

Slovenská technická univerzita v Bratislave
FAKULTA INFORMATIKY A INFORMAČNÝCH
TECHNOLÓGIÍ

Tímový projekt

REKONŠTRUKCIA 3D SCÉNY V2

Inžinierske dielo

Vedúci projektu:

Ing. Vanda Benešová, PhD.

Členovia tímu:

Bc. Jakub Ginter (SI)
Bc. Miroslav Haščič (SI)
Bc. Mário Hunka (SI)
Bc. Viktor Košťan (IS)
Bc. Richard Pintér (IS)

Názov tímu: 3DRecon (tím č. 11)

Web: <http://team11-16.studenti.fiit.stuba.sk/>

Kontakt: teamfiit11@gmail.com

Akademický rok: 2016/2017

Dátum odovzdania: 21. 11. 2016

OBSAH

1.	Úvod.....	1
2.	Globálne ciele pre ZS.....	1
3.	Celkový pohľad na systém.....	2
3.1	Interior3DRecon.....	4
3.2	QT_GUI.....	5
3.3	Visualization.....	6
4.	Moduly systému.....	6
4.1	Úplná segmentácia Outlayerov.....	7
4.1.1	Analýza.....	7
4.1.2	Návrh a Implementácia.....	7
4.1.3	Testovanie.....	7
4.2	Rekonštrukcia dát z Kincetu.....	8
4.2.1	Analýza.....	8
4.2.2	Návrh.....	8
4.2.3	Implementácia.....	8
4.2.4	Testovanie.....	9
4.3	Zlepšenie vizualizácie.....	9
4.3.1	Analýza.....	9
4.3.2	Návrh.....	9
4.3.3	Implementácia.....	10
4.3.4	Testovanie.....	10

1. ÚVOD

Tento dokument vznikol ako technická dokumentácia k projektu 3DRecon, ktorý sme dostali ako zadanie na tímový projekt na FIIT STU. Rekonštrukcia 3D scény je v oblasti výskumu veľmi aktuálnou témou, pretože ma pomerne veľké využitie. Aktuálne existuje veľké množstvo komerčných riešení, ktoré vedia spracovávať dáta zo senzorov skenujúcich 3d scénu. Ich hlavným problémom je, že dokážu spracovávať dáta iba z určitých senzorov (značky).

My chceme vytvoriť systém, ktorý bude spracovávať dáta v rôznom formáte bez závislosti na senzore čím umožníme rozšíriť využitie medzi širokú verejnosť. Snažíme sa umožniť naskenované dáta jednoducho spracovať a konvertovať ich do uhladeného formátu kde budú ďalej ručne spracovávané v programoch ako AutoCAD a jemu podobné. To výrazne zjednoduší prácu architektov a interiérových návrhárov ale odvetvia kde sa skenovanie objektov a priestorov tu zďaleka nekončia.

Tento dokument ponúka detailný pohľad na náš produkt a na proces jeho vývoja. Nachádzajú sa v ňom naše čiastočné i globálne ciele ale aj architektúra systému a opis fungovania.

2. GLOBÁLNE CIELE PRE ZS

Projekt sme dostali v rozpracovanej forme preto bolo prvotným cieľom celého tímu oboznámiť sa so systémom, s jeho architektúrou, triedami a celkovým spôsobom fungovania. Súčasne s týmto oboznamovaním sme museli študovať princípy segmentácie 3D scény a používané metódy a algoritmy.

Ďalšie definované ciele na semester boli:

- Refaktoring projektu
 - Zmena štruktúry kódu
 - Odstránenie nepotrebných tried a častí kódu

- Návrh novej architektúry
- Tvorba nového GUI
- Analýza možných spôsobov selekcie outliers
 - Analýza dostupných metód
 - Konzultácia v rámci tímu
 - Implementácia vybraných metód
- Získanie nových datasetov
 - Získanie senzoru od spolupracujúcej spoločnosti
 - Naskenovanie nových priestorov
 - Rekonštrukcia týchto dát
 - Testovanie na týchto dátach
- Získanie používaných datasetov
 - Nájsť na internete nové datasety, ktoré sme ešte nepoužili
- Ovládanie vizualizácie prostredníctvom 3D myši
 - Interakcia s vizualizérom pohybmi 3D myšou
- Presnejšie výsledky používaných algoritmov
- Ukladanie medzivýsledkov
 - Uloženie medzivýsledkov pre zrýchlenie výpočtov

Ďalšie úlohy sa špecifikujú postupom času na základe výstupov a výsledok predošlých úloh. Výsledky sa pravidelne konzultujú s vedúcou projektu, ktorá nám radí akým ďalším smerom sa máme pohybovať. Nakoľko tento projekt má výskumný charakter nie je možné dopredu definovať veľké množstvo konkrétnych cieľov, pretože smerovanie sa vždy odvíja od predošlých poznatkov.

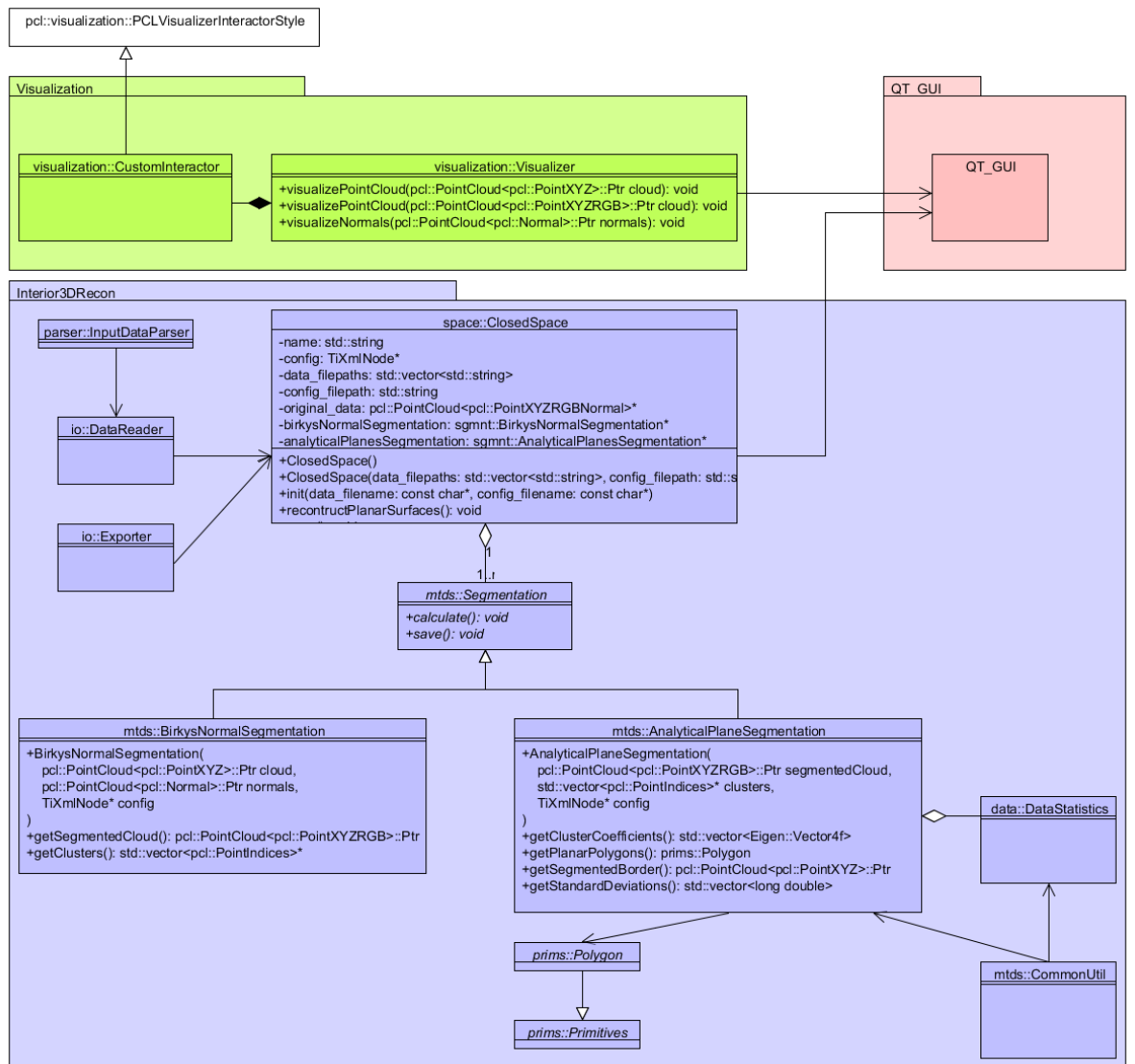
3. CELKOVÝ POHĽAD NA SYSTÉM

V tomto projekte pokračujeme po minuloročnej skupine, ktorá na projekte začínala. Preto sme museli ako prvú vec analyzovať aktuálnu formu projektu.

Celý náš tím (5 členov) musel prejsť projekt a pochopiť presné fungovanie jednotlivých tried. Nakoľko bola forma pomerne rozhádzaná a projekt obsahoval množstvo nepotrebného kódu a nepoužívaných funkcií išlo o zdĺhavý proces.

Po tejto analýze sme sa rozhodli, že pristúpime k celkovému refaktoringu tohto projektu a zmeníme jeho štruktúru. Bol navrhnutý nový architektonický model systému založený na predošlom systéme avšak s väčším ohľadom na zameniteľnosť jednotlivých častí a s oddelením používateľského rozhrania od funkcionality. V starom projekte sa časť funkcionality nachádzala aj v projekte používateľského rozhrania čo nám nepripadalo správne v rámci zaužívaných pravidiel a nechceli sme takto postavený projekt ďalej nabaľovať o ďalšiu funkcionality.

Nová architektúra je postavená na rovnakých knižniciach ako tá predošlá. Jediný rozdiel je v organizácii projektu. Analýzu možných alternatívnych riešení sme nerobili, pretože sme to nepovažovali za rozumné a časovo efektívne. Aj keby sme našli lepšie riešenia na problematiku knižníc v obore 3D rekonštrukcie bolo pri by časovo nerentabilné prerábať celý projekt. Preto sme sa zamerali na použitie PCL teda PointCloudLibrary, ktorá ponúka rozsiahlu funkcionality v tejto oblasti.



Ako zo zadania vyplýva museli sme dbať najmä na dátovo-formátovú nezávislosť aby bola aplikácia skutočne použiteľná na všeobecnej úrovni. S tým je zviazaná aj dátovo-typová nezávislosť. Zlepšili sme zameniteľnosť a flexibilitu projektu. Z dôvodu náročných výpočtov sme výrazne dbali aj na zníženie pamäťovej náročnosti tejto aplikácie.

3.1 INTERIOR3DRECON

Interior3DRecon predstavuje hlavnú funkcionálnosť nášho projektu. Obsahuje triedy na segmentáciu, import a export 3D dát.

Jej aplikačné rozhranie predstavuje trieda *ClosedSpace*, ktorá v sebe zoskupuje načítané pôvodne 3D dáta, konfiguračné parametre a metódy na výpočet/prístup ku segmentovaným dátam. Je navrhnutá tak, aby jej inštancia obsahovala segmentované dáta pre práve jednu kombináciu pôvodných dát a konfiguračných parametrov. Je to preto, aby sa zabezpečila lepšia viditeľnosť a umožňuje to jednoduchšie a prehľadnejšie ukladanie dát inštancie.

Jednotlivú segmentáciu 3D dát vykonávajú podtriedy triedy *Segmentation*. Podobne ako trieda *ClosedSpace* sú navrhnuté tak, aby si po inicializácii a vykonaní *calculate* metódy uchovali vypočítané dáta a v prípade, že bude potrebné segmentovať iné dáta alebo použiť inú konfiguráciu vytvorí sa nová inštancia. Umožní to uchovať si výsledky tej istej segmentačnej metódy s rôznymi konfiguráciami, čo sa môže hodiť v prípade, že sú potrebné tieto rozličné výsledky na ďalšie spracovanie.

V súčasnom projekte sú implementované dve triedy, ktoré dedia od triedy *Segmentation* - *BirkysNormalSegmentation* a *AnalyticalPlanesSegmentation*. *BirkysNormalSegmentation* segmentuje point cloud na základe odhadnutých normál bodov. Tieto segmentované dáta berie na vstupe *AnalyticalPlanesSegmentation*, ktorá ďalej segmentuje 3D scénu na planárne polygóny pomocou fitovania rovín cez clustre bodov zo vstupu.

Triedy *CommonUtil* a *DataStatistics* obsahujú podporné metódy pre podtriedy triedy *Segmentation*.

Metódy a funkcie na import a export dát sú obsiahnuté v triedach *Exporter* a *DataReader* pričom *DataReader* používa triedu *InputDataParser*, ktorá parsuje ľubovoľné ASCII dáta do point cloudu v XYZ koordinačnom systéme.

3.2 QT_GUI

Pomocou tohto modulu je spúšťaný celý projekt. Zvyšné moduly sú kompilované do knižníc, ktoré *QT_GUI* používa. Jeho jediná funkcionálna je pritom používateľské okno, ktoré umožňuje volať funkcionálnosť modulov *Interior3DRecon* a *Visualization*.

Jeho jediná rovnomenná trieda *QT_GUI* obsahuje premennú typu *ClosedSpace*, ktorá poskytuje aplikačné rozhranie modulu *Interior3DRecon* a premennú typu *Visualizer*, ktorá poskytuje aplikačné rozhranie modulu *Visualization*. Po vybratí súborov s 3D dátami a konfiguračného súboru sa zavolá inicializačná metóda triedy *ClosedSpace* a sprístupnia sa tak tlačidlá, ktoré volajú jej zvyšné metódy a metódy *Vizualizéru*.

Na vytvorenie používateľského rozhrania sme použili populárny softvér *QT*. Umožnil nám pomerne rýchlo vytvoriť grafické používateľské rozhranie, a taktiež jeho podpora cross-platformového vývoja sa môže do budúcnosti hodiť, v prípade že sa bude implementovať rekonštrukcia dát z MS Kinect.

3.3 VISUALIZATION

Vizualizáciu sme dokázali kompletne oddeliť od používateľského rozhrania a pomocné výpočtové hodnoty budeme posielat' z GUI ako parametre, takže v GUI bude iba volanie funkcie. Vizualizér pracuje s inštanciou `ClosedSpace`, ktorá je hlavnou inštanciou celého programu a všetky operácie sa robia nad ňou.

Vizualizácia prebieha s využitím metód knižnice PCL. Máme vytvorený vlastný spôsob interakcie ktorý sme prebrali od minuloročného tímu ale je potrebné prepracovať ho. Interakcia je príliš neintuitívna a nepraktická. V budúcnosti by sme chceli túto interakciu podporiť 3D myšou.

4. MODULY SYSTÉMU

Jednotlivé moduly systému sme previazali po predošlom tíme. Tie sú podrobne zdokumentované na ich prezentačnej stránke¹. Medzi tie časti, ktoré sme si osvojili a zahrnuli do novej štruktúry systému, ktorý má vzísť z refaktoringu patria:

- **BirkysNormalSegmenation** - segmentácia point cloud do clusterov na základe vypočítaných normál bodov.
- **AnalyticatPlaneReconstruction** - po novom *AnalyticalPlaneSegmentation* keďže sa jedná o segmentáciu a nie rekonštrukciu je modul, ktorý má po refaktoringu segmentovať 3D scénu na planárne polygóny. Tieto polygóny sa neskôr použijú na segmentáciu stien miestností a menej členitých objektov ako napríklad stôl alebo dvere.
- **CustomInteractor** - trieda na ovládanie PCL vizualizačného okna. Bude upravená aby poskytovala intuitívnejšie ovládanie.
- **Načítavanie konfiguračných údajov z XML** - v novej verzii projektu je taktiež použitý opensource TinyXML na načítavanie konfiguračných údajov
- **Export segmentovaných dát** - trieda *Exporter*, ktorá bude doplnená o metódy na export dát, ktoré sú potrebné na to aby bolo možné uložiť inštanciu triedy *ClosedSpace*.

¹ Prezentačná stránka tímu RED

link: <http://labss2.fkit.stuba.sk/TeamProject/2015/team05is-si/#final-product>

Novú funkcionálnosť zatiaľ náš systém neponúka. Vzhľadom na globálne ciele (uvedené vyššie) sme identifikovali 3 problémy, ktoré budú predmetom riešenia nášho tímu:

1. Úplná segmentácia outlayerov
2. Rekonštrukcia dát z kinectu
3. Zlepšenie vizualizácie

4.1 ÚPLNÁ SEGMENTÁCIA OUTLAYEROV

4.1.1 ANALÝZA

Analýza požiadaviek prebieha počas spoločných stretnutí s product ownerom. Na základe diskusií sme identifikovali problém s tzv. *outlayer-mi* - výčnelkami (kľučka, svetlo apod.). Problém úzko súvisí so segmentáciou polygónov, na ktorú je využitá metóda opísaná vyššie (*AnalyticalPlaneReconstruction*). Odstránením tohto problému je možné značne vylepšiť presnosť segmentácie polygónov, ktoré sú kľúčové pri identifikácii stien či objektov.

Konkrétne, segmentácia polygónov prebieha v týchto krokoch. Z troch bodov sa definuje rovina, táto rovina sa preloží cez vytvorené segmenty. Pri evaluácii sa zistí koľko percent zo segmentovaných clusterov sa pretína s definovanou rovinou. Samozrejme, je potrebná určitá odchýlka, ktorá určuje kedy bod patrí a kedy nepatrí do roviny. Odchýlka je určená vzhľadom na šum, ktorý vstupné dáta obsahujú. Náš problém spočíva teda v tom, ako oddeliť šum od reálnych objektov. Napríklad, sme si vedomý toho aký nepresný je náš hardware. Avšak, objekty, ktoré sú relatívne malé vzhľadom na snímanú miestnosť - napr. kľučka - sa nám javia ako šum - čo ale nie je pravda. V prípade, že by sme tieto objekty vedeli identifikovať a nejakým spôsobom ich zo scény odstrániť (vysegmentovať) vieme zaručiť lepšiu identifikáciu polygónov.

4.1.2 NÁVRH A IMPLEMENTÁCIA

Tento problém je zatiaľ v procese návrhu. Jeho implementácia bude prebiehať v rámci ďalšieho šprintu.

4.1.3 TESTOVANIE

Evaluácia výsledkov riešenia tohto problému bude prebiehať nasledovným spôsobom. Je potrebné vždy poznať aj výsledky starej metódy, preto je potrebné

si ich niekam uložiť. Pri tej istej konfigurácii, identifikujeme polygóny bez toho aby sme odstránili outlier-i a uložíme si výsledky o úspešnosti (koľko percent bodov zo segmentovaných častí patrilo do roviny). Následne pripojíme našu metódu segmentácie outlierov a znovu spustíme proces identifikácie polygónov. Výsledky vieme teda ľahko pomocou tejto metriky porovnať. Ak dosiahneme značne lepšie výsledky metóda bude integrovaná do projektu.

Je potrebné pôvodný výstup kontrolovať aj ručne - pomocou vizualizéru, a teda porovnať výstup pôvodných metód v podobe point cloud s výstupom nových metód segmentácie a identifikácie outlier-ov.

4.2 REKONŠTRUKCIA DÁT Z KINCETU

4.2.1 ANALÝZA

Získavanie testovacích dát - voľne dostupné datasety na internete, datasety od firmy *Photoneo s.r.o.* - obmedzuje vývojový tím pri testovaní nových výskumných metódach. Ak by sme chceli dataset šitý na mieru, je časovo náročné (možno aj nemožné) takýto dataset zabezpečiť. Vzhľadom nato, sme na spoločných stretnutiach prišli k záveru, že je vhodné implementovať vlastnú rekonštrukciu dát.

4.2.2 NÁVRH

Na fakulte máme dostupný *Kinect* pomocou, ktorého je možné nasnímať 3D dáta. Pre zvýšenie presnosti sme navrhli zber dát z viacerých uhl'ov pohľadu. To znamená, že je potrebné tieto dáta nakoniec spojiť dokopy, aby predstavovali jeden použiteľný dataset.

4.2.3 IMPLEMENTÁCIA

Implementácia bude prebiehať v rámci 4. šprintu.

4.2.4 TESTOVANIE

Testovanie riešenia tohto problému bude prebiehať manuálne s použitím vizualizéru. V tomto prípade je to najlepšie riešenie, ak vieme čo sme nasnímali, vieme si to zobrazit' a odhadnúť podobnosť s realitou.

Pri jednotlivých metódach rekonštrukcie je vhodné ich neskôr vyhodnocovať aj kvantitatívne a navzájom porovnávať presnosť s akou boli dáta namerané.

4.3 ZLEPŠENIE VIZUALIZÁCIE

4.3.1 ANALÝZA

V rámci nášho projektu je vizualizácia podstatná časť. Správnosť metód je v mnohých prípadoch overovaná vizuálne - dá sa pohľadom povedať či má zmysel sa metóde venovať alebo nie. Práve preto je potrebné mať funkčný a intuitívne použiteľný vizualizér.

Ďalším dôvodom pre zlepšenie vizualizéru je naša tímová prezentácia. V rámci súťaže *TP CUP*, je vhodné mať lepší vizualizér. Pri prezentácii je práve ten náš jediný výstup, ktorý vníma pozorovateľ, ktorý nie je úplne zainteresovaný v princípoch rekonštrukcie 3D scény.

4.3.2 NÁVRH

V našej práci sa zameriavame hlavne na implementáciu intuitívneho pohybu pomocou myši. Ďalšou navrhnutou metódou pohybu je ovládanie pomocou 3D myši. Pre správnu a hlavne intuitívnu implementáciu pohybu v priestore je nutné navrhnuť interakciu s vizualizérom tak aby bola priateľská pre používateľa a ľahko naučiteľná. Samotné dáta, ktoré vizualizujeme sú pripravené na prezentáciu implementovanými metódami segmentácie.

Nový vizualizér, ktorý je založený na vstavanom vizualizéri PCL knižnice, spočíva prevažne v implementácii vlastného ovládania (správne reakcie na pohyb myšou, správne interpretované klávesové vstupy do pohybov kamery vo vizualizéri) pre vizualizér a vyvinutie metód na spracovanie vstupov z 3D myši. Okrem tohto hlavného problému vizualizéru sa tiež venujeme správne vkladaniu dát do priestoru v ktorom sa kamera pohybuje. Štartovacia pozícia kamery a jej orientácie v priestore je tiež veľmi dôležitá pre zorientovanie používateľa.

4.3.3 IMPLEMENTÁCIA

Implementácia vizualizéru je riešená prostredníctvom funkcií z PCL knižnice. Jedna funkcia vytvorí okno s parametrami ako farba pozadia, koordináty a inicializačný bod pre kameru. Druhá funkcia inicializuje interakciu pre toto okno. Interakcia je sprostredkovaná našou triedou kde máme definovaný spôsob interakcie. Celá táto funkcionálnosť je obsiahnutá v jednej metóde, ktorá sa zavolá pri kliknutí na tlačidlo vizualizovať v GUI. Metóda je volaná s parametrom typu PointCloud, ktorý je získaný z inštancie ClosedSpace. Ten sa v metóde vloží do vytvoreného okna kde je možné s ním pracovať kým nebude okno zatvorené alebo zastavená vizualizácia. Vizualizér v tejto fáze zobrazuje iba originálne nezrekonštruované dáta. Keď bude implementovaná rekonštrukcia bude sa ako parameter posielat' aj typ dát a na základe toho sa naplní okno vizualizéru. Samotný spôsob naplnenia je vytvorený, stačí už iba implementovať pozmenené volanie. Dôvod prečo je toto potrebné je ten, že funkcie PCL knižnice potrebujú poznať charakter dát, ktoré vizualizujú.

4.3.4 TESTOVANIE

Overenie správnosti riešenia pre vizualizér sa bude testovať hlavne pomocou UX experimentu, do ktorého zapojíme členov tímu, ale aj neskúseného používateľa, ktorý nám môže priniesť užitočné návrhy na spríjemnenie používania vizualizéru.