VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

# DIPLOMOVÁ PRÁCE

Brno, 2020 Bc. David Šála

# 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


# SYSTÉM PRO AUTOMATICKOU KALIBRACI ROBOTICKÉHO NÁSTROJE
SYSTEM FOR AUTOMATIC CALIBRATION OF A ROBOTIC TOOL

DIPLOMOVÁ PRÁCE
MASTER'S THESIS


AUTOR PRÁCE — Bc. David Šála
AUTHOR

VEDOUCÍ PRÁCE — prof. Ing. Luděk Žalud, Ph.D.
SUPERVISOR



BRNO 2020

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

# Diplomová práce

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

Ústav automatizace a měřicí techniky

***Student:*** Bc. David Šála ***ID:*** 186200
***Ročník:*** 2 ***Akademický rok:*** 2019/20

**NÁZEV TÉMATU:**

## Systém pro automatickou kalibraci robotického nástroje

**POKYNY PRO VYPRACOVÁNÍ:**

Cílem práce je navrhnout a realizovat experimentální vzorek zařízení pro automatickou nástrojovou kalibraci robotického manipulátoru pomocí metod strojového vidění. Systém sestává ze skupiny kamer umístěných definovaným způsobem kolem průmyslového manipulátoru. Na základě obrazu z těchto kamer je vypočtena kalibrace nástroje uchopeného robotickým manipulátorem.

1. Proveďte rešerši stávajících metod manuální a automatické nástrojové kalibrace robota. Popište specifika prostudovaných metod.
2. Prozkoumejte techniky strojového vidění, které jsou vhodné pro realizaci kalibračního sytému.
3. Podrobně navrhněte testovací vzorek zařízení pro automatickou kalibraci robotického nástroje. Popište cílovou robotickou platformu, konstrukční návrh testovacího zařízení, vyberte a popište volbu vhodné výpočetní jednotky a kamer pro realizaci vzorového řešení.
4. Popište postup práce s kalibračním zařízením, navrhněte grafické rozhraní výsledné kalibrační aplikace.
5. Vytvořte výslednou implementaci navrženého kalibračního systému.
6. Proveďte vyhodnocení výsledků kalibrace robotického nástroje a porovnejte dosažené výsledky s obdobnými zařízeními.

**DOPORUČENÁ LITERATURA:**

Šonka M., Hlaváč, V., Boyle R.: Image Processing, Understanding, and Machine Vision, 4th edition Cengage Learning 2015

***Termín zadání:*** 3.2.2020 ***Termín odevzdání:*** 1.6.2020

***Vedoucí práce:*** prof. Ing. Luděk Žalud, Ph.D.

**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

Tato diplomová práce popisuje návrh a realizaci experimentálního vzorku pro automatickou nástrojovou kalibraci robotického manipulátoru pomocí metod strojového vidění pod záštitou společnosti SANEZOO EUROPE s.r.o. Zabývá se rozborem všech používaných metodik provádění TCP kalibrace, na jejichž základech je realizována. Aplikace vychází z Point-counterpoint metody, kdy je robot naveden proti kalibračnímu bodu nastavenému pomocí ArUco markeru ze tří různých směrů, a to všech vůči sobě navzájem kolmých. K detekci koncového bodu je použito snímků ze dvou kamer, které mezi sebou svírají pravý úhel. Na snímcích je pomocí konvenčních metod počítačového vidění a HSV filtru nalezen koncový bod nástroje, který je naváděn na kalibrační bod. Ze získaných souřadnic je pomocí optimalizační metody Optimalizace hejnem částic (PSO) nalezen hledaný koncový bod robotického nástroje v souřadnicích robota. Tato aplikace tedy v rychlém čase poskytuje TCP kalibraci, čímž snižuje odstávku ve výrobě, a to téměř bez zásahu člověka.

## KLÍČOVÁ SLOVA

TCP kalibrace, Optimalizace hejnem částic, počítačové vidění, robot, uživatelské grafické rozhraní, JSON

## ABSTRACT

This Master's thesis describes the design and implementation of an experimental sample for automatic calibration of a robotic tool using machine vision methods under the auspices of the company SANEZOO EUROPE s.r.o. It deals with the analysis of all used methods of performing TCP calibration, on the basis of which it is implemented. The application is based on the Point-counterpoint method, where the robot is guided against the calibration point from three different directions, all perpendicular to each other. The calibration point is set using the ArUco marker. In order to detect the endpoint are used images from two cameras that are at the right angles to each other. Using conventional computer vision methods and an HSV filter, the endpoint of the instrument is found in the images and is guided to the calibration point. From the obtained coordinates, the searched endpoint of the robotic tool in the robot coordinates is found using the optimization method Particle Swarm Optimization. This application, therefore, performs TCP calibration in a fast time, thus reducing production downtime almost without human intervention.

## KEYWORDS

TCP calibration, Particle swarm optimization, Computer vision, robot, graphical user interface, JSON

ŠÁLA, David. *Systém pro automatickou kalibraci robotického nástroje*. Brno, Rok, 87 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce: prof. Ing. Luděk Žalud, Ph.D.

Vysázeno pomocí balíčku `thesis` verze 3.05; `http://latex.feec.vutbr.cz`

# PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Systém pro automatickou kalibraci robotického nástroje“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

Brno 1. 6. 2020 .....

podpis autora

# PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu prof. Ing. Luďkovi Žaludovi, Ph.D. a konzultantovy Ing. Adamu Ligockému za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci. Také bych rád poděkoval společnosti SANEZOO EUROPE s.r.o., za zázemí, trpělivost a prostor potřebný pro realizaci této práce.

Brno 1. 6. 2020 .....

podpis autora

# Obsah

**Úvod** **17**

**1 Prerekvizity Systému** **19**
1.1 Homogenní transformace . . . 19
1.2 Přepočet souřadnic robota . . . 19
1.3 Optimalizace hejnem částic (PSO) . . . 20
1.4 Kalibrace kamer . . . 21
1.4.1 Dírkový model kamery . . . 21
1.4.2 Distorze čoček . . . 23
1.4.3 OpenCV . . . 24

**2 Metody počítačového vidění** **27**
2.1 Filtrace šumu . . . 27
2.1.1 Konvoluční maska . . . 28
2.2 Detekce hran . . . 29
2.2.1 Cannyho hranový detektor . . . 29
2.3 Morfologické operace . . . 31
2.3.1 Dilatace (Dilation) . . . 32
2.3.2 Eroze(Erosion) . . . 32

**3 Metody kalibrací robotického nástroje** **35**
3.1 Tool Center Point . . . 35
3.2 Manuální kalibrace . . . 36
3.3 Automatická kalibrace . . . 37
3.3.1 BullsEye . . . 37
3.3.2 Kalibrace na základě dotekové síly . . . 38
3.3.3 LaserLab . . . 39
3.3.4 State of The Art - LEONI . . . 40
3.4 Shrnutí . . . 40

**4 Návrh Pracoviště** **43**
4.1 Návrh testovacího zařízení . . . 43
4.1.1 Konstrukce . . . 44
4.1.2 Realizace . . . 44
4.2 Robotická platforma . . . 44
4.3 Použitý hardware . . . 46
4.3.1 Kamery . . . 46
4.3.2 Objektivy . . . 46

4.3.3 Svítidlo . . . . . 47
4.3.4 Výpočetní jednotka . . . . . 47
4.3.5 Rozhraní . . . . . 47
4.3.6 Označení chapadla robota . . . . . 47
4.3.7 Kalibrační marker . . . . . 47
4.4 Shrnutí . . . . . 48

**5 Implementace** **49**
5.1 Proces kalibrace obecně . . . . . 49
5.2 Detekce koncového bodu nástroje . . . . . 50
5.3 Stanovení translace TCP vůči flange . . . . . 52
5.3.1 Algoritmus výpočtu . . . . . 52
5.4 Grafické uživatelské rozhraní . . . . . 54
5.4.1 Hlavní stránka . . . . . 55
5.4.2 Menu Settings . . . . . 55
5.4.3 Menu Camera . . . . . 59
5.4.4 Menu Help . . . . . 62
5.5 JSON konfigurační soubor . . . . . 62
5.5.1 Vytvoření JSON souboru . . . . . 64
5.5.2 Výsledný JSON . . . . . 65
5.6 Logování . . . . . 66
5.7 Programovací jazyk . . . . . 66
5.8 Shrnutí . . . . . 66

**6 Vyhodnocení výsledků kalibrace** **69**
6.1 Stanovený cíl . . . . . 69
6.2 Dílčí kalibrace . . . . . 69
6.3 Automatická kalibrace vs. Manuální kalibrace . . . . . 70

**Závěr** **73**

**Literatura** **75**

**Seznam příloh** **79**

**A Návrh kalibračního přípravku** **81**

**B Realizace kalibračního přípravku** **83**

**C Výpis použitého PSO algoritmu** **85**

**D Obsah přiloženého CD** **87**

# Seznam obrázků


1.1 Model dírkové kamery . . . . . 22
1.2 Orientace kamery v prostoru . . . . . 23
1.3 Distorze čočky a její odstranění . . . . . 25
2.1 Ukázka filtrace šumu . . . . . 27
2.2 Ukázka výpočtu konvoluce . . . . . 28
2.3 Ukázka aplikace Cannyho hranového detektoru . . . . . 30
2.4 Ukázka dilatace . . . . . 32
2.5 Ukázka eroze . . . . . 32
3.1 Výsledek TCP kalibrace . . . . . 35
3.2 Manuální TCP kalibrace . . . . . 36
3.3 BullsEye kalibrace . . . . . 37
3.4 Touch probe kalibrace . . . . . 38
3.5 LaserLAB kalibrace . . . . . 39
3.6 6D Laserová kalibrace . . . . . 41
4.1 Návrh konstrukce kalibračního vzorku . . . . . 43
4.2 Realizace kalibračního vzorku I . . . . . 44
4.3 Realizace kalibračního vzorku II . . . . . 45
4.4 6-osý robotický manipulátor Stäubli TX2-90 . . . . . 45
4.5 MATRIX VISION BlueFox3 USB3 Vision Camera . . . . . 46
4.6 Označené 2-prsté chapadlo . . . . . 48
4.7 ArUco marker . . . . . 48
5.1 Blokový diagram aplikace . . . . . 49
5.2 Aplikace HSV filtru . . . . . 51
5.3 Hledání kontur a TCP . . . . . 52
5.4 Vizualizace optimalizační úlohy . . . . . 53
5.5 Hlavní stránka GUI . . . . . 54
5.6 vestavěný nástroj HSV filtr . . . . . 56
5.7 vestavěný nástroj nastavení parametrů OpenCV metod . . . . . 57
5.8 Okno nastavení připojení robota . . . . . 58
5.9 Kalibrační pozice vůči kalibračnímu bodu . . . . . 59
5.10 Nastavení výchozích souřadnic nástroje . . . . . 60
5.11 Menu stereo kalibrace . . . . . 61
5.12 Ukázka logovacího výpisu . . . . . 66
6.1 Srovnání výsledků optimalizační úlohy . . . . . 72
A.1 Nárys návrhu přípravku . . . . . 81
A.2 Půdorys návrhu přípravku . . . . . 81
A.3 Bokorys návrhu přípravku . . . . . 82

A.4 Celkový náhled návrhu přípravku . . . . . 82
B.1 Horní pohled na přípravek . . . . . 83
B.2 Čelní pohled na přípravek . . . . . 84

# Seznam tabulek

6.1 Srovnání výsledků kalibrace kamer . . . . . 70
6.2 Výsledek kalibrace robot-kamera . . . . . 70
6.3 Srovnání vybraných parametrů automatické a manuální kalibrace . . 71

# Úvod

V dnešní uspěchané době pokročilých technologií, se klade ve všech odvětvích průmyslu stále vyšší nárok na kvalitu, rychlost, cenu a kvantitu. K tomu dopomáhají nejmodernější stroje, prostřednictvím kterých se daří tyto stále se zvyšující nároky plnit.

Nejprve všechnu práci v továrnách prováděl sám člověk. Používal primitivní nástroje, díky nimž jeho práce byla z dnešního pohledu velmi nepřesná. Později s příchodem lepších nástrojů a rozšířením elektrifikace se naplno začaly rozvíjet nám známé požadavky na průmyslovou výrobu. Protože každý člověk má omezenou kapacitu úkonů, které může provádět a mnohonásobné znásobení této pracovní síly znamenalo i nárůst nákladů, byly vynalezeny první poloautomatické až automatické stroje, které zvládaly více, lépe a kvalitněji produkovat výrobky než člověk...

Tento koloběh neustále pokračoval a pokračuje dodnes. V dnešní době průmyslu 4.0 se nám neustále otevírají nové možnosti a výzvy, které nás posouvají stále dál a dál. Jsou objevovány nové technologie a vyvíjeny jejich aplikace. Vezmeme-li například průmyslový robotický 6-osý manipulátor, tak pro jeho správnou a bezporuchovou funkčnost je nezbytné, aby tento stroj řídil dobře napsaný program. Celá automatizace pak skvěle funguje až do doby první kolize robotického manipulátoru, kdy je nezbytné nástroj, který tento robot drží znovu nakalibrovat. Pokud se jedná o obrovskou firmu, tak to není takový problém. Seřizovač pomocí drahého přípravku nakalibruje manuálně nebo automaticky robotický nástroj a automatizace za pár minut opět běží. V případě menší firmy, která si nemůže tyto technologie dovolit, musí seřizovač vše provést manuálně a přesnost provedené práce je přímo úměrná jeho zručnosti.

Tato skutečnost mě vedla k tomu, abych se zamyslel a pokusil se přijít s řešením, které bude dostatečným kompromisem jak z pohledu ceny, tak z pohledu kvality. Tato práce se tedy bude zabývat návrhem kalibračního přípravku, který zvládne automaticky nakalibrovat robotický nástroj za použití počítačového vidění tak, aby co nejpřesněji provedl kalibraci a tím vykryl odhalenou mezeru na trhu...

# 1 Prerekvizity Systému

V této části budou rozebrány všechny prerekvizity potřebné k pochopení postupů a řešení této práce.

## 1.1 Homogenní transformace

Homogenní transformace slouží pro přepočet kartézských souřadnic jednoho systému do druhého. V případě lineární transformace je měřítko (scale) rovno jedné. Jak je patrné z rovnice 1.2 pro transformatice 3-rozměrného bodu ze souřadnic $\boldsymbol{M}$ původního souřadnicového systému do nového souřadnicového systému $m$, tak výpočet je založen pouze na násobení matice původní s maticí rotace a translace $[R|t]$.[20]

$$m = [R|t]M \tag{1.1}$$

$$\begin{bmatrix} m_x \\ m_y \\ m_z \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} M_x \\ M_y \\ M_z \\ 1 \end{bmatrix} \tag{1.2}$$

## 1.2 Přepočet souřadnic robota

Pro zápis souřadnic robota se běžně používá několik způsobů. V této práci budu používat pro vyjádření polohy příruby (flange) kartézské souřadnice $x$,$y$ a $z$ a Eulerovy úhly $r_x$,$r_y$ a $r_z$ (tzv. yaw, pitch, and roll). Souřadnice flange jsou ve své podstatě vyjádřením rotace a translace flange vůči základně robota (robot base). V některých případech je výhodnější použít maticový zápis. Ten lze získat jednoduchým převodem, kdy jsou roznásobeny matice rotací 1.3, 1.4 a 1.5 a výsledná matice se rozšíří o vektor translace 1.7. Tím je získána Afinní transformační matice báze robota.[25]

$$\mathbf{R}_x(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{bmatrix} \tag{1.3}$$

$$\mathbf{R}_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{1.4}$$

$$\mathbf{R}_z(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1.5}$$

$$\mathbf{R} = \mathbf{R}_x(\psi) \cdot \mathbf{R}_y(\theta) \cdot \mathbf{R}_z(\phi) \tag{1.6}$$

$$\mathbf{t} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{1.7}$$

$$x, y, z, r_x, r_y, r_z = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix} \tag{1.8}$$

## 1.3 Optimalizace hejnem částic (PSO)

Particle swarm optimization (PSO) neboli Optimalizace hejnem částic je evoluční optimalizační technika, která je inspirována pohybem ptáků v hejnech. Svoje uplatnění nachází v úlohách, které nelze řešit analyticky. Výhodou této techniky je její jednoduchá implementovatelnost a vede převážně k rychlejší konvergenci oproti ostatním genetickým algoritmům.[22]

Mějme prohledávaný prostor, v kterém hledáme jeho globální minimum. Nejprve do tohoto prostoru umístíme hejno s náhodným rovnoměrným rozložením. Toto hejno je složeno z částic (tedy potenciálních řešení), kdy každá částice je definována svojí rychlostí $\vec{v}$, polohou $\vec{x}$ a svým nejlépe dosaženým výsledkem $\vec{p}$. V každém kroku iterace dochází k výpočtu nové rychlosti částice 1.9 a polohy 1.10 na základě předchozí polohy a rychlosti. Výpočet je ukončen v okamžiku ustálení nalezené nejlepší globální pozice nebo po dosažení maximální iterace.[23]

$$\vec{v}_i(t+1) = w \cdot \vec{v}_i(t) + rand(0, \varphi_1) \cdot (\vec{p}_i - \vec{x}_i(t)) + rand(0, \varphi_2) \cdot (\vec{p}_g - \vec{x}_i(t)) \tag{1.9}$$

Kde:

- $w$ - parametr utlumení rychlosti[1],
- $rand(a, b)$ - náhodné číslo v rozsahu $[a, b)$,
- $\varphi_1$ - parametr s jakou tendencí se bude částice vracet k její nejlepší pozici,
- $\vec{p}_i$ - nejlepší nalezená pozice částice $i$,

- $\varphi_2$ - parametr s jakou tendencí se bude částice vracet k nejlepší globální pozici,
- $\vec{p}_g$ - nejlepší globální nalezená pozice.

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \vec{v}_i(t) \tag{1.10}$$

## 1.4 Kalibrace kamer

Pro aplikace počítačového vidění je nezbytné před použitím kamer provést jejich kalibraci, díky které lze přepočítat jednotky a souřadnice obrazu na ty reálného světa a odstranit zkreslení vzniklé na čočce. Nutností pro tuto kalibraci je znalost matice vnitřních a vnějších parametrů kamery.

### 1.4.1 Dírkový model kamery

Pomocí matice vnitřních parametrů kamery 1.11, založené na vlastnostech ohniska lze přepočítat jednotky z obrazových souřadnic (nejčastěji vyjádřených v px) na reálná (např. v mm). Důležité je si uvědomit, že matice vnitřních parametrů nezávisí na poloze ani orientaci kamery.[6]

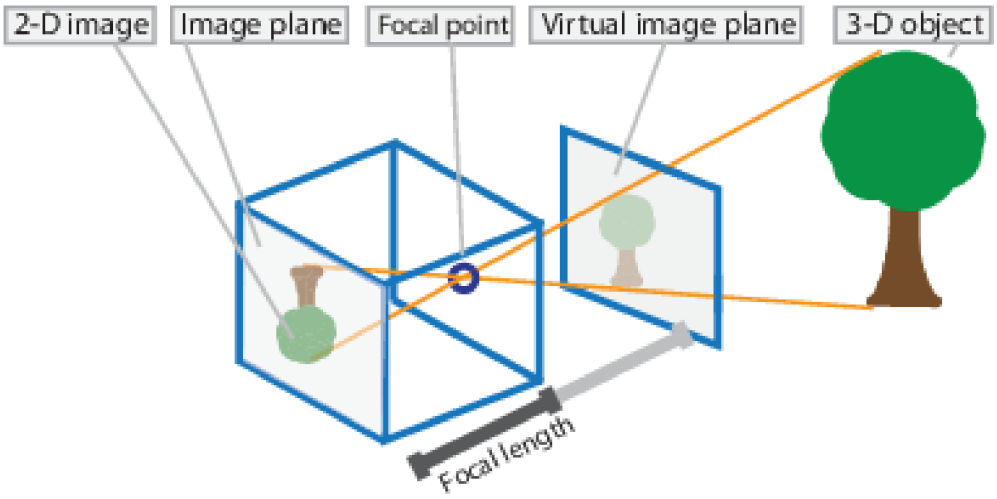
$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{1.11}$$

Kde:

- [x, y] - obrazové souřadnice projekce bodu,
- w - měřítko bodu (nejčastěji $w = 1$),
- $[f_x, f_y]$ - ohnisková vzdálenost,
- $[c_x, c_y]$ - prostorové souřadnice ohniska,
- [X, Y, Z] - Prostorové souřadnice bodu (3D).

Tento vztah vychází z modelu jednoduché kamery tzv. Dírkového modelu kamery (*Pinhole camera model*) obrázek 1.1 , který zanedbává distorzi nebo-li zkreslení obrazu na čočce oproti reálné kameře. Jednotlivé světelné paprsky s informací o 3D objektu prochází úzkou štěrbinou (ohniskem) a jsou promítány jako převrácený a zmenšený 2D obraz. Velikost obrazu je závislá jak na vzdálenosti kamery od scény, tak na velikosti ohniskové vzdálenosti.[6]

[1]Na počátku je doporučena hodnota $w \approx 0,9$, což má za následek prohledávání většího prostoru. S přibývajícím časem (iteracemi) dochází k snižování parametru, což způsobuje detailnější prohledávání okolí.




Obr. 1.1: Model dírkové kamery - výsledek promítání 3D objektu na obrazovou rovinu v závislosti na ohniskové vzdálenosti[3]

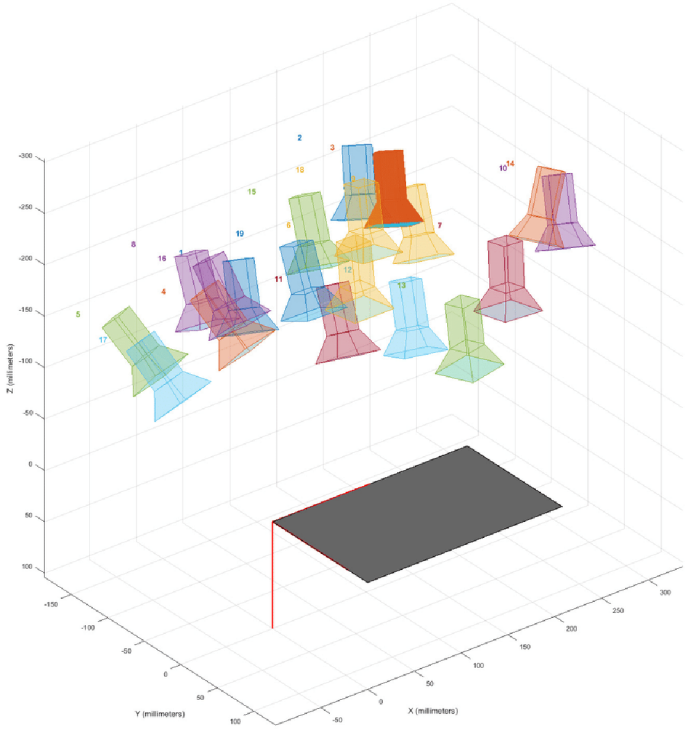
Matice vnějších parametrů kamery 1.12 je složena z matice rotace $R$ a vektoru translace $t$. Na základě této matice lze pro jednotlivé snímky určit reálné souřadnice kamery a její natočení vůči scéně nebo přepočíst souřadnice bodu na průmětně na souřadnice reálného světa.[7]

$$[R|t] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \tag{1.12}$$

$$sm' = A[R|t]M' \tag{1.13}$$

Spojením matic vnitřních a vnějších parametrů lze při volbě parametru $s$ rovnému jedna, podle rovnice 1.13 za pomoci homogenní transformace transformovat souřadnice obrazu na souřadnice reálného světa. Výsledek sekvence totožného objektu z různých poloh a natočení lze vidět na obrázku 1.2, který ukazuje právě využití této matice v praxi. Jak lze vidět, tak pouze na základě této matice lze pozorovat pohyb po prostoru při statické scéně.[6]

[3]Převzato z: `https://www.mathworks.com/help/vision/ug/camera-calibration.html/#bu1ahkv`




Obr. 1.2: Orientace kamery v prostoru - ilustrační zobrazení pozic kamery vůči šachovnici získaných kalibrací kamery[5]

### 1.4.2 Distorze čoček

Až doteď, stačilo zohledňovat pouze model ideální dírkové kamery. Proto, aby bylo možno korektně transformovat reálné souřadnice je zapotřebí u reálné kamery provést korekci distorze čoček. Při pořízení snímku reálnou kamerou, dochází k "vyboulení" obrazu, to je největší při pořízení snímku objektivem typu rybí oko, kdy dochází k radiální deformaci. K tangenciální deformaci dochází v případě, kdy čočka snímající obraz není dokonale rovnoběžná se zobrazovací rovinou. K odstranění těchto deformací pomocí rovnic 1.14 a 1.15 je nezbytná znalost vektoru Distorze 1.16, kde koeficienty $k_1$, $k_2$ a $k_3$ jsou koeficienty pro korekci radiální deformace a koeficienty $p_1$ a $p_2$ slouží pro korekci tangenciální deformace. Pro stanovení těchto koeficientů se nejčastěji využívá Matlabu nebo open-source knihovny OpenCV, která má algoritmy pro získání vektoru Distorze 1.16 již implementovány.[7]

[5]Převzato z: `https://www.researchgate.net/publication/323782586_Technical_note_A_simple_approach_for_efficient_collection_of_field_reference_data_for_calibrating_remote_sensing_mapping_of_northern_wetlands/figures?lo=1`

Korekce Radiální deformace:

$$\begin{aligned} x_{corrected} &= x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \\ y_{corrected} &= y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \end{aligned} \tag{1.14}$$

Korekce Tangenciální deformace:

$$\begin{aligned} x_{corrected} &= x + [2p_1 xy + p_2(r^2 + 2x^2)] \\ y_{corrected} &= y + [p_1(r^2 + 2y^2) + 2p_2 xy] \end{aligned} \tag{1.15}$$

$$Distortion = \begin{pmatrix} k_1 & k_2 & p_1 & p_2 & k_3 \end{pmatrix} \tag{1.16}$$

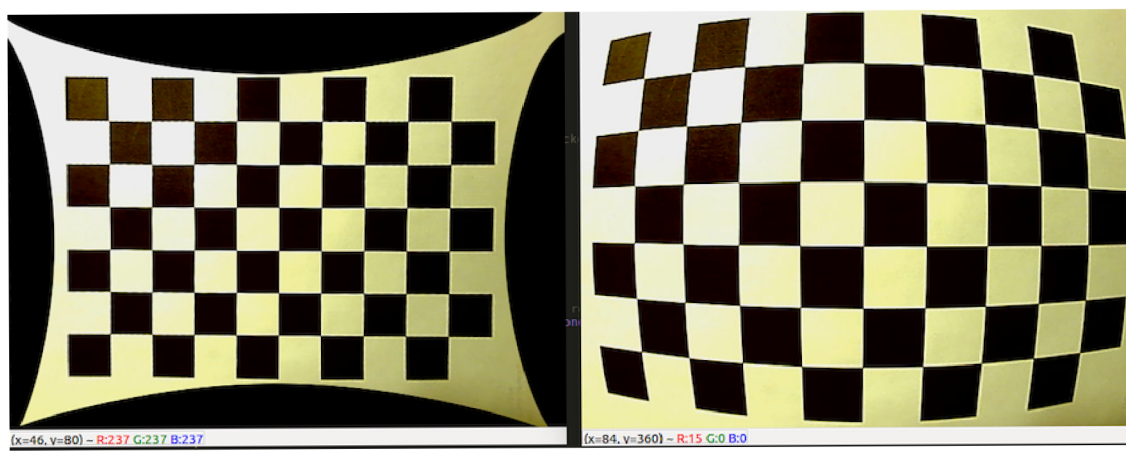
### 1.4.3 OpenCV

OpenCV (Open source computer vision) je knihovna algoritmů počítačového vidění určených pro všechny dnes běžně používané platformy. V dnešní době je nejvíce využívanou knihovnou zaměřenou na počítačové vidění napříč všemi odvětvími, a proto je v této práci využita.[8]

V dokumentaci OpenCV je uváděno, že pro dobrou kalibraci je potřeba minimálně 10 snímků. Já doporučuji přibližně 20-30 snímků s různým natočením a polohou šachovnice vůči kameře. Samotná kalibrace bude prováděna pomocí upraveného vzorového skriptu *camera_calibration.cpp*. Výsledkem bude matice vnitřních parametrů kamery 1.11, deformační vektor 1.16 a chyba reprojekce.[7]

Na obrázku 1.3 lze vidět ilustrativní výsledek provedení kalibrace kamery. V pravé části se nachází originální obraz scény a nalevo je tentýž snímek, ale už po korekci parametry, jež byly získány kalibrací kamery.

[7]Převzato z: `https://curticapean.com/projects/6370465`




Obr. 1.3: Distorze čočky a její odstranění - Vpravo obrázek původní a vlevo narovnaný pomocí distorzních parametrů získaných kalibrací kamery[7]

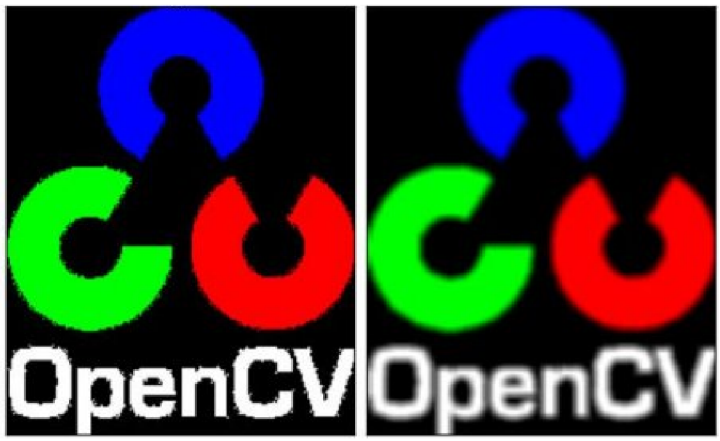
# 2 Metody počítačového vidění

V této kapitole budou popsány používané metody počítačového vidění, které slouží k zpracování obrazu a jsou stěžejní částí detekce koncového bodu robotického nástroje v obraze.

## 2.1 Filtrace šumu

Je spolu s detekcí hran jednou ze základních operací využívaných v počítačovém vidění. Šum jsou ty části obrazu, které nenesou žádnou užitečnou informaci a způsobují náhodnou změnu obrazové funkce. Do obrazu jej dostáváme především při kvantizaci a kompresi obrazu, ambientním osvětlením a šumem samotného senzoru. Šum může být aditivní, multiplikativní, typu "sůl a pepř", Gaussovský, Poissonovský, bílý nebo kvantizační. Potlačit jej lze Časovou filtrací, tedy násobným zachycením scény a následným zprůměrováním nebo za pomocí mediánu všech pořízených snímků. V případě, že je k dispozici pouze jeden snímek, je aplikována jedna z metod Prostorové filtrace:

- Konvoluční maska
- Rotující maska
- Medián




Obr. 2.1: Ukázka filtrace šumu - nalevo zašuměný obrázek a napravo rozostřený pomocí aplikace konvoluce, která průměruje okolí 3x3[2]

Předpokladem využití jedné z této metod je, že snímek obsahuje nadbytečné informace a hodnoty jasu sousedních pixelů se výrazně nemění. U výsledného obrazu pak dojde k vyhlazení (potlačení vyšších frekvencí) obrazu.[10]

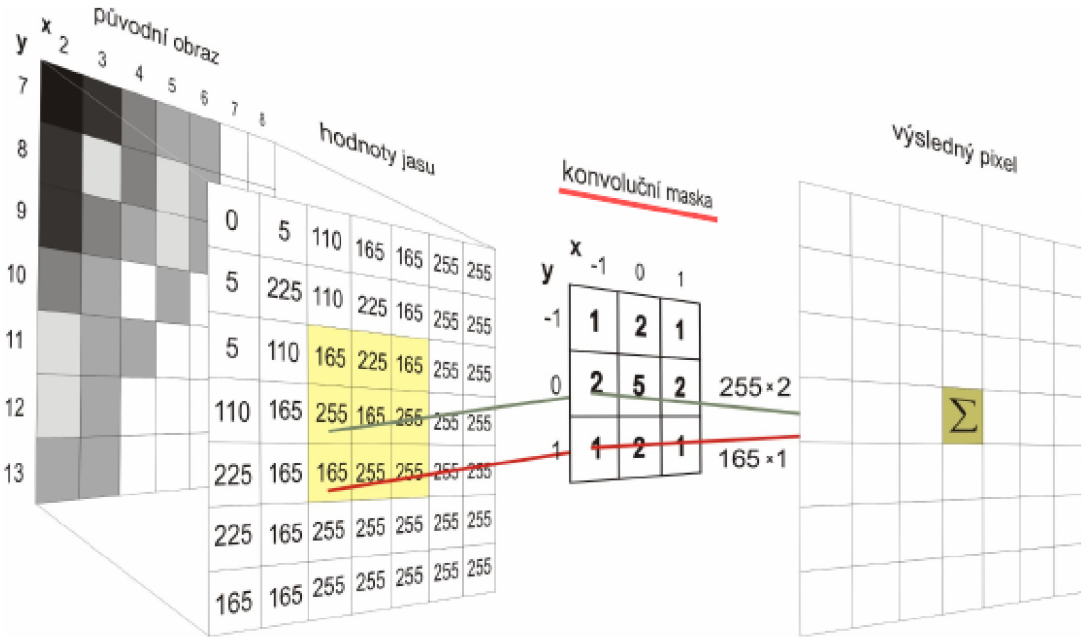
[2]Převzato z: `https://docs.opencv.org/master/d4/d13/tutorial_py_filtering.html`

### 2.1.1 Konvoluční maska

Konvoluční maska provádí v podstatě váhované průměrování libovolného okolí kolem zvoleného bodu (pixelu). Jak lze vidět na obrázku 2.2, konvoluce dle rovnice 2.1 pronásobí konvoluční masku s okolím zvoleného pixelu podle její velikosti, čímž následně získáme novou hodnotu zvoleného pixelu.

$$f(i,j) = \sum_{(m,n)\in O} h(i-m, j-n) * g(m,n) \tag{2.1}$$

Tento jednoduchý algoritmus je následně aplikován na celý snímek. Velikost filtru je libovolná, avšak běžně se volí čtvercová matice s lichým počtem řádků a sloupců, pro přesnou specifikaci bodu (pixelu), pro který tuto konvoluci provádíme. Důležitou podmínkou je, aby hodnota použité masky byla rovna jedné. V případě nedodržení této podmínky je výsledný obraz v případě hodnoty masky vyšší než jedna světlejší a naopak v případě hodnoty menší než jedna tmavší. Ve výsledném obraze tak sice je potlačený šum (vysoké frekvence), ale dochází také k jeho rozmazání (viz obrázek 2.1).[11]




Obr. 2.2: Ukázka výpočtu konvoluce - výsledný pixel vpravo je získán pronásobením konvoluční masky s původním pixelem a jeho okolím[4]

[4]Převzato z: `http://midas.uamt.feec.vutbr.cz/ZVS/Exercise06/content_cz.php`

## 2.2 Detekce hran

Detekce hran je jedna ze základních operací využívaných v počítačovém vidění. Matematicky lze hrany definovat jako maxima derivace jasové funkce. U počítačového vidění se určuje v obraze na místě, kde dochází k prudké změně jasu mezi sousedními pixely. V digitálním zpracování obrazu nelze nikdy docílit spojité funkce jasu obrazu vlivem kvantování a tím vzniká šum. Abychom mohli přejít k samotnému "hranování", musíme nejprve potlačit tento šum.

### 2.2.1 Cannyho hranový detektor

Tento algoritmus objevený roku 1986 Johnem F. Cannym byl oproti jeho současným algoritmům velice robustní a to hned z několika pohledů. Pro hranové detektory jsou to tyto tři:

- Minimální počet chyb
  - Cílem je co největší odstup signálu od šumu. Musí být detekovány všechny hrany a nesmí být detekovány místa, které hranami nejsou.
- Přesnost
  - Cílem je co nejpřesnější lokalizace hrany v obraze.
- Jednoznačnost
  - Cílem je zamezit zdvojené detekci hrany.

Algoritmus tohoto detektoru se skládá ze čtyř kroků. Jen jejich sestupným provedením je docíleno robustní detekce hran (viz obrázek 2.3). Jednotlivé fáze jsou:

- Eliminace šumu
- Určení gradientu
- Nalezení lokálních maxim
- Eliminace nevýznamných hran

Jednotlivé kroky algoritmu si níže stručně rozebereme.[12]

#### Eliminace šumu - Gaussův filtr

Gaussův filtr vychází z Gaussovy funkce 2.2, pomocí které je vytvořeno konvoluční jádro. Jediným parametrem funkce je směrodatná odchylka $\sigma$, která určuje poloměr rozostření v pixelech. Vícenásobná aplikace konvoluční masky odpovídá aplikaci jediné masky s větším poloměrem rozostření, který je roven odmocnině součtu druhých mocnin poloměrů rozostření aplikovaných vícenásobných masek. Postup aplikace konvoluční masky byl popsán dříve v kapitole 2.1.1.[11]

[6]Převzato z: `https://ai.stanford.edu/~syyeung/cvweb/Pictures1/canny3.png`

Obr. 2.3: Ukázka aplikace Cannyho hranového detektoru - vlevo původní obraz a vpravo výsledný pomocí aplikace detektoru[6]

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2.2}$$

### Určení gradientu - Aproximace 1. derivace

V tomto kroce může být aplikován kterýkoliv ze standartních operátorů pro detekci hran a to především Sobelův operátor pro svoji robustnost vůči šumu a schopnosti stanovit nejen velikost gradientu 2.3, ale i jeho směr 2.4. V praxi se také používá operátor 1. derivace Gaussovské funkce.[12]

$$G(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2} \tag{2.3}$$

$$\theta(x, y) = \tan^{-1} \frac{G_y(x, y)}{G_x(x, y)} \tag{2.4}$$

Pro stanovení gradientu je využíváno rovnice 2.3. Stanovení gradientu v horizontálním i vertikálním směru probíhá odděleně. Rovnici lze zjednodušit pouhým sečtením absolutních hodnot těchto jednosměrových gradientů, avšak v některých případech může následně docházet k zhoršení výsledků.[11]

Při aplikaci Sobelova operátoru je výsledný směr gradientu vzhledem k vodorovnému směru definován rovnicí 2.4. V případě použití 1. derivace Gaussovské funkce se vychází z toho, že pro eliminaci šumu je využito také Gaussovské funkce. Díky

tomu lze následně prohodit pořadí konvoluce a derivace, protože obě tyto operace jsou lineární, takže stačí provést pouze jednu konvoluci, a to 1. derivací Gaussovské funkce 2.5. Tím se zjednoduší výpočet dílčích horizontálních i vertikálních gradientů. Výsledný gradient je pak získán totožně jako u Sobela pomocí rovnice 2.3 nebo jejím zjednodušením.[11]

$$G'(x) = (e^{-\frac{x^2}{2\sigma^2}})' = (-\frac{x}{\sigma^2})e^{-\frac{x^2}{2\sigma^2}} \tag{2.5}$$

Při použití konvoluce u detekce hran neplatí ta podmínka, že hodnota masky se musí rovnat 1 (jako při eliminaci šumu), ale hodnota tohoto jádra musí být rovna 0. Díky tomu je zajištěno, že pro regiony s konstantní hodnotou bude výsledná konvoluce nulová.[11]

#### Nalezení lokálních maxim - Thinning

Tento krok se stará o "ztenčení" hran. Cílem je odebrat všechny
NMS (Non-maximum suppression) body, které nejsou lokálním maximem. Je-li některý pixel označený za lokální maximum, znamená to, že hodnota jeho gradientu je ve směru i proti směru gradientu jeho souseda vyšší. Dva sousední pixely, které jsou porovnávány jsou dány vždy směrem gradientu zvoleného pixelu.[12]

#### Eliminace nevýznamných hran - Thresholding

V posledním kroku zbývá už jen eliminovat ty hrany, které nejsou významné. To je prováděno tzv. Prahováním. Běžně se používá buď globální prahování (global thresholding) nebo lépe prahování s hysterezí (hysteresis thresholding).[11]

Prahování s hysterezí nevyužívá pouze jedné prahové hodnoty, ale rovnou dvou minimální práh $T_1$ a maximální práh $T_2$. Dosáhne-li hodnota gradientu označeného pixelu hodnoty vyšší než $T_2$ potom je automaticky pixel označen jako hranový. V případě, že se hodnota gradientu nachází v mezích $T_1$ a $T_2$, je pixel označen za hranu jen tehdy, byl-li za hranu označen i jeho sousední pixel.[12]

## 2.3 Morfologické operace

Matematická morfologie je teorií k analýze plošných a prostorových struktur. Je nejčastěji využívána k analyzování tvarů objektu, kde se uplatňuje díky své triviálnosti. Původní teorie George Matherona a Jeana Serry z roku 1964 byla určena pro analýzu binárních obrazů, později se dočkala zobecněna i pro analýzu šedotónových obrazů. Základní operace binární morfologie jsou eroze a dilatace. [21]

### 2.3.1 Dilatace (Dilation)

Dilatace je využívána pro schopnost zaplnění malých děr a úzkých zálivů v objektech. Její nevýhodou je zvětšení velikosti objektu na úkor pozadí. Dilataci lze ekvivalentně vyjádřit jako funkci $\delta$.[21]

$$\delta_B(\mathbf{X}) = \mathbf{X} \oplus \mathbf{B} = p \in \mathbb{E}^2 : p = x + b,\ x \in \mathbf{X},\ b \in \mathbf{B}$$

Obr. 2.4: Ukázka dilatace - Vlevo původní obraz a napravo výsledek po aplikaci dilatace[8]

### 2.3.2 Eroze(Erosion)

Eroze je duální operace k dilataci. Nejčastěji se používá k rozložení objektu na jednodušší části - zjednodušení. Erozi lze ekvivalentně vyjádřit jako funkci $\epsilon$ . Výhodou je, že zachovává velikost objektu.[21]

$$\epsilon_B(\mathbf{X}) = \mathbf{X} \ominus \mathbf{B} = p \in \mathbb{E}^2 : p + b \in \mathbf{X}\, pro\, \forall b \in \mathbf{B}$$

Obr. 2.5: Ukázka eroze - vlevo původní obraz a napravo výsledek po aplikaci eroze[10]

[8]Převzato z: `https://docs.opencv.org/trunk/d9/d61/tutorial_py_morphological_ops.html`

Eroze je často používaná ve spojení s dilatací. Binární otevření - eroze je následována dilatací, nebo binární uzavření - dilatace je následována erozí. Obě metody jsou invariantní vůči posunu.

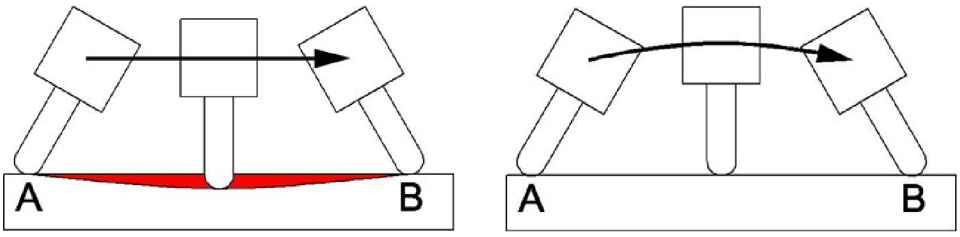
[10]Převzato z: https://docs.opencv.org/trunk/d9/d61/tutorial_py_morphological_ops.html

# 3 Metody kalibrací robotického nástroje

V této kapitole budou popsány nejčastěji používané metody kalibrace pracovního bodu nástroje se samotnou definicí tohoto pojmu a vysvětlením určení samotného TCP nástroje.

## 3.1 Tool Center Point

Středový bod nástroje (TCP) je definován pomocí vektoru (matice translace a rotace), který se nachází mezi přírubou robota a "hrotem"koncového efektoru. Jeho hlavním úkolem je určit počátek tohoto efektoru. Ten je nezbytný pro nespočet průmyslových aplikací, kde je potřeba s efektorem manipulovat po definované ploše (např. svařování, obrábění, atd.).[4]




Obr. 3.1: Výsledek TCP kalibrace - vlevo špatně a vpravo správně nakalibrovaného TCP (koncového bodu nástroje)[2]

Tento problém je demonstrován na obrázku 3.1, kde je definovaný pohyb efektoru po ploše rovného a hladkého kvádru z bodu A do bodu B. Vlevo vidíme příklad špatné TCP kalibrace, která sice provede lineární pohyb, ale nezohledňuje geometrické vlastnosti nástroje a vzniká kolize. Vpravo lze vidět tento pohyb se správně kalibrovaným nástrojem, kdy se hrot tohoto nástroje se pohybuje plynule po povrchu kvádru. Nezarývá se do povrchu, ačkoliv pohyb robota není lineární vůči pracovnímu prostoru robota, ale je lineární vůči požadované ploše. To je hlavní důvod použití TCP kalibrací.

V následujících podkapitolách budou probrány metody průmyslově běžně používaných TCP kalibrací.

[2]Převzato z: `https://www.fanucamerica.com/products/cnc/cnc-technology/tool-center-point-control`

## 3.2 Manuální kalibrace

Manuální kalibrace je nejstarší a nejjednodušší metodou TCP kalibrace. Tato metoda je časově nejnáročnější a nejméně přesná ze všech známých metod. Její reprodukovatelnost je takřka nulová, protože je prováděna přímo operátorem, který navádí robotický manipulátor na zvolený bod z různých směrů pod různými úhly. Přesnost této kalibrace je tedy založena pouze na zručnosti operátora.

Před samotnou kalibrací se připevní na přírubu robotického manipulátoru hrot. Další hrot je umístěný v pracovním prostoru robota. Nyní několikrát operátor přiblíží hroty co nejblíže proti sobě. Souřadnice ideální polohy uloží a následně oddálí robota. Tento krok provádí opětovně avšak pod jiným úhlem a z jiného směru. Ačkoliv tato metoda není moc přesná, ve spoustě aplikací je dostačující. Mezi její další výhodu patří cena, která stojí pouze čas operátora.[1]

Obr. 3.2: Manuální TCP kalibrace - kalibrace hrotem připevněným na přírubě robota vůči kalibračnímu hrotu[4]

V případě, že již v minulosti byla kalibrace provedena, lze pro re-kalibraci použít tzv. Referenční metodu. Ta se nejčastěji používá po kolizi nástroje nebo jeho výměně (např. při výměně svařovací trysky za jiný typ). Nejprve se najede robotickým manipulátorem se starou tryskou na pracovní bod nástroje (TCP) a tyto souřadnice se uloží. Poté se robotický manipulátor oddálí, tryska je vyměněna za novou a operátor od stejným úhlem navede robota na pomocný hrot. Souřadnice se uloží a je přepočítán nový pracovní bod nástroje.[2]

[4]Převzato z: `https://www.youtube.com/watch?v=CExtMfvRabo`

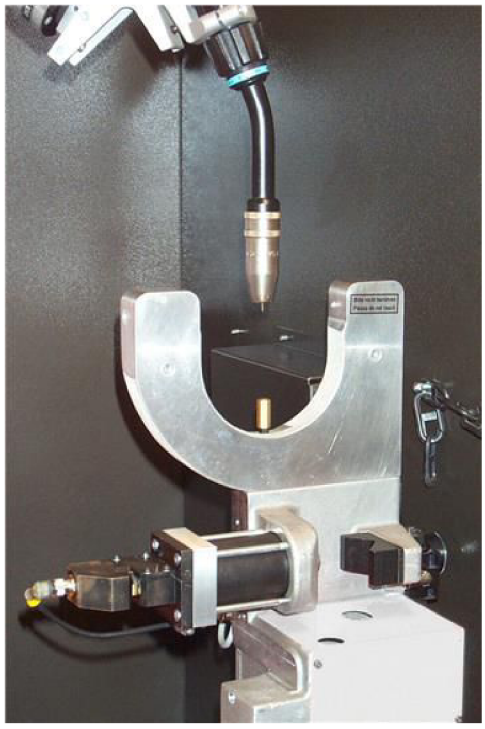
Tyto manuální metody jsou vetšinou integrovány již výrobci robotických manipulátorů, takže nepředstavují žádné dodatečné pořizovací náklady proti kalibracím automatickým.

## 3.3 Automatická kalibrace

Tyto kalibrace jsou založeny většinou na použití laserů. Pro všechny tyto kalibrace je společnou vlastností to, že kalibrují osově soustředné předměty podél jedné osy a mají vysokou pořizovací cenu.

### 3.3.1 BullsEye

Je automatickou kalibrací od společnosti ABB primárně určenou pro jejich průmyslové roboty. Výstupem této kalibrace jsou souřadnice pracovního bodu nástroje a jeho natočení, které jsou určeny vzhledem k přírubě robota. Tato kalibrace je vhodná pouze pro nástroje, které jsou soustředné podél jedné osy (např. svařovací tryska).[3]



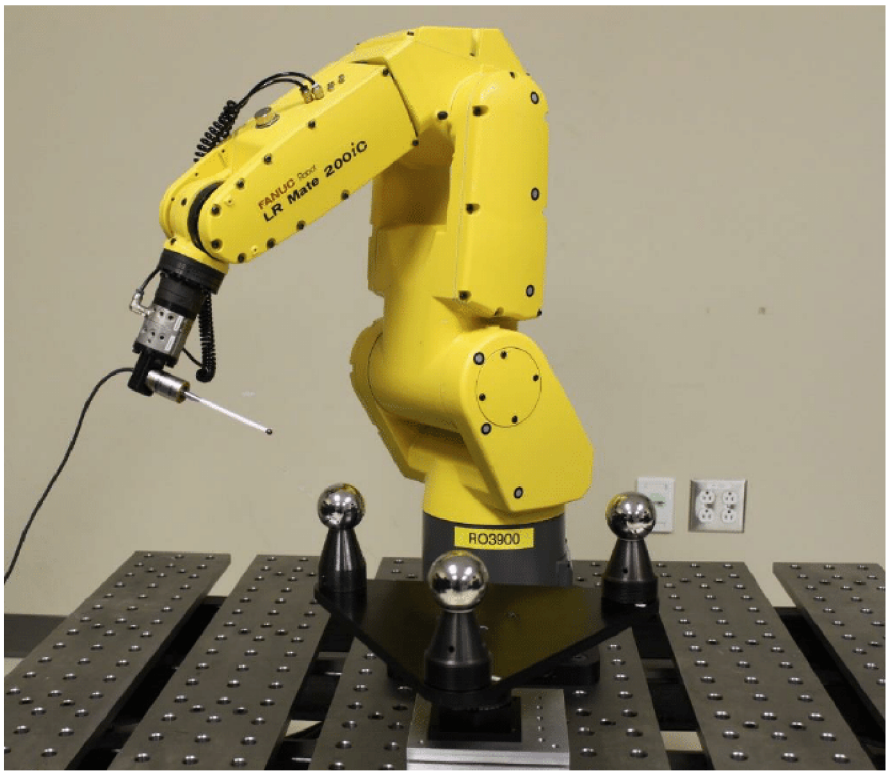
Obr. 3.3: BullsEye kalibrace - ukázka kalibrace svařovací trysky[5]

[5]Převzato z: https://www.directindustry.es/prod/abb-robotics/product-30265-2026362.html

Metoda je postavena na použití laseru, kdy robot pohybem protíná tento paprsek a hledá osu nástroje spolu s jeho koncovým bodem (špičkou), jak lze vidět na obrázku 3.3. Z toho vyplývá další omezení, kdy tato kalibrace není schopna nalézt TCP nástroje sférického tvaru. To lze obejít při znalosti geometrických vlastností tohoto nástroje, kdy souřadnice $X$ a $Y$ definují osu nástroje a $Z$ definuje střed špičky.[1]

### 3.3.2 Kalibrace na základě dotekové síly

Zde se nabízí dva možné přístupy. První přístup využívá integrované snímače síly (přetížení) os robota. Druhý způsob využívá externí snímače umístěné na kalibračním přípravku na které robot najíždí nebo je na robotické rameno připevněna dotyková sonda a je aplikován podobný postup jako v prvním případě.




Obr. 3.4: Touch probe kalibrace - Ukázka kalibrace touto metodou[6]

Pro první přístup je v pracovním prostoru před robotem umístěn kalibrační předmět známých rozměrů (většinou kovová krychle). Robot se pomalu přibližuje k této krychli a neustále si kontroluje své snímače přetížení. V okamžiku, kdy některý

[6]Převzato z: https://www.researchgate.net/publication/267391760_Kinematic_calibration_of_a_six-axis_serial_robot_using_distance_and_sphere_constraints/figures?lo=1

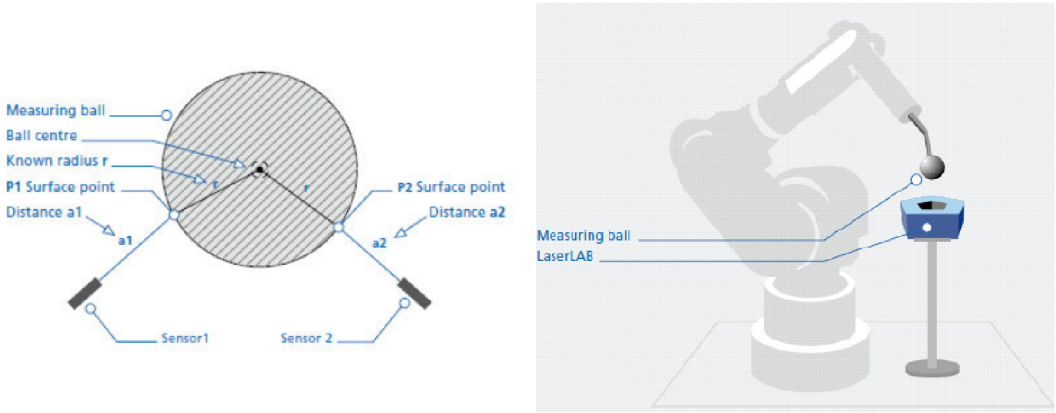
z těchto snímačů hlásí přetížení předem nastavené jako kontakt, program zastaví pohyb robota a uloží souřadnice polohy kontaktu. Následně se robotické rameno oddálí a přijíždí se k předmětu z jiné strany nebo pod jiným úhlem. Tento postup je opakován v závislosti na počtu dimenzí robotického nástroje.[1]

Při aplikaci externího snímače je postup téměř totožný jako v prvním případě, až na ten rozdíl, že robot je zastavován externími signály z kalibračního přípravku a vyžaduje externí řídící jednotku. Jak lze vidět na obrázku 3.4, robot s dotykovou sondou se přibližuje ke kouli pod stanoveným úhlem z definovaného směru až do okamžiku kdy dojde ke kontaktu. Pozice robota je zaznamenána a daný krok se provede s jinými parametry. Na základě těchto měření následně externí jednotka optimalizuje TCP robota.[1]

Tato metoda je využívána především u dokončovacích procesů jako je broušení nebo leštění. Výhodou je také vysoká přesnost, díky které je možné definovat TCP s přesností až $\pm 0,05 - 0,015$mm. Nevýhodou je pořizovací cena a nutnost dalšího hardwaru. V prvním případě je nezbytné, aby robotické rameno mělo předpřipravené nástroje pro tento typ kalibrace.[1]

### 3.3.3 LaserLab

Další metoda využívající laserovou technologii. Kalibrační přípravek je sestaven z kalibrační koule připevněné na přírubu robota a měřící jednotky osazené pěti na sobě nezávislými lasery s každým v jednom z rohů pravidelného pětiúhelníku tak, aby se všechny lasery protly v jednom společném bodě. Aplikace vyžaduje dodatečnou řídící jednotku v podobě PC.[1]




Obr. 3.5: LaserLAB kalibrace - vlevo obecné schéma přípravku a vpravo ukázka kalibrace[8]

[8]Převzato z: `http://publications.lib.chalmers.se/records/fulltext/153281.pdf`

Program navede kouli k přípravku na střed společného bodu laserů, aby bylo možné provádět měření vychýleni této koule ze společného bodu laserů, jak je demonstrováno na obrázku 3.5. Robotem je pohybováno a natáčeno v různých směrech tak, aby bylo docíleno co nejpřesnějšího určení TCP. Pokud je při přesunu do nové pozice koule vychýlena (změní se délka paprsku laseru), dochází ke korekci TCP až do doby, kdy je provedena celá testovací sekvence bez vychýlení koule z kalibračního bodu přípravku. Tato metoda nabízí rychlou a velmi přesnou kalibraci s přesností až $\pm 0,1$mm.[1]

### 3.3.4 State of The Art - LEONI

Nejsofistikovanějším řešením na trhu je v současné době přípravek německé společnosti LEONI, který nabízí kalibraci pro 3, 5 a 6 dimenzionální nástroj s přesností kalibrace až $\pm 0,02$mm za dobu začínající už na 3s. Řešení spočívá v případě 6D kalibrace v umístění n přesných laserových snímačů na kalibrační přípravek tak, aby co nejlépe detekovaly hrany přípravku. Robotem je následně pohybováno a natáčeno nad tímto přípravkem. Na základě naměřených rozměrů nástroje robotického ramene je určen jeho pracovní bod. Ovládání robotického ramene i vyhodnocování souřadnic TCP robota probíhá na uzavřené řídící jednotce této společnosti a výsledek je následně předán kontroléru robota pomocí Profibus sběrnice.[5]

## 3.4 Shrnutí

V kapitole byla objasněna nejběžněji používaná metoda kalibrace a to tzv. Point-counterpoint manuální metoda. Z automatických kalibrací jsou zde dvě nejrozšířenější metody a to ty, které využívají k měření jeden nebo více laserů a ty, které měří dotekovou sílu pomocí externích nebo vestavěných snímačů síly. Jako poslední je popsána State of the Art kalibrace až 6D nástrojů a to pomocí laserů. Aplikace je schopna provést TCP kalibraci jednoduchého nástroje za čas okolo 3s s přesnosti $\pm 0,02$mm.

[10] Převzato z: `https://www.leoni-factory-automation.com/en/products-and-services/unracking/gripper-measurement/`

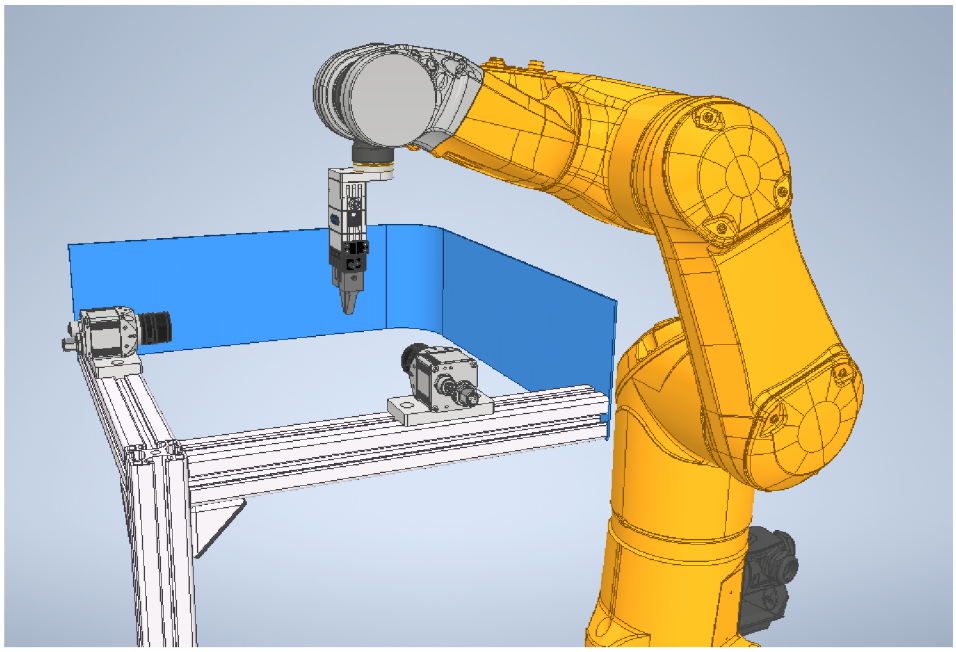
Obr. 3.6: LEONI 6D laserová kalibrace - ukázka State of The Art kalibrace robotického nástroje[10]

# 4 Návrh Pracoviště

V této kapitole bude podrobně specifikován testovací vzorek pro automatickou kalibraci nastroje robotického manipulátoru včetně použité robotické platformy a zvoleného hardware pro realizaci tohoto zařízení.

## 4.1 Návrh testovacího zařízení

Na navržený testovací vzorek (příloha A) byl od počátku kladen zvýšený důrazu na jeho univerzální využití pro všechny platformy robotických ramen různých standardních velikostí a jeho kompaktnost. Cílový typ robotického chapadla, pro který je přípravek navrhnut, je dvouprsté chapadlo, které je nejčastěji používáno v aplikacích zaměřených na bin picking (vybírání zásobníku).



Obr. 4.1: Návrh konstrukce kalibračního vzorku

Přípravek sestává ze tří hliníkových V-slot Profilů, které při pohledu z perspektivy tvoří písmeno 'Y' připevněné na kovový podstavec pro stabilitu přípravku. Díky použití tohoto profilu je docíleno vysoké flexibility z pohledu nastavení výšky přípravku a umístění kamer na odvěsnách pravoúhlého trojúhelníku, který je imaginárně tvořen pomocí kamer a vrchní části přípravku. Pro upevnění kamer k profilu je využito speciálních podložek, které zajišťují jejich posun právě v ose profilu. Na přípravek lze též připevnit odnímatelný plastový kryt, pomocí kterého je docíleno monotónního pozadí při snímání pro vyšší přesnost a robustnost kalibrace.

### 4.1.1 Konstrukce

Pro konstrukci byl zvolen hliníkový V-slot Profil s modulárním upínacím systémem, pomocí kterého je docíleno variabilnosti a kompaktnosti přípravku. Umožňuje jak laboratorní použití, tak i možné následné využití v průmyslu.

Obr. 4.2: Realizace kalibračního vzorku - perspektivní pohled na celé pracoviště

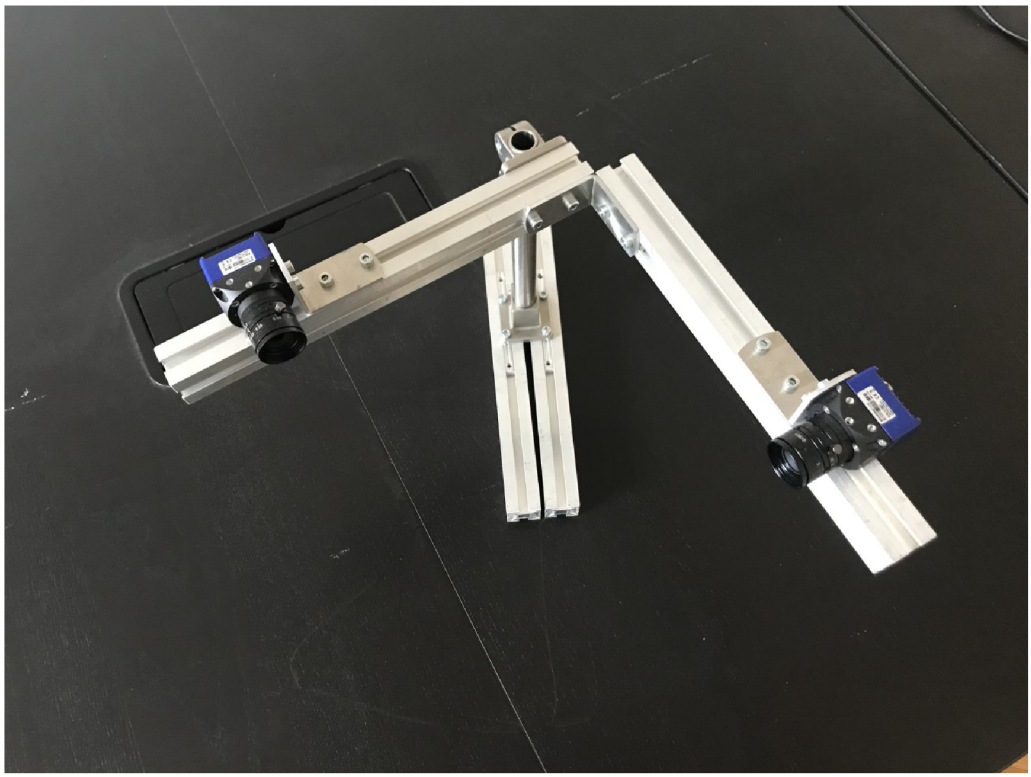
### 4.1.2 Realizace

Realizace (viz obrázky 4.2 a 4.3) byla vyhotovena podle návrhu až na odnímatelný plastový kryt, který měl zajišťovat statické pozadí. Snímky výsledné realizace jsou součástí CD přílohy.

## 4.2 Robotická platforma

Jako robotická platforma byl zvolen 6-osý robotický manipulátor od Stäubli řady TX2. Tato volba byla učiněna firmou, která zajišťuje téma této diplomové práce. Důvodem volby této platformy je jak užší spolupráce s výrobcem Stäubli, tak především vysoká opakovatelnsot pohybu (pro TX2-90) s přesností na $\pm 0,03$mm, které je docíleno díky Stäubli patentované převodovce JCS smart gearbox. Díky tomu jsou tyto robotické manipulátory používány v aplikacích, kde je kladen zvýšený důraz na přesnost.[13]

[1]Převzato z: `https://www.allrobots.cz/produkty/staubli/tx2-90/`



Obr. 4.3: Realizace kalibračního vzorku - horní pohled na kalibrační přípravek

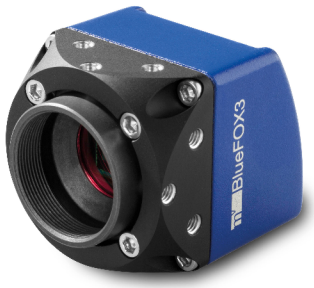
Obr. 4.4: 6-osý robotický manipulátor Stäubli TX2-90[1]

## 4.3 Použitý hardware

Zde je odůvodněna volba použitého hardwaru se specifikací důležitých parametrů každé komponenty.

### 4.3.1 Kamery

Pro aplikaci byly vybrány dvě průmyslové kamery Matrix Vision mvBlueFOX3-2032aC. Typ snímače je CMOS 1/1.8"s rozlišením 2048 x 1536px a velikostí pixelu 3,45x3,45$\mu$m. Rozhraní bylo zvoleno USB3.0. Napájení kamer je zajištěno pomocí USB připojení.[14]




Obr. 4.5: MATRIX VISION BlueFox3 USB3 Vision Camera[2]

### 4.3.2 Objektivy

Pro výpočet objektivu jsem vycházeno z parametrů použitého snímače u zvolené kamery Matrix Vision mvBlueFOX3-2032aC, který je SONY Sony IMX265 s velikostí snímače 1/1.8", rozlišením 2048 x 1536px a velikostí pixelu 3,45x3,45$\mu$m. Pracovní vzdálenost od objektu je $WD = 300mm$ a horizontální FOV je $FOV_h = 200mm$. Pro výpočet ohniskové vzdálenosti je využito rovnice 4.2, pro jejíž použití je nezbytná znalost horizontální velikosti snímače $sensor_h$, která je stanovena pomocí pomocného výpočtu 4.1, kde je vynásobena šířka pixelu s horizontálním rozlišením.[15]

$$sensor_h = 3.45 \cdot 10^{-3} \cdot 2048 = 7,0656mm \tag{4.1}$$

$$f \cdot FOV_h = sensor_h \cdot WD \tag{4.2}$$

[2]Převzato z: `https://www.bhphotovideo.com/c/product/1318418-REG/matrix_vision_rt_mvbf3_2124ag_12_4mp_1_1_4112_x.html`

$$f = \frac{sensor_h \cdot WD)}{FOV_h} = \frac{7,0656 \cdot 300}{200} = 10,6mm$$

Dle vypočtené ohniskové vzdálenosti byly zvoleny objektivy VS-LDA10 F2,3-2,4 s ohniskovou vzdálenosti $10mm$ a pracovní vzdálenosti $330,2 - 91,0mm$.[16]

### 4.3.3 Svítidlo

Jsou použity dva studiové zdroje daylight světla. V každém softboxu je za difuzní fólií umístěna žárovka o výkonu 85W, čímž je docíleno rovnoměrného světla pro nasvícení scény.

Do budoucna se počítá s navržením a vyrobením kruhového makro svítidla opatřeného difuzní fólií pro potlačení odlesků na lesklých částech nástroje. Toto svítidlo bude umístěno na objektivu kamery, čímž bude docíleno kompaktnosti řešení.

### 4.3.4 Výpočetní jednotka

Přípravek je navržen tak, aby podporoval všechny běžně dostupné počítače s podporou USB3.0 a Ethernet portem RJ-45. Ostatní volba hardwaru by měla ovlivnit především rychlost výpočtu kalibrace. Jediný podporovaný operační systém je Linux (doporučený kernel 5.3.0 a vyšší).

### 4.3.5 Rozhraní

Za rozhraní byl zvolen Ethernet a USB3.0 z důvodů univerzálního použití. Všechny komponenty byly vybrány nebo navrženy tak, aby podporovaly toto rozhraní.

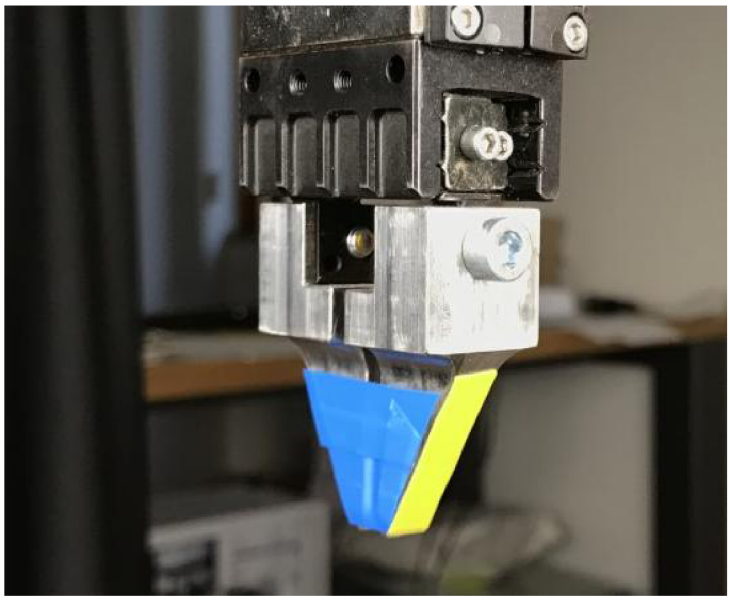
S kamerami dochází ke komunikaci pomocí USB3.0, pomocí mvIMPACT Acquire SDK, nad který jsem vytvořil wrapper. S robotem dochází ke komunikaci pomocí TCP protokolu přes Ethernet, kdy je využíváno na straně robota firemního API společnosti SANEZOO, pod jejíž záštitou byla tato práce vytvořena.

### 4.3.6 Označení chapadla robota

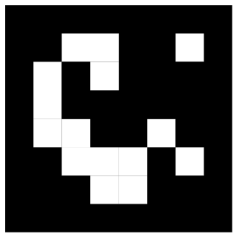
Pro kalibraci robota je nezbytné opatřit jeho plošky barevným nátěrem nebo fólií a to z obou detekovaných stran jinou kontrastní barvou (viz obrázek 4.6), což následně umožní detekci TCP. V případě, že chceme detekovat hrot, stačí pouze jedna barva pro přední a boční pohled.

### 4.3.7 Kalibrační marker

Pro nastavení kalibračního bodu a provedení kalibrace robot-kamera je nezbytné umístit na robota ArUco marker s id 21 ze slovníku 6x6 a o velikosti 25x25mm.



Obr. 4.6: Označené 2-prsté chapadlo - příklad označení chapadla



Obr. 4.7: ArUco marker

## 4.4 Shrnutí

V kapitole je popsán a odůvodněn návrh a materiály použité k realizaci přípravku a celého stanoviště, spolu s použitým hardwarem a objasněním jeho volby. U objektivů je tato volba doložena i výpočtem. Také je zde ukázána možná varianta označení plošek nástroje a ArUco marker využívaný k potřebám dílčích kalibrací.

# 5 Implementace

V této části bude popsán nejprve algoritmus automatické kalibrace obecně a následně budou popsány stěžejní části aplikace samostatně spolu s uživatelským rozhraním a vším co nabízí. Navrhnutá kalibrační aplikace je založena na principech manuální kalibrace (kapitola 3.2), kdy operátor je nahrazen dvěma kamerami. Nevýhodou kalibrace je to, že se nedokáže vypořádat s natočením TCP. Poskytuje informaci pouze o jeho pozici s ose x,y a z vůči přírubě (flange) robota.

Aplikace je spustitelná s grafickým rozhraním pro uživatelskou přívětivost. Byl plánován i mód bez grafického rozhraní, který měl být uzpůsoben k rekalibraci bez zásahu člověka. Tento způsob by ale pro svou realizovatelnost vyžadoval vyřešení kolizí, a proto se jím tato práce nebude zabývat. Tato práce se tedy bude dále výhradně zabývat módem s GUI.

## 5.1 Proces kalibrace obecně

Celkový proces kalibrace se dá shrnout do 5 základních fází, kdy pro přechod do další fáze je nezbytné splnit všechny potřebné kroky ve fázi předchozí.

Obr. 5.1: Blokový diagram aplikace

V první fázi **IDLE**, dochází k spuštění aplikace, dílčím kalibracím a korekcím samotných parametrů. Hlavní krok fáze je načtení JSON souboru s nastaveními samotné kalibrace nebo jeho vytvoření (viz kapitola 5.5).

Ve fázi **Inicializace** dochází ke kontrole všech načtených parametrů nezbytných pro kalibraci, kontrole připojení kamer a robota, kdy je robotem najeto do výchozí pozice a je otestována detekce. V případě nedetekování hledaného bodu je kalibrace neprodleně ukončena. V opačném případě dochází k spuštění vlastní kalibrace.

Fáze **Měření** vyžaduje nastavené tři výchozí pozice a to takové, že každá musí být kolmá vůči ostatním a se stejným natočením vůči kalibračnímu bodu z důvodu již zmíněného problému detekce natočení viz 5.9. Měření probíhá v každé ze tří pozic stejně.

Nejprve dojde k pohybu robotem do výchozí pozice. Následně proběhne synchronní vyfocení a detekce koncového bodu na obou snímcích. Spočítá se vzdálenost mezi kalibračním bodem a detekovaným bodem. Pokud tato vzdálenost je menší

než požadovaná přesnost (parametr v JSON souboru), pozice je uložena a robot přejíždí do další výchozí pozice. V opačném případě dochází k určení nových souřadnic upravených o rozdíl výše popsaných bodů a dochází k nové detekci a měření. Tento iterativní algoritmus je opakován až do doby úspěšné detekce nebo do maximálního počtu iterací nadefinovaného v JSON souboru. V případě úspěšného najetí na kalibrační bod ze všech výchozích pozic dochází k přesunu robota do "home" pozice a přecházíme k další fázi.

Fáze **Výpočtu** vezme uložené pozice robota a výchozí pozici chapadla a to buď z předchozí kalibrace nebo zadanou podle výkresu. Následně je pomocí optimalizačního algoritmu nalezen bod, který leží uprostřed dříve nalezených pozic robota, ale ve stejné vzdálenosti od každého z nich. Výpočet probíhá iterativně do nalezení tohoto bodu nebo do dosáhnutí maximální iterace nadefinované pomocí JSON souboru.

V závěrečné fázi **Vyhodnocení** je výsledek interpretován uživateli aplikace, který může výsledek uložit, nahrát do robota pod zvoleným nástrojem pomocí dialogového okna nebo provést celou kalibraci znova.

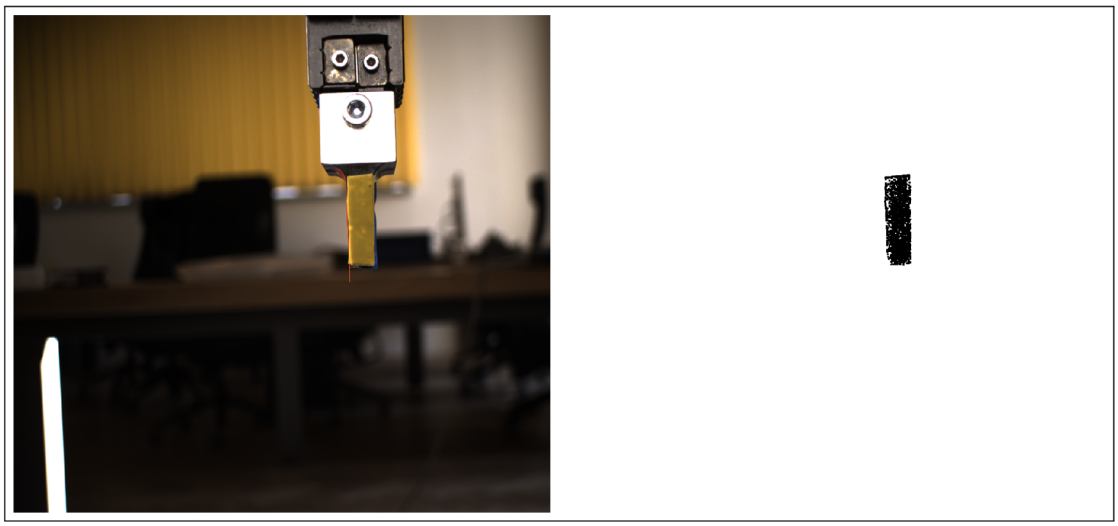
## 5.2 Detekce koncového bodu nástroje

Pro detekci koncového bodu robotického nástroje jsou používány standardní metody počítačového vidění z knihovny OpenCV doplněné o detekční funkce založené na výstupy těchto metod.

Správná detekce chapadla je stěžejní úlohou pro přesnou TCP kalibraci, proto je nesmírně důležité zvolit správnou metodu. Jako první by se nabízel pro detekci template matching nebo surface matching. Tyto metody bohužel nemholy být použity, protože jednak popírají potřeby této kalibrace a to je reagovat na například změnu uchopovací části nástroje nebo jeho opotřebení, stejně jako možnost získání více snímků s různým natočením pro stacionaritu kamer. Další možností byla volba detekce hran objektu a Houghova transformace se stanovení pravidel pro určení TCP na základě získaných informací. Tato metoda se neosvědčila pro náchylnost na odlesky, obtížné nastavování a nestálost parametrů metod počítačového vidění pro různé natočení nástroje a složitě vytvářená pravidla. Nabízelo se i použití například ArUco markerů. Tím by ovšem docházelo k zanesení nepřesnosti pro stanovení polohy TCP vůči markeru. Poslední testovanou metodou, která se nejvíce osvědčila byla následně použita byla aplikace HSV detektoru.

Snímky po jejich pořízení jsou nejprve převedeny z RGB do HSV modelu barev. Následně se vezme tento snímek a je na něj aplikován HSV filtr, který vytvoří masku z původního snímku na základě horních a spodních parametrů filtru viz obrázek 5.2, kde vlevo lze vidět původní obraz a vpravo vyfiltrovaný pomocí HSV filtru i při

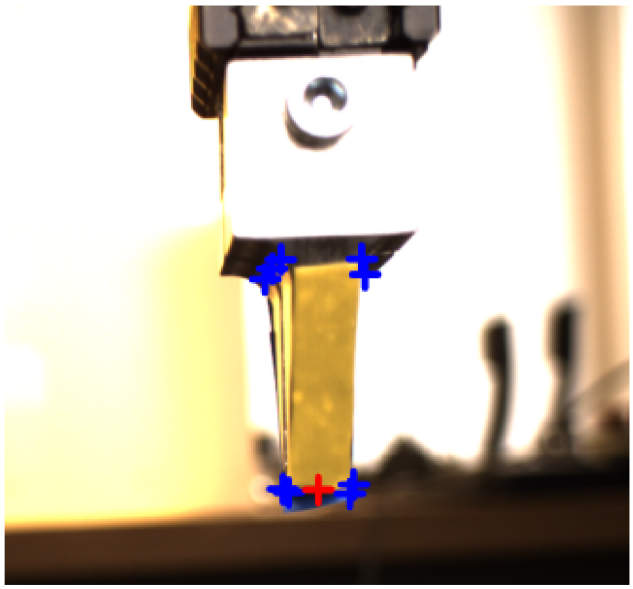
nekontrastním pozadí. Výsledná maska je upravena morfologickými operacemi, aby byl odstraněn zbylý šum a detekovaný objekt byl souvislý.



Obr. 5.2: Aplikace HSV filtru - vlevo původní snímek a vpravo vymaskovaný pomocí HSV filtru

V dalším kroku je aplikován Cannyho hranový detektor, čímž je získán pouze obrys hledaného objektu. V následujícím kroku nalezneme kontury obrysu objektu. Těch je v tuto chvíli mnoho a je potřeba je vyfiltrovat. Protože detekovaný objekt nemusí být přesný geometrický útvar a může obsahovat malé hrany, je nejprve aplikována aproximace díky které jsou kontury v těchto místech potlačeny. V posledním kroku na straně openCV je použit konvexní obal (convex Hull), který vyfiltruje nevýznamné body a následně optimalizuje ty významné viz 5.3 (modré křížky).

Nyní přichází na řadu mnou vytvořená pravidla, která z výsledných kontur stanoví koncový bod nástroje. Jako první si určím těžiště mezi výslednými kontury. Na základě x-ové nebo y-ové souřadnice těžiště vyfiltruji nežádoucí body. Například detekce TCP na obrázku 5.3 je provede tak, že po nalezení těžiště jsou vyfiltrovány všechny kontury, které mají y-ovou souřadnici menší než právě těžiště (levý horní roh obrázku má souřadnice [0,0]). V posledním kroku se spojí všechny body, které nejsou těsně u sebe pomocí přímek a následně z průměru středů těchto přímek je stanoven koncový bod nástroje viz 5.3 (červený křížek).



Obr. 5.3: Hledání kontur a TCP - označené modré body jsou nejvýznamnější kontury nalezené pomocí masky získané z HSV filtru a červený bod je následně detekovaný koncový bod nástroje

## 5.3 Stanovení translace TCP vůči flange

Poté co jsou nalezeny souřadnice příruby (flange) pro všechny tři pozice pootočené vůči sobě o 90° (viz obrázek 5.9), dojde k iterativnímu výpočtu hledané translace pomocí optimalizačního algoritmu PSO, tzv. Optimalizace hejnem částic 1.3.

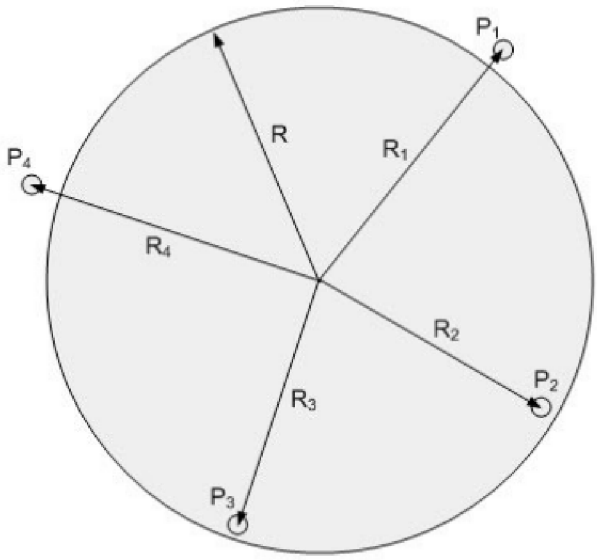
Cílem této optimalizace je nalezení takového poloměru $r$ koule, který bude mít co nejmenší odchylku od vzdáleností (translací) všech bodů okolo povrchu k středu koule v prostoru. Jak tedy vyplývá z obrázku 5.4, body $p_i$ představují získané souřadnice flange a vzdálenosti $R_i$ jsou translace mezi těmito body a jejich detekovanými koncovými body nástroje (TCP). Pro nalezení výsledné translace stačí získat minimálně tři pozice příruby, a to z důvodu omezené možnosti detekce natočení nástroje.[1]

### 5.3.1 Algoritmus výpočtu

Ve výpisu C.1, je zobrazen aplikovaný algoritmus, který vychází z první publikované verze algoritmu z roku 1995 od pánů Jamese Keneddyho a Russela Eberharta.[24]

První fází je inicializace. Ta probíhá pro každou částici $i$ hejna $S$ ve for smyčce. Jako výchozí pozice $x$ částice je nastavena náhodná pozice s rovnoměrným rozlože-

[2]Převzato z: `https://odr.chalmers.se/bitstream/20.500.12380/153281/1/153281.pdf/`




Obr. 5.4: Vizualizace optimalizační úlohy - dokonalá koule s body okolo jejího povrchu[2]

ním. Tato pozice *best* je také označena za nejlepší známou pozici částice a počáteční rychlost $v$ je nulová. Pro tuto částici je následně spočtena její výchozí fitness *fitness* (5.3.1), která je označena za nejlepší fitness *best_fitness*. Následně je hledána nová nejlepší globální fitness $g$, které je na začátku stanovena na maximální možnou hodnotu. Pozice částice, která má nejlepší globální fitness, je uložena jako nejlepší globální pozice *best_global_position*.

V hlavní smyčce, jsou v každé iteraci pro každou částici $i$ vygenerovány nové parametry $r_p$ a $r_g$ pomocí generátoru náhodných čísel rovnoměrného rozložení v rozsahu $(0, 1)$. Je spočtena nová rychlost pomocí rovnice 1.9 a pozice pomocí rovnice 1.10. Následně je pomocí hodnotící funkce 5.3.1 spočten nový $fitness$. Pokud je nový *fitness* lepší než současný *best_fitness*, tak je hodnota *best_fitness* nahrazena novým nejlepším výsledkem a aktuální pozice $x$ je uložena jako nejlepší pozice *best*. Poté, co jsou aktualizovány všechny částice, je hledáno nové nejlepší globální fitness $g$ a nejlepší globální pozice *best_global_position*. Ustáli-li se hodnota $g$, tak je nalezena nejlepší globální pozice *best_global_position* a hlavní smyčka je ukončena. V opačném případě je zvýšena iterace a jsou vypočítány nové hodnoty, a to do doby, než je dosažen limit maximální iterace a optimalizační úloha je prohlášena za neřešitelnou.

### Fitness funkce

Cílem fitness funkce je stanovit jak moc je daná částice vhodná jako nejlepší řešení. Výstupem fitness funkce tedy je průměr kvadratických odchylek jednotlivých souřadnic získaných pomocí rovnice 5.2.[1]

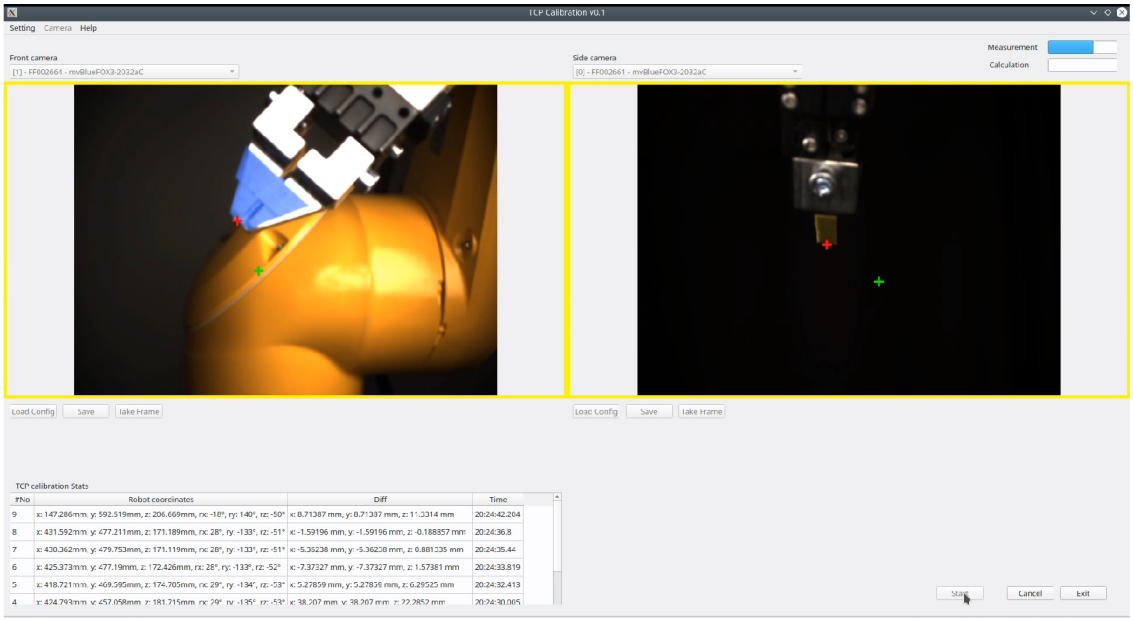
$$\mathbf{R_n} = \mathbf{flange} \cdot \mathbf{x} \cdot \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T \tag{5.1}$$

$$\mathbf{E} = \sqrt{\left(\frac{1}{n}\sum_{i=1}^{n}(\mathbf{R_i} - \mathbf{R_{mean}})^2\right)} = \sqrt{\left(\frac{1}{n}\sum_{i=1}^{n}(\mathbf{R_i} - \frac{1}{n}\sum_{j=1}^{n}\mathbf{R_j})^2\right)} \tag{5.2}$$

## 5.4 Grafické uživatelské rozhraní

Pro grafické uživatelské rozhraní (GUI) je použito open-source knihovny Qt5 pro jednoduchost při vytváření GUI a kompatibilitu se zvoleným programovacím jazykem C++. Tato knihovna byla vytvořena v roce 1999 společností Trolltech. V současnosti je na tomto toolkitu založeno například svobodné desktopové prostředí KDE, které využívá například operační systém Kubuntu.[17]

Navržené grafické rozhraní je vytvořeno v anglickém jazyce pro možnosti dalšího rozšíření.




Obr. 5.5: Hlavní stránka GUI - ukázka průběhu aplikace, kdy je robotické rameno naváděno na kalibrační bod

### 5.4.1 Hlavní stránka

Na hlavní stránce grafického rozhraní je uživateli poskytnuto vše pro dynamické zobrazení průběhu kalibrace s celou řadou možností.

**Front camera a Side camera** Zde je uživateli nabídnuta manuální volba kamer a zobrazují se zde pořízené snímky a grafické výstupy.

**Progress bar** Jsou zde dva ukazatele průběhu kalibrace. První je *Measuremnt*, který zobrazuje postup získávání tří kalibračních pozic. Druhý je *Calculation* a zobrazuje postup výpočtu souřadnic koncového bodu nástroje.

**TCP calibration Stats** Zde jsou zobrazeny při kalibraci aktuální souřadnice flange robota, vzdálenost mezi koncovým bodem nástroje a kalibračním bodem v milimetrech a čas pořízení tohoto záznamu.

**Save** Tlačítko slouží k uložení aktuálního snímku příslušné kamery.

**Load Config** Po stisknutí je otevřeno dialogové okno s volbou otevření *.xml konfiguračního souboru pro příslušnou kameru.

**Take Frame** Slouží k zachycení snímku příslušnou kamerou, který je následně zobrazen.

**Start** Tlačítko slouží pro spuštění kalibrace. Ta je ale spuštěna pouze v okamžiku, kdy jsou načteny všechny náležitosti a algoritmus detekuje ve výchozí pozici koncový bod robotického nástroje. V opačném případě se objeví okno, které oznamuje, co není připraveno nebo nastaveno.

**Cancel** Tlačítko je k dispozici pouze při kalibraci a slouží pro přerušení kalibračního vlákna. Po jeho stisknutí musí být kalibrace spuštěna znovu od začátku tlačítkem “Start”.

**Exit** Stisknutím tohoto tlačítka je přerušena kalibrace. Po opětovném spuštění poběží kalibrace od začátku.

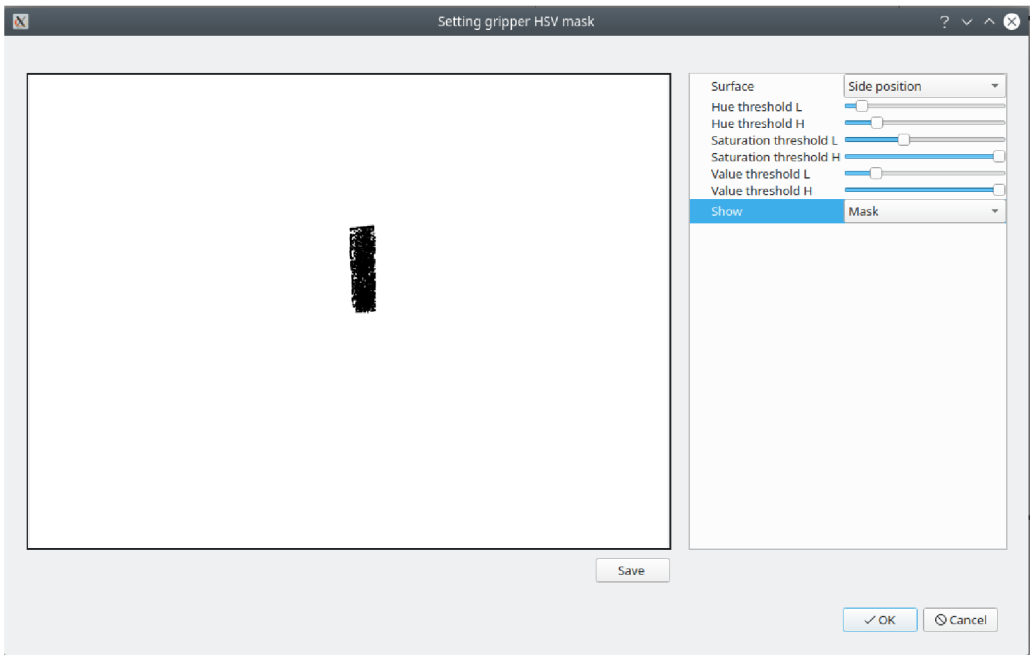
### 5.4.2 Menu Settings

Zde je uživateli umožněno vykonat všechno potřebné nastavení pro načtení, uložení, otestování a upravení všech parametrů.

### Parameters

Tato část složí pro načtení “Save to file” a uložení “Load from file” všech nastavení z a do konfiguračního *.JSON souboru. Dále se zde nachází “Set in scope” s podmenu “CV parameters” a “Gripper colors”.

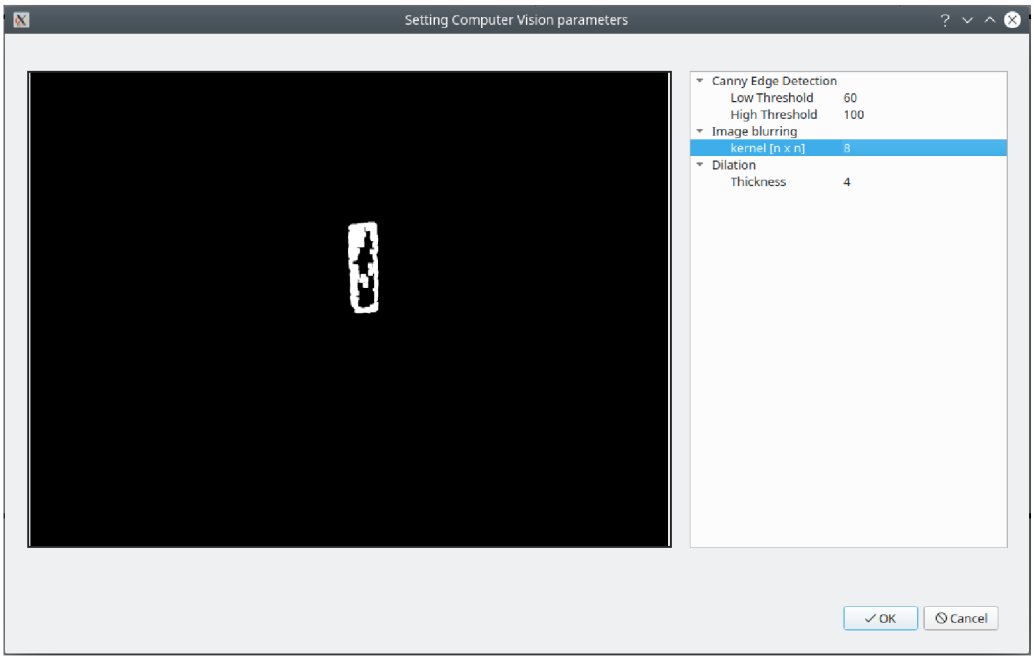
**Grippper colors** Tento nástroj viz obrázek 5.6, umožňuje nastavit prahové hodnoty pro HSV filtr používaný při detekci TCP. Po nahrání snímku lze zvolit, která strana bude detekována a na ní nastavit či nalézt požadované prahové hodnoty filtru. Při změně hodnoty dojde k okamžitému vyhodnocení a snímek s novým nastavením je zobrazen real time. Při zobrazení jde přepnout mezi vyfiltrovaným snímkem a vytvořenou maskou. Zobrazený snímek jde následně uložit. To slouží pro nástroj *CV parameters*, pro který je uložená maska vstupem. Je doporučeno pro správnou funkčnost hodně dynamickému pozadí.




Obr. 5.6: vestavěný nástroj HSV filtr - ukázka hledání optimálního nastavení filtru boční plochy (pohled na výslednou masku)

**CV parameters** Tento nástroj (viz obrázek 5.7), umožňuje nastavit hodnoty pro použité metody počítačového vidění z knihovny openCV. Tyto metody jsou využívány při detekci koncového bodu robotického nástroje. Po načtení masky vytvořené pomocí nástroje *Gripper colors* lze upravovat dostupné parametry a real time sledovat výslednou změnu. V případě, že není možno použít jednu z metod s nastaveným

parametrem, je to uživateli oznámeno ve vyskakovacím oknu a jsou aplikována výchozí nastavení, která byla nastavena při spuštění nástroje.




Obr. 5.7: vestavěný nástroj nastavení parametrů OpenCV metod - ukázka hledání optimální konfigurace parametrů s grafickým výstupem provedené změny

### Camera

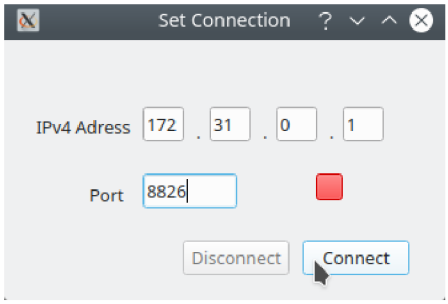
Tato položka nabízí jedinou možnost a to nastavit kalibrační bod "Set calibration point". Pro jeho nastavení je nezbytné, aby uživatel vyfotil pomocí obou kamer snímky, na kterých je kalibrační bod k lokalizaci. V opačném případě je uživateli oznámeno, že není možno nastavit kalibrační bod. Pro nastavení kalibračního bodu se využívá ArUco markeru dle kapitoly 4.3.7.

### Robot

Pod touto položkou je uživateli umožněno provádět všechny úkony s robotem včetně připojení k robotovi.

**Set Connection** Připojení k robotovi je zajištěno pomocí TCP protokolu. Pro připojení je nezbytné, aby bylo na straně robota spuštěno komunikační API. Po nastavení ip adresy a komunikačního portu lze následně navázat spojení stisknutím tlačítka "Connect", viz obrázek 5.8. V případě, že se následně barva tlačítka změní

na šedou, a místo červené značky se rozsvítí zelená, je vytvořeno spojení mezi aplikací a robotem. Následně lze toto spojení i ukončit pomocí tlačítka “Disconnect”.




Obr. 5.8: Okno nastavení připojení robota - ukázka připojení aplikace k robotickému kontroléru pomocí ethernetu

**Set This position as Default** V případě navázaného spojení s robotem je aktuální pozice robota nastavena jako výchozí tzv. “Home”.

**Go Home** V případě navázaného spojení je robot po zvolení této volby přemístěn do výchozí nastavené “Home” pozice.

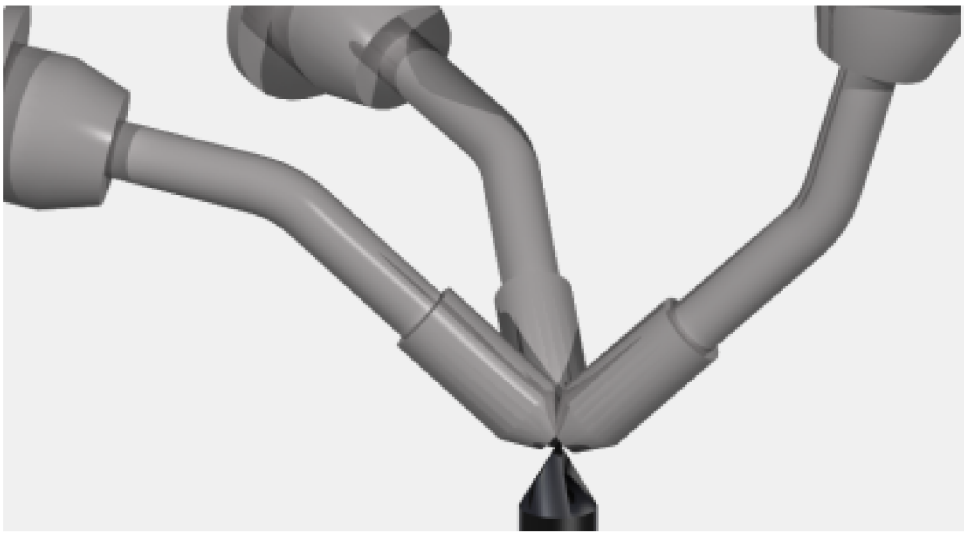
**Default TCPC positions** Pomocí této volby lze nastavit výchozí polohy kalibrace viz obrázek 5.9, najet do zvolené polohy a validovat detekci ve zvolené poloze. V případě, že není navázáno spojení s robotem, je umožněno pouze provést validaci (hledání TCP), ale to samozřejmě za předpokladu, že jsou připojeny a nastaveny kamery.

Při nastavování těchto pozic je potřeba dbát na kolmost všech tří pozic vůči sobě, čímž se zvedá přesnost kalibrace. Dále je potřeba dbát na dodržení pozic. První pozice (*1st position*) je pozice při pohledu na obrázek 5.9 uprostřed. Druhá pozice (*2nd position*) je pozice zleva a poslední (*3rd position*) je pozice zprava.

### Robot-Camera calibration

Protože najíždění na kalibrační bod je iterační algoritmus který vychází z přepočtu pixelového offsetu na kartézský, je nezbytné provést kalibraci robot-kamera. Čím přesněji bude tato kalibrace provedena, tím bude výsledná kalibrace nástroje přesnější. Na robota jsou umístěny ArUco markery (viz 4.3.7), a to na přední a boční stranu v přibližně stejné výšce. Robot je přesunut do pozice, kde je umožněno lokalizovat tyto markery, každý pomocí jedné kamery. Pro uložení této nastavené pozice

[4]Převzato z: `https://robodk.com/doc/en/General.html`



Obr. 5.9: Kalibrační pozice vůči kalibračnímu bodu - přední pohled na ideální kalibrační pozice a natočení nástroje[4]

je potřeba zvolit v podmenu "Set default position". Následně lze spustit kalibraci pomocí volby "Calibrate".

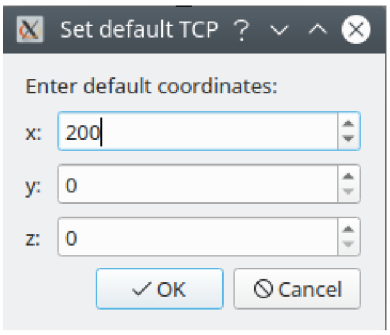
Kalibrace je prováděna pomocí pohybů robota v ose x (x-ové souřadnice přední kamery), y (x-ové souřadnice boční kamery) a z (y-ové souřadnice obou kamer). Robot se vždy vychýlí z výchozí pozice o přesně danou vzdálenost a následně se vrátí. Z výsledného průměrného pixelového ofsetu je při známém pohybu robta stanoven přepočet pixelu na milimetr.

#### TCP

Toto podmenu slouží pro načtení a uložení kalibrací robotického nástroje a nastavení výchozího nástroje pro výpočet. Dále je zde možné pomocí "Test Detection" provést test detekce kontur na aktuálních snímcích. Při volbě "Load", je načtena kalibrace. V případě, že již proběhla úspěšná kalibrace, při volbě "Save" dojde k uložení poslední kalibrace. Před volbou umístění a názvu souboru vyskočí dialogové okno s aktuálním nastaveným nástrojem a nabídne uživateli možnost změny tohoto nástroje pro uložení. Pro nastavení výchozích souřadnic nástroje, viz obrázek 5.10 slouží volba "Set Default". Díky tomu je docíleno rychlejší konvergence a potlačeno uvíznutí v lokálním minimu optimalizačního algoritmu.

### 5.4.3 Menu Camera

Zde se nachází vše ohledně nastavení, kalibrace a kontinuálního režimu kamer.




Obr. 5.10: Nastavení výchozích souřadnic nástroje - sloužících pro rychlejší nalezení TCP a verifikaci výsledku

### mvImpact Settings

Po zvolení "mvImpact Settings" je otevřen konfigurační nástroj *wxPropView* od společnosti Matrix Vision, který je poskytován přímo výrobcem kamer a slouží pro nastavení a odladění nastavení kamer. Pomocí tohoto nástroje lze také vygenerovat konfigurační *.xml soubor s nastavením kamer, který se používá při kalibraci nástroje pro nastavení a kontrolu správnosti nastavených parametrů.

### Refresh

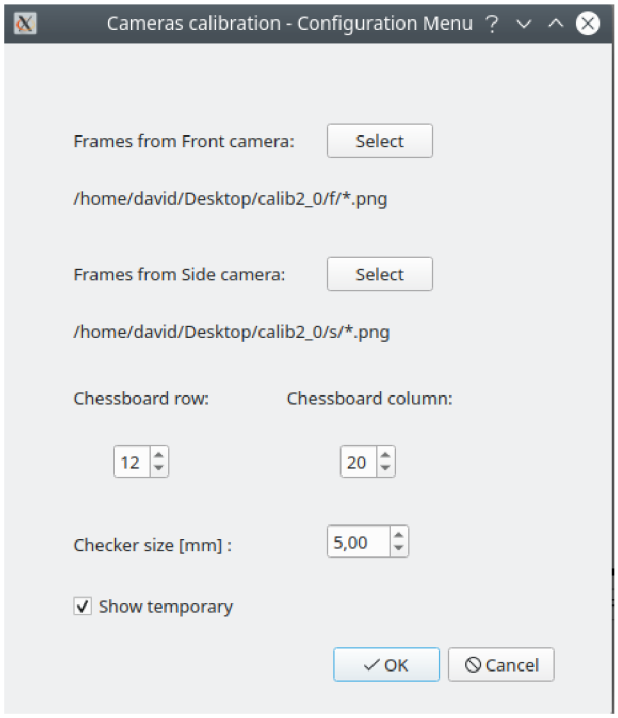
Pomocí této volby dojde k aktualizaci připojených kamer. Připojené kamery jsou následně nabídnuty uživateli při manuální volbě kamer nad jednotlivými pozicemi (Front camera a Side camera).

### Calibration

Zde je uživateli poskytnuto vše pro kalibraci kamer. Kalibrace probíhá společně - je tedy prováděna stereo kalibrace. Prerekvizitou pro tuto kalibraci je libovolná šachovnice. Doporučená je o rozměrech 12x20 dílků při velikosti dílu 5mm. Uživatel musí docílit při pořizování datasetu toho, aby šachovnice byla zřetelně viditelná pomocí obou kamer zároveň. Pro synchronní pořízení snímků slouží volba "Shoot All", která zachytí a následně zobrazí snímky. Jejich uložení následně probíhá pomocí volby "Save All Frames". Pro kalibraci je nezbytné ukládat snímky z přední a boční kamery separovaně do vlastní složky avšak pod stejným názvem.

**Calibrate** Při zvolení této volby je zobrazeno konfigurační menu stereo kalibrace (viz obrázek 5.11). Uživatel musí zvolit složku se snímky z přední kamery (*Frames from Front camera*) a boční kamery (*Frames from Side camera*). Důležité je, aby v každé složce byl stejný počet snímků a vždy pro konkrétní pozici v každé složce

jeden snímek a to ideálně stejně pojmenovaný. V opačném případě nelze zaručit správnost výsledku kalibrace. Jelikož uživatel může použít libovolnou šachovnici, musí zadat parametry této šachovnice. Počet řádků šachovnice (*Chessboard row*), počet sloupců šachovnice (*Chessboard column*) a velikost dílku v milimetrech (*Checker size*). Uživateli je poskytnuta možnost zobrazení detekovaných šachovnic (*Show temporary*). Kalibrace je spuštěna tlačítkem "Ok". V případě úspěšné kalibrace je zobrazen poměr validních detekcí z celkového počtu scén a výsledky jsou uloženy do paměti programu a mohou být následně uloženy pomocí konfiguračního *.JSON souboru viz kapitola 5.4.2 (volba "Save to file"). Je doporučeno kalibrovat optimálně 20 validními pozicemi šachovnice.




Obr. 5.11: Menu stereo kalibrace - ukázka nastavování parametrů pro provedení stereo kalibrace kamer přípravku

### Continuous mode

Po zvolení volby "Start" při nastavení obou kamer, je v tomto podmenu spuštěno kontinuální zobrazování snímků (živě). Protože zde není implementována žádná komprese obrazu a jsou zobrazovány plné snímky, je uživateli před samotným spuštěním přenosu nabídnuto snížení rozlišení pomocí dialogového okna. Pro ukončení kontinuálního módu je nezbytné stisknout v tomto podmenu tlačítko "Stop". Tím je přenos

přerušen a v případě že bylo uživatelem zvoleno snížení rozlišení, je nastaveno opět rozlišení původní.

### 5.4.4 Menu Help

Zde je uživateli poskytnut návod (instrukce) pro obsluhu této kalibrační aplikace. A při volbě "About" uživatel získá informace o verzi této aplikace.

## 5.5 JSON konfigurační soubor

Pro správu parametrů kalibrace je použit JavaScriptový objektový zápis tzv. JSON, který byl standardizovaný v roce 1999. Výhodou tohoto způsobu zápisu je, že není vztažen k žádné počítačové platformě a je jednoduše srozumitelný jak pro člověka tak pro stroj. I proto se dnes těší stále větší oblibě a je rozšířený napříč všemi odvětvími.[26]

Konfigurační soubor sestavený pro potřeby kalibrace využívá devět Objektů. V následujících odstavcích popíšu každý objekt a přiblížím jeho význam.

#### objekt CV parameters

Tento objekt obsahuje parametry pro konvenční počítačové metody knihovny openCV.

**pod-objekt Canny Edge Detection** Tento podobjekt obsahuje hodnoty **High Threshold** a **Low Threshold**, které definují hranice, kdy je bod označen za významný.

**pod-objekt Other parameters** Zde jsou ukryty hodnoty **Blur kernel**, která udává velikost "n x n" konvolučního jádra pro tuto metodu a **Dilate thickness**, která udává šířku dilatace v pixelech.

#### objekt Calibration piont

Tento objekt schovává pole **Front view** a **Side view**, které obsahují souřadnice x a y kalibračního bodu v pixelech na jednotlivých kamerách.

#### objekt Camera-Robot calibration

Objekt obsahuje pod-objekt **Position**, s uloženými souřadnicemi bodu pro kalibraci robot-kamera a pod-objekt **Coordinates offset**, který obsahuje hodnoty přepočtu pro jednotlivé souřadnice z pixelu na milimetr.

#### objekty Front camera a Side camera

Tyto dva objekty jsou totožné. Oba obsahují konfiguraci ke konkrétní kameře.

**hodonta Calibrated** Obsahuje boolovskou hodnotu, která ukazuje, zda je kamera nakalibrována či nikoliv.

**hodonta Exposure time [us]** Hodnota obsahující velikost expozice, která se nastaví na kameře v mikrosekundách. Je to jeden z možných způsobů nastavení této hodnoty.

**hodonta Path *.xml** Hodnota obsahuje string v podobě cesty ke konfiguračnímu souboru kamery. Soubor je dobré použít pro vypnutí všech automatických nastavení a korekcí na kameře pro zabráněni libovolné změny nastavení parametrů kamery.

**hodonta Pixel format** Hodnota obsahující string s nastaveným režimem formátu. Doporučeno je používat formát "RGB888Packed".

**hodonta Serial no.** Obsahuje sériové číslo, pomocí kterého je následně možné přiřadit konkrétní kameru do požadované pozice.

**pole Dist coeffs, K matrix, Projection matrix a Rectification transform** Pole obsahující matice a vektor parametrů kamery získaných z kalibrace.

#### objekt HSV filter

Objekt obsahuje pod-objekty **Front position** a **Side position**, s hodontami jednotlivých složek filtrů pro danou pozici.

**pod-pod-objekty Hue, Saturation a Value** Zde jsou uloženy spodní a horní hranice jednotlivých složek HSV filtru, díky kterým je možno detekovat hledanou část nástroje na snímku.

#### objekt Robot interface

Objekt složí k nastavení připojení robota.

**pod-objekt Connection** Tento podobjekt obsahuje hodnoty **IPv4** a **Port**, díky nimž je možno navázat s robotem TCP komunikaci.

**pole Default position** Toto pole nastavuje výchozí pozici robota jeho tzv. "Home" pozici pro kalibraci.

**hodnota Type** Tento string obsahuje označení výrobce robota. V případě, že v budoucnu by byla tato aplikace rozšířena o dalšího výrobce, sloužila by tato hodnota k jejich odlišení.

**objekt Stereo camera**

Tento objekt sdružuje společné výsledky stereo kalibrace kamer.

**pole Essential matrix, Fundamental matrix, Q matrix, Rotation matrix a Translation vector** V těchto polích jsou uloženy matice a vektor získané pomocí stereo kalibrace.

**objekt TCPC parameters**

Tento poslední objekt slouží pro samotná nastavení TCP kalibrace.

**pod-objekt Default gripper** Jsou zde hodnoty s **x**, **y** a **z** souřadnicemi nástroje. Tato hodnota by měla být získána z výkresové dokumentace nástroje.

**pod-objekt Default positions** Obsahuje pod-pod-objekty **1st position**, **2nd position** a **3rd position** s hodnotami jednotlivých výchozích souřadnic příruby pro kalibraci nástroje.

**hodonta Maximal find iteration** Hodnota stanovuje limit iterativního procesu pro najetí na kalibrační bod.

**hodonta Precision [px]** Hodnota vymezuje hranici kalibračního bodu. Je-li robot uvnitř krychle vymezené právě hranou rovné této hodnotě, tak je označena tato pozice za dosaženou.

### 5.5.1 Vytvoření JSON souboru

Konfigurační *.JSON soubor lze vytvořit manuálně. Tedy odhadem nebo přepisem předchozího souboru nebo automaticky pomocí nástrojů této aplikace, což je i doporučeno. Níže bude představen doporučený postup vytvoření konfiguračního souboru.

Nejprve je potřeba nastavit kamery. K tomu je dobré využit konfigurační nástroj Matrix Vision dle kapitoly (5.4.3). Následně přiřadíme v hlavním okně jednotlivým pohledům kamery. V dalším kroku je provedena kalibrace kamer. Nejprve pomocí

tlačítka “Load Config” se nahraje konfigurace z *.xml souboru do jednotlivých kamer. Následně je pomocí šachovnice vytvořen dataset (viz5.4.3). Doporučeno je získání třiceti scén (ne všechny následně mohou být validní). Po pořízení datasetu je provedena kalibrace viz 5.4.3.

Následně je potřeba označit chapadlo robota (viz 4.3.6) a ideálně nad tuto zónu připevnit i ArUco kalibrační marker (viz 4.3.7). Poté je připojen robot viz 5.4.2 a nastavena “Home” pozice (5.4.2), kdy ideálně hledaný koncový bod nástroje je uprostřed záběrů.

V další fázi je potřeba navést robotické rameno do pozice, kdy je ArUco marker ideálně uprostřed obou záběrů kamer. Je provedeno nastavení kalibračního bodu (5.4.2) a následně je pozice uložena a provedena kalibrace robot-kamera (5.4.2). Po přivedení robota do “Home” pozice a pořízení snímků pomocí tlačítek “Take Frame” a jejich uložení (tlačítko “save”) je provedeno nastavení HSV filtru viz 5.4.2. Výslednou masku je potřeba uložit pro nastavení parametrů konvenčních počítačových metod (5.4.2).

V poslední fázi je potřeba nastavit výchozí pozice pro kalibraci viz 5.4.2. Pro ověření je možno i následnou pozici validovat a ověřit funkčnost nastavených parametrů.

Parametry “Maximal find iteration” a “Precision [px]” jsou nastaveny již ve výchozí hodnotě. Pro jejich změnu je nutné editovat vytvořený konfigurační soubor.

Po nastavení všech parametrů a provedení všech kalibrací je nezbytné tyto parametry uložit. V menu “Settings” zvolíme možnost “Save to file”. Zde je důležité nastavit i cestu ke konfiguračnímu *.xml souboru kamer. K tomu slouží dialogové okno. Při dalším spuštění stačí jen načíst tento vytvořený konfigurační soubor.

### 5.5.2 Výsledný JSON

Výsledek kalibrace je uchováván také v konfiguračním souboru *.JSON. Tento kalibrační soubor obsahuje:

**pole TCP** Zde jsou souřadnice výsledné kalibrace (translace flange k TCP).

**hodnota TOOL** Zde je číslo nástroje, kterému má být tato kalibrace přiřazena.

**hodnota Timestamp** Časová značka uložení kalibrace robotického nástroje.

## 5.6 Logování

V adresáři *home* je od prvního spuštění aplikace vytvořen soubor “log.log”, do kterého jsou zaznamenávány všechny prováděné úkony aplikace. Tyto zápisy jsou opatřeny časovou značkou a slouží monitorování chodu a k případnému ladění nežádoucích stavů během provozu. Ukázka výpisu kalibrace viz obrázek 5.12.

```
21.05.2020 20:11:40 called `getCameras()` with founded: 2 cameras
21.05.2020 20:11:54 *.JSON file was succesfully loaded
21.05.2020 20:11:55 Calibration started
21.05.2020 20:13:19 *.JSON TCP Calibration file was saved to: /home/david/TCP_calibration/tool_6.json
21.05.2020 20:17:31 Calibration canceled
21.05.2020 20:17:38 Calibration started
21.05.2020 20:20:02 Calibration canceled
21.05.2020 20:20:12 Calibration started
21.05.2020 20:22:03 called `getCameras()` with founded: 2 cameras
21.05.2020 20:22:18 *.JSON file was succesfully loaded
21.05.2020 20:22:26 Calibration started
21.05.2020 20:23:08 Calibration canceled
21.05.2020 20:23:15 Robot-Camera calibration Done (x:1mm=11,935000px,y:1mm=14,530000px,z:1mm=15,885000px)
21.05.2020 20:24:13 Calibration started
21.05.2020 20:25:39 *.JSON TCP Calibration file was saved to: /home/david/TCP_calibration/TOOL_4.json
21.05.2020 20:28:52 GUI closed
23.05.2020 20:04:31 called `getCameras()` with founded: 0 cameras
23.05.2020 20:04:36 Openned Gripper color detection parameters menu
```

Obr. 5.12: Ukázka logovacího výpisu

## 5.7 Programovací jazyk

Programovací jazyk pro aplikaci byl zvolen C++ pro svou univerzálnost a dobré optimalizační schopnosti. Vývojové prostředí je použito Visual Studio Code a Qt Creator.

## 5.8 Shrnutí

V kapitole byl popsán nejprve proces kalibrační aplikace obecně a následně byly rozebrány jeho dvě stěžejní části a to detekce koncového bodu nástroje a nalezení souřadnic tohoto nástroje v souřadném systému robota. Pro detekci TCP bylo využito konvenčních metod počítačového vidění a především HSV filtru pro nalezení požadované plochy nástroje na kterých je detekován TCP, což umožňuje navést robota na kalibrační bod. Pro nalezení souřadnic TCP v souřadnicích robota, tedy translace mezi flange a TCP, je využito optimalizační metody - optimalizace hejnem částic (PSO), která hledá ideální střed mezi získanými souřadnicemi (nalezené souřadnice kalibračního bodu v souřadném systému robota) okolo myšleného povrchu koule.

V další části této kapitoly bylo popsáno Grafické uživatelské rozhraní (GUI), jeho nástroje a obsluha. Byl také rozebrán konfigurační a výsledný JSON soubor, který je nezbytný pro spuštění kalibrace. Také bylo zmíněno logování chodu aplikace a popis spuštění konzolové aplikace TCP kalibrace bez grafického uživatelského rozhraní.

# 6 Vyhodnocení výsledků kalibrace

V této kapitole budou vyhodnoceny výsledky navrhnuté a realizované kalibrace a následně srovnány s metodicky podobnou kalibrací.

## 6.1 Stanovený cíl

Před zahájením vývoje tohoto experimentálního vzorku zařízení pro automatickou nástrojovou kalibraci robotického manipulátoru pomocí metod strojového vidění, jsem si položil za hlavní cíl dostat se s přesností kalibrace pod 1mm. Další cíl byl nahradit manuální kalibraci, která probíhá kalibrací pomocí kalibračního kolíku najetím ze tří stran na tento kolík. Za kritéria jsem si zvolil čas, přesnost, opakovatelnost a cenu. Cena této kalibrace byla jedním z motivačních faktorů, protože ceny "State of The Art" kalibračních přípravků se pohybují v tisících eur. Cíl byl tedy vyrobit dostupnou kalibrační metodu pro konkurenceschopnost schopnost průmyslovým řešením za zlomkovou cenu.

## 6.2 Dílčí kalibrace

Výsledek samotné kalibrace je součtem všech dílčích kalibrací, které jsou nezbytné pro provedení kalibrace koncového bodu robotického nástroje. Robotický 6-osý manipulátor TX2-90, jak již bylo dříve zmíněno je schopen vykonávat svůj pohyb s přesností na ±0.03 mm. Jeho nepřesnost tedy může být vzhledem k celé kalibraci zanedbatelná. Přesnost kamer lze stanovit výpočtem z matice vnitřních parametrů přední kamery $K_1$ a matice vnitřních parametrů boční kamery $K_2$.

$$K_1 = \begin{bmatrix} 2916,53 & 0 & 943,07 \\ 0 & 3075,17 & 853,48 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.1}$$

$$K_2 = \begin{bmatrix} 2910,54 & 0 & 865,60 \\ 0 & 2962,63 & 692,80 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.2}$$

Při znalosti rozlišení (2048x1536 px) a velikosti (7,0656x5,2992 mm) čipu SONY IMX265 lze jednoduše ověřit ohniskovou vzdálenost a souřadnice středu obrazu v px. Ohniskovou vzdálenost lze vypočítat pomocí rovnice 6.3. Souřadnice středu obrazu lze získat z rozlišení.[27]

$$F = vzdálenost\ ohniska\ v\ px \frac{rozměry\ senzoru}{rozlišení} \tag{6.3}$$

| | $F_h$ [mm] | $F_v$ [mm] | $c_x$ [px] | $c_y$ [px] |
|---|---|---|---|---|
| přední kamera ($K_1$) | 10,06 | 10,61 | 943,07 | 853,48 |
| boční kamera ($K_2$) | 10,04 | 10,22 | 865,60 | 692,80 |
| skutečné parametry | 10 | 10 | 1024 | 768 |

Tab. 6.1: Srovnání výsledků kalibrace kamer

Z tabulky 6.1 je patrné, že kalibrační dataset není optimální a vnáší do kalibrační aplikace nepřesnost.

Kalibrace robot-kamera je klíčovou kalibrací pro realizaci kalibrace. Původně bylo místo přepočtených koeficientů zamýšleno použití transformační matice. Ta bohužel nebyla kvůli implementačním problémům a špatným výsledkům použita a byla vytvořena kalibrace robot-kamera.

| Souřadnice | $\Delta$ [$\mu m$] | $\Delta$ [px] | $\delta$ [%] |
|---|---|---|---|
| X | 2 | 6,1 | 2,51 |
| Y | 0,13 | 0,38 | 0,16 |
| Z | 1,01 | 4,6 | 1,5 |

Tab. 6.2: Výsledek kalibrace robot-kamera

Z tabulky 6.2 je patrné, že kalibrace robot-kamera je dostatečně přesná, a proto ji můžeme zanedbat.

Do dílčí kalibrace by mohla být započítána i detekce koncového bodu nástroje pomocí metod počítačového vidění. Ta je největším úskalím celé kalibrace, protože opakovatelnost detekce konkrétního pixelu na detekovaném objektu lze velmi složitě odhadnout. Přesnost detekce je ovlivněna nejvíce tvarem a přesností detekční plochy, nastavením HSV filtru a nerovnostmi (nepravidelnostmi, opotřebením) samotného nástroje. Můj odhad nepřesnosti vzniklou detekcí je $\pm 2 - 3$ mm. Vyšší přesnosti by šlo docílit použitím nástroje se zakončeným hrotem nebo výrobně upraveným nástrojem pro potřeby této kalibrace.

## 6.3 Automatická kalibrace vs. Manuální kalibrace

Jak již bylo řečeno, pro srovnání výsledků kalibrace jsem zvolil manuální kalibraci, protože na její metodice (point-counterpoint) je založena realizovaná kalibrace. Byly

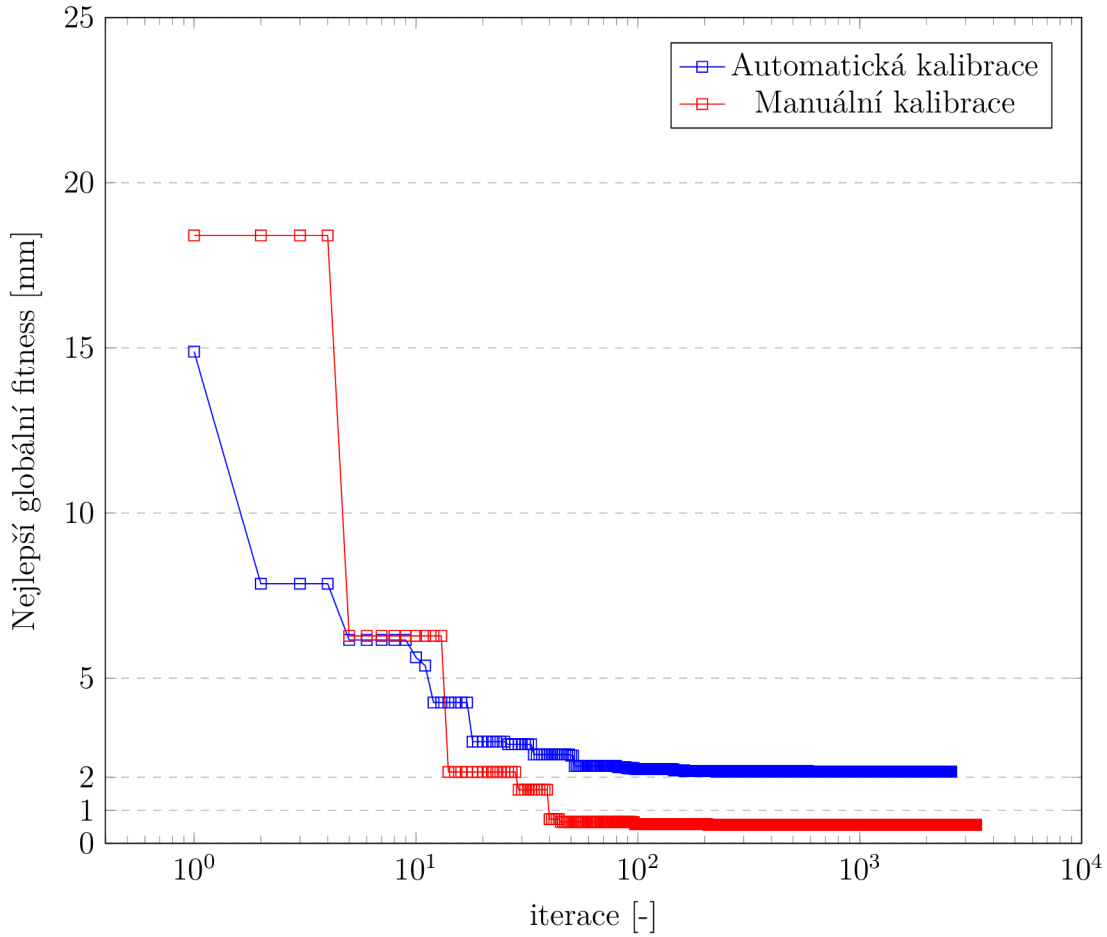
provedeny tři manuální kalibrace a tři automatické kalibrace, na základě jejichž výsledků byla sestavena tato srovnávací tabulka obou kalibrací. Pro manuální kalibraci jsem využil upravené verze firemní robotické kalibrace a tuto kalibraci jsem prováděl sám.

| Manuální kalibrace vs. Automatická kalibrace | | | | | |
|---|---|---|---|---|---|
| Typ kalibrace | čas měření | čas výpočtu | přesnost | opakovatelnost | cena |
| Manuální | 5 min 21 s | 35 s | 0,925 mm | ± 0,754 mm | - |
| Automatická | 31 s | 30 s | 3,111 mm | ± 1,285 mm | - |

Tab. 6.3: Srovnání vybraných parametrů automatické a manuální kalibrace

Z tabulky 6.3 je patrné, že hlavní předností vytvořené kalibrační aplikace je rychlost, i když za cenu přesnosti. Nicméně i přes všechny zavedené nepřesnosti vzniklé dílčími kalibracemi (6.2) považuji výsledek za obstojný. Výsledek manuální kalibrace je však velmi ovlivněn proveditelem tohoto úkonu. Má nejlepší dosažená přesnost byla okolo 0,5 mm, avšak kolega provedl stejnou kalibraci s přesností pod 0,1 mm, ovšem za delší čas. Z toho plyne, že srovnávání těchto kalibrací není úplné ideální, ale může sloužit jako dobré vodítko. Důležitým, a v tuto chvíli těžko srovnatelným parametrem je cena prováděné kalibrace z pohledu ceny aplikace, a ceny technika, jež tuto kalibraci provede. Z pohledu odstávky výroby, nutné pro nutné vykonání kalibrace, má navrch vytvořená kalibrační aplikace, která dovede provést stejnou kalibraci za jednu šestinu manuální kalibrace. Dalším faktorem, který by mohl hrát ve prospěch této aplikace je fakt, že provádí-li kalibraci pokaždé jiný technik, může být docíleno naprosto odlišné přesnosti Tento problém však výsledná aplikace odstraňuje.

Z grafu 6.1, který srovnává výsledky nejlepších validačních kalibrací optimalizační úlohy manuální a automatické kalibrace, je patrné, že obě metody přibližně stejně rychle konvergují k výsledku, a to do sté iterace optimalizačního algoritmu.




Obr. 6.1: Srovnání výsledků optimalizační úlohy manuální a automatické kalibrace

# Závěr

Cílem této diplomové práce bylo především navrhnout experimentální vzorek zařízení pro automatickou nástrojovou kalibraci robotického manipulátoru pomocí metod strojového učení.

Za tímto účelem byla provedena rešerše nejpoužívanějších metod manuální a automatické kalibrace robotického nástroje, kdy všechny metody vycházejí ze základních tří metodik:

- Point-counterpoint method (metoda různých směrů přiblíženi)
- Touch probe method (metoda dotekové síly)
- Laser pointer method (metoda měření s využitím laserového paprsku)

Byly popsány a vysvětleny použité motody strojového vidění spolu s objasněním postupu, stejně jako teorie kalibrace kamer a matematických prerekvizit potřebných k pochopení (objasnění) principů výsledné kalibrace. Tyto získané teoretické znalosti následně byly využity pro realizaci této práce.

Byl navržen prototyp testovacího vzorku zařízení pro automatickou kalibraci nástroje robotického manipulátoru s definovanou robotickou platformou. A také zvolen a popsán hardware využitý pro tento kalibrační přípravek.

Byl navržen algoritmus kalibrace, který je popsaný a vysvětlený v kapitole 5, spolu s navrženým uživatelským grafickým rozhraním (GUI). Dále byly vytvořeny a popsány nástroje pro jednoduchou obsluhu tohoto experimentálního vzorku a další náležitosti nezbytné k chodu aplikace.

Výsledná aplikace byla následně validována s manuální kalibrací, jež je také založena na metodice "Point-counterpoint" metody. Tyto výsledky jsou reprezentovány spolu s popisem nejslabších částí kalibrace.

Cílem mé aplikace bylo navrhnout a vytvořit kalibrační přípravek pro kalibraci ideálně všech nástrojů bez použití kalibračního hrotu s přesností pod 1 mm a tím vytvořit konkurenceschopnou kalibrační aplikaci. Tento cíl byl téměř dosažen s přihlédnutím na omezené možnosti při vývoji této aplikace. Největším úskalím této práce bylo nalezení vhodné metody pro detekci koncového bodu nástroje. Jako nejvhodnější se nakonec ukázala metoda s využitím HSV filtru a označením nástroje pomocí barev. Omezením tohoto řešení je použití v místech s výrazně dynamickou změnou pozadí. Vhodný nástroj pro detekci je jakýkoliv nástroj libovolného tvaru,

vhodně barevně označený pro detekci (viz 5.2). Čím přesněji a robustněji je toto označení provedeno, tím je výsledná aplikace účinější. Nejpřesnější kalibrace bude docíleno v případě, že nástroj má hrot. Omezením je také je nerealizovatelnost při současném nastavení pro detekci natočení nástroje. To by šlo v budoucnu odstranit pomocí markerů. Zajímavým parametrem srovnání s běžně dostupnými komerčními řešeními je cena, která se nepohybuje v desítkách, ale v jednotkách tisíc eur.

Při rozšíření o zabezpečení kolizí by již tato aplikace mohla být provozována plně automaticky bez zásahu člověka. Možným rozšířením je vyřešení kolizí robota pro plně automatický režim, nahrazení kalibrace robot-kamera pomocí přepočtových koeficientů transformační maticí mezi souřadnými systémy robota a kamer nebo integrování svítidel na konstrukci přípravku. Zajímavým směrem vývoje by bylo využití UV kamer a UV barev. V případě kvalitního nanesení barvy na detekční plochu by bylo možno docílit ještě lepších výsledků. Za otestování by také stálo použití neuronových sítí pro detekci TCP.

# Literatura


[1] BERGSTRÖM, G.: *Method for calibration of off-line generated robot program* [online]. [cit. 20. 12. 2019]. Göteborg, Sweden, 2011. Master thesis. Dostupné z URL: <http://publications.lib.chalmers.se/records/fulltext/153281.pdf>.

[2] *KUKA System Software 8.3: Operating and Programming Instructions for End Users.* 237 stran. Augsburg, Germany, 2013.

[3] ABB: *Application manual - BullsEye.* revision C. 90 stran. Västerås, Sweden, 2018.

[4] FANUC: *Tool Center Point Control (TCP)* [online]. [cit. 1. 11. 2019]. Dostupné z URL: <https://www.fanucamerica.com/products/cnc-products-ii/cnc-technology/tool-center-point-control>.

[5] LEONI AG: *Sensor & vision solutions* [online]. [cit. 26. 12. 2019]. 2019. Dostupné z URL: <https://d1619fmrcx9c43.cloudfront.net/fileadmin/factory_automation/publications/catalogues/machine_perceptio.pdf?1552292586>.

[6] UHLÍŘ, J.: *Kalibrace robotického pracoviště* [online]. [cit. 1. 11. 2019]. Brno, 2019. Diplomová práce. FIT VUT v Brně. Dostupné z URL: <https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=198027>.

[7] OpenCV: *Camera calibration With OpenCV* [online]. [cit. 3. 11. 2019]. Dostupné z URL: <https://docs.opencv.org/2.4/doc/tutorials/calib3d/camera_calibration/camera_calibration.html>.

[8] LIŠTVAN, J.: *Kalibrace kamery pro robotické pracoviště* [online]. [cit. 3. 11. 2019]. Praha, 2017. Bakalářská práce. ČVUT v Praze. Dostupné z URL: <https://dspace.cvut.cz/bitstream/handle/10467/68387/F3-BP-2017-Listvan-Jaroslav-Kalibrace%20kamery%20pro%20roboticke%20pracoviste.pdf>.

[9] MAINBERGER, M.: *Pinhole camera model* [online]. [cit. 5. 11. 2019]. 2010. Dostupné z URL: <http://faculty.pucit.edu.pk/nazarkhan/teaching/Fall2018/CS465/Lectures/lecture11_camera_geometry.pdf>.

[10] JANÁKOVÁ, I.: *Filtrace šumu a poruch* [online]. [cit. 18. 11. 2019]. Dostupné z URL: <http://midas.uamt.feec.vutbr.cz/ZVS/Lectures/07_Filtrace_sumu_a_poruch.pdf>.

[11] JANDA, M.: *Cannyho operátor a další používané hranové detektory* [online]. [cit. 6. 12. 2019]. Brno, 2010. Bakalářská práce. FIT VUT v Brně. Dostupné z URL: <https://core.ac.uk/download/pdf/44392459.pdf>.

[12] *Cannyho hranový detektor* [online]. [cit. 26. 12. 2019]. 2018. Wikipedia: the free encyclopedia. Dostupné z URL: <https://cs.wikipedia.org/wiki/Cannyho_hranov%C3%BD_detektor>.

[13] STÄUBLI: *TX2-90 Range* [online]. [cit. 27. 12. 2019]. 2019. Dostupné z URL: <https://www.staubli.com/en-hk/file/5260.show>.

[14] MATRIX VISION GmbH.: *USB3 Vision camera with Sony CMOS sensors - mvBlueFOX3-2* [online]. [cit. 27. 12. 2019]. Dostupné z URL: <https://www.matrix-vision.com/USB3-vision-camera-mvbluefox3-2.html>.

[15] *NI: Calculating Camera Sensor Resolution and Lens Focal Length* [online]. [cit. 27. 12. 2019]. Dostupné z URL: <https://www.ni.com/cs-cz/support/documentation/supplemental/18/calculating-camera-sensor-resolution-and-lens-focal-length.html>.

[16] *VS-LDA series: Product spec* [online]. [cit. 27. 12. 2019]. Dostupné z URL: <https://vst.co.jp/en/machine-vision-lenses-en/vs-lda-series/>.

[17] *Trolltech* [online]. [cit. 2. 1. 2020]. 2018. Wikipedia: the free encyclopedia. Dostupné z URL: <https://cs.wikipedia.org/wiki/Qt_Development_Frameworks>.

[18] HALLENBERG, J.: *Robot Tool Center Point Calibration using Computer Vision* [online]. [cit. 2. 1. 2020]. Linköping, Sweden, 2007. Master thesis. Dostupné z URL: <https://www.diva-portal.org/smash/get/diva2:23964/FULLTEXT01.pdf>.

[19] TSAI, L.-W.: *Robot analysis: the mechanics of serial and parallel manipulators*, New York, N.Y. 0158-0012: John Wiley & Sons, INC., 1998. 520 s. ISBN 978-0-471-32593-2.

[20] MARSH, D.: *Applied geometry for computer graphics and CAD*, London: Springer-Verlag London Limited, 2005. 361 s. ISBN 9781846281099.

[21] HLAVÁČ, V.: *Matematická morfologie* [online]. [cit. 10. 2. 2020]. Dostupné z URL: <http://people.ciirc.cvut.cz/~hlavac/TeachPresCz/11DigZprObr/71-3MatMorpholBinCz.pdf>.

[22] KIRANYAZ, S.; PULKINNEN, J.; GABBOUJ, M.: *Multi-dimensional particle swarm optimization for dynamic environments,* 26 stran. Al Ain: 2008 International Conference on Innovations in Information Technology, 2008.

[23] NĚMEC, F: *Částicové optimalizační techniky – Particle Swarm Optimization* 7 stran. Brno: VUT v Brně, 2015.

[24] KENNEDY, J.; EBERHART, R.: *Particle swarm optimization* 7 stran. Washington, DC: IEEE, 1995. ISBN 0-7803-2768-3.

[25] SLABAUGH, G., G.: *Computing Euler angles from a rotation matrix* [online]. [cit. 8. 4. 2020]. Dostupné z URL: <`https://www.gregslabaugh.net/publications/euler.pdf`>.

[26] *Introducing JSON* [online]. [cit. 10. 5. 2020]. Dostupné z URL: <`https://www.json.org/json-en.html`>.

[27] LIŠTVAN, J.: *Kalibrace kamery pro robotické pracoviště* [online]. [cit. 20. 5. 2020]. Dostupné z URL: <`https://dspace.cvut.cz/bitstream/handle/10467/68387/F3-BP-2017-Listvan-Jaroslav-Kalibrace%20kamery%20pro%20roboticke%20pracoviste.pdf`>.

# Seznam příloh

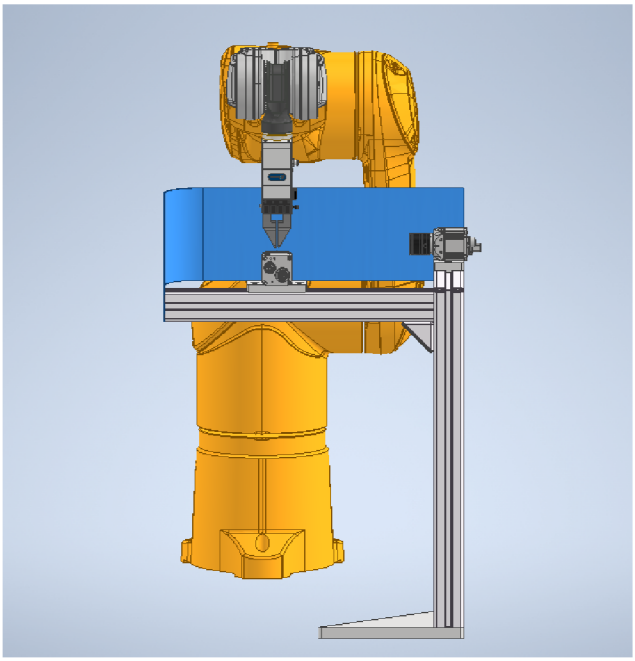
**A Návrh kalibračního přípravku** **81**

**B Realizace kalibračního přípravku** **83**

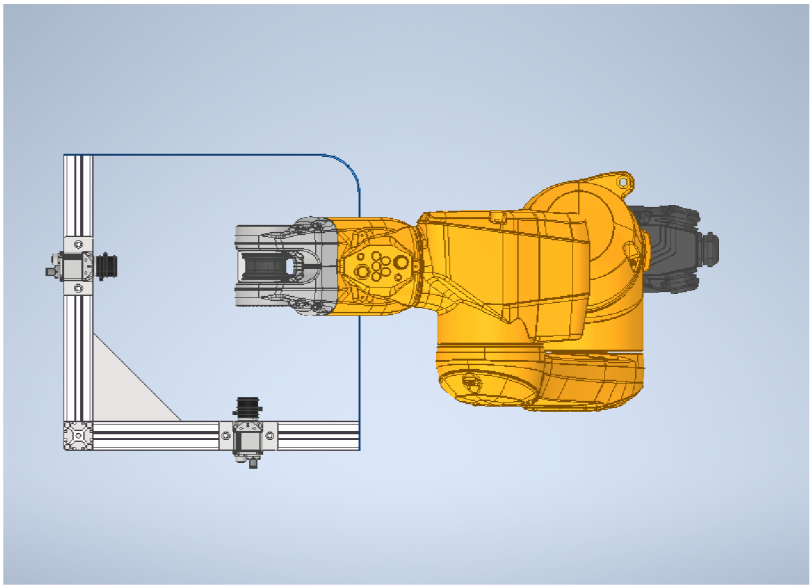
**C Výpis použitého PSO algoritmu** **85**

**D Obsah přiloženého CD** **87**

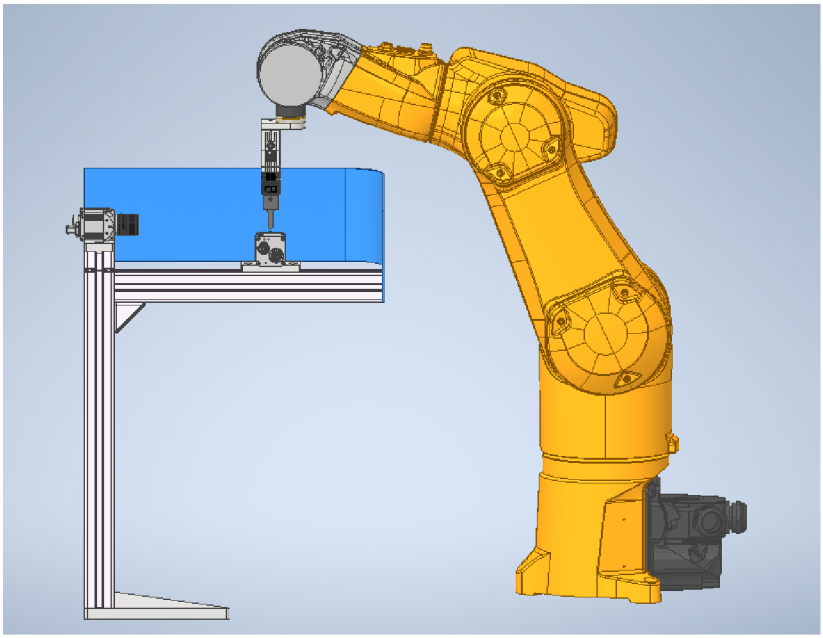
# A Návrh kalibračního přípravku



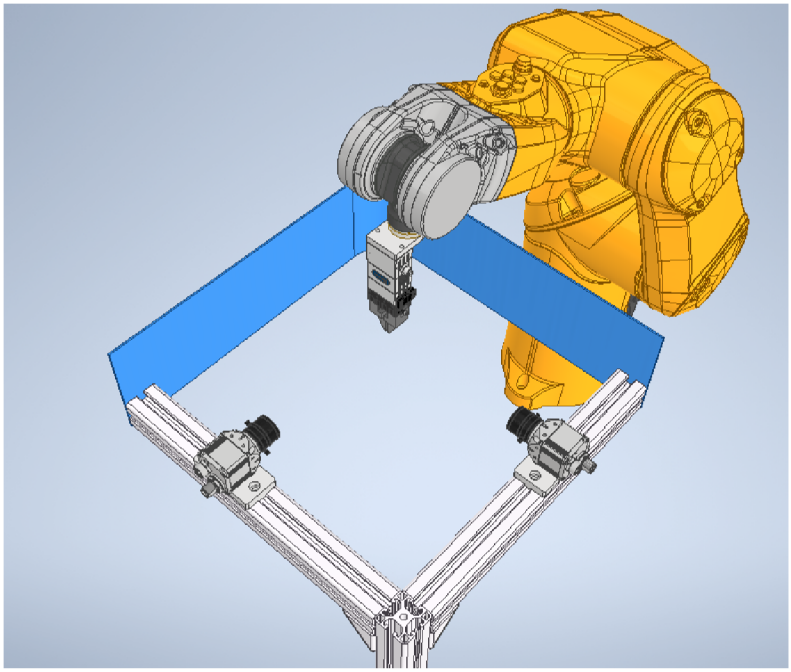
Obr. A.1: Nárys návrhu přípravku



Obr. A.2: Půdorys návrhu přípravku



Obr. A.3: Bokorys návrhu přípravku



Obr. A.4: Celkový náhled návrhu přípravku

# B Realizace kalibračního přípravku

Obr. B.1: Horní pohled na přípravek

Obr. B.2: Čelní pohled na přípravku

# C Výpis použitého PSO algoritmu

Výpis C.1: Algoritmus PSO

```
//Initialization
for(i=0;i<S;i++){
    x[i] = rand();
    best[i] = x[i];
    v[i] = zeros();
    fitness[i] = calculate_new_fitness(x[i]);
    best_fitness[i] = fitness[i];
}
for(i=0;i<S;i++){
    if(p[i] < g){
    g = p[i];
    best_global_position = x[i];
    }
}
//Main loop
while(iteration=0;iteration<max_iteration;iteration++){
    for(i=0;i<S;i++){
        r_p = rand();
        r_g = rand();
        v[i] = calculate_new_speed();
        x[i] = calculate_new_position();
        fitness[i] = calculate_new_fitness(x[i]);
        if(fitness[i] < best_fitness[i]){
            best_fitness[i] = fitness[i];
            best[i] = x[i];
        }
    }
    for(i=0;i<S;i++){
        if(p[i] < g){
            g = p[i];
            best_global_position = x[i];
        }
    }
    if(global_best_founded()) break;
}
```

# D Obsah přiloženého CD

```
/ ..................................... kořenový adresář přiloženého CD
├── camera_configuration.xml ........... konfigurační soubor s konfigurací kamer
├── configuration.json ................. ukázkový konfigurační soubor kalibrační aplikace
├── tool.json........................... ukázkový soubor s výsledkem TCP kalibrace
├── source ............................. adresář se zdrojovými kódy aplikace
├── requirements.txt ................... soubor s nutnostmi pro spuštění kalibrace
└── README.md........................... READ ME s informacemi o spouštění aplikace
```