

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ  
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF TELECOMMUNICATIONS

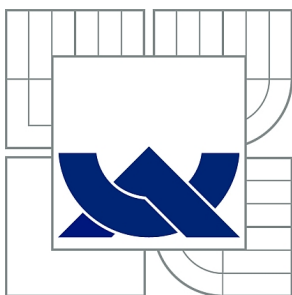
SOUSTAVA KAMER JAKO STEREOSKOPICKÝ SENZOR PRO MĚŘENÍ  
VZDÁLENOSTI V REÁLNÉM ČASE

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

AUTOR PRÁCE  
AUTHOR

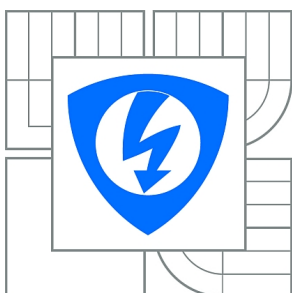
Bc. MARTIN JANEČEK

BRNO 2014



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ**

**ÚSTAV TELEKOMUNIKACÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF TELECOMMUNICATIONS

# **SOUSTAVA KAMER JAKO STEREOSKOPICKÝ SENZOR PRO MĚŘENÍ VZDÁLENOSTI V REÁLNÉM ČASE**

REAL-TIME DISTANCE MEASUREMENT WITH STEREOSCOPIIC SENSOR

**DIPLOMOVÁ PRÁCE**

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Bc. MARTIN JANEČEK**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. MARTIN HASMANDA**

BRNO 2014



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav telekomunikací

# Diplomová práce

magisterský navazující studijní obor  
Telekomunikační a informační technika

**Student:** Bc. Martin Janeček

**ID:** 109664

**Ročník:** 2

**Akademický rok:** 2013/2014

## NÁZEV TÉMATU:

**Soustava kamer jako stereoskopický senzor pro měření vzdálenosti v reálném čase**

## POKYNY PRO VYPRACOVÁNÍ:

Jako stereoskopický senzor bude v diplomové práci sloužit soustava dvou kalibrovaných kamer. Cílem studenta bude za pomoci takovéto soustavy zpracovat pořízené stereoskopické snímky do podoby hloubkové mapy. Výpočet hloubkové mapy bude z důvodu rychlosti výpočtu navržen pro GPU. Z takto vypočtené hloubkové mapy a známých kalibračních parametrů se dále určí vzdálenosti předmětů před senzorem. Pro vyšší přesnost bude uvažována hloubková mapa se sub-pixelovou přesností.

## DOPORUČENÁ LITERATURA:

[1] Andrew Hartley and Andrew Zisserman. Multiple view geometry in computer vision (2. ed.) . Cambridge University Press, 2006.

[2] BRADSKI, Gary; KAEHLER, Adrian. Learning OpenCV. [s.l.] : O'Reilly, 2009.

[3] CYGANEK, B.; SIEBERT, P., J.; An Introduction to 3D Computer Vision Techniques and Algorithms: Wiley 2010.

**Termín zadání:** 10.2.2014

**Termín odevzdání:** 30.5.2014

**Vedoucí práce:** Ing. Martin Hasmanda

**Konzultanti diplomové práce:**

**prof. Ing. Kamil Vrba, CSc.**

*Předseda oborové rady*

## UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové 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.

## **ANOTACE**

Práce ukazuje kalibraci stereoskopického senzoru. Popisuje základní metody stereokorespondence za použití knihovny OpenCV. Obsahuje výpočet disparitních map pomocí procesoru nebo grafické karty (s použitím knihovny OpenCL).

## **KLÍČOVÁ SLOVA**

Semi Global Block Matching, Block Matching, OpenCV, OpenCL, disparitní mapa, stereoskopie, kalibrace stereoskopického senzoru

## **ABSTRAKT**

Project shows calibration stereoscopic sensor. Also describes basic methods stereocorrespondence using library OpenCV. Project contains calculations of disparity maps on CPU or graphic card(using library OpenCL).

## **KEY WORDS**

Semi Global Block Matching, Block Matching, OpenCV, OpenCL, disparity map, stereovision, calibration stereoscopic sensor.

## **BIBLIOGRAFICKÁ CITACE DÍLA**

JANEČEK, M. Soustava kamer jako stereoskopický senzor pro měření vzdálenosti v reálném čase. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2014. 60 s. Vedoucí diplomové práce Ing. Martin Hasmanda.

## PROHLÁŠENÍ

Prohlašuji, že svoji Diplomovou práci na téma „**Soustava kamer jako stereoskopický senzor pro měření vzdálenosti v reálném čase**“ jsem vypracoval samostatně pod vedením vedoucího Diplomové 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 uvedené Diplomové práce dále prohlašuji, že v souvislosti s vytvořením tohoto projektu jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků souvisejících s porušením platné a účinné právní úpravy.

V Brně dne 30. května 2014

.....

podpis autora

## PODĚKOVÁNÍ

Tímto děkuji vedoucímu Diplomové práce panu Ing. Martinu Hasmandovi za cenné připomínky, rady a odborné vedení, které mně po celou dobu zpracování poskytoval.

V Brně dne 30. května 2014

.....

podpis autora

VÝZKUM POPSANÝ V TÉTO DIPLOMOVÉ PRÁCI BYL  
REALIZOVÁN V LABORATOŘÍCH PODPOŘENÝCH Z PROJEKTU  
SIX; REGISTRAČNÍ ČÍSLO CZ.1.05/2.1.00/03.0072, OPERAČNÍ  
PROGRAM VÝZKUM A VÝVOJ PRO INOVACE.

# OBSAH

<b>ÚVOD .....</b>	<b>11</b>
<b>1 POŘÍZENÍ STEREOSKOPICKÝCH OBRAZŮ .....</b>	<b>13</b>
1.1 Metoda Toe – in .....	13
1.2 Metoda Off – axis.....	14
<b>2 EPIPOLÁRNÍ GEOMETRIE.....</b>	<b>15</b>
2.1 Fundamentální matice.....	16
<b>3 KALIBRACE KAMER .....</b>	<b>19</b>
3.1 Kalibrace pomocí obrázků .....	19
3.2 Výsledky kalibrace .....	20
3.3 Parametry kamer.....	21
3.3.1 Matice kamery.....	21
3.3.2 Distorzní vektor.....	21
3.3.3 Rotační matice.....	21
3.3.4 Translační vektor.....	21
3.3.5 Rektifikační transformační matice .....	22
3.3.6 Projekční matice.....	22
3.3.7 Hloubková matice .....	22
<b>4 REKTIFIKACE STEREOSKOPICKÉHO OBRAZU .....</b>	<b>23</b>
<b>5 STEREO KORESPONDENČNÍ METODY .....</b>	<b>25</b>
5.1 Block Matching metoda .....	25
5.2 Semi Global Block Matching.....	28
5.3 Předzpracování obrazu .....	30
5.3.1 Filtr Sobel.....	30
5.3.2 Prostá normalizace .....	32
5.4 Odhad disparity se sub-pixelovou přesností .....	33
<b>6 ZPRACOVÁNÍ PROGRAMU NA GPU .....</b>	<b>35</b>
6.1 CPU vs. GPU .....	35
6.2 Paralelismus .....	36
6.3 OpenCL .....	36
6.4 CPU vs. GPU pro BlockMatching.....	38



<b>7</b>	<b>PRAKTICKÁ ČÁST.....</b>	<b>39</b>
7.1	Popis programu .....	39
7.2	Grafy – statistické hodnoty z kalibrací.....	46
7.3	Grafy – chyby měření vzdálenosti.....	49
7.4	Převod disparitní mapy z šedotónového do barevného spektra.....	50
7.5	Určení vzdálenosti objektu z disparitní mapy .....	52
7.6	Ukázky disparitních map.....	54
<b>8</b>	<b>ZÁVĚR.....</b>	<b>59</b>
<b>9</b>	<b>ZDROJE.....</b>	<b>61</b>
<b>10</b>	<b>SEZNAM PŘÍLOH NA DVD .....</b>	<b>63</b>

## SEZNAM OBRÁZKŮ

Obr. 1: Metoda Toe-in .....	13
Obr. 2: Metoda Off-axis .....	14
Obr. 3: Epipolární geometrie .....	15
Obr. 4: Epipolární geometrie – epipóly v nekonečnu.....	16
Obr. 5: Epipolární geometrie – přenášení bodu.....	17
Obr. 6: Ukázka kalibračního vzoru.....	19
Obr. 7: Radiální zkreslení.....	20
Obr. 8: Ukázka rektifikovaného snímku.....	23
Obr. 9: Diagram kalibrace .....	40
Obr. 10: Diagram matching .....	44
Obr. 11: Disparitní mapa BM (CPU) – originál .....	50
Obr. 12: Disparitní mapa BM (CPU) – upravená, 256 tónů šedi v celém spektru.....	50
Obr. 13: Disparitní mapa BM (CPU) – upravená, 256 tónů barev palety Color Jet.....	51
Obr. 14: Paleta Color Jet .....	51
Obr. 15: Určení vzdálenosti bodu.....	53
Obr. 16: Fotografie z levé a pravé kamery .....	54
Obr. 17: Rektifikovaný stereopár .....	54
Obr. 18: Anaglifické zobrazení .....	55
Obr. 19: Disparitní mapa šedotónová (CPU:BM) .....	56
Obr. 20: Disparitní mapa obarvená Color Jet (CPU:BM) .....	56
Obr. 21: Disparitní mapa šedotónová (CPU:SGBM) .....	57
Obr. 22 : Disparitní mapa obarvená Color Jet (CPU:SGBM) .....	57
Obr. 23: Disparitní mapa šedotónová (OCL:BM).....	58
Obr. 24: Disparitní mapa obarvená Color Jet (OCL:BM).....	58

## SEZNAM GRAFŮ

Graf 1: Graf chybovosti reprojekce (10cm) .....	46
Graf 2: Graf chybovosti reprojekce (15cm) .....	46
Graf 3: Graf chybovosti reprojekce (20cm) .....	47
Graf 4: Graf závislosti chyby měření na vzdálenosti předmětu (20 cm).....	49
Graf 5: Graf závislosti chyby měření na vzdálenosti předmětu (30 cm).....	49

## ÚVOD

Tématem, které jsem si zvolil v rámci semestrálního projektu a následně pro diplomovou práci, je „Soustava kamer jako stereoskopický senzor pro měření vzdálenosti v reálném čase“. Hlavním cílem diplomové práce bude zpracovat stereoskopické snímky do podoby disparitní mapy. K dosažení tohoto cíle bude nezbytné nastudovat metody pro pořizování stereoskopických obrazů za pomoci dvojice kamer a nalézt tak vhodná řešení pro zpracování těchto výsledných snímků.

V úvodní části práce se budu věnovat způsobům pořizování stereoskopických obrazů tedy způsobům snímání scény. Jde o technologie, které jsou založeny na klasickém principu vnímání člověka světa kolem něj. Uvedu a definuji dvě možné metody snímání a to Toe-In a Off-Axis. V rámci praxe budu pracovat s druhou z nich tedy s metodou Off-Axis. Dále se budu věnovat problematice epipolární geometrie, která je základem pro následnou kalibraci kamer respektive pro mapování a popis stereoskopické soustavy a pro transformaci obrazu tedy rektifikaci. V rámci této problematiky se zaměřím na význam fundamentální matice tedy mapování bodu v jednom obraze na korespondující epipolární přímku ve druhém obraze. Mapování vzájemně korespondujících bodů je důležité pro stanovení projektivity snímků. Na tuto problematiku naváže zmíněný proces kalibrace kamer a jeho popis. Tento bude základem pro získání vnitřních a vnějších parametrů kamer. Dále se v práci budu věnovat rektifikaci stereoskopického obrazu, díky které dochází ke zjednodušení úlohy při přiřazení bodu na pravém snímku bodu na levém snímku stereopáru. Zmíněný proces bude nezbytný pro správné určení vnitřních a vnějších parametrů kamer v rámci kalibrace. Pořízené stereoskopické snímky spolu se zjištěnými parametry kamer následně pomocí stereo korespondenční metody využiji pro vytvoření hloubkové neboli disparitní mapy. V neposlední řadě bude třeba vypořádat se se sub-pixelovou přesností vytvořené disparitní mapy. V závěru práce přiblížím problematiku paralelního programování grafického procesoru GPU. Uvedu obecně funkce a využití GPU a také komparaci funkčních vlastností CPU a GPU. Ke zpracování shora uváděného budu využívat funkce knihovny OpenCV. Dále zmíním standard OpenCL, díky kterému je možné paralelní programování na CPU a GPU. Tyto poznatky budou pak základem pro další zpracování již vytvořené aplikace, která bude umožňovat zpracování pořízených snímků do disparitní mapy. Po aplikování znalostí paralelního programování a funkčností OpenCL bude aplikace schopna vykreslovat barevnou disparitní mapu scény v reálném čase.

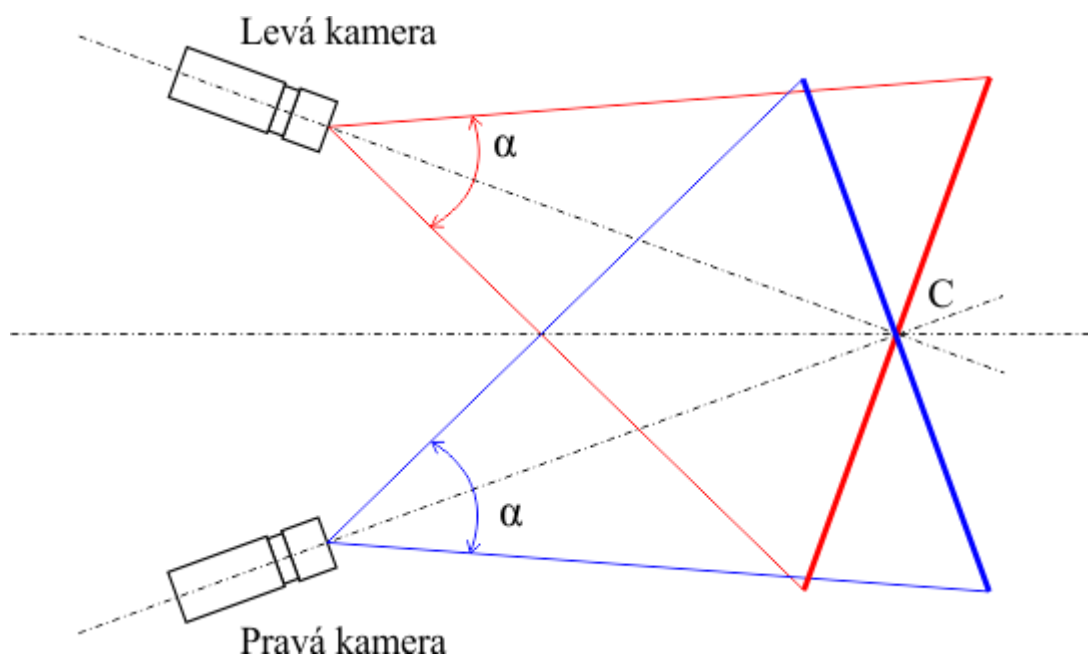
V samotném závěru uvedu práce grafy se statistickými údaji o chybách reprojekce, grafy s naměřenými chybami měření vzdálenosti, ukázky disparitních map a zhodnotím dosažené cíle diplomové práce.

# 1 POŘÍZENÍ STEREOSKOPICKÝCH OBRAZŮ

Stereoskopická technologie nám umožňuje vytvořit určitý prostorový vjem. Celá technologie je založena na klasickém principu vnímání člověka světa kolem něj. Intenzita vjemu závisí na vzájemné vzdálenosti kamer a jejich vzdálenosti od průmětny. Existuje několik metod k pořízení těchto vjemů z prostoru. Každá z těchto metod vychází z různého rozestavení kamer. Ne všechny polohy kamer jsou ke snímání obrazu vhodné. V této práci uvedu dvě metody možné pro snímání prostoru dvěma kamerami. V praxi využívám metodu Off-axis.

## 1.1 Metoda Toe – in

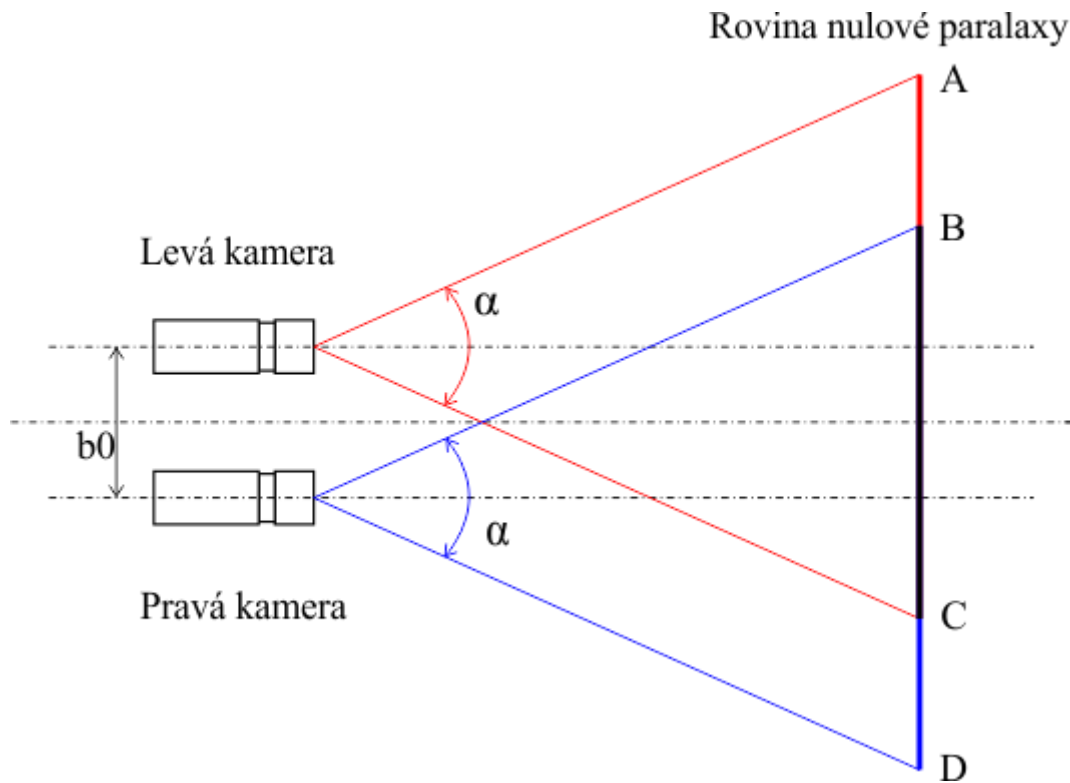
První z možných a častých metod je metoda Toe-in. Obě kamery jsou zaostřeny na totožný bod. Tato metoda není tou nejvhodnější. Nevýhodou je vznik vertikální paralaxy, která je následkem toho, že snímané body se nezobrazují v jedné rovině. Obraz tak může působit nepřírozně.[3]



Obr. 1: Metoda Toe-in

## 1.2 Metoda Off – axis

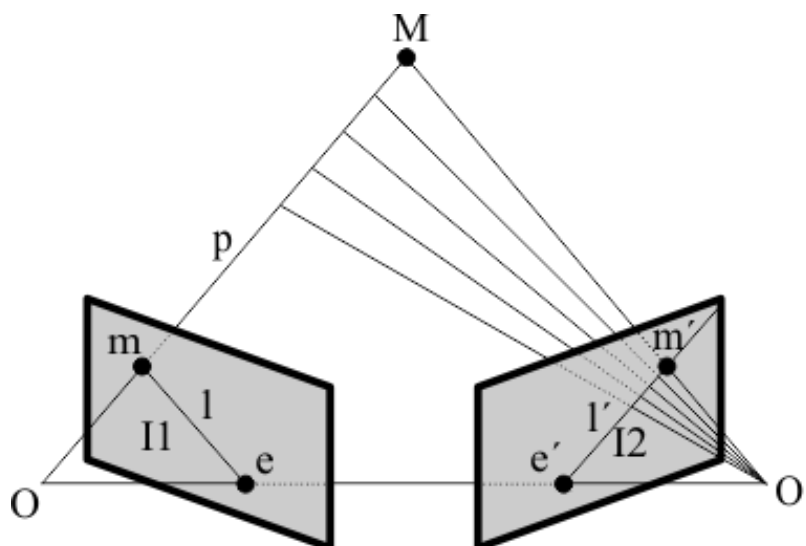
Vhodnější metodou pro snímání prostorového vjemu je metoda Off – axis. Objektivy kamer jsou nastaveny rovnoběžně a nehrozí zde proto vznik vertikální paralaxy. Obraz tedy působí přirozeně. Tato metoda je složitější v tom, že obrazy, které jsou kamerami snímány, se překrývají mezi body BC a zároveň každá z kamer snímá část obrazu, kterou druhá nesnímá.[1]



Obr. 2: Metoda Off-axis

## 2 EPIPOLÁRNÍ GEOMETRIE

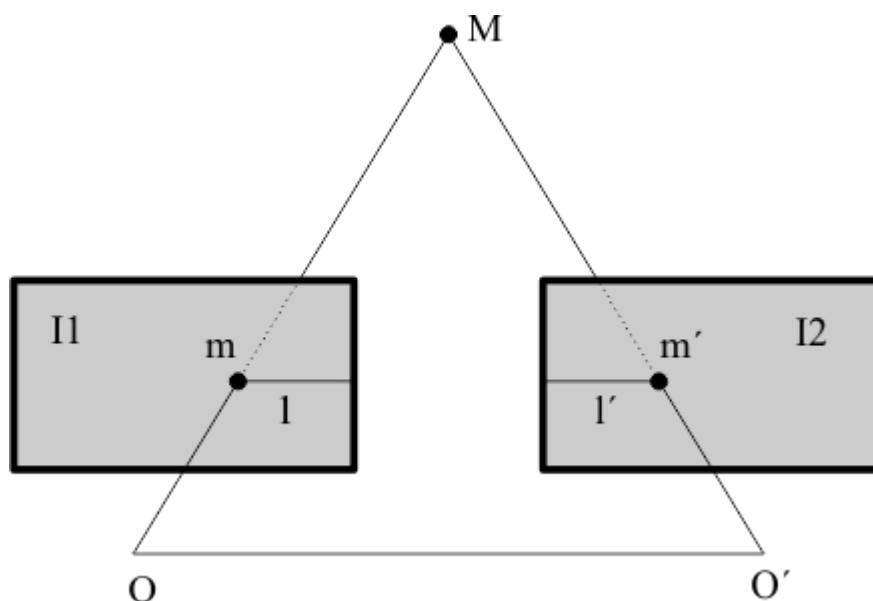
Epipolární geometrie mapuje a popisuje stereoskopickou soustavu. Tou je uspořádání protínajících se rovin, jejichž průsečíkem je báze. Bázi je přímka, která spojuje středy kamer. Prostřednictvím epipolární geometrie se mapují body z jednoho snímku, kterým odpovídají body na přímce ve druhém snímku. Na základě toho pak tedy zmapujeme body, které korespondují v obou snímcích stereo páru. Je zřejmé, že obrazy si nemusí odpovídat absolutně z důvodu odlišnosti scén. Cílem je tedy zmapovat dostatečné množství takovýchto bodů, které korespondují v obou snímcích a nalézt jejich odlišnosti. Získáme tak základ pro nalezení fundamentální matice. K dosažení cíle je rozhodující umístění obou kamer a jejich vnitřní parametry.[4]



Obr. 3: Epipolární geometrie

Na shora uvedeném obrázku je zobrazen princip epipolární geometrie. Středů kamer respektive body  $O$  a  $O'$  spojuje přímka shora nazvána jako báze. Středem kamery je bod středového promítání. Na obrázku je zobrazeno promítání bodu  $M$  do obrazové roviny  $I1$  a  $I2$ . V rámci roviny  $I1$  je bod  $M$  označen jako  $m$  a v rámci roviny  $I2$  jako  $m'$ . Všechny uvedené body ( $O$ ,  $O'$ ,  $M$ ,  $m$ ,  $m'$ ) leží v jedné epipolární rovině. Tato skutečnost je důležitá pro zpracování stereopáru na základě korespondencí. Takovýchto rovin existuje nekonečně mnoho. Soubor takovýchto rovin je nazýván jako epipolární svazek. Průsečík epipolární roviny s rovinou obrazovou označujeme jako epipolární přímku. Body  $e$  a  $e'$  jsou

označovány jako epipóly. Jde o body, které získáme protnutím báze s rovinou obrazu. Na níže uvedeném obrázku je zřejmé, že parametry kamer mohou být nastaveny tak, že epipolární přímky budou rovnoběžné s bází a epipóly budou ležet v nekonečnu. Na shora uvedeném obrázku je zřejmé, že všechny epipolární přímky budou procházet epipólem. Na níže uvedeném obrázku tuto skutečnost tvrdit nemůžeme.



Obr. 4: Epipolární geometrie – epipóly v nekonečnu

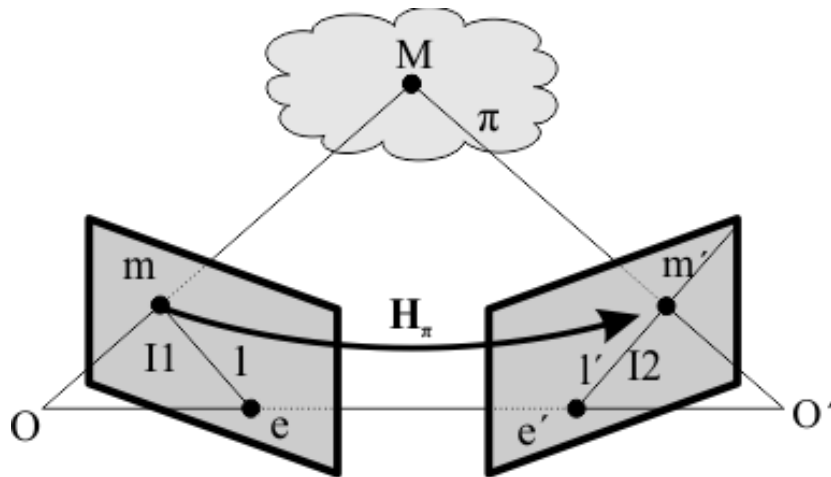
## 2.1 Fundamentální matice

Fundamentální matice je mapování bodu v jednom obraze na korespondující epipolární přímku ve druhém obraze. Jde o vyjádření základních vlastností systému. Je základem pro matematický výpočet epipolární geometrie. Pro zjištění je třeba znát umístění kamer a jejich vnitřní parametry. Na základě fundamentální matice tedy můžeme dát do souvislosti dva obrazy jednoho bodu v jednom obraze na korespondující epipolární přímku v obraze druhém.

Existuje několik možností, jak fundamentální matici určit. Záleží na tom, zda jsou známy projekční matice či nejsou. Metody, které se využívají, jsou většinou nazývány podle počtu korespondujících bodů, které se používají k výpočtu.

Pro přesné určení fundamentální matice je důležité, aby korespondující páry byly stanoveny s velkou přesností, jinak by mohly být výpočty mylné.





Obr. 5: Epipolární geometrie – přenášení bodu

Jak uvedeno shora, fundamentální matice je mapování bodu v jednom obraze na korespondující epipolární přímku ve druhém obraze. Na uvedeném obrázku je toto zřejmé. Paprsek respektive přímka vycházející ze středu kamery  $O$  a procházející bodem  $m$  se protíná s danou rovinou v bodě  $M$ . Bod  $M$  je pak v rámci přesunu v rovině  $\pi$  promítnut do bodu  $m'$ , který leží na epipolární přímce  $l'$ . Body  $m$  a  $m'$  jsou tedy obrazy 3D bodu  $M$ . Každý další bod  $m_i$  prvního obrazu a jemu korespondující bod  $m_i'$  budou ve vztahu vzájemné ekvivalence. Existuje tedy 2D homografie  $H_\pi$  mapující každý  $m_i$  na  $m_i'$ , což znamená, že mapováním vzájemně si odpovídajících bodů stanovíme projektivitu snímků a to i kdybychom neznali vnitřní parametry kamery.

Máme-li dán bod  $x'$  a epipolární přímku  $l'$  procházející tímto bodem a epipólem  $e'$ , lze stanovit parametry epipolární přímky vztahem

$$l' = e' \times m' = [e']_x m'. \quad (1)$$

Protože bod  $m'$  může být stanoven ze vztahu

$$m' = H_\pi m, \quad (2)$$

lze předchozí vztah zapsat jako:

$$l' = [e']_x H_\pi m = Fm. \quad (3)$$

Ze vztahu 3 je následně definována fundamentální matice, tato definice je vyjádřena vztahem:

$$F = [e']_x H_\pi \quad (4)$$

$H_\pi$  je mapování jednoho bodu do druhého bodu v libovolné rovině.

Vztah 4 geometricky reprezentuje  $F$  mapování z dvourozměrného prostoru (obrazový prostor  $I_1$ ) do jednorozměrného prostoru (epipolární přímka  $l_1$ ).

Ze vztahu vyjadřující fakt, že bod  $m'$  leží na přímce  $l'$  ( $l'^T m' = 0$ ) a z rovnice 3 vyplývá vztah:

$$m'^T F m = 0 \quad (5)$$

Nalezením epipolárních přímek se velice usnadní hledání korespondujících si bodů v obrazech. Není totiž nutné hledat odpovídající bod v celém obraze, ale jen v blízkém okolí epipolární přímky. Po stanovení fundamentální matice a mapovacích matic je možné určit polohu bodu v prostoru.[6]

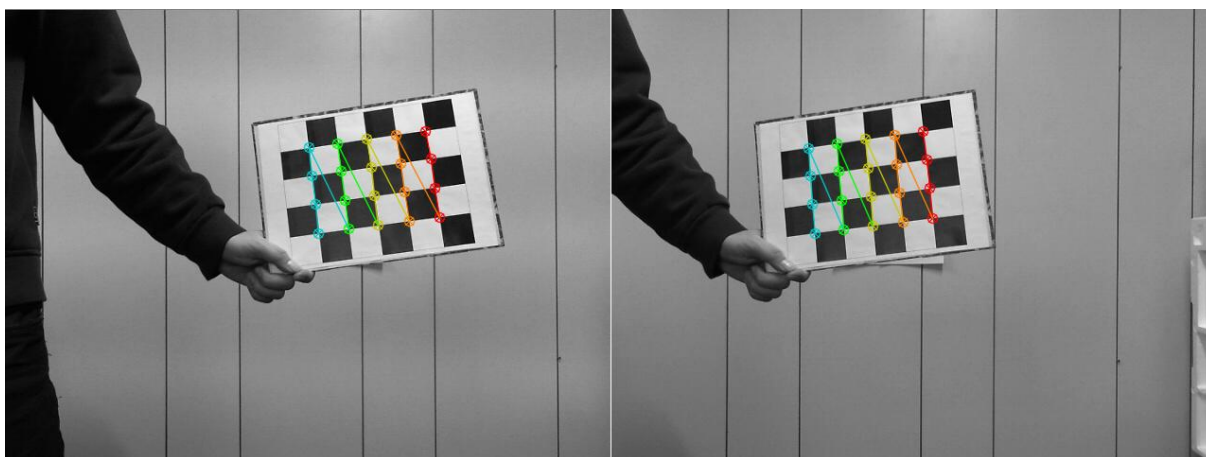
### 3 KALIBRACE KAMER

Kalibrace je proces, při kterém získáváme vnitřní a vnější parametry kamer potřebné pro další použití. V této práci používám pro snímání scény dvě web-kamery. Kalibrace se provádí tak, že aparátem nasnímáme scénu se známými obrazcovými vzory a následně provedeme analýzu získaných stereografických obrazů.

#### 3.1 Kalibrace pomocí obrazců

V práci bude využívána knihovna pro zpracování obrazu OpenCV. Je to open source knihovna s řadou funkcí pro zpracování obrazových dat. Ta je vydána pod licencí BSD, která umožňuje volné šíření a použití jednotlivých funkcí knihovny. Tento multiplatformní software je napsaný v jazyce C a C++, původně vyvinutý společností Intel.

Princip kalibrace dvojice kamer spočívá v nasnímání určitého množství stereo-párů, tedy dvojic fotek totožné scény snímané v totožný okamžik dvojicí kamer. Při použití výše zmiňované knihovny pro zpracování obrazu, je doporučeno použít obrazce šachovnicového vzoru, které jsou snadno viditelné ve scéně. Pro získání prostorových bodů potřebných ke kalibraci, je potřebné, aby tyto obrazce byly v různých úhlech natočení a vzdálenostech vůči kamerám. Tento vzor vytvoří body v místech křížení čtverců. Pokud je tento vzor na pevné a rovné ploše, lze předpokládat, že body jsou jednotlivě umístěny ve vzdálenosti  $Z=0$  s osami X a Y spojenými mřížkou šachovnice. Na obrázku je ukázka snímků kalibračního vzoru použitých v této práci.[2]

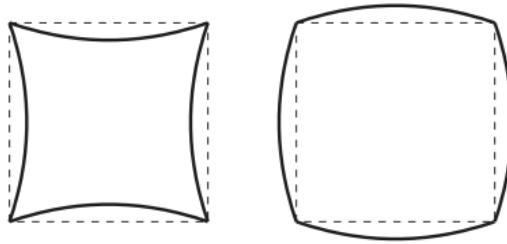


Obr. 6: Ukázka kalibračního vzoru

### 3.2 Výsledky kalibrace

Je možné použít několik metod pro určení kalibrační matice kamer. V této práci používám metodu, pomocí které lze získat intrinstické a excentrické parametry soustavy kamer a také parametry radiálního zkreslení.

Radiální zkreslení je odchylka, která nastává zkreslením čoček během počáteční projekce světla do obrazové roviny. Na obrázku 7 je znázorněno vlevo poduškovité a vpravo soudkovité radiální zkreslení.



Obr. 7: Radiální zkreslení

Zkreslení je možné eliminovat pomocí kvalitnějšího objektivu, nebo softwarově pomocí kalibrace.

Ve funkci `stereoCalibrate` je možné zadávat kritéria výpočtů, jako je maximální počet iterací. [5]

### 3.3 Parametry kamer

#### 3.3.1 Matice kamery

$$\begin{bmatrix} f_x^{(j)} & 0 & c_x^{(j)} \\ 0 & f_y^{(j)} & c_y^{(j)} \\ 0 & 0 & 1 \end{bmatrix} \quad (6)$$

$f_x, f_y$  - ohnisková vzdálenost v ose x a ose y (v pixelech)

$c_x, c_y$  - pozice středu obrazu

#### 3.3.2 Distorzní vektor

$$(k_1, k_2, p_1, p_2, [k_3[k_4, k_5, k_6]]) \quad (7)$$

$k_1 - k_3$  - definují radiální zkreslení

#### 3.3.3 Rotační matice

Rotační matice  $\mathbf{R}$  určuje rotaci pravé kamery vůči levé kameře. Levá kamera nebude rotována.

#### 3.3.4 Translační vektor

$$t = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (8)$$

Translační vektor  $\mathbf{t}$  určuje posunutí pravé kamery vůči levé kameře.

### 3.3.5 Rektifikační transformační matice

Určuje rotaci obrazu potřebnou pro rektifikaci (rotace okolo všech os). Tato matice je výstupem rektifikace.

### 3.3.6 Projekční matice

$$P1 = \begin{bmatrix} f & 0 & cx_1 & 0 \\ 0 & f & cy & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (9)$$

$$P2 = \begin{bmatrix} f & 0 & cx_2 & T_x * f \\ 0 & f & cy & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (10)$$

Sjednocené (projekční) matice kamer tak, aby byly rektifikované. Matice **P1** a **P2** jsou výstupem rektifikace.

### 3.3.7 Hloubková matice

$$Q = \begin{bmatrix} 1 & 0 & 0 & -c_{x1} \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 0 & f \\ 0 & 0 & \frac{1}{|T|} & 0 \end{bmatrix} \quad (11)$$

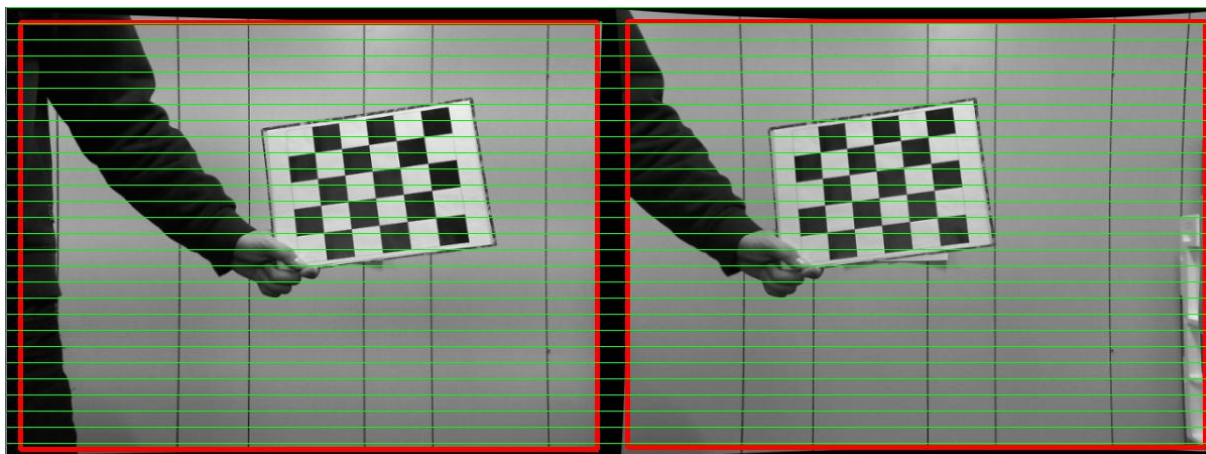
Hloubková matice **Q** přepočítává perspektivu a je výstupem rektifikace.

## 4 REKTIKACE STEREOSKOPICKÉHO OBRAZU

Rektifikace je obecně jakákoliv geometrická transformace obrazu. V případě stereo snímků je to taková transformace, díky které dochází ke zjednodušení úlohy přiřazení bodu na pravém snímku bodu na levém snímku stereo páru.

Pro pořízení scény je nutné správné nastavení kamer, podle metody Off-axis. Proto kamery by měly tvořit přesnou paralelní soustavu, tedy být vůči sobě posunuty pouze v horizontálním směru. To je však v reálném prostředí a připevněním kamer nedosažitelné. A i když jsou kamery od jednoho výrobce a měly by mít stejné parametry, mohou se tyto parametry ve skutečnosti mírně lišit. Vzhledem k těmto výše zmíněným okolnostem se přistupuje k stereoskopické rektifikaci obrazu.

Transformace obrazu je taková, aby korespondující body na snímcích stereo páru byly ve stejné výšce, tedy měly stejné hodnoty ypsilonové souřadnice v obrazové rovině (snímky budou koplánární). Rektifikace se bude provádět u stereo snímků získaných ze soustavy takových kamer, které mají odchylky v rovnoběžnosti optických os, nebo došlo k rotaci kamery, nebo v případě, že kamery nejsou ve stejné vzdálenosti (jedna z nich je předsunuta). Cílem takové rektifikace je matematicky přemístit zobrazovací roviny kamer tak, aby byly ve stejné rovině.[4]



Obr. 8: Ukázka rektifikovaného snímku

Rektifikace může být provedena tak, že se jeden snímek promítne do obrazové roviny snímku druhého.

Rektifikace se provádí `StereoRectify`, kde vstupními parametry jsou prostorové souřadnice křížů šachovnice a souřadnice těchto bodů na snímcích. Výsledkem této funkce

jsou matice rotace ( $\mathbf{R}_1$ ,  $\mathbf{R}_2$ ), nové projekční matice ( $\mathbf{P}_1$ ,  $\mathbf{P}_2$ ) obou kamer a hloubkovou matici ( $\mathbf{Q}$ ).

Ze získaných koeficientů zkreslení, matic kamer, vzájemné rotace a translace se vypočítá funkce `StereoRectify` rektifikační matice a projekční matice v rektifikovaném souřadném systému pro obě kamery.[2]



## 5 STEREO KORESPONDENČNÍ METODY

Stereovize je důležitou součástí pro monitorovací systémy, navigace a robotiku. Takové systémy mohou mít náročné požadavky na zpracování v reálném čase, a proto musí být funkce stereo korespondence navrženy tak, aby běžely rychle. A právě proto nejsou vyrovnávací paměti součástí funkcí OpenCV, ale jsou drženy vedle, aby si je mohl tvůrce systému řídit podle potřeby.

### 5.1 Block Matching metoda

Stereo korespondence, respektive zobrazení odpovídající 3D bodu ve dvou různých zobrazeních kamer lze vypočítat pouze ve vizuálních oblastech, ve kterých se zobrazení obou kamer překrývají. To je jeden z důvodů proč budete mít tendenci k dosažení nejlepších výsledků aranžovat vaše kamery blízko a rovnoběžně tak jak je to možné, alespoň dokud se nestanete odborníkem na stereo-vidění. Poté co víme fyzické parametry kamer nebo velikosti objektů na scéně, můžeme odvodit měření hloubky triangulované disparity.

$$d = x^l - x^r \text{ (nebo } d = x^l - x^r - (c_x^{left} - c_x^{right}) \text{)} \quad (12)$$

V případě, že se paprsky protínají v konečné vzdálenosti mezi korespondujícími body ve dvou různých zobrazeních kamer.

Rychlý a účinný block-matching algoritmus `cvFindStereoCorrespondenceBM()`, je podobný tomu, který vyvinul Kurt K. Funguje to tak, že pomocí malých oken (tzv. SAD – Sum Absolut of Differnce) součtů absolutních rozdílů, najde odpovídající body mezi levými a pravými rektifikovanými obrazy. Tento algoritmus nachází pouze silně odpovídající si body mezi dvěma obrazy. Čím více je scéna strukturovaná, tím více bodů bude mít vypočítanou hloubku. BM má tři kroky, které pracují s nezkreslenými rektifikovanými obrazy stereopáru.

- a) Normalizace obrazového jasu a textury
- b) Hledání korespondencí podél vodorovných epipolárních přímek za použití SAD oken
- c) Postfiltering - eliminuje špatné korespondence

Ad a) Rozebráno v kapitole 5.3 Předzpracování obrazu.

Ad b) Korespondence se vypočítá za použití SAD oken. Parametrem `SADWindowSize` se nastaví délka hrany těchto oken, na které se zpracováváný obraz rozdělí. Pro každý specifický tvar v levém obrazu hledáme odpovídající řádek v pravém obrazu. Po rektifikaci si řádky v obrázcích odpovídají. Takže odpovídající umístění v pravém obrazu musí být ve stejném řádku jako v levém obraze. Takto odpovídající umístění lze nalézt v případě dostatečně výrazné struktury obrazu. Pokud má levý bod hledané struktury koordináty  $x_0$ ,  $y_0$ , potom pro paralelně umístěné kamery, musí být objevena shoda (pokud bude existovat) na stejném řádku na pozici  $x_0$ , nebo vlevo od ní.

První parametr, který řídí matching je `minDisparity`. Určuje pozici, odkud se začnou hledat disparity. Výchozí hodnota je 0. Disparity jsou hledány přes tolik pixelů, jako určuje `numberOfDisparities`. Výchozí hodnota je 64 pixelů. Disparity se nehledají po pixelech, ale na jejich částech, jejichž počet určí parametr `subPixelDisparities`. Výchozí hodnota je 16.

Snížení hodnoty parametru `numberOfDisparities` může pomoci snížení času zpracování, zkrácením vzdálenosti, na které se hledá disparita v rámci epipoláry. Velké disparity umožňují hledání blízkých předmětů.

Parametry `minDisparity` a `numberOfDisparities` určují horopter, což je trojrozměrný výseč prostoru před kamerami. V této oblasti, horopteru, hledá stereo algoritmus shody. Všechno mimo tuto oblast je označeno jako „díra“ protože tam nelze spočítat vzdálenost od kamery. Oblast horopteru se zvětšuje snížením vzdálenosti mezi kamerami nebo zmenšením ohniskové vzdálenosti kamer nebo zvětšením `numberOfDisparities` nebo použitím kamer s větším rozlišením.

Korespondující body v rámci horopteru mají vnitřní omezení, nazývané omezení pořadí, které jednoduše říká, že pořadí charakteristických struktur se nemůže měnit z levého do pravého obrázku. Kvůli šumu může některá charakteristická struktura chybět na pravém obrazu, i když na levém je. Ale pořadí charakteristických struktur zůstává pořád stejné. Podobně mohou existovat charakteristické struktury na pravém obrazu, které nelze najít na levém, tzv. vsuvky (insertions). Avšak ani ty nemohly změnit pořadí charakteristických struktur, pouze je mohou rozšířit.

Ad c) Postfiltering má za úkol odstranit špatně detekované shody. Většina shod při průřezu po epipoláře tvoří centrální vrchol obklopený na obou stranách strmě svažujícími se křivkami. K odstranění špatně detekované shody se použije následující funkce řízená parametrem `uniqueRatio`. Výchozí hodnota je 12. Pro špatně detekované shody platí

$$uniquessRatio > (match\_val - min\_match) / min\_match$$

`match_val` je hodnota v bodě shody.

`min_match` je minimální hodnota v kontrolované oblasti

Pro ujištění, že oblast je dostatečně strukturovaná a nezkresloval ji náhodný šum, používá parametr `textureThreshold`. SAD okno ho využívá pro vyhodnocení. Pokud je průřez po epipoláře v aktuálním okně pod touto hodnotou, nehledá se v daném okně žádná shoda. Výchozí hodnota je 12.

Na závěr, základní blokový matching má problém blízko hranic objektu, protože vyhledávací okno vidí na jedné straně popředí a na druhé straně pozadí objektu. Díky tomu se na hranicích objektu detekovány shody, tzv. `speckle` (zrnitá oblast). Těmto shodám se předchází nastavením parametru `speckleWindowSize` a `speckleRange`. `speckleWindowSize` určuje velikost okna, které hledá okraje objektu. Výchozí hodnota je 9, což znamená čtverec 9x9 pixelů. Rozdíl minimální a maximální disparity uvnitř tohoto okna musí být menší nebo roven parametru `speckleRange`, jinak jsou všechny disparity z okna ignorovány. Výchozí hodnota je 4.

Disparitní mapa se počítá z levého do pravého obrázku. Pokud je hodnota `disp12MaxDiff` nastavena na větší než -1, tak se provádí ještě výpočet z pravého do levého. Projdou se všechny spočítané disparity, a pokud jsou rozdíly mezi nimi větší než tento parametr, tak jsou ignorovány.[2]

## 5.2 Semi Global Block Matching

SGBM (Semi Global Block Matching) algoritmus rozšiřuje obyčejný BM (Block Matching) o více řádkové hledání disparity. Z toho vyplívají jeho hlavní negativa a pozitiva. Dosahuje sice lepších výsledků, ale je výrazně pomalejší a výrazně náročnější na paměť. Je velmi dobře paralelizovatelný.

### 1. Výpočet pixelové korespondence

Pro všechny body na zkoumané epipoláře se spočítají korespondence pro všechny potencionální nezáporné disparity. Spočítají se výdaje pro pixelovou korespondenci pro každý bod a pro všechny nezáporné disparity z definovaného rozsahu.

$$C[x, y, d] = |L(x) - R(x - d)| \quad (13)$$

$C$  - korespondence,

$L$  - hodnota bodu pro levý obrázek,

$R$  - hodnota bodu pro pravý obrázek,

$d$  - disparita.

### 2. Hledání cest

SGBM spočítá váhu cesty, což je minimum korespondencí bodů, které jsou od počítaného bodu daným směrem, a váhy dané cesty před tím než dospěla k danému bodu podle vzorečku níže. Většinou se počítá pro 8 směrů.

$$L_r(p, d) = C(p, d) + \min \begin{cases} L_r(p - r, d), \\ L_r(p - r, d + 1) + P_1, \\ L_r(p - r, d - 1) + P_1 \\ \min_i L_r(p - r, i) + P_2 \end{cases} \quad (14)$$

$P_1, P_2$  - konstanty pomáhající odstranění nespojitosti,

$L_r$  - váha cesty (nejmenší rozdíl),

$C$  - korespondence.

### 3. Disparita

Samotná disparita se pak určí jako cesta s minimální vahou.

Pozn.: OpenCV umožňuje spustit SGBM pouze na CPU. Parametr FullDP vynucuje výpočet výdajů ve všech osmi cestách, jinak je vypočítáno pouze 5 cest.

Stejně jako u BM, OpenCV umožňuje použít XSobel filtr. [10]

## 5.3 Předzpracování obrazu

První fází algoritmu matching je předzpracování obrazu.

Během ní jsou ve vstupních obrazech snižovány rozdíly jasu a zvýrazňovány struktury. Hodnota se upraví výstupem funkce, která potřebuje hodnoty z jeho okolí (oblasti) o velikosti 5x5, 7x7, 9x9, ... až 21x21 bodů. Velikost této oblasti určuje parametr `preFilterSize`.

$$\min[\max(X - I_{cap}), I_{cap}] \quad (15)$$

$I_{cap}$  - parametr `preFilterCap` (výchozí hodnota je 30),

$X$  - proměnná určená druhem předzpracování.

### 5.3.1 Filtr Sobel

Aproximace druhé derivace na nespojitém prostoru. Výsledek je korespondující vektor gradiendu nebo norma tohoto vektoru.

Filtr Sobel se běžně používá pro detekci hran.

Aproximace se provádí násobením bodu a jeho okolí maticí pro horizontální nebo vertikální směr. Matice  $G_x$  pro horizontální a matice  $G_y$  pro vertikální aproximaci.

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad (16)$$

$$G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (17)$$

Tyto matice jsou základní, ale lze generovat i větší matice.

Celková aproximace  $\mathbf{G}$  se spočítá:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * \begin{bmatrix} A_{i-1,i-1} & A_{i,i-1} & A_{i+1,i-1} \\ A_{i-1,i} & A_{i,i} & A_{i+1,i} \\ A_{i-1,i+1} & A_{i,i+1} & A_{i+1,i+1} \end{bmatrix} \quad (18)$$

$$G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \begin{bmatrix} A_{i-1,i-1} & A_{i,i-1} & A_{i+1,i-1} \\ A_{i-1,i} & A_{i,i} & A_{i+1,i} \\ A_{i-1,i+1} & A_{i,i+1} & A_{i+1,i+1} \end{bmatrix} \quad (19)$$

$$G = \sqrt{G_x^2 + G_y^2} \quad (20)$$

Sobelovy operátory mohou být větší než 3x3, např.:

$$\begin{bmatrix} +1 & +4 & +7 & +4 & +1 \\ +2 & +10 & +17 & +10 & +2 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & -10 & -17 & -10 & -2 \\ -1 & -4 & -7 & -4 & -1 \end{bmatrix} \quad (21)$$

$$\begin{bmatrix} +1 & +4 & +9 & +13 & +9 & +4 & +1 \\ +3 & +11 & +26 & +34 & +26 & +11 & +3 \\ +3 & +13 & +30 & +40 & +30 & +13 & +3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -3 & -13 & -30 & -40 & -30 & -13 & -3 \\ -3 & -11 & -26 & -34 & -26 & -11 & -3 \\ -1 & -4 & -9 & -13 & -9 & -4 & -1 \end{bmatrix} \quad (22)$$

V BlockMatchingu se používá jen horizontální matice, protože se prochází přes epipoláry. [13]

Základní vzorec pro předzpracování obrazu:

$$\min[\max(X - I_{cap}), I_{cap}] \quad (23)$$

je upraven tak, že  $X$  je nahrazeno  $G_x$ ,

$$\min[\max(G_x - I_{cap}), I_{cap}] \quad (24)$$

$I_{cap}$  - parametr `preFitlerCap` (výchozí hodnota je 30).

### 5.3.2 Prostá normalizace

Upravuje hodnotu bodu na základě průměru hodnot jeho okolí.

Základní vzorec pro předzpracování obrazu:

$$\min[\max(X - I_{cap}), I_{cap}] \quad (25)$$

je upraven tak, že  $X$  je nahrazeno  $I_c - \bar{I}$

$$\min[\max(I_c - \bar{I} - I_{cap}), I_{cap}] \quad (26)$$

$I_{cap}$  - parametr `preFitlerCap` (výchozí hodnota je 30),

$\bar{I}$  - průměr spočítaný z bodů v rámci této oblasti,

$I_c$  - hodnota, na kterou se přepočítá hodnota aktuálního bodu. [2]

.



## 5.4 Odhad disparity se sub-pixelovou přesností

Disparity jsou spočítány jako rozdíly korespondencí obrázků a jsou zarovnány na celá čísla. U vzdálenějších objektů nebo kamer s menším rozlišením, dochází k tomu, že disparita daného bodu nemůže být vždy celočíselná.

A naopak u objektů, které jsou skoro rovnoběžné s kolmicí na osu  $z$ , z pohledu kamer, dochází k viditelným skokům, kde je nespojitost disparit dobře vidět.

K odstranění těchto neduhů se používá právě subpixelová disparita. Je to přepočítání aktuálních hodnot disparit na základě už spočítaných disparit a jejich proložení polynomem obvykle druhého stupně. Pro dané okno (velikost 5 až 11 bodů) se po jeho proložení najde minimum. To určuje novou hodnotu disparity ve středovém bodě okna s přesností, kterou si nově určíme (pro zpracování na CPU platí, že zaokrouhluje nové hodnoty na šestnáctiny disparity).

$$ad_x^2 + bd_x + c = v_x \quad (27)$$

Derivace (hledáme minimum):

$$2ad_x + b = 0 \quad (28)$$

$$d_x = \frac{-b}{2a} \quad (29)$$

Vytvoření soustavy rovnic pro tři okolní body

$$\begin{cases} ad_{i-1}^2 + bd_{i-1} + c = m_{i-1} \\ ad_i^2 + bd_i + c = m_i \\ ad_{i+1}^2 + bd_{i+1} + c = m_{i+1} \end{cases} \quad (30)$$

Aplikujieme Kramerovo pravidlo a zároveň posuneme střed systému koordinátů do bodu  $[d_i, 0]$ , z toho vyplývá:

$$\frac{b'}{a'} = - \frac{\begin{vmatrix} 1 & m_{i-1} & 1 \\ 0 & m_i & 1 \\ 1 & m_{i+1} & 1 \end{vmatrix}}{\begin{vmatrix} -1 & m_{i-1} & 1 \\ 0 & m_i & 1 \\ 1 & m_{i+1} & 1 \end{vmatrix}} = - \frac{m_{i-1} - m_{i+1}}{m_{i-1} - 2m_i + m_{i+1}} \quad (31)$$

Dosadíme zpět do rovnice výpočtu  $d_x$  a přičteme posunutí z minulého kroku.

$$d_x = d_i + \frac{m_i - m_{i+1}}{2(m_{i-1} - 2m_i + m_{i+1})} \quad (32)$$

Tím získáme nové minimum. [11]

## 6 ZPRACOVÁNÍ PROGRAMU NA GPU

Tato část práce bude věnována přiblížení problematiky paralelního programování na GPU a obecně jeho funkcím a využití.

GPU respektive grafický procesor (graphic processing unit) je procesor, který zajišťuje promítání dat uložených v operační paměti na zobrazovacím zařízení. Výkonné grafické procesory se v současné době využívají i k jiným výpočtům, než jsou výpočty nutné pro zobrazování dat. Grafický procesor je používán v mnoha zařízeních. Jeho konstrukce je závislá na konkrétních potřebách a požadovaném výkonu.

V osobním počítači může být GPU jako rozšiřující grafická karta, nebo jako grafická karta integrována na základní desce počítače v podobě integrovaného grafického čipu, nebo je GPU integrován do CPU, jako APU.

Použitím paralelního programování a následujícím zpracování těchto programů na GPU, lze několika násobně zkrátit dobu náročných a dlouhotrvajících výpočtů.[8]

### 6.1 CPU vs. GPU

CPU a GPU byly postupem času vyvíjeny každý pro jiný účel. CPU je využíváno jako výpočetní jednotka a pomocí přerušení se stará o jiná zařízení. Má za cíl zpracovat pouze jednu informaci a to co nejrychleji.

GPU jakožto specializovaný procesor je přizpůsoben ke zrychlování vykreslování grafiky, takže musí být schopen zpracovat operaci s velkým množstvím informací za co nejkratší dobu. Je optimalizován pro výpočty s čísly s plovoucí desetinou čárkou.

GPU poskytuje velké zvýšení výkonu v určitých operacích, obsahuje velké množství SIMD procesorů (pro jednu instrukci je několik datových toků) a podpora pro paralelismus je zabudována v hardware. CPU sice poskytuje uspokojivý výkon ve všech operacích, ale nemá zase takovou podporu pro paralelismus.

Grafické karty se dlouhou dobu používali pouze pro zobrazování grafického obsahu. Postupným vývojem se dopracovali k tomu, že je dnes možné je používat i k jiným operacím. Tato nová schopnost grafických karet se označuje jako GPGPU neboli general-purpose computing on graphics processing unit (obecné výpočty na grafických kartách). Snahou je zpravidla nahradit CPU u výpočtů, kde je GPU lepší volbou.[7]

## 6.2 Paralelismus

Výpočetní operace lze řešit sériově nebo paralelně. Sériové řešení výpočtu je složeno ze samostatných kroků. Každý z nich se provádí po dokončení předcházejícího. Většina programů používaných na osobních počítačích funguje na principu sériového zpracování. Programy se skládají ze sady instrukcí, kdy je jedna po druhé zpracovávána na CPU. Druhým způsobem zpracování programu je paralelní řešení výpočtu. V tomto případě je řešený program rozdělen na části a některé instrukce jsou prováděny ve stejnou dobu, tedy paralelně. Jinak řečeno, pro paralelní výpočet je ve stejnou dobu použito vícero výpočetních zdrojů.

Aplikováním paralelních výpočtů je možné zlepšit výkon mnoha programů. Například při násobení matic se může urychlit doba výpočtu. Jednotlivé operace násobení matic jsou totiž na sebe nezávislé, a proto mohou být provedeny současně. Avšak výpočty, ve kterých jsou jednotlivé kroky na sebe vázány, nebudou pro paralelní výpočet vhodné.

Je více možností k vytvoření paralelního výpočetního prostředí. Je možné využít více jádrový procesor, výpočetní jednotky v rámci jednoho počítače nebo sadu počítačů propojenou počítačovou sítí. Různé techniky mohou být kombinovány. V této práci operuji s paralelizmem v rámci jednoho počítače, s využitím CPU, GPU a standardu OpenCL.

## 6.3 OpenCL

OpenCL (Open Computing Language) je otevřený standard, díky kterému je možné paralelní programování různorodých počítačových systémů. Podporuje velkou škálu aplikací. Hraje velkou roli v grafických aplikacích, které kombinují obecné paralelní výpočty s vykreslováním grafiky. Součástí OpenCL je API (Application Programming Interface) pro lepší fungování paralelních výpočtů v různorodých procesech.

Původní návrh standardu OpenCL pochází od společnosti Apple. Od roku 2008 se o vývoj tohoto standardu stará organizace Khronos s aktuální verzí OpenCL 2.0.

Tento standart umožňuje paralelní programování na CPU, GPU a jiných výpočetních jednotek, které jsou uspořádány do jediné platformy. OpenCL je plnohodnotný framework, který slučuje API, programovací jazyk, knihovny a runtime systém pro podporu vytváření programů.

## Architektura OpenCL

1. Platformní model
2. Vykonávací model
3. Paměťový model
  - a) Globální paměť (Global Memory)
  - b) Konstantní paměť (Constant Memory)
  - c) Lokální paměť (Local Memory)
  - d) Privátní paměť (Private Memory)
4. Programovací model

Programování OpenCL je složeno ze dvou částí. V první části host soubor zjišťuje zařízení a jejich nastavení. Dále určí, na kterém zařízení by se měl program spustit. V druhé části se programuje kernel, což je samotná funkce programu, která se spustí na zařízení. Programování má jistá omezení z hlediska programovacího jazyka. Probíhá v C99 dialektu programovacího jazyka C. Možná je spolupráce s OpenGL a dalšími API. Host soubor nemusí být naprogramován v C, ale je možné použít jazyky Java, Python, C++ a další.

Výhodou OpenCL, díky abstrakci, je nezávislost na zařízení, kdy je možné využívat OpenCL na grafických kartách AMD a Nvidia, se správnou implementací i na různých procesorech.[8][9]

## 6.4 CPU vs. GPU pro BlockMatching

CPU počítá v subpixelových disparitách, a proto je disparitní mapa jemnější. Pomáhá si tím, že využívá na výpočet korespondence SAD okno, které se počítá rychleji. [11]

Ve výchozím stavu používá XSobel Filtr.

Vzorec pro SAD:

$$D_{SAD} = \sum_{(i,j) \in U} |I_1(x+i, y+j) - I_2(x+d_x+i, y+d_y+j)| \quad (33)$$

GPU si většinu parametrů je schopný zajistit sám a používá SSD. [11]

Vzorec pro SSD:

$$D_{SSD} = \sum_{(i,j) \in U} (I_1(x+i, y+j) - I_2(x+d_x+i, y+d_y+j))^2 \quad (34)$$

## 7 PRAKTICKÁ ČÁST

Cílem práce je navrhnout aplikaci, která bude vykreslovat disparitní (hloubkovou) mapu snímané scény. Jakožto stereoskopický senzor scény snímají dvě webové kamery Microsoft LifeCam Studio, model 1425. Aplikace je navržena tak, že v prvním kroku se kalibruje kamerová soustava a v druhém kroku je možné pořizovat disparitní mapu, jako jednotlivé snímky.

### 7.1 Popis programu

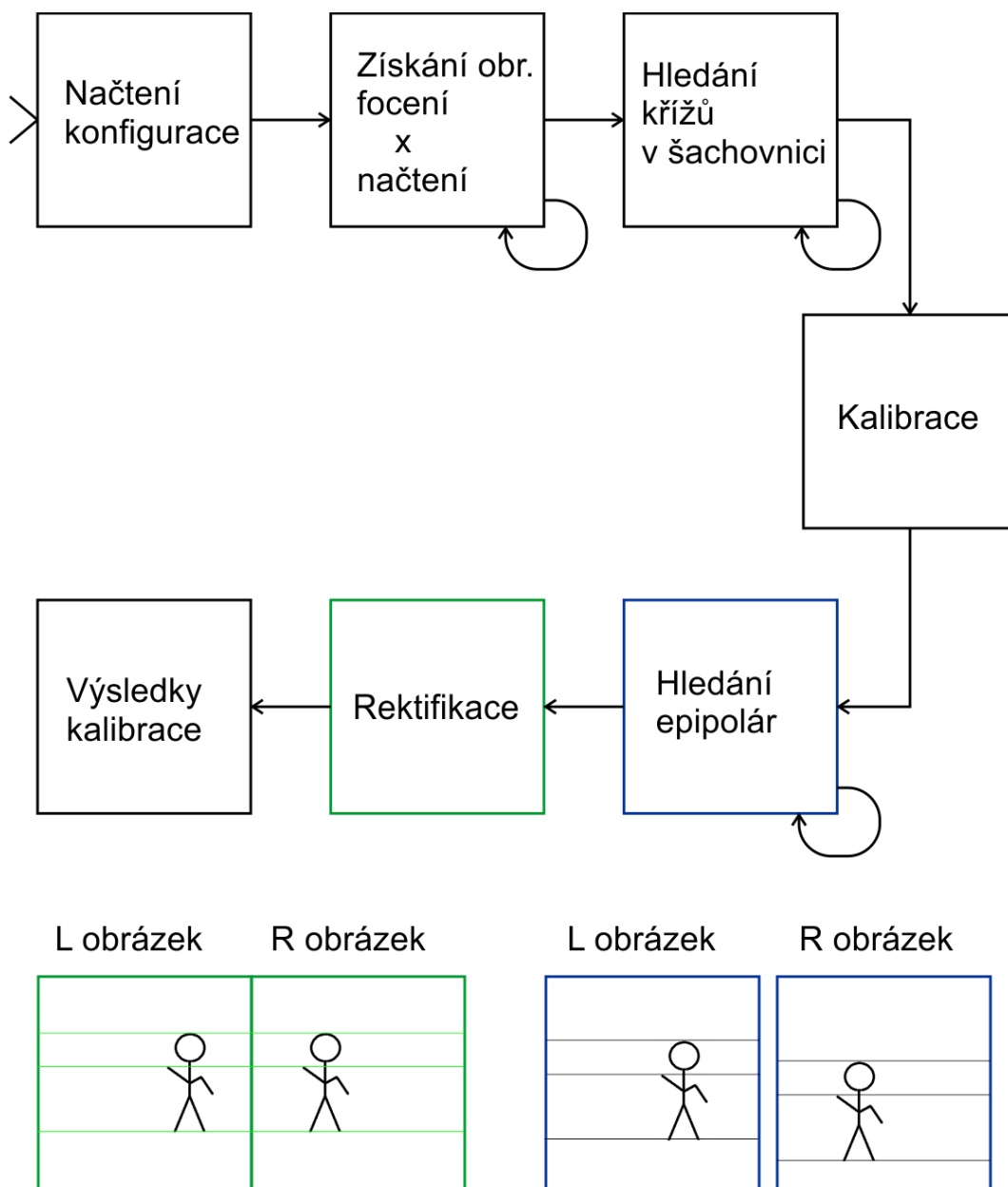
Program je rozdělen na základní dvě části. V první části se provádí snímání scény a následná kalibrace kamerové soustavy z nasnímaných fotek. V druhé části pak následuje výpočet disparitní mapy z použitých stereo-snímků.

#### **Kalibrace**

V hlavičce programu kalibrace je možné nastavit, jestli bude program fotit a následně kalibrovat, nebo jenom fotit, nebo pouze spustí kalibraci pro pořízené snímky jiným stereoskopickým senzorem (fotoaparát). Dále je možné nastavit počet pořizovaných párů fotek, z níž má být následně provedena kalibrace soustavy. Pro kalibraci používá program automatizovanou metodu, kdy ve scéně program hledá šachovnici. Proto je dalším parametrem k nastavení počet křížů šachovnice na ose x a na ose y. Pokud nastavíme možnost fotit a kalibrovat, program po spuštění aktivuje kamery a zobrazí v samostatných oknech (z levé a pravé kamery) snímanou scénu. V dalších dvou oknech jsou totožné obrazy scény, ale zvětšené dle nastavení proměnné zoom. Po zmáčknutí tlačítka „u“ na klávesnici, program začne ukládat požadovaný počet fotek v nastaveném intervalu a tlačítkem „x“ je možné předčasně ukončit snímání.

V dalším kroku programu se automaticky spustí kalibrace (pokud je nastaveno). Zobrazí se okno, ve kterém se zobrazuje zpracovaná fotografie s nalezenými body v místech křížení na šachovnici (pokud ji program na fotografii nalezne). Až se projdou všechny fotografie, zobrazí se další okno, kde se bude zobrazovat výstup kalibrace stereofotografií. Ten se skládá z dvojice snímků, tedy snímku s radiálním zkreslením a snímku s odstraněným radiálním zkreslením. Výsledkem kalibrace jsou soubory extrinsics,

intrinsics s vnějšími a vnitřními parametry kamer a parametry radiálního zkreslení. Navíc se vytvoří soubor info, kde jsou uloženy informace o průběhu a chybovosti zpracování.



Obr. 9: Diagram kalibrace



## Matching

Další částí programu je matching. Celý program je konfigurovatelný přes jediný soubor. Jméno tohoto souboru je první a jediný parametr při spuštění programu z příkazové řádky. Pokud nebude jeho jméno uvedeno, je použito jako výchozí jméno "config.txt". Pokud nebude konfigurační soubor nalezen, uloží program jeho výchozí podobu, aby ho mohl uživatel jednodušeji nastavit.

Druhy parametru:

Systémové parametry:

calibrationDirectory - určuje jméno adresáře, ve kterém jsou uložena data z kalibrace,

photoDirectory - určuje adresář, kam se mají ukládat nasnímané fotografie,

leftCameraIndex - index levé kamery (pokud bude použit vstup z kamer),

rightCameraIndex - index pravé kamery (pokud bude použit vstup z kamer),

leftCameraInput - cesta ke zdroji dat pro levou kameru, buď cesta k videosouboru, nebo formát cesty k fotografiím, kde se %d nahradí číslem fotky,

rightCameraInput - cesta ke zdroji dat pro pravou kameru, buď cesta k videosouboru nebo formát cesty k fotografiím, kde se %d nahradí číslem fotky,

openCLDeviceIndex - index zařízení použitého pro OpenCL, pokud je -1, je použita autodetekce (najde se zařízení s maximálním počtem výpočetních jednotek), výpis všech zařízení pro OpenCL uživatel nalezne v souboru openccl\_devices.txt po prvním spuštění programu,

maxFPS - určuje počet snímků za sekundu, které budou použity, pokud se bude zaznamenávat barevná disparitní mapa nebo vstup z kamer do videosouboru,

videoMode - zapíná video mód, viz. dále input - určuje druh vstupu 0 pro vstup z kamer, 1 pro načítání fotek ze souboru, 2 pro načítání obrázků z videosouboru,

storeVideo - 0/1 má program ukládat barevnou disparitní mapu do videosouboru, pokud zapnuto a vstup je brán z kamer, tak se ukládají i vstupy do videosouboru.

Zobrazovací parametry:

Anaglif - pro každý snímek se zobrazí okno s Anaglifem,

BWDisparityMap - pro každý snímek se ukáže okno s šedotónovou disparitní mapou,

RectifiedInput - ukáže okno s rektifikovanými vstupy složenými vedle sebe,

Legend - 0/1 vypíná/zapíná legendu, která ukazuje vzdálenosti pro minimální a maximální nalezenou disparitu a pro 4 hodnoty mezi nimi,

MouseDistance - 0/1 vypíná/zapíná zobrazování vzdálenosti objektu od kamer při pohybu myši v okně barevné disparitní mapy.

Startovní parametry matchingu:

OpenCL - 0/1 vypíná/zapíná použití OpenCL,

BM - 0/1 určuje metodu block matchingu SGBM/BM,

preFilterCap, preFilterSize, SADWindowSize, minDisparity, numberOfDisparities, textureThreshold, averageTexThreshold, uniquenessRatio, speckleWindowSize, speckleRange=32, disp12MaxDiff, fullDP, XSobelFilter jsou vstupní parametry pro block matchingové metody.

Koncová filtrace:

UseRangePostFiltering - 0/1 vypíná/zapíná ořezání disparitní mapy v daných mezích před převedovem na 256 stupňovou šedotónovou disparitní mapu,

MinDistance - určuje minimální vzdálenost, která nebude ořezána,

MaxDistance - určuje maximální vzdálenost, která nebude ořezána,

RealUnits - 0/1 určuje typ jednotek ořezání (disparity/ centimetry).

Módy průchodu programu:

Vypnutý videomód:

Program postupně nachází dříve nafocené snímky a provádí nad nimi matching. Výsledek matchingu se zobrazí v okně a čeká se na vstup z klávesnice. Při stisku tlačítka

„mezerník“ a „enter“ se přechází na zpracování dalšího stereopáru, ale při stisku tlačítka „mezerník“ se výsledek zpracování uloží.

Pokud zpracováváme aktuální dění před kamerami, tak program vyfotí scénu a provede nad ní matching a zobrazí ho v okně. Na stisk „mezerník“ a „enter“ se vyfotí nová scéna, ale při stisku tlačítka „mezerník“ se výsledek zpracování uloží.

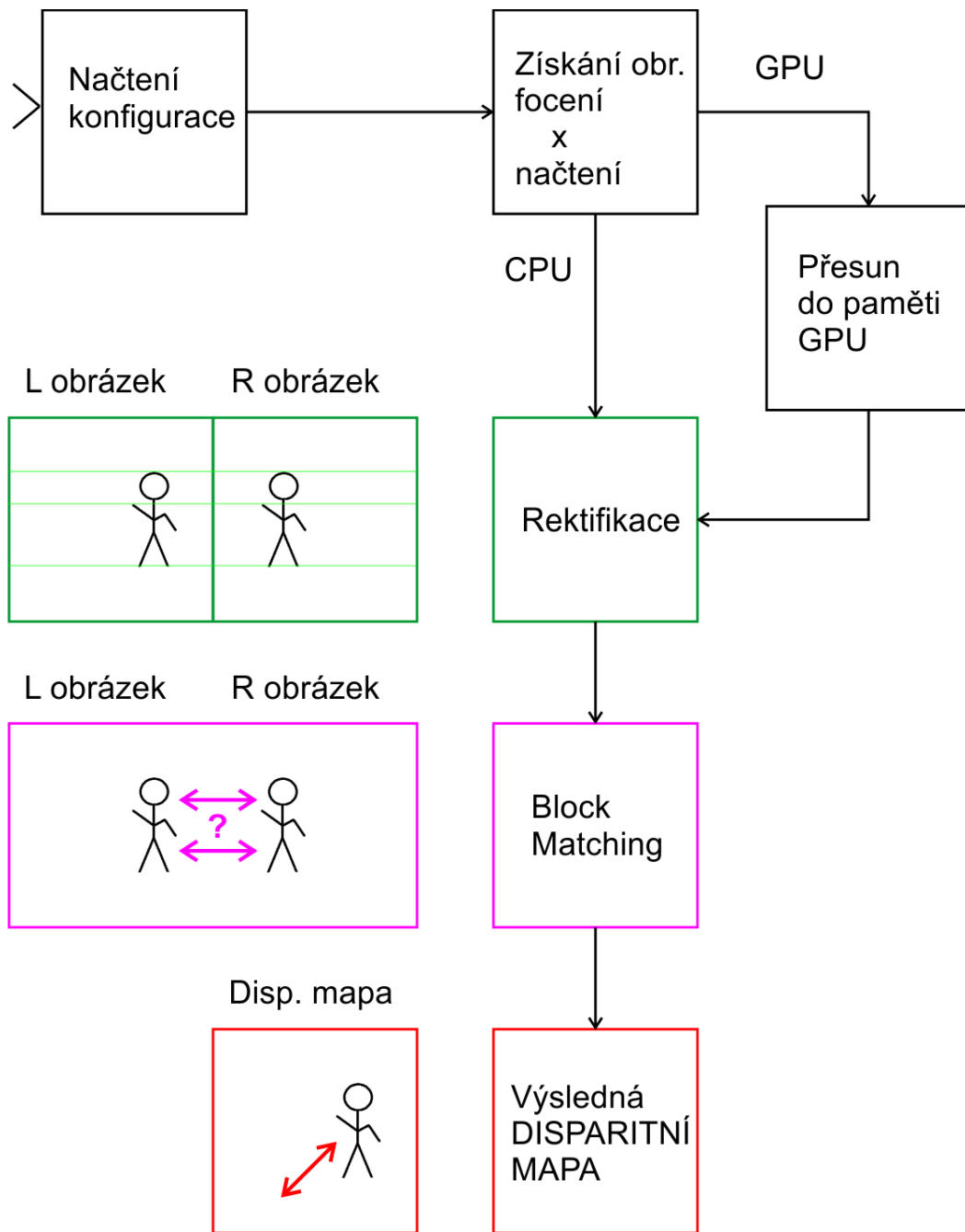
Klávesou „x“ je možné program kdykoliv ukončit.

Zapnutý videomód:

Program bez čekání nahraje vstup a zpracuje ho podle parametru konfigurace. Nečeká, až uživatel stiskne nějakou klávesu, ale zkouší zpracovávat ihned další a další obrazy ze vstupu. Ukončení a změna parametru probíhá stejně jako u vypnutého videomódu. Ukládat výstup v tomto módu lze pouze do videosouboru.

Klávesou „x“ je možné program kdykoliv ukončit. V hlavičce lze měnit výchozí hodnoty všech parametrů.

Důležitým parametrem je míra zvětšení zpracovávaných fotografií. To nám může zpřesnit výsledek, ale prodloužit dobu zpracování. Míra zvětšení se určuje ve stovkách procent.



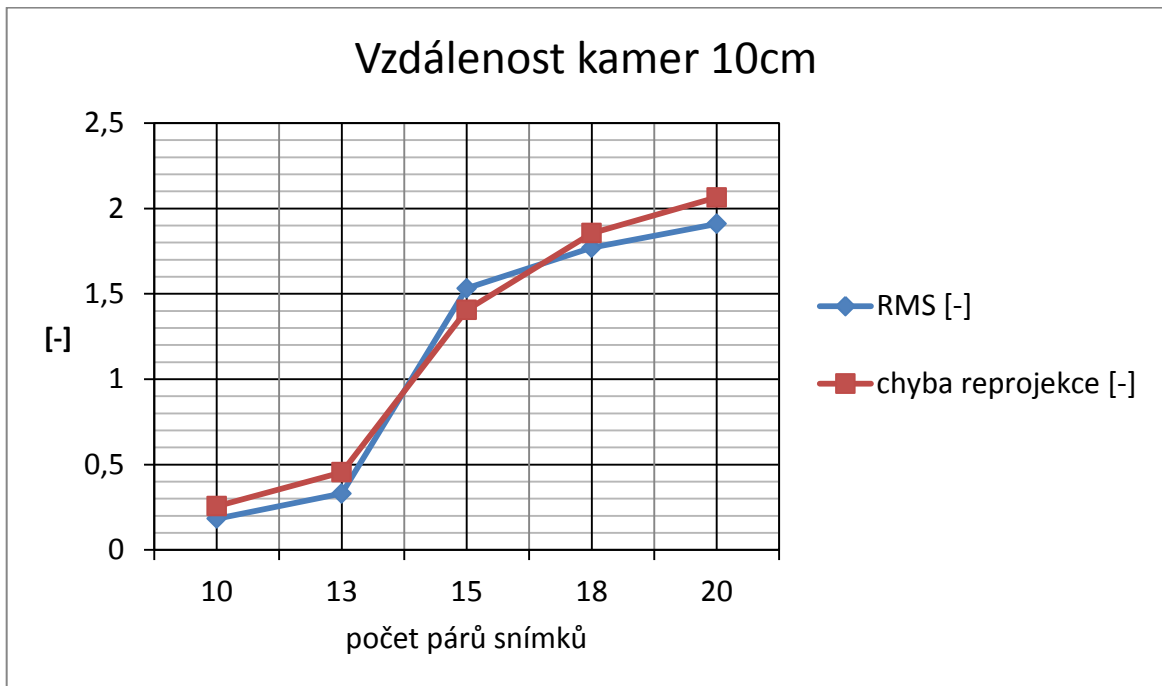
Obr. 10: Diagram matching

Po zobrazení výstupu matchingu lze také měnit jeho parametry přes následující klávesové zkratky:

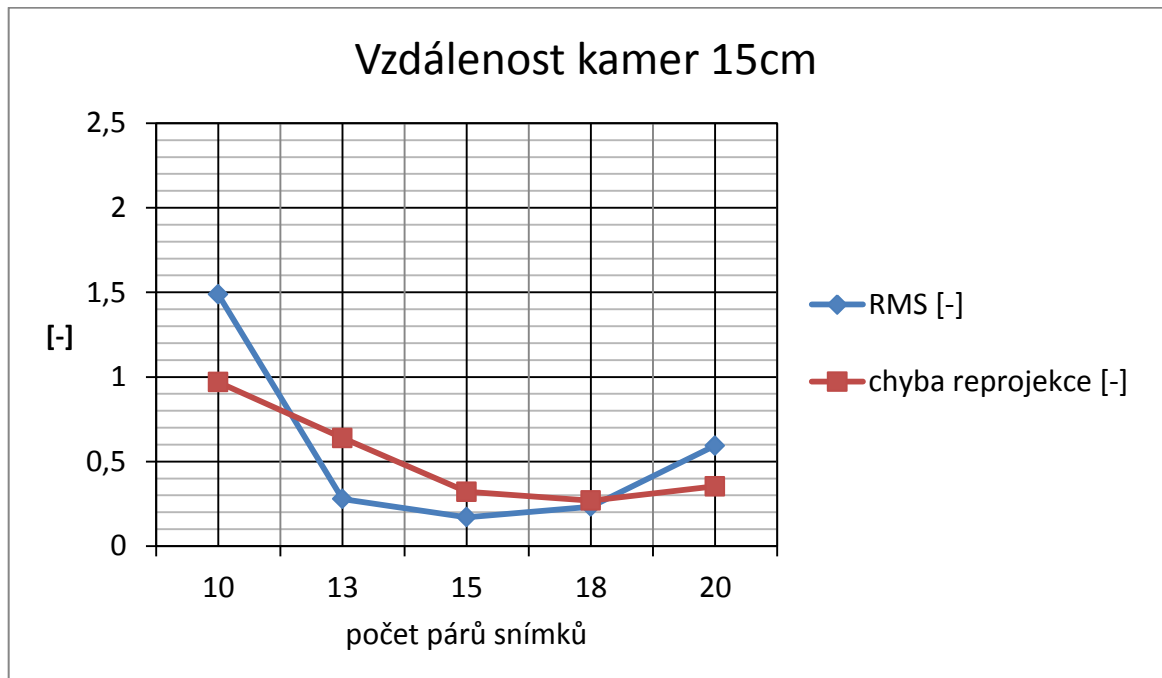
Tab. 1: Tabulka parametrů

Parametr	Klávesa pro zvýšení hodnoty	Klávesa pro snížení hodnoty	Hodnota změny
preFilterCap	Q	W	1
preFilterSize	B	N	2
SADWindowSize	E	R	2
minDisparity	U	I	16
numberOfDisparities	O	P	16
textureThreshold (CPU)	A	S	1
uniquenessRatio	D	F	1
speckleWindowSize	G	H	10
speckleRange	J	K	5
disp12MaxDiff	C	V	3
Přepínání BM/SGBM	M		true/false
Přepínání OCL/CPU	T		true/false
FullDP (pro SGBM)	L		true/false
XSobel filtr	1		true/false
texture Threshold (OCL)	2	3	*2 / *(1/2)
UseRange	5		true/false

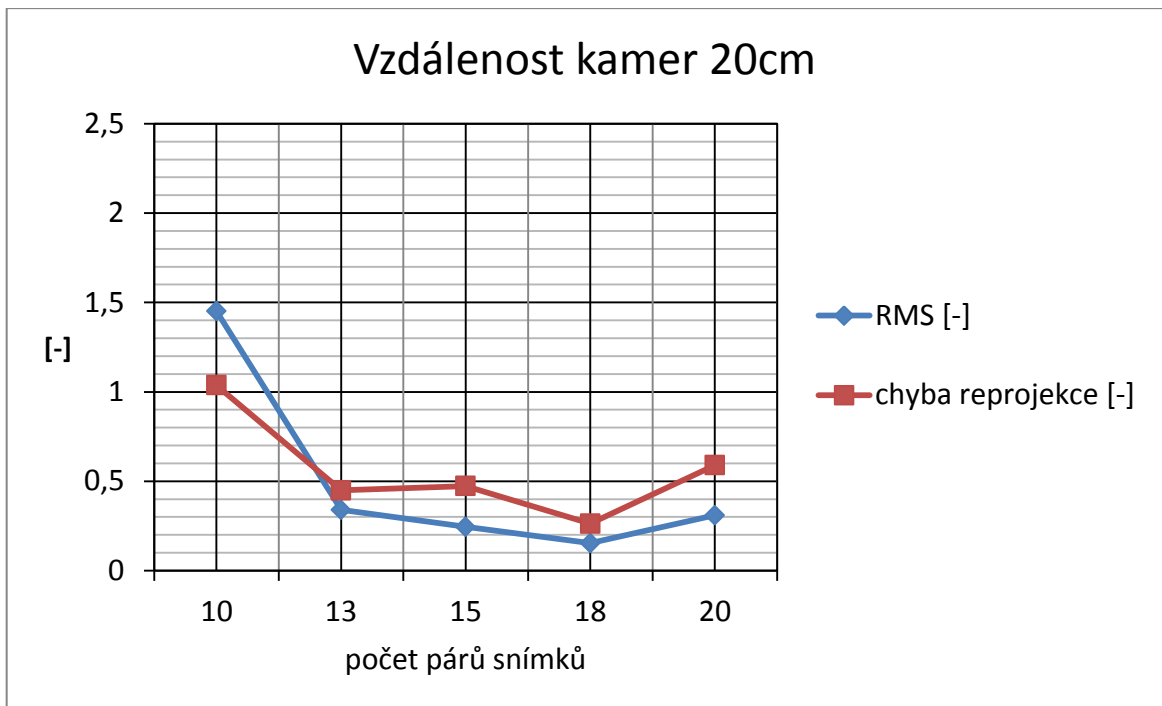
## 7.2 Grafy – statistické hodnoty z kalibrací



Graf 1: Graf chybovosti reprojekce (10cm)



Graf 2: Graf chybovosti reprojekce (15cm)



Graf 3: Graf chybovosti reprojekce (20cm)

Chyba reprojekce je průměrná odchylka každého bodu obrazu od epipoláry odpovídajícího bodu v druhém obrázku neboli je průměrná vzdálenost zaměřeného bodu od jeho skutečné pozice.

RMS je kvadratický průměr chyby reprojekce.

Grafy chybovosti na začátku definičního oboru připomínají parabolu. Nejprve se se zvyšujícím počtem snímků zmenšuje chyba, protože systém nabírá data, která kalibraci zpřesňují. To ovšem do kritického bodu než jsou zpřesňující data znehodnocována šumem a nepřesností snímáního postupu a od něj začne chybovost stoupat a už se na původní hodnoty nevrátí. Počet snímků potřebný pro dosažení tohoto kritického bodu se zvětšuje se vzdáleností kamer od sebe. A parabola se se zvyšující vzdáleností kamer zplošťuje.

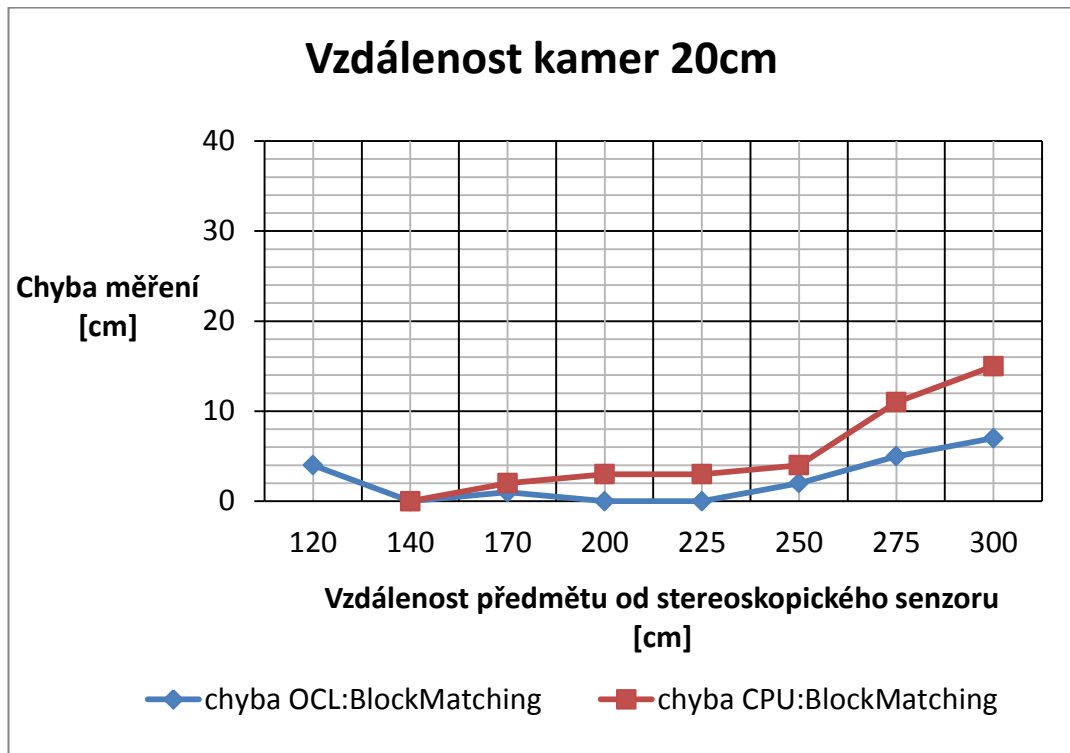
V průběhu měření jsem také narazil na problém se samotným snímáním. Jelikož jsou snímky snímány po sobě a nikoliv naprosto ve stejnou chvíli, figurant se může pohnout mezi snímáním první a druhé kamery a pak každý snímek představuje „jinou“ scénu. Zmenšit chybu kalibrace je tak možné snímáním statické scény, kde šachovnici neдрží člověk, ale stojan. Člověk se díky biologickým pochodům a neschopnosti přesně odhadnout čas snímání snímků hýbe a obraz je buď rozmazanější, nebo „jiné“ scény.

Tuto chybu jsem maximálně zredukoval tím, že nejprve kamery vyfotí snímky a teprve poté se snímky uloží do paměti počítače. Tím se zmenší prodleva mezi jednotlivým pořízením snímků a dojde ke zvýšení přesnosti při kalibraci a hlavně při snímání dynamické scény.

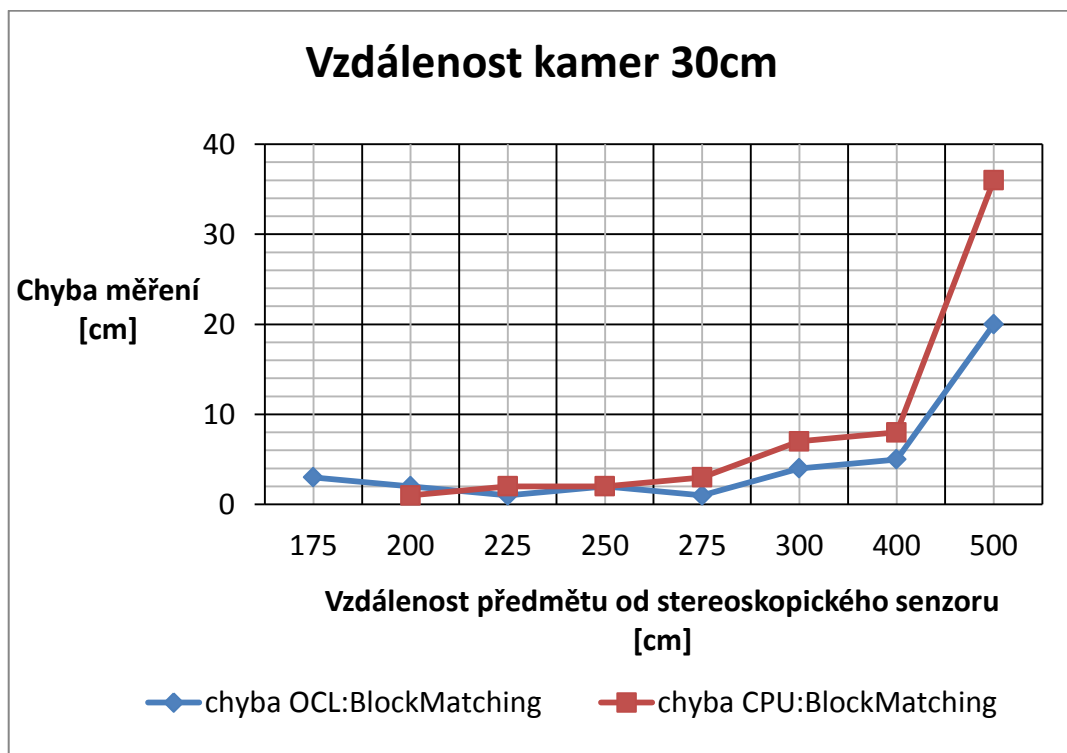
Zdržení při zpracování snímků nastává při přesunu dat mezi pamětí GPU a RAM. Grafické kartě (respektive OpenCL zařízením) trvá zpracování matchingových algoritmů řádově milisekundy. Další zdržení nastává převedením obrázku na šedotónový, rektifikací, obarvením disparitní mapy. Čas výpočtu čistě disparitní mapy z jednotlivých obrazových stereopárů jsou desítky milisekund.



### 7.3 Grafy – chyby měření vzdálenosti



Graf 4: Graf závislosti chyby měření na vzdálenosti předmětu (20 cm)



Graf 5: Graf závislosti chyby měření na vzdálenosti předmětu (30 cm)

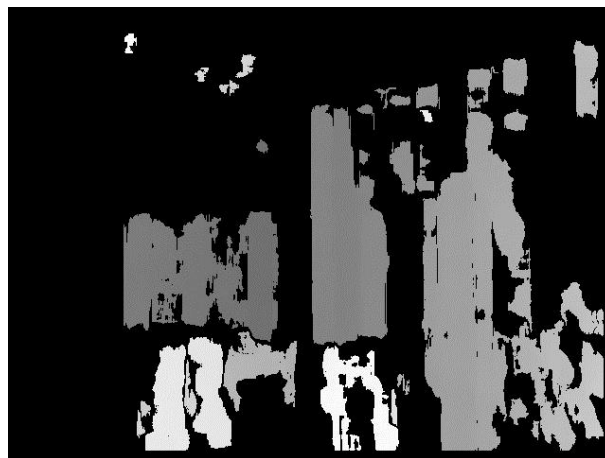
## 7.4 Převod disparitní mapy z šedotónového do barevného spektra

Disparitní mapy původně šedotónové. Pro lidské oko je lepší použít barevné spektrum, protože více vyniknou jednotlivé hloubky obrazu, dané barevným spektrem.

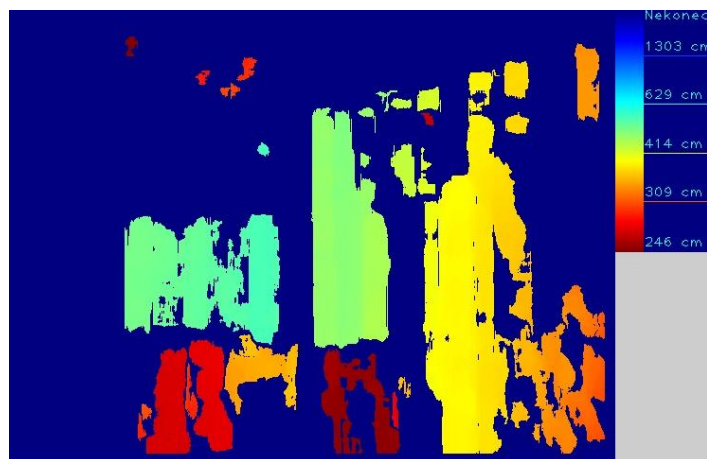
Výstupem BM jsou šedotónové disparitní mapy, ve kterých je znázorněna disparita různě šedými tóny. Při zpracování procesorem zabírají disparity jen malou část spektra obrazu. Je třeba toto spektrum rozšířit natolik aby byl obraz pro člověka přehlednější. První fází je převedení na ideální šedotónovou disparitní mapu, která má 256 odstínů a disparity se nacházejí v celém jejím spektru. Blízké předměty jsou obarveny bíle, a čím jsou vzdálenější, tím jsou tmavší.



Obr. 11: Disparitní mapa BM (CPU) – originál



Obr. 12: Disparitní mapa BM (CPU) – upravená, 256 tónů šedi v celém spektru



Obr. 13: Disparitní mapa BM (CPU) – upravená, 256 tónů barev palety Color Jet

Druhou fází je převedení šedotónové disparitní mapy na barevnou, pomocí barevné palety Color Jet. Zde jsou blízké předměty zobrazeny červeně, a čím více jsou vzdáleny, tím více se přebarvují do modra.



Obr. 14: Paleta Color Jet

Tato paleta zabírá ze spektra, které je lidské oko schopné pozorovat, daleko více než pouhá šedotónová paleta. [12]

## 7.5 Určení vzdálenosti objektu z disparitní mapy

Výstupem BM algoritmů je zase disparitní mapa, která sama o sobě určuje vzdálenosti bodu od kamer, ale určuje ji v disparitních jednotkách, což jsou pixely (rozdíl v pixelech) a jsou závislé na rozmístění kamer a na jejich natočení. Pro člověka je jednotka disparita těžko představitelná a proto je nutné tuto hodnotu převést například na centimetry.

Výpočet reálné vzdálenosti od kamer:

$$Z = f \frac{T}{d} \quad (35)$$

$f$  - ohnisková vzdálenost,

$T$  - vzdálenost kamer (velikost translačního vektoru),

$d$  - disparita,

$Z$  - výsledná vzdálenost.

Parametry  $f$  a  $T$  určuje kalibrace kamer. Čím přesněji se kamery zkalibrují, tím přesnější bude výpočet vzdálenosti ze dvou obrázků, tedy přesnější disparitní mapa. Tyto parametry jsou součástí hloubkové matice, taky aby dokázala převádět z projekčního prostoru kamer do reálného prostoru, jehož střed je v levé kameře. Natočení levé kamery určuje osu  $Z$ . Do rovnice níže se dosadí matice  $\mathbf{Q}$  a vektor  $\mathbf{V}$ . Výsledkem je vektor  $\mathbf{p}$ .

$$\mathbf{Q} * \mathbf{V} = [x \quad y \quad z \quad w] \quad (36)$$

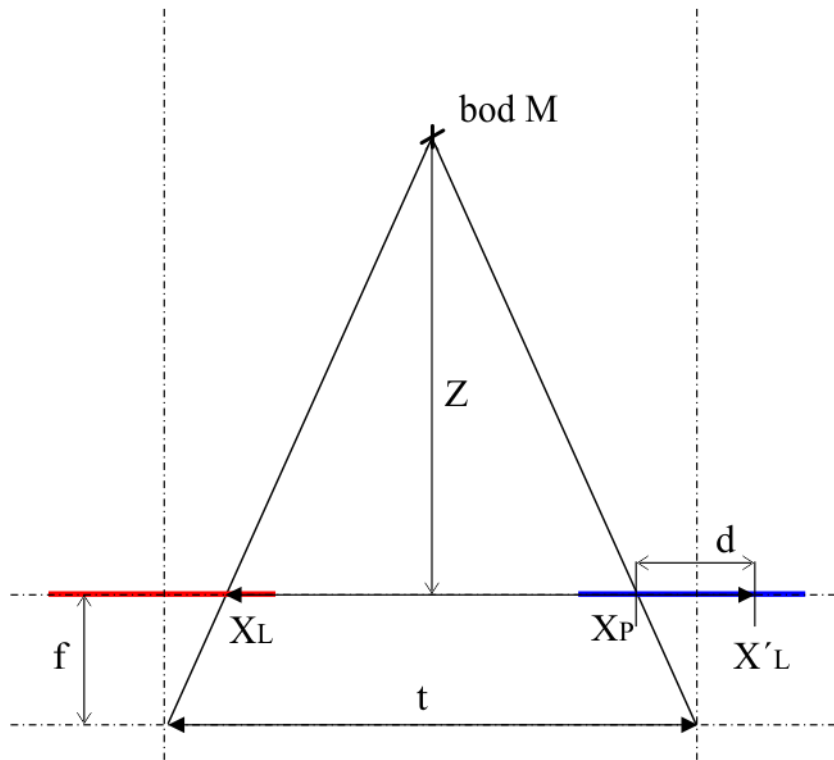
$$\mathbf{p} = \left[ \frac{x}{w} \quad \frac{y}{w} \quad \frac{z}{w} \right] \quad (37)$$

$$\mathbf{V} = \begin{bmatrix} B_x \\ B_y \\ d \\ 1 \end{bmatrix} \quad (38)$$

$Q$  - hloubková matice,

$B_x, B_y$  - pozice bodu v obr,

$p$  - fyzická pozice, vektor vůči levé kameře.[11]



Obr. 15: Určení vzdálenosti bodu

## 7.6 Ukázky disparitních map

V této kapitole jsou zařazeny ukázky výsledných disparitních map pořízené stereokamerou, tedy párem kalibrovaných kamer

**Snímaná scéna – kamery od sebe vzdálené 30 cm**



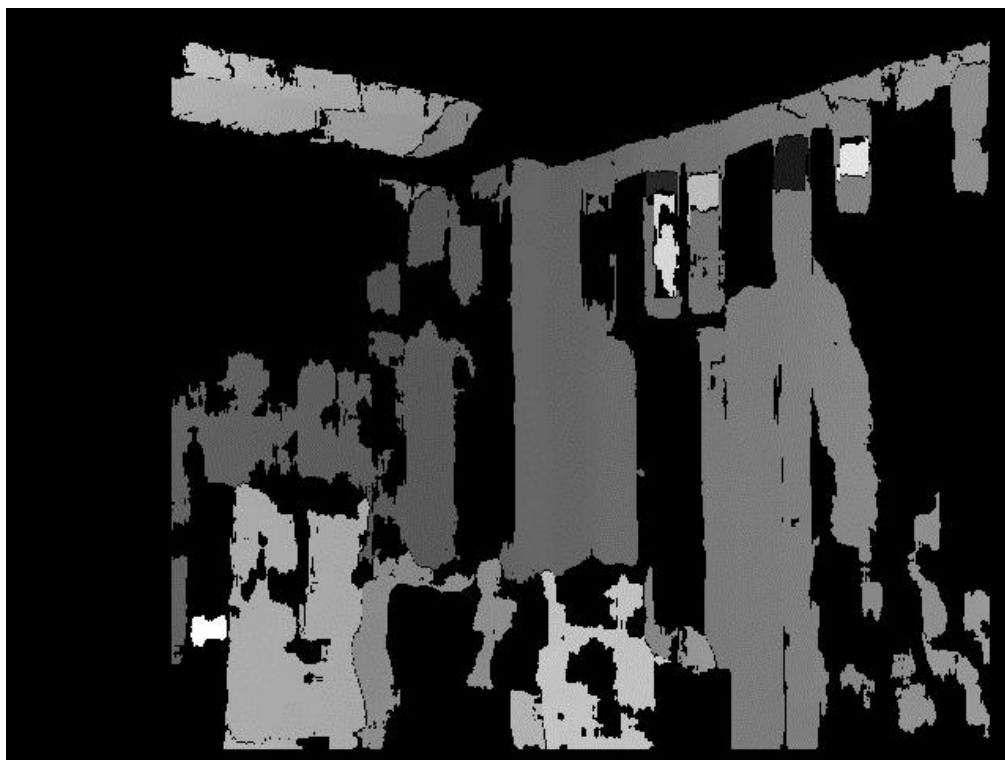
Obr. 16: Fotografie z levé a pravé kamery



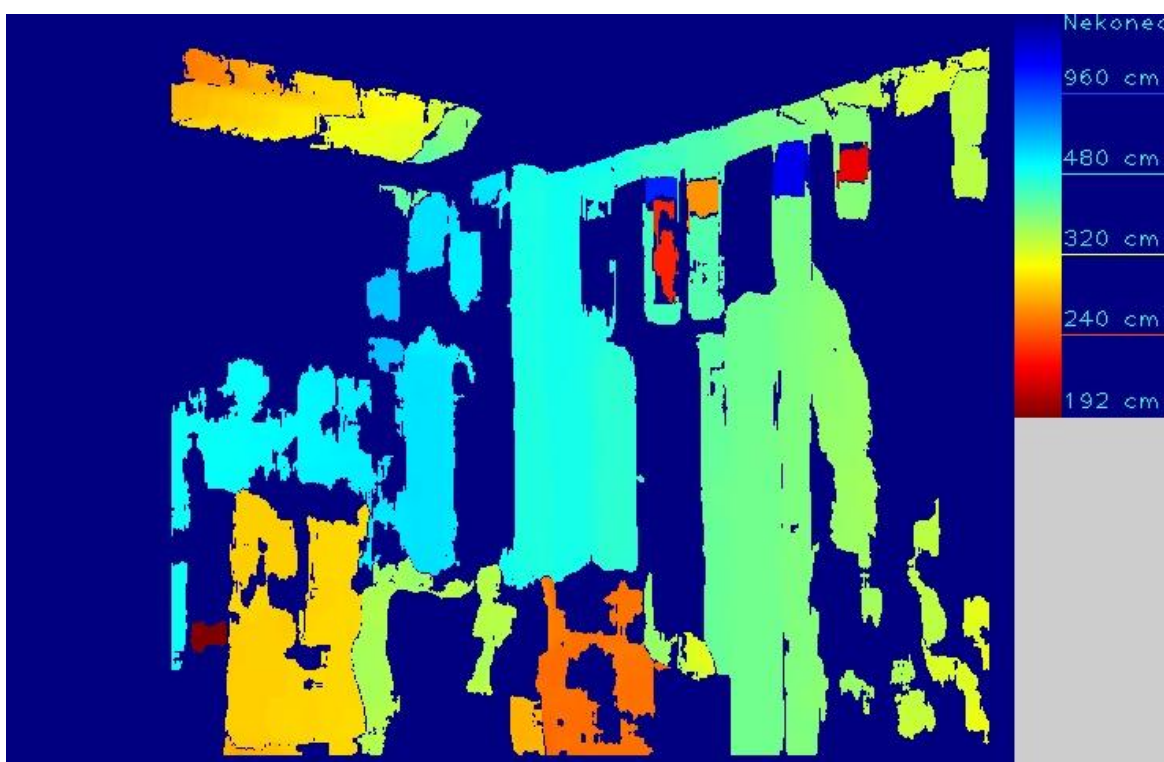
Obr. 17: Rektifikovaný stereopár



Obr. 18: Anaglifické zobrazení

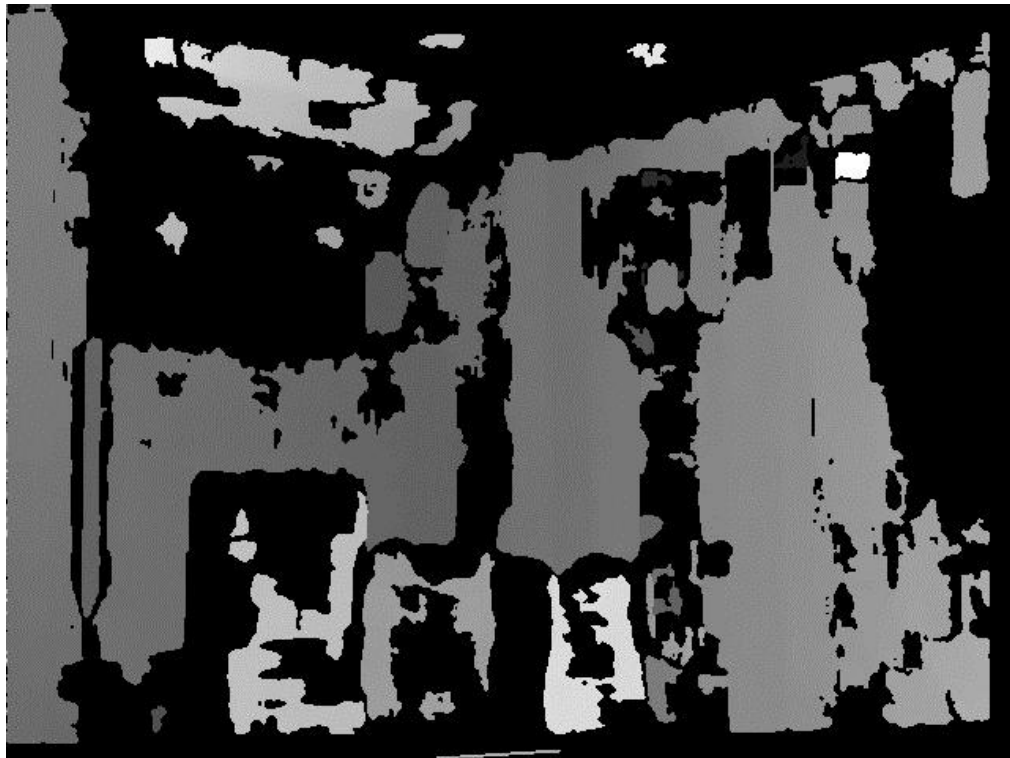


Obr. 19: Disparitní mapa šedotónová (CPU:BM)

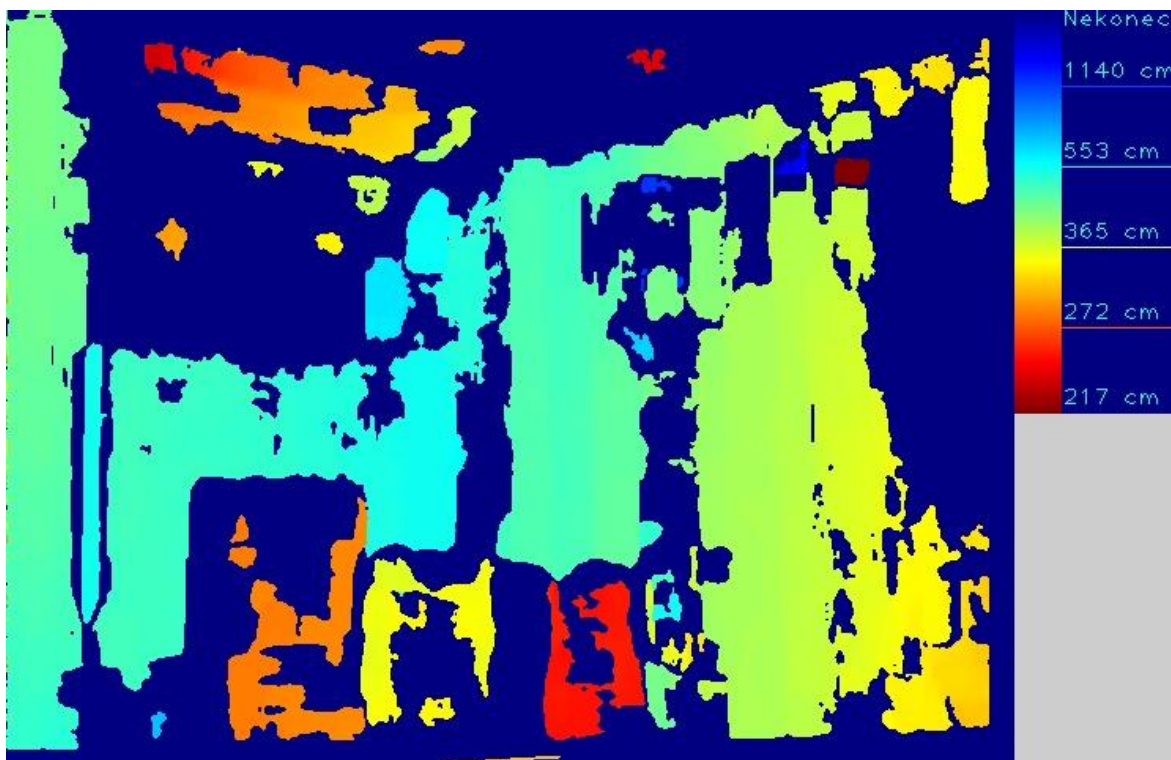


Obr. 20: Disparitní mapa obarvená Color Jet (CPU:BM)

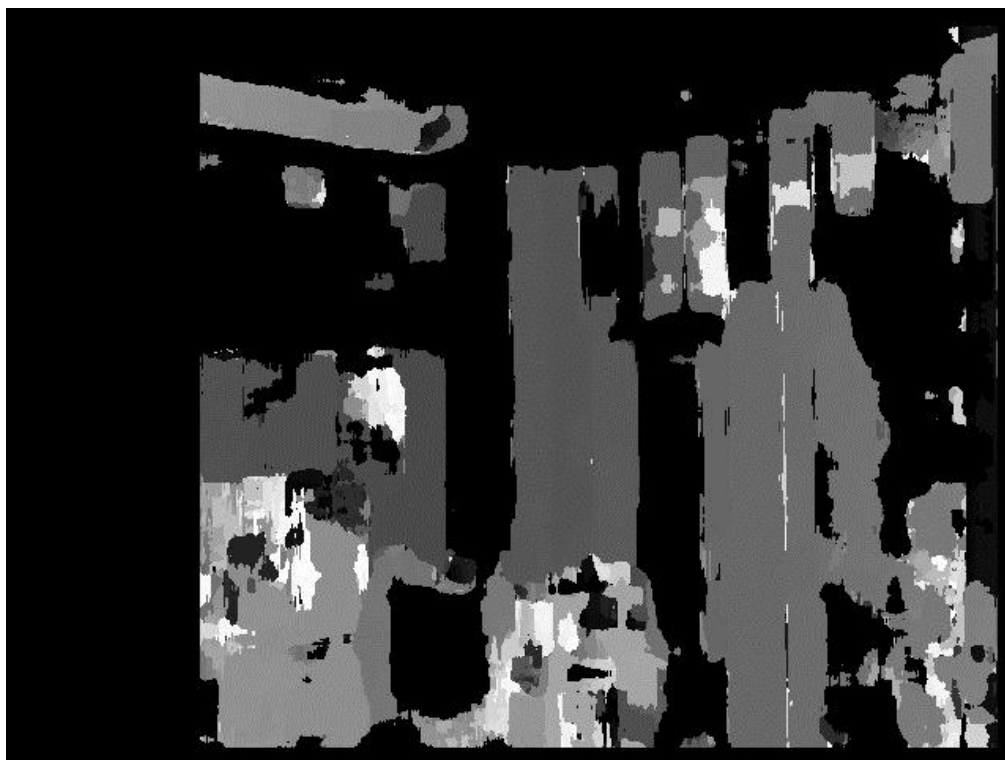




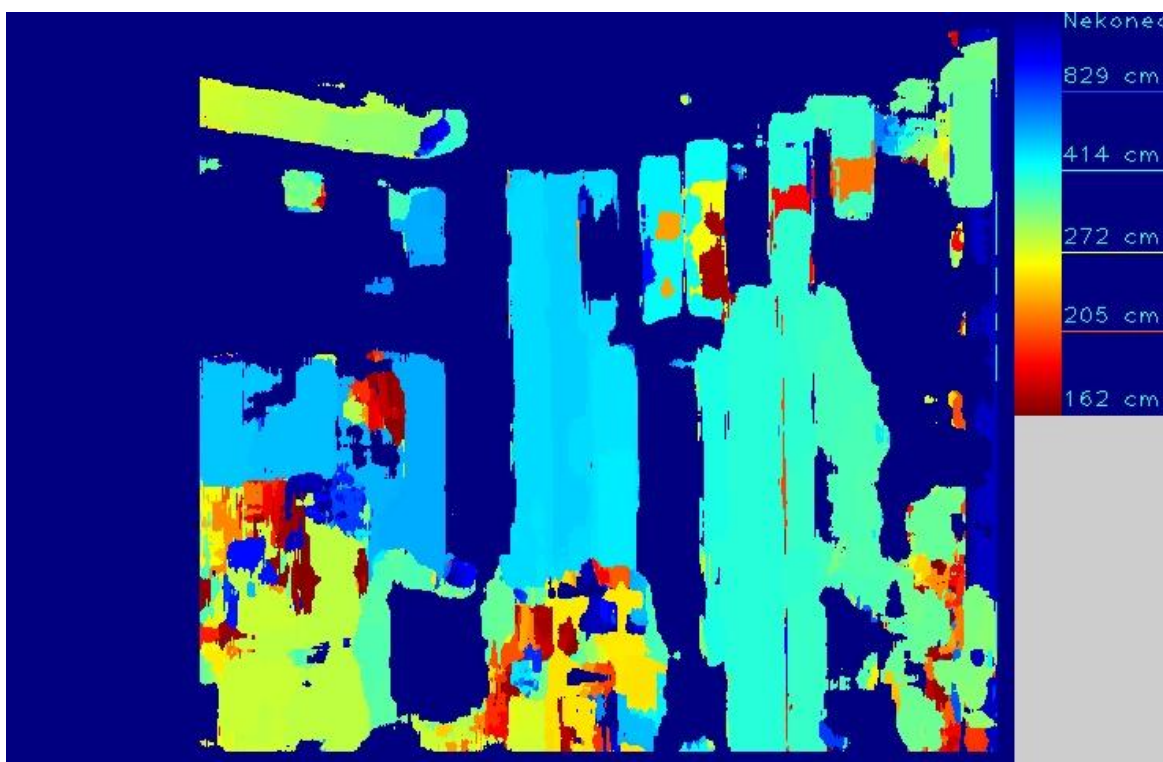
Obr. 21: Disparitní mapa šedotónová (CPU:SGBM)



Obr. 22 : Disparitní mapa obarvená Color Jet (CPU:SGBM)



Obr. 23: Disparitní mapa šedotónová (OCL:BM)



Obr. 24: Disparitní mapa obarvená Color Jet (OCL:BM)

## 8 ZÁVĚR

Dle postupů uvedených v úvodu práce, jsem zjistil potřebné parametry kamer. Pořízené stereoskopické snímky respektive snímky obecné scény spolu se zjištěnými parametry kamer byly základem pro doražení cíle práce a to vytvoření hloubkové neboli disparitní mapy.

Grafy chybovosti shora uvedené poukazují na chyby reprojekce, které nastaly při kalibrování kamer v závislosti na počtu snímků použitých při kalibraci. Bylo zjištěno, že při menším než optimálním počtu snímků chybovost roste. Stejný výsledek byl zjištěn při větším počtu snímků, než je počet optimální. Graf chybovosti se tedy blíží parabole.

Problémem se ukázalo i samotné snímání. Snímky nejsou snímány ve stejný okamžik a mohou se tak změnit okolnosti scény. Toto se nakonec podařilo zredukovat na absolutní minimum tím, že obě kamery necháme pouze vyfotit snímek a teprve až budou mít obě vyfoceno, necháme si od nich obrázek poslat. To zredukovalo prodlevu mezi snímky z desítek milisekund na setiny milisekundy a můžeme říct v tom případě si oba vyfocené obrázky odpovídají.

Grafy chybovosti v měření vzdálenosti ukazují, že nejpřesněji block-matching měří nejbližší předměty, které jsou schopné rozpoznat, a se stoupající vzdáleností klesá přesnost, takže grafy opět připomínají parabolu. To jen potvrzuje, že rozdíl dvou po sobě jdoucích disparit roste kvadraticky. Grafy navíc ukazují, že zpracování na CPU rozpozná předměty dále a mnohem rychleji u něj roste chybovost měření.

Pro testování výše popisovaných metod jsem použil sestavu AMD Athlon X2 250 (3GHz) se 4 GB RAM a grafiku AMD Radeon HD 7770. Výpočet disparitní mapy na CPU trval 250-300 ms. Při použití OpenCL na grafické kartě se disparitní mapa počítala 36-58 ms, což už skoro dosahuje 24 snímků za sekundu a dosáhlo tak výpočtu disparitní mapy v reálném čase.

Pro dosažení lepších výsledků můžeme použít přesnější matchingové algoritmy, ale obávám se, že budou velmi výrazně prodlužovat čas výpočtu disparitní mapy. Dále by šly použít kamery určené pro stereo snímání, některé dnes prodávané se dají synchronizovat, aby fotily zároveň. Přístup k fotografiím přes rozhraní USB není úplně nejvhodnější, protože má velké prodlevy a neumožňuje přímý přístup do paměti kamer.

Dokumentace nasnímané scény a z ní pocházejících disparitních map je uvedena shora.

Závěrem lze říci, že shora uvedeného cíle práce bylo dosaženo.

## 9 ZDROJE

- [1] BOURKE, Paul. *Calculating Stereo Pairs* [online]. 1999 [cit. 2013-12-08]. Dostupný z WWW: <<http://local.wasp.uwa.edu.au/~pbourke/miscellaneous/stereographics/stereorender/>>
- [2] BRADSKI, Gary; KAEHLER, Adrian. *Learning OpenCV*. [s.l.] : O'Reilly, 2008. Stereo Rectification, s. 430-450.
- [3] ČÍŽEK, Petr. *Prostorové zobrazování*. [s.l.], 2005. 59 s. Diplomová práce. Dostupný z WWW: <[http://herakles.zcu.cz/~skala/VID/Data/PetrCizek\\_DP2005.pdf](http://herakles.zcu.cz/~skala/VID/Data/PetrCizek_DP2005.pdf)>.
- [4] HARTLEY, Richard; ZISSERMAN, Andrew. *Multiple View Geometry in Computer Vision : Second Edition*. [s.l.] : [s.n.], 2003. 672 s.
- [5] REZEK, Štěpán. *Vizualizace v e-learningových systémech pro mobilní robotiku*. Praha, 2009. Dostupné z WWW: <[https://dip.felk.cvut.cz/browse/pdfcache/rezeks1\\_2009dipl.pdf](https://dip.felk.cvut.cz/browse/pdfcache/rezeks1_2009dipl.pdf)>. Diplomová práce. ČVUT v Praze.
- [6] ING. ŘÍHA, Kamil; ING. HUJKA, Petr. *Elektrorevue* [online]. 2005 [cit. 2014-01-22]. *Epipolární geometrie* Dostupné z WWW: <<http://www.elektrorevue.cz/clanky/05017/index.html>>.
- [7] TRIOLET D., "Nvidia CUDA: preview," *BeHardware*, [online] 2007 [cit. 2013-11-26]. Dostupné z WWW: <<http://www.behardware.com/articles/659-1/nvidia-cuda-preview.html>>
- [8] WOOLLEY, Cliff. *Introduction to OpenCL*, [online] 2011 [cit. 22. 12. 2013]. Dostupné z WWW: <[http://www.cc.gatech.edu/~vetter/keeneland/tutorial-2011-04-14/06intro\\_to\\_opencl.pdf](http://www.cc.gatech.edu/~vetter/keeneland/tutorial-2011-04-14/06intro_to_opencl.pdf)>
- [9] OpenCL Specification 2.0[online]. 2011 [cit. 2014-01-28] Dostupné z WWW: <http://www.khronos.org/registry/cl/specs/opencl-2.0.pdf>
- [10] LUNOKHOD, Semi-Global Matching, [cit. 2014-05-26] Dostupné z WWW: <<http://lunokhod.org/?p=1356>>
- [11] CYGANEK, B.; SIEBERT, P., J.; *An Introduction to 3D Computer Vision Techniques and Algorithms*: John Wiley & Sons 2009, ISBN: 978-0-470-01704
- [12] OpenCV 2.4.9.0 documentation [online] [cit. 2014-05-20] Dostupné z WWW: <[>](http://docs.opencv.org/modules/contrib/doc/facerec/colormaps.html?highlight=color%20map#void_applyColorMap(InputArray_src,OutputArray_dst,int_colormap))

- [13] Configuring Image Processing Properties [online] [cit. 2014-05-21]  
Dostupné z WWW:  
<[http://www1.adept.com/main/KE/Data/ACE/AdeptSight\\_User/ImageProcessing\\_Operations.html](http://www1.adept.com/main/KE/Data/ACE/AdeptSight_User/ImageProcessing_Operations.html)>

## **10 SEZNAM PŘÍLOH NA DVD**

Zdrojové soubory

Zkompilované programy (Debug, Release) včetně ukázkové konfigurace a vstupních dat

Ukázková videa

Ukázkové fotografie