze určit nejlepší metodu pro sledování, základní myšlenky byly prezentovány v [9] a [10]. Obecné přístupy jsou popsány v [2]. Dále byly prezentovány přístupy založené na odhadu gradientu hustotní funkce [11] a metody využívající particle filtr [12].

Stereo vidění a projekce jsou stále aktuální úlohy, které jsou ve světě řešeny předními odborníky v oblasti počítačového vidění. Daný problém 3D rekonstrukce celé obrazové scény je natolik složitý a komplexní, že doposud neexistuje obecný postup nebo metoda, která by detailně rekonstruovala veškerou obrazovou informaci scény. Hlavním problémem je určení korespondujících bodů v jednotlivých pohledech kamer. Jsou dostupné pouze obecné metody, postupy a způsoby využívající se v oblasti zpracování obrazu. Současné aplikace, které řeší 3D rekonstrukci obrazových dat jsou vždy omezené na úzký a konkrétní případ použití. Například postupy zaměřující se na rekonstrukci velkých obrazových scén [13]. Práce zabývající se rekonstrukcí 3D scény s využitím ručních kamer [14]. Nelze tedy říci, že by v současné době bylo možno jednoduše rekonstruovat obrazová data v takové obecnosti, aby se vnímání prostoru daným systémem alespoň vzdáleně přibližovalo lidskému vnímání prostoru.

V současné době se objevují snímače hloubky, které poskytují informaci o hloubce objektů ve scéně v reálném čase. Tyto snímače nejsou v této práci uvažovány.

Z výše uvedeného lze říci, že v oblasti počítačového vidění existují obecné přístupy pro řešení dílčích problémů. Použitím těchto metod a přístupů lze řešit velmi komplexní a složitá zadání, avšak prozatím se nelze rovnat s člověkem.

Kapitola 2

Počítačové 3D vidění a geometrie

Počítačové vidění je odvětví výpočetní techniky a vývoje softwaru zabývající se vytvářením zařízení schopných získávat informaci ze zachyceného obrazu. Může se jednat mimo jiné i o zpracování videa, obrazu z více zdrojů, nebo dat z lékařských vyšetření. Počítačového vidění je široce využíváno například v oblastech ovládnutí procesů, detekce jevů, organizace informací, modelování objektů nebo prostředí a interakce počítače s člověkem [1].

V oblasti počítačového vidění byl zaznamenán rychlý rozvoj v zobrazování geometrie, která se zabývá matematickými vztahy mezi:

- 3D body ve scéně a obecněji jinými jednoduchými geometrickými objekty,
- jejich projekcí,
- vztahy mezi více projekcemi 3D scény.

Z oblasti počítačového vidění se vyvinula **fotogrammetrie**, která se zabývá rekonstrukcí tvarů, měřením rozměrů a určováním polohy předmětů, které jsou zobrazeny na fotografických snímcích. Fotogrammetrii lze definovat jako vědní obor zabývající se zpracováním informací ve fotografiích. Zaměřuje se na poměrně omezený okruh úkolů oproti 3D počítačovému vidění, které zkoumá používání běžně dostupných kamer. Ty jsou buď částečně kalibrované, nebo vůbec ne. Pracuje oproti fotogrammetrii s velkými nepřesnostmi v obraze. Nedávný vývoj umožnil plnou automatizaci úkolů, jako je 3D rekonstrukce bodů a parametrů kamery z neznámé sekvence snímků. Dále automatickou rekonstrukci 3D scény z velkého počtu odlišných pohledů.

Základním senzorem, který poskytuje informaci o okolním světě počítačového 3D vidění je kamera, která zachycuje jeden snímek a nebo sekvenci snímků, tedy video, důraz je kladen hlavně na geometrii. Dále bude vysvětleno jak se dají 2D obrazové informace využít k 3D rekonstrukci. K tomu je potřeba si říci něco o **perspektivní projekci**, která je také nazývána centrální projekcí. V modelu dírkové kamery, popsaného v podkapitole 3.1, je využita perspektivní projekce. Podstatou této projekce je, že dvě rovnoběžné čáry se zobrazí jako sbíhající se.

2.1 Body a přímky v projekčním prostoru

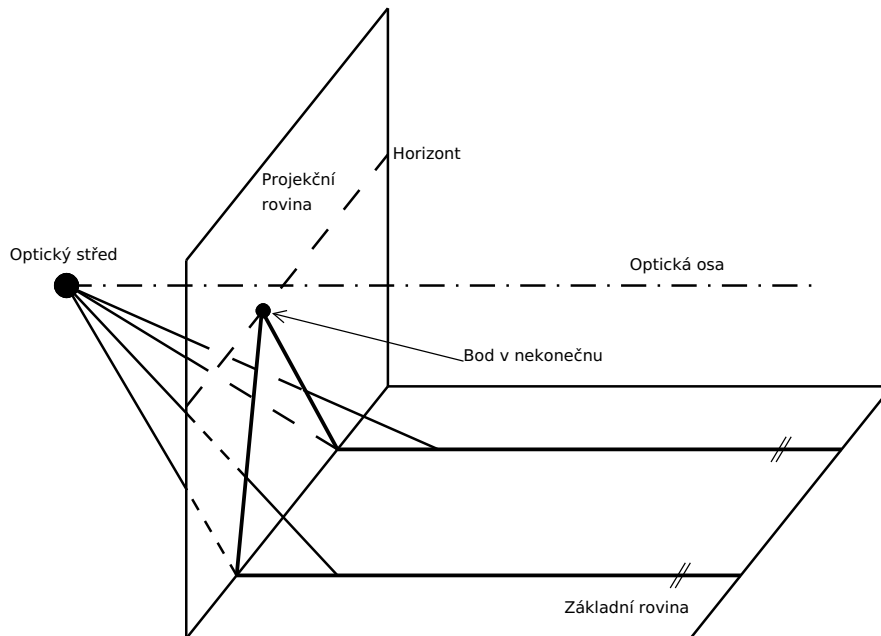
Je nutné začít definicí projekčního prostoru. Uvažuje se $d + 1$ dimenzionální lineární prostor bez jeho počátku $\mathcal{R}^{d+1} - \{[0, \dots, 0]^\top\}$, dále je definován ekvivalentní vztah

$$[x_1, \dots, x_{d+1}]^\top \simeq [x'_1, \dots, x'_{d+1}]^\top, \quad (2.1)$$

právě tehdy, když

$$\exists \alpha \neq 0 : [x_1, \dots, x_{d+1}]^\top = \alpha [x'_1, \dots, x'_{d+1}]^\top. \quad (2.2)$$

To znamená, že dva vektory v \mathcal{R}^{d+1} jsou ekvivalentní, pokud jsou stejné, až na nenulové měřítko vyjádřené konstantou α . **Projekční prostor** \mathcal{P}^d si lze představit jako množinu všech přímek v \mathcal{R}^{d+1} procházejících počátkem.



Obrázek 2.1: Projekční perspektiva rovnoběžných přímek.

Bod v \mathcal{P}^d odpovídá nekonečně rovnoběžným vektorům v \mathcal{R}^{d+1} a je jednoznačně dán jakýmkoliv jedním vektorem v \mathcal{R}^{d+1} . Takový vektor se nazývá **homogenní** (také projekční) reprezentace bodu v \mathcal{P}^d . Homogenní vektor reprezentuje stejný bod jako jakýkoliv vektor, který se liší o nenulové měřítko vyjádřené konstantou α . Měřítko α je často voleno tak, že vektor má na pozici nejvíce vpravo číslo 1, například $[x'_1, \dots, x'_d, 1]^\top$. Více v [2].

Kapitola 3

Model kamery

Modelování je vytváření abstraktní, koncepční, grafické a matematické formy modelu. Ten je vždy pouze přiblížení se skutečnosti, protože skutečný systém může být natolik složitý, že jeho realizace není možná. Dále bude popsán model dírkové kamery, vycházejícího z modelu dírkové komory (camera obscura). Což je optické zařízení používané jako pomůcka malířů, předchůdce fotoaparátu a kamery.

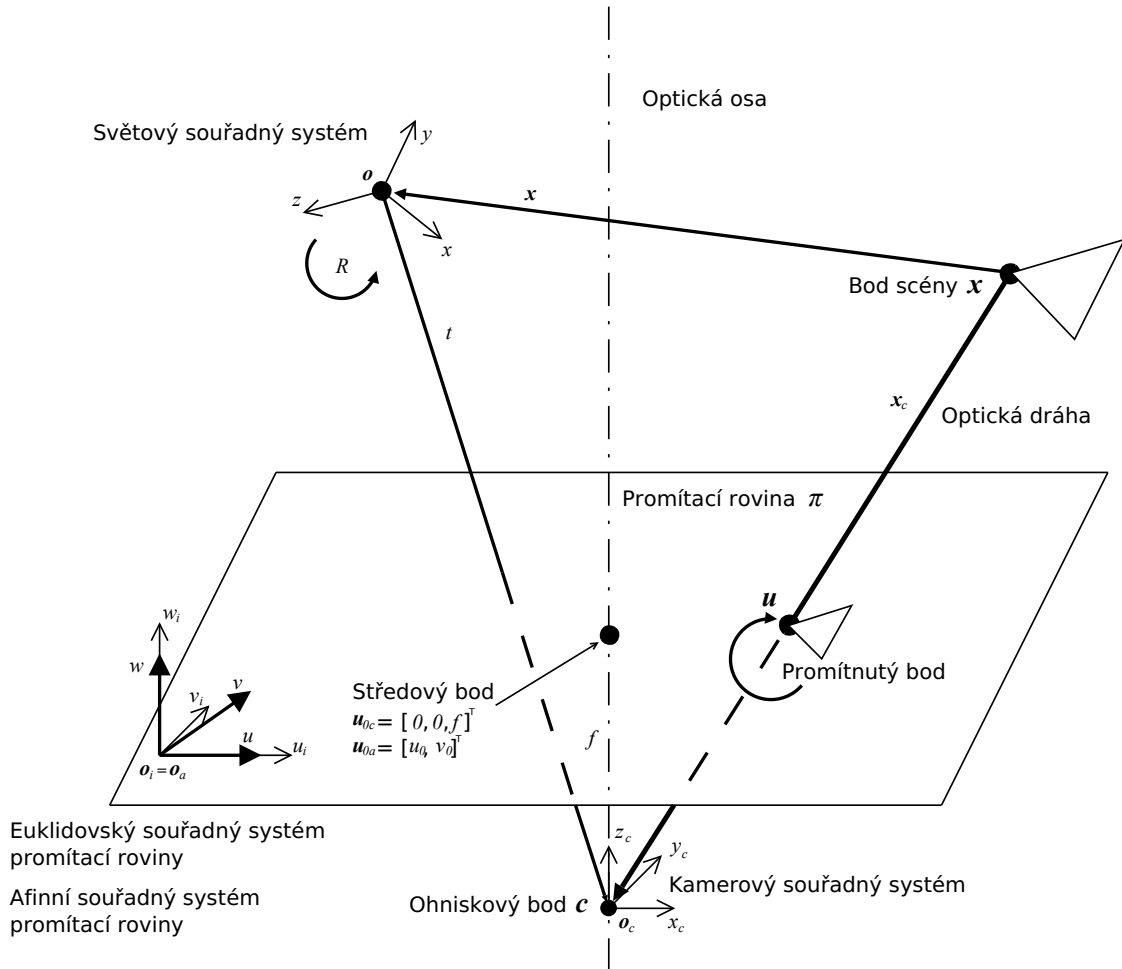
3.1 Model dírkové kamery

Dírková kamera, v angličtině nazývaná jako pinhole camera, je používána jako vhodný model pro mnoho aplikací počítačového vidění. Model dírkové kamery vykonává středovou projekci, geometrie je zobrazena na Obr. 3.1. Rovina π je umístěna horizontálně a je na ni promítán reálný svět. Optická osa prochází středovým bodem a ohniskovým bodem, ten se také nazývá optický střed nebo střed projekce. **Ohnisková vzdálenost** f je jedním z parametrů čočky.

Kamera vykonává lineární transformaci z 3D prostoru \mathcal{P}^3 do 2D prostoru \mathcal{P}^2 . Projekce je přenášena optickým paprskem, který je odražený od bodu scény \mathbf{x} nebo vznikl ze zdroje světla. Optický paprsek projde skrz optický střed \mathbf{c} a na promítací rovině skrz bod \mathbf{u} .

Pro popsání je potřeba čtyř souřadných systémů:

1. Světový souřadný systém, který má střed v bodu \mathbf{o} . Body \mathbf{x} a \mathbf{u} jsou vyjádřeny ve světovém souřadném systému.
2. Kamerový souřadný systém (s indexem c) má střed v bodu $\mathbf{c} \equiv \mathbf{o}_c$. Tento bod je také ohniskem. Osa z_c je sjednocena s optickou osou. Její orientace je z ohniskového bodu \mathbf{c} směrem k promítací rovině. Je získán vztah mezi světovým a kamerovým souřadným systémem, který je dán Euklidovskou transformací skládající se z rotace R a translace \mathbf{t} .



Obrázek 3.1: Geometrie lineární perspektivy kamery.

3. Euklidovský souřadný systém (s indexem i) promítací roviny má osy zarovnané s kamerovým souřadným systémem. To znamená, že osy u_i , v_i , w_i jsou kolinéární se souřadným systémem x_c , y_c , z_c . Osy u_i a v_i leží na promítací rovině.
4. Afinní souřadný systém (s indexem a) je tvořen souřadnými osami u , v , w a počátkem \mathbf{o}_a , který je totožný s Euklidovským souřadným systémem \mathbf{o}_i . Souřadné osy u , w jsou zarovnané s u_i , w_i . Naopak osa v má jinou orientaci vzhledem k ose v_i .

Důvod proč je zaveden afinní souřadný systém je, že body se mohou zobrazovat zdeformovaně a souřadné osy nemusí mít stejná měřítka.

Projekční transformace se dá rozdělit do tří jednoduchých transformací, které odpovídají jednotlivým transformacím mezi těmito čtyřmi souřadnými systémy.

První transformace (mezi 1 a 2 výše) tvoří přechod ze světového souřadného systému (\mathbf{o} ; x , y , z) do kamerového souřadného systému (\mathbf{o}_c ; x_c , y_c , z_c). Světový souřadný systém může být zarovnán s kamerovým souřadným systémem, translací středu \mathbf{o} do \mathbf{o}_c o vektor \mathbf{t} a rotací souřadných os o rotační matici R .

Transformace bodu \mathbf{x} do \mathbf{x}_c vyjádřená v nehomogenních souřadnicích je

$$\mathbf{x}_c = R(\mathbf{x} - \mathbf{t}). \quad (3.1)$$

Rotační matice R vyjadřuje tři základní rotace kolem jednotlivých os x , y a z . Translační vektor \mathbf{t} vyjadřuje tři složky translace počátku světového souřadného systému s ohledem na kamerový souřadný systém. Nyní je získáno šest vnějších kamerových parametrů, tři rotační a tři translační.

Parametry R a \mathbf{t} se nazývají **vnější kalibrační parametry kamery**.

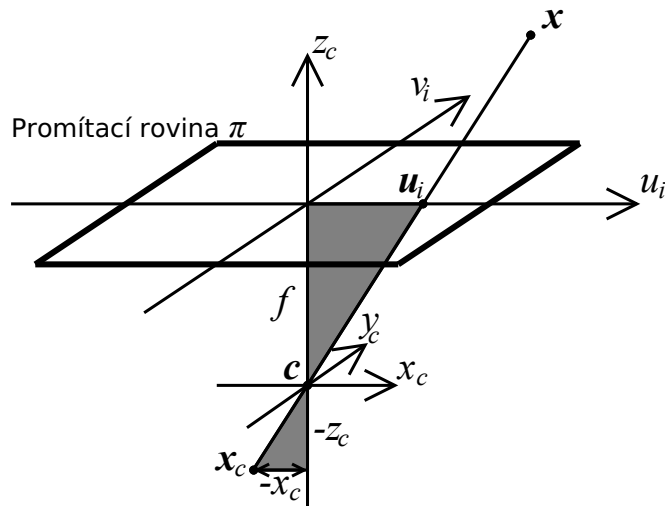
Dále lze rovnici (3.1) přepsat na tvar

$$\mathbf{x}_c = R\mathbf{x} - R\mathbf{t}, \quad (3.2)$$

který je po přepsání do homogenních souřadnic

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R & -R\mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}. \quad (3.3)$$

Druhá transformace (mezi 2 a 3 výše) promítá bod \mathbf{x}_c z 3D scény, který je vyjádřený v kamerovém středovém souřadném systému $(\mathbf{o}_c; x_c, y_c, z_c)$, do bodu \mathbf{u}_i . Tento bod ležící na promítací rovině π je vyjádřen pomocí Euklidovského souřadného systému promítací roviny $(\mathbf{o}_i; u_i, v_i, w_i)$, viz Obr. 3.2 níže.



Obrázek 3.2: Geometrie podobnosti trojúhelníků.

Projekce $\mathcal{R}^3 \rightarrow \mathcal{R}^2$ v nehomogenních souřadnicích je dána dvěma nelineárními rovnicemi, které plynou z podobnosti trojúhelníků na Obr. 3.2

$$\frac{u_i}{-x_c} = \frac{f}{-z_c}, \quad (3.4)$$

$$\frac{v_i}{-y_c} = \frac{f}{-z_c}, \quad (3.5)$$

po úpravě lze psát

$$u_i = \frac{fx_c}{z_c}, \quad (3.6)$$

$$v_i = \frac{fy_c}{z_c}, \quad (3.7)$$

kde f je ohnisková vzdálenost. Jestliže je projekce dána rovnicemi (3.6) a (3.7) v projekčním prostoru, potom projekci $\mathcal{P}^3 \rightarrow \mathcal{P}^2$ lze zapsat lineárně v homogenních souřadnicích jako

$$[\lambda u_i, \lambda v_i, \lambda]^\top = [fx_c, fy_c, z_c]^\top, \quad (3.8)$$

tento zápis může být převeden do maticové formy následovně

$$\lambda \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}, \quad (3.9)$$

kde $\lambda = z_c$ je homogenní koeficient změny měřítka. V případě, že je ohnisková vzdálenost $f = 1$, pak rovnice (3.9) se změní na tvar

$$\lambda \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}, \quad (3.10)$$

který je někdy nazýván kamera s normalizovanou promítací rovinou.

Třetí transformace (mezi 3 a 4 výše) mapuje Euklidovský souřadný systém $(\mathbf{o}_i; u_i, v_i, w_i)$ na afinní souřadný systém $(\mathbf{o}_a; u, v, w)$. Jedná se o speciální případ afinní $\mathcal{P}^2 \rightarrow \mathcal{P}^2$ transformace.

Většina současných zobrazovacích systémů definuje počátek systému souřadnic v levém horním rohu obrazu. Dříve bylo předpokládáno, že se počátek nachází v bodě, kde optická osa prochází skrze promítací rovinu π . Proto lze tuto skutečnost, ve formě posunutí počátku do bodu $\mathbf{u}_{0a} = [u_0, v_0]^\top$, zapsat rovnicí

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix}. \quad (3.11)$$

Pro odvození rovnice (3.11), bylo předpokládáno, že pixely obrazového snímače jsou čtvercové, tedy poměr stran je $1 : 1$ a pixely nejsou šikmé (osy pixelu svírají pravý úhel). Tyto parametry vyplývají z fyzikálního zpracování kamery. Nicméně, předpoklady nejsou vždy naplněny, proto je nutné, je do rovnice (3.11) zanést pomocí parametrů η a τ . První parametr η vyjadřuje poměr stran pixelu poměrem $1 : \eta$, tedy pokud je $\eta = 1$, pak jsou pixely dokonale čtvercové. Druhý parametr τ

určuje sešikmenost os pixelu, pokud je $\tau = 0$, tak jsou osy pixelů dokonale pravoúhlé. Tyto skutečnosti ve spojení s rovnicí (3.11) dávají výsledný tvar

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & \tau & u_0 \\ 0 & \eta f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = K \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix}, \quad (3.12)$$

kde K označuje matici **vnitřních kalibračních parametrů kamery**, má velikost 3×3 , je horní trojúhelníková a vyjadřuje mapování $\mathcal{P}^2 \rightarrow \mathcal{P}^2$, které je speciálním případem afinní transformace. Parametry matice K jsou závislé na samotné fyzikální realizaci kamery. Nelineární vnitřní parametry kamery ve formě zkreslení objektivu jsou také důležité, avšak nemohou být zahrnuty v lineárním modelu kamery popsané maticí K .

Nyní je možno zcela obecně specifikovat projekci dírkové kamery. Jedná se o lineární transformaci z 3D projekčního prostoru \mathcal{P}^3 do 2D projekčního prostoru \mathcal{P}^2 . Pro získání 2D pozice bodu \mathbf{u} na promítací rovině z 3D bodu \mathbf{x} ve světových souřadnicích je nutné využít rovnice (3.12), (3.10) a (3.3) pro přepis na tvar

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & -R\mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \quad (3.13)$$

což lze psát jako

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = [K \quad \mathbf{0}] \begin{bmatrix} R & -R\mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}. \quad (3.14)$$

Součinem prvního a druhého koeficientu v rovnici (3.14) lze získat užitečnou vnitřní strukturu

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K [R \quad -R\mathbf{t}] \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = M \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}. \quad (3.15)$$

Pokud se vyjádří bod 3D scény v homogenních souřadnicích, může se psát projekce perspektivy v lineární podobě pomocí jediné 3×4 matice M . Je nazývána **projekční maticí** (nebo také matice kamery) a obsahuje všechny vnitřní a vnější kalibrační parametry kamery

$$M = [KR \quad -KR\mathbf{t}]. \quad (3.16)$$

V této kapitole byly čerpány informace ze zdroje [2].

Kapitola 4

3D vidění a projektivní geometrie

Kalibrace je soubor úkonů, kterými se za specifikovaných podmínek stanoví vztah mezi hodnotami veličin. Tyto jsou indikovány měřicím systémem, měřicím přístrojem, hodnotami reprezentovanými ztělesněnou mírou, nebo referenčním materiálem a odpovídajícími hodnotami, které jsou realizovány etalony (standards) [15].

Kalibrace je postup, jímž lze odhadnout, za pomoci výpočtů, projekční matice jednotlivých kamer. Kalibraci lze provádět s využitím jedné nebo více kamer. Například při použití dvou kamer mluvíme o takzvané stereo kalibraci, která přináší více obrazové informace než při kalibraci jedné kamery. Na druhou stranu je složitější.

Bude ukázáno, jak postup kalibrace aplikovat na jednu kameru. Následovně bude popsána epipolární geometrie.

4.1 Kalibrace jedné kamery

Při kalibraci jedné kamery je potřeba nalézt projekční matici M z rovnice (3.16), která obsahuje všechny vnitřní a vnější parametry. Nyní se musí určit, kolik parametrů v matici M je volných.

Z tvaru rovnice získané v předešlé kapitole

$$M = [KR \quad -KRt], \quad (4.1)$$

lze říci, že projekční matice M má 5 volných parametrů z matice K , dále 3 volné parametry z rotační matice R a 3 z translačního vektoru t . Celkem tedy 11 volných parametrů, které je nutné odhadnout.

Dále je nutné si uvědomit, že projekce 3D bodu určeného světovým souřadným systémem do korespondujícího 2D bodu v afiním souřadném systému projekční roviny bude stejná pro všechny body ležící na optickém paprsku. Tento paprsek je určen ohniskovým bodem c a bodem na projekční rovině u . Proto lze jinými slovy říci, že všechny body ležící na tomto optickém paprsku se promítnou do stejného bodu a nelze tedy určit jejich hloubkovou souřadnici, což vyjadřuje parametr λ v rovnicích (3.8) - (3.15).

4.1.1 Kalibrace kamery ze známé scény

Nyní bude detailněji popsána kalibrace kamery ze známé scény, kde musí být brán ohled na to, že tento proces se dělí na dvě části. V první části je odhadnuta projekční matice M , z rovnice (4.1), ze známých souřadnic bodů scény. Ve druhé jsou vnější a vnitřní parametry kamery odhadnuty z projekční matice M .

V prvním kroku je získána matice M následovně. Pro každý bod $\mathbf{x} = [x, y, z, 1]^T$ ze známé 3D scény a jeho korespondující 2D bod $\mathbf{u} = [u, v, 1]^T$ ležící na projekční rovině vznikne jedna rovnice. V projekční matici M , která má velikost 3×4 , je nutno nalézt číselné hodnoty m_{ij} . Jednoduchými úpravami rovnice (4.2) bude získán tvar rovnice (4.4)

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \quad (4.2)$$

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11}x & m_{12}y & m_{13}z & m_{14} \\ m_{21}x & m_{22}y & m_{23}z & m_{24} \\ m_{31}x & m_{32}y & m_{33}z & m_{34} \end{bmatrix}, \quad (4.3)$$

$$\begin{aligned} u(m_{31}x + m_{32}y + m_{33}z + m_{34}) &= m_{11}x + m_{12}y + m_{13}z + m_{14}, \\ v(m_{31}x + m_{32}y + m_{33}z + m_{34}) &= m_{21}x + m_{22}y + m_{23}z + m_{24}. \end{aligned} \quad (4.4)$$

Takto získané dvě lineární rovnice, které obsahují 12 neznámých m_{11}, \dots, m_{34} pro každý bod ze známé scény a k němu korespondující 2D bod.

Pokud n bude počet dvojic korespondujících bodů, mohou být rovnice (4.4) přepsány do maticového reprezentace. V ní figuruje známá matice o velikosti $2n \times 12$ a vektor neznámých parametrů m_{11}, \dots, m_{34}

$$\begin{bmatrix} x & y & z & 1 & 0 & 0 & 0 & 0 & -ux & -uy & -uz & -u \\ 0 & 0 & 0 & 0 & x & y & z & 1 & -vx & -vy & -vz & -v \\ & & & & & & & & \vdots & & & \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{12} \\ \vdots \\ m_{34} \end{bmatrix} = 0. \quad (4.5)$$

Matice M má pouze 11 neznámých parametrů, jak bylo ukázáno v podkapitole (4.1). Pro každou dvojici korespondujících bodů $(\mathbf{u}_i, \mathbf{x}_i)$ je potřeba vyřešit homogenní lineární rovnici

$$\lambda_i \mathbf{u}_i = M \mathbf{x}_i, \quad i = 1, 2, \dots, m, \quad (4.6)$$

pro M a λ_i . Matice M má určenou velikost, proto má pouze 11 volných parametrů. Pro $m = 5$ je systém podurčen a pro $m = 6$ je naopak přeurčen. Tedy je potřeba minimálně $m = 6$ dvojic korespondujících bodů pro výpočet matice M . Většinou

je použito více bodů a přeúčřená rovnice (4.5) je vyřešena pomocí robustní metody nejmenších čtverců, která je použita pro minimalizaci chyby vzniklé šumem. Výsledkem výpočtu je projekční matice M .

Ve druhém kroku je potřeba oddělit vnější parametry (rotační matici R a translační vektor \mathbf{t}) z odhadnuté projekční matice M , kterou lze zapsat úpravou rovnice (3.16) jako

$$M = [KR \quad -KR\mathbf{t}] = [A \quad \mathbf{b}]. \quad (4.7)$$

Submatice 3×3 je označována jako A a vektor nejvíce vpravo jako \mathbf{b} .

Výpočet translačního vektoru je jednoduchý, je možno nahradit $A = KR$ a $\mathbf{b} = -A\mathbf{t}$ v rovnici (4.7), pak $\mathbf{t} = -A^{-1}\mathbf{b}$.

Pro určení rotační matice R se musí brát v úvahu, že kalibrační matice K je horní trojúhelníková a rotační matice R je ortogonální. Pro rozklad matice A se používá metoda nazývaná QR rozklad, jejíž výsledkem jsou přesně takové tvary matic K a R jaké jsou hledány.

Alternativně by mohl být použit singulární rozklad (SVD), který se řadí mezi hlavní nástroje pro řešení geometrických problémů spojených s 3D viděním.

Informace byly čerpány ze zdroje [3].

4.1.2 Další metody kalibrace

Mezi další metody kalibrace patří například metoda, kterou lze využít pro kalibraci více kamer. Tato metoda se specializuje na kalibraci tří a více kamer (neexistuje horní limit), kamery však nemusí být nutně stejných parametrů. Ke kalibraci se využívá laserové ukazovátko pro tvorbu známého bodu, jímž se pohybuje po pracovním prostoru, který má být zkalibrován. Není vyžadováno, aby byl barevný bod přítomen ve všech kamerách, ale kamery by měly rozumně pokrývat pracovní prostor s určitým překrýváním se. Barevný bod z laserového ukazovátko je v obrazových rovinách jednotlivých kamer nalezen s přesností na subpixel. Metoda byla testována až pro kalibraci 16 kamer, kde takto rozsáhlou kalibraci lze provést v čase menším než půl hodiny s chybou odpovídající přibližně 0.2 pixelu. Tato metoda je vhodná pro kalibraci virtuálních prostředí, jelikož neumožňuje kalibraci v reálné metrice. Více informací lze nalézt v [4].

4.2 Dvě kamery, stereo vidění

Největší rozdíl mezi lidským vizuálním systémem a přístupem popsáním v kapitole 3 je, že člověk má dvě oči a proto získává více obrazové informace, než v případě jedné kamery. Již ve viktoriánských časech bylo normální využít dvou mírně odlišných pohledů pro vytvoření 3D iluze, následně vyvrcholením v padesátých letech minulého století v podobě 3D filmů. Pokud je 3D scéna složena ze dvou mírně odlišných pohledů, dvou očí, je možné získat informaci o hloubce. Za předpokladu, že je lidský mozek schopný automaticky nalézt korespondující body ve scéně.

Stereo vidění má obrovský význam. Vyvolalo mnoho výzkumů v počítačových vizuálních systémech se dvěma vstupy z kamer, které využívají znalost jejich relativní geometrie k odvození hloubkové informace ze dvou získaných pohledů.

Kalibrované kamery a znalost souřadnic jednoho obrazového bodu v první kameře umožňuje určit paprsek v obrazovém prostoru druhé kamery jednoznačně. Pokud dvě kalibrované kamery pozorují stejný bod \mathbf{x} ve scéně, pak mohou být jeho 3D souřadnice vypočítány jako průsečík dvou paprsků. To je základním principem **stereo vidění**, které typicky obsahuje tři kroky:

1. kalibraci kamer,
2. stanovení bodové korespondence mezi dvojicemi bodů z levého a pravého obrazu a
3. rekonstrukci 3D souřadnic bodů ve scéně.

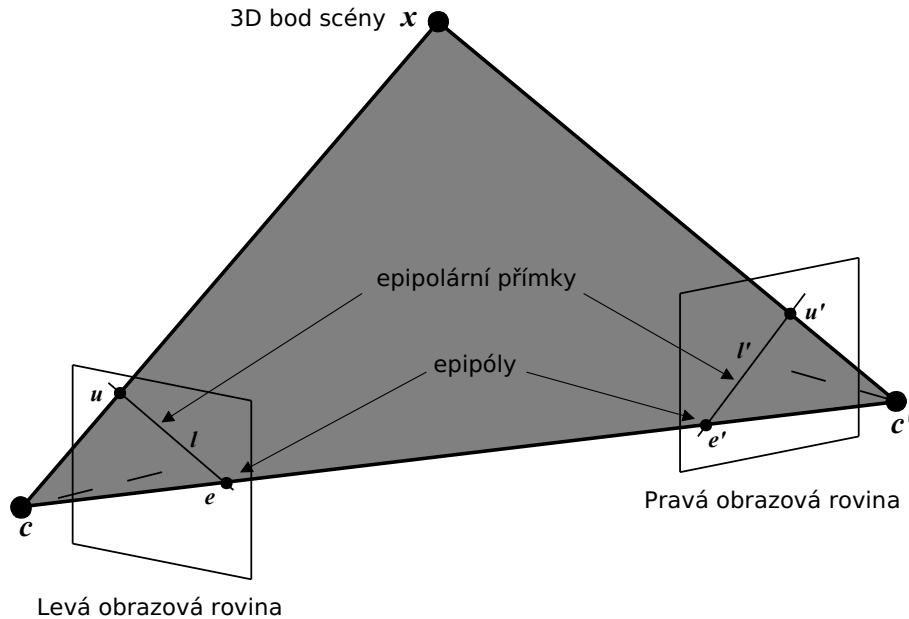
V této kapitole budou označovány matematické entity vztahující se k prvnímu snímku bez apostrofu a stejný subjekt vztahující se k druhému obrázku s apostrofem '. Například \mathbf{u} a \mathbf{u}' .

4.2.1 Epipolární geometrie

Epipolární geometrie je vnitřní projektivní geometrie mezi dvěma pohledy. Je nezávislá na struktuře scény, závisí na vnitřních parametrech a relativní poloze kamery.

Geometrie systému se dvěma kamerami je ukázána na Obr. 4.1. Přímka spojující optické středy \mathbf{c} a \mathbf{c}' se nazývá spojnice kamerových středů (baseline). Baseline protíná obrazovou rovinu v **epipólech** \mathbf{e} a \mathbf{e}' . **Epipól** je bod průniku spojnice kamerových středů s obrazovou rovinou, nebo-li epipól je obrazem jednoho kamerového středu v druhém pohledu, $\mathbf{e} = M\mathbf{c}'$ a $\mathbf{e}' = M'\mathbf{c}$.

Jakýkoliv bod scény \mathbf{x} pozorovaný dvěma kamerami a dvěma odpovídajícími paprsky z optických středů \mathbf{c} , \mathbf{c}' definují **epipolární rovinu**. Epipolární rovina je rovina, která obsahuje spojnici kamerových středů. Tato rovina protíná obrazovou rovinu v epipolárních přímkách (epipolárách) \mathbf{l} a \mathbf{l}' . Alternativně je epipolární přímka projekcí paprsku z jedné kamery do druhé. Všechny epipolární přímky se protínají v epipólu.



Obrázek 4.1: Geometrie systému se dvěma kamerami.

Nechť \mathbf{u} a \mathbf{u}' je projekce bodu 3D scény \mathbf{x} v první respektive druhé obrazové rovině. Paprsek $\mathbf{c}\mathbf{x}$ reprezentuje všechny možné pozice bodu \mathbf{x} pro první kameru (levá obrazová rovina) a je zobrazen jako epipolární přímka \mathbf{l}' v druhé kameře (pravá obrazová rovina). Bod \mathbf{u}' v druhé obrazové rovině, který koresponduje k bodu \mathbf{u} musí ležet na epipolární přímce \mathbf{l}' v druhé obrazové rovině, tedy $\mathbf{l}'^\top \mathbf{u}' = 0$. Situace je samozřejmě symetrická, proto platí $\mathbf{l}^\top \mathbf{u} = 0$. Skutečnost, že poloha dvou odpovídajících obrazových bodů není libovolná je známá jako **epipolární omezení**. Jinými slovy, se prohledávání korespondujícího bodu zúží z prohledávání roviny na prohledávání úsečky, což velmi zjednoduší daný problém.

Matici F , reprezentující lineární zobrazení, lze nyní napsat

$$\mathbf{l}' = F\mathbf{u}, \quad (4.8)$$

pro získání podmínky pro korespondenci dvou bodů v odpovídajících kamerách, je nutno využít rovnici $\mathbf{l}'^\top \mathbf{u}' = 0$ a tak získat

$$\mathbf{u}'^\top F\mathbf{u} = 0. \quad (4.9)$$

Rovnice (4.9) je **epipolární omezení** v algebraické formě. Pan Longuet-Higgins v roce 1981 byl první, kdo v počítačovém vidění objevil tento bilineární vztah, přestože byl znám z fotogrammetrie z konce 19. století. Matice F je nazývána **fundamentální maticí**, což je poněkud zavádějící název, který je používán z historických důvodů. Vhodnějším pojmenováním je **bifokální matice**. Tento název je používán některými komunitami počítačového vidění.

Fundamentální matice je velmi důležitá pro geometrii využívající více pohledů. Obsahuje veškeré informace, které mohou být získány o dvojici kamer pouze z korespondencí.

4.2.2 Fundamentální matice

Fundamentální matice je algebraickou reprezentací epipolární geometrie. Jedná se o matici velikosti 3×3 s hodnotí 2. Dále víme, že pro každý bod scény \mathbf{u} v jednom obraze existuje odpovídající epipolární přímka \mathbf{l}' v druhém obraze. Tedy všechny korespondující body \mathbf{u}' k bodu \mathbf{u} musí ležet na epipolární přímce \mathbf{l}' , tedy platí vztah uvedený výše

$$\mathbf{l}' = F\mathbf{u}, \quad (4.10)$$

jedná se tedy o zobrazení

$$\mathbf{u} \mapsto \mathbf{l}', \quad (4.11)$$

které reprezentuje vzájemnou korelaci vyjádřenou fundamentální maticí F .

Pokud je bod ve 3D prostoru v prvním pohledu označen \mathbf{u} a ve druhém \mathbf{u}' , pak jsou obrazové body svázány rovnicí

$$\mathbf{u}'^\top F\mathbf{u} = 0. \quad (4.12)$$

Tato rovnice je využívána k řešení fundamentální matice F .

4.2.3 Osmibodový algoritmus

Osmibodový algoritmus je často využívaný algoritmus pro výpočet fundamentální matice. Fundamentální matice je definována rovnicí

$$\mathbf{u}'^\top F \mathbf{u} = 0. \quad (4.13)$$

Výpočet se provádí na základě minimálně osmi korespondujících bodů v obrazových rovinách kamer. Jeho výhodou je snadná implementace a použitelnost, naopak mezi jeho hlavní nevýhody patří velká citlivost na obrazový šum. Proto je lepší využít větší počet bodů pro minimalizaci chyby.

Nyní bude nastíněn osmibodový algoritmus, necht' je $m \geq 8$ korespondujících stereo párů bodů $(\mathbf{u}_i, \mathbf{u}'_i)$ v homogenním souřadném systému, tedy je řešena soustava rovnic

$$\mathbf{u}'_i{}^\top F \mathbf{u}_i = 0, \quad i = 1, \dots, m. \quad (4.14)$$

Rovnici (4.14) lze přepsat na tvar

$$\mathbf{u}'^\top F \mathbf{u} = \left[\mathbf{u}^\top \otimes \mathbf{u}'^\top \right] \mathbf{f} = \begin{bmatrix} uu' & uv' & uw' & vu' & vv' & vw' & wu' & wv' & ww' \end{bmatrix} \mathbf{f} = \mathbf{0}, \quad (4.15)$$

kde

$$\mathbf{f} = \begin{bmatrix} f_{11} & f_{12} & \dots & f_{23} & f_{33} \end{bmatrix}^\top \quad (4.16)$$

a \otimes je Kroneckerovo násobení. Vzhledem ke všem m korespondujícím bodům je možno psát

$$\begin{bmatrix} \mathbf{u}_1^\top \otimes \mathbf{u}'_1{}^\top \\ \dots \\ \mathbf{u}_m^\top \otimes \mathbf{u}'_m{}^\top \end{bmatrix} \mathbf{f} = W \mathbf{f} = \mathbf{0}. \quad (4.17)$$

Pro $m = 8$ korespondujících bodů má rovnice (4.17) jediné řešení. Pro $m > 8$ korespondujících bodů je soustava rovnic přeuredená a proto je nutné využít metodu rozkladu na singulární čísla (SVD). Tato metoda minimalizuje algebraickou vzdálenost, tedy v obou případech ($m = 8$ a $m > 8$) je získáno nejlepší řešení.

Fundamentální matice F vypočtená pomocí osmibodového algoritmu nebude obecně singulární, což znamená, že se nejedná o pravou fundamentální matici. Může být nalezena matice \tilde{F} s hodnotou 2, která je nejbližší k F při reprezentování Frobeniovi normy: $F = UDV^\top$ (rozklad pomocí SVD). Dále se nastaví nejmenší singulární číslo v diagonální matici D na nulu a vznikne nová matice \tilde{D} pouze se dvěma nenulovými pozicemi. Na závěr je nutno spočítat novou matici \tilde{F} jako $\tilde{F} = U\tilde{D}V^\top$ [2].

Jak bylo zmíněno výše, algoritmus je velmi citlivý na obrazový šum, proto byl vylepšen následujícím způsobem. Předpoklad je, že souřadnice \mathbf{u} v jednom obrázku je nahrazena $\hat{\mathbf{u}} = T\mathbf{u}$ a souřadnice \mathbf{u}' v druhém obrázku nahrazena $\hat{\mathbf{u}}' = T'\mathbf{u}'$. Substitucí v rovnici 4.13 vznikne

$$\hat{\mathbf{u}}'{}^\top T'^{-\top} F T^{-1} \hat{\mathbf{u}} = 0, \quad (4.18)$$

kde $T'^{-\top}$ je inverzní transpozice matice T' , dále matice transformace T a T' představují translaci a zvětšení ze starých souřadnic do nových normalizovaných souřadnic. Konkrétně se jedná o translaci bodů tak, aby jejich centroid byl v počátku, tedy $\bar{\mathbf{u}} = 0$. Body jsou následně zvětšeny izotropicky (jednotně ve všech směrech), tedy průměrná vzdálenost od počátku je rovna $\sqrt{2}$.

Z rovnice 4.18 vyplývá, že $T'^{-\top}FT^{-1}$ je fundamentální matice odpovídající bodu korespondencí $\hat{\mathbf{u}}' \leftrightarrow \hat{\mathbf{u}}$, tedy

$$\hat{F} = T'^{-\top}FT^{-1}. \quad (4.19)$$

Což znamená, možnost použít normalizované homogenní souřadnice obrázku pro odhad transformované fundamentální matice použitím standardního osmibodového algoritmu popsaného výše.

Alternativní metoda pro nalezení fundamentální matice byla navržena následovně.

1. Transformovat souřadnice obrázku podle transformací $\hat{\mathbf{u}}_i = T\mathbf{u}_i$ a $\hat{\mathbf{u}}'_i = T'\mathbf{u}'_i$.
2. Najít fundamentální matici \hat{F} odpovídající shodám $\hat{\mathbf{u}}'_i \leftrightarrow \hat{\mathbf{u}}_i$.
3. Nastavit $F = T'^{\top}\hat{F}T$.

Fundamentální matice nalezena touto cestou odpovídá originálním netransformovaným korespondujícím bodům $\mathbf{u}'_i \leftrightarrow \mathbf{u}_i$. Obecně lze říci, že odhad za pomoci normalizovaných obrazových bodů dává lepší fundamentální matici než v případě základního nenormalizovaného přístupu.

Pomocí normalizace souřadnic můžeme vylepšit základní osmibodový algoritmus, který v určitých případech může dosahovat chyb v řádech desítek pixelů, až na úroveň nejlepších iterativních algoritmů. Výhoda tohoto vylepšeného osmibodového algoritmu spočívá v přibližně 20-krát větší rychlosti výpočtu než u ostatních iterativních algoritmů. Algoritmus má velmi jednoduchou programovou realizaci ve srovnání s iterativními metodami [16].

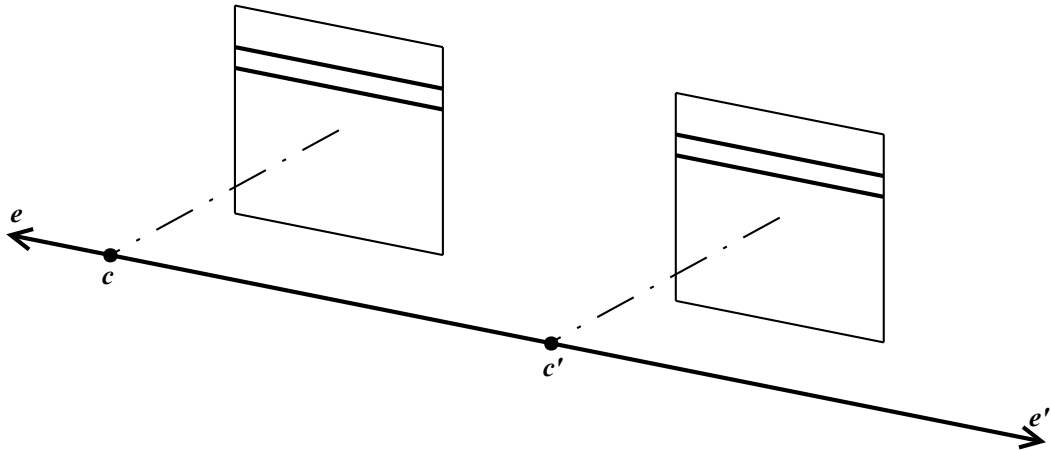
4.3 Rektifikace

Rektifikace je obecně v geometrii definována jako obecný proces transformace polohy všech obrazových prvků z jednoho souřadnicového systému do jiného souřadnicového systému.

Epipolární podmínka redukuje dimenzi prostoru, kterou je potřeba prohledat, pro korespondenci mezi \mathbf{u} v prvním pohledu a \mathbf{u}' v druhém pohledu z 2D problému na 1D problém.

Speciální uspořádání kamer zobrazené na Obr. 4.2 je nazýváno **rektifikovaná konfigurace kamer** (v některých literaturách se používá výraz kanonická konfigurace).

Obrazové roviny kamer splývají a úsečka \mathbf{cc}' je s nimi rovnoběžná. Toto uspořádání pošle epipóly do nekonečna. Dále epipolární přímky odpovídají jednotlivým řádkům



Obrázek 4.2: Rektifikovaná konfigurace dvou pohledů na scénu, kde epipolární přímky jsou orientované rovnoběžně a epipóly leží v nekonečnu.

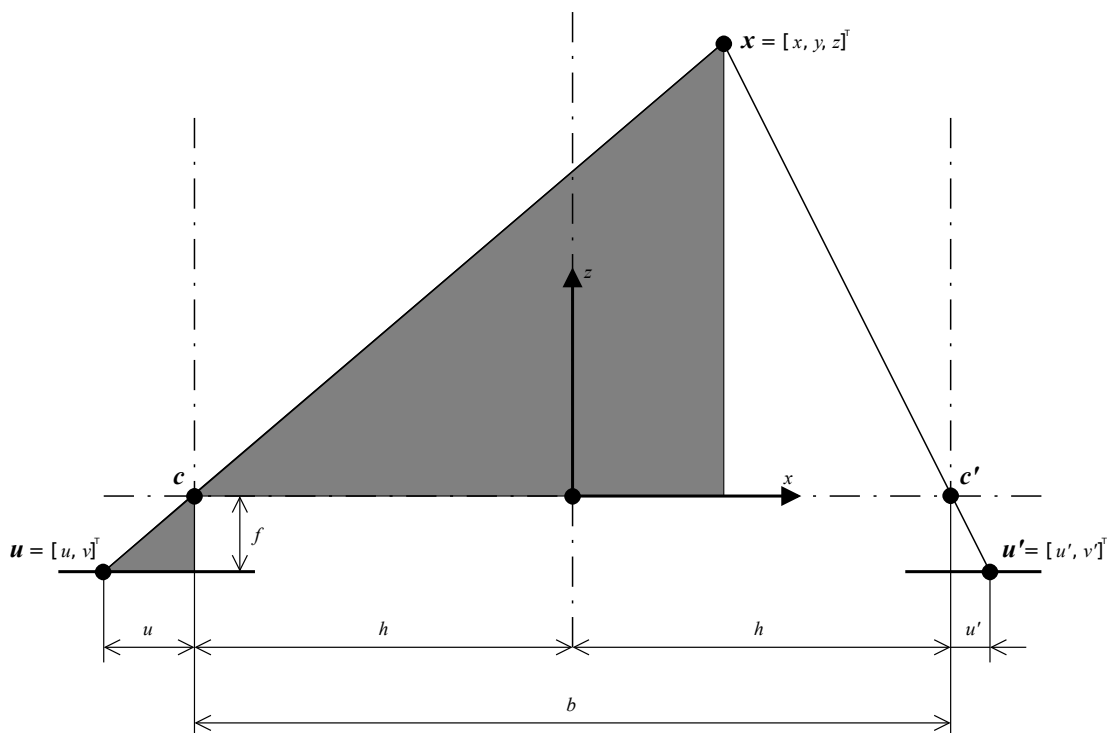
v obrazových rovinách. Dále se předpokládá, že vnitřní parametry kamer jsou stejné. Pro rektifikovanou konfiguraci kamer jsou výpočty značně zjednodušeny. Je často využívána pokud má být stereo korespondence ověřena člověkem, pro kterého je prohledávání určitého řádku jednodušší (tento neautomatický přístup se stále využívá ve fotogrammetrii a při dálkovém průzkumu Země). Podobný závěr lze učinit i pro počítačové programy. Je jednodušší prohledávat podél řádku než podél obecné přímky. Geometrická transformace, která mění obecnou konfiguraci kamer s neparalelními epipolárními přímkami do rektifikované konfigurace kamer se nazývá **rektifikace obrazu**, Obr. 4.3.



Obrázek 4.3: Rektifikovaná dvojice obrázků, ve kterých si jednotlivé řádky odpovídají.

Je uvažována rektifikovaná konfigurace kamer, pro výpočet hloubky obrazu. Optické osy jsou vzájemně paralelní, což vede k zavedení pojmu **disparita**, která je často používána v literatuře zabývající se stereo viděním. Při výpočtu disparity celé scény, se jedná o takzvanou **mapu disparit**. Jednoduchý Obr. 4.4 vystihuje popísanou situaci, kde jsou pohledy dvou kamer, které mají optické osy paralelně ve vzdálenosti $b = 2h$. Obrazy obou kamer snímají 3D bod scény \mathbf{x} se souřadnicemi

$\mathbf{x} = [x, y, z]^T$. Tento bod je promítnutý na obrazové roviny, konkrétně do levé obrazové roviny jako \mathbf{u} a do pravé obrazové roviny jako \mathbf{u}' . V souřadném systému na Obr. 4.4 reprezentuje souřadná osa z vzdálenost od kamer (také počátek, kdy $z = 0$), dále souřadná osa x určuje horizontální vzdálenost (souřadná osa y není zobrazena, určovala by vertikální vzdálenost). Souřadnice $x = 0$ se nachází uprostřed mezi kamerami. Každá obrazová rovina má vlastní lokální souřadnicový systém a lze jej dopočítat z globálního souřadného systému. Z důvodu jednoduchosti obrázku nejsou lokální souřadnicové systémy uváděny. Jedná se o prostou translaci v souřadné ose x z globálního počátku souřadného systému. Hodnoty u, u', v a v' udávají souřadnice v obrazové rovině v lokálním souřadném systému levé a pravé kamery, respektive měření je prováděno ve stejné výšce (řádku), $v = v'$.



Obrázek 4.4: Jednoduchá stereo geometrie v rektifikované konfiguraci kamer.

Hloubka z bodu \mathbf{x} ve 3D scéně může být vypočítána z disparity, která je definována

$$d = u' - u. \quad (4.20)$$

Hodnoty u a u' jsou měřeny ve stejné výšce (řádku), $v = v'$. Nyní lze v následující kapitole 4.4 využít základní geometrie pro odvození hloubkové souřadnice z 3D bodu scény \mathbf{x} . Další přístupy a rozšíření lze nalézt v [17] a [18].

Ve zdroji [19] je popsán postup pro rektifikaci nekalibrované dvojice kamer (nejsou známé vnitřní parametry kamer ani jejich vzájemná poloha).

4.4 3D rekonstrukce

Je nutno si uvědomit, že u , c a c , x jsou podobné pravoúhlé trojúhelníky, viz Obr. 4.4. Hodnoty h a ohnisková vzdálenost f jsou kladná čísla, z je kladná souřadnice a x , u , u' mohou být kladné nebo záporné. Lze napsat rovnice ve tvaru

$$\frac{u}{f} = -\frac{h+x}{z}, \quad (4.21)$$

$$\frac{u'}{f} = \frac{h-x}{z}. \quad (4.22)$$

Vyjádřením x z rovnic (4.21), (4.22) a úpravách přejdou rovnice na tvar

$$z(u' - u) = 2hf, \quad (4.23)$$

ze kterého je vyjádřena z -ová souřadnice jako

$$z = \frac{2hf}{u' - u} = \frac{bf}{u' - u} = \frac{bf}{d}. \quad (4.24)$$

Z rovnice (4.24) je zřejmé, že $d = u' - u$ je disparita pozorovaného 3D bodu scény \mathbf{x} . Pokud $(u' - u) \rightarrow 0$, pak $z \rightarrow \infty$. Nulová disparita indikuje, že pozorovaný bod leží (skutečně) v nekonečnu směrem od pozorovatele, tedy vzdálené 3D body mají malou disparitu. Relativní chyba v hloubce z je velká pro malou hodnotu disparity. Vzdálenost kamer b ovlivňuje relativní chybu v hloubce z , pokud jsou kamery od sebe vzdálenější je chyba menší.

Zbývající dvě souřadnice jsou dopočítány následovně

$$x = \frac{-b(u + u')}{2d}, \quad (4.25)$$

$$y = \frac{bv}{d}. \quad (4.26)$$

V této kapitole byly čerpány informace ze zdroje [2]. V [13] lze nalézt přístup k 3D rekonstrukci velkých scén. Dále je 3D rekonstrukce scény nastíněna v [14].

Kapitola 5

Metoda odečítání pozadí

Metoda odečítání pozadí je technika v oblasti zpracování obrazu a počítačového vidění, při níž dochází k extrakci popředí pro další zpracování (například rozpoznávání objektů atd.). Obecně tvoří oblasti zájmu objekty (lidé, auta atd.), které jsou v popředí. Po fázi předzpracování, která může například obsahovat odstranění šumu, je nasazena metoda odečítání pozadí a její výstupy mohou sloužit jako vstupy pro další metody lokalizace objektu. Odečítání pozadí je široce používaný přístup pro detekci pohybujících se objektů ve videích ze statických kamer. Princip metody spočívá v detekci pohybujících se objektů z rozdílu mezi aktuálním snímkem a referenčním snímkem, kterým může být například obrázek pozadí. Tato metoda se většinou používá v případě, že je snímek součástí video sekvence. V této kapitole byly čerpány informace z [2], [5] a [6].

5.1 Obecný přístup

V této podkapitole budou stručně nastíněny některé základní přístupy k metodě odečítání pozadí, dále zde bude označováno popředí jako F (Foreground) a pozadí obdobně jako B (Background). Všechny metody spojuje základní myšlenka, spočívající ve statickém pozadí a pohybujících se objektech v popředí. Za předpokladu, že pohybující se objekt má v čase t barvu (nebo rozložení barev) lišící se od pozadí B může být tento princip shrnut do následujícího vzorce,

$$F_t(s) = \begin{cases} 1 & \text{když } d(I_{s,t}, B_s) > \tau, \\ 0 & \text{jinak.} \end{cases} \quad (5.1)$$

Kde $F_t(s)$ je popředí F v čase t na pozici pixelu s , $d(I_{s,t}, B_s)$ označuje vzdálenost mezi aktuálním obrázkem I v čase t na pozici pixelu s a obrázkem pozadí B na pozici pixelu s , τ je hodnota prahu. Největší rozdíl mezi většinou metod odečítání pozadí je, jakým způsobem je modelováno pozadí B a jaká vzdálenostní metrika je použita pro výpočet d . V následujících podkapitolách bude představeno několik základních metod.

5.1.1 Základní přístup

Nejjednodušší cesta jak lze získat model pozadí B je použít obrázek ve stupních šedi nebo barevný obrázek, který neobsahuje žádné pohybuující se objekty v popředí F . Za účelem udržení stále aktuálního modelu pozadí může být využito následujícího iterativního postupu aktualizace modelu pozadí B ,

$$\mathbf{B}_{s,t+1} = (1 - \alpha)\mathbf{B}_{s,t} + \alpha\mathbf{I}_{s,t}, \quad (5.2)$$

kde α je aktualizací konstanta, jejíž hodnota může nabývat hodnot $\alpha \in \langle 0, 1 \rangle$. V extrémních případech pro $\alpha = 1$ nebude brán model pozadí $B_{s,t}$ v potaz, naopak v případě $\alpha = 0$ se model pozadí nebude aktualizovat. Pixely náležící do popředí mohou být detekovány prahem s různými metrikami, například

$$d_0 = |I_{s,t} - B_{s,t}|, \quad (5.3)$$

$$d_1 = |I_{s,t}^R - B_{s,t}^R| + |I_{s,t}^G - B_{s,t}^G| + |I_{s,t}^B - B_{s,t}^B|, \quad (5.4)$$

$$d_2 = (I_{s,t}^R - B_{s,t}^R)^2 + (I_{s,t}^G - B_{s,t}^G)^2 + (I_{s,t}^B - B_{s,t}^B)^2, \quad (5.5)$$

$$d_\infty = \max\{|I_{s,t}^R - B_{s,t}^R|, |I_{s,t}^G - B_{s,t}^G|, |I_{s,t}^B - B_{s,t}^B|\}, \quad (5.6)$$

kde exponenty R , G a B reprezentují jednotlivé barevné kanály: červený, zelený a modrý (Red, Green, Blue). Metrika d_0 je speciálně pro obrázky v odstínech šedi.

5.1.2 Filtrace mediánem

Filtrace mediánem patří mezi nelineární filtraci, která vybírá z blízkého, většinou uživatelem definovaného okolí hodnotu mediánu, kterou dosadí na aktuální pozici. Často se využívá ve fázi předzpracování pro zlepšení výsledků následného zpracování. Tento filtr je velmi účinný při odstraňování zrnitosti/šumu v obraze. Jeho nevýhodou je, že může měnit tvary hran objektů. Po fázi předzpracování se využije postup naznačený v (5.1.1).

5.1.3 Gaussovské rozložení

Tato metoda je založená na modelování každého pixelu pozadí pomocí funkce pravděpodobnosti, která je učena pomocí sekvence trénovacích snímků pozadí. V tomto případě základní problém určení prahu přechází v problém nalezení prahu pro funkce pravděpodobností. S přihlédnutím k obrazovému šumu, lze trénovat každý pixel pomocí Gaussovského rozložení $\eta(\boldsymbol{\mu}_{s,t}, \Sigma_{s,t})$, kde $\boldsymbol{\mu}_{s,t}$ jsou střední hodnoty pixelu s do času t a $\Sigma_{s,t}$ je kovarianční matice pixelu s do času t . Gaussovské rozložení $\eta(\boldsymbol{\mu}_{s,t}, \Sigma_{s,t})$ v tomto případě má tvar

$$\eta(\mathbf{I}_{s,t}, \boldsymbol{\mu}_{s,t}, \Sigma_{s,t}) = \frac{1}{(2\pi)^{\frac{K}{2}} |\Sigma|^{\frac{1}{2}}} \cdot e^{-\frac{1}{2}(\mathbf{I}_{s,t} - \boldsymbol{\mu}_{s,t})^\top \Sigma_{s,t}^{-1} (\mathbf{I}_{s,t} - \boldsymbol{\mu}_{s,t})}, \quad (5.7)$$

kde K je dimenze. Metriku lze určit pomocí Mahalanobisovi vzdálenosti

$$d(\mathbf{I}_{s,t}, \boldsymbol{\mu}_{s,t}) = \sqrt{(\mathbf{I}_{s,t} - \boldsymbol{\mu}_{s,t})^\top \Sigma_{s,t}^{-1} (\mathbf{I}_{s,t} - \boldsymbol{\mu}_{s,t})}, \quad (5.8)$$

kde $\mathbf{I}_{s,t}$ a $\boldsymbol{\mu}_{s,t}$ jsou vektory a $\Sigma_{s,t}$ je kovarianční matice. Aby byly zohledněny změny osvětlení, tak se střední hodnoty a kovarianční matice mohou iterativně přepočítávat podle následujících vztahů

$$\boldsymbol{\mu}_{s,t+1} = (1 - \alpha) \boldsymbol{\mu}_{s,t} + \alpha \mathbf{I}_{s,t}, \quad (5.9)$$

$$\Sigma_{s,t+1} = (1 - \alpha) \Sigma_{s,t} + \alpha (\mathbf{I}_{s,t} - \boldsymbol{\mu}_{s,t})^\top (\mathbf{I}_{s,t} - \boldsymbol{\mu}_{s,t}). \quad (5.10)$$

Kovarianční matice může být plná s velikostí $K \times K$ nebo může být pouze diagonální z důvodu ušetření výpočetního času. V případě, že se jedná o barevný třídimenzionální RGB prostor bude $K = 3$.

5.1.4 Gaussovská směs (GMM)

Pro zohlednění pozadí, které obsahuje animované textury, jako například vlny na vodě, nebo listy zmítané ve větru, byly použity Gaussovské směsi (Gaussian Mixture Model - GMM). Princip spočívá v modelování každého pixelu pozadí jako K Gaussovských směsí. Tedy pravděpodobnost výskytu barvy v daném pixelu je zastoupena následovně,

$$P(\mathbf{I}_{s,t}) = \sum_{i=1}^K \omega_{i,s,t} \cdot \eta(\mathbf{I}_{s,t}, \boldsymbol{\mu}_{i,s,t}, \Sigma_{i,s,t}), \quad (5.11)$$

kde $\eta(\mathbf{I}_{s,t}, \boldsymbol{\mu}_{i,s,t}, \Sigma_{i,s,t})$ je i -tý Gaussovský model a $\omega_{i,s,t}$ je jeho váha. Jednotlivé parametry se mohou iterativně měnit podle vztahů,

$$\omega_{i,s,t} = (1 - \alpha) \omega_{i,s,t-1} + \alpha, \quad (5.12)$$

$$\boldsymbol{\mu}_{i,s,t} = (1 - \rho) \boldsymbol{\mu}_{i,s,t-1} + \rho \mathbf{I}_{i,s,t}, \quad (5.13)$$

$$\Sigma_{i,s,t} = (1 - \rho) \Sigma_{i,s,t} + \rho (\mathbf{I}_{i,s,t} - \boldsymbol{\mu}_{i,s,t})^\top (\mathbf{I}_{i,s,t} - \boldsymbol{\mu}_{i,s,t}), \quad (5.14)$$

kde α je uživatelem volený parametr učení a ρ je uživatelem druhý volený parametr, který je definován jako

$$\rho = \alpha \cdot \eta(\mathbf{I}_{s,t}, \boldsymbol{\mu}_{i,s,t}, \Sigma_{i,s,t}). \quad (5.15)$$

Parametry μ a σ nevyužitých rozdělení zůstávají stejné, zatímco jejich váhy jsou redukovány podle vztahu $\omega_{i,s,t} = (1 - \alpha) \omega_{i,s,t-1}$, pro dosažení nulového ovlivnění výsledné pravděpodobnosti. Pokaždé, když neodpovídá žádná komponenta $\mathbf{I}_{s,t}$, tak je rozložení s nejmenší váhou nahrazenou Gaussovským rozložením s velkou počáteční variancí σ_0 a malou počáteční váhou ω_0 . Při každé aktualizaci Gaussovské směsi je K vah $\omega_{i,s,t}$ normalizováno tak, aby v součtu dávaly 1. Následně je K rozdělení seřazeno podle hodnoty $\omega_{i,s,t}$ nebo $\sigma_{i,s,t}$ a pouze H nejvíce se hodících, je použito jako část pozadí. Následně pixely, které jsou vzdálené více než 2.5 násobek směrodatné odchylky od některého z těchto H rozdělení, jsou teprve označeny za pohybující se objekty v popředí.

5.1.5 Kernel Density Estimace (KDE)

Neparametrický přístup může být také využit pro modelování Gaussovských směrů. V tomto ohledu byl navržen odhad pomocí Parzen-okénka. Pokud se jedná o barevné snímky z videa, může být jednodimenzionální jádro vytvořeno následovně,

$$P(\mathbf{I}_{s,t}) = \frac{1}{N} \sum_{i=t-N}^{t-1} \prod_{j=\{R,G,B\}} K\left(\frac{(I_{s,t}^j - I_{s,i}^j)}{\sigma_j}\right), \quad (5.16)$$

kde K je jádro (typicky Gaussovské rozložení) a N je počet předchozích snímků použitých pro odhad. Pixel je označen jako popředí, pokud je nepravděpodobné, že pixel pochází z tohoto rozdělení. Tedy pokud je $P(\mathbf{I}_{s,t})$ menší než předdefinovaná prahová hodnota. Kde σ_j může být neměnná nebo může být odhadnuta předem.

5.1.6 Modelování histogramem

Mezi další možnost jak oddělit pozadí od pohybujících se objektů v popředí, pomocí metody odečítání pozadí, je modelování pomocí histogramu. Hlavní myšlenka spočívá v rozdělení obrázku na menší obrazové části, ze kterých se vytvoří 1D histogramy v případě obrázku v odstínech šedi, nebo 3D histogramy v případě barevného obrázku. Tento postup se opakuje pro každý snímek videosekvence pro všechny obrazové části. Na základě porovnání dvou snímků lze určit, zda nastala změna v histogramech a vyhodnotit pohybující se objekt. Tedy pokud H_1 a H_2 označují histogramy ve stejné obrazové části ve dvou snímcích, je možno vzdálenost $d(H_1, H_2)$ určit pomocí Pearsonovi korelace r_{H_1, H_2}

$$d(H_1, H_2) = 1 - r_{H_1, H_2}, \quad (5.17)$$

kde

$$r_{H_1, H_2} = \frac{\sum_{i=1}^N (H_1^i - \bar{H}_1) (H_2^i - \bar{H}_2)}{\sqrt{\sum_{i=1}^N (H_1^i - \bar{H}_1)^2 \cdot \sum_{i=1}^N (H_2^i - \bar{H}_2)^2}}. \quad (5.18)$$

\bar{H}_1 a \bar{H}_2 jsou střední hodnoty

$$\bar{H}_j = \frac{1}{N} \sum_{i=1}^N H_j^i, \quad j \in \{1, 2\} \quad (5.19)$$

a N je počet intervalů (bins) umocněný na počet dimenzí histogramu (v případě, že je počet intervalů ve všech dimenzích stejný). Mezi další možné metriky patří Chi-Square vzdálenost

$$d(H_1, H_2) = \sum_{i=1}^N \frac{(H_1^i - H_2^i)^2}{H_1^i} \quad (5.20)$$

a Bhattacharyyaova vzdálenost

$$d(H_1, H_2) = \sqrt{1 - \frac{1}{\bar{H}_1 \bar{H}_2 N^2} \sum_{i=1}^N \sqrt{H_1^i H_2^i}}. \quad (5.21)$$

Na závěr se určí prahová hodnota τ , při které pixely patří do popředí a tvoří pohybující se objekty.

5.2 Barevné prostory

Metody odečítání pozadí založené na modelování pozadí s pomocí barevných informací dávají lepší výsledky oproti reprezentaci v odstínech šedi. Bude následovat stručný přehled některých existujících barevných modelů využívaných v počítačovém vidění. Barevný prostor je abstraktní matematický model popisující, jak mohou být barvy reprezentovány jako n -tice čísel, typicky jako tři nebo čtyři hodnoty komponent barev.

Základní barevné prostory - RGB, normované RGB a CIE-XYZ. Pro tyto základní barevné prostory je charakteristické, že jsou nativně využívány v počítačích a tudíž není potřeba žádné konverze, která vyžaduje určitý čas. S využitím normovaného RGB docílíme invariance na intenzitu osvětlení. Lze ušetřit jednu dimenzi při reprezentaci, pokud hodnoty RGB znormujeme v sumě na jedničku. CIE (Commission Internationale de l'Eclairage) barevný prostor popisuje barvy jako složku jasu Y a dvě další složky X a Z. CIE-XYZ barevný prostor, byl vytvořen z psychofyzických experimentů a barevně odpovídá charakteristice lidského vizuálního systému.

Vjemové barevné prostory - HSI, HSV, HSL a TSL. Tyto prostory se řadí mezi vjemové, což znamená, že se snaží popsat barvu tak, jak jí vnímá člověk očima. Modely rozdělují barvu na složky, jaké jsou člověkem jednoduše vnímány. HSV barevný prostor je například složen z barevného tónu, neboli odstínu (Hue) definujícího základní barvu. Dále je složen ze sytosti barvy (Saturation) a z hodnoty jasu (Value), která udává jak bílá/šedá/černá je barva.

Ortogonální barevné prostory - YCbCr, YIQ, YUV a YES. Ortogonální barevné prostory reprezentují originální RGB barvy formované nezávislými barevnými komponenty. V YCbCr reprezentuje Y jasovou komponentu, C_b a C_r reprezentují modrou a červenou chrominanční komponentu. Tyto barevné prostory jsou velmi využívány například v oblasti segmentace barvy kůže. Bylo vytvořeno mnoho modifikací této skupiny barevných prostorů.

Vjemově jednotné barevné prostory - CIE-Lab a CIE-Luv. Tyto barevné prostory jsou vytvořeny na základě toho, jak pozoruje svět člověk. Jsou vjemově jednotné, což znamená, že je to míra odlišnosti ve vnímání rozdílů barev, které jsou v barevném prostoru stejně vzdálené. To je docíleno pomocí nelineární transformací z CIE-XYZ barevného prostoru. Obecně lze komponenty rozdělit na svítivost L a chromatické komponenty a, b, u a v.

Kapitola 6

Sledování

Sledování v oblasti počítačového vidění je proces lokalizace pohybujícího se objektu (nebo více objektů) v čase za pomoci kamery (nebo více kamer). Tato metoda má celou řadu využití, například interakce člověk-počítač, zabezpečení a ostraha, rozšířená realita, řízení dopravy, lékařské zpracování a mnohé další. Video sledování může být velmi časově náročný proces, v závislosti k množství dat obsažených ve videu.

6.1 Optický tok

Optický tok je metoda v počítačovém vidění, která na základě relativního pohybu mezi pozorovatelem (lidské oko nebo kamera) a obrazovou scénou vypočítává pohyb. Tato metoda se velmi využívá v oblastech robotiky, detekce pohybu, objektové segmentace, výpočtu stereo disparity a mnohé další.

Sekvence seřazených snímků umožňuje odhad pohybu, buďto ve formě okamžité rychlosti nebo ve formě diskrétního obrazového posuvu. Metody optického toku vypočítávají pohyb mezi dvěma obrázky, které byly pořízeny v čase t a $t + \Delta t$. Tato metoda se nazývá také rozdílovou metodou, jelikož je založena na lokální aproximaci obrazového signálu pomocí Taylorova rozvoje, tedy pomocí parciálních derivací s ohledem na prostorové a časové souřadnice.

Pro 2D + t dimenzionální případ uvažujme voxel na pozici (x, y, t) o intenzitě $I(x, y, t)$, která bude změněna o hodnoty Δx , Δy a Δt v následujícím obrázku. Pro následující obrázek platí vazební rovnice

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t). \quad (6.1)$$

Za předpokladu, že bude pohyb mezi snímky malý, může být rovnice (6.1) přepsána pomocí Taylorova rozvoje jako

$$I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x, y, t) + \frac{\partial I}{\partial x} \Delta x + \frac{\partial I}{\partial y} \Delta y + \frac{\partial I}{\partial t} \Delta t + \text{členy vyšších řádů}. \quad (6.2)$$

Pro rovnici (6.2) platí

$$\frac{\partial I}{\partial x} \Delta x + \frac{\partial I}{\partial y} \Delta y + \frac{\partial I}{\partial t} \Delta t = 0, \quad (6.3)$$

neboli

$$\frac{\partial I}{\partial x} \frac{\Delta x}{\Delta t} + \frac{\partial I}{\partial y} \frac{\Delta y}{\Delta t} + \frac{\partial I}{\partial t} \frac{\Delta t}{\Delta t} = 0, \quad (6.4)$$

což dává

$$\frac{\partial I}{\partial x} V_x + \frac{\partial I}{\partial y} V_y + \frac{\partial I}{\partial t} = 0, \quad (6.5)$$

kde V_x a V_y jsou x -ová a y -ová složka rychlosti, nebo optický tok $I(x, y, t)$ a $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$ a $\frac{\partial I}{\partial t}$ jsou parciální derivace obrázku na pozici $I(x, y, t)$. Rovnici (6.5) lze přepsat na tvar

$$I_x V_x + I_y V_y = -I_t, \quad (6.6)$$

kde

$$I_x = \frac{\partial I}{\partial x}, \quad (6.7)$$

$$I_y = \frac{\partial I}{\partial y}, \quad (6.8)$$

$$I_t = \frac{\partial I}{\partial t}. \quad (6.9)$$

Rovnice (6.6) může být přepsána do výsledného tvaru

$$\nabla I^\top \mathbf{V} = -I_t, \quad (6.10)$$

který reprezentuje rovnici o dvou neznámých a nemůže být jednoznačně vyřešena v tomto tvaru, což je známo jako problém optického toku. Pro výpočet je tedy potřeba řešit soustavu o minimálně dvou rovnicích, proto je nutné zavést další omezující podmínky. Všechny metody řešící optický tok, zavádějí dodatečné omezující podmínky pro řešení skutečného optického toku. Více lze nalézt v [2].

6.2 Lucas-Kanade

V počítačovém vidění je metoda Lucas-Kanade ([9], [10]) široce používaná diferenční metoda pro odhad optického toku. Tuto metodu vytvořili pánové Bruce D. Lucas a Takeo Kanade. Tato metoda předpokládá, že optický tok je v okolí uvažovaného pixelu konstantní. Řeší základní rovnice optického toku pro pixely, které jsou v určitém okolí od uvažovaného středního pixelu, za pomoci metody nejmenších čtverců.

Pomocí informace z několika okolních pixelů lze metodou Lucas-Kanade často vyřešit nejednoznačnost rovnic optického toku. Mezi výhody této metody patří menší citlivost na obrazový šum oproti jiným metodám. Naopak mezi nevýhody se řadí, jelikož se jedná pouze o lokální metodu, nemožnost získání optického toku v interiérech, kde je mnoho jednolitých obrazových oblastí.

Předpokládá se, že posun obrazového obsahu (pixelů) mezi dvěma po sobě jdoucími blízkými okamžiky (obrázky) je velmi malý, v okolí bodu \mathbf{x} přibližně konstantní. Rovnice optického toku předpokládá, že pro všechny body x_i v malém okolí bodu \mathbf{x} platí

$$\begin{aligned} I_x(\mathbf{x}_1) V_x + I_y(\mathbf{x}_1) V_y &= -I_t(\mathbf{x}_1) \\ I_x(\mathbf{x}_2) V_x + I_y(\mathbf{x}_2) V_y &= -I_t(\mathbf{x}_2) \\ &\vdots \\ I_x(\mathbf{x}_N) V_x + I_y(\mathbf{x}_N) V_y &= -I_t(\mathbf{x}_N) \end{aligned} \quad (6.11)$$

kde $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$ jsou pixely v okolí středového bodu a $I_x(\mathbf{x}_i), I_y(\mathbf{x}_i), I_t(\mathbf{x}_i)$ jsou parciální derivace obrázku I za respektování souřadnic x, y a času t . Lokální optický tok je reprezentován vektorem $\mathbf{v} = [V_x, V_y]$. Soustava rovnic (6.11) může být zapsána v maticové formě jako $A\mathbf{v} = \mathbf{b}$, kde

$$A = \begin{bmatrix} I_x(\mathbf{x}_1) & I_y(\mathbf{x}_1) \\ I_x(\mathbf{x}_2) & I_y(\mathbf{x}_2) \\ \vdots & \vdots \\ I_x(\mathbf{x}_N) & I_y(\mathbf{x}_N) \end{bmatrix}, \quad (6.12)$$

$$\mathbf{v} = \begin{bmatrix} V_x \\ V_y \end{bmatrix} \quad (6.13)$$

a

$$\mathbf{b} = \begin{bmatrix} -I_t(\mathbf{x}_1) \\ -I_t(\mathbf{x}_2) \\ \vdots \\ -I_t(\mathbf{x}_N) \end{bmatrix}. \quad (6.14)$$

Tento systém má více rovnic než neznámých, proto se jedná o přeúrcenou soustavu rovnic. Lucas-Kanade metoda získává kompromisní řešení soustavy rovnic pomocí metody nejmenších čtverců. Konkrétně je třeba vyřešit soustavu rovnic o velikosti 2×2 ,

$$A^\top A \mathbf{v} = A^\top \mathbf{b}, \quad (6.15)$$

tedy po vyjádření neznámých

$$\mathbf{v} = (A^\top A)^{-1} A^\top \mathbf{b}, \quad (6.16)$$

nebo-li v rozepsaném zápisu

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^N I_x(\mathbf{x}_i)^2 & \sum_{i=1}^N I_x(\mathbf{x}_i) I_y(\mathbf{x}_i) \\ \sum_{i=1}^N I_y(\mathbf{x}_i) I_x(\mathbf{x}_i) & \sum_{i=1}^N I_y(\mathbf{x}_i)^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_{i=1}^N I_x(\mathbf{x}_i) I_t(\mathbf{x}_i) \\ -\sum_{i=1}^N I_y(\mathbf{x}_i) I_t(\mathbf{x}_i) \end{bmatrix}. \quad (6.17)$$

Výše popsaná metoda využívá standardní metody nejmenších čtverců, váha všech N pixelů \mathbf{x}_i v okolí středového bodu je stejná. Pro praktické použití je většinou lepší dát větší váhu bodům, které jsou blíže ke středovému bodu. Proto se využívá vážené verze rovnice (6.15), ve tvaru

$$A^\top W A \mathbf{v} = A^\top W \mathbf{b}, \quad (6.18)$$

po vyjádření neznámých

$$\mathbf{v} = (A^\top W A)^{-1} A^\top W \mathbf{b}, \quad (6.19)$$

kde W je diagonální matice o velikosti $N \times N$, která má na diagonále na pozicích $W_{ii} = w_i$ což jsou váhy odpovídající bodům x_i . Rozepsaně zapsáno

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^N w_i I_x(\mathbf{x}_i)^2 & \sum_{i=1}^N w_i I_x(\mathbf{x}_i) I_y(\mathbf{x}_i) \\ \sum_{i=1}^N w_i I_y(\mathbf{x}_i) I_x(\mathbf{x}_i) & \sum_{i=1}^N w_i I_y(\mathbf{x}_i)^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_{i=1}^N w_i I_x(\mathbf{x}_i) I_t(\mathbf{x}_i) \\ -\sum_{i=1}^N w_i I_y(\mathbf{x}_i) I_t(\mathbf{x}_i) \end{bmatrix}. \quad (6.20)$$

6.3 Kontinuální adaptivní mean-shift

Tento algoritmus pro sledování objektů je založen na starší metodě mean-shift, která bude nastíněna nejprve a následně bude popsána kontinuální adaptivní mean-shift metoda, neboli camshift (Continuously Adaptive Mean Shift).

6.3.1 Mean-shift

Metoda byla poprvé prezentována roku 1970 pány Fukunaga a Hostetler [11], jedná se o neparametrickou metodu. Využívá se v oblastech počítačového vidění a zpracování obrazu. Mean-shift je metoda odhadu hustotní funkce na základě diskrétních dat. Jedná se o iterativní postup, využívající počátečního odhadu x . Nechť je dána kernel funkce

$$K(x_i - x), \quad (6.21)$$

tato funkce určuje váhu blízkých bodů pro znovu odhadnutí střední hodnoty. Většinou je použit Gaussovský kernel pro odhad aktuální vzdálenosti, například

$$K(x_i - x) = e^{-c\|x_i - x\|^2}. \quad (6.22)$$

Vážená střední hodnota hustotní funkce v okolí bodu je určena pomocí K následovně

$$m(x) = \frac{\sum_{x_i \in N(x)} K(x_i - x) x_i}{\sum_{x_i \in N(x)} K(x_i - x)}, \quad (6.23)$$

kde $N(x)$ je okolí bodu x , tedy množina bodů pro které platí

$$K(x) \neq 0. \quad (6.24)$$

Mean-shift algoritmus nyní nastaví

$$x \leftarrow m(x), \quad (6.25)$$

což je opakováno dokud $m(x)$ konverguje.

Stručně lze algoritmus shrnout do následujících kroků:

1. Volba velikosti prohledávaného okna.
2. Volba počátečního umístění prohledávaného okna.
3. Výpočet umístění střední hodnoty v prohledávaném okně.
4. Vycentrování prohledávacího okna na pozici střední hodnoty z kroku (3).
5. Opakování kroků (3) a (4) do konce konvergence.

Tato metoda je využívána pro vizuální sledování. V nejjednodušší podobě je vytvořena pravděpodobnostní mapa aktuálního obrázku na základě barevného histogramu sledovaného objektu v předchozím snímku. Dále je využito mean-shift metody pro nalezení vrcholu z pravděpodobnostní mapy v blízkosti objektu v předchozím snímku. Více lze nalézt v [11].

6.3.2 Camshift

Tato metoda je založena na algoritmu mean-shift, který robustně nalézá vrcholy pravděpodobnostní mapy. Využití nalézá například, při sledování objektů ve video sekvencích a počítačových hrách. Camshift oproti mean-shift využívá dynamicky se v čase měnícího pravděpodobnostního rozdělení. Tímto přechází metoda mean-shift ze statisticky robustního prohledávacího algoritmu do módu dynamického sledování, které je využíváno pro sledování objektů ve video sekvencích. Camshift lze využít pro sledování na levných kamerách bez nutnosti jejich kalibrace.

Pro 2D obrázek a jeho pravděpodobnostní rozdělení se umístění centroidu střední hodnoty uvnitř prohledávaného okna nalezne následovně. Nalezne se nultý moment jako

$$M_{00} = \sum_x \sum_y I(x, y). \quad (6.26)$$

Naleznou se první momenty pro x a y

$$M_{10} = \sum_x \sum_y xI(x, y), \quad (6.27)$$

$$M_{01} = \sum_x \sum_y yI(x, y). \quad (6.28)$$

Následně je vypočteno umístění centroidu střední hodnoty prohledávacího okna jako

$$x_c = \frac{M_{10}}{M_{00}}, \quad (6.29)$$

$$y_c = \frac{M_{01}}{M_{00}}, \quad (6.30)$$

kde $I(x, y)$ je hodnota pravděpodobnosti pixelu na pozici (x, y) .

2D orientace pravděpodobnostního rozložení je také lehce zjistitelná s využitím druhých momentů, která jsou definovány následovně

$$M_{20} = \sum_x \sum_y x^2 I(x, y), \quad (6.31)$$

$$M_{02} = \sum_x \sum_y y^2 I(x, y). \quad (6.32)$$

Následně orientace objektu, konkrétně jeho hlavních os je dána

$$\Theta = \frac{\arctan\left(\frac{2\left(\frac{M_{11}}{M_{00}} - x_c y_c\right)}{\left(\frac{M_{20}}{M_{00}} - x_c^2\right) - \left(\frac{M_{02}}{M_{00}} - y_c^2\right)}\right)}{2}. \quad (6.33)$$

Délka a šířka (blobu) pravděpodobnostního rozložení může být vypočítána následovně, necht'

$$a = \frac{M_{20}}{M_{00}} - x_c^2, \quad (6.34)$$

$$b = 2\left(\frac{M_{11}}{M_{00}} - x_c y_c\right), \quad (6.35)$$

$$c = \frac{M_{02}}{M_{00}} - y_c^2. \quad (6.36)$$

Následně mohou být délka l a šířka w vypočteny jako

$$l = \sqrt{\frac{(a+c) + \sqrt{b^2 + (a-c)^2}}{2}}, \quad (6.37)$$

$$w = \sqrt{\frac{(a+c) - \sqrt{b^2 + (a-c)^2}}{2}}. \quad (6.38)$$

Pomocí výpočtů v rovnicích (6.33), (6.37) a (6.38) lze určit velikost sledovaného objektu a jeho natočení, přesněji velikost a natočení jeho dynamicky se měnícího pravděpodobnostního modelu v aktuálním snímku. Více informací lze nalézt v [20].

6.4 Kalmanův filtr

Pro určité třídy modelů a pozorování může klasický přístup přinést optimální řešení, pro jiné ne, proto je nutné využít sofistikovanější metodu, například Kalmanův filtr. Předpokládá se, že systém a pozorování jsou lineární. Další předpoklad je kladen na stavový šum a šum měření, oba musí být bílé a Gaussovské (několik modifikací a rozšíření může být nalezeno ve více odborných textech). Ve skutečnosti dokáže Kalmanův filtr poskytnout nejlepší odhad chování systému ve smyslu nejmenších čtverců.

Kalmanův filtr lze popsat modelem

$$\mathbf{x}_{k+1} = A_k \mathbf{x}_k + \mathbf{w}_k, \quad (6.39)$$

$$\mathbf{z}_k = H_k \mathbf{x}_k + \mathbf{v}_k, \quad (6.40)$$

kde matice A_k popisuje vývoj stavového modelu, v rovnici (6.39) není obsaženo $B_k \mathbf{u}_k$ jelikož vektor vstupů $\mathbf{u}_k = \mathbf{0}$, dále \mathbf{w}_k reprezentuje Gaussovský stavový šum s nulovou střední hodnotou a kovarianční maticí Σ_k , tedy

$$\mathbf{w}_k \sim \mathcal{N}(0, \Sigma_k). \quad (6.41)$$

Matice Σ_k je vypočtena následovně

$$\Sigma_k = E[\mathbf{w}_k^\top \mathbf{w}_k], \quad (6.42)$$

nebo alternativně zapsáno po složkách

$$(\Sigma_k)_{ij} = E(w_k^i w_k^j), \quad (6.43)$$

kde w_k^i označuje i -tou složku vektoru \mathbf{w}_k . Matice H_k z rovnice (6.40) je matice měření, která popisuje v jaké relaci jsou měření a model systému. Vektor \mathbf{v}_k je Gaussovský šum měření

$$\mathbf{v}_k \sim \mathcal{N}(0, R_k), \quad (6.44)$$

který má nulovou střední hodnotu a kovarianční maticí R_k .

Stav \mathbf{x} v diskrétním čase $k-1$, tedy \mathbf{x}_{k-1} (nebo jeho odhad $\hat{\mathbf{x}}_{k-1}$) může být spočítán pomocí rovnice (6.39) jako apriorní odhad

$$\mathbf{x}_k = A_{k-1} \mathbf{x}_{k-1}. \quad (6.45)$$

Běžně je tento odhad označován jako $\hat{\mathbf{x}}_k^-$, kde $\hat{}$ označuje odhad a znak $-$ označuje „před“ pozorováním. Obdobně je definováno $\hat{\mathbf{x}}_k^+$ kde $\hat{}$ označuje odhad a znak $+$ označuje „po“ pozorování, jinak řečeno odhad po výpočtu Kalmanova filtru v aktuálním čase. $\hat{\mathbf{x}}_k^+$ představuje vylepšení $\hat{\mathbf{x}}_k^-$.

Přidruženě ke zmíněným odhadům mohou být definovány chyby následovně

$$\mathbf{e}_k^- = \mathbf{x}_k - \hat{\mathbf{x}}_k^-, \quad (6.46)$$

$$\mathbf{e}_k^+ = \mathbf{x}_k - \hat{\mathbf{x}}_k^+, \quad (6.47)$$

s korespondujícími kovariančními maticemi P_k^- a P_k^+ . Tyto chyby jsou způsobeny šumem \mathbf{w}_k a chybou odhadu.

Kalmanův filtr posuzuje rezidua vyjádřená z rovnice (6.40), konkrétně

$$\mathbf{z}_k - H_k \hat{\mathbf{x}}_k^- = \mathbf{v}_k, \quad (6.48)$$

tedy pokud bude šum měření $\mathbf{v}_k = 0$ a odhad $\hat{\mathbf{x}}_k^-$ přesný pak

$$\mathbf{z}_k - H_k \hat{\mathbf{x}}_k^- = 0, \quad (6.49)$$

o toto usiluje matice K_k . Takzvaný Kalmanův zisk K_k , který aktualizuje hodnotu $\hat{\mathbf{x}}_k^-$ na $\hat{\mathbf{x}}_k^+$ ve smyslu metody nejmenších čtverců následovně

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K_k (\mathbf{z}_k - H_k \hat{\mathbf{x}}_k^-). \quad (6.50)$$

Pokud je K_k známý, pak je hodnota $\hat{\mathbf{x}}_k^-$ aktualizována na $\hat{\mathbf{x}}_k^+$ a (všechny ostatní proměnné jsou známé) rovnice (6.50) je vyřešena.

Rovnice (6.46), (6.50) dohromady dávají

$$\mathbf{e}_k^+ = \mathbf{x}_k - \hat{\mathbf{x}}_k^+ = \quad (6.51)$$

$$= \mathbf{x}_k - ((I - K_k H_k) \hat{\mathbf{x}}_k^- - K_k \mathbf{z}_k) \quad (6.52)$$

pomocí (6.40) je získáno

$$\mathbf{e}_k^+ = \mathbf{x}_k - (I - K_k H_k) \hat{\mathbf{x}}_k^- - K_k (H_k \mathbf{x}_k + \mathbf{v}_k), \quad (6.53)$$

dále po úpravě

$$\mathbf{e}_k^+ = (I - K_k H_k) \mathbf{e}_k^- + K_k \mathbf{v}_k. \quad (6.54)$$

Nechť je definováno

$$P_k^- = E \left[\mathbf{e}_k^{-\top} \mathbf{e}_k^- \right], \quad (6.55)$$

$$P_k^+ = E \left[\mathbf{e}_k^{+\top} \mathbf{e}_k^+ \right], \quad (6.56)$$

$$R_k = E \left[\mathbf{v}_k^\top \mathbf{v}_k \right] \quad (6.57)$$

a nezávislost chyby, která je dána

$$E \left[\mathbf{e}_k^{-\top} \mathbf{v}_k \right] = E \left[\mathbf{v}_k^{-\top} \mathbf{e}_k \right] = 0 \quad (6.58)$$

a s využitím rovnice (6.54) je získáno

$$P_k^+ = (I - K_k H_k) P_k^- (I - K_k H_k)^\top + K_k R_k K_k^\top, \quad (6.59)$$

tato rovnice se nazývá Riccatiova rovnice.

Kalmanův zisk K_k je nyní volen tak, aby minimalizoval součet prvků na diagonále matice P_k^+ , neboli $trace(P_k^+)$, která reprezentuje sumu aposterioriálních chybových variancí, toto je aspekt „metody nejmenších čtverců“ tohoto algoritmu. Toho je

docíleno tak, že se parciálně zderivujeme P_k^+ s respektováním K_k a položením rovnice rovno nule. Je známo, že pokud matice B je symetrická pak platí

$$\frac{\partial}{\partial A} \text{trace} (ABA^\top) = 2AB, \quad (6.60)$$

pokud je tato vlastnost využita v rovnici (6.59) lze získat

$$-2(I - K_k H_k) P_k^- H_k^\top + 2K_k R_k = 0, \quad (6.61)$$

tato rovnice je řešena pro K_k následovně

$$K_k = P_k^- H_k^\top (H_k P_k^- H_k^\top + R_k)^{-1}, \quad (6.62)$$

kde

$$P_k^- = A_k P_{k-1}^- A_k^\top + \Sigma_{k-1}. \quad (6.63)$$

Na závěr je úpravami získán vztah

$$P_k^+ = (I - K_k H_k) P_k^-. \quad (6.64)$$

Pokud je funkce popisující stav a/nebo měření je nelineární lze využít metodu zvanou rozšířený Kalmanův filtr, který využívá linearizace, za předpokladu, že pravděpodobnostní rozložení jsou stále Gaussovské [21]. Obecné informace lze nalézt v [2].

6.5 Particle filtr

Kalmanův filtr je založen na teorii řízení a ukázal se velmi nápomocný při řešení úlohy sledování, nicméně předpoklady pro jeho použití jsou velmi omezující a jsou často limitující při řešení reálných úloh. Často je vyžadováno sledování i přes značné zašumění, což bývá požadavek mnohých systémů. To může být velmi problematické a v případě využití unimodálních Gaussovských rozložení je často neodpovídající. V mnoha aplikačních případech není možné nalézt odpovídající matematický model. Tento problém vedl k použití obecnějšího přístupu využívajícího **particle filtr**, kde jsou systémy reprezentovány souborem pravděpodobnostně získaných vzorků, které popisují empirický popis toho, co je a co není pravděpodobné. Zajímavostí je, že tento obecnější přístup byl proveden tak, aby uspokojivě pracoval v reálném čase a dosahoval lepších výsledků než při využití Kalmanova filtru.

Particle filtr je založen na metodě výběru vzorků, které aproximují rozdělení dočas-nou strukturou. V počítačovém vidění byla tato metoda popularizována pány Isard a Blake v jejich **CONDitional DENSity propagATION = CONDENSATION** metodě (metoda propagace podmíněné hustoty pravděpodobnosti) [12]. Předpokladem je systém ve stavech $X_t = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_t\}$, kde index t představuje čas. Mějme v čase t hustotu pravděpodobnosti, která nám říká jak je stav \mathbf{x}_t pravděpodobný, tento stav je reprezentován souborem částic, neboli souborem vzorků stavu, jejichž výskyt se řídí hustotou pravděpodobnosti. Mějme sekvenci pozorování

$Z_t = \{z_1, z_2, \dots, z_t\}$ pravděpodobnostně odpovídající stavu $\{x_i\}$ a Markovovský předpoklad, že x_t je závislá pravděpodobnostně na předchozím stavu x_{t-1} což může být modelováno jako $P(x_t|x_{t-1})$. Významný rozdíl spočívá v neexistenci žádného omezení (na linearitu nebo Gaussovské chování), na některý z těchto vztahů.

CONDENSATION je iterativní metoda, která udržuje v každém kroku sadu vzorků s_i s přidruženými váhami π_i

$$\begin{aligned} S_t &= \{(s_i, \pi_i)\}, \quad i = 1, 2, \dots, N, \quad \sum_i \pi_i = 1, \\ &= \{(s_{ti}, \pi_{ti})\}. \end{aligned} \quad (6.65)$$

Dohromady tyto vzorky a váhy reprezentují hustotu pravděpodobnosti x_t danou Z_t . Toto rozdělení nebude mít daný předpis. Úkolem je odvodit S_t z S_{t-1} .

Za účelem vytvoření N nových vzorků a s nimi souvisejících vah v čase t , je vybráno N vzorků z S_{t-1} s ohledem na jejich váhy. Vzorek je podroben Markovově předpokladu a následně se znovu převáží váhy s ohledem na nové pozorování z_t . Formálně lze algoritmus CONDENSATION zapsat následovně:

1. Je předpokládána znalost vážené sady vzorků v čase $t - 1$

$$S_{t-1} = \{(s_{(t-1)i}, \pi_{(t-1)'i})\}, \quad i = 1, 2, \dots, N. \quad (6.66)$$

Dále je nastaveno

$$c_0 = 0 \quad (6.67)$$

$$c_i = c_{i-1} + \pi_{(t-1)i}, \quad i = 1, 2, \dots, N \quad (6.68)$$

(kumulativní pravděpodobnost).

2. Pro učení n -tého vzorku z S_t , je vybráno náhodné číslo z množiny $[0, 1]$ a určeno $j = \arg \min_i (c_i > r)$, podle kterého budou propagovány vzorky dále. Toto je nazýváno vzorkováním na základě důležitosti, neboli větší váhu mají pravděpodobnější vzorky.
3. **Predikce:** Je použita znalost Markovovo chování x_t pro odvození s_{tn} . Jak přesně se tak stane závisí na Markovově vztahu, v Kalmanově případě bude tento vztah nabývat tvaru

$$s_{tn} = A_{t-1}s_{(t-1)j} + w_{t-1}, \quad (6.69)$$

kde matice A_{t-1} představuje matici systému a w_{t-1} šum, tvar tohoto vztahu nemá danou strukturu. Důležité je vzít na vědomí, že $s_{(t-1)j}$ může být vybrán vícekrát v iteracích kroku (2), ale tato propagace bude generovat jiné s_{tn} v důsledku šumu.

4. **Korekce:** Je využito aktuální pozorování z_t a znalost pravděpodobnosti pozorování pro nastavení

$$\pi_{tn} = p(z_t|x_t = s_{tn}). \quad (6.70)$$

5. Je iterováno N krát od kroku (2).

6. Normalizace vah $\{\pi_{ti}\}$, tak že

$$\sum_i \pi_{ti} = 1. \quad (6.71)$$

7. Nejlepší odhad \mathbf{x}_t je nyní

$$\mathbf{x}_t = \sum_{i=1}^N \pi_{ti} \mathbf{s}_{ti}, \quad (6.72)$$

nebo obecněji

$$E[f(\mathbf{x}_t)] = \sum_{i=1}^N \pi_{ti} f(\mathbf{s}_{ti}). \quad (6.73)$$

Tento algoritmus může modelovat libovolnou složitou hustotu pravděpodobnosti, což dovoluje udržovat libovolný počet dvojic hypotéz současně. Eventuálně je možné snížit počet současně rozvíjených hypotéz až na jedinou dvojici [2].

Kapitola 7

System stereo kalibrace

Tato kapitole je již zaměřena na praktickou část vypracování řešené úlohy. Využil jsem teorii kalibrace a stereo vidění, popsanou v kapitole 4. Nejprve v úvodu popíši použitý hardware a software, který jsem využil pro realizaci všech praktických částí. Dále bude podrobněji rozebrána programová realizace systému stereo kalibrace a závěrem budou zmíněny možnosti samotné stereo kalibrace vytvořeného programu.

Cílem této dílčí praktické části bylo vytvořit program, provádějící stereo kalibraci. Uložit výstupní data v požadovaném formátu pro další využití a vizuálně reprezentovat úspěšnost stereo kalibrace a rektifikace.

7.1 Použitý hardware a software

V této úvodní podkapitole je popsán použitý hardware a jeho základní specifikace, následuje popis softwaru využitého pro řešení, vizualizaci a vypracování jednotlivých úloh.

Všechny výpočty jsem realizoval na běžně dostupném notebooku, který obsahoval dvoujádrový procesor a 8 gigabajtů operační paměti. Dostatek operační paměti byl pro dílčí výpočty a uložení potřebných dat klíčový, jak je popsáno v následujících kapitolách, jež daný problém popisují a řeší.

Jako vstup video signálu jsem zvolil dvě shodné webové kamery od firmy Trust. Jedná se o obyčejná zařízení, běžně k dostání. Nebyly využity žádné speciální studiové nebo jiné typy kamer, které se vyznačují lepším zpracováním optického senzoru a samotné optiky ve formě použitých čoček. Vzhled kamer je ilustrován na Obr. 7.1. Tyto webové kamery jsem využil, jelikož jsou schopny natáčet video až do vysokého rozlišení 1920×1080 pixelů. Webové kamery obsahují mikrofon pro nahrávání zvuku a integrované diody pro lepší obraz při špatném okolním osvětlení. Zmíněné dvě funkce jsem při experimentech nevyužíval, jelikož zpracování zvuku nebylo předmětem řešené úlohy a experimenty probíhaly za odpovídajících světelných podmínek.



Obrázek 7.1: Typ použitých běžně dostupných webových kamer značky Trust.

Řešení jsem realizoval na 64 bitovém operačním systému Windows 7 Professional s pomocí vývojového prostředí Microsoft Visual Studio 2008. Veškeré programy jsem realizoval s využitím programovacího jazyka C++ pomocí objektově orientovaného programování (OOP).

Dále jsem využil volně dostupnou knihovnu OpenCV (Open Source Computer Vision Library) verze 2.4.2, což je knihovna programovacích metod zaměřena především na počítačové vidění a zpracování obrazu v reálném čase. OpenCV je zdarma k použití v rámci open source licencí BSD. Knihovna podporuje celou řadu programovacích jazyků (Java, C++, atd.) a typů operačních systémů (Windows, Linux, atd.). Pro vypracování programové části využívající OpenCV jsem čerpal informace ze zdroje [22].

K vizualizaci dat ve formě grafů, k ověřování výpočtů a prvotním verzím programů jsem využil 64 bitový MATLAB (MATrix LABoratory) verze R2010a, což je interaktivní programové prostředí a skriptovací jazyk čtvrté generace. MATLAB umožňuje počítání s maticemi, vykreslování 2D i 3D grafů funkcí, implementaci algoritmů, počítačovou simulaci, analýzu a prezentaci dat i vytváření aplikací včetně uživatelského rozhraní.

V poslední řadě jsem využil mnoho dalších programů pro práci s videem a obrázky, konkrétně stojí za zmínění pouze Gimp, Sony Vegas, VirtualDub a mnohé další.

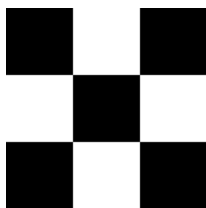
Výše zmíněný hardware a software byl použit ve všech následujících kapitolách zabývajících se praktickým řešením dané úlohy.

7.2 Kalibrační vzor

Pro stereo kalibraci bylo nutno zvolit vhodný kalibrační obrazec. Metody z knihovny OpenCV, které byly využity, mohou vypočítat kalibraci kamer za pomoci kalibrační černobílé šachovnice nebo asymetrické kruhové mřížky.

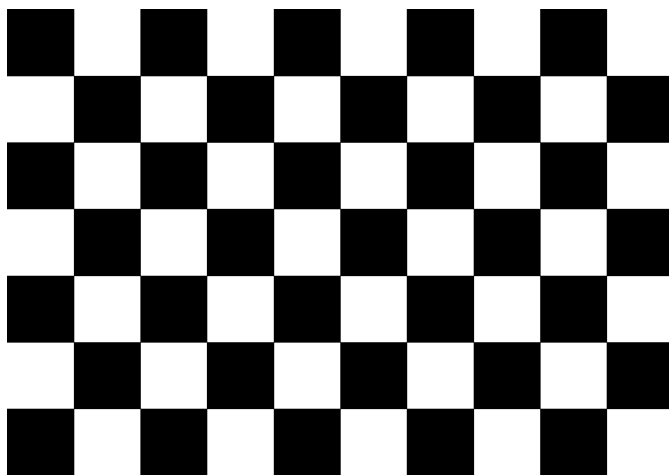
Pro kalibraci jsem zvolil černobílý šachovnicový kalibrační obrazec se čtvercovými poli, z důvodu jednoduchého měření vzdáleností mezi vnitřními rohy, která je důležitá pro výpočet reálné metriky.

Nejmenší možný počet detekovaných vnitřních rohů, pomocí použitých OpenCV metod, je 4. Tento extrémní případ je zobrazen na Obr. 7.2. Kalibrační vzor je velikosti 3×3 polí a obsahuje právě 4 vnitřní rohy pro kalibraci.



Obrázek 7.2: Nejmenší možný kalibrační vzor obsahující pouze 4 vnitřní rohy.

Na Obr. 7.3 je zobrazen kalibrační vzor, který jsem využil. Jeho velikost je 6×9 (řádků \times sloupců) vnitřních rohů. Takto velký (v porovnání s nejmenším možným kalibračním vzorem) kalibrační vzor byl využit z důvodu lepší kalibrace. V případě většího počtu kalibračních rohů (bodů) se snižuje ovlivnění obrazovým šumem a výpočet samotné kalibrace se stává přesnějším.



Obrázek 7.3: Použitý kalibrační vzor o velikosti 6×9 vnitřních rohů.

Pro experimentální účely byl vytvořen kalibrační vzor o velikosti popsané výše a velikosti roztečí vnitřních rohů 25 mm.

7.3 Popis implementace systému stereo kalibrace

V této podkapitole bude popsána obecná funkčnost vytvořeného programu pro stereo kalibraci, jednotlivé metody a souslednost jejich použití.

Vstupem do programu je vždy dvojice časově jednotných obrazových sekvencí. První možností je využít video v reálném čase z kamer, druhou možností je jako vstup použít dvojici předem nahraných videí. Tato dvojice videí nesmí pocházet ze stejného zdroje (kamery), musí se jednat o dvě navzájem prostorově posunuté kamery pro nahrávání vstupních videí.

Prostorové umístění kamer se během kalibrace a následného použití programů, využívajících výstupy ze stereo kalibračního programu, nesmí měnit. V případě prostorového přemístění zdroje video signálu je během stereo kalibrace, nebo při použití programů závislých na ní, nutné kalibraci opakovat. Jinak může dojít k nesprávným výpočtům a špatné funkčnosti programů.

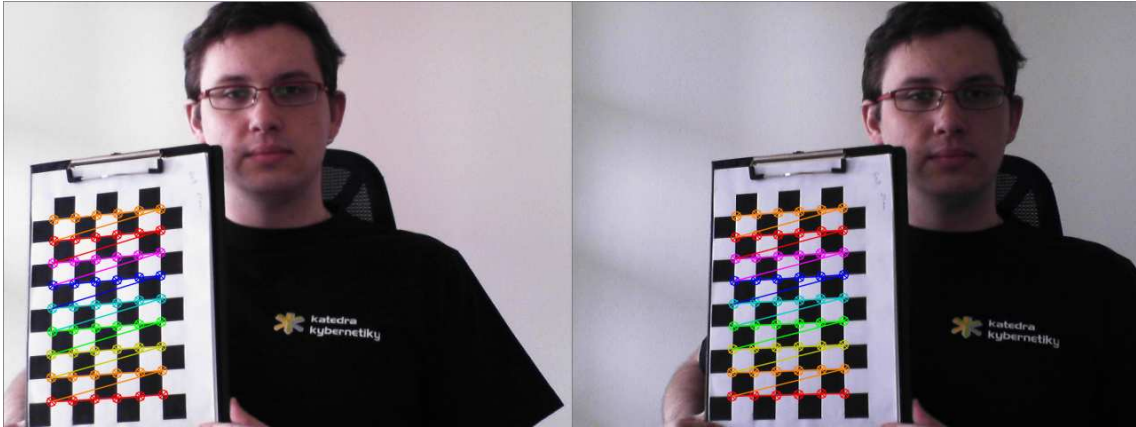
Po spuštění programu je nutné zadat několik vstupních parametrů, bez kterých nelze pokračovat ve stereo kalibraci.

- Počet řádků vnitřních rohů aktuálně používaného kalibračního vzoru.
- Počet sloupců vnitřních rohů aktuálně používaného kalibračního vzoru.
- Velikost rozteče vnitřních rohů kalibračního vzoru v milimetrech.
- Počet párů snímků, ze kterých bude vypočtena stereo kalibrace.

Při využití kalibračního vzoru z kapitoly 7.2 a po vytisknutí ve vhodné velikosti na papír formátu A4, lze docílit rozteče mezi vnitřními rohy o velikosti 25 mm. To je dostačující pro kalibraci ve vzdálenosti 1-2 metry při rozlišení 640×480 pixelů. Pro kalibraci pracovního prostoru ve větší vzdálenosti je nutné využít větší rozteče mezi vnitřními body u kalibračního obrazce, nebo využít vyšší rozlišení vstupního video signálu.

Po zadání výše uvedených parametrů se zobrazí aktuální náhledy dvojice kamer (nebo se začne přehrávat dvojice vstupních video sekvencí). V těchto náhledech je již ve zjednodušené podobě vizualizováno nalezení vnitřních rohů kalibračního vzoru. Z toho plyne možnost vyzkoušet si natočení kalibračního vzoru a rozvrhnutí poloh jednotlivých kalibračních snímků, což lze vidět na Obr. 7.4. V této fázi lze ještě nastavit prostorovou pozici, orientaci a natočení kamer. Dále je již vyžadováno, aby nebylo s kamerami pohybováno.

Následně je uživatel vyzván, aby spustil kalibraci v určitém módu, tyto možnosti budou popsány dále v 7.4. Veškeré dále uváděné metody jsou implementovány v použité knihovně OpenCV. Po spuštění konkrétního módu kalibrace se použije tolik párů vstupních snímků, kolik bylo při spuštění zadáno. Každý vstupní kalibrační snímek musí obsahovat kalibrační vzor v poloze, ve které lze nalézt všechny vnitřní rohy. Každý kalibrační snímek se pomocí metody `cvtColor` převede do odstínů šedi. Dále



Obrázek 7.4: Náhled dvojice kamer s vizualizací nalezených vnitřních rohů pro následnou stereo kalibraci.

se pomocí metody `findChessboardCorners` nalezne maximální možný počet vnitřních rohů. Pokud není kalibrační vzor ve scéně celý, nebo je ve špatné poloze, nelze detekovat veškeré vnitřní rohy a aktuální pár kalibračních snímků nebude zahrnut do výpočtu výsledné stereo kalibrace. V opačném případě, pokud jsou nalezeny všechny vnitřní rohy (počet sloupců krát počet řádků vnitřních rohů), je výpočet zpřesněn pomocí metody `cornerSubPix`, která určí pozici vnitřního rohu s přesností na subpixel. Takto nalezené rohy jsou uloženy ve vektoru vektorů 2D bodů pro následnou kalibraci. Poté co jsou úspěšně nalezeny, v zadaném počtu párů kalibračních snímků, veškeré vnitřní kalibrační rohy jsou dva (pro každou kameru zvlášť) plně obsazené vektory vektorů 2D bodů předány následujícím metodám pro kalibraci.

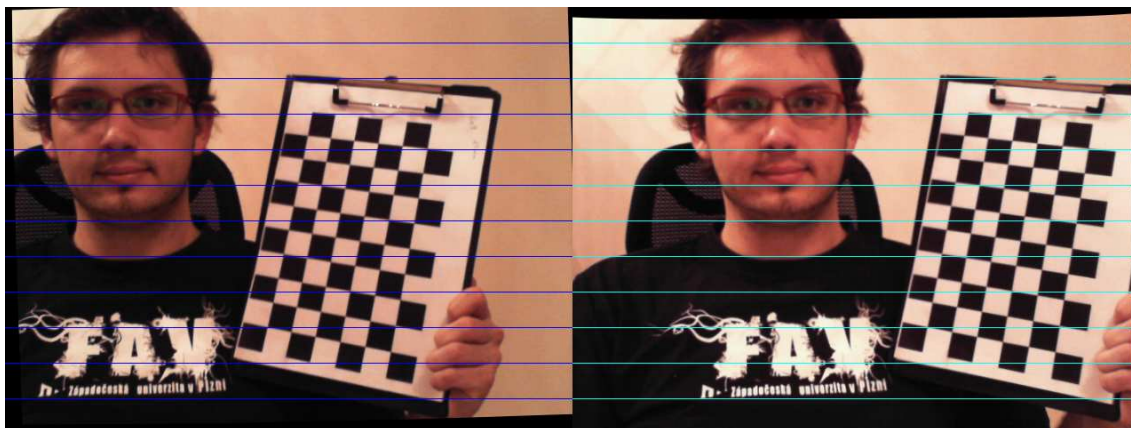
Dalším vstupem do metody stereo kalibrace jsou synteticky vytvořené 3D body, jejichž počet je totožný jako počet na reálném kalibračním vzoru (stejný počet řádků a sloupců). Jediný rozdíl je ten, že jejich 3. souřadnice (z -ová osa, hloubková dimenze) je rovna 0. Důležité je, aby rozteče mezi jednotlivými synteticky vygenerovanými 3D body odpovídaly reálné vzdálenosti, která byla v milimetrech zadána jako jeden ze vstupních parametrů. Takto se vytvoří vektor o velikosti odpovídající počtu párů kalibračních snímků. Na každé pozici v tomto vektoru se nachází vektor obsahující jednotlivé 3D synteticky vygenerované body. Tyto synteticky vygenerované 3D body odpovídají virtuálnímu kalibračnímu vzoru s počátkem na souřadnicích $x = 0$, $y = 0$ zvětšujících se v kladném směru os (jelikož se jedná o souřadný systém obrázku, tak bude počátek v rohu, který je nejvíce vlevo nahoře, dále kladné směry os budou doprava a dolů).

Pro samotnou stereo kalibraci jsem využil metodu `stereoCalibrate`. Jako vstup tato metoda vyžaduje vektor vektorů synteticky generovaných 3D bodů, dvojici vektor vektorů detekovaných 2D rohů v jednotlivých kalibračních snímcích. Dále jsou jako vstupně-výstupní parametry vyžadovány jednotlivé matice M_1 , M_2 vnitřních parametrů kamer, vektory \mathbf{d}_1 , \mathbf{d}_2 , které reprezentují deformační parametry kamer. Matice vnitřních parametrů kamer a vektory deformačních parametrů mohou být nulové a jsou následně vypočteny, nebo mohou být inicializovány určitými předem

známými hodnotami a jsou pouze vylepšeny. V mém případě jsem využil variantu, kdy jsou matice vnitřních parametrů kamer a vektory deformačních parametrů nulové a jsou vypočteny. Posledním pouze vstupním parametrem je velikost obrázku v pixelech. Výstupními vypočtenými parametry jsou matice R a vektor \mathbf{t} , které reprezentují rotaci a translaci mezi souřadnými systémy první a druhé kamery. Dále jsou výstupy esenciální matice E a fundamentální matice F . Metoda provádí výpočty iterativně, proto vyžaduje ukončovací podmínky `cvTermCriteria`, ty jsem zvolil jako maximální počet iterací $i \leq 100$, nebo pokud chyba daného algoritmu je $\epsilon \leq 10^{-5}$. Závěrem je nutné definovat příznakové bity. Použil jsem následující: `CV_CALIB_ZERO_TANGENT_DIST` nastavující tangenciální deformační koeficienty obou kamer na nulu, dále `CV_CALIB_SAME_FOCAL_LENGTH` nastaví stejné ohniskové vzdálenosti a `CV_CALIB_FIX_ASPECT_RATIO` stanoví poměr mezi ohniskovými vzdálenostmi. Hlavní důvod proč jsem využil výše zmíněnou metodu (`stereoCalibrate`) je, že její výstupy jsou využity jako vstup do bezprostředně následující metody, která se zabývá stereo rektifikací.

K výpočtu rektifikace, jejíž teorie byla popsána v podkapitole 4.3, jsem využil metodu `stereoRectify`. Jejímž vstupem jsou matice M_1 , M_2 vnitřních parametrů kamer, vektory \mathbf{d}_1 , \mathbf{d}_2 reprezentující deformační parametry, matice R a vektor \mathbf{t} , které reprezentují rotaci a translaci mezi souřadnými systémy první a druhé kamery. Jako poslední parametr, který je nutné zadat je velikost obrázku v pixelech. Výstupem této metody jsou R_1 , R_2 představující jednotlivé rotační matice kamer. Na závěr z této metody vystupuje matice Q , tato reprezentuje transformaci disparitou a hloubkovou mapou, což bude podrobněji popsáno v jedné z následujících kapitol.

Bezprostředně po výpočtu `stereoRectify` jsou matice a vektory M_1 , M_2 , \mathbf{d}_1 , \mathbf{d}_2 , R_1 , R_2 , P_1 , P_2 a Q binárně uloženy do souborů s odpovídajícím názvem. Tyto parametry slouží jako vstupy do dalších systémů a budou popsány dále, slouží pro výpočet hloubkové souřadnice bodů scény.



Obrázek 7.5: Rektifikovaná dvojice obrázků.

Po výpočtu všech výše uvedených metod lze stereo kalibraci a konkrétně stereo rektifikaci vizuálně ověřit. K tomuto účelu je volána metoda `initUndistortRectifyMap`, jejíž vstupními parametry jsou M_k , D_k , R_k , P_k a velikost obrázku v pixelech, kde

$k \in \{1, 2\}$. Tato metoda je volána pro každou kameru k samostatně. Výstupem jsou transformační mapy pro jednotlivé osy x a y . Tyto mapy transformují obraz vzhledem k deformačním parametrům a odhadnutým maticím. Praktickou ukázkou, jak může rektifikace dopadnout lze vidět na Obr. 7.5, tento obrázek obsahuje dvojici rektifikovaných kamer s vykreslenými úsečkami pro možnou lepší kontrolu navzájem odpovídajících bodů. Rektifikace byla úspěšná (dokonalá), pokud bodu v jedné kameře na libovolném řádku odpovídá bod na stejném řádku v druhé kameře (neplatí pro body, které se nevyskytují v obou snímcích z kamer).

7.4 Možnosti kalibrace

Jak bylo naznačeno v podkapitole 7.3 lze kalibraci provést více způsoby. V následujících odstavcích budou popsány dvě možnosti provedení stereo kalibrace a to automatická a manuální. Tyto možnosti jsou přístupné až po správném zadání vstupních parametrů popsaných dříve. Systém kalibrace reaguje na vstupy z klávesnice až po kliknutí na aktuální zobrazení jedné z kamer.

V prvním případě lze stereo kalibraci provést v automatickém režimu. Tento režim se aktivuje stisknutím klávesy s (jako Start). Od stisku klávesy má uživatel čas, o délce trvání 30 snímků na to, aby uvedl kalibrační vzor ve scéně do požadované polohy. Poté se použije aktuální snímek pro nalezení kalibračního vzoru a jeho vnitřních rohů, při úspěšném nalezení se detekované body uloží. V opačném případě bude uživatel upozorněn, formou výpisu do konzole, o nenalezení správného počtu vnitřních bodů a aktuální snímek nebude použit pro kalibraci ani připočítán do počtu párů kalibračních snímků. V případě, že nejsou kalibrační rohy nalezeny v první, druhé nebo obou kamerách, není ani jeden z dvojice snímků použit pro výpočet kalibrace. Program snímá snímky do té doby, než uloží požadovaný počet párů kalibračních snímků. Následně provede kalibraci, uloží vypočtená data a zobrazí uživateli vypočtenou stereo rektifikaci.

V druhém případě jak lze stereo kalibrační program použít, je využít manuálního režimu. Tento režim se aktivuje stisknutím klávesy n (jako New). Na rozdíl od automatického způsobu stereo kalibrace je v tomto případě vyžadována větší interakce ze strany uživatele. Konkrétně pokaždé, když uživatel uzná, že kalibrační vzor je v požadované poloze ve scéně, tak je vyžadováno stisknutí klávesy n . Dále je postup programu stejný jako u automatické stereo kalibrace.

Stereo kalibraci lze dokonce začít pomocí manuální možnosti a následně dokončit spuštěním automatického módu.

Kapitola 8

System odečítání pozadí

Tato kapitola popisuje praktické použití metody odečítání pozadí, její konkrétní implementaci a hlavní využití myšlenky. K návrhu tohoto systému jsem se inspiroval dílčími myšlenkami metod popsanych v kapitole 5.

8.1 Úvod a definice vyhledávací tabulky

Pro vypracování, testování, ověřování výsledků, implementaci a vizualizaci jsem použil hardware a software popsany v kapitole 7.1.

V následujících podkapitolách bude konkrétněji popsán mnou vytvořený program a dále experiment na jehož základě jsem zvolil použitý barevný model. Závěrem budou vizualizovány výsledky a reprezentace uložených dat.

Cílem této dílčí části bylo navrhnout a vytvořit program, pro automatické trénování barevných modelů objektů, určených pro sledování a 3D rekonstrukci trajektorie. Dále zvolit vhodnou datovou a barevnou reprezentaci dat a navrhnout metodiku pro automatické trénování.

Hned v úvodu bude dobré vysvětlit, v jakém konkrétním kontextu bude uvažován v této a následujících kapitolách hojně se vyskytující pojem vyhledávací tabulka, anglicky **lookup table** (dále pouze **LUT**). Obecně lze říci, že v oboru informačních technologií LUT představuje rejstřík, číselník, přepočítávající či kategorizační tabulku a podobně. Na rozdíl od přepočítacích, kategorizačních, hashovacích či kódovacích tabulek obsahuje LUT výsledek operace, kde vstupem je vyhledávaný subjekt. Z matematického pohledu se dá říci, že LUT funguje jako funkce jedné či více proměnných, s omezeným konečným definičním oborem. Avšak v mém případě bude uvažováno, že LUT je tří dimenzionální (funkce tří proměnných). Jednotlivé dimenze mohou nabývat hodnot v intervalu $i \in \langle 0, 255 \rangle$. Výsledkem mohou být pouze dva stavy $s \in \{0, 255\}$, které reprezentují zda-li hledaný prvek není obsažen v LUT nebo naopak je.

8.2 Popis implementace systému odečítání pozadí

V této podkapitole bude popsána obecná funkčnost vytvořeného programu pro trénování LUT, jednotlivé metody a souslednost jejich použití.

Stejně jako v případě systému stereo kalibrace (kapitola 7.3) je vstupem do programu vždy dvojice video sekvencí. První možností je využít video v reálném čase z kamer, druhou možností je jako vstup použít dvojici předem nahraných videí. Tato dvojice videí musí pocházet ze stejného zdroje (kamery) jako videa pro kalibraci.

Nejprve bude nutné definovat několik pojmů. V následujícím textu se pod slovním spojením **pozadí** bude uvažovat LUT barev obsažených ve videosekvenci, které nemají být natrénovány pro sledování objektu a tvoří veškeré okolí v dané scéně. Pod pojmem **popředí** bude uvažována LUT, obsahující pouze barevné pixely reprezentující sledovaný objekt, tuto LUT je nutné natrénovat co nejlépe. Na její reprezentaci závisí odpovídající segmentace a sledování popsané v následující kapitole 9.

Nejprve se pokusím obecně nastínit hlavní myšlenku trénování LUT popředí, na niž je program založen. V první části se po dobu určitého času popřípadě počtu snímků ukládají odpovídající barvy pixelů do LUT modelu pozadí (v této části není přítomný objekt, jehož barva má být natrénována do LUT popředí). Následně se do obrazové scény vloží objekt, který má být natrénován do popředí. Tímto objektem je pohybováno a program se z pohybu objektu a informace o barvách odpovídajícím pozadí naučí barvy popředí. Tedy barevný model reprezentující objekt v popředí, jež bude následně možno sledovat. V následujících odstavcích budou tyto myšlenky popsány podrobněji společně s programovou implementací a využitými metodami z knihovny OpenCV.

8.2.1 Trénování LUT pozadí

Model ve formě LUT pozadí se vytvoří pro každou videosekvenci unikátně. Při trénování LUT pozadí **nesmí** být přítomen objekt, který se následně bude trénovat do popředí a bude použit pro sledování, v ideálním případě ani jemu podobná barva. Po spuštění programu se v prvních 200 snímcích analyzují obsažené barvy na všech pozicích snímku a ve formě informace se uloží na pozici jako záznam do LUT pozadí. Pozice je určena odpovídající hodnotou konkrétního barevného bodu. Hodnotu počtu 200 snímků jsem určil za dostačující pro natrénování modelu LUT pozadí.

Při trénovacích snímcích je vyžadováno, aby se uživatel, který bude následně pohybovat s barevným objektem, co nejvíce pohyboval ve scéně tak, jak bude provádět pohyby s trénovaným barevným objektem. To z důvodu lepšího natrénování LUT pozadí.

8.2.2 Trénování LUT popředí

Trénování LUT popředí je oproti trénování LUT pozadí složitější. V následujících

odstavcích se pokusím popsat více podrobněji hlavní myšlenky a formu implementace, části programu za tímto účelem vytvořeného.

Po natrénování LUT pozadí (po 200 snímcích) se program automaticky přepne do módu pro trénování LUT popředí. V této chvíli je nutné, aby uživatel začal pohybovat objektem, který má být natrénován. Nejlepší je s objektem pohybovat takovými pohyby, ve kterých bude objekt následovně sledován a jehož 3D trajektorie budou rekonstruovány.

Předpoklad je takový, že trénovaný objekt se pohybuje, proto je nutné detekovat pohybující se oblasti v obraze. Je nutné si připomenout, že následující postup je aplikován na každé video zvlášť.

První vstupní obrázek o rozlišení 640×480 byl rozdělen na dílčí obrázky o velikostech 20×20 obrazových bodů, tímto způsobem tedy vzniklo $32 \cdot 24 = 768$ obrázků. Z každého tohoto miniaturního obrázku byl vytvořen 3D histogram pomocí OpenCV metody `calcHist`. Všechny dimenze jsou nastaveny stejně, konkrétně počet možných úrovní byl nastaven na 256 (quants, y -ová osa) a počet možných hodnot byl nastaven na 32 (bins, x -ová osa). Takto vytvořené 3D histogramy jsou uloženy až do následujícího snímku.

Pro všechny následující vstupní snímky je postup stejný a lze jej shrnout do tří kroků.

1. Získání binární masky popředí: nejprve jsou vytvořeny 3D histogramy jako v případě prvního vstupního obrázku (popsáno v předchozím odstavci). Následně jsou porovnány 3D histogramy v aktuálním snímku s histogramy ve snímku předchozím, na stejných pozicích. Porovnání je provedeno pomocí metody `compareHist`, kde jako vstupy figurují dvojice porovnávaných 3D histogramů a příznakový bit, který určuje typ porovnávací metody. Zvolil jsem `CV_COMP_CORREL` což vyjadřuje, že podobnost histogramů je vypočítávána pomocí Pearsonovi korelace, v tomto případě pro histogramy H_1, H_2 následovně

$$r_{H_1, H_2} = \frac{\sum_{i=1}^N (H_1^i - \bar{H}_1) (H_2^i - \bar{H}_2)}{\sqrt{\sum_{i=1}^N (H_1^i - \bar{H}_1)^2 \cdot \sum_{i=1}^N (H_2^i - \bar{H}_2)^2}}, \quad N = 1, 2, \dots, 256^3, \quad (8.1)$$

kde \bar{H}_1 a \bar{H}_2 jsou střední hodnoty

$$\bar{H}_j = \frac{1}{N} \sum_{i=1}^N H_j^i, \quad j \in \{1, 2\}, \quad N = 1, 2, \dots, 256^3. \quad (8.2)$$

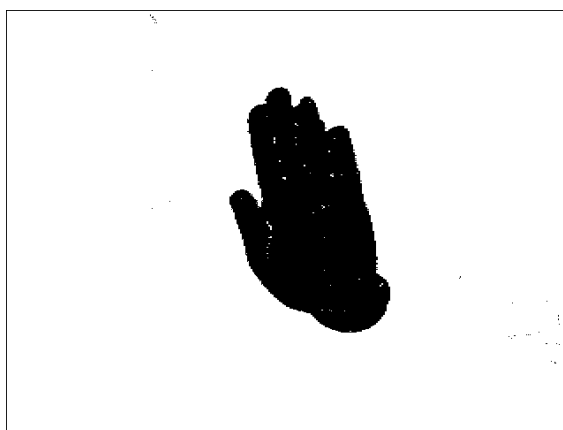
Metoda `compareHist` vrací vypočtenou hodnotu Pearsonovi korelace. Tato hodnota je využita pro určení jaké části obrazu (o velikosti 20×20 obrazových bodů zmiňovaných výše) představují pohyb mezi dvěma po sobě následujícími snímky. Pokud je tato hodnota Pearsonovi korelace menší než mnou zvolená konstanta 0.5, která podává dobré výsledky, tak je daná obrazová část vyhodnocena jako pohybující se popředí.

V ideálním případě, kdy by byly porovnávané histogramy totožné, tak by hodnota korelace byla rovná 1. Tímto způsobem je tedy vytvořená binární maska (hodnoty 0 nebo 255), která reprezentuje obrazové části, v nichž byl vyhodnocen pohyb v popředí. Na Obr. 8.1 je možno vidět příklad binární masky popředí. Hodnoty 255 (bílá) reprezentují části obrazu, kde byl vyhodnocen pohyb a hodnoty 0 (černá) reprezentují obrazové části, kde pohyb nenastal.



Obrázek 8.1: Vizualizace binární masky popředí.

2. Získání binární masky pozadí: pro získání binární masky pozadí se využije segmentace pomocí LUT, která se natrénovala z prvních 200 snímků. Tímto postupem je získána binární maska pozadí, příklad lze vidět na Obr. 8.2. Hodnoty 255 (bílá) reprezentují natrénované pozadí a hodnoty 0 (černá) reprezentují pixely, které nebyly do modelu pozadí natrénovány.



Obrázek 8.2: Vizualizace binární masky pozadí.

3. Výsledné trénování LUT: na závěr se využijí binární masky popředí a pozadí z předchozích kroků. Výpočet pixelů, které se výsledně natrénují do LUT, se provede následovně. Pro odlišení pojmů je nutné zavést **pravé popředí**. Toto pravé popředí je opět binární maska obsahující pouze pixely získané následujícím postupem. Na binární masku pozadí se použijí dvě iterace morfologické operace dilatace se strukturálním elementem o velikosti 3×3 , kde počátek se nachází na prostřední

pozici. Tímto je získáno rozšířené pozadí. Dále jsou na binární masku popředí aplikovány dvě iterace morfologické operace eroze o stejném strukturním elementu jako v předchozím případě. Tímto je získáno zmenšené popředí. K morfologickým operacím jsem využil metodu `morphologyEx`. Výsledný výpočet pravého popředí je proveden následovně. Pokud je aktuální pixel obsažen v masce popředí (hodnota 255 (bílá) na Obr. 8.1) a zároveň není obsažen v masce pozadí (hodnota 0 (černá) na Obr. 8.2)), tak je prohlášen za pixel, který je obsažen v pravém popředí (vizualizaci odpovídajících barevných pixelů lze vidět na Obr. 8.3).

Získaná binární maska pravého popředí je použita na vymaskování původního barevného snímku, pro získání odpovídajících barevných bodů, což lze vidět na Obr. 8.3 níže. Ty jsou následně natrénovány do LUT pravého popředí. Tento postup



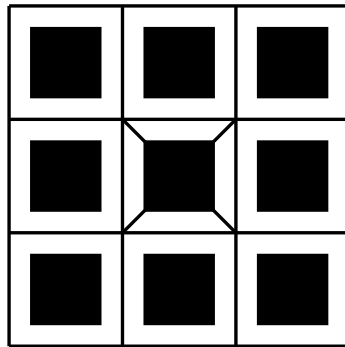
Obrázek 8.3: Vizualizace výsledných barevných bodů, které jsou natrénovány do LUT pravého popředí.

je opakován pro každý snímek, a v každém videu do té doby, než uživatel stiskne klávesu `e` (jako `End`). Před závěrečným uložením LUT je aplikována dilatace na hodnoty v LUT, popsána v podkapitole 8.2.3. Na závěr je vizualizována segmentace pomocí LUT pravého popředí, které bylo pomocí pohybujícího se barevného objektu natrénováno. Tato vizualizace slouží k optickému ověření správnosti natrénovaných barev.

Závěrem bych se chtěl věnovat otázce paměťové náročnosti dané implementace. Jak bylo naznačeno v kapitole 7.1, tak paměťové nároky jsou relativně velké. Při videu o rozlišení 640×480 obrazových bodů, velikosti obrázků pro vytvoření 3D histogramů 20×20 a počtu 32 hodnot v histogramech se využívaná paměť pohybuje kolem 750 MB. Avšak v případě pokud by bylo použito video o vysokém rozlišení, například 1920×1080 obrazových bodů, nebo se zmenšila velikost obrázků pro vytvoření 3D histogramů, popřípadě se zvětšil počet hodnot v 3D histogramech, tak by se paměťové nároky úměrně zvýšily až na jednotky GB. Proto je nutné nalézt nastavení, odpovídající požadavkům na přesnost. Mnou nastavené hodnoty, které byly popsány výše plně dostačují pro videa o velikosti 640×480 obrazových bodů.

8.2.3 Dilatace LUT

V podkapitole 8.2.2 byl podrobněji popsán postup získání LUT pravého popředí. Před samotným uložením takto získané LUT jsem navrhl vylepšení ve smyslu robustnosti, které je založeno na morfologické operaci dilatace. Tato operace byla použita na data natrénovaná v LUT. Použil jsem strukturní element (viz Obr. 8.4) o velikosti 3×3 s počátkem ve středu.



Obrázek 8.4: Použitý strukturní element s počátečním bodem ve středu.

Před závěrečným uložením natrénovaných LUT jsou aplikovány dvě iterace dilatace a to ve všech třech dimenzích. Jinak řečeno, data jsou rozšířena o dvě úrovně do všech směrů. To má za následek přidání barev do LUT, které nebyly natrénovány, ale jsou velmi blízké natrénovaným. Tímto se zaplní nespojitosti ve všech směrech. Výsledný model se stane robustnějším a přesněji dokáže segmentovat požadovaný barevný objekt.

Počet iterací dilatace, s výše zmíněným strukturním elementem, jsem zvolil o hodnotě dva, jelikož takovéto rozšíření podává dobré výsledky a nezanáší do LUT příliš mnoho neodpovídajících barev.

V následující podkapitole 8.4 jsou vizualizovány segmentace barevného objektu pomocí nedilatované a dilatované LUT, kde je názorně vidět, jak provedená dilatace pozitivně ovlivňuje získané výsledky. Následně v 8.5 je vizualizována nedilatovaná a dilatovaná LUT s odpovídajícími barvami.

8.3 Experiment volby barevného prostoru

Jak bylo nastíněno v kapitole 5.2 existuje velké množství barevných prostorů využívaných v počítačovém vidění. Tato podkapitola se zaměřuje na vhodný výběr barevného prostoru pro trénování LUT, což může být zásadní otázka při praktické použitelnosti a robustnosti metody odečítání pozadí.

Nejprve budou zmíněny barevné prostory, se kterými byl proveden následně experiment, na jehož základě byl vybrán použitý barevný model. Dále budou prezentovány výsledky, porovnání a vizualizace provedeného experimentu. Závěrem bude proveden výběr barevného prostoru a zdůvodnění tohoto výběru.

8.3.1 Uvažované barevné prostory

V této podkapitole je popsán experiment pro volbu barevného prostoru, na jehož základě jsem zvolil reprezentující prostor. Dále jsem se zaměřil na porovnání barevných prostorů. Vzhledem k rozdělení v kapitole 5.2 jsem vybral alespoň jeden barevný prostor reprezentující danou skupinu. Vyjma vjemových barevných prostorů, které nebyly uvažovány, protože vychází ze skupiny popisující barvu tak, jak jí vnímá člověk.

RGB se řadí mezi základní barevné prostory. Je tvořen červenou, zelenou a modrou složkou. Jedná se aditivní způsob míchání barev používaný v monitorech a projekto-rech. Jednotlivé složky vyzařují světlo své barvy a tím se řídí jejich míchání. Jedná se tedy o trojrozměrný prostor reprezentovaný krychlí, kde vrchol tvořený z kombinace minim hodnot složek barev nabývá černé barvy, naopak je to u maxim, které tvoří bílou barvu. Na diagonále mezi těmito dvěma extrémy leží hodnoty nazývané odstíny šedi, tedy jedná se o barvy, ve kterých jsou všechny složky zastoupeny rovnoměrně. Nespornou výhodou tohoto barevného prostoru je, že se jedná o nativní barevný prostor (reprezentaci) používaný v počítačích a také knihovně OpenCV.

YCbCr patří do skupiny ortogonálních barevných prostorů, využívá se u videa a digitálních fotografií. **Y** složka reprezentuje luminanci (hodnotu jasu) a zbývající dva **C_b** a **C_r** reprezentují modrou a červenou chromatickou komponentu. Nejedná se o absolutní barevný model, nýbrž o způsob kódování RGB barevného prostoru. Přímé zobrazení barev závisí na aktuálním užití RGB barev v datech, proto je vyjádřena hodnota **YCbCr** pouze předpokládaná.

YUV patří stejně jako předchozí barevný prostor do skupiny ortogonálních barevných prostorů. Tento barevný prostor je využíván v televizním vysílání v normě PAL i HDTV. K popisu barvy slouží **Y**, **U** a **V** komponenty, kde **Y** je jasová složka a **U**, **V** jsou barevné složky. Barevné složky se používají v rozsahu od -0.5 do +0.5, jasová složka má rozsah od 0 do 1. Výhodou **YUV** je oddělení jasové složky, kterou člověk přesněji vnímá. Pak je možné vyhradit pro chromatickou složku menší šířku přenosového pásma. Z historického hlediska, byl tento barevný prostor vytvořen za účelem potřeby vytvořit způsob přenosu barevného signálu, který by byl kompa-

tibilní s černobílým vysíláním. Kde ke stávající jasové složce byla přidána složka barevná.

CIE-Lab je členem vjemově jednotných barevných prostorů. Jedná se o jeden z mnoha možných grafických gramutů, dále se jedná o absolutní vyjádření barvy nezávisle na zařízení. Jednotlivé osy reprezentují, L světelnost (lightness) v rozmezí 0 až 100. Osa **a** popisuje barevný přechod zelená-purpurová (od záporných hodnot po kladné) a osa **b** přechod modrá-žlutá (od záporných hodnot po kladné). Jedná se o nelineární barevný prostor založený na vnímání barev pomocí lidských očí. CIE-Lab byl odvozen z barevného prostoru CIE-XYZ.

CIE-Luv patří do stejné skupiny jako CIE-Lab, byl taktéž odvozen z barevného prostoru CIE-XYZ. Je široce používán tam, kde se pracuje s barevnými světelnými zdroji, například v počítačové grafice. Zde jednotlivé komponenty reprezentují světelnost L (lightness) a u, v jsou chromatické komponenty. Jedná se také jako v předchozím případě o nelineární barevný prostor založený na vnímání barev pomocí lidských očí.

Všechny výše zmíněné barevné prostory mají společné vlastnosti a to, že jsou reprezentovány pomocí tří komponent a jejich reprezentace je implementována v OpenCV.

8.3.2 Nastavení a popis experimentu

Úvodem je nutné zmínit, že popis implementace daného programu použitého pro experiment byl popsán v podkapitole s pořadovým číslem 8.2.

Nejprve je potřeba specifikovat předpoklady experimentu, nastavení obrazové scény a omezení. Provedl jsem omezení ve formě konečného počtu testovaných barevných prostorů, jejichž popis lze najít v předchozí podkapitole 8.3.1 a obecně v 5.2. Dále jsem předpokládal, že barevný objekt, který byl natrénován do LUT byl zbarven do unikátní barvy, která se jinde ve scéně nevyskytovala. Barva objektu nebyla předem známa programem. Program jsem navrhl tak, aby byl co nejjobecnější, nezáleželo na tvaru a barvě objektu, jehož barevný model se má natrénovat. Dále, aby program nebyl nepříznivě ovlivňován světelnými podmínkami, bylo nutné zajistit neměnné osvětlení. Toho jsem docílil provedením experimentu v noci za využití umělého osvětlení. Postup trénování byl podrobněji popsán v podkapitole 8.2.

Na Obr. 8.5 v levé části je ukázán pohled na obrazovou scénu společně s červeně zbarveným čtvercovým obrazcem, určeným pro natrénování, v pravé části lze vidět ruční segmentaci červeného obrazce.

Pro účel experimentu byla nahrána dvě videa. První trénovací video, pomocí kterého se natrénovaly jednotlivé LUT odpovídajících barevných modelů pro následnou segmentaci. Dále druhé testovací video, které se lišilo od prvního videa pohyby s trénovacím obrazcem.

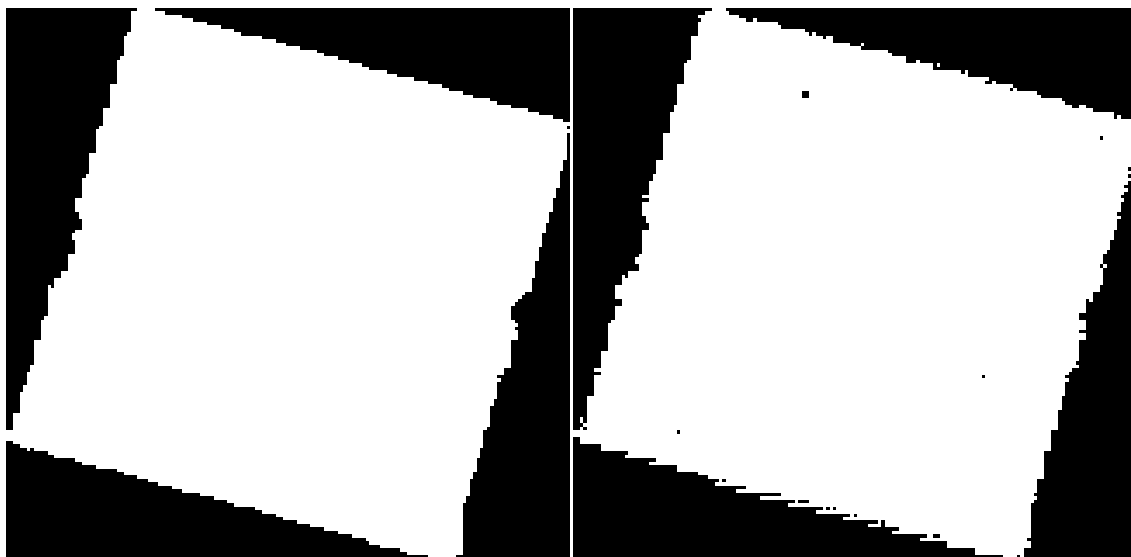
Po natrénování všech zmíněných LUT barevných modelů, byly aplikovány natrénované LUT na druhé testovací video, které obsahovalo jiné pohyby s trénovacím



Obrázek 8.5: V levé části je zobrazen pohled na obrazovou scénu a trénovaný barevný obrazec. V pravé části se nachází odpovídající ruční segmentace červeného trénovacího obrazce.

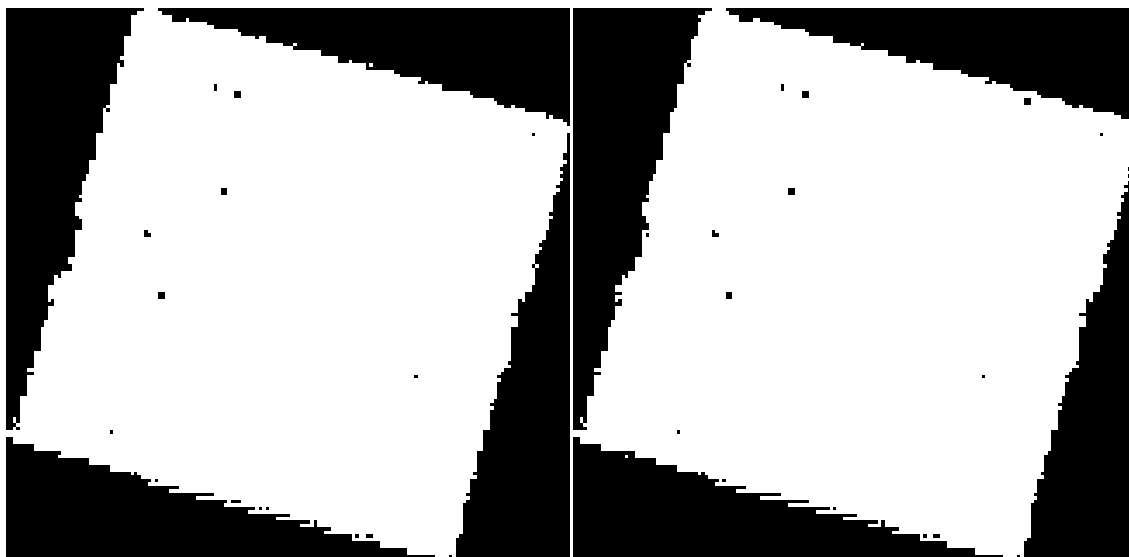
vzorem. Následně byla provedena segmentace pomocí jednotlivých LUT a uloženy výsledky ve formě segmentovaných videí. Tyto videa byla rozdělena na 436 jednotlivých snímků. Originální nasegmentované video bylo rozděleno také na jednotlivé snímky a byla provedena ruční segmentace, jejíž příklad lze vidět na Obr. 8.5 v pravé části, nebo detailněji na Obr. 8.6.

Na následujících obrázcích lze vidět detail ruční segmentace (Obr. 8.6), dále automatické segmentace pomocí LUT barevných modelů RGB (Obr. 8.7), $YCbCr$ (Obr. 8.8), YUV (Obr. 8.9), CIE-Lab (Obr. 8.10) a CIE-Luv (Obr. 8.11). Pro vizualizované segmentace byl zvolen 156. obrázek z videosekvencí.



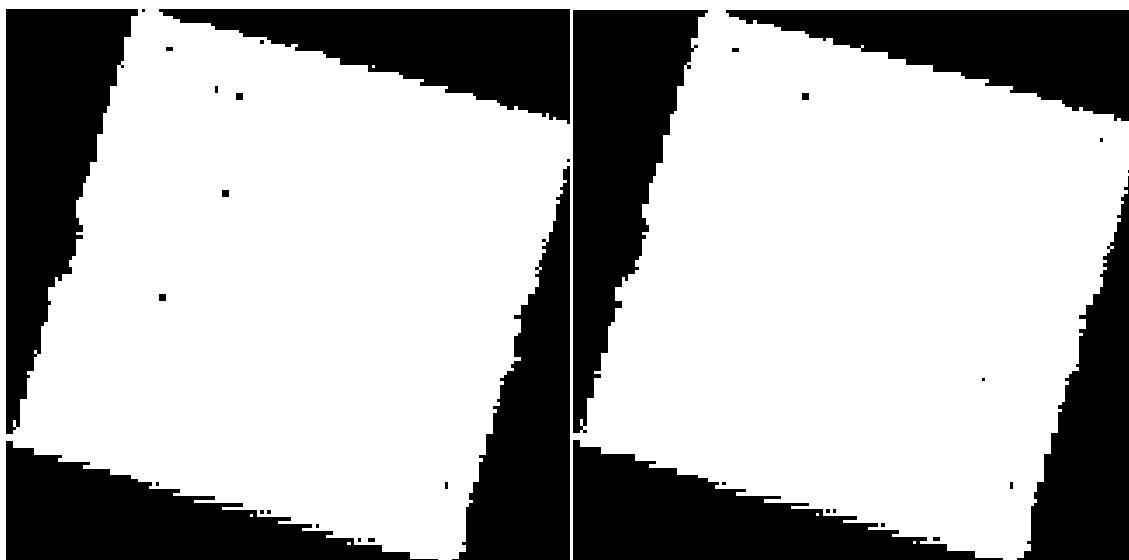
Obrázek 8.6: Detail ruční segmentace.

Obrázek 8.7: Detail RGB segmentace.



Obrázek 8.8: Detail YC_bC_r segmentace.

Obrázek 8.9: Detail YUV segmentace.



Obrázek 8.10: Detail CIE-Lab segmentace. Obrázek 8.11: Detail CIE-Luv segmentace.

Získané výsledky jsou prezentovány v podkapitole 8.3.3, dále postup jejich vyhodnocení je popsán v 8.3.4.

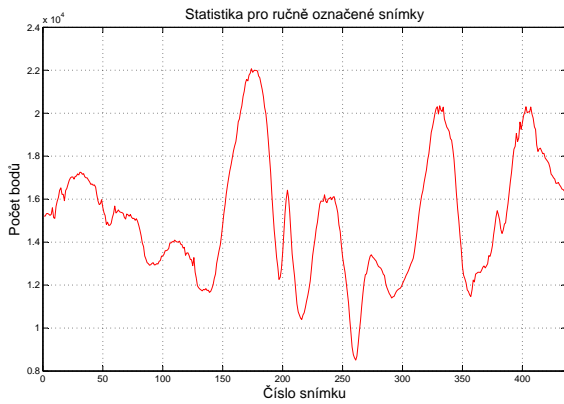
8.3.3 Výsledky

V této podkapitole budou podrobněji popsány konkrétní výpočty na získaných datech. Nejprve je nutné zmínit, že video bylo o rozlišení 640×480 obrazových bodů, celkový počet snímků v nahraném testovacím videu byl 436. Pohyby s červeným obrazcem se neshodovali s trénovacím videem. Na natrénované LUT nebyly aplikovány žádné vylepšení, například formou dilatace viz podkapitola 8.2.3.

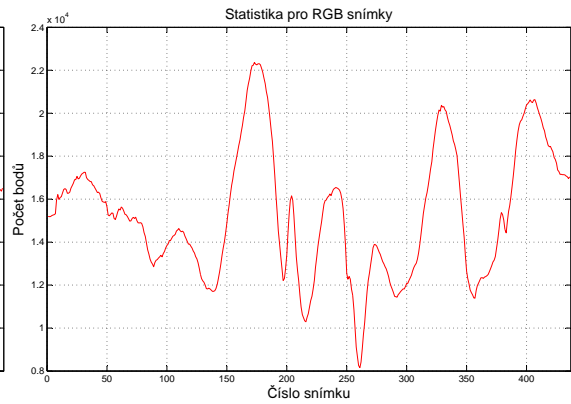
Pomocí makra v MATLABU, které jsem vytvořil, bylo nutné spočítat počty bodů v jednotlivých snímcích ručně označené segmentace a všech ostatních automatických

KAPITOLA 8. SYSTÉM ODEČÍTÁNÍ POZADÍ

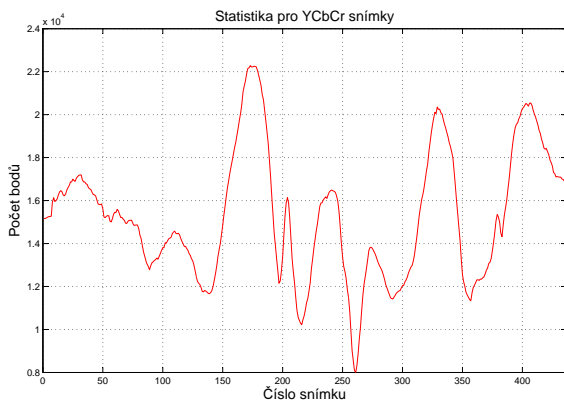
segmentací pomocí natrénovaných LUT. Grafy počtů pixelů v jednotlivých snímcích jsou zobrazeny níže na Obr. 8.12 - 8.17. Na x -ové ose grafu jsou jednotlivé snímky a na y -ové ose k nim odpovídající počet segmentovaných pixelů.



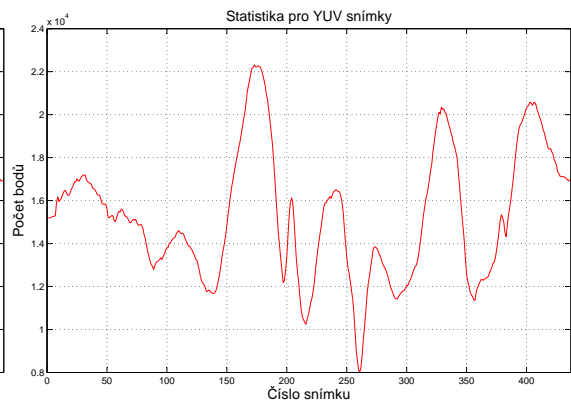
Obrázek 8.12: Detekované počty bodů v ručně označených snímcích.



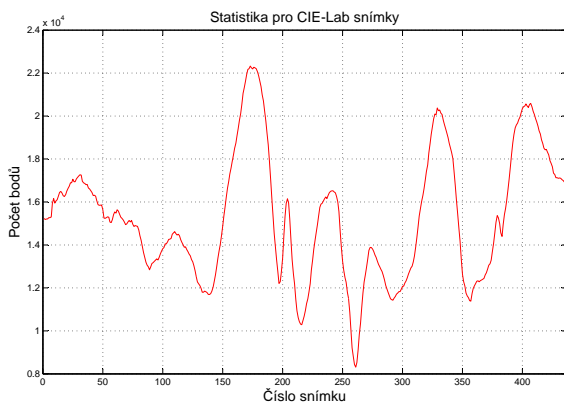
Obrázek 8.13: Detekované počty bodů v RGB snímcích.



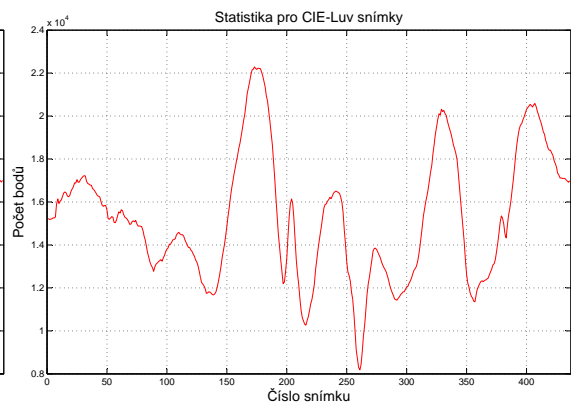
Obrázek 8.14: Detekované počty bodů v YCbCr snímcích.



Obrázek 8.15: Detekované počty bodů v YUV snímcích.



Obrázek 8.16: Detekované počty bodů v CIE-Lab snímcích.



Obrázek 8.17: Detekované počty bodů v CIE-Luv snímcích.

Z grafů lze vidět, že průběhy počtů pixelů přes jednotlivé segmentace jsou si velice podobné, avšak vzhledem k velikosti počtů pixelů v řádech 10^4 není dostatečně zřetelný rozdíl, což bude diskutováno dále v této podkapitole.

Počty segmentovaných pixelů byly vypočteny následovně

$$p = \sum_{i=1}^H \sum_{j=1}^W I(i, j), \quad \forall I(i, j) > 0, \quad (8.3)$$

kde $I(i, j)$ reprezentuje daný snímek na pozici (i, j) , v mém případě jsou hodnoty mezí $H = 480$ (počet řádků v obrázku, height) a $W = 640$ (počet sloupců v obrázku, width).

Výpočet celkového počtu pixelů přes všech $N = 436$ snímků jsem získal následovně

$$s_k = \sum_{n=1}^N \sum_{i=1}^H \sum_{j=1}^W I_n(i, j), \quad \forall I_n(i, j) > 0, \quad k = 1, 2, \dots, 6, \quad (8.4)$$

kde s_k reprezentuje k -tou segmentaci pomocí odpovídajícího barevného prostoru v pořadí, jak byly prezentovány na Obr. 8.12 - 8.17, I_n představuje n -tý segmentovaný snímek, H a W jsou stejné jako v předchozím případě.

Dále jsem určil absolutní procentuální odchylku pr_k od s_1 , představující ruční segmentaci, následovně

$$pr_k = \left| \frac{100 \cdot s_k}{s_1} - 100 \right| [\%], \quad k = 1, 2, \dots, 6. \quad (8.5)$$

Výsledky výpočtů jsou přehledně shrnuty v následující tabulce 8.1.

	Ručně označené	RGB	YCbCr	YUV	CIE-Lab	CIE-Luv
s_k [px]	6620660	6672173	6654509	6658216	6668386	6660115
pr_k [%]	0	0.7781	0.5113	0.5673	0.7209	0.5959

Tabulka 8.1: Počty bodů a absolutní procentuální rozdíl od ručně označené segmentace.

Z výše uvedené tabulky lze říci, že LUT natrénovaná barevným modelem YCbCr v mnou provedeném experimentu poskytuje nejmenší absolutní procentuální rozdíl přes všechny snímky, to o hodnotě $pr_3 = 0.5113$ [%]. V obecnosti lze říci, že všechny posuzované barevné prostory na dané zkoumané množině snímků dosahují absolutní procentuální chyby menší než 0.8% a maximální difference $d_{\max} = |pr_2 - pr_3| = 0.2108$ [%], což na daném počtu snímků představuje značně zanedbatelnou hodnotu.

Na následujících grafech na Obr. 8.18 - 8.22 jsou zobrazeny průběhy velikostí absolutních chyb

$$\epsilon_{n,k} = |p_{n,k} - p_{n,1}|, \quad n = 1, 2, \dots, 436, \quad k = 1, 2, \dots, 6, \quad (8.6)$$

kde $\epsilon_{n,k}$ představuje k -tý průběh velikosti absolutní chyby vzhledem k ručně označené segmentaci bodů $p_{n,1}$. První hodnota $\epsilon_{n,1}$, $\forall n$ bude nulová vzhledem k tomu, že se porovnávají stejná data. Proto graf průběhu této chyby není dále prezentován.

Hodnotu součtu velikosti absolutní chyby přes všechny snímky jsem vypočítal jako

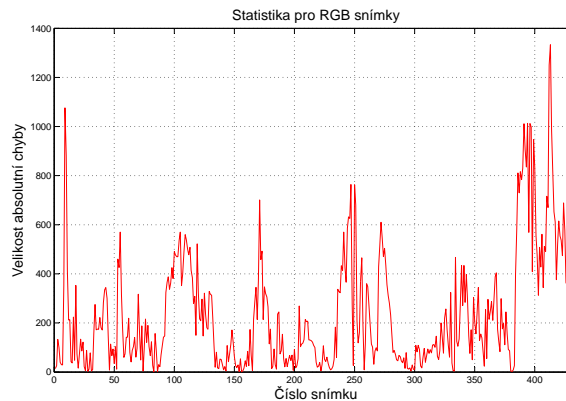
$$s_{\epsilon_k} = \sum_{n=1}^N |p_{n,k} - p_{n,1}|, \quad k = 1, 2, \dots, 6, \quad (8.7)$$

přičemž $N = 436$ představuje všechny snímky a výsledné hodnoty lze vidět v následující tabulce 8.2. Z vypočtených hodnot, byl učiněn obdobný závěr jako v případě absolutních procentuálních rozdílů z tabulky 8.1. Avšak v tomto případě je možno si lépe představit velikost chyby. Pokud bude v úvahu brán součet nejmenší velikosti absolutní chyby, tedy $s_{\epsilon_3} = 33849$ pixelů v případě barevného modelu YC_bC_r , tedy velikost průměrné absolutní chyby na jeden snímek je roven přibližně 78 pixelům. Což odpovídá 0.0253% celkové plochy snímku o rozlišení 640×480 obrazových bodů. Je možno usuzovat, že se jedná o zanedbatelnou chybu i v případě největšího součtu velikosti absolutní chyby (barevný model RGB, $s_{\epsilon_2} = 51513$ pixelů), která tvoří pouze 0.0385% (118 pixelů) celkové plochy jednoho snímku o stejném rozlišení.

	Ručně označené	RGB	YC_bC_r	YUV	CIE-Lab	CIE-Luv
s_{ϵ_k} [px]	0	51513	33849	37556	47726	39455

Tabulka 8.2: Hodnoty součtů velikostí absolutních chyb v pixelech přes všechny snímky.

Dále následuje grafická reprezentace průběhů velikostí absolutních chyb pro jednotlivé segmentace pomocí natrénovaných LUT v konkrétních barevných prostorech, viz Obr. 8.18 - 8.22.

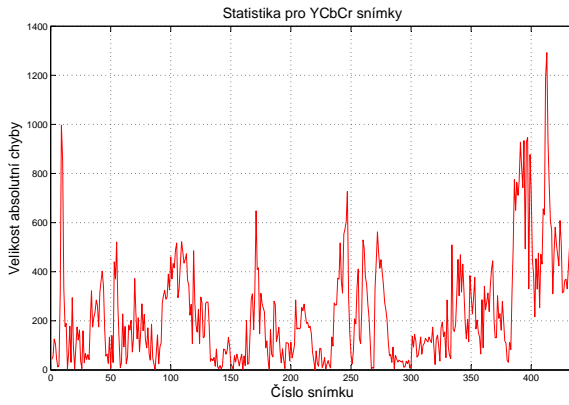


Obrázek 8.18: Chyba RGB.

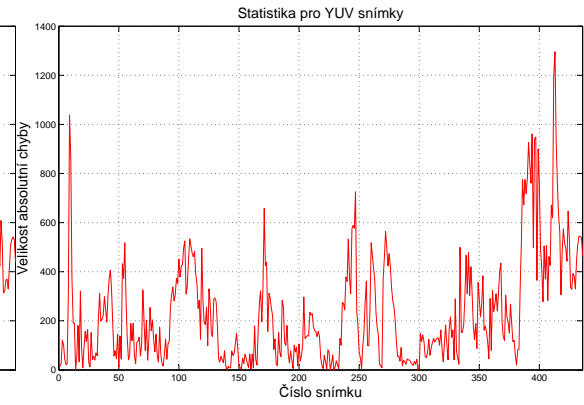
Dále jsem určil relativní chybu v každém snímku jako

$$\eta_{n,k} = \left| \frac{p_{n,k} - p_{n,1}}{p_{n,1}} \right|, \quad k = 2, 3, \dots, 6, \quad n = 1, 2, \dots, 436, \quad (8.8)$$

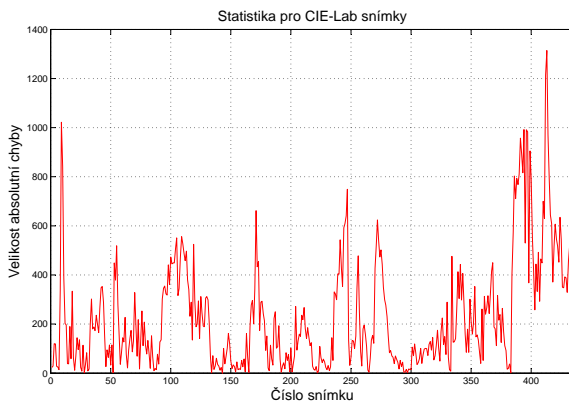
KAPITOLA 8. SYSTÉM ODEČÍTÁNÍ POZADÍ



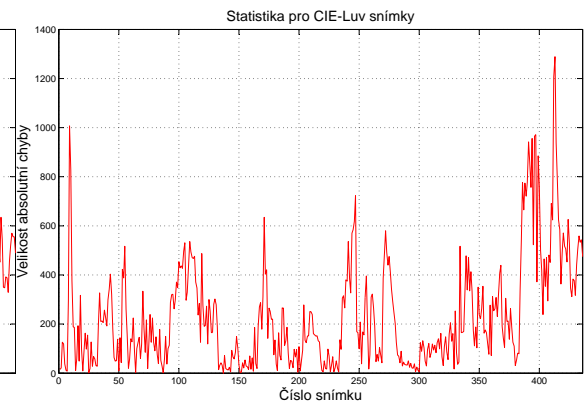
Obrázek 8.19: Chyba $YCbCr$.



Obrázek 8.20: Chyba YUV .



Obrázek 8.21: Chyba $CIE-Lab$.



Obrázek 8.22: Chyba $CIE-Luv$.

kde $p_{n,k}$ označuje součet segmentovaných bodů v n -tém snímku k -tého barevného prostoru. Odpovídající střední hodnotu relativní $\bar{\eta}_{n,k}$ vypočítám následovně

$$\bar{\eta}_k = \frac{1}{N} \sum_{n=1}^N \left| \frac{p_{n,k} - p_{n,1}}{p_{n,1}} \right|, \quad k = 2, 3, \dots, 6, \quad N = 436, \quad (8.9)$$

kteřou je možné vyjádřit procentuálně

$$\bar{\delta}_k = \left(\frac{1}{N} \sum_{n=1}^N \left| \frac{p_{n,k} - p_{n,1}}{p_{n,1}} \right| \right) \cdot 100 [\%], \quad k = 2, 3, \dots, 6, \quad N = 436. \quad (8.10)$$

	Ručně označené	RGB	$YCbCr$	YUV	CIE-Lab	CIE-Luv
$\bar{\eta}_k$	0	0.0155	0.0152	0.0151	0.0148	0.0148
$\bar{\delta}_k$ [%]	0	1.5472	1.5170	1.5064	1.4829	1.4794

Tabulka 8.3: Střední hodnoty relativních chyb a jejich procentuální hodnoty.

Konkrétní vypočtené hodnoty rovnic 8.9 a 8.10 jsou vidět v tabulce 8.3 výše, z porovnání hodnot vyplývá, že žádný z barevných prostorů není výrazně lepší než zbývající.

8.3.4 Vyhodnocení výsledků a výběr barevného prostoru

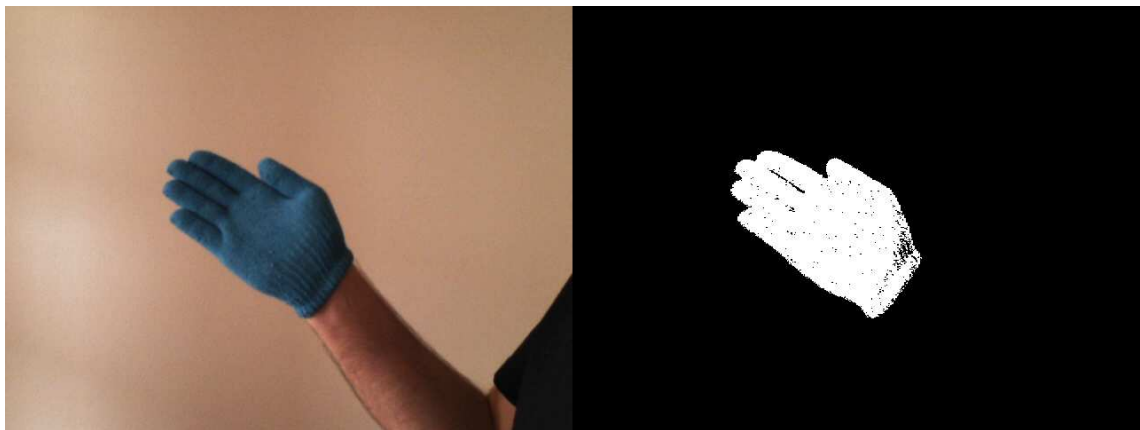
Vzhledem k výsledkům získaným v předešlé podkapitole 8.3.3, nelze jednoznačně rozhodnout o volbě nejlepšího barevného prostoru, jelikož rozdíly ve výsledcích jsou ve srovnání s ostatními, velmi malé a zanedbatelné.

Z důvodu velké náročnosti nebyl experiment opakován, ale ze získaných dat je možné usuzovat, že by dopadl velmi podobně. Tedy rozdíly mezi barevnými modely by byly minimální. Dále by mohlo být diskutováno nad zanesením chyby díky lidskému faktoru, jelikož ruční segmentace byla velmi obtížně proveditelná například ve snímcích, ve kterých barevný objekt vykazoval rychlejší pohyb. Hrany byly rozmazané a velice těžce bylo rozhodováno na úrovni pixelů, který z daných pixelů patří do barevného objektu, a který již nikoliv.

Vzhledem k dosaženým výsledkům uskutečněného experimentu jsem se rozhodl, že pro trénování LUT použiji RGB barevný prostor. Dalším faktem pro použití RGB barevné reprezentace je, že představuje nativní reprezentaci v počítači a hlavně v použité knihovně OpenCV. Nebude proto nutné každou dvojici snímků převádět do požadovaného barevného prostoru a ušetřím tím výpočetní čas.

8.4 Vizualizace segmentace pomocí LUT

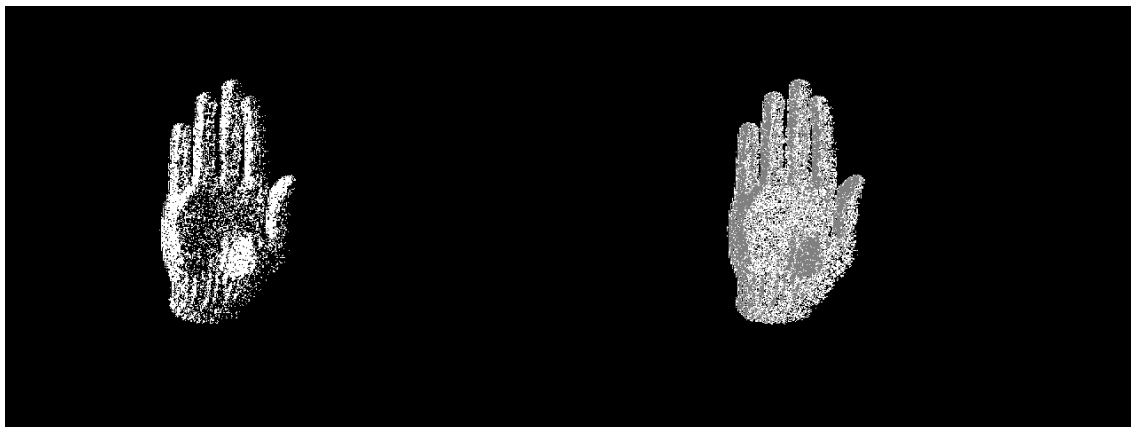
Na Obr. 8.23 je možno vidět v levé části originální barevný snímek z kamery a v pravé polovině výslednou segmentaci pomocí natrénované LUT dilatované postupem popsaným v podkapitole 8.2.3.



Obrázek 8.23: Levá část představuje originální snímek z kamery a pravá polovina výslednou segmentaci pomocí dilatované LUT.

Pro vizualizaci segmentace před a po aplikování dilatace jsem vytvořil jednoduchý program, který načte natrénované a dilatované LUT obou kamer a vizualizuje odpovídající segmentace. Možný výsledek je vidět na Obr. 8.24. Kde v levé části je hodnotou 255 (bílá) zobrazena segmentace LUT před dilatací. V pravé části jsou hodnotou 128 (šedá) obarveny ty pixely, které pochází z LUT před dilatací a hod-

notou 255 (bílá) jsou obarveny ty pixely, které byly přidány pomocí dilatace.



Obrázek 8.24: Detail segmentací, v levé polovině segmentace pomocí LUT před dilatací, v pravé polovině po dilataci.

Z Obr. 8.24 je zřejmé, že využití dilatace pozitivně ovlivňuje výslednou segmentaci, aniž by bylo nutné provádět další složité výpočetní operace.

8.5 Vizualizace LUT

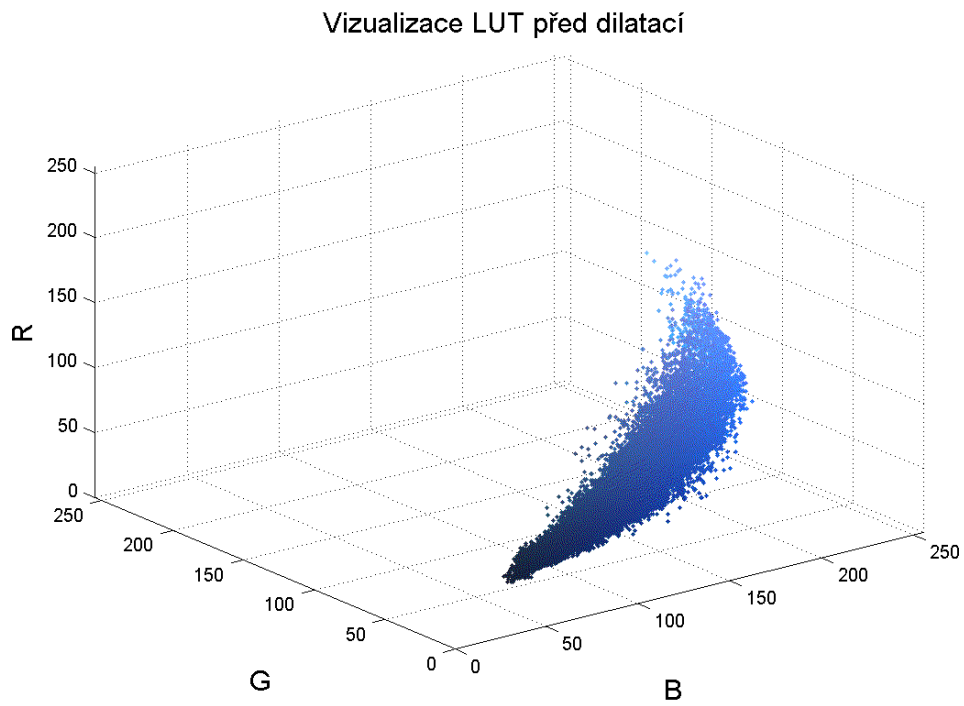
Pro účel vizualizace jednotlivých LUT jsem vytvořil v MATLABu několik funkcí, které vykreslí jednotlivé body na pozici ve 3D prostoru. Tento barevný prostor, jak bylo popsáno dříve, je omezen na jednotlivých osách intervalem $i \in \langle 0, 255 \rangle$. Tyto hodnoty dohromady reprezentují barevné složky, v mém případě RGB. Na Obr. 8.25 je vidět vizualizace možného výsledku natrénované LUT, kde jednotlivé body mají odpovídající barvu.

Na Obr. 8.26 je vidět výsledek po dvou iteracích dilatace aplikovaných na data vizualizovaná na Obr. 8.25. Byl použit strukturní element popsáný v podkapitole 8.2.3. Jednotlivé barvy bodů odpovídají jejich skutečným RGB hodnotám.

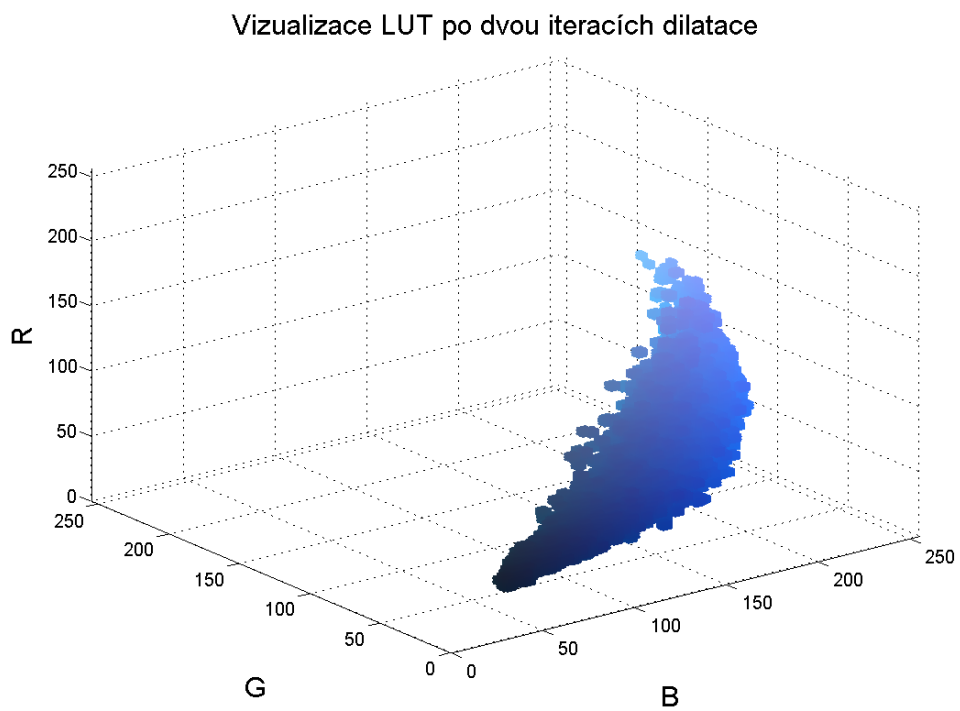
	Před dilatací	Po dilataci
Počet pixelů	12689	235896

Tabulka 8.4: Počty pixelů v jednotlivých LUT.

Počty pixelů před a po dilataci LUT jsou prezentovány v tabulce 8.4. Je zřejmé, že pokud by byl postup popsané dilatace aplikován na jediný bod, tak by výsledkem bylo 5^3 bodů, což odpovídá 125-ti násobku nárůstu informace. V praktickém konkrétním případě, jak vyplývá z tabulky 8.4, je tento nárůst informace pouze přibližně 18.6-ti násobkem. Počet přidávaných bodů pomocí dilatace je závislý na rozmístění jednotlivých bodů v barevném prostoru.



Obrázek 8.25: Vizualizace možného výsledku natrénované RGB barevné LUT.



Obrázek 8.26: Vizualizace možného výsledku natrénované RGB barevné LUT po aplikování dvou iterací dilatace.

Kapitola 9

System sledování a 3D rekonstrukce

V předešlých kapitolách byl popsán postup stereo kalibrace a trénování LUT, výstupy z těchto systémů budou využity pro sledování barevného objektu a rekonstrukci jeho 3D trajektorie.

Níže bude podrobněji popsána programová implementace a obecné myšlenky, na kterých byl systém sledování a 3D rekonstrukce založen. V závěru budou vizualizovány příklady 3D trajektorií.

9.1 Popis implementace systému sledování a 3D rekonstrukce

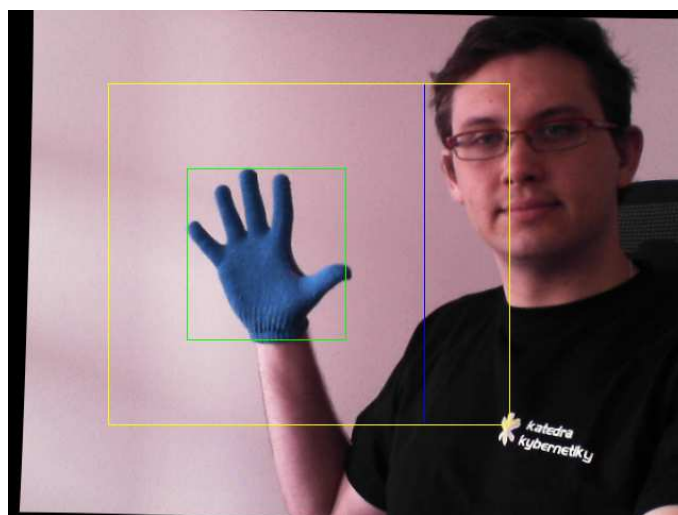
V následujících odstavcích bude popsán postup pro jednu kameru, tyto operace se vždy provádí totožně pro obě kamery (pokud není psáno jinak). K realizaci a implementaci jsem využil hardware a software popsány v podkapitole 7.1. Pro úspěšné spuštění programu je nutné dodat výstupy ze systémů popsány v předešlých kapitolách 7 a 8. Konkrétně jsou vyžadovány

- matice vnitřních parametrů kamer M_1, M_2 ,
- vektory deformačních parametrů kamer $\mathbf{d}_1, \mathbf{d}_2$,
- rotační matice jednotlivých kamer R_1, R_2 ,
- projekční matice kamer P_1, P_2 ,
- matice Q reprezentující transformaci mezi mapou disparit a hloubkovou mapou a
- natrénované dilatované LUT pro dvojici kamer.

Po úspěšném načtení výše uvedených parametrů jsou vypočteny transformační mapy pro jednotlivé osy x a y . K tomuto účelu je volána metoda `initUndistortRectifyMap`, jejíž vstupními parametry jsou M , D , R , P a velikost obrázku v pixelech. Tyto mapy transformují obraz vzhledem k deformačním parametrům a odhadnutým maticím.

Před dalším popisem funkčnosti je nutné se zmínit o algoritmu pro sledování barevného objektu. Přehled metod pro sledování byl prezentován v podkapitole 6. Veškeré zmíněné metody jsou schopny sledovat daný objekt, avšak pro účel této jednoduché úlohy by byly příliš složité. Proto jsem se jimi inspiroval a navrhl vlastní jednoduchý algoritmus. Nyní popíši jeho hlavní myšlenku, na které je algoritmus založen. V prvním snímku algoritmus nemá žádnou počáteční informaci, proto je provedena segmentace pomocí LUT na celé velikosti snímku. Po segmentaci barevného objektu je nalezen **nejmenší pravoúhlý ohraničující rámeček (zelená barva)**. Dále předpokládám, že změna polohy barevného objektu je ve dvou po sobě jdoucích snímcích malá. Proto na základě velikosti nejmenšího ohraničujícího rámečku (je nutno dodat, že ohraničující rámeček má osy zarovnané s osami snímku) určím dvojnásobnou velikost **adaptivního ohraničujícího rámečku (modrá barva)**. Ten se využije pro následující snímek, proto se již nemusí segmentovat celý snímek, ale jen jeho malá část. Což velmi šetří výpočetní čas. Dále pokud se segmentace barevného objektu zmenší, alespoň v jedné z os, pod velikost 20 pixelů, tak je adaptivní ohraničující rámeček nastaven na velikost celého snímku. To samé se stane v případě pokud sledovaný objekt není ve scéně. Tudíž se segmentuje opět celý snímek. Tento postup je proveden pro každou kameru zvlášť.

Před spuštěním hlavní smyčky, je nastaven adaptivní pravoúhlý ohraničující rámeček na velikost celého snímku. Dále jsou nastaveny parametry pro metodu `StereoBM`. Parametry lze před samotným spuštěním 3D rekonstrukce nastavit manuálně, pro konkrétní vzdálenosti od kamer, pomocí klávesových zkratk popsanych v příloze C.



Obrázek 9.1: Vizualizace jednotlivých ohraničujících rámečků.

Využil jsem metody `remap` pro transformaci původního snímku z kamery podle předem vypočítaných x -ových a y -ových transformačních map. Ze snímku se vyřízne

obrázek na stejné pozici a velikosti jako adaptivní rámeček. Tento menší obrázek je segmentován pomocí natrénované LUT a na získaný binární obraz se aplikují dvě iterace morfologické operace otevření, pro zaplnění děr. V binární masce se naleznou nová pozice objektu a je přepočítán adaptivní rámeček na odpovídající hodnoty. Pokud není nalezen objekt, tak je nastaven adaptivní rámeček na celý snímek.

V případě úspěšného nalezení objektu v obou kamerách je vypočítán, z dvou adaptivních rámečků, **nejmenší společný obdélník** (žlutá barva), představující rámeček, ve kterém je objekt v obou kamerách. Společný obdélník je využit pro vyřiznutí oblasti v barevných obrázcích, ty jsou následně segmentovány pomocí LUT a pixely odpovídající binární masce jsou převedeny do odstínů šedi. Metoda StereoBM pro výpočet mapy disparit vyžaduje jako vstup dva snímky v odstínech šedi s odpovídajícími pixely. Vypočtená mapa disparit je následně vizualizována.

V dalším kroku je využita mapa disparit pro výpočet 3D souřadnic v reálné metrice. V OpenCV existuje pro tento účel metoda `reprojectImageTo3D`, která jako vstup vyžaduje právě mapu disparit a výše zmíněnou matici Q . 3D souřadnice jsou vypočítány následovně

$$[\tilde{x}, \tilde{y}, \tilde{z}, \tilde{w}]^T = Q \cdot [x, y, \text{Disparity}(x, y), 1]^T, \quad (9.1)$$

$$I_{3D}(x, y) = \begin{pmatrix} \tilde{x} & \tilde{y} & \tilde{z} \\ \tilde{w} & \tilde{w} & \tilde{w} \end{pmatrix}. \quad (9.2)$$

Souřadnice se znakem $\tilde{}$ odpovídají nenormovaným hodnotám ve 4. dimenzi, x a y určují polohu v původním velkém obrázku, pro který byla vypočítána matice Q reprezentující transformaci mezi mapou disparit a hloubkovou mapou. $\text{Disparity}(x, y)$ je hodnota mapy disparit na pozici x, y a $I_{3D}(x, y)$ odpovídá pixelu na pozici x, y , pro který byly zrekonstruovány výsledné 3D souřadnice v reálné metrice.

Metodu `reprojectImageTo3D` lze využít pouze pro případ rekonstrukce celého snímku, což je patrné i z výpočtu v rovnici (9.1), jelikož je k výpočtu využívána informace o pozici rekonstruovaného pixelu. Z uvedeného vyplývá, že jsem tuto metodu nemohl využít, jelikož 3D rekonstrukce není aplikována na celý snímek. Proto jsem využil zmíněné vzorce (9.1), (9.2) a nejmenší společný obdélník k vlastní implementaci této metody, která je schopna rekonstruovat libovolně velký obrázek, až do velikosti originálního snímku. 3D rekonstrukce a volání metody pro její výpočet je velmi časově náročná operace. Popsaným způsobem se podařilo velmi razantně snížit časové nároky pro výpočet 3D souřadnic, v závislosti na velikosti části obrázku určeného pro rekonstrukci.

V předchozích odstavcích byl popsán postup pro 3D rekonstrukci bodů části snímku. Pro reprezentaci trajektorie objektu ve 3D prostoru je nutné určit bod, který bude reprezentovat daný objekt v každém snímku. Uložením takto získaných bodů lze následně vizualizovat 3D trajektorii. V podkapitole 9.1.1 bude podrobněji popsán způsob výběru reprezentujícího 3D bodu a vylepšení odstraňující špatně rekonstruované body.

Pro uložení trajektorie je vyžadován od uživatele vstup ve formě stisknutí klávesy `s` (jako Save). Pro ukončení je nutné stisk klávesy `s` opakovat ještě jednou. Trajek-

torie je uložena do souboru `xyzt.txt`, kde jednotlivá písmena reprezentují pořadí daných souřadnic. Jednotlivý řádek představuje jednu 3D polohu bodu (v milimetrech) a informace o čase v jakém byl bod zaznamenán vzhledem k předchozímu bodu (v milisekundách). Z této časové informace by mohla být zrekonstruována popřípadě i rychlost a zrychlení sledovaného objektu v prostoru. Při opakovaném spuštění 3D rekonstrukce trajektorie je tento soubor přemazán aktuálními daty.

Na závěr v podkapitole 9.2 budou prezentovány příklady rekonstruovaných 3D trajektorií.

9.1.1 Výpočet a vylepšení reprezentujícího 3D bodu

Postup 3D rekonstrukce objektu v části snímku ohraničeného nejmenším společným obdélníkem byl popsán v podkapitole 9.1. Nyní bude popsáno jakým způsobem je vypočítán bod reprezentující objekt v daném snímku. Tento postup je aplikován na každý snímek, výsledkem je trajektorie pohybu objektu v 3D prostoru.

Z rekonstruovaných 3D bodů objektu $\mathbf{x}_i = [x_i, y_i, z_i]^\top$ se spočítají průměrné hodnoty ve všech osách. Příklad výpočtu pro x -ovou osu

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i. \quad (9.3)$$

Průměrné hodnoty společně reprezentují jeden 3D bod nazvaný **centroid** $\mathbf{x}_c = [\bar{x}, \bar{y}, \bar{z}]^\top$. Následně se určí euklidovské vzdálenosti od centroidu ke všem 3D bodům objektu jako

$$l_i = \sqrt{(x_i - \bar{x})^2 + (y_i - \bar{y})^2 + (z_i - \bar{z})^2}. \quad (9.4)$$

Ze vzdáleností se vypočítá průměrná hodnota

$$\bar{l} = \frac{1}{N} \sum_{i=1}^N l_i. \quad (9.5)$$

Všechny body \mathbf{x}_j , jejichž vzdálenost l_j k centroidu \mathbf{x}_c je menší než průměrná vzdálenost, tedy platí

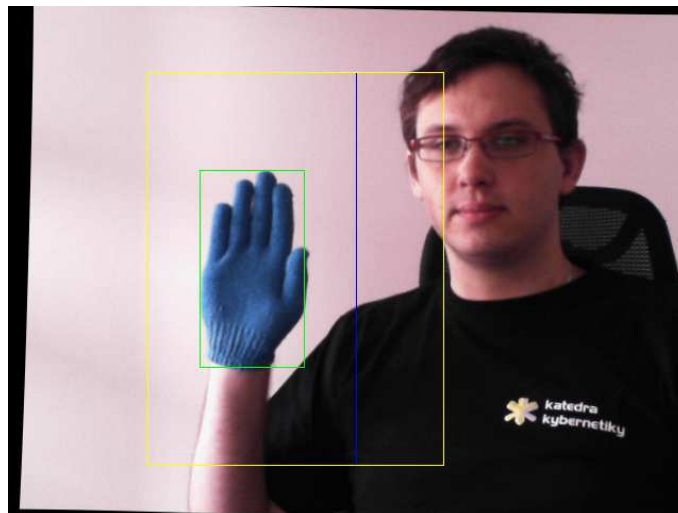
$$l_j < \bar{l}, \quad (9.6)$$

jsou použity pro výpočet **reprezentujícího 3D bodu** $\mathbf{x}_r = [\bar{x}_r, \bar{y}_r, \bar{z}_r]^\top$, který se určí opět pomocí průměrných hodnot v jednotlivých osách. Příklad výpočtu pro x -ovou osu

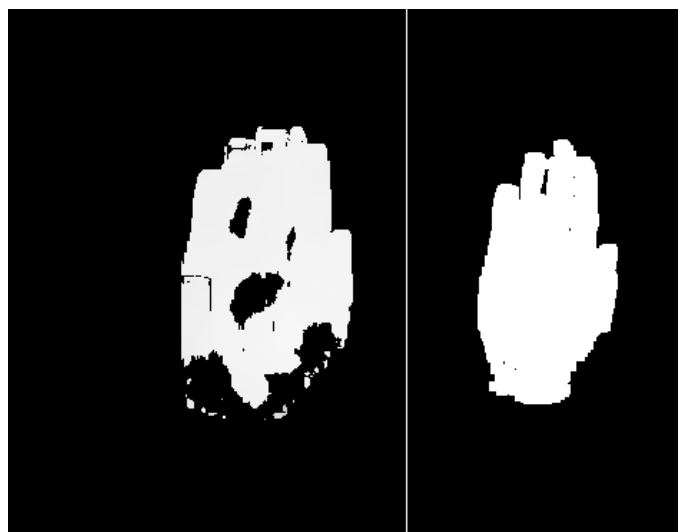
$$\bar{x}_r = \frac{1}{N} \sum_{j=1}^N x_j, \quad \forall \mathbf{x}_j \text{ pro které platí } l_j < \bar{l}. \quad (9.7)$$

Tímto postupem se dají eliminovat vzdálené body a zpřesnit reprezentující 3D bod.

Postup prezentuji na následujícím příkladě. Originální snímek lze vidět na Obr. 9.2. Obr. 9.3 obsahuje odpovídající mapu disparit (levá část) a binární masku (pravá část) získanou pomocí segmentace s natrénovanou LUT.

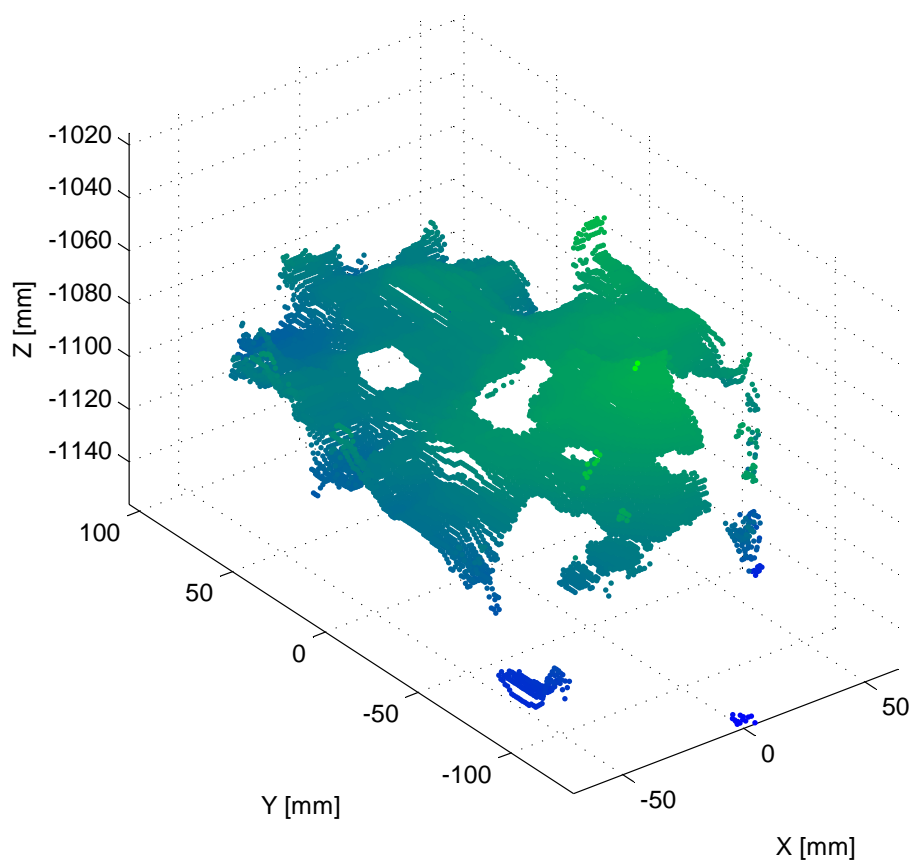


Obrázek 9.2: Snímek určený k 3D rekonstrukci a výpočtu reprezentujícího bodu trajektorie.

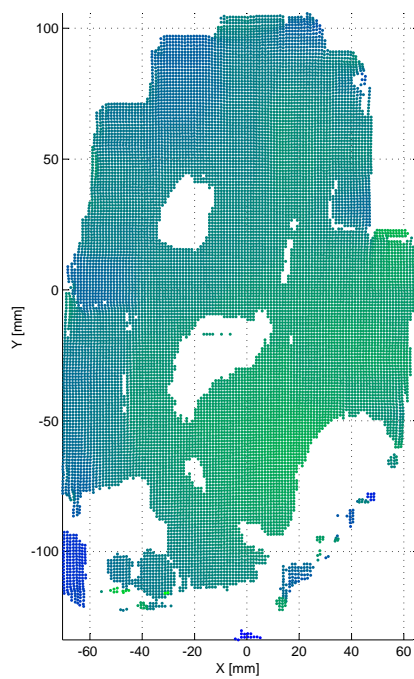


Obrázek 9.3: Levá část obsahuje mapu disparit a pravá binární masku odpovídající snímku na Obr. 9.2.

Pro vizualizaci rekonstruovaných bodů snímku z Obr. 9.2 jsem vytvořil makro v MATLABu, výsledky lze vidět na Obr. 9.4. Rekonstruované body kopírují povrch dlaně a prstů, avšak některé pixely byly rekonstruovány chybně. Body blíže ke kameře jsou obarveny zeleně a body dále od kamery modře.

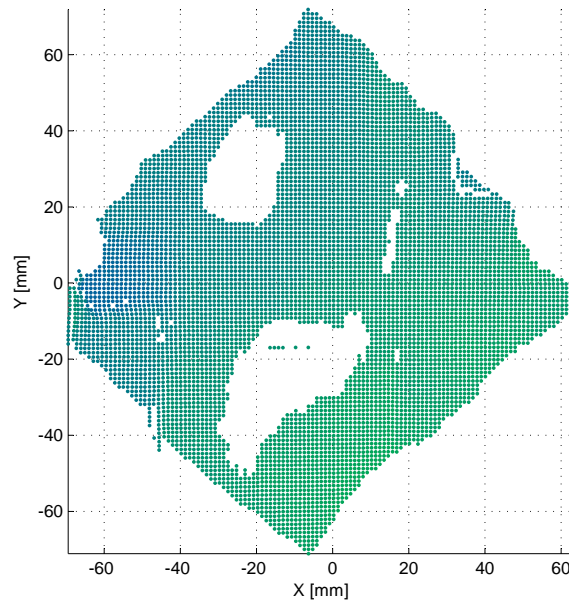


Obrázek 9.4: Izometrický pohled na 3D rekonstruované body.



Obrázek 9.5: Přední pohled na 3D rekonstruované body.

Po aplikování výše popsaného postupu jsou brány v potaz, pro výpočet reprezentujícího 3D bodu, pouze body na Obr. 9.6.



Obrázek 9.6: Přední pohled, body pro výpočet reprezentujícího 3D bodu trajektorie.

Tabulka 9.1 obsahuje souřadnice jednotlivých centroidů. **A** označuje centroid vypočítaný z původních 3D bodů a **B** centroid z oříznutých 3D bodů.

	A	B
x [mm]	-5.9560	-4.8158
y [mm]	-3.1293	0.4714
z [m]	-1.0832	-1.0804

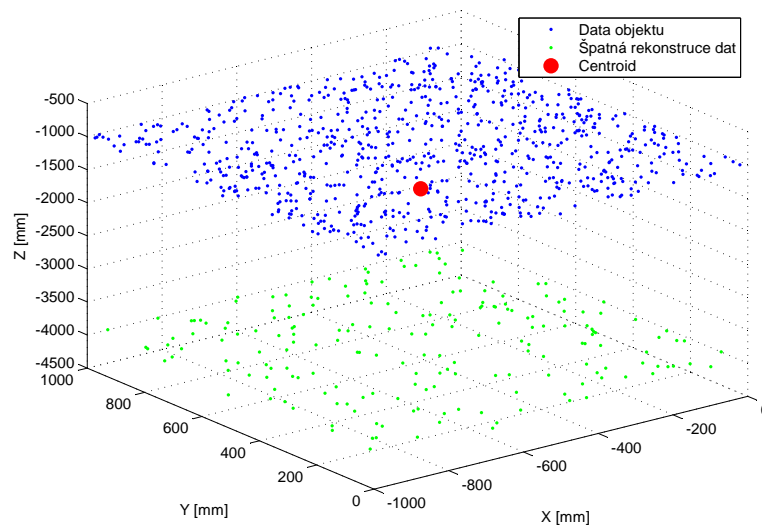
Tabulka 9.1: Souřadnice centroidů.

Výše popsaný postup je velmi výhodný při situacích, kdy několik bodů je rekonstruováno s velkou odchylkou od zbylých. V těchto případech lze velmi pozitivně ovlivnit polohu výsledného reprezentujícího 3D bodu trajektorie, což je prezentováno na následujícím modelovém příkladě.

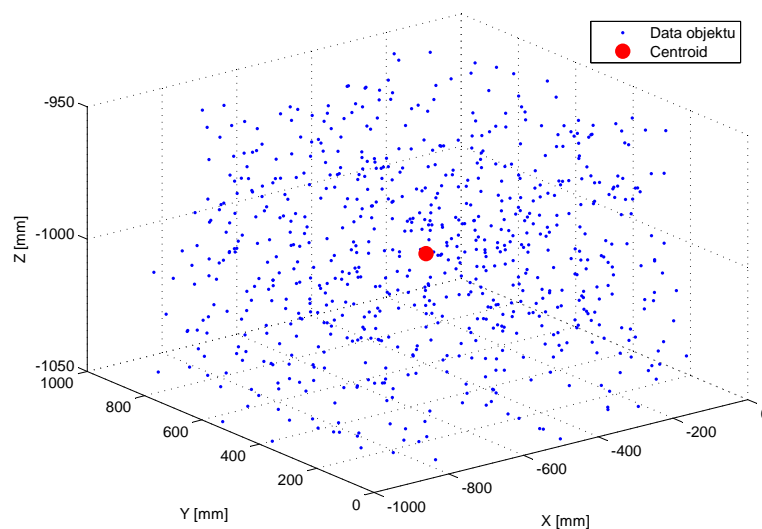
Zvolil jsem synteticky vygenerovaná data viz Obr. 9.7 (modře správně rekonstruovaná 3D data a zeleně špatně rekonstruovaná 3D data), na která jsem použil výše popsaný postup a oříznutá data lze vidět na Obr. 9.8. Bylo zvoleno 80 % bodů jako objekt a zbylé představují špatně rekonstruované body. Souřadnice centroidů přehledně shrnuje tabulka 9.2, kde označení odpovídá předchozímu případu.

	A	B
x [m]	0.4886	0.4841
y [m]	0.5042	0.4916
z [m]	-1.6002	-1.0003

Tabulka 9.2: Souřadnice centroidů pro modelový příklad.



Obrázek 9.7: Data objektu společně s chybně rekonstruovanými body.



Obrázek 9.8: Oříznutá data objektu.

Bylo ukázáno, že pomocí představeného postupu lze opravit polohu centroidu z dat, ve kterých byla 20% část rekonstruována s velkou odchylkou.

9.2 Vizualizace 3D trajektorií

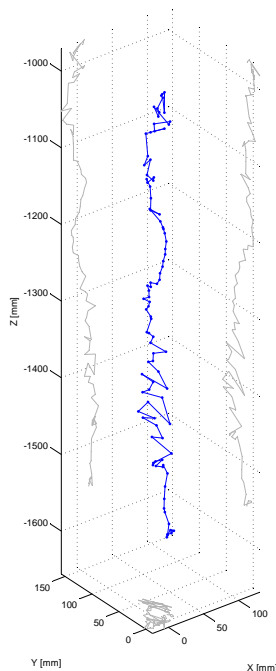
V této podkapitole budou vykresleny jednoduché příklady výsledných trajektorií pohybů a odpovídající slovní popis pohybu. Jako sledovaný objekt byla zvolena barevná rukavice, stejná jako na Obr. 9.1 a 9.2.

Vzhledem k velikosti plochy rukavice a 3D rekonstrukci reprezentujícího bodu v jednotlivých snímcích lze říci, že „chybné body trajektorie“ s odchylkou v řádech několika centimetrů (2 – 4 cm) od zbytku bodů trajektorie jsou vzhledem k velikosti dlaně ruky (10 × 20 cm) malé.

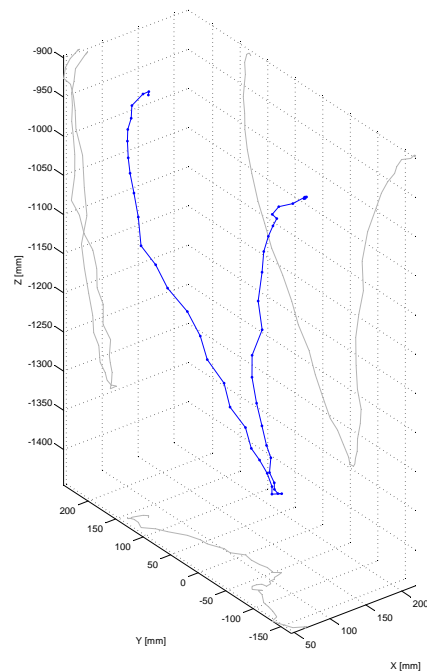
Na následujících příkladech jsou vždy modře obarveny 3D data trajektorie pohybu a šedě jednotlivé průměty do rovin xz , yz a xy .

Příklad 1: představuje postupné předpažení, tedy trajektorii pohybu podél z -ové osy, viz Obr. 9.9.

Příklad 2: trajektorie pohybu z předpažení dále k ramenu a do předpažení směrem k zemi, viz Obr. 9.10. Pohyb odpovídá rovině yz . Jedná se o rychlejší pohyb, což lze usuzovat ze vzdáleností mezi jednotlivými body.

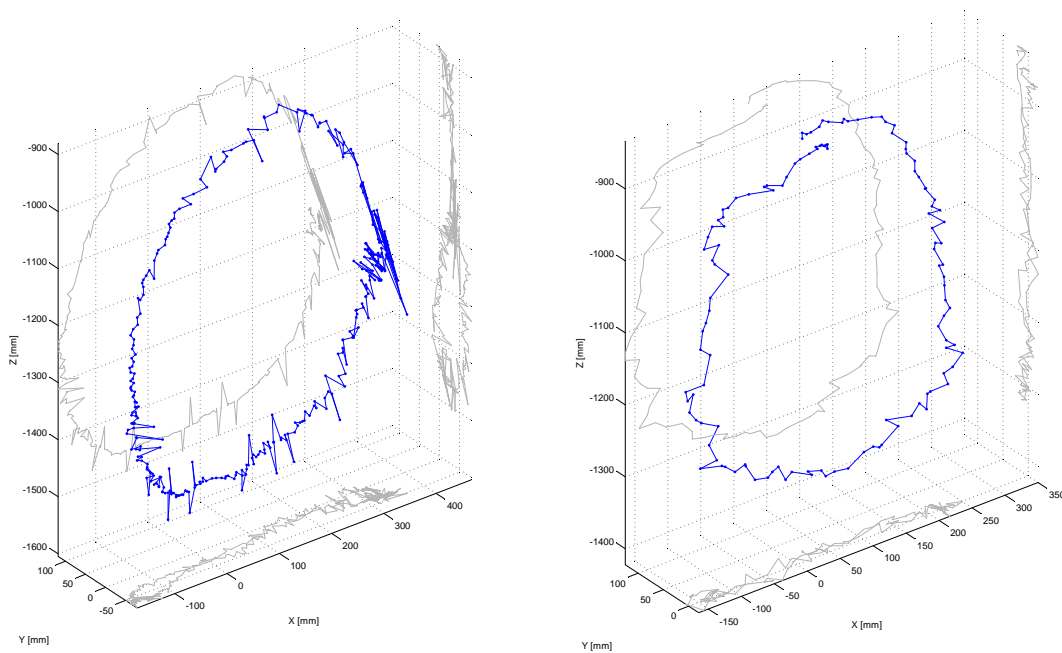


Obrázek 9.9: Trajektorie pohybu ve směru osy z .



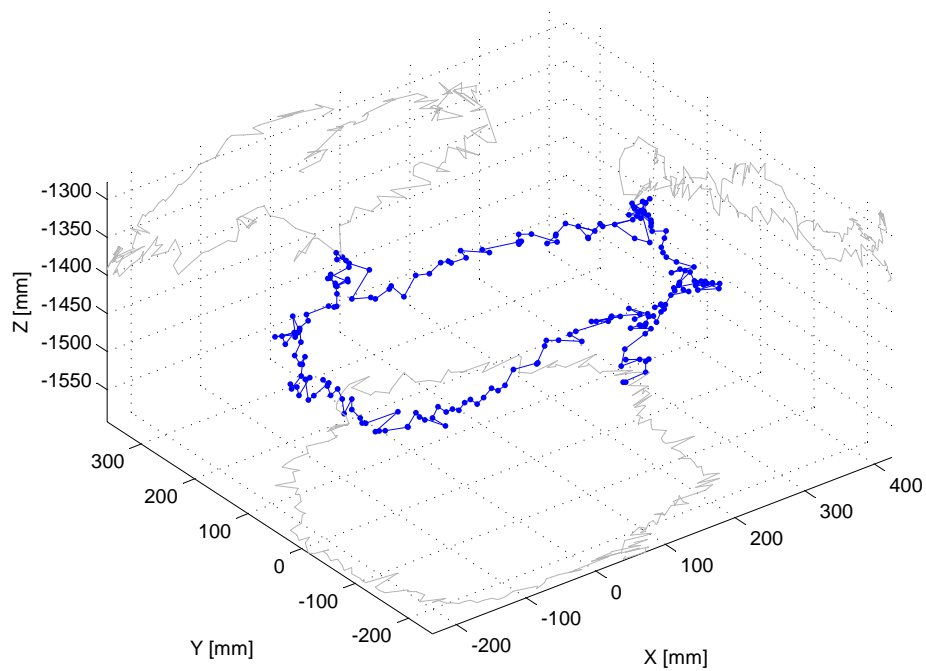
Obrázek 9.10: Trajektorie písmene V v rovině yz .

Příklad 3: na Obr. 9.11 jsou zobrazeny trajektorie kruhových pohybů v rovině xz .



Obrázek 9.11: Trajektorie kruhových pohybů v rovině xz .

Příklad 4: trajektorii kruhového pohybu v rovině xy lze vidět na Obr. 9.12.



Obrázek 9.12: Trajektorie kruhového pohybu v rovině xy .

Kapitola 10

Další možná rozšíření a vylepšení

V kapitole 8 byl popsán postup pro trénování jednoho barevného objektu na základě pohybu. Vytvořený program může být uzpůsoben pro trénování barevných modelů více objektů. Jednou z možností je trénovat každý objekt zvlášť do samostatné LUT, nebo natrénovat několik objektů do stejné LUT. Dále by mohl být upraven program pro sledování a 3D rekonstrukci trajektorií (popsaný v kapitole 9) na verzi, ve které by bylo možné sledovat a rekonstruovat více objektů současně. Což by mohlo být využito například k rekonstrukci 3D trajektorie obou rukou znakovajícího člověka.

Postup implementace stereo kalibrace je popsán v kapitole 7. Tento postup je z jisté míry závislý na kvalitě kamer, proto využití profesionálních kamer o vyšším rozlišení, s lepší optikou a pevnou konstrukcí upevnění kamer v prostoru by výsledky této úlohy jistě zlepšilo. Stereo kalibrace by byla vypočítána přesněji a následně by byla vylepšena i 3D rekonstrukce objektů.

Dále pro zlepšení návaznosti jednotlivých rekonstruovaných 3D bodů v trajektoriích, by bylo vhodné použít objekt, který bude malý, tedy co nejvíce se blížit reprezentujícímu bodu. Například by mohl být použit dostatečně malý/velký míček upevněný na tyči pro lepší manipulaci.

Vzhledem z podstaty použité metody **StereoBM** z knihovny **OpenCV**, která je určena pro vyhledávání navzájem korespondujících bodů v rektifikovaném páru snímků, lze usuzovat, že při použití nejednotité textury povrchu objektu dojde ke zvýšení počtu rekonstruovaných 3D bodů. Takto lze dosáhnout zisku větší informace o poloze objektu a jeho členitosti. Například pokud bude uvažována opět rukavice jako sledovaný objekt, tak změna textury na jemnou mřížku různých odstínů jedné barvy by pozitivně ovlivnila výslednou 3D rekonstrukci bodů. Tato myšlenka je spojena s použitím kamer odpovídající kvality.

Kapitola 11

Závěr a zhodnocení výsledků

V kapitole 1 byla obecně popsána problematika řešená v této diplomové práci a stručně představen současný stav řešeného problému společně s odkazy na související literaturu.

Kapitoly 2-6 popisují teoretickou část práce. Konkrétně v kapitole 2 jsou nastíněny základní myšlenky počítačového vidění a 3D geometrie. Následně v kapitole 3 byl popsán základní model dírkové kamery pomocí projekční matice obsahující vnitřní a vnější kalibrační parametry kamery. V kapitole 4 byl představen postup kalibrace jedné kamery ze známé scény s pomocí kalibračního vzoru. Dále se tato kapitola zabývá stereo viděním, konkrétně epipolární geometrií, výpočtem fundamentální matice pomocí osmibodového algoritmu a jeho vylepšením. Následuje popis rektifikace kamer a postup výpočtu 3D rekonstrukce bodů. Kapitola 5 se zaměřuje na popis základních přístupů metody odečítání pozadí a zmiňuje některé ze základních barevných prostorů. Závěrečná teoretická 6. kapitola obsahuje přehled metod pro sledování objektů v počítačovém vidění.

Cílem této diplomové práce bylo navrhnout a implementovat postupy a metody pro 3D rekonstrukci trajektorie pohybu barevného objektu v prostoru. Tento problém jsem rozdělil na několik dílčích. Nejprve bylo nutné implementovat systém pro stereo kalibraci a rektifikaci. Dále bylo nutné navrhnout a implementovat systém pro automatické trénování barevného modelu objektu. Závěrem využít získaná data pro sledování barevného objektu v obrázku a rekonstrukci jeho 3D trajektorie v reálném měřítku.

V kapitole 7 jsem implementoval systém pro stereo kalibraci. Využívá známého kalibračního obrazce ve formě černobílé šachovnice, která může být libovolné velikosti. Tento systém je velmi jednoduchý na použití a podává velmi dobré výsledky. Ty byly ověřeny formou vizualizace rektifikace dvojice obrázků, kde si jednotlivé řádky obrázků odpovídají, tedy představují epipolární přímky. Výstupem systému stereo kalibrace jsou konfigurační parametry kamer, ty jsou využity pro závěrečnou 3D rekonstrukci trajektorie.

Následující praktická kapitola 8 se zabývá automatickým trénováním barevného modelu objektu. Využil jsem metody odečítání pozadí. K reprezentaci barevného mo-

delu objektu jsem využil LUT tvaru krychle o délce hrany 256-ti hodnot. Výsledný model barevného objektu byl získán za pomoci modelování pozadí obrazové scény formou LUT a pohybem barevného objektu v popředí obrazové scény. Takto natrénované modely objektu by podávaly dobré výsledky pouze v případě aplikování na stejné snímky, ze kterých byl barevný model objektu vytvořen. Proto bylo nutné aplikovat vylepšení ve formě dvou iterací tří dimenzionální morfologické operace dilatace. Vzniklé modely byly robustnější a velmi dobře reprezentovaly barevný objekt i v polohách, ze kterých nebyl model trénován. Teoretická hranice přírůstku barevné informace po použití zmíněné dilatace v LUT modelu je až 125-ti násobkem počtu původní barevné informace. V reálném případě se jednalo o přibližně 18.6-ti násobek. Tento velký nárůst barevné informace, v LUT barevného modelu, velmi pozitivně ovlivňuje výslednou segmentaci barevného objektu. Dále jsem v této 8. kapitole prezentoval experiment spočívající ve výběru reprezentujícího barevného prostoru. Tento experiment ukázal, že v daném případě si byly zvolené barevné prostory velmi podobné, proto mezi nimi vznikly jen nepatrné rozdíly. V případě nejmenšího součtu velikosti absolutní chyby, přes všech 436 snímků uvažovaných v experimentu, odpovídající 33849 pixelům v barevném modelu $YCbCr$, byla velikost průměrné absolutní chyby na jeden snímek rovna přibližně 78 pixelům. Což odpovídá 0.0253% celkové plochy jednoho snímku o rozlišení 640×480 obrazových bodů. V případě největšího součtu velikosti absolutní chyby odpovídající barevnému modelu RGB, kdy byla hodnota rovna 51513 pixelů, průměrná absolutní chyba na jeden snímek tvořila pouze 0.0385% (118 pixelů) celkové plochy jednoho snímku o stejném rozlišení. Protože se rozdíl mezi nejlepším a nejhorším barevným modelem pohyboval v řádech setin procenta zvolil jsem RGB barevný prostor, jelikož se jedná o nativní reprezentaci snímku v OpenCV knihovně a nebylo proto třeba konvertovat snímky, což výrazně ušetřilo výpočetní čas. Výsledné dilatované LUT byly uloženy pro následné použití.

V kapitole 9 jsem prezentoval systém sledování a 3D rekonstrukce, který využívá výstupů ze systému stereo kalibrace a systému odečítání pozadí. Nejprve jsem popsal implementaci samotného systému a navržené jednoduché metody pro sledování objektu. Dále jsem vizualizoval možnou 3D rekonstrukci barevného objektu a popsal způsob výběru reprezentujícího 3D bodu pro daný snímek. Následně jsem popsal vylepšení, které umožňuje odstranění bodů jejichž z -ová souřadnice byla rekonstruována s velkou odchylkou od objektu. Toto vylepšení redukuje počet bodů, ze kterých se vypočítává reprezentující centroid. Výhodou je, že se uvažují pouze body, které daný objekt skutečně tvoří, nebo jsou mu velmi blízko. Tato metoda velmi pozitivně ovlivňuje výpočet reprezentujícího 3D bodu. Na modelovém příkladě bylo ukázáno, že použitý postup funguje i v případě, kdy bylo 20% bodů objektu rekonstruováno s velkou odchylkou od skutečné 3D polohy objektu. Závěrem kapitoly byly vizualizovány příklady možných trajektorií pomalých i dynamičtějších pohybů barevného objektu.

Kapitola 10 popisuje a navrhuje další možná vylepšení implementovaných systémů jako celku.

Všechny popsané problémy v úvodní 1. kapitole jsem úspěšně vyřešil, otestoval a zhodnotil. Výsledné 3D rekonstruované trajektorie odpovídají reálné metrice. Přesnost rekonstrukce je velmi dobrá a odpovídá zvoleným kamerám. Výsledné 3D trajektorie by po určité úpravě a vyhlazení, popřípadě proložení křivkami mohly být využity jako vstupy pro manipulátor, lanového robota, dále sloužit jako vizualizace trajektorie lidských rukou nebo pro měření vzdálenosti objektů a jejich pohybu.

Výsledný systém 3D rekonstrukce trajektorie barevného objektu není dokonalý z hlediska výstupů, avšak může tvořit dobrý základ pro následné rozšíření ve formě komplexnějšího a sofistikovanějšího systému. Například s využitím hloubkové informace z Kinectu by mohla být výsledná rekonstruovaná 3D trajektorie hladší a souvislejší. Tento postup nebyl testován. Vzhledem k rozsáhlé teorii 3D rekonstrukce scény pomocí dvou kamer se domnívám, že tento přístup měl větší naučnou hodnotu než v případě přímého využití hloubkového snímače.

Příloha A

Uživatelská příručka k systému stereo kalibrace

Spuštění: program s názvem `stereo_calibration.exe` je spustitelný dvojklikem, kde při této možnosti jsou jako vstupy použity kamery. Dále lze program spustit pomocí příkazového řádku, například

```
C:\stereo_calibration.exe
```

a následně potvrdit klávesou **Enter**. Program může jako vstupy použít předem nahrané videa, cesta k jejich umístění se zadá do volitelných parametrů programu, například následovně (pokud se program a videa nacházejí v kořenovém adresáři)

```
C:\stereo_calibration.exe C:\video_01.avi C:\video_02.avi
```

a následně potvrdit klávesou **Enter**.

Práce s programem: po spuštění program postupně vyžaduje celočíselné hodnoty parametrů počtu řádků rohů v kalibračním obrazci, počet sloupců rohů v kalibračním obrazci, vzdálenost v milimetrech mezi jednotlivými vnitřními rohy kalibračního obrazce a počet kalibračních párů snímků, ze kterých bude provedena stereo kalibrace. Program může pracovat ve dvou režimech. Oba se aktivují kliknutím na libovolné okno a stisknutím odpovídající klávesy. První možnost je využít automatickou kalibraci, která snímá dvojice obrázků v pevně daném intervalu dokud není dosaženo počtu snímků zvoleného uživatelem jako jeden ze čtyř parametrů na začátku. Tento režim se zpřístupní pomocí stisku klávesy **s**. Druhá možnost spočívá v manuální kalibraci, která se spustí po stisku klávesy **n**. Pro snímání dvojic obrázků je vyžadována interakce od uživatele ve formě opětovného stisku klávesy **n**, dokud nejsou veškeré snímky uloženy. Následně je vizualizována vypočtená kalibrace formou aktuální rektifikované dvojice snímků. Stereo kalibraci lze opakovat, dokud se nedosáhne dobrých výsledků. Lze použít různé režimy kalibrace, dokonce je možno začít kalibraci pomocí manuálního režimu a dokončit jí pomocí automatického režimu.

Ukončení: program lze ukončit stiskem klávesy **Esc**.

Příloha B

Uživatelská příručka k systému odečítání pozadí

Spuštění: program s názvem `background_subtraction.exe` je spustitelný dvojklikem, kde při této možnosti jsou jako vstupy použity kamery. Dále lze program spustit pomocí příkazového řádku, například

```
C:\background_subtraction.exe
```

a následně potvrdit klávesou `Enter`. Program může jako vstupy použít předem nahrané videa, cesta k jejich umístění se zadá do volitelných parametrů programu, například následovně (pokud se program a videa nacházejí v kořenovém adresáři)

```
C:\background_subtraction.exe C:\video_01.avi C:\video_02.avi
```

a následně potvrdit klávesou `Enter`. V případě předem nahraných videí je možno zadat počet snímků, které budou natrénovány do pozadí a program spustit následovně

```
C:\background_subtraction.exe XX C:\video_01.avi C:\video_02.avi
```

kde `XX` značí počet prvotních snímků, které budou natrénovány do modelu pozadí. Následně potvrdit klávesou `Enter`.

Práce s programem: po spuštění programu je nutné, aby se během prvních 200 (nebo zadaném počtu v případě použití videí v souborech) snímků v prostoru před kamerami pohyboval uživatel co nejvíce a bez barevného objektu, který chce natrénovat do LUT. Následně by měl uživatel pohybovat před kamerami s barevným objektem, ideálně v co nejrozmanitějších polohách objektu, aby byl vytvořen co nejvíce reprezentativní barevný model objektu. Ve fázi trénování barevného objektu jsou vizualizovány pixely, které jsou natrénovány do LUT barevného objektu pro každou kameru. Pro ukončení trénování je nutné kliknout na libovolné okno a stisknout klávesu `e`. Následně je vizualizována segmentace pomocí natrénovaných LUT pro jednotlivé kamery.

PŘÍLOHA B. UŽIVATELSKÁ PŘÍRUČKA K SYSTÉMU ODEČÍTÁNÍ POZADÍ

Pokud je výsledná segmentace provedena špatně, tedy barevný model ve formě LUT byl nedostatečně natrénován je nutné program spustit znovu a proces trénování opakovat.

Ukončení: program lze ukončit stiskem klávesy Esc.

Příloha C

Uživatelská příručka k systému sledování a 3D rekonstrukce

Spuštění: program s názvem `3D_reconstruction.exe` je spustitelný dvojklikem, kde při této možnosti jsou jako vstupy použity kamery. Dále lze program spustit pomocí příkazového řádku, například

```
C:\3D_reconstruction.exe
```

a následně potvrdit klávesou `Enter`. Program může jako vstupy použít předem nahrané videa, cesta k jejich umístění se zadá do volitelných parametrů programu, například následovně (pokud se program a videa nacházejí v kořenovém adresáři)

```
C:\3D_reconstruction.exe C:\video_01.avi C:\video_02.avi
```

a následně potvrdit klávesou `Enter`. Pro úspěšné spuštění programu musí být ve stejném adresáři i stereo kalibrační data a LUT vytvořené dříve popsányými programy.

Práce s programem: po spuštění je v okně zobrazen náhled aktuálního snímku z jedné kamery, dále segmentace pomocí LUT a odpovídající mapa disparit. Před začátkem 3D rekonstrukce je nutné nastavit mapu disparit pomocí níže popsáných parametrů, aby výsledná rekonstrukce dávala co nejlepší výsledky. Cílem je mít vizualizovaný objekt v okně s mapou disparit ve všech polohách pohybu, který chce uživatel rekonstruovat. Jedná se tedy o nastavení parametrů pro mapu disparit, které ovlivňují hloubku pracovního prostoru, kde bude probíhat následně rekonstrukce.

V následujících bodech bude znak (+) značit zvyšování hodnoty a znak (-) snižování, což je důležité hlavně pro nastavení parametrů metody pro hledání odpovídajících pixelů (StereoBM) ve dvojici snímků.

- Klávesou `w` lze vypsat na konzoli aktuální 3D souřadnice centroidu objektu pro ověření správnosti metriky.

PŘÍLOHA C. UŽIVATELSKÁ PŘÍRUČKA K SYSTÉMU SLEDOVÁNÍ A 3D REKONSTRUKCE

- Pomocí stisku klávesy **s** se začne ukládat rekonstruovaná trajektorie do souboru `xyzt.txt`, kde jednotlivé písmena reprezentují formát uložení dat. V souboru jsou uloženy data na řádku v pořadí `x`, `y`, `z` (v milimetrech) a `t` (v milisekundách), kde `t` reprezentuje časovou diferenci mezi jednotlivými snímky. Pro ukončení nahrávání trajektorie je nutné opět stisknout klávesu **s**. Výše zmíněný soubor je při opětovném spuštění ukládání trajektorie přemazán.
- Změna (+) parametru `preFilterCap` pomocí stisku klávesy **e**.
- Změna (-) parametru `preFilterCap` pomocí stisku klávesy **d**.
- Změna (+) parametru `preFilterSize` pomocí stisku klávesy **r**.
- Změna (-) parametru `preFilterSize` pomocí stisku klávesy **f**.
- Změna (+) parametru `SADWindowSize` pomocí stisku klávesy **t**.
- Změna (-) parametru `SADWindowSize` pomocí stisku klávesy **g**.
- Změna (+) parametru `numberOfDisparities` pomocí stisku klávesy **z**.
- Změna (-) parametru `numberOfDisparities` pomocí stisku klávesy **h**.
- Změna (+) parametru `minDisparity` pomocí stisku klávesy **u**.
- Změna (-) parametru `minDisparity` pomocí stisku klávesy **j**.
- Změna (+) parametru `textureThreshold` pomocí stisku klávesy **i**.
- Změna (-) parametru `textureThreshold` pomocí stisku klávesy **k**.
- Změna (+) parametru `uniquenessRatio` pomocí stisku klávesy **o**.
- Změna (-) parametru `uniquenessRatio` pomocí stisku klávesy **l**.

Ukončení: program lze ukončit stiskem klávesy **Esc**.

Literatura

- [1] L. Shapiro and G. Stockman, *Computer vision*. Prentice Hall, 2001.
- [2] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*. Nelson Education Limited, 2008.
- [3] Z. Zhang, “A flexible new technique for camera calibration,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 11, pp. 1330–1334, 2000.
- [4] T. Svoboda, D. Martinec, and T. Pajdla, “A convenient multi-camera self-calibration for virtual environments,” *PRESENCE: Teleoperators and Virtual Environments*, vol. 14, no. 4, pp. 407–422, August 2005.
- [5] Y. Benezeth, P.-M. Jodoin, B. Emile, H. Laurent, and C. Rosenberger, “Review and evaluation of commonly-implemented background subtraction algorithms,” in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, 2008, pp. 1–4.
- [6] M. Piccardi, “Background subtraction techniques: a review,” in *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, vol. 4, 2004, pp. 3099–3104 vol.4.
- [7] H. Wang and D. Suter, “A novel robust statistical method for background initialization and visual surveillance,” in *Computer Vision – ACCV 2006*, ser. Lecture Notes in Computer Science, P. Narayanan, S. Nayar, and H.-Y. Shum, Eds. Springer Berlin Heidelberg, 2006, vol. 3851, pp. 328–337.
- [8] Y.-f. Zhu, “Background subtraction and color clustering based moving objects detection,” in *Information Engineering and Computer Science, 2009. ICIECS 2009. International Conference on*, 2009, pp. 1–5.
- [9] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision (ijcai),” in *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81)*, April 1981, pp. 674–679.
- [10] B. D. Lucas, “Generalized image matching by the method of differences,” Ph.D. dissertation, Robotics Institute, Carnegie Mellon University, July 1984.

- [11] K. Fukunaga and L. Hostetler, “The estimation of the gradient of a density function, with applications in pattern recognition,” *Information Theory, IEEE Transactions on*, vol. 21, no. 1, pp. 32–40, 1975.
- [12] M. Isard and A. Blake, “Condensation - conditional density propagation for visual tracking,” *International Journal of Computer Vision*, vol. 29, pp. 5–28, 1998.
- [13] S. Shen, “Accurate multiple view 3d reconstruction using patch-based stereo for large-scale scenes,” *Image Processing, IEEE Transactions on*, vol. 22, no. 5, pp. 1901–1914, 2013.
- [14] D. Kim, D. Min, and K. Sohn, “3d scene reconstruction system with hand-held stereo cameras,” in *3DTV Conference, 2007*, 2007, pp. 1–4.
- [15] A. Hobson, *The Oxford Dictionary of Difficult Words*. Oxford University Press, 2004.
- [16] R. Hartley, “In defence of the 8-point algorithm,” in *Computer Vision, 1995. Proceedings., Fifth International Conference on*, 1995, pp. 1064–1070.
- [17] C. Loop and Z. Zhang, “Computing rectifying homographies for stereo vision,” in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, vol. 1, 1999, pp. –131 Vol. 1.
- [18] M. Matousek, R. Sara, and V. Hlavac, “Data-optimal rectification for fast and accurate stereovision,” in *Multi-Agent Security and Survivability, 2004 IEEE First Symposium on*, 2004, pp. 212–215.
- [19] R. I. Hartley, “Theory and practice of projective rectification,” *Int. J. Comput. Vision*, vol. 35, no. 2, pp. 115–127, Nov. 1999.
- [20] G. Bradski, “Real time face and object tracking as a component of a perceptual user interface,” in *Applications of Computer Vision, 1998. WACV '98. Proceedings., Fourth IEEE Workshop on*, 1998, pp. 214–219.
- [21] Y. Bar-Shalom and T. Fortmann, *Tracking and Data Association*, ser. Mathematics in Science and Engineering Series. Academic Press, 1988.
- [22] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*. O’Reilly Media, 2008.