

Kolaborativní robot jako oponent ve hře Air Hockey

Bc. Vojtěch Birgus

Diplomová práce
2025



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav automatizace a řídicí techniky

Akademický rok: 2024/2025

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	Bc. Vojtěch Birgus
Osobní číslo:	A23342
Studijní program:	N0714A150006 Automatické řízení a informatika v průmyslu 4.0
Forma studia:	Prezenční
Téma práce:	Kolaborativní robot jako oponent ve hře Air Hockey
Téma práce anglicky:	Collaborative Robot as an Opponent in the Air Hockey Game

Zásady pro vypracování

1. Vypracujte literární rešerši na dané téma.
2. Vytipujte hrací stůl vhodný pro vision a umístění robotického manipulátoru.
3. Navrhněte vhodný vision systém pro detekci air hockey puku a predikování jeho trajektorie.
4. Systém připravte v simulačním prostředí RobotStudio.
5. Připravte program do robotického manipulátoru pro pasivní i aktivní odezvu robota na hru.
6. Výsledky analyzujte a vyhodnoťte úspěšnost robota.

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. KOLÍBAL, Zdeněk. Roboty a robotizované výrobní technologie. Brno: Vysoké učení technické v Brně – nakladatelství VUTUM, 2016. ISBN 978-80-214-4828-5.
2. ABB Product specification – CRB 15000: 3HAC077390-001. Revize N. 2024.
3. Technical reference manual RAPID Instructions, Functions and Data types: 3HAC065038-001. Revize M. 2024.
4. LI, You-Cheng a HUANG, Shih-Yen. Fast-moving object tracking in air hockey. Online. In: 2017 IEEE International Conference on Mechatronics and Automation (ICMA). IEEE, 2017, s. 1693-1697. ISBN 978-1-5090-6758-9. Dostupné z: <https://doi.org/10.1109/ICMA.2017.8016072>.
5. CHANG, Ching-Lung; CHEN, Shuo-Tsung; CHANG, Chuan-Yu a JHOU, You-Chen. Application of Machine Learning in Air Hockey Interactive Control System. Online. Sensors. 2020, roč. 20, č. 24. ISSN 1424-8220. Dostupné z: <https://doi.org/10.3390/s20247233>.

Vedoucí diplomové práce: **Ing. Ľuboš Spaček, Ph.D.**
Ústav řízení procesů

Datum zadání diplomové práce: **13. prosince 2024**
Termín odevzdání diplomové práce: **27. května 2025**

doc. Ing. Jiří Vojtěšek, Ph.D. v.r.
děkan



prof. Ing. Vladimír Vašek, CSc. v.r.
ředitel ústavu

Ve Zlíně dne **13. prosince 2024**

PROHLÁŠENÍ AUTORA DIPLOMOVÉ PRÁCE

Beru na vědomí, že

- odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb., v platném znění bez ohledu na výsledek obhajoby;
- diplomová práce bude uložena v elektronické podobě v univerzitním informačním systému a bude dostupná k nahlédnutí;
- jedno vyhotovení diplomové práce v listinné podobě bude ponecháno Univerzitě Tomáše Bati ve Zlíně k uložení;
- na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- podle § 60 odst. 1 autorského zákona má Univerzita Tomáše Bati ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- podle § 60 odst. 2 a 3 mohu užít své dílo – diplomovou práci – nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tj. k nekomerčnímu využití), nelze výsledky diplomové práce využít ke komerčním účelům;
- pokud je výstupem diplomové práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá; neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji, že

- jsem na diplomové práci pracoval(a) samostatně a použitou literaturu jsem řádně citoval(a); v případě publikace výsledků budu uveden(a) jako spoluautor;
- odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou obsahově totožné.
- při tvorbě této práce jsem použil nástroj generativního modelu AI ChatGPT (<https://chat.openai.com>) za účelem jazykové korektury, stylistických úprav a revize formulací textu. Po použití tohoto nástroje jsem provedl kontrolu obsahu a přebírám za něj plnou zodpovědnost.

Ve Zlíně, dne

.....

podpis autora

ABSTRAKT

Tato diplomová práce se věnuje návrhu a realizaci kolaborativního robota jako protivníka pro hru Air Hockey. Cílem bylo vytvořit systém, který pomocí strojového vidění a predikce pohybu puku umožní robotu reagovat v reálném čase na herní situaci a soupeřit s člověkem. V práci je popsán návrh vision systému s využitím OpenCV, metoda detekce puku a predikce jeho trajektorie, stejně jako komunikace a řízení robota značky ABB. Součástí řešení je i testování v simulačním a reálném prostředí. Projekt propojuje oblasti robotiky, počítačového vidění a reálné aplikace kolaborativních technologií. Výsledkem je funkční systém schopný aktivně reagovat a adaptovat se na dynamiku.

Klíčová slova: kolaborativní robot, air hockey, predikce trajektorie, řízení robota, EGM, Python

ABSTRACT

This thesis focuses on the design and implementation of a collaborative robot as an adversary for the game Air Hockey. The goal was to create a system that, using machine vision and puck movement prediction, allows the robot to react in real time to the game situation and compete with a human. The thesis describes the design of the vision system using OpenCV, the method of puck detection and trajectory prediction, as well as communication and control of the ABB robot. The solution includes testing in simulation and real-world environments. The project bridges the fields of robotics, computer vision and real-world applications of collaborative technologies. The result is a functional system capable of actively responding and adapting to dynamics.

Keywords: collaborative robot, air hockey, trajectory prediction, robot control, EGM, Python

Rád bych touto cestou poděkoval svému vedoucímu, Ing. Ľuboši Spačkovi, Ph.D., za jeho odborné rady, trpělivost a podporu při zpracování této práce. Poděkování patří také mé rodině a přátelům za jejich pomoc a morální podporu během celého studia.

OBSAH

ÚVOD.....	13
I TEORETICKÁ ČÁST	14
1 AIR HOCKEY	15
1.1 PRAVIDLA A STANDARDY	15
1.2 STYL HRY	17
1.2.1 Správný úchop hokejky	17
1.2.2 Útočné strategie	17
1.2.3 Obranná strategie	18
1.3 EXISTUJÍCÍ ŘEŠENÍ ROBOTICKÝCH SOUPEŘŮ V AIR HOCKEY	19
1.3.1 Historie a vývoj robotických systému pro air hockey	20
1.3.2 Moderní přístupy v konstrukci robotických soupeřů	22
2 STROJOVÉ VÍDĚNÍ	24
2.1 HISTORIE	24
2.1.1 Strojové vidění dnes	26
2.2 ROZDÍL MEZI STROJOVÝM VIDĚNÍM A POČÍTAČOVÝM VIDĚNÍM	27
2.3 ROZDĚLENÍ KAMER	27
2.3.1 Obrazové snímače a jejich typy	28
2.3.2 Řádkový snímač	28
2.3.3 Plošný snímač	29
2.4 PARAMETRY A PROVEDENÍ KAMER	31
2.5 OBJEKTIVY	35
2.6 VYHODNOCOVÁNÍ OBRAZU	36
2.7 ZPRACOVÁNÍ OBRAZU	37
2.8 KOMUNIKACE	37
2.8.1 TCP/IP	38
2.8.2 UDP/IP	39
2.9 KNIHOVNA OPENCV	40
2.9.1 Historie	40
2.10 ZPRACOVÁNÍ OBRAZU A NALEZENÍ PUKU	41
2.10.1 Filtrování barvy	41
2.10.2 Převod na binární obraz	41
2.10.3 Detekce objektu podle barvy pixelů	41
2.10.4 Určování polohy objektu pomocí HoughCircles	42
2.10.5 Určení polohy objektu nalezením středu kontury	43
2.10.6 Transformace souřadnic obrazu	43
3 KOLABORATIVNÍ A PRŮMYSLOVÍ ROBOTI	45
3.1 KOLABORATIVNÍ ROBOTI	45

3.2	PRŮMYSLOVÍ ROBOTI	46
3.3	ROZDÍLY MEZI KOLABORATIVNÍMI A PRŮMYSLOVÝMI ROBOTY	46
4	ABB.....	48
4.1	HISTORIE	48
4.1.1	Vybrané milníky společnosti ABB.....	48
4.2	ROBOTSTUDIO.....	48
4.2.1	Algoryx.....	49
4.2.2	Rapid	49
4.2.3	SmartComponent.....	49
II	PRAKTICKÁ ČÁST	50
5	POŽADAVKY	51
5.1	VÝBĚR STOLU PRO AIR HOCKEY	51
5.2	VÝBĚR ROBOTA	52
5.3	VÝBĚR VISION SYSTÉMU.....	54
5.3.1	Výběr kamery	54
5.3.2	Výběr objektivu.....	54
5.3.3	Výběr programovacího jazyku	56
6	NÁVRH VISION SYSTÉMU	57
6.1	VISUAL STUDIO CODE	58
6.2	POUŽITÉ KNIHOVNY.....	59
6.2.1	Numpy.....	59
6.2.2	Pypylon.....	59
6.2.3	Pillow.....	60
6.2.4	Deque.....	60
6.3	DETEKCE PUKU.....	60
6.3.1	Použité metody detekce.....	61
6.4	PREDIKCE TRAJEKTORIE	64
6.5	ŘÍZENÍ ROBOTA	67
6.5.1	Knihovna Socket	67
6.5.2	Knihovna pro EGM pohyb robota.....	68
7	SIMULAČNÍ PROSTŘEDÍ.....	70
7.1	VYTVOŘENÍ MODELŮ A PŘÍPRAVA SIMULAČNÍHO PROSTŘEDÍ	70
7.2	PROGRAM PRO ROBOTA	72
7.3	OTESTOVÁNÍ VISION SYSTÉMU V SIMULAČNÍM PROSTŘEDÍ	73
8	REÁLNÉ PROSTŘEDÍ	75
8.1	NÁVRH KONCOVÉHO EFEKTORU	75
8.2	KONSTRUKCE PRO KAMERU.....	75
8.3	PROGRAM PRO ROBOTA	77

8.4	OTESTOVÁNÍ VISION SYSTÉMU A ROBOTA V REÁLNÉM PROSTŘEDÍ.....	77
8.4.1	Nastavení parametrů kamery Basler	78
8.4.2	Detekce ArUco značek.....	78
9	VÝSLEDKY	81
9.1	PASIVNÍ I AKTIVNÍ HRA ROBOTA	82
9.2	HODNOCENÍ PŘESNOSTI DETEKCE A PREDIKCE	83
9.3	POROVNÁNÍ SIMULACE A REÁLNÉHO PROSTŘEDÍ.....	83
9.4	OMEZENÍ A NÁVRH NA ZLEPŠENÍ	83
9.5	AR A VR	84
	ZÁVĚR	85
	SEZNAM POUŽITÉ LITERATURY.....	86
	SEZNAM OBRÁZKŮ	90
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	92

ÚVOD

Současný rozvoj automatizace a inteligentních systémů otevírá nové možnosti využití robotiky i v oblastech, které nejsou přímo spojené s průmyslovou výrobou. Významným směrem je rozvoj kolaborativních robotů, kteří jsou schopni interagovat s lidmi nebo samostatně plnit úkoly v dynamickém prostředí. Jedním z netradičních, avšak technicky náročných úkolů je návrh robota schopného sehrát roli protivníka v rychlé a nepředvídatelné hře, jako je air hockey.

Diplomová práce se zabývá návrhem a realizací systému, který propojuje oblast počítačového vidění, predikce trajektorie objektu a řízení kolaborativního robota značky ABB. Cílem bylo vytvořit funkčního robotického soupeře, který dokáže v reálném čase reagovat na pohyb puku, předvídat jeho dráhu a efektivně se rozhodovat při hře.

Teoretická část práce se zaměřuje na popis pravidel a strategií hry air hockey, principy strojového a počítačového vidění a způsoby detekce objektů. Dále jsou zde uvedeny základní charakteristiky kolaborativních a průmyslových robotů, se zaměřením na roboty ABB a jejich programovací prostředí RobotStudio.

Praktická část popisuje návrh a realizaci vision systému pro detekci puku, jeho predikci a následné řízení pohybů robota. Součástí je i návrh simulačního prostředí, testování jednotlivých komponent a implementace v reálném prostředí s kamerou a robotem. Závěrečná kapitola pak shrnuje výsledky testování a dosažené poznatky.

I TEORETICKÁ ČÁST

1 AIR HOCKEY

Air hockey je soutěžní zábavná hra, která kombinuje inteligenci a reakční schopnosti. Herní prostředí se skládá ze stolu s brankami na každé straně stolu. Povrch stolu má rovnoměrně rozmístěné vzduchové otvory. Ventilátor pod stolem vytváří proudění vzduchu ze vzduchových otvorů, což umožňuje, aby plastový disk, hokejový puk, téměř klouzal po desce stolu bez tření. Hráč se pomocí plastové hokejky snaží odrazit puk do soupeřovy branky a zároveň zablokovat soupeřův útok. [1]

V postupu realizace této práce je potřeba zakládat na standardizovaných vlastnostech a pravidlech hry Air Hockey. Z hlediska herního stylu je zásadní, aby robotický hráč dodržoval pravidla a současně dokázal být rovnocenným soupeřem lidskému protihráči.



Obrázek 1 Stůl pro Air Hockey [2]

1.1 Pravidla a standardy

Dva hráči se utkávají na hrací ploše s jedním pukem a dvěma hokejkami. Hráči se mohou pohybovat kdekoliv na své polovině stolu, přičemž je zakázáno překračovat na soupeřovu stranu. Pohyb hokejky, stejně jako pohyb hráče, je limitován na jednu polovinu hrací plochy. V případě, že se puk nachází ve styku se středovou čarou, mají možnost jej zasáhnout oba hráči. Trajektorii puku je možné ovlivnit pouze kontaktem s hokejkou, jiné zásahy jsou

zakázány. Hra začíná vhazováním, kdy rozhodčí uvolní puk ve středu hřiště. Každé další rozehrání pak probíhá ze strany stolu, kde byla naposledy inkasována branka. Puk nesmí zůstat na jedné polovině hrací plochy déle než sedm sekund – pokud hráč v tomto čase nevystřelí, puk připadne soupeři. Hru vyhrává hráč, který jako první vstřelí sedm gólů (bodů), v některých případech lze hrát až do devíti bodů. [2]

Rozměry stolu pro hru jsou určeny pravidly asociace USSA (United States Air Hockey Association). Jedná se o stoly v plné velikosti (tzv. full-size 8' models), jejichž rozměry jsou následující: délka 217,17 cm, šířka 115,57 cm a výška 78,74 cm. Tyto rozměry zajišťují standardizaci vybavení pro profesionální turnaje a soutěže organizované touto asociací. [2]

Rozměry a specifikace hokejek (Mallets) jsou přesně definovány. Hokejky musí být schváleny rozhodčími hry tak, aby vyhovovaly pravidlům asociace USAA. Požadavky zahrnují následující: hokejka nesmí vážit více než 6 oz (170 g), její průměr musí být menší než 4-1/16" (10,32 cm). Barva hokejky může být libovolná, avšak její vnější okraj musí mít jinou barvu než herní povrch stolu. Palička musí být vyrobena z jednotného materiálu a musí být po celém obvodu stejnoměrná a symetrická. [2]

Dále je zakázáno jakékoliv upravování hokejek, které by naklánělo jejich hrací plochu a tím vytvářelo jakýkoli šikmý úderový nebo obranný povrch. [2]



Obrázek 2 Vzhled hokejek a puků pro Air Hockey

Další parametry, jako je velikost a hmotnost puku nebo hokejky, nejsou přesně definovány a závisí na výrobcí nebo na velikosti stolu, ke kterému jsou určeny. Průměry puků se obvykle pohybují od 45 mm do 90 mm a jejich hmotnost může dosahovat až 41 gramů. Pro turnajové a profesionální účely se většinou používají větší a těžší puky, aby se snížilo riziko ztráty kontaktu puku s hrací plochou při silnějších úderech. Hmotnost hokejky

není standardizována, těžší hokejky však umožňují prudší údery, a proto jsou často na turnajích omezeny na maximálně 170 gramů. [2]

1.2 Styl hry

Air hockey, známý svou rychlostí a dynamikou, vyžaduje promyšlenou strategii jak v útoku, tak v obraně. Pro dosažení úspěchu je klíčové ovládnout techniky správného úchopu hokejky, osvojit si účinné útočné i obranné taktiky a zároveň se naučit kontrolovat pohyb puku. [3]

1.2.1 Správný úchop hokejky

Správný úchop hokejky je klíčový pro optimální výkon. Nevhodný úchop, při kterém hráč drží hokejku celou rukou, může zpomalit pohyb a snížit přesnost úderu. Pokročilí hráči často drží hokejku tak, že umístí pouze několik prstů do prostoru uvnitř madla hokejky. Tímto způsobem držení hokejky dosahují lepší stability a kontroly než s běžným úchopem. Optimální úchop umožňuje hráči flexibilitu v rychlých změnách směru pohybu hokejky a lépe reagovat na různé herní situace. [3]



Obrázek 3 Správný úchop hokejky

1.2.2 Útočné strategie

Úspěšná útočná strategie spočívá v dovednosti zmást soupeře a využít jakékoliv příležitosti ke střelení gólu. Mezi klíčové útočné taktiky, které mohou výrazně zvýšit šance na úspěch, patří následující techniky:

- Driftování: Tato taktika spočívá v jemném pohybu puku na jednu stranu, následovaném rychlým výstřelem na opačnou stranu. Cílem je vyvést soupeře z

rovnováhy a využít momentu, kdy se soupeř nestihne vrátit do ideální pozice před brankou. [3]

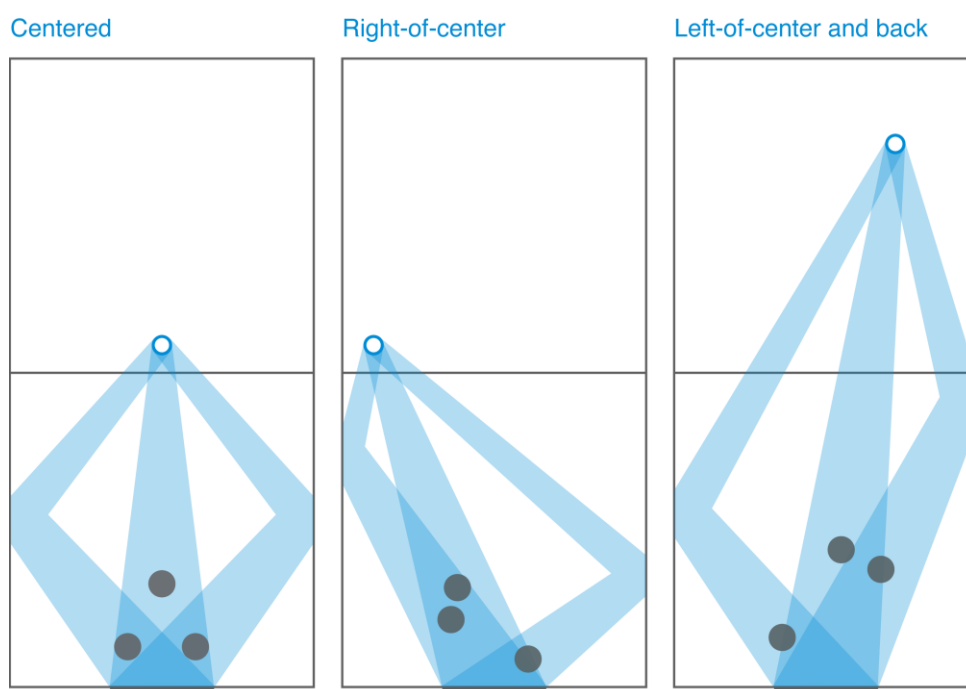
- Trikové střely: [3]
 - One-Two Shot: Tento typ střely zahrnuje úder do rohu soupeřovy branky, kde se puk odráží zpět k hráči. Jakmile se puk vrátí, je okamžitě vystřelen zpět na branku. Tato technika vyžaduje přesné načasování, ale je velmi obtížně čitelná pro soupeře.
 - Casting Cut Shot: Tato technika využívá momentu nepozornosti soupeře, typicky po jeho úspěšném gólu. Při pohybu puku je třeba použít hokejku k jeho vystřelení do opačného rohu branky soupeře. Jinými slovy, pokud je puk směřován doprava, cílem by mělo být zamířit do levého rohu soupeřovy branky. Pokud je tato technika provedena správně, soupeř bude soustředěný na to, kde puk klouže a nebude očekávat střelu do opačného rohu.
 - Moving Puck Shot (Chase Shot): Tento typ střely, známý také jako „Chase Shot“, je podobný technice „One-Two“ v tom, že využívá mantinely ke správnému nastavení střely. Hlavním cílem je zmást soupeře udržováním puku v pohybu podél mantinelů.

Dovednost udržet kontrolu nad pukem je základem pro kvalitní herní výkon. Namísto impulsivního a nepřesného úderu je žádoucí „zachytit“ puk jemným pohybem hokejky směrem dozadu. Hráč tak může získat lepší kontrolu nad dalším pohybem puku. Tento pohyb, označovaný jako „zachytávací“, poskytuje hráči čas na přemýšlení o dalším kroku a zajišťuje lepší pozici pro přípravu následného útoku nebo vykonání trikové střely. Důležitost kontroly puku spočívá v tom, že umožňuje hráči udržet tempo hry a využít příležitosti v optimálním okamžiku. [3]

1.2.3 Obranná strategie

Kvalitní obranná strategie je nezbytná pro rovnováhu mezi útokem a obranou v průběhu zápasu. Častou chybou začátečníků je tendence držet hokejku příliš blízko branky. Tímto způsobem vytváří prostor pro přímé střely. Hlavní obrannou technikou je takzvaná trianglová obrana: [3]

- Trianglová obrana: Tento obranný styl spočívá v umístění hokejky nad branku, přibližně v první třetině hrací plochy. Tato pozice umožňuje pokrýt tři hlavní útočné úhly: přímé střely, šikmé střely a střely odražené od mantinelu.
 - Přímé střely: Díky této pozici hokejky je hráč schopen blokovat přímé střely bez potřeby výrazných změn v pohybu.
 - Šikmé střely: Pro blokování šikmých střel stačí mírný pohyb hokejky na stranu.
 - Střely od mantinelu: Střely, které směřují na mantinel, vyžadují rychlou reakci a přesunutí hokejky směrem k mantinelu ve tvaru trojúhelníku, aby bylo možné zablokování střely.



Obrázek 4 Trianglová obrana [4]

1.3 Existující řešení robotických soupeřů v air hockey

Robotika v oblasti air hockey je předmětem výzkumu již několik desetiletí. Air hockey je ideálním testovacím prostředím pro vývoj robotických systémů, protože zahrnuje vysokorychlostní pohyb, rychlou reakci na dynamické změny a přesné plánování trajektorie. Výzvou v této oblasti je především velká nejistota dynamiky, hardwarové limitace a rychlosti robotického soupeře. [5]

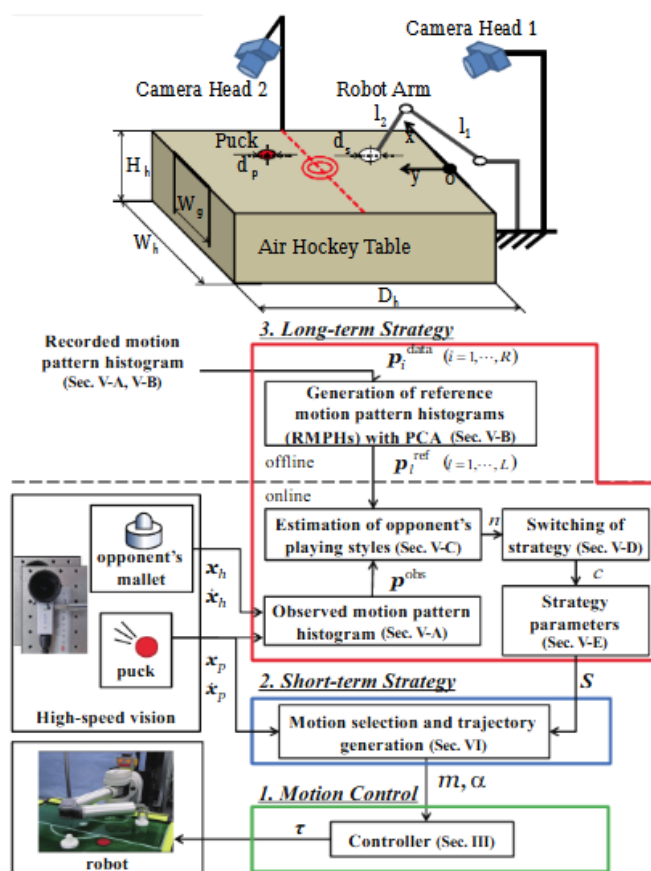
1.3.1 Historie a vývoj robotických systému pro air hockey

Jedním z prvních pokusů o vytvoření robotického hráče v air hockey bylo řešení s použitím tříčlánkového planárního redundantního manipulátoru řízeného kamerovým systémem. Tento výzkum se zaměřil na přesné sledování puku a implementaci základních zásahů. Následné práce se soustředily na vylepšení dynamiky hry a přizpůsobení taktiky proti lidským soupeřům. [5] [6]



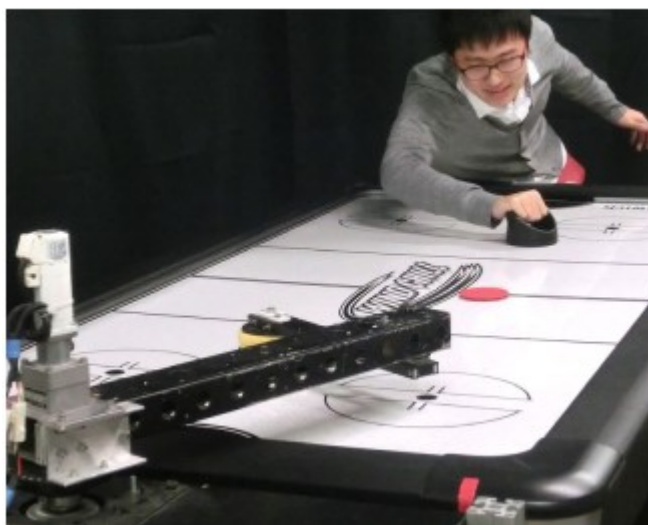
Obrázek 5 Tří článkový robotický hráč [6]

V roce 2013 byla představena hierarchická architektura řízení pro air hockey robota, která integrovala strategické rozhodování na vysoké úrovni s nízkoúrovňovým řízením pohybů. Tato práce položila základy pro sofistikovanější modely robotických hráčů. [5] [7]



Obrázek 6 hierarchická architektura řízení [7]

Další významné řešení bylo navrženo v roce 2017, kdy byla vyvinuta dvouvrstvá taktická architektura. Tento přístup umožňoval robotu přizpůsobit svůj herní styl na základě pozice puku a předpovědi pohybu soupeře. [5] [8]



Obrázek 7 Robotický soupeř v air hockey [8]

1.3.2 Moderní přístupy v konstrukci robotických soupeřů

Tradiční řešení robotických soupeřů pro air hockey byla založena převážně na průmyslových robotických ramenech. Tyto systémy poskytují vysokou přesnost a dynamiku pohybu, ale jejich implementace je často nákladná a vyžaduje složitou kalibraci. Alternativní přístupy se proto zaměřují na jednodušší a efektivnější kinematické systémy, jako jsou CoreXY a H-bot. [9]

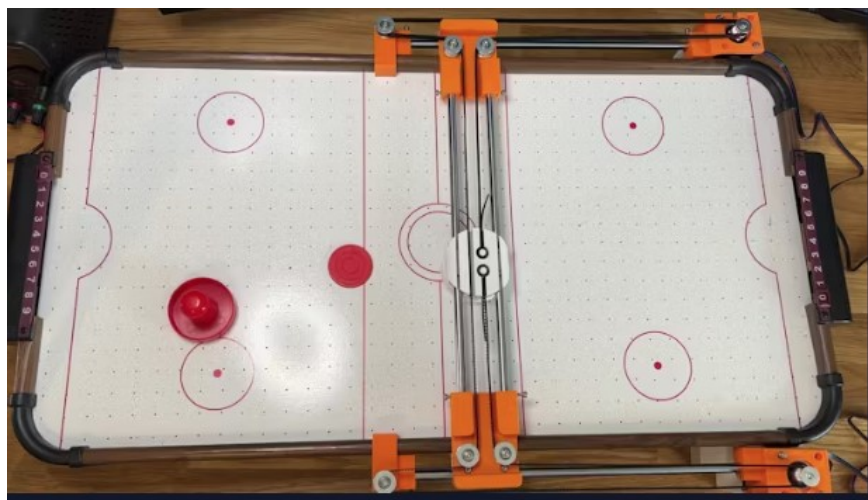
Moderní robotické hráče pro air hockey stále častěji pohání právě systémy, které pracují na principu dvou nezávisle řízených krokových motorů propojených řemenovým mechanismem. Tento přístup minimalizuje hmotnost pohyblivých částí a umožňuje vyšší rychlost oproti tradičním kartézským manipulátorům. Například diplomová práce Bc. Dominika Jaška (2020) popisuje návrh a realizaci robotického hráče založeného na H-bot kinematice. Tento systém využívá krokové motory NEMA 17 řízené pomocí Arduino Micro. Hlavní výhodou je schopnost rychlých reakcí na pohyb puku při zachování dostatečné přesnosti. [10]



Obrázek 8 Diplomová práce práce Bc. Dominika Jaška [10]

Novější vývoj robotických soupeřů se soustředí nejen na konstrukční vylepšení, ale také na zlepšení herní strategie. Příkladem může být robotický hráč představený v prosinci 2024, který využívá H-Bot konstrukci doplněnou o prediktivní algoritmy. Tento systém je

postaven na menším air hockey stole s pohonem krokových motorů a herní situaci monitoruje pomocí kamery připojené k Raspberry Pi 3B. Obraz je zpracováván knihovnou OpenCV, díky tomu systém umožňuje nejen sledování puku, ale i predikci pohybů soupeře. Systém tedy dokáže efektivně reagovat na strategii protivníka a přizpůsobit svůj herní styl. [9]



Obrázek 9 H-bot air hockey hráč [9]

Vývoj robotických soupeřů pro air hockey prošel během posledních desetiletí výrazným pokrokem. Počáteční experimenty se zaměřovaly na základní řízení pohybu a přesné sledování puku, zatímco moderní systémy integrují sofistikované algoritmy a prediktivní modely pro adaptivní herní strategii.

Historie ukazuje, že zásadním milníkem byl přechod od průmyslových robotických ramen k lehčím a efektivnějším kinematickým systémům, jako jsou H-bot a CoreXY. Tato konstrukční vylepšení umožnila nejen vyšší rychlost pohybu, ale také snížení nákladů a složitosti implementace.

Nejnovější přístupy kombinují optimalizovanou mechaniku s pokročilým zpracováním obrazu a umělou inteligencí. Robotické systémy dnes využívají prediktivní algoritmy, které analyzují trajektorii puku i pohyb soupeře, což jim umožňuje flexibilní herní taktiku. Tento trend naznačuje, že budoucnost robotického hraní air hockey bude směřovat ke stále inteligentnějším a autonomnějším systémům, které se budou schopny efektivně přizpůsobit dynamice hry a strategiím lidských soupeřů.

2 STROJOVÉ VÍDĚNÍ

Strojové vidění je zásadní součástí interakce digitálních systémů s reálným světem. Umožňuje automatizovaným systémům vidět součásti, výrobky, vzory, kódy nebo jiné objekty a využívat tyto informace k rozhodování. [11] [12]

Protože umožňuje kontrolovat, měřit a třídit vyráběné součásti a produkty, strojové vidění výrazně zvýšilo výkon a flexibilitu průmyslové automatizace. Strojové vidění to vše provádí vysokou rychlostí a s vysokou přesností tak, že se zlepšuje kvalita výrobků a snižuje množství odpadu. [11] [12]

Strojové vidění také umožňuje automatizovaným zařízením lokalizovat objekty, identifikovat je a ukládat informace o jejich materiálu, stavu, orientaci a dalších detailech pro pozdější analýzu. Tato data jsou pro továrny, které chtějí zvýšit efektivitu, velmi důležitá. [11] [12]

Mezi běžné aplikace rychlého sledování objektů patří sledování obličejů, sledování produktů na výrobních linkách, sledování pohybujících se objektů pro natáčení kamerou, mobilní taktické analýzy pro sportovce, rozpoznávání registračních značek u pohybujících se vozidel, sledování puku ve hře air hockey a další. [11] [12]

2.1 Historie

Historie počítačového vidění a souvisejících technologií sahá až do 30. let 20. století. V roce 1933 představila výzkumná skupina RCA pod vedením Vladimira Zworykina první praktický elektronický televizní systém. V roce 1948 publikoval Norbert Wiener knihu *Cybernetics*, ve které formuloval principy řízení a komunikace v elektronických, mechanických a biologických systémech. Tato práce významně ovlivnila vývoj technologií. [13]



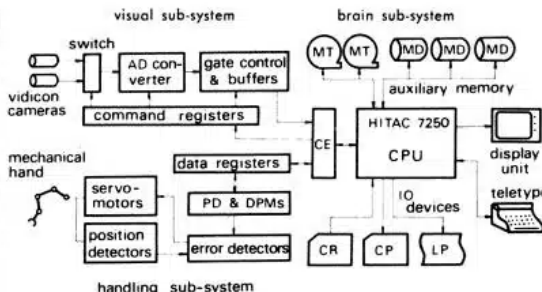
Obrázek 10 První elektronický televizní systém [13]

V 50. letech byly položeny základy zpracování obrazových dat: James J. Gibson v roce 1950 definoval optický tok a Russell Kirsch v roce 1957 vytvořil první digitální obraz. V roce 1963 publikoval Lawrence Roberts průlomovou práci o trojrozměrném vnímání objektů z 2D obrazů, která významně podnítila rozvoj výzkumu v oblasti počítačového vidění. [13]

V 70. a 80. letech došlo k prvním průmyslovým aplikacím. Hitachi R&D vyvinula prvního robota se systémem vidění (1970) a General Motors implementovala první systém strojového vidění ve výrobě (1981). V roce 1984 vznikla Asociace pro automatizované zobrazování (AIA), která později definovala standardy, jako Camera Link (2000) nebo GigE Vision (2006). [13]

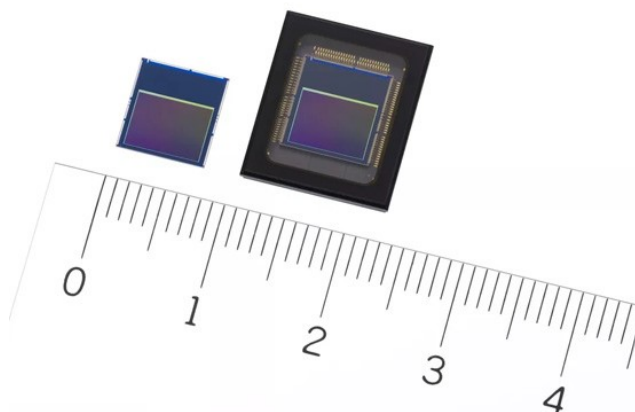
A Prototype Intelligent Robot that Assembles Objects from Plan Drawings

MASAKAZU EJIRI, TAKESHI UNO, HARUO YODA,
TATSUO GOTO, AND KIYOO TAKEYASU



Obrázek 11 HIVIP Mk.1, První inteligentní robot s vision systémem [13]

Od 90. let zaznamenal obor rychlý technologický pokrok. Například Sony vyvinula první CMOS snímáče (1996), byl uveden standard USB 3.0 (2008) a koncept CoaXPress (2008). V posledních letech se prosazují kamery s vysokorychlostními přenosy dat, jako 25 GigE (2018) nebo 100 GigE (2020) a obrazové snímáče s vestavěnou umělou inteligencí, například Sony IMX500 (2020). [13]



Obrázek 12 Sony IMX500 [14]

2.1.1 Strojové vidění dnes

Dnes je strojové vidění v podnikání využíváno pro širokou škálu úkolů, včetně kontroly kvality, identifikace, inspekce, navádění a dalších. Příklady použití zahrnují opravy vad na výrobních linkách, detekci a sklizeň plodin v zemědělství, správu zásob, sledování produktů v regulovaných průmyslech, měření a kalibraci a zlepšení bezpečnosti v různých odvětvích. Jak technologie strojového vidění stále pokračuje v pokroku, její využití se i nadále rozšiřuje

a nachází nové aplikace v různých oblastech, včetně robotiky, automatizace a zpracování dat. [15]

2.2 Rozdíl mezi strojovým viděním a počítačovým viděním

Ačkoli jak strojové vidění, tak počítačové vidění zahrnují zachytávání a analýzu vizuálních vstupů, mezi těmito technologiemi existují zásadní rozdíly. Systémy strojového vidění zachycují obrazy pomocí digitálních kamer, které následně zpracovávají a generují rozhodnutí. Tato rozhodnutí mohou zahrnovat například určení, zda produkt na výrobní lince prošel nebo neprošel kontrolou kvality na základě detekované vady. Systémy strojového vidění obvykle zahrnují kameru, objektiv, procesor a software, které umožňují stroji na základě analýzy obrazu přijmout rozhodnutí. Jinými slovy, strojové vidění musí být součástí širšího strojového systému. Na druhé straně, počítačové vidění může fungovat i samostatně, bez nutnosti být integrováno do fyzického zařízení. [16]

Jedním z klíčových rozdílů mezi počítačovým a strojovým viděním je skutečnost, že počítačové vidění není závislé na nutnosti zachytit nový obraz pomocí kamery, neboť dokáže pracovat s již uloženými obrazy. Počítačové vidění umožňuje interpretovat data z těchto obrazů a na jejich základě generovat výsledky nebo soubory výstupů. Na rozdíl od strojového vidění počítačové vidění nevyžaduje přímé použití kamery, díky čemuž jde o flexibilnější technologii. Umožňuje zpracovávat nejen reálné snímky, ale také synteticky vytvořené obrazy. Tato schopnost umožňuje počítačovému vidění analyzovat a extrahovat informace z obrazů, videí a dalších vizuálních zdrojů, zatímco strojové vidění se primárně spoléhá na obraz zachycený kamerou v reálném čase. [16]

Další rozdíl spočívá v odlišném zaměření obou technologií. Počítačové vidění se často využívá k extrakci a analýze co největšího množství dat o objektu nebo scéně, zatímco strojové vidění se zaměřuje na konkrétní části objektu či jeho klíčové vlastnosti, které následně zpracovává. Díky tomuto užšímu zaměření bývá strojové vidění preferováno v aplikacích vyžadujících rychlá a přesná rozhodnutí v kontrolovaných prostředích, jako jsou výrobní linky nebo automatizované inspekční systémy. [16]

2.3 Rozdělení kamer

Kamery používané pro vision systémy většinou poskytují větší výkon než standardní kamery používané běžnými spotřebiteli. Pro příklad, tyto typy kamer nabízejí vyšší rychlost závěrky

a snímkovou frekvenci pro rychlejší fungování a používají robustnější obrazové snímače pro průmyslové použití. Existuje mnoho typů kamer používaných pro strojové vidění. [17]

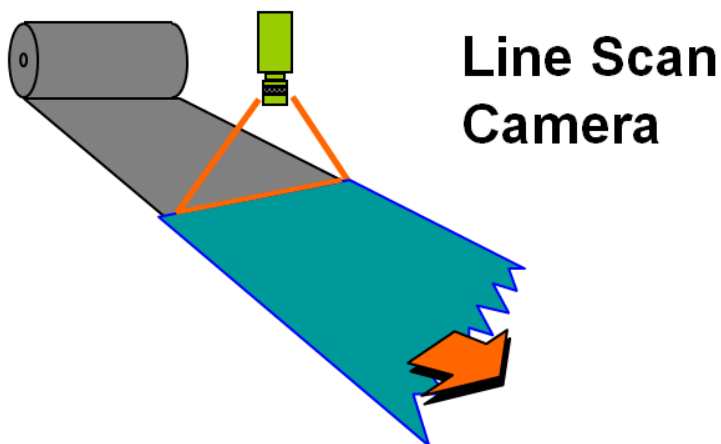
2.3.1 Obrazové snímače a jejich typy

Obrazový snímač je zařízení, které detekuje a přenáší data využívaná k vytvoření obrazu. Tento proces probíhá převodem proměnlivého zeslabení světelných vln, které mohou být světelné nebo jiné formy elektromagnetického záření, na elektrické signály, jež představují informace o obraze. Snímače přitom často využívají fotoelektrický jev k detekci světelných vln, které dopadají na senzory a uvolňují elektrony, které jsou následně přeměněny na elektrické napětí. [18]

Obrazové snímače se podle způsobu snímání dělí na dva hlavní typy: řádkové a plošné snímače. Kromě toho je lze klasifikovat také podle technologie záznamu obrazu na CCD (Charge-Coupled Device) a CMOS (Complementary Metal-Oxide-Semiconductor). [18]

2.3.2 Řádkový snímač

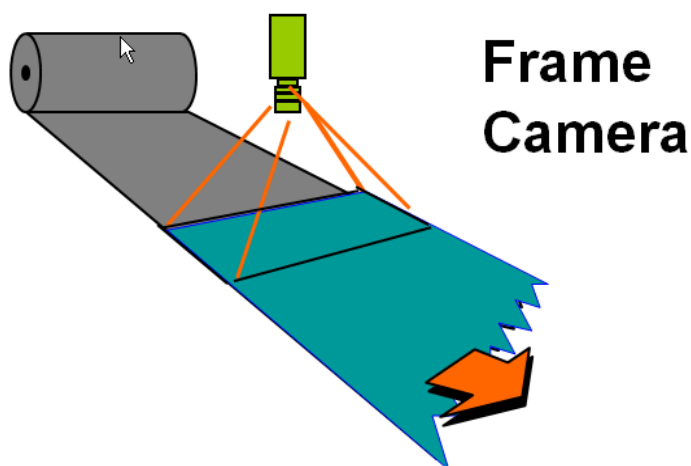
Řádkové snímače zaznamenávají pouze jeden rozměr obrazu, tedy jednotlivé řady pixelů. Tento typ snímače vytváří obraz postupným skenováním po jednotlivých řádcích, přičemž druhý rozměr obrazu je doplňován softwarově, a to buď pohybem kamery, nebo pohybem sledovaného objektu (například na dopravníku). Mezi hlavní výhody řádkových snímačů patří kompaktní rozměry kamer a schopnost pořizovat ostrý obraz i při vysokých rychlostech pohybu objektu nebo kamery. Nevýhodou je nutnost přesné synchronizace pohybu objektu nebo kamery s maximální rychlostí skenování pixelových řad. Tato synchronizace je klíčová pro zachování optimální snímkovací frekvence. [19]



Obrázek 13 Řádkový snímač [19]

2.3.3 Plošný snímač

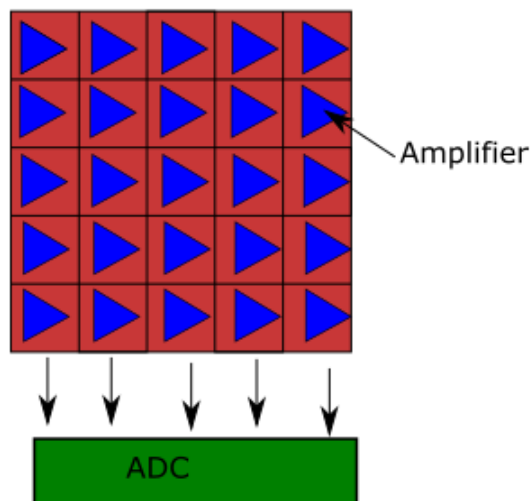
Plošné snímače na rozdíl od řádkových snímačů zachycují celý obraz ve 2D najednou. Tato technologie má výhodu větší univerzálnosti, protože umožňuje okamžité pořízení kompletního obrazu bez nutnosti pohybu objektu nebo kamery. Nevýhodou je, že sledovaný objekt musí být alespoň na krátkou dobu v klidu. Plošné snímače jsou často využívány v průmyslové automatizaci, kde je požadována rychlá a spolehlivá detekce obrazu. Mezi plošné snímače patří CCD a CMOS senzory. [20]



Obrázek 14 Plošný snímač [19]

2.3.3.1 CCD snímač

Technologie CCD (Charge-Coupled Device) je jedním z nejběžnějších typů obrazových snímačů. Tento snímač využívá fotoelektrický jev, kdy fotony dopadající na snímač uvolňují elektrony z atomů v materiálu snímače. Elektrony se shromažďují v jednotlivých pixelech a vytvářejí elektrický náboj, který reprezentuje daný pixel. Tento náboj je na výstupu zesílen a převeden na elektrické napětí. Hlavní výhodou CCD snímačů je jejich vysoká citlivost na světlo, díky které lze dosáhnout kvalitního obrazu i při méně intenzivním osvětlení. Zároveň umožňují rychlý přenos signálu s minimálním šumem, takže jsou vhodné pro aplikace vyžadující vysokou kvalitu obrazu. [21] [22]

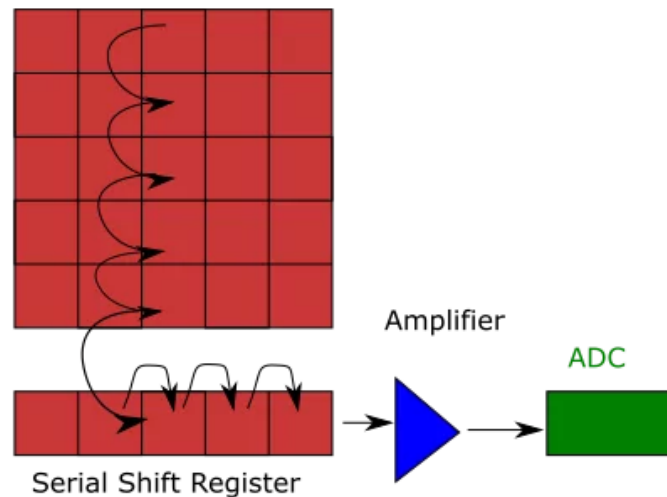


Obrázek 15 Princip CCD snímače [22]

Mezi nevýhody CCD technologie patří vyšší cena způsobená náročnou výrobou a implementací této technologie do kamer. Dalším nedostatkem je jev zvaný „blooming“. Tento jev nastává při snímání velmi světlých objektů a spočívá v interferenci mezi náboji v jednotlivých pixelech. Interference nábojů v pixelech vede ke zkreslení obrazu. [22] [23]

2.3.3.2 CMOS snímač

Technologie Complementary-Metal-Oxide-Semiconductor (CMOS) je jednou z metod pro zachytávání obrazu, která využívá fotoelektrický jev, podobně jako technologie Charge-Coupled Device (CCD). Při dopadu fotonů na snímač dochází k uvolnění elektronů z valenčních sfér materiálu snímače. V případě CMOS technologie se elektrony na jednotlivých pixelech nehromadí, protože každý pixel je vybaven vlastním zesilovačem, který umožňuje okamžitý převod uvolněného elektronu na napětí přímo u fotodiody. Mezi hlavní výhody CMOS technologie patří nižší spotřeba energie, která je důsledkem nižšího napájecího napětí, a nižší výrobní náklady, které jsou umožněny sériovou produkcí. Naopak, mezi její nevýhody patří nižší kvalita obrazu při rychlém pohybu objektu a vyšší hladina šumu při slabém osvětlení.



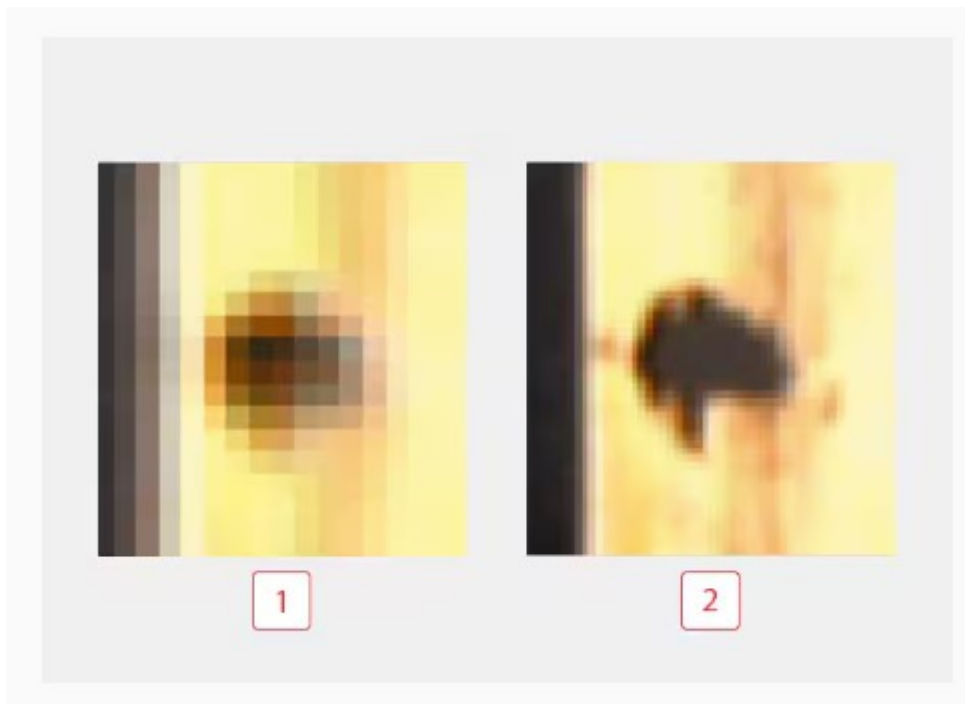
Obrázek 16 Princip CMOS snímače [22]

Tato různorodá škála obrazových snímačů, ať už řádkových, plošných, CCD nebo CMOS, umožňuje široké využití v různých aplikacích strojového vidění a počítačového vidění. Výběr konkrétního typu snímače závisí na specifických požadavcích aplikace, jako je rychlost snímání, požadavky na osvětlení nebo přesnost snímače. [22]

2.4 Parametry a provedení kamer

Při výběru kamer pro průmyslové aplikace je nutné zohlednit celou řadu parametrů, které se liší v závislosti na specifických požadavcích a zaměření dané úlohy. Mezi klíčová kritéria patří rozlišení, snímací frekvence, barevná škála a rozsah barevného spektra. [24]

Rozlišení kamery je měřeno počtem pixelů na snímacím čipu a obvykle se vyjadřuje v megapixelech (Mpx). Vyšší rozlišení znamená schopnost kamery zachytit jemnější detaily obrazu. Tato vlastnost je klíčová pro přesné zobrazení tvarů a struktur, zejména v aplikacích vyžadujících vysokou úroveň detailu. [25]



Obrázek 17 Pohled na částici v zorném poli v různém rozlišení [24]

Snímací frekvence je dalším důležitým parametrem vyjadřujícím počet snímků, které kamera dokáže pořídit za jednu sekundu (fps - frames per second). Frekvence snímání je klíčová pro aplikace, které vyžadují sledování rychle se pohybujících objektů, jako je detekce vad na rychlých výrobních linkách. Standardní kamery pracují v rozmezí od několika snímků za sekundu až po tisíce snímků za sekundu u specializovaných modelů. Pokročilé technologie mohou dosahovat frekvencí v řádu milionů snímků za sekundu, ale tyto kamery obvykle nevyužívají běžné snímače. [25]

Kamery lze rozdělit také podle barevné škály na monochromatické a barevné. Monochromatické kamery jsou preferovány v aplikacích, kde je důležitá přesná identifikace tvarů objektů, protože poskytují vyšší kontrast a lepší rozlišení detailů než kamery barevné. Na druhé straně barevné kamery jsou nezbytné pro aplikace, které vyžadují rozpoznání barev, například při kontrole správnosti barvy výrobků. [26]

Dalším faktorem je rozsah barevného spektra, který kamera dokáže zachytit. Některé aplikace vyžadují specifické spektrum světla pro správnou detekci objektů. Například detekce průhledných materiálů, jako je sklo, bývá efektivnější při použití ultrafialového (UV) světla, které zvýrazní tyto objekty, které jsou běžně těžko detekovatelné ve viditelném spektru. [24] [27]

Na současném trhu existuje široká nabídka kamerových systémů, které se neustále vyvíjejí a vybavují novými funkcemi a technologiemi. Kamery lze dělit podle typu snímacího prvku, přičemž základními kategoriemi jsou plošné, řádkové, 2,5D a 3D kamery. [24] [27]

Plošné kamery jsou nejběžnějším typem a dělí se na několik podkategorií. První jsou standardní kamery, které jsou specifikovány na základě parametrů, jako je rozlišení, snímací frekvence, barevná škála a rozsah spektra. Dalším typem jsou inteligentní (smart) kamery, které integrují zpracování obrazu přímo do kamery. Tyto kamery nejen snímají obraz, ale také poskytují konkrétní informace, protože obsahují minipočítač a odpovídající software pro analýzu dat. Výhodou je kompaktnost a schopnost nahradit více senzorů či čidel, čímž se zjednodušuje celkový systém. V poslední době se rychle vyvíjejí, přičemž zejména jejich softwarová část získává na významu. [24] [27]



Obrázek 18 Keyence Inteligentní průmyslová smart kamera [24]

Další podkategorií jsou kamerové senzory, označované také jako video senzory. Tyto kamery jsou zjednodušenou verzí inteligentních kamer a poskytují jednodušší výstupy, například binární informace typu „dobrý/špatný“ nebo „ano/ne“. Tyto senzory bývají využívány tam, kde není potřeba složitá analýza obrazu. [24] [27]

Řádkové kamery se liší v tom, že jejich snímací body jsou uspořádány v jedné nebo několika řadách. Tento design umožňuje zachytit pouze úzký záběr a obraz se skládá řádek po řádku při pohybu objektu nebo kamery. Tento typ kamer se vyznačuje vysokým rozlišením a snímací frekvencí, avšak pro správnou funkci vyžaduje konstantní pohyb objektu vůči kameře. Tato podmínka omezuje jejich využití na konkrétní aplikace, například skenery na dopravních pásech nebo tiskárny. [24] [27]

2,5D kamery kombinují 2D snímání s dalším hardwarem, který poskytuje informace o vzdálenosti. Tento typ senzoru, označovaný jako „fusion sensorika“, kombinuje data z různých kamer nebo senzorů a umožňuje analýzu objektů ve 2,5D. Tato metoda je užitečná při hrubém rozpoznání hloubky. [28]

3D kamery používají tři hlavní metody pro tvorbu 3D obrazu. První metodou je laserová triangulace, která spojuje kameru s vysokorychlostním snímačem a liniovým laserem. Tato metoda umožňuje přesné a rychlé skenování povrchů objektů. Další metodou je Time-of-Flight (ToF) kamera, která měří čas, jak dlouho trvá, než světlo odražené od objektu dosáhne zpět do kamery. Tato technologie je ideální pro analýzu statických objektů, avšak má nižší přesnost a vyšší náklady. Poslední metodou jsou stereo kamery, které používají dvě kamery umístěné v určité vzdálenosti, aby vytvořily 3D obraz pomocí analýzy rozdílů mezi snímky. Tento systém čelí problémům při detekci objektů s jednoduchou texturou, které lze řešit přidáním dodatečného osvětlení. [29]



Obrázek 19 ToF kamera [30]

Každý z těchto kamerových systémů nabízí unikátní možnosti a nachází uplatnění v různých průmyslových odvětvích, přičemž konkrétní výběr závisí na specifických požadavcích aplikace a pracovním prostředí.

2.5 Objektivy

Úkolem objektivu kamery je efektivně přenést světelné záření na snímač. Na snímači tak vzniká obraz, který je následně dále zpracováván. Objektivy mohou být tvořeny jednou čočkou nebo sestaveny z více čoček. Tato konfigurace přímo ovlivňuje jejich optické vlastnosti a konstrukci. Zaostření obrazu je obvykle nastavováno pomocí kroužku na objektivu, který umožňuje uživateli upravit ostrost podle aktuálních požadavků. V mnoha objektivěch je rovněž integrována mechanická clona, jež reguluje množství světla procházejícího objektivem a tím i expozici snímaného obrazu. [31]



Obrázek 20 Objektivy pro průmyslové kamery [31]

Jedním z klíčových parametrů objektivu je ohnisková vzdálenost, která určuje úhel záběru pro konkrétní velikost snímače. Ohnisková vzdálenost výrazně ovlivňuje perspektivu a zorný úhel pořízeného snímku. Většina objektivů používaných v průmyslových aplikacích má fixní ohniskovou vzdálenost, a tudíž nelze měnit jejich úhel záběru bez výměny samotného objektivu. Objektivy lze podle ohniskové vzdálenosti rozdělit do několika základních kategorií. [31] [32]

První skupinou jsou normální objektivy, jejichž ohnisková vzdálenost je obvykle přizpůsobena tak, že úhel záběru bývá $50^\circ - 90^\circ$. Tento úhel se blíží zornému poli lidského oka, a proto snímky pořízené těmito objektivy působí pro lidské vnímání nej přirozeněji. [31] [32]

Další kategorií jsou širokoúhlé objektivy, které mají kratší ohniskovou vzdálenost než normální objektivy. Tato konstrukce vede k širšímu úhlu záběru a umožňuje zachytit širší scénu. Širokoúhlé objektivy se běžně používají tam, kde je potřeba zachytit rozsáhlý prostor, například při fotografování krajin nebo architektury. [32]

Třetí skupinou jsou teleobjektivy, které disponují delší ohniskovou vzdáleností a mají užší úhel záběru. Tyto objektivy jsou ideální pro fotografování vzdálených objektů, jelikož umožňují „přiblížit“ objekty na velkou vzdálenost a nacházejí široké uplatnění zejména v přírodovědecké fotografii nebo při sportovních událostech. [32]

Kromě těchto základních typů objektivů existují i specializované objektivy, které jsou určeny pro konkrétní aplikace. Mezi ně patří makro objektivy, které umožňují zaostřit na velmi krátkou vzdálenost. Tyto objektivy jsou nezbytné pro fotografování drobných objektů, jako jsou hmyz nebo květiny, kde je vyžadováno zachování vysoké úrovně detailu i při velmi malých vzdálenostech mezi objektivem a objektem. [32]

Dalším specifickým typem objektivu je telecentrický objektiv, který je navržen tak, aby eliminoval perspektivní zkreslení. Tento typ objektivu je často používán v průmyslových aplikacích, kde je kladen důraz na vysokou přesnost při měření rozměrů a tvarů, například v optických měřicích systémech. [32]

Tato klasifikace objektivů ukazuje, jak různé konstrukce objektivů a jejich optické parametry ovlivňují způsob, jakým jsou obrazy zachycovány a interpretovány. Výběr objektivu je zásadní pro dosažení požadované kvality a charakteru snímků, přičemž každá kategorie objektivů má své specifické využití v závislosti na konkrétních potřebách aplikace.

2.6 Vyhodnocování obrazu

V průmyslových aplikacích je pro správnou funkci robotických systémů nezbytné, aby robot získával informace o poloze objektů, kterými má manipulovat. Je tedy důležité, aby robot přesně znal souřadnice objektu, například v rovině x a y , aby mohl objekt správně uchopit a manipulovat s ním. Tato informace je získávána prostřednictvím kamerového systému, jehož obrazová data jsou následně zpracována softwarovými algoritmy. Způsob, jakým jsou tato data zpracovávána, lze rozdělit do dvou hlavních kategorií.

První kategorií je kombinace kamery s počítačovým nebo vision systémem. Tento systém se skládá z kamery, která pořizuje obrazová data a řídicího systému, který na základě

předem definovaných kritérií provádí analýzu obrazových dat a předává výsledky zpět robotu. [33]

Druhou kategorií jsou inteligentní kamery, u nichž probíhá vyhodnocení obrazu přímo v samotné kameře. Tato zařízení jsou vybavena vestavěným počítačem, který umožňuje zpracování obrazových dat přímo v kameře, čímž se minimalizuje potřeba externího výpočetního zařízení. [34]

2.7 Zpracování obrazu

Současný vývoj kamerových systémů se soustředí především na extrakci relevantních informací z pořízených obrazů. Tento krok představuje jedno z klíčových úskalí v oblasti vision systémů, protože samotné pořízení obrazu je již technicky dobře zvládnutelné, zejména pro běžné objekty, které nejsou transparentní nebo lesklé. Extrakce informací je proces, který lze přirovnat k pyramidě informační hierarchie, kde na každé úrovni jsou získávány stále podrobnější a relevantnější informace pro další analýzu. [35]

Proces zpracování obrazu začíná u senzoru kamery, který převádí optické signály (například "šum") na elektrický signál. Tento signál je následně digitalizován pomocí A/D převodníku a převeden na digitální formu, která je reprezentována jako matice o rozměrech $U \times V$, představující souřadnice jednotlivých pixelů obrazu. Výsledný digitální obraz je často upravován pomocí transformačních funkcí, které mění parametry obrazu jako jas, kontrast, barevnou škálu nebo odstraňují šum. Tímto způsobem je dosaženo výstupního obrazu, který je vhodný pro další analýzu a vyhodnocení. [36] [37]

V oblasti robotického vidění se rovněž využívá matematická morfologie, která je aplikována zejména na nástroje zaměřené na detekci tvarů a objektů v obraze. Tato metoda se používá k analýze a zpracování geometrických tvarů a struktur v obrazových datech. Morfologické operace, jako jsou dilatace, eroze, otevření a uzavření, umožňují zpracovávat obrazové objekty i jejich vzory, a tím podstatně zvyšují robustnost a přesnost detekce v různých průmyslových aplikacích. [36]

2.8 Komunikace

Propojení kamery s řídicí jednotkou (například počítačem nebo průmyslovým robotem) či vision systémem a řídicí jednotkou je realizováno prostřednictvím různých komunikačních rozhraní.

2.8.1 TCP/IP

Protokol TCP (Transmission Control Protocol) je jedním z hlavních transportních protokolů sady internetových protokolů (IP). Představuje hlavní síťovou komunikační vrstvu pro poskytování spolehlivých doručovacích služeb. Jedná se o komunikační protokol orientovaný na spojení, který pomáhá při výměně dat mezi různými aplikacemi a zařízeními v síti. [38]

Pro řízení toku (flow control) TCP omezuje rychlost přenosu dat, aby se zajistilo, že příjemce zvládne přijímat data v odpovídajícím tempu. Tento mechanismus je klíčový pro zajištění spolehlivého doručení dat a prevenci přetížení. TCP rovněž implementuje mechanismus pro detekci a opravu chyb, který zajišťuje spolehlivý přenos dat i v případě ztráty nebo poškození některých segmentů. K tomu využívá metody jako je opakování ztracených segmentů a kontrolní součty [38]

Protokol TCP (Transmission Control Protocol) se využívá v širokém spektru internetových aplikací, kde je kladeno důraz na spolehlivý přenos dat: [38]

- World Wide Web (WWW): Při prohlížení webových stránek TCP zajišťuje spolehlivý přenos dat mezi webovým prohlížečem a webovými servery. Tento protokol je klíčový pro správné načítání stránek, kde je potřeba, aby všechny požadované součásti (texty, obrázky, skripty) byly správně doručeny bez chyb.
- Email: TCP se využívá při odesílání a přijímání e-mailů. Protokoly jako SMTP (Simple Mail Transfer Protocol) využívají TCP pro spolehlivý přenos zpráv mezi servery a jejich doručení do poštovních schránek uživatelů.
- File Transfer Protocol (FTP): FTP je protokol pro přenos souborů, který spoléhá na TCP pro bezpečný přenos dat. Ať už jde o nahrávání nebo stahování souborů, TCP zajišťuje integritu dat a jejich bezchybné doručení i při přenosu velkých souborů.
- Secure Shell (SSH): Protokol SSH, který se běžně používá pro vzdálenou správu serverů, spoléhá na TCP pro šifrovanou komunikaci mezi klientem a serverem. TCP zajišťuje spolehlivý a bezpečný přenos dat.
- Streaming Media: Služby pro streamování médií, jako jsou Netflix, YouTube nebo Spotify, využívají TCP k přenosu videí a hudby. Protokol zajišťuje plynulé přehrávání tím, že spravuje segmenty dat a retransmise v případě ztráty dat, takže minimalizuje výpadky při přehrávání.

2.8.2 UDP/IP

Protokol UDP (User Datagram Protocol) je protokol transportní vrstvy. Protokol UDP je součástí sady internetových protokolů (IP), označované jako sada UDP/IP. Na rozdíl od protokolu TCP se jedná o protokol, který nevyžaduje potvrzení přijetí dat. Protokol UDP pomáhá navazovat spojení s nízkou latencí a odolností proti ztrátám v síti. UDP umožňuje komunikaci mezi aplikacemi a procesy. [38]

UDP (User Datagram Protocol) je protokol vhodný pro jednoduchou komunikaci typu request-response, zejména v případech, kdy velikost přenášených dat je malá a tudíž není nutná náročná kontrola toku a chyb. Dále se často uplatňuje v reálných aplikacích, které jsou schopny tolerovat nerovnoměrné zpoždění mezi jednotlivými částmi přijaté zprávy a nevyžadují potvrzení o doručení. [38]

Protokol UDP (User Datagram Protocol) se používá v různých aplikacích, kde je kladen důraz na rychlost přenosu a nízkou latenci i za cenu občasné ztráty dat: [38]

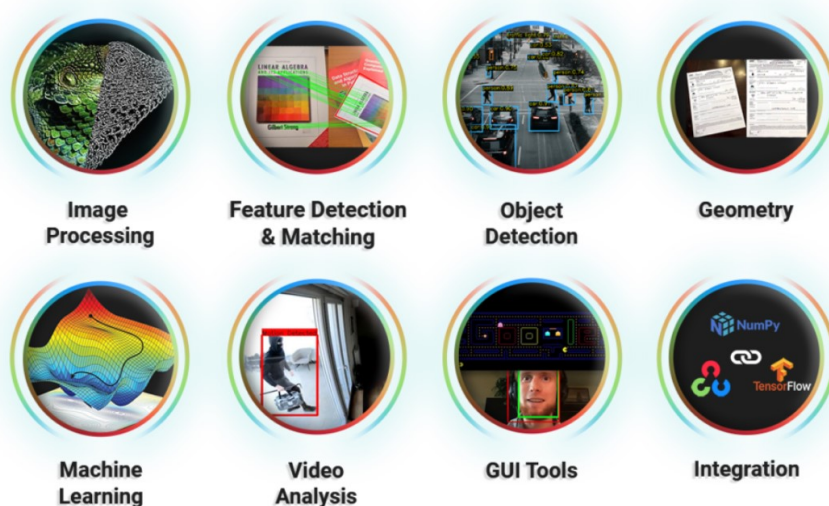
- Multimediální streamování v reálném čase: UDP je ideální pro streamování audio a video obsahu, protože jeho nízká latence zajišťuje plynulé přehrávání, i když dojde k občasné ztrátě dat. Tato vlastnost je zvláště užitečná u živých přenosů nebo při sledování on-demand videí a poslouchání hudby.
- Online hraní: Mnoho online her využívá UDP pro rychlou komunikaci mezi hráči. Tato aplikace vyžaduje minimální zpoždění při přenosu dat, které je zajištěno rychlým přenosem nabízeným protokolem UDP. Tato výhoda však může být vykoupena ztrátou některých paketů.
- Dotazy na DNS (Domain Name System): Při vyhledávání doménových jmen, například při převodu adresy "www.example.com" na IP adresu, UDP zajišťuje zpracování těchto požadavků. UDP je zde preferováno díky své rychlosti a nízké režii.
- Síťový monitoring: Nástroje pro monitorování výkonnosti sítě často používají UDP pro rychlou výměnu lehkých datových paketů. Tento protokol je výhodný pro aplikace, které vyžadují rychlé shromažďování informací o stavu sítě.
- Multicast: UDP podporuje paketové přepínání a díky tomu je vhodný pro multikanálové aplikace, v nichž je potřeba odesílat data více příjemcům současně.

2.9 Knihovna OpenCV

OpenCV (Open Source Computer Vision Library) je softwarová knihovna s otevřeným zdrojovým kódem pro počítačové vidění a strojové učení. OpenCV byla vytvořena s cílem poskytnout společnou infrastrukturu pro aplikace počítačového vidění a urychlení využití strojového vnímání v komerčních produktech. [39]

Knihovna je napsána v jazyce C++ a poskytuje rozhraní pro různé programovací jazyky, včetně Pythonu, Javy a MATLABu. [39]

OpenCV nabízí širokou škálu funkcí pro zpracování obrazu, jako je filtrování, detekce hran, segmentace, rozpoznávání objektů a sledování pohybu. Díky své modularitě a rozsáhlé dokumentaci je vhodná pro různé aplikace, od jednoduchých úloh až po komplexní systémy počítačového vidění. [39]



Obrázek 21 Možnosti využití knihovny OpenCV [39]

2.9.1 Historie

Projekt OpenCV, který byl oficiálně spuštěn v roce 1999, byl původně iniciativou společnosti Intel Research, jejímž cílem bylo zdokonalit aplikace náročné na výkon procesoru a byl součástí řady projektů zahrnujících ray tracing v reálném čase a 3D zobrazovací stěny. Mezi hlavní přispěvatele projektu patřila řada odborníků na optimalizaci v ruské pobočce společnosti Intel a také tým Performance Library společnosti Intel. [39]

V současné době je OpenCV ve verzi 4.x, která přináší vylepšení v oblasti výkonu, podpory hardwarové akcelerace a integrace s moderními frameworky pro strojové učení, jako je TensorFlow či PyTorch. Tato integrace umožňuje vývojářům snadno kombinovat

tradiční metody zpracování obrazu s pokročilými technikami hlubokého učení pro tvorbu výkonných a efektivních aplikací. [39]

2.10 Zpracování obrazu a nalezení puku

Poloha puku je detekována na snímcích, které jsou získávány kamerou. Podle vlastností kamery a typu je možno použít různé metody této detekce.

2.10.1 Filtrování barvy

Lokalizace objektu v obraze začíná rozpoznáním jeho základních charakteristik, které slouží k jeho identifikaci. Mezi tyto základní rysy patří především barva objektu. Toto je možné využít při převodu jednotlivých snímků na binární obraz, tedy na masku snímku.

2.10.2 Převod na binární obraz

Každému pixelu, jehož hodnota odstínu šedi spadá do stanoveného vyhledávacího intervalu, je nastavena binární hodnota 1. Pokud hodnota pixelu leží mimo tento rozsah, přiřadí se mu hodnota 0. Matice těchto hodnot tvoří masku snímku, kde pixely s hodnotou 1 odpovídají nalezenému objektu. Pozici hledaného objektu pak určuje umístění pixelů masky s hodnotou 1. Pokud by poloha těžiště neodpovídala pozici objektu, je nutné upravit vyhledávací parametry nebo přidat další kritéria, jako například minimální velikost shluku nalezených pixelů. [40]

Tato metoda, založená pouze na intenzitě jednotlivých pixelů, je výhodná při použití s černobílými kamerami. Díky jednoduchosti algoritmu jsou požadavky na výpočetní výkon nízké. Nicméně, použití pouze intenzity pixelů může být nevýhodné, pokud se v záběru kamery vyskytují parazitní objekty s podobnou šedou hodnotou jako hledaný objekt. [40]

2.10.3 Detekce objektu podle barvy pixelů

Před samotným filtrováním barev je nezbytné vybrat vhodný barevný model, který bude použit pro analýzu obrazových dat. Většina kamer standardně zobrazuje výstupní obraz v barevném modelu RGB (Red, Green, Blue). Na rozdíl od modelu RGB, je ale HSV model blíže tomu, jak lidé vnímají barvy. [41]

Z tohoto důvodu je možné použít pro tento typ operace například funkci z knihovny OpenCV, která umožňuje konverzi obrazu do jiných barevných modelů.

Barevný model HSV rozděluje barvu do tří hlavních složek: Hue (barvený tón), Saturation (sytost) a Value (jas). Hue označuje základní barevný tón, který je klíčovým parametrem pro určení barvy. Saturation vyjadřuje intenzitu barvy, tedy míru jejího zředění bílou barvou – vyšší saturace znamená čistší a živější barvu. Value se týká jasu barvy, přičemž vyšší hodnota znamená světlejší barvu, zatímco nižší hodnota odpovídá tmavší barvě. [41]

Výhodou použití modelu HSV v robotickém vidění je, že složka Hue (barevný tón) zůstává relativně stabilní i při změnách osvětlení, protože není ovlivněna intenzitou světla. Naopak, změny v osvětlení mají přímý vliv na složky Saturation a Value, což je výhodné při filtrování barev v různých světelných podmínkách. Tento model poskytuje lepší kontrolu nad detekcí specifických barev a umožňuje oddělení požadovaných barev od pozadí či jiných objektů v obraze.

2.10.4 Určování polohy objektu pomocí HoughCircles

Pokud je barva nebo odstín šedi hledaného objektu na snímku nejednoznačný, může být vhodné využít algoritmy zaměřené na detekci specifických tvarů. V případě, že se hledá objekt s určitou geometrickou formou, jako je například kruh, je jedním z vhodných nástrojů Houghova transformace. Tento algoritmus je široce používán v oblasti počítačového vidění pro detekci geometrických objektů, a to zejména pro detekci přímek a kruhů v obrazech. [42]

Houghova transformace pro detekci kruhů je využívána zejména v případech, kdy nelze spolehlivě pracovat s vizuálními vlastnostmi objektu, jako jsou barva, jas nebo textura. Tento algoritmus nevyžaduje přesnou znalost specifických charakteristik objektu a díky tomu je velmi robustní i při proměnlivosti těchto vlastností nebo jejich obtížné definovatelnosti. Pracuje na principu detekce geometrických parametrů, jako je například poloměr nebo střed kruhu. [42]

Tato metoda se osvědčuje zejména při detekci více objektů na jednom snímku a je často využívána v aplikacích, jako je sledování pohybujících se dopravních prostředků, detekce dopravních značek nebo automatizovaná kontrola vadných kusů v průmyslové výrobě. Houghova transformace umožňuje nejen detekovat jednotlivé objekty, ale také přesně určit jejich pozici v rámci obrazu. [42]

2.10.5 Určení polohy objektu nalezením středu kontury

Pro určení přesného středu kontury objektu je nezbytné provést několik předběžných kroků v procesu zpracování obrazu. Prvním krokem je převedení obrazu do stupňů šedi. Tím se zjednodušuje analýza a zároveň odstraňují barevné informace, které nejsou pro detekci tvaru relevantní. Následuje aplikace rozostření, obvykle pomocí Gaussova filtru, které pomáhá snížit šum a zajišťuje přesnější detekci hran objektu. Tento krok je klíčový pro zajištění hladkého přechodu mezi objektem a pozadím a zároveň zlepšuje následné kroky detekce. [43]

Následně je obraz převeden do binárního formátu, kde každý pixel získá jednu ze dvou hodnot: černou (pozadí) nebo bílou (objekt). V této fázi se často používá detekce hran, například pomocí Cannyho detektoru, společně s prahováním, které napomáhá oddělit objekt od pozadí. [43]

Po získání binárního obrazu je dalším krokem analýza kontur. Každý obrys je prozkoumán a na základě jeho geometrických vlastností, jako je plocha, tvar a orientace, se vypočítá střed obrysu. Tento střed, vypočítaný z geometrických vlastností objektu, slouží jako klíčový parametr pro určení jeho polohy, například v případě puku jeho střed. [43]

2.10.6 Transformace souřadnic obrazu

V závěrečné fázi zpracování obrazu je nezbytné upravit souřadnice lokalizovaného objektu, v tomto případě puku tak, aby odpovídaly souřadnicím definovaným pro konkrétní hrací plochu. Souřadnice získané z kamerového systému obvykle označují pozici pixelu, kde se nachází střed puku. Tyto hodnoty však závisí na orientaci kamery, která může být nakloněná nebo sledovat plochu pod určitým úhlem. Tento problém je třeba vyřešit, aby souřadnice odpovídaly reálnému uspořádání hrací plochy. [44]

Jedním z možných řešení je buď přesné nastavení kamery, vyžadující správnou kalibraci a pevné umístění, nebo aplikace geometrické transformace, která přepočítá souřadnice lokalizačního systému na odpovídající souřadnice hrací plochy. Tento přístup je výhodný, protože umožňuje flexibilitu při změně úhlu pohledu kamery a přitom zachovává konzistenci souřadnic bez nutnosti změny fyzického nastavení zařízení. [44]

Pro implementaci této transformace je potřeba provést manuální kalibraci, při které se lokalizují rohy hrací plochy. Pomocí těchto referenčních bodů lze následně vytvořit transformační matici, která bude přepočítávat souřadnice získané z kamery na souřadnice

hrací plochy. Tato transformační matice může být získána využitím funkce `perspectiveTransform` z knihovny OpenCV, která umožňuje provádět perspektivní transformace mezi dvěma souřadnicovými systémy. Po aplikaci této transformace budou souřadnice puku správně zmapovány na plochu a umožní tak přesnou lokalizaci objektu v reálném prostoru. [44]

3 KOLABORATIVNÍ A PRŮMYSLOVÍ ROBOTI

S postupujícím trendem automatizace v průmyslu se objevují nové technologie, které mění způsob, jakým jsou prováděny různé pracovní úkoly. Mezi klíčové inovace patří kolaborativní roboti (Coboti) a tradiční průmysloví roboti. Oba typy robotů se používají k zvýšení efektivity, produktivity a bezpečnosti, avšak každý z nich je navržen pro odlišné účely a má specifické vlastnosti, které určují jejich využití v konkrétních aplikacích. [45]

3.1 Kolaborativní roboti

Kolaborativní roboti, známí také jako coboti, jsou navrženi pro spolupráci s lidmi ve společném pracovním prostoru. Jsou vybaveni technologiemi, které umožňují bezpečnou interakci s lidskými pracovníky, takže jsou vhodné pro použití tam, kde je vyžadována flexibilita a přímá spolupráce. Coboti jsou obvykle kompaktní, lehkí a snadno přenosní, umožňující jejich rychlé přizpůsobení různým pracovním úkolům. Důležitým prvkem cobotů jsou senzory síly a točivého momentu, které umožňují detekci kontaktu s člověkem a automatické zpomalení nebo zastavení jejich pohybu, zajišťujících jejich bezpečnost. Mezi běžné aplikace cobotů patří montáž, manipulace s materiálem, kontrola kvality nebo obsluha strojů. Výhodou cobotů je také snadnost programování a jejich integrace do pracovního prostředí. [45]



Obrázek 22 Kolaborativní robot [46]

3.2 Průmysloví roboti

Naopak průmysloví roboti jsou navrženi pro autonomní vykonávání specifických úkolů, které nevyžadují přímý lidský zásah. Jsou obvykle větší, rychlejší a výkonnější než coboti a jejich aplikace jsou často zaměřeny na opakující se úkoly, které jsou nebezpečné, vysoce přesné nebo fyzicky náročné pro člověka. Průmysloví roboti se často nacházejí v uzavřených prostorech, kde jsou chráněni bezpečnostními bariérami nebo klecemi, minimalizující riziko zranění pracovníků. Mezi typické aplikace průmyslových robotů patří svařování, lakování, manipulace s těžkými materiály nebo balení. Tyto roboty jsou zpravidla složitější na integraci a programování, vyžadující specializované odborné znalosti. [45]



Obrázek 23 Průmysloví roboti [47]

3.3 Rozdíly mezi kolaborativními a průmyslovými roboty

Kolaborativní roboti (coboti) a průmysloví roboti se liší v několika klíčových aspektech, které ovlivňují jejich použití v praxi. Coboti jsou navrženi pro práci v přímém kontaktu s lidmi. Jsou vybaveni senzory pro detekci kontaktu, které umožňují automatickou reakci v podobě zpomalení nebo zastavení pohybu, zajišťující bezpečné pracovní prostředí. Na rozdíl od nich průmysloví roboti bývají obvykle odděleni od lidí fyzickými bariérami nebo ochrannými klecemi, které chrání pracovníky před možným zraněním. [45]

Další významný rozdíl spočívá ve flexibilitě a přizpůsobivosti těchto dvou typů robotů. Coboti jsou mnohem flexibilnější, jejich kompaktní konstrukce umožňuje snadné přemísťování a přizpůsobení různým pracovním úkolům. Programování cobotů bývá

intuitivní a nevyžaduje pokročilé znalosti robotiky a programování. Naopak průmysloví roboti bývají často specializováni na konkrétní úkoly, to znamená, že jejich přeprogramování a přizpůsobení novým prostředím může být časově náročné a vyžaduje odborné dovednosti. [45]

Pokud jde o výkon a kapacitu, průmysloví roboti mají obvykle vyšší nosnost a rychlost, proto jsou ideálními pro použití pro náročné úkoly. Coboti mají omezenější nosnost a jsou spíše určeni pro lehčí úkoly, kde je zároveň kladeno důraz na spolupráci s člověkem. [45]

Úroveň lidského zásahu se rovněž liší. Coboti jsou navrženi pro úzkou spolupráci s lidmi a často vyžadují lidskou pomoc při programování, údržbě a dokonce i při některých operacích. Naopak průmysloví roboti, jakmile jsou naprogramováni, mohou vykonávat úkoly autonomně po dlouhou dobu, přičemž lidská účast je potřebná převážně pro údržbu, opravy nebo řešení problémů. [45]

Pokud jde o aplikace v praxi, coboti jsou často využíváni v prostředích, kde je potřebná flexibilita a přímá spolupráce s lidskými pracovníky. Vhodní jsou pro montážní linky, logistiku, nebo zdravotnictví, kde je důležitá jejich přizpůsobivost a schopnost reagovat na změny. Průmysloví roboti se naopak osvědčují v aplikacích, které vyžadují vysokou přesnost, opakovatelnost a výkon, jako je svařování, manipulace s materiály, lakování nebo balení. Tyto úkoly často nevyžadují přímou interakci s lidmi, takže umožňují robotům vykonávat práci autonomně a efektivně. [45]

4 ABB

Společnost ABB, která sídlí ve švýcarském Zurichu, je lídrem v oblasti technologií zaměřených na elektrifikaci a automatizaci. Zaměřuje se na podporu udržitelné a efektivní budoucnosti. Cílem ABB je propojit inženýrské know-how se softwarem, což umožňuje optimalizaci výroby, pohybu a provozu produktů. Společnost má více než 130 let zkušeností a zaměstnává přes 105 000 lidí. [48]

4.1 Historie

Společnost ABB (Asea Brown Boveri) vznikla v roce 1988 sloučením dvou firem: ASEA, která vyráběla elektrické osvětlení a generátory, a BBC, specializované na výrobu AC a DC motorů, elektrických generátorů, transformátorů, parních a plynových turbín a vybavení pro lokomotivy. [48]

4.1.1 Vybrané milníky společnosti ABB

- 1975: Společnost ASEA představila prvního plně elektricky poháněného robota, řízeného mikrokontrolérem.
- 1990: Představení Azipod – elektrických pohonných systémů montovaných na vnější část lodí, poskytujících tah a řízení.
- 1998: Představení delta robota FlexPicker, navrženého pro aplikace sběru a balení.
- 2000: ABB jako první na světě uvedla síť shore-to-ship pro připojení lodí v přístavech ke zdroji elektrické energie z pevniny. Tento přístup vedl ke snížením emisí skleníkových plynů.
- 2012: Vývoj hybridního DC spínače pro velké mezuregionální DC sítě, který přinesl řešení technologické výzvy známé jako „válka proudů“.
- 2015: ABB představila prvního skutečně kolaborativního robota, který je určen ke spolupráci s lidmi a vybaven dvěma rameny.

4.2 RobotStudio

RobotStudio je aplikace pro modelování, offline programování a simulaci robotických buněk. Umožňuje optimalizaci výrobních procesů bez nutnosti přerušení výroby, čímž přispívá ke zvýšení produktivity. RobotStudio podporuje offline virtuálním ovladač běžící lokálně na počítači, a zároveň umožňuje připojení k fyzickému ovladači.

4.2.1 Algoryx

RobotStudio využívá simulační engine AGX Dynamics od firmy Algoryx pro fyzikální simulace v reálném čase. Tento engine zajišťuje věrné simulace dopravníkových pásů, manipulace s materiálem a interakce s virtuální realitou. [49]

4.2.2 Rapid

RAPID je programovací jazyk pro průmyslové roboty ABB, představený v roce 1994 spolu s řídicím systémem S4. Mezi jeho hlavní výhody patří snadná použitelnost a efektivita. Funkce jazyka zahrnují multitasking, práci s parametry rutin (procedury, funkce, trap routines), aritmetické a logické výrazy, automatické zpracování chyb a podporu modulárních programů. [50]

4.2.3 SmartComponent

SmartComponents v RobotStudiu umožňují simulovat reálné procesy, které se netýkají přímo programu nebo pohybu robota, jako je logika signálů, aritmetické operace, pohyby objektů a senzory. Pomocí těchto komponent lze vytvářet komplexní simulace, jako jsou pohyby koncového efektoru nebo přesuny objektů, s intuitivním ovládáním přes grafické rozhraní. [50]

II PRAKTICKÁ ČÁST

5 POŽADAVKY

Cílem této diplomové práce je vytvořit robotického oponenta pro hru air hockey. Robot by měl být schopen hrát jak pasivní obranu, tak i aktivní útok, tedy simulovat hru stejně, jako by hráč hrál proti lidskému protivníkovi. Zároveň bude potřeba navrhnout i vhodný vision systém pro detekci puku a výpočet jeho trajektorie. Celý tento systém by měl být navrhnut a otestován v simulačním prostředí, a poté být otestován i v reálném prostředí.

5.1 Výběr stolu pro Air Hockey

Základem této práce je samotný stůl pro air hockey. Bylo porovnáno několik komerčně dostupných stolů, přičemž hlavním rozhodovacím parametrem pro výběr byla cena. Ceny stolů pro profesionální hráče se mohou pohybovat ve výši několika desítek tisíc korun. Všechny nalezené stoly disponují podobnou konstrukcí, nicméně by bylo vhodné, aby k nim mohla být přimontována nosná konstrukce pro uchycení kamery.



Obrázek 24 Stoly pro Air Hockey [51]

Pro účely této práce byl zvolen stůl Buffalo Dominator 7FT, který se řadí mezi stoly pro domácí a nekomerční použití. Rozměry stolu odpovídají velikosti stolů pro profesionální turnaje. Tento vybraný stůl navíc nemá žádnou konstrukci nad hrací plochou, takže nic nebude blokovat výhled pro kameru. Jeho vnější rozměry činí 213 cm na délku, 122 cm na šířku a 81 cm na výšku, přičemž hrací plocha má rozměry 193 cm x 102 cm. Hmotnost 63

kg zajišťuje stabilitu stolu během hry. Napájení ventilátoru je zajištěno prostřednictvím sítě 230 V, zatímco elektronické počítadlo skóre vyžaduje čtyři AA baterie.



Obrázek 25 Vybraný stůl Buffalo Dominator 7FT [51]

5.2 Výběr robota

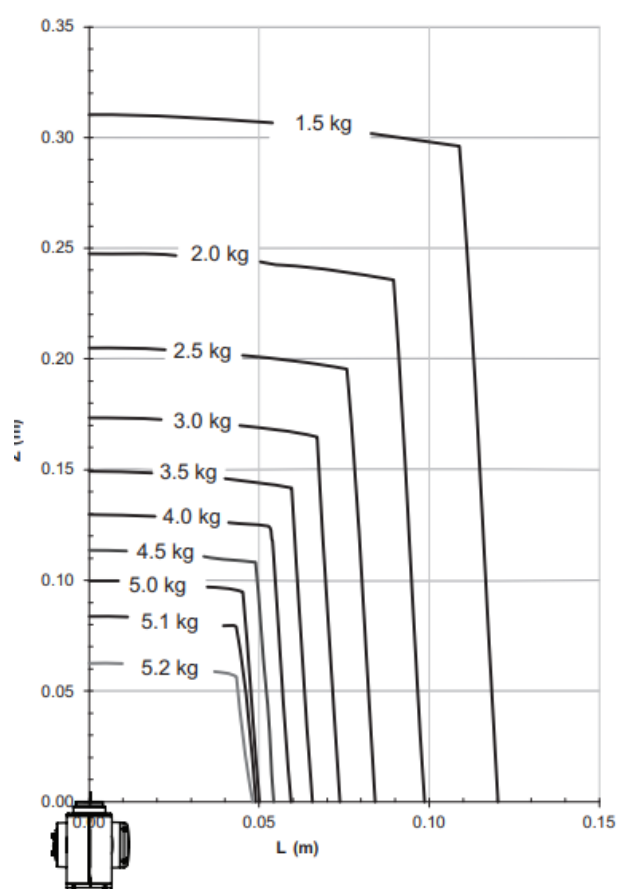
Pro tuto práci byl zvolen kolaborativní robot ABB GoFa™ CRB 15000, který představuje moderní a efektivní řešení pro automatizaci úloh vyžadujících těsnou spolupráci mezi člověkem a strojem. Tento model je navržen pro bezpečný provoz v bezprostřední blízkosti lidských operátorů, a to bez nutnosti fyzického oddělení pomocí bezpečnostních zábran či klecí. Tímto způsobem přispívá ke zvýšení flexibility a efektivity pracovního prostředí. [52]

Robot dosahuje maximální rychlosti až 2,2 m/s a je vybaven senzory krouticího momentu ve všech šesti osách. Tyto senzory umožňují přesné řízení síly a výkonu a v kombinaci s bezpečnostním systémem os zajišťují okamžité zastavení robota v případě neočekávaného kontaktu s člověkem, čímž výrazně snižují riziko úrazu. [52]

Hlavní výhodou využití tohoto typu robota v rámci řešení je možnost jeho umístění přímo k pracovnímu stolu bez nutnosti vymezeného bezpečnostního prostoru. To výrazně usnadňuje jeho integraci do stávajícího prostředí a umožňuje efektivní spolupráci s obsluhujícím pracovníkem. [52]



Obrázek 26 Kolaborativní robot ABB GoFa CRB 15000 [52]



Obrázek 27 Load diagram robota ABB GoFa CRB 15000 [52]

5.3 Výběr vision systému

K realizaci vision systému je zapotřebí několik klíčových komponent. Jako hlavní a základní součást systému je použita kamera. Pro účely simulace, ve kterých nebude možné využít reálnou kameru, bude její funkce nahrazena záznamem obrazovky, který bude sloužit jako zdroj obrazových dat pro následné zpracování.

5.3.1 Výběr kamery

Pro účely testování systému v reálném prostředí byla v rámci této práce vybrána průmyslová kamera Basler ace U acA1920-40uc. Jedná se o zařízení z řady Basler ace, které je vybaveno 2,3megapixelovým snímačem Sony IMX249 CMOS s globální závěrkou. Kamera poskytuje rozlišení 1920×1200 pixelů při maximální snímkové frekvenci až 40 snímků za sekundu.



Obrázek 28 Kamera Basler ace U acA1920-40uc [53]

5.3.2 Výběr objektivu

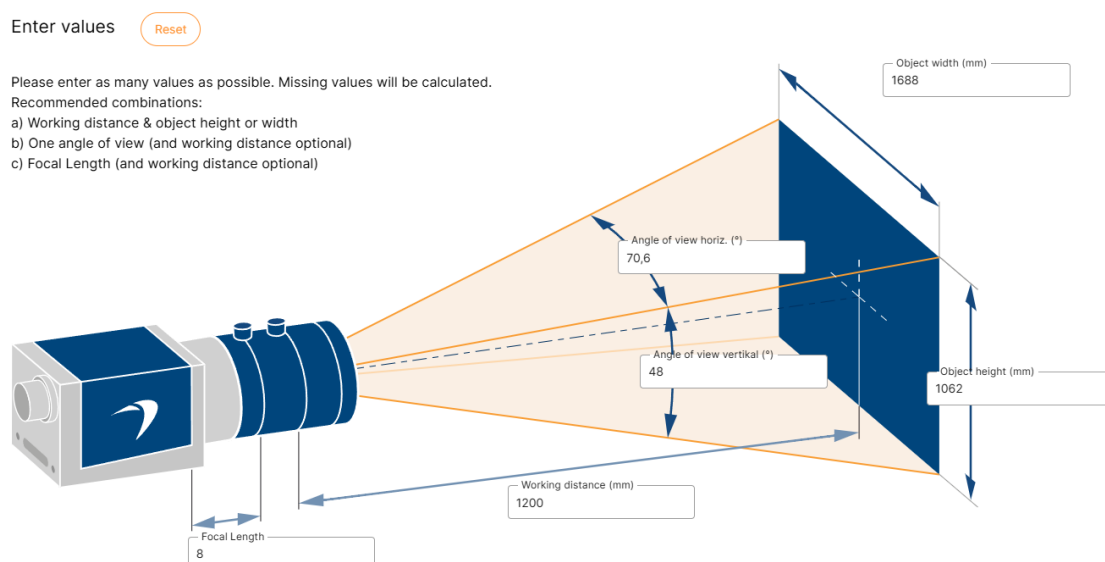
Dále byl vybrán objektiv Computar M0814-MP2. Jedná se o megapixelový objektiv s pevnou ohniskovou vzdáleností 8 mm, určený pro použití v průmyslových aplikacích, jako je strojové vidění a automatizace. Objektiv je kompatibilní se snímači o velikosti až 2/3" a dosahuje maximální světelnosti f/1.4, což umožňuje dosažení vysoké kvality obrazu i za zhoršených světelných podmínek.

Mezi technické specifikace objektivu patří minimální zaostřovací vzdálenost 0,1 m, manuální ovládání clony v rozsahu f/1.4 až f/16 a závit pro filtr s průměrem $M30,5 \times 0,5$ mm. Rozměry objektivu jsou $\varnothing 33,5 \times 28,2$ mm a jeho hmotnost činí 70 g.



Obrázek 29 Objektiv Computar M0814-MP2 [54]

Výběr tohoto objektivu a jeho parametrů byl proveden na základě výpočtu pomocí jednoduché online kalkulačky pro optické systémy. Na základě znalosti rozměrů hracího stolu bylo možné stanovit orientační vzdálenost kamery od hrací plochy a následně určit vhodnou ohniskovou vzdálenost objektivu tak, aby bylo dosaženo požadovaného zorného pole.



Obrázek 30 Výpočet parametrů objektivu pomocí online kalkulačky Basler

5.3.3 Výběr programovacího jazyku

Pro implementaci vision systému byl zvolen programovací jazyk Python. Tento jazyk nabízí širokou škálu volně dostupných knihoven určených pro zpracování obrazových dat. Pro detekci puku byla využita knihovna OpenCV, která je v oblasti počítačového vidění široce používaná a podporuje řadu funkcí potřebných pro analýzu obrazu v reálném čase.

Vision systém bude následně sloužit k řízení pohybů robota. Jako primární způsob komunikace mezi vision systémem a robotem byl zvolen přenos dat prostřednictvím protokolu TCP/IP. V této souvislosti bude nutné ověřit rychlost přenosu dat i následnou odezvu robota při provádění pohybů. Existuje možnost, že tento způsob komunikace nebude dostatečně rychlý a plynulý pro účely reálné hry. V takovém případě bude otestováno alternativní řízení robota pomocí technologie EGM (Externally Guided Motion).

Celý systém bude provozován na výpočetní jednotce (PC), která disponuje dostatečným výpočetním výkonem pro splnění požadavků jednotlivých komponent systému. Alternativně je možné systém nasadit i na kompaktních platformách, jako je například Raspberry Pi nebo jiné jednočipové počítače, což umožňuje flexibilnější a mobilní řešení.

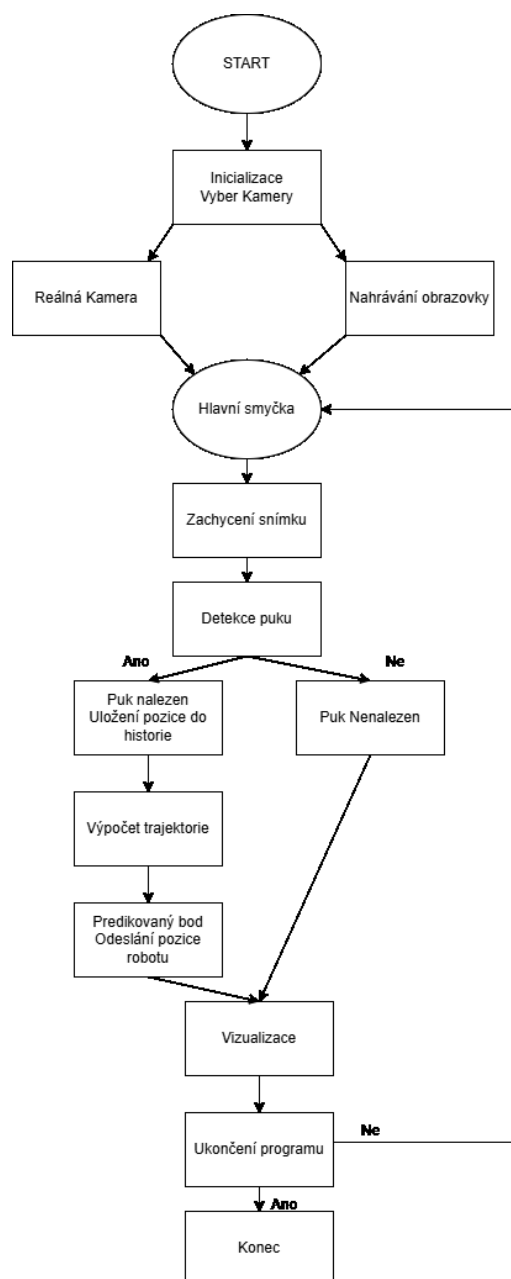
6 NÁVRH VISION SYSTÉMU

Jak již bylo uvedeno, pro zpracování obrazu a návrh vision systému byla využita knihovna OpenCV v kombinaci s programovacím jazykem Python. Navržený vision systém je koncipován tak, aby umožňoval detekci puku, predikci jeho trajektorie a následné řízení pohybů robota.

Systém zároveň umožňuje volbu mezi použitím obrazu z reálné kamery a simulovaného vstupu ve formě záznamu obrazovky z prostředí RobotStudio. Tato funkcionality je využívána zejména při testování v simulačním režimu.

Aplikace je navržena tak, aby poskytovala přehlednou vizualizaci predikované trajektorie puku, umožňovala nastavování jednotlivých parametrů systému a zároveň byla dostatečně rychlá pro zajištění plynulé a bezproblémové interakce robota s herním prostředím.

Základní struktura a návrh celého systému jsou znázorněny na následujícím vývojovém diagramu.



Obrázek 31 Vývojový diagram funkce vision systému

6.1 Visual Studio Code

Pro vývoj vision systému byl využit editor Visual Studio Code (VS Code), který patří mezi nejrozšířenější a nejuniverzálnější nástroje pro tvorbu softwaru. Tento editor, vyvíjený společností Microsoft, je volně dostupný a nabízí širokou škálu funkcí usnadňujících proces programování, jako je zvýraznění syntaxe, integrovaný terminál, podpora verzovacích systémů či možnost rozšíření prostřednictvím doplňků.

Mezi hlavní výhody VS Code patří:

- Podpora různých programovacích jazyků – i když byl projekt vyvíjen v Pythonu, editor umožňuje práci i s jinými jazyky.
- Možnost rozšíření funkcionality – prostřednictvím rozhraní Marketplace lze do editoru přidávat rozšíření, která zlepšují podporu konkrétních jazyků (např. Python), usnadňují správu závislostí nebo integrují nástroje pro práci s verzovacím systémem Git.
- Integrovaný terminál – editor obsahuje vestavěný terminál, který umožňuje spouštění skriptů a správu prostředí přímo v rámci vývojového prostředí.
- Ladicí nástroje – možnost krokového ladění kódu a sledování hodnot proměnných během běhu programu.
- Nízké systémové nároky – oproti některým robustnějším vývojovým prostředím, jako je PyCharm, je VS Code rychlejší a méně zatěžuje systém.

V rámci vývoje byl VS Code využíván pro psaní kódu, ladění algoritmů a testování různých funkcí vision systému. Pomocí rozšíření Python extension bylo možné pracovat s knihovnou OpenCV, analyzovat obrazová data a optimalizovat výkon aplikace.

6.2 Použité knihovny

Pro vytvoření vision systému v jazyce Python byly kromě knihovny OpenCV použity i další knihovny, které rozšiřují funkčnost a přidávají užitečné nástroje pro zpracování obrazu a řízení aplikace.

6.2.1 Numpy

Knihovna NumPy slouží jako základní nástroj pro práci s vícerozměrnými poli a maticemi v jazyce Python. Umožňuje efektivní provádění numerických operací, vektorových výpočtů a základních statistických analýz. V rámci praktické části diplomové práce je využita k ukládání a úpravě obrazových dat, provádění matematických operací nad snímky (například normalizace nebo filtrace) a k transformaci souřadnic při analýze pohybu objektů. [55]

6.2.2 Pypylon

Pro zajištění připojení kamery a zobrazení jejího výstupu v jazyce Python je nutné nainstalovat knihovnu Pypylon. Jedná se o oficiální Python wrapper ke knihovně Basler Pylon SDK, která slouží pro komunikaci s průmyslovými kamerami značky Basler.

Knihovna Pypylon umožňuje plnohodnotné ovládání kamery a poskytuje přístup k jejím funkcím, jako je nastavení parametrů snímání (například expozice, zesílení, vyvážení bílé či snímková frekvence), spouštění snímání (triggering) a přímý přístup k obrazovým datům pro jejich další zpracování v Pythonu. [56]

Výhodou této knihovny je její přímá kompatibilita s knihovnou OpenCV, která umožňuje efektivní propojení mezi získáváním obrazu a jeho následným zpracováním. [56]

6.2.3 Pillow

V rámci práce byla použita knihovna Pillow (PIL), konkrétně modul ImageGrab, který slouží pro práci s rastrovými obrázky a umožňuje zachytávání obsahu obrazovky. Tato funkcionality je využívána při získávání obrazových dat v reálném čase, zejména v případě, kdy není připojena reálná kamera. [57]

Získané snímky jsou následně převáděny do formátu polí knihovny NumPy. Tento přístup umožňuje jejich další analýzu a zpracování pomocí ostatních nástrojů použitých v rámci systému. [57]

6.2.4 Deque

V rámci práce byl využit modul collections.deque, který poskytuje datovou strukturu typu deque (double-ended queue). Tato struktura umožňuje efektivní vkládání a odebírání prvků na obou koncích seznamu.

Použití této struktury je výhodné zejména při zpracování časově závislých dat. V této práci je deque využíván pro uchovávání trajektorie pohybujícího se objektu (puku) a její následnou analýzu v reálném čase, například při predikci směru pohybu nebo při filtraci dat.

6.3 Detekce puku

Detekce puku představuje klíčový prvek vision systému, protože poskytuje vstupní data pro predikci trajektorie a následné řízení pohybů robota. Přesnost a rychlost detekce jsou zásadní pro zajištění plynulého průběhu hry a minimalizaci výskytu chyb.

Pro detekci puku byl zvolen kombinovaný přístup, který zahrnuje segmentaci barev v barevném prostoru HSV, aplikaci morfologických operací za účelem vyčištění obrazu a následnou analýzu kontur pro ověření tvaru detekovaného objektu.



Obrázek 32 Puk pro hru Air Hockey

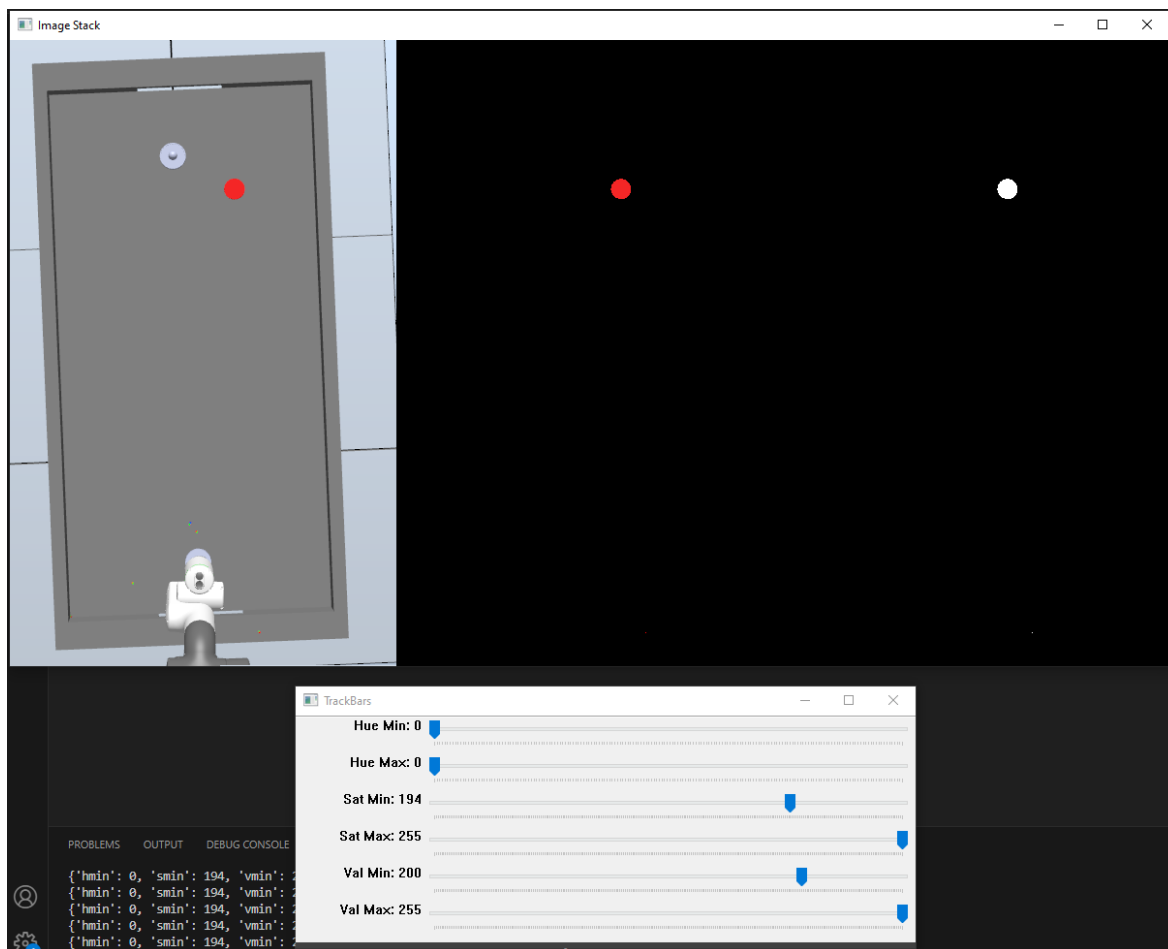
6.3.1 Použité metody detekce

Pro nalezení puku v obraze je nejprve nutné provést předzpracování snímku. Prvním krokem po načtení obrazu je jeho převedení do barevného prostoru HSV, který umožňuje efektivnější separaci objektů na základě barevných vlastností.

Tento přístup byl zvolen vzhledem k barevnému kontrastu mezi červeným pukem a bílou hrací plochou stolu. Díky této barevné odlišnosti lze puk v obraze snadno izolovat pomocí prahování a vytvoření binární masky.

Po převedení obrazu do barevného prostoru HSV je prováděna barevná segmentace pomocí funkce `cv2.inRange()`, která umožňuje oddělit puk od zbytku obrazu na základě zvoleného barevného rozsahu. Výstupem této operace je binární maska, ve které jsou pixely odpovídající definovanému barevnému rozsahu označeny hodnotou 1, zatímco ostatní pixely mají hodnotu 0.

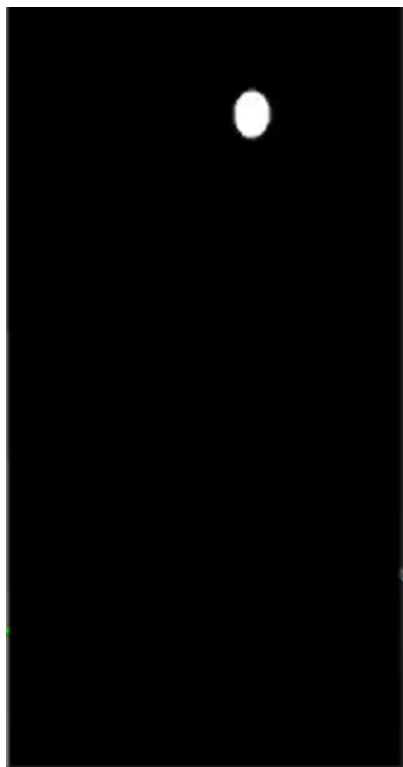
Pro určení vhodného rozsahu hodnot v prostoru HSV byla využita knihovna CVZone, která umožňuje snadné vytvoření jednoduchého skriptu pro interaktivní ladění parametrů masky pomocí posuvníků. Tímto způsobem bylo možné efektivně určit optimální rozsah pro detekci červené barvy puku.



Obrázek 33 Script pro výběr HSV masky

Na výstupu jsou zobrazována tři samostatná okna. První okno zobrazuje původní načtený obraz. Ve druhém okně je zobrazen obraz po aplikaci barevné filtrace v HSV prostoru. Třetí okno pak prezentuje výsledný binární obraz, který vznikl po aplikaci prahování na základě zvoleného HSV rozsahu.

Po provedení segmentace obrazu může být výsledná binární maska ovlivněna šumem nebo nesprávně detekovanými objekty. Z tohoto důvodu je na masku aplikována sada morfologických operací, které slouží k odstranění nežádoucích artefaktů a ke zlepšení celkové kvality detekovaných oblastí. V rámci implementace byly využity různé operace, jako je Gaussovské rozostření, eroze, dilatace, otevření a uzavření. Aplikací těchto metod došlo ke zvýšení přesnosti výsledné masky, a to i v případech se zhoršenými světelnými podmínkami.

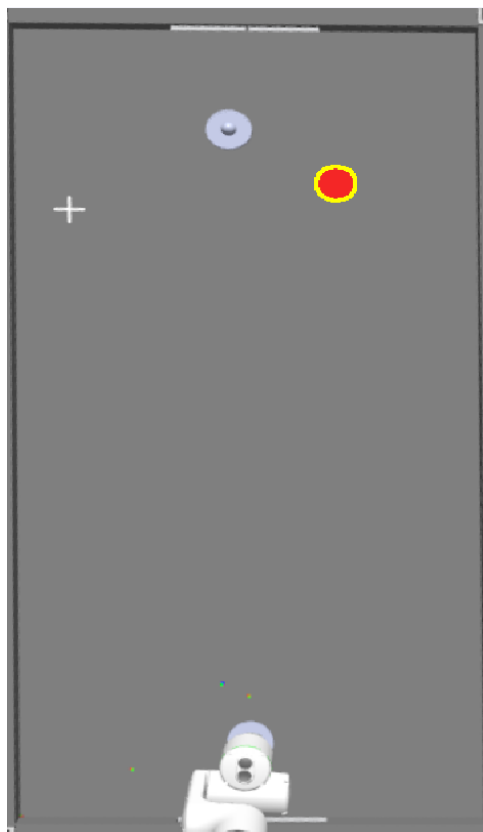


Obrázek 34 Binární obraz s pukem

Za účelem eliminace falešných detekcí je na výsledné binární masce provedena analýza kontur. Tato metoda umožňuje identifikovat relevantní objekty na základě jejich geometrických vlastností, jako je velikost nebo tvar. Pomocí konturové analýzy lze účinně odfiltrout objekty, které neodpovídají očekávaným parametrům puku.

Pro detekci kontur v binární masce byla využita funkce *cv2.findContours()*, která umožňuje nalezení uzavřených oblastí v obraze. Vzhledem k tomu, že hokejky mají podobnou barvu jako puk, mohlo docházet k jejich chybnému zařazení mezi detekované objekty.

Za účelem omezení těchto falešných detekcí byla implementována podmínka, která filtruje kontury na základě jejich velikosti. Následně byla provedena kontrola tvaru pomocí výpočtu kruhovitosti, která slouží k ověření, zda detekovaný objekt odpovídá přibližně kruhovému tvaru charakteristickému pro puk.



Obrázek 35 Nalezený objekt pomocí funkce `cv2.findContours()`

Na obrázku (Obrázek 35) je detekovaný puk označen žlutou barvou. Konturová filtrace identifikovala v obraze pouze jeden objekt odpovídající požadovaným podmínkám. To potvrzuje správné nastavení parametrů detekčního procesu.

Výsledkem je robustní a přesná detekce puku, která poskytuje spolehlivá vstupní data pro predikci trajektorie a řízení pohybů robota.

6.4 Predikce trajektorie

Po úspěšné detekci puku následuje predikce jeho trajektorie. Správná predikce umožňuje robotu se připravit na vhodné místo, kde může puk zachytit a zabránit soupeři ve skórování.

Pro predikci trajektorie bylo použito jednoduché lineární modelování pohybu, které zahrnuje výpočet rychlosti, odhad směru pohybu a simulaci odrazů od bočních hran herního stolu.

Pro správné predikování pohybu je nejprve nutné spočítat rychlost puku na základě změny jeho polohy v několika po sobě jdoucích snímcích. K tomu slouží historie pozic puku, která se ukládá do FIFO fronty. Pokud je ve frontě dostatek dat, lze vypočítat průměrnou změnu polohy ve směru os X a Y.

```
if len(history) >= 4: # Počítáme predikci až po nasbírání dostatečného počtu snímků
    dx_total = sum(history[i + 1][0] - history[i][0] for i in range(len(history) - 1))
    dy_total = sum(history[i + 1][1] - history[i][1] for i in range(len(history) - 1))
    dt = len(history) - 1 # Počet snímků mezi výpočtem

    vx = dx_total / dt
    vy = dy_total / dt
```

Obrázek 36 Výpočet rychlosti puku

Jakmile je rychlost puku určena, může být jeho budoucí poloha odhadnuta simulací pohybu po přímce. Predikce je prováděna na několik kroků dopředu. V rámci simulace je nutné zohlednit možnost odrazů puku od bočních hran herního stolu, které musí být v predikčním modelu správně zpracovány.

Cílem predikce je nalezení bodu, ve kterém puk protne přímkou představující hranu hrací plochy robota (označenou jako $y = \text{target_y}$). Tento bod je označován jako finální predikovaný bod a určuje místo, do kterého by měl být robot přesunut, aby mohl účinně zareagovat a puk odrazit.

```
for _ in range(40): # Předpověď na 40 kroků
    pred_x += vx
    pred_y += vy

    if pred_x - puck_radius <= 0 or pred_x + puck_radius >= roi_w:
        vx = -vx # Odrážení na ose X

    if pred_y >= target_y:
        prev_x, prev_y = predicted_trajectory[-1]
        if pred_y - prev_y != 0:
            t = (target_y - prev_y) / (pred_y - prev_y)
            final_predicted_x = int(prev_x + t * (pred_x - prev_x))
            final_predicted_y = target_y
            final_predicted_point = (final_predicted_x, final_predicted_y)
        else:
            final_predicted_point = (int(pred_x), target_y)
        break

    predicted_trajectory.append((int(pred_x), int(pred_y)))

prediction_active = True
```

Obrázek 37 Predikování finálního bodu

Velkou výhodou navrženého systému je schopnost pokračovat v predikci trajektorie i v případě krátkodobé ztráty vizuální detekce puku. Pokud není puk na některých snímcích detekován – například v důsledku rozmazání, rychlého pohybu nebo zakrytí – již vypočítaná trajektorie není zahozena. Místo toho je dočasně využita poslední známá rychlost a směr pohybu puku k pokračování v simulaci

Tím je zajištěno, že robot pokračuje v pohybu směrem k očekávanému místu kontaktu s pukem, i když dojde k výpadku vizuální informace. Díky tomu nedochází k prodlevám v reakci a je zachována plynulost hry.

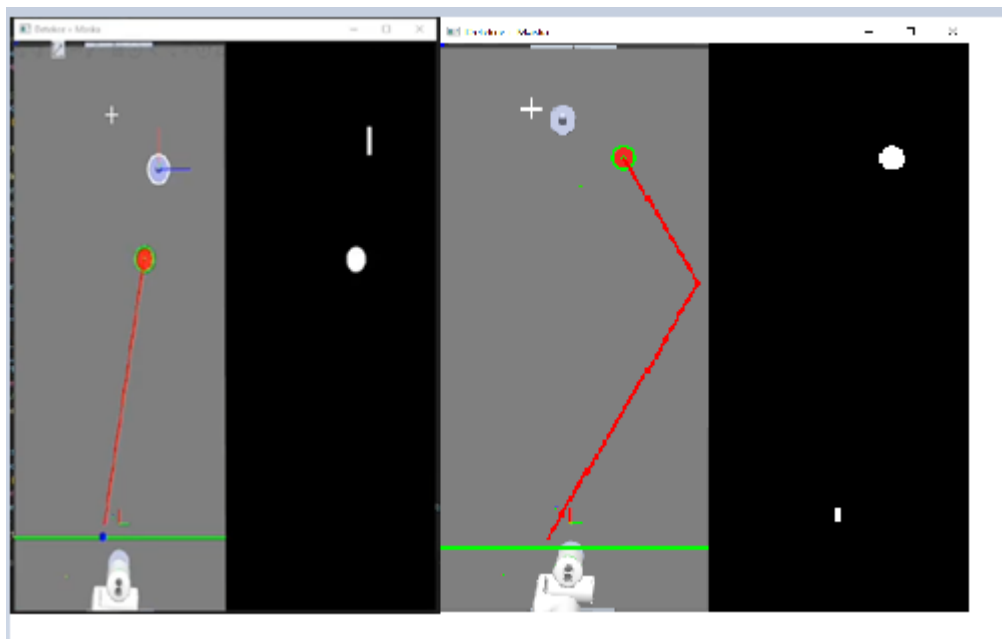
```
else:
    missed_frames += 1
    if missed_frames >= max_missed_frames:
        print("Puk nebyl detekován několik snímků. Trajektorie resetována.")
        prediction_active = False
        predicted_trajectory = []
        final_predicted_point = None
```

Obrázek 38 Pokračování predikce trajektorie při výpadku detekce

Tato forma předvídání je omezena parametrem, který určuje maximální počet snímků, během nichž může být detekce puku ztracena. Pokud není puk po překročení tohoto limitu znovu detekován, probíhající predikce je resetována.

V případě, že je puk opět detekován, je jeho aktuální poloha porovnána s dříve predikovanou trajektorií. Pokud je zjištěna výrazná odchylka, dojde k přepočítání nové trajektorie na základě aktuálních dat.

Díky tomuto postupu je navržený systém schopen spolehlivě předvídat místo dopadu puku a zajišťovat plynulou a včasnou reakci robota i v případě dočasného výpadku detekce.



Obrázek 39 Predikce trajektorie puku

6.5 Řízení robota

Jako první způsob odesílání dat robotu byl zvolen přenos pomocí protokolu TCP/IP. Vision systém na základě výpočtu předpovídá, kde by se měl puk nacházet, a následně jsou souřadnice predikovaného bodu odesílány robotu k provedení požadované reakce.

Aby mohly být souřadnice správně interpretovány robotem, je nutné je převést ze souřadnic v pixelovém prostoru (obrazu) do souřadnicového systému robota. Tento převod je umožněn díky znalosti reálných rozměrů hrací plochy a vymezeného regionu zájmu (ROI), který na obrazu odpovídá herní ploše stolu.

Převod se provádí na základě poměru mezi rozměry ROI v pixelech a skutečnými rozměry stolu. Reálná pozice ve směru osy X je získána vynásobením souřadnice v pixelech reálnou šířkou hrací plochy a následným podělením šířkou ROI. Souřadnice ve směru osy Y je počítána obdobně, avšak s ohledem na rozdílnou orientaci souřadnicových systémů. Vision systém má počátek v levém horním rohu, zatímco robot v levém dolním. Z tohoto důvodu je po přepočtu odečtena reálná délka stolu, aby byla souřadnice transformována do systému robota.

6.5.1 Knihovna Socket

Takto převedené souřadnice mohou být odeslány robotu prostřednictvím síťové komunikace. K odesílání dat byla využita knihovna socket, která je standardní součástí jazyka Python a slouží k implementaci síťových klient-server aplikací na různých úrovních síťové architektury. Tato knihovna poskytuje rozhraní pro vytváření a správu síťových socketů, které tvoří základní komunikační prvek pro přenos dat. [58]

Knihovna umožňuje vytváření jak TCP (spolehlivé, spojované) spojení, tak UDP (rychlé, ale bez záruky doručení) spojení. V rámci této práce byl zvolen protokol TCP/IP jako první způsob komunikace mezi vision systémem a robotem.

Po detekci puku a výpočtu jeho predikované trajektorie byly souřadnice predikovaného bodu odeslány robotu ve formě datového typu string, obsahujícího hodnoty souřadnic predikovaného body X a Y a dále pak souřadnice X a Y puku.

```
def send_to_rob():
    real_x = int(predicted_endpoint[0] * real_width / roi[2])
    real_y = int(predicted_endpoint[1] * real_lengh / roi[3])
    real_y_otoc = real_lengh - real_y
    puk_poziceY_otoc = int(real_lengh - (Cy * real_lengh / roi[3]))
    puk_poziceX = int(Cx * real_width / roi[2])
    predicted_str = f"{real_x:03d}{real_y_otoc:03d}{puk_poziceY_otoc:03d}{puk_poziceX:03d}"
    if predicted_endpoint[1] == target_y:
        #if Cy > Py:
            client_socket.sendall(predicted_str.encode())
            print(f"torob {predicted_str}")
    else:
        if Cy >= Py:
            fejkovej_stred = int(roi[2] / 2 * real_width / roi[2])
            predicted_str = f"{fejkovej_stred:03d}{000:03d}{puk_poziceY_otoc:03d}{puk_poziceX:03d}"
            client_socket.sendall(predicted_str.encode())
            print(f"torob prostedek{predicted_str}")
```

Obrázek 40 Funkce pro odesílání dat robotu

Druhým způsobem předávání souřadnic robotu bylo využití protokolu UDP. Hlavní výhodou tohoto přístupu je, že data jsou odesílána nepřetržitě bez ověřování jejich doručení. Odesílání tedy probíhá neblokovaně, bez zpětné vazby o úspěšném přijetí ze strany robota.

V případě, že robot některý z datových paketů nezpracuje nebo zmešká, jednoduše počká na další přijatý paket.

6.5.2 Knihovna pro EGM pohyb robota

Externě řízený pohyb (Externally Guided Motion, EGM) je funkcionalita softwaru RobotWare od společnosti ABB, která umožňuje externím systémům dynamicky řídit pohyb průmyslového robota v reálném čase. Tato technologie využívá síťovou komunikaci k příjmu korekčních signálů z externích senzorů nebo řídicích systémů.

EGM funguje na principu zpětnovazebního řízení, kdy je pohyb robota kontinuálně přizpůsobován na základě přijatých dat. Technologie podporuje různé režimy řízení, včetně polohového a rychlostního řízení.

Díky EGM je možné integrovat robota do pokročilých automatizačních systémů, jako jsou adaptivní výrobní linky nebo kolaborativní aplikace, kde je požadováno, aby robot reagoval na změny v prostředí v reálném čase.

Pro implementaci řízení robota pomocí EGM do vision systému byla využita knihovna ABBRobotEGM. Knihovna ABB-EGM-Python slouží jako rozhraní mezi robotem ABB a řídicím softwarem běžícím na externím počítači. Tato knihovna umožňuje odesílání řídicích příkazů v reálném čase a získávání zpětné vazby o aktuální poloze robota. Využívá UDP komunikaci a poskytuje API, které umožňuje snadnou integraci do Python aplikací. Mezi klíčové funkce této knihovny patří nastavování cílových pozic, řízení rychlosti a získávání

informací o poloze kloubů robota. Díky nízké latenci je umožněna implementace pokročilých řídicích algoritmů, například pro sledování trajektorie objektů v reálném čase. [<https://github.com/FLo-ABB/ABB-EGM-Python>].

Pro implementaci EGM do řízení robota bylo nutné upravit funkci pro odesílání souřadnic robota. Funkce již neodesílá souřadnice, které by robot následně zpracoval a plánoval svou trajektorii. Tato implementace zahrnuje přímé odesílání souřadnic cílového bodu, přičemž vypočítaná pozice puku je transformována na reálné souřadnice pracovního prostoru robota. Tyto souřadnice jsou následně odeslány prostřednictvím UDP komunikace do systému EGM robota, který okamžitě reaguje a upravuje svou polohu podle přijatých dat.

Funkce *send_to_robot()* přijímá vypočítanou predikovanou pozici puku a na jejím základě rozhoduje o pohybu robota. V případě, že není k dispozici žádný platný predikovaný bod, robot se vrací do výchozí neutrální pozice. Pokud je predikovaný bod k dispozici a odpovídá cílové oblasti pohybu robota, funkce přepočítá souřadnice na reálné hodnoty v milimetrech a odešle je přímo robotu. Tento přístup minimalizuje zpoždění mezi detekcí a reakcí robota.

Dále byla implementována funkce *rob_attack*, která umožňuje vystřelení puku směrem do soupeřovy branky. Funkce pracuje se souřadnicemi puku. Jakmile se puk přiblíží k robotovi, je na základě aktuální polohy robota a puku vypočítán úhel a souřadnice potřebné pro odraz. Po určení správného směru a úhlu odrazu je následně proveden pohyb robota, jehož výsledkem je odražení puku směrem k soupeřově brance.

Pro zvýšení dynamiky hry byla implementována funkce *rob_clear_puck*, jejímž účelem je automatické odstranění nehybného puku z poloviny hrací plochy robota. Pokud je detekováno, že se puk po definovaný časový úsek nepohybuje a nachází se na robotově polovině hrací plochy, je tímto mechanismem puk posunut směrem k soupeřově straně.

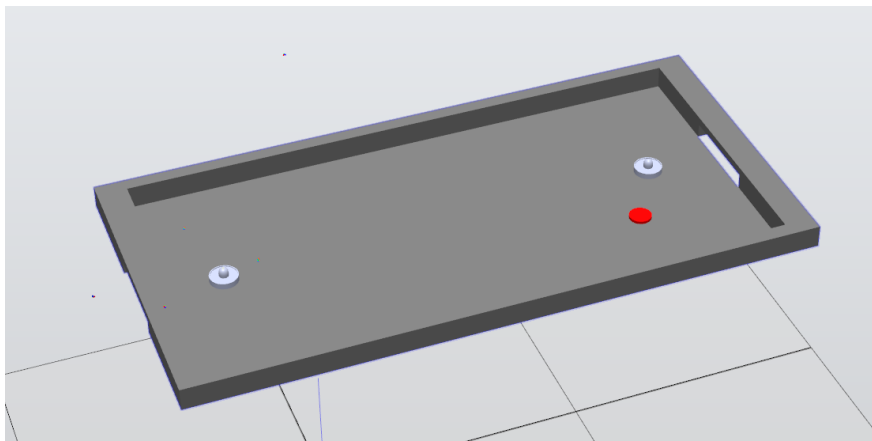
Díky těmto úpravám nyní robot reaguje na pohyb puku v reálném čase s minimální latencí a bude schopen provádět rychlou a aktivní hru simulující reálného hráče.

7 SIMULAČNÍ PROSTŘEDÍ

Pro otestování celého systému v reálném prostředí bylo nutné vytvořit několik součástí, které byly nezbytné pro provedení testů v reálných podmínkách. Pro účely simulace byl využit software ABB RobotStudio, který umožnil testování systému ve virtuálním prostředí před jeho nasazením do skutečné aplikace.

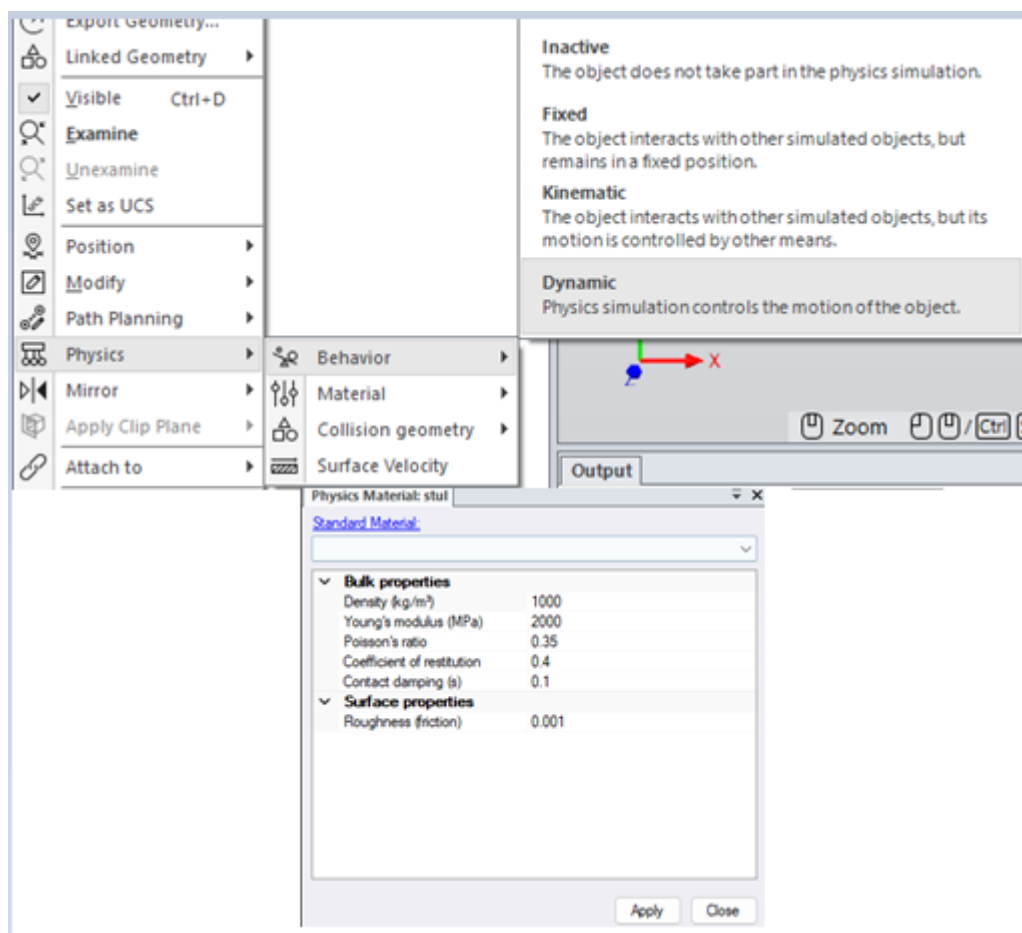
7.1 Vytvoření modelů a příprava simulačního prostředí

Z vybraného stolu Buffalo Dominator 7FT byl vytvořen jeho 3D model. Tento 3D model následně posloužil pro vytvoření prvotní simulace, ve které bylo možné otestovat navržený vision systém, zjistit základní chování puku a ověřit, zda bude robot dostatečně rychlý pro reakci na změny pohybu a zastavení puku.



Obrázek 41 Vytvořený model stolu pro Air Hockey

Pro simulaci klouzání puku po stole bylo nutné upravit vlastnosti objektu v RobotStudiu. Vzhledem k tomu, že RobotStudio neumožňuje simulaci proudění vzduchu skrze otvory ve stole, jak je tomu u reálného stolu, byly nastaveny fyzikální vlastnosti povrchu tak, aby odpovídaly kluzkosti ledu. Tímto způsobem bylo dosaženo realistického chování puku, který se po stole pohybuje podobně jako v reálných podmínkách.

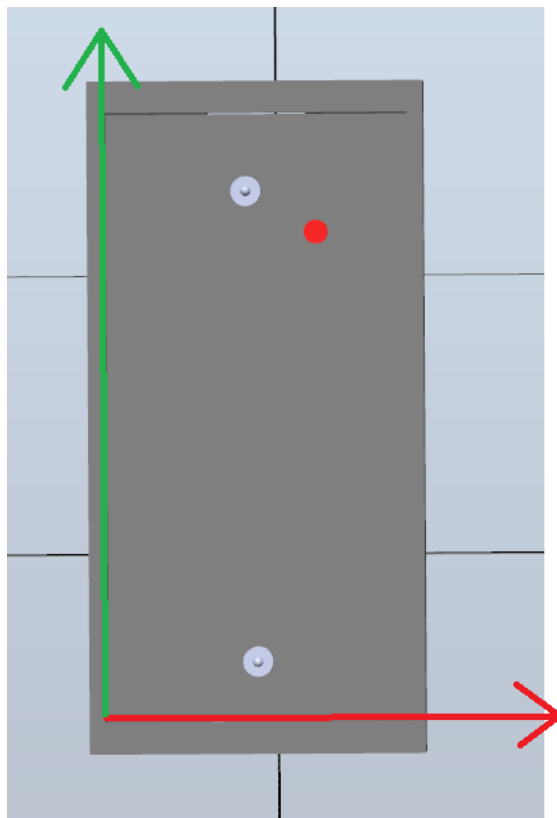


Obrázek 42 Nastavení fyzikálních vlastností stolu a objektů

Model puku a modely obou hokejek byly nastaveny jako aktivní objekty. Po spuštění simulace s nimi bylo možné pohybovat a simulovat reálnou hru Air Hockey. Tato interakce s objekty umožnila testování a ladění chování systému v reálných herních podmínkách.

Do simulačního prostředí byl vložen robot, který je identický s vybraným robotem v předchozí kapitole. Pro základní otestování byl na robota umístěn ABB smart gripper, který umožnil robotu uchopit hokejku a následně s ní pohybovat.

Dále byl vytvořen workobject, který slouží jako základní souřadnicový systém robota. Tento workobject má počátek souřadnicového systému umístěn v levém spodním rohu stolu.



Obrázek 43 Workobject jako základní souřadnicový systém

Červená šipka označuje X souřadnici, zatímco zelená šipka udává směr Y souřadnice. Takto nastavený souřadnicový systém by měl korespondovat se souřadnicovým systémem používaným vision systémem.

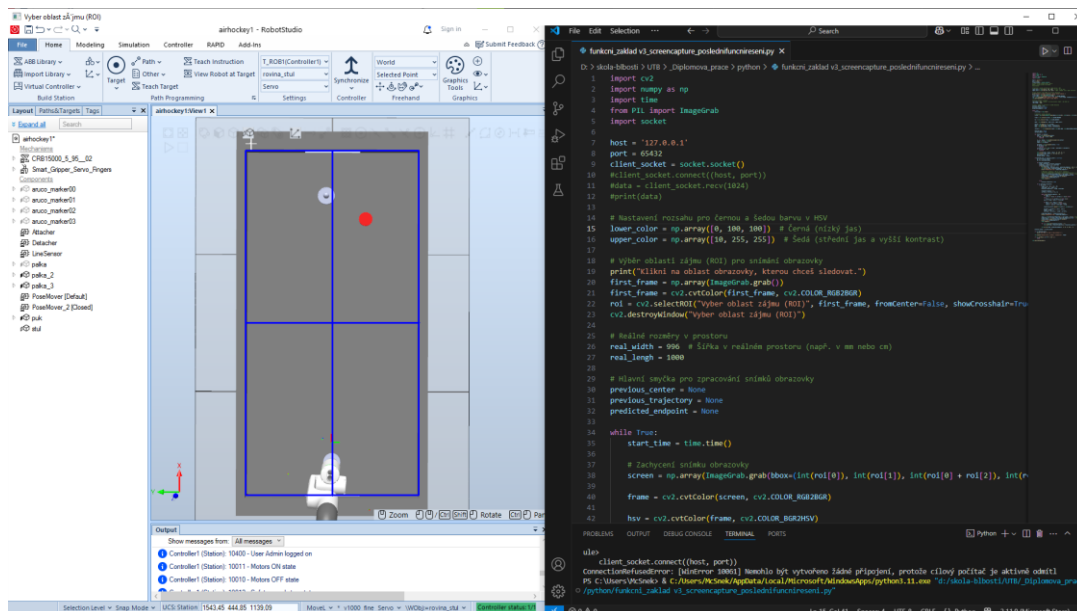
7.2 Program pro robota

Pro zpracování souřadnic predikovaného bodu, které poskytuje vision systém, bylo nutné naprogramovat robota. Robot je nastaven tak, aby poslouchal na otevřeném portu, kde běží TCP komunikace. Po přijetí řetězce (stringu) je tento řetězec převeden na numerické proměnné, které jsou následně předány jako bod ve vytvořeném souřadnicovém systému. Robot následně přejede do tohoto bodu s cílem zastavit puk.

Robot může ještě provést funkci *hit_puck*, která mu umožňuje vystřelit puk do soupeřovy branky. Tato funkce využívá souřadnice puku získané při každém přijetí dat. Jakmile se puk přiblíží k robotovi, funkce vypočítá úhel a souřadnice odrazení puku na základě aktuální pozice robota a puku. Po výpočtu správného směru a úhlu odrazu robot vykoná pohyb, který umožní odrazení puku do soupeřovy branky, čímž se provede střelba na branku.

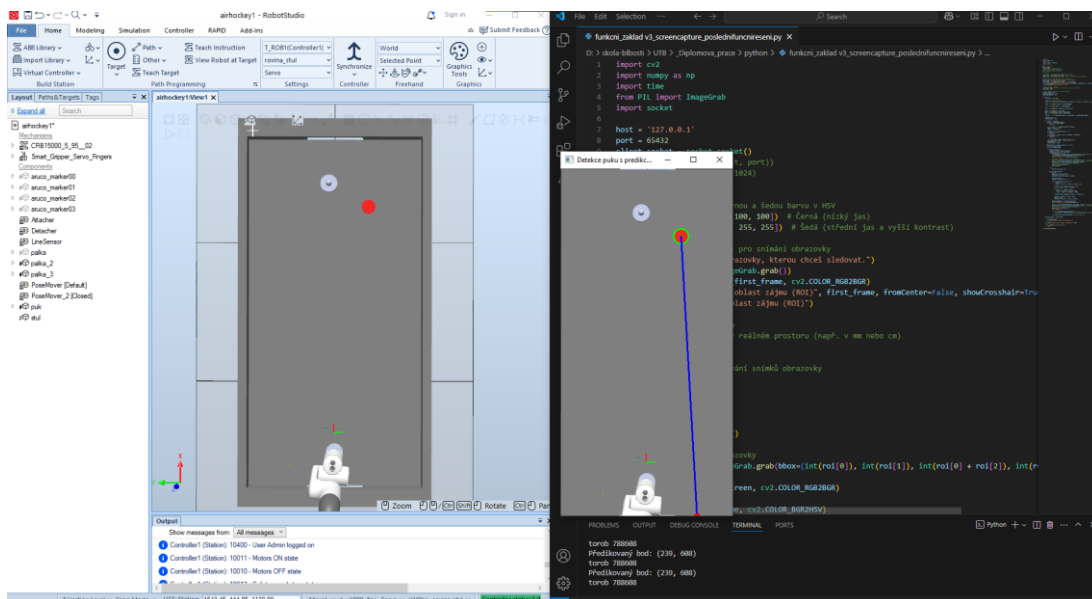
7.3 Otestování vision systému v simulačním prostředí

Nyní byl celý systém připraven k otestování. Pro testování bylo nutné, aby bylo otevřeno RobotStudio na jedné straně obrazovky a Visual Studio na straně druhé. Po spuštění programu je potřeba vybrat myší obdélníkové ROI, které označuje hrací plochy stolu.



Obrázek 44 Spuštění programu a výběr ROI

Jakmile je ROI vybráno, postačuje jeho potvrzení klávesou Enter. Po jejím stisknutí je program automaticky spuštěn.



Obrázek 45 Spuštění program

V prostředí RobotStudia je k dispozici možnost uchopení hokejky a odražení puku, aby mohla být hra simulována. Postačuje aktivace možnosti Drag v prostředí RobotStudia. Následně může být hra simulována.

Při pohybu puku je trajektorie automaticky počítána a robot je přesouván na příslušné souřadnice.

V případě komunikace prostřednictvím TCP/IP a UDP musela být rychlost puku omezena, jelikož při vyšších rychlostech nebylo možné zajistit včasný přesun robota na predikovaný bod.

Při řízení robota prostřednictvím EGM (Externally Guided Motion) byla umožněna vyšší rychlost puku, neboť byl robot schopen realizovat rychlejší a přesnější pohyby.

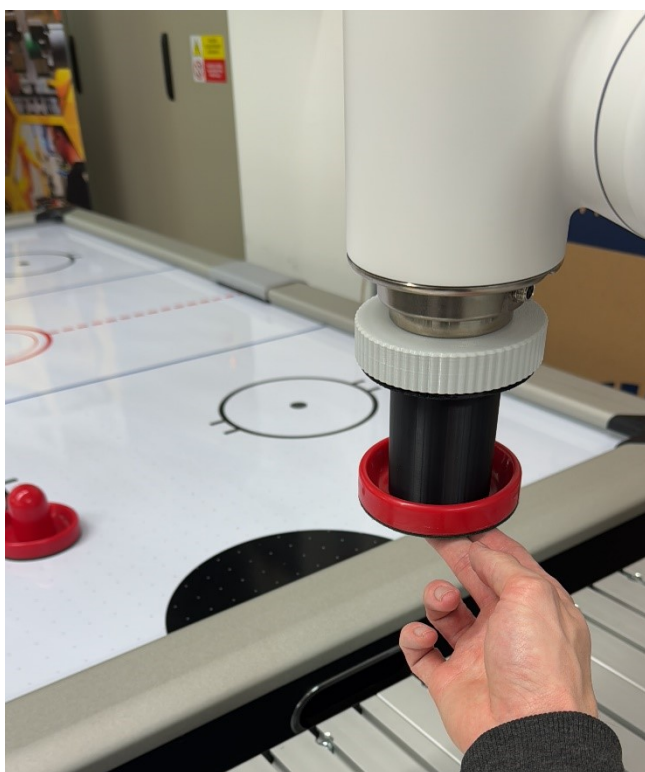
Pro další testování bude tedy použito řízení robota přes EGM, protože řízení přes TCP by bylo znatelně pomalejší. Robot by nebyl schopen se rychle přesunout na predikované místo a to by výrazně snížilo dynamiku hry.

8 REÁLNÉ PROSTŘEDÍ

Pro otestování celého systému v reálném prostředí bylo nutné vytvořit několik komponent, které byly nezbytné pro správné fungování a testování systému.

8.1 Návrh koncového efektoru

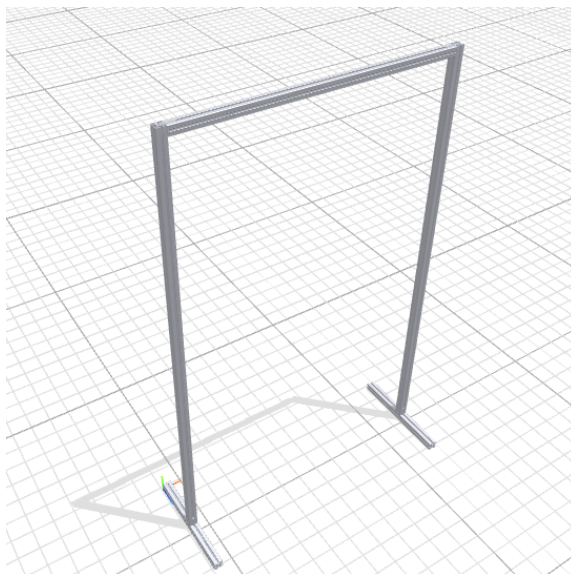
Aby byl robot schopen uchopit hokejku, byl navržen jednoduchý koncový efektor, do kterého lze hokejku uchytit. Výhodou tohoto koncového efektoru je, že robot může použít originální a povolenou hokejku, aniž by bylo nutné ji jakkoli upravovat pro uchopení. Pro vytvoření modelu byl použit program SolidWorks. Tento model byl poté vytisknut na 3D tiskárně.



Obrázek 46 Koncový efektor pro uchopení hokejky

8.2 Konstrukce pro kameru

Jelikož byla na začátku této práce určena orientační vzdálenost pro umístění kamery, bylo nutné navrhnout a postavit konstrukci pro její uchycení. Pro sestavení konstrukce byly zvoleny hliníkové profily, které se běžně používají v průmyslových aplikacích pro stavbu stabilních a flexibilních konstrukcí.

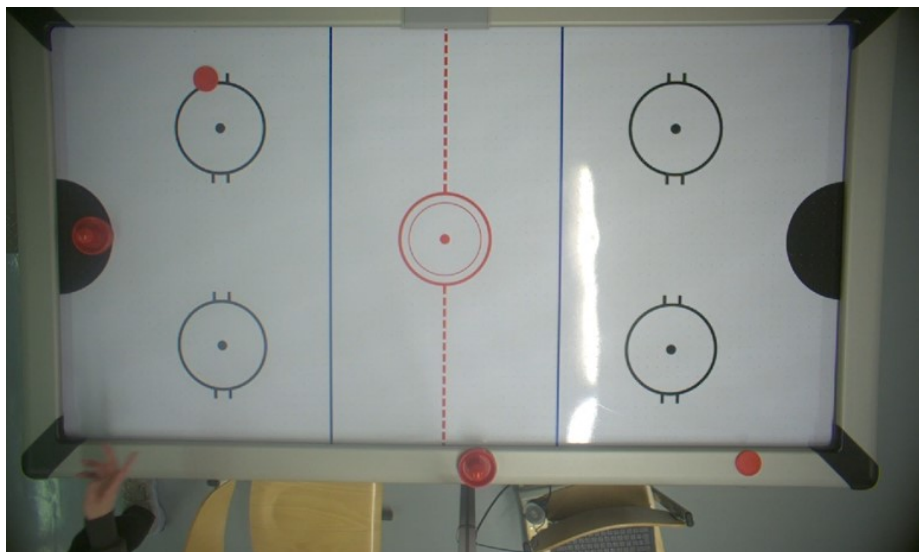


Obrázek 47 Návrh konstrukce pro kameru



Obrázek 48 Postavená konstrukce pro kameru

Takto navrhnutá a sestavená konstrukce nezasahuje do stolu, takže není nutné provádět žádné úpravy stolu. Konstrukce zabraňuje přenosu vibrací ze stolu do rámu a zajišťuje minimální pohyb kamery. Kamera je umístěna v dostatečné výšce a umožňuje sledování celé hrací plochy stolu.



Obrázek 49 Obraz pořízený kamerou na nosné konstrukci

8.3 Program pro robota

Robot je v současné implementaci řízen prostřednictvím rozhraní EGM (Externally Guided Motion). Z tohoto důvodu musel být upraven řídicí kód přímo na robota. Přenos souřadnic již není realizován prostřednictvím TCP komunikace a žádné další převody souřadnic nejsou prováděny. Veškerý pohyb robota je nyní řízen výhradně pomocí vision systému.

```
PROC egm()
  EGMReset egmID1;
  EGMGetId egmID1;
  egmSt1:=EGMGetState(egmID1);
  TPWrite "EGM state: "\Num:=egmSt1;
  IF egmSt1<EGM_STATE_CONNECTED THEN
    EGMSetupUC ROB_1,egmID1,"default","UCdevice"\pose;
  ENDIF
  EGMActPose egmID1\Tool:=Servo\Wobj:=rovina_stul,pose0,EGM_FRAME_WOBJ,pose0,EGM_FRAME_WOBJ\X:=egm_minmax_lin\Y:=egm_minmax_lin\LpFilter:=5\MaxSpeedDeviation:=100;
  WHILE TRUE DO
    EGMRunPose egmID1,EGM_STOP_HOLD\X\Y\CondTime:=60\RampInTime:=0.1;
  ENDWHILE
  ! EGMRunPose egmID1,EGM_STOP_HOLD\X\Y\CondTime:=60\RampInTime:=0.1;

  IF egmSt1=EGM_STATE_CONNECTED THEN
    TPWrite "Reset EGM instance egmID1";
    EGMReset egmID1;
  ENDIF

  EGMStop egmID1,EGM_STOP_HOLD;

  ERROR
  IF ERRNO = ERR_UDPUC_COMM THEN
    TPWrite "Communication timedout";
    TRYNEXT;
  ENDIF
ENDPROC
```

Obrázek 50 Kód v robotu pro řízení pohybu přes EGM

8.4 Otestování vision systému a robota v reálném prostředí

Pro otestování v reálném prostředí bylo nutné přepnout vstup obrazu v programu. Obraz byl nyní získáván z reálné kamery místo nahrávání obrazovky. Po spuštění programu bylo nutné vybrat pomocí myši obdélníkový ROI, který označuje hrací plochu stolu. Po spuštění programu bylo také nutné spustit robota.

Chování robota v reálném prostředí bylo stejné jako chování v simulaci. Bylo však nutné upravit maximální zrychlení robota, protože při vyšších rychlostech se robot zastavoval s chybou. Robot nedokázal provést tak velké zrychlení při výrazných změnách pohybu.

8.4.1 Nastavení parametrů kamery Basler

Z důvodu specifického osvětlení v laboratoři, kde je umístěn stůl s robotem, bylo nezbytné upravit parametry kamery tak, aby byl získaný obraz vhodný pro detekci puku. Rozlišení kamery bylo nastaveno na 1920×1080 pixelů a snímkovací frekvence na 38 snímků za sekundu. Pomocí úpravy rychlosti závěrky byla zvolena taková frekvence snímání, která umožnila eliminaci blikání zářivek přítomných v laboratoři.

8.4.2 Detekce ArUco značek

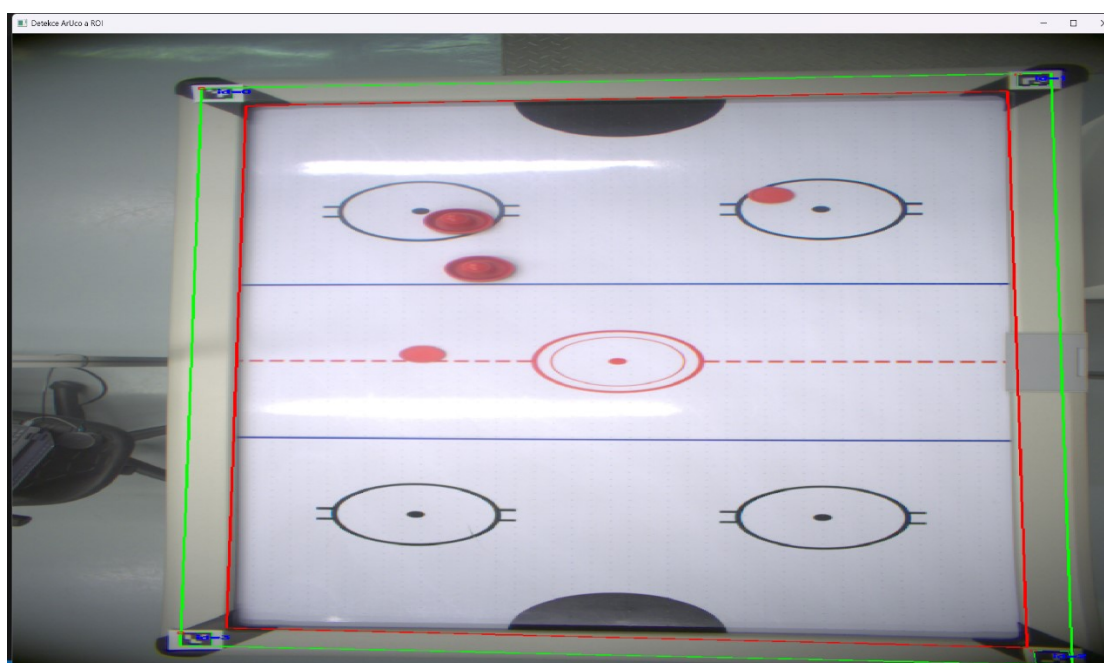
ArUco marker je speciální typ vizuálního markeru používaný pro detekci a lokalizaci v reálném prostředí. Je to čtvercový fiducialní marker, který obsahuje unikátní kódování ve formě binárního vzoru. Tento vzor je obvykle složen z černobílých polí, které jsou uspořádány tak, aby vytvářely jednoznačnou identifikaci pro každý marker.

ArUco marker byl použit pro korekci obrazu. Kamera není umístěna přesně rovně s hrací plochou, a proto při výběru obdélníkového ROI není možné přesně označit hrací oblast. Do rohů stolu byly umístěny čtyři ArUco značky, které označují hrací plochu. Každá značka je vytisknuta na papír ve velikosti 50 mm a má předem definovanou pozici. Značka s ID 0 je umístěna v levém horním rohu, značka s ID 1 v pravém horním rohu, značka s ID 2 v pravém dolním rohu a značka s ID 3 v levém dolním rohu.

Při spuštění programu se automaticky načtou ArUco značky a na jejich základě se vytvoří obdélníkové ROI. Tento přístup zjednodušuje výběr ROI a celý proces více automatizuje. Dále se používá perspektivní transformace, která obraz vyrovná a tím zlepší a zpřesní detekci.

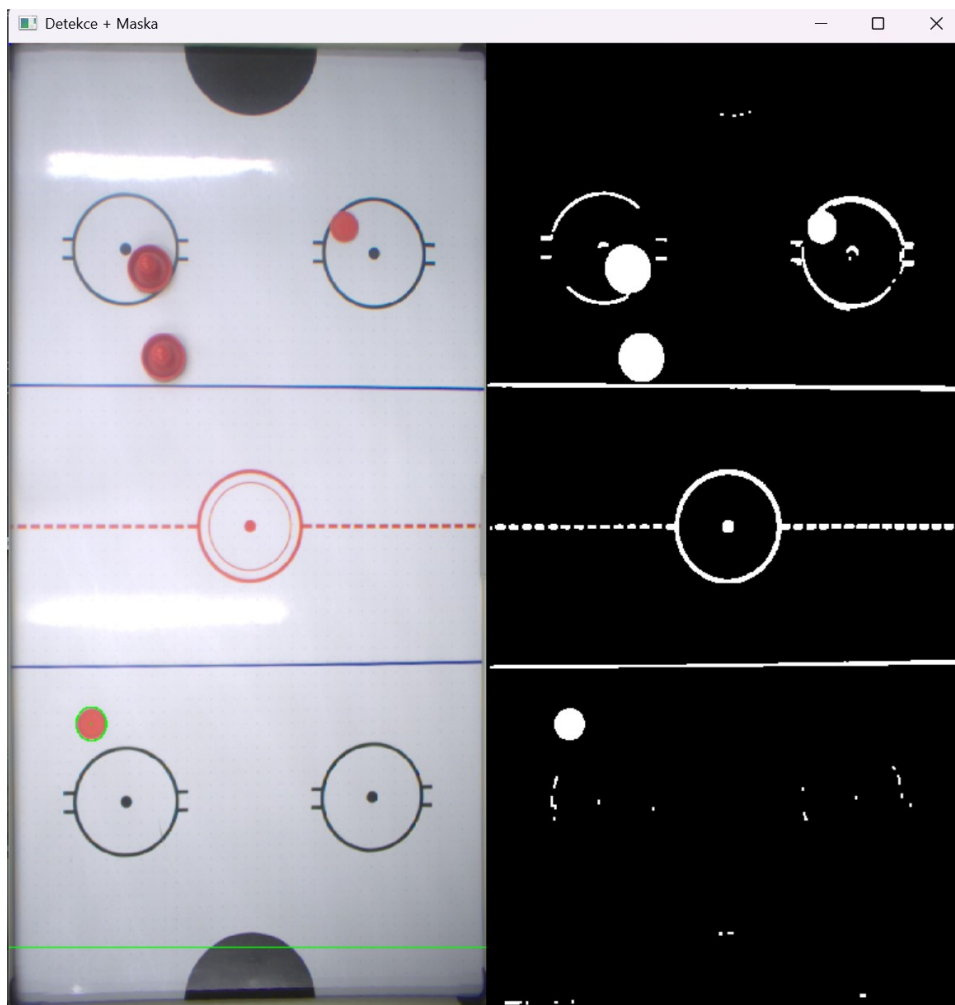


Obrázek 51 Stůl s ArUco značkami



Obrázek 52 Automaticky vybraný obdélník značící hrací plochu

Na Obrázek 52 jsou zobrazeny detekované ArUco značky. Z jejich jednotlivých rohů je následně vypočten zelený obdélník. Červený obdélník je poté posunut dovnitř pomocí offsetů, aby byla simulována přesná hrací plocha stolu.

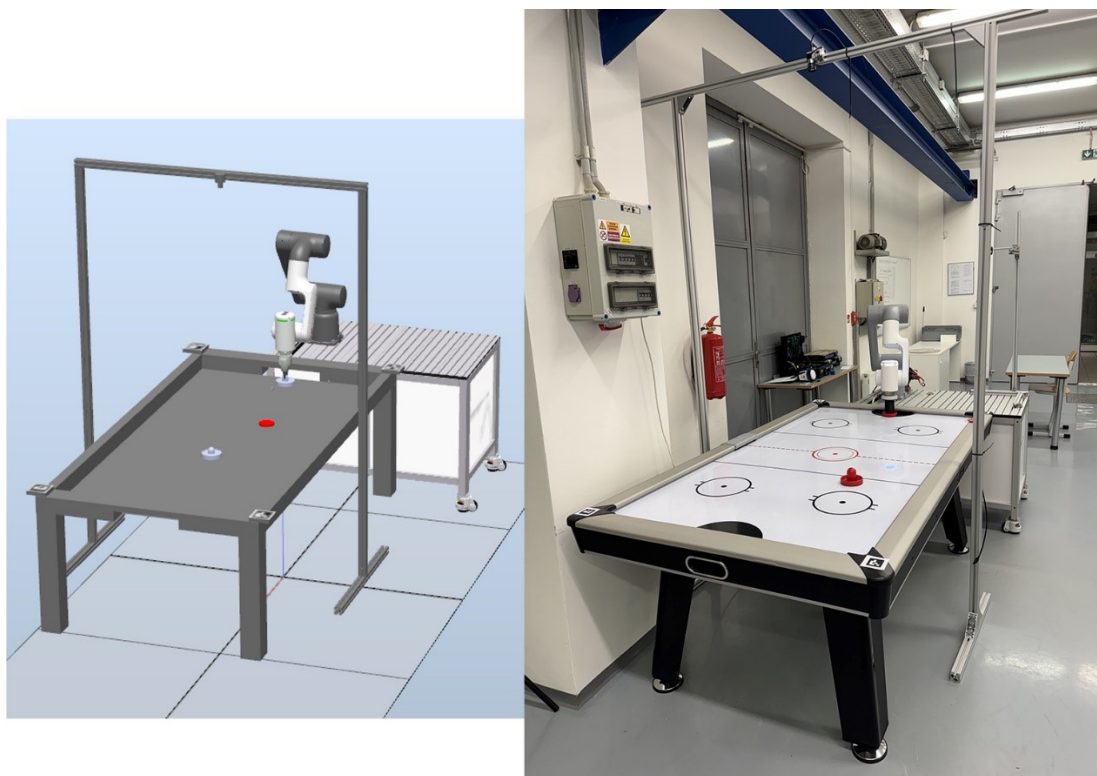


Obrázek 53 Výstup programu s obrazem reálné kamery

9 VÝSLEDKY

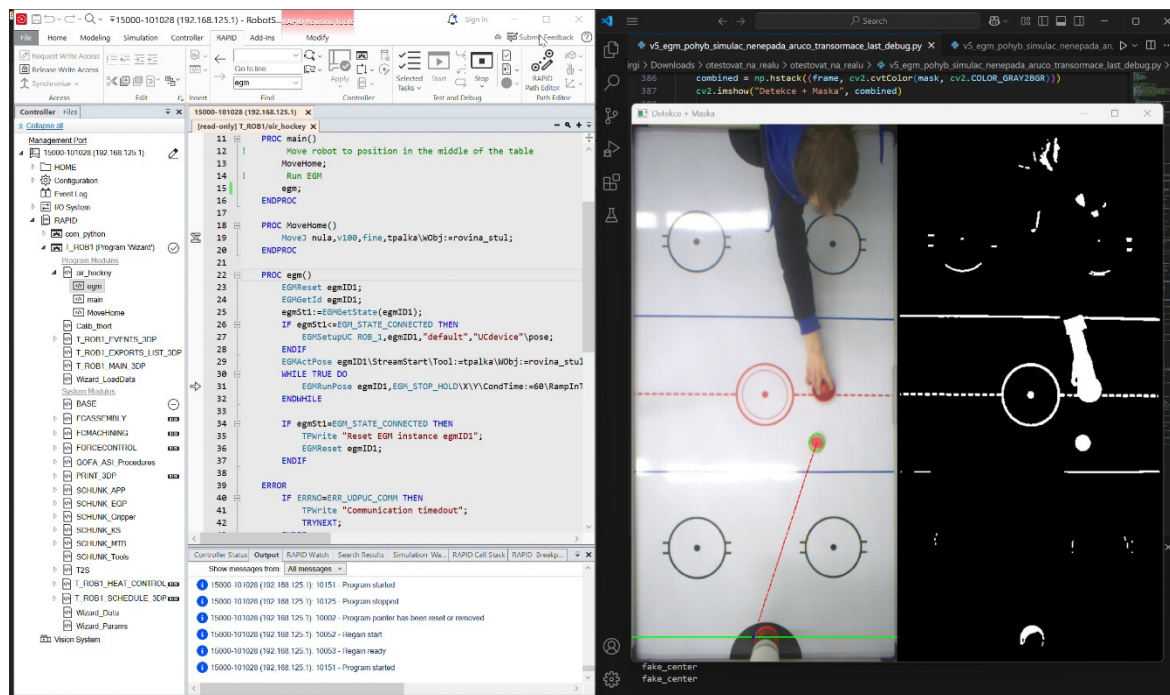
Kompletní vision systém je možné použít jak v simulačním prostředí RobotStudio, tak v reálném prostředí. Systém umožňuje automatickou detekci ROI, nebo je možné ROI zvolit ručně v případě, že ArUco značky nebudou nalezeny. Navržený systém je schopen přesně a rychle detekovat puk a predikovat jeho trajektorii.

Vision systém řídí robota pomocí EGM, které zajišťuje větší rychlost robota pro plynulou a dynamickou hru.



Obrázek 54 Kolaborativní robot jako oponent ve hře Air Hockey

Na Obrázek 54 lze vidět na levé straně vytvořený systém v simulační verzi, zatímco na pravé části systém v reálném prostředí. Výhodou je, že program pro robota je zaměnitelný jak pro simulační verzi, tak pro jeho reálný protějšek.



Obrázek 55 Spuštění program s robotem

9.1 Pasivní i aktivní hra robota

Robot je schopen aktivní i pasivní hry. Pokud trajektorie puku směřuje na branku, puk je zachycen a následně odpálen směrem k soupeřově brance. Robot puk neodpaluje náhodně, ale počítá směr, kam puk střelit, aby trefil soupeřovu branku. Pokud trajektorie puku směřuje mimo zónu, kde se nachází branka, puk je ignorován. Tato zóna se dá nastavit, a je zde pro omezení pohybu robota. Pokud by zde toto omezení nebylo, robot by zbytečně přesouval na vzdálenější pozice. Toto omezení zde bylo přidáno z důvodu omezení opotřebování převodovek robota.

Robot je standardně umístěn ve středu obranné zóny své branky. Pokud je trajektorie puku směřována směrem pryč od robota, případně pokud není puk po delší dobu detekován, robot zůstává ve své výchozí pozici, popřípadě se do této pozice automaticky vrací.

V případě simulace byla implementována funkce, při níž je puk automaticky odražen z robotovy poloviny stolu v situaci, kdy se nachází v klidovém stavu. Při testování této funkcionality na reálném robotu se však její efektivita neprokázala, jelikož skutečné umístění robota není ideální a jeho dosah neumožňuje spolehlivý zásah do rohů stolu.

Robot dokázal být kvalitním protivníkem, který nehraje pouze defenzivní hru, ale také je schopen dát svému soupeři gól.

9.2 Hodnocení přesnosti detekce a predikce

Detekce puku může být vyhodnocena jako poměrně přesná. Puk se podařilo správně detekovat přibližně v 95 % pořízených snímků. Přesnost detekce byla ovlivněna několika faktory, zejména použitým osvětlením, přítomností čar na ploše stolu s podobnou barvou jako měl puk, a také občas nedostatečnou snímkovací frekvencí kamery, která se projevovala zejména při rychlém pohybu puku.

Přesnost predikce trajektorie puku závisela především na kvalitě detekce a stabilitě záznamu pohybu v čase. Při plynulém pohybu puku a bez výrazných výpadků detekce bylo dosaženo odchylky predikované trajektorie od skutečné dráhy v průměru do 4 cm. V případě náhlých změn směru, odrazů od mantinelu nebo dočasné ztráty detekce však mohla být odchylka výrazně vyšší. Na základě provedených testů lze konstatovat, že predikce je pro základní obranné reakce robota dostatečně přesná, avšak pro pokročilejší taktické pohyby by bylo vhodné predikční model dále rozšířit, například o zohlednění dynamiky odrazů nebo využití strojového učení.

9.3 Porovnání simulace a reálného prostředí

Při porovnání výsledků ze simulačního a reálného prostředí byly zaznamenány určité rozdíly, které pramení zejména z odlišných podmínek snímání a dynamiky pohybu puku. V simulačním prostředí byly podmínky zcela kontrolované – pohyb puku byl předvídatelný, osvětlení stabilní a neexistovala žádná rušivá pozadí. Díky tomu mohla být detekce i predikce prováděna s velmi vysokou přesností a bez výpadků.

V reálném prostředí však do hry vstupovaly proměnné, jako je nerovnoměrné osvětlení, odlesky od povrchu stolu, či vizuální podobnost puku s některými prvky na ploše. Tyto faktory vedly k častějšímu výskytu chyb v detekci a menší přesnosti predikce.

Celkově lze konstatovat, že výsledky dosažené v simulaci poskytly cenné podklady pro ladění algoritmů a logiky systému, avšak při přechodu do reálného prostředí bylo nutné provést úpravy detekčních parametrů a přizpůsobit reakční logiku robota aktuálním podmínkám.

9.4 Omezení a návrh na zlepšení

Nejvýznamnějším omezením celého systému byla identifikována snímkovací rychlost kamery a rychlost pohybu robota. V případech, kdy rychlost puku překročí určitou mez,

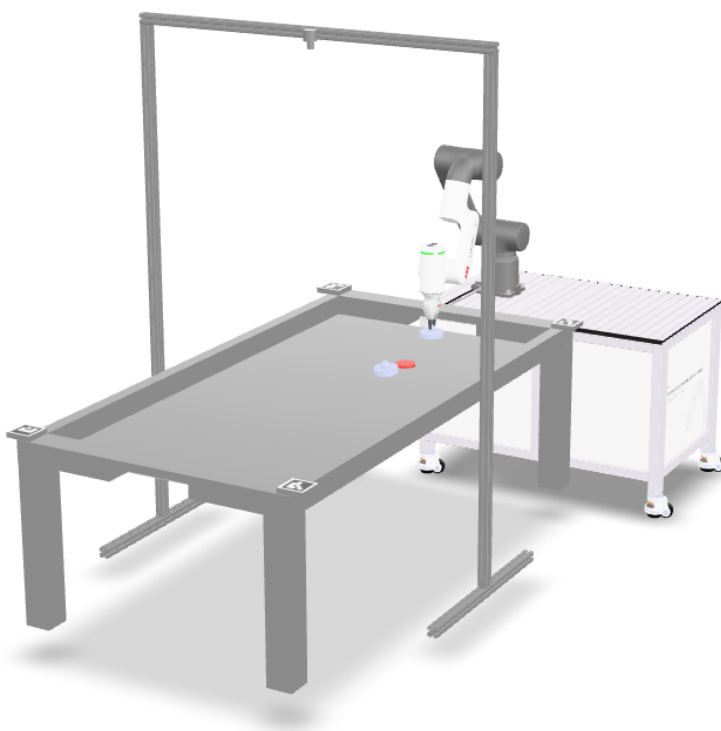
nemůže být puk detekován, neboť dochází k jeho rozmazání v obraze. Zároveň, pokud urazí mezi jednotlivými snímky příliš velkou vzdálenost, není možné spolehlivě predikovat jeho trajektorii.

Dalším omezením, jak již bylo zmíněno, je rychlost samotného robota. Vzhledem k tomu, že byl použit kolaborativní robot, jeho schopnost akcelerace je omezená. Pro dosažení rychlejší reakce by bylo vhodnější využití průmyslového robota, který by nabídl znatelně kratší reakční dobu a lepší dynamické vlastnosti ve hře. Za nejvhodnější typ robota pro tento typ aplikace by byl považován robot typu SCARA.

Navzdory těmto omezením se však ukázalo, že reakční schopnosti kolaborativního robota jsou pro účely demonstrace a ověření funkčnosti celého systému dostatečné.

9.5 AR A VR

Pro účely vizualizace může být simulace prezentována jako model rozšířené reality (AR), případně jako model virtuální reality (VR) při použití brýlí pro VR. V případě AR modelu není umožněna žádná interakce s objektem, model je pouze zobrazen v prostoru. Naopak při využití VR technologie může být prostřednictvím ovladačů realizována interakce s hokejkou a pukem, čímž je umožněna simulace hry v prostředí blízkém realitě.



Obrázek 56 AR model

ZÁVĚR

Cílem diplomové práce bylo vytvořit systém, ve kterém kolaborativní robot ABB plní roli protivníka ve hře air hockey. Systém je navržen tak, aby samostatně detekoval pohyb puku, předvídal jeho trajektorii a na základě těchto dat reagoval. Dále si systém dokáže samostatně provést kalibraci kamery, díky schopnosti detekovat ArUco značky a na základě těchto značek poté upravit a přizpůsobit hrací plochu.

Celý systém byl sestaven za využití knihovny OpenCV pro počítačové vidění, vlastních algoritmů pro predikci trajektorie a komunikačního rozhraní pro řízení pohybu robota prostřednictvím EGM (Externally Guided Motion). Pro testování funkčnosti byl vytvořen model v prostředí ABB RobotStudio, který umožnil ověření základních principů systému bez rizika poškození hardwaru. Vytvořený model navíc sloužil jako základ pro návrh a doladění reálného prostředí, ve kterém byl robot úspěšně implementován.

Systém je schopen vést aktivní i pasivní hru a reagovat na změnu trajektorie puku v reálném čase. Funkčnost celého řešení byla ověřena jak v simulačním, tak i v reálném prostředí. Výsledky testování potvrdily, že navržené řešení dosahuje očekávané úrovně přesnosti a spolehlivosti v rámci definovaných parametrů.

SEZNAM POUŽITÉ LITERATURY

- [1] CHANG, Ching-Lung; CHEN, Shuo-Tsung; CHANG, Chuan-Yu a JHOU, You-Chen. Application of Machine Learning in Air Hockey Interactive Control System. online. *Sensors*. 2020, roč. 20, č. 24. ISSN 1424-8220. Dostupné z: <https://doi.org/10.3390/s20247233>. [cit. 2024-11-25].
- [2] GOLD STANDARD AIR HOCKEY, . *USAA Official Rules of Air Hockey*. online. In: Gold Standard Air Hockey. . Dostupné z: <https://goldstandardairhockey.com/usaa-official-rules-of-air-hockey/>. [cit. 2025-01-18].
- [3] AIR HOCKEY NERDS, . *Air Hockey Strategy: 7 Tips to Win Every Time*. online. In: Air hockey nerds. 2021. Dostupné z: <https://airhockeynerds.com/air-hockey-strategy/>. [cit. 2025-01-18].
- [4] *The floating triangle*. online. In: . Dostupné z: <https://billystubbs.wordpress.com/2013/08/29/floating-triangle/>. [cit. 2025-04-11].
- [5] LIU, Puze; TATEO, Davide; BOU-AMMAR, Haitham a PETERS, Jan. Efficient and Reactive Planning for High Speed Robot Air Hockey. online. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, s. 586-593. ISBN 978-1-6654-1714-3. Dostupné z: <https://doi.org/10.1109/IROS51168.2021.9636263>. [cit. 2025-02-15].
- [6] TADOKORO, Koichiro; FUKUDA, Shotaro a NAMIKI, Akio. Development of Air Hockey Robot with High-Speed Vision and High-Speed Wrist. online. *Journal of Robotics and Mechatronics*. 2022, roč. 34, č. 5, s. 956-964. ISSN 1883-8049. Dostupné z: <https://doi.org/10.20965/jrm.2022.p0956>. [cit. 2025-02-15].
- [7] NAMIKI, Akio; MATSUSHITA, Sakyo; OZEKI, Takahiro a NONAMI, Kenzo. Hierarchical processing architecture for an air-hockey robot system. online. In: *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, s. 1187-1192. ISBN 978-1-4673-5643-5. Dostupné z: <https://doi.org/10.1109/ICRA.2013.6630722>. [cit. 2025-02-15].
- [8] SHIMADA, Hideaki; KUTSUNA, Yusuke; KUDOH, Shunsuke a SUEHIRO, Takashi. A two-layer tactical system for an air-hockey-playing robot. online. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, s. 6985-6990. ISBN 978-1-5386-2682-5. Dostupné z: <https://doi.org/10.1109/IROS.2017.8206624>. [cit. 2025-04-11].
- [9] DAY, Lewin. *Robot Air Hockey Player Predicts Your Next Move*. online. In: Hackaday. Dostupné z: <https://hackaday.com/2024/12/16/robot-air-hockey-player-predicts-your-next-move/>. [cit. 2025-02-15].
- [10] JAŠEK, Dominik. *Vzdušný hokej - realizace a návrh automatického hráče*. Diplomová práce. Brno: Vysoké učení technické v Brně, 2020.
- [11] LI, You-Cheng a HUANG, Shih-Yen. Fast-moving object tracking in air hockey. online. In: *2017 IEEE International Conference on Mechatronics and Automation (ICMA)*. IEEE, 2017, s. 1693-1697. ISBN 978-1-5090-6758-9. Dostupné z: <https://doi.org/10.1109/ICMA.2017.8016072>. [cit. 2024-11-25].
- [12] COGNEX. *What is machine vision?*. online. In: Cognex. 2021. Dostupné z: <https://www.cognex.com/what-is/machine-vision>. [cit. 2025-01-18].

- [13] *Evolution of Machine Vision*. online. In: Emergent Vision technologies. Dostupné z: <https://emergentvisiontec.com/tech-portal/evolution-of-machine-vision/>. [cit. 2025-01-18].
- [14] ŠURKALA, Milan. Umělá inteligence míří do snímačů: Sony uvádí IMX500 s AI procesorem. online. In: *Digimanie*. Dostupné z: <https://www.digimanie.cz/umela-inteligence-miri-do-snimacu-sony-uvadi-imx500-s-ai-procesorem/8337>. [cit. 2025-04-11].
- [15] MARR, Bernard. *What is Machine Vision And How Is It Used In Business Today?*. online. In: *Forbes*. Dostupné z: <https://www.forbes.com/sites/bernardmarr/2019/10/11/what-is-machine-vision-and-how-is-it-used-in-business-today/>. [cit. 2025-01-18].
- [16] ZEBRA. *What Is the Difference Between Machine Vision and Computer Vision?*. online. In: ZEBRA. Dostupné z: <https://www.zebra.com/us/en/resource-library/faq/what-is-the-difference-between-machine-vision-computer-vision.html>. [cit. 2025-01-18].
- [17] *Top 3 Camera Types Used in Vision Systems*. online. In: Sciotex. Dostupné z: <https://sciotex.com/camera-types-used-in-vision-systems/>. [cit. 2025-01-18].
- [18] CANON. *Image sensors explained*. online. In: . Dostupné z: <https://www.canon.cz/pro/infobank/image-sensors-explained/>. [cit. 2025-03-20].
- [19] ASSOCIATION FOR ADVANCING AUTOMATION. *Understanding Line Scan Applications*. online. In: Association for Advancing Automation. Dostupné z: <https://www.automate.org/vision/tech-papers/understanding-line-scan-applications>. [cit. 2025-03-20].
- [20] KASHYAPA, Raghava. *Area Scan vs Line Scan: What Fits You Better?*. online. In: *Qualitas Technologies*. Dostupné z: <https://qualitastech.com/image-acquisition/area-scan-vs-line-scan-cameras/>. [cit. 2025-03-20].
- [21] KOLÍBAL, Zdeněk. *Roboty a robotizované výrobní technologie*. První vydání. Brno: Vysoké učení technické v Brně - nakladatelství VUTIAM, 2016. ISBN 978-80-214-4828-5.
- [22] ASHTARI, Hossein. *CCD vs. CMOS: 5 Differences To Know*. online. In: *Spiceworks*. Dostupné z: <https://www.spiceworks.com/tech/tech-general/articles/ccd-vs-cmos/>. [cit. 2025-04-11].
- [23] *CCD vs CMOS: A Review of Sensor Technology*. online. In: CMOS Sensor INC. Dostupné z: <https://www.csensor.com/ccd-vs-cmos>. [cit. 2025-03-20].
- [24] *Selecting the Correct Industrial Machine Vision Camera*. online. In: Keyence. Dostupné z: <https://www.keyence.com/products/vision/vision-sys/resources/vision-sys-resources/selecting-the-correct-industrial-machine-vision-camera.jsp>. [cit. 2025-03-20].
- [25] *How much resolution is really needed? Part 2*. online. In: Videology. Dostupné z: <https://www.videologyinc.com/blog/how-much-resolution-for-industrial-camera>. [cit. 2025-03-20].
- [26] KUMAR, Prabu. *A short guide to why monochrome cameras have the edge over color cameras*. online. In: E-con System. Dostupné z: <https://www.e-consystems.com/blog/camera/technology/a-short-guide-to-why-monochrome-cameras-have-the-edge-over-color-cameras/>. [cit. 2025-03-20].

- [27] TECH, Enthusiast. *Considerations while working with (industrial) cameras*. online. In: Cryptiot. Dostupné z: <https://www.cryptiot.de/computer-vision/important-considerations-about-industrial-cameras/>. [cit. 2025-03-20].
- [28] *Sensor data fusion*. online. In: Bosch. Dostupné z: <https://www.bosch-mobility.com/en/solutions/sensors/sensor-data-fusion/>. [cit. 2025-03-20].
- [29] *Introduction to Industrial 3D Camera*. online. In: Voltrium systems. Dostupné z: <https://www.voltrium.com.sg/en/introduction-to-industrial-3d-camera/>. [cit. 2025-03-20].
- [30] *ToF kamera O3D311*. online. In: Ifm. Dostupné z: <https://www.ifm.com/cz/cs/product/O3D311>. [cit. 2025-04-11].
- [31] *A guide to the different lenses used in industrial vision systems*. online. In: Industrial Vision Systems. Dostupné z: <https://www.industrialvision.co.uk/news/a-guide-to-the-different-lenses-used-in-industrial-vision-systems>. [cit. 2025-03-20].
- [32] *Types of Machine Vision Lenses*. online. In: Edmund. Dostupné z: <https://www.edmundoptics.com/knowledge-center/application-notes/imaging/imaging-lens-selection-guide>. [cit. 2025-03-20].
- [33] SHARMA, Aman a KULKARNI, Ambarish. Vision System for Smart Manufacturing: A Review. online. In: *2023 IEEE Engineering Informatics*. IEEE, 2023, s. 1-9. ISBN 979-8-3503-3852-2. Dostupné z: <https://doi.org/10.1109/IEEECONF58110.2023.10520579>. [cit. 2025-03-20].
- [34] *Industrial AI Cameras*. online. In: Advantech. Dostupné z: https://www.advantech.com/en/products/industrial-ai-cameras/sub_f85f4265-fae1-4edc-900f-07b578ad857b. [cit. 2025-03-20].
- [35] GE, Fan; WANG, Guizhou; HE, Guojin; ZHOU, Dengji; YIN, Ranyu et al. A Hierarchical Information Extraction Method for Large-Scale Centralized Photovoltaic Power Plants Based on Multi-Source Remote Sensing Images. online. *Remote Sensing*. 2022, roč. 14, č. 17. ISSN 2072-4292. Dostupné z: <https://doi.org/10.3390/rs14174211>. [cit. 2025-03-20].
- [36] 8. *Digital image processing*. online. In: Esa. Dostupné z: https://www.esa.int/Education/8._Digital_image_processing. [cit. 2025-03-20].
- [37] AHIRE, Shrikant. *Vectors and Matrices: The Building Blocks of Computer Vision*. online. In: Medium. Dostupné z: <https://medium.com/@shrivahire/vectors-and-matrices-the-building-blocks-of-computer-vision-5098fd311bb4>. [cit. 2025-03-20].
- [38] *Differences between TCP and UDP*. online. In: GeeksforGeeks. Dostupné z: https://www.geeksforgeeks.org/differences-between-tcp-and-udp/?ref=header_outind. [cit. 2025-03-20].
- [39] OPENCV. *About OpenCV*. online. In: OPENCV. Dostupné z: <https://opencv.org/about/>. [cit. 2025-03-20].
- [40] ATMAJA, Ratri; MURTI, Muhammad; HALOMOAN, Junartha a SURATMAN, Fiky. An Image Processing Method to Convert RGB Image into Binary. online. *Indonesian Journal of Electrical Engineering and Computer Science*. 2016, roč. 3, č. 2, s. 377-382. ISSN 2502-4760. Dostupné z: <https://doi.org/10.11591/ijeecs.v3.i2.pp377-382>. [cit. 2025-03-20].
- [41] MONTANARI, Tobia. *HSL and HSV Explained: Which Color Model Should You Use?*. online. In: Tobia Montanari. Dostupné z: <https://www.tobiamontanari.com/hsl-and-hsv-explained-which-color-model-should-you-use/>. [cit. 2025-03-20].

- [42] ROSEBROCK, Adrian. *Detecting Circles in Images using OpenCV and Hough Circles*. online. In: PyImageSearch. Dostupné z: <https://pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/>. [cit. 2025-03-20].
- [43] ROSEBROCK, Adrian. *OpenCV center of contour*. online. In: PyImageSearch. Dostupné z: <https://pyimagesearch.com/2016/02/01/opencv-center-of-contour/>. [cit. 2025-03-20].
- [44] SHAIKH, Raqueeb. *OpenCv Perspective Transformation*. online. In: Medium. Dostupné z: <https://medium.com/analytics-vidhya/opencv-perspective-transformation-9edffefb2143>. [cit. 2025-03-20].
- [45] RAO, Ravia. *Cobots vs Robots: Understanding the Key Differences and Applications*. online. In: Wevolver. Dostupné z: <https://www.wevolver.com/article/cobot-vs-robot>. [cit. 2025-03-20].
- [46] *ABB píše budoucnost digitálního průmyslu*. online. In: České noviny. Dostupné z: <https://www.ceskenoviny.cz/zpravy/1805097>. [cit. 2025-04-11].
- [47] *ABB vrátilo z Ostravy do výroby už 3.878 robotů od roku 2007*. online. In: ABB. Dostupné z: <https://new.abb.com/news/cs/detail/102747/abb-vratilo-z-ostravy-do-vyroby-uz-3878-robotu-od-roku-2007>. [cit. 2025-04-11].
- [48] *About ABB*. online. In: ABB Group. Dostupné z: <https://global.abb/group/en/about>. [cit. 2023-03-28].
- [49] *Physics simulation from Algorix shapes the future of robotics at ABB*. online. In: Algorix. Dostupné z: <https://www.algorix.se/abb-shapes-future-of-robotics-with-algorix/>. [cit. 2023-03-28].
- [50] *Technical reference manual RAPID Instructions, Functions and Data types, 3HAC065038-001*. online. In: . [cit. 2025-03-20].
- [51] *Air Hockey Stoly*. online. In: Game-center. Dostupné z: <https://www.game-center.cz/sct/910490/Air-hockey>. [cit. 2025-04-11].
- [52] *ABB Product specification - CRB 15000: 3HAC077390-001. Revize N. 2024*. online. In: . [cit. 2025-04-07].
- [53] *Aca1920-40uc*. online. In: ATEsystem. Dostupné z: <https://eshop.atesystem.cz/d/837195/aca1920-40uc>. [cit. 2025-04-28].
- [54] *M0814-MP2*. online. In: Elcom. Dostupné z: <https://eshop.prumyslovekamery.cz/Megapixelove/539-M0814-MP2>. [cit. 2025-04-28].
- [55] *NumPy documentation*. online. In: . Dostupné z: <https://numpy.org/doc/stable/>. [cit. 2025-04-11].
- [56] *Pypylon*. online. In: Github. Dostupné z: <https://github.com/basler/pypylon>. [cit. 2025-04-11].
- [57] *Pillow*. online. In: . Dostupné z: <https://pillow.readthedocs.io/en/stable/>. [cit. 2025-04-11].
- [58] *Socket — Low-level networking interface*. online. In: . Dostupné z: <https://docs.python.org/3/library/socket.html>. [cit. 2025-04-16].

SEZNAM OBRÁZKŮ

Obrázek 1 Stůl pro Air Hockey [2]	15
Obrázek 2 Vzhled hokejek a puků pro Air Hockey	16
Obrázek 3 Správný úchop hokejky	17
Obrázek 4 Trianglová obrana [4]	19
Obrázek 5 Tří člankový robotický hráč [6]	20
Obrázek 6 hierarchická architektura řízení [7]	21
Obrázek 7 Robotický soupeř v air hockey [8]	21
Obrázek 8 Diplomová práce práce Bc. Dominika Jaška [10]	22
Obrázek 9 H-bot air hockey hráč [9]	23
Obrázek 10 První elektronický televizní systém [13]	25
Obrázek 11 HIVIP Mk.1, První inteligentní robot s vision systémem [13]	26
Obrázek 12 Sony IMX500 [14]	26
Obrázek 13 Řádkový snímač [19]	28
Obrázek 14 Plošný snímač [19]	29
Obrázek 15 Princip CCD snímače [22]	30
Obrázek 16 Princip CMOS snímače [22]	31
Obrázek 17 Pohled na částici v zorném poli v různém rozlišení [24]	32
Obrázek 18 Keyence Inteligentní průmyslová smart kamera [24]	33
Obrázek 19 ToF kamera [30]	34
Obrázek 20 Objektivy pro průmyslové kamery [31]	35
Obrázek 21 Možnosti využití knihovny OpenCV [39]	40
Obrázek 22 Kolaborativní robot [46]	45
Obrázek 23 Průmysloví roboti [47]	46
Obrázek 24 Stoly pro Air Hockey [51]	51
Obrázek 25 Vybraný stůl Buffalo Dominator 7FT [51]	52
Obrázek 26 Kolaborativní robot ABB GoFa CRB 15000 [52]	53
Obrázek 27 Load diagram robota ABB GoFa CRB 15000 [52]	53
Obrázek 28 Kamera Basler ace U acA1920-40uc [53]	54
Obrázek 29 Objektiv Computar M0814-MP2 [54]	55
Obrázek 30 Výpočet parametrů objektivu pomocí online kalkulačky Basler	55
Obrázek 31 Vývojový diagram funkce vision systému	58
Obrázek 32 Puk pro hru Air Hockey	61
Obrázek 33 Script pro výběr HSV masky	62
Obrázek 34 Binární obraz s pukem	63

Obrázek 35 Nalezený objekt pomocí funkce cv2.findContours()	64
Obrázek 36 Výpočet rychlosti puku	65
Obrázek 37 Predikování finálního bodu	65
Obrázek 38 Pokračování predikce trajektorie při výpadku detekce	66
Obrázek 39 Predikce trajektorie puku	66
Obrázek 40 Funkce pro odesílání dat robotu	68
Obrázek 41 Vytvořený model stolu pro Air Hockey	70
Obrázek 42 Nastavení fyzikálních vlastností stolu a objektů	71
Obrázek 43 Workobject jako základní souřadnicový systém	72
Obrázek 44 Spuštění programu a výběr ROI.....	73
Obrázek 45 Spuštění program.....	73
Obrázek 46 Koncový efektor pro uchopení hokejky	75
Obrázek 47 Návrh konstrukce pro kameru	76
Obrázek 48 Postavená konstrukce pro kameru	76
Obrázek 49 Obraz pořízený kamerou na nosné konstrukci	77
Obrázek 50 Kód v robotu pro řízení pohybu přes EGM.....	77
Obrázek 51 Stůl s ArUco značkami.....	79
Obrázek 52 Automaticky vybraný obdélník značící hrací plochu.....	79
Obrázek 53 Výstup programu s obrazem reálné kamery.....	80
Obrázek 54 Kolaborativní robot jako oponent ve hře Air Hockey	81
Obrázek 55 Spuštění program s robotem.....	82
Obrázek 56 AR model	84

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

ToF time-of-flight

ROI Region of interest (region zájmu)

EGM Externally Guided Motion

AR Augmented reality (Rozšířená realita)

VR Virtual Reality (Virtuální Realita)