VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ
FACULTY OF MECHANICAL ENGINEERING

ÚSTAV AUTOMATIZACE A INFORMATIKY
INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

NÁVRH A REALIZACE 3D SKENOVACÍ METODY ZALOŽENÉ NA ZPRACOVÁNÍ OBRAZU

DESIGN OF 3D SCAN METHOD BASED ON COMPUTER VISION

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — Jaromír Kratochvíl
AUTHOR

VEDOUCÍ PRÁCE — Ing. Michal Růžička
SUPERVISOR

BRNO 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA STROJNÍHO INŽENÝRSTVÍ

Zadání bakalářské práce

Ústav:	Ústav automatizace a informatiky
Student:	**Jaromír Kratochvíl**
Studijní program:	Strojírenství
Studijní obor:	Aplikovaná informatika a řízení
Vedoucí práce:	**Ing. Michal Růžička**
Akademický rok:	2016/17

Ředitel ústavu Vám v souladu se zákonem č. 111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Návrh a realizace 3D skenovací metody založené na zpracování obrazu

Stručná charakteristika problematiky úkolu:

Tato bakalářská práce je zaměřena na návrh a realizaci 3D skenovací metody pro účel vytvoření 3D modelu objektu, který bude dále využit pro tisk na 3D tiskárně. Pro skenování objektu je připraveno zařízení, které je složeno z jedné kamery a laserového paprsku, který je promítán na skenovaný objekt a má charakter úsečky. Dále zařízení obsahuje otočnou platformu pomocí které je možné provést sken celého objektu. Cílem teto práce je navrhnout
a prakticky realizovat software, který bude schopen analyzovat deformaci laserového paprsku a na základě toho určit geometrii daného objektu. Součástí teto bakalářské práce není vytvořit samotný 3D sken, ale pouze určit zmíněnou geometrii skenovaného objektu v aktuální pozici na otočné platformě. Vytvoření 3D skenu přesahuje náročnost bakalářské práce. Na software není kladen požadavek na běh v reálném čase. Získaná obrazová data budou zpracovány v offline rezimu.

Cíle bakalářské práce:

Prozkoumat možnosti 3D skenování objektu.
Vytvořit ovládací software pro otočnou platformu.
Seznámit se s knihovnami OpenCV pro programovací jazyk C++.
Prostudovat možnosti detekce kontur a vybrat vhodnou metodu pro precizní detekci laserového paprsku, který má charakter přímky.
Provést kalibraci kamery a vytvořit stručný návod, jak tuto proceduru provést.
Navrhnout a prakticky realizovat metodu pro precizní detekce laseru.
Navrhnout a prakticky realizovat metodu pro vyhodnocení geometrie skenovaného objektu.
Vytvořit software, který bude obsahovat obě výše zmíněné metody. Součástí softwaru by měla být i 3D vizualizace skenovaného objektu v aktuální pozici na otoční platformě.
Provést řadu praktických ověřovacích experimentů pro navržené řešení.

Seznam literatury:

SHERVIN, E. Mastering OpenCV with Practical Computer Vision Projects, Packt Publishing Limited.
KAEHLER, A. Learning OpenCV 3, O'Reilly Media, Inc, USA, 2015, ISBN 9781491937990.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2016/17.

V Brně, dne 24. 11. 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA STROJNÍHO INŽENÝRSTVÍ
DĚKANÁT
L. S.

doc. Ing. Radomil Matoušek, Ph.D.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

ABSTRAKT

Tato práce se zabývá vytvořením 3D skenovací metody pro vizualizaci laserového paprsku. V první časti se zabývá rozdělením 3D skenerů a vysvětluje, jak fungují. V druhé časti jsou porovnávány různé nástroje pro zpracování obrazu a je vybrán ten nejvhodnější. Ve třetí části je navrženo vlastní řešení, kde jsou vysvětleny jednotlivé metody, které byly použity, nebo které byly zvažovány pro tuto práci. Čtvrtá část se zabývá vlastním řešením, kde je vysvětleno, jak byla práce řešena. Další část se zabývá praktickými experimenty, kde se navržené řešení aplikuje pro různé předměty v různých pozicích. Poslední částí je závěr, který shrnuje výsledky této práce.

ABSTRACT

This thesis deals with designing 3D scan method for visualization of laser beam. The first part deals with the division of 3D scanners and explains how they work. The second part compares various image processing tools and selects the most suitable for this work. The third part proposes a solution that explains the individual methods that were used or considered for this work. The fourth part deals with the solution, explaining how the methods has been created. The next part deals with practical experiments where proposed solutions are applied to different objects in different positions. The last part is the conclusion summarizing the results of this work.

KLÍČOVÁ SLOVA

3D skenování, 3D skener, detekování laserového paprsku, zpracování obrazu v OpenCV knihovně, aproximace kontur, získávání kontur, kalibrace kamery, vizualizace obrazových dat.

KEYWORDS

3D scanning, 3D scanner, laser detection, image processing using OpenCV library, contours approximation, contour detection, camera calibration, visualization of image data.

BIBLIOGRAFICKÁ CITACE

KRATOCHVÍL, J. *Návrh a realizace 3D skenovací metody založené na zpracování obrazu.* Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2017. 61 s. Vedoucí bakalářské práce Ing. Michal Růžička.

PODĚKOVÁNÍ

Rád bych touto cestou poděkoval své přítelkyni, rodině a přátelům za podporu mého studia. Dále děkuji vedoucímu bakalářské práce Ing. Michalu Růžičkovi za odbornou pomoc a cenné rady v oblasti zpracování obrazu.

ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že tato práce je mým původním dílem, zpracoval jsem ji samostatně pod vedením Ing. Michala Růžičky a s použitím literatury uvedené v seznamu literatury.

V Brně dne 24. 5. 2017 ……

Jaromír Kratochvíl

OBSAH

1 ÚVOD

3D skenování neboli 3D digitalizace či 3D měření má podstatu v přenesení fyzického předmětu nebo prostoru do digitální podoby. Tento proces je pro většinu lidí zcela běžný v provedení ve 2D prostoru. Každý zná skener dokumentů a každý, kdo vlastní mobilní telefon, fotí vestaveným digitálním fotoaparátem. Poloha libovolného bodu dokumentu, nebo fotografie je určena dvěma parametry (X, Y). Poloha libovolného bodu na povrchu 3D objektu je k tomu určena hloubkou, takže třemi parametry (X, Y, Z), k jejich zjištění se využívají 3D skenery. 3D skenování se od 2D tedy liší v tom, že zjišťuje jednu souřadnici bodu navíc. V praxi to tak jednoduché není, protože do výpočtu vstupují i jiné vlivy, které se ve 2D skenování neřeší. [3]

3D skenování je považováno za mladé, rychle rostoucí a rozšiřující se technické odvětví. Problém dříve nebyl v matematických výpočtech 3D bodu z 2D fotek, ale spíše ve velkých výpočtových nárocích. S růstem procesorového výkonu se začala rozšiřovat i tato oblast. Kvalitní 3D skener se nerozlišuje podle rozlišení kamery, ale spíše podle schopnosti softwaru a kombinace všech hardwarových komponent jako jsou objektivy, kamery a projekční jednotka. Dnes jsou 3D skenery uživatelsky tak přívětivé, že naučení se s nimi pracovat zabere jen několik hodin. [3]

Cílem této bakalářské práce je seznámit se s technologií zpracování obrazových dat pro využití ve 3D skenování. Je potřeba navrhnout a prakticky realizovat software, který bude schopen analyzovat deformaci laserového paprsku a na základě toho zobrazit geometrii daného objektu. Cílem není vytvořit samotný 3D sken, ale pouze zobrazit ve 3D geometrii skenovaného objektu v aktuální pozici na otočné platformě. Na základě této metody bude poté možno vytvořit software pro 3D modelaci objektu, to ale není součástí této bakalářské práce.

2 3D SKENERY

3D skenery jsou zařízeni sloužící k zachycení tvarů a textur reálných objektů nebo prostředí. Tato data jsou převáděna do digitální podoby, ze které jsou poté vytvářeny trojrozměrné modely. Existuje více technologií pro 3D skenování. Každá technologie má různé výhody, limity a cenu. Většina z nich snímá jednotlivé body na povrchu objektu nebo prostředí a vytváří mračna bodů geometrických vzorků. Sbírají se informace o vzdálenosti bodů na povrchu, což umožňuje jejich rekonstrukci a vytvoření trojrozměrného modelu. Tyto modely mají více využití, používají se například ve filmovém průmyslu k nahrazení postav nereálnými, v herním průmyslu, v reverzním inženýrství, v průmyslovém designu, v marketingu a jinde. [4]

Nejčastěji používané základní rozdělení je na kontaktní a bezkontaktní 3D skenery. Tyto kategorie se dále člení, jak je popsáno níže. [4]

- Kontaktní
  - Nedestruktivní
    - Mechanické paže
    - Robotické paže
    - CMM
  - Destruktivní
- Bezkontaktní
  - Reflexní
    - Optické
      - Aktivní
      - Pasivní
    - Laserové
    - Akustické
  - Transmisivní
    - Průmyslové CT
  - Magnetické
    - Magnetická rezonance
    - Magnetické sondy

2.1 Kontaktní 3D skenery

Kontaktní 3D skenery snímají předmět prostřednictvím fyzického kontaktu se snímací sondou. Mohou být destruktivní, nebo nedestruktivní. [4]

2.1.1 Nedestruktivní

Tyto skenery na rozdíl od destruktivních skenerů objekt při procesu digitalizace nijak nepoškodí. Zkoumají povrch předmětu, který je pevně připevněn k položce fyzickým hmotným dotykem se sondou. Do této skupiny patří mechanické paže, robotické paže a souřadnicové měřící stroje CMM (Coordinate Measuring Machine). [3]

1) U mechanické paže se získávají data polohy bodu, který sonda snímá, na základě informací o poloze jednotlivých kloubů, ze kterých se skener skládá a které v reálném čase posílají data o své poloze. Software, který tato data přijímá, přepočítává polohu kuličky na koncovém rameni. K přepočtu je potřeba znát i délku ramen. Během skenování musí být skener připevněn k pracovní desce a jeho základna se nesmí pohnout, protože se poloha objektu vůči skeneru nesmí změnit. Výhodou je finanční nenáročnost a mobilita. Nevýhodou je nižší přesnost a nutnost obsluhy. V praxi se tyto skenery používají pro kontrolování dimenzí objektu. [3]

Obr. 1: Mechanická paže FaroArm [3]

2) Robotická paže funguje na stejném principu jako mechanická paže. Rozdíl je ten, že provoz je zautomatizován, nebo probíhá prostřednictvím dálkového řízení. Využívá se pro provoz v nebezpečném nebo nehostinném prostředí. [3]
3) CMM je polohovací rameno se sondou, díky které je možno bodově snímat 3D data objektu. Poloha je zaznamenávána CMM enkodérem. Proces je zdlouhavý, protože pohyby CMM nejsou příliš rychlé. [3]

Obr. 2: CMM DEA GLOBAL [3]

2.1.2 Destruktivní

Předmět snímaný tímto skenerem bývá během procesu zničen. Výhodou ale je, že jeho použití umožňuje snímat nejen vnější povrch objektu, ale i vnitřní geometrii. Oproti CT skenerům, které jsou toho schopny také a nejsou destruktivní, jsou finančně méně náročné. V dnešní době jsou používány hlavně v reverzním inženýrstvím i s kombinací bezkontaktních skenerů, kdy je potřeba vytvořit digitální model různých součástí, které mají složitou vnitřní geometrii a bylo by příliš obtížné a nákladné ji napodobit. Nejprve se skenovaný objekt zalije pomocným materiálem, který se musí dostat do všech dutin, a musí mít kontrastní barvu oproti předmětu, který je potřeba digitalizovat. Takto připravený objekt se připne na desku frézky, která od něj postupně frézuje malé vrstvy povrchu. Tyto vrstvy jsou zaznamenávány kamerou a výsledkem je sada 2D záznamů s informací o výšce každé vrstvy. Z těchto fotografií se získají okraje objektu a postupně se vytvoří 3D mrak bodů. [3]

Obr. 3: Destruktivní skener Pearl 700 [3]

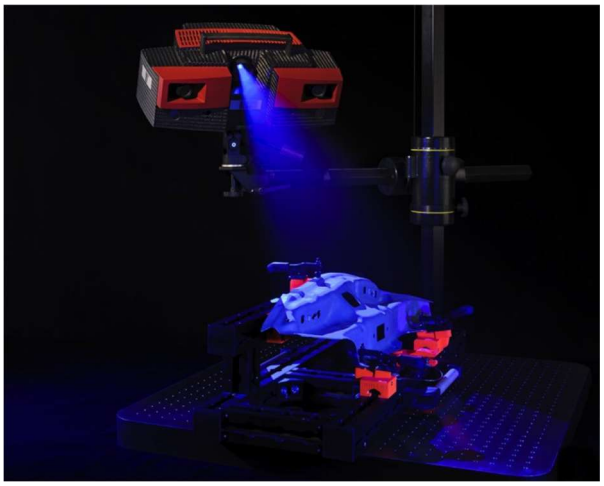
2.2 Bezkontaktní 3D skenery

Bezkontaktní 3D skenery oproti kontaktním nepotřebují ke svému provozu fyzický dotyk s povrchem objektu. Mohou být reflexní, transmisní, nebo magnetické. V dnešní době jsou více rozšířené než kontaktní, a to především díky jednoduchosti obsluhy, časové nenáročnosti a přesnosti. [4]

2.2.1 Reflexní

Mezi reflexní 3D skenery patří hlavně skenery optické, laserové a akustické. [4]

1) U optických metod je na povrch skenovaného předmětu promítán pravidelný vzor, který se podle geometrie objektu zakřiví. Takto zakřivený vzor je snímán optickým zařízením skládajícím se z levé a pravé kamery, které objekt zachytávají současně. Zachytávání objektu se vykoná vícekrát, po každém se promítaný vzor na povrchu objektu mírně posune. Pro všechny snímané polohy skenerem je zachycena sada snímků, ze kterých se softwarem získají prostorové souřadnice bodů na povrchu snímaného předmětu. Vypočítané skeny různých poloh jsou prostorově spojovány pro vytvoření modelu, a to buď s využitím referenčních značek, nebo metodou BestFit. Někdy jsou optické skenery používány jako trackovací zařízení, díky čemuž je možno použít kontaktní skener v prostředí, kam kamery nevidí. Umožňují také zpětnou projekci. Do modelu je možno přidat bod, kružnici či osy a jejich body se naprojektují na objekt podle jeho geometrie. Poté se tyto prvky převedou na reálný objekt. [3]



Obr. 4: Optický skener III Triple Scan [3]

2) Laserové skenery jsou velmi rozšířené. Používají se jako rozšiřující skenovací hlavy pro kontaktní skenery i jako samostatná zařízení. Oba typy pracují odrazem laseru od povrchu snímaného předmětu. Pro vytvoření modelu je nutno snímaný předmět či laser otáčet v prostoru a poloha skeneru musí být při každém snímání známá. Laserový skener však bývá obsluhován uživatelem, který stále mění jeho polohu, tudíž tyto informace známy nejsou. Proto se používá metoda trackování, nebo metoda referenčních značek. Při trackování laserového skeneru se používá pevně připevněné optické zařízení. Na laserový skener jsou přichyceny reflexní body, které dané trackovací zařízení sleduje a určuje vzdálenost a natočení reflexních bodů v prostoru, díky čemuž je možno určit absolutní polohu bodu na povrchu snímaného předmětu. Největší nepřesnost z celé metody má trackovací zařízení. [3]

Obr. 5: Laserový skener T-Scan s trackovacím zařízení Leica [3]

Více dostupnou metodou díky absenci trackovacího zařízení je metoda absolutního polohování používající referenční značky, které se na snímaný předmět přidávají před procesem skenování. V každém snímku z každé polohy musí být vidět alespoň 3 tyto značky. Kvůli dosahu laseru je vzdálenost kamery od snímaného předmětu relativně nízká, tudíž je záběr omezen a je nutno mít na předmětu poměrně velké množství značek. Tyto laserové snímače jsou cenově dostupné, ale jsou spíše vhodné pro nenáročné aplikace. Nevýhodou může být také hmotnost tohoto skeneru, pokud je nutnost ručního snímání. [3]

Obr. 6: Laserový skener HandyScan [3]

3) Akustické 3D skenery využívají sondu emitující zvukové vlny (ultrazvuk, nebo sonar). Tento typ skenerů patří mezi přesnější, ale finančně náročnější metody, a je nejvíce používán v oblasti kontroly kvality. Pro snímání povrchu objektu se používá manuální ultrazvuková sonda, která svým vzhledem připomíná pistoli a má kovový hrot, který se pokládá na skenovaný předmět. Stisknutím spouště je vyslána zvuková vlna, která se převede do prostorových souřadnic pomocí ultrazvukových čidel. [4]

2.2.2 Transmisivní

Do skupiny transmisivních skenerů patří zařízení pracující na principu počítačové tomografie (CT). Stejně jako u kontaktních destruktivních skenerů se tento typ skenerů používá k získávání informací o vnitřní stavbě snímaného předmětu. Data se přenáší pomocí rentgenového záření, které má pro průmyslové účely větší intenzitu než verze používaná ve zdravotnictví. [4]

CT skenery využívají X-paprsky rentgenového záření. Původně se používaly pro určení vnitřních efektů odlitků důležitých součástek, ale postupem času rostla nutnost přesného nedestruktivního vnitřního skenování. Snímaný předmět se vloží do uzavřené komory, ve kterém se ze všech stran zrentgenuje. Z takto vytvořených záznamů se na

základě kontrastu a známé polohy natočení spočítají prostorové body všech vnějších i vnitřních kontur snímaného předmětu. Skenování nevyžaduje žádnou přípravu, navíc je i rychlé a přesné. Nevýhodou je finanční náročnost, malý prostor komor a problémy v případě, že je skenovaný předmět vyroben z více materiálů [3, 4]

Obr. 7: CT skener Wenzel ExaCT M100 [3]

2.2.3 Magnetické

Magnetické skenery dělíme na skenery s magnetickou sondou a skenery využívající magnetickou rezonanci. Magnetická rezonance se používá pro nedestruktivní snímání vnitřní geometrie objektu, bývá často mobilní a používá se hlavně ke kontrole uzavřených nádob. Využívá stejný princip jako magnetické rezonance v nemocnicích. [4]

3 ZPRACOVÁNÍ OBRAZU

Svět je plný tvarů, barev, textur a pohybů. Lidské vnímání získává, integruje a interpretuje tyto vizuální informace. Zpracování obrazu se snaží pomocí technik skladování, zpracování, přenosu a interpretace vizuálních snímků předat tuto schopnost strojům. Právě zpracování obrazu je jednou z nejdůležitějších součástí 3D skenovacích metod. Prvním krokem k navrhování systémů analýzy obrazu je digitální snímání obrazu pomocí senzorů optických, nebo tepelných vlnových délek. Dvojrozměrný obraz, který je zaznamenán v těchto snímcích, je mapa trojrozměrného světa, která může být pomocí série takovýchto snímků zrekonstruována. Snímky bývají často poškozené, chyba může být na straně snímače, uživatele, nebo prostředí. Snímací kamera s obrazem nijak nepracuje, pouze předává jednotlivé snímky softwaru. Ten je pomocí pokročilých metod upraví, získá informace o vyhodnocovaných bodech snímaného objektu, a nakonec vytvoří požadovaný model. [6]

3.1 Nástroje pro zpracování obrazu

V této části práce jsou zmíněny různé nástroje pro zpracování obrazu, které budou poté srovnány, a bude vybrán takový nástroj, který je pro praktickou část této práce nejvhodnější.

3.1.1 OpenCV

OpenCV (*Open Source Computer Vision*) knihovna je open source knihovna, která se zabývá zpracováním obrazu. Knihovna je napsána v C a C ++ a běží pod operačními systémy Linux, Windows a Mac OS. Dostupná je kromě C a C++ také pro programovací jazyky Python, Ruby, Matlab a Java. OpenCV bylo navrženo s důrazem pro výpočetní efektivitu a pro aplikace pracující v reálném čase. Může využívat vícejádrové procesory. Pro lepší automatickou optimalizaci na architektuře Intel je možnost pracovat s knihovnou IPP (*Integrated Performance Primitives*). Dostupná je i hardwarová optimalizace s podporou technologie Nvidia CUDA (*Computer Unified Device Architecture*), kdy se výpočty neprovádí na procesoru, ale na grafické kartě, která je pro tento typ výpočtu vhodnější. Jedním z cílů OpenCV je poskytnout uživatelům jednoduchou infrastrukturu, která pomáhá rychle budovat sofistikované aplikace počítačového vidění. OpenCV obsahuje mnoho funkcí, které pokrývají více oblastí počítačového vidění, například tovární kontrolu výrobku, lékařské snímkování, bezpečnost, uživatelské rozhraní, kalibraci kamery, stereo vidění, nebo robotiku. Také umožňuje 3D vizualizaci, která je pro tuto práci potřebná. Knihovna má také aktivní komunitu, která se skládá z čtyřiceti sedmi tisíc členů a má propracovanou dokumentaci. [2, 7]

3.1.2 Intel® IPP

Již zmíněná IPP knihovna od společnosti Intel je složena z množství programovacích algoritmů, které optimalizují běh aplikací na platformách procesorů Intel® Quark™, Intel® Atom™, Intel® Core™, Intel® Xeon™ a Intel® Xeon Phi™. Aplikace s touto knihovnou budou pětkrát až desetkrát výkonnější než aplikace zkompilované bez ní. Intel IPP obsahuje postupy pro zpracování obrazu, počítačové vidění, komprese dat i zpracování signálu. Komunitní licence je šířena zdarma pro nekomerční použití. A jak již bylo zmíněno, pokud je tato knihovna nainstalována, automaticky optimalizuje aplikace napsané v knihovně OpenCV do té doby, dokud výpočty probíhají na výše uvedených platformách procesorů Intel. [8]

3.1.3 SimpleCV

SimpleCV knihovna je založena na knihovně OpenCV, je také open source a také se zabývá hlavně zpracováním obrazu. Je určena výhradně pro programovací jazyk Python a tím pádem pracuje na operačních systémech Windows, Linux i Mac OS. Pro náročné algoritmy není vhodná právě proto, že funguje výhradně pro Python, který není tolik výkonný jako například jazyky C a C++. Pro jednoduché aplikace s menší snímkovací frekvencí se však používá. Z toho by se dalo by dedukovat, že jazyk Python je pro tuto knihovnu nevýhodou, což ale není úplně pravda, protože díky jeho jednoduchosti a názornosti se přes jeho nevýhody hojně využívá. A jelikož má SimpleCV za cíl vytvářet jednoduché a uživatelsky pochopitelné metody zpracování obrazu bez nutnosti hlubokých předchozích znalostí, Python je pro tuto knihovnu výhodný partner. [9, 10]

3.1.4 Emgu CV

Emgu CV je multiplatformní knihovna pro jazyk C#, VB, C++ a IronPython. Je opět založena na OpenCV knihovně a hlavním cílem je umožnit volání OpenCV funkcí z .Net kompatibilních programovacích jazyků. Emgu CV je napsaná čistě v jazyce C# a používá se ke zpracování obrazu v reálném čase. Mimo desktopové operační systémy je také určena pro mobilní telefony. Pro open source projekty je volně dostupná a pro komerční projekty, které chtějí kód utajit, je nutno koupit od společnosti EMGU licenci. Hlavní výhodou této knihovny je, že je multiplatformní a umožňuje použití vyšších programovacích jazyků. [11]

3.1.5 BoofCV

BoofCV knihovna je určena pro programovací jazyk Java a je také open source. Jejím hlavním využitím je zpracování obrazu, ale používá se i pro práci v dalších oblastech robotiky. Její součástí je mimo zpracování obrazu i geometrická interpretace obrazu, kalibrace kamer a vizualizace snímaných dat. Je zpracována s důrazem na akceleraci nízko úrovňových procesů. Výhodou této knihovny je dostupnost a rozsah dokumentace, která je zdarma dostupná přímo na jejích webových stránkách. Další výhodou je

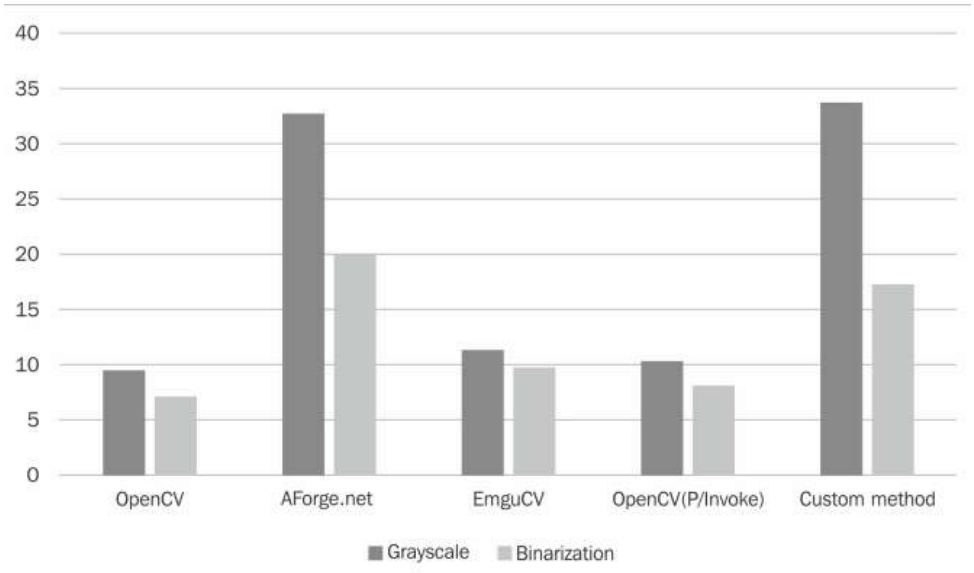
jednoduchost a názornost vyššího jazyka Java, který však stejně jako Python způsobuje pomalejší chod složitějších aplikací. [12]

3.1.6 MATLAB Computer Vision System Toolbox

MATLAB Computer Vision System Toolbox je rozšíření, které je možno dokoupit k softwaru Matlab, což je nástroj pro inženýrské a vědecké aplikace. Tento toolbox, který se zabývá pokročilými algoritmy počítačového vidění, potřebuje ke své práci mimo samotný software MATLAB ještě Image Processing Toolbox, který se zabývá zpracováním obrazu. Computer Vision System Toolbox umožňuje simulaci a zpracování obrazu v reálném čase, detekci objektu, sledování objektu, zpracování obrazu ze stereoskopických kamer, 3D rekonstrukci obrazu či 3D bodovou projekci dat. Zahrnuje v sobě i funkce strojového učení. Programování je možné pomocí grafického programování Simulink, nebo strukturovaného textu. Výhodou je univerzálnost nástroje, který má velkou škálu využití. Nevýhodou je pomalejší chod u složitějších algoritmů a také cena produktu. Standardní cena Computer Vision System Toolbox je stanovena na 1250 EUR, Image Processing Toolbox má hodnotu 1000 EUR a cena samotného softwaru MATLAB se pohybuje okolo 2000 EUR. [13]

3.2 Volba nástroje

Pro praktickou část této práce byla zvolena knihovna OpenCV. Její výhodou je možnost práce s programovacími jazyky C/C++, které jsou nejlepší volbou pro optimalizaci chodu. Nepotřebují totiž interpretační programy, které zatěžují chod programu. A jak již bylo zmíněno, OpenCV má obsáhlou dokumentaci a velkou komunitu, která tuto knihovnu využívá. Navíc tuto knihovnu vyzdvihují i výkonnostní testy. [14]



Obr. 8: Výkonost knihoven pro zpracování obrazu. Tmavě šedá barva ukazuje čas v milisekundách pro převod obrazu na černobílý a světlešedá na binární. [14]

4 NÁVRH ŘEŠENÍ

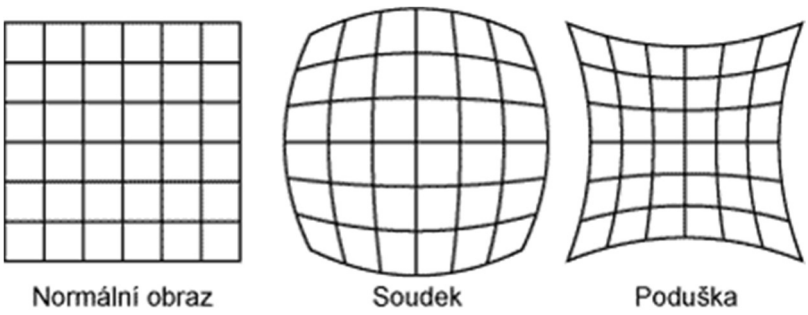
Cílem této bakalářské práce je navrhnout a prakticky realizovat 3D skenovací metodu pro účel vytvoření 3D modelu objektu, který bude dále využit pro tisk na 3D tiskárně. Součástí teto bakalářské práce není vytvořit samotný 3D sken, ale pouze určit geometrii skenovaného objektu v aktuální pozici na otočné platformě. Skenování objektu je prováděno v zařízení, které se skládá z jedné kamery a liniového laserového paprsku, který je promítán na skenovaný objekt. Dále zařízení obsahuje otočnou platformu, pomocí které je možné provést sken celého objektu. Získaná obrazová data budou zpracována offline. Součástí této práce je navrhnout ovládací software pro otočnou platformu a realizovat software, který bude schopen analyzovat deformaci laserového paprsku.

4.1 Návrh algoritmu pro zpracování laserového paprsku

V následující části práce jsou shrnuty jednotlivé kroky navrženého algoritmu. Algoritmus začíná nejdříve získáním obrazových dat z kamery, které převádí do vhodné podoby. Celý program se skládá z několika částí. První část se skládá z kalibrace kamery, kdy se odstraní zkreslení obrazu. V druhé části se řeší detekce laserového paprsku. V třetí části programu, po nalezení laserového paprsku, jsou detekovány kontury výstupu předchozí části a následně jsou aproximovány. Poslední část algoritmu se zabývá vizualizací výsledku ve 3D.

4.1.1 Kalibrace kamery

Kalibrace kamer se provádí kvůli nedokonalosti čoček průmyslových kamer, které zkreslují snímaný obraz, tudíž zkreslují i vzdálenosti mezi body obrazu, což pro 3D skenování není vhodné. Geometrické zkreslení může být typu soudek nebo poduška. [15]



Obr. 9: Sférické zkreslení obrazu [16]

OpenCV upravuje zkreslení pomocí radiálního a tangenciálního faktoru. Pixely na vstupním obrazu na souřadnicích (x, y) budou posunuty na pozici $x_{opraveno}$, $y_{opraveno}$), čímž se obraz opraví. Pro radiální faktor se používá následující vzorce: [17]

$$x_{opraveno} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (1)$$

$$y_{opraveno} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (2)$$

Tangenciální zkreslení nastává proto, že obrazové objektivy nejsou dokonale paralelní se zobrazovací rovinou. Může být opraveno pomocí vzorců: [17]

$$x_{opraveno} = x + [2p_1 xy + p_2(r^2 + 2x^2)] \quad (3)$$

$$y_{opraveno} = y + [p_1(r^2 + 2y^2) + 2p_2 xy] \quad (4)$$

Existuje pět parametrů zkreslení, které jsou v OpenCV prezentovány jako matice jednoho řádu s 5 sloupci: [17]

$$Koeficienty\ Zkreslení = (k_1\ \ k_2\ \ p_1\ \ p_2\ \ k_3) \quad (5)$$

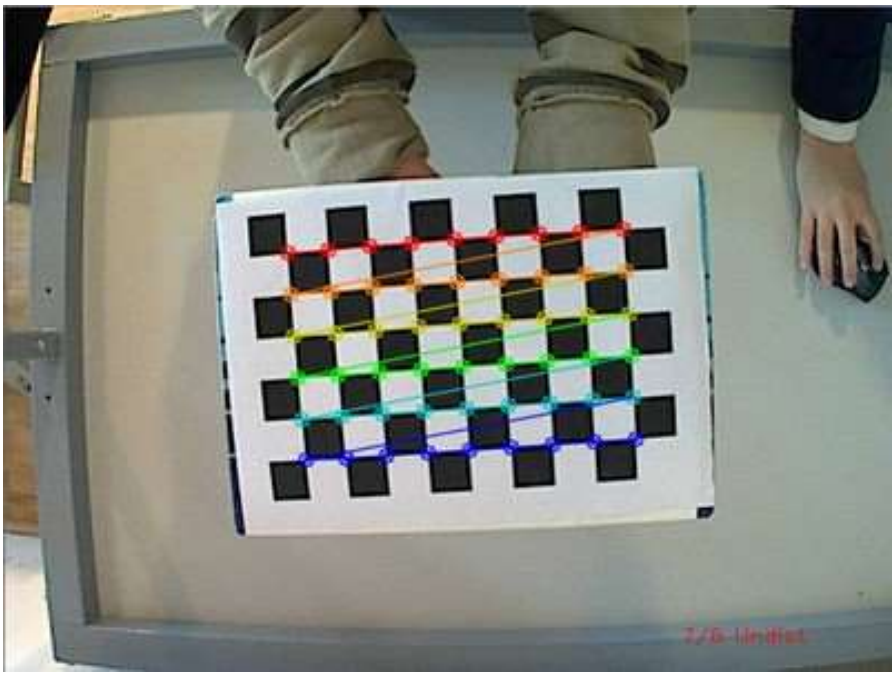
Pro konverzi celku používáme následující vzorec: [17]

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (6)$$

Použitím homografického souřadného systému bylo určeno w ($w = Z$). Neznámé parametry jsou f_x, f_y (ohnisková vzdálenost kamery) a c_x, c_y (optická centra vyjádřená v souřadnicích obrazových bodů). Pokud se pro obě osy používá společná ohnisková vzdálenost s poměrem stran a (obvykle 1), pak $f_y = f_x * a$ a ve vzorci (6) budeme mít jednu ohniskovou vzdálenost kamery f. Matice obsahující tyto čtyři parametry je označována jako matice kamery. Koeficienty zkreslení jsou stejné bez ohledu na použité rozlišení fotoaparátu. [17]

Proces určení koeficientů zkreslení a matice kamery se nazývá kalibrace. Výpočet těchto parametrů se provádí pomocí základních geometrických rovnic. Použité rovnice závisí na zvolených kalibračních objektech. V současné době OpenCV podporuje tři typy objektů pro kalibraci (černobílá šachovnice, symetrický kruhový vzor, asymetrický kruhový vzor). Pro tuto práci byla zvolena černobílá šachovnice. [17]

Pro kalibraci je potřeba snímat kalibrační objekt kamerou a nechat ho softwarem detekovat. Každá nová pozice objektu vede k nové rovnici. Pro vytvoření dobře kladeného systému rovnic je potřeba větší množství snímků. Teoreticky šachovnice vyžaduje pouze dva snímky. V praxi se ale ve snímcích vyskytuje šum, tudíž se objekt snímá alespoň desetkrát v různých pozicích. [17]



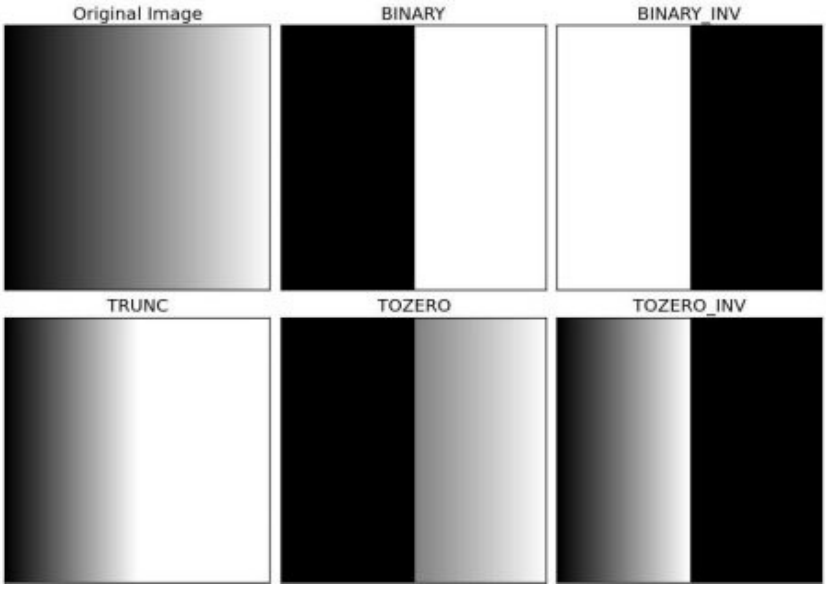
Obr. 10: Šachovnice použitá pro kalibraci [17]

4.1.2 Detekce laserového paprsku

Detekce laserového paprsku se dá řešit pomocí detekce barvy laserového paprsku, pokud jsou během pracovního režimu skenovacího zařízení stabilní světelné podmínky. Také se dá využít detekce přímek, pokud je laserový paprsek liniového tvaru. [17]

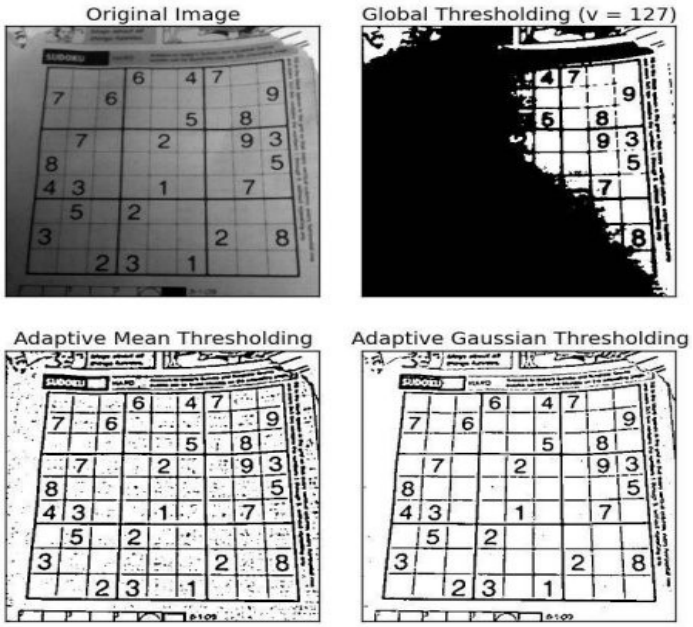
Zařízení použité v této práci splňuje obě podmínky, tudíž se dají využít jak oba přístupy, tak jejich kombinace. Mohou se použít metody Threshold, Adaptivní Threshold, Houghova transformace, Randomizovaná Houghova transformace nebo Sobelův filtr. [17]

1) Threshold je jednoduchá a přímá metoda. Je-li hodnota pixelu větší než prahová hodnota, je mu přiřazena jedna barva (často bílá), v opačném případě je přiřazena jiná barva (často černá). Vstupem funkce v OpenCV je obraz, který musí být ve stupních šedi. Metoda také vyžaduje prahovou hodnotu, která se používá pro klasifikaci hodnot pixelů. Dále je potřeba barva, která má být dosazena pixelu, jestliže jeho hodnota je větší (nebo menší) než prahová hodnota. OpenCV poskytuje různé styly Thresholdu, jsou jimi BINARY, BINARY_INV, TRUNC, TOZERO a TOZERO_INV. [17]



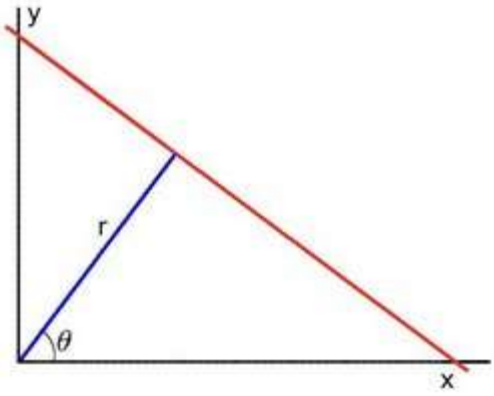
Obr. 11: Použití různých stylů Thresholdu v OpenCV [17]

2) Adaptivní Threshold na rozdíl od Thresholdu nepoužívá prahovou hodnotu globálně pro celý snímek. Pro některé případy je tato metoda více vhodná, například v místnostech, kde má obraz v různých svých částech odlišné světelné podmínky. Tato metoda počítá prahovou hodnotu pro malé oblasti obrazu, takže existuje více různých prahových hodnot pro různé oblasti stejného obrazu. Vstupem je také obraz, který musí být ve stupních šedi. Je potřeba metodě v OpenCV zadat velikost bloků, pro které se prahová hodnota bude počítat. Má dva styly, jedním z nich je MEAN_C, kde je prahová hodnota vypočítána jako střední hodnota oblasti, druhým z nich je GAUSSIAN_C, kde je prahová hodnota získána pomocí Gaussovy křivky. [17]



Obr. 12: Použití různých stylů Adaptivního Thresholdu v OpenCV [17]

3) Houghova transformace patří mezi metody detekce hran. Může se použít pro detekci přímek, ale také pro určení jiných tvarů, nejčastěji kruhovitých, nebo eliptických. V této práci je použita metoda pro detekci přímek. Přímka může být v 2D prostoru vyjádřena dvěma parametry. OpenCV využívá pro určení přímky pomocí Houghovy transformace polární souřadný systém, kde jako parametry používá délku normály r a úhel θ. [17]

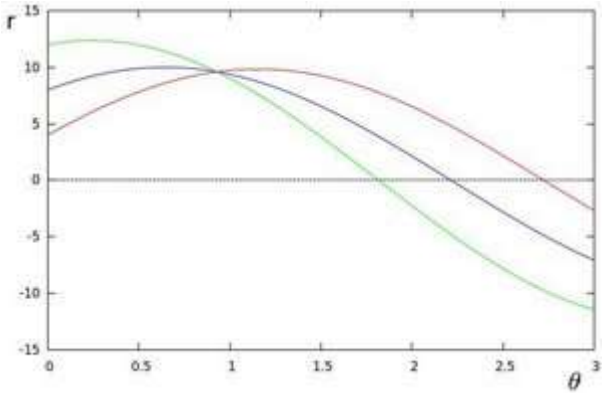


Obr. 13: Přímka v polárním souřadném systému [17]

Pro určení přímky je platná rovnice: [17]

$$r = x\cos\theta + y\sin\theta \tag{7}$$

To znamená, že pokud se do této rovnice (7) dosadí souřadnice bodu (x_i, y_i), pak množina všech možných řešení (r, θ) vytvoří v Houghově prostoru spojitou křivku. Pokud se takto zobrazí do Houghova prostoru všechny body ležící na přímce p, křivky jednotlivých bodů se (x_i, y_i) se protnou v jednom bodě (r_{max}, θ_{max}). Tyto body jsou hledané parametry přímky p. [17]



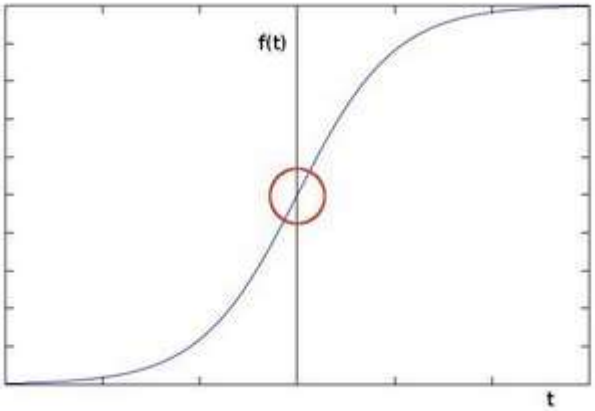
Obr. 14: Zobrazení bodů přímky p v Houghově prostoru [17]

Houghova transformace sleduje průnik mezi křivkami každého bodu v obraze. Pokud je počet průsečíků křivek v jednom bodě nad určitou prahovou hodnotou, deklaruje tyto body jako součást jedné přímky s parametry (r, θ). [17]

4) Randomizovaná Houghova transformace pracuje na stejném principu jako Houghova transformace, ale neuchovává všechny body přímky, jen její extrémy. [17]

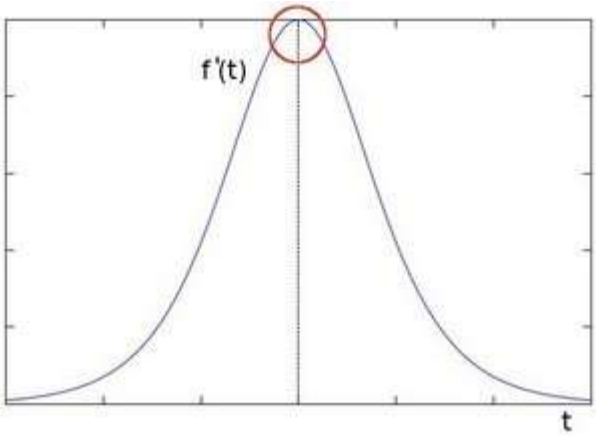
Obr. 15: Randomizovaná Houghova transformace v OpenCV [17]

5) Sobelův filtr je operátor používaný pro detekci hrany. Vypočítá aproximaci gradientu intenzity obrazu, hrana se totiž dá definovat jako změna intenzity pixelu. Tato změna se pak dá vyjádřit pomocí derivace. Vysoká změna gradientu signalizuje významnou změnu v obraze. Také v 1D obraze je hrana definovaná jako změna funkce intenzity. [17]



Obr. 16: Změna funkce intenzity v 1D obraze [17]

Tato změna je lépe viditelná v první derivaci této funkce. [17]



Obr. 17: Změna první derivace funkce intenzity [17]

Z toho vyplývá, že metoda detekce okrajů v obraze může být provedena lokalizací pixelových poloh, kde je gradient vyšší než u sousedních pixelů. Sobelův filtr vyžaduje výpočet dvou derivací, kde jsou detekovány hrany v horizontálním a vertikální směru G_x a G_y. Z těchto výsledků lze poté zjistit gradient G. Počítá se se vstupním obrazem I. Použité vzorce jsou: [17]

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * I \tag{8}$$

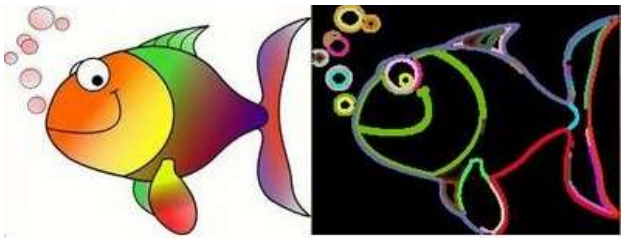
$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * I \tag{9}$$

$$G = \sqrt{{G_x}^2 + {G_y}^2} \tag{10}$$

Po porovnání výše zmíněných metod bylo zvažováno použití kombinace metod Thresholdu a Adaptivního Thresholdu, dále bylo vhodné použití metody Sobelového filtru a samostatné použití Adaptivního Thresholdu. Pro určování kontur laserového paprsku byla ale zvolena právě dříve zmíněná kombinace Thresholdu a Adaptivního Thresholdu, protože z ní byly získány nejlepší výsledky.

4.1.3 Detekce kontur

Kontury lze jednoduše vysvětlit jako křivku spojující všechny spojité body (podél hran), které mají stejnou barvu nebo intenzitu. Kontury jsou užitečným nástrojem pro analýzu tvaru a detekci a rozpoznávání objektů. Pro lepší přesnost se používají binární snímky. Takže se před vyhledáním kontur využívá některá z metod Thresholdu nebo detekce hran. OpenCV neupravuje zdrojový obraz, ale vrací seskupené body kontur jako výstup. Vstupní obraz musí mít černé pozadí a objekt musí mít bílou barvu. Zvolí se také metoda uložení bodů, kde se buď uloží všechny body kontur, nebo jen hraniční body jednotlivých částí. Seskupené body se poté musí vykreslit. Metoda na vykreslení kontur potřebuje zadat obraz, do kterého se budou kontury vykreslovat, barvu a tloušťku. [17]



Obr. 18: Vyhledání a vykreslení kontur pomocí OpenCV [17]

4.1.4 Aproximace kontur

OpenCV aproximuje kontury tak, že přizpůsobuje tvar obrysu jinému tvaru s menším počtem vrcholů v závislosti na přesnosti, kterou specifikujeme. Například pokud se v obraze snažíme najít čtverec, ale kvůli nepřesnostem a světelným podmínkám ho nedostáváme, dá se tato metoda použít. Je jen potřeba zadat správný parametr přesnosti. [17]

Obr. 19: Aproximace kontur v OpenCV [17]

Jelikož metoda neaproximuje laserový paprsek tak přesně, jak je třeba, byla aproximace kontur vytvořena tak, že byly pomocí Randomizované Houghovy transformace nalezeny všechny přímky v obraze. Tyto přímky byly následně seskupeny do kategorií, které je rozdělovaly podle úhlu a vzdálenosti od svých středů, a tím tvořily jednotlivé celky. Z těchto celků se poté uložily extrémy a ty byly vykresleny.

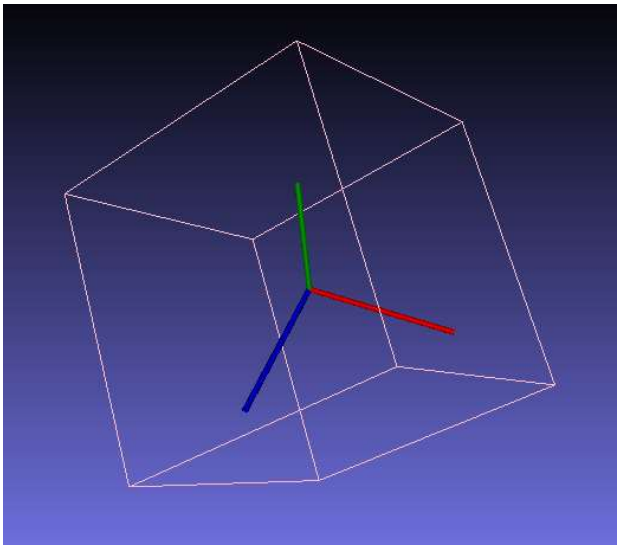
4.1.5 Vizualizace výsledků ve 3D

V OpenCV se objevuje stále více algoritmů pracujících s 3D daty. Autoři se proto snaží vytvořit infrastrukturu, která bude práci v 3D prostoru umožňovat. Jednotka Viz byla prvním krokem daného směru. OpenCV je knihovna, která obsahuje základ, na kterém byly vyvinuty základní algoritmy a aplikace počítačového vidění. Je pohodlná, protože zahrnuje spoustu používaných operací pro manipulaci s obrazy a vizuálními daty. Viz se snaží reputaci OpenCV dostát také. [18]

Viz jako 3D vizualizátor může obrazová data přenášet do 3D prostoru. Dá se do něj přenést i oblak bodů přicházejících z kinectu, který je často uložen v souřadnicovém systému kamery a pro vizualizaci je často nutné převádět všechny mraky bodů získané z různých pozic kamery do globálního souřadného systému. [18]

Knihovna VTK pro manipulaci a vizualizaci vědeckých dat přináší tento přístup, proto OpenCV založila Viz právě na základě balíku VTK. Pro OpenCV bylo toto řešení jasnou volbou i přesto, že závislost na VTK knihovně může být pro vývojáře OpenCV v budoucnu nepříjemná, na druhou stranu bylo toto řešení pohodlné a rozšiřitelné. [18]

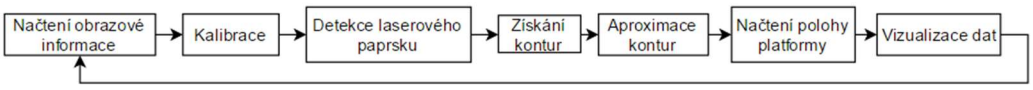
Poloha v euklidovském prostoru je definována otočením a posunem. Otočení může být reprezentováno ve formě otočné matice nebo vektoru otočení. Posun je definován jednoduchým trojrozměrným vektorem. Otočení a posun lze uložit do samostatných proměnných nebo společně do matice 4x4. [18]



Obr. 20: 3D vizualizace krychle pomocí Viz modulu v OpenCV [18]

4.2 Zjednodušené blokové schéma

Následující zjednodušené blokové schéma ukazuje algoritmus pro 3D skenovací metodu.



Obr. 21: Zjednodušené blokové schéma

5 REALIZACE ŘEŠENÍ

Tato část práce se zabývá samotnou realizací projektu. První část kapitoly bude věnována postupu instalace a samotnému začlenění knihoven OpenCV a VTK do prostředí Visual Studia, jelikož pro fungování 3D vizualizace v OpenCV je potřeba i VTK knihovna, jak již bylo zmíněno. Další část bude věnována nastavení zařízení a jeho řízení pomocí Arduino UNO. Poslední část pracuje s vybranými metodami detekce a zpracování obrazu.

5.1 Instalace knihoven OpenCV a VTK

Jak již bylo zmíněno, instalace OpenCV knihovny s 3D vizualizací vyžaduje VTK knihovnu. Pro správnou kompilaci knihoven byl použit open-source multiplatformního software CMake. OpenCV knihovna byla zkompilována pro IDE Visual Studio 14 použité pro tuto práci.

5.1.1 CMake 3.8.1

CMake je multiplatformní open-source software pro automatizaci překladu algoritmu. Je využíván pro tvorbu adresářové struktury a přípravu zdrojových souborů. CMake se používá ke kontrole procesu kompilace softwaru pomocí jednoduchých konfiguračních souborů nezávislých na platformě a kompilátoru. Vytváří nativní soubory a pracovní prostory, které lze použít v prostředí libovolného kompilátoru. Řada nástrojů CMake byla vytvořena firmou Kitware v reakci na potřebu silného, multiplatformního prostředí pro kompilaci open-source projektů. [19]

5.1.2 VTK 7.1

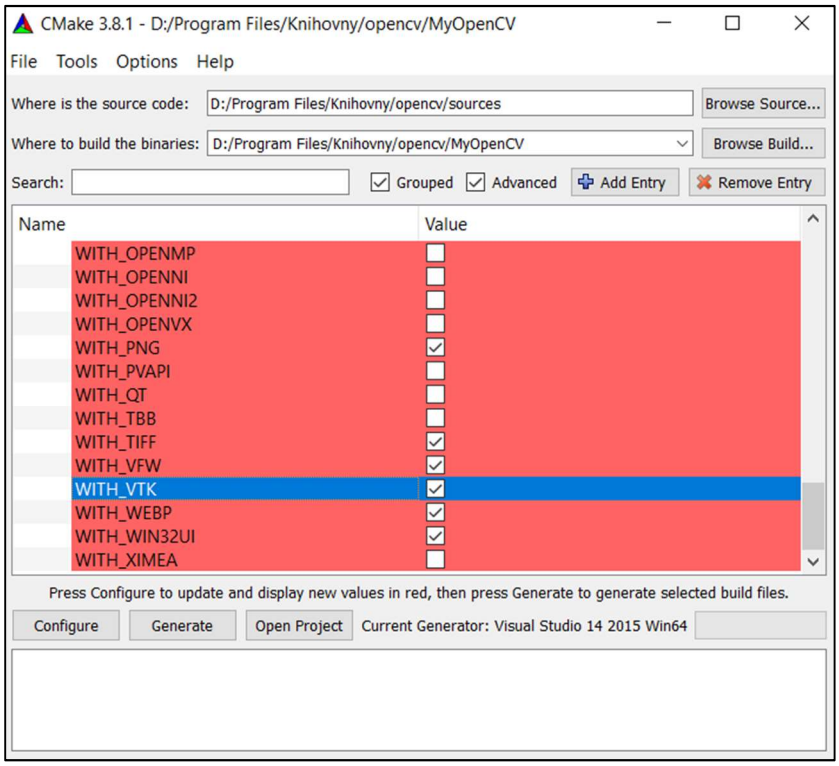
VTK (Visualization Toolkit) je multiplatformní open-source knihovna pro 3D počítačovou grafiku, zpracování obrazu a vizualizaci. VTK podporuje širokou škálu vizualizačních algoritmů včetně skalárních, vektorových, tenzorových, texturních a objemových metod, stejně jako pokročilé modelovací techniky, jako je implicitní modelování, zmenšení polygonů, vyhlazování sítí, řezání, tvarování a triangulace. VTK má rozsáhlý rámec pro vizualizaci informací a sadu metod pro práci s 3D. Dá se integrovat s různými nástroji uživatelského rozhraní jako Qt a Tk. VTK byla vytvořena firmou Kitware, která se zabývá vývojem nástrojů zjednodušujících vývoj aplikací. [20]

Instalace VTK knihovny je pomocí CMake celkem jednoduchá, ale zdlouhavá. VTK knihovna se stáhne a rozbalí do adresáře, kam chceme knihovnu nainstalovat. Poté se pomocí CMake nakonfiguruje pro Visual Studio. Povolí se kompilace jako statické knihovny, znovu se nakonfiguruje a nechá se vygenerovat. Generace trvá cca 20 minut, záleží na výkonu počítače. Vygenerovaný projekt se otevře a nechá zkompilovat. Zde kompilace trvá cca hodinu, ale opět záleží na výkonu počítače. Poté se přes vygenerovaný a zkompilovaný projekt VTK knihovna nainstaluje. [21]

5.1.3 OpenCV 3.2

OpenCV knihovna byla v této práci už představená, v této časti se vysvětlí její instalace tak, aby fungoval Viz modul. Instalace je stejně jako instalace VTK knihovny jednoduchá, ale zdlouhavá.

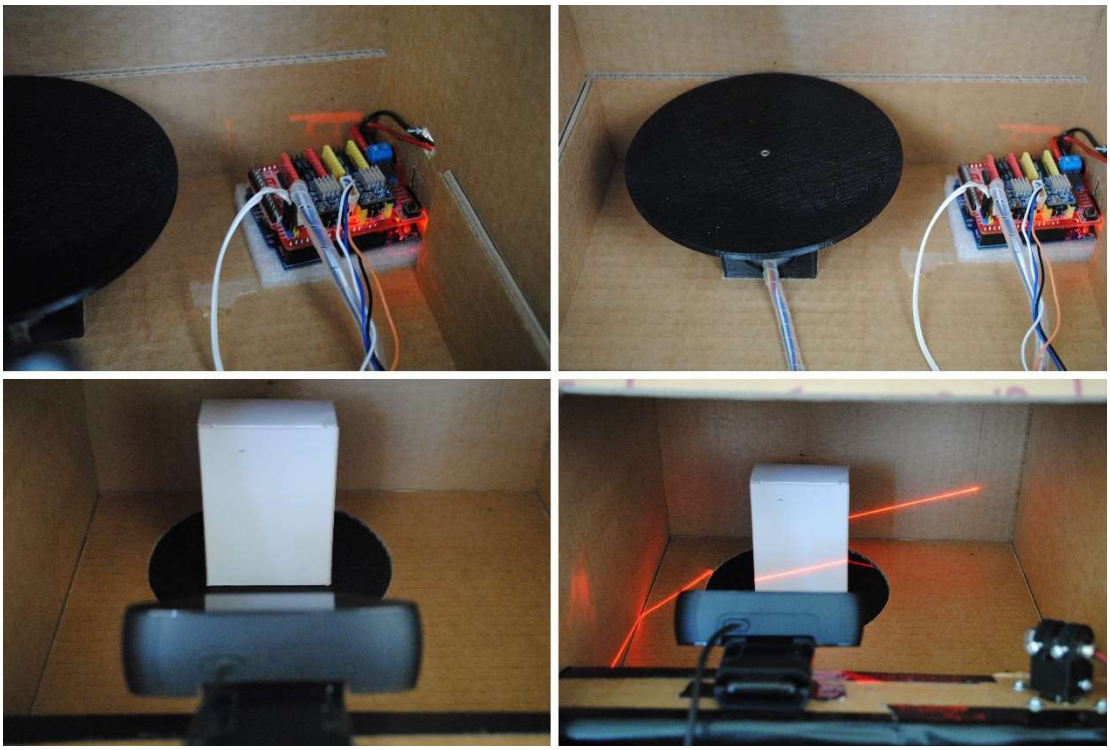
Po nainstalování VTK knihovny se stáhne OpenCV knihovna a rozbalí se do adresáře, kam chceme knihovnu nainstalovat. Instalace je skoro stejná jako u VTK knihovny. Přes CMake se nakonfiguruje pro Visual Studio, nastaví se kompilace s VTK knihovnou a zadá se cesta k VTK knihovně. Poté se nechá vygenerovat, vygenerovaný projekt se zkompiluje a nainstaluje. Vytvoří se nový projekt ve Visual Studiu a do nastavení se přidají složky nainstalované knihovny. [21]



Obr. 22: Generování OpenCV projektu pomocí CMake

5.2 Skenovací zařízení

Skenovací zařízení je uzavíratelné a skládá se z otočné platformy, která je připojena na krokový motor ovládaný pomocí Arduina UNA sériovou komunikací s počítačem. Součástí zařízení je i liniový laser mířící na otočnou platformu a kamera, která celý pohled snímá.



Obr. 23: Vnitřní pohled skenovacího zařízení

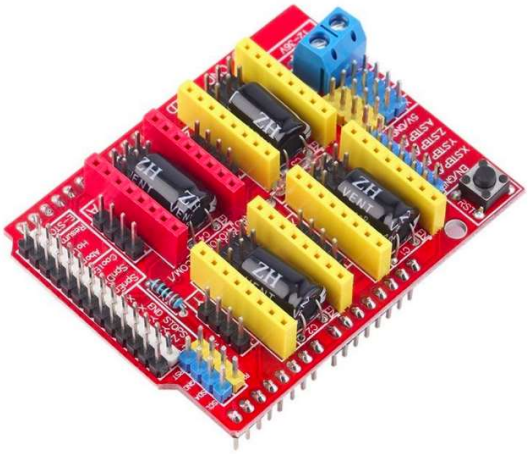
Obr. 24: Zavřené skenovací zařízení

5.2.1 Arduino UNO

Arduino UNO je vývojová deska založená na mikroprocesoru ATmega328. Deska má 14 digitálních pinů (6 PWM), 6 analogových vstupů, 16 MHz krystal, připojení pomocí USB, napájecí konektor, ICSP rozhraní a resetovací tlačítko. Programuje se pomocí Arduino IDE, které je založeno na jazyce Java a může být použito pro programování jakékoliv Arduino desky. [22, 23]

Obr. 26: Arduino UNO [22]

S Arduinem UNEM byl použit i rozšiřující modul CNC Shield, jehož doporučené napájení je 12 V (maximum 36 V) a díky kterému je možnost k Arduinu UNU zapojit až 4 drivery, nebo motory. Díky němu bylo možné zapojit a řídit krokový motor. [22]



Obr. 27: CNC Shield [22]

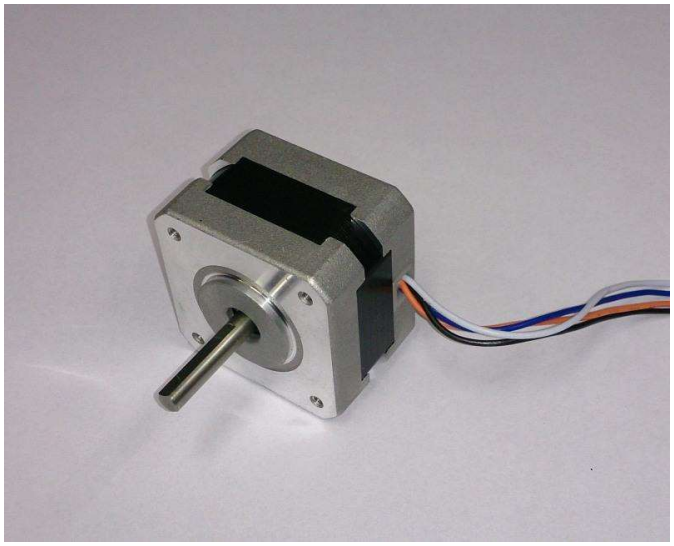
5.2.2 Kamera Logitech HD Pro Webcam C920

Logitech HD Pro Webcam C920 snímá obraz v rozlišení Full HD (až 1920 x 1080 pixelů). Jádrem webové kamery je technologie Fluid Crystal, která zajišťuje ostré barvy, čistý zvuk a ostrý obraz bez trhání. Kamera má optiku Carl Zeiss s automatickým ostřením ve 20 krocích. Komprese videa je ve formátu H.264. Ke kameře jsou vestavěny duální stereofonní mikrofony s automatickým potlačením šumu na stranách kamery, které ale pro tuto práci nejsou potřeba. Probíhá automatická korekce špatného osvětlení. Připojení je umožněno pomocí USB kabelu. Typ senzoru je CMOS. Rozměry senzoru jsou 5,14 x 3,50 mm. [24]

Obr. 28: Logitech HD Pro Webcam C920 [24]

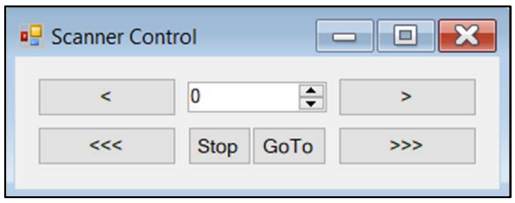
5.2.3 Krokový motor SX17-1003LQCEF

Pro otočnou platformu byl použit krokový motor SX17-1003LQCEF. Motor je lehký, naprosto dostačující pro tuto práci. Jeden krok má 1,8 stupně, kroutící moment je 0,3 Nm, maximální proud je 1 A. Má 4 vývody s konektory o délce 70 cm. [25]



Obr. 29: Krokový motor SX17-1003LQCEF [25]

Ovládání krokového motoru je řešeno sériovou komunikací a bylo napsáno v jazyce C#, které umožňuje zastavení, nebo roztočení platformy po nebo proti směru hodinových ručiček buď na určitý úhel, nebo o určitý úhel, nebo po neomezenou dobu (dokud nedostane další signál). Arduino přepočítává a vysílá stupeň absolutního natočení platformy vůči její počáteční pozici.



Obr. 30: Ovládání otočné platformy

5.2.4 Laserový modul pro Arduino

Laserový modul má výkon 5mW. Optika tohoto laseru rozptyluje paprsek do linie v červené barvě. Jeho vlnová délka dosahuje 650 nm. Napájecí napětí je stejnosměrné s rozsahem 3,5 až 5 V. Průměr má 12 mm. [22]

Obr. 31: Laserový modul pro Arduino [22]

5.3 Vybrané metody detekce a zpracování obrazu

V této části práce se shrnou použité funkce, z již zmíněné knihovny. U každé z funkcí se uvede její princip a ukázku funkce na příkladu obrazových dat.

5.3.1 Kalibrace kamery

Kalibrace kamery byla v této práci již vysvětlena, zde bude předvedeno její použití přímo v OpenCV knihovně. Uživatel musí nejprve nakalibrovat kameru zdrojovým kódem z dokumentace OpenCV pomocí šachovnice, nebo jiných vzorů, který uloží xml soubor se všemi důležitými informacemi o kalibraci. Pro samotné použití těchto informací byl vytvořen algoritmus, který čte potřebná dat z xml souboru a aplikuje je bez potřeby nové kalibrace. [17]

Nejdříve bylo potřeba vytvořit funkci, která čte informace o kalibraci ze zadaného souboru. Vstupem je zadaný název souboru, popřípadě i s cestou, pokud soubor není součástí projektu a výstupem jsou mapy pro vytvoření kalibrovaného obrazu a také šířka a výška snímaného obrazu.

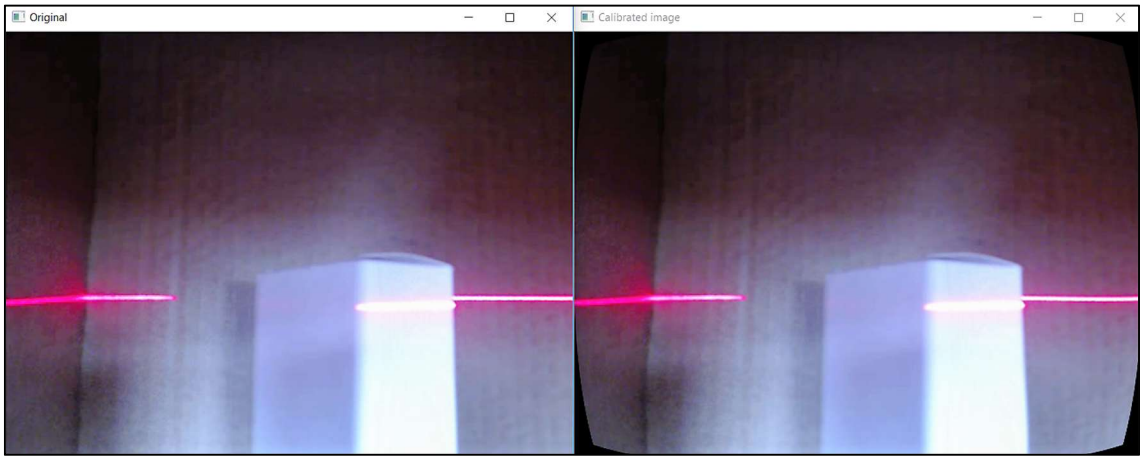
```
void loadSettings(string nameOfDataInput, Mat &outputMap1, Mat &outputMap2, int
&imWidth, int &imHeight) {
	//Calibration Settings
	Mat distCoeffs, cameraMatrix;
	FileStorage fs2(nameOfDataInput, FileStorage::READ);
	fs2["Camera_Matrix"] >> cameraMatrix;
	fs2["Distortion_Coefficients"] >> distCoeffs;
	fs2["image_Width"] >> imWidth;
	fs2["image_Height"] >> imHeight;
	fs2.release();
	initUndistortRectifyMap(cameraMatrix, distCoeffs, Mat(), cameraMatrix,
Size(imWidth, imHeight), CV_16SC2, outputMap1, outputMap2);
}
```

Obr. 32: Funkce pro čtení informací o kalibraci ze zadaného souboru

Poté je potřeba tuto funkci řádně využít a přemapovat vstupní obraz pomocí map vytvořených z informací o kalibraci pomocí funkce remap, jejímž vstupem jsou již zmíněné mapy a vstupní obraz a výstupem je zkalibrovaný obraz. [17]

```
loadSettings("out_camera_data.xml", map1, map2, imWidth, imHeight);
remap(img, calImg, map1, map2, INTER_LINEAR);
```

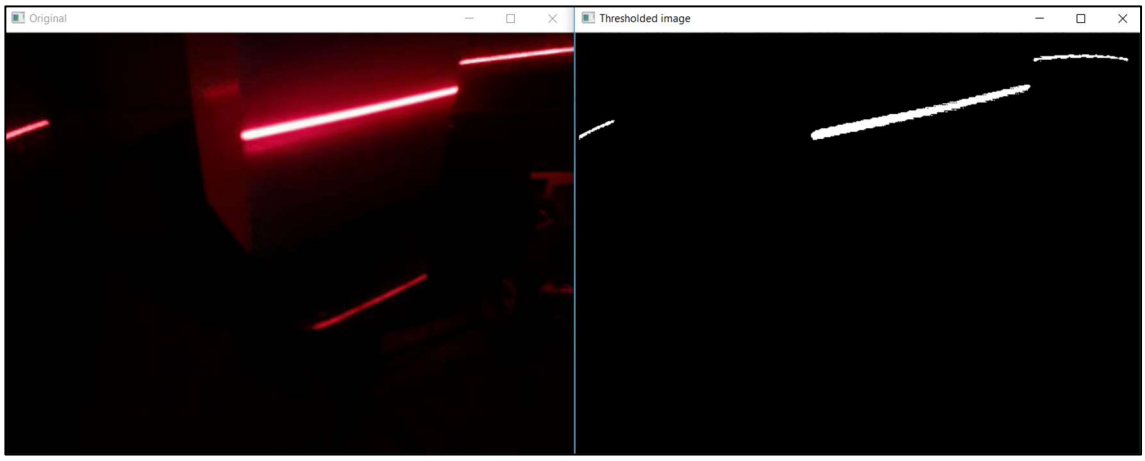
Obr. 33: Kalibrace obrazu



Obr. 34: Originální a zkalibrovaný obraz

5.3.2 Threshold

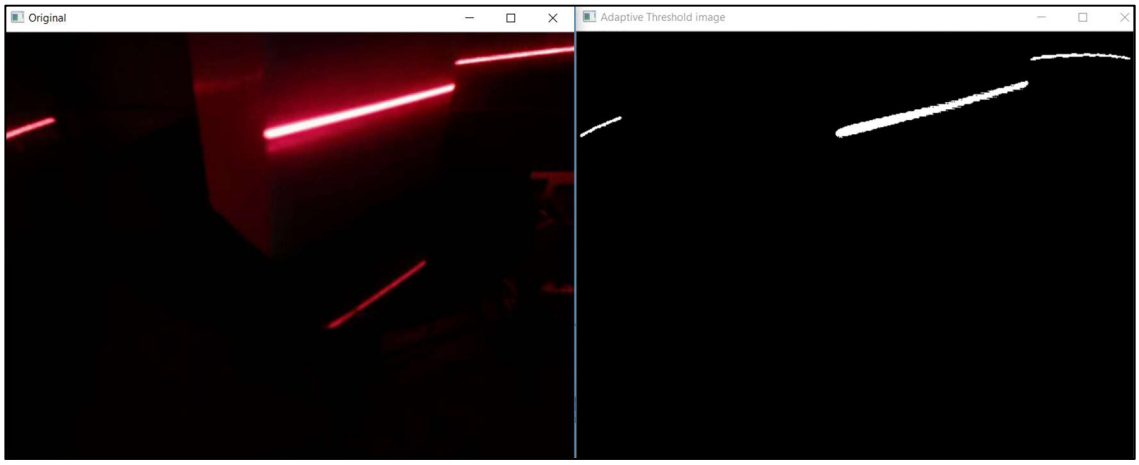
Další použitou funkcí je funkce inRange, která provádí Threshold vstupního snímku, který předtím musí být převeden do HSV (Hue, Saturation, Value) formátu, který se skládá ze tří složek, kterými jsou barevný tón, sytost bary a hodnota jasu. Do funkce je nutno zadat maximální a minimální hodnoty HSV, díky kterým je schopna detekovat požadovanou barvu laserového paprsku. [17]



Obr. 35: Threshold laserového paprsku

5.3.3 Adaptivní Threshold

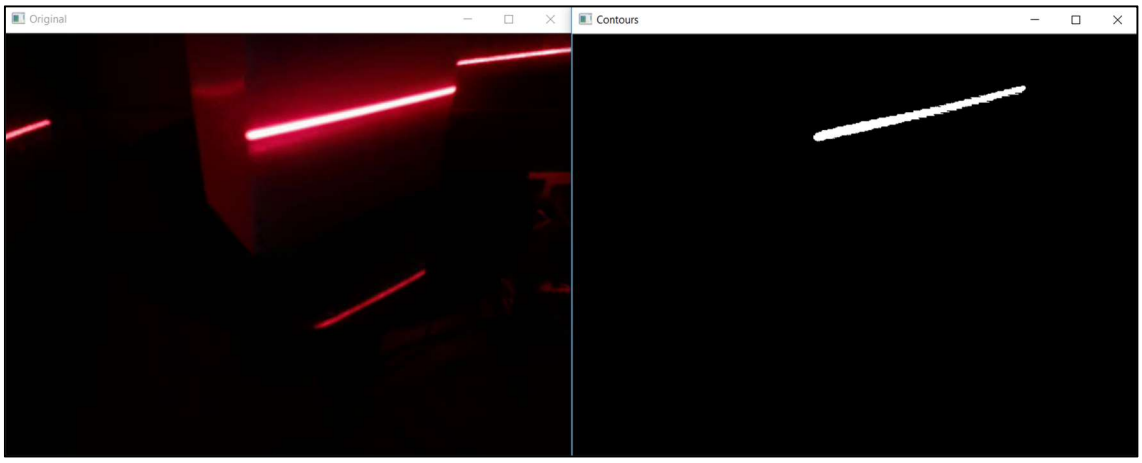
Adaptivní Threshold je použit přímo po Thresholdu. Může se zdát, že Adaptivní Threshold výsledky Thresholdu nevylepší, ale obraz je po použití hladší, což je pro budoucí práci s obrazem výhodnější. Je použit již zmíněný styl GAUSSIAN_C. [17]



Obr. 36: Adaptivní Threshold

5.3.4 Detekce a vykreslení kontur

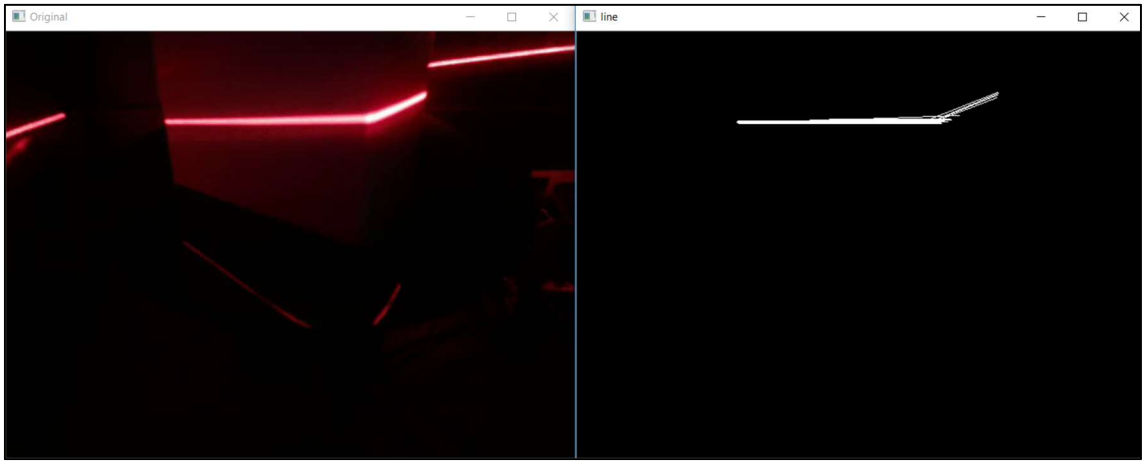
Funkce pro detekci kontur findContours je použita tak, že ve stylu APPROX_SIMPLE najde všechny kontury ve vstupním obraze. Tato metoda je použita po Adaptivním Thresholdu. Poté se pomocí jejich obsahu odfiltrují nadbytečné kontury a ty hledané jsou potom vykresleny pomocí funkce drawContours. Aproximace kontur, kterou nabízí OpenCV, je použita, ale nemá požadované výsledky, proto k tomu byla navržena i vlastní metoda aproximace. [17]



Obr. 37: Detekce kontur

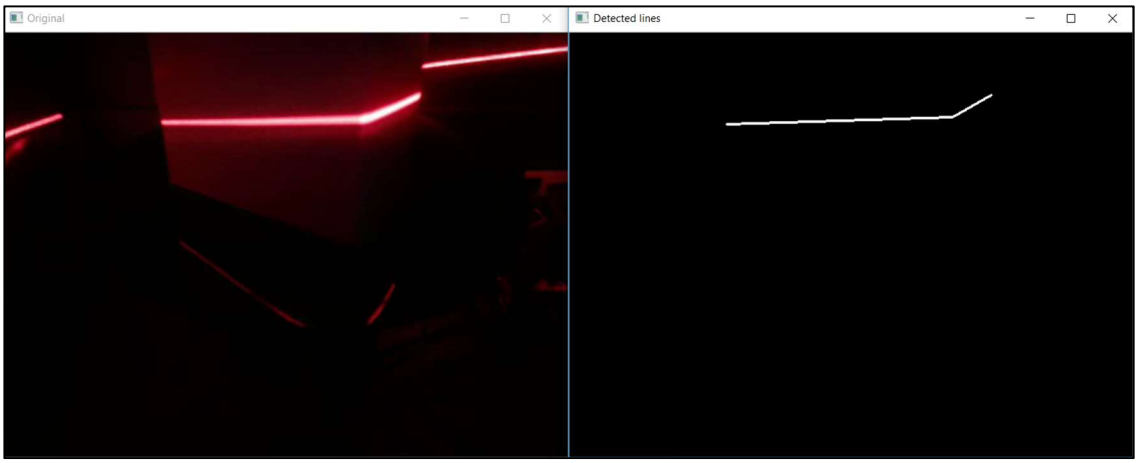
5.3.5 Aproximace kontur

První částí vytvořené metody aproximace kontur je detekce přímek pomocí Randomizované Houghovy transformace HoughLinesP, která po zvolení správných parametrů detekuje přímky v obraze. Aplikuje se na obraz, na který jsou vykresleny vhodné kontury. [17]



Obr. 38: Randomizovaná Houghova transformace pro detekci přímek

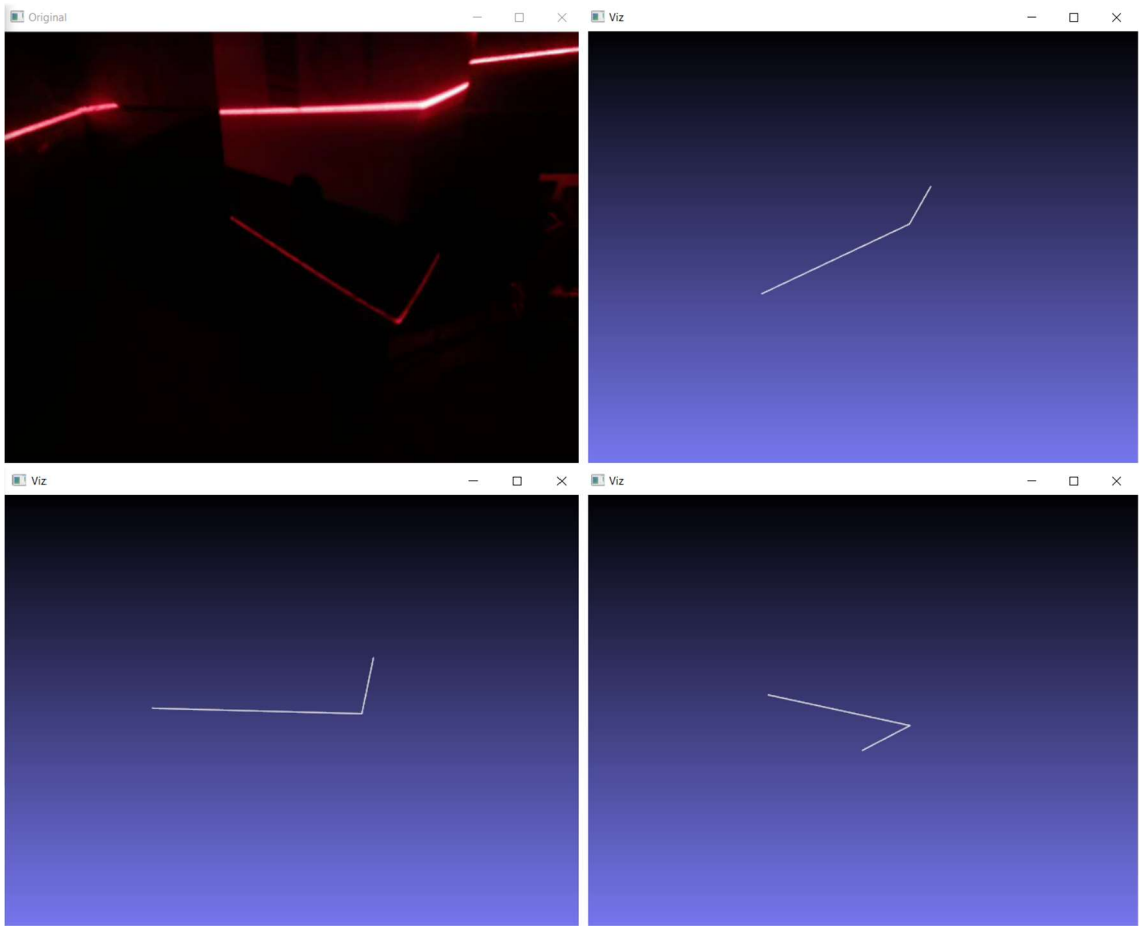
Nalezené přímky jsou následně seskupeny do kategorií, které je rozdělují podle úhlu (pokud je úhel mezi přímkami menší než 3°) a vzdálenosti od svých středů (vzdálenost středů by měla být menší než polovina délky přímek) a tím tvoří jednotlivé celky. Z těchto celků se poté uloží extrémy a ty vykreslí.



Obr. 39: Aproximace kontur

5.3.6 Vizualizace laserového paprsku ve 3D prostoru

Vizualizace je provedena pomocí Viz modulu v OpenCV. Extrémní body jsou získány z předchozí metody a převedeny pomocí matematických metod do 3D prostoru. Tyto body jsou poté spojeny, celý pohled je zobrazen Viz modulem.



Obr. 40: Různé pohledy vizualizace laserového paprsku ve 3D prostoru

6 PRAKTICKÉ EXPERIMENTY

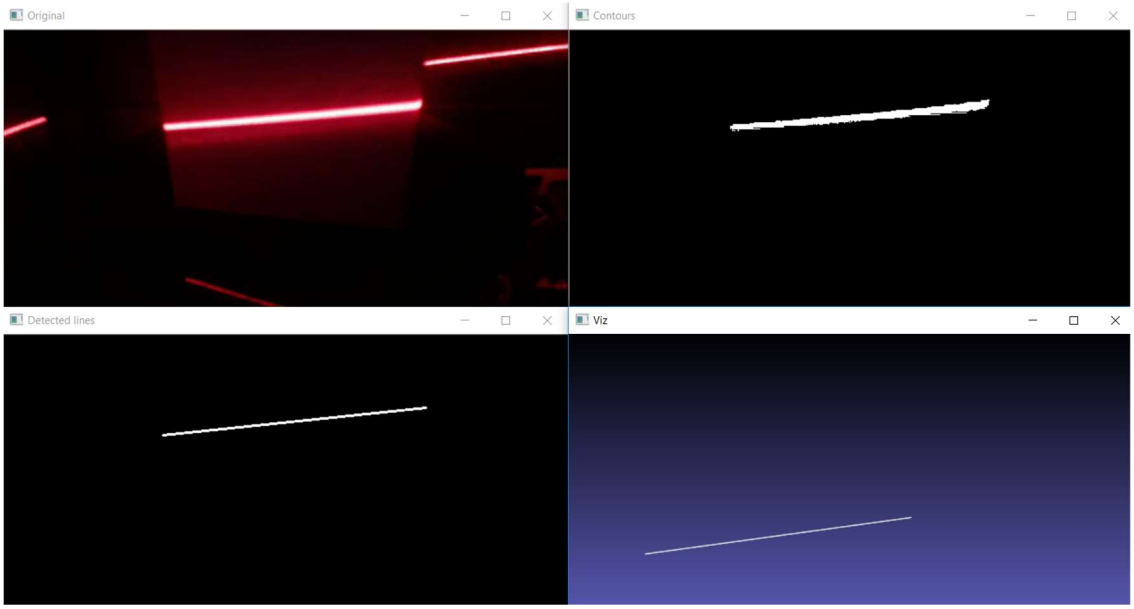
Tato kapitola se zabývá praktickými experimenty a je rozdělena na analýzu FPS a na aplikaci řešení na různé objekty v různých pozicích.

6.1 Aplikace řešení

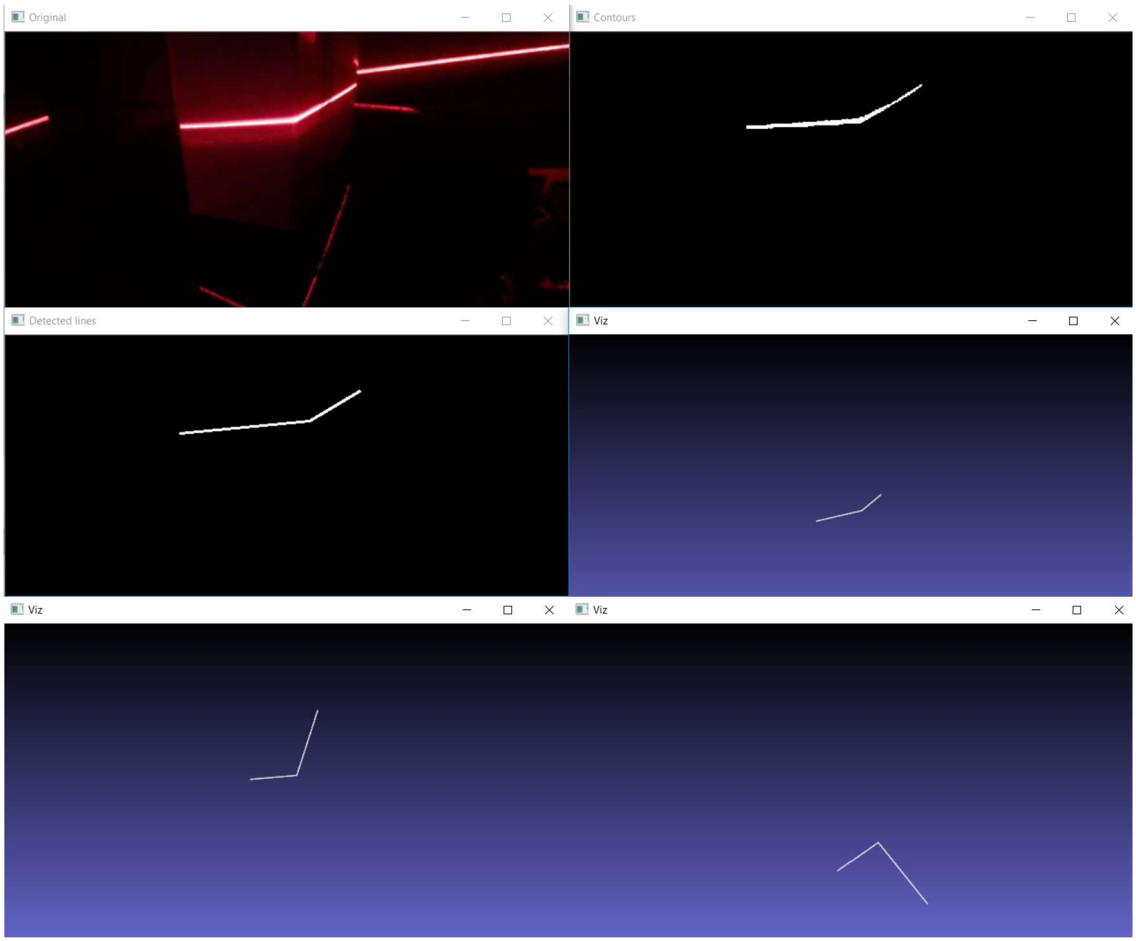
Řešení bylo aplikováno na dva objekty v různých pozicích. Prvním objektem byl bílý kvádr a druhým objektem byl černý válec.

6.1.1 Bílý kvádr

Bílý kvádr je zde ukázán ve dvou pozicích, z vizualizace jde vidět geometrie části objektu, kam svítí laserový paprsek. Z první pozice je patrné, že zobrazená část má rovný povrch bez jakéhokoliv zaoblení, nebo přerušení. Pozice dvě ukazuje roh kvádru, z vizualizace jde také poznat jeho geometrie. Dělí se na dvě spojité úsečky, které zobrazují dvě hrany kvádru. Ty nejsou stejně dlouhé, proto lze usuzovat, že pokud se jedná o pravidelný objekt, tak se jde o kvádr a ne krychli.



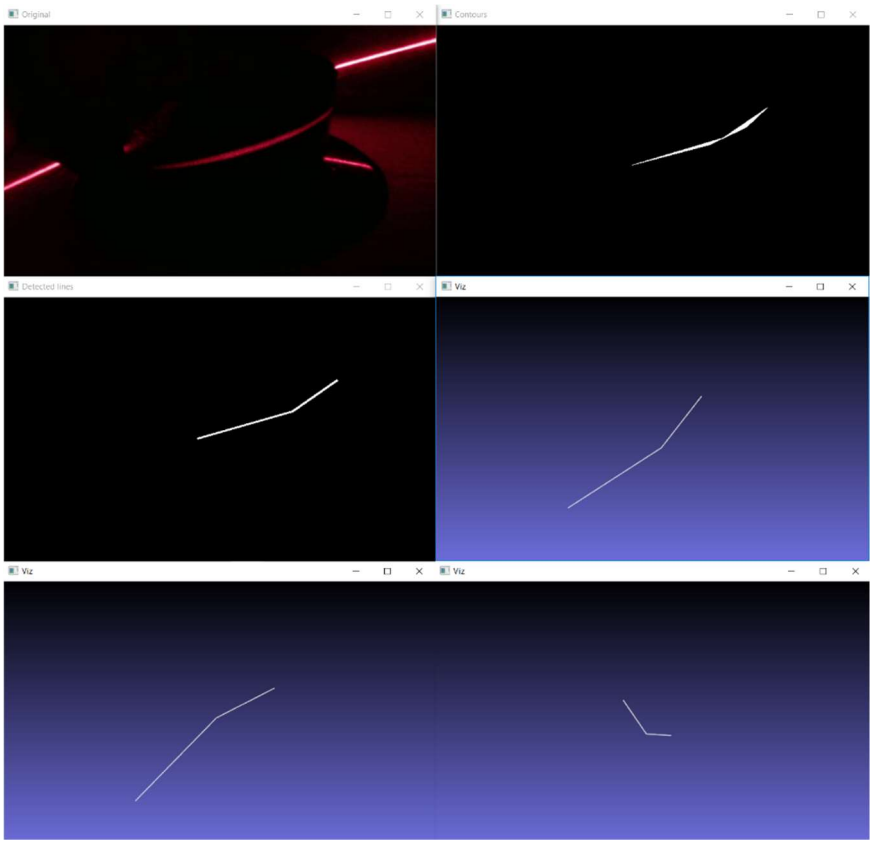
Obr. 41: První pohled na bílý kvádr



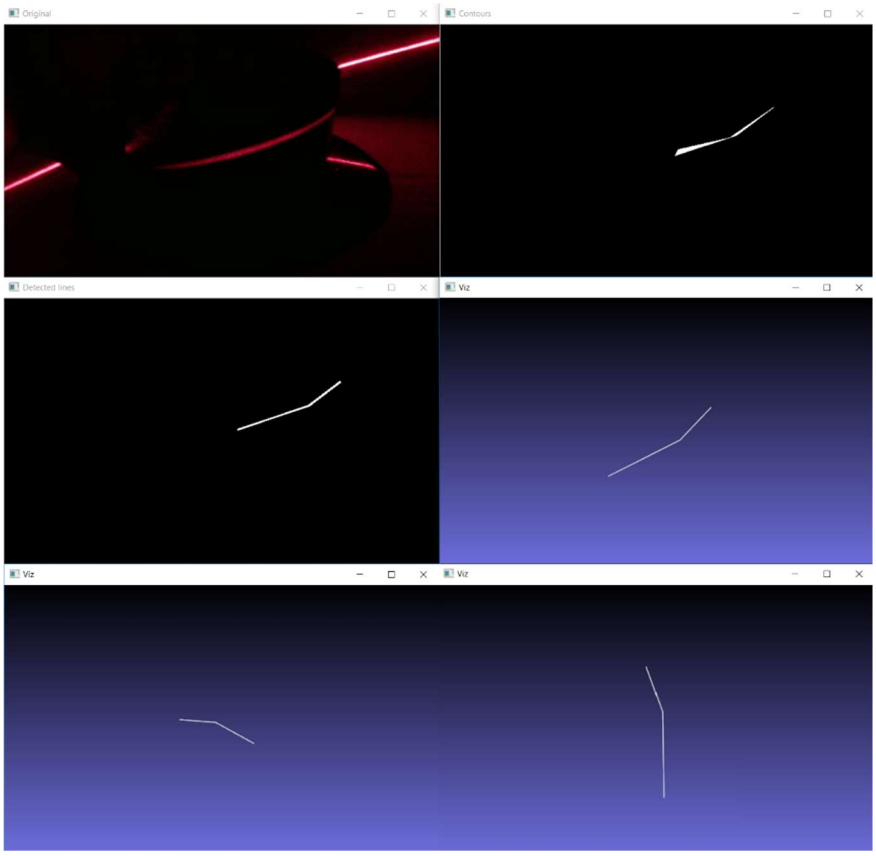
Obr. 42: Druhý pohled na bílý kvádr

6.1.2 Černý válec

Černý válec je zde zobrazen v jedné pozici, ale ve dvou různých snímcích, tím pádem byly také vytvořeny dva pohledy. Platforma byla zastavena, byl vytvořen první pohled, na kterém kvůli aproximaci a metodě vykreslení nejde poznat, že se jedná o válcovitý tvar. Dalo by se uvažovat, že se jedná o roh objektu, který má jako podstavu mnohoúhelník. Proto byl ve stejné pozici vytvořen druhý pohled, na kterém se roh posunul, což značí, že roh není reálný a vznikl kvůli aproximaci. Díky tomu by uživatel mohl usoudit, že se jedná o válcovitý tvar, jistotu by mohl získat až po modelaci objektu, což ale není součástí této bakalářské práce.



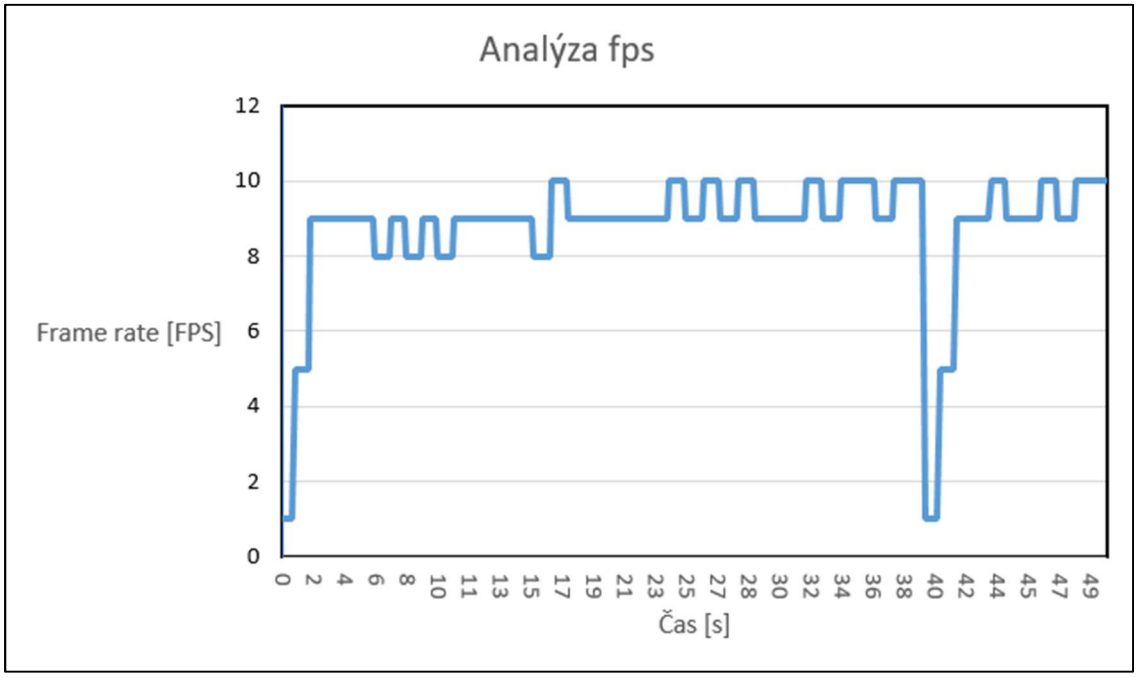
Obr. 43: První pohled na černý válec



Obr. 44: Druhý pohled na černý válec

6.2 FPS analýza

Měření hodnot FPS je provedeno pomocí paralelního vlákna, které do iterační proměnné při načtení prvního snímku přičte jedničku. Paralelní vlákno je ovládáno jednoduchým semaforem. Spuštěné paralelní vlákno je uspáváno na jednu sekundu. Po probuzení načte aktuální hodnotu iterační proměnné, což značí FPS.



Obr. 45: Naměřené hodnoty FPS

Z naměřených hodnot lze zjistit, že nejmenší FPS bylo při spuštění programu, poté byla jeho hodnota víceméně stabilní. Velká změna nastala až kolem 40 s, kdy se obracel pohled ve Viz modulu, tím pádem se propad dal předpokládat. Poté se hodnota zase stabilizovala. Průměrná hodnota FPS byla 8,75 a medián byl 9, což jsou poměrně malé hodnoty, avšak vhledem k náročnosti výpočtů se takový výsledek dal očekávat. Na řešení není požadavek chodu v reálném čase, tudíž se práce optimalizací zabývat nebude.

7 ZÁVĚR

Cílem teto práce bylo navrhnout a prakticky realizovat software, který bude schopen analyzovat deformaci laserového paprsku a na základě této analýzy zobrazit geometrii daného objektu. Na základě této metody je možno vytvořit software pro 3D modelaci objektu, což ale není součástí této bakalářské práce.

Práce se v druhé kapitole zabývá rozdělením používaných technických řešení. Z této rešeršní části je zřejmé, že 3D skenování je mladé odvětví, které se však rychle rozšiřuje. Pokrok je možné vidět hlavně v medicíně, strojírenské diagnostice, strojírenském designu, ale i v herním průmyslu.

Třetí kapitola je zaměřena na zpracování obrazových dat a na porovnání nástrojů, které se v tomto oboru využívají, a byl vybrán nástroj, který byl nejvhodnější pro tuto práci. Součástí kapitoly je i zdůvodnění použití OpenCV knihovny. Důvodem je především její rychlost a možnost 3D vizualizace.

Ve čtvrté kapitole se pracuje s návrhem samotného řešení systému. Je zde uveden postup zpracování obrazových dat za účelem detekce laserového paprsku a jeho vizualizace v 3D prostoru. Je zde navrženo zjednodušené blokové schéma řešení.

Pátá kapitola se zabývá praktickou realizací navrženého řešení. V první části je vysvětlena instalace potřebných knihoven do prostředí Visual Studia. Druhá část se zabývá samotným skenovacím zařízením. Dále následuje přehled metod zpracování obrazu s popisy jejich funkce.

Poslední kapitola je zaměřena na praktické experimenty, kde se provádí analýza FPS (frames per second). Dále je aplikováno řešení na různé objekty v různých pozicích a zkoumá přesnost metody.

Zadané cíle práce byly splněny a navržené řešení je funkční. Je možné na tuto metodu navázat a vytvořit software, který snímaný objekt převede na 3D model.

8 SEZNAM POUŽITÉ LITERATURY

[1] SHERVIN, E. *Mastering OpenCV with Practical Computer Vision Projects*, Packt Publishing Limited.

[2] KAEHLER, A. *Learning OpenCV 3,* O'Reilly Media, Inc, USA, 2015, ISBN 9781491937990.

[3] Technologie skenování ve 3D. In: *SPSKS* [online]. [cit. 2017-04-04]. Dostupné z: http://www.spsks.cz/wp-content/uploads/2016/03/Technologie-skenov%C3%A1n%C3%AD-ve-3D-0.pdf.

[4] ČERMAK, J. *Metody 3D skenování objektů.* Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2015. Vedoucí bakalářské práce Ing. Tomáš Marada, Ph.D.

[5] Princip MRI. In: *fMRI* [online]. [cit. 2017-04-04]. Dostupné z: http://fmri.mchmi.com/main_index.php?strana=13.

[6] ACHARYA, Tinku; AJOY, Ray. *Image processing: principles and applications*. Hoboken: John Wiley & Sons, 2005.

[7] OpenCV. In: *OpenCV* [online]. [cit. 2017-04-10]. Dostupné z: http://www.opencv.org/

[8] Intel IPP. In: *Intel* [online]. [cit. 2017-04-10]. Dostupné z https://software.intel.com/en-us/intel-ipp.

[9] SimpleCV. In: *SimpleCV* [online]. [cit. 2017-04-10]. Dostupné z http://simplecv.org/.

[10] TRIPATHY, Rajashree; DASCHOUDHURY, R. Real-time Face Detection and Tracking Using Haar Classifier on SoC. *International Journal of Electronics and Computer Science Engineering*, 2014, 3: 175-84.

[11] Emgu CV. In: *Emgu CV* [online]. [cit. 2017-04-10]. Dostupné z http://www.emgu.com/.

[12] BoofCV. In: *BoofCV* [online]. [cit. 2017-04-10]. Dostupné z https://boofcv.org/.

[13] Computer Vision System Toolbox. In: *MathWorks* [online]. [cit. 2017-04-10]. Dostupné z https://www.mathworks.com/products/computer-vision.html.

[14] Shi, Shin. *Emgu Cv Essentials*. Community Experience Distilled. ackt Publishing: P, 2013. ISBN 9781783559527.

[15] OPRAVIL, J. *Kalibrace kamery na základě přímkových úseků.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. Počet stran 56. Vedoucí bakalářské práce Ing. Miloslav Richter, Ph.D.

[16] Sférická vada. In: *Fotoroman* [online]. [cit. 2017-05-04]. Dostupné z http://fotoroman.cz/glossary/2_sfera.htm.

[17] Opencv Documentation. In: *Docs.opencv.org* [online]. San Francisco: OpenCV team, c2011- 2014 [cit. 2017-03-23]. Dostupné z: http://docs.opencv.org/2.4.13.2/index.html

[18] Viz. In: *Developers club* [online]. [cit. 2017-05-09]. Dostupné z http://developers-club.com/posts/217021/.

[19] CMake. In: *CMake* [online]. [cit. 2017-05-09]. Dostupné z https://cmake.org/.

[20] VTK. In: *VTK* [online]. [cit. 2017-05-09]. Dostupné z http://www.vtk.org/.

[21] How to build 'opencv_viz' module. In: *OpenCV* [online]. [cit. 2017-05-09]. Dostupné z http://answers.opencv.org/question/32502/opencv-249-viz-module-not-there/?answer=32847#post-id-32847.

[22] Arduino shop. In: *Arduino shop* [online]. [cit. 2017-05-09]. Dostupné z http://arduino-shop.cz/.

[23] Arduino. In: *Arduino* [online]. [cit. 2017-05-09]. Dostupné z https://www.arduino.cc/.

[24] Logitech HD Pro Webcam C920. In: *Logitech* [online]. [cit. 2017-05-13]. Dostupné z https://www.logitech.com/.

[25] Vše pro 3D tisk. In: *Vše pro 3D tisk* [online]. [cit. 2017-05-13]. Dostupné z https://www.vsepro3dtisk.cz.

9 SEZNAM OBRÁZKŮ

10 SEZNAM ZÁKLADNÍCH POJMŮ A ZKRATEK

1D – jednorozměrný

2D – dvojrozměrný

3D – trojrozměrný

Aproximace – přiblížení

C a C++ – programovací jazyky

C# – programovací jazyk

CMOS – Complementary Metal–Oxide–Semiconductor, doplňkový polovodič na bázi kovu a oxidu, technologie používána ve většině integrovaných obvodech (IO je spojení jednoduchých součástek, které vykonávají složitější funkci)

Derivace – základní pojem matematické analýzy

Euklidovský prostor – intuitivní představa prostoru

FPS – snímky za sekundu

Gradient – směr změny

Homografický – stejný

ICSP – In-Circuit Serial Programming, protokol sériového programování MCU

IDE – vývojové prostředí

Java – programovací jazyk

MCU – mikroprocesor

Offline – nepřipojen k síti

PWM – pulzně šířková modulace, diskrétní modulace pro přenos analogového signálu pomocí dvouhodnotového signálu

Python – programovací jazyk

Radiální faktor – středový faktor

Ruby – programovací jazyk

Tangenciální faktor – tečný faktor (tečna je přímka s jedním bodem dotyku s křivkou)

Trackovací zařízení – sledovací zařízení

USB – univerzální sériová sběrnice, používaný způsob připojení k počítači

XML – eXtensible Markup Language, rozšiřitelný značkovací jazyk