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

OPTICKÉ ZNAČKY PRO LOKALIZACI V MOBILNÍ ROBOTICE
LOCALIZATION WITH OPTICAL MARKERS IN MOBILE ROBOTICS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — Filip Vítek
AUTHOR

VEDOUCÍ PRÁCE — Ing. Petr Gábrlík
SUPERVISOR

BRNO 2020

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Filip Vítek ***ID:*** 203540

Ročník: 3 ***Akademický rok:*** 2019/20

NÁZEV TÉMATU:

Optické značky pro lokalizaci v mobilní robotice

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je seznámit se s dostupnými implementacemi lokalizačních systémů založených na vizuálních značkách (fiducial markers/tags) a vybrat jedno řešení, jehož vlastnosti budou podrobně prověřeny.

1. Proveďte rešerši na téma optických lokalizačních systémů využívající binární značky (např. ARTag, AprilTag).
2. Vyberte vhodné řešení pro lokalizaci v mobilní robotice, seznamte se s principem funkce a zprovozněte jej.
3. Vymyslete metodiku na komplexní prověření vlastností lokalizace, zejména na přesnost translace a rotace, chyby detekce, rozsah použití atd. Proveďte měření při různých konfiguracích systému.
4. Naměřená data přehledně zpracujte a výsledky stručně interpretujte.
5. Navrhněte pravidla pro efektivní pokrytí místnosti značkami za účelem lokalizace mobilního robotu.

DOPORUČENÁ LITERATURA:

ARTag, AprilTag and CALTag Fiducial Marker Systems: Comparison in a Presence of Partial Marker Occlusion and Rotation. Proceedings of the 14th International Conference on Informatics in Control, Automation and Robotics. SCITEPRESS - Science and Technology Publications, 2017, 2017, (Volume 2), 182-191. DOI: 10.5220/0006478901820191.

Termín zadání: 3.2.2020 ***Termín odevzdání:*** 8.6.2020

Vedoucí práce: Ing. Petr Gábrlík

doc. Ing. Václav Jirsík, CSc.
předseda rady studijního programu

Abstrakt

Mezi problémy mobilní robotiky patří například způsob navigace a určování polohy mobilního robota. Jednou z mnoha možností je použití optických značek, kterými se zabývá tato práce. V první části práce je provedena rešerše na různé druhy implementací optických značek a jejím výsledkem je zvolení knihovny ArUco. Pod open source licencí je dostupná programová knihovna pro detekce těchto značek, která byla použita pro měření. V druhé části práce proběhla tvorba a úprava programového vybavení a příprava či konstrukce dalších nezbytných pomůcek. Následně proběhlo měření, které bylo zaměřeno převážně na přesnost a dále také na úspěšnost detekce. Výsledky si je možné prohlédnout ve zpracovaných grafech a shrnuté v závěru práce.

Klíčová slova

ArUco, Knihovna ArUco, Optické značky, Lokalizace pomocí optických značek, Přesnost lokalizace

Abstract

The problems of mobile robotics includes, for example, method of navigation or positioning of mobile robot. This thesis deals with one of the many posibilities, which is the use of optical markers. The first part of this thesis is recherche about implementation of optical markers and the result is slection of the ArUco library. Under the open source license is available program library for the detection of these markers, which was used for measurement. The second part of thesis is concerning with making and adjusting program equipment and preparation or construction others necessary utilities. Then the mesurement was made, which was mainly focused on accuracy and success of detection. The results may be viewed in graphs and summarized in coclusion.

Keywords

ArUco, ArUco library, Optical markers, Localization with optical markers, Accuracy of localization

Bibliografická citace:

VÍTEK, Filip. *Optické značky pro lokalizaci v mobilní robotice.* Brno, 2020. Dostupné také z: https://www.vutbr.cz/studenti/zav-prace/detail/127080. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Petr Gábrlík.

Prohlášení

„Prohlašuji, že svou bakalářskou práci na téma Optické značky pro lokalizaci v mobilní robotice jsem vypracoval samostatně pod vedením vedoucí/ho bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si 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.

V Brně dne: **7. června 2020** ……

podpis autora

Poděkování

Tímto bych chtěl poděkovat všem, kteří mi byli nápomocni při vytváření bakalářské práce, především bych však chtěl poděkovat vedoucímu mé práce Ing. Petru Gábrlíkovi za odbornou a pedagogickou pomoc a další cenné rady.

V Brně dne: **7. června 2020** ……

podpis autora

Obsah

Seznam symbolů a zkratek

Zkratky:

FEKT	-	Fakulta elektrotechniky a komunikačních technologií
VUT	-	Vysoké učení technické v Brně
AR	-	Augmented Reality – Rozšířená realita
3D	-	Three-Dimensional – Trojrozměrné
2D	-	Two-Dimensional – Dvojrozměrné
ID	-	Identification – Identifikace
USB	-	Universal Serial Bus – Univerzální Sériová Sběrnice
PC	-	Personal Computer – Počítač

Symboly:

R	-	Rotační matice	[-]
R_X	-	Rotační matice osy X	[-]
R_Y	-	Rotační matice osy Y	[-]
R_Z	-	Rotační matice osy Z	[-]
R_{XX}	-	Prvek xx rotační matice R	[-]
ψ	-	Úhel rotace osy X	[rad]
θ	-	Úhel rotace osy Y	[rad]
ϕ	-	Úhel rotace osy Z	[rad]
α	-	Převedený úhel ve stupních	[°]
β	-	Převáděný úhel v radiánech	[rad]
π	-	Ludolfovo číslo	[-]
S	-	Obsah čtyřúhelníku	[pix]
e	-	Délka první úhlopříčky	[pix]
f	-	Délka druhé úhlopříčky	[pix]
φ	-	Úhel který úhlopříčky svírají	[°]
$\vec{u}$	-	Směrový vektor první úhlopříčky	[-]
$\vec{v}$	-	Směrový vektor druhé úhlopříčky	[-]
x	-	Souřadnice osy X rohu značky	[-]
y	-	Souřadnice osy Y rohu značky	[-]

Seznam obrázků

1 ÚVOD

Binární optické značky slouží k určení polohy mobilních robotů, rozšířenou realitu nebo také k přenosu informace a jsou navrženy tak, aby usnadnily své rozpoznání vzhledem k okolnímu prostředí. Používají se tam, kde je zapotřebí rozpoznat objekt či polohu s vysokou spolehlivostí, jakou nám v dané situaci neumožňuje prostředí [1].

Existuje několik možných designů, ke kterým patří například ARToolkit, ARTag, AprilTag, ChiliTag a další [2]. Značky jsou většinou čtvercové, ty se skládají z rámečku, uvnitř kterého je vzor se zakódovanou binární informací (buňky reprezentující 0 nebo 1), ale existují i v kruhové podobě. Značky jsou zpravidla bitonální (černobílé), a to z důvodu snížení citlivosti na osvětlení. Tímto se vyhneme nutnosti identifikovat jednotlivé odstíny šedé a u jednotlivých pixelů tak stačí rozhodovat prahovou hodnotou [4].

Tyto výše zmíněné designy jsou určeny spíše pro určování polohy, ale existují i další, které jsou určené přímo pro přenos určité informace, ke kterým patří například DataMatrix, Maxicode nebo QRcode (obrázek 2.1). Systém Maxicode používá americké pošta, která tímto způsobem udává přepravní informace [2]. Nejznámější a nejrozšířenější je ovšem v dnešní době nejspíše QR (Quick Response), kde kromě průmyslového využití na výrobních linkách je i využití v běžném životě, kde většinou přímo člověk s použitím fotoaparátu na chytrém mobilním zařízení „naskenuje“ QR značku, která může obsahovat až stovky bytů, které software přeloží, například na webovou adresu [3]. Nevýhodou těchto designů určených pro přenos informace na rozdíl od těch určených pro získání polohy je náročnost na detekci v podobě malého úhlu natočení značky či kamery, nebo v podobě intenzity osvětlení, která musí být dostačující, další nevýhodou je možnost detekce pouze jedné značky ve stejný moment [3]. Tato práce se ale bude věnovat převážně značkám určeným pro zjištění polohy a úhlu natočení.

Důležité jsou také parametry, podle nichž můžeme určit spolehlivost daného systému. Mezi základní patří „False positive rate“ (Falešně pozitivní četnost), „False negative rate“ (Falešně negativní četnost) a „inter-marker confusion“ (vlastní záměna značky) [2]. První parametr „False positive rate“ udává četnost toho, kdy systém najde značku, i když žádná není přítomna. „False negative rate“ udává opačnou četnost, kdy systém značku nenajde, i když je přítomná a nakonec „inter-marker confusion“ znamená že systém našel značku, když je přítomná, ale zaměnil identifikační číslo značky za jinou. Dalšími parametry může být minimální velikost značky v pixelech, nebo imunita k rozmanitosti osvětlení.

2 EXISTUJÍCÍ IMPLEMENTACE OPTICKÝCH ZNAČEK

Jak již bylo zmíněno v úvodu, existují různé druhy implementací optických značek. Tato kapitola se bude věnovat popisu jednotlivých typů těchto značek, převážně pak těm, určeným k lokalizaci.

Obrázek 2.1 Data Markers [2]

2.1 Maxicode, Datamatrix, QR

Tyto tři systémy jsou využívány převážně v průmyslu, a to pro přenos informace. V úvodu bylo stručně popsáno použití těchto značek, nejsou tedy navrženy, ani využívány pro AR aplikace či určování polohy, ale budou zde krátce popsány. [2]

Protože značky nesou informaci, je zde tedy při zpracovávání softwarem kladen největší důraz na metody korekce při nesprávném přečtení některých bitů. Například DataMatrix používá standard ECC200 pro určení rozměrů značek a Reedovy–Solomonovy kódy (Reed–Solomon error correction) pro korekci chyb, kde můžou obnovit informaci, pokud je kód při čtení částečně poškozen. [2]

Všechny tři implementace mají bitonální (černobílý) vzor buněk, který redukuje určování jednotlivých pixelů pouze na určování obrysů. Tím snižujeme náročnost na vlastnosti kamery, světelné podmínky a odstraníme nutnost linearizovat signál, tedy nerozlišují se odstíny šedé. Další vlastností je obsažení přebytečné informace, která umožňuje identifikovat a opravit chyby pro zvýšení celkové úspěšnosti. Tyto korekce ovšem nejsou příliš používány v počítačovém vidění jako spíše v telekomunikacích, kde můžou zajistit integritu dat na velmi spolehlivou úroveň. [2]

Lze tedy říci, že uvedené systémy jsou užitečné pro přenos informace ale nehodí se k aplikacím, kde je třeba určovat polohu a to ze dvou důvodů. Za prvé nejsou k tomu určeny a neposkytují dostatečný počet bodů ke zjištění 3D polohy. Za druhé vyžadují velké množství prostoru, což omezuje vzdálenost pro jakou se dají použít. [2]

Při návrhu značek pro lokalizaci zde informace buď není obsažená vůbec, nebo je jen pro rozlišení jednotlivých značek. Je to pro to, že čím méně informací je obsaženo, tím menší velikost v pixelech je vyžadována a na tím větší vzdálenost může být značka rozpoznána. [2]

2.2 ARToolkit

Tato implementace se poměrně liší od ostatních. Černý čtvercový rámeček je stejný jako u většiny druhů značek, ovšem interiér již není poskládán z čtvercových buněk se zakódovanou binární informací, ale nachází se zde jedinečný znak (obrázek 2.2). [2]

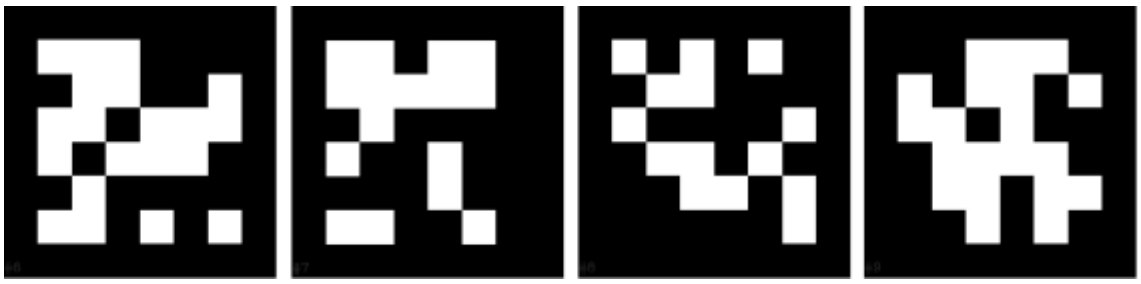
Obrázek 2.2 Příklad ARToolkit značek [5]

Rámeček je používán k vypočtení homografie na definování vzorkovací mřížky pro znak uvnitř rámečku, který je poté navzorkován do vektoru o délce 256 (nebo 1024), který je korelací porovnáván s knihovnou již známých značek. Výstup ARToolkit je takzvaný „confidence factor“ (Faktor spolehlivosti) což je výsledek normalizovaného vektorového součinu mezi snímaným 16x16 (nebo 32x32) vektorem a uloženým prototypem. Je-li značka detekována nebo není je rozhodnuto prahovou hodnotou tohoto faktoru spolehlivosti. [2]
Tato implementace je použitelná pro mnoho aplikací, má ale také několik nevýhod. Používáním korelace k identifikaci a ověření značky způsobuje vyšší „false positive rate“ a „inter-marker confusion“. Také čím větší je knihovna tím jsou znaky méně originálnější, a tím déle trvá, než se při korelaci porovnají všechny prototypy knihovně. Pro určení 4 možných rotací je pro každou značku v knihovně 12 prototypů. [2]

2.3 ARTag

Systém ARTag obsahuje 2002 značek skládajících se z čtvercové hranice uvnitř které je 6x6 čtvercová mřížka z bílých a černých buněk (obrázek 2.3). To znamená že zde máme k dispozici 36 buněk pro zakódování znaku, a když značky máme pouze černé a bílé, můžeme tedy číst 36bitové slovo. 1001 značek má černou hranici na bílém podkladu a zbylých 1001 značek naopak. [2]

Tato implementace má vylepšení v identifikaci a ověřování oproti výše zmíněnému ARToolkit, a také není potřeba uchovávat prototypy jednotlivých značek. ARToolkit má také problém, že často identifikuje značky tam, kde žádné nejsou a často je zaměňuje. ARTag značky mají také černé ohraničení, ale interiér je zpracováván digitálně. [2]



Obrázek 2.3 Příklad ARTag značek [19]

2.3.1 Detekce

[2] V obrázku jsou detekovány obrysy čtyřúhelníku, které mohou patřit vnější hranici značky. Ty jsou detekovány metodou detekce hran (přechody tmavého a světlého odstínu, obrázek 4.3 vpravo), tedy že krajní pixely jsou označeny jako hrany (přechody) a jsou spojeny do segmentů, které jsou následně seskupeny do tzv. „quads“ neboli quadů. Čtyři rohy tohoto quadu jsou použity k vytvoření homografického mapování k navzorkování interiéru značky. Tato metoda vykazuje zlepšení vlastností oproti přístupu, kdy přechodové hodnoty určujeme pomocí odstínů šedi jako u ARToolkit. Díky prostorové derivaci intenzity přechodové hodnoty šedi místo prosté intenzity přechodové hodnoty šedi umožňují identifikovat značku i za horších světelných podmínek, dokonce i pokud jsou bílé části značky tmavší než ty černé, značka bude stále detekována. [2]

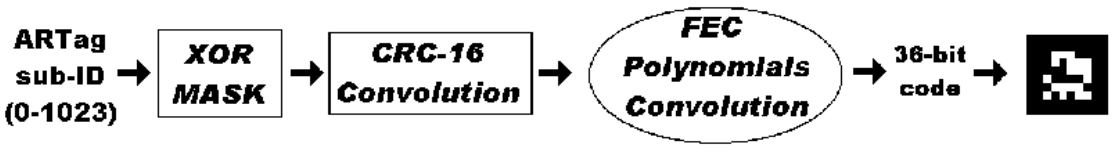
Pro další snížení falešně pozitivní četnosti ARTag hledá quady ve třech měřítkách. Extrakce čar a seskupení do quadů probíhá v originální velikosti, v poloviční velikosti a nakonec ve čtvrtinové velikosti. Toto umožní detekci i rozmazaných hranic a nemající prostorovou derivaci nad přechodovou hodnotou. [2]

2.3.2 Digitální zpracování

Jakmile je čtyřúhelníková hranice zaznamenána, vnitřní prostor je navzorkován do mřížky 6x6 vzorků a jsou přiděleny digitální symboly „0“ nebo „1“. Všechny procesy k identifikaci a ověření značky již probíhají digitálně. Čtyři 36bitové binární řady jsou získány z 2D 6x6 pole digitálních znaků, jedna pro každou z možných rotací. Pouze jedna ze čtyř binárních řad může být ověřena v dekódovacím procesu. 36bitová binární řada dekódovaná z optické značky obsahuje 10bitové ID s použitím digitálních metod. Zbylých 26 bitů je nadbytečných a používají se pro snížení falešné identifikace a poskytují unikátnost mezi čtyřmi možnými rotacemi. Cyclical redundancy chceck (CRC) a forward error correction (FEC) jsou digitální metody využívané ke zjištění, je-li 36bitový kód součástí ARTag značek a dále k extrakci jeho ID. [2]

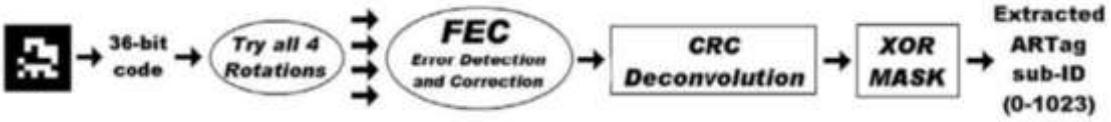
Metody používají digitální algebru zvanou GF-2 nebo Modulo-2 matematiku a zahrnuje koncept sčítání za použití logické funkce XOR, konvoluční a dekonvoluční operátory a generování polynomů což jsou prvočísla v tomto dvojkovém systému. [2]

Systém může být abstraktně popsán jako komunikační systém, kde 10bitové ID se zkouší poslat přes medium zachycení obrazu, aby bylo přijato ARTag softwarem. Vytvoření vzoru značky z ID je kódovací fáze (obrázek 2.4) a rozeznání ID z extrahovaného 36bitového kódu je dekódovací fáze (obrázek 2.5). [2]



Obrázek 2.4 kódovací fáze [2]

Blok FEC je nejsofistikovanější digitální komponenta ARTag systému a umožňuje několik chybných bitů ve vstupu 36bitového kódu, aby byl detekován a opraven. Toto zvyšuje falešně pozitivní četnost, ale zlepšuje falešně negativní četnost rozpoznáním kódu, jenž je blízko korektnímu kódu, který pravděpodobně patří ARTagu, ale s vzorkovací chybou způsobenou různými zdroji jako nesouosost detekované čtyřúhelníkové hranice, zrcadlové odrazy uvnitř vzoru, částečnou absorbcí nebo všeobecným šumem obrazu. [2]

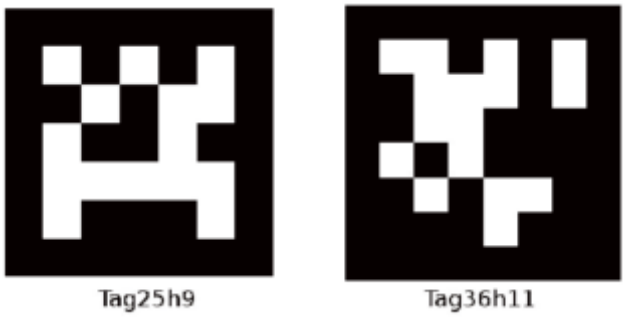


Obrázek 2.5 dekódovací fáze [2]

Dvě ID čísla jsou z ARTagu vynechána a to 682 a 1706, snižující tak celkovou velikost knihovny na 2046 značek. Důvod k vynechání je ten, že číslo 682 vede k samým nulám v 36bitovém kódu a značka by tak měla celý interiér bílý nebo černý, což by vedlo k velké chybovosti při detekci v prostředí. Dále je odebráno z knihovny celkem 44 ID ke zlepšení vlastní záměny značek, což nechává výslednou velikost knihovny na 2002 značek. [2]

2.4 AprilTag

Systém AprilTag byl vyvinut laboratoří April Robotics Laboratory na University of Michigan. Je aplikovatelný v širokém měřítku: kalibrace kamer, robotika, AR a další. Umožňuje zjištění přesné pozice, orientace a identitu značky vzhledem ke kameře. [4]



Obrázek 2.6 AprilTag Famillies [6]

Proces detekce se skládá z několika kroků: hledání lineárních segmentů, detekování přechodových hran, výpočet pozice a orientace značky, dekódování kódu. Proces hledání lineárních segmentů má podobný přístup jako ARTag, a dále je sekvence segmentů zpracována jako čtverec. Detekování čtverců používá rekurzivní čtyř úrovňovou hloubku hledání, a na každé úrovni se přidá jedna strana čtverce. V identifikační fázi se ověřuje platnost kódu uvnitř objevené značky. K dekódování vnitřního obrázku AprilTag používá lexicode systém charakterizován dvěma parametry: číslo kódového slova (vnitřní vzor) bitů a minimální vzdálenost mezi dvěma kódy. Lexicode generuje kódy pro značky, které umožňují detekování a opravu bitových chyb. AprilTag má několik skupin značek,

které se liší dvěma parametry: počet bitů k dekódování a minimální vzdálenost mezi nimi. Například Tag36h11 znamená 36bitová značka (6x6) s minimální vzdáleností 11 bitů mezi kterýmikoli dvěma kódy (obrázek 2.6). Tag 16h5 má 16bitovou značku (4x4) s minimální vzdáleností 5 bitů mezi kódy. [4]

Systém AprilTag je charakteristický zvýšeným počtem rozdílných kódů a zvýšeným počtem bitových chyb které mohou být detekovány a opraveny, redukování falešně pozitivní četnosti a vlastní záměny značek, redukce celkového počtu bitů ve značce a snížení velikosti značky. [4]

2.5 CALTAG

Po analyzování klasické šachovnicové kalibrace kamer a přístupu optických značek vznikl CALTag (obrázek 2.7), systém navržený speciálně pro řešení kalibrací kamer. Skládá se ze dvou částí: design značky a rozpoznávací algoritmus. Kalibrační značka je použita v navržené kalibrační mřížce, která je navenek podobná se šachovnicí. Rozložení značek v této mřížce má dvě možnosti hustoty značek. Mřížky s nejvyšší hustotou značek poskytují vysoký počet kalibračních bodů a tím je více spolehlivé a účinné na rozpoznání. Každá značka se skládá z MxN matice černých a bílých čtverečků, které jsou uzavřeny hranicí, která se skládá striktně z černých nebo bílých pixelů. Po počáteční detekci potenciálních značek jsou filtrovány a ověřovány přístupem k jejich binárním kódům. Každý nezachycený kalibrační bod je obnoven, protože CALTag systém zná předem rozložení šachovnice. Binární kód značky je ověřován kalkulací kontrolním součtem prvních P bitů a jejich porovnáním s kontrolním součtem, který je obdržen ze čtyřech možných rotací značky. [4]

Obrázek 2.7 CALTag Checkerboard [7]

2.6 ArUco

Značky jsou složeny z venkovní černé hranice a vnitřního prostoru ve kterém je zakódován binární vzor. Binární vzor je jedinečný pro identifikaci každé značky. Počet bitů ve vzoru je závislý na zvolené knihovně. Čím více bitů, tím více znaků v knihovně a tím menší šance záměny značek, ovšem více znaků znamená že je potřeba vyšší rozlišení k rozpoznání značky. [8]

2.6.1 Detekce a identifikace

Segmentace obrazu. Díky tomu že značky mají černou hranici obklopenou bílým prostorem, hranice může být nalezena segmentací. Používá se adaptivní metoda: střední hodnota intenzity m každého pixelu je spočítána za použití velikosti okna w_t. Pixel je nastaven na nulu, pokud je jeho intenzita menši než $m - c$, kde c je konstantní hodnota. Tato metoda je robustní a vykazuje dobré výsledky pro široký rozsah hodnot parametrů w_t a c. [8]

Extrakce a filtrování kontur. Je použit algoritmus pro získávání sady obrysů z obrázku detekce obrysových hran (obrázek 4.4 vpravo). Protože většina extrahovaných kontur patří irelevantním prvkům pozadí, je zapotřebí krok filtrace. Nejdříve jsou příliš malé kontury zahozeny. Poté zbývající kontury jsou aproximovány do jim nejpodobnějšího polygonu za použití Douglas and Peucker algoritmu [11]. Ty, které nejsou dobře aproximovány do konvexního čtyřúhelníku jsou vyřazeny z dalšího zpracovávání. [8]

Extrakce kódu značky. Dalším krok se skládá z analyzování vnitřní oblasti zbývajících obrysů k určení, které z nich jsou validní značky. Nejdříve je vypočítána homografická matice a výsledný obraz je zpracován prahovými hodnotami pomocí Otsuovi metody [12]. Výsledný obraz je rozdělen mřížkou na buňky, a každý prvek dostane binární hodnotu podle převahy pixelů v dané buňce. Pro každou možnou značku je nutné rozhodnout, zda patří ke knihovně značek, nebo je to jen prvek pozadí. Čtyři možné identifikátory jsou obdrženy pro každého kandidáta na značku odpovídajícím čtyřem možným rotacím. Pokud je nalezena shoda identifikátoru obdrženého značky a jednoho ze čtyř možných rotací, je značka přijata. [8]

Jemnost rohových subpixelů. Dalším krokem je určení lokace jednotlivých rohů s přesností subpixelů. K tomu metody využívá lineární regresy obrysových pixelů. Jinými slovy odhaduje linii krajů značky za použití všech krajních pixelů a spočítá jejich průsečíky. Ovšem tato metoda je nespolehlivá pro nekalibrované kamery s malými fokálními čočkami. [8]

3 KNIHOVNA ARUCO

V tomto semestrálním projektu byla jako řešení zvolena knihovna ArUco Library [13] a ke zvolení této knihovny vedlo několik důvodů. První výhodou je, že i když tato knihovna byla navržena pro ArUco, dokáže detekovat několik dalších implementací optických značek jako například AprilTag, ARTag, ARToolKit+ nebo ChiliTags. Autoři této knihovny také uvádějí, že detekce je velmi rychlá, dokonce rychlejší než ostatní knihovny pro detekci optických značek. A nakonec knihovna je také multiplatformní, to znamená že je možné ji zprovoznit na několika různých operačních systémech (Windows, Linux, Mac OS, Android).

Dobré je zde také zmínit referenční systém značek. Osy mají počátek přesně uprostřed značky, jak je možné vidět na obrázku 4.2. Ve výchozí poloze značky osa y směřuje nahoru, osa x směrem doprava a osa z která není na výše zmíněném obrázku zobrazena směřuje kolmo ke značce směrem ke kameře. Parametr s na obrázku udává velikost značky.

Obrázek 3.1 ArUco markers [19]

3.1 Instalace knihovny

Tato knihovna má open source licenci, a je možné si ji tedy volně stáhnout [19], v této práci je použita verze 3.1.5, která byla v době začátku práce nejnovější. Vzhledem k tomu že tato knihovna používá OpenCV [14], je nutné si ji nejdříve nainstalovat, respektive stáhnout. OpenCV je také pod open source licencí, a slouží jako knihovna pro manipulaci s obrazem. Po její instalaci je dalším krokem stažení ArUco library, kterou je poté nutno sestavit. To je možné provést z příkazové řádky, ale v systému Windows, který je v této práci používán, je nutné doinstalovat program CMake [15] sloužící k překladu programů. Pro příklad v příkazové řádce ve složce s knihovnou ArUco se knihovna zkompiluje pomocí příkazů:

```
mkdir build; cd  build ; cmake .. -DOpenCV_DIR=<pathTo-OpenCVConfig.cmake>;
```

kdy se vytvoří složka *build,* a do které se knihovna příkazem *cmake* zkompiluje. Parametr příkazu *cmake* je cesta k souboru knihovny opencv *OpenCVConfig.cmake*.

3.2 Popis a použití programů knihovny

Součástí této knihovny jsou také programy a ukázky jejího použití. V této kapitole bude krátce zmíněn obsah či použití těchto programů.

Ve složce *utils* můžeme najít základní programy knihovny, jako například *aruco_print_marker*, který vytvoří značku a uloží ji jako soubor jpg. Pro použití tohoto programu musíme zadat jako parametry příkazové řádky nejdříve ID značky kterou chceme tisknout a poté název, jak se bude soubor, do kterého bude značka uložena, jmenovat. Dále jsou k dispozici i možnosti, jako například velikost bitu v pixelech (výchozí nastavení je 50), druh značky (ArUco, ARTag…) ze kterého ji chceme tisknout (výchozí je ARUCO_MIP_36h12), a nebo také můžeme přidat bílou hranici kolem značky.

Dalším dostupným programem je *aruco_simple*, kde již může být detekována optická značka z obrazu. Vstupem tohoto programu jako první parametr příkazové řádky je videosoubor, ve kterém bude probíhat detekce značky. Chceme-li určit také pozici kamery vůči značce, musí být poskytnuty kalibrační parametry kamery v souboru s příponou yml, jehož název přidáme jako parametr příkazové řádky, a dalším parametrem nutným poskytnout v příkazové řádce je velikost značky (délka hrany) v metrech. Pokud chceme pracovat s konkrétní implementací značek, je třeba poskytnout její název, jinak program pracuje se všemi dostupnými implementacemi zároveň.

Hlavním programem nacházejícím se v této knihovně je *aruco_test*, ve kterém probíhá detekce v živém obraze. Jako první poskytnutý parametr je index kamery například *live:1* nebo *live:0*. Chceme-li určit polohu značky vůči kameře, tak musí být stejně jako v předchozím případě poskytnuty kalibrační parametry kamery a velikost detekované značky v metrech. Taktéž je možné zvolit druh značek který bude detekován předáván názvem implementace do příkazové řádky. Při spuštění programu se následně otevřou tři nová okna. První z oken je konzole, ve které se vypisují čísla nalezených značek a také jejich pozice. Dalším otevřeným oknem je obraz detekování hran (obrázek 4.4 vpravo), a poslední se otevře okno s obrazem z kamery (obrázek 4.5). V tomto okně se v levém horním rohu zobrazuje počítadlo snímků za vteřinu kamery, a hned pod ním je rozbalovací menu s popisem několika funkcí, které program poskytuje. Pomocí kláves je možné například program pozastavit či uložit aktuální snímek do souboru, a také jde otevřít další okno možností, kde jde nastavit různé parametry. Pro příklad je možné zvolit druh značek jaký má být detekován, minimální velikost detekované značky nebo také četnost chyby, kde tímto se zvyšujícím se parametrem se zvyšuje pravděpodobnost falešně pozitivní četnosti, ale snižuje se falešně negativní četnost. Do obrazu, pokud je detekována značka, je také vykreslována krychle s osami a ID značky.

Za zmínku také stojí, že knihovna umožňuje více způsobů detekcí, což je jeden z nastavitelných parametrů výše zmíněného programu. U prvního způsobu v knihovně označeném DM_Normal se jedná o běžný způsob prahování zatímco druhý způsob značený DM_Fast používá tzv. „Global thresholding“.

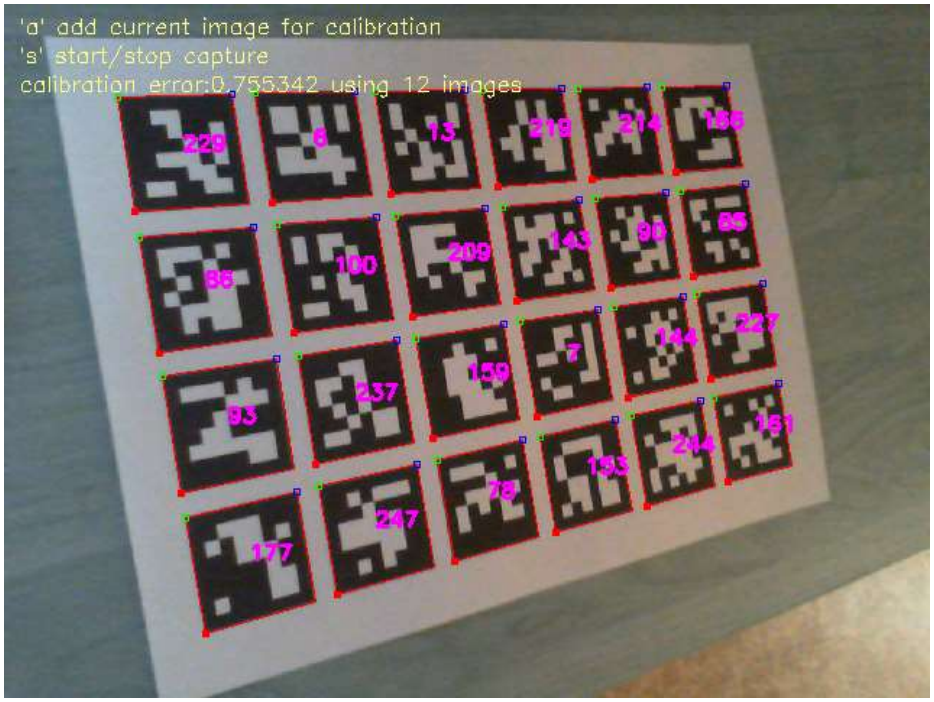
Další složkou obsahující programy je *Utils_calibration*, ve které je například program *aruco_calibration_fromimages*, která slouží pro kalibraci kamery ze snímků poskytnutých programu. Kalibrace probíhá pomocí tzv. „Markermapy“, což je soubor několika značek uspořádaných do dvourozměrného pole (obrázek 3.2), který se pomocí kamery nafotí z různých úhlů, a poté jsou tyto snímky poskytnuty programu zároveň s hodnotou velikosti jedné značky. Kamera je poté zkalibrována a informace jsou uloženy do souboru, jehož jméno je taktéž poskytnuto jako parametr programu.

Obrázek 3.2 MarkerMap [13]

Dalším zde dostupným programem je *aruco_calibration*, kde probíhá kalibrace kamery živě. Vstupem je opět index kamery pro živý obraz, dále název souboru, ve kterém budou uložena kalibrační data kamery, a nakonec velikost značky na kalibrační mapě (obrázek 3.2). Při spuštění programu se spustí okno se živým vstupem z kamery (obrázek 4.1), na kterém jsou detekovány značky, na kterých jsou vykreslovány rámečky a jejich ID a pomocí kláves se dají snímky přidávat do kalibrace. Pro úspěšnou kalibraci je potřeba zachytit alespoň 4 snímky, ovšem pro kvalitnější kalibraci je vhodnější použít snímků více.

4 PŘÍPRAVA VYBAVENÍ PRO MĚŘENÍ

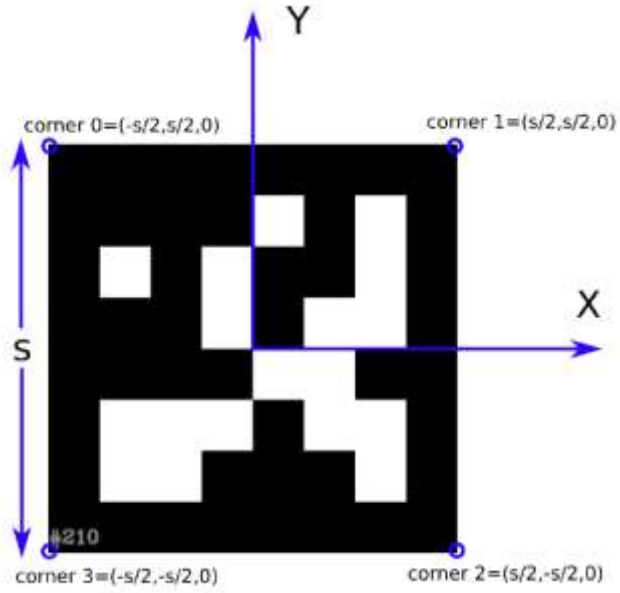
Před samotným testováním vybrané knihovny je třeba provést ještě přípravu programového vybavení a pomůcek, jako je například kalibrace kamery, nebo úprava dostupných programů pro potřeby měření. Tyto kroky budou dále popsány v této kapitole.



Obrázek 4.1 Kalibrační mapa

4.1 Kalibrace

Pro spolehlivé určení polohy kamery vůči značce je důležité znát také parametry kamery jako je ohnisková vzdálenost nebo zkreslení. Parametry je možné získat pomocí kalibrace, která je v tomto případě provedena pomocí programu *aruco_calibration*. Ke kalibraci je zde použita kalibrační mapa značek (obrázek 3.2) vytisknutá na papíře formátu A4, kde velikost strany jedné značky je 3,7 cm, což je jedna z hodnot předaná jako parametr programu (viz kapitola 3.2). Za pomocí kamery bylo pořízeno z několika různých úhlů 20 kalibračních snímků, díky kterým byli zjištěny výše zmíněné parametry použité kamery.



Obrázek 4.2 Referenční systém [13]

4.2 Úprava programu

Pro program k testování byl jako základ zvolen aruco_test dostupný v knihovně, který byl následně upraven, doplněn a rozšířen o další funkce potřebné k zjišťování parametrů a hodnot měření. Tyto úpravy jsou dále popsány v této kapitole.

4.2.1 Převod úhlů

Výstupem programu jsou hodnoty translace pro osy x, y a z a rotace pro tytéž osy. Hodnoty translace jsou udány v metrech, ovšem hodnoty rotace jsou v radiánech a v nevhodné soustavě, která se nehodí pro odečítání hodnot člověkem i při převedení na stupně. Proto je vhodné převést tyto hodnoty na Eulerovy úhly [16] a k tomu je potřeba získat rotační matici (vzorec 4.1). Tu získáme pomocí funkce knihovny OpenCV s názvem Rodrigues [14], jelikož získané hodnoty rotace jsou v knihovně OpenCV obdrženy z Rodriguesova vzorce [17]. Do této funkce pak stačí vložit jako první parametr získaný vektor s hodnotami rotace, a druhým parametrem bude prázdná matice 3x3 do které se uloží výsledná rotační matice.

$$R = \begin{pmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{pmatrix} \tag{4.1}$$

Pro získání Eulerových úhlů je třeba si nejprve uvědomit že existují rotační matice pro každou ze tří os:

$$R_x(\psi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{pmatrix} \tag{4.2}$$

$$R_y(\theta) = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \tag{4.3}$$

$$R_z(\phi) = \begin{pmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{4.4}$$

Pak výsledná rotační matice pro vyjádření rotace všech tří os je dána:

$$R = R_z(\phi) \cdot R_y(\theta) \cdot R_x(\psi) \tag{4.5}$$

Po dosazení matic jednotlivých os je výsledná matice rovna:

$$R = \begin{pmatrix} \cos\theta \cdot \cos\phi & \sin\psi \cdot \sin\theta \cdot \cos\phi - \cos\psi \cdot \sin\phi & \cos\psi \cdot \sin\theta \cdot \cos\phi + \sin\psi \cdot \sin\phi \\ \cos\theta \cdot \sin\phi & \sin\psi \cdot \sin\theta \cdot \sin\phi + \cos\psi \cdot \sin\phi & \cos\psi \cdot \sin\theta \cdot \sin\phi - \sin\psi \cdot \cos\phi \\ -\sin\theta & \sin\psi \cdot \cos\theta & \cos\psi \cdot \cos\theta \end{pmatrix}$$

(4.6)

Z této matice pak můžeme vyjádřit úhly pro jednotlivé osy. Pro osu θ platí:

$$R_{31} = -\sin\theta \tag{4.7}$$

Z rovnice pak vyjádříme θ v radiánech:

$$\theta = -\sin^{-1}(R_{31}) \tag{4.8}$$

Pro úhel ψ platí:

$$\frac{R_{32}}{R_{33}} = \tan(\psi) \tag{4.9}$$

Vyjádřením úhlu pak dostaneme ψ v radiánech:

$$\psi = \arctan 2(R_{32}, R_{33}) \tag{4.10}$$

V případě že platí:

$$\cos(\theta) < 0 \tag{4.11}$$

Pak rovnice 4.10 odpovídá ψ v radiánech:

$$\psi = \arctan 2(-R_{32}, -R_{33}) \tag{4.12}$$

A pro poslední úhel ϕ platí:

$$\frac{R_{21}}{R_{11}} = \tan\phi \tag{4.13}$$

Vyjádření úhlu v radiánech:

$$\phi = \arctan 2(-R_{21}, -R_{11}) \tag{4.14}$$

Protože tato osa míří opačným směrem, než jak je myšleno v programu, je tedy zapotřebí pro zachování pravidla pravé ruky nutno osu invertovat (úhel zůstává v radiánech):

$$\phi = -\arctan 2(-R_{21}, -R_{11}) \tag{4.15}$$

Tímto jsou získány za pomocí vzorců 4.2 až 4.15 [18] všechny potřebné rovnice (4.8, 4.12 a 4.15) ke zjištění Eulerových úhlů z rotační matice.
Vše je následně nutné implementovat do programu. Toho je dosaženo následujícím kódem:

```
cv::Mat R(3, 3, CV_64F);
Vec3f Euler(0, 0, 0);
Vec3f Euler_deg(0, 0, 0);

cv::Rodrigues(TheMarkers[i].Rvec, R);
Euler = ToEulerAngles(R);

Euler_deg[0] = Euler[0] * (180 / CV_PI);
Euler_deg[1] = Euler[1] * (180 / CV_PI);
Euler_deg[2] = Euler[2] * (180 / CV_PI);
```

Kde je nejdříve deklarována matice a dále dva vektory pro uložení Eulerových úhlů. Do již dříve zmíněné funkce Rodrigues jsou jako parametry zadány vektor rotačních úhlů a prázdná matice, do které se uloží hodnoty rotační matice. Následně je volána funkce, ve které jsou implementovány rovnice získané v předchozí podkapitole a jejíž kód je možné vidět zde:

```
Vec3f ToEulerAngles(Mat & R)
	{
float x, y, z;

x = atan2(-R.at<float>(2, 1), -R.at<float>(2, 2));
y = -asin(-R.at<float>(2, 0));
z = -atan2(R.at<float>(1, 0), R.at<float>(0, 0));

return Vec3f(x, y, z);
}
```

Výstupem funkce jsou Eulerovy úhly v radiánech, a je proto ještě vhodné je převést do stupňů, což je provedeno na posledních 3 řádcích v horní ukázce kódu pomocí vzorce:

$$\alpha = \beta \cdot \frac{180}{\pi} \tag{4.16}$$

Kde α je úhel ve stupních a β úhel v radiánech.

4.2.2 Obsah značky v obraze

Aby bylo možné zjistit jakou část obrazu zabírá značka, je nutné spočítat obsah obrazu a značky v pixelech. Z údajů získaných kalibrací je zjištěno rozlišení kamery 640x480 pixelů. Díky funkcím knihovny lze také snadno zjistit souřadnice jednotlivých rohů značky (rohový pixel). Z těchto údajů je možné vypočítat požadované obsahy.

I když je značka fyzicky čtvercová, při různých natočeních kamery nebo značky se tak v obraze jevit nebude, a proto je nutné pro výpočet použít obecný vzorec (4.17) pro obsah čtyřúhelníku [20].

$$S = \frac{1}{2} \cdot e \cdot f \cdot \sin\varphi \tag{4.17}$$

Kde *e* a *f* jsou délky úhlopříček a φ libovolný úhel který svírají.

Chceme-li tedy znát obsah, je nutné zjistit délky úhlopříček a úhel který svírají. Jak je popsáno výše, známe souřadnice [x,y] jednotlivých rohů značky, pro zjištění úhlopříčky tedy stačí spočítat vzdálenost mezi dvěma body (protější rohy značky) v souřadném systému.

$$e = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{4.18}$$

Kde (x_1,y_1) a (x_2,y_2) jsou souřadnice navzájem protějších rohů značky.

Nyní jsou známy délky úhlopříček a zbývá zjistit jaký úhel svírají. To lze spočítat pomocí vektorů dle následujícího vzorce.

$$\cos\varphi = \frac{|\vec{u} \cdot \vec{v}|}{|\vec{u}| \cdot |\vec{v}|} = \frac{|u_1 \cdot v_1 + u_2 \cdot v_2|}{\sqrt{u_1^2 + u_2^2} \cdot \sqrt{v_1^2 + v_2^2}} \tag{4.19}$$

Kde *u* a *v* jsou směrové vektory úhlopříček které svírají úhel *φ*. Po vyjádření úhlu z tohoto vzorce:

$$\varphi = \arccos\frac{|u_1 \cdot v_1 + u_2 \cdot v_2|}{\sqrt{u_1^2 + u_2^2} \cdot \sqrt{v_1^2 + v_2^2}} \tag{4.20}$$

Pro jednotlivé vektory $\vec{u} = (u_1, u_2)$ platí

$$u_1 = x_1 - x_2 \tag{4.21}$$

$$u_2 = y_1 - y_2 \tag{4.22}$$

Kde (x_1,y_1) a (x_2,y_2) jsou souřadnice navzájem protějších rohů značky. Obdobně to platí i pro vektory $\vec{v} = (v_1, v_2)$.

Nyní jsou k dispozici všechny potřebné rovnice a zbývá je tudíž implementovat do kódu. Rovnice jsou shrnuty do jedné funkce pro výpočet obsahu značky, která má jako parametry vektor obsahující mimo jiné souřadnice rohů nalezené značky, a proměnnou s číslem cyklu ve kterém je funkce volána. Volání probíhá v cyklu pro případ nalezení více značek, ovšem vyhledávání bude při testování probíhat pouze pro jednu značku, není to tudíž nutné. Návratová hodnota funkce je vypočítaný obsah a celá její implementace je uvedena zde:

```
double ObsahMarker(std::vector<aruco::Marker> &TheMarker, int i){

	double corner1x, corner1y, corner2x, corner2y, diagonal1, diagonal2,
uhel_diag1, Obsah_Marker;

	corner1x = (TheMarker[i][0].x) - (TheMarker[i][2].x);
	corner1y = (TheMarker[i][0].y) - (TheMarker[i][2].y);
	corner2x = (TheMarker[i][3].x) - (TheMarker[i][1].x);
	corner2y = (TheMarker[i][3].y) - (TheMarker[i][1].y);
	diagonal1 = sqrt((corner1x) * (corner1x) + (corner1y) * (corner1y));
	diagonal2 = sqrt((corner2x) * (corner2x) + (corner2y) * (corner2y));

	uhel_diag1 = acos(abs((corner1x * corner2x) + (corner1y * corner2y)) /
(sqrt(corner1x * corner1x + corner1y * corner1y) * sqrt(corner2x * corner2x +
corner2y * corner2y)));

	Obsah_Marker = 0.5 * diagonal1 * diagonal2 * sin(uhel_diag1);

	return (Obsah_Marker);}
```

4.2.3 Ukládání dat

Při měření je nutné získaná data ukládat, toho je docíleno zapisováním měřených dat do textového souboru (obrázek 4.3). Pro každé nastavení předem zvolených hodnot translace a rotace proběhne jedno měření, přičemž se pro každé takovéto měření vytvoří samostatný textový soubor s unikátním názvem. Soubory jsou otevírány (vytvářeny) cyklicky a jejich název odpovídá aktuálně nastaveným hodnotám translace a rotace. Přesný formát názvů souborů je „File_T_x_y_z_R_x_y_z.txt“ kde místo názvů os X, Y a Z jsou přímo číselně vyjádřené jejich hodnoty translace a rotace (v názvu rozlišené písmeny R a T).

File_T_0_0_30_R_+0_+30_+0.txt – Poznámkový blok

Soubor Úpravy Formát Zobrazení Nápověda

Radek	Tx[m]	Ty[m]	Tz[m]	Rx[°]	Ry[°]	Rz[°]	S[pix]	t[ms]	Nalez
1	0.003	0.002	0.297	0.4	30.2	0.2	67296	67	1
2	0.003	0.002	0.297	0.4	30.2	0.2	67296	128	2
3	0.003	0.002	0.296	0.5	30.2	0.3	67350	193	3
4	0.003	0.002	0.296	0.5	30.2	0.3	67350	260	4
5	0.003	0.002	0.297	0.5	30.2	0.3	67319	326	5
6	0.003	0.002	0.296	0.5	30.2	0.2	67354	462	6
7	0.003	0.002	0.296	0.5	30.2	0.2	67354	528	7
8	0.003	0.002	0.297	0.5	30.2	0.2	67322	591	8
9	0.003	0.002	0.297	0.5	30.2	0.2	67322	656	9
10	0.003	0.002	0.297	0.5	30.2	0.2	67300	720	10
11	0.003	0.002	0.297	0.5	30.2	0.2	67308	865	11
12	0.003	0.002	0.297	0.5	30.2	0.2	67308	928	12
13	0.003	0.002	0.296	0.5	30.3	0.2	67326	992	13
14	0.003	0.002	0.296	0.5	30.3	0.2	67326	1055	14
15	0.003	0.002	0.297	0.5	30.2	0.2	67300	1129	15
16	0.003	0.002	0.297	0.5	30.1	0.3	67287	1265	16
17	0.003	0.002	0.297	0.5	30.1	0.3	67287	1332	17
18	0.003	0.002	0.296	0.5	30.2	0.2	67342	1391	18
19	0.003	0.002	0.297	0.5	30.2	0.3	67303	1521	19
20	0.003	0.002	0.297	0.5	30.2	0.3	67303	1588	20
21	0.003	0.002	0.297	0.5	30.2	0.2	67304	1671	21
22	0.003	0.002	0.297	0.5	30.2	0.2	67304	1735	22

Řádek 1, sloupec 1 100 % Windows (CRLF) UTF-8

Obrázek 4.3 Soubor s daty

Při jednom měření proběhne stokrát cyklus detekce značky a každému jednomu cyklu odpovídá jeden řádek v souboru. První řádek je vytvořen ještě před měřením při vytváření souboru a obsahuje legendu k jednotlivým sloupcům. V prvním sloupci je vždy uvedeno pořadové číslo řádku, v dalších šesti sloupcích jsou hodnoty translace a rotace, následující sloupec udává obsah značky v pixelech, v předposledním sloupci je uveden uplynulý procesní čas od začátku konkrétního měření a poslední sloupec obsahuje počet nálezů značky. Počet nálezů odpovídá počtu cyklů ze sta, ve kterých byla značka detekována.

4.2.4 Automatizované měření

Jak již bylo zmíněno výše, pro každou kombinaci proměnných (translace a rotace) je potřeba vytvořit speciální název souboru. Aby nebylo nutné zadávat všechny hodnoty pro každé měření ručně, což by bylo zdlouhavé, a aby bylo možné hodnoty zadávat za běhu, byl program doplněn vhodnými funkcemi.

Jelikož hodnoty translace se v tomto měření mění jen občasně, jsou tedy zadávány ručně za běhu programu. Po stisknutí příslušného tlačítka (*x* pro změnu translace osy X, *y* pro změnu Y a *z* pro změnu Z) je uživatel vyzván, aby do konzole zadal novou hodnotu translace.

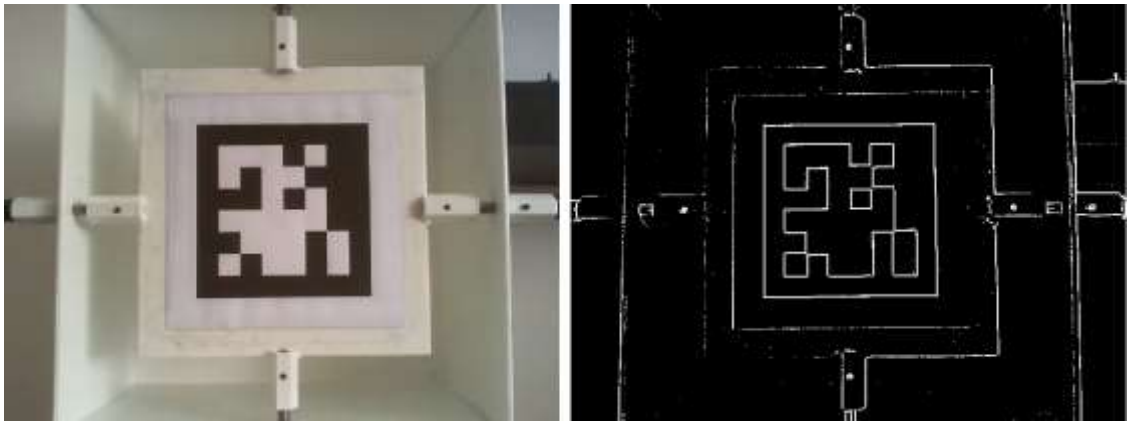
Naopak hodnoty rotace se mění v každém měření, a proto jsou doplňovány z větší části automaticky. Po stisku klávesy *k* je uživatel vyzván, aby zadal do konzole, jaká osa bude měřena. Po zadání písmene reprezentujícího osu (X, Y nebo Z) je provedeno měření pro úhel natočení 0 stupňů na všech osách. Jak bylo dříve zmíněno, pro každé měření je provedeno sto cyklů detekce. Po skončení měření neprobíhá žádná změna, dokud není opět stisknuta klávesa *k*, po jejímž zmáčknutí proběhne opět měření, ovšem tentokrát se změní úhel natočení, respektive název souboru, pro měřenou osu o 10 stupňů při měření osy X či Y a o 20 stupňů při měření osy Z. Vždy je měřena pouze, respektive mění se název jen jedné z os a ostatní zůstávají neměnné. Takto měření probíhá až po dosažení 80 stupňů pro osy X a Y, a 180 stupňů pro osu Z. Poté se hodnota rotace změní na -10 stupňů pro X a Y a pokračuje do -80 a pro osu Z se změní na -20 stupňů a pokračuje do -160. Když je po dokončení měření opět stisknuta klávesa *k*, je uživatel znovu vyzván pro zadání názvu měřené osy a celý cyklus se může opakovat.

4.2.5 Text v obraze

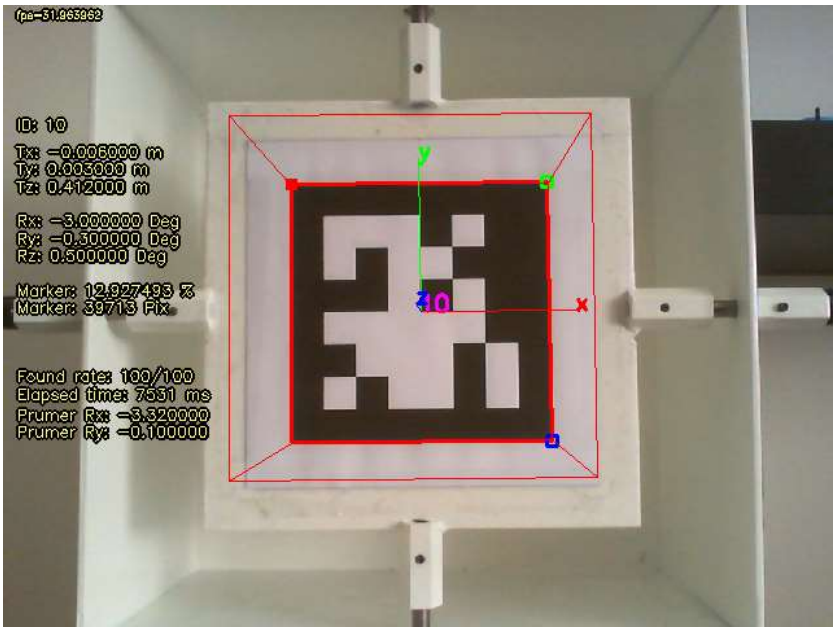
Hodnoty získané detekcí jsou původně vypisovány programem do konzole. Jelikož vypisování probíhá rychle jsou hodnoty za běhu nečitelné a uživatel tak nemá přehled o datech, pokud program nepozastaví. Proto je vhodné program doplnit tak, aby požadované hodnoty byly vypisovány do okna s živým přenosem z kamery a tím by byl zajištěn plný přehled o okamžitých hodnotách detekce. Tohoto bylo dosaženo pomocí funkce knihovny OpenCV *putText* [14]. Vstupem této funkce je název okna, do kterého bude text vkládán, řetězec znaků, který se bude vypisovat, dále souřadnice umístění textu v okně, a nakonec font pro písmo. Ukázka použití je uvedena zde:

```
putText(TheInputImageCopy,"ID:"+to_string(MarkerID),cv::Point(10,prom*180),prom*0
.75f);
```

Pomocí této funkce jsou do obrazu dle obrázku 4.5 vypisovány informace. Nejprve je přidáno ID značky, následují hodnoty translace a rotace pro všechny osy, poté je uvedena velikost značky v pixelech a následně v procentech z celkového obrazu. Mezi další uvedené informace patří počet nálezů značky ze sta cyklů detekce, uplynulý čas opět pro sto cyklů detekce, a nakonec také průměrné hodnoty pro rotace osy X a Y odpovídající průměru posledních pěti hodnot rotací X a Y.



Obrázek 4.4 Detekce



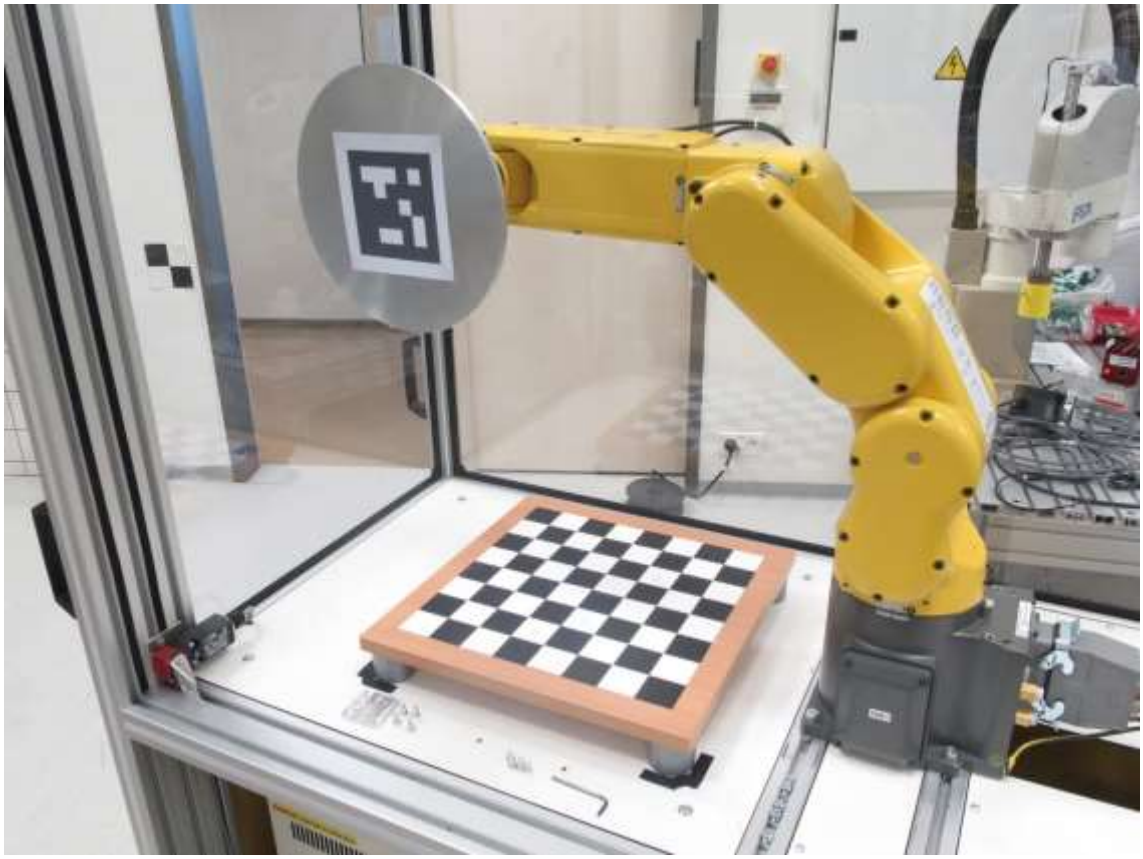
Obrázek 4.5 Nalezená značka

4.2.6 Nastavení kamery

Aby byly při měření, pokud možno, co nejvíce dodrženy konstantní podmínky, byla kamera nastavena na stálou expozici. V defaultním nastavení má kamera expozici automatickou a ta se může měnit v průběhu měření, a tak i ovlivňovat výsledky. Nastavení konstantní expozice bylo dosaženo s pomocí funkcí knihovny OpenCV. Třída TheVideoCapturer obsahuje metodu set, která má dva parametry. Prvním parametrem je makro, které udává, jaké kritérium kamery se bude měnit, a druhým parametrem je hodnota daného kritéria. Použití je uvedeno zde:

```
TheVideoCapturer.set(CV_CAP_PROP_AUTO_EXPOSURE, 0.25);
TheVideoCapturer.set(CV_CAP_PROP_EXPOSURE, -3);
```

Na prvním řádku je nejdříve přepnuta expozice z automatické na manuální a to hodnotou 0,25. V případě potřeby je opět možno přepnout expozici na automatickou a to stejným příkazem, ovšem s hodnotou 0,75. Ve druhém řádku je již nastaven konkrétní čas expozice a to hodnotou -3. Hodnoty pro nastavení se pohybují od -1 do -13 a zde nastavená hodnota byla zvolena experimentálně v závislosti na okolním osvětlení, tak aby bylo dosaženo co nejlepšího obrazu. Konkrétní čas expozice, který reprezentují jednotlivé hodnoty zadávané jako parametr, není znám a pro jeho zjištění by muselo být provedeno samostatné měření.



Obrázek 4.6 Robotický manipulátor

4.3 Přípravek pro nastavení rotací

Původním záměrem, jak nastavovat přesný úhel rotace bylo použití robotického manipulátoru Fanuc LR Mate 200id/4S (obrázek 4.6). Robot je šestiosý a výrobce udává opakovatelnost až ±0.01 mm. Značka pro měření by byla umístěna na kotouči, který by byl připevněn na robotovi, jak je patrné z obrázku. Byla by následně naprogramována sekvence kroků, v kterých by se měnila rotace tak, aby korespondovala s funkcemi programu popsanými v kapitole 4.2.4. Tlačítkem by se

pak posouvalo po jednotlivých krocích rotace a klávesou na počítači by se pro každý krok spouštělo měření. Ovšem robot byl nakonec nedostupný pro měření, a tak muselo být vytvořeno náhradní řešení.

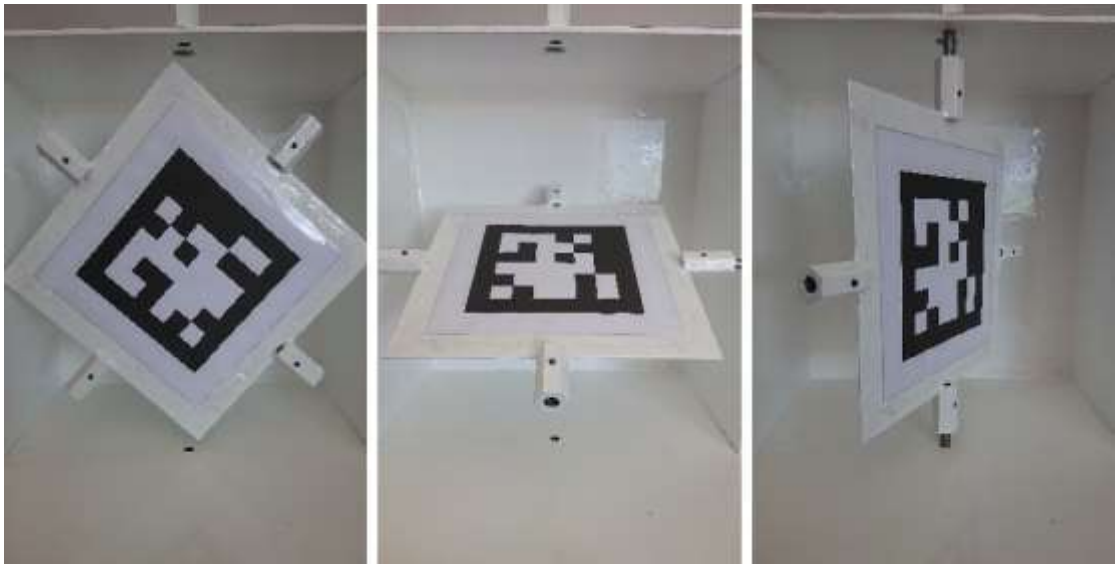
Náhradním řešením tedy bylo sestrojení přípravku pro rotaci (obrázek 4.7) s možností otáčet značku ve všech třech osách. Konstrukce přípravku umožňuje, že otáčet se může v daný okamžik vždy pouze jedna osa, což pro toto měření není nevýhodou, neboť je vždy rotováno pouze s jednou osou. Tato konstrukce ovšem také poskytuje značnou výhodu, a to že plocha na které je značka umístěna se otáčí přesně ve středu osy a tím se i značka otáčí přesně ve středu osy a nedochází tak při měření k chybě.

Obrázek 4.7 Přípravek

Celá konstrukce stojí na čtyřech nohách, na kterých jsou ze spodu umístěny šroubovací podložky, se kterými je možno podle potřeby upravovat výšku jednotlivých nohou tak, aby se celý přípravek na nerovném povrchu nehoupal. Nohy jsou upevněny k základně celého přípravku, na které stojí stěny pro umístění stupnic. Dále jsou na stěnách a na konstrukci z pěti stran umístěny šestihrany s vysoustruženým otvorem pro umístění hřídele. Stejné šestihrany jsou umístěny i na ploše, po stranách a zezadu, pro umístění značky a jsou také vybaveny závitem pro utažení hřídele, aby v otvoru neprokluzovala. Jednotlivé závity je možné utáhnout imbusovým klíčem. Při zasunutí všech hřídelí se nachází značka ve výchozí poloze, a je možné nastavit rafičku do nulové polohy. Každá osa má na hřídeli rafičku, kterou je možno povolit či utáhnout, opět pomocí imbusového klíče. Pro otáčení v jednotlivých osách je nutné vždy ponechat dvě protilehlé nebo jednu

zadní hřídel, v závislosti jakou osou chceme otáčet, zasunutou, a ostatní vysunout jak je znázorněno na obrázku 4.8. Při utažení dané hřídele v drážce a utažení rafičky na stejné hřídeli se pak při otáčení značky rafička pohybuje po přilepené stupnici a je možné odečítat úhel natočení.

Byla také stanovena nejistota přípravku. Vzhledem k tomu že rafička je zakončena do špičky, lze s její pomocí odečítat poměrně přesně a nejistota byla stanovena šířkou dílku na stupnici. Dílek byl změřen posuvným měřítkem jako 1,7 mm a délka rafičky od středu osy otáčení značky jako 142 mm. Z trojúhelníku s rameny odpovídající délce rafičky a základně odpovídající šířce dílku, pak lze snadno zjistit úhel, jaký jeho ramena svírají. Tento byl stanoven jako 0,7 stupně což je také výsledná nejistota



Obrázek 4.8 Přípravek, rotace

Obrázek 4.9 Stativ s kamerou

5 MĚŘENÍ

Po tom, co je vše připraveno, je možné přejít k samotnému měření. Způsob, jakým bylo postupováno bude podrobně popsán v následujících kapitolách včetně použitého vybavení.

5.1 Použité vybavení

Značka, s identifikačním číslem 10 a o velikosti 12x12 cm, byla připevněna na přípravek blíže popsaný v kapitole 4.3. Ke snímání obrazu je použita běžná externí webkamera s výstupem v podobě USB kabelu a připevněna na stativ (obrázek 4.9). Ohnisková vzdálenost zjištěná pomocí kalibrace kamery je 682 pixelů a její rozlišení bylo zjištěno jako 640x480 pixelů. Stativ je tvořen teleskopicky se vysouvající trojnožkou a se samotným držákem na kameru je možné manipulovat ve všech třech osách rotace, stejně tak je možné ještě jemně upravovat výšku vysouváním prostřední tyče. Vzdálenost mezi značkou a kamerou byla měřena pomocí svinovacího metru. Tento metr je označený třídou přesnosti 2 a v takovém případě se nejistota měření stanovuje na 2 dílky stupnice včetně zohlednění chyby lidského faktoru. Samotné měření pak probíhalo na počítači, ke kterému byla kamera připojena.

5.2 Popis měření

Poté co značka byla upevněna do přípravku a kamera na stativ a zároveň zapojena do PC, bylo dále nutné nastavit nulovou polohu kamery vůči značce. Tato poloha byla nastavena pomocí dat odečítaných přímo z programu. Nejdříve byla nastavena referenční poloha na přípravku dle kapitoly 4.3 a poté bylo pomocí odečítání dat z programu a dle nich natáčením kamery na stativu docíleno, pokud možno, nulové polohy, tedy polohy, kdy jsou rotace nulové. Rotaci kolem osy z bylo možno nastavit bez problémů, ovšem v nulových polohách os x a y tato data v programu značně kmitala kolem nuly. Z tohoto důvodu bylo v programu implementována filtrace, respektive průměrování posledních pěti hodnot rotací na těchto osách (zmíněno také v kapitole 4.2.5), aby bylo možné snadněji odhadnout rotaci, ve které se kamera nachází. Tímto způsobem je nastavena nula, a stiskem klávesy na PC může být odstartováno měření. Funkce měření programem jsou blíže popsány v kapitole 4.2.4. Zároveň s programem je ručně nastavována rotace značky na přípravku. Takto jsou pro danou vzdálenost proměřeny rotace všech tří os a to pro dva různé druhy detekce, zpracována ovšem byla nakonec jen jedna.

Takto bylo měření provedeno pro 5 vzdáleností. Nejmenší možná vzdálenost detekce byla určena experimentálně, stejně jako největší. Největší

měřená vzdálenost, není největší při níž byla značka ještě detekována, ale největší při níž detekce probíhala ještě v rozumných mezích a nebyla spíše jen občasná.

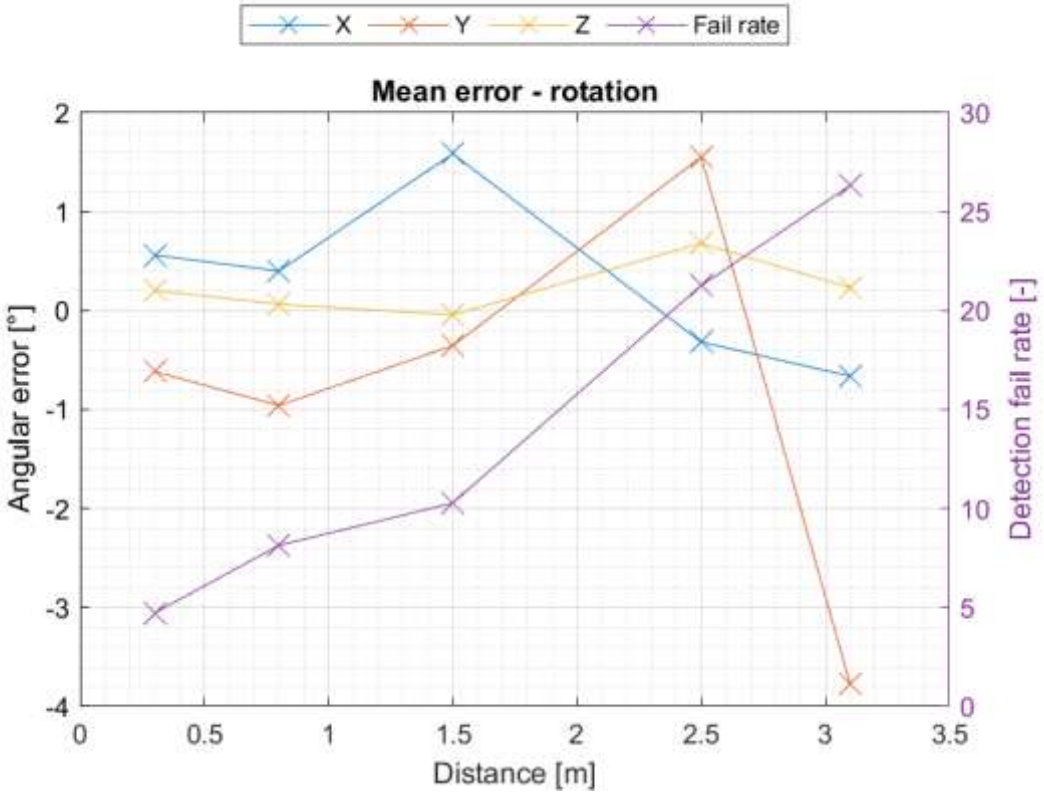
5.3 Grafické zpracování dat

Poté co byla data naměřena a uložena, je třeba je dále zpracovat do přehledné podoby, respektive do grafického znázornění. Vzhledem k objemu dat je jejich zpracování prováděno pomocí programu Matlab. Zpracovány byly například závislosti střední chyby či směrodatné odchylky translace a rotace na vzdálenosti a rotaci značky. V programu byly použity například funkce *mean* či *std*.

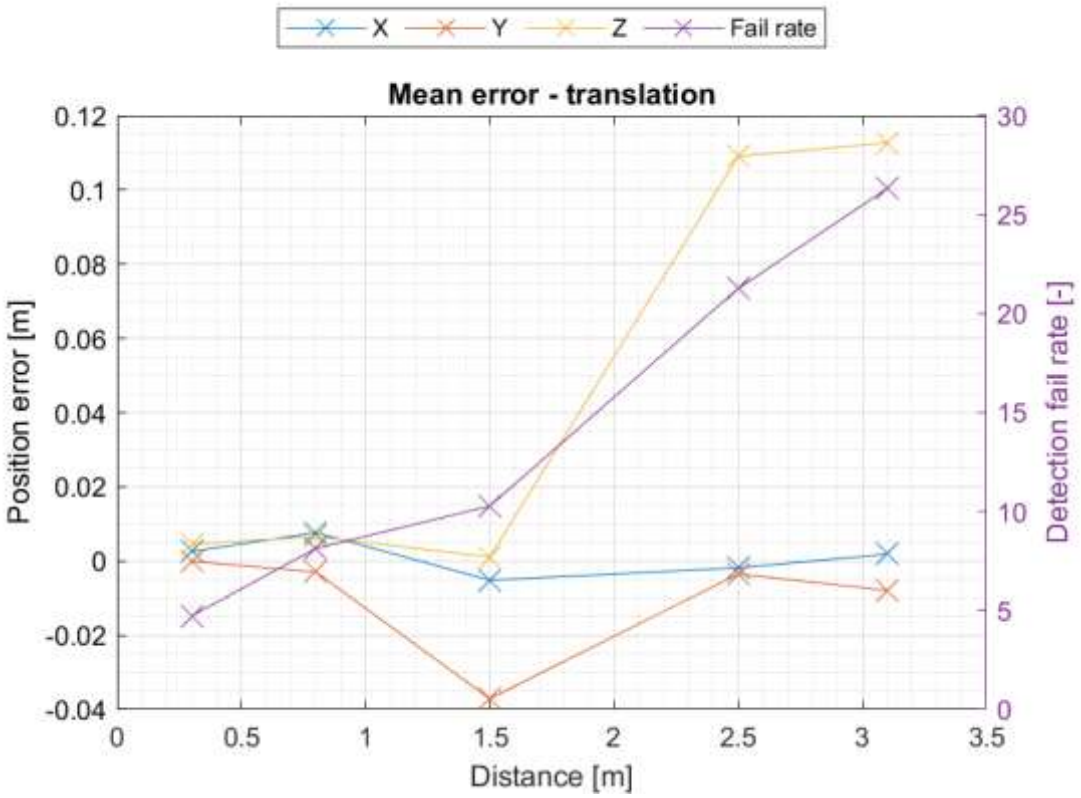
V grafy mají v horní části uvedenou legendu pro jednotlivé průběhy hodnot, pod kterou se nachází název grafu značící, co je v grafu za závislosti. U jednotlivých os je pak uveden popis, co se na dané ose nachází. Na ose X je to buď *Distance* udávající vzdálenost značky od kamery, nebo *Angle* udávající úhel natočení značky. Na ose Y se pak nachází *Position error*, tedy chyba translace či *Angular error*, tedy chyba rotace značící buď střední absolutní chybu nebo směrodatnou odchylku. Na vedlejší ose Y je také *Detection fail rate* udávající četnost chyby detekce, také označované jako falešně negativní četnost.

5.3.1 Závislosti na vzdálenosti

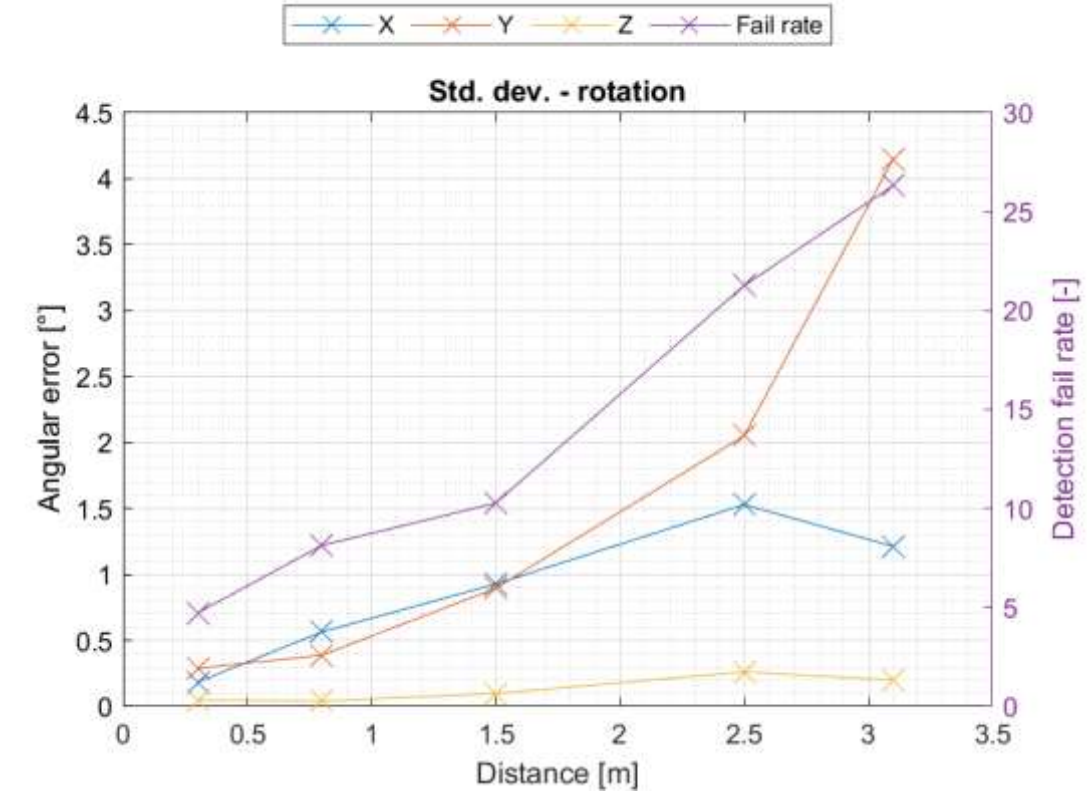
Ve zde uvedených grafech jsou závislosti střední absolutní chyby a směrodatné odchylky na vzdálenostech značky od kamery, ve kterých bylo měření prováděno. Ve všech grafech je také na vedlejší ose Y vynesena závislost chyby detekce v závislosti na vzdálenosti.



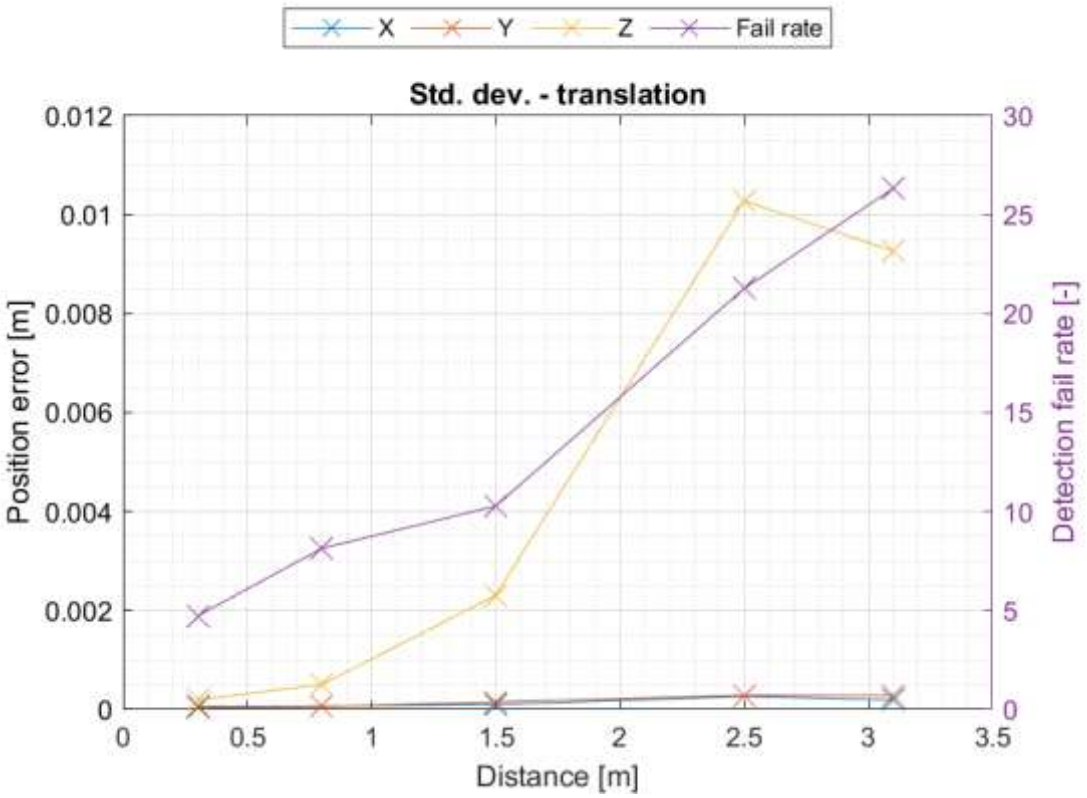
Obrázek 5.1 Závislost střední chyby rotace na vzdálenosti



Obrázek 5.2 Závislost střední chyby translace na vzdálenosti



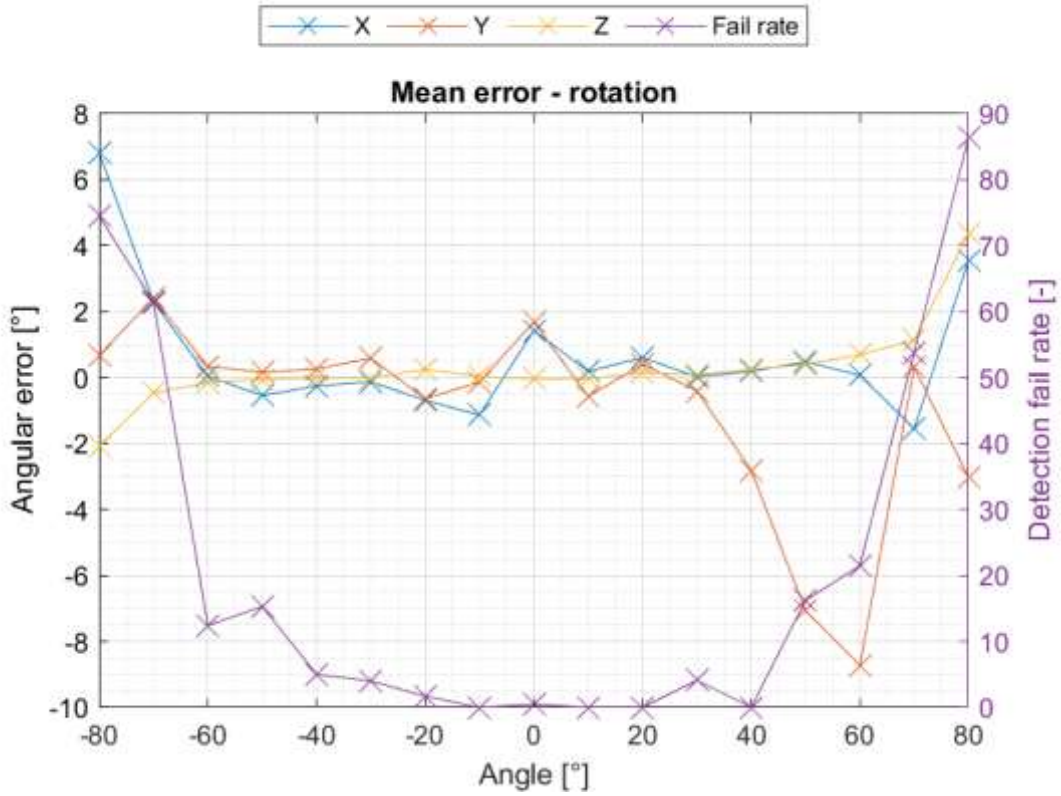
Obrázek 5.3 Závislost směrodatné odchylky rotace na vzdálenosti



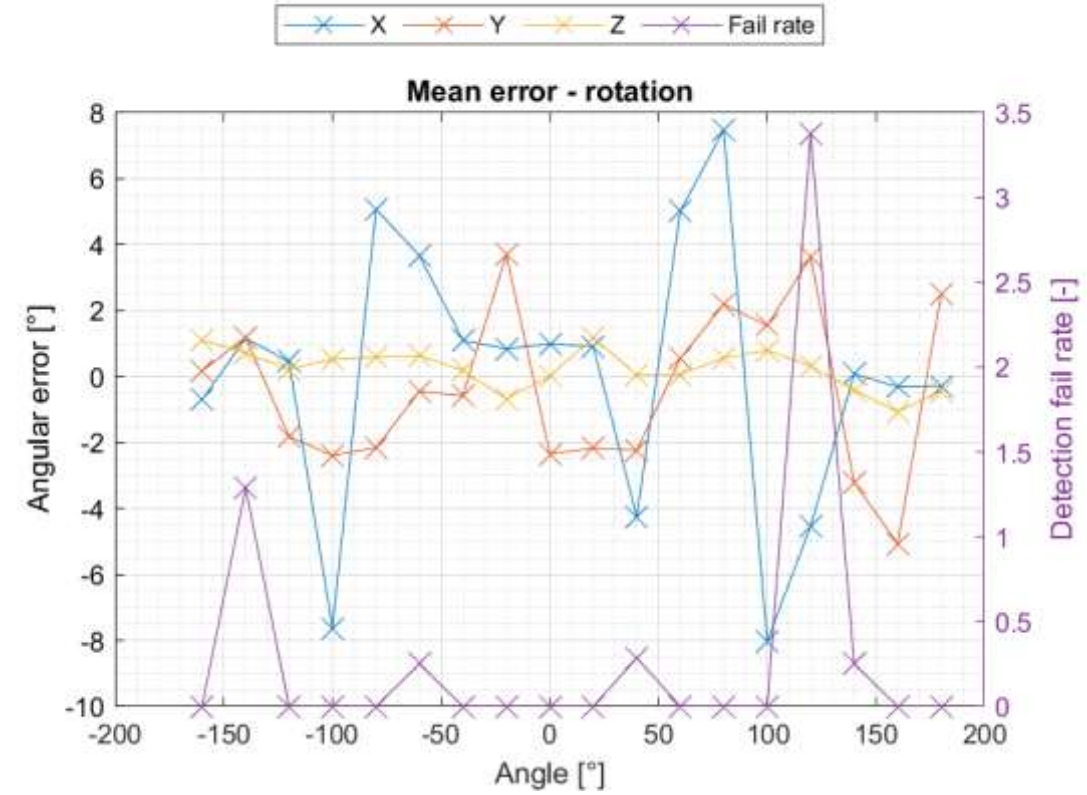
Obrázek 5.4 Závislost směrodatné odchylky translace na vzdálenosti

5.3.2 Závislosti na rotaci

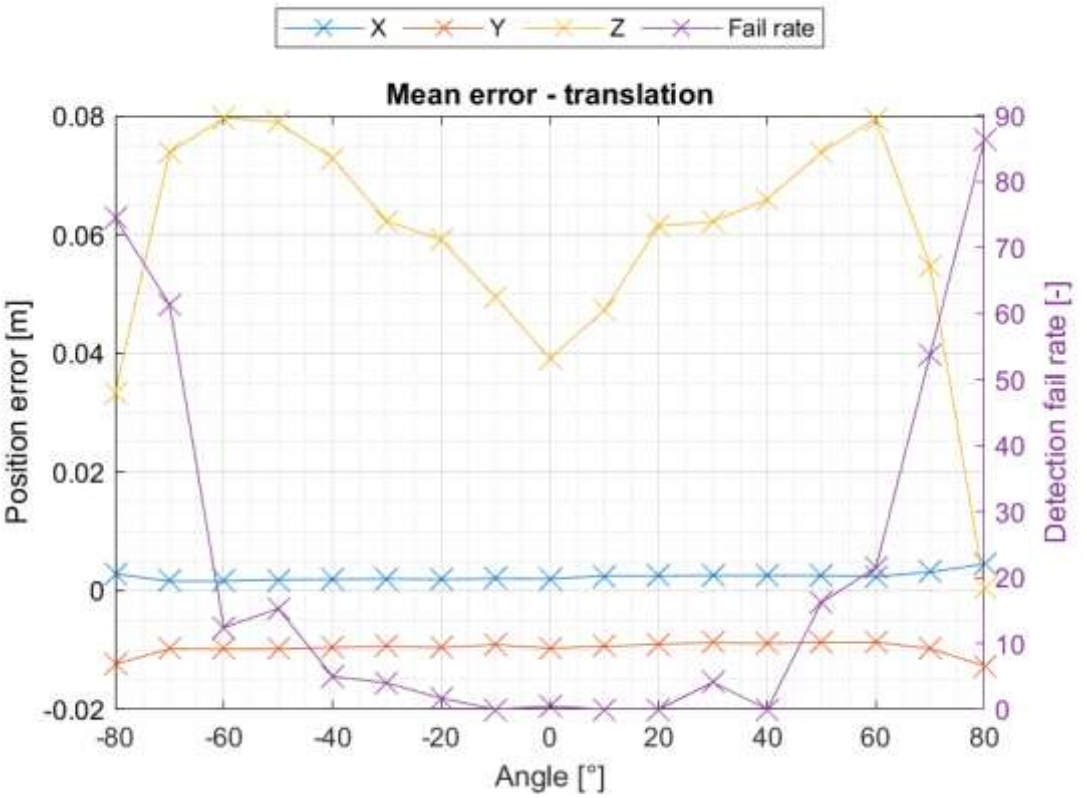
V této podkapitole jsou grafy znázorňující závislosti rotací značky. Grafů je zde dvojnásobně víc, respektive jsou zde dvě sady. Jedna sada je pro rotaci v osách X a Y a druhá pro rotaci v ose Z. Vzhledem k tomu že značka je pro kameru zřetelná ve všech rotacích v ose Z na rozdíl od os X a Y, mohlo být proměřeno celých 360 stupňů, a aby nevzniklo zbytečně mnoho hodnot, bylo zvoleno jiné měřítko pro proměření. Kdyby byly všechny tři osy v jednom grafu, byl by pak nepřehledný a z tohoto důvodu byly závislosti na rotaci po ose Z vyneseny do zvláštních grafů.



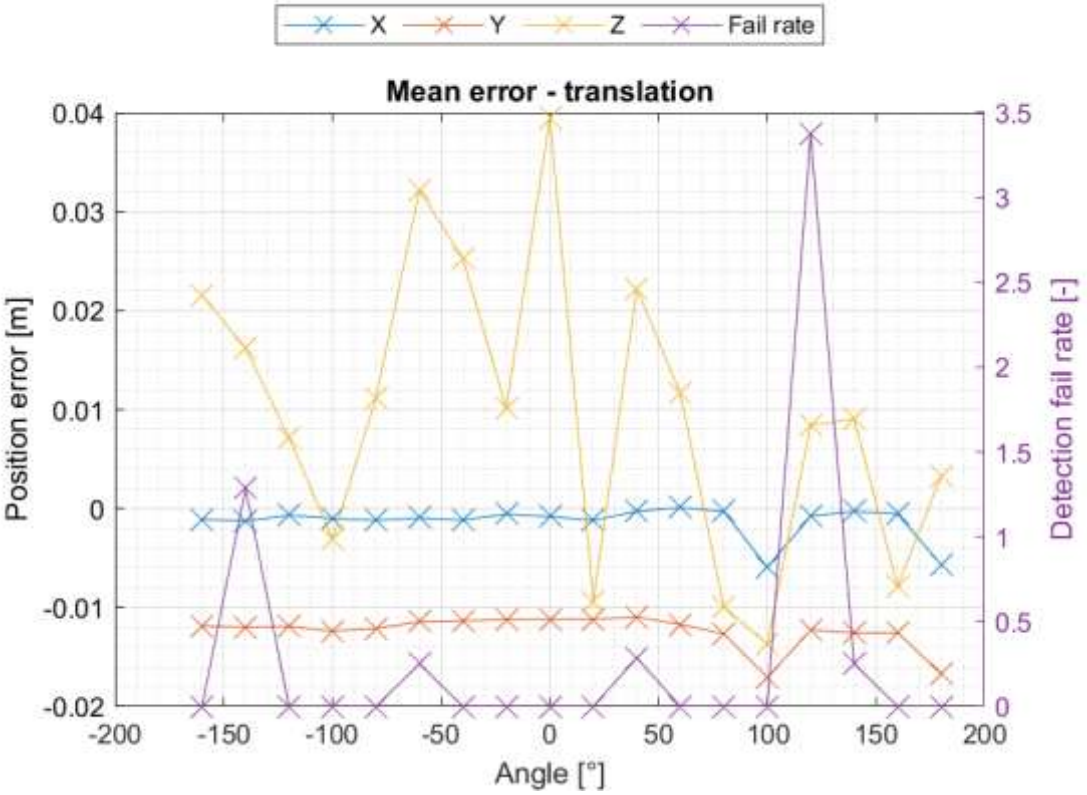
Obrázek 5.5 Závislost střední chyby rotace na rotaci os X a Y



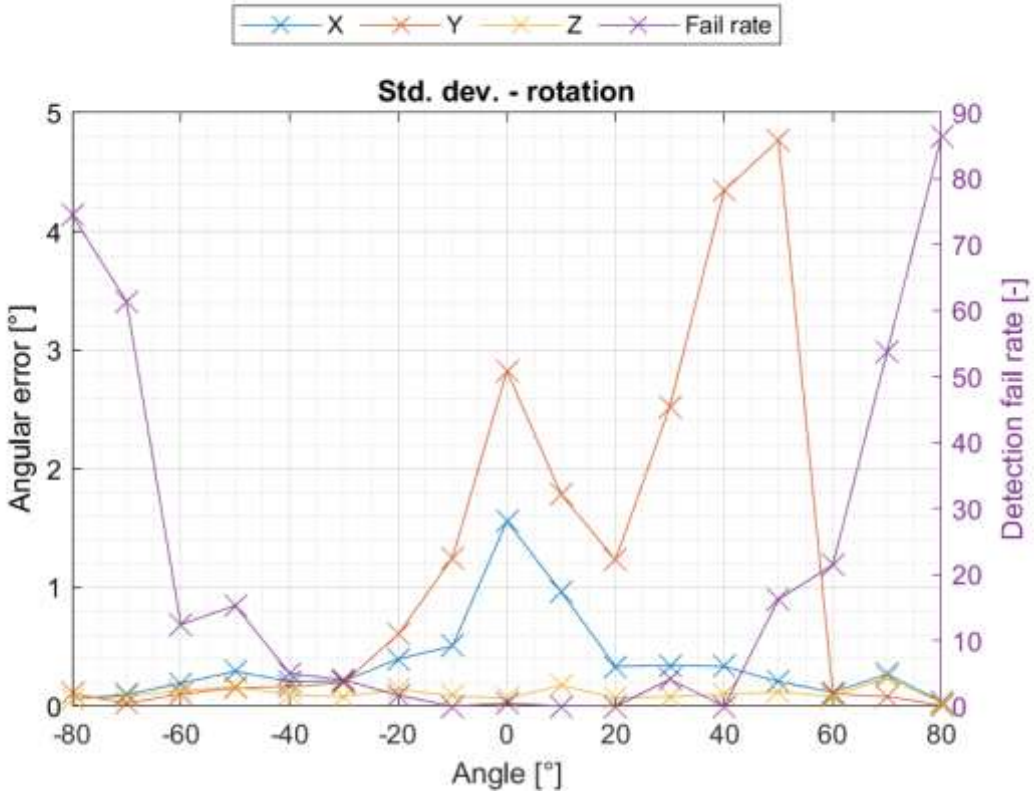
Obrázek 5.6 Závislost střední chyby rotace na rotaci osy Z



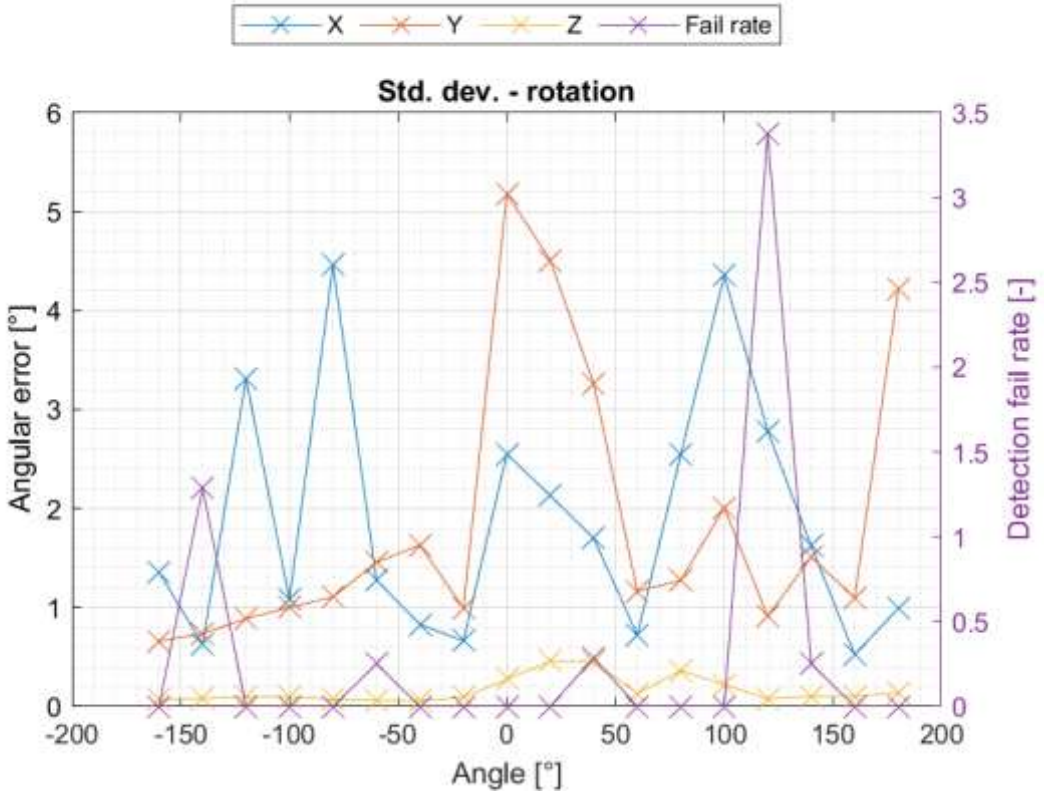
Obrázek 5.7 Závislost střední chyby translace na rotaci os X a Y



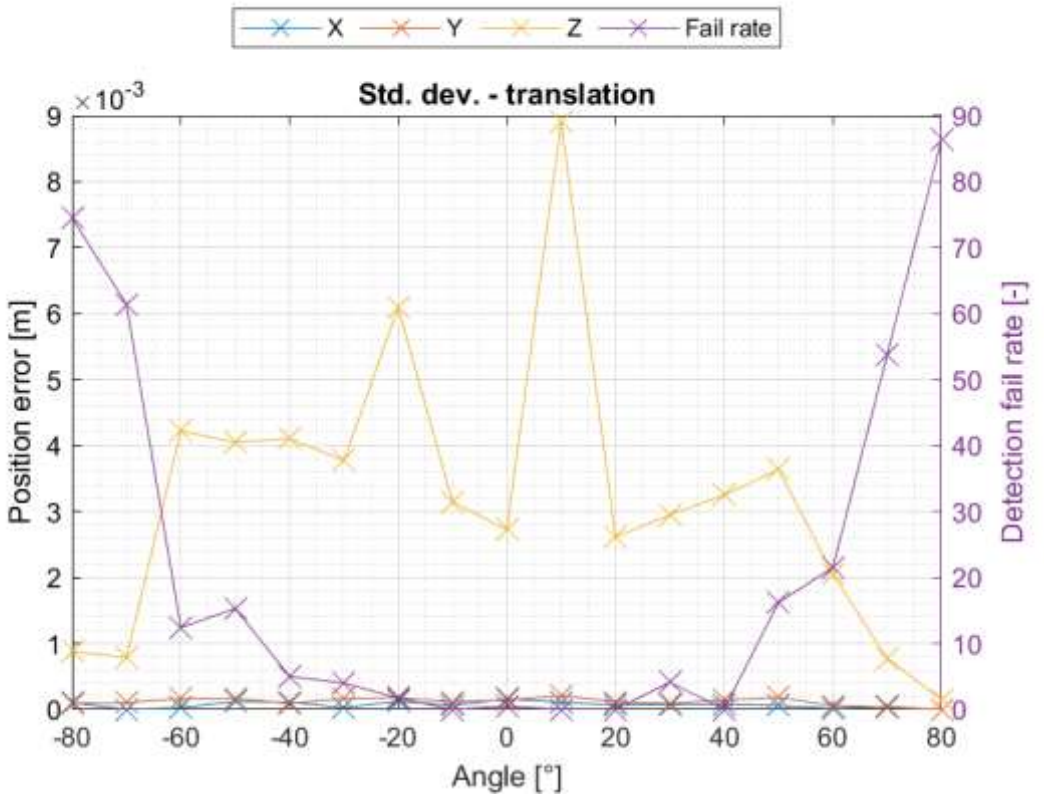
Obrázek 5.8 Závislost střední chyby translace na rotaci osy Z



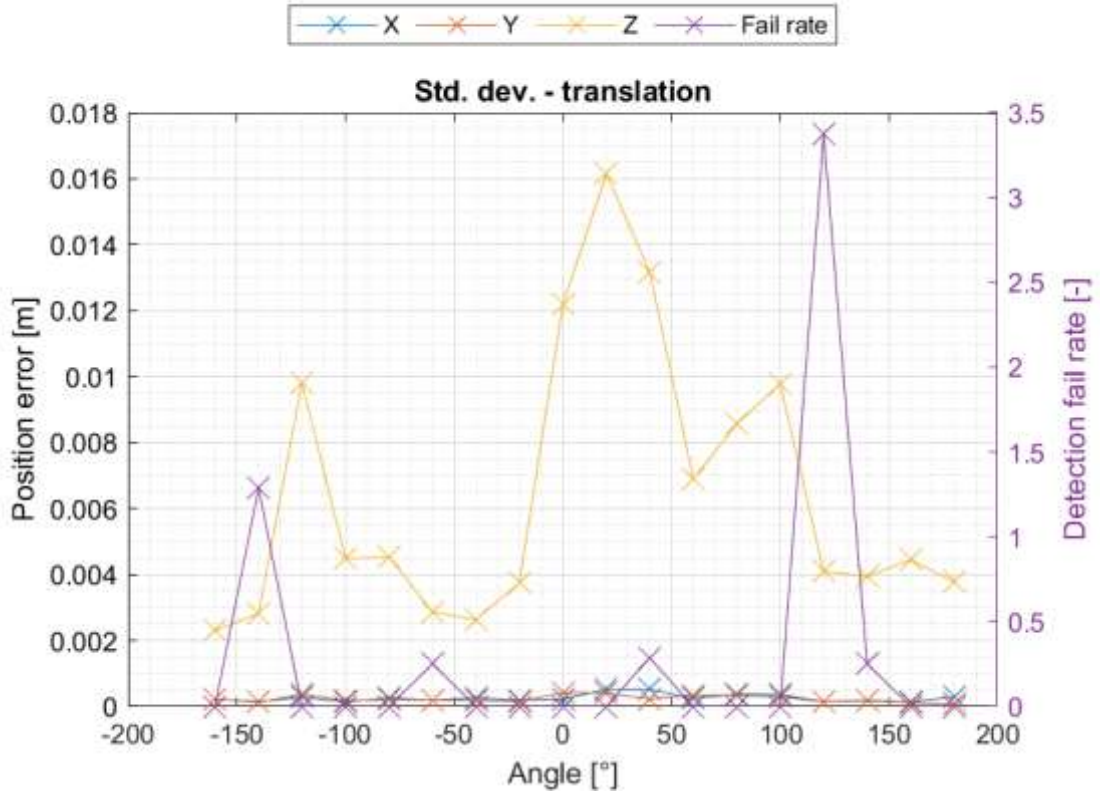
Obrázek 5.9 Závislost směrodatné odchylky rotace na rotaci os X a Y



Obrázek 5.10 Závislost směrodatné odchylky na rotaci osy Z



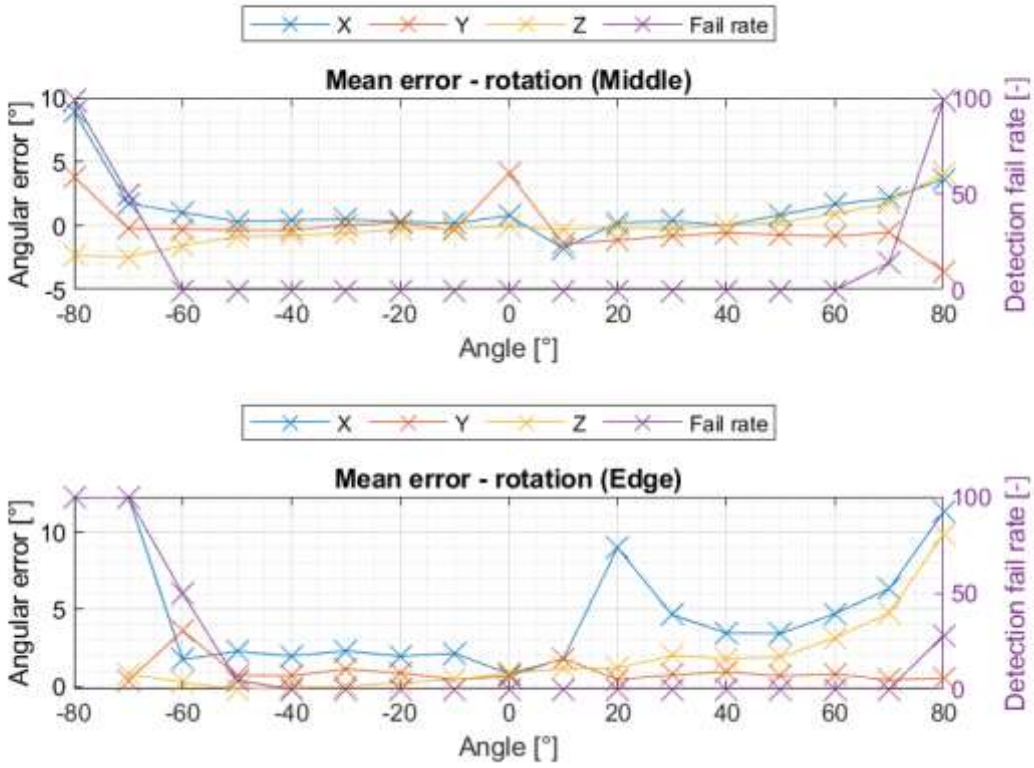
Obrázek 5.11 Závislost směrodatné odchylky translace na rotaci os X a Y



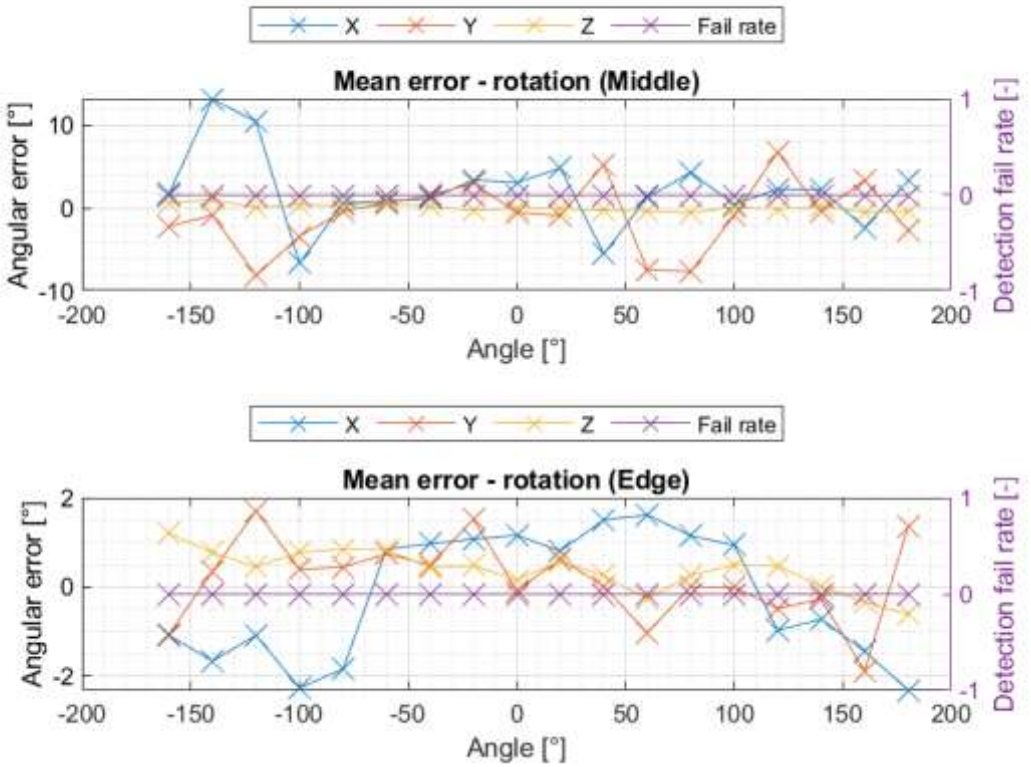
Obrázek 5.12 Závislost směrodatné odchylky translace na rotaci osy Z

5.3.3 Porovnání s krajní polohou

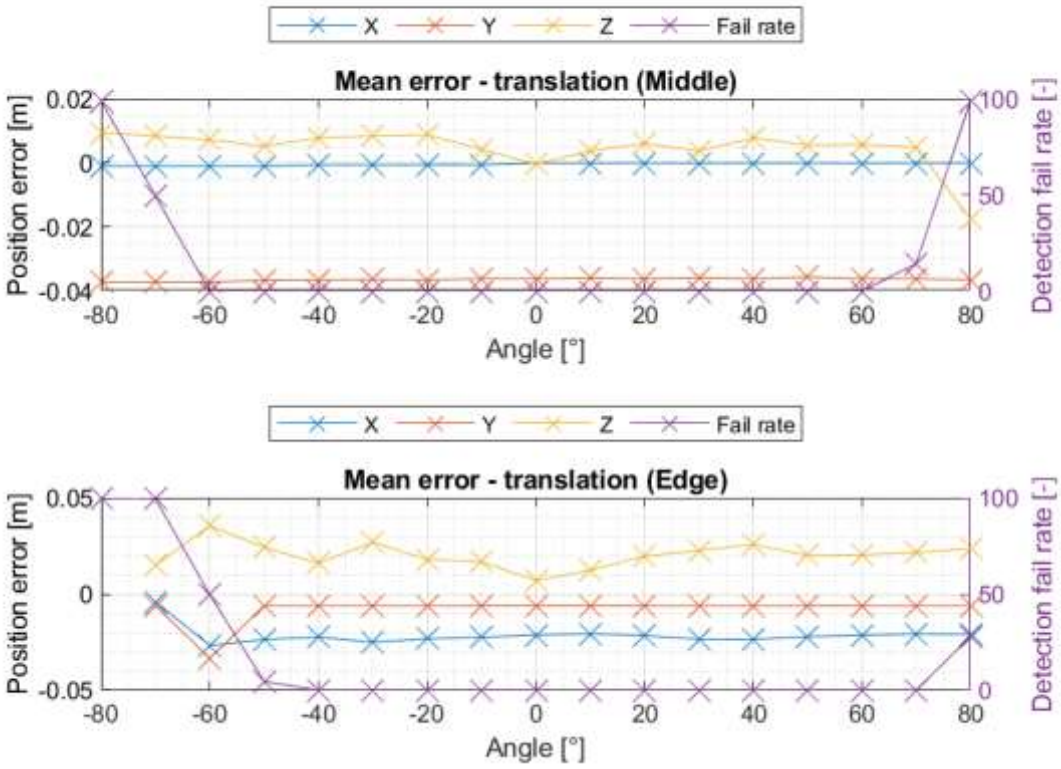
V této podkapitole jsou uvedeny grafy hodnot střední absolutní chyby a směrodatné odchylky v závislosti na úhlu natočení značky v porovnání kdy je značka ve středu obrazu a kdy je v kraji obrazu, respektive levém horním rohu (obrázek 5.21).



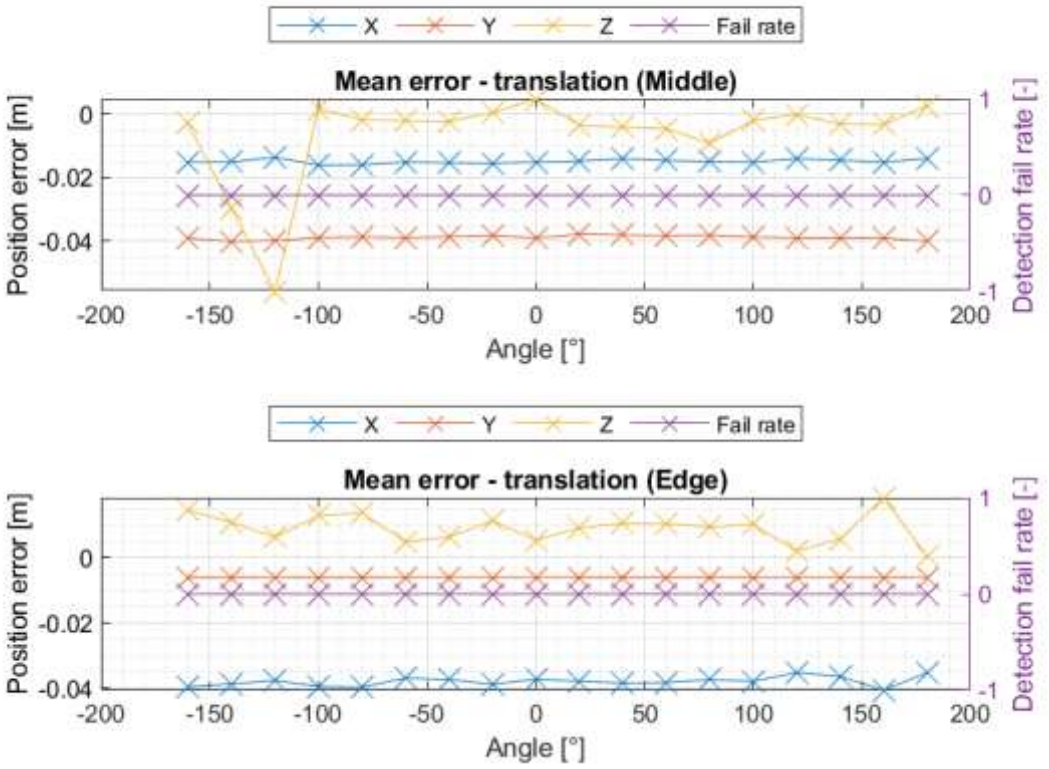
Obrázek 5.13 Závislosti střední absolutní chyby rotace na rotaci os X a Y



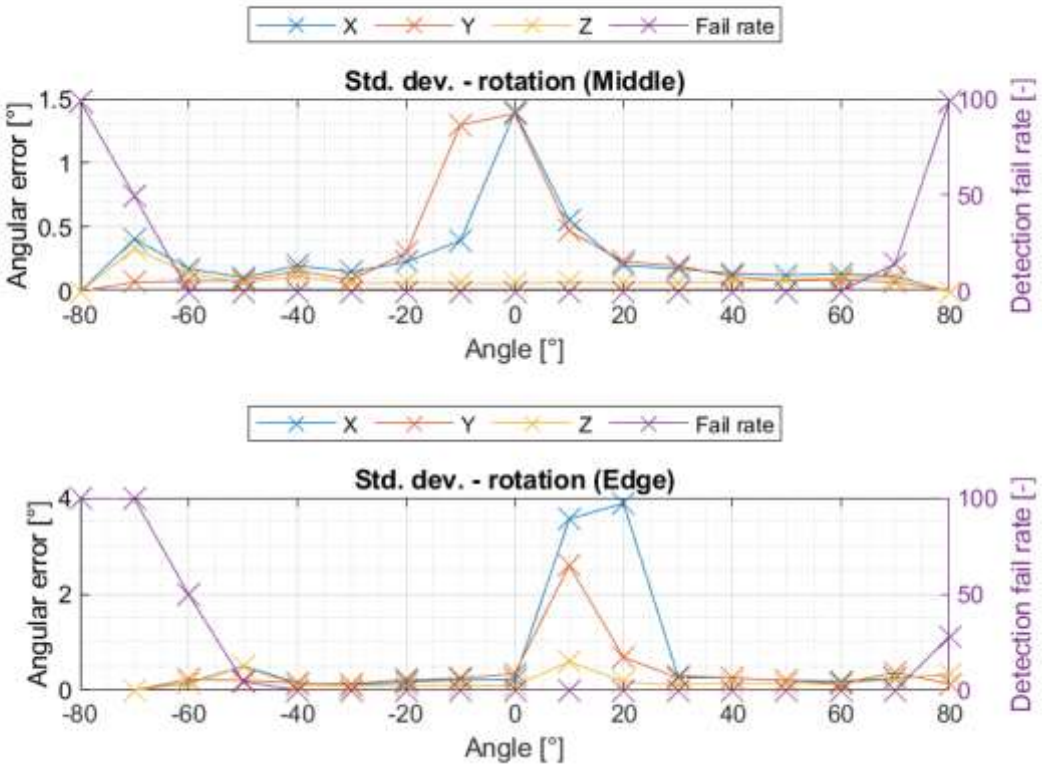
Obrázek 5.14 Závislosti střední absolutní chyby rotace na rotaci osy Z



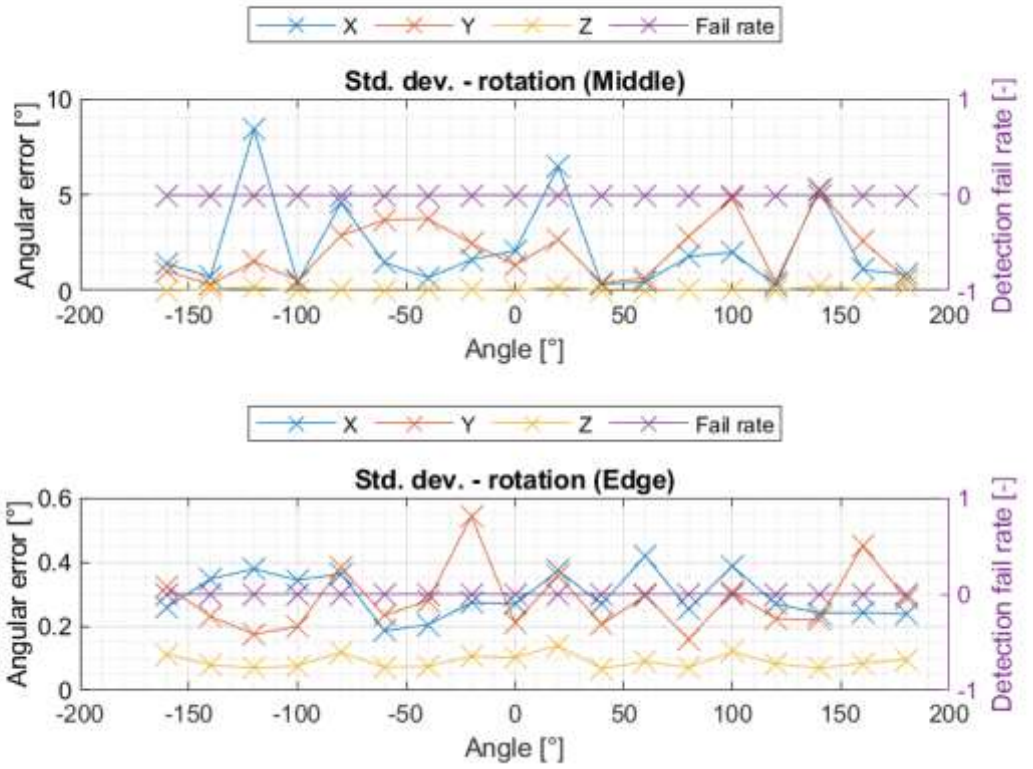
Obrázek 5.15 Závislosti střední absolutní chyby translace na rotaci os X a Y



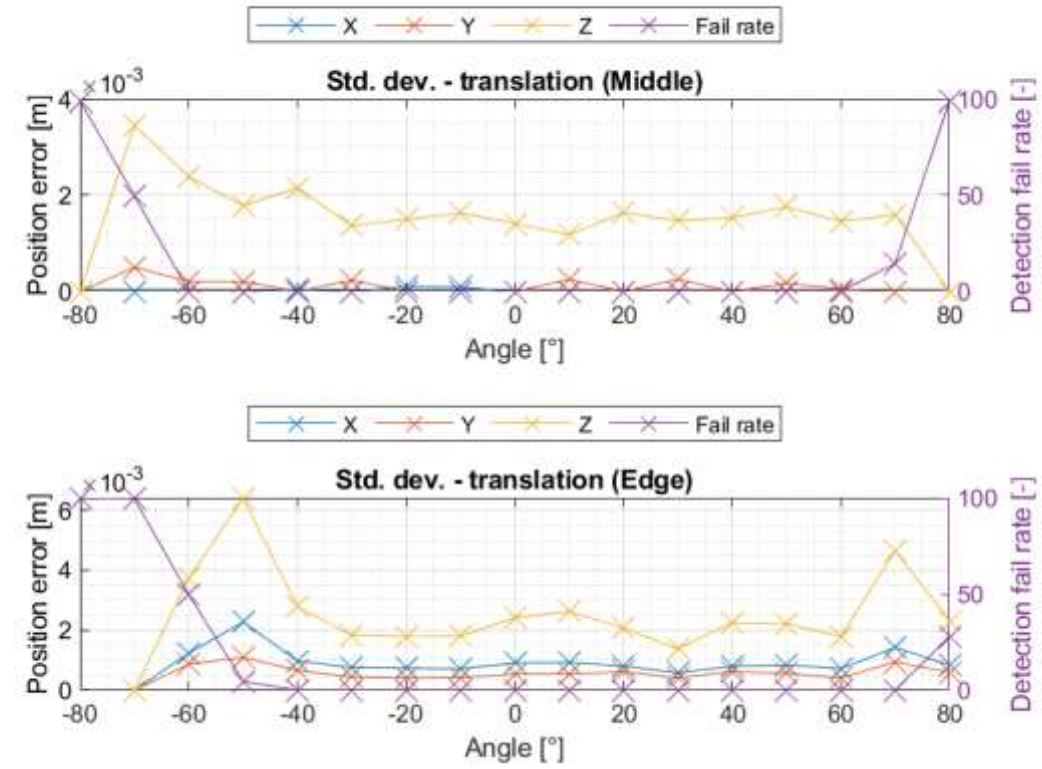
Obrázek 5.16 Závislosti střední absolutní chyby translace na rotaci osy Z



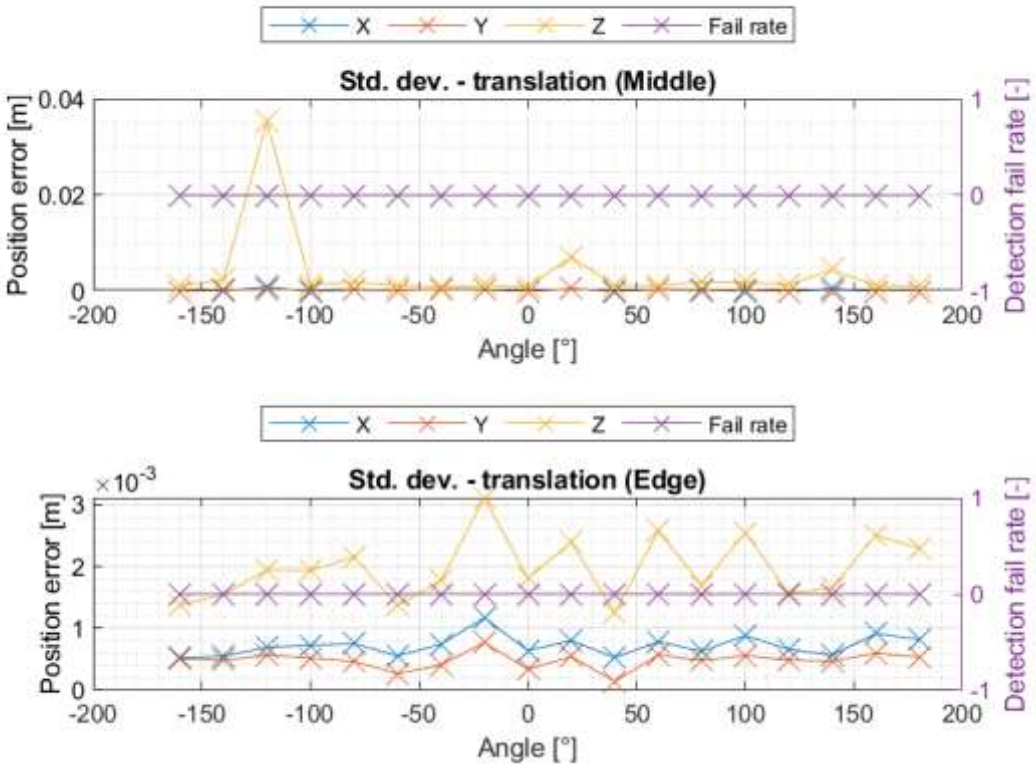
Obrázek 5.17 Závislosti směrodatné odchylky rotace na rotaci os X a Y



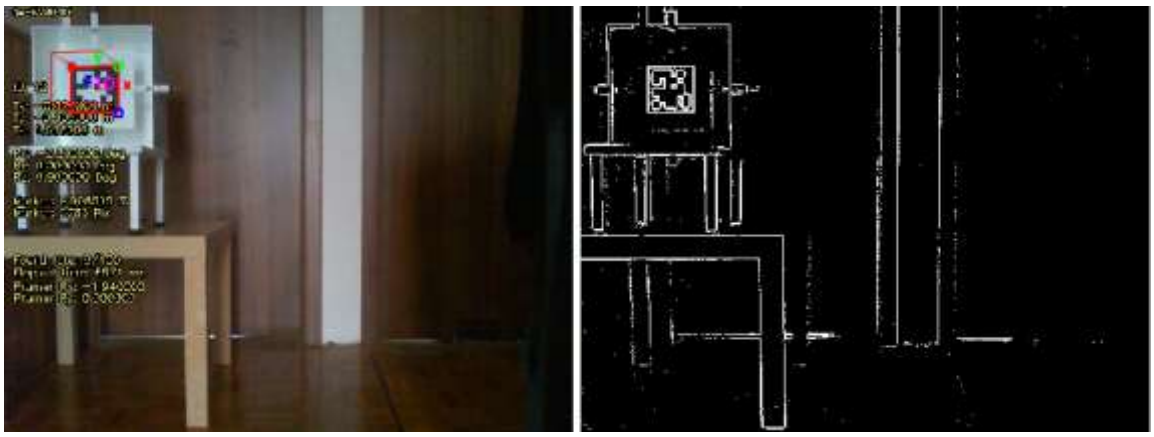
Obrázek 5.18 Závislosti směrodatné odchylky rotace na rotaci osy Z



Obrázek 5.19 Závislosti směrodatné odchylky translace na rotaci os X a Y



Obrázek 5.20 Závislosti směrodatné odchylky translace na rotaci osy Z



Obrázek 5.21 Detekce v kraji obrazu

5.4 Zhodnocení výsledných grafů

V kapitole 5.3.1 jsou uvedeny závislosti střední hodnoty absolutní chyby a směrodatné odchylky na vzdálenosti značky od kamery a v kapitole 5.3.2 jsou uvedeny stejné parametry, ovšem v závislosti na natočení značky vůči kameře. Střední hodnoty absolutní chyby v tomto měření mohou být zatíženy další chybou, a to nedokonalým nastavením nulových hodnot rotací či translací na počátku každého měření, které způsobí posun nuly a tím zvětšením či zmenšením zkreslí absolutní chybu. Na tento fakt je tedy třeba brát zřetel při čtení informací z grafů.

Ve všech grafech je na vedlejší ose Y uvedena také závislost chyby detekce neboli průměrná hodnota pro danou vzdálenost či rotaci, kolikrát značka nebyla detekována ze sta cyklů, respektive pokusů o detekci. Hodnoty na osách udávají četnost, a proto nemají žádnou jednotku, ovšem vzhledem k tomu že počet chyb je z celkového počtu sta detekcí, lze hodnoty uvažovat i jako v procentech. Při závislosti tohoto parametru na vzdálenosti, zobrazeném například v grafu na obrázku 5.1 lze pozorovat, že závislost je nepříliš vzdálená lineární závislosti. V nejmenší nastavené vzdálenosti 30 cm je průměrná chyba přibližně 5 a největší vzdálenosti se dostáváme k počtu chybných detekcí asi 26. V kapitole 5.3.2 je možné ve všech grafech vidět závislost chyby detekce na natočení značky. Na obrázku 5.5 jsou uvedeny v grafu na ose X rotace značky v osách X a Y. Je patrné že chyba je minimální do úhlu čtyřiceti stupňů, do šedesáti stupňů je přibližně v rozumných mezích chyby a pro sedmdesát a osmdesát stupňů hodnoty prudce narůstají. V závislosti na rotacích v ose Z, jak je znázorněno například na obrázku 5.6, lze očekávat, že k chyby budou minimální nebo žádné, a budou mít konstantní hodnotu pro všechny rotace. Z průběhu je vidět že chyby jsou většinou nulové až na několik výjimek, a to zejména v rotacích -140 stupňů a 120 stupňů kdy chyba stoupá přibližně na 1,5 a 3,5. Tyto špičky jsou pravděpodobně způsobeny nahodilou chybou jako je například změna vnějších světelných podmínek.

V grafu na obrázku 5.1 vykazuje nejmenší chyby rotace osa Z, dalo by se čekat, že chyby budou růst se zvětšující se vzdáleností, ovšem chyby se mění spíše nepravidelně a pohybují se v rozmezí 2 a -1 stupně kromě poslední hodnoty osy Y blížící se chybě -4 stupně. Na obrázku 5.2 je opět chyba tentokrát translace v závislosti na vzdálenosti osy X a Y se pohybují v chybě do jednoho centimetru kromě chyby v ose Y ve vzdálenosti 1,5 metru kdy je chyba téměř -4 centimetry. Osa Z je v prvních třech hodnotách vzdáleností také blízká nule jako osa X, ovšem v posledních dvou vzdálenostech se chyba prudce vyšplhala až k hodnotě 11 centimetrů.

Graf směrodatné odchylky rotace na obrázku 5.3 ukazuje že osa Z v podstatě nešumí a hodnoty odchylky se pohybují téměř na nule. Odchylka pro osy X a Y jsou přibližně totožné a postupně se vzdáleností stoupají, až na poslední hodnotu ve

vzdálenosti 3,1 metru, kde osa X zůstává v mezích 1,5 stupně a osa Y vzroste až k hodnotě odchylky 4 stupně. Na obrázku 5.4 v závislosti směrodatné odchylky translace na vzdálenosti je to naopak než v předchozím případě, kdy osy X a Y se pohybují téměř při nule a osa z postupně roste až k hodnotě odchylky kolem jednoho centimetru.

V grafu na obrázku 5.5 zobrazující závislost chyby na úhlu natočení značky v osách X a Y je vidět dle očekávání, že chyby se začínají zvětšovat v krajních hodnotách, jinak jsou téměř konstantní, kromě nulové hodnoty natočení v osách X a Y, kde je chyba zvětšená. Je také patrná značná odchylka pro osu Y v kladných rotacích pro úhly natočení 40 až 60 stupňů, kdy se chyba velmi výrazně zvětšila. Podobný jev je pak také možné pozorovat v grafu na obrázku 5.9 pro závislost směrodatné odchylky a v obrázcích 5.1 a 5.3 kdy se v největší vzdálenosti chyba a odchylka pro osu Y výrazně zvětšuje. Toto je s největší pravděpodobností způsobeno nevhodnými světelnými podmínkami, respektive nehomogenitou osvětlení, kdy při větších vzdálenostech v kladné rotaci osy Y docházelo pro úhly patrné z obrázku 5.5 k jevu, při kterém hodnota úhlu měla sice stále stejnou absolutní velikost, ale přeskakovala mezi kladnou a zápornou hodnotou. Tedy že program měl problémy s vyhodnocením, zda se jedná o kladnou či zápornou rotaci a například hodnota úhlu se měnila na 40,2 a -40,2.

Chyba rotace v závislosti na rotaci značky v ose Z je vyobrazena v grafu na obrázku číslo 5.6. Chyby pro všechny osy jsou nepravidelné a nejmenší hodnoty má právě osa Z. Osy X a Y zde vykazují značnou chybovost, ta může být částečně způsobena nepřesným umístěním středu značky na střed osy otáčení v ose Z.

Obrázky 5.7 a 5.8 zobrazují grafy závislosti chyby translace na úhlu natočení značky. Osy X a Y jsou v obou případech v podstatě konstantní s určitým offsetem, rozdíl je pak chybě translace osy Z. Ta se v rotacích osy X a Y zvětšuje do obou směrů rotace, až na poslední dvě hodnoty, kdy se opět prudce zmenší. To je nejspíše způsobeno tím, že ve větších vzdálenostech, kdy by byla chyba větší už nejsou tak ostré úhly natočení detekovány vůbec, a průměrná hodnota je tak dána pouze průměrem hodnot z menších vzdáleností kde chyba není tak velká. To může být částečně patrné z vedlejšího průběhy chyby detekce. Pro rotace značky v ose Z pak chyba nabývá menších hodnot a nepravidelně osciluje.

Na obrázcích 5.9 a 5.10 jsou závislosti směrodatné odchylky na úhlu natočení. Na obrázku 5.9 je vidět že hodnoty odchylky jsou větší v nulových rotacích os X a Y, toto šumění bylo pozorováno už při měření a v ose Y je možné pozorovat v kladných rotacích vysokou špičku, což bylo zdůvodněno výše v textu. Osa Z je pak na minimálních hodnotách odchylky a rozptyl nemá téměř vůbec. Na obrázku 5.10 závislosti na rotacích osy Z je vidět že odchylka osy Z je opět minimální a osy X a Y nepravidelně oscilují ve větších hodnotách odchylky.

Na průbězích v obrázcích 5.11 a 5.12 pak jde vidět, že translace je pro osy X a Y značně stálá a šumí pouze osa Z. Osa Z v závislosti na rotacích os X a Y se pohybuje v řádech milimetrů, a pro závislost na rotaci osy Z se translace po ose Z pohybuje do maximální hodnoty 1,6 centimetrů.

V kapitole 5.3.3 jsou stejné průběhy jako v kapitole 5.3.2 ovšem tentokrát jsou hodnoty jen pro vzdálenost 1,5 metru a v každém obrázku jsou dva grafy. První graf (Middle) vždy zobrazuje hodnoty pro značku nacházející se ve středu obrazu a druhý graf (Edge) zobrazuje hodnoty pro krajní polohu značky v obraze.

Na obrázku 5.15 jsou jasně patrné průběhy pro chybu detekce na rotacích os X a Y. Zatímco průběh pro značku ve středu obrazu je přibližně souměrný kolem nuly, průběhu pro krajní polohu ukazuje že lepší detekce probíhá pro kladné natočení značky, což je očekávané. Kamera totiž v krajní poloze nevidí značku rovnoměrně jako je tomu v poloze středové, ale v levém horním rohu obrazu, kde byla značka umístěna, kamera vidí lépe kladné rotace pro obě osy. Pokud by byla značka proměřována v pravém dolním rohu obrazu, lze s největší pravděpodobností očekávat, že detekce by naopak probíhala lépe pro záporné rotace. Pro rotaci na ose Z se je možné podívat například na obrázek 5.20 kde je jasně patrné pro oba případy, že chyba detekce je pro všechny rotace nulová, jak by se dalo očekávat.

Jak již bylo naznačeno výše v popisu průběhu chyby detekce pro osy X a Y, a jak je také patrné například z 5.13 a 5.17, charakteristiky směrodatné odchylky a chyby jsou pro krajní polohu značky posunuty o deset stupňů do kladné rotace oproti charakteristice střední polohy. V předchozím odstavci byl také vysvětlen důvod tohoto jevu, a proto ho není nutné dále popisovat. V některých grafech vyobrazující závislost chyby si je možné povšimnout větších hodnot chyb krajní polohy značky v obraze oproti střední poloze, ovšem to opět může být způsobeno neideálním nastavením výchozí polohy na začátku měření (nastavení translací a rotací), obzvlášť když přesné nastavení této polohy v kraji obrazu může být obtížnější. V některých grafech je také vidět, že pro krajní polohu značky v obraze jsou hodnoty směrodatné odchylky mírně vyšší, takže hodnoty více šumí.

6 EFEKTIVNÍ POKRYTÍ ZNAČKAMI

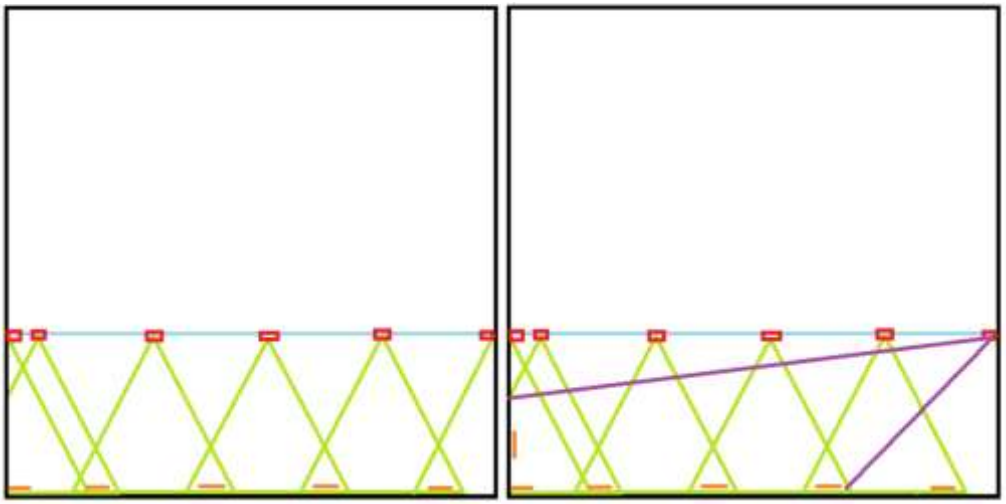
Tato kapitola se zabývá snahou o efektivní pokrytí místnosti optickými značkami za účelem navigace mobilního robota.

Pro pokrytí by měla být pracovní oblast translací a rotací zvolená tak, aby byla v kterékoliv hodnotě rotace a translace vždy spolehlivá detekce. Z naměřených hodnot zpracovaných v grafech lze určit vhodnou kombinaci, a to maximální vzdálenost jako 1,5 metru a rozsah rotací os X a Y od -60 po 60 stupňů. Plocha v obrazu značky se ve zvolené vzdálenosti pohybuje přibližně od 1000 pixelů do 3000 pixelů, v závislosti na rotaci. Hodnoty rotace byly zvoleny na základě grafu závislosti chyby detekce na úhlu natočení na obrázku 5.17, kdy se značka nachází v kraji obrazu. Detekce pro krajní hodnotu -60 stupňů vykazuje sice vyšší chybovost, ovšem je dostačující. Plocha značky v obraze se pro nejmenší vzdálenost pohybuje od 77 000 pixelů po 12 000 pixelů v závislosti na rotaci.

Pomocí výše zvolených parametrů pak lze navrhnout pokrytí místnosti značkami tak, aby vždy byla mobilním robotem nějaká detekována. Pro příklad mějme místnost 6x6 metru, pro kterou zde bude navrhnuto pokrytí značkami. Dále je dobré zvolit nějakou minimální vzdálenost od zdi, pro kterou musí být nějaká značka vždy viditelná. Pro tento příklad by minimální vzdálenost od zdi pro detekci mohla být zvolena jako 2 metry.

Nyní je třeba určit potřebné velikosti značek tak, aby byla pokryta celá délka místnosti, tedy 6 metrů. Vzdálenost do 1,5 metru je již určeno měřením v předchozí kapitole, a tedy velikost značky bude 12x12 centimetrů. Další velikosti budou určeny experimentálně tak, aby pro daný interval vzdáleností vykazovali stejné vlastnosti jako již zvolená značka, tedy v jaké vzdálenosti má určitá velikost značky stejnou plochu v obraze. Aby nemusely být tisknuty různé velikosti a následně testovány v různých vzdálenostech, byly velikosti značek změněny pouze virtuálně. Byla použita proměřená značka 12x12 cm a kamera byla umístěna na hraniční vzdálenost zvoleného pracovní oblasti, tedy 1,5 metru. Následně byly programu předávány falešné fyzické velikosti značky s tím, že plocha značky v obraze zůstala vždy stejná, program pouze vyhodnotil jinou vzdálenost kamery od značky. Takto bylo postupováno, dokud nebyla nalezena dostatečná vzdálenost pro pokrytí velikosti místnosti. Následně bylo stejným způsobem zjištěna nejmenší vzdálenost značky a kamery, aby bylo ověřeno, že mezi pracovními oblastmi pro obě značky není slepé místo. Výsledná fyzická velikost značky byla určena jako 50x50 centimetrů, která má pracovní oblast 1 až 6,2 metrů. Vzhledem k tomu, že nejmenší vzdálenost pro detekci značky byla stanovena na 2 metry, lze tuto nově zjištěnou velikost značky použít pro celý rozsah místnosti a není nutné používat více velikostí.

Dalším krokem je určení počtu značek a jejich rozmístění. Bylo zjištěno že kamera při vzdálenosti dvou metrů od stěny snímá šířku přibližně také dvou metrů stěny. Tuto úlohu lze řešit graficky, jak je znázorněno na obrázku 6.1 vlevo (obrázek je spíše ilustrační). Černá obrysová čára znázorňuje místnost, modrá čára označuje vzdálenost dvou metrů od spodní stěny, tedy nejmenší zvolenou vzdálenost pro detekci. Červený čtvereček na obrázku znázorňuje různé polohy robota s kamerou a zelený kužel značí jeho zorné pole. Polohy robota jsou zde voleny kolmo ke stěně tak, aby se zorná pole vždy protínala aspoň šedesáti centimetrovou plochou, takže dostatečně pro 50 centimetrů velkou značku. Značky jsou na obrázku znázorněny jako oranžové obdélníky. Střed první značky zprava je vzdálen 70 centimetru od stěny, další tři rozestupy značek jsou 140 centimetrů a poslední značka je těsně u levé stěny, protože v případné krajní levé poloze robota by čtvrtá značka nebyla detekována, jak je patrné z obrázku vlevo.



Obrázek 6.1 Rozmístění značek

Na obrázku 6.1 vpravo je pak vidět že když se robot natočí, zorné pole znázorněno fialovými čarami, stále detekuje bez problémů prostřední značku, která má vzájemnou rotaci vůči kameře menší než zvolených 60 stupňů, tedy je v toleranci. Zároveň je také detekována značka na protější stěně, takže detekce je bezpečně zajištěna.

Výsledkem je tedy že na jednu stěnu je třeba umístit 5 značek ve vzdálenostech od pravé stěny 70, 210, 350, 490 a 575. Celkem je tedy do místnosti umístěno 20 značek.

7 ZÁVĚR

Na základě provedené rešerše v kapitole 2 na téma existujících implementací optických značek byla pro tuto práci zvolena knihovna ArUco, a to z důvodů jako například dostupnost, multiplatformní provedení či množství funkcí. Knihovna je blíže popsána v kapitole 3.

Po zprovoznění knihovny byla zkalibrována kamera pro získání potřebných parametrů na určování polohy. Dále byl dostupný program pro detekci upraven a doplněn dalšími nezbytnými funkcemi. Tyto úpravy jsou podrobně popsány v kapitole 4. Ve stejné kapitole je také popsán přípravek pro nastavování rotací, vyrobený speciálně pro tuto práci.

V následující kapitole číslo 5 je popsáno samotné měření, stejně jako použité vybavení a výsledky měření. Grafické zpracování výsledných hodnot je v kapitole 5.3 pro různé závislosti a v kapitole 5.4 jsou tyto grafy slovně zhodnoceny včetně možných nejistot vstupujících do měření.

V měření byly ověřeny vlastnosti detekce, a to jak úspěšnost, tak i přesnost. Pracovní oblast byla proměřena v rozsahu vzdáleností 0,3 až 3,1 metru a rozsah natočení značky je -80 až 80 stupňů pro osy X a Y a -160 až 180 stupňů pro osu Z. V této pracovní oblasti byla přesnost translace stanovena v řádu jednotek až nižších desítek milimetrů, a to konkrétně 12,5 milimetrů pro střední absolutní chybu a 1,6 milimetrů pro směrodatnou odchylku s nejistotou referenční hodnoty 2 milimetry. Ve stejné pracovní oblasti byla také stanovena přesnost rotace, která je v řádech desetin stupně, a to 0,1 stupně pro střední absolutní chybu a 0,9 stupně pro směrodatnou odchylku s nejistotou referenční hodnoty stanovenou jako 0,7 stupně.

Průměrná úspěšnost detekce se v závislosti na vzdálenosti mění přibližně od 5 do 26 procent chybovosti. Závislost na vzdálenosti zde patrná je, ovšem není tak kritická jako v případě závislosti průměrné chyby detekce na rotaci značky v osách X a Y. Zde se chybovost detekce pohybuje v rozmezí od 0 do přibližně 85 procent. Pro rotaci v ose Z je chybovost malá, a vzniklé chyby jsou spíše způsobeny okolními podmínkami než samotnou rotací.

V kapitole číslo 6 je uveden příklad pro efektivní rozmístění značek v místnosti a zároveň je zde určena pracovní oblast (0,3 až 1,5 metrů pro translaci a -60 až 60 stupňů pro rotaci) pro bezpečnou detekci

Hlavní nejistotou vstupující do měření byly při testování zpozorovány jako světelné podmínky, tedy stíny či odlesky a nehomogenita osvětlení, které mohli mít negativní vliv na výsledky.

LITERATURA

[1] *M. Fiala, "Designing Highly Reliable Fiducial Markers,"* in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 7, pp. 1317-1324, July 2010.
DOI: 10.1109/TPAMI.2009.146

[2] *M. Fiala, "ARTag, a fiducial marker system using digital techniques,"* 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 2005, pp. 590-596 vol. 2.
DOI: 10.1109/CVPR.2005.74

[3] *E. Olson, "AprilTag: A robust and flexible visual fiducial system,"* 2011 IEEE International Conference on Robotics and Automation, Shanghai, 2011, pp. 3400-3407.
DOI: 10.1109/ICRA.2011.5979561

[4] *SAGITOV, Artur, Ksenia SHABALINA, Leysan SABIROVA, Hongbing LI a Evgeni MAGID. ARTag, AprilTag and CALTag Fiducial Marker Systems: Comparison in a Presence of Partial Marker Occlusion and Rotation.* 14th International Conference on Informatics in Control, Automation and Robotics, 2017, 10.
DOI: 10.5220/0006478901820191.

[5] *Sample Markers of ARToolKit. In: ResearchGate* [online]. Pakistan, 2014 [cit. 2019-12-10]. Dostupné z*: https://www.researchgate.net/figure/Sample-Markers-of-ARToolKit-ARToolKit-2012_fig2_236461845*

[6] *Examples of different AprilTag families*. In: *ResearchGate* [online]. [cit. 2019-12-13]. Dostupné z: https://www.researchgate.net/figure/Examples-of-different-AprilTag-families_fig1_328188091

[7] *CALTag Checkerboard*. In: *Brown University School of Engineering* [online]. Providence, 2016 [cit. 2019-12-13]. Dostupné z: http://mesh.brown.edu/3DP-2018/hw3/hw3.html

[8] Speeded up detection of squared fiducial markers. *Image and Vision Computing* [online]. 2018, (72), 38-47 [cit. 2019-12-20]. Dostupné z: https://www.sciencedirect.com/science/article/pii/S0262885618300799?via%3Dihub

[9] Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition* [online]. 2014, 6(47), 2280-2292 [cit. 2019-12-20]. Dostupné z: https://www.sciencedirect.com/science/article/abs/pii/S0031320314000235

[10] Generation of fiducial marker dictionaries using Mixed Integer Linear Programming. *Pattern Recognition* [online]. 2016, (51), 481-491 [cit. 2019-12-20]. Dostupné z: https://www.sciencedirect.com/science/article/abs/pii/S0031320315003544

[11] D. H. Douglas, T. K. Peucker, Algorithms for the reduction of the number of points required to represent a digitized line or its caricature, Cartographica: The International Journal for Geographic Information and Geovisualization 2 (10) (1973) 112 - 122.
[12] N. Otsu, A threshold selection method from gray-level his tograms, IEEE Transactions on Systems, Man, and Cybernetics 9 (1) (1979) 62-66.
[13] ArUco: a minimal library for Augmented Reality applications based on OpenCV. *Uco.es* [online]. University of Cordoba [cit. 2019-12-20]. Dostupné z: https://www.uco.es/investiga/grupos/ava/node/26
[14] OpenCV [online]. 2019 [cit. 2019-12-21]. Dostupné z: https://opencv.org/
[15] CMake [online]. 2019 [cit. 2019-12-21]. Dostupné z: https://cmake.org/
[16] Rotations and the Euler angles [online]. In: . s. 5 [cit. 2020-06-07]. Dostupné z:https://www.phas.ubc.ca/~berciu/TEACHING/PHYS206/LECTURES/FILES/euler.pdf
[17] The Rodrigues Formula and Polynomial Differential Operators. Journal of Mathematical Analysis and Applications. 1981, 84, 443-482. [18] SLABAUGH, Gregory G. Computing Euler angles from a rotation matrix. 2017. Dostupné také z: https://www.gregslabaugh.net/publications/euler.pdf
[19] ArUco. SourceForge [online]. 2019 [cit. 2019-12-26]. Dostupné z: https://sourceforge.net/projects/aruco/files/
[20] Quadrilateral. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2020-03-19]. Dostupné z: https://en.wikipedia.org/wiki/Quadrilateral