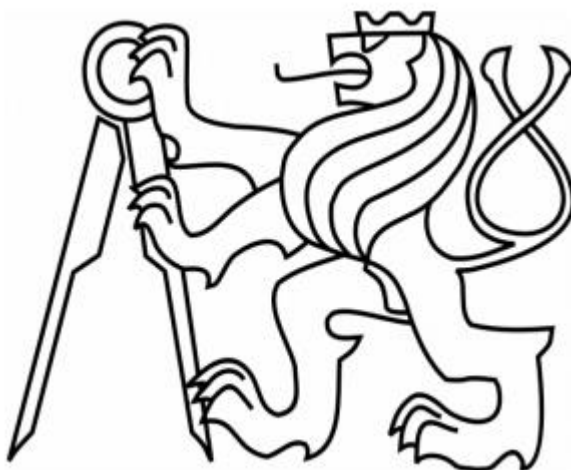


**ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA STROJNÍ**

Ústav přístrojové a řídicí techniky



3D SCANNER

**BAKALÁŘSKÁ PRÁCE
2023**

VALERII LISNYCHUK

Studijní program:

Strojírenství

Studijní obor:

Informační a automatizační technika

Vedoucí práce:

doc. Ing. Martin Novák, Ph.D

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Lisnychuk** Jméno: **Valerii** Osobní číslo: **492560**
Fakulta/ústav: **Fakulta strojní**
Zadávající katedra/ústav: **Ústav přístrojové a řídicí techniky**
Studijní program: **Strojirenství**
Studijní obor: **Informační a automatizační technika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

3D scanner

Název bakalářské práce anglicky:

3D scanner

Pokyny pro vypracování:

Cílem práce je vytvoření 3D scanneru na skenování malých objektů (přibližně 20x20x20) a převod do zpracovatelného souboru.

Úkoly:

- 1) rešerše principu, výběr vhodných komponent (laser, kamera, motor)
- 2) mechanický návrh a montáž
- 3) vytvoření programu pro Matlab, který bude signály zpracovávat. Výstup bude soubor
- 4) experimentální ověření na několika objektech

Seznam doporučené literatury:

An Introduction to 3D Scanning, dostupné online na
<https://www.ems-usa.com/tech-papers/An_Introduction_to_3D_Scanning_E-Book.pdf>, přístup 12.12.2017
3D Laser Scanning for Heritage, dostupné online na
<<https://historicengland.org.uk/images-books/publications/3d-laser-scanning-heritage2/>>, přístup 12.12.2017

Jméno a pracoviště vedoucí(ho) bakalářské práce:

doc. Ing. Martin Novák, Ph.D. odbor elektrotechniky FS

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **28.04.2023** Termín odevzdání bakalářské práce: **15.08.2023**

Platnost zadání bakalářské práce: _____

doc. Ing. Martin Novák, Ph.D.
podpis vedoucí(ho) práce

prof. Ing. Tomáš Vyhliďal, Ph.D.
podpis vedoucí(ho) ústavu/katedry

doc. Ing. Miroslav Španiel, CSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací.
Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně s využitím zdrojů, které jsou všechny v práci uvedeny.

Datum

Podpis

Poděkování

Rád bych poděkovat panu doc. Ing. Martinu Novákovi, Ph.D za rady, připomínky a pomoc při zpracování této bakalářské práce. Děkuji také své rodině za jejich neustálou podporu během celého studia.

Abstrakt

Tato bakalářská práce se zabývá vytvořením stolního bezkontaktního 3D skeneru a řídicího programu v MATLAB pro skenování malých objektů. V teoretické části se popisují základní metody 3D skenování a jejich vlastnosti. Podrobněji je věnována technologie laserové triangulace, protože tato metoda se používá v zařízení. Praktická část obsahuje návrh mechaniky a montáž komponent skeneru a také realizaci řídicího programu na zpracování obrazů z webkamery. Na konci práce jsou uvedeny výsledky experimentálního ověření navrženého 3D skeneru na několika objektech.

Klíčová slova

3D skener, laserová triangulace, laserový modul vykreslující čáru, webkamera, otočný stůl, MATLAB, Arduino, řízení.

Abstract

This bachelor thesis deals with the creation of a 3D scanner and control program in MATLAB for scanning small objects. The theoretical part describes the basic methods of 3D scanning and their properties. The laser triangulation technology is discussed in detail, as this method is used in the device. The practical part includes the design of the mechanics and assembly of the scanner components and the implementation of the control program for processing images from the webcam. At the end of the thesis, the results of experimental verification of the designed 3D scanner on several objects are presented.

Key words

3D scanner, laser triangulation, line laser module, webcam, rotary table, MATLAB, Arduino, control.

Obsah

Seznam použitých zkratk a symbolů	1
1. Úvod.....	2
2. Teoretická část.....	3
2.1 Definice a princip činnosti 3D skeneru.....	3
2.2 Přehled technologií 3D skenování	3
2.2.1 Rozdělení technologií	3
2.2.2 Kritéria a parametry při výběru 3D skeneru.....	5
2.2.3 Vlastnosti jednotlivých technologií.....	6
2.3 Technologie laserové triangulace	7
2.3.1 Triangulace	7
2.3.2 Obecná geometrie 3D skeneru.....	7
2.3.3 Zvolená geometrie 3D skeneru.....	9
2.4 Matematický model kamery	10
2.4.1 Model camera obscura.....	10
2.4.2 Zkreslení objektivu.....	13
3. Praktická část.....	14
3.1 Vstupní požadavky	14
3.2 Výběr základních komponent	15
3.2.1 USB webkamera.....	15
3.2.2 Zdroj laserového záření	15
3.2.3 Krokový motor a driver krokového motoru	16
3.2.4 Napájení.....	16
3.3 Návrh konstrukce	17
3.3.1 Konstrukce skenovací hlavy kamera-laser	17
3.3.2 Návrh převodovky	18
3.3.3 Dynamika otočného stolku	20
3.3.4 Konstrukce pohonu.....	24
3.3.5 Celková konstrukce 3D skeneru	25
3.4 Kalibrace webkamery	26
3.5 Kalibrace otočného stolku	28
3.6 Zpracování obrazu a určení polohy laserové čáry	30
3.6.1 Přehled algoritmů z podobných projektů	30
3.6.2 Zkoušení cizích algoritmů	32

3.6.3	Zkoušení cizích algoritmů s červeným filtrem	39
3.6.4	Návrh a zkoušení vlastního algoritmu	43
3.7	Návrh řízení	45
3.7.1	Řízení krokového motoru a laserového modulu.....	45
3.7.2	Komunikace mezi počítačem a Arduino	46
3.7.3	Výsledný program	47
3.8	Experimentální ověření 3D skeneru.....	49
4.	Závěr.....	50
	Seznam použité literatury	52
	Přílohy	54

Seznam použitých zkratek a symbolů

Symbol	Jednotka	Význam
J_z	$[kg \cdot mm^2]$	moment setrvačnosti zátěže
J_{km}	$[kg \cdot mm^2]$	moment setrvačnosti rotoru krokového motoru
J_s	$[kg \cdot mm^2]$	moment setrvačnosti otočného stolku
J_{so}	$[kg \cdot mm^2]$	nejhorší moment setrvačnosti skenovaného objektu
J_{kolo1}	$[kg \cdot mm^2]$	moment setrvačnosti prvního kola/šneku
J_{kolo2}	$[kg \cdot mm^2]$	moment setrvačnosti druhého kola
i	$[-]$	převodový poměr
m	$[mm]$	modul
d	$[mm]$	roztečný průměr soukolí
M_k	$[N \cdot mm]$	krouticí moment
M_z	$[N \cdot mm]$	zatěžovací moment vůči stolu
M'_z	$[N \cdot mm]$	zatěžovací moment vůči rotoru
M_{zr}	$[N \cdot mm]$	moment zrychlení
M_{max}	$[N \cdot mm]$	špičkový točivý moment
M_{km}	$[N \cdot mm]$	točivý moment potřebný pro krokový motor
ε	$[rad/s^2]$	úhlové zrychlení
m_{so}	$[kg]$	maximální hmotnost skenovaného objektu
m_s	$[kg]$	hmotnost otočného stolku
g	$[m/s^2]$	gravitační zrychlení
G	$[N]$	tíhová síla
N	$[N]$	normálová síla
F_t	$[N]$	třecí síla
μ	$[-]$	součinitel tření v ložisku
D	$[m]$	průměr mezi středy kuliček ložiska
t_0	$[s]$	doba polohování
t_1	$[s]$	doba zrychlení
Θ_s	$[^\circ]$	úhel pootočení stolku
Θ_{km}	$[^\circ]$	jmenovitý úhel krokového motoru
mk	$[-]$	velikost mikrokroku
A	$[-]$	počet pulzů
f_{km}	$[Hz]$	pracovní frekvence krokového motoru
n_{km}	$[ot/min]$	otáčky krokového motoru
ω	$[rad/s]$	úhlová rychlost krokového motoru
ε_{km}	$[rad/s^2]$	úhlové zrychlení krokového motoru
η	$[-]$	účinnost převodovky
k	$[-]$	bezpečnostní koeficient

1. Úvod

S rozvojem automatizace a digitalizace výrobních procesů se stává technologie 3D skenování důležitým nástrojem v mnohých průmyslových oblastech a také v lidském životě. 3D skenery umožňují inženýrům, technologům, designérům a dalším rychle, efektivně a relativně přesně vytvářet digitální modely reálných objektů a začleňovat je do výrobních procesů a projektů. Díky tomu se zkracuje čas a zlepšuje se efektivita výroby. Hlavní aplikace 3D skeneru jsou:

- vytváření a úpravy CAD modelů reálných objektů (Obr. 1)
- 3D tisk náhradních dílů v průmyslu a v domácnosti
- reverzní inženýrství součástí (např. získání chybějící výkresů starých součástí)
- digitální dokumentace a archivace (např. kulturních artefaktů)
- kontrola kvality výrobku porovnáním jeho naskenované geometrie s referenčním CAD modelem
- detekce opotřebení a deformace, údržba výrobku
- návrh a výroba zakázkových dílů a výrobků v lékařství, stomatologii (protézy, ortézy, implantáty atd.)
- skenování budov, průmyslových areálů a dalších velkých prostorů
- filmový a herní průmysl, design, umění

Dalším příkladem je použití 3D skeneru s 3D tiskárnou, která naopak vytváří fyzický objekt na základě připraveného modelu. Kombinace těchto dvou zařízení vytváří výkonný digitální pracovní postup, který umožňuje navrhovat a vytvářet zcela nové nebo modifikované výrobky rozšířením modelů CAD o naskenované 3D modely.

Tato práce popisuje vytvoření 3D skeneru a řídicího programu s následujícími požadavky:

- bezkontaktní skenování malých objektů (přibližně 200x200x200 mm)
- možnost skenování těžkých objektů (do 5 kg)
- výstupem je soubor
- jednoduchá montáž a demontáž komponent
- nastavitelnost jednotlivých částí soustavy
- většina dílů se tiskne na 3D tiskárně
- předpokládaná přesnost ± 1 mm
- řídicí program v programovacím prostředí MATLAB



Obr. 1 – Příklad aplikace ručního 3D skeneru [1]

2. Teoretická část

2.1 Definice a princip činnosti 3D skeneru

3D skener je zařízení, které sbírá trojrozměrnou informaci o tvaru a pokud možno i vzhledu objektu nebo okolního prostředí a na základě měření vytváří jeho digitální kopii.

Skenování probíhá pomocí optických, laserových nebo jiných senzorů, které snímají povrch objektu a zachycují jeho geometrii. K rekonstrukci celého objektu může být zapotřebí desítky skenů z různých stran, které musí být převedeny do společného souřadnicového systému a poté slučovány. Výsledkem skenování je mrak bodů (body v souřadnicích X, Y, Z tvořící povrch objektu), které je možné dále převést na polygonální síť (povrch z trojúhelníků, tzv. STL formát). Pro další zpracování a úpravy výstupních dat je ale potřeba speciální software, jako například MeshLab.

2.2 Přehled technologií 3D skenování

2.2.1 Rozdělení technologií

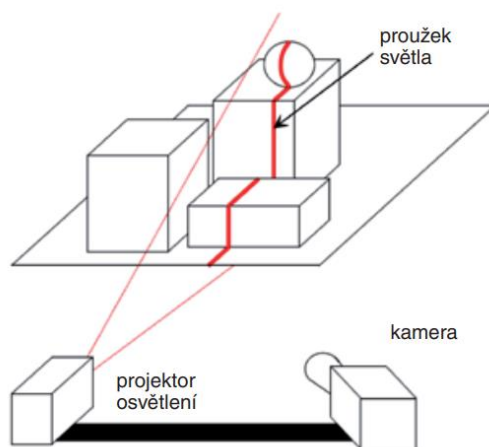
Existují různé způsoby 3D skenování v závislosti na druhu zařízení, použité technologie a také na vlastnostech objektu, který chceme naskenovat, například velký/malý, pohyblivý/statický. Podle [[2];[3]; [4]; [5]] se 3D skenery v zásadě dělí na kontaktní a bezkontaktní skenery, které lze dále rozdělit na aktivní a pasivní.

Kontaktní 3D skenery mají dotykovou sondu se speciálním senzorem, která se pohybuje po objektu a provádí měření rozměrů. Zkoumaný objekt je umístěn na speciálním povrchu a je fixován, aby se nehýbal. Těsný fyzický kontakt umožňuje co nejpřesněji postavit mrak bodů a následně sestavit 3D model objektu. Výhody tohoto typu skeneru jsou možnost skenování průhledných a reflexních povrchů, což neumožňují další metody a velká přesnost (řádově 1 mikron). Nevýhody jsou nízká rychlost, riziko poškození povrchu, velké rozměry, cena a omezení na skenování tvarově složitých ploch [2]. Příklady kontaktních skenerů jsou souřadnicové měřicí stroje a měřicí ramena.

Bezkontaktní 3D skenery, jak už z názvu vyplývá, jsou schopné provádět měření na dálku, a právě tento typ skenerů je na trhu nejvíce zastoupen. Oproti klasickým dotykovým skenerům jsou rychlejší, lépe snímají tvarově složité objekty a jsou ideální pro komplexní analýzu měřené součásti [2]. Mají však omezení na skenování některých povrchů a detailních prvků na povrchu objektu.

- *Aktivní bezkontaktní* typy obsahují zdroj záření (světlo, IR, laser), které je nasměrováno na snímáný objekt. Následně snímač detekuje odraz tohoto záření a v závislosti na principu činnosti stanoví tvar objektu. Do tohoto typu spadají následující technologie:
 - Laserová triangulace – skener promítá jednu nebo více laserových čar na snímáný objekt a poté zachycuje jejich odraz kamerou. Tyto 3D laserové skenery používají triangulační princip a k určení tvaru tělesa jsou známy relativní poloha zdroje laserové záření a kamery (Obr. 2) a úhel odrazu laserového paprsku od objektu, který se dopočítá ze zpracování obrazu z kamery.

· Strukturované světlo – také používají triangulaci, ale místo laserového zdroje tyto 3D skenery obsahují projektor, který na objekt promítá světelné vzory různých tvarů. Pomocí jedné nebo více kamer je osvětlený objekt snímán z více stran a ze zdeformovaných vzorů se stanoví mrak bodů. Výhodou tohoto typu je skenování celého zorného pole najednou, namísto jednoho bodu nebo čáry jako u skenerů s laserovou triangulací.



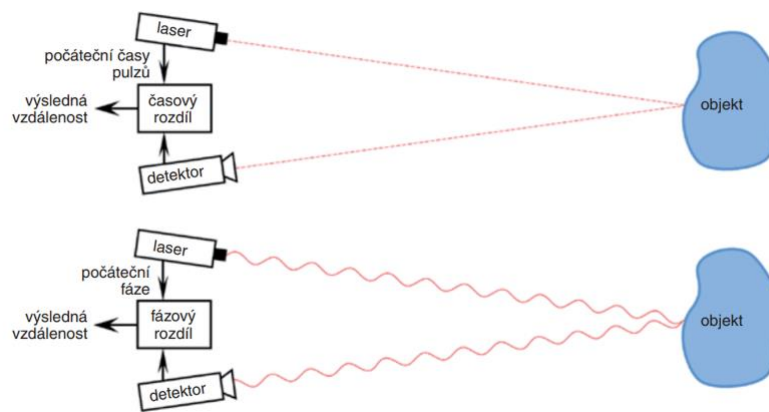
Obr. 2 – Základní konfigurace 3D skeneru na triangulačním principu [2]

· Time-of-Flight (také pulzní skenery) – fungují na principu výpočtu doby průchodu laserového pulzu od zdroje (skeneru) do bodu povrchu snímaného objektu a zpátky (Obr. 3). Protože přesně víme rychlost světla, tak po detekování snímačem zpětného impulsu systém vypočítá vzdálenost do tělesa. Pro přesné určení času se využívají obvody s pikosekundovou přesností. Tyto 3D skenery snímají své zorné pole jeden bod po druhém a pro změnu směru se obvykle používá systém zrcadel.

· Phase shift (fázový posun) – fungují stejně jako skenery doby letu ale navíc systém moduluje výkon laserového paprsku. Porovnáním fází vysílaného a odraženého pulzu lze určit vzdálenost do bodu na povrchu objektu (Obr. 3).

3D skenery pulzní a s fázovým posuvem jsou široce použitelné při vytváření 3D modelů různých budov a konstrukcí ale nejsou vhodné na krátké vzdálenosti, protože v takových případech je doba odezvy velmi krátká a přesnost měření je snížena [3].

- *Pasivní bezkontaktní* typy naopak nemají zdroj záření a spoléhají na odrazení okolního světla. Nejznámější metodou je fotogrammetrie – jednoduchá technologie, která sešívá překrývající se fotografie objektu z různých úhlů pohledu. K pořizování fotografií stačí kamera nebo dokonce smartphone, zatímco spojování probíhá ve speciálním softwaru. Program pomocí algoritmů identifikuje pixely se stejnými místy, odpovídajícím způsobem spojuje obraz a následně jej transformuje do 3D modelu [4].

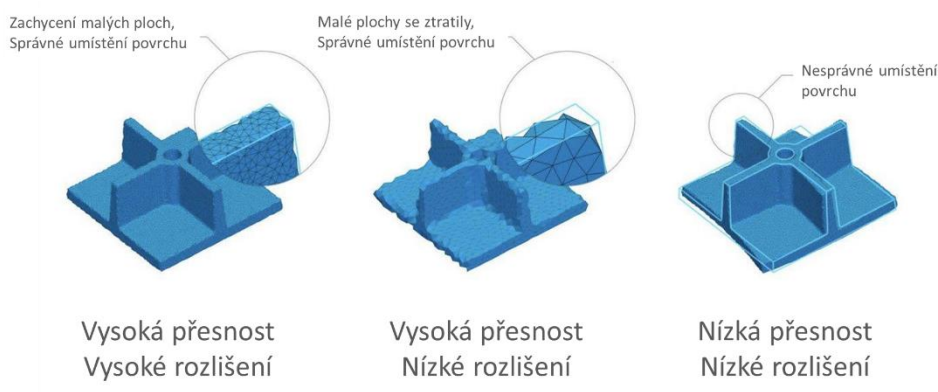


Obr. 3 – Znáznornění pulzního skeneru (nahore) a skeneru s fázovým posuvem (dole) [2]

2.2.2 Kritéria a parametry při výběru 3D skeneru

Na trhu existuje široká škála 3D skenerů a někdy může být obtížné vybrat ten nejlepší pro konkrétní aplikaci. Proto uvádím důležité parametry, které pomohou při výběru 3D skeneru a při porovnávání jednotlivých skenovacích technologií.

- Účel použití 3D skeneru (reverzní inženýrství, kontrola součástí, archivace atd.)
- Přesnost – chyba mezi naměřenými body a jejich skutečným umístěním. Pro stacionární skenery (stolní) je hodnota přesnosti konstantní v rámci zorného pole, zatímco u ručních skeneru závisí na vzdálenosti k objektu
- Rozlišení – vzdálenost mezi zachycenými body, tzn. jaký nejmenší rozměr objektu dokáže 3D skener rozlišit. Čím je hustota změřených bodů vyšší, tím drobnější detaily je systém schopen zachytit (Obr. 4)



Obr. 4 – Vliv hodnot přesnosti a rozlišení na výsledek skenování, přeloženo z [5]

- Doba skenování – rychlost sběru, přenosu a zpracování dat
- Rozměry snímaného objektu
- Skenovací prostředí – osvětlení a pracovní podmínky (např. vlhkost, teplota, vibrace atd)
- Schopnost skenování barvy

- Pracovní rozsah – jak blízko/daleko může objekt být
- Oblast skenování – pokrytí 3D skeneru v jednom záběru
- Omezení – tvar tělesa (hluboké dutiny, žebry, ostré hrany), textura povrchu (lesklá, průhledná, tmavá)
- Cena – závisí na použité technologii, parametrech, snadnosti použití, značce atd.

2.2.3 Vlastnosti jednotlivých technologií

Tabulka 1: Vlastnosti technologií 3D skenování

Technologie 3D skenování	Výhody	Nevýhody
Kontaktní	- skenování průhledných a reflexních povrchů - velká přesnost až 0,001 mm	- malá rychlost a dlouhý čas skenování - riziko poškození povrchu - cena
Laserová triangulace	- jednodušší na použití - skenování nepřipravených objektů (nanášení speciálních prostředků nebo lepicích markerů) - méně citlivé na okolní prostředí a osvětlení - cena	- méně přesné a mají nižší rozlišení než strukturované světlo - pouze statické objekty - nebezpečné pro oči (zaleží na třídě laseru) - omezená snímací vzdálenost (do 1 m) a rozměry objektu (10 – 50 cm) - většinou nepřenosné
Strukturované světlo	- přesnější (~0,1 mm) a lepší rozlišení (16 μm) - rychlé skenování celého zorného pole najednou (až 1,2 m za 2 s) - snímání pohybujících se objektů - bezpečné pro oči - většinou přenosné, nejběžnější technologie v ručních skenerech	- zvláštní pracovní podmínky a osvětlení - příprava objektu před skenováním (nanášení speciálních prostředků nebo lepicích markerů)
Time-of-flight	- skenování velkých objektů (např. výrobní haly, budovy)	- šumy na okrajích objektů - méně přesné než triangulace a strukturované světlo (~ 1 mm)
Phase shift	- měření na dlouhé vzdálenosti (až 300 – 1000 m)	- potřebují systémy s velkou časovou přesností (~ pikosekundy)
Fotogrammetrie	- levné - jednoduché - přenosné (každý současný smartphone umožňuje fotogrammetrii) - menší citlivost na problematické povrchy	- citlivé na kvalitu fotografie - nejméně přesné (ale závisí na softwaru a počtu pixelů) - práce ve speciálním softwaru - dlouhá doba zpracování obrazů (hodiny)

Zdroj: Vytvořeno na základě popisu vlastností jednotlivých metod skenování ze zdrojů [[2]; [3]; [4]; [5]]

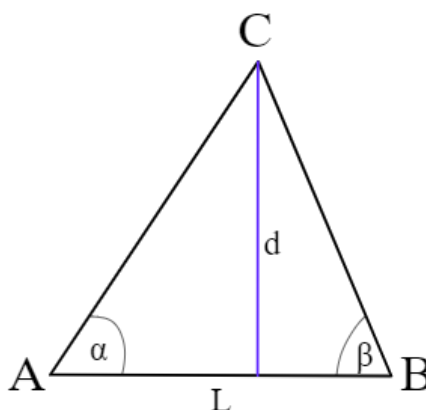
2.3 Technologie laserové triangulace

Z probraných technologií 3D skenování jsem zvolil laserovou triangulaci pro použití v našem 3D skeneru. Je ideální pro stolní 3D skenery, je jednodušší než strukturované světlo z hlediska programování a principu děje a potřebuje menší počet komponent.

2.3.1 Triangulace

Triangulace je metoda zjišťování souřadnic a vzdáleností pomocí trigonometrického výpočtu. Vychází z toho, že pokud v trojúhelníku známe délku jedné strany a vnitřní úhly, dokážeme dopočítat jeho zbývající strany využitím goniometrických funkcí, sinové a kosinové věty. Například na Obr. 5 máme trojúhelník, který je tvořen dvěma referenčními body A a B a bodem C , do kterého chceme vzdálenost spočítat. Pokud jsou známy vzdálenost L mezi body A a B a odpovídající úhly v těchto bodech mezi stranou AB a neznámým bodem C (α a β), tak vzdálenost d od základny AB do bodu C lze spočítat pomocí ekvivalentních vzorců:

$$d = L \frac{1}{\cot \alpha + \cot \beta} = L \frac{\sin \alpha \cdot \sin \beta}{\sin(180^\circ - \alpha - \beta)} \quad (1)$$

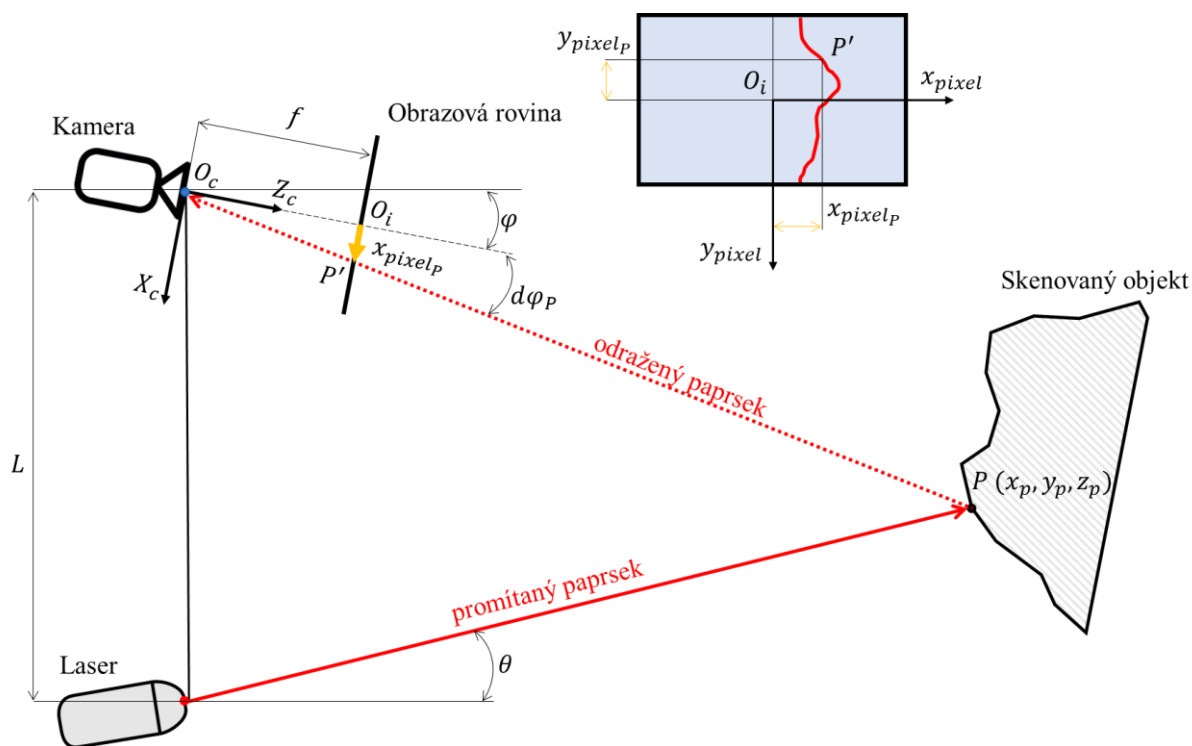


Obr. 5 – Geometrie triangulace [autor]

2.3.2 Obecná geometrie 3D skeneru

Typická geometrie laserové triangulace je zobrazená na Obr. 6 a byla inspirována schémata z [2], [6], [7] která jsou totožná a liší se pouze zvoleným souřadnicovým systémem a vzájemným uspořádáním komponent. Soustava se skládá z laseru, kamery ve formě camery obscury (podrobněji v 2.4.1) a bodu P na povrchu objektu. Jsou známy relativní polohy laseru a kamery – vzdálenost mezi ohniskem kamery a laserem L , úhel paprsku θ , úhel optické osy kamery φ . Také je známá poloha obrazové roviny (ve skutečnosti virtuální obrazové roviny), která je v ohniskové vzdálenosti f kamery.

Souřadnicový systém zvolíme v optickém středu kamery O_c : osy X_c a Y_c jsou rovnoběžné s obrazovou rovinou kamery, ale osa X_c je ve směru šířky roviny a osa Y_c ve směru výšky roviny. Osa Z_c je kolmá na obě předchozí a je totožná s optickou osou kamery. Hlavním cílem je výpočet souřadnice z_P bodu P ve zvoleném souřadnicovém systému.



Obr. 6 – Obecná geometrie laserové triangulace, pohled v rovině XZ kamery [autor]

Prvním krokem je zpracování obrazu z kamery a určení polohy promítnuté laserové čáry. Po zpracování dostáváme sadu obrazových souřadnic (x_{pixel}, y_{pixel}) nalezených pixelů laserové čáry v souřadnicovém systému obrazu, kde x_{pixel} je sloupec a y_{pixel} je řádek pixelu.

Dalším krokem je určení úhlu $d\varphi$ odraženého paprsku vzhledem k optické ose kamery. Souřadnice obrazu P' bodu P jsou $(x_{pixel_p}, y_{pixel_p})$ tedy se odpovídající $d\varphi_p$ rovná:

$$d\varphi_p = \tan^{-1}\left(\frac{x_{pixel_p}}{f}\right) \quad (2)$$

Je vidět, že promítaný laserový paprsek, odražený laserový paprsek a vzdálenost L tvoří podobný trojúhelník jako na Obr. 5. Ze sinové věty dostáváme:

$$\frac{\text{promítaný paprsek}}{\sin(90^\circ - (\varphi + d\varphi_p))} = \frac{\text{odražený paprsek}}{\sin(90^\circ - \theta)} = \frac{L}{\sin(\theta + \varphi + d\varphi_p)} \quad (3)$$

$$\text{odražený paprsek} = \frac{L \cdot \sin(90^\circ - \theta)}{\sin(\theta + \varphi + d\varphi_p)} \quad (4)$$

Promítnutím délky odraženého paprsku z rovnice (4) na osu Z_c dostaneme souřadnice z_p bodu P v souřadnicovém systému kamery:

$$z_p = \text{odražený paprsek} \cdot \cos(d\varphi_p) = \frac{L \cdot \sin(90^\circ - \theta) \cdot \cos(d\varphi_p)}{\sin(\theta + \varphi + d\varphi_p)} \quad (5)$$

S využitím podobnosti trojúhelníků v optické soustavě kamery a spočítané souřadnice z_p z rovnice (5) můžeme dopočítat zbývající souřadnice bodu P :

$$x_P = \frac{x_{pixel_P} \cdot Z_P}{f} \quad (6)$$

$$y_P = \frac{y_{pixel_P} \cdot Z_P}{f} \quad (7)$$

Stejným způsobem můžeme použít rovnice (2), (5), (6), (7) i pro další nalezené pixely laserové čáry v obraze (posunutí podél y_{pixel} obrazu) a tím získáme prostorové souřadnice všech bodů laserové čáry na povrchu objektu v souřadnicovém systému kamery.

Posledním krokem bude transformace těchto bodů z lokálního souřadnicového systému kamery do globálního souřadnicového systému a jejich sloučení s předchozími naměřenými body.

2.3.3 Zvolená geometrie 3D skeneru

Z konstrukčních důvodů bylo zvoleno, že zdroj laserového záření se bude dívat přímo na skenovaný objekt, tzn. $\theta = 0^\circ$. Úhel optické osy kamery φ a vzdálenost základny L je možné nastavovat podle rozměru skenovaného objektu. Lokální souřadnicový systém je jako na Obr. 6 umístěn v optickém středu kamery, pak se rovnice (5) zjednoduší do tvaru (8). Rovnice (6) a (7) budou mít drobný rozdíl – místo jediné ohniskové vzdálenosti f osy kamery X_c a Y_c budou mít odpovídající ohniskové vzdálenosti f_x a f_y . To je dáno matematickým modelem kamery (2.4) a výsledky kalibrace (3.4).

Prostorové souřadnice i -tého bodu laserové čáry, kterému odpovídá pixel se souřadnicemi $(x_{pixel_i}, y_{pixel_i})$, se budou v soustavě kamery rovnat:

$$z_i = \frac{L \cdot \cos(d\varphi_i)}{\sin(\varphi + d\varphi_i)} \quad (8)$$

$$x_i = \frac{x_{pixel_i} \cdot z_i}{f_x} \quad (9)$$

$$y_i = \frac{y_{pixel_i} \cdot z_i}{f_y} \quad (10)$$

Globální souřadnicový systém je umístěn ve středu rotace stolku, na kterém se bude nacházet objekt během skenování. Transformace probíhá pomocí matice transformace, která vyjadřuje polohu kamery vůči středu rotace stolku a aktuální úhel otočení stolku. Výsledkem jsou souřadnice bodů laserové čáry globálním souřadnicovým systémem a musejí se shodovat s reálnými body skenovaného objektu.

2.4 Matematický model kamery

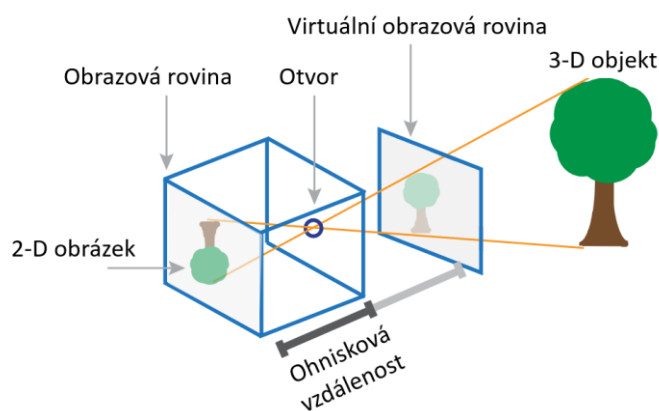
Z popisu v 2.2.1 a z geometrie v 2.3.2 už víme, že laserová triangulace využívá kameru pro zachycení laserové čáry na skenovaném objektu a pro správnou práci skeneru ji budeme muset zkalibrovat.

Kalibrace kamery je proces odhadu parametrů obrazového snímače a objektivu kamery podle pořízených snímků. Je nezbytnou součástí v počítačovém vidění nebo v robotice, např. pro navigační systémy a 3D rekonstrukce [8]. Nalezené parametry kamery budou užitečné v dalších výpočtech spojených s určováním vzdálenosti do skenovaného objektu.

Proces kalibrace bude probíhat v softwaru MATLAB, jehož kalibrační algoritmus pro běžné kamery využívá matematický model kamery zahrnující model camera obscura a zkreslení objektivu. Takový model přiměřeně popisuje vlastnosti a tvorbu obrazu reálné kamery například v počítačovém vidění.

2.4.1 Model camera obscura

Model camera obscura je matematický model kamery, který popisuje vztah mezi 3D souřadnicemi bodu ve scéně a 2D souřadnicemi jeho projekce v rovině obrazu. Samotná camera obscura (též dírková komora nebo pinhole, dále jen "dírková komora") je jednoduché optické zařízení ve tvaru světlotěsné krabice s malým otvorem místo objektivu. Světelné paprsky procházejí tímto otvorem a promítají převrácený obraz na opačnou stranu kamery (Obr. 7).



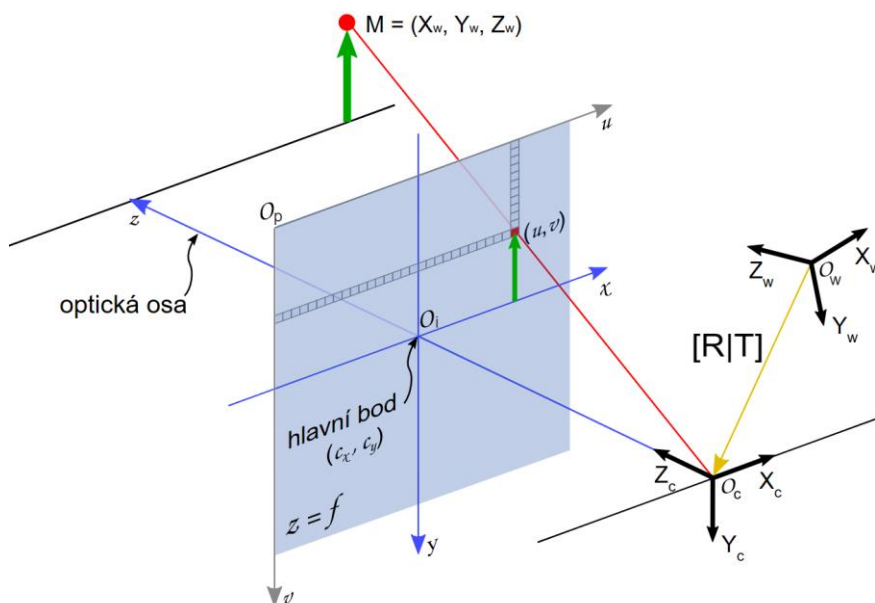
Obr. 7 – Camera obscura (dírková komora), přeloženo z [8]

Protože dírková komora neobsahuje objektiv, tak tento matematický model nezohledňuje geometrické zkreslení objektivu a je považován za ideální. Aby přesně reprezentovat reálné kamery, úplný matematický model kamery popisuje geometrické zkreslení objektivu zvlášť pomocí koeficientů distorze, o nichž je podrobněji napsáno v 2.4.2.

Obraz získávaný promítáním je převrácený o 180°, což trochu ztěžuje jeho vnímání. Proto pro pohodlnou práci a geometrické výpočty místo obrazové roviny budeme uvažovat virtuální obrazovou rovinu (Obr. 7), která se nachází před kamerou ve vzdálenosti f od otvoru kamery a na ní "virtuálně" vzniká stejný obraz jako na obrazové rovině, ale již není převrácený. Tím získáme

ekvivalentní geometrii dírkové komory se vzpřímeným obrazem (lze si také představit, že před objektivem kamery je umístěn průhledný papír, přes který se díváme na objekt) a právě takové schéma modelu dírkové komory se nejčastěji vyskytuje v dokumentacích a návodech ke kalibraci kamery. Příklad schématu z knihovny OpenCV je zobrazen za Obr. 8, kde

- O_w – počátek světových souřadnic.
- O_c – střed kamery (také optický střed), počátek souřadnicového systému kamery a zde se také nachází otvor dírkové komory. Středem kamery procházejí všechny projekční čáry (červené), které promítají body prostoru do roviny obrazu a optická osa kamery.
- Rovina kolmá na optickou osu a ve vzdálenost f od O_c – obrazová rovina s promítnutými body prostoru.
- O_i – hlavní bod, průsečík optické osy kamery a roviny obrazu (obvykle blízko středu obrazu).
- O_p – počátek souřadnicového systému obrazu v počítačové reprezentaci.
- M – bod v prostoru se světovými souřadnicemi X_w, Y_w, Z_w .
- $[u, v]$ – souřadnice (v pixelech) projekce bodu M v obrazu.



Obr. 8 – Geometrie modelu dírkové komory, přeloženo z [9]

Parametry modelu dírkové komory jsou popsány takzvanou maticí kamery P (2), která mapuje 3D scénu (souřadnice X_w, Y_w, Z_w) do obrazové roviny kamery (do souřadnic u, v v pixelech). Matice P se skládá z matice vnějších parametrů $[R \cdot T]$ a matice vnitřních parametrů K .

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = P \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (11)$$

Matice vnějších parametrů $[R \cdot T]$ představuje orientaci a polohu kamery v prostoru, skládá se z matice rotace R a matice posunutí T . Hlavní funkce je transformace světového souřadnicového systému O_w do souřadnicového systému kamery O_c v homogenních souřadnicích (12).

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (12)$$

Matice vnitřních parametrů K představuje fyzikální vlastnosti související s mechanikou a optickou konstrukcí kamery a také popisuje transformaci bodů ze 3D souřadnicového systému kamery O_c do 2D souřadnicového systému obrazové roviny O_p (13).

Matice má tvar (14), kde f_x a f_y jsou ohniskové vzdálenosti kamery v pixelech (dvě hodnoty, protože pixely na snímači mohou mít obdélníkový tvar), s je koeficient šikmosti pixelů, který je u většiny kamer nulový, $[c_x, c_y]$ jsou souřadnice hlavního bodu O_i v obraze v pixelech a vyjadřují posunutí souřadnicového systému kamery vůči souřadnicovému systému obrazu O_p (pokud hlavní bod není ve středu obrazu, tzn. střed senzoru kamery není přesně na optické ose).

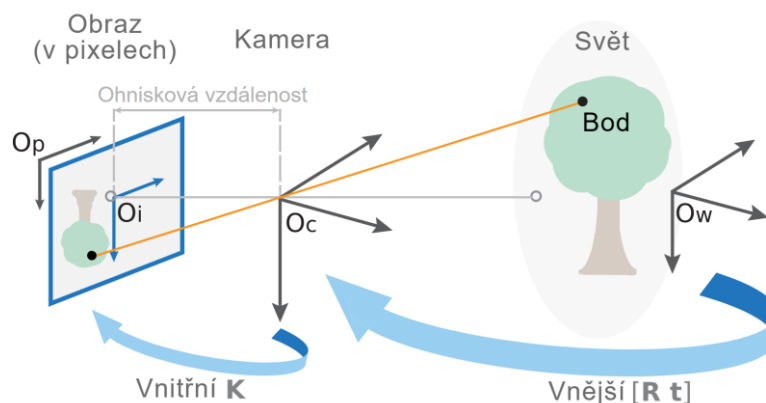
$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} \quad (13)$$

$$K = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (14)$$

Matice vnitřních parametrů K nezávisí na prohlížené scéně a orientaci v prostoru, je konstantní pro dané rozlišení obrazu a ohniskovou vzdálenost kamery [9]. Je důležité to pamatovat při použití kamer se zoom objektivy.

Jestliže shrneme všechny probrané parametry kamery, získáme obecnou projekční rovnici modelu dírkové komory (15), která je rozšířením rovnice (11). Na Obr. 9 je schematicky znázorněn proces transformace prostorových souřadnic O_w do souřadnicového systému kamery O_c a pak transformace z O_c do obrazové roviny O_p (na obrázku je původní geometrie dírkové komory s převraceným obrazem, což nemá vliv na znázornění transformaci).

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (15)$$



Obr. 9 – Transformace prostorových souřadnic do obrazové roviny, přeloženo z [8]

2.4.2 Zkreslení objektivu

Matice kamery P neobsahuje informaci o zkreslení objektivu, protože je založená na ideálním modelu kamery bez objektivu (Obr. 7). Proto se geometrické zkreslení objektivu modeluje pomocí zvláštních rovnic, a to pro radiální a tangenciální zkreslení objektivu.

Radiální zkreslení je způsobeno neideálním parabolickým tvarem čočky. To způsobuje odlišný lom paprsků v blízkosti optické osy a na okrajích čočky, což vede k zakřivení obrazu (Obr. 10). Rozlišují dva druhy radiálního zkreslení:

- pozitivní (poduškovité) – obraz je směrem k okrajům roztažen a jeví se jako dutý.
- negativní (soudkovité) – obraz se zmenšuje při vzdalování od optické osy a jeví se jako vypuklý.

Radiální zkreslení se popisuje rovnicí (7),

$$\begin{aligned}x_{zkreslený} &= x \cdot (1 + k_1 \cdot r^2 + k_2 \cdot r^4 + k_3 \cdot r^6) \\y_{zkreslený} &= y \cdot (1 + k_1 \cdot r^2 + k_2 \cdot r^4 + k_3 \cdot r^6)\end{aligned}\tag{16}$$

kde k_1, k_2, k_3 jsou koeficienty radiálního zkreslení (k_3 se používá při silném zkreslení, např. u širokoúhlých objektivů), které pak využívá algoritmus na odstranění radiálního zkreslení z obrazu, (x, y) je poloha nezkreslených pixelu, $r^2 = x^2 + y^2$.

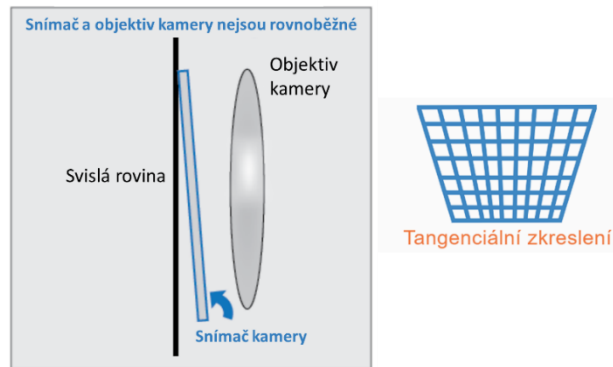


Obr. 10 – Radiální zkreslení, přeloženo z [8]

Tangenciální zkreslení vzniká, když snímač a objektiv kamery nejsou rovnoběžné, jako je zobrazeno na Obr. 11. To způsobuje zešikmení obrazu a zdá se, že některé oblasti obrazu vypadají blíž, než se očekávalo. Toto zkreslení se popisuje rovnicí (17),

$$\begin{aligned}x_{zkreslený} &= x + [2 \cdot p_1 \cdot x \cdot y + p_2 \cdot (r^2 + 2 \cdot x^2)] \\y_{zkreslený} &= y + [p_1 \cdot (r^2 + 2 \cdot y^2) + 2 \cdot p_2 \cdot x \cdot y]\end{aligned}\tag{17}$$

kde p_1, p_2 , jsou koeficienty tangenciálního zkreslení, které se také využívají ke korekci tangenciálního zkreslení z obrazu.



Obr. 11 – Uspořádání při tangenciálním zkreslení a samotné zkreslení, přeloženo z [8]

3. Praktická část

3.1 Vstupní požadavky

Cílem této práce je návrh a realizace bezkontaktního 3D skeneru založeného na technologii laserové triangulace s otočným stolcem. V počáteční fázi práce jsem dostal od svého vedoucího konstrukci s železným stojanem a otočným stolcem (Obr. 12), která byla v minulosti využita k demonstraci práci 3D skeneru. K dispozici jsem měl také krokový motor, driver krokového motoru, USB kameru a řídicí jednotku Arduino MEGA 2560 a přístup do laboratoře s 3D tiskárnou. Bylo rozhodnuto modernizovat poskytnutou konstrukci tak, aby 3D skener splňoval následující požadavky:

- skenování malých objektů přibližně 200x200x200 mm
- možnost skenování těžkých objektů (do 5 kg)
- jednoduchá montáž a demontáž komponent
- nastavitelnost jednotlivých částí soustavy (např. lineární vedení)
- většina dílů se tiskne na 3D tiskárně
- očekávaná přesnost měření ± 1 mm
- možnost skenování horní a dolní částí objektu, aby nevznikaly mezery v 3D modelu



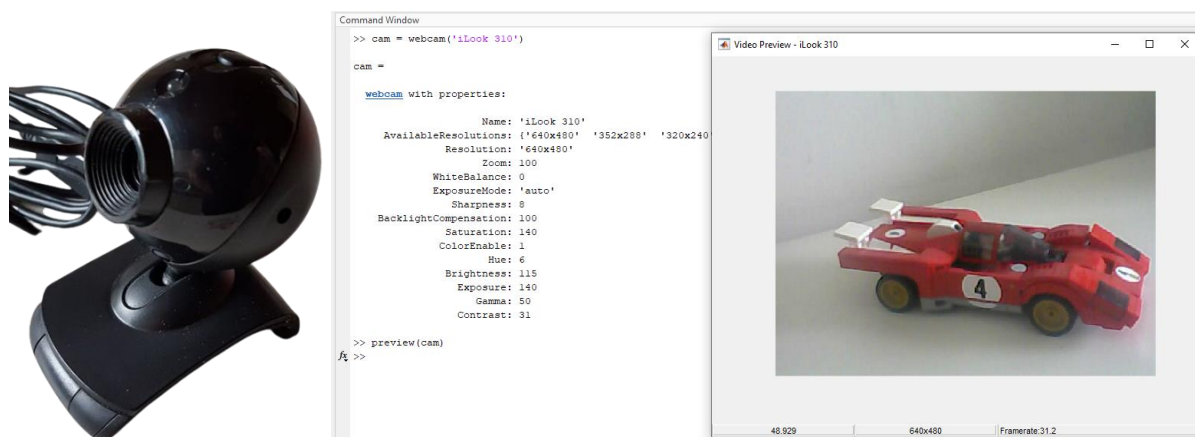
Obr. 12 – Původní konstrukce: železný stojan (zleva) a otočný stolek (zprava) [autor]

3.2 Výběr základních komponent

Z popisu technologie laserové triangulace víme, že daný typ 3D skeneru vyžaduje kameru, zdroj laserového záření, řídicí jednotku, počítač se softwarem a pohon pro otáčení stolku, na kterém se bude nacházet skenovaný objekt. Popis jednotlivých komponent je uveden níže.

3.2.1 USB webkamera

Poskytnutá webkamera je Genius iLook 310 (Obr. 13) s rozlišením 0,3 Mpx, CMOS senzorem a ručně nastavitelným objektivem. Kamera je stará a podle mého názoru má špatnou kvalitu snímků, což může ztěžovat detekci laserové čáry. Během ovládaní někdy dochází k chybám a předpokládám, že to souvisí s vlastní kamerou notebooku a nějakým způsobem dochází ke konfliktu. Chyby neovlivňují měření, pouze čas nastavení systému. Pro vysílání videa z kamery a import obrazů do MATLAB používáme knihovnu *MATLAB Support Package for USB Webcams*, která také umožňuje nastavovat některé vlastnosti kamery, například rozměr obrazu, expozici, jas, kontrast atd jako je uvedeno na Obr. 13.



Obr. 13 – Kamera iLook 310 (zleva) a obraz z kamery v MATLAB (zprava) [autor]

3.2.2 Zdroj laserového záření

Pro promítání laserové čáry na skenovaný objekt byl zvolen červený laserový modul o výkonu 5mW s optikou vykreslující čáru, která se ručně zaostřuje otáčením prstence. Napájecí napětí je 3 - 5V DC a jmenovitý proud méně 17mA, což umožňuje připojit laserový modul přímo k digitálnímu pinu Arduino bez dalších součástek.



Obr. 14 – Laserový modul s optikou vykreslující čáru [10]

3.2.3 Krokový motor a driver krokového motoru

Poskytnutý motor (Obr. 15) je krokový motor NEMA 17 modelové řady 17HS8401 s následujícími důležitými vlastnostmi pro řízení [11]:

- druh: hybridní bipolární dvoufázový
- úhel kroku: $1,8^\circ$
- počet kroků dokola: 200
- jmenovitý proud: 1,7A
- statický moment: 0,5Nm

Krokový motor je synchronní bezkartáčový motor s několika vinutími, který se ovládá proudovými impulsy. Při průchodu proudem jedním z vinutí statoru nastává pootočení a zablokování rotoru v rovnovážné poloze. Při další sekvenční aktivaci vinutí vzniká diskretní uhlový posuv rotoru, což jsou kroky. Výhodou tohoto typu elektromotoru je řízení úhlové polohy rotoru bez zpětné vazby s vysokou přesností za normálního provozu.

K ovládání krokového motoru se používá speciální ovladač – driver krokového motoru, který aktivuje vinutí motoru ve správném pořadí, aby zajistil otáčení rotoru. Poskytnutý ovladač je mikrokrokový driver 4A TB6600 (Obr. 15), který je vhodný na ovládání bipolárních krokových motorů různých rozměrů. Má následující důležité vlastnosti dle [12], ale mohou se mírně lišit od výrobce k výrobcu:

- napájecí napětí: 9 - 42V DC, doporučené 36V DC
- maximální výstupní proud: 3,5A na fázi, pikový 4A
- rozlišení mikrokroků: plný, 1/2, 1/4, 1/8, 1/16, 1/32
- ohraná: ochrana proti nadproudu, podpětí, přehřátí a proti zkratu na výstupu



Obr. 15 – Krokový motor (zleva) a driver (zprava) [autor]

3.2.4 Napájení

V mé soustavě 3D skeneru neexistuje nějakého jediného napájecího zdroje, který by napájel všechny komponenty. USB kamera a deska Arduino jsou napájeny od počítače přes USB kabel, laserový modul je díky nízké spotřebě napájen z Arduino. Nejvyšší spotřebu mají pouze krokový motor a jeho driver s maximálně možnými hodnotami 42V DC a 4A (ve skutečnosti s hodnotou proudu 1,7A dle vlastností krokového motoru NEMA 17). Proto jsem nepovažoval za

nutné volit samostatný napájecí zdroj pouze pro krokový motor. Při experimentálním ověření 3D skeneru bude použit laboratorní zdroj MANSION NP-9615 (Obr. 16) s výstupním napětím až 30V a výstupním proudem až 5A, což je dostatečné pro ovládání krokového motoru.



Obr. 16 – Laboratorní zdroj MANSION NP-9615 [autor]

3.3 Návrh konstrukce

3.3.1 Konstrukce skenovací hlavy kamera-laser

Aby fungovala geometrie laserové triangulace, musí být USB kamera a laserový modul umístěny na společné základně a nasměrovány na skenovaný objekt. Hlavním požadavkem je možnost nastavení polohy jednotlivých prvků konstrukce a jejich snadná montáž/demontáž.

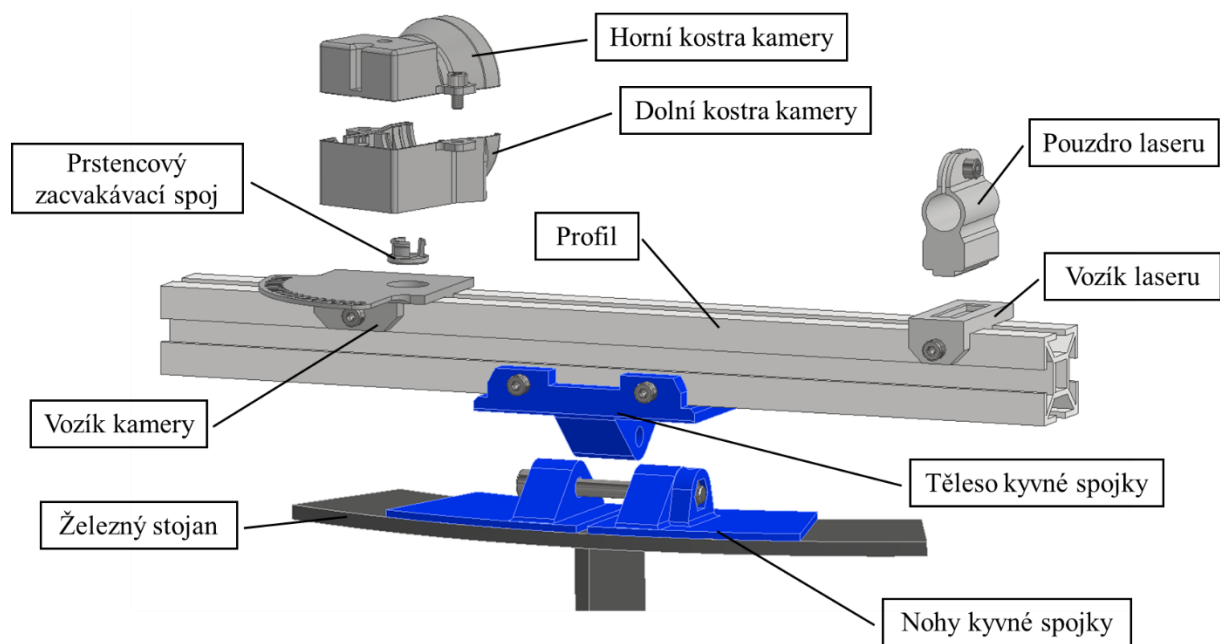
Jako základná byl zvolen hliníkový profil 30x30 se čtyřmi T-drážkami, který byl k dispozici v laboratoři. Tyto drážky budou spojovat skenovací hlavu se zbytkem konstrukce a jedná z nich bude využívána pro lineární pohyb kamery a laseru.

Hliníkový profil spojím s horní částí železného stojanu z Obr. 12 pomocí kyvné spojky, aby byla možnost nastavovat směr pohledu skenovací hlavy. Upevnění profilu v drážce kyvné spojky se provádí třecími silami vznikajícími při utahování dvou šroubů. Sklon kyvné spojky je fixován utažením šroubového spoje, který přitahuje nohy spojky k jejímu tělesu. Nohy kyvné spojky k železnému stojanu připevním pomocí šroubových spojů.

Původní kostra webkamery umožňovala rotaci o 360° a naklápění o 45° dopředu a dozadu, přičemž poslední je nežádoucí, protože osa laserového modulu a osa kamery musí být ve stejné rovině rovnoběžné se základnou. Plánuju také použít červený filtr, který musí být umístěn před kamerou. Proto byla navržena nová kostra kamery, která má drážku pro červený filtr a osu rotace v ohnisku kamery. Rotace probíhá pomocí prstencového zacvakávacího spoje, který se vkládá do otvoru kostry kamery.

Kamera a laser se pohybují po drážce hliníkového profilu na vozících a jejich poloha je také fixovaná utahováním šroubů. Vozík kamery má stupnici pro nastavení úhlu osy kamery φ vůči základně a otvor, do kterého se umísťuje zacvakávací spoj, aby jej bylo možné v případě poruchy snadno vyměnit. Vozík laseru je jednoduchý a má pouze drážku pro vložení pouzdra (tzn. úhel laserového paprsku je $\theta = 0^\circ$), do kterého se laserový modul umísťuje pomocí svěrného spoje.

Popsaná konstrukce skenovací hlavy je zobrazena na Obr. 17.



Obr. 17 – Konstrukce skenovací hlavy [autor]

3.3.2 Návrh převodovky

Existují návody, které popisují proces dimenzování motoru na základě požadovaných parametrů pohybu a mechanismu pohonu jako například v návodu od společnosti Oriental Motor [13]. Máme trochu jiný cíl – chceme modernizovat poskytnutý otočný stolek z Obr. 12 přidáním převodovky pro dosažení požadované dynamiky a využitím poskytnutého krokového motoru. Další návrh se řídí návodem od Oriental Motor, jenom se prohodí známé a neznámé parametry.

Popis mechanismu: Objekt se umísťuje na otočný stolek, který je přes převodovku spojen s krokovým motorem. V každém kroku skenování se stolek pootočí o určitý úhel, a to se opakuje, dokud není naskenován celý tvar předmětu.

Setrvačnost zátěže: Moment setrvačnosti je mírou odporu jakéhokoli fyzického objektu vůči změně jeho rychlosti rotace, je označován $J [kg \cdot m^2]$. Má velký vliv na celou dynamiku systému, například při výpočtu momentu zrychlení a převodového poměru převodovky. Moment setrvačnosti zátěže (označíme J_z) je součet momentů setrvačnosti všech komponent, které budou poháněny krokovým motorem: rotor, vstupní kolo převodovky, výstupní kolo převodovky, otočný stolek, skenovaný objekt.

Moment setrvačnosti rotoru jsem převzal z manuálu krokového motoru, $J_{km} = 6,8 kg \cdot mm^2$. Setrvačnost otočného stolku byla odhadnuta z modelu stolku vytvořeného v Autodesk Inventor, $J_s = 14900 kg \cdot mm^2$. Moment setrvačnosti skenovaného objektu J_{so} předem nevíme, ale ze vstupních požadavků víme maximální hmotnost (do 5 kg) a rozměry objektu (200x200x200 mm), takže zkusíme odhadnout “nejhorší” moment setrvačnosti, který by teoretický mohl mít skenovaný objekt.

Za předpokladu, že celý objekt je soustředěn v jednom bodě a je vzdálen 100 mm od osy rotace, “nejhorší” moment setrvačnosti se bude rovnat momentu setrvačnosti tenké obruče:

$$J_{so} = m \cdot r^2 = 5 \cdot 100^2 = 50000 \text{ kg} \cdot \text{mm}^2, \quad (18)$$

Poměr setrvačnosti a převodový poměr: Krokové motory a servomotory nemohou pohanět nekonečnou setrvačností zátěže, neboť může docházet ke ztrátě kroku, zahřátí a vibraci motoru. Pro dané typy motorů je ekvivalentem přípustné setrvačnosti zátěže poměr setrvačnosti, což je poměr celkové setrvačnosti zátěže J_z k setrvačnosti rotoru motoru J_{km} . Spočítáme předběžný poměr setrvačnosti, jako kdyby byl motor připojen přímo ke stolku:

$$\frac{J_z}{J_{km}} = \frac{J_{so} + J_s}{J_{km}} = \frac{50000 + 14900}{6,8} \doteq 9544 \quad (19)$$

Podle návodu je pro náš krokový motor NEMA 17 doporučený maximální poměr 10:1, takže takový mechanismus by byl nestabilní. Nejlepším řešením je přidání převodovky, která zmenšuje moment setrvačnosti o druhou mocninu převodového poměru. Taková setrvačnost se jmenuje odražená setrvačnost zátěže a lze její vnímat jako setrvačnost zátěže, která byla přemístěná na hřídel motoru.

Použijeme dané tvrzení pro výpočet potřebného převodového poměru převodovky. Na výstupním hřídeli převodovky je uloženo druhé kolo převodovky s J_{kolo2} , stolek s J_s a skenovaný objekt s J_{so} , takže budou mít odraženou setrvačnost na rotoru motoru. Moment setrvačnosti prvního kola J_{kolo1} zůstává stejný. Rovnice pro výpočet poměru setrvačnosti pro mechanismus s převodovým poměrem i má tvar:

$$\frac{J_z}{J_{km}} = \frac{\frac{J_{so}}{i^2} + \frac{J_s}{i^2} + \frac{J_{kolo2}}{i^2} + J_{kolo1}}{J_{km}} \quad (20)$$

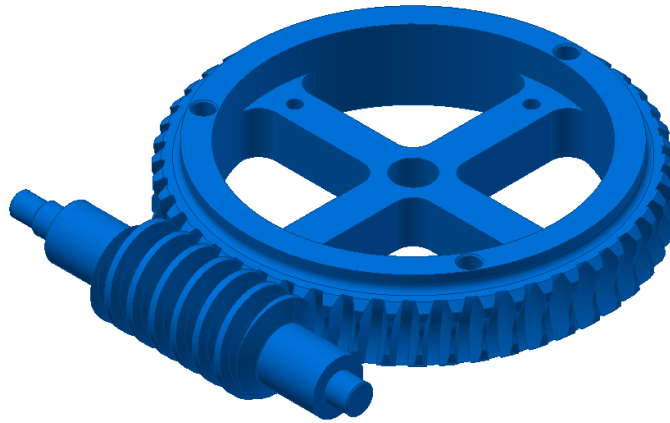
Chceme, aby systém byl stabilní, a proto hodnota poměru z rovnice (20) by měla být menší než maximální doporučená hodnota 10:1 (5:1 s rezervou, protože nevíme momenty setrvačnosti kol). Z tohoto požadavku vyjádříme předběžný převodový poměr i' převodovky:

$$i' \geq \sqrt{\frac{J_{so} + J_s}{5 \cdot J_{km}}} = \sqrt{\frac{50000 + 14900}{5 \cdot 6,8}} = 43,7 \rightarrow i = 50 \quad (21)$$

Pro návrh ozubení používám generátor v Autodesk Inventor. Po několika generování bylo zjištěno, že ozubené kolo je příliš velkým, přestože byl zvolen minimální modul (2 mm), který 3D tiskárna schopna přesně tisknout, a minimální možný počet zubů na pastorku (12 zubů). Zároveň nechtěl jsem příliš zvyšovat výšku pod stolem, proto jsem rozhodnul použít šnekový převod. Navržené soukolí je zobrazeno na Obr. 18 s následujícími geometrickými vlastnostmi:

Tabulka 2: Geometrické vlastnosti šnekového převodu

Parametr	Šnek	Šnekové kolo
m [mm]	2,5	
počet chodů / zubů	1	50
d [mm]	25	125
délka šneku / šířka ozubení	56	20
J [kg · mm ²]	3,7	556,4



Obr. 18 – Navržený šnekový převod [autor]

Nakonec, po určení momentů setrvačnosti všech komponent pohaněných krokovým motorem, zkontrolujeme poměr setrvačnosti mechanismu podle rovnice (20). Výsledek výpočtu v (22) je vyhovující a systém je stabilní.

$$\frac{J_z}{J_{km}} = \frac{\frac{J_{so}}{i^2} + \frac{J_s}{i^2} + \frac{J_{kolo2}}{i^2} + J_{kolo1}}{J_{km}} = \frac{50000 + 14900 + 556,4}{50^2} + 3,7 = 4,4 \leq 5 \quad (22)$$

3.3.3 Dynamika otočného stolku

Krouticí moment: Po navržení převodu zkontrolujeme, zda krokový motor zvládne pohanět daný systém. Potřebný krouticí moment se skládá ze dvou složek:

$$M_k = M_z + M_{zr} \quad (23)$$

kde M_z je zatěžovací moment způsobený třením, tíhou a dalšími vnějšími silami působící na mechanismus při rotaci, M_{zr} je moment zrychlení působící pouze při zrychlení a zpomalování, je určen momentem setrvačnosti zátěže J_z a uhlovým zrychlením ε .

Zatěžovací moment: Na Obr. 19 jsou zobrazeny síly působící na stůl při jeho rotaci, kde G – tíhová síla skenovaného objektu a stolu,

$$G = (m_{so} + m_s) \cdot g = (5 + 0,8) \cdot 9,81 = 56,9 \text{ N} \quad (24)$$

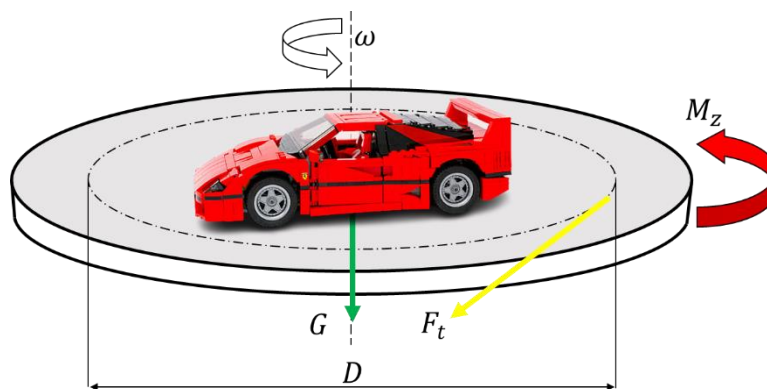
F_t – třecí síla v ložisku,

$$F_t = \mu \cdot N = \mu \cdot G = 0,01 \cdot 56,9 = 0,6 \text{ N} \quad (25)$$

kde $\mu = 0,01$ je součinitel tření v axiálním ložisku. Zatěžovací moment M_z se v tomto případě rovná:

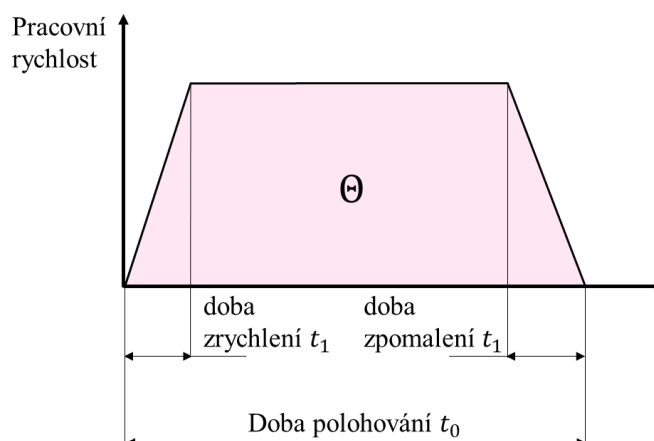
$$M_z = F_t \cdot \frac{D}{2} = 0,6 \cdot \frac{50}{2} = 15 \text{ N} \cdot \text{mm} \quad (26)$$

Vypočítali jsme tedy stálou složku točivého momentu, kterou musíme překonávat pro neustálé rovnoměrné otáčení.



Obr. 19 – Trvalé síly a zatěžovací moment působící na stůl [autor]

Rychlostní profil: Abychom určili moment zrychlení M_{zr} , potřebujeme vědět úhlové zrychlení ε neboli jak se stůl otáčí v čase. Typický rychlostní profil je znázorněn na Obr. 20 a zvolíme jej pro popis pootočení stolku v každém kroku.



Obr. 20 – Lichoběžníkový rychlostní profil [autor]

Doba polohování t_0 , doba zrychlení/zpomalení t_1 , úhel pootočení stolu Θ_s , velikost mikrokroku mk a další parametry pohybu se volí obtížně, protože korelují s úhlovým zrychlením a zrychlovacím momentem. Zvolím následující předběžné hodnoty, ale mohou být změněny při kontrole činnosti mechanismu:

- úhel pootočení stolku v každém kroku: $\Theta_s = 1,8^\circ$
- doba polohování stolku: $t_0 = 1 \text{ s}$
- doba zrychlení (doporučují 25% od t_0): $t_1 = 0,25 \text{ s}$
- převodový poměr: $i = 50$
- jmenovitý úhel krokového motoru: $\Theta_{km} = 1,8^\circ$
- nastavený mikrokrok driveru: $mk = 1/8$

Krokový motor řídíme pomocí pulzních signálů (více v 3.7.1), spočítáme potřebný počet pulzů A pro pootočení stolu o úhel Θ_s :

$$A = \frac{\theta_s \cdot i}{\theta_{km} \cdot mk} = \frac{1,8 \cdot 50}{1,8 \cdot 1/8} = 400 \text{ pulzů} \quad (27)$$

Pracovní frekvenci f_{km} (ekvivalent pracovní rychlosti) krokového motoru lze určit z plochy pod rychlostním profilem, která se rovná A :

$$A = \frac{t_0 + (t_0 - 2t_1)}{2} \cdot f_{km} \rightarrow f_{km} = \frac{A}{t_0 - t_1} = \frac{400}{1 - 0,25} = 533,3 \text{ Hz} \quad (28)$$

Otáčky n_{km} , úhlovou rychlost ω_{km} a zrychlení ε_{km} krokového motoru, které potřebujeme pro výpočet momentu zrychlení M_{zr} , dopočítáme podle následujících rovnic:

$$n_{km} = \frac{f_{km} \cdot \theta_{km} \cdot mk}{360^\circ} \cdot 60 = \frac{533,3 \cdot 1,8 \cdot 1/8}{360^\circ} \cdot 60 = 20 \text{ ot/min} \quad (29)$$

$$\omega_{km} = 2 \cdot \pi \cdot n = 2 \cdot \pi \cdot \frac{20}{60} = 2,1 \text{ rad/s} \quad (30)$$

$$\varepsilon_{km} = \frac{\omega}{t_1} = \frac{2,1}{0,25} = 8,36 \text{ rad/s}^2 \quad (31)$$

Moment zrychlení a špičkový krouticí moment: Shrňme si důležité výsledky výše uvedených výpočtů:

- moment setrvačnosti skenovaného objektu: $J_{so} = 50000 \text{ kg} \cdot \text{mm}^2$
- moment setrvačnosti stolku: $J_s = 14900 \text{ kg} \cdot \text{mm}^2$
- moment setrvačnosti šneku: $J_{kolo1} = 3,7 \text{ kg} \cdot \text{mm}^2$
- moment setrvačnosti šnekového kola: $J_{kolo2} = 556,4 \text{ kg} \cdot \text{mm}^2$
- moment setrvačnosti rotoru: $J_{km} = 6,8 \text{ kg} \cdot \text{mm}^2$
- převodový poměr: $i = 50$
- otáčky motoru: $n_{km} = 20 \text{ ot/min}$
- úhlové zrychlení motoru: $\varepsilon_{km} = 8,36 \text{ rad/s}^2$
- zatěžovací moment na stole: $M_z = 15 \text{ N} \cdot \text{mm}$

Moment zrychlení M_{zr} na hřídeli krokového motoru je roven součinu setrvačnosti zátěže J_z (s přihlédnutím odražené setrvačnosti) a úhlového zrychlení ε_{km} motoru:

$$\begin{aligned} M_{zr} &= J_z \cdot \varepsilon_{km} = \left(\frac{J_{so} + J_s + J_{kolo2}}{i^2} + J_{kolo1} + J_{rotor} \right) \cdot \varepsilon_{km} = \\ &= \left(\frac{50000 + 14900 + 556,4}{50^2} + 3,7 + 6,8 \right) \cdot 8,36 = 0,31 \text{ N} \cdot \text{mm} \end{aligned} \quad (32)$$

Zatěžovací moment M_z z rovnice (26) byl vypočítán vzhledem k stolu, musíme jej přenést na hřídel krokového motoru:

$$M'_z = \frac{M_z}{i \cdot \eta} = \frac{15 \text{ Nmm}}{50 \cdot 70\%} = 0,4 \text{ N} \cdot \text{mm} \quad (33)$$

kde $\eta = 70\%$ je předpokládaná účinnost navržené převodovky.

Celkově, špičkový krouticí moment M_{max} , který může teoreticky vzniknout na rotoru krokového motoru, se rovná:

$$M_{max} = M'_z + M_{zr} = 0,4 + 0,31 = 0,71 \text{ N} \cdot \text{mm} \quad (34)$$

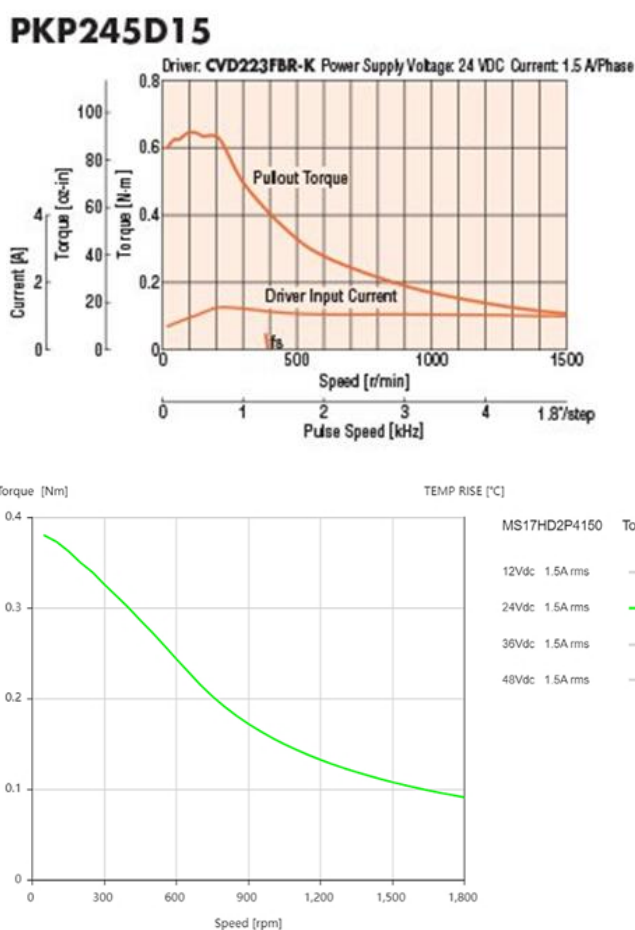
Předpoklady a odhady ve výpočtech zohledníme v bezpečnostním koeficientu $k = 2$. Ve výsledku máme točivý moment M_{km} , který musí krokový motor NEMA 17 vyrobít při otáčkách n_{km} :

$$M_{km} = M_{max} \cdot k = M_{max} \cdot 2 = 1,5 \text{ N} \cdot \text{mm} \quad (35)$$

$$n_{km} = 20 \text{ ot/min}$$

Jak uvádí Johann Tang (2020), “točivý moment produkováný krokovým motorem klesá s rostoucí rychlostí, musíme se podívat na křivku rychlosti a točivého momentu, abychom zjistili, zda bude krokový motor pracovat při této rychlosti nebo ne. “[14]

Pro poskytnutý model krokového motoru 17HS8401 jsem nenašel křivku točivého momentu, ale našel jsem modely motoru NEMA 17 s jinými označeními a praktický shodnými parametry, které již mají momentovou charakteristiku ve specifikacích. Z příkladů křivek na Obr. 21 je vidět, že při nízkých otáčkách (méně 100 ot/min) se hodnota točivého momentu pohybuje kolem hodnoty statického momentu, která se pro 17HS8401 rovná 0,5 Nm, což je více než dostatečné pro pohánění stolu dle rovnice (35).



Obr. 21 – Momentová charakteristika modelu PKP245D15A2 (nahore) [15] a modelu MS17HD2P4150 (dole) [16]

3.3.4 Konstrukce pohonu

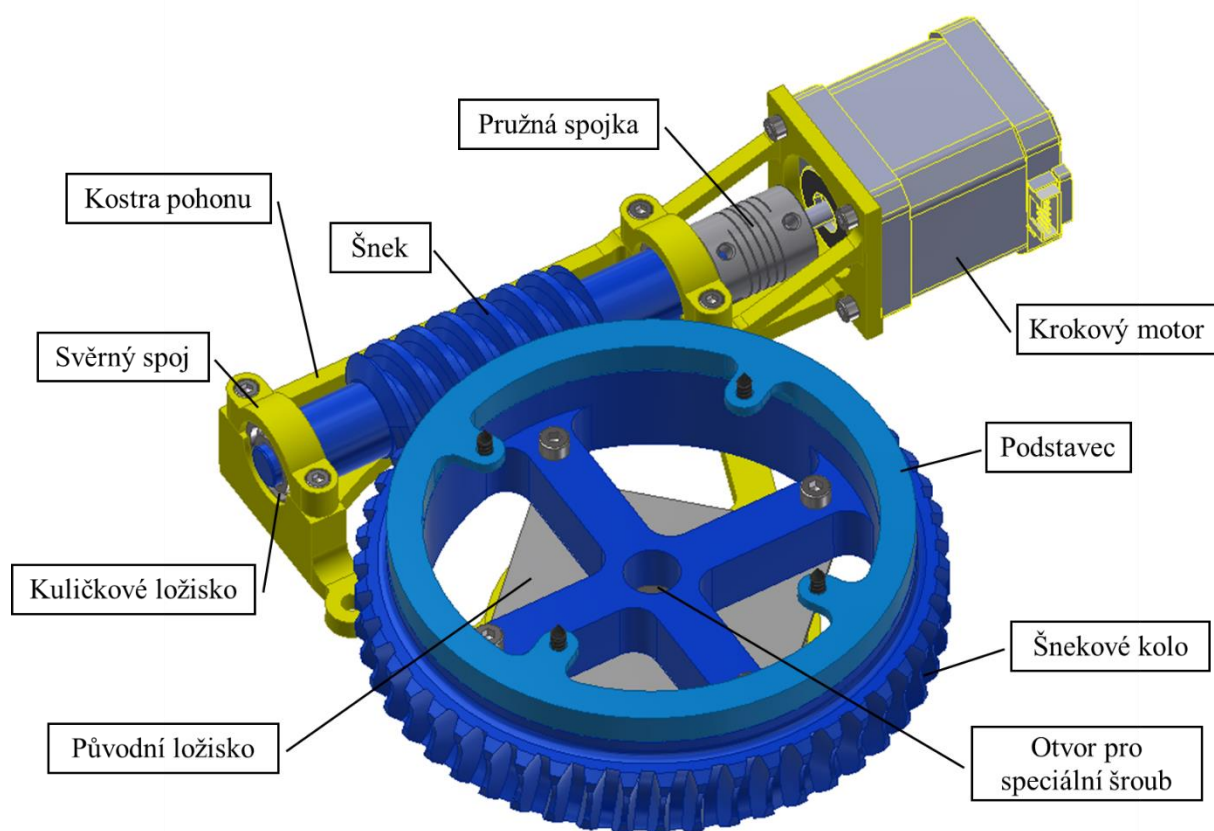
Původní konstrukce otočného stolu z Obr. 12 byla modifikovaná přidáním šnekového převodu navrženého v 3.3.2. Šnek a šnekové kolo se umísťují v kostře, která je pak k desce připevněna pomocí samořezných šroubů. K rámu kostry je pomocí šroubů připevněn krokový motor pohánějící celý mechanismus.

K otáčení šneku se používají dvojice jednořadových kuličkových ložisek 61900, která jsou upevněna v drážkách kostry pomocí svěrného spoje. Protože krokový motor rotuje přerušovaně s nízkými otáčkami, nepovažoval jsem za nutné kontrolovat ložiska. Testová rotace ukázala jejich spolehlivost a pevnost celého spoje. Rotor motoru a šnek jsou spojeny pomocí CNC pružné spojky 5x8.

Šnekové kolo je spojeno s původním ložiskem otočného stolku pomocí čtyř šroubových spojů. Dané ložisko je specifické a dle obchodů se používá v nábytků. K jeho upevnění se používá speciální šroub, který po utažení tlačí na kruhovou část ložiska, a tím ho přitlačuje k povrchu. Tato specifika byla zohledněna při návrhu spojení kola s kostrou.

Otvory na povrchu šnekového kola slouží k umístění podstavce, který je spojen s horním kotoučem stolu samořeznými šrouby. To umožňuje sundání stolku a zvětšuje prostor pro umístění krokového motoru.

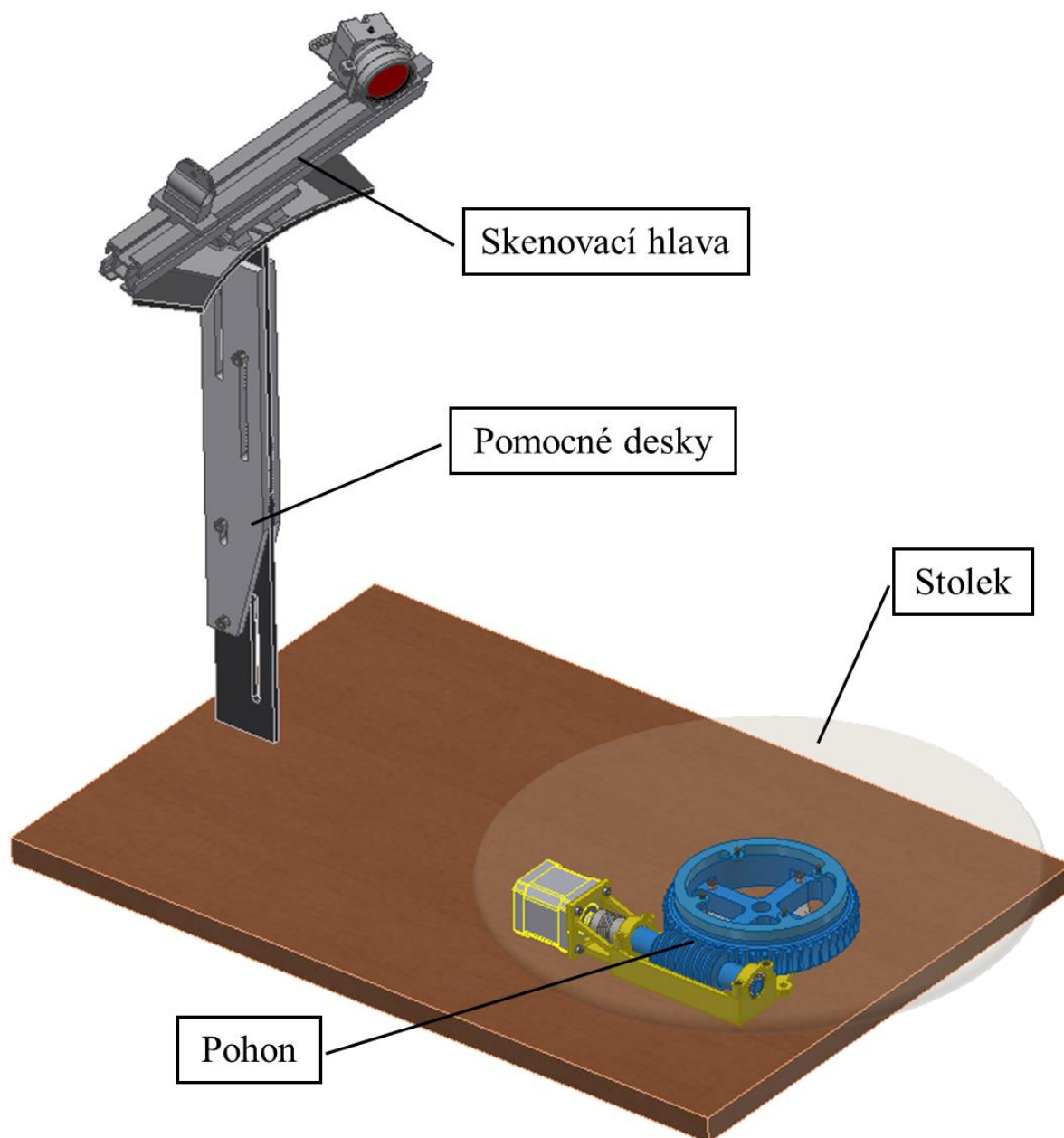
Popsaná konstrukce pohonu je zobrazena na Obr. 22 (bez modelu stolu, protože je velký a zakrýval by celou konstrukci).



Obr. 22 – Konstrukce pohonu stolku [autor]

3.3.5 Celková konstrukce 3D skeneru

Spojením všech částí dohromady dostáváme celkovou konstrukcí 3D skeneru, která je modernizovanou verzí původní poskytnuté konstrukce z Obr. 12. Při instalaci nastal drobný problém – výška železného stojanu byla nízká. Proto jsem vytisknul dvě desky, které umožňují umístit horní část stojanu (se skenovací hlavou) na dolní část a nastavovat výšku pomocí šroubů, jak to bylo původně. Navržená konstrukce 3D skeneru je zobrazená na Obr. 23.



Obr. 23 – Celková konstrukce 3D skeneru [autor]

3.4 Kalibrace webkamery

Prvním krokem k zajištění správné práce 3D skeneru je kalibrace kamery. Kalibrace bude probíhat v MATLAB pomocí aplikace *Camera Calibrator* (Obr. 24) z knihovny *Computer Vision Toolbox*. Pomocí několika snímků s kalibračním vzorem odhadneme vnitřní a vnější parametry webkamery a koeficienty zkreslení objektivu. Protože webkamera má nastavitelný objektiv, musíme mít na paměti, že při změně ohniskové vzdálenosti bude nutná rekalibrace.

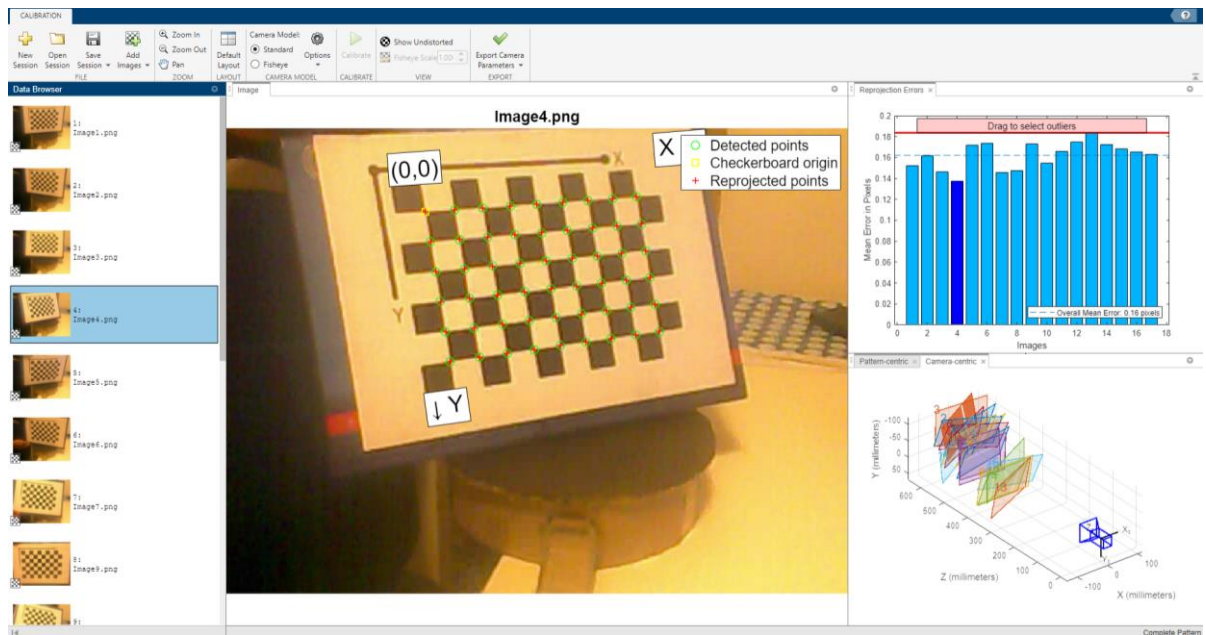
Před kalibrací je nutné buď nahrát hotové snímky nebo nastavit webkameru na pořizování snímků. V obou případech potřebujeme 10 až 20 fotografií s kalibračním vzorem pro získání přesných výsledků. Použil jsem šachovnicový vzor s velikostí čtverce 21 mm a pořídil jsem 20 snímků. V nastavení kalibrace jsem zvolil standartní model kamery, výpočet pouze dvou koeficientů radiálního zkreslení (k_1, k_2) a navíc jsem zvolil výpočet koeficientů tangenciálního zkreslení p_1, p_2 , protože nová kostra kamery z Obr. 17 nezaručuje kolmost osy objektivu na snímač kamery.

Po spuštění kalibrace algoritmus analyzuje nahrané fotografie, hledá rohy šachovnicových čtverců atd., a tím identifikuje vnitřní a vnější parametry kamery a koeficienty zkreslení čočky. Matematický model webkamery iLook 310 má při daném nastavení objektivu následující vnitřní parametry:

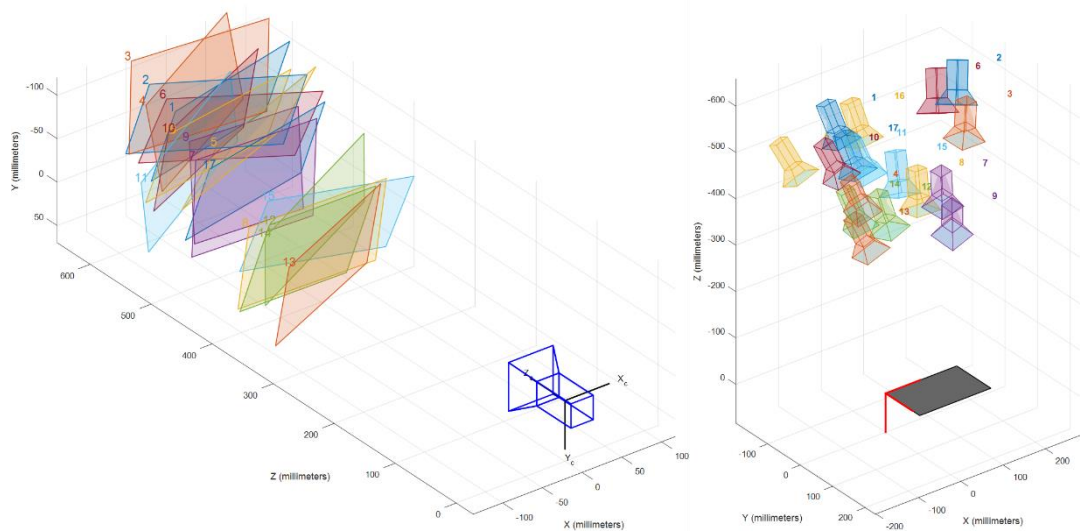
- optický střed: $\begin{bmatrix} c_x \\ c_y \end{bmatrix} = \begin{bmatrix} 320,2 \pm 1,5 \\ 232,6 \pm 1,2 \end{bmatrix} \text{ pixelů}$
- ohniskové vzdálenosti: $\begin{matrix} f_x = 874,9 \pm 1,7 \\ f_y = 870,9 \pm 1,6 \end{matrix} \text{ pixelů}$
- koeficienty radiálního zkreslení: $k_1 = 0,1877, k_2 = -0,4863$
- koeficienty tangenciálního zkreslení: $p_1 = 0,0011, p_2 = -0,0016$

Dalšími výsledky kalibrace jsou 3D grafy na Obr. 25, které vizualizují vnější parametry webkamery, T a R . Grafy slouží k posouzení přesnosti matematického modelu kamery. Pokud vzor nebo kamera jsou příliš daleko nebo blízko (chybná poloha v prostoru ve srovnání s očekávanou), tak takový snímek vnáší chyby do procesu kalibrace a model není zcela přesný.

Získané hodnoty budou užitečné v algoritmu určování vzdáleností do skenovaného objektu podle metody laserové triangulace a pomohou odstranit zkreslení ze snímků.



Obr. 24 – Prostředí Camera Calibrator: nastavení kalibrace (nahore), pořizené snímky (zleva), vybraný snímek s detekovanými rohy čtverců (uprostřed), graf s chybami reprojekce (vpravo nahore), grafy vnějších parametrů (vpravo dole) [autor]



Obr. 25 – Vizualizace vnějších parametrů webkamery: umístění kalibračních vzorů v souřadnicovém systému kamery (zleva), umístění kamer v souřadnicovém systému vzoru (zprava) [autor]

3.5 Kalibrace otočného stolku

Výsledkem algoritmu laserové triangulace v každém kroku skenování jsou souřadnice bodů laserové čáry na objektu v souřadnicovém systému kamery. Tyto body musí být přepočteny z lokálního souřadnicového systému do globálního souřadnicového systému 3D skeneru, aby získat celostní model skenovaného objektu.

Globální souřadnicový systém umístíme ve středu rotace stolku, kde rovina xy bude shodná s rovinou stolku a osa z bude směřovat nahoru. Cílem kalibrace otočného stolku je určení jeho středu rotace. Algoritmus kalibrace byl převzat z [17] a přizpůsoben našim podmínkám.

Myšlenka spočívá v tom, že na povrch stolu nalepíme kalibrační vzor (jako v kalibraci kamery) a pořídíme sadu snímků při různých natočeních stolu. Poloha středu rotace v obraze kamery před a po otáčení zůstává nezměněná.

Z každého snímku $k \in N$ určíme vnější parametry kamery – matice T_k (3x1) a R_k (3x3) pomocí příkazů z knihovny *Computer Vision Toolbox*. Střed rotace se nachází na povrchu stolu a má souřadnice $q = [q_x, q_y, 0]$ v souřadnicovém systému kalibračního vzoru a tento bod musí splňovat projekční rovnici (15) jako jakýkoli jiný bod v zorném pole kamery. Rovnice (36) je upravená rovnice (15) dle [17], kde w_k je součinitel měřítka, $[\tilde{u}_k, \tilde{v}_k, 1]^T = K^{-1} \cdot [u_k, v_k, 1]^T$ a $[u_k, v_k]$ jsou souřadnice projekce bodu q v obraze.

$$w_k \begin{bmatrix} \tilde{u}_k \\ \tilde{v}_k \\ 1 \end{bmatrix} = R_k \begin{bmatrix} q_x \\ q_y \\ 0 \end{bmatrix} + T_k \quad (36)$$

Protože souřadnice $[u_k, v_k]$ zůstávají neměnné, lze pro každý snímek k kalibračního vzoru definovat $[Q_x, Q_y, Q_z] \equiv w_k \cdot [\tilde{u}_k, \tilde{v}_k, 1]$. Rovnice (36) se přepíše takto:

$$\begin{bmatrix} Q_x \\ Q_y \\ Q_z \end{bmatrix} = R_k \begin{bmatrix} q_x \\ q_y \\ 0 \end{bmatrix} + T_k \quad (37)$$

$$\begin{bmatrix} R_1 & -E \\ R_2 & -E \\ \vdots & \vdots \\ R_N & -E \end{bmatrix} \cdot \begin{bmatrix} q_x \\ q_y \\ 0 \\ Q_x \\ Q_y \\ Q_z \end{bmatrix} = \begin{bmatrix} -T_1 \\ -T_2 \\ \vdots \\ -T_N \end{bmatrix} \Rightarrow A \cdot X = b \quad (38)$$

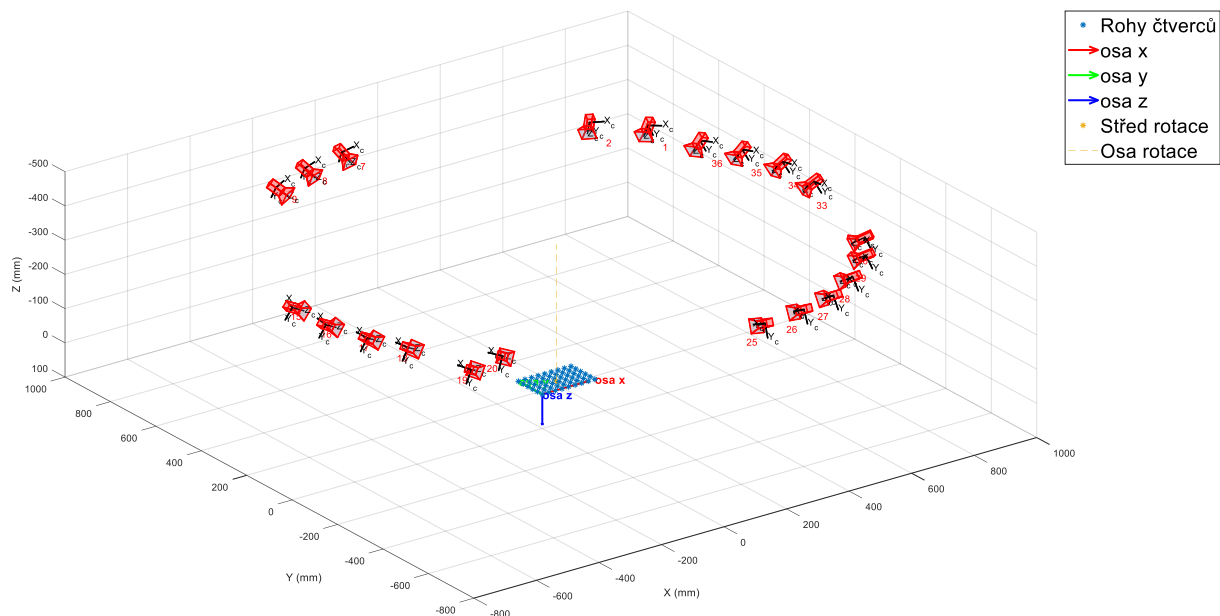
$$\hat{X} = \arg \min \|A \cdot X - b\| \quad (39)$$

Pořídil jsem 36 snímků stolku s kalibračním vzorem při různých natočeních, ale na některých z nich MATLAB nenašel šachovnice, proto byly odstraněny. Ze zbylých obrazů byly určeny jednotlivé R_k a T_k a sestavená funkce dle rovnice (39). Řešením úlohy optimalizace jsem získal polohu středu rotace v souřadnicovém systému kalibračního vzoru:

$$q = \begin{bmatrix} 78,8 \\ 48,1 \\ 0 \end{bmatrix} mm \quad (40)$$

Obr. 26 ukazuje určený střed rotace stolku a polohy kamery v MATLAB, ze kterých byly pořízeny snímky pro kalibraci. Na Obr. 27 je zobrazena poloha středu rotace v obrazu kamery a lze přibližně určit polohu bodu na kalibračním vzoru.

Ověření výsledku bylo provedeno pomocí tužky. Během rotace stolku jsem se dotýkal kalibračního vzoru a nakreslily se kružnice. Postupným přibližováním tužky k předpokládanému středu rotace se nakreslené kružnice zmenšovaly, až nevniknul bod – střed rotace. Jeho poloha byla velmi blízka vypočtené poloze z rovnice (40).



Obr. 26 – Střed rotace otočného stolku a polohy kamery v souřadnicovém systému kalibračního vzoru [autor]



Obr. 27 – Přibližná poloha středu rotace na kalibračním vzoru [autor]

3.6 Zpracování obrazu a určení polohy laserové čáry

Prvním krokem skenování je zpracování obrazu z kamery a nalezení na něm polohy promítnuté laserové čáry. Konkrétně chceme určit souřadnice pixelů laserové čáry, jejichž hodnoty budou později použity pro výpočet vzdálenosti k odpovídajícím bodům na objektu. Níže jsou uvedeny příklady algoritmu určení polohy laserové čáry na obrazu, které byly použity v jiných podobných projektech s využitím kamery a zdroje laserového záření.

3.6.1 Přehled algoritmů z podobných projektů

1) Design of a laser distance sensor with a web camera for a mobile robot (2016) [18]

V této práci Ashykhmin Mykhailo postavil přístroj skládající se z laserového zářiče a webkamery, který v reálném čase detekuje překážku a určuje k ní vzdálenost na základě obrazu z webkamery. Algoritmus je založen na určení polohy laserové čáry a využití laserové triangulace k výpočtu vzdálenosti.

V prvním kroku webkamera přístroje získává červený obraz, protože před objektivem je připevněn filtr červeného světla. Filtr slouží ke korekci a zesílení červené složky RGB obrazu pro lepší výsledek normalizace a následné segmentace obrazu, což podle autora má vliv na přesnost měření vzdálenosti. Dalším krokem je transformace červeného obrazu na obraz ve stupních šedi a získání normalizovaného obrazu, na kterém bude mít laserová čára pravděpodobně bílou (nejjasnější) barvu.

K nalezení laserové čáry na obrazu autor používá algoritmus “nejjasnější pixel v řádku”, který hledá nejjasnější pixely v každém řádku šedého obrazu a zvýrazňuje je. Výsledkem je binární obraz, ve kterém je laserová čára označena bílými pixely a ostatní místa černě, a matice souřadnic nejjasnějších pixelů vyjadřující polohu laserové čáry na tomto obrazu. Výhodou tohoto algoritmu je snížení vlivu velkých jasných objektů a zvýšení počtu správně detekovaných pixelů ve srovnání s falešnými.

Ashykhmin Mykhailo taky poznamenal, že existují jiné metody určování laserové čáry, například metoda “ocenění pozadí”, která využívá rozdíl mezi obrazy se zapnutým a vypnutým laserem.

Podle mého názoru, autor měl omezení ve výběru vhodného algoritmu zpracování obrazu. Za prvé, snímání, zpracování a výpočet vzdáleností do překážky probíhá v reálném čase, a proto autor potřeboval rychlý algoritmus. Za druhé, zdrojem laserového záření je laserová vodováha, která je dle autora napájena dvěma baterkami, tzn. že vodováhu nelze zapínat a vypínat řídicími signály, pouze manuálně, což v automatickém přístroji nedává smysl. Za třetí, přístroj je navržen pro upevnění na pohybujícího se robota, což znamená, že pozadí se pořád mění a při použití metody “ocenění pozadí” ve výsledném obrazu budou vznikat velké šумы.

2) 3D Scanning with Raspberry Pi and MATLAB (2016) [19]

V daném projektu Siddharth Vaghela postavil jednoduchý 3D skener, který se skládá z laserového modulu, kamery Pi, krokového motoru s driverem a řídicí jednotky Raspberry Pi.

Hlavním cílem autora bylo, aby celý systém skládající se z těchto hotových komponentů fungoval, nikoli aby dosáhnul co nejlepších výsledků.

Metodou 3D skenování je laserová triangulace a prvním krokem je opět nalezení laserové čáry. Autor nabízí pořídit dva snímky – jeden se zapnutým laserem a jeden s vypnutým laserem. Protože celé zorné pole kamery zůstává stejné, rozdíl těchto dvou obrazů dává polohu promítnuté laserové čáry. Jedná se právě o metodu “ocenění pozadí“, která již byla zmíněná v odstavci 1).

V dalších krocích autor převádí rozdílový obraz na binární podle prahu svítivosti a odstraňuje malé pixely, které vznikly binarizací obrazu a vnášely šum. Výsledkem je konečný čistý obraz, na kterém je laserová čára označena bílou barvou a ostatní pixely černou.

Tento projekt byl užitečný také v tom, že autor uvedl seznam použitých knihoven MATLAB, což zrychlilo hledání potřebných nástrojů pro zpracování obrazu z webkamery.

3) DIY: How to Make a Low-Cost 3D Scanner (2016) [6]

Daný návod popisuje tvorbu levného 3D skeneru na základě výzkumu, v němž skupina autorů nabízela první levný DIY laserový 3D skener s otevřeným zdrojovým kódem, který mohou členové laboratoře postavit a používat k ukládání digitálních souborů pro své výzkumné činnosti. Tento skener se skládá ze stejných základních komponent (laser, kamera, motor) a používá stejnou metodu laserové triangulace jako v probraném projektu 2).

K určení polohy laserové čáry autoři využívají kombinaci metod “ocenění pozadí“ a “nejjasnější pixel v řádku“ s dalšími algoritmy předběžného a následného zpracování obrazu, které dohromady jmenují “filter”. *“Pravděpodobně nejtěžší část tohoto skeneru a jediná věc, která doufám, bude fungovat i nadále, ale tento filtr vám pomůže získat pěknou jedinou laserovou čáru ve většině podmínek bez ohledu na osvětlení. Čím méně okolního světla, tím lépe (john1375, 2016).“*

Na rozdíl od algoritmu v bodě 2), v tomto projektu se pořízené snímky s a bez laserové čáry transformují do stupňů šedi před samotným rozdílem obrazů. Přeměna probíhá podle zvláštní rovnice (6), která podle mého názoru způsobuje podivné šedé obrazy (více na Obr. 33).

$$Img_{grey} = Img_R - \frac{Img_G + Img_B}{2} \quad (41)$$

Algoritmy zpracování obrazu v projektech 3) a 2) jsou téměř shodné, ale algoritmus v 3) je více zaměřen na předběžné a následné zpracování obrazu.

4) Obstacle detection using Laser and image processing on LOGI-Bone (2014) [20]

Další zajímavá metoda detekce laserové čáry, kterou autor použil ve svém robotu, je založená na jedné vlastnosti laserového záření – velká intenzita paprsku, která vytváří vysoký gradient na okrajích promítnuté laserové čáry. Algoritmus se podobá metodě “nejjasnější pixel v řádku“, jen místo hodnoty pixelu vypočítá gradient podél řádku obrazu a hledá maximální hodnotu gradientu v každém řádku obrazu. Výsledkem je gradientní obraz se zvýrazněnou

laserovou čarou. Autor poznamenal, že ačkoli systém celkově fungoval dobře, je citlivý na okolní světlo, odrazení scény a zrychlení robota.

Výše popsané příklady algoritmu určení laserové čáry jsou shrnuty do jedné tabulky, abychom mohli porovnávat jednotlivé kroky.

Tabulka 3: Kroky algoritmů určení laserové čáry z jiných projektů

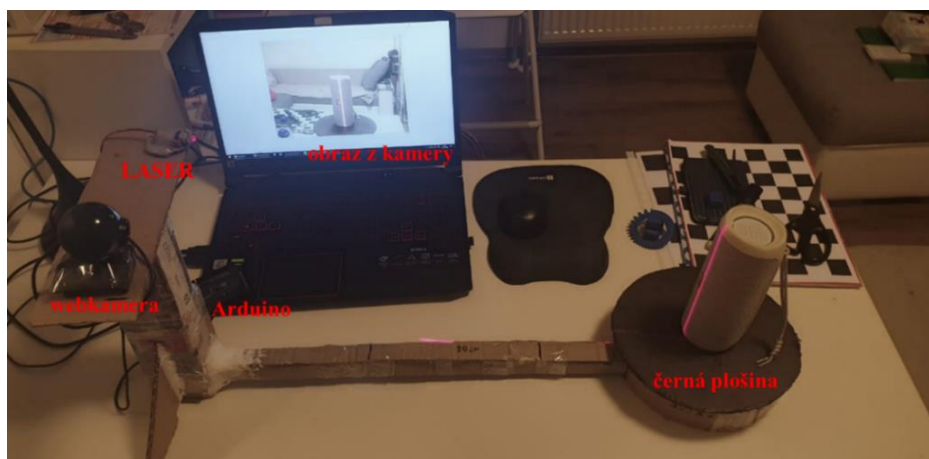
Číslo projektu	
1)	2)
1. Zachycení červeného obrazu s laserovou čarou (používá se červený filtr) 2. Transformace barevného obrazu do stupňů šedi 3. Vytvoření binárního obrazu vyhledáváním nejjasnějších pixelů v každém řádku obrazu	1. Zachycení dvou barevných obrazů s a bez laserové čáry 2. Rozdíl obrazů 3. Vytvoření binárního obrazu podle prahu svítivosti 4. Odstranění šumu (drobných objektů) z binárního obrazu
3)	4)
1. Zachycení dvou barevných obrazů s a bez laserové čáry 2. Transformace obrazů do stupňů šedi 3. Rozdíl šedých obrazů s a bez laserové čáry 4. Gaussův filtr 5. Vytvoření binárního obrazu vyhledáním nejjasnějších pixelů v řádku 6. Odstranění drobných objektů z binárního obrazu	1. Zachycení barevného obrazu s laserovou čarou 2. Transformace obrazu do stupňů šedi 3. Vypočet gradientu podél řádku obrazu 4. Určení laserové čáry vyhledáváním maximálního gradientu v každém řádku obrazu

3.6.2 Zkoušení cizích algoritmů

Testování algoritmů z Tabulka 3 probíhalo ještě před návrhem celé soustavy, proto jsem pro zjednodušení promítání laserové čáry a snímání objektu udělal testovou soustavu z kartonu, lepidla a izolepy (Obr. 28). Webkamera a laserový modul jsou upevněny na společné základně a nasměrovány k černé plošině, na které se umísťuje objekt. Laserový modul je napájen řídicí jednotkou Arduino, která je stejně jako webkamera připojená k počítači, na kterém běží řídicí program v MATLAB.

Pro zpracování a analýzu obrazů se používají knihovny *Image Processing Toolbox* a *Computer Vision Toolbox*. Tyto knihovny obsahují velké množství standartních algoritmů zpracování obrazů, jako například transformace RGB obrazu na obraz ve stupních šedi, Gaussův filtr, logaritmické a exponenciální transformace atd.

Obraz v MATLAB se popisuje trojrozměrným polem o velikosti rozlišení kamery (v našem případě 480 řádku a 640 sloupců) a hloubce 3, kde každý prvek matice odpovídá jednomu pixelu a každá hloubka představuje jednotlivý RGB kanál (červený, zelený, modrý) jako je zobrazeno na Obr. 29. Celkem je každá barva pixelu popsána třemi složkami: hodnoty červené, zelené a modré barvy v rozsahu od 0 do 255 (nebo od 0 do 1, závisí na druhu obrazu). Proto lze také používat běžné maticové a matematické příkazy na zpracování obrazů v MATLAB.



Obr. 28 – Testová soustava [autor]



Obr. 29 – Popis obrazu v Matlab [21]

Pro testování algoritmů jsem pořídil 11-12 snímků objektu při různém okolním osvětlení, z nichž některé jsou zobrazeny na Obr. 30. Tyto snímky budou zpracovány algoritmy z Tabulka 3 v MATLAB a hlavním výsledkem bude binární obraz, na kterém by měly bílé pixely označovat polohu laserové čáry a černé pixely zbytek obrazu. Samozřejmě kvůli nízkému rozlišení webkamery, různému osvětlení, povrchu objektu a dalším vlivům budou některé pixely špatně detekovány, ale počet správných pixelů bude charakterizovat kvalitu algoritmu.

Výstupem každého algoritmu je mozaikový obrázek (např. Obr. 31) skládající se ze čtyřech obrazů, z nichž každý zobrazuje výsledek vybraného kroku algoritmu s krátkým popisem v horní části mozaiky. Číslo v levém horním rohu ukazuje na číslo pořízeného snímku, který byl algoritmem zpracován. Pro stručnost budou zobrazeny jenom výsledky zpracování snímků z Obr. 30, ale hodnocení algoritmů bude probíhat na základě zpracování všech 12 snímků.



Obr. 30 – Příklad pořízených snímků (snímky číslo 2, 3, 5) [autor]

Příklad výsledků algoritmu 1) jsou zobrazeny na Obr. 31. Černobílé obrazy v pravých dolních rozích jsou výsledné binární obrazy pro každý pořízený snímek. Hlavní chybou, která byla udělaná před zpracováním obrazu, je, že při pořizování snímků nebyl použit filtr červeného světla, tzn. na rozdíl od práce v bodě 1) nebyla červená složka obrazu v testu zesílená. Ve výsledku dostáváme, že na jasných snímcích není v některých místech laserová čára nejjasnější částí, takže algoritmus “nejjasnější pixel v řádku” znázorňoval chybné pixely. Naproti tomu tmavší snímky byly lépe zpracovány a binární obrazy obsahovaly téměř celou laserovou čáru.

Příklad výsledků algoritmu 2) (“rozdíl obrazů”) jsou zobrazeny na Obr. 32. Daný algoritmus zpracoval většinu snímků dobře a prakticky nezávisel na okolním osvětlení. Podle popisu algoritmu v bodě 2) (nebo přehledněji v Tabulka 3) se rozdílový obraz transformuje na binární obraz podle prahu svítivosti, který je v intervalu 0 až 1. Změnou hodnoty jasnosti z tohoto intervalu lze měnit “tloušťku” laserové čáry a množství světla v pozadí, což ovlivňuje počet správně detekovaných pixelů. Na konci můžeme projít všemi řádky binárního obrazu, spočítat průměrnou hodnotu pixelu a získat finální obraz s tenkou spojitou laserovou čárou jako je zobrazeno v pravých dolních rozích na Obr. 32.

Příklad výsledků algoritmu 3) jsou zobrazeny na Obr. 33. Hlavním rysem daného algoritmu je rovnice (6), podle které probíhá přeměna barevných obrazů do stupňů šedi. Příklady šedých obrazu jsou uvedeny v levých horních rozích na Obr. 33 a je vidět, že rovnice potlačuje pozadí a zvýrazňuje popředí s laserovou čárou. Z testu vyplývá, že čím rovnoměrněji je objekt osvětlen přirozeným nebo žádným světlem, tím výraznější je oblast kolem laserové čáry. Ačkoli jsou algoritmy 3) a 2) podobné a používají stejnou metodu “ocenění pozadí” a navíc algoritmus 3) obsahuje příkazy pro předběžné a následné zpracování obrazu, je více závislý na okolním osvětlení a na binárních obrazech vznikají přerušovanější čáry než v algoritmu 2).

Příklad výsledků algoritmu 4) jsou zobrazeny na Obr. 34. Tento algoritmus se liší od předchozích tím, že hledá správné pixely podle hodnoty gradientu (změny intenzity světla) ve směru řádků obrazu. Daný algoritmus dobře pracuje s tmavými obrazy s homogenním pozadím a vytváří gradientní obraz (pravé dolní rohy na Obr. 34), na kterém je laserová čára s velkou pravděpodobností nejvíce zvýrazněná. Naopak při vysokém osvětlení se na gradientním obrazu vznikají skoky kvůli náhlé změně jasnosti v jiných místech obrazu (vznikají obrysy oblastí). To způsobuje šum a chybnou detekci pixelů laserové čáry.

Společný problém všech testovaných algoritmů (a zejména všech aktivních 3D skeneru) je neschopnost zpracovat obraz objektu se skleněnými nebo zrcadlovými plochy nebo s takovým osvětlením, že laserová čára je na objektu sotva viditelná.

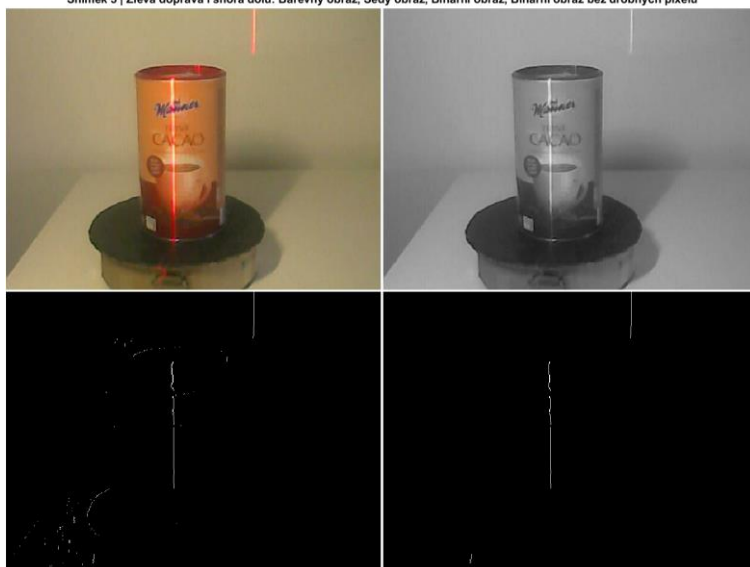
Snímek 2 | Zleva doprava i shora dolů: Barevný obraz, Šedý obraz, Binární obraz, Binární obraz bez drobných pixelů



Snímek 3 | Zleva doprava i shora dolů: Barevný obraz, Šedý obraz, Binární obraz, Binární obraz bez drobných pixelů

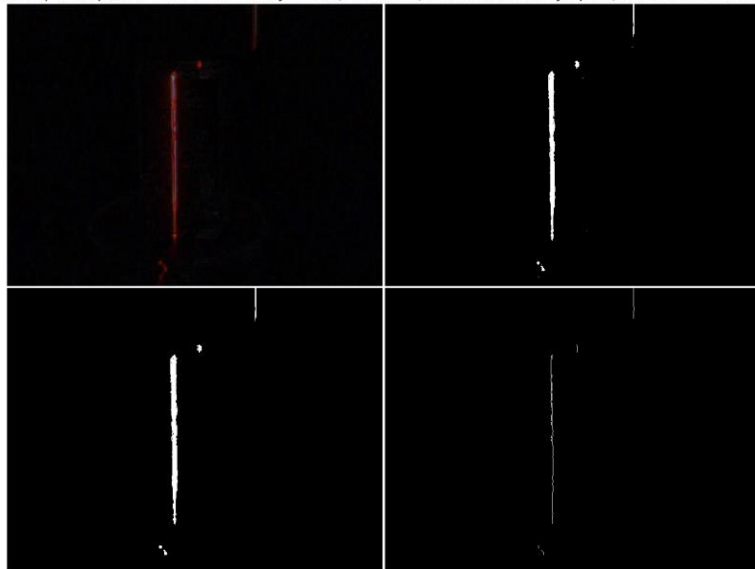


Snímek 5 | Zleva doprava i shora dolů: Barevný obraz, Šedý obraz, Binární obraz, Binární obraz bez drobných pixelů

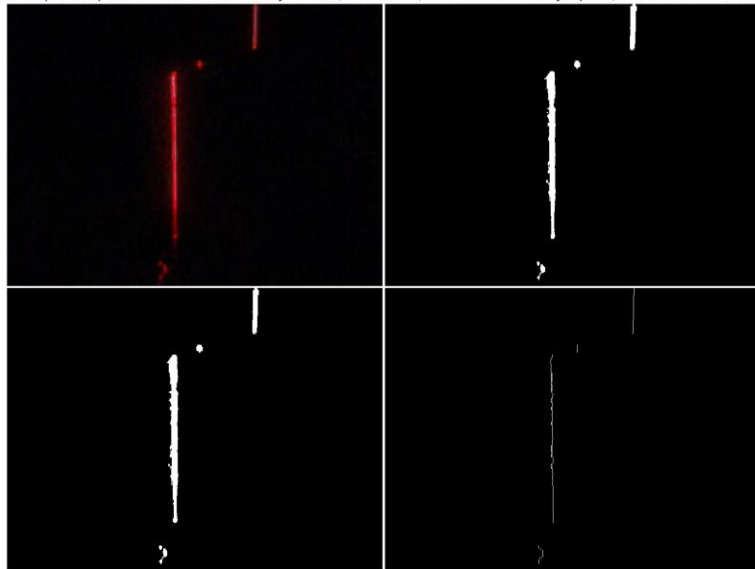


Obr. 31 – Výsledek algoritmu 1) při různém osvětlení objektu [autor]

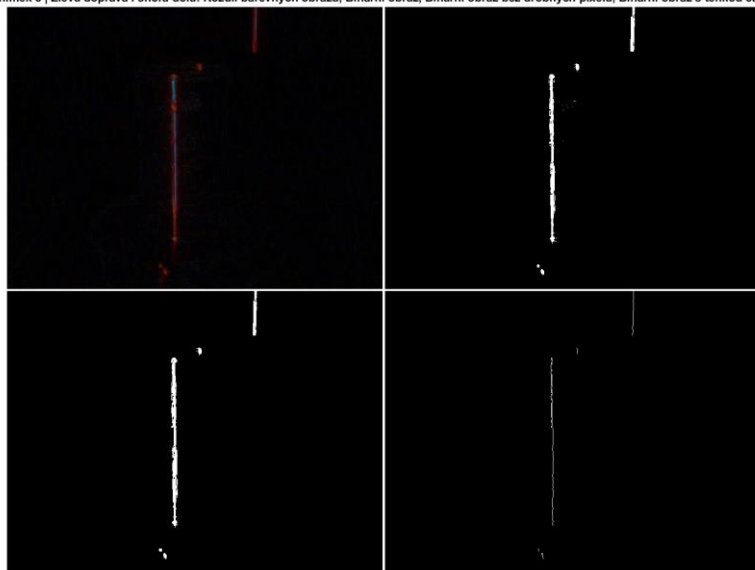
Snímek 2 | Zleva doprava i shora dolů: Rozdíl barevných obrazů, Binární obraz, Binární obraz bez drobných pixelů, Binární obraz s tenkou čarou



Snímek 3 | Zleva doprava i shora dolů: Rozdíl barevných obrazů, Binární obraz, Binární obraz bez drobných pixelů, Binární obraz s tenkou čarou

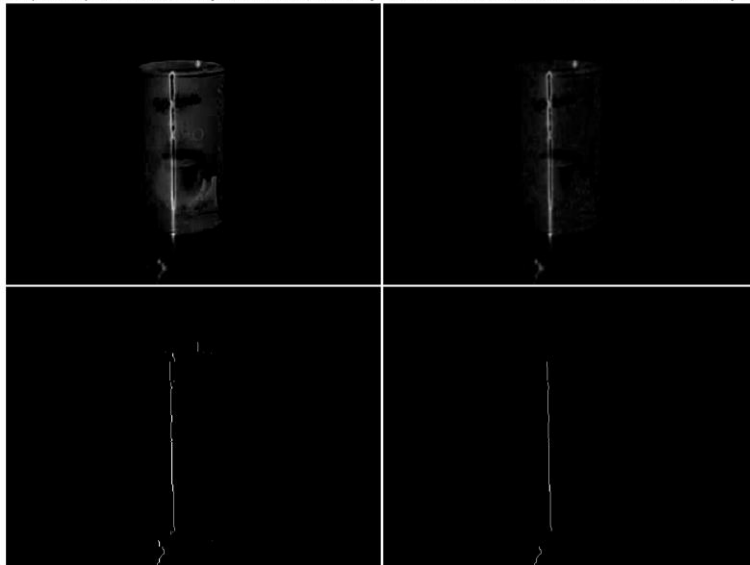


Snímek 5 | Zleva doprava i shora dolů: Rozdíl barevných obrazů, Binární obraz, Binární obraz bez drobných pixelů, Binární obraz s tenkou čarou

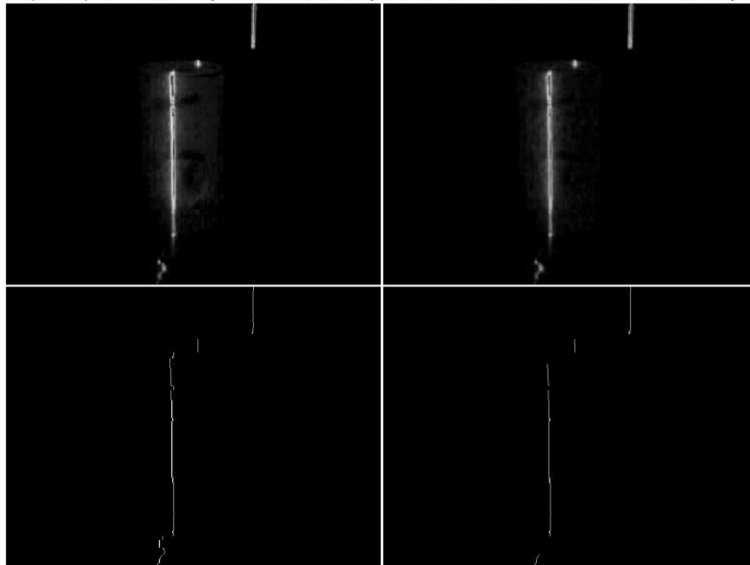


Obr. 32 – Výsledek algoritmu 2) při různém osvětlení objektu [autor]

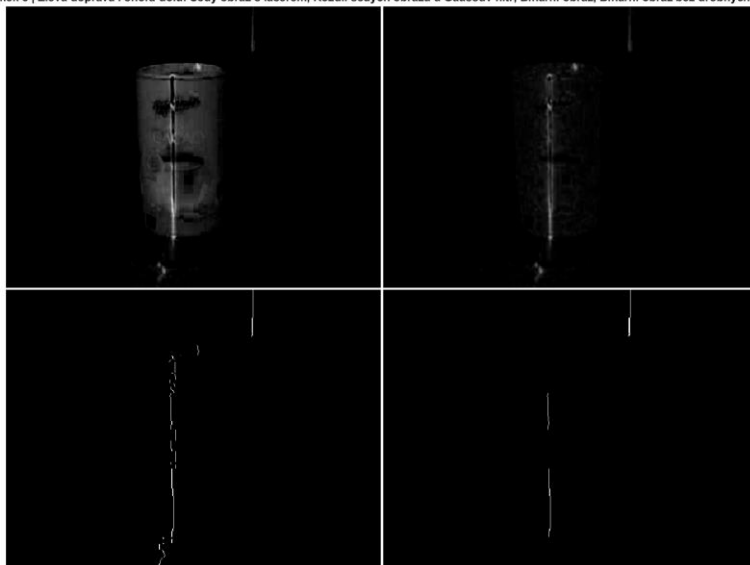
Snímek 2 | Zleva doprava i shora dolů: Šedý obraz s laserem, Rozdíl šedých obrazů a Gaussův filtr, Binární obraz, Binární obraz bez drobných pixelů



Snímek 3 | Zleva doprava i shora dolů: Šedý obraz s laserem, Rozdíl šedých obrazů a Gaussův filtr, Binární obraz, Binární obraz bez drobných pixelů



Snímek 5 | Zleva doprava i shora dolů: Šedý obraz s laserem, Rozdíl šedých obrazů a Gaussův filtr, Binární obraz, Binární obraz bez drobných pixelů

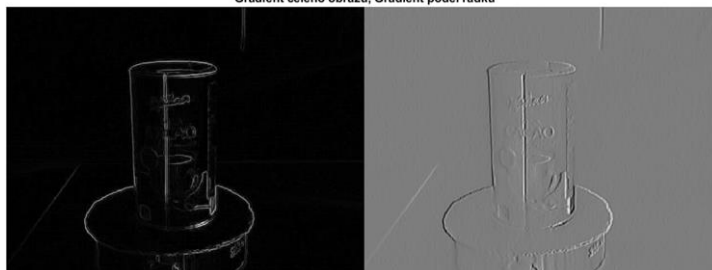


Obr. 33 – Výsledek algoritmu 3) při různém osvětlení objektu [autor]

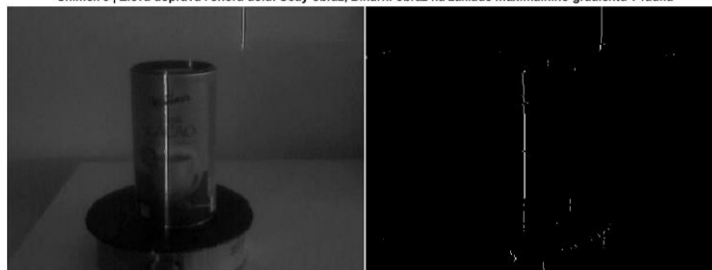
Snímek 2 | Zleva doprava i shora dolů: Šedý obraz, Binární obraz na zaklade maximálního gradientu v řádku



Gradient celého obrazu, Gradient podél řádku



Snímek 3 | Zleva doprava i shora dolů: Šedý obraz, Binární obraz na zaklade maximálního gradientu v řádku



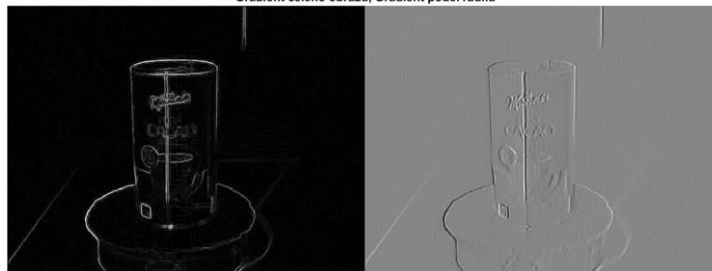
Gradient celého obrazu, Gradient podél řádku



Snímek 5 | Zleva doprava i shora dolů: Šedý obraz, Binární obraz na zaklade maximálního gradientu v řádku



Gradient celého obrazu, Gradient podél řádku



Obr. 34 – Výsledek algoritmu 4) při různém osvětlení objektu [autor]

3.6.3 Zkoušení cizích algoritmů s červeným filtrem

Světelný filtr je optické zařízení, které slouží k potlačení nebo zvýraznění části spektra elektromagnetického záření. Patří k základním příslušenstvím fotoaparátu, které chrání objektiv, potlačují nežádoucí reflexe a vytváří speciální barevné efekty ve výsledném obraze. Podle [22], foto filtry se dělí do následujících skupin:

- Protektory – neovlivňují reprodukci barev, ale chrání objektivy před nečistotami, nárazy a poškrábáním.
- UV filtry – pohlcují ultrafialové záření způsobující nevýrazný vzhled snímku.
- Polarizační filtry – vylepšují reprodukci barev, zvyšují sytost barev a kontrast.
- Barevné filtry – zvýrazňují konkrétní část barevného spektra na barevné fotografii a korigují převod barevných tonů do stupňů šedi na černobíle fotografii.
- Filtry pro zvláštní efekty

Naším cílem je zvýraznění červené laserové čáry na snímcích, aby algoritmus zpracování obrazu lépe hledal tuto promítnutou laserovou čáru na skenovaném objektu. K tomuto účelu bude užitečný červený barevný filtr, který bude umístěn před webkamerou a bude propouštět pouze červený spektrum a potlačovat nebo blokovat ostatní barvy ve scéně.

Dalšími účinky při použití červeného filtru jsou zvýšení kontrastu a viditelnosti objektů na snímku, snížení šumu a ovlivnění převodu jednotlivých barevných kanálů (R, G, B) do stupňů šedi. V testovaných algoritmech z Tabulka 3 jsou kroky s transformací barevného obrazu do stupňů šedi a filtr může zlepšit výsledky této transformaci.

Testování algoritmů (hlavně algoritmu číslo 1)) v části 3.6.2 probíhalo bez filtru červeného světla, protože v době testování nebyl k dispozici. Proto v části 3.6.3 jsou představeny výsledky testování stejných algoritmů z Tabulka 3 se stejným výstupním formátem (mozaikové obrázky), jenom byly zpracovány nové snímky objektu pořízené přes červené filtry při různém okolním osvětlení (příklad snímků je na Obr. 35).

Hlavním úkolem testování s červeným filtrem je správně překontrolovat algoritmus 1) a ocenit vliv filtru na výsledek jiných algoritmů, které v původních zdrojích filtr nepoužívaly.



Obr. 35 – Příklad pořízených snímků s červeným filtrem (snímky číslo b1, b2) [autor]

Příklad výsledků algoritmu 1) s červeným filtrem jsou zobrazeny na Obr. 36. V testování bez filtru nastával problém, že pixel promítnuté laserové čáry nemusel být nejjasnějším v řádku a algoritmus znázorňoval chybné pixely. Po použití filtru byla laserová čára detekována na všech

pořízených snímcích a výsledné binární obrazy obsahovaly spojitou čáru jako je zobrazeno v pravých dolních rozích na Obr. 36. Osvětlení mělo vliv pouze na množství šumu a délku získané čáry (konce laserové čáry mohly být uříznuty) na finálním obrazu.

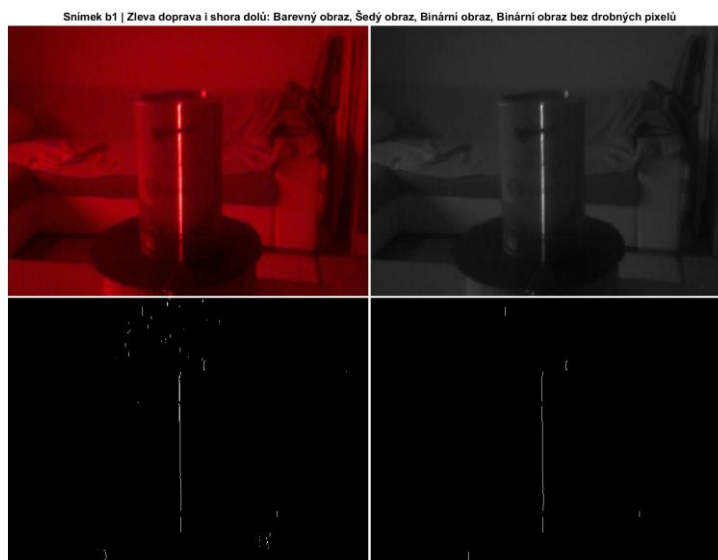
Příklad výsledků algoritmu 2) s červeným filtrem jsou zobrazeny na Obr. 37. Tento algoritmus dobře zpracoval snímky i bez filtru a prakticky nezávisel na okolním osvětlení. S červeným filtrem také dostáváme krásné laserové čáry a podle mého názoru s menším množstvím šumu než bez filtru. Předpokládám, že velký vliv na zpracování obrazu má vnitřní funkce MATLAB *im2bw*, která dobře transformuje rozdílový obraz podle prahu svítivosti na binární obraz a filtr prostě zlepšuje kontrast snímku.

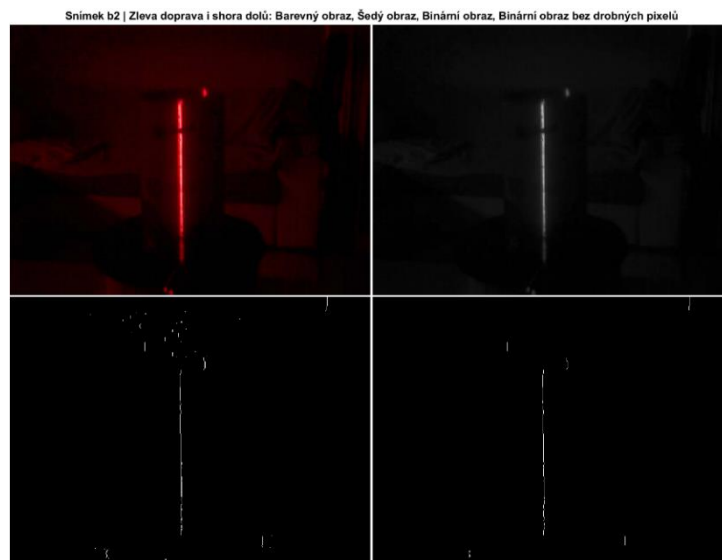
Příklad výsledků algoritmu 3) s červeným filtrem jsou zobrazeny na Obr. 20. Bez filtru byl daný algoritmus závislý na okolním osvětlení a dosahoval nejlepších výsledků při zpracování tmavých snímků, ale využití filtru zlepšilo zpracování světlých snímků. Většina nových binárních obrazů obsahují výrazné laserové čáry (Obr. 38), které jsou delší než ve výsledcích bez filtru (Obr. 33), ale více přerušované než v podobném algoritmu 2) s filtrem (Obr. 37).

Příklad výsledků algoritmu 4) s červeným filtrem jsou zobrazeny na Obr. 21. Problém této metody spočívá v tom, že náhlá změna jasnosti v jiných místech obrazu, než kolem laserové čáry vedla k chybnému určení pixelů laserové čáry. Díky účinku červeného filtru byla na gradientních obrazech laserová čára zvýrazněna a pozadí bylo potlačeno, i když nebylo homogenní. Ve výsledku dostáváme mnohem lepší binární obrazy se spojitou laserovou čarou než ve výsledcích bez červeného filtru (Obr. 34).

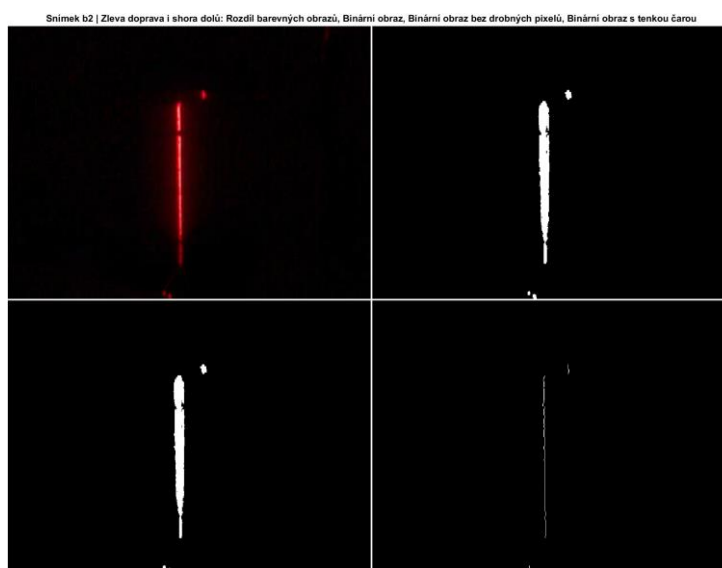
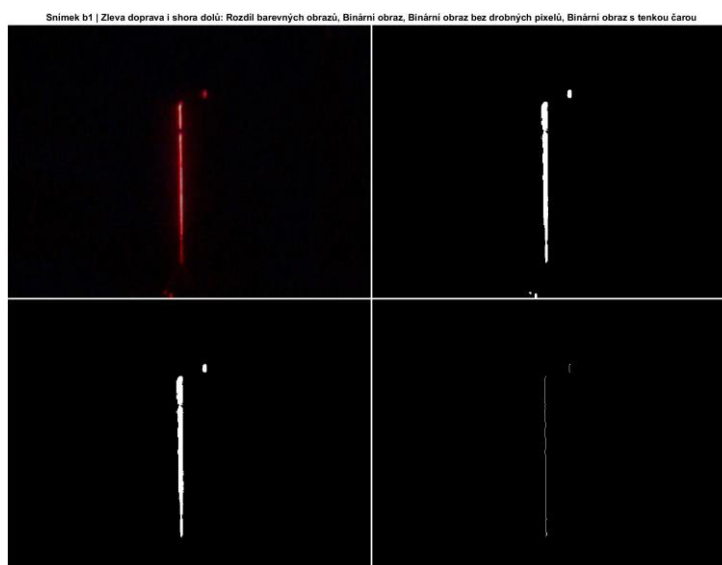
Největší zlepšení přinesl červený filtr v algoritmus 1) “nejjasnější pixel v řádku”, ve kterém měl být od začátku podle popisu algoritmu v bodě 1). Další zlepšení byla provedena v algoritmu 4) zvýrazněním laserové čáry v gradientním obrazu a potlačením obrysu v ostatních místech.

Ačkoli metoda určení laserové čáry podle hodnoty gradientu poskytuje dobré výsledky při použití červeného filtru, není podle mého názoru spolehlivá. Naopak algoritmy 1), 2), 3) jsou důvěryhodnější a použil bych je jako inspiraci k vytvoření vlastního algoritmu zpracování obrazu.



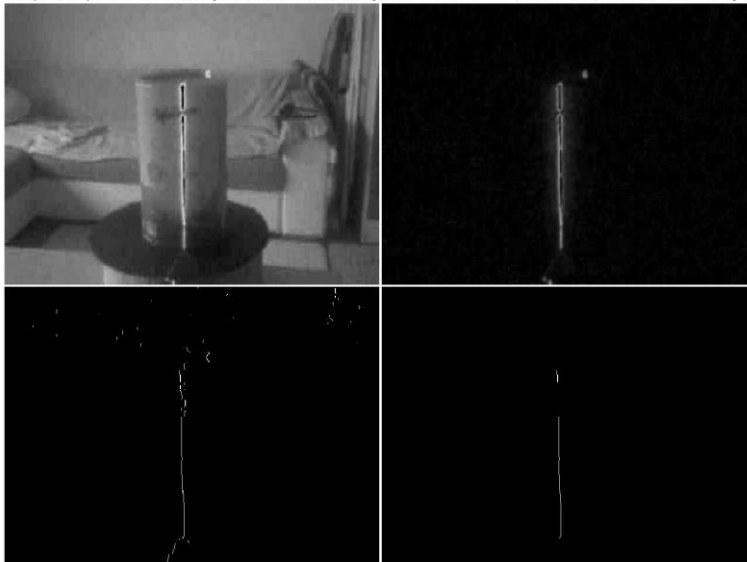


Obr. 36 – Výsledek algoritmu 1) s červeným filtrem při různém osvětlení objektu [autor]

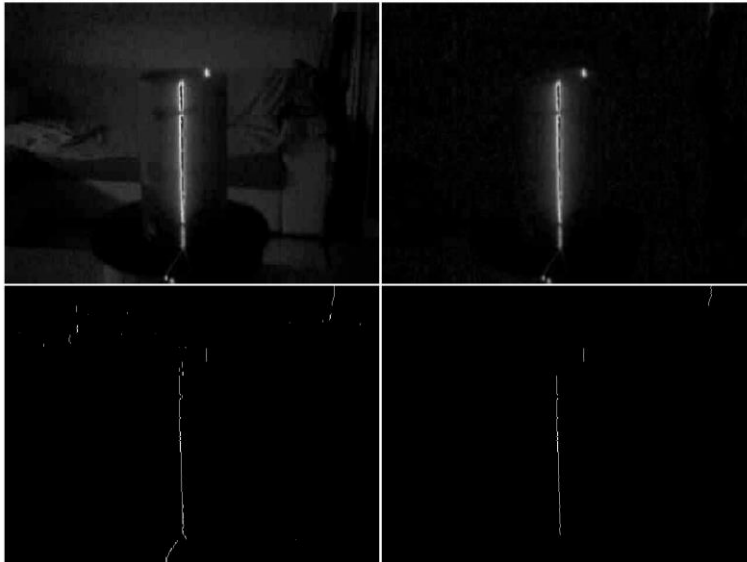


Obr. 37 – Výsledek algoritmu 2) s červeným filtrem při různém osvětlení objektu [autor]

Snímek b1 | Zleva doprava i shora dolů: Šedý obraz s laserem, Rozdíl šedých obrazů a Gaussův filtr, Binární obraz, Binární obraz bez drobných pixelů

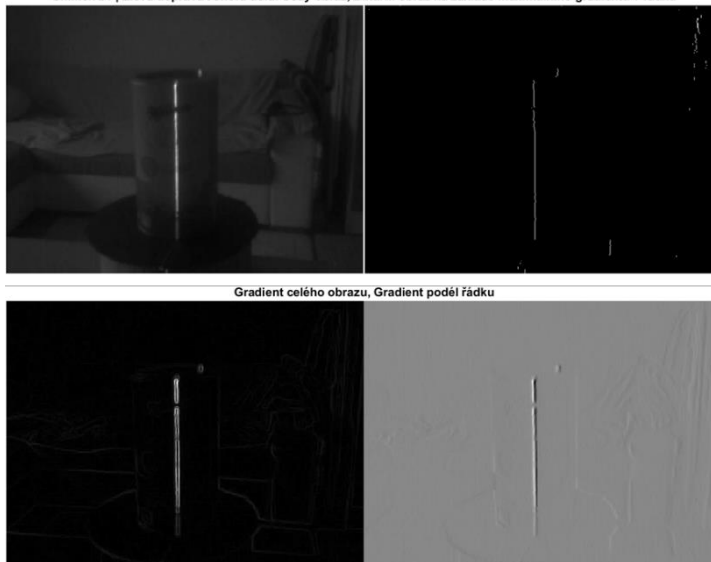


Snímek b2 | Zleva doprava i shora dolů: Šedý obraz s laserem, Rozdíl šedých obrazů a Gaussův filtr, Binární obraz, Binární obraz bez drobných pixelů

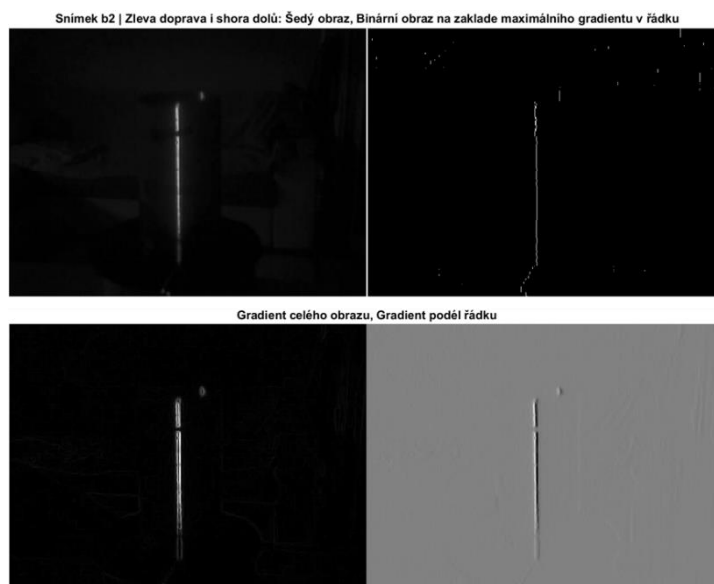


Obr. 38 – Výsledek algoritmu 3) s červeným filtrem při různém osvětlení objektu [autor]

Snímek b1 | Zleva doprava i shora dolů: Šedý obraz, Binární obraz na zaklade maximálního gradientu v řádku



Gradient celého obrazu, Gradient podél řádku



Obr. 39 – Výsledek algoritmu 4) s červeným filtrem při různém osvětlení objektu [autor]

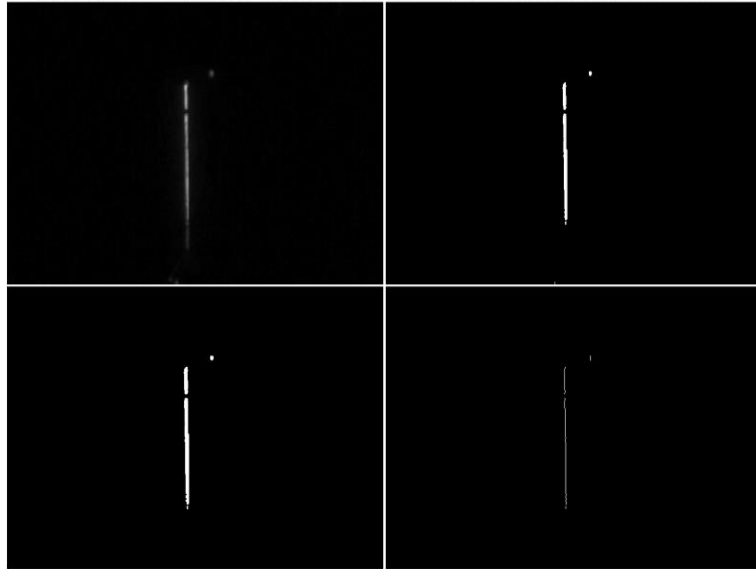
3.6.4 Návrh a zkoušení vlastního algoritmu

Příklad výsledku mého algoritmu určení laserové čáry jsou zobrazeny na Obr. 40. Jako základ jsem zvolil metodu “ocenění pozadí” (též “rozdíl obrazů”), protože je méně závislá na okolním osvětlení a mám možnost zapínat a vypínat laserový modul. Další část algoritmu je kombinace příkazů z algoritmů 2) a 3) a má následující kroky:

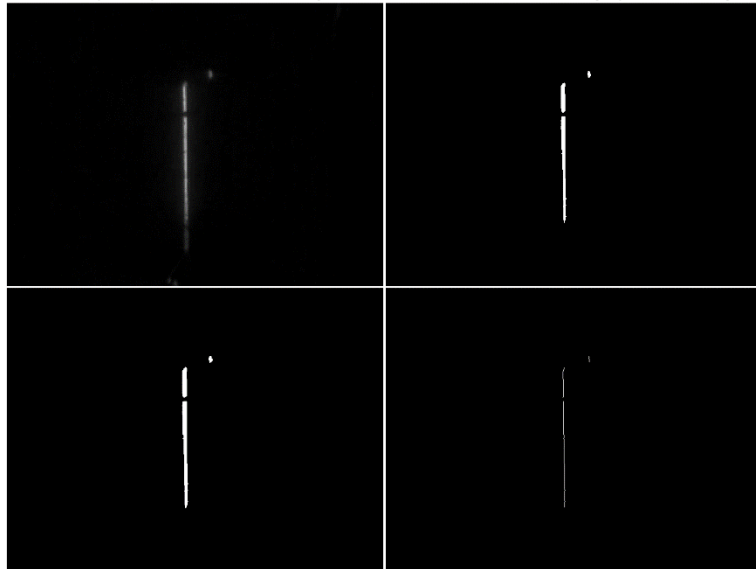
1. Zachycení dvou barevných obrazů s a bez laserové čáry
2. Gaussův filtr
3. Transformace obrazů do stupňů šedi
4. Rozdíl šedých obrazů
5. Vytvoření binárního obrazu podle prahu svítivosti
6. Odstranění drobných pixelů z binárního obrazu
7. Ztenčení laserové čáry

Algoritmus vykazuje dobré výsledky jak bez červeného filtru při slabém osvětlení, tak s filtrem při jakémkoli osvětlení. Avšak může nastat problém při skenování objektu složitějšího tvaru, protože při testování cizích algoritmů a vlastního jsme zpracovávali obrazy s rovnou laserovou čarou, která se nedeformovala na válcovém objektu. Při testování 3D skeneru může být nutná doplňková modifikace algoritmu a jeho příkazů.

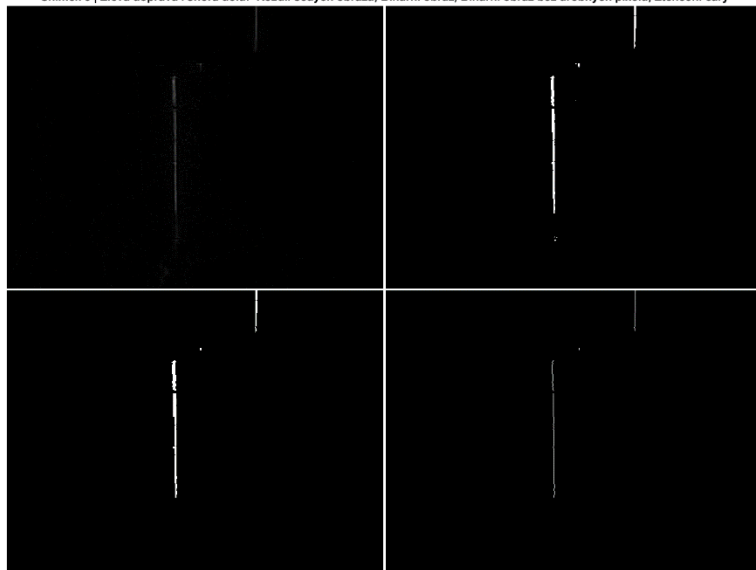
Snímek b1 | Zleva doprava i shora dolů: Rozdíl šedých obrazů, Binární obraz, Binární obraz bez drobných pixelů, Ztenčení čáry



Snímek b2 | Zleva doprava i shora dolů: Rozdíl šedých obrazů, Binární obraz, Binární obraz bez drobných pixelů, Ztenčení čáry



Snímek 5 | Zleva doprava i shora dolů: Rozdíl šedých obrazů, Binární obraz, Binární obraz bez drobných pixelů, Ztenčení čáry



Obr. 40 – Výsledek navrženého algoritmu zpracování obrazu pro snímky b1, b2 (s filtrem) a snímek číslo 5 (bez filtru) [autor]

3.7 Návrh řízení

Prvním nápadem bylo vytvořit 3D skener bez nutnosti externího připojení k počítači. Takový skener by byl řízen jednodeskovým počítačem Raspberry Pi a výsledky skenování by se ukládaly na externí úložiště. Avšak kvůli pandemii Covid-19 a výpadku dodávek polovodičů se ceny tohoto mikropočítače výrazně zvýšily a někdy nebyl k dispozici vůbec. Proto bylo rozhodnuto použít vlastní notebook a prostředí MATLAB k psaní hlavního kódu. Kromě toho mám více zkušeností s programováním v MATLAB a vnitřní knihovny usnadnily psaní kódu.

Řízení 3D skeneru se skládá ze dvou částí – na počítači v MATLAB běží hlavní řídicí program, který ovládá USB kameru, zpracovává z ní přijaté obrazy, posílá příkazy do Arduino a zobrazí výsledky skenování. Arduino zase ovládá krokový motor nebo laserový modul podle poslaného z počítače příkazu. Komunikace mezi počítačem a Arduino probíhá přes kabel USB A-B, který Arduino také napájí.

3.7.1 Řízení krokového motoru a laserového modulu

V dané kapitole se popisuje řídicí část na Arduino, která je ovládaná příkazy z počítače po kabelu USB A-B, jejichž princip komunikace je popsán v 3.7.2. Na Obr. 41 je ukázána schéma zapojení krokového motoru NEMA 17, driveru krokového motoru TB6600 a laserového modulu k mikrokontroléru Arduino MEGA 2560. Popis jednotlivých zapojení je uveden v Tabulka 4.

Tabulka 4: Zapojení v řídicí části Arduino

Název pinu v driveru	Vyznám	Zapojení v Arduino
ENA+ / ENA-	zapnutí a vypnutí driveru i když je napájen	D35 / zem
DIR+ / DIR-	nastavení směru otáčení	D33 / zem
PUL+ / PUL-	pulzní signály k driveru (1 pulz = 1 krok)	D31 / zem
A+ / A-	první vinutí krokového motoru	
B+ / B-	druhé vinutí krokového motoru	
VCC / GND	napájení driveru od zdroje	
	napájení laserového modulu od Arduino	D7 / zem

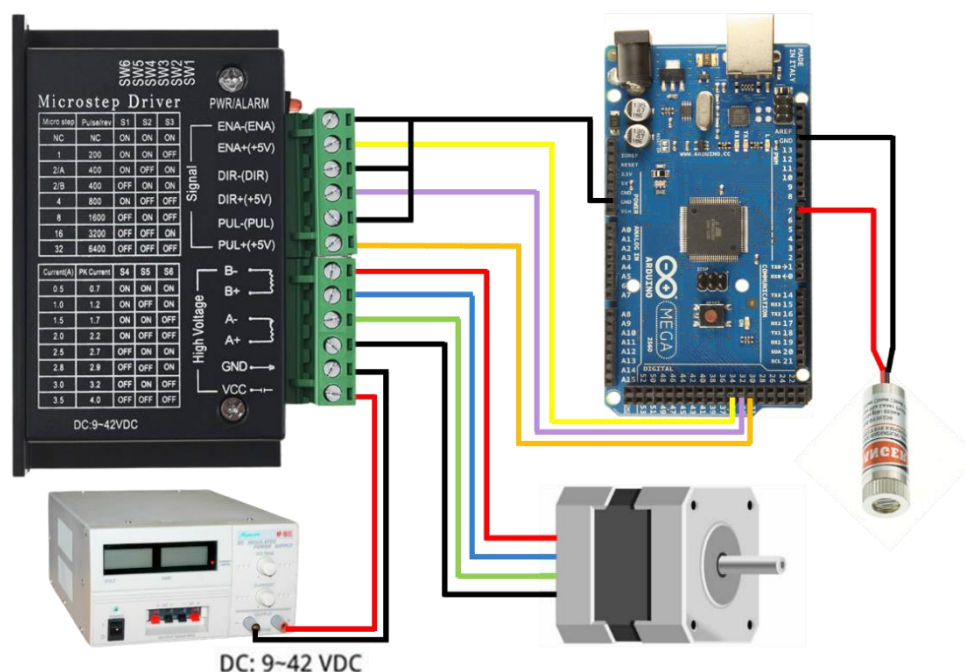
Zdroj: Vytvořeno na základě tabulky z [12]

Ovládání laserového modulu je jednoduché a provádí se přivedením napětí na připojený pin. Řízení krokovým motorem je trochu složitější a spočívá v posílání pulzních signálů z Arduino na vstup PUL+ driveru, které se následně přemění na proud vinutími motoru. Počet aplikovaných impulsů určuje velikost úhlu natočení rotoru a frekvence pulzů určuje jeho rychlost.

Hlavní funkcí tohoto driveru je nastavení mikrokroku a velikosti proudu protékajícího vinutími motoru. To se provádí zapínáním nebo vypínáním jednotlivých prepínačů umístěných na boku driveru (pod grafickým označením prepínačů, SW1 - SW6 v horní části driveru na Obr. 41). Z vlastností motoru v 3.2.3 je jeho jmenovitý proud 1,7A, takže pro normální provoz je driver nastaven na danou hodnotu proudu podle tabulky na krytu, tj. SW4: ON, SW5: ON, SW6: OFF.

Mikrokrokování umožňuje rozdělit jmenovitý krok motoru na další kroky prostřednictvím změny poměru proudů ve fázích motoru. Výhodou tohoto režimu je zvýšení rozlišení polohování, hladší pohyb rotoru, nižší vibrace, hluku a opotřebení celé mechanické soustavy. Nevýhodou je

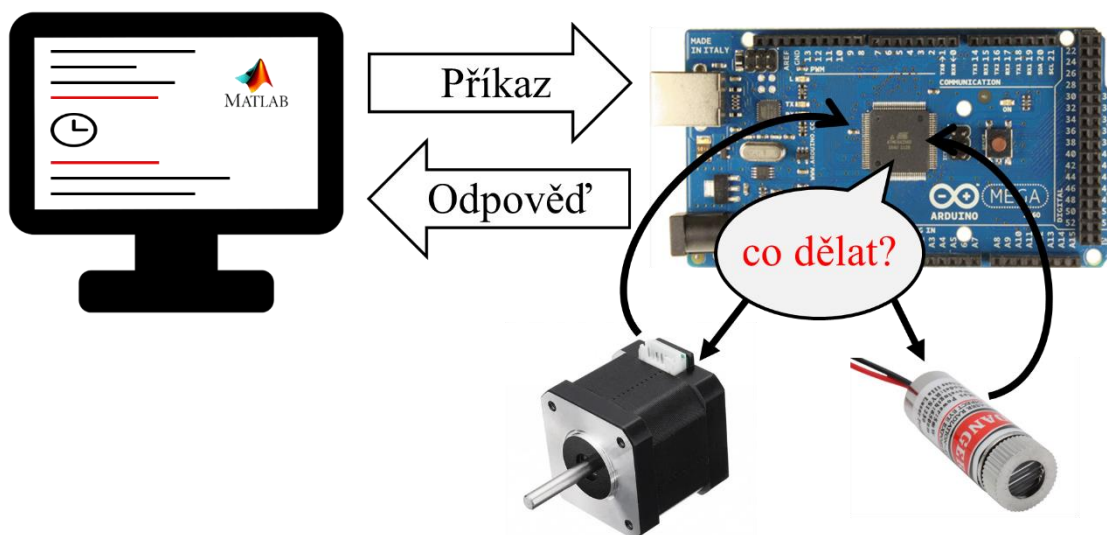
snížení statického momentu motoru, což může mít za následek, že při velkém mechanickém odporu se motor nemusí roztočit. Proto velikost mikrokroku zvolím při zkoušení cele soustavy.



Obr. 41 – Schéma řízení krokového motoru a laserového modulu [autor]

3.7.2 Komunikace mezi počítačem a Arduino

Původně bylo plánováno komunikovat mezi počítačem a Arduino pomocí rozšíření *MATLAB Support Package for Arduino Hardware*, protože zjednodušovalo interakci s deskou Arduino a nevyžadovalo dlouhé nastavování. V našem případě stačilo jednoduše se připojit k Arduino přes MATLAB a pomocí příkazu `writeDigitalPin(arduino, 'D11', 0)` s názvem připojené desky, názvem pinu a hodnotou pinu zapsat data na digitální pin. Dané rozšíření dobře fungovalo při zapínání a vypínání laserového modulu, protože jeho digitální pin dostával pouze hodnoty 1 nebo 0. Potíže nastaly při ovládní krokového motoru, protože driver krokového motoru, jak bylo popsáno v 3.7.1, přijímá z Arduino pulzní signál s určitou frekvencí. V MATLAB to bylo realizováno postupným použitím příkazu `writeDigitalPin()` s hodnotami 0 a 1 s nějakou frekvencí. Během zkoušek bylo zjištěno, že frekvence pulzních signálů na vstup driveru krokového motoru byla nižší než frekvence příkazů z počítače. V důsledku toho se krokový motor otáčel mnohem pomaleji, než se očekávalo, a vznikal hluk. S tímto problémem se setkala mnoho lidí a hlavním důvodem je, že MATLAB je příliš pomalý na provádění vysokofrekvenčních operací. Řešením je přenos řízení krokového motoru na lokální úroveň, tzn. kód pro ovládní motoru je na Arduino a k jeho provedení dojde až po obdržení příslušného příkazu z MATLAB.



Obr. 42 – Sériová komunikace mezi počítačem a Arduino [autor]

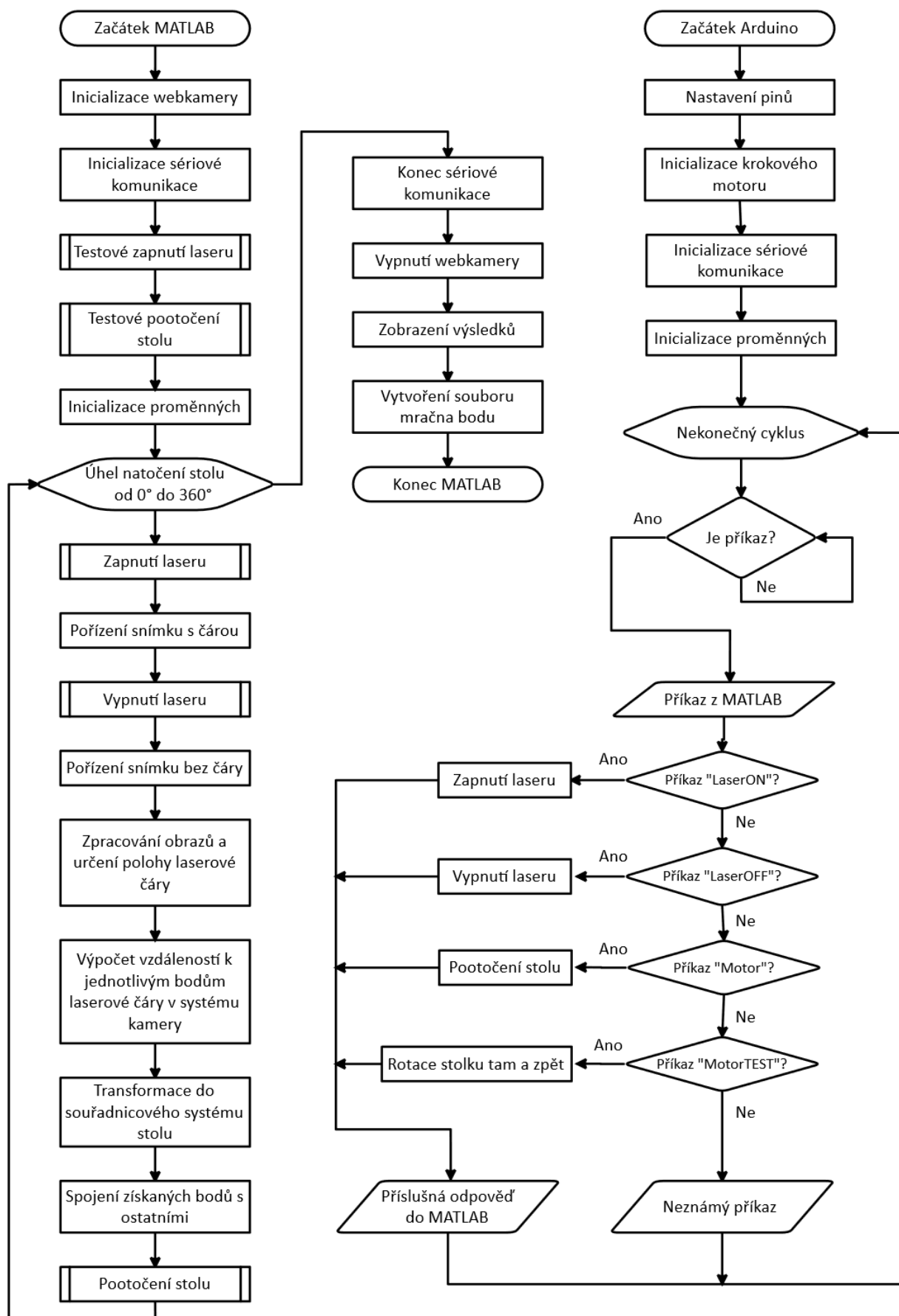
Výměna informací mezi počítačem a Arduino probíhá prostřednictvím sériové linky, jejíž princip činnosti je zobrazen na Obr. 42. Hlavní program posílá příkaz po sériové lince přes kabel USB A-B do Arduino a nepokračuje dál, dokud nedostane očekávanou odpověď zpět. Arduino přijímá informace a porovnává ji s připravenými vnitřními příkazy a tím rozhoduje, co dělat. Po provedení akce Arduino posílá odpovídající odpověď zpět do hlavního kódu, a pokud se shoduje s očekávanou, tak program MATLAB pokračuje dál. V případě chybného příkazu Arduino vrací chybu a hlavní kód se přerušuje. Seznam příkazů a odpovědí je uveden v Tabulka 5.

Tabulka 5: Seznam příkazů a odpovědí při sériové komunikaci [autor]

Příkaz	Odpověď
“LaserON”	“Zapnutí Laseru na pinu...”
“LaserOFF”	“Vypnutí Laseru na pinu...”
“ON”	“Zapnutí LED”
“OFF”	“Vypnutí LED”
“Motor”	“Pootocení motoru”
“MotorTEST”	“Testové pootocení motoru”
Neznámý příkaz	“Neznámý příkaz”

3.7.3 Výsledný program

Vývojový diagram řídicího programu 3D skeneru je znázorněn na Obr. 43. Jak už bylo zmíněno, program se skládá z částí MATLAB a Arduino. Obdélník s čárami po stranách označuje spuštění podprogramu na Arduino podle principu popsaného v 3.7.2. Výsledkem programu je soubor mračna bodů skenovaného objektu.

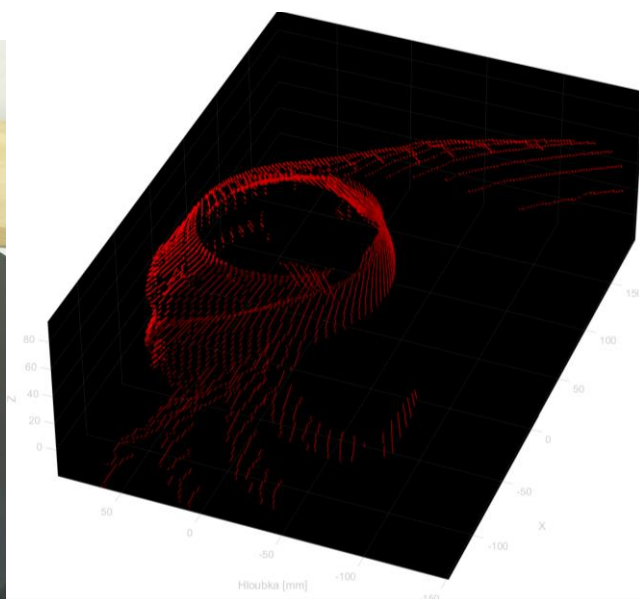
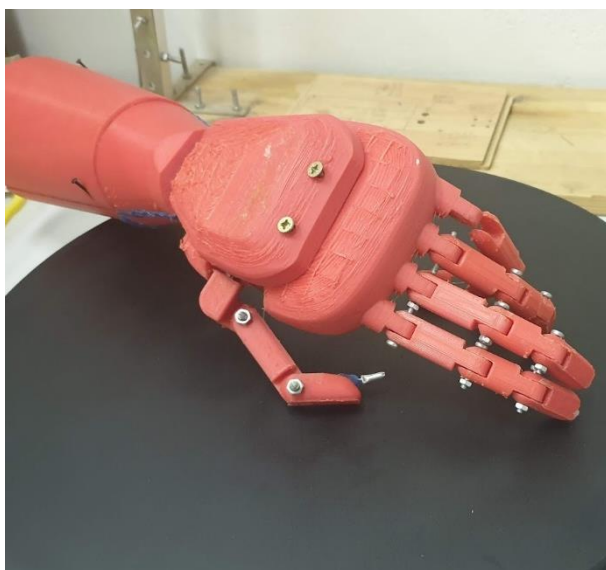
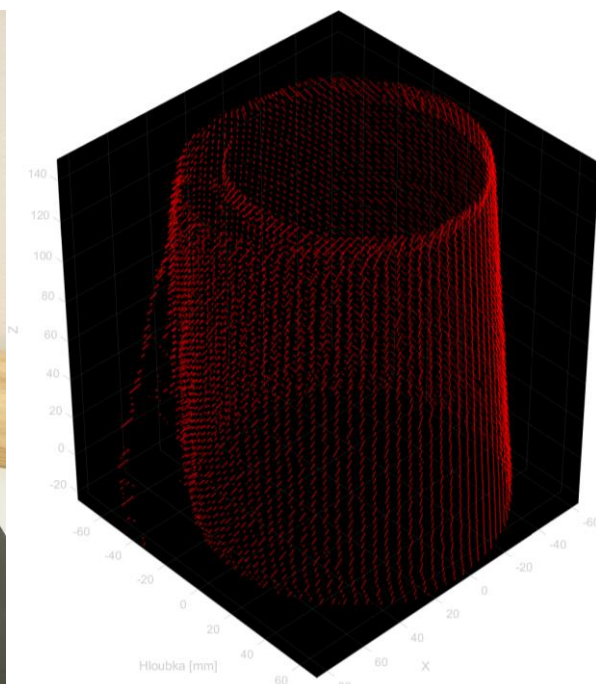


Obr. 43 – Vývojový digram řídicího programu 3D skeneru [autor]

3.8 Experimentální ověření 3D skeneru

Hlavním úkolem navrženého 3D skeneru je správné naskenování objektu a vytvoření souborů s naměřenými body. K testování jsem použil dva objekty: válcový zvukový reproduktor a mechanickou ruku. Dané objekty neměly výchozí 3D modely, takže jsem nemohl porovnat skeny s ideálními tvary, proto jsem použil měřicí pásmo a data z 3D grafů.

Výsledná mračna bodů sledují tvar objektů, ale jsou poněkud zkreslená kvůli nepřesnosti skeneru. Rozlišení skeneru není dostatečné pro zobrazení například tlačítek zvukového reproduktoru. Postavený cíl ± 1 mm se nepodařilo splnit, ale bych řekl, že výsledky jsou uspokojivé.



Obr. 44 - Výsledky skenování: skenované objekty a získaná mračno bodů, zvukový reproduktor (nahore) a mechanická ruka (dole) [autor]

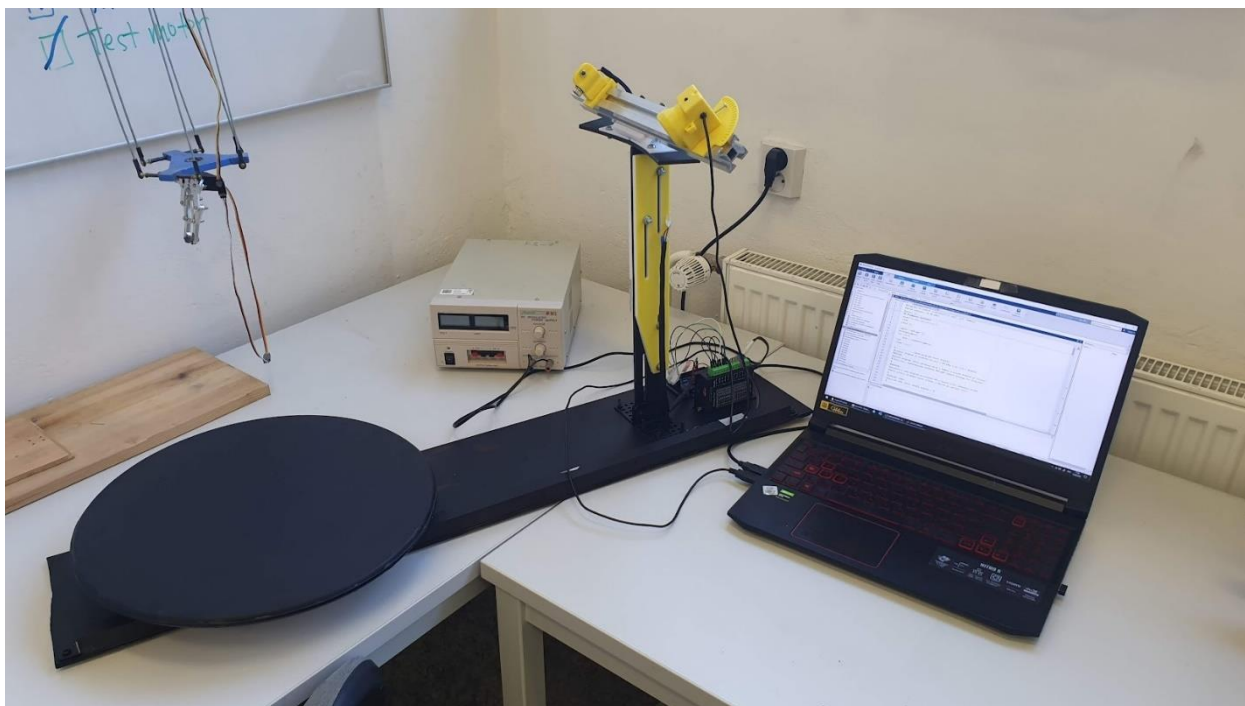
4. Závěr

V dané bakalářské práci jsem navrhl a postavil stolní bezkontaktní 3D skener fungující metodou laserové triangulace. Dále jsem realizoval řídicí program v MATLAB na zpracování obrazů a podprogram na ovládní krokového motoru a laserového modulu v Arduino. Komunikace mezi nimi byla realizovaná pomocí sériové komunikace, což odstranilo chyby s ovládním krokového motoru přímo z MATLAB. Výsledkem skenování je 3D graf v MATLAB naskenovaného objektu a soubor mračna bodů, které lze dále použít v speciálních softwarech na vytvoření CAD modelů.

V rámci této práce jsem se naučil ovládat krokový motor a používat 3D tiskárnu k tištění potřebných součástí, seznámil jsem se s metody zpracování obrazu a prohloubil jsem své znalosti programování.

3D skener umožňuje skenování těžkých objektu a má nastavitelné součásti, které lze přizpůsobit v závislosti na velikosti objektů. Bohužel kvůli poslední vlastnosti je nutná recalibrace 3D skeneru, která zhruba trvá hodinu kvůli velkému množství ruční práce. Samotný proces skenování trvá 8-12 minut v závislosti na velikosti kroku skenování.

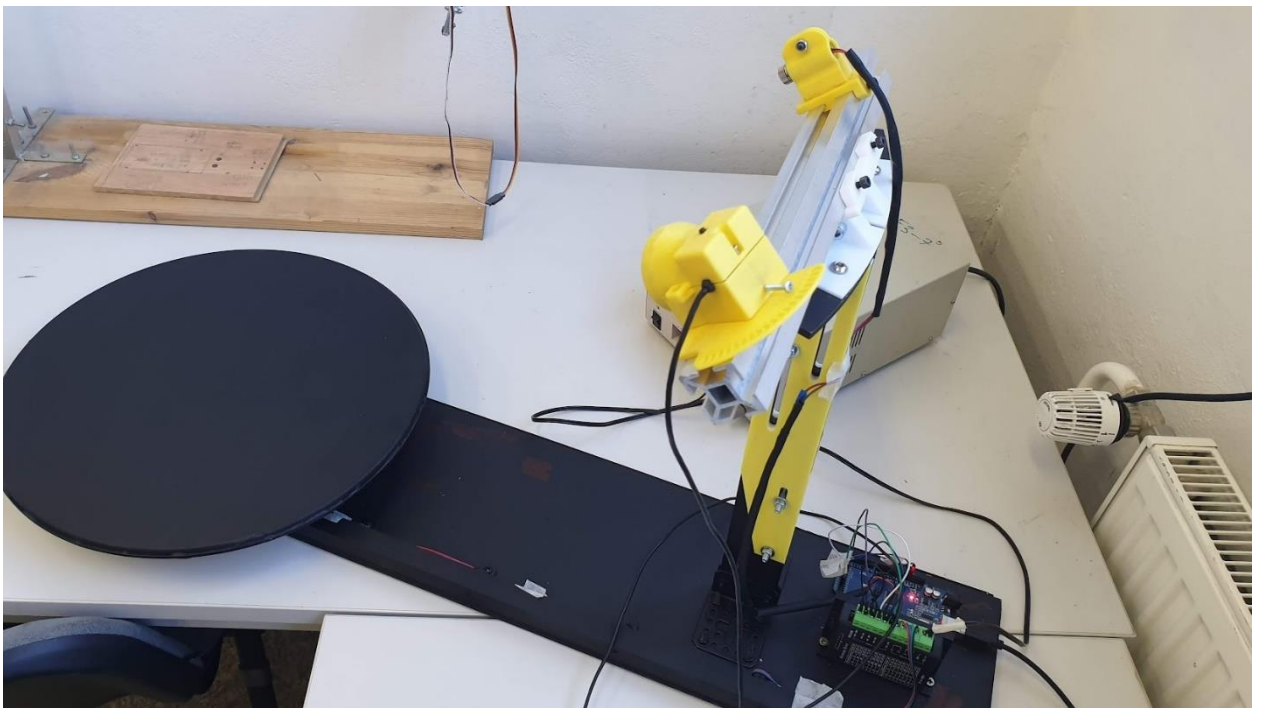
Dalšími nedostatky 3D skeneru jsou hluk motoru, vibrace otočného stolku kvůli nepřesnému tisku šnekového kola a deformace tvaru výsledného modelu kvůli nepřesné transformaci souřadnic. Chyby lze odstínit postavením lepší konstrukce a použitím pokročilejších algoritmů kalibrace.



Obr. 45 – Postavený 3D skener a hlavní kód na počítači [autor]



Obr. 46 - Skenovací hlava, pohled 1 [autor]



Obr. 47 – Skenovací hlava, pohled 2 [autor]

Seznam použité literatury

- [1] KAZAKEVICH, Georgij. Otrastli primenenija 3D-skanerov: Mashinostroenie. In: *I3d.ru* [online]. Moskva: i3D, ©2020 [cit. 2023-03-06]. Dostupné z: https://i3d.ru/blog/dlya_mozayki/mashinostroenie/
- [2] MENDŘICKÝ, Radomír a Petr KELLER. 3D skenery pro použití v průmyslu. *Automa*. Automa-časopis pro automatizační techniku, 2015, 2015(12), 34-37. ISSN 1210-9592.
- [3] BALASHOV, Aleksandr. Principy raboty 3D-skanera. Vidy skanerov, tehnologii i metody skanirovaniya. In: *Cvetmir3d.ru* [online]. Moskva, 16.04.2021 [cit. 2023-03-06]. Dostupné z: <https://cvetmir3d.ru/blog/poleznoe/printsipy-raboty-3d-skanera-vidy-skanerov-tekhnologii-i-metody-skanirovaniya/>
- [4] KAMANI, Karan. Types of 3D Scanning Technologies and 3D Scanners. In: *Linkedin.com* [online]. Sunnyvale: LinkedIn, © 2023, Jun 29, 2020 [cit. 2023-03-06]. Dostupné z: <https://www.linkedin.com/pulse/types-3d-scanning-technologies-scanners-karan-kamani>
- [5] How to Choose the Best 3D Scanner to Use With Your 3D Printer. In: *Formlabs.com* [online]. Somerville, MA: Formlabs, ©2023 [cit. 2023-03-06]. Dostupné z: <https://formlabs.com/blog/how-to-choose-a-3d-scanner-for-3d-printing/>
- [6] LOMBA, John. DIY: How to Make a Low Cost 3D Scanner. In: *Instructables.com* [online]. Autodesk, Inc., © 2023, 2016-07-17 [cit. 2023-08-15]. Dostupné z: <https://www.instructables.com/3D-Laser-Scanning-DIY/>
- [7] PAVLIDIS, George. Laser scanning using a laser pointing device and a camera [online]. 2006. Xanthi, 2006 [cit. 2023-08-15]. Dostupné z: <https://georgepavlides.info/research/LaserScanningAndTriangulation.php>
- [8] What Is Camera Calibration? In: *Mathworks.com* [online]. Natick, © 1994-2023 [cit. 2023-08-15]. Dostupné z: <https://www.mathworks.com/help/vision/ug/camera-calibration.html>
- [9] Camera Calibration and 3D Reconstruction. In: *Docs.opencv.org* [online]. [cit. 2023-08-15]. Dostupné z: https://docs.opencv.org/3.4/d9/d0c/group_calib3d.html#:~:text=figure%20illustrates%20the-,pinhole%20camera%20model,-
- [10] Laserový modul s optikou vykreslující čáru - Červený, 4~6mW 12mm 3 - 5V. In: *Dratek.cz* [online]. Havlíčkův Brod [cit. 2023-08-15]. Dostupné z: <https://dratek.cz/laserove-moduly/35-laserovy-modul-cervený-4-6mw-12mm-5v-s-optikou-vykreslující-caru.html>
- [11] Krokový motor NEMA 17 17HS8401 0,5Nm. In: *Laskakit.cz* [online]. © 2023 [cit. 2023-03-05]. Dostupné z: <https://www.laskakit.cz/krokovy-motor-nema-17-17hs8401-0-5nm/>
- [12] BENNE. TB6600 Stepper Motor Driver with Arduino Tutorial. In: *Makerguides.com* [online]. © 2023, 2019-10-04 [cit. 2023-03-05]. Dostupné z: <https://www.makerguides.com/tb6600-stepper-motor-driver-arduino-tutorial/>

- [13] ORIENTAL MOTOR GENERAL CATALOGUE: Technical Reference: Selection Calculations [katalog výrobků]. In: *Orientalmotor.com* [online]. ORIENTAL MOTOR USA, 2017/2018 [cit. 2023-03-05]. Dostupné z: https://www.orientalmotor.com/products/pdfs/2018-2019/technical-reference/Technical_Reference_Overview.pdf
- [14] TANG, Johann. Motor Sizing Basics Part 3: How to Calculate Speed, Acceleration Torque, and RMS Torque. In: *Blog.orientalmotor.com* [online]. Torrance CA: ORIENTAL MOTOR USA, ©2022, Feb 27, 2020 [cit. 2023-03-05]. Dostupné z: <https://blog.orientalmotor.com/motor-sizing-basics-part-3-acceleration-torque-and-rms-torque>
- [15] ORIENTAL MOTOR. Item # PKP245D15A2-R2EL. In: *catalog.orientalmotor.com* [online]. [cit. 2023-03-05]. Dostupné z: <https://catalog.orientalmotor.com/item/42mm-frame-stepper-motors/42mm-pkp-series-2-phase-bipolar-stepper-motors/pkp245d15a2-r2el>
- [16] MOONS'. MS17HD2P4150. In: *moonsindustries.com* [online]. [cit. 2023-03-05]. Dostupné z: <https://www.moonsindustries.com/p/nema-17-standard-hybrid-stepper-motors/ms17hd2p4150-000004611110008900>
- [17] TAUBIN, Gabriel, Daniel MORENO a Douglas LANMAN. *3D scanning for personal 3D printing: Build your own desktop 3D scanner* [online]. ACM SIGGRAPH 2014 Studio, SIGGRAPH 2014. 10.1145/2619195.2656314, 2014 [cit. 2023-08-15]. Dostupné z: https://www.researchgate.net/publication/266660058_3D_scanning_for_personal_3D_printing_Build_your_own_desktop_3D_scanner
- [18] ASHYKHMIN, MYKHAILO. *Design of a laser distance sensor with a web camera for a mobile robot* [online]. Prague, 2016 [cit. 2023-08-15]. Dostupné z: https://dspace.cvut.cz/bitstream/handle/10467/70944/F2-DP-2016-Ashykhmin-Mykhailo-Ashykhmin_Diploma.pdf?isAllowed=y&sequence=2. Diploma thesis. Czech technical university in Prague.
- [19] VAGHELA, Siddharth. 3D Scanning with Raspberry Pi and MATLAB. In: *Hackster.io* [online]. © 2023 [cit. 2023-08-15]. Dostupné z: <https://www.hackster.io/strangeloop/3d-scanning-with-raspberry-pi-and-matlab-cc30e8>
- [20] JPIAT. Obstacle detection using Laser and image processing on LOGI-Bone. In: *Community.element14.com* [online]. © 2023 [cit. 2023-08-15]. Dostupné z: <https://community.element14.com/technologies/fpga-group/b/blog/posts/obstacle-detection-using-laser-and-image-processing-on-logi-bone>
- [21] Image Types in the Toolbox: Truecolor Images. In: *Mathworks.com* [online]. Natick, © 1994-2023 [cit. 2023-08-15]. Dostupné z: <https://www.mathworks.com/help/images/image-types-in-the-toolbox.html>
- [22] Filtry, příslušenství: Typy fotografických filtrů. In: *Fotoskoda.cz* [online]. Praha, © 2023 [cit. 2023-08-15]. Dostupné z: <https://www.fotoskoda.cz/filtry-prislusenstvi/>

Přílohy

Příloha 1: CD-R