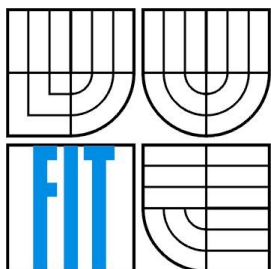


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ODHAD GEOMETRIE MÍSTNOSTI POMOCÍ KINECTU

3D MODEL OF A ROOM USING KINECT

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MARTIN ZEMEK

VEDOUCÍ PRÁCE
SUPERVISOR

ING. MICHAL ŠPANĚL, PH.D.

BRNO 2016

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií

Akademický rok 2015/2016

Zadání bakalářské práce

Řešitel: **Zemek Martin**

Obor: Informační technologie

Téma: **Odhad geometrie místnosti pomocí Kinectu**

3D Model of a Room Using Kinect

Kategorie: Zpracování obrazu

Pokyny:

1. Prostudujte základy počítačového vidění. Zaměřte se na problematiku detekce rovinných ploch v tzv. mračnecích bodů, která produkuje zařízení typu MS Kinect.
2. Seznamte se s existujícími metodami a nástroji pro spojení (těž registraci) sekvence mračen bodů.
3. Vyberte vhodné metody a navrhnete nástroj pro odhad základní geometrie a rozměrů místnosti na základě dat pořízených Kinectem.
4. Experimentujte s vaší implementací a případně navrhnete vlastní modifikace metod.
5. Porovnejte dosažené výsledky a diskutujte možnosti budoucího vývoje.
6. Vytvořte video nebo plakát prezentující vaši bakalářskou práci, její cíle a výsledky.

Literatura:

- Dle pokynů vedoucího.

Pro udělení zápočtu za první semestr je požadováno:

- Splnění prvních tří bodů zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

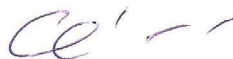
Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Španěl Michal, Ing., Ph.D.**, UPGM FIT VUT

Datum zadání: 1. listopadu 2015

Datum odevzdání: 18. května 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
642 06 Brno, Božetěchova 2



doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Tato práce se zabývá nalezením významných rovinných ploch v mračně bodů a jejich převodem na polygony, což je důležitý krok pro odhad geometrie místnosti. Vstupem je mračno bodů, které bylo získáno pomocí senzoru Kinect v2. Součástí práce je program, který dokáže zachytit jeden snímek z Kinectu. Pro detailnější mračna bodů je potřeba použít některý z externích programů, které jsou zmíněny dále v této práci. Pro detekci rovin je použit algoritmus RANSAC. Roviny se pomocí algoritmu extrakce Euklidovského shlukování rozdělí na menší plochy. Tyto plochy se následně pomocí konvexní nebo konkávní obálky převedou na polygony.

Výsledná aplikace je schopná zpracovat jak jeden snímek nebo větší mračno získané registrací dílčích snímků a detekovat hlavní i vedlejší roviny. Pro největší rovinu dokáže nachystat body k vytvoření textury a vypsát rozměry do příkazové řádky.

Abstract

This thesis is about finding significant planar surfaces in point cloud and their conversion to polygons. Which is important step for making a 3D model of a room. Input is point cloud, which was recorded by Kinect v2 sensor. Tool for capturing one snapshot from Kinect is included in this thesis. For recording more detailed point cloud is needed external program. Some of the programs are mentioned further in this thesis. For plane detection is used RANSAC. Inliers are divided using Euclidean Cluster Extraction. These clusters are converted to polygon using convex or concave hull.

Application is capable of working with one snapshot or bigger point cloud assembled by registration of particular snapshots and detect primary and secondary planar surfaces. For the largest plane points can be prepared for creation of a texture and dimensions of this plan can be printed in CLI.

Klíčová slova

mračno bodů, detekce roviny, počítačové vidění, PCL, Kinect v2, Kinect Fusion

Keywords

point cloud, plane detection, computer vision, PCL, Kinect v2, Kinect Fusion

Citace

ZEMEK, Martin. *Odhad geometrie místnosti pomocí Kinectu*. Brno, 2016. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Španěl Michal.

Odhad geometrie místnosti pomocí Kinectu

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Michala Španěla Ph.D.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

18.05.2016

.....
Martin Zemek

Poděkování

Rád bych poděkoval svému vedoucímu bakalářské práce Ing. Michalu Španělovi Ph.D. za rady k vypracování práce a pomoc při řešení problémů, které se vyskytly při tvorbě práce.

© Martin Zemek, 2016

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů..

Obsah

1	Úvod	2
2	Pořizování a zpracování dat ve formě 3D mračna bodů	3
2.1	RGB-D kamery a MS Kinect.....	3
2.2	Pořizování 3D mračna bodů pomocí MS Kinect v2	6
2.3	Metody registrace mračna bodů.....	6
3	Existující nástroje pro pořízení a zpracování mračna bodů	8
3.1	Nástroje pro registraci mračna bodů.....	8
3.2	Knihovna Point Cloud Library (PCL)	10
3.3	Nástroje pro úpravu mračna bodů.....	11
4	Návrh metody pro detekci rovinných ploch v místnosti a jejich převod na polygony.....	12
4.1	Snímání mračna bodů	12
4.2	Nalezení roviny.....	13
4.3	Rozdělení roviny na plochy	13
4.4	Projekce bodů do 2D a vytvoření konvexní nebo konkávní obálky	14
4.5	Vytvoření textury.....	15
4.6	Export modelu	15
5	Implementace.....	16
5.1	Snímání mračna bodů	16
5.2	Filtrace mračna bodů	18
5.3	Nalezení roviny.....	18
5.4	Rozdělení roviny na plochy	18
5.5	Textura.....	19
5.6	Další možnosti aplikace 3DModel.....	19
6	Výsledky	20
6.1	Jednoduché scény	20
6.2	Složitější scény s objekty.....	20
7	Možnosti další práce	23
7.1	Voxelizace mračna bodů	23
7.2	SLAM optimalizovaný pro zaznamenávání barvy	23
7.3	Automatizace hledání rovin.....	23
7.4	Export pro snadné použití v programech pro 3D grafiku	24
8	Závěr	25

1 Úvod

Už léta lidé tvoří počítačové 3D modely svého okolí. Ať už při využití pro vizualizaci změn prostředí, počítačových her a různé další účely. Vždy je k vytvoření modelu potřeba 3D grafik. Pokud chcete mít detailní model místnosti, je potřeba udělat fotky, případně i měření, interiéru. V dnešní době je snaha vše automatizovat. Od automatického řízení aut po doručení zásilek létajícím dronem. Konkrétně tyto dvě zmíněné oblasti využívají počítačové vidění, aby si auta nebo droni mohli vytvořit představu, co je kolem nich. Cílem této práce je částečně zautomatizovat proces tvoření modelu místnosti.

Pro záznam okolí se obecně používá LiDAR nebo RGB-D kamery, obě technologie mají společné, že výstup je mračno bodů. Zatímco LiDAR zařízení jsou poměrně drahé^{1,2} a používají je spíše u větších společností. RGBD kamery jsou už rozšířeny v domácnostech v podobě zařízení Microsoft Kinect.

Ze zařízení MS Kinect můžeme získat mračno bodů, tedy body rozmístěné v prostoru. Tyto body mohou obsahovat také údaje o barvě. Z rozmístění bodů můžeme zjistit geometrii místnosti, a pokud body obsahují barvu, můžeme z nich vytvořit texturu.

Pomocí algoritmu RANSAC se najde rovina, ve které leží nejvíce bodů. Tato rovina se pomocí algoritmu extrakce Euklidovského shlukování rozdělí na menší plochy. Tyto plochy se následně pomocí konvexní obálky převedou na polygony.

Výsledná aplikace dokáže najít stěny místnosti i rovinné plochy objektů v ní. Nalezené plochy jsou převedeny na polygony u kterých známe relativní umístění, velikost i orientaci v prostoru. V práci je také algoritmus pro nachystání bodů plochy pro tvorbu textury. Lze vypsát rozměr největší plochy v místnosti.

Tato práce je rozdělená do osmi kapitol. V kapitole 2 se popisuje problematika pořizování a zpracování mračna bodů. Jsou zde zmíněny senzory používané ke snímání mračna bodů, popis jak funguje Kinect první a druhé verze. Dále je zde zmíněn princip získání dat z Kinectu v2. V kapitole 3 jsou zmíněny existující nástroje pro pořizování a zpracování mračna bodů. V kapitole 4 je popsáno, jak by měla aplikace fungovat. V kapitole 5 je popsána implementace jednotlivých částí zmíněných v předchozí kapitole. Následují výsledky, kde lze názorně vidět, jak se aplikace chová. A na konci se nachází několik nápadů, jak by se mohla aplikace dále rozvíjet.

¹ Velodyne Lidar Puck – 7999\$. <https://www.wired.com/2014/09/velodyne-lidar-self-driving-cars/>

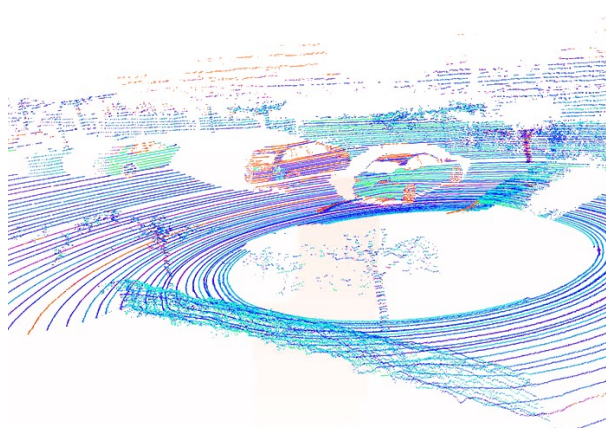
² Na trhu se však začínají objevovat i výrazně levnější a jednodušší modely (250\$).
Viz. <https://www.kickstarter.com/projects/scanse/sweep-scanning-lidar/description>

2 Pořizování a zpracování dat ve formě 3D mračna bodů

Mračno bodů je skupina bodů rozmístěných v souřadnicovém systému. Jednotlivé skupiny bodů mohou reprezentovat reálné objekty. Mračna bodů se často využívají k vytvoření mapy reálného prostředí.

Pro získání jednotlivých bodů v prostoru je potřeba znát pozici senzoru, vzdálenost od snímaného bodu a směr, ve kterém je bod snímán. Pomocí těchto vlastností jde spočítat, kde se body nachází v prostoru. Polohu senzoru si při vytváření nového mračna bodů můžeme zvolit libovolně, standardně se používá počátek souřadnicového systému. Pokud chceme snímat větší mračno bodů z více pozic, je potřeba nejprve senzor lokalizovat. Pro měření vzdálenosti se obecně používá laserový radar LiDAR nebo RGB-D kamery.

Zařízení LiDAR jsou populární v průmyslu pro svoji přesnost a vzdálenost, kterou jsou schopny změřit. K měření vzdálenosti používají osvětlení laserem a podle doby letu světla k měřenému bodu a zpět se spočítá vzdálenost. Laser často rotuje a pokrývá 360° okolí v jedné ose. Existují i senzory, které měří 64 takových os. Takový senzor může změřit až 2.2 milionu bodů za sekundu s přesností pod 2 cm na vzdálenost 120 metrů. (VELODYNE LIDAR, INC., 2016)



Obrázek 1 – Záznam ze senzoru Velodyne LiDAR HDL-64E³

2.1 RGB-D kamery a MS Kinect

RGB-D kamery oproti senzorům LiDAR zaznamenávají hloubku a navíc i barvu bodů. Obraz zaznamenávají do 2D matice.

Kromě nejznámější RGB-D kamery od Microsoftu, Kinectu, se začínají rozšiřovat kamery RealSense od Intelu. Pro RealSense se používají sestavy kamer nebo infračervené projekce. V podobě dvou kamer se Intel RealSense dostal i do komerčních tabletů.⁴ V následujících měsících lze s RealSense očekávat oznámený smartphone od Lenova a vývojářský smartphone od Intelu. Intel také pracuje na samostatných kamerách, které používají infračervenou projekci. Podobně jako tomu je u prvního Kinectu.

³ Obrázek pochází z <http://velodynelidar.com/hdl-64e.html> a byl upraven pro lepší tisk.

⁴ <http://www.dell.com/us/p/dell-venue-8-7840-tablet/pd>

Tato zařízení mají výhodu oproti Kinectu, protože mají senzory na snímání pohybu. Tyto senzory se hodí při registraci mračka bodů pomocí *SLAMu*. V neposlední řadě jsou bezdrátová. Kinect v2 potřebuje připojení k počítači i do elektrické sítě, to umožňuje snadnější manipulaci a pohyb.

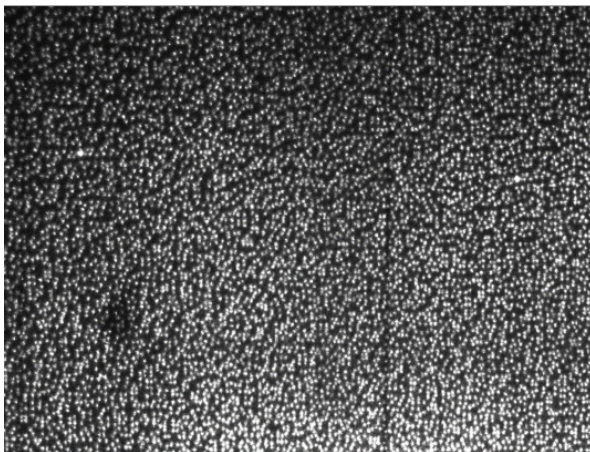
Společnost Google se svým projektem Tango vede vývoj v tomto odvětví a lze tedy očekávat větší rozšíření *RGB-D* kamer v rámci smartphone a tabletů s cenou v řádech stovek amerických dolarů.



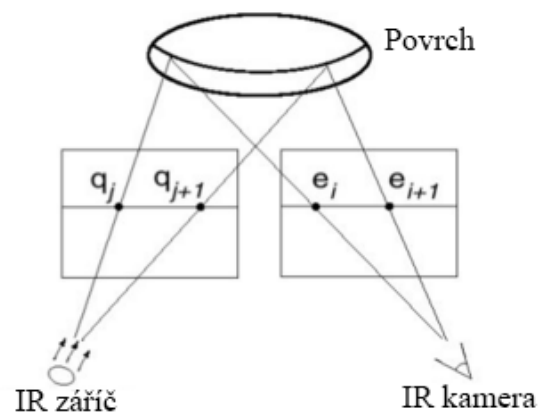
Obrázek 2 – Intel® RealSense™ Smartphone Developer Kit⁵

MS Kinect v1

Microsoft Kinect pro Xbox 360 snímá hloubku pomocí infračervené projekce pseudonáhodně rozmístěných bodů (Obrázek 3). Infračervená kamera, která je umístěna o několik centimetrů vedle zářiče, snímá promítnuté body. Kinect má 3 typy promítnutých bodů pro různé vzdálenosti. V bodech hledá stejné skupiny bodů, jako promítané. Protože stejnou skupinu bodů vidí z jiného úhlu a známe vzdálenost mezi zářičem a kamerou, můžeme spočítat vzdálenost od kamery.



Obrázek 3 – Pseudonáhodně promítnuté body Kinectem.



Obrázek 4 – Triangulace promítnutých bodů⁶

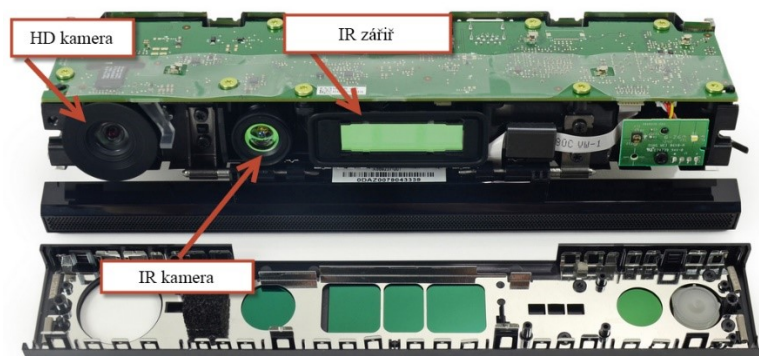
⁵ Převzato z:

<https://software.intel.com/en-us/blogs/2016/02/22/introducing-intel-realsense-smartphone-developer-kit>

⁶ Převzato z:

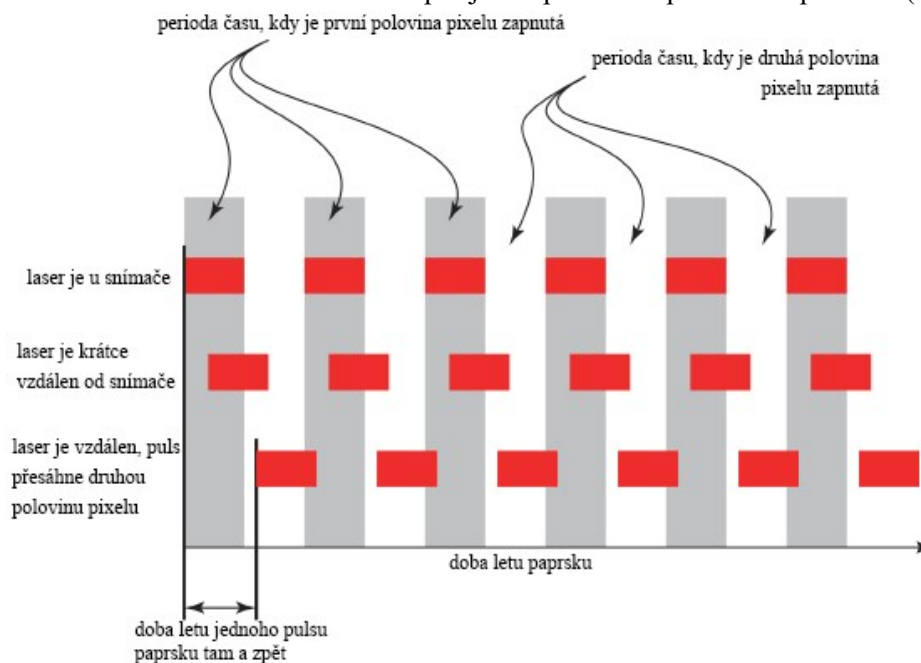
http://campar.in.tum.de/twiki/pub/Chair/TeachingSs11Kinect/2011-DSensors_LabCourse_Kinect.pdf

MS Kinect v2



Obrázek 5 – Rozebraný Kinect v2⁷

Novější Kinect pro Xbox One je odlišný typ senzoru. K měření vzdálenosti se používá doba letu laserového pulsu. Každý pixel v hloubkovém obraze je rozdělen na dvě poloviny. Při frekvenci 1 GHz se střídavě vypínají a zapínají. Zapnutá polovina přijímá světlo a vypnutá jej nepřijímá. První polovina a laser jsou zapnuty po stejnou dobu. Poté se první polovina a laser vypnou. Zapne se druhá polovina pixelu, která přijímá světlo. Laserový puls na prvním řádku na Obrázek 6 je přijat pouze první polovinou pixelu, to znamená, že má téměř nulové zpoždění a vzdálenost je taky téměř nulová. Na druhém řádku je část pulsu přijata i druhou polovinou pixelu. Podle množství přijatého světla, můžeme zjistit dobu letu pulsu a následně vzdálenost. Na třetím řádku je snímač příliš vzdálen od laseru a doba letu přesáhne do první poloviny pixelu. To způsobí nejasný výsledek. Kvůli tomuto jevu Kinect v2 provádí dvě taková měření. První pouze v nižším rozlišení pro zjištění přibližné vzdálenosti. Poté podle výsledku nastaví delší nebo kratší časové okno pro jednu periodu zapnutí obou polovin. (LAU, 2013)



Obrázek 6 – Puls na prvním řádku se vrátí v nulovém čase. Na druhém řádku puls stihne zapnuté obě poloviny pixelu, podle rozdílu přijatého světla se pozná vzdálenost. Na třetím řádku puls přesáhne pixel a neměří se, je totiž příliš daleko pro daný časový úsek. (LAU, 2013)

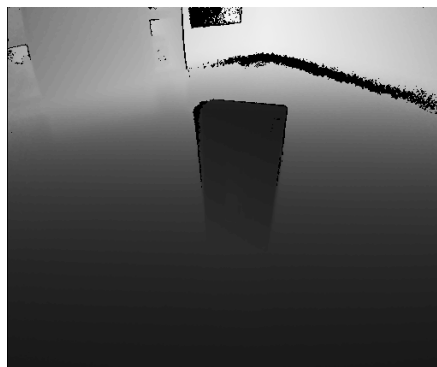
⁷ Převzato z: <https://www.ifixit.com/Teardown/Xbox+One+Kinect+Teardown/19725>

2.2 Pořizování 3D mračna bodů pomocí MS Kinect v2

Z Kinectu můžeme získat hloubkový (Obrázek 7), infračervený a barevný (Obrázek 8) obraz. Hloubkový obraz se spojí s barevným a promítne se do prostoru. Rozlišení hloubkového obrazu je 512 x 424⁸. Barevný obraz je v rozlišení 1920 x 1080. Tedy při sloučení těchto dvou obrazů se použije rozlišení hloubky a ztratí se část informací o barvě. Mapování barvy není úplně přesné (Obrázek 9). U jednoho snímku nám logicky vznikne stín za snímaným objektem.



Obrázek 8 – Záběr RGB kamery Kinectu v2



Obrázek 7 – Záběr hloubkové kamery Kinectu v2



Obrázek 9 – Spojení hloubky a RGB snímku do mračna bodů

2.3 Metody registrace mračna bodů

Protože obvykle nechceme pouze jeden snímek, ale zajímá nás spíše mapa okolí, je potřeba jednotlivé snímky spojovat dohromady. Pro správné spojení nového snímku a předchozích je potřeba znát polohu a orientaci senzoru vůči předchozím snímkům. Respektive je potřeba znát nové mračno bodů a již nasnímané mračno bodů.

⁸Specifikace je dostupná zde: <https://developer.microsoft.com/en-us/windows/kinect/hardware>

Pokud má senzor jednotku pro měření pohybu (IMU – Inertial Measurement Units), lze využít data z této jednotky pro zjištění polohy. Pokud senzor tuto jednotku nemá, jako například Kinect, je potřeba sledovat z dat, jak se senzor pohybuje. Samozřejmě jde tyto přístupy kombinovat.

Pro sledování pohybu z dat je možné využít algoritmus Iterative Closest Points (ICP) (HENRY et. al. , 2012).

3 Existující nástroje pro pořízení a zpracování mračna bodů

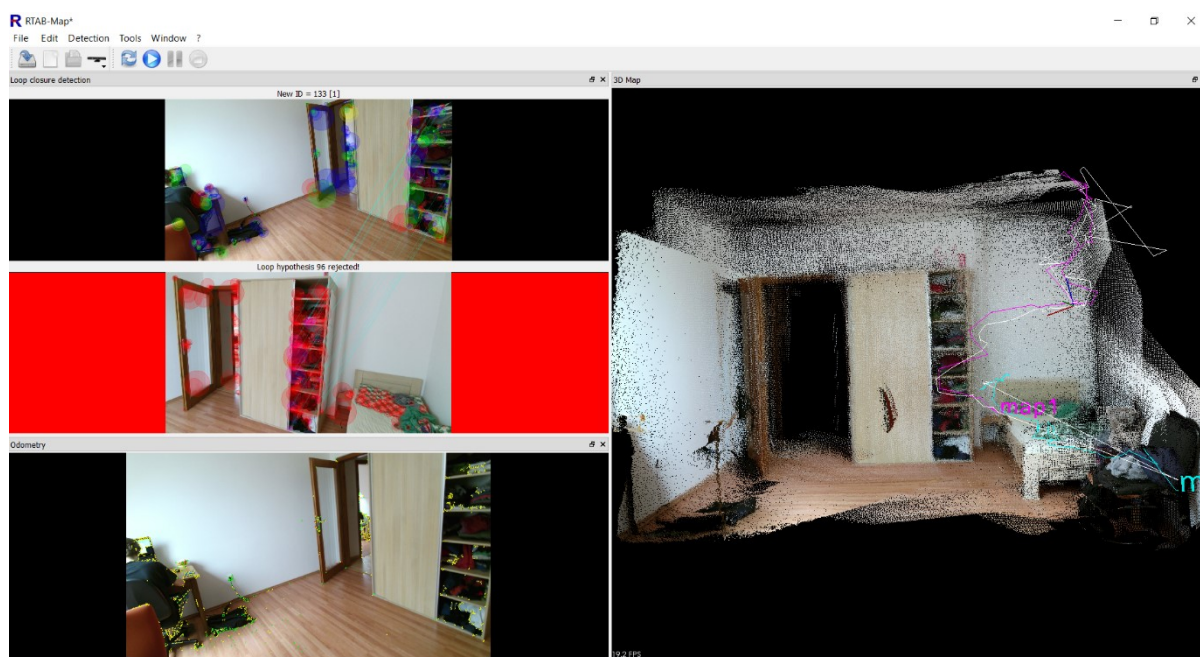
Programů, které umí pracovat s mračny bodů je mnoho. Zde jsou zmíněné nástroje pro vytvoření, úpravu a implementaci (knihovna) mračna bodů.

3.1 Nástroje pro registraci mračna bodů

Zde jsou zmíněné ty programy, které jsem našel a stály za vyzkoušením.

RTAB-Map⁹

Tento multiplatformní nástroj pro RGB-D SLAM pomocí grafu je založený na globální Bayesovské detekci uzavírání smyčky. Pro detekci uzavírání smyčky se používá přístup *bag-of-words* pro určení, zda nový snímek pochází z předchozí lokace nebo nové lokace. Pokud je akceptována detekovaná smyčka, přidá se do mapy grafu nový záznam. RTAB-map podporuje několik různých senzorů včetně Kinectu.



Obrázek 10 – RTAB-Map. Snímané mračno bodů obsahuje hodně šumu.

RTAB-Map lze použít pro mapování velkého prostoru. Ale jednotlivé rovinné plochy (např. podlaha na Obrázek 10) mohou být nepřesně zaznamenány a ve výsledném mračně bodů nejsou příliš rovné. Pro výpočet používá procesor a může se stát, že nestíhá zpracovávat snímky v reálném čase.

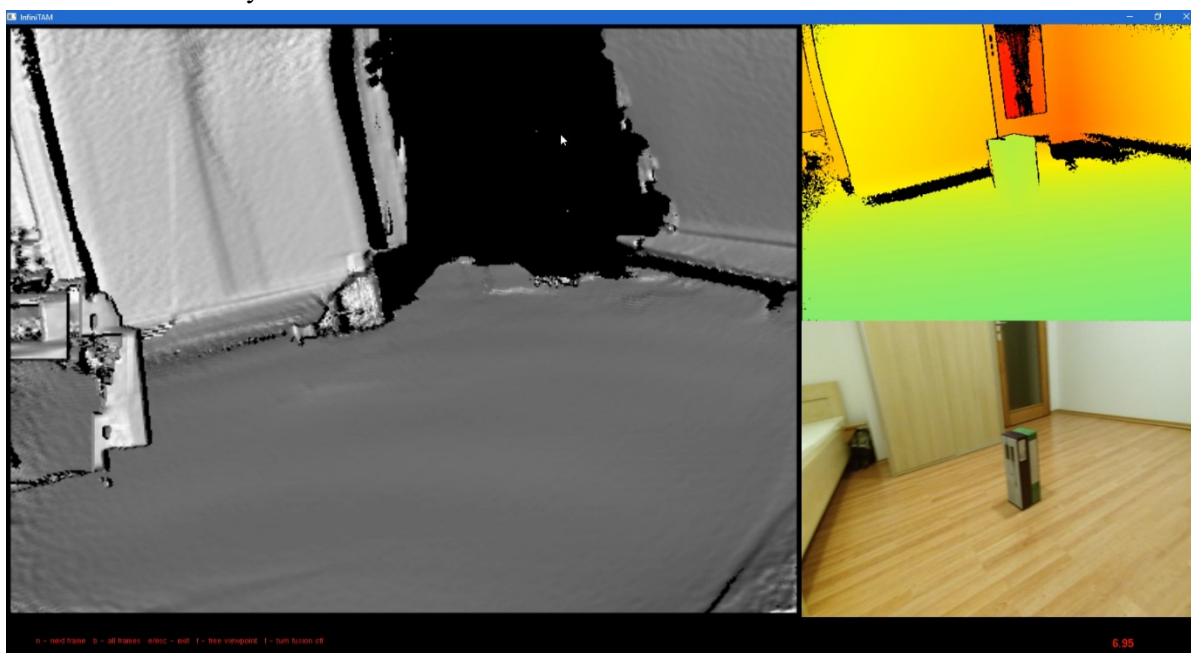
RTAB-Map zvládne exportovat mračna bodů v obvyklých formátech (.pcd, .obj, .ply) i jednotlivé barevné a hloubkové snímky.

⁹ Dostupné z: <http://introlab.github.io/rtabmap/>

InfiniTAM¹⁰

InfiniTAM je multiplatformní framework pro zaznamenání mračna bodů v reálném čase. Podporuje senzory od *RealSense*, *OpenNI* rozhraní i Kinect pomocí oficiálního SDK. Pro výpočet umí použít procesor nebo grafiku. Grafický výpočet lze provést pomocí CUDA od Nvidie nebo Metal od Applu. Vyměňuje si paměť mezi grafickou kartou a procesorem. To umožňuje skenovat virtuálně nekonečně velké prostředí¹¹.

Tento framework značně modulární a jde překládat jen částečně. Tedy není potřeba instalovat všechny knihovny, které zvládne použít. Povinné jsou pouze OpenGL a GLUT pro grafické rozhraní. Pro použití s Kinectem v2 stačí pouze Kinect SDK. Přímě pro Kinect v2 ale nepodporuje zaznamenávání barvy.



Obrázek 11 – InfiniTAM - záznam místnosti

Při mém testování nemělo InfiniTAM dobré výsledky. V nastavení je možné si zvolit různé trackery: podle barevného obrazu (nebyl k dispozici pro Kinect), algoritmus ICP, vážené ICP, tracker Ren (Ren et al 2012) a napojení na jednotku IMU (*inertial measurement unit*), kterou Kinect neobsahuje. Nejlépe se choval tracker Ren, ale i ten se ztrácel a vznikalo nepřesné mračno bodů. Například na Obrázek 11 lze vidět, že v mračně bodů chybí krabice.

Exportovat se dá mesh ve formátu obj nebo stl. Vzhledem ke skladbě frameworku by nebylo složité přidat další formáty.

Kinect Fusion

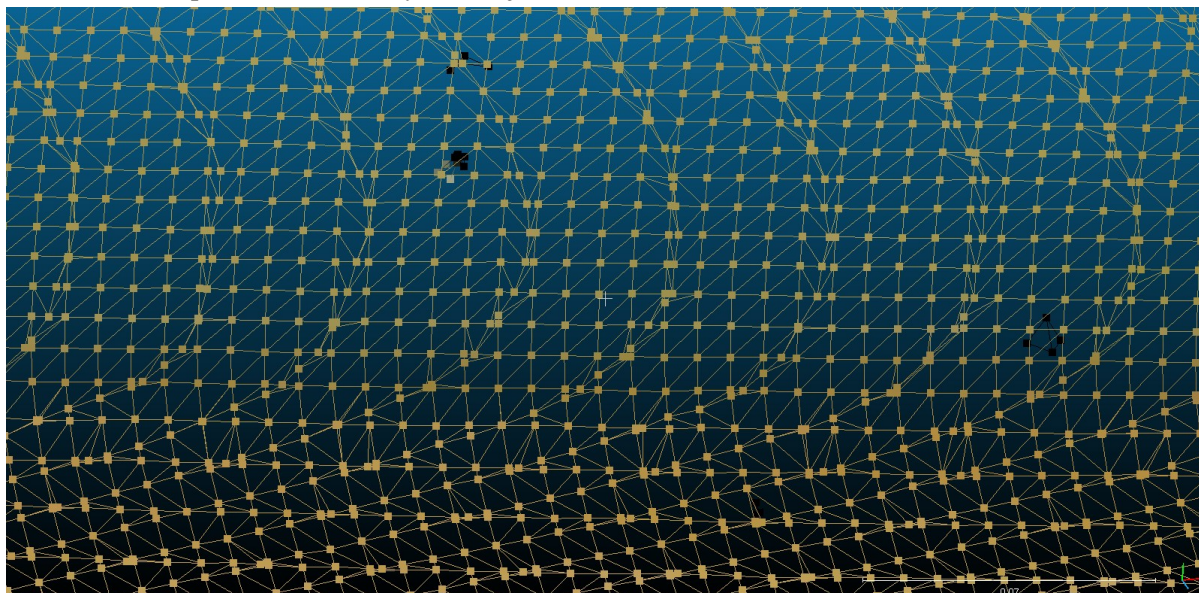
Kinect Fusion je přímo obsažen v SDK od Microsoftu. Má vyladěné sledování pohybu senzoru. Ze všech mnou testovaných programů zachytával prostředí nejlépe. Pokud při prudkém pohybu ztratí orientaci o tom, kde se senzor nachází, čeká na vrácení senzoru do poslední známé lokace. Většinou se správně podaří znovu najít, kde se nachází. Nedochozí tedy ke špatnému spojení mračen bodů.

¹⁰ Dostupné z: <http://www.robots.ox.ac.uk/~victor/infinitam/>

¹¹ Odtud taky pochází název InfiniTAM, kde první část znamená anglické Infinite – nekonečný.

Největší omezení Kinectu Fusion je maximální velikost mračka bodů. Používá jeden souvislý blok paměti grafické karty. Maximální velikost takového bloku je typicky kolem 1GB (Microsoft Developer Network, n.d.).

Pomocí triangulace umí rekonstruovat povrch objektů. Tyto objekty potom exportuje jako mesh vrcholů s povrchem. Vrcholy obsahují RGB hodnotu.



Obrázek 12 – Body a wireframe výstupu z Kinect Fusion

3.2 Knihovna Point Cloud Library (PCL)¹²

PCL je open-source knihovna pro práci s mračky bodů. Obsahuje několik modulů pro různé účely.

PCD formát

Jedná se o hlavní formát knihovny PCL a je rozšířen i mimo jiné externí aplikace. Umožňuje ASCII i binární zápis.

Výhody oproti ostatním formátům jsou:

- Možnost ukládat organizovaný dataset mračka bodů. Díky tomu je možná lepší práce v reálném čase.
- Do formátu je možné uložit různé číselné datové typy (od char po double). Je tak více efektivní při ukládání různých dat.
- Umožňuje ukládat n-D histogramy pro deskriptory příznaků.

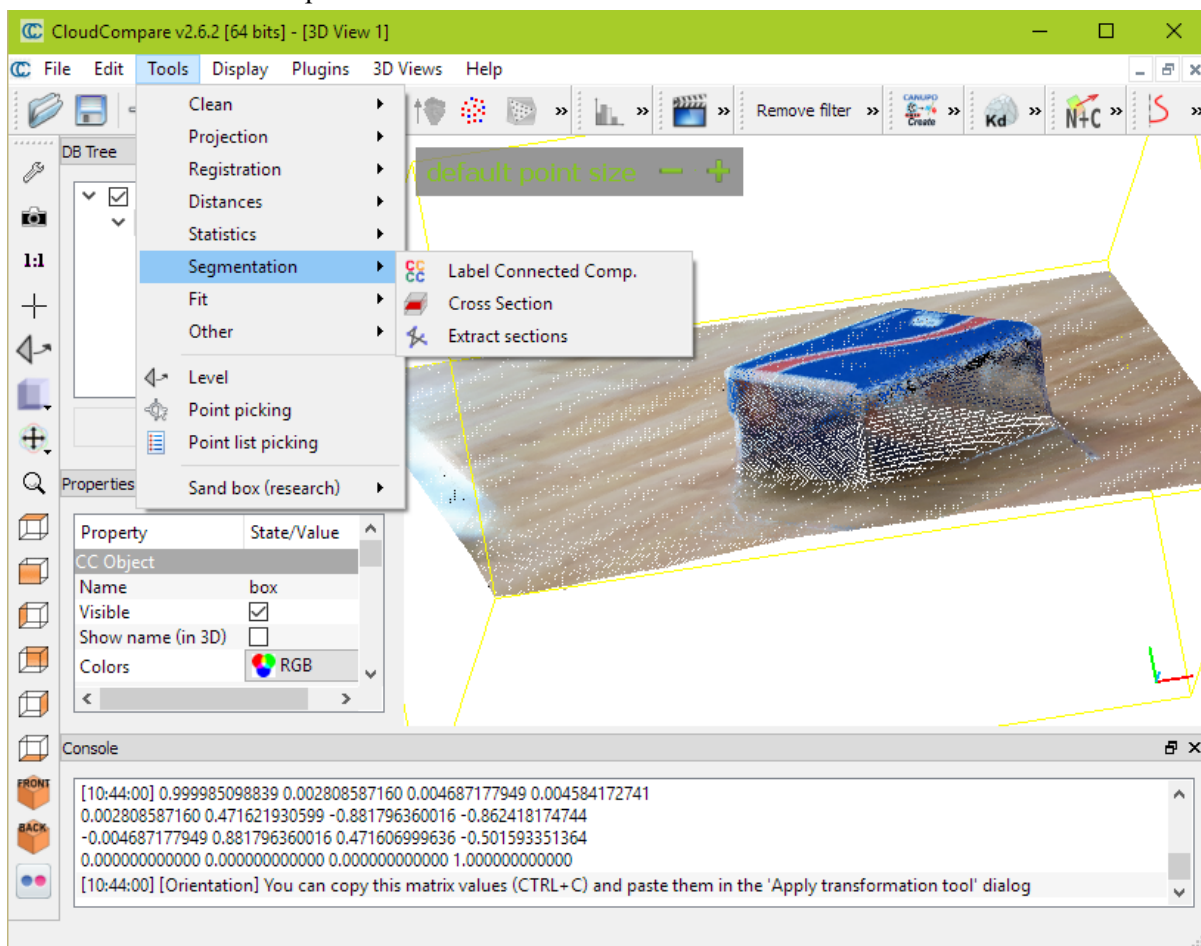
Hledání rovin

Pro hledání geometrických modelů v mračce bodů je v PCL modul Sample Consensus (SAC). Podporuje různé útvary jako kužel, válec a taky roviny.

¹² Více informací na: <http://pointclouds.org/documentation/>

3.3 Nástroje pro úpravu mračna bodů

Pro práci s mračny bodů je ideální nástroj CloudCompare¹³. Umožňuje úpravy jako posuvy, rotace, ořezání mračna bodů, spojení více mračen dohromady, výpočet normál. To jsou jen vlastnosti, které jsem použil. Všechny vlastnosti tu snad ani nejde vyjmenovat. Podporuje externí pluginy, takže další funkcionality se dá lehce přidat.



Obrázek 13 – CloudCompare – Nástroj pro úpravu mračna bodů

¹³ Dostupný z: <http://cloudcompare.org/>

4 Návrh metody pro detekci rovinných ploch v místnosti a jejich převod na polygony

Cílem práce je vytvořit polygonální model významných ploch místnosti, které představují odhad geometrie místnosti. Tedy je potřeba vyřešit následující problémy. Nejprve je nutné získat mračno bodů. Poté najít největší roviny v mračně bodů. V jedné rovině může být více ploch, proto je potřeba tyto plochy rozlišit. Body, které leží v dané ploše, se převedou na polygon.

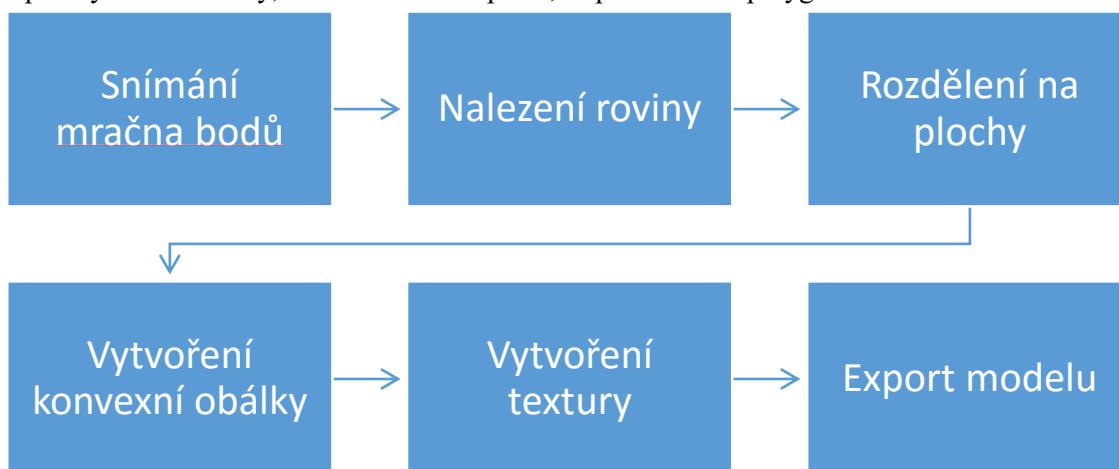


Diagram 1 – Návrh jednotlivých kroků

4.1 Snímání mračna bodů

Zde je potřeba rozlišit, zda chceme snímat pouze jeden snímek (jeden pohled) nebo snímat mapu prostoru.

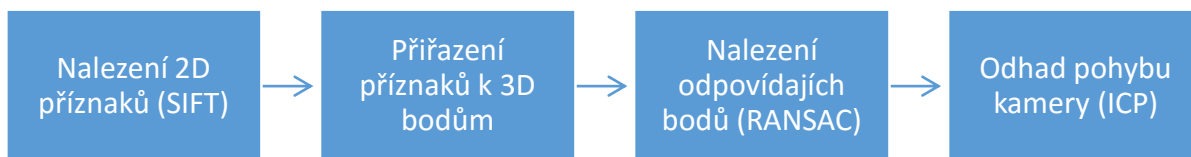
Jeden snímek

Mračno bodů z jednoho snímku můžeme jednoduše pixel po pixelu složit pomocí oficiálního SDK pro Kinect v2. Umí spojit hloubkový a barevný obraz a umístit ho do souřadnicového systému. (Obrázek 9)

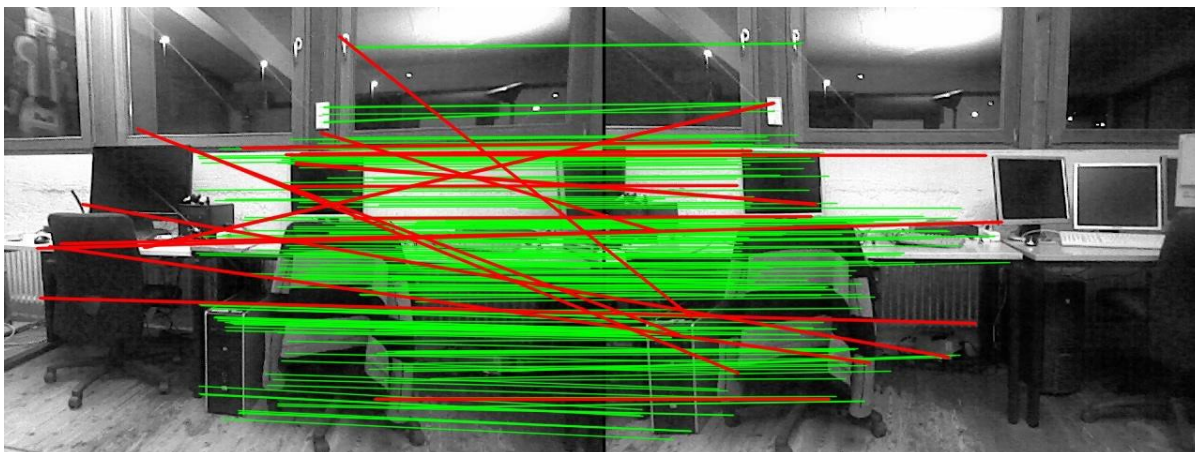
Více snímků (RGBD SLAM)

Při vytváření mračna bodů z více snímků (videozáznamu) je potřeba spojit jednotlivé snímky v prostoru. Ze zařízení MS Kinect v2 není možné získat pohyb senzoru v prostoru. Kinect je určen pro statické umístění, proto neobsahuje např. gyroskopické senzory, které by měřily pohyb zařízení.

Pohyb senzoru se dá zjistit porovnáním existujícího mračna bodů s novým snímkem. Je tedy potřeba se lokalizovat v již existujícím mračně bodů a správně připojit nový snímek. Je potřeba najít shodné body v obou mračnách a zjistit jejich vzájemné posunutí a rotaci. Poté co víme, kde je umístěn senzor, můžeme přidat nový snímek do existujícího mračna. Tento postup se nazývá *SLAM* (simultánní lokalizace a mapování).



Obrázek 14 – Postup pro SLAM (ENGELHARD et. al., 2011)



Obrázek 15 – Spojení obrazu s chybami (J. Sturm)

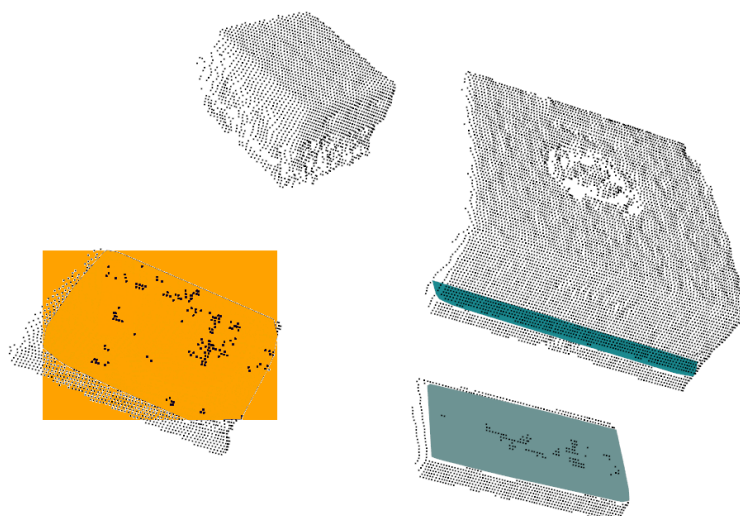
Vzhledem k výše zmíněným problémům jsem usoudil, že vlastní implementace *SLAMu*, která by měla mít přijatelné výsledky, by byla mimo rozsah této práce.

4.2 Nalezení roviny

V mračně bodů lze najít body, které leží v jedné rovině. Zde je možné použít algoritmus jako RANSAC a jemu podobné.

4.3 Rozdělení roviny na plochy

Po nalezení roviny nás nezajímá samotná rovina, ale jednotlivé body, které tvoří plochy v této rovině. Tedy plochy, které mohou ležet v jedné rovině, ale mohou být oddělené mezerou v prostoru. K rozdělení těchto ploch se využije algoritmus extrakce Euklidovského shlukování.

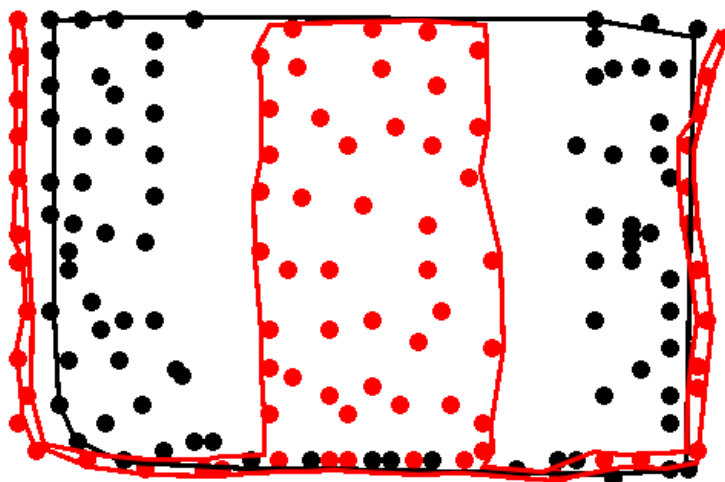


Obrázek 16 – Rozdělené plochy v jedné rovině

4.4 Projekce bodů do 2D a vytvoření konvexní nebo konkávní obálky

Nalezené body nejsou v rovině, ale spíše v nějakém rozmezí, které se dá specifikovat. Toto rozmezí, v kterém se bodu mohou vyskytovat, je nutné, kvůli nepřesnosti senzoru. Proto se body promítnou do jedné roviny, standardně té, kterou jsme našli. Následně můžeme udělat konvexní nebo konkávní obálku, která později určí tvar polygonu.

Konvexní obálka se více hodí pro velké plochy v pozadí například stěna za nábytkem (znázorněna černě na Obrázek 17). Takovou plochu chceme mít typicky kompletní bez výřezu. Konvexní obálka není vhodná pro vyčnívající objekt v popředí (znázorněno červeně na Obrázek 17). Rovina, která obsahuje body objektu v popředí, dost možná obsahuje také jiné body, které mohou být s tímto objektem spojené.



Obrázek 17 – Porovnání konvexní a konkávní obálky. Černé body jsou v pozadí a jsou obaleny konvexně. Červené body jsou v popředí a jsou obaleny konkávně.

4.5 Vytvoření textury

Pro zbavení 3D souřadnic, rovinu s body transformujeme tak, aby jedna ze souřadnic byla nulová. Poté najdeme krajní bod a odečteme jeho souřadnice od ostatních. Dále bude potřeba odhadnout rozlišení obrazu podle počtu bodů. Poté se postupně projde každý bod a spočítá se pro něj souřadnice v novém obrazu.



Obrázek 18 – kompletní mračno bodů



Obrázek 19 – mračno bodů v rovině pro tvorbu textury

Jak můžeme vidět na Obrázek 19, v textuře vznikají díry pod ostatními objekty, místa, které nejsou vidět. Podobně mohou vznikat stíny za objekty, pokud senzor dané místo neviděl. Pro tyto oblasti bude potřeba odhadnout výplň. Můžeme nastavit stejnou barvu, pro všechny neviděná místa. Nebo by se mohl použít algoritmus PatchMatch¹⁴ na doplnění podle okolí. (BARNES et. al. , 2009)

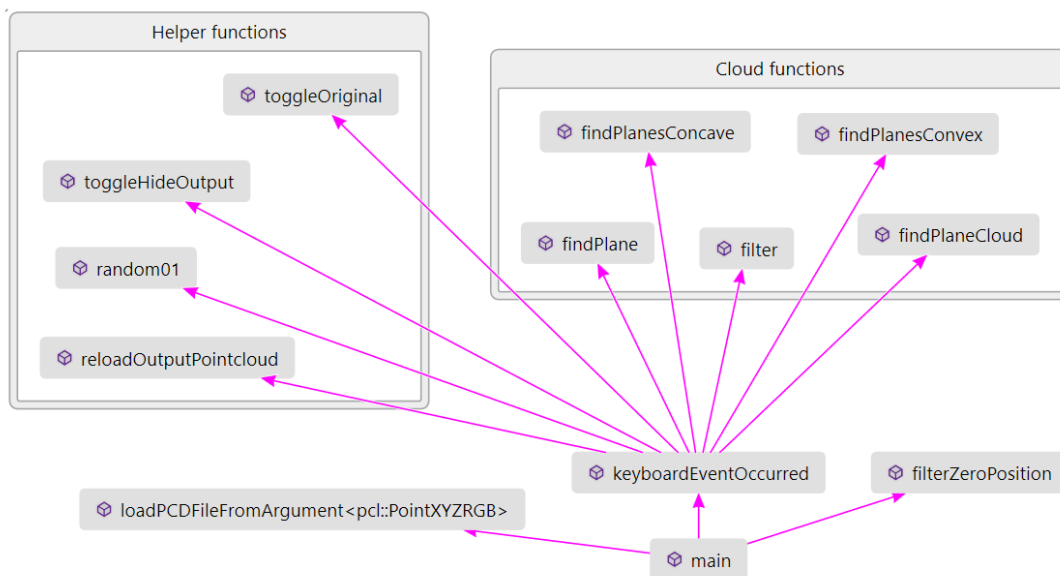
4.6 Export modelu

Pro export modelu je vhodné pospojovat nalezené plochy do meshe a napojit na něj texturu. Musí se nalézt hrany různých ploch, které jsou blízko u sebe, a následně je spojit. Tento cíl je prakticky dosažitelný jen pro jednodušší scény (například Obrázek 18).

¹⁴ Algoritmus PatchMatch můžete znát z aplikace Adobe Photoshop jako Content-Aware Fill.

5 Implementace

Práce obsahuje 2 aplikace. První je Cloud Grabber, slouží pro zachycení jednoho snímku z Kinectu a uložení ve formátu pcd. Druhá aplikace se jmenuje 3D model. Je založená na instanci třídy `visualization::PCLVisualizer` z knihovny PCL.



Obrázek 20 – Mapa kódu aplikace 3D model

5.1 Snímání mračna bodů

Snímání je rozlišeno pro jeden snímek a složitější registraci jednotlivých

Jeden snímek

Pro snímání mračna bodů jsem částečně upravil kód aplikace Cloud Viewer¹⁵. Aplikace je v práci pod názvem Cloud Grabber. Tato aplikace dokáže zaznamenat jeden snímek a uložit ho jako mračno bodů. Používá oficiální SDK od Microsoftu pro Kinect v2. `<kinect.h>`. Aplikace

Pro každý bod v hloubkovém obraze se spočítá barva z barevného obrazu a souřadnice v prostoru pomocí funkcí ze SDK. Tyto funkce má rozhraní `CoordinateMapper * pCoordinateMapper`.

Klávesou F10 se zachytí aktuální snímek a po zadání názvu se souboru, se uloží v pcd formátu.

Více snímků

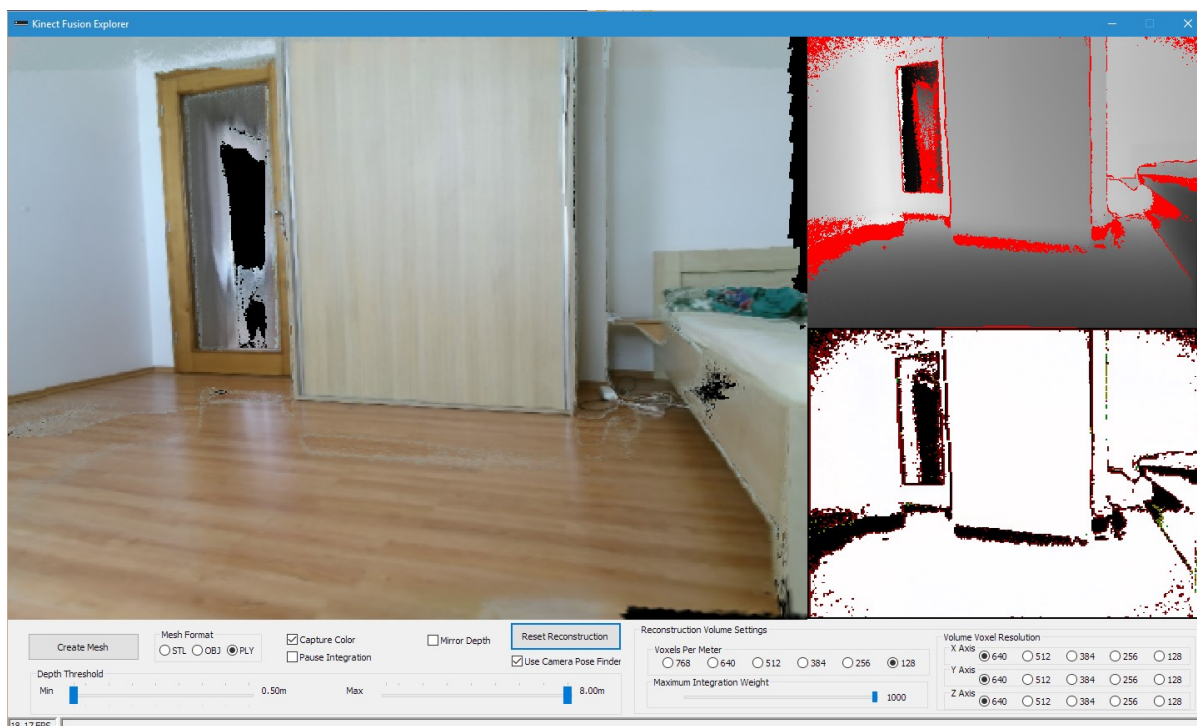
Pro zachycení více snímků je potřeba použít externí aplikaci. Po vyzkoušení aplikací *RTAB-Map*, *InfiniTAM* i algoritmů obsažených v PCL knihovně¹⁶ jsem usoudil, že nejlepší bude použít aplikaci

¹⁵ Kód je dostupný zde: <http://unanancyowen.com/?p=1220>.

¹⁶ Kinfu Largescale

Kinect Fusion Explorer obsaženou přímo v Kinect 2 SDK. Tato aplikace je výrazně omezena maximální velikostí bloku paměti, který lze na grafické kartě alokovat. (Kinect Fusion, n.d.)

V Kinect Fusion Exploreru je potřeba nastavit aby zachytával barvu (*Capture Color*). Rozlišení os x, y, z na 640 voxelů. A v závislosti na velikosti místnosti počet voxelů na metr. S větším rozsahem klesá počet detailů, které se dají uložit. Váhu integrace (*Maximum Integration Weight*) nastavíme na maximum (1000), protože chceme zaznamenat statickou místnost a ne pohybující se prvky v ní. Pro výstup nastavíme formát PLY. Nastavení lze vidět na Obrázek 21.



Obrázek 21 – Nastavení Kinect Fusion Explorer

Při samotném skenování místnosti je možné zařízením Kinect pohybovat všemi směry a poměrně rychle. Pokud se zařízení při skenování ztratí, to znamená, že neví, kam má přidat nové body, je potřeba se vrátit do poslední známé lokace.

Kinect Fusion Explorer vytvoří mesh všech pospojovaných bodů. Ten můžeme uložit ve formátu PLY a pomocí nástroje *ply2pcd.exe*, obsaženém v knihovně PCL, jej převést na mračno bodů ve formátu PCD.

Aplikace 3D model

Aplikace 3D model je hlavní část práce. Obsahuje vizualizaci načteného mračna bodů ze souboru a umožňuje nad ním provádět operace, které jsou popsány v následujících kapitolách.

Mračno bodů se zadá jako cesta k souboru do prvního argumentu programu. Program si vytvoří jednu barevnou kopii a jednu bez barvy.

5.2 Filtrace mračna bodů

Nasnímané mračno bodů obvykle obsahuje nežádoucí šum. V aplikaci 3DModel je pro tento účel implementována funkce `filter()`. Používá třídu `StatisticalOutlierRemoval<PointXYZ>`, která používá algoritmus k-nejbližších sousedů. Na vstupu očekává počet sousedních bodů a směrodatná odchylka, podle které se spočítá, které sousední body má zahrnovat.

Lze ji spustit klávesou F1. Při stisknutí Ctrl+F1 je možné zadat parametry funkce. Standardně má nastavený počet sousedících bodů na 10 a směrodatná odchylka je nastavena na hodnotu 5.

5.3 Nalezení roviny

Pro nalezení rovin je v projektu implementována funkce `findPlane()`.

Tato funkce pomocí objektu třídy `SACSegmentation<PointXYZ>` najde rovinu ve vstupním mračně bodů `input`. Pro nalezení roviny je nastaven algoritmus RANSAC a typ modelu je nastaven na rovinu. Rozsah, ve kterém může rovina ležet, se specifikuje parametrem funkce `distanceThreshold`. Rozsah se udává v metrech. Objekt této třídy nalezne rovinu, která obsahuje nejvíce bodů. Výstupem jsou pak koeficienty roviny a body `inliers`, které v ní leží (v rámci daného rozmezí). Tyto body se promítnou pomocí objektu třídy `ProjectInliers<PointXYZ>` do roviny, protože předtím mohly být v rozmezí specifikovaném pomocí parametru `distanceThreshold`.

Pomocí objektu třídy `ConvexHull<PointXYZ>` se vytvoří konvexní obálka kolem promítnutých bodů. Protože nyní známe orientaci roviny z koeficientů a taky vrcholové body z konvexní obálky, můžeme vytvořit polygon `pcl::PlanarPolygon<PointXYZ>`.

Z výstupního mračna bodů `output` se následně odstraní body ležící v rovině `inliers`. Díky tomu půjde ve výstupním mračně bodů najít další rovina, pokud tam zůstal dostatečně velký počet bodů.

V aplikaci 3DModel jde funkci použít stisknutím klávesy F2. Kombinací Ctrl+F2 lze zadat parametr `distanceThreshold`.

Dále existuje podobná funkce `findPlaneCloud()`. Tato funkce najde rovinu stejně jako funkce `findPlane()`. A vrací pouze `inliers` jako mračno bodů a koeficienty roviny.

5.4 Rozdělení roviny na plochy

Rozdělení roviny na plochy je implementováno ve funkcích `findPlanesConcave()` a `findPlanesConvex()`. Tyto funkce na vstupu očekávají `inliers` a koeficienty roviny. Nejprve body pomocí objektu třídy `EuclideanClusterExtraction<PointXYZ>` rozliší na jednotlivé shluky bodů pomocí Euklidovské extrakce shluků. Tento objekt má nastavený minimální velikost shluku na 100 bodů, maximální velikost shluku na 1 000 000 bodů a do shluku zahrnuje body do vzdálenosti 10 centimetrů.

Poté se po jednom prochází každý shluk bodů. Znovu se pomocí RANSACu vyhledá rovina, protože původní rovina mohla být mírně odkloněná a zároveň zahrnovala tyto body. Nově nalezené body se promítnou do jedné roviny.

Funkce `findPlanesConvex()` vytvoří konvexní obálku a následně polygon. Funkce `findPlanesConcave()` navíc na vstupu očekává minimální vzdálenost Alpha, na kterou má okrajové body spojit. Vytvoří konvexní obálku a následně polygon.

Obě funkce vrací vektor polygonů.

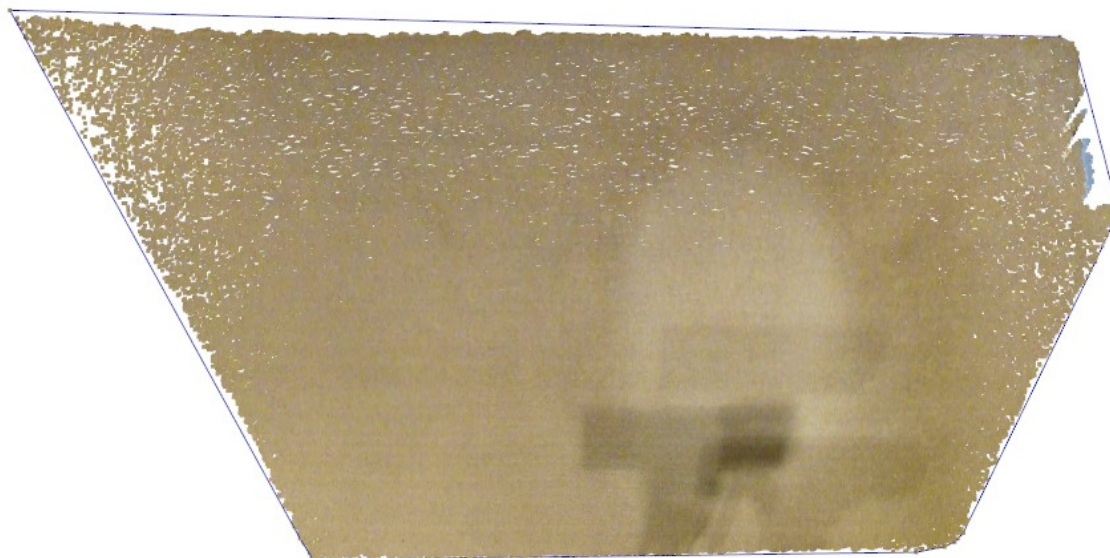
V aplikaci 3D model lze funkci `findPlanesConvex()` spustit klávesou F3. Funkce `findPlanesConcave()` se spustí po stisknutí klávesy F4 a podporuje zadání vlastních parametrů při stisknutí Ctrl+F4.

U hledání plochy s konkávní obálkou doporučuji zadávat tyto hodnoty. Pro velké plochy rozmezí pro segmentaci nastavit na 0.04 a Alphu na 4. Pro menší plochy 0.005 a 0.2.

Výpočet metody může trvat hodně dlouho, pokud mračno obsahuje několik milionů bodů. Doporučuji první vyfiltrovat. Poté pokud je v mračně zřejmá stěna, kde není potřeba rovinu dělit na menší plochy, použít `findPlane()` pod klávesou F2.

5.5 Textura

Po stisknutí klávesy F6 v aplikaci 3DModel se spustí algoritmus pro tvorbu textury. Algoritmus najde největší rovinu stejně, jak je popsáno v kapitole 5.2. Z koeficientů roviny se pomocí metody `getTransFromUnitVectorsZY` spočítá transformační matice pro rotaci bodů roviny do roviny, pro kterou platí, že osa $z = 0$. Výsledek je možný vidět na Obrázek 22 a Obrázek 19.



Obrázek 22 – Body z podlahy připravené pro tvorbu textury.¹⁷

5.6 Další možnosti aplikace 3DModel

Aplikace bere jako svůj jediný argument cestu k souboru s mračnem bodů ve formátu pcd. Dané mračno bodů zobrazí na levé straně barevně a na pravé straně černobíle. Nahrazování ploch polygony probíhá na pravé straně. Klávesou PgUp se dá přepnout na pravé straně zobrazení původního a upraveného mračna bodů. Klávesa PgDn se dá skrýt mračno bodů, zůstanou tak vidět pouze polygony.

Klávesa Home načte znovu původní mračno bodů pro pravou stranu. Klávesa BackSpace umožňuje jeden krok zpět. Díky této funkcionalitě se dá jednoduše experimentovat s funkcemi programu.

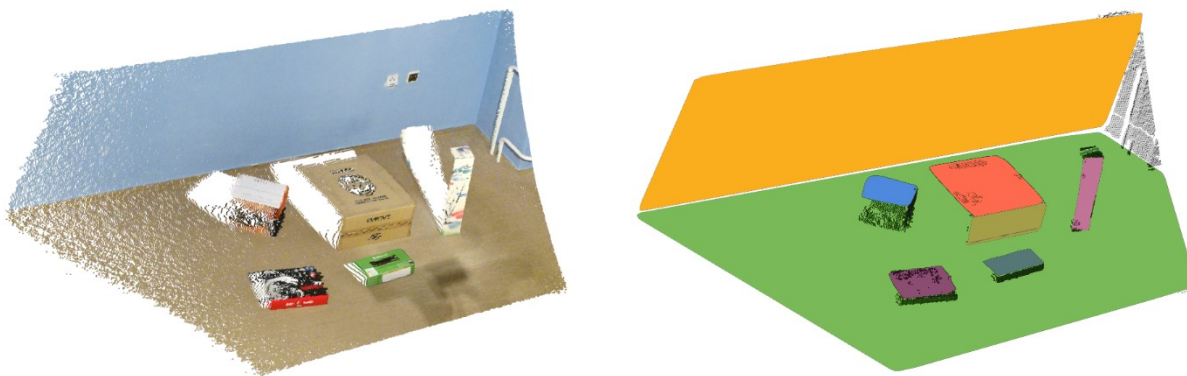
¹⁷ Na obrázku lze vidět i stín Kinectu, který případnou texturu znehodnotí.

6 Výsledky

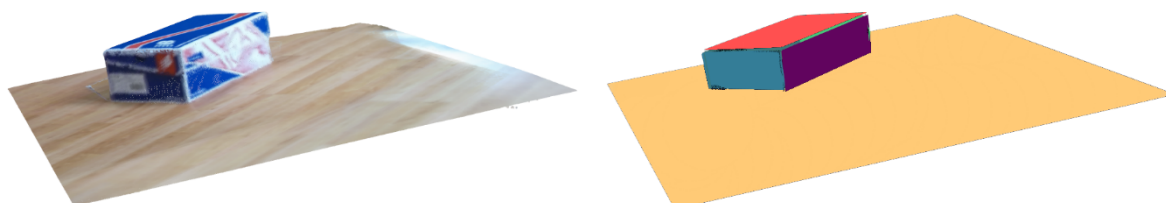
Aplikace, po zadání parametrů pro hledání rovin, rozpozná jednotlivé plochy v místnosti a určí jejich velikost, umístění a orientaci v prostoru.

6.1 Jednoduché scény

Aplikace zvládne zpracovat jednoduché scény i s jednoduchými objekty. Najdou se hlavní stěny místnosti i menší plochy na objektech.



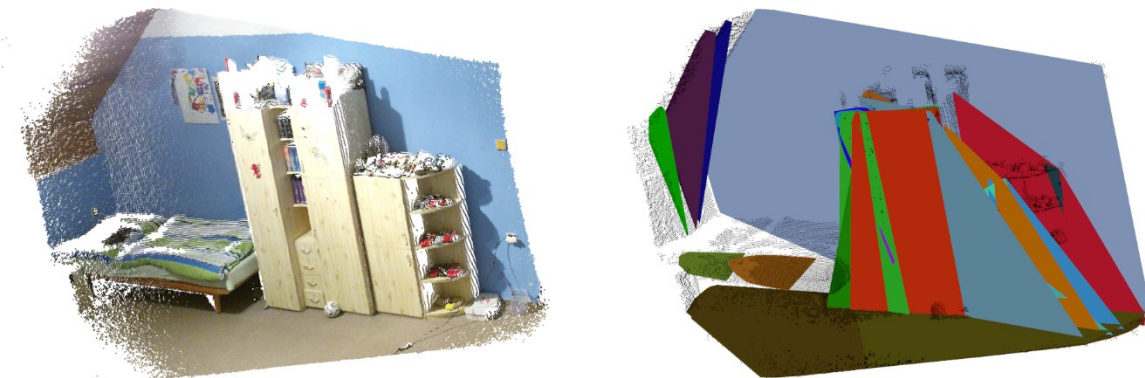
Obrázek 23 – Jednoduchá scéna s objekty. Mračno bodů je vytvořeno z 1 snímku



Obrázek 24 – Mračno bodů (vlevo) nahrazeno polygony (vpravo). Mračno bodů je vytvořeno pomocí Kinect Fusion

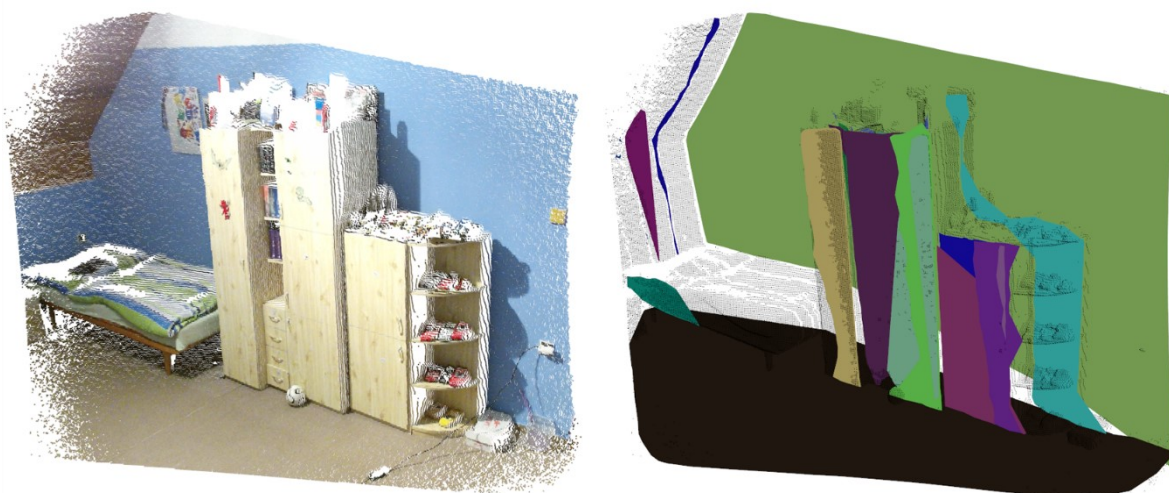
6.2 Složitější scény s objekty

Při komplexnějších scénách s nábytkem se najdou velké plochy, ale pro malé plochy se vytvoří nepřesná konvexní obálka (Obrázek 25).



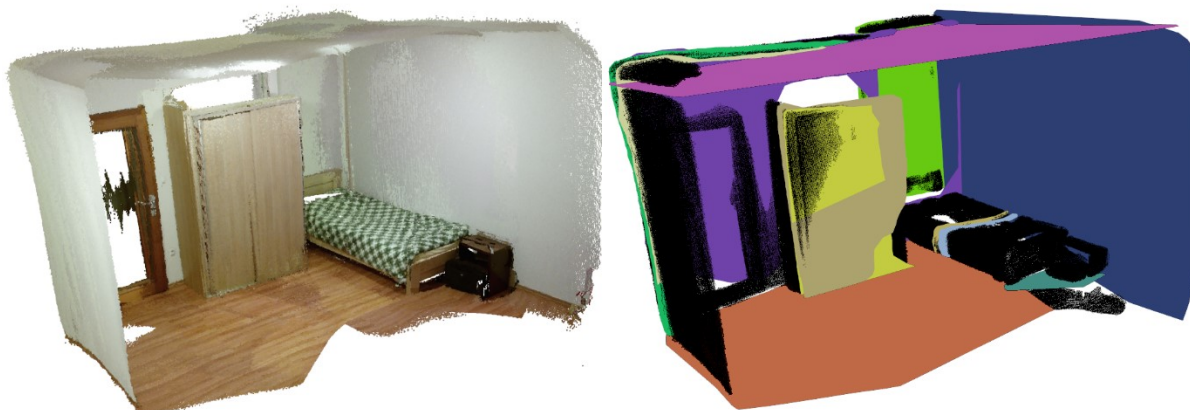
Obrázek 25 – Konvexní obálky u malých ploch. Mračno bodů je vytvořeno z 1 snímku.

Pokud se na stejné scéně použijí postupně různé parametry pro každou rovinu, jde dosáhnout mnohem lépe vypadající výsledek. Zde je použita konkávní obálka s určenou maximální vzdáleností mezi body.



Obrázek 26 – Výstup po ručním nastavování parametrů pro každou rovinu a maximální délky konkávní obálky. Mračno bodů je vytvořeno z 1 snímku.

Pro komplexnější scénu vytvořenou pomocí Kinect Fusion se rovinné plochy detekují obtížněji. Zaznamenané mračno nemá rovnoměrnou hustotu bodů. Toto by možná šlo vyřešit voxelizací mračna bodů předem. Dále plochy v jedné rovině nemusí být dostatečně odděleny.



Obrázek 27 – Výstup po ručním nastavování parametrů pro každou rovinu a maximální délky konkávní obálky. Mračno bodů je vytvořeno pomocí Kinect Fusion

```

Current path is : "C:\Users\Martin\Dropbox\FIT\Bachelor Thesis\Projects\3Dmodel\
3Dmodel"
loading pcd/box.pcd
F6 was pressed => texture prototype
Working...
PointCloud after projection has: 1295571 data points.
Concave hull has: 0 data points.
-0.0140411  -0.87331  -0.486963
0 0 1
Transformation matrix:
  0.999871  -0.0160759  0  0
-0.00782838  -0.4869  0.873423  0
-0.0140411  -0.87331  -0.486963  0
          0  0  0  1

minimum border point:  x=-0.503023  y=-1.4393  z=0.499608
maximum border point:  x=0.50357  y=-0.289845  z=0.499608
plane dimensions(aprox): 1.00659m x 1.14945m
Done.

```

Obrázek 28 – Výpis z konzole aplikace k Obrázek 19. Na osmém řádku lze vidět koeficienty nalezené roviny. Na dalším řádku jsou koeficienty roviny pro transformaci. Následuje transformační matice, nejmenší a největší krajní bod. Na předposledním řádku jsou přibližné rozměry plochy.

7 Možnosti další práce

7.1 Voxelizace mračna bodů

Mračno bodů není nasnímáno rovnoměrně a to ovlivní hledání rovin. Proto by se mračno bodů mělo zredukovat tak, aby hustota bodů byla všude stejná. Toho můžeme dosáhnout například použitím VoxelGrid filtru z knihovny PCL.

7.2 SLAM optimalizovaný pro zaznamenávání barvy

Na SLAMu se dá vždy pracovat, protože vytvoření kvalitního a rychlého algoritmu není vůbec jednoduchá záležitost. Zde by bylo dobré provést podrobnou analýzu, jak funguje Kinect Fusion z SDK. Případně na něm začít stavět. Sledování pohybu Kinectu by pomohlo připevnění chytrého telefonu s gyroskopem a používat jeho synchronizovaná data.

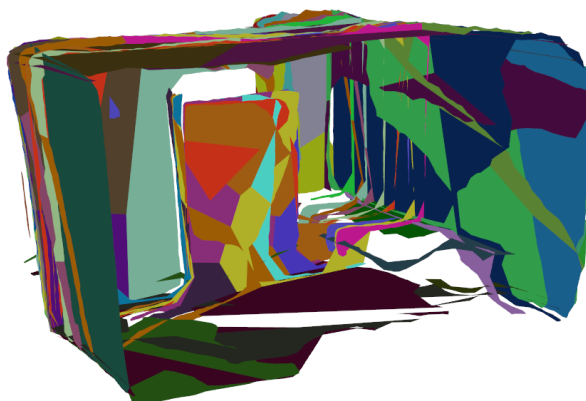
Ke sledování pohybu by se mohly zároveň používat příznaky z infračervené, barevné i hloubkové kamery. Dále by šlo postupně počítat konfiguraci osvětlení místnosti, určit a umístit zdroje světla. Tím bychom získali barvu nezávislou na osvětlení místnosti. (JIN et. al. , 2008) Tuto barvu můžeme taky využít pro vytváření textury.

Protože porovnávání všech příznaků je velmi výpočetně náročné a teoreticky by šlo dělat paralelně, většina by měla být implementována pomocí CUDA.

Pro začátek by se mohl rozšířit a upravit framework InfiniTAM.

7.3 Automatizace hledání rovin

Aktuální aplikace je potřeba manuálně ovládat a odhadovat parametry pro hledání rovin. Podle počtu bodů nalezených v poslední rovině a zbývajících počtu bodů v mračně by šlo odhadovat, jak velká bude další plocha. Nastavení by se postupně zjemňovalo. Poté by bylo potřeba udělat další zpracování. Spojit plochy, které se protínají a mají podobné koeficienty.



Obrázek 29 – Nalezené plochy po několika iteracích hledání rovin při stejně zvolených parametrech.

Taky by se mohlo paralelně spustit více hledání rovin s různými parametry. Současná implementace využívá pouze jedno jádro procesoru. Případně by se to mohlo spustit na grafické kartě. Po doběhnutí těchto hledání by se vybrala například nejrozměrnější rovina. Nebo samotný RANSAC spustit paralelně. (ALEHDAGHI et. al., 2015)

7.4 Export pro snadné použití v programech pro 3D grafiku

Výsledkem aplikace je set polygonů umístěných v prostoru. Pro lepší využití by bylo vhodné tyto polygony pospojovat do jednoho nebo více modelů. Musela by se také poskládat textura pro jednotlivé modely. Potom tyto modely exportovat do univerzálních formátů pro 3D model.

8 Závěr

Práce se zabývala nalezením rovinných ploch, za účelem odhadu geometrie, v mračně bodů pomocí zařízení MS Kinect v2. Senzor Kinect v2 byl zvolen pro svoji cenovou dostupnost a přesnost měření. Při pořizování záznamů se projeví nevýhody Kinectu. Mezi hlavní se řadí překážející kabel a nutnost připojení k elektřině. Kvůli tomu senzor není tolik mobilní. Kinectu také chybí senzor pohybu a musí se spoléhat na odhad pohybu z dat. Z tohoto pohledu by mohly být lepší zařízení s podporou Google Tango, ale nejspíše za cenu menší kvality.

Operace s mračnem dat jsou prováděny pomocí knihovny PCL. Pro vyhledání rovin se používá algoritmus RANSAC. Následně se body ležící v rovině rozdělí na shluky pomocí Euklidovské extrakce. Pro vytvoření polygonu se kolem těchto bodů se vytvoří konvexní nebo konkávní obálka.

Výsledná aplikace dokáže najít stěny místnosti i rovinné plochy objektů v ní. Nalezené plochy jsou převedeny na polygony a známe jejich relativní umístění, velikost i orientaci v prostoru. V práci je také algoritmus pro nachystání bodů plochy pro tvorbu textury. Lze vypsát rozměr největší plochy v místnosti.

Vzhledem k náročnosti průzkumu nástrojů pro registraci mračna bodů, kde jsem musel překládat různé verze knihoven třetích stran, mi nezbyl čas pro dotažení práce s texturami. Konkrétně není textura napojena na polygon. V aplikaci je pouze naznačeno, jak by se mohla textura vytvářet. Stejně tak nejsou dostupné rozměry pro všechny roviny, ale pouze pro první největší rovinu.

Aplikaci je možné dále vylepšovat, jak je podrobněji zmíněno v kapitole 8. Mezi hlavní body patří zajištění rovnoměrné hustoty mračna bodů, vylepšený SLAM s ukládáním barvy, automatizace při hledání rovin a poskládání ploch do modelu a jeho export.

Literatura

ALEHDAGHI, M., M. A. ESFAHANI a H. AHAD, 2015. Parallel RANSAC: Speeding up plane extraction in RGBD image sequences using GPU. In: *Computer and Knowledge Engineering (ICCKE), 2015 5th International Conference*. IEEE, s. 295-300.

BARNES, C. et al., 2009. PatchMatch: A Randomized Correspondence Algorithm for, **28** (3).

ENGELHARD, N. et al., 2011. Real-time 3D visual SLAM with a hand-held RGB-D camera. In: *Proc. of the RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum*. Vasteras, Sweden, s. 180.

HENRY, P. et al., 2012. RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. *The International Journal of Robotics Research* [Článek v periodiku], **31** (5), a2647-663.

JIN, H. et al., 2008. 3-d reconstruction of shaded objects from multiple images under unknown illumination. *International Journal of Computer Vision*. Springer, **76** (3), a2245--256.

Kinect Fusion [online]. [cit. 2016-05-11]. Dostupné z: <https://msdn.microsoft.com/en-us/library/dn188670.aspx>

LAU, D., 2013. In: *The Science Behind Kinects or Kinect 1.0 versus 2.0* [online]. 27. Listopad. 2013 [cit. 2016-Květen-16]. Dostupné z: http://www.gamasutra.com/blogs/DanielLau/20131127/205820/The_Science_Behind_Kinects_or_Kinect_10_versus_20.php

Microsoft Developer Network. *Kinect Fusion* [online]. [cit. 2016-Květen-10]. Dostupné z: <https://msdn.microsoft.com/en-us/library/dn188670.aspx>

STURM, J. et al., 2013. Technical University of Munich (TUM), Munich, Germany. In: *RANSAC, ICP, and SLAM Lecture: Visual Navigation for Flying Robots* [online].2013. Dostupné také z: http://vision.in.tum.de/_media/teaching/ss2013/visnav2013/lecture7_graph_slam.pdf

VELODYNE LIDAR, INC., 2016. In: *HDL-64E Data Sheet* [online].2016 [cit. 2016-Květen-10]. Dostupné z: http://velodynelidar.com/docs/datasheet/63-9194_Rev-D_HDL-64E_Data%20Sheet_Web.pdf

Seznam příloh

Příloha 1. DVD, na kterém se nachází zdrojové texty aplikace, nasnímané mračna bodů, plakát a tato práce v elektronické podobě.