

VOŠ A SPŠ ŽDÁR NAD SÁZAVOU

OBOR INFORMAČNÍ TECHNOLOGIE

TRACKBOT

TRACKBOT

2022/2023

LUKÁŠ MACHOVEC



Zadání maturitní práce

profilová část – praktická maturitní zkouška

Jméno, příjmení a e-mail žáka:	Lukáš Machovec, machy.lukas@seznam.cz
Číslo a název oboru studia:	18-20-M/01 Informační technologie
Třída:	4. E
Vedoucí práce (jméno, pracoviště, e-mail):	Ing. Petr Fořt, VOŠ a SPŠ Žďár nad Sázavou, fort@spszr.cz

Ředitelství školy Vám zadává maturitní práci na téma: Trackbot

Zadání práce: Trackbot

Teoretická část

- a) Úvod
- b) 3D tisk
- c) Arduino
- d) Algoritmy pro rozpoznávání objektů
- e) knihovna OpenCV jazyka Python

Praktická část

- f) Návrh konceptu
- g) Program na sledování uživatele
- h) Komunikace mobilního zařízení a mikro-ovladače
- i) Grafické uživatelské rozhraní – TrackBotUI
- j) Finální model a jeho tisk
- k) Zhotovení a testování prototypu
- l) Dodatečné úpravy

Datum zadání maturitní práce: 14.12.2022

Datum odevzdání maturitní práce: 19.04.2023

.....
Ing. Jaroslav Kletečka



Anotace

Má práce se zabývá problematikou počítačového vidění ve vazbě na mikroprocesorovou technologii. V teoretické části popisuji princip 3D tisku, mikroprocesor Arduino, různé algoritmy pro rozpoznávání objektů a na konec pár slov o programovacím jazyku Python. Druhá, praktická část pojednává o mém pokusu vytvořit funkční prototyp robota, který za pomoci uživatelského telefonu dokáže sledovat a natáčet lidský pohyb.

Klíčová slova

OpenCV; Tracking; Arduino; Python-kivy; 3D Tisk

Annotation

My work deals with the concept of computer vision in relation to microprocessor technology. In the theoretical part I describe the fundamentals of 3D printing, microprocessor Arduino, different kinds of computer vision algorithms and a few closing words about programming language Python. Second part follows my efforts at constructing a functioning prototype, which thanks to the user's mobile phone can track and film human movement.

Keywords

OpenCV; Tracking; Arduino; Python-kivy; 3D Printing

Prohlášení

Prohlašuji, že jsem svou maturitní práci vypracoval samostatně pod vedením pana Ing. Petra Fořta a použil jsem pouze prameny a literaturu uvedenou p seznamu bibliografických záznamů.

Prohlašuji, že tištěná i elektronická verze maturitní práce jsou shodné.

.....

Lukáš Machovec

16. 4. 2023

Poděkování

Primárně bych chtěl poděkovat svému dědovi, Ing. Ladislavu Němcovi, za pomoc s technickým návrhem modelu. Byl nevídanou oporou. Dále Ing. Jaroslavovi Příbramskému za pomoc s prací na aplikaci, několikrát mi pomohl v situacích, kdy jsem nenacházel řešení. Děkuji Ing. Petru Nobstovi za konzultace ohledně funkčnosti projektu. A také děkuji Janu Kalábovi za pomoc se stylistikou, tvorbou dokumentu, který se chystáte přečíst a jeho četné rady k tvorbě modelu.

Obsah

Úvod	8
a) Model	8
b) Elektronika	8
c) Software	8
Teoretická část	9
1 3D tisk	9
1.1 Modelování objektu a příprava tisku	9
1.2 Proces tisknutí	10
1.3 Druhy filamentu	11
1.3.1 PLA	11
1.3.2 PETG	11
1.3.3 ABS	12
2 Arduino	12
2.1 Práce s Arduinem	13
2.2 Využití a druhy	13
3 Algoritmy pro rozpoznávání objektů	14
3.1 YOLOv4 algoritmus	15
3.2 Konkrétní příklady algoritmů	15
4 knihovna OpenCV jazyka Python	15
Praktická část	17
5 Návrh konceptu	17
6 Program pro sledování uživatele	17
6.1 Programování v C++	18
6.2 Algoritmizace a Python	18
6.3 Tvorba detekčního algoritmu	19
7 Komunikace mobilního zařízení a mikro-ovladače	20
8 Změna operačního systému	21
8.1 Testování Bluetooth	21
9 Android aplikace	22
9.1 Implementace funkcí	22
10 Grafické uživatelské rozhraní – TrackBotUI	23
11 Finální model a jeho tisk	24

12	Zhotovení a testování prototypu	24
13	Dodatečné úpravy	25
	Závěr	26
	Použitá literatura a zdroje	27
	Přílohy	29
1.	Součásti modelu	29
2.	Program trackbotu	29
3.	Kód Arduina	29

Seznam ilustrací

Obrázek 1:	Vývojové prostředí CAD softwaru Autodesk Inventor	9
Obrázek 2:	Vývojové prostředí softwaru Prusaslicer	10
Obrázek 3:	Ilustrace FDM tisku	11
Obrázek 4:	Porovnání filamentů PLA, ABS a PETG	12
Obrázek 5:	Arduino UNO	14
Obrázek 6:	Arduino Nano	14
Obrázek 7:	Příklad užití mého algoritmu	20

Úvod

Mým cílem bylo vytvořit kompaktní, přenosné zařízení, které po upevnění uživateleova telefonu bylo schopné sledovat každý jeho pohyb. Důvod tvorby zařízení pro „tracking“ a natáčení uživatele (dále jenom Trackbot) byl motivován primárně zámkou zlepšit se v užívání technologií použitých při jeho tvorbě. Dále vidina praktického uplatnění v oblasti sociálních sítí, telekomunikací a video tvorby. Samotná realizace se skládá ze tří hlavních částí:

a) Model

Nejdůležitější část každého modelu je návrh. Při jeho tvorbě je třeba myslet nejen na konečnou funkčnost, ale i na technické specifikace, design a proces výroby. V mém případě jsem se zaměřil primárně na intuitivní ovládání a plynulý chod modelu s kompaktním designem, nezávislým na internetovém připojení. Konečný výrobek je složen ze tří hlavních komponent – podstavy a dvou ramen. Podstava zodpovídá za stabilitu celé sestavy, horizontální otáčení ramen, spolu s vytvořením prostoru pro

umístění ovládacího mikroprocesoru a další elektroniky. Hlavní funkce ramen zahrnuje uchycení telefonu uživatele, včetně jeho vertikální rotace.

b) Elektronika

Stejně jako vnitřnosti v našem těle, elektronika má za úkol vytvořit mechanismus, díky kterému bude moci hlavní řídicí algoritmus – mozek, ovládat pohyb a momentální stav zbytku těla. Propojení všech dílčích elektronických součástí bude provedeno metalickými kabely, s výjimkou propojení telefonu a mikroprocesoru, kde je přenos dat zajištěn bezdrátově, skrze Bluetooth adaptér. Elektrický proud, na pohánění samotných komponent, bude poskytovat 9 voltová vyměnitelná baterie.

c) Software

Funkce softwaru je vdechnout celému zařízení život. Bez řídicího algoritmu by byl Trackbot pouze hromada plastu a kovu. Mnou naprogramovaná aplikace dává uživateli plnou kontrolu nad celým zařízením, spolu s různými nastaveními upravující jeho chod, zatímco řídicí algoritmus nespouští zrak z uživatele. Souhra softwaru a hardwaru je nezbytná pro úspěšný průběh natáčení videa. A právě z tohoto důvodu byl jak software, tak hardware dodatečně laděn, aby dokázali bezchybně spolupracovat.

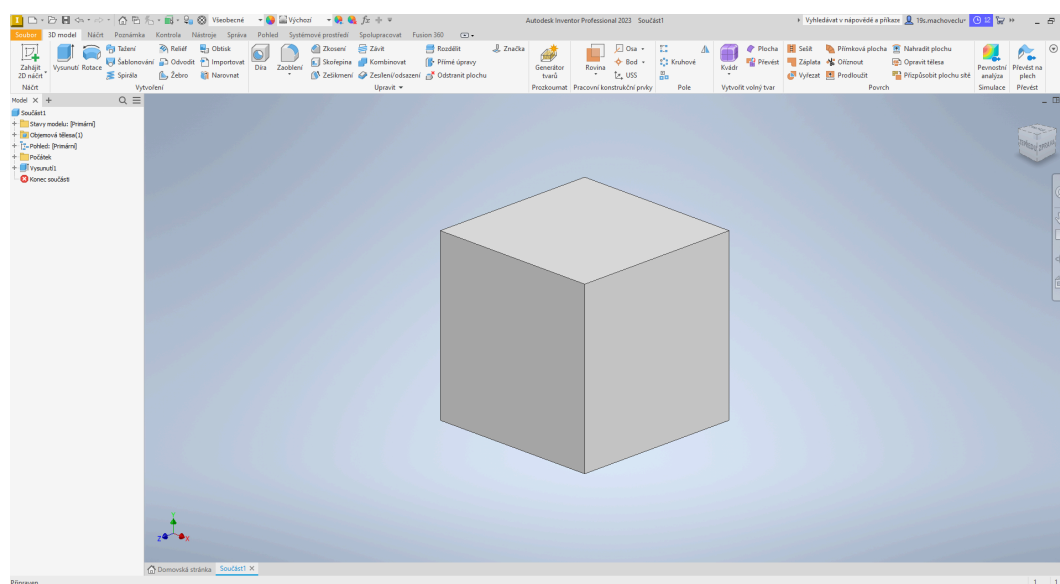
TEORETICKÁ ČÁST

1 3D TISK

Technologie 3D tisku už léta nabírá na popularitě bez známky zpomalování.¹ A není divu. Tento revoluční vynález nanášení malých vrstviček plastu nám umožňuje vytvořit de facto jakýkoliv objekt nás napadne, což si našlo využití v mnoha oborech. Patří mezi ně například automobilový průmysl, stavebnictví, letectví nebo průmyslová výroba.² V praxi se využívá velké množství druhů 3D tisku, podle jeho využití. V této práci se budu zabývat tím pro nás nejdostupnějším typem tisku – FDM. Ale jak vlastně takový 3D tisk funguje?

1.1 Modelování objektu a příprava tisku

Ačkoli je to škoda, 3D tisk nedokáže vytisknout cokoli nás napadne pouze na základě myšlenky v naší hlavě. Aby tiskárna věděla, co chceme vytisknout, je za potřebí předem vytvořit tzv. digitální předlohu neboli model. „Model je digitální soubor, který zobrazuje třídimenzionální pevný objekt. 3D modely jsou obvykle vytvářeny ve specializovaných softwarech – CAD nástrojích.“³ Příkladem těchto nástrojů je Autodesk Inventor, Fusion 360 nebo Blender.



Obrázek 1: Vývojové prostředí CAD softwaru Autodesk Inventor

Když už máme model hotový, je na čase ho připravit k tisku. Příkladem toho je třeba nastavení průměru vrstviček plastu, teploty tavení plastu, prostorová orientace modelu, anebo druh výplně konečného výrobku. Všechny předem zmíněné činnosti, již ale nelze upravit v CAD softwaru a je za potřebí použít další dedikovaný software – Slicer. Pravděpodobně

¹ Global 3D printing market size 2013-2021 | Statista.

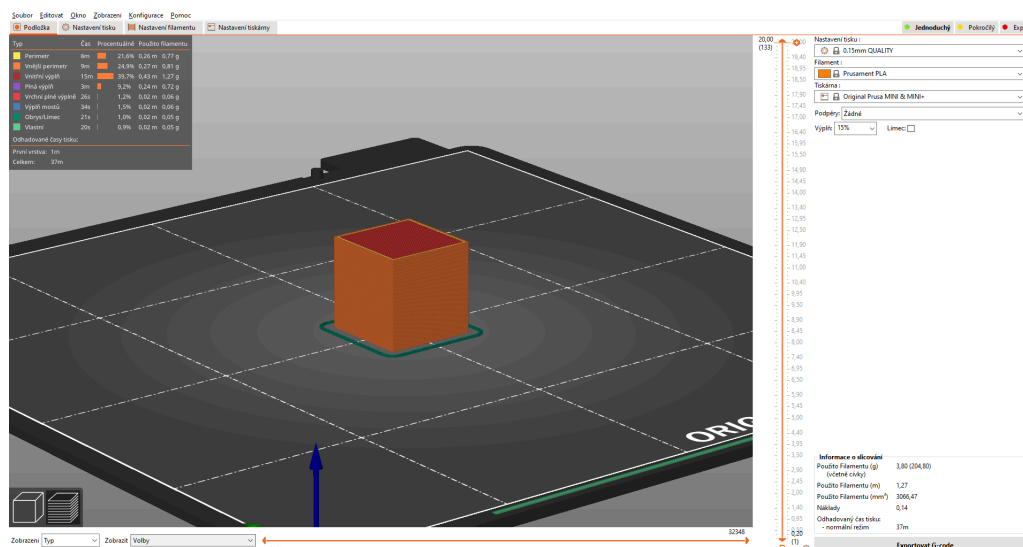
² WAYKEN, Rapid. 5 industrial applications of 3D printing services.

³ [FAQ] Co je to 3D model? | 3Dees.

Čechům nejznámějším slicerem bude Prusaslicer, neboť jeho tvář je Josef Průša – český vynálezce 3D tiskáren.⁴

Po nastavení samotného tisku už zbývá pouze poslední krok, a to převést model společně s jeho nastaveními, do podoby, které dokáže tiskárna rozumět. Proces, nímž tohoto docílíme se nazývá slicování. Během něho software vygeneruje také předpokládanou vizualizaci modelu a informace o množství spotřebovaného materiálu nebo orientační délce tisku. Po skončení slicování na nás čeká soubor plný počítačových instrukcí – Gcode. Do tiskárny můžeme soubor nahrát buď přes SD kartu nebo USB flash disk.

⁴ Josef Průša (vývojář) – Wikipedie.



Obrázek 2: Vývojové prostředí softwaru Prusaslicer

1.2 Proces tisknutí

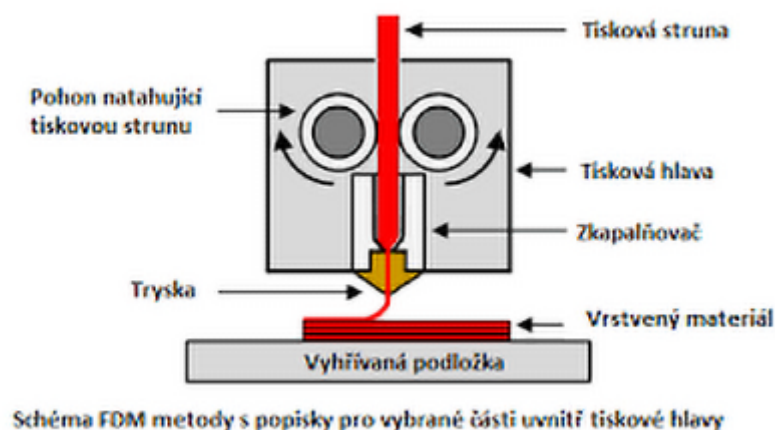
Nehledě na použité technologii 3D tisku, všechny potřebují výchozí model, avšak proces jeho tisku se velmi liší. V případě SLA tisku je tekutý materiál vytvrzován v jednotlivých vrstvách. U SLS tisku je na sebe materiál navázán díky laseru nebo lepidla. V našem případě FDM tisku se však materiál vytlačuje vrstvu po vrstvě tiskovou hlavou.⁵

Materiál používaný u při FDM tisku se nazývá filament. Jedná se o tenké struny plastu, namotané na cívce s běžnou tloušťkou v rozmezí 1,75 a 2,85 cm.⁶ Filament je následně zaveden do extruderu, což je pohyblivá část tiskárny, kde se plastová struna taví. Extruder potom nanáší jednotlivé roztavené vrstvy plastu na pohyblivou podložku, kde rychle tuhnou. Ve chvíli, kdy tiskárna dotiskne celou první vrstvu modelu, extruder se posune o trochu výš a začne tisknout vrstvu druhou. Tento proces se opakuje, dokud není celý model vytisknutý.⁷ Pohyb jednotlivých dílů tiskárny se liší značka od značky, ale princip, kdy se podložka pohybuje po ose Y a extruder se pohybuje po ose X a Z, je dle mého nejběžnější.

⁵ Jak funguje 3D tiskárna. 3D Tiskárna - Abc3D nejlepší pro 3D tisk

⁶ 3D tisk: Podle čeho a jaký filament vybrat?. Alza.cz

⁷ 3D Printing with FDM - YouTube. YouTube



8

Obrázek 3: Ilustrace FDM tisku

1.3 Druhy filamentu

Stejně jako existuje mnoho druhů tisku, existuje také velké množství filamentů. Materiály se mezi sebou liší například teplotní odolností, rázovou houževnatostí, teplotou tání a samozřejmě cenou. Zde si přiblížíme vlastnosti a použití tří druhů nejpopulárnějších typů filamentu.

1.3.1 PLA

Nejoblíbenějším materiálem, a to obzvláště pro začátečníky, je filament typu PLA.⁹ Důvodem pro jeho popularitu je především jednoduchá práce při tisku, odolnost a nízká cena materiálu. Na rozdíl od jiných typů, je filament PLA vyroben z rostlinných zdrojů, což z něj tvoří velmi bezpečný a netoxický materiál. Mezi jeho nevýhody patří však křehkost a malá flexibilita. Filament si díky všem jeho vlastnostem našel využití třeba v prototypování, tvorbě kostýmu a všude jinde, kdy není na materiál vystavován velkým nárazům nebo ohýbání.¹⁰

1.3.2 PETG

Další velmi používaný materiál, jehož jméno naznačuje, že se jedná o plast. Na rozdíl od klasického plastu, který všichni známe z nápojových láhví, má tento filament příměs glykolu. Výsledný materiál je charakteristický svou pevností a odolností na čemž se glykol z větší části podílí. Mezi jeho vlastnosti dále patří tepelná odolnost a mírná flexibilita. Filament je potřeba skladovat v suchém prostředí, protože jednoduše nasává vlhkost.¹¹ PETG je perfektní pro výrobu součástek na které bude vyvíjen tlak nebo prudké nárazy.¹²

⁸ Ekonomické zhodnocení FDM technologie 3D tisku v elektrotechnickém průmyslu. ČVUT DSpace

⁹ Typy filamentů a resinů | 3D tisk Silicon Hill.

¹⁰ 3D tisk: Podle čeho a jaký filament vybrat?. Alza.cz

¹¹ 3D tisk: Podle čeho a jaký filament vybrat?. Alza.cz

¹² Typy filamentů a resinů | 3D tisk Silicon Hill.

1.3.3 ABS

Posledním zástupcem z řady populárních materiálů bude filament ABS. Využití si našel primárně kvůli své houževnatosti nebo teplotní a mechanické odolnosti. Skvělá vlastnost je také možnost lepení a vyhlazení acetonem. Používán je například při výrobě LEGO kostek, dílů do aut nebo, krytů od telefonu atd. Na co je u tohoto typu filamentu dbát je dobře větraná místnost při tisku. Materiál totiž vylučuje nebezpečné výpary, které by mohli ohrozit zdraví.¹³

	PLA	ABS	PET-G
Teplota trysky	180 - 230 °C	210 - 250 °C	220 - 260 °C
Teplota podložky	20 - 60 °C	80 - 110 °C	60 - 90 °C
Tisková podložka	Nepovinná	Povinná	Doporučuje se
Komora při tisku	Nepovinná	Doporučuje se	Nepovinná
Přilnutí první vrstvy	Dobré	Drobné problémy	Drobné problémy
Výpary	Skoro žádné	Silné	Silné
Absorbce vlhkosti	Ano	Ano	Ano
Youngův modul (GPa)	3,5	2,4	2,2
Modul pružnosti (GPa)	4	2,2	1,9
Pevnost v ohybu (MPa)	80	65	64
Poměr pevnosti k hmotnosti (kN*m/kg)	40	31-80	42
Pevnost v tahu (MPa)	110	37-110	53
Odolnost proti nárazu (J/m)	-	70-370	77
Tvrdost Rockwell R	-	94	108
Teplota skelného přechodu (°C)	60	105	81
Deformace při teplotě (°C)	65	100	70
Teplota tání (°C)	160	-	140
Tepelná kapacita (J/kg*K)	1800	1470	1200
Tepelná vodivost (W/m*K)	0,13	0,17	0,29
Tepelná difuzivita (m ² /s)	0,058	0,12	0,19

14

Obrázek 4: Porovnání filamentů PLA, ABS a PETG

2 ARDUINO

Doby, kdy jsme i pro z dnešního pohledu banální funkce, museli zaplnit obrovské haly počítačovou technikou, jsou dávno pryč. Pro srovnání, průměrný smartphone je 900milionkrát výkonnější než počítač, který používala NASA při misi APOLLA 11 v roce 1969¹⁵, přičemž je zhruba 400krát menší.¹⁶ Problémem je, že pro určité použití, počítače velikosti telefonu stále nejsou dostačující.

Zde přichází řada na mikropočítače. Ačkoli nedisponují takovým výkonem, jako dnešní mobilní telefony nebo stolní počítače, jsou menší, lehčí a nespotebovávají tolik elektrické energie. Příkladem takového mikropočítače je například Arduino.

2.1 Práce s Arduinem

Arduino je inovativní platforma pro vývoj elektronických projektů, která se stává stále populárnější mezi technologickými nadšenci po celém světě. Přirovnání k počítači by nebylo

¹³ ABS | Prusa Knowledge Base. *Prusa Knowledge Base*

¹⁴ Rozdíl mezi ABS, PLA a PETG. *Materialpro3d.cz*

¹⁵ Fast-forward — comparing a 1980s supercomputer to a modern smartphone | Adobe Blog

¹⁶ What size smartphone do you need?. *Coolblue*

daleko od pravdy, neboť Arduino lze v mnohém považovat za malý počítač, který se skládá z mikro ovladače, paměti a dalších komponent.

Práce s Arduinem je velmi snadná a přístupná i pro ty, kteří nemají předchozí zkušenosti s programováním a elektronikou. Vývojové prostředí Arduina je intuitivní a umožňuje uživatelům jednoduše vytvářet programy v jazyce Wiring, který je velmi podobný jazyku C. Tyto programy jsou následně nahrány do mikro ovladače¹⁷.

Nejčastější využití Arduina je v oblasti robotiky a automatizace, kde se Arduino používá k ovládání motorů, senzorů a dalších komponent. V oblasti internetu věcí se Arduino používá k propojení různých zařízení a k získávání dat z okolního prostředí. Příklady projektů s Arduinem zahrnují robotická auta, senzory pro sledování kvality ovzduší, inteligentní domácnosti a mnoho dalšího.

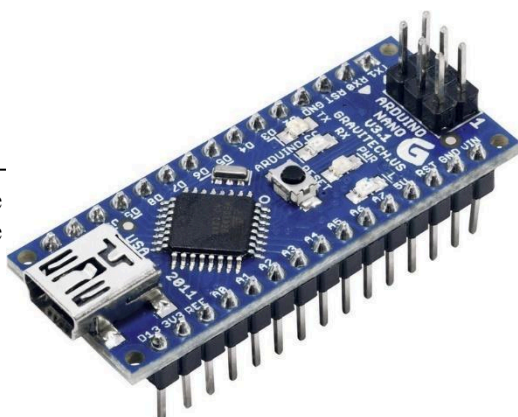
2.2 Využití a druhy

Existuje mnoho druhů Arduina, které se liší velikostí, funkcemi a cenou. Arduino Uno je základní model, který je vhodný pro začátečníky a stojí kolem 200 korun. Pro pokročilé uživatele jsou k dispozici výkonnější modely jako Arduino Mega a Arduino Due. Existují také speciální verze Arduina, které jsou navrženy pro konkrétní účely, jako například Arduino Nano pro malé projekty nebo Arduino Yun pro projekty IoT.

Velikost Arduina je ideální pro vývoj projektů, kde je potřeba malé zařízení, které je snadno přenosné. Navíc je cena Arduina relativně nízká v porovnání s jinými vývojovými platformami, což z něj dělá přístupnou volbu pro amatérské i profesionální vývojáře.¹⁸



Obrázek 5: Arduino UNO



¹⁷ Co je

¹⁸ Co je

Obrázek 6: Arduino Nano

3 ALGORITMY PRO ROZPOZNÁVÁNÍ OBJEKTŮ

Moravcův paradox praví, že počítače je jednoduché naučit věci, které jsou pro lidi těžké, jako například matematika nebo logika, ale složité naučit věci které lidem jdou; třeba chůze.¹⁹ Stejně tak to platí u rozpoznávání objektů. Všichni dokážeme (za normálních podmínek) bez problému rozpoznat reproduktor od rychlovarné konvice, člověka od lampy. Počítač to ale tak jednoduše nezvládne. Museli jsme pro to vyvinout speciálně trénované algoritmy, které budou schopné objekty identifikovat, od sebe rozlišit, anebo je i najít.

Algoritmy pro rozpoznávání objektů jsou dnes nezbytné pro mnoho aplikací, jako například autonomní řízení, sledování objektů, bezpečnostní systémy a další. Tyto algoritmy se staly součástí strojového učení, kde se snažíme naučit počítače rozpoznávat objekty podobně jako to děláme my lidé.²⁰

3.1 YOLOv4 algoritmus

Jedním z nejúspěšnějších algoritmů pro rozpoznávání objektů je YOLOv4 (You Only Look Once version 4), který byl vytvořen v roce 2020. Tento algoritmus funguje tak, že celý obraz je rozdělen na mřížku a každé buňce této mřížky je přiřazen tzv. anchor box.²¹ Tyto anchor boxy se používají k určení pozice a velikosti objektů v každé buňce.

Algoritmus YOLOv4 pracuje v několika fázích. Nejprve se vstupní obraz zpracuje pomocí konvoluční neuronové sítě, která extrahuje různé funkce z obrazu. Poté se použijí anchor boxy k detekci objektů v každé buňce. Algoritmus YOLOv4 používá také techniky, pro odstranění překrývajících se detekcí, čímž se zvyšuje přesnost.²²

¹⁹ Moravec's Paradox: Definition, Explanation and Examples. *Science ABC*

²⁰ Počítačové vidění – Wikipedie.

²¹ Getting Started with YOLO v4 | *MathWorks*

²² YOLOv4 - An explanation of how it works. *Roboflow Blog*

3.2 Konkrétní příklady algoritmů

Kromě YOLOv4 existuje mnoho dalších algoritmů pro rozpoznávání objektů, které mají své výhody a nevýhody v porovnání s YOLOv4. Mezi nejznámější algoritmy patří například Faster R-CNN, SSD nebo Mask R-CNN.²³

Faster R-CNN je jeden z nejstarších a nejúspěšnějších algoritmů pro rozpoznávání objektů. Používá složitější architekturu a výpočetně náročnější postup než YOLOv4, ale vyniká vysokou přesností detekce a lokalizace objektů.

SSD je algoritmus, který využívá podobný přístup jako YOLOv4. Celý obraz se rozdělí na mřížku a v každé buňce se detekují objekty pomocí anchor boxů. Rozdílem je, že SSD používá více mřížek s různými velikostmi boxů, což umožňuje detekovat objekty různých velikostí.

Mask R-CNN je algoritmus, který navíc k detekci objektů umožňuje i segmentaci, tedy přesné vymezení hranic objektů na obrazu. Tento algoritmus se používá například v medicíně k diagnostice nádorů nebo v robotice k rozpoznávání předmětů.

YOLOv4 se v porovnání s ostatními algoritmy vyznačuje vysokou rychlostí detekce, čímž je ideální pro aplikace, které vyžadují rychlou odezvu. Existuje také verze YOLOv4-tiny, která je menší a jednodušší, ale méně přesná než plná verze YOLOv4. YOLOv4 je v současné době jeden z nejpoužívanějších algoritmů pro rozpoznávání objektů v praxi, především v oblasti autonomního řízení a bezpečnostních systémů.

4 KNIHOVNA OPENCV JAZYKA PYTHON

Tvorba algoritmů počítačové vize, jako rozpoznávání předmětů nebo sledování objektů, je těžká disciplína. Naštěstí dnes už máme k dispozici sady nástrojů, které nám tvorbu algoritmů velmi ulehčují. Knihovny pro počítačovou vizi obsahují širokou škálu vestavěných funkcí s nejrůznějšími uplatněním. Tyto funkce nám nejen ulehčují práci, ale i zlepšují chod celého programu. Ačkoli různé knihovny podporují různé jazyky, mezi zdaleka nejpopulárnější patří jazyk C++ a Python.²⁴ C++ si našel využití hlavně díky své rychlosti, ačkoli s ním není práce jednoduchá. Python je pravý opak, jelikož jeho krása tkví v jednoduchosti a všestrannosti, která přichází na úkor výkonu. Jak jsem už zmínil knihoven je spousta. Zde se zaměřím pouze na jednu, a to tu zdaleka nejpoužívanější – OpenCV.²⁵

OpenCV je open source knihovna pro počítačovou vizi, vyvinutá primárně pro zpracování obrazu a videa. Tato knihovna obsahuje několik stovek algoritmů, které umožňují rozpoznávání předmětů, sledování objektů, detekci obličejů a mnoho dalších úloh v oblasti počítačové vize. OpenCV byla původně vyvinuta v roce 1999 Vladimírem Vapnikem a dalšími autory v Intelu a nyní ji spravuje organizace OpenCV.org.²⁶

²³ Top 8 Algorithms For Object Detection One Must Know. *Analytics India Magazine*

²⁴ How to Learn Computer Vision from Scratch in 2023?. ProjectPro - Solved Big Data and Data Science Projects

²⁵ Top 15 computer vision libraries | SuperAnnotate.

²⁶ About - OpenCV

Knihovna se dá použít v různých programovacích jazycích, ale v současnosti je nejrozšířenější verze pro jazyk Python. Primárně kvůli jednoduchosti syntaxe a velké podpoře ze strany komunity.

OpenCV v Pythonu se často používá pro úpravu obrázků a videí. Jedním z nejpoužívanějších modulů OpenCV v Pythonu je cv2, který umožňuje jejich otevření a zpracování, aplikování různých filtrů, jako jsou rozostření, detekce hran, změnu velikosti, převod barevných prostorů, rotaci a překlápění obrázků a mnoho dalšího.

Kromě toho je OpenCV v Pythonu velmi užitečný pro počítačovou vizualizaci, umožňuje tvoření grafů, kreslení geometrických tvarů a přidávání textu do obrázků a videí. Všechny tyto funkce umožňují snadno vytvořit vizuální reprezentaci dat nebo výsledků zpracování obrazu.²⁷

PRAKTICKÁ ČÁST

5 NÁVRH KONCEPTU

První koncept Trackbotu jsem navrhnul již v červnu 2022. Představoval jsem si, že model bude fungovat na stejném principu jako vstupní brána do středověkého hradu. Brána je dole pevně zachycená a řetězy je uchycena za horní část vrat. Já chtěl fungování upravit.

Řetězy, v mém případě odlišný typ uchycení, by držel model dole, a pevně uchycená část by byla v horní části modelu. Důvod, proč jsem takto přemýšlel je, protože robot se má točit do stran, což jsem nevnímal jako problém – řešení jsem viděl jako otočnou desku, ale výzva byla technologicky zpracovat vertikální naklápění, což by řešila variace zmíněných řetězů.

²⁷ OpenCV - Overview - GeeksforGeeks

Během navrhování konceptu, v tu chvíli pouze náčrtu na papíru, jsem zjistil, že takový model je zbytečně komplikovaný a existuje lepší řešení. Návrh podobný bráně středověkého hradu jsem tedy zavrhl.

Přešel jsem tedy k práci s (v tu dobu pouze jedním) ramenem, otočnou plošinou a servomotory. Co mě k tomu vedlo? Kombinace ramena a servomotoru se mi zdála jako nejjednodušší řešení vertikálního naklápění. Otočná plošina je k horizontálním posunům často využívána i v komerční sféře (otočná plošina, na které je vystavované zboží, točna, na které se balí zboží na paletách apod.) u razantně větších komponent a servomotor, který jsem později nahradil krokovým taktěž. To jsem ovšem z počátku nevěděl.

Z načrtnutého návrhu jsem zkusil vymodelovat prvotní projekt v softwaru Autodesk Inventor, viz obrázek. Šlo o prvotní koncept, který obsahoval rameno, plošinu, 2 servomotory, držák na telefon s telefonem a krabičku, do které umístím kabely a zbylou elektroniku, která zároveň bude sloužit jako podstava a těžiště.

Původně jsem chtěl dát na stranu jedno rameno se servomotorem, na který bych napojil hřídel s držákem, čímž bych vyřešil problém vertikálního naklápění.

Později jsem zjistil, že jedno rameno nemůže být řešení – silový vektor gravitace je na jedno rameno moc velký, a navíc by působil mimo osu modelu, což by vedlo k vyvrácení servomotoru z ramene, a neúměrnému horizontálnímu pohybu. Z těchto důvodů jsem přidal druhé symetrické rameno. I do druhého modelu jsem chtěl vložit samostatný pohon, ale nakonec jsem zjistil, že jeden servomotor je dostačující.

6 PROGRAM PRO SLEDOVÁNÍ UŽIVATELE

S hotovým návrhem modelu jsem se vydal vstříc další části projektu, kterou jsem považoval za zdaleka nejnáročnější část. Potřeboval jsem vytvořit mobilní aplikaci, která bude skrze fotoaparát telefonu vyhledávat a dále sledovat uživatele. S prakticky nulovými znalostmi algoritmicke a počítačového vidění jsem začal vymýšlet, jak by bylo možné program vytvořit. V tu dobu jsem již studoval programovací jazyk C++, snažil jsem se proto najít způsob, jak tyto znalosti využít.

Po chvíli hledání jsem našel knihovnu OpenCV, která slouží pro práci s počítačovým viděním. Přestože byla primárně stavěná pro jazyk Python, umožňovala také práci v C++. Můj plán byl následující – využít vestavěných funkcí knihovny OpenCV k nalezení a sledování uživatele, dále bezdrátově přeposlat jeho souřadnice mikro ovladači Arduino, který by data zpracoval a podle potřeby otočil motory tak, aby byl uživatel vždy vycentrovaný. Zvolil jsem vývojové prostředí Microsoft Visual Studio, se kterým jsem byl již dobře seznámený.

6.1 Programování v C++

Prvním krokem bylo nainstalovat knihovnu OpenCV. Ačkoliv to zní velmi jednoduše, šlo o dosti náročnou výzvu, jelikož OpenCV není nativně podporována ve Visual Studiu a musela

být importována externě. Po dnu zápasení se mi povedlo knihovnu zprovoznit a mohl jsem se pustit do programování. Přešel jsem proto do dokumentace a začal pomalu tvořit mozek celého zařízení.

Ačkoli jsem se asi po dvou týdnech dostal do bodu, kdy mnou vytvořený program dokázal uživatele sledovat, bylo to možné pouze v případě, kdy jsem já manuálně označil uživatele myší. Neprostudoval jsem si totiž, že OpenCV nemá žádnou funkci na automatické vyhledávání objektů. A to nebyl jediný problém. C++ je jazyk, který je velmi striktní na syntax kódu a typ dat, která vstupují do funkcí. Toto v kombinaci s řídkou dokumentací tvořilo obtížné podmínky pro vývoj aplikace. Časem jsem tedy musel přiznat porážku a hledat řešení těchto problémů jinde.

6.2 Algoritmizace a Python

Při větším průzkumu jsem přišel na řešení jednoho z problémů. Rozpoznávání objektů se v praxi řeší použitím speciálních algoritmů, které dokáží přesně rozpoznat škálu různých předmětů a určit jejich polohu. Ačkoliv sice dokáží velmi přesně určit typ a pozici objektů, kladou také velké nároky na hardware. Tato skutečnost představovala nezanedbatelný problém, neboť tento algoritmus spolu s dalšími procesy jsem plánoval aplikovat pro mobilní telefon. Nakonec jsem se i přes jisté obavy rozhodnul algoritmus vyzkoušet.

Existuje mnoho různých algoritmů pro tento typ práce. Já si vybral YOLOv4 – název představuje akronym pro “You Only Look Once”, a “v4” odkazuje na čtvrtou verzi algoritmu. Přestože YOLO dokáže spolehlivě rozpoznat desítky různých objektů, já jsem potřeboval rozpoznat pouze jeden druh objektů – lidi. Algoritmus bylo možné s trochou úprav přenastavit, aby se zaměřoval pouze na zaměřování lidí. Toto mi později výrazně zjednodušilo programování.

Jeden problém jsem sice vyřešil, i přesto mě další stále čekal. Stálo přede mnou rozhodnutí – zkusit program znovu naprogramovat v C++, s využitím znalostí, které jsem načerpal během mých neúspěšných pokusů, anebo zcela opustím jazyk C a přejdu na jazyk, který jsem nikdy předtím nikdy nepoužil – Python.

Po analýzách možnostech použití jsem se rozhodl naučit třetímu jazyku, který pro tento program využiji. Již zmíněná knihovna OpenCV má v jazyce Python rozsáhlou dokumentaci, spolu s mnoha výukovými videi, které mohu použít při vývoji programu. Uživatel nebude program ovládat přes příkazový řádek, ale přes uživatelské rozhraní (dále GUI), pro které má Python také specializovanou knihovnu – Kivy.

Stejně jako Python samotný, tato knihovna je relativně jednoduchá na práci a zároveň je cross-platform. To je naprosto kritické z hlediska funkčnosti celé aplikace, neboť by jinak nemohla fungovat na mobilních zařízeních. Při volbě vývojového prostředí jsem dal přednost PycharmIDE před Visual Studio Code, neboť VS Code není plnohodnotné vývojové prostředí, ale pouze textový editor, což mi zavánělo problémy.

Následujících pár dní jsem strávil studováním Pythonu. Jelikož jsem už znalosti programování měl, jednalo se spíše o učení nové syntaxe kódu, než nových principů a metod fungování. Ke studiu jsem použil volně dostupná videa a dokumentaci. Netrvalo dlouho a Python jsem takzvaně dostal pod kůži a začal pracovat na svém programu.

6.3 Tvorba detekčního algoritmu

Ze všeho nejdříve bylo nutné integrovat YOLO algoritmus do Pythonu. S touto okolností jsem se před tím nikdy nepotkal, neboť to bylo poprvé co jsem pracoval s umělou inteligencí. Rozhodl jsem se proto využít pro integraci algoritmu kód z internetu²⁸.

Tento kód importoval do programu předem stažené váhy – data, na jejichž základě algoritmus posuzuje vstupní informace (v mém případě obrázky) a dll model - set instrukcí obsahující parametry a specifikace, které upřesňují vyhledávání algoritmu. Současně kód obsahoval funkci "Detect", která na vstupním snímku rozpozná všechny, pro algoritmus známé, objekty. Tento kód jsem použil jako základ pro svůj program.

Bohužel, tvorba programu nebyla tak jednoduchá, že by stačilo zkopírovat pár řádků kódu a program by byl na světě. Kód a model samotný jsem musel upravit, aby splňoval mé potřeby. Nejdříve jsem odstranil zapojení grafické akcelerace do chodu programu. Měl jsem obavy, že na telefonu by nefungovala správně a pouze by zkreslovala, reálný chod programu. Dále jsem nastavil, aby YOLO algoritmus mohl detekovat pouze lidi. To mi výrazně zjednodušilo následné programování.

Nyní jsem konečně mohl přejít k programování samotného programu. Založil jsem nový soubor a importoval do něj vzniklý algoritmus společně s knihovnou OpenCV. Pomocí algoritmu jsem detekoval osoby na obrázku, který jsem využíval pro testování.

První testování přineslo problém – algoritmus detekuje několik lidí z fotografie. Já potřeboval, aby byl algoritmus schopen detekovat pouze jediného, který je přímo před fotoaparátem. Při běhu programu může být víc lidí před telefonem, a tudíž bylo nutné, aby se algoritmus zaměřil pouze na jediného.

Řešení jsem vymyslel následovně – algoritmus bude detekovat pouze člověka, který zabírá největší množství plochy na fotografii – osobu, která mojí aplikaci používá. Takto se vyfiltrují osoby, které se mohou nacházet na pozadí. Řešení jsem úspěšně otestoval na několika fotografiích.

Algoritmus mi vedle filtrace osob dokáže poskytnout i souřadnice, kde se nachází osoba na fotografii – tzv. Bounding Box, který má podobu obdélníku. Bez této funkce by trackování uživatele nebylo možné, neboť bude později zapotřebí k inicializaci trackeru.

²⁸ Object Tracking with Opencv and Python | Pysource



Obrázek 7: Příklad užití mého algoritmu

7 KOMUNIKACE MOBILNÍHO ZAŘÍZENÍ A MIKRO-OVLADAČE

Komunikaci mezi zařízením a mikro ovladačem bylo možné vyřešit s pomocí kabelu, šlo by o nejjednodušší řešení, i přesto jsem tuto možnost propojení nezvolil. Kabel by limitoval funkčnost Trackbotu – nebylo by možné, aby se zařízení otáčelo o 360 stupňů, jelikož by kabel musel procházet osou celého zařízení, kde se ovšem již nacházela spojka, která propojovala motor ramena a mikropočítač v dolní části zařízení.

Mým cílem bylo, aby se Trackbot mohl otáčet neustále, dle potřeb uživatele, a proto jsem zvolil Bluetooth přenos. Aplikace telefonu se bezdrátově připojí k Bluetooth modemu, který je napojený na Arduino, které se stará o natáčení motoru.

Bezdrátové připojení bylo teoreticky velmi jednoduché, praxe byla ovšem jiná. Jako první bylo nutné zvýšit propustnost informací v Bluetooth modemu, bez které by modem nestíhal přijímat informace, které by přijímal z telefonu, tedy od uživatele.

Propustnost jsem zvýšil z 9600 baudů na 115 200 baudů v nastavení modemu, do kterého jsem se dostal s pomocí sériové linky Arduina.

Do mobilní aplikace bylo nutné integrovat funkčnost, díky které se uživatel automaticky připojí k Bluetooth modulu a zašle data na propojený mikroprocesor. Zde jsem narazil na problém.

8 ZMĚNA OPERAČNÍHO SYSTÉMU

Během prvního testování se zmíněný postup osvědčil, ovšem ale pouze při zasílání dat z počítače, na které jsem testoval. Nyní bylo nutné provést to stejné z telefonního zařízení. Veškeré programování jsem do této doby dělal v programovacím jazyku Python, kterému chyběla funkční knihovna, se kterou bych replikoval úspěch na počítači.

S pomocí jazyku Python by byla i samotná tvorba uživatelského rozhraní i následné testování nesmírně náročné, a tudíž jsem se po dlouhých úvahách a zvažování možností rozhodl konvertovat k programovacímu jazyku Java a softwaru Android Studio, jenž je vyvíjeno pro tvorbu mobilních aplikací.

Obával jsem se, že vývoj a hotové výstupy v Pythonu přijdou vniveč právě z důvodu přechodu do jiného jazyka. Naštěstí jsem se zmýlil a hotovou práci následně zužitkoval. Využil jsem výše zmíněný algoritmus, který jsem použil v Android Java aplikaci.

S programovacím jazykem Java jsem neměl zkušenosti, čas na dokončení se krátil, a tudíž bylo nutné pracovat precizně. Ze všeho nejdříve jsem si musel osvojit základy Javy a strukturu Android aplikací, abych mohl následně vyřešit problém s propojením Bluetooth.

8.1 Testování Bluetooth

Mým cílem bylo vytvořit jednoduchou Bluetooth aplikaci, kterou bych otestoval spojení mezi modulem a telefonem.

Prvně jsem musel vyřešit oprávnění a bezpečnost. V nejnovějších verzích operačního systému Android se klade velký důraz na bezpečnost a oprávnění – telefon se dříve bez problému připojil např. ke galerii, nyní telefon vyžaduje uživatele oprávnění k přístupu k citlivým datům.

Připojení k Bluetooth je z pohledu operačního systému Android považováno jako potenciální bezpečnostní riziko, k němuž jsem musel získat přístup – oprávnění. Jelikož jsem v této problematice nedisponoval dostatečným znalostmi, vyhledal jsem pomoc Ing. Jaroslava Příbramského, který mi ukázal, jak tyto oprávnění získat.

S dosaženými oprávněními jsem se pustil do testovacího programu, abych ozkoušel jak připojení, tak zasílání dat přes Bluetooth – tedy z telefonu do track robota a zpět. Přenos Bluetooth byl vyřešený a nyní bylo nutné vyřešit samotnou aplikaci.

9 Android aplikace

Bylo nutné vyřešit několik funkcí, ke kterým aplikace měla sloužit. Z pohledu uživatele šlo o:

- Zobrazování videa z přední kamery na displeji telefonu
- Natáčení videa

Uživatel v aplikaci nastaví frame rate (snímkovou frekvenci), rozlišení videa a zvolí možnost, zda chce video z aplikace natáčet. Tyto 3 nastavení nechávám na uživateli, zbytek se nastaví automaticky. Minimalistické nastavení jsem nechal, aby aplikace zůstala uživatelsky přívětivá, jednoduchá na ovládání a použitelná pro každého.

Z pohledu vývojáře se bylo potřeba zaměřit na:

- Vytvoření souboru na disku,
- Připojení telefonu prostřednictvím aplikace k Bluetooth modemu
- Zobrazování snímků na displeji
- Integrace hotového algoritmu z Pythonu do aplikace – nalezení uživatele.
- Sledování uživatele za pomoci sledovacího algoritmu

9.1 Implementace funkcí

Ze všeho nejdříve, tedy před začátkem nahrávání videa uživatele, je nutné vytvořit složku na disku, do které bude video nahráváno. Název souboru se volí dle data pořízení. Následuje přizpůsobení funkcí v aplikaci dle uživatele nastavení. Video se defaultně ukládá do formátu MP4, jenž je považovaný za standart video formátů. Všechny nyní zmíněné procesy probíhají plynule a na pozadí, bez toho, aby uživatele vyrušovali.

Připojení bylo popsáno výše, a proto jen zjednodušeně. Aplikace se připojí k modemu a data, jež jsou odesílána skrze Bluetooth slouží k výpočtu o kolik se mají otočit jednotlivé motory, tak aby byl uživatel vždy vycentrovaný ve snímku.

Aby aplikace plnila svou hlavní funkci, je nutné, aby uživatel vždy viděl na displeji, co vidí čočka fotoaparátu – musí se tedy v reálném čase promítat záběr z kamery. V aplikaci jsem musel vytvořit funkci, jenž požádá senzor telefonu o nový snímek, který jej ukáže na displeji – na *textureview*.

Tato funkce probíhá 30krát za sekundu, jedná se o ekvivalent 30 snímků za sekundu na displeji videa. Funkce současně slouží k zprostředkování jednotlivých snímků ostatním částím aplikace – detekování, trasování.

Co se týče integrace algoritmus do aplikace, nabízely se dvě možnosti – přepsat hotové procesy do Javy anebo integrovat Python algoritmus do aplikace. Vybral jsem si možnost druhou.

Udělal jsem několik rešerší, při kterých jsem hledal knihovny, které by mi umožnily přenést hotový algoritmus do Javy. Našel jsem několik potenciálních kandidátů, mezi které patří např. Py4J – Python for Java a další. Nakonec jsem zvolil knihovnu se jménem Chaquopy. Knihovnu jsem zvolil, protože byla jednoduchá na spuštění, práci s ní a relativně rychlá. Svůj hotový algoritmus jsem vložil do knihovnou vygenerované složky a následně ji spustil ve svém programu. Výstup tohoto procesu byl Bounding Box, který reprezentoval polohu uživatele na snímku.

Jak bylo již zmíněno výše, algoritmus je přesný, dokáže detekovat umístění uživatele i za zhoršených světelných podmínek, a právě proto je náročný na výkon. Tuto náročnost jsem vyřešil vytvořením tzv. trackeru, což je funkce knihovny OpenCV. Tento tracker je méně náročný na výkon, díky čemuž dokáže zvládat nároky na hardware, jimiž disponuje běžný uživatel. Bez tohoto trackeru by algoritmus nebyl schopen nalézt polohu uživatele alespoň 15krát za sekundu.

Po inicializaci trackeru program načte snímek z kamery, pomocí kterého tracker aktualizuje, který vrátí polohu uživatele zpátky programu, a ten pošle hodnoty přes Bluetooth do Arduina, kde se s výpočty dále pracuje. Celý tento proces proběhne 15krát za sekundu.

Nutno podotknout že jsem při tvorbě aplikace využil online video sérii²⁹ se zaměřením na Android Camera2 API, která mi pomohla vytvořit jádro aplikace a provést některé nastavení programu.

10 GRAFICKÉ UŽIVATELSKÉ ROZHRAŇÍ – TRACKBOTUI

V průběhu vypracovávání Trackbotu jsem zjistil, že původní plán, ve kterém jsem uživatelské rozhraní chtěl vytvořit v Pythonu a za pomoci knihovny Kivy nebyl možný – z Pythonu jsem přešel na Javu. Uživatelské rozhraní bylo sice již hotové, i přesto jsem ho musel nahradit novým řešením, jenž odpovídalo aktuální situaci.

Z toho důvodu vzniklo uživatelské rozhraní v XML za pomoci zabudovaných funkcí v Android Studiu. Konečné uživatelské prostředí jsem navrhl s důrazem na minimalismus. Na displeji v aplikaci se uživateli ukazuje hlavní tlačítko v dolní části, jenž slouží ke spuštění hlavní funkce programu. Druhé tlačítko, které umožňuje uživateli upravovat nastavení dle vlastních preferencí – např. snímkovou frekvenci nebo kvalitu videa. Ke tlačítkům jsem přidal plynulé animace pro zlepšení uživatelského zážitku.

Nastavení jsem následně integroval do hlavního programu.

11 FINÁLNÍ MODEL A JEHO TISK

Výše jsem popsal celý model i jeho specifika, a tudíž se v této kapitole zaměřím na malé změny modelu, které během tvorby vznikly a vzhled finálního produktu.

Nejmarkantnější rozdíl byla výměna servo pohonu, jenž slouží k otáčení plošiny s telefonem po ose x, tedy horizontálně. Servo jsem nahradil krokovým motorem, protože bylo nutné, aby se plošina dokázala otáčet o 360 stupňů, což by mi servo neumožnilo, jelikož jeho maximální rozsah je defaultně 180 stupňů. Ačkoliv je možné servo motor upravit k otáčení o 360 stupňů, nedostačovalo nárokům mého zadání.

Spolu s výměnou motoru bylo nezbytné upravit součástky. Krokový motor se od servo motoru liší nejen výkonem a funkcionalitami, ale stejně tak tvarem. Tudíž bylo nutné změnit model,

²⁹ Android Camera2 API Video App

ve kterém bude motor uložený. Doslovně popsáno jsem z hranatého tvaru serva přeměnil na kulatý tvar krokového pohonu.

Co se týče tisku, bylo potřeba každou součást exportovat ve formátu STL, aby mohly být vytisknuty, a tedy exportovány do Prusha sliceru, kde jsem nastavil 3D tiskárnu pro tisk. Nastavoval jsem teplotu, které vyhovovala materiálu, která vyhovuje materiálu, který jsem využíval pro tisk – PLA materiál. Dále jsem nastavil typ a míru tzv. infillu – výplň materiálu.

Jako poslední jsem nastavil parametry, které zabránily deformaci materiálu při tisku. Po úspěšném nastavení jsem tiskl jednu součást po druhé. Doba tisku dosáhla více než 30 h.

12 ZHOTOVENÍ A TESTOVÁNÍ PROTOTYPU

Vytisklé součástky bylo nutné sestavit dohromady, aby vytvořily celistvý model. Během modelování jsem dopředu vytvořil místa, ve kterých se jednotlivé části spojí např. s pomocí šroubků a maticek.

I přes tuto přípravu jsem musel do součástek zasáhnout manuálně. Během tisku vzniklo několik nepřesností, které vyústily v nemožnost součástky spojit dohromady dle předem stanoveného plánu v předem vymodelovaných spojnicích. A proto jsem je musel nahradit běžně využívanými šrouby a spojit model do sebe.

Dále bylo potřeba přilepit k sobě několik malých částí jako – zarážka u dolního krokového motoru, aby se zabránilo jeho otáčení do stran.

Testování probíhalo následovně – oba motory, jež byly namontovány ve finálním modelu, jsem připojil k Arduinu do kterého jsem nahrál jednoduchý program, abych naprogramovanou aplikaci otestoval s reálnými součástky. Tímto jsem zjistil, že dolní krokový motor funguje bez problému, a navíc se dobře ovládá. Za to motor v rameni nebyl nejlepší volba, jelikož po umístění telefonu do držáků zařízení a spuštění serva se celá horní část zařízení začala chvět. Důvod byl nevycentrované těžiště telefonu, což vytvářelo rozklepaný záběr.

Tuto skutečnost jsem se pokusil vytvořit softwarově, úpravou programu, což se mi do určité míry povedlo, ale bohužel záchvěvy neodstranilo kompletně.

13 DODATEČNÉ ÚPRAVY

Po sestavení celého modelu, dokončení funkcí programu a testování všech částí projektu současně jsem narazil na zásadní problém. Pokaždé kdy se uživatel pohne na ose y, se musí servo přiměřeně natočit. Tento pohyb však servo motor nemůže udělat najednou, jelikož by se mohlo stát, že by váha telefonu při zastavení utrhla hřídel, na které je držák připevněn.

Jediný způsob, jak tomuto zabránit je rozdělit jeden velký krok na několik menších. Toto má sice za následek viditelné sekání při natáčení, ale zamezí to možnému poškození zařízení. Abych tohoto docílil, nastavil jsem krok serva na jednu čtvrtinu stupně, přičemž se motor bude natáčet, dokud nedojede do konečné polohy.

Zde nastává primární problém tohoto řešení. Mezi každým krokem musí být pauza alespoň 15 milisekund, aby nedošlo k poškození serva. To se na první pohled nemusí zdát jako problém, ale pokud chci zajistit co možná nejpřesnější sledování, nesmí doba natáčení motorů trvat déle jak 50 milisekund. Kdyby jejich pohyb trval déle jak 50 milisekund, nedokázal bych zajistit nejpřesnější sledování, neboť by natáčení motorů trvalo déle jak aktualizování trackeru.

Došel jsem tedy k názoru, že nejjednodušší cesta, jak tento problém vyřešit je použít stejný motor jak pro osu x, tak osu y. To by znamenalo přemodelovat jedno rameno, přidělat část na uložení baterie a upravit obvod. Po dokončení těchto úprav by zařízení bylo kompletně funkční, neboť program je už hotový.

Bohužel z časových důvodů nejsem schopen tyto úpravy zrealizovat.

ZÁVĚR

Svou práci za poslední rok, hodnotím velmi pozitivně. Finální zařízení není perfektní, ale když vezmu v potaz celkovou komplexnost zadání, časovou náročnost, rozpočet, a především skutečnost toho, že jsem celou problematiku řešil od nuly bez jakékoli předlohy, tak jsem více než spokojen.

Během práce jsem nabyl mnohých zkušeností, naučil jsem se několik programovacích jazyků, v podstatě z ničeho. Zdokonalil jsem se v práci na dlouhodobých projektech a na práci chci navázat v příštích letech – trackbot chci zdokonalovat a vylepšovat.

Doufám, že má práce inspiruje další k podobným výtvorům.

POUŽITÁ LITERATURA A ZDROJE

- Global 3D printing market size 2013-2021 | Statista. Statista - The Statistics Portal for Market Data, Market Research and Market Studies [online]. Copyright © Statista 2022 [cit. 25.12.2022]. Dostupné z: <https://www.statista.com/statistics/796237/worldwide-forecast-growth-3d-printing-market/>
- WAYKEN, Rapid. 5 industrial applications of 3D printing services. In: *Geospatialworld* [online]. 15. 6. 2021 [cit. 2023-01-04]. Dostupné z: <https://www.geospatialworld.net/article/5-industrial-applications-of-3d-printing-services/>
- [FAQ] Co je to 3D model? | 3Dees. *3D tisk pro průmysl – kompletní řešení aditivní výroby* | 3Dees [online]. Dostupné z: <https://www.3dees.cz/faq/89-co-je-3d-model>
- Josef Průša (vývojář) – Wikipedie. [online]. Dostupné z: [https://cs.wikipedia.org/wiki/Josef_Pr%C5%AF%C5%A1a_\(v%C3%BDvoj%C3%A1%C5%99\)](https://cs.wikipedia.org/wiki/Josef_Pr%C5%AF%C5%A1a_(v%C3%BDvoj%C3%A1%C5%99))
- Jak funguje 3D tiskárna. 3D Tiskárna - Abc3D nejlepší pro 3D tisk [online]. Copyright © 2014 [cit. 16.01.2023]. Dostupné z: <https://www.abc3d.cz/o-3d-tisku-neprehlednete/3d-tisk-principy-technologie>
- 3D tisk: Podle čeho a jaký filament vybrat?. Alza.cz [online]. 2019 [cit. 2023-01-16]. Dostupné z: <https://www.alza.cz/filamenty-3d-tisk>
- 3D Printing with FDM - YouTube. YouTube [online]. Copyright © 2023 Google LLC [cit. 16.01.2023]. Dostupné z: https://www.youtube.com/watch?v=rkUILE3zv98&ab_channel=StratasysDirectManufacturing
- Ekonomické zhodnocení FDM technologie 3D tisku v elektrotechnickém průmyslu. ČVUT DSpace [online]. Copyright © [cit. 16.01.2023]. Dostupné z: <https://dspace.cvut.cz/bitstream/handle/10467/82758/F3-BP-2019-Behal-Daniel-Bakalarska%20prace%20Daniel%20Behal.pdf?sequence=-1&isAllowed=y>
- Rozdíl mezi ABS, PLA a PETG. *Materiály a doplňky pro 3D tisk - filamenty - Materialpro3d.cz* [online]. Dostupné z: <https://www.materialpro3d.cz/blog/rozdily-abs-pla-petg/>
- ABS | Prusa Knowledge Base. *Prusa Knowledge Base* [online]. Copyright © Prusa Research a.s. [cit. 18.01.2023]. Dostupné z: https://help.prusa3d.com/cs/article/abs_2058
- Fast-forward — comparing a 1980s supercomputer to a modern smartphone | Adobe Blog. *Welcome to the Adobe Blog* [online]. Dostupné z: <https://blog.adobe.com/en/publish/2022/11/08/fast-forward-comparing-1980s-supercomputer-to-modern-smartphone>
- What size smartphone do you need?. In: *Coolblue* [online]. 2022 [cit. 2023-01-18]. Dostupné z: <https://www.coolblue.be/en/advice/smartphone-screens.html>
- Moravec's Paradox: Definition, Explanation and Examples. *Science ABC » Science Explained In Simple Words* [online]. Copyright © 2023. [cit. 18.01.2023]. Dostupné z: <https://www.scienceabc.com/innovation/what-is-moravecs-paradox-definition.html>

- How to Learn Computer Vision from Scratch in 2023?. ProjectPro - Solved Big Data and Data Science Projects [online]. Copyright © 2023 Iconiq Inc. [cit. 19.01.2023]. Dostupné z: <https://www.projectpro.io/article/how-to-learn-computer-vision/515>
- Top 15 computer vision libraries | SuperAnnotate. The ultimate training data platform for AI | SuperAnnotate [online]. Copyright © 2023 SuperAnnotate AI, Inc. All rights reserved. [cit. 19.01.2023]. Dostupné z: <https://www.superannotate.com/blog/computer-vision-libraries>
- Object Tracking with Opencv and Python | Pysource [online]. Dostupné z: <https://youtu.be/O3b8lVF93jU>
- Android Camera2 API Video App - Part 1 How to add icons using android studio - YouTube. *YouTube* [online]. Copyright © 2023 Google LLC [cit. 23.03.2023]. Dostupné z: <https://www.youtube.com/watch?v=CuvVpsFc77w&list=PL9jCwTXYWjDIHNEGtsRdCTk79I9-95TbI>
- Getting Started with YOLO v4 | *MathWorks - Makers of MATLAB and Simulink - MATLAB & Simulink* [online]. Copyright © 1994 [cit. 23.03.2023]. Dostupné z: <https://www.mathworks.com/help/vision/ug/getting-started-with-yolo-v4.html>
- YOLOv4 - An explanation of how it works. *Roboflow Blog* [online]. Copyright © [cit. 23.03.2023]. Dostupné z: <https://blog.roboflow.com/a-thorough-breakdown-of-yolov4/>
- Top 8 Algorithms For Object Detection One Must Know. *Analytics India Magazine | Artificial Intelligence, Data Science, Machine Learning* [online]. Copyright © Analytics India Magazine Pvt Ltd [cit. 23.03.2023]. Dostupné z: <https://analyticsindiamag.com/top-8-algorithms-for-object-detection/>
- Co je to Arduino?. *Úvodní stránka - Bastlírna HWKITCHEN* [online]. Dostupné z: <https://bastlirna.hwkitchen.cz/co-je-to-arduino/>
- Počítačové vidění – Wikipedie. [online]. Dostupné z: https://cs.wikipedia.org/wiki/Po%C4%8D%C3%ADta%C4%8Dov%C3%A9_vid%C4%9Bn%C3%AD
- About - OpenCV. *Home - OpenCV* [online]. Dostupné z: <https://opencv.org/about/>
- OpenCV - Overview - GeeksforGeeks. *GeeksforGeeks | A computer science portal for geeks* [online]. Dostupné z: <https://www.geeksforgeeks.org/opencv-overview/>

PŘÍLOHY

- 1. Součásti modelu**
- 2. Program trackbotu**
- 3. Kód Arduina**