DAVIDE ZANIN – 1035601

ELABORAZIONE DI DATI TRIDIMENSIONALI

RELAZIONE HOMEWORK 2

SOMMARIO

OBIETTIVO

Realizzare un programma che analizzi una point cloud rappresentante la scheda da verificare in modo da controllare se la posizione del cavo nero è corretta.

DESCRIZIONE DELLA PROCEDURA SEGUITA

Per facilitare l'individuazione dei pioli e del cavo si è deciso di scegliere una point cloud di riferimento e di registrare la point cloud da analizzare sulla point cloud scelta. In questo modo è possibile sapere in quale posizione si troveranno gli oggetti di interesse in modo sufficientemente preciso. Questa scelta ha il vantaggio di facilitare l'individuazione del cavo e dei pioli ma ha lo svantaggio di rendere necessario il processo di registrazione.

Dopo che la point cloud è stata registrata si passa all'individuazione dei pioli; viene quindi selezionata una specifica zona della cloud e si procede con la clusterizzazione. Ogni cluster viene analizzato e, conscendo la posizione in cui il piolo destro e quello sinistro si trovano, si stabilisce se è un piolo o no. L'estrazione dei punti del cavo avviene in modo analogo.

Infine si contano quanti punti del cavo si trovano nel volume compreso fra i due pioli e si stabilisce se la posizione del cavo è corretta oppure no.

Quella appena descritta è la procedura generale che è stata seguita, di seguito si trova la descrizione con maggior dettaglio.

CARICAMENTO DELLA POINT CLOUD DI RIFERIMENTO

Come cloud di riferimento è stata scelta "correct02". Il caricamento di questa point cloud si compone dei seguenti passi:

1. Caricamento dei punti da file e spostamento della cloud in modo che il centroide si trovi nel punto (0, 0, 0);
2. Rimozione del tavolo. Questa operazione è molto molto semplice, infatti è sufficiente togliere dalla cloud tutti i punti con coordinata z minore di 0, e indipendente dalla cloud scelta; questo grazie allo

spostamento eseguito al passo precedente, inoltre la rimozione del tavolo rende molto più leggere la cloud;
3. Calcolo della versione filtrata con un filtro voxel;
4. Calcolo dei keypoint SIFT e delle feature FPFH sulla cloud calcolata al punto precedente.

La cloud filtrata, quella dei keypoint e quella delle feature vengono salvate perché successivamente verranno usate per registrare le point cloud da analizzare.

ANALISI DI UNA POINT CLOUD

Dopo aver caricato la cloud di riferimento, vengono analizzate una alla volta le cloud che sono state passate come parametro al programma.

Anche in questo caso si procede con il caricamento della point cloud, lo spostamento del centroide in (0, 0, 0) e la rimozione del tavolo, in modo analogo a quanto fatto per la cloud di riferimento. A questo punto viene eseguita la registrazione.

REGISTRAZIONE DI UNA POINT CLOUD

Per prima cosa viene applicato un filtro voxel alla cloud da registrare, in questo modo il processo di registrazione viene eseguito più velocemente.

Vengono quindi calcolati i keypoint SIFT e le feature FPFH e utilizzando la classe `SampleConsensusInitialAlignment` viene effettuato un primo allineamento della cloud. Questo allineamento viene calcolato basandosi sui keypoint e le feature ad essi associati.

Per rifinire l'allineamento viene usato l'algoritmo iterative closest point che però ha bisogno di tutti i punti della cloud, per questo motivo la cloud filtrata con il voxel viene spostata secondo quanto appena calcolato con `SampleConsensusInitialAlignment` e quest'ultima cloud viene fornita come input per la rifinitura della registrazione.

Ora che si dispone della traslazione finale che permette di spostare la point cloud su quella di riferimento in modo che siano allineate, questa viene applica alla cloud a cui non è stato applicato il

filtro voxel in modo da disporre di tutti i punti per le elaborazioni successeve.

INDIVIDUAZIONE DEI PIOLI

Dopo la registrazione vengono individuati i pioli, viene quindi creata una point cloud che conterrà solo i punti che si suppone essere appartenere ai pioli. Per fare ciò la point cloud registrata viene filtrata in modo da tenere solo i punti che si trovano in certo volume, volume in cui sicuramente saranno presenti i pioli (se visibili).

Per estrarre i pioli viene eseguita una clusterizzazione sulla cloud appena filtrata; ciascun cluster viene poi analizzato e se corrispondente ad un piolo, i punti vengono colorati di verde e aggiunti alla point cloud che rappresenta i pioli.

La tecnica usata per capire se un cluster è un piolo è la seguente; vengono conteggiati i punti che stanno in una determinata zona (quella occupata da un piolo) e se questo numero supera una certa frazione dei punti totali del cluster allora viene classificato come un piolo.

INDIVIDUAZIONE DEL CAVO E VERIFICA DELLA CORRETTEZZA

Terminata l'individuazione dei pioli viene individuato il cavo, la procedura utilizzata è la stessa di quella usata per i pioli descritta nel paragrafo precedente, cambiano solo i parametri.

Per verificare se il cavo è nella posizione corretta viene contato il numero di punti del cavo che stanno nel volume individuato dai due pioli. Se questo numero è maggiore di 55 allora il cavo è posizionato correttamente.

METODI E PARAMETRI UTILIZZATI

Per calcolare il centroide di una point cloud e spostarla in modo che esso si trovi alle coordinate (0, 0, 0) sono state utilizzate le funzioni `compute3DCentroid` e `demeanPointCloud`, mentre per eliminare il tavolo è stata usata la classe `ConditionalRemoval` definendo un'apposita condizione (z minore di 0) attraverso la classe `ConditionAnd`.

Prima di effettuare la registrazione viene applicato un filtro voxel con leaf size pari a 0.9, per applicarlo è stata usata la classe `VoxelGrid`. Il valore è stato scelto dopo una serie di prove in modo da garantire una buona velocità nella registrazione ma anche un risultato sufficientemente preciso.

Come keypoint per eseguire la registrazione sono stati usati i keypoint SIFT; la scelta è dovuta principalmente al fatto che questi keypoint sono scala invarianti e fornivano risultati migliori rispetto agli altri. Per il calcolo sono è stata usata la classe `SIFTKeypoint` impostando i parametri come segue:

- `sift.setScales(0.05, 3, 5);`
- `sift.setMinimumContrast(0.5);`

I parametri sono stati trovati partendo da alcuni parametri forniti in un tutorial successivamente modificandoli in modo da ottenere buoni risultati ma in tempi più rapidi.

Per calcolare le feature FPFH è necessario prima calcolare le normali; per fare ciò è stata usata la classe `NormalEstimationOMP` che permette di calcolare le normali sfruttando tutti i core del processore e impostando il raggio della sfera usata per l'individuazione dei vicini di un punto poi usati per il calcolo della stima, pari a 6. Per le feature FPFH invece, è stata usata la classe `FPFHEstimationOMP`, che anche in questo caso permette di utilizzare tutti i core, e il raggio della sfera è stato impostato a 6.75.

A questo punto per effettuare l'allineamento iniziale è stata usata la classe `SampleConsensusInitialAlignment` fornendo come input le cloud dei keypoint e le cloud delle feature FPFH inoltre i parametri sono stati impostanti come segue:

- `sac.setMinSampleDistance(35);`
- `sac.setMaximumIterations(400);`
- `sac.setTransformationEpsilon(0.001);`

Anche in questo caso per i parametri sono stati ottenuti dopo una serie di prove, in modo che permettessero di ottenere buoni risultati ma nel minor tempo possibile.

Per completare l'allineamento è stato usato l'algoritmo iterative closest point e in particolare la classe `IterativeClosestPoint`. I parametri usati in questo caso sono i seguenti:

- `icp.setMaxCorrespondenceDistance(40);`
- `icp.setMaximumIterations(50);`
- `icp.setTransformationEpsilon(0.01);`
- `icp.setRANSACOutlierRejectionThreshold(50);`

L'ultimo parametro è stato ottenuto adattando il valore di default alle point cloud fornite, infatti solitamente pcl considera le posizioni dei punti come espresse in metri mentre nel nostro caso sono espresse in millimetri.

Per trovare i pioli i parametri usati nella clusterizzazione e la classe `EuclideanClusterExtraction` usando i seguenti parametri:

- `ec.setClusterTolerance(0.6);`
- `ec.setMinClusterSize(3);`
- `ec.setMaxClusterSize(75);`

Anche per individuare il cavo è stata utilizzata la stessa classe ma in questo caso i parametri sono:

- `ec.setClusterTolerance(0.7);`
- `ec.setMinClusterSize(700);`
- `ec.setMaxClusterSize(3000);`

Le differenze tra i parametri usati per i pioli e per il cavo sono dovute principalmente al fatto che un piolo è composto da un piccolissimo numero di punti mentre il cavo viene solitamente riconosciuto come un unico cluster

RISULTATI OTTENUTI

Tutte le point cloud vengono correttamente identificate.

Point cloud	Risultato
cable02	NO
cable02	NO
correct01	SI
correct01	SI

Tabella 1: risultati ottenuti.

IMMAGINI DELLE POINT CLOUD OTTENUTE

Per alcune delle point cloud la rilevazione del cavo non è perfetta, vengono infatti incluse nelle estremità parti che non appartengono al cavo. Siccome questa incongruenza non influenza la valutazione del corretto posizionamento del cavo fra i pioli e non risulta essere eccessivo, non è stato corretto. Il rilevamento dei pioli invece risulta sempre perfetto.

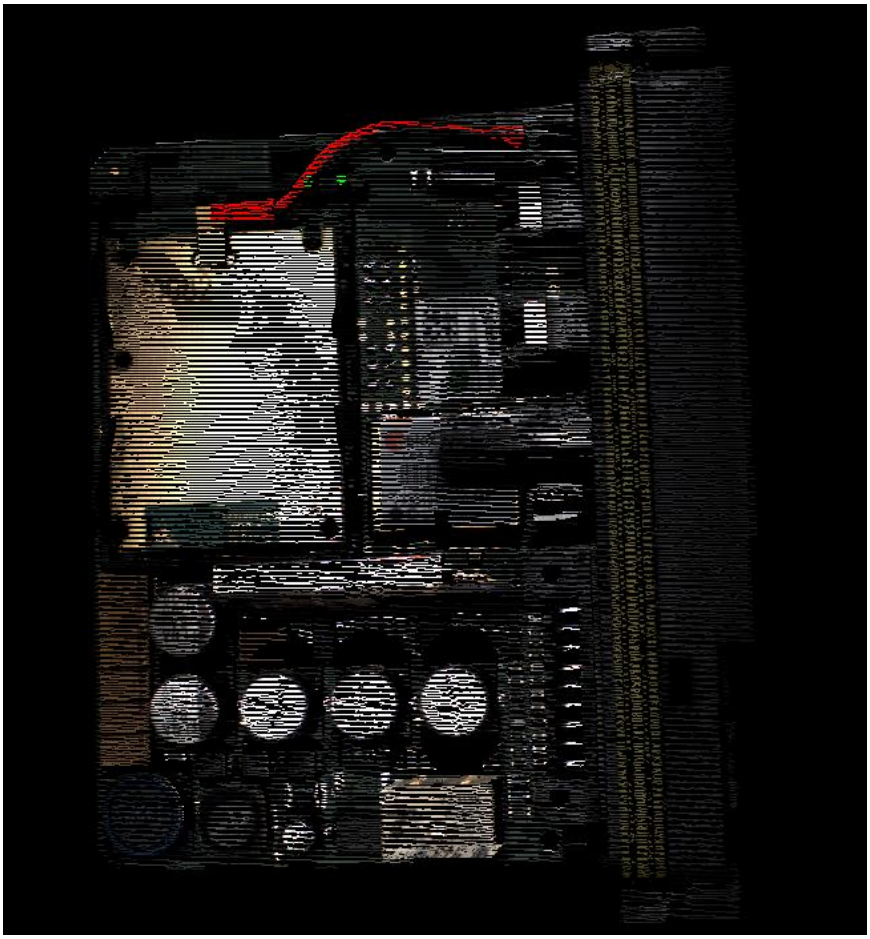

Figura 1: point cloud “cable02”.

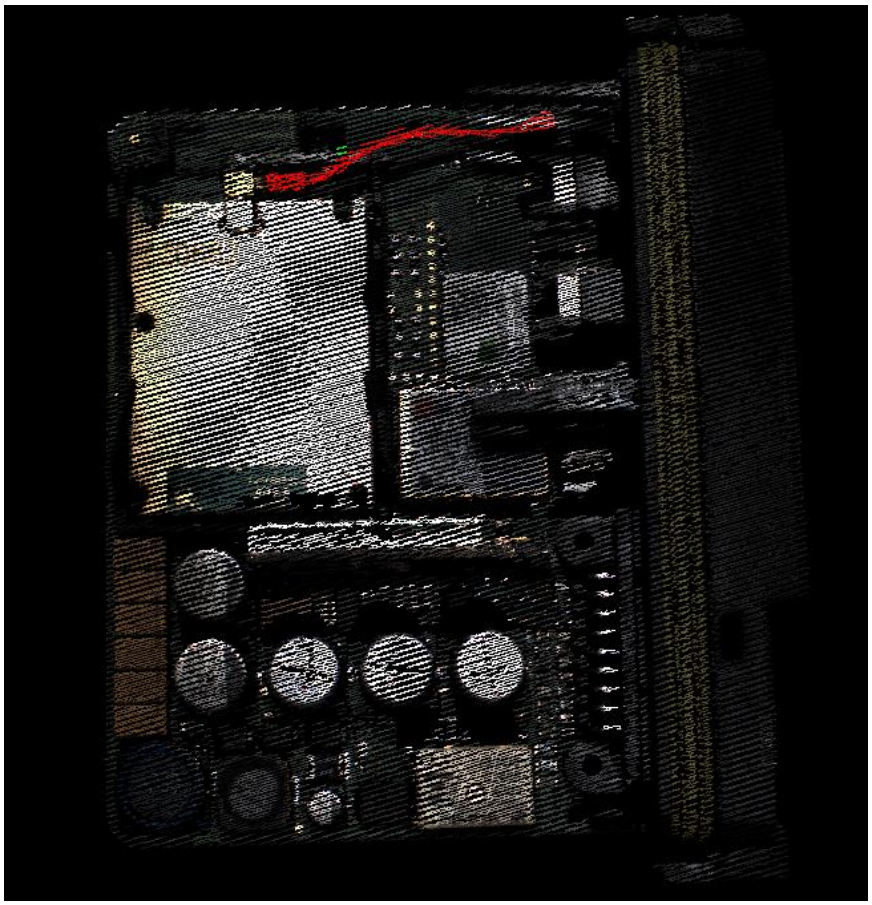

Figura 2: point cloud “cable03”.

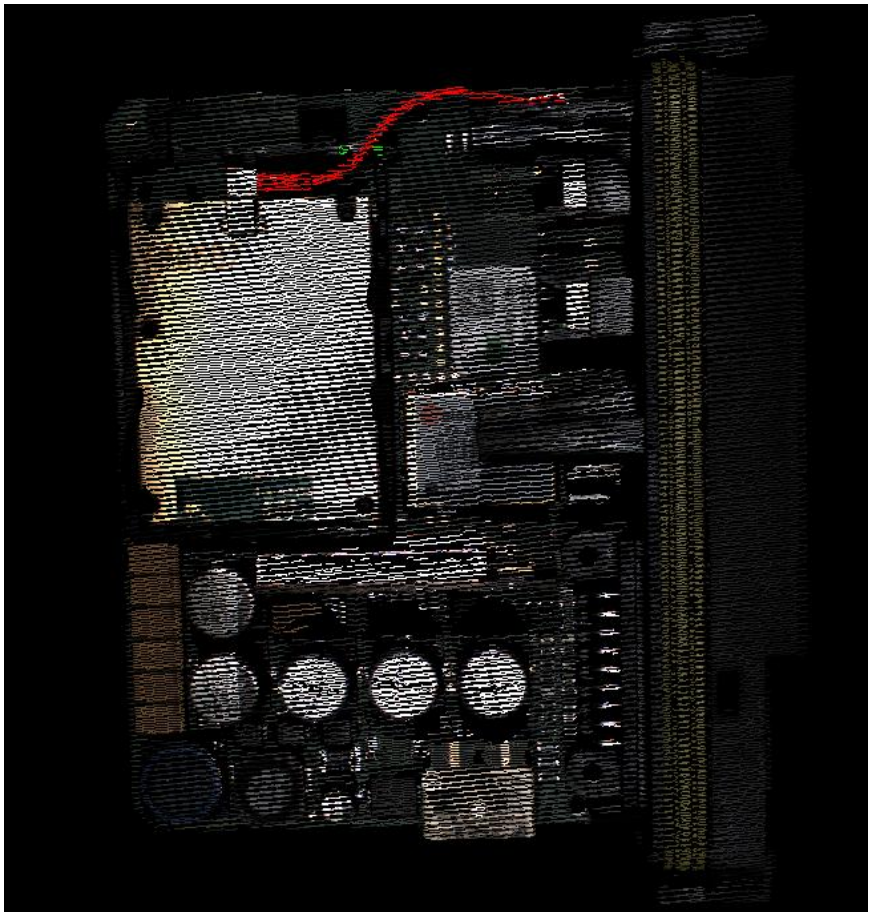

Figura 3: point cloud “correct01”.

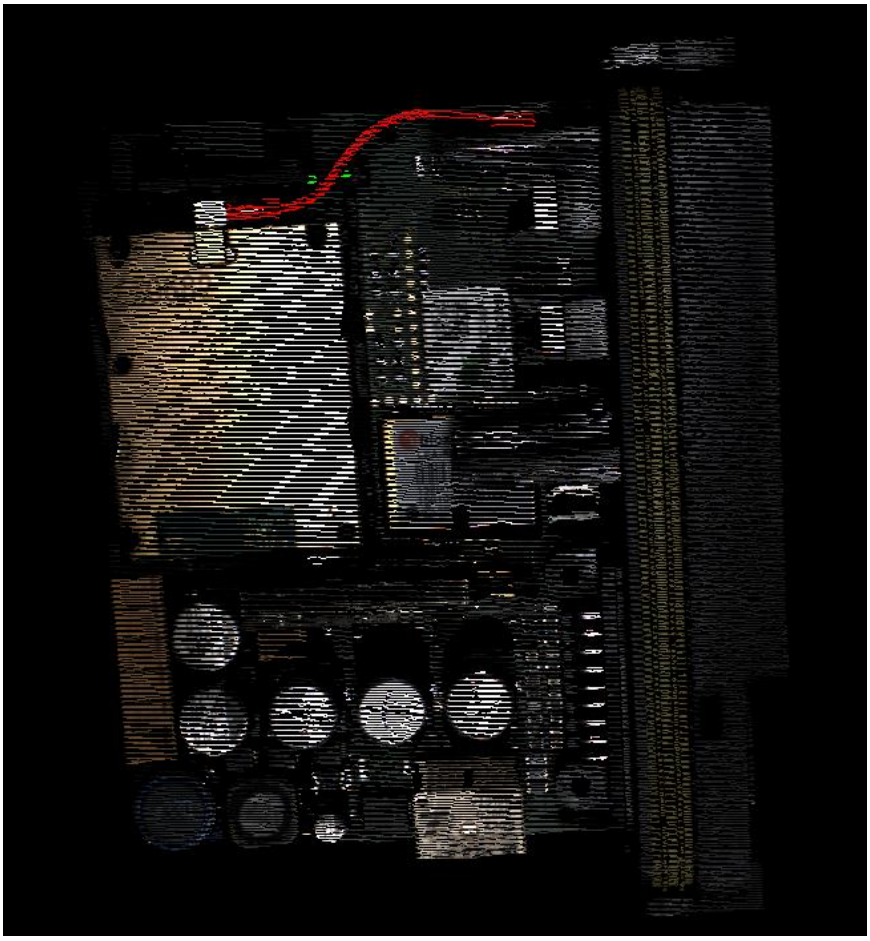

Figura 4: point cloud “correct02”.