



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

STANOVENÍ POZICE OBJEKTU

DETECTION OF OBJECT POSITION

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Filip Baáš

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Miloslav Richter, Ph.D.

BRNO 2019



Diplomová práce

magisterský navazující studijní obor **Kybernetika, automatizace a měření**

Ústav automatizace a měřicí techniky

Student: Bc. Filip Baáš

ID: 173555

Ročník: 2

Akademický rok: 2018/19

NÁZEV TÉMATU:

Stanovení pozice objektu

POKYNY PRO VYPRACOVÁNÍ:

- 1) Nastudujte metody pro stanovení pozice objektu při snímání jednou kamerou.
- 2) Zvolte s vedoucím jednu z metod detekce (stanovte její výhody a nevýhody oproti ostatním) a tu realizujte.
- 3) Zjistěte současný stav řešení dané problematiky a případně dostupné komerční produkty či open source zdrojové texty vztahující se k této problematice.
- 4) Stanovte parametry měřicí sestavy - vzájemné polohy kamery a měřeného objektu, celkové rozměry.
- 5) Stanovte parametry a vlastnosti měřeného objektu pro úspěšné stanovení zjištěných parametrů.
- 6) Stanovte dosažitelnou přesnost/kvalitu.
- 7) Vytvořte materiály pro testování. Navrhněte a realizujte měřicí stanoviště, nasnímejte několik měřicích sad snímků pro testování algoritmů.
- 8) Navrhněte a realizujte algoritmy zvolené metody detekce polohy. Zhodnoťte dosažené výsledky a vhodnost metod pro různé typy objektů a jejich polohy v prostoru.

DOPORUČENÁ LITERATURA:

Žára J., Beneš B., Sochor J., Felkel P.: Moderní počítačová grafika, Computer Press, 2005, ISBN 978-80-2-1-0454-5

Sonka M., Hlavac V., Boyle R.: Image Processing, Analysis, and Machine Vision, 3rd Edition, CL Engineering, 2007, ISBN: 978-0495082521

Termín zadání: 4.2.2019

Termín odevzdání: 13.5.2019

Vedoucí práce: Ing. Miloslav Richter, Ph.D.

Konzultant:

doc. Ing. Václav Jirsík, 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.

ABSTRAKT

Diplomová práca sa zaoberá stanovením pozície objektu pomocou monokulárnej kamery. Za objekt je v tejto práci považované každé pevné, tvarovo nemenné teleso s výraznými hranami a ideálne bez textúr. Pozícia objektu je v tejto práci reprezentovaná transformačnou maticou, ktorá popisuje transláciu a rotáciu objektu voči svetovému súradnicovému systému. Prvá kapitola je venovaná vysvetleniu teórie geometrických transformácií a vnútorným a vonkajším parametrom kamery. Ďalej je v tejto kapitole popísaný detekčný algoritmus *Chamfer Matching*, ktorý je v práci použitý. Druhá kapitola popisuje vývojové nástroje použité pri riešení diplomovej práce. Tretia, štvrtá a piata kapitola sú venované praktickej realizácii zadania práce a dosiahnutým výsledkom. Posledná kapitola je venovaná popisu vytvorenej aplikácie, ktorá realizuje stanovenie pózy známeho objektu v scéne.

KLÚČOVÉ SLOVÁ

stanovenie pozície, detekcia objektu, chamfer matching, monokulárna kamera, kalibrácia kamery, Visual Studio, C++, OpenCV, OpenGL, Qt, MVTech HALCON

ABSTRACT

Master's thesis deals with object pose estimation using monocular camera. As an object is considered every rigid, shape fixed entity with strong edges, ideally textureless. Object position in this work is represented by transformation matrix, which describes object translation and rotation towards world coordinate system. First chapter is dedicated to explanation of theory of geometric transformations and intrinsic and extrinsic parameters of camera. This chapter also describes detection algorithm *Chamfer Matching*, which is used in this work. Second chapter describes all development tools used in this work. Third, fourth and fifth chapter are dedicated to practical realization of this works goal and achieved results. Last chapter describes created application, that realizes known object pose estimation in scene.

KEYWORDS

pose estimation, object detection, chamfer matching, monocular camera, camera calibration, Visual Studio, C++, OpenCV, OpenGL, Qt, MVTech HALCON

BIBLIOGRAFICKÁ CITÁCIA

BAÁŠ, Filip. *Stanovení pozice objektu* [online]. Brno, 2019 [cit. 2019-05-08]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/119310>. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Miloslav Richter.

VYHLÁSENIE

Vyhlasujem, že svoju diplomovú prácu na tému „Stanovení pozice objektu“ som vypracoval samostatne pod vedením vedúceho diplomovej práce, s využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej diplomovej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto diplomovej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a som si plne vedomý následkov porušenia ustanovenia §11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovení časti druhej, hlavy VI. diel 4 Trestného zákonníka Českej republiky č. 40/2009 Sb.

V Brne dňa:

.....

Filip Baáš, podpis

POĎAKOVANIE

Týmto spôsobom by som sa chcel poďakovať vedúcemu tejto diplomovej práce, pánovi Ing. Miloslavovi Richterovi, Ph.D., za odborné vedenie, rady a návrhy pri riešení práce, pohotovú komunikáciu a ochotu poradiť.

V Brne dňa:

.....

Filip Baáš, podpis

OBSAH

Úvod	1
1 Teória stanovenia pozície objektu	3
1.1 Geometrické transformácie	3
1.1.1 Translácia.....	3
1.1.2 Rotácia.....	4
1.1.3 Spájanie matíc	5
1.1.4 Prepočet bázy	5
1.1.5 Chyba stanovenia pózy	6
1.2 Kalibrácia kamery.....	6
1.2.1 Vnútorne parametre kamery.....	7
1.2.2 Vonkajšie parametre kamery	8
1.3 Metódy stanovenia pozície objektu	10
1.3.1 Chamfer Matching.....	10
1.3.2 Orientované hranové párovanie	13
1.4 Histogram	15
1.4.1 Priemer	15
1.4.2 Kvadratický priemer.....	16
1.4.3 Vážený priemer.....	16
1.4.4 Hausdorff.....	16
1.4.5 Hausdorff _{0,5}	16
1.4.6 Početnosť najvyššieho bin.....	17
2 Využitie vývojové nástroje	18
2.1 Visual Studio	18
2.1.1 Image Watch	18
2.2 OpenCV	18
2.3 OpenGL	19
2.4 Qt.....	19
2.5 MVTec HALCON.....	19
2.6 Calib.io.....	20
3 Kalibrácia kamery.....	21
3.1 Kamera.....	21
3.2 Vnútorne parametre kamery	22
3.2.1 Popis kalibračného pracoviska a spôsob kalibrácie.....	22
3.2.2 Priemerovanie snímok.....	23
3.2.3 Materiál kalibračného vzoru.....	23
3.2.4 OpenCV.....	25
3.2.5 Halcon.....	32

3.3	Vonkajšie parametre kamery.....	33
3.3.1	Popis kalibračného pracoviska.....	34
3.3.2	Návrh kalibračného vzoru	34
3.3.3	Detekcia kalibračného vzoru	35
3.3.4	Stanovenie pózy kamery	38
3.3.5	Stanovenie reprojekčnej chyby	38
4	Tvorba referenčnej sady snímok.....	39
4.1	Použité referenčné objekty	39
4.2	Vytvorené sady scén.....	41
4.3	Kontrola správnosti získaných parametrov kamery	43
4.4	Referenčná póza pomocou prípravku	45
4.5	Referenčná pozícia pomocou filtru Sobel.....	47
4.5.1	Popis algoritmu.....	47
5	Realizácia stanovenia pózy objektu.....	52
5.1	Chamfer Matching	52
5.1.1	Vytvorenie sady vzorov	52
5.1.2	Predspracovanie scény	55
5.1.3	Detekčný algoritmus.....	57
5.2	Porovnanie dosiahnutých výsledkov	58
5.2.1	Čisté scény	59
5.2.2	Scény s pohybom	60
5.2.3	Scény s rotovaným objektom	61
5.2.4	Vizualizácia výsledkov detekcie.....	62
5.3	Stanovenie pózy objektu voči svetovému súradnicovému systému.....	63
6	Popis vytvorenej aplikácie	64
6.1	Vstupné parametre	64
6.2	Nastavenie detekcie.....	66
6.3	Výstupné parametre.....	66
6.4	Vizualizácia detekcie	67
	Záver.....	68
	Literatúra	70
	Zoznam symbolov, veličín a skratiek.....	72
	Zoznam príloh.....	73
	Obsah priloženého DVD.....	74

Zoznam obrázkov

Obr. 1 Princíp stanovenia pozície Stanfordského králika	2
Obr. 2 Poloha bodu P v dvoch rôznych súradnicových systémoch.....	5
Obr. 3 Radiálne skreslenia kamery zľava: žiadne, súdkovité, poduškovité	8
Obr. 4 Zobrazenie transformačnej matice vonkajších parametrov	9
Obr. 5 Príklady vzorov použitých v článku [8]	11
Obr. 6 Vizualizácia hranového párovania [8]	13
Obr. 7 Spôsob vytvorenia orientovaného vzoru, prah nastavený na hodnotu 120 [22]	14
Obr. 8 Logitech HD Pro Webcam C920	21
Obr. 9 Pohyb kalibračného vzoru v scéne.....	22
Obr. 10 Kalibračný vzor – šachovnica	28
Obr. 11 Kalibračný vzor – kruhová mriežka.....	28
Obr. 12 Kalibračný vzor – asymetrická kruhová mriežka	29
Obr. 13 Kalibračný vzor - ChArUco	29
Obr. 14 Kalibračné vzory použité pre porovnanie presnosti kalibrácie.....	30
Obr. 15 Použité vzory pre kalibráciu pomocou knižnice Halcon.....	32
Obr. 16 Kalibračný vzor použitý na stanovenie pózy kamery	34
Obr. 17 Snímka kalibračného vzoru stolu	35
Obr. 18 Konvolučné jadro o rozmeroch 4×4	36
Obr. 19 Výsledok konvolúcie kalibračnej snímky stola s maskou.....	36
Obr. 20 3D zobrazenie konvolúcie kalibračnej snímky stola s maskou	36
Obr. 21 Výsledok detekcie kalibračného vzoru stola	37
Obr. 22 Scény bez ruchu	41
Obr. 23 Scény s pohybom	42
Obr. 24 Scény s rotáciou objektu v X a Y osi.....	42
Obr. 25 Vizualizácia správnosti stanovenia parametrov kamery	44
Obr. 26 Objekt umiestnený na prípravku	45
Obr. 27 Príklad stanovenia referenčnej pozície objektu.....	45
Obr. 28 Zobrazenie rôznych polôh renderu objektu na ploche stola.....	47
Obr. 29 Výsledok aplikácie filtru Sobel na snímku.....	48
Obr. 30 Mriežka použitá pri hľadaní referenčnej pózy objektu	49
Obr. 31 Dvadsaťsten a jeho tri iterácie triangulácie gule	53
Obr. 32 Príklad použitia dvadsaťstenu (červená) a jeho prvej iterácie (modrá) pri vytváraní vzoru [23]	53
Obr. 33 Príklad renderu objektu použitím knižnice OpenGL.....	54
Obr. 34 Zobrazenie rovnomernosti rozloženia kľúčových bodov.....	55
Obr. 35 Zobrazenie zachovania vzdialenosti kľúčových bodov	55
Obr. 36 Reťazec predspracovania scény.....	56

Obr. 37 Ukážka výpočtu spätnej chyby zľava: vzdialenostná mapa scény, vzdialenostná mapa vzoru, maska okolia objektu.....	58
Obr. 38 Vizualizácia robustnosti algoritmu v scénach s ruchom, zľava, referencia, Chamfer Matching algoritmus, Halcon.....	62
Obr. 39 Hlavné okno aplikácie	64
Obr. 40 Vstupné nastavenia aplikácie	65
Obr. 41 Nastavenia detekcie	66
Obr. 42 Výsledky detekcie	66
Obr. 43 Módy zobrazenia scény, zľava hora: pôvodná snímka, odtiene sivej, hranová snímka, vzdialenostná mapa.....	67
Obr. 44 Módy zobrazenia detekcie, zľava: súradnicový systém, kľúčové body.....	67

Zoznam grafov

Graf 1 Histogram vytvorený párovaním z Obr. 6.....	13
Graf 2 Príklad metódy Hausdorff.....	16
Graf 3 Kumulovaný histogram použitý v metóde Hausdorff _{0,5}	17
Graf 4 Príklad metódy najpočetnejšieho binu.....	17
Graf 5 Vývoj reprojekčnej chyby kalibrácie v závislosti od počtu priemerovaných snímok.....	23
Graf 6 Porovnanie reprojekčnej chyby kalibrácie v závislosti od použitého kalibračného materiálu.....	25
Graf 7 Porovnanie časovej náročnosti kalibrácie kamery pri použití rôznych modelov kamery.....	26
Graf 8 Porovnanie rýchlosti odstránenia skreslenia kamery pri použití rôznych modelov kamery (vyššie je horšie).....	27
Graf 9 Priemerný čas detekcie vzoru v snímke.....	30
Graf 10 Priemerná reprojekčná chyba detekcie vzoru v jednej snímke.....	31
Graf 11 Výsledná reprojekčná chyba kalibrácie kamery.....	31
Graf 12 Chyba stanovenia výšky objektu nad stolom.....	46
Graf 13 Chyba stanovenia rotácií okolo osí X a Y.....	46
Graf 14 Chyba pozícií objektu stanovených manuálne a pomocou prípravku.....	50
Graf 15 Chyba rotácií objektu stanovených manuálne a pomocou prípravku.....	51
Graf 16 Percentuálna presnosť detekcie v čistých scénach.....	59
Graf 17 Chyba stanovenia pozície v čistých scénach.....	59
Graf 18 Percentuálna presnosť detekcie v scéne s ruchom.....	60
Graf 19 Chyba stanovenia pozície v scénach s ruchom.....	60
Graf 20 Percentuálna presnosť detekcie v scéne s rotovaným objektom.....	61
Graf 21 Chyba stanovenia pozície v scénach s rotovaným objektom.....	61

Zoznam výpisov

Výpis 1 Sekvenčný vzdialenostný algoritmus.....	12
Výpis 2 Použitie funkcie calibrateCamera.....	25
Výpis 3 Príklad detekcie vzoru – šachovnica.....	28
Výpis 4 Príklad detekcie vzoru – kruhová mriežka	28
Výpis 5 Príklad detekcie vzoru – asymetrická kruhová mriežka	29
Výpis 6 Príklad stanovenia pózy kamery.....	38
Výpis 7 Príklad aplikácie filtru Sobel na snímku [20].....	48
Výpis 8 Príklad výpočtu zhody stanovenej pózy s objektom v scéne	49
Výpis 9 Pseudokód algoritmu hľadania referenčnej pózy objektu.....	49
Výpis 10 Predspracovanie scény a vytvorenie vzdialenostnej mapy	56
Výpis 11 Spätný výpočet chyby kvadratickým priemerom	58
Výpis 12 Výpočet pozície objektu	63

ÚVOD

Ľudský zrak patrí medzi jeden z najdôležitejších zmyslov, pretože človek prijíma svojím zrakom až 70% informácií o svojom okolí. V dnešnej dobe narastá snaha tento ľudský zmysel využiť v priemysle vo forme kamier. Táto snaha je často zúročená zrýchlením kontrolných procesov, spresnením výroby či znížením chybovosti.

Napriek tomu, že v súčasnosti sa čím ďalej tým viac rozrastá trh s 3D kamerami, sú tieto, často lacné, senzory nepoužiteľné v priemysle, kde sú priemyselné riešenia oveľa drahšie a majú väčšie rozmery. Tieto kamery sú taktiež často náchylné na špecifické vlastnosti objektov, napríklad lesklý povrch materiálu. Preto je táto diplomová práca zameraná na stanovenie pozície objektu voči svetovému súradnicovému systému za pomoci monokulárnej kamery. To znamená stanovenie polohy a orientácie objektu v definovanom súradnicovom systéme za použitia jednej kamery.

Objekt

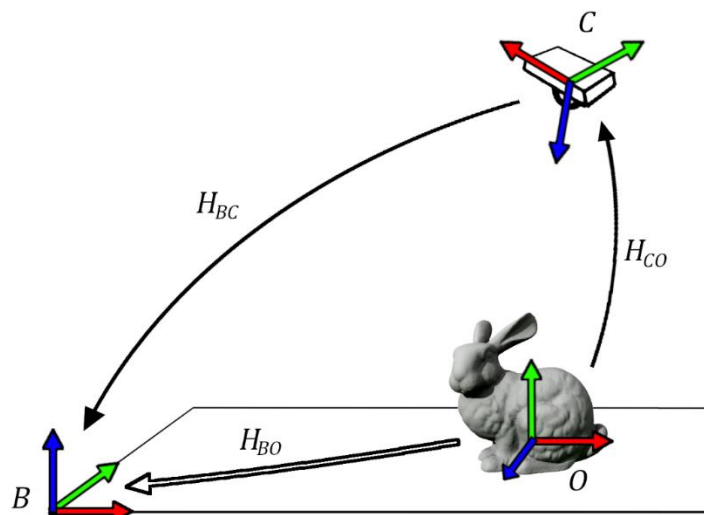
Objekt je v tejto práci vnímaný ako zostrojiteľné, priestorové, pevné teleso. Objekt je ideálne bez textúr, má ostré, viditeľné hrany a je čo najmenej symetrický. Tento objekt má vytvorenú svoju virtuálnu kópiu, ktorá má rovnaké rozmery ako jeho fyzická verzia. Ako formát, v ktorom je objekt uložený, bol pre túto prácu zvolený formát *OBJ* kvôli jeho jednoduchosti a vhodnosti.

Detekcia objektu

Na nájdenie objektu v snímke je potrebné zvoliť algoritmus, ktorý je schopný správne určiť pozíciu objektu v scéne a následne určiť jeho pózu voči kamere. Pri riešení práce bol použitý algoritmus *Chamfer Matching*, ktorý je ľahký na pochopenie a jednoducho implementovateľný. Keďže algoritmus je sám o sebe náchylný na falošné detekcie, je v tejto práci použitých niekoľko vylepšení, ktoré zvyšujú jeho robustnosť.

Pozícia objektu

Stanovenie pozície objektu je v tejto práci vnímané ako hľadanie transformačnej matice H_{BO} , ktorá popisuje polohu a orientáciu súradnicového systému objektu O voči bázovému súradnicovému systému B , ktorý býva taktiež nazvaný ako svetový súradnicový systém. Výpočet tejto matice je v tejto práci vykonaný pomocou zistenia transformačných matíc H_{CO} a H_{BC} . Spôsob stanovenia pozície je znázornený na Obr. 1.



Obr. 1 Princíp stanovenia pozície Stanfordského králika

Práca je rozdelená do šiestich častí. Prvá časť sa venuje teoretickému popisu práce. Nachádza sa v nej popis geometrických transformácií, vnútornej a vonkajšej kalibrácie kamery a popis detekčných algoritmov. V druhej časti sú popísané vývojové nástroje použité pri riešení tejto práce a dôvody ich výberu. Tretia časť sa venuje vnútornej a vonkajšej kalibrácii kamery. Vo štvrtej časti je popísaná metóda stanovenia referenčnej pózy objektu v scéne. Piata časť popisuje praktickú realizáciu detekcie a v šiestej časti je popísaná aplikácia vytvorená v tejto práci.

1 TEÓRIA STANOVENIA POZÍCIE OBJEKTU

Táto kapitola teoreticky rozoberá najdôležitejšie časti stanovovania pozície objektu v priestore a je rozdelená na štyri podkapitoly. V podkapitole 1.1 sú popísané priestorové geometrické transformácie, spôsob prepočtu bázy súradnicového systému a výpočet chyby stanovovania pózy. Podkapitola 1.2 je venovaná popisu vnútorných a vonkajších parametrov kamery. V podkapitole 1.3 sú popísané metódy detekcie objektu a podkapitola 1.4 bližšie popisuje detekčnú metódu zvolenú v tejto práci.

1.1 Geometrické transformácie

Geometrické transformácie v 3D priestore je možné reprezentovať transformačnými maticami. Transformačné matice vykonávajú lineárnu geometrickú transformáciu bodu P do bodu P' . Tieto matice sú s výhodou reprezentované v homogénnom tvare, ktorý zaisťuje vyjadrenie transformácií pomocou jednej matice A , čo v kartézskych súradniciach nie je možné. V tejto práci budú popísané iba translačné a rotačné transformačné matice, ktoré sú potrebné pri riešení zadania práce. [14]

$$P' = \begin{pmatrix} x' \\ y' \\ z' \\ w' \end{pmatrix} = A \cdot P = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} \quad (1)$$

1.1.1 Translácia

Translačné matice, alebo matice posuvu menia polohu bodu P o vektor posunu $\vec{t} = (t_x, t_y, t_z)$.

$$Trans(\vec{t}) = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2)$$

1.1.2 Rotácia

Rotačné matice, alebo matice otáčania, otáčajú bod P okolo súradnicových osí X , Y , a Z o uhly α , β , a γ .

$$Rot_X(\alpha) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3)$$

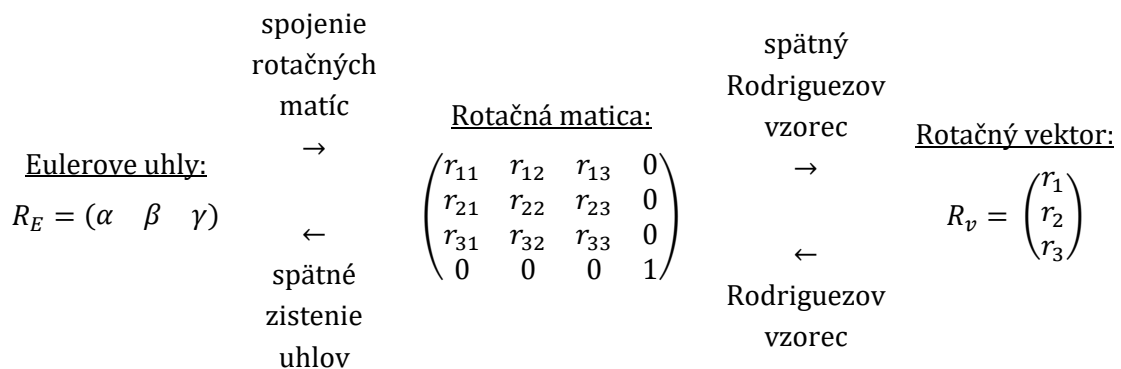
$$Rot_Y(\beta) = \begin{pmatrix} \cos \beta & 0 & \sin \beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \beta & 0 & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4)$$

$$Rot_Z(\gamma) = \begin{pmatrix} \cos \gamma & -\sin \gamma & 0 & 0 \\ \sin \gamma & \cos \gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5)$$

Knižnica *OpenCV*, ktorá je v tejto práci použitá, využíva pre vyjadrenie rotácie, namiesto rotačnej matice, rotačný vektor R_v , pozostávajúci z troch zložiek. Tieto zložky sa často chybne považujú za uhly rotácie okolo súradnicových osí X , Y , a Z . Pre ich prepočet do rotačnej matice je nutné použiť Rodrigueзов vzorec. Pre spätný prepočet rotačnej matice do rotačného vektoru existuje spätný Rodrigueзов vzorec. [11]

$$R_v = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix} \quad (6)$$

V tejto práci bol ako základný tvar vyjadrenia rotácie objektu zvolené Eulerove uhly aplikované v poradí rotácií okolo X , Y a Z osí. Dôvodom bola ľahšia predstava rotácie objektu pri použití tejto reprezentácie a jednoduché porovnanie dvoch rôznych rotácií pri výpočte chyby pózy. Spôsoby prevodu medzi jednotlivými vyjadreniami rotácií sú uvedené nižšie:



1.1.3 Spájanie matíc

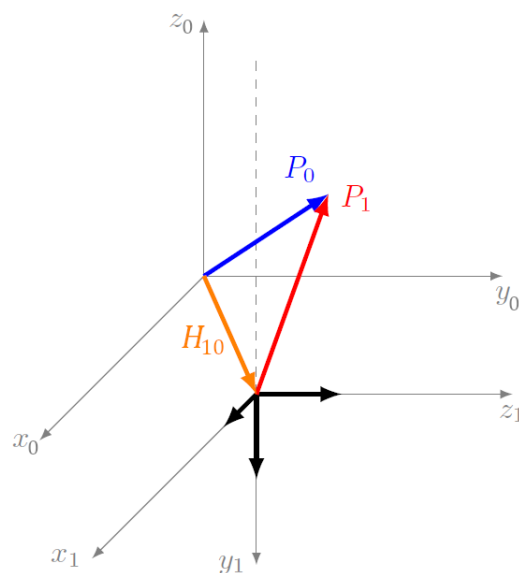
Pre spojenie viacerých transformačných matíc do jednej transformačnej matice A sa používa násobenie matíc. Poradie matíc určuje poradie, akým sú jednotlivé transformácie vykonané. Matice však musia byť v opačnom poradí, než v akom chceme, aby boli operácie vykonané. Pre vytvorenie transformačnej matice, ktorá bod P otočí okolo súradnicových osí X , Y , a Z a následne bod posunie o vektor $\vec{t} = (t_x, t_y, t_z)$, bude poradie matíc nasledovné [12]

$$A = Trans(\vec{t}).Rot_Z(\gamma).Rot_Y(\beta).Rot_X(\alpha). \quad (7)$$

1.1.4 Prepočet bázy

Prepočet bázy je prepočet polohy a orientácie bodu P v súradnicovom systéme 0 do polohy a orientácie v súradnicovom systéme 1. Pre tento výpočet je nutné poznať vzťah medzi súradnicovými systémami, ktorý je popísaný transformačnou maticou H_{10} . Táto matica je matica prevodu súradníc systému 0 do systému 1. Výpočet polohy a orientácie bodu P_1 je nasledovný [12]

$$P_1 = H_{10} \cdot P_0. \quad (8)$$



Obr. 2 Poloha bodu P v dvoch rôznych súradnicových systémoch

1.1.5 Chyba stanovenia pózy

Keďže sa táto práca zaoberá stanovením presnej pózy objektu v scéne, bolo potrebné definovať výpočet chyby stanovenia pózy voči referenčnej póze. Chyba pózy bola po dohode s vedúcim tejto práce rozdelená do chyby pozície E_t a chyby rotácie E_R .

Chyba pozície je vypočítaná ako absolútna vzdialenosť medzi pozíciou referenčnej pózy P_R a meranej pózy P_M použitím vzťahu

$$E_t = \sqrt{(P_{Rx} - P_{Mx})^2 + (P_{Ry} - P_{My})^2 + (P_{Rz} - P_{Mz})^2} \text{ [m]}. \quad (9)$$

Chyba rotácie je vypočítaná podobne ako chyba pozície, ale namiesto translačných súradníc sú použité hodnoty Eulerových uhlov.

$$E_R = \sqrt{(P_{R\alpha} - P_{M\alpha})^2 + (P_{R\beta} - P_{M\beta})^2 + (P_{R\gamma} - P_{M\gamma})^2} \text{ [}^\circ\text{]} \quad (10)$$

Pred použitím vzťahu je potrebné každú porovnávanú dvojicu uhlov normalizovať tak, aby absolútna hodnota ich rozdielu nebola väčšia ako 180° . To sa dá docieľiť napríklad pričítaním uhlu 360° k menšiemu uhlu z dvojice.

1.2 Kalibrácia kamery

Kalibrácia kamery je v tejto práci veľmi dôležitý krok a pozostáva z určenia projekčnej matice M , pomocou ktorej je možné popísať parametre kamery popísanej modelom dierkovej komory. Táto matica pozostáva z matice K , ktorá popisuje vnútorné parametre kamery a matice $(R|t)$, ktorá popisuje vonkajšie parametre kamery. Projekcia bodu je daná vzťahom [2]

$$w \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = M \begin{pmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{pmatrix} = K(R|t) \begin{pmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{pmatrix}. \quad (11)$$

Súradnice X_W, Y_W, Z_W sú 3D súradnice bodu vo svetovom súradnicovom systéme, u, v sú súradnice priemetu bodu na rovinu kamery a parameter w je váhový parameter zaisťujúci homogenitu pre súradnice priemetu. [9]

Zistením vnútorných parametrov kamery je možné presne určiť, ako bude bod v danej pozícii pred kamerou premietaný na rovinu kamery. Takýmto spôsobom je možné v tejto práci vytvoriť vzor objektu, ktorý sa bude zhodovať s projekciou reálneho modelu na rovinu kamery.

Pomocou vonkajších parametrov kamery je následne možné prepočítať súradnice detegovaného objektu zo súradnicového systému kamery do svetového súradnicového systému.

1.2.1 Vnútorne parametre kamery

Vnútorne, alebo intrinzické, parametre kamery popisujú vnútornú geometriu a optickú charakteristiku danej kamery a delia sa na lineárne a nelineárne.

1.2.1.1 Lineárne

Lineárne vnútorné parametre kamery sa dajú vyjadriť maticou

$$K = \begin{pmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \quad (12)$$

a patrí medzi nich ohnisková vzdialenosť kamery f a optický stred kamery c . Niekedy je uvádzaný aj parameter skosenia s , ktorý je nenulový, ak osi kamery nie sú pravouhlé. Táto matica je viazaná na použité rozlíšenie kamery. [3]

Ohniskové vzdialenosti f sú udávané v pixeloch. Pre ich prepočet je možné použiť vzťah

$$F_{x/y} = f_{x/y} \frac{D_{x/y}}{d_{x/y}} [m], \quad (13)$$

kde F je hľadaná ohnisková vzdialenosť, d je rozmer snímky v pixeloch a D je rozmer snímacieho čipu. [5]

1.2.1.2 Nelineárne

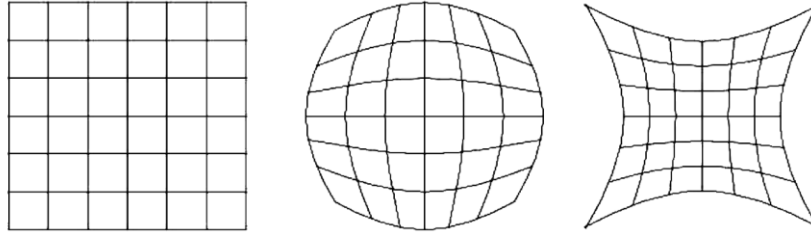
Medzi nelineárne vnútorné parametre kamery patria koeficienty skreslenia kamery. Tieto koeficienty popisujú, akým spôsobom je obraz skreslený vplyvom nedokonalosti šošoviek kamery.

V knižnici *OpenCV*, ktorá je v tejto práci použitá, sú tieto koeficienty popísané v matici pozostávajúcej z ôsmich prvkov. Tieto koeficienty sú rovnaké pre všetky rozlíšenia kamery, delia sa na radiálne k_n a tangenciálne p_n , a sú uložené vo vektore skreslenia [6]

$$Distortion_{coefficients} = (k_1, k_2, p_1, p_2 [k_3 [k_4, k_5, k_6]]).$$

Radiálne skreslenie je u kamier najčastejším druhom skreslenia. Spôsobuje súdkovitost' alebo poduškovitost' a snímku je možné zbaviť tohto skreslenia použitím rovníc

$$\begin{aligned}x' &= x(1 + k_1r^2 + k_2r^4 + k_3r^6), \\y' &= y(1 + k_1r^2 + k_2r^4 + k_3r^6).\end{aligned}\tag{14}$$



Obr. 3 Radiálne skreslenia kamery zľava: žiadne, súdkovité, poduškovité

Tangenciálne skreslenie je spôsobené tým, že šošovky kamery nie sú dokonale rovnobežné s rovinou priemetu (čipom kamery) a toto skreslenie je možné odstrániť použitím rovníc

$$\begin{aligned}x' &= x + [2p_1xy + p_2(r^2 + 2x^2)], \\y' &= y + [p_1(r^2 + 2y^2) + 2p_2xy].\end{aligned}\tag{15}$$

Ďalšia knižnica použitá v tejto práci s názvom *Halcon* využíva pre odstránenie skreslenia šošovky iba jeden parameter *Kappa*. Tento parameter popisuje radiálne skreslenie kamery a má jednotku m^{-2} .

1.2.2 Vonkajšie parametre kamery

Vonkajšie parametre, taktiež nazývané ako extrinzičné parametre kamery, popisujú vzťah medzi súradnicovým systémom kamery C a svetovým súradnicovým systémom B , a sú popísané maticou

$$(R|t) = \begin{pmatrix} r_{11} & r_{12} & r_{13} & | & t_x \\ r_{21} & r_{22} & r_{23} & | & t_y \\ r_{31} & r_{32} & r_{33} & | & t_z \end{pmatrix}.\tag{16}$$

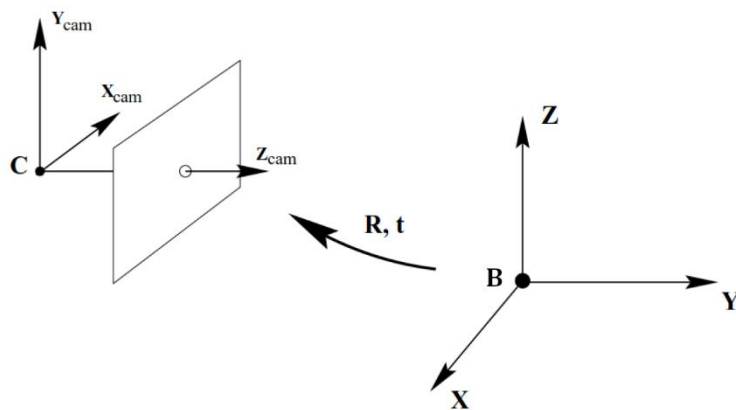
Táto matica vznikne spojením rotačnej matice v nehomogénnom tvare a translačného vektora. Rotačná matica vznikne spojením troch základných rotácií súradnicového systému okolo osí X , Y , a Z . Translačný vektor pozostáva z troch základných súradníc popisujúcich transláciu svetového súradnicového systému voči súradnicovému systému kamery. Rozšírením tejto matice do homogénneho tvaru je možné získať maticu H_{CB} , popísanú v Obr. 1. [13]

$$H_{CB} = \begin{pmatrix} R & | & t \\ 0 & 0 & 0 & | & 1 \end{pmatrix}\tag{17}$$

Matica H_{CB} je často nesprávne považovaná za pozíciu kamery vo svetovom súradnicovom systéme B . Pre získanie pozície kamery voči svetovému súradnicovému systému je nutné maticu H_{CB} invertovať a tak získať maticu H_{BC} . [9]

$$H_{BC} = H_{CB}^{-1} = \left(\begin{array}{c|c} R^T & -R^T t \\ \hline 0 & 1 \end{array} \right) \quad (18)$$

Vo vyššie uvedenom výpočte je využitá ortogonálna vlastnosť rotačných matic, kde $Q^T Q = Q Q^T = I$. Inverziu týchto matic je možné vypočítať ako $Q^{-1} = Q^T$.



Obr. 4 Zobrazenie transformačnej matice vonkajších parametrov

Vonkajšie parametre kamery je možné zistiť riešením Perspective- n -Point (PnP) problému. Tento výpočet s použitím n 3D referenčných bodov vo svetových súradniciach a ich n 2D projekcií v snímke určí pózu danej kamery. Vnútorne parametre tejto kamery musia byť predom plne známe a pre jednoznačnosť riešenia musí byť splnená podmienka $n \geq 6$. Táto podmienka vychádza z postupného znižovania stupňov voľnosti objektu v scéne a znížením počtu riešení problému na jedno. V Tabuľka 1 je vysvetlená podmienka minimálneho počtu bodov, ktoré sú použité pri riešení problému PnP. Pri použití 4 alebo 5 bodov môže stále nastať prípad, kedy má problém viac ako 1 riešenie, a to vtedy, ak sú body umiestnené na rovnakej ploche. [10]

Tabuľka 1 Zobrazenie počtu riešení problému PnP pri použití rôzneho počtu bodov

n	Počet stupňov voľnosti	Počet riešení
0	6	∞
1	4	∞
2	2	∞
3	0	2/4
4, 5	0	1/>1
≥ 6	0	1

Pre správne určenie pózy kamery nesmú body ležať v jednom bode ani na rovnakej priamke. Pre čo najpresnejšie stanovenie vonkajších parametrov kamery sa odporúča rozmiestnenie bodov s variáciou vo všetkých súradnicových osiach a zvýšenie počtu bodov minimálne na 6.

1.3 Metódy stanovenia pozície objektu

Táto podkapitola popisuje dostupné metódy stanovenia pozície objektu v scéne a ich princíp. Keďže je táto práca venovaná detekcii objektov použitím monokulárnej kamery, sú v tejto podkapitole vynechané algoritmy využívajúce stereo kamery, prípadne algoritmy s použitím viacerých monokulárnych kamier.

Veľká kapitola počítačového videnia je venovaná popisu objektov pomocou priradovania príznakov. Táto kapitola je však venovaná uzavretým objektom, ktoré je možné popísať jednou kontúrou. Objekty, ktoré sú v tejto práci detegované často nie je možné popísať jednou kontúrou, a preto je popis týchto metód stanovenia pozície objektu v práci vynechaný.

Metóda stanovenia pozície objektu použitím konvolučných neurónových sietí v tejto práci taktiež nie je spomenutá. Konvolučné siete síce dokážu stanoviť pózu objektu, no ich výpočtová náročnosť je vysoká. Ďalšou nevýhodou je iba približné stanovenie pózy objektu pri použití monokulárnej kamery, a preto sa často používajú v spojení s hĺbkovým sensorom, ktorý v tejto práci nie je použitý. V neposlednom rade sa pri neurónových sieťach objavuje nutnosť vytvárania rozsiahlych datasetov objektu a časovo náročné učenie neurónovej siete.

Na detekciu pózy objektu sú najvhodnejšie algoritmy založené na vytváraní vzorov, ktoré reprezentujú 2D projekciu hrán detegovaného objektu. Tieto algoritmy vykazujú najvyššiu presnosť a rýchlosť. V tejto podkapitole sú popísané dve metódy určené pre detekciu objektov. [21]

1.3.1 Chamfer Matching

Táto metóda bola prvýkrát popísaná v roku 1977 v článku [7] ako metóda hľadania najväčšej zhody hrán pri porovnávaní dvoch snímok. Originálna verzia tohto

algoritmu je ale použitá iba v obmedzenom množstve aplikácií a v roku 1988 bol švédskou matematickou Gunilla Borgefors v článku [8] popísaný návrh vylepšenia tohto algoritmu určený na detekciu objektov. Metóda pozostáva z troch častí, vytvorenie vzoru objektu, predspracovanie scény a hľadanie vzoru v scéne.

Metóda patrí medzi najjednoduchšie a najrýchlejšie metódy porovnávania vzoru so scénou. Pozostáva z predspracovania scény a konvolúcie kľúčových bodov vzoru so scénou. Táto metóda je však v základe málo odolná voči ruchu v scéne a závislá od správnosti spracovania scény (detekcie hrán), čo spôsobuje veľké množstvo falošných detekcií. Taktiež táto metóda stráca presnosť pri pohybe objektu či zmene jas v scéne.

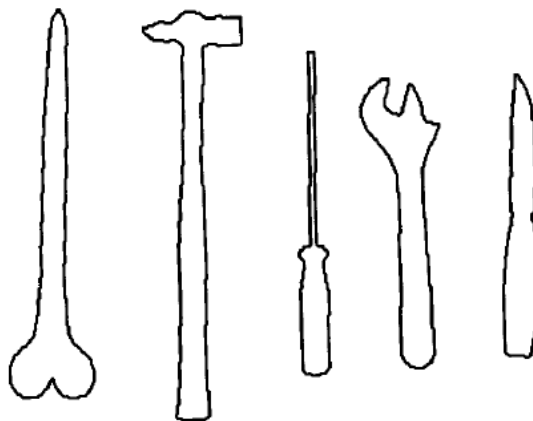
1.3.1.1 Vytvorenie vzoru

K vytvoreniu vzoru sú v článku použité fotky objektov, na ktoré je aplikovaný hranový filter. Z takýchto hranových snímok je následne extrahovaných približne 600 bodov, ktorých pozície reprezentujú pozície hrán objektu. Pre správnu detekciu objektu je v článku poukázané na to, že body musia byť rozmiestnené rovnomerne po celom objekte a nie iba v jednej jeho časti.

Vzor je v tejto metóde reprezentovaný ako

$$T: \{x_i, y_i | i = 1, \dots, n\}, \quad (19)$$

kde n reprezentuje počet bodov vzoru.



Obr. 5 Príklady vzorov použitých v článku [8]

1.3.1.2 Predspracovanie scény

Predspracovanie scény pozostáva z aplikovania hranového filtra a vytvorenia vzdialenostnej mapy. Vzdialenostná mapa je vytvorená použitím sekvenčného vzdialenostného algoritmu, ktorý je známy pod názvom „chamfer“. Vzdialenosť, s ktorou algoritmus počíta, je oproti vzdialenosti mestských blokov alebo šachovnicovej vzdialenosti presnejšia a v porovnaní s euklidovskou vzdialenosťou je výpočtovo menej náročná.

V prvom kroku sa jasová hodnota hranových pixelov nastaví na hodnotu 0 a zvyšné pixely sú nastavené na maximálnu jasovú hodnotu. Následne algoritmus dvakrát prechádza snímku. Prvý, dopredný prechod je zľava doprava a zhora nadol. Druhý, spätný prechod, je sprava doľava a zdola nahor. Algoritmus je popísaný vo Výpise 1. Dôvod nastavenia minimálneho kroku algoritmu na 3 je vysvetlený v nasledujúcom odseku.

Výpis 1 Sekvenčný vzdialenostný algoritmus

```
//Dopredný prechod:
for (i = 2; i < rows; i++)
{
    for j = 2; j < columns; j++)
    {
        vi,j = minimum(vi-1,j-1 + 4, vi-1,j + 3, vi-1,j+1 + 4, vi,j-1 + 3, vi,j)
    }
}

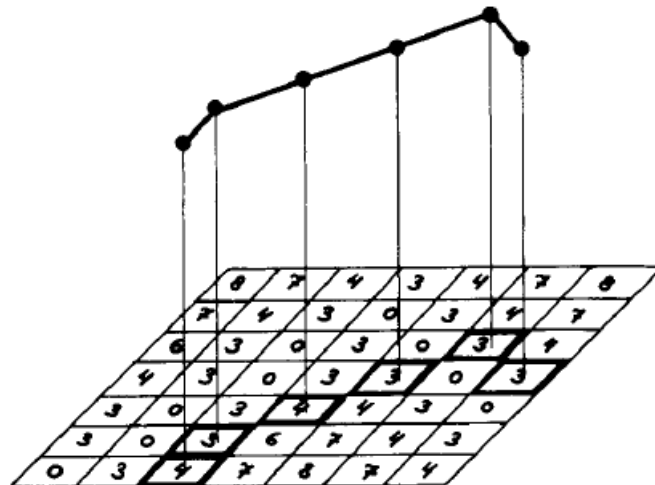
//Spätný prechod:
for (i = rows - 1; i > 1; i--)
{
    for (j = columns - 1; j > 1; j--)
    {
        vi,j = minimum(vi,j, vi,j+1 + 3, vi+1,j-1 + 4, vi+1,j + 3, vi+1,j+1 + 4).
    }
}
```

1.3.1.3 Párovanie

Detekcia objektu prebieha výpočtom konvolúcie vzdialenostnej mapy a vzoru. Pre každú pozíciu vzoru na vzdialenostnej mape je vypočítaná hranová vzdialenosť. Tá by mala pri správnej pozícii vzoru mať globálne minimálnu hodnotu. Pre jej výpočet je v tomto článku zvolený vzťah:

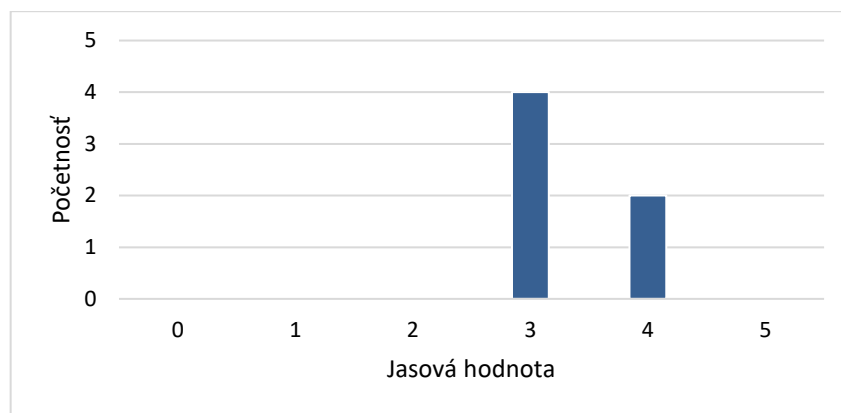
$$D_{R.M.S.} = \frac{1}{3} \sqrt{\frac{1}{n} \sum_{i=1}^n v_i^2}, \quad (20)$$

v ktorom v_i predstavuje jasové hodnoty pixelov vzdialenostnej mapy, ktoré sú prekryté vzorom. Vydelením výsledku 3 je zabezpečená normalizácia vzdialenostnej mapy. Takýmto prístupom je možné v celočíselnej snímke dosiahnuť vyššiu presnosť určenia vzdialenosti, na rozdiel od použitia metriky mestských blokov alebo šachovnicovej metriky. Dôvodom použitia kvadratického priemeru vo vzťahu je zníženie falošných lokálnych miním vo výsledku konvolúcie. Spôsob detekcie tvaru je zobrazený na Obr. 6.



Obr. 6 Vizualizácia hranového párovania [8]

Túto metódu je možné vďaka porovnávaniu jasových hodnôt upraviť na metódu založenú na práci s histogramom. To znamená, že výsledok každého hranového párovania je možné reprezentovať histogramom. Následne je možné aplikovať rôzne štatistické metódy a porovnať ich presnosti detekcie. Histogram, ktorý vznikne pri párovaní zobrazenom na Obr. 6, je zobrazený v Graf 1.



Graf 1 Histogram vytvorený párovaním z Obr. 6

1.3.2 Orientované hranové párovanie

Predošlá popísaná metóda bola založená na hľadaní hrán scény a porovnávaní s hranami objektu. Táto metóda však nezohľadňovala orientáciu hrán objektu. Metóda *Orientovaného hranového párovania* využíva pri detekcii parameter orientácie hrany, čím zvyšuje odolnosť metódy voči šumu. V tejto metóde je využitý filter Sobel, pomocou ktorého je možné zvýrazniť hrany v scéne a taktiež vypočítať ich frekvenciu a orientáciu. [21][22]

1.3.2.1 Vytvorenie vzoru

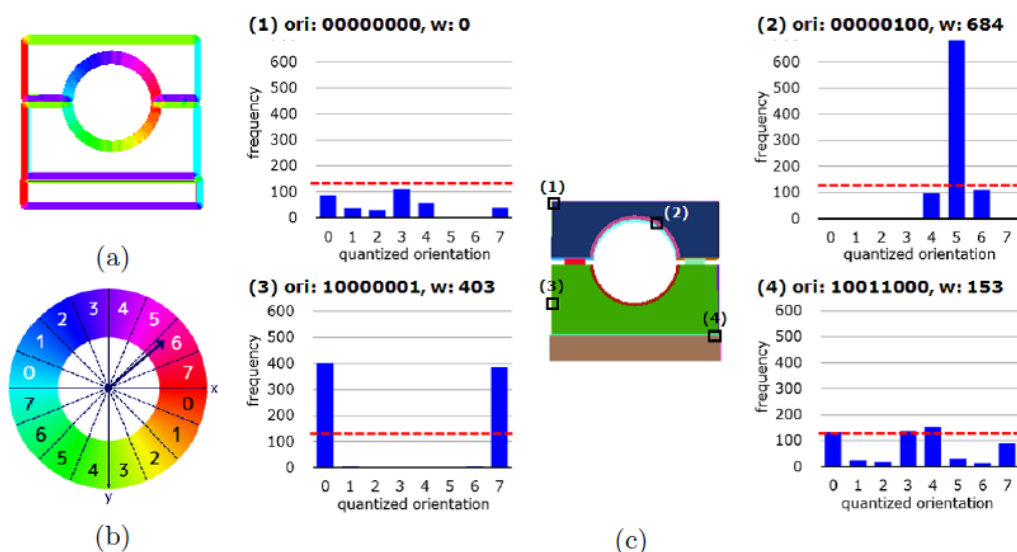
Vzor je v tejto metóde opäť založený na hranovej reprezentácii, ale je mu pridaný parameter orientácie hrany v danom bode a váha hrany. Objekt je pre tieto účely vymodelovaný v CAD programe a následne je použitím grafickej knižnice vytvorený render objektu v požadovanej póze. Na takýto render je aplikovaný filter Sobel. Výsledok aplikovania filtru je zobrazený na Obr. 7(a), kde farba hrany závisí od jej orientácie podľa Obr. 7(b). Na Obr. 7(c) sú náhodne zvolené 4 body na objekte a zobrazené ich frekvencie a váhy w .

Pri tvorbe vzoru sú náhodne vyberané body na objekte. Ak frekvencia daného bodu prekročí stanovený prah aspoň v jednej z orientácií, je tento bod vložený do vzoru. Orientácia ori tohto bodu je 8-bitové číslo, ktoré má nastavené všetky bity v závislosti od orientácií ktoré, prekročili frekvenčný prah. Váha takéhoto bodu je udaná ako maximálna frekvencia tohto bodu. Takýmto spôsobom je vytvorený orientovaný vzor ori^T .

Vzor je v tejto metóde reprezentovaný nasledovne:

$$T: \{x_i, y_i, ori_i, w_i | i = 1, \dots, n\}, \quad (21)$$

kde n reprezentuje počet bodov vzoru.



Obr. 7 Spôsob vytvorenia orientovaného vzoru, prah nastavený na hodnotu 120 [22]

1.3.2.2 Predspracovanie scény

Pri predspracovaní scény je, rovnako ako pri tvorbe vzoru, využitý filter Sobel. Pomocou neho sa rovnakým spôsobom zisťuje frekvencia a orientácia hrán v scéne, pričom je však kontrolovaný každý jeden pixel scény. Ak hodnota frekvencie aspoň v jednej z orientácií prekročí požadovaný frekvenčný prah, bit na pozícii danej orientácie sa nastaví na hodnotu 1, inak je hodnota bitu nastavená na 0. Týmto

spôsobom je vytvorená orientovaná scéna ori^l , ktorej pixel je reprezentovaný 8 - bitovým číslom.

1.3.2.3 Párovanie

Detekcia objektu prebieha výpočtom konvolúcie predspracovanej scény a vzoru. Pre každú pozíciu vzoru v scéne je vypočítané skóre. To by malo pri správnej pozícii vzoru v scéne dosahovať globálne maximálnu hodnotu. Jeho výpočet je nasledovný

$$score(x, y) = \frac{\sum_{j=1}^n \delta_j(ori^l_{(x+x_j, y+y_j)} \in ori^t_j)}{\sum_{j=1}^n w_j}, \quad (22)$$

kde funkcia δ_j porovnáva orientáciu bodov v danej pozícii použitím bitovej operácie AND. Pri zhode aspoň v jednej z orientácií je k aktuálnej hodnote skóre pričítaná váha daného bodu vzoru podľa:

$$\delta_j(ori^l \in ori^t) = \begin{cases} w_j & \text{ak } ori^l \wedge ori^t > 0, \\ 0 & \text{inak.} \end{cases} \quad (23)$$

Po dohode s vedúcim tejto práce bola ako metóda detekcie objektu zvolená metóda *Chamfer Matching*, ktorá spĺňala všetky podmienky požadované pre stanovenie pozície objektu v scéne a je možné ju upraviť na prácu s histogramom, čím je možné dosiahnuť vyšší počet detekčných metód.

1.4 Histogram

Keďže ako detekčná metóda bola v tejto práci použitá metóda *Chamfer Matching*, sú v tejto podkapitole popísané metódy použité pre spracovanie histogramu h , ktorý vzniká pri hranovom párovaní. Jednotlivé stĺpce histogramu sa taktiež nazývajú biny a počet prvkov alebo početnosť v bine s indexom i je reprezentovaná ako h_i . Veľkosť jedného binu udáva počet prvkov s danou vzdialenosťou od hrany. Parameter n označuje počet všetkých prvkov v histograme a parameter n_h celkový počet binov histogramu.

Jednotlivé metódy udávajú spôsob výpočtu parametru D , ktorý popisuje celkovú vzdialenosť vzoru od optimálnej polohy. Pri detekcii je požiadavka na nájdenie minimálnej hodnoty tohto parametra.

1.4.1 Priemer

Táto metóda počíta priemernú hodnotu v histograme h .

$$D = \frac{1}{n} \sum_{i=0}^{n_h} h_i \cdot i \quad (24)$$

1.4.2 Kvadratický priemer

Metóda je založená na výpočte kvadratického priemeru z histogramu h .

$$D = \sqrt{\frac{1}{n} \sum_{i=0}^{n_h} h_i \cdot i^2} \quad (25)$$

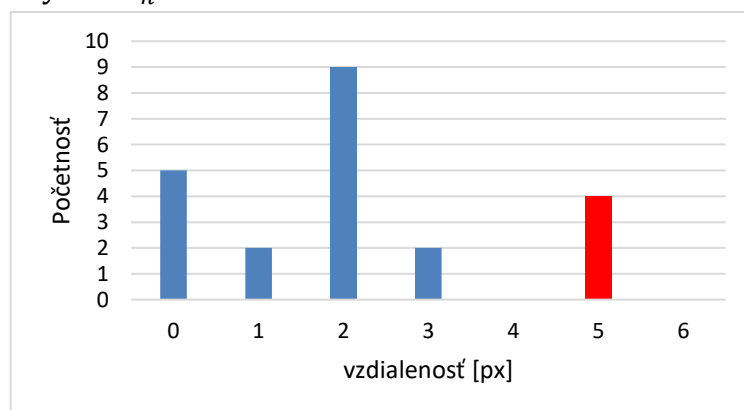
1.4.3 Vážený priemer

Táto metóda váži kvadratickú početnosť binov v závislosti od ich pozície v histograme h . Prevrátená hodnota je vo vzťahu použitá kvôli priradeniu nižšej hodnoty menšej vzdialenosti a naopak.

$$D = \frac{1}{\frac{1}{n} \sum_{i=0}^{n_h} \frac{1}{i+1} h_i^2} \quad (26)$$

1.4.4 Hausdorff

V tejto metóde je parameter D nastavený ako index posledného binu histogramu, ktorý je označený ako n_h .



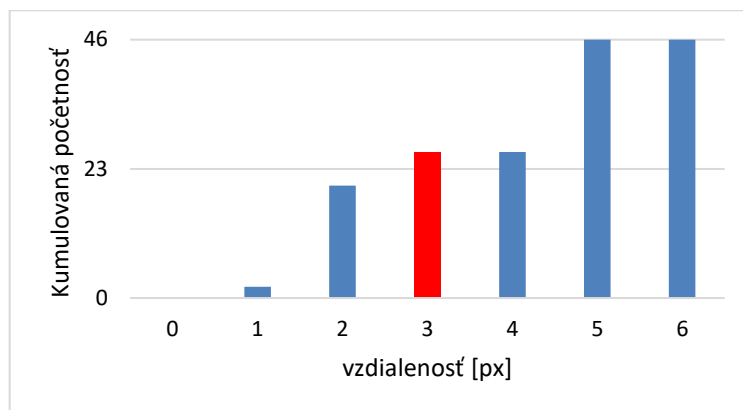
Graf 2 Príklad metódy Hausdorff

$$D = n_h \quad (27)$$

1.4.5 Hausdorff_{0,5}

Táto metóda vychádza z predošlej metódy, ale pre stanovenia parametru D v nej nie je použitý posledný bin, ale bin, ktorému prináleží prvok s kvantilom 0,5. Takýto prvok je možné nájsť vytvorením kumulatívneho histogramu. Index binu, ktorý v kumulovanom histograme nadobudne alebo prekročí svojou početnosťou

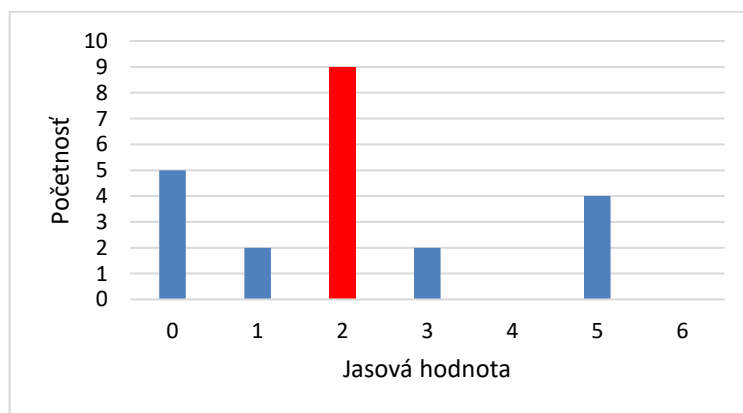
hodnotu $\frac{n_h}{2}$, je nastavený ako parameter D . Takýto prístup zrobustňuje predošlú metódu v prípadoch, kedy nie sú zo scény správne extrahované všetky hrany.



Graf 3 Kumulovaný histogram použítý v metóde Hausdorff_{0,5}

1.4.6 Početnosť najvyššieho bin

Vzdialenosť D je v tejto metóde udaná ako prevrátená hodnota početnosti binu s najväčším počtom prvkov.



Graf 4 Príklad metódy najpočetnejšieho binu

$$D = \frac{1}{\max(h_i)} \mid i = 0, \dots, n_h \quad (28)$$

2 VYUŽITÉ VÝVOJOVÉ NÁSTROJE

V tejto kapitole sú popísané hlavné vývojové nástroje použité pri riešení tejto práce a dôvody ich výberu.

2.1 Visual Studio

Visual Studio je vývojový nástroj určený pre tvorbu a vývoj programov, konzolových aplikácií, webových stránok, webových aplikácií či mobilných aplikácií. Je vyvíjané spoločnosťou Microsoft od roku 1997. Výhodou *Visual Studia* je široké spektrum možností a veľká užívateľská komunita. *Visual Studio* podporuje programovanie v jazykoch ako C, C++, C++/CLI, VB, C#, JavaScript, Phyton, Ruby či HTML. Jediný podporovaný operačný systém je Windows.

Najzákladnejšia verzia *Visual Studia*, Community edition, je dostupná zdarma. Microsoft taktiež v rámci programu *MSDN Academic Alliance* ponúka študentom škôl voľné licencie na tento softvér. [15]

Pri riešení tejto diplomovej práce bolo ako *IDE* použitá verzia *VS 2017*, ktorá v čase riešenia práce, na rozdiel od verzie 2019, umožňovala doinštalovanie rozšírenia *Image Watch*. Ako programovací jazyk bolo použité C++.

2.1.1 Image Watch

Vývojové prostredie *VS* taktiež ponúka širokú ponuku rozšírení, ktoré je možné si doinštalovať. Jedným z rozšírení použitých pri vypracovaní tejto diplomovej práce bolo *Image Watch 2017*, ktoré ponúka vizualizovanie všetkých snímok uložených v pamäti programu a je kompatibilné s knižnicou *OpenCV*. Vďaka tomuto rozšíreniu je možné v režime *Debug* rýchlo skontrolovať správnosť spracovania snímky, prípadne kontrolovať jasové hodnoty jednotlivých pixelov.

2.2 OpenCV

Táto voľne dostupná knižnica je určená pre počítačové videnie, spracovanie obrazu a strojové učenie. Vďaka voľnej dostupnosti a viac ako 2500 optimalizovaným algoritmom patrí medzi najrozšírenejšie knižnice svojho druhu s viac ako 14 miliónmi stiahnutí. Podporuje programovacie jazyky ako C++, Phyton, Java či MATLAB a operačné systémy Windows, Linux, Android a Mac OS. V súčasnosti sú dostupné tri stabilné verzie tejto knižnice v2, v3 a v4. [1]

Pri riešení tejto diplomovej práce bola použitá verzia knižnice 4.0.1. V diplomovej práci sú popisované funkcie tejto knižnice, a preto sa v celej práci predpokladá použitie menného priestoru *cv*.

2.3 OpenGL

Táto grafická knižnica slúži na prácu s 2D a 3D grafikou. Má množstvo funkcií pre priestorové výpočty a vysoký výkon vďaka využitiu grafického procesora.

V tejto práci je knižnica *OpenGL* využitá pri vytváraní vzorov objektu, ktoré sú použité v detekčnom algoritme. Vďaka tejto knižnici sú vzory presné a rýchlo vyrendrované.

2.4 Qt

Qt je multiplatformová knižnica určená na vývoj a tvorbu *GUI* aplikácií. Vývoj na tejto knižnici začal v roku 1990 v Nórsku. Knižnica podporuje všetky známe operačné systémy a je písaná v jazyku C++. Táto knižnica ponúka bezplatnú licenciu pre tvorbu neziskových programov a má veľkú užívateľskú komunitu. [4]

Knižnica *Qt* vo verzii 5.11.1 bola v tejto práci použitá na vytvorenie *GUI*, ktoré urýchlilo vývoj a testovanie aplikácie. V tejto knižnici je taktiež vytvorený program *Object Position*, ktorý je výstupom diplomovej práce.

2.5 MVTec HALCON

Knižnica *MVTec HALCON*, alebo iba *Halcon*, je knižnica určená na počítačové videnie a strojové učenie. Je celosvetovo známa vo svojom odbore a vyniká algoritmami počítačového videnia všetkých druhov. Jej hlavné zameranie je priemysel, kde je použitá v státisícoch zariadení na analýzu tvaru, morfologické operácie, porovnávanie, meranie a identifikáciu. Knižnica ponúka tie najnovšie a najlepšie technológie v oblasti počítačového videnia, napríklad spracovanie 3D obrazu či algoritmy hlbokého strojového učenia. Pre zvýšenie výkonu využíva viacjadrové spracovanie a taktiež *GPU* akceleráciu.

Knižnica podporuje širokú škálu operačných systémov a prácu so stovkami priemyselných kamier. Výhodou tejto knižnice je taktiež napríklad podpora embedded zariadení.

Táto knižnica taktiež obsahuje *IDE* s názvom *HDevelop*, ktoré uľahčuje vytváranie aplikácie a obsahuje množstvo vzorových programov na rôzne priemyselné aplikácie. Mimo tohto vývojového prostredia knižnica podporuje programovanie v jazykoch C, C++, C# a Visual Basic .NET. V tejto práci bola použitá verzia 18.11.

Dôvodom výberu tejto knižnice bola nemožnosť nájdania iného, plateného či open-source, riešenia, ktoré by dokázalo riešiť problém stanovenia pozície objektu použitím monokulárnej kamery. To, že táto knižnica patrí medzi svetovú špičku v algoritmoch pre spracovanie obrazu, potvrdzuje aj fakt, že články často porovnávajú svoje algoritmy s algoritmami tejto knižnice. Ďalej má oproti iným

priemyselným knižniciam veľkú podporu práce s 3D obrazom a modelmi, takže predstavuje vhodného kandidáta na porovnávanie s algoritmom použitým v tejto práci.

Táto knižnica bola v tejto práci použitá iba na porovnanie s dosiahnutými výsledkami tejto práce. Preto táto práca neobsahuje žiadne príklady použitia tejto knižnice. [24]

2.6 Calib.io

Táto webová stránka sa venuje kalibračným vzorom určeným pre kalibráciu vnútorných parametrov kamery. Je na nej možné zakúpiť už vyrobené, vysoko kvalitné kalibračné vzory rôznych druhov. Keďže pri kalibrácii sa predpokladá dokonalá rovina kalibračného vzoru, sú vzory zhotovené z hliníku/polyetylénu, ktoré ponúkajú výbornú rovinu, teplotnú stabilitu, odolnosť voči korózii a pevnosť.

Taktiež je možné na tejto stránke nájsť konfiguračný program pre kalibračný vzor, s ktorým je možné si vygenerovať kalibračný vzor rôznych rozmerov a druhov. Medzi podporované druhy patrí šachovnicový, kruhová mriežka, asymetrická kruhová mriežka, či ChArUco. Kalibračné vzory na tejto stránke sú kompatibilné s *OpenCV*, *MVTec HALCON* či *MATLAB*. [18]

3 KALIBRÁCIA KAMERY

V tejto kapitole je popísaný spôsob, akým boli v tejto práci zistené vnútorné a vonkajšie parametre kamery. Vnútorné parametre kamery boli zistené osobitne použitím knižnice *OpenCV* a *Halcon*, a to kvôli rozdielnym parametrom, ktoré tieto knižnice používajú. Vonkajšie parametre boli zistené iba za použitia knižnice *OpenCV*. V poslednej podkapitole je popísaná kontrola správnosti parametrov kamery, získaných pomocou knižnice *OpenCV*.

3.1 Kamera

V prvom kroku celej praktickej realizácie tejto práce bolo potrebné vhodne vybrať kameru, ktorá bude v celej práci využitá na stanovenie polohy objektu. Dôvod použitia práve jednej kamery v celej práci je ten, že vnútorné parametre kamery zistené pri kalibrácii sa vždy vzťahujú na kameru, s ktorou bola kalibrácia robená. Preto ich použitie na inej kamere rovnakého typu by mohlo spôsobiť nepresnosti v meraní.

Požiadavky na kameru boli minimálne rozlíšenie 1920×1080, frekvencia získavania snímok minimálne 30 FPS, nízka cena, kvalitný obraz, kompatibilita s knižnicou *OpenCV* a upevnenie na statív. Podľa týchto parametrov bola zvolená kamera *Logitech HD Pro Webcam C920*, ktorá disponuje USB rozhraním. Tento typ kamery je možné často vidieť pri prezentovaní rôznych aplikácií počítačového videnia, práve z dôvodu jej kvality, dostupnosti, podpory a nízkej ceny oproti priemyselným kamerám. Túto kameru je možné zakúpiť v cene nižšej ako 100 €, pričom ceny priemyselných kamier a optík začínajú približne na trojnásobku tejto ceny. [16]



Obr. 8 Logitech HD Pro Webcam C920

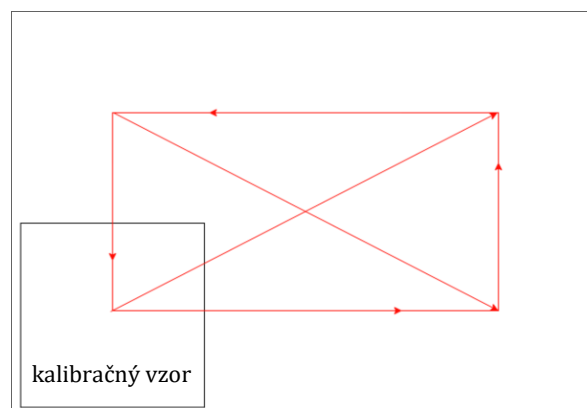
3.2 Vnútorne parametre kamery

V tejto podkapitole je vysvetlený spôsob stanovenia vnútorných parametrov kamery s knižnicou *OpenCV* a *Halcon*. V prvej časti tejto podkapitoly je obecné popísaný spôsob kalibrácie a zvolené metódy použité v tejto práci. Posledné dve časti tejto kapitoly konkrétne popisujú vnútornú kalibráciu kamery použitím knižníc *OpenCV* a *Halcon*.

3.2.1 Popis kalibračného pracoviska a spôsob kalibrácie

Pre praktickú realizáciu kalibrácie bolo zhotovené pracovisko pozostávajúce z bieleho stola a kamery zachytávajúcej plochu stola umiestnenej nad ním vo výške približne 45 cm. Biela farba stola bola zvolená kvôli zvýšeniu kontrastu a zamedzeniu ruchu v scéne. Zaostrenie kamery bolo nastavené do čo najväčšej vzdialenosti a v priebehu celej práce sa nemenilo. Keďže kameru pri kalibrácii nie je nutné nastavovať do konkrétnej polohy, výška kamery nad stolom bola navrhnutá tak, aby boli kalibračné značky vzoru pri všetkých očakávaných kalibračných polohách detekovateľné a ostré, aby tak nedochádzalo ku zníženiu presnosti kalibrácie vplyvom zlého zaostrenia. Kalibračná miestnosť bola osvetlená rovnomerne použitím neónových svietidiel.

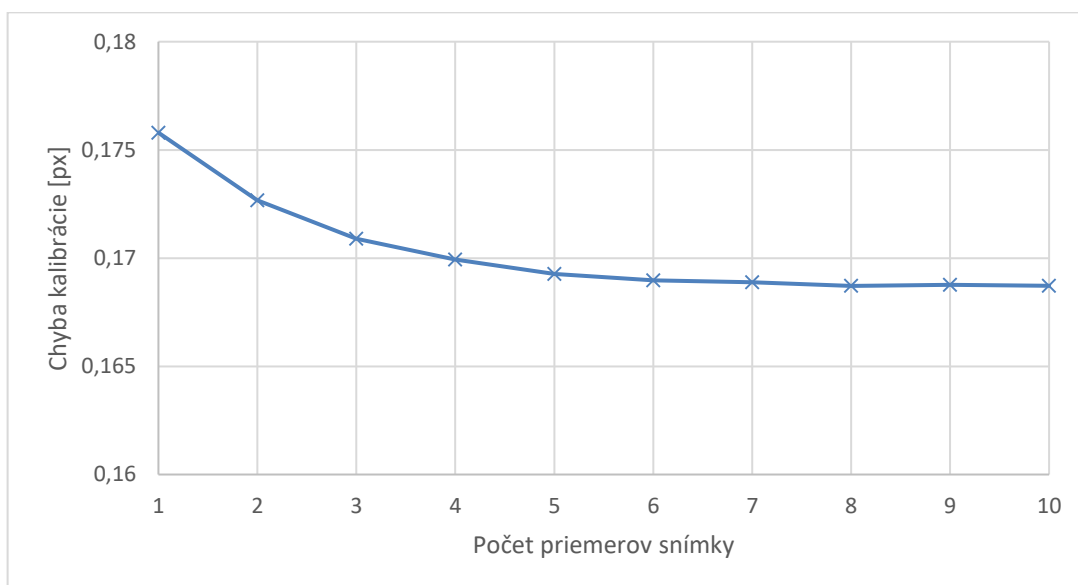
Spôsob kalibrácie bol inšpirovaný vzorovými kalibračnými fotkami v knižnici *Halcon*. V prvom kroku kalibrácie bol kalibračný vzor položený na stôl a posúvaný po krajoch a uhlopriečne cez snímku podľa Obr. 9. Tento pohyb bol dvakrát zopakovaný pri zmene výšky kalibračnej platne na 86 mm a 140 mm nad stolom. V ďalšom kroku bol kalibračný vzor naklonený na stranu objektmi o výške 25 mm a 86 mm a pri takomto náklone bol posúvaný po okrajoch snímky. Takýmto postupom bola pri každej kalibrácii vytvorená kalibračná sada pozostávajúca z minimálne 100 snímok.



Obr. 9 Pohyb kalibračného vzoru v scéne

3.2.2 Priemerovanie snímok

Pre získanie čo najlepších výsledkov kalibrácie je potrebné zabezpečiť čo najkvalitnejšie snímky, na ktorých je zreteľne viditeľný celý kalibračný vzor. Kvalita nájdenia kalibračného vzoru môže byť ovplyvnená šumom kamery. Pre zistenie vplyvu šumu použitej kamery na presnosť kalibrácie bolo vytvorených 10 kalibračných sád so 122 rôznymi polohami kalibračného vzoru. Ako kalibračný vzor bola použitá šachovnica s rozmermi 10×7 s veľkosťou štvorca 20,45 mm. V každej polohe kalibračného vzoru bolo vytvorených 10 snímok, ktoré boli postupne priemerované a následne uložené. Pomocou vytvorených kalibračných sád bola následne vypočítaná kalibrácia kamery, ktorej výsledky sú zobrazené v Graf 5.



Graf 5 Vývoj reprojekčnej chyby kalibrácie v závislosti od počtu priemerovaných snímok

Z Graf 5 vyplýva, že zvyšovaním množstva priemerovaných snímok dochádza ku redukcii šumu kamery a ku zvýšeniu presnosti kalibrácie. Presnosť pri 10 priemeroch snímky sa oproti snímkam bez priemeru zlepšila o približne 0,007 px. Táto zmena nie je až taká razantná, a preto bolo v tejto práci zvolené priemerovanie troch snímok, ktoré je použité v ďalších častiach kalibrácie kamery.

3.2.3 Materiál kalibračného vzoru

Väčšina známych kalibračných vzorov má plošný charakter – súradnica Z je pre všetky kalibračné body nulová. Z toho vyplýva predpoklad, že všetky body ležia na jednej, dokonale rovnej, ploche. Týmto predpokladom vzniká podmienka pre fyzický model kalibračného vzoru, a to tá, že musí byť vytvorený z dokonale

rovného materiálu. V tejto diplomovej práci boli otestované štyri rôzne materiály, z ktorých bol kalibračný vzor vytvorený, a pre každý materiál bola vytvorená kalibračná sada. Ako kalibračný vzor bol použitý šachovnicový vzor o rozmeroch 10×7.

3.2.3.1 Papier

Ako prvý kalibračný materiál bol použitý papier, na ktorý bol vytlačený kalibračný vzor. Pri potrebe nakloniť kalibračný vzor bol pod papier umiestnený kartón, čo je nevýhodou použitia tohto materiálu.

3.2.3.2 Kartón

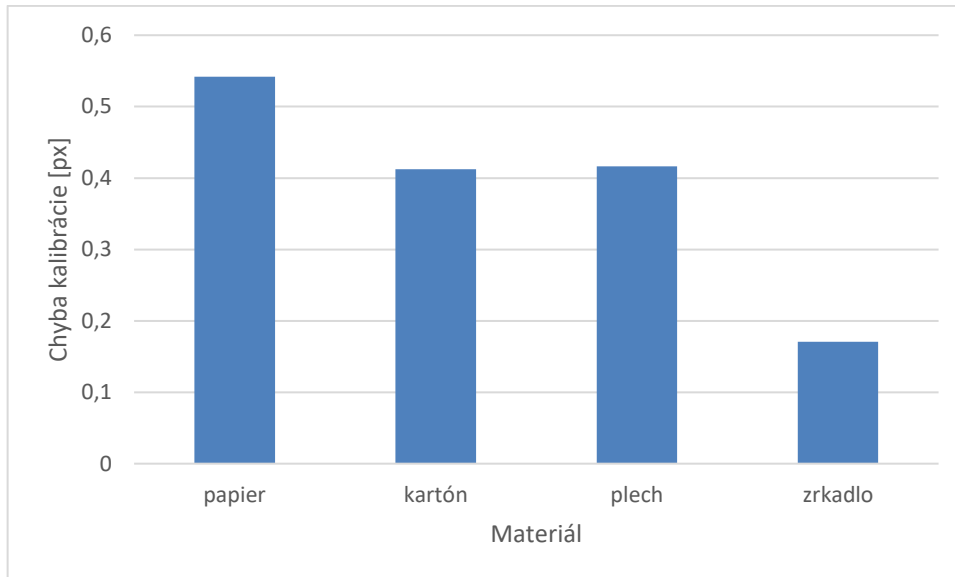
Ďalším použitým kalibračným materiálom bol kartón, na ktorý bol nalepený papier s kalibračným vzorom. Výhodou tohto materiálu je, že oproti papieru je pevný a drží svoj tvar aj bez využitia iného podporného materiálu.

3.2.3.3 Plech

Tretí použitý materiál bol plech s hrúbkou 0,5 mm, na ktorom bol kalibračný vzor vytlačený. Jeho výhodou je vyššia rovnosť oproti kartónu, ktorý býva zvyčajne prehnutý v jednej osi.

3.2.3.4 Zrkadlo

Ako posledný materiál bolo použité zrkadlo s hrúbkou 4 mm. Keďže pri zrkadlách je v odraze požadované dokonalé zachovanie tvaru, musí v nich byť zaistená dokonalá rovnosť. Preto majú zrkadlá zvyčajne rovnejší povrch na rozdiel od obyčajného skla, pri ktorom táto vlastnosť nie je požadovaná. Súčasne sa zrkadlá vyznačujú vyššou tuhosťou oproti plechu, a preto pri nich nedochádza ku žiadnemu prehýbaniu vplyvom podkladania či držania. Táto vlastnosť zrkadla je využívaná napríklad pri 3D tlači, kde je zrkadlo vďaka takmer dokonalkej rovnosti často používané ako základová plocha.



Graf 6 Porovnanie reprojekčnej chyby kalibrácie v závislosti od použitého kalibračného materiálu

Z výsledkov zobrazených v Graf 6 je zrejmé, že najmenšia reprojekčná chyba kalibrácie vznikla pri použití zrkadla ako základu kalibračného vzoru. Týmto meraním bolo potvrdené, že zrkadlo má takmer dokonale rovnú plochu a nepodlieha ohýbaniu tak ako zvyšné materiály. Z tohto dôvodu je zrkadlo vo zvyšných častiach tejto práce použité ako kalibračný materiál. Najhoršie zo všetkých materiálov skončil papier s vytlačeným kalibračným vzorom. Kartón a plech dosiahli približne rovnakú chybu. Nevýhodou plechu je, že je lesklý, čo vnášalo do merania chybu a pri podložení sa mierne prehýbal.

Pri kalibrácii je možné držať sa pravidla, ktoré hovorí, že ak je výstupná chyba kalibrácie menšia ako jeden pixel, kalibráciu možno považovať za úspešnú. [17]

3.2.4 OpenCV

Výpočet parametrov kamery v knižnici *OpenCV* sa vykonáva použitím funkcie *calibrateCamera*. Výstupom tejto funkcie je matica kamery K a koeficienty skreslenia dC . Príklad použitia tejto funkcie je vo Výpis 2.

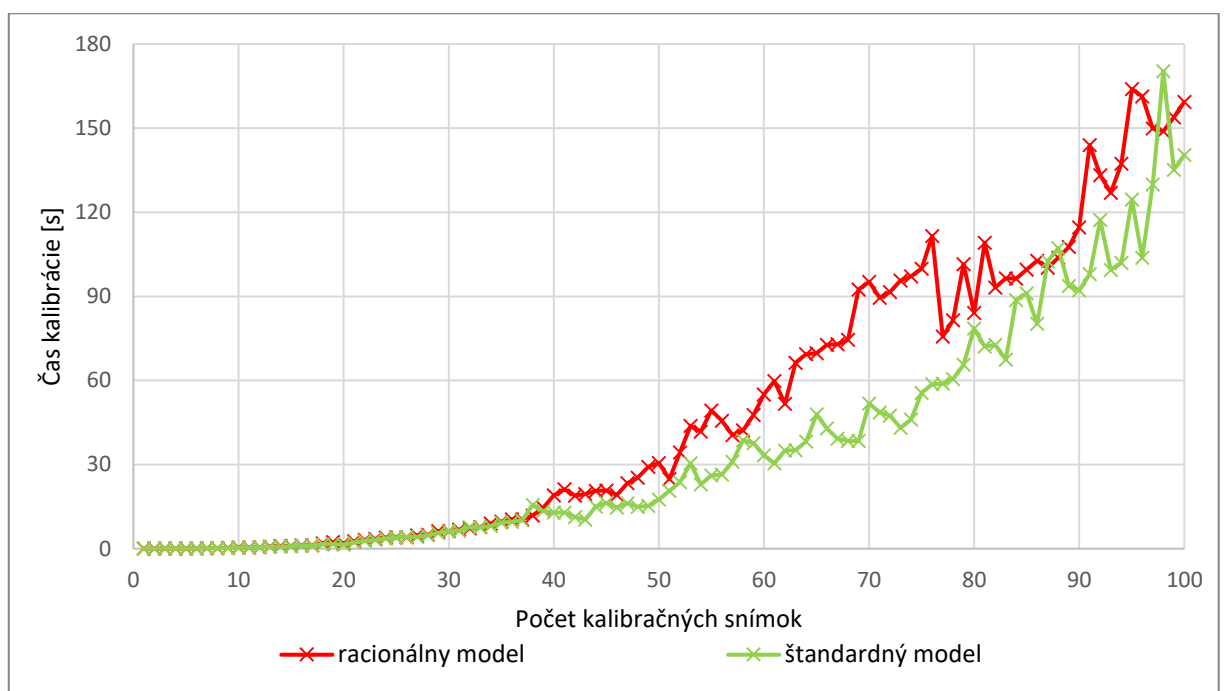
Výpis 2 Použitie funkcie *calibrateCamera*

```
calibrateCamera(worldPoints, //vstupné pole bodov objektu
               imagePoints, //vstupné pole bodov obrazu
               imageSize, //rozlíšenie kamery
               cameraMatrix, //výstupná matica kamery
               distCoeffs, //výstupné koeficienty skreslenia
               rvecs, //výstupné rotačné vektory kamery
               tvecs); //výstupné translačné vektory kamery
```

3.2.4.1 Výber modelu kamery

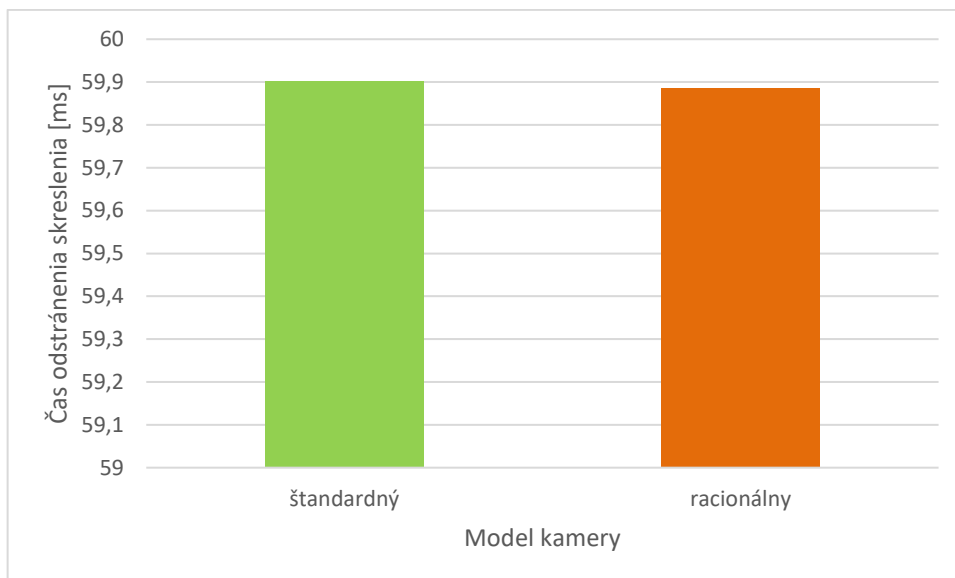
Počet koeficientov skreslenia závisí od použitého modelu kamery. Knižnica *OpenCV* rozlišuje dva druhy modelov kamery a to štandardný a racionálny. Pri kalibrácii kamery použitím štandardného modelu pre skreslenie vracia kalibračná funkcia päť koeficientov skreslenia. Nastavením flagu `CV_CALIB_RATIONAL_MODEL` je pri výpočte koeficientov skreslenia použitý racionálny model, ktorý používa osem koeficientov skreslenia a je určený prednostne pre kamery so širokým uhlom záberu. [19]

Pre porovnanie bola vykonaná testovacia kalibrácia použitím oboch modelov. Pre správne porovnanie boli pri kalibrácii použité rovnaké kalibračné snímky. Časová náročnosť kalibrácie pri použití jednotlivých modelov je zobrazená v Graf 7.



Graf 7 Porovnanie časovej náročnosti kalibrácie kamery pri použití rôznych modelov kamery

Pre odstránenie skreslenia je v knižnici *OpenCV* funkcia `undistort`, ktorá vo vstupnej snímke za pomoci predložených koeficientov skreslenia odstráni skreslenie šošovky. Priemerné časy odstránenia skreslenia šošovky kamery pri použití štandardného a racionálneho modelu sú zobrazené v Graf 8. Priemerný čas bol určený zo spracovania 1000 snímok o rozlíšení 1920×1080. Obsah snímok nijakým spôsobom neovplyvňoval čas spracovania.



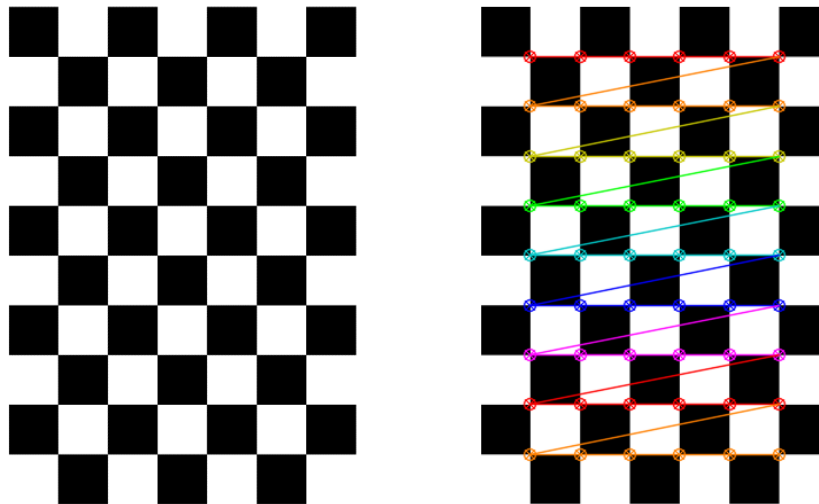
Graf 8 Porovnanie rýchlosti odstránenia skreslenia kamery pri použití rôznych modelov kamery (vyššie je horšie)

Pri porovnávaní štandardného a racionálneho modelu kamery bolo zistené, že pre štandardný model je časová náročnosť kalibrácie podľa Graf 7 nižšia oproti racionálnemu modelu a rýchlosť odstránenia skreslenia, zobrazená v Graf 8, je takmer rovnaká. V tejto diplomovej práci nebola použitá kamera so širokým uhlom záberu, a preto je v ďalších častiach tejto práce pri kalibrácii použitý štandardný model kamery.

3.2.4.2 Podporované kalibračné vzory

Knižnica *OpenCV* pre kalibráciu vnútorných parametrov kamery podporuje tri základné kalibračné vzory: šachovnica, kruhová mriežka a asymetrická kruhová mriežka. Každý kalibračný vzor je možné popísať pomocou priestorových pozícií jeho kalibračných bodov. Takéto pole kalibračných bodov je nazývané *objectPoints*. Keďže všetky popísané kalibračné vzory sú plošné, pri definovaní priestorových pozícií kalibračných bodov sa súradnica *Z* vždy nastavuje na hodnotu 0.

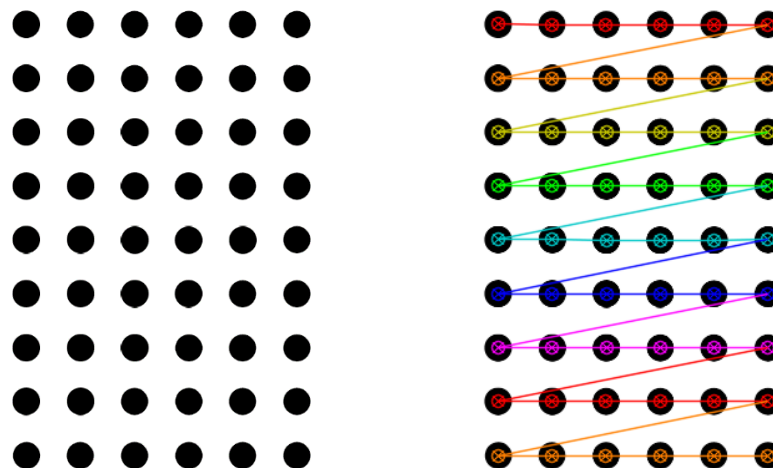
Kalibračné vzory s rozmermi 9×6, podporované knižnicou *OpenCV*, a ich detekcie sú zobrazené na Obr. 10, Obr. 11 a Obr. 12 a spôsoby ich detekcie vo Výpis 3, Výpis 4 a Výpis 5.



Obr. 10 Kalibračný vzor – šachovnica

Výpis 3 Príklad detekcie vzoru – šachovnica

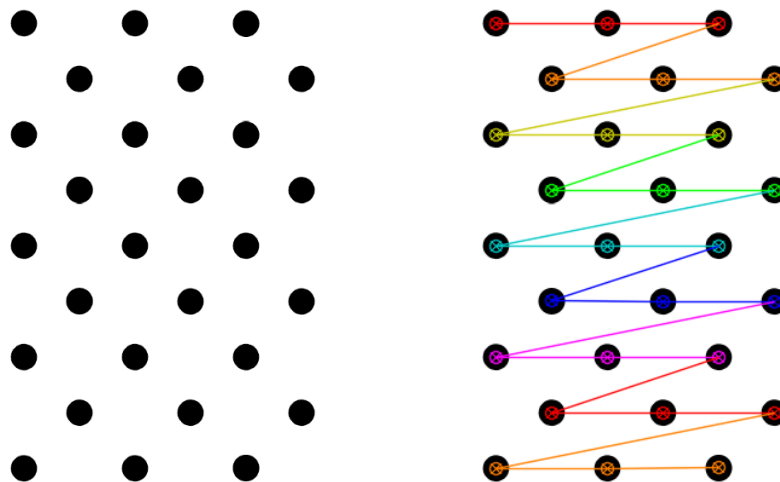
```
findChessboardCorners(img, //vstupný obraz
    patternSize, //vstupná veľkosť kalibračného vzoru
    imagePoints); //výstupné pole bodov obrazu
```



Obr. 11 Kalibračný vzor – kruhová mriežka

Výpis 4 Príklad detekcie vzoru – kruhová mriežka

```
findCirclesGrid(img, //vstupný obraz
    patternSize, //vstupná veľkosť kalibračného vzoru
    imagePoints); //výstupné pole bodov obrazu
```

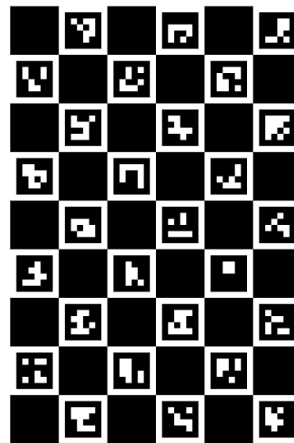



Obr. 12 Kalibračný vzor – asymetrická kruhová mriežka

Výpis 5 Príklad detekcie vzoru – asymetrická kruhová mriežka

```
findCirclesGrid(img, //vstupný obraz
  patternSize, //vstupná veľkosť kalibračného vzoru
  imagePoints, //výstupné pole bodov obrazu
  CALIB_CB_ASYMMETRIC_GRID); //flag pre detekciu asymetrického kruhového vzoru
```

Ďalší kalibračný vzor, ktorý je podporovaný knižnicou *OpenCV* je *ChArUco*. Tento vzor však patrí pod rozšírené moduly knižnice a v tejto práci nie je pri kalibrácii kamery použitý.



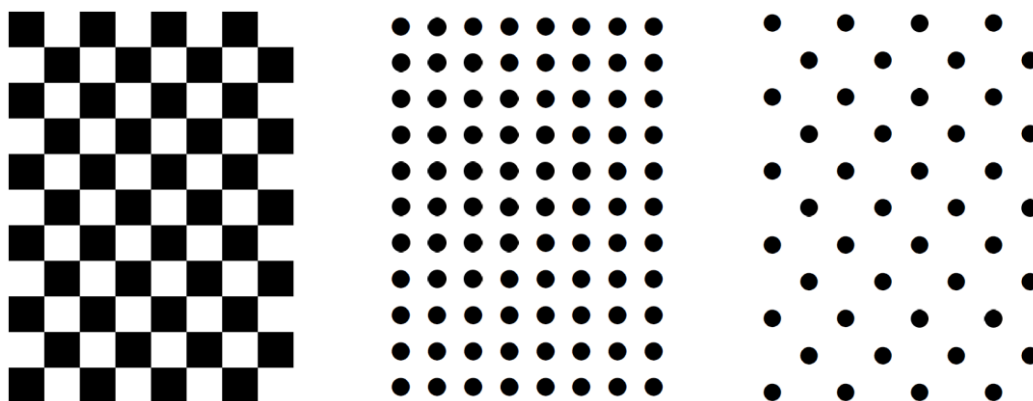
Obr. 13 Kalibračný vzor - ChArUco

3.2.4.3 Závislosť presnosti a rýchlosti kalibrácie od použitého vzoru

V tomto odseku je popísané porovnanie troch kalibračných vzorov podporovaných knižnicou *OpenCV*. Aby sa zaistili približne rovnaké podmienky pre všetky kalibračné vzory, boli ich rozmery navrhnuté tak, aby mali všetky kalibračné platne rovnakú plochu, a to 180×240 mm. Parametre kalibračných vzorov použitých pre porovnanie sú uvedené v Tabuľka 2 a zobrazené na Obr. 14.

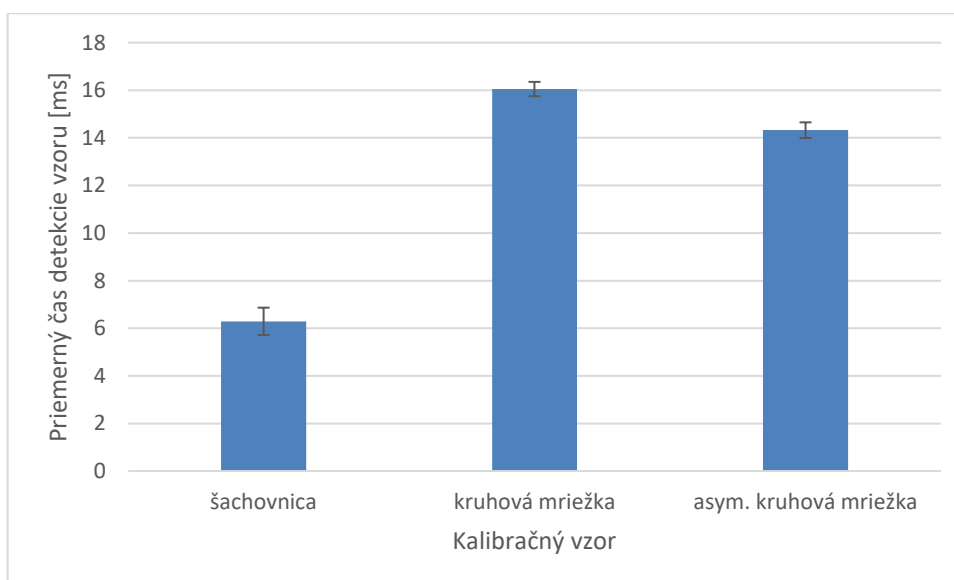
Tabuľka 2 Parametre kalibračných vzorov

Typ kalibračného vzoru	Počet stĺpcov	Počet riadkov	Počet kalibračných bodov	Vzdialenosť kalibračných bodov [mm]
šachovnica	10	7	70	20
kruhová mriežka	11	8	88	20
asym. kruhová mriežka	11	8	44	40

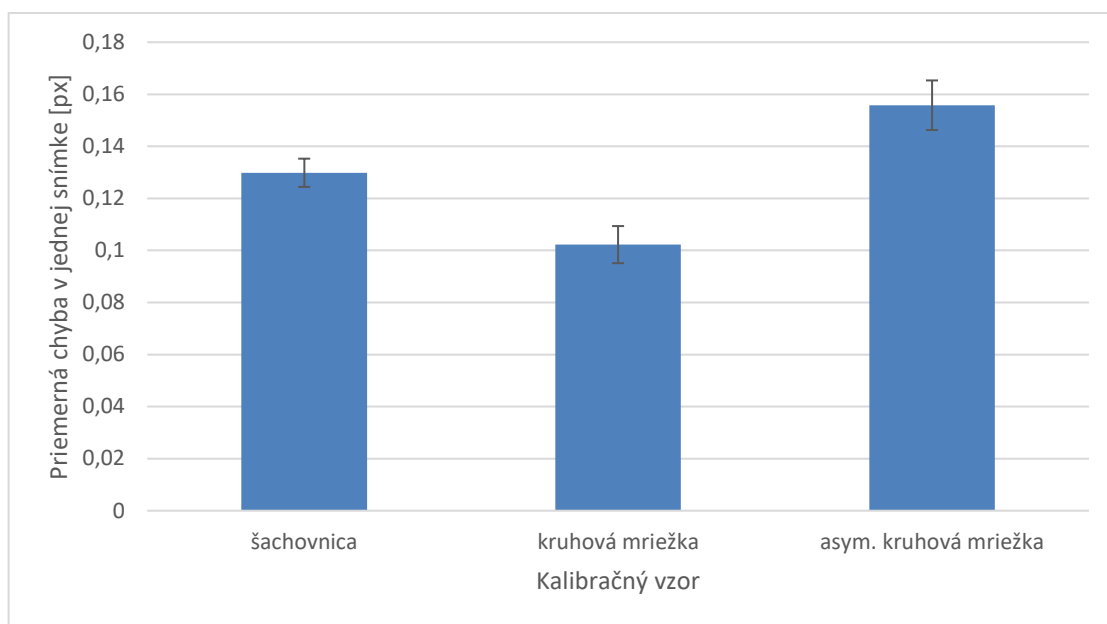


Obr. 14 Kalibračné vzory použité pre porovnanie presnosti kalibrácie

Pre porovnanie kvality kalibrácie kamery pri použití rôznych vzorov boli vytvorené tri kalibračné sady. V prvom kroku bola na jednotlivých snímkach zistená rýchlosť detekcie vzoru a chyba vzniknutá na jednotlivých snímkach, ktoré sú zobrazené v Graf 9 a Graf 10.

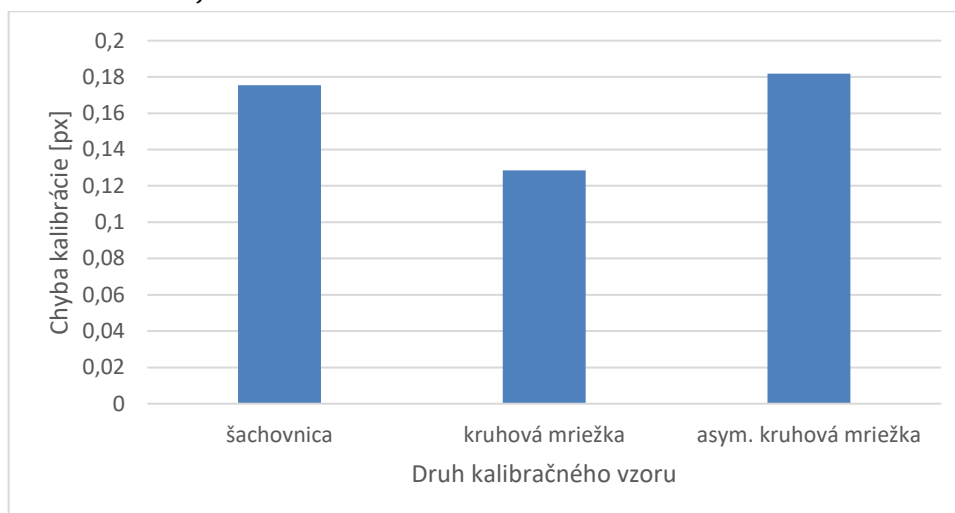


Graf 9 Priemerný čas detekcie vzoru v snímke



Graf 10 Priemerná reprojekčná chyba detekcie vzoru v jednej snímke

Následne bola pomocou všetkých snímok kamera skalibrovaná. Výsledná presnosť kalibrácie je zobrazená v Graf 11.



Graf 11 Výsledná reprojekčná chyba kalibrácie kamery

Vo výsledných grafoch je vidieť, že šachovnicový vzor je zo všetkých vzorov najľahšie a najrýchlejšie detekovateľný, no jeho nevýhodou je vyššia reprojekčná chyba výslednej kalibrácie.

Kruhový vzor mal spomedzi všetkých použitých kalibračných vzorov najvyšší detekčný čas, ale s jeho použitím bola dosiahnutá najnižšia chyba výslednej kalibrácie. Najvyšší detekčný čas a presnosť kalibrácie boli pravdepodobne spôsobené najvyšším počtom kalibračných bodov zo všetkých použitých vzorov.

Asymetrický kruhový vzor dosiahol najvyššiu chybu pri kalibrácii a jeho priemerný detekčný čas, bol napriek najnižšiemu počtu kalibračných vzorov takmer rovnaký s kruhovým vzorom. Pri tomto vzore taktiež nastávala neschopnosť detekcie vzoru v scéne pri väčšom naklonení kalibračnej platne, čo obmedzilo rozmanitosť jeho kalibračnej sady.

Z výsledkov bolo usúdené, že napriek pomalšiemu detekčnému času je pre presnosť kalibrácie kamery najvýhodnejšie použitie kruhového vzoru. V tejto práci boli ako referenčné parametre kamery vybrané parametre, pri ktorých mala výsledná chyba kalibrácie najnižšiu hodnotu, a to pri použití vzoru kruhovej mriežky prilepenej na zrkadle. Výsledná reprojekčná chyba tejto kalibrácie bola 0,129 px a zistené parametre boli nasledovné:

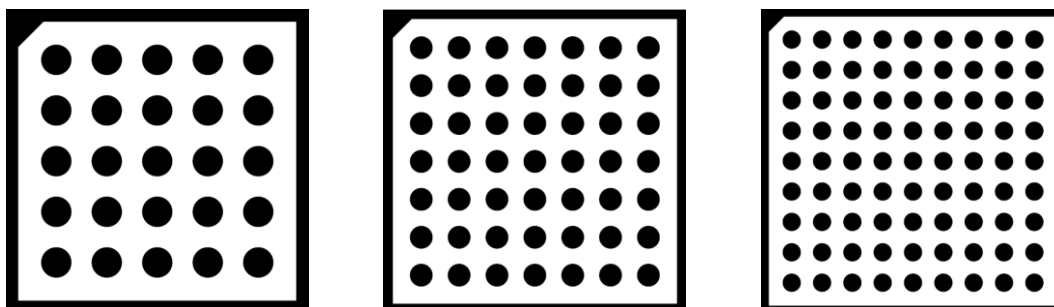
$$K = \begin{pmatrix} 1442,38 & 0 & 942,25 \\ 0 & 1441,32 & 539,12 \\ 0 & 0 & 1 \end{pmatrix}$$

$$Dist_{coefficientis} = (0,047627 \quad -0,186555 \quad -0,000313 \quad -0,000316 \quad 0,143417)$$

3.2.5 Halcon

3.2.5.1 Podporované kalibračné vzory

Pre kalibráciu kamery použitím knižnice *Halcon* disponuje táto knižnica možnosťou vygenerovania vlastného kalibračného vzoru. Je možné si navoliť požadované rozmery, vzdialenosti kalibračných bodov a ich veľkosť. Táto knižnica využíva pri kalibrácii iba vzor kruhovej mriežky, takže nebolo nutné porovnávať presnosť kalibrácie pri použití rôznych druhov vzorov. Pre porovnanie presnosti však boli vygenerované kalibračné vzory s tromi rôznymi rozmermi a to 5×5, 7×7 a 9×9, zobrazené v Obr. 15. Výstupom generátora je súbor typu DESCR, ktorý popisuje pozície kalibračných bodov a je využitý pri výpočte kalibrácie kamery. Druhý súbor je typu PS a je ním možné vygenerovať súbor PDF určený na vytlačenie kalibračného vzoru.



Obr. 15 Použité vzory pre kalibráciu pomocou knižnice Halcon

3.2.5.2 Kalibrácia kamery

Knižnica *Halcon* má pre kalibráciu kamery možnosť použitia kalibračného asistenta, ktorý je umiestnený vo vývojovom prostredí *HDevelop*. V ňom je potrebné nastaviť približnú veľkosť pixelu na snímači kamery S a ohniskovú vzdialenosť F . Ďalším krokom je načítanie parametrov kalibračného vzoru uloženého v súbore typu DESCR, ktorý bol vytvorený pri generovaní vzoru. Posledným krokom je načítanie kalibračných snímok. Výsledkom kalibrácie sú upravené rozmery pixelu na snímači kamery, upravená ohnisková vzdialenosť kamery, parameter Kappa popisujúci radiálne skreslenie kamery a optický stred kamery.

Pri kalibrácii kamery bola ohnisková vzdialenosť nastavená na hodnotu 3,67 mm, ktorá je udávaná výrobcom na oficiálnej stránke kamery. Odhad veľkosti pixelu bol vypočítaný z referenčných parametrov kamery, zistených v predošlej podkapitole pomocou knižnice *OpenCV*. Výsledky kalibrácie kamery sú vypísané v Tabuľka 3. [16]

Tabuľka 3 Výsledky kalibrácie získané knižnicou *Halcon*

Kalibračný vzor	Chyba kalibrácie [px]	S_x [μm]	S_y [μm]	F [mm]	Kappa [m^{-2}]	c_x [px]	c_y [px]
5×5	0,2399	2,54077	2,546	3,709	-2219,44	936,746	523,679
7×7	0,2965	2,54117	2,546	3,67559	-1950,48	942,563	541,569
9×9	0,30115	2,54297	2,546	3,70576	-2534,22	944,204	521,373

Z výsledkov je vidieť, že najnižšia chyba kalibrácie bola dosiahnutá pri použití vzoru s rozmermi 5×5. Pri použití týchto parametrov však bola pri detekcii objektov zistená veľká chyba stanovenia pózy objektu. Táto chyba bola pravdepodobne spôsobená veľkou zmenou hodnoty ohniskovej vzdialenosti F , ktorú je možné si všimnúť aj vo výsledkoch vzoru s rozmermi 9×9. Z tohto dôvodu bola ako referenčná kalibrácia kamery v tejto práci použitá kalibrácia získaná použitím vzoru s rozmermi 7×7, prostredníctvom ktorej bola detekcia objektu najpresnejšia.

Výsledná hodnota ohniskovej vzdialenosti F bola pri tejto kalibrácii iba mierne upravená od oficiálnej hodnoty, ktorú udáva výrobca. Taktiež sa v nej pozícia optického stredu c najviac zhodovala s pozíciou optického stredu referenčnej kalibrácie kamery získanej v predošlej podkapitole.

3.3 Vonkajšie parametre kamery

Po zistení vnútorných parametrov kamery bolo možné stanoviť jej vonkajšie parametre. V tejto práci bolo potrebné stanoviť pozíciu kamery voči stolu, na ktorom bol objekt položený. Preto bol pre riešenie stanovenia pozície kamery vytvorený kalibračný vzor stola s presne rozmiestnenými značkami, ktoré

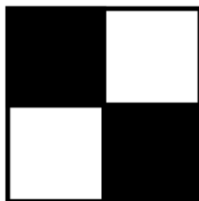
predstavovali kalibračné body. Stanovenie vonkajších parametrov kamery je v tejto práci riešené iba použitím knižnice *OpenCV*.

3.3.1 Popis kalibračného pracoviska

Pri stanovení vonkajších parametrov kamery bola kamera premiestnená do výšky približne 930 cm nad plochu stola a orientovaná tak, aby jej súradnicová os Z smerovala kolmo na rovinu stola. Pre rovnomernosť osvetlenia boli použité neónové svietidlá.

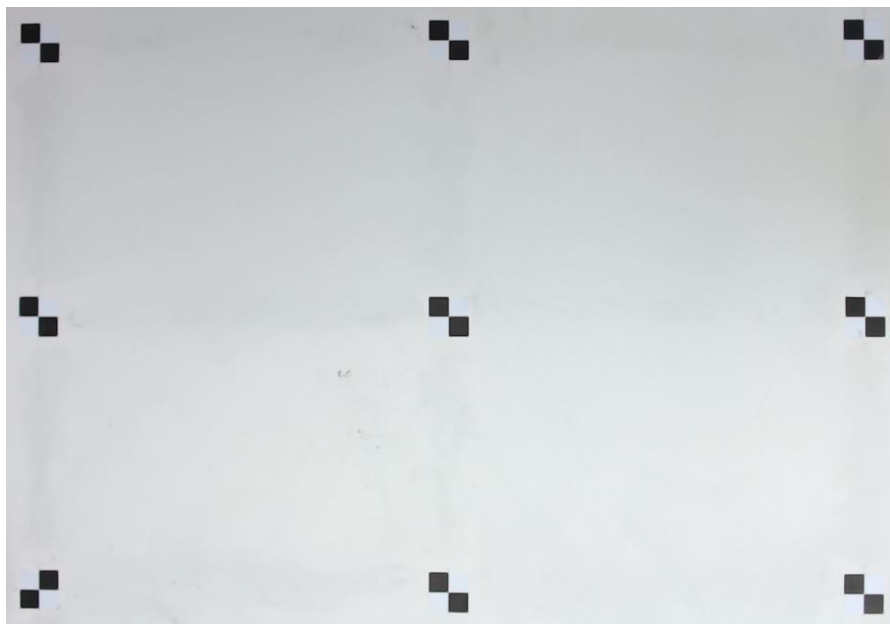
3.3.2 Návrh kalibračného vzoru

Ako značka bol zvolený štvorcový vzor zobrazený na Obr. 16, pozostávajúci zo 4 menších štvorcových plôch bielej a čiernej farby, pričom dve protíahlé plochy sú biele a dve čierne. Výhodou tohto vzoru je jednoduchá a presná detekovateľnosť vďaka vysokému kontrastu a schopnosť presného umiestnenia značky na požadované miesto vďaka jej tvaru.



Obr. 16 Kalibračný vzor použitý na stanovenie pózy kamery

Tieto značky boli následne umiestnené na požadované pozície na kalibračnom prípravku tak, aby tvorili mriežku. Pozície značiek boli stanovené tak, aby boli značky rozmiestnené čo najrovnomernejšie po celej snímke a pri danej výške kamery boli všetky zreteľne viditeľné. Ďalšou podmienkou boli rovnaké vzájomné vzdialenosti v X a Y osiach. Materiál, z ktorého bol vytvorený kalibračný prípravok, bol laminát s rozmermi 1×1 m. Ten mal na rozdiel od iných testovaných materiálov najmenší prehyb a tým vnášal do merania najmenšiu chybu. Snímka kalibračného prípravku je zobrazená na Obr. 17.



Obr. 17 Snímka kalibračného vzoru stolu

Na Obr. 17 je vidieť, že jedna zo značiek je na rozdiel od zvyšných otočená o 90° . Táto značka je označená ako bázová a vďaka nej je možné pri akomkoľvek otočení kamery správne stanoviť orientáciu stola a určiť pózu kamery. Táto značka, napriek označeniu bázová, nemusí nutne predstavovať počiatok svetovej súradnicovej sústavy. Jej charakter je čisto orientačný a počiatok súradnicového systému sa môže nachádzať v inom mieste stola.

3.3.3 Detekcia kalibračného vzoru

Pre detekciu kalibračných značiek bolo potrebné navrhnuť metódu, ktorá správne deteguje kalibračné značky. Bolo možné využiť tvar a vysoký kontrast použitých značiek.

Ako prvá voľba sa naskytl detektor rohov, ktorý je realizovaný funkciou `cornerHarris`. Pre správnu detekciu však musela byť scéna prázdna, pretože detektor bol náchylný na detekciu nežiaducich rohov.

Ďalším návrhom bolo prahovanie scény tak, aby sa vďaka vysokému kontrastu značiek dali nájsť požadované pozície značiek. Táto metóda bola oproti prvej metóde oveľa robustnejšia, ale veľmi závislá od rovnomernosti osvetlenia scény a náchylná na odlesky stola.

Ako posledná metóda bola testovaná konvolučná metóda. V tejto metóde je celá snímka prevedená do odtieňov sivej. Na takúto snímku je aplikovaná konvolúcia použitím jadra rovnakého tvaru ako detegovaná značka. Tento typ jadra je možné nájsť ako jeden z Haarovných príznakov. Veľkosť jadra bola nastavená na hodnotu

približne rovnú veľkosti značky v pixeloch. Jadro s veľkosťou 4 je zobrazené na Obr. 18.

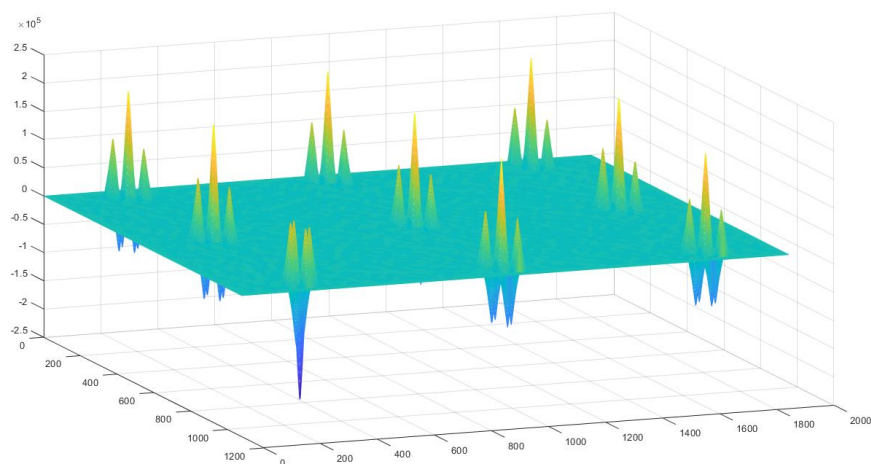
-1	-1	1	1
-1	-1	1	1
1	1	-1	-1
1	1	-1	-1

Obr. 18 Konvolučné jadro o rozmeroch 4×4

Na výpočet konvolúcie je použitá funkcia `filter2d`, ktorej výstupom je matica rovnakých rozmerov ako vstupná snímka. V tejto matici zobrazenej na Obr. 19 a Obr. 20 sú zreteľne viditeľné pozície jednotlivých značiek a taktiež rozdielna orientácia bázovej značky.



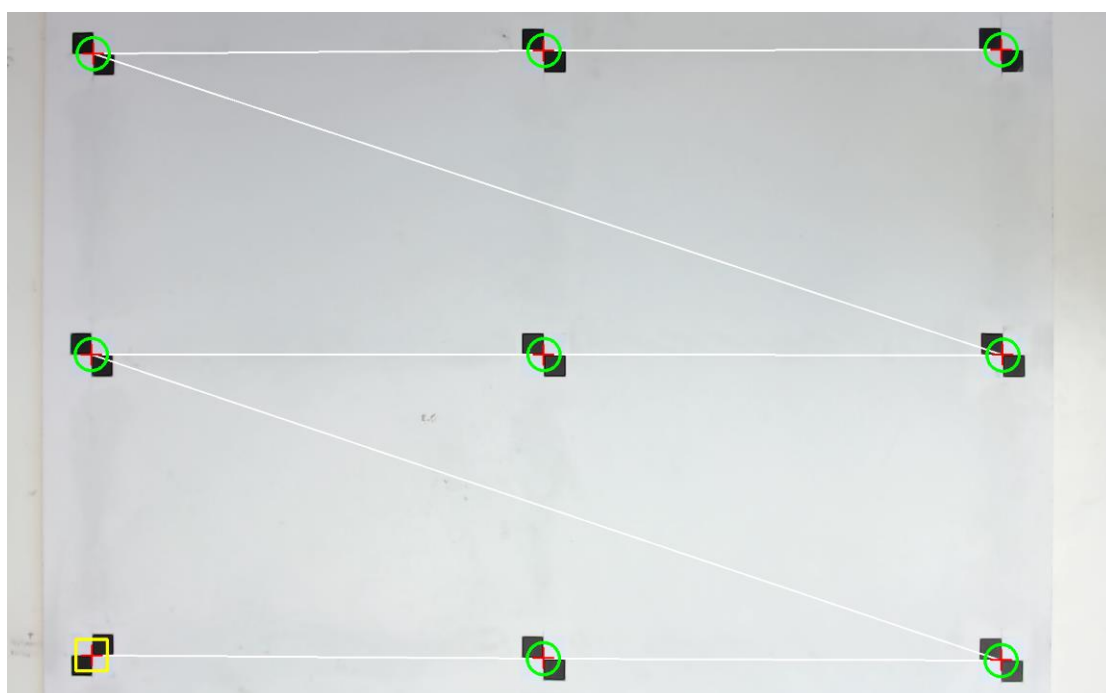
Obr. 19 Výsledok konvolúcie kalibračnej snímky stola s maskou



Obr. 20 3D zobrazenie konvolúcie kalibračnej snímky stola s maskou

Z tejto snímky sú zistené polohy bázevej značky a zvyšných značiek. Keďže orientácia bázevej značky a zvyšných značiek sa môžu otočením kamery alebo opačným nalepením značiek na stôl zmeniť, nie je možné sa spoliehať na to, že podľa Obr. 20 má bázeová značka vždy minimálnu hodnotu a zvyšné značky majú maximálne hodnoty. Z tohto dôvodu je v celej snímke nájdených n globálnych maxím a miním a je vypočítaná absolútna hodnota ich priemeru. Hodnota n predstavuje počet detegovaných značiek. Následne je množina nájdených bodov s nižšou hodnotou absolútneho priemeru priradená bázevej značke a druhá množina zvyšným značkám. Takéto jednoduché stanovenie značiek bolo otestované a preukázalo veľmi dobré výsledky detekcie.

Takýmto spôsobom je nájdená približná poloha všetkých značiek na stole. Ďalším krokom je zoradenie nájdených bodov a nájdenie ich presnej, subpixelovej polohy. Zoradenie prebieha nájdením krajných značiek a následne použitím parametrickej priamky. Pre subpixelové spresnenie polohy značky je použitá funkcia `cornerSubPix`.



Obr. 21 Výsledok detekcie kalibračného vzoru stola

Výsledkom detekcie sú zoradené, subpixelové polohy jednotlivých značiek stola v snímke. Algoritmus je odolný voči globálnej zmene jasu a ruchu v scéne. Pre ideálnu detekciu je požadované umiestniť kameru kolmo nad stôl.

Nevýhodou tohto spôsobu detekcie je neodolnosť voči natočeniu kamery o $\pm 45^\circ$ voči značkám na stole. Taktiež je tento algoritmus náchylný na nerovnomerné osvetlenie, kedy detekcia nemusí fungovať správne.

3.3.4 Stanovenie pózy kamery

V predošlej časti bol vysvetlený spôsob získania pozícií značiek stola v obraze, taktiež nazývaných ako body obrazu. Pre nájdanie pozície kamery voči stolu je potrebné poznať reálne polohy značiek v súradnicovom systéme, v ktorom chceme určiť vonkajšie parametre kamery. Následným spojením polôh značiek v obraze a reálnych polôh značiek kalibračného prípravku je možné použitím funkcie `solvePnP` stanoviť pózu kamery H_{CB} .

Výpis 6 Príklad stanovenia pózy kamery

```
solvePnP(worldPoints, //vstupné pole bodov objektu
         imagePoints, //vstupné pole bodov obrazu
         cameraMatrix, //vstupná matica kamery
         distCoeffs, //vstupné koeficienty skreslenia
         rvec, //výstupný rotačný vektor kamery
         tvec); //výstupný translačný vektor kamery
```

3.3.5 Stanovenie reprojekčnej chyby

Funkcia `solvePnP` na rozdiel od funkcie `calibrateCamera` vo svojej návratovej hodnote nevracia reprojekčnú chybu stanovenia vonkajších parametrov kamery. Pre výpočet chyby bola vypočítaná projekcia bodov objektu použitím funkcie `projectPoints`. Následne boli zistené vzdialenosti projekcií p a detegovaných bodov i a bola vypočítaná chyba Err_{RMS} . Stanovenie polohy stola s chybou nižšou ako 1 px bolo v tejto práci považované za správne.

$$dist_{Err} = \sqrt{(x_p - x_i)^2 + (y_p - y_i)^2} [px] \quad (29)$$

$$Err_{RMS} = \sqrt{\frac{1}{n} \left(\sum_{i=1}^n dist_{Err}^2 \right)} [px] \quad (30)$$

4 TVORBA REFERENČNEJ SADY SNÍMOK

Táto kapitola popisuje spôsob tvorby referenčných sád snímok, ktoré sú určené na stanovenie pozície objektu, a zisťovanie referenčných póz objektu v nich.

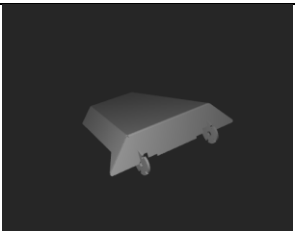
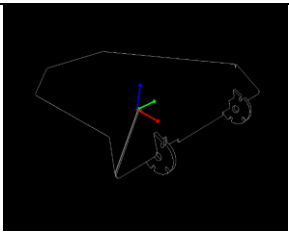
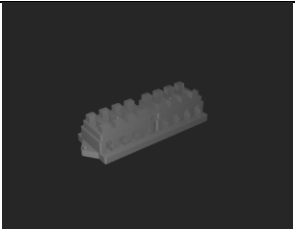
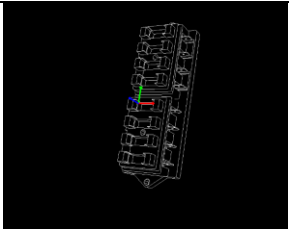
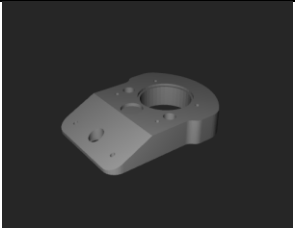
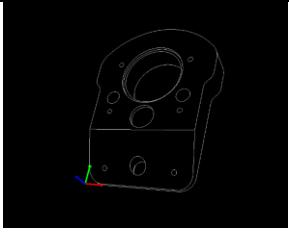
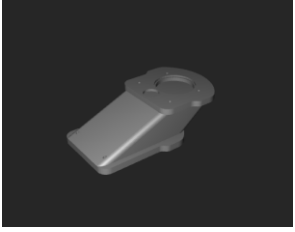
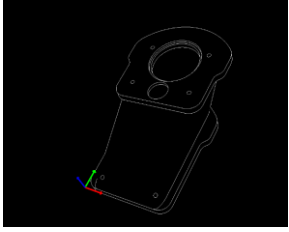
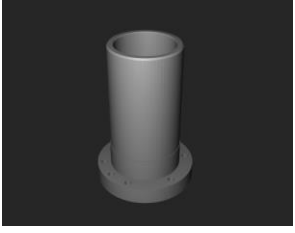
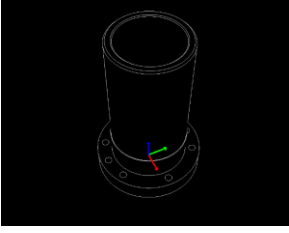
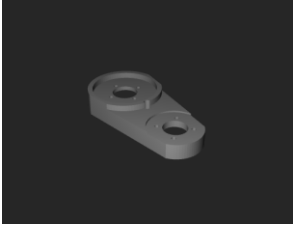
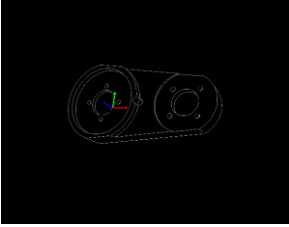
V prvej podkapitole sú popísané referenčné objekty použité pri tvorbe referenčných snímok. Druhá podkapitola popisuje vytvorené sady scén, ich rozdelenie a príklady. V tretej podkapitole je popísaný spôsob overenia presnosti parametrov kamery získaných v predošlej kapitole. V ďalších dvoch častiach tejto kapitoly sú popísané dve metódy použité pre stanovenie referenčnej pózy objektu.

4.1 Použité referenčné objekty

Pri riešení tejto práce bol pre detekciu objektu použitý algoritmus využívajúci detekciu hran v scéne. Preto boli ako referenčné objekty použité objekty s výraznými hranami. Ďalšou podmienkou objektu bolo umiestnenie počiatku súradnicového systému tak, aby sa pri položení dotýkal plochy stola. Zvolené referenčné objekty a ich parametre sú popísané v Tabuľka 4. Názvy objektov použitých v tejto práci sú iba orientačné a nemusia sa zhodovať s reálnym označením týchto objektov.

Všetky použité objekty boli vymodelované v *CAD* programe a uložené vo formáte *OBJ* tak, že všetky ich strany sú vytvorené z trojuholníkov.

Tabuľka 4 Referenčné objekty

Názov objektu	Render objektu	Hrany objektu	Rozmery [mm× mm× mm]
Clona			214,4×259×64
Poistková skriňa			52,7×153,2×33,5
Príruba			128×168,8×30
Veľká príruha			128×226,8×65
Valec			90×90×132
Príruba 2			172×88×27

Objekt poistková skriňa nakoniec nebol kvôli ťažkej detegovateľnosti v práci použitý, avšak fotky s ním sú dostupné v prílohe tejto práce.

Objektu s názvom valec, ako jedinému rotačnému objektu spomedzi všetkých, bola počítaná iba chyba pozície a výpočet chyby rotácie bol, kvôli nemožnosti presnej detekcie orientácie, vynechaný.

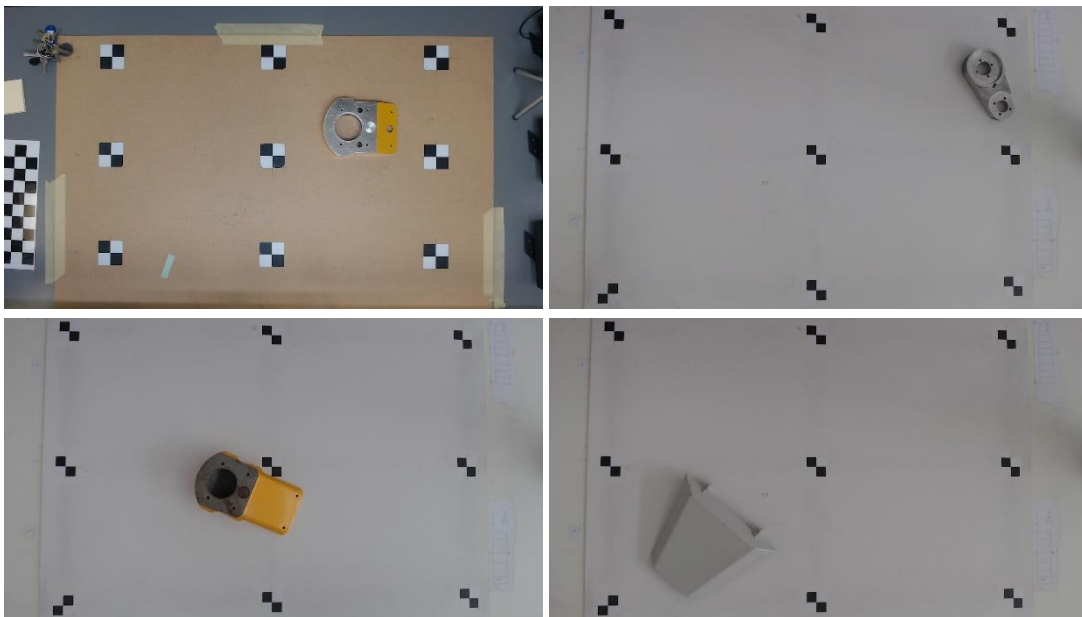
Ako hlavný objekt, použitý vo väčšine meraní, bol zvolený objekt s názvom *Príruba*, ktorý mal dobre viditeľné hrany, bol ľahko umiestniteľný do požadovanej pózy a vďaka svojej veľkej váhe a malej výške bol zo všetkých objektov najstabilnejší.

4.2 Vytvorené sady scén

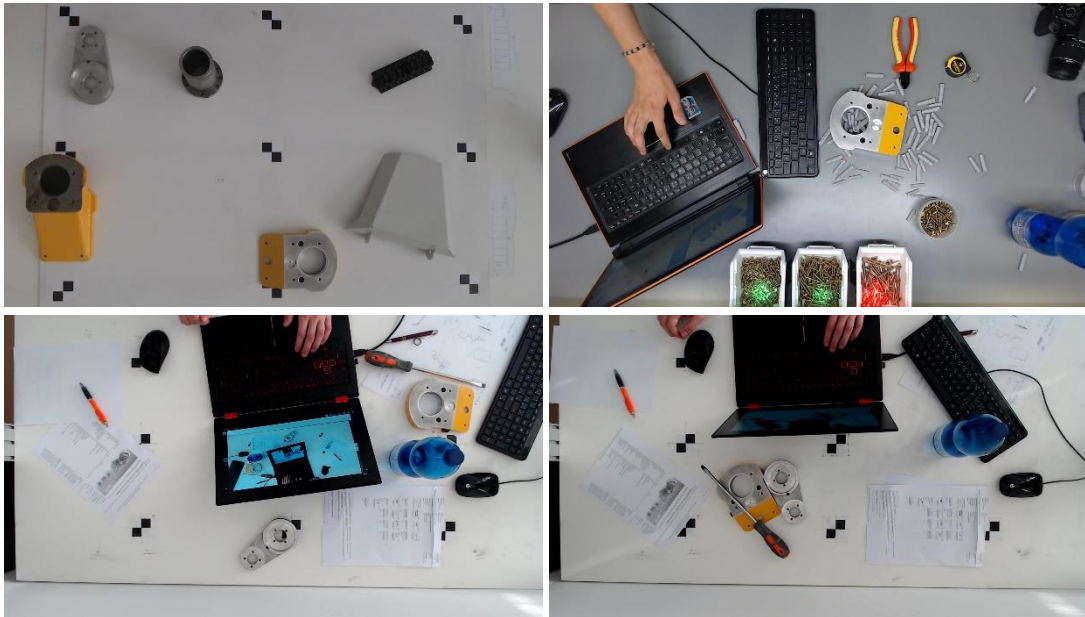
Celkovo boli pri realizácii práce vytvorené tri sady referenčných fotiek. Prvá sada zachytávala čisté scény bez ruchu s jedným objektom.

V druhej sade boli zachytené scény s objektom a okolitým ruchom, ktorý bol tvorený buď inými referenčnými objektmi, alebo náhodnými objektmi.

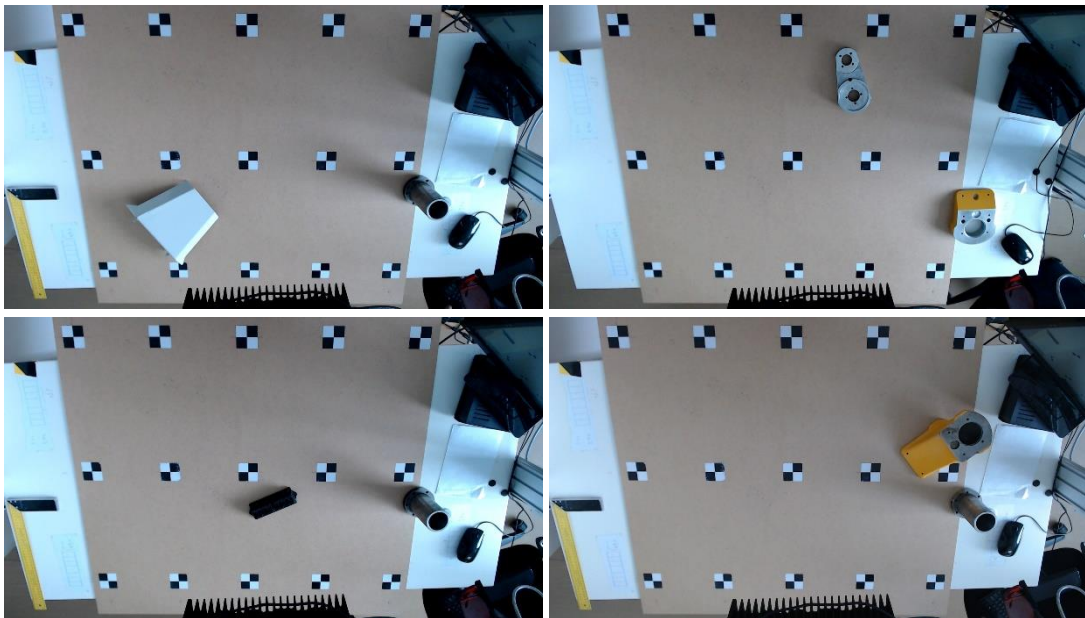
Posledná sada zachytávala scénu pod uhlom, čím bol objekt nepriamo rotovaný voči kamere až o približne $\pm 30^\circ$ okolo osí X a Y .



Obr. 22 Scény bez ruchu



Obr. 23 Scény s ruchom



Obr. 24 Scény s rotáciou objektu v X a Y osi

4.3 Kontrola správnosti získaných parametrov kamery

Skalibrovaním kamery boli získané všetky parametre potrebné na to, aby bolo možné určiť presnú projekciu ľubovoľného bodu na rovinu obrazu kamery, vďaka čomu je možné presne určiť, ako bude objekt o známej póze premietnutý na rovinu obrazu.

Vďaka vonkajším parametrom kamery je možné zo známej pózy objektu, položeného na stole, ktorá je definovaná ako H_{BO} , vypočítať pózu objektu voči kamere v matici H_{CO} . Pomocou tejto matice a použitím vnútorných parametrov kamery je možné presne určiť, ako bude objekt položený na stole premietnutý na rovinu obrazu kamery a porovnať reálnu scénu s vypočítanou pózou.

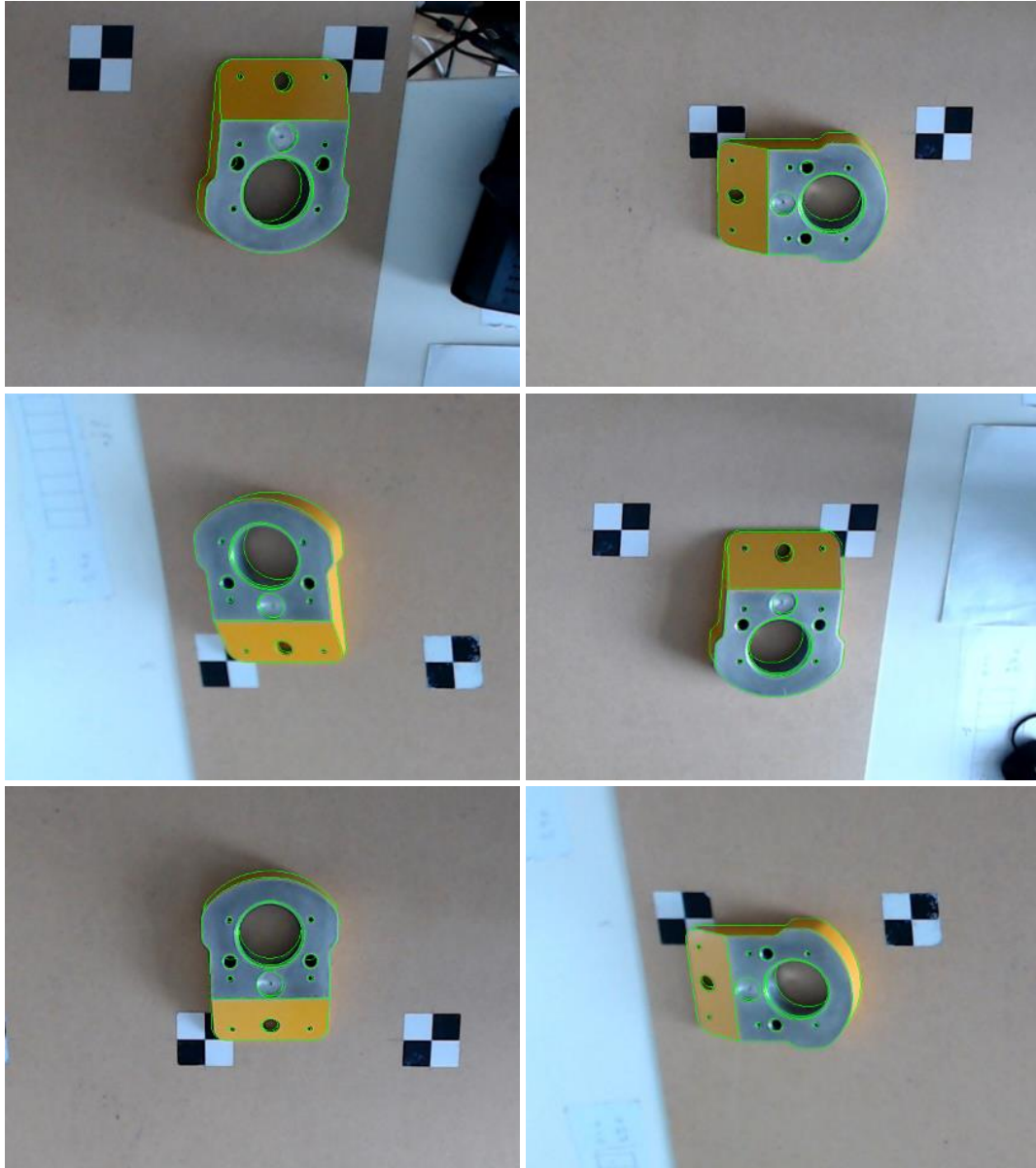
Tento predpoklad bol overený nasledovne. Boli vytvorené snímky známeho objektu umiestneného do predom určených pozícií na stole. Na týchto snímkach bolo následne odstránené skreslenie šošovky použitím funkcie `undistort`. Za použitia známych póz bol vytvorený hranový render objektu. Pre vytvorenie renderu bola vytvorená trieda využívajúca knižnicu *OpenCV*. Výstupný render však nedosahoval požadované presnosti, vplyvom nedokonalých extrakcií hrán a absenciou Z-bufferu, a preto bola pre vytvorenie renderu použitá knižnica *OpenGL*. Pre správne renderovanie bola projekčná matica v knižnici *OpenGL* definovaná tak, aby bola totožná s projekčnou maticou v *OpenCV* tvare, bol použitý vzťah

$$P_{GL} = \begin{pmatrix} \frac{2f_x}{w} & 0 & 1 - \frac{2c_x}{w} & 0 \\ 0 & \frac{2f_y}{h} & \frac{2c_y}{h} - 1 & 0 \\ 0 & 0 & \frac{Z_{max} + Z_{min}}{(Z_{min} - Z_{max})} & \frac{2Z_{max}Z_{min}}{(Z_{min} - Z_{max})} \\ 0 & 0 & -1 & 0 \end{pmatrix}, a \quad (31)$$

$$V_{GL} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad (32)$$

kde w a h sú šírka a výška renerovanej snímky a Z_{max} a Z_{min} sú rozsahy hĺbky, v ktorej je scéna renderovaná. Keďže knižnice *OpenCV* a *OpenGL* majú rôzne orientácie súradnicových systémov, je nutné použiť maticu V_{GL} (pohľadová matica), ktorá nastavuje súradnicové osi kamery v *OpenGL* do tvaru, v akom ich má knižnica *OpenCV*.

Render vytvorený podľa zadanej pózy objektu bol následne porovnaný s objektom na snímke z kamery. Výsledky porovnania sú zobrazené na Obr. 25, kde je poloha renderovaného objektu znázornená zelenými hranami objektu.



Obr. 25 Vizualizácia správnosti stanovenia parametrov kamery

Na snímkach je zreteľne vidieť, že poloha objektu a jeho render majú takmer rovnakú pozíciu bez ohľadu na polohu na snímke. Odchýlky sú väčšinou spôsobené nepresným umiestnením objektu do požadovanej pózy na stole. Malú nepresnosť taktiež vnáša chyba výpočtu vonkajších a vnútorných parametrov kamery.

Touto kontrolou bolo potvrdené, že hodnoty parametrov kamery získaných pomocou knižnice *OpenCV* sú správne a je možné ich použiť pre vytváranie vzoru a stanovenie referenčnej pózy objektu v scéne.

4.4 Referenčná póza pomocou prípravku

Metóda použitím prípravku bola navrhnutá ako jedna z metód stanovenia referenčnej pózy objektu. Prípravok pozostával z podkladového materiálu na ktorom bolo šesť referenčných značiek, ktorých vzdialenosti boli známe. Na takýto podkladový materiál bol prilepený objekt tak, aby nezakrýval ani jednu referenčnú značku. Podmienka podkladového materiálu bola minimálna hrúbka, keďže bolo požadované simulovať objekt položený na ploche stola, a preto bol ako podkladový materiál použitý papier.



Obr. 26 Objekt umiestnený na prípravku

Po vytvorení prípravku bolo potrebné presne definovať pózu objektu na ňom označenú ako H_{PO} . Táto póza bola stanovená vytvorením detailnej sady fotiek prípravku v rôznych pozíciách a rotáciách. Použitím knižnice *OpenGL* bola na každej snímke stanovená približná póza objektu. Následným priemerom získaných pozícií a rotácií bola stanovená matica H_{PO} .

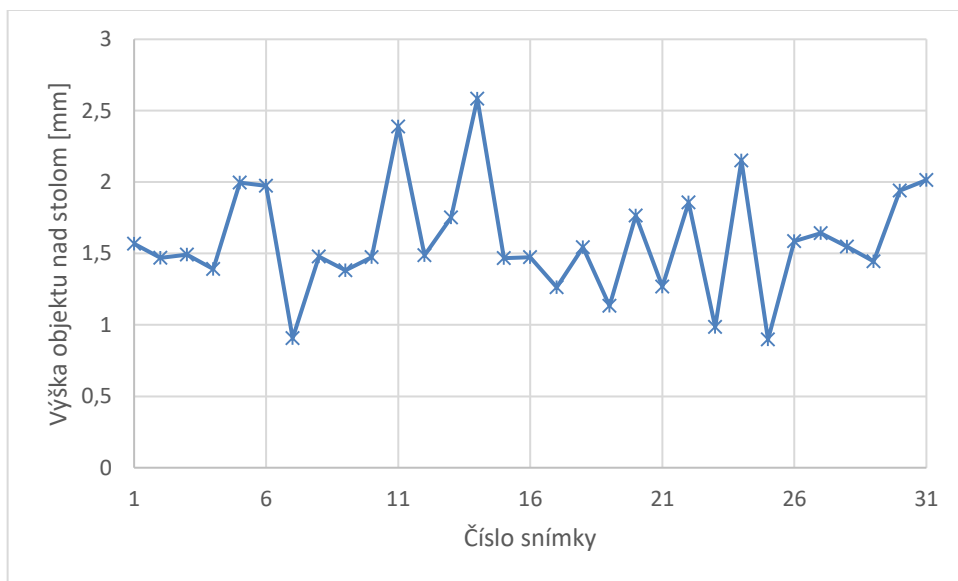


Obr. 27 Príklad stanovenia referenčnej pozície objektu

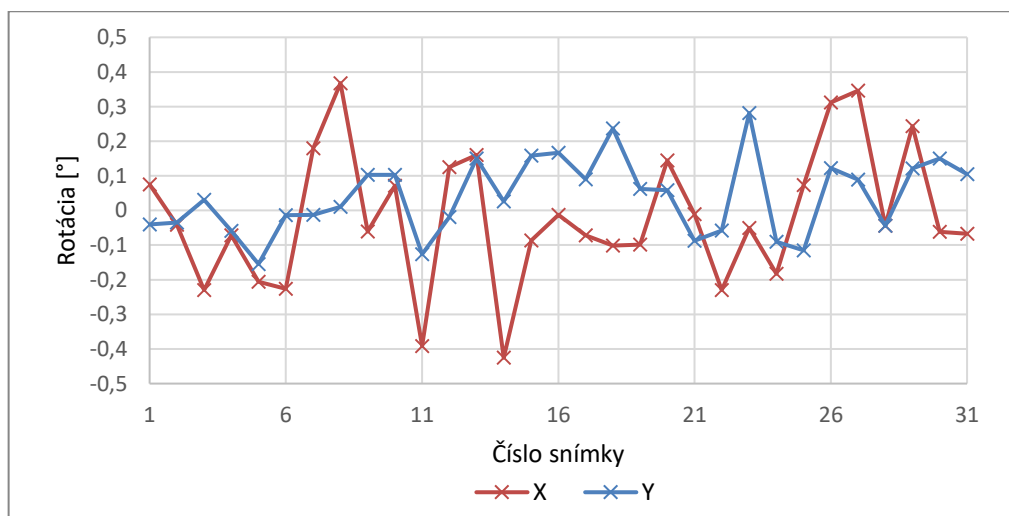
Následne bola použitím funkcie `solvePnP` určená matica H_{CP} a vypočítaná póza objektu voči kamere použitím vzťahu

$$H_{CO} = H_{CP} \cdot H_{PO} \quad (33)$$

Nevýhodou tejto metódy stanovenia referenčnej pózy objektu je oddelené stanovenie pózy kamery voči stolu H_{CB} a pózy prípravku voči kamere H_{CP} . Tento proces má až 6 stupňov voľnosti, čo spôsobuje, že výsledná výška objektu položeného na stole môže byť nenulová. Nenulové môžu byť aj rotácie okolo X a Y osí. Všetky tieto parametre by mali byť pri použitom objekte položenom na rovinu stola nulové. Ich hodnoty stanovené touto metódou sú zobrazené v Graf 12 a Graf 13.



Graf 12 Chyba stanovenia výšky objektu nad stolom

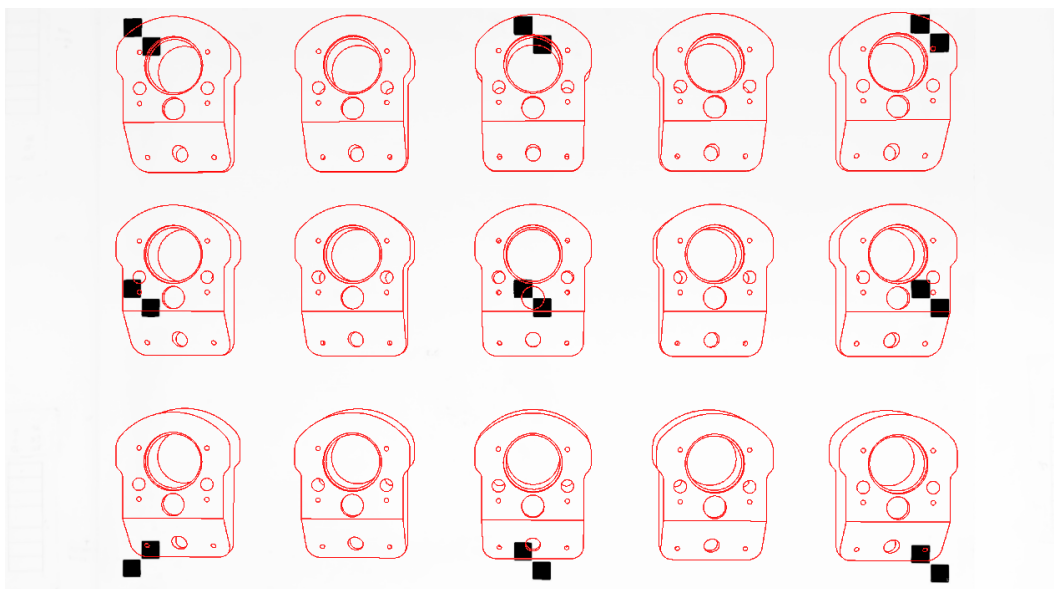


Graf 13 Chyba stanovenia rotácií okolo osí X a Y

Túto chybu stanovenia pózy by bolo možné znížiť väčším počtom značiek na prípravku a použitím rovnejšieho materiálu ako papier, ktorý svojou nerovnosťou taktiež vnášal do merania chyby. Napriek týmto opatreniam by sa chyba znížila, avšak nevyvynulovala. Z tohto dôvodu bola navrhnutá druhá metóda.

4.5 Referenčná pozícia pomocou filtru Sobel

Druhá navrhnutá metóda stanovenia referenčnej pozície objektu voči kamere využívala filter Sobel. Táto metóda pozostáva z dvoch krokov. V prvom kroku je manuálne stanovená hrubá póza objektu v scéne. Za použitia známej pozície kamery H_{CO} je do súradnicového systému stola pridaný render objektu. Využitím znalosti, že objekt je položený na ploche stola, môžeme konštatovať, že poloha objektu v osi Z a rotácie objektu v osiach X a Y sú nulové. Vďaka tejto znalosti je možné znížiť počet stupňov voľnosti objektu v scéne na 3. Objekt bol následne manuálne, posúvaním v osiach X a Y a rotáciou okolo osi Z , presne prispôbený objektu v scéne. Presnosť takéhoto manuálneho stanovenia pozície bola ± 1 mm v osiach X a Y a rotácie $\pm 0,5^\circ$ v osi Z , ktoré sú znázornené v Graf 14 a Graf 15.



Obr. 28 Zobrazenie rôznych polôh renderu objektu na ploche stola

Na takto získanú pózu objektu je následne aplikovaný algoritmus, ktorý sa postupným iterovaním snažil nájsť čo najpresnejšie pózu objektu v scéne.

4.5.1 Popis algoritmu

Vstupným parametrom algoritmu je scéna, v ktorej sa objekt nachádza, hrubo definovaná póza objektu a parametre kamery. V prvom kroku je pôvodná scéna prevedená do odtieňov sivej a následne je na ňu aplikovaný filter Sobel. Tento filter

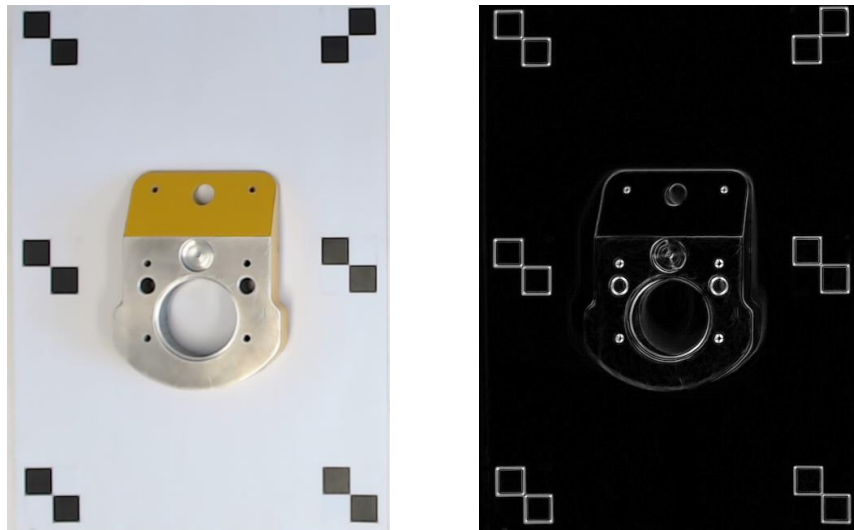
sa v spracovaní obrazu využíva na zvýraznenie hrán v scéne a príklad jeho použitia je vo Výpis 7 a zobrazený na Obr. 29.

Výpis 7 Príklad aplikácie filtru Sobel na snímku [20]

```
//Gradient X
Mat gradX, gradXAbs;
Sobel(gray, gradX, CV_16S, 1, 0, 3, 1, 0, BORDER_DEFAULT);
convertScaleAbs(gradX, gradXAbs);

//Gradient Y
Mat gradY, gradYAbs;
Sobel(gray, gradY, CV_16S, 0, 1, 3, 1, 0, BORDER_DEFAULT);
convertScaleAbs(gradY, gradYAbs);

//Celkový Gradient
Mat sobel;
addWeighted(gradXAbs, 0.5, gradYAbs, 0.5, 0, sobel);
```



Obr. 29 Výsledok aplikácie filtru Sobel na snímku

Pre stanovenie presnosti pozície objektu v scéne bol navrhnutý algoritmus, ktorý spojením Sobel snímky s renderom objektu vypočítal skóre, ktoré udáva mieru zhody renderu a objektu v scéne. Tento parameter je bezrozmerný a je využitý na porovnanie presnosti stanovenia póz. Funkcia pre jeho výpočet je zobrazená vo Výpis 8, kde je znázornený výpočet priemernej jasovej hodnoty z pixelov v Sobel snímke, ktoré sú prekryté hranovým renderom objektu.

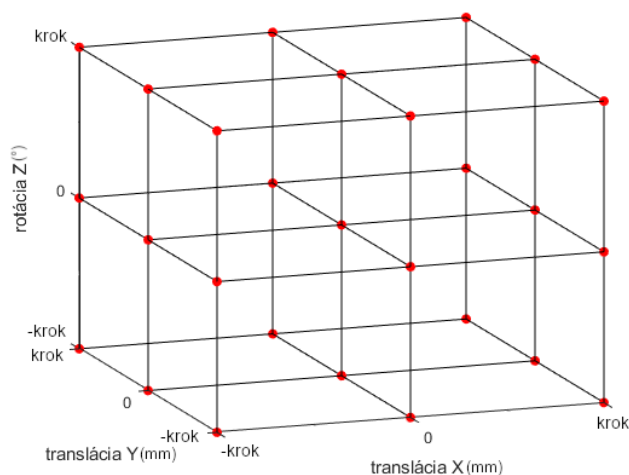
Výpis 8 Príklad výpočtu zhody stanovenej pózy s objektom v scéne

```
double calculateScore(Mat& sobel, Mat& render)
{
    Mat nonZero;
    findNonZero(render, nonZero);

    Mat score = sobel & render;

    return sum(score)[0] / (double)nonZero.rows;
}
```

Následne bola vytvorená trojrozmerná mriežka s rozmermi $3 \times 3 \times 3$, kde každej osi patril jeden zo zvyšných parametrov voľnosti. Každému prvku v mriežke tak prináležala vlastná póza v scéne, pre ktorú bolo určené skóre. V pseudokóde vo Výpise 9 je popísaný algoritmus hľadania najlepšej pózy. Funkcia `nájst_najlepšiu_pózu` hľadá v mriežke prvok s najlepšiu pózou objektu a ukladá ju do premennej `aktuálna_póza`. Ak je skóre všetkých prvkov v mriežke rovnaké, predpokladá sa, že krok mriežky je už taký jemný, že všetky jej prvky vytvárajú rovnaký výstup a funkcia vracia návratovú hodnotu 1, čím je algoritmus ukončený.



Obr. 30 Mriežka použitá pri hľadaní referenčnej pózy objektu

Výpis 9 Pseudokód algoritmu hľadania referenčnej pózy objektu

```
aktuálna_póza = hrubá_póza;
krok = 10; //inicializačný krok 10mm / 10°
while (1)
{
    mriežka = vytvoriť_mriežku(krok);

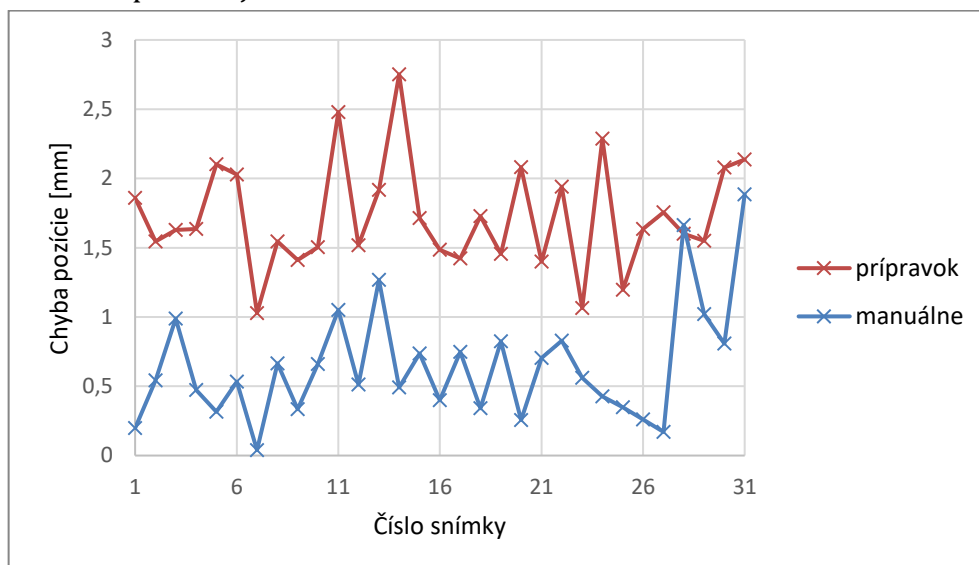
    if (nájst_najlepšiu_pózu(mriežka, aktuálna_póza) == 1)
    {
        break;
    }

    krok = krok / 1.2;
}
```

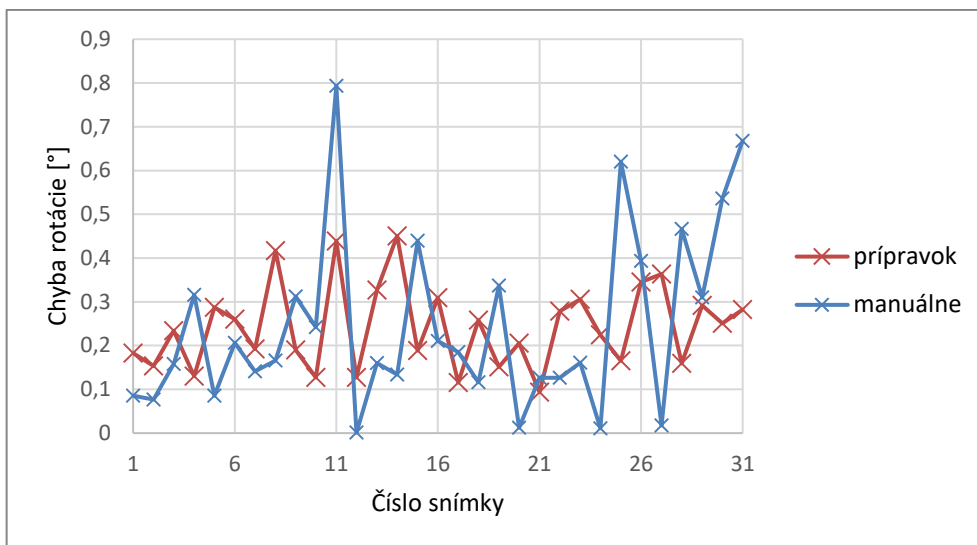
Veľkosť mriežky a miera znižovania kroku boli v tomto algoritme nastavené tak, aby bol výsledok čo najpresnejší a aby iterovanie nebolo príliš časovo náročné (~3 minúty na snímku).

Namiesto filtra Sobel je taktiež možné aplikovať na scénu Cannyho hranový filter a následne vytvoriť vzdialenostnú mapu scény. Pri riešení tejto práce však bolo po konzultácii s vedúcim práce od tejto možnosti upustené. Dôvodom bola tendencia čiastočnej alebo úplnej straty hrán objektu v scéne pri zle nastavených prahoch tohto filtra.

V tejto kapitole boli popísané dve možné metódy stanovenia referenčnej pozície objektu v scéne. Oba spôsoby preukázali správne výsledky a v ďalších častiach tejto práce je ako hlavná metóda určovania referenčnej pozície objektu použitá metóda s využitím filtra Sobel, ktorá na určenie pózy objektu nepotrebuje objekt umiestnený na prípravku a vďaka použitiu iteračného algoritmu vždy nájde lokálne maximálnu zhodu. Ďalšou výhodou tejto metódy stanovenia pozície objektu je nulovanie súradnice Z a rotácií X a Y objektu voči stolu, čím sa počet stupňov voľnosti objektu znižuje zo 6 na 3. V Graf 14 a Graf 15 sú namerané referenčné hodnoty porovnané s hodnotami stanovenými manuálne a pomocou prípravku. Všetky referenčné pózy boli uložené v tvare H_{CO} do súboru typu POSE, ktorý reprezentoval pózu objektu v scéne.



Graf 14 Chyba pozícií objektu stanovených manuálne a pomocou prípravku



Graf 15 Chyba rotácií objektu stanovených manuálne a pomocou prípravku

5 REALIZÁCIA STANOVENIA PÓZY OBJEKTU

Táto kapitola je venovaná praktickej realizácii stanovenia polohy a orientácie objektu voči kamere. Prvá podkapitola je venovaná algoritmu *Chamfer Matching*, konkrétne tvorbe vzoru, predspracovaniu scény a vylepšeným metódam detekcie objektu v scéne. V druhej podkapitole sú porovnané výsledky rôznych metód, dosiahnuté pri detekcii objektu. Posledná podkapitola je venovaná praktickému stanoveniu pózy objektu voči svetovému súradnicovému systému B .

5.1 Chamfer Matching

Táto podkapitola popisuje praktickú realizáciu algoritmu *Chamfer Matching*.

5.1.1 Vytvorenie sady vzorov

Vytvorenie sady vzorov v tejto práci bolo tvorené z dvoch krokov. V prvom kroku bolo potrebné definovať všetky pózy, v ktorých majú byť jednotlivé vzory vytvorené. Druhý krok pozostával z vytvorenia hranového renderu objektu a extrakcie kľúčových bodov pre vytvorenie vzoru.

5.1.1.1 Parametre sady vzorov

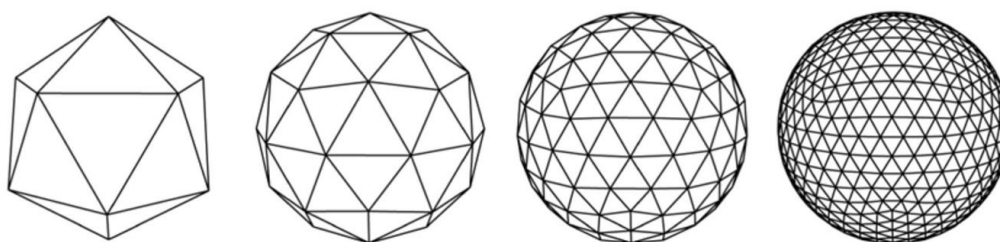
Pre vytvorenie sady vzorov bolo potrebné definovať predpokladané rozsahy póz objektu pred kamerou. Parametre stanovenia rozsahu póz boli: vzdialenosť od kamery, rotácie X a Y , rotácia kamery a perspektíva.

- **Vzdialenosť od kamery**

Tento parameter popisuje rozsah vzdialeností objektu od kamery.

- **Rotácie X a Y**

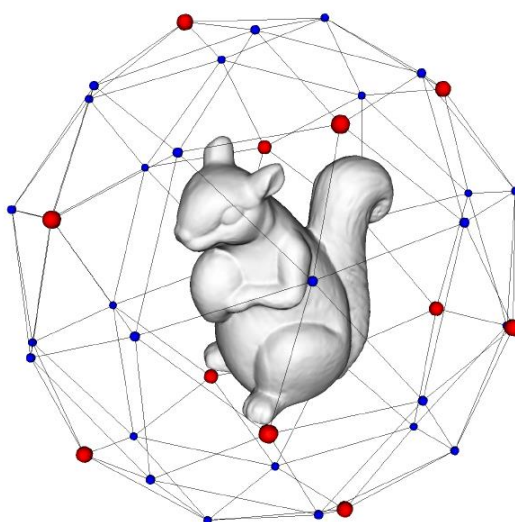
Tieto dve rotácie sú obsiahnuté v jednom parametri a popisujú rotáciu objektu pred kamerou v osiach X a Y . Pri výpočte týchto parametrov je pre rovnomerné rozloženie pohľadov kamery použitý dvadsaťsten (icosahedron). Použitím iteračného algoritmu pre trianguláciu gule je týmto tvarom možné dosiahnuť jemnejší krok zobrazený na Obr. 31 a v Tabuľka 5. Tento prístup bol inšpirovaný prácou [23] a je zobrazený na Obr. 32.



Obr. 31 Dvadsaťsten a jeho tri iterácie triangulácie gule

Tabuľka 5 Popis parametrov dvadsaťstenu pri rôznych iteráciách

Iterácia	Počet vrcholov	Počet stien	Uhol medzi susednými vrcholmi [°]
0	12	20	63,4349
1	42	80	31,7175
2	162	320	15,8587
3	642	1280	7,9294



Obr. 32 Príklad použitia dvadsaťstenu (červená) a jeho prvej iterácie (modrá) pri vytváraní vzoru [23]

- **Rotácia kamery**

Tento parameter popisuje rotáciu kamery okolo vlastnej osi Z. Pre presnú detekciu sa však odporúča tento parameter nechať v rozsahu uhlov 0-360° a radšej meniť krok tohto parametra.

- **Perspektíva**

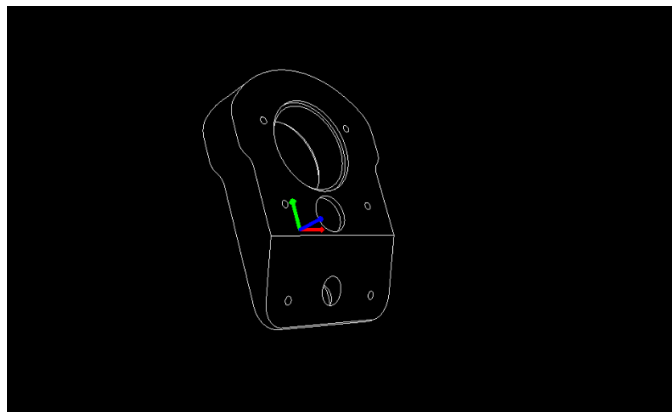
Keďže pri umiestnení objektu mimo optického stredu kamery dochádza ku skresleniu objektu vplyvom perspektívy, zobrazené na Obr. 28, sú v tejto práci vytvorené sady vzorov s rôznou perspektívou. Tento prístup síce zväčšuje veľkosť

celej sady, ale nespomaľuje detekciu a zaisťuje nájdenie objektu aj mimo optického stredu kamery.

Keďže počet nastaviteľných parametrov pre vytvorenie sady vzorov je päť, je vždy pri vytváraní vzoru potrebné vhodne zvoliť rozsah a krok daného parametra tak, aby vytvorená sada vzorov dosahovala dostatočne dobré výsledky a nízky detekčný čas. Všetky vzory sú uložené v 5-rozmernom dynamickom poli dátového typu `vector` a sú reprezentované ako body v triede `Point2`.

5.1.1.2 Redukcia bodov

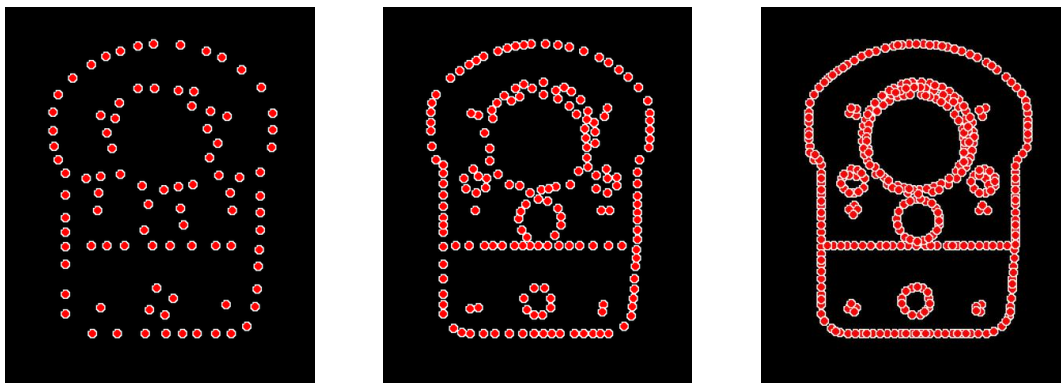
Z parametrov definovaných v predošlom odstavci boli jednotlivy z 3D modelu objektu vytvorené hranové projekcie. Pri vytváraní projekcií bola použitá knižnica *OpenGL*, ktorej výstup je zobrazený na Obr. 33. Následne boli z takejto snímky, použitím funkcie `findNonZero`, nájdené pozície všetkých nenulových bodov. Na ich redukciu boli navrhnuté a realizované dve metódy:



Obr. 33 Príklad renderu objektu použitím knižnice OpenGL

- **Náhodný výber**

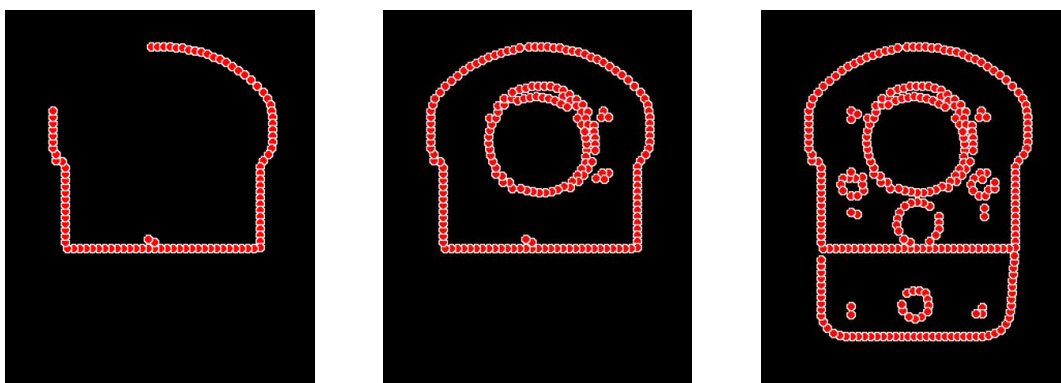
Táto metóda je založená na náhodnom výbere kľúčového bodu. Po výbere je však kontrolovaná jeho vzdialenosť od zvyšných kľúčových bodov. Ak je táto vzdialenosť väčšia ako nastavený prah, bod je pridaný medzi kľúčové body. V prípade, že náhodným výberom už nie je možné nájsť takýto bod, nastavený vzdialenostný prah je znížený a algoritmus pokračuje v hľadaní. Vďaka tomuto prístupu je zaručené rovnomerné rozloženie kľúčových bodov vzoru, pri rôznom počte použitých kľúčových bodov, zobrazené na Obr. 34. Vstupným parametrom algoritmu je percentuálna hodnota počtu bodov, ktoré majú byť zvolené ako kľúčové.



Obr. 34 Zobrazenie rovnomernosti rozloženia kľúčových bodov

- **Redukcia okolia**

Táto metóda bola navrhnutá pre zachovanie rovnakej vzdialenosti medzi jednotlivými kľúčovými bodmi. Algoritmus začína náhodným výberom pozície jedného kľúčového bodu. Následne sú všetky body v jeho okolí, vo vzdialenosti meranej šachovnicovou metrikou, odstránené a do vzoru je pridaný ďalší bod, najbližšie položený od predošlého. Takto sa postupuje až do úplného odstránenia všetkých kľúčových bodov. Vstupným parametrom algoritmu je vzdialenosť použitá pri redukcii okolia kľúčového bodu.



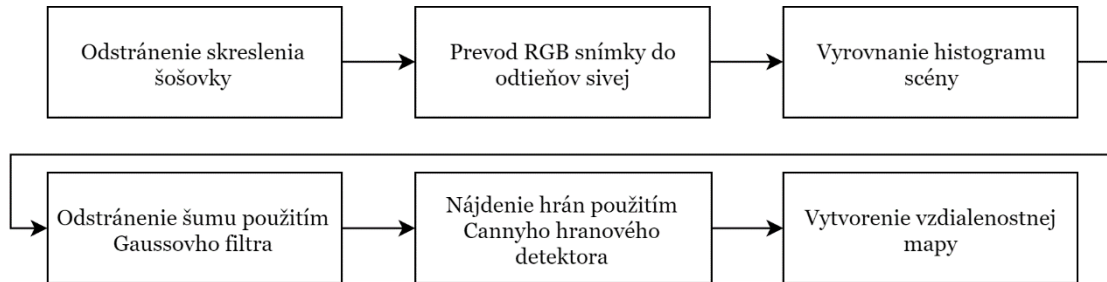
Obr. 35 Zobrazenie zachovania vzdialenosti kľúčových bodov

V tejto práci sa pri vytváraní sady vzorov osvedčilo použitie metódy náhodného výberu bodov, ktorá v porovnaní s redukciiou okolia dosahovala vyššiu rýchlosť extrakcie bodov a zaručovala rovnomerné rozloženie pri rôznom počte použitých kľúčových bodov vzoru. V práci bol pri vytváraní vzoru pôvodný počet kľúčových bodov redukovaný na 5-10 %, v závislosti od použitého objektu.

5.1.2 Predspracovanie scény

Pre vytvorenie vzdialenostnej mapy, ktorá je základom detekčnej metódy použitej v tejto práci, bol použitý reťazec zobrazený na Obr. 36. Príklad predspracovania

scény a vytvorenia vzdialenostnej mapy použitím knižnice *OpenCV* je zobrazený vo Výpis 10. Pri výpočte vzdialenostnej mapy bola použitá euklidovská metrika.



Obr. 36 Reťazec predspracovania scény

Výpis 10 Predspracovanie scény a vytvorenie vzdialenostnej mapy

```

/--Odstránenie skreslenia kamery
undistort(orig, //vstupný obraz
           undist, //výstupný obraz
           M, //vstupná matica kamery
           dC); //vstupné koeficienty skreslenia

/--Prevod RGB snímky do odtieňov sivej
cvtColor(undist, //vstupný obraz
         gray, //výstupný obraz
         COLOR_BGR2GRAY); //flag konverzie RGB2GRAY

/--Vyrovnanie histogramu použitím CLAHE
//Contrast Limited Adaptive Histogram Equalization
Ptr<CLAHE> clahe = createCLAHE();
clahe->setClipLimit(1);
clahe->apply(gray, gray);

/--Odstránenie šumu použitím Gaussovho filtra
GaussianBlur(gray, //vstupný obraz
             filter, //výstupný obraz
             Size(GAUSSIAN_SIZE, GAUSSIAN_SIZE), //veľkosť jadra
             0); //sigma

/--Použitie hranového filtra, metóda OTSU
Mat thresholdImg;
const double CANNY_AUTOMATIC_THRESH = threshold(filter, //vstupný obraz
        thresholdImg, //výstupný obraz
        0, //prah
        UCHAR_MAX, //maximálna hodnota
        THRESH_BINARY | THRESH_OTSU); //flag binárneho prahovania a metódy Otsu

Canny(filter, //vstupný obraz
      edge, //výstupný obraz
      CANNY_AUTOMATIC_THRESH / 4, //dolný prah
      CANNY_AUTOMATIC_THRESH / 2); //horný prah

/--Vytvorenie vzdialenostnej mapy
bitwise_not(edge, distance);
distanceTransform(distance, //vstupný obraz
                distance, //výstupný obraz
                DIST_L2, //Euklidova metrika
                DIST_MASK_3); //veľkosť masky
distance.convertTo(distance, CV_8UC1);
  
```

5.1.3 Detekčný algoritmus

Pri praktickej realizácii detekčného algoritmu boli použité všetky metódy popísané v podkapitole 1.4. Pre zvýšenie rýchlosti a robustnosti algoritmu boli pridané dva nové prístupy.

5.1.3.1 Hľadanie kandidátov

Pre presné nájdenie pozície objektu v scéne (nájdenie pozície vzoru v scéne, ktorý dosahuje globálne minimálnu hodnotu D) je potrebné prejsť celú scénu každým vzorom s krokom 1 px. Takýto prístup je však časovo extrémne náročný a takmer nerealizovateľný. Preto je v tejto práci použitá metóda, ktorá v scéne nahrubo nájde kandidátov s najnižšími hodnotami D . Následne je na každého kandidáta aplikovaný iteračný algoritmus, ktorý hľadá minimálnu hodnotu parametru D v ich okolí. Takýmto prístupom je znížený detekčný čas algoritmu pri miernom znížení robustnosti.

Hrubá detekcia prechádza celú scénu so stanoveným krokom a na každej pozícii nájde vzor s najnižšou hodnotou parametru D . V závislosti od nastaveného kroku sa priamoúmerne mení rýchlosť a taktiež chyba detekcie. Takýmto prístupom je vytvorená mriežka, z ktorej sú následne vybraní kandidáti. Pre výber kandidátov boli navrhnuté dve metódy:

- **Globálne minimá**

Táto metóda využíva prístup hľadania globálnych miním parametru D vo vytvorenej mriežke a je vhodná pri detekcii v scénach bez ruchu.

- **Lokálne minimá**

Táto metóda hľadá lokálne minimálne skóre v 8-okolí a je určená pre detekciu v scénach s pohybom.

Celý algoritmus hľadania kandidátov a následné hľadanie presnej pózy objektu sú paralelizované tak, aby program pracoval multi-vláknovo, vďaka čomu je znížený celkový čas detekcie.

5.1.3.2 Spätná kontrola chyby

Po nájdení konečnej pózy pre všetkých kandidátov je posledným krokom správne určiť výherného kandidáta. V scéne s pohybom často nastáva prípad, kedy kandidát s najnižšou hodnotou parametru D nie je hľadaný objekt, ale pohyb v scéne. Preto je výherný kandidát vybraný použitím metódy spätnej kontroly chyby, ktorá bola navrhnutá spolu s vedúcim tejto práce.

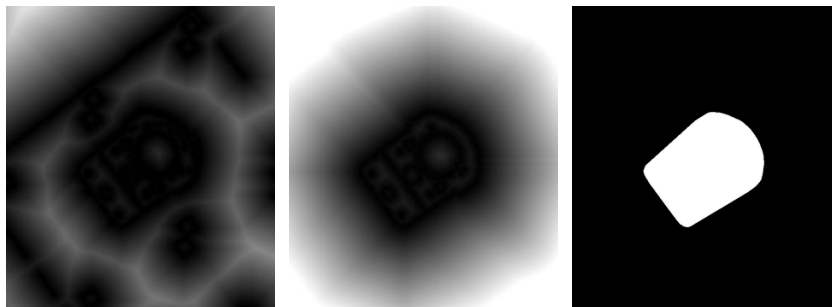
Táto metóda použitím kľúčových bodov vzoru kandidáta vytvorí vzdialenostnú mapu dis^T , ktorá je spätne porovnaná so vzdialenostnou mapou

scény dis^I v okolí objektu. Okolie objektu je dané maskou, ktorá je vytvorená ako konvexný obal kľúčových bodov použitím funkcie `convexHull`.

Týmto porovnaním je stanovená spätná chyba E_B , ktorá definuje celkovú chybu detekcie kandidáta. Je nutné dodať, že spätná kontrola chyby zaručuje správny výber iba z detegovaných kandidátov. Ak sa však medzi kandidátmi nenachádza správna póza, výsledkom je nepresná detekcia. Snímky použité pre výpočet spätnej chyby sú zobrazené na Obr. 37 a jej stanovenie vo Výpis 11.

Výpis 11 Spätný výpočet chyby kvadratickým priemerom

```
E_B = 0;
for (y = 0; y < dis_I.rows; y++)
{
    for (x = 0; x < dis_I.cols; x++)
    {
        if (mask(x, y))
        {
            E_B += (dis_I(x, y) - dis_T(x, y))^2;
        }
    }
}
E_B /= (sum(mask)[0] / UCHAR_MAX);
```



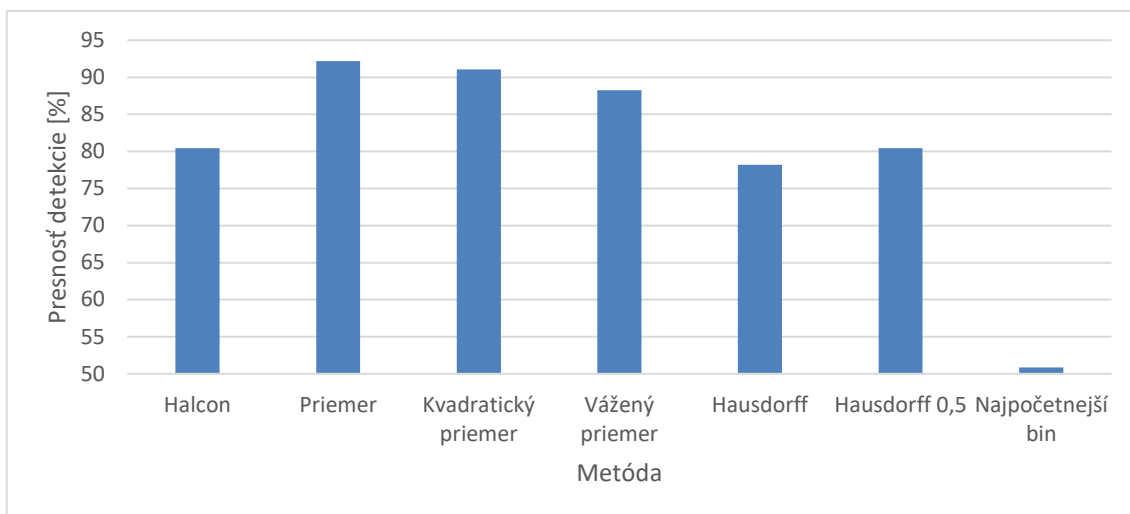
Obr. 37 Ukážka výpočtu spätnej chyby zľava: vzdialenostná mapa scény, vzdialenostná mapa vzoru, maska okolia objektu

5.2 Porovnanie dosiahnutých výsledkov

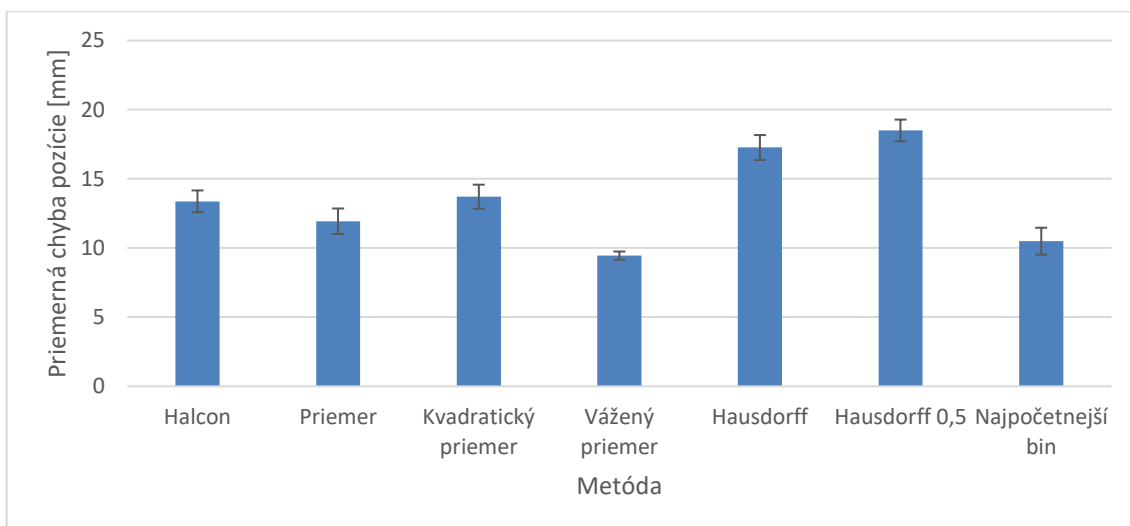
Táto podkapitola je venovaná porovnaniu dosiahnutých výsledkov použitím metód aplikovaných v tejto práci a knižnice *Halcon*. Výsledky sú rozdelené do troch skupín, a to podľa druhu scén, v ktorých boli merané.

Pre stanovenie percentuálnej presnosti detekcie boli z vizualizácie chýb detekcie zvolené prahy pozície a rotácie. Ak kontrolovaná detekcia presiahla stanovený prah aspoň v jednom z týchto parametrov, bola považovaná za chybnú.

5.2.1 Čisté scény



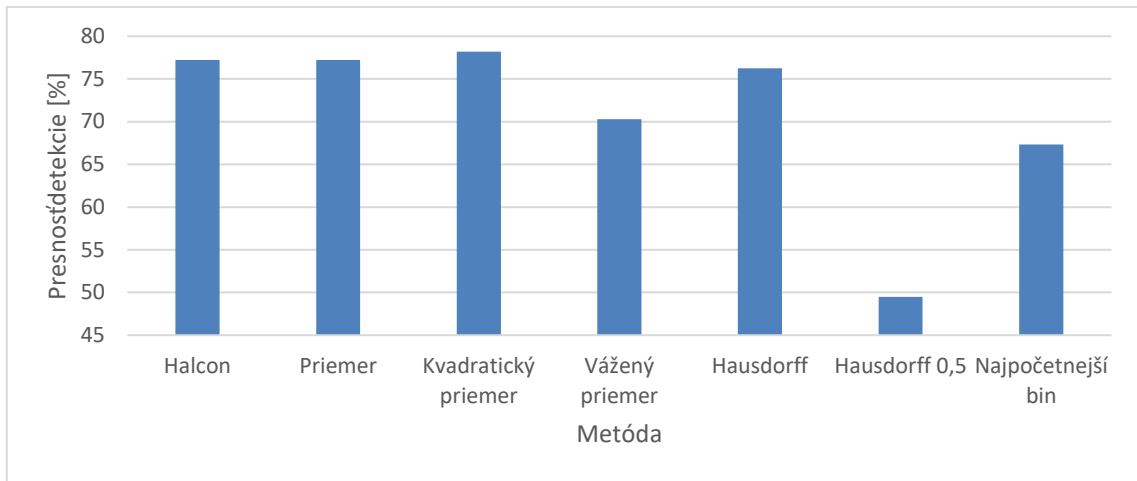
Graf 16 Percentuálna presnosť detekcie v čistých scénach



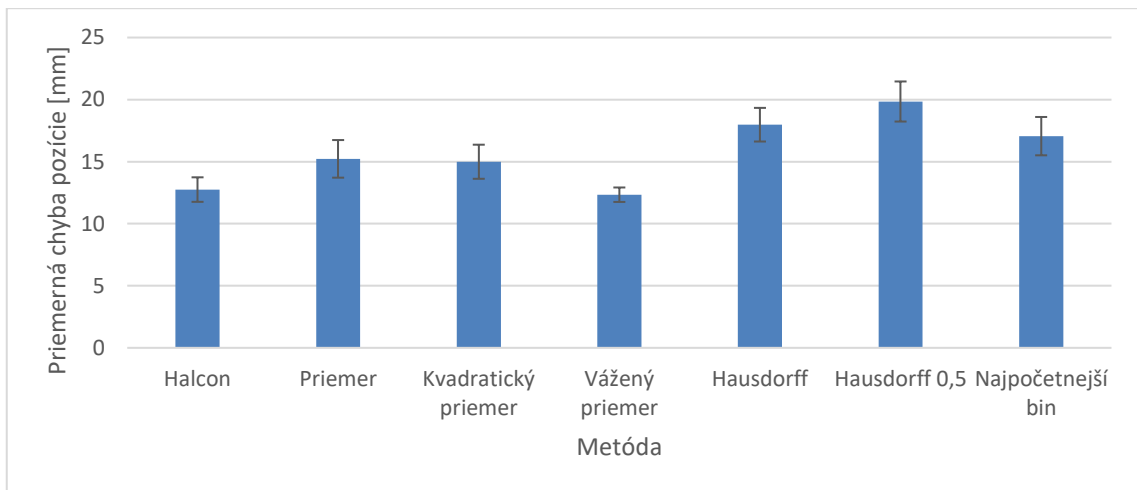
Graf 17 Chyba stanovenia pozície v čistých scénach

V čistých scénach bola dosiahnutá najvyššia presnosť detekcie použitím metódy priemeru a kvadratického priemeru. Knižnica *Halcon* dosiahla nízkych hodnôt kvôli neschopnosti detegovať objekty s nízkym kontrastom s podložkou stola. Najnižšia chyba stanovenia pozície bola dosiahnutá metódou váženého priemeru.

5.2.2 Scény s ruchom



Graf 18 Percentuálna presnosť detekcie v scéne s ruchom



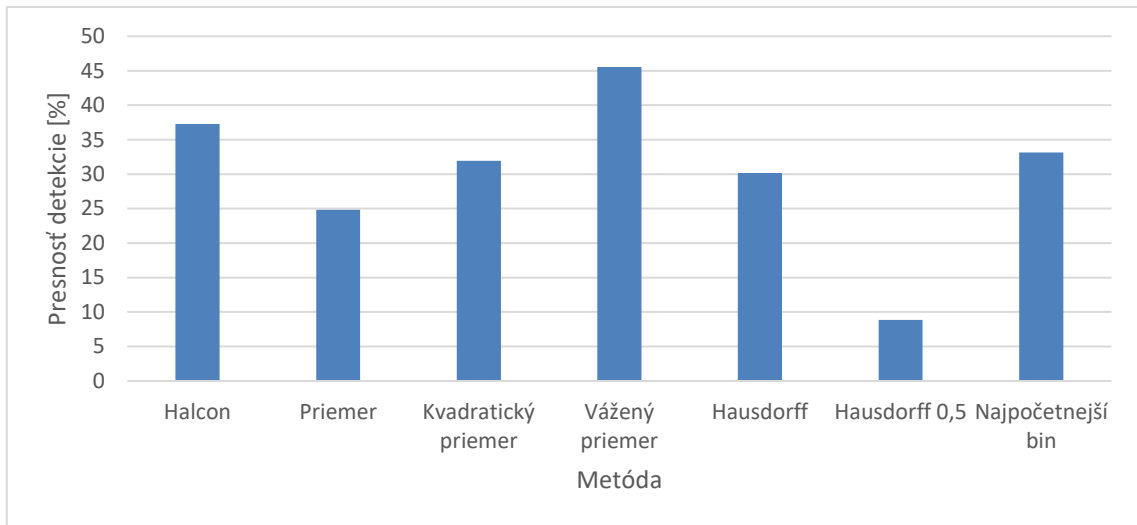
Graf 19 Chyba stanovenia pozície v scénach s ruchom

V scénach s ruchom nastal pokles celkovej percentuálnej presnosti detekcie kvôli zvýšeniu možných pozícií objektu vplyvom šumu.

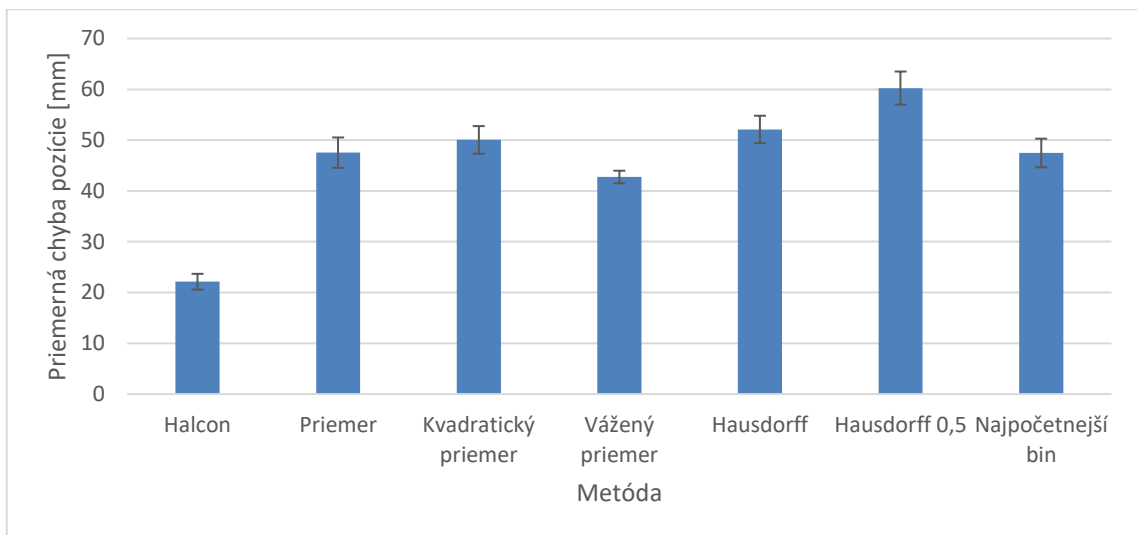
V týchto scénach dosiahli najvyššiu presnosť detekcie knižnica *Halcon*, metóda priemeru a metóda kvadratického priemeru čím sa potvrdilo tvrdenie o odolnosti kvadratického priemeru voči šumu z článku [8].

Najnižšiu chybu spomedzi testovaných metód dosiahla opäť metóda váženého priemeru a taktiež knižnica *Halcon*.

5.2.3 Scény s rotovaným objektom



Graf 20 Percentuálna presnosť detekcie v scéne s rotovaným objektom



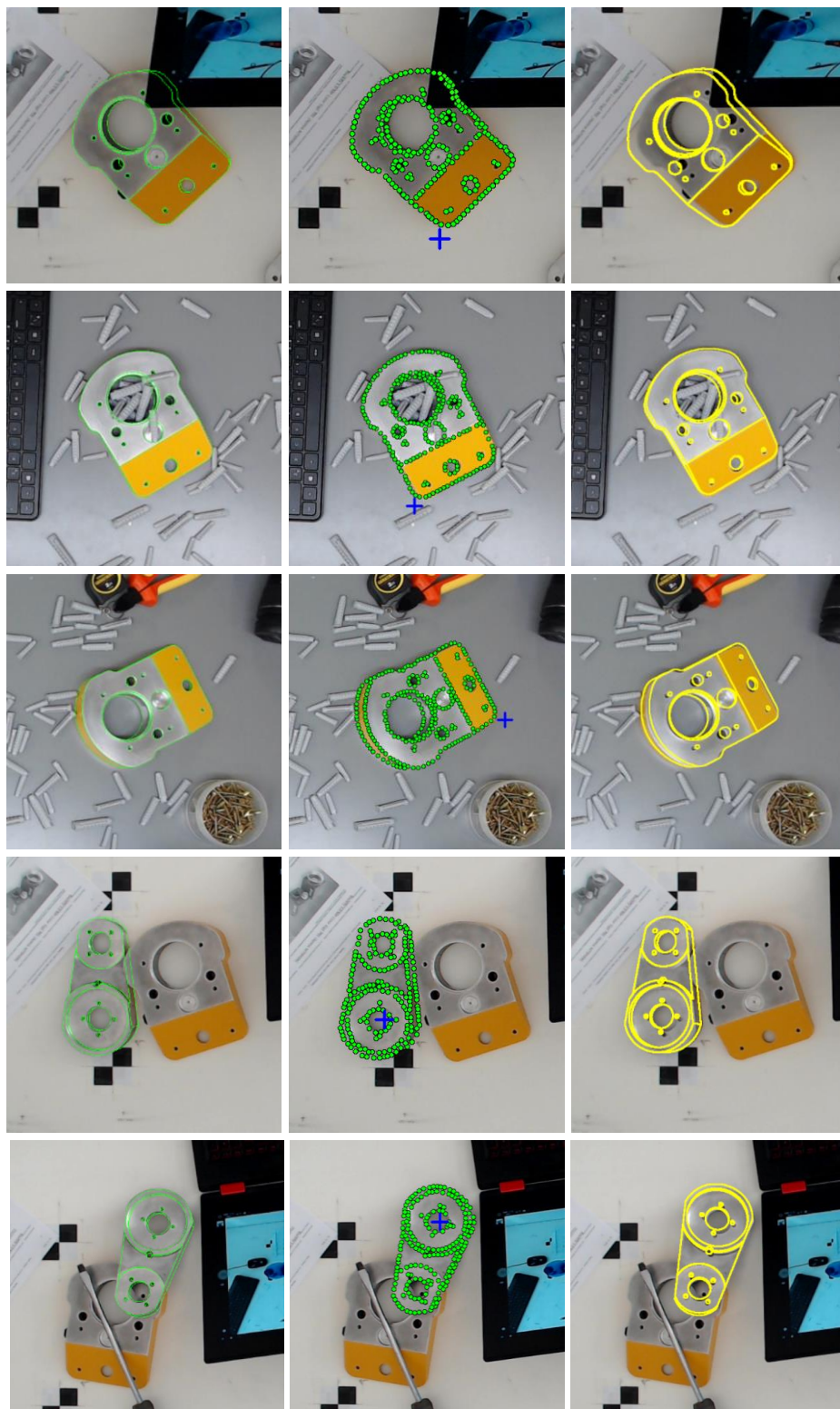
Graf 21 Chyba stanovenia pozície v scénach s rotovaným objektom

Scény s rotovaným objektom dosiahli najnižších presností detekcie kvôli zväčšenej sade vzorov a miernemu ruchu v scéne. Najvyššiu presnosť dosiahla metóda váženého priemeru a najnižšiu chybu pozície dosiahla knižnica *Halcon* a za ňou opäť metóda váženého priemeru.

Zo všetkých výsledkov vyplýva, že spomedzi testovaných metód dosiahli najlepšie výsledky presnosti detekcie metódy kvadratického priemeru a priemeru. Najnižšiu chybu pozície dosahovala v každom meraní metóda váženého priemeru.

Časy stanovenia pozície objektu, pre porovnanie jednotlivých metód, neboli pri riešení tejto práce merané, a preto táto práca neobsahuje parametre zostavy, na ktorej bola testovaná.

5.2.4 Vizualizácia výsledkov detekcie



Obr. 38 Vizualizácia robustnosti algoritmu v scénach s ruchom, zľava, referencia, Chamfer Matching algoritmus, Halcon

5.3 Stanovenie pózy objektu voči svetovému súradnicovému systému

Nájdením vzoru, ktorý dosahuje najvyššiu mieru zhody s objektom v scéne, je nájdená transformačná matica H_{CO} popisujúca vzťah medzi súradnicovým systémom kamery a objektu. Táto matica popisuje polohu a orientáciu objektu pred kamerou.

Výpočet transformačnej matice H_{BO} si je možné predstaviť ako prepočet pozície bodu P , ktorý leží v počiatku súradnicového systému objektu O s nulovou rotáciou. Vtedy platí vzťah

$$P_O = Trans(0,0,0).Rot_Z(0).Rot_Y(0).Rot_X(0) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (34)$$

Vďaka známej transformačnej matici H_{CO} je možné prepočítať polohu objektu do súradnicového systému kamery C , čím získame polohu P_C . Následne inverziou matice H_{CB} , ktorú sme získali vonkajšou kalibráciou kamery, získame maticu H_{BC} , pomocou ktorej môžeme prepočítať polohu bodu do bazového súradnicového systému B . Vynásobením matíc H_{BC} a H_{CO} vznikne matica H_{BO} popisujúca transformáciu ľubovoľného bodu zo súradnicového systému objektu O do bazového súradnicového systému B . Príklad implementácie výpočtu je ukázaný vo Výpis 12.

$$P_B = H_{CB}^{-1}.H_{CO}.P_O = H_{BC}.H_{CO}.P_O = H_{BO}.P_O \quad (35)$$

Výpis 12 Výpočet pozície objektu

```
//Póza objektu v súradnicovom systéme objektu
Mat P_O = Mat::eye(Size(4, 4), CV_64F);

//Póza objektu v súradnicovom systéme kamery
Mat P_C = H_CO * P_O;

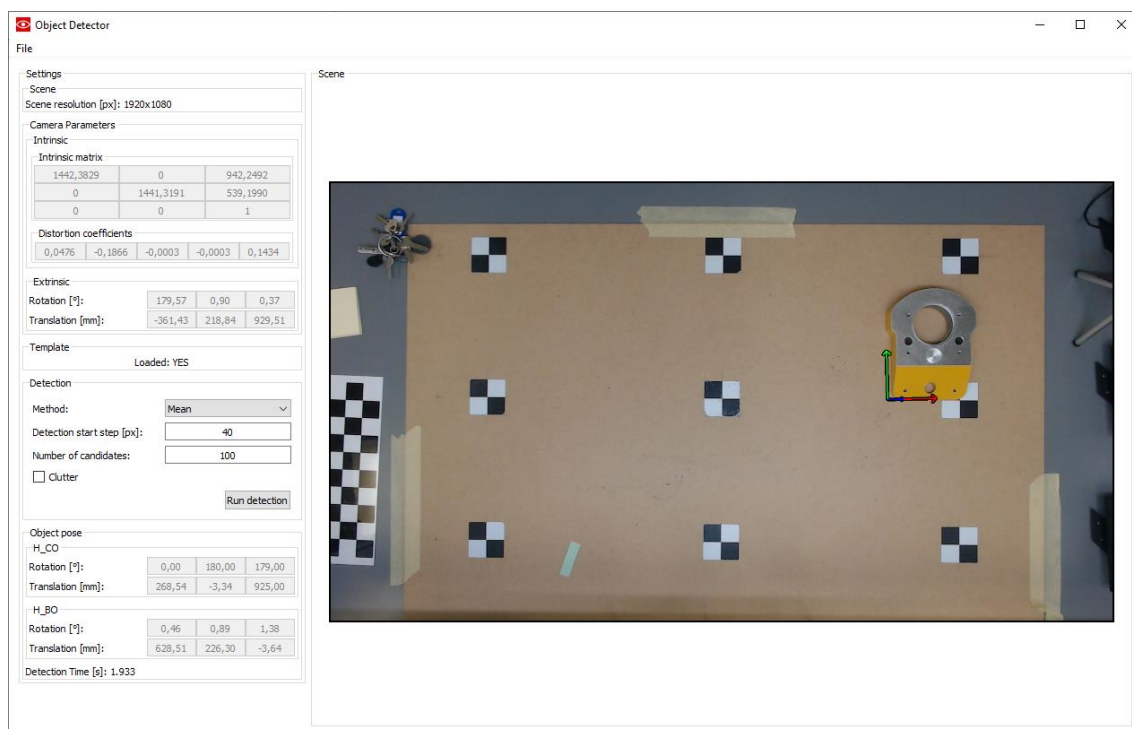
//Výpočet matice H_CB
Mat H_CB_R = homoRot(camera.rvec); //vytvorenie homogénnej rotačnej matice
Mat H_CB_t = homoTrans(camera.tvec); //vytvorenie homogénnej translačnej matice
Mat H_CB = H_CB_t * H_CB_R;

//Výpočet matice H_BC
Mat H_BC = H_CB.inv();

//Póza objektu vo svetovom súradnicovom systéme
Mat P_B = H_BC * P_C;
```

6 POPIS VYTVORENEJ APLIKÁCIE

V rámci prezentácie výsledkov dosiahnutých v tejto práci bola vytvorená aplikácia a názvom *Object Position*, ktorá realizuje detekciu a stanovenie pózy objektu. Aplikácia v sebe zahŕňa všetky metódy spracovania histogramu popísané v tejto práci. Grafické užívateľské prostredie bolo vytvorené použitím knižnice *Qt*.



Obr. 39 Hlavné okno aplikácie

6.1 Vstupné parametre

Parametre, ktoré je potrebné pred detekciou načítať do aplikácie sú nasledovné:

- **Vnútorne parametre kamery (*.cam)**

Tento súbor obsahuje maticu kamery, koeficienty skreslenia a rozlíšenie, pri ktorom boli dané parametre kamery získané.

- **Sada vzorov objektu (*.cmt)**

Tento súbor typu *Chamfer Matching Template* obsahuje vytvorené vzory, určené pre stanovenie pózy objektu. Obsahuje jednotlivé vzory reprezentované dátovým typom *Point2f*.

- **Scéna**

Obrazový súbor reprezentujúci scénu, v ktorej sa nachádza objekt, ktorého pozíciu chceme zistiť.

- **Vonkajšie parametre kamery (*.pose)**

Tento súbor obsahuje pózu kamery voči svetovému súradnicovému systému. Rotácie sú reprezentované Eulerovými uhlami v stupňoch v poradí rotácií X , Y , Z . Translácia je reprezentovaná v milimetroch. Tento parameter je nepovinný, ale bez jeho načítania nie je možné vypočítať polohu objektu vo svetovom súradnicovom systéme.

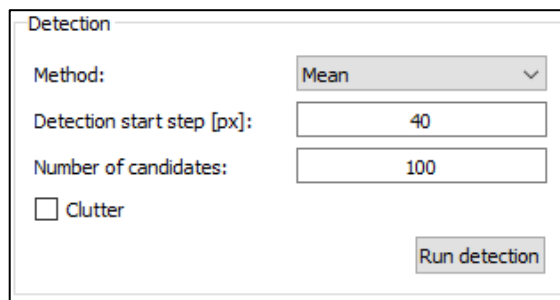
Načítané vstupné parametre sú hneď po vložení zobrazené vo vrchnej časti nastavení aplikácie.

Scene					
Scene resolution [px]:					
Camera Parameters					
Intrinsic					
Intrinsic matrix					
1442,3829	0	942,2492			
0	1441,3191	539,1990			
0	0	1			
Distortion coefficients					
0,0476	-0,1866	-0,0003	-0,0003	0,1434	
Extrinsic					
Rotation [°]:	0,00	0,00	0,00		
Translation [mm]:	0,00	0,00	0,00		
Template					
Loaded: NO					

Obr. 40 Vstupné nastavenia aplikácie

6.2 Nastavenie detekcie

Pred spustením detekcie je možné v strednej časti nastavení aplikácie zvoliť jednu zo šiestich metód stanovenia pozície objektu. Ďalšími voliteľnými parametrami sú počiatočný krok a počet kandidátov detekcie. Znížením počiatočného kroku a zvýšením počtu kandidátov je možné dosiahnuť v scénach s ruchom vyššiu presnosť detekcie. Taktiež sa odporúča pri detekcii v scéne s ruchom zaškrtnutie parametra *Clutter*, čím sú kandidáti vybraní lokálne s cieľom nájdania správneho objektu.

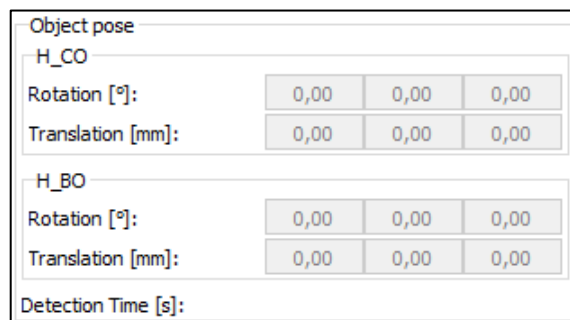


Detection	
Method:	Mean
Detection start step [px]:	40
Number of candidates:	100
<input type="checkbox"/> Clutter	
Run detection	

Obr. 41 Nastavenia detekcie

6.3 Výstupné parametre

Výstupom detekcie je póza objektu voči kamere H_{CO} a voči svetovému súradnicovému systému H_{BO} .



Object pose			
H_CO			
Rotation [°]:	0,00	0,00	0,00
Translation [mm]:	0,00	0,00	0,00
H_BO			
Rotation [°]:	0,00	0,00	0,00
Translation [mm]:	0,00	0,00	0,00
Detection Time [s]:			

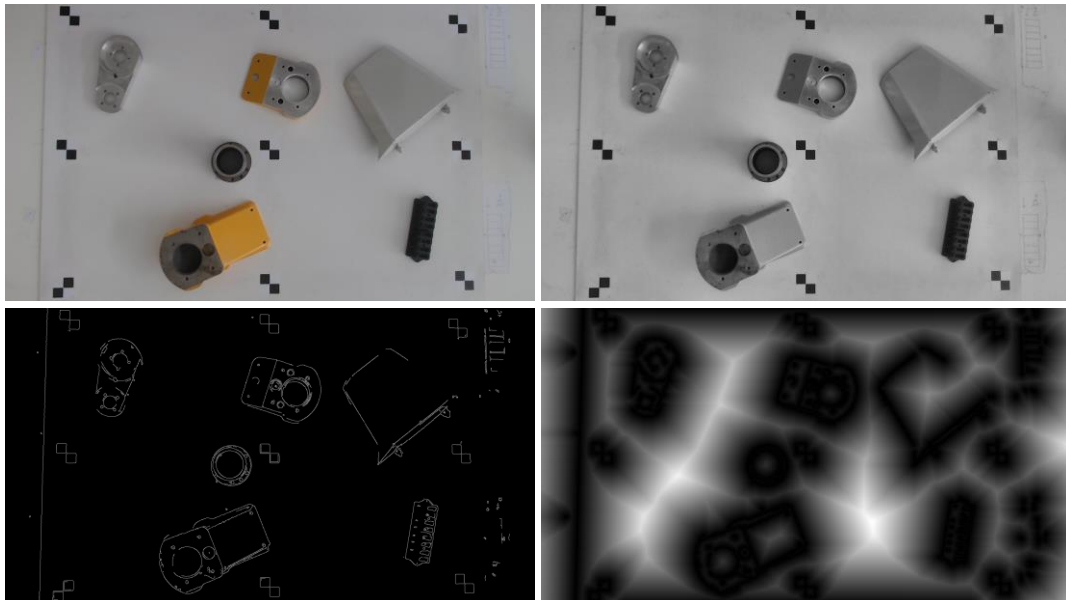
Obr. 42 Výsledky detekcie

6.4 Vizualizácia detekcie

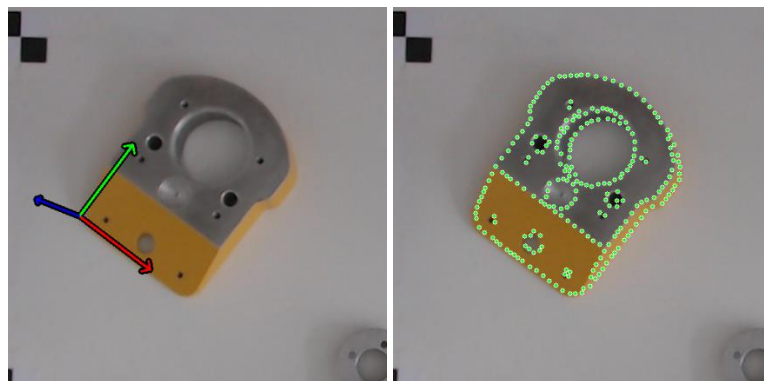
Pre vizualizáciu výsledku je v aplikácii možné nastaviť dva parametre zobrazenia. Prejdením kurzora myši na scénu a použitím kolieska na myši je možné zmeniť mód zobrazenia scény do štyroch módov zobrazených na Obr. 43.

Medzi tieto módy patrí zobrazenie pôvodnej scény s odstráneným skreslením kamery, scéna v odtieňoch sivej s aplikovaným vyrovnaním histogramu, hranová scéna vytvorená pomocou Cannyho hranového detektora a adaptívneho zistenia prahu a vzdialenostná mapa vytvorená v euklidovskej metrike.

Ďalšia možnosť zmeny zobrazenia výsledku detekcie je dvojitým kliknutím na scénu, čím sa zmení spôsob zobrazenia detegovaného objektu z vykresľovania súradnicových osí na vykreslenie kľúčových bodov vzoru a naopak podľa Obr. 44.



Obr. 43 Módy zobrazenia scény, zľava hore: pôvodná snímka, odtiene sivej, hranová snímka, vzdialenostná mapa



Obr. 44 Módy zobrazenia detekcie, zľava: súradnicový systém, kľúčové body

Záver

Cieľom diplomovej práce bolo stanovenie pozície objektu vo svetovom súradnicovom systéme pomocou monokulárnej kamery.

Postup stanovenia pozície objektu je teoreticky vysvetlený v prvej časti tejto práce, v ktorej sú popísané dve metódy stanovenia pozície objektu. Metóda *Chamfer Matching* bola po dohode s vedúcim práce zvolená ako detekčná metóda použitá v tejto práci.

V druhej časti sú popísané vývojové nástroje použité pri riešení zadania tejto práce a dôvody ich použitia. Ako komerčne dostupné riešenie pre detekciu objektu monokulárnou kamerou bola zvolená knižnica *Halcon*, ktorá je použitá na porovnanie s výsledkami dosiahnutými metódou zvolenou v tejto práci.

Kalibrácia kamery použitím knižníc *OpenCV* a *Halcon* je popísaná v tretej kapitole. V tejto kapitole je popísaná kamera *Logitech HD Pro Webcam C920* zvolená pri riešení práce a dôvody jej voľby. Pre zlepšenie kalibrácie bol v tejto práci testovaný vplyv priemerovania snímok na presnosť kalibrácie zobrazený v Graf 5. Ďalším testovaným parametrom, vplývajúcim na kvalitu kalibrácie, bol materiál, z ktorého bol zhotovený kalibračný vzor. Podľa výsledkov zobrazených v Graf 6 bol pre vytvorenie kalibračného vzoru zvolený materiál: zrkadlo. Závislosť vplyvu vzoru použitého pre kalibráciu je zobrazená v Graf 11, na základe čoho bola zvolená ako referenčná kalibrácia použitím kruhovej mriežky. V tejto kapitole je taktiež popísaný spôsob, akým boli stanovené vonkajšie parametre kamery, popis voľby kalibračných značiek, spôsoby ich detekcie a výpočet pózy kamery. Výsledok detekcie značiek je zobrazený v Obr. 21.

Štvrtá kapitola popisuje objekty použité v scénach a tri sady snímok, ktoré boli vytvorené pre stanovenie pozície objektu. Ďalej sa venuje overeniu správnosti získaných parametrov kamery pomocou renderu hranového objektu za použitia knižnice *OpenGL*. Taktiež sú v tejto kapitole popísané dve metódy, pomocou ktorých bola zisťovaná referenčná póza objektu v scéne. Ako referenčná metóda bola použitá metóda využívajúca filter Sobel.

Praktická realizácia detekcie objektu je popísaná v piatej kapitole, kde je popísaný spôsob vytvorenia sady vzorov z *OBJ* súboru obsahujúceho model objektu. Ďalej sú tu porovnané praktické výsledky použitých metód detekcie objektu. Z metód použitých v tejto práci dosiahli najlepšiu presnosť metódy kvadratického priemeru, priemeru a váženého priemeru. Metódou s najmenšou chybou stanovenia pozície bola metóda váženého priemeru. Výsledky spolu s porovnaním s knižnicou *Halcon* sú zobrazené v Graf 16 až Graf 21. Knižnica *Halcon* dosiahla taktiež vysoké hodnoty presnosti. Jej nevýhoda sa však ukázala v scénach s nízkym kontrastom,

kde nebola schopná nájsť objekt v scéne, a taktiež v scénach s rotovaným objektom, kde mali výsledky tejto knižnice vysoké chyby v rotáciách.

Spomedzi testovaných objektov boli najlepšie detekovateľné objekty *Príruba*, *Veľká príruba* a *Valec*. *Príruba 2* bola kvôli nízkemu kontrastu s pozadím často ťažko detekovateľná knižnicou *Halcon*. Najhoršie detekovateľné boli objekty *Clona* a *Poistková skriňa*. *Clona* bola často ťažko rozlíšiteľná s pozadím a obsahovala málo hrán. *Poistková skriňa* bola naopak malá a mala veľké množstvo hrán. Preto je pre správnu detekciu objektu nutné zaistiť kontrast objektu s pozadím a primerané množstvo hrán objektu.

Ideálne umiestnenie objektu v scéne je závislé od rozsahu, v akom je vygenerovaná sada vzorov. Nízka presnosť detekcie bola zistená v rohoch scény, kde bol objekt najviac vzdialený od optického stredu kamery a v týchto pozíciách neboli vygenerované vzory objektu.

Posledná kapitola popisuje aplikáciu, ktorá bola vytvorená pre prezentáciu výsledkov detekcie objektu. Pomocou tejto aplikácie je možné, načítaním scény, parametrov kamery a sady vzorov objektu, realizovať detekciu za použitia zvolenej metódy. Aplikácia podporuje rôzne režimy zobrazenia scény a výsledku detekcie.

Výhodou metódy použitej v tejto práci je vytvorenie vzorov iba za pomoci 3D modelu objektu bez nutnosti vytvárania jeho reálnych snímok. Ďalšou výhodou je možnosť vytvorenia sady vzorov pre všetky rotácie okolo objektu použitím dvadsaťstenu a triangulácie gule, pričom knižnica *Halcon* umožňuje iba rotáciu $\pm 90^\circ$ v osiach *X* a *Y*. Metóda je taktiež odolná voči zmene jasú, vďaka úprave kontrastu scény a adaptívnej detekcii hrán.

Nevýhodou tejto metódy je nemožnosť presného stanovenia pozície objektu, ktorý je symetrický. Ďalšou nevýhodou je zníženie rýchlosti detekcie pri použití sady s veľkým počtom vzorov. Rýchlosť je možné zvýšiť na úkor presnosti detekcie, nastavením vyššieho inicializačného kroku detekcie a znížením počtu vybraných kandidátov.

V tejto práci bolo potvrdené, že presnosť stanovenia pozície objektu v scéne, použitím algoritmov založených na vzoroch, vo veľkej miere závisí od množstva vytvorených vzorov objektu. Zatiaľ čo presnosť *X* a *Y* súradníc zobrazená na Obr. 38 bola vysoká, súradnica *Z* mala kvôli vysokému kroku vzorov nižšiu presnosť. Zvýšením počtu vzorov je preto možné dosiahnuť vyššiu presnosť stanovenia pozície, na úkor zvýšenia detekčného času.

Literatúra

- [1] OpenCV [online]. [cit. 2018-12-31]. Dostupné z: <https://opencv.org/>
- [2] OpenCV Documentation: *Camera Calibration and 3D Reconstruction* [online]. [cit. 2018-12-31]. Dostupné z: https://docs.opencv.org/master/d9/d0c/group_calib3d.html
- [3] MathWorks: *What Is Camera Calibration?* [online]. [cit. 2018-12-31]. Dostupné z: <https://www.mathworks.com/help/vision/ug/camera-calibration.html>
- [4] Wiki Qt: *About Qt* [online]. [cit. 2018-12-31]. Dostupné z: https://wiki.qt.io/About_Qt
- [5] Sightations: *Dissecting the Camera Matrix, Part 3: The Intrinsic Matrix* [online]. [cit. 2018-12-31]. Dostupné z: <http://ksimek.github.io/2013/08/13/intrinsic/>
- [6] OpenCV Documentation: *Camera calibration With OpenCV* [online]. [cit. 2019-01-01]. Dostupné z: https://docs.opencv.org/2.4/doc/tutorials/calib3d/camera_calibration/camera_calibration.html
- [7] Parametric correspondence and chamfer matching: *Two new techniques for image matching* [online]. [cit. 2019-04-30]. Dostupné z: <https://www.ijcai.org/Proceedings/77-2/Papers/024.pdf>
- [8] BORGEFORS, Gunilla. *Hierarchical Chamfer Matching: A Parametric Edge Matching Algorithm*. 1988.
- [9] SONKA, Milan, Vaclav HLAVAC a Roger BOYLE. *Image Processing, Analysis, and Machine Vision: International Edition*. 4th edition. United States of America: Cengage Learning, 2015. International Edition. Rozsah strán (589 – 602). ISBN 11-335-9369-0.
- [10] Computer Vision Lecture 23: *3-D Pose Object Recognition* [online]. [cit. 2019-01-01]. Dostupné z: http://users.umiacs.umd.edu/~ramani/cmsc426/Lecture23_3Dpose.pdf
- [11] TOMASI, Carlo. *Vector Representation of Rotations* [online]. [cit. 2019-01-03]. Dostupné z: <https://www2.cs.duke.edu/courses/compsci527/fall13/notes/rodrigues.pdf>
- [12] BURIAN, František. *Robotika: Stacionární robotika* [online]. [cit. 2019-01-01]. Dostupné z: http://www.uamt.feec.vutbr.cz/~robotika/prednasky/2017_MRBT_stacionarni.pdf
- [13] KLEČKA, Jan. *Rozpoznávání: Vícesnímková rekonstrukce scény* [online]. [cit. 2019-01-01]. Dostupné z: http://midas.uamt.feec.vutbr.cz/ROZ/roz_cz.php

- [14] ŽÁRA, Jiří, Bedřich BENEŠ, Jiří SOCHOR a Petr FELKEL. Moderní počítačová grafika. 2. přepracované a rozšířené vydání. Brno: Computer Press, 2010. Rozsah stránek (541 – 549). ISBN 80-251-0454-0.
- [15] Microsoft Visual Studio: *Wikipedia* [online]. [cit. 2019-01-02]. Dostupné z: https://en.wikipedia.org/wiki/Microsoft_Visual_Studio
- [16] Logitech HD Pro Webcam C920: *Specifications* [online]. [cit. 2019-04-29]. Dostupné z: https://support.logitech.com/en_us/product/hd-pro-webcam-c920/specs
- [17] LIŠTVAN, Jaroslav. Kalibrace kamery pro robotické pracoviště [online]. [cit. 2019-04-09]. Dostupné z: <https://dspace.cvut.cz/bitstream/handle/10467/68387/F3-BP-2017-Listvan-Jaroslav-Kalibrace%20kamery%20pro%20roboticke%20pracoviste.pdf>
- [18] calib.io: *Calibration targets* [online]. [cit. 2019-04-09]. Dostupné z: <https://calib.io/>
- [19] CLAUS, David, Andrew W. FITZGIBBON. A Rational Function Lens Distortion Model for General Cameras [online]. [cit. 2019-04-12]. Dostupné z: http://www.robots.ox.ac.uk/~dclaus/publications/clus05rf_model.pdf
- [20] OpenCV: *Sobel Derivatives* [online]. [cit. 2019-05-04]. Dostupné z: https://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/sobel_derivatives/sobel_derivatives.html
- [21] KONISHI Yoshinori, Yuki HANZAWA, Masato KAWADE, Manabu HASHIMOTO. Fast 6D Pose Estimation from a Monocular Image using Hierarchical Pose Trees. 2016.
- [22] KONISHI Yoshinori, Yoshihisa IJIRI, Masaki SUWA, Masato KAWADE. Textureless object detection using cumulative orientation feature. [online]. [cit. 2019-04-30]. Dostupné z: https://www.researchgate.net/profile/Yoshinori_Konishi/publication/288344393_Textureless_object_detection_using_cumulative_orientation_feature/links/56805b8008ae1e63f1e95679/Textureless-object-detection-using-cumulative-orientation-feature.pdf
- [23] TJADEN Henning, Ulrich SCHWANECKE, Elmar SCHÖMER. Real-Time Monocular Pose Estimation of 3D Objects using Temporally Consistent Local Color Histograms. [online]. [cit. 2019-05-01]. Dostupné z: http://cvmr.info/resources/Paper/TCLCH_ICCV2017.pdf
- [24] MVTec Halcon: *Product Information*. [online]. [cit. 2019-05-03]. <https://www.mvtec.com/products/halcon/product-information/>

Zoznam symbolov, veličín a skratiek

RGB	farebný model pozostávajúci z troch farebných zložiek: červená, zelená a modrá
2D	dvojrozmerný
3D	trojrozmerný
MSDN	Microsoft Developer Network
IDE	integrované vývojové prostredie (z angl. Integrated Development Environment)
OpenCV	Open Source Computer Vision
OpenGL	Open Graphics Library
OBJ	typ súboru reprezentujúci priestorovú geometriu
VS	Microsoft Visual Studio
GUI	grafické používateľské rozhranie (z angl. Graphical User Interface)
FPS	snímky za sekundu (z angl. Frames per second)
CAD	počítačom podporovaný návrh (z angl. Computer-aided design)
GPU	grafický procesor (z angl. Graphics processing unit)

Zoznam príloh

Príloha 1. priložené DVD

Obsah priloženého DVD

Na DVD priloženom k tejto práci je uložená elektronická verzia práce a snímky vytvorené pri riešení práce. Ďalej sa tu nachádzajú sady vzorov použité pri detekcii, projekt s aplikáciou vytvorenou pre túto prácu a výsledná aplikácia. Názvy zložiek a ich obsahy sú nasledovné:

/0. dokumentácia/

Elektronická verzia práce.

/1. kalibrácia kamery - vnútorná/

Snímky pre vnútornú/intrinzičnú kalibráciu kamery a výsledky kalibrácií.

/2. kalibrácia kamery - vonkajšia/

Snímky pre vonkajšiu/extrinzičnú kalibráciu kamery, výsledky kalibrácií.

/3. referenčná póza/

Snímky pre stanovenie referenčnej pózy objektu a získané výsledky.

/4. detekcia objektu/

Snímky objektov, ich 3D modely, vzory a výsledky stanovenia pozície.

/5. projekt/

Projekt s vytvorenou aplikáciou a potrebné knižnice.

/6. aplikácia/

Výsledná aplikácia.