

UNIVERSITÀ DEGLI STUDI DI PADOVA

Facoltà di Ingegneria

Corso di Laurea Magistrale in Ingegneria Informatica

Tesi di Laurea

**RICOSTRUZIONE 3D
IN TEMPO REALE
CON SENSORE A TEMPO DI VOLO**

Relatore:

Prof. GUIDO MARIA CORTELLAZZO

Laureando:

CLAUDIO PAOLINI

Correlatore:

Prof. PIETRO ZANUTTIGH

ANNO ACCADEMICO 2010-2011

*Alla mia famiglia
che ha sempre creduto in me*

Sommario

La tesi riguarda la ricostruzione di modelli tridimensionali a partire dai dati di un sensore a tempo di volo, tecnologia ad oggi ancora prematura per questo tipo di applicazione, solitamente affidata a scanner laser. In questa attività di ricerca sono stati sviluppati approcci in grado di compensare i limiti della videocamera e finalizzati ad ottenere una ricostruzione del modello tridimensionale quanto più accurata possibile, evidenziando i limiti e i vantaggi di questa tecnologia.

Il contributo principale di questa tesi è la realizzazione in tempo reale dell'acquisizione e della ricostruzione 3d del modello. Il sistema non richiede pc di elevata potenza computazionale, rendendo dunque il programma accessibile ai computer oggi in commercio.

La tesi è costituita da quattro capitoli fondamentali: una breve descrizione del sensore a tempo di volo, in cui vengono messi in evidenza i principi di funzionamento, i vantaggi e i limiti della tecnologia utilizzata; la descrizione della sequenza di operazioni che vanno dall'acquisizione dell'oggetto con la videocamera alla visualizzazione del modello 3d; una descrizione dell'architettura del software e infine un'analisi illustrata dei risultati ottenuti alla fine del progetto di ricerca.

Indice

Sommario	v
1 Introduzione	1
1.1 Scenario attuale	1
1.2 Descrizione del problema	2
1.3 Difficoltà e priorità del progetto	2
1.4 Letteratura sull'argomento	3
2 Tecnologia di acquisizione	5
2.1 Principio di funzionamento	5
2.2 Limiti e difficoltà tecniche	6
2.3 Vantaggi della tecnologia TOF	8
3 3D Model Acquisition pipeline del software	11
3.1 Introduzione	11
3.2 Acquisizione dati, segmentazione e filtraggio	13
3.3 Allineamento	16
3.4 Merging	18
3.5 Visualizzazione	19
4 Descrizione del programma	25
4.1 Introduzione	25
4.2 Scelte di implementazione	25
4.3 Architettura del software	25
4.3.1 Ut3D	27
4.3.2 ToolAcquisizione	28
5 Analisi dei risultati	39
5.1 Configurazione di prova	39
5.2 Rumore sui dati di profondità	39
5.3 Limiti di applicabilità	42

5.4	Ricostruzione 3D di un oggetto	45
5.5	Ricostruzione 3D di una parete	55
5.6	Ricostruzione 3D di una mano	62
6	Conclusioni	65
7	Appendice	67
7.1	Antidistorsione e conversione da range map a nuvola di punti . .	67
8	Ringraziamenti	69
	Bibliografia	71

Capitolo 1

Introduzione

1.1 Scenario attuale

Negli ultimi vent'anni, il desiderio di ridurre l'intervento umano nel processo di costruzione di scene tridimensionali virtuali, la disponibilità nel mercato consumer di hardware sufficientemente potente per la visualizzazione grafica, la crescente potenzialità e i costi decrescenti dei sensori controllati dai computer, hanno allargato da un lato il campo di ricerca della computer graphic, dall'altro ampliato l'utenza raggiungibile nel mercato. Dal momento che gli oggetti che figurano nelle immagini virtuali sono spesso oggetti che appartengono al mondo reale, si è affermato in questi ultimi anni un forte interesse nell'acquistare geometrie direttamente da oggetti reali per la costruzione di scene virtuali.

Il progetto di ricerca di questa tesi verte sulla ricostruzione tridimensionale di oggetti a partire da un'acquisizione con una videocamera dotata di sensore a tempo di volo ("TOF", per brevità). Questa tecnologia trova applicazione in diversi campi: computer vision (video-sorveglianza, rilevamento automatico di collisioni nei veicoli), e-commerce (visualizzazione 3d dei modelli in catalogo), intrattenimento (generazione di scene virtuali sempre più simili alla realtà, nuove interfacce e sistemi di interazione), robotica (ricostruzione di ambienti e navigazione autonoma), architettura (costruzione di modelli di edifici e ponti per l'analisi strutturale, l'ispezione e la ristrutturazione), medicina (diagnosi e pianificazione chirurgica), arte e istruzione (digitalizzazione in modelli 3d di statue).

1.2 Descrizione del problema

A partire da un insieme di frame, che contengono informazioni di profondità dei pixel dell'immagine, viene ricostruito e visualizzato a display, in tempo reale e in modo incrementale, il corrispondente modello tridimensionale dell'oggetto.

L'intervento umano è richiesto solo nella fase di scanning, in cui l'oggetto viene ripreso manualmente con la videocamera dall'operatore. Non sono necessari dunque complessi sistemi di tracking o di set-up, contrariamente a quanto avviene con altri scanner 3d più costosi come gli scanner laser.

1.3 Difficoltà e priorità del progetto

Le principali difficoltà del progetto sono state le seguenti:

1. Scarsa accuratezza della tecnologia usata

I limiti principali della videocamera TOF impiegata sono una limitata risoluzione video X-Y e scarsa affidabilità e ripetibilità delle misurazioni, affette da rumore significativo sui dati di profondità e fenomeni di sbilanciamento delle misurazioni.

2. Realizzazione real-time della ricostruzione 3d

Molti approcci al problema sono stati percorsi negli ultimi dieci anni, tuttavia la ricostruzione 3d in tempo reale rimane una sfida piuttosto aperta. Acquisire e visualizzare contemporaneamente il modello 3d dell'oggetto permette di verificare "on-fly" che l'intera superficie dell'oggetto sia stata scansionata nella sua interezza, fornendo all'utente un feedback immediato sulla generazione del modello 3d. In un approccio non real-time vi è invece la necessità di una seconda scansione per ricostruire le parti della superficie eventualmente non catturate da una prima scansione.

La videocamera inoltre acquisisce 176×144 punti fino 54 volte al secondo, ovvero 25344 punti ogni 18 ms. Realizzare l'intera pipeline, dall'acquisizione alla visualizzazione, in tempo reale si scontra con la necessità di elaborare grandi quantità di dati da processare velocemente in streaming.

In vista delle difficoltà progettuali elencate sopra, le priorità di design che si sono imposte al progetto sono state le seguenti:

1. Velocità e accuratezza

Se la finalità è quella di processare grandi quantità di dati in real-time, la priorità maggiore del progetto è la scelta di strutture dati efficienti e la velocità di computazione a tutti i livelli della pipeline.

I risultati inoltre devono essere sufficientemente accurati da rendere adoperabili i modelli 3d forniti dal programma al termine della sua esecuzione.

2. Automazione

Come già anticipato, l'operatore umano si occupa solo della fase di acquisizione, in cui l'oggetto viene ripreso con la videocamera manualmente e liberamente da angolazioni diverse. Nel caso in cui l'utente noti "buchi" sulla superficie del modello, basterà riprendere la zona interessata con la videocamera e la superficie verrà ricostruita.

1.4 Letteratura sull'argomento

La letteratura sull'argomento è vastissima, in particolare se consideriamo i paper apparsi dal 1991 in poi. Molti degli articoli trattano tuttavia sistemi di acquisizione differenti dai sensori a tempo di volo, tecnologia sviluppata negli ultimi dieci anni e che presenta caratteristiche specifiche da considerare nella fase progettuale [1].

Una realizzazione in tempo reale della pipeline di ricostruzione del modello 3d è stata realizzata da Rusinkiewicz nel 2001 [2], tuttavia questo progetto utilizzava un sistema di acquisizione dato da un proiettore attivo e una videocamera, di natura dunque molto differente rispetto alle videocamere a tempo di volo.

Shuon e Theobalt [3] hanno sviluppato un algoritmo che è in grado di aumentare la risoluzione di range map [4, 5] e ridurre il rumore, il limite principale di questa tecnologia. Tuttavia i tempi di computazione richiesti rendono tali operazioni assolutamente non applicabili ad un contesto real-time e dunque si sono resi necessari altri approcci per la risoluzione del problema.

Capitolo 2

Tecnologia di acquisizione

2.1 Principio di funzionamento

I sistemi a tempo di volo si basano su un principio di funzionamento abbastanza semplice e simile a quello dei RADAR: emettono un segnale infrarosso modulato sinusoidalmente e stimano la profondità dei dati misurando la differenza di tempo tra la trasmissione del segnale e la sua ricezione, più precisamente dalla differenza di fase tra il segnale inviato e il segnale riflesso dalla superficie [6, 7]. I sistemi TOF necessitano dunque di grande precisione nelle misurazioni temporali, da cui dipende l'accuratezza della misura di profondità.

I sistemi a tempo di volo sono composti essenzialmente da una matrice di trasmettitori sincronizzati che emettono il segnale infrarosso e una matrice di ricevitori integrati in un sensore CCD/CMOS.

Per la tesi è stato impiegato un prototipo Mesa [8] SwissRanger4000 (in figura 2.1), che può raggiungere un frame-rate massimo di 54 frame per secondo.

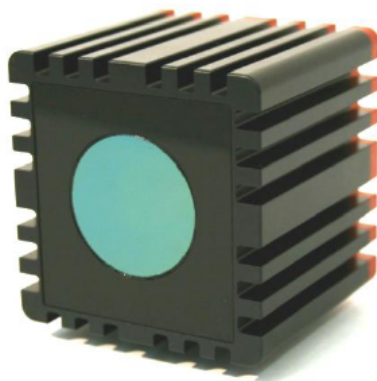


Figura 2.1: Mesa SR4000

2.2 Limiti e difficoltà tecniche

Dato che l'utilizzo di una videocamera TOF è semplice quanto usare una comune videocamera 2d, vi è un enorme potenziale per lo sviluppo di uno strumento "user-friendly" per la misurazione 3d di scene e oggetti. Tuttavia, le particolarità del metodo utilizzato per estrarre informazioni di profondità solleva problematiche non trascurabili riguardo la qualità delle misurazioni ottenute, e di questo è necessario tenere conto in fase di analisi dei risultati. In particolare si elencano:

- Limitata risoluzione X-Y

Le videocamere TOF attualmente nel mercato presentano una risoluzione massima di 320×240 (videocamere Canesta), mentre la videocamera Mesa SwissRanger4000 presenta una risoluzione di 176×144 . Ciò si traduce in una generale povertà di informazione sulla geometria della scena e delle superfici.

- Scarsa accuratezza e rumore

In aggiunta al rumore aleatorio [6, 7] sui dati di profondità (rumore termico, rumore di quantizzazione della distanza, rumore shot tipico dei dispositivi a ricezione di fotoni), i sensori TOF sono affetti anche da sbilanciamenti sistematici sulle misurazioni.

Il modello di rilevazione della distanza si basa infatti sull'assunzione che il segnale trasmesso viaggi direttamente dai led emettitori della videocamera all'oggetto, per poi essere riflesso dall'oggetto al sensore della videocamera, cosicché la distanza totale percorsa dal segnale sia esattamente il doppio della distanza tra la videocamera e l'oggetto. Tuttavia l'oggetto può essere disposto nella scena in modo tale che il segnale percorra una direzione diversa da quella sopra descritta: il raggio emesso dalla videocamera può essere riflesso da un primo oggetto prima di essere riflesso dall'oggetto di interesse. In questa situazione il raggio percorre una direzione diretta e una indiretta (questo fenomeno è chiamato "multipath error" o "scattering"). Il risultato finale è dunque una misurazione sovrastimata della distanza.

Un'altra situazione comune di multipath avviene quando si misurano oggetti con geometrie concave, ad esempio due mura incidenti. In questo caso, come mostrato in figura 2.2, la distanza della regione del muro adiacente all'angolo sarà maggiore rispetto a quella reale, poiché una non trascurabile percentuale del segnale IR è riflessa sia dal muro in oggetto che dall'altro muro adiacente.

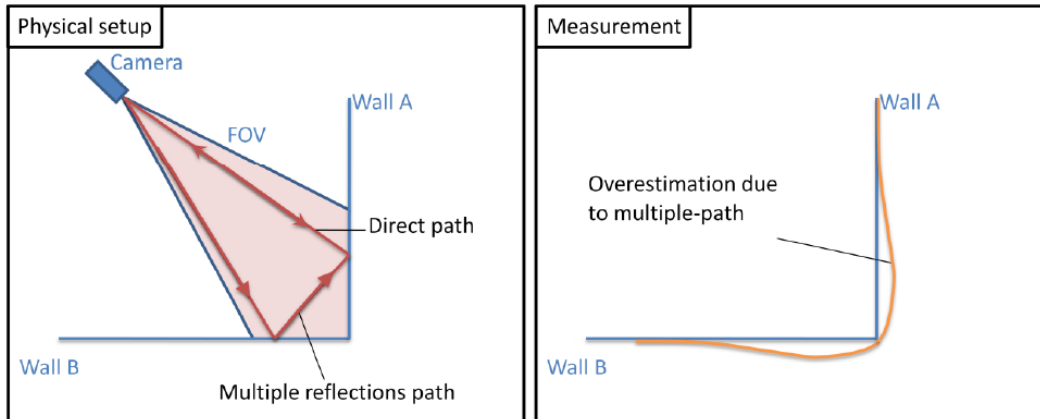


Figura 2.2: Scattering dei sensori TOF

Durante il processo di acquisizione possono verificarsi due tipi di “saturazione”, una relativa ad una eccessiva potenza associata all’illuminazione di fondo e una relativa ad una eccessiva ampiezza del segnale ricevuto, dovuto all’eccessiva riflettanza o vicinanza dell’oggetto misurato: superfici speculari tendono facilmente alla saturazione in ampiezza, mentre superfici con alto grado di assorbimento come stoffa o affini (specie se di colore nero) attenuano eccessivamente il segnale. Inoltre per oggetti molto piccoli il rapporto segnale-rumore è significativamente basso, rendendo difficile la stima accurata della geometria.

Un’altra problematica dei sensori TOF è la stima corretta della profondità dei bordi di un oggetto: i punti appartenenti a superfici che presentano un angolo di curvatura troppo ampio con la direzione della camera sono affetti da scarsa affidabilità nella stima della loro distanza. Parte del raggio IR incidente viene infatti disperso e il raggio riflesso dalla superficie giunge al sensore con una potenza notevolmente inferiore, di conseguenza la stima dei punti non è precisa.

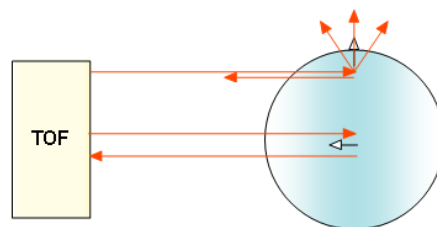


Figura 2.3: Attenuazione e riflessione del segnale IR sui bordi di un oggetto

Per quanto riguarda le prestazioni nominali, Mesa, il costruttore del sensore SR4000, dichiara che la ripetibilità dello SR4000 è caratterizzata da una deviazione standard inferiore a 5 mm e la sua precisione assoluta può avere al massimo uno scarto di 1 cm rispetto all'effettiva distanza dell'oggetto, nel caso in cui le misure vengano effettuate nelle condizioni di illuminazione e riflettanza di oggetti migliori possibili. Per diminuire la deviazione standard si può applicare al segnale ricevuto un filtro che effettua una media sia spaziale che temporale, o anche agire sul tempo di integrazione, ossia il tempo di esposizione dei sensori. La ripetibilità della misura, e quindi l'inverso della deviazione standard dell'errore, crescono con l'aumentare del tempo di integrazione. Se il tempo di integrazione è però troppo elevato, c'è il rischio che il sensore saturi. Per questo motivo, si ha che la ripetibilità è ottimizzata se il tempo di integrazione è il più grande possibile, senza che però sia presente saturazione. Il comportamento dello SR4000 risulta essere molto meno ideale rispetto alla descrizione nominale fornita dal costruttore, sia in termini di ripetibilità che di accuratezza assoluta. Queste considerazioni legate ai limiti di operabilità delle videocamere TOF vanno tenute ben presenti durante il loro utilizzo e per una migliore comprensione dei risultati ottenuti.

2.3 Vantaggi della tecnologia TOF

Nonostante i limiti attuali, le videocamere TOF rappresentano oggi una tecnologia ricca di potenzialità e vantaggi rispetto ad altre soluzioni presenti sul mercato. Tra i motivi di interesse elenchiamo i seguenti:

- Facilità d'uso

Le camere TOF risultano fisicamente molto compatte, non richiedono complesse operazioni di set-up né operatori specializzati per il loro utilizzo, contrariamente a tecnologie che presentano accuratezza superiore come gli scanner laser.

- Alta frequenza di aggiornamento

Possono estrarre informazioni di profondità in tempo reale ad alta frequenza di aggiornamento, ciò le rende particolarmente indicate per scene dinamiche.

- Costi contenuti

Sono dispositivi a basso costo e dunque hanno buone potenzialità di diffusione nel mercato di massa.

- Minore sensibilità alla luce

Poiché la tecnologia utilizza un segnale nello spettro dell'infrarosso, le rilevazioni sono relativamente più insensibili allo spettro del visibile rispetto ad altre soluzioni (ad esempio i sistemi stereo 2D).

Capitolo 3

3D Model Acquisition pipeline del software

3.1 Introduzione

Nonostante vi siano molti strumenti per acquisire dati 3d, dalle videocamere TOF agli scanner laser, esiste una sequenza di operazioni di base che vanno dall'acquisizione dei dati 3d alla visualizzazione del modello 3d corrispondente. Questa serie di operazioni nella letteratura prende il nome di "3D Model Acquisition pipeline", espressione tratta dall'articolo di Fausto Bernardini and Holly Rushmeier [9].

La model acquisition pipeline assume, generalizzando, che il dispositivo di scanner 3d produca un output sotto forma di "range map", ovvero una immagine nella quale ciascun pixel contiene la distanza dal sensore ad un punto visibile della scena, invece della sua luminosità. Una immagine di distanza è costituita da misure (discrete) di una superficie 3D rispetto ad un piano 2D (il piano immagine del sensore).

Il recupero della informazione 3D non esaurisce però il processo di acquisizione della forma, per quanto ne costituisca il passo fondamentale. Per ottenere un modello completo di un oggetto, o di una scena, servono molte viste 3d, che devono essere allineate e "fuse" tra di loro, ottenendo una superficie 3D finale. Nel programma sviluppato in questa attività di tesi, la ricostruzione del modello dell'oggetto a partire dai dati restituiti dalla videocamera TOF prevede le seguenti fasi:

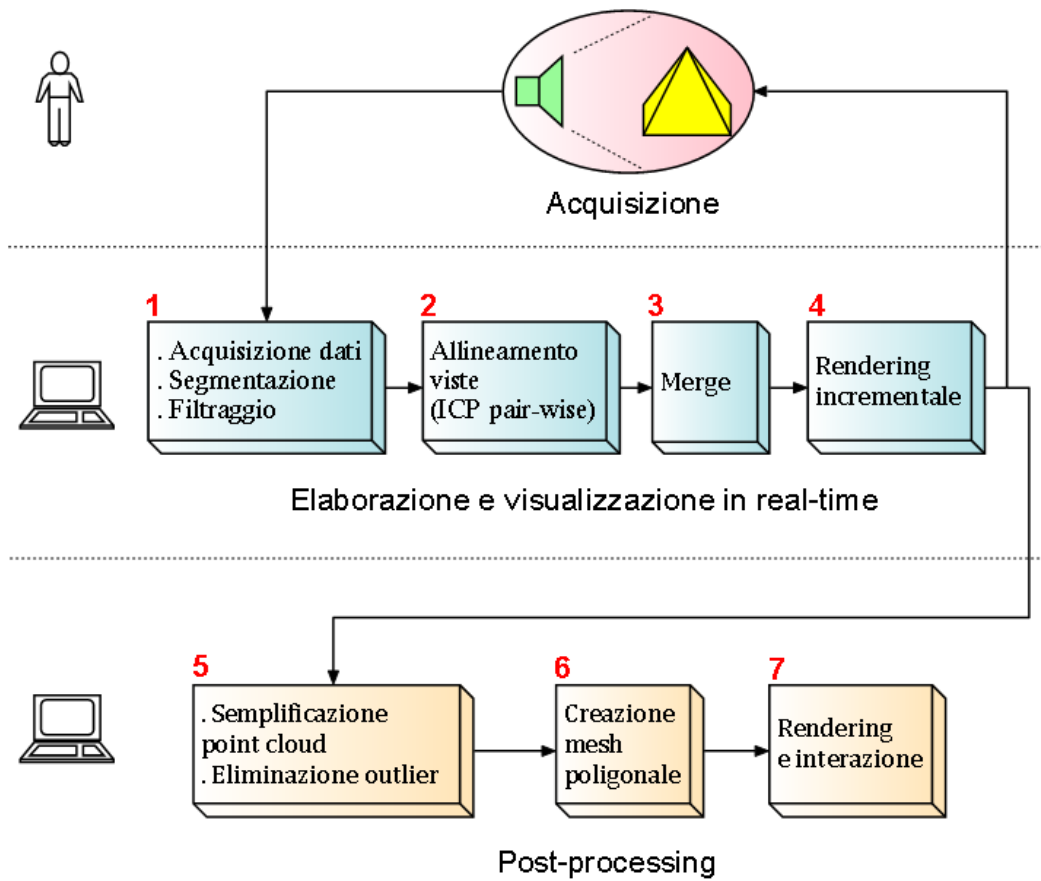


Figura 3.1: Pipeline realizzata dal software

La figura 3.1 sottolinea due aspetti importanti della pipeline. Il primo riguarda il ruolo dell'utente nel programma: l'intervento dell'operatore umano è richiesto solo nella fase di scansione dell'oggetto con la videocamera (eventualmente l'utente potrà interagire con il modello poligonale finale per osservarne meglio la geometria da punti di vista arbitrari), mentre le rimanenti operazioni sono completamente automatizzate dal programma. Come verrà mostrato in seguito, dall'interfaccia grafica sarà possibile variare alcuni parametri del software per meglio adattare l'esecuzione delle operazioni alle istanze in ingresso. Il secondo aspetto messo in evidenza dalla figura è il ciclo che rappresenta la successione di frame processati dal programma in streaming e in tempo reale: tutte le operazioni che vanno dall'acquisizione dei dati alla visualizzazione vengono eseguite per ogni iterazione del ciclo, mentre il filtraggio del modello finale e la creazione della mesh poligonale appartengono alla fase di post-processing.

Vediamo ora nel dettaglio le varie operazioni che hanno luogo durante l'esecuzione del programma.

3.2 Acquisizione dati, segmentazione e filtraggio

L'output della camera SR4000 è un insieme di range map, dalle quali è possibile ricavare la nuvola di punti nello spazio 3d utilizzando i parametri intrinseci della videocamera. I dati forniti dalle range image inoltre sono afflitti da distorsione radiale e tangenziale, per tanto è necessario un ulteriore passo di compensazione della distorsione invertendo la matrice dei parametri intrinseci della videocamera. Per illustrare come effettuare questa operazione, è necessario descrivere il problema della determinazione dei parametri intrinseci del modello (calibrazione). Si rimanda il lettore al breve paragrafo relativo descritto in appendice.

La segmentazione dell'oggetto dallo sfondo è necessaria per distinguere quali dei punti sono di interesse per la ricostruzione 3d dell'oggetto e quali invece appartengono al background dell'immagine.

Per motivi di efficienza e velocità di computazione, questa operazione è stata implementata con un thresholding sulla coordinata "z": i punti con una distanza superiore ad una certa soglia, impostata a priori dall'utente, verranno esclusi dai successivi stadi della pipeline. Il thresholding è ottimale in presenza di una certa distanza tra l'oggetto e lo sfondo: in questo modo la segmentazione è sufficientemente accurata da non escludere punti effettivamente appartenenti

alla geometria dell'oggetto che si vuole riprendere.

Il processo di eliminazione degli outlier è costituito da due stadi successivi. La prima ha luogo direttamente dopo la fase di acquisizione dei dati dalla videocamera (in figura 3.1, blocco 1) ed ha complessità computazionale lineare. Il secondo stadio di eliminazione di outlier, come verrà spiegato in seguito, richiede tempi di esecuzione superiori, perciò appartiene alla fase di post processing (in figura 3.1, blocco 5).

Nella prima fase vengono filtrati i punti dell'immagine che presentano valori di confidenza e ampiezza al di sotto delle soglie impostate inizialmente dall'utente: i punti con valori di confidenza e ampiezza inferiori a queste soglie vengono considerati come risultato di misurazioni affette da scarsa affidabilità e robustezza, e dunque ignorati.

Vengono inoltre ignorati i punti che presentano un angolo troppo grande tra la normale del punto e la media delle normali dei punti che hanno superato il precedente thresholding. Questo filtraggio limita parzialmente il problema evidenziato nel capitolo precedente, riguardante la corretta stima di profondità dei bordi di un oggetto da parte dei sensori TOF.

Nella seconda fase (post-processing), viene applicato il seguente algoritmo:

Input : La nuvola di punti 3D, i parametri: “*radius*” e “*min_neighbors*”

Output : La nuvola di punti 3D filtrata

Esecuzione :

1. i punti vengono memorizzati in una struttura dati efficiente per l'operazione di ricerca, il kd-tree [10];
2. per ogni punto viene effettuata una ricerca in un intorno sferico di raggio *radius* (range search [11, 10]);
3. se il punto presenta un numero di vicini inferiore alla soglia *min_neighbors*, verrà considerato outlier e per tanto verrà eliminato dalla nuvola di punti.

Questo algoritmo presenta una complessità $O(n \log n)$ nella fase di costruzione del kd-tree (pre-processing), mentre una singola operazione di ricerca dei vicini costa $O(n^{\frac{2}{3}} + m)$ (con m numero dei vicini trovati) [12, 11].

I parametri richiesti dall'algoritmo sono solo due: il raggio della sfera, che rappresenta l'intorno del punto nello spazio 3D, e il numero minimo di vicini che un punto deve avere per essere classificato come inlier.

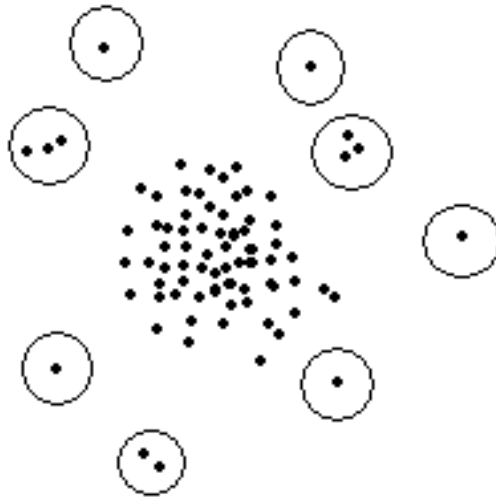


Figura 3.2: Procedura di eliminazione outlier. I punti cerchiati rappresentano outlier

Nella fase iniziale di filtraggio è stata adottata una politica di eliminazione degli outlier piuttosto aggressiva in considerazione del fatto che uno dei vantaggi operativi della tecnologia time-of-flight è l'alta frequenza di acquisizione delle immagini: i dettagli della superficie che vengono erroneamente filtrati in un frame possono essere opportunamente “reintegrati” nel frame successivo, se le misure di affidabilità superano le soglie di ampiezza e confidenza. Dal momento che la visualizzazione del modello 3d avviene in tempo reale, l'utente ha la possibilità di tornare a riprendere con la videocamera punti della superficie rimasti scoperti da una prima scansione, incrementando quindi l'informazione visiva della superficie gradualmente e in modo incrementale.

Minore assenza di rumore nei dati consente inoltre un'applicazione più accurata della successiva fase di allineamento delle viste, nonché un minore numero di dati da gestire per le successive operazioni. L'eliminazione degli outlier tuttavia non deve escludere troppi punti, altrimenti la successiva operazione di allineamento non possiede sufficienti informazioni per eseguire correttamente le elaborazioni.

Un'altra funzione che il programma consente di eseguire in real-time è l'applicazione, sui dati delle range map, di un filtro bilaterale [13] nello spazio 3D, capace di ridurre sensibilmente la rumorosità dei dati sulle stime di profondità e allo stesso tempo preservare gli edge della superficie.

Se tradizionalmente nelle immagini bidimensionali il kernel gaussiano tiene conto della vicinanza dei pixel e di una funzione di “somiglianza” delle in-

tensità di colore, nello spazio 3D si considera la distanza euclidea tra i punti e, come “somiglianza”, la differenza di profondità dei punti. Il programma inoltre consente di selezionare la funzione “distanza” diversamente da quanto descritto sopra, come numero di pixel che separano due punti della range map. Il peso assegnato al punto \vec{p}_{ij} , appartenente alla finestra di \vec{p} , è il seguente:

$$w_{ij} = e^{\left(-\frac{1}{2} \cdot \left(\frac{\text{dist}(\vec{p}_{ij}, \vec{p})}{\sigma_d} \right)^2 - \frac{1}{2} \cdot \left(\frac{z(\vec{p}_{ij}) - z(\vec{p})}{\sigma_r} \right)^2 \right)}$$

Il bilateral filter può essere applicato nell’elaborazione real-time del programma: una finestra 5×5 applicata a 5000 punti impiega circa 10 millisecondi di tempo. Inoltre il bilateral filter, migliorando la qualità dei dati, può rendere più veloce la successiva operazione di allineamento delle viste con ICP (è necessaria una attento bilanciamento dei parametri dell’algoritmo per migliorare significativamente la qualità dell’immagine finale).

3.3 Allineamento

I sensori TOF non catturano completamente la forma di un oggetto con una sola immagine, ne servono molte, ciascuna delle quali cattura una parte della superficie dell’oggetto. Le porzioni della superficie dell’oggetto che si ottengono da diverse viste sono espresse ciascuna nel proprio sistema di riferimento (legato alla posizione del sensore). Lo scopo della registrazione è di portarle nello stesso sistema di riferimento tramite una opportuna trasformazione rigida (rototraslazione nello spazio 3D). Se la posizione ed orientazione del sensore sono note (perché vengono rilevate da un sistema di tracking oppure perché esso è controllato meccanicamente), il problema si risolve banalmente. Nel nostro caso la videocamera (o l’oggetto da acquisire) sono mossi liberamente dall’operatore e dunque la posizione e l’orientazione del sensore non sono noti a priori. Si è reso necessario dunque l’uso dell’algoritmo di allineamento ICP (iterative-closest-point) [14, 15].

In ingresso sono date le posizioni dei punti dell’oggetto rispetto a due diversi sistemi di riferimento 3D, in corrispondenza a due frame successivi dello stream di dati; l’algoritmo ICP calcola la trasformazione rigida che porta una nuvola di punti (“source”) ad “allinearsi” con l’altra (“target”). È fondamentale notare che non sono note a priori le corrispondenze dei punti da allineare, ma ICP risolve simultaneamente il problema delle corrispondenze e la stima della trasformazione rigida.

L’algoritmo ICP implementato dal programma si riassume così:

Input : Due nuvole di punti 3D, S (*source*) e T (*target*), i parametri: “ n_s ”, “ max_RMS ”, “ max_iter ”

Output : Matrice di rototraslazione che allinea S su T

Esecuzione :

1. Scelti n_s punti di S nella range map attraverso il criterio dell'uniform sampling, trova i più vicini punti corrispondenti in T;
2. Con le corrispondenze trovate al passo 1., calcola la trasformazione incrementale con la procedura SVD[16, 17] (singular value decomposition);
3. Applica la trasformazione trovata nel passo 2 a tutti i punti di S;
4. Se la media dell'errore quadratico (RMS) è minore della soglia max_RMS , o se l'algoritmo è giunto al suo massimo numero di iterazioni di convergenza impostato dall'utente (max_iter), termina l'esecuzione, altrimenti vai al passo 1.

Questo metodo, come dimostrato da Besl e McKay [14], convergerà al più vicino minimo locale della somma delle distanze al quadrato tra i punti più vicini.

La fase di maggior costo computazionale è la ricerca del closest-point. Un metodo ingenuo richiede, per ciascun punto, tempo $O(n)$, dove n è la cardinalità di T. Il programma invece usa una struttura dati efficiente, il kd-tree, che permette di abbassare a $O(\log n)$ la complessità della ricerca del closet-point di un punto. La fase 1. dell'algoritmo ha dunque complessità $O(n_s \log n)$ anziché $O(n_s n)$ (rispetto all'implementazione base, in termini di velocità di esecuzione si ottiene un fattore di speed-up pari a 4 per un'istanza di 10000 punti).

Per quanto riguarda l'applicazione dell'algoritmo ICP nel progetto, le due esigenze opposte da coniugare sono velocità di esecuzione e robustezza della stima della rototraslazione. In questo trade-off è opportuno tenere a mente le specificità dell'applicazione in cui viene impiegato ICP. La camera TOF è in grado di generare frame fino a 54 volte al secondo, per tanto i frame successivi presentano verosimilmente vaste zone di sovrapposizione (“overlap”): ciò favorisce il rate di convergenza dell'algoritmo ICP, dunque il suo tempo di esecuzione. I dati tuttavia sono affetti da rumore sull'asse z, perciò è necessario forzare l'esecuzione dell'algoritmo per un numero sufficiente di iterazioni e migliorare la robustezza della stima della trasformazione rigida che allinea i frame.

La registrazione di più di due insiemi di punti in un unico sistema di riferimento è stato ottenuto dalla concatenazione di registrazioni di coppie di frame

consecutivi (ICP pair-wise). Tale procedimento è stato adottato perché particolarmente indicato dal punto di vista della velocità di esecuzione. Il costo però di questa scelta è l'introduzione di un errore incrementale nella stima delle rototraslazioni, che non conduce alla soluzione globale ottimale. In assenza di un algoritmo di allineamento globale (come quello presentato da Pulli [18]), che operi tuttavia in tempi decisamente più lunghi, è possibile compensare l'effetto dell'errore incrementale adottando delle accortezze pratiche nella fase di scansione dell'oggetto, in primis compiendo rotazioni lente della camera (o dell'oggetto).

Dell'algoritmo ICP esistono numerose varianti [19, 20, 21, 22, 23, 24, 25], che apportano piccole modifiche ai passi fondamentali dell'algoritmo, come metriche di distanza diverse per la funzione residuo o, per il calcolo delle corrispondenze, euristiche particolari nella scelta dei punti da allineare, come la sogliatura sulle distanze o l'eliminazione degli accoppiamenti sui bordi (questa variante è già implicitamente implementata dal programma dopo l'acquisizione dei dati, durante il filtraggio dei punti rispetto alla direzione delle normali). Sebbene queste varianti possano teoricamente migliorare la velocità o la robustezza dell'implementazione di base di ICP, è altrettanto vero che queste euristiche possono fallire con le istanze specifiche fornite da una telecamera con sensore a tempo di volo, che presenta in uscita range map affette da significativo rumore sull'asse z . Inoltre frame successivi, come già specificato, presentano idealmente vaste zone di overlap della superficie, per tanto non si sono rese indispensabili euristiche particolari che avrebbero potuto manifestare risultati contrari alle aspettative.

3.4 Merging

Il processo di merging consiste nel fondere i dati 3D in una singola rappresentazione geometrica.

Questa operazione, similmente al processo di rimozione dei punti outlier, avviene in due punti differenti della pipeline, appena dopo la fase di allineamento e nella fase di post processing. In entrambi i casi, operando su istanze rappresentate da nuvole di punti 3D, non abbiamo alcuna informazione di connettività tra i punti.

Nella prima fase il merging consiste semplicemente nel mantenere in memoria il modello tridimensionale incrementale, costituito dalle viste registrate dall'algoritmo ICP per ogni iterazione. Per ottenere ciò è stata utilizzata un'operazione di "append" che mantiene in memoria la lista dei punti allineati delle

varie viste.

Nella fase di post processing invece, la lista dei vertici accumulata dall'appendere viene epurata dai punti duplicati, ossia punti che presentano le medesime coordinate (x,y,z) . Impostando una soglia di tolleranza diversa da zero, è anche possibile eliminare i punti con una distanza inferiore alla soglia impostata (i punti eliminati vengono considerati dunque come punti duplicati). Dopo questa elaborazione ha luogo il processo di eliminazione degli outlier descritto in precedenza, bastato sulla procedura di range search nel kd-tree.

Infine si passa da una rappresentazione in nuvola di punti 3d a mesh poligonale. Questo è il problema della ricostruzione della superficie, che si può formulare come la stima di una varietà bidimensionale che approssimi la superficie dell'oggetto incognito da un insieme di punti 3D campionati. Il programma permette di scegliere tre algoritmi di ricostruzione della superficie a partire da una nuvola di punti: "Poisson Surface Reconstruction" [26], "PowerCrust Surface Reconstruction" [27] e "Hoppe Surface reconstruction" [28]. Poiché questi algoritmi sono stati già implementati e poiché non costituiscono un argomento di interesse preminente per la tesi, si rimanda la lettura degli stessi nei relativi articoli citati in bibliografia.

Dopo questa operazione, è possibile inoltre attivare una funzione di smoothing laplaciano sulla mesh poligonale risultante, ottenendo un effetto di "rilassamento" della superficie ma anche una generale perdita di dettagli sulla geometria dell'oggetto.

3.5 Visualizzazione

L'obiettivo del rendering durante l'esecuzione real-time della pipeline è una visualizzazione quanto più rapida possibile dei dati a disposizione, a patto di una generale perdita di accuratezza nella rappresentazione visiva del modello geometrico.

A tal fine è stato scelto una modalità di rendering che visualizza semplicemente i punti con una data grandezza scelta dall'utente. I punti dunque verranno rappresentati come quadrati ("*splat*") di lato k pixel. La variabile k assume dunque implicitamente il significato di "risoluzione" con cui visualizzare la nuvola di punti 3d. In questo modo inoltre, dopo alcune iterazioni, punti relativamente vicini (ossia con una distanza prossima a k) daranno l'impressione di appartenere ad un'unica superficie.

Per quanto riguarda la colorazione, è innanzitutto necessario constatare che la videocamera SwissRanger4000 non mette a disposizione informazioni di colore ma solo dati di profondità. Per ottenere una rappresentazione grafica che evidenzi quantomeno le variazioni di gradiente date dalle curvature delle superfici è necessario avere informazioni di connettività dei punti, in modo tale da stimare la normale associata ad ogni punto.

Il calcolo delle normali viene eseguito in modo efficiente (complessità lineare) appena dopo la fase di acquisizione dei dati dalla videocamera, ed opera direttamente sulle informazioni fornite dalle range map.

L'algoritmo impiegato per il calcolo delle normali, illustrato in figura 3.3, è il seguente:

Input : Range map (con punti memorizzati in ordine row-major)

Output : Range map con le normali associate a ciascun punto

Esecuzione :

1. Un punto \vec{P} interno alla range map, di indice (r, c) , ha quattro vicini: un vicino "nord" (N), di indice $(r - 1, c)$, un vicino "est" (E), di indice $(r, c + 1)$, un vicino "sud" (S), di indice $(r + 1, c)$ e un vicino "ovest" (O), di indice $(r, c - 1)$. Dato \vec{P} , calcola i vettori che collegano il punto \vec{P} ai punti adiacenti come differenza vettoriale tra la coordinata del vicino meno la coordinata di \vec{P} (in questo modo la differenza è un vettore che va da \vec{P} al vicino). Si ottengono così i vettori differenza \vec{a} , \vec{b} , \vec{c} , \vec{d} in figura 3.3.
2. Calcola il prodotto vettoriale tra i vettori differenza adiacenti ($\vec{b} \times \vec{a}$, $\vec{c} \times \vec{b}$, $\vec{d} \times \vec{c}$, $\vec{a} \times \vec{d}$) e infine calcola la media dei vettori ottenuti come somma vettoriale. Il vettore risultante rappresenta l'orientazione della normale nel punto considerato.

I vantaggi dell'applicazione di questo metodo anziché metodi più complessi (come la stima del *best fitting plane* tramite RANSAC) sono molteplici:

1. L'algoritmo, di complessità lineare, risulta estremamente veloce (circa 5 ms con un'istanza di 10000 punti);
2. Si basa su informazioni locali dei punti e non necessita della stima di piani passanti per un dato intorno del punto (come nel metodo di RANSAC);
3. Le orientazioni delle normali si rivelano sempre coerenti: non si verifica il fenomeno delle normali orientate in senso opposto a quello reale ("*flipping*");

4. È un algoritmo robusto al rumore sull'asse z , perciò particolarmente indicato per il progetto.

L'algoritmo necessita che i dati memorizzati nelle range map siano disposti in ordine row-major, in modo tale che l'accesso alle informazioni sui punti adiacenti avvenga per mezzo di una semplice indicizzazione.

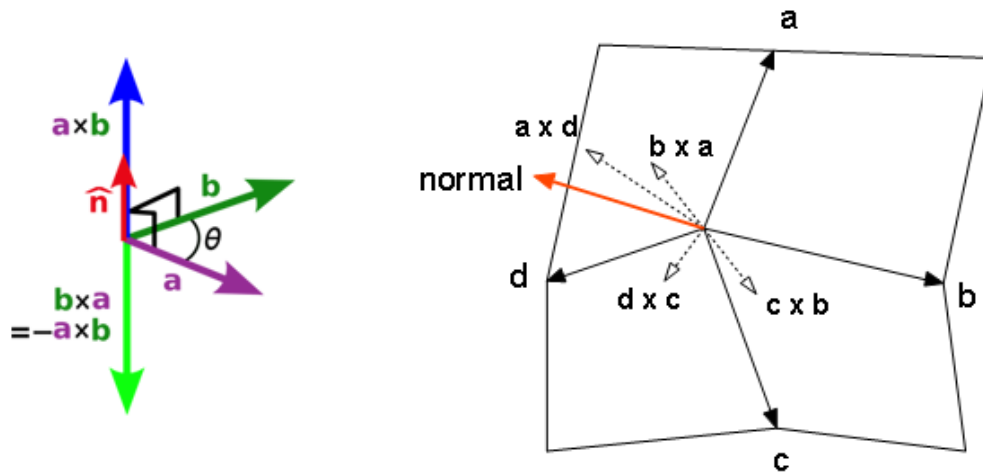


Figura 3.3: Calcolo della normale di un punto

Note le normali, è possibile associare un colore a ciascun punto in base all'orientazione della normale e della camera virtuale (come in figura 3.4): viene calcolato il valore del coseno dell'angolo tra la normale e il vettore che collega il punto alla posizione della camera, se questo è prossimo 1, il punto verrà colorato di bianco, se il coseno è prossimo a 0, il punto tenderà al nero.

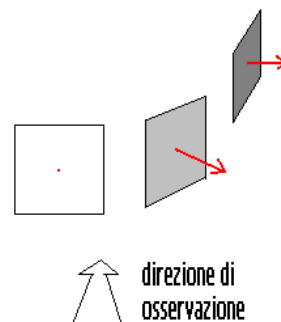


Figura 3.4: Colorazione delle splat in base all'orientazione della normale

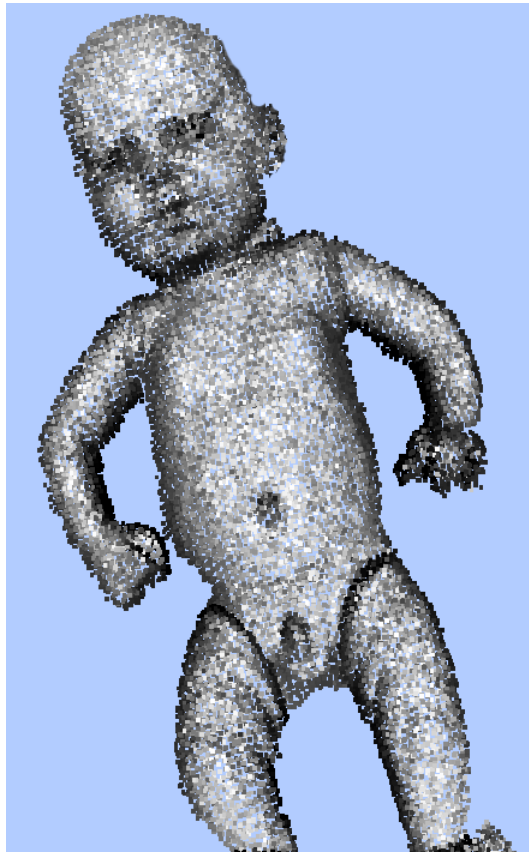


Figura 3.5: Visualizzazione di un modello completo

Un'ulteriore considerazione riguarda la posizione della camera virtuale e la sua orientazione nello spazio tridimensionale. L'obiettivo della visualizzazione è mostrare in tempo reale, durante l'acquisizione, l'evoluzione del modello tridimensionale dell'oggetto: si ritiene dunque necessario aggiornare la posizione della camera virtuale per far sì che la superficie visualizzata corrisponda esattamente alla porzione dell'oggetto che si sta scansionando.

Il software risolve questo problema posizionando il focus della camera virtuale in corrispondenza del centroide della nuvola di punti da visualizzare e con orientazione (direzione camera-focus) coincidente con la media delle normali della nuvola di punti da visualizzare. In questo modo la camera virtuale si pone sempre "frontalmente" rispetto alla porzione della superficie appena acquisita.

Un'ulteriore accortezza consiste nel rendere sempre visibile la porzione di interesse, senza che questa fuoriesca dalla finestra di visualizzazione, adattando la distanza della videocamera in base all'estensione della diagonale della boun-

ding box della nuvola di punti da visualizzare. Questa operazione evita che parte dei punti fuoriescano dal campo visivo della camera virtuale e allo stesso tempo permette di inquadrare con il giusto grado di dettaglio la porzione di interesse. È possibile inoltre far sì che la visualizzazione segua il comportamento sopra esposto per l'intero modello incrementale visualizzato fino in quel momento anziché per l'ultima nuvola di punti acquisita. In questo caso il programma regola la distanza della camera in base alla diagonale della bounding box dell'intero modello incrementale, non più dell'ultima nuvola di punti dello stream.

Dal momento che l'intera operazione di aggiornamento della camera non coinvolge trasformazioni geometriche del modello incrementale, la funzione si è rivelata estremamente veloce (tempi di esecuzione inferiori al millisecondo, con prestazioni che non degradano all'aumentare del numero di punti del modello) e sufficientemente accurata nell'inquadrare di volta in volta le porzioni di interesse dell'oggetto.

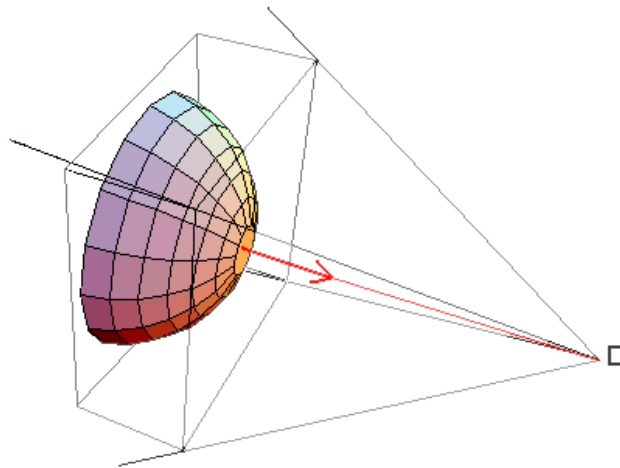


Figura 3.6: Disposizione della camera

Capitolo 4

Descrizione del programma

4.1 Introduzione

Il software sviluppato in questa attività di ricerca implementa l'intera pipeline descritta nel capitolo precedente ed aggiunge ulteriori funzionalità che verranno descritte in questo capitolo.

Prima di descrivere l'interfaccia grafica e le sue funzioni, verrà data una panoramica all'architettura del software (la documentazione è disponibile in formato html e latex).

4.2 Scelte di implementazione

La scelta del linguaggio di programmazione è ricaduta sul linguaggio C++, che coniuga velocità di esecuzione, espressività e capacità di manipolazione dei dati a basso livello, nonché ampia compatibilità con molte delle librerie software disponibili in rete.

Sono state impiegate principalmente due librerie di programmazione:

- OpenCV [29, 30], che offre numerose funzioni legate alla computer vision e all'elaborazione delle immagini;
- VTK [31], un ricco framework open-source per l'elaborazione dei dati tridimensionali e la loro visualizzazione.

4.3 Architettura del software

Il software in bundle con la videocamera Mesa mette a disposizione un set di API per l'acquisizione e la manipolazione dei dati forniti dall'hardware (per

una descrizione più precisa delle funzioni messe a disposizione dai driver, consultare il relativo manuale della videocamera Mesa SwissRanger4000 [32]). Sopra questo layer, un wrapper delle API Mesa permette una descrizione ad alto livello delle funzioni svolte dai driver, semplificando così, a livello di scrittura e lettura del codice, il modulo di interfaccia con il dispositivo. Infine nell'application layer risiede il software vero e proprio di acquisizione e ricostruzione 3D, che fa uso dei servizi messi a disposizione nei layer inferiori e allo stesso tempo presenta l'interfaccia grafica per l'utente (quest'ultima è stata sviluppata con le librerie Microsoft MFC [33]).

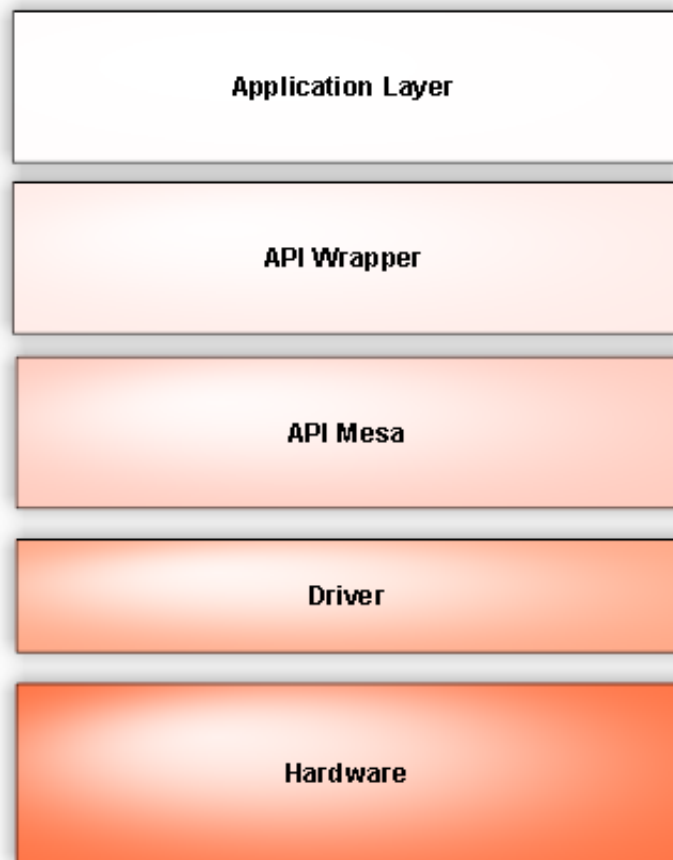


Figura 4.1: Layer del software

I componenti software del progetto sono cinque:

1. “*Ut*”
Un insieme di routine di servizio e di utility per la gestione delle eccezioni.

2. “*Ut3D*”
Una libreria che contiene funzioni utili nello spazio 3D, come filtri, modalità di visualizzazione e l’algoritmo di allineamento dei frame (ICP).
3. “*TrinocularSystem*”
Modulo che contiene un wrapper dei driver, definisce alcune strutture dati comuni all’intero progetto e presenta funzioni utili come l’antidistorsione delle coordinate dei punti.
4. “*ToolAcquisizione*”
Il software che gestisce l’acquisizione e la ricostruzione 3D del modello e che presenta a video i risultati ottenuti.
5. “*ToolCalibrazione*”
Programma per l’estrazione dei parametri intrinseci della videocamera.

Poiché “*Ut*”, “*TrinocularSystem*” e “*ToolCalibrazione*” sono stati già sviluppati in precedenza da un altro dottorando, la loro descrizione verrà tralasciata a favore di una trattazione più approfondita di “*Ut3D*” e “*ToolAcquisizione*”, i moduli software più rilevanti del progetto.

Altri due componenti sviluppati sono un analizzatore del log di testo, per estrarre alcuni dati utili alla formulazione di grafici di analisi, e il tester di ICP, che simula l’esecuzione di *ToolAcquisizione* su istanze di test create da alcuni metodi di *Ut3D*, così è possibile verificare la corretta funzionalità dei filtri e degli algoritmi di allineamento e di visualizzazione senza necessariamente interfacciarsi con la camera.

4.3.1 Ut3D

La libreria Ut3D mette a disposizione principalmente quattro classi di funzioni:

- Funzioni di test
Funzioni attivabili mediante l’istruzione di pre-processore “INCLUDE_TEST_FUNCTIONS”, permettono la creazione e la manipolazione (colorazione, trasformazioni, ecc..) di istanze che simulano le range map fornite dalla videocamera. Questi metodi sono stati impiegati nel tester di cui si fa riferimento nel paragrafo precedente.
- Filtri
La libreria implementa filtri per la rimozione di outlier descritti nel capitolo “3D model acquisition pipeline del software”, un filtro per la rimozione, in una range map, dei triangoli con perimetro superiore ad una

certa soglia (questa funzione non è stata attivamente adoperata dal programma), il bilateral filter nello spazio 3d (con la possibilità di definire, nel kernel gaussiano, la metrica di distanza espressa come distanza euclidea o come distanza in termini di numero di pixel) e infine un filtro che rimuove i punti la cui normale presenta un angolo troppo ampio rispetto ad un vettore dato in ingresso (solitamente la media delle normali di un frame).

- Utility

Metodi per la colorazione delle nuvole di punti (feature utile in fase di debug), per la visualizzazione di forme geometriche in una finestra OpenGL (con la possibilità di interazione tramite un controllo trackball via mouse e tastiera) e per l'aggiornamento della camera virtuale in una scena (consultare il paragrafo “Visualizzazione” nel capitolo “3D model acquisition pipeline del software”).

- Algoritmo di allineamento

L'implementazione dell'algoritmo ICP (iterative-closest-point) descritto nel capitolo “3D model acquisition pipeline del software”.

- Algoritmi di ricostruzione delle superfici

I tre algoritmi, citati nel capitolo precedente, che permettono la ricostruzione di una superficie poligonale a partire da una generica nuvola di punti.

4.3.2 ToolAcquisizione

Come già anticipato, ToolAcquisizione è il software che gestisce l'acquisizione e la ricostruzione 3D del modello e che presenta a video i risultati ottenuti. Rappresenta, di fatto, il componente software principale di questo progetto. L'interfaccia che si presenta all'utente è la seguente:

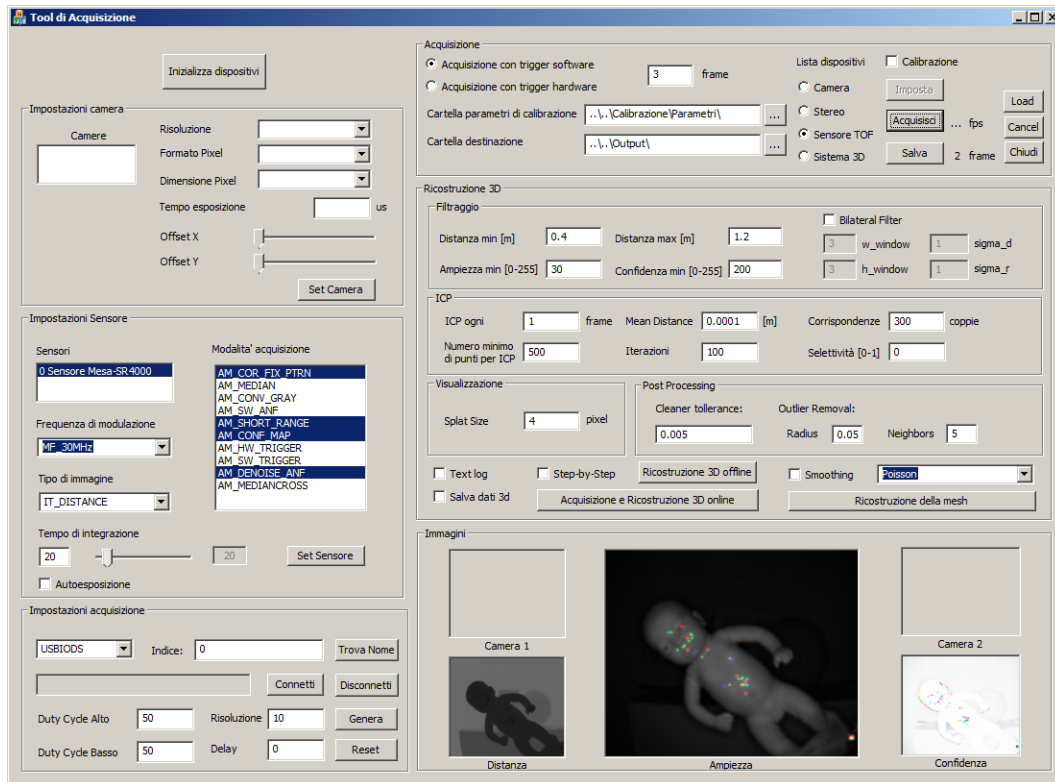


Figura 4.2: Interfaccia grafica di ToolAcquisizione

Per praticità è opportuno descrivere le funzionalità del software a partire da sotto sezioni dell'interfaccia grafica, illustrandone il ruolo all'interno del programma, soffermandoci in particolare sulla sezione 5 e 6, le più significative per il software:

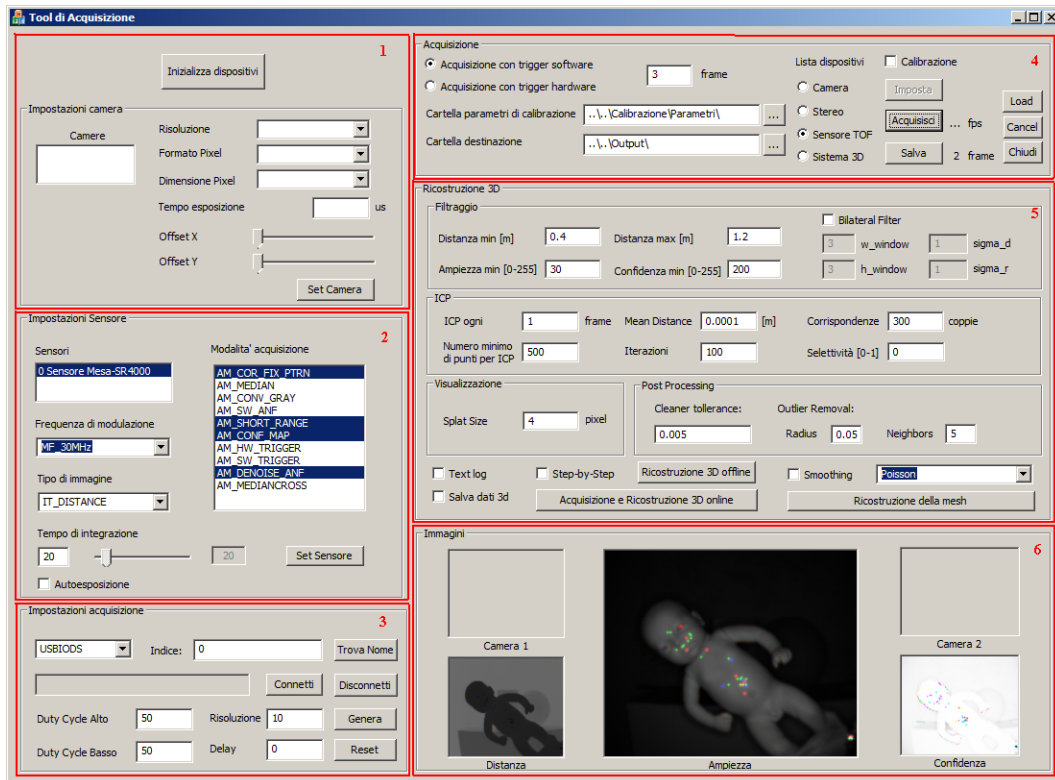


Figura 4.3: Sezioni dell'interfaccia grafica di ToolAcquisizione

Sezione 1

Il programma prevede la possibilità affiancare al sensore SR4000 videocamere 2D, realizzando così un sistema trinoculare [6, 7]. Questa funzionalità per ora non è stata impiegata ai fini di questo progetto.

Sezione 2

Questa sezione consente di inizializzare e configurare il sensore time-of-flight. E' possibile impostare la frequenza di modulazione del segnale inviato dal sensore, il tempo di integrazione e l'attivazione di alcune modalità di acquisizione che il sensore mette a disposizione. Tra questi citiamo il filtro anti rumore, che combina informazione sull'ampiezza e sulla distanza in un intorno 5×5 di un punto, e un semplice filtro mediano che usa una finestra di 3×3 punti. Tra i due è consigliato l'uso del primo filtro, in grado di compensare parzialmente il rumore sulle stime di profondità dei punti. Questi filtri sono implementati via hardware dalla camera stessa, non comportano dunque un costo computa-

zionale addizionale alla CPU del terminale.

Sempre in questa sezione è possibile attivare la generazione, da parte della camera, della mappa di confidenza, la cui utilità verrà spiegata nella sezione 6.

Un'ultima opzione riguarda il focus della camera. Attivando infatti l'opzione "AM_SHORT_RANGE" è possibile migliorare l'accuratezza dei dati per scansioni di oggetti collocati entro una distanza di un metro dal sensore. E' possibile inoltre variare la distanza del focus manualmente attraverso una vite posta sul retro della camera.

Sezione 3

Il sistema consente di regolare la temporizzazione delle acquisizioni mediante una scheda hardware che fornisce un clock esterno al sistema di trigger. In questa sezione è possibile configurare il duty cycle dell'onda di clock, dunque il rate di acquisizione desiderato.

Sezione 4

Nella sezione 4 l'utente immette il numero di frame che vuole acquisire, imposta la cartella che contiene i parametri intrinseci della camera, sceglie la cartella di destinazione dell'output del programma, ed invia l'effettivo comando di acquisizione dei dati. Questi possono essere salvati grazie al bottone "salva", che memorizzerà nella cartella di output specificata le immagini relative alle videocamere 2D (se presenti) e i dati provenienti dal sensore a tempo di volo (dati di profondità, nuvole di punti 3D, immagini di ampiezza e confidenza). La descrizione dei formati di output forniti dalla camera TOF è contenuta nella descrizione della sezione 6.

Come verrà meglio illustrato in seguito, è importante sottolineare che il comando di acquisizione qui presente non è accompagnato dalla ricostruzione in tempo reale dell'oggetto che si sta scansionando, perché questa funzione è delegata ad un apposito bottone presente nella sezione successiva. Il comando "acquisisci" si occupa dunque di coordinare la sola acquisizione dei dati provenienti dai dispositivi connessi, sui quali è possibile in seguito effettuare operazioni di ricostruzione 3D offline, non concorrenti dunque alla fase di acquisizione.

Sezione 5

Figura 4.4: Sezione 5 dell'interfaccia grafica di ToolAcquisizione

Nei parametri di pre-processing figurano la distanza minima e la distanza massima dei punti che si vogliono acquisire. Viene dunque definito un “volume utile” al di fuori del quale i dati semplicemente non vengono considerati e tutte le elaborazioni che seguono verranno effettuate in riferimento ai punti che hanno superato il filtraggio. Questo permette tempi di esecuzione più rapidi e migliore accuratezza nella fase di ricostruzione. Una seconda soglia di filtraggio è data dai valori minimi di ampiezza e di confidenza (valore che indica l'affidabilità della misura a partire dalle informazioni di ampiezza e distanza fornite dal sensore). È possibile infine attivare il filtro bilaterale (implementato via software) e variarne i parametri.

Per quanto riguarda l'algoritmo di allineamento, ICP, è possibile selezionare una varietà di parametri per meglio adeguare l'algoritmo alle particolarità della scansione che si vuole effettuare e per migliorarne le prestazioni, sia in termini di velocità che robustezza. L'ICP implementato dal programma può allineare frame successivi o frame “distanti” k frame l'uno dall'altro, con k scelto dall'utente. Poiché infatti frame successivi sono fortemente “overlapping”, l'allineamento di frame successivi può risultare ridondante per piccole rotazioni della camera (o dell'oggetto), viene dunque data all'utente la libertà di scelta sulla “frequenza” dell'allineamento. Impostando $k > 1$ i frame intermedi tra quelli da allineare verranno semplicemente scartati dal ciclo.

Il “*numero minimo di punti per ICP*” presente nella GUI fa riferimento al mini-

mo numero di punti che un frame deve possedere per essere considerato valido: se in fase di acquisizione l'oggetto fuoriesce dal campo visivo della camera o semplicemente si verifica un'eccessiva saturazione, il contenuto informativo del frame è considerato insufficiente per il processo di allineamento. Il programma gestisce questa situazione fermando il processo di registrazione delle viste e riattivandolo nel momento in cui un frame torna ad avere un numero sufficiente di punti per l'allineamento. E' importante inoltre che la posizione relativa del frame che permette il rientro del processo di allineamento sia quanto più possibile simile a quella dell'ultimo frame valido, catturato prima che il ciclo di allineamento si arrestasse. In caso contrario infatti la registrazione potrebbe risultare poco accurata e le successive registrazioni nel ciclo pair-wise sarebbero afflitte da un errore di allineamento consistente. L'utente può osservare la posizione dell'ultimo frame valido osservando le immagini nella sezione 6. Sempre dalla GUI l'utente può forzare la convergenza di un allineamento con ICP a un determinato numero di iterazioni, o più semplicemente forzare la convergenza ad ottenere un errore (espresso in metrica *Root Mean Square*) inferiore alla soglia impostata. E' possibile determinare inoltre il numero massimo di corrispondenze di punti per la stima della rototraslazione tramite SVD. La variazione di questi parametri influenza la robustezza e la velocità del processo di allineamento dei frame.

La “*selettività*” è una soglia che indica il valore minimo del coseno tra la normale di un punto e la media delle normali dei punti che hanno superato la fase di pre-processing. Punti con una normale eccessivamente divergente rispetto alla media non vengono considerati nella fase di allineamento e successiva visualizzazione, come spiegato nel capitolo “3D model acquisition pipeline del software”.

Per quanto riguarda la visualizzazione, con “*splat size*” si intende la grandezza con cui vengono rasterizzati i punti, come già spiegato in precedenza.

Per la fase di post-processing la “*cleaner tollerance*” è la distanza (espressa come percentuale della diagonale della bounding box del modello) entro la quale punti vicini vengono considerati come un unico punto e dunque filtrati. Invece “*radius*” e “*neighbors*” controllano i parametri dell'algoritmo di eliminazione degli outlier (un punto è outlier se, in un intorno sferico di raggio *radius*, ha un numero di vicini inferiore a *neighbors*).

Il bottone “*Text Log*” attiva l'opzione di generazione del log testuale, nel quale sono contenute numerose informazioni sull'esecuzione del programma, come la velocità (in ms) di alcune operazioni, o il numero di punti totale del modello finale. “*Salva dati 3D*” è una feature preziosa in fase di debug e analisi, in quanto permette di salvare, in un'unica vista tridimensionale, i risultati di un allineamento, con i modelli di *source*, di *target* e dell'applicazione della rototra-

slazione su *source* (*solution*) rappresentati con tre colorazioni differenti (*source* in rosso, *target* in verde e *solution* in blu). Dal momento che il salvataggio dei modelli tridimensionali coinvolge operazioni di scrittura su disco, questa feature è disponibile soltanto in caso di ricostruzione 3D offline, in cui scansione e ricostruzione non procedono parallelamente. Ad arricchire questa funzionalità si aggiunge la modalità “*Step-by-step*”, che offre la possibilità all’utente di visualizzare l’esecuzione di ICP iterazione per iterazione, con la possibilità di interagire via mouse e tastiera per ispezionare la qualità della mappatura di allineamento. Anche questa feature è disponibile, per ovvi motivi, solo in caso di ricostruzione 3D offline.

Due bottoni separati, “*Ricostruzione 3D offline*” e “*Acquisizione e ricostruzione 3d online*”, sottolineano la differenza tra le due modalità di impiego del programma: la modalità offline effettua la ricostruzione 3D a partire dai dati forniti da una acquisizione già effettuata, nella modalità online invece acquisizione e ricostruzione procedono in pipeline, e il processo di elaborazione ha inizio non appena i dati vengono resi disponibili dalla camera.

Il modello finale (nuvola di punti 3D) viene automaticamente salvato su disco nella cartella di output.

E’ prevista infine la possibilità di ricostruire la mesh poligonale a partire da una nuvola di punti 3D attraverso i tre algoritmi di ricostruzione delle superfici citati nel precedente capitolo. Selezionando l’opzione “*Smoothing*”, verrà effettuato uno smoothing laplaciano della mesh poligonale risultante.

Sezione 6



Figura 4.5: Sezione 6 dell’interfaccia grafica di ToolAcquisizione

Ci sono tre diversi tipi di immagini che possono essere visualizzate dal software e che vengono prodotte dall’ SR4000:

1. Distanza

Per ogni pixel questa immagine contiene il valore della distanza dell'oggetto identificato dal pixel.

2. Ampiezza

Questa immagine fornisce, in scala di grigio, la luminanza dei vari oggetti presenti nella scena ed è proporzionale all'ampiezza del segnale in ricezione.

3. Confidenza

SR4000 utilizza una combinazione della misura della distanza e dell'ampiezza, e le loro variazioni, per calcolare una misura di probabilità che la distanza misurata sia effettivamente la distanza corretta. Questa misura è per l'appunto la mappa di confidenza (confidence map), e può essere utilizzata per selezionare regioni contenenti misure ad alta qualità e scartare quelle povere di qualità.

Queste tre immagini compaiono nella GUI, assieme alle immagini visualizzate da eventuali due videocamere 2D, nel caso che il sensore TOF costituisca un sistema trinoculare con le due videocamere.

Durante la fase di ricostruzione 3D, un'ulteriore finestra verrà aperta per la visualizzazione incrementale del modello 3D dell'oggetto che si sta scansionando (o che è già stato scansionato, nel caso di ricostruzione offline). Al termine della ricostruzione, è possibile interagire con il modello 3D mediante mouse e tastiera, per ruotare il modello, osservandolo da punti di vista arbitrari.

Formato del log di testo

Il log di testo contiene informazioni preziose per l'analisi e il debug del programma ed è costituito da tre parti principali: la prima elenca i parametri di ricostruzione impostati dall'utente all'avvio del software, la seconda stampa le informazioni inerenti a ciascuna iterazione di ricostruzione 3d e la terza riassume alcune caratteristiche del modello 3d finale e calcola la media del tempo di esecuzione di alcune operazioni.

Il log è dunque strutturato così:

Prima parte:

1. Parametri iniziali

- (a) Numero di frame acquisiti
- (b) Filtraggio
 - i. Soglia di distanza minima
 - ii. Soglia di distanza massima
 - iii. Soglia di confidenza minima
 - iv. Soglia di ampiezza minima
 - v. Bilateral filter (se presente)
 - A. Dimensione orizzontale finestra
 - B. Dimensione verticale finestra
 - C. σ_d
 - D. σ_r
- (c) ICP
 - i. Passo di frame di icp
 - ii. Numero iterazioni massimo icp
 - iii. Numero massimo di corrispondenze
 - iv. Minimo numero di punti per eseguire ICP
 - v. Maximum mean distance
 - vi. Selettività
- (d) Visualizzazione
 - i. Splat size

Seconda parte:

1. Indice frame iterazione
2. Millisecondi impiegati per il calcolo delle normali
3. Numero punti target
4. Numero punti source
5. Bounding box diagonal media
6. Millisecondi impiegati per icp
7. Mean Distance (RMS)

8. Matrice di rototraslazione ICP
9. Millisecondi impiegati per la visualizzazione
10. Millisecondi impiegati in totale per l'iterazione
11. Stima di frame al secondo per l'iterazione

Terza parte:

1. Mean Distance media
2. Tempo di allineamento medio ICP
3. Tempo medio per l'iterazione
4. Stima del numero medio di frame al secondo
5. Numero punti del modello finale
6. Numero di punti del modello finale ripulito

Capitolo 5

Analisi dei risultati

5.1 Configurazione di prova

La configurazione di prova per i test è data da un pc Dell XPS420 con le seguenti caratteristiche tecniche:

- Processore: Intel Core 2 Quad CPU Q6600 @ 2.40 GHz
- Memoria RAM: 4 GB DDR2
- Scheda video: ATI Radeon 3870 512 MB memoria integrata
- Sistema operativo: Windows 7 x64 Enterprise

Come è possibile notare, si tratta di una configurazione appartenente al mercato consumer di circa 3 anni fa, inferiore quindi agli standard odierni, specie per quanto riguarda il processore in dotazione.

5.2 Rumore sui dati di profondità

Prima di affrontare l'analisi dei risultati del processo di ricostruzione 3d, è utile osservare la qualità delle range map di un singolo frame per meglio comprendere l'entità del rumore sulle stime di profondità.

L'oggetto scelto per l'analisi è un bambino giocattolo alto circa trenta centimetri: un oggetto di piccole dimensioni, con superficie opaca (plastica non riflettente) e con una geometria relativamente semplice ma non priva di dettagli di interesse per l'analisi, come i lineamenti del viso e l'ombelico.

I dati presentano ottima precisione per quanto riguarda le coordinate x-y dell'immagine ma è rilevabile una forte componente di rumore lungo l'asse z, in prossimità

di superfici più articolate come il viso. Il rumore della camera TOF rappresenta l'ostacolo principale per un allineamento e una ricostruzione 3D di buona qualità.

Un altro aspetto da tenere in considerazione è la difficoltà, per un sensore TOF, di stimare correttamente le profondità dei bordi dell'oggetto, ossia quando la superficie dell'oggetto presenta un certo angolo di inclinazione con la direzione del raggio IR incidente. Questo infatti viene riflesso in modo proporzionale al coefficiente di riflettanza della superficie. Il raggio viene dunque deviato e la porzione riflessa giunge al sensore con una potenza insufficiente per una corretta stima della profondità. Per limitare parzialmente questo problema, come descritto nel capitolo "3D Model Acquisition pipeline del software", è stato applicato un filtro che elimina i punti la cui normale diverge eccessivamente dalla media delle normali del frame.

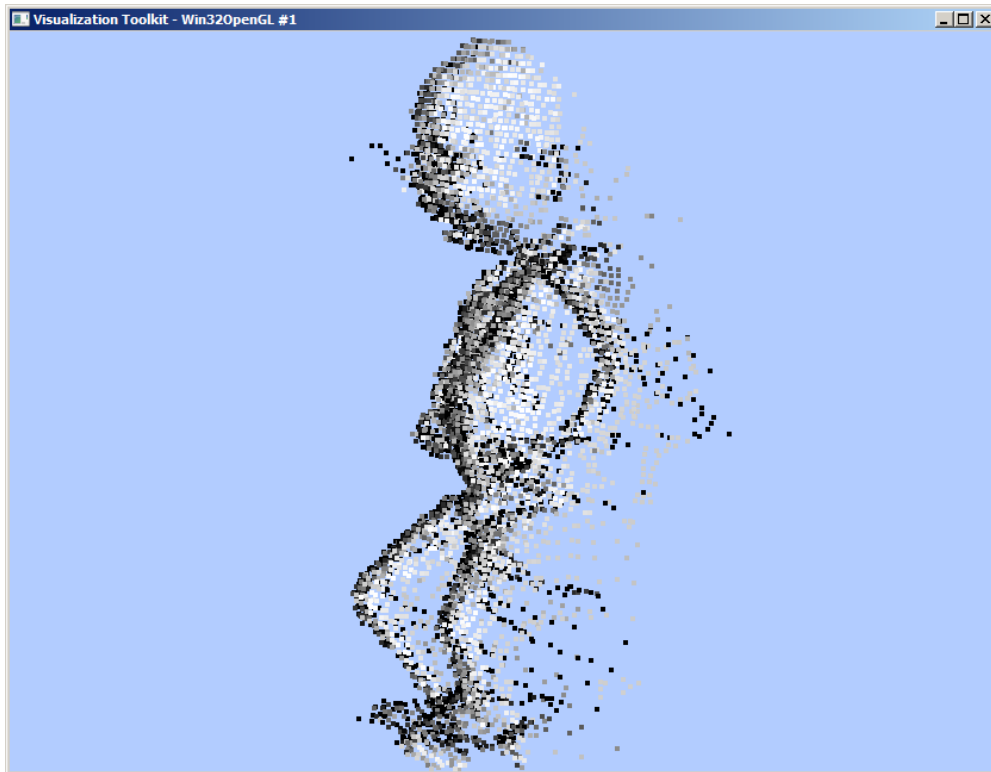


Figura 5.1: Modello 3d senza l'applicazione di filtri (tempo di integrazione: 20 ms)

Aumentando il tempo di integrazione da 20 ms a 40 ms le stime di profondità migliorano lievemente, tuttavia è necessario mantenere l'oggetto ad una distanza dal sensore superiore, per evitare il rischio di saturazione. Ciò si traduce in una generale perdita di contenuto informativo sui dettagli della superficie, non più catturata da vicino ma da una distanza maggiore. Inoltre un tempo di integrazione elevato espone il sistema a problemi di motion blur, dunque è necessario ruotare gli oggetti con una certa lentezza.

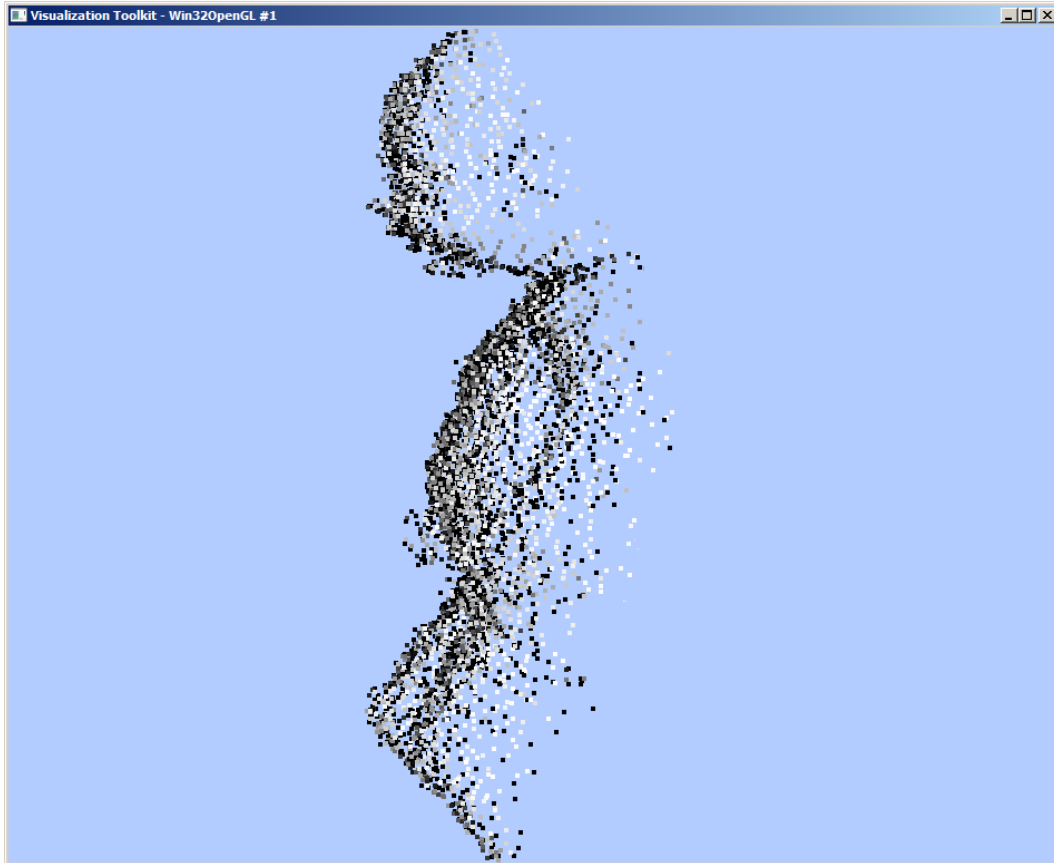


Figura 5.2: Modello 3d con l'applicazione del filtro AM_SHORT_RANGE e AM_DENOISE_ANF (tempo di integrazione: 40 ms)

La seguente figura mostra l'applicazione del bilateral filter con una finestra 5×5 pixel e σ_d e σ_r pari a 5. Come è possibile osservare, il rumore sull'asse z viene drasticamente ridotto e gli edge vengono preservati, anche se alcuni dettagli, non strettamente classificabili come edge, scompaiono per effetto dello smooth prodotto dal filtro. I particolari come occhi, ombelico e bocca vengono invece conservati.

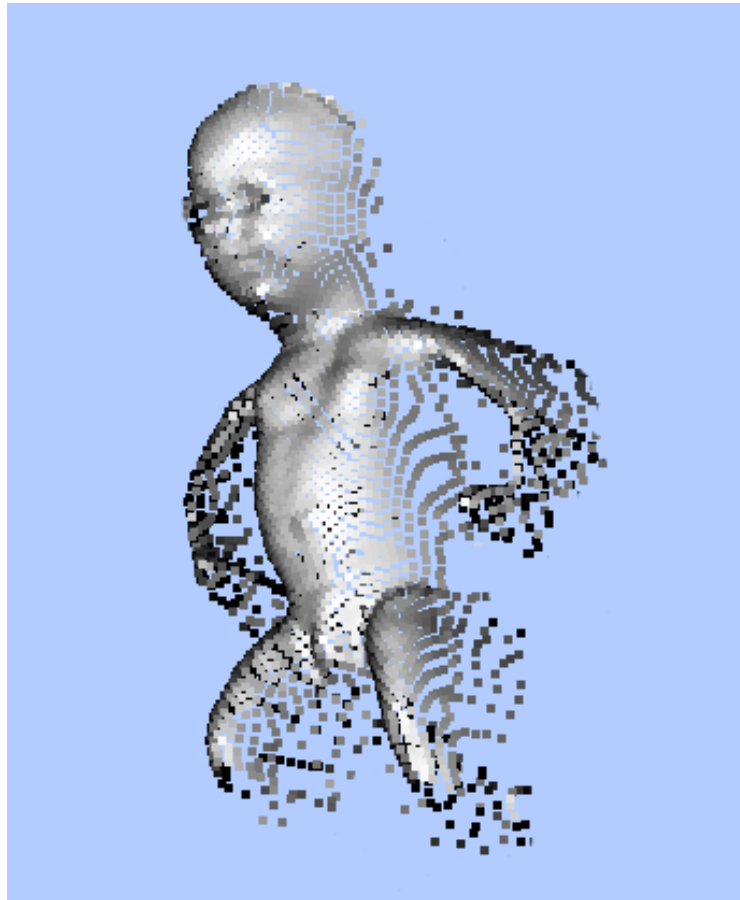


Figura 5.3: Modello 3d dopo l'applicazione del bilateral filter (finestra 5×5 , $\sigma_d = 5$, $\sigma_r = 5$). Poichè l'oggetto è distante dallo sfondo, il filtro bilaterale tende a distribuire i punti dei bordi verso lo sfondo.

5.3 Limiti di applicabilità

In questi test vengono presi in considerazione due oggetti critici per il sistema:

- una pentola, la cui superficie speculare tende alla saturazione del sensore;
- un pallone, ossia un oggetto regolare, geometricamente privo di feature.

Il primo caso mette in evidenza le insufficienze o i limiti di applicabilità della tecnologia utilizzata, mentre il secondo oggetto sottolinea un limite dell'algoritmo di allineamento adottato, che fallisce per classi di oggetti particolari, privi di feature geometriche.

Per quanto riguarda la pentola, una superficie eccessivamente riflettente porta con facilità alla saturazione del sensore, specie se il tempo di integrazione configurato è alto. In caso di saturazione, i pixel corrispondenti vengono filtrati dal software e di conseguenza il frame presenta scarso contenuto informativo sulla geometria dell'oggetto (in caso di numero di punti totale eccessivamente basso, il programma scarta automaticamente il frame, come già descritto nel capitolo precedente). Nell'immagine 5.4 infatti l'unica superficie ricostruita è quella della mano che afferra la pentola. Un altro problema, legato a questo tipo di superfici, si verifica quando la riflessione devia il raggio incidente lontano dal sensore, impedendo a quest'ultimo di catturare un segnale di ritorno sufficientemente potente da stabilire correttamente le misurazioni di profondità. Questo aspetto è particolarmente evidente nella mappa di confidenza in figura 5.4: ai lati della pentola i pixel presentano gradazione scura, indicando dunque valori di confidenza estremamente bassi.

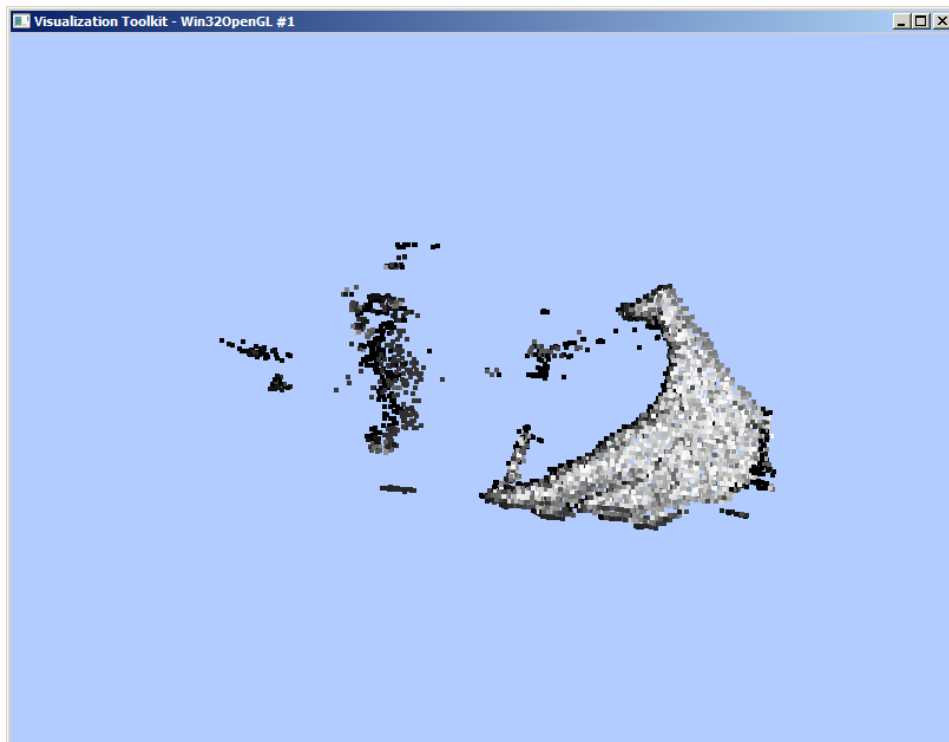


Figura 5.4: Frame 3D di una pentola

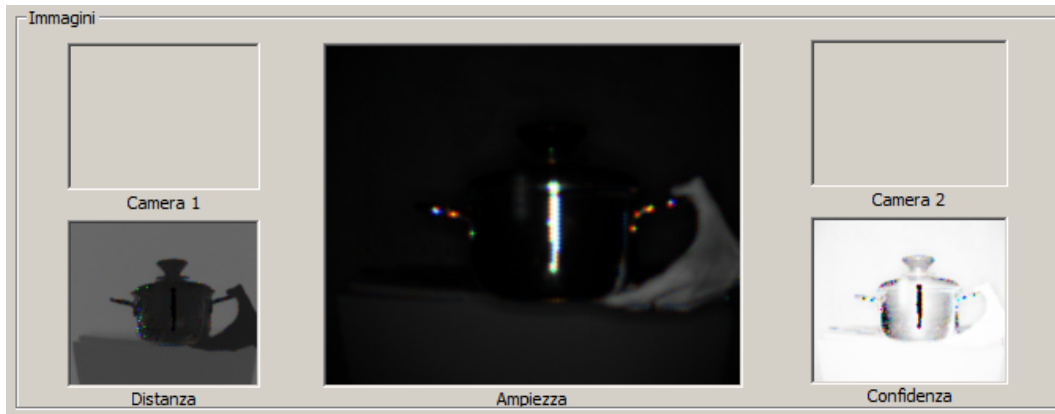


Figura 5.5: Immagini 2D di una pentola

L'algoritmo di allineamento ICP fallisce in presenza di oggetti regolari con superfici privi di feature geometriche. Si consideri ad esempio il caso di un pallone, assimilabile idealmente ad una sfera. Se ruotiamo il pallone attorno al suo asse verticale, frame successivi presenteranno al sensore la medesima forma geometrica e la trasformazione di allineamento fornita dall'algoritmo sarà un'identità. Anziché una sfera completa quindi, il modello finale sarà solo una porzione di sfera, come raffigurato nell'immagine 5.6, ottenuta compiendo una rotazione di circa 120 gradi e osservando dall'alto la ricostruzione risultante. Dall'immagine è anche possibile osservare la consistente componente di rumore che degrada l'accuratezza della ricostruzione tridimensionale a tutti i livelli della pipeline, dal processo di allineamento (che risulta imperfetto a causa del rumore geometrico) a quello di visualizzazione. Come verrà accennato nel capitolo conclusivo, una possibile idea per affrontare questo problema è affiancare al sensore TOF una videocamera 2d in grado di offrire informazione di colore sulla superficie. L'algoritmo di allineamento dovrà prevedere una metrica di distanza tra le corrispondenze di punti che tenga conto sia della loro distanza euclidea, sia della loro "distanza" di colore (intesa come differenza scalare dei valori RGB).

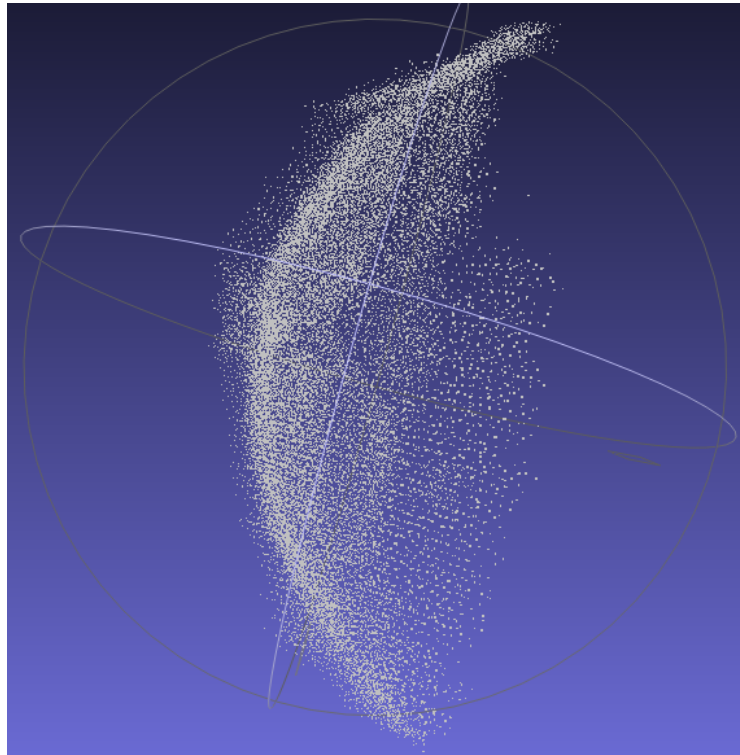


Figura 5.6: Vista dall'alto della ricostruzione 3d del pallone

5.4 Ricostruzione 3D di un oggetto

In questo paragrafo vengono mostrati i risultati di un processo di ricostruzione 3d. L'oggetto scelto è il bambino descritto nel primo paragrafo. L'acquisizione è stata eseguita ruotando lentamente a mano l'oggetto davanti alla telecamera, fissata in una postazione a treppiede, ad una distanza di circa 50 cm dall'oggetto, compiendo un angolo totale di circa 120 gradi attorno all'asse verticale dell'oggetto. Rotazioni superiori ai 100 gradi tendono a produrre risultati insoddisfacenti, in quanto la scarsa ripetibilità delle misure e l'errore incrementale dei processi di allineamento pair-wise tendono a degradare la qualità della ricostruzione. Per questo test sono stati acquisiti 100 frame e il processo di allineamento è stato impostato ogni 2 frame.

Le immagini 5.7,5.8,5.9 illustrano un allineamento tra due successivi frame: la figura in rosso rappresenta la nuvola di punti (*source*) che deve essere allineata alla nuvola di punti verde (*target*), mentre la figura in blu (*solution*) rappresenta l'applicazione della rototraslazione che allinea *source* su *target*. La qualità della trasformazione è data dunque dallo scostamento tra *source* e *target* (lo scarto quadratico medio viene indicato nel log del programma).

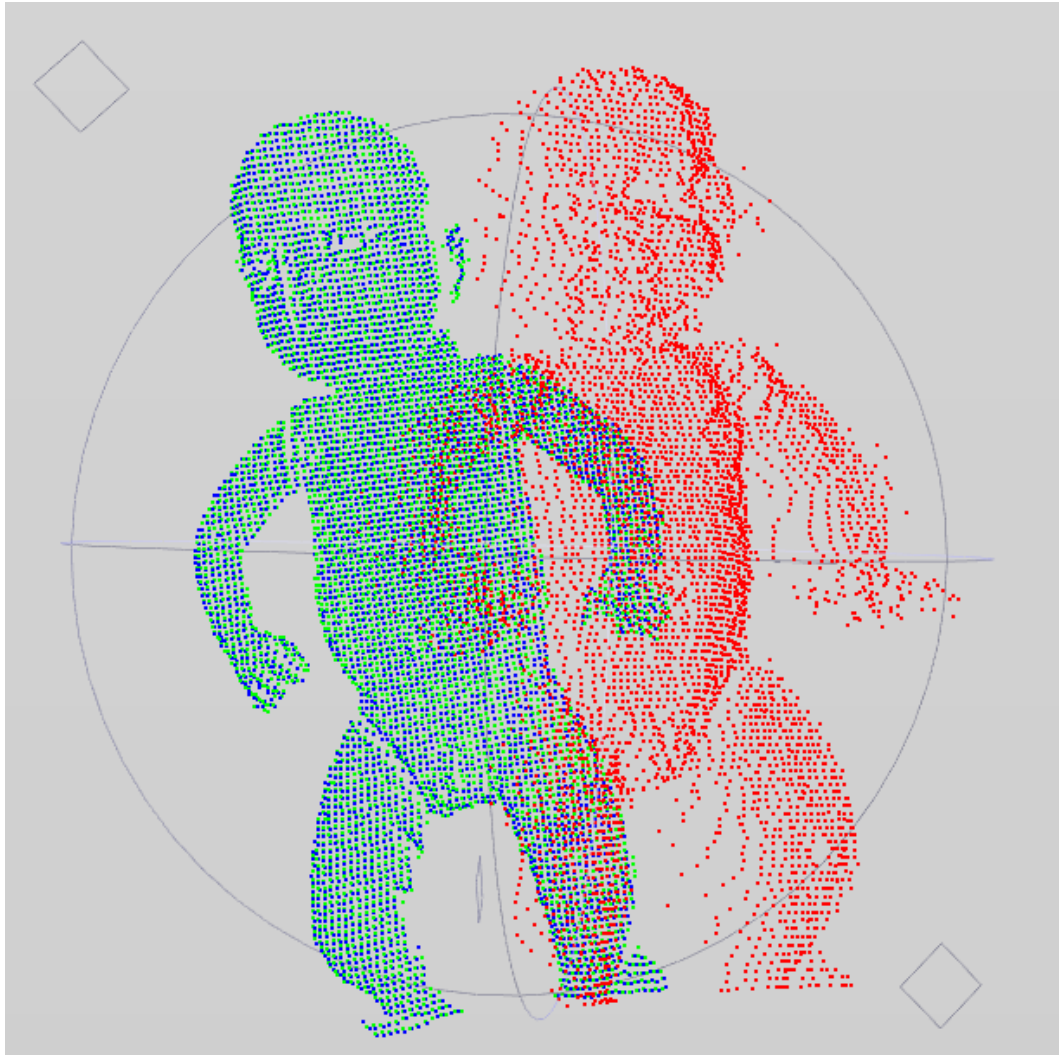


Figura 5.7: Allineamento tra 2 frame tramite ICP (vista frontale)

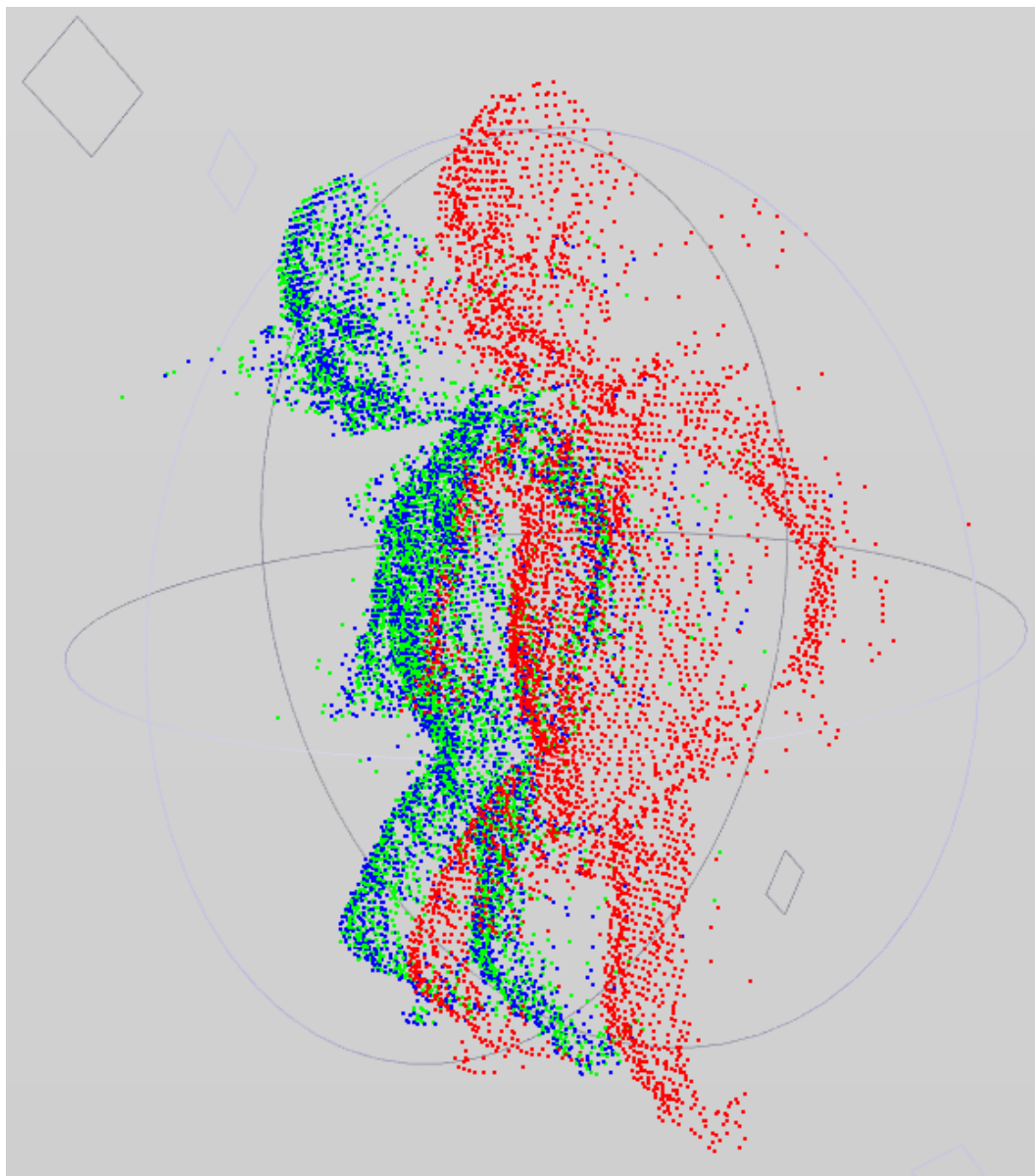


Figura 5.8: Allineamento tra 2 frame tramite ICP (vista laterale)

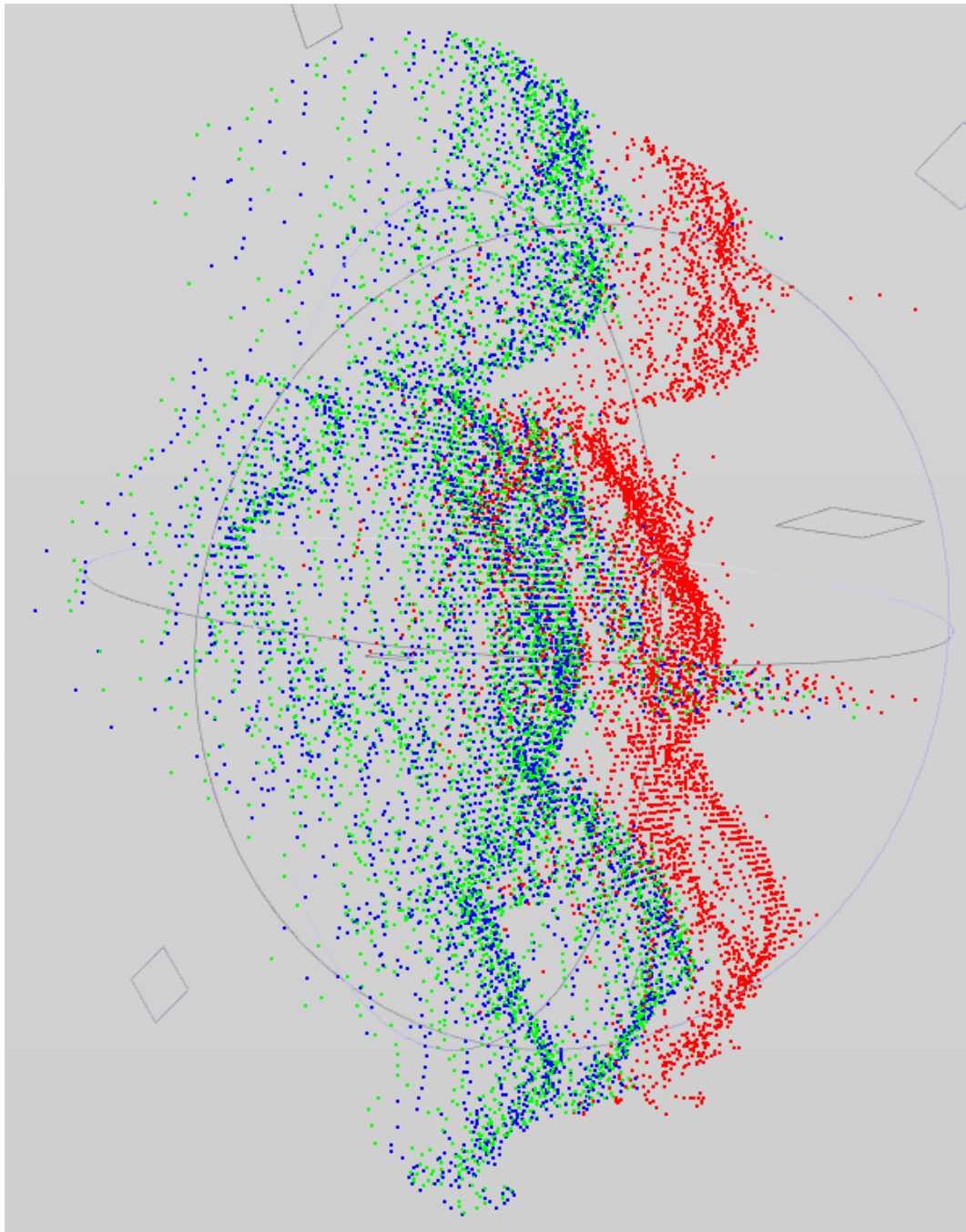


Figura 5.9: Allineamento tra 2 frame tramite ICP (vista laterale)

Il modello 3D finale è il seguente:

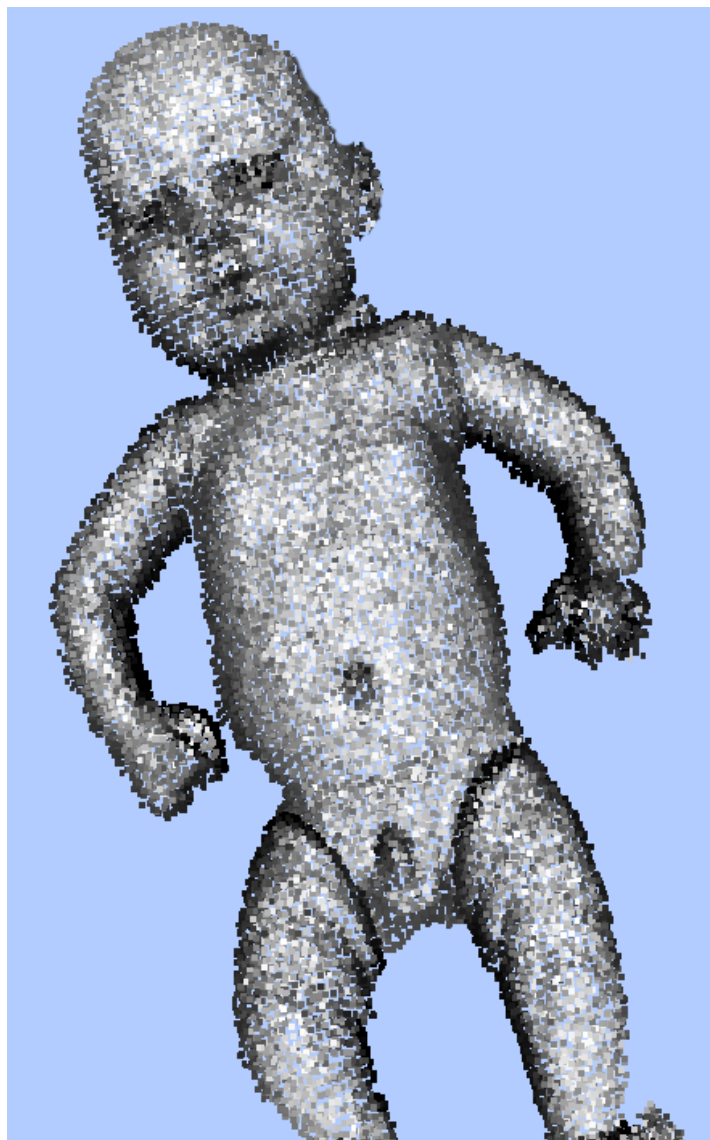


Figura 5.10: Modello 3d finale (nuvola di punti)

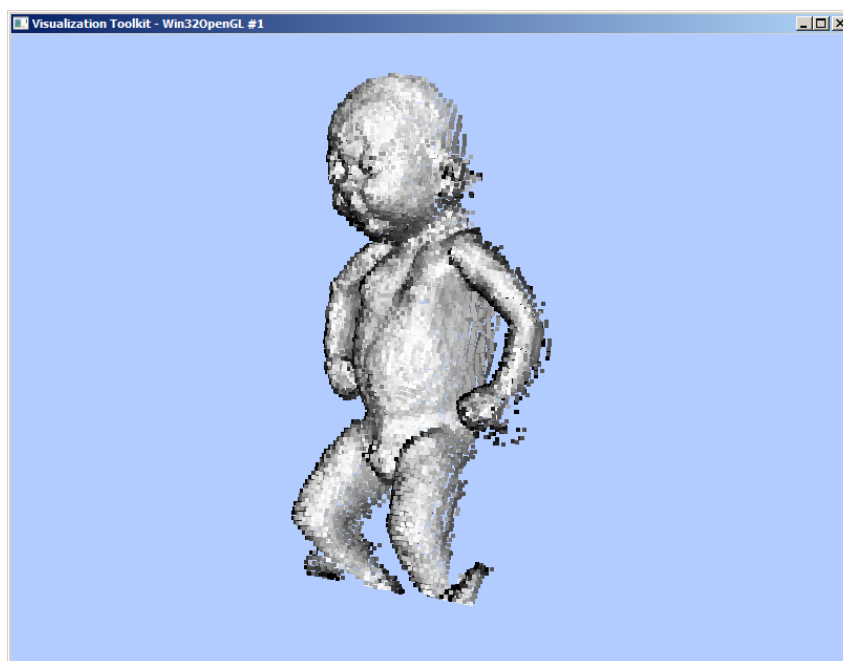


Figura 5.11: Modello 3d finale

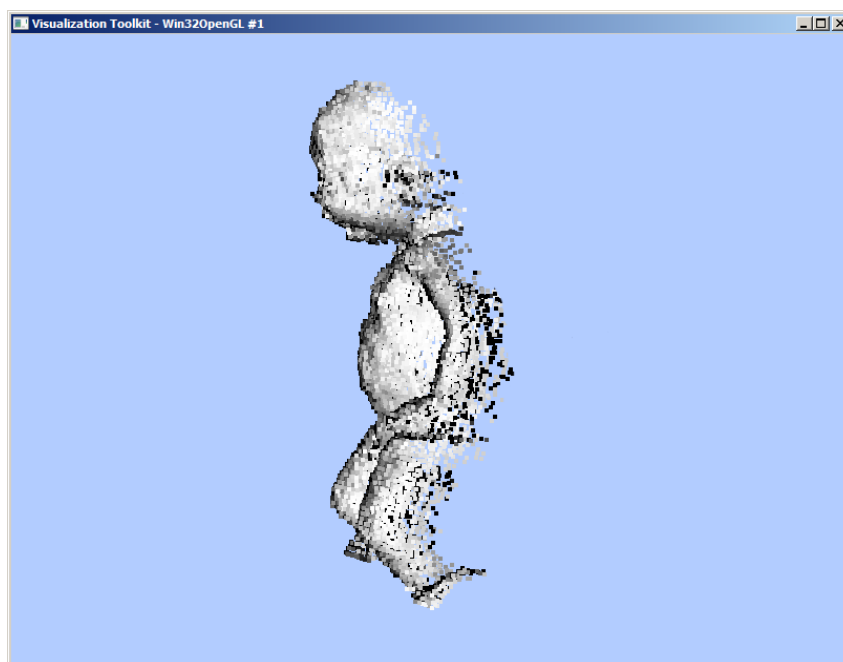
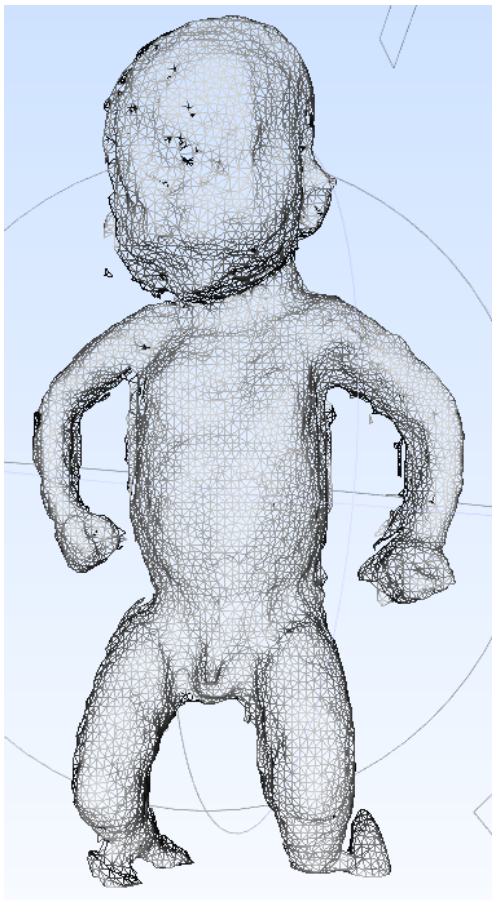
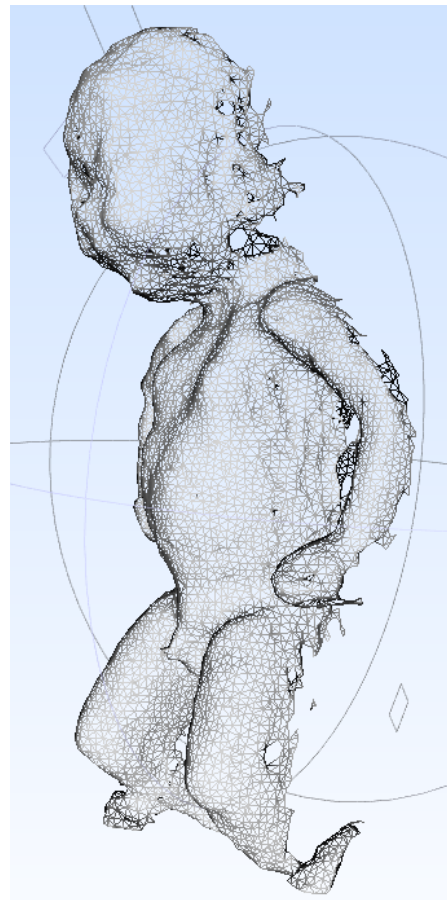


Figura 5.12: Modello 3d finale

Nell'immagine 5.13 viene raffigurato il modello dopo la fase di post processing e dopo la creazione della mesh, utilizzando il software "MeshLab" [34] e l'algoritmo di ricostruzione *Ball Pivoting* [35]. Tra gli algoritmi di ricostruzione della superficie a partire da una generica nuvola di punti, *Ball Pivoting* si è rivelato il più robusto per i modelli finali restituiti dal software. L'algoritmo tuttavia opera un'ulteriore semplificazione del modello, sacrificando parte del dettaglio visivo e introducendo alcuni artefatti nella superficie risultante.



(a) Mesh 3d (fronte)



(b) Mesh 3d (lato)

Figura 5.13: Mesh 3d

Nella figura 5.14 la mesh viene messa a confronto con il modello 3d del bambino dopo la ricostruzione 3D dello scanner laser “NextEngine 3D Scanner HD” [36].

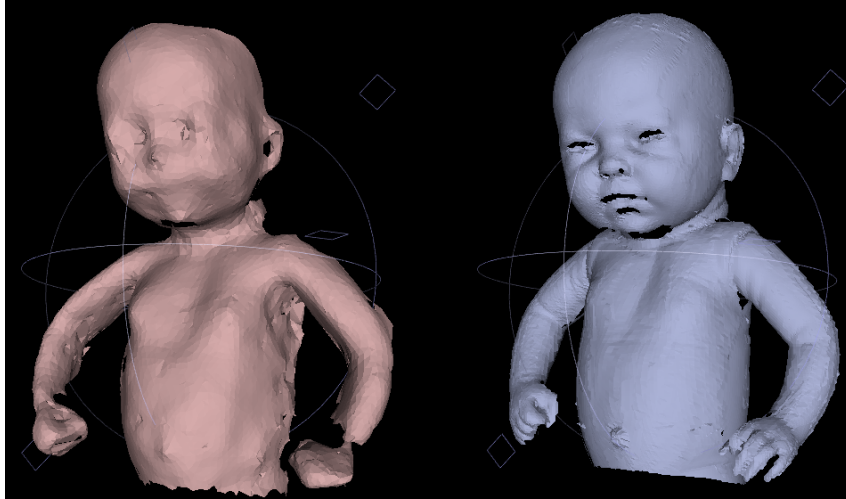


Figura 5.14: Mesh 3d TOF (sinistra), mesh 3d laser scanner (destra)

L'immagine 5.15 mostra qualitativamente l'entità dell'errore di ricostruzione dell'oggetto: è stato effettuato un campionamento dei vertici del modello TOF ed è stata calcolata la distanza di Hausdorff prendendo come riferimento il modello 3d fornito dallo scanner laser. Poichè i due sistemi di acquisizione hanno due sistemi di riferimento differenti, prima di eseguire il confronto si è reso necessario allineare le mesh inizialmente a mano e poi con ICP. L'errore di ricostruzione include dunque l'errore di allineamento delle mesh e l'errore di ricostruzione della superficie a partire dalla nuvola di punti.

L'errore minimo viene evidenziato in rosso, l'errore massimo corrisponde invece al colore blu. La media dell'errore è 4,476 millimetri, l'RMS è 5,540 millimetri, in linea dunque con le specifiche di errore dei sensori TOF. Come è possibile osservare, l'errore si concentra sui dettagli del viso e su alcuni bordi della braccia.

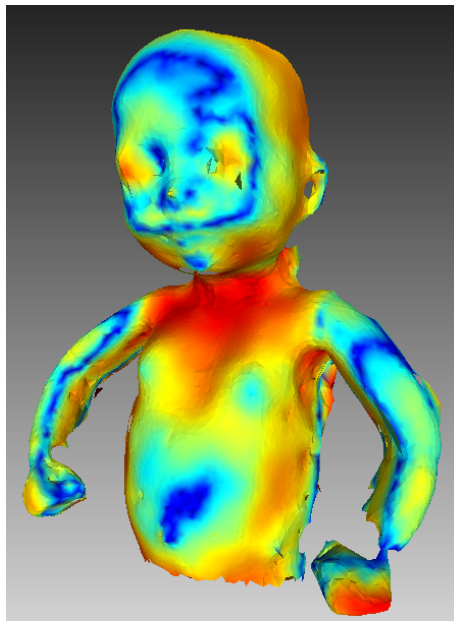


Figura 5.15: Quality map della mesh finale

Nel seguente grafico vengono riportate le prestazioni del programma, in termini di velocità di esecuzione.

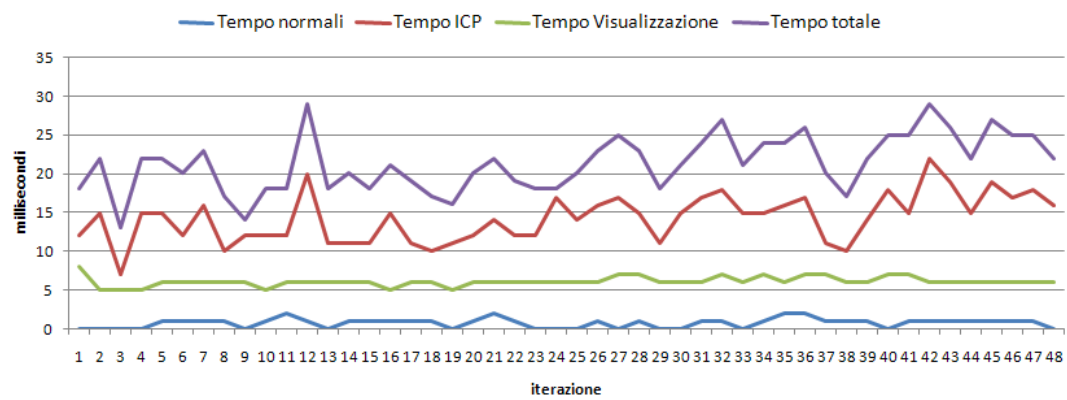


Figura 5.16: Prestazioni temporali del software

Osservazioni

I risultati mettono in evidenza l'ostacolo principale per la ricostruzione 3D di un oggetto: il rumore sulle misure di profondità delle immagini. La procedura di registrazione allinea correttamente i frame e in tempi di esecuzione brevi, ma la scarsa ripetibilità delle misure e il rumore delle range map fornite dalla camera TOF limitano la ricostruzione di un modello finale altamente preciso e rappresentativo dell'oggetto reale, almeno per quanto riguarda l'acquisizione di oggetti di dimensioni inferiori ai 50 cm. Gli algoritmi di filtraggio e di compensazione del rumore permettono di limitare in parte questo problema, la cui origine è da attribuirsi al principio di funzionamento della tecnologia stessa, anche se non si esclude che i prossimi sensori TOF di Microsoft e Mesa possano in futuro apportare significativi miglioramenti ai risultati attuali.

5.5 Ricostruzione 3D di una parete

Sebbene i sensori TOF non siano tra le tecnologie più orientate per la ricostruzione di stanze, per via dei già citati fenomeni di *scattering* e per l'eventuale presenza di oggetti con coefficienti di riflettanza diversi, è interessante indagare le potenzialità delle videocamere time-of-flight per questo tipo di applicazione, che potrebbe avere notevoli sviluppi in campi quali la navigazione autonoma (robotica) o la digitalizzazione di ambienti (edilizia).

Diversamente dalla costruzione 3d di oggetti, da cui ha avuto origine la tesi, l'incidenza del rumore sui dati di profondità è notevolmente più bassa e non rappresenta dunque un ostacolo fondamentale per una ricostruzione 3d di qualità. I limiti principali si sono rivelati invece i seguenti:

- Eventuale mancanza di feature geometriche in una parete

In assenza di informazione di colore, il processo di allineamento può fallire, similmente a quanto dimostrato con l'oggetto di test pallone nel paragrafo precedente.

- Scarsa risoluzione X-Y

La risoluzione del sensore SR4000, 176×144 , limita la quantità di informazione catturata ad ogni frame, di conseguenza alcuni dettagli visivi sulla scena vengono inevitabilmente persi.

- Numero di punti da elaborare per ogni iterazione

Nel test precedente l'oggetto veniva segmentato dallo sfondo e filtrato ed era composto in media da circa 7000 punti, un frame di una parete consiste invece di quasi 25000 punti. Istanze in ingresso di taglia maggiore impongono un aumento di alcuni parametri dell'algoritmo di allineamento per una ricostruzione 3d più robusta. In particolare il numero di corrispondenze per il calcolo della rototraslazione con SVD passa da 500 corrispondenze a 8000 corrispondenze. Le prestazioni del programma dunque, in termini di velocità di esecuzione, degradano, passando da circa 20 ms in media per un'iterazione a circa 700 ms. Nonostante ciò, si può affermare che le prestazioni del software rientrino comunque nelle specifiche soft real-time del progetto.

Per testare le capacità di ricostruzione degli ambienti, è stata effettuata una scansione di una porzione di uno studio, lunga circa 2,5 metri, da un angolo della parete all'angolo adiacente.

È possibile acquisire una parte con un singolo *snapshot*, tuttavia il ridotto angolo di visione dello SwissRanger4000 (circa 30 gradi) costringe l'operatore ad acquisire la scena da una certa distanza. In questa situazione la bassa risoluzione dello strumento ciò non permette di catturare sufficienti dettagli visivi della parete.

Per quanto riguarda la qualità dei dati, l'applicazione del bilateral filter permette di ridurre notevolmente il rumore sull'asse z e allo stesso tempo preservare gli *edge*, incrementando di molto la qualità dei dati e la loro visualizzazione. Nell'immagine 5.17 è raffigurata la ricostruzione dell'angolo di una parete così come viene visualizzata dal software, dopo l'applicazione di un filtro bilaterale con una finestra 5×5 , $\sigma_d = 5$ e $\sigma_r = 10$. E' possibile notare gli *edge* della porta e una scatola in basso a destra.

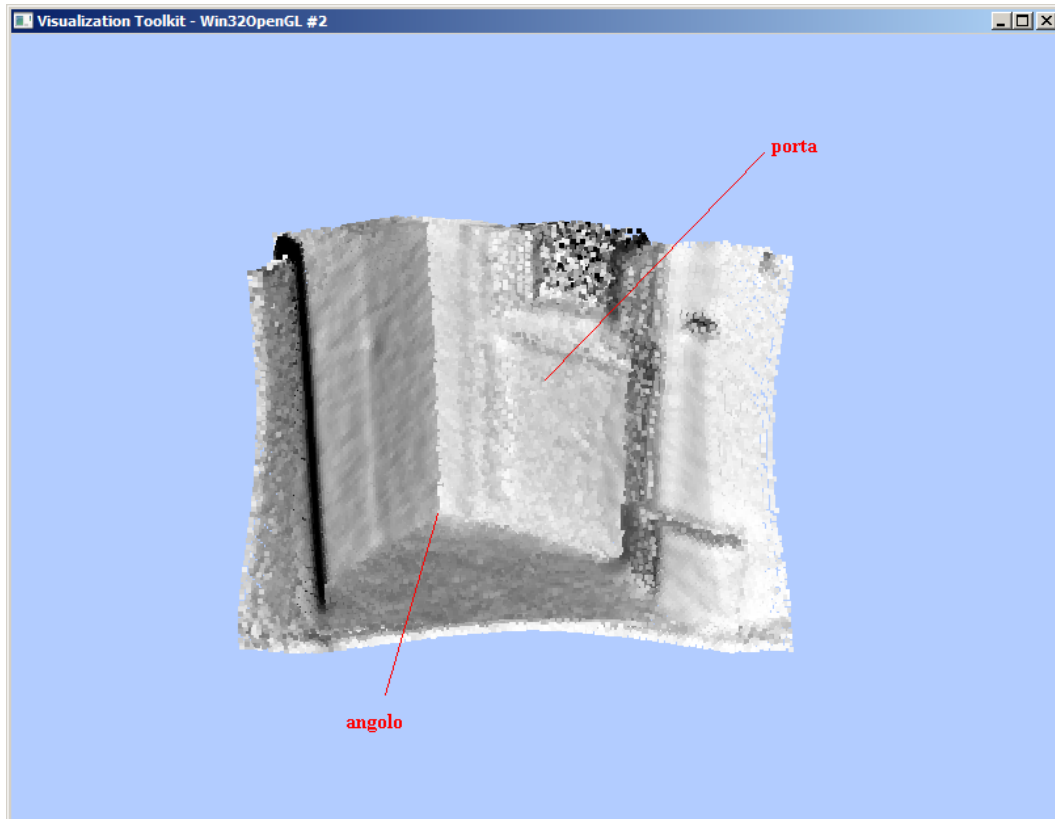


Figura 5.17: Ricostruzione dell'angolo della parete con bilateral filter

Dopo aver forzato i parametri di ICP come già spiegato in precedenza, è possibile osservare dalle seguenti immagini la qualità dell'allineamento tra frame successivi della parete (i due frame vengono allineati sovrapponendosi nella loro porzione comune):

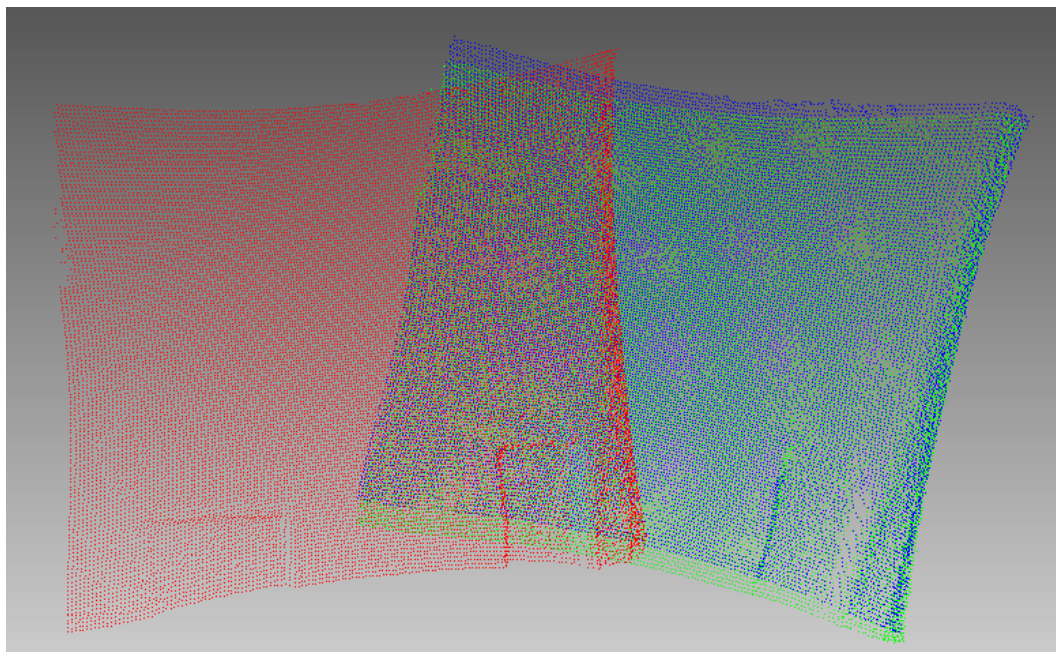


Figura 5.18: Allineamento (vista frontale)

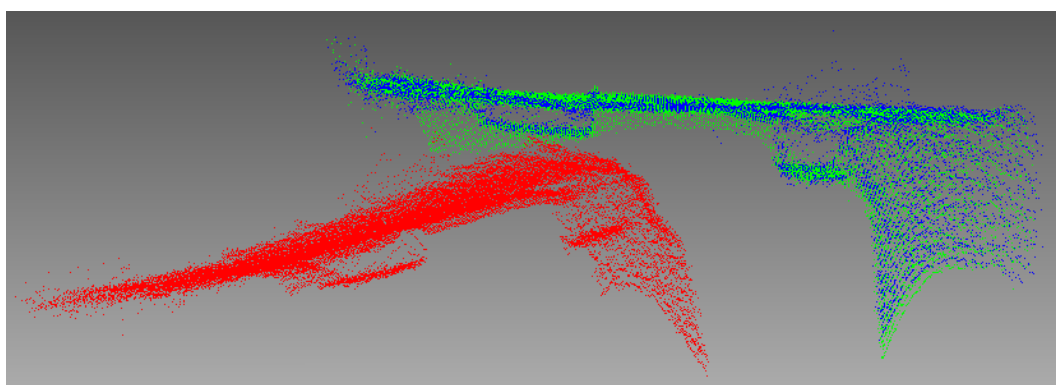


Figura 5.19: Allineamento (vista piana)

Qui in seguito sono riportate alcune foto della parete e alcuni *screenshot* del programma in esecuzione, durante il processo di ricostruzione. Nella scena sono stati inseriti alcuni elementi come un cestino, accanto alla porta, e una borsa a tracolla, appesa alla parete. L'acquisizione segue l'ordine delle foto 2D della parete:



Figura 5.20: Fotografia della stanza (angolo a sinistra)

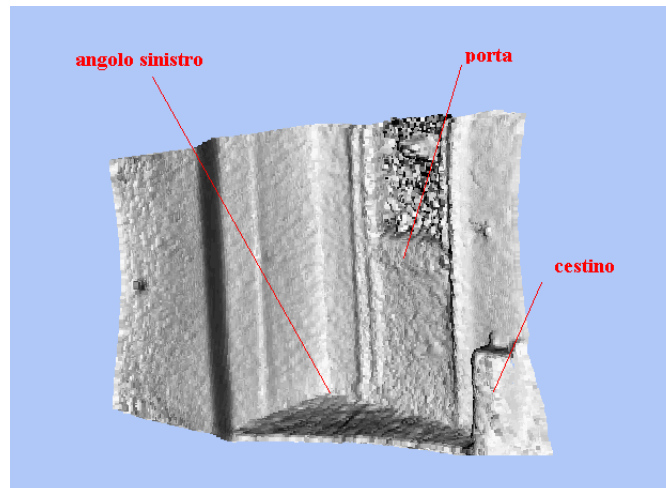


Figura 5.21: Ricostruzione 3d (angolo a sinistra)



Figura 5.22: Fotografia della stanza (fronte)

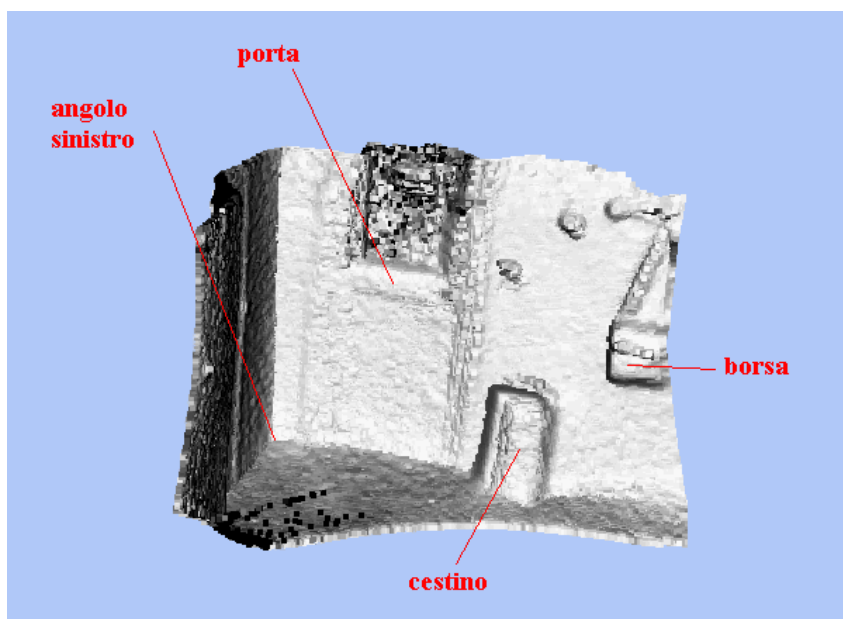


Figura 5.23: Ricostruzione 3d (fronte)



Figura 5.24: Fotografia della stanza (angolo a destra)

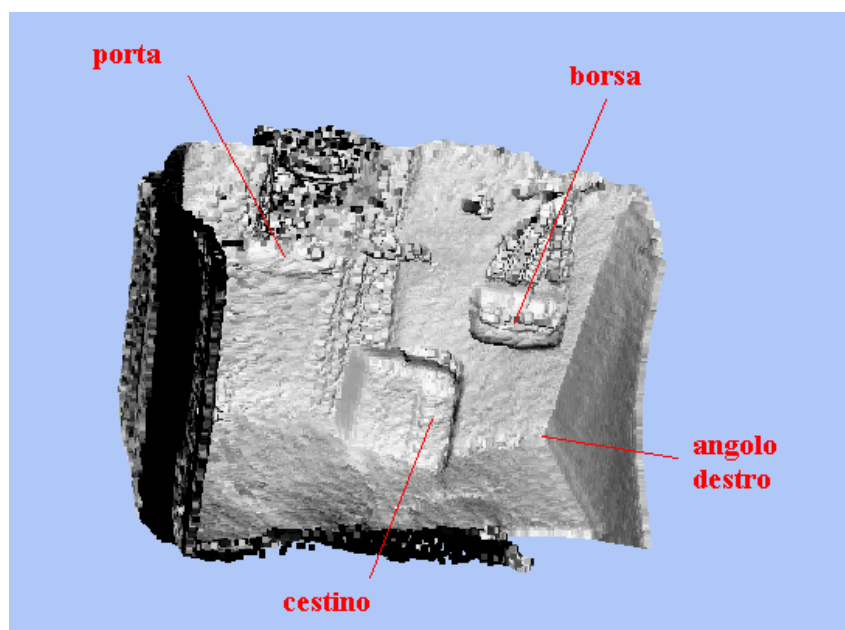


Figura 5.25: Ricostruzione 3d (angolo a destra)

Come è possibile osservare, l'errore incrementale del processo di registrazione pair-wise tende a manifestarsi in modo consistente dopo circa 100 frame (figura 5.25), disallineando progressivamente alcuni dettagli visivi (in particolare il cestino in basso) lungo il processo di ricostruzione.

Si è inoltre constatato che l'allineamento risulta impreciso quando compaiono nella scena oggetti non presenti nel frame precedente: nel calcolo delle corrispondenze dei punti in ICP, i punti appartenenti all'oggetto introdotto nella scena vengono associati a punti che tuttavia non rappresentano lo stesso oggetto, poiché questo non è ancora presente nella scena. Di conseguenza alcune delle coppie di punti calcolate da ICP risultano spurie e minano la robustezza della stima della rototraslazione.

5.6 Ricostruzione 3D di una mano

Uno degli interessi principali della computer vision è il posizionamento e il tracking della mano nello spazio 3d. Un'applicazioni di questi studi è l'elaborazione di interfacce di interazione uomo-macchina che facciano uso direttamente dei movimenti della mano (o delle dita) senza l'intervento di controller fisici come mouse e tastiera. Da questo punto di vista la mano rappresenta un oggetto particolare e articolato per la ricostruzione 3d ed è utile esplorare le potenzialità dei sensori TOF in questo senso.

Il test è stato effettuato ruotando la mano di circa 100 gradi, passando da una posizione in cui il palmo è frontale rispetto al sensore ad una in cui si pone orizzontalmente rispetto ad esso. Il tempo di integrazione è stato impostato a 30 ms, la mano distava circa 40 cm dal sensore e sono stati acquisiti 100 frame.

Si è rivelato particolarmente importante per questo tipo di test il parametro *selettività* in fase di filtraggio: per eliminare i punti appartenenti ai bordi della mano, che presentano scarsa affidabilità nelle misure di profondità, è stata impostata una *selettività* di 0.6 (sono stati dunque rimossi i punti con coseno tra la normale e la media delle normali inferiore a 0.6).

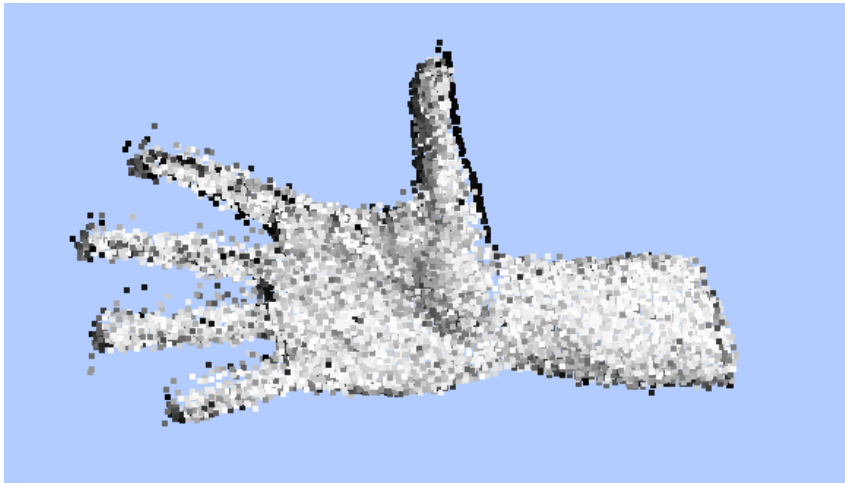


Figura 5.26: Ricostruzione 3D della mano



Figura 5.27: Ricostruzione 3D della mano (post-processing)

Capitolo 6

Conclusioni

La ricostruzione tridimensionale di oggetti a partire da un sensore a tempo di volo è un problema interessantissimo che trova applicazione in svariati ambiti. Il coinvolgimento di molte aziende e organi di ricerca sui sensori a tempo di volo ci dice molto sulle potenzialità applicative degli stessi.

Il contributo principale di questa ricerca è stata l'applicazione di un sensore a tempo di volo per la ricostruzione di oggetti tridimensionali in tempo reale, operazione tradizionalmente legata ad altri dispositivi più costosi e precisi come gli scanner laser, che tuttavia non effettuano elaborazioni di ricostruzione in parallelo alla fase di acquisizione e in tempo reale. I risultati ottenuti mettono in risalto in trade-off tra accuratezza e velocità di esecuzione, le potenzialità della tecnologia, i suoi limiti ma anche i suoi vantaggi rispetto ad altri tipi di tecnologie.

I principali avanzamenti che possono essere apportati al progetto in futuro sono l'elaborazione di un filtro anti-rumore specifico per sensori a tempo di volo, in grado di migliorare la qualità dei dati, e l'uso di informazioni di colore, sia per migliorare la robustezza dell'algoritmo ICP, sia per arricchire la visualizzazione dei risultati ottenuti.

Prevedibilmente questi avanzamenti porteranno ad un degrado delle prestazioni del programma in termini di velocità di esecuzione. Significativi miglioramenti su questo versante possono essere ottenuti con l'implementazione del programma su architetture parallele, in particolare su schede grafiche CUDA [37].

Capitolo 7

Appendice

7.1 Antidistorsione e conversione da range map a nuvola di punti

Per una caratterizzazione accurata della camera (modellata secondo il noto modello pinhole [16, 17]) è necessario venire a conoscenza delle distanze focali, del centro ottico e dei parametri di distorsione radiale e tangenziale. Le distorsioni (radiale e tangenziale), causate dalle imprecisioni dei componenti ottici presenti nella camera, devono essere compensate da una opportuna operazione di rettifica. La calibrazione dei parametri intrinseci è il procedimento che permette di ricavare una descrizione della proiezione dei punti del mondo reale nell'immagine prodotta dalla camera. Una volta in possesso delle immagini rettificate, è possibile ricavare una rappresentazione matriciale in coordinate omogenee che modelli la proiezione dei punti tridimensionali [17].

$$z * \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} -fk_u & fk_u \cot(\theta) & u_0 & 0 \\ 0 & -fk_v / \sin(\theta) & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} * \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}$$

$m = [u, v]^T$ riporta le coordinate del punto proiettato sull'immagine e $P = [x, y, z]^T$ le coordinate del punto nello spazio tridimensionale nel sistema di riferimento delle camere. Le lunghezze focali orizzontali e verticali, espresse in pixel, sono rispettivamente $-fk_u$ e $-fk_v$, θ è l'angolo tra gli assi u e v e $[u_0, v_0]$ sono le coordinate del punto principale.

Per calibrazione dei parametri intrinseci si intende quindi l'operazione di stima dei parametri:

1. $-fk_u$
2. $-fk_v$

3. u_0
4. v_0
5. θ
6. distorsione radiale e tangenziale

che permettono di descrivere accuratamente la proiezione dei punti del mondo reale durante il processo di formazione dell'immagine. Questo modello di proiezione risulta verificato solamente se le coordinate dei punti nello spazio tridimensionale $[x, y, z]^T$ sono espresse relativamente al sistema di riferimento identificato dalla camera stessa, ovvero quando il sistema di riferimento del mondo (WRF) e quello della camera (CRF) coincidono.

L'operazione di calibrazione [6, 7], che fa uso di immagini con pattern noti come i quadrati di una scacchiera, è affidata al componente software *ToolCalibrazione*, di cui si è accennato nel capitolo “*Descrizione del programma*”.

Capitolo 8

Ringraziamenti

Un particolare ringraziamento va al Professore Guido Maria Cortellazzo, per la preziosa possibilità di crescita che mi ha concesso e per la sua costante disponibilità nei miei confronti.

Ringrazio anche i Dott. Pietro Zanuttigh e Carlo Dal Mutto, che hanno supportato la mia attività di ricerca presso il laboratorio di tecnologie e telecomunicazioni multimediali.

Infine un “grazie” speciale ai miei genitori, che hanno sempre saputo mettermi nelle condizione di rendere al meglio, e ai miei amici, che non hanno mai mancato di sostenermi nelle sfide di ogni giorno.

Bibliografia

- [1] M. Scherer, “The 3d tof camera as an innovative and low cost tool,” October 2009. cipa.icomos.org/text%20files/KYOTO/30.pdf.
- [2] O. H.-H. Szymon Rusinkiewicz and M. Levoy, “Real-time 3d model acquisition,” 2002. http://graphics.stanford.edu/papers/rt_model/rt_model.pdf.
- [3] Cui, Yan, Schuon, Sebastian, Derek, Chan, Thrun, Sebastian, Theobalt, and Christian, “3d shape scanning with a time-of-flight camera,” *In Proc. of IEEE CVPR 2010*, 2010. ai.stanford.edu/~schuon/sr/cvpr10_scanning.pdf.
- [4] S. Schuon, C. Theobalt, J. Davis, and S. Thrun, “High-quality scanning using time-of-flight depth superresolution,” *CVPR Workshop on Time-of-Flight Computer Vision 2008*, 2008. ai.stanford.edu/~schuon/sr/cvpr08_superresolution.pdf.
- [5] S. Schuon, C. Theobalt, J. Davis, and S. Thrun, “Lidarboost: Depth super-resolution for tof 3d shape scanning,” *In Proc. of IEEE CVPR 2009*, 2009. www.mpi-inf.mpg.de/~theobalt/cvpr09_superresolution.pdf.
- [6] C. Dal Mutto, P. Zanuttigh, and G. M. Cortelazzo, “Accurate 3d reconstruction by stereo and tof data fusion,” in *GTTI*, (Brescia, Italy), June 2010. http://www.dei.unipd.it/~dalmutto/Papers/10_GTTI.pdf.
- [7] G. M. C. Carlo Dal Mutto, Pietro Zanuttigh, “A probabilistic approach to tof and stereo data fusion,” in *3DPVT*, (Paris, France), May 2010. lttm.dei.unipd.it/nuovo/Papers/10_3DPVT.pdf.
- [8] “Mesa imaging.” <http://www.mesa-imaging.ch/index.php>.
- [9] F. Bernardini and H. Rushmeier, “The 3d model acquisition pipeline,” *Computer Graphics Forum*, vol. 21, no. 2, pp. 149–172, 2002. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.94.7486&rep=rep1&type=pdf>.
- [10] A. W. Moore, “Chapter 6 kd-trees for cheap learning,” 1991. <http://www.mlr.ufl.edu/~nechyba/papers/kdtree-2.pdf>.
- [11] F. P. Preparata and M. I. Shamos, *Computational geometry: an introduction*. New York, NY, USA: Springer-Verlag New York, Inc., 1985.

- [12] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. New York: The MIT Press, 2001.
- [13] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proceedings of the Sixth International Conference on Computer Vision, ICCV '98*, (Washington, DC, USA), pp. 839–, IEEE Computer Society, 1998. users.soe.ucsc.edu/~manduchi/Papers/ICCV98.pdf.
- [14] P. Besl and N. McKay, "A method for registration of 3-d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 239–256, 1992. <http://www.martinkodde.nl/glacier/data/bibliography/289461888100121791.pdf>.
- [15] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image Vision Comput.*, vol. 10, pp. 145–155, April 1992. <http://portal.acm.org/citation.cfm?id=138628.138633>.
- [16] R. Szeliski, *Computer Vision: Algorithms and Applications*. New York, NY, USA: Springer-Verlag New York, Inc., September 2010.
- [17] A. Fusiello, "Visione computazionale - appunti delle lezioni." Pubblicato a cura dell'autore, Giugno 2008. <http://ilmiolibro.kataweb.it/schedalibro.asp?id=387047>.
- [18] K. Pulli, "Multiview registration for large data sets," *3D Digital Imaging and Modeling, International Conference on*, vol. 0, p. 0160, 1999. graphics.stanford.edu/~kapu/3dim99.pdf.
- [19] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Third International Conference on 3D Digital Imaging and Modeling (3DIM)*, June 2001. http://www.cs.princeton.edu/~smr/papers/fasticp/fasticp_paper.pdf.
- [20] D. Simon, *Fast and Accurate Shape-Based Registration*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, December 1996. http://www.ri.cmu.edu/pub_files/pub1/simon_david_1996_1/simon_david_1996_1.pdf.
- [21] T. Jost and H. Hügli, "Fast icp algorithms for shape registration," in *Proceedings of the 24th DAGM Symposium on Pattern Recognition*, (London, UK, UK), pp. 91–99, Springer-Verlag, 2002. www.springerlink.com/index/r6bnpm9p23hpathr.pdf.
- [22] P. J. Neugebauer, "Geometrical cloning of 3d objects via simultaneous registration of multiple range images," in *Proceedings of the 1997 International Conference on Shape Modeling and Applications (SMA '97)*, (Washington, DC, USA), pp. 130–, IEEE Computer Society, 1997. <http://www.eecs.umich.edu/~silvio/teaching/EECS598/papers/NeugebSMI97.pdf>.

- [23] S. R. Natasha Gelfand, Leslie Ikemoto and M. Levoy, “Geometrically stable sampling for the ICP algorithm,” in *Fourth International Conference on 3D Digital Imaging and Modeling (3DIM)*, October 2003. www-graphics.stanford.edu/papers/stabicp/stabicp.pdf.
- [24] G. Blais, M. D. Levine, and M. D. Levine, “Registering multiview range data to create 3d computer objects,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 820–824, 1993. <http://citeseer.ist.psu.edu/viewdoc/download;jsessionid=DDFBD67E06812E91D682706889B993A3?doi=10.1.1.57.2506&rep=rep1&type=pdf>.
- [25] D. Hyun, I. D. Yun, and S. U. Lee, “Registration of multiple range views using the reverse calibration technique,” 1998. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.34.7910&rep=rep1&type=pdf>.
- [26] M. Kazhdan, M. Bolitho, and H. Hoppe, “Poisson surface reconstruction,” in *Proceedings of the fourth Eurographics symposium on Geometry processing, SGP '06*, (Aire-la-Ville, Switzerland, Switzerland), pp. 61–70, Eurographics Association, 2006. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.153.4724&rep=rep1&type=pdf>.
- [27] N. Amenta, S. Choi, and R. K. Kolluri, “The power crust,” in *Proceedings of the sixth ACM symposium on Solid modeling and applications, SMA '01*, (New York, NY, USA), pp. 249–266, ACM, 2001. www.cs.ucdavis.edu/~amenta/pubs/sm.pdf.
- [28] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, “Surface reconstruction from unorganized points,” *SIGGRAPH Comput. Graph.*, vol. 26, pp. 71–78, July 1992. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.5.569&rep=rep1&type=pdf>.
- [29] “Opencv library.” <http://opencv.willowgarage.com/wiki/>.
- [30] A. K. Gary Bradski, *Learning OpenCV*. O’Reilly Media, September 2008.
- [31] “Vtk library.” <http://www.vtk.org/>.
- [32] “Mesa sr4000 user manual,” 2010. http://www.mesa-imaging.ch/dlm.php?fname=customer/Custom_CD/SR4000_Manual.pdf.
- [33] “Microsoft MFC.” [http://msdn.microsoft.com/en-us/library/d06h2x6e\(v=vs.80\).aspx](http://msdn.microsoft.com/en-us/library/d06h2x6e(v=vs.80).aspx).
- [34] “Meshlab.” <http://meshlab.sourceforge.net/>.
- [35] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin, “The ball-pivoting algorithm for surface reconstruction,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, pp. 349–359, 1999. http://www.research.ibm.com/vistechnology/pdf/bpa_tvvcg.pdf.

- [36] "Nextengine 3d scanner hd." <https://www.nextengine.com/>.
- [37] "Cuda framework." http://www.nvidia.it/object/cuda_home_new_it.html.