



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY**

**A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

**ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY**

DEPARTMENT OF CONTROL AND INSTRUMENTATION

**AUTOMATICKÝ VÝPOČET RELATIVNÍ ORIENTACE DVOU  
KAMER**

AUTOMATIC CALCULATION OF RELATIVE ORIENTATION OF TWO CAMERAS

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Daniel Adámek**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. Jan Klečka**

**BRNO 2016**



# Bakalářská práce

bakalářský studijní obor **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

**Student:** Daniel Adámek

**ID:** 159199

**Ročník:** 3

**Akademický rok:** 2015/16

**NÁZEV TÉMATU:**

## Automatický výpočet relativní orientace dvou kamer

### POKyny PRO VYPRACOVÁNÍ:

Cílem práce je vyvinout a implementovat algoritmus pro výpočet relativní orientace dvou kamer v prostoru z korespondencí nalezených v předem nedefinované scéně.

1. Seznamte se s problematikou epipolární geometrie a automatické detekce korespondenčních bodů ve dvojici snímků.
2. Experimentálně ověřte na umělých datech nastudovanou teorii v oblasti epipolární geometrie.
3. Navrhněte a popište vlastní algoritmus pro výpočet relativní orientace dvou kamer.
4. Implementujte vámi zvolený algoritmus v jazyce C++.
5. Na sérii experimentů vyhodnoťte funkčnost a přesnost Vámi vytvořeného řešení.

### DOPORUČENÁ LITERATURA:

[1] SONKA M., HLAVAC V., BOYLE R.: Image Processing, Analysis and Machine Vision. 3rd edition.

Toronto : Thomson, 2008. 829 s. ISBN 978-0-495-08252-1.

[2] RUSS, J.C. The Image Processing Handbook. Boca Raton : CRC Press, 1995. 674 p. ISBN

0-8493-2516-1.

**Termín zadání:** 8.2.2016

**Termín odevzdání:** 23.5.2016

**Vedoucí práce:** Ing. Jan Klečka

**Konzultant bakalářské práce:**

**doc. Ing. Václav Jirsík, CSc., předseda oborové rady**

### UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## ABSTRAKT

Při určování relativní orientace dvou kamer z dvojice snímků je potřeba porozumět širokému spektru oborů. Proto byl v této práci nastíněn princip metod detekujících a popisujících významné body v obraze a jejich následné párování, čtenáři byl také poskytnut vhled do problematiky epipolární geometrie a lineární triangulace a byla navržena implementace představené teorie v jazyce C++, jejíž testy odhalily v závěru práce slabiny nastíněného přístupu.

## KLÍČOVÁ SLOVA

Počítačové vidění, Epipolární geometrie, SIFT, SURF, BRIEF, FAST, Fundamentální matice, Esenciální matice, Lineární triangulace, openCV, C++

## ABSTRACT

To determine a relative orientation of two cameras, it is necessary to understand wide range of fields. Therefore, this thesis shows principles of basic methods of detection and description of keypoints in an image as well as methods of matching these keypoints. The paper also provides brief summary of epipolar geometry and linear triangulation. C++ implementation of presented theory was proposed and its disadvantages were discussed.

## KEYWORDS

Computer Vision, Epipolar geometry, SIFT, SURF, BRIEF, FAST, Fundamental matrix, Essential matrix, Linear triangulation, openCV, C++

ADÁMEK, Daniel *Automatický výpočet relativní orientace dvou kamer*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2016. 51 s. Vedoucí práce byl Ing. Jan Klečka

## PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Automatický výpočet relativní orientace dvou kamer“ jsem vypracoval(a) samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor(ka) uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno .....

.....

podpis autora(-ky)

## PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu mé bakalářské práce, panu Ing. Janu Klečkovi, za odborné vedení, vstřícný přístup, trpělivost a čas věnovaný konzultacím.

Brno .....

.....

podpis autora(-ky)

# OBSAH

<b>Úvod</b>	<b>10</b>
<b>1 Epipolární geometrie</b>	<b>11</b>
1.1 Úvod a základní pojmy . . . . .	11
1.2 Fundamentální matice . . . . .	12
1.3 Esenciální matice . . . . .	13
1.4 Výpočet fundamenální matice . . . . .	13
1.5 Extrakce projekčních matic z esenciální matice . . . . .	15
1.6 Lineární triangulace . . . . .	15
<b>2 Základní metody detekce a popisu významných bodů</b>	<b>18</b>
2.1 Harrisův detektor . . . . .	18
2.2 Shi-Tomasi modifikace Harrisova detektoru . . . . .	19
2.3 Scale Invariant Feature Transform . . . . .	20
2.3.1 Detekce extrémů . . . . .	20
2.3.2 Lokalizace významných bodů . . . . .	21
2.3.3 Přidělení orientace . . . . .	22
2.3.4 Deskriptor . . . . .	22
2.4 Speed-Up Robust Features . . . . .	23
2.4.1 Detekce významných bodů . . . . .	23
2.4.2 Deskriptor . . . . .	24
2.5 Features from Accelerated Segment Test . . . . .	25
2.5.1 Strojové učení . . . . .	26
2.5.2 Non-Maximal Suppression . . . . .	26
2.6 Binary Robust Independent Elementary Features . . . . .	27
2.7 Srovnání jednotlivých metod . . . . .	27
2.8 Párování . . . . .	28
2.8.1 Metoda hrubé síly . . . . .	29
2.8.2 Vzdálenosti . . . . .	29
<b>3 Implementace</b>	<b>30</b>
3.1 Experimentální ověření epipolární gemoetrie na umělých datech . . .	30
3.2 C kód . . . . .	32
3.2.1 Načtení obrazů . . . . .	32
3.2.2 Detekce významných bodů a hledání korespondencí . . . . .	33
3.2.3 Výpočet esenciální matice a nalezení všech možných řešení projekční matice $P'$ . . . . .	33

3.2.4	Lineární triangulace a vyšetření hloubky bodů . . . . .	34
3.2.5	Extrakce rotace . . . . .	35
<b>4</b>	<b>Experimentální ověření funkčnosti</b>	<b>37</b>
4.1	Referenční testy . . . . .	37
4.2	Testování algoritmu . . . . .	39
4.2.1	Výsledky prvního testu . . . . .	40
4.2.2	Výsledky druhého testu . . . . .	41
4.2.3	Výsledky třetího testu . . . . .	42
4.2.4	Srovnání . . . . .	42
<b>5</b>	<b>Závěr</b>	<b>44</b>
	<b>Literatura</b>	<b>46</b>
	<b>Seznam symbolů, veličin a zkratk</b>	<b>47</b>
	<b>Seznam příloh</b>	<b>48</b>
<b>A</b>	<b>Testové snímky</b>	<b>49</b>

# SEZNAM OBRÁZKŮ

1.1	Projekční rovina v prostoru.[3]	12
1.2	Geometrická interpretace čtyř možných řešení extrakce projekční matice $\mathbf{P}'$ z esenciální matice $\mathbf{E}$ . [3]	16
2.1	Odezva operátoru $R$ pro různé $\alpha, \beta$ . [5]	19
2.2	Odezva rozhodovací funkce podle Shi-Tomasi. Pokud se obě vlastní čísla matice objevují v zeleném regionu, bod je považován za roh. [6]	20
2.3	26okolí kandidáta na významný bod brané z přilehlých zvětšení/zmenšení. [6]	21
2.4	Šedě podbarvená výseč značí $60^\circ$ , pro které je vypočtena určující orientace (znázorněna červenou šipkou). Poté se výseč přesune a výpočet se opakuje. [9]	24
2.5	Znázornění možných situací při porovnávání znamének Laplaciánu. Na šedém pozadí jsou porovnávány pouze světlejší, nebo tmavší obrazy. Při rozdílném znaménku nejsou připuštěny k dalšímu porovnání. [6]	25
2.6	Zvolené okolí bodu sloužící pro detekci rohů metodou FAST. [10]	25
3.1	Vývojový diagram testovacího .m skriptu	30
3.2	Vizualizace tvorby umělých korespondencí	31
3.3	Vývojový diagram volaných funkcí ve zdrojovém kódu aplikace.	32
3.4	Nalezené korespondence.	34
4.1	Ukázka páru vstupních snímků do Stereo Calibration Toolboxu.	37
4.2	Korespondence detekované v prvním páru testovaných snímků.	39
A.1	První test, snímek A.	49
A.2	První test, snímek B.	49
A.3	Druhý test, snímek A.	50
A.4	Druhý test, snímek B.	50
A.5	Třetí test, snímek A.	51
A.6	Třetí test, snímek B.	51



# SEZNAM TABULEK

2.1	Výsledky detekce bodů jednotlivých metod v [1]	28
2.2	Výsledky porovnání SIFT a SURF podle [2]	28
2.3	Počty spárovaných bodů pro jednotlivé metody v práci [1]	29
4.1	Počet použitých snímků pro jednotlivé testy	37
4.2	Referenční hodnoty rotací a translací.	38
4.3	Výsledky prvního měření	40
4.4	Absolutní chyby a směrodatná odchylka rotace pro první měření.	40
4.5	Normalizované translační vektory pro první měření.	40
4.6	Výsledky druhého měření	41
4.7	Absolutní chyby a směrodatná odchylka rotace pro druhé měření.	41
4.8	Normalizované translační vektory pro druhé měření.	41
4.9	Výsledky třetího měření	42
4.10	Absolutní chyby a směrodatná odchylka rotace pro třetí měření.	42
4.11	Normalizované translační vektory pro třetí měření.	42
4.12	Počet nalezených významných bodů, korespondencí a inlierů pro jednotlivé testy.	43
4.13	Srovnání směrodatných odchylek rotace všech testů a úspěšnosti vybrání správných inlierů.	43
4.14	Srovnání směrodatných odchylek translace všech testů a úspěšnosti vybrání správných inlierů.	43

# ÚVOD

Počítačové vidění je čím dál rychleji se rozvíjejícím odvětvím. Jedním poměrně mladým směrem v této oblasti je problematika rekonstrukce 3D scény z množiny 2D snímků. Řešení této úlohy má opodstatnění v mnoha oblastech průmyslu, například při mapování interiérů či exteriérů, nebo vytváření počítačových modelů reálných objektů. Takovýto úkol vyžaduje aplikaci značného matematického aparátu, zejména pak odvětví geometrie epipolární a projektivní. Při šetření neznámé polohy snímajících kamer musí postup nutně zahrnovat operaci hledání korespondujících bodů v množině snímků, popřípadě jejich popis. Proto se od začátku devadesátých let objevila řada metod, jak najít ve snímku body čímsi význačné, které by usnadnili následné párování skrz vícero snímků. Spolu s metodou popisu těchto bodů a párovacími algoritmy, novými i z dob dřívějších, vznikla řada postupů, jak tento problém vyřešit v myslitelném čase. S pomocí otevřené knihovny openCV v3.1 bude mým úkolem navrhnout v jazyce C++ aplikaci, která provede rekonstrukci relativní orientace dvou kamer z dvojice snímků. K implementaci bude poskytnuta potřebná teorie z oblasti epipolární a projektivní geometrie, lineární triangulace a budou popsány a diskutovány metody detekující významné body ve snímcích. V neposlední řadě bude aplikace testována a výsledky testů diskutovány.

# 1 EPIPOLÁRNÍ GEOMETRIE

Základním nástrojem pro splnění cíle této práce je epipolární geometrie. Jde o projektivní geometrii mezi dvěma snímky, která je nezávislá na rozestavení scény, závisí pouze na vnitřních parametrech kamer a jejich relativní orientaci. Tuto geometrii reprezentuje fundamentální matice  $\mathbf{F}$ , ze které se dá extrahovat relativní úhlový posun kamer. Úvod do epipolární geometrie, popis vlastností fundamentální matice a způsoby jejího stanovení budou přiblíženy v této kapitole.

## 1.1 Úvod a základní pojmy

Každý jednotlivý třídimenzionální bod  $\mathbf{X}$  ve snímané scéně je promítán na obrazové roviny obou kamer pomocí projekčních matic  $\mathbf{P}$  a  $\mathbf{P}'$ , platí

$$\mathbf{P}\mathbf{X} = \mathbf{x} \quad (1.1)$$

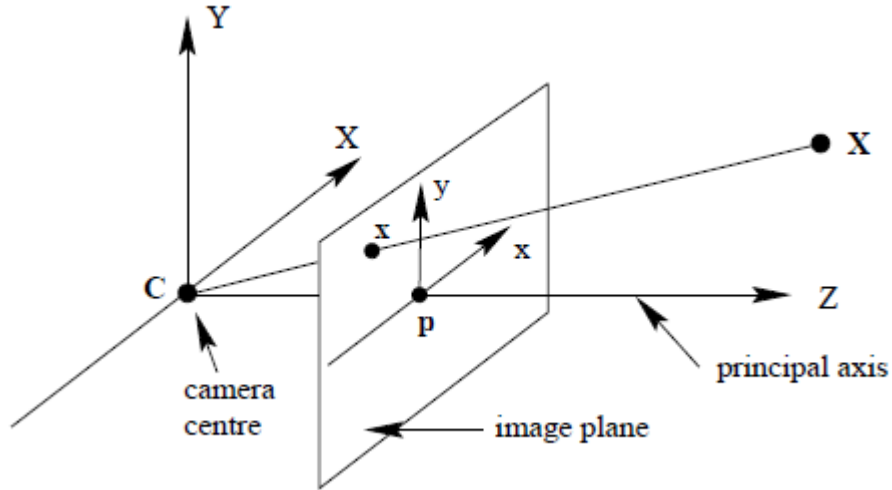
Projektivní matice  $\mathbf{P}$  převádí třídimenzionální bod v homogenních souřadnicích do dvourozměrného bodu. V případě, že souřadný systém projekční roviny je rovnoběžný se souřadným systémem prostoru a střed promítání  $\mathbf{C}$  leží v počátku (viz obr. 1.1), má matice  $\mathbf{P}$  tvar:

$$\mathbf{P} = \mathbf{K}[\mathbf{I}|\mathbf{0}] \quad (1.2)$$

kde  $\mathbf{K}$  je kalibrační matice. Pokud je střed promítání posunut o vektor  $\mathbf{t}$  a projekční rovina je otočená o rotační matici  $\mathbf{R}$ , má projekční matice tvar:

$$\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}] \quad (1.3)$$

V místě průsečíku obrazové roviny a přímky určené bodem  $\mathbf{X}$  a středem kamery  $\mathbf{C}$  (resp.  $\mathbf{C}'$ ) vzniká obraz bodu  $\mathbf{X} \rightarrow \mathbf{x}$  (resp.  $\mathbf{x}'$ ). Poté body  $\mathbf{X}$ ,  $\mathbf{C}$ ,  $\mathbf{C}'$ ,  $\mathbf{x}$  a  $\mathbf{x}'$  leží ve společné rovině  $\pi$ , neboli epipolární rovině. Přímka určená středy kamer  $\mathbf{C}$  a  $\mathbf{C}'$  se nazývá báze. V místě průniku obrazových rovin a báze vznikají epipóly  $\mathbf{e}$  a  $\mathbf{e}'$ . Potom průnik epipolární roviny a obrazové roviny dává vzniknout epipoláře  $\mathbf{l}$  (resp.  $\mathbf{l}'$ ), která vždy prochází epipólem a náležejícím obrazem  $\mathbf{x}$  (resp.  $\mathbf{x}'$ ). Z toho vyplývá, že pro každé  $\mathbf{x}$  lze najít v druhé obrazové rovině epipoláru  $\mathbf{l}'$ , na které se nachází korespondující bod  $\mathbf{x}'$ . Vztah mezi přechodem  $\mathbf{x} \rightarrow \mathbf{l}'$  popisuje fundamentální matice. Toho se využívá při hledání korespondencí v druhém snímku ze stereopáru. Nicméně v případě, že jsou korespondující páry bodů známy, lze z těchto korespondencí určit fundamentální matici  $\mathbf{F}$  a z té následně informace o relativní pozici kamer.



Obr. 1.1: Projekční rovina v prostoru.[3]

## 1.2 Fundamentální matice

Fundamentální matice je algebraickým vyjádřením epipolární geometrie. Je to matice velikosti  $3 \times 3$ , s hodnotí 2, jejíž determinant je roven nule. Jak bylo řečeno v předchozí kapitole, fundamentální matice má za úkol popsat přechod z obrazu bodu  $\mathbf{X}$  na první obrazové rovině k epipoláře na druhé obrazové rovině. Neboli

$$\mathbf{l}' = \mathbf{F}\mathbf{x}. \quad (1.4)$$

Zároveň platí vztah mezi oběma obrazy bodu  $\mathbf{X}$ :

$$\mathbf{x}^T \mathbf{F} \mathbf{x} = 0 \quad (1.5)$$

Přechod obrazu  $\mathbf{x}$  z jedné projekční roviny do druhé lze provést pomocí projekčních matic:

$$\mathbf{x}' = \mathbf{P}'\mathbf{P}^+\mathbf{x} \quad (1.6)$$

kde  $\mathbf{P}^+$  je pseudoinverzní matice. Ze známého vztahu pro epipoláru  $\mathbf{l} = \mathbf{e} \times \mathbf{x}$  lze odvodit:

$$\mathbf{l}' = \mathbf{e}' \times \mathbf{P}'\mathbf{P}^+\mathbf{x} \quad (1.7)$$

Epipól  $\mathbf{e}'$  lze vypočítat jako projekci středu první kamery na obrazovou rovinu druhé, neboli

$$\mathbf{e}' = \mathbf{P}'\mathbf{C} \quad (1.8)$$

závěrem stačí dosadit za epipoláru  $\mathbf{l}'$  výraz dle vztahu 1.7,

$$\mathbf{F}\mathbf{x} = \mathbf{P}'\mathbf{C} \times \mathbf{P}'\mathbf{P}^+\mathbf{x}, \quad (1.9)$$

fundamentální matici  $\mathbf{F}$  lze tedy vyjádřit jako

$$\mathbf{F} = \mathbf{P}'\mathbf{C} \times \mathbf{P}'\mathbf{P}^+ \quad (1.10)$$

Pokud první projekční matici pro zjednodušení zvolím  $\mathbf{P} = \mathbf{K}[\mathbf{I}|\mathbf{0}]$ , tedy střed promítání první kamery prohlásím za počátek souřadného systému prostoru a zároveň bude souřadný systém projekční roviny rovnoběžný s tím prostorovým, potom je druhá projekční matice rovna  $\mathbf{P}' = \mathbf{K}'[\mathbf{R}|\mathbf{t}]$ . Je vidět, že po dosazení do předchozího vzorce by bylo možné extrahovat matici rotace. Pro nalezení matice rotace je však vhodné použít jiné metody, a to rozklad esenciální matice  $\mathbf{E}$  na singulární hodnoty (SVD), což je speciální případ fundamentální matice.

### 1.3 Esenciální matice

Esenciální matice je v mnoha ohledech podobná té fundamentální. Hlavním rozdílem je, že počítá s normalizovanými souřadnicemi  $\hat{\mathbf{x}}$  a  $\hat{\mathbf{x}}'$ , pro které platí

$$\hat{\mathbf{x}} = \mathbf{K}^{-1}\mathbf{x} \quad (1.11)$$

čímž se odstraní efekt kalibrační matice na kameru (a body jí promítnutými). Platí pak podobný vztah jako pro fundamentální matici:

$$\hat{\mathbf{x}}'^T \mathbf{E} \hat{\mathbf{x}} = 0 \quad (1.12)$$

pokud dosadíme za body  $\hat{\mathbf{x}}$  a  $\hat{\mathbf{x}}'$ , dostaneme převodní vztah mezi  $\mathbf{E}$  a  $\mathbf{F}$ :

$$\mathbf{E} = \mathbf{K}'^T \mathbf{F} \mathbf{K} \quad (1.13)$$

Jak bylo zmíněno, odstraněním vlivu kalibračních matic na kamery bylo dosaženo značného zjednodušení vztahu 1.10

$$\mathbf{E} = [\mathbf{t}]_{\times} \mathbf{R} \quad (1.14)$$

Před tím, než bude objasněn postup, jak extrahovat z fundamentální, resp. esenciální matice relativní rotaci dvou kamer, podívejme se na způsoby, jak získat tuto matici samotnou.

### 1.4 Výpočet fundamenální matice

Pokud by nám byly známy projekční matice kamer, byl by k získání matice  $\mathbf{F}$  použit výpočet dle vzorce 1.9. V případě, že nám však není známa relativní rotace a translace dvou snímajících kamer, je třeba zvolit jiný přístup. Pro následné řešení

je potřeba získat v obou snímcích páry bodů, které promítají stejný bod v prostoru, ale každý na své vlastní projekční rovině (tzv. korespondence).

Fundamentální matice je definována jako  $\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0$ . Pro každý korespondující si pár bodů  $\mathbf{x}$  a  $\mathbf{x}'$  tedy dostáváme jednu lineární rovnici přispívající k řešení neznámé matice  $\mathbf{F}$ . Pokud uvažuju zápis bodů  $\mathbf{x} = (x, y, 1)^T$  a  $\mathbf{x}' = (x', y', 1)^T$ , po roznásobení definičního vztahu dostávám:

$$xx'f_{11} + yx'f_{12} + x'f_{13} + xy'f_{21} + yy'f_{22} + y'f_{23} + xf_{31} + yf_{32} + f_{33} = 0 \quad (1.15)$$

Vytknu-li vektor  $\mathbf{f}$  složený ze všech koeficientů matice  $\mathbf{F}$ , dostanu vztah

$$(xx', yx', x', xy', yy', y', x, y, 1)\mathbf{f} = 0 \quad (1.16)$$

Pro  $n$  korespondencí ( $n$  lineárních rovnic):

$$\mathbf{A}\mathbf{f} = \begin{bmatrix} x_1x'_1 & y_1x'_1 & x'_1 & x_1y'_1 & y_1y'_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_nx'_n & y_nx'_n & x'_n & x_ny'_n & y_ny'_n & y'_n & x_n & y_n & 1 \end{bmatrix} \quad (1.17)$$

Pokud má matice  $\mathbf{A}$  hodnot přesně 8, pak lze k výsledku dojít lineárními metodami. Tento případ však není příliš častý, proto je potřeba k řešení dojít některou z optimalizačních metod, například metodou nejmenších čtverců (popsána v kapitole 1.6/str.18). Obecně řešení nalezená těmito metodami nesplňují důležitou vlastnost fundamentální matice, a to její hodnot, která je žádaná o velikosti 2. Proto je potřeba upravit matici  $\mathbf{F}$  tak, aby tuto vlastnost měla. Nejjednodušším způsobem, jak toho dosáhnout, je pomocí singular value decomposition (SVD). Díky SVD získáme matice  $\mathbf{U}, \mathbf{D}, \mathbf{V}$  pro které platí

$$\mathbf{F} = \mathbf{U}\mathbf{D}\mathbf{V}^* \quad (1.18)$$

viz [3], str.281. Při SVD pro obecnou matici o rozměrech  $m \times n$  vzniká ortogonální matice  $\mathbf{U}$  o rozměrech  $m \times m$ , ortogonální matice  $\mathbf{V}$  o rozměrech  $n \times n$  a diagonální matice  $\mathbf{D}$  o rozměrech  $n \times n$ . Sloupce  $\mathbf{U}$  a  $\mathbf{V}$  jsou pak nazývány singulární vektory, prvky na diagonále matice  $\mathbf{D}$  se nazývají singulární hodnoty [4]. Je potřeba zachovat vlastnost fundamentální matice:  $\det \mathbf{F} = 0$ . K tomu použijeme podle [3] diagonální matici  $\mathbf{D}$ , které nastavíme prvek  $\mathbf{D}[3,3] = 0$  a znovu vypočteme matici  $\mathbf{F}$  podle vzorce 1.18 s novou maticí  $\mathbf{D}$ .

Postup nastíněný v této kapitole lze použít i s normalizovanými souřadnicemi korespondujících bodů, potom je výsledkem matice esenciální. Teď, když už je ustavena esenciální matice, bude ukázán postup získání žádané rotace.

## 1.5 Extrakce projekčních matic z esenciální matice

Jak bylo řečeno dříve, fundamentální matice v sobě obsahuje informaci o mapování z jedné obrazové roviny do druhé. Toho lze využít při hledání relativní orientace dvou kamer. Pomocí SVD je potřeba získat matice  $\mathbf{U}$  a  $\mathbf{V}$  (viz vzorec 1.18). Dále bude potřeba ortogonální matice  $\mathbf{W}$

$$\mathbf{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1.19)$$

Pokud budeme předpokládat první projekční matici  $\mathbf{P} = [\mathbf{I}|\mathbf{0}]$ , lze podle [3](str. 258) dojít pomocí SVD k dvěma možným řešením rotace matice  $\mathbf{P}'$  oproti  $\mathbf{P}$ .

$$\mathbf{R} = \mathbf{U}\mathbf{W}\mathbf{V}^T \text{ a } \mathbf{R} = \mathbf{U}\mathbf{W}^T\mathbf{V}^T \quad (1.20)$$

Při stejném předpokladu lze dojít k řešení translace  $\mathbf{t}$ :

$$\mathbf{t} = \mathbf{U}(0, 0, 1)^T = \pm \mathbf{u}_3 \quad (1.21)$$

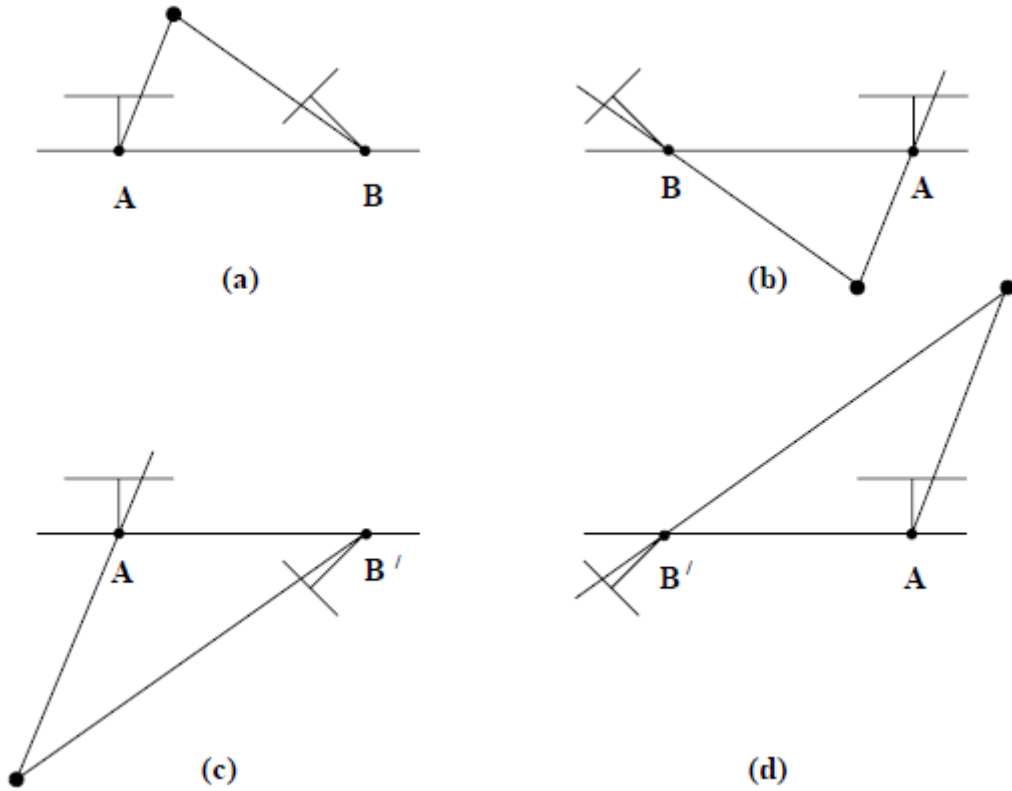
neboli třetí sloupec matice  $\mathbf{U}$ . Pro výslednou projekční matici  $\mathbf{P}'$  pak vyplývají čtyři možná řešení, kombinace dvou rotací a dvou translací.

$$\begin{aligned} \mathbf{P}' &= [\mathbf{U}\mathbf{W}\mathbf{V}^T \mid \mathbf{u}_3] \\ \text{nebo} \\ \mathbf{P}' &= [\mathbf{U}\mathbf{W}\mathbf{V}^T \mid -\mathbf{u}_3] \\ \text{nebo} \\ \mathbf{P}' &= [\mathbf{U}\mathbf{W}^T\mathbf{V}^T \mid \mathbf{u}_3] \\ \text{nebo} \\ \mathbf{P}' &= [\mathbf{U}\mathbf{W}^T\mathbf{V}^T \mid -\mathbf{u}_3] \end{aligned} \quad (1.22)$$

Je tedy zřejmé, že je potřeba určit tu správnou možnost z těchto čtyř. Musí být splněna podmínka, že snímané 3D body jsou před oběma kamerami zároveň. Řešením tohoto problému je použití lineární triangulace pro rekonstrukci 3D bodu z jeho dvou obrazů  $(\mathbf{x}, \mathbf{x}')$  a poté vyšetření pozice získaných 3D bodů vůči kamerám.

## 1.6 Lineární triangulace

V této kapitole bude probrána metoda objasněná v [3]. Vychází z rovnic pro obraz  $\mathbf{x}$  bodu  $\mathbf{X}$  ve tvaru  $\mathbf{x} = \mathbf{P}\mathbf{X}$ , resp.  $\mathbf{x}' = \mathbf{P}'\mathbf{X}$ . Tento tvar používá homogenní souřadnice



Obr. 1.2: Geometrická interpretace čtyř možných řešení extrakce projekční matice  $\mathbf{P}'$  z esenciální matice  $\mathbf{E}$ . [3]

vektoru, proto si nemusí být  $\mathbf{x}$  a  $\mathbf{P}\mathbf{X}$  rovny ve velikosti, ale pouze ve směru. Tento problém je odstraněn vypočtením vektorového součinu  $\mathbf{x} \times (\mathbf{P}\mathbf{X}) = 0$ , který lze zapsat také ve tvaru  $\mathbf{A}\mathbf{X} = 0$ , proto cílem bude stanovit matici  $\mathbf{A}$ . Vektorový součin lze rozepsat jako:

$$\begin{aligned} x(\mathbf{p}^{3T}\mathbf{X}) - (\mathbf{p}^{1T}\mathbf{X}) &= 0 \\ y(\mathbf{p}^{3T}\mathbf{X}) - (\mathbf{p}^{2T}\mathbf{X}) &= 0 \\ x(\mathbf{p}^{2T}\mathbf{X}) - y(\mathbf{p}^{1T}\mathbf{X}) &= 0 \end{aligned} \tag{1.23}$$

kde  $\mathbf{p}^{iT}$  jsou  $i$ -té řádky matice  $\mathbf{P}$ . Pouze dvě z těchto tří rovnic jsou lineárně nezávislé (podle [3], str.89), ty poslouží k sestavení matice  $\mathbf{A}$ . Stejný postup se uplatní na obraz  $\mathbf{x}'$ , matice  $\mathbf{A}$  je pak stanovena ve tvaru:

$$\mathbf{A} = \begin{bmatrix} x\mathbf{p}^{3T} - \mathbf{p}^{1T} \\ y\mathbf{p}^{3T} - \mathbf{p}^{2T} \\ x'\mathbf{p}^{3T} - \mathbf{p}'^{1T} \\ y'\mathbf{p}^{3T} - \mathbf{p}'^{2T} \end{bmatrix} \tag{1.24}$$



Kvůli přítomnosti šumu v obrazech nelze předpokládat nalezení vektoru  $\mathbf{X}$  takového, aby platila rovnost  $\mathbf{A}\mathbf{X} = 0$ , proto je vhodné použít metodu nejmenších čtverců. K tomu použijeme SVD, díky němuž získáme matici  $\mathbf{V}$  (vztah 1.18). Poslední sloupec matice  $\mathbf{V}$  potom vrací optimální řešení vektoru  $\mathbf{X}$ , aby se součin  $\mathbf{A}\mathbf{X}$  co nejvíce přiblížil nule.

## 2 ZÁKLADNÍ METODY DETEKCE A POPISU VÝZNAMNÝCH BODŮ

V následující kapitole bude popsán princip základních metod detekce a deskripce významných bodů. Metody jsou různé komplexity, některé popisují pouze nalezení významných bodů, jiné v sobě sdružují celý souhrn postupů, jak bod nejen nalézt, ale i popsat pro následné párování.

### 2.1 Harrisův detektor

Harrisův detektor rohů vychází z metody Hanse Moravce, která funguje na základě detekce změn intenzity jasu v okolí bodu. K získání rozdílu intenzity jasu je u Moravcova detektoru využito posunu obrazu o krátkou vzdálenost (nejčastěji 1 pixel) v různých směrech (nejčastěji 8okolí). Podle [5] je pak Moravec matematicky popsán jako:

$$E_{x,y} = \sum_{u,v} w_{u,v} (I_{x+u,y+v} - I_{u,v})^2 \quad (2.1)$$

kde

- $w$  udává okenní funkci popisující okolí bodu o rozměrech  $(u,v)$
- $E$  udává velikost změny intenzity jasu v okolí bodu  $(x,y)$
- $I$  je obrazová funkce.

Tento gradient hodnot jasu v originálním a posunutém obraze je v Harrisovi aproximován konvolucí původního obrazu s některým z operátorů (Sobel, Prewitt...) v horizontálním a vertikálním směru.

Dále Harrisova metoda navrhuje použít jako okenní funkci Gaussián, na rozdíl od binárního okna rozkládá exponenciálně váhu na jednotlivé body v okolí a tím potlačuje vliv šumu na výsledek.

Pro malé posuny  $(x,y)$  lze zapsat změnu  $E$  následovně:

$$E_{x,y} = (x, y) M \begin{pmatrix} x \\ y \end{pmatrix} \quad (2.2)$$

kde

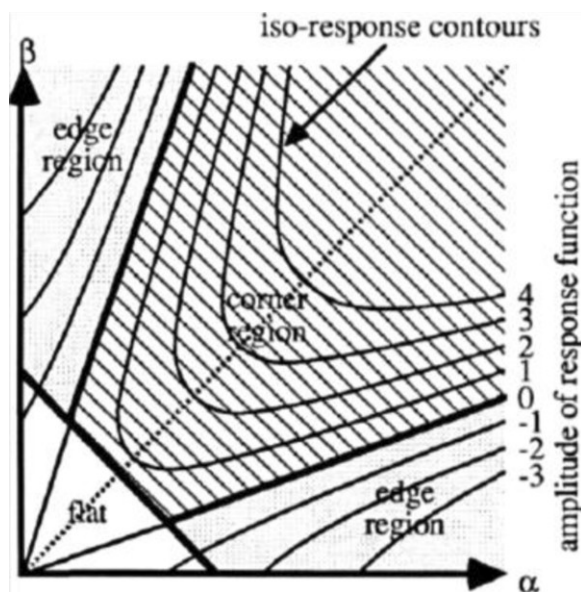
$$M = \sum_{u,v} w_{u,v} \begin{bmatrix} X^2 & XY \\ XY & Y^2 \end{bmatrix} \quad (2.3)$$

$X$  a  $Y$  jsou pak výsledky konvolucí obrazu s maskou (Prewitt, Sobel) ve směru osy  $x$ , resp.  $y$ .

Teď se dostáváme k jádru Harrisovy metody. Kvůli poměrně velké odezvě Moravcova operátoru na hrany v obraze zavádí Harrisův detektor operátor  $R$ , který vrací maximum pro rohy.

Nechť  $\alpha, \beta$  jsou vlastní čísla matice  $M$ . Pak mohou nastat tři případy.

- $\alpha$  i  $\beta$  jsou malé, tedy vyšetřované místo v obraze je ploché, nevyskytují se hrany ani rohy.
- Jedno z vlastních čísel  $\alpha, \beta$  je vysoké a to druhé pak nízké. To značí výskyt hrany v oblasti.
- Obě vlastní čísla  $\alpha, \beta$  jsou vysoká, tento případ značí výskyt rohu.



Obr. 2.1: Odezva operátoru  $R$  pro různé  $\alpha, \beta$ . [5]

Jde tedy o to maximalizovat  $\alpha$  i  $\beta$  zároveň. K tomu slouží dříve zmiňované  $R$ .

$$R = \det(M) - k \text{trace}(M)^2 \quad (2.4)$$

kde

$$\det(M) = \alpha\beta$$

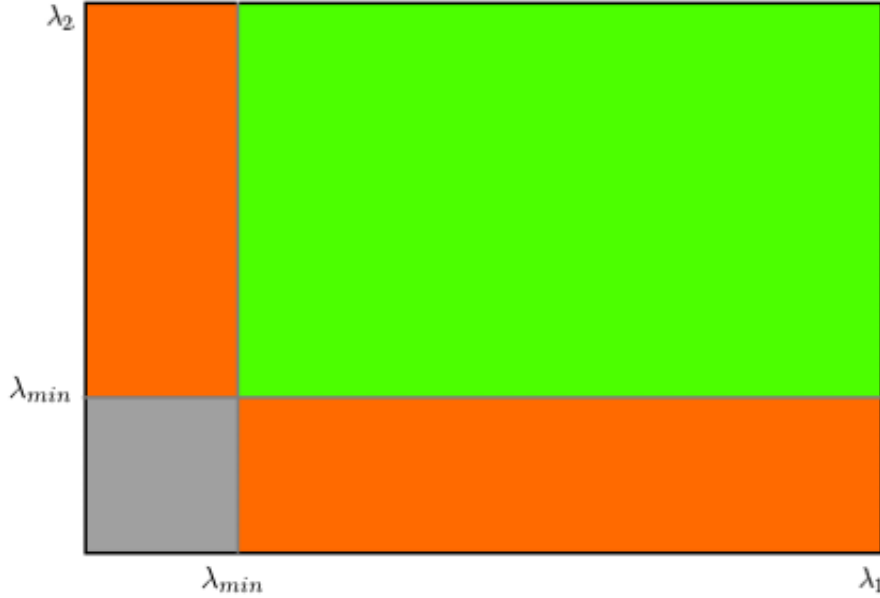
Potom je  $R$  záporné pro hrany, malé pro ploché oblasti a velké pro významné body.

## 2.2 Shi-Tomasi modifikace Harrisova detektoru

V roce 1993 byla navržena v práci [7] změna rozhodujícího operátoru  $R$  (vztah 2.4). Bod v obraze je považován za roh, pokud

$$\min(\alpha, \beta) > \lambda$$

kde  $\lambda$  je předem definovaný práh.



Obr. 2.2: Odezva rozhodovací funkce podle Shi-Tomasi. Pokud se obě vlastní čísla matice objevují v zeleném regionu, bod je považován za roh. [6]

## 2.3 Scale Invariant Feature Transform

Neboli SIFT, je metoda, která je schopna najít významné body skrz různá měřítka  $\delta$  původního obrazu. Výsledné body jsou pak zpárovatelné s množinou bodů získanou na přiblíženém/oddáleném obrazu díky navrženému deskriptoru.

### 2.3.1 Detekce extrémů

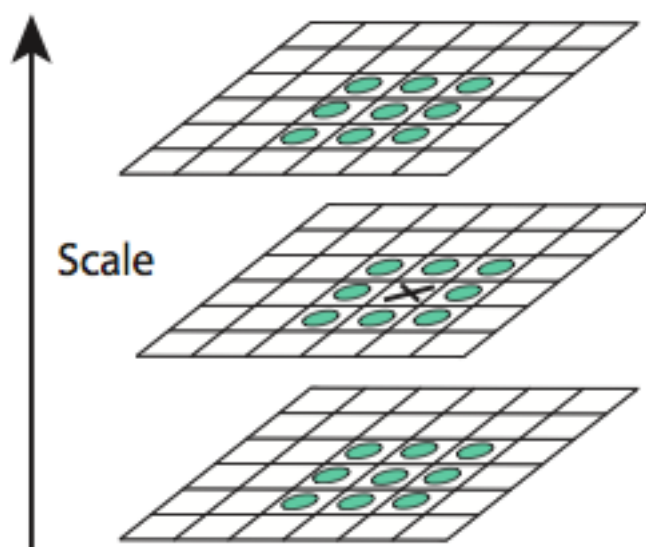
Prvním krokem je nalezení lokálních extrémů pro všechny body obrazu a pro předem definovaný rozsah škál. Pro úplnou měřítkovou nezávislost by bylo zapotřebí vypočítat Laplaceán pro každé zvětšení, což je výpočetně velmi náročné. Proto je zvolena aproximace diferencí Gaussiánů (DoG – difference-of-Gaussian). Výsledná funkce  $D$ , závislá na  $\delta$ , je podle [8] výsledkem konvoluce DoG a originálního obrazu:

$$D(x, y, \delta) = [G(x, y, k\delta) - G(x, y, \delta)] * I(x, y) \quad (2.5)$$

kde  $G$  je funkcí Gausiánu o zvětšení  $\delta$ , resp.  $k\delta$ .

Prakticky je tedy obraz nejdříve zvětšen o  $k\delta$ , poté se na oba obrazy – původní i zvětšený – aplikuje konvolucí gaussova funkce a následně se vypočte jejich difference. Poté se tento proces aplikuje znovu, s tím, že ze zvětšeného obrazu se stává původní, dokud nejsou získány DoG pro celý definovaný rozsah škál.

Po této operaci jsou hledány extrémy porovnáním každého pixelu s jeho (nejčastěji) 26okolí. Neboli pro 8okolí pro původní  $\delta$  a pro stejných 9 bodů v předchozím a následujícím  $\delta$  (viz. Obr. 2.3).



Obr. 2.3: 26okolí kandidáta na významný bod brané z přilehlých zvětšení/zmenšení. [6]

Pro urychlení celého algoritmu je proces prováděn po tzv. oktávách. Pokaždé, když se postupnou změnou měřítka dojde na 4krát menší obraz, je z původního obrazu vzat každý druhý pixel v horizontálním i vertikálním směru, čímž se zachová naintegrované zmenšení, ale zároveň se výrazně sníží výpočetní náročnost.

Metoda také poskytuje empiricky zjištěné optimální parametry. Doporučený počet oktáv je 4,  $\delta = 1,6$  a  $k = \sqrt{2}$ .

### 2.3.2 Lokalizace významných bodů

Prvním vylepšením k získání stabilnějších a přesnějších významných bodů je posunutí získaných extrémů za použití rozvoje do Taylerovy řady. Funkce  $D(x, y, \delta)$  (vztah 2.5) je aproximována Taylorovým polynomem 2. řádu v místě nalezeného extrému, díky čehož se poté nalezne interpolované maximum.

Za další se vyřadí všechny extrémy, jejichž jasová hodnota nepřesáhne stanovený práh (metoda navrhuje 3% rozsahu).

Posledním problémem, se kterým se metoda potýká, je velká odezva DoG na hrany. Tento problém se řeší obdobně jako u Harrisova detektoru. Je použita Hessova matice  $2 \times 2$ . Na rozdíl od Harrisovy metody, je zde pro případné odmítnutí extrému, jakožto významného bodu, použit poměr  $r$  mezi vlastními čísly Hessovy matice. Nechtě  $\alpha, \beta$  jsou vlastní čísla Hessovy matice  $H$ , pak  $r = \alpha/\beta$ . Poté je zkoumána velikost této proměnné, podle které se rozhoduje, zdali je extrém vhodným kandidátem na významný bod. Metoda navrhuje použít mezní  $r = 10$ .

### 2.3.3 Přidělení orientace

Znalost orientace každého detekovaného významného bodu je nezbytnou podmínkou pro invariantnost metody na případné rotaci. Tato orientace se hledá pomocí okolí všech bodů v obraze, kde jsou vypočteny gradienty velikosti a směru okolí, poté je sestaven histogram směrů, který obsahuje 36 hodnot popisujících  $360^\circ$  kolem bodu. Vkládané hodnoty jsou váženy pomocí gradientu velikosti okolí bodu.

Prvním krokem je sestavení histogramu směrů. Podle  $\delta$  významného bodu se zvolí Gausián, kterým se filtruje původní obraz, čímž vznikne filtrovaný obraz  $L$ . Poté se pro každý bod vypočte gradient směru  $\theta(x, y)$  a velikosti  $m(x, y)$ . Podle [8]

$$\theta(x, y) = \tan^{-1} \left( \frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \right),$$

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (2.6)$$

Každý bod, jehož vypočtený směr  $\theta(x, y)$  je vložen do histogramu, je vážen svým gradientem velikosti  $m(x, y)$  a Gausiánem váženým kruhovým oknem, jehož  $\delta$  je 1,5krát zvětšená oproti významnému bodu.

Špičky v histogramu směrů pak odpovídají dominantním směrům lokálních gradientů.

Za druhé je detekována největší špička v histogramu a je přiřazena k významnému bodu. Pokud se v histogramu nachází další špička, jejíž velikost je 80% a více té nejvyšší, je vytvořen stejný významný bod lišící se pouze touto orientací.

### 2.3.4 Deskriptor

K vytvoření deskriptoru se použije  $16 \times 16$  okolí významného bodu. Tento prostor je rozčleněn do 16 menších celků o velikosti  $4 \times 4$ . Podobným způsobem jako při přiřazení orientace je pro každý tento menší celek vytvořen histogram. Místo 36 hodnot je však celý rozsah  $360^\circ$  popsán 8 hodnotami. Každá orientace jednotlivého bodu je

ve svém histogramu vážena svou velikostí. Výsledný deskriptor je pak reprezentován jako vektor hodnot všech histogramů. Při použití osmi hodnot na histogram a rozčlenění na 16 menších bloků, z nichž každý bude mít svůj histogram orientací, je velikost vektorů  $16 * 8 = 128$ .

Metoda navrhuje další vylepšení. Protože je hodnot v histogramu pouze 8 na celý rozsah  $360^\circ$ , je navrženo metodou trilineární interpolace každý gradient orientace rozdělit poměrně také do dvou sousedních orientací. Dále je navrženo normalizovat vektor deskriptoru na jednotnou výslednou délku. Nelineární změny osvětlení nejvíce ovlivňují gradienty magnitudy, proto je jejich vliv potlačen prahováním každého jednoho sub-bloku na hodnotu 0,2. Poté je znovu celý vektor normalizován.

## 2.4 Speed-Up Robust Features

Vzhledem k výpočetní náročnosti výše popsané metody SIFT byla roku 2006 skupinou kolem Herberta Baye vyvinuta urychlená metoda – SURF [9]. Výpočetní čas je pro některé fáze algoritmu výrazně snížen při uchování podobné přesnosti.

### 2.4.1 Detekce významných bodů

Metoda SURF aproximuje LoG pomocí box filtrace. Velkou výhodou tohoto přístupu je možnost použít integrální obraz pro konvoluci filtru a původního obrazu. Jednotlivé body integrálního obrazu  $I_\sigma$  se pak vypočtou jako

$$I_\sigma(x, y) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j) \quad (2.7)$$

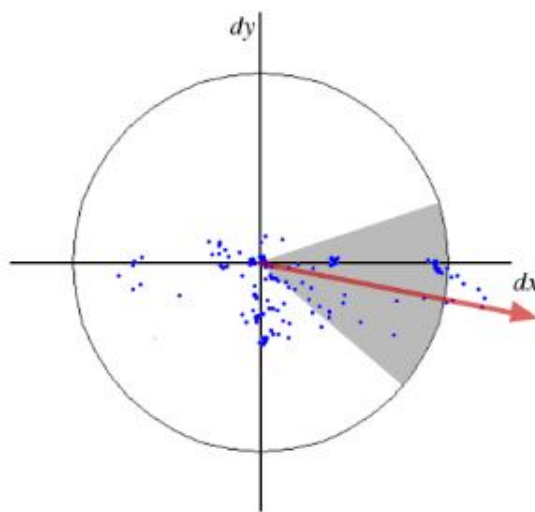
Integrální obraz stačí vypočítat jednou, poté už lze box filtr realizovat jako součet/odečet jeho 4 rohových bodů. Je ale nutné zdůraznit, že výpočetní náročnost narůstá spolu s velikostí filtru.

Škálové invariance je dosaženo postupným aplikováním box filtrů různých velikostí pokaždé na původní obraz, což nahrazuje postupné „rozmazávání“ obrazu pomocí Gausiánu u metody SIFT. Tyto filtry jsou aplikovány na jednotlivé oktávy (v podobném smyslu jako u SIFTu). Jako počáteční je použit filtr  $9 \times 9$ , který má zvětšení  $s = 1, 2$  (což aproximuje Gausián s  $\delta = 1, 2$ ). Další vrstvy jsou získány konvolucí původního obrazu s většími maskami, přesněji to jsou filtry  $15 \times 15$ ,  $21 \times 21$  a  $27 \times 27$  (vždy s přírůstkem o velikosti 6). Pro každou novou oktávu je přírůstek zdvojnásoben a počáteční filtr poslouží jako druhý nejmenší v předchozí oktávě. Tedy pro druhou oktávu budou rozměry filtru 15, 27, 39, 51.

Poté jsou v okolí bodu  $3 \times 3 \times 3$  hledány maxima a jejich umístění je interpolováno stejně jako v metodě SIFT, tedy rozvojem do Taylorovy řady.

## 2.4.2 Deskriptor

Pro určení orientace každého významného bodu se nejdříve vypočte Haarova vlnková transformace ve směru os  $x$  i  $y$  v okolí bodu o velikosti  $6s$ , kde  $s$  je zvětšení, pro které byl bod detekován. Pro filtraci se použije integrálního obrazu, díky čemuž se znovu výrazně sníží výpočetní náročnost. Poté se vlnkové odezvy ještě přefiltrují pomocí Gausiánu (kde  $\delta = 2s$ ). Poté je posuvným oknem o velikosti  $60^\circ$  postupně určena suma orientací pro každý takový blok zvlášť v celém rozsahu  $360^\circ$  (viz obr. 2.4). Největší ze sum je pak přiřazena bodu jako jeho vlastní orientace.



Obr. 2.4: Šedě podbarvená výseč značí  $60^\circ$ , pro které je vypočtena určující orientace (znázorněna červenou šipkou). Poté se výseč přesune a výpočet se opakuje. [9]

K určení deskriptoru je vzato okolí významného bodu o velikosti  $20s \times 20s$  ve směru jemu určené orientace. Následně je tento prostor rozdělen na 16 menších bloků o stejné velikosti. Pro každý takový blok je spočtena Haarova vlnková transformace v horizontálním ( $dx$ ) a vertikálním ( $dy$ ) směru. Poté je určen 4D vektor  $\mathbf{v}$  pro každý takový blok

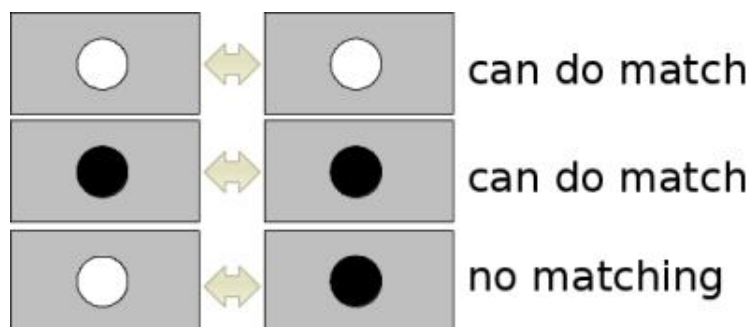
$$\mathbf{v} = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|) \quad (2.8)$$

Konečná podoba deskriptoru je pak vektor obsahující 16 těchto vektorů  $\mathbf{v}$  pro každý blok. Jeho velikost je tedy 64.

Posledním krokem je přidání znaménka Laplaciánu (aproximovaného funkcí trace Hessovy matice). Tento parametr rozlišuje intenzitu jasu okolí významného bodu od pozadí. Při párování významných bodů pak porovnává jen ty skupiny bodů, které jsou buď tmavé na světlém pozadí, nebo naopak. (viz obr. 2.5) Vzhledem k tomu,



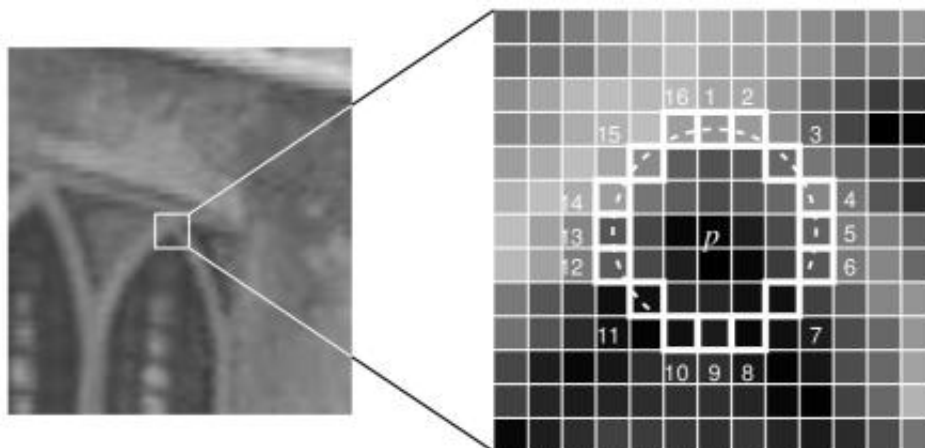
že tento výpočet byl již proveden dříve, nezvyšuje se tím výpočetní náročnost, ale zkrátí se čas potřebný pro párování samotné.



Obr. 2.5: Znázornění možných situací při porovnávání znamének Laplaciánu. Na šedém pozadí jsou porovnávány pouze světlejší, nebo tmavší obrazy. Při rozdílném znaménku nejsou připuštěny k dalšímu porovnání. [6]

## 2.5 Features from Accelerated Segment Test

Tato metoda, zvaná také jako FAST, byla navržena pro použití v real-time aplikacích. Jak už název napovídá, vyniká hlavně rychlou detekcí významných bodů.



Obr. 2.6: Zvolené okolí bodu sloužící pro detekci rohů metodou FAST. [10]

Pro zvolený bod  $\mathbf{p}$  obrazu je určena kružnice o obvodu 16 bodů se samotným bodem  $\mathbf{p}$  ve středu (viz obr. 2.6). Nechť  $t$  je zvolený práh a  $I_p$  je intenzita jasů bodu  $\mathbf{p}$ . Pak je bod  $\mathbf{p}$  považován za roh v případě, že existuje  $n$  sousedících bodů na kružnici, jejichž jas je menší než  $I_p - t$ , nebo větší než  $I_p + t$ . Neboli musí existovat  $n$

sousedících bodů, které jsou tmavší nebo světlejší než bod  $\mathbf{p}$  minimálně o hodnotu prahu  $t$ .  $n$  je doporučeno volit rovno 12, neboť tím pádem je možné použít zrychlený test, při kterém se testují pouze čtyři body – 1, 5, 9, 13. Tento postup rychle vyloučí velké množství bodů, které nejsou rohy. Pro to, aby bod mohl být rohem, musí minimálně 3 z těchto 4 bodů splňovat výše uvedenou podmínku. Pokud nesplňuje, je jakožto roh zamítnut, pokud ano, je připuštěn k plnému testu.

### 2.5.1 Strojové učení

Nevýhodami výše popsaného postupu jsou mezi jinými:

- Algoritmus není příliš vhodný pro  $n < 12$
- Volba pixelů není optimální, neboť efektivita nalezení rohů je závislá na způsobu průchodu obrazu a rozložení rohů samotných
- Výsledky zrychleného testu nejsou nadále použity.

Jako řešení těchto problémů přichází metoda s použitím strojového učení.

K tomuto postupu je zapotřebí mít množinu testovacích obrazů. V každém z obrazů se naleznou pomocí FAST významné body. Pro každý takový bod se uloží jeho 16 okolních bodů (jak je popsáno výše) jako vektor  $\mathbf{P}$ . Pro pixel  $\mathbf{x}$  z tohoto vektoru a jeho intenzitu  $I_x$  mohou nastat 3 stavy

- pixel je světlejší než roh
- pixel je podobný jako roh (je v rozsahu  $\pm t$ )
- pixel je tmavší než roh

Na základě stavu každého pixelu je vektor  $\mathbf{P}$  rozdělen na „podvektory“  $P_d$  (darker),  $P_s$  (similar) a  $P_b$  (brighter).

Zavedeme boolovskou proměnnou  $K_p$ , která je ‘true’, pokud je bod  $\mathbf{p}$  roh a ‘false’, pokud není. Pomocí metody ID3, která aplikuje algoritmus rozhodovacího stromu, získáme z každého podvektoru pixel, který nese nejvíce informací o tom, zda je bod  $\mathbf{p}$  roh (což je měřeno entropií  $K_p$ ). Tento postup je rekurzivně opakován pro všechny podvektory, dokud jejich entropie  $K_p$  neklesne na nulu. Nakonec je uložen vytvořený rozhodovací strom, který poté slouží pro rychlé detekování rohů ve zbylých obrazech.

### 2.5.2 Non-Maximal Suppression

Dalším problémem, který vyplývá z podstaty metody, je detekce dvou (a více) významných bodů vedle sebe.

Nechť je  $V$  váhová funkce, kterou získáme jako sumu absolutních diferencí významného bodu  $\mathbf{p}$  a okolních 16 pixelů. Potom je ze všech sousedících bodů ponechán pouze ten s nejvyšší hodnotou funkce  $V$  a zbytek je smazán.

## 2.6 Binary Robust Independent Elementary Features

Zkráceně BRIEF [11], je metoda pro nalezení paměťově efektivního deskriptoru. Vzhledem k tomu, že metody SIFT, SURF, apod. používají jako deskriptor vektory o délce 64 až 128, kdy každý člen je reprezentován typem float, může se velikost paměti zabírána tímto deskriptorem vyšplhat až na 512 bytů. V případě, že aplikace vyžaduje tisíce významných bodů, je už paměťová náročnost aspekt, se kterým je nutné počítat.

Prvním krokem je filtrace Gausiánem pro zmenšení odezvy na šum. Ve vybraném okolí bodu o velikosti  $S \times S$  pak vybereme  $n$  párů  $(x_i, y_i)$ , kteréžto budou přiřazeny do vektorů  $x$  a  $y$ . Způsobů, jak provést výběr, se nabízí několik, nicméně nejlepší výsledky se podle návrhu metody dostaví s náhodným výběrem za použití Gausova rozdělení, tedy preferovány budou body blíže středu (významnému bodu). Deskriptor je pak proveden jako binární o velikosti  $n$ . Nechť  $i$  náleží od 1 do  $n$ , pak každý  $i$ -tý prvek deskriptoru popisuje vztah mezi bodem  $x_i$  a  $y_i$ . Je-li  $x$  větší intenzity, zapíše se 1, není-li větší, zapíše se 0.

Velkou výhodou je možnost použít pro párování Hammingovu vzdálenost.

## 2.7 Srovnání jednotlivých metod

V kapitole uvedu výčet kladů a záporů výše popsaných postupů. Vyňaty budou metody Harrisova detektoru a Shi-Tomasi, neboť tyto nejsou nezávislé na zvětšení, neboli nesplňují jeden z hlavních předpokladů pro správnou detekci při změně úhlu pohledu.

Jedním z důležitých ukazatelů u detekčních metod je počet detekovaných bodů. Nejedná se o ukazatel rozhodující, neboť důležitou informací je počet bodů spárovaných, nicméně každý jednotlivý významný bod navíc zvyšuje šanci k následnému navýšení nalezených párů. Jak ukazuje tabulka 2.1, nejvíce nalezených bodů má podle [1] pro různé scény metoda SIFT. Pro detailnější popis analýzy odkazují čtenáře na samotnou práci [1].

Podobně P. M. Panchal, S. R. Panchal a S. K. Shah v práci [2] ukazují převahu SIFTu co se týče nalezených významných bodů.

Tabulka 2.2 mimo jiné ukazuje i další velmi důležitý parametr každé metody. Tím je počet spárovaných významných bodů. Vzhledem k vyššímu počtu nalezených významných bodů lze očekávat, že metoda SIFT bude mít také více shod mezi množinami bodů. Tento předpoklad potvrzují i další výsledky práce [1], jak ukazuje tabulka 2.3

Pair	Window		Brick Wall		Boat		Building	
Image	A	B	A	B	A	B	A	B
<b>SIFT</b>	859	754	1117	1106	484	604	842	550
<b>SURF</b>	770	734	507	516	209	326	606	202
<b>FAST</b>	86	128	13	6	50	294	216	52

Tab. 2.1: Výsledky detekce bodů jednotlivých metod v [1]

Algorithm	Detected feature Points		Matching feature point	Feature matching Time
	Image1	Image2		
<b>SIFT</b>	892	934	41	1.543 s
<b>SURF</b>	281	245	28	0.546

Tab. 2.2: Výsledky porovnání SIFT a SURF podle [2]

Účelem této práce je spíše nalézt přesný a spolehlivý postup k získání relativní orientace dvou kamer, než-li minimalizovat čas potřebný ke stanovení takové orientace, proto aspekt časové náročnosti představených metod nebude v této části diskutován. Ze závěrů zmíněných prací [1] a [2] a z představené teorie lze odvodit následující:

- Počet detekovaných bodů je velmi závislý na vstupním snímku.
- SIFT vrací největší počet významných bodů pr většinu testovaných snímků.
- FAST považuje za roh velmi malé množství bodů, čímž je dosaženo značné rychlosti výpočtu oproti zbylým dvěma metodám.
- SURF vykazuje menší výpočetní náročnost při zachování podobných výsledků jako SIFT.
- SIFT i SURF jsou díky svým deskriptorům značně paměťově náročné metody.
- Při kombinaci FASTu s deskriptorem BRIEF je naopak paměťová náročnost značně snížena.

## 2.8 Párování

Po získání množin významných bodů z obou pořízených snímků přichází na řadu jejich vzájemné párování. Znalost vzájemných vazeb mezi dvěma rohy v odlišných snímcích zobrazujících stejné místo scény je základním předpokladem pro další postup. Právě tato informace je za využití epipolární geometrie klíčovou k určení vzájemného úhlu kamer.

Images	SIFT	SURF	FAST
Window	17	24	17
Brick Wall	113	72	5
Boat	23	10	18
Building	203	102	26

Tab. 2.3: Počty spárovaných bodů pro jednotlivé metody v práci [1]

### 2.8.1 Metoda hrubé síly

Pro nalezení takových dvojic rohů použiju prosté porovnávání deskriptoru každého jednoho rohu z jednoho snímku s celou množinou deskriptorů druhého snímku. Poté podle daného kritéria bude vrácen ten „nejbližší“ prvek. Takto bude nalezena dvojice pro každý roh prvního snímku. Jako kritérium je vhodné zvolit vzdálenost v  $n$  rozměrném prostoru deskriptoru.

### 2.8.2 Vzdálenosti

Podle vybrané metody z výše popsaných je možné uvažovat 2 typy vzdáleností.

#### Hammingova vzdálenost

Jak je výše zmíněno, tento druh párování významných bodů je vhodné zvolit při použití deskriptoru popsaného metodou BRIEF. Hammingova vzdálenost je definována jako počet míst, na kterých se dva vektory  $\mathbf{a}$  a  $\mathbf{b}$  o velikosti  $n$  liší.

$$d_H(\mathbf{a}, \mathbf{b}) = \sum_{i=1}^n |a_i - b_i| \quad (2.9)$$

Výpočetně je tento postup velmi nenáročný.

#### Euklidovská vzdálenost

Princip euklidovského prostoru a metriky je obecně znám, pro úplnost pouze uvádím vzorec pro výpočet euklidovské vzdálenosti v  $n$  rozměrném prostoru pro dva vektory  $\mathbf{x}$  a  $\mathbf{y}$ .

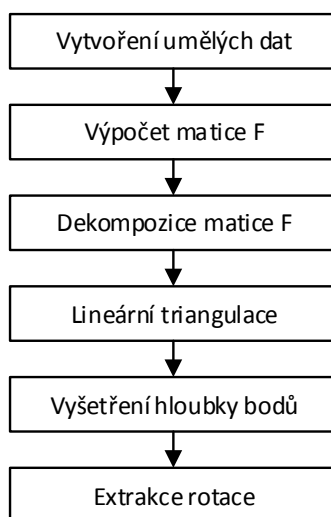
$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2.10)$$

### 3 IMPLEMENTACE

Nastudovanou teorii epipolární geometrie a detekce významných bodů bylo cílem implementovat v jazyce C/C++. K tomuto byla použita otevřená knihovna OpenCV [6] pro C++, která poskytuje značnou část potřebných funkcí. Před samotnou implementací ve zmíněném jazyce proběhlo experimentální ověření a vyladění části algoritmu zabývajícím se epipolární geometrií na umělých datech v programu Matlab, jakožto velmi vhodném nástroji pro práci s obrazem a maticemi a tím samym nástrojem byla později stanovena i referenční měření díky stereo calibration toolboxu. O měření bude pojednávat následující kapitola, teď si pojďme přiblížit samotný algoritmus testovaný nejdříve v Matlabu, poté přepsaný do C++.

#### 3.1 Experimentální ověření epipolární gemoetrie na umělých datech

I přes řadu nástrojů, které knihovna OpenCV poskytuje, je práce s obrazem a maticemi v jazyce C++ obecně náročnější a méně přehledná, než je tomu v jiných prostředích, jakým je například Matlab. Proto prvotní návrhy algoritmů extrahujících relativní orientaci dvou kamer z fundamentální matice jsou vytvořeny ve zmiňovaném programu. Stručně bude představen postup pro testování, neboť většina skriptu je velmi podobná s implementací v C/C++, proto by hlubší popis pouze zdvojoval později předložené informace. Celý test běží ve skriptu `main.m`, ve kterém se dají

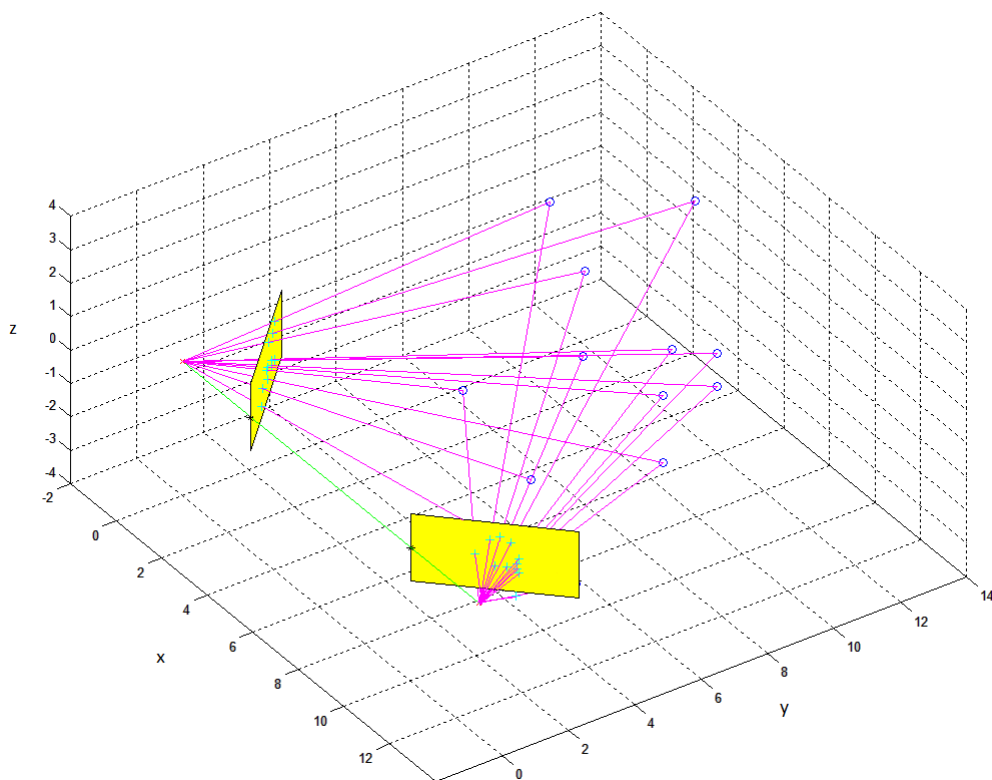


Obr. 3.1: Vývojový diagram testovacího .m skriptu

nastavit 2 parametry:

- `rot_origin` - vektor úhlů pro rotace podle jednotlivých os
- `n` - počet vytvořených 3D bodů

O vytvoření umělých dat se stará funkce `init`, která na základě zadané rotace a počtu `n` vrátí dvě pole korespondencí, každé o velikosti `n`, které odpovídají projekci vygenerovaných třidimenzionálních bodů pro jednotlivé projekční matice. Výpočet

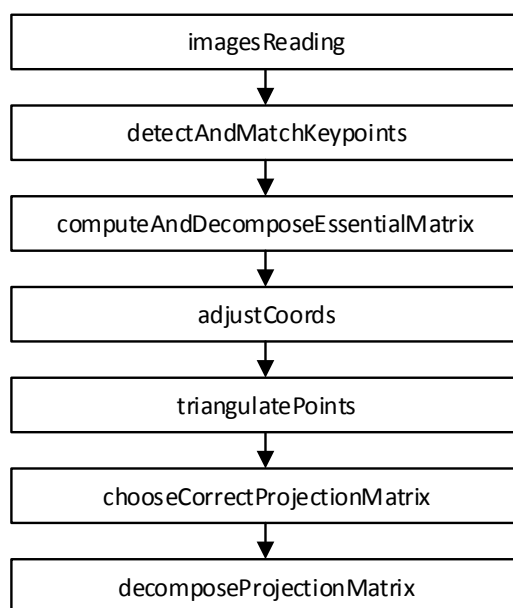


Obr. 3.2: Vizualizace tvorby umělých korespondencí

fundamentální matice  $\mathbf{F}$  metodou nejmenších čtverců (podle kapitoly 1.4) obstarává funkce `FMatrixbyLeastSquares`, jejímiž vstupy jsou korespondence  $x_1, x_2$  a jediným výstupem je matice  $\mathbf{F}$ . Protože všechny další úkony, jako je lineární triangulace, vyšetření hloubky zrekonstruovaných 3D bodů vůči oběma kamerám a na základě toho i výběr správné matice  $\mathbf{P}'$ , je potřeba udělat pro všechny 4 kombinace dvou extrapolovaných rotací a dvou translací (viz. vzorec 1.22), byly všechny výše jmenované postupy vloženy do jedné funkce `ChooseCorrectP2`, která vrací pole všech projekčních matic  $\mathbf{P}'$ , k nim náležející triangulované body a integer `corr`, který nese informaci o tom, na kterém indexu leží vybraná matice  $\mathbf{P}'$ . Vzhledem k povaze testování bylo zanedbáno mnoho faktorů, zejména vliv kalibrace kamer, proto nebylo potřeba použít esenciální matici s normovanými souřadnicemi.

## 3.2 C kód

Jak bylo zmíněno v úvodu, k vytvoření programu v jazyce C++ byla použita knihovna OpenCV, verze 3.1, s přidanou knihovnou `contrib`, která (mimo jiné) obsahuje algoritmy pro detekci a popis významných bodů jako jsou SIFT nebo SURF. Aplikace vznikla v prostředí Microsoft Visual Studio 2015 Community. Celý program běží z hlavního souboru `main.cpp`, ze kterého jsou volány vytvořené funkce.



Obr. 3.3: Vývojový diagram volaných funkcí ve zdrojovém kódu aplikace.

### 3.2.1 Načtení obrazů

V prvním kroku je potřeba načíst testované snímky, o což se stará funkce `imagesReading`. Jejími vstupy jsou:

- `int aNum`
- `Mat img1`
- `Mat img2`

`img1` a `img2` jsou typu `Mat`, což je typ implementovaný knihovnou OpenCV, suplující funkci matice. Uvnitř funkce je přítomen `switch`, který podle `aNum` načte jeden pár snímků do zmíněných matic. Funkce může vrátit 2 výjimky - v případě, že bylo do `aNum` zadáno větší číslo než je počet poskytnutých dvojic snímků, nebo v případě, že



se snímek nepodařilo načíst/najít. To, který snímek se nahraje, určuje v `main.cpp` integer `n`.

### 3.2.2 Detekce významných bodů a hledání korespondencí

Poté, co jsou vybrány testované snímky a nahrány do jim náležejících matic, jsou pomocí funkce `detectAndMatchKeypoints` nalezeny korespondující si body pomocí metody SIFT. Jejimi vstupy jsou:

- `Mat aImg1`
- `Mat aImg2`
- `vector<Point2f> matchedPoints1`
- `vector<Point2f> matchedPoints2`

Nejdříve jsou objektem detektoru nalezeny významné body ve snímcích `aImg1` a `aImg2`, poté je inicializován Brute-Force matcher, který nalezne korespondence.

---

```
1  Ptr<Feature2D> detector = xfeatures2d::SIFT::create();
2
3  vector<KeyPoint> keypoints1, keypoints2;
4  Mat descriptors1, descriptors2;
5
6  detector->detectAndCompute(aImg1, noArray(), keypoints1
7      , descriptors1);
8  detector->detectAndCompute(aImg2, noArray(), keypoints2
9      , descriptors2);
10
11 BFMatcher matcher(NORM_L2, true);
12 vector< DMatch> matches;
13 matcher.match(descriptors1, descriptors2, matches);
```

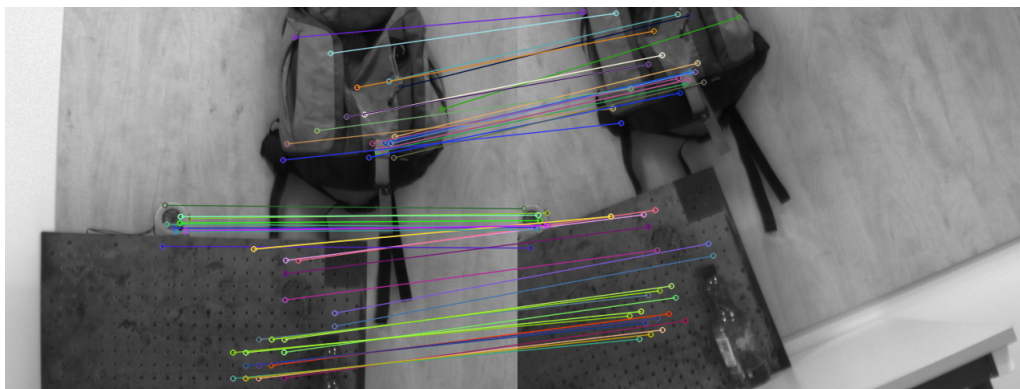
---

Výpis 3.1: Detekce a párování významných bodů

Tyto korespondence jsou vráceny pomocí vektorů `matchedPoints1` a `matchedPoints2`.

### 3.2.3 Výpočet esenciální matice a nalezení všech možných řešení projekční matice $P'$

Z korespondujících si dvojic bodů lze vypočítat fundamentální matici, jak bylo ukázáno v kapitole 1.4 (vzorec 1.15 až 1.17). Pokud je v době výpočtu k dispozici kalibrační matice použitých kamer, lze stanovit matici esenciální (kapitola 1.3). Což



Obr. 3.4: Nalezené korespondence.

je náš případ, neboť kalibrační matice byla stanovena společně s referenčním měřením (viz. kapitola 4). O celý proces výpočtu a dekompozice esenciální matice se stará funkce `computeAndDecomposeEssentialMatrix`. Jejími vstupy jsou:

- `vector<Point2f> aPnts1`
- `vector<Point2f> aPnts2`
- `Mat aK`
- `Mat inliers`
- `vector<Mat> P2`

Za použití knihovny `openCV` funkce `findEssentialMat` je metodou RANSAC nalezena esenciální matice. Tato funkce vrací také v matici `inliers` indexy bodů vyhovujících výstupu RANSACu, tzv. inlierů. Ty se budou dále hodit při triangulaci a hledání správného ze 4 řešení projekční matice. Další funkcí využitou z knihovny `OpenCV` je `decomposeEssentialMat`, jejímž vstupem je esenciální matice `E`, výstupem pak dvě možné rotace a jedna translace. Druhá translace je jednoduše dopočtena vynásobením  $-1$ . Všechny 4 kombinace rotací a translací jsou vráceny ve vektoru čtyř matic `P2`.

### 3.2.4 Lineární triangulace a vyšetření hloubky bodů

Lineární triangulací jsou z korespondencí získány 3D body, které poté slouží ke zvolení správné projekční matice druhé kamery. Pro samotnou triangulaci je použita `openCV` funkce `triangulatePoints`. Ještě před jejím voláním je však potřeba upravit vhodně korespondující body. K tomuto účelu slouží funkce `adjustCoords`, jejímiž vstupy jsou:

- `Mat inliers`
- `vector<Point2f> aPnts1`
- `vector<Point2f> aPnts2`

- Mat `aK`
- vector<Point2f> `npoints1`
- vector<Point2f> `npoints2`

Funkce do předaných vektorů `npoints1` a `npoints2` uloží pouze inliery (vybrané body z vektorů `aPnts1` a `aPnts2` podle masky `inliers`), které normuje podle vstupní kalibrační matice `aK`. Takto upravené body se pak spolu s každou možnou maticí `P2` předají do `triangulatePoints`, výstupem jsou zrekonstruované 3D body. Výběr správné projekční matice `P2` řeší funkce `chooseCorrectProjectionMatrix`, která šetří hloubky zrekonstruovaných 3D bodů. Vstupy jí náležející jsou:

- vector<Mat> `points3D`
- vector<Mat> `P2`
- int `correctOne`

Jak ilustruje obr. 1.2, je potřeba, aby byly nejlépe všechny body před obě kamerami. Postup je pro každý prvek vektorů `points3D` a `P2` následující:

1. Vyšetří se znaménko hodnoty v ose **z** pro vybraný počet 3D bodů.
2. Ty samé body se otáčejí podle testované matice `P2`.
3. Vyšetří se znaménko hodnoty v ose **z** pro vybraný počet takto otáčených 3D bodů.

Poté se přistoupí k dalšímu prvku vektoru `points3D` a `P2`. Během cyklu je průběžně ukládán největší počet kladných výsledků testu pro obě (rotované i nerotované) množiny 3D bodů. Vzhledem k možné přítomnosti špatně detekovaných a přiřazených bodů není vhodné podmínit správnost výsledku požadavkem, aby před kamerami byly všechny triangulované body. Proto je za správné řešení vybráno to, které má výsledek největší. Index vybrané matice `P2` je vrácen v proměnné `correctOne`.

### 3.2.5 Extrakce rotace

Posledním krokem je získání úhlů rotace pro jednotlivé osy z vybraného řešení. Tuto funkci plní `decomposeProjectionMatrix`, která má vstupy:

- vector<Mat> `P2`
- float `Rx`
- float `Ry`
- float `Rz`
- Mat `t`

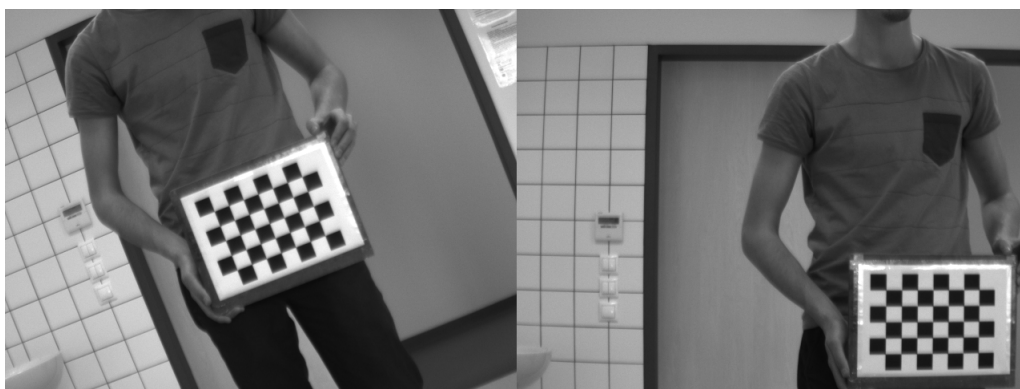
Ze vstupní matice P2 se extrahuje vektor translace a rotace pro jednotlivé osy podle známých vztahů:

$$\begin{aligned} R_x &= \tan^{-1}(r_{2,1}/r_{2,2}) \\ R_y &= -\sin^{-1}(r_{2,0}) \\ R_z &= \tan^{-1}(r_{1,0}/r_{0,0}) \end{aligned} \tag{3.1}$$

kde  $r_{i,j}$  je prvek matice rotace i-tého řádku a j-tého sloupce. Použity jsou standardní funkce C++ `atan2` a `asin`. Vektor translace je získán jakožto poslední sloupec matice P2.

## 4 EXPERIMENTÁLNÍ OVĚŘENÍ FUNKČNOSTI

K ověření funkčnosti aplikace byly použity dvě kamery Imaging Source typ DMK 21AF04 s rozlišením  $640 \times 480$  px, upevněné společně na polohovatelném stativu, kterými byly pořízeny 3 sady snímků. Každá sada obsahuje 20 párů fotek s kalibrační šachovnicí, které slouží jak k získání kalibrační matice, tak k výpočtu referenční orientace pomocí Stereo Calibration Toolboxu v Matlabu 2016a. Dále je v každé sadě jeden pár testovaných snímků přímo pro aplikaci, který už neobsahuje kalibrační šachovnici.



Obr. 4.1: Ukázka páru vstupních snímků do Stereo Calibration Toolboxu.

### 4.1 Referenční testy

Pro nejlepší výsledky je podle dokumentace programu Matlab 2016a potřeba mezi 10 a 20 snímky z každé kamery. Ze vstupních snímků nejdříve Matlab vybral ty, na kterých je schopný nalézt šachovnici, poté jsem manuálně vybral ty snímky, které měly šachovnici detekovanou správně. Počty ukazuje tabulka 4.1. Pro každý test

test	vstupní snímky [-]	vyhovuje matlabu [-]	manuální kontrola [-]
1	20	16	16
2	20	19	7
3	20	18	18

Tab. 4.1: Počet použitých snímků pro jednotlivé testy

byly spočítány dvě kalibrační matice (pro každou kameru zvlášť). Vzhledem k faktu, že jsou kamery stejného výrobce i typu, pro zjednodušení bude na obě dvě zařízení

aplikována stejná kalibrační matice. Je tedy k dispozici celkem 6 kalibračních matic:

$$\begin{aligned}
K_{1,1} &= \begin{bmatrix} 1576,3 & 0 & 119,1 \\ 0 & 1582,4 & 184,9 \\ 0 & 0 & 1 \end{bmatrix}, \quad K_{1,2} = \begin{bmatrix} 1611,0 & 0 & 337,7 \\ 0 & 1611,8 & 214,5 \\ 0 & 0 & 1 \end{bmatrix} \\
K_{2,1} &= \begin{bmatrix} 1574,6 & 0 & 409,7 \\ 0 & 1571,3 & 241,1 \\ 0 & 0 & 1 \end{bmatrix}, \quad K_{2,2} = \begin{bmatrix} 1538,6 & 0 & 564,3 \\ 0 & 1548,0 & 139,0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.1) \\
K_{3,1} &= \begin{bmatrix} 1432,3 & 0 & 389,2 \\ 0 & 1432,9 & 242,0 \\ 0 & 0 & 1 \end{bmatrix}, \quad K_{3,2} = \begin{bmatrix} 1429,3 & 0 & 358,1 \\ 0 & 1431,6 & 196,5 \\ 0 & 0 & 1 \end{bmatrix}
\end{aligned}$$

kde  $K_{i,j}$  odpovídá  $i$ -tému testu a  $j$ -té kameře. Na první pohled jde vidět, že obě kalibrační matice z měření číslo 2 se značně liší od ostatních. Vzhledem k rozlišení snímků ( $640 \times 480$ ) lze předpokládat, že výpočet středů promítání matic  $K_{2,1}$  a  $K_{2,2}$  byl zatížen hrubou chybou při výpočtu. Jak ukazuje tabulka č. 4.1, bylo pro test 2 použito nejméně snímků, Matlab tedy nezaručuje nejlepší možné výsledky. Kapitola 4.2 později potvrdí, že ač bylo původně zamýšleno brát výsledky Stereo Calibration Toolboxu jako referenční, k takovému kroku je potřeba zajistit maximální možný počet kalibračních snímků, což u testu 2 splněno nebylo. Proto byly z následného testování vyřazeny matice  $K_{2,1}$  a  $K_{2,2}$ , pár snímků z měření 2 byl v testování ponechán, ač nelze plně spoléhat na referenční údaje o rotaci a translaci.

Jak bylo naznačeno, kalibrace pomocí Stereo Calibration Toolboxu v Matlabu vrací také vypočtený úhel dvou snímajících kamer, který byl pro účely testování považován jako referenční. Spolu s rotační maticí poskytuje také výslednou translaci obou kamer. Tu však nelze porovnat přímo, vzhledem k povaze úlohy lze translační vektor použít pouze pro kontrolu směru posunu. Výsledky jednotlivých testů rotace uvádí tabulka 4.2.

test	rotace podle osy [°]			translace v ose [cm]		
[-]	<b>x</b>	<b>y</b>	<b>z</b>	<b>x</b>	<b>y</b>	<b>z</b>
1	6,49	-19,44	0,59	43,242	0,872	3,535
2	5,46	-13,66	-54,74	31,179	10,243	10,086
3	-6,69	-3,68	21,98	48,012	3,501	-4,587

Tab. 4.2: Referenční hodnoty rotací a translací.

## 4.2 Testování algoritmu

V této kapitole budou představeny výsledky testování aplikace pro 3 páry snímků o různých relativních orientacích. Každý z párů byl proměřen pro čtyři různá nastavení kalibrační matice, jak uvádí předchozí kapitola. Dále budou tyto výsledky porovnány s referenčními daty podle tabulky 4.2 a bude vypočtena absolutní chyba a směrodatná odchylka. Absolutní chyba byla vypočtena podle vzorce:

$$\Delta_x = x_M - x_S \quad (4.2)$$

kde  $x_M$  je měřená hodnota veličiny  $x$  a  $x_S$  je hodnota skutečná, referenční.

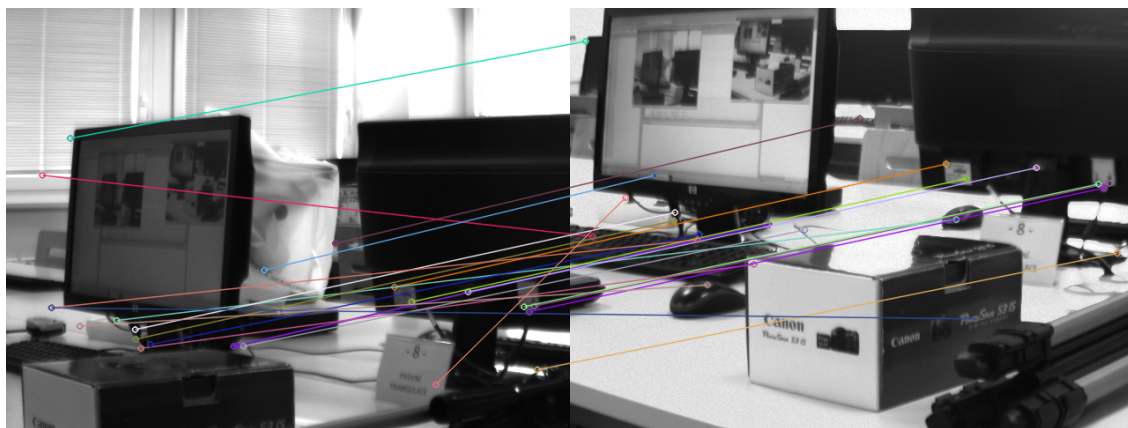
Směrodatná odchylka byla vypočtena podle vzorce:

$$\sigma_x = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x}) \quad (4.3)$$

kde  $\bar{x}$  je aritmetický průměr naměřených hodnot.

Vzhledem k tomu, že lze porovnat pouze směr translace, musí být nejdříve provedena normalizace translačních vektorů jak měřených, tak referenčních. Pokud translační vektor  $\mathbf{t}$  je popsán jako  $\mathbf{t} = [x, y, z]^T$ , pak normalizovaný vektor získám takto:

$$\hat{\mathbf{t}} = \frac{\mathbf{t}}{|\mathbf{t}|} \quad (4.4)$$



Obr. 4.2: Korespondence detekované v prvním páru testovaných snímků.

### 4.2.1 Výsledky prvního testu

Kalibrační matice	rotace [°]			translace [j]		
[-]	x	y	z	x	y	z
$K_{1,1}$	-179,81	21,17	-133,77	0,037	-0,111	0,993
$K_{1,2}$	-172,96	-2,13	-175,81	-0,069	0,028	0,997
$K_{3,1}$	-174,12	-2,31	-175,82	-0,113	0,013	0,994
$K_{3,2}$	-170,57	-2,48	-175,90	-0,091	0,044	0,995

Tab. 4.3: Výsledky prvního měření

Kalibrační matice	$\Delta_{rotace}$ [°]		
[-]	x	y	z
$K_{1,1}$	186,30	40,61	134,36
$K_{1,2}$	179,45	17,31	176,40
$K_{3,1}$	180,61	17,13	176,41
$K_{3,2}$	177,06	16,96	176,49
$\sigma$ [°]	180,86	23,00	165,92

Tab. 4.4: Absolutní chyby a směrodatná odchylka rotace pro první měření.

translační vektor	translace [j]		
[-]	x	y	z
referenční	0,908	0,018	0,074
pro $\mathbf{K}_{1,1}$	0,032	-0,097	0,870
pro $\mathbf{K}_{1,2}$	-0,063	0,026	0,911
pro $\mathbf{K}_{3,1}$	-0,101	0,012	0,888
pro $\mathbf{K}_{3,2}$	-0,081	0,039	0,881
$\sigma$ [j]	0,961	0,038	0,813

Tab. 4.5: Normalizované translační vektory pro první měření.



### 4.2.2 Výsledky druhého testu

Kalibrační matice	rotace [°]			translace [j]		
[-]	x	y	z	x	y	z
$K_{1,1}$	-0,31	-2,44	-55,88	0,365	0,120	-0,923
$K_{1,2}$	5,84	1,80	-55,76	0,433	0,100	-0,896
$K_{3,1}$	7,89	3,76	-55,62	0,473	0,103	-0,875
$K_{3,2}$	7,64	1,75	-55,55	0,516	0,109	-0,850

Tab. 4.6: Výsledky druhého měření

Kalibrační matice	$\Delta_{rotace}$ [°]		
[-]	x	y	z
$K_{1,1}$	5,15	11,22	1,14
$K_{1,2}$	0,38	15,46	1,02
$K_{3,1}$	2,43	17,42	0,88
$K_{3,2}$	2,18	15,41	0,81
$\sigma$ [°]	2,54	14,88	0,96

Tab. 4.7: Absolutní chyby a směrodatná odchylka rotace pro druhé měření.

translační vektor	translace [j]		
[-]	x	y	z
referenční	0,605	0,199	0,196
pro $\mathbf{K}_{1,1}$	0,259	0,085	-0,656
pro $\mathbf{K}_{1,2}$	0,303	0,070	-0,627
pro $\mathbf{K}_{3,1}$	0,326	0,071	-0,603
pro $\mathbf{K}_{3,2}$	0,350	0,074	-0,576
$\sigma$ [j]	0,296	0,124	0,811

Tab. 4.8: Normalizované translační vektory pro druhé měření.

### 4.2.3 Výsledky třetího testu

Kalibrační matice	rotace [°]			translace [j]		
[-]	x	y	z	x	y	z
$K_{1,1}$	-3,62	-8,00	23,33	0,991	0,080	-0,107
$K_{1,2}$	-6,19	-6,08	22,50	0,995	0,063	-0,082
$K_{3,1}$	-6,02	-0,46	22,38	0,975	0,026	-0,222
$K_{3,2}$	-5,87	-1,29	22,54	0,988	0,031	-0,154

Tab. 4.9: Výsledky třetího měření

Kalibrační matice	$\Delta_{rotace}$ [°]		
[-]	x	y	z
$K_{1,1}$	3,07	4,32	1,35
$K_{1,2}$	0,50	2,40	0,52
$K_{3,1}$	0,67	3,22	0,40
$K_{3,2}$	0,82	2,39	0,56
$\sigma$ [°]	1,27	3,08	0,71

Tab. 4.10: Absolutní chyby a směrodatná odchylka rotace pro třetí měření.

translační vektor	translace [j]		
[-]	x	y	z
referenční	0,855	0,062	-0,082
pro $\mathbf{K}_{1,1}$	0,841	0,068	-0,091
pro $\mathbf{K}_{1,2}$	0,873	0,055	-0,072
pro $\mathbf{K}_{3,1}$	0,797	0,021	-0,182
pro $\mathbf{K}_{3,2}$	0,842	0,026	-0,131
$\sigma$ [j]	0,026	0,022	0,042

Tab. 4.11: Normalizované translační vektory pro třetí měření.

### 4.2.4 Srovnání

Důležitou informací ke správné analýze výsledků jsou počty detekovaných významných bodů, korespondencí i inlierů. Vzhledem k automatickému výpočtu všech jmenovaných jsem poté ještě manuálně zkontroloval správnost všech inlierů a vypočítal úspěšnost automatického výběru, jak ukazuje tabulka 4.12

test	1		2		3	
snímek	A	B	A	B	A	B
významné body	338	680	169	219	206	334
korespondence	250		124		137	
inliery	23		69		63	
ruční kontrola	12		56		61	
úspěšnost inlierů [%]	52,2		81,2		96,8	

Tab. 4.12: Počet nalezených významných bodů, korespondencí a inlierů pro jednotlivé testy.

test	$\sigma_{rotace}$ [°]			úspěšnost inlierů
[-]	x	y	z	[%]
1	180,86	23,00	165,92	52,2
2	2,54	14,88	0,96	81,2
3	1,27	3,08	0,71	96,8

Tab. 4.13: Srovnání směrodatných odchylek rotace všech testů a úspěšnosti vybrání správných inlierů.

test	$\sigma_{translace}$ [j]			úspěšnost inlierů
[-]	x	y	z	[%]
1	0,961	0,038	0,813	52,2
2	0,296	0,124	0,811	81,2
3	0,026	0,022	0,042	96,8

Tab. 4.14: Srovnání směrodatných odchylek translace všech testů a úspěšnosti vybrání správných inlierů.

## 5 ZÁVĚR

Pro dosažení cíle této práce bylo stěžejní pochopit základy epipolární geometrie. Proto byly popsány vlastnosti fundamentální a esenciální matice a způsoby získání těchto matic pomocí výpočtů opírajících se o znalost korespondencí ve dvojici snímků. Byl ukázán postup pro získání projekční matice z matice esenciální pomocí SVD. Vzhledem k tomu, že nastíněná metoda neústí v jedno řešení projekční matice, bylo představeno použití lineární triangulace jako nástroje pro výběr správného řešení. Dále je v práci uveden postup pro získání relativní orientace z vybrané projekční matice.

Díky nutnosti nalézt korespondence ve dvojici snímků byly popsány základní metody detekce a deskripce významných bodů a jejich následné párování. Vzhledem k upřednostnění přesnosti před rychlostí, byla pro detekci významných bodů po zhodnocení všech hledisek představených v kapitole 2 vybrána metoda SIFT. Čas potřebný pro výpočet nehraje větší roli, proto je hlavním aspektem počet nalezených a spárovaných bodů, a to i při nízké efektivitě metody.

Po ověření nastudované teorie z oblasti epipolární geometrie na umělých datech v programu Matlab 2016a, byly probrané postupy implementovány v jazyce C++ s použitím knihovny OpenCV v3.1 v prostředí Visual Studio 2015 Community. Na vytvořené aplikaci pak byla provedena série testů, které jsou k vidění v kapitole 4.

Testy byly prováděny na třech párech snímků (přiložených v příloze). Pro každý takový pár bylo vyfoceno 20 kalibračních snímků s kalibrační šachovnicí, pomocí kterých byly v Stereo Calibration Toolboxu v prostředí Matlab 2016a pořízeny referenční úhly rotací a směry translačních vektorů (tab.4.2), spolu s kalibračními maticemi (vzorec 4.1). Bohužel u druhé sady snímků nebylo připuštěno (Matlabem a poté manuálně) přesně 13 z pořízených 20 dvojic kalibračních snímků, proto byly vypočtené kalibrační matice z tohoto měření vyňaty z testování. Vzhledem k tomu, že zmíněný toolbox vracel pro každé měření 2 kalibrační matice, byl každý pár snímků testován pro každou z těchto matic. První testovaný pár snímků ukázal slabinu navrhnutého algoritmu. Jak ukazuje tabulka 4.12, v tomto testu byl značný počet korespondencí přiřazen nesprávně, tudíž veškerý následující výpočet musel nutně trpět velkou chybou. To také ukazuje tabulka 4.3, kde je vidět, že směrodatná odchylka dosahuje pro rotaci podél os  $x$ ,  $y$  a  $z$  postupně hodnot  $180,86^\circ$ ,  $23,00^\circ$  a  $165,92^\circ$ . Tablka 4.5 porovnávající směr translace reference a měření ukazuje, že došlo pravděpodobně k prohození translace v ose  $x$  a v ose  $z$ . V tomto případě metoda naprosto selhala. Druhý testovaný pár měl již správně přiřazených korespondencí více. S úspěšností 81% bylo proto dosaženo lepších výsledků při určování úhlů rotace. Směrodatné odchylky jednotlivých složek rotace vyšly pro osy  $x$ ,  $y$  a  $z$  postupně  $2,54^\circ$ ,  $14,88^\circ$  a  $0,96^\circ$ . V tabulce 4.8 lze bohužel znovu vypožorovat přehození

translací v osách  $\mathbf{x}$  a  $\mathbf{z}$ . Třetí test dopadl nejlépe co se korespondencí týče. Z 63 připuštěných inlierů bylo správně přiřazeno 61, což se pozitivně odrazilo na výsledcích. Směrodatné odchylky pro rotaci podle osy  $\mathbf{x}$ ,  $\mathbf{y}$  a  $\mathbf{z}$  vyšly postupně  $1,27^\circ$ ,  $3,08^\circ$  a  $0,71^\circ$ . I směrodatné odchylky normalizovaných translačních vektorů vyšly nejlepší, postupně pro posun v ose  $\mathbf{x}$ ,  $\mathbf{y}$  a  $\mathbf{z}$  nabývaly hodnot 0,026, 0,022 a 0,042 normalizované jednotky. Z celého testování lze vypožorovat zásadní vliv správného přiřazení korespondencí na budoucí výsledek. Spolu s poklesem úspěšnosti přiřazení výrazně klesá přesnost vypočtené relativní orientace. Taktéž má na výsledek částečný vliv použitá kalibrační matice, jak lze vypožorovat v průběhu celého měření. Nicméně tento vliv není tolik zásadní, jako zmiňovaná detekce.

# LITERATURA

- [1] M. GUERRERO, *A Comparative Study of Three Image Matching Algorithms: Sift, Surf and Fast* [online]. All Graduate Theses and Dissertations, Paper 1040. 2011 Dostupné z URL: <<http://digitalcommons.usu.edu/etd/1040>>.
- [2] P.M. PANCHAL, S.R. PANCHAL, S.K. SHAH, *A Comparison of SIFT and SURF* [online]. International Journal of Innovative Research in Computer and Communication Engineering. Vol.1, Issue 2, April 2013. ISSN (Online): 2320-9801
- [3] R. HARTLEY, A. ZISSERMAN, *Multiple View Geometry in Computer Vision, Second Edition*. Cambridge University Press, 2003
- [4] Univerzita Palackého v Olomouci, *Rozklad na singulární hodnoty (SVD; Singular values decomposition)* [online]. Dostupné z URL: <<http://phoenix.inf.upol.cz/~konecnja/vyuka/2013/ALS1files/svd.pdf>>.
- [5] C. HARRIS, M. STEPHENS, *A Combined Corner and Edge Detector*. Fourth Alvey Vision Conference, Manchester, UK, 1988
- [6] openCV, *Documentation* [online]. 2014 Dostupné z URL: <<http://docs.opencv.org>>.
- [7] J. SHI, C. THOMASI, *Good Features to Track* 9. IEEE Conference on Computer Vision and Pattern Recognition, Springer, 1994.
- [8] D.G. LOWE, *Distinctive Image Feature from Scale-Invariant Keypoints*. Internal Journal of Computer Vision 60:91-110, 2004.
- [9] H. BAY, A. ESS, T. TUYTELAARS, L. Van GOOL, *Speeded-Up Robust Features (SURF)* Computer Vision and Image Understanding 110, 2008
- [10] E. ROSTEN, T. DRUMMOND, *Machine learning for high-speed corner detection* Trans. Pattern Analysis and Machine Intelligence. p.430-443, (2006)
- [11] M. CALONDER, V. LEPETIT, C. STRECHA, P. FUA, *BRIEF: Binary Robust Independent Elementary Features* Conference on Computer Vision, 2010.

## SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

SIFT	Scale Invariant Feature Transform
DoG	Difference of Gaussian
LoG	Laplacian of Gaussian
SURF	Speed-Up Robust Features
FAST	Features from Accelerated Segment Test
BRIEF	Binary Robust Independent Elementary Features
SVD	Singular Value Decomposition

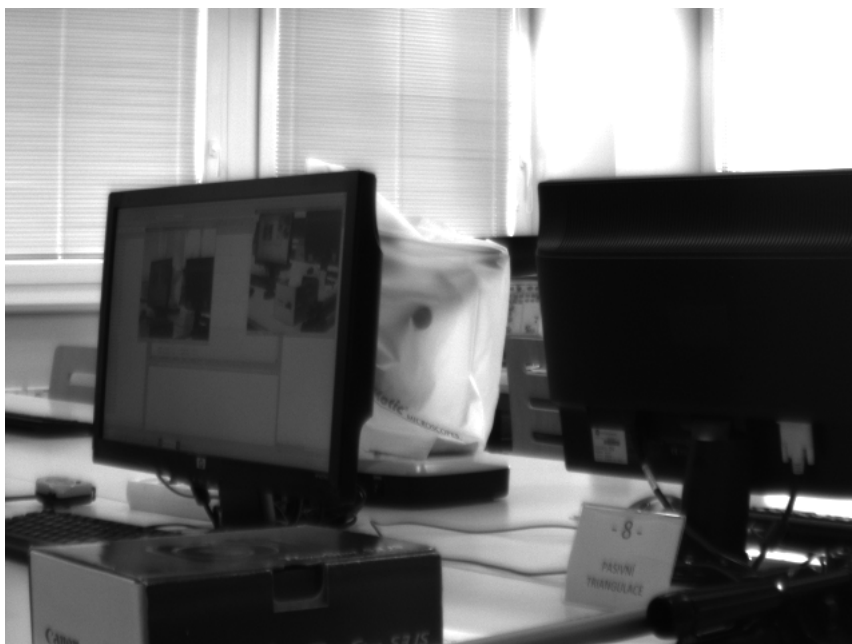
# SEZNAM PŘÍLOH

A Testové snímky

49



## A TESTOVÉ SNÍMKY



Obr. A.1: První test, snímek A.



Obr. A.2: První test, snímek B.



Obr. A.3: Druhý test, snímek A.



Obr. A.4: Druhý test, snímek B.



Obr. A.5: Třetí test, snímek A.



Obr. A.6: Třetí test, snímek B.