

# **Konstrukce 3D skeneru a analýza modelu naskenovaného objektu**

**Diplomová práce**

**Vedoucí práce:**

**Dr. Ing. Radovan Kukla**

**Bc. Petr Palko**

**Brno 2016**



Chtěl bych tímto poděkovat mému učiteli a vedoucímu mé práce panu Dr. Ing. Radovanu Kuklovi za odborné vedení při této práci. Dále bych chtěl poděkovat Ing. Michalu Hammerschmiedtovi za poskytnuté konzultace při tvorbě této práce. Dále bych chtěl poděkovat Janu Čopákovi za poskytnutí technických rad při tvorbě této práce.



## Čestné prohlášení

Prohlašuji, že jsem tuto práci: **Konstrukce 3D skeneru a analýza modelu naskenovaného objektu**

vypracoval samostatně a veškeré použité prameny a informace jsou uvedeny v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů, a v souladu s platnou *Směrnicí o zveřejňování vysokoškolských závěrečných prací*.

Jsem si vědom/a, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 Autorského zákona.

Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity o tom, že předmetná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

V Brně dne 1. prosince 2016

---



## **Abstract**

Palko, P. 3D scanner design and model analysis of the scanned object. Diploma thesis. Brno: Mendel University, 2016.

The work introduces the basic principles of 3D scanning technology. In the work describes the basic hardware components required to construct a 3D scanner and implement applications processing the scanned data. The work also includes an analysis of the scanned object.

## **Keywords**

Technology of 3D scanning, hardware components, application, analysis of the scanned object.

## **Abstrakt**

Palko, P. Konstrukce 3D skeneru a analýza modelu naskenovaného objektu. Diplomová práce. Brno: Mendelova univerzita v Brně, 2016.

Práce seznamuje se základními principy technologie 3D skenování. V práci jsou popsány základní hardwarové prvky nutné ke konstrukci 3D skeneru a realizaci aplikace zpracovávající naskenovaná data. Práce dále obsahuje analýzu naskenovaného objektu.

## **Klíčová slova**

Technologie 3D skenování, hardwarové prvky, aplikace, analýza naskenovaného objektu.





# Obsah

<b>1</b>	<b>Úvod a cíl práce</b>	<b>19</b>
1.1	Úvod.....	19
1.2	Cíl práce.....	19
<b>2</b>	<b>Rozbor aktuálního stavu informací o technologii 3D skeneru</b>	<b>20</b>
2.1	Rozdělení podle typu a užití.....	20
2.1.1	Principy digitalizace.....	21
2.2	Rozdělení 3D skenerů podle způsobu a metody měření .....	21
2.2.1	Mechanické 3D skenery.....	21
2.2.2	Optické 3D skenery .....	22
2.2.3	Laserové 3D skenery.....	22
2.2.4	Destruktivní 3D skenery.....	22
2.2.5	Ultrazvukové 3D skenery.....	22
2.2.6	Rentgenové 3D skenery.....	22
2.3	Rozdělení 3D skenerů podle způsobu skenování .....	23
2.3.1	Polární laserové skenery.....	23
2.3.2	Základnové skenovací systémy .....	23
2.4	Ukázky 3D skenerů .....	27
2.4.1	Víceúčelové polární skenery.....	27
2.4.2	Víceúčelové triangulační skenovací systémy. ....	27
2.4.3	Speciální a kinematické systémy.....	28
2.5	Způsoby měření .....	29
2.6	Metody měření .....	30
2.6.1	Triangulační metody.....	30
<b>3</b>	<b>Analýza možnosti zpracování grafických dat</b>	<b>35</b>
3.1	Snímače obrazu.....	36
3.2	Rozdělení snímačů.....	37
3.2.1	CCD (Charge-Coupled Device).....	37
3.2.2	Super CCD.....	39

---

3.2.3	CMOS (Complementary Metal Oxide Semiconductor) .....	39
<b>4</b>	<b>Návrh technického řešení 3D skeneru</b>	<b>41</b>
4.1	Nosná konstrukce .....	41
4.1.1	Stavebnice Merkur .....	42
4.2	Laserová dioda .....	43
4.2.1	Bezpečnostní třídy laserů .....	44
4.2.2	Krokový motor .....	44
4.2.3	Raspberry Pi .....	46
4.2.4	GPIO .....	47
4.2.5	Motorový shield pro Raspberry Pi .....	48
4.2.6	Nepájivé pole .....	49
4.2.7	Kamera .....	50
4.2.8	Pygame .....	50
<b>5</b>	<b>Aplikace 3D skeneru</b>	<b>52</b>
5.1	Program pro řízení 3D skeneru .....	52
5.1.1	Řízení krokového motoru .....	52
5.1.2	Řízení laserů .....	52
5.2	Metodika skenování .....	53
5.2.1	Nastavení kalibračních parametrů .....	54
5.2.2	Inicializační proměnné .....	55
5.2.3	Běh aplikace .....	55
<b>6</b>	<b>Analýza naskenovaného objektu</b>	<b>62</b>
6.1	Světelné podmínky .....	62
6.2	Odrazivost snímaného objektu .....	62
6.3	Barva snímaného objektu .....	63
6.4	Filtry .....	63
6.5	Zobrazení objektu v programu Matlab .....	65
6.5.1	Ukázka reálného objektu a mračna bodů .....	66
6.5.2	Ukázka modelu s vygenerovaným povrchem .....	67
<b>7</b>	<b>Skenování tvarově členitějšího objektu</b>	<b>69</b>

Obsah	11
<b>8 Diskuze</b>	<b>73</b>
<b>9 Závěr</b>	<b>75</b>
<b>10 Literatura</b>	<b>76</b>
<b>A Schéma zapojení motorového shieldu a jeho periférií</b>	<b>79</b>
<b>B 3D skener</b>	<b>84</b>
<b>C Metodika skenování</b>	<b>85</b>
<b>D UML schéma aplikace</b>	<b>86</b>
<b>E Ukázky z analýzy</b>	<b>87</b>
<b>F Tvarově členitější objekt</b>	<b>94</b>
<b>G Ukázka kódu aplikace</b>	<b>102</b>



## Seznam obrázků

<b>Obr. 1</b>	<b>Ukázka principu laserové triangulace</b>	
	<b>Zdroj: (Urban, 2010)</b>	<b>24</b>
<b>Obr. 2</b>	<b>Princip použití strukturovaného světla – pruhů</b>	
	<b>Zdroj: (Urban, 2010)</b>	<b>25</b>
<b>Obr. 3</b>	<b>Ukázka vzorů pro metodu strukturovaného světla</b>	
	<b>Zdroj: (Koska, 2009)</b>	<b>26</b>
<b>Obr. 4</b>	<b>Ukázka panoramatického skeneru Faro Photon 80</b>	
	<b>Zdroj: (Štroner, 2009)</b>	<b>27</b>
<b>Obr. 5</b>	<b>Ukázka skeneru Callidus CT 900</b>	
	<b>Zdroj: (Štroner, 2009)</b>	<b>28</b>
<b>Obr. 6</b>	<b>Ukázka ručního skeneru Creafom Handyscan 700</b>	
	<b>Zdroj: (Creafom, 2016)</b>	<b>29</b>
<b>Obr. 7</b>	<b>Triangulační trojúhelník (1D triangulace)</b>	
	<b>Zdroj: (Kalová, 2005)</b>	<b>31</b>
<b>Obr. 8</b>	<b>Problém viditelnosti u triangulačních metod (2D triangulace)</b>	
	<b>Zdroj: (Kalová, 2005)</b>	<b>32</b>
<b>Obr. 9</b>	<b>Technika světelného vzoru (3D triangulace)</b>	
	<b>Zdroj: (Kalová, 2005)</b>	<b>33</b>
<b>Obr. 10</b>	<b>Schéma znázorňující rozdělení jednotlivých snímačů obrazu</b>	
	<b>Zdroj: (Vojáček, 2008)</b>	<b>37</b>
<b>Obr. 11</b>	<b>Rozložení barevného filtru na barevném CCD snímači dle Bayerovy masky</b>	
	<b>Zdroj: (Vojáček, 2008)</b>	<b>39</b>
<b>Obr. 12</b>	<b>Zhotovený 3D skener</b>	<b>42</b>
<b>Obr. 13</b>	<b>Laserová dioda</b>	<b>44</b>
<b>Obr. 14</b>	<b>Krokový motor Microcon SX17-1003LQCEF</b>	<b>45</b>
<b>Obr. 15</b>	<b>Schéma jednotlivých částí mikropočítače Raspberry Pi</b>	
	<b>Zdroj: (Raspberrypi.org, 2016)</b>	<b>47</b>

<b>Obr. 16</b>	<b>Základní deska mikropočítače Raspberry Pi 3</b>	
	<b>Zdroj: (Raspberrypi.org, 2016)</b>	<b>47</b>
<b>Obr. 17</b>	<b>Schéma rozložení pinů rozhraní GPIO</b>	
	<b>Zdroj: (Raspberrypi.org, 2016)</b>	<b>48</b>
<b>Obr. 18</b>	<b>Motorový shield pro Raspberry Pi</b>	<b>49</b>
<b>Obr. 19</b>	<b>Nepájivé pole s elektrotechnickými součástkami</b>	<b>50</b>
<b>Obr. 20</b>	<b>Kamera Raspberry Pi s rozhraním CSI</b>	
	<b>Zdroj: (Raspberrypi.org, 2016)</b>	<b>50</b>
<b>Obr. 21</b>	<b>Schéma metodiky skenování</b>	<b>54</b>
<b>Obr. 22</b>	<b>Ukázka vymezení skenovací plochy s linií nasnímaných pixelů</b>	<b>58</b>
<b>Obr. 23</b>	<b>Ukázka vygenerovaných souřadnic znázorněných v 2D obraze</b>	<b>62</b>
<b>Obr. 24</b>	<b>Ukázka vygenerovaných souřadnic znázorněných v 2D obraze</b>	<b>63</b>
<b>Obr. 25</b>	<b>Ukázka nahodilé detekce a vygenerovaných souřadnic</b>	<b>63</b>
<b>Obr. 26</b>	<b>Ukázka vygenerovaných souřadnic bez užití filtrů</b>	<b>64</b>
<b>Obr. 27</b>	<b>Ukázka vygenerovaných souřadnic s užitím Filtru 1 a Filtru 2</b>	<b>64</b>
<b>Obr. 28</b>	<b>Ukázka vygenerovaných souřadnic s užitím všech filtrů</b>	<b>65</b>
<b>Obr. 29</b>	<b>Porovnání reálného objektu s mračny bodů</b>	<b>66</b>
<b>Obr. 30</b>	<b>Ukázka modelu s vygenerovaným povrchem</b>	<b>67</b>
<b>Obr. 31</b>	<b>Ukázka tvarově členitějšího objektu osazeném na podstavci</b>	<b>69</b>
<b>Obr. 32</b>	<b>Ukázka detekce bodů podél linie laseru</b>	<b>70</b>
<b>Obr. 33</b>	<b>Ukázka vygenerovaných souřadnic s jasně viditelnými ústřely znázorněných v 2D obraze</b>	<b>70</b>
<b>Obr. 34</b>	<b>Prezentace mračen bodů</b>	<b>71</b>
<b>Obr. 35</b>	<b>Prezentace povrchu modelu vytvořeném z mračna bodů</b>	<b>71</b>

---

<b>Obr. 36</b>	<b>Schéma zapojení motorového shieldu a bloku řízení laserů</b>	<b>79</b>
<b>Obr. 37</b>	<b>Schéma zapojení konvertoru se spínacím relé</b>	<b>80</b>
<b>Obr. 38</b>	<b>Schéma zapojení vlastního modulu konvertoru s jednotkou LM2596-ADJ</b>	<b>81</b>
<b>Obr. 39</b>	<b>Schéma zapojení Raspberry Pi Zdroj: (Raspberrypi.org, 2016)</b>	<b>82</b>
<b>Obr. 40</b>	<b>Blokové schéma Raspberry Pi 3</b>	<b>83</b>
<b>Obr. 41</b>	<b>Zhotovený 3D skener</b>	<b>84</b>
<b>Obr. 42</b>	<b>Schéma metodiky skenování</b>	<b>85</b>
<b>Obr. 43</b>	<b>UML schéma aplikace</b>	<b>86</b>
<b>Obr. 44</b>	<b>Ukázka vygenerovaných souřadnic bez užití filtrů</b>	<b>87</b>
<b>Obr. 45</b>	<b>Ukázka vygenerovaných souřadnic s užitím Filtru 1 a Filtru 2</b>	<b>88</b>
<b>Obr. 46</b>	<b>Ukázka vygenerovaných souřadnic s užitím všech filtrů</b>	<b>89</b>
<b>Obr. 47</b>	<b>Ukázka mračna bodů pohled z boku</b>	<b>90</b>
<b>Obr. 48</b>	<b>Ukázka mračna bodů pohled ze strany z vrchu</b>	<b>91</b>
<b>Obr. 49</b>	<b>Ukázka mračna bodů pohled ze shora</b>	<b>92</b>
<b>Obr. 50</b>	<b>Ukázka modelu s vygenerovaným povrchem</b>	<b>93</b>
<b>Obr. 51</b>	<b>Tvarově složitější objekt pohled zepředu</b>	<b>94</b>
<b>Obr. 52</b>	<b>Tvarově složitější objekt pohled z boku</b>	<b>95</b>
<b>Obr. 53</b>	<b>Tvarově složitější objekt pohled ze shora</b>	<b>96</b>
<b>Obr. 54</b>	<b>Tvarově složitější objekt usazený na podstavci</b>	<b>97</b>
<b>Obr. 55</b>	<b>Ukázka vygenerovaných souřadnic v 2D zobrazení</b>	<b>98</b>
<b>Obr. 56</b>	<b>Ukázka mračna bodů pohled ze shora</b>	<b>99</b>
<b>Obr. 57</b>	<b>Ukázka mračna bodů pohled z boku</b>	<b>99</b>
<b>Obr. 58</b>	<b>Ukázka mračna bodů pohled z boku ze shora</b>	<b>100</b>

<b>Obr. 59</b>	<b>Ukázka mračna bodů pohled z boku zespoda</b>	<b>100</b>
<b>Obr. 60</b>	<b>Ukázka povrchu modelu vytvořeném z mračna bodů</b>	<b>101</b>



## Seznam zkratek

**CMOS** (doplňující se kov-oxid-polovodič) je technologie používána na převážnou většinu integrovaných obvodů. Používá se na výrobu čipů včetně mikroprocesorů, jednočipových počítačů.

**CCD** (Charge-Coupled Device) je elektronická součástka používaná pro snímání obrazové informace.

**OKS** (optické korelační systémy) má stejný význam jako fotogrammetrie.

**PPS** (Passive-pixel sensors) - pasivní senzor pro CMOS.

**APS** (Active-pixel sensors) - aktivní senzor pro CMOS.

### Orientace v textu

**Příklad kódu** - takto jsou v práci značeny ukázky kódu

*Procedura* - takto jsou v práci značeny názvy procedur, funkcí a proměnných



# 1 Úvod a cíl práce

## 1.1 Úvod

V dnešní uspěchané době, kdy je vše rychlé a zároveň pomíjivé, se může velice snadno stát, že vše, co jsme jako lidstvo vytvořili, může být nenávratně zničeno. Zde se nabízí otázka, jak tyto vytvořené hodnoty, cenné předměty a jiné věci zachovat i pro budoucí generace? Toto je hnací impuls, který vede vědce a vynálezce k vytvoření takových zařízení, které by byly schopny replikovat předměty v takové podobě a s takovou přesností jakou byla originální předloha. Jak už však dějiny ukázaly v mnoha jiných případech, tak samotná myšlenka nestačí. Je pouze jednou stranou mince, kde vlastní realizace a skutečnost je stranou druhou. Technologie 3D skenování jde ruku v ruce s technologií 3D tisku, který zažívá v dnešní době značný boom. Jak je to krásné si něco naskenovat a poté si to vytisknout. Toto již znala naše generace otců a dědů. Bylo tu však podstatné omezení, že šlo pouze o reprodukování ve 2D. Další rozměr, umožňuje více tyto produkty přiblížit k účelnosti, kdy se už na ně jen nedíváme nebo nečteme, ale můžeme je uchopit do rukou a zapojit tak náš další smysl - hmat. Posouvá nás to dál v chápání světa a jeho utváření. Stojíme tak dnes na prahu nové průmyslové revoluce, kdy si i obyčejný člověk může sám vytvářet a reprodukovat věci potřebné k životu prostřednictvím moderních 3D technologií. Dále nám tyto technologie umožňují, zachovávat cenné nebo citlivé věci, které nelze jiným způsobem reprodukovat i pro budoucí generace. A je jedno, jestli se jedná o historický předmět nebo budovu. V případě jeho ztráty, například příliš vysokým stářím, si můžeme později pro získání představy vyvolat, jak asi vypadal a dokonce ho i podle této předlohy zhotovit znovu. Z rozbitých úlomků kostí si dokážeme složit původní kostru živočicha a díky tomu vidět, jak mohl vypadat. Dalším důležitým aspektem je skutečnost, že nám tato technologie umožňují pomocí zpětného inženýrství zdokonalovat výrobu a urychlovat vývoj, nebo dokonce duplikovat již hotové výrobky.

Tato práce je rozčleněna na několik částí věnující se teoretickým principům 3D skenování, dále pak konstrukci 3D skeneru, popisu běhu aplikace vytvářející model skenovaného objektu. V poslední části zabývající se analýzou naskenovaného objektu jsou popsány jednotlivé aspekty ovlivňující kalibraci, samotné skenování a filtry zpracovávající surová data.

## 1.2 Cíl práce

Cílem této práce je vedle analýzy teoretických principů 3D skenování, také řešení samotné konstrukce 3D skeneru s následnou analýzou naskenovaného objektu. Dílčím způsobem rozebírá, jak získaná data analyzovat, zpracovat a jakými nástroji vytvořit výsledný model ze skenované předlohy.

## 2 Rozbor aktuálního stavu informací o technologii 3D skeneru

### 2.1 Rozdělení podle typu a užití

Stále rychleji se rozvíjející oblast technologií, počínaje výrobními linkami s poloautomatickým či plně automatizovaným provozem v různých sférách výroby vyžaduje změnu standardního přístupu při řešení a návrhu samotné výrobní technologie. Jedná se zejména o zajištění částečné nebo úplné autonomie a soběstačnosti provozu ve snaze úplně vyloučit člověka nejen pro usnadnění pro něj namáhavé práce, ale i v oblasti kontrolní, kdy donedávna byla jeho role jedinečná a nezastupitelná. S rozvojem nejen výpočetní techniky samotné, ale i různých dalších navazujících specifických periférií, které již nyní umožňují provádět prostřednictvím speciálních skenerů digitální kontrolu technologických procesů, včetně již hotových nebo aktuálně vyráběných součástí a zařízení.

Nové požadavky a přístupy pak vyžadují v rozvoji pokrokových technologií získávat digitální podobu nejrůznějších objektů, počínaje strojnými a elektronickými součástmi, pokračují přes lidské tělo a konče u budov. Získávat digitální podobu je prakticky možné u všech pevných objektů vymezených v prostoru svými třírozměrnými obrysy. (Štroner, Pospíšil, 2009)

Základní digitální model objektu lze vytvořit dvěma způsoby:

- Softwarově pomocí programů zpracovávající 3D modely (Studio 3D Max, Lightwave, CAD systémy).
- Hardwarově pomocí 3D skenerů (Microscribe, Freepoint, Atos).

Nejjednodušší a nejpřímější cestou, jak přenést fotografii do počítače je za použití 2D skeneru. Naproti tomu prostorové objekty je možné nejjednodušším a nejefektivnějším způsobem přenášet do počítače k dalšímu zpracování pomocí 3D skenerů. Zároveň je však u tohoto způsobu přenosu reálných prostorových objektů počítat s mnohem větším objemem přenášených dat. Pro plynulé zpracování vysokého objemu pak tento proces klade mnohem vyšší nároky na výkonnost výpočetní techniky, která musí být nutně mnohem výkonnější oproti zpracování dat 2D skenerů. Oblast použití 3D skenerů se dnes neomezuje pouze na průmyslové technologie, ale jejich využití se uplatňuje i v medicíně nebo v kosmetickém průmyslu, například k měření působení a účinků krému na vrásky. Typickým představitelem, který využívá tyto technologie, je např. firma Procter & Gamble. V průmyslové výrobě tyto metody najdou uplatnění při měření deformací výrobků v technologickém procesu výroby nebo pro přesnou montáž citlivých zařízení.

Je zřejmé, že dnes existuje celá řada oblastí a oborů, kde se 3D skenery využívají, ale nejčastěji je tato technologie využívána ve strojírenské výrobě, kde je proces 3D snímání prostorové součásti do digitální podoby nazýván Reverse Engineering. Pod tímto pojmenováním se skrývá široká základna mnoha různých způsobů

využití. V dnešní době špičkoví designéři vytvářejí své návrhy do podoby modelů, kde jako materiál používají sochařskou hlínu. Takto vzniklý model je pro potřeby výroby převeden použitím technologie 3D skenování do CAD aplikací, k dalšímu zpracování. Tento způsob je využíván nejvíce v automobilovém průmyslu k získání velmi složitých geometrických tvarů, které by se v CAD systémech vytvářely velmi obtížně nebo by to bylo úplně nemožné.

Velice významnou a neopomenutelnou oblastí využití tvorby modelů je v případech, kdy neexistuje výkresová dokumentace u velmi starých součástek, které je potřeba znovu vyrobit, přičemž jako vzor může posloužit součástka samotná nebo její model.

Využívat 3D skenerů lze i při měření odchylek, které je velice důležité všude tam, kde je důležité, aby konečný výrobek přesně odpovídal původnímu návrhu.

Princip spočívá v tom, že 3D skener nasnímá tvar konečného výrobku ve formě prostorových dat, které jsou ve speciálním programu, kterým může být např. program Verdict, porovnány aktuálně naměřené hodnoty s původními daty CAD modelu. Dále je možné pomocí barevným rozlišením zvýraznit odchylky nebo je přímo vyjádřit jejich číselnou hodnotu.

Další oblastí využití, je analýza tažení plechu, ke které se s úspěchem používají optické 3D skenery. (Štroner, Pospíšil, 2009)

### **2.1.1 Principy digitalizace**

Prostorová digitalizace pracuje se sadou naměřených 3D bodů sdružených do souboru s výstupními hodnotami, které se nazývají cloud of points v překladu mrakem bodů. Tento typ souboru může generován celou řadou různých typů 3D digitizérů a skenerů, které je možné rozdělit do několika skupin.

Základní členění zařízení určuje způsob snímání, zda tato zařízení snímají 3D data měřených objektů kontaktním nebo bezkontaktním způsobem.

Zařízení pro 3D měření můžeme dále členit podle toho, zda jde o skener stacionární, kdy skenovaný objekt se musí dopravit až přímo k zařízení, nebo hovoříme o mobilním systému, který lze na místo měření přenášet.

Dále můžeme zařízení rozdělit také podle dosahovaného stupně přesnosti. Pro dosažení požadavku vysoké přesnosti skenování dělíme zařízení na použitelná pro Rapid Prototyping nebo Rapid Inspection a na ostatní zařízení, která se s úspěchem dají použít v reklamě, ve filmu nebo videohrách. (Štroner, Pospíšil, 2009)

## **2.2 Rozdělení 3D skenerů podle způsobu a metody měření**

### **2.2.1 Mechanické 3D skenery**

Princip tohoto typu zařízení spočívá v tom, že skenovaný objekt je fyzicky snímán hrotem, který je upevněn na mechanickém rameni. Pohyb ramena v závislosti na poloze snímacího hrotu je měřen v každém svém kloubu pomocí senzoru, který zaznamenává natočení ramene. Poloha skenovaného bodu se získá vyhodnocením údajů ze všech kloubů ramene. (Štroner, Pospíšil, 2009)

### 2.2.2 Optické 3D skenery

Skenovaný objekt je tímto typem skeneru snímán z několika úhlů pomocí optického systému. Při každém natočení objektu, které se provádí buď ručně, nebo pohybovým zařízením, kterým může být krokový motorek řízený počítačem, se data nafocenému objektu odešlou do počítače. Ze snímků pořízených ze všech stran a úhlů pohledu se data zpracují a digitalizovaný model se vytvoří metodou aproximace. (Štroner, Pospíšil, 2009)

### 2.2.3 Laserové 3D skenery

Laserový skener pracuje s využitím laserového paprsku na podobném principu, jako pracuje sonar, kdy využívá jeho vlastností. Vlastní skenování je založeno ve vyslání laserového paprsku kolmo proti předmětu, který se od něho odrazí a vrátí se zpět do skenovacího zařízení. Doba, která uplyne od vyslání do vrácení paprsku, se vyhodnotí a tím získáme informaci o rozměru předmětu ve směru letu paprsku. Informaci o zakřivení povrchu plyne z úhlu, pod jakým se paprsek vrátí zpět do zařízení. Spojením obou informací skener získá přesnou polohu bodu, kterou poté odešle do počítače. Tímto způsobem skener laserovým paprskem obsáhne celé těleso. Dalším způsobem je opačný postup, kdy se otáčí těleso a skenovací zařízení je stacionární. (Štroner, Pospíšil, 2009)

### 2.2.4 Destruktivní 3D skenery

Jedná se o typy zařízení, která mohou digitalizovat u dané součásti nejen její vnější povrch, ale i vnitřní dutiny. Nevýhodou je, že při použití této metody bude digitalizovaný objekt zničen. Naproti tomu u tohoto způsobu digitalizace je jeho velkou výhodou ve schopnosti skenovat i velice složité vnitřní prostory. Tento způsob skenování najde uplatnění zejména v oblasti Reverse Engineeringu. (Štroner, Pospíšil, 2009)

### 2.2.5 Ultrazvukové 3D skenery

Ultrazvuková 3D digitalizace je založena na principu bezkontaktního snímání povrchu součásti pomocí ultrazvukové sondy. Tento způsob skenování je řešením nejméně cenově náročným v nejméně oblasti prostorového snímání povrchů. (Štroner, Pospíšil, 2009)

### 2.2.6 Rentgenové 3D skenery

Použití zařízení tohoto typu umožňuje získávat informace o vnitřní geometrii součástí nedestruktivní metodou digitalizace využitím rentgenového záření. (Štroner, Pospíšil, 2009)

## 2.3 Rozdělení 3D skenerů podle způsobu skenování

V dnešní době existuje již celá řada způsobů a metod, jak skenovat ve volném prostoru. V průběhu doby bylo vyvinuto několik různých typů řešení, které se dají zahrnout pod jednu základní kategorii, která představuje bezkontaktní metody skenování. Zařízení, které využívají tohoto principu, lze rozdělit do následujících dvou základních kategorií:

- polární skenery (laserové)
- základnové skenery (optické)

### 2.3.1 Polární laserové skenery

Polární skenery je možné rozdělit do dvou kategorií. Na pulzní a fázové. Oba typy aktivně využívají k měření vzdáleností laseru s využitím měření tranzitního času nebo fázového posunu. Vedle vzdálenosti je možné měřit i svislé a horizontální úhly získané rotací zrcátka rozmítajícího laserový svazek a limbu. Měření délky, vodorovného úhlu a svislého úhlu se tyto systémy podobají geodetickým stanicím, které také k měření využívají prostorovou polární metodu.

Se svou přesností převážně několika milimetrů, plně panoramatickým úhlovým rozsahem a dosahem okolo desítek až stovek metrů nacházejí polární skenery uplatnění ve stavebnictví. (Štroner, Pospíšil, 2009)

### 2.3.2 Základnové skenovací systémy

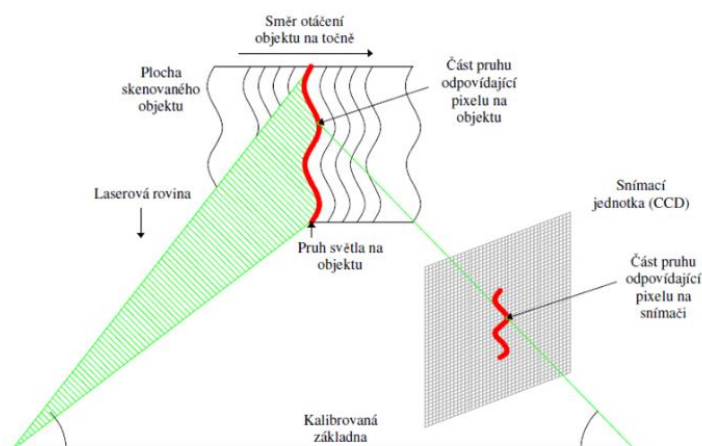
Základnové skenery, využívají principu triangulace, který určuje polohu jednotlivých bodů na základě vidění skenované oblasti více zařízeními. Snímací či signalizační zařízení svou vzájemnou polohou definují základnu pomyslného trojúhelníku. Zbylá dvě ramena trojúhelníku pak představují osy vidění zařízení, které konvergují k sobě. (Štroner, Pospíšil, 2009)

Základnové skenery lze podrobněji rozdělit do těchto tří hlavních podskupin:

- laserová triangulace
- strukturované světlo
- fotogrammetrie (OKS)

#### Laserová triangulace

Skenovací zařízení je soustava, kterou tvoří jedna kamera, laserový zdroj a signalizační systém spolu s otočným mechanismem. Laserový modul vytváří rovinu, která na povrchu měřeného předmětu zanechá liniovou laserovou stopu, kterou vidí kamera jako záměrnou přímkou v prostoru. Prostorový bod je pak vypočten jako průsečík laserové roviny a záměrné přímkou. Výpočet pro určení vztahu mezi kamerou a laserovým modulem je založen na projektivní transformaci. (Štroner, Pospíšil, 2009)



Obr. 1 Ukázka principu laserové triangulace

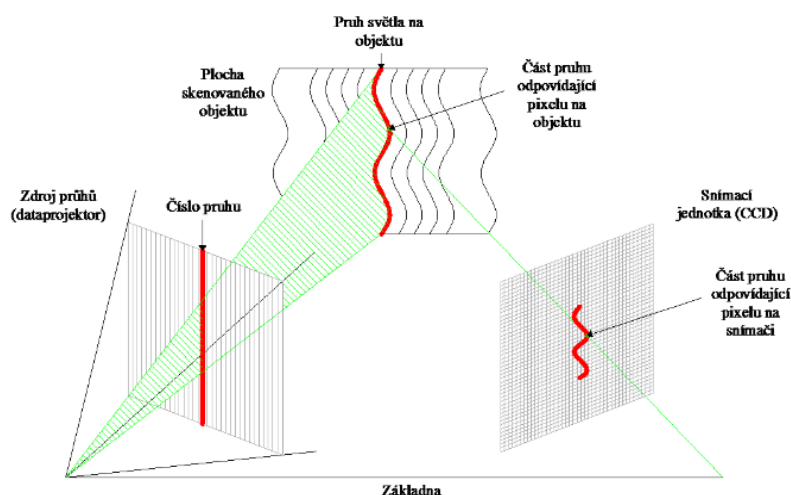
Zdroj: (Urban, 2010)

Laserová triangulace je metoda s možností rychlého sběru dat, která není příliš závislá na světelných podmínkách ve srovnání s jinými principy základnového skenování s jednou významnou nevýhodou zvýšeného bezpečnostního rizika při práci s laserovým zářením. (Štroner, Pospíšil, 2009)

### Strukturované světlo

Dalším typem základnových skenerů je soustava zařízení, která na snímáný objekt promítá značkovacím světlem speciální vzor. Tato metoda skenování využívá princip strukturovaného světla. Celou soustavu tvoří minimálně jedna nebo dvě kamery spolu s jedním datovým projektorem. Ve speciálních případech může být použit větší počet kamer. (Štroner, Pospíšil, 2009)



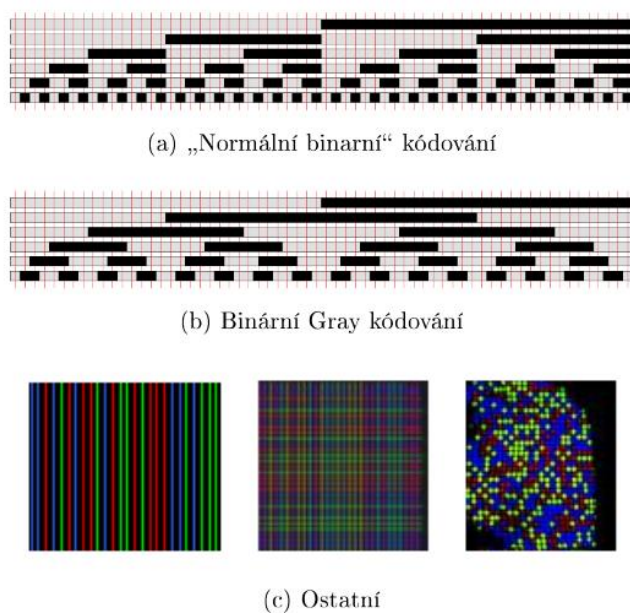


Obr. 2 Princip použití strukturovaného světla – pruhů  
Zdroj: (Urban, 2010)

V praxi se používá několik druhů vzorů. Nejčastěji používanými vzory jsou série černobílých binárních kódů. Konkrétně na obr. 3.(a) a 3.(b) je ukázána posloupnost projektovaných kódů. Každý řádek představuje, jeden snímek.

Na obrázku jsou pro snadnější pochopení principu zvýrazněny červené linie, které v reálných obrazcích neexistují. Zde však zobrazují nejmenší šířku pruhu, kde každý sloupec představuje unikátní binární kód, s který lze pro následné zpracování jednoznačně identifikovat.

Metoda s jedním snímkem využívá jedinečně vygenerované vzory, viz obr. 3.(c). (Štroner, Pospíšil, 2009)



Obr. 3 Ukázka vzorů pro metodu strukturovaného světla

Zdroj: (Koska, 2009)

Skenování metodou strukturovaného světla je velmi rychlým způsobem sběru dat. Naproti této výhodě je zde velká závislost na světelných podmínkách prostředí, a zároveň je nevhodná pro skenování předmětů s vysokým leskem nebo příliš tmavých objektů. (Štroner, Pospíšil, 2009)

### Fotogrammetrie (OKS)

Laserová triangulace a metoda strukturovaného světla využívají k určení prostorových bodů fotogrammetrických výpočtů. Díky tomu se pro tento typ zařízení vžilo příhodné pojmenované jako „fotogrammetrie“ označované také zkratkou OKS skenery.

Vyhodnocení dat získaných těmito systémy pracuje na základě postupů z epipolární geometrie a obrazové korelace.

S použitím fundamentálních matic lze při výpočtu dodatečně určit prvky vnější orientace obou kamer, což je výhodné především z důvodu možnosti měnit konfiguraci kamer v terénu, kdy kalibraci sestavy a výsledné mračno je možné vyhotovit kdykoli později. Naproti tomu velkou nevýhodou je pak skutečnost, že proces zpracování je celkově oproti laserové triangulaci a strukturovanému světlu výpočetně a časově velmi náročný. (Štroner, Pospíšil, 2009)

## 2.4 Ukázky 3D skenerů

### 2.4.1 Víceúčelové polární skenery

Využívají se pro běžné práce a jsou nejpočetněji zastoupenou skupinou na trhu. Uváděné maximální dosahy jsou typické pro objekty s odrazivostí povrchu 80% - 90%.

Parametry těchto typů jsou 80 m na dosah, 5mm-200mm na přesnost a rychlost měření více než 1000 bodů za sekundu. (Pospíšil, 2009)

#### **Faro Photon 80**

Panoramatický skener využívá pro měření délek fázový dálkoměr s rozsahem měření 0,6 m až 76 m a s přesností do 5 mm na vzdálenost 25 m. Rychlost měření je 120 000 bodů za sekundu a zorným polem 360° x 270°. (Štroner, 2009)



Obr. 4 Ukázka panoramatického skeneru Faro Photon 80  
Zdroj: (Štroner, 2009)

### 2.4.2 Víceúčelové triangulační skenovací systémy.

Jsou určeny zejména pro skenování malých předmětů s vysokou přesností na krátké vzdálenosti. Díky tomu nalézají uplatnění zejména ve strojírenských aplikacích, ale i v oblasti archeologie a památkové péče. (Štroner, 2009)

#### **Callidus CT 900**

U tohoto přístroje mohou být maximální rozměry skenovaného objektu v rozsahu 1600 mm x 1400 mm, přičemž hmotnost skenovaného objektu může být až 150 kg. U speciálního modelu je možné skenovat objekt až do hmotnosti 1000 kg. Skenovací hlava se dvěma snímacími senzory se pohybuje ve svislém směru v rozsahu

0 mm až 1000 mm a zároveň se může sklápět okolo svislé osy v rozsahu  $\pm 30^\circ$  a okolo vodorovné osy  $\pm 40^\circ$ . K měření je použit diodový laser s vlnovou délkou 670 nm, který skenuje 4 000 bodů za sekundu s přesností lepší než 0,1 mm. Celková hmotnost skeneru je 450 kg. (Štroner, 2009)



Obr. 5 Ukázka skeneru Callidus CT 900  
Zdroj: (Štroner, 2009)

### 2.4.3 Speciální a kinematické systémy.

Mezi speciální přístroje je možné zařadit všechna jednoúčelová zařízení, která jsou navržena například pro skenování dutin, štol nebo jeskyní.

Kinematické systémy se vyznačují pro ně typickou vlastností, kdy je skener v neustálém pohybu. Ve většině případů se však jedná o nepravidelný pohyb. Pro tyto systémy je také charakteristické, že pro okamžité určení polohy vyžadují další senzory. (Štroner, 2009)

#### HandyScan

Jedná se o ruční skener bez dalších nutných přístrojových doplňků, jako jsou laser trackery, pohyblivá ramena apod. Pro určení své polohy využívá nalepených reflexních terčů. Váží 980 gramů, snímá 18 000 snímků za sekundu a měří s přesností až 50  $\mu$ m. K počítači se připojuje prostřednictvím Fire-Wire rozhraní, v průběhu měření je k dispozici okamžitý náhled změřených bodů. (Štroner, 2009)



Obr. 6 Ukázka ručního skeneru Creaform Handyscan 700  
Zdroj: (Creaform, 2016)

## 2.5 Způsoby měření

Většina běžných měřicích i zobrazovacích zařízení, např. CCD kamera nebo monitor, transformuje 3D scénu na 2D obraz. Díky tomu dojde ke ztrátě informace v 3D souřadnicovém systému v ose z. Jedná se o souřadnici, která určuje hloubku obrazu. Na základě snahy, vrátit tuto informaci k 2D datům pro nové zpracování, vznikla takzvaná zpětná úloha, která se snaží odvodit trojrozměrné vlastnosti objektů z obrazu kamery. Kurčení tohoto chybějícího údaje je za těchto okolností k dispozici nekonečné množství řešení. Vedle toho člověk pozorováním dvourozměrného obrazu neztrácí jeho 3D interpretaci a nemá problémy s rozpoznáváním 3D scény promítané např. v televizi nebo ve filmu. Je zřejmé, že lidské vnímání používá pro odhad hloubky předmětů nějaké apriorní informace vycházející z jejich z-ových souřadnic. Je schopné kupříkladu využít stínů či pozorování těles při pohybu.

Proto i u počítačového vidění, které napodobuje lidský zrak, je potřeba vedle získaného 2D obrazu znát ještě nějakou další informaci k dotvoření 3D obrazu. Optické metody měření jsou v současné době založeny na několika následujících principech. (Štroner, 2009)

Jedná se o tyto hlavní oblasti:

- triangulace
- optická interferometrie
- měření doby letu modulovaného světla

O tom, která metoda je pro optickou formu měření v daném konkrétním případě nejvhodnější rozhoduje celá řada faktorů. Mezi nejdůležitější rozhodující faktory patří vzdálenost zkoumaného předmětu od sensorů. Dále jsou pak důležité i jeho

vnější rozměry a požadovaná přesnost měření. Nezanedbatelné jsou také vlastnosti povrchu předmětu jeho nerovnost, drsnost a odrazivost světla. Mnohdy může rozhodovat i vlastní přístupnost k měřenému objektu a samotné rozměry měřicího systému, které mohou rozhodovat o tom, zda je systém možné umístit například na již fungující výrobní linku. Neméně důležité jsou vlastnosti okolních zdrojů světla, jejich intenzita, spektrum a koherence. Podstatná je i potřebná doba měření. Neopomenutelné jsou i náklady spojené se samotnou realizací systému. (Štroner, 2009)

## 2.6 Metody měření

### 2.6.1 Triangulační metody

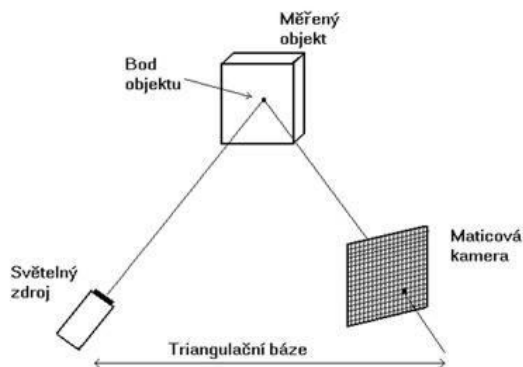
Nejpoužívanější technickou metodou optického měření je v současnosti triangulace, která se dělí na celou řadu různých variant. Na první pohled se mohou jednotlivé varianty jevit jako naprosto rozdílné, ale přesto všechny využívají stejných principů měření. Základní rozdělení triangulačních technik měření je toto:

- aktivní triangulace,
- pasivní triangulace,
- měřicí systémy s teodolitem,
- fokusovací techniky,
- techniky "podoba ze stínování".

#### Aktivní triangulace

Aktivní triangulace využívá techniky založené na takzvané fotogrammetrické rekonstrukci snímaného objektu. Jedná se o metodu, kdy je povrch měřeného objektu nasvícen světelným zdrojem a současně snímán kamerou. Důležitým prvkem u této techniky je ve využití kamerového CCD snímače.

Základem měření u této metody měření je tzv. **triangulační trojúhelník**, jak je možno vidět na obrázku 7, který je tvořen spojnicemi mezi zdrojem světla, snímačem a osvětleným bodem na zkoumaném objektu. Hlavní spojnicí mezi světelným zdrojem a snímačem nazýváme **triangulační bází** (základnou). Na straně světelného zdroje je úhel svíraný s triangulační bází pevně daný, kdežto na straně snímače je úhel vymezen proměnnou pozicí vysvíceného bodu CCD snímače. Na základě velikost tohoto úhlu spolu se znalostí parametrů triangulační báze je možné určit z-ovou souřadnici objektu. (Kalová, 2005)

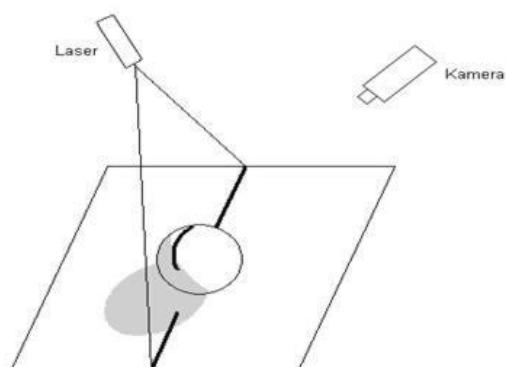


Obr. 7 Triangulační trojúhelník (1D triangulace)  
Zdroj: (Kalová, 2005)

K označení měřeného povrchu se používá:

- světelný paprsek (1D triangulace),
- světelný pruh (2D triangulace),
- strukturovaný světelný svazek (3D triangulace).

Nevýhodou triangulačních metod, která vychází ze samotné podstaty způsobu měření je to, že díky nerovnostem povrchu měřeného objektu nemusí být promítaný bod, pruh či vzor za všech okolností viditelný, a díky tomu pak neexistuje o tomto místě povrchu objektu žádná informace. Na obrázku 8 je vidět, jakým způsobem je povrch v rovině laserového paprsku nasvícen. Vše snímá kamera. Na obrázku šedé plošky znázorňují oblast, kterou kamera nedokáže snímat, protože ji nevidí. Zároveň není laserem osvětlena dolní část koule. (Kalová, 2005)



Obr. 8 Problém viditelnosti u triangulačních metod (2D triangulace)

Zdroj: (Kalová, 2005)

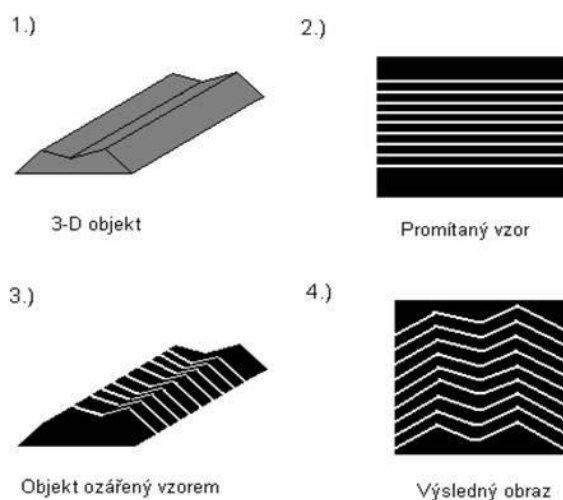
Naproti tomu při použití strukturovaného světelného svazku, je najednou označen celý objekt, a to je velký přínos oproti metodám 1D a 2D triangulace, kdy je nutné pracně a zdlouhavě skenovat scénu z různých úhlů a pozic. (Kalová, 2005)

3D triangulaci je možné rozdělit na následující skupiny:

- technika moiré,
- technika světelného vzoru,
- technika barevného kódu,
- technika fázového posuvu.

Na obrázku 9 je zobrazena jedna z možností 3D triangulace, kdy je na trojrozměrný objekt promítnut vzor, který mohou tvořit pruhy nebo mřížka. Podle zaznamenaných deformací vzoru přímo na objektu lze určit tvar objektu. (Kalová, 2005)





Obr. 9 Technika světelného vzoru (3D triangulace)  
Zdroj: (Kalová, 2005)

### Pasivní triangulace

Pasivní triangulační metody měření využívají v principu různé formy digitální fotogrammetrie. Označení "Pasivní" determinuje způsob osvětlení, kdy není uvažováno přesné geometrické uspořádání osvětlení. Používá se několik základních metod, které jsou rozděleny takto:

- více kamer se známou orientací
- více kamer se samokalibrací
- jedna kamera v různých polohách se samokalibrací

U dynamických systémů se často využívá současně více kamer spolu se znalostí relativních poloh nebo jsou aplikovány samokalibrující metody. Jedna kamera se používá většinou pro statické scény, kdy jsou získávány snímky minimálně ze dvou, ale většinou z více různých pohledů.

U systémů se samokalibrací nemusí být dopředu známa poloha kamery nebo více kamer. Využívá se také i informace o vzájemné poloze kamer. Pro účely spolehlivé kalibrace je vhodné vložit do scény kalibrační předmět. Potřebné parametry pro měření jsou pak určovány na základě změny měřítka a natočení předmětu v každém jednotlivém snímku. Doplnění způsobu měření za použití více kamer je jednoduchá metoda stereovidění, která tvoří speciální podskupinu metod s více kamerami. (Kalová, 2005)

### Stereovidění

Základním principem stereovidění jsou vždy dvě oči, kdy lidské oko vytváří na sítnici obraz pozorovaného předmětu. Pozorujeme-li objekty a předměty zároveň oběma očima, tak je vidíme prostorově nebo stereoskopicky. Tento efekt je vytvá-

řen tak, že levé a pravé oko vytváří na sítnici úplně samostatný obraz, přičemž se tyto obrazy vzájemně od sebe liší. Tento rozdíl obrazů je dán vzdáleností očí od sebe, kdy sdružené zorné paprsky se protínají v příslušném bodě v prostoru. Systémy založené na stereovidění se snaží lidský zrakový systém napodobit. Namísto očí jsou však používány dvě kamery s CCD snímači s rovnoběžnými optickými osami. Vzdálenost středů snímačů byla zvolena na 65 mm. To odpovídá přibližně stejné vzdálenosti jako u lidských očí. Tímto uspořádáním jsou získávány dva **stereoskopické snímky**. (Kalová, 2005)

### 3 Analýza možnosti zpracování grafických dat

K získávání prostorových informací o měřeném objektu pomocí metody laserového skenování, využíváme prostorové souřadnice bodů ležících na povrchu skenovaného objektu. Vedle prostorových souřadnic jsou ukládány i další informace o měřených bodech. Jedná se například o velikost hodnoty intenzity vráceného laserového svazku použitého pro měření délky nebo o informace o textuře a barvě měřeného bodu. Tato hodnota je určována pomocí digitální kamery, která je integrální součástí laserového skenovacího systému.

Vlastní měření probíhá na základě nastavených parametrů pro skenování úplně automaticky. Celý proces je řízen obslužným software. Zvolené parametry určují oblast skenování a také hustotu skenování. Všechny údaje a hodnoty naměřené během skenování jsou průběžně ukládány na paměťové medium. Souhrn všech naměřených bodů jako výsledná množina se nazývá mračno bodů a je základním výstupem z laserového skenování.

Jednotlivé body mračna a jejich prostorové souřadnice jsou vymezeny v pravoúhlém souřadnicovém systému, jehož počátek se vždy totožný s umístěním a vlastní pozicí skeneru. Je-li měření prováděno z vícero stanovišť a pozic skeneru, je nutné data jednotlivých skenů spojit do jednoho homogenního mračna v rámci jednoho souřadnicového systému.

Další fází zpracování změřených údajů ze skenů je pročištění mračna bodů s jeho případným zředěním. Pod pojmem očištění je možné si představit odstranění všech bodů, které vznikly zaměřením nežádoucích předmětů, překážek nebo jsou tvořeny chybami v měření. Proces očištění – decimace, mění v určitých místech hustotu bodů jejich zředěním. Většinou se jedná o místa překryvu dvou či více skenů nebo o místa, která odpovídají zaměření tvarově pravidelných ploch nebo předmětů s malou křivostí.

Transformací, očištěním, nebo zředěním upravené mračno je následně vyhodnocováno pro vytvoření cílového výstupu do podoby modelu. Kvalita a detail vyhodnocení je závislá zejména na požadovaném využití a účelu této práce. Vyhodnocení skenů může probíhat dvěma možnými způsoby, eventuálně jejich kombinací. (Štroner, 2008)

Jedná se o zpracování na základě:

- **Aproximace objektů matematickými primitivy**

Tato metoda spočívá v proložení části nebo celého mračna bodů geometricky exaktně definovaným předmětem, například rovinou, koulí, kuželem, válcem, nebo úsečkou. Standardní geometrická tělesa mohou být u komerčních softwarů rozšířena o celou řadu dalších, mnohdy velice specializovaných tvarů, jako jsou např. nosníky, potrubí, kohouty, kolena, ventily, a celá řada dalších. Výhodou této metody je zejména značné zredukování velikosti dat. Je to díky

tomu, že tisíce měřených bodů jsou nahrazeny tělesem, které je definováno pouze několika konstantami a rovnicí. Naproti tomu nevýhodou této metody je její omezené využití, u objektů, které svým tvarem odpovídají výše uvedeným tělesům nebo se jim velmi podobají. Tímto omezením jsou vyloučeny téměř všechny přírodní útvary nebo složité architektonické objekty. (Štroner, 2008)

- **Trojúhelníkových sítí**

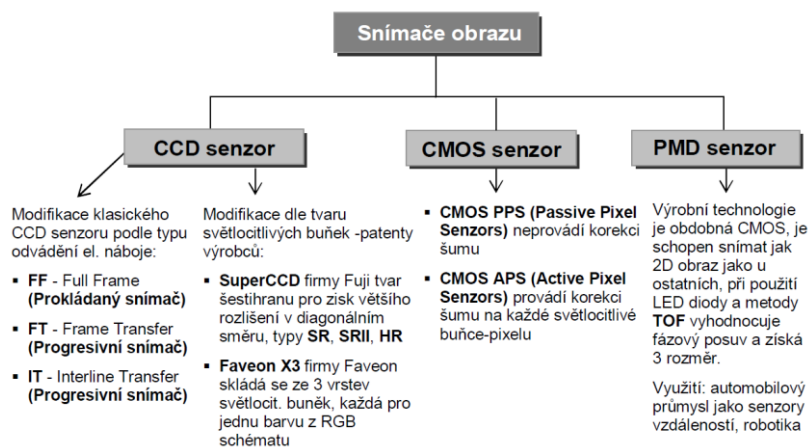
Jedná se o metodu, která je založena na využití velkého počtu rovin, vždy vymezených třemi body mračna. Tímto způsobem dojde k nahrazení povrchu objektu velkým množstvím trojúhelníků. Generování trojúhelníkové sítě je možné s použitím celé řady algoritmů, ale většina z nich pracuje na bázi tzv. Delaunayovy triangulace, která pracuje na základě požadavku, aby se trojúhelníky se přiblížily co nejvíce trojúhelníkům rovnostranným. Tento typ triangulace pracuje s dodržением podmínky, že v kružnici opsané jakémukoli trojúhelníku nesmí být již žádný další bod. (Štroner, 2008)

Celý proces zpracování dat laserového skenování je dokončen v závěrečné části vizualizací výstupních modelů s jejich následnou prezentací. Pojmem vizualizace představuje konečnou úpravu vzhledu měřeného objektu s cílem dosažení co možná největší věrnosti a přiblížení se skutečné předlohy. (Štroner, 2008)

### 3.1 Snímače obrazu

V současné době se nacházejí na trhu dvě hlavní technologie senzorů pro převod světelného záření do digitální podoby aplikovaných ve 3D skenerech. Jsou to snímače CCD a CMOS. Další perspektivní konstrukcí je snímač PMD vycházející z technologie CMOS. Hlavními parametry snímačů jsou rozlišení, doba zpracování snímku (obnovitelnost snímání), velikost šumu, kvantová účinnost a bitové rozlišení A/D převodníku. (Vojáček, 2008)

## 3.2 Rozdělení snímačů



Obr. 10 Schéma znázorňující rozdělení jednotlivých snímačů obrazu  
Zdroj: (Vojáček, 2008)

### 3.2.1 CCD (Charge-Coupled Device)

Tento čip byl původně vyvíjen jako paměťová polovodičová součástka, dnes je výhradně používán pro převod dopadajícího světelného záření na velikost elektrického náboje.

Snímač tvoří matice světlo citlivých polovodičových buněk, kde každá jedna buňka zastupuje jeden pixel. Celkové rozlišení je dáno počtem těchto pixelů (buněk). Tyto buňky tvoří tzv. potenciálové jámy, ve kterých se ukládá energie fotonů jako elektrický náboj. Velikost zachyceného náboje je hlavně ovlivňována intenzitou dopadajícího světla a dobou, po kterou necháme CCD snímač světlu vystaven. Vzniklý náboj se poté musí „svést“ a pomocí A/D převodníku konvertovat na napěťový signál. Praktický problém je ten, že se během tohoto procesu nesmí snímat obraz. CCD snímač by tedy měl být zcloněn, aby nedocházelo k ovlivňování náboje. (Vojáček, 2008)

Proces odvádění a převodu náboje je obecně řešen **3 principy**:

- **FF - Full Frame (Prokládaný snímač)** - dopadajícímu světlu je vystavena celá plocha snímače, tzn. všechny dostupné pixely. (Vojáček, 2008)
- **FT - Frame Transfer (Progresivní snímač)** - snímač je rozdělen na plochu stále vystavenou dopadajícímu světlu (Imaging Area- IA) a plochu trvale zakrytou (Storage Area- SA), do které se v daný okamžik rychle převede celá snímací matice. Tato funkce se zde nazývá paralelizmem. Princip se také označuje jako elektronická clona. Snímač patří mezi nejlevnější, ale zároveň je to kvalitativně nejhorší možné řešení. (Vojáček, 2008)
- **IT - Interline Transfer** - princip je podobný jako u FT. Také zde se jedná o elektronickou clonu, ale s jiným provedením. Vedle každého na světlo citlivého sloupce je zde podobný, ale na světlo necitlivý sloupec, který je navíc za-

krytý. Tato technologie je využita u dražších zařízení, s požadavkem na vysokou obnovitelnost snímání. (Vojáček, 2008)

Aby se co nejvíce zabránilo ztrátám v detekci světla způsobené neregistrováním fotonů dopadajících na zakryté plošky, tak se na povrch CCD snímače implantují miniaturní čočky. Toto zdokonalení je hlavně využito u snímačů s označením Super CCD. (Vojáček, 2008)

### Parametry CCD snímačů

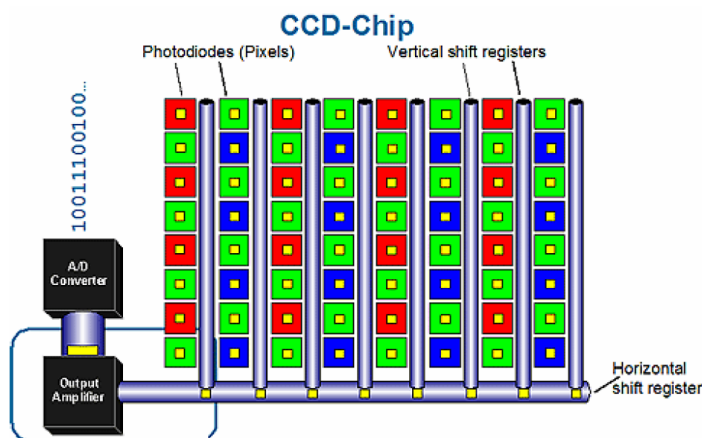
- **Kvantová účinnost CCD čipu (Quantum efficiency - QE)**  
Určuje kolik fotonů dopadajících na čip je přeměněno na signál. Např. QE kolem 30% znamená, že přibližně každý třetí foton vygeneruje elektron. (Vojáček, 2008)
- **Čtecí šum (Read noise)**  
Tato vlastnost udává, s jakou přesností lze změřit náboj každého pixelu. Obvykle se udává v počtu elektronů, které lze díky šumu rozlišit. (Vojáček, 2008)
- **Bitové rozlišení A/D převodníku**  
Všude, kde se provádí digitalizace analogového signálu, je hlavní složkou nepřesností a šumu rozlišení A/D převodníku. Stejně tak je tomu u každého zařízení s CCD snímačem. (Vojáček, 2008)
- **Temný proud (Dark Current)**  
Tento proud reprezentuje náboj generovaný v jednotlivých pixelech i bez osvětlení. Temný proud je lineárně závislý na teplotě přičemž se zdvojnásobuje každých 6 až 7 °C. Za pokojové teploty temný proud zahltlí CCD čip obvykle již během několika desítek sekund. Dojde k tomu, že místo černého místa na snímku je zde světlý bod. (Vojáček, 2008)

### Metody získání barevné informace

Snímače jako takové barvu dopadajícího světla nerozlišují. Každá buňka registruje pouze intenzitu světla, nikoli jeho frekvenci, která udává barvu světla. Pro získání barevného obrazu se využívá barevných filtrů. (Vojáček, 2008)

Podle uspořádání barevného filtru je možno rozdělit dva způsoby získání informace o barvě:

- **Implementace 3-barevného filtru přímo na CCD snímač**  
Toto řešení se realizuje v podobě střídání sloupců pixelů pro jednotlivé barvy nebo „barevné“ pixely rozloženy podle Bayerovy masky (Bayer mask) - viz obrázek 11. Tady je využita vlastnost lidského oka, které má větší citlivost na jas než barevné podání. (Vojáček, 2008)



Obr. 11 Rozložení barevného filtru na barevném CCD snímači dle Bayerovy masky  
Zdroj: (Vojáček, 2008)

- **Použití 3 separátních filtrů na 3 nezávislých CCD snímačích**

Toto řešení se používá u velmi kvalitních přístrojů na zpracování obrazu, kde lze využít další samostatné zpracování obrazů snímaných v různých spektrech podle druhu filtrů. Vysoké nároky jsou také kladeny na provedení optických hranolů, u kterých může docházet k difrakci světelného svazku a tudíž ke zkreslení. Díky vysoké náročnosti je využití převážně ve vědeckých kamerách, například v astronomii při spektrální analýze. (Vojáček, 2008)

### 3.2.2 Super CCD

Princip tohoto CCD snímače je založen na poznatku, že lidské oko citlivěji vnímá vertikály a horizontály, oproti diagonále. Proto je struktura CCD oproti tradičnímu řešení posunuta o  $45^\circ$ , což umožňuje těsnější uspořádání prvků. Celkové výsledné rozlišení se pohybuje okolo 1,6 až 2,3 krát větším, než u klasických CCD. Tato technologie byla vyvinuta firmou Fuji roku 1999 a v době tvorby této práce je na trhu již 6-tá generace. (Macenauer, 2003)

### 3.2.3 CMOS (Complementary Metal Oxide Semiconductor)

Hlavní výhodou těchto snímačů spočívá ve stejné výrobní technologii jako u výroby procesorů, což umožňuje snížení ceny na  $1/3$  oproti stávajícím CCD. Hlavní rozdíl a z něho plynoucí výhody či omezení spočívá v integraci sběrnice dat ke každému jednotlivému pixelu, čímž došlo k výrazné eliminaci ztrát elektrického náboje (tzv. šumu) jako k tomu docházelo u raných typech CCD.

Daná konstrukce však vede ke zmenšení plochy světlocitlivých buněk, což lze opět kompenzovat miniaturizací obvodu, přidáním čoček a zvýšením účinnosti světelné soustavy před snímačem. (Neff, 2008)

V současnosti jsou na trhu dva typy snímačů, liší se způsobem generování elektrického náboje:

- **CMOS PPS (Passive Pixel Sensors)**

Toto je nejjednodušší typ snímače, jenž generuje elektrický náboj úměrně s energií dopadajícího záření. Náboj jde přes zesilovač do A/D převodníku stejně jako u běžného CCD. Toto řešení ale poskytuje nekvalitní obraz a proto je pro využití v 3D skenerech nevhodné. (Neff, 2008)

- **CMOS APS (Active Pixel Sensors)**

U této technologie je každý pixel vybaven analytickým obvodem, který vyhodnocuje šum (ztráty el. náboje) a následně ho eliminuje. (Neff, 2008)



## 4 Návrh technického řešení 3D skeneru

Při návrhu 3D skeneru jsem si stanovil za cíl, aby výsledné řešení bylo jednoduché a zároveň nevyžadovalo příliš náročnou konstrukci s dodržáním další neméně důležité podmínky, aby běh procedur a funkcí v aplikaci nevyžadoval velmi vysoký výpočetní výkon. Dalším nezanedbatelným faktorem při rozhodování o vlastním návrhu byla i dostupnost aktivních prvků skeneru, v tomto případě vstupních zařízení. Zde nastal rozhodující moment pro výběr metody snímání skenovaného objektu, kdy se nabízelo použít soustavu dvou kamer, ale které jsem díky velké výpočetní náročnosti a vysoké pořizovací ceně snímacích kamer, zavrhl. Jako další možný způsob řešení principu snímání jsem proto zvolil metodu triangulace, která vyžaduje pouze snímač kamery a linkový laser.

V této souvislosti jsem také vedl diskusi s odborníky na 3D skenování při mé návštěvě Průmyslového veletrhu v Brně, ze které vyplynulo, že jde zatím o nejvíce rozšířené řešení skenování ve 3D, kdy přesnost skenerů je určena počtem a frekvencí přepínání mezi zapnutím, sejmutím obrázku a vypnutím laseru a sepnutím dalšího laseru v sestavě. Přičemž v základu jde stále o stejný princip triangulace. Dá se předpokládat, že v budoucnu však může převážít jiný způsob a metoda snímání.

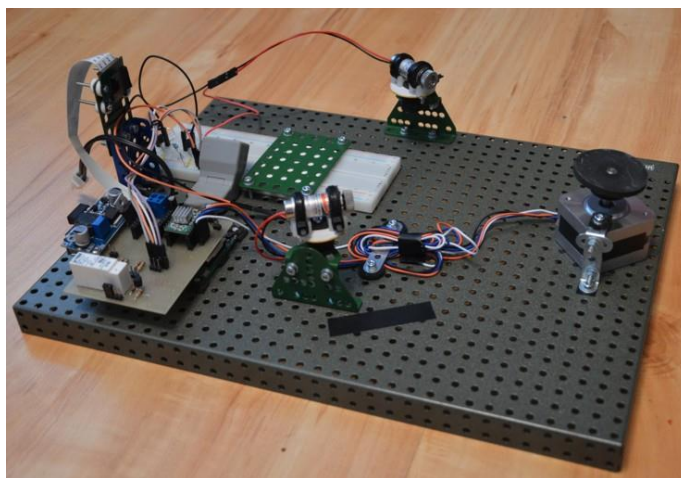
### 4.1 Nosná konstrukce

Proces rozhodování jaké nejlepší řešení pro tvorbu nosné konstrukce zvolit, jsem se řídil zásadou, že musí být především odolná vůči nárazům a případným pádům, Proto jsem nejdříve zvažoval možnost, že by se dala konstrukce celého nosného systému skeneru vytisknout na 3D tiskárně. Toto řešení se ale později po analýze ukázalo jako finančně nákladné a jen samotná příprava konstrukčních dílů v některém z CAD programů by byla časově velmi náročná. Celou stavbu by to zbytečně komplikovalo a tak jsem od tohoto záměru upustil. Dalším velice důležitým kritériem pro stavbu, tak i následnou práci při kalibraci aktivních prvků budoucího 3D skeneru je, aby prvky konstrukce navíc umožňovaly snadnou manipulaci s jednotlivými moduly konstrukce, a zejména pak rychlou flexibilní změnu jejich polohy. Proto jako další možnost řešení konstrukce mě napadlo použít díly ze stavebnice LEGO Technik. Důvodem, proč jsem ani tuto volbu nakonec nevyužil, bylo zejména to, že i přes svou modularitu jsou konstrukce z dílků LEGO křehké. Navíc spoje mají velkou vůli a při zatížení mohou povolit. Také jsem neměl k dispozici potřebný počet konstrukčních dílků potřebných pro stavbu a pořízení celého kompletu NXT by bylo finančně opět velmi nákladné. S tímto vědomím jsem se rozhodl, že nosná konstrukce skeneru by měla být lehká a pevná, což mohou nejlépe zajistit součásti z kovu, respektive díly vyrobené z plechu.

Zahájil jsem proto etapu procházení různých stavebních a kutilských obchodů. Velice rychle jsem zjistil, že některé zde nabízené díly jsou příliš těžké nebo nedokonale opracované a mnohdy nepřesně vyrobené. Přizpůsobení většiny dílů do konstrukčního celku by byl zbytečně vynaložený čas a stejně by sestava nebyla

úplně přesná. Ze všech výše uvedených důvodů jsem nakonec dospěl k rozhodnutí, že neoptimálnější cesta k přesné, pevné a snadno sestavitelné konstrukci skeneru vede v použití stavebnice Merkur. Navíc výrobce Merkuru umožňuje dokupovat jednotlivé díly a cena přitom nedosahuje ceny stavebnice LEGO NXT. Pro urychlení montáže a lepšího spojování jednotlivých dílků stavebnice Merkur jsem pořídil speciální šroubky s imbusovou hlavou, k tomu matice a podložky. Díky této volbě je nosná konstrukce nyní mnohem pevnější a lépe se na ní provádějí úpravy, kterých jsem při samotné stavbě, ale i při pozdější kalibraci 3D skeneru provedl mnoho.

Základem 3D skeneru je deska o rozměrech 00x00. Tato deska představuje nosný systém pro všechny funkční části skeneru. Přímo k desce je připevněna soustava Raspberry Pi s motorovým shieldem. Dále je na desce upevněno nepájivé pole, které na místě drží destička použitá opět ze stavebnice Merkur. Kromě funkčních částí jsou k desce přišroubovány další konstrukční díly. Kamera Raspberry Pi je přimontována ke dvěma destičkám, které umožňují namíření objektivu na hřídel krokového motoru. Krokový motor je připevněn na okraji nosné desky pomocí dalších konstrukčních dílů ze stavebnice Merkur. K nosné desce jsou dále připevněny držáky Laserových diod na otočných šroubech. Všechny díly jsem podložil plastovými fóliemi, abych zajistil odizolování jednotlivých dílů od sebe navzájem.



Obr. 12 Zhotovený 3D skener

#### 4.1.1 Stavebnice Merkur

Mezi výrazné výhody stavebnice Merkur, díky kterým se dá použít pro hobby konstrukce, bych spatřoval v tom, že převážná část stavebnice tvoří lakované ocelové profily a pásy o různých rozměrech, které obsahují přesně definované otvory. Díky těmto otvorům se jednotlivé díly dají přesně spojit šrouby a maticemi. Kromě děrovaných ocelových plátů a pásů, tvoří stavebnici ozubená a neozubená kola různých průměrů.

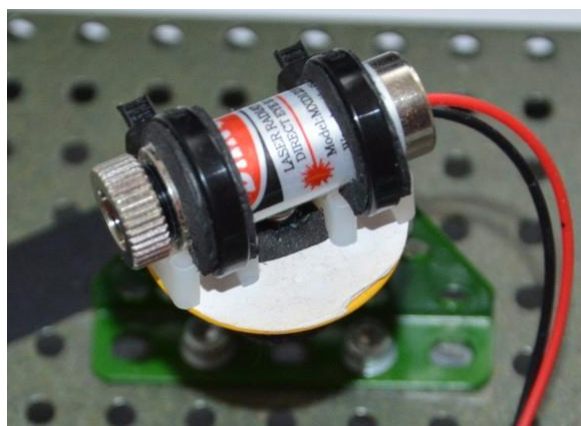
Díky své vysoké variabilitě a cenové dostupnosti, byla tato stavebnice užívána i ve vědeckém výzkumu. Oproti tomu výroba speciálních dílů by byla časově náročná a velice nákladná, Náš významný český chemik Otto Wichterle použil tuto stavebnici při konstrukci přístroje na výrobu kontaktních čoček. Při osazení servomotorů lze z této stavebnice vytvořit jednoduchý plotter pro tisk nebo vyřezávání nebo vrtání děr například do desek plošných spojů. S přidáním třetí osy můžeme uvažovat i o konstrukci 3D tiskáren, které v současné době zažívají velký rozmach.

Naproti tomu ve srovnání s "konkurenční stavebnicí LEGO", která se vesměs skládá z plastových dílů, vykazuje stavebnice Merkur mnohem větší robustnost konstrukce a odolnost proti případnému poškození, při manipulaci s prototypem. A v neposlední řadě bych mohl zmínit, že stavebnici Merkur lze rozšířit i o díly, které byly vytvořeny pro jiné užití, ale jejich napojení, respektive smontování je díky vysoké flexibilitě dílů Merkur jednodušší, oproti spojování podobných dílů ze stavebnice LEGO Technik.

## 4.2 Laserová dioda

Laser je zkratka vytvořená z anglického názvu Light Amplification by Stimulated Emission of Radiation - v překladu zesilování světla pomocí stimulované emise záření. Jedná se o optický zdroj, který produkuje elektromagnetické záření – světlo. To je na rozdíl od ostatních zdrojů světla polarizované, monochromatické a koherentní se sinusovým průběhem.

V aplikacích pro 3D laserové skenery se převážně uplatňuje laserová dioda třídy II. Tyto diody jsou polovodičové, a ve většině případů emitují záření o vlnové délce  $\lambda = 670 \text{ nm}$  a výkonu menším než 1mW. Tyto diody vysílají intenzivní modulovaný signál a mezi vyslaným a přijímaným signálem se měří změna fáze.



Obr. 13 Laserová dioda

#### 4.2.1 Bezpečnostní třídy laserů

Jednotlivé třídy laserů se dělí podle závislosti na tom, jak mohou svým paprskem poškodit lidský zrak.

- **třída I** : je možný trvalý pohled do svazku paprsků, oblast infračerveného spektra
- **třída II** : kontinuální a viditelné záření, přímý pohled do zdroje je možný, oko ochrání mrkací reflex
- **třída III a)** : totéž jako II., ale oko již může být poškozeno, projde-li paprsek ze zdroje do oka prostřednictvím optické soustavy (např. dalekohled)
- **třída III b)** : nebezpečí poškození oka, kdy je nutné používat ochranné pomůcky, například brýle i při pozorování odrazu, max. emise 0,5 W
- **třída IV** : totéž jako III b), ale emise překračuje výkon 0,5 W

Pro mnou zkonstruovaný 3D skener jsem použil Lasery, které dle výrobce mají stanovenou bezpečnostní třídu III a). Velikost výkonu je pak maximálně 1mW. Při práci jsem používal ochranné brýle. Tyto brýle mají tmavě zelený ochranný filtr s rozsahem ochrany proti vlnovým délkám 600 až 700 nm.

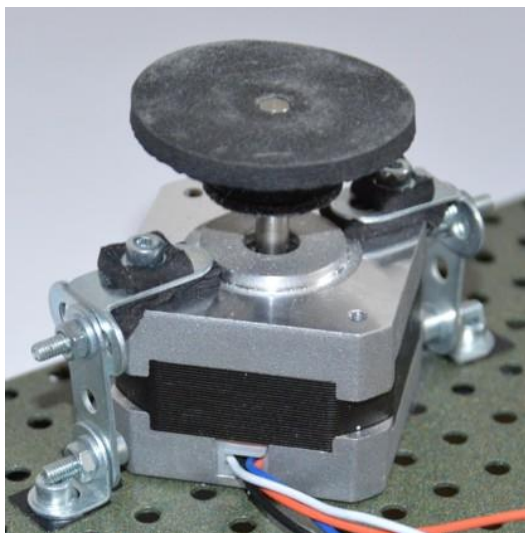
#### 4.2.2 Krokový motor

Krokový motor je typ elektromotoru, který je napájen impulsy, na základě kterých synchronně otáčí vlastním rotorem. Tyto impulsy vytváří stejnosměrný proud. Magnetické pole statoru je vytvářeno spínáním jednotlivých statorových cívek.

Krokovost motoru vzniká tak, že při nízkých rychlostech otáčení je pohyb rotoru nespojitý, díky ustálení rotoru vždy pod sepnutou cívkou. Následující krok je pak vykonán tak, že se sepne vedlejší cívka a vytvořené magnetické pole poté rotor opět posune o úhel, který svírají tyto cívky vzájemně mezi sebou. Někdy se tomuto stavu, kdy se pól rotoru ustálí pod danou cívkou, nazývá stabilní klidová poloha.

Při složitějších konstrukcích se pro výpočet úhlu kroku bere v potaz poměr fází statoru s počtem pólů rotoru vynásobený 360 stupni.

Krokový motor musí být díky své konstrukci, kterou tvoří velký počet spínacích cívek, důležitých pro ovlivnění posunu rotoru ve statoru, řízen elektronicky pomocí ovladače krokového motoru. Někdy je toto zařízení nazýváno řadič nebo anglickým názvem driver. Tento řadič krokového motoru generuje impulzy, které budí jednotlivá vinutí rotoru v daném pořadí. Pořadí impulsů a jejich délka a frekvence má poté vliv na točitý moment, směr otáčení a počet kroků otočení.



Obr. 14 Krokový motor Microcon SX17-1003LQCEF

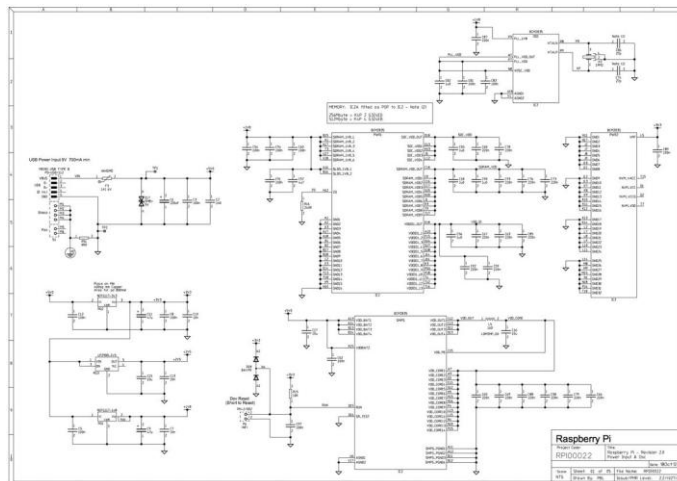
Krokové motory jsou rozděleny na několik druhů:

- **Pasivní** - Prvním typem jsou pasivní krokové motory anglicky označované VR, jejichž rotor je tvořen póly z magneticky měkkého materiálu.
- **Aktivní** - Dalším typem jsou aktivní krokové motory anglicky označované PM, jejich rotor je tvořen permanentním magnetem, kdy se severní a jižní pól permanentního magnetu střídá po obvodu rotoru.
- **Hybridní** - Kombinací předchozích dvou typů vznikly hybridní krokové motory anglicky označované jako HB-Hybrid, kdy na koncích axiálně uloženého rotoru, tvořeného permanentním magnetem, jsou uloženy feromagnetické pólové nástavce.
- **Lineární** - Speciálním druhem krokového motoru je pak lineární krokový motor, kdy stator je rozvinut do přímky. Tento typ je nyní hodně využíván v aplikacích, kde byl dříve zastoupen klasický krokový motor s lineárními pojedy, jako jsou plotterové stroje nebo kartézské 3D tiskárny.

Pro konstrukci 3D skeneru jsem si vybral hybridní dvoufázový krokový motor řady SX17 od firmy Microcon.

### 4.2.3 Raspberry Pi

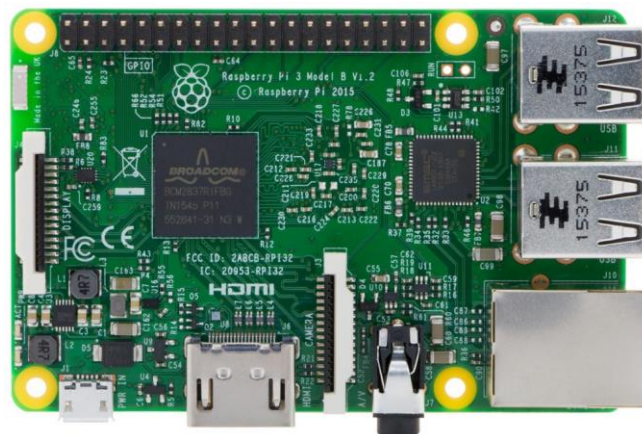
Raspberry Pi představuje v IT odvětví malý jednodeskový počítač, který má rozměry jako platební karta. Byl vyvinut britskou nadací Raspberry Pi Foundation v roce 2012. Já sám jsem se o jeho existenci dozvěděl na přednášce mého vedoucího práce Ing. Radovana Kukly. Díky tomu jsem od té doby věděl, že toto zařízení spolu s mikrokontrolerem Arduino znamená pro svět elektrotechniky výrazný zlom nebo dokonce malou revoluci. Raspberry Pi bylo původně navrženo pro britské studenty, kterým mělo zpřístupnit výuku informatiky a programování, aby si student zajímající se o informatiku nemusel pořizovat drahé zařízení v hodnotách desítek tisíc korun a zároveň bylo toto zařízení přenosné a jednoduše propojitelné se zařízeními, které již měl doma. Sami tvůrci byli překvapeni z obrovské poptávky po zařízení, které vytvořili. Kromě žáků a studentů byli mezi zájemci i zástupci výrobních odvětví, kteří Raspberry Pi užívají pro prototyping, podobně jako platformu Arduino. Na rozdíl od Arduina je však Raspberry Pi 32 bitový respektive 64 bitový plnohodnotný počítač, na kterém běží operační systém Linux. Na rozdíl od klasických osobních počítačů má Raspberry Pi také unikátní rozhraní GPIO (General-purpose input/output), umožňující přímou kontrolu externích zařízení, například mikrokontrolery platformy Arduino. Přínos platformy Raspberry Pi osobně vidím v tom, že spojuje výhody práce s mikrokontrolery a zároveň disponuje dostatečným výkonem, aby mohly být řídicí aplikace přímo součástí zařízení, a nemusely být umístěny na vzdálených osobních počítačích. Dále samotný vývoj těchto řídicích aplikací může být prováděn přímo na této platformě. V neposlední řadě lze Raspberry Pi užívat jako osobní počítač s operačním systémem Linux. K desce počítače, která je vybavena různými typy rozhraní počínaje standartními USB konektory pro připojení klávesnice, myši, ale i externích datových uložení jako je USB Flash disk nebo externí Hard disk, přes HDMI konektor pro připojení monitoru nebo LCD televizoru, nebo speciálními CSI konektory pro připojení kamery nebo monitoru, a konče ethernetovým adaptérem pro připojení do sítě. Speciálně vyvinutá distribuce operačního systému Linux, nazvaná Raspbian, podle složení z názvů Raspberry Pi a linuxové distribuce Debian, má v základu, kromě vývojových nástrojů pro výuku programování v jazyce Python, ale i dalších programovacích jazyků, také aplikační nástroje pro práci s textem a tabulkami Libre Office a další, které umožňují i přehrávání hudby a videa. Tato koncepce "vše v jednom" byla hlavním důvodem, proč jsem si pro konstrukci 3D skeneru vybral právě platformu Raspberry Pi.



Obr. 15 Schéma jednotlivých částí mikropočítače Raspberry Pi – zvětšenina v příloze A  
Zdroj: (Raspberrypi.org, 2016)

### Raspberry Pi 3

Pro řešení řízení 3D skeneru jsem si zvolil model Raspberry Pi ve verzi 3. Uveden na trh byl v roce 2016. Je to první model, který v sobě ukrývá čtyřjádrový 64bitový procesor ARM Cortex-A53 o taktu 1,2 Ghz. Dále je vybaven operační pamětí o velikosti 1 GB, a oproti starším modelům navíc obsahuje integrované moduly Wi-Fi a Bluetooth.



Obr. 16 Základní deska mikropočítače Raspberry Pi 3  
Zdroj: (Raspberrypi.org, 2016)

#### 4.2.4 GPIO

GPIO (General-purpose input/output) lze do češtiny přeložit jako „Obecně účelový vstup a výstup“, je v reálném použití obecný pin integrovaného obvodu nebo umístěný na počítačové desce, jehož chování a stav, tedy zda je pin aktivní, a zda se chová jako vstupní nebo výstupní pin, může definovat uživatel za běhu a v reálném

čase. Maximální napětí přivedené na pin GPIO může být 3,3V. Proto je vhodné navrhovat řídicí logiku jen pro hodnotu 3,3V. Pokud bude nějaká periférie vyžadovat řídicí logiku v rozmezí do 5V, je nutné použít konvertor napětí, aby nedošlo k poškození vnitřních částí čipu, který ovládá GPIO rozhraní.

GPIO blok umístěný na desce Raspberry Pi má ve verzi 3 celkem 40 pinů. 26 pinů je programovatelných. Některé z pinů lze nastavit pro rozhraní sběrnice I2C a dva piny pak pro rozhraní UART. Zbylé piny slouží jako napájecí pro napětí 3,3V, 5V a zemnění. Rozložení pinů je ukázáno na obrázku 16.

Raspberry Pi B+

Pin#	NAME		NAME	Pin#
01	3.3v DC Power	⬛	DC Power 5v	02
03	GPIO02 (SDA1 , I2C)	⬛	DC Power 5v	04
05	GPIO03 (SCL1 , I2C)	⬛	Ground	06
07	GPIO04 (GPIO_GCLK)	⬛	(TXD0) GPIO14	08
09	Ground	⬛	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	⬛	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	⬛	Ground	14
15	GPIO22 (GPIO_GEN3)	⬛	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	⬛	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	⬛	Ground	20
21	GPIO09 (SPI_MISO)	⬛	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	⬛	(SPI_CE0_N) GPIO08	24
25	Ground	⬛	(SPI_CE1_N) GPIO07	26
27	ID_SD (I2C ID EEPROM)	⬛	(I2C ID EEPROM) ID_SC	28
29	GPIO05	⬛	Ground	30
31	GPIO06	⬛	GPIO12	32
33	GPIO13	⬛	Ground	34
35	GPIO19	⬛	GPIO16	36
37	GPIO26	⬛	GPIO20	38
39	Ground	⬛	GPIO21	40

Obr. 17 Schéma rozložení pinů rozhraní GPIO  
Zdroj: (Raspberrypi.org, 2016)

#### 4.2.5 Motorový shield pro Raspberry Pi

Modul připojený na rozhraní Raspberry Pi je koncipován jako základní motorový shield, který je osazen dvěma drivery A4988 určené pro obsluhu dvou motorů. V mém případě je osazen pouze jen jeden ze dvou slotů. K shieldu je ještě připojen konvertor napětí s čipem LM2596-ADJ, který mění vstupní napětí 12V dodávané ze zdroje na 5V, které poté napájí logickou část obvodu a samotné Raspberry Pi. Pří- má větev napětí 12V je přivedena jen k napájení krokového motoru SX17-1003LQCEF. K ochraně logických částí a samotného mikropočítače Raspberry Pi je využíváno spínané relé CQ1-12V. Tím je zajištěno, aby logické obvody a mikropočítač byly připojeny až po oživení 12V větve. Schémata elektrických obvodů jsou znázorněny v příloze A.

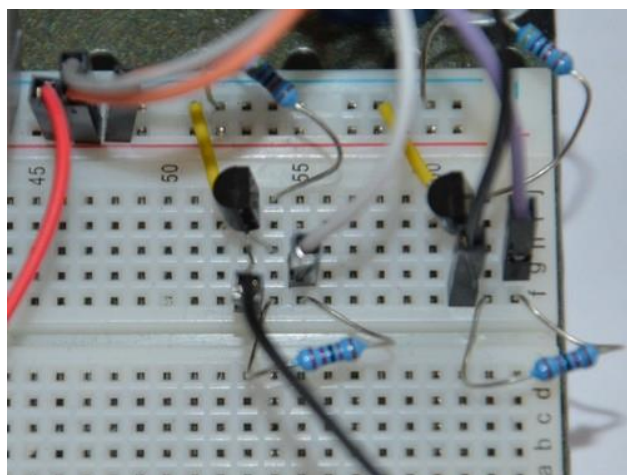




Obr. 18 Motorový shield pro Raspberry Pi

#### 4.2.6 Nepájivé pole

Do nepájivého pole jsou propojovací vodiči vyvedeny z motorového shieldu porty UART. V tomto případě jsou však tyto porty nastaveny jako standardní porty GPIO. Jmenovitě GPIO 14 a GPIO 15. Nepracuje se proto s nimi jako s porty TXD0 a RXD0, které patří pod typ UART. Dále jsou na nepájivém poli umístěny odpory a tranzistory BC 547, určené pro řízení spínání Laserů. Tento konstrukt byl zvolen z toho důvodu, že samotný Laser je napájen 5V, ale GPIO port disponuje pouze napětím 3,3V.



Obr. 19 Nepájivé pole s elektrotechnickými součástkami

#### 4.2.7 Kamera

Modul kamery je vybaven rozhraním CSI, které umožňuje připojení modulu přímo na sběrnici Raspberry Pi bez nutnosti obsazovat piny rozhraní GPIO. Přes rozhraní CSI lze taktéž k Raspberry Pi připojit i TFT display. Velikost modulu je 25×20×9 mm a je opatřen čtyřmi otvory pro uchycení.



Obr. 20 Kamera Raspberry Pi s rozhraním CSI  
Zdroj: (Raspberrypi.org, 2016)

#### 4.2.8 Pygame

Jedná se o multiplatformní sadu modulů jazyka Pythonje navrženého pro tvorbu počítačových her. Obsahuje knihovny, které umožňují práci s grafikou a zvukem. Umožňují také práci se vstupními rastrovými zařízeními, jako jsou kamery a fotoaparáty. Díky knihovně Simple DirectMedia Layer (SDL), nad kterými jsou moduly Pygame postaveny, umožňuje vývojářům vytvářet hry bez nutnosti znát a používat nízko úroňové programování v jazyce C a jeho variant. Využívá se zde předpoklad, že převážná část složitých funkcí v počítačových hrách pracuje s grafikou, a ta jako taková může být oddělena od herní logiky díky použití vysokoúroňového jazyka Python. Funkce náročné na výkon jsou napsány v jazyce

C a jazyce symbolických adres (anglicky assembly language). Pygame byl vytvořen Petem Shinnersem jako Opensource a je volně dostupný pod licencí GNU Lesser General Public License.

## 5 Aplikace 3D skeneru

### 5.1 Program pro řízení 3D skeneru

#### 5.1.1 Řízení krokového motoru

V první řadě je nutné nastavit, které porty z rozhraní GPIO budou ovládat činnost krokového motoru skeneru. Jedná se o parametry DIR, STEP a EN. Dále se o samotné řízení motoru starají tři procedury. Procedura *nastavMotor* inicializuje porty pro krokový motor, procedura *posunMotor* spouští motor a procedura *vypniMotor* deaktivuje motor.

Ukázka kódu:

```
def nastavMotor():
    global EN,DIR,STEP
    GPIO.setwarnings(False)
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(EN, GPIO.OUT)
    GPIO.setup(DIR, GPIO.OUT)
    GPIO.setup(STEP, GPIO.OUT)

def posunMotor(kroky, rychlost):
    global EN,DIR,STEP, presnost
    nastaveni=2
    GPIO.output(EN, False)
    GPIO.output(DIR, False)
    for tx in range(kroky*nastaveni):
        time.sleep(presnost*rychlost)
        GPIO.output(STEP, True)
        time.sleep(presnost*rychlost)
        GPIO.output(STEP, False)
    GPIO.output(EN, True)

def vypniMotor():
    GPIO.output(EN, True)
```

#### 5.1.2 Řízení laserů

Tak jako u krokového motoru, je nutné nastavit porty z rozhraní GPIO pro jednotlivé Lasery. Každý jeden port spíná tranzistor, který spouští napájení Laseru. Procedura *nastavLaser* inicializuje porty pro ovládání laserů. Procedura *laserJednaZap* zapne laser jedna, procedura *laserJednaVyp* vypne laser jedna. Obdobně pracují procedury pro druhý Laser. Procedura *vsechnyLaseryVyp* deaktivuje porty pro ovládání laserů.

Ukázka kódu:

```
def nastavLaser():
    GPIO.setmode(GPIO.BCM)
    GPIO.setwarnings(False)
    GPIO.setup(14, GPIO.OUT)
    GPIO.setup(15, GPIO.OUT)

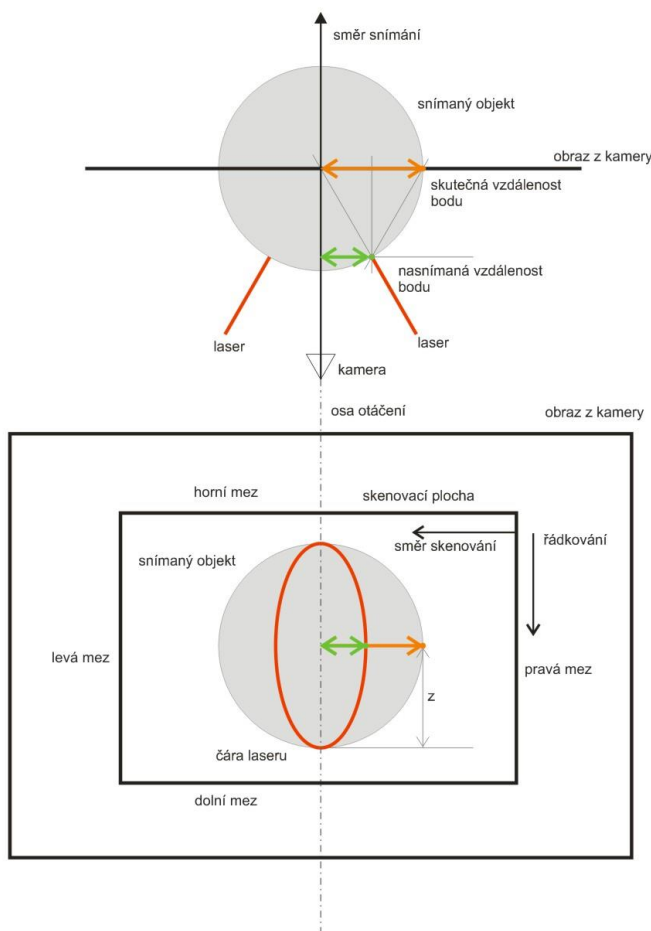
def laserJednaZap():
    GPIO.output(14, 1)

def laserJednaVyp():
    GPIO.output(14, 0)

def vsechnyLaseryVyp():
    GPIO.output(14, 0)
    GPIO.output(15, 0)
```

## 5.2 Metodika skenování

Po sejmutí obrázku kamerou se definuje skenovací plocha, její horní a dolní mez a levá a pravá mez. Samotné skenování probíhá tak, že se prochází pixel po pixelu v každém řádku skenovací plochy. Skenování se provádí od pravé meze skenovací plochy k levé mezi skenovací plochy. Při tomto procesu se vyhledává pixel, který nese známky ozáření červeným laserem. Po jeho detekci se změří vzdálenost od osy rotace. Tato hodnota je násobena koeficientem. Pro zjištění souřadnice  $x$  je vypočtená hodnota násobena sinem daného úhlu. Obdobně pro zjištění souřadnice  $y$  je vypočtená hodnota násobena cosinem daného úhlu. Souřadnice  $z$  je definována vzdáleností od podstavce objektu.



Obr. 21 Schéma metodiky skenování – pohled ze shora (nahore) a pohled ze směru kamery (dole)

### 5.2.1 Nastavení kalibračních parametrů

Před samotným startem aplikace je nutné nastavit některé parametry, které určují samotný běh aplikace. Hodnota 1 u parametru určuje, že daný parametr bude aktivní. Hodnota 0 u parametru určuje, že daný parametr bude neaktivní. S ohledem na světelné podmínky, je také nutné nastavit, zda bude k osvětlení objektu použito denní světlo nebo lampa. Dále je nutné nastavit, zda objekt, který je snímán má světlou nebo tmavou barvu. Tyto parametry poté určují, jaký kontrast kamery bude při snímání použit. Mezi další parametry patří přepínače filtrů, které určují, jaké filtry budou při zpracování obrazu použity a které ne.

Ukázka kódu:

```
#prepinace svetelnych podminek
denniSvetlo=1
svetlyObjekt=0

#prepinace filteru
```

```
filter1=1
filter2=1
filter3=1
filter4=0
```

### 5.2.2 Inicializační proměnné

Dále jsou deklarovány proměnné, které nejsou závislé na konkrétních podmínkách při skenování, ale definují parametry skeneru jako takového. Při změně konstrukce skeneru, jakou je například změna typu snímací kamery, nebo i jiných parametrů, jako je například změna vzdálenosti kamery od osy krokového motoru, nebo změna jejich vzájemné polohy, je nutné tyto parametry přenastavit. Mezi hlavní parametry patří rozlišení kamery, vymezení skenovací plochy nastavením horní, dolní, levé a pravé meze a také polohu osy otáčení. Dále se inicializují globální proměnné, které nejsou závislé na vstupních argumentech při spuštění hlavní aplikace. Mezi tyto proměnné patří celkový počet bodů, kroky a definice barev pro vykreslování.

Ukázka kódu:

```
#rozliseni obrazu
sirka = 1600
vyska = 1200

osa = sirka/2-100
skenovaciSirka = 200
horniMez = 300
dolniMez = 600
levaMez = osa-skenovaciSirka
pravaMez = sirka-osa-skenovaciSirka-20

#krokovani radku ve skenovaci plose
kroky=1

#pocet naskenovanych bodu celkem
celkovyPocetBodu=0

#definovani barev pro vykreslovani
cervena=(255,0,0)
zelena=(0,255,0)
modra=(0,0,255)
bila=(255,255,255)
```

### 5.2.3 Běh aplikace

Spuštění aplikace probíhá přes příkazový řádek terminálového okna systému Rasbian. Je nutné zadat dva vstupní argumenty. Prvním vstupním parametrem je

parametr *nazev*, což by měl být alfanumerický řetězec bez oddělovačů. Pod tímto názvem se uloží výstupní soubor se souřadnicemi, které jsou výsledkem skenování. Druhým parametrem je *pocetOtoceni*. Tento parametr určuje krokovému motoru, o kolik kroků se má otočit skenovaný objekt. Oba argumenty jsou povinné a v případě, že nebudou korektně zadány, tak aplikace nebude pracovat správně.

Po zadání vstupních parametrů dojde ke spuštění aplikace, načtení inicializačních proměnných a kalibračních proměnných. Také dojde k definování a vytvoření pracovních a pomocných matic a vektorů. Do těchto matic se poté budou ukládat částečně zpracovaná data. V dalším kroku dojde k inicializaci kamery a zároveň proběhne první oživení s osvětlením čipu snímače. Toto oživení a krátký osvit čipu CCD kamery před samotným sejmutím obrázku pro zpracování má speciální účel, aby obrázek, který je později pořízen kamerou, byl v lepší kvalitě, než kdyby k tomu došlo ve chvíli sejmutí obrázku.

Ukázka kódu:

```
import picamera
cam = picamera.PiCamera()

cam.start_preview()
sleep(3)
cam.stop_preview()
```

Spolu s předchozími kroky jsou také zavolány inicializační procedury, které aktivují vstupní a výstupní porty na rozhraní GPIO. Procedura *nastavLaser* připravuje k činnosti Lasery. Procedura *nastavMotor* připravuje k činnosti krokový motor. Ukázky kódu těchto procedur byly ukázány výše.

O samotný cyklický běh aplikace se stará cyklus *for*, který je omezen parametrem *pocetOtoceni* určující počet cyklů. V cyklu jsou postupně zavolány procedury *laserJednaZap*, *provedSken*, *laserJednaVyp*, *laserDvaZap*, *provedSken*, *laserDvaVyp*, *posunMotor*.

Procedura *laserJednaZap* aktivuje první laser, který osvítl skenovaný objekt paprskem. Procedura *provedSken* provede samotné sejmutí obrázku a provede první zpracování dat. Procedura *laserJednaVyp* vypne první laser a procedura *laserDvaZap* aktivuje druhý laser a osvítl skenovací paprsek z druhé strany objektu. Poté dojde k opětovnému zavolání procedury snímající obrázek z kamery a jeho zpracování. Procedura *laserDvaVyp* deaktivuje druhý laser. Nakonec procedura *posunMotor* provede pootočení krokového motoru se skenovaným objektem na další pozici.

Ukázka kódu:

```
for i in range (pocetOtoceni-1):
    laserJednaZap()
    provedSken(i)
    laserJednaVyp()
    print hodnoty
    laserDvaZap()
```



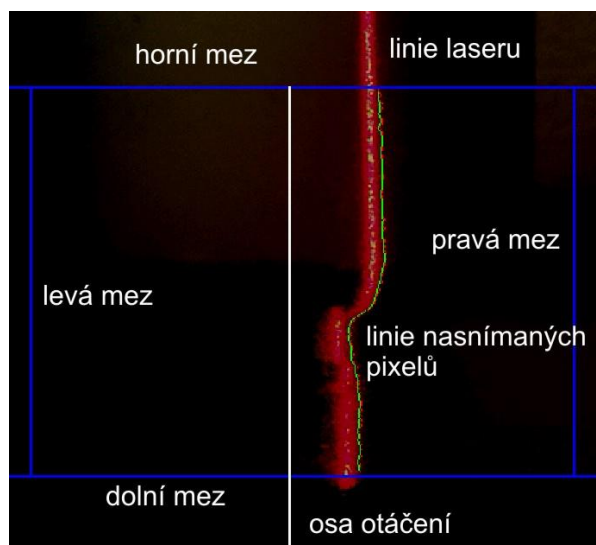
```
provedSken(i)
laserDvaVyp()
posunMotor(1600/(pocetOtoceni-1),100)
```

Procedura *provedSken* provádí samotné snímání obrázku a jeho zpracování do pozdější podoby sady souřadnic. Funkce *capture* provede sejmutí obrázku z kamery. Tento obrázek je poté uložen na disk pro další zpracování. Po opětovném načtení souboru obrázku procedura *rotate* otočí nasnímaný obrázek, protože samotná kamera je díky upevnění k nosné konstrukci obráceně, tzv. vzhůru nohama. K upevnění kamery ve správné orientaci brání skutečnost, že plochý konektorový svazek pro rozhraní SPI, neumožňuje takové připevnění z důvodu jeho křehkosti a délce.

Ukázka kódu:

```
#sejmuti obrazku
cam.capture(cestaSouboru)
#Inicializace obrazovky pro zobrazeni
obrazovka=
pygame.display.set_mode([sirkaObrazovky,vyskaObrazovky])
#nacteni obrazku z ulozeneho souboru
obrn = pygame.image.load(cestaSouboru)
#otoceni obrazku o 180 stupnu
obrazek = pygame.transform.rotate(obrn, 180)
#ziskani obdelnikovou plochu povrchu
obrObdelnik = obrazek.get_rect()
#prekresleni jednoho obrazku na druhy
obrazovka.blit(obrazek, obrobrObdelnik)
#aktualizace plneho zobrazeni na obrazovku
pygame.display.flip()
```

K obrázku jsou přidány čáry, které vykreslují ohraničení horní, dolní, levé a pravé meze a také osu rotace. Tyto čáry jsou vytvářeny funkcí *line*.



Obr. 22 Ukázka vymezení skenovací plochy s linií nasnímaných pixelů

Ukázka kódu:

```
#horni vodorovna cara
pygame.draw.line(obrazovka,modra,(10,horniMez),(sirka-
10,horniMez),2)
#spodni vodorovna cara
pygame.draw.line(obrazovka,modra,(10,vyska-dolniMez),(sirka-
10,vyska-dolniMez),2)
#prava kolma cara
pygame.draw.line(obrazovka,modra,(sirka-
pravaMez,horniMez),(sirka-pravaMez,vyska-dolniMez),2)
#leva kolma cara
pyga-
me.draw.line(obrazovka,modra,(levaMez,horniMez),(levaMez,vyska-
dolniMez),2)
#osova cara
pygame.draw.line(obrazovka,bila,(osa,horniMez),(osa,vyska-
10),2)
```

Po vykreslení čar, které vymezují meze, dojde ve vymezené ploše na každém řádku pomocí cyklu *while* k prohledávání pixel po pixelu pomocí funkce *get\_at*, která snímá každou složku pixelu z modelu RGB, do té doby dokud není nalezen takový pixel, který svými složkami modelu RGB odpovídá složkám RGB modelu barevného filtru, který je nastaven pro snímání červené barvy.

Je-li takovýto pixel nalezen, dojde k jeho uložení do datové matice a počítadlo počtu nalezených bodů je navýšen o jedna.

Ukázka kódu:

```
#hledani v rozsahu (dolniMez a horniMez) x (pravaMez a levaMez)
```

```

while y<vyska-dolniMez:
    #ziskani RGB slozek z jednoho bodu z obrazku
    #cervena slozka
    slozkaCervena = obrazovka.get_at((sirka-x,y))[0]
    #zelena slozka
    slozkaZelena = obrazovka.get_at((sirka-x,y))[1]
    #modra slozka
    slozkaModra = obrazovka.get_at((sirka-x,y))[2]

    if slozkaCervena>filterCervena and slozkaZelena<filterZelena
and slozkaModra<filterModra:
        slozkaCervena1 = obrazovka.get_at((sirka-x-1,y))[0]
        slozkaCervena2 = obrazovka.get_at((sirka-x-2,y))[0]
        if slozkaCervena1>slozkaCervena:
            x=x-1
            if slozkaCervena2>slozkaCervena1:
                x=x-1

        vzdalenostOdOsy=sirka-x-osa
        #matice (y-horniMez,uhelKroku,vzdalenostOdOsy)
        maticeDat[y-horniMez][uhelKroku]=vzdalenostOdOsy
        celkovyPocetBodu = celkovyPocetBodu+1

        y=y+kroky
        x=pravaMez

```

Zároveň je do patřičných souřadnic na obrázku zanesen tento nalezený pixel a je zvýrazněn:

```
obrazovka.set_at((sirka-x,y),barva)
```

Pokud byl v kalibrační části zvolen některý z filtrů, jsou nad uloženými daty prováděny korekce pomocí zvoleného filtru. Tyto filtry pracují na principu plovoucího průměru, kdy podle zvoleného filtru jsou vůči sobě prováděny korekce určeného počtu bodů v okolí určujícího bodu.

Ukázka kódu:

```

if filter1:
    y=horniMez+2
    while y<vyska-dolniMez-2:
        d = plovouciPrumer(maticeDat[y-horniMez-
1][uhelKroku],maticeDat[y-horniMez][uhelKroku],maticeDat[y-
horniMez+1][uhelKroku])
        pomocnyVector[y-horniMez] = d
        y=y+kroky

```

```

y=horniMez+2
while y<vyska-dolniMez-2:
    maticeDat[y-horniMez][uhelKroku] = pomocnyVector[y-
horniMez]
    y=y+kroky

```

Funkce plovoucího průměru:

```

def plovouciPrumer(a1,a2,a3):
    prumer=a2
    if (a1>0 and a2>0 and a3>0):
        prumer = (a1+a2+a3)/3
    else:
        if (a1==0 and a2 and a3):
            prumer = (a2+a3)/2
        if (a1 and a2==0 and a3):
            prumer = (a1+a3)/2
        if (a1 and a2 and a3==0):
            prumer = (a1+a2)/2
    return prumer

```

Poté, co je zakončen hlavní cyklus *for* aplikace, tak jsou podle nastavení v kalibrační části prováděny další filtry, které se provádějí nad konečnými daty. Jakmile jsou tyto dodatečné úpravy nad daty hotové, jsou pomocí procedury *ulozSouradnice* uloženy do souboru s koncovkou *xyz*. Z *maticeDat* je cyklem *while* postupně zpracovávána hodnota *vzdalenostOdOsy* pomocí vzorců, kdy je hodnota násobena koeficientem určujícím zakřivení vůči ose a dále vynásobena sinem daného úhlu pro vytvoření složky souřadnice *x* a pro vytvoření složky souřadnice *y* je násoben cosinem daného úhlu.

Ukázka kódu:

```

def ulozSouradnice(uhelKroku, souborXYZ):
    global maticeDat, celkovyPocetBodu
    y=horniMez
    y=horniMez+1
    while y<vyska-dolniMez:
        vzdalenostOdOsy = maticeDat[y-horniMez][uhelKroku]
        if vzdalenostOdOsy!=0 and vzdalenostOdOsy>-200:
            uhel=float(2*pi/(pocetOtoceni-1)*uhelKroku)
            rx=float(math.sin(uhel)*vzdalenostOdOsy*koef)
            ry=float(math.cos(uhel)*vzdalenostOdOsy*koef)
            rz = y
            souradnice = str(rx)+" "+str(ry)+" "+str(rz)
            souborXYZ.write(souradnice+"\n")
            y=y+kroky

```

Pro účely analýzy objektu je z uložených souřadnic pomocí procedury *vygenerujObrazSouradnic* vygenerován 2D obrázek souřadnic. Pomocí dvojitého cyklu *for* z *maticeDat* je vyseparována hodnota *x*, která je pomocí procedury *set\_at* zanesena do obrázku, který má černou plochu, jako zelený pixel. Do *x*-ové složky vstupních parametrů procedury *set\_at* je vkládána hodnota *x* násobená parametrem *posun*, který každý nový naskenovaný sloupec posune o patřičnou vzdálenost na obrázku. Do *y*-ové složky vstupních parametrů procedury *set\_at* je vkládána hodnota *r* reprezentující pořadí řádku naskenovaného pixelu. Obě vstupní hodnoty jsou ještě vynásobeny parametrem pro přiblížení.

Ukázka kódu:

```
def vygenerujObrazekSouradnic():
    posun=5
    priblizeni=2

    naskenovanaData = cestaRamdisk+datum +'.txt'
    obrazekSouradnic = cestaRamdisk+datum +'.jpg'
    souborTXT = open(naskenovanaData,"a")
    obrazovka=
pygame.display.set_mode([sirkaObrazovky,vyskaObrazovky])
    for u in range (pocetOtoceni-1):
        for r in range (radek):
            x=maticeDat[r][u]
            souborTXT.write(str(x)+",")
            obrazov-
ka.set_at((u*posun+x*priblizeni,r*priblizeni),zelena)
            #aktualizace plneho zobrazeni na obrazovku
            pygame.display.flip()
            souborTXT.write("\n")
        souborTXT.close()

    #aktualizace plneho zobrazeni na obrazovku
    pygame.display.flip()
    #ulozeni obrazku na disk
    pygame.image.save(obrazovka,obrazekSouradnic)
```

Před samotným koncem běhu aplikace pomocí procedury *vsechnyLaseryVyp* dojde k deaktivaci portů rozhraní GPIO ovládající Lasery. Procedura *vypniMotor* deaktivuje porty rozhraní GPIO a uspí krokový motor.

## 6 Analýza naskenovaného objektu

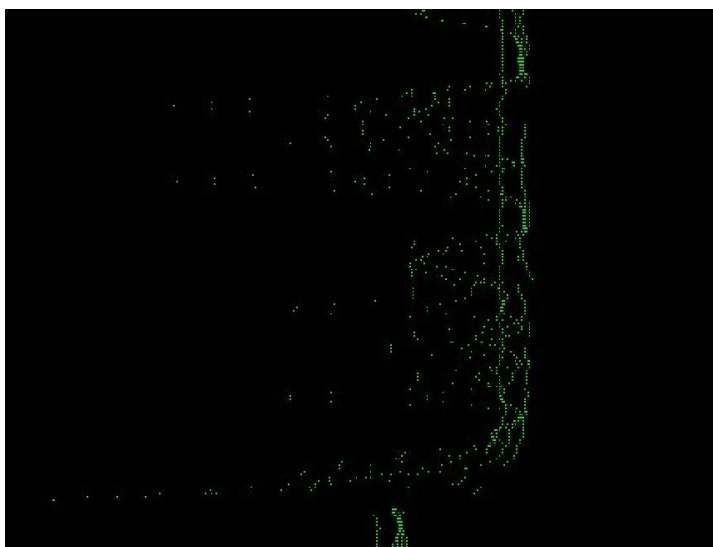
V této kapitole se budu zabývat rozbořem výsledných dat vzešlých ze skenování testovacího objektu. Definuji si, jaké vlivy mají vliv na výsledný model. A které daný model ovlivňují více a které méně.

Na kvalitu naskenovaného objektu má vliv několik vlivů:

- Světelné podmínky
- Odrazivost snímaného objektu
- Barva snímaného objektu
- Užití jednotlivých filtrů

### 6.1 Světelné podmínky

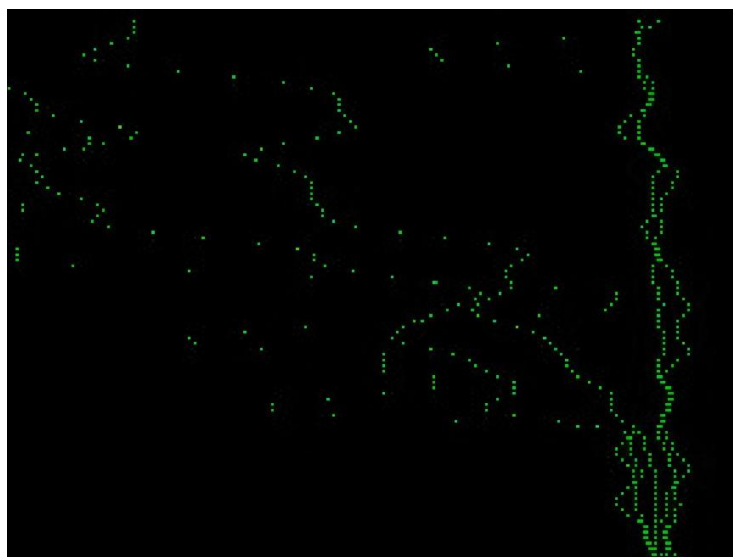
Na obrázku je vidět, že kvůli příliš světlému snímku nedochází ke kontinuálnímu skenování podél červené linie osvětlené laserem. Jsou detekovány použitelné pixely i ve velké vzdálenosti od linie paprsku laseru. Výsledné souřadnice jsou vychýleny.



Obr. 23 Ukázka vygenerovaných souřadnic znázorněných v 2D obraze

### 6.2 Odrazivost snímaného objektu

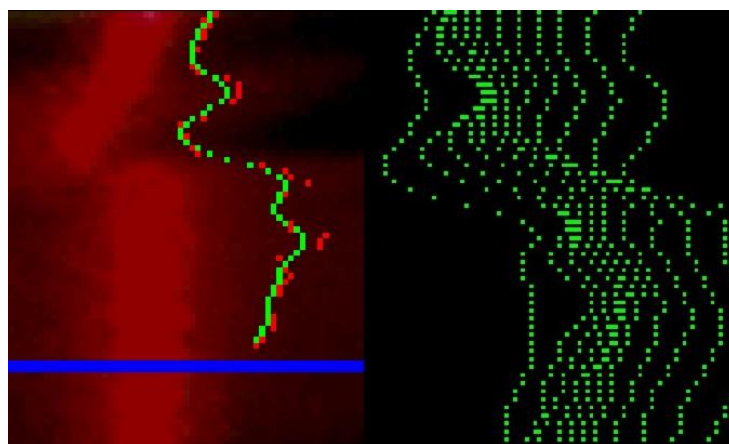
Na obrázku je vidět, že kvůli vysoké odrazivosti objektu dochází k dodatečným odrazům červeného paprsku a tím není dosaženo snímání podél červené linie. Dochází ke snímání odrazů červeného laseru i v místech, kde není linie červeného paprsku laseru.



Obr. 24 Ukázka vygenerovaných souřadnic znázorněných v 2D obraze

### 6.3 Barva snímaného objektu

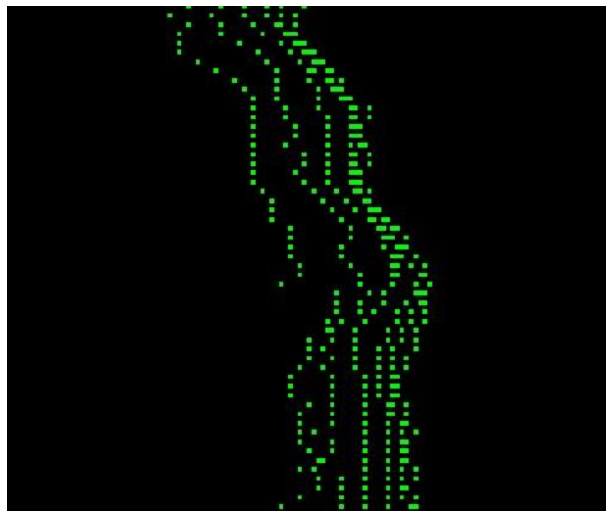
Pokud snímaný objekt má barvu blízkou červené, dochází ke splynutí červené linie laseru s barvou povrchu objektu a snímání je poté nahodilé, jak je patrné z obrázku.



Obr. 25 Ukázka nahodilé detekce (vlevo) a vygenerovaných souřadnic (vpravo)

### 6.4 Filtry

Jednotlivé filtry mají vliv na výsledný naskenovaný model. Při užití základního filtru 0, implementovaného při samotném skenování, dochází k velkému rozptylu vytvořených bodů jak je vidět na obrázku. Je na uvážení, jestli míra šumu je v zanedbatelné míře vůči nasnímaným detailům, které jsou na skenovaném objektu. Pokud ano, tak výsledný model reprezentuje nejvěrněji svoji předlohu.



Obr. 26 Ukázka vygenerovaných souřadnic bez užití filtrů

Užitím filtrů 1 a 2 dochází ke kompenzaci extrémních bodů v modelu, jak je vidět na obrázku. Filtr 1 upravuje parametry třech okolních bodů určujícího bodu. Filtr 2 upravuje parametry pěti okolních bodů určujícího bodu. Jednotlivé body jsou k sobě více přidružené a jsou rozeznatelné jednotlivé linie. Tímto již ale dochází ke zkreslení vůči skenované předloze.



Obr. 27 Ukázka vygenerovaných souřadnic s užitím Filtru 1 a Filtru 2

Filtr 3 upravuje parametry devíti okolních bodů vůči určujícímu bodu. Filtr 4 provede filtrování filtru 3 dvakrát po sobě. Užíváním filtrů 3 a 4 můžeme docílit úplného zahlazení extrémů, dochází však také k degradaci ostrých hran na modelu. Proto je užití těchto filtrů dobré u oblých nebo válcovitých objektů. Při extrémně členitém objektu s ostrými výstupky není dobré užívat filtry 3 a 4, jak je vidět na



obrázku. Jednotlivé lokální extrémy složené z více bodů jsou zarovnány k ostatním bodům. V konečném důsledku při několikanásobném užití tohoto filtru dojde k zarovnání jednotlivých linií do rovných sloupců.



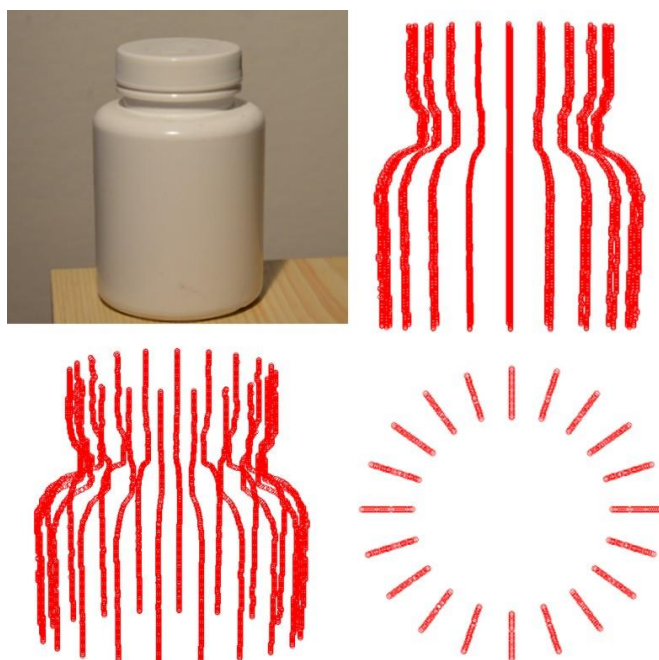
Obr. 28 Ukázka vygenerovaných souřadnic s užitím všech filtrů

Z výše uvedených kritérií, které mají přímý vliv na výsledný model vytvořený z naskenovaného objektu, tedy vyplývá, že pro výsledný model, aby jeho reprezentace byla co nejbližší skenovanému objektu, je důležité provést patřičnou kalibraci již před samotným skenováním. Dále je potřeba s rozmyslem volit, které filtry použijeme pro začištění a vyhlazení linie bodů, a které pro daný typ skenovaného objektu nejsou vhodné, a proto je nepoužijeme vůbec.

## 6.5 Zobrazení objektu v programu Matlab

Na obrázku jsou vidět jednotlivé shluky bodů podél osy  $x$ , tak jak byly nasnímány podél kolmé linie červeného laseru. Jednotlivé shluky jsou vůči sobě orientovány pod úhlem, který byl určen na začátku skenování ze vstupního argumentu definující počet skenovacích kroků.

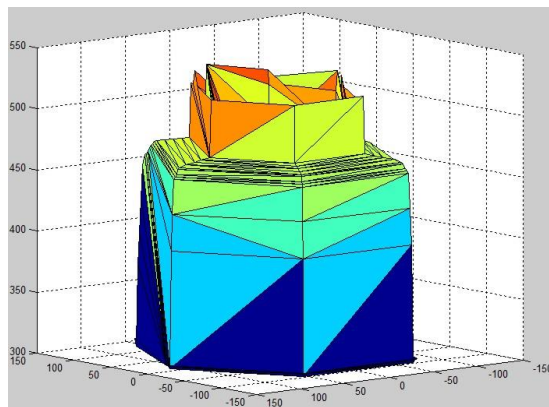
### 6.5.1 Ukázka reálného objektu a mračna bodů



Obr. 29 Porovnání reálného objektu (vlevo nahoře) s mračny bodů z různých pohledů – pohled z boku (vpravo nahoře), pohled ze strany z vrchu (vlevo dole) a pohled ze shora (vpravo dole)

Z obrázku je patrné, že výsledný model zobrazuje určující extrémy ve tvaru skenovaného objektu. Jednotlivé body mračna jsou spojeny do podoby trojúhelníkové sítě. Díky použitým filtrům, které hodnoty na ose  $x$  značně zkreslují, nejsou jednotlivé body z jednotlivých kolmých mračen bodů v jedné horizontální rovině a dochází tak ke zkreslení při vykreslování trojúhelníkové sítě. Jelikož použité technologie vykreslování, kdy výsledná plocha musí být uzavřena a nevytváří válcovou plochu bez obou podstav, dochází v prostoru uvažované podstavy válce k uzavření plochy. Tato část je díky rozdílným výškám bodů z jednotlivých mračen viditelně deformována.

### 6.5.2 Ukázka modelu s vygenerovaným povrchem



Obr. 30 Ukázka modelu s vygenerovaným povrchem

Skript v programu Matlab pracuje tak, že dojde nejprve k načtení souboru souřadnic s koncovkou *xyz*. Funkce *importdata* po vložení jejich parametrů *jmenoSouboru* a *oddelovac* zpracuje vstupní data do podoby matice. Kde každý ze tří sloupců reprezentuje jednu složku souřadnic. Z této matice jsou jednotlivé sloupce vyextrahovány a uloženy do patřičných vektorů příslušného jména složky souřadnice. Tyto vektory jsou posléze vykresleny pomocí procedury *plot3* do trojrozměrného grafu. Pro zpracování shluku bodů do povrchu se užije funkce *delaunay*, která z prvních dvou vektorů pomocí Delaunayovi<sup>1</sup> triangulace vytvoří matici. Procedura *trisurf* zobrazí trojúhelníky definované v matici jako povrch. Každý řádek matice definuje jednu trojúhelníkovou plochu pomocí indexování do vektorů nebo matic, které obsahují vrcholy *X*, *Y* a *Z*.

Ukázka kódu pro zpracování výstupního souboru souřadnic do podoby mračna bodů a do modelu s vygenerovaným povrchem:

```
jmenoSouboru = 'soubor.xyz';
oddelovac = ' ';
maticeDat = importdata(jmenoSouboru,oddelovac);

x = maticeDat(:,1);
y = maticeDat(:,2);
z = maticeDat(:,3);

plot3(x,y,z,'or');

pomocnaMatice = delaunay(x,y);
trisurf(pomocnaMatice,x,y,z);
```

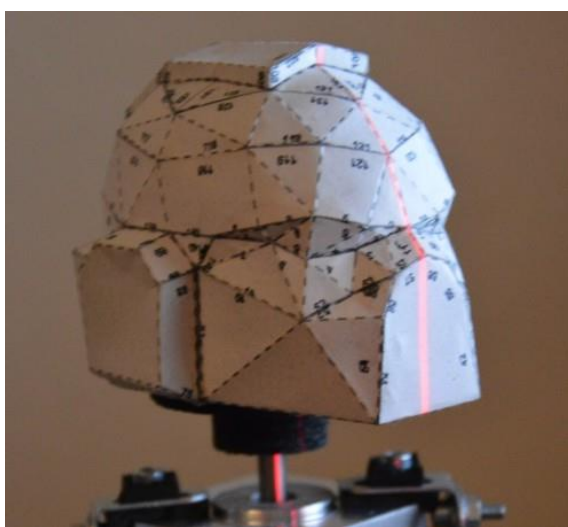
<sup>1</sup> Boris Delaunay (1890-1980). – Ruský matematik, Delaunayova triangulace má za cíl maximalizovat minimální úhel všech úhlů trojúhelníků v triangulaci, tak aby nevznikaly příliš dlouhé a tenké trojúhelníky.

Z předložených výsledků je patrné, že zobrazení shluků bodů je pomocí nástrojů programu Matlab poměrně věrné a skenovaný objekt je z nich rozpoznatelný. Na druhou stranu je patrné, že proložením těchto shluků povrchem, který je složený z trojúhelníků, dochází ke značnému zkreslení vůči předloze.

## 7 Skenování tvarově členitějšího objektu

V této kapitole bych chtěl představit výsledky skenování tvarově členitějšího objektu. Pro potřeby skenování jsem proto vybral tvarově členitý objekt, který jsem sám vyrobil z papíru. Samotný objekt je dostatečně členitý s různě skloněnými plochami a výstupky, který je pro demonstraci schopnosti naskenovat tento typ objektu 3D skenerem vhodný. Oproti tomu stojí skutečnost, že mnou zkonstruovaný skener má omezení, daná samotnou konstrukcí. Určující v tomto je zejména rozsah v jakém je schopen přijatelně naskenovat objekt tak, aby v prezentaci mračna bodů a z transformovaného povrchu se dalo rozpoznat, že naskenovaný model odpovídá svou podobou předloze.

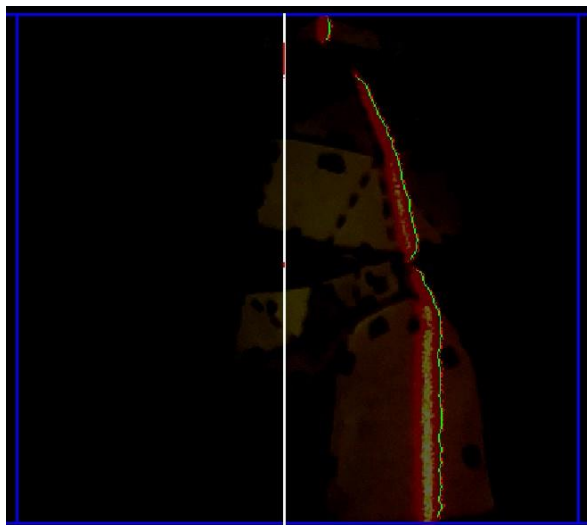
Z obrázku je patrné, že si skenovaný objekt můžeme představit jako modifikovanou hlavu. Ve skutečnosti se jedná o helmu jisté futuristické postavy. Samotná struktura objektu a jeho povrch je ve své konkrétní podobě značně členitý. Navíc je na jeho povrchu umístěno několik dalších geometrických útvarů, které vhodně dotváří různorodost samotného objektu jako celku. Zároveň splňuje rozměrová kritéria pro vlastní realizaci demonstrace skenování tak, aby se bez problémů vešel do skenovací plochy skeneru.



Obr. 31 Ukázka tvarově členitějšího objektu osazeném na podstavci

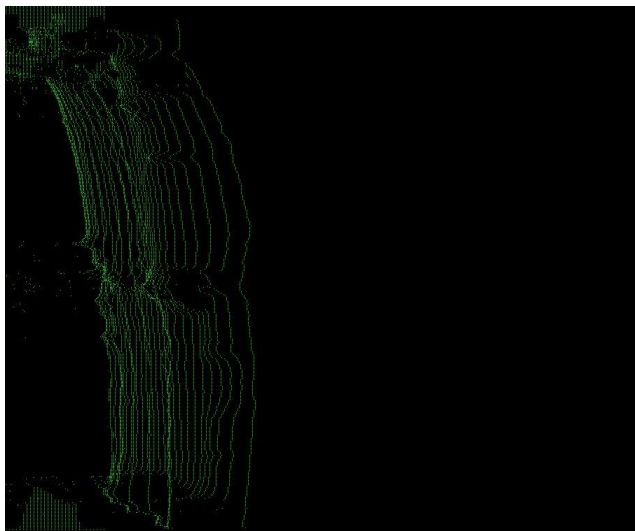
Díky tomu, že je objekt vyroben z papíru, tak má jasně bílý, ale matný povrch. Přesto odráží dostatek světla. Všechny tyto informace byly důležité pro vlastní kalibraci skeneru. Bylo nutné snížit jas pod standardní hodnotu, která normálně odpovídá světlému objektu. Dále bylo nutné upravit světelné podmínky pro osvětlení skenovaného objektu. Docházelo totiž k tomu, že při sníženém osvětlení červená linie laseru zářila ve středu bíle. Tudíž docházelo k chybným detekcím pixelů červené barvy. Toto bylo eliminováno, když jsem umístil přímý zdroj světla před objekt těsně nad kamerou. Jako další upravovaný parametr byl jas kamery, který bylo

nutné snížit. Těmito korekcemi se vytvořila na povrchu objektu již přijatelná čára z laseru, kterou už bylo snadné detekovat, jak je vidět na obrázku.



Obr. 32 Ukázka detekce bodů podél linie laseru

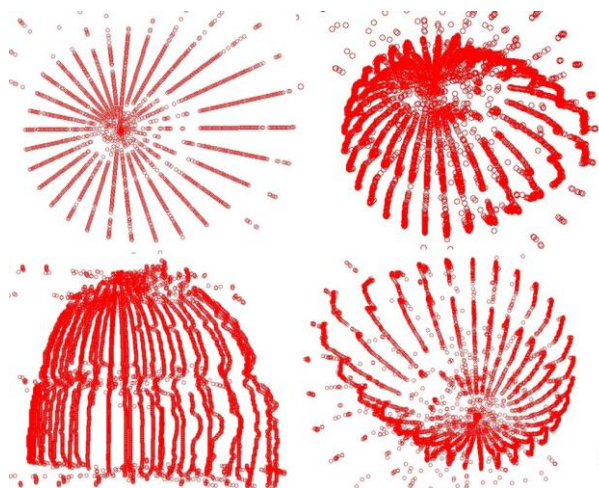
I když byla věnována dostatečná příprava na kalibraci, docházelo v průběhu skenování k ústřelům detekovaných bodů mimo požadovanou linii laseru. Docházelo k tomu například v případech, kdy díky zakřivení povrchu skenovaného objektu, nedošlo k úplnému osvitu po celé délce linie laseru. Toto je důkaz jevu, který byl popsán v oddíle „2.6.1 Triangulační metody“, konkrétněji v části „Aktivní triangulace“. Tyto ústřely jsou na obrázku se souřadnicemi jasně viditelné.



Obr. 33 Ukázka vygenerovaných souřadnic s jasně viditelnými ústřely znázorněných v 2D obraze

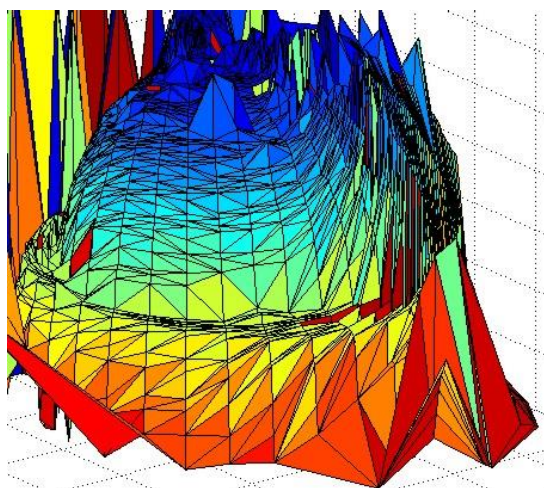
Na obrázku s prezentací mračna bodů je jasně vidět, že model je vymezen tvarem mračna a seskupením dominantních bodů do útvarů, které kopírují svým uspořá-

dáním vzhled a tvar originálního skenovaného objektu. V místech, kde skenovaný objekt má znatelnou ostrou hranu dochází už k výše zmíněným ústřelům. Tyto jsou vidět jako osamocené body oddělené vně hlavního mračna bodů. Díky použitým filtrům dochází v daném řádku mračna bodů k deformaci blízkých bodů, podle toho jak se filtry snažily daný extrém srovnat.



Obr. 34    Presentace mračna bodů – Pohled ze shora (vlevo nahoře), pohled z boku ze shora (vpravo nahoře), pohled z boku (vlevo dole) a pohled z boku zespoda (vpravo dole)

Na obrázku, který ukazuje prezentaci povrchu modelu vytvořeném z mračna bodů je zřetelně vidět, jak jsou do povrchu modelu započítány i ustřelené body. Díky tomu je model značně deformován. Ale v oblasti, kde nedocházelo k ústřelům bodů, je vidět povrch s hladkou strukturou odpovídající předloze.



Obr. 35    Presentace povrchu modelu vytvořeném z mračna bodů

Tato demonstrace mnou zkonstruovaného skeneru ukázala, že je schopen převést do digitální podoby i tvarově velmi složitý objekt. Pro dosažení dokonalejší prezen-

tace, by bylo nutné, ještě před samotným zpracováním provést vyčištění všech extrémních bodů způsobených ústřely při skenování. Tato ukázka také demonstruje technické meze skeneru, který je schopen obstojně zpracovat objekty, jejichž tvar je jednodušší a povrch neobsahuje příliš mnoho ostrých hran na předělech ploch daného objektu.



## 8 Diskuze

Stávající 3D skener určený pro prezentaci je z valné části koncipován jako laboratorní zařízení. Možná vylepšení by proto mohla začít hned u takto provedených částí. Jednou z hlavních částí je hardwarová část, která je zodpovědná za řízení krokového motoru a soustavy Laserů. V první řadě by zde bylo nutné nahradit laboratorní nepájivé pole s osazenými součástkami za desku s plošnými spoji. Dále motorový shield je využíván jen z jedné poloviny, přičemž je na straně jeho desky plošných spojů připájen i napěťový konvertor. Dochází tím k omezení možnosti připojit další periférie, například kamery prostřednictvím konektoru SPI. V rámci vylepšení se proto nabízí možnost vytvořit úplně novou desku plošných spojů. Ta by již obsahovala tištěné spoje pro zapojení jak jednoho motorového řadiče, tak i spoje pro připojení konvertoru napětí. Dále by pak byly piny, které jsou využívány pro řízení spínání laserů, přímo propojeny s obvody, které jsou nyní umístěny na nepájivém poli. Napájecí piny pro lasery by byly vyvedeny ke každému použitému laseru zvlášť. Zbylé piny z rozhraní GPIO by byly vyvedeny nad současné rozhraní GPIO mikropočítače Raspberry Pi, a umožňovaly by tak připojení dalšího rozšiřujícího modulu.

Nosná konstrukce ze stavebnice Merkur se ukázala jako velmi dobrá volba. Při budoucích úpravách bych s ní jistě počítal, protože na její platformě, se provádějí změny geometrie jednotlivých aktivních části velice snadno. Zejména pak, nahradili se místo šroubů dodávaných ke stavebnici speciální konstrukční šrouby s hlavou s vnitřním šestihranem, tak je díky tomu celá konstrukce velice pevná a stabilní.

Jako další možné vylepšení by bylo v použití další sady lineárních Laserů. Tímto by se zrychlila rychlost snímání mezi jednotlivými posuvnými kroky motoru. Zkvalitnění snímaného obrazu by mohlo být docíleno přidáním kamery o větším rozlišení a s větší citlivostí.

K lepšímu snímání by také mohlo dojít výměnou dosavadní jednoduché podložky, na které je položen skenovaný objekt, za upevňovací čelisti s vymezenou geometrií středu otáčení. Tyto čelisti by mohly být alternativně navrženy jako 3D model a vytištěny na 3D tiskárně.

Do softwarové části by mohlo být přidáno grafické rozhraní, které by umožňovalo lepší práci při kalibraci 3D skeneru. Jako další možnost, která mě napadá, je možnost fragmentovat skenovanou plochu na více segmentů a nad těmito segmenty uplatňovat jednotlivé filtry. Tímto by byl skenovaný objekt rozparcelován na oblasti s určitými geometrickými tendencemi. Například v oblasti s hladkým rovným povrchem skenovaného objektu by byly aplikovány filtry, které by danou část začistily do rovných linií, a v těch místech objektu, kde by bylo velké množství nerovností, případně ostrých hran, by nebyly aplikovány specifické typy filtrů, které by danou oblast mohly ještě více zkreslit.

Jako další vylepšení dosavadní konstrukce s cílem dosáhnout přesnějších výsledků při skenování by bylo možné prostým přidáním záštity, která by omezovala vliv vnějších světelných podmínek. K této dodatečné konstrukci by bylo možné přidat vlastní zdroje světla, které by mohly být řízeny elektronicky, přes již zmíně-

---

né grafické rozhraní. V extrémním případě by tyto zdroje světla mohly být doplněny jedním pohyblivým světlem vychylovaným obdobně jako hlavice na 3D tiskárně, nebo nahrazeny maticí světelných zdrojů, kdy by u každého segmentu této matrice byla řízena jeho intenzita, případně jeho vyřazované spektrum řízené opět přes grafické rozhraní.

## 9 Závěr

Cílem mé diplomové práce bylo zkonstruovat 3D skener a provést analýzu nad naskenovaným modelem.

V úvodu, kde popisuji teorii 3D skenování, jsem obsáhl hlavní principy, které jsou důležitými atributy pro následující práci při konstrukci 3D skeneru a při vývoji jeho aplikací. V části „Konstrukce 3D skeneru“ představuji potřebné hardwarové součásti a zařízení, které byly použity pro vytvoření samotného 3D skeneru. V části „Metodika skenování“ jsem uvedl princip funkce skenování s použitím 3D skeneru. V části „Běh aplikace“ jsem představil vlastní fungování aplikace pro skenování a analýzu. V části „Analýza naskenovaného objektu“ jsem se věnoval popisu jednotlivých aspektů ovlivňující kalibraci, samotné skenování a filtrům zpracovávající surová data. V kapitole „Skenování složitějšího objektu“ jsem ukázal možnosti využití mnou zkonstruovaného 3D skeneru, jeho nedostatky a omezení.

Podle výše uvedeného zadání mnou zhotovený 3D skener umožňuje skenování menších předmětů v rozsahu do několika jednotek centimetrů. Výsledný naskenovaný model reprezentuje skenovaný objekt jako předlohu více než zdařile.

Práci chápu jako ukázkou možností, co lze s minimem hardwarových prostředků vytvořit. Mnou zhotovený 3D skener se sice nemůže rovnat výkonem a přesností komerčním 3D skenerům, jejichž pořizovací cena začíná na hranici ceny srovnatelné s cenou za pořízení luxusního vozu, ale pro představení možností 3D skenování a za zlomek ceny plně vyhovuje. Už samotný proces studia teoretických základů a samotná tvorba 3D skeneru mě hodně obohatily a posunuly dál ve v mých znalostech.

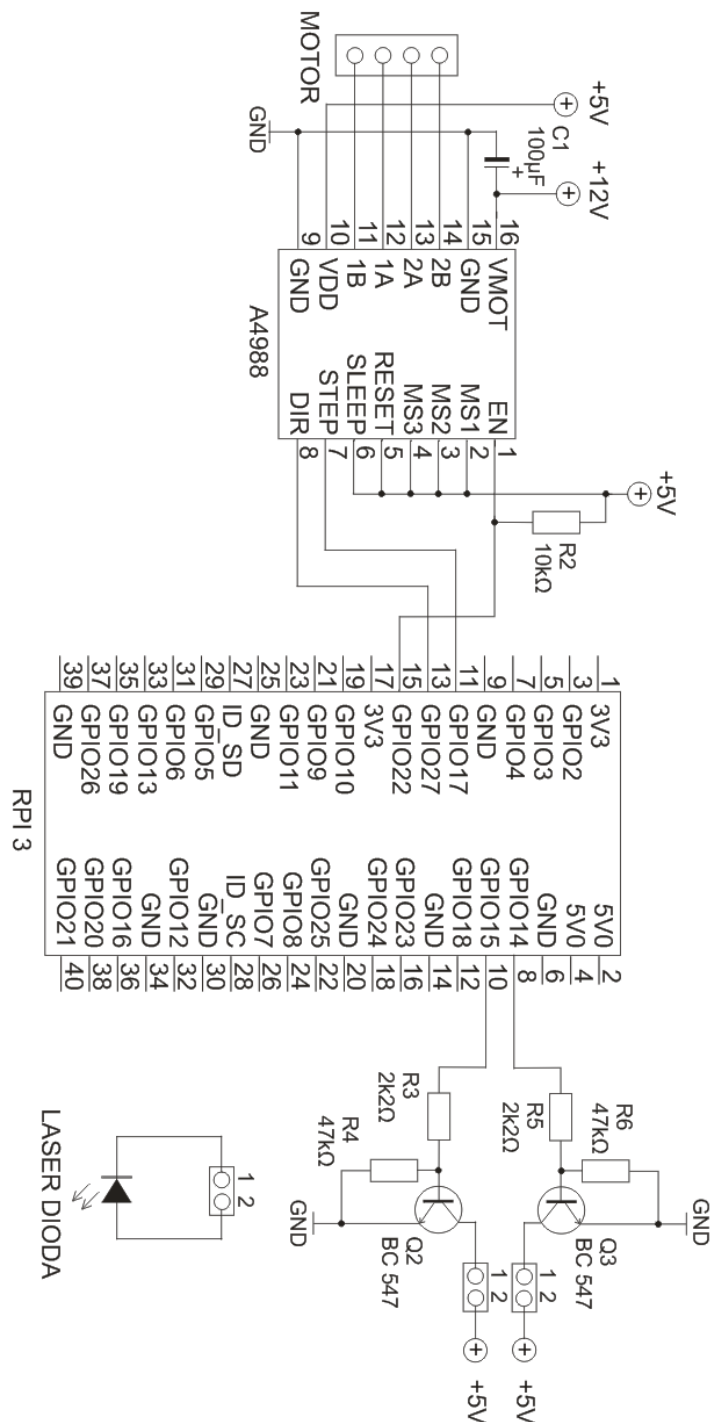
## 10 Literatura

- ČERNECKÝ, J., PIVARČIOVÁ, E. *Holografia a jej technické aplikácie* [online]. 2002. Dostupné z URL: <<http://www.holografia.szm.sk>>.
- DELAUNAY, B. *Sur la sphère vide. A la mémoire de Georges Voronoi*". In: Bulletin de l'Académie des Sciences de l'URSS. Classe des sciences mathématiques et na, 1934, no. 6, s. 793–800.
- HAUBECKER, H. – Geißler, P. *Handbook of Computer Vision and Applications: Volume 1, Sensors and Imaging*. Edited by Jähne Bernd. San Diego: Academic press, c1999. 657 p. ISBN 0-12-379771-3 (v 1).
- HLAVÁČ, V., ŠONKA, M. *Počítačové vidění*. 1. vyd. Praha: Grada, 1992. 272 s. ISBN 80-85424-67-3.
- KALOVÁ, I. *Quality Inspection of Passenger Car Wheels*. Brno: VUT Vutium, 2004. 376 - 380 s. ISBN 80-214-2636-5.
- KOSKA, B. *Optoelektronické metody 3D zaměření povrchů předmětů* : disertační práce. Praha : ČVUT, Fakulta stavební, 2009. 161 l., 21 l. Příloh. Školitel Jiří Pospíšil.
- KOZÁK, J. *Testování parametrů základnových skenovacích systémů na Fsv*, ČVUT. Vedoucí práce Bronislav Koska. [online]. 2013 [cit 2016-2-28] Dostupné z URL: <<http://geo.fsv.cvut.cz/proj/dp/2013/jakub-kozak-dp-2013.pdf>>.
- MACENAUER, A. *Technologie Super CCD SR*. [online]. 2003 [cit. 2016-03-07]. Dostupné z: <<https://www.fotoaparát.cz/clanek/76/technologie-super-ccd-sr-2183/>>.
- NIKOS, D. *Interferenční světla* [online]. 1998. Dostupné z URL: <<http://vega.fjfi.cvut.cz/docs/sfbe/optika/node7.html>>.
- NEFF, O. *Co je to CMOS*. [online]. 2008 [cit. 2016-03-07]. Dostupné z: <<http://www.digineff.cz/cojeto/cmos/cmos.html>>.
- POŠPÍŠIL, J., ŠTRONER, M. *Soubor přednášek k předmětu 154LSK (Laserové skenování) na FSV*, ČVUT [online]. 2008-2009 [cit 2013-02-20] Dostupné z URL: <<http://k154.fsv.cvut.cz/vyuka/geo/dezie/lisk.php>>.
- ŠTRONER, M., POSPÍŠIL, J. *Terestrické skenovací systémy*. Praha: ČVUT v Praze, 2008. 185stran. ISBN 978-80-01-04141-3.
- URBAN, R. *3D skenovací systém s využitím virtuálních binárních značek : studie disertační práce*. Praha : ČVUT, Fakulta stavební, 2010. 20 l. Školitel Martin Štroner.
- VOJÁČEK, A. *Principy a málo známé vlastnosti CCD snímačů obrazu* [online]. [cit. 2008-29-04]. Dostupné z URL: <<http://automatizace.hw.cz/mereni-a-regulace/ART235-principy-a-malo-zname-vlastnosti-ccd-snimacu-obrazu.html>>.

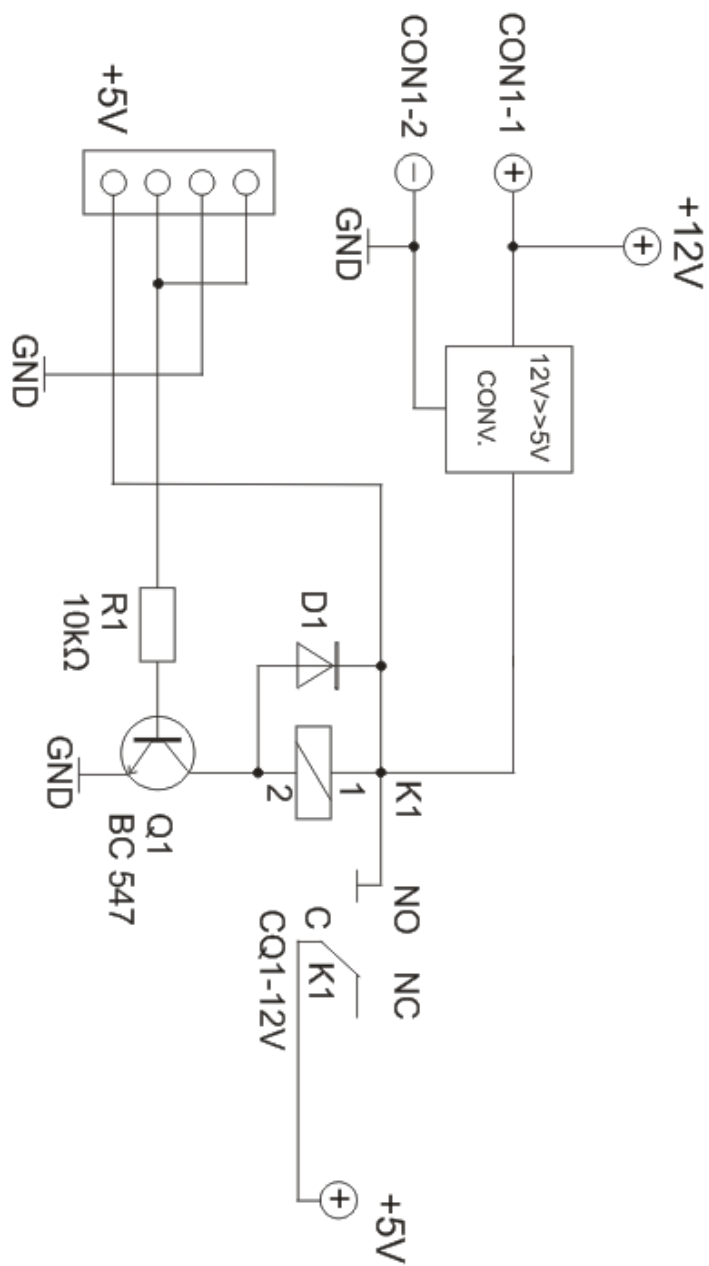
VERNON, D. *Machine Vision*. New York: Prentice Hall, 1991. 255 p. ISBN 0-13-543398-3.

# **Přílohy**

## A Schéma zapojení motorového shieldu a jeho periférií

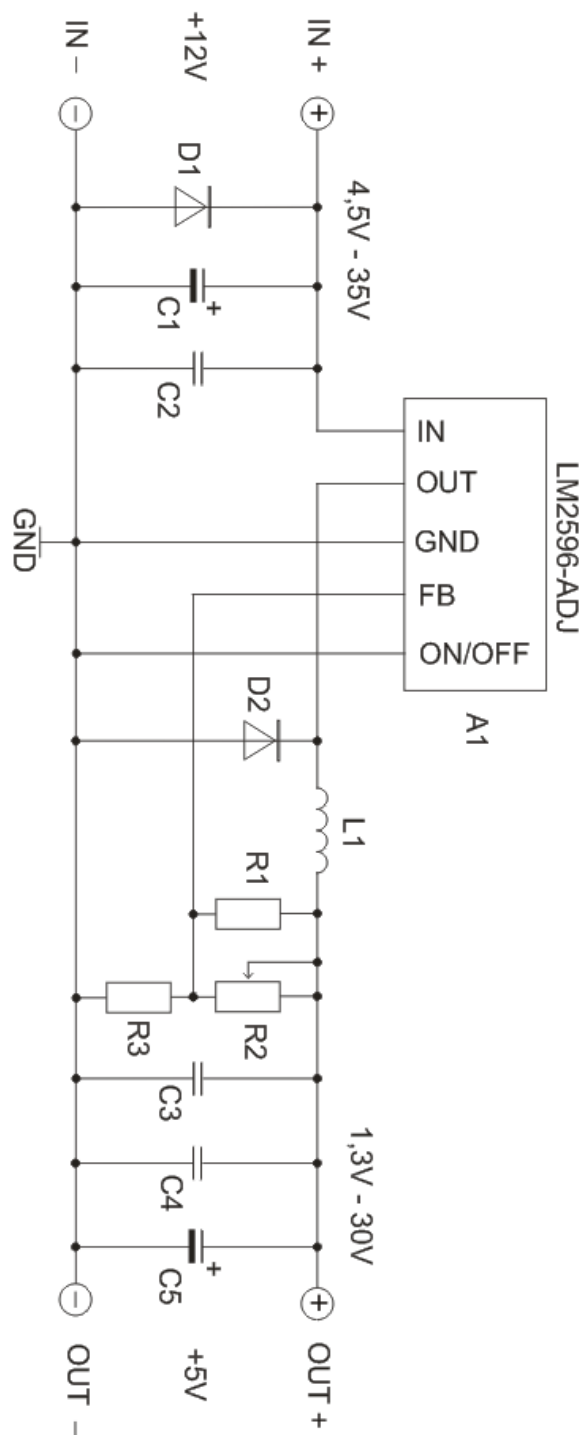


Obr. 36 Schéma zapojení motorového shieldu a bloku řízení laserů

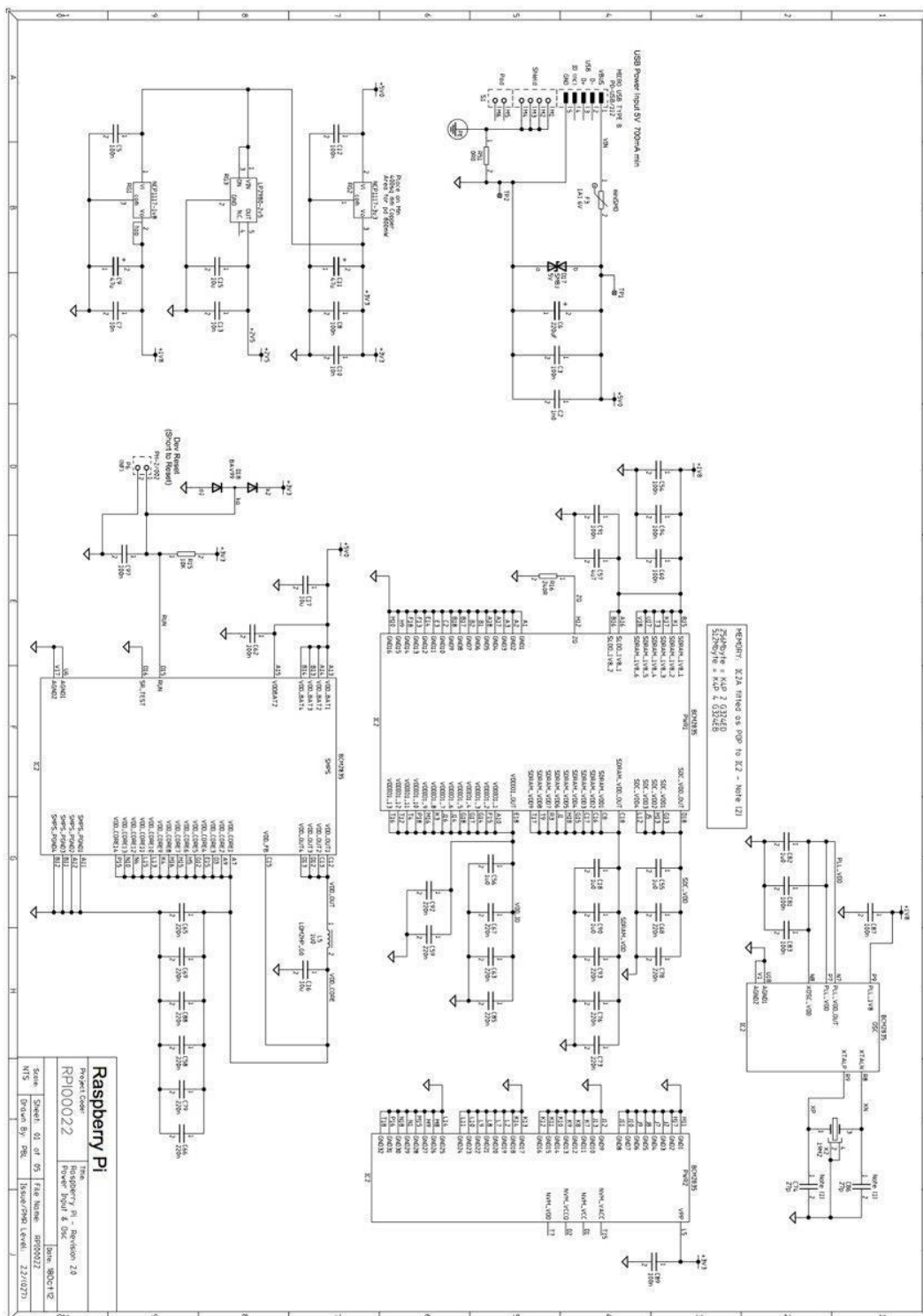


Obr. 37 Schéma zapojení konvertoru se spínacím relé

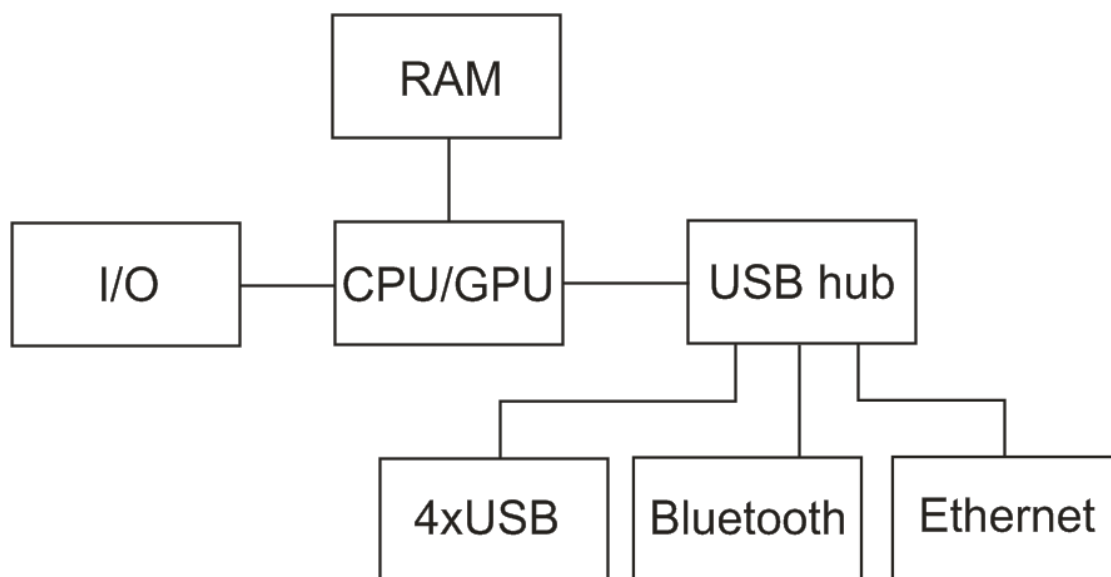




Obr. 38 Schéma zapojení vlastního modulu konvertoru s jednotkou LM2596-ADJ

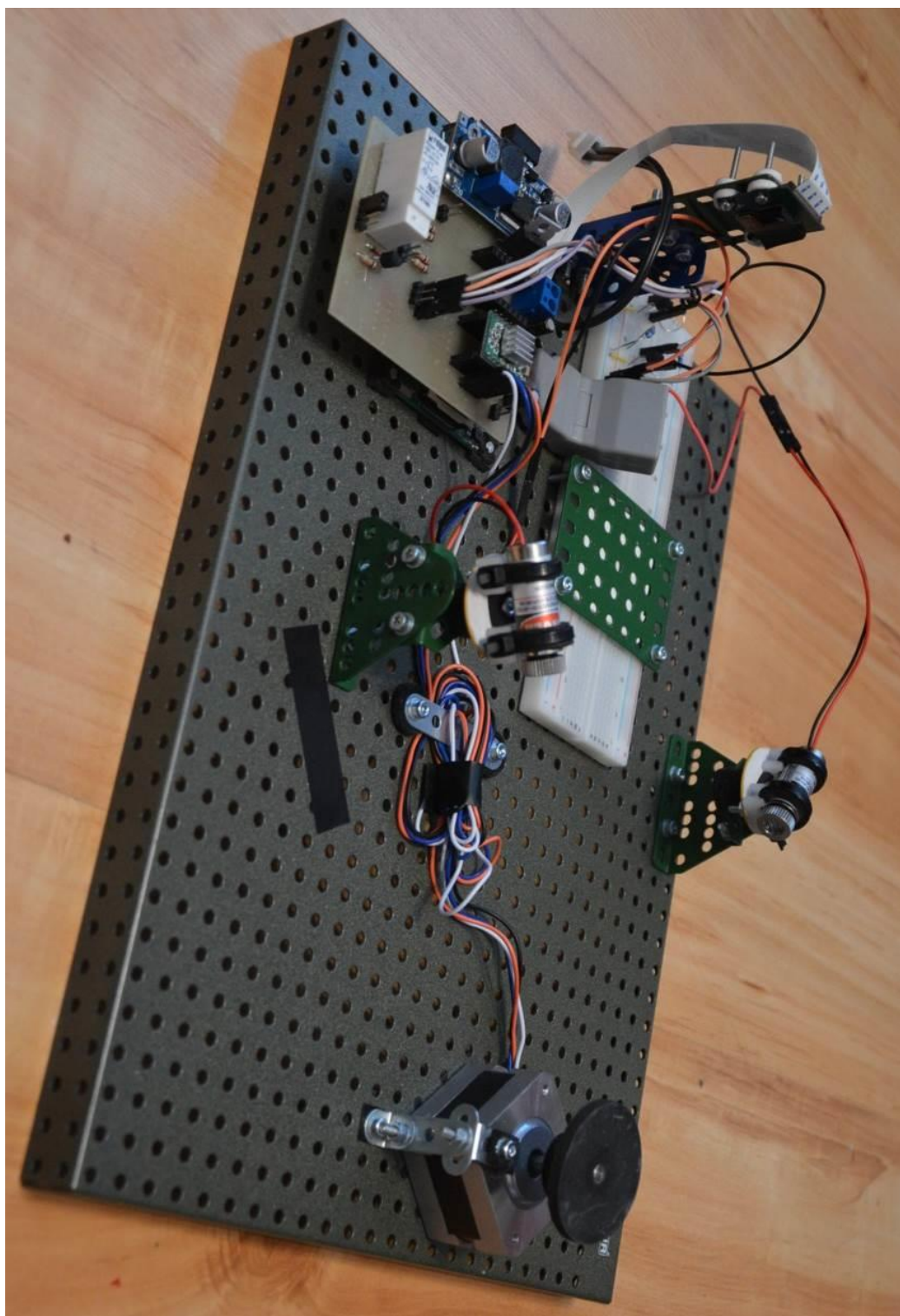


Obr. 39 Schéma zapojení Raspberry Pi  
Zdroj: (Raspberrypi.org, 2016)



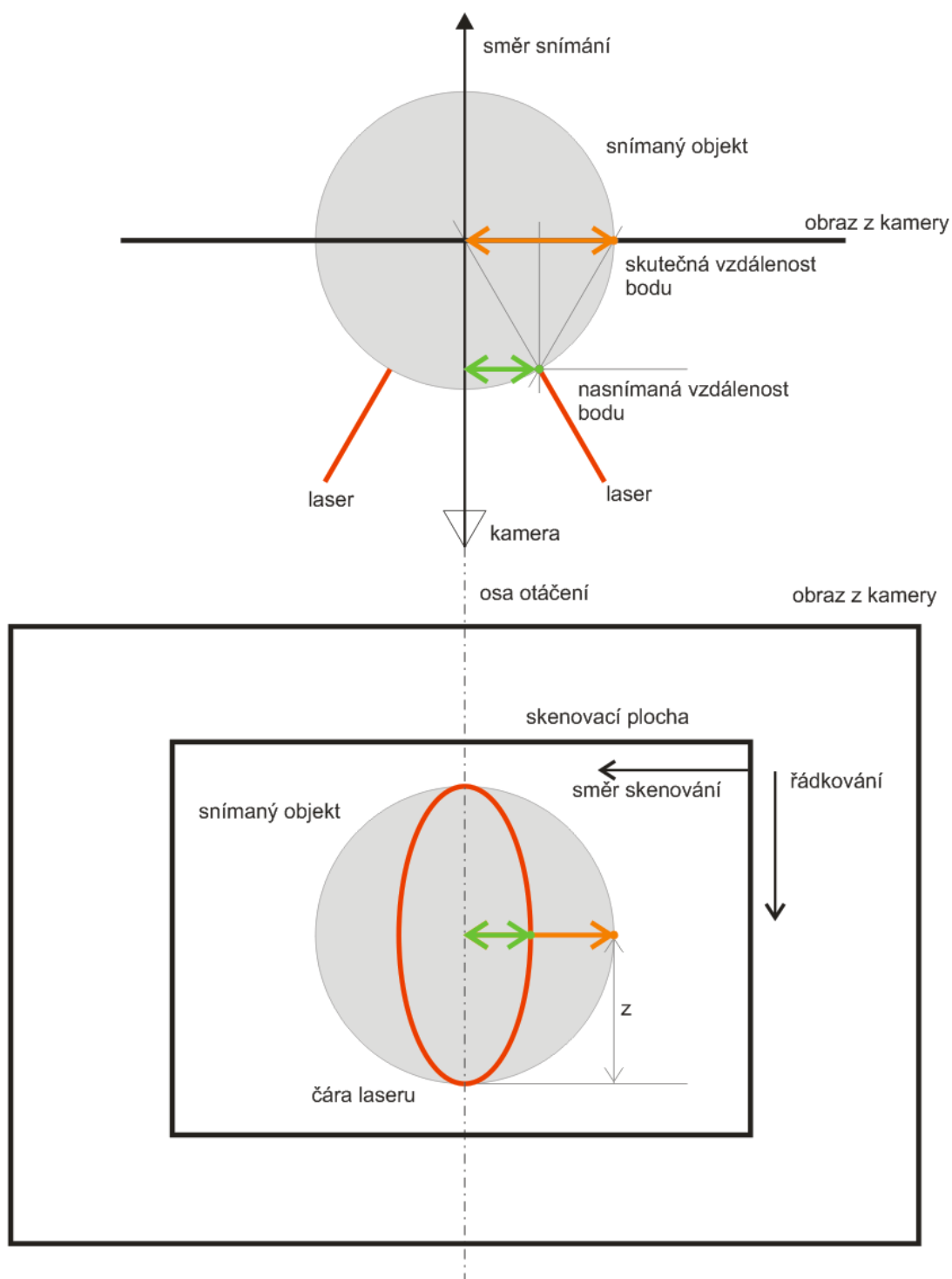
Obr. 40 Blokové schéma Raspberry Pi 3

## B 3D skener



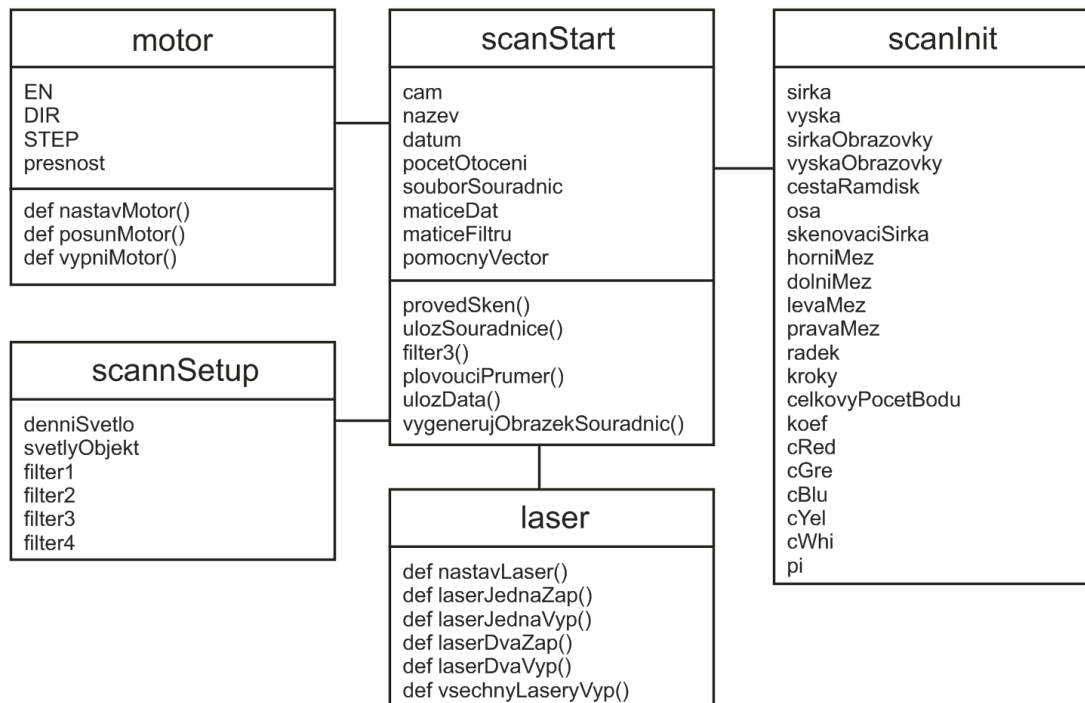
Obr. 41 Zhotovený 3D skener

## C Metodika skenování



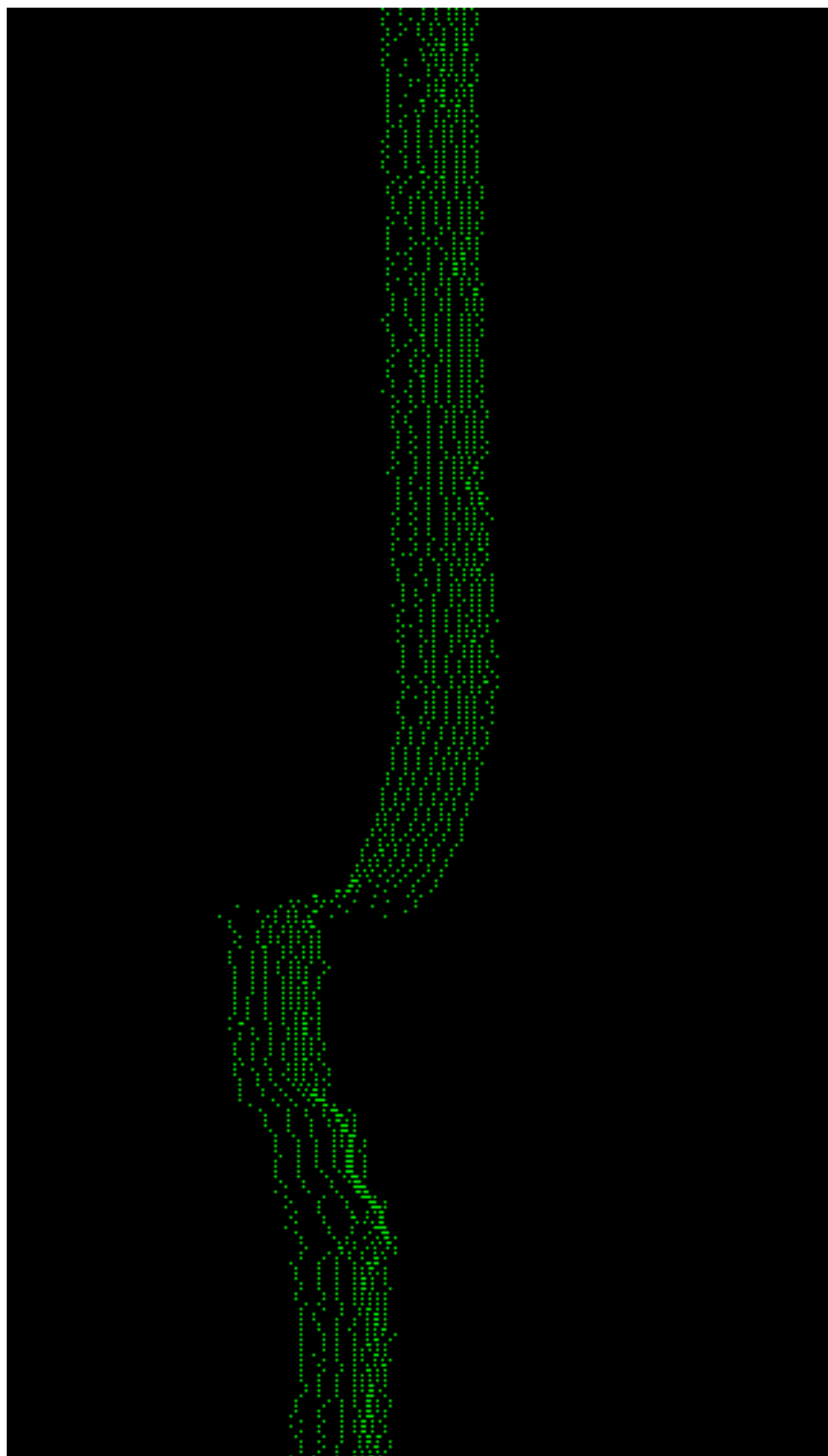
Obr. 42 Schéma metodiky skenování – pohled ze shora (nahore) a pohled ze směru kamery (dole)

## D UML schéma aplikace

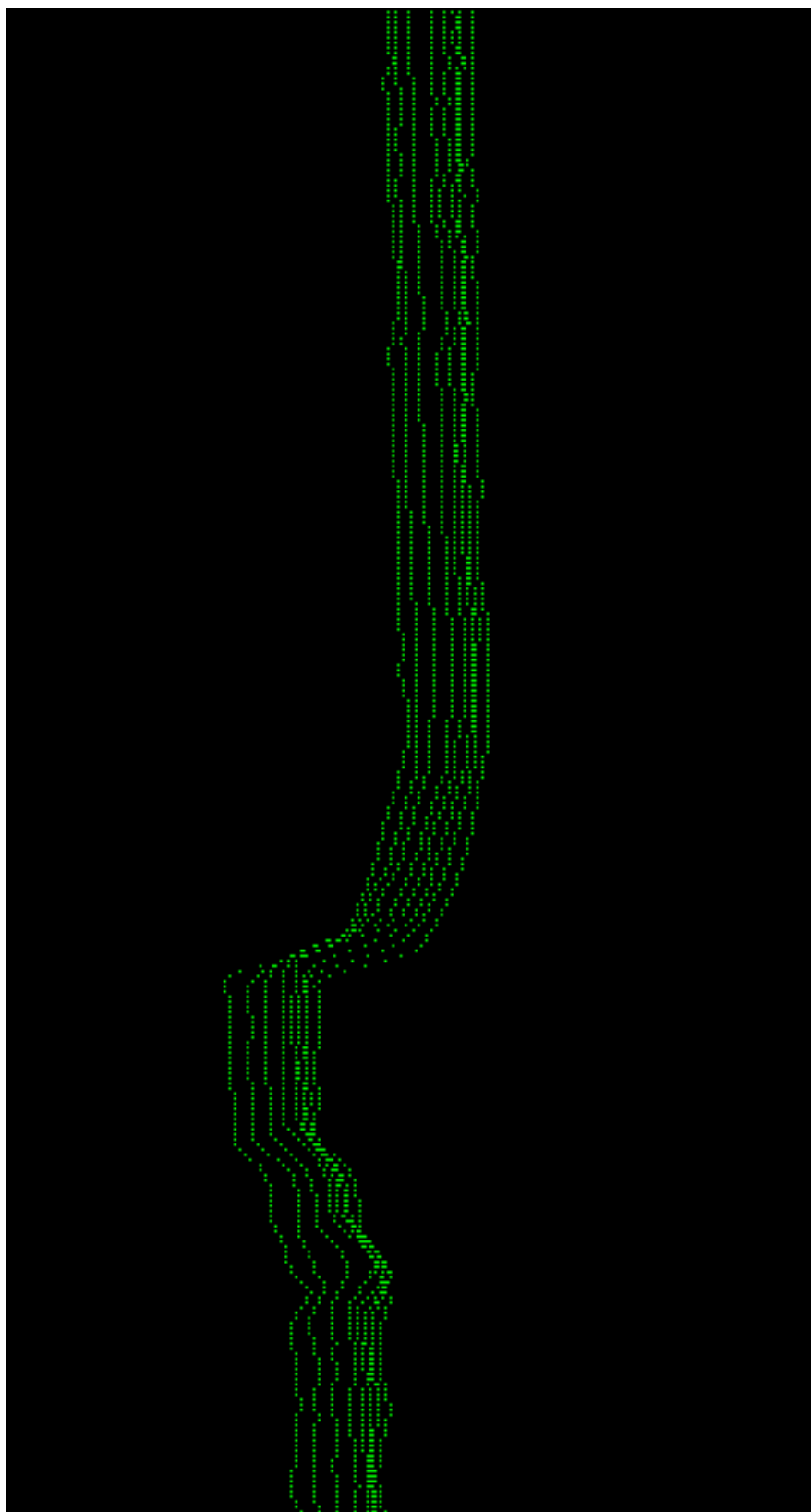


Obr. 43 UML schéma aplikace

## E Ukázky z analýzy



Obr. 44 Ukázka vygenerovaných souřadnic bez užití filtrů

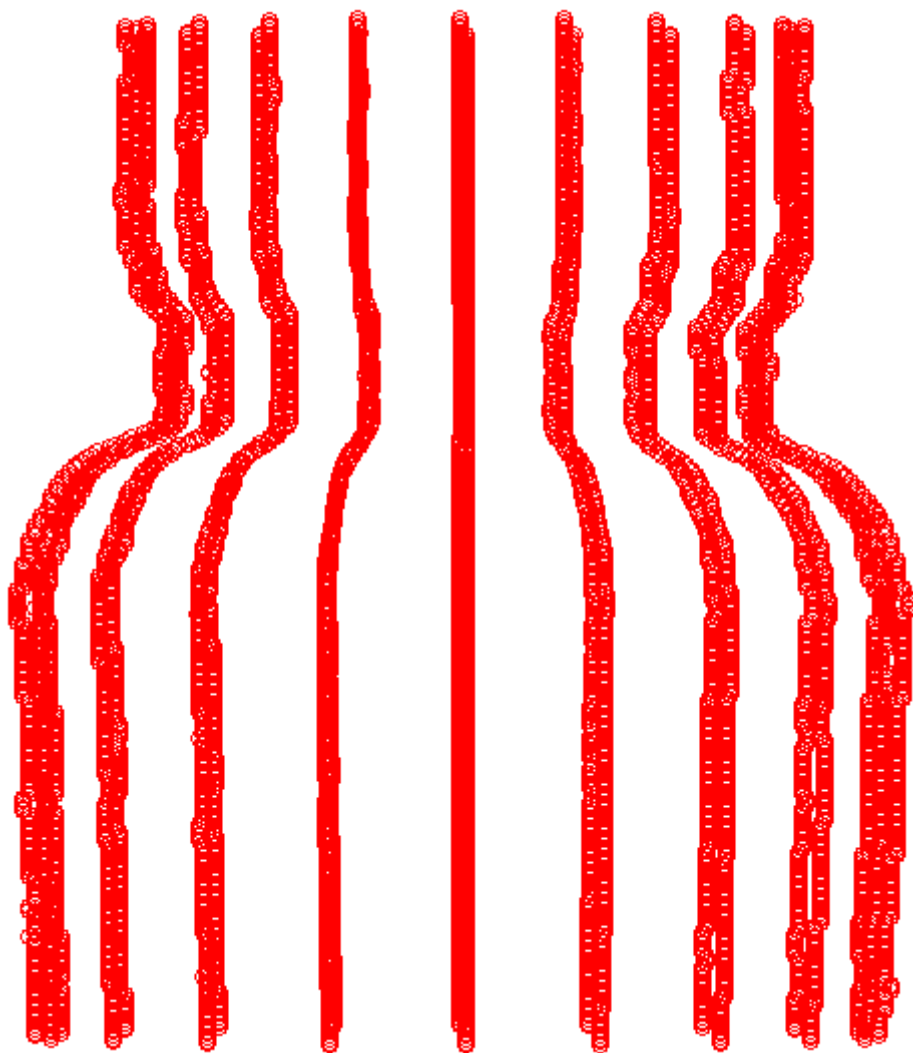


Obr. 45 Ukázka vygenerovaných souřadnic s užitím Filtru 1 a Filtru 2

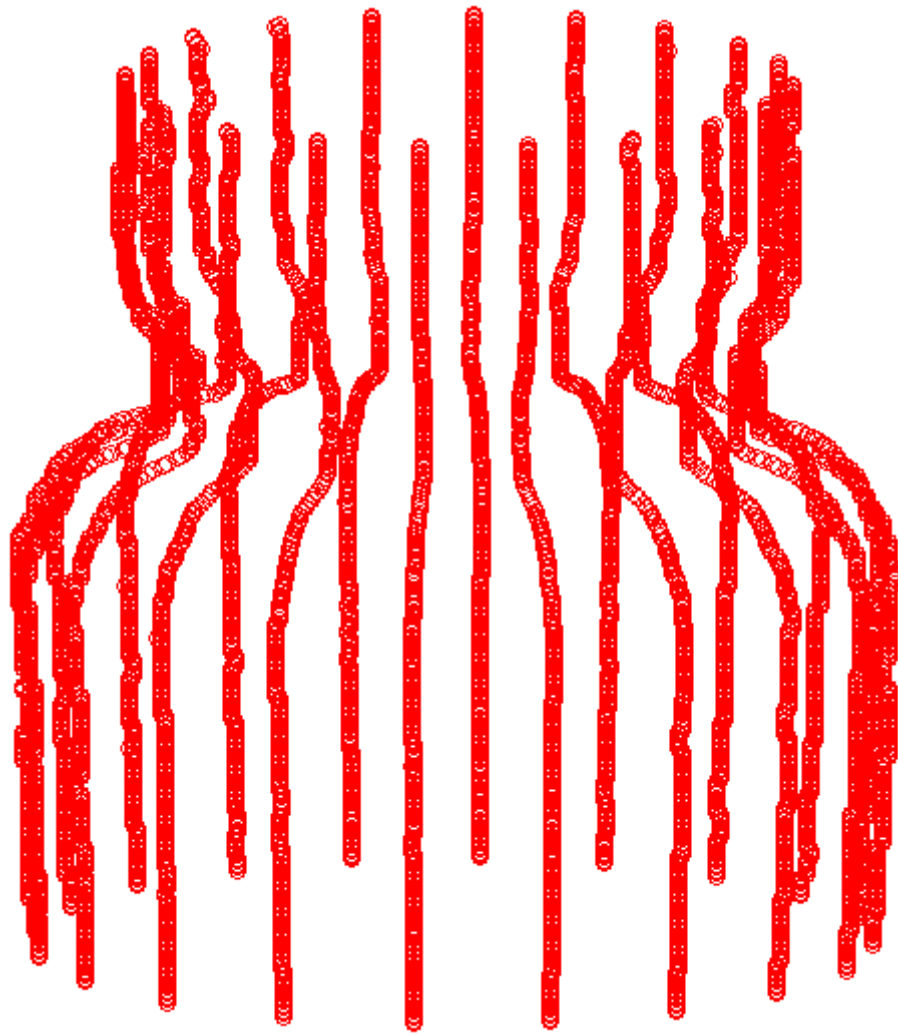




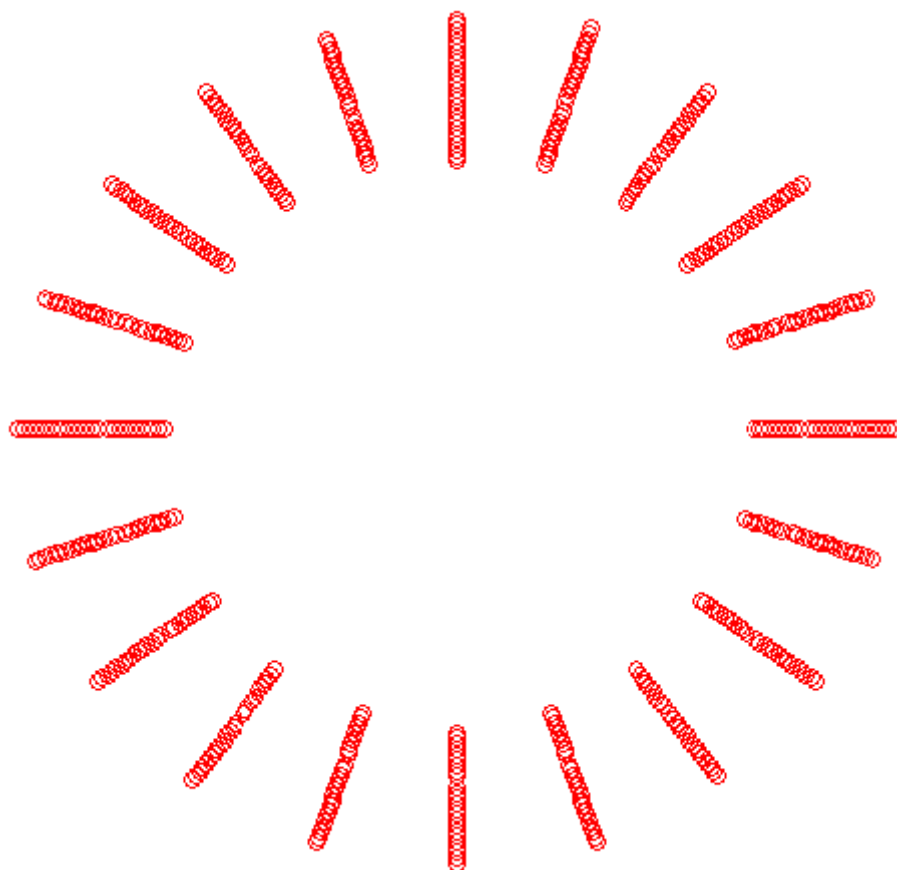
Obr. 46 Ukázka vygenerovaných souřadnic s užitím všech filtrů



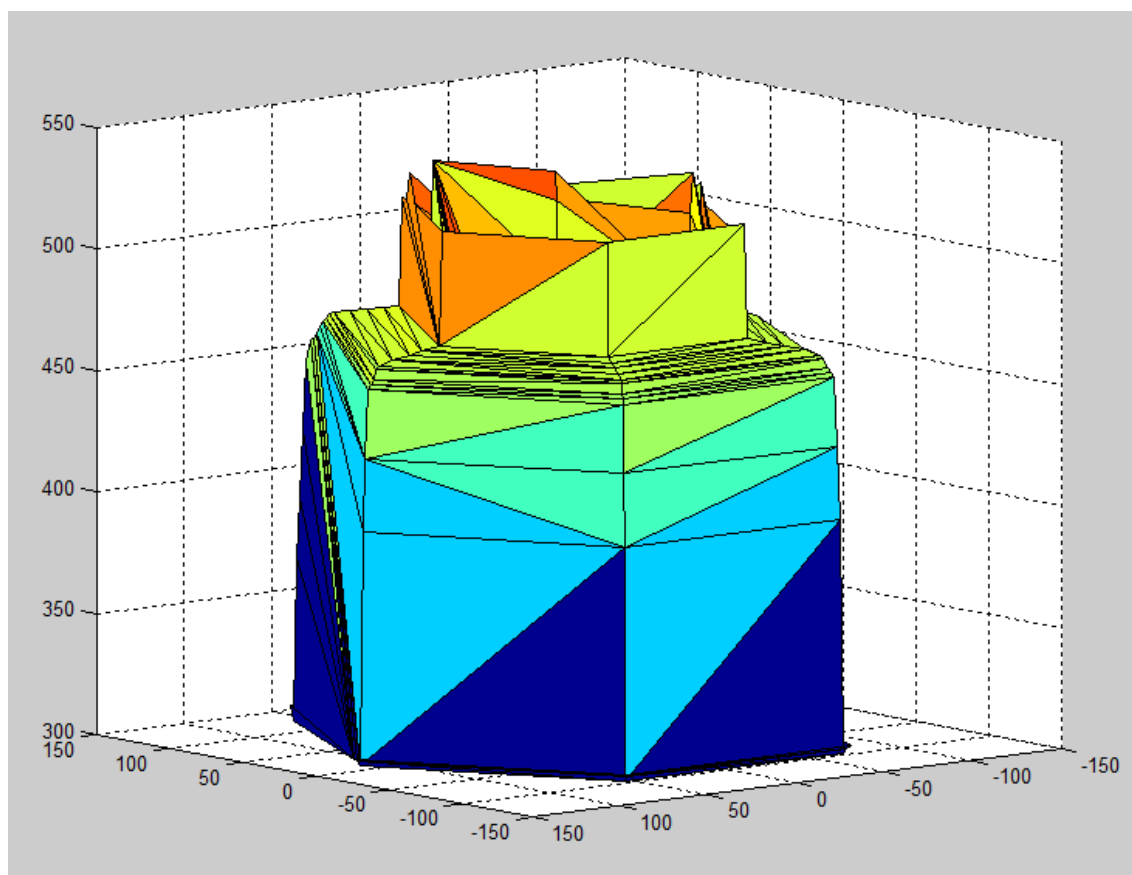
Obr. 47 Ukázka mračna bodů pohled z boku



Obr. 48 Ukázka mračna bodů pohled ze strany z vrchu

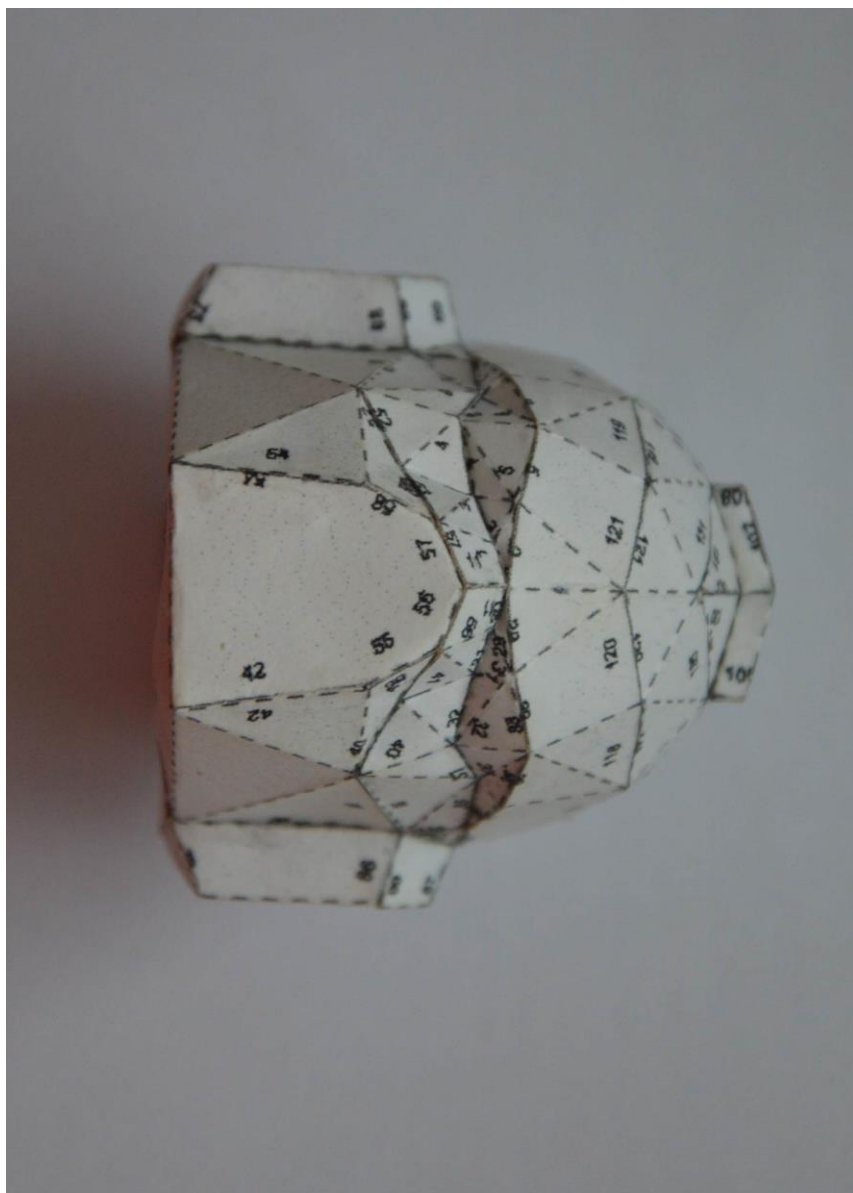


Obr. 49 Ukázka mračna bodů pohled ze shora

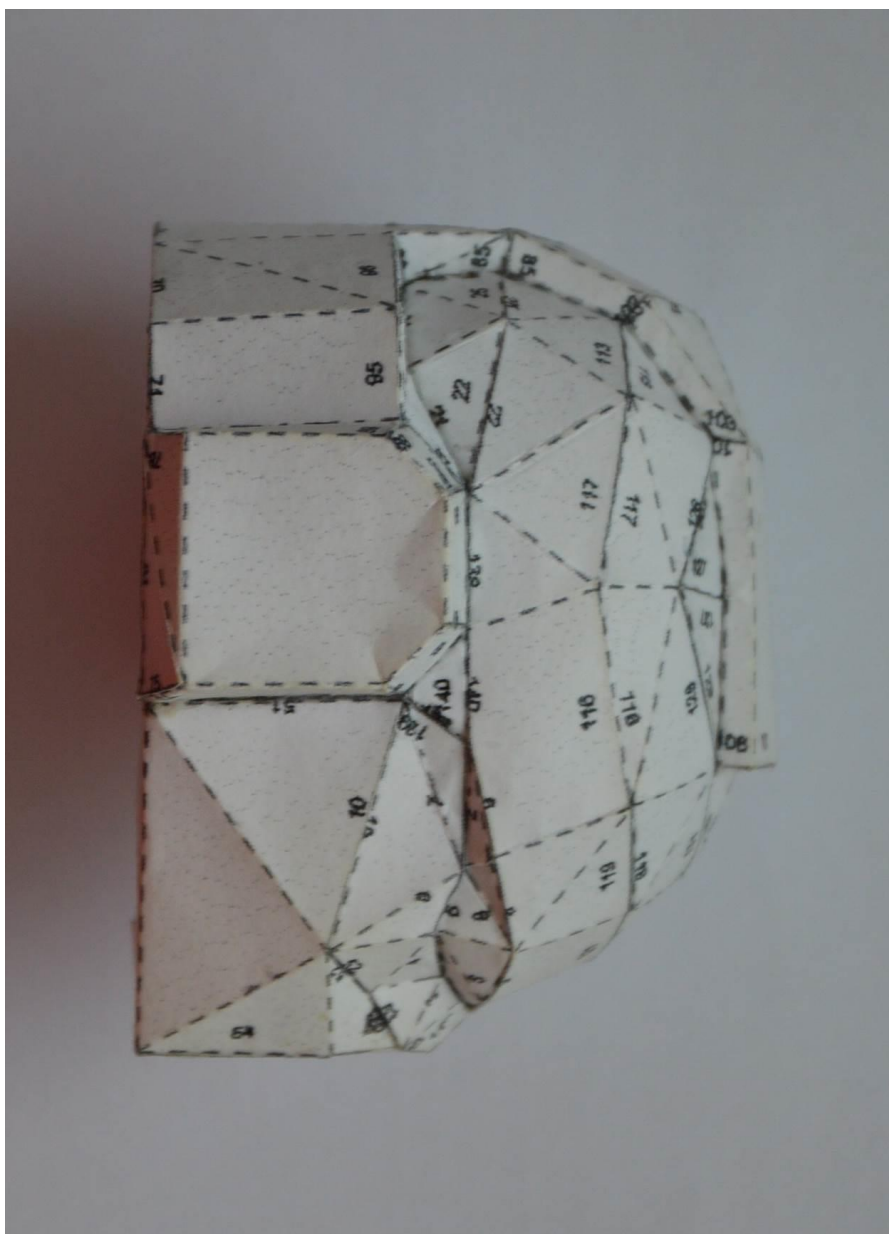


Obr. 50 Ukázka modelu s vygenerovaným povrchem

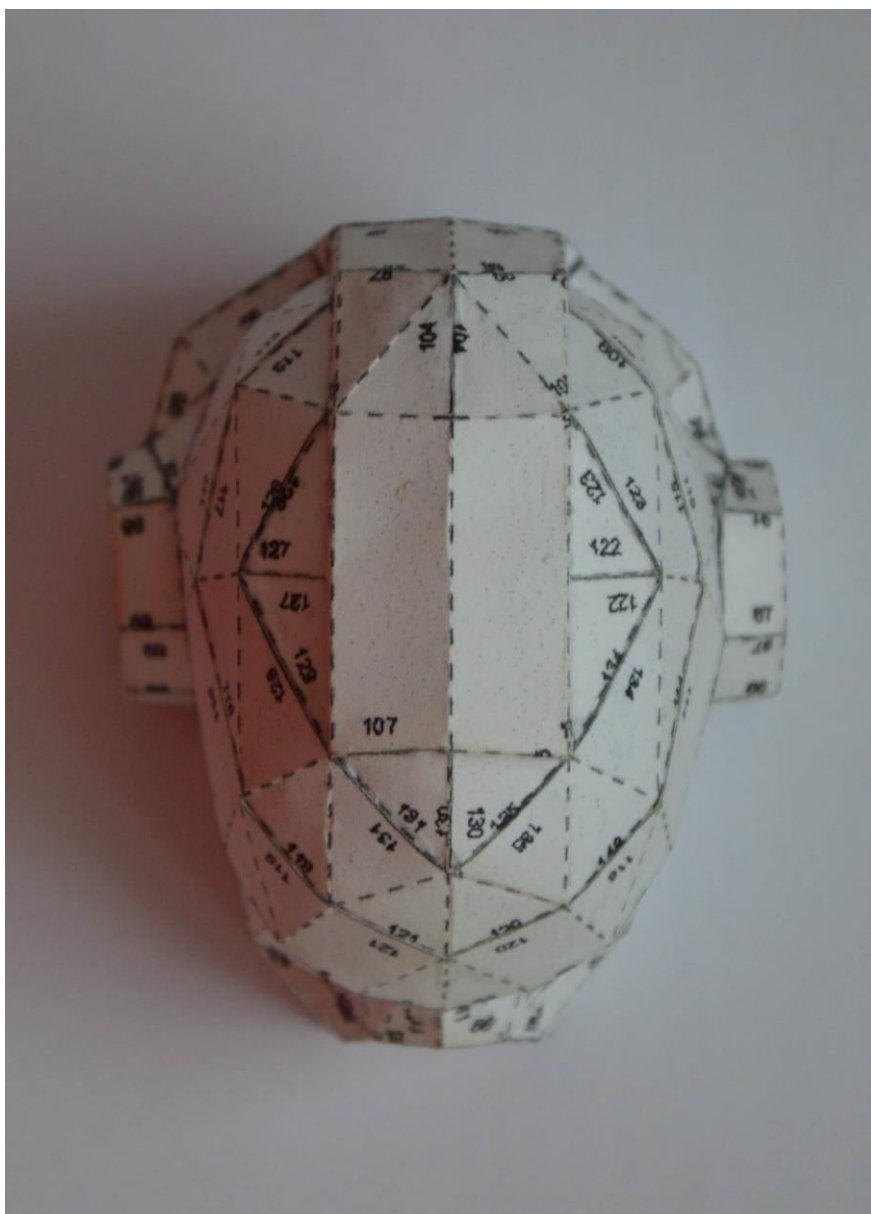
## F Tvarově členitější objekt



Obr. 51 Tvarově složitější objekt pohled zepředu

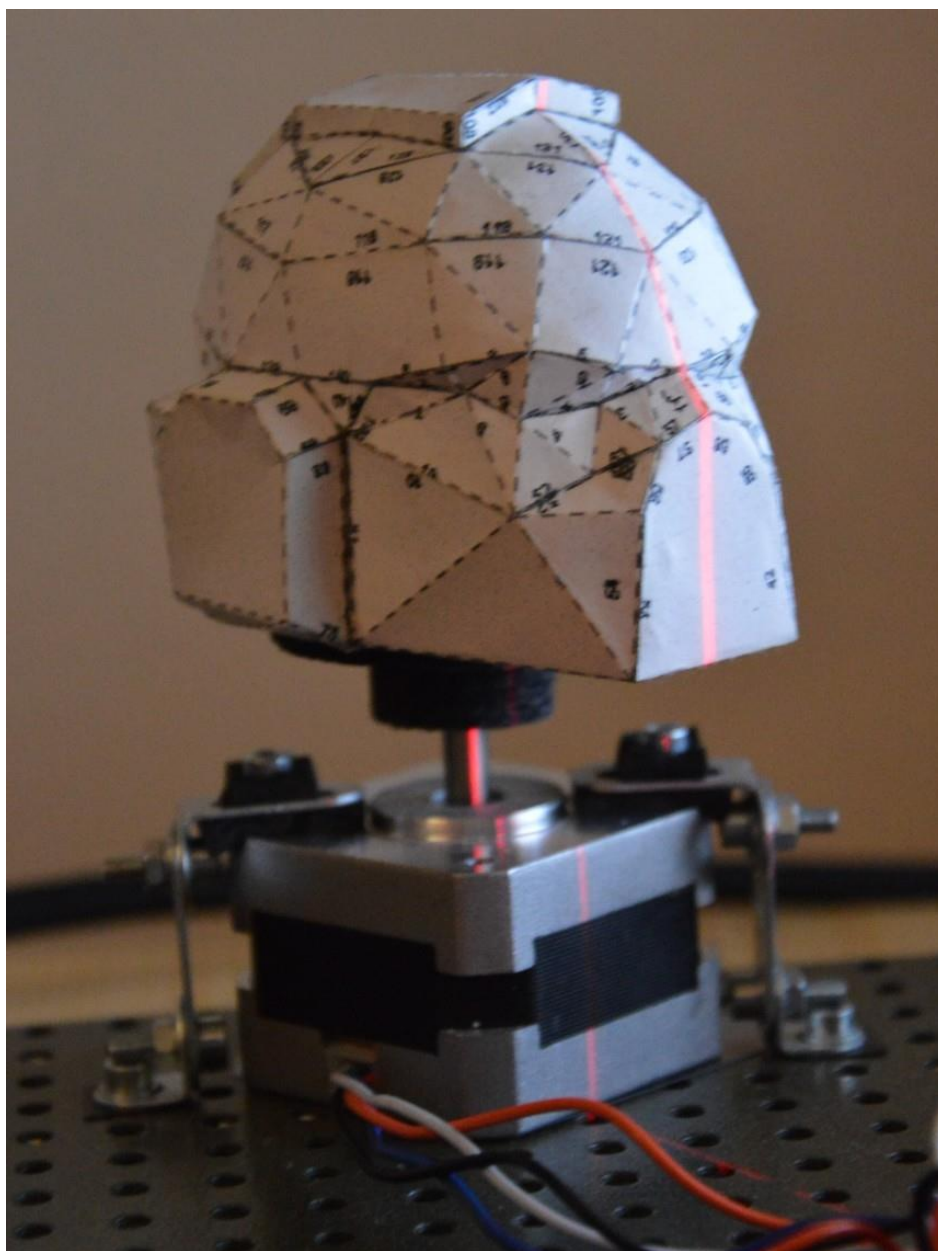


Obr. 52 Tvarově složitější objekt pohled z boku

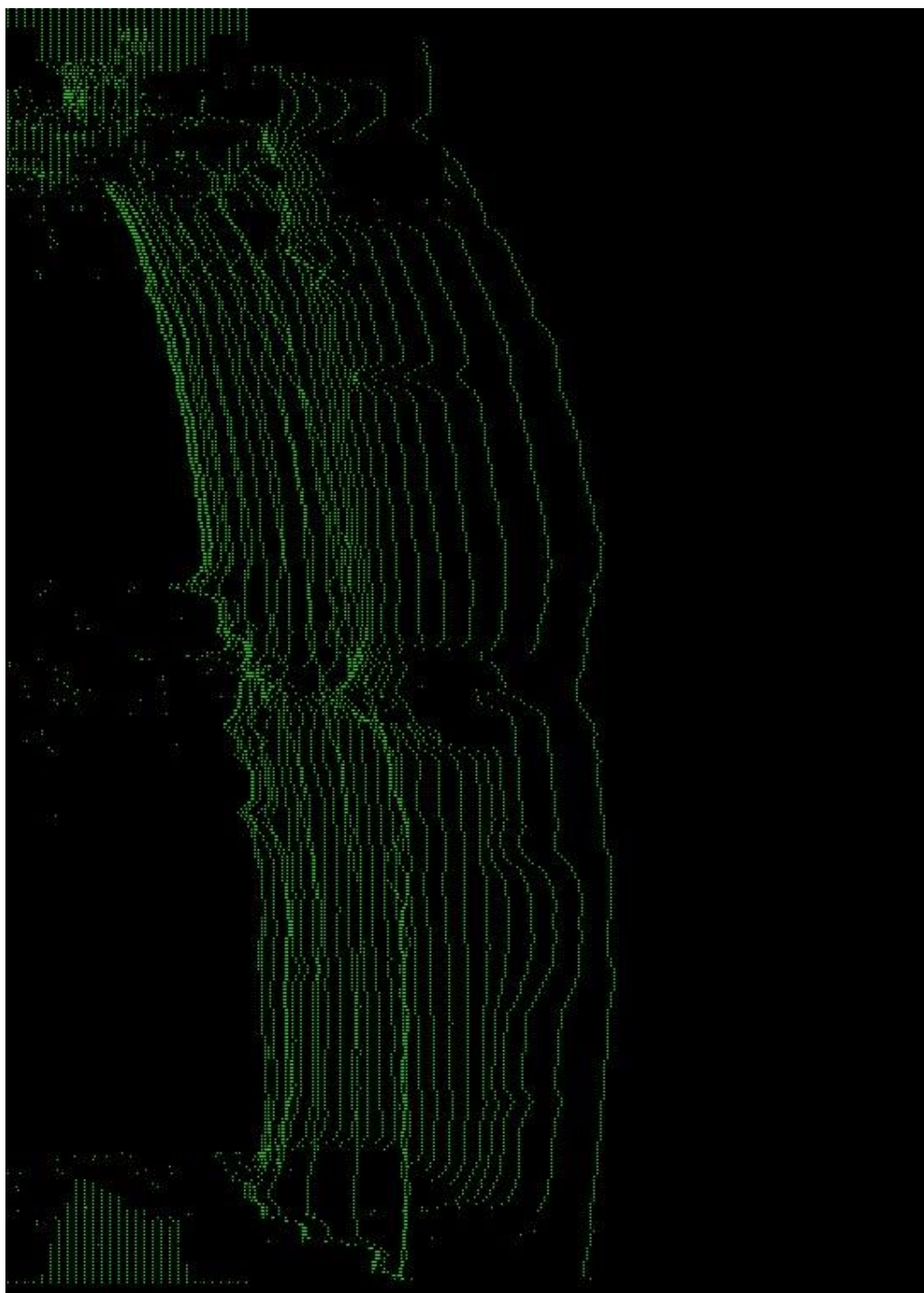


Obr. 53 Tvarově složitější objekt pohled ze shora

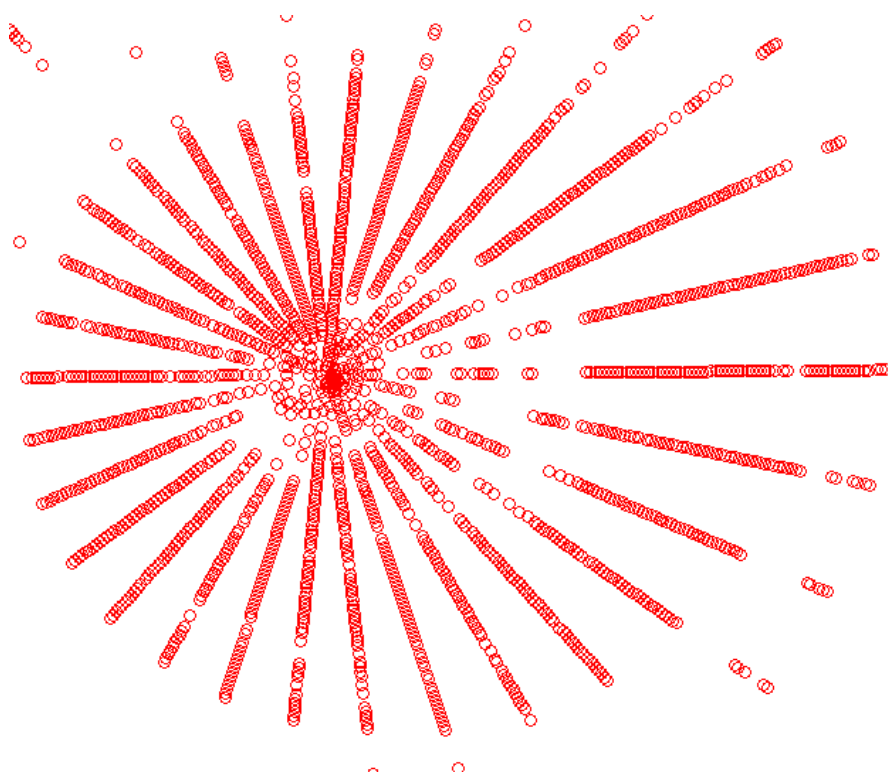




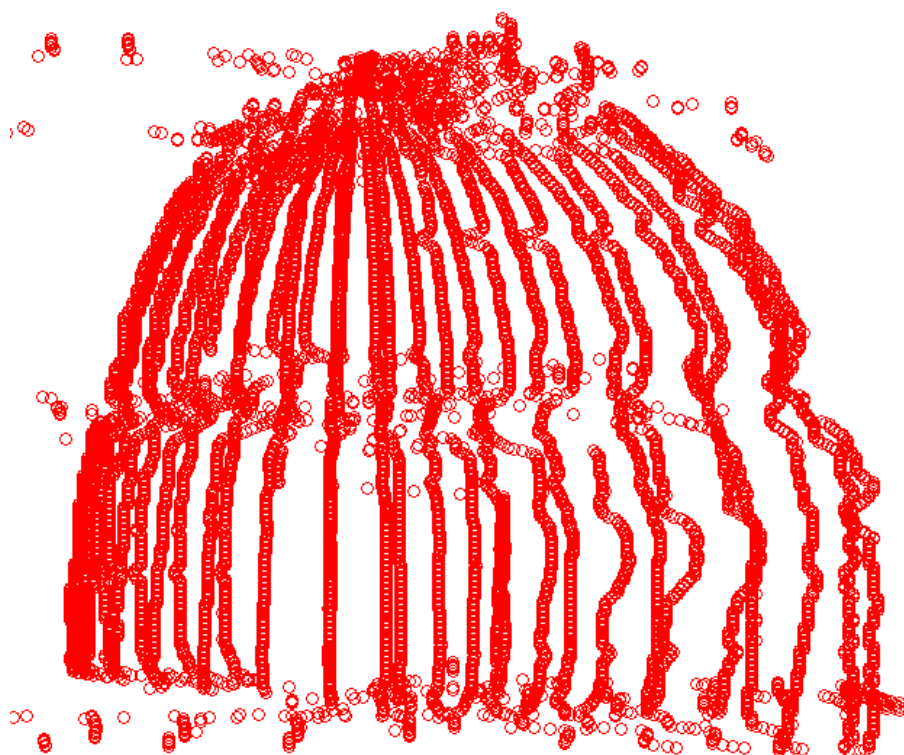
Obr. 54 Tvarově složitější objekt usazený na podstavci



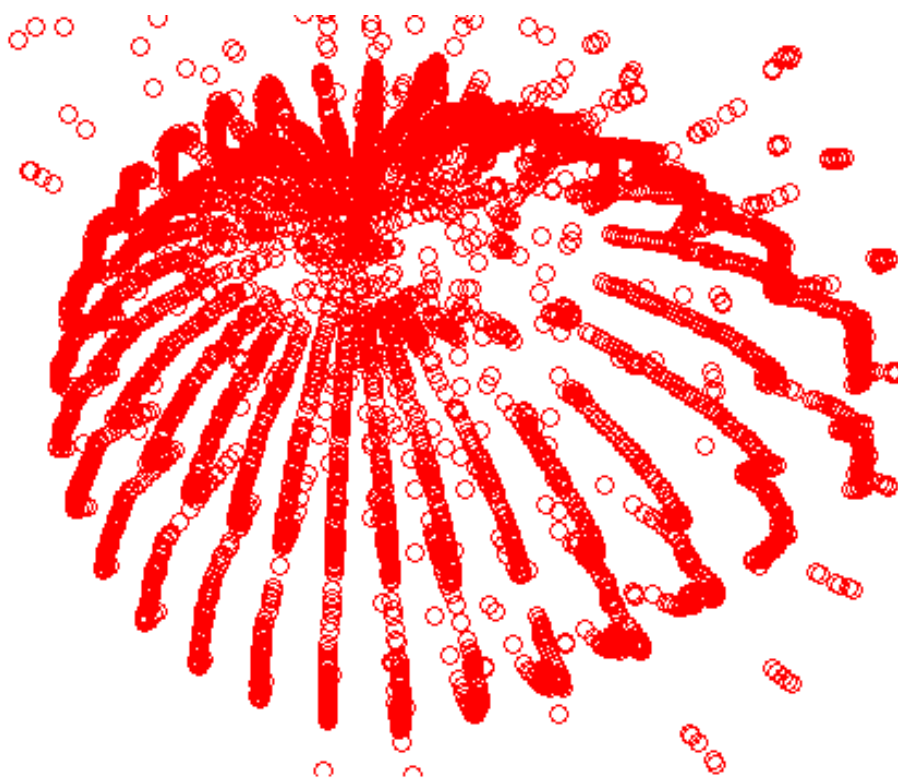
Obr. 55 Ukázka vygenerovaných souřadnic v 2D zobrazení



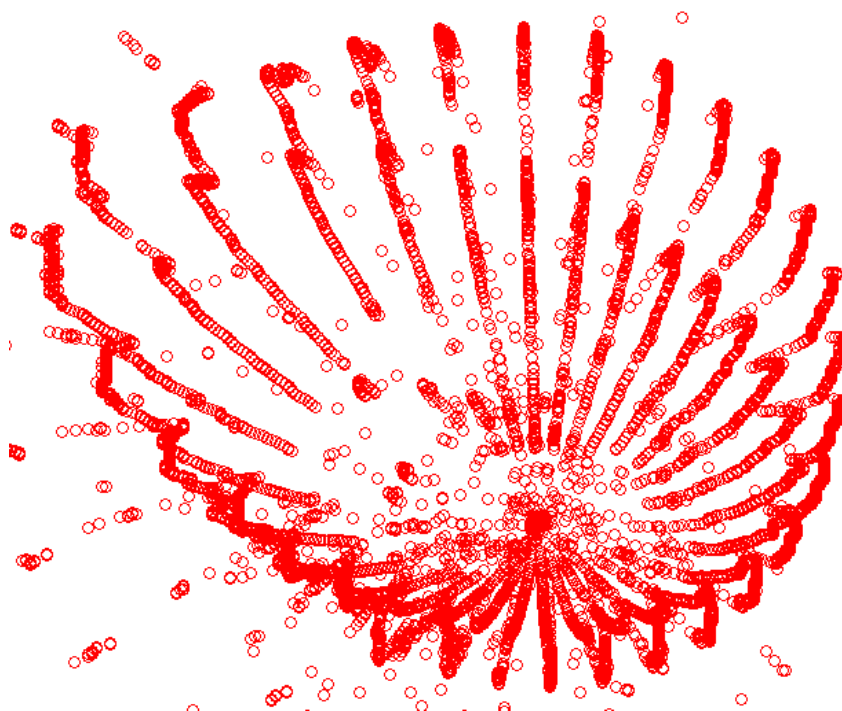
Obr. 56 Ukázka mračna bodů pohled ze shora



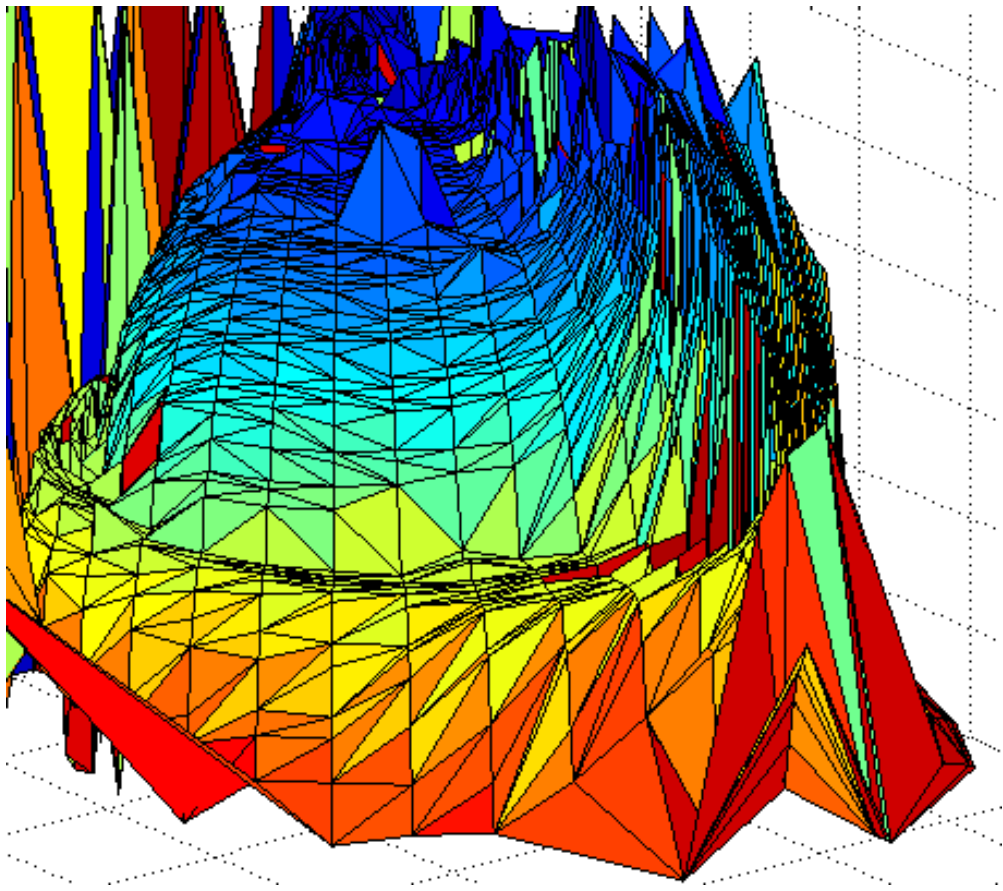
Obr. 57 Ukázka mračna bodů pohled z boku



Obr. 58 Ukázka mračna bodů pohled z boku ze shora



Obr. 59 Ukázka mračna bodů pohled z boku zespoda



Obr. 60 Ukázka povrchu modelu vytvořeném z mračna bodů

## G Ukázka kódu aplikace

Ukázka kódu řešící řízení laserů:

```
import RPi.GPIO as GPIO
from time import sleep

def nastavLaser():
    GPIO.setmode(GPIO.BCM)
    GPIO.setwarnings(False)
    GPIO.setup(14, GPIO.OUT)
    GPIO.setup(15, GPIO.OUT)

def laserJednaZap():
    GPIO.output(14, 1)

def laserJednaVyp():
    GPIO.output(14, 0)

def laserDvaZap():
    GPIO.output(15, 1)

def laserDvaVyp():
    GPIO.output(15, 0)

def vsechnyLaseryVyp():
    GPIO.output(14, 0)
    GPIO.output(15, 0)
```

Ukázka kódu řešící řízení motoru:

```
import time
from time import sleep
import RPi.GPIO as GPIO

EN = 22
DIR = 27
STEP = 17
presnost=0.00001

def nastavMotor():
    global EN,DIR,STEP
    GPIO.setwarnings(False)
    GPIO.setmode(GPIO.BCM)
```

```
GPIO.setup(EN, GPIO.OUT)
GPIO.setup(DIR, GPIO.OUT)
GPIO.setup(STEP, GPIO.OUT)

def posunMotor(kroky, rychlost):
    global EN, DIR, STEP, presnost
    nastaveni=2
    GPIO.output(EN, False)
    GPIO.output(DIR, False)
    for tx in range(kroky*nastaveni):
        time.sleep(presnost*rychlost)
        GPIO.output(STEP, True)
        time.sleep(presnost*rychlost)
        GPIO.output(STEP, False)
        GPIO.output(EN, True)

def vypniMotor():
    GPIO.output(EN, True)
```

Ukázka kódu určená pro nastavení hlavních proměnných:

```
import math

#rozliseni obrazu
sirka = 1600
vyska = 1200
sirkaObrazovky = sirka
vyskaObrazovky = vyska

cestaRamdisk = "/home/pi/ramdisk/"

osa = sirka/2-100
skenovaciSirka = 200
horniMez = 300
dolniMez = 600
levaMez = osa-skenovaciSirka
pravaMez = sirka-osa-skenovaciSirka-20

#celkovy pocet radek ve skenovaci plose
radek = vyska-horniMez-dolniMez+1

#krokovani radku ve skenovaci plose
kroky=1

#pocet naskenovanych bodu celkem
```

```
celkovyPocetBodu=0

koef = 1.55

#definovani barev pro vykreslovani
cRed=(255,0,0)
cGre=(0,255,0)
cBlu=(0,0,255)
cYel=(255,255,0)
cWhi=(255,255,255)

#definovani matematicke promene
pi=math.pi
```

Ukázka kódu určená pro kalibraci 3D skeneru:

```
#prepinace svetelných podmínek
denniSvetlo=1
svetlyObjekt=1

#prepinace filteru
filter1=1
filter2=1
filter3=1
filter4=1

#nastaveni barev

if denniSvetlo:
    fR=60
    fG=64
    fB=64
else:
    fR=90
    fG=50
    fB=50
```

Ukázka kódu řešící samotný chod aplikace:

```
import os, sys, time, math, pygame
from datetime import datetime
from time import sleep
from motor import *
from laser import *
from scannSetup import *
from scannInit import *
#-inicializace modulu pygame
```



```
pygame.init()

#nastaveni kamery
import picamera
cam = picamera.PiCamera()

#nastaveni jasu
if denniSvetlo:
    cam.brightness = 30
else:
    cam.brightness = 60

if svetlyObjekt:
    cam.brightness = 25
    cam.contrast=30

cam.start_preview()
sleep(3)
cam.stop_preview()

#definovani vstupnich argumentu
nazev=str(sys.argv[1])
datum=nazev+datetime.now().strftime("%Y_%m_%d_%H_%M")
pocetOtoceni = int(sys.argv[2])+1

souborSouradnic = cestaRamdisk+datum+'.xyz'

#definovani pracovnich matic a vectoru
maticeDat = [[ 0 for i in range(pocetOtoceni+1)] for j in range(radek+1) ] #main data matrix
maticeFiltru = [[ 0 for i in range(pocetOtoceni+1)] for j in range(radek+1) ] #filter data matrix
pomocnyVector = [ 0 for j in range(radek+1) ]
#auxiliary vector

#oziveni laseru a motoru
nastavLaser()
nastavMotor()

#procedura provadejici skenovani
def provedSken(uhelKroku):
```

```
global maticeDat, celkovyPocetBodu, souborTXT
nazevSouboru = "temp"+datum+".jpg"
print nazevSouboru
cestaSouboru = cestaRamdisk+nazevSouboru

#sejmuti obrazku
cam.capture(cestaSouboru)
#Inicializace obrazovky pro zobrazeni
obrazov-
ka=pygame.display.set_mode([sirkaObrazovky,vyskaObrazovky])
#nacteni obrazku z ulozeneho souboru
obrn = pygame.image.load(cestaSouboru)
#otoceni obrazku o 180 stupnu
obr = pygame.transform.rotate(obrn, 180)
#ziskani obdelnikovou plochu povrchu
obrRect = obr.get_rect()
#prekresleni jednoho obrazku na druhy
obrazovka.blit(obr, obrRect)
#aktualizace plneho zobrazeni na obrazovku
pygame.display.flip()

#vykresleni skenovaci plochy
#vykresleni usecky
#line(povrch, barva, zacatek(x,y), konec(x,y), tloustka)

#horni vodorovna cara
pygame.draw.line(obrazovka,cBlu,(10,horniMez),(sirka-
10,horniMez),2)
#spodni vodorovna cara
pygame.draw.line(obrazovka,cBlu,(10,vyska-dolniMez),(sirka-
10,vyska-dolniMez),2)
#prava kolma cara
pygame.draw.line(obrazovka,cBlu,(sirka-
pravaMez,horniMez),(sirka-pravaMez,vyska-dolniMez),2)
#leva kolma cara
pyga-
me.draw.line(obrazovka,cBlu,(levaMez,horniMez),(levaMez,vyska-
dolniMez),2)
#osova cara
pygame.draw.line(obrazovka,cWhi,(osa,horniMez),(osa,vyska-
10),2)
#nastavit hodnotu barvy pro jeden pixel
#set_at((x, y), barva)
obrazovka.set_at((10,10),cRed)
```

```
obrazovka.set_at((11,11),cRed)

#aktualizace plneho zobrazeni na obrazovku
pygame.display.flip()

x=pravaMez
y=horniMez

#hledani v rozsahu (dolniMez a horniMez) x (pravaMez a levaMez)
pro kazdy uhel
while y<vyska-dolniMez:
    #ziskani RGB slozek z jednoho bodu z obrazku
    #cervena slozka
    cR = obrazovka.get_at((sirka-x,y))[0]
    #zelena slozka
    cG = obrazovka.get_at((sirka-x,y))[1]
    #modra slozka
    cB = obrazovka.get_at((sirka-x,y))[2]

    if cR>fR and cG<fG and cB<fB:
        cR1 = obrazovka.get_at((sirka-x-1,y))[0]
        cR2 = obrazovka.get_at((sirka-x-2,y))[0]
        if cR1>cR:
            x=x-1
            if cR2>cR1:
                x=x-1

        vzdalenostOdOsy=sirka-x-osa
        # matice(y-horniMez,uhelKroku,vzdalenostOdOsy)
        maticeDat[y-horniMez][uhelKroku]=vzdalenostOdOsy
        celkovyPocetBodu = celkovyPocetBodu+1

        y=y+kroky
        x=pravaMez

x=x+1
if x>sirka-levaMez:
    x=pravaMez
    y=y+kroky
    obrazovka.set_at((sirka-x,y),cBlu)
```

```
y=horniMez+2

#filter0
while y<vyska-dolniMez-2:
    d = maticeDat[y-horniMez][uhelKroku]
    x=sirka-osa-d
    obrazovka.set_at((sirka-x,y),cRed)
    y=y+kroky

#filter1
if filter1:
    y=horniMez+2
    while y<vyska-dolniMez-2:
        d = plovouciPrumer(maticeDat[y-horniMez-
1][uhelKroku],maticeDat[y-horniMez][uhelKroku],maticeDat[y-
horniMez+1][uhelKroku])
        pomocnyVector[y-horniMez] = d
        y=y+kroky

    y=horniMez+2
    while y<vyska-dolniMez-2:
        maticeDat[y-horniMez][uhelKroku] = pomocnyVector[y-
horniMez]
        y=y+kroky

#filter2
if filter2:
    y=horniMez+3
    while y<vyska-dolniMez-3:
        if (maticeDat[y-horniMez-2][uhelKroku] and maticeDat[y-
horniMez-1][uhelKroku] and maticeDat[y-horniMez][uhelKroku] and
maticeDat[y-horniMez+1][uhelKroku] and maticeDat[y-
horniMez+2][uhelKroku]):
            d = (maticeDat[y-horniMez-2][uhelKroku]+maticeDat[y-
horniMez-1][uhelKroku]+maticeDat[y-
horniMez][uhelKroku]+maticeDat[y-
horniMez+1][uhelKroku]+maticeDat[y-horniMez+2][uhelKroku])/5
            pomocnyVector[y-horniMez] = d
            x=sirka-osa-d
            obrazovka.set_at((sirka-x,y),cGre) #
        else:
            pomocnyVector[y-horniMez] = maticeDat[y-
horniMez][uhelKroku]
```

```
        y=y+kroky

    y=horniMez+2
    while y<vyska-dolniMez-2:
        maticeDat[y-horniMez][uhelKroku] = pomocnyVector[y-
horniMez]
        y=y+kroky

#aktualizace plneho zobrazeni na obrazovku
pygame.display.flip()

#procedura ukladajici souradnice x,y,z do souboru
def ulozSouradnice(uhelKroku, souborXYZ):
    global maticeDat, celkovyPocetBodu
    y=horniMez
    y=horniMez+1
    while y<vyska-dolniMez:
        vzdalenostOdOsy = maticeDat[y-horniMez][uhelKroku]
        if vzdalenostOdOsy!=0 and vzdalenostOdOsy>-200:
            uhel=float(2*pi/(pocetOtoceni-1)*uhelKroku)
            rx=float(math.sin(uhel)*vzdalenostOdOsy*koef)
            ry=float(math.cos(uhel)*vzdalenostOdOsy*koef)
            rz = y
            co = str(rx)+" "+str(ry)+" "+str(rz)
            souborXYZ.write(co+"\n")
            y=y+kroky

#filter3
def filter3():
    global maticeDat, maticeFiltru, celkovyPocetBodu, souborXYZ
    for uhelKroku in range (2,pocetOtoceni-2):
        y=horniMez+1
        while y<vyska-dolniMez:
            af1 = plovouciPrumer(maticeDat[y-
horniMez+1][uhelKroku],maticeDat[y-horniMez+1][uhelKroku-
1],maticeDat[y-horniMez+1][uhelKroku+1])
            af2 = plovouciPrumer(maticeDat[y-
horniMez][uhelKroku],maticeDat[y-horniMez][uhelKroku-
1],maticeDat[y-horniMez][uhelKroku+1])
            af3 = plovouciPrumer(maticeDat[y-horniMez-
1][uhelKroku],maticeDat[y-horniMez-1][uhelKroku-1],maticeDat[y-
horniMez-1][uhelKroku+1])
            af4 = plovouciPrumer(af1,af2,af3)
```

```
maticeFiltru[y-horniMez][uhelKroku]=af4
y=y+1

for uhelKroku in range (2,pocetOtoceni-2):
    y=horniMez+1
    while y<vyska-dolniMez:
        maticeDat[y-horniMez][uhelKroku]=maticeFiltru[y-
horniMez][uhelKroku]
        y=y+1

#funkce plovouci prumer - matematicky prumer okolnich pixelu
def plovouciPrumer(a1,a2,a3):
    prumer=a2
    if (a1>0 and a2>0 and a3>0):
        prumer = (a1+a2+a3)/3
    else:
        if (a1==0 and a2 and a3):
            prumer = (a2+a3)/2
        if (a1 and a2==0 and a3):
            prumer = (a1+a3)/2
        if (a1 and a2 and a3==0):
            prumer = (a1+a2)/2
    return prumer

#hlavni cast skenovani
for i in range (pocetOtoceni-1):
    laserJednaZap()
    provedSken(i)
    laserJednaVyp()
    print "-----"
    hodnoty=nazev+" "+str(i+1)+"/"+str(pocetOtoceni-1)+"
("+str(celkovyPocetBodu)+")"
    print hodnoty
    laserDvaZap()
    provedSken(i)
    laserDvaVyp()
    posunMotor(1600/(pocetOtoceni-1),100)

#dohlazovaci filtry
if filter3:
    filter3()
```

```
if filter4:
    filter3()
    filter3()

#procedura zajistujici uzavreni a ulozeni souboru se souradnice-
mi
def ulozData():
    #ulozeni_dat
    souborXYZ = open(souborSouradnic,"a")
    print "-----"
    print "ulozeno"
    for i in range (pocetOtoceni-1):
        ulozSouradnice(i, souborXYZ)
    souborXYZ.close()

#procedura generujici obraz naskenovanych souradnic ve 2D
def vygenerujObrazekSouradnic():

    posun=5
    priblizeni=2

    naskenovanaData = cestaRamdisk+datum +'.txt' #RAW scann data
    obrazekSouradnic = cestaRamdisk+datum +'.jpg'
    souborTXT = open(naskenovanaData,"a")
    obrazov-
ka=pygame.display.set_mode([sirkaObrazovky,vyskaObrazovky])
    for u in range (pocetOtoceni-1): #uhel
        for r in range (radek): #row
            x=maticeDat[r][u]
            souborTXT.write(str(x)+",")
            obrazov-
ka.set_at((u*posun+x*priblizeni,r*priblizeni),cGre)
            #aktualizace plneho zobrazeni na obrazovku
            pygame.display.flip()
            souborTXT.write("\n")
    souborTXT.close()

    #aktualizace plneho zobrazeni na obrazovku
    pygame.display.flip()
    #ulozeni obrazku na disk
    pygame.image.save(obrazovka,obrazekSouradnic)
```

```
#volani hlavnich procedur
ulozData()
vygenerujObrazekSouradnic()

#vypinani laseru a motoru
vsechnyLaseryVyp()
vypniMotor()
time.sleep(6)
```