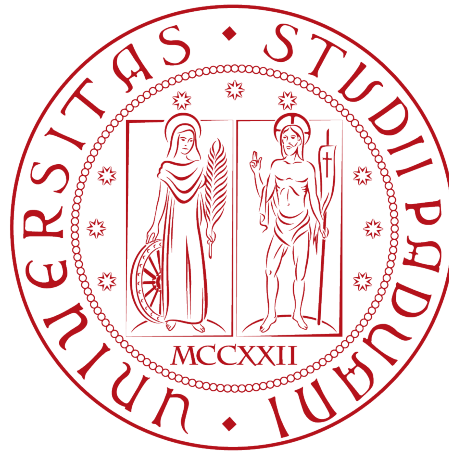


ntty]ntty/global

Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA

CORSO DI LAUREA IN INFORMATICA



Project Tango e PCL: un'applicazione
pratica alle 'Nuvole di Punti'

Tesi di laurea

Relatore

Prof. Ombretta Gaggi

Laureando

Davide Bortot

ANNO ACCADEMICO 2015-2016

Sommario

Il presente documento descrive il lavoro svolto durante il periodo di stage, della durata di circa trecentoventi ore, dal laureando Davide Bortot presso l'azienda VIC S.r.l. Obiettivo del tirocinio era lo sviluppo di un'applicazione prototipale per il tablet Project Tango per l'acquisizione di scansioni di oggetti reali sotto forma di 'Nuvola di Punti'.

In primo luogo era richiesto il filtraggio e l'elaborazione dei dati acquisiti in modo da isolare l'oggetto scansionato, e successivamente di creare una mesh tridimensionale a partire dai punti rimanenti. Scopo finale era calcolare il volume dell'oggetto tridimensionale così ottenuto.

Ringraziamenti

Innanzitutto, vorrei esprimere la mia gratitudine alla Prof.ssa Ombretta Gaggi, relatore della mia tesi, per l'aiuto e il sostegno fornitomi durante la stesura del lavoro.

In secondo luogo, ringrazio Michele Marchetto, tutor aziendale, così come tutti i colleghi Tommaso Padovan, Lorenzo Ceccon e Luca D'Ambros per aver reso lo stage un'esperienza di crescita.

Ringrazio col cuore i miei genitori Giulia e Carlo per aver creduto in me in ogni momento e avermi sempre sostenuto.

Infine ringrazio i miei amici di sempre e gli amici nuovi che ho incontrato nel mio percorso universitario, per avermi reso quello che sono.

Padova, Oct 2016

Davide Bortot

Indice

1	Introduzione	1
1.1	L'azienda	1
1.2	L'idea	1
1.3	Cos'è Project Tango	2
1.4	Lavoro svolto	3
1.5	Il Prodotto - lato client	3
1.5.1	Il prototipo ad inizio stage: Samba	4
1.6	Organizzazione del testo	6
2	Processi, tecnologie e strumenti	7
2.1	Processo sviluppo prodotto	7
2.1.1	Kick-Off	7
2.1.2	Concept Preview	7
2.1.3	Product Prototype	8
2.1.4	Fasi successive	8
2.2	Tecnologie e Strumenti	8
2.2.1	Codice	8
2.2.2	IDE ed editor	9
2.2.3	Framework	9
3	Studio di fattibilità ed analisi dei rischi	11
3.1	Introduzione al progetto	11
3.2	Studio di fattibilità	11
3.2.1	Funzionalità per il trattamento di Point Cloud	11
3.2.2	Funzionalità per il filtraggio di Point Cloud	12
3.2.3	Funzionalità per il meshing	12
3.2.4	Metodologie per il calcolo del volume	13
3.2.5	Visualizzazione di file in formato OBJ	13
3.2.6	Comunicazione client-server	13
3.2.7	Conclusioni	13
3.3	Analisi preventiva dei rischi	14
3.3.1	Rischi generali	14
3.3.2	Rischi specifici	14
4	Analisi dei requisiti	17
4.1	Casi d'uso - lato client	17
4.1.1	UC1.1: Visualizzazione lista delle mesh salvate	18
4.1.2	UC1.2: Refresh delle mesh da server	18

4.1.3	UC1.3: Visualizzazione di una mesh	18
4.1.4	UC1.4: Eliminazione di una mesh	20
4.1.5	UC1.5: Ritorno all'activity principale	20
4.1.6	UC1.6: Errore nessuna mesh da caricare	20
4.1.7	UC1.7: Errore di connessione	20
4.2	Casi d'uso - lato server	21
4.2.1	UC2 Elaborazione di un Point Cloud	21
4.2.2	UC2.1 Salvataggio dei dati ricevuti su file PCD	22
4.2.3	UC2.2 Filtraggio del Point Cloud	22
4.2.4	UC2.3 Estrazione dell'oggetto	23
4.2.5	UC2.4 Generazione mesh	23
4.2.6	UC2.5 Calcolo del volume	24
4.3	Tracciamento dei requisiti	24
5	Progettazione e codifica	27
5.1	Tecnologie e strumenti	27
5.2	Ciclo di vita del software	27
5.3	Progettazione	27
5.4	Design Pattern utilizzati	27
5.5	Codifica	27
6	Verifica e validazione	29
7	Conclusioni	31
7.1	Consuntivo finale	31
7.2	Raggiungimento degli obiettivi	31
7.3	Conoscenze acquisite	31
7.4	Valutazione personale	31
A	Appendice A	33
	Bibliografia	37

Elenco delle figure

1.1	L' <i>hardware</i> di <i>Project Tango</i>	2
1.2	Esempio di <i>Point Cloud</i> : un cestino cilindrico	3
1.3	<i>Drifting</i> nel <i>Motion Tracking</i>	4
1.4	Benefici della <i>Drift Correction</i>	5
1.5	Effetti del <i>voxeling</i>	6
4.1	Use Case - UC1: Operazioni sulla lista di mesh salvate	17
4.2	Use Case - UC0: Scenario Principale	21
4.3	Use Case - UC2: Elaborazione di un Point Cloud	22

Elenco delle tabelle

4.1	Tabella del tracciamento dei requisiti funzionali	25
4.2	Tabella del tracciamento dei requisiti qualitativi	25
4.3	Tabella del tracciamento dei requisiti di vincolo	25

Capitolo 1

Introduzione

1.1 L'azienda

VIC è stata fondata da Alessio Bisutti che, dopo aver sviluppato una lunga esperienza nel campo ispettivo, ha deciso di costituire una società in grado di offrire ai propri clienti un servizio professionale, chiaro ed affidabile, appoggiandosi alle nuove tecnologie. Si occupa di controlli per grandi ordini, sia di materie prime che di semi-lavorati, di cui individua e riporta eventuali danni, carenze nella spedizione e non conformità con quanto ordinato. VIC viene fondata a Venezia 7 anni fa come piccola società di ispezione locale. Fin dall'inizio, l'obiettivo principale di VIC è stato la riduzione del tempo tra ispezione e reporting al cliente. Ora l'obiettivo è raggiunto, perchè VIC sta fornendo ai suoi clienti tutti i risultati e le informazioni importanti in tempo reale, senza alcun ritardo, grazie agli investimenti fatti nel campo della tecnologia e delle applicazioni mobili.

1.2 L'idea

I più importanti obbiettivi del controllo qualità effettuato durante un ispezione sono il determinare la corretta forma, peso, quantità e dimensioni degli oggetti da esaminare. Gli ispettori possono scattare fotografie, prendere appunti e sfruttare la loro esperienza per fornire stime accurate; si è manifestata però la necessità di affiancare queste ultime a dei dati quanto più possibile oggettivi e rapidi da ottenere.

Da qui nasce l'idea di fornire agli ispettori uno strumento informatico in grado di effettuare queste stime. Grazie alla ricostruzione computerizzata resa disponibile dai *Tango device* sarà possibile non solo visualizzare su uno schermo il modello 3D del soggetto della ispezione, ma anche ottenere ulteriori dati utili quali:

- * Una stima del volume, e se necessario del peso, dell'oggetto.
- * L'esito del confronto dell'oggetto con un modello ideale, per evidenziare eventuali danni o deformazioni.

Con il prototipo realizzato durante lo *stage* sono rese disponibili solamente le funzionalità di ricostruzione dell'oggetto e calcolo (approssimato) del volume.

Le operazioni troppo computazionalmente costose da effettuare su tablet, quali il filtraggio e *meshing* dei punti acquisiti, sono state delegate ad un *backend server* che effettui le elaborazioni necessarie ed invii i risultati al richiedente, mentre all'applicativo

per tablet è stato affidato il compito di acquisire e visualizzare un oggetto sotto forma di 'Nuvola di Punti' e poterne esaminare la *mesh* ottenuta.

1.3 Cos'è Project Tango

Project Tango è un *tablet* sperimentale prodotto da *Google* in grado di scandire tridimensionalmente l'ambiente circostante e di tracciare la propria posizione rispetto ad esso. Ciò è possibile attraverso il ricco hardware di cui è dotato (si veda la figura 1.1), tra cui:

- * una fotocamera RGB-IR
- * una fotocamera *Fisheye*
- * un sensore di profondità IR
- * accelerometro e giroscopio

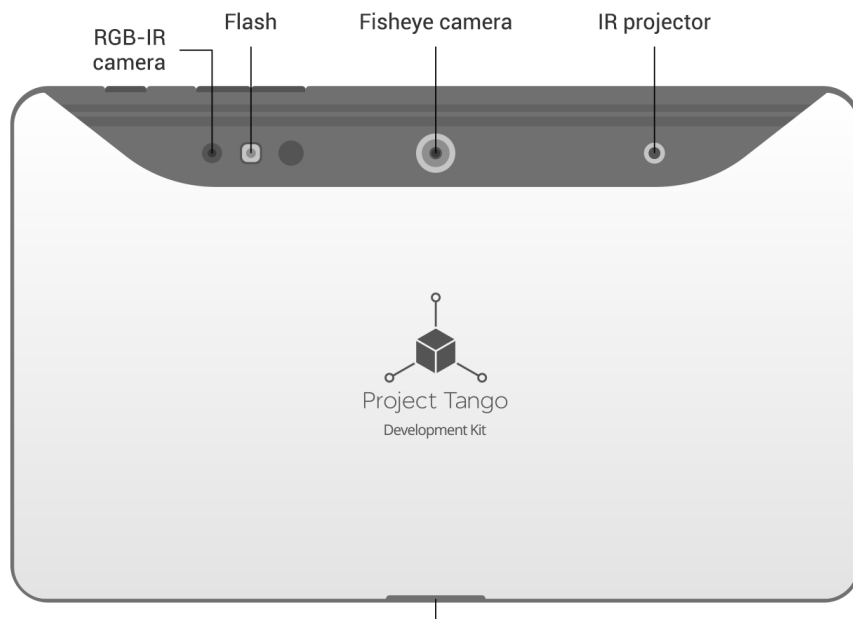


figura 1.1: L' hardware di *Project Tango*

Project Tango è quindi specificamente pensato per sviluppare applicazioni che necessitano di comprendere ed estrapolare informazioni dal mondo reale (ad es. realtà aumentata). Le funzionalità del *tablet* sono accessibili attraverso le *Tango API*, le *API* ufficiali per lo sviluppo di applicazioni *Tango*.

1.4 Lavoro svolto

Al tirocinante è stato richiesto di focalizzarsi sullo sviluppo del lato server dell'applicazione, in modo che fosse possibile ricevere ed elaborare 'Nuvole di Punti', ed ottenere i risultati elaborati, lavorando nel mentre in stretta collaborazione con un altro stagista, Tommaso Padovan (laureando il 13 Ottobre, relatore il Prof. Gilberto Filè) che si occupava dell'applicazione lato client. Nell'ultimo periodo di stage la parte server era già a buon punto, quindi il focus del tirocinio si è spostato sul miglioramento dell'applicativo lato client.

1.5 Il Prodotto - lato client

L'applicazione su *tablet* prodotta realizza, seppur non ancora in maniera completa, le esigenze richieste.

La sua realizzazione ha incontrato molte problematiche talvolta critiche e difficili da prevedere. Per questo, durante lo sviluppo, sono stati implementati più prototipi, al fine di esplorare le potenzialità e soprattutto i limiti del *tablet* e delle *Tango API*.

Lo scopo principale dell'applicazione lato *tablet* è quello di rilevare una corretta 'Nuvola di Punti' dell'oggetto che si vuole esaminare.

Una 'Nuvola di Punti' è una descrizione matematica di un oggetto tridimensionale ottenuta tramite un insieme, il più possibile fitto, di punti che lo compongono, definiti dalle loro coordinate (x,y,z) rispetto ad un fissato sistema di riferimento.

Tale rappresentazione, riferita spesso d'ora in poi con il più elegante termine inglese *Point Cloud*, è facilmente comprensibile all'utente se visualizzata come in figura 1.2.

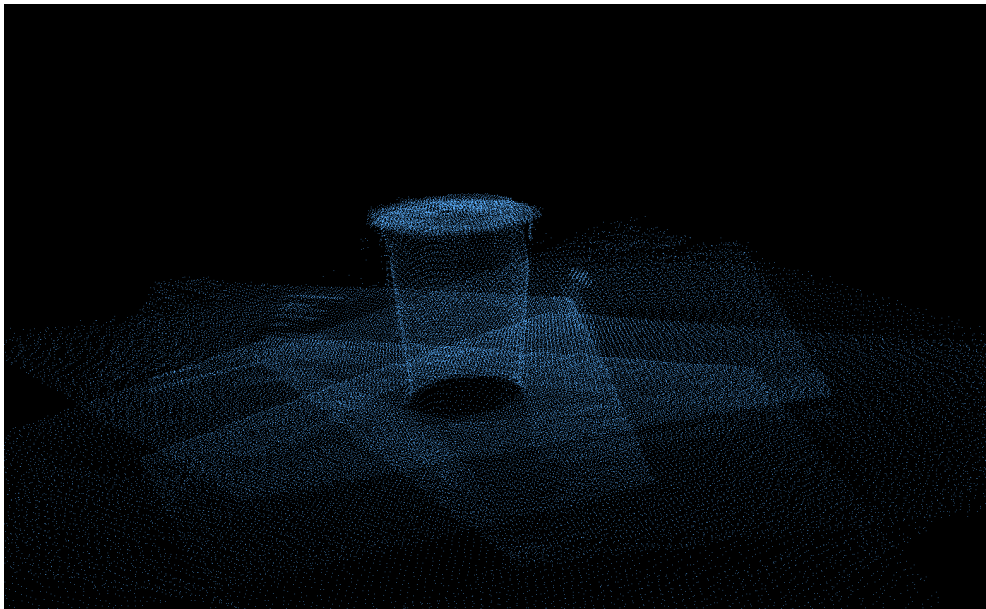


figura 1.2: Esempio di *Point Cloud*: un cestino cilindrico

1.5.1 Il prototipo ad inizio stage: Samba

In questa sezione viene presentato il prototipo nel suo stato all'inizio dello stage, per capirne le funzionalità e limiti e introdurre il lettore alle problematiche che un tale progetto sperimentale ha comportato. Sulla base dei risultati di *Samba* ho iniziato il mio lavoro sulla parte di elaborazione lato *server* dei *Point Cloud* acquisiti; questo stadio prototipale è stato inoltre un'occasione di collaborazione con l'altro stagista e il punto di inizio per lo sviluppo del prototipo successivo. *Samba* affronta e risolve alcuni problemi dei prototipi precedenti, come il "*Drifting*" e alcuni problemi prestazionali dovuti alla mole elevata di dati acquisiti.

Drift correction

Il prototipo introduce funzionalità che tentano di arginare il problema del "*drifting*", presente in stadi prototipali precedenti. Il *drifting* è un problema comune nelle applicazioni di realtà aumentata, che come *Project Tango* usano la tecnica del *Motion Tracking*, cioè aggiornano costantemente la propria posizione relativamente alle coordinate acquisite nella posizione precedente, mantenendo così una storia dei movimenti del *device* rispetto all'ambiente circostante.

Ad ogni aggiornamento della posizione è normale ed inevitabile che la misurazione, per quanto precisa, introduca un piccolo errore; la catena di errori sommati porta quindi col passare del tempo ad un'importante discrepanza tra la posizione stimata del *device* e la sua posizione reale, come evidenziato in figura 1.3

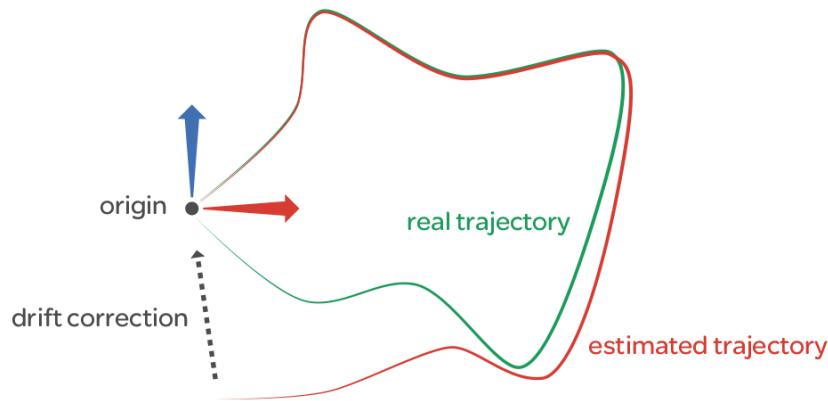


figura 1.3: *Drifting* nel *Motion Tracking*

Per ovviare in parte a questo problema *Samba* utilizza una tecnica chiamata *Drift Correction* che si appoggia sulle funzionalità di *Area Learning* del *device Tango*.

L'*Area Learning* consiste nella capacità del dispositivo di estrarre dallo spazio fisico che sta analizzando una serie di punti significativi (o *key features*), facilmente riconoscibili, e di salvare tali informazioni per confrontarle con le successive acquisizioni. In questo modo il dispositivo è capace di riconoscere un'area precedentemente visitata, e può quindi applicare le necessarie correzioni alla propria stima della traiettoria, di qui il nome *Drift Correction*.

Con l'implementazione di questa funzionalità è necessario, prima di iniziare la ricostruzione di un *Point Cloud*, riprendere l'oggetto e i suoi dintorni per un po' di tempo e

da diverse angolazioni, in modo da permettere al *tablet Tango* di costruire una mappa, detta *Area description*, dell'area circostante, e di stabilizzare la traiettoria stimata con le informazioni acquisite.

I risultati si vedono confrontando queste due ricostruzioni di una scatola, vista dall'alto (fig. 1.4a e fig. 1.4b), rispettivamente senza e con *Drift correction*.

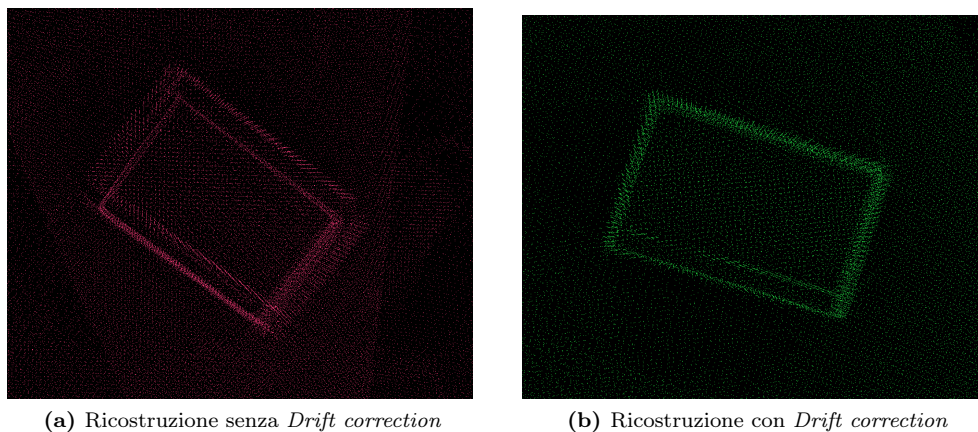


figura 1.4: Benefici della *Drift Correction*

Come si nota in figura 1.4b il risultato non è ancora ottimale, infatti un lato della scatola appare sdoppiato e spostato di qualche centimetro; si tratta di un problema di *ghosting* di cui verrà trattato più avanti.

Prestazioni

Il prototipo precedente presentava due problemi prestazionali principali:

- * La ricostruzione passo per passo del *Point Cloud* finale era troppo lenta
- * La mole dei dati trattati al sovrapporsi di più *Point Cloud* era proibitiva

Il primo problema si presentava utilizzando i metodi forniti dalla libreria *Tango* per trasformare le coordinate relative dei punti in coordinate assolute per permetterne la giusta sovrapposizione. Tale metodo, seppur di facile utilizzo, diventava troppo dispendioso all'aumentare delle dimensioni del *Point Cloud*, che raggiunge facilmente gli 80.000 punti.

Per risolvere il problema viene quindi creata, per ogni *Point Cloud*, una *matrice di rototraslazione* che rappresenta lo spostamento e la rotazione del dispositivo rispetto al sistema di riferimento, e viene moltiplicato il vettore delle coordinate di ogni singolo punto per la matrice generata.

In questo modo si sono ridotti i tempi di elaborazione dell'80%;

Il secondo problema era causato dal sovrapporsi di molti punti quasi identici, ad esempio i punti rappresentanti il pavimento. Partendo dall'osservazione che una certa quantità di punti molto vicini può essere trasformata in un unico punto, valore medio di tutti gli altri, senza una significativa perdita d'informazione, *Samba* risolve il problema attraverso una tecnica di *voxeling*.

Lo spazio viene suddiviso in tanti piccoli parallelepipedi (tipicamente cubi) di uguali

dimensioni; tutti i punti che ricadono all'interno di un singolo parallelepipedo, o *voxel*, vengono considerati come un unico punto. Così facendo si riducono sensibilmente le dimensioni del *Point Cloud* senza alterarne negativamente la precisione. Di seguito la differenza visivamente evidente tra lo stesso Point Cloud filtrato prima con voxel cubici di lato 1cm (fig. 1.5a) e poi di lato 3cm (fig. 1.5b), con una differenza di circa 60.000 punti rimossi in più nella seconda elaborazione.

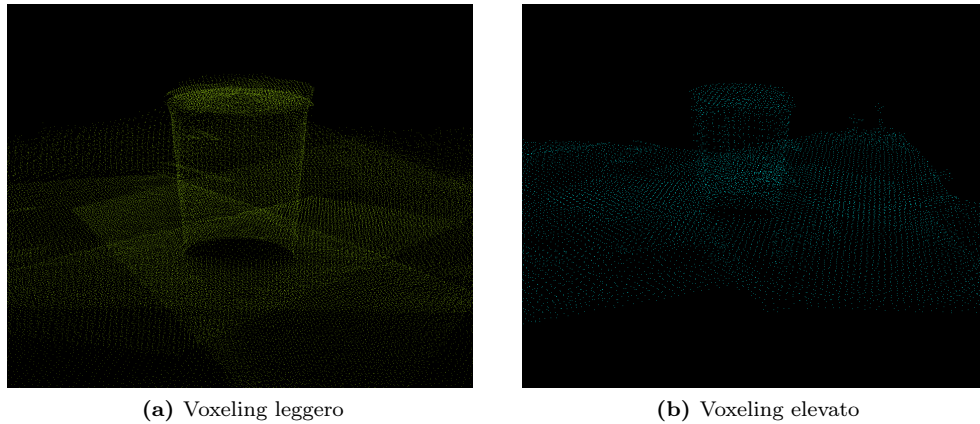


figura 1.5: Effetti del *voxeling*

1.6 Organizzazione del testo

Riguardo la stesura del testo, relativamente al documento sono state adottate le seguenti convenzioni tipografiche:

- * gli acronimi, le abbreviazioni e i termini ambigui o di uso non comune menzionati vengono definiti nel glossario, situato alla fine del presente documento;
- * per la prima occorrenza dei termini riportati nel glossario viene utilizzata la seguente nomenclatura: *parola*^[g];
- * i termini in lingua straniera o facenti parti del gergo tecnico sono evidenziati con il carattere *corsivo*.

Capitolo 2

Processi, tecnologie e strumenti

Una buona definizione dei processi produttivi ed un'adeguata scelta delle tecnologie e strumenti da utilizzare è fondamentale per il successo del progetto

2.1 Processo sviluppo prodotto

In ambito aziendale si è scelto di provare, per questo progetto, un processo di sviluppo *software* basato sulla filosofia *Lean*.

Dato che l'obiettivo principale era fornire un prototipo ci si è limitati solamente alle tre fasi iniziali dello sviluppo di *Lean*, ovvero *Kick-Off*, *Concept Preview* e *Product Prototype*.

2.1.1 Kick-Off

Questa è la prima fase dello sviluppo *software*, coincide con la prima riunione ufficiale del team di progetto, aperta anche agli *Stakeholder*.

Si pone lo scopo di iniziare la fase l'*allestimento* e l'*avviamento* in cui viene determinata la natura e lo scopo del progetto.

2.1.2 Concept Preview

Fase in cui è reso disponibile un primo campione di prova del prodotto, detto *concept*. Esso può essere incompleto e affetto da errori, ma deve essere in grado di dimostrare agli *stakeholder* le caratteristiche principali che avrà il prodotto finito.

Esso è soggetto a un riesame che ha lo scopo di valutare se è in linea con gli obiettivi definiti nella *Value Proposition*, la *milestone* non può essere raggiunta senza che questo riesame abbia esito positivo.

Questa fase coincide anche con l'inizio della progettazione, che deve essere portata avanti fino ad un livello di dettaglio ritenuto opportuno dal *team*.

2.1.3 Product Prototype

Fase in cui è messo a disposizione il primo prototipo del nuovo prodotto, completo nelle sue funzioni (sviluppo finito) ma non ancora messo a punto mediante verifiche e correzioni, per garantirne funzionalità e prestazioni. Il prototipo deve essere ad un stato tale da poter essere dato in valutazione agli *stakeholder*.

Esso è soggetto a un riesame che ha lo scopo di valutare se è in linea con gli obiettivi definiti sia *Value Proposition* che nella *Requirements Specification*, la *milestone* non può essere raggiunta senza che questo riesame abbia esito positivo.

Questa fase coincide con il termine della fase di progettazione e l'inizio della fase di esecuzione, cioè l'insieme dei processi necessari a soddisfare i requisiti del progetto.

2.1.4 Fasi successive

Le fasi successive, ovvero *Product Design Freeze* e *Start Of Production*, possono essere avviate nel futuro a partire dal *Product Prototype* se ciò verrà ritenuto opportuno dall'azienda.

2.2 Tecnologie e Strumenti

L'azienda ha lasciato grande libertà riguardo agli strumenti da utilizzare per questo progetto, quindi essi sono stati fissati inizialmente e successivamente incrementati al crescere delle necessità.

2.2.1 Codice

Segue la lista dei linguaggi utilizzati per la codifica.

- * **Java**¹: Il linguaggio preferito per l'applicazione lato *tablet*. È stato scelto seguendo le *Best Practice* dello sviluppo *Android*.
- * **C++**²: Il linguaggio preferito per l'applicazione lato *Server*. È stato scelto perché tutti le più diffuse librerie per l'elaborazione dei *Point Cloud*, ed in particolare *PCL* sono disponibili in questo linguaggio.
- * **PCL, Point Cloud Library**³: Una delle librerie di maggior rilievo nel campo della *Computer Vision*, mette a disposizione notevoli funzionalità per l'elaborazione, il filtraggio e l'ottimizzazione dei *Point Cloud*.
- * **Tango API**⁴: Le *API* ufficiali, disponibili in linguaggio Java, C++ e C#, per lo sviluppo di applicazioni *Tango*.
- * **PHP**⁵: Il linguaggio usato per ricevere ed inviare le richieste *HTTP* necessarie alla comunicazione tra *Server* e dispositivo.
- * **Python**⁶: Il linguaggio usato in combinazione con *PHP* per elaborare le richieste ed implementare la logica del lato *Server*.

¹<https://www.java.com>

²www.cplusplus.com

³<http://pointclouds.org>

⁴<https://developers.google.com/tango/apis/overview>

⁵<https://secure.php.net>

⁶<https://www.python.org>

2.2.2 IDE ed editor

Segue la lista degli ambienti per la codifica utilizzati durante il progetto.

- * **Android Studio**⁷: L'*IDE* ufficiale per le applicazioni *Android*.
- * **QT**⁸: L'*IDE* scelto per lo sviluppo del codice *C++*.
- * **Geany**⁹: L'*editor* di testo usato per scrivere gli *script php* e *Python*.

2.2.3 Framework

Segue la lista dei *Framework* usati per il tirocinio.

- * **Gradle**¹⁰: È stato usato come *tool* di *build* per tutta l'applicazione *Android*.
- * **Rajawali3D**¹¹: È stato usato come *framework* grafico per la realizzazione del *render*.
- * **OkHttp**¹²: È stata usata come *framework* di riferimento per le richieste *http*, come indicato nella *Android Best Practices*.
- * **TangoUx**¹³: *Framework* messo a disposizione dalla *Google* assieme alle *API Tango*. È stato usato per gestire le notifiche all'utente relative ai comportamenti che deve tenere per permettere il buon funzionamento del dispositivo e dei sensori.
- * **Jni**¹⁴: Questo *framework* permette di richiamare metodi 'nativi' (scritti in *C/C++*) dal codice *Java*.
- * **Firebase**¹⁵: *Framework* associato all'omonimo *web service* offerto da *Google* per lo scambio di messaggi tra applicazioni *Android* e un *server*.
- * **GOICP**¹⁶: libreria che implementa una variante dell'algoritmo *ICP* per la *Registration* di set di *Point Cloud*.

⁷<https://developer.android.com/studio/index.html>

⁸<https://www.qt.io/ide>

⁹<https://www.geany.org>

¹⁰<https://gradle.org>

¹¹<https://github.com/Rajawali/Rajawali>

¹²<http://square.github.io/okhttp>

¹³<https://developers.google.com/tango/ux/ux-framework>

¹⁴<http://docs.oracle.com/javase/7/docs/technotes/guides/jni>

¹⁵<https://firebase.google.com>

¹⁶<http://iitlab.bit.edu.cn/mcislabs/yangjiaolong/go-icp/>

Capitolo 3

Studio di fattibilità ed analisi dei rischi

Il progetto si è subito presentato come sperimentale, impegnativo e facilmente soggetto a fallimento. Per questo si è reso necessario studiarne attentamente la fattibilità e valutarne i rischi

3.1 Introduzione al progetto

Data la natura innovativa del progetto è stato necessario produrre diversi prototipi ed effettuare l'Analisi dei Rischi e lo Studio di Fattibilità in diverse fasi. Seguendo quest'approccio è stato possibile valutare le potenzialità e i limiti del *device Tango* e delle API annesse, e considerare l'applicabilità degli algoritmi general purpose della libreria PCL al caso particolare del progetto in questione.

3.2 Studio di fattibilità

Prima di iniziare il progetto è stato effettuato un accurato studio di fattibilità basato sullo studio della libreria PCL e della vasta quantità di esempi d'utilizzo della stessa reperibili online. Si sono cercate poi soluzioni per il calcolo del volume di una mesh. Per il lato client si sono esplorate soluzioni per la visualizzazione di mesh salvate in formato OBJ e per la comunicazione client-server.

3.2.1 Funzionalità per il trattamento di Point Cloud

É parso subito evidente che l'applicativo necessita di un formato standard, trasferibile via Web, per il trattamento di Point Cloud, e delle funzionalità di I/O associate. A tal riguardo è risultata particolarmente utile la documentazione della PCL:

- * **The PCD (Point Cloud Data) file format¹**: un formato facilmente comprensibile e portabile per definire nuvole di punti

¹http://pointclouds.org/documentation/tutorials/pcd_file_format.php/pcd-file-format.

- * **Reading Point Cloud data from PCD files²**: tutorial che espone le funzionalità di lettura e caricamento di un file PCD
- * **Writing Point Cloud data to PCD files³**: tutorial che espone le funzionalità di salvataggio di un file PCD

3.2.2 Funzionalità per il filtraggio di Point Cloud

Sono disponibili sul sito di PCL svariati esempi open-source che utilizzano le potenzialità della libreria per filtrare un Point Cloud, ad esempio:

- * **Filtering a PointCloud using a PassThrough filter⁴**: tutorial per la creazione di un filtro capace di eliminare i punti che non rientrano in un certo range definito dall'utente.
- * **Downsampling a PointCloud using a VoxelGrid filter⁵**: tutorial per la creazione di un filtro capace di effettuare il downsampling di un PointCloud
- * **Removing outliers using a StatisticalOutlierRemoval filter⁶**: tutorial per la creazione di un filtro capace di eliminare i punti isolati, quindi non interessanti, di un Point Cloud
- * **Plane model segmentation⁷**: tutorial per la creazione di un filtro capace di eliminare le componenti planari di un Point Cloud.
- * **Euclidean Cluster Extraction⁸**: tutorial per la creazione di un filtro capace di isolare ed estrarre gruppi consistenti e distinguibili di punti da un Point Cloud.

3.2.3 Funzionalità per il meshing

Sono disponibili sul sito di PCL alcuni esempi open-source che utilizzano le potenzialità della libreria per generare una mesh tridimensionale a partire da un Point Cloud, ad esempio:

- * **Fast triangulation of unordered point clouds⁹**: tutorial per la creazione di una mesh triangolare a partire da una nuvola di punti
- * **Fitting trimmed B-splines to unordered point clouds¹⁰**: tutorial per la creazione di una mesh regolare adattata iterativamente alla nuvola di punti

²http://pointclouds.org/documentation/tutorials/reading_pcd.php/reading-pcd.

³http://pointclouds.org/documentation/tutorials/writing_pcd.php/writing-pcd.

⁴<http://pointclouds.org/documentation/tutorials/passthrough.php/passthrough>.

⁵http://pointclouds.org/documentation/tutorials/voxel_grid.php/voxelgrid.

⁶http://pointclouds.org/documentation/tutorials/statistical_outlier.php/statistical-outlier-removal.

⁷http://pointclouds.org/documentation/tutorials/planar_segmentation.php/planar-segmentation.

⁸http://pointclouds.org/documentation/tutorials/cluster_extraction.php/cluster-extraction.

⁹http://pointclouds.org/documentation/tutorials/greedy_projection.php/greedy-triangulation.

¹⁰http://pointclouds.org/documentation/tutorials/bspline_fitting.php/bspline-fitting.

3.2.4 Metodologie per il calcolo del volume

Durante la ricerca di un affidabile metodo matematico per il calcolo del volume di una mesh è risultato che il più adatto, validato ormai da anni di utilizzo, è il calcolo del volume con segno di una mesh triangolare come descritto ad esempio nel paper "EFFICIENT FEATURE EXTRACTION FOR 2D/3D OBJECTS IN MESH REPRESENTATION"¹¹ da Cha Zhang e Tsuhan Chen.

3.2.5 Visualizzazione di file in formato OBJ

Per visualizzare le mesh prodotte dal server su tablet è stato necessario ricercare esempi di applicativi che permettessero di trattare oggetti tridimensionali in formato OBJ. A tal riguardo si sono studiate due applicazioni:

- * **3D viewer**¹²: applicazione android per la visualizzazione di file OBJ e 3DS
- * **3D Model Viewer Open Source**¹³: applicazione open-source android per la visualizzazione di file OBJ realizzata da Andres Oviedo, il cui codice è liberamente reperibile dalla repository¹⁴ GitHub

3.2.6 Comunicazione client-server

Per implementare la comunicazione client-server la scelta più logica è stata di seguire esplorare la documentazione ufficiale Android, che mette a disposizione il package "**Apache HTTP**"¹⁵ che risponde alle necessità richieste.

3.2.7 Conclusioni

Alla luce della disponibilità e qualità degli esempi sopracitati il progetto è apparso fattibile e quindi è stato possibile dare il via al ciclo di vita del progetto software.

¹¹http://research.microsoft.com/en-us/um/people/chazhang/publications/icip01_Chazhang.pdf.

¹²<https://play.google.com/store/apps/details?id=com.pcvirt.Viewer3D&hl=it>.

¹³<https://play.google.com/store/apps/details?id=org.andresoviedo.dddmodel&hl=it>.

¹⁴<https://github.com/andresoviedo/android-3D-model-viewer>.

¹⁵<https://developer.android.com/reference/org/apache/http/package-summary.html>.

3.3 Analisi preventiva dei rischi

Durante la fase di analisi iniziale sono stati individuati alcuni possibili rischi a cui si potrà andare incontro. Si è quindi proceduto a elaborare delle possibili soluzioni per far fronte a tali rischi.

3.3.1 Rischi generali

1. Limiti fisici del tablet

Probabilità: Alta.

Impatto: Medio.

Descrizione: Il dispositivo è dotato di sensori IR che sfruttano la riflessione della luce per determinare la distanza dei punti che è in grado di individuare, e di fotocamere RGB. Superfici lucide/riflettenti o tendenti al nero possono compromettere gravemente la qualità della misurazione, così come gli ambienti con illuminazione scarsa o troppo intensa.

Gestione: Attento studio delle potenzialità dell'hardware del tablet¹⁶ e del loro utilizzo¹⁷ nelle Tango API.

2. Incomprensibilità dello stato del progetto

Probabilità: Media.

Impatto: Medio.

Descrizione: Venendo inserito in un progetto già avviato, lo studente necessita di comprendere a fondo lo stato dell'arte dell'applicativo.

Gestione: Lo studente deve, consultando la documentazione creata e collaborando strettamente con gli altri elementi del team e con il tutor aziendale, assicurarsi di comprendere quanto è stato fatto, come è stato fatto e le necessità future del progetto..

3. Scarsa competenza nello sviluppo Android

Probabilità: Alta.

Impatto: Basso.

Descrizione: Lo studente ha scarse conoscenze dello sviluppo Android, ma conosce il linguaggio Java.

Gestione: Lo studente deve effettuare un adeguato periodo di studio ed assumere familiarità con la documentazione ufficiale.

3.3.2 Rischi specifici

4. Inadeguatezza della libreria PCL e relativa documentazione

Probabilità: Bassa.

Impatto: Basso.

Descrizione: La libreria PCL espone molti algoritmi general-purpose per la manipolazione di Point Cloud, che non necessariamente si adattano alle necessità specifiche del progetto in questione.

Gestione: La libreria espone anche molte funzionalità basilari che è possibile utilizzare e combinare per ovviare alle necessità che algoritmi specifici non sempre soddisfano.

¹⁶<https://developers.google.com/tango/hardware/tablet>.

¹⁷<https://developers.google.com/tango/overview/depth-perception>.

5. Impredicibilità dei Point Cloud

Probabilità: Alta.

Impatto: Medio.

Descrizione: La qualità di un oggetto ricostruito come Point Cloud è imprevedibile, a causa dei fenomeni di *drifting*, *ghosting* ed artefatti di cui soffre la ricostruzione. Inoltre la tipologia di oggetti scansionati è altamente variabile.

Gestione: Il processo di filtraggio dev'essere attentamente impostato in modo da trattare al meglio tali imperfezioni e tipologie di oggetti. Bisogna inevitabilmente prendere in considerazione che non sempre il filtraggio può ottenere i risultati voluti, e richiedere quindi una nuova ricostruzione all'utente.

6. Difficoltà nella comunicazione client-server

Probabilità: Media.

Impatto: Basso.

Descrizione: L'applicazione verrà usata in futuro sul campo da addetti aziendali, che potrebbero non disporre di una connessione stabile con il server.

Gestione: Bisogna prevedere e controllare una possibile assenza di connessione e regolarsi di conseguenza, fornendo ad esempio all'utente la possibilità di salvare il proprio lavoro e ritentare la comunicazione col server in un secondo momento. Dato lo stadio prototipale del progetto e l'utilizzo indoor dell'applicativo, il rischio è altamente improbabile.

7. Difficoltà nel calcolo del volume

Probabilità: Alta.

Impatto: Media.

Descrizione: Il calcolo del volume dipende fortemente dal risultato dell'elaborazione attraverso le funzionalità della PCL, in particolare dall'assenza di punti estranei all'oggetto prima di effettuarne il meshing, al quale segue il calcolo..

Gestione: Lo studente deve eseguire una *suite* di test adeguati per valutare la bontà del calcolo del volume e la percentuale d'errore ammissibile, e capire le cause che portano il calcolo a divergere dal volume reale.

Capitolo 4

Analisi dei requisiti

4.1 Casi d'uso - lato client

Per lo studio dei casi di utilizzo del prodotto sono stati creati dei diagrammi. I diagrammi dei casi d'uso (in inglese *Use Case Diagram*) sono diagrammi di tipo [Unified Modeling Language \(UML\)](#) dedicati alla descrizione delle funzioni o servizi offerti da un sistema, così come sono percepiti e utilizzati dagli attori che interagiscono col sistema stesso.

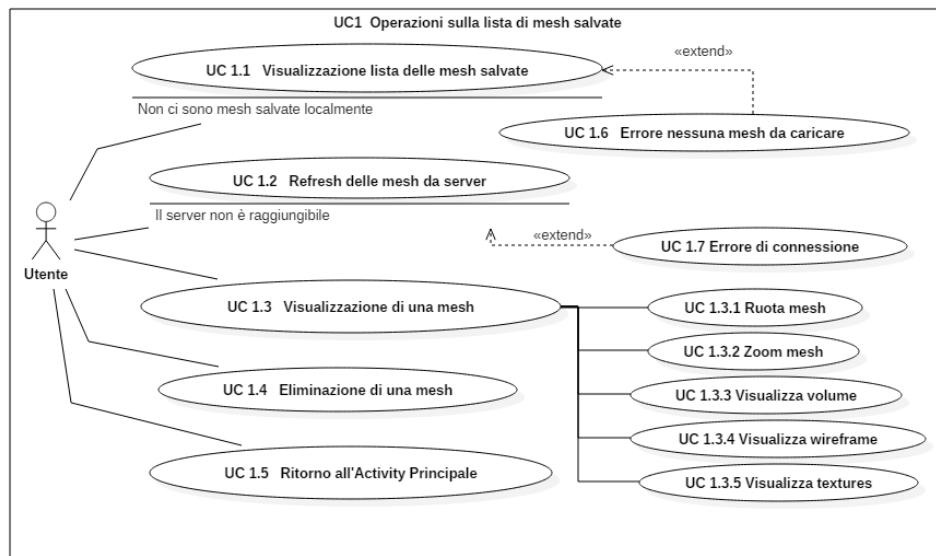


figura 4.1: Use Case - UC1: Operazioni sulla lista di mesh salvate

UC1: Operazioni sulla lista di mesh salvate

Attori Principali: User.

Precondizioni: L'utente ha aperto l'applicazione ed ha premuto sul pulsante per visualizzare la lista delle mesh 3D salvate localmente.

Descrizione: L'utente vede sullo schermo la lista delle mesh 3D salvate su disco su cui può effettuare diverse azioni.

Postcondizioni: Il sistema è pronto per permettere una nuova interazione.

4.1.1 UC1.1: Visualizzazione lista delle mesh salvate

Attori Principali: Utente.

Precondizioni: L'utente ha aperto l'applicazione ed ha premuto sul pulsante per visualizzare la lista delle *mesh* salvate localmente.

Descrizione: L'utente consulta la lista delle *mesh* salvate localmente.

Postcondizioni: Viene visualizzata la lista delle mesh salvate localmente.

4.1.2 UC1.2: Refresh delle mesh da server

Attori Principali: Utente.

Precondizioni: L'utente ha aperto l'applicazione, ha premuto sul pulsante per visualizzare la lista delle *mesh* salvate su disco ed ha intenzione di aggiornare la lista di *mesh* aggiungendo le altre presenti sul *Server*.

Descrizione: L'utente preme sul simbolo di *refresh* in alto a destra e ricarica la lista di *mesh* eventualmente scaricando quelle sul *Server* ma non sul dispositivo. L'utente viene informato dell'avvenuto refreshing.

Postcondizioni: L'utente ha a disposizione una lista aggiornata di *mesh*.

4.1.3 UC1.3: Visualizzazione di una mesh

Attori Principali: Utente.

Precondizioni: L'utente ha aperto l'applicazione, ha premuto sul pulsante per visualizzare la lista delle *mesh* salvate su disco ed intende visualizzare una specifica *mesh* in 3D.

Descrizione: L'utente preme sul nome della *mesh* che intende visualizzare, a questo punto si apre un piccolo ambiente grafico 3D dove l'utente può osservare la ricostruzione ed effettuare operazioni su di essa.

Postcondizioni: Nessuna.

UC1.3.1: Ruota mesh

Attori Principali: Utente.

Precondizioni: L'utente sta visualizzando una mesh e vuole ruotarla.

Descrizione: L'utente, scorrendo lo schermo con un dito, ruota l'oggetto visualizzato.

Postcondizioni: La mesh è stata ruotata.

UC1.3.2: Zoom mesh

Attori Principali: Utente.

Precondizioni: L'utente sta visualizzando una mesh e vuole ingrandirla/rimpicciolirla.

Descrizione: L'utente, usando due dita, ingrandisce/rimpicciolisce l'oggetto visualizzato.

Postcondizioni: La mesh è stata ingrandita/rimpicciolita.

UC1.3.3: Visualizza volume

Attori Principali: Utente.

Precondizioni: L'utente sta visualizzando una mesh e vuole conoscerne il volume calcolato.

Descrizione: L'utente, cliccando un apposito bottone in alto a destra, visualizza un popup con il volume calcolato della mesh.

Postcondizioni: L'utente ha visualizzato il volume della mesh.

UC1.3.4: Visualizza wireframe

Attori Principali: Utente.

Precondizioni: L'utente sta visualizzando una mesh e vuole osservarne solamente il *wireframe*, cioè lo scheletro di triangoli di cui è composta.

Descrizione: L'utente, cliccando un apposito bottone in alto a destra, visualizza solamente il *wireframe* dell'oggetto.

Postcondizioni: Viene visualizzato il *wireframe* dell'oggetto.

UC1.3.5: Visualizza textures

Attori Principali: Utente.

Precondizioni: L'utente sta visualizzando una mesh e vuole applicarvi delle textures di default.

Descrizione: L'utente, cliccando un apposito bottone in alto a destra, visualizza la mesh alla quale vengono applicate delle semplici textures di default.

Postcondizioni: Viene visualizzata la mesh dell'oggetto, alla quale sono state applicate textures di default.

4.1.4 UC1.4: Eliminazione di una mesh

Attori Principali: Utente.

Precondizioni: L'utente ha aperto l'applicazione, ha premuto sul pulsante per visualizzare la lista delle *mesh* salvate localmente.

Descrizione: L'utente preme sul pulsante "Delete" affianco al nome di una specifica mesh per cancellarla dai *File* salvati localmente.

Postcondizioni: Il *File* selezionato è stato cancellato e non è più presente sulla memoria del dispositivo.

4.1.5 UC1.5: Ritorno all'activity principale

Attori Principali: Utente.

Precondizioni: L'utente ha aperto l'applicazione, ha premuto sul pulsante per visualizzare la lista delle *mesh* salvate su disco ma desidera ritornare all'*activity* principale.

Descrizione: L'utente preme sul tasto "Back" e ritorna all'*activity* principale.

Postcondizioni: L'utente ritorna all'*activity* principale.

4.1.6 UC1.6: Errore nessuna mesh da caricare

Attori Principali: Utente.

Precondizioni: L'utente ha aperto l'applicazione e ha premuto sul pulsante per visualizzare la lista delle *mesh* salvate su disco.

Descrizione: L'utente viene informato dell'assenza di mesh salvate localmente.

Postcondizioni: La lista di mesh non è stata popolata.

4.1.7 UC1.7: Errore di connessione

Attori Principali: Utente.

Precondizioni: L'utente ha selezionato il pulsante di refresh per aggiornare la lista di *mesh* aggiungendo le altre presenti sul *Server*.

Descrizione: L'utente viene informato dell'assenza di connessione con il *Server*.

Postcondizioni: La comunicazione tra dispositivo e *Server* non è possibile.

4.2 Casi d'uso - lato server

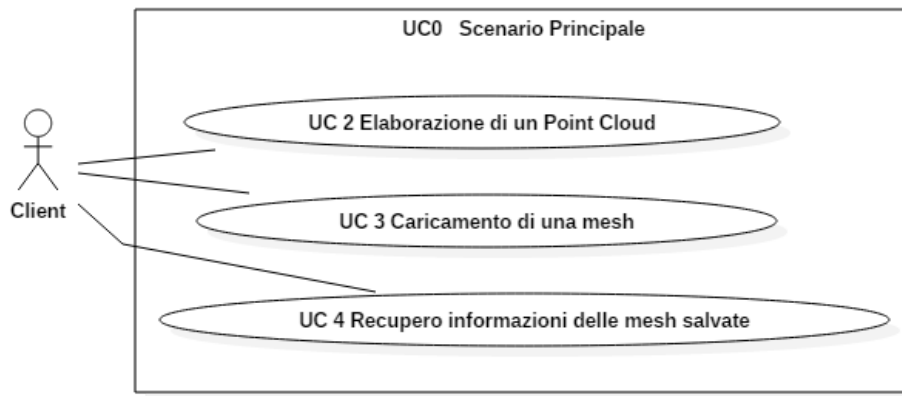


figura 4.2: Use Case - UC0: Scenario Principale

UC0: Scenario Principale

Attori Principali: Client.

Precondizioni: Il client invia una specifica richiesta al server.

Descrizione: Il server elabora la richiesta del client e ne restituisce i risultati.

Postcondizioni: Il server ha elaborato la richiesta ed inviato il responso al client.

4.2.1 UC2 Elaborazione di un Point Cloud

UC2: Elaborazione di un Point Cloud

Attori Principali: Client.

Precondizioni: Il client richiede l'elaborazione di un Point Cloud al server.

Descrizione: Il server elabora il Point Cloud ricevuto e notifica l'esito al client.

Postcondizioni: Il client riceve l'esito dell'elaborazione dal server.

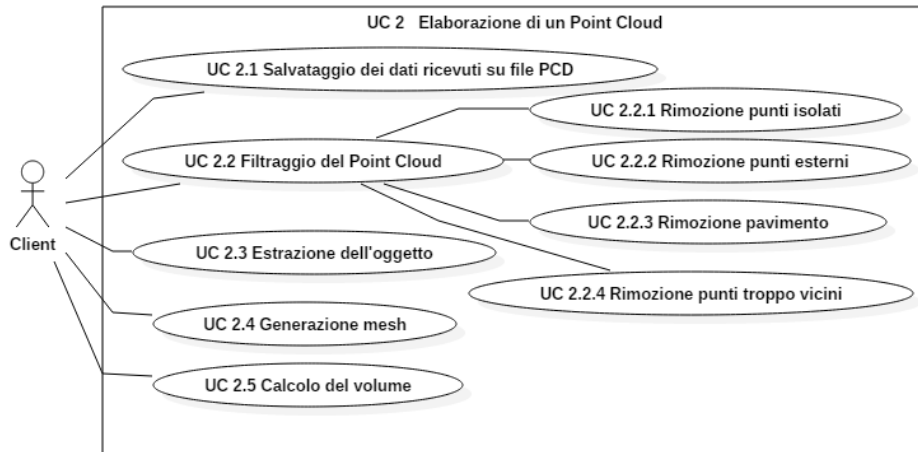


figura 4.3: Use Case - UC2: Elaborazione di un Point Cloud

4.2.2 UC2.1 Salvataggio dei dati ricevuti su file PCD

Attori Principali: Client.

Precondizioni: Il Client invia tramite protocollo HTTP dati in formato PCD.

Descrizione: I dati ricevuti vengono salvati sul server in un file in formato PCD.

Postcondizioni: I dati sono stati correttamente salvati su file.

4.2.3 UC2.2 Filtraggio del Point Cloud

Attori Principali: Client.

Precondizioni: Il Client ha richiesto l'elaborazione di un Point Cloud.

Descrizione: Il Point Cloud viene caricato da file e filtrato.

Postcondizioni: Il Point Cloud è stato filtrato.

UC2.2.1 Rimozione punti isolati

Attori Principali: Client.

Precondizioni: Viene richiesto di eliminare i punti isolati della nuvola.

Descrizione: Il Point Cloud viene filtrato eliminando i punti isolati.

Postcondizioni: I punti isolati sono stati rimossi dal Point Cloud.

UC2.2.2 Rimozione punti esterni

Attori Principali: Client.

Precondizioni: Viene richiesto di eliminare i punti esterni della nuvola.

Descrizione: Il Point Cloud viene filtrato eliminando i punti più esterni, che non appartengono all'oggetto scansionato.

Postcondizioni: I punti esterni sono stati rimossi dal Point Cloud.

UC2.2.3 Rimozione pavimento

Attori Principali: Client.

Precondizioni: Viene richiesto di eliminare i punti del pavimento dalla nuvola.

Descrizione: Il Point Cloud viene filtrato eliminando i punti planari che corrispondono al pavimento.

Postcondizioni: I punti del pavimento sono stati rimossi dal Point Cloud.

UC2.2.4 Rimozione punti troppo vicini

Attori Principali: Client.

Precondizioni: Viene richiesto di eliminare i punti ravvicinati della nuvola.

Descrizione: Il Point Cloud viene filtrato eliminando i punti eccessivamente ravvicinati, che possono essere approssimati in un unico punto.

Postcondizioni: I punti ravvicinati sono stati rimossi dal Point Cloud.

4.2.4 UC2.3 Estrazione dell'oggetto

Attori Principali: Client.

Precondizioni: Viene richiesto di estrarre dalla nuvola i soli punti appartenenti all'oggetto scansionato.

Descrizione: La parte di Point Cloud che rappresenta l'oggetto scansionato viene isolata dal resto dei punti.

Postcondizioni: I punti del solo oggetto scansionato sono stati estratti dal Point Cloud.

4.2.5 UC2.4 Generazione mesh

Attori Principali: Client.

Precondizioni: Il Client richiede di generare una mesh dell'oggetto ispezionato.

Descrizione: Viene effettuato il meshing del Point Cloud del solo oggetto scansionato.

Postcondizioni: È stata generata correttamente una mesh dell'oggetto ispezionato.

4.2.6 UC2.5 Calcolo del volume

Attori Principali: Client.

Precondizioni: il Client richiede di calcolare il volume dell'oggetto scansionato.

Descrizione: Viene calcolato il volume della mesh generata a partire dal Point Cloud.

Postcondizioni: Il volume dell'oggetto è stato calcolato.

4.3 Tracciamento dei requisiti

Da un'attenta analisi dei requisiti e degli use case effettuata sul progetto è stata stilata la tabella che traccia i requisiti in rapporto agli use case.

Sono stati individuati diversi tipi di requisiti e si è quindi fatto utilizzo di un codice identificativo per distinguerli.

Il codice dei requisiti è così strutturato $R(F/Q/V)(N/D/O)$ dove:

R = requisito

F = funzionale

Q = qualitativo

V = di vincolo

N = obbligatorio (necessario)

D = desiderabile

Z = opzionale

Nelle tabelle 4.1, 4.2 e 4.3 sono riassunti i requisiti e il loro tracciamento con gli use case delineati in fase di analisi.

tabella 4.1: Tabella del tracciamento dei requisiti funzionali

Requisito	Descrizione	Use Case
RFN-1		UC1

tabella 4.2: Tabella del tracciamento dei requisiti qualitativi

Requisito	Descrizione	Use Case
RQD-1		-

tabella 4.3: Tabella del tracciamento dei requisiti di vincolo

Requisito	Descrizione	Use Case
RVO-1		-

Capitolo 5

Progettazione e codifica

Breve introduzione al capitolo

5.1 Tecnologie e strumenti

Di seguito viene data una panoramica delle tecnologie e strumenti utilizzati.

Tecnologia 1

Descrizione Tecnologia 1.

Tecnologia 2

Descrizione Tecnologia 2

5.2 Ciclo di vita del software

5.3 Progettazione

Namespace 1

Descrizione namespace 1.

Classe 1: Descrizione classe 1

Classe 2: Descrizione classe 2

5.4 Design Pattern utilizzati

5.5 Codifica

Capitolo 6

Verifica e validazione

Capitolo 7

Conclusioni

7.1 Consuntivo finale

7.2 Raggiungimento degli obiettivi

7.3 Conoscenze acquisite

7.4 Valutazione personale

Appendice A

Appendice A

Citazione

Autore della citazione

Bibliografia

+

Bibliografia

- [1] Cha Zhang and Tsuhan Chen, *EFFICIENT FEATURE EXTRACTION FOR 2D/3D OBJECTS IN MESH REPRESENTATION*, Dept. of Electrical and Computer Engineering, Carnegie Mellon University 5000 Forbes Avenue, Pittsburgh, PA 15213, USA.