1. Základné informácie

Systém kontroly kvality je implementovaný v ROS Melodic a napísaný v jazyku C++. Meno ROS packagu systému kontroly kvality je quality_inspection, pričom package má nasledujúce súčasti:

Priečinky:

- action obsahuje .action súbory, ktoré špecifikujú požité ROS akcie
- config obsahuje .yaml súbory, ktoré obsahujú vstupné parameter pre systém
- non_ros- obsahuje súbory, ktoré sa netýkajú ROSu (Dokumentácia, súbory v robote, vstupné súbory)
- include obsahuje hlavičkové .h súbory
- launch obsahuje .launch súbory používané na spustenie systému
- src obsahuje zdrojové súbory .cpp
- srv obsahuje .srv súbory, ktoré špecifikujú použité ROS servisy
- testClientInPC obsahuje program simulujúci robota komunikáciu s robotom*

Súbory:

- CMakelist.txt potrebne pre build packagu
- Package.xml potrebne pre build packagu
- README.txt

Jednotlivé moduly systému kontroly kvality sú realizované ako samostatné ROS node. Každý modul je realizovaný ako samostatná class pričom zdrojový kód je delený nasledovne (predpokladajme modul s názvom modul):

```
modul_node.cpp – zdrojový kód daného modulu
modul_node.h - .h file pre modul_node.cpp
```

Rozsiahlejšie moduly sú ale rozdelené na časť obsahujúcu realizáciu architektúry, a na časť obsahujúcu implementáciu spracovania dát.

modul_node.cpp – obsahuje realizáciu architektúry daného modulu (rozbehnutie vlákien, medziprocesová komunikácia, volania funkcií na spracovanie, ...)

```
modul_node.h - .h file pre modulNode.cpp
```

modul.cpp – obsahuje implementáciu spracovania dát (transformácia, ICP, filtrácia, matematické prevody, vytvorenie projekcií)

modul.h - .h file pre modul.cpp, avšak v prípade že je nejaký krok spracovania dát implementovaný ako template, tak je v tomto súbore

2. Spustenie systému

Systém je možné spustiť v dvoch režimoch. Prvým režimom je online režim pre prácu s reálnymi zariadeniami (robot, skener). Druhý režim je offline režim, kedy je proces skenovania simulovaný.

a) Online režim

sudo PhoXiControl
quality_inspection.launch

b) Offline režim

quality_inspection_test.launch

Samozrejme, na spustenie sa nemusí využiť launch file, ale potrebné nody sa môžu spustit aj samostatne cez rosrun. Pokial je systém spustený využitím roslaunch, sú jednotlivé nody spustené v samostatných termináloch. Keď z nejakého dôvodu padne nejaká node, automaticky sa aj zavrie aj terminál v ktorom je spustená. Aby bolo vidieť errorovú hlášku je preto pri debugovani vhodné v .launch file pre problémovú node vymazať launch-prefix="gnome-terminal --command", pričom potom bude daná node spustená v terminále, v ktorom je spúšťaný kde sa spustí roslaunch.

3. Pokyny pred spustením

Nainštalovať ROS Melodic. Odporúčam využiť plnú inštaláciu ROS podľa pokynov http://wiki.ros.org/melodic/Installation/Ubuntu. Treba mať nainštalované (minimálne) PCL 1.8.1, OpenCV 2.6.0, TinyXML, Eigen, atd...

Nainštalovať PhoXi Control a stiahnuť package https://github.com/photoneo/phoxi_camera, pričom na linku sú uvedené bližšie pokyny. Nezabudnúť nastaviť parametre v phoxi_camera/config/phoxi_camera.yaml. Pozor phoxi_camera node je potrebné spúšťať ako. roslaunch phoxi_camera phoxi_camera.launch (rosrun phoxi_camera phoxi_camera nestačí, lebo treba načítať parametre z .yaml na param server).

Rozbehať robota a jeho sieťové prepojenie s PC. Nainštalovať EthernetKRL do robota a vložiť doňho súbory z quality_inspection/non_ros/filesInRobot. Na kuka robot je heslo "kuka". Bližší popis obsahuje rozbehania robota obsahuje dokumentácia non_ros/ Rýchla príručka na rozbehanie robota.pdf.

HLAVNE pri prvom spustení mať pre istotu nastavenú malú rýchlosť robota (program override napr. 10%), aby sa nediali viac než nemilé veci.

Aktualizovať config/quality_inspection.yaml. Nastaviť cesty k vstupným a výstupným priečinkom. Do vstupného priečinku (v prípade offline režimu aj do vstupného priečinku pre simuláciu) dať potrebné súbory. Všetky vstupné .txt súbory majú mať preddefinovaný formát, pričom ukážka vstupných súborov (okrem vstupov offline režimu, lebo tie sú veľmi veľké) je v priečinku no_ros/input_data_quality_inspection. V prípade offline režimu je pre vykonávanie simulácie potrebné mať pre každý sken jednu .bmp fotku a jeden .pcd súbor mračna bodov. Je ich potrebné pomenovať nasledovne (samozrejme dá sa to zmeniť v scanning_simulation_node.cpp):

- "imgX.bmp", kde X je číslo skenu. Napr. "img0.bmp".
- "pointCloud_originalX.pcd", kde X je číslo skenu. Napr. "pointCloud_original0.pcd"

4. Moduly systému

Detailný opis modulov systému sa nachádza v diplomovej práci. Moduly systému sú v quality_inspection package implementované nasledovne.

Komunikačný modul robota:

```
node robot_communication

class RobotCommunicationNode

robot_communication_node.cpp

robot_communication_node.h
```

Hlavný modul:

```
node main_control

class MainControlNode

main_control_node.cpp

main_control_node.h
```

Modul spracovania skenu:

Časť architektúry:

node scan_processing

class ScanProcessingNode

scan_processing_node.cpp

scan_processing_node.h

Časť algoritmov spracovania:

scan_processing.cpp

scan_processing.h

Modul vytvorenia projekcií:

Časť architektúry:

node projections_creation

class Projections

projections_creation_node.cpp

projections_creation_node.h

Časť algoritmov spracovania:

projections_creation.h

5. Ďalšie súbory

Okrem implementácií uvedených modulov sa v /src a /include priečinku nachádzajú aj ďalšie súbory:

- my_semaphore.h implementácia použitého semaforu (nakoľko Melodic podporuje max C++ 14)
- auxiliaries.h header file, kde sa nachádzajú templaty pomocných funkcii, ktoré sú využívané viacerými modulmi
- my_transform.hpp implementácia používanej transformácie mračna bodov získana z PCL
 1.10 (pretože Melodic podporuje PCL 1.8, kde transformacia trva 33 krát dlhšie). Pre využitie rýchlych transformácií treba použiť "newPcl::transformPointCloudWithNormals" namiesto "pcl::transformPointCloudWithNormals".
- transformCAD.cpp zdrojový kód programu, ktorý zabezpečuje:
 - Transformáciu jednovrstvového mračna bodov vytvoreného z CAD modelu do začiatku súradnicovej sústavy. Na základe získanej transformačnej matice bolo dvojvrstvové mračno bodov bolo zatiaľ transformované v CloudCompare (lebo je to .ply).
 - Generáciu .txt súboru obsahujúceho XYZ pozície nitov. Pozície sú generované na základe indexov vertices dvojvrstvového CAD modelu, ktoré boli v CloudCompare nájdené ako tie čo odpovedajú stredu jednotlivých nitov.
- scanning_simulation_node.cpp zdrojový kód ROS node "scanning_simulation", ktorá simuluje process skenovania v PC. Simuluje sa v tomto prípade len príchod skenovacích dát z jednotlivých skenovacích polôh na príslušné ROS topic. Dáta na zverejnenie sú načítavané z príslušného priečinku (ten je definovaný spolu s ostatnými vstupmi v quality_inspection.yaml).
- scanning_simulation_node.h .h file pre scanning_simulation_node.cpp