



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

## 3D SKENER SE STRUKTUROVANÝM OSVĚTLENÍM

STRUCTURED-LIGHT 3D SCANNER

### BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

### AUTOR PRÁCE

AUTHOR

Jaroslav Hadzima

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Aleš Jelínek

BRNO 2017

# Bakalářská práce

bakalářský studijní obor **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

**Student:** Jaroslav Hadzima

**ID:** 173648

**Ročník:** 3

**Akademický rok:** 2016/17

**NÁZEV TÉMATU:**

## **3D skener se strukturovaným osvětlením**

**POKyny PRO VYPRACOVÁNÍ:**

1. Na základě semestrálního projektu vyhodnoťte klíčové vlastnosti systému pro skenování se strukturovaným osvětlením a upravte ho pro nasazení na mobilním robotu.
2. Otestujte skenovací systém na široké škále objektů za různých světelných podmínek.
3. Vyhodnoťte experimenty, identifikujte slabá místa této metody a popište je v dokumentaci.
4. Na základě experimentů dále odvoďte přesnost měření v daném rozsahu podmínek a proberte metody kalibrace systému.
5. Navrhněte komunikační protokol pro ovládání vašeho 3D skeneru, sepište dokumentaci projektu a prezentujte dosažené výsledky.

**DOPORUČENÁ LITERATURA:**

HLAVÁČEK, Jiří. 3D skener na bázi strukturovaného světla. Plzeň, 2013. Bakalářská práce. Západočeská univerzita v Plzni.

**Termín zadání:** 6.2.2017

**Termín odevzdání:** 29.5.2017

**Vedoucí práce:** Ing. Aleš Jelínek

**Konzultant:**

**doc. Ing. Václav Jirsík, CSc.**  
*předseda oborové rady*

**UPOZORNĚNÍ:**

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **ABSTRAKT**

3D skener je zariadenie, ktoré je schopné analýzou najčastejšie obrazových dát vytvoriť model skenovaného objektu. Rozdelenie 3D skenerov sa delí na kontaktné a bezkontaktné. Táto práca je zameraná na problematiku 3D skenovania s využitím bezkontaktných optických skenerov, konkrétne využívajúcich štruktúrovaného osvetlenia vo forme vzorov v Grayovom kóde. Postupne vyhodnocujem jednotlivé hardvérové a softvérové úskalí v priebehu skenovania, od kalibrácie až po výsledné mračno bodov. V práci sú uvedené rozborov testov, skúmajúcich kvalitu, presnosť a limitácie skenovania touto metódou. Celý systém je pri tom aplikovateľný na mobilnom robotovi a zároveň plne vzdialene ovládateľný. Väčšina výpočtov sa odohráva na Raspberry Pi 3 model B a samotná vizualizácia prebieha na počítači.

## **KLÚČOVÉ SLOVÁ**

3D skenovanie, štruktúrované svetlo, vzor štruktúrovaného svetla v Grayovom kóde, kalibrácia, vizualizácia, rekonštrukcia, mesh, mračno bodov, Raspberry Pi 3 model B, projektor

## **ABSTRACT**

3D scanner is a device capable of creating a model using analysis, most commonly from image data. 3D scanners could be divided into two major groups, contact and noncontact scanners. Aim of this study is to interpret problematics of 3D scanning with structured light. Specifically usage of Gray code patterns. There is step by step evaluation of individual hardware and software obstacles and possible errors, from calibration to final point cloud. In next section, there are included tests determining quality, precision and accuracy of scanning with given method. System is applicable to mobile robots and yet fully remotely controlled. The majority of enumerations is taking place in Raspberry Pi 3 model B, just visualization alone is done from computer.

## **KEYWORDS**

3D scanning, structured light, Gray code structured light pattern, calibration, visualization, reconstruction, mesh, point cloud, Raspberry Pi 3 model B, projector

## **Bibliografická citácia:**

HADZIMA, J. *3D skener se strukturovaným osvětlením*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2017. 81 s. Vedoucí bakalářské práce Ing. Aleš Jelínek.

# VYHLÁSENIE

Vyhlasujem, že som svoju bakalársku prácu na téma 3D skener so štruktúrovaným osvetlením vypracoval samostatne pod vedením vedúceho bakalárskej práce a s použitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej bakalárskej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto bakalárskej práce som neporušil autorské práva tretích osôb, hlavne som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý následkov porušenia ustanovení § 11 a nasledujúcich zo zákona č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovení časti druhej, hlavy VI. diel 4 Trestného zákonníku č. 40/2009 Sb.

V Brne dňa .....

.....

(podpis autora)

## PodĎakovanie

Touto cestou by som chcel úprimne poďakovať môjmu vedúcemu práce Ing. Alešovi Jelínkovi za účinnú metodickú, pedagogickú a odbornú pomoc, za zapožičanie potrebného hardvérového vybavenia k vypracovaniu mojej práce a hlavne za prejavenu trpezlivosť.

# Obsah

Úvod.....	10
1. 3D skenovanie.....	11
1.1 Využitie 3D skenerov .....	11
1.2 Rozdelenie technologických metód pre 3D skenovanie.....	12
1.2.1 Kontaktné skenery.....	13
1.2.2 Bezkontaktné skenery .....	14
2. 3D skenovanie s využitím štruktúrovaného osvetlenia.....	17
2.1 Kľúčové vlastnosti 3D skenera využívajúceho štruktúrované osvetlenie nasadeného na mobilnom robotovi .....	18
2.1.1 Konštrukčné a mechanické vlastnosti .....	18
2.1.2 Programové vlastnosti.....	22
3. Základné nastavenie zariadení a ich rozmiestnenie .....	30
3.1 Prvotné nastavenie Raspberry Pi 3 model B.....	30
3.1.1 Nastavenie automatického prenosu súborov na počítač.....	35
3.1.2 Inštalácia Octave [21] .....	36
3.1.3 Pripojenie a nastavenie camera v2 modulu .....	37
3.1.4 Kalibrácia kamery a projektoru.....	37
4. Popis programu pre 3D skenovanie .....	45
4.1 Test camera.....	45
4.2 Get camera/projector calibration data.....	46
4.3 Show triangulation data.....	47
4.4 Start streaming session.....	47
4.5 Jump directly to data gathering.....	48
4.6 Process data .....	49
4.7 View already processed data .....	50
4.8 Export .....	50
4.9 Close uncorrectly closed display .....	52
4.10 Quit .....	52
4.11 3Dvisualizer .....	52
5. Testovanie skeneru.....	54
5.1 Testovanie presnosti .....	54
5.1.1 Presnosť rozmerov a šumu.....	54
5.1.2 Zhoda prekrytím farebnou mapou.....	56

5.2	Testovanie vplyvov prostredia.....	56
5.2.1	Intenzita osvetlenia.....	56
5.2.2	Vhodnosť pozadia za skenovaným objektom .....	60
5.3	Vplyv vizuálnych vlastností skenovaného objektu .....	61
5.3.1	Farba objektu.....	61
5.3.2	Vplyv štruktúrovanosti objektu a zachytenia drobných detailov .....	63
5.3.3	Difúzne svetlo a odrazené štruktúrované vzory .....	64
5.3.4	Reflexné a veľmi lesklé povrchy.....	65
5.3.5	Priehľadnosť materiálu.....	66
5.3.6	Využitie práškových nástrekov na povrch .....	67
6.	Vyhodnotenie experimentov a identifikácia slabých miest metódy.....	70
7.	Záver .....	71
	Citácie .....	72
	Prílohy.....	78

## Zoznam tabuliek

Tabuľka 1:	Základné nastavenie a špecifikácie pre RaspberryPi 3 model B.....	20
Tabuľka 2:	Základné nastavenie a špecifikácie projektoru K11 [13].....	20
Tabuľka 3:	Základné nastavenie a špecifikácie pre CAMERA MODULE V2 .....	21
Tabuľka 4:	Závislosť doby zachytenia 100 obrázkov na rozlíšení kamery .....	27

# Zoznam obrázkov

Obrázok 1: Hlavné rozdelenie metód pre 3D skenovanie [14][5] .....	12
Obrázok 2: Princíp skenovania s využitím štruktúrovaného osvetlenia [10][4] .....	17
Obrázok 3: Postup pri digitalizácii trojrozmerného objektu [11] .....	18
Obrázok 4: Stojan vytvorený pre prepojenie jednotlivých prístrojových častí.....	19
Obrázok 5: Konštrukčné riešenie obalu na Raspberry + kameru.....	19
Obrázok 6: Rozdelenie metód pre 3D skenovanie s využitím štruktúrovaného svetla [27]	22
Obrázok 7: Priebeh premietania štruktúrovaného vzoru v binárnom a Grayovom kóde .....	23
Obrázok 8: Ukážka rozkalibrovania, kamera bola vychýlená o niekoľko stupňov .....	27
Obrázok 9: Výsledný zachytený 3D model pre rozkalibrovaný merací systém .....	27
Obrázok 10: Závislosť trvania programu na rozlíšení .....	28
Obrázok 11: Inštalácia OS Debian Jessie na Micro SD kartu.....	30
Obrázok 12: Plocha OS Debian Jessie .....	31
Obrázok 13: Konfiguračná utilita pre Debian Jessie .....	32
Obrázok 14: Konfiguračná utilita .....	32
Obrázok 15: Nastavenie statickej IP adresy.....	33
Obrázok 16: Prístup k zdieľaným médiám.....	36
Obrázok 17: Kalibračný vzor pre Jean-Yves Bouguet toolbox .....	38
Obrázok 18: Hlavné menu kalibračného nástroja Jean-Yves Bouguet calibration Toolbox	38
Obrázok 19: Ukážka nasnímaných obrázkov v Jean-Yves Bouguet calibration Toolbox ...	39
Obrázok 20: Odhadované pozície rohov štvorcov šachovnice a skutočné nájdené rohy štvorcov šachovnice .....	39
Obrázok 21: Zobrazenie pixelových chýb jednotlivých kalibračných obrázkov .....	40
Obrázok 22: Projekcia extrinsických parametrov kalibrácie .....	40
Obrázok 23: Výpis základných získaných kalibračných dát a pixelových chýb .....	40
Obrázok 24: Zobrazenie pixelových chýb jednotlivých kalibračných snímkou .....	41
Obrázok 25: Vizualizácia radiálneho a tangenciálneho modelu skreslenia .....	42
Obrázok 26: Kombinácia radiálneho a tangenciálneho modelu skreslenia.....	43
Obrázok 27: Hlavné menu kalibračného softvéru pre projektor .....	43
Obrázok 28: Výber rohov vytlačeného vzoru pre kalibráciu projektora.....	44
Obrázok 29: Ukážka intersekcie projektora a kamery .....	44
Obrázok 30: Hlavné menu programu pre 3D skenovanie.....	45
Obrázok 31: Flowchart pre možnosť Test camera .....	46
Obrázok 32: Flow chart pre možnosť Get camera/projector calibration data.....	46
Obrázok 33: Ukážka kalibrácie kamery .....	46
Obrázok 34: Show triangulation data.....	47
Obrázok 35: Flow chart pre možnosť Streaming.....	48
Obrázok 36: Flow chart pre možnosť Jump directly to data gathering.....	48
Obrázok 37: Priebeh skenovania hrnčeku po nanesení práškoveho náteru (1.20.5).....	49
Obrázok 38: Flow chart pre možnosť Process data .....	50
Obrázok 39: Časť Export z hlavného menu .....	50
Obrázok 40: Výberové okno pre export.....	51
Obrázok 41: Export na disk .....	51
Obrázok 42: Flowchart pre Export dát.....	51
Obrázok 43: Spracovanie dát priložených s kalibračnými dátami od tvorca SLPROCESS	53
Obrázok 44: Vizualizácia spracovaných priestorových tvorcov SLPROCESS [14] .....	53
Obrázok 45: Meranie presnosti metódy pre 3D skenovanie .....	54
Obrázok 46: Vlastnosti kalibrácie a zmeny osvetlenia na výsledný šum .....	55
Obrázok 47: Vplyv osvetlenia prostredia na kvalitu skenovania.....	57
Obrázok 48: Výsledný rozoznaný model pri (1) maximálnej hodnote jasnosti projektora a absolútnej tme v miestnosti.....	58
Obrázok 49: Výsledný rozoznaný model pri (4) vysokej hodnote jasnosti projektora a vysokého stupňa vonkajšieho osvetlenia .....	58



Obrázok 50: Čierny, jemne štruktúrovaný objekt pri absolútnej tme v miestnosti a bežnom osvetlení .....	59
Obrázok 51: Rekonštrukcia čierneho, jemne štruktúrovaného objektu pri absolútnej tme v miestnosti a maximálnom jase projektora.....	59
Obrázok 52: Rekonštrukcia čierneho, jemne štruktúrovaného objektu pri bežnom osvetlení v miestnosti a priemernom jase projektora .....	60
Obrázok 53: Nevhodnosť pozadia za objektom, ktorý chceme rekonštruovať .....	60
Obrázok 54: Vplyv farby skenovaného objektu .....	61
Obrázok 55: Rozoznanie jednotlivých radov a stĺpcov pre rekonštrukciu farebných objektov .....	61
Obrázok 56: Výsledná rekonštrukcia farebných objektov .....	62
Obrázok 57: Neschopnosť rozoznania tieňov pre veľmi tmavé až čierne objekty .....	62
Obrázok 58: Nasnímanie krepového papiera pri meraní detailnosti objektu.....	63
Obrázok 59: Zachytený detail povrchu krepového papiera .....	63
Obrázok 60: Vplyv odrazených štruktúrovaných vzorov od zrkadlovitého predmetu .....	64
Obrázok 61: Rekonštrukcia objektu s vysokou mierou difúzie alebo odrazov .....	65
Obrázok 62: Vplyv reflexných povrchov na rekonštrukciu objektu .....	65
Obrázok 63: Získanie riadkov a stĺpcov pre rekonštrukciu priehľadných objektov .....	66
Obrázok 64: Pohľad na rekonštruovaný priehľadný objekt z pohľadu kamery .....	66
Obrázok 65: Pohľad na rekonštruovaný priehľadný objekt, pohľad zhora.....	67
Obrázok 66: Predmet po a pred nanosením práškového náteru .....	68
Obrázok 67: Výpočet riadkov a stĺpcov pre rekonštrukciu predmetu ošetrovaného práškovým náterom .....	68
Obrázok 68: Rekonštrukcia lesklého objektu bez práškového náteru .....	69
Obrázok 69: Rekonštrukcia lesklého objektu po nanosení práškového náteru .....	69

# Úvod

3D skenovanie nie je v dnešnej dobe žiadnou novinkou. Jeho použitie je stále častejšie a na zväčšujúcom sa množstve aplikácií v rôznych odvetviach. 3D skener je zariadenie na vytváranie trojrozmerných modelov vyhotovených digitalizáciou reálnych objektov so snahou o čo najvyššie rozlíšenie a veľkú presnosť. Najčastejšie je 3D skener vytvorený pomocou stereovízie dvoch fotografických prístrojov a štruktúrovaného svetla alebo laseru. Výsledkom skenovania v troch rozmeroch je tzv. mračno bodov, ktoré môže obsahovať až stovky miliónov bodov. Celý priebeh skenovania je riadený počítačovým softvérom. Kvalita výsledného modelu pri tom závisí od vhodnej implementácii metód pre získanie polohy bodov v priestore s využitím lúčovo planárnych výpočtov. Pri výrobe a návrhu nových súčastí môžeme postupovať buď vytvorením úplne novej súčiastky alebo použiť metódu Reverse engineering. Pri tejto metóde sa už vytvorená súčiastka naskenuje a vytvorí sa model ktorý sa podľa potrieb upraví. Táto metóda sa využíva hlavne vtedy, keď nevieme presné rozmery alebo vnútornú štruktúru danej súčiastky.

Táto práca je zameraná na problematiku bezkontaktného optického merania tvaru a proporcií analyzovaného objektu s využitím premietania štruktúrovaného osvetlenia. Výsledkom práce by malo byť zariadenie, ktoré je schopné automatizovaného merania, digitalizácie a aj vizualizácie skenovaného objektu. Celé toto zariadenie bude pri tom osadené na mobilnom robotovi a ovládané na Raspberry Pi 3 model B.

V prvej a druhej kapitole sa venujem základnému zhodnoteniu jednotlivých metód skenovania. Hlavnú pozornosť pri tom venujem komplexnejšej prezentácii skenovania s metódou využívajúcou štruktúrované vzory. V práci sú rozpísané hlavné vlastnosti a požiadavky, kladené na skenovanie touto metódou, opis priebehu digitalizácie modelu od zosnímania obrázkov až po vizualizáciu. Venujem sa aj kalibrácii a možným príčinám vzniku chýb. V tretej kapitole je z dôvodu reprodukovateľnosti merania uvedený kompletný postup nastavenia jednotlivých zariadení. Na zariadenie je kladený predpoklad využitia daného systému na mobilnom robotovi, preto sa uberám cestou využitia Raspberry Pi ako riadiacej jednotky. V štvrtej časti prezentujem mnou vytvorený program, ktorý zabezpečuje všetky potrebné kroky od nasnímania až po rekonštrukciu objektu. Využívam pri tom Jean-Yves Bouguet Calibration Toolbox a jeho nadstavbu ProCamCalib, ktoré sú určené na kalibráciu kamery a projektoru. Ďalej využívam kód edukačného charakteru BYO3D Scanner, vytvorený na Brownovej univerzite. Samotný program je upravený a spúšťaný z Pythonu a ovládaný pomocou vzdialenej plochy z počítača. Piata časť sa zaoberá vyhodnotením problematickým miest, ktoré môžu vzniknúť pri procese digitalizácie modelu a šiesta časť obsahuje informačné zhrnutie experimentov.

## 1.3D skenovanie

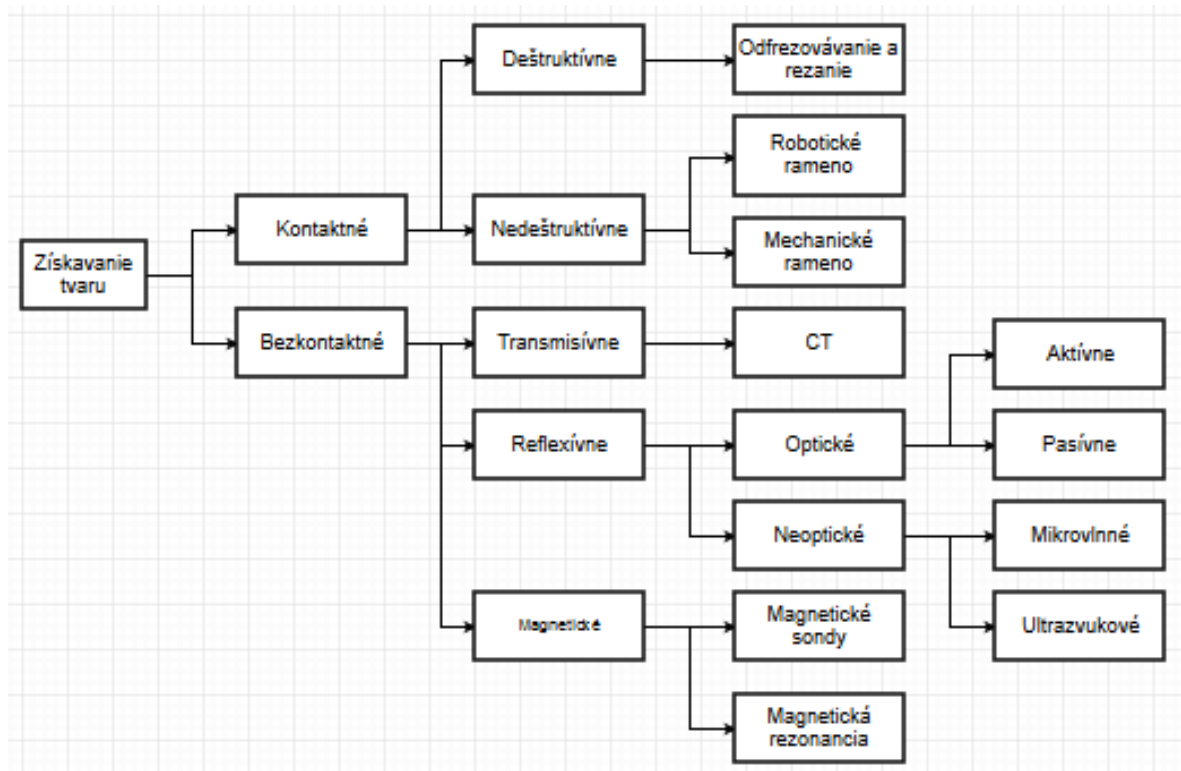
V tejto dobe sa spoliehame stále častejšie a častejšie na stroje. Stali sa pre nás dobrým pomocníkom a aj spôsobom, ako si prácu zjednodušiť alebo ju výrazne skrátiť. Množstvo strojov a automatizácie vo výrobných zariadeniach po celom svete uľahčilo prácu miliónom ľudí. Okrem samotnej výroby je dôležitým krokom aj návrh samotného dizajnu, vhodné zvolenie objektových štruktúr, vnútorných mechanizmov a vhodný návrh zakomponovania všetkých súčastí. Zariadení, umožňujúcich meranie priestorových vlastností objektov je veľké množstvo. Môže sa jednať o prenosné ručné zariadenia, ktoré sme schopné priniesť až k predmetu alebo veľké neprenosné robustné prístroje, ktoré si vyžadujú vkladanie predmetov do ich zorného poľa. Neprenosné skenery sú pevne prichytené s podložkou a analyzované objekty sa buď ručne, automatizovane robotickými ramenami alebo s využitím dopravníkov dostávajú do aktívnej meracej oblasti.[1] Skenery sa rovnako rozlišujú aj kvalitou a rýchlosťou získavania dát. Niektoré zariadenia snímajú bod po bode, iné sú schopné zachytiť širokú plochu. Väčšina skenerov poskytuje možnosť vytvárania počítačových modelov využitím mračna bodov, polygónov, kriviek typu spline alebo iných štandardných geometrických entít. V tejto kapitole je načrtnutý rýchly prehľad 3D skenovania a jednotlivých používaných technologických metód.

### 1.1 Využitie 3D skenerov

3D skenery si našli uplatnenie v širokom spektre odlišných odvetví. Stále širšie uplatnenie trojrozmerných skenerov dovoľuje rozsiahlejšiu automatizáciu procesov. V budúcnosti sa očakáva výrazné zvýšenie použitia týchto zariadení. V priemysle sa využívajú napríklad pre kontrolu kvality, reverse engineering, kontrolu robotov kde zaobstarávajú funkciu ľudského oka, v zisťovaní rotácie alebo poškodenia. Využitie si našli aj pri výrobe filmov a hier. Virtuálna realita dovoľuje zobrazovanie vlastných digitalizovaných objektov v priestore a teší sa stále väčšej obľube. V internetových obchodoch sa stále častejšie nachádza možnosť zobrazenia tovaru v 3D. Širokoplošné využite predstavujú prieskumy terénu alebo rozšírenia typu Google street view. S využitím 3D skenerov môžeme získať modely kultúrnych diel a exponátov nedeštruktívnym spôsobom. V zdravotníctve sa využívajú na modelovanie častí tela a v stomatológii k vytvoreniu modelov chrupu. [2][3]

## 1.2 Rozdelenie technologických metód pre 3D skenovanie

V súčasnosti sú v priemyselnej oblasti najrozšírenejšie nedeštruktívne metódy robotických rúk ktoré snímajú bod po bode alebo optické laserové a optické snímače s využitím bieleho svetla- štruktúrovaných vzorov. V Európe prevažuje použitie snímania založené na bielom svetle a v Severnej Amerike na laserovej technológii. Celkové delenie 3D skenerov podľa spôsobu snímania je naznačené v obrázku 1.



Obrázok 1: Hlavné rozdelenie metód pre 3D skenovanie [14][5]

Prevažná väčšina dnes používaných komerčných produktov môže byť rozdelená na (prevzaté z [3]) :

- laserový snímač povrchu pripevnený na CMM,
- prenosný systém na projekciu bieleho alebo kódovaného svetla,
- stolná verzia s laserovým profilometrom alebo systémom s kódovaným svetlom na multiosej platforme,
- laserové bodové meracie systémy, kde sa aj meraný objekt, aj senzor pohybujú,
- ručne ovládané laserové profiléry alebo bodové meracie prístroje.

## 1.2.1 Kontaktné skenery

Nachádzajú sa v priamom styku s meraným objektom. Sú pomalšie ako bezkontaktné a dokážu zabrať menšiu plochu ale ich výhodou je schopnosť analyzovať aj malé otvory a štruktúry. [1] Tieto skenery môžeme rozdeliť ďalej na deštruktívne a nedeštruktívne.

**Deštruktívny typ** skeneru dokáže okrem povrchu analyzovaného objektu získať dáta o vnútornej štruktúre, avšak pri tomto procese prichádza k deštrukcii daného objektu. Pred začiatkom skenovania je analyzovaný predmet prichytený na rám, kde je pokrytý špeciálnym materiálom, ktorý sa po vyčerpaní všetkého vzduchu dostane do všetkých medzier. Tento materiál poskytuje vysoký farebný kontrast medzi súčasťami a výplňovým materiálom. Následne sa objekt pripevnený v ráme umiestni do frézovacieho stola kde prebieha odfrézovanie veľmi malých vrstiev materiálu cirkulárnou frézku pre presné obrábanie a zaznamenávanie optickým skenerom. Výsledný 3D model je teda poskladaný z jednotlivých skenov.[6] Tento systém skenovania je asi o 70 až 80% lacnejší ako röntgenový 3D skener.

**Nedeštruktívne typy** nespôsobujú pri bežnej prevádzke poškodenie meraného objektu, pri nesprávnej manipulácii však škody na predmete môžu nastať, preto sa nehodí na meranie drahých objektov ako sú cenné historické pamiatky. Jedná sa o veľmi presné meranie ktoré je relatívne pomalé. Dosahujeme pri ňom iba niekoľko stoviek Hz a pri optických metódach dosahujeme kHz. Výstupom je usporiadané mračno bodov. Zatiaľ čo je objekt nehybne pripevnený, CMM, robotické alebo mechanické rameno prechádza po povrchu bodovou alebo guľôčkovou sondou a umožňuje analýzu povrchu a priestorových súradníc bodov pomocou hmotného dotyku. [6] Výhodou je určite aj možnosť zmeny nástrojov počas merania jedného objektu, dokážeme tak získať aj ťažšie dostupné miesta. Podľa potreby môžeme pre meranie s bodovou alebo guľôčkovou sondou použiť aj ručné skenery. Ručné polohovanie pohyblivého ramena sa využíva na dizajnové návrhy, kde nie je potrebná vysoká presnosť. Touto metódou sme schopní získať rýchle dizajnové krivky, meriame ňou niekoľko konkrétnych bodov v priestore. Jedná sa o zariadenie, vykonávajúce pohyb. To ich robí náchylné na vibrácie a vonkajšie vplyvy, ktoré môžu vplývať na kvalitu výsledku.

## 1.2.2 Bezkontaktné skenery

Nenachádzajú sa v priamom kontakte s meraným povrchom, nesie to so sebou obrovské výhody. Napríklad odpadá potreba vkladať predmet do meracieho zariadenia alebo možnosť skenovania objemných a ťažkých predmetov. Vďaka ich jednoduchosti a rýchlosti sú najrozšírenejšou metódou skenovania. Najpoužívanejšie sú bodové alebo profilové laserové skenery a optické skenery na báze štruktúrovaného osvetlenia.

### 1.2.2.1 Transmisívne

Zástupcami transmisívnych skenerov sú skenery využívajúce technológiu počítačovej tomografie. Technológiou počítačovej tomografie, rovnako ako pri skeneroch využívajúcich magnetickú rezonanciu je možné získať údaje o vnútornej stavbe skúmaného objektu. Pre prenos informácií sa využíva röntgenové žiarenie. Plynie z toho aj jeho obmedzenie, je ním možné merať iba objekty z vhodného materiálu a je nutné daný objekt umiestniť do meracieho zariadenia. Na rozdiel od zdravotných verzií CT (Computer Tomography) sa pri tomto použití využíva vyššia intenzita žiarenia . [6]

### 1.2.2.2 Magnetické

Rozdeľujú sa na magnetické sondy a prístroje využívajúce magnetickú rezonanciu. Prístroje využívajúce magnetickú rezonanciu pre 3D skenovanie dokážu získať dáta o vnútornej štruktúre súčastky, rovnako ako obdobné prístroje ktoré využívajú magnetickú rezonanciu v zdravotníctve . Nevýhoda je nutnosť ich umiestnenia do meracieho zariadenia a aj ich cena. Rozdiel od iných 3D skenerov je aj to, že výstupom z MRI nie je mračno bodov ale séria 2D snímok umiestnených na sebe.

### 1.2.2.3 Reflexívne

Spôsob zisťovania informácií o ich tvare prípadne štruktúre vychádza z ich odrazivosti určitého typu vln. Buď odrazu svetla, zvuku či mikrovlnných vln.

**Metódy nevyužívajúce optické vlastnosti** nevynikajú svojou presnosťou. Ultrazvukové 3D skenery využívajúce sonar alebo iné ultrazvukové snímače vysielajú signál z ultrazvukovej sondy v tvare pištole s kovovým hrotom, ktorá sa prikladá k meranému objektu. Stlačením spúšte je vyslaný zvukový signál. Signál potom zachytí špeciálna konštrukcia s ultrazvukovými snímačmi, ktorých úlohou je signál prijať a dekodovať do priestorových súradníc. Ich výhodou je nízka cena, ktorá ich robí ideálnymi pre kontrolu kvality v oblastiach priemyslu, ktoré nevyžadujú vysokú kvalitu.[6]

Mikrovlnné radary sa používa na detekciu prítomnosti alebo povrchu vo veľkých ale aj malých vzdialenostiach, v závislosti na vyhotovení . Princíp radaru je založený na vysielaní magnetických vln prostredím a ich odraze od objektu. Po

odraze sú zachytávané anténou a ďalej vyhodnocované. Okrem prítomnosti, veľkosti a tvaru dávajú aj informáciu o jeho rýchlosti pohybu. [7][8]

**Metódy využívajúce optické vlastnosti** najčastejšie snímajú objekt z niekoľkých uhlov za pomoci optického zariadenia. Pretože optické skenery zachytávajú v jednom momente veľké množstvo dát a snímajú veľkú plochu, musí predmet počas skenovania jedného snímku zostať v nehybnej polohe. Pred samotným skenovaním sa na objekte môžu vyznačiť body, ktoré slúžia na zarovnanie skenov cez seba pri tvorbe 3D modelu. Tieto orientačné body sa nazývajú referenčné body a lepia sa priamo na povrch snímaného objektu. Ich množstvo môže dosahovať počtu aj niekoľko tisíc. Problém pri skenovaní optickou metódou nastáva ak je povrch veľmi lesklý alebo zrkadlový. Rovnako vznikajú problémy pri meraní transparentných materiálov alebo veľmi tmavých materiálov. [6] [2]

Pri optických skeneroch sa využívajú najčastejšie laserové bodové alebo plošné skenery a skenery so štruktúrovaným osvetlením. Laserové skenery sú určené k rýchlemu zosnímaniu povrchu. Využíva sa tu buď bodového laserového lúču, ktorý je vysielaný na povrch analyzovaného objektu a postupne zosníma celý objekt, prípadne zosníma iba ťažko zoskenovateľné oblasti ako otvory pri priemyselnej výrobe. Podľa odrazu a doby letu laserového lúču sa vyhodnotí poloha a vzdialenosť objektu od meracieho zariadenia. Medzi nevýhody patrí napríklad vyššia kúpna cena a chýbajúca informácia o textúre povrchu, preto sa často laserové skenery vybavujú aj kamerami s CCD snímačmi. Druhým spôsobom je využívanie laserového prúžku, ktorým sa postupne prechádza po celom objekte a sníma sa jeho profil. Laserový prúžok je snímaný špeciálnym CCD snímačom. Rýchlosť pohybu je limitovaná dobou expozície a dobou vypočítavania z CCD snímača, ktorá sa pohybuje od 10ms do 100ms v závislosti na svetelnom výkone laseru. Toto riešenie je výrazne rýchlejšie ako meranie s použitím bodového laseru.[1] Pri optických 3D skeneroch využívajúcich štruktúrovaného osvetlenia sa využíva nadefinovaný vzor, ktorý je premietaný na meraný objekt. Tento objekt je snímaný kamerou ktorá sa nachádza pod uhl'om od žiariču svetla, najčastejšie projektoru. Ak by vzor osvetľoval iba rovnú plochu, vzor by obsahoval najčastejšie rovné línie. Ak je vložený objekt do uhlu vyžarovania projektoru, tak podľa úrovne deformácie línie sa vyhodnotí tvar predmetu. Táto metóda je pomalšia na spracovanie, avšak skenovanie prebieha rýchlejšie ako pri bodovom laserovom skenovaní. Pre Zistenie komplexnejšej informácie o tvare objektu je ho nutné zosnímať z viacerých strán.[9]

Pasívne metódy - Princíp je analýza obrazu z pohybujúcej sa kamery prípadne snímača, alebo rekonštrukcia 3D povrchu z viacerých 2D obrazov. V rámci geometrického usporiadania meracieho systému tu nie je zahrnutý zdroj osvetlenia. Energia je v tomto prípade iba prijímaná, nie je žiadnym spôsobom emitovaná. Tým, že neemitujú na objekt žiaden typ žiarenia neprichádza k ovplyvňovaniu objektu. Pre modely prekryté farebnou mapou to môže byť obrovská výhoda. Pre systémy s dynamickým snímaním je uprednostňované použitie viacerých kamier so samokalibráciou.[6] Z týchto metód má najväčšie uplatnenie metóda stereovízie, ktorá je založená na princípe stereoskopie, kde je obraz zachytávaný dvoma kamerami z iného uhla pohľadu. Výsledná geometria sa získa z rozdielných pozícií bodov a vzájomnej orientácie kamier. Nevýhoda tejto metódy je v nepresnej korešpondencii bodov s nevýraznou textúrou. Z tohto dôvodu sa táto metóda často kombinuje s aktívnymi metódami. [10]

Medzi pasívne metódy patrí (prevzaté z [10]):

- Stereovízia
- Analýza pohybu
- Získavanie tvaru z tieňu objektu
- Využívanie siluet (kontúr) objektu
- Zmeny ohniska – rozostrenia
- Textúra objektu

Aktívne metódy - Na rozdiel od pasívnych metód je na skenovaný objekt emitovaný nejaký druh energie. To môže byť viditeľné svetlo, zvuk alebo akékoľvek elektromagnetické žiarenie. Lúč tohto žiarenia je nemierený na povrch snímaného telesa a ten sa následne odráža späť do snímača. Podľa úrovne deformácie reflektovaného svetelného odrazu sa zistí 3D povrch predmetu. Optické metódy majú veľmi široké uplatnenie v mnohých odvetviach. Sú rýchle, presné a často vyhotovené ako mobilné zariadenia.[14]

Medzi aktívne metódy patrí:

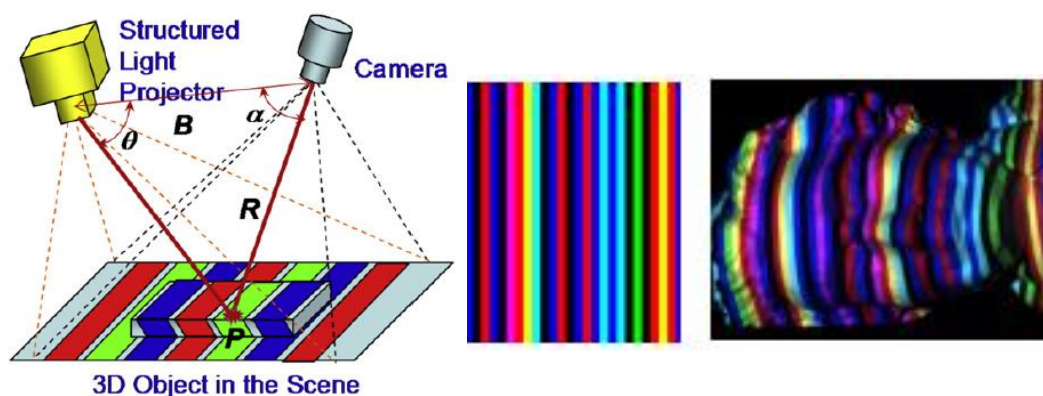
- Triangulácia – zo znalosti uhlov a natočenia medzi kamerami, prípadne aj projektorom dokážeme zo znalosti uhlov trojuholníka vypočítať polohu v priestore.
- Meranie doby letu svetla (time of flight) – laserové meracie zariadenia vyhodnocujúce vzdialenosť predmetu zo znalosti doby letu svetla, zástupca napríklad Lidar
- Interferometrické skenery – pre skenovanie mikroskopických štruktúr, zakladá sa na princípe skladanie(superpozícia) svetelných vln



## 2.3D skenovanie s využitím štruktúrovaného osvetlenia

Jedná sa o aktívny optický spôsob získavania informácií o hĺbke a tvare objektu, ktorý využíva pri svojej činnosti projektor a jeden alebo viac fotoaparátov. Využíva pri tom aktívnu trianguláciu. Projektor premietá predefinované vzory na skenovaný objekt, fotoaparáty snímajú osvetlený predmet. Táto metóda je založená na unikátnom kódovaní každého svetelného znaku premietaného na objekt. Následne ak je nájdená zhoda v obrázku, vzájomná korešpondencia je priamo vyriešená odkódovaním.

Ak by sa v ceste projekcie vzoru nenachádzal žiaden objekt, iba rovná plocha, výsledný obraz premietaného vzoru na rovnej ploche by bol podobný tomu vyžiarenému, prípadne rozmerovo skreslený. Ak ale vložíme predmet do uhlu premietania vzoru, môžeme vidieť z uhlu pohľadu kamery deformáciu vzoru. Podľa rozdielu odhadovaného miesta, kde by línia štruktúrovaného vzoru zasahovala v prípade, že by nebol použitý predmet a z reálne zachyteného miesta sme schopný získať údaj o tvare objektu. Projektor a fotoaparát je pri skenovaní natočený pod známym nenulovým uhlom (väčšinou  $15^\circ$ ) od analyzovaného objektu. Na obrázku 2 je vidieť princíp skenovania pre využitie farebných vzorov. V strede sa nachádza obraz premietaný projektorom a v pravej časti obrázku sa nachádza obraz zachytený kamerou.

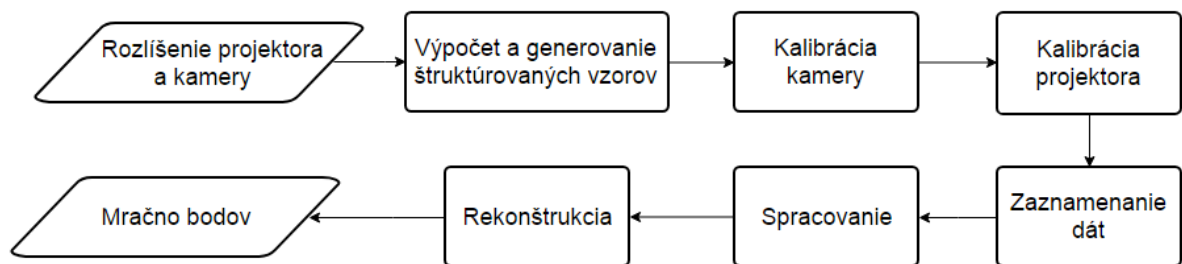


Obrázok 2: Princíp skenovania s využitím štruktúrovaného osvetlenia [10][4]

Skenovanie s využitím štruktúrovaného svetla od svojho prvého použitia dosiahlo výrazné zlepšenie svojich vlastností. Výskum bol zameraný hlavne na zmenšenie doby, potrebnej k digitalizácii objektu a k zlepšeniu presnosti hĺbkového odhadu. V súčasnosti sme schopný skenovať s rýchlosťou až 1000Hz a hĺbkovým rozlíšením 30 mikróvov. Každým zlepšením jedného z krokov zo sekvencie, potrebných k vyhotoveniu 3D modelu, začínajúc vytvorením štruktúrovaných vzorov až po kompletnú digitalizáciu, môžeme zlepšiť celkovú kvalitu výsledného modelu.

## 2.1 Klúčové vlastnosti 3D skenera využívajúceho štruktúrované osvetlenie nasadeného na mobilnom robotovi

Detailnejším prezretím skenovacieho reťazca (Obrázok 3) sme schopní určiť hlavné klúčové vlastnosti skeneru, využívajúceho štruktúrované osvetlenie. V našom prípade sa jedná o skener, nasadený na mobilnom robotovi, preto musíme zahrnúť aj tieto jednotlivé parametre. V nasledujúcej časti preberiem jednotlivé klúčové vlastnosti s aplikáciou pre naše meranie. V texte sa budú nachádzať konkrétne použité zariadenia a ovládací softvér pre náš prípad.



Obrázok 3: Postup pri digitalizácii trojrozmerného objektu [11]

### 2.1.1 Konštrukčné a mechanické vlastnosti

Počas skenovania s využitím štruktúrovaného osvetlenia musí predmet, ako aj všetky časti skenovacieho zariadenia zostať nehybné. Stabilné statické prepojenie hardvérových častí zabezpečuje mechanická konštrukcia. Pre účely testovania som vytvoril konštrukciu, ktorá je polohovateľná. Kamera je v našom prípade zafixovaná na Raspberry Pi 3 model B.

#### 2.1.1.1 Mechanická konštrukcia

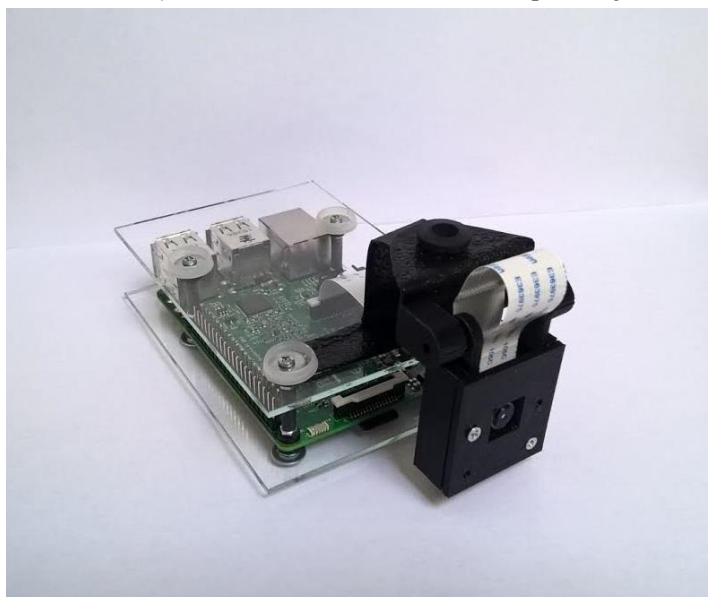
Metóda skenovania využívajúca štruktúrované vzory potrebuje pre svoju funkčnosť iba zdroj svetla, ktorý bude premietat' vzory a kameru, ktorou budeme snímať osvetľovaný predmet. Spôsob pripojenia k mobilnému robotovi nebol špecifikovaný. Konštrukcia je na stojan pripojená pomocou skrutky so šesťhrannou hlavou veľkosti M8 x 20. Na obrázku 4 je vidieť celá konštrukcia aj s hardvérovými súčasťami. V prílohe č. 3 je pohľad na zariadenie zhora. V prílohe č. 4 sa nachádza realizácia pripojenia projektora na konštrukciu a v prílohe č. 5 sa nachádza ukážka polohovateľnosti zariadenia a nastavenia priblíženia kol'ajničkami. Rameno dokáže počas nastavovania spraviť plné otočenie hlavy o 360° a kol'ajničky dovoľujú predĺženie ramena až o 13 cm na obe strany.



**Obrázok 4: Stojan vytvorený pre prepojenie jednotlivých prístrojových častí**

### **Obal na Raspberry Pi a kameru**

Obal na Raspberry Pi by mal byť konštrukčne odolný, pretože pri použití na mobilnom robotovi môže ľahko prísť k mechanickému zásahu do zariadenia a jeho následnému poškodeniu. Rovnako musí byť stavaný na možné vibrácie spôsobené pohybom robota. Vhodným obalom môže byť napríklad na mieru vyrobená krabička s pomocou 3D tlačiarne alebo krabička vyrobená z plexiskla. Z cenových dôvodov som zvolil ako obalový materiál sklo, síce sa jedná o krehký materiál ale pre účel testovania to nehrá rolu. Plexisklo sa vizuálnymi vlastnosťami podobá sklu ale má vhodnejšie vlastnosti väčšiu odolnosť. Na obrázku 5 je naznačený vzhľad obalu na Raspberry Pi. Obrázky z jednotlivých uhlov sa nachádza v prílohe č.1. Rozmery dier sa zhodujú s rozmermi zariadenia z prílohy č.2.



**Obrázok 5: Konštrukčné riešenie obalu na Raspberry + kameru**

### 2.1.1.2 Hardvér

Na to aby sme mohli zachytiť samotný trojrozmerný tvar objektu s využitím štruktúrovaného osvetlenia potrebujeme projektor alebo zdroj osvetlenia, ktorý bude premietat' dopredu predefinované vzory, jednu prípadne viac kamier, ktoré budú snímať predmet a zariadenie ktoré bude spracovávať dáta z kamery a posielat' príkazy projektoru. Vizualizácia bude prebiehať na počítači, ktorý obsahuje Matlab a pre zobrazenie 3D modelu aj napríklad Meshlab.

#### Raspberry Pi 3 model B

Raspberry Pi je malý a lacný jednodoskový minipočítač (cena iba cca 35€). Navzdory svojej veľkosti ale prezentuje relatívne vysoký výkon a dobré špecifikácie. Jeho využitie je hlavne v oblasti vzdelávania na školách ale nachádza uplatnenie aj v zložitejších domácich menších projektoch. Pre skenovanie objektov nám vyhovuje použitie Raspberry Pi 3 hlavne z dôvodu jeho jednoduchosti.[12]

**Tabuľka 1: Základné nastavenie a špecifikácie pre RaspberryPi 3 model B**

CPU	1.2GHz 64-bit quad-core ARMv8
RAM	1GB
Napájanie	5V 2,5A (0,7A vlastná spotreba + periférie)

Na Raspberry sa môžeme pripojiť pomocou 4 USB portov, Ethernet portu. Obsahuje slot na Micro SD kartu. Okrem toho ponúka možnosť Bluetooth 4.1, 802.11n Wireless LAN. Dobrá kompatibilita s klávesnicami a myšami. Z Raspberry môžeme streamovať video okrem Wlan a Ethernetu aj pomocou HDMI portu, portu pre kompozitné video/audio ale aj vlastného DSI Display Portu pre pripojenie malej obrazovky. Výhodná je aj možnosť pripojenia kamery pomocou CSI Portu.

#### Projektor

RPi ponúka dve výstupné rozhrania pre pripojenie zobrazovacích jednotiek. HDMI pre vysoké rozlíšenie a kompozitný videosignál pre nízke rozlíšenie. Port DSI je určený pre vlastné malé obrazovky ale v tomto projekte nebol použitý. Pre premietanie štruktúrovaných vzorov bude použitý projektor Acer K11, ktorý je pripojený k Raspberry Pi 3 model B pomocou HDMI 1.4 kábla.[13] Na Raspberry je možné si nastavenie rozlíšenia overiť pomocou príkazu v terminále:

```
~ $ tvservice -s
```

Pri správnom nastavení by nám malo vypísať nasledujúci riadok:

```
state 0x12000a [HDMI DMT (18) RGB full 4:3], 1024x768 @ 75.00Hz, progressive
```

Ak sa nastavenie nebude zhodovať, v časti inicializácie zariadenia v kapitole 3 je popísaný postup k nastaveniu.

**Tabuľka 2: Základné nastavenie a špecifikácie projektoru K11 [13]**

Napájanie	19V DC 4,7A 90W
Vzdialenosť pre premietanie	0,64m až 2,40m
Uhlopriečka premietanej plochy	0,64m až 3,40m
Rozlíšenie	1024x768 4:3
Frekvencia	75Hz

## Kamera

Pre zaznamenávanie obrazu bola použitá štandardná *Camera Module V2* pripojená pomocou plochého flex káblu na Raspberry Pi. Raspberry Pi nám zabezpečí plne automatizované ovládanie a zaznamenávanie obrazu kamerou a jednoduchosť kódovej implementácie. Kamera nám zároveň zabezpečuje splnenie podmienky pre 3D skenovanie, ktorá hovorí, že rozlíšenie kamery by malo byť aspoň o niečo lepšie ako pixelové rozlíšenie projektoru.

**Tabuľka 3: Základné nastavenie a špecifikácie pre CAMERA MODULE V2**

Senzor	Sony IMX219 (CMOS)
Rozlíšenie	8MP (3280x2464)
Snímkovacia frekvencia	Max 90fps pre väčšie rozlíšenia
Výstupný formát obrázkov	PEG, JPEG + RAW, GIF, BMP, PNG, YUV420, RGB888
Výstupný formát videa	raw h.264 (accelerated)

### 2.1.1.3 Napájanie

Jedna z najnáročnejších vecí na tomto projekte je práve spôsob jeho napájania. Samotné Raspberry Pi nepotrebuje veľké napätie a ani prúd. Pri napájaní s 5.1V by prúd nemal presiahnuť 2A, jeho vlastná spotreba je okolo 0.5 až 0.7 A. Zvyšok je dostupný pre jednotlivé periférie a zariadenia, napríklad zariadenia pripojené pomocou USB slotu alebo displeje či kameru. Projektor v nečinnom stave s neaktivovaným premietaním potrebuje iba 0.2W, avšak v stave aktívneho premietania sa dostáva až na 80W. Pre účely testovania je vhodné napájať aj Raspberry aj projektor z ich vlastných adaptérov. Pri nasadení zariadenia ako hlavy mobilného robota bude nutné napájať skener z batérie, rovnako aj vlastné motory a senzory motora. V prípade pripojenia na fixované robotické rameno, predpoklad napájania bude zo sieťových zdrojov.

### 2.1.1.4 Ovládanie

Výhodou využívania Raspberry Pi je aj jeho jednoduchá konfigurovateľnosť. Dovoľuje nám okrem priameho ovládania aj vzdialené (remote control) ovládanie plochy. Práve tento typ ovládania využijeme a dokážeme tým splniť požiadavky kladené na tento výrobok. Ovládanie realizujeme cez Wi-Fi bez nutnosti fyzického pripojenia cez káble. Hlavné výpočty prebiehajú na samotnom Raspberry a až samotná vizualizácia prebieha na PC, tým nezaťažujeme výrazne PC. Zaťažujeme iba sieť prenosom dát modelu. V prípade prenášania modelov bez ošetrovania odstránením redundantných dát a bez komprimácie sa veľkosť modelov môže dostať až na hodnotu 500 Mb, v prípade ošetrovania sa maximálna veľkosť dát pohybovala okolo 80 Mb.

### 2.1.1.5 Mechanická prepojitelnosť na iné zariadenia alebo robota, periférie

Zariadenie je konštruované práve pre pripojiteľnosť na pohyblivého robota, prípadne mechanické rameno. Dopredu nebol stanovený presný typ ani spôsob pripojenia, preto samotné pripojenie nechávam na konkrétnej aplikácii, v ktorej bude použitý. Popis manipulovateľnosti už bol popísaný v časti 2.1.1.1.

## 2.1.2 Programové vlastnosti

V nasledujúcich bodoch preberiem hlavné softvérové vlastnosti spojené s procesom 3D skenovania. Zameriam sa výrazne na skenovanie s využitím štruktúrovaného osvetlenia pre statické objekty.

### 2.1.2.1 Nadviazanie spojenia so zariadením

Po lokálnom povolení SSH a VNC nám Raspberry dovoľuje pripojenie ako aj cez kábel, tak aj cez virtuálny vzdialený prístup. Pre nás je vhodný práve vzdialený prístup z dôvodu komfortnosti. Je silne ovplyvňovaný prenosovou rýchlosťou medzi RPi – Routerom - PC.

### 2.1.2.2 Typy štruktúrovaných obrazcov a ich premietanie

Prvý nutný krok k získaniu informácií o tvare objektu je premietanie špeciálnych vzorov, ktoré v sebe nesú zakódovanú informáciu o polohe každého pixelu projektoru. Je tu niekoľko použiteľných schém pre zakódovanie informácie o polohe pixelu. Rozlišujeme ich využitie hlavne podľa typu použitia, či chceme snímať pohybujúce sa alebo nehybné objekty (Obrázok 6).

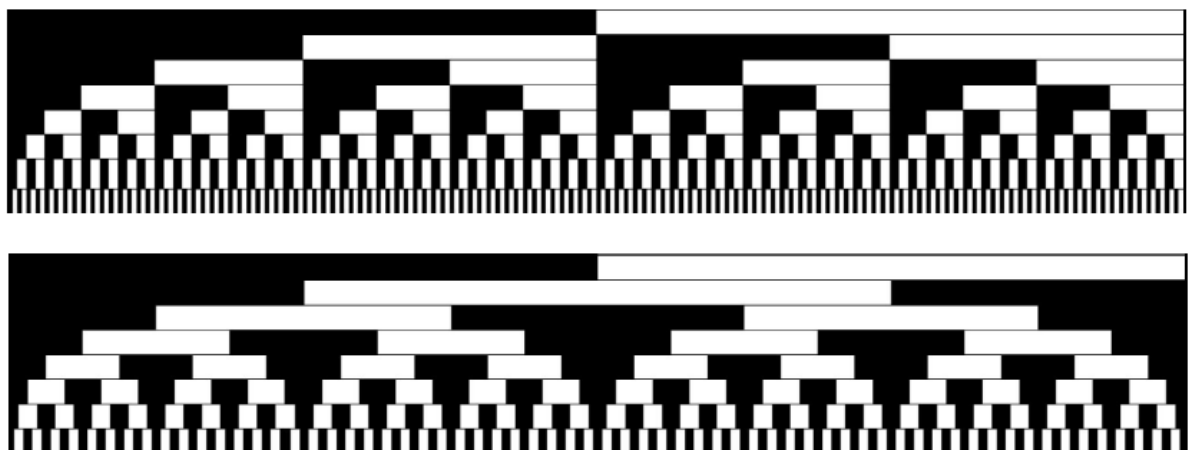


Obrázok 6: Rozdelenie metód pre 3D skenovanie s využitím štruktúrovaného svetla [27]

**Pre snímanie pohybujúcich sa alebo tvarovo premenných objektov sa snažíme získať prehľad o tvare objektu iba z jedného obrázku. Pri pohybe objektu by sme na ďalšej snímke nasnímali úplne odlišné dáta a bolo by príliš komplikované vyhodnotiť tvar objektu. Museli by sme brať do úvahy aj rýchlosť objektu a jeho natočenie a posuv v priestore. Jednotlivé snímky by mohli vykazovať chybu rozmazania spôsobenú telesami pohybujúcimi sa rýchlejšie ako je snímacia rýchlosť. Každá z uvedených metód má svoje výhody a nevýhody. Táto práca je však zameraná na skenovanie nepohybujúcich sa objektov, preto sa budeme venovať radšej podrobnejšiemu opisu princípov týchto metód.**

**Pre skenovanie statických objektov** sú najrozšírenejšie vzory v binárnom, Grayovom kóde alebo vzory s fázovým posunutím. V súčasnosti je možné využiť aj hybridné metódy využívajúce viac ako len čiernu a bielu farbu, napríklad pridaním šedej. Pri vzoroch s Grayovým alebo binárnym vzorom postupne premietame za sebou sekvenčne všetky štruktúrované vzory daného kódu a následne ich snímame. Jedná sa teda o kódovanie pozície umiestnenia pixelov v čase. Každý premietaný pixel má svoju unikátnu pozíciu. Veľkosť zakódovanej informácie závisí aj na rozlíšení projektoru.

Binárny a Grayov kód - Vzor v Grayovom kóde je založený na podobnom princípe ako binárny vzor. Obe sú určené pre snímanie statických, v čase skenovania nehybných objektov. Vzor v Grayovom kóde vznikne prepočítaním binárneho vzoru. Rozdiel je v usporiadaní a následnosti prúžkov. Pri Grayovom kóde je rozdiel medzi dvoma nasledujúcimi kódovými hodnotami vždy maximálne jedna. Na nasledujúcom obrázku (Obrázok 7) je v hornej časti vidieť priebeh štiepenia binárneho kódu a v dolnej časti Grayov kód.



Obrázok 7: Priebeh premietania štruktúrovaného vzoru v binárnom a Grayovom kóde [14]

Prepočet binárneho kódu do Grayovho a naopak môžeme nasledujúcim spôsobom:

*Binárny kód do Grayovho* [11]:

1. Prvé číslo sprava resp. Most significant bit sa skopíruje z binárneho vzoru
2. Druhé číslo získame numerickým výpočtom XOR prvého a druhého bitu sprava
3. Tretie číslo získame výpočtom XOR druhého a tretieho bitu sprava atď..

*Grayov kód do binárneho* [11]:

1. Prvé číslo sprava resp. Most significant bit sa skopíruje z Grayovho vzoru
2. Druhé číslo získame numerickým výpočtom XOR prvého bitu už v prepočítanom binárnom kóde a druhého bitu v Grayovom kóde
3. Tretie číslo získame numerickým výpočtom XOR druhého bitu už v prepočítanom binárnom kóde a tretieho bitu v Grayovom kóde atď.



Pre skenovanie s našim zvoleným softvérom, s ktorým sa zoznámime neskôr, môžeme použiť ako binárny, tak aj Grayov vzor. V našom prípade sme zvolili Grayov vzor. Ak berieme v úvahu bežné rozlíšenie projektoru 1024x768px, ktoré bolo použité aj v našom prípade, môžeme vyčítať nasledujúce hodnoty pre počet štruktúrovaných vzorov podľa [14] :

$$\text{Počet horizontálnych vzorov} \quad N_H = \text{ceil}(\log_2 768) = 10 \quad (1)$$

$$\text{pri ofsete} \quad \text{offset}_H = \text{floor}[(2^{N_H} - 768)/2] = 128px \quad (2)$$

$$\text{Počet vertikálnych vzorov} \quad N_V = \text{ceil}(\log_2 1024) = 10 \quad (3)$$

$$\text{pri ofsete} \quad \text{offset}_H = \text{floor}[(2^{N_V} - 1024)/2] = 0px \quad (4)$$

Výpočtom sme získali pre dané rozlíšenie potrebu 10 vertikálnych a 10 horizontálnych pásov, teda dokopy 20 pásov kódujúcich pozíciu pixelov projektoru. Treba si všimnúť offsetové posunutie vertikálneho vzoru o 128pixelov. Vychádza z predpokladu, že skenovaný objekt sa nachádza v strede obrázkov a okrajové body nás nezaujímajú. Preto je obraz zhora aj zdola vycentrovaním o 128 pixelov. Počas skenovania však z dôvodu nerovnomerného osvetlenia môže prísť k nesprávnemu priradeniu pixelu k čiernemu alebo biemu pásku. Ak by sme spracúvali takto zosnímané objekty, vznikalo by veľké množstvo plôch, ktoré by kvôli kontrastu tvorili chybné plôšky spôsobené dvojznačnosťou. Z tohto dôvodu sa využívajú inverzné vzory, ktoré pre každý vzor vytvoria aj jeho farebnú inverziu. Z oboch obrázkov sa následne porovná kontrast a vyberie správna hodnota. Z dôvodu počiatočného nastavenia hodnoty kontrastu pre zaradenie k správnej rozoznaniu farby pásiku a zároveň pre prekrytie farebnou mapou pri vizualizácii sa pridávajú ešte čisto biela a čisto čierna projekcia. Celkovo tak získame až 42 obrázcov, ktoré potrebujeme premietnuť na objekt.

### 2.1.2.3 Kalibrácia

Pri snímaní obrazu kamerou si niekedy môžeme všimnúť drobné skreslenia obrazu. Asi najtypickejšie skreslenie obrazu, ktoré môžeme pozorovať voľným okom je sudovité skreslenie alebo skreslenie typu rybie oko. Ostatné skreslenia sú pre naše oko väčšinou neviditeľné, ale pre zariadenia s vysokou presnosťou môžu viesť k fatálnym chybám. Zariadenie, ktoré spracúvajú obraz z kamier a je od neho požadovaná vysoká presnosť by preto mali byť skalibrované. Pre získanie informácií o pixelovom skreslení kamery a aj projektoru sa používajú rôzne kalibračné metódy. Kalibrácia sa ale nepoužíva iba na odstránenie geometrických chýb obrazu, ale aj na vzájomnú kalibráciu prístrojov v priestore. Získame tak rovnice, ktoré nám udávajú ich niektoré vzájomné spoločné vlastnosti. V našom prípade sme schopný získať napríklad vlastnosti ako ich vzájomné natočenie, vzájomnú vzdialenosť, vyžarovacie uhly a pre snímanie objektu aj ich vzájomné pretínanie snímacích plôch. Pri aplikáciách ako je 3D skenovanie alebo podobných aplikáciách, kde je kladený vysoký dôraz na presnosť je kalibrácia nutnosťou.



V prípadoch, kde nám na presnosti až tak nezáleží a stačí nám vedieť iba vzájomnú polohu a orientáciu kamier a projektorov, môžeme vynechať zisťovanie skreslenia. Systémová kalibrácia, napríklad pri kombinácii viacerých kamier a projektorov, je väčšinou zložitý a hlavne časovo náročný proces. V procese kalibrácie musíme najprv skalibrovať kameru alebo iné zariadenie, z ktorého získavame obraz. Následne zo znalosti skreslenia snímacieho zariadenia môžeme prejsť ku kalibrácii projektora. Hlavný princíp väčšiny kalibračných programov kalibrujúcich projektor je takzvaný „*pinhole camera model*“ v kombinácii s úvahou, že projektor dokáže „zachytávať“ obraz rovnako ako kamera. Prechádzajúce lúče svetla majú rovnakú deformáciu spôsobené nepresnosťami výroby na šošovke kamery a aj projektora. [15]

**Kalibrácia kamery - Jean-Yves Bouquet Calibration Toolbox** – Prevzaté z [16]: Voľne stiahnuteľný toolbox. Keďže je voľne dostupný existuje na neho veľké množstvo adaptácií pre konkrétne technické použitie užívateľmi, avšak nie všetky sú bez komplexnej znalosti technických riešení a zmien aplikovateľné. Je vhodný hlavne z dôvodu, že dokáže vypočítať ako extrinsické(vonkajšie) tak aj intrinsické(vnútorne) parametre. Tento toolbox pre svoju správnu funkciu vyžaduje *Psychophysics toolbox*, ktorý je rovnako voľne dostupný na internete. Tento toolbox ponúka podobné funkcie ako *Image Aquisition toolbox*.

#### Intrinsické parametre:

Medzi vnútorné parametre patrí:

- Ohnisková vzdialenosť (*focal length*) -  $f_c$  v pixeloch uložená vo vektore 2x1
- Stred obrázka (*principal point*) - priesečník optickej osi a roviny obrázku  $C_c$ , ktorého súradnice sú uložené vo vektore 2x1
- Uhol (*skew factor*) – koeficient  $alpha_c$  určujúci natočenie medzi x a y osou
- Skreslenie (*distortions*) – koeficienty  $k_c$  uložené vo vektore 5x1 určujú radiálne a tangenciálne skreslenie

Ak predpokladáme bod v priestore v referenčnom súradnicovom systéme  $XX_c = [X_c; Y_c; Z_c]$  kamery, premietneme daný bod na plochu obrázku podľa intrinsických parametrov. Ak je  $x_n$  normalizovaná dierková (pinhole) projekcia

$$x_n = \begin{bmatrix} X_c/Z_c \\ Y_c/Z_c \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \quad (5)$$

Ďalej ak  $r^2 = x^2 + y^2$  a po započítaní skreslenia objektívu kamery je nový normalizovaný bodový koordinačný systém  $x_d$  definovaný:

$$x_d = \begin{bmatrix} x_{d1} \\ x_{d2} \end{bmatrix} = (1 + kc_1r^2 + kc_2r^4 + kc_5r^6)x_n + dx \quad (6)$$

Kde je  $dx$  definované ako

$$dx = \begin{bmatrix} 2kc_3xy + kc_4(r^2 + 2x^2) \\ kc_3(r^2 + 2y^2) + 2kc_4xy \end{bmatrix} \quad (7)$$

Takže vektor skreslenia  $kc$  obsahuje 5 prvkov pre radiálne aj tangenciálne skreslenia. Po aplikovaní matice skreslenia, výsledný súradnicový systém  $x_{pixel}$  pre daný bod obrázku je:

$$x_{pixel} = \begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} (x_{d1} + \alpha_c \cdot x_{d2})fc_1 + cc_1 \\ fc_2x_{d2} + cc_2 \end{bmatrix} \quad (8)$$

Prepočet medzi normalizovaným skresleným súradnicovým systémom a súradnicovým systémom  $x_{pixel}$  je možný s použitím lineárnych rovníc:

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \begin{bmatrix} fc_1 & \alpha_c \cdot fc_1 & cc_1 \\ 0 & fc_2 & cc_2 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{d1} \\ x_{d2} \\ 1 \end{bmatrix} \quad (9)$$

$fc_1$  nám označuje ohniskovú vzdialenosť pre horizontálny smer a  $fc_2$  pre vertikálny.

### Extrinsické parametre

Medzi extrinsické parametre patrí:

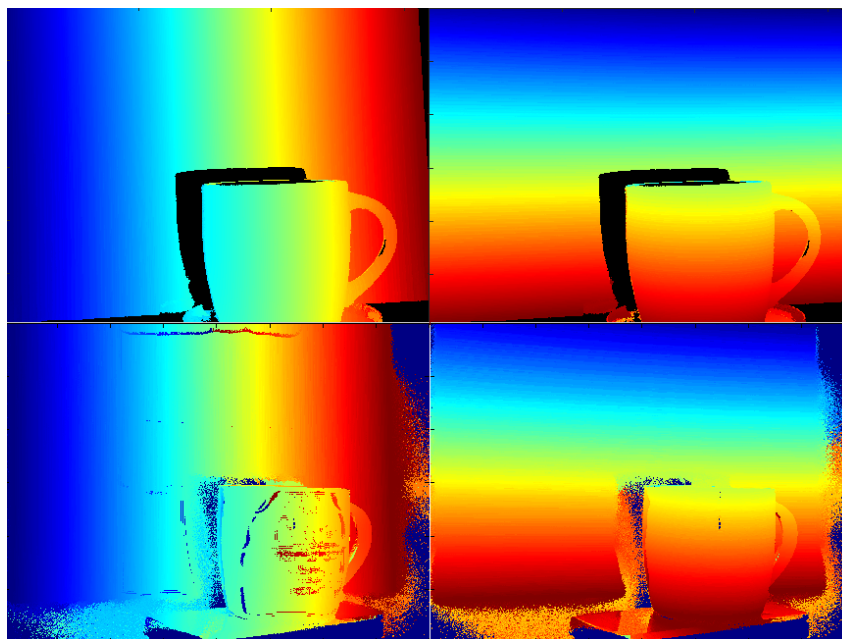
Rotácia- súbor matíc  $n_{ima}$  s rozmermi 3x3

Translácia - súbor vektorov  $n_{ima}$  s rozmermi 3x1

### **Kalibrácia projektoru -ProCam\_Calib** – prevzaté z [17]

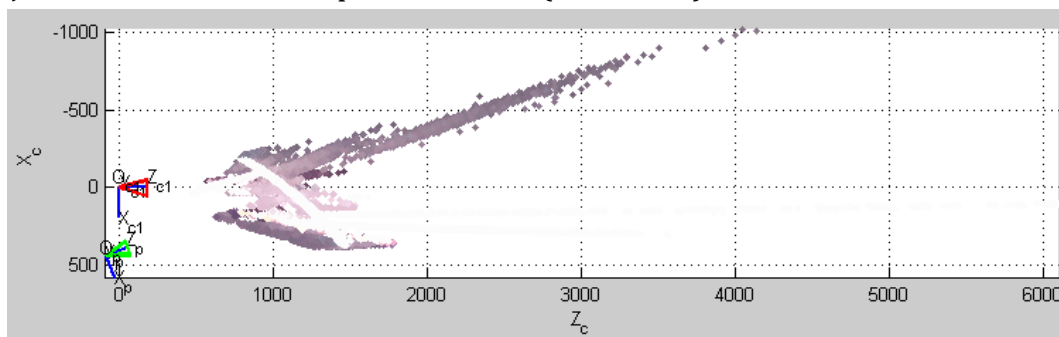
Spracovaný vo výpočtovom prostredí Matlab. Ako svoj základ berie kalibráciu kamery pomocou Zhangovej metódy spracovanej *Jean-Yves Bouguetom* v práci *Calibration Toolbox* v Matlabe. Rozširuje ju o kalibráciu samotného projektora. Pracuje na princípe inverznej kamery. Ako hlavný predpoklad uvažuje fakt, že pre projektor aj kameru platia rovnaké lineárne rovnice, jediný rozdiel je v tom, že projektor premieta namiesto zachytávania obrazu. Vzor šachovnice je premietaný na plochu, ktorú zaznamenávame kamerou, ktorej parametre skreslenia sú známe. Zo zachytených obrázkov premietaného vzoru môžeme následne extrahovať rohy. Algoritmus následne nájde očakávané a reálne polohy rohov jednotlivých štvorcov šachovnice. S použitím metódy najmenších štvorcov sa nájde najvhodnejší súbor rovníc pre popísanie skreslenia projektoru. Výhodou tohto rozšírenia *Jean-Yves Bouguetovho toolboxu* je možnosť zobrazenia spoločnej vzájomnej polohy kamery a projektoru, a aj ich parametrov ako ohniskové vzdialenosti a podobne.

Na nasledujúcom obrázku (Obrázok 8) je viditeľný vplyv rozkalibrovanejosti systému. Na hornej fotke je správne skalibrovaný systém a na spodnej fotke je spracovaný obraz po posunutí kamerou a teda rozkalibrovaní celého systému.



Obrázok 8: Ukážka rozkalibrovania, kamera bola vychýlená o niekoľko stupňov

Rekonštrukcia bodov pri nenakalibrovanom systéme je absolútne scestná. Objekt ani náznakom neodpovedá realite (Obrázok 9).



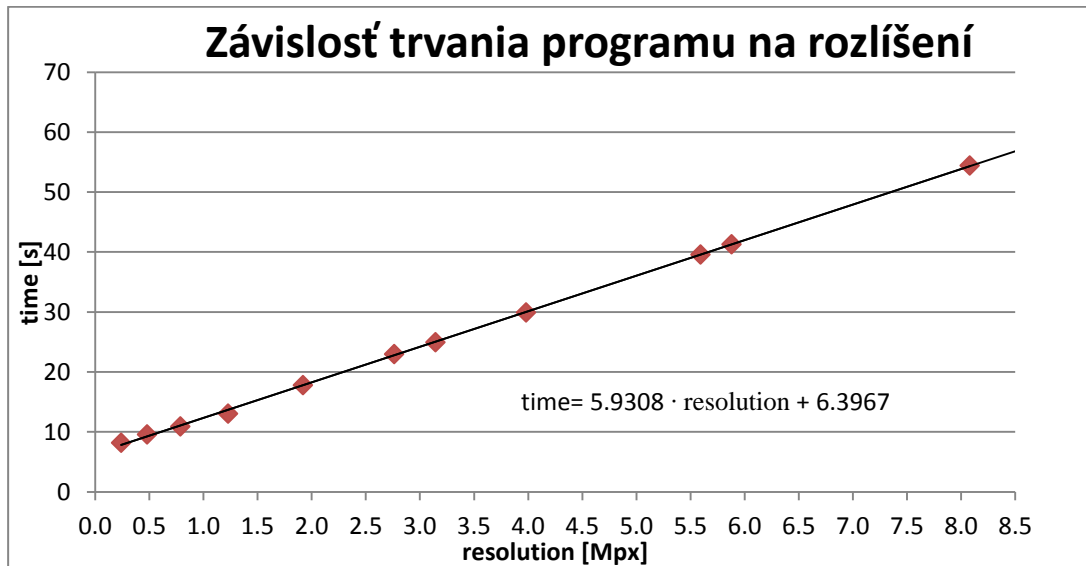
Obrázok 9: Výsledný zachytený 3D model pre rozkalibrovaný merací systém

#### 2.1.2.4 Zachytávanie obrazu

Pre reálny odhad vplyvu rozlíšenia kamery na dobu trvania som napísal jednoduchý program, ktorý zachytí pre dané rozlíšenie celú sekvenciu štruktúrovaných obrázkov pomocou *camera module v2* a uloží ich, bez ich následného vykresľovania. Z vizualizácie výsledkov v grafe je zjavnejší vplyv rozlíšenia na dobu snímania (Obrázok 10).

Tabuľka 4: Závislosť doby zachytenia 100 obrázkov na rozlíšení kamery

width [px]	3280	2800	2732	2304	2048	1920	1600	1280	1024	800	600
height [px]	2464	2100	2048	1728	1536	1440	1200	960	768	600	400
pixels [Mpx]	8.08	5.88	5.60	3.98	3.15	2.76	1.92	1.23	0.79	0.48	0.24
time [s]	54.43	41.29	39.57	29.91	24.96	22.95	17.80	13.06	10.90	9.59	8.17



Obrázok 10: Závislosť trvania programu na rozlíšení

Z výsledkov je vidieť podľa očakávania výrazné predĺženie doby trvania programu. Pre samotný realtime streaming a ovládanie podradeného robota je vhodné použiť menšie rozlíšenie, ktoré nám pre navigáciu a ovládanie robota stačí. Treba brať v ohľad aj skutočnosť, že pri ovládaní cez WiFi pomocou programu *VNC viewer* sa nám prenosová rýchlosť dát tiež výrazne zníži. Pre streaming a ovládanie robota nie je potrebná vysoká kvalita, naopak rýchlosť je výraznou výhodou a preto som zvolil rozlíšenie 800x600 pixelov. Je to výrazne menej ako maximálne rozlíšenie, dosiahnuteľné pomocou *camera module v2* ale pre tento účel dostačujúce.

#### 2.1.2.5 Spracovanie obrazu a jeho rekonštrukcia

##### BYO3D Scanner- prevzaté z [14]

Na spracovanie zosnímaných dát môže byť použitý voľne dostupný program napísaný v Matlabe určený pre študentské účely. Jedná sa o jednoduchý program určený študentom pre ich prvé kroky v oblasti 3D skenovania. Tento program obsahuje zdrojový kód aj pre kalibráciu sústavy projektor a kamera, pre snímanie a zaznamenávanie obrázkov a samotný program na rekonštrukciu obrazu. Samotná kalibračná časť je navrhnutá na autonómne premietanie a snímanie vzoru, takže potrebuje pre svoju funkciu prístup ku kamere. Druhou možnosťou je využitie dát z *ProCamCalib toolboxu*. V našom prípade je jednoduchšie využiť dáta generované z *ProCamCalib Toolboxu*, pretože nepotrebujeme upravovať celý kód, ktorý je formulovaný pre ovládanie webkamier.

*BYO3D Scanner* podporuje spracovanie ako binárneho vzoru, tak aj zrkadlového vzoru v Grayovom kóde. Je rozdelený na dve hlavné časti. V prvej časti zisťuje korešpondenciu bodov riadkov a stĺpcov kamery. Korešpondenciu zobrazuje použitím *jet colormap*. Body ktorým nebolo možné prideliť zhodu riadkov alebo stĺpcov sú vykreslené čiernou farbou. Tieto body neboli osvetľované

projektorom. V druhom kroku sa jednotlivým bodom prideli pozícia v priestore. Toto mračno bodov je prekryté farebnou mapou odpovedajúcou farbou obrázku pre jednotlivé body.

Pre naše účely bude prebraný základný algoritmus zisťovania polohy pixelov projektoru a prepísaný tak, aby bol schopný prepočtu dát na zariadení neobsahujúcom Matlab.

#### **2.1.2.6 Chyby pri skenovaní**

3D skenovanie vyžaduje vysokú mieru presnosti. Aby sme boli schopný získať kvalitný a presný model, musíme eliminovať čo najväčšie množstvo chýb, ktoré môžu vzniknúť. Niektoré chyby sú odstrániteľné priamo, iné je nutné odstrániť až po skenovaní korekciou. Chybové vplyvy pôsobiace na proces skenovania môžeme rozdeliť na vnútorné chyby zariadení, vonkajšie chybové vplyvy a chyby spôsobené spracovaním.

Medzi vnútorné chyby zariadení môžeme zaradiť napríklad nedokonalé zarovnanie vnútorných súčastí fotoaparátu a v našom prípade aj projektoru, nedokonalé vyhladenie a povrchové nepresnosti šošovky, nedokonalosť CCD alebo CMOS súčiastok, prach alebo zašpinenie objektívu. Opakovaním merania sa chyba nezmenší, zostáva konštantná. Tieto chyby sú až na odstránenie zašpinenia vonkajších častí vo veľkej miere neodstrániteľné. Minimalizácia chýb sa vykonáva kalibráciou aj kamery, aj projektoru.

K vonkajším vplyvom môžeme zaradiť napríklad vplyv prostredia na priechod zväzkov lúčov alebo vplyv povrchu skenovaného objektu na meranie. Pre skenovanie 3D objektov sú najviditeľnejšie chyby spôsobené prostredím svetelné podmienky, farba objektu, difúzia lúčov alebo nehomogenita prostredia. Základné merateľné chyby sú prebraté v kapitole 5.

#### **2.1.2.7 Post processing a vizualizácia**

V našom prípade celý proces spracovania dát prebieha na zariadení Raspberry Pi. Po exporte dát na počítač je nutná ich vizualizácia. K tomu využijeme vizualizačnú časť programu z odseku 2.1.2.5. S touto časťou sa detailnejšie zoznámime v časti opisu programu (4.11). *BYO3D Scanner* poskytuje okrem prepočtu aj možnosť vytvorenia priestorových dát s koncovkou *.wrm*. Ukážka vytvorených dát je v prílohe č.7 na obrázku 3.

Optické prístroje dokážu v jednom momente zaznamenávať iba plochu objektu viditeľnú kamerou. Ak chceme však dostať kompletný model zo všetkých strán, musíme buď otáčať s predmetom alebo pohybovať so zariadením. Oba princípy si ale vyžadujú následné spájanie meshov. Pre zobrazenie viacerých meshov a ich úpravu a následné spojenie je použiteľný softvér Meshlab.

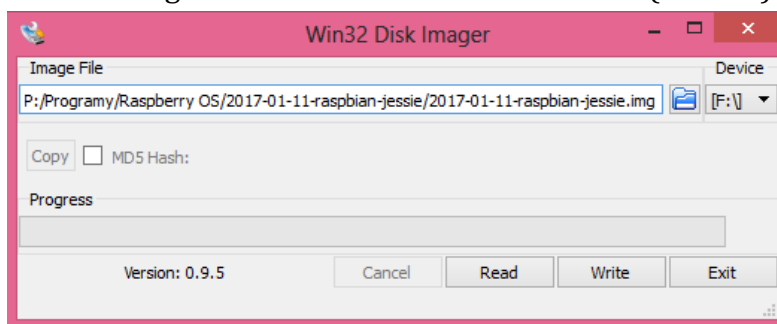
## 3. Základné nastavenie zariadení a ich rozmiestnenie

Ako aj u väčšiny technických procesov, aj tu je nutná príprava. Ku krokom nutným pred samotným začatím skenovania patrí zaobstaranie nutných prístrojov a súčastí, vytvorenie konštrukcie dovoľujúcej prepojenie ich funkcií a v neposlednej rade aj samotná programová predpríprava zariadení. Pre snímanie objektov a testovanie budeme využívať prístroje popísané v druhej kapitole tejto práce.

### 3.1 Prvotné nastavenie Raspberry Pi 3 model B

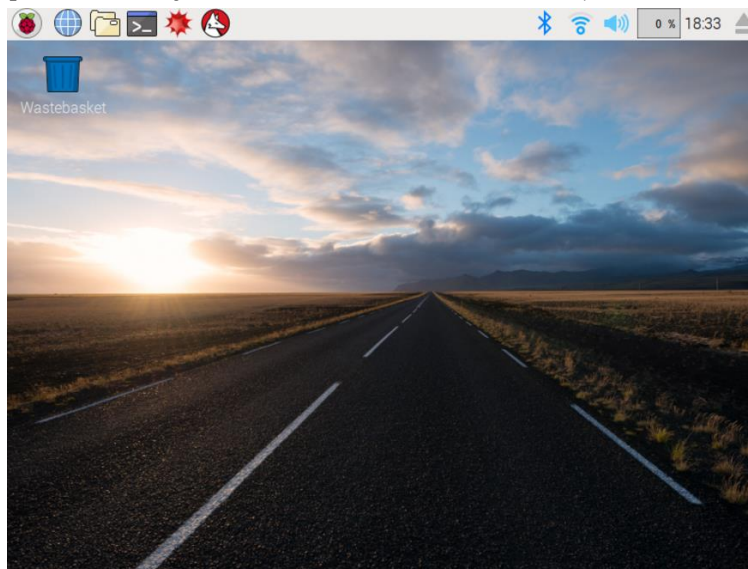
Pred prvotným spustením si zaobstaráme USB myš, USB klávesnicu, Ethernet kábel a dostatočne veľkú micro SD kartu. Vybral som SanDisk 32GB microSDHC Extreme 90MB/s, UHS-I kartu. Táto karta je plne kompatibilná s Raspberry Pi 3 model B. V oficiálnych rýchlostných testoch na OS Raspbian Jessie nainštalovanom na Raspberry Pi 2 dosahovala rýchlosť čítania 22.8MB/s a rýchlosť zápisu 25.2MB/s ([http://elinux.org/RPi\\_SD\\_cards](http://elinux.org/RPi_SD_cards)). Veľkosť karty je dostatočná pre nainštalovanie OS *Raspbian Jessie with Pixel* ktorý zaberá po nainštalovaní niečo cez 4GB pamäte. Zároveň má táto karta dobrú rýchlosť zápisu a čítania a aj dostatočné miesto pre použitie pri priestorovom skenovaní objektov, kde sa predpokladá časté a rýchle ukladanie obrázkov na kartu a ich následné spracovanie. Veľkosť modelov po spracovaní je približne 50 až 80 Mb, Séria obrázkov pre kalibráciu alebo pre spracovanie má veľkosť až 250 Mb. Veľmi rýchlo sa tak zaplní voľné miesto na karte.

Pre prvotné nainštalovanie operačného systému si ako prvé stiahneme OS *Raspbian Jessie with Pixel* z oficiálnej stránky pre Raspberry Pi, súbor v tvare *.zip* rozbalíme a dostaneme súbor s príponou *.img*. Vložíme micro SD kartu s adaptérom do USB čítačky alebo do slotu na notebooku. Použijeme freeware program Win32DiskImager na nahratie obrazu OS na kartu (Obr. 11).



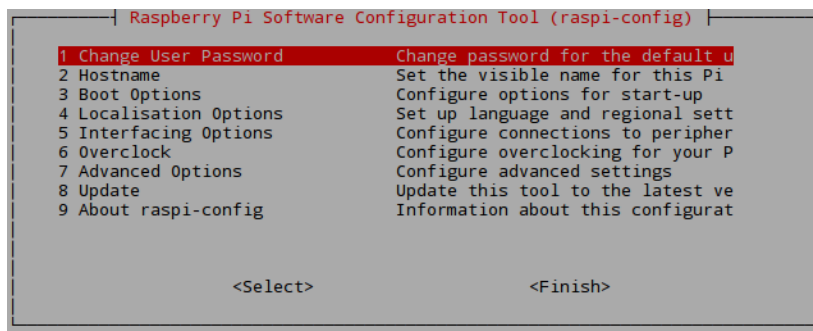
Obrázok 11: Inštalácia OS Debian Jessie na Micro SD kartu

V kolónke *Image File* vyberieme rozbalený obraz operačného systému *Debian Jessie* s koncovkou *.img*, do kolónky *Device* vyberieme našu micro SD kartu. Klikneme na *Write* a prebehne inštalácia. Po úspešnom nainštalovaní vyberieme SD kartu z USB čítačky kariet alebo slotu na PC a vložíme ju do slotu na Raspberry. Keďže SSH prístup je od novej verzie Rasbianu defaultne zakázaný, musíme ho povoliť lokálne. Preto je pre nás jednoduchšie všetko vykonať priamo v Raspberry. Pripojíme USB myš, klávesnicu do voľných USB slotov. Pripojíme HDMI kábel do Raspberry a druhý koniec do Obrazovky podporujúcej HDMI. V mojom prípade som využil projektor a pripojil Raspberry pomocou HDMI káblu priamo na neho. Ak potrebujeme internet, môžeme pripojiť Ethernet kábel alebo využiť zabudovanú 802.11n Wireless LAN. Zapneme postupne projektor a následne pripojíme adaptér 5.1V 2,5A do micro USB slotu na Raspberry. Po pripojení napájania sa Raspberry začať ihneď bootovať, rozoznáme to rozsvietením oranžovej LED a rozblíkaním zelenej LED. Rovnako sa nám ukáže na obrazovke alebo v našom prípade projektore priebeh bootovania. Po chvíľke sa nám zapne hlavné okno operačného systému Debian Jessie. Ukážka je na obrázku 12.



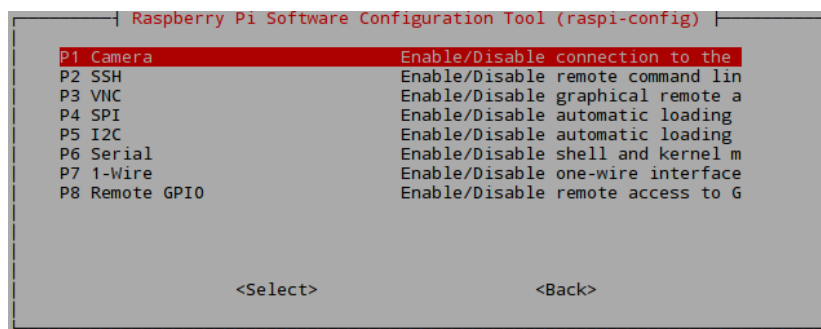
**Obrázok 12: Plocha OS Debian Jessie**

Následne pre budúce použitie je potreba povoliť prístup SSH. Otvoríme príkazový terminál a napíšeme `$ sudo raspi-config`, čím otvoríme konfiguračnú utilitu vzhľadovo odpovedajúcu BIOS-u (Obrázok 13). Touto utilitou môžeme priamo editovať obsah textového súboru `„/boot/config.txt“`, ktorý definuje nastavenia Raspberry počas bootovania.



Obrázok 13: Konfiguračná utilita pre Debian Jessie

Po zápise obrazu na kartu sa využije iba miesto veľkosťou zodpovedajúcou veľkosti inštalácie. Aby sme mohli využiť aj zvyšné miesto na karte, musíme sprístupniť aj zvyšok miesta na karte. Vyberieme možnosť *Advanced Options* a následne klikneme na *Expand Filesystem* a potvrdíme tlačidlom *Select*. Keďže sa prepisuje bootovací súbor, zmeny sa prejavia až po reštarte. Pred reštartom však môžeme povoliť SSH prístup a aj kameru. Vrátime sa teda do hlavného menu konfiguračnej utility a vyberieme možnosť *Interfacing Options*. Zobrazí sa nám nasledujúce okno (Obr.14).



Obrázok 14: Konfiguračná utilita

V tomto okne povolíme možnosť *Camera*, *VNC* pre ovládanie zo vzdialenej plochy, *SSH* pre prístup priamo pomocou Ethernet kábla a pre ovládanie budúceho robota môžeme využiť aj *I2C*. Vrátime sa do hlavného menu a klikneme na *Finish*. Následne pre vykonanie zmien je treba systém rebootovať. Správime to príkazom v terminále `$ sudo reboot`. Po opätovnom zapnutí systému potrebujeme systém updatovať, na to však potrebujeme buď stiahnuté súbory pre update alebo pripojenie na internet. Manuálne sa v OS Raspberry pripojíme na vhodnú Wi-Fi. Otvoríme terminál a napíšeme postupne príkazy:

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

Samotný update a upgrade zaberie nejaký čas. Po skončení upgradu si môžeme otvoriť plochu Debianu pomocou VNC vzdialenej plochy s prístupom z PC. Stačí na to v počítači, z ktorého budeme ovládať RPi, stiahnuť a otvoriť freeware VNC viewer. Po zapnutí programu sa môžeme prihlásiť pomocou IP adresy RPi ktorú zistíme buď príkazom `$ ifconfig` v terminály a následne pod kolónkou



Wlan0 alebo otvorením VNC v Raspberry, kde je táto IP zadaná. Pre pripájanie a budúci export dát je vhodné nastavenie statickej IP pre Raspberry. Nemôže sa nám tak stať že router prideli nášmu zariadeniu inú IP a už sa k nemu nepripojíme inak, ako cez SSH alebo priamo. Musíme teda v teda v terminále nastaviť v konfiguračnom súbore *dhcpd.conf* statickú adresu, a dns [18]:

```
$ sudo nano /etc/dhcp.conf
```

V mojom prípade som doplnil:

```
interface eth0
static ip_address=10.0.1.200/24
static routers=10.0.1.1
static domain_name_servers=8.8.8.8 8.8.4.4 10.0.1.1

interface wlan0
static ip_address=192.168.254.201/24
static routers=192.168.254.1
static domain_name_servers=8.8.8.8 8.8.4.4 10.0.1.1
```

**Obrázok 15: Nastavenie statickej IP adresy**

Následne už môžeme ovládať Raspberry pohodlne vzdialene pomocou počítača. Keď máme spustené VNC, môžeme prejsť k inštalácií jednotlivých súčastí. Pre ovládanie robota je pravdepodobné využitie I2C, využívanie jednotlivých GPIO pinov. Na ich ovládanie musíme nainštalovať knižnicu na ovládanie, napríklad *WiringPi*. Pre inštaláciu môžeme jednotlivé súbory stiahnuť a preniesť pomocou VNC prehliadača, konkrétne možnosti *file transfer* alebo v terminály napísaním:

```
$ sudo apt-get install git-core
```

Aby sme predišli duplicitnej inštalácii, skontrolujeme, či už knižnica nebola nainštalovaná príkazom:

```
$ gpio -v
```

Ak nám niečo vypíše, už máme nejakú verziu nainštalovanú. Ak bola nainštalovaná ako súčasť inštaláčného balíka, musíme odstrániť danú časť balíka.

```
$ sudo apt-get purge wiringpi
$ hash -r
```

Následne môže prejsť k samotnej inštalácii *WiringPi*.

```
$ git clone git://git.drogon.net/wiringPi
$ cd wiringPi
$ ./build
```

Pre skontrolovanie inštalácie využijeme nasledujúce príkazy:

```
$ gpio -v
$ gpio readall
```

Ak nám to vypíše tabuľku s viditeľnými GPIO pinmi a ich *WiringPi* označenie, inštalácia prebehla v poriadku. Následne prejdeme k inštalácii knižníc použiteľných k skenovaniu tvaru objektov.

### **Inštalácia nutného softvéru pre funkčnosť nášho programu [19][20]**

Všetky programové časti okrem samotného prepočtu triangulácie a zisťovania hĺbky objektov sú písané v programovacom jazyku Python verzia 3.6. Hlavná výhoda toho programovacieho jazyku je jeho jednoduchosť na pochopenie a zároveň fakt, že nie je potrebné kód neustále kompilovať ale stačí program spustiť v IDLE. Nainštalujeme aj balík pre obrazové spracovanie *PIL* pre prácu s

obrázkami a *Tkinter* pre grafický interface. *Tkinter* je vo väčšine inštalácií automaticky nainštalovaný spolu so základným balíkom pre Python. Na prácu s maticami môžeme využiť modul *numpy*, avšak hlavné maticové prepočty budú prebiehať v *OCTAVE*.

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt-get install build-essential cmake pkg-config
$ sudo apt-get install python3-dev
$ sudo apt-get install python3-pip
$ sudo apt-get install python-tk idle python-imaging
$ sudo pip3 install numpy
```

V našom prípade pre ovládanie obrazu využívam modul *Pygame*. Okrem neho nainštalujeme aj knižnice pre spracovanie a prípadné streamovanie videa a obrazu. Nasledujúcim spôsobom nainštalujeme všetky potrebné balíky (pozn.: balíky je možné reťaziť ):

```
$ sudo pip install pygame
$ sudo apt-get install NAZOV_BALIKU
Spracovanie obrazu: libjpeg-dev, libtiff5-dev, libjasper-dev, libpng12-dev,
libavcodec-dev, libavformat-dev, libswscale-dev, libv4l-dev, libxvidcore-dev,
libx264-dev, libatlas-base-dev, gfortran,
Streaming: libav-tools, libjpeg8-dev, libv4l-dev, imagemagick
```

Pri spracovaní obrazu je predpoklad využitia knižníc *OpenCV*. Vytvoríme zložku *repo* a z *gitu* naklonujeme *OpenCV* súčasti. Následne môžeme *OpenCV* skompilovať.

```
$ mkdir repo
$ cd repo
$ git clone https://github.com/Itseez/opencv.git
$ git clone https://github.com/Itseez/opencv_contrib.git
$ cd opencv
$ mkdir build
$ cd build
$ cmake -D CMAKE_BUILD_TYPE=RELEASE \
  -D CMAKE_INSTALL_PREFIX=/usr/local \
  -D INSTALL_PYTHON_EXAMPLES=ON \
  -D OPENCV_EXTRA_MODULES_PATH=/home/pi/repo/opencv_contrib/modules \
  -D PYTHON_INCLUDE_DIR=/usr/include/python3.4/ \
  -D PYTHON3_NUMPY_INCLUDE_DIRS=/usr/local/lib/python3.4/dist-
packages/numpy/core/include/ \
  -D BUILD_TESTS=OFF \
  -D BUILD_PERF_TESTS=OFF \
  -D BUILD_EXAMPLES=ON ..
```

Pre kompiláciu použijeme príkaz

```
$ make -j2
```

Príkazom *-j* špecifikujeme, koľko jadier chceme pri kompilácii využiť. Raspberry Pi 3 model B má síce quad-core ARMv8 CPU, ale pri kompilácii so 4 jadrami môže prísť ku výraznému zahriatiu a následnému spomaleniu až zastaveniu kompilácie. Ak tento prípad nastane, je nutné zmenšiť počet jadier. Proces kompilácie bude

síce trvať dlhšie, ale nestane sa nám, že by sa proces zastavil. Po úspešnom skompilovaní môžeme skompilované *OpenCV* nainštalovať.

```
$ sudo make install
```

### 3.1.1 Nastavenie automatického prenosu súborov na počítač

Pre prenos dát cez internet medzi skenerom a počítačmi nemusí byť vždy vhodné. Niekedy sa vyžaduje preniesť dáta a modely napríklad pre 3D tlač do prístrojov, ktoré nemusia mať pripojenie k sieti. V takýchto prípadoch je najvhodnejšie dáta preniesť pomocou vymeniteľných diskov. Raspberry Pi má SD kartu. tá však nie je vhodná na takéto prenosy dát. Okrem toho, že by sme museli Raspberry pi vždy vypínať, ak by sme chceli preniesť dáta, rovnako by mohlo prísť k poškodeniu pamäťovej karty. Samotná manipulácia s SD kartou je tiež nepraktická a pri jej výbere môže prísť k poškodeniu ostatných častí na Raspberry Pi. Jednoduchšie riešenie je použiť USB disky. Ich cena je relatívne malá a preto aj pri strate nie je problém ich rýchlo nahradiť. Odpadá tu nutnosť vypínať zariadenie ako by to bolo pri SD karte. Dokážeme takto ušetriť veľké množstvo času a aj samotný prenos sa stáva výrazne pohodlnejším. Ak budeme schopný zároveň nastaviť tieto USB disky ako zdieľané cez sieť, môžeme uložené dáta z Raspberry vyberať priamo, aj bez nutnosti používania VNC pripojenia a jeho vstavanej funkcie File Transfer. Týmto spôsobom môžeme ovládať skener z jedného počítača a z ostatných počítačov prípadne sťahovať potrebné dáta.

Samotné Raspberry pri pripojení USB zariadenie rozpozná, ale nedá nám k nemu prístup. Musíme ho teda pred použitím ešte *mountnúť*. Ako prvé musíme nájsť meno pridelené systémom danému USB. Najprv sa v terminály navigujeme do zložky `/dev /`. V Linuxe sú všetky zariadenia reprezentované súbormi, vrátane pamäťových diskov ako SD karta alebo naše USB. Ak sa nachádzame v zložke tak si vypíšeme všetky pripojené disky.

```
$cd /dev
$sudo -fdisk -l
```

V mojom prípade sú pripojené až dva USB disky, ktoré sú označené ako `sda1` a `sdb1`. Následne sa presunieme do zložky *media* a vytvoríme tu novú zložku s názvom *usb* a *usb2*.

```
/media $sudo mkdir usb
/media $sudo mkdir usb2
```

Pokračujeme v *mountnutí* oboch zariadení.

```
/media $sudo mount -t vfat -o uid=pi,gid=pi /dev/sda1 /media/usb
/media $sudo mount -t vfat -o uid=pi,gid=pi /dev/sdb1 /media/usb2
```

Ak budeme potrebovať *unmountnúť* USB, pokračujeme príkazmy:

```
/media $sudo umount /media/usb
/media $sudo umount /media/usb2
```

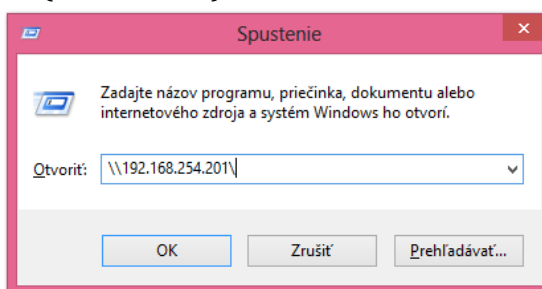
Pre nastavenie zdieľania potrebujeme stiahnuť na Raspberry open source softvér *Samba*.

```
/ $sudo apt-get install samba
```

Následne upravíme súbor z konfiguračnými nastaveniami pre sambu podľa nášho priania v ceste `/etc/samba/smb.conf`. V mojom prípade som nastavil ako zdieľané iba jedno UBS, druhé zostane iba pre manuálne prenosy. Heslo zostalo zatiaľ nenastavené, takže k prístupu stačí vedieť iba IP adresu. Predstavuje to riziko, ale v našom prípade sme si istí, že nespôsobíme chybu alebo cieleňé poškodenie. Riešiť to môžeme zmenením prístupových práv iba na čítanie alebo nastavením hesla. Po úprave reštartujeme Sambu príkazom:

```
/ $ sudo /etc/init.d/samba restart
```

V počítači môžeme prezerat' obsah USB po stlačení klávesovej kombinácie Ctrl+R a napísaním IP adresy nášho USB, alebo v počítači ho manuálne nájsť pod sieťovými pripojeniami (Obrázok 16).



Obrázok 16: Prístup k zdieľaným médiám

Obdobne môžeme kliknúť na ploche PC pravým tlačidlom, vybrať možnosť „Nový“ a „Odkaz“. Zadáme adresu zdieľanej zložky a potvrdíme.

### 3.1.2 Inštalácia Octave [21]

Na samotné spracovanie obrazových dát je možné využiť metódy prezentované v časti 2.1.2. Program obsahujúci princípy geometrickej triangulácie a stereovízie vytvorený na Brownovej univerzite v USA využíva sekvenciu kódov, napísanú v skriptovacom programovacom jazyku Matlab. Samotná inštalácia Matlab-u na Raspberry Pi 3 je možná, ale objemovo veľmi náročná. Preto využijeme voľne stiahnuteľný program OCTAVE. S týmto programom sa pracuje veľmi podobne ako v Matlab-e. Jeho výhoda oproti písaniu v inom programovacom jazyku je jeho defaultná možnosť počítania s maticami, 2D a 3D zobrazenie a rovnako aj práca s obrazovými dátami. Inštaláciu OCTAVE vykonáme príkazmi:

```
$ sudo apt-get update  
$ sudo apt-get install octave
```

Po nainštalovaní máme k dispozícii iba základnú kostru programu. Preto musíme doinštalovať využiteľné balíky. Pre naše účely budeme využívať hlavne balík *image*. Príkazom

```
$ apt-cache search octave-
```

zobrazíme všetky dostupné balíky. Budeme využívať balíky *image*, *common*, *control*, *fpl*, *plot*, *plotplot*. Nainštalujeme ich postupne obdobne príkazom

```
$ sudo apt-get install octave-control
```

### 3.1.3 Pripojenie a nastavenie camera v2 modulu

Kamerový výstup sme už v predchádzajúcej časti povolili, rovnako máme nainštalované *pygame*. Vypneme zariadenie a pripojíme kameru pomocou flex káblu. Zapneme Raspberry a spustíme program prezentovaný v štvrtej kapitole. V hlavnom menu vyberieme možnosť *Test camera*. V príkazovom riadku nám vypíše, či našlo pripojenú kameru a jej umiestnenie. Zároveň sa nám otvorí okno s kamerou, kde skontrolujeme správne otočenie obrazu. V prípade nutnosti otočenia alebo zrkadlenia obrazu stačí v kóde odkomentovať nasledujúce riadky.

```
#pygame.transform.flip( Surface, horizontalne_otoc, vertikalne_otoc )  
#pygame.transform.rotate( uhol )
```

Poslednú vec, ktorú musíme s kamerou spraviť, je nutnosť nastaviť automatické načítanie modulu *bcm2835-v4l2* pri bootovaní Raspberry. Ak tento krok vynecháme, pri nasledujúcom vypnutí a zapnutí Raspberry zistíme, že nebudeme schopní znovu otvoriť kernelový modul pre kameru až do jej opätovného načítania pomocou príkazu

```
$ sudo modprobe bcm2835-v4l2
```

Potrebuje do súboru */etc/modules*, ktorý obsahuje kernelové moduly načítané počas bootu systému, dopísať modul pre kameru. V príkazovom riadku postupne vypíšeme:

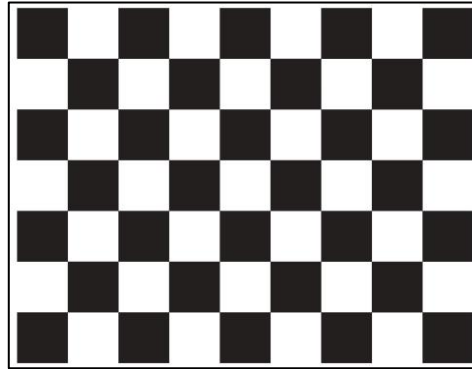
```
$ sudo nano /etc/modules
```

Do otvoreného súboru dopíšeme nasledujúci riadok, uložíme a zavrieme súbor.

```
bcm2835-v4l2
```

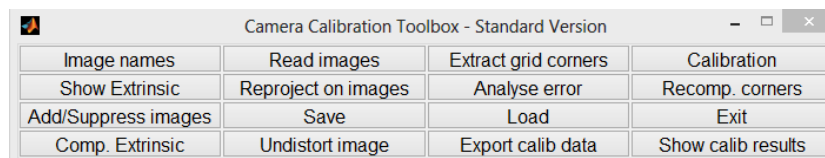
### 3.1.4 Kalibrácia kamery a projektoru

Ako prvé musíme **skalibrovať snímacie zariadenie *camera module v2***. Keďže vysoká presnosť snímaných dát sa zakladá na presnej kalibrácii, snažíme sa využiť čo najvhodnejšiu kalibračnú metódu. V tomto prípade využijeme kalibračnú metódu vyvinutú *Jean-Yves Bouquet*-om, keďže sa nám táto metóda kalibrácie osvedčila už v predchádzajúcej práci. Pre kalibráciu potrebujeme nasnímať aspoň 20 zachytených obrázkov z kamery, na ktorých je jasne vidieť kalibračný vzor z rôznych uhlov a v rôznych vzdialenostiach a následne ich na počítači spracujeme pomocou špeciálne vyvinutého toolboxu. *Jean-Yves Bouquet Calibration Toolbox* využíva šachovnicový vzor s veľkosťou jednotlivých štvorcíkov o veľkosti 30x30mm (Obrázok 17).



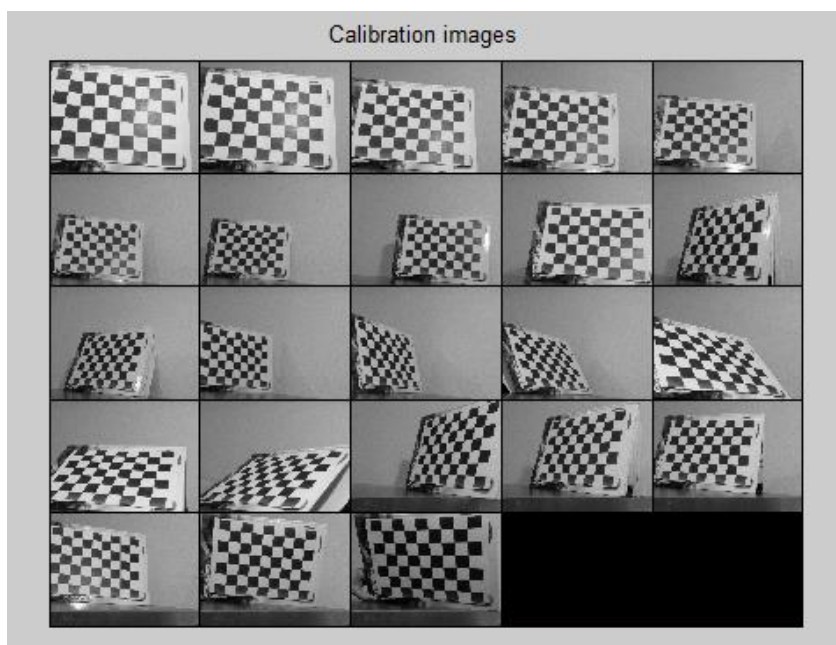
Obrázok 17: Kalibračný vzor pre Jean-Yves Bouguet toolbox

V zaznamenávajúcej aplikačnej časti *data gathering* (kapitola 4.5) s ktorým sa zoznámime neskôr v texte je z konštrukčne-pozíčných dôvodov zaznamenávaný výsek časti plného rozlíšenia kamery. Plné rozlíšenie kamery je 3280x2464 pixelov a my zaznamenávame výsek s rozlíšením 1600x1200 pixelov. Výsledný formát kalibračných obrázkov je TIF(8 bit), vychádzajúci z preferencie programu. Sú použiteľné aj iné formáty, tie však neposkytujú požadované výsledky alebo vedú k kritickým chybám programu. V časti *Get camera calibration data* (kapitola 4.2) zaznamenáme aspoň 20 obrázkov a následne ich presunieme do zložky s kalibračným programom na počítači. Program je spustiteľný v Matlabe po otvorení skriptu *calib\_gui.m*. Po spustení vyskočí okno, v ktorom sú možnosti pre štandardné otvorenie a otvorenie v pamäťovo úspornom režime, pri ktorom je v pamäti v každom momente načítaný maximálne jeden obrázok. Klikneme na možnosť Standard a otvorí sa hlavné menu (Obrázok 18).



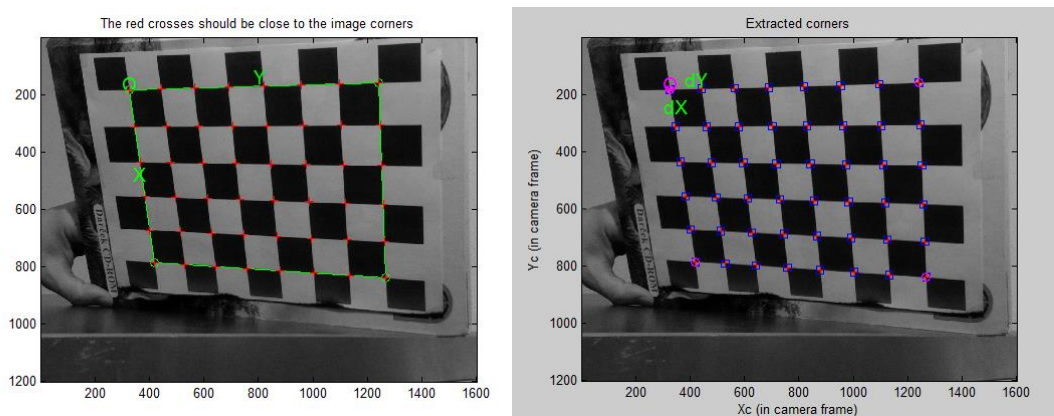
Obrázok 18: Hlavné menu kalibračného nástroja Jean-Yves Bouguet calibration Toolbox

Ako prvé musíme načítať v menu naše zaznamenané kalibračné obrázky. Klikneme na možnosť *Image names* a do príkazového riadku postupne napíšeme základ názvu obrázkov a ich koncovku. Po načítaní všetkých obrázkov, ktoré skript automaticky nájde, sa nám otvorí okno s ukážkou všetkých obrázkov (Obrázok 19).



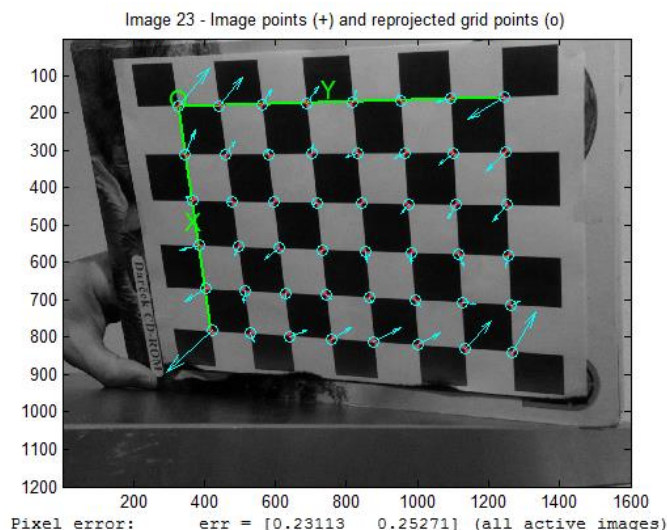
**Obrázok 19: Ukážka nasnímaných obrázkov v Jean-Yves Bouguet calibration Toolbox**

Následne v menu vyberieme možnosť *Extract grid corners*. V príkazovom riadku dva krát „odenterujeme“ defaultné nastavenie veľkosti okna, v ktorom sa budú obrázky zobrazovať. Jednotlivé obrázky sa nám budú postupne otvárať. V týchto obrázkoch označíme vonkajšie rohy šachovnice, poradie výberu rohov musí zostať pre všetky obrázky rovnaké. Po výbere všetkých 4 rohov šachovnice musíme zadať veľkosť rozmerov štvorcov, v našom prípade 30 x 30mm. Zobrazí sa nám okno s odhadovanými rohmi všetkých štvorcov a reálne nájdenými z kamery (Obrázok 20).



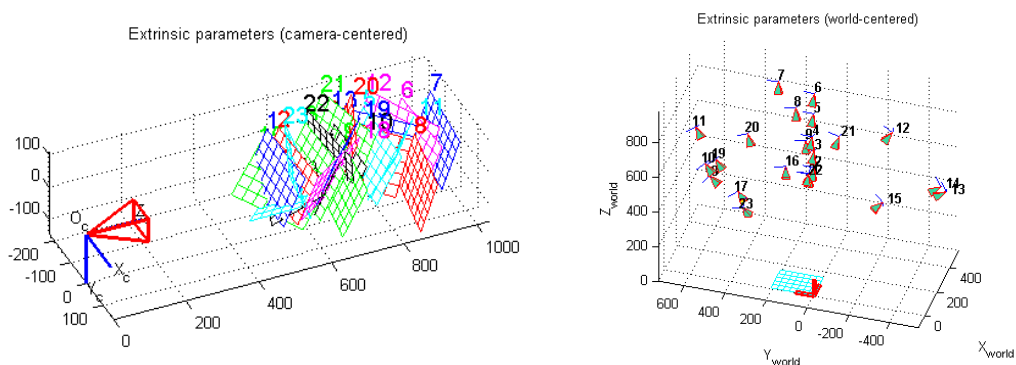
**Obrázok 20: Odhadované pozície rohov štvorcov šachovnice a skutočné nájdené rohy štvorcov šachovnice**

Pre každý obrázok máme možnosť zobrazit' nájdenú pixelovú odchýlku, napríklad pre obrázok 23 získame priemernú pixelovú chybu  $err = [0.23113 \ 0.25271]$  pre súradnice X a Y (Obrázok 21).



**Obrázok 21: Zobrazenie pixelových chýb jednotlivých kalibračných obrázkov**

Možnosť v menu *Show Extrinsic* nám zobrazí polohu všetkých zosnímaných obrázkov vzhľadom na polohu kamery (Obrázok 22). V tejto práci budeme skenovať objekty vo vzdialenosti 60 až 100 cm od projektoru, keďže 60 cm je práve najmenšia možná vzdialenosť, na ktorú sme schopný projektor podľa dokumentácie zaostriť.



**Obrázok 22: Projekcia extrinsických parametrov kalibrácie**

Možnosť v menu *Calibration* nám vypíše základné získané kalibračné údaje pre našu kameru *camera module v2* (Obrázok 23) pred a po optimalizácii.

```

Calibration parameters after initialization:

Focal Length:      fc = [ 2554.79639  2554.79639 ]
Principal point:   cc = [ 799.50000  599.50000 ]
Skew:              alpha_c = [ 0.00000 ] => angle of pixel = 90.00000 degrees
Distortion:        kc = [ 0.00000  0.00000  0.00000  0.00000  0.00000 ]

Calibration results after optimization (with uncertainties):

Focal Length:      fc = [ 2590.77767  2591.57276 ] +/- [ 4.64921  4.64536 ]
Principal point:   cc = [ 795.11403  611.91357 ] +/- [ 8.86296  7.11154 ]
Skew:              alpha_c = [ 0.00000 ] +/- [ 0.00000 ] => angle of pixel axes = 90.00000 +/- 0.00000
Distortion:        kc = [ 0.11291  0.19735  0.00316  0.00326  0.00000 ] +/- [ 0.01256  0.13824 ]
Pixel error:       err = [ 0.23113  0.25271 ]

```

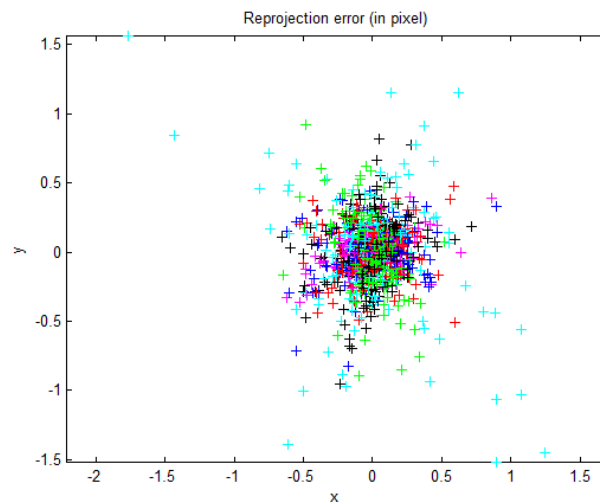
**Obrázok 23: Výpis základných získaných kalibračných dát a pixelových chýb**



Vysvetlivky:

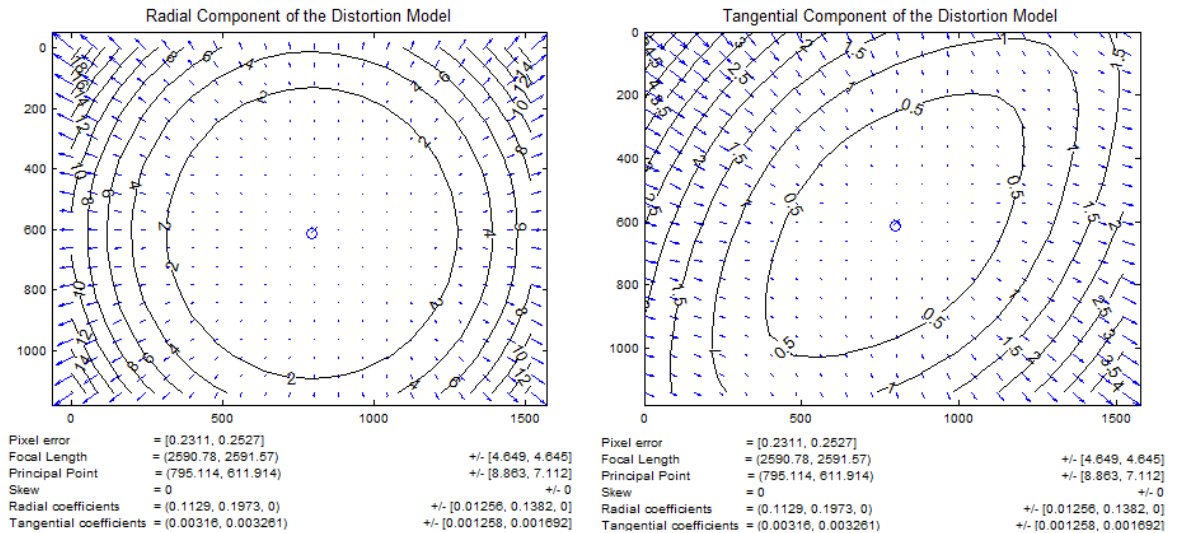
- Focal Length* - Ohnisková vzdialenosť vyjadrená v pixeloch
- Principal point* - Súradnice stredového bodu
- Skew coefficient* - Uhol natočenia medzi osami X a Y uložené v premennej  $\alpha_c$
- Distortion* - Koeficienty skreslenia pre radiálne a tangenciálne skreslenie
- Pixel error* - Maximálna chyba posunutia pixelov pre X a Y

Uhol natočenia medzi osami X a Y je  $90^\circ$ , preto je parameter *Skew coefficient* nulový. Koeficienty skreslenia pre bežné aplikácie stačia do tretieho až štvrtého rádu, v prípade veľmi náročných aplikáciách sa dá zmeniť získavanie na získanie až piatich koeficientov. Ak porovnáme hodnoty skreslenia kamerou *camera module v2*(3280x2464px) a *NIKON D3100* (rozlíšenie 4608x3072px), ktorý bol použitý v predchádzajúcej práci, môžeme pozorovať že hodnoty *Pixel error* vychádzajú pre *camera module v2* výrazne nižšie pre osu X aj Y. Pre *NIKON D3100* sme zaznamenali 0,61805 pre osu X a pre osu Y 0.54877 pixelovú chybu, to sú skoro trojnásobné hodnoty. Reprojekciu pixelových chýb zobrazíme voľbou *Analyse error*. Každý obrázok na obrázku je odlišný inou farbou. Pri nulovej pixelovej odchýlke sa nachádzajú pixely v strede grafu(Obrázok 24). Rovnako tu môžeme pozorovať užšiu distribúciu chyby ako v predchádzajúcej semestrálnej práci.



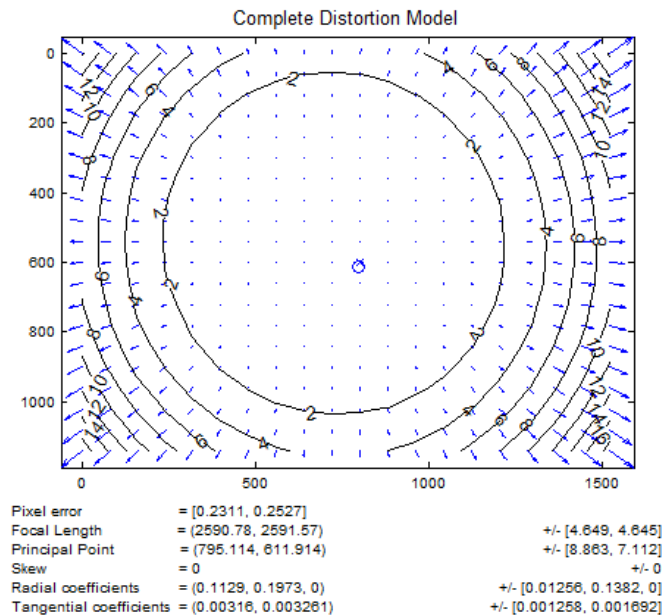
**Obrázok 24: Zobrazenie pixelových chýb jednotlivých kalibračných snímokou**

Po spustení skriptu *visualize\_distortions.m* sa nám zobrazia tri okná s modelom skreslenia pre tangenciálnu, radiálnu zložku a kombináciu oboch.



**Obrázok 25: Vizualizácia radiálneho a tangenciálneho modelu skreslenia**

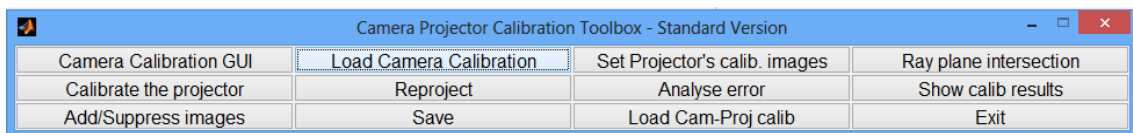
Z porovnania zo semestrálneho prácou, môžeme vidieť že tangenciálna zložka má pre *camera module v2* maximálnu pixelovú chybu po krajných oblastiach okolo 4 pixelov. *NIKON D3100* mal v maximálnych hodnotách niečo cez 30. Ak však porovnáme radiálnu zložku, tak v predchádzajúcej práci dosahovala maxima pre rohové oblasti cca 12 pixelov a pre *camera module v2* cca 14 pixelov, čo predstavuje drobné zvýšenie chyby pre radiálny model skreslenia. V obrázkoch môžeme pozorovať len mierny odklon stredového bodu od súradnicového stredy, na rozdiel od predchádzajúcej práce. Očakávaná hodnota tangenciálnej zložky býva väčšinou o rád nižšia ako pre radiálnu zložku a teda často zanedbateľná pre aplikácie s bežnou presnosťou. Tangenciálna zložka je spôsobená nedokonalosťou pri zarovnaní fyzických častí objektívu. *Camera module v2* teda vykazuje relatívne presné zarovnanie vnútorných elementov. Radiálna zložka je spôsobená hlavne symetriou tvaru fotografických objektívov pri výrobe alebo opracovaní. Najčastejšie skreslenie spôsobené radiálnou zložkou je tzv. *fish eye*. Z kombinovaného modelu skreslenia je zjavná prevaha radiálnej zložky nad tangenciálnou, ktorá sa tu skoro neuplatňuje.



Obrázok 26: Kombinácia radiálneho a tangenciálneho modelu skreslenia

V hlavnom menu vyberieme možnosť *Save* a môžeme prejsť ku kalibrácii projektoru, ako prvé však musíme zachytiť kalibračný vzor premietaný z projektoru a zároveň vytlačený vzor z rovinatej dosky. V hlavnom menu programu, popísanom v kapitole 4 vyberieme možnosť s názvom *Get projector calibration data*. Začne sa premietajú šachovnicový vzor, po stlačení klávesy SPACEBAR sa uloží obrázok do pamäte. Šachovnicový vzor musí byť na rovinatej doske spolu s premietaným šachovnicovým vzorom. Je možné ho exportovať do počítaču pomocou funkcie *export* v hlavnom menu.

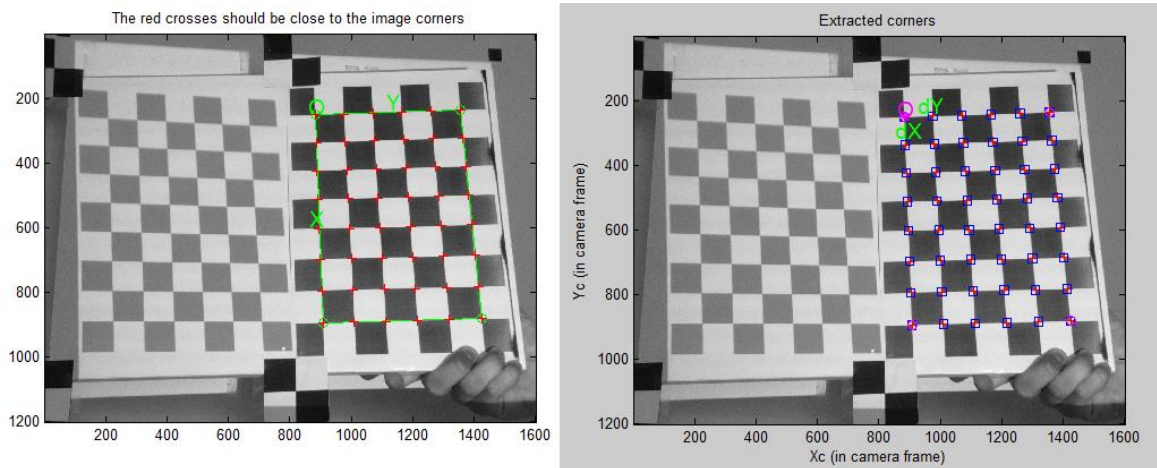
Zvolená **kalibračná metóda projektoru** bola vyvinutá ako nadstavba *Jean-Yves Bouguet Calibration Toolboxu* použitého pre kalibráciu kamery. Pre spustenie programu stačí zložku skopírovať do zložky s *Jean-Yves Bouguet Calibration Toolboxom* a spustiť v Matlab-e skript *cam\_proj\_gui.m*. Po spustení sa otvorí okno s hlavným menu, ukážka je na obrázku č. 27.



Obrázok 27: Hlavné menu kalibračného softvéru pre projektor

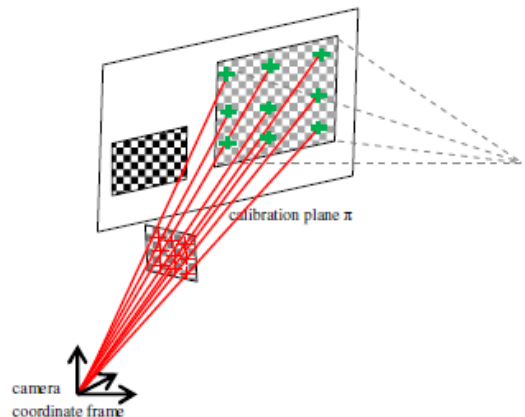
Keďže kalibrácia kamery prebehla v predchádzajúcom kroku, preto pre nahratie kalibračných dát zvolíme možnosť *Load Camera Calibration*. Zvolíme možnosť *Set Projector's calib. images*. Matlab sa nás spýta, či chceme použiť rovnaké snímky ako boli použité pri kalibrácii kamery. Keďže pri kalibrácii kamery nebol zároveň zaznamenaný premietaný vzor projektoru, napíšeme akýkoľvek iný znak do konzoly pre výber iných obrázkov a do konzoly napíšeme

začiatkový názov a koncovku obrázkov. Následne postupne nájdeme všetky rohy vytlačeného šachovnicového vzoru počínajúc počiatkom v ľavom hornom rohu.



**Obrázok 28: Výber rohov vytlačeného vzoru pre kalibráciu projektora**

Tento krok je potrebný pre zistenie relácie medzi reálnym priestorom a premietaným vzorom. Výber prebieha totožne s výberom rohov pri kalibrácii kamery. Ak máme vybraté všetky rohy vo všetkých obrázkoch, môžeme prejsť k označeniu rohov samotného premietaného vzoru. Spustíme *Ray plane intersection* z hlavného menu. Na obrázku 29 môžeme vidieť princíp lúčovo plošnej intersekcie.

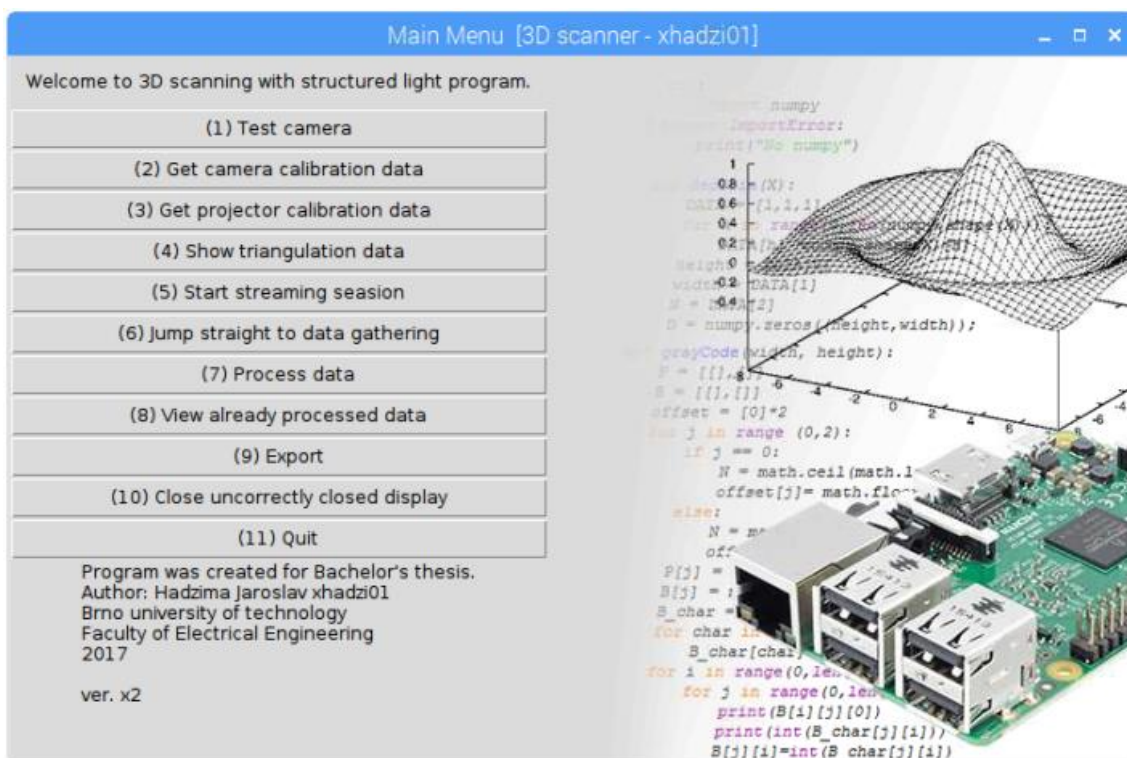


**Obrázok 29: Ukážka intersekcie projektora a kamery**

Otvoria sa nám postupne tie isté snímky, v ktorých sme hľadali rohy šachovnice, avšak vyberáme rohy premietaného vzoru. Výber rohov prebieha totožne s výberom rohov pre vytlačený vzor. Zachytenie vytlačeného aj premietaného vzoru vychádza z potreby znalosti plošnej referencie. Následne napíšeme do konzoly názov kalibračného vzoru a vyberieme v ňom počet a rozmery jednotlivých štvorcíkov. V našom prípade sme použili už zmienený šachovnicový vzor, s veľkosťou štvorcíkov 30x30mm pri tlači na A4. Program si z veľkosti a znalosti rozlíšenia projektora dopočíta premietacie hodnoty veľkosti štvorcíkov. V prílohe č.6 je odfotený proces kalibrácie projektora. Kalibračné údaje sú uložené do matíc *calib\_data.mat* a *Calib\_Results.mat*

## 4. Popis programu pre 3D skenovanie

Pre danú problematiku skenovania trojrozmerného charakteru objektov s využitím štruktúrovaného osvetlenia som vytvoril program písaný v programovacom jazyku Python (verzia 3.6). Tento program dovoľuje ako aj samotné zaznamenávanie dát pre 3D skenovanie, tak aj napríklad overenie funkčnosti fotoaparátu, ukážku triangulačných dát alebo export. Samotný program je schopný vypínať a zapínať premietanie podľa aktuálnej potreby pre skenovanie. Pre výpočet triangulácie využíva kalibračné dáta získané pomocou *Jean-Yves Bouquet Calibration Toolbox* a pre výpočet hĺbky využíva open source softvér vytvorený na Brownovej univerzite. V nasledujúcej časti sa budem venovať jednotlivým častiam programu. Vzhľad hlavného menu je viditeľný na obrázku 30.

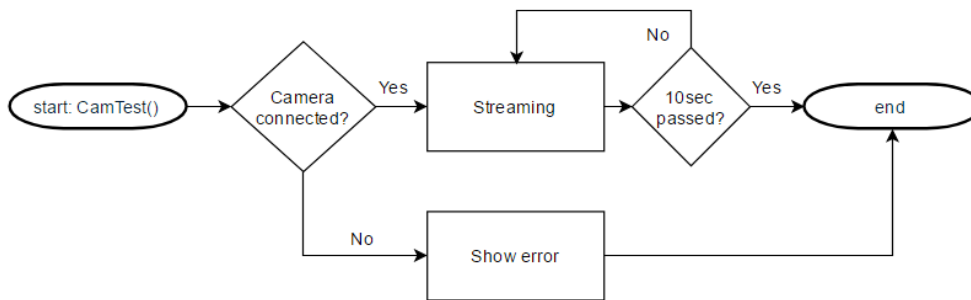


Obrázok 30: Hlavné menu programu pre 3D skenovanie

### 4.1 Test camera

Po pripojení kamery je vhodné vyskúšať jej funkčnosť. Táto možnosť po rozkliknutí zistí, či je pripojená a detekovateľná kamera, ak nie je vypíše upozornenie. Ak je kamera funkčná, otvorí okno s ukážkou obrazu zaznamenávaného kamerou. Toto okno je možné vypnúť kombináciou tlačidiel Ctrl+C alebo počkaním 10 sekúnd, po ktorých sa okno samo vypne. Tento test slúži aj na detekciu nutnosti prípadného otočenia alebo zrkadlenia obrazu.

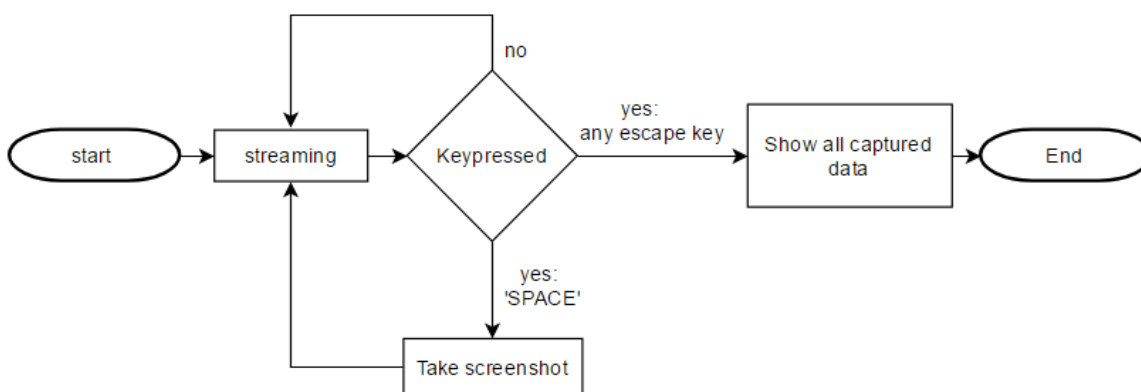




Obrázok 31: Flowchart pre možnosť Test camera

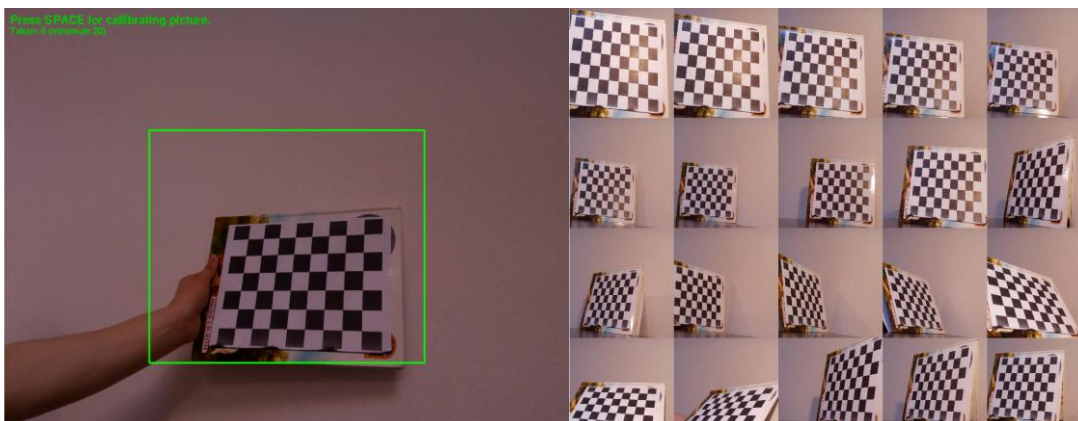
## 4.2 Get camera/projector calibration data

Pri kalibrácii kamery sa po spustení otvorí okno s náhľadom na pohľad kamery. Rozlíšenie pre celé viditeľné okno je rovnaké ako v priebehu *data gathering* a teda maximálne možné rozlíšenie 3280x2464px. V prípade kalibrácie projektoru sa zobrazuje iba premietaný šachovnicový vzor. Flowchart je pre oba prípady podobný, rozdiel je iba v ukážke viditeľnej užívateľom na ploche.



Obrázok 32: Flow chart pre možnosť Get camera/projector calibration data

Pri kalibrácii kamery sa v strede obrazovky nachádza obdĺžnik vymedzujúci rozlíšenie vnútorného okna 1600x1200px. Snažíme sa teda simulovať rovnaké podmienky ako pri možnosti *data gathering* (Obrázok 33). V ľavom rohu obrazovky sa nám zobrazuje aktuálny počet zaznamenaných obrázkov, ktoré zaznamenávame stlačením SPACEBAR-u .

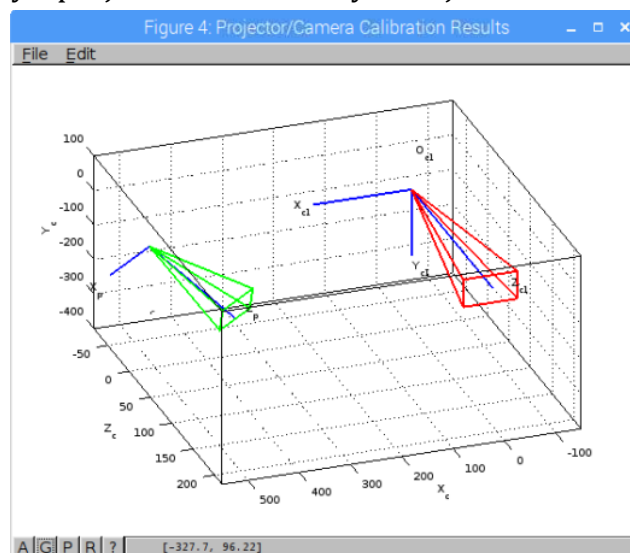


Obrázok 33: Ukážka kalibrácie kamery

Pre dostatočnú kvalitu kalibrácie zaznamenáme aspoň 20 obrázkov a zaznamenávanie vypneme jedným z escape príkazov (Esc, Ctrl+C , tlačidlo X) a zobrazí sa nám náhľad na zachytené obrázky, kde môžeme skontrolovať, či sú obrázky vhodné alebo sa na nich prejavil nejaký typ chyby ako je rozmazanie alebo umiestnenie vzoru mimo zachytávanú oblasť. Pri kalibrácii sa musia všetky štvorčky šachovnice, okrem vonkajšieho radu nachádzať v obrázku. Ak sme spokojní so zachytenými obrázkami, tak ich môžeme exportovať do počítača a následne získať kalibračné dáta (3.1.4).

### 4.3 Show triangulation data

Spustí nám posúvateľnú ukážku rozloženia kamery a projektoru v priestore pomocou programu Octave. Údaje o priestorovom rozložení a náklone kamery a projektoru vznikli ako výstupné dáta s kalibrácie pomocou *Jean-Yves Bouguet Calibration Toolboxu*. Jednotlivé osi sú v milimetroch. Samotný Octave neponúka defaultne rotáciu bodov v priestore, preto pre reálny náhľad na orientáciu a vlastnosti kamery a projektoru môžeme využiť aj Matlab.

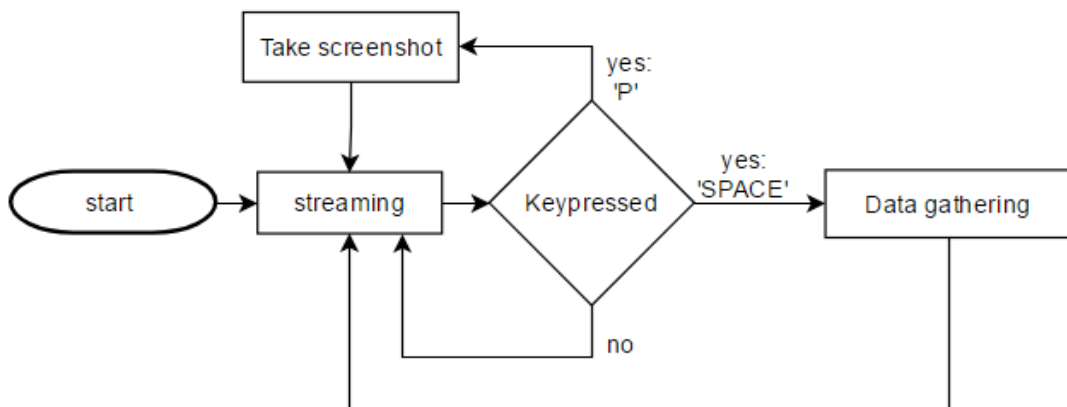


Obrázok 34: Show triangulation data

### 4.4 Start streaming session

Po stlačení tlačidla *Start streaming session* sa nám otvorí okno s zobrazením obrazu z kamery. Rozlíšenie obrazu je nastavené na 800x600 pixelov. Vyššie rozlíšenie nie je pre nás vhodné, pretože požadujeme čo najrýchlejšiu možnú odozvu v rámci možností. Pri príliš vysokej kvalite obrazu by sme zbytočne zmenšili frekvenciu snímkovania. Rovnako by sa zvýšil výpočtový čas, potrebný je funkcie upravujúce obraz ako je otočenia alebo zrkadlenie. Rozlíšenie 800x600 ale stále poskytuje dostatočnú rozlíšiteľnosť objektov a aj rýchlosť snímania. Táto možnosť z menu predkladá možnosť zakomponovania ovládania samotného robota. Modul *PyGame* pre Python, ktorý je svojimi niektorými knižnicami priamo určený na ovládanie výstupov z RPi, predkladá jednoduchú komunikáciu pomocou

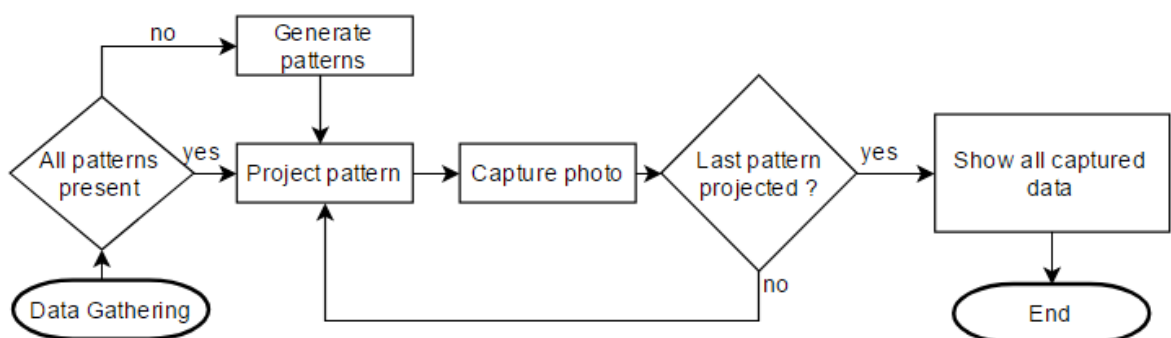
GPIO alebo I2C. medzi funkciami tejto funkcie z menu je aj možnosť tvorenia obrázkov obrazovky kamery, ktoré sa ukladajú do vlastnej zložky. Táto funkcia zaznamenáva obrázky v rozlíšení kamery. K získaniu dát k rekonštrukcii a následnej vizualizácii stlačíme klávesu SPACEBAR, ktorá nám zavolá skript pre *data gathering*. Počas zberu dát je nevyhovujúci pohyb robota, program v tejto fáze nezaznamenáva vstup z klávesnice, mimo escape sekvencií. Po ukončení funkcie znovu prebieha streaming s možnosťou ovládania robota.



Obrázok 35: Flow chart pre možnosť Streaming

## 4.5 Jump directly to data gathering

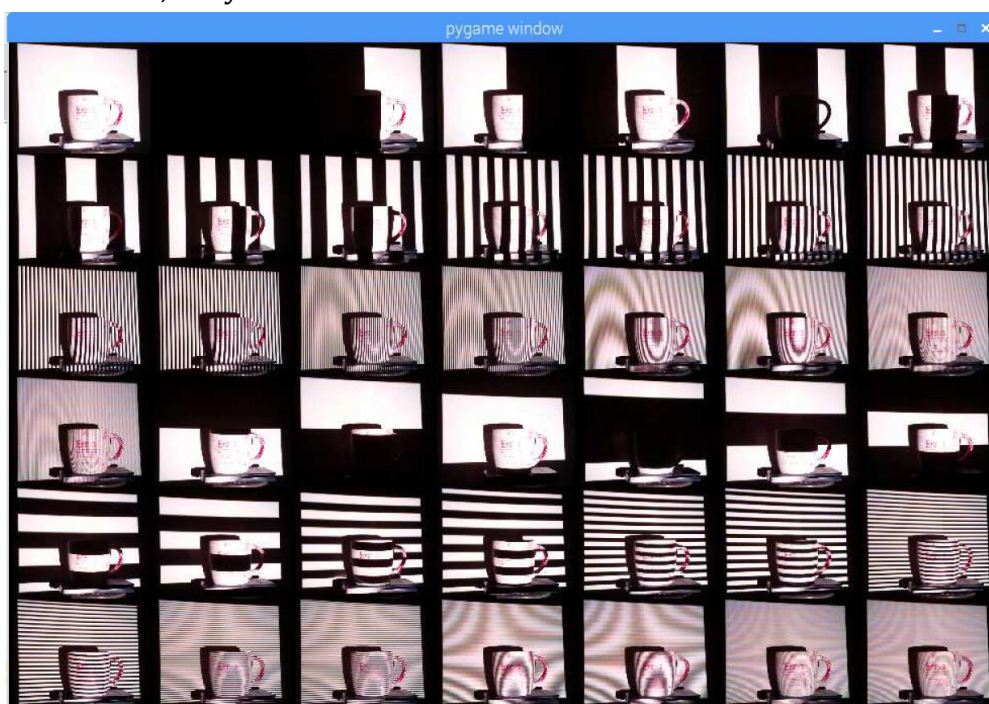
Hlavná časť programu, ktorá nám zachytí všetky potrebné dáta potrebné k spracovaniu obrazu do 3D dát. V tejto časti programu sa požaduje nehybnosť prípadného budúceho robota, preto sú zablokované všetky výstupy okrem tých nutných k skenovaniu. Proces sa dá prerušiť iba escape sekvenciou Ctrl+C. Po spustení sa zapne HDMI výstup. Projektor sa ale nemusí zapnúť hneď. Závisí to na aktuálnej pozícii v opakovanej sekvencii zisťovania aktivity na vstupoch. Postupne prepína medzi vstupmi a sleduje, či prichádza na daný vstup signál, ak nie tak zmení vstup. Samotné zapínanie trvá maximálne 11 sekúnd, takže po danú dobu prebieha inicializácia kamery. Po uplynutí danej doby sa spúšťa premietanie vzorov a ich zaznamenávanie. Ak program nenájde žiadne vzory, prípadne sú neúplné, príde k vygenerovaniu úplne nového setu štruktúrovaných vzorov.



Obrázok 36: Flow chart pre možnosť Jump directly to data gathering



Celkovo sa premietne všetkých 42 vzorov. Na pamäťovej karte sa v zložke DATA vytvorí z časom a dátumom aktuálneho skenovania, do ktorej sa uložia všetky obrázky a následne počas spracovania aj kalibračné dáta a model zobraziteľný na počítači v programe Matlab. Po ukončení skenovania sa zobrazia všetky zaznamenané obrázky (Obrázok 37). Po vypnutí okna sukážkou zaznamenaných obrázkov sa podľa funkcie pred spustením *data gathering*-u vykoná akcia. Ak bola možnosť *data gathering*-u spustená z možnosti *Start streaming session*, otvorí sa znovu okno so streamingom a po opätovnom stlačení SPACEBAR-u môžeme skenovať ďalší predmet. Ak sme túto možnosť spustili z hlavného menu, nevykoná sa žiadna funkcia.

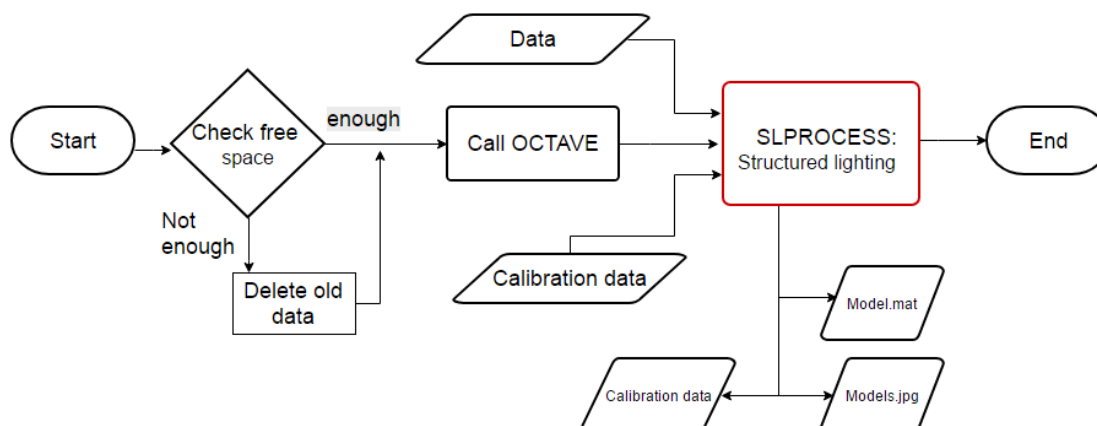


Obrázok 37: Priebek skenovania hrnčeku po nanesení práškového náteru (1.20.5)

## 4.6 Process data

Po spustení sa skontroluje dostatok miesta na disku, pretože táto funkcia môže generovať dáta veľkých objemov. Ak program zistí, že nie je na disku dostatok miesta, spustí sa automatické čistenie starých súborov. Po zabezpečení dostatku miesta na disku sa skriptom cez terminál automaticky spustí Octave spolu so zvolanými parametrami, ktoré predstavujú kalibračné dáta a umiestnenie dát na spracovanie. Program pracuje na pozadí a počas procesu môžeme ďalej ovládať robota. Táto akcia je však procesne náročná, môže prísť k zníženiu výkonu. Samotný skript v Octave predstavuje upravenú verziu skriptu napísaného pre študijné účely na Brownovej univerzite pod názvom BYO3D Scanner. Tento skript bol schopný zo znalosti kalibračných dát zistiť priestorovú zhodu premietaných pásov projektoru na predmete s využitím triangulácie. Upravený skript skopíruje

kalibračné dáta a modelové dáta potrebné k priestorovej vizualizácii vo dome súboru s koncovkou .mat.



Obrázok 38: Flow chart pre možnosť Process data

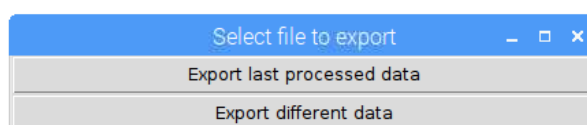
Tento formát je rozoznatelný ako natívny v Matlabe. Octave dokáže načítať .mat súbory z Matlabu ale naopak Octave ukladá defaultne dáta vo formáte nerozoznatelnom pre Matlab. Musíme previesť v Octave kompresiu dát a uloženie v binárnom formáte. Celkové uložené dáta o modeli dosahujú bez kompresie hodnotu až jedného gigabajtu, po kompresii polovicu. Po odstránení redundantných údajov a zbytočných maticových kópií sme schopný znížiť dátový objem na maximálne približne 100Mb. Pre prenos cez sieť sa stále jedná o veľký objem a je vhodné prísť s novými metódami na zmenšenie dátovej záťaže na sieti.

## 4.7 View already processed data

Skontroluje, či sa na zdieľanom disku už nachádza priečinok s dátami zo spracovania, ak nie tak ho vytvorí a presunie tam dáta pre zobrazenie na PC a tie sú programovo automaticky rozoznané a interpretované v prostredí Matlab. Otvorí sa nám aj graf obsahujúci obrázky pre rozoznané riadky, stĺpce a farebnú mapu na prekrytie bodov.

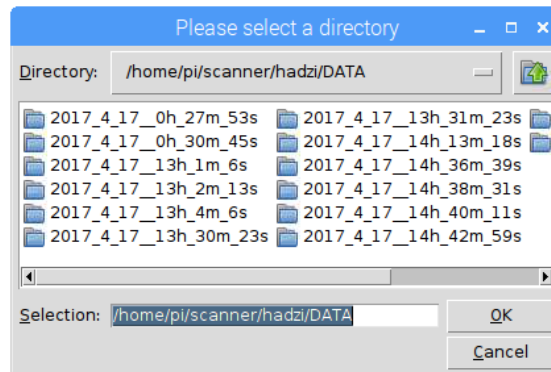
## 4.8 Export

Tlačidlo menu Export nám dovolí presunúť vybrané spracované dáta použiteľné pre nejaký 3D zobrazovací program, ktorý je schopný otvoriť súbory s danou koncovkou. Pretože nie všetky skeny sú vhodné ďalšieho spracovania, vyberáme si tie, ktoré presunieme a ktoré nie. Pre kompaktnosť sú mountnuté až dve USB zariadenia na prenos dát.



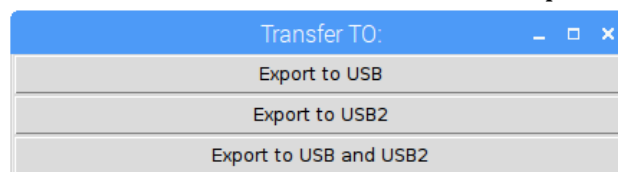
Obrázok 39: Časť Export z hlavného menu

Po rozkliknutí možnosti Export nám vyskočí výberové okno, v ktorom si môžeme vybrať, či chceme presunúť iba posledné spracované dáta alebo úplne iné spracované dáta. Prvá možnosť je tu z dôvodu rýchlosti, kde existuje predpoklad, že máme záujem presunúť posledný práve vyhotovený sken. Ak však klikneme na možnosť výberu iných dát, otvorí sa nám okno s výberom súboru, v ktorom sa nachádzajú dáta.



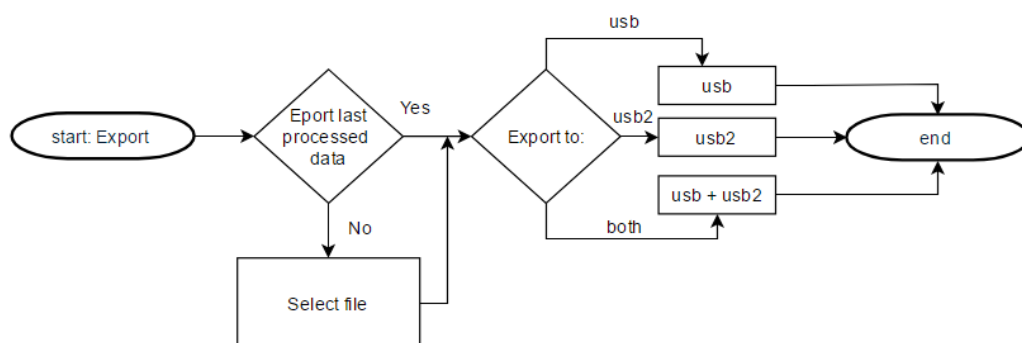
**Obrázok 40: Výberové okno pre export**

Po vybratí vhodného súboru obsahujúceho naše požadované dáta, máme možnosť špecifikovať USB zariadenie, na ktoré budú dáta prenesené.



**Obrázok 41: Export na disk**

Keďže v praxi môže byť potrebné prenášať dáta hneď na oba disky, nachádza sa tu aj takáto možnosť. Flowchart s vizualizáciou priebehu funkcie je na obrázku 42.



**Obrázok 42: Flowchart pre Export dát**

## 4.9 Close uncorrectly closed display

V niektorých prípadoch sa môže stať, že okno prestane reagovať, pretože program narazil na chybu. V takomto prípade zostáva okno zobrazovania vykresľovania obrázkov alebo stavu kamery otvorené a nedá sa vypnúť priamo. Jednou z možností je vpísaním do Python Shell-u napísať nasledujúce príkazy:

```
pygame.display.quit()
```

```
pygame.quit()
```

Prvý príkaz nám uzavrie nereagujúce okno a následne druhý príkaz nám vypne inštanciu pygame. Druhou možnosťou je znovu zavolanie z Python Shell-u nasledujúci príkaz, ktorý nám zastaví priebeh celého skriptu a dovolí interpretéru upratať za nereagujúcim skriptom:

```
sys.exit()
```

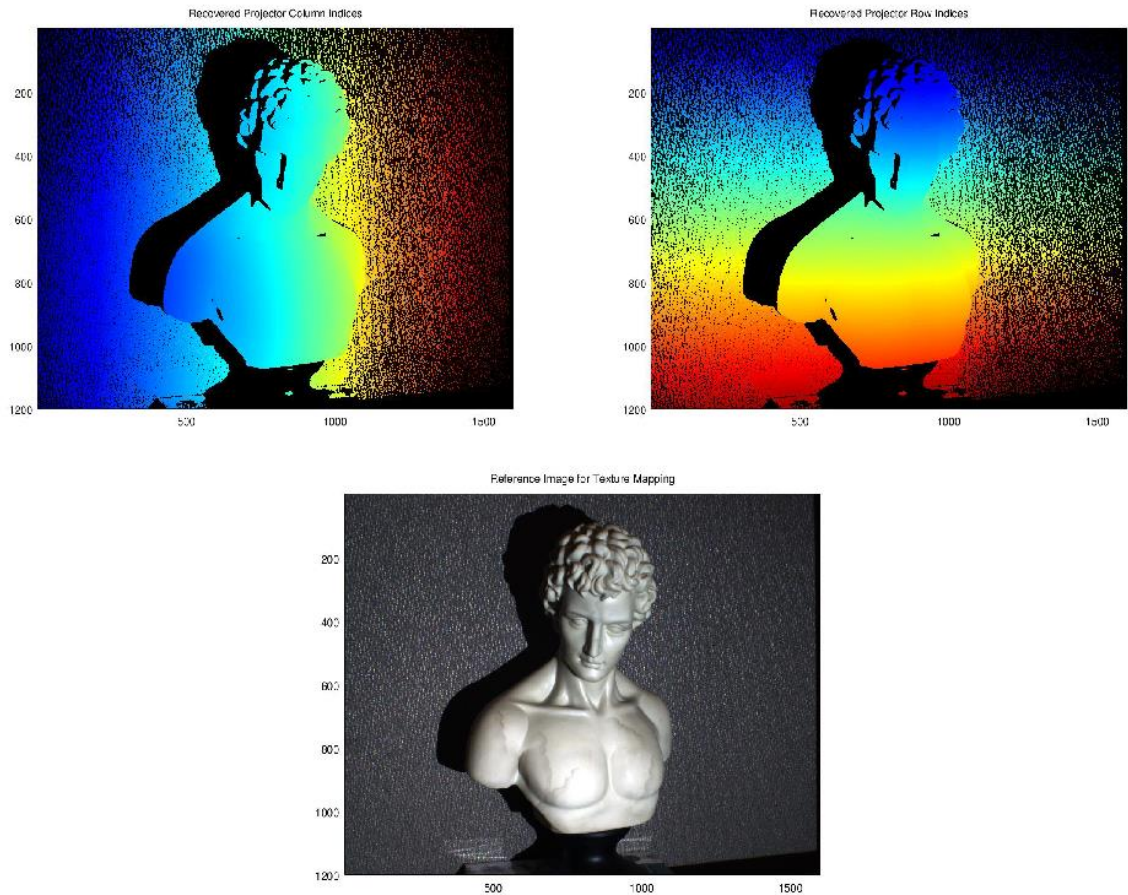
V tomto prípade som využil prvej možnosti, pretože chcem iba uzavrieť nereagujúce okno, nie ukončiť všetky otvorené skripty ani menu.

## 4.10 Quit

Posledná možnosť v menu nám dovolí ukončiť program. Pri ukončení príde k zatvoreniu všetkých otvorených okien aj ostatných podokien vytvorených knižnicou *tkinter*, prípadne aj nereagujúcich okien rovnako ako v predchádzajúcom tlačidle z menu. Povolí sa výstup na HDMI. Následne sa ukončí program.

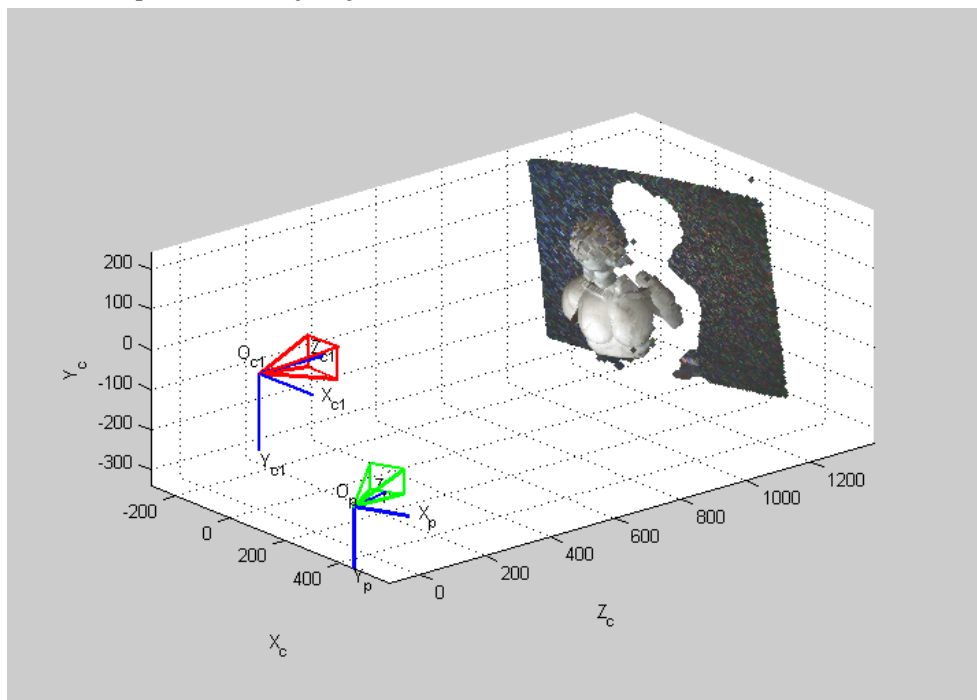
## 4.11 3Dvisualizer

Pre možnosť zobrazovania priestorových dát, ktoré sú spracované na zariadení RPi som vytvoril skript *3Dvisualizer*. Po spustení hlavného programu je automaticky vygenerovaný a nahratý spolu so všetkými nutnými súbormi na USB disk a tým sa vďaka zdieľaniu súborov pomocou *Samby* stane prístupný pre všetky počítače v sieti. Medzi súbormi sa nachádza aj README súbor s postupom pri spustení skriptu. Celý súbor v ktorom sa nachádza samotný skript a aj jeho nutné súbory sa presunú na plochu počítača, v ktorom chceme zobraziť vizualizáciu. Spustí sa hlavný skript *3Dvisualizer.py*, ktorý automaticky nájde cestu k .exe súboru Matlab-u. Ak je nainštalovaných viac verzií Matlab-u, spýta sa ktorú chceme použiť pre otvorenie. Rovnako aj overí existenciu všetkých potrebných súborov a uloží systémovú cestu, z ktorej sme jednotlivé súbory stiahli (pozn.: Máme nastavenú statickú IP, takže cesta by mala byť vždy rovnaká). V nekonečnej slučke nenápadne na pozadí detekuje export dát z Raspberry Pi. Ak detekuje prijatie všetkých modelov a aj kalibračných údajov, spustí vizualizáciu. Zobrazia sa vyčítané dáta pre odhadnuté riadky a stĺpce projektora (Obrázok 43). Z obrázkov je vidieť kvalita rekonštrukcie pri kalibrácii s vysokou presnosťou.



**Obrázok 43: Spracovanie dát priložených s kalibračnými dátami od tvorcu SLPROCESS [14]**

Následne sa spustí programové prostredie Matlab-u a zobrazí sa nám reprezentácia dát v priestore (Obrázok 44). Vďaka dobrej kalibrácii sa tu nachádza len minimálne percento chybných bodov.



**Obrázok 44: Vizualizácia spracovaných priestorových tvorcu SLPROCESS [14]**



## 5. Testovanie skeneru

### 5.1 Testovanie presnosti

Pri výbere vhodného zariadenia pre skenovanie modelov do 3D, je jedným z hlavných parametrov presnosť. Hlavne sa jedná o rozmerovú súhlasnosť, nízky obsah šumu a v prípade nutnosti zachovania farby aj súhlasnosť prekrytia bodov v priestore farebnou mapou. Ak je chybná systémová kalibrácia, zanáša sa do celého merania systematická chyba, ktorú je ťažké odstrániť.

#### 5.1.1 Presnosť rozmerov a šumu

Pre približné overenie zhody reálnych rozmerov objektu a vyhodnotených rozmerov vo vizualizácii využijeme možnosť odčítať dáta z vizualizácie. Jedná sa o približné odčítanie presnosti. presné by prebiehalo porovnaním s modelom presného skalibrovaného 3D skeneru alebo väčším počtom meraní. Pre približné stanovenie rozdielu sme využili kalibračný vzor so známou veľkosťou jednotlivých štvorčekov.



Obrázok 45: Meranie presnosti metódy pre 3D skenovanie

S využitím Pytagorovej vety môžeme vypočítať zmeranú vzdialenosť dvoch krajných bodov. Reálna veľkosť rozmerov je 30x30mm. Medzi vyznačenými bodmi sa nachádza celkovo 7 štvorčekov, takže dĺžka medzi bodmi by sa mala rovnať 210mm.

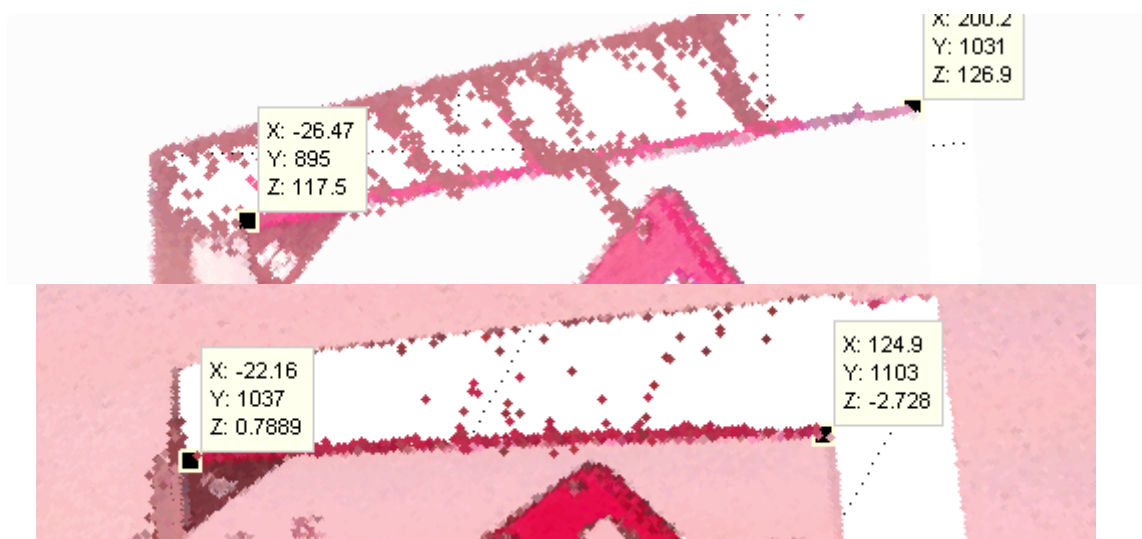
$$\begin{aligned}
& d([-27.12, 1015, 155.8], [165.1, 1141, 150.7]) \\
& = \sqrt{(165.1 + 27.12)^2 + (1141 - 1015)^2 + (150.7 + 155.8)^2} \\
& = 230.562 \text{ mm}
\end{aligned} \tag{10}$$

$$\Delta d = 230.562 - 210 = 20.562 \text{ mm} \tag{11}$$

$$\delta d = \frac{20.562}{230.562} \cdot 100 = 8.918\% \tag{12}$$

Jednoduchými výpočtami môžeme získať odchýlku absolútnej hodnoty (vzorec č. 11) niečo cez 2 cm a relatívnu (vzorec č.12) skoro 9%. Jedná sa ale iba o rozmerové skreslenie, spôsobené nechceným miernym posunom častí zariadenia medzi jednotlivými meraniami, ktoré sa vykonávali v rôzne dni. Chyba bola v tomto prípade jemný posun kamery ďalej od meraného objektu. Pri zmeraní vlastností okamžite po kalibrácii by boli výsledky výrazne menšie. Z obrázku môžeme posúdiť aj charakter šumu. Náhodný šum sa na rovných plochách vyskytuje len veľmi zriedkavo, pretože väčšina chybných bodov bola softvérovo odstránená. Problémové oblasti sú väčšinou príliš lesklé, priehľadné predmety, prípadne s nevhodnou farbou. S týmito chybami skenovania sa stretneme neskôr v texte.

Ďalší faktor, ktorý ovplyvňuje množstvo šumových bodov sú aj svetelné podmienky a kalibrácia. Na nasledujúcom obrázku (Obrázok 46) je vidieť efekt premiestnenia skenovacieho zariadenia do inej, tmavšej miestnosti. Obrázok sa javí svetlejší, ale je to spôsobené väčším kontrastom projektovaných vzorov. Rovnako pri premiestňovaní prišlo k jemnému posunutiu častí testovacieho systému.



**Obrázok 46: Vlastnosti kalibrácie a zmeny osvetlenia na výsledný šum**

## 5.1.2 Zhoda prekrytím farebnou mapou

Farba modelu pri samotnej výrobe nie je väčšinou potrebná. Potrebnou sa stáva pri návrhoch dielov, kde sa snažíme zladit' farebnosť alebo skopírovať vlastnosti modelu. Ako je možné vidieť neskôr v testoch, farebná mapa sa prenáša relatívne správne. Jej kvalita je daná hlavne rozlíšením kamery, kde sme schopný získať väčší počet farebných bodov pre väčšie rozlíšenia. Svetelné podmienky v miestnosti môžu spôsobiť pri príliš nízkom osvetlení a následne príliš vysokou intenzitou projektoru nepravú farbu. Ktorá je jasovo skreslená. Možnosť odstránenia môže predstavovať doplnenie jasovej kalibrácie, ktorá pred samotným meraním premietne na predmet čisto bielu plochu a zo zaznamenaného obrazu na kamere čiastočne zatmaví biele časti v premietaných vzoroch. Farebná kalibrácia pri premietaní vzoru s čiernymi a bielymi pásikmi nie je potrebná. Pre metódy skenovania so štruktúrovaným osvetlením, kde využívame farebné vzory však môže predstavovať nutný krok, keďže intenzita vyžarovania farieb nesleduje lineárnu charakteristiku.

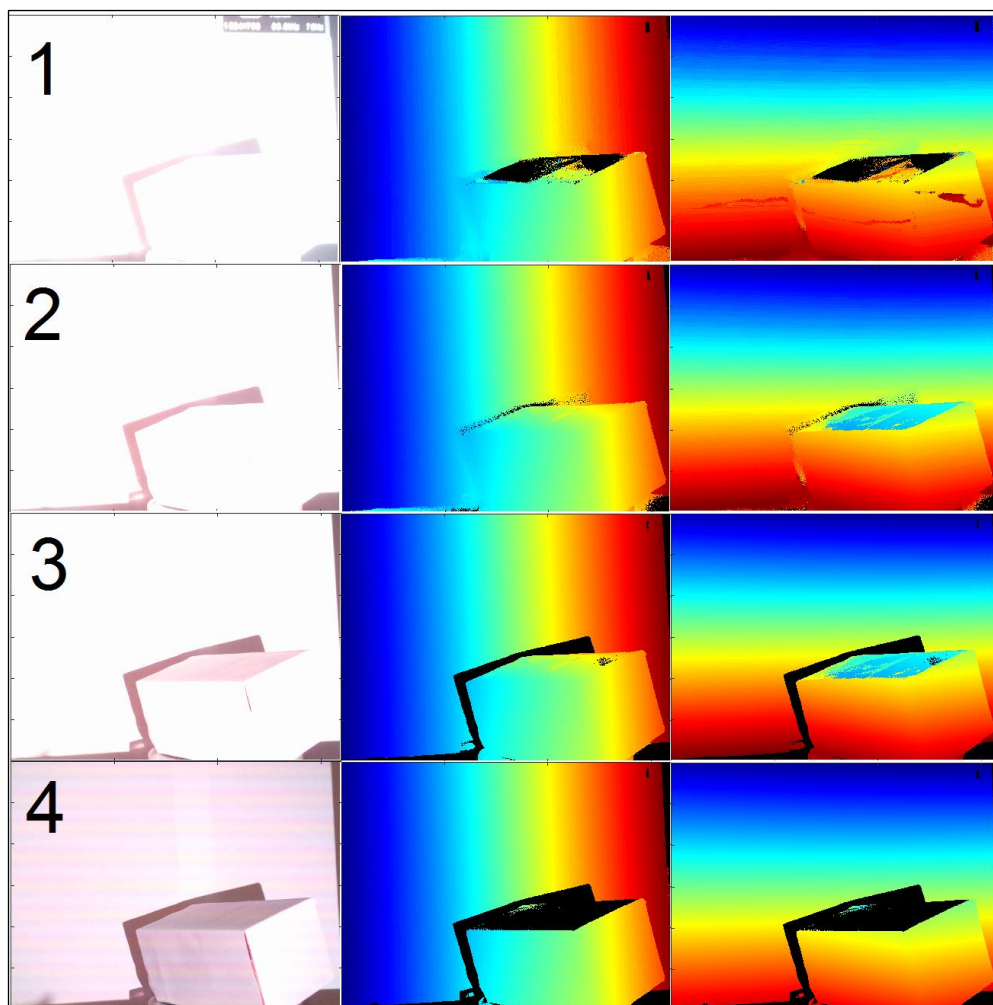
## 5.2 Testovanie vplyvov prostredia

Znehodnotenie spracovania modelu spôsobujú aj vonkajšie parametre ako úroveň intenzity osvetlenia predmetu, nerovnomernosť osvetlenia, prašnosť prostredia, nevhodnosť pozadia alebo náhodné odrazy spôsobené okolím. Pre intenzitu osvetlenia a vhodnosť pozadia boli spracované testy v nasledujúcich bodoch práce.

### 5.2.1 Intenzita osvetlenia

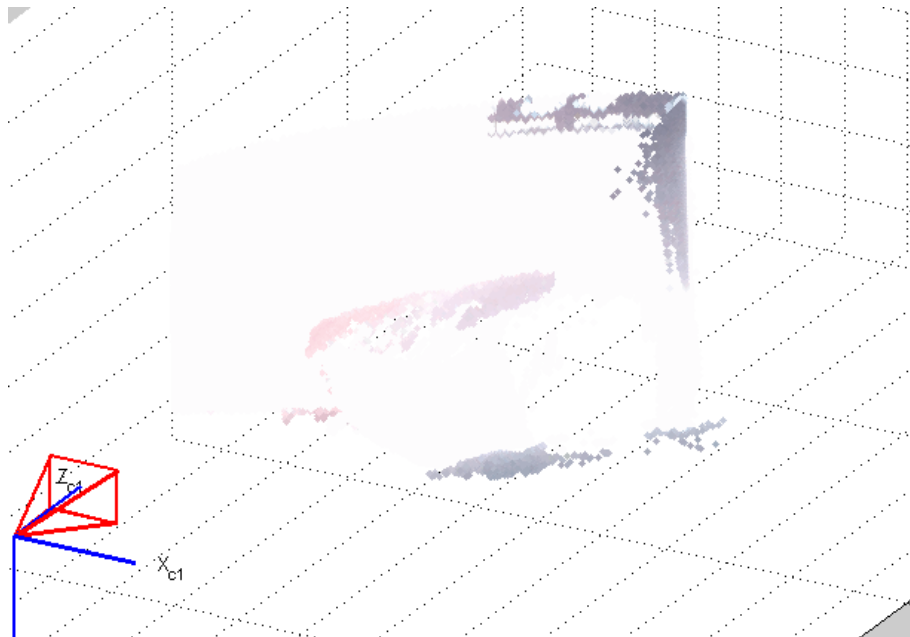
Pre bežné používané farby objektov nie je intenzita osvetlenia až tak viditeľná. Najviac sa prejaví pri čisto bielom svetle objektu spolu s bielym pozadím alebo čisto čiernych objektoch s jemnou povrchovou štruktúrou. Pre toto testovanie som využil objekt, zabalený do bieleho papiera. Ako pozadie som vybral rovnako biely papier. Biely papier podľa typu prísad a opracovania, ktoré má vplyv na jeho výsledný povrch. Reflektivita môže dosahovať podľa experimentálnych meraní až okolo 70-80% [22], v extrémnych prípadoch bolo možné namerať hodnoty blížiacie sa totálnemu odrazu [23]. Pre tmavé povrchy s jemnou povrchovou štruktúrovanosťou je reflektivita rádovo v jednotkách percent (pozn.: merania prebehli pre úzky vyžarovací uhol) [22]. Na obrázkoch 47 až 49 je zobrazenie kvality rozpoznania tvaru na intenzite osvetlenia.



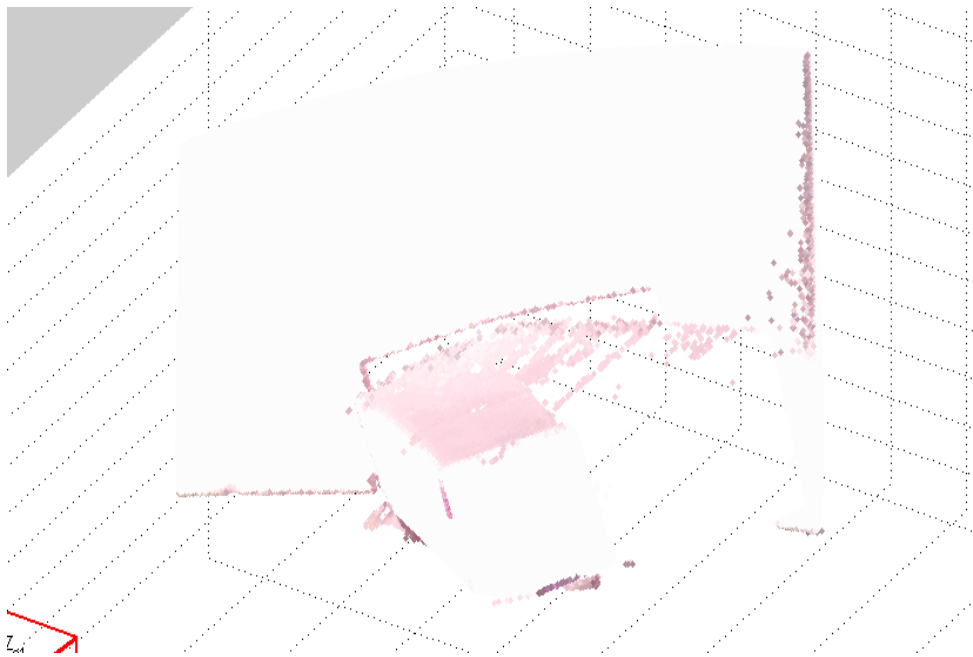


**Obrázok 47: Vplyv osvetlenia prostredia na kvalitu skenovania**

Na obrázku 47 môžeme vidieť rekonštruované riadky a stĺpce projektoru pre rôznu intenzitu vyžarovania, počínajúc s najväčšou intenzitou vyžarovania a absolútnej tme pre '1' až po bežné vonkajšie osvetlenie v '4'. Na obrázku 48 je následné spracovanie do modelu pre absolútnu tmú a maximálny jas projektoru. Je zjavné, že rozpoznanie neprebehlo ideálne. Softvér nebol schopný správne identifikovať tvar objektu, model je necelistvý a obsahuje veľké množstvo dier. Časť '2' a '3' predstavuje bežné podmienky v miestnosti, kde nie je príliš vysoká miera vonkajšieho osvetlenia. Pre možnosť '4', ktorá predstavuje vysoký jas ale vonkajšie osvetlenie (odtemnené všetky okná v miestnosti a zapnuté všetky svietidlá) prebehlo zisťovanie riadkov a stĺpcov projektoru výrazne kvalitnejšie. Výsledný model je možné vidieť na obrázku 49. Z modelu je vidieť priaznivejšie vypočítanie jednotlivých plôch. Nachádza sa tu menšie množstvo dier ako pre absolútnu tmú v miestnosti.

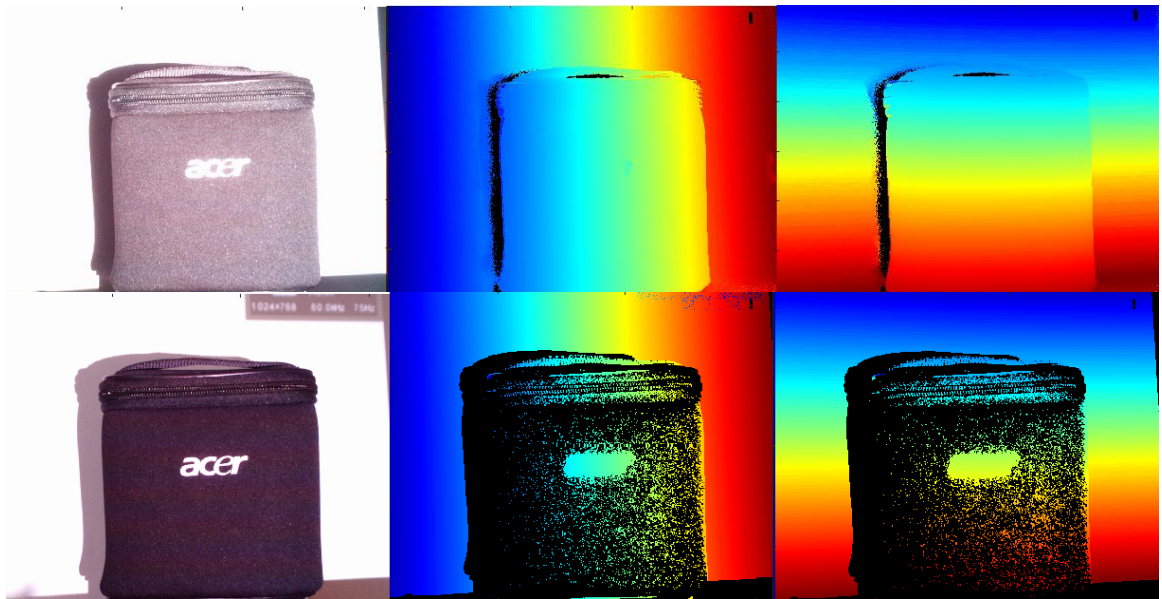


**Obrázok 48: Výsledný rozoznaný model pri (1) maximálnej hodnote jasnosti projektoru a absolútnej tme v miestnosti**

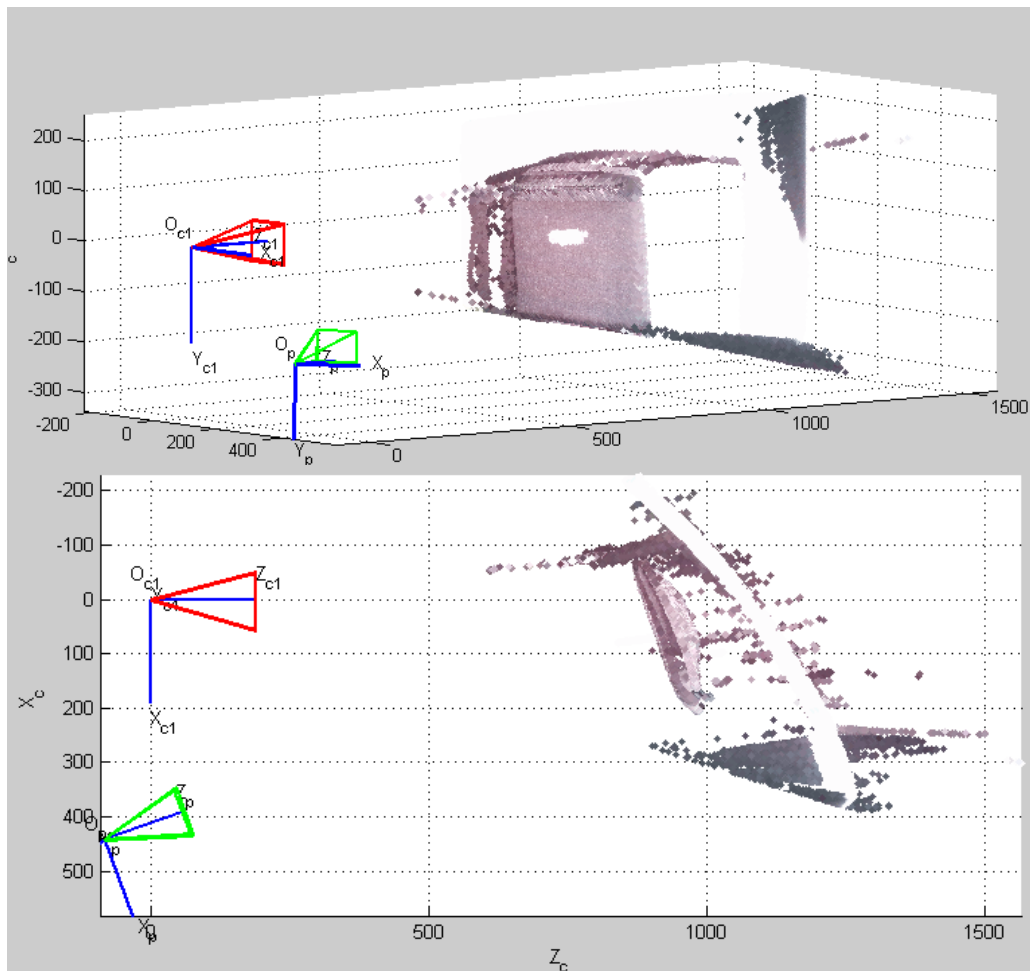


**Obrázok 49: Výsledný rozoznaný model pri (4) vysokej hodnote jasnosti projektoru a vysokého stupňa vonkajšieho osvetlenia**

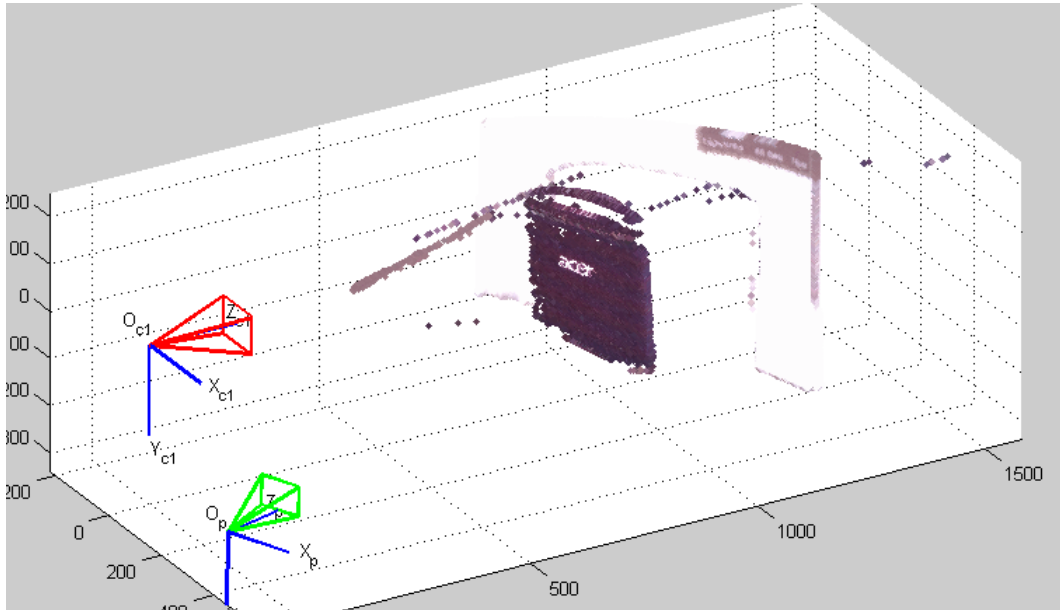
Pre čierne, jemne štruktúrované predmety prebehol obdobný test. V obrázku 50 možno vidieť pre tmu v miestnosti relatívne správne určenie riadkov a stĺpcov odpovedajúcich projektoru, avšak aj veľké množstvo šumu, ktoré je vidieť na obrázku 51. Šum v tomto prípade vzniká v na snímaných plochách objektu kamerou, ktoré tvoria príliš malý uhol s lúčom svetla zachytávaným kamerou.



Obrázok 50: Čierny, jemne štruktúrovaný objekt pri absolútnej tme v miestnosti a bežnom osvetlení



Obrázok 51: Rekonštrukcia čierneho, jemne štruktúrovaného objektu pri absolútnej tme v miestnosti a maximálnom jase projektora

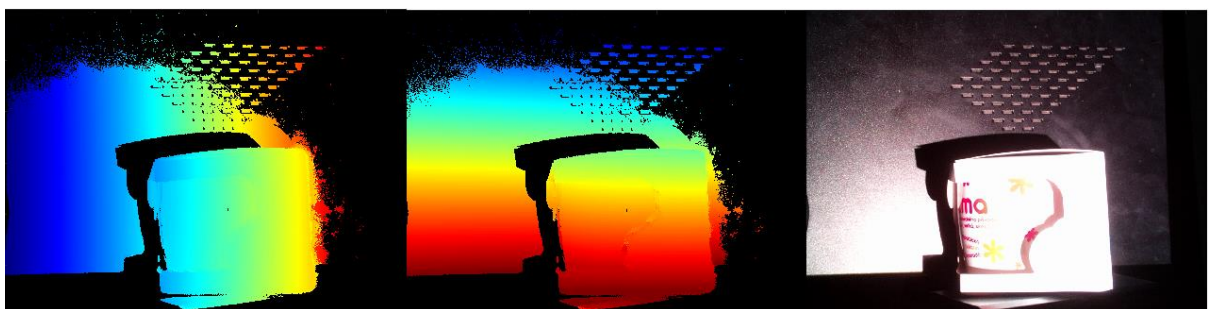


**Obrázok 52: Rekonštrukcia čierneho, jemne štruktúrovaného objektu pri bežnom osvetlení v miestnosti a priemernom jase projektora**

Pre silnejšie vonkajšie osvetlenie vzniklo veľké množstvo nespracovaných bodov, avšak množstvo šumu a aj farebné prekrytie farebnou mapou je lepšie. Vysvetlenie príčin straty pixelových informácií pre čierne predmety je vysvetlené v časti testovania farby.

### 5.2.2 Vhodnosť pozadia za skenovaným objektom

Pozadie pre skenovanie s využitím štruktúrovaného osvetlenia by nemalo byť lesklé. Zároveň z výsledkov testov vyplýva, že by nemalo byť príliš tmavé a pokiaľ možno čo najrovnejšie. Navzdory teoretickým zdrojom [14][24], ktoré odporúčajú čierne pozadie sa zdá lepšou alternatívou šedé pozadie. Príliš tmavé plochy pohltia veľké množstvo svetla a softvér následne takéto chybné zosnímané body odstráni. Zrkadlové alebo lesklé pozadia môžu vrhať nechcené odrazy vzoru na predmet a zhoršiť výpočtovú schopnosť programu určiť správne tvar. Rovnako pri rozptyle svetla môžu vzniknúť nepriamo osvetlené miesta. Na obrázku 53 je vidieť rekonštruovaný odhad pre riadky a stĺpce projektora pre jemne lesklé čierne pozadie, ktoré obsahuje povrchové nerovnosti.



**Obrázok 53: Nevhodnosť pozadia za objektom, ktorý chceme rekonštruovať**

## 5.3 Vplyv vizuálnych vlastností skenovaného objektu

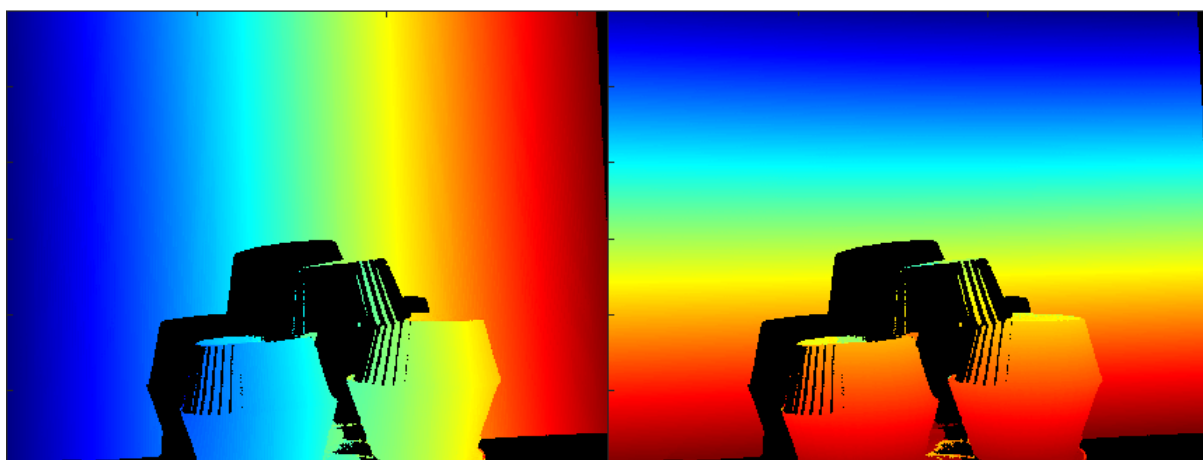
Medzi vizuálne vlastnosti možno zaradiť hlavne lesklosť alebo priehľadnosť povrchu, farbu a povrchovú štruktúru.

### 5.3.1 Farba objektu

Reálne telesá odrážajú pre rôzne farby povrchu svetlo rozdielne. Farba a textúra objektu ovplyvňujú, koľko svetelnej energie je absorbovanej alebo reflektovanej. Pri tom každá farba reflektuje určité množstvo svetla, zatiaľ čo zvyšok pohltí vo forme tepla. Množstvo reflektovaného svetla od farby je označené hodnotou odrazivosti svetla. Čierne alebo veľmi tmavé objekty s nízkou hodnotou odrazivosti neodrážajú svetlo v dostatočnej miere, väčšinu pohltia. Na rozdiel od toho svetlé predmety s vysokou hodnotou odrazivosti pohlcujú málo svetla a väčšinu odrazia [25]. Najväčší rozdiel môžeme pozorovať pri porovnaní kvality skenu dvoch predmetov rovnakého tvaru a vlastností, iba s rozdielnou farbou. Na overenie vplyvu farby som teda vybral čiernu, dve tmavé a dve svetlé farby na bielom pozadí (Obrázok 54).



Obrázok 54: Vplyv farby skenovaného objektu

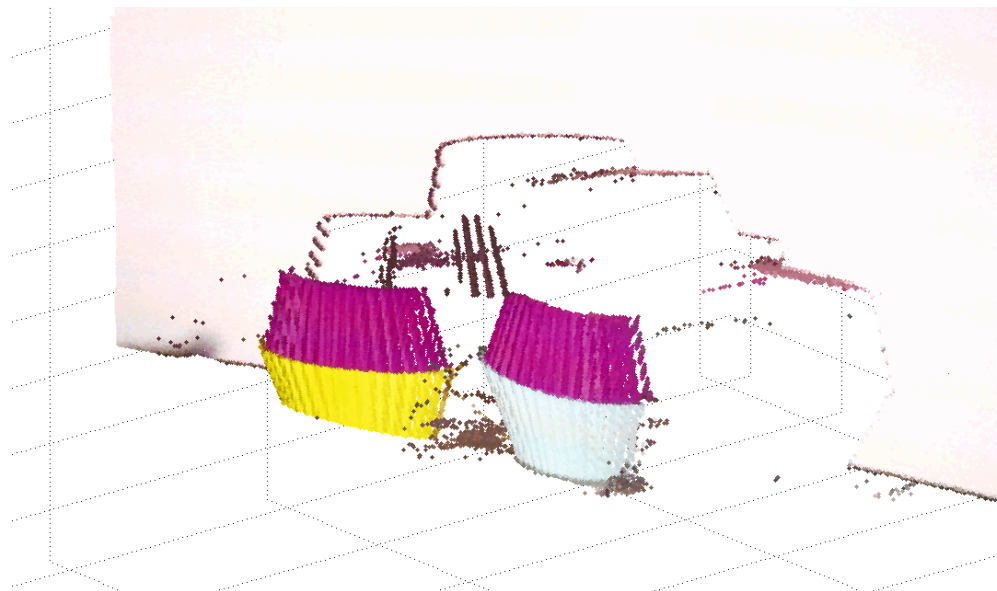


Obrázok 55: Rozoznanie jednotlivých riadkov a stĺpcov pre rekonštrukciu farebných objektov

Zo zistených odhadnutých riadkov a stĺpcov projektoru je jasne vidieť stratu informácie o tvare objektu s veľmi tmavou farbou. Čierny objekt je nerozoznatelný

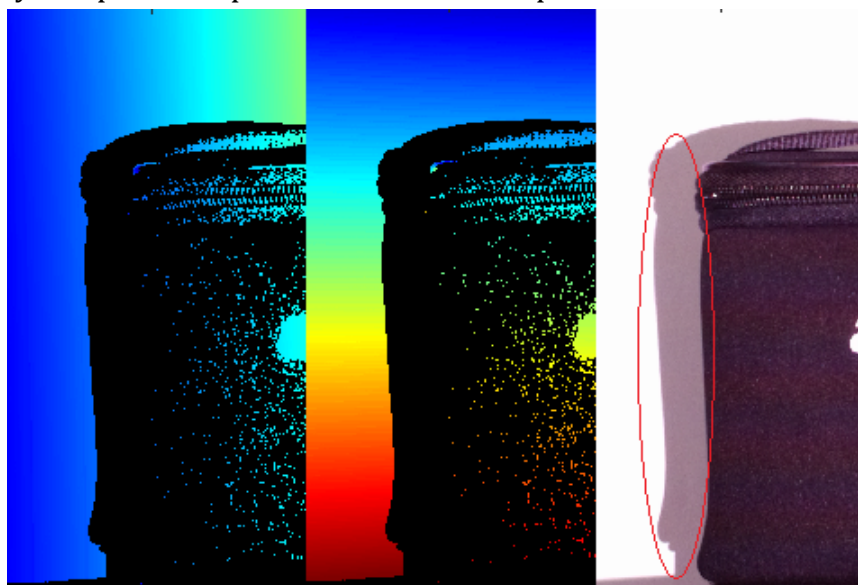


od tieňu, preto je aj softvérovo rovnako interpretovaný a odstránený ako chybový pixel. Mierne tmavé objekty pohltili tiež určité množstvo svetla, ale je ťažké z obrázku odhadnúť, či sa jedná iba o vplyv farby alebo aj rozdielnej orientácie košíčkov oproti spodným s bledou farbou. Model (Obrázok 56) ukazuje úplnú stratu informácie o tvare pre čierny košíček.



Obrázok 56: Výsledná rekonštrukcia farebných objektov

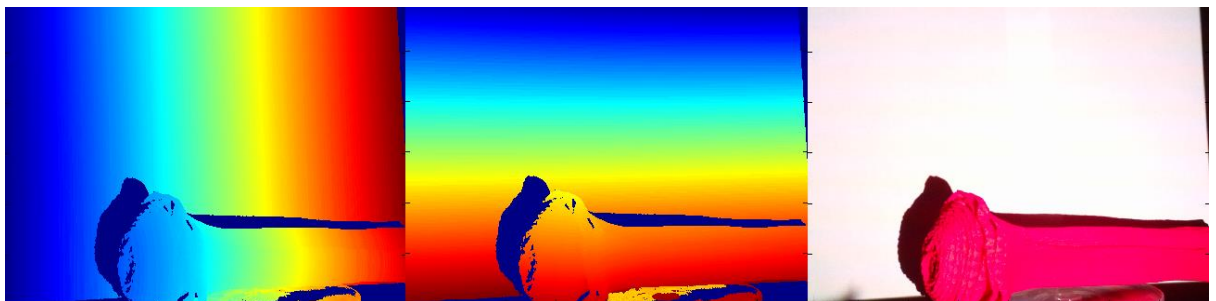
Pri veľmi tmavých objektoch, nie úplne čiernych, ako bol košíček z predchádzajúceho príkladu môžeme vidieť rovnako stratu informácie. V obrázku 57 je demonštrácia nesprávneho posúdenia a vyhodnotenia čiernej farby, zameniteľnej za tieň. Chybné pixely sú v tomto prípade nesprávne interpretované a nie sú odstránené ale vizualizované ako objektové pixely. Prejaví sa to bodmi rozhádzanými v priestore po línii medzi reálnou polohou tieňu a kamerou.



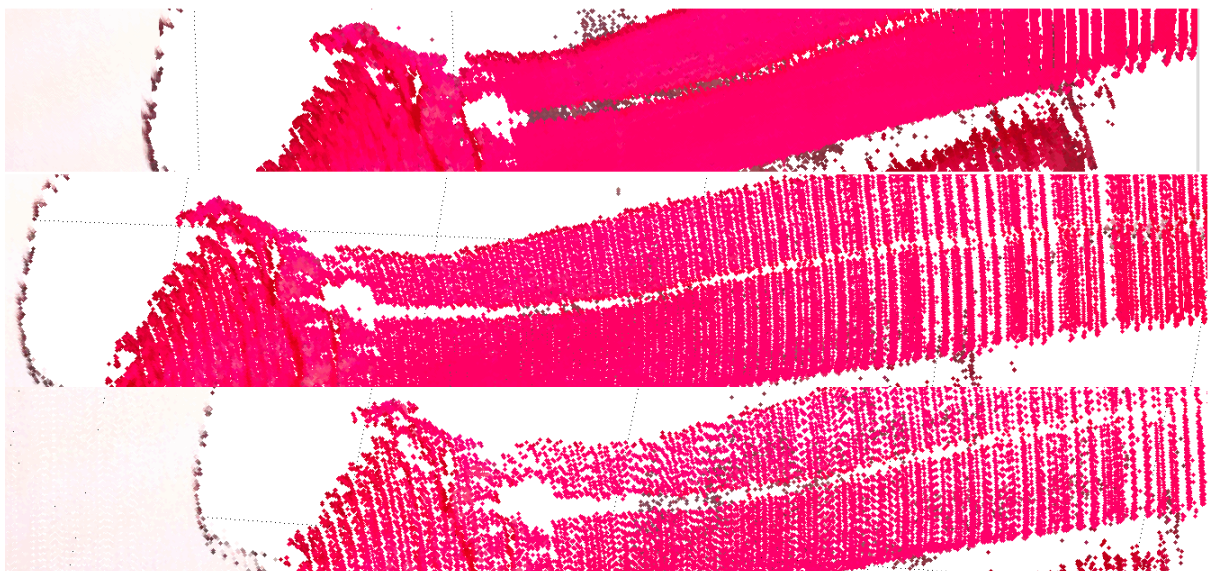
Obrázok 57: Neschopnosť rozoznania tieňov pre veľmi tmavé až čierne objekty

### 5.3.2 Vplyv štruktúrovanosti objektu a zachytenia drobných detailov

Na kvalitu štruktúry jemných detailov rekonštruovaného modelu najviac vplyva rozlíšenie kamery, metóda výpočtu a interpretácie bodov a kamery. Samotná kvalita vykreslenia modelu ďalej závisí aj na počte spracovaných pixelov prepočítaných do jedného bodu vo vizualizácii. V bežných projektoch nemusí počet získaných pixelov zodpovedať počtu vykreslených bodov v mračne bodov. Pri predmetoch s jednoduchou štruktúrou sa môže uprednostniť odstránenie redundantných bodov, aby sa zmenšil dátový objem. Na vizualizácii zaznamenávaného krepového papiera (Obrázok 58) môžeme uplatniť rôzne počty zmenšenia počtu vykresľovaných bodov. Na (Obrázok 59) môžeme vidieť vykreslený krepový papier pre *downsampling* 1px , 5px , 10px postupne od vrchu.



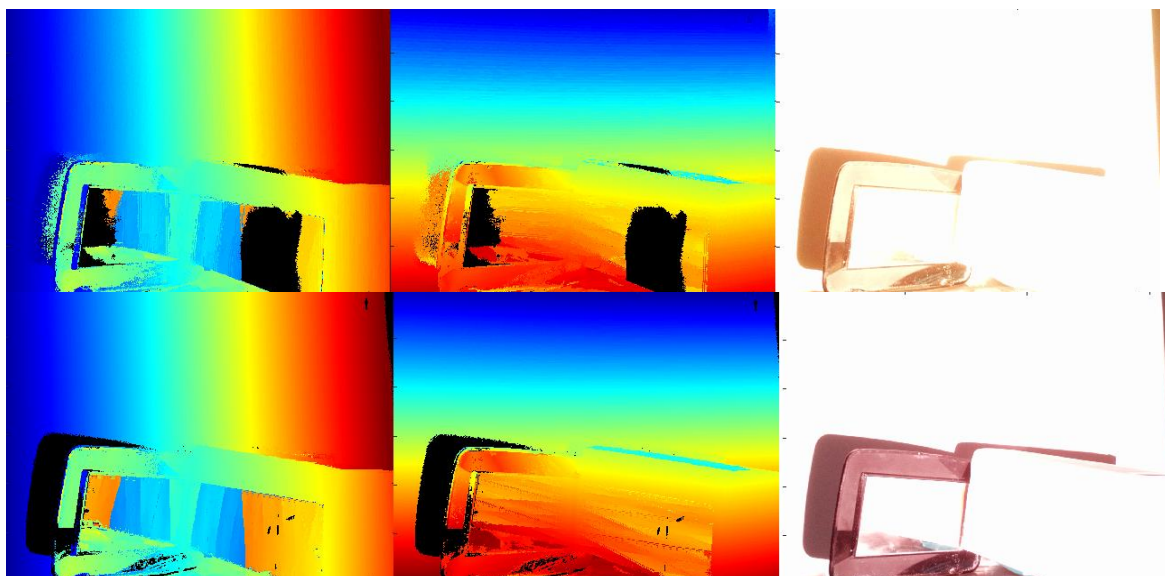
Obrázok 58: Nasnímanie krepového papiera pri meraní detailnosti objektu



Obrázok 59: Zachytený detail povrchu krepového papiera

### 5.3.3 Difúzne svetlo a odrazené štruktúrované vzory

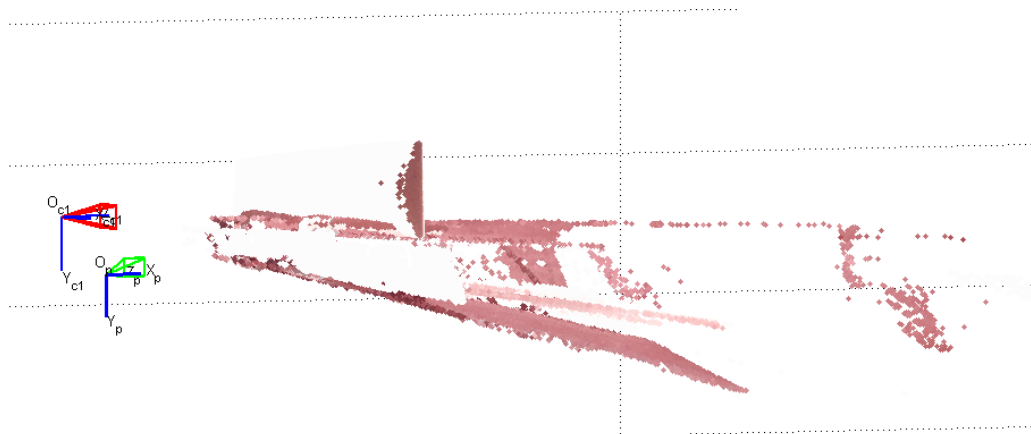
Štruktúrované vzory s veľkou frekvenciou vzoru sú na väčšiu vzdialenosť bez prítomnosti lokálne krátko rozsahových efektov ako odrazy a rozptyl na objekte vhodnou voľbou. Vhodné sú aj pre to, že pomer čiernych a bielych pásov sa rovná. V skutočnosti ale ak berieme v úvahu aj rozptyl a odraz svetla na povrchu objektu, väčšina plochy je osvetlená okrem priameho osvetlenia z projektora aj rozptylovým alebo odrazeným svetlom. Vzory s vysokou frekvenciou pásov (menšia pixelová šírka pásov) sú teda osvetlené a odrazy spôsobujú rozmazanie pásov zachytených pomocou kamery. Počas vyhodnotenia teda prichádza k nesprávnemu určeniu polohy pixelu kvôli nesprávnemu zaradeniu spôsobeného neodpovedajúcim kontrastom. Odstraňuje sa to zmenšením frekvencie pásov (väčšia pixelová šírka pásov). Je nutný väčší počet premietaných vzorov ale neprichádza k rozmazaniu. Dnešné programy premietajú postupne vzory so zvyšujúcou sa frekvenciou pásov. Získajú tak presnejší obraz o povrchu objektu. [26] Ďalším spôsobom odstránenia difúzneho svetla je premietanie inverzných vzorov pre každý premietnutý vzor. Následne sa pre daný pixel porovnaním základného aj inverzného vzoru vyhodnotí, či je pixel naozaj osvetlený, alebo sa jedná iba o difúzne svetlo. [14] Horším prípadom sú odrazené štruktúrované vzory od lesklého alebo zrkadlovitého povrchu. Pre simulovanie som použil najhorší možný prípad a to zrkadlo, odrážajúce vzory naspäť na predmet. Na obrázku 60 je znázornený odhad riadkov a stĺpcov projektora pre predmet, na ktorý dopadá okrem priameho svetla z projektora aj odrazené od zrkadla. V hornej časti obrázku nastal predmet pre tmú v miestnosti a plný jas projektora a v dolnej časti obrázku pre bežné osvetlenie v miestnosti. Obe rekonštrukcie bodov sú chybné, väčšiu chybu však vykazuje skenovanie pri tme v miestnosti.



Obrázok 60: Vplyv odrazených štruktúrovaných vzorov od zrkadlovitého predmetu

Pre obe možnosti je výsledná rekonštrukcia absolútne nezmyselná. Body modelu sú rozhádzané v priestore (Obrázok 61).

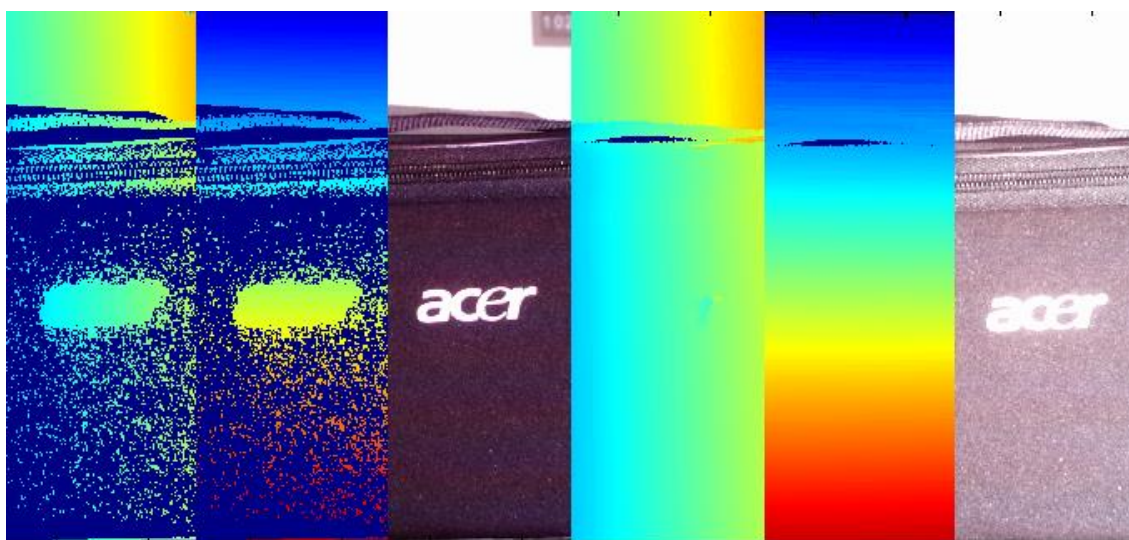




Obrázok 61: Rekonštrukcia objektu s vysokou mierou difúzie alebo odrazov

### 5.3.4 Reflexné a veľmi lesklé povrchy

Príliš lesklé objekty pri skenovaní spôsobujú presvetlenie senzoru. Ak je lesklá plocha v prostriedku objektu, má s tým skener menšie problémy. Väčšie množstvo diery a chýb môžeme pozorovať pri lesklosti v blízkosti zlomov. Na obrázku 62 sa nachádza mierne reflexný nápis na objekte. Nápis nebol výrazne znehodnotený a relatívne rozoznatelný programom.

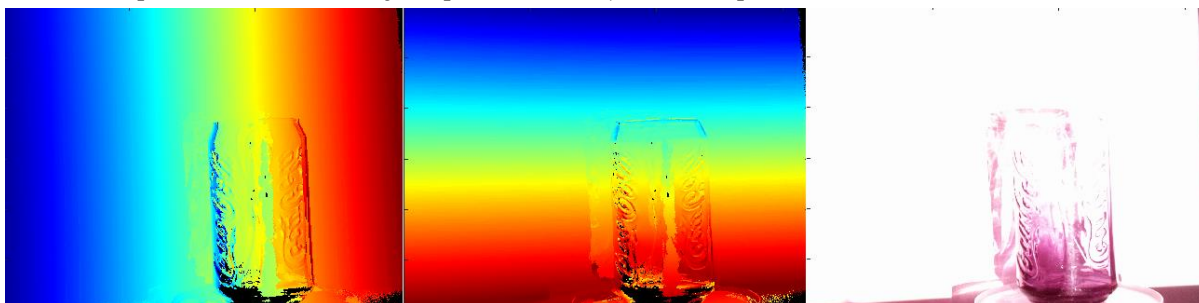


Obrázok 62: Vplyv reflexných povrchov na rekonštrukciu objektu

Pri predmetoch s vyšším stupňom odrazivosti by vznikali diery alebo rozmazané plochy v modeli. Rekonštruovaný model pre bežné svetelné podmienky je na obrázku 52. V tomto rekonštruovanom obrázku nie je výrazne zjavné rozmazanie ale v obrázku 51, kde bola použitá vyššia intenzita osvetlenia pre tmú miestnosť je rozmazanie textu viditeľné. Text je prakticky nečitateľný. V tomto bode treba brať v úvahu aj automatické nastavovanie snímaného jasú kamery, ktoré ovplyvňuje výsledok.

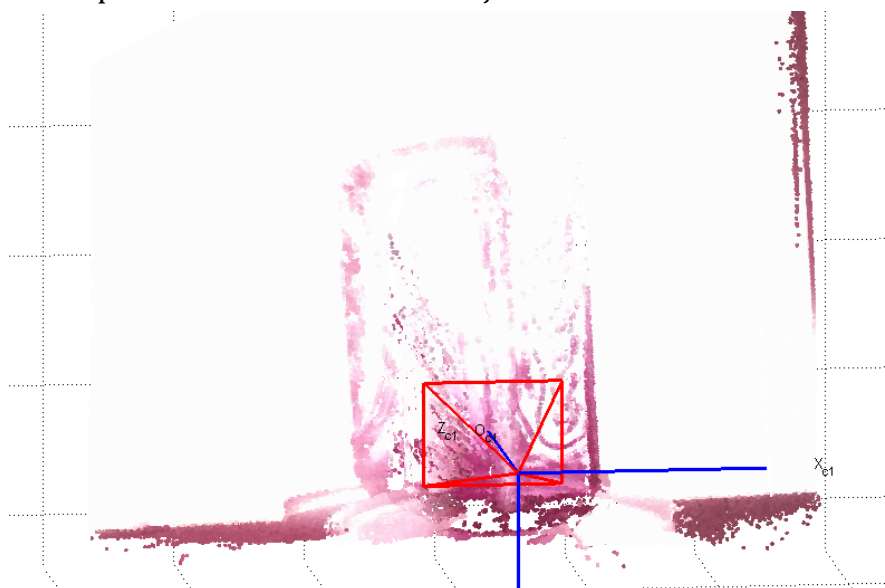
### 5.3.5 Priehľadnosť materiálu

Priehľadné objekty nespĺňajú väčšinu základných predpokladov, ktoré využívajú obrazové algoritmy. Napríklad čiastočne transparentné objekty môžu spôsobiť deformáciu pozadia, zvýšenie množstva šumu či pri veľmi transparentných objektoch úplne skreslenie objektu. Celkovo tieto objekty výrazne komplikujú proces modelovania. Z optických metód je pre skenovanie priehľadných objektov vhodnejšie využitie siluet alebo zisťovanie tvaru pomocou termovízie nahriatych objektov trianguláciou s predchádzajúcou kalibráciou.[2] Na overenie vplyvu transparentnosti materiálu som naskenoval priehľadný pohár (Obrázok 63). Z odhadu riadkov a stĺpcov je zjavné, že softvér priradil kvôli prechádzajúcemu svetlu cez predmety vzdialenosť pozadia, nie povrchu objektu. Na odhade riadkov a stĺpcov nie je pre časti okrem okrajov a písma, kde je zhoršená priehľadnosť, viditeľné posunutie, ktoré by odpovedalo objektom v priestore.

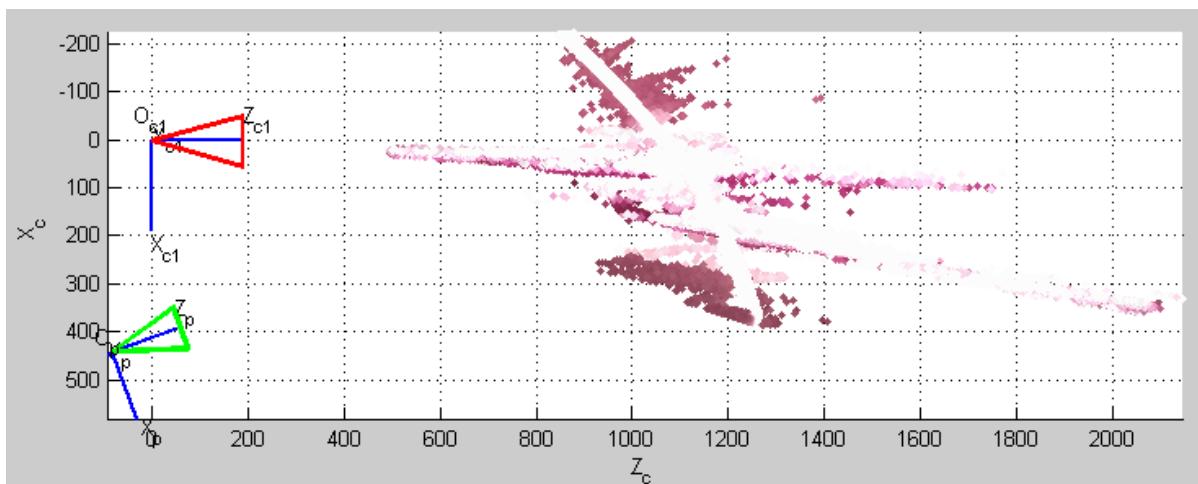


Obrázok 63: Získanie riadkov a stĺpcov pre rekonštrukciu priehľadných objektov

Na rekonštrukcii objektu z pohľadu kamery (Obrázok 64) nie je zjavný efekt nesprávneho určenia jednotlivých bodov v mračne bodov. Viditeľné to je až pri pohľade zhora alebo zboku (Obrázok 65), kde je vidieť nelogické rozmazanie a absolútne nesprávne zrekonštruovanie objektu.



Obrázok 64: Pohľad na zrekonštruovaný priehľadný objekt z pohľadu kamery



Obrázok 65: Pohľad na rekonštruovaný priehľadný objekt, pohľad z hora

### 5.3.6 Využitie práškových nástrekov na povrch

V praxi existuje niekoľko spôsobov, ako aspoň čiastočne potlačiť niektoré pre skenovanie nepriaznivé vlastnosti objektov ako sú práve lesklosť, priehľadnosť alebo prílišná farebná pohltivosť tmavých objektov. Jedna z najjednoduchších dostupných alternatív, ako môžeme práve tieto nedostatky odstrániť je aplikácia práškových nástrekov. Vzniká tu však závislosť kvality 3D skenu na hrúbke vrstvy nástreku, prípadne jej zrnitosti. Ak nanesieme príliš málo práškového nástreku, nepotlačíme dostatočne nepriaznivé vlastnosti a môže vzniknúť množstvo náhodných chýb. Ak použijeme príliš veľké množstvo, strácame presnosť. Najčastejšie sa používajú tri druhy práškových nástrekov:

*Titanový prášok (Oxid Titaničitý -  $TiO_2$ )* - Výhodou je malá veľkosť častíc. používa sa v kombinácii s rozpúšťadlom, napríklad ethanolom. poskytuje veľmi tenkú matniacu vrstvu. Cena prášku je vyššia ako v ostatných možnostiach.

*Kriedový prášok (Uhlíčan Vápenatý -  $CaCO_3$ )* - Patrí k najpoužívanejším matnaciim prostriedkom. Jednoduchý na použitie. Nevýhoda je väčšia vrstva nástreku, ktorá zhoršuje presnosť.

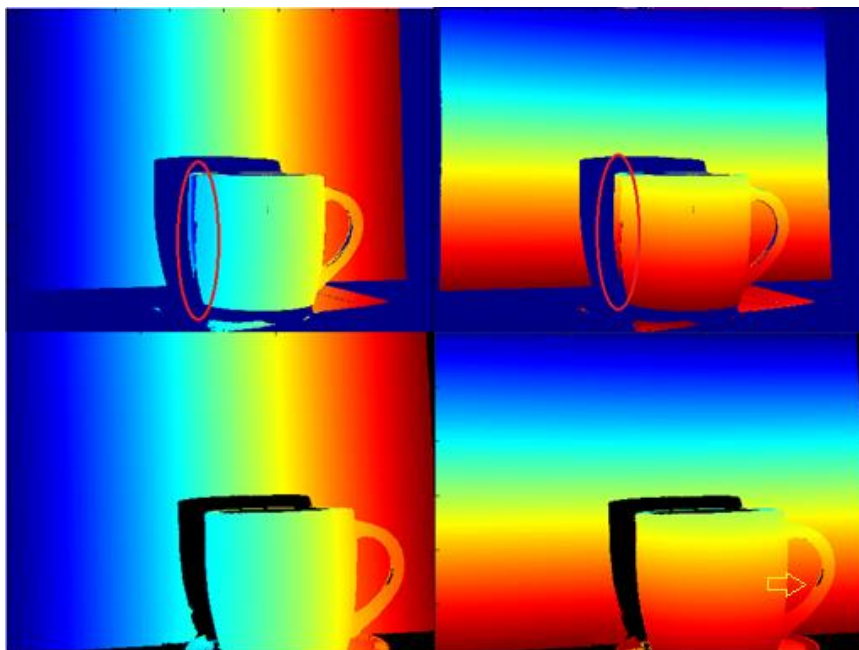
*Cyklododekanový prášok ( $C_{12}H_{24}$ )* - Jedná sa o organickú zlúčeninu. Jej vlastnosťou je samovoľná sublimácia, vďaka ktorej odpadá nutnosť čistenia objektu po nanesení matniacej vrstvy. [28]

Pre testovanie použiteľnosti práškového nástreku som využil magnéziový kriedový prášok v ethanole. Zmenu kvality 3D skenu za použitia matniaceho prášku som testoval na objektoch s nepriaznivou lesklosťou. Na obrázku 66 sa nachádza hrnček po aplikácii práškového náteru a pred. Rozdiel v lesklosti povrchu je viditeľný.



**Obrázok 66: Predmet po a pred nanosením práškového náteru**

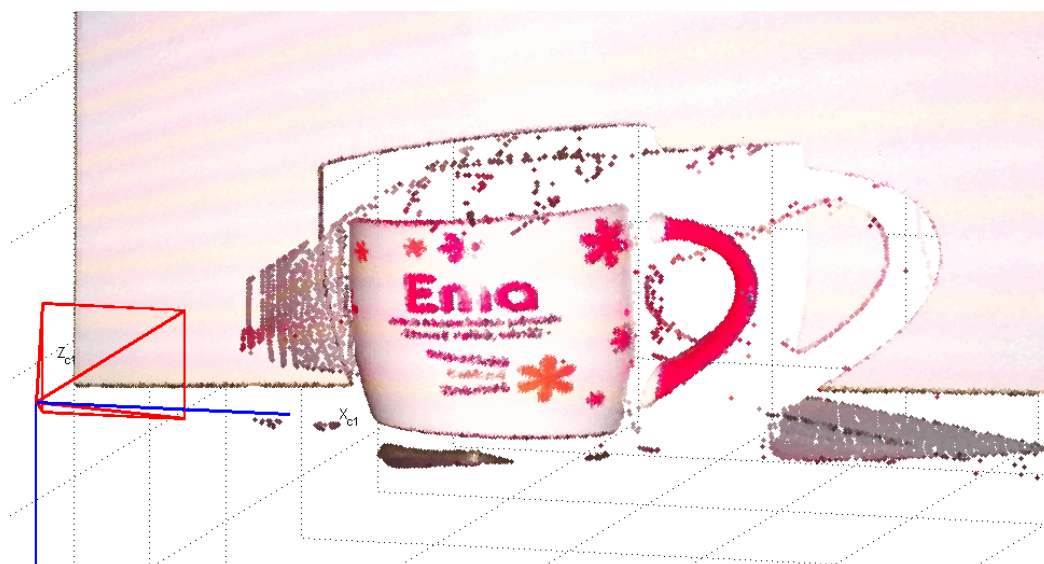
Pre oba prípady som zhotovil meranie, porovnávajúce ich vlastnosti. Na obrázku 67 je v hornej časti vidieť vypočítaná pozícia pixelov projektoru pre riadky a stĺpce, pred nanosením práškového náteru. V spodnej časti obrázku sa nachádza hrnček po nanosení práškového náteru. Nesprávne vypočítané pixely sú minimalizované pre prípad náteru. Na ušku hrnčeka, v mieste uchopenia je vidieť prejavenie zmazania časti práškového náteru.



**Obrázok 67: Výpočet riadkov a stĺpcov pre rekonštrukciu predmetu ošetrovaného práškovým náterom**

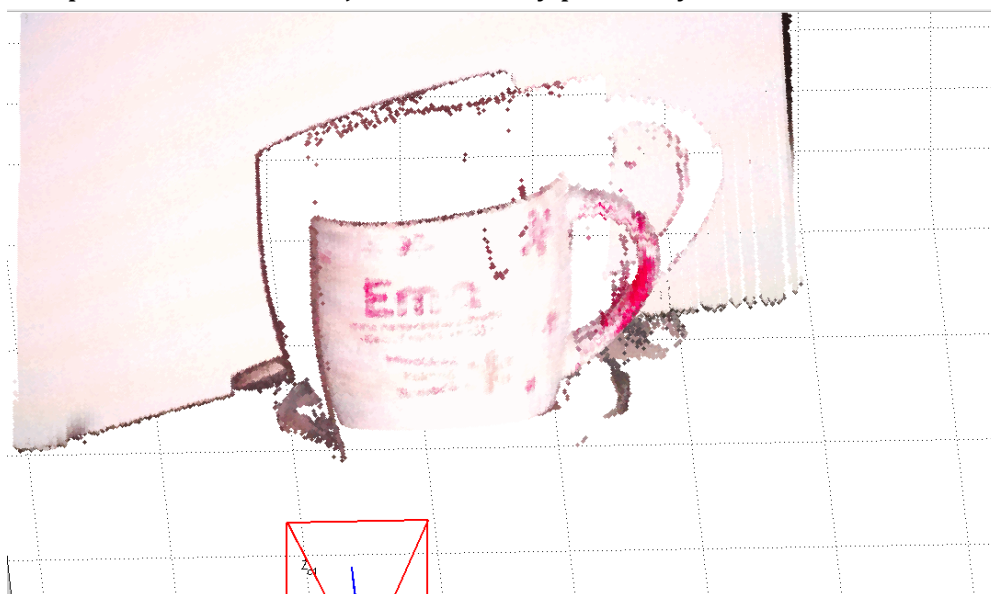
Vplyv lesklosti je viditeľný z rozdielu rekonštruovaných modelov pre oba prípady. V (Obrázok 65) sa prejavuje lesklosť strany hrnčeka, a program nesprávne vyhodnotil pixelovú polohu.





**Obrázok 68: Rekonštrukcia lesklého objektu bez práškového náteru**

V (Obrázok 69) sa rozptýlenie bodov v priestore skoro nevyskytuje. Podobných výsledkoch by sa dokázalo doceliť aj pre reflexné a čiastočne aj transparentné objekty. Pri transparentných objektoch sa bude pri bežných optických metódach chyba vyskytovať vždy. Odstrániť to môžeme jedným zo spôsobov komentovaným v časti testovania skenovania priehľadných objektov. Po skončení procesu skenovania je nutné všetky predmety s náterom očistiť.



**Obrázok 69: Rekonštrukcia lesklého objektu po nanesení práškového náteru**

## 6. Vyhodnotenie experimentov a identifikácia slabých miest metódy

V tejto práci som zahrnul výsledky testov pre niektoré hlavné oblasti, v ktorých skenovanie s využitím štruktúrovaného osvetlenia naráža na problémy. Jeden z hlavných prvkov pri zostavení modelu z reálneho objektu je jeho nasnímanie a následne spracovanie. Práve v tomto bode sa môžu vyskytnúť najzávažnejšie chyby. Nesprávna kalibrácia môže spôsobiť až absolútne scestné výsledky. Pritom samotná kalibrácia je veľmi zložitý a hlavne časovo náročný krok. 3D skenery využívajúce štruktúrované osvetlenie sa preto väčšinou vyrábajú ako statické systémy, bez možnosti pohybu jednotlivých častí voči sebe. Niektoré zložitejšie systémy schopné samokalibrácie môžu byť polohovateľné, ale stále je kalibrácia časovo náročný proces. Prístroj na získanie 3D modelu preto musí byť maximálne nehybný, aby sa minimalizovala nutnosť rekalibrácie. Pri uložení prístroja na robota vzniká vibráciami snaha pohyblivých častí meniť svoju pozíciu. Túto skutočnosť by sme mali brať na vedomie a v určitých časových intervaloch vyhotoviť preventívnu rekalibráciu.

Prebehlo overenie vizuálnych vlastností skenovaného objektu na kvalitu skenu. Samotné skenovanie lesklých, priehľadných alebo reflexných predmetov predstavuje pre optické metódy problém. Kompromis v kvalite skenu za cenu znečistenia výrobku sa dá docieľiť práškovými nátermi. V aplikáciách kde je zašpinenie neprípustné je však táto možnosť nevhodná. Z testov vyplýva vhodnosť využitia odlišnej metódy pre skenovanie týchto problematických predmetov alebo kombináciu viacerých metód. V prípade využitia štruktúrovaného osvetlenia treba myslieť na následnú úpravu modelu, kde je nutné ručne odstrániť chybné analyzované body. Pri potrebe spájania viacerých meshov pre vyhotovenie komplexného obrazu objektu zo všetkých strán je to však nevhodné z časového hľadiska. Algoritmizáciou odstraňovania chybných bodov je možné čiastočne automatizovať a urýchliť proces.

Analyzovaná metóda nie je robustná skrz problematickú kvalitu kalibrácie. Pri dobrej kvalite je aj výsledný model dostatočnej kvality. Pre plnohodnotné využitie tejto metódy sa predpokladá zlepšenie kalibračnej techniky systému a prepracovanie systému odlišovania čiernych objektov od tieňu.

## 7. Záver

Táto práca rozoberá metódy 3D skenovania, hlavne metódu využívajúcu štruktúrované osvetlenie. Hlavným cieľom tejto práce bolo skonštruovanie a naprogramovanie skenovacej hlavy, nasaditeľnej na mobilného robota. Táto skenovacia hlava, využívajúca štruktúrované osvetlenie, má byť diaľkovo ovládateľná a čo najviac automatizovaná.

Bakalárska práca je tvorená šiestimi hlavnými kapitolami. V prvej kapitole stručne rozoberám najpoužívanejšie metódy 3D skenovania. V druhej kapitole sa zameriavam na metódu 3D skenovania využívajúcu štruktúrované osvetlenie. Detailne rozoberám jednotlivé konštrukčné, mechanické a softvérové kľúčové vlastnosti skeneru. Štvrtá kapitola je venovaná prvotnému nastaveniu zariadení a ich rozmiestnení. V tejto kapitole je aj detailne rozobratá metóda kalibrácie kamery a projektoru. V štvrtej kapitole prezentujem mnou vytvorený program, umožňujúci ovládanie skeneru z počítaču. Program je schopný získať dáta pre kalibráciu kamery a projektoru, odtestovať samotnú kameru, zobrazíť kalibračné dáta, spracovať zaznamenané dáta a exportovať ich. Špeciálnu pozornosť si zaslúži programová súčasť, ktorá dovoľuje ovládanie robota a zahájenie procesu skenovania. Z predpokladu pripojiteľnosti na robota získavame možnosť ovládania samotného robota pomocou Raspberry Pi 3 model B s využitím nášho programu. Proces zberu dát dokáže premietat' programovo vytvorené štruktúrované vzory v Grayovom kóde a zaznamenávat' ich pomocou kamery. Piata kapitola tejto práce opisuje priebeh testov, určujúcich kvalitu skeneru.

V praktickej časti tejto práce som mal možnosť prejsť jednotlivými fázami nutnými k získaniu 3D modelu. V prvej časti bolo potrebné skalibrovať fotoaparát a projektor. Kalibrácia fotoaparátu prebehla pomocou *Jean-Yves Bouguet Calibration Toolbox*. Tento toolbox je veľmi rozšírený a prepracovaný, ponúka komplexný pohľad na hodnoty koeficientov skreslenia, tak aj hodnoty neistôt a pohľad na radiálne a tangenciálne skreslenie fotoaparátu. Následne sme vykonali kalibráciu projektoru, ktorá nám dovolila získať údaje o vzájomnej polohe a orientácii kamery a projektoru v priestore. Po získaní dát vo forme obrázkov, na ktorých je na skenovaný objekt premietaná séria vzorov, sme mohli dáta spracovať v programe Octave, stále na zariadení Raspberry Pi. Získali sme zhodu riadkov a stĺpcov pixelov projektoru a farebnú mapu, použitú pre prekrytie výsledného modelu. Po exporte sme boli schopný vizualizovať objekt v 3D. Práca ukazuje výhody a nedostatky skenovania touto metódou. Pre získanie širšieho obzoru v problematike 3D skenovania a v programovacom jazyku Python bola táto práca ideálnym učiteľom.

## Citácie

- [1] SKOUPÝ, PAVEL. *3D OPTICKÉ MERÍCÍ A SKENOVACÍ SYSTÉMY PRO STROJÍRENSTVÍ: OPTICAL MEASURING TECHNIQUES* [online]. Brno, 2007 [cit. 2017-02-09]. Dostupné z: [dl.uk.fme.vutbr.cz/zobraz\\_soubor.php?id=264](http://dl.uk.fme.vutbr.cz/zobraz_soubor.php?id=264). BAKALÁRSKÁ PRÁCE. VYSOKÉ UCENÍ TECHNICKÉ V BRNE. Vedoucí práce Ing. DAVID PALOUŠEK.
- [2] EREN, Gönen. *3D scanning of transparent objects* [online]. 2010 [cit. 2017-02-25]. Dostupné z: [https://tel.archives-ouvertes.fr/file/index/docid/584061/filename/these\\_A\\_EREN\\_Gonen\\_2010.pdf](https://tel.archives-ouvertes.fr/file/index/docid/584061/filename/these_A_EREN_Gonen_2010.pdf). Requirements for the degree of Docteur en Instrumentation et Informatique de l'Image. Université de Bourgogne.
- [3] D' APUZZO, Nicola. Overview of 3D surface digitization technologies in Europe. *Hometrica* [online]. San Jose (CA), USA., 2006, (6056) [cit. 2017-02-25]. DOI: 10.1117/12.650123. Dostupné z: [http://hometrica.ch/publ/2006\\_3dimg.pdf](http://hometrica.ch/publ/2006_3dimg.pdf)
- [4] ZHANG, Li, Brian CURLESS a Steven M. SEITZ. *Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming.: In Proceedings of the 1st International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT)* [online]. In: . June 19-21, 2002. Padova, Italy [cit. 2017-02-26].
- [5] ROCCHINI, C., P. CIGNONI, C. MONTANI, P. PINGI a R. SCOPIGNO. A low cost 3D scanner based on structured light. *Computer Graphics forum* [online]. Pisa, Italy: EUROGRAPHICS, 2001, (6056, Number 3), 299–308 [cit. 2017-03-01]. DOI: 10.1111/1467-8659.00522. Dostupné z: <http://onlinelibrary.wiley.com/doi/10.1111/1467-8659.00522/epdf>
- [6] ČERMÁK, JAN. *METODY 3D SKENOVÁNÍ OBJEKTŮ: 3D SCANNING METHODS* [online]. Brno, 2015 [cit. 2017-03-01]. Dostupné z: <https://core.ac.uk/download/pdf/30294566.pdf>. BAKALÁRSKÁ PRÁCE. VYSOKÉ UCENÍ TECHNICKÉ V BRNE. Vedoucí práce ING. TOMÁŠ MARADA, PH.D.
- [7] CUCCARO, A., A. BRANCACCIO, B. BASILE, M. J. AMMANN, R. SOLIMENE a G. RUVIO. 3-D Microwave Scanner for Biomedical Applications: A Preliminary Prototype. *PIERS Proceedings* [online]. Prague, Czech Republic: The Electromagnetics Academy, 2015, , 1502-1505 [cit. 2017-03-08]. ISSN 1559-9450. Dostupné z: <https://piers.org/piersproceedings/download.php?file=cGllcnMyMDE1UHJhZ3VlfdJQNmJfMTUwMi5wZGZ8MTUwMjA1MDcyMDUw>
- [8] *Chapter 4 Microwave sensors* [online]. [cit. 2017-03-08]. Dostupné z: [www.jars1974.net/pdf/05\\_Chapter04.pdf](http://www.jars1974.net/pdf/05_Chapter04.pdf)



- [9] Y. CHEN, S., Y. F. LI a Jianwei ZHANG. Vision Processing for Realtime 3-D Data Acquisition Based on Coded Structured Light. *IEEE Transactions on Image Processing* [online]. IEEE, **2008**(17), 167 - 176 [cit. 2017-03-15]. DOI: 10.1109/TIP.2007.914755. ISSN 1941-0042. Dostupné z: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4429304>
- [10] DOKUPIL, Filip. *STANOVENÍ ODCHYLEK MERENÍ 3D OPTICKÉHO SKENERU* [online]. Brno, 2013 [cit. 2017-03-21]. Dostupné z: [https://www.vutbr.cz/www\\_base/zav\\_prace\\_soubor\\_verejne.php?file\\_id=67627](https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=67627). BAKALÁRSKÁ PRÁCE. VYSOKÉ UCENÍ TECHNICKÉ V BRNE. Vedoucí práce Ing. DAVID PALOUŠEK, Ph.D.
- [11] HERAKLEOUS, Kyriakos a Charalambos POULLIS. *3DUNDERWORLD-SLS: An Open-Source Structured-Light Scanning System for Rapid Geometry Acquisition* [online]. In: . June 26, 2014. Immersive and Creative Technologies Lab, Cyprus University of Technology [cit. 2017-03-21]. Dostupné z: <https://arxiv.org/pdf/1406.6595>
- [12] ONDRIČKA, Matej. *Rýchlý vstup dat pre Raspberry PI: High Speed Data Input for Raspberry PI* [online]. Praha, 2014 [cit. . 2017-03-22]. Dostupné z: <https://dspace.cvut.cz/bitstream/handle/10467/24193/F3-BP-2014-Ondricka-Matej-prace.pdf>. Bakalárska práca. ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAHE. Vedoucí práce Doc. Ing. Jan Fischer, CSc.
- [13] *Acer Projector K11 Series: User's Guide*. 04/2010. Regulation Center, Acer Inc. Taipei Hsien 221, Taiwan, 2010. [cit. . 2017-03-23]. Dostupné z: [http://www.projectorcentral.com/pdf/projector\\_manual\\_5573.pdf](http://www.projectorcentral.com/pdf/projector_manual_5573.pdf)
- [14] LANMAN, Douglas a Gabriel TAUBIN. *Build Your Own 3D Scanner: 3D Photography for Beginners* [online]. In: . 2009 [cit. . 2017-03-30]. Dostupné z: [mesh.brown.edu/byo3d/notes/byo3D.pdf](http://mesh.brown.edu/byo3d/notes/byo3D.pdf)
- [15] ZHANG, Song a Peisen S. HUANG. Novel method for structured light system calibration. In: *Society of Photo-Optical Instrumentation Engineers* [online]. Aug. 21, 2006. 2006 [cit. . 2017-04-06]. DOI: 10.1117/1.2336196. Dostupné z: <https://engineering.purdue.edu/ZhangLab/publications/papers/2006-oe-calib.pdf>
- [16] Camera Calibration Toolbox for Matlab: Jean-Yves Bouguet. *Computational Vision* [online]. 2015 [cit. 2017-04-09]. Dostupné z: [http://www.vision.caltech.edu/bouguetj/calib\\_doc/htmls/parameters.html](http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/parameters.html)

- [17] FALCAO, Gabriel, Natalia HURTOS a Joan MASSICH. *Plane-based calibration of a projector-camera system* [online]. Edinburgh, Scotland, 2008 [cit. 2017-04-26]. Dostupné z: [https://www.researchgate.net/profile/Joan\\_Massich/publication/265230871\\_Plane-based\\_calibration\\_of\\_a\\_projector-camera\\_system/links/54db26570cf261ce15cf4cd1.pdf?origin=publication\\_detail](https://www.researchgate.net/profile/Joan_Massich/publication/265230871_Plane-based_calibration_of_a_projector-camera_system/links/54db26570cf261ce15cf4cd1.pdf?origin=publication_detail)
- [18] ADA, Lady. Setting up a Raspberry Pi as a WiFi access point. In: *Adafruit Industries* [online]. 2013-06-12. 2013 [cit. 2017-04-26]. Dostupné z: <https://cdn-learn.adafruit.com/downloads/pdf/setting-up-a-raspberry-pi-as-a-wifi-access-point.pdf>
- [19] *BPRP - Robotika a počítačové vidění* [online]. [cit. 2017-04-26]. Dostupné z: <https://sites.google.com/a/vutbr.cz/bprp/>
- [20] The OpenCV Tutorials: Release 2.4.13.2. In: *GENERAL TUTORIALS* [online]. May 18, 2017. [cit. 2017-05-01]. Dostupné z: [http://docs.opencv.org/2.4/opencv\\_tutorials.pdf](http://docs.opencv.org/2.4/opencv_tutorials.pdf)
- [21] *GNU Octave - Free Your Numbers: A high-level interactive language for numerical computations* [online]. In: EATON, John W., David BATEMAN, Søren HAUBERG a Rik WEHBRING. 2017 [cit. 2017-05-01]. Dostupné z: <https://www.gnu.org/software/octave/octave.pdf>
- [22] *Automation.com: Fundamentals of Photoelectric Sensors* [online]. [cit. 2017-05-12]. Dostupné z: <https://www.automation.com/library/articles-white-papers/sensors-sensing-technologies/fundamentals-of-photoelectric-sensors>
- [23] KŘEMEN, Tomáš , Ing. Ph.D. *Úvod do 3D skenování* [online]. [cit. 2017-05-12]. Dostupné z: <http://www.kgk.cz/wp-content/uploads/2014/06/Kremen-uvod-do-skenovani.pptx>. Katedra speciální geodézie , Fakulta stavební, ČVUT v Praze.
- [24] In: *DAVID SLS-2 Tutorial: v1.2* [online]. 2017 [cit. 2017-05-12]. Dostupné z: <https://productrealization.stanford.edu/content/05-resources/02-processes/09-3d-scanning/3D%20Scanner%20Tutorial.pdf>
- [25] JOHNSON, Robyn L., Gretchen Stahmer DEMOSS a Richard SORENSEN. *Earth Science with Vernier: Reflection and Absorption of Light* [online]. Vernier Software & Technology , Beaverton [cit. 2017-05-16]. ISBN 978-1-929075-46-1. Dostupné z: [http://www2.vernier.com/sample\\_labs/ESV-23-COMP-reflection\\_absorption\\_of\\_light.pdf](http://www2.vernier.com/sample_labs/ESV-23-COMP-reflection_absorption_of_light.pdf)

- [26] Structured light 3D scanning in the presence of global illumination. In: GUPTA, Mohit, Amit AGRAWAL, Ashok VEERARAGHAVAN a Srinivasa G. NARASIMHAN. *Proceedings / CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Conference on Computer Vision and Pattern Recognition* [online]. 2011 [cit. 2017-05-18]. DOI: 10.1109/CVPR.2011.5995321. Dostupné z: [https://www.researchgate.net/profile/Ashok\\_Veeraraghavan/publication/24254728\\_Structured\\_light\\_3D\\_scanning\\_in\\_the\\_presence\\_of\\_global\\_illumination/links/573b30c908ae9f741b2d7a7c/Structured-light-3D-scanning-in-the-presence-of-global-illumination.pdf](https://www.researchgate.net/profile/Ashok_Veeraraghavan/publication/24254728_Structured_light_3D_scanning_in_the_presence_of_global_illumination/links/573b30c908ae9f741b2d7a7c/Structured-light-3D-scanning-in-the-presence-of-global-illumination.pdf)
- [27] GENG, Jason. Structured-light 3D surface imaging: a tutorial. *IEEE Intelligent Transportation Systems Society* [online]. 11001 Sugarbush Terrace, Rockville, Maryland 20852, USA, 2011 [cit. 2017-05-20]. DOI: 10.1364/AOP.3.000128. Dostupné z: <http://grail.cs.washington.edu/projects/moscan/paper.pdf>
- [28] PALOUSEK David, Milan OMASTA, Daniel KOUTNY, Josef BEDNAR, Tomas KOUTECKY a Filip DOKOUPIL. Effect of matte coating on 3D optical measurement accuracy. *Optical Materials* [online]. 2015, 40, 1-9 [cit. 2017-05-20]. DOI: 10.1016/j.optmat.2014.11.020. ISSN 09253467. Dostupné z: <http://linkinghub.elsevier.com/retrieve/pii/S0925346714005606>
- [29] ADAMS, James. Raspberry Pi model B: Dimensions. In: *Raspberry Pi* [online]. 2014 [cit. 2017-05-24]. Dostupné z: <https://www.raspberrypi.org/app/uploads/2014/07/mechanicalspecB+.png>

## Zoznam použitých skratiek, symbolov a výrazov

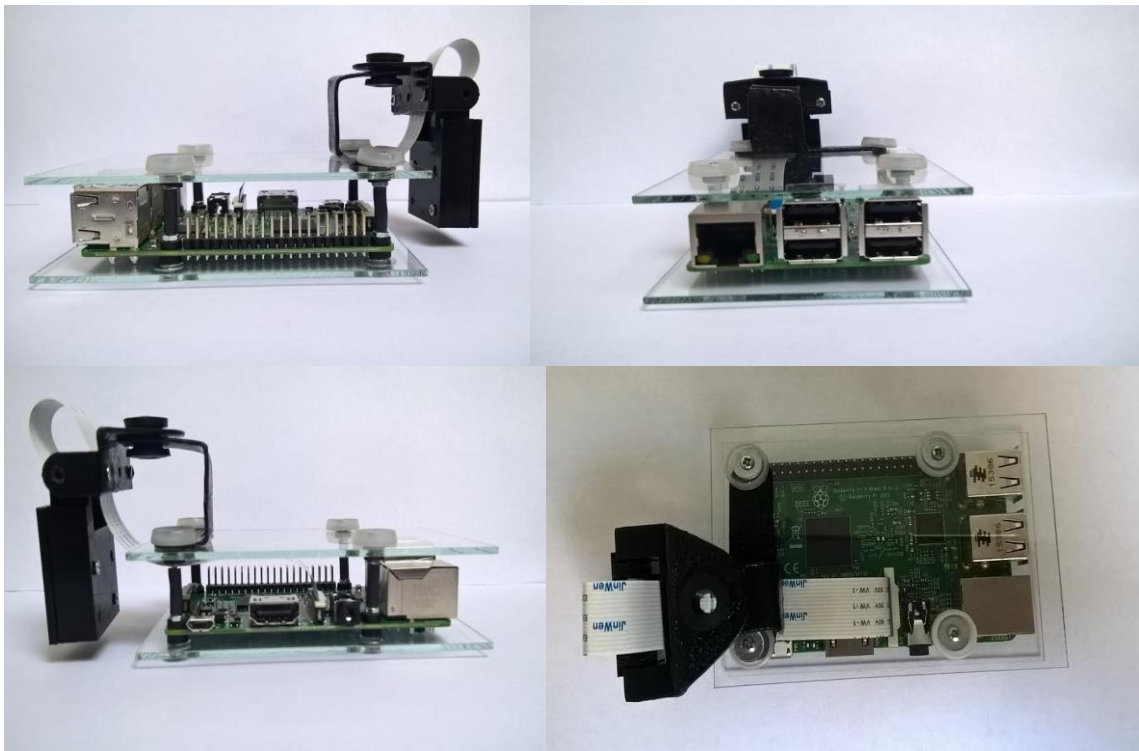
Downsampling	- redukcia dátových vzoriek
Point cloud	- mračno bodov
Mesh	- pospájanie bodov v mrečne bodov
Spline	- interpolácia povrchu meshu pre dosiahnutie hladšieho povrchu
reverse engineering	- reverzné inžinierstvo
RPi	- Raspberry Pi
Escape sekvencia	- sekvencia znakov, dovoľujúca zmenu chovania program, napríklad predčasné ukončenie programu
OpenCV	- knižnica programovacích funkcií zameraná hlavne na počítačové videnie
GPIO	- (General-purpose input/output) vstupy a výstupy na integrovanom obvode
OS	- operačný systém
CCD	- (charge-coupled device) zariadenie určené na zachytávanie obrazu
CMOS	- (Complementary Metal–Oxide–Semiconductor) podobne ako CCD na výrobu obrazových senzorov

## Zoznam príloh

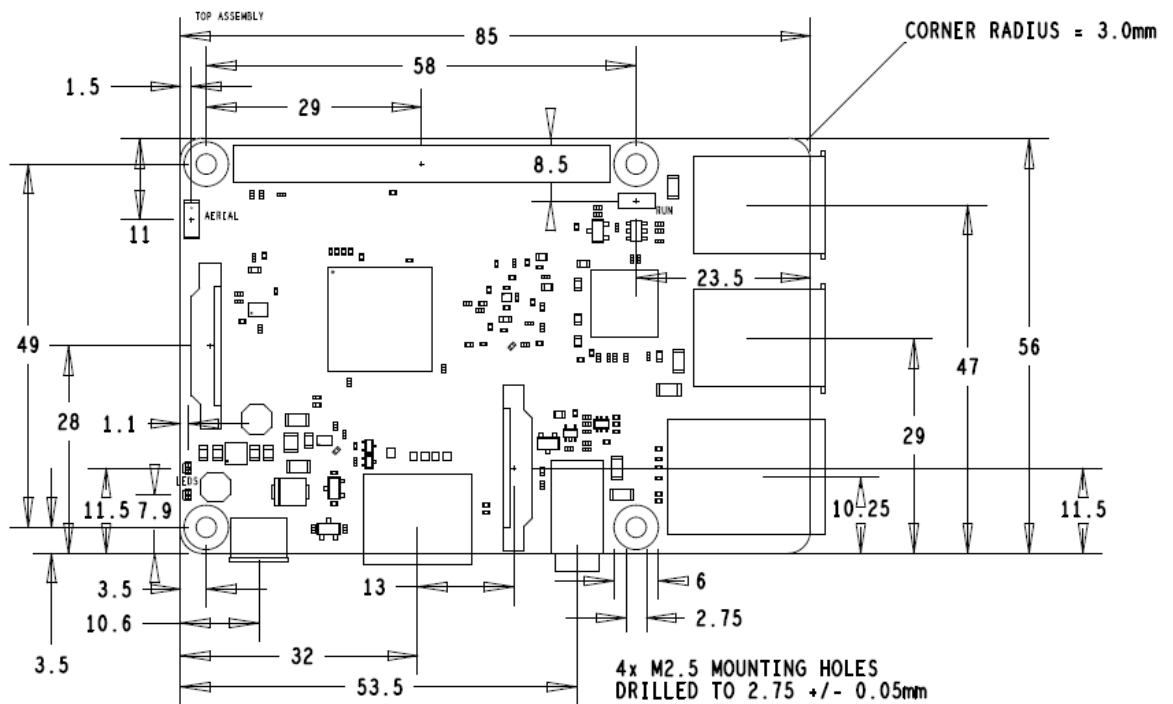
Príloha 1: Mechanická konštrukcia pre Raspberry a kameru.....	78
Príloha 2: Rozmerové špecifikácie Raspberry Pi 3 model B [29] .....	78
Príloha 3: Konštrukcia – pohľad zhora .....	79
Príloha 4: Pripojiteľnosť projektoru ku konštrukcii .....	79
Príloha 5: Polohovateľnosť ramena a možnosť pohybu koľajničiek .....	80
Príloha 6: Ukážka kalibrácie projektoru .....	80
Príloha 7: Spracovanie priloženej dát pre BYO3D Scanner[14] .....	81

# Prílohy

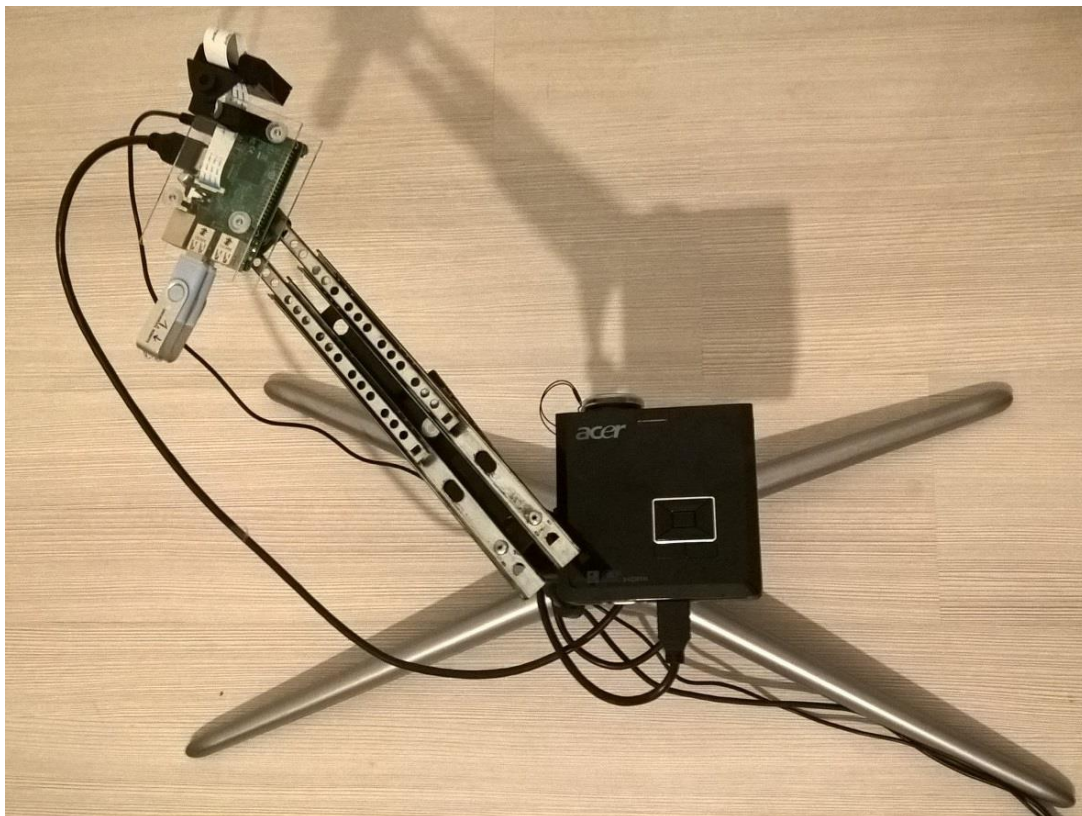
## Príloha 1: Mechanická konštrukcia pre Raspberry a kameru



## Príloha 2: Rozmerové špecifikácie Raspberry Pi 3 model B [29]



### Príloha 3: Konštrukcia - pohľad zhora

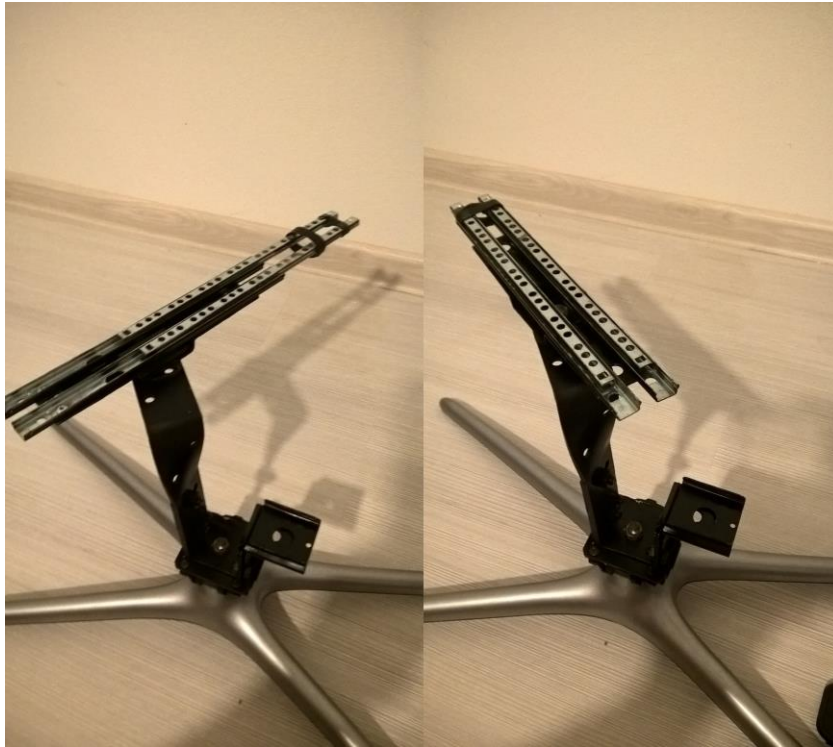


### Príloha 4: Pripojiteľnosť projektoru ku konštrukcii





**Príloha 5: Polohovateľnosť ramena a možnosť pohybu kol'ajničiek**

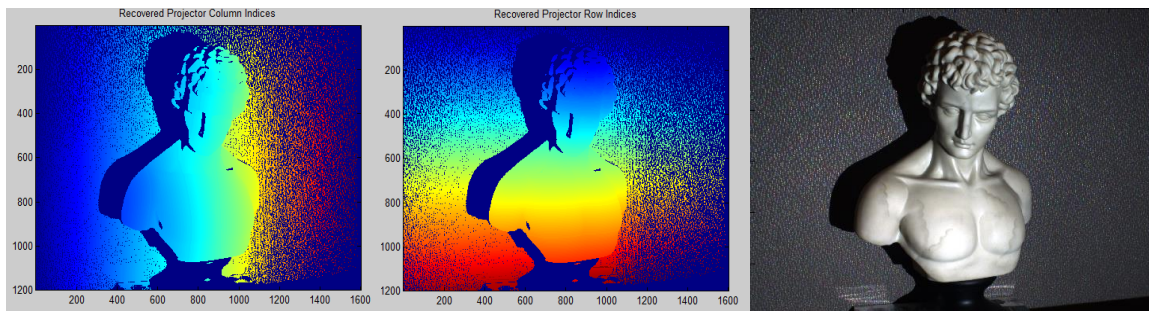


**Príloha 6: Ukážka kalibrácie projektoru**

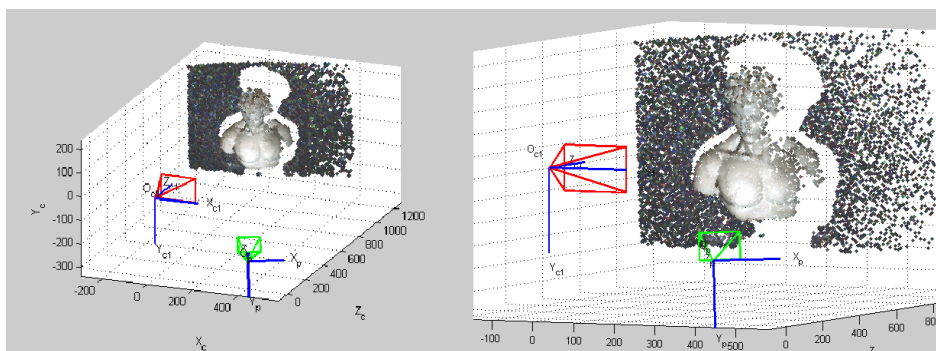




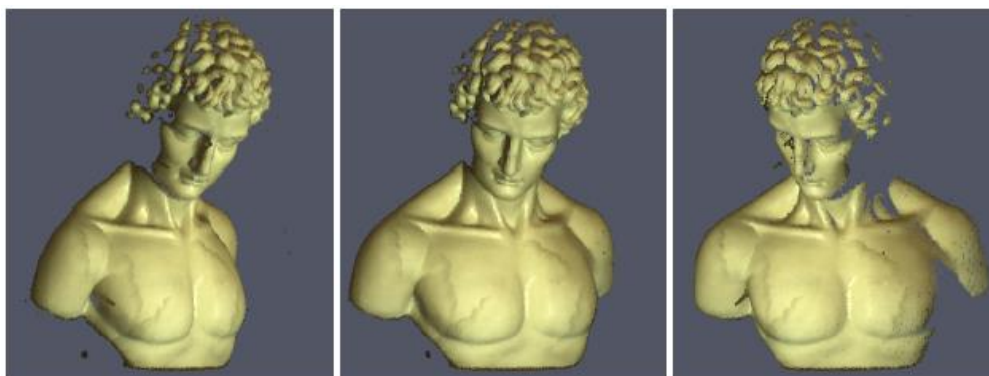
## Príloha 7: Spracovanie priloženej dát pre BYO3D Scanner[14]



Obr.1: Nájdená zhoda stĺpcov a riadkov pixelov projektora, ukážka skenovaného objektu



Obr.2: 3D vizualizácia v prostredí Matlab



Obr.3: Ukážka spracovania v programe typu CAD