



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

**3D laserový skener pro skenování malých
tvarových objektů**

3D LASER SCANNER FOR SMALL OBJECTS SCANNING

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Tomáš Konečný

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Tomáš Marada, Ph.D.

BRNO 2017

Zadání bakalářské práce

Ústav:	Ústav automatizace a informatiky
Student:	Tomáš Konečný
Studijní program:	Strojírenství
Studijní obor:	Aplikovaná informatika a řízení
Vedoucí práce:	Ing. Tomáš Marada, Ph.D.
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:

3D laserový skener pro skenování malých tvarových objektů

Stručná charakteristika problematiky úkolu:

Cílem bakalářské práce je navrhnout a realizovat 3D skener, který bude využívat liniového laseru a kamery pro skenování malých objektů. Stěžejní částí práce je popis použitých metod pro získávání a rekonstrukci 3D dat, volba vhodného programu pro snímání a analýzu dat z kamery, případně tvorba vlastního za použití např. OpenCV knihovny.

Cíle bakalářské práce:

1. Seznamte se s metodami 3D skenování a realizacemi 3D laserových skenerů na internetu.
2. Seznamte se se stávající konstrukcí a elektronikou 3D skeneru.
3. Proveďte nezbytné mechanické a elektronické změny ve stávající konstrukci.
4. Nastudujte a přehledně popište možnosti kalibrace použité kamery.
5. Nastudujte a přehledně popište metodu triangulace, která bude použita pro získávání dat.
6. Proveďte volbu vhodného programu pro snímání a analýzu dat z kamery, případně proveďte tvorbu vlastního za použití např. OpenCV knihovny nebo Matlabu.
7. Navržené řešení realizujte, demonstруйте jeho funkčnost a zhodnoťte dosažené výsledky.

Seznam doporučené literatury:

VODA, Zbyšek. Průvodce světem Arduina. Bučovice: Martin Stříž, 2015. ISBN 978-80-87106-90-7.

Instructables - DIY How To Make Instructions [online]. Copyright © 2016 Autodesk, Inc. [cit. 01.12.2016]. Dostupné z: <http://www.instructables.com/id/3-D-Laser-Scanner/>

Instructables - DIY How To Make Instructions [online]. Copyright © 2016 Autodesk, Inc. [cit. 01.12.2016]. Dostupné z: <http://www.instructables.com/id/Using-Meshlab-to-Clean-and-Assemble-Laser-Scan-Dat/>

[online]. Copyright © [cit. 01.12.2016]. Dostupné z: <https://hci.rwth-aachen.de/fabscan>

[online]. Copyright © [cit. 01.12.2016]. Dostupné z:
<http://hci.rwth-aachen.de/materials/publications/lukas2015a.pdf>

[online]. Copyright © [cit. 01.12.2016]. Dostupné z: <http://mesh.brown.edu/byo3d/notes/byo3D.pdf>

[online]. Dostupné z: <http://georgepavlidis.info/research/LaserScanningAndTriangulation.php>

[online]. Copyright © [cit. 01.12.2016]. Dostupné z:
https://alicedownthecoffeepot.files.wordpress.com/2014/04/documentation_laser_scanner1.pdf

Index of /land/courses/ece4760/FinalProjects/s2009/dat38. [online]. Dostupné z:
<http://people.ece.cornell.edu/land/courses/ece4760/FinalProjects/s2009/dat38/>

[online]. Dostupné z: <https://bib.irb.hr/datoteka/170686.p126.pdf>

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

V Brně, dne

L. S.

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

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

ABSTRAKT

Tato práce je zaměřená především na realizaci 3D skeneru, který využívá liniového laseru a webové kamery. Bude zde popsána kalibrace použité webové kamery, metoda pro získávání a rekonstrukci 3D dat a vytvořený program pro snímání a analýzu dat z kamery.

ABSTRACT

This thesis is focused on realization of 3D scanner, which uses line laser and webcam. Next there will be described the camera calibration, the used method for gain and reconstruction 3D data and created program for scanning and analysis data from webcam.

KLÍČOVÁ SLOVA

3D laserový skener, kalibrace kamery, metoda triangulace, matlab

KEYWORDS

3D laser scanner, camera calibration, triangulation, matlab

BIBLIOGRAFICKÁ CITACE

KONEČNÝ, T. *3D laserový skener pro skenování malých tvarových objektů*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2017. 44 s. Vedoucí bakalářské práce Ing. Tomáš Marada, Ph.D.

.

ČESTNÉ PROHLÁŠENÍ

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

V Brně dne 23. 5. 2017

.....

Tomáš Konečný

OBSAH

1	ÚVOD	13
2	METODY 3D SKENOVÁNÍ	15
2.1	Kontaktní skenery	15
2.1.1	Destruktivní	15
2.1.2	Nedestruktivní	15
2.2	Bezkontaktní skenery.....	16
2.2.1	Snímající pouze vnější geometrii.....	16
2.2.2	Snímající i vnitřní geometrii	16
3	3D LASEROVÉ SKENERY NA INTERNETU	17
3.1	FabScan CUBE	17
3.2	MakerBot Di gitizer	17
3.3	BQ Ciclop	18
3.4	Matter and Form 3D Scanner.....	18
4	KONSTRUKCE 3D SKENERU	19
4.1	Liniový laser	19
4.2	Arduino UNO a CNC shield.....	20
4.3	Krokové motory	20
4.4	Otáčecí stůl	21
4.5	Webová kamera	21
5	ZÍSKÁVÁNÍ SOUŘADNIC	22
5.1	Metoda triangulace	22
5.1.1	Vymezení triangulační metody.....	22
5.2	Kalibrace kamery	23
5.2.1	Zkreslení obrazu čočkami	23
5.2.2	Kalibrační algoritmus	25
5.2.3	Aplikace v programu Matlab	25
5.3	Aplikace triangulační metody na dírkový model.....	28
5.3.1	Sestavení rovnic	30
6	PROGRAM PRO SNÍMÁNÍ A ANALÝZU DAT	32
6.1	Vytvoření objektu Arduina	32
6.2	Inicializace laseru	32
6.3	Definice rozměrů a konstant	33
6.4	Nastavení kamery	33
6.5	Nastavení laseru do výchozí pozice.....	34
6.6	Vytvoření snímků	35
6.7	Segmentace barev a výpočet souřadnic bodů	35
6.7.1	Segmentace barev	35
6.7.2	Výpočet souřadnic bodů	37
6.8	Kompenzace šířky světelného pruhu	38
6.9	Zvýšení přesnosti skenování otáčením laseru.....	38
6.10	Otáčení se stolem.....	39
6.11	Zobrazení 3D modelu	39
7	ZHODNOCENÍ A DISKUZE	40
8	ZÁVĚR	41

1 ÚVOD

Tato diplomová práce se bude mimo jiné zabývat základním rozdělením 3D skenerů. Jsou zde stručně popsány kontaktní destruktivní i nedestruktivní skenery a také bezkontaktní skenery snímající pouze vnější geometrii a skenery snímající i vnitřní geometrii.

Dále se zabývá hodnocením 3D skenerů, které jsou zveřejněné na internetu a jsou podobné skeneru, který bude použit k této práci. Mezi tyto skenery patří FabScan CUBE, MakerBot Digitizer, BQ Ciclop a Matter and Form 3D Scanner.

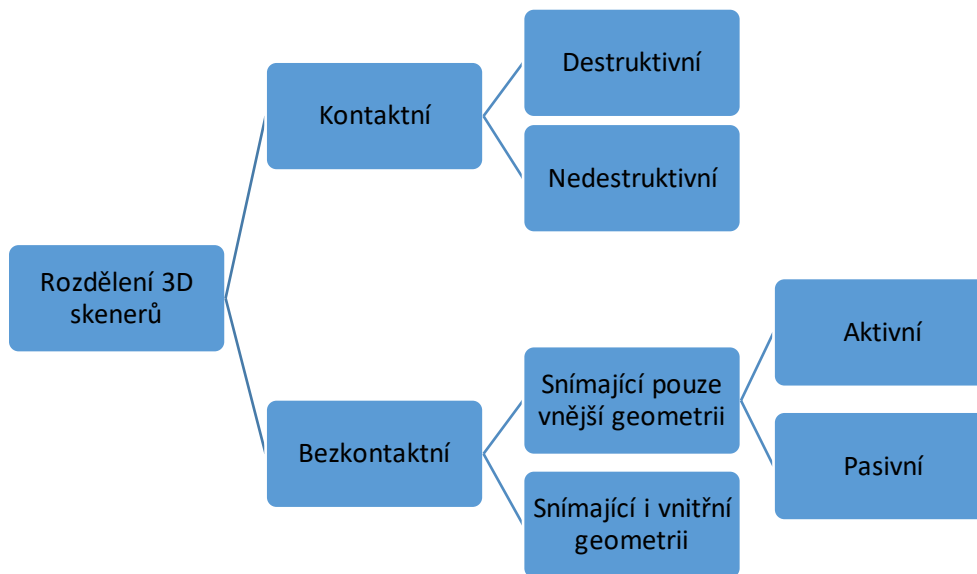
Kapitola 4 popisuje nejdůležitější části použitého 3D skeneru, mezi které patří liniový laser, webkamera Logitech HD Pro Webcam C920, Arduino UNO s rozšiřující deskou CNC, otáčecí stůl a krokové motory, které otáčí s liniovým laserem a otáčecím stolem.

Stěžejní částí této práce je popis triangulační metody a její aplikace na dírkový model, kalibrace použité kamery s pomocí aplikace v programovacím prostředí Matlab a popis vad optických čoček použitých v kamerách. Dále se zabývá aplikací triangulační metody na dírkový model a sestavením rovnic pro výpočet souřadnic neznámého bodu v prostoru.

V závěru jsou popsány jednotlivé části programu pro snímání a analýzu dat, mezi které patří například definice známých rozměrů a konstant, vytváření snímků, ze kterých je odstraněno zkreslení způsobené vadami optických čoček, metoda segmentace barev, která je použita k detekci liniového laseru, výpočty souřadnic bodů pomocí odvozených rovnic, kompenzace šířky světelného pruhu, zvyšování přesnosti pomocí změny úhlu natočení laseru a nakonec zobrazení 3D modelu.

2 METODY 3D SKENOVÁNÍ

Většina metod skenování se nazývá podle technologií, které jsou použity při realizaci skenerů. Nejčastěji se skenery dělí podle způsobu získávání dat. Základní dělení skenerů je znázorněno na obr. 1. [1]



Obr. 1: Rozdělení skenerů

2.1 Kontaktní skenery

Kontaktní 3D skenery využívají dotykovou sondu, která snímá povrch skenovaného objektu. Dotyková sonda se zpravidla skládá z kuličkového hrotu a snímačů citlivých na vychýlení hrotu. Při kontaktu se skenovaným objektem dojde k vychýlení hrotu a zaznamená se aktuální poloha os, tedy jeden bod. [2]

2.1.1 Destruktivní

Tyto skenery jsou schopné skenovat kromě vnějšího povrchu také vnitřní strukturu. Skenování probíhá tak, že objekt je po malých vrstvách frézován a každá vrstva je naskenována. Skenování trvá do doby, než objekt zcela zanikne. [1]

2.1.2 Nedestruktivní

Během skenování je snímán pouze povrch, který není nijak poškozován. [1]

2.2 Bezkontaktní skenery

Bezkontaktní skenery jsou jednoduché, rychlé a přesné. V současnosti patří mezi nejčastěji používané 3D skenery. [1]

2.2.1 Snímající pouze vnější geometrii

Tyto skenery získávají pouze data o povrchu objektu. Do této skupiny patří zejména akustické a optické skenery, které patří mezi nejčastěji používané. Optické skenery můžeme dále dělit na aktivní a pasivní.

U aktivních optických skenerů je emitována energie, která může být ve formě viditelného světla nebo jakéhokoliv elektromagnetického záření. Vysílaný paprsek dopadá na povrch objektu a následně se odráží do snímače. Hlavní výhody těchto skenerů jsou rychlost, mobilita a přesnost. Pro získávání souřadnic se u aktivních skenerů využívají především metody měření doby letu světla, optické interferometrie nebo triangulace. Triangulace se liší podle typu použitého světla na 1D triangulaci při použití světelného paprsku, 2D triangulaci při použití světelného pruhu a 3D triangulaci při použití strukturovaného světla.

Oproti aktivním skenerům nemají pasivní skenery zdroj osvětlení. Data se získávají minimálně ze dvou snímků, které se liší natočením objektu, nebo změnou polohy snímače. Mezi pasivní metody patří stereovidění, analýza pohybu, zoomování, stínování, využívání kontur, změny ohniska a textura objektu. [1]

2.2.2 Snímající i vnitřní geometrii

Mohou získávat informace o vnitřní geometrii objektu. Patří sem například magnetické a transmisivní skenery. Magnetický skener funguje na stejném principu jako magnetická rezonance používaná ve zdravotnictví. Transmisivní skener využívá počítačové tomografie známé jako CT. Při skenování je ale použita vyšší intenzita záření, než je tomu u běžných zdravotnických CT. [1]

3 3D LASEROVÉ SKENERY NA INTERNETU

Tato část práce popisuje 3D skenery zveřejněné na internetu, které mají podobnou konstrukci jako 3D skener použitý k této práci.

3.1 FabScan CUBE

Skener použitý k této práci se nejvíce podobá skeneru s názvem FabScan CUBE (obr. 2). Skládá se z liniového laseru, webové kamery, otáčecího stolu, arduina a dřevěné konstrukce. Hlavní výhodou tohoto skeneru je v tom, že je situovaný do uzavřeného prostoru, což pomáhá vyrovnávat hladinu osvětlení, a tím zabraňuje deformaci při skenování. Jeho cena se pohybuje kolem 130 €. [3]



Obr. 2: FabScan CUBE [4]

3.2 MakerBot Digitizer

Makerbot Digitizer (obr. 3) skenuje objekty dvěma liniovými lasery a kamerou. Objekty mohou mít výšku a průměr 20 cm. Kompletní oskenování na střední rozlišení trvá 12 minut, přičemž otáčecí stůl vykoná 800 kroků. Makerbot Digitizer není schopen snímat barvu či texturu povrchu. Skenovací software s názvem Makerware for Digitizer běží pouze pod operačním systémem Windows. Cena tohoto skeneru je asi 950 €. [3]



Obr. 3: MakerBot Digitizer [3]

3.3 BQ Ciclop

Skener BQ Ciclop, zobrazený na obr. 4, používá dva červené liniové lasery a standardní webkameru. BQ vyvinula vlastní skenovací aplikaci Hours 8. Hours 8 je open source aplikace napsaná v Pythonu, která je kompatibilní s operačními systémy Windows nebo Linux. [3]



Obr. 4: BQ Ciclop [3]

3.4 Matter and Form 3D Scanner

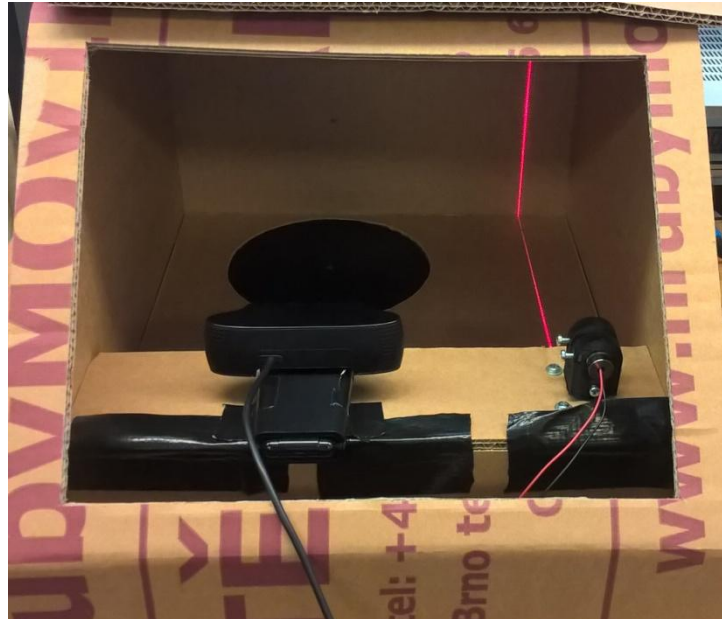
Matter and Form 3D Scanner se skládá ze dvou liniových laserů a zabudovaného kamerového modulu. Když není skener v chodu, je možné ho složit (obr. 5). Kamera s laserem se mohou pohybovat nahoru a dolů, takže mohou skenovat větší objekty. Skener je dodáván kompletně sestavený a jeho cena je \$600. Doba skenování trvá kolem pěti minut a může být barevné. Skenování s vyšším rozlišením může trvat i několik hodin. Software je kompatibilní s operačním systémem Windows i Mac OS. [3]



Obr. 5: Složený (vpravo) a rozložený (vlevo) Matter and Form 3D Scanner [5]

4 KONSTRUKCE 3D SKENERU

V následující části práce budou popsány nejdůležitější prvky použitého 3D skeneru, vyobrazeného na obr. 6.

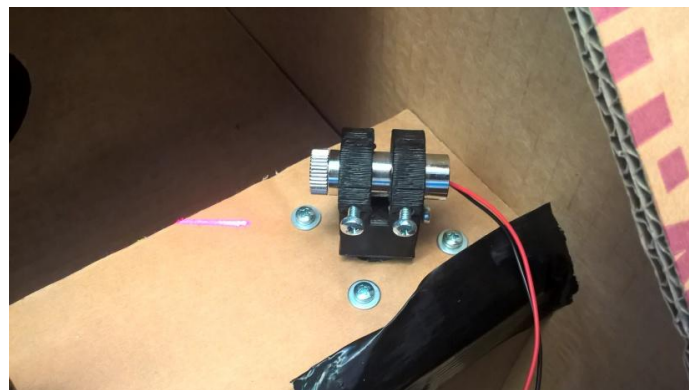


Obr. 6: Použitý 3D skener

4.1 Liniový laser

Liniový laser je zařízení, které se skládá z bodového laseru a optických čoček. Pomocí Powellových nebo cylindrických čoček je z bodu vytvořena linie. [6]

Liniový laser je upevněn do držáku (obr. 7), který je připojen ke krokovému motoru tak, že s ním může motor otáčet. Na zadní straně krabice je zakreslena svíslá čára, podle které bude laser orientován. Laser je připojen k Arduino na portu A4, přes který bude zapínán a vypínán.



Obr. 7: Liniový laser v držáku

4.2 Arduino UNO a CNC shield

Arduino UNO je vývojová deska, jejímž základním prvkem je mikroprocesor ATmega328P. Arduino má 14 pinů, z nichž může být každý použit jako digitální vstup nebo výstup. Z těchto 14 pinů může být 6 použito jako PWM výstup. Vývojová deska dále obsahuje 16 MHz krystal, USB a ICSP rozhraní, napájecí konektor a resetovací tlačítko. Arduino může být napájeno buď přes USB rozhraní, nebo přes napájecí konektor.[7]

Pro ovládání krokových motorů Arduinem, bylo nutné připojit Arduino CNC shield V3.10, neboli rozšiřující desku CNC (obr. 8), která obsahuje ovladače krokových motorů. Díky této desce je možné ovládat až 4 krokové motory. Protože napájení desky pouze Arduinem by nebylo dostatečné k rozpohybování motorů, je třeba CNC desku připojit k 12V zdroji napětí.

Arduino Uno s připojenou CNC deskou byly uloženy pod rovinou otáčecího stolu. K deskám je připojen jak krokový motor, který otáčí stolem, tak i krokový motor otáčející s liniovým laserem. Dále byl k Arduino připojen liniový laser, aby bylo možné ovládat jeho zapínání a vypínání. Programování probíhalo pomocí USB rozhraní, přes které bylo Arduino připojeno k počítači.



Obr. 8: Arduino UNO s CNC shieldem (v pravém rohu)

4.3 Krokové motory

Krokové motory jsou synchronní točivé stroje ovládané impulsy stejnosměrného proudu. Každý impuls napájí pólovou dvojici, která vytvoří magnetické pole. Rotor, který se nachází v tomto magnetickém poli, se natočí k napájené pólové dvojici, a tím vykoná jeden krok. Počet těchto kroků závisí na počtu pólových dvojic umístěných na statoru. [8]

Oba použité krokové motory jsou stejné a velikost jejich fyzického kroku je $1,8^\circ$. Pro plynulejší přechod mezi kroky existuje i tzv. mikrokrokování, které může jednotlivé kroky ještě rozdělit na poloviny, čtvrtiny, osminy nebo šestnáctiny. Krokový

motor připevněný k liniovému laseru byl připojen k Arduino na pinu D2. Motor, který otáčí stolem, byl připojen na pinu D3.

4.4 Otáčecí stůl

Otáčecí stůl byla použita deska z plastu kruhového průřezu vytištěná na 3D tiskárně a připojená ke krokovému motoru, který s ní otáčí.

4.5 Webová kamera

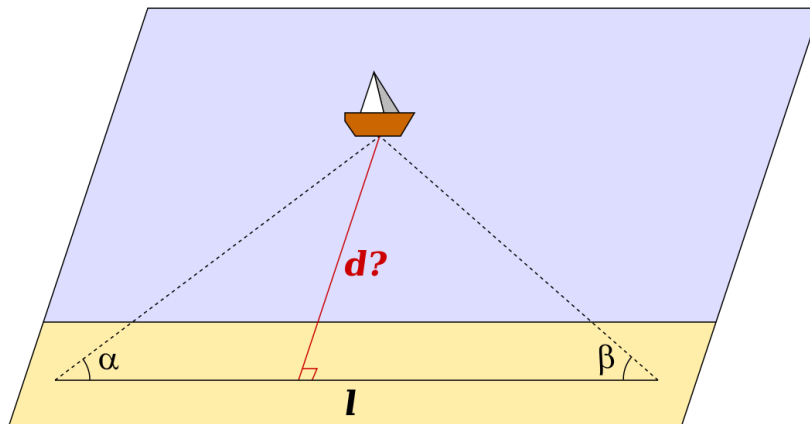
Jako obrazový snímač byla použita webkamera Logitech HD Pro Webcam C920 s maximálním rozlišením 2304x1536. Webkamera má zorný úhel 78° a ohniskovou vzdálenost 3,67 mm a byla připojena k počítači pomocí rozhraní USB. [9]

Střed kamery i laser byly umístěny ve výšce 79 mm nad rovinou otáčecího stolu. Vzdálenost mezi laserem a kamerou byla 140 mm. Referenční značka, která je zakreslena na zadní straně krabice, byla ve stejné výšce jako kamera.

5 ZÍSKÁVÁNÍ SOUŘADNIC

5.1 Metoda triangulace

Metoda spočívá v tom, že je sestaven pomyslný trojúhelník, ve kterém je známá délka jedné strany a oba úhly přilehlé k této straně (viz obr. 9). Souřadnice se poté vypočítají pomocí trigonometrických vzorců. [10]



Obr. 9: Triangulační trojúhelník [10]

5.1.1 Vymezení triangulační metody

Protože se 3D skener skládá z liniového laseru a webové kamery, je třeba aplikovat triangulační metodu právě na tato zařízení. Aby bylo možné provádět výpočty, bylo nutné převést reálná zařízení na matematické modely. Liniový laser je matematicky popsán jako rovina. Reálnou kameru je možné převést na dírkový model, kde ji lze chápat jako dírkovou komoru, za předpokladu kompenzace tohoto zjednodušení. [10, 11, 12, 13]

Dírková komora

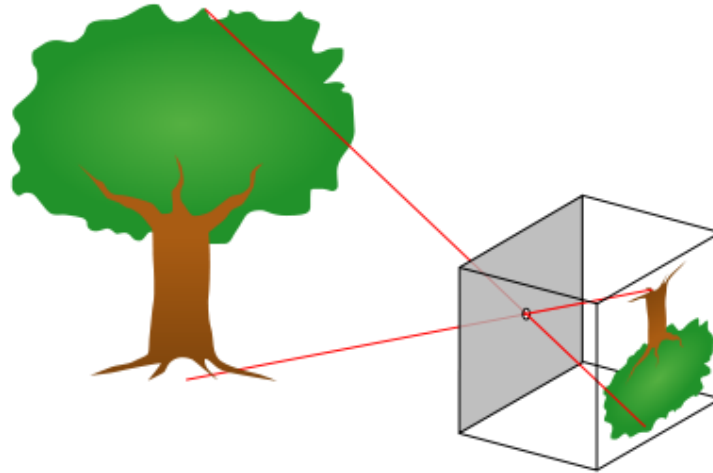
Dírková komora, také známá pod pojmem camera obscura, je zobrazovací zařízení, ve kterém není použita žádná optická čočka. Světlo zde prochází úzkou štěrbinou a vytváří převrácený obraz na zadní straně dírkové komory. [12]

Dírkový model

Dírkový model popisuje matematické vztahy mezi souřadnicemi bodu v prostoru a jeho zobrazením na zobrazovací ploše za předpokladu, že je použita ideální dírková komora. U ideální dírkové komory je štěrbinu popsána jako bod a světlo procházející štěrbinou není zaostřováno pomocí optických čoček. Toto zobrazení je znázorněno na obr. 10. [13]

V modelu není zahrnuto zkreslení nebo rozmazání nezaostřených objektů způsobené čočkami nebo konečnou velikostí štěrbin. Také platí, že většina reálných kamer nemá spojitý souřadný systém snímku. [13]

Některé vlastnosti, které nejsou zahrnuty v modelu, mohou být kompenzovány. Například použitím vhodné transformace souřadného systému snímku může být kompenzováno jak zkreslení způsobené čočkami, tak i nespojitost souřadného systému. Pokud je použita kvalitní kamera, tak dopad ostatních nezahrnutých vlastností je tak malý, že může být zanedbán. [13]



Obr. 10: Zobrazení objektu v dírkové komoře [13]

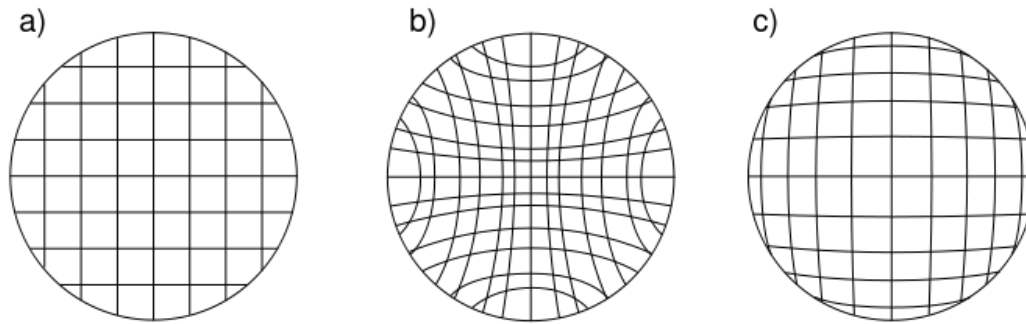
5.2 Kalibrace kamery

Protože v dírkovém modelu není zahrnuto zkreslení, které způsobují vady optických čoček, je třeba tyto vady kompenzovat kalibrací kamery. Ke kalibraci kamery byla použita aplikace v programu Matlab, která vypočítala přibližné hodnoty vnitřních parametrů, vnějších parametrů a koeficienty zkreslení. Aplikace lze použít pro kamery se zorným úhlem do 95° . [14]

5.2.1 Zkreslení obrazu čočkami

Radiální zkreslení

Zkreslení obrazu je způsobeno větším lomem světla při okraji čočky, než v jejím středu. Je-li obraz více zvětšen u okrajů, jedná se o poduškovité zkreslení, pokud je více zvětšen ve středu, jedná se o soudkovité zkreslení. Tento jev je dobře pozorovatelný na tzv. rastru obr. 11. [14, 15]



Obr. 11: a) rastr, b) poduškové zkreslení, c) soudkovité zkreslení [15]

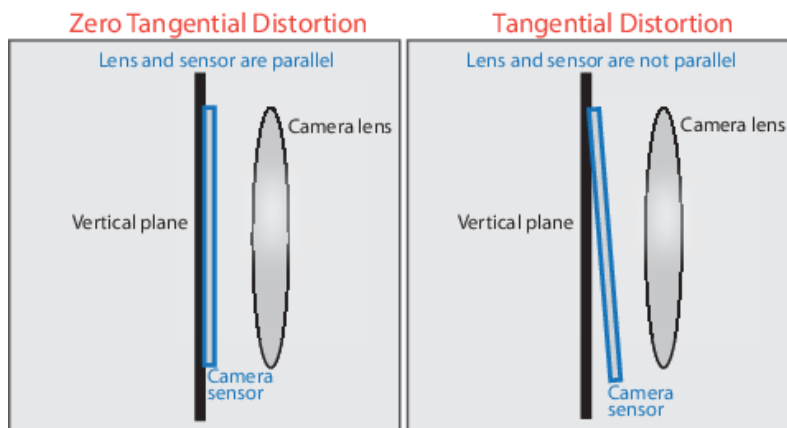
Pro zkreslené body platí:

$$\begin{aligned}x_{zkreslené} &= x(1 + k_1r^2 + k_2r^4 + k_3r^6) \\y_{zkreslené} &= y(1 + k_1r^2 + k_2r^4 + k_3r^6) \\r^2 &= x^2 + y^2\end{aligned}$$

kde x a y jsou nezkreslené souřadnice obrazu, k_1 , k_2 a k_3 jsou koeficienty radiálního zkreslení. Pro kalibraci obvykle stačí dva koeficienty, ale v případě velkého zkreslení, ke kterému dochází například u širokoúhlých čoček, je možné počítat i s třetím koeficientem k_3 . Nezkreslené souřadnice mají počátek v optické ose. [14,15]

Tangenciální zkreslení

Zkreslení obrazu je způsobeno nerovnoběžností snímacího senzoru a optické čočky. Na obr. 12 je znázorněna kamera, která má nulové tangenciální zkreslení, a kamera, u které se projeví tangenciální zkreslení vlivem nerovnoběžnosti senzoru a čočky. [14]



Obr. 12: a) kamera s nulovým tangenciálním zkreslením b) kamera s tangenciálním zkreslením [14]

Pro zkreslené body platí:

$$\begin{aligned}x_{zkreslené} &= x + [2p_1xy + p_2(r^2 + 2x^2)] \\y_{zkreslené} &= y + [p_1(r^2 + 2y^2) + 2p_2xy] \\r^2 &= x^2 + y^2\end{aligned}$$

kde x a y jsou nezkrácené souřadnice obrazu a p_1 a p_2 jsou koeficienty tangenciálního zkreslení. [14]

5.2.2 Kalibrační algoritmus

Kalibrační algoritmus zahrnující dírkový model je dán rovnicí:

$$w \begin{bmatrix} x & y & 1 \end{bmatrix} = \begin{bmatrix} X & Y & Z & 1 \end{bmatrix} \begin{bmatrix} R \\ t \end{bmatrix} K$$

kde w je libovolné měřítko homogenního souřadného systému, x a y jsou obrazové souřadnice udávané v pixelech, X , Y a Z jsou souřadnice bodu v obecném souřadném systému. Mezi **vnější parametry** patří matice R , která popisuje rotaci kamery, a translační vektor t . Mezi **vnitřní parametry** patří souřadnice optického centra (hlavního bodu) c_x a c_y , které jsou uvedeny v pixelech, parametr sklonu os s (při kolmosti os $s=0$), ohnisková vzdálenost F v milimetrech a koeficienty s_x a s_y , které udávají počet pixelů na milimetr. Všechny vnitřní parametry jsou zahrnuty do matice K . [14]

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

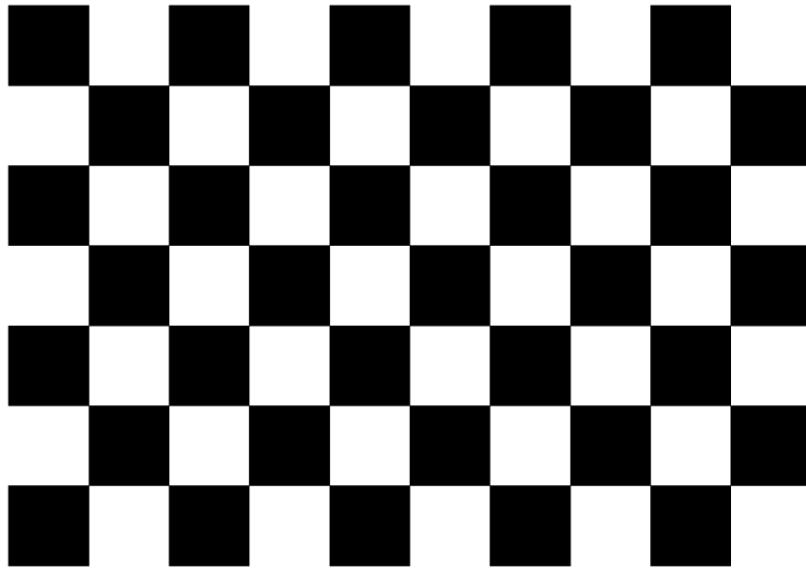
$$(1) f_x = F \cdot s_x$$

$$(2) f_y = F \cdot s_y$$

5.2.3 Aplikace v programu Matlab

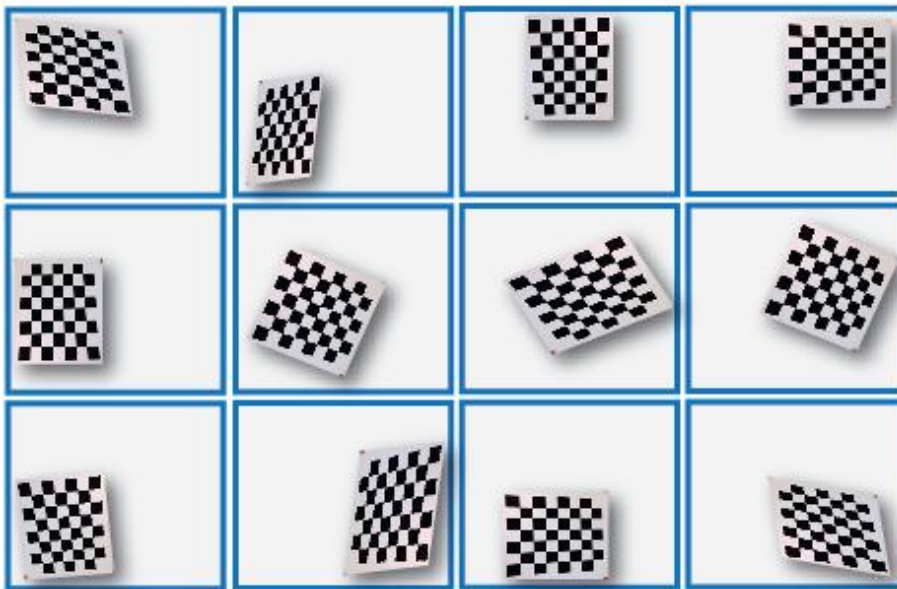
Kalibrace kamery byla provedena pomocí aplikace Camera Calibrator. Ke kalibraci kamery je potřeba kalibrační vzor (obr. 13), který je nutné připevnit na rovnou plochu, protože jakákoli nerovnost může ovlivnit přesnost kalibrace. Také je třeba dbát na to, aby byla kamera zaostřena na kalibrační vzor. Ke kalibraci nesmí být používáno automatické zaostřování, ani se nesmí měnit zoom mezi jednotlivými snímky, protože by se změnila ohnisková vzdálenost. Vzorek je třeba snímat přibližně ze stejné vzdálenosti, kterou následně budeme chtít měřit. Kalibrační vzor nesmí být natočen o více než 45° od osy kamery a musí pokrývat nejméně 20 % snímku. Díky tomu, že má vzor obdélníkový tvar, může aplikace jednoznačně identifikovat osu x , která je určena delší stranou, a osu y , která je určena kratší stranou. Vzorek je třeba nasnímat s různou orientací os vzhledem ke kameře. [14]

Dále je třeba změřit stranu jednoho čtverce vzoru, protože velikost čtverce se může lišit v závislosti na nastavení tiskárny. Aby byla kalibrace dostatečně přesná, je nutné udělat odlišné snímky kalibračního vzoru tak, aby byly otestovány všechny pixely snímacího čipu. Běžně je potřeba 10 až 20 snímků v nekomprimovaném formátu (např. PNG). Pro správný výpočet zkreslení čoček, které se od středu zvětšují a nemusí být rovnoměrné, je vzorek snímán u okrajů snímku. Příklady správného snímání vzoru jsou na obr. 14. [14]



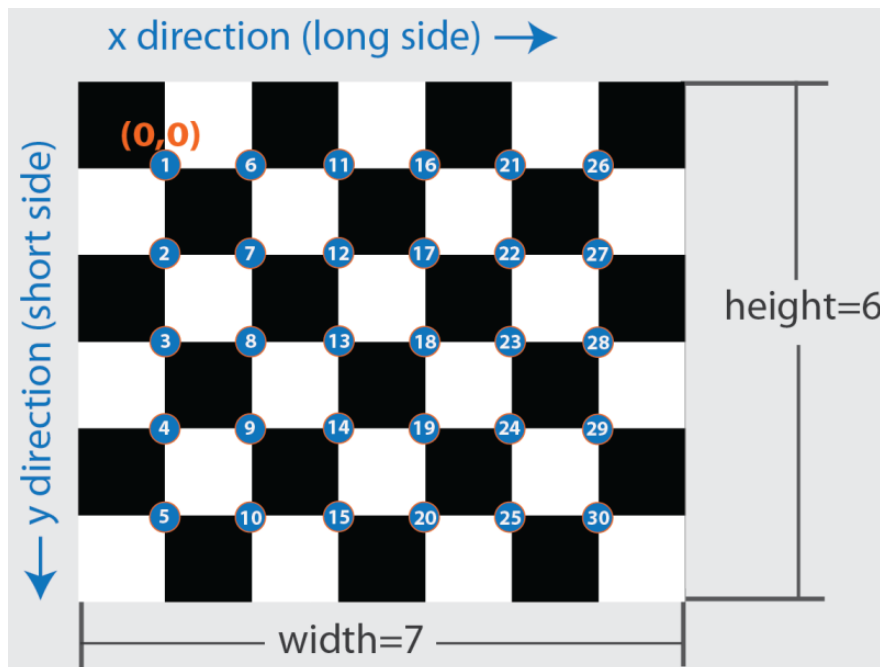
This is a 3x3
OpenCV chessboard
<http://sourceforge.net/projects/opencvlibrary/>

Obr. 13: Kalibrační vzor [16]



Obr. 14: Příklad snímání vzoru [14]

Po nahrání všech snímků a zadání rozměru čtverce byly nastaveny koeficienty, které se mají vypočítat. Aplikace dokáže vypočítat 2 nebo 3 parametry radiálního zkreslení, sklon os a tangenciální zkreslení. Po dokončení analýzy se nám zobrazí počet přijatých a zamítnutých snímků. Důvody k zamítnutí snímku mohou být následující: duplicita snímků, nenalezení vzoru ve snímku, velký úhel natočení vzoru. Na přijatých snímcích program detekuje body definované na obr. 15. [14]

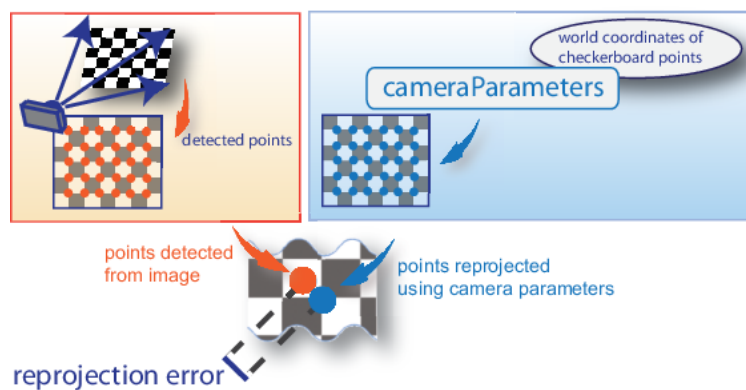


Obr. 15: Kalibrační vzor s detekovanými body [21]

Zhodnocení analýzy

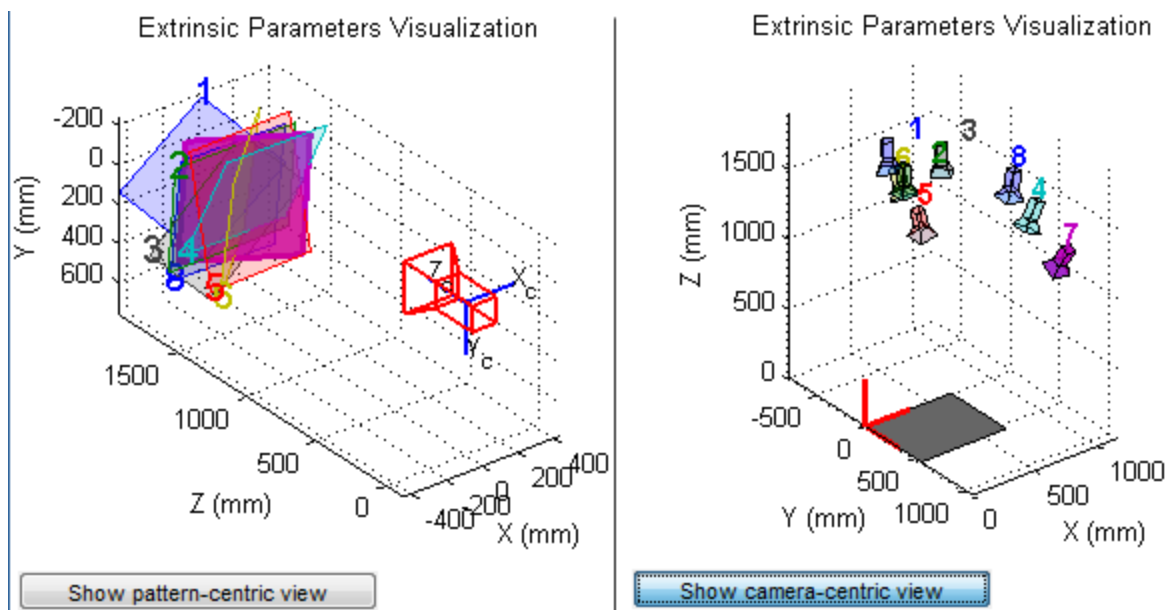
Analýza proběhla v pořádku, jestliže reprojekční chyba nebyla moc velká, ve vizualizaci vnějších parametrů nebyla chyba a snímky po odstranění zkreslení jsou opravdu nezkrácené. [14]

Reprojekční chyba (udávaná v pixelech) je vzdálenost mezi detekovaným bodem a bodem vypočítaným z parametrů kamery (obr. 16). Obecně platí, že chyby, které jsou menší než jeden pixel, jsou přijatelné. [14]



Obr. 16: Reprojekční chyba [14]

Vizualizace vnějších parametrů může být kamerově centrovaná, nebo centrovaná na kalibrační vzor (obr. 17). Pokud je kamera statická a mění se pozice vzoru, používá se vizualizace kamerově centrovaná. Jestliže je statický kalibrační vzor a mění se pozice kamery, využívá se vizualizace centrovaná na vzor. Vizualizace jsou správné, odpovídají-li předpokladům. Například pokud by se vzor ve vizualizaci objevil za rovinou kamery, znamenalo by to chybu. [14]



Obr. 17: Kamerově centrovaná vizualizace (vlevo) a vizualizace centrovaná na vzor (vpravo) [14]

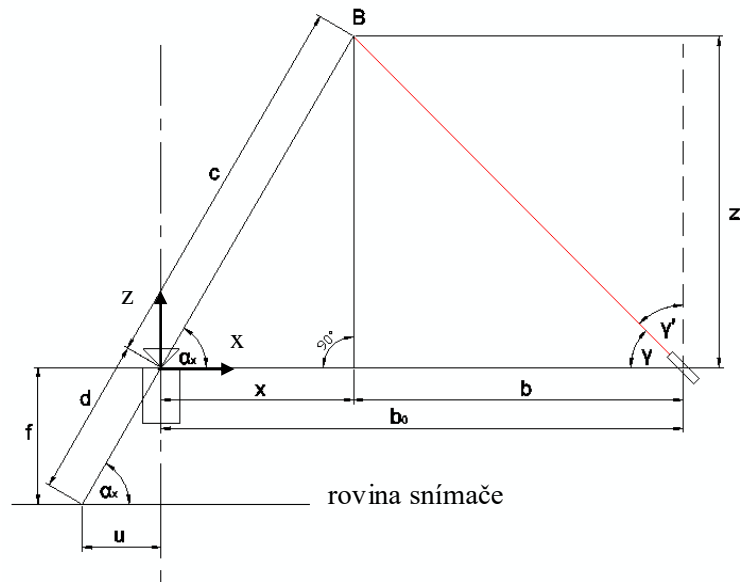
Důležité je také zobrazení snímků po odstranění zkreslení a ověření, zda linie, které mají být přímé, opravdu přímé jsou. Pokud je třeba analýzu více zpřesnit, je třeba odebrat snímky s vysokou reprojekční chybou, případně přidat další snímky. Pokud analýza splnila očekávání, mohou se exportovat parametry, nebo vygenerovat skript. [14]

5.3 Aplikace triangulační metody na dírkový model

Aplikace triangulační metody na dírkový model znamená, že triangulační trojúhelník je tvořen dírkovou kamerou, laserem a neznámým bodem v prostoru. Pomocí triangulace je možné vypočítat vzdálenost neznámého bodu od kamery, a díky tomu (za použití dalších goniometrických nebo trigonometrických funkcí) vypočítat souřadnice tohoto bodu. [13, 17]

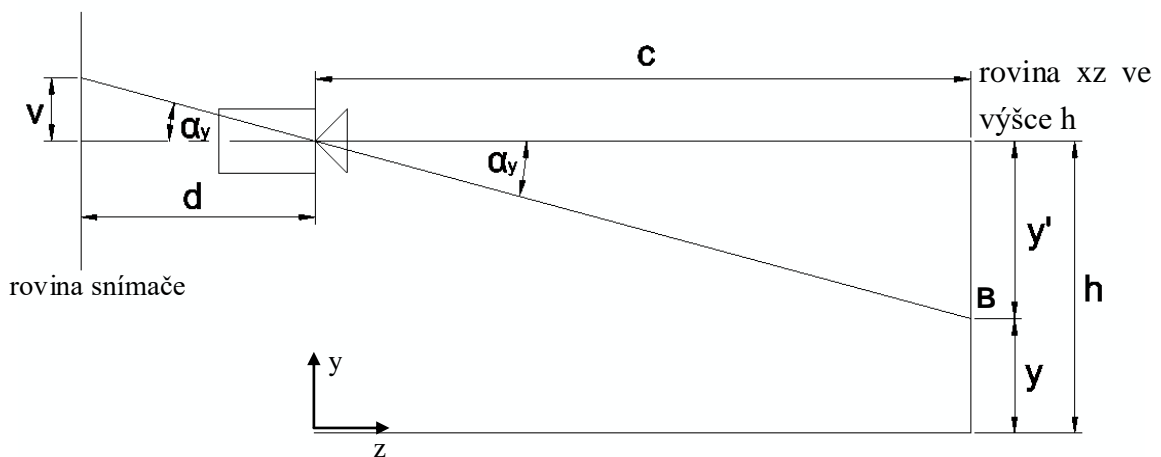
Na obr. 18 je zobrazen pohled shora na triangulační trojúhelník. Z důvodu umístění kamery do výšky h , je tento trojúhelník zobrazen v rovině xz , která je vzdálena od středu souřadného systému o výšku h . Souřadnice počátku byly určeny podle bodu, kterým je popsána štěrbina kamery (viz kapitola 5.1.1) a umístěny do výšky $h=0$. Hodnoty souřadnic středu vyobrazené roviny xz jsou tedy v prostoru $[0, h, 0]$. Na obr. 18 je vidět, že bod B je neznámý bod v prostoru, b_0 je vzdálenost mezi kamerou a liniovým laserem (tzv. báze), γ' je úhel natočení liniového laseru, γ je úhel mezi bází a rovinou laseru, f je ohnisková vzdálenost kamery, u je vzdálenost mezi místem dopadu světelného paprsku na senzor a osou z v rovině xz , d je vzdálenost mezi místem dopadu světelného paprsku a počátkem souřadného systému v rovině xz , c je vzdálenost mezi počátkem a bodem B v rovině xz , α_x je úhel mezi projekcí světelného paprsku do roviny xz a rovinou senzoru, nebo bází, x a z jsou souřadnice bodu B a b je vzdálenost mezi

liniovým laserem a bodem B v rovině xy. Rovina laseru je kolmá k rovině xz a rovina senzoru je rovnoběžná s rovinou xy a kolmá k rovinám xz a yz. [11, 13, 17]



Obr. 18: Pohled shora na triangulační trojúhelník (rovinu xz)

Na obr. 19 je vidět pohled z boku (rovinu yz), kde počátek souřadného systému leží pod štěrbinou kamery ve výšce $h=0$, α_y je úhel mezi světelným paprskem a průmětem paprsku do roviny xz, v je vzdálenost mezi místem dopadu světelného paprsku na senzor a osou z v rovině yz, h je výška, ve které je kamera umístěna, y' je vzdálenost bodu B od roviny xz a y je souřadnice výšky bodu B. Protože světelný paprsek není rovnoběžný s rovinou yz, jsou vzdálenosti c a d a úhel α_y v tomto pohledu zobrazeny zkresleně a slouží pouze pro představu situace. [11, 13, 17]



Obr. 19: Pohled z boku na triangulační trojúhelník (rovinu yz)

5.3.1 Sestavení rovnic

K výpočtu souřadnic je nutné znát velikost pixelů. Velikost pixelu se vypočítá z vnitřních parametrů kamery obsažených v matici K . K výpočtu velikosti pixelů se použijí parametry f_x a f_y , které vypočítá aplikace ke kalibraci kamery, vztahy (1) a (2) a znalost ohniskové vzdálenosti F . Ze vztahu (1) můžeme vyjádřit parametr s_x :

$$s_x = \frac{f_x}{F}$$

Parametr udává počet pixelů na 1 mm ve směru osy x snímače. Když tuto hodnotu převrátíme, dostaneme šířku pixelu v milimetrech.

$$(3)s_x^{-1} = \frac{F}{f_x}$$

Stejným způsobem zjistíme výšku pixelu z parametru f_y a vztahu (2):

$$(4)s_y^{-1} = \frac{F}{f_y}$$

Velikost u a v je potom dána jako:

$$(5)u = p_{pix} \cdot s_x^{-1}$$

$$(6)v = p_{pix} \cdot s_y^{-1}$$

kde p_{pix} je počet pixelů. Ze znalostí goniometrických funkcí je možné odvodit rovnice pro výpočet úhlu γ a α_x , vzdálenosti b a souřadnic x a z , z triangulačního trojúhelníku při pohledu shora. Pokud je úhel γ' v radiánech, platí rovnice (7):

$$(7)\gamma = \frac{\pi}{2} - \gamma'$$

Odvození rovnice (8) pro výpočet úhlu α_x :

$$tg(\alpha_x) = \frac{F}{u}$$

$$(8)\alpha_x = arctg\left(\frac{F}{u}\right)$$

Odvození rovnic (9) a (10) pro výpočet souřadnice z :

$$tg(\gamma) = \frac{z}{b}$$

$$(9)z = tg(\gamma) \cdot b$$

$$tg(\alpha_x) = \frac{z}{x}$$

$$(10)z = tg(\alpha_x) \cdot x$$

Porovnáním rovnic (9) a (10) a následnou úpravou dostaneme rovnici (11):

$$tg(\gamma) \cdot b = tg(\alpha_x) \cdot x$$

$$(11)x = \frac{tg(\gamma) \cdot b}{tg(\alpha_x)}$$

Souřadnici x lze vypočítat i podle rovnice (12):

$$(12)x = b_0 - b$$

Porovnáním rovnic (11) a (12) a jejich úpravou dostaneme rovnici (13):

$$b_0 - b = \frac{\operatorname{tg}(\gamma) \cdot b}{\operatorname{tg}(\alpha_x)}$$

$$b_0 = \frac{\operatorname{tg}(\gamma) \cdot b}{\operatorname{tg}(\alpha_x)} + b$$

$$b_0 = b \cdot \left(\frac{\operatorname{tg}(\gamma)}{\operatorname{tg}(\alpha_x)} + 1 \right)$$

$$(13) b = \frac{b_0}{\frac{\operatorname{tg}(\gamma)}{\operatorname{tg}(\alpha_x)} + 1}$$

Ze znalosti Pythagorovy věty odvodíme velikost vzdálenosti c a d.

$$(14) c = \sqrt{x^2 + z^2}$$

$$(15) d = \sqrt{f^2 + u^2}$$

Při pohledu z boku na triangulační trojúhelník odvodíme souřadnici y:

$$(16) \operatorname{tg}(\alpha_y) = \frac{v}{d}$$

$$(17) \alpha_y = \operatorname{arctg} \left(\frac{v}{d} \right)$$

$$(18) \operatorname{tg}(\alpha_y) = \frac{y'}{c}$$

Porovnáním rovnic (16) a (18) dostaneme:

$$\frac{v}{d} = \frac{y'}{c}$$

$$(19) y' = \frac{v \cdot c}{d}$$

$$(20) y = h - y'$$

[11, 13, 14, 17]

6 PROGRAM PRO SNÍMÁNÍ A ANALÝZU DAT

V programu Matlab byl napsán program pro snímání a analýzu dat z webkamery. Na úvod programu byly použity funkce `clear all` a `clc`, jejichž funkce je popsána poznámkou vloženou do programu. Poznámka je část programu, který při běhu programu není vykonáván. V programovacím prostředí Matlab je komentář označen zelenou barvou a k vytvoření komentáře slouží symbol procenta. Téměř každý příkaz bude takto okomentován.

```
%smazání všech uložených proměnných  
clear all  
%smazání historie příkazového okna  
clc
```

6.1 Vytvoření objektu Arduina

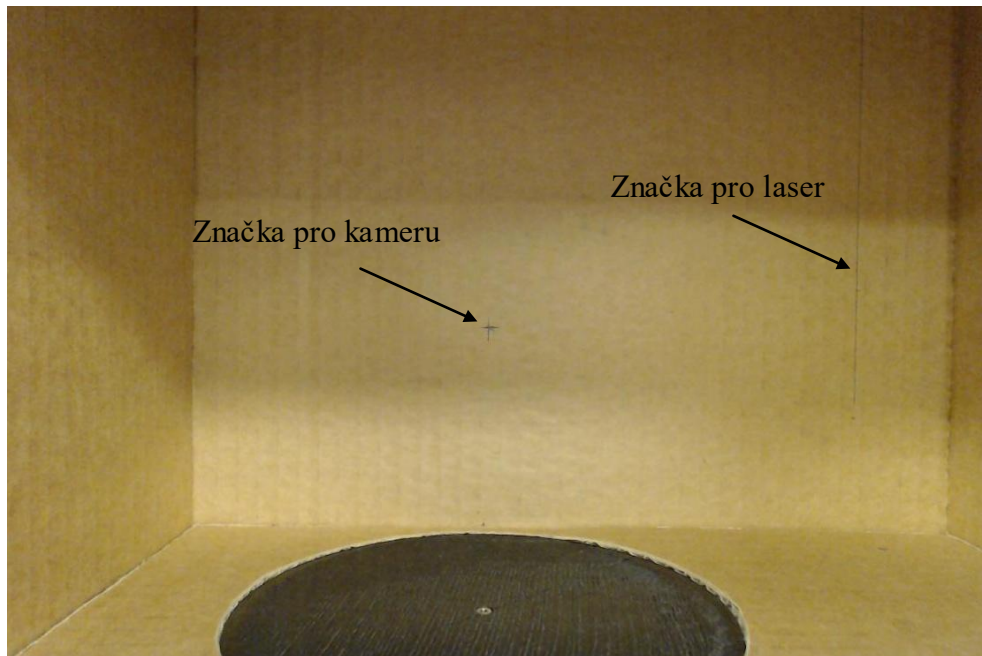
Aby bylo možné používat periferní zařízení připojené k desce Arduino, musí se připojit Arduino k počítači pomocí USB rozhraní. Spojení mezi programem Matlab a Arduinem se vytvoří, jestliže vytvoříme např. objekt `A`, který definujeme jako Arduino, které je připojené k portu `com4` a typ desky je `uno`.

```
%vytvoření objektu (port, typ desky)  
A=arduino('com4', 'uno');
```

6.2 Inicializace laseru

Pro kontrolu nastavení liniového laseru byla vytvořena následující část programu, kdy se zapne laser a `enable pin` se nastaví na logickou 0. Pokud laser nesvítí na protilehlou značku (obr. 20), je třeba laser uvolnit a pootočit s ním do výchozí pozice, případně (pokud je laser nerovnoběžný se značkou) upravit rovnoběžnost laseru a znovu laser připevnit. Po připevnění se stiskne ENTER pro pokračování programu.

```
%zapnutí laseru  
writeDigitalPin(A, 'A4', 1);  
%nastavení ENABLE pin (zapíná a vypíná výstup)  
writeDigitalPin(A, 'D8', 0);  
%zobrazení hlášky + čekání na pokračování  
input('Nastav liniový laser do výchozí pozice a stiskni Enter');  
%vypnutí laseru  
writeDigitalPin(A, 'A4', 0);
```

Obr. 20: Inicializační značka laseru a kamery

6.3 Definice rozměrů a konstant

Aby program mohl počítat s rozměry a konstantami, je třeba tyto údaje uložit do proměnných. Koefficienty K, K1, K2 a K3 jsou konstanty, které určují počet opakování příslušných cyklů. Proměnné i, i1 a fi jsou pouze inicializovány, protože při výpočtu volají samy sebe, a je třeba znát nějakou výchozí hodnotu. Hodnoty u_str a v_str jsou určeny podle rozlišení snímků a určují středovou souřadnici. Při rozlišení 2304x1536 má středová souřadnice hodnoty [šířka/2, výška/2], což je v našem případě [1152, 768].

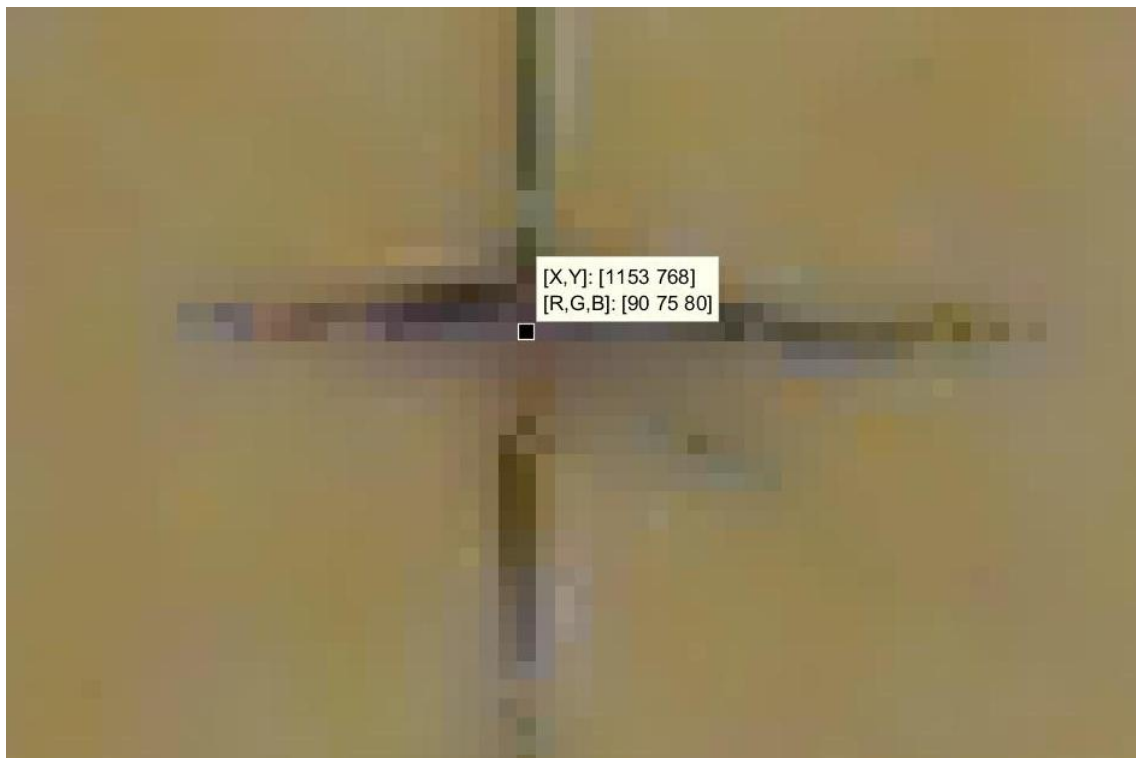
```
%nedefinování rozměrů a konstant
F=3.67; %ohnisková vzdálenost (mm)
b0=140; %délka báze (mm)
h=79; %vzdálenost kamery ode dna (mm)
z0=207 %vzdálenost kamery od středu otáčení (mm)
K=38; %počet kroků do výchozí pozice (z pozice 0°, když 400 kroků = 1 otáčka)
K1=200; %počet pootočení platformy (když 3200 kroků = 1 otočka o 360°)
K2=0; %počet pootočení laseru ke zlepšení přesnosti
K3=16;
i=0; %inicializace počítadla i
i1=0; %inicializace počítadla i1
u_str=1152; %číslo prostředního řádku snímku
v_str=768; %číslo prostředního sloupce snímku
r1=67; %poloměr platformy
fi=0; %úhel natočení platformy na začátku
```

6.4 Nastavení kamery

Podobně jako u Arduina, je třeba vytvořit spojení s webkamerou. Dále je nutné nastavit požadované rozlišení a zaostření. Zaostření je třeba nastavit vyzkoušením různých hodnot pro různé snímky a vybrat hodnotu, při které je objekt nejlépe zaostřený.

Souřadnice středu by měla být na středu značky, která je zakreslena na protilehlé straně skeneru, viz obr. 21. Dále je nutné nahrát uložené proměnné z kalibrace kamery. Z těchto proměnných si načteme koeficienty f_x a f_y , ze kterých vypočítáme šířku a výšku pixelů pomocí rovnic (3) a (4).

```
% nadeřinování kamery
cam = webcam('Logitech HD Pro Webcam C920'); %vytvoření objektu kamery
cam.Resolution = '2304x1536'; %nastavení rozlišení kamery na 2304x1536
cam.Focus = 25; %nastavení zaostřeni
load ('cameraParams'); %načtení parametrů kamery (kalibrace)
fx=cameraParams.IntrinsicMatrix(1,1);
fy=cameraParams.IntrinsicMatrix(2,2);
x_pix=F/fx; %šířka pixelu (mm)
y_pix=F/fy; %výška pixelu (mm)
```



Obr. 21: Přibližná kalibrační značka

6.5 Nastavení laseru do výchozí pozice

Pro otáčení krokovým motorem, ke kterému je připevněn laser, je třeba posílat signály logické 1 a 0 na pin D2. Výchozí nastavení motorů je, že pokud přijde signál, tak se motor pootočí o 1 krok po směru hodinových ručiček. Pokud chceme změnit směr otáčení, musíme poslat logickou 1 na digitální pin D5. Pokud chceme udělat více kroků, musíme opakovat posílání signálů na pin D2. K tomuto účelu slouží cyklus for, který se provede K krát. Protože je K nastaveno na 38 a velikost kroku je $0,9^\circ$, provede se 38 kroků po $0,9^\circ$, což dává úhel natočení $34,2^\circ$.

```
%nastavení laseru do výchozí pozice
```

```
writeDigitalPin(A, 'D5' , 1); %nastavení směru otáčení laseru
for n=1:K
    writeDigitalPin(A, 'D2' , 1); %pootočení laseru
    writeDigitalPin(A, 'D2' , 0); %pootočení laseru
end
```

6.6 Vytvoření snímku

První cyklus for nám udává, kolikrát se otáčecí stůl pootočí. Druhý cyklus for se provede vícekrát jen v případě, že chceme zlepšit přesnost skeneru otáčením laseru. Proměnná K2+1 udává, s kolika různými polohami laseru bude počítáno. Příkazem I=snapshot(mycam) se do proměnné I uloží snímek vyfocený kamerou, která je nadefinována v proměnné cam. Fotka se uloží ve formátu RGB, což znamená, že každý pixel obrázku je tříhodnotový a udává intenzitu červeného, zeleného a modrého světla v hodnotách od 0 do 255. Díky tomu je možné snímek rozložit na jednotlivé složky RGB spektra. Dále je pomocí příkazů undistortImage odstraněno ze snímků zkreslení způsobené vadami optických čoček. Snímek I, který není ozářen laserem, byl použit pro určování barvy jednotlivých vypočtených bodů. Dalším krokem bylo vyfocení objektu ozářeného laserem, a opět rozložení snímku na jednotlivé složky spektra. Protože komunikace s Arduinem má určité zpoždění, bylo třeba mezi snímky udělat pauzu, aby se stihnul zapnout laser před vyfocením scény. Po vyfocení snímku byl laser opět vypnut.

```
for n=1:K1
    for nl=1:(K2+1)
        I=snapshot(cam); %udělat foto
        I=undistortImage(I, cameraParams); %odstranit zakřivení
        redChannel = I(:, :, 1);
        greenChannel = I(:, :, 2);
        blueChannel = I(:, :, 3);
        writeDigitalPin(A, 'A4' , 1); %zapnutí laseru
        pause(2);
        J=snapshot(cam); %udělat foto
        J=undistortImage(J, cameraParams); %odstranit zakřivení
        redChannel2 = J(:, :, 1);
        greenChannel2 = J(:, :, 2);
        blueChannel2 = J(:, :, 3);
        writeDigitalPin(A, 'A4' , 0); %vypnutí laseru
    end
end
```

6.7 Segmentace barev a výpočet souřadnic bodů

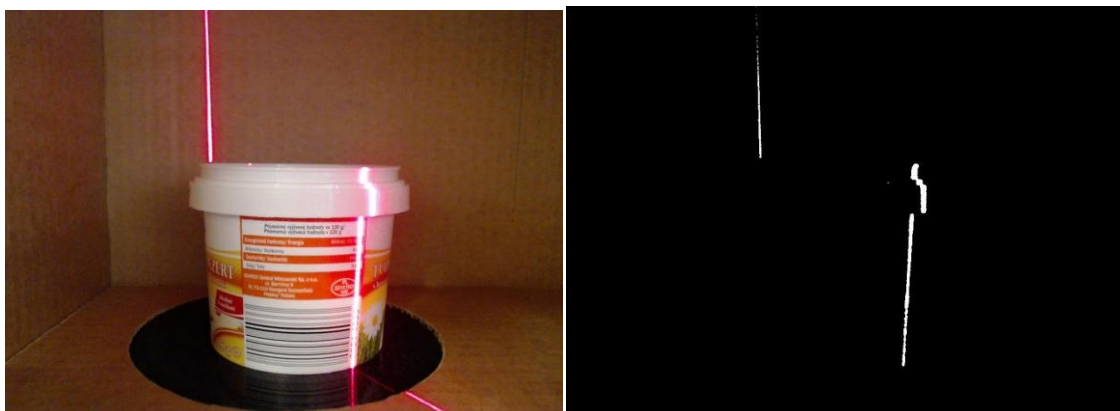
6.7.1 Segmentace barev

Segmentace barev je způsob digitálního zpracování obrazu založená na základě metody prahování. Metoda prahování je založená na porovnávání intenzity jednotlivých barev všech pixelů. Ve spektru RGB jde o porovnávání intenzity červené, zelené a modré barvy. V tomto případě se využívá této metody k detekci červeného laseru. Aby se mohly nastavit jednotlivé hranice intenzit, musí se vytvořit testovací snímek (obr. 22),

podle kterého se zjistí, jaké hodnoty mají body ozářené laserem a body v pozadí. Podle těchto hodnot se určí hladiny jednotlivých barev. Aby byly otestovány všechny pixely, byly vytvořeny 2 cykly for, kde byl počet opakování určen počtem řádků a sloupců. Pomocí těchto cyklů byly zkontrolovány všechny pixely. Pro kontrolu je možné zobrazit funkčnost nastavených hladin (obr. 23). [18, 19, 20]



Obr. 22: Testovací snímek



Obr. 23: Snímek před segmentací (vlevo) a po segmentaci (vpravo)

6.7.2 Výpočet souřadnic bodů

Výpočet souřadnic probíhá podle odvozených rovnic z kapitoly 5.3.1. Nejdříve probíhá výpočet vzdálenosti u podle rovnice (5), kde $k_s - u_{str}$ je počet pixelů mezi středem souřadného systému v rovině xz a dopadajícím paprskem. Pokud je počet pixelů záporný, znamená to, že paprsek dopadá na zápornou část osy x . Výpočet vzdálenosti v je analogický k výpočtu vzdálenosti u a počítá se podle rovnice (6). Gama je počítána podle rovnice (7), kde γ' je určena podle úhlu natočení laseru od výchozí pozice (při nastavení na vyšší přesnost) a úhlem natočení ve výchozí pozici. Alfax je počítána podle vzorce (8), b podle (13), x podle (12), z podle (9), d podle (15), α_f podle (17), c podle (14) a y_1 podle (19). Protože software kamery přepočítává souřadnice pixelů, aby obraz nebyl převrácený, jsou převrácené i souřadnice pixelů osy y . Proto musíme vzorec (20) upravit na:

$$y = h - (-y')$$

$$(21)y = h + y'$$

Pro přepočet souřadnic o úhel natočení, byly převedeny kartézské souřadnice do válcových, k válcovým souřadnicím byl připočten úhel natočení, a poté byly opět převedeny na kartézské. Nerovnice $r \leq r_1$ udává, že vzdálenost bodu od středu otáčení musí být menší nebo rovna poloměru otáčecího stolu. To znamená, že s body zachycenými mimo otáčecí stůl nebude počítáno, takže nebudou způsobovat nechtěný šum na modelu. Nakonec jsou všechny detekované souřadnice uloženy jako jednotlivé řádky matic $X1$, $Y1$ a $Z1$. Do řádků matice $C1$ jsou uloženy barvy jednotlivých bodů.

```

for k_r = 1:1536
    for k_s = 1:2304
        %detekce laseru
        if (redChannel2(k_r,k_s) > 250 && greenChannel2(k_r,k_s) > 240 && blueChannel2
(k_r,k_s) > 240)
            u=(k_s-u_str)*x_pix;
            v=(v_str-k_r)*y_pix;
            gama=pi/2-(n1-1)*0.9*pi/180-K*0.9*pi/180;
            alfax=atan(F/u);
            b=b0/(tan(gama)/tan(alfax)+1);
            x=b0-b;
            z=tan(gama)*b;
            d=sqrt(F^2+u^2);
            alfay=atan(v/d);
            c=sqrt(x^2+z^2);
            y1=(v*c)/d;
            y=h+y1;
            z=z-z0;
            r=sqrt(x^2+z^2);
            if(r<=r1)
                fil=atan2(z,x);
                x=r*cos(fil+fi);
                z=r*sin(fil+fi);
                i=i+1;
                X1(i,:)=x;
                Y1(i,:)=y;
                Z1(i,:)=z;
                C1(i,:)=[redChannel(k_r,k_s) ;greenChannel(k_r,k_s) ; blueChannel(k_r,k_s)];
            end
        end
    end
end

```

6.8 Kompenzace šířky světelného pruhu

Aby model nebyl vytvořen z malých ploch, které by se mohly překrývat, bylo třeba z těchto ploch udělat linie. Aby nevznikla chyba v programu, bylo nutné nejprve definovat, za jakých podmínek se vykoná následující část programu. Podmínka $i \sim 0$ znamená, že pokud v předchozí části programu nebyla definována žádná souřadnice, je tato část programu se přeskočena. Příkaz `size(X1)` uloží do proměnných `s1` a `s2` počet řádků a sloupců matice `X1`. Protože detekce probíhá po řádcích snímku, matice `X1` reprezentuje šířku světelného pruhu. Proto byla jako reprezentativní vzorek vybrána prostřední souřadnici z každé matice a uložena do nových matic `X`, `Y`, `Z` nebo `C`, jež budou shromažďovat všechny takto získané souřadnice. Nakonec byly matice `X1`, `Y1`, `Z1` a `C1` smazány, aby bylo možné do nich opět ukládat souřadnice z dalších řádků snímku. Proměnné `i` byla opět přiřazena hodnota 0.

```
if (i~=0)
    i1=i1+1;
    [s1,s2]=size(X1);
    s1=round(s1/2);
    x1=X1(s1,1);
    y1=Y1(s1,1);
    z1=Z1(s1,1);
    c1=C1(s1,:);
    X(i1,:)=x1;
    Y(i1,:)=y1;
    Z(i1,:)=z1;
    C(i1,:)=c1;
    clear X1 Y1 Z1 C1
    i=0;
end
end
```

6.9 Zvýšení přesnosti skenování otáčením laseru

Pokud je třeba zvýšit počet souřadnic, ze kterých se bude model skládat, může se otáčet s laserem, a tím měnit úhel natočení laseru. S novými údaji můžeme při jednom pootočení stolu vypočítat souřadnice z dvou různých laserových rovin. Z první podmínky `if` vyplývá, že pokud cyklus neběží poprvé a zároveň je nastavena zvýšená přesnost, provede se otáčení laseru. Druhá podmínka udává, že pokud cyklus neběží poprvé, ale běžel už tolikrát, kolik byla zvolena úroveň přesnosti, natočí se laser zpět do výchozí pozice.

```
if (n1~=K2+1 && K2~=0)
    writeDigitalPin(A, 'D2', 1); %pootočení laseru
    writeDigitalPin(A, 'D2', 0); %pootočení laseru
    pause(2);
end
if (n1==K2+1 && K2~=0)
    writeDigitalPin(A, 'D5', 0); %nastavení směru otáčení laseru
    for n2=1:K2
        writeDigitalPin(A, 'D2', 1); %pootočení laseru
        writeDigitalPin(A, 'D2', 0); %pootočení laseru
    end
    writeDigitalPin(A, 'D5', 1); %nastavení směru otáčení laseru
end
end
```

6.10 Otáčení se stolem

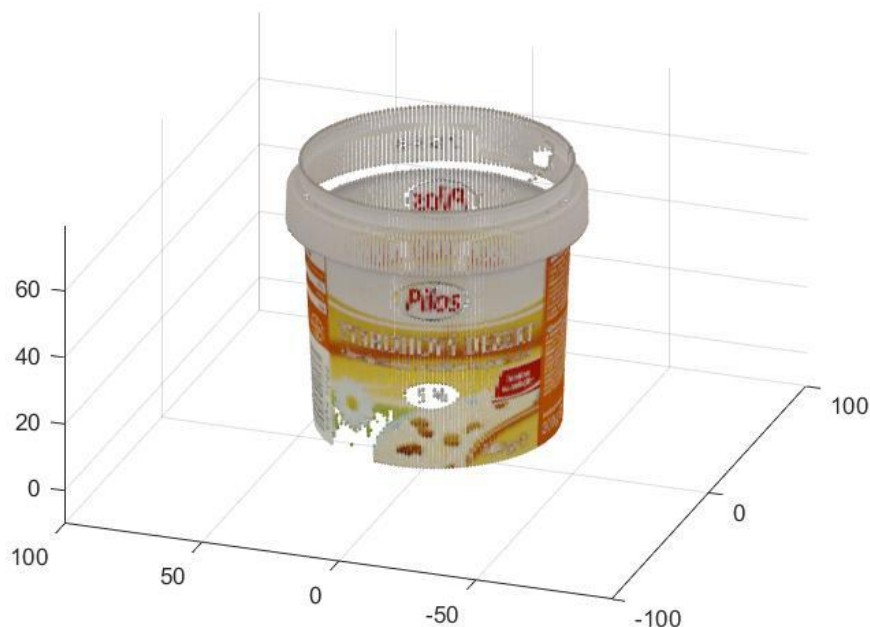
Po dokončení výpočtů souřadnic se pootočí stůl a vypočítá se úhel natočení ϕ .

```
for n3=1:K3
    writeDigitalPin(A, 'D3' , 1); %pootočení platformy ?=pin 1=ZAP
    writeDigitalPin(A, 'D3' , 0); %pootočení platformy ?=pin 0=VYP
end
fi=fi+(K3*0.1125*pi/180); %úhel natočení (v rad)
pause(2);
end
```

6.11 Zobrazení 3D modelu

Protože funkce `scatter3` potřebuje mít zadané údaje o barvě v hodnotách od 0 do 1, je třeba hodnoty matice `C` převést z rozsahu 0 až 255 na hodnoty 0 až 1. Tento převod je realizován převedením hodnot matice `C` do formátu `double` a následným dělením každého prvku této matice hodnotou 255. Na obr. 24 můžeme vidět 3D model vykreslený pomocí funkce `scatter3` bez zvyšování přesnosti otáčením laseru.

```
writeDigitalPin(A, 'D8' , 1); %nastavení logické 1 na ENABLE pin
C=double(C);
C=C/255;
scatter3(X,Y,Z,1,C);
```



Obr. 24: 3D model objektu se základní přesností skenování

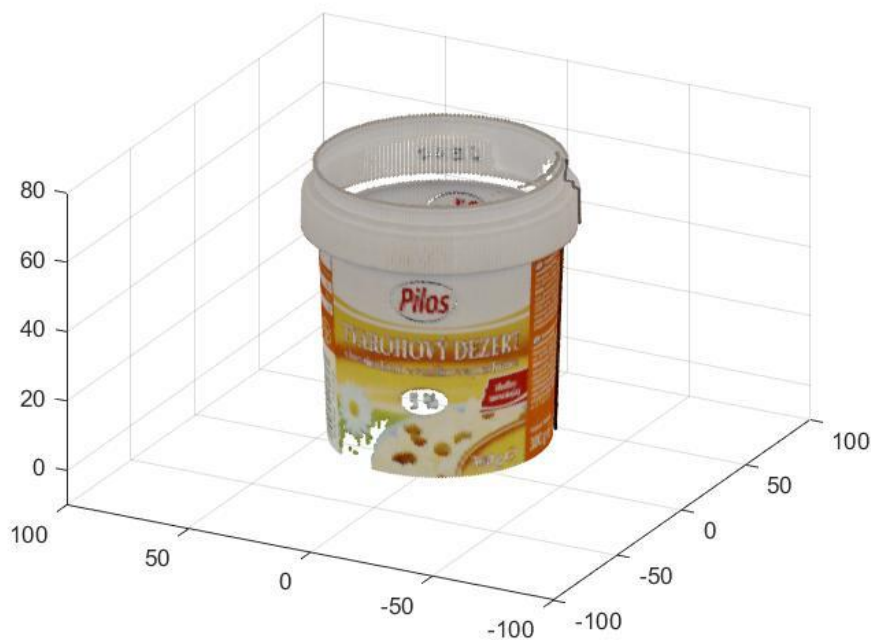
7 ZHODNOCENÍ A DISKUZE

Faktor, který nejvíce ovlivňuje kvalitu výsledného modelu je intenzita dopadajícího světla na skenovaný objekt. Při příliš velké intenzitě světla může docházet k detekci bodů, na které nedopadá laserový paprsek a může tak vznikat rušivý šum. V budoucnu by mohl být tento problém vyřešen úplným uzavřením skenovacího prostoru a instalací umělého osvětlení (např. led diod), se kterým by bylo možné regulovat jas.

Pokud bude kladen důraz na přesnost 3D modelu, je třeba počítat s dlouhou dobou skenování, která při vyšší přesnosti skenování trvá až 45 minut. Doba skenování při základní přesnosti trvá asi 22 minut. Na obr. 25 je zobrazen 3D model s vyšší přesností skenování, který je možný porovnat s modelem s nižší přesností, vyobrazeném na obr. 24. Přesnost skenování je také částečně závislá na použitém materiálu, ze kterého je vytvořena konstrukce. Stávající konstrukce je z vlnité lepenky, která nemá ideální pevnost, takže vytvoření konstrukce s pravými úhly není snadné. Použitý způsob připevnění kamery ke konstrukci také není ideální, protože i sebemenší kontakt s kamerou může způsobit její vychýlení.

Dále je skener omezen na skenování menších objektů. Kvůli možnosti skenování vyšších objektů nepokrývá zorné pole kamery celý otáčecí stůl. Tato omezení by bylo možné v budoucnu řešit kamerou, která by byla upevněna do pojezdu, s nímž by bylo možné měnit výšku, ve které je kamera umístěna.

Při pořízení všech součástí na českém trhu by cena skeneru byla přibližně 4 700 Kč (v uvedené ceně nejsou zahrnuty součásti vytisknuté na 3D tiskárně).



Obr. 25: 3D model objektu s vyšší přesností skenování

8 ZÁVĚR

V této práci bylo pojednáno o metodách skenování a jejich dalším dělení. Dále zde byly uvedeny informace o 3D skenerech zveřejněných na internetu, které jsou svou konstrukcí podobné použitému 3D skeneru. Dále byly popsány nejdůležitější části 3D skeneru, mezi které patří liniový laser, Arduino UNO s rozšiřující deskou CNC shield, otáčecí stůl, webkamera a krokové motory, které otáčí s liniovým laserem a otáčecím stolem.

Dále byla popsána metoda triangulace, která byla aplikována na dírkový model. Pomocí této metody byly sestaveny rovnice pro výpočet neznámého bodu v prostoru. Při použití této metody bylo nutné provést kalibraci kamery, pomocí které byly kompenzovány především vady optických čoček. Kalibrace kamery byla provedena pomocí aplikace v programovacím prostředí Matlab.

V poslední části práce jsou popisány jednotlivé kroky programu pro snímání a analýzu dat z kamery. Dále je popsáno nastavení laseru a kamery, tvorba snímků a jejich následná úprava. Program využívá metodu segmentace barev, která slouží k detekci laseru. V další části programu jsou prováděny výpočty souřadnic bodů z odvozených rovnic, které jsou následně zobrazeny jako 3D model skenovaného objektu.

V závěru práce je zhodnocena problematika detekce bodů při vysoké intenzitě dopadajícího světla, přesnost 3D modelu v závislosti na čase skenování a kvalita použitých materiálů. Dále jsou zhodnoceny možnosti skenování vzhledem k výšce skenovaného objektu a je uvedena orientační cena použitého skeneru.

9 SEZNAM POUŽITÉ LITERATURY

- [1] ČERMÁK, Jan. *Metody 3D skenování objektů* [online]. Vysoké učení technické v Brně, 2015 [cit. 2017-05-22]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=103850
- [2] NAVRÁTIL, Robert. *Technologie skenování ve 3D* [online]. [cit. 2017-05-22]. Dostupné z: <http://www.spsks.cz/wp-content/uploads/2016/03/Technologie-skenov%C3%A1n%C3%AD-ve-3D-0.pdf>
- [3] LUCAS, Mario. *FabScan Pi - an open-hardware stand-alone web-enabled 3D scanner* [online]. RWTH Aachen University, 2015 [cit. 2017-05-22]. Dostupné z: <http://hci.rwth-aachen.de/materials/publications/lukas2015a.pdf>
- [4] FabScan Cube. In: *Fabscan cube* [online]. HANION [cit. 2017-05-23]. Dostupné z: <http://www.filaments4u.com/en/fabscan/94-fabscan-cube.html>
- [5] Matter and Form 3D Scanner. In: *MFSIVIR B&H Photo Video* [online]. New York: B & H Foto & Electronics, 2017 [cit. 2017-05-23]. Dostupné z: https://www.bhphotovideo.com/c/product/1214191-REG/matter_and_form_mfs1v1r_3d_scanner.html
- [6] Line laser. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2014 [cit. 2017-05-22]. Dostupné z: https://en.wikipedia.org/wiki/Line_laser
- [7] *Arduino/Genuino UNO* [online]. 2017 [cit. 2017-05-22]. Dostupné z: <https://www.arduino.cc/en/main/arduinoBoardUno>
- [8] Krokový motor. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2016 [cit. 2017-05-23]. Dostupné z: https://cs.wikipedia.org/wiki/Krokov%C3%BD_motor
- [9] DIKMEN, Serkan. *Development of Star Tracker Attitude and Position Determination System for Spacecraft Maneuvering and Docking Facility* [online]. Luleå University of Technology, 2016 [cit. 2017-05-22]. Dostupné z: <http://ltu.diva-portal.org/smash/get/diva2:1044866/FULLTEXT01.pdf>
- [10] Triangulace. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2016 [cit. 2017-05-23]. Dostupné z: <https://cs.wikipedia.org/wiki/Triangulace>
- [11] LANMAN, Douglas a Gabriel TAUBIN. *Build Your Own 3D Scanner: 3D Photography for Beginners* [online]. Brown University, 2009 [cit. 2017-05-23]. Dostupné z: <http://mesh.brown.edu/byo3d/notes/byo3D.pdf>
- [12] Pinhole camera. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2017 [cit. 2017-05-23]. Dostupné z: https://en.wikipedia.org/wiki/Pinhole_camera
- [13] Pinhole camera model. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2017 [cit. 2017-05-23]. Dostupné z: https://en.wikipedia.org/wiki/Pinhole_camera_model
- [14] Single Camera Calibration App. *MATLAB & Simulink* [online]. Natick: The MathWorks, 2017 [cit. 2017-05-23]. Dostupné z: <https://www.mathworks.com/help/vision/ug/single-camera-calibrator-app.html>
- [15] Čočka (optika). In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2017 [cit. 2017-05-23]. Dostupné z: [https://cs.wikipedia.org/wiki/Čočka_\(optika\)](https://cs.wikipedia.org/wiki/Čočka_(optika))

- [16] [pattern]. In: *Pattern.png* [online]. opencv, 2016 [cit. 2017-05-23]. Dostupné z: <http://docs.opencv.org/2.4/downloads/pattern.png>
- [17] Laser scanning using a laser pointing device and a camera. *Laser scanning and Triangulation* [online]. CETI/ATHENA: Pavlidis, 2006 [cit. 2017-05-23]. Dostupné z: <http://georgepavlides.info/research/LaserScanningAndTriangulation.php>
- [18] CHMELAR, Pavel a Martin DOBROVOLNY. *The Laser Line Detection for Autonomous Mapping Based On Color Segmentation* [online]. University of Pardubice, 2013 [cit. 2017-05-23]. Dostupné z: <http://waset.org/publications/9996622/the-laser-line-detection-for-autonomous-mapping-based-on-color-segmentation>
- [19] SANCHEZ, David. How to set up an RGB treshold. In: *MATLAB Answers - MATLAB Central* [online]. 2013 [cit. 2017-05-23]. Dostupné z: <https://www.mathworks.com/matlabcentral/answers/104920-how-can-i-set-up-an-rgb-threshold>
- [20] Segmentace obrazu. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2013 [cit. 2017-05-23]. Dostupné z: https://cs.wikipedia.org/wiki/Segmentace_obrazu
- [21] Detect checkerboard pattern in image. In: *MATLAB detectCheckerboardPoints* [online]. Natick: The MathWorks, 2017 [cit. 2017-05-23]. Dostupné z: <https://www.mathworks.com/help/vision/ref/detectcheckerboardpoints.html>