

**ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA ELEKTROTECHNICKÁ**

Katedra aplikované elektroniky a telekomunikací

DIPLOMOVÁ PRÁCE

3D scanner založený na multikamerovém systému

ZÁPADOČESKÁ UNIVERZITA V PLZNI
Fakulta elektrotechnická
Akademický rok: 2016/2017

ZADÁNÍ DIPLOMOVÉ PRÁCE
(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Martin JUŘÍK**
Osobní číslo: **E15N0066P**
Studijní program: **N2612 Elektrotechnika a informatika**
Studijní obor: **Dopravní elektroinženýrství a autoelektronika**
Název tématu: **3D scanner založený na multikamerovém systému**
Zadávající katedra: **Katedra aplikované elektroniky a telekomunikací**

Z á s a d y p r o v y p r a c o v á n í :

1. Prostudujte problematiku snímání 3D scén kamerou.
2. Navrhněte a vytvořte nosnou konstrukci s potřebnými rozvody napájení a komunikace.
3. Navrhněte a implementujte software pro synchronizaci příslušných kamer pro snímání fotografií.
4. Zajistěte ukládání snímků a následné odeslání do centrálního úložiště.
5. Vyberte a otestujte dostupný software pro vytváření 3D scén ze získaných snímků.

Rozsah grafických prací: podle doporučení vedoucího

Rozsah kvalifikační práce: 40 - 60 stran

Forma zpracování diplomové práce: tištěná/elektronická

Seznam odborné literatury:

1. The pi 3D Scanner[online](c)2014[cit.4.4.2014]Dostupné z:
<http://www.pi3dscan.com>
2. Raspberry pi - multi camera setup[online](c)2014[cit.4.4.2014]Dostupné z:
http://feedback.autodesk.com/cloudservices/topics/raspberry_pi_multi_camera_se
3. Raspberry Pi as low-cost HD surveillance camera[online](c)2014[cit.4.4.2014]Dostupné z: <http://www.codeproject.com/Articles/665518/Raspberry-Pi-as-low-cost-HD-surveillance-camera>
4. Autodesk ReCap [online](c)2014[cit.4.4.2014]Dostupné z:
<http://www.autodesk.com/products/recap/overview>
5. Upton,E.; Halfacree,G.; Raspberry,Pi.: Uživatelská příručka, Brno: Computer Press, 2013, ISBN 978-80-2514-116-8

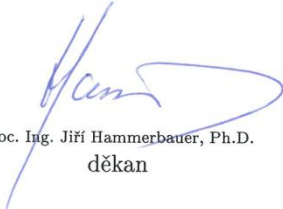
Vedoucí diplomové práce:

Ing. Petr Kropík, Ph.D.

Katedra teoretické elektrotechniky

Datum zadání diplomové práce: 14. října 2016

Termín odevzdání diplomové práce: 19. května 2017


Doc. Ing. Jiří Hammerbauer, Ph.D.
děkan




Doc. Dr. Ing. Vjačeslav Georgiev
vedoucí katedry

V Plzni dne 14. října 2016

Abstrakt

Předkládaná diplomová práce je zaměřena na konverzi fotografií v trojrozměrný model objektu. První část se zabývá teorií, jak lze této konverze docílit. Na tuto část navazuje popis praktické realizace. Pro tyto účely bylo vytvořeno zařízení, které snímky pořídí v totožný okamžik. Zároveň byly vytvořeny programy, které zařízení ovládají a které tvoří rozhraní pro snadnou komunikaci uživatele se scannerem.

Klíčová slova

Rekonstrukce modelu objektu, 3D scanner, stereo vize, klíčové body, Java, RaspberryPi, kamera

Abstract

This master thesis is focused on reconstruction of 3D model from several photos. First part describes theory which was used in building final device. This work connects the worlds of hardware and software. Both parts were covered in creating this thesis.

Key words

Model reconstruction, 3D scanner, stereo vision, points of interest, Java, RaspberryPi, camera

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této práce.

Dále prohlašuji, že veškerý software, použitý při řešení této diplomové práce, je legální.

.....
podpis

V Plzni dne 17.5.2017

Bc. Martin Juřík

Poděkování

Tímto bych rád poděkoval vedoucímu diplomové práce Ing. Petru Kropíkovi, Ph.D. za cenné profesionální rady, připomínky a metodické vedení práce.

Dále bych rád poděkoval lidem, kteří mi s mými projekty významně pomohli. Jsou jimi Vojtěch Kadlec, Václav Král a Pavel Jindra. Patří jim mé poděkování především za předané zkušenosti, rady a věcné připomínky při řešení projektů.

Obsah

OBSAH	8
ÚVOD	11
1 TEORETICKÝ ROZBOR	12
1.1 OBECNÝ POPIS PROBLEMATIKY REKONSTRUKCE TROJROZMĚRNÉHO OBJEKTU	12
1.2 SNÍMÁNÍ SCÉNY	13
1.2.1 Kamery.....	13
1.2.2 Synchronizace vyfocení.....	14
1.3 DETEKCE KLÍČOVÝCH BODŮ V OBRAZE	17
1.3.1 Houghova transformace	17
1.3.2 SIFT	18
1.3.3 SURF.....	19
1.4 PRINCIP REKONSTRUKCE OBJEKTU Z OBRAZU	22
1.5 PROGRAMY PRO REKONSTRUKCI TROJROZMĚRNÝCH MODELŮ.....	24
2 PRAKTICKÁ REALIZACE	26
2.1 MECHANICKÁ KONSTRUKCE	26
2.1.1 Kamery.....	26
2.1.2 Prototyp	30
2.1.3 Finální výrobek.....	32
2.2 PROGRAM PRO REKONSTRUKCI OBJEKTU	35
2.2.1 Experimentální implementace.....	35
2.2.2 Výsledné řešení	36
2.3 PROGRAM PRO OVLÁDÁNÍ SCANNERU.....	37
2.3.1 Centrální úložiště snímků.....	37
2.3.2 Řídicí program scanneru	37
2.3.3 Uživatelské rozhraní	39
2.4 POROVNÁNÍ VLASTNÍHO ŘEŠENÍ S KOMERČNÍMI PRODUKTY	40
2.4.1 Vlastní řešení	40
2.4.2 Komerční produkty	42
2.5 NÁVOD NA INSTALACI	43
2.6 NÁVOD NA POUŽÍVÁNÍ.....	46
3 DALŠÍ MOŽNÁ VYLEPŠENÍ	49
3.1 ROZŠÍŘENÍ MECHANICKÉ KONSTRUKCE	49
3.2 REKONSTRUKCE OBJEKTU SCANNEREM	49
3.3 REKONSTRUKCE OBJEKTU EXTERNĚ	49
3.4 REKONSTRUKCE VIDEO ZÁZNAMU VE 3D	50
3.5 NAPOJENÍ NA 3D TISKÁRNU.....	51
4 ZÁVĚR	52
PŘÍLOHY	I

Seznam obrázků

OBR. 1: NASTÍNĚNÍ KONCEPTU SNÍMÁNÍ PROSTORU VÍCE KAMERAMI. V PRAVÉ ČÁSTI JSOU SNÍMKY Z KAMER.	12
OBR. 2: DETEKTOR SIFT ALGORITMU. PŘEVZATO Z [16].....	18
OBR. 3: SNÍMEK S VYZNAČENÝMI DESKRIPTORY ZÍSKANÉ METODOU SIFT. PŘEVZATO Z [14]...19	19
OBR. 4: LEVÉ DVA SNÍMKY ZNÁZORNĚJÍ PARCIÁLNÍ DERIVACI DRUHÉHO ŘÁDU GAUSSOVY KŘIVKY. V PRAVO JE JEJICH APROXIMACE POUŽITÁ V METODĚ SURF. OBRÁZEK PŘEVZAT Z [15].....	20
OBR. 5: SNÍMEK S VYZNAČENÝMI DESKRIPTORY ZÍSKANÉ METODOU SURF.....	21
OBR. 6: RŮZNÉ POHLEDY NA STEJNÝ OBJEKT.....	22
OBR. 7: STEREO VIZE (ZDROJ [21])	23
OBR. 8: PROGRAM VISUALSFM PRO REKONSTRUKCI MODELU OBJEKTU	24
OBR. 9: PROGRAM RECAP360 PRO REKONSTRUKCI MODELU OBJEKTU.....	25
OBR. 10: PŮVODNÍ UCHYCENÍ KAMEROVÝCH MODULŮ V OFICIÁLNĚ DODÁVANÉ KRABÍČCE.	26
OBR. 11: POHLED SHORA (VLEVO) A ZBOKU (V PRAVO) NA KAMERU RASPBERRYPI CAMERA VERZE 1.3 S MODŘE NAZNAČENÝM UCHYCENÍM ČOČKY (1) A ČERVENĚ NAZNAČENOU ZÁKLADNOU (2)	27
OBR. 12: ÚPRAVA OSTŘÍČÍHO BODU POUŽITÝCH KAMER. VLEVO JE VÝŘEZ FOTOGRAFIE PŘED KOREKČÍ, V PRAVO PO KOREKCI.	28
OBR. 13: ZNÁZORNĚNÍ POČTU ROZMAZANÝCH PIXELŮ NA VZDÁLENOSTI OBJEKTU OD KAMERY PRO ZAOSTŘENÝ BOD 50CM OD KAMEROVÉ ČOČKY.....	29
OBR. 14: ZNÁZORNĚNÍ POČTU ROZMAZANÝCH PIXELŮ NA ÚHLU VYTOČENÍ ČOČKY KAMERY RASPBERRYPI CAMERA VERZE 1.3 PRO POŽADOVANOU OSTŘÍČÍ VZDÁLENOST 50CM	30
OBR. 15: PROTOTYP SCANNERU	31
OBR. 16: ZPŮSOB UCHYCENÍ KAMER VE FINÁLNÍM VÝROBKU	32
OBR. 17: FINÁLNÍ VÝROBEK	33
OBR. 18: PRINCIP TŘÍKLOBOVÉHO POLOHOVACÍHO RAMENA KAMER	34
OBR. 19: OSAZENÍ KAMER NA MECHANICKÉ KONSTRUKCI. ČERVENĚ JSOU ZNÁZORNĚNY KAMERY PŘIPEVNĚNÉ K PODKLADOVÉ DESCE A ZELENĚ POZICE KAMER NA POLOHOVACÍCH RAMENECH (VČETNĚ NAZNAČENÉ OSY POHYBU).....	34
OBR. 20: VÝSLEDEK VLASTNÍ IMPLEMENTACE REKONSTRUKČNÍCH ALGORITMŮ.....	36
OBR. 21: TEST SYNCHRONIZACE JEDNOTLIVÝCH PRVKŮ.....	39
OBR. 22: PROJEKT PI3DSCAN (ZDROJ: HTTP://WWW.PI3DSCAN.COM/IMAGES/3DSCANNER/RESIZEDIMAGES/IMG_0066.JPG)	40
OBR. 23: VZHLED GRAFICKÉHO ROZHRANÍ APLIKACE STRATO	47
OBR. 24: NASTAVENÍ APLIKACE STRATO	47
OBR. 25: NOVÝ PROJEKT V APLIKACE STRATO	47
OBR. 26: OBRAZOVKA NASTAVENÍ SCANNERU APLIKACE STRATO	48
OBR. 27: RPI CLUSTER.....	50
OBR. 28: ARTEC SHAPIFY 3D SCANNER (ZDROJ: HTTP://WWW.ANIWAA.COM/WP-CONTENT/UPLOADS/2015/07/3D-SCANNER-ARTEC-SHAPIFY-BOOTH-MAN.JPG)	I
OBR. 29: DOOBLICATOR 3D SCANNER (ZDROJ: HTTP://WWW.DOOBGROUP.COM/FILEADMIN/USER_UPLOAD/BILDER_CONTENT/DOOBLICATOR_DOME_WEB.JPG)	II
OBR. 30: TWINSTANT MOBILE 3D SCANNER (ZDROJ: HTTP://WWW.ANIWAA.COM/WP-CONTENT/UPLOADS/2016/05/3D-SCANNER-TWINDOM-TWINSTANT-MOBILE.JPG)	II
OBR. 31: ARTEC EVA (VLEVO) A ARTEC SPACE SPIDER (V PRAVO) (ZDROJ: HTTPS://ARTEC3D-	

PRODUCTION.S3-EU-WEST-1.AMAZONAWS.COM/PRODUCTS/ARTEC-EVA.PNG	A
HTTPS://WWW.3DZ.IT//WORDPRESS/WP-CONTENT/UPLOADS/2016/01/MAIN-STAMPANTI-3D-SCANNER3D-3DZ-ARTEC-SPACE-SPIDER.PNG)	III
OBR. 32: HANDYSCAN 700 (VLEVO) A EinSCAN-Pro (UPROSTŘED) A SENSE 3D SCANNER (ZDROJ: HTTP://WWW.ANIWAA.COM/WP-CONTENT/UPLOADS/2014/05/3D-SCANNER-CREAFORM-HANDYSCAN-700-PERSPECTIVE.JPG , HTTP://WWW.ANIWAA.COM/WP-CONTENT/UPLOADS/2016/12/3D-SCANNER-SHINING-3D-EINSCAN-PRO-PLUS-PERSPECTIVE.JPG A HTTPS://STATIC.WEBSHOPAPP.COM/SHOPS/024440/FILES/065316598/3D-SYSTEMS-CUBIFY-SENSE-RS-3D-SCANNER.JPG)	III
OBR. 33: ROZLOŽENÉ VÝSLEDNÉ ZAŘÍZENÍ. V PRAVÉ ČÁSTI JE PODKLADOVÁ KONSTRUKCE A V ČÁSTI LEVÉ KRYCÍ DÍL S KAMERAMI.	IV
OBR. 34: UKÁZKA JEDNOHO Z NASKENOVANÝCH OBJEKTŮ POMOCÍ VYTVOŘENÉHO ZAŘÍZENÍ V PROGRAMU MESH LAB	IV

Úvod

Hlavním cílem práce, jak ostatně vyplývá z názvu, je vytvořit zařízení, které bude schopné převést tvar reálného objektu do trojrozměrného počítačového modelu. S tímto modelem je dále možné provádět simulace a různé tvarové úpravy. Trojrozměrné modely objektů jsou v dnešní době trend a bude stále přibývat aplikací, kde budou zapotřebí. Ať už se jedná o virtuální (a rozšířenou) realitu nebo trojrozměrné skenování a rozpoznávání okolí autonomními vozidly v reálném čase.

V první kapitole se práce zabývá teoretickým úvodem do problematiky rekonstrukce trojrozměrných modelů, kde jsou nastíněny matematické postupy, které se pro účely rekonstrukce používají. Na tuto část navazuje kapitola druhá, ve které je nastíněno implementované programové řešení a mechanická konstrukce vyrobeného scanneru. V kapitole třetí se práce zabývá možnými dalšími vylepšeními vytvořeného zařízení.

Práce spojuje velké množství odvětví. Od elektroniky přes programování až po komplexnější matematiku. Pro úspěšné pochopení práce je nutné mít v těchto odvětvích základní znalosti nebo si v případě nejasností dohledat o použitém termínu více informací.

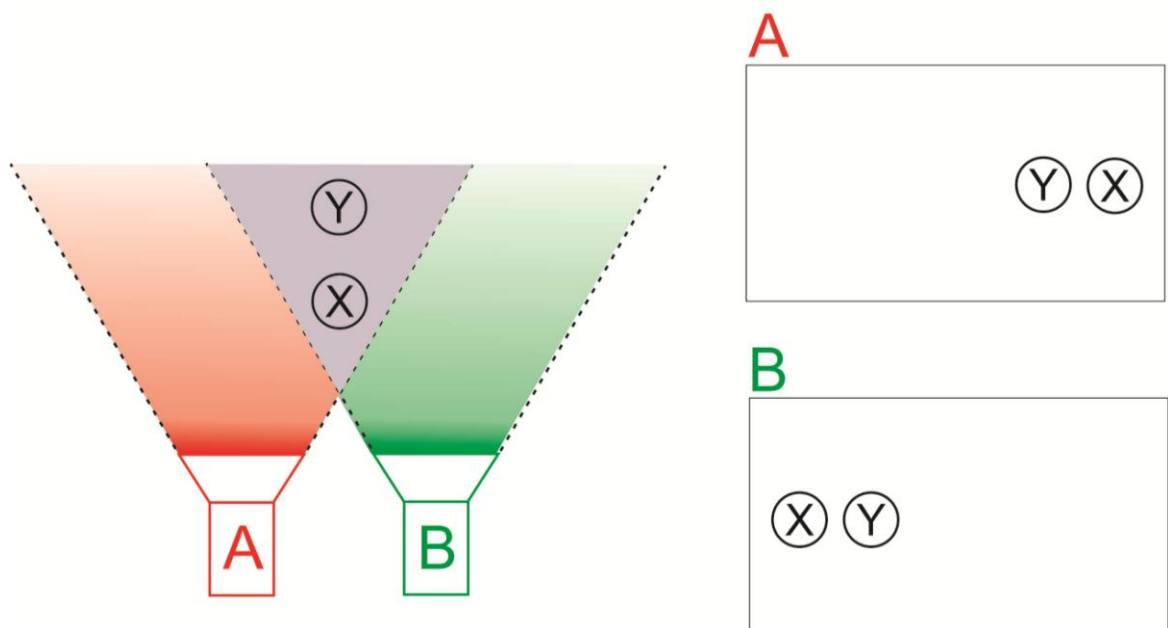
Celý projekt byl pojmenován Strato.

1 Teoretický rozbor

Cílem teoretického rozboru je nastínit čtenáři matematickou podstatu, která se ukrývá za několika tisíci řádky programového kódu. V první části je nastíněn celkový koncept a různé principy snímání objektu. Dále už je kapitola zaměřena více na způsob rekonstrukce, který byl využit v této práci. Kapitulu uzavírá soupis dostupných řešení vyvíjených univerzitami a možná komerční řešení.

1.1 Obecný popis problematiky rekonstrukce trojrozměrného objektu

Princip snímání objektu, pro účely rekonstrukce jeho trojrozměrného modelu, je možné realizovat dvěma základními způsoby. Tím prvním je snímání laserem, kde se měří vzdálenost snímacího čipu od bodu umístěného na objektu. Pro tento druh měření se využívá principu interferometrie nebo triangulace. Laserové snímače bývají často doplněny o kamerový snímač, který dodá modelu texturu. Druhý princip rekonstrukce modelu objektu je založen na vytvoření snímků z několika úhlů. Z těchto snímků je následně vypočítána pozice jednotlivých bodů ve vztahu k pozici kamer. Právě na tomto principu je založena celá tato práce.



Obr. 1: Nastínění konceptu snímání prostoru více kamerami. V pravé části jsou snímky z kamer.

Tato podkapitola si dává za cíl vytvořit elementární přehled o základní terminologii a zjednodušeném principu rekonstrukce. Detailněji se touto problematikou zabývají všechny zbylé podkapitoly této části práce. Prvním krokem je rozpoznání jednoho bodu na více snímcích. Tento bod je odborně označován jako bod klíčový. Po úspěšném rozpoznání tohoto bodu a následném spárování stejného bodu na více snímcích je možné začít počítat jeho vzdálenost od pomyslné spojnice dvou kamer. Základní princip rozpoznání, který bod je blíže a který dál od této spojnice je vyobrazen na *Obr. 1*. Kamera A vidí objekty X a Y v jiném pořadí než kamera B. Algoritmus je tedy schopen zjistit, že tyto body nebudou ve stejné vzdálenosti od kamer. Jedná se pouze o modelovou situaci, která má nastínit základní princip. Algoritmy řešící tuto problematiku jsou mnohem složitější zahrnující velké množství matematiky a práce s maticemi. Konkrétní způsoby výpočtu jsou vysvětleny v podkapitole 1.4.

Dnešní pokročilé mobilní technologie umožňují tvořit trojrozměrné modely pomocí mobilních aplikací. Po otevření aplikace stačí s telefonem obcházet skenovaný objekt a aplikace si v průběhu tvoří snímky. Následně se tyto snímky odešlou na server vývojářů aplikace, kde se provede výpočetně náročná rekonstrukce. Výhodou tohoto řešení je, že pokud uživateli nejde o vysokou přesnost, tak nemusí vlastnit drahý scanner.

1.2 Snímání scény

Ze zadání vyplývá, že pro snímání objektu byly v tomto projektu využity kamery. Pomocí těchto kamer je zachycen vzhled objektu z několika stran v určitém časovém rozpětí. Pro úspěšnou rekonstrukci je vhodné, aby bylo toto rozpětí co nejmenší.

1.2.1 Kamery

Kamery jsou bezesporu nejdůležitějším prvkem mechanické konstrukce. Od parametrů kamer se odvíjí kvalita zrekonstruovaného modelu objektu. Na kvalitu zrekonstruovaného modelu objektu má největší vliv kvalita pořízených snímků, osvětlení scény, pozice a množství kamer.

Kvalita pořízených snímků je vyšší u kamer, které mají schopnost automatického optického zaostření na objekt. S touto funkcí roste ovšem i cena kamer. Levnější kamery tuto funkci nemají a je tedy vhodné jim nastavit ostřicí bod manuálně. V opačném případě jsou

zaostřeny na bod jiný a výsledné snímky mají zbytečně nižší kvalitu, než by mohly mít. Druhým významným bodem, ovlivňujícím kvalitu, je rozlišení kamer. Čím vyšší rozlišení, tím lépe je zachycena informace o přesné pozici hran a dalších důležitých detailů objektu. Neznamená to ovšem, že úměrně s vyšším rozlišením poroste kvalita výsledku, protože fotoaparáty dnešních mobilních telefonů mají ohromné rozlišení, avšak kamerový čip není zdaleka tak kvalitní jako v profesionálních fotoaparátech se stejným rozlišením. Výsledku ovšem tento parametr pomůže. Vliv na kvalitu mají i další parametry kamery jako jsou světelnost, kvalita objektivu atp.

Osvětlení scény není jednoduchou problematikou, neboť by toto osvětlení nemělo oslnit ostatní kamery a zkazit tak jimi zachycené snímky. Pokud je požadována rekonstrukce modelu objektu včetně textury, tak by toto osvětlení nemělo barvu objektu příliš změnit. Pokud je požadována přesná textura, tak je důležitým bodem u osvětlení také rovnoměrnost. Objekt by tedy neměl být nasvětlen pouze z jedné strany, aby nedocházelo ke zkreslení textury stíny.

Umístění kamer a jejich počet má bezpochyby nezanedbatelný vliv na výsledný model. Obecně platí, že čím více fotografií objektu, tím lepší výsledek rekonstrukce. S množstvím kamer roste ovšem i cena celého zařízení. Pro úspěšnou rekonstrukci je doporučován (viz [8]) největší vzájemný úhel mezi kamerami 30° . Z toho je možné odvodit nejmenší počet kamer pro snímání ze všech stran, a to 12. Zároveň je ovšem profesionálními programy na rekonstrukci vyžadován nejmenší počet snímků 20. V kapitole 2.1.2 jsou popsány získané zkušenosti s touto problematikou, díky kterým byl úhel 30° shledán jako příliš veliký. Vliv na výslednou kvalitu rekonstruovaného objektu nemá vliv pouze množství kamer, ale i jejich umístění. Empiricky bylo zjištěno, že je vhodné kamery umístit do více rovin. Pro názornost je možné vidět umístění v jedné rovině na *Obr. 15* a umístění ve více rovinách na *Obr. 17*.

1.2.2 Synchronizace vyfocení

Problematika synchronizace zachycení snímků v totožný okamžik všemi kamerami nebyla příliš důležitou pro vytvářený typ scanneru, protože se jedná o scanner rekonstruující statické objekty. V případě použití scanneru pro zachycení pohyblivého se objektu (například elektrický točivý stroj v provozu) je nutné se touto problematikou zabývat mnohem důkladněji. Tato problematika byla ovšem shledána jako zajímavý technický úkol na řešení a díky tomu jí byl věnován značný čas. Informaci o způsobu synchronizace prvků není možné u

komerčních produktů dohledat, a tudíž jsou následující odstavce pouhými předpoklady, jaké technologie by pro tento účel bylo možné použít. Některé z nich byly empiricky otestovány (viz 2.3.2).

Jako první možností synchronizace se nabízelo využití komunikační sítě Ethernet, do které jsou všechny jednotky připojeny kvůli odeslání vyfocené scény do centrálního úložiště (viz 2.3.1). Realizace je možná odesláním broadcast paketu, který je odeslán do všech zařízení ve stejném segmentu sítě. Zpoždění na této sběrnici není ovšem přesně definováno a zařízením není ani nijak detekováno. Proti tomu ovšem vyvstává fakt, že síť Ethernet s použitím fyzické vrstvy *Gigabit Ethernet* (definováno normou IEEE 802.3ae) je schopná přenést paket o velikosti 1kB přibližně za jednu mikrosekundu¹. K tomu se přičte doba odesílání paketu z vysílače, doba šíření na vedení a doba přijímání paketu přijímačem. Tyto parametry se ovšem pohybují řádově v desítkách nanosekund a stávají se tím zanedbatelné ve výsledné sumě času. Zpoždění tedy oficiální specifikací není přesně definováno, avšak pohybuje se řádově na jednotkách mikrosekund. Větším zdržením ve výsledku bude reakce obslužného programu na tuto událost. Velikou výhodou této implementace je standardní způsob získávání dat ze sítě Ethernet napříč všemi operačními systémy.

Další uvažovanou možností je synchronizace pomocí vnějšího přerušení jádra mikroprocesoru použitého mini počítače. Realizace je možná například pomocí vysílací LED na řídicím členu scanneru a přijímacích fotodiod na prvcích realizujících focení. Celé to může být realizováno v neviditelném infračerveném pásmu, aby nebyl uživatel rušen blikáním scanneru. Výhodu to má v doručení informace všem prvkům v naprosto shodný časový okamžik, protože se tato informace šíří prostředím rychlostí světla (přibližně 300 000 km/s). Nevýhod má tento systém ovšem více, než systém předchozí. První z nich je, že řídicí prvek musí být v přímé viditelnosti se všemi prvky ostatními. To je nevýhoda u scannerů větší velikosti a složitější konstrukce. Druhou velkou nevýhodou je, že doba reakce na externí přerušení mikroprocesoru není shodná ve všech zařízeních a ve všech operačních systémech. Dalo by se to vyřešit použitím RTOS² a udělit události vnějšího přerušení nejvyšší prioritu. To by do projektu ovšem vneslo velké množství komplikací z hlediska instalace, nastavení a údržby tohoto systému.

¹ Tyto časy byly spočítány pomocí vzorců na následující adrese https://en.wikipedia.org/wiki/Transmission_time

² Real Time Operating System – operační systém reálného času

U většiny kamer by mělo být umístěno i LED osvětlení objektu, které by mohlo být ovládáno centrálně z řídicího prvku. Informace o momentu vyfocení by se tedy dala přenést i po napájecích vodičích tohoto osvětlení. Jednodušším způsobem přenosu informace je udělat jednoduchou reakci na náběžnou hranu napájecího napětí. Sofistikovanější metodou je po těchto vodičích dokonce přenést několikabitovou informaci. Tato informace by šla přenést modulací některé ze sériových sběrnic (například UART) na napájecí napětí nebo měřením střidy PWM použitého osvětlení. Nevýhodou má tento systém již ve dříve zmíněné zhoršené reakci na externí přerušení.

Synchronizace pomocí systémového času je jeden ze standardních způsobů synchronizace prvků v počítačových sítích. Hlavním prvkem tohoto systému je server, který vysílá do sítě zprávy protokolem NTP (Network Time Protocol). Prvky v síti si v reakci na tuto zprávu upraví systémový čas. Díky přesnému času je následně možné poslat prvkům v síti příkaz, že v konkrétní čas má proběhnout vyfocení. Ve zdroji [1] je v sekci 5.1.3.1 rozebíráno zpoždění této synchronizace, které se obvykle pohybuje v rozmezí 5 až 100 milisekund a může se dokonce stát, že bude i vyšší. Na toto zpoždění má vliv několik faktorů a tím nejvýraznějším je neurčitý počet prvků mezi zařízením a NTP serverem, kde každý z prvků vnáší do této informace zpoždění. Neprospívá tomu ani nedefinované zpoždění v síti Ethernet.

Jednou z dalších možností synchronizace je použití systému RBS (Reference Broadcast Synchronization [2]). Jedná se o vylepšený systém NTP, kde ovšem prvky v síti nekomunikují pouze se serverem, ale také mezi sebou. Prvky tedy nepotřebují vědět přesný čas, ale stačí jim znát, jak moc se jejich čas liší od prvků, se kterými mají být v synchronním spojení.

Poslední z možných alternativ je implementace protokolu PTP (Precision Time Protocol), který je definován normou IEEE 1588-2008. Tento protokol realizuje synchronizaci systémových hodin s mikrosekundovou přesností. Byl navržen tak, aby doplnil protokol NTP a zvýšil tím přesnost synchronizace prvků v lokální síti.

1.3 Detekce klíčových bodů v obraze

Základem rekonstrukce obrazu je detekce klíčových bodů v obraze (v anglickém originále „point of interest“). Tyto body jsou něčím typické a snadno rozpoznatelné i na jiných snímcích. Na základě pozice těchto bodů je možné dopočítat vzájemný vztah dvou obrázků. Počátky hledání těchto bodů ovšem nebyly matematicky příliš sofistikované a jednalo se o hledání přímek, kružnic a jiných základních tvarů pomocí metody hrubé síly (*Houghova transformace*). V dnešní době jsou již algoritmy sofistikovanější (*SIFT* a *SURF*) a pohlížejí na obraz jako na matematickou funkci, ve které například vyhledávají body se shodnými gradienty ve všech směrech. Jednou z velmi mála méně sofistikovaných metod, která je stále hojně využívána, je detekce hran a rohů. Nejčastějším způsobem implementace je Harrisův hranový detektor nebo složitější Cannyho hranový detektor.

Pro účel hledání shodných bodů v obrazech by šla využít i neuronová síť. Této síti bychom se následně dotazovali na shodnost různých výřezů dvou snímků a ona by nám odpovídala procentuální shodou. Pro účely rekonstrukce objektu z obrazu se tyto sítě zatím nepoužívají, protože vždy potřebujeme znát přesnou polohu (ideálně s přesností na jeden pixel) klíčového bodu. V případě použití neuronové sítě bychom museli stále zpřesňovat oblast, ve které daný bod je, dokud bychom nenašli maximální shodu. Výpočty pomocí neuronové sítě jsou obecně náročnější na výpočetní výkon, což také nehovoří v jejich prospěch.

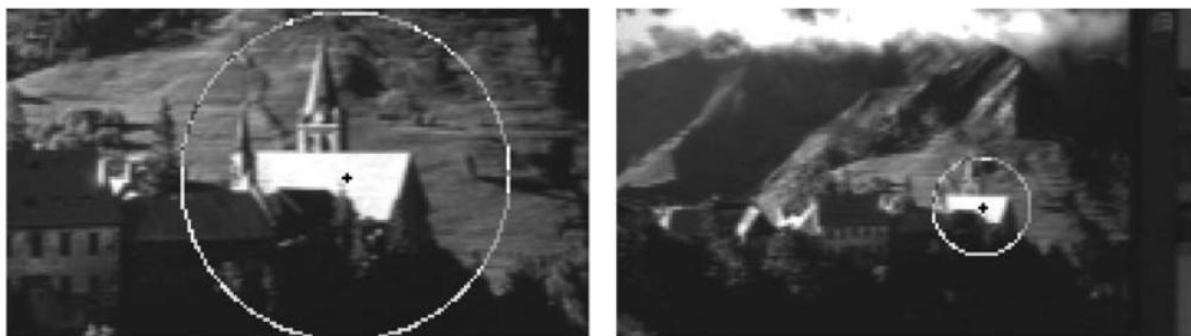
1.3.1 Houghova transformace

Tento způsob detekce klíčových bodů lze rozdělit do dvou základních skupin. Klasická Houghova transformace se zabývá hledáním snadno popsatelných tvarů, jako jsou přímky nebo kružnice. Složitějším způsobem implementace je tzv. Obecná Houghova transformace, která umožňuje detekci složitějších vzorů v obraze, které nelze popsat parametricky. Princip metody je založen na postupném posouvání hledaného motivu obrazem. Z toho plyne relativně vysoká náročnost na výpočetní výkon. Velkou nevýhodou je invariance vůči velikosti a natočení hledaného objektu. Pro nalezení kompletní shody by tedy bylo nutné obraz prozkoumat několikrát a po každém průchodu měnit tvar hledaného objektu. Na druhé straně metoda přináší výhodu ve vysoké odolnosti proti šumu. Tato metoda nebyla v práci použita, proto se jejím vysvětlením práce dále nezabývá. Detailnější vysvětlení je možné nalézt v [9], [10], [11] a celou problematiku shrnuje [12].

1.3.2 SIFT

První myšlenky detektoru, který bude ze samotného principu invariantní vůči velikosti a natočení, představil David G. Lowe v roce 1999 (viz [13]). Na své práci dále pokračoval a tak v roce 2004 představil první implementaci algoritmu SIFT (Scale Invariant Feature Transform) popsáno v [14]. V následujících odstavcích je princip algoritmu zjednodušen. Pro detailní pochopení detektoru je vhodné si přečíst [14], kde je princip popsán více do detailu, včetně názorných grafů a obrázků.

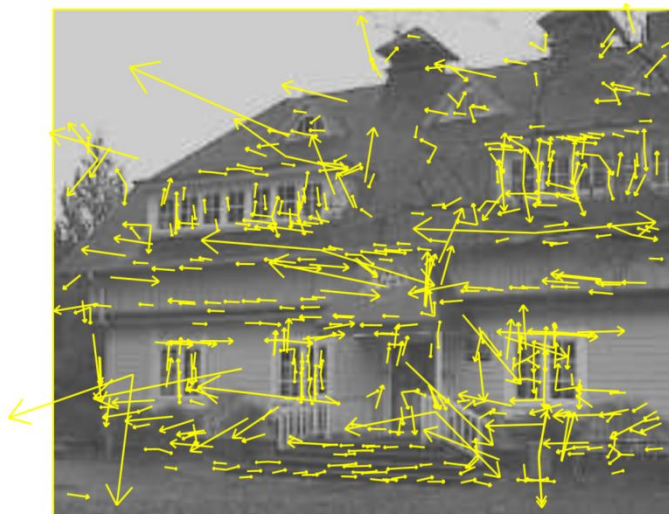
Princip algoritmu je rozvržen do dvou základních částí, kde každá z nich má několik kroků. Tou první je detektor, který lokalizuje pozici klíčových bodů. Druhou částí je deskriptor, který nalezené body popíše deskriptorem, neboli velikostí klíčového bodu a hlavním směrem jeho gradientu. Prvním krokem detekce bodů je aplikování filtrace obrazu pomocí Gaussova filtru s různým koeficientem sigma. Vznikne tedy několik verzí obrazu (nazývané Gaussiány) od původního až po úplně vyfiltrovaný. Následně by se podle původní studie měl aplikovat Laplaceův operátor na právě vypočtené Gaussiány (v originální terminologii – Laplacian of Gaussians – LoG). Implementace SIFT publikovaná v roce 2004 ovšem přinesla značné zjednodušení v tomto výpočtu. Namísto aplikace, výpočetně náročného, LoG stačí provést pouze jejich rozdíl (v originální terminologii – Difference of Gaussians – DoG). Tam, kde je vysoký rozdíl mezi jednotlivými Gaussiány, je podezření na klíčový bod. Aplikace různého koeficientu sigma Gaussova filtru je základem pro invarianci detekce na velikosti oblasti, kde je klíčový bod. Názorná ukázka je na *Obr. 2*, kde je na obou fotografiích zachycen stejný kostel, avšak na každé fotografii je jinak veliký. Kružnice znázorňují oblasti, kde detektor zareagoval a označil bod jako možný klíčový.



Obr. 2: Detektor SIFT algoritmu. Převzato z [16].

Z podezřelých klíčových bodů je následně určeno, který z podezřelých bodů je skutečně bodem klíčovým. Toto rozlišení je založeno na tzv. stabilitě bodu. Stabilita se určuje z několika parametrů a má za cíl vyřadit body, které mají například nízký kontrast a tudíž je šance, že se jedná o šum, nebo se jedná o body u okraje.

Tím je zakončena detekce klíčových bodů. A nyní je nutné vypočítat deskriptor. Pomocí deskriptoru je možné od sebe body rozlišit. Při porovnávání klíčových bodů dvou snímků jsme schopni určit, které dva body jsou podobné a následně je napříč snímky spárovat. Prvním krokem výpočtu deskriptoru je výpočet lokálních gradientů v okolí všech klíčových bodů. Výpočet probíhá v každém pixelu vyfiltrovaného obrazu v okolí klíčového bodu. Počítá se u něj velikost a natočení, což zajišťuje invarianci k natočení objektu. Z těchto lokálních gradientů se pro každý klíčový bod vytvoří jejich histogram. Pomocí nejvyšší položky tohoto histogramu se natočí oblast v okolí bodu a opět se vypočítá histogram lokálních gradientů v tomto okolí. Tento krok napomůže v invarianci na natočení kamery v okamžiku pořízení fotografie. Nový histogram je následně normalizován kvůli vytvoření invariance na různém osvětlení objektu. Výsledný úhel a velikost deskriptoru je určen z nejvyšší hodnoty histogramu.



Obr. 3: Snímek s vyznačenými deskriptory získané metodou SIFT. Převzato z [14].

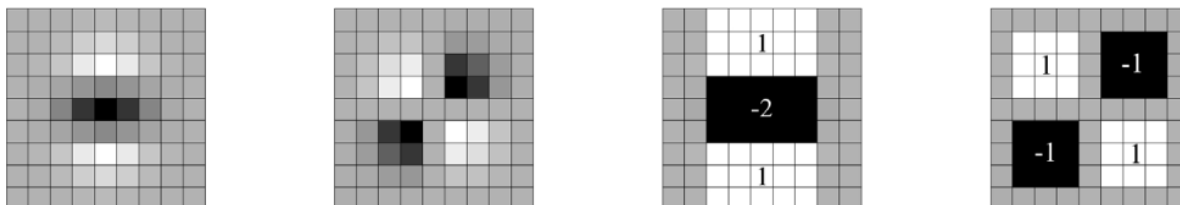
1.3.3 SURF

Metoda SURF (Speeded-Up Robust Features) je modernější verzí metody *SIFT*. Byla publikována roku 2006 Herbertem Bay (viz [15]). Hlavním kladem této metody oproti metodě původní je její rychlost. Detekce klíčových bodů je 3 krát rychlejší než u metody *SIFT*

(viz Table 2 ve zdroji [15]). Rychlejší metody obvykle začínají postrádat některé funkce. Typickým příkladem je detektor FAST s popisovačem BRIEF, který patří k nejrychlejším ve zpracování, ale nenabízí tak vysokou spolehlivost (není invariantní na natočení a různé nasvětlení objektu). Detektor a deskriptor metody SURF ovšem žádnou výraznou ztrátou informace ani stability netrpí. Jeho autoři dokonce slibují větší odolnost proti různým transformacím obrazu. Díky těmto vlivům začal ve velkém nahrazovat metodu *SIFT*.

V současné době byl algoritmus výpočtu zjednodušen o eliminaci invariance vůči natočení. Zjednodušení výpočtu vede k jeho celkovému zrychlení. Tato nová verze se nazývá U-SURF. Pro účely této práce ovšem není vhodná, protože ve vytvořeném zařízení je mezi každou kamerou a objektem velmi odlišný úhel. Algoritmus tedy nebude dále vysvětlován a následující odstavce se budou věnovat popisu původně publikované metodě SURF.

Namísto výpočtu Gausiánů se využívá jejich aproximace čtvercovými filtry. Pro ještě větší zjednodušení je realizován výpočet jejich tzv. integrálního obrazu. Jeho jednotlivé prvky jsou součtem původních hodnot intenzit pixelů v daném řádku a v daném sloupci konkrétní oblasti. Z tohoto obrazu se následně vezmou jen jeho krajní hodnoty. Zjednodušení filtru oproti metodě *SIFT* je možné vidět na *Obr. 4*.



Obr. 4: Levé dva snímky znázorňují parciální derivaci druhého řádu Gaussovy křivky. Vpravo je jejich aproximace použita v metodě SURF. Obrázek převzat z [15].

K nalezení klíčových bodů se následně využívá tzv. blob detektor založený na Hessově matici. „*Tento detektor není příliš náročný na výpočetní výkon a zároveň disponuje velmi vysokou přesností.*“ (přeloženo ze zdroje [15] z kapitoly *Fast-Hessian Detector*). Hessova matice je vytvořena konvolucí obrazu v daném bodě se zjednodušenou druhou derivací Gaussovy křivky (viz *Obr. 4*). Jak je uvedeno ve zdroji [15], tak právě díky zmíněné konvoluci je výpočet stejně dlouhý pro libovolnou velikost Gaussovy křivky, což vede k nezávislosti délky výpočtu na velikosti této křivky. Tato nezávislost vede k dalšímu zrychlení

1.4 Princip rekonstrukce objektu z obrazu

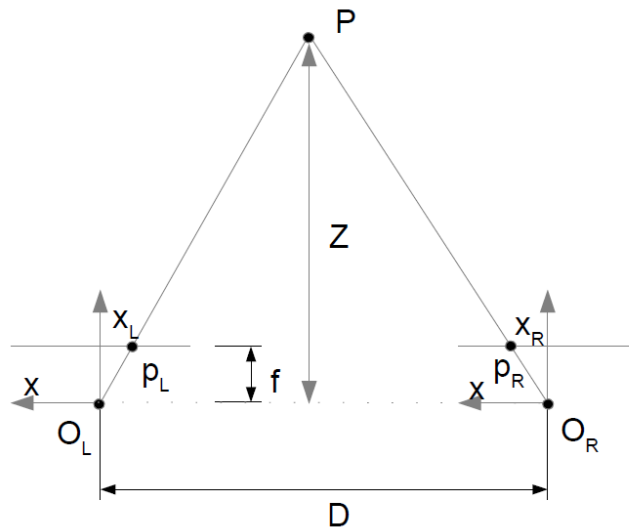
Základním předpokladem rekonstrukce modelu objektu z obrazu je, že jsou k dispozici alespoň dvě fotografie objektu pořízené z různých pozic. V lepším případě je k dispozici fotografií několik desítek a v reálných aplikacích, kde je důraz na přesnost, jsou jich pořizovány až stovky. Dalším základním kamenem pro úspěšnou rekonstrukci je, že na obou těchto fotografiích je objekt znázorněn ze stejné strany. Pozice fotografa tedy nesmí být příliš daleko od sebe. Různé zdroje se shodují, že nejvyšší úhel pro úspěšnou rekonstrukci je 30° . Jedná se o úhel ($O_L P O_R$) na *Obr. 7*. Při praktické realizaci se ukázalo, že je tento úhel příliš velký (více informací v kapitole 2.1.3).



Obr. 6: Různé pohledy na stejný objekt

Základním nástrojem pro rekonstrukci trojrozměrného modelu z jednotlivých fotografií je stereo vize. Jedná se o principy založené na dvou snímacích prvcích. Celá tato problematika byla inspirována viděním trojrozměrného světa člověkem (tzv. binokulární vize). Z matematického hlediska se jedná o princip triangulace, kde z definované pozice kamer je možné dopočítat vzdálenost bodu od jejich spojnice a následně definování pozice bodu v trojrozměrném prostoru. Situace naznačená na *Obr. 7* je spíše ilustrativního charakteru. Obě kamery (naznačené body O_L a O_R) totiž mají paralelní optickou osu, čehož je v reálné aplikaci prakticky nemožné docílit. Body P_L a P_R reprezentují pozici promítaného bodu P na pomyslné rovině reprezentující výsledný snímek. Vzdálenost f reprezentuje ohniskovou vzdálenost použité kamery a Z vzdálenost promítaného bodu od pomyslné spojnice kamer. Postupů, jak zjistit vzdálenost Z je několik a tím nejjednodušším je pomocí podobnosti trojúhelníků $O_L P O_R$ a $P_L P P_R$. Se známou vzdáleností Z je následně možné spočítat vzdálenost od

jednotlivých kamer. S pomocí pozice bodů na snímku je následně možné určit přesnou pozici bodu v prostoru.



Obr. 7: Stereo vize (zdroj [21])

Nastíněný princip vychází z předpokladu, že je na obou snímcích nalezen totožný bod. Toho se obvykle docílí pomocí *Detekce klíčových bodů v obraze* a následné filtraci spárovaných hodnot. Filtrace se obvykle realizuje porovnáním jednotlivých párů. Jednotlivé páry musí mít podobné parametry, např. z hlediska vzdálenosti jednotlivých bodů mezi snímky atp. Pokud některý z párů vybočuje z parametrů, které splňuje většina párů, tak je označen jako nesprávný a rekonstrukce se pro něj neprovádí.

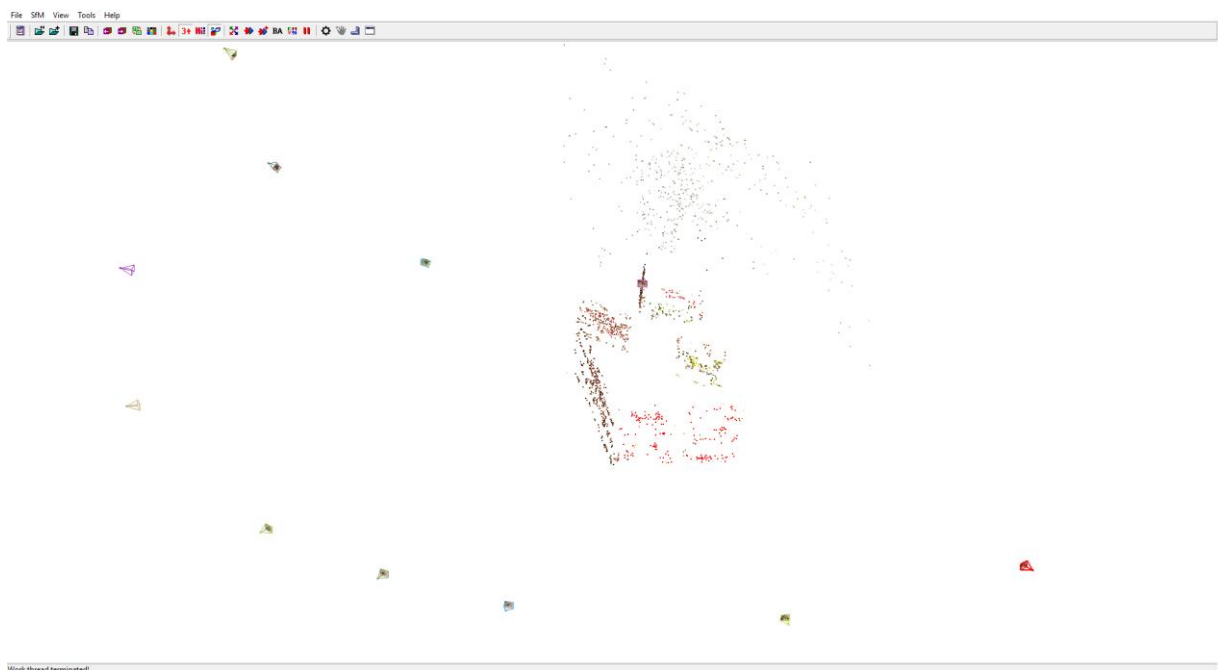
Tyto základní myšlenky umožňují výpočet vzdálenosti bodu od kamer. Při snaze rekonstruovat model objektu, který bude složen například ze 100.000 bodů, by bylo zapotřebí nalézt 100.000 párů klíčových bodů na všech snímcích. Toho lze docílit velice komplikovaně, a proto se využívá jiného principu – *epipolární geometrie*. Tato problematika je velice kvalitně vysvětlena v [22].

1.5 Programy pro rekonstrukci trojrozměrných modelů

Programů, které jsou schopny rekonstruovat trojrozměrné objekty, je velké množství a dělí se do dvou základních kategorií. Produkty zdarma a produkty komerční. Z každé této kategorie byl vybrán jeden zástupce, který byl shledán jako vhodný pro použití.

Zdarma

Tyto programy jsou nejčastěji vyvíjeny univerzitami, které se věnují výzkumu algoritmů pro tuto oblast. V rámci jejich vědecké činnosti vznikají tyto programy, aby v nich mohli testovat vlastní řešení problematiky rekonstrukce modelu objektu z fotografií. Pro tuto práci byl vybrán program VisualSFM vyvíjený univerzitou *University of Washington*. Jeden z hlavních autorů Changchang Wu této aplikace se podílel i na známém zajímavém projektu (viz [20]), při kterém byly vytvářeny rekonstrukce modelů známých památek pomocí fotek získaných na internetu.



Obr. 8: Program VisualSFM pro rekonstrukci modelu objektu

Testovanými alternativami byly programy Regard3D, ve kterém nebyly výsledky rekonstrukce dostatečně uspokojivé, a 123DCatch, který ovšem nenabízí snadné programové rozhraní pro přístup z jiného programu (API).

Komerční produkty

Jako zástupce komerčních produktů byl vybrán program Recap360 vyvíjený společností Autodesk. Tento program je možné využít jako aplikaci nainstalovanou v počítači nebo formou internetového prohlížeče (viz *Obr. 9*). Při použití internetového prohlížeče se rekonstrukce provádí na serverech výrobce aplikace a při jejím dokončení přijde uživateli email. Služba vyžaduje nejméně 20 fotografií a v případě nižšího počtu upozorní uživatele, že rekonstrukce nebude pravděpodobně příliš kvalitní. Nejlevnější licence (pro jeden počítač na jeden rok) stojí 300 \$³ (při kurzu 1 \$ = 24,54 Kč se jedná o 7360 Kč). Výhodou je, že pro studentské účely je použití programu zdarma. V tomto projektu tedy bylo využito této nabídky.



Obr. 9: Program Recap360 pro rekonstrukci modelu objektu

³ Cena přejata z <https://www.autodesk.com/products/recap/subscribe> ke dni 9.5.2017

2 Praktická realizace

Realizace zařízení se skládá ze dvou základních částí. Tou první je realizace mechanické konstrukce (viz 2.1) a tou druhou je programové vybavení (viz 2.2 a 2.3). Obě tyto části tvoří nerozdělitelný celek, pomocí kterého lze dosáhnout požadované rekonstrukce modelu objektu. Obě části mají potenciál na zlepšení, avšak se současnou konfigurací je možné dosáhnout více než uspokojivých výsledků rekonstrukce modelu objektu (viz Obr. 9).

2.1 Mechanická konstrukce

Stavba mechanické konstrukce byla rozdělena do dvou fází. Nejdříve byl vytvořen prototyp (viz 2.1.2) a poté z nasbíraných zkušeností bylo postaveno výsledné zařízení (viz 2.1.3). Jelikož byl výběr kamer a jejich následné mechanické úpravy obtížným úkolem, je mu věnována celá následující podkapitola.

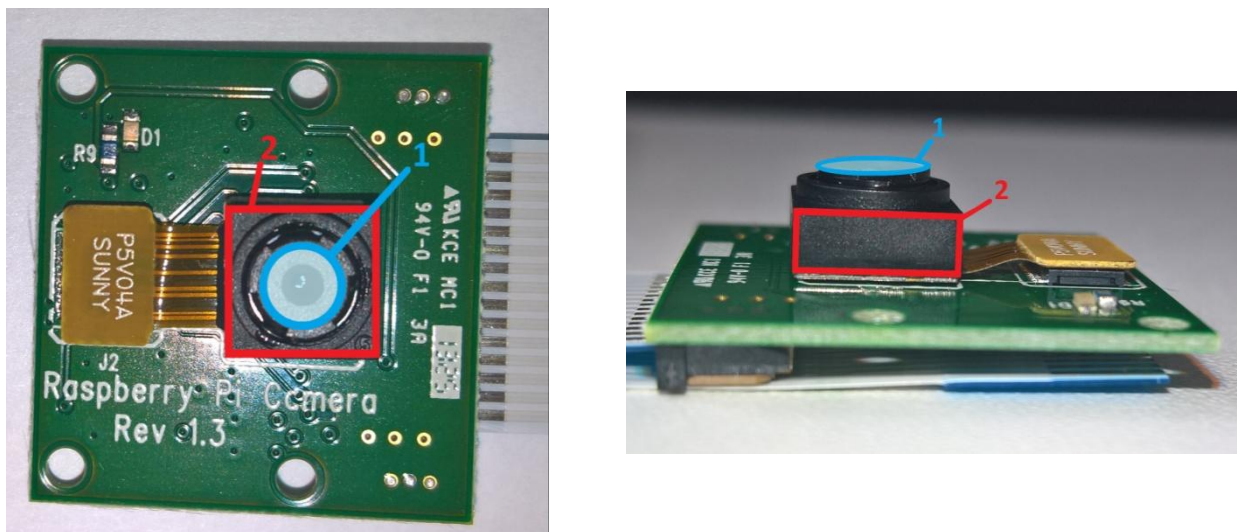
2.1.1 Kamery

Pro praktickou realizaci byly vybrány kamery RaspberryPi Camera ve verzi 1.3. Jejich výběr nebyl ani tak volbou jako jediným východiskem. Byly totiž pro účel této práce pořízeny v dostatečném množství ještě před samotným vypsáním zadání této práce. V prototypu byly tyto kamery umístěny v oficiálně prodávané krabičce k tomuto kamerovému modulu (viz Obr. 10). Kvůli zbytečně vysoké ceně nebyly tyto kryty do finálního výrobku použity a kamery byly připevněny přímo na konstrukci (viz Obr. 16).



Obr. 10: Původní uchycení kamerových modulů v oficiálně dodávané krabičce.

Tyto kamery přinesly ovšem nečekané komplikace z hlediska ostření. Některé kamery připojitelné k PC disponují funkcí automatického optického ostření. To, bohužel, není případ těchto kamer. Nejlevnějším způsobem je zaostřit kameru na nekonečnou vzdálenost a tím je použitelně pokryto celé vzdálenostní spektrum. Není to ovšem příliš použitelné, pokud je od kamery vyžadována vysoká ostrost pro konkrétní vzdálenost. Zaostření na nekonečno je u těchto kamer také nevýhodné v relativně nízké ostrosti pro objekty blíže než 1 metr. Pro účel zvýšení ostrosti byla těmto kamerám mechanicky upravena vzdálenost čočky od snímacího čipu, čímž byla docílena změna ostřicí vzdálenosti. Změnu vzdálenosti je možné docílit vyšroubováním vnitřního plastového kroužku, který uprostřed drží čočku (viz modré označení 1 na Obr. 11). Toho lze docílit pevným uchycením základny čipu (viz červené označení 2), následným pevným mechanickým uchopením části s čočkou a pootočením proti směru hodinových ručiček. Velikost otočení byla dlouho řešeným problémem, protože výrobce kamer tuto úpravu nedoporučuje (je možné kameru u této operace nevratně mechanicky poškodit). Nakonec byl nalezen empiricky získaný vzorec (viz (2) - převzatý ze zdroje [3]), kterým byl pro vzdálenost 50cm vypočítán úhel 25,2°.



Obr. 11: Pohled shora (vlevo) a z boku (vpravo) na kameru RaspberryPi Camera verze 1.3 s modře naznačeným uchycením čočky (1) a červeně naznačenou základnou (2)

$$\text{úhel } [^\circ] = \frac{3,5}{\text{vzdálenost od objektu [cm]}} * 360 \quad (2)$$

Výsledné zlepšení je patrné z *Obr. 12*, kde je z fotografií patrné značné zlepšení. Na obou snímcích je zachycena etiketa láhve ze vzdálenosti 50cm. Na levém snímku byl přechod dvou barev, který je v realitě velice ostrý, rozprostřen na přibližně 5 až 6 pixelů. Na snímku pravém (snímek pořízen po korekci ostřicího bodu) je tento přechod na úrovni přibližně 2 až 3 pixelů. Zlepšení je značné, což ovšem nevylučuje potenciál v dalším zlepšování.



*Obr. 12: Úprava ostřicího bodu použitých kamer.
Vlevo je výřez fotografie před korekcí, vpravo po korekci.*

Zaostřením kamer se zabývá problematika hloubky ostrosti (v anglickém originále Depth of Field). Díky této problematice je možné spočítat, pro jaké vzdálenosti bude obraz ještě v přijatelné ostrosti. Základní rovnice (3) této problematiky nám umožňuje spočítat hyperfokální vzdálenost. V této rovnici f zastupuje ohniskovou vzdálenost, N clonové číslo (v anglickém originále f -number) a c rozptyl (v anglickém originále *Circle of Confusion*).

$$H = \frac{f^2}{N * c} + f \quad (3)$$

Spočítaná hyperfokální vzdálenost dále umožňuje spočítat nejmenší (viz rovnice (4)) a největší (viz rovnice (5)) vzdálenost objektu od kamery, kdy je objekt na snímku možné považovat ještě za ostrý. V obou těchto rovnicích vystupuje proměnná s , zastupující vzdálenost kamery od objektu.

$$D_n = \frac{s * (H - f)}{H + s - 2f} \quad (4)$$

$$D_f = \frac{s * (H - f)}{H - s} \quad (5)$$

Rovnice (3), (4) a (5) byly převzaty ze zdroje [4]. V případě použití kamer RaspberryPi Camera ve verzi 1.3 byly za jednotlivé proměnné dosazeny následující hodnoty:

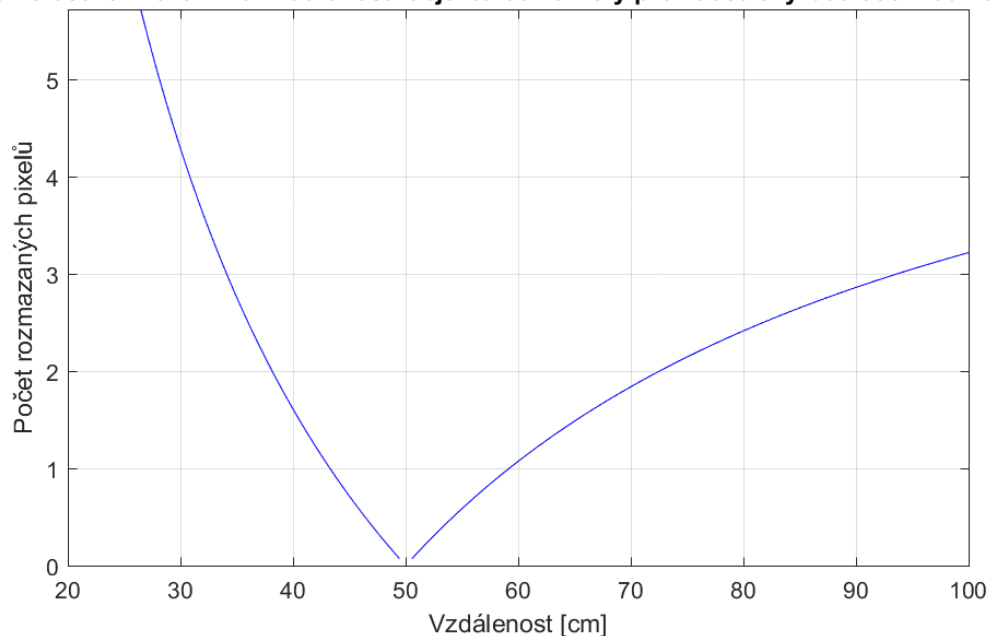
- $f = 3,6\text{mm}$
- $N = 2,9$
- $c = 0,0028\text{mm}$ (šířka jednoho pixelu je $1,4\mu\text{m}$, akceptované rozostření jsou 2 pixely)
- $s = 500\text{mm}$

Z těchto hodnot byly spočítány hodnoty vzdáleností:

- $D_n = 38,14\text{cm}$
- $D_f = 72,57\text{cm}$

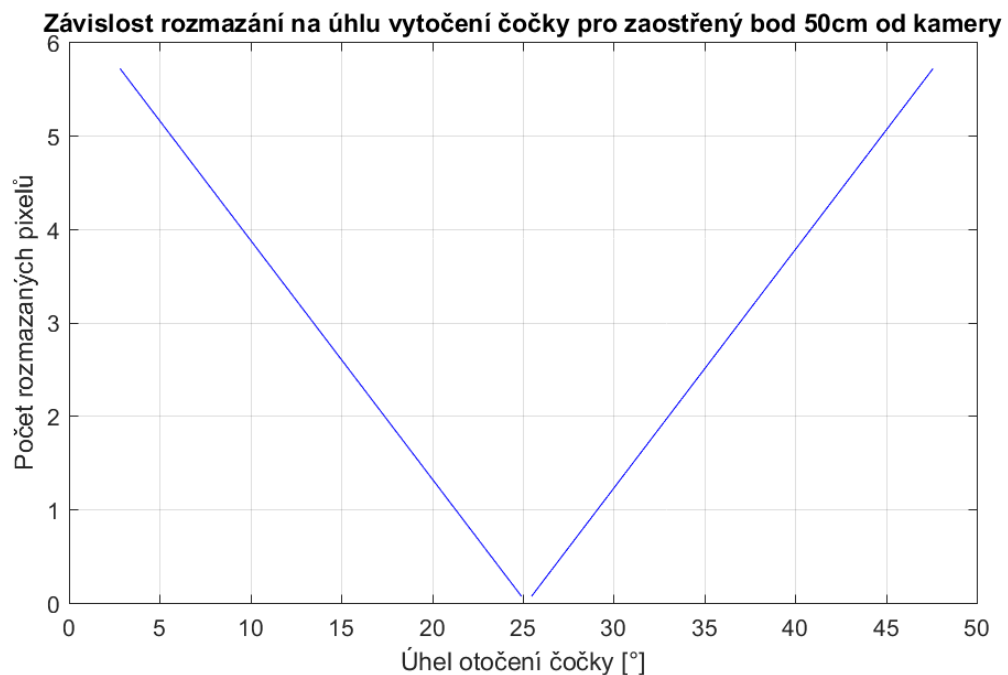
Pro názornost byla pro tento scanner vykreslena závislost počtu rozmazaných bodů na vzdálenosti (viz Obr. 13). Z té je možné vyčíst poloměr kruhu pixelů, které budou na obrázku reprezentovat bod, který by při zaostření na danou vzdálenost byl reprezentován jediným pixelem. Graf na zmíněném obrázku je ovšem platný pouze v případě, že se podaří kameru zaostřit přesně na vzdálenost 50cm.

Závislost rozmazání na vzdálenosti objektu od kamery pro zaostřený bod 50cm od kamery



Obr. 13: Znáznornění počtu rozmazaných pixelů na vzdálenosti objektu od kamery pro zaostřený bod 50cm od kamerové čočky

Graf vykreslený na *Obr. 14* znázorňuje podobnou charakteristiku jako graf na *Obr. 13*. Neznázorňuje ovšem závislost polohy objektu na rozmazání, ale závislost přesnosti úprav kamery RaspberryPi verze 1.3. Konkrétně se jedná o závislost počtu rozmazaných pixelů na úhlu, o který je nutné otočit čočku umístěnou ve šroubovém osazení, pokud chceme docílit ostřicí vzdálenosti 50cm. Z grafu je také patrné, že pokud není na kameře provedena úprava (úhel otočení čočky je roven 0), tak je rozmazání přibližně 6 pixelů, což bylo potvrzeno při porovnání fotografií před a po korekci (viz *Obr. 12*).

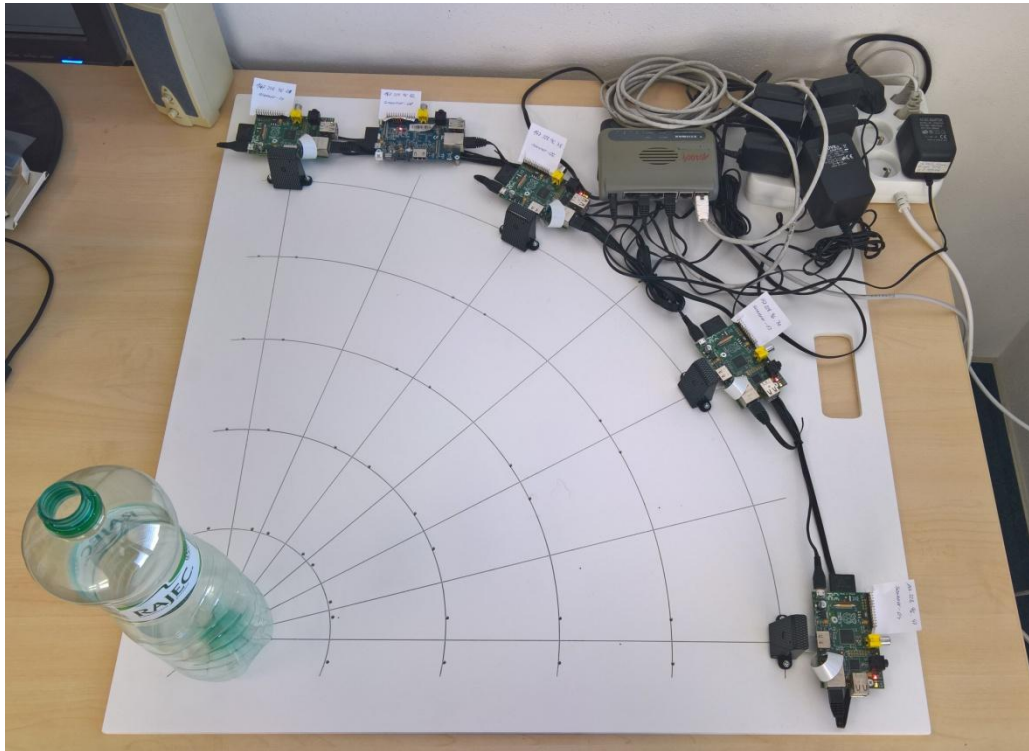


Obr. 14: Znázornění počtu rozmazaných pixelů na úhlu vytočení čočky kamery RaspberryPi Camera verze 1.3 pro požadovanou ostřicí vzdálenost 50cm

2.1.2 Prototyp

Prototypová verze mechanické konstrukce sloužila primárně k nalezení nejvhodnější vzdálenosti kamer od objektu a vzájemnému úhlu mezi kamerami. Pro nalezení požadovaných parametrů byla vytvořena konstrukce zobrazená na *Obr. 15*. Jedná se o scanner snímající objekt pouze z jedné čtvrtiny (úhel mezi postraními kamerami je 90°). Tento prototyp také umožnil odladění algoritmů, které byly později použity ve výsledném výrobku.

U této verze nebyl primárním cílem vzhled, ale funkčnost. To je hlavní důvod, proč jsou veškeré kabely a zařízení na povrchu mechanické konstrukce. Pozice kamer byly často měněny, než byla nalezena požadovaná konfigurace. Bylo tedy i zapotřebí často měnit délku vodičů nebo umístění pomocných přístrojů (například zdroje elektrické energie).



Obr. 15: Prototyp scanneru

Výsledné zjištění bylo, že kamery by měly být od objektu vzdáleny 50cm. Pro menší vzdálenosti omezovala pozice kamer maximální velikost objektu. Pro vzdálenosti větší docházelo naopak ke zbytečně ztrátě informací, protože objekt zabíral pouze malou část fotografie a nepříliš vysoké rozlišení kamer objekt začínalo zkreslovat. Větší vzdálenost by měla také negativní vliv na celkovou velikost mechanické konstrukce. Úhel mezi kamerami byl v prototypu 30° , protože to je nejvyšší doporučovaný úhel mezi kamerami v podobných aplikacích. Bylo ovšem zjištěno, že 30° je úhel příliš veliký a s menším úhlem je možné docílit značného zlepšení. Přináší to ovšem požadavek na dvojnásobné množství kamer nebo se stejným počtem kamer skenovat pouze jednu polovinu objektu.

2.1.3 Finální výrobek

Konečná verze mechanické konstrukce se od prototypu liší primárně typem mechanické konstrukce ve stylu uzavřené skříně a dvojitým způsobem uchycení kamer. Zatímco prototyp obsahoval pouze 4 kamerové moduly (kamera + minipočítač RaspberryPi) s rozestupy 30° v jedné rovině, konečná verze obsahuje kamerových modulů 12 s rozestupy nejvýše 15°. Tato konfigurace umožní získat mnohem lepší výsledky v rekonstrukci.



Obr. 16: Způsob uchycení kamer ve finálním výrobku

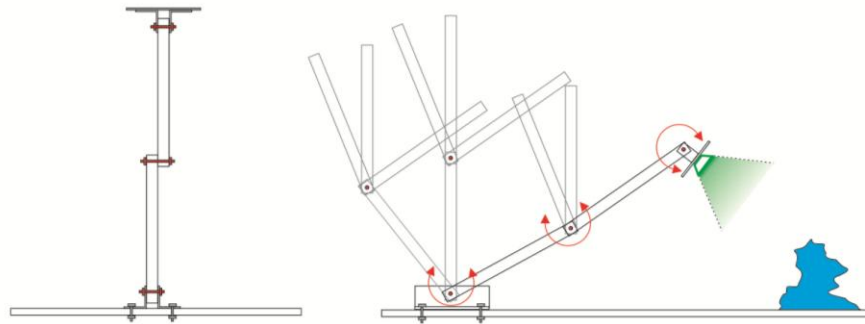
Myšlenka uzavřené skříně byla nevyhnutelná z důvodu snadné použitelnosti a snadné přenositelnosti celého zařízení. Na Obr. 15 je možné vidět příliš velké množství napájecích vodičů, síťových vodičů a dalších prvků, které snadnému použití běžným uživatelem brání. Do výsledné mechanické konstrukce byl umístěn jednotný zdroj energie, ze kterého jsou napájeny všechny elektrické komponenty a jeden síťový přepínač (v anglickém originále switch) s dostatkem portů, pro zajištění komunikace všech prvků. Veškeré vodiče jsou tedy umístěny uvnitř této skříně, kde jsou připevněny k mechanické konstrukci. Skříň se skládá ze dvou základních částí, kde spodní část obsahuje komponenty jako například zdroj energie a krycí díl, kde jsou umístěny kamery. Mezi těmito prvky jsou mechanické vzpěry, které zajistí stabilitu konstrukce v případě skenování těžkého objektu. Krycí díl je se spodní částí elektricky propojen pomocí snadno rozpojitelných konektorů, čímž zjednodušuje jakoukoliv demontáž konstrukce. Obě tyto části je možné vidět na Obr. 33 (umístěn v přílohách), kde spodní část je umístěna na pravé straně fotografie a krycí díl na straně levé. Mechanicky robustní a uzavřená konstrukce umožnila mít v konstrukci zabudován síťový napájecí konektor typu EURO (dnes je standardem všech elektronických zařízení napájených z elektrické rozvodné sítě) a panelový konektor RJ45 pro přenos dat. Pro provoz scanneru ho tedy stačí připojit jedním kabelem do elektrické rozvodné sítě a druhým do sítě ethernetové.



Obr. 17: Finální výrobek

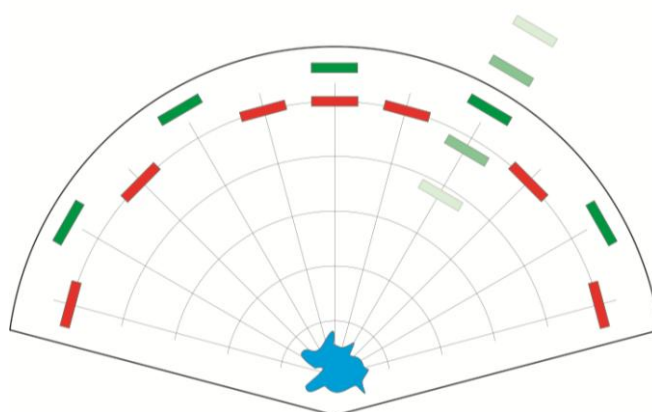
Jedním z hlavních důvodů použití uzavřené mechanické konstrukce je i důraz na kvalitní design výrobku. V mnoha směrech byla snaha minimalizovat počet prvků narušujících vzhled. Do konstrukce byly přidány prvky, které zaručují unikátní design, i za cenu značných komplikací ve výrobním procesu. Jedním ze zástupců jsou gravírované drážky do krycího dílu znázorňující umístění objektu, kamer a jejich vzájemných mechanických vztahů (vzájemný úhel a vzdálenost). Tyto drážky je možné vidět na *Obr. 17*. Z hlediska designu výrobku je dozajisté co zlepšovat, protože to nebylo hlavním cílem tohoto projektu.

Zatímco prototypová verze má kamery pouze v jedné rovině, tak tato verze obsahuje osazení kamer ve více rovinách se snadno měnitelnými pozicemi. Umožnila to především hliníková konstrukce polohovacích ramen. Pozice kamer na konci těchto hliníkových ramen lze snadno měnit povolením křídlových matic v jednotlivých kloubech a změnou natočení ramen. Variabilitu natočení lze vidět na *Obr. 18*, kde modrý prvek znázorňuje snímáný objekt. Původní koncepce sice počítala s umístěním kamer ve více rovinách, ale kamery měly být připevněny na pevnou konstrukci. Následně byla ovšem vymyšlena tato konstrukce umožňující snadnou změnu polohy a díky tomu je možné do scanneru vložit i objekty větších rozměrů. Poté stačí pouze změnit polohu ramen.



Obr. 18: Princip tříkloubového polohovacího ramena kamer

Původní koncepce výsledné mechanické konstrukce počítala s pokrytím snímků ze všech stran objektu. Na prototypu bylo ovšem odzkoušeno, že rozpětí mezi kamerami 30° je příliš velké a nedosahuje se díky tomu k výsledkům požadované kvality. Z toho důvodu byl proces skenování omezen jen z jedné poloviny objektu, ale o to detailněji. Byl tedy zvolen úhel mezi kamerami 15° v celkovém úhlu 150° . Pro detailnější rekonstrukci z pohledu ze středu scanneru byla do této oblasti přidána další kamera. Osazení kamer na konstrukci je znázorněno na Obr. 19, kde červené obdélníky znázorňují umístění kamer v rovině podkladové desky a zelené obdélníky kamery umístěné na ramenech. Modře je symbolizován snímáný objekt. Skenování objektu jen z jedné poloviny otevírá nové možnosti v pokračování na projektu ve směru automatizovaného složení výsledného modelu ze dvou polovin nebo točící se konstrukci, na které by byl objekt umístěn. S každým otočením této konstrukce by se provedla aktualizace stávajícího modelu o nově získaný pohled.



Obr. 19: Osazení kamer na mechanické konstrukci. Červeně jsou znázorněny kamery připevněné k podkladové desce a zeleně pozice kamer na polohovacích ramenech (včetně naznačené osy pohybu).

2.2 Program pro rekonstrukci objektu

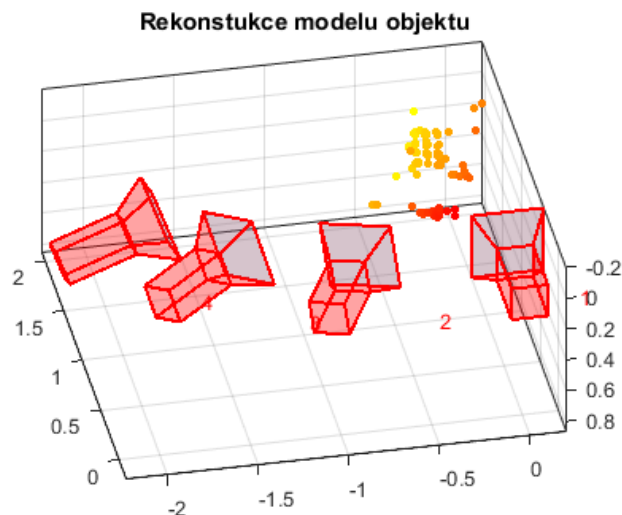
Jak již bylo popsáno v kapitole 1.4, je problematika rekonstrukce objektu principiálně jednoduchá, avšak její programová implementace již tak triviální není. I přes tento fakt byla snaha o implementaci vlastního programového řešení. Touto částí vývoje se dále zabývá kapitola 2.2.1. Tímto způsobem bylo docíleno jistých výsledků, avšak tyto výsledky nedosahovaly kvalit profesionálních programů. Z toho důvodu bylo postupně upuštěno od tvorby vlastních algoritmů a bylo použito hotové řešení, které je popsáno v kapitole 2.2.2.

2.2.1 Experimentální implementace

Vlastní implementace algoritmů popsaných v 1.4 byla motivována snahou si tyto algoritmy vyzkoušet v praktickém nasazení. Pro zjednodušení byly všechny tyto algoritmy psány v prostředí MATLAB ve verzi 2016 s rozšířením zvaným Computer Vision. Veškerá obrazová data je možné v počítači reprezentovat jako matici pixelů a právě proto byl použit program MATLAB, který je v tomto směru velice efektivní a vysoce optimalizovaný. Zároveň je již skoro nepsaným standardem v tomto odvětví. Rozšíření Computer Vision přináší do programu MATLAB funkce právě v odvětví zpracování obrazu. Nevýhodou v praktickém použití je vysoká pořizovací cena programu MATLAB a navíc cena za rozšíření Computer Vision.

Prvním krokem vytvořených algoritmů je nalezení párů fotek, které byly vytvořeny kamerami fyzicky umístěných blízko sebe. Toho je možné docílit pomocí detekce klíčových bodů pomocí algoritmu *SURF*. Použití této metody se ukázalo problematické v tom, že nejsilnější výsledné deskriptory byly často umístěné mimo objekt a algoritmus se tak snažil rekonstruovat úplně jiný objekt (například stropní svítidlo). Tento problém byl vyřešen tím, že byla pořízena fotografie prázdné scény a následně scény s objektem. Následně byla vybrána oblast, kde neproběhla žádná změna (neboli nezajímavé okolí objektu) a to bylo nahrazeno šumem. V tomto šumu metoda *SURF* nenalezla žádný klíčový bod a všechny její výsledné deskriptory byly umístěny na objektu.

Následně bylo postupováno podle zdroje [19], kde je kompletní postup potřebný pro rekonstrukci objektu. Tyto příklady byly upraveny pro použití s více kamerami a otestovány. Výsledky ovšem nebyly požadované kvality. Značné množství bodů neleželo na původním objektu a velké množství bodů chybělo. Výsledek je ostatně možné vidět na *Obr. 20*.



Obr. 20: Výsledek vlastní implementace rekonstrukčních algoritmů

2.2.2 Výsledné řešení

Pro výsledné zařízení bylo nakonec použito programu, který je vyvíjen velkými týmy a dosahuje tak značně lepších výsledků, než vlastní implementace. Jak bylo popsáno v 1.5, tak je možné použít program zdarma nebo placený. Program, vytvořený pro komunikaci se scannerem (viz 2.3.3 a 2.6), je uzpůsoben na přímé napojení na program VisualSFM vyvíjený *University of Washington*. Program je možné spustit samostatně a ovládat si parametry rekonstrukce manuálně nebo tento program ovládat pomocí vytvořené uživatelské aplikace.

Alternativou je použití služby Recap360, která je sice placená, avšak pro studijní účely je možné získat licenci zdarma. Rozdíl kvality modelu mezi těmito dvěma programy je velmi náročné určit. Největší rozdíl mezi těmito programy je ve výsledném formátu souboru. Zatímco program VisualSFM poskytuje soubory ve formátu *ply*, které obsahují pouze pozice bodů, tak výsledným produktem programu Recap360 je soubor ve formátu *obj*, který obsahuje i spojení bodů pomocí triangulace. Rozdíl je na první pohled patrný při zobrazení obou těchto souborů, protože při zobrazení souboru s triangulací se zobrazí i přechody mezi body (viz Obr. 9) a model tak vypadá uceleněji. Při zobrazení souboru, který obsahuje pouze pozice bodů, se zobrazí skutečně pouze tyto body, což nebudí dojem uceleného objektu (viz Obr. 8). Tyto programy mohou být porovnány kvantitativně zadáním stejných vstupních dat a zjištěním počtu zrekonstruovaných bodů. U zmíněných obrázků je při použití programu VisualSFM počet bodů 163.671 a u programu Recap360 20.514. V tomto směru je tedy program VisualSFM osmkrát lepší. Nelze tedy jednoznačně říci, který z těchto dvou programů je lepší, či vhodnější použít.

2.3 Program pro ovládání scanneru

2.3.1 Centrální úložiště snímků

Ukládání snímků je realizováno pomocí programu Samba. Jedná se o programovou implementaci síťového protokolu SMB⁴ (v novější verzi nazýván CIFS⁵). Tento program se skládá ze dvou hlavních částí:

- 1) Část běžící na serveru (u vytvořeného scanneru se jedná o řídicí minipočítač), která zajišťuje datové úložiště a politiku přístupů.
- 2) Program běžící na všech ostatních prvcích. Při realizaci scanneru byl pro tento účel vybrán program *smbclient*, který umožňuje snadný přístup ke vzdálenému serveru.

Hlavním důvodem volby programu Samba bylo jeho masové rozšíření pro podobné účely a snadná konfigurace. Jedním z vedlejších důvodů bylo jeho snadné napojení na operační systém Windows. Je tedy možné v konfiguračním souboru povolit, aby měly do sdíleného úložiště přístup zařízení i mimo scanner. Následně již stačí v osobním počítači pouze připojit síťovou jednotku a tím se otevře snadný přístup k souborům ve sdílené složce uvnitř scanneru. Takový přístup je z hlediska bezpečnosti v současné době zakázán, protože pro funkci scanneru není potřebný, a tudíž se jedná o zbytečné riziko. Autentizace uživatele probíhá pomocí jména a hesla.

Alternativou bylo použít spojení pomocí protokolu FTP⁶ nebo využít nástroj *scp* obsažený ve všech operačních systémech UNIX. Služba FTP byla shledána jako méně bezpečná (jedná se o mnohem starší systém) a zároveň náročnější na zprovoznění. Nástroj *scp* je určen pouze na přenos souborů a bylo by tedy nutné si logiku přenosu souborů naprogramovat.

2.3.2 Řídicí program scanneru

Software řídicí scanner je napsán v jazyce Java. Tento vysokoúrovňový programovací jazyk umožnil rychlý vývoj a vysokou komplexnost celé aplikace. Software se skládá ze dvou základních částí – software běžící na hlavním prvku (pojmenovaný StratoScanner) a software

⁴ Server Message Block - https://en.wikipedia.org/wiki/Server_Message_Block

⁵ Common Internet File System

⁶ File Transfer Protocol - https://en.wikipedia.org/wiki/File_Transfer_Protocol

běžící na všech ostatních prvcích (pojmenovaný Worker). Hlavní prvek má za úkol realizovat komunikaci s uživatelským rozhraním pomocí spojení TCP šifrovaného algoritmem AES a vydávat úkoly podružným prvkům ve scanneru. Program Worker má za úkol co nejdříve zareagovat na přijatý příkaz o vyfocení scény a spustit snímání. Následně tyto data uloží do složky sdílené s hlavním prvkem. Oba tyto programy se nastavují pomocí konfiguračního souboru nazvaného *Strato.conf*, který musí být umístěn ve složce, odkud jsou spuštěny oba programy. Ukázkový konfigurační soubor je možné nalézt na přiloženém médiu ve složce *Programy/Scanner* stejně tak jako zdrojové kódy obou programů.

Jak již bylo zmíněno v sekci 1.2.2, tak jedním z nejdůležitějších úkolů této aplikace bylo zajištění synchronního okamžiku vyfocení. Prvotním nápadem a pokusem o realizaci byla synchronizace externě pomocí světelné signalizace. Tento pokus byl ovšem brzy ukončen kvůli náročným zásahům do mechanické konstrukce. Každý prvek by totiž musel být v přímé viditelnosti s prvkem zařizujícím synchronizaci. Dalším uvažovaným systémem byl standardní systém na synchronizaci času v počítačových sítích – NTP. Možné zpoždění „až 100 milisekund“ (viz [1]) je pro tuto aplikaci ovšem nevhodné. Pro účely synchronizace byla tedy nakonec použita první popisovaná možnost ve zmíněné sekci 1.2.2 a to pomocí jediného UDP⁷ paketu, který je rozeslán do všech zařízení v dané podsíti. UDP pakety mají nevýhodu v tom, že není zpětná vazba, zda všechny prvky přijaly tento paket, ale na druhou stranu umožňuje komunikaci se všemi prvky pouze pomocí jednoho paketu rozeslaného všem. Bylo otestováno, že UDP pakety se obvykle ztrácejí pouze v případě, že je daná síť datově přetížená, nebo když je datově přetížen jeden z přijímacích prvků. V této práci se ovšem v jedné vteřině odešlou nanejvýš 3 synchronizační UDP pakety a poté se vyšle jeden paket, který přikazuje vyfocení scény. Přetížení tedy není příliš aktuálním problémem. V době testování řídicích algoritmů nebyl ztracen jediný paket, a tudíž byl tento systém vyhodnocen jako dostatečně spolehlivý pro tuto konkrétní aplikaci.

Při realizaci synchronizace bylo problematické měřit, jaké je časové rozpětí mezi snímky. Tato problematika byla dlouho promyšlena, až byl vymyšlen velice jednoduchý způsob – vyfocení stopek. Jedná se o myšlenku velice jednoduchou s velmi jednoduchou realizací, avšak velice přesnými výsledky. Výsledky měření jsou znázorněny na *Obr. 21*. Jak je ze snímků patrné, tak časový rozdíl mezi jednotlivými snímky je nerozeznatelný. Z naměřených

⁷ User Datagram Protocol - https://en.wikipedia.org/wiki/User_Datagram_Protocol

dat je patrné, že pro účely scanneru snímajícího statické objekty byla tato problematika vyřešena velice kvalitně. Přesnost synchronizace je vyšší než 10ms a ke kvalitnějšímu měření by byly zapotřebí lepší stopky.



Obr. 21: Test synchronizace jednotlivých prvků

2.3.3 Uživatelské rozhraní

Pro program uživatelského rozhraní byl také použit programovací jazyk Java. Primárně se jedná o grafické rozhraní mezi uživatelem a scannerem. Byla tedy využita grafická knihovna jazyka Java – JavaFX. V současné době se jedná o nejmodernější způsob programování grafických aplikací v jazyce Java. JavaFX se oproti starší knihovně Swing pyšní především multiplatformním nasazením, napojením vlastností objektů (nazýváno *Data binding*), snadnou prací s multimédií, snadnou prací s animacemi a v neposlední řadě definicí vzhledu pomocí FXML. Aplikaci, včetně zdrojových kódů, je možné nalézt na přiloženém datovém nosiči ve složce *Programy/Uživatelská_aplikace*.

Při tvorbě aplikace byla snaha postupovat podle moderních způsobů vývoje aplikací. V první řadě byla snaha o tvorbu responsivního vzhledu kvůli možnému použití na mobilních telefonech. Později bylo od tohoto kroku odstoupeno kvůli vyšším nárokům na výpočetní výkon při rekonstrukci modelu a na mobilním telefonu by byla zbytečně zdlouhavá. Druhým hlavním pilířem byla podpora pro více jazyků. Je tedy možné si v nastavení přepnout mezi anglickou a českou variantou popisů ovládacích prvků. Posledním významným prvkem bylo napojení vlastností objektů. Díky tomu je možné povolovat jednotlivá tlačítka například logickou proměnnou značící, jestli je připojení ke scanneru aktivní. V okamžiku, kdy se spojení ukončí a proměnná se nastaví na *false*, tak se automaticky (bez zásahu programu) zakážou prvky ovládající scanner.

Detailněji se vzhledem aplikace a způsobem jejího používání zabývá kapitola *Návod na používání*. Ve zmíněné kapitole lze nalézt i obrázky zachycující vzhled aplikace při jednotlivých úkonech.

Do aplikace byla snaha zanést i zobrazování trojrozměrných objektů, avšak ukázalo se to, jako nevhodná cesta. K tomuto účelu začala být využívána aplikace MeshLab (viz [6]), která umožňuje velké množství operací s modely objektů. Zprovoznění podobné funkčnosti by zabralo obrovské množství času s nejasným výsledkem. Aplikace MeshLab je zdarma ke stažení.

2.4 Porovnání vlastního řešení s komerčními produkty

Je velice náročné srovnávat vlastní řešení s komerčně dostupnými produkty. Všechny společnosti nabízející komerční produkty se pyšní několikaletou tradicí v práci s trojrozměrnými modely. Zároveň je náročné srovnání, protože komerční svět je rozdělen do dvou kategorií – scannery veliké, do kterých se pohodlně vejde člověk, a scannery ruční. Ani do jedné z těchto kategorií vyrobené zařízení nezapadá. Vyrobené zařízení je tedy primárně porovnáváno s projektem, kterým byl vznik této práce inspirován – pi3Dscan. Všechny uvedené ceny jsou platné ke dni 5.5.2017 a přepočítány podle současného kurzu 1 \$ = 24,32 Kč a 1 € = 26,74 Kč.



Obr. 22: Projekt pi3Dscan

(zdroj: http://www.pi3dscan.com/images/3dscanner/resizedimages/IMG_0066.JPG)

2.4.1 Vlastní řešení

Jak již bylo popsáno v kapitole 2.1.3, výsledné zařízení je složeno z 12ti minipočítačů RaspberryPi s kamerou a jedním řídicím prvkem také v podobě minipočítače RaspberryPi.

Všechny tyto zařízení jsou propojeny sběrníci Ethernet a napájeny ze zdroje o napětí 5V. Přibližné pořizovací ceny všech komponent jsou znázorněny v tabulce níže (viz *Tab. 1*). Výsledná cena celého zařízení je tedy přibližně **27.440 Kč**. Pokud se tato cena srovná s cenou projektu pi3Dscan 8627 € + 750 € licence na obslužný program / rok⁸ (**230.000 Kč + 20.000 Kč / rok**), tak je cena vyrobeného zařízení přibližně desetinná. Je ovšem nutné zdůraznit, že v inspirujícím projektu bylo použito 100 kusů kamer s obslužným minipočítačem. Výsledné skeny budou tedy jistě lepší. Z hlediska mechanické konstrukce disponuje vyrobené zařízení nastavitelnou pozicí kamer z hlediska vzdálenosti k objektu. Touto vlastností inspirující projekt nedisponuje. Na straně druhé je inspirující projekt lepší ve velikosti snímaného objektu, kde dosahuje nesrovnatelně větších rozměrů (porovnáván objekt o velikost člověka s objektem o největším rozměru 20cm v každém směru). Vyrobené zařízení je lépe zpracované po designové stránce. Pro vlastní posouzení je možné porovnat *Obr. 17* a *Obr. 22*.

Tab. 1: Výsledná cena vyrobeného zařízení

Zařízení	Množství	Cena za kus [Kč]	Cena celkem [Kč]
RaspberryPi 1 Model B Rev. 2.0	13	880 ⁹	11.440
RaspberryPi Camera Module v1	12	500 ¹⁰	6.000
Zdroj 5V, 200W	1	1.300 ¹¹	1.300
Síťový přepínač (switch)	1	700 ¹²	700
Mechanická konstrukce	1	5.000	5.000
Ostatní ¹³	1	3.000	3.000
Celkem			27.440

Pokud se cena srovná s komerčně dostupnými produkty (uvedeny níže), tak je až na jednu výjimku cena několikanásobně nižší. Dá se to tedy považovat za jednu z největších výhod této implementace. Bylo by hezké projekt ohodnotit jako jeden z nejlepších i z hlediska kvality výsledného modelu, ale realisticky je nutné uznat, že tomu tak není. Převážně je tento fakt dán největší nevýhodou tohoto řešení, a to neschopností skenovat objekt ze všech stran. Touto nevýhodou žádný komerčně dodávaný produkt netrpí. Není to ovšem slabina, která by nešla

⁸ Viz <http://www.pi3dscan.com/index.php/cost>

⁹ Cena přejata z <http://rpishop.cz/raspberry-pi-pocitace/8-raspberry-pi-0766897151323.html> ke dni 5.5.2017

¹⁰ K datu 5.5.2017 již kamera není dostupná, protože byla nahrazena verzí 2. Cena byla přejata z <https://www.pi-supply.com/product/raspberry-pi-camera-board-v1-3-5mp-1080p/>

¹¹ Cena přejata z <http://www.tme.eu/cz/details/hsp-200-5/spinane-zdroje-prumyslove/mean-well/> ke dni 5.5.2017

¹² Nejnižší cena přepínače se 16ti porty na internetovém obchodě CZC ke dni 5.5.2017

¹³ Další mechanické konstrukční prvky jako jsou kabely, šrouby atp.

napravit. Program je na to plně připraven a jedná se tak pouze o zvětšení mechanické konstrukce, kterou znemožňují pouze vyšší finanční nároky na projekt. Z hlediska kvality výsledného modelu je také velmi těžké konkurovat produktu, který stojí například půl milionu Kč. Jedná se tak o projekty s několikanásobně vyšším rozpočtem, a tedy vyšší kvalitou použitých zařízení nebo velikým vývojovým týmem, který zlepšuje rekonstrukční algoritmy. Další výhodou tohoto projektu je, že je možné si ho po přečtení této práce postavit vlastními silami. Ke komerčním produktům se takto detailní návod, pochopitelně, nepřikládá.

2.4.2 Komerční produkty

Obvykle je u komerčních 3D scannerů problematické dohledat cenu. Ceny se totiž pohybují relativně vysoko a poptávka po těchto zařízeních je velice specifická. Proto výrobci tají své ceny a sdělují je pouze na vyžádání. I přesto byly nalezeny informace o ceně následujících komerčních výrobků (obrázky umístěny v sekci *Přílohy*):

- Shapify – technické detaily neuvedeny, cena 39.000 \$¹⁴ (**948.400 Kč**), viz *Obr. 28*
- Dooblicator – technické detaily neuvedeny, cena 20.000 \$ až 50.000 \$¹⁵ (**486.000 Kč až 1.216.000 Kč**), viz *Obr. 29*
- Twinstant Mobile – rozlišení 0,7mm, cena 26.995 \$¹⁶ (**656.500 Kč**), viz *Obr. 30*

Všechny výše zmíněné scannery jsou rozměrově připravené na skenování lidí. Jedná se tedy o velice rozměrnou mechanickou konstrukci s velkým množstvím statických snímacích prvků (kamery nebo lasery) nebo s menším množstvím snímacích prvků, které kolem objektu rotují. Všechny tyto scannery mají prodlevou mezi vyfocením a získáním modelu objektu 15 minut. Tato doba je srovnatelná (v případě lepší konfigurace obslužného PC i vyšší – viz 2.6) s dobou rekonstrukce u vytvořeného zařízení.

Odlišnou kategorií jsou ruční scannery. Ty je možné využít pouze v případě statického objektu, který musíme tímto scannerem obejít ze všech stran. Tyto scannery nepotřebují žádnou robustní mechanickou konstrukci jako scannery uvedené v předchozím odstavci. V dnešní době již tyto scannery mají výhodu převážně v profesionálních aplikacích, kde je potřeba velice vysoké přesnosti. Pro (dnes velice populární) skenování lidí je možné využít

¹⁴ Viz <http://www.aniwaa.com/product/3d-scanners/artec-shapify-booth/>

¹⁵ Viz <http://www.aniwaa.com/product/3d-scanners/doob-dooblicator/>

¹⁶ Viz <http://www.aniwaa.com/product/3d-scanners/twindom-twinstant-mobile/>

mobilní telefon vyšší třídy s kvalitním fotoaparátem, kde princip snímání je totožný s ručním scannerem. Po nainstalování jedné z mnoha aplikací na skenování objektů (například 123DCatch) již stačí postupovat podle kroků v aplikaci. Největší výhodou ručních scannerů (nebo použití mobilní aplikace) je množství pořízených fotografií. Čím více fotografií je k dispozici, tím přesnější trojrozměrný model je možné vytvořit. Pokud tedy bude požadavkem velmi přesný model statického objektu, tak se fotografií vytvoří stovky a následně se z nich provede rekonstrukce. Tak vysoké množství snímků se obvykle ze scannerů v předchozím odstavci nezíská. I v této kategorii je ovšem vhodné uvést seznam zařízení a jejich velmi stručně shrnuté parametry:

- Artec Eva – rozlišení 0,5mm, vzdálenost od objektu 0,4m až 1m, cena 13.700 €¹⁷ (**366.400 Kč**) + licence na obslužný program nejméně 400 € (10.700 Kč) / rok, viz Obr. 31
- Artec Space Spider – rozlišení 0,1mm, vzdálenost od objektu 0,2m až 0,3m, cena 19.700 €¹⁷ (**526.900 Kč**) + licence na obslužný program nejméně 400 € (10.700 Kč) / rok, viz Obr. 31
- HandySCAN 700 – rozlišení 0,05mm, vzdálenost od objektu 0,3m až 0,55m, cena 50.000 \$ až 100.000 \$¹⁸ (**1.216.000 Kč až 2.431.000 Kč**), viz Obr. 32
- EinScan-Pro – rozlišení 0,24mm, cena 3990 €¹⁹ + kamera (pro vykreslení textury) 600 € (**106.700 Kč + 16.000 Kč**), viz Obr. 32
- Sense 3D scanner – rozlišení 0,9mm, vzdálenost od objektu 0,2m až 1,6m, cena 349 €²⁰ (**9.300 Kč**), viz Obr. 32

2.5 Návod na instalaci

Instalace uživatelské aplikace

Uživatelská aplikace se skládá z několika částí. Tou první je program popsáný v kapitole 2.3.3 a pro jeho zprovoznění stačí pouze vytvořit kopii adresáře *Programy/Uživatelská_aplikace* z příloženého paměťového média a spustit soubor *Strato.jar*. Pro úspěšné spuštění je nutné mít nainstalované JRE²¹ ve verzi 8 a vyšší. Druhou částí je zprovoznění programu VisualSFM, který zajišťuje samotnou rekonstrukci objektu z

¹⁷ Viz <https://www.artec3d.com/files/pdf/ArtecScanners-Booklet-EURO.pdf>

¹⁸ Viz <http://www.aniwaa.com/product/3d-scanners/creaform-handyscan-700/>

¹⁹ Viz <http://3dscanexpert.com/shining-3d-einscan-pro-3d-scanner-review/>

²⁰ Viz <https://www.3dsystems.com/shop/sense/techspecs>

²¹ Java Runtime Environment

fotografií. Pro jeho spuštění je nutné vytvořit kopii adresáře *Programy/VisualSFM* z příloženého paměťového média. Aktuální verzi je možné stáhnout ze zdroje [5], kde je i návod na zprovoznění. Pro rekonstrukci objektu využívá tento program nástroj CMVS, který je také přiložen ve zmíněné složce a byl převzat z [7]. Posledním krokem pro zprovoznění nástroje CMVS je přidání cesty k jeho adresáři do systémové cesty. Poté je program VisualSFM plně schopen využít tento nástroj.

Pro vizualizaci vygenerovaného trojrozměrného modelu je doporučen program MeshLab (viz [6]). Tento program umožňuje velké množství operací s trojrozměrným objektem od jednoduchého zobrazení a měření vzdáleností bodů až po komplexní práce s objekty jako je slučování nebo porovnání více objektů. Pro vizualizaci je možné využít i jiný program, avšak tento byl shledán jako nejméně náročný a uživatelsky velmi pohodlný. Vizualizaci zrekonstruovaného objektu je možné vidět v přílohách na *Obr. 34*.

Instalace programového vybavení scanneru

Instalaci softwaru scanneru je mnohem složitější než instalace uživatelské aplikace a lze ji rozložit do několika úkonů. V následujících odstavcích je popsáno, jak byla instalace řešena ve vytvořeném zařízení. Pro každý krok ovšem existuje alternativní řešení. Je tedy možné jednotlivé kroky přeskočit nebo je řešit jinak, ale není poté zaručena funkčnost.

Prvním krokem instalace softwaru scanneru je zprovoznění operačního systému Linux na všech prvcích. V případě použití minipočítače RaspberryPi byl použit systém Raspbian, který se umístí na SD kartu a ta se následně vloží do minipočítače. Po úspěšném nastartování systému je nutné povolit vzdálený přístup do systému pomocí *ssh*, povolit snímání kamerou a změnit standardní heslo. Všechny tyto úkony lze vykonat v grafickém rozhraní, které se vyvolá příkazem *raspi-config*. Všechny kroky od začátku tohoto odstavce je nutné zopakovat tolikrát, kolik máme k dispozici kamerových modulů a jednou navíc pro řídicí prvek.

Druhým důležitým krokem instalace je zprovoznění autentizace bez nutnosti zadávání hesla v rámci zařízení. Hlavní prvek potřebuje ke všem ostatním prvkům přistupovat a tento přístup se nesmí zastavit na zadávání hesla. Nejjednodušším a nejefektivnějším způsobem jak tohoto docílit je autentizace pomocí klíče. Na hlavním prvku je nejdříve nutné si vygenerovat onen klíč pomocí příkazu *ssh-keygen -t rsa*. Nyní je nutné na všech ostatních prvcích vytvořit složku *.ssh* v domovském adresáři uživatele (při standardní instalaci se jedná o uživatele *pi*).

Posledním krokem je distribuce vygenerovaného klíče (respektive jeho veřejné složky) na všechny prvky. Klíč umístěný v hlavním prvku na cestě `.ssh/id_rsa.pub` je možné odeslat do jednoho z prvků pomocí příkazu `cat .ssh/id_rsa.pub | ssh IP 'cat >> .ssh/authorized_keys'`, kde IP zastupuje IP adresu jednoho z prvků. Tento krok je nutné zopakovat pro všechny prvky ve scanneru.

Aby nebylo nutné opakovat každý krok pro všechny prvky ve scanneru zvlášť, tak je vhodné na hlavním prvku nainstalovat službu `dsh`, kde v konfiguračním souboru této služby je možné nastavit IP adresy všech prvků ve scanneru a následným zavoláním příkazu `dsh -a -c „uptime“` se nám zobrazí doba běhu všech zařízení ve scanneru. Sice tento nástroj není možné použít pro všechny úkony, ale zjednoduší to velkou část z nich.

Nevyhnutelným krokem je nastavení názvu prvků (tzv. `hostname`). Tento krok je velice důležitý pro pojmenovávání fotek. Je tedy nutné, aby každý prvek měl tento název unikátní. Nejjednodušším způsobem změny je provést tuto akci opět pomocí nástroje `raspi-config`. V tomto scanneru byla ovšem použita sofistikovanější metoda, která umožňuje velmi snadnou změnu. Požadovaný `hostname` byl společně s MAC adresou a IP adresou umístěn do DHCP serveru, který na základě shody MAC adresy zařízení s uloženou tabulkou nabídne zařízení IP adresu a `hostname`. Standardně ovšem zařízení nepřijímají `hostname` z DHCP, a proto bylo nutné upravit konfigurační soubory služby `dhclient` na všech prvcích, aby nabízený `hostname` přijala za svůj. Po těchto úpravách stačí pouze v DHCP upravit jeden soubor a všechny zařízení si aktualizují svůj `hostname`.

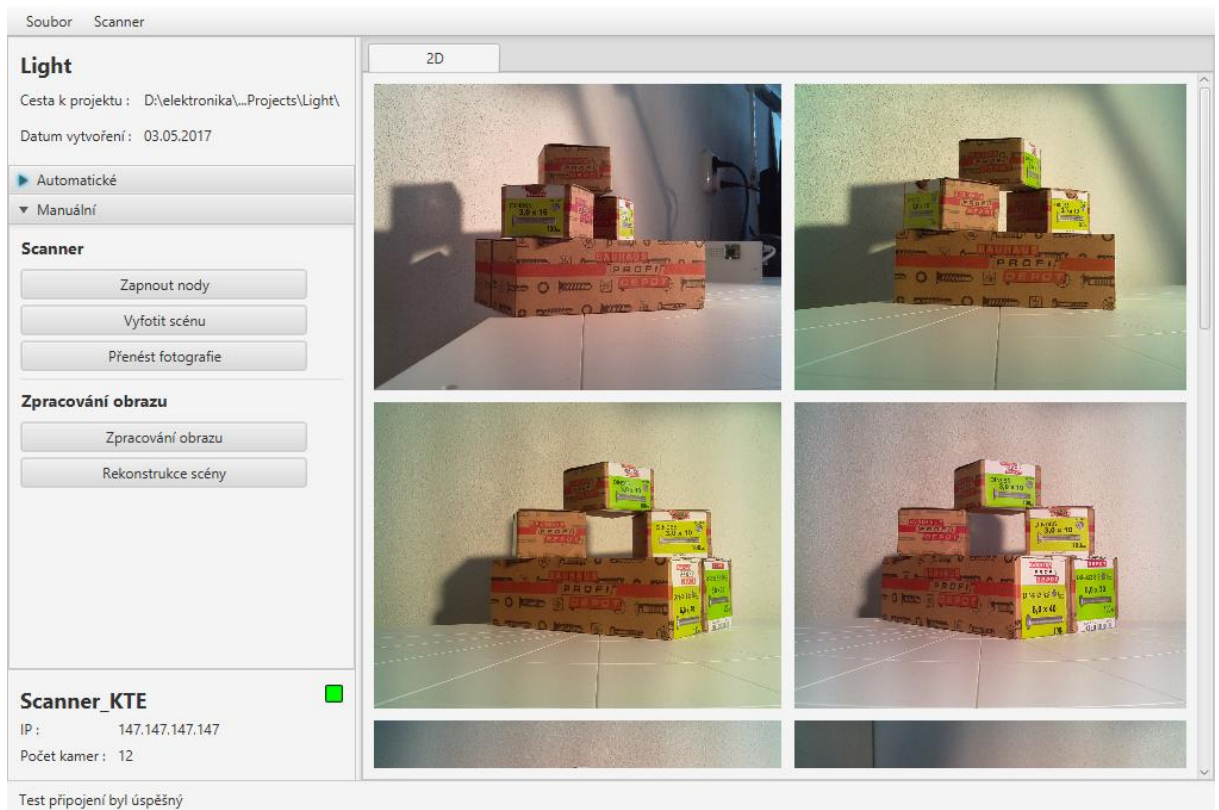
Pro sdílení dat mezi prvky byl využit systém `samba`. Instalace probíhá ve dvou krocích. Nejprve je nutné nainstalovat a nastavit část na hlavním prvku (tzv. `samba server`). Instalace balíku `samba` probíhá ze standardních repositářů systému Linux. Dále je nutné vytvořit systémového uživatele (například takto: `sudo adduser --no-create-home scanner-share`), nastavit mu heslo pomocí `sudo smbpasswd -a scanner-share` a konfigurační soubory rozšířit o část umístěnou v přílohách na straně *i*. Z konfiguračního souboru je patrné, že se sdílí adresář `/mnt/data`. Je tedy nutné ho vytvořit a nastavit vhodná oprávnění pro přístup uživatelů. Druhým krokem je zprovoznění tohoto systému na všech ostatních prvcích scanneru. Samotná instalace probíhá pomocí `dsh` a to příkazem `dsh -a -c „sudo apt-get install smbclient -y“`. Konfigurace se provádí pomocí konfiguračního souboru umístěného v `/etc/samba/credentials`. Ukázkový soubor je také v přílohách na straně *i*. Poté je nutné na všech prvcích vytvoření

adresáře `/mnt/data` s vhodnými přístupovými oprávněními. Pro automatické připojení klientů na server po spuštění operačního systému je nutné umístit řádek `//IP/scanner-share /mnt/data cifs uid=1000,rw,credentials=/etc/samba/credentials 0 0` (IP je nutné nahradit IP adresou serveru) do souboru `/etc/fstab` a změnit oprávnění k souboru `/etc/samba/credentials` na 600 (hodnota je v osmičkové soustavě).

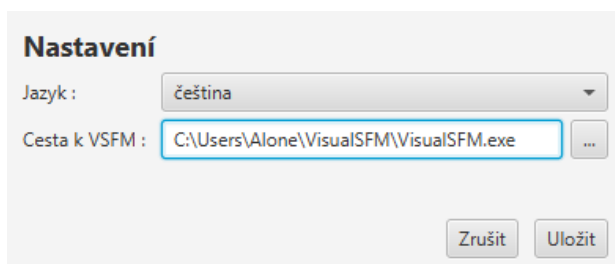
Posledním důležitým krokem je instalace JRE ve verzi 8 a vyšší na všech prvcích. S použitím služby `dsh` se opět jedná o velice jednoduchý příkaz: `dsh -a -c „sudo apt-get install openjdk-8-jre -y“`.

2.6 Návod na používání

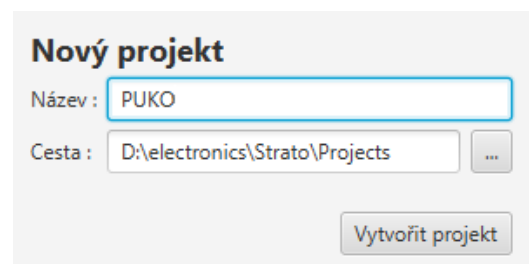
Prvním krokem je spuštění řídicího programu na hlavním prvku scanneru (viz 2.3.2). Lze toho docílit příkazem `java -jar StratoScanner.jar` z adresáře, kde je umístěn tento soubor včetně konfiguračního souboru. Tento program si již sám zapne všechny ostatní prvky ve scanneru a zajistí komunikaci s uživatelským rozhraním. Pro jednoduchost používání je možné tuto aplikaci nechat pouštět automaticky po startu operačního systému nebo jako službu ve virtuálním terminálu (například pomocí aplikace `screen`), který zůstává aktivní i po odhlášení uživatele.



Obr. 23: Vzhled grafického rozhraní aplikace Strato



Obr. 24: Nastavení aplikace Strato



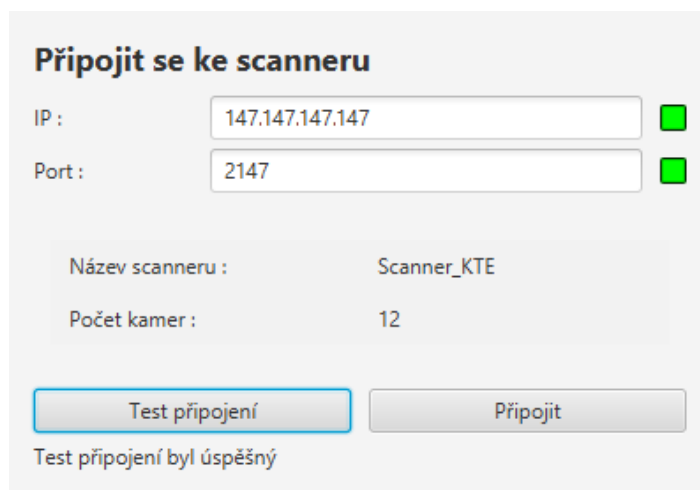
Obr. 25: Nový projekt v aplikaci Strato

Pokud řídicí program již běží, tak je možné spustit uživatelskou aplikaci na uživatelském počítači. Vzhled aplikace je znázorněn na Obr. 23. Po prvním spuštění je nutné v menu vybrat položku *Soubor* a zvolit *Nastavení*. V nově otevřeném okně (viz Obr. 24) se musí nastavit umístění programu VisualSFM. Je zde také možné vybrat jazykovou verzi aplikace (čeština nebo angličtina). Zbytek návodu na používání se odkazuje na českou verzi. Obvyklý způsob používání začíná vytvořením nového projektu tím, že se v menu klikne na položku *Soubor* a vybere se *Nový projekt*. V reakci na to se otevře nové okno (viz Obr. 25), kde se vybere umístění projektu a jeho název. Po vytvoření projektu je možné se připojit na scanner. Toho lze docílit kliknutím v menu na *Scanner* a vybrat položku *Nastavení*. V nově otevřeném okně (viz Obr. 26) se vybere IP adresa a port pro komunikaci se scannerem. Spojení je možné

otestovat tlačítkem *Test* a v okamžiku, kdy spojení úspěšně projde testem je možné toto spojení otevřít ke komunikaci. Pokud je spojení navázáno, tak je možné provést rekonstrukci objektu dvěma způsoby – automaticky nebo manuálně. Režim je možné si zvolit v levé postranní liště hlavního okna. V automatickém režimu stačí kliknout na tlačítko *Skenovat* a aplikace se o zbytek úkonů postará. V režimu manuálním je možné jednotlivé kroky rekonstrukce ovládat zvlášť. Je zde tedy několik tlačítek, které ovládají různé úkony a zde jsou v rychlosti vysvětleny:

- *Zapnout nody* – Zapne program pro skenování na všech prvcích.
- *Vyfotit scénu* – Pořídí snímky objektu ze všech kamer.
- *Přenést fotografie* – Realizuje přenos fotografií ze scanneru do složky projektu.
- *Zpracování obrazu* – Najde společné body snímků a zjistí vzájemnou pozici kamer.
- *Rekonstrukce scény* – Rekonstrukce modelu ze společných bodů.

První 3 kroky trvají relativně krátkou dobu (do minuty). Body realizující rekonstrukci jsou časově náročnější a mohou trvat až 10 minut v závislosti na velikosti výpočetního výkonu a množství fotografií. Rekonstrukce objektu vytvoří ve složce projektu soubor s příponou *ply*, který je možné si zobrazit například pomocí, již zmíněného, programu MeshLab.



Připojit se ke scanneru

IP : 147.147.147.147

Port : 2147

Název scanneru : Scanner_KTE

Počet kamer : 12

Test připojení byl úspěšný

Obr. 26: Obrazovka nastavení scanneru aplikace Strato

3 Další možná vylepšení

Tato sekce se zabývá možnými vylepšeními a rozšířeními současné práce. Většina z níže zmíněných podkapitol byla v průběhu realizace celého projektu intenzivně projednávána a plánována jako sekundární cíle projektu. Některé ze sekundárních cílů byly do práce zapracovány, ale na tyto již nebyl dostatek času a otevírá to tedy možnost pokračování práce na tomto projektu v rámci dalších studentských prací.

Sekce 3.2 a 3.3 by do projektu vnesly obrovskou výhodu v možném ovládní scanneru včetně procesu rekonstrukce vzdáleně z mobilních zařízení. Naneštěstí velikost užítku je přímo úměrná množství stráveného času nad touto problematikou.

3.1 Rozšíření mechanické konstrukce

Nejjednodušším, avšak finančně nejnákladnějším, vylepšením je rozšíření mechanické konstrukce na možnost skenování objektu ze všech stran. V současné konfiguraci umístění kamer by to znamenalo přidání alespoň 15ti kamerových modulů (kamera + RaspberryPi).

Z hlediska mechanické konstrukce je možné konstrukci zvětšit pro možnost skenování objektů o velikosti dospělého člověka. Primárním cílem projektu bylo ovšem skenovat objekty do maximálního rozměru 20cm v každé z os ortogonální soustavy souřadné.

3.2 Rekonstrukce objektu scannerem

V současné době se provádí rekonstrukce objektu na uživatelském počítači. Tento způsob má výhodu v tom, že pokud má uživatel dostatečně výkonný počítač, tak je možné dosáhnout relativně příjemné rychlosti odezvy. V případě nízkého výkonu a vysokého počtu fotografií se může čas výpočtu značně prodloužit. Zajímavým způsobem rozšíření současného projektu by bylo zprovoznit rekonstrukci přímo na minipočítačích ve scanneru, které realizují pořízení fotografií.

3.3 Rekonstrukce objektu externě

Externí službou je v tomto případě míněn vzdálený cluster, na kterém se po pořízení fotografií spustí paralelní rekonstrukce bodů objektu. Díky vysoké úrovni paralelizaci a předpokládanému vysokému výkonu clusteru je možné zrychlit proces rekonstrukce.

Pro tento účel byl vytvořen cluster vytvořený z 32 kusů RaspberryPi (viz Obr. 27). Byl vytvořen mou osobou, vedoucím projektu – Ing. Petrem Kropíkem, Ph.D. a Katedrou teoretické elektrotechniky Fakulty elektrotechnické. Účelem tohoto clusteru není výkonově překonat jakýkoliv osobní počítač na světě. Na tomto clusteru byly ověřeny základní teorie paralelního programování (například Amdahlův zákon). Největším důvodem realizace tohoto clusteru bylo, dát studentům možnost otestovat si vlastní algoritmy pro paralelní výpočty, aniž by byl vytěžován (a z bezpečnostního hlediska ohrožen) cluster, který je určen pro výzkumné účely. Dalším z primárních cílů bylo, dát studentům volný přístup ke všem prostředkům v rámci clusteru. Mohou si tedy „sáhnout“ (z programového hlediska) až na nejnížší úroveň operačního systému a zapříčinit tím pád celého operačního systému. Nikomu tím ovšem nezpůsobí žádnou škodu a ze své chyby se mají šanci poučit, aby se jim neopakovala v produkčních verzích jejich prací.

Pro účel této práce byl sice vytvořen malý výpočetní cluster, avšak do výsledné práce nebyl použit kvůli vysoké časové náročnosti samotné práce na scanneru. Do budoucna se ovšem tato možnost propojení nabízí v rámci samostatné diplomové práce.



Obr. 27: RPi cluster

3.4 Rekonstrukce video záznamu ve 3D

Jednou ze zajímavých aplikací scanneru by mohla být i realizace trojrozměrného video záznamu, ve kterém by bylo možné pohybovat si s pozicí pozorovatele. Základní předpoklad, kterým je přesná synchronizace snímků (viz 2.3.2), byl pro tuto aplikaci splněn, nýbrž

standardní snímkovací frekvence při snímání videozáznamů je 24 snímků za sekundu. Z toho vyplývá, že přibližně každých 42ms vznikne nový snímek, který by měl být vyfocen s přesností na jednotky milisekund. Tohoto cíle bylo v této práci dosaženo a je možné se tedy zabývat pouze zpracováním získaných dat a jejich následnou vizualizací. Zpracování dat bude převyšovat náročnost současných požadavků a bylo by tedy vhodné k tomu využít systém popsáný v 3.3.

3.5 Napojení na 3D tiskárnu

Zajímavé použití scanneru je vytvoření modelu objektu a jeho následné odeslání do 3D tiskárny. Tento krok by vyžadoval možnost snadné úpravy velikosti modelu a jeho export do podporovaných formátů (například *stl*). O samotný tisk by se následně postaral obslužný program tiskárny. Jednou z nabízených možností je projekt OctoPrint (viz [18]), který umožňuje vzdálené ovládání tiskárny. Pokud by byla požadována složitější funkcionalita, tak je možné tento tiskový systém rozšířit o různé moduly.

4 Závěr

Tato práce se zabývá velice perspektivním oborem, kterým je rozpoznání obrazu a vypracování trojrozměrného modelu objektu. V první části jsou představeny matematické postupy, pomocí kterých je možné proměnit fotografie v trojrozměrný model. Na tento teoretický úvod navazuje popis praktické realizace, ve které jsou vysvětleny postupy, pomocí kterých byla vytvořena mechanická konstrukce a obslužné programové vybavení výsledného zařízení. Práci uzavírá kapitola s možnými vylepšeními vyrobeného zařízení.

Hlavním cílem práce bylo vytvořit zařízení, které bude schopné rekonstruovat modely objektů o největších rozměrech 20 cm ve všech směrech. Pro otestování nápadů byla vytvořena prototypová konstrukce, na které byly vyzkoušeny jak mechanické, tak programové postupy. Získané zkušenosti byly využity při stavbě výsledného zařízení.

Zařízení splňuje veškeré požadavky, které na něj byly kladeny, s výjimkou jednoho. Z důvodu nedostatku kamerových modulů je sken objektu realizován pouze z jedné poloviny. Výsledný model neobsahuje celé zařízení, ale jen jeho jednu polovinu. Tento problém je ovšem způsoben pouze nedostatečnými finančními prostředky. Veškeré vytvořené programy jsou navrženy pro práci s větším množstvím kamer a rozšíření mechanické konstrukce také není neřešitelným problémem. Výsledná cena celého zařízení je několikanásobně nižší než u komerčně dostupných produktů a výsledné skeny disponují požadovanou kvalitou.

Seznam literatury a informačních zdrojů

- [1] NTP [online]. [Cit 29.4.2017]. Dostupné z: <http://www.ntp.org/ntpfaq/NTP-s-algo.htm>
- [2] Reference Broadcast Synchronization. [online]. [Cit 23.4.2017]. Dostupné z: https://www.cse.wustl.edu/~jain/cse574-06/ftp/time_sync/index.html#Section2.0
- [3] Odkaz na fórum k nastavení kamer RaspberryPi Camera v1.3. [online]. Poslední změna 06/2013. [Cit 23.4.2017]. Dostupné z: <https://www.raspberrypi.org/forums/viewtopic.php?t=46637&p=366719>
- [4] Depth of Field. [online]. [Cit 1.5.2017]. Dostupné z: <http://www.dofmaster.com/equations.html>
- [5] VisualSFM [online]. [Cit 4.5.2017]. Dostupné z: <http://ccwu.me/vsfm/>
- [6] MeshLab [online]. [Cit 4.5.2017]. Dostupné z: <http://www.meshlab.net/>
- [7] CMVS [online]. [Cit 4.5.2017]. Dostupné z: <http://www.di.ens.fr/cmvs/>
- [8] Pitzer, Benjamin. Automatic Reconstruction of Textured 3D Models. Karlsruhe: Dissertation KIT, 2011
- [9] Turkel E., Hough Transform [online]. [Cit. 6.5.2017]. Dostupné z: <http://www.math.tau.ac.il/~turkel/notes/HoughTransform.pdf>
- [10] Hledání parametrického popisu pomocí Houghovy Transformace [online], Poslední změna 24.10. 2008, [Cit. 6.5.2017], Dostupné z: <http://cmp.felk.cvut.cz/cmp/courses/ZS1/Cviceni/cv5/hough.htm>
- [11] Ballard D.H., Generalizing the Hough Transform to detect arbitrary shapes [online], 1980, [Cit. 6.5.2017], Dostupné z: <http://www.sci.utah.edu/~gerig/CS7960-S2010/handouts/Ballard-Generalized-HoughT.pdf>
- [12] Fessl, Lukáš, Detekce vzorů v obrazech a užití pro analýzu kulturního dědictví [online], 2016, [Cit. 6.5.2017], Dostupné z: <https://theses.cz/id/jc79y8/DP2016.pdf>
- [13] Lowe, David G. , Object recognition from Local Scale-Invariant Features [online], 1999, [Cit. 7.5.2017], Dostupné z: <http://www.cs.ubc.ca/~lowe/papers/iccv99.pdf>
- [14] Lowe, David G., Distinctive Image Features from Scale-Invariant Keypoints [online], 2004, [Cit. 7.5.2017], Dostupné z: <http://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf>
- [15] Bay, Herbert, SURF: Speeded Up Robust Features [online], 2006, [Cit. 7.5.2017], Dostupné z: <http://www.vision.ee.ethz.ch/~surf/eccv06.pdf>
- [16] Mikolajczyk, Krystian, Scale & Affine Invariant Interest Point Detectors [online], 2004, [Cit 8.5.2017], Dostupné z: https://www.robots.ox.ac.uk/~vgg/research/affine/det_eval_files/mikolajczyk_ijcv2004.pdf
- [17] Bílek, Petr, Významné body v obraze: detekce, korespondence a lokalizace ve 3D [online], 2007, [Cit. 8.5.2017], Dostupné z: https://dip.felk.cvut.cz/browse/pdfcache/bilekp3_2007bach.pdf
- [18] OctoPrint [online]. [Cit. 9.5.2017]. Dostupné z: <http://octoprint.org/>
- [19] Structure from Motion by MathWorks [online]. [Cit. 10.5.2017]. Dostupné z: mathworks.com/help/vision/examples/structure-from-motion-from-two-views.html

- [20] Xiaowei Li, Changchang Wu, Christopher Zach, Svetlana Lazebnik and Jan-Michael Frahm. Modeling and Recognition of Landmark Image Collections Using Iconic Scene Graphs [online]. [Cit 10.5.2017] Dostupné z: <http://www.cs.unc.edu/~jmf/1436/>
- [21] Kyrki, Ville. Robotic Vision course lecture 9 [online]. [Cit 10.5.2017] Dostupné z: https://mycourses.aalto.fi/pluginfile.php/191787/mod_resource/content/3/lecture9.pdf
- [22] Říha, Kamil. Epipolární geometrie [online]. [Cit 10.5.2017]. Dostupné z: <http://www.elektrorevue.cz/clanky/05017/index.html>

Přílohy

Rozšíření konfiguračního souboru služby samba na serveru

```
[scanner-share]
comment = Scanner shared directory
path = /mnt/data
valid-users = scanner-share
writable = yes
locking = no
hosts allow = 172.16.2.*
hosts deny = 0.0.0.0/0
```

Konfigurační soubor služby samba na klientských stanicích

```
username=scanner-share
password=XXX
domain=//IP/scanner-share
```

XXX nahraďte nastaveným heslem a IP nahraďte IP adresou zařízení, kde je spuštěn samba server



Obr. 28: Artec Shapify 3D scanner
(zdroj: <http://www.aniwaa.com/wp-content/uploads/2015/07/3D-scanner-Artec-Shapify-booth-man.jpg>)



Obr. 29: Dooblicator 3D scanner
(zdroj: http://www.doobgroup.com/fileadmin/user_upload/Bilder_Content/dooblicator_dome_web.jpg)



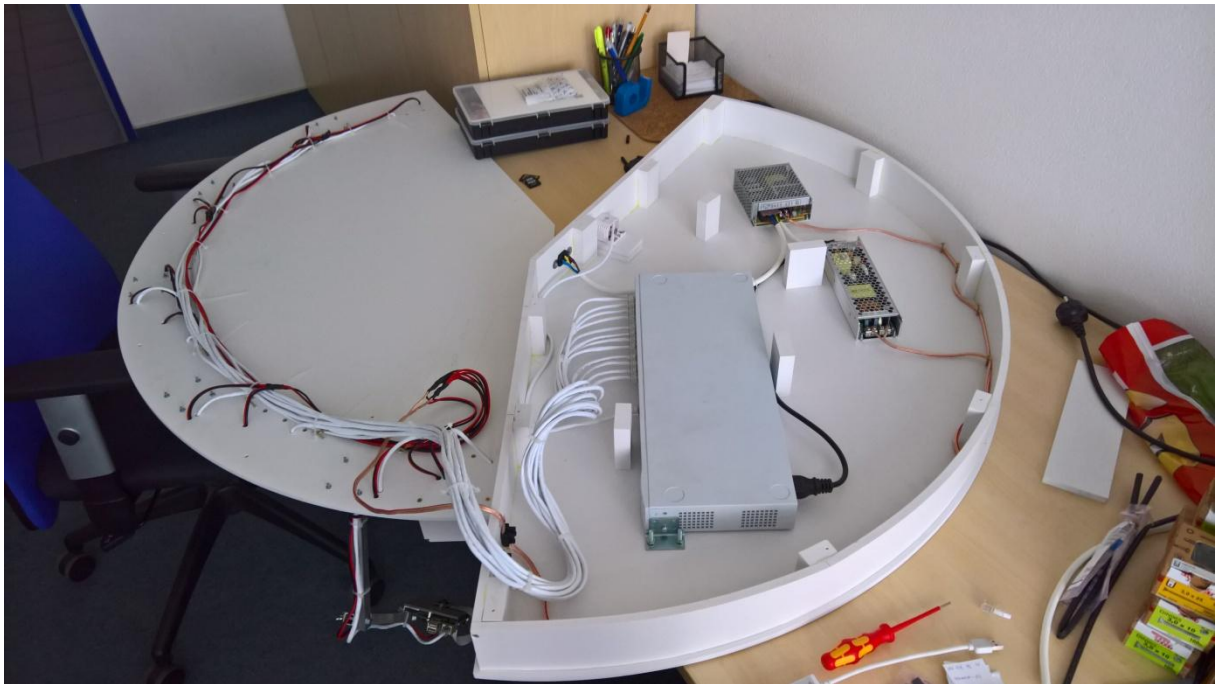
Obr. 30: Twinstant Mobile 3D scanner
(zdroj: <http://www.aniwaa.com/wp-content/uploads/2016/05/3D-scanner-Twindom-Twinstant-mobile.jpg>)



Obr. 31: Artec Eva (vlevo) a Artec Space Spider (vpravo)
(zdroj: <https://artec3d-production.s3-eu-west-1.amazonaws.com/products/artec-eva.png>
a <https://www.3dz.it/wordpress/wp-content/uploads/2016/01/main-stampanti-3d-scanner3d-3dz-artec-space-spider.png>)



Obr. 32: HandySCAN 700 (vlevo) a EinSCAN-Pro (uprostřed) a Sense 3D scanner
(zdroj: <http://www.aniwaa.com/wp-content/uploads/2014/05/3D-scanner-creaform-handyscan-700-perspective.jpg>,
<http://www.aniwaa.com/wp-content/uploads/2016/12/3D-scanner-Shining-3D-EinScan-Pro-Plus-perspective.jpg>
a <https://static.webshopapp.com/shops/024440/files/065316598/3d-systems-cubify-sense-rs-3d-scanner.jpg>)



Obr. 33: Rozložené výsledné zařízení. V pravé části je podkladová konstrukce a v části levé krycí díl s kamerami.



Obr. 34: Ukázka jednoho z naskenovaných objektů pomocí vytvořeného zařízení v programu Meshlab