

ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA

Dipartimento di Informatica - Scienza e Ingegneria
Corso di Laurea Magistrale in Ingegneria e Scienze Informatiche

Sviluppo di una Libreria di Geometria Computazionale per Comparare Metodi di Presa Misura in Pazienti Amputati di Arto Inferiore

Tesi di laurea in
VISIONE ARTIFICIALE E RICONOSCIMENTO

Relatore
Prof.ssa Annalisa Franco

Candidata
Linda Guiducci

Correlatore
Ing. Andrea Giovanni Cutti, PhD, TO

III Sessione di Laurea
Anno Accademico 2019-2020

*Se non avete ancora trovato ciò che fa per voi, continuate a cercare,
non fermatevi, come capita per le faccende di cuore,
saprete di averlo trovato non appena ce l'avrete davanti.*
Steve Jobs

Sommario

Il progetto descritto in questo elaborato di tesi è stato svolto presso Il Centro Protesi INAIL (Vigorso di Budrio, BO).

Il lavoro è stato realizzato a supporto di un progetto di ricerca, finanziato dal Dipartimento della Difesa USA, in collaborazione con la Northwestern University di Chicago e il Minneapolis Veteran Affairs Health Care Sytem. La ricerca ha lo scopo di determinare l'efficacia comparativa di metodi alternativi per realizzare il calco del moncone dell'amputato di arto inferiore e la successiva invasatura su misura.

Il progetto di tesi nasce dall'assenza di un software commerciale in grado di analizzare come evolve la forma del moncone, dal calco all'invasatura finita, basandosi sulla digitalizzazione tridimensionale delle superfici. La libreria sviluppata è implementata in Python e utilizza algoritmi e strumenti di geometria computazionale al fine di supportare i processi di elaborazione dati.

Il flusso di lavoro si sviluppa nelle seguenti fasi:

- Acquisizione e pre-processing del dato;
- Identificazione digitale dei punti di repere;
- Allineamento dei modelli per orientarli in un sistema di riferimento globale secondo una logica comune;
- Registrazione di due modelli per allinearli l'uno all'altro;
- Generazione di outcome e parametri dimensionali, derivanti da mappe di stanza, sezioni, cammini geodetici e regioni di interesse;
- Estrazione di indicatori statistici riassuntivi delle differenze, correlate ad un insieme di scansioni tramite la PCA.

Le funzionalità sono state validate tramite appositi test su dati clinici rilevati dallo studio o dati sintetici con caratteristiche note a priori. La libreria fornisce un insieme di interfacce che permette l'accesso anche a utenti non esperti ed

è caratterizzata da modularità, semplicità di installazione ed estensibilità delle funzionalità.

Tra gli sviluppi futuri si prevede l'identificazione di possibili ottimizzazioni individuate da un utilizzo degli strumenti esteso a più casi d'uso.

Indice

Sommario	v
1 Introduzione	1
1.1 Il Centro Protesi INAIL	1
1.2 Tipologie di amputazione e protesi	2
1.3 Tecniche di presa misura e sviluppo dell'invasatura	6
1.4 Obiettivo della tesi	7
1.4.1 Organizzazione dell'elaborato	9
2 Stato dell'arte	11
2.1 Metodi di presa misura	11
2.1.1 Metodo tradizionale	11
2.1.2 Metodo idrostatico	13
2.1.3 Scanner 3D per la digitalizzazione dei modelli	14
2.2 Scopo dello studio	16
2.3 Disegno dello studio	17
2.3.1 Punti di repere anatomici per comparazioni inter-tecnica, inter-operatore e inter-soggetto	19
3 Materiali e metodi	21
3.1 Flusso di lavoro	21
3.1.1 Acquisizione dati	24
3.1.2 Pre-processing dato	26
3.1.3 Selezione landmark	26
3.1.4 Allineamento	26
3.1.5 Registrazione	30
3.1.6 Preparazione dato per il mesh morphing	43
3.1.7 Outcomes	44
3.1.8 PCA	54
3.2 Framework e librerie	56
3.2.1 Visualization Toolkit (VTK)	56

3.2.2	AmpScan	62
3.2.3	MeshLab	63
4	Risultati	65
4.1	Prototipo realizzato	65
4.1.1	Implementazione algoritmo registrazione Sanders	67
4.2	Test e validazione	70
4.2.1	Selezione landmark	70
4.2.2	Allineamento	71
4.2.3	Registrazione	74
4.2.4	Outcomes	76
4.2.5	PCA	86
5	Conclusioni e sviluppi futuri	91
A	Test di registrazione	93
A.1	Registrazione basata su landmark	93
A.2	Iterative Closest Point	95
A.3	Thin plate spline	98
A.4	Analisi di Procrustes ordinaria	99
A.5	Algoritmo ispirato a Sanders	102

Elenco delle figure

1.1	Livelli di amputazione dell'arto inferiore	3
1.2	Liner di diverse misure	4
1.3	Invasatura realizzata dal Centro Protesi INAIL	5
1.4	Principali componenti protesi modulare	6
1.5	Bende gessate	7
2.1	Fasi di presa misura tramite metodo manuale	12
2.2	Symphonie Aqua System	13
2.3	Dispositivo di controllo Aqua System	14
2.4	Cilindri Aqua System	14
2.5	Robot meccanico Ortis	15
2.6	Scanner da interni e scanner portatile	15
2.7	Fasi dello studio e obiettivi da soddisfare	18
2.8	Riferimenti repere anatomici TT	19
2.9	Riferimenti repere anatomici TF	20
2.10	Diverse tipologie di marcatori	20
3.1	Flusso di lavoro.	22
3.2	Esempio di triangolazione per file STL	25
3.3	Faccia in formato STL	25
3.4	Rappresentazione delle fasi di generazione dell'asse d'inerzia con metodo "robusto".	28
3.5	Punti omologhi su cui si basa la registrazione	31
3.6	Schema generico del flusso di registrazione.	31
3.7	Flusso di operazioni algoritmo ICP.	33
3.8	Allineamento basato sull'algoritmo ICP	35
3.9	Partizionamento dello spazio 3D per la creazione del k-d tree	36
3.10	Esempio di deformazione con TPS per portare i landmark in corrispondenza	37
3.11	Trasformazione con TPS per diversi valori di λ	38
3.12	Fasi dell'allineamento di Procrustes	40

3.13	Esempio di analisi di Procrustes generalizzata	41
3.14	Proiezione di d per l'ottenimento di RE-INAIL	45
3.15	Confronto tra distanza RE-INAIL e Vertex-Vertex	46
3.16	Raffigurazione IQR	48
3.17	Esempi di grafi generati da diverse mesh	50
3.18	Esempi di percorsi geodetici	52
3.19	Esempio d'evoluzione dell'algoritmo di Dijkstra	54
3.20	Esempi di data object di VTK	57
3.21	Struttura del dataset in VTK	58
3.22	Esempio schematico della VTK visualization network.	58
3.23	Schema della pipeline di visualizzazione in VTK	60
3.24	Esempio di interfaccia sviluppata tramite PyQt e VTK.	61
3.25	Flusso dei processi gestiti da AmpScan	63
4.1	Diagramma dei moduli del software.	66
4.2	Use case diagram dell'applicazione.	67
4.3	Visualizzazione dei landmark selezionati su una superficie	71
4.4	Confronto tra il metodo di allineamento inerziale e anatomico	73
4.5	Confronto aree e perimetri sezioni TT e TF	73
4.6	Risultato dell'analisi di Procrustes generalizzata	76
4.7	Esempio Mappa RE-INAIL	78
4.8	Esempio di box plot associato ad RE-INAIL	79
4.9	Esempio di mappe SNAE e Normal Similarity	80
4.10	Esempio di ROI estratta da una mesh.	81
4.11	Risultato della query eseguita sui valori della ROI	81
4.12	Istogramma ROI	82
4.13	Istogramma query 1	82
4.14	Istogramma query 2	82
4.15	Raffigurazione Sezioni	84
4.16	Cammini geodetici per TT	85
4.17	Raffigurazione del percorso geodetico fra due landmark e punto d'interrogazione.	85
4.18	Use case diagram PCA	86
4.19	Esempi provenienti dal dataset su cui è stata testata la PCA	87
4.20	Forma media estratta tramite la PCA.	88
4.21	Forme generate dalla PCA al variare dei pesi associati alle componenti	88
4.22	Dataset utilizzato per la PCA differenziale.	89
4.23	Esempi di generazione di forme tramite la PCA differenziale	89

Capitolo 1

Introduzione

Il lavoro descritto in questo progetto di tesi è stato svolto presso il Centro Protesi INAIL, presso Vigorso di Budrio (BO). Il centro si occupa della creazione di protesi e ortesi su misura per pazienti con amputazione, ed è anche noto come centro di ricerca di nuove tecniche e tecnologie per tali dispositivi.

In particolare, il presente lavoro è stato realizzato a supporto di un progetto di ricerca, finanziato dal Dipartimento della Difesa USA, in collaborazione con la Northwestern University di Chicago (USA) e il Minneapolis Veteran Affairs Health Care System (USA). La ricerca ha lo scopo di determinare l'efficacia comparativa di metodi alternativi per la realizzazione del calco del moncone dell'amputato di arto inferiore e la successiva realizzazione di un'invasatura su misura.

1.1 Il Centro Protesi INAIL

Il Centro Protesi INAIL, fondato nel 1961, è un'azienda certificata ISO 9001-2015 ed è una delle strutture sanitarie dedicate alla fornitura di dispositivi ortopedici su misura più grandi al mondo.

Dal 1984 il Centro Protesi ha raggiunto l'assetto definitivo nella veste di *Centro per la sperimentazione ed applicazione di protesi e presidi ortopedici* [1] e le attività che lo interessano sono principalmente:

- La diretta fornitura di cure mediche in aggiunta alle cure standard fornite dal Sistema Sanitario Nazionale (SSN);
- La ricerca di nuove tecnologie finalizzate alla produzione di protesi e ortesi;
- La produzione e fornitura di protesi e presidi ortopedici;
- La riabilitazione ed addestramento all'uso della protesi;

- La completa reintegrazione dell'infortunato nel mondo del lavoro, nella famiglia e più ampiamente nella società.

La struttura è suddivisa in linee di produzione sulla base della tipologia di protesi realizzate. Esistono, quindi, la linea protesi di arto superiore, la linea protesi in silicone, la linea protesi transfemorale, la linea protesi transtibiale, la linea delle ortesi e la linea calzature ortopediche.

1.2 Tipologie di amputazione e protesi

Con il termine *amputazione* si fa riferimento all'asportazione chirurgica di un arto, superiore o inferiore, o di parte di esso, a seguito di un trauma fisico o di una patologia. Le cause più comuni che portano all'amputazione sono il diabete mellito, la malattia vascolare periferica, la neuropatia e il trauma. Secondo il Ministero della Salute l'incidenza delle amputazioni in Italia legate al diabete è del 60% [2]. Durante il periodo 2001-2010, un numero medio annuo di 11.639 individui ha subito un'amputazione di arto inferiore, avvenuta nella maggior parte dei casi a livello transtibiale [3].

L'amputazione può essere eseguita a vari livelli, a seconda della condizione in cui si trova il paziente. Il principio di base è quello di preservare l'arto il più possibile, poiché un moncone più lungo consentirà una migliore guida muscolare dell'arto una volta indossata la protesi. In Figura 1.1 sono riportati i vari livelli ai quali può avvenire l'amputazione. Quelli d'interesse per questo lavoro di tesi sono in particolare:

- Transtibiale (TT): amputazione dell'osso tibiale;
- Transfemorale (TF): amputazione dell'osso femorale.

La **protesi** è un dispositivo che sostituisce l'arto amputato e permette al paziente di recuperare il più possibile la funzionalità motoria e poter condurre uno stile di vita simile a quello precedente l'amputazione. Le protesi transtibiali e transfemorali si distinguono in:

- Esoscheletrica (tradizionale), ad oggi poco utilizzate in quanto risultano essere pesanti e non è possibile applicare alcun tipo di piede a tecnologia avanzata;
- Endoscheletrica (modulare), hanno un peso inferiore essendo costruite da leghe leggere, poiché hanno lo scopo di ridurre l'affaticamento del paziente dovuto al suo utilizzo quotidiano.

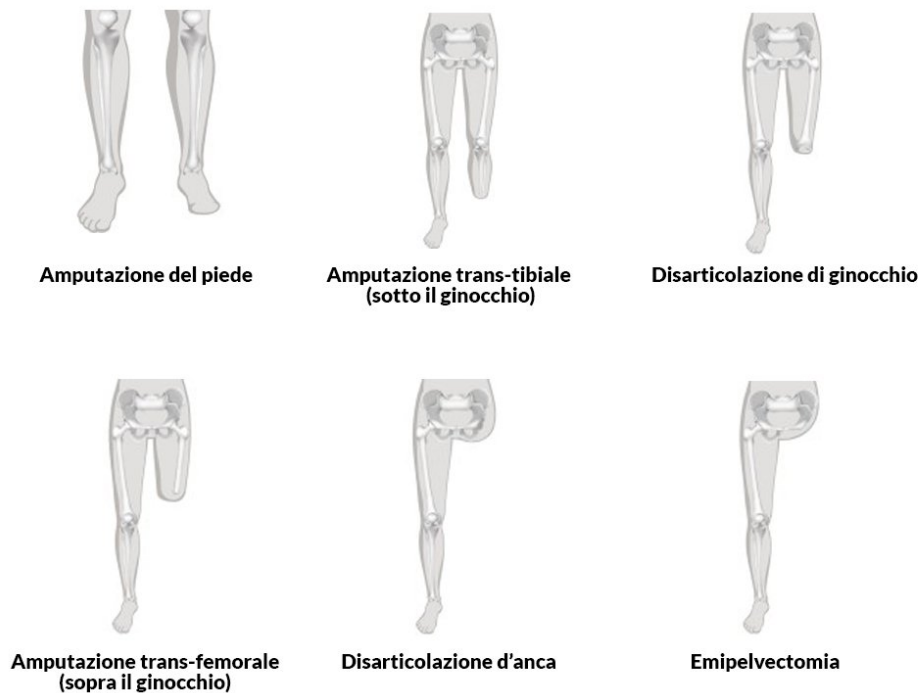


Figura 1.1: Livelli di amputazione dell'arto inferiore. Fonte: <https://biomedicalcuc.it/feedback-sensoriale-integrato-protesi-aiutare/28737/>

La protesi modulare è formata da vari componenti (visibili in Figura 1.4), con diverse funzioni, che si interfacciano tra loro grazie alla presenza di collegamenti. Tali componenti sono:

- **Liner:** il liner (fig. 1.2) è un rivestimento per il moncone simile a una cuffia che viene realizzato in materiale molto flessibile e ammortizzante. È l'unico componente a diretto contatto con la pelle del paziente e funge da interfaccia tra il tessuto molle e mobile del moncone e l'invasatura. Esso è fondamentale per il comfort del paziente e per creare uno strato protettivo al moncone, evitando urti e sfregamenti diretti con l'invasatura. Ha una forma conico-cilindrica, chiusa nella parte distale, e viene scelto per ogni paziente in base alla misura del diametro del moncone ad una determinata altezza dall'apice. È essenziale scegliere il liner corretto per assicurare che la protesi sia applicata correttamente e risulti piacevole da indossare [4].



Figura 1.2: Liner di diverse misure. [4]

- **Invasatura:** l'invasatura (fig. 1.3) è l'elemento strutturale rigido o flessibile che contiene il moncone d'amputazione. Deve assolvere a funzioni fra loro opposte che devono essere ottimizzate:
 - Contenere ed aderire al moncone affinché la protesi non sfugga e venga percepita come un'estensione naturale del corpo del paziente;
 - Consentire alle masse muscolari di espandersi durante il movimento;
 - Permettere la massima ampiezza dei movimenti delle articolazioni residue;
 - Supportare il paziente in caso di deficit muscolare;
 - Essere confortevole, tutelare la cute e le zone sensibili o potenzialmente dolorose;
 - Essere leggera;
 - Resistere alle sollecitazioni meccaniche;
 - Contenere o ridurre la carica batterica ed essere igienizzabile [5].

É quindi indispensabile per l'accettazione della protesi e il benessere dell'utente. Per raggiungere questi obiettivi, l'invasatura viene realizzata su misura dal tecnico ortopedico, a seconda della forma del moncone e in base al grado di mobilità del paziente, permettendo, tipicamente, una distribuzione uniforme del carico [4]. Di particolare importanza è anche la scelta dei materiali, come ad esempio, silicone e carbonio pre-preg. La struttura e il comfort vengono infatti ottimizzati anche tramite l'integrazione multipla dei materiali.



Figura 1.3: Invasatura con copertura estetica realizzata dal Centro Protesi INAIL. [5]

- **Pilone:** il pilone è un tubo metallico cavo che collega l'invasatura al piede protesico: equivale alla tibia nei soggetti sani. I materiali di realizzazione sono di norma acciaio inox e titanio.
- **Piede:** il piede protesico è un arto artificiale che sostituisce il piede fisiologico, è dunque il componente che si interfaccia con il terreno. Per questo motivo, è necessario che sia di sostegno al paziente e che riproduca il comportamento fisiologico del piede, ovvero che assorba gli urti nella parte posteriore e restituisca energia nella fase di spinta della parte anteriore. Il piede migliore per un paziente viene scelto in relazione a peso, grado di mobilità e tipo attività che il paziente si troverà a svolgere.
- **Moduli di collegamento:** permettono il collegamento tra i vari componenti. Si utilizza una placchetta metallica, fissata con delle viti all'invasatura, per connetterla al pilone. Un giunto di torsione, montato sopra il piede protesico, consente un movimento di rotazione della protesi senza dover muovere il piede.



Figura 1.4: Principali componenti protesi modulare. Fonte: <https://www.infinitetech.org/above-knee-leg-prosthetics/>

1.3 Tecniche di presa misura e sviluppo dell'invasatura

Il primo passo per la realizzazione dell'invasatura è la *presa misura* del moncone del paziente, realizzata dal tecnico ortopedico. La presa misura si realizza mediante un calco gessato dell'arto amputato. Esistono diverse metodiche per completare questo passaggio, che per lo scopo di questa tesi possiamo sinteticamente riassumere come *tecnica manuale* e *tecnica idrostatica*.

Alla tecnica manuale fanno capo tutte quelle metodiche che richiedono l'esecuzione del calcolo totalmente a mano. Il tecnico ortopedico, dopo avere protetto il liner del paziente con della pellicola di PVC, srotola una benda gessata (fig. 1.5) bagnata sul moncone e la modella con le proprie mani, applicando delle pressioni in aree specifiche. Una volta asciutta, la benda gessata rappresenta il negativo grezzo su cui poi verrà sviluppata l'invasatura.

Nella tecnica idrostatica, è un sistema meccanico ad applicare la pressione alla benda gessata. Nello specifico, il paziente posiziona il moncone all'interno di un cilindro entro il quale vi è una membrana in silicone, che può essere espansa con acqua sotto pressione. Il tecnico regola la pressione fino a quanto il paziente, in ortostatismo all'interno del cilindro, non si trova in condizione di "galleggiamento". In altre parole, la pressione applicata dalla membrana è tale da equilibrare il peso



Figura 1.5: Esempi di bende gessate asciutte utilizzate per la creazione del negativo

del paziente in carico monopodalico nel sistema. Da qui il nome tecnica idrostatica.

A partire dal negativo grezzo, il tecnico ortopedico ottiene un positivo colando al suo interno del gesso e modifica quindi il positivo grezzo come fosse una “scultura”. Questa azione viene definita *stilizzazione del positivo*, o *rectification*. Si ottiene quindi un positivo stilizzato.

Dal positivo stilizzato viene realizzata l’invasatura mediante termoformatura di una lastra di materiale termoplastico o mediante laminazione di un *sandwich* di nylon e carbonio con resina acrilica.

1.4 Obiettivo della tesi

Attualmente è assente in letteratura uno studio comparativo di efficacia fra tecnica manuale e tecnica idrostatica, sia per pazienti transtibiali che transfemorali.

A tal proposito, nel 2020 ha preso il via uno studio scientifico di durata triennale, finanziato dal Dipartimento della Difesa degli Stati Uniti, che coinvolge la Northwestern University di Chicago, il Minneapolis VA Health Care System e il Centro Protesi di Vigorso di Budrio.

Il titolo dello studio è *Comparative Effectiveness of Socket Casting Methods: Improving Form and Fit*, è un progetto interventistico non farmacologico, monocentrico sul territorio italiano e multicentrico a livello internazionale [6].

Lo studio prevede il coinvolgimento, in ognuno dei centri partner presenti sui due lati dell’Atlantico, di 30 pazienti – 10 con amputazione transfemorale e 20 con amputazione transtibiale – che collaboreranno attivamente con i ricercatori, testando due invasature realizzate secondo la tecnica manuale ed idrostatica e individuando quella per loro più confortevole.

I risultati potranno supportare la scelta della tecnica di costruzione da preferire nella routine clinica e porteranno alla stesura di linee guida applicative, fondamentali per i pazienti e per chiunque operi in ambito tecnico-protetico.

La tecnica idrostatica è già applicata presso il Centro Protesi INAIL e la Northwestern University. Il Centro Protesi INAIL ha raccolto evidenze preliminari circa la soddisfazione dei pazienti con amputazione transfemorale per le invasature realizzate mediante questa tecnica, riportando un numero minimo di rifacimenti ed un miglioramento della soddisfazione dei pazienti [7].

In modo simile, la Northwestern University ha evidenziato un miglioramento della soddisfazione dei pazienti per la tecnica idrostatica in pazienti transtibiali. L'ipotesi complessiva è che

la tecnica idrostatica sia un'alternativa più ripetibile ed efficiente rispetto alla tecnica manuale per realizzare il calco del moncone del paziente, e che permetta di ottenere un'invasatura immediatamente più confortevole [7].

Per potere quantificare le differenze in termini di volume e forma fra i negativi ottenuti con le due tecniche, così come i relativi positivi grezzi, positivi stilizzati ed invasature, il progetto prevede di digitalizzare la forma dei manufatti durante la loro costruzione, mediante scansione digitale tridimensionale (scanner 3D). Successivamente, le scansioni vengono elaborate per estrarre parametri quantitativi e permetterne la comparazione.

Il progetto di tesi nasce dall'assenza di un software commerciale in grado di realizzare tali analisi. A tal proposito si è realizzata una libreria di geometria computazionale che permetta di agevolare l'elaborazione dei dati. Le funzionalità implementate saranno di ausilio per poter comprendere gli errori insiti nella produzione dei dispositivi ortopedici su misura, compresa la propagazione dell'errore durante la catena di processo.

Nello specifico, gli obiettivi sono i seguenti:

- Analisi del flusso di misura e del formato dei dati di lavoro;
- Analisi dello stato dell'arte e delle librerie specifiche già disponibili;
- Implementazione di una libreria per l'analisi dei dati e la comparazione delle scansioni mediante tecniche di valutazione dimensionale;
- Test delle funzioni implementate e applicazione a dati reali per trarne evidenze e conclusioni.

1.4.1 Organizzazione dell'elaborato

Nel **Capitolo 2** verrà descritto nel dettaglio lo stato dell'arte, ovvero verrà eseguita una panoramica del contesto in cui si inserisce il lavoro di tesi. Verranno descritte le principali modalità di presa misura e gli strumenti di digitalizzazione dei modelli. Sarà successivamente approfondito lo studio scientifico, descrivendone scopo, struttura e risultati attesi.

Il **Capitolo 3** descriverà il flusso di lavoro elaborato per raggiungere gli obiettivi prefissati. Le fasi di esso saranno illustrate alla luce di strumenti, algoritmi, framework e librerie utilizzati.

Il **Capitolo 4** tratterà nel dettaglio la libreria sviluppata indicandone architettura e casi d'uso specifici. Saranno presentate inoltre le attività di validazione e test delle funzionalità offerte dalla libreria, presentandone l'applicazione su dati reali.

Nel **Capitolo 5** verranno specificati i requisiti soddisfatti e presentati gli sviluppi futuri associati al lavoro svolto.

Capitolo 2

Stato dell'arte

2.1 Metodi di presa misura

Non esiste un modo univoco per la costruzione dell'invasatura e il processo è largamente dipendente da abilità, conoscenze ed esperienza del tecnico ortopedico (TO).

Nel campo ortopedico vi sono più opzioni per la realizzazione del calco del moncone, per la modifica del relativo positivo, per la scelta dei materiali dell'invasatura e per l'allineamento di essa con le altre parti della protesi. Non è quindi raro dover realizzare più invasature per giungere a quella soddisfacente, il che può comportare notevole stress al paziente, alti costi e lunghi tempi per completare la fornitura.

2.1.1 Metodo tradizionale

Il metodo tradizionale utilizzato ad oggi per realizzare l'invasatura definitiva di pazienti con amputazione di arto inferiore è complesso e coinvolge diversi tecnici specializzati. Il processo convenzionale di fabbricazione dell'invasatura protesica consiste di varie fasi [7]:

- **La fase preliminare:**
 - Analisi del moncone;
 - Acquisizione delle misure del moncone del paziente tramite l'utilizzo di un calibro e un metro da sarta non elasticizzato;
 - Determinazione della misura della cuffia;
- La creazione del **calco di gesso negativo** (fig. 2.1.a):

- Realizzazione del calco tramite l'utilizzo di bende gessate: esse vengono avvolte attorno all'arto e con la pressione delle mani il TO impone una forma al gesso, utile per la successiva forma da dare all'invasatura;
- Identificazione dei punti di repere anatomici e contorno della rotula, nel caso di amputazione transtibiale, tramite pressione manuale (fig. 2.1.b);
- La creazione del **calco positivo** (fig. 2.1.c): riempimento dello stampo negativo con gesso liquido per realizzare lo stampo che possa poi essere rettificato (fig. 2.1.d);
- La fabbricazione di una **invasatura diagnostica** in materiale termoformabile solitamente trasparente, consentendo al TO una visione completa per valutare l'adattamento e apportare modifiche (fig. 2.1.e e fig. 2.1.f);
- La realizzazione dell'invasatura personalizzata, tramite processi di laminazione utilizzando carbonio e resina per creare la **protesi definitiva**.



Figura 2.1: Principali fasi della presa misura con metodo manuale.

La tecnica manuale è una tecnica ormai consolidata e di ampio utilizzo, tuttavia è difficile catturare con precisione i contorni dell'anatomia ossea e distribuire la

pressione in modo uniforme attorno all'arto residuo. Di conseguenza, sono necessarie successive modifiche durante la prova dell'invasatura per garantire un comfort adeguato e i *cast* (modelli in gesso) sono raramente riproducibili e coerenti [7].

2.1.2 Metodo idrostatico

In alternativa alle tecniche manuali per la realizzazione del calco si sta diffondendo a livello internazionale la tecnica denominata *A pressione idrostatica con paziente in ortostatismo monopodalico* sul lato amputato, che verrà chiamata nel seguito tecnica idrostatica.

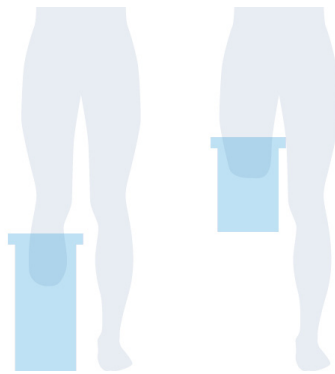


Figura 2.2: Symphonie Aqua System: per la cura delle amputazioni transtibiali (VC TT) e per la cura delle amputazioni femorali (VC TF). [8]

Tale tecnica viene realizzata oggi tipicamente mediante l'ausilio di un sistema commerciale denominato Symphonie Aqua System (Romedis GmbH, Germania) [8].

Il sistema consiste in un cilindro cavo con all'interno una sottile membrana di silicone. Il TO riempie con acqua lo spazio fra la parete esterna del cilindro e la membrana mediante un sistema idraulico (fig. 2.3) collegato alla rete idrica.

Una volta applicate delle bende gessate umide sul moncone, al paziente viene chiesto di portare il moncone all'interno del cilindro. Il TO può quindi insufflare la membrana con acqua. Man mano che la membrana si espande, il paziente sposta il peso sempre più sul lato amputato, finché non si realizza una sospensione idrostatica del paziente in appoggio monopodalico sul moncone: il paziente è come se "galleggiasse" nel cilindro. Una volta che la benda si è solidificata, il tecnico pompa l'acqua fuori dal cilindro. Sulla base del principio di Pascal, si realizza una distribuzione uniforme della pressione sull'area del moncone, grazie a cui si ottiene la forma del modello negativo. Ciò è estremamente difficile da ottenere con una tecnica totalmente manuale. Il sistema supporta quindi il tecnico ortopedico nel definire volumi e forma dell'invasatura.



Figura 2.3: Dispositivo di controllo Aqua System, consente di regolare la pressione ottimale nel cilindro. [8]



Figura 2.4: Cilindri Aqua System, forniscono sul gesso un'impressione precisa e dettagliata dei tessuti molli e delle strutture ossee dell'arto amputato mentre è sottoposto a pieno carico. (A destra TT, a sinistra TF) [8]

2.1.3 Scanner 3D per la digitalizzazione dei modelli

A fianco delle tradizionali metodologie per la realizzazione di protesi vi è il supporto delle tecniche CAD-CAM.¹

La ricostruzione della superficie interna dell'invasatura è eseguita mediante scansione tridimensionale, successivamente elaborata tramite software CAD. La scansione è il metodo principale per la realizzazione dei modelli virtuali: viene creata una "nuvola di punti" che viene interpretata dal software ed elaborata ottenendo una riproduzione digitale del modello scansionato. Una volta ottenuto il modello in formato STL si effettua la valutazione del volume e della forma dell'invasatura con successiva rettifica CAD.

¹Impiego congiunto e integrato di sistemi software per la progettazione assistita da computer (Computer-Aided Design, CAD) e fabbricazione assistita dal computer (Computer-Aided Manufacturing, CAM).

La fase CAM consiste nella fresatura del modello positivo da un blocco di poliuretano espanso, il quale viene montato su un apposito sostegno metallico e lavorato tramite un robot meccanico a 7 assi ORTIS (fig. 2.5 *Roboticom*, Pisa).



Figura 2.5: Roboticom, Ortis.



Figura 2.6: Scanner da interni e portatile.

L'utilizzo di tale robot permette di visualizzare l'intero processo in maniera virtuale e di effettuare modifiche complete non realizzabili a banco da frese a 2 assi.

Durante le fasi del processo di acquisizione della forma e di lavorazione vengono eseguite le scansioni, utilizzate per valutare come la forma acquisita cambia nel tempo. In particolare, vengono eseguite le scansioni di: calchi iniziali, modelli positivi pre e post rettifica e invasatura di controllo. A tal scopo vengono utilizzate due tipologie di scanner (fig. 2.6):

- Lo **scanner meccanico laser da interni** Echo Digitizer (*Rodin 4D*, Francia), è dotato di un laser rotativo su due assi che permette di digitalizzare l'estremità delle forme [9];
- Lo **scanner portatile da esterno a luce strutturata** EinScan-Pro HD 3D (*Shining 3D Tech.*, Cina). Tale tipologia di scanner si basa sulla proiezione di un pattern luminoso sulla scena, il quale viene distorto a causa della superficie dell'oggetto [10]. Da tale distorsione si acquisiscono informazioni sulla conformazione e topologia della superficie colpita.

2.2 Scopo dello studio

Il dolore è estremamente comune nelle persone con amputazione di arto inferiore ed è spesso correlato a problemi di abrasioni cutanee e di pressione tra invasatura e moncone. Esso si verifica nel 61-76% dei casi ed è diretta conseguenza di minor accettazione della protesi [7]. Ottenere un'invasatura comoda che non contribuisca al dolore all'arto residuo è complesso poiché il moncone è dinamico nella forma e nel volume. La qualità dell'invasatura è influenzata dai processi di fabbricazione, compreso l'ottenimento della forma residua degli arti, la rettifica del modello positivo e l'allineamento della protesi, così come la selezione del materiale dell'interfaccia. Si ipotizza che la tecnica idrostatica, grazie al principio su cui si basa (sezione 2.1.2), permetta di migliorare la coerenza dell'acquisizione della forma e di ridurre l'entità della stilizzazione del modello positivo e i tempi di consegna.

Sulla base dei vantaggi potenziali dalla tecnica idrostatica l'ipotesi complessiva dello studio scientifico è che tale tecnica sia un'alternativa più ripetibile ed efficiente rispetto alla tecnica manuale per realizzare il calco del moncone del paziente, e che permetta di ottenere un'invasatura immediatamente più confortevole per il paziente (*outcome* primario). Altri *outcome* secondari si basano sulla documentazione dei processi di confronto delle forme: essi sono importanti per migliorare la comprensione di come l'*outcome* primario si ottenga.

Per valutare questa ipotesi complessiva, gli obiettivi specifici dello studio sono: [7]

- *Obiettivo 1:* comparare la ripetibilità dei risultati (*consistency*) di due tecniche per la realizzazione del calco del moncone per quanto attiene la forma del modello in persone con amputazione transtibiale e transfemorale;
- *Obiettivo 2:* comparare l'efficienza (*efficiency*) di due tecniche per la realizzazione del calco del moncone in persone con amputazione transtibiale e transfemorale;
- *Obiettivo 3:* comparare gli esiti (*outcomes*) di due tecniche per la realizzazione del calco del moncone in persone con amputazione transtibiale e transfemorale.

In funzione degli esiti, i risultati dello studio potrebbero supportare l'adozione della tecnica idrostatica su più larga scala, sia presso le strutture del Veterans Affairs, le strutture per il trattamento dei militari in servizio, sia presso le sedi del Centro Protesi INAIL, a beneficio di invalidi civili e degli infortunati sul lavoro.

2.3 Disegno dello studio

Lo studio segue il processo clinico di fabbricazione dell'invasatura in modo tale che ogni fase sia mirata ad affrontare un determinato obiettivo della ricerca. I soggetti coinvolti sono stati reclutati tra i pazienti già utilizzatori di protesi che accedono al centro protesi INAIL.

Tali pazienti devono rispettare alcuni criteri di inclusione: [7]

- Essere maggiorenni;
- Avere un'amputazione monolaterale transtibiale o transfemorale;
- Avere sufficiente sensibilità all'arto residuo;
- Non avere neuromi superficiali dolorosi o ferite aperte sul moncone;
- Essere in grado di stare in piedi per i 4-6 minuti necessari alla presa misura;
- Avere l'arto residuo sufficientemente stabile in volume (amputazione avvenuta da almeno un anno);
- Nel caso di transfemorali, non avere allergie al silicone o una lunghezza del moncone inferiore a 12 cm;
- Non avere una circonferenza del moncone o un peso non supportati dal Symphonie Aqua System.

Parteciperanno alla ricerca 20 pazienti con amputazione transtibiale e 10 pazienti con amputazione transfemorale. Ciò avverrà nell'arco di 2 anni (sui 3 di progetto) e lo stesso avverrà in ciascuno dei due centri negli Stati Uniti. Questa numerosità campionaria eccede di almeno il 10% il minimo stabilito dall'analisi della potenza statistica [7].

La prima visita è di arruolamento (fig. 2.7, *Visit 1*): i pazienti sono valutati e vengono raccolte informazioni su età, genere, peso, altezza, eventuali problemi di diabete, eziologia e tempo dall'amputazione, mobilità funzionale, lunghezza e tipo di tessuto dell'arto residuo, sensibilità o aree dolorose e tipo di progettazione dell'invasatura e della protesi in uso.

Ciascun paziente viene successivamente chiamato per ulteriori due visite: una per la realizzazione dei calchi del moncone (seconda visita) e l'altra per la prova ed eventuale l'adattamento delle invasature (terza visita).

Durante la seconda visita (fig. 2.7, *Visit 2*) viene realizzato un modello negativo da parte di due differenti tecnici ortopedici (di cui uno identificato come protesista principale), sia con la tecnica idrostatica che con la tecnica manuale. I protesisti coinvolti nello studio devono avere tra i 6 e i 30 anni di esperienza nel trattamento

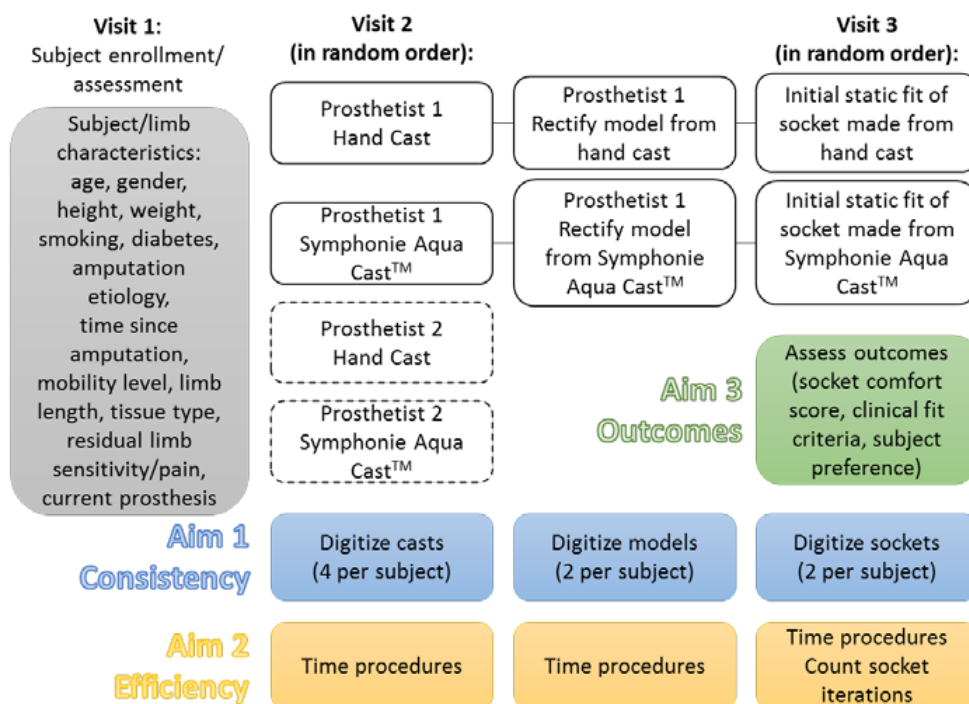


Figura 2.7: Fasi dello studio e obiettivi da soddisfare. [7]

di pazienti con amputazione di arto inferiore. La maggior parte degli operatori possiede già una vasta esperienza con la tecnica tradizionale, mentre per la tecnica idrostatica i livelli di esperienza sono vari; lo studio prevede pertanto anche una fase di formazione. Si realizzano, quindi, 4 modelli negativi per ciascun paziente, in modo da poter valutare l'affidabilità inter-protésista (*inter-rater reliability*). Solo il tecnico ortopedico leader procede alla stilizzazione dei propri modelli positivi per poi realizzare due invasature di prova che vengono testate sul paziente durante la terza visita.

Durante la terza visita (fig. 2.7, *Visit 3*) vengono completate le valutazioni di outcome per comfort (outcome primario) e adeguatezza dell'invasatura (*socket fit*), procedendo quindi con eventuali modifiche. Ciò può comportare il rifacimento dell'invasatura. Le modifiche apportate e il loro numero, così come il tempo impiegato, vengono annotati. Una volta raggiunto un adattamento soddisfacente per paziente e TO, viene ripetuta la valutazione del comfort. L'outcome primario di interesse per lo studio è la valutazione del comfort alla prima prova di ciascuna invasatura.

Il modello risultante da ogni fase di lavorazione viene digitalizzato. Le scansioni ottenute corrispondono a:

- 4 calchi negativi;
- 4 modelli positivi grezzi;
- 2 modelli positivi stilizzati;
- 2 invasature di prova.

La scansione dei modelli e il calcolo delle mappe distanza sono funzionali al raggiungimento dell'*Obiettivo 1*. Il monitoraggio dei tempi e dei rifacimenti sono funzionali al raggiungimento dell'*Obiettivo 2*. La valutazione del comfort, adeguatezza e soddisfazione sono funzionali al raggiungimento dell'*Obiettivo 3* [7].

2.3.1 Reperi anatomici per comparazioni inter-tecnica, inter-operatore e inter-soggetto

Per facilitare l'elaborazione dei dati, sono stati definiti nel Manuale delle Procedure (MOP [7]) del progetto dei punti di riferimento (*landmark*) per pazienti TT (fig. 2.8) e TF (fig. 2.9). Tali punti coincidono con punti di repere anatomici e punti di interesse definiti matematicamente in base alle misure del moncone.



Figura 2.8: Riferimenti utilizzati per i punti d'interesse e assegnazione associata dei landmark anatomici per amputazioni transtibiali. Fonte: Case Report Form del progetto.

Sulla base di essi, durante la seconda visita vengono individuati i landmark tramite palpazione sul paziente e posti sul liner dei marcatori. I marcatori utilizzati possono essere di varie tipologie (fig. 2.10), in base alla fase di applicazione. Di norma sono colorati e in rilievo per poter essere identificati dallo scanner e impressi nel calco gessato facilitando le successive fasi di scansione e riposizionamento dei landmark.

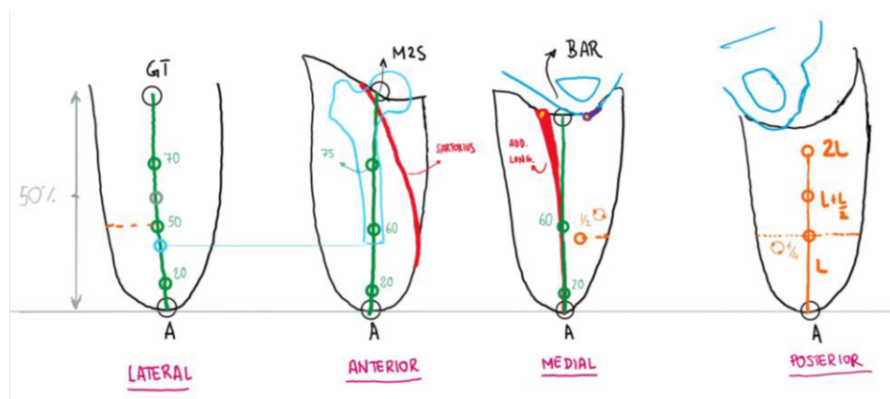


Figura 2.9: Riferimenti utilizzati per i punti d'interesse per amputazioni transfemorali e assegnazione associata. Fonte: Case Report Form del progetto.

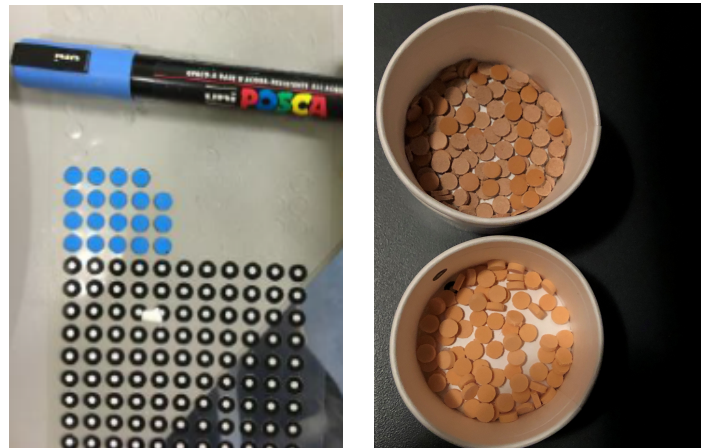


Figura 2.10: Diverse tipologie di marcatori utilizzati: a sinistra marcatori per EinScan colorati con pennarello acrilico opaco; a destra marcatori in *pelite* e *pelle* con diametro 7 mm e spessore 2 mm.

Una volta acquisite le scansioni, ci sono vari potenziali approcci per quantificare e analizzare le differenze di forma e volume derivanti dal confronto di esse. Ad esempio, l'uso di mappe colore insieme a istogrammi permettono di dettagliare il *match* tra le forme catturate in momenti diversi, permettendo di mappare le differenze attraverso le fasi di lavorazione. Tale metodo è anche utile a confrontare il risultato di diversi tecnici allo stesso step del processo di produzione.

Nel seguito saranno descritti i metodi e gli strumenti identificati come adeguati a supporto dei processi di analisi dei dati.

Capitolo 3

Materiali e metodi

3.1 Flusso di lavoro

Il flusso di elaborazione dati necessario ad ottenere i risultati fondamentali per il test delle ipotesi dello studio è riportato in Figura 3.1. Le prime fasi avvengono in maniera sequenziale, per poi proseguire con il calcolo di parametri dimensionali ed infine giungere ad una fase finale che raggruppa e sintetizza più dati, ottenendone informazioni d'aggregazione.

Il flusso di lavoro può essere diviso nelle seguenti fasi:

- **Acquisizione dati:** i dati vengono acquisiti tramite i metodi definiti in Sezione 2.1.3. Sono dati grezzi e non confrontabili dal momento che dipendono dal sistema d'acquisizione usati per generarli.
- **Pre-processing dati:** i dati grezzi sono sottoposti a operazioni di pulizia ed elaborazione per prepararli alle fasi successive;
- **Selezione landmark:** sulle scansioni vengono identificati e resi disponibili per le fasi di analisi successive dei landmark basati sull'anatomia del paziente. Saranno utili per avere dei punti di riferimento fissi, sui quali basare algoritmi di allineamento, registrazione e calcolo di parametri quantitativi.
- **Allineamento:** una volta identificati i punti di repere e le equazioni necessarie per la creazione della terna di riferimento, l'allineamento viene eseguito tramite l'applicazione di matrici di roto-traslazione che permettono di orientare in un sistema di riferimento (SdR) globale tutti i dati secondo una logica comune. Ciò consente il confronto intraprocesso e intrapaziente e il proseguimento del flusso di lavoro.
- **Registrazione:** le mesh sono sottoposte ad algoritmi che cercano di allinearle l'una all'altra nel modo migliore possibile. È un'operazione fondamentale

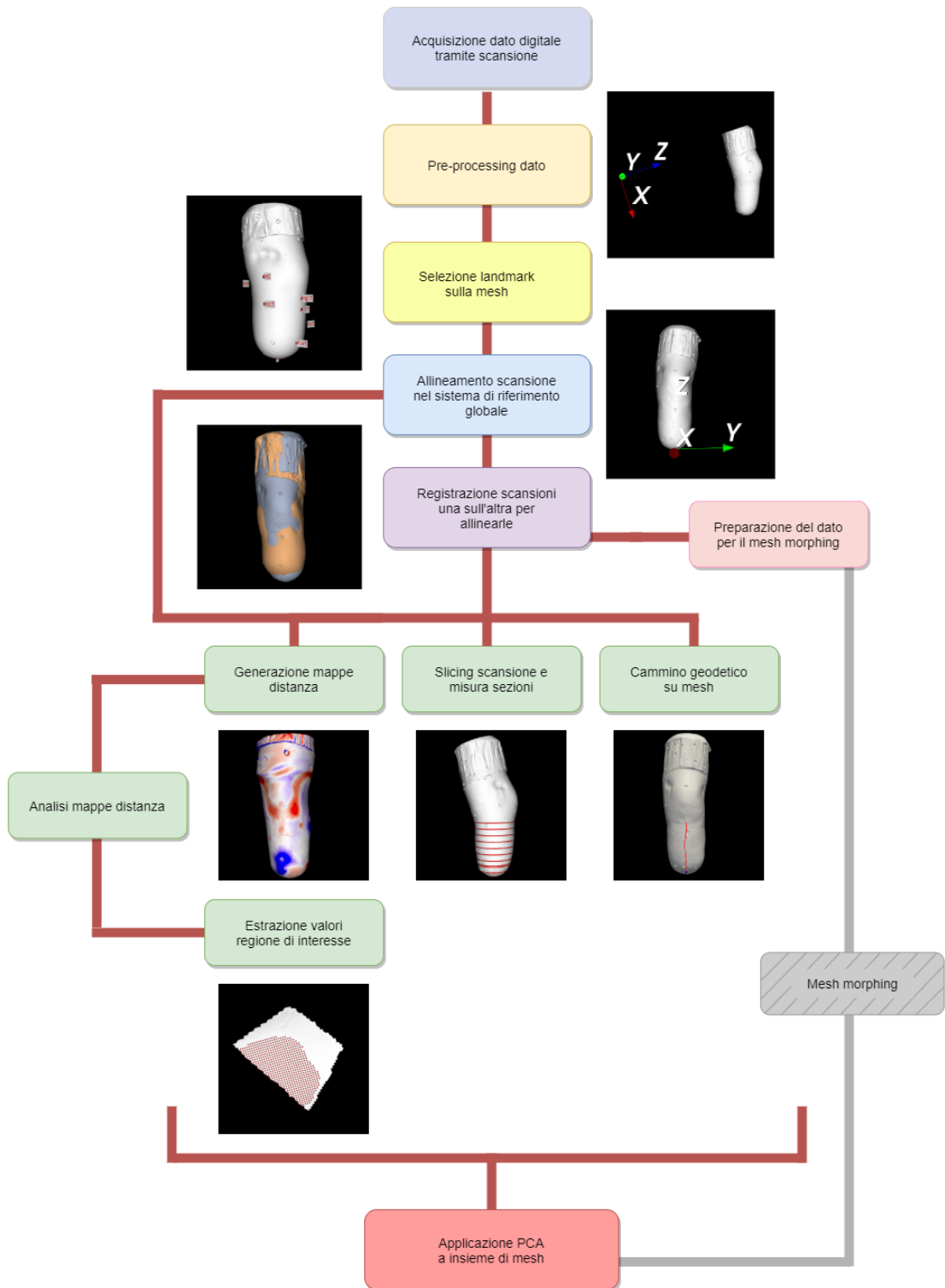


Figura 3.1: Flusso di lavoro.

che prepara le scansioni alle successive fasi di calcolo di parametri e risultati che scaturiscono dal confronto.

- **Preparazione dato per il mesh morphing:** i dati vengono preparati per operazioni di *morphing*, ovvero la modellazione di una superficie su un'altra preservandone la topologia. Vengono confrontate due mesh e calcolate le differenze tra landmark omologhi. Ciò viene incapsulato in un formato file specifico del software di morphing.
- **Generazione *outcome*:**
 - **Mappe distanza:** una volta registrate le scansioni, vengono calcolati dei valori di qualità (in base a varie metriche), mantenendo una mesh come riferimento. Da tali valori vengono poi generate delle mappe colore, grazie a cui analizzare la distribuzione delle differenze sulla superficie della mesh.
 - **Sezionamento:** dalle mesh sono ottenute delle sezioni, tramite il “taglio” su un piano perpendicolare all'asse longitudinale. Sulla base di queste sezioni sono calcolate delle misure, come perimetro e area, usate come metrica di confronto tra scansioni.
 - **Cammino geodetico:** viene individuato sulle mesh il percorso più breve che unisce due punti noti e calcolata la lunghezza di esso. Ciò è utile a confrontare tra loro le lunghezze e trarre evidenze circa le misure prese dal reale e digitalmente. Inoltre, è possibile individuare punti di interesse che si trovino a una certa porzione del cammino geodetico creato.
 - **Estrazione regione di interesse:** talvolta può essere utile analizzare una specifica regione individuata sulla scansione. Ciò viene fatto di norma a partire dalla mappa distanza, per fare analisi mirate sulle regioni con concentrazioni di valori che necessitano di un approfondimento.
- **Applicazione PCA:** una volta raccolto un dataset di dati è possibile analizzarlo sfruttando la *Principal Component Analysis*. Per eseguire tale operazione è necessario riportare la stessa topologia su tutte le mesh, risultato ottenibile tramite il mesh morphing (blocco esterno alle attività del lavoro attuale). Si può quindi estrarre un indicatore statistico riassuntivo delle differenze.

Nelle sezioni seguenti si descriveranno più in dettaglio le singole fasi, alla luce di come ogni problema è stato risolto e tramite quali algoritmi o soluzioni elaborate.

3.1.1 Acquisizione dati

Acquisire il dato significa ottenere una rappresentazione digitale dell'oggetto di interesse. Gli strumenti utilizzati a questo proposito sono scanner laser, il cui funzionamento si basa su una tecnica di scansioni successive che permettono di rilevare un oggetto tridimensionale mediante una nuvola di punti e la loro connessione triangolare. La posizione di ciascuno dei punti viene poi individuata secondo un sistema di coordinate spaziali (x, y, z) per essere visualizzata su di un monitor come una fotografia tridimensionale, risultante dall'unione di questi elementi.

I formati dei dati generati sono principalmente due:

- **STL**: *STereo Lithography interface format* o acronimo di *Standard Triangulation Language* o *Standard Tassellation Language*. È un formato file, binario o ascii, diventato il formato standard di trasmissione dei dati di tutti i tipi di sistemi di prototipazione rapida, usato dai software di stereolitografia CAD [11].

Un file STL rappresenta un oggetto la cui superficie è stata discretizzata in triangoli tramite algoritmi di tassellazione (fig. 3.2). Ogni triangolo è descritto dalle coordinate (x, y, z) dei tre punti che lo formano e da un vettore normale che punta verso l'esterno della superficie (fig. 3.3) [11]. Il contenuto di un file STL è un set di triangoli orientati, ognuno con proprie coordinate dei vertici e normale alla faccia. Ogni vertice è riportato pertanto nel file quanti sono i triangoli che lo contengono. Questo formato presenta alcuni vantaggi, tra cui la semplicità di generazione e di processamento, mentre a suo sfavore genera una geometria approssimata e una struttura dati che può presentare la ripetizione dello stesso vertice più volte. Di seguito è riportata la struttura base del file in codifica ascii, nel caso di un triangolo:

```

solid name
facet normal ni nj nk
  outer loop
    vertex v1x v1y v1z
    vertex v2x v2y v2z
    vertex v3x v3y v3z
  endloop
endfacet
endsolid name

```

- **PLY**: il *Polygon File Format* (o *Stanford Triangle Format*) è un formato creato per memorizzare dati 3D in modo flessibile e portabile [13]. Gli oggetti

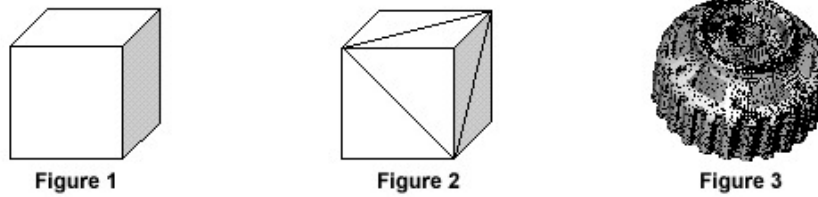


Figura 3.2: Esempio di triangolazione per file STL: per solidi semplici sono necessari pochi triangoli, per superfici più complesse ne servono molti di più. [12]

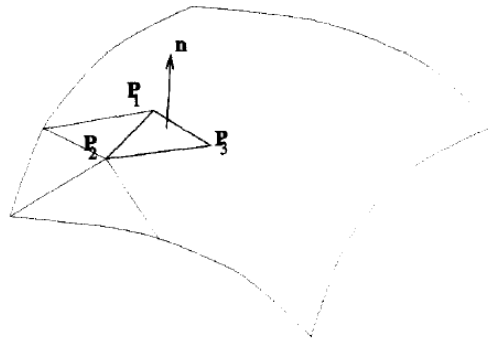


Figura 3.3: Faccia in formato STL. [11]

sono tradotti e salvati come un insieme di poligoni piani. I file possono essere in formato binario o *ascii*. In formato binario sono incluse informazioni che rendono i file indipendenti dalla macchina: sono specificati i tipi utilizzati per ogni informazione memorizzata, il numero di byte per tipo e la tipologia di codifica (*big* o *little endian*) [13]. È possibile anche estendere il formato con tipi specificati dall'utente. Si possono salvare quindi anche un insieme di informazioni specifiche del dato di interesse, come colore, trasparenza, normali alla superficie e valori di qualità associati ai vertici. La struttura dei file è composta da un *header*, che specifica gli elementi della mesh, il loro numero e tipo, seguito da una lista di tali elementi. Un esempio di file PLY, che descrive un unico triangolo, è il seguente:

```
ply format ascii 1.0
element vertex 3
property float x
property float y
property float z
element face 1
property list uchar int vertex_indices
end_header
```

```

v1x v1y v1z
v2x v2y v2z
v3x v3y v3z
3 0 1 2

```

L'ultima riga indica gli indici dei vertici che formano una faccia della mesh. Il vantaggio rispetto al formato STL è pertanto l'essere più sintetico ed evitare di ripetere più volte lo stesso vertice, riferendosi ad esso tramite l'indice.

3.1.2 Pre-processing dato

Una volta acquisiti i dati vengono effettuate operazioni di *pre-processing*, come l'eliminazione di aree rilevate dalla scansione ma non desiderate e l'eliminazione di vertici ripetuti, che possono essere eseguite in qualsiasi software CAD.

La mesh può essere poi sottoposta a decimazione, specificando la percentuale di riduzione. Tale operazione appartiene alle funzionalità offerte dalla libreria sviluppata. La riduzione di vertici e facce permette di proseguire con le fasi di elaborazione del dato riducendo il tempo di computazione. Data infatti l'alta risoluzione degli scanner si potrebbero ottenere scansioni con milioni di punti, che comporterebbero tempi di elaborazione eccessivi.

3.1.3 Selezione landmark

L'elaborazione della scansione prende inizio con l'identificazione dei landmark sulla superficie digitalizzata e l'assegnazione della nomenclatura secondo riferimento prestabilito (sezione 2.3.1). È importante effettuare tale operazione all'inizio dell'elaborazione e rendere tali punti disponibili anche in seguito, salvandoli in file di formato prestabilito. In questo modo, i landmark potranno essere riutilizzati in tutte le fasi di necessità del flusso globale definito Figura 3.1 così da non introdurre l'errore di ri-selezione dello stesso punto per la stessa mesh.

Il salvataggio dei punti avviene attraverso il loro *point id* e non tramite le coordinate. Questo permette di poter rimanere indipendenti dalle trasformazioni che saranno applicate alla mesh: il *point id* rimane invariato ed è possibile da esso risalire alle coordinate.

3.1.4 Allineamento

Una volta acquisite le scansioni si è reso necessario allinearle secondo uno stesso SdR. Gli strumenti di acquisizione generano infatti delle rappresentazioni che mappano le coordinate x, y, z dei punti scansionati nel SdR dell'emettitore. La

fase di allineamento è quindi atta a portare la mesh in un SdR globale comune, definito rispettivamente per TT e TF. Ciò viene effettuato applicando a tutti i punti una trasformazione rigida.

Per eseguire tale elaborazione è necessario stabilire le assunzioni generali per la definizione dei tre assi di riferimento:

1. Asse longitudinale → diretto dalla parte distale alla parte prossimale della scansione;
2. Asse antero-posteriore → diretto dalla parte posteriore alla parte anteriore;
3. Asse medio-laterale → diretto dalla parte sinistra alla parte destra.

A tal proposito sono stati ideati vari metodi di allineamento sia per le mesh di amputati TT che TF, basati sia su proprietà geometriche delle superfici che su punti di repere anatomici. Prerequisito fondamentale per la creazione di tale SdR è l'identificazione dei landmark anatomici, avvenuta durante la seconda visita del paziente (sezione 2.3.1).

Calcolo asse principale d'inerzia

Data la forma degli oggetti da allineare, che si sviluppa principalmente lungo l'asse longitudinale, si è pensato di poter sfruttare tale proprietà geometrica per definire da subito tale asse e derivare successivamente gli altri. Per ottenere le coordinate di tale asse si è sfruttata la PCA (*Principal Component Analysis*) [14]. La PCA permette infatti di trovare le direzioni della massima varianza nei dati ad alta dimensionalità. Per il caso d'uso specifico è bastato estrarre la prima componente, ovvero quella lungo la quale si individua la massima varianza dei dati, ed utilizzare tale direzione come asse longitudinale.

Nonostante le ipotesi corrette si è osservato che l'asse individuato tramite questo metodo era talvolta inadeguato, in quanto divergeva da quello che doveva essere l'asse longitudinale atteso. Questo problema è stato ricondotto al fatto che le superfici d'analisi talvolta tendevano ad una forma sferica e presentavano una distribuzione di punti non omogenea sulla superficie. Mentre l'imprevedibilità della PCA applicata a superfici sferiche non è un problema risolvibile, il fatto di avere spazialità non uniforme tra i punti, zone a diversa densità o punte fuori asse principale è risolvibile. Per avere un dataset d'applicazione della PCA meno sensibile a tale errore è stato elaborato un metodo "robusto" di calcolo. Esso consiste nel generare un nuovo dataset di punti, ottenuti come distribuzione uniforme interna al volume dell'oggetto.

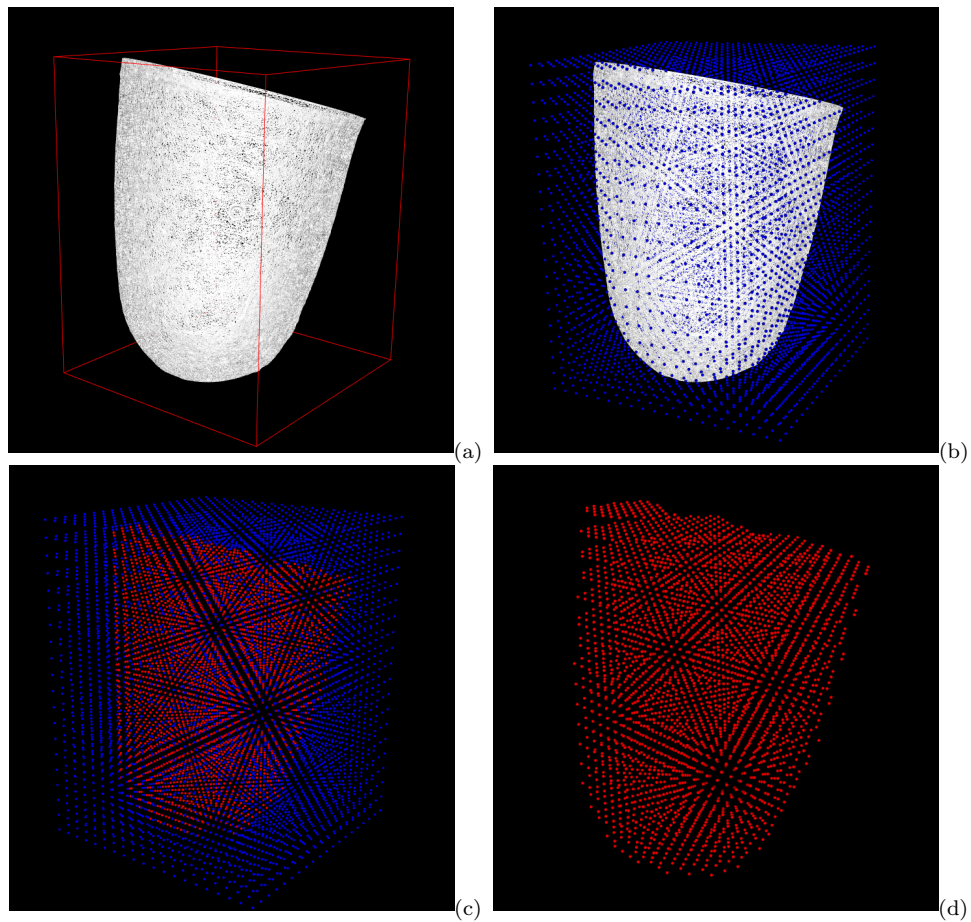


Figura 3.4: Rappresentazione delle fasi di generazione dell'asse d'inerzia con metodo "robusto".

Il procedimento applicato è il seguente:

1. Estrarre il *minimum bounding box* della scansione, ovvero il riquadro di delimitazione minimo in cui è contenuta la mesh (fig. 3.4.a);
2. Generare un insieme di punti uniformemente distribuiti nel volume della bounding box, ottenendo una sorta di griglia molto fitta (fig. 3.4.b);
3. Filtrare la nuvola di punti ottenuta mantenendo solo quelli interni al volume della scansione (fig. 3.4.c e fig. 3.4.d);
4. Applicare a tale dataset la PCA.

Best Fitting di un piano tramite regressione della distanza ortogonale

Tra i passaggi di creazione della terna ve ne sono alcuni che, dati un insieme di punti, determinano il piano che meglio li approssima e di conseguenza l'asse perpendicolare a tale piano. L'identificazione del piano viene effettuata tramite un algoritmo basato sulla decomposizione ai valori singolari (SVD). Il risultato desiderato è il piano ottimo che approssima un insieme di n punti nello spazio 3D, ottenuto minimizzando la somma dei quadrati delle distanze ortogonali tra il piano ed i punti [15].

La scomposizione ai valori singolari di una matrice $A \in \mathbb{R}^{n \times m}$ è la fattorizzazione di A in prodotto di tre matrici:

$$A = USV^T$$

dove le colonne di $U \in \mathbb{R}^{n \times m}$ e $V \in \mathbb{R}^{n \times m}$ sono ortonormali e la matrice $S \in \mathbb{R}^{n \times m}$ è una matrice contenente i valori singolari $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r \geq 0$, con $r = \min(m, n)$. La posizione del piano viene rappresentata dal punto c appartenente al piano e il vettore unitario n è la normale al piano e ne determina la direzione. Di conseguenza la *distanza ortogonale* tra un punto p_i e il piano è $(p_i - c)^T n$. Il piano si ottiene risolvendo:

$$\min_{c, \|n\|=1} \sum_{i=1}^n (p_i - c)^T n)^2$$

L'algoritmo per l'ottenimento del piano utilizzando la regressione della distanza ortogonale e l'SVD consiste in: [15]

1. $c = \frac{1}{n} \sum_{i=1}^n p_i$
2. $A = [p_1 - c, \dots, p_n - c]$
3. $USV^T = A$, *Computazione della singola decomposizione di A*
4. $n = U(:, 3)$, *i.e., la normale è data dalla terza colonna di U .*

Poiché la normale è data da $U(:, 3)$, segue dall'ortogonalità di U che il piano è attraversato dalle prime due colonne di U . Pertanto l'asse perpendicolare al piano coinciderà con n .

Procedimento

È necessario un metodo di allineamento clinicamente significativo e ripetibile tra i soggetti. Il raggiungimento di tale obiettivo si basa su passaggi comuni che differiscono principalmente per il metodo di creazione dell'asse z e per l'insieme dei landmark utilizzati per la definizione dell'asse temporaneo preparatorio all'identificazione degli assi x e y . I principali passaggi consistono in:

1. Identificazione dell'asse longitudinale, attraverso l'utilizzo di landmark anatomici o l'asse principale d'inerzia;
2. Sezione della scansione in corrispondenza di un punto di repere anatomico di taglio sul piano perpendicolare all'asse definito al punto 1. Tale operazione ha lo scopo di ottenere la parte di scansione "di interesse" per l'allineamento, escludendo i punti appartenenti al prolungamento dell'invasatura. Essi potrebbero deviare la direzione dell'asse longitudinale se calcolato come asse principale di inerzia;
3. Eventuale ricalcolo dell'asse longitudinale come asse di inerzia (calcolo PCA "robusto" sezione 3.1.4);
4. Verifica orientazione dell'asse desiderata (assunzione SdR Voce 1), nel caso di asse longitudinale inerziale, tramite la definizione di un asse anatomico con direzione nota e il controllo della concordanza fra tale asse e l'asse principale d'inerzia;
5. Identificazione dell'asse temporaneo (assegnato momentaneamente x), attraverso l'utilizzo della decomposizione ai valori singolari;
6. Creazione della terna ortogonale di riferimento tramite prodotto vettoriale di asse temporaneo e asse longitudinale;
7. Verifica assegnazione corretta di assi x e y tramite confronto con l'asse antero-posteriore ed eventuale inversione di tali assi;
8. Verifica orientazione assi x e y (assunzione SdR Voce 2);
9. Applicazione della rotazione per allineare la mesh nel SdR globale;
10. Traslazione della mesh per posizionare un determinato punto appartenente alla superficie sull'origine del SdR globale.

3.1.5 Registrazione

La **registrazione** di immagini bi o tri dimensionali è il processo che permette di trovare la trasformazione spaziale che ottimizzi la sovrapposizione di due immagini, trovando il *mapping* (corrispondenza) tra i punti dell'immagine presa come riferimento su punti omologhi appartenenti all'altra immagine (Figura 3.5) [16]. Ciò permette di evidenziare ogni possibile cambiamento in dimensioni, forma o posizione.

In ambito biomedico la registrazione sta acquisendo sempre più importanza: dall'allineamento delle immagini si possono ricavare informazioni utili per quanto

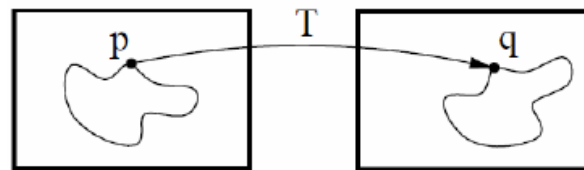


Figura 3.5: Punti omologhi su cui si basa la registrazione. [16]

riguarda la diagnosi, la pianificazione chirurgica e la guida al trattamento e al monitoraggio delle malattie [16]. Nel nostro caso, la registrazione viene applicata alle scansioni per eseguire comparazioni inter e intra paziente e per valutare il modificarsi dell'invasatura nelle varie fasi di lavorazione.

Le principali componenti della registrazione sono le seguenti (Figura 3.6):

- Immagine o superficie *source*, *static* o *reference*, che non subisce trasformazioni ma che viene mantenuta come riferimento;
- Immagine o superficie *target* o *moving*, che deve essere sottoposta a una trasformazione per adattarsi alla static;
- Una metrica tramite cui si misura il *match* tra la source e la target trasformata: dà una misura della bontà dell'allineamento;
- Un ottimizzatore, ovvero un metodo di ottimizzazione che cerca la soluzione migliore (basandosi sulla metrica definita) nello spazio di ricerca definito dai parametri della trasformazione;
- Una trasformazione, che prende come input i parametri ottimi ricavati dall'ottimizzatore e li applica all'immagine moving, modificandola.

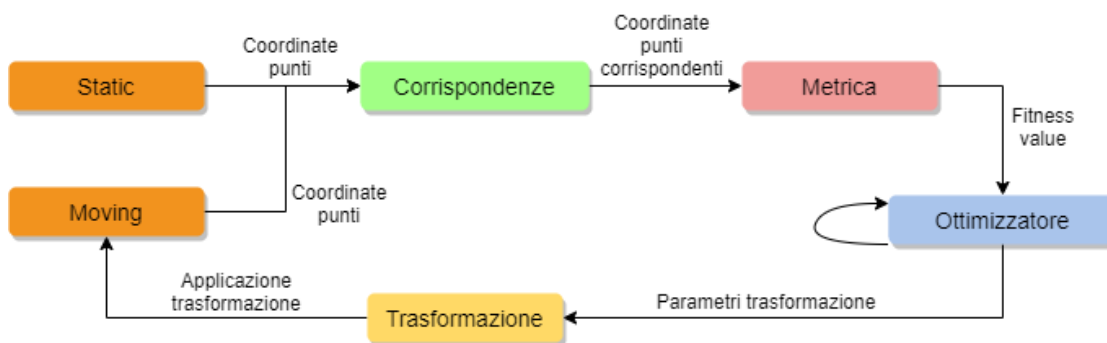


Figura 3.6: Schema generico del flusso di registrazione.

Il processo di registrazione può essere di varie tipologie, che si differenziano per il numero di gradi di libertà imposti:

- **Rigida:** sono possibili solo traslazioni e rotazioni: gli oggetti non cambiano forma e la distanza tra due punti viene preservata dopo la trasformazione;
- **Similare:** le trasformazioni che è possibile applicare sono traslazioni, rotazioni e *scaling*;
- **Affine:** il modello affine include trasformazioni che mantengano il parallelismo tra rette, senza necessariamente preservare angoli e distanze. Appartengono a questa classe rototraslazioni, riflessioni, trasformazioni di *shear* e omotetie ad esempio;
- **Deformabile:** creazione di un mapping di linee e curve; equivale a deformare una superficie per adattarsi ad un'altra. È la trasformazione che permette il maggior numero di gradi di libertà, poiché rende possibile il cambiamento della forma dell'oggetto al fine di massimizzare la similarità con la superficie di riferimento.

Una volta identificato il problema, sono stati analizzati alcuni dei principali algoritmi di registrazione presenti in letteratura.

Registrazione basata su landmark

Un metodo di allineamento a cui spesso si fa riferimento è quello basato sui landmark. Esso consiste nel trovare la trasformazione che meglio approssima la sovrapposizione di due set di punti in corrispondenza biunivoca.

Assumiamo quindi di avere n landmark su un corpo rigido, dove $\{x_1, \dots, x_n\}$ sono le posizioni di tali punti prima della trasformazione e $\{y_1, \dots, y_n\}$ le posizioni dopo la trasformazione. Ciò che si vuole determinare sono la matrice di rotazione R e il vettore di traslazione t che permettono di mappare i punti x_i sui punti y_i per $i = 1, \dots, n$. A causa degli errori di misura insiti nell'identificazione dei landmark, il mapping non è esatto ma il problema può essere risolto come problema ai minimi quadrati: [15]

$$\min_{R \in \Omega, t} \sum_{i=1}^n \|Rx_i + t - y_i\|^2$$

dove $\Omega = \{R | R^T R = R R^T = I_3; \det(R) = 1\}$ è il set di matrici di rotazione. Dati:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i, \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$$

si possono definire le matrici $A = [x_1 - \bar{x}, \dots, x_n - \bar{x}]$ e $B = [y_1 - \bar{y}, \dots, y_n - \bar{y}]$. Il problema di determinazione della matrice di rotazione diventa

$$\min_{R \in \Omega} \|RA - B\|_F$$

dove la norma di Frobenius di una matrice Z è definita come $\|Z\|_F^2 = \sum_{i,j} z_{i,j}^2$. Il problema di ottimizzazione può essere quindi risolto sfruttando SVD:

$$C = BA^T = USV^T$$

$$R = U \text{diag}(1, 1, \det(UV^T)) V^T$$

$$t = \bar{y} - R\bar{x}$$

Si ottengono in questo modo i parametri di rotazione e traslazione da imporre all'oggetto per allinearsi al riferimento [15].

Questo metodo è particolarmente utile per registrare oggetti di cui si hanno a disposizione molti landmark in corrispondenza, poiché con alto numero di essi l'errore di localizzazione peserà meno sul risultato dell'allineamento, grazie a effetti di compensazione.

Iterative Closest Point (ICP)

L'algoritmo *Iterative Closest Point* fu proposto nel 1992 da Besl e McKay [17] come metodo di registrazione di forme 3D. In origine non era stato pensato per l'applicazione a immagini e dati medici, ma ottenne poi grande successo proprio in questo campo, divenendo l'algoritmo di corrispondenza di superfici più utilizzato nell'*imaging* medico [17].

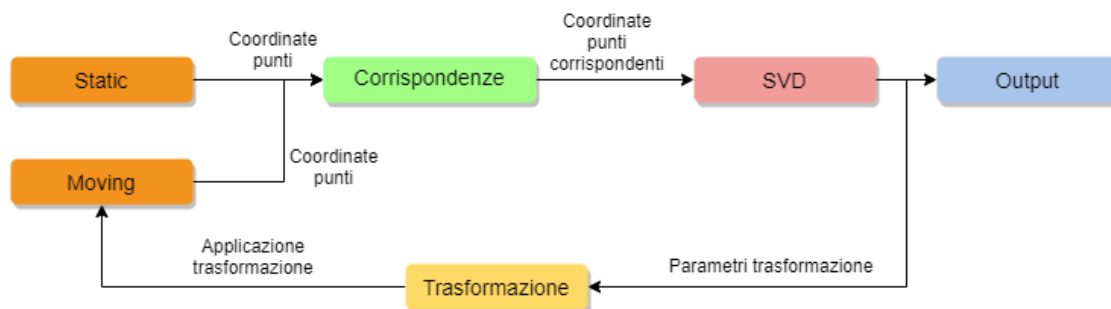


Figura 3.7: Flusso di operazioni algoritmo ICP.

L'algoritmo è iterativo e prevede i seguenti passi (Figura 3.7): [18]

1. Ricerca delle **corrispondenze** tra i punti appartenenti alla nuvola di punti moving (q_i) e quelli della static (p_j) tramite approcci *Nearest Neighbor*

(NN, che mirano a identificare il punto più vicino a uno dato secondo una determinata metrica, come la distanza euclidea):

$$\hat{i} = \operatorname{argmin}_i \|p_j - q_i\|^2$$

dove $i \in [0, 1, \dots, N]$ ed N è il numero di punti della superficie target.

2. Applicazione di **SVD** per trovare una stima della matrice di trasformazione da applicare per ottenere l'allineamento. SVD è infatti un modo affidabile per risolvere il problema lineare dei minimi quadrati, ovvero la minimizzazione della somma delle distanze tra punti omologhi. Data quindi M la matrice delle *cross-correlation* tra le due nuvole di punti in corrispondenza possiamo ottenere la decomposizione ai valori singolari come

$$M = USV^T$$

La soluzione ottimale al problema dei minimi quadrati definisce la matrice di rotazione R e di traslazione t come:

$$R = UV^T$$

$$t = c_s - Rc_t$$

dove c_s e c_t sono i rispettivamente i centroidi della source e della target.

3. Applicazione dei parametri di trasformazione ottenuti alla nuvola di punti target;
4. Iterazione sui passi precedenti ricalcolando le corrispondenze ed eventualmente rimuovendo punti *outlier* fino a convergenza o al raggiungimento di un massimo di iterazioni. L'output è la trasformazione globale, ovvero che porta la superficie originale ad allinearsi alla static (Figura 3.8) secondo il criterio prefissato:

$$\hat{R}, \hat{t} = \operatorname{argmin}_{R,t} \sum_{i=1}^N \|(Rq_i + t) - p_j\|^2$$

Esistono numerosissime implementazioni dell'algoritmo ICP, che differiscono per metodi di ricerca delle corrispondenze o per la funzione d'ottimizzazione. Ne esistono implementazioni che non hanno come output solo trasformazioni rigide ma che ammettono come soluzione anche trasformazioni similari o affini.

Un limite alle performance dell'algoritmo è rappresentato dalla ricerca del NN per ogni punto, da ripetere ad ogni iterazione (passo 1 dell'algoritmo). Nell'articolo

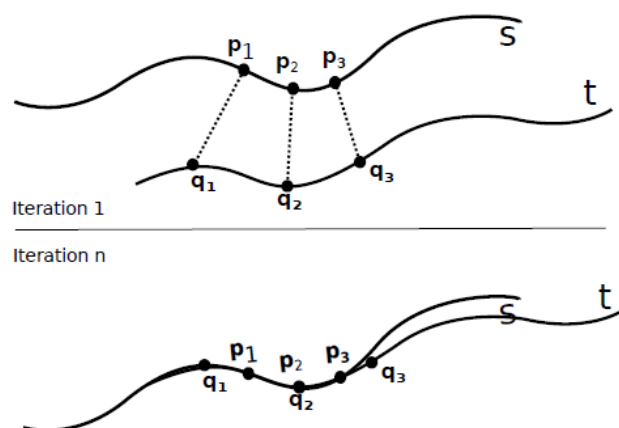


Figura 3.8: Allineamento basato sull'algoritmo ICP. [18]

di presentazione di ICP gli stessi autori propongono di sfruttare a questo proposito i **k-d tree** [19].

Il **k-d tree** (k-dimensional tree) è la struttura dati più usata per risolvere problemi di ricerca dei NN. È un albero binario nel quale ogni nodo rappresenta un partizionamento dello spazio k -dimensionale [19]. La radice dell'albero rappresenta lo spazio totale in cui si trovano i punti inseriti e i nodi foglia rappresentano sottospazi che contengono sottoinsiemi mutuamente esclusivi dell'insieme totale di punti inseriti [19]. Ad ogni nodo il partizionamento dello spazio, eseguito tramite iperpiani di separazione, può essere fatto sulla base di una sola delle k dimensioni dello spazio. I punti sono quindi memorizzati in base alla loro posizione spaziale in una certa parte dell'albero e ciò permette di ottimizzare la *query* di ricerca del punto più vicino a uno dato.

Dati n il numero di punti e d il numero di dimensioni, il costo di ricerca del NN passa da $O(nd)$ nel caso di ricerca lineare a $O(\log n)$ nel caso di uso del **k-d tree**, con costo di costruzione dell'albero nell'ordine di $O(nd \log n)$ [19].

Thin Plate Spline

Le *thin plate spline* (TPS) sono una tipologia di funzione spline poliarmonica comunemente utilizzata in interpolazione, introdotta da Duchon in geometria computazionale [17]. In analisi matematica, una spline è una funzione, costituita da un insieme di polinomi raccordati tra loro, il cui scopo è interpolare in un intervallo un insieme di punti (detti nodi della spline) in modo tale che la funzione sia continua almeno fino ad un dato ordine di derivate in ogni punto dell'intervallo.

Le TPS sono una delle tipologie più comuni di funzioni radiali di base (*Radial Basis Function*, RBF). Le funzioni RBF furono introdotte negli anni '60 per gestire

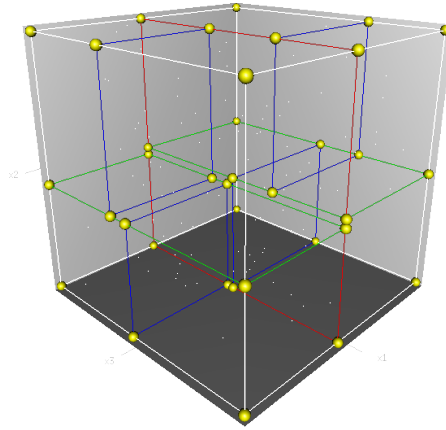


Figura 3.9: Partizionamento dello spazio tridimensionale per la creazione del k-d tree. Fonte: https://en.wikipedia.org/wiki/K-d_tree

problemi di interpolazione multidimensionale. Dati un insieme di N valori scalari sparsi come serie di punti x_{k_i} nello spazio \mathbb{R}^n , viene introdotto l'interpolante s per determinare una funzione regolare nello stesso spazio che approssimi i punti. Alla posizione x il suo valore è:

$$s(x) = \sum_{i=1}^N \gamma_i \phi(\|x - x_{k_i}\|) \quad (3.1)$$

La sommatoria in eq. 3.1 è estesa a tutti gli N punti, detti punti *source*. I punti trovati dall'interpolazione g_i sono invece detti punti *target*: $s(x_{k_i}) = g_i$, per $i = 1, \dots, N$.

ϕ è la funzione RBF, ovvero una funzione scalare ($\mathbb{R} \rightarrow \mathbb{R}$) definita in base alla distanza euclidea tra i punti sorgente e target. γ_i sono invece i pesi della base radiale; per calcolarli è necessario risolvere un sistema lineare di equazioni di ordine uguale al numero di punti sorgente.

La funzione RBF che definisce la TPS è:

$$\phi(r) = r^n \log(r) \quad (3.2)$$

con $r = \|x - x_{k_i}\|$ [20]. L'analogia fisica alla base delle TPS, da cui prendono il nome, è quella di determinare la deformazione di una sottile lastra di metallo. Così come il metallo ha una certa rigidità alla deformazione, la modellazione con TPS permette di definire un fattore di resistenza (λ) che permette di realizzare deformazioni più o meno rigide.

Le TPS sono largamente usate nell'ambito dell'elaborazione di immagini digitali per modellare trasformazioni elastiche¹ che permettano di mappare un insieme di

¹Deformazioni proporzionali allo sforzo o tensione che agisce sul corpo.

landmark presi sulla superficie moving sui corrispondenti appartenenti alla static. Il principio che sta alla base del *warping* effettuato dalle TPS è la minimizzazione di una funzione d'energia o costo. Tale funzione tiene conto sia del movimento dell'oggetto che deve essere deformato in base alle sue proprietà fisiche, che della massimizzazione della similarità tra le superfici, basandosi ad esempio sul minimizzare la distanza tra i landmark. La funzione di energia definita in \mathbb{R}^2 da minimizzare è la seguente: [21]

$$I_f = \iint_{\mathbb{R}^2} (f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2) dx dy$$

dove $f(x, y)$ è la funzione di interpolazione e ha la forma:

$$f(x, y) = a_1 + a_x x + a_y y + \sum_{i=1}^p w_i \phi(\|(x_i, y_i) - (x, y)\|)$$

dove $\{(x_i, y_i), i = 1, 2, \dots, p\}$ sono i punti di controllo, $\{w_i\}$ è un set di coefficienti e ϕ la funzione RBF di Equazione 3.2.

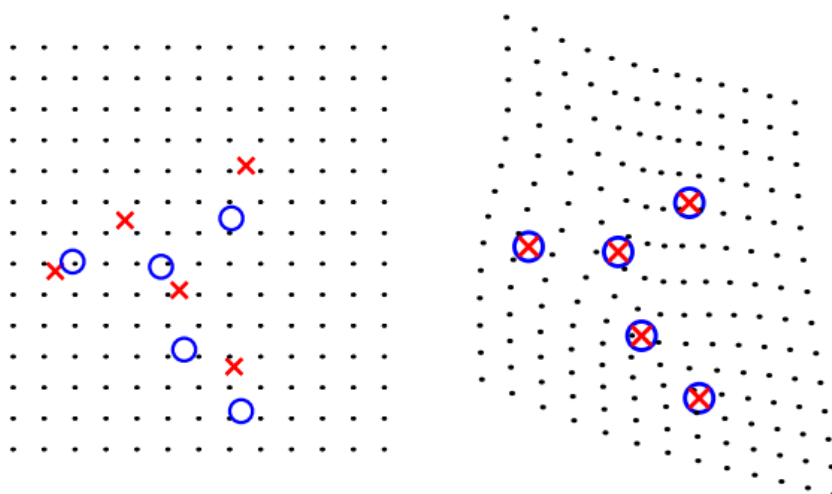


Figura 3.10: Esempio di deformazione con TPS per portare i landmark in corrispondenza. [21]

In Figura 3.10 è riportato un esempio di deformazione di una superficie: dati due set di punti in corrispondenza, come si vede nella rappresentazione a sinistra, il *warping* imposto dall'applicazione della TPS porta i due set in sovrapposizione esatta (pannello a destra).

Dal momento che la trasformazione si basa su un insieme di landmark selezionati sulle superfici, può accadere che ad essi sia associato un errore di localizzazione,

che potrebbe compromettere l'allineamento. Per sopperire a tale problema, presente soprattutto in applicazioni reali, è possibile calibrare il parametro di *smoothness* della trasformazione ($\lambda \geq 0$), che rende l'interpolazione più o meno rigida. Se λ assume valori piccoli la soluzione tenderà ad essere più rigida ($\lambda = 0$ corrisponde a un'interpolazione esatta). Se invece λ assume valori più alti avremo una trasformazione più morbida, con poca deformazione e meno sensibile alle distorsioni dovute all'errore di localizzazione dei landmark. Come si vede in Figura 3.11, la

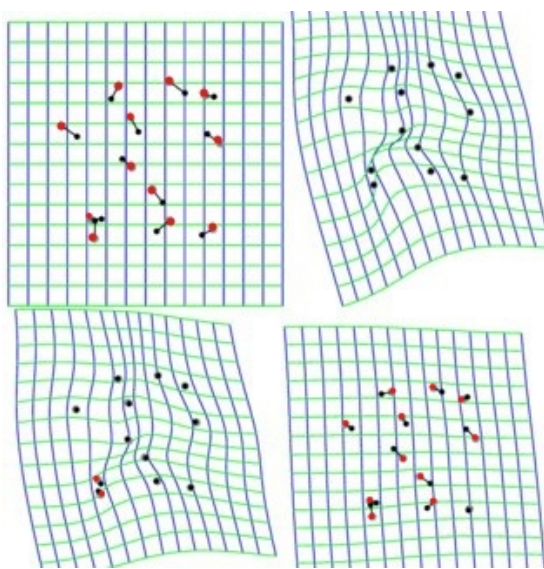


Figura 3.11: Trasformazione con TPS per diversi valori di λ . [22]

deformazione varia molto al variare di λ : nel pannello in alto a sinistra è riportata la superficie originale, dove i punti neri sono i landmark appartenenti alla source e quelli rossi quelli target. Nel pannello in alto a destra vediamo la deformazione ottenuta con $\lambda = 0$, ovvero l'interpolazione esatta, che porta a forti deformazioni della superficie e completa sovrapposizione dei landmark, in basso a sinistra è riportata la deformazione per $\lambda = 0,001$ mentre in basso a destra per $\lambda = 0,1$. Si può notare in quest'ultima come la deformazione sia meno accentuata, non portando i landmark in esatta sovrapposizione, ma si limiti ad avvicinarli [22].

Analisi di Procrustes

L'analisi di Procrustes consiste in un insieme di metodi matematici ai minimi quadrati che permettono di applicare trasformazioni che utilizzano traslazioni, rotazioni e *scaling* isomorfo per allineare al meglio due forme o un set di esse [23]. Il nome di tale algoritmo deriva da un personaggio della mitologia greca che costringeva le sue vittime a sdraiarsi su un giaciglio e adeguava la loro statura alle dimensioni del letto, tagliando o allungando loro le membra.

L'analisi di Procrustes si basa su determinate definizioni e assunzioni di validità:

- *Forma*: indica l'insieme delle informazioni geometriche una volta che gli effetti di posizione, rotazione e scala sono rimossi dall'oggetto [24];
- *Landmark*: è un punto di corrispondenza per ogni oggetto omologo tra e all'interno di insiemi di forme. Un set di landmark permette di descrivere accuratamente una forma [24];
- L'analisi sarà valida su set di forme che abbiano lo stesso numero di landmark e che essi siano in corrispondenza uno a uno.

L'insieme delle coordinate dei landmark è detto **configurazione** ed è definito da una matrice $X \in \mathbb{R}^{n \times m}$, dove n è il numero di landmark e m il numero delle dimensioni [23].

L'allineamento si basa su alcuni passi fondamentali (Figura 3.12):

1. **Filtraggio effetti traslazione**: le matrici di configurazione vengono centrate tramite la *centering matrix*:

$$C = I_n - \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T$$

dove $\mathbf{1}_n \in \mathbb{R}^n$ è un vettore composto da tutti elementi di valore 1 e I_n è la matrice identità di ordine n . Possiamo quindi definire $X_C = CX$ come la configurazione centrata nell'origine: una volta calcolato il centroide di ogni forma, esso è sottratto a tutti i punti per centrare la forma nell'origine;

2. **Filtraggio trasformazioni di scala**: la trasformazione di scala da rimuovere è di tipo isomorfo, ovvero che mantiene le proporzioni delle forme. Tale effetto può essere rimosso tramite:

$$X_n = X_C \left(\frac{1}{\|X_C\|} \right)$$

dove X_n sono le coordinate centrate e normalizzate;

3. **Filtraggio effetti rotazione**: l'ultimo step è la rimozione degli effetti di rotazione. È lo step più delicato e complesso, poiché si basa su tecniche di minimizzazione ai minimi quadrati che portano le forme all'allineamento migliore possibile. In particolare, la distanza che va minimizzata è detta distanza di Procrustes ed è definita come:

$$d_p(X_1, X_2) = \min_R \|X_{2n} - X_{1n}R\|^2$$

dove R è la matrice ortogonale di rotazione e X_1 e X_2 le configurazioni da allineare [23].

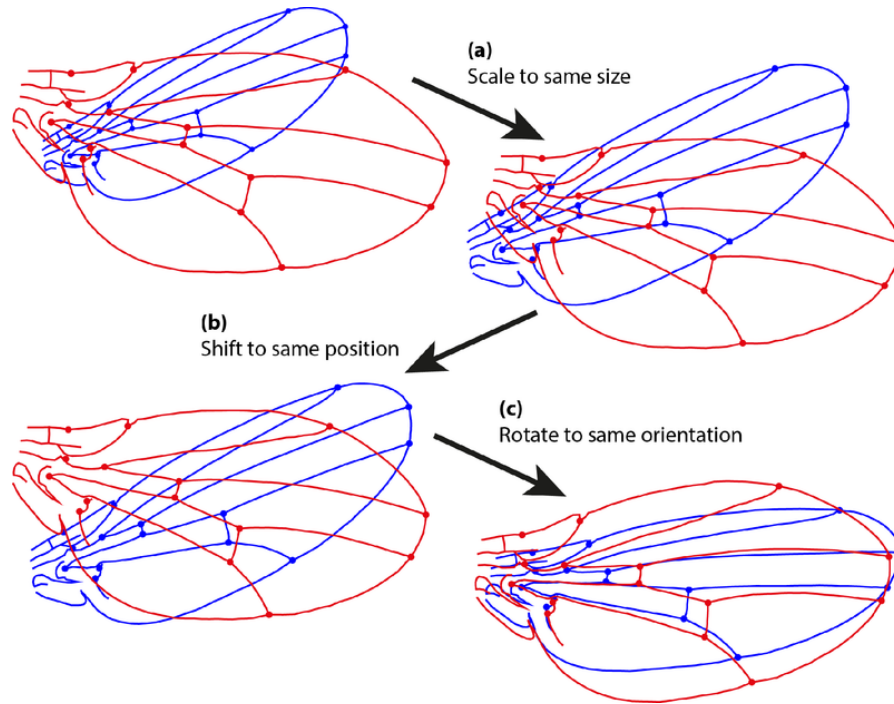


Figura 3.12: Fasi dell'allineamento di Procrustes. Fonte: https://www.wikiwand.com/en/Procrustes_analysis

Vi sono due tipi di analisi di Procrustes:

1. Analisi di Procrustes **ordinaria**, che mira a trovare il *match* tra due forme;
2. Analisi di Procrustes **generalizzata**, che cerca di analizzare e allineare insieme composti da più di due forme (Figura 3.13).

Il primo tipo, quello che più si avvicina agli algoritmi di registrazione precedentemente descritti, segue le fasi descritte sopra per cercare il *match* ottimo tra due forme minimizzando la funzione obiettivo:

$$(\hat{\beta}, \hat{R}) = \operatorname{argmin}_{\beta, R} \|X_2 - \beta X_1 R\|^2$$

dove $\beta > 0$ indica il fattore di scala da applicare e R la matrice di rotazione per ottenere l'allineamento che permette di minimizzare la distanza tra X_1 e X_2 [23].

Il secondo tipo, invece, permette di allineare k set di forme a una target o di allinearle tra loro. I passi da seguire per effettuare l'allineamento sono i seguenti [24]:

1. Selezionare una forma del set (ad esempio la prima) come *reference*;
2. Allineare le altre forme a quella selezionata:

- a. Calcolare il centroide di ogni forma;
 - b. Allineare tutti i centroidi con l'origine;
 - c. Normalizzare le forme;
 - d. Ruotare ogni forma per allinearla alla *reference*;
3. Calcolare la forma media dalle forme allineate e sostituirla alla *reference*;
 4. Se la forma media è cambiata dal passo 2 al passo 3, iterare dal passo 2, altrimenti è stata trovata la forma media del set e l'allineamento è concluso.

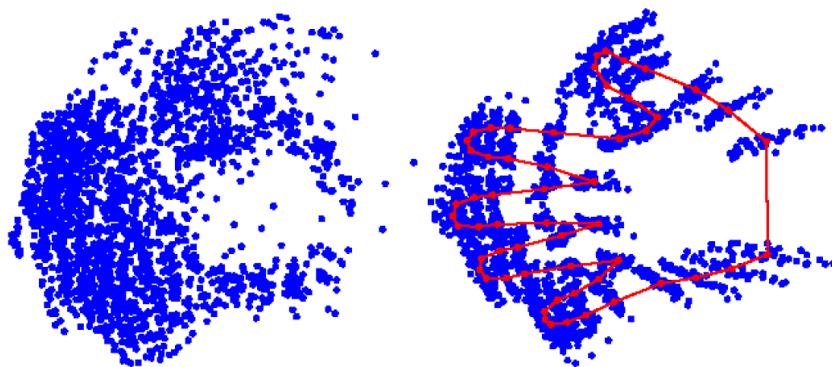


Figura 3.13: Esempio di analisi di Procrustes generalizzata: a sinistra sono riportati i landmark di 40 diverse forme, a destra esse sono allineate e in rosso è rappresentata la forma media. [24]

Il vantaggio principale dell'analisi di Procrustes generalizzata è il fatto di essere un algoritmo di facile implementazione per allineare due o più forme a partire dalle loro corrispondenze. D'altra parte, il principale limite è costituito dal fatto che sia necessario avere dataset con stesso numero di punti e che essi siano in corrispondenza uno a uno. L'algoritmo non prevede infatti fasi di ricerca delle corrispondenze tramite criteri di similarità, come avviene invece nel caso di ICP (Sezione 3.1.5). Un ulteriore svantaggio è il fatto che non sia sempre garantita la convergenza, soprattutto in caso di forme che differiscono molto le une dalle altre [24].

Algoritmo di Sanders et al.

L'algoritmo di registrazione proposto da Sanders et al. [25] è una valida alternativa per allineare due scansioni l'una all'altra nel modo migliore possibile. Tale algoritmo è stato introdotto all'interno di uno studio condotto su pazienti con amputazione transtibiale con lo scopo di definire delle metriche quantitative che

possano individuare e quantificare tipi di errori diversi (forma, volume). La ricerca mira quindi a individuare quale valore di errore sia da considerare clinicamente rilevante e di conseguenza definire degli standard di qualità da seguire nella produzione delle protesi, basandosi sul confronto di invasature prodotte da centri produttivi diversi [25].

L'obiettivo dell'algoritmo di registrazione è trovare la trasformazione da imporre a una superficie per allinearla all'altra. L'output della routine di ottimizzazione è quindi i set di parametri T_x, T_y, T_z e $\theta_x, \theta_y, \theta_z$, ovvero le componenti di traslazione e rotazione rispetto agli assi x, y e z . Non è previsto alcun termine di scala, poiché la trasformazione da imporre è di tipo rigido. L'algoritmo si basa su determinate procedure di preparazione dei dati:

1. Creazione di superfici continue a partire dai punti delle scansioni da allineare tramite interpolazione B-spline;
2. Campionamento della superficie reference tramite una griglia uniformemente spaziata (angolo θ , intervallo z), definendo un nuovo set di punti

$$v_1^R = [x, y, z, 1]$$

Calcolo delle normali alla superficie in corrispondenza dei punti campionati, definendo il dataset

$$n_1^R = [n_x, n_y, n_z, 1]$$

3. Definizione dei parametri di ottimizzazione. Vengono forniti come parametri iniziali valori di $T_x, T_y, T_z, \theta_x, \theta_y, \theta_z$ pari a $[0, 0, 0, 0, 0, 0]$ e posti come limiti di traslazione -40 mm e $+40$ mm e di rotazione -45° e $+45^\circ$.

Il vero e proprio algoritmo iterativo di allineamento avviene in due step, entrambi eseguiti per un certo numero di iterazioni e uno di seguito all'altro. I due step differiscono per i valori associati ai pesi *RadialWeight* e *NormalWeight* (Equazione 3.5): nel primo step tali pesi sono posti rispettivamente a 1 e 0, nel secondo a 0,8 e 0,2. I passi su cui iterare sono i seguenti:

- a. Trasformazione della superficie v_1^R in v_2^R e delle sue normali tramite i parametri di traslazione e rotazione iniziali o calcolati all'iterazione precedente;
- b. Campionamento della superficie target allo stesso modo fatto per la reference al punto 2, ottenendo il dataset v_2^M e le rispettive normali come n_2^M . Si ottengono in questo modo punti appartenenti alle due superfici in corrispondenza uno a uno;

- c. Calcolo dell'**errore radiale medio (MRE)**, dato dalla media dell'errore radiale calcolato tra ogni punto della superficie static e il corrispondente sulla moving:

$$r_2^R = \sqrt{(v_{2x}^R)^2 + (v_{2y}^R)^2 + (v_{2z}^R)^2}$$

$$r_2^M = \sqrt{(v_{2x}^M)^2 + (v_{2y}^M)^2 + (v_{2z}^M)^2}$$

$$MeanRadialError = mean(r_2^M - r_2^R) \quad (3.3)$$

- d. Calcolo della **similarità delle normali**, calcolata in base all'angolo formato dalle normali dei punti corrispondenti:

$$NormalSimilarity = mean(\tanh^{-1}(n_2^R \cdot n_2^M - 10^{-7})) \quad (3.4)$$

La similarità è maggiore per vettori che tendono ad essere paralleli. Il termine di sottrazione 10^{-7} è dovuto al fatto che se le normali fossero tra loro parallele il loro prodotto scalare avrebbe valore 1 e di conseguenza l'arcotangente iperbolica tenderebbe ad infinito;

- e. Calcolo della **funzione obiettivo** da minimizzare:

$$f = RWeight \times MeanRadialError - NWeight \times NormalSimilarity \quad (3.5)$$

Dal momento che la funzione obiettivo viene minimizzata durante la routine d'ottimizzazione, il secondo termine, che indica la similarità tra le superfici, è sottratto al primo, il quale invece è una misura di errore. I valori di traslazione e rotazione ottenuti dall'ottimizzazione sono usati come input per la prossima iterazione, in modo da convergere gradualmente alla soluzione migliore.

Una volta raggiunto il numero di iterazioni massimo o la soglia di terminazione della funzione obiettivo (10^{-6}) si considera raggiunto l'allineamento.

3.1.6 Preparazione dato per il mesh morphing

Una volta registrate le mesh, esse possono essere preparate per eseguire operazioni di morphing. Il mesh morphing permette di modellare una mesh per ottenere modifiche di forma e volume preservandone la topologia. Un buon software di morphing deve essere in grado di operare tale modellazione senza introdurre eccessiva perdita di qualità della mesh [26]. Il software commerciale usato per compiere tale operazione è *RBF-Morph* (RBF Morph srl. Roma) [27]. Esso utilizza in insieme di funzioni RBF, introdotte in Sezione 3.1.5, per generare un campo di deformazione

sulla base di punti in corrispondenza e applicare tale deformazione alla superficie target [27].

L'input a tale metodo di modellazione è l'insieme dei vettori spostamento che portano i landmark della mesh target sui rispettivi della reference. Pertanto, viene generato un file in formato specifico del software che eseguirà il morphing (.pts) che conterrà le coordinate dei landmark individuati sulla superficie target insieme alle coordinate dei vettori spostamento che portano tali punti sui rispettivi della superficie reference.

Il file generato avrà quindi il seguente formato:

```
n
v1x v1y v1z d1x d1y d1z 0 0 s p
...
vnx vny vnz dnx dny dnz 0 0 s p
```

La prima riga indica il numero di punti considerati, per ognuno dei quali viene aggiunta una riga contenente le coordinate x, y, z del punto e del *displacement*, seguiti dai caratteri $0, 0, s, p$.

3.1.7 Outcomes

Per eseguire valutazioni qualitative e quantitative sulle scansioni sono stati elaborati una serie di parametri di *outcome*, che mirano a fornire differenti informazioni ed evidenze cliniche.

Mappe distanza

Le mappe distanza-colore 3D sono una rappresentazione grafica delle differenze tra due immagini 3D sovrapposte. Esse permettono una modalità di visualizzazione ed elaborazione dati efficace e la loro analisi consente di identificare e quantificare agevolmente zone problematiche e valutare i cambiamenti di forma. Riprendono una prassi ordinaria nei controlli dimensionali della meccanica, basandosi sul calcolo dello scarto fra le superfici. Esse possono essere utilizzate ad esempio per l'analisi di scansioni che rappresentano la stessa invasatura in diverse fasi di fabbricazione.

Date due superfici $S1$ ed $S2$, la definizione di tali mappe dipende da due fattori principali:

1. Modalità di calcolo della distanza;
2. Posizione relativa di $S1$ ed $S2$, dipendente dai precedenti processi di allineamento e registrazione.

La terminologia associata all'identificazione delle due superfici a confronto consiste in *measured*, mesh a cui vengono associati i valori di distanza e *reference*, ovvero la mesh rispetto alla quale viene calcolata la distanza.

Per quanto riguarda la voce 1, dati due punti corrispondenti $v^{\vec{M}}$ e $v^{\vec{R}}$ e le rispettive normali $n^{\vec{M}}$ e $n^{\vec{R}}$, appartenenti a *measured* e *reference*, sono state definite diverse tipologie di computazione, basate su differenti associazioni di punti e diverse metriche. Le metriche di riferimento sono:

- **RE-INAIL**, tale distanza è stata ideata durante lo studio come valida alternativa a quella elaborata da Sanders et al. in Equazione 3.3. Inoltre, sopperisce alla scarsa chiarezza relativa al calcolo della distanza introdotta in alcuni software di geometria computazionale, fra cui MeshLab. Permette infatti di calcolare lo scostamento tra due superfici a confronto, dati due punti omologhi. Viene definita come:

$$RE_{INAIL} = \vec{d} \cdot \vec{n}^{\vec{R}} \quad (3.6)$$

$$\vec{d} = v^{\vec{M}} - v^{\vec{R}} \quad (3.7)$$

In particolare, essa è la proiezione del vettore differenza tra i due punti associati \vec{d} sulla normale al piano tangente a *reference* in $v^{\vec{R}}$ (fig. 3.15). Il segno è definito secondo la concordanza tra la normale e il vettore differenza (determinato direttamente dal prodotto scalare), rappresentato in fig. 3.14.

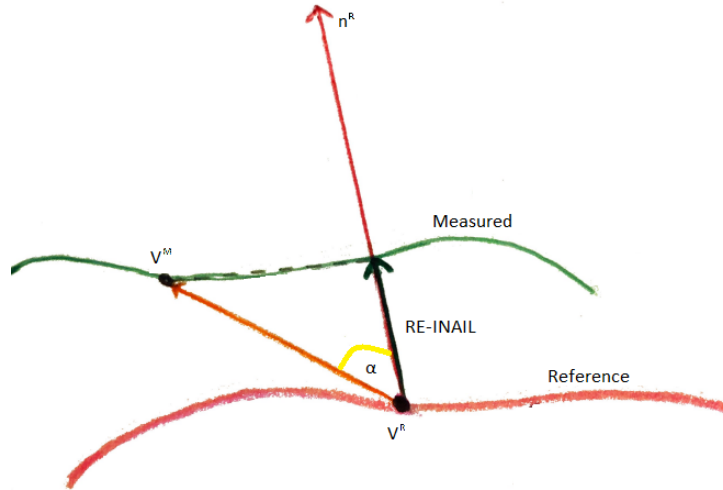


Figura 3.14: Ottenimento di RE-INAIL (in nero) tramite proiezione di \vec{d} (in arancione) sulla normale $n^{\vec{R}}$ a $v^{\vec{R}}$ (in rosso) in *reference*.

- **Vertice-Vertice**, si basa sul concetto di distanza Euclidea (fig. 3.15):

$$VV = \|\vec{v}^M - \vec{v}^R\|$$

A tale parametro si attribuisce il segno tramite analisi di:

$$d_{k_{proiettata}} = \vec{d} \cdot \vec{n}^R$$

Il valore di VV viene infatti moltiplicato per -1 se la proiezione ortogonale $d_{k_{proiettata}}$ è minore di 0.

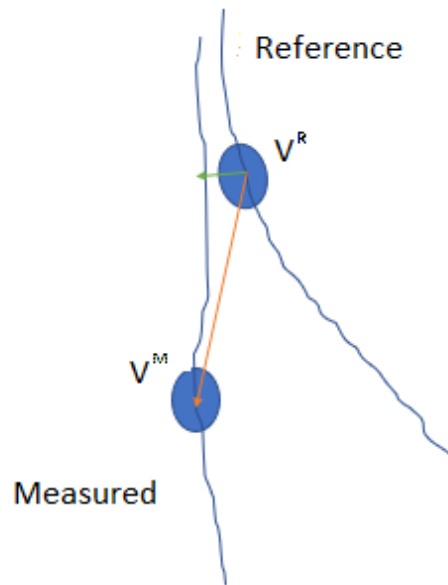


Figura 3.15: Confronto tra distanza RE-INAIL (in verde) e Vertex-Vertex (in arancione) definite tra gli stessi punti omologhi.

- **Meshlab**, utilizzando la funzione `distance_from_reference_mesh`², misura la distanza da ogni vertice di una mesh al punto più vicino sulla superficie della mesh di riferimento. Quindi itera i vertici della measured e i triangoli della reference, proiettando ogni vertice su ogni triangolo e prendendo la distanza minima per ogni coppia vertice-triangolo;
- **SNAE**, dà informazioni circa le differenze di forma locali ed è calcolato in base all' tra l'angolo formato dalle normali dei punti corrispondenti:

$$SNAE = \arccos(\vec{n}^M \cdot \vec{n}^R) \quad (3.8)$$

²Libreria Python per l'elaborazione batch di mesh utilizzando i filtri MeshLab https://pymeshlab.readthedocs.io/en/latest/filter_list.html

- **Normal Similarity**, rappresentativo della similarità tra le normali $n^{\vec{M}}$ e $n^{\vec{R}}$ delle corrispondenze, anch'esso si basa sulla differenza tra gli angoli formati dalle normali dei punti a confronto ed è definito da:

$$NormalSimilarity = \tanh^{-1}(n^{\vec{M}} \cdot n^{\vec{R}} - 10^{-7}) \quad (3.9)$$

Tale parametro coincide con quello utilizzato per l'algoritmo di allineamento di Sanders et al. eq. 3.4.

Come anticipato, la creazione delle mappe distanza, oltre che su diverse metriche, si basa su differenti associazioni tra i punti a confronto.

Per quanto riguarda RE-INAIL, SNAE e NormalSimilarity, le corrispondenze vengono identificate tramite la ricerca per ogni vertice della measured del *closest point* in reference. L'individuazione del closest point si effettua tramite la creazione del k-d tree (sezione 3.1.5): l'albero viene creato sui vertici della reference e viene interrogato tramite i vertici della measured. L'interrogazione restituisce l'indice della corrispondenza (NN) in reference rispetto ad ogni vertice della measured. Invece, per quanto riguarda VV, le corrispondenze vengono assegnate sulla base dello stesso indice, pertanto tale metrica può essere applicata solo a mesh con stesso numero di vertici. Le corrispondenze permettono la computazione del parametro l'output d'interesse per ogni vertice della measured.

La distanza VV risulta di particolare interesse quando le due superfici sono ottenute per deformazione diretta della prima verso la seconda. A titolo di esempio si può pensare ad un flusso di materia che evolve nel tempo o nello spazio, come fa un palloncino su cui si sia disegnato un set di punti e che venga successivamente gonfiato o il moncone di una persona all'esterno ed all'interno del sistema Aquasystem. La distanza RE-INAIL è invece di interesse quando la seconda superficie viene ottenuta per addizione o sottrazione di materiale, come ad esempio nella stilizzazione di un negativo.

I **passaggi fondamentali** per la creazione delle mappe consistono in:

1. Calcolo della distanza desiderata per ogni vertice della measured;
2. Allocazione del dato come parametro qualità associato al vertice;
3. Analisi della distribuzione dei valori di distanza per l'identificazione degli estremi utili;
4. Definizione della scala colore e assegnazione del valore colore ad ogni vertice;
5. Salvataggio dei valori di distanza su file formato *.xlsx*, per successive analisi statistiche delle differenze;

6. Salvataggio dei valori di distanza e colore associati ad ogni vertice come parametri qualità in formato PLY.

La Voce 3 si basa sull'analisi della distribuzione della qualità: essa viene effettuata tramite lo *scarto interquartile (IQR)* mantenendo tale analisi non parametrica. L'IQR è un indice di dispersione: quantifica l'estensione del 50% della distribuzione del dato che si trova attorno alla mediana. È definito come:

$$IQR = Q3 - Q1$$

ovvero la differenza tra il terzo e il primo quartile (*25esimo* e *75esimo* percentile).

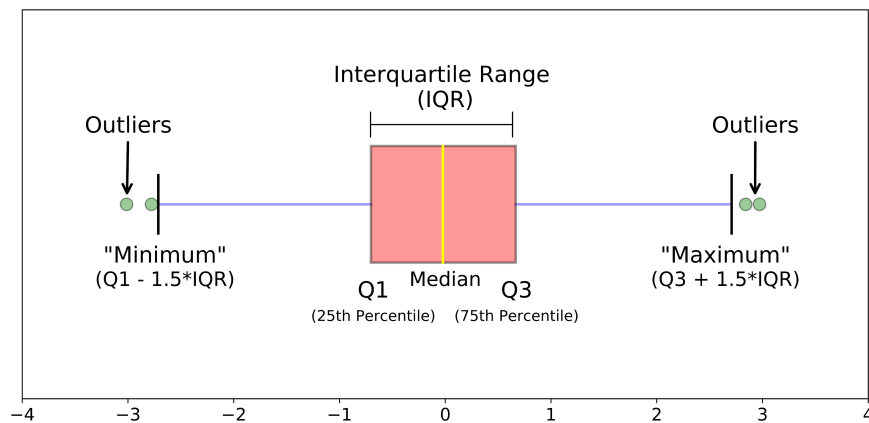


Figura 3.16: Rappresentazione grafica degli elementi che caratterizzano la distribuzione, tramite *box plot*.

L'obiettivo è l'**identificazione dei baffi** (*whiskers*) della distribuzione, ovvero i valori estremi della distribuzione su cui definire gli *outlier*, tale operazione consiste in (fig. 3.16):

1. Ordinamento degli n valori della distribuzione in ordine crescente;
2. Calcolo di $Q1$ e $Q3$;
3. Definizione del valore minimo e massimo del range come:

$$Minimum = Q1 - 1,5 \times IQR$$

$$Maximum = Q3 + 1,5 \times IQR$$

Una volta ottenuti i *whiskers* si procede con l'assegnazione del colore (voce 4) che permette un riepilogo visivo rapido della variabilità dei valori del dataset in studio. Tale operazione consiste nella definizione di una funzione che permetta di mappare una distribuzione di dati su uno spazio colore. Lo spazio colore viene identificato da tre colori:

1. Un colore associato al *Minimum*, rappresentante la soglia di saturazione minima;
2. Un colore associato sempre allo zero, per avere un punto di riferimento costante;
3. Un colore associato al *Maximum*, rappresentante la soglia di saturazione oltre la quale tutti i valori vengono associati a tale colore.

Nel caso di SNAE e NormalSimilarity, vengono utilizzati soltanto due colori data la natura del parametro, attribuendo un colore allo zero e l'altro al massimo.

Le mappe rappresentano un valido strumento di analisi dimensionale permettendo l'interpretazione rapida della variabilità o dispersione dei dati.

Regione di Interesse

Talvolta può rendersi necessario analizzare una specifica regione della mesh, specialmente alla luce di quanto emerso dalle mappe distanza comparative. A tale scopo si utilizza un metodo di "ritaglio" della zona di interesse, rispetto alla quale valutare i valori di qualità punto a punto. Questo approfondimento può essere utile per poter confrontare la stessa zona di interesse ad esempio su pazienti diversi. Ciò consente l'analisi delle distribuzioni dei valori di qualità associati a specifiche zone confrontate inter-paziente. Questo permette di evidenziare problematiche comuni o ricorrenti, che potrebbero essere dovute a lavorazioni errate.

Il **procedimento** segue le seguenti fasi:

1. Estrazione della regione di interesse (ROI) sulla base di tre o più punti selezionati sulla mesh, che delimitano una superficie;
2. Analisi dei punti interni a tale regione, ad ognuno dei quali viene correlato un valore di qualità (precedentemente calcolato);
3. Generazione di un grafo contenente i vertici nei nodi, insieme al valore di qualità associato;
4. Interrogazione del grafo per selezionare i vertici (nodi) che soddisfano un determinato criterio definito da utente sulla base del range di valori associati alla ROI.

Una mesh di fatto può essere vista come un grafo non orientato, che ha come nodi i vertici e gli archi collegano tra loro vertici coinvolti nello stesso triangolo. È anche possibile associare un peso all'arco, ad esempio con la distanza tra i vertici che lega. In Figura 3.17 sono riportati esempi di grafi generati da mesh con numero incrementale di vertici: a sinistra il grafo generato da un cubo, a destra da una mesh di 60 vertici e in basso da una di 1200 vertici. L'interrogazione sul grafo analizza tutti i nodi selezionando quelli a cui è associato un peso interno al range di valori di qualità specificato. I valori di qualità risultanti dalla query vengono allocati in un file: ad essi sono associati l'indice rispetto alla mesh originale e rispetto alla ROI.

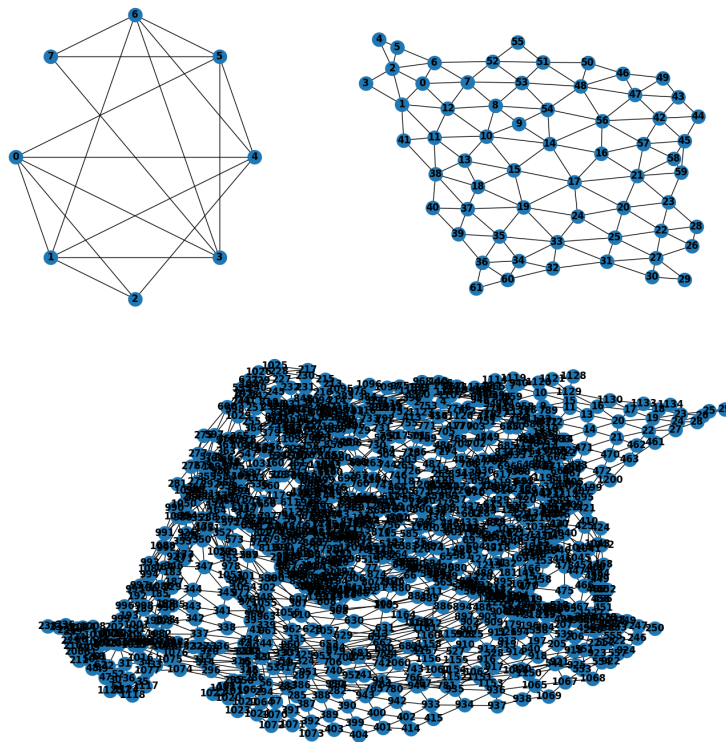


Figura 3.17: Grafi generati rispettivamente da un cubo, una mesh di 60 e 1200 vertici.

Sezionamento

Nella pratica clinica dell'ortopedia tecnica il rilievo delle misure da parte del TO durante la valutazione del moncone è un passaggio indispensabile. Tali misure vengono riportate sull'apposita scheda progetto, supportano il TO nella scelta della cuffia e, successivamente, vengono utilizzate per la stilizzazione del modello positivo e per l'assemblaggio della protesi.

Alcune delle misure d'interesse possono riguardare:

- Le circonferenze in corrispondenza di determinati landmark;
- Le lunghezze determinate proporzionalmente tra un riferimento prossimale e uno distale;
- Il diametro medio-laterale e antero-posteriore.

Con il termine *sezione* si intende la figura piana risultante dall'intersezione di un solido con un piano. Il *piano di taglio* viene definito sulla base di un punto e un asse: il *punto* definisce la quota alla quale applicare il taglio, mentre l'*asse* l'orientazione che il taglio dovrà avere.

Nel nostro caso le sezioni vengono estratte dopo l'allineamento: l'asse è dato dalla componente longitudinale z e i piani di sezione sono perpendicolari ad esso.

Una volta estratte le sezioni i parametri calcolati per ognuna di esse sono:

- Perimetro;
- Area superficiale, data dalla somma delle aree dei singoli triangoli che costituiscono la mesh della sezione;
- Volume residuo, calcolato come volume della mesh sottostante la quota di taglio;
- Massima elongazione medio-laterale, ovvero calcolata sulla coordinata y come $(y_{max} - y_{min})$;
- Massima elongazione antero-posteriore, ovvero calcolata sulla coordinata x come $(x_{max} - x_{min})$.

Sulla base dei dati estratti si possono effettuare controlli dimensionali e analisi statistiche. I cambiamenti nel volume e nella forma dell'arto residuo nel corso della giornata possono infatti portare a problemi con la creazione e il mantenimento di un'invasatura che permetta un adattamento accurato e confortevole. Per un buon accoppiamento dell'interfaccia dell'invasatura protesica con l'arto residuo è opportuno che ci sia una precisa corrispondenza tra il volume del moncone ed il volume interno all'invasatura senza causare dolore o disagio. Pertanto la rilevazione delle misure dell'arto residuo e la valutazione delle variazioni di esse durante il processo di lavorazione è importante ottenere un buon contenimento dell'arto.

Cammino geodetico

In ambito clinico, ad esempio durante l'individuazione dei landmark anatomici da parte del TO, vengono tracciati dei percorsi geodetici. Tali cammini rappresentano il percorso più breve tra due estremi corrispondenti ad un punto prossimale e ad un punto distale appartenenti all'arto residuo. Su tale percorso vengono identificati dei punti a determinate percentuali corrispondenti a specifici landmark. Da un punto di vista computazionale, la possibilità di identificare la geodetica nello spazio tridimensionale della mesh permette di aggiungere a posteriori al cammino individuato dal TO ulteriori punti in funzione di una distanza percentuale. Inoltre, ciò consente la verifica a posteriori delle misure identificate durante la visita al paziente.

Il termine "geodetica" deriva da *geodesia*, ovvero scienza della misurazione delle dimensioni e della forma del globo terrestre [28]. Tale misurazione si basa sul concetto distanza geodetica ovvero la minima distanza necessaria a connettere due punti appartenenti ad una superficie curva, la raffigurazione di ciò è visibile in Figura 3.18.

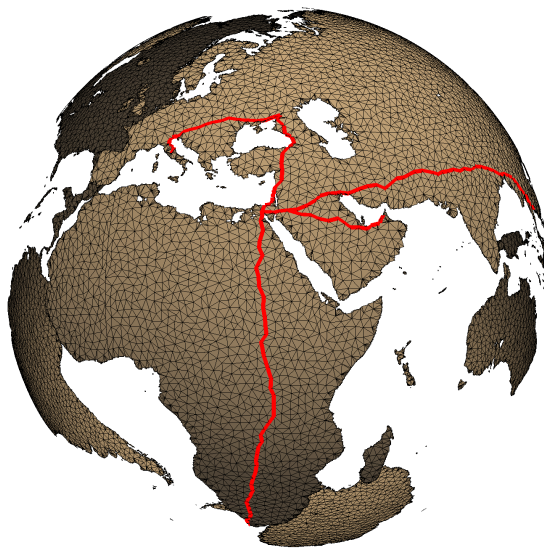


Figura 3.18: Esempi di percorsi geodetici. Fonte: <https://docs.pyvista.org/examples/01-filter/geodesic.html?highlight=geodesic>

Nel caso in analisi, la geodetica è una curva vincolata a una superficie che si basa sulla triangolazione della mesh e che minimizza localmente la distanza intrinseca tra due vertici, in modo tale da soddisfare il principio su cui si basa il problema del cammino minimo.

Dato un grafo $G = (V, E)$, definiamo come V l'insieme dei nodi e come E l'insieme degli archi $e = (u, v)$. A ciascun arco viene associato un peso $p(u, v) \in \mathbb{R}$ [29]. Per ogni cammino orientato $P = \{v_1, e_1, \dots, e_{p-1}, v_p\}$, definiamo peso $p(P)$ del cammino P la somma dei pesi degli archi che appartengono a P , ovvero

$$p(P) = \sum_{u,v \in P} p(u, v)$$

Il problema del cammino minimo può essere enunciato nel modo seguente [29]:

Dati due nodi $s \in V$ e $t \in V$, trovare un cammino orientato P in G da s a t che abbia peso minimo.

Per il caso d'uso specifico l'ottenimento del parametro d'output si basa sull'algoritmo iterativo di Dijkstra: dato un grafo, con pesi sugli archi non negativi, e un vertice sorgente nel grafo, esso trova i percorsi più brevi dalla sorgente a tutti i vertici nel grafo dato.

Per un grafo G la complessità computazionale dell'algoritmo di Dijkstra è pari a:

$$O(|E| + |V| \log |V|)$$

dove $|E|$ è il numero di archi e $|V|$ il numero di nodi [30]. L'implementazione inizia con la creazione di due insiemi: uno contenente i vertici inclusi nell'albero dei percorsi più brevi, l'altro i vertici non ancora inclusi nel precedente. I passaggi principali nell'algoritmo di Dijkstra consistono in [30]:

1. Creazione del set *shortest path tree* (*sptSet*), che tiene traccia dei vertici inclusi nell'albero del percorso più breve; inizialmente tale set è vuoto;
2. Assegnazione di un valore di distanza a tutti i vertici appartenenti al grafo. Inizializzazione di tutti i valori di distanza ad *infinito* ad eccezione del vertice di origine al quale viene assegnato il valore di distanza 0.
3. Esecuzione iterativa finché *sptSet* non include tutti i vertici dei seguenti passaggi:
 - Selezione del vertice u che ha distanza minima e non è incluso in *sptSet*. Per la prima iterazione corrisponde ad u con distanza zero;
 - Inclusione di tale vertice u in *sptSet*;
 - Calcolo e aggiornamento delle distanze dei vertici adiacenti a u rispetto ad esso. Per ogni vertice adiacente v , se la somma del valore della distanza di u (dalla sorgente) e del peso dell'arco $u - v$ è inferiore al valore della distanza di v , aggiorna il valore della distanza di v .

In Figura 3.19 è possibile osservare che ad ogni iterazione a ogni nuovo vertice del fronte di avanzamento viene assegnata localmente una distanza ottimale dal vertice iniziale.

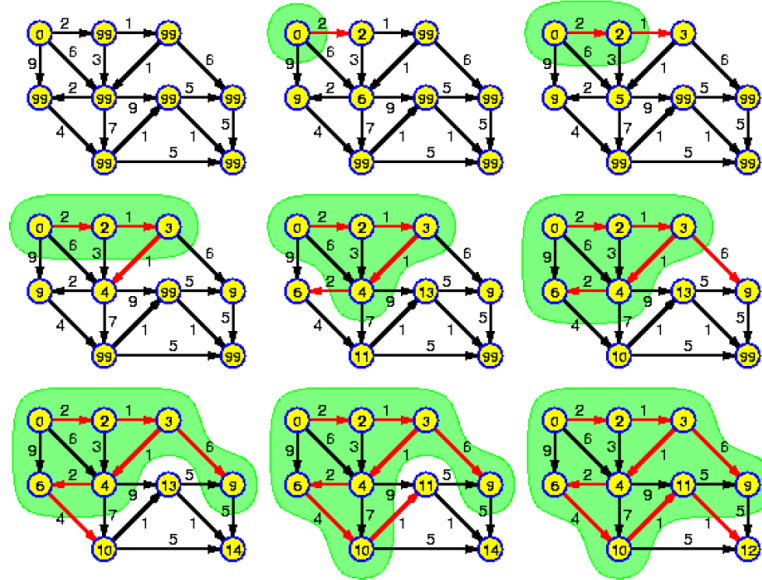


Figura 3.19: Esempio d'evoluzione dell'algoritmo di Dijkstra

L'algoritmo di Dijkstra fornisce un metodo esatto per trovare la distanza più breve che collega due punti; un suo limite riguarda la mancata convergenza nel caso in cui un qualsiasi arco ha valore negativo.

3.1.8 PCA

Esistono software che permettono di processare una mesh di riferimento, relativa all'anatomia di un paziente, e modellarla su un'altra mesh, relativa all'anatomia dello stesso paziente ma modificata. Il risultato di tale procedimento è la medesima mesh in termini di topologia e numero di punti applicata per ogni paziente prima e dopo un cambiamento avvenuto intra soggetto. Un esempio clinico corrisponde al passaggio da modello grezzo a stilizzato. Questo procedimento può essere realizzato attraverso tecniche di *Statistical Shape Modelling* (SSM [31]). Al termine dello SSM si ottengono quindi insiemi di dati accoppiati.

Sono stati definiti due metodi di analisi: la PCA, che può essere utilizzata per comprendere le differenze tra le anatomie di pazienti diversi, e la PCA differenziale, la quale può essere utile per estrarre una regola comune a un insieme di pazienti.

PCA per differenze inter-paziente

I modelli statistici di forma basati sulla PCA assumono che lo spazio delle possibili deformazioni possa essere appreso da un set di forme $\{\Gamma_1, \dots, \Gamma_n\}$, dove ogni forma Γ_i è rappresentata da un set di N landmark:

$$\Gamma_i = \{v_k^i | v_k \in \mathbb{R}^3, k = 1, \dots, N\}$$

Alcuni approcci utilizzano come landmark punti anatomici o di interesse, lavorando con un numero N abbastanza piccolo. Altri approcci, più moderni, utilizzano invece un insieme denso di punti, incrementando N . L'assunzione fondamentale necessaria in ogni caso è che i punti siano in corrispondenza esatta tra le forme: il k -esimo landmark di Γ_1 , deve corrispondere al k -esimo landmark in Γ_2 e così via fino a Γ_n . Tali corrispondenze possono essere definite manualmente o automaticamente tramite algoritmi di registrazione o *morphing*.

Per costruire il modello, ogni forma Γ_i viene rappresentata come un vettore $s_i \in \mathbb{R}^{3N}$, dove le componenti x, y, z di ogni punto sono disposte nel seguente modo: [31]

$$\vec{s}_i = (v_{1x}^i, v_{1y}^i, v_{1z}^i, \dots, v_{Nx}^i, v_{Ny}^i, v_{Nz}^i) \quad (3.10)$$

La matrice di input avrà quindi dimensioni $n \times 3N$. L'assunzione che si fa è che le variazioni di forma possano essere modellate usando una distribuzione normale:

$$s \sim N(\mu, \Sigma)$$

La media μ e la matrice di covarianza Σ sono stimate sui dati:

$$\mu = \bar{s} := \frac{1}{n} \sum_{i=1}^n \vec{s}_i$$

$$\Sigma = S := \frac{1}{n-1} \sum_{i=1}^n (\vec{s}_i - \bar{s}_i)(\vec{s}_i - \bar{s}_i)^T$$

La media μ rappresenta la forma media relativa al dataset analizzato. Il rango della matrice di covarianza è al massimo pari a n , ovvero il numero di autovalori non nulli. Nella sua interpretazione probabilistica, la PCA porta a un modello nella forma:

$$s = \bar{s} + \sum_{i=1}^n \alpha_i \sqrt{d_i} \vec{u}_i \quad (3.11)$$

dove (u_i, d_i) , $i = 1, \dots, n$ sono gli autovettori e autovalori della matrice di covarianza. Assumendo che $\alpha_i \sim N(0, 1)$, è facile verificare che $s \sim N(\bar{s}, S)$. È stata così ottenuta una rappresentazione parametrica della distribuzione [31].

A partire da questo modello, assegnando arbitrariamente ad α_i valori compresi nell'intervallo $[-3, +3]$ e riportando il risultato alle dimensioni $N \times 3$, si otterranno le coordinate relative alla nuova forma prodotta dal modello generato.

PCA differenziale

L'altro tipo di analisi viene eseguito tramite la "PCA differenziale". Tale approccio è simile a quello standard, a cui viene aggiunto uno step precedente di preparazione del dataset e uno step finale per ottenere la nuova forma. Presi due set di forme in corrispondenza biunivoca, definiti dagli indici i e i' , si calcola per ogni coppia di forme il vettore differenza punto a punto:

$$\vec{v}_i = ((v_{1x}^i - v_{1x}^{i'}, v_{1y}^i - v_{1y}^{i'}, v_{1z}^i - v_{1z}^{i'}), \dots, (v_{Nx}^i - v_{Nx}^{i'}, v_{Ny}^i - v_{Ny}^{i'}, v_{Nz}^i - v_{Nz}^{i'}))$$

Una volta ottenuto questo insieme di vettori delle differenze, è necessario disporli come in Equazione 3.10, e creare una matrice da dare in input alla PCA. Dopo aver seguito tutti i passaggi di calcolo della matrice di covarianza, autovettori e autovalori, ciò che si ottiene è un modello che rappresenta le variazioni di forma del dataset, ovvero rappresenta le deformazioni che legano tutte le forme confrontate.

Scelti un insieme di valori da assegnare ai parametri α_i , da applicare in Equazione 3.11, si ottiene un vettore differenza v . Tale vettore viene quindi sommato a una forma base, punto a punto, ottenendo una forma probabilistica derivante da ciò che il modello ha imparato.

3.2 Framework e librerie

3.2.1 Visualization Toolkit (VTK)

Il *Visualization Toolkit* (VTK, Ktiware³) è una *Application Programming Interface* (API) per l'elaborazione, la visualizzazione e l'analisi di dati scientifici. Tale software è open-source, con licenza BSD a tre clausole e *cross-platform*, supporta infatti sia sistemi operativi Unix che Windows [32]. Nato nel 1993, fu sviluppato per offrire metodi di visualizzazione e analisi riproducibili per dati scientifici, in domini come imaging medico, chimica, dati cosmologici, fluidodinamica computazionale e analisi agli elementi finiti: è considerato dunque un sistema *general purpose*.

Architettura

La libreria VTK si compone di due parti principali: una parte *core* compilata (implementata in C++) e un layer interpretato generato automaticamente, il quale genera *binding* del software in Java, Python e Tcl [33]. La parte core è formata da più di 1500 classi, suddivise in moduli, che incapsulano un vasto insieme di

³<https://vtk.org/>

funzionalità, algoritmi e strutture dati. Mentre la parte core fornisce alte prestazioni, velocità ed efficienza, il layer interpretato fornisce flessibilità ed estensibilità. VTK è infatti indipendente da interfaccia grafica, che può essere implementata dallo sviluppatore, e favorisce l'estensibilità delle sue funzioni e l'integrazione con altri software [33].

VTK è orientato agli oggetti: le astrazioni su cui si basa sono organizzate in una complessa gerarchia di classi, derivanti tutte da `VtkObject`. Tra gli oggetti fondamentali su cui di basa VTK vi sono:

- **Data Object** (`VtkDataObject`): rappresentazione dei dati. Sono disponibili numerose tipologie di data object, come alberi, grafi e *data set* (`VtkDataSet`), i quali rappresentano dati spaziali come griglie uniformi, griglie strutturate e non strutturate e dati non strutturati (fig. 3.20).

I dataset consistono di (fig. 3.21):

- Celle: rappresentano entità geometriche che definiscono volumi nello spazio e la topologia dell'oggetto;
- Punti: definiscono i vertici associati alle celle e rappresentano la geometria dell'oggetto tramite coordinate spaziali.

A celle e punti possono essere associati valori scalari o vettoriali rappresentanti quantità (pressione, temperatura, velocità, ecc) [34]. I data object espongono metodi di accesso ai dati che incapsulano.

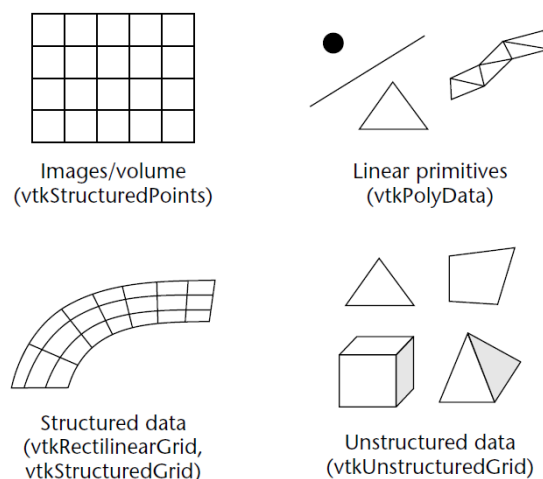


Figura 3.20: Esempi di data object di VTK. [33]

- **Process Object** (`VtkProcessObject`): sono filtri che applicati ai data object ne producono di nuovi, modificati. Molti dei filtri operano su un solo

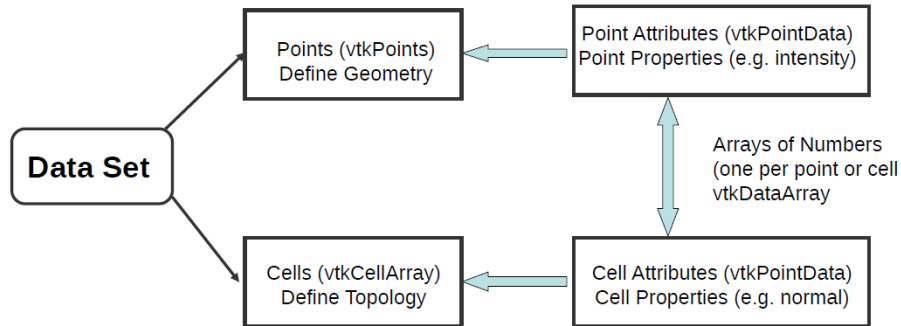


Figura 3.21: Struttura del dataset in VTK. [35]

tipo di dato, mentre altri prendono un certo tipo in input e ne danno uno differente in output. I process object si suddividono in tre categorie:

- *Source*: non hanno alcun input, ma producono uno o più oggetti;
- *Filter*: accettano uno o più input e generano uno o più output;
- *Mapper*: gestiscono la terminazione della gestione del dato, preparandolo per la visualizzazione o scrittura su disco [33].

VTK Pipeline

Il processamento dati in VTK segue il paradigma *data-flow*, ovvero secondo cui i dati fluiscono attraverso una serie di algoritmi di trasformazione. Tali algoritmi sono concatenati a formare la *pipeline*, detta anche *visualization network* [34]. Essa viene ottenuta dalla connessione di object data e process data, come fossero tanti “blocchetti” collegati a formare delle reti, che possono includere anche *branch* e cicli (fig. 3.22).

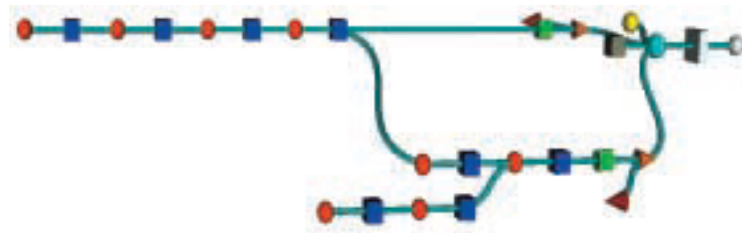


Figura 3.22: Esempio schematico della VTK visualization network: formata da process object (sfere rosse), data object (cubi blu) e proprietà, renderes e render window. [33]

Nella definizione della pipeline è importante rispettare la coerenza dei tipi di input e output delle connessioni. Il *type-checking* viene fatto a tempo di compilazione o a runtime a seconda del linguaggio usato per l'implementazione [33].

Si possono individuare due sotto-processi della pipeline: la *process pipeline* e la *graphical pipeline* (fig. 3.23). La prima, formata da process object, prende in input i dati e li mappa in primitive grafiche. Ad essa appartengono tre tipi di oggetti [36]:

- *Source*: forniscono l'input a tutta la pipeline. L'input può essere generato, in caso di primitive rappresentabili matematicamente (sfere, coni, ecc), oppure letto da un file (STL, PLY, ecc);
- *Filter*: elaborano i dati di input grezzi applicando trasformazioni di vario tipo o traducendo i dati in formati adatti alla visualizzazione;
- *Sink*: gestiscono la fine della pipeline, scrivendo i dati su file oppure associando un *mapper* per convertire un set di dati in un set di istruzioni di rendering.

La graphical pipeline ha lo scopo di eseguire il rendering degli oggetti per visualizzarli a schermo. Il flusso inizia dal momento in cui si connette un mapper al dato ed è formato dai seguenti elementi:

- *Actor*: rappresentazione dell'oggetto nel rendering. Ad esso possono essere associate delle *proprietà (Props)*, che definiscono come l'oggetto dovrà essere visualizzato (scala, colore, orientazione, texture, ecc);
- *Renderer*: controlla il processo di rendering degli attori istanziati;
- *Render window*: finestra nella quale i renders disegnano;
- *Window interactor*: componente che permette l'interazione dell'utente con la visualizzazione (zoom o rotazione della visuale, selezione o manipolazione di oggetti nella scena).

Una volta creata la pipeline, l'esecuzione di essa deve essere controllata attentamente. Il tipo di modello d'esecuzione usato da VTK è *demand driven*, ovvero basato su richieste di aggiornamento che vengono inviate ai segmenti della rete da eseguire [36]. Tali aggiornamenti vengono gestiti da VTK tramite un processo implicito, automatico e distribuito: esso fa sì che gli algoritmi vengano eseguiti solo in caso di cambiamento del loro stato interno o alla ricezione di nuovi dati da elaborare.

Ad ogni oggetto il sistema associa un *time stamp* che memorizza l'ultimo momento in cui l'oggetto è stato aggiornato. Gli object data e i process data possiedono time stamp aggiuntivi che sono aggiornati durante l'esecuzione della pipeline.

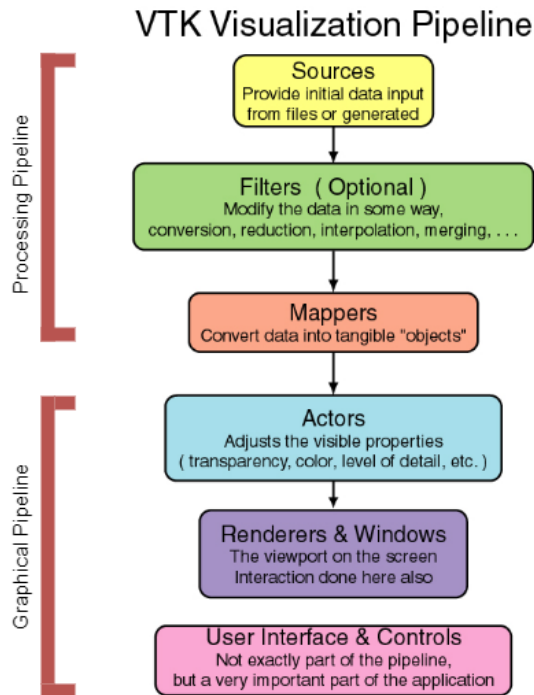


Figura 3.23: Schema della pipeline di visualizzazione in VTK. Fonte: <https://www.cs.uic.edu/~jbell/CS526/Tutorial/Tutorial.html>

La routine compara questi tempi per determinare quali oggetti non sono aggiornati e di conseguenza quale porzione della rete eseguire. Ciò fa sì che solo i segmenti della pipeline il cui stato è effettivamente cambiato si aggiornino, evitando elaborazioni inutili.

Mentre molti sistemi di visualizzazione usano controlli centralizzati, creando dei colli di bottiglia in applicazioni di grandi dimensioni, VTK utilizza un meccanismo di controllo distribuito che permette di mantenere alte le performance [33].

PyVista

Talvolta il codice Python risultante dall'utilizzo dei bindings di VTK risulta piuttosto complesso e verboso, dato che si basa su chiamate ad API C++. Pertanto, è stata creata una libreria nativa Python, PyVista⁴, che espone un'interfaccia concisa e ben documentata preservando il potente backend di visualizzazione di VTK. PyVista cerca quindi di semplificare la creazione e manipolazione di mesh e le routine di visualizzazione senza compromettere la velocità del backend C++. Vi sono nella libreria un insieme di tipi che wrappano ed espandono quelli propri di

⁴<https://pyvista-doc.readthedocs.io/en/latest/index.html>

VTK, associando ad essi metodi, attributi e filtri di trasformazione. Ciò permette di avere un maggiore controllo sugli algoritmi e sul loro funzionamento [37].

PyVista agisce pertanto come modulo di supporto all'utilizzo di VTK, ma dal momento che non tutte le funzionalità di VTK sono state wrappate, PyVista non può essere utilizzato in sostituzione, ma solo come ausilio all'uso di VTK.

Integrazione con PyQt

Come detto in precedenza, VTK non fornisce alcun modulo di interfaccia grafica ma lascia spazio a varie soluzioni di integrazione. Una di esse è l'utilizzo della libreria PyQt, binding Python del framework originale C++ Qt⁵.

Qt è un framework *cross-platform* che permette di creare interfacce basate su widget (`QWidget`) posizionati all'interno di specifici layout (`QLayout`). È un framework molto utilizzato, anche per sviluppo di applicazioni professionali, e dotato di una documentazione completa.

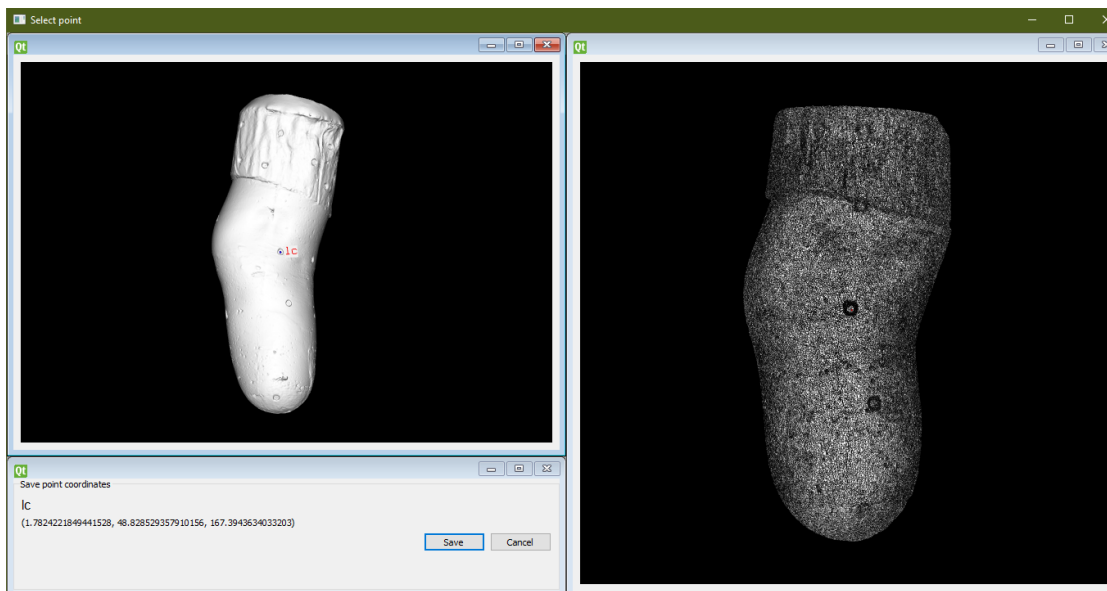


Figura 3.24: Esempio di interfaccia sviluppata tramite PyQt e VTK.

Le applicazioni GUI si basano sul modello *event-driven*: il flusso del programma è determinato dal verificarsi di eventi, che possono essere generati dall'utente o dall'applicazione stessa. In Qt tale modello è gestito per mezzo di *signal* e *slot*: ogni widget può emettere “segnali” in risposta a eventi; tali segnali sono quindi collegati a “slot”, che corrispondono a chiamate a funzione. Uno slot è pertanto

⁵<https://www.qt.io/>

eseguito quando il segnale ad esso collegato viene emesso. Nel caso di un bottone, ad esempio, il segnale di click viene connesso alla relativa funzione da chiamare.

Qt è supportato nativamente da VTK: l'integrazione è possibile grazie all'uso della classe `QVTKRenderWindowInteractor`, appartenente ai moduli di VTK, che permette di eseguire il rendering di oggetti VTK all'interno dell'interfaccia generata da Qt. Lo stesso avviene usando i relativi binding Python.

Un esempio di interfaccia realizzata integrando PyQt e VTK è riportato in Figura 3.24.

3.2.2 AmpScan

AmpScan⁶ è una libreria Python open-source che offre strumenti e funzionalità per analisi e comparazione di scansioni di dispositivi protesici e ortesici. Il software si basa su un insieme di moduli che espongono varie funzionalità (fig. 3.25):

- **AmpObject**: è l'oggetto core della libreria, che incapsula l'oggetto digitalizzato. Contiene informazioni su vertici, facce e normali alla superficie dell'oggetto ed espone metodi di salvataggio e trasformazione dei dati;
- **Alignment**: presi due `AmpObject`, applica una trasformazione rigida a uno di essi per minimizzare la distanza spaziale tra i due. Ciò è ottenuto tramite l'algoritmo ICP;
- **Registration**: presi due `AmpObject` esegue una trasformazione non rigida per allineare le due, tramite il metodo *point-to-plane* [38];
- **Analysis**: modulo per l'estrazione di informazioni geometriche di forma, come volume e aree e perimetri delle sezioni;
- **GUI**: modulo che abilita la visualizzazione di `AmpObject` nella stessa finestra, dando accesso ai metodi di allineamento e registrazione [39].

⁶<https://ampscan.readthedocs.io/en/latest/>



Figura 3.25: Flusso dei processi gestiti da AmpScan. [39]

3.2.3 MeshLab

MeshLab⁷ è un software open source per l'elaborazione di mesh 3D [40]. Esso dispone di un vasto insieme di strumenti per la modifica, pulizia, riparazione, analisi e conversione di modelli 3D. Tale strumento è particolarmente indicato per gestire mesh di grandi dimensioni e non strutturate (*raw data*), che possono derivare da digitalizzazioni 3D.

MeshLab è disponibile su varie piattaforme, come Linux, Mac OS e Windows, ed è in grado di importare/esportare file in vari formati, tra cui PLY, STL, OBJ e 3DS. Permette, inoltre, la gestione di nuvole di punti (*point cloud*), supportate nei formati PLY e OBJ. Le funzionalità di elaborazione della mesh attualmente fornite da MeshLab sono molte, tra cui sono di nostro interesse:

- Strumenti di selezione ed eliminazione interattiva di porzioni di mesh;
- Filtri per pulizia della mesh;
- Filtri di misurazione e taglio;
- Strumenti di allineamento e registrazione di mesh;
- Filtri per la generazione di mappe distanza.

⁷<https://www.meshlab.net/>

Il software è stato utilizzato come riferimento per la validazione di alcune delle funzionalità della libreria implementata, principalmente per le funzioni di *outcome* (sezione 3.1.7). La verifica consiste in confronti degli output qualitativi e quantitativi ottenuti con entrambi i metodi di analisi.

Inoltre, recentemente, da parte degli sviluppatori di MeshLab, è stato rilasciato PyMeshLab⁸, ovvero una libreria Python per l'elaborazione batch di mesh utilizzando i filtri MeshLab. Essa è stata integrata all'interno della libreria realizzata, in supporto ad alcuni processi di elaborazione.

⁸<https://pymeshlab.readthedocs.io/en/latest/index.html>

Capitolo 4

Risultati

4.1 Prototipo realizzato

L'implementazione del software è stata realizzata utilizzando il linguaggio Python (versione 3.8¹) e seguendo alcuni criteri di qualità:

1. *Semplicità di installazione*: installazione in modo facile e veloce delle dipendenze necessarie ad accedere alle funzionalità della libreria. Ciò è stato possibile grazie all'utilizzo del *package-manager-system* di Python, **pip**²;
2. *Modularità*: organizzazione del codice in moduli ben separati e indipendenti, per una migliore manutenibilità ed estensibilità delle funzionalità, perseguendo il principio della *separation of concerns*;
3. *Usabilità*: interfacce chiare ed intuitive, nonché esteticamente accattivanti e di facile navigazione;
4. *Documentazione*: corredare le funzioni esposte dalla libreria con apposita documentazione, che ne spieghi input, output ed eventuali algoritmi usati internamente. Ciò sarà utile specialmente alla comprensione, manutenzione ed eventuale espansione della libreria in futuro.

Per rispondere al requisito 2, la libreria è stata organizzata in moduli, in base alle funzionalità a cui rispondono. L'organizzazione dei moduli è riportata in Figura 4.1 ed è conforme con i blocchi del flusso di lavoro precedentemente identificati.

Per ogni *macro-funzionalità* è stata realizzata un'interfaccia, allo scopo di creare piccole applicazioni direttamente utilizzabili dall'utente, anche non esperto (requisito 3), permettendo l'assegnamento guidato dei parametri necessari e la visualizzazione degli output.

¹<https://www.python.org/downloads/release/python-380/>

²<https://pypi.org/project/pip/>

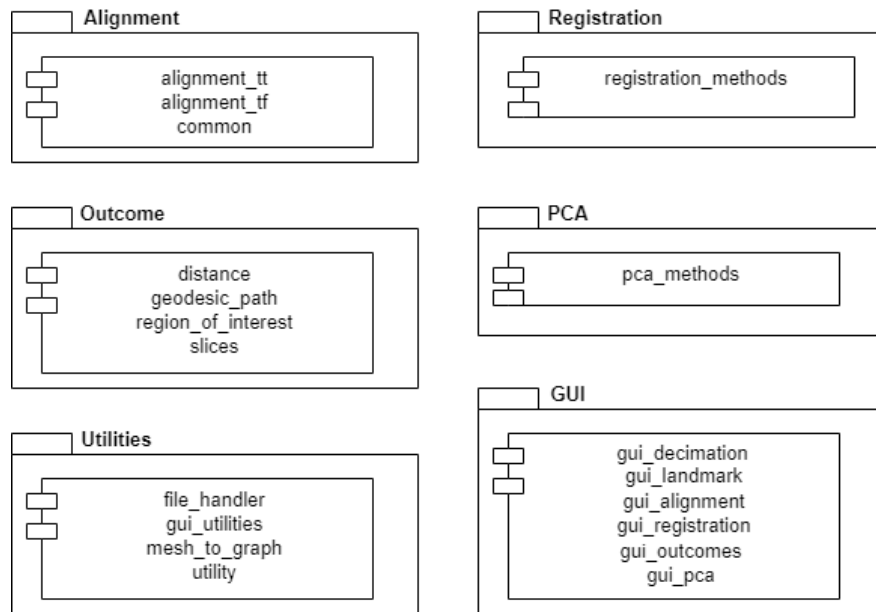


Figura 4.1: Diagramma dei moduli del software.

Tali applicazioni rispondono alle seguenti funzionalità:

1. Decimazione della mesh;
2. Selezione dei landmark secondo varie modalità;
3. Allineamento della scansione secondo un metodo scelto;
4. Registrazione di due scansioni tramite algoritmi e parametri scelti da utente;
5. Creazione della mappa distanza 3D date due mesh tramite la metrica scelta da utente;
6. Applicazione a N mappe distanza della stessa scala colore, per confronto inter-mappa;
7. Estrazione di sezioni in base a vari metodi e calcolo delle misure geometriche di esse;
8. Estrazione zona di interesse e analisi dei parametri di qualità associati ad essa;
9. Generazione del cammino geodetico tra punti della mesh, calcolo della lunghezza ed estrazione di punti a determinate percentuali del percorso;

10. Generazione del file .pts necessario al software di morphing, date due mesh;
11. Applicazione della PCA a un insieme di mesh;
12. Applicazione della PCA differenziale dato un insieme di differenze uno a uno tra due gruppi di mesh.

In Figura 4.2 è rappresentato lo *use-case diagram* dell'applicazione, che mostra l'interazione dell'utente con il sistema, seguendo il flusso di lavoro. In esso, sono rappresentate dello stesso colore attività che possono essere realizzate tramite funzioni che appartengono allo stesso modulo.

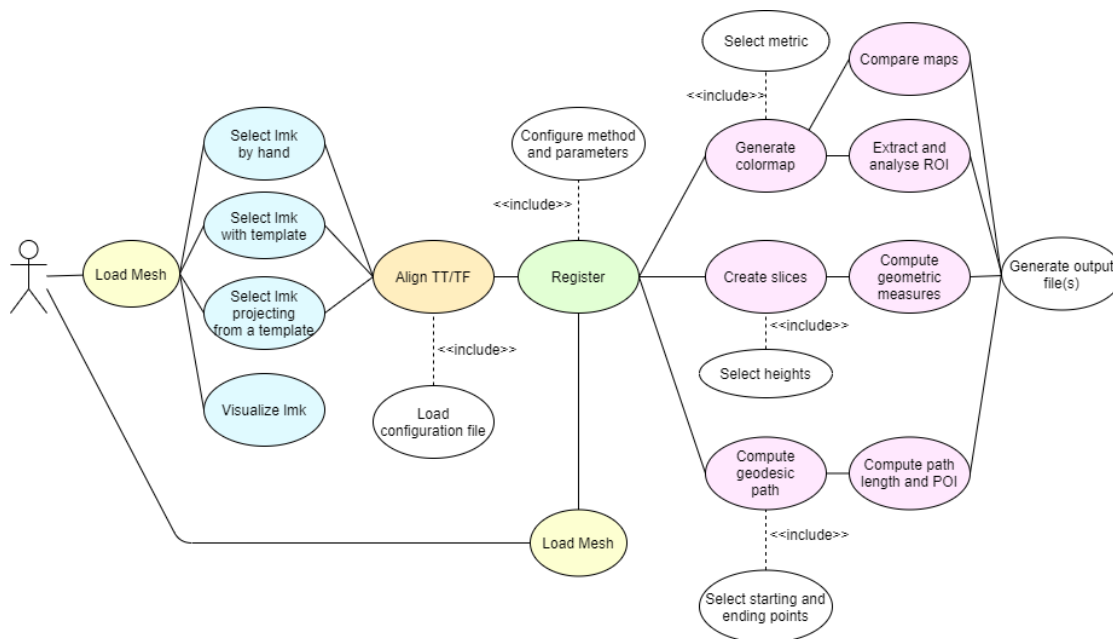


Figura 4.2: Use case diagram dell'applicazione.

4.1.1 Implementazione algoritmo registrazione Sanders

Si è realizzata un'implementazione dell'algoritmo ideato da Sanders et al. (sezione 3.1.5) tramite l'espansione dei metodi di allineamento esposti dalla libreria AmpScan (sezione 3.2.2).

Sono state introdotte alcune differenze rispetto alla versione originale, elencate in Tabella 4.1.

Dal momento che si è scelto, per semplicità, di non eseguire l'interpolazione, si è dovuto per ovvi motivi utilizzare un diverso metodo di determinazione delle corrispondenze tra punti. A causa di ciò, la metrica usata per calcolo dell'errore

	Sanders	Nostra implementazione
Pre-processing punti	Interpolazione B-spline	Nessuna
Correlazioni	Campionamento su stessa griglia	Nearest Neighbor (k-d tree)
Componente errore	Radial Error	RE-INAIL Error
Funzione ottimizzazione	<i>fmincon</i> (Matlab)	<i>minimize</i> (scipy)

Tabella 4.1: Differenze tra la versione originale dell’algoritmo e la nostra implementazione.

utilizzato nell’algoritmo originale si è rilevato non adeguato alla versione implementata. Esso è valido infatti per punti che si trovano allo stesso livello dell’asse longitudinale e differiscono solo per le coordinate x e y , per diretta conseguenza del campionamento. Pertanto si è elaborata una metrica d’errore differente, che calcola lo scostamento tra punti definiti omologhi, proiettando il vettore differenza tra i due punti sulla normale al punto sulla superficie reference (eq. 3.6).

Date due superfici R ed M , i relativi vertici sono definiti da V^R e V^M e le normali da N^R ed N^M . Sulla base di ciò, viene definita la funzione d’ottimizzazione f , dove si indicano con stesso indice i i punti omologhi:

$$MeanError = mean|(v_i^M - v_i^R) \cdot n_i^R| \quad (4.1)$$

$$NormalSimilarity = mean(\tanh^{-1}(n_i^R \cdot n_i^M - 10^{-7})) \quad (4.2)$$

$$f = ErrorWeight \times MeanError - SimilarityWeight \times NormalSimilarity \quad (4.3)$$

In Equazione 4.1 viene utilizzato il modulo del prodotto scalare per evitare che differenze positive e negative si annullino a vicenda nel computare la media.

Come ultima differenza, dal momento che le implementazioni sono state eseguite in linguaggi diversi, sono state utilizzate funzioni d’ottimizzazione differenti. I restanti passaggi dell’algoritmo sono stati eseguiti come descritti nell’articolo [25].

I pesi *ErrorWeight* e *SimilarityWeight* sono posti di default agli stessi valori utilizzati dalla Sanders (0 e 1 per il primo set di iterazioni e 0,8 e 0,2 per il secondo step). Successivamente ad un confronto con la University of Washington Sanders Lab, è scaturito che l’applicazione dei pesi dell’algoritmo per il secondo step come 0,2 e 0,8 sia una valida alternativa per il raggiungimento di un buon allineamento.

La valutazione del set di parametri ottimo che permetta un buon equilibrio tra le due componenti sarà studiata in uno step futuro del progetto, tramite test che

verifichino l'efficacia della registrazione in funzione dei valori associati ai pesi. É possibile modificare tali parametri arbitrariamente, tenendo presente il fatto che specificato un numero totale di iterazioni, ne verranno eseguite metà con il primo set di pesi e metà con il secondo.

L'algoritmo di registrazione implementato può essere visto come un'equivalente di ICP, che minimizza la distanza RE-INAIL tra i punti *closest*, nel caso in cui le iterazioni siano eseguite tutte con set di pesi 1 e 0, ovvero dando peso massimo alla prima componente e annullando la seconda. I passaggi dell'algoritmo sono schematizzati in pseudocodice 1.

Algorithm 1: Registrazione superfici R e M

Result: R, T
 $V^R, N^R \leftarrow$ reference vertices, reference normals;
 $V^M, N^M \leftarrow$ moving vertices, moving normals;
 initialize R, T;
for $i \leftarrow 0$ **to** *Number of iterations* **do**
 | $V^M = \text{Transform surface}(V^M, R, T)$;
 | **foreach** v_i^M **in** V^M **do**
 | | $v_i^R = \text{nearest neighbor}(v_i^M, V^R)$;
 | **end**
 | $ME = \text{mean}(|v_i^M - v_i^R| \cdot n_i^R)$;
 | $NS = \text{mean}(\tanh^{-1}(n_i^R \cdot n_i^M - 10^{-7}))$;
 | $f = ME \text{Weight} \times ME - NS \text{Weight} \times NS$;
 | R, T = minimize(f);
end
return R, T

4.2 Test e validazione

Per ognuno dei principali moduli di cui si compone la libreria sono stati eseguiti test e operazioni di validazione dei risultati, applicando le funzioni anche a dati reali.

4.2.1 Selezione landmark

L’implementazione delle funzionalità connesse alla selezione dei landmark è stata possibile attraverso l’impiego di funzionalità di VTK come `vtkPointPicker` e di PyVista come `find_closest_point`.

La selezione dei landmark può avvenire secondo tre modalità:

1. Selezione diretta dei landmark sulla superficie, con nome arbitrario dato da utente;
2. Selezione dei landmark seguendo un template, ovvero una mesh con landmark già selezionati che funge da “guida” alla selezione di quelli da determinare;
3. Ricerca dei *closest point* dei landmark selezionati su una superficie per crearne un set sulla nuova mesh. Tale operazione crea delle “proiezioni” dei punti selezionati su una superficie su un’altra.

Il file che viene generato in output è un file testuale che riporta una lista di elementi, ognuno composto dall’indice del punto della mesh corrispondente al landmark seguito dal nome associato. Un esempio è il seguente:

96688 - TT	45959 - BF
63785 - FAT	11260 - PAP
77358 - MC	4602 - PAD
35885 - MCT	21101 - A
85329 - LC	102971 - MPT
67001 - HF	49915 - MAT
18426 - PF	49615 - AF
21600 - SS	

La validazione delle funzioni di selezione può essere eseguita tramite un’opzione di visualizzazione (fig. 4.3), che permette di vedere rappresentati sulla mesh l’insieme dei landmark letti dal file corrispondente.

In applicazioni reali, la funzione di proiezione dei landmark (voce 3) viene eseguita a partire dalla scansione del positivo grezzo, sulla quale i punti di repere sono identificabili, per rilevare gli stessi punti sul modello stilizzato (dopo aver

allineato accuratamente le superfici). In quest ultimo non è possibile individuare i landmark, se non grossolanamente, dato che durante la lavorazione vengono rimossi i marker di riferimento.

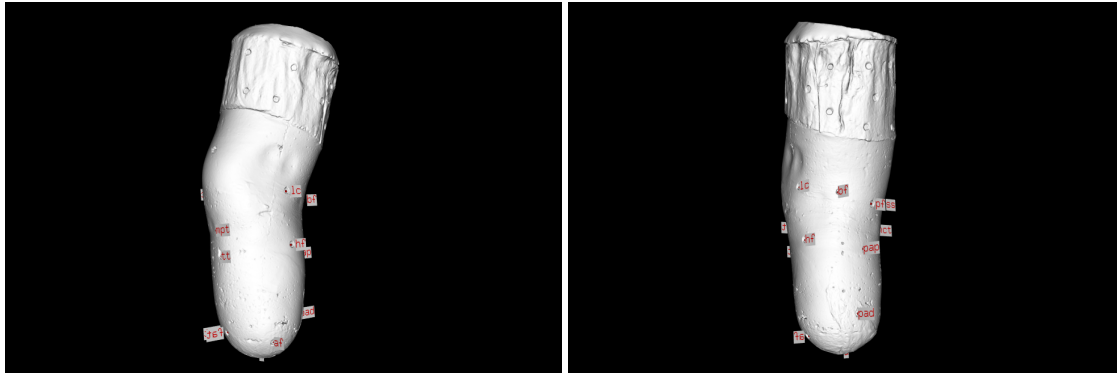


Figura 4.3: Visualizzazione dei landmark selezionati su una superficie.

4.2.2 Allineamento

L'allineamento di una scansione può essere effettuato secondo uno dei metodi elaborati e scelti come migliori. L'applicazione che esegue l'allineamento prende come input un file di configurazione, che ha la stessa struttura sia per allineamenti TT che TF. Il file deve essere in formato testuale e contenere in ordine le seguenti righe:

1. *Cut point*: specifica il punto di taglio, ovvero il livello al quale eseguire la sezione della scansione, mantenendo la parte sottostante al “taglio”;
2. *Control axis*: determina i landmark sulla base dei quali ottenere gli assi anatomici di controllo.
 - Nel caso di TT, i punti da specificare sono tre: il primo è il punto distale per cui passa l'asse longitudinale, la media degli altri due costituisce il punto prossimale di tale asse;
 - Nel caso di TF vanno specificati due punti: l'asse longitudinale passerà per il *cut point* e il primo punto specificato (considerato distale), mentre l'asse antero-posteriore per il *cut point* e il secondo punto specificato (appartenente alla superficie posteriore della mesh);
3. *Method*: metodo scelto per eseguire l'allineamento tra quelli implementati (attualmente *anatomic* o *inertial*);

4. *Landmark*: punti su cui si basa la creazione dell'asse temporaneo:
- Per TT insieme di punti usati per eseguire il *fitting* del piano ottimo da cui determinare l'asse temporaneo;
 - Per TF, nel caso di metodo inerziale, specifica l'insieme di punti usati per eseguire il *fitting* del piano ottimo da cui derivare l'asse temporaneo. In caso di scelta di metodo anatomico, i primi 5 landmark sono usati per la creazione dell'asse x temporaneo, gli ultimi 3 per l'asse y , sempre utilizzando il metodo del *plane fitting*;
5. *Origin point*: punto su cui porre l'origine del SdR. É possibile specificare un qualsiasi landmark oppure *min*, nel qual caso l'origine coinciderà con il punto della scansione che ha coordinata z minima. Ciò permetterà di avere esclusivamente valori di z positivi.

Seguono un esempio di file di configurazione per allineamento TT (a sinistra) e TF (a destra).

<pre>cut point: MPT control axis: A MPT PF method: inertial landmark: TT FAT PF origin point: A</pre>	<pre>cut point: A75 control axis: A L15 method: anatomic landmark: L70 L50 L20 M15 A A A60 L15 origin point: min</pre>
---	---

In Figura 4.4 sono riportati a confronto i due metodi scelti per l'allineamento di scansioni TT e TF. La validazione dei metodi è stata effettuata grazie alla valutazione di un gruppo di 5 tecnici ortopedici, sulla base della loro esperienza clinica.

Inoltre, per confrontare metodi di allineamento diversi sono state comparate le misure geometriche delle sezioni estratte sul database disponibile di mesh allineate. Infatti, un allineamento dell'asse longitudinale ottimale risulta dare, per la stessa mesh, aree e perimetri delle sezioni inferiori, mentre un asse z non coincidente con l'asse longitudinale crea sezioni ellittiche, con area e perimetro maggiori. Un esempio di tale confronto è riportato nei grafici in Figura 4.5.

Per avere una valutazione significativa di quale dei due metodi sia migliore è necessario raccogliere un insieme più ampio di quello attuale di dati a confronto, generati dai due algoritmi di allineamento, cosa che proseguirà nelle attività future del progetto. Ad oggi la preferenza è associata al metodo anatomico.

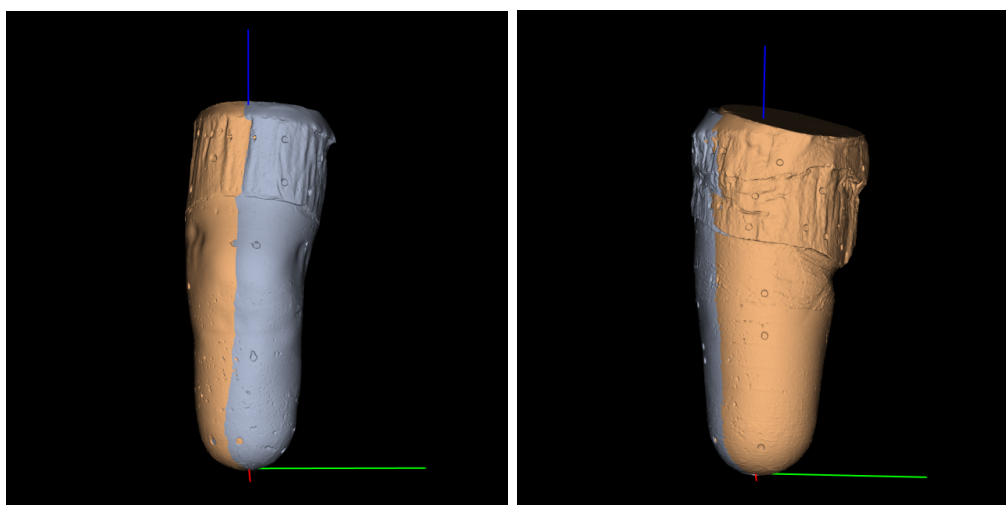


Figura 4.4: Confronto tra il metodo di allineamento inerziale (mesh rosata) e anatomico (mesh lilla). Gli assi x, y e z sono rappresentati rispettivamente in rosso, verde e blu.

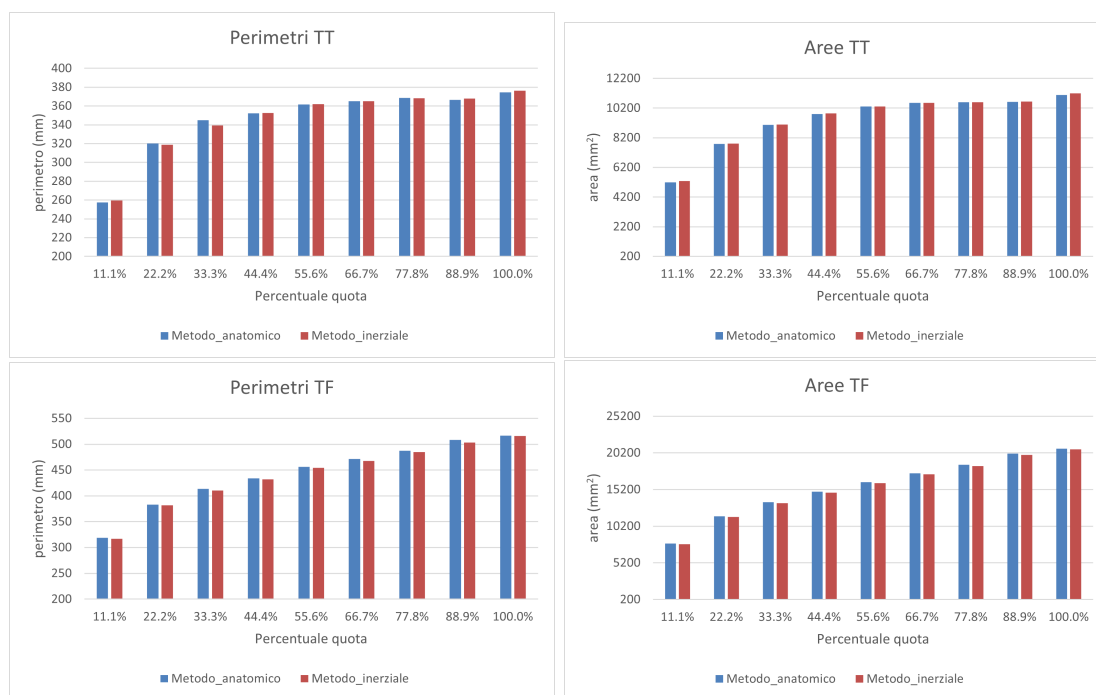


Figura 4.5: Confronto di perimetri (a sinistra) e aree (a destra) di sezioni estratte dalla stessa scansione con diverso allineamento per pazienti TT (in alto) e TF (in basso). Le sezioni sono ottenute suddividendo uniformemente la parte di scansione compresa tra due punti di repere definiti (in ascissa la percentuale di riferimento rispetto alla lunghezza della parte considerata).

4.2.3 Registrazione

Al fine di validare gli algoritmi di registrazione, sono stati creati degli appositi *unit test*, usando il framework Python `unittest`³. Tali test permettono di verificare la correttezza dell'output degli algoritmi lavorando in un ambiente controllato, ovvero nel quale la trasformazione che porta all'allineamento è nota.

A partire da una mesh, ne viene creata una seconda versione trasformando l'originale tramite una matrice di trasformazione specificata. Si ottengono così le superfici static e moving da dare in input all'algoritmo di registrazione. L'output, ovvero la trasformazione geometrica che applicata alla moving la allinea alla static, viene confrontata con la matrice di trasformazione iniziale e si valuta la coincidenza delle due, ammettendo una certa percentuale d'errore. Viene inoltre calcolato per le mesh registrate un valore d'aggregazione, in modo da valutare il risultato. Esso corrisponde al RMSE:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n \|v_i^R - v_i^M\|^2} \quad (4.4)$$

dove v_i^R e v_i^M sono i punti omologhi delle superfici R e M .

I risultati dei test in forma estesa sono riportati in Appendice A.

Registrazione basata su landmark

Per l'algoritmo di registrazione basata su landmark è stata utilizzata l'implementazione presente in VTK. La validazione di tale tipo di registrazione è stata eseguita sottoponendo la mesh originale a trasformazioni rigide, similari e affini ed estraendo le coordinate dei relativi set di landmark. Una volta ottenute le due mesh e i corrispondenti landmark, viene eseguito l'algoritmo di registrazione e confrontata la matrice di output con quella di trasformazione imposta inizialmente.

Iterative Closest Point

Per l'applicazione dell'algoritmo di registrazione ICP è stata utilizzata l'implementazione offerta da VTK. I parametri che è possibile settare sono:

- Il numero di iterazioni massime;
- I gradi di libertà ammessi (trasformazione rigida, similare o affine);
- Il tipo di distanza da minimizzare, da scegliere tra:

³<https://docs.python.org/3/library/unittest.html>

- **RMS**: coincide con Equazione 4.4;
- **Absolute value**: $\frac{1}{n} \sum_{i=1}^n \|v_i^R - v_i^M\|$
- Controllo o meno della distanza media tra due iterazioni. Nel caso di controllo, possibilità di settare il valore limite di distanza media sotto il quale considerare l'algoritmo arrivato a convergenza;
- Numero di punti campionati dal dataset per la ricerca del CP.

I parametri utilizzati per i test di ICP rigido e similare sono 200 iterazioni massime, soglia di convergenza 10^{-6} e 600 punti di campionamento. Per quel che riguarda la registrazione affine, operazione più complesso, sono stati usati come parametri 400 iterazioni massime, soglia di convergenza 10^{-6} e 800 punti di campionamento.

Thin plate spline

L'implementazione dell'algoritmo di registrazione basato sulle thin plate spline fa parte delle funzionalità offerte da VTK. L'input a tale algoritmo è costituito dalla superficie da registrare, i landmark delle superfici da allineare in corrispondenza uno a uno e il fattore di rigidità λ .

Il warping è un'operazione complessa che esegue l'allineamento di due superfici ammettendo la deformazione. L'output di tale algoritmo non è quindi una matrice di trasformazione, ma una trasformazione più complessa e imprevedibile. Di conseguenza per la registrazione TPS si è utilizzato un metodo di verifica differente dai precedenti: una volta applicata la trasformazione alla mesh static, sono stati dati in input all'algoritmo la mesh trasformata e i due set di landmark, ovvero le coordinate di essi prima e dopo la trasformazione. Una volta eseguito l'allineamento si è proceduto con il confronto delle due superfici e il calcolo di RMSE tra i punti NN, confrontando tale valore di errore con un valore soglia.

Uno dei test effettuati mira a verificare il corretto funzionamento della registrazione nel caso di modifica della topologia della mesh, per verificare che il warping ne sia indipendente.

Il valore di λ usato per eseguire i test è quello di default, ovvero 1.

Procrustes

Per gli algoritmi di analisi di Procrustes ordinaria e generalizzata si è fatto riferimento alla funzione esposta da VTK.

La validazione dell'analisi ordinaria è stata effettuata, come nel caso delle TPS, analizzando il valore dell'RMSE delle mesh allineate, poiché la matrice di trasformazione non è fornita in output.

L'analisi generalizzata è stata testata modificando la stessa mesh in modo da crearne un insieme con la stessa topologia. Come si nota in Figura 4.6, le registrazioni rigida, similare e affine allineano le mesh filtrando in sequenza gli effetti di traslazione e rotazione, scala, e in ultimo trasformazioni di *shear*.

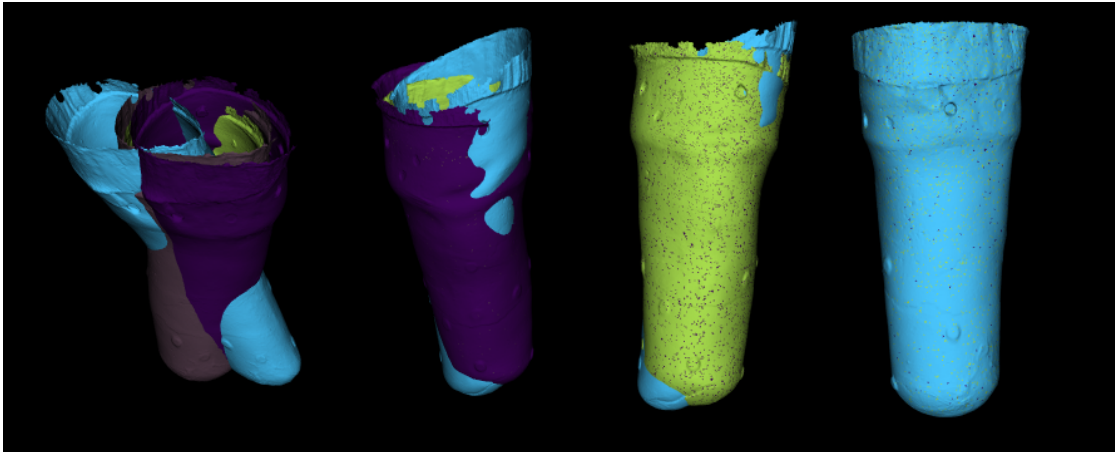


Figura 4.6: Risultato dell'analisi di Procrustes generalizzata. Da sinistra: le mesh originali, l'allineamento rigido, similare e affine.

Sanders

I test effettuati sull'algoritmo ispirato a quello ideato da Sanders è stato testato allo stesso modo di ICP. Le trasformazioni a cui sono state sottoposte le mesh sono state però limitate a rotazioni e traslazioni, data la natura dell'algoritmo, che lavora con trasformazioni rigide.

I parametri utilizzati per i test sono: 40 iterazioni e pesi delle componenti della funzione d'ottimizzare associati agli step di registrazione pari a 1 e 0 per le prime 20 iterazioni e 0,8 e 0,2 per le successive 20.

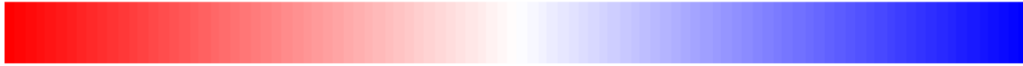
4.2.4 Outcomes

Mappe distanza

Le funzionalità correlate alla generazione delle mappe sono state implementate grazie a classi di VTK, come `vtkColorTransferFunction`, che permette di mappare una proprietà ad un colore.

Dal confronto di due mesh vengono calcolati gli scostamenti punto a punto, e ai valori di differenza ottenuti viene assegnato il corrispettivo colore. Per quanto riguarda la generazione delle mappe, secondo i parametri di qualità introdotti sono stati utilizzati due modelli di colore:

- **RWB**, mappa colore divergente dal rosso al blu passante per il bianco:



- Rosso: assegnato al whisker minimo;
- Bianco: assegnato allo zero;
- Blu: assegnato al whisker massimo;

- **WB**, mappa colore sequenziale dal bianco al blu:



- Bianco: assegnato allo zero;
- Blu: assegnato al whisker massimo;

Un'ulteriore funzionalità presente nella libreria permette di confrontare N mesh sul medesimo range di colore (blocco *compare mesh* in Figura 4.2). Una volta caricate le mesh, vengono mostrati i valori dei whiskers: essi corrispondono al minimo dei minimi e al massimo dei massimi associati alle N mesh in input. Tramite assegnazione dei valori estremi sopra ottenuti o diversi valori a scelta dell'utente vengono generate le mappe. Ciò permette un confronto multiplo sulla base delle caratteristiche dei dati al fine di poterle comparare sullo stesso intervallo. Le scale colore utilizzate per il confronto sono la RBW, sopra descritta, e la:

- **JET**, mappa colore divergente dal blu al rosso passante per il verde; in MeshLab nominata *MeshaLab RGB*:



- Blu: assegnato al whisker minimo;
- Verde: assegnato allo zero;

– Rosso: assegnato al whisker massimo.

Un esempio di elaborazione consiste nell'analisi delle scansioni del positivo grezzo e del positivo stilizzato. Da tale elaborazione si possono quantificare le modifiche eseguite da parte del TO durante la stilizzazione, permettendo l'identificazione di zone problematiche e valutandone l'entità. Tale analisi eseguita su un ampio range di modelli può portare all'identificazione di una regola di riduzione o addizione di materiale.

In Figura 4.7 si osserva la mappa relativa all'analisi, sopra citata, computata tramite la metrica RE-INAIL (eq. 3.6) e la codifica colore RWB.

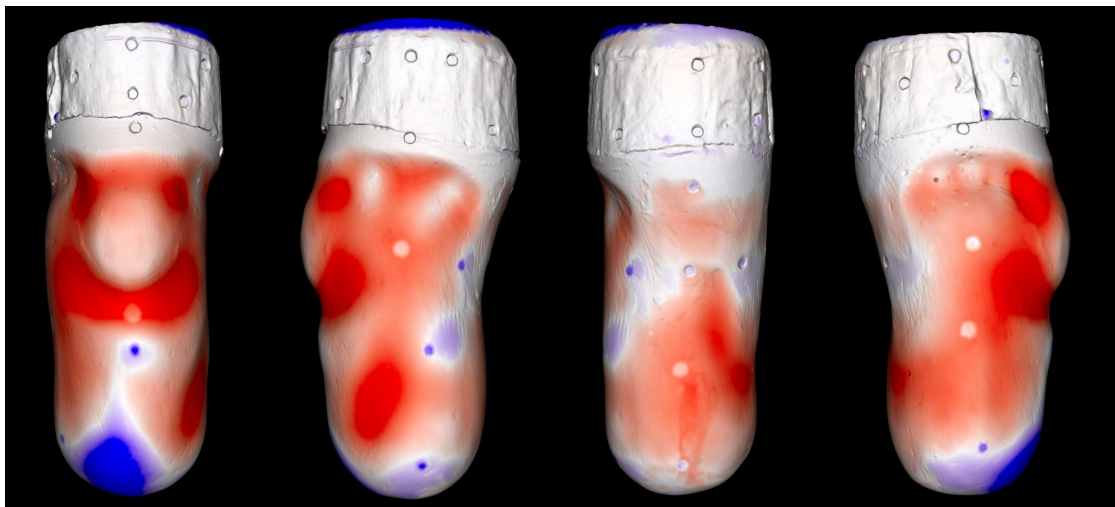


Figura 4.7: Mappa risultante dall'analisi di un positivo grezzo TT e il suo corrispondente positivo stilizzato.

La mappa distanza è stata riportata sullo stilizzato: le zone rosse corrispondono al materiale rimosso e le zone blu al materiale aggiunto durante la lavorazione per ottenere lo stilizzato.

I file di output della funzione di generazione delle mappe corrispondono alla mesh colorata, salvata in formato PLY, e ad un foglio di calcolo .xlsx contenente i valori di distanza. Ad ogni file PLY riferito alla mesh è aggiunto un commento contenente la metrica utilizzata e il valore dei baffi. Ciò permette a posteriori di identificare i parametri sulla base dei quali la mappa è stata ottenuta.

Il commento associato è sviluppato nel seguente modo (esempio riferito alla Figura 4.7):

```
ply
format binary_little_endian 1.0
element vertex 141212
```

```
comment RE/INAIL distance color map range from -4.0646 to 2.4548
...
```

La validazione della mappa distanza ottenuta viene effettuata attraverso il filtro *Quality Mapper* del software *MeshLab*, da cui si ricavano risultati pressoché uguali. I parametri connessi alla creazione della mappa sono verificati attraverso il box-plot, generato tramite *Excel*, sui parametri di qualità estratti, visibile in *Figura 4.8*. Da esso si osserva che i valori dei whiskers corrispondono a quelli computati dalla funzione implementata.

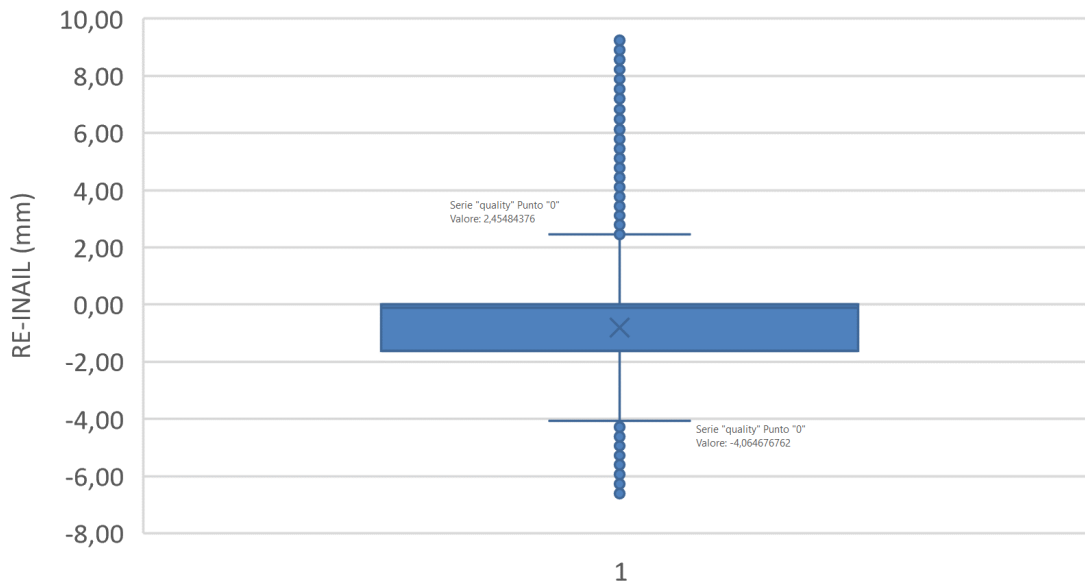


Figura 4.8: Box plot associato alla mappa in analisi.

In *Figura 4.9* possiamo visualizzare le mappe generate usando due ulteriori metriche di analisi, ovvero *SNAE* (eq. 3.8) e *Normal Similarity* (eq. 3.9).

La scala colore per la raffigurazione di tali parametri è BW. Nella parte alta della *Figura 4.9* è rappresentato lo *SNAE*: esso rappresenta l'errore derivante dall'angolo delle normali, perciò tanto è maggiore tanto più le due normali divergono. Nella parte bassa troviamo la *Normal Similarity*, che a differenza dello *SNAE* rappresenta un parametro di similarità e di conseguenza valori alti sono associati ad angoli fra le normali pressoché nulli (normali pressoché parallele).

Tali parametri di qualità sono ben visibili nelle zone circolari corrispondenti ai marcatori, in quanto riportano sullo stilizzato le zone raffiguranti i marcatori sul grezzo. Tali marcatori, dopo la rettifica, non sono più visibili sullo stilizzato, ma soltanto sul positivo grezzo come piccole dune, portando a diverse angolazioni tra le normali.

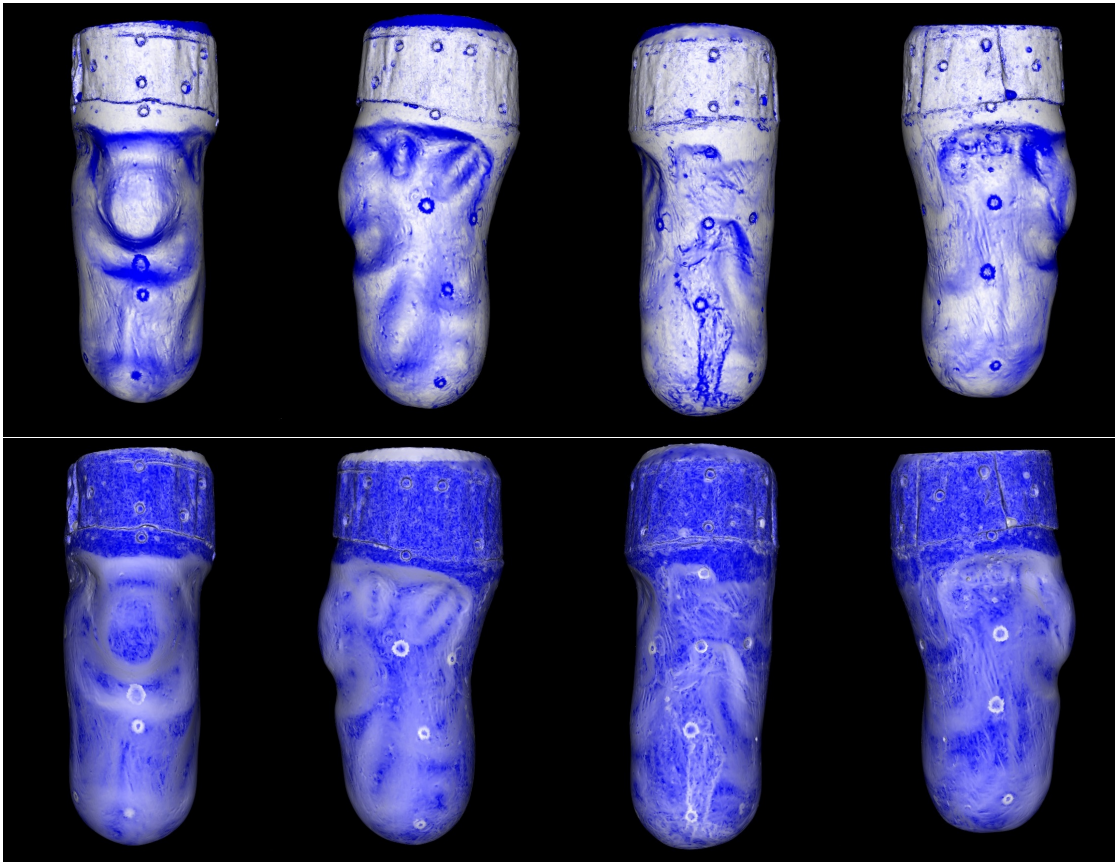


Figura 4.9: Rappresentazione dello SNAE (in alto) e della Similarity (in basso) nelle diverse viste.

Regione di Interesse

La funzionalità di estrazione e analisi della regione di interesse è stata implementata grazie a funzioni di VTK, che permettono la selezione e isolamento della zona da analizzare, insieme a funzioni delle librerie **NetworkX**⁴ e **NetworkX-query**⁵ che permettono di creare e interrogare un grafo.

L'estrazione della zona di interesse è eseguita specificando dapprima i punti che delimitano la regione da isolare. Essi vengono determinati direttamente tramite click sulla mesh, mostrata come mappa colore per individuare meglio la zona di interesse.

Un esempio di ROI estratta è riportato in Figura 4.10, la quale è stata selezionata a partire dalla mappa colore in Figura 4.7: corrisponde alla zona del marker nella parte frontale della scansione.

⁴<https://networkx.org/>

⁵<https://pypi.org/project/networkx-query/>

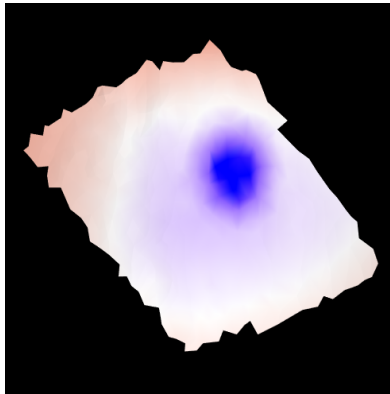


Figura 4.10: Esempio di ROI estratta da una mesh.

Per analizzare statisticamente i valori di qualità associati ai vertici della ROI estratta è possibile effettuare delle interrogazioni. Nell'esempio presentato i valori di qualità della ROI sono contenuti nell'intervallo $[-1.5, 3]$. Volendo analizzare la zona che in Figura 4.10 ha colore blu, associato a valori positivi, sono state fatte due interrogazioni: una mirata a localizzare i valori maggiori di 1,5 e una i valori maggiori di 0. Gli output visuali di tali interrogazioni sono riportati in Figura 4.11, i relativi istogrammi in Figura 4.13 e Figura 4.14.

La validazione dei risultati delle query viene effettuata tramite il confronto della distribuzione totale dei valori di qualità della ROI (fig. 4.12) con la distribuzione dei valori risultanti dalle interrogazioni. Utilizzando gli stessi bin per la creazione degli istogrammi si può notare come i valori delle occorrenze in output all'interrogazione siano gli stessi di quelli riportati nella porzione del grafico totale (fig. 4.12) identificata dalla condizione di filtro.

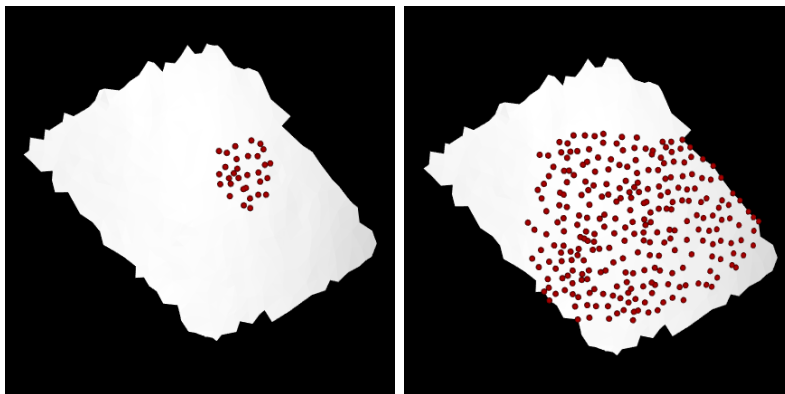


Figura 4.11: Risultato della query eseguita mantenendo valori maggiori di 1,5 (sinistra) e maggiori di 0 (destra).

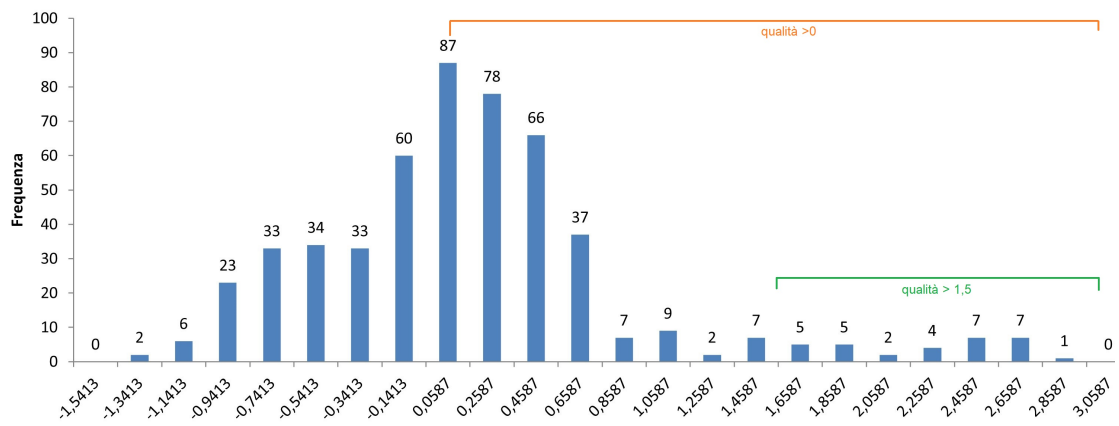


Figura 4.12: Istogramma dei valori di qualità associati ai punti della ROI, identificando le porzioni corrispondenti alla due interrogazioni.

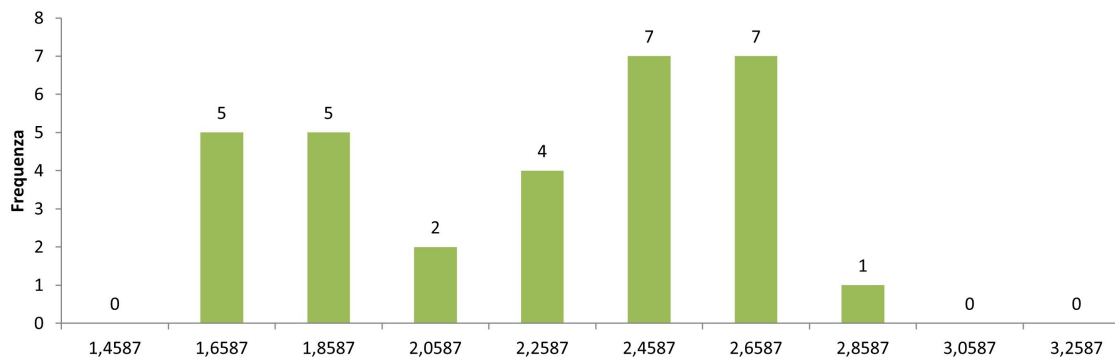


Figura 4.13: Istogramma che evidenzia i valori della ROI che rispondono alla query *quality > 1.5*, corrispondente all'intervallo verde nell'istogramma totale della ROI

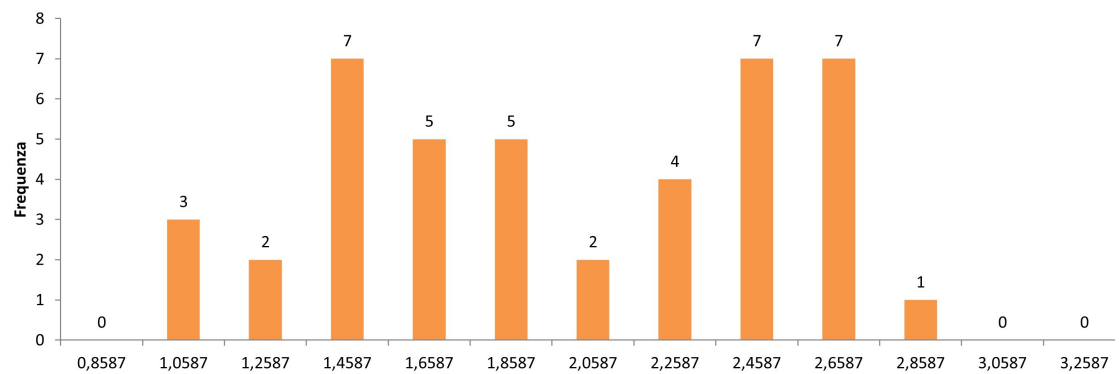


Figura 4.14: Istogramma che evidenzia i valori della ROI che rispondono alla query *quality > 0*, corrispondente all'intervallo arancione nell'istogramma totale della ROI

Sezionamento

La funzionalità di sezionamento e il corrispettivo calcolo di misure geometriche è stata implementata utilizzando funzioni offerte da VTK. Il sezionamento può essere eseguito tramite tre modalità:

1. **Basata sui landmark**, ovvero estrazione di sezioni e parametri dimensionali all'altezza di ogni landmark associato alla mesh (fig. 4.15.a);
2. **Basata sulla definizione del numero di sezioni desiderate** equidistanziate su una lunghezza definita da un landmark iniziale e uno finale (fig. 4.15.b);
3. **Basata sulla definizione dell'intervallo tra una sezione e la successiva**. Essa necessita dell'assegnazione di due landmark, rappresentanti il punto prossimale e distale da considerare per la lunghezza totale, e dell'intervallo in mm tra una sezione e l'altra (fig. 4.15.c). Per tale funzionalità viene sempre computata la sezione prossimale in quanto potrebbe non essere inclusa nelle quote identificate, dal momento che il sezionamento parte dai punti distale.

Indipendentemente dalla modalità selezionata, vista l'importanza clinica della misura della circonferenza relativa a 40 mm sopra l'apice (A+40), viene sempre eseguita la sezione in tal punto. I parametri dimensionali in output vengono allocati in file formato .xlsx, in cui:

- Ogni riga del file corrisponde ai valori associati alla stessa mesh;
- Nelle celle di tale riga seguono in ordine per ogni sezione:
 - Nome sezione, differente per ogni modalità selezionata:
 - 1 \Rightarrow *Nome del landmark*, in ordine alfabetico per permettere la corrispondenza;
 - 2 \Rightarrow *Percentuale di riferimento sulla lunghezza totale*
 - 3 \Rightarrow *Quota z*;
 - Quota z (mm), non ripetuta in caso di applicazione della terza modalità;
 - Perimetro (mm);
 - Area (mm²);
 - Volume Residuo (mm³);
 - Massima elongazione antero-posteriore (mm);
 - Massima elongazione medio-laterale (mm).

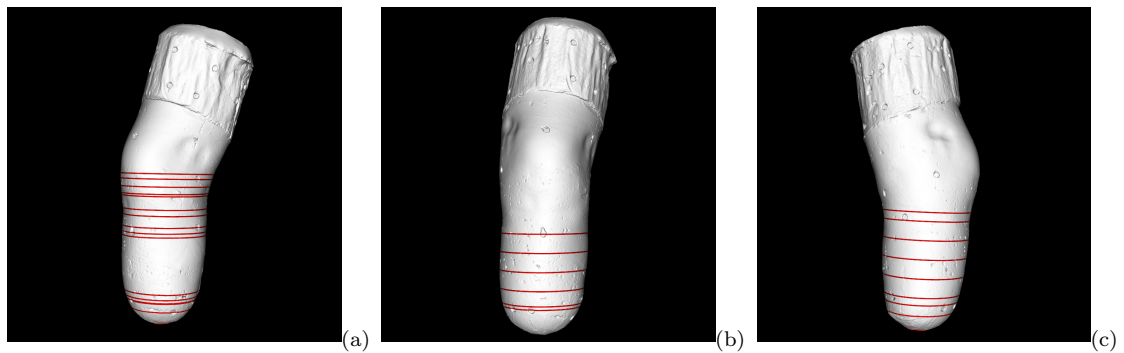


Figura 4.15: Rappresentazione delle sezioni ottenute secondo le tre modalità implementate

La verifica dei dati di outcome ottenuti si basa sul confronto dei dati tramite il software MeshLab. Tale controllo dei valori dimensionali è stato eseguito sulle sezioni di output della libreria sviluppata. Ad esse è stato applicato il filtro **Compute Geometric Measures** di MeshLab: ottenendo gli stessi valori è stata verificata la correttezza delle misurazioni.

Cammino geodetico

Per l'implementazione della ricerca del cammino geodetico si è fatto riferimento a funzioni esposte dalla libreria PyVista. La ricerca del cammino geodetico può essere effettuata in due modalità:

1. Identificazione del cammino minimo da ogni landmark, già associato alla mesh, rispetto all'apice della scansione. A titolo esemplificativo in Figura 4.16 si mostra il risultato ottenuto per una mesh TT;
2. Identificazione del cammino minimo tra due landmark scelti dall'utente. Vi è inoltre la possibilità di determinare un punto P, posizionato a una determinata percentuale del percorso data in input (illustrazione in Figura 4.17).

Ogni cammino geodetico generato viene esportato come nuvola di punti in formato PLY. Inoltre, ad essi è associato un file .xlsx all'interno di cui sono allocati i parametri di output:

- Nome degli estremi Iniziale (I) e finale (F) del cammino;
- Lunghezza del percorso \overline{FI} (mm);

Inoltre, per la seconda modalità, ai precedenti si aggiungono:

- Percentuale associata alla ricerca di P;

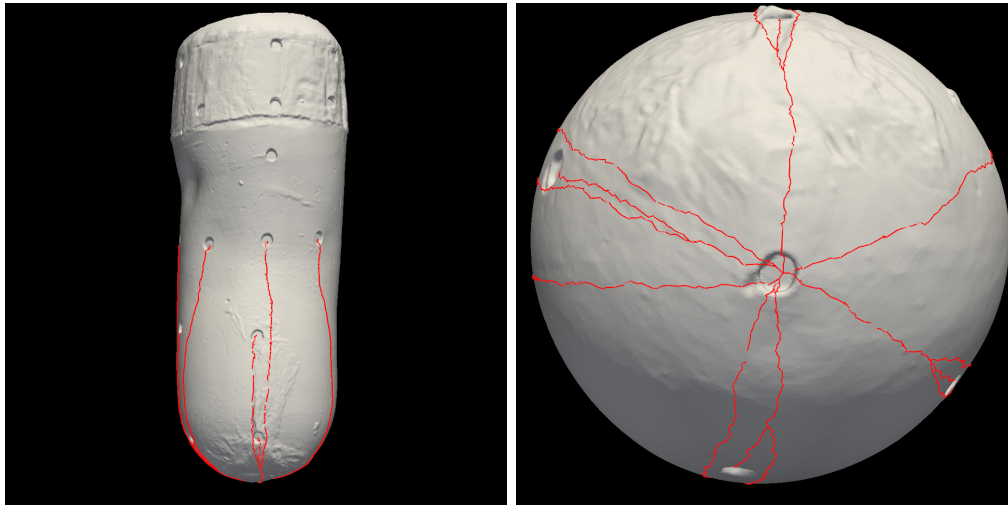


Figura 4.16: Rappresentazione dei cammini geodetici di tutti i landmark rispetto all'apice.

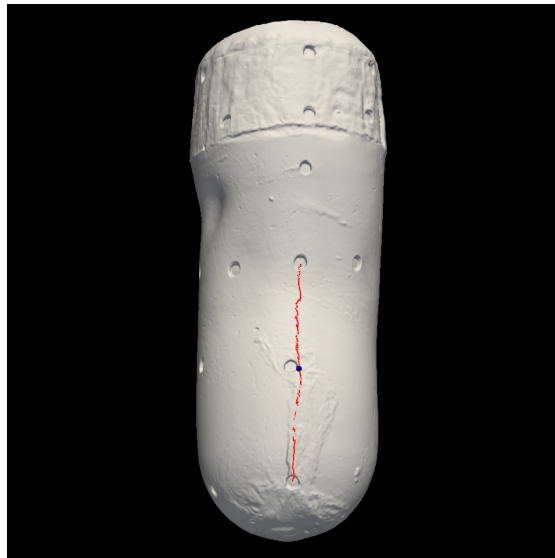


Figura 4.17: Raffigurazione del percorso geodetico fra due landmark e punto d'interrogazione.

- Indice del vertice sulla mesh corrispondente al punto P;
- Lunghezza percorso \overline{IP} (mm);
- Lunghezza percorso \overline{PF} (mm).

Per quanto riguarda percorsi estratti da differenti mesh TT e TF, essi hanno diversa rilevanza clinica. Per TF tutti i cammini sono rilevanti, in quanto la maggior parte dei punti di repere sono definiti proporzionalmente dalle lunghezze

del moncone, lungo geodetiche. Per quanto riguarda i TT, la rilevanza è minore in quanto la maggior parte dei punti di reperi sono anatomici, quindi identificati tramite palpazione.

4.2.5 PCA

Per l'implementazione degli algoritmi basati sulla PCA è stata utilizzata la funzione di decomposizione fornita da **scikit-learn**⁶.

Nell'implementazione della libreria sviluppata, una volta generata la forma media, è possibile selezionare quali componenti (autovalori) utilizzare per l'analisi. Tale selezione può essere fatta in base alla varianza spiegata legata alle componenti, ossia la quantità di informazione associata agli autovalori. Una volta scelte le componenti, è possibile assegnare ad ognuna di esse un peso, compreso tra -3 e 3, che sarà il fattore moltiplicativo di ogni autovalore (con valore 0 il relativo autovalore non sarà considerato). Tali pesi permettono di generare nuove forme applicando le modifiche legate alle relative componenti, con cambiamenti più o meno evidenti in base al peso.

In Figura 4.18 è rappresentato il diagramma dei casi d'uso relativo all'applicazione della PCA e della PCA differenziale. Si nota da essi che il "cuore" delle attività è comune ai due flussi, ma che nel caso della PCA differenziale sono aggiunte attività all'inizio e al termine (blocchi in arancione).

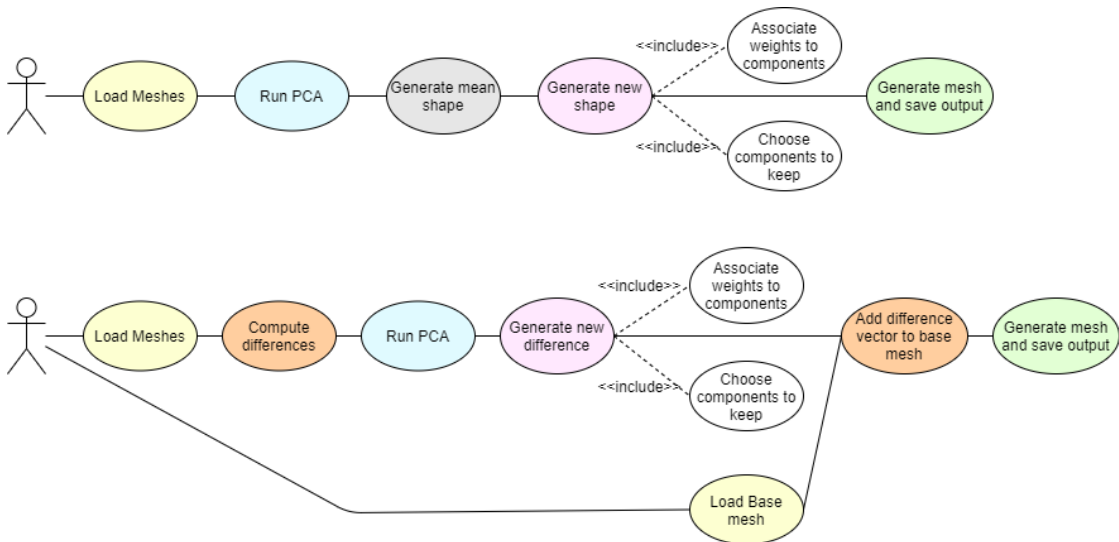


Figura 4.18: Use case diagram PCA: in alto riguardante la PCA standard, in basso la PCA differenziale.

⁶<https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA>

Di seguito sono riportati i test fatti sulla PCA, sia per quel che riguarda l'applicazione di essa nella forma standard che nel caso della PCA differenziale.

PCA standard

Per testare il funzionamento della PCA è stato utilizzato un dataset creato appositamente. Esso è stato generato a partire da un database di forme, diverse tra loro sia per caratteristiche che per topologia. Da esse ne è stata selezionata una, presa come riferimento, sulla quale tutte le altre mesh sono state “modellate”, tramite un software di morphing. Ciò ha permesso di ottenere un dataset di forme che hanno stessa topologia: tale caratteristica è fondamentale per il funzionamento della PCA. Esempi di forme appartenenti al dataset sono visibili in Figura 4.19: a sinistra è riportata la forma di riferimento e a destra quelle modellate su di essa.

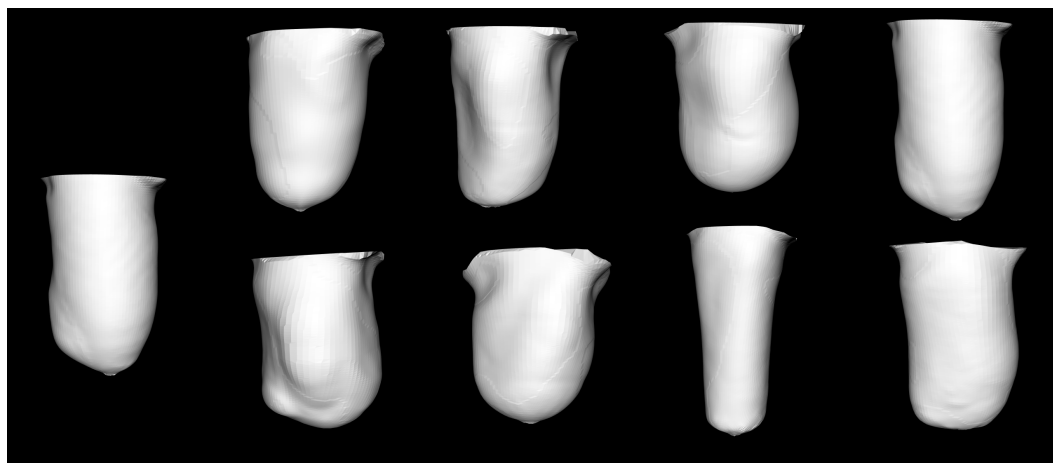


Figura 4.19: Esempi provenienti dal dataset su cui è stata testata la PCA. A sinistra la forma di riferimento, a destra le forme generate a partire da essa, con stessa topologia.

Il primo output della PCA è la forma media del dataset, visibile in Figura 4.20. Vengono successivamente scelte le componenti da considerare per i successivi passaggi. Nel caso d'esempio vengono analizzate le prime 3, alle quali corrisponde il 91% della varianza spiegata dei dati.

In Figura 4.21 sono riportate le forme generate associando pesi diversi alle varie componenti. Si nota come ad ogni componente sia associato un diverso modo di variazione della forma: la prima componente, a cui è associato l'82% della varianza spiegata, evidenzia i cambiamenti di forma più importanti, alle altre sono associati invece cambiamenti via via meno rilevanti.

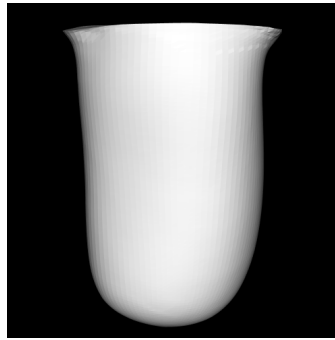


Figura 4.20: Forma media estratta tramite la PCA.



Figura 4.21: Forme generate dalla PCA a partire dal dataset di test al variare dei pesi associati alle componenti.

PCA differenziale

La PCA differenziale ha lo scopo di trovare una regola di modifica delle forme. Imparata tale regola, essa può essere poi applicata a una forma data in input, per generarne una nuova modificata.

Il dataset usato per i test è stato creato seguendo una regola nota: preso un insieme di forme, ognuna è stata modificata creando delle estrusioni in determinate zone (esempi del dataset ottenuto sono visibili in Figura 4.22). Dando tale dataset in input alla PCA, essa estrae la regola di deformazione e la applica a una forma base. Come si vede in Figura 4.23, infatti, presa una forma che non presenta

estrusioni, la PCA riesce a dare in output la stessa forma applicandole le modifiche apprese, più o meno evidenti in base ai pesi utilizzati.

La PCA differenziale può essere utilizzata su dati reali ad esempio applicandola alle differenze calcolate sui modelli grezzi e stilizzati derivanti dal lavoro dello stesso protesista. Ciò permette di analizzare ed estrarre la regola che sta alla base delle modifiche introdotte dalla stilizzazione di un TO. Tramite questo procedimento è possibile, dato un grezzo, prevederne la stilizzazione e generare diverse entità di modifica.

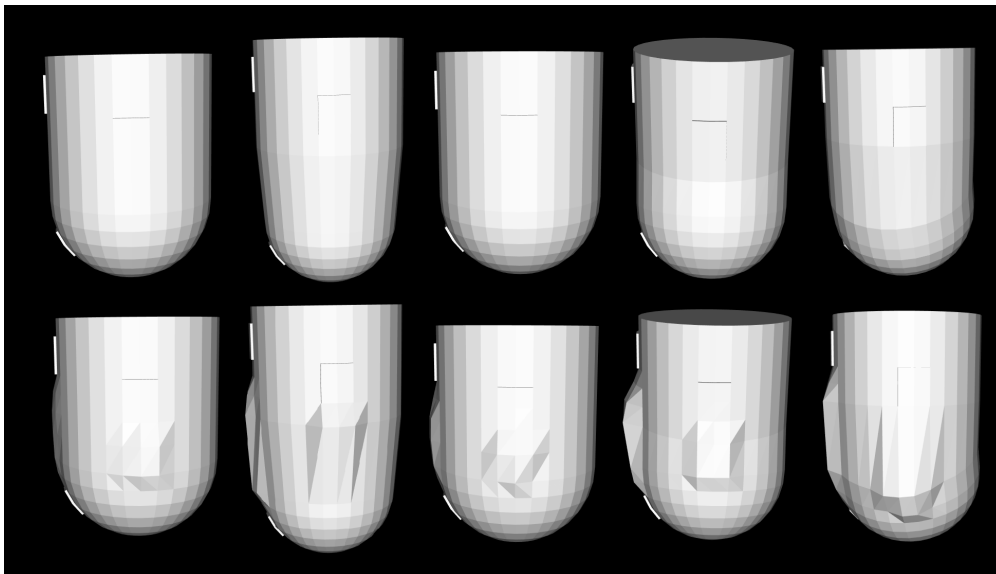


Figura 4.22: Dataset utilizzato per la PCA differenziale.

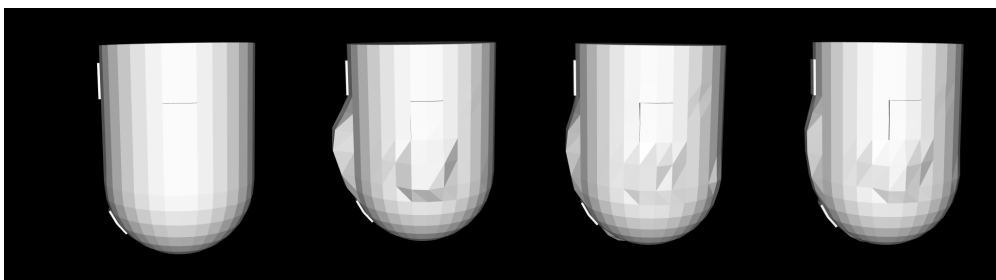


Figura 4.23: Esempi di generazione di forme dalla medesima di partenza (a sinistra) tramite la PCA differenziale.

Capitolo 5

Conclusioni e sviluppi futuri

Il presente lavoro di tesi è stato condotto al fine di realizzare un software in grado di analizzare come evolve la forma del moncone, dal calco all'invasatura finita, basandosi sulla digitalizzazione tridimensionale delle superfici. La libreria implementata risponde infatti ai requisiti di analisi ed elaborazione dei dati previsti dagli obiettivi di progetto:

- ◇ Supporta il flusso di misura e permette la gestione di formati multipli di dati;
- ◇ Integra librerie di geometria computazionale e di manipolazione di dati scientifici per soddisfare le funzionalità necessarie al caso d'uso;
- ◇ Permette la comparazione delle scansioni attraverso tecniche di valutazione dimensionale;
- ◇ Fornisce strumenti di facile uso per condurre flussi di analisi personalizzati;
- ◇ È aperta a futura espansione o modifica delle funzionalità attualmente supportate;
- ◇ Permette di estrarre dati per analisi statistiche in maniera agile facilitandone la gestione e l'ottenimento.

Il lavoro svolto sarà di ausilio al raggiungimento dell'*Obiettivo 1* dello studio sperimentale (sezione 2.2). Le funzioni di comparazione permetteranno infatti di analizzare la ripetibilità dei risultati ottenuti dalle tecniche manuale e idrostatica per la realizzazione del calco del moncone.

La validazione ad ampio spettro delle funzionalità sarà effettuata tramite l'utilizzo della libreria da parte di diversi utenti e su una più ampia varietà di dati. Ciò permetterà di identificare e risolvere eventuali limiti delle funzioni o casi d'uso non gestiti, nonché di introdurre funzionalità aggiuntive.

Tra i possibili **sviluppi futuri** del lavoro si prevedono:

- ◇ Aggiunta di un maggior numero di funzionalità interne di pre-processing del dato, come l'eliminazione di zone indesiderate o di vertici ripetuti, attualmente eseguite tramite software CAD esterni;
- ◇ Identificazione del metodo di allineamento clinicamente più adeguato tramite test su un dataset di scansioni più ampio;
- ◇ Ricerca del set ottimo di parametri legati all'algoritmo di registrazione ispirato a Sanders che permetta di ottenere un buon allineamento su un'ampia variabilità di mesh in ingresso;
- ◇ Introduzione nella libreria dell'operazione di mesh morphing, attualmente effettuata da software esterno;
- ◇ Integrazione delle singole interfacce in una unica che permetta lo svolgimento delle fasi del flusso di lavoro in cascata.

Appendice A

Test di registrazione

A.1 Registrazione basata su landmark

Sono qui evidenziati i risultati dei test eseguiti sull' algoritmo di registrazione basato su associazioni di landmark.

In Figura A.1 è riportato il risultato dell'allineamento basato su trasformazioni rigide: come si vede nella figura a sinistra, alla mesh originale sono applicate rotazioni e traslazioni.

In Figura A.2 è possibile visualizzare la registrazione similare: alla mesh d'origine vengono applicate trasformazioni rigide unite allo *scaling*.

In Figura A.3 è infine visibile il risultato della registrazione che ammette trasformazioni affini, ovvero che permette di allineare le mesh nel caso in cui siano state eseguite sulla static trasformazioni similari unite allo *shear*.

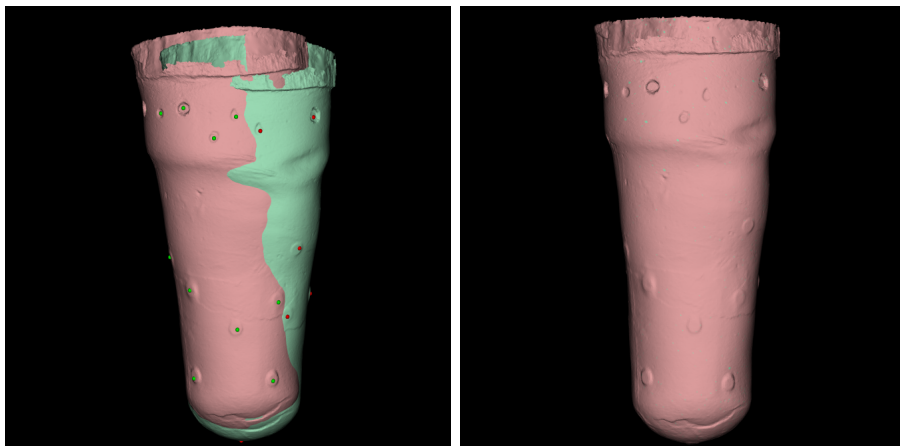


Figura A.1: Risultati visuali dei test sull'algoritmo con trasformazioni rigide.

```

[[ 0.9961947  0.          0.08715574 10.          ]
 [ 0.          1.          0.          10.          ]
 [-0.08715574 0.          0.9961947  10.          ]
 [ 0.          0.          0.          1.           ]]
[[ 9.96194698e-01  5.94912101e-09  8.71557453e-02  1.00000001e+01]
 [-6.09018689e-09  1.00000000e+00  1.35264612e-09  1.00000002e+01]
 [-8.71557453e-02 -1.87829367e-09  9.96194698e-01  1.00000008e+01]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.00000000e+00]]

```

rmse: 2.9875096389221404e-06

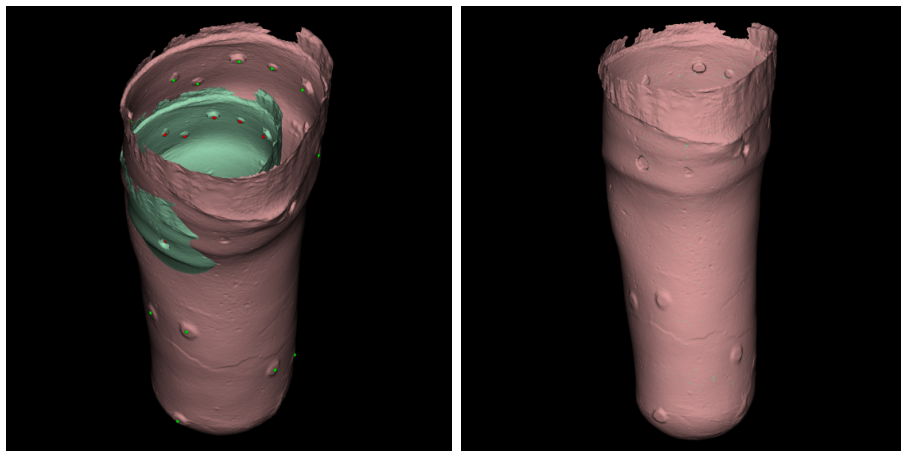


Figura A.2: Risultati visuali dei test sull' algoritmo con trasformazioni similari.

```

[[ 1.29505311  0.          0.11330247 10.          ]
 [ 0.          1.3         0.          10.          ]
 [-0.11330247 0.          1.29505311 10.          ]
 [ 0.          0.          0.          1.           ]]
[[ 1.29505311e+00 -4.41703617e-09  1.13302464e-01  9.99999987e+00]
 [ 4.55233312e-09  1.30000000e+00 -1.35359920e-09  1.00000001e+01]
 [-1.13302464e-01  1.74521031e-09  1.29505311e+00  1.00000006e+01]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.00000000e+00]]

```

rmse: 2.6379435126760107e-06

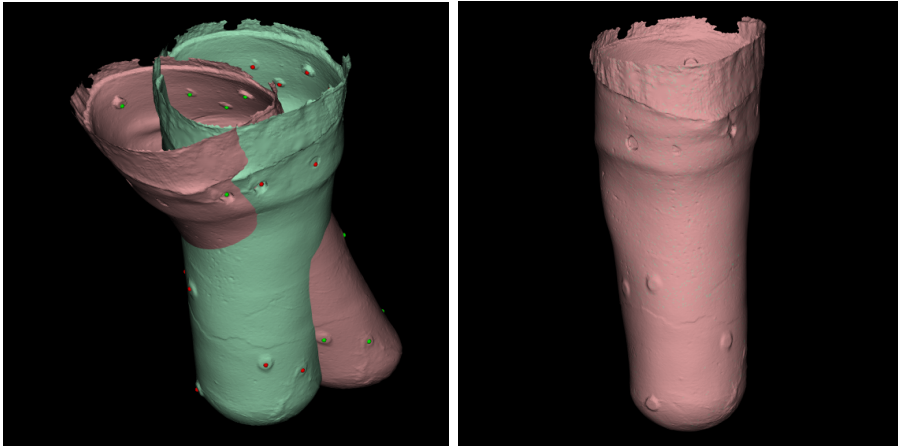


Figura A.3: Risultati visuali dei test sull'algorithmo con trasformazioni affini.

```
[[ 1.   -0.03 -0.18  0.44]
 [ 0.18  0.91 -0.41  0.3 ]
 [ 0.    0.    1.   -0.09]
 [ 0.    0.    0.    1.   ]]
[[ 9.99999988e-01 -3.00000024e-02 -1.80000000e-01  4.40000276e-01]
 [ 1.79999999e-01  9.10000007e-01 -4.10000001e-01  3.00000081e-01]
 [ 5.20257068e-09  1.58977265e-09  1.00000000e+00 -8.99971918e-02]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.00000000e+00]]
```

rmse: 3.2957083770601066e-06

A.2 Iterative Closest Point

Di seguito sono riportati gli output dei test di registrazione dell'algorithmo ICP, sia in forma grafica che in forma matriciale.

In Figura A.4 è riportato il risultato dell'allineamento basato su trasformazioni rigide: come si vede nella figura a sinistra, alla mesh originale sono applicate rotazioni e traslazioni.

In Figura A.5 è possibile visualizzare la registrazione similare: alla mesh d'origine vengono applicate trasformazioni rigide unite allo *scaling*.

In Figura A.6 è infine visibile il risultato della registrazione che ammette trasformazioni affini, ovvero che permette di allineare le mesh nel caso in cui siano state eseguite sulla static trasformazioni similari unite allo *shear*.

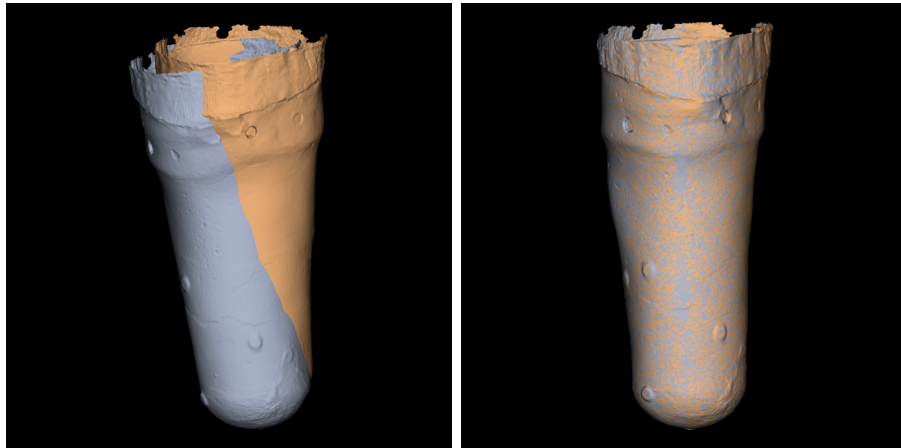


Figura A.4: Risultati visuali dei test sull'algoritmo ICP.

```

[[ 0.9961947  0.          0.08715574 10.          ]
 [ 0.          1.          0.          10.          ]
 [-0.08715574 0.          0.9961947  10.          ]
 [ 0.          0.          0.          1.           ]]
[[ 9.96194689e-01 -2.01678905e-05  8.71558416e-02  1.00000373e+01]
 [ 2.02586767e-05  1.00000000e+00 -1.57140265e-07  1.00000070e+01]
 [-8.71558416e-02  1.92220432e-06  9.96194689e-01  1.00001088e+01]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.00000000e+00]]

```

rmse: 0.0010910083443374794

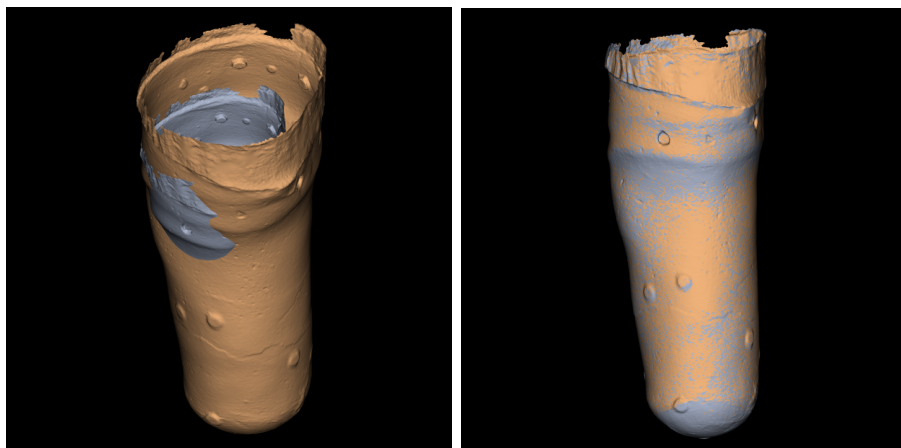


Figura A.5: Risultati visuali dei test sull'algoritmo ICP.

```

[[ 1.29505311  0.          0.11330247 10.          ]
 [ 0.          1.3        0.          10.          ]
 [-0.11330247  0.          1.29505311 10.          ]
 [ 0.          0.          0.          1.          ]]
[[ 1.29501949e+00 -1.55361374e-04  1.13293455e-01  9.99855991e+00]
 [ 1.55499359e-04  1.29996573e+00  5.20560107e-06  1.00001941e+01]
 [-1.13293455e-01  8.36614727e-06  1.29501950e+00  9.98421216e+00]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.00000000e+00]]

```

rmse: 0.003968473863411367

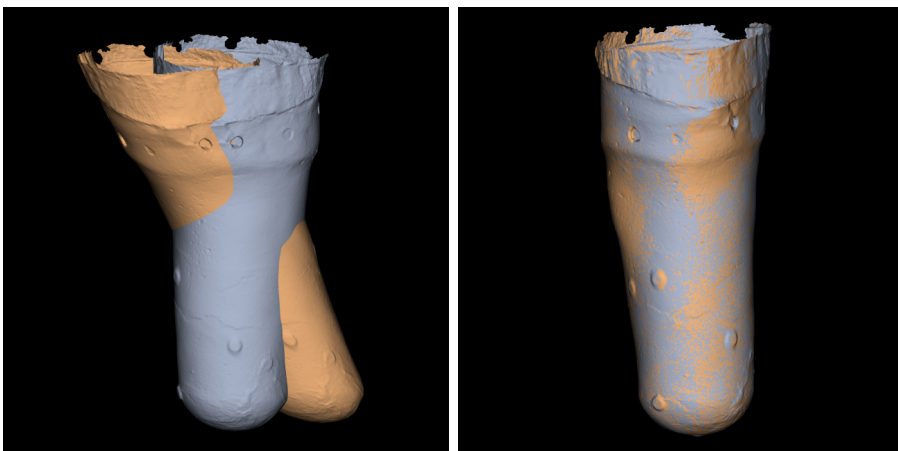


Figura A.6: Risultati visuali dei test sull'algoritmo ICP.

```

[[ 1.   -0.03 -0.18  0.44]
 [ 0.18  0.91 -0.41  0.3 ]
 [ 0.    0.    1.   -0.09]
 [ 0.    0.    0.    1.   ]]
[[ 9.99906729e-01 -2.95014557e-02 -1.80022054e-01  4.41427875e-01]
 [ 1.79354545e-01  9.10113249e-01 -4.10013788e-01  2.97648955e-01]
 [ 4.99960929e-04 -1.46130599e-05  1.00002368e+00 -8.53334678e-02]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.00000000e+00]]

```

rmse: 0.006162302757616113

A.3 Thin plate spline

Di seguito sono riportati i risultati dei test effettuati sulla registrazione con TPS.

In Figura A.7 è visibile il risultato della registrazione della mesh originale con quella sottoposta a traslazioni e rotazioni. Lo stesso risultato, sia grafico che numerico, è stato ottenuto cambiando la topologia della mesh moving.

In Figura A.8 è riportato l'output della registrazione della mesh originale con quella sottoposta a traslazioni, rotazioni e trasformazioni di scala.

La Figura A.9 mostra la registrazione della mesh originale con quella sottoposta a traslazioni, rotazioni e *shear*.

Come si vede dai valori di RMSE, che raggiunge valori nell'ordine di 10^{-6} , l'allineamento raggiunto è pressoché perfetto. Bisogna considerare il fatto che l'ambiente di test crea i due set di landmark applicando ad essi la trasformazione. In un contesto reale, l'allineamento è penalizzato dall'errore commesso nella selezione dei landmark in corrispondenza.

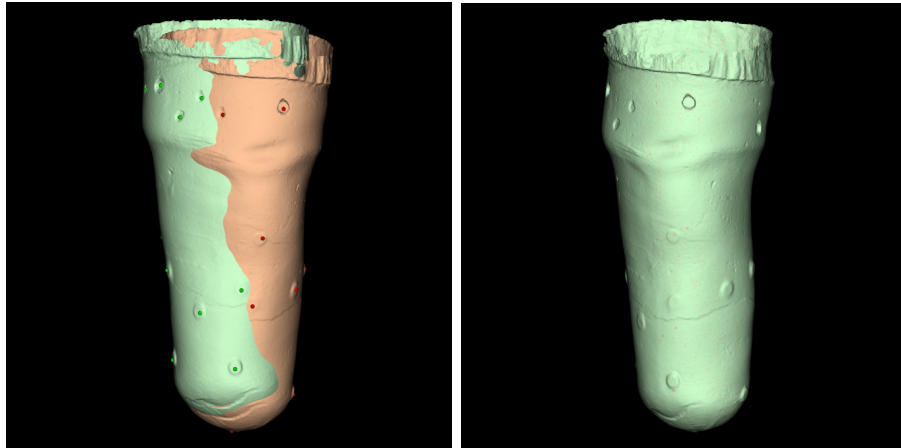


Figura A.7: Risultato grafico dell'allineamento con TPS in caso di rotazioni e traslazioni.

rmse: 4.1891157802124845e-06

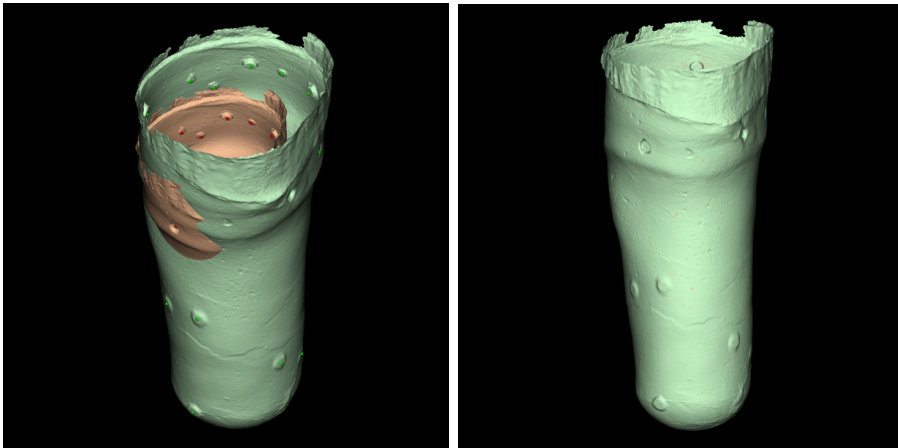


Figura A.8: Risultato grafico dell'allineamento con TPS in caso di rotazioni, traslazioni e riduzione.

rmse: 3.888269570054929e-06

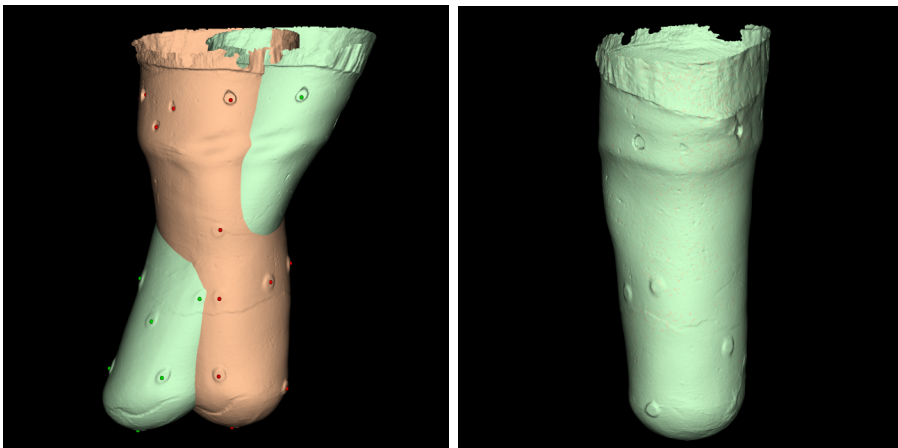


Figura A.9: Risultato grafico dell'allineamento con TPS in caso di rotazioni, traslazioni e shear.

rmse: 3.405156780993335e-06

A.4 Analisi di Procrustes ordinaria

Sono evidenziati di seguito i risultati dell'allineamento tramite algoritmo di Procrustes (analisi ordinaria).

In Figura A.10 è riportato il risultato della registrazione della mesh originale con quella sottoposta a traslazione e rotazione.

In Figura A.11 si può visualizzare la registrazione della mesh originale con quella sottoposta a traslazione, rotazione e scala.

La Figura A.12 mostra la registrazione della mesh originale con quella sottoposta a traslazione, rotazione e shear.

La Figura A.13 mostra la registrazione della mesh originale con quella sottoposta a traslazione, rotazione e modifica della topologia. È il caso che mostra i limiti dell'analisi di Procrustes, che presenta un buon funzionamento solo su mesh con stessa topologia.

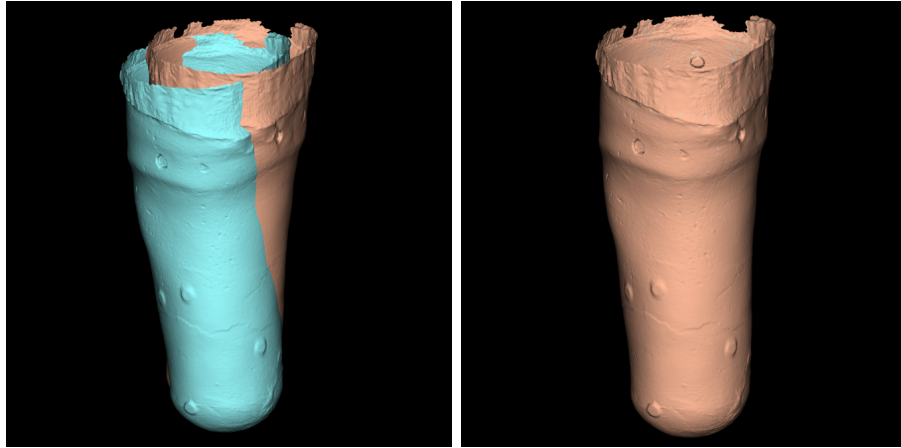


Figura A.10: Risultato grafico dell'allineamento di Procrustes in caso di traslazione e rotazione.

rmse: 1.8425612285710627e-06

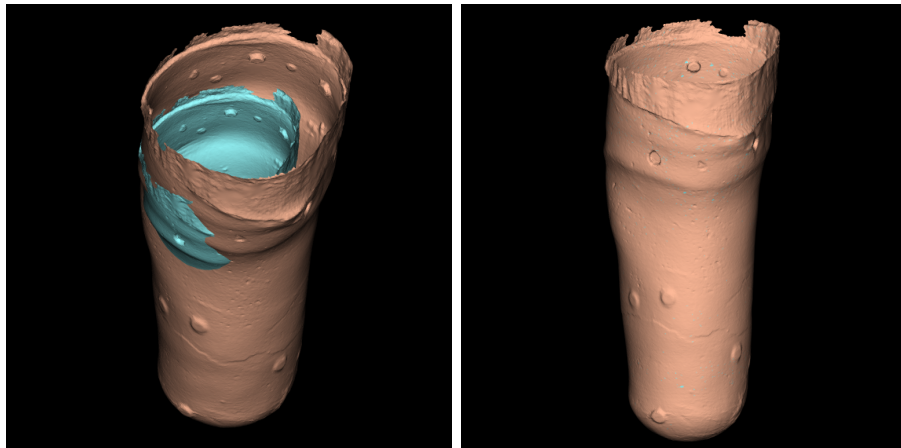


Figura A.11: Risultato grafico dell'allineamento di Procrustes in caso di traslazione, rotazione e scala.

rmse: 2.6125526637292358e-06

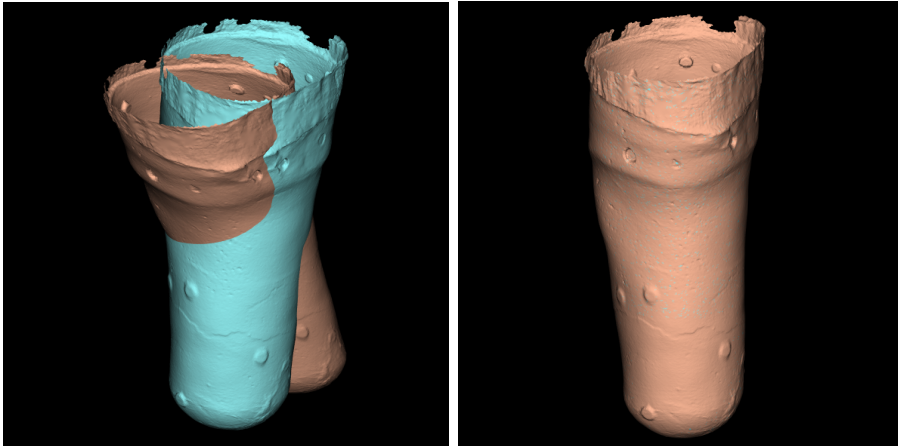


Figura A.12: Risultato grafico dell'allineamento di Procrustes in caso di traslazione, rotazione e shear.

rmse: 2.9651739596566522e-06

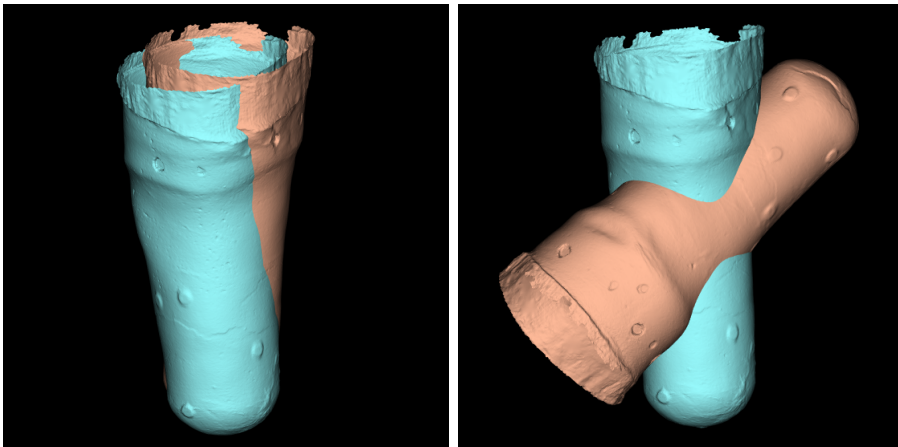


Figura A.13: Risultato grafico dell'allineamento di Procrustes in caso di traslazione, rotazione e cambio della topologia della mesh.

rmse: 48.16215621423104

A.5 Algoritmo ispirato a Sanders

Di seguito sono riportati i risultati dei test effettuati sulla registrazione eseguita con l'algoritmo implementato su ispirazione dell'allineamento di Sanders et al.

In Figura A.14 è riportato il risultato della registrazione della mesh originale con quella sottoposta a traslazione.

In Figura A.15 si può visualizzare la registrazione della mesh originale con quella sottoposta a rotazione.

La Figura A.16 mostra la registrazione della mesh originale con quella sottoposta a traslazione e rotazione.

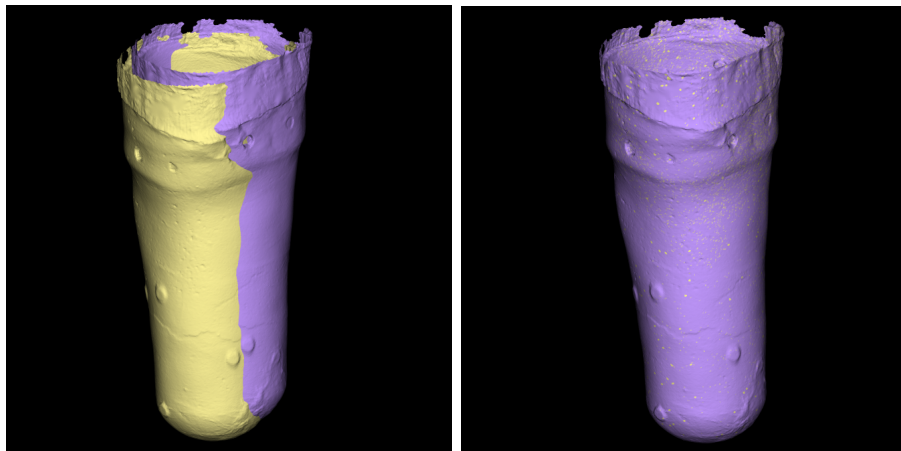


Figura A.14: Risultato grafico dell'allineamento in caso di traslazione.

```
[[ 1.  0.  0.  0.]
 [ 0.  1.  0.  0.]
 [ 0.  0.  1.  0.]
 [10. 10. 10.  1.]]
[[ 1.00000000e+00 -3.19890737e-08 -7.92693567e-09  0.00000000e+00]
 [ 3.19890737e-08  1.00000000e+00  7.25194500e-09  0.00000000e+00]
 [ 7.92693543e-09 -7.25194525e-09  1.00000000e+00  0.00000000e+00]
 [ 1.00000009e+01  9.99999961e+00  1.00000001e+01  1.00000000e+00]]
```

rmse: 7.548523523128765e-06

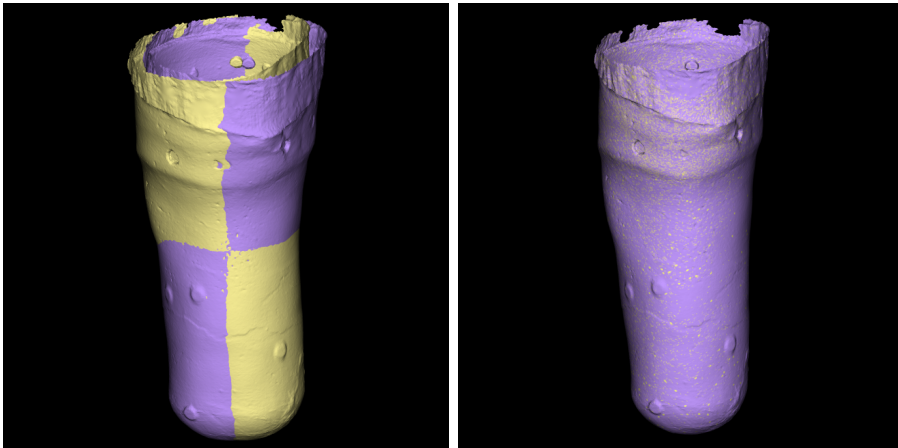


Figura A.15: Risultato grafico dell'allineamento in caso di rotazione.

```

[[ 0.9961947  0.          0.08715574  0.          ]
 [ 0.          1.          0.          0.          ]
 [-0.08715574 0.          0.9961947  0.          ]
 [ 0.          0.          0.          1.          ]]
[[ 9.96194698e-01  5.28626094e-08  8.71557446e-02  0.00000000e+00]
 [-5.31687361e-08  1.00000000e+00  1.19101319e-09  0.00000000e+00]
 [-8.71557446e-02 -5.82044148e-09  9.96194698e-01  0.00000000e+00]
 [ 1.32496333e-06  5.86542946e-07  3.55803530e-06  1.00000000e+00]]

```

rmse: 8.124985898085903e-06

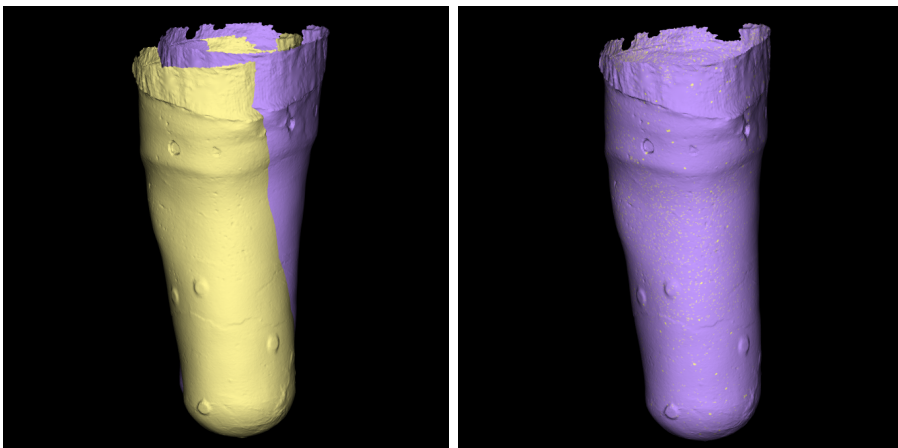


Figura A.16: Risultato grafico dell'allineamento in caso di traslazione e rotazione.

```
[[ 0.9961947  0.          0.08715574  0.          ]
 [ 0.          1.          0.          0.          ]
 [-0.08715574 0.          0.9961947  0.          ]
 [10.          10.         10.          1.          ]]]
[[ 9.96194698e-01  2.52393560e-08  8.71557435e-02  0.00000000e+00]
 [-2.63399267e-08  1.00000000e+00  1.14776061e-08  0.00000000e+00]
 [-8.71557435e-02 -1.37296062e-08  9.96194698e-01  0.00000000e+00]
 [ 1.00000002e+01  9.99999984e+00  9.99999243e+00  1.00000000e+00]]
```

rmse: 9.168459049068966e-06

Bibliografia

- [1] Centro Protesi Vigorso di Budrio. <https://www.inail.it/cs/internet/attivita/prestazioni/centro-protesi-vigorso-di-budrio.html>.
- [2] Ministero della Salute. http://www.salute.gov.it/portale/documentazione/p6_2_8.jsp?lingua=italiano.
- [3] F. Lombardo, M. Maggini, A. De Bellis, G. Seghieri, and R. Anichini. Lower extremity amputations in persons with and without diabetes in Italy: 2001–2010. *Plos One*, 9(1), 2014.
- [4] Ottobock. <https://www.ottobock.it/soluzioni-protesiche/informazioni-per-amputati/termini-tecnici/termini-tecnici-arto-inferiore/>.
- [5] Dentro il Centro INAIL dove nascono le protesi d'oro. <https://www.socialnews.it/blog/2017/01/03/dentro-il-centro-inail-dove-nascono-le-protesi-doro/>.
- [6] Clinical Trials. <https://clinicaltrials.gov/ct2/show/NCT04141748>.
- [7] PhD BPO(Hons) S. Fatone, PhD A. Hansen, and PhD A. G. Cutti. Manual of procedures for comparative effectiveness of socket casting methods: Improving form and fit. *Project Document*, August 2020.
- [8] Symphonie Aqua System VC. <https://www.romedis.com/symphonie-aqua-system-vc.html>.
- [9] Echo Digitizer. <http://rodin4d.com/it/prodotti/acquisizione/echodigitalizzatore>.
- [10] A. Franco. Sistemi di visione. *Slide corso Visione Artificiale e riconoscimento*, 2019-2020.
- [11] M. Szilvási-Nagi and Gy. Máttyási. Analysis of STL files. *Mathematical and Computer Modelling*, 38:945–960, 2003.

- [12] What is stl file. <https://it.3dsystems.com/quickparts/learning-center/what-is-stl-file>.
- [13] Kenton McHenry and Peter Bajcsy. An overview of 3d data content, file formats and viewers. *National Center for Supercomputing Applications*, 1205:22, October 2008.
- [14] J. Shlens. A tutorial on principal component analysis. *Google Research*, April 2014.
- [15] I. Söderkvist. Using SVD for some fitting problems. September 2009.
- [16] Zhen-Huan Zhou and Xiao-Jun Wen. A survey of registration algorithms based on ITK. *Journal of Computers*, 30(2):192–201, 2019.
- [17] D. L G Hill, Philipp G Batchelor, M. Holden, and D. J Hawkes. Medical image registration. *Physics in medicine and biology*, 46, June 2001.
- [18] B. Bellekens, V. Spruyt, R Berkvens, and M Weyn. A survey of rigid 3d point-cloud registration algorithms. *AMBIENT 2014 : The Fourth International Conference on Ambient Computing, Applications, Services and Technologies*, August 2014.
- [19] M. Greenspan and M. Yurick. Approximate k-d tree search for efficient ICP. *Proceedings of the Fourth International Conference on 3-D Digital Imaging and Modeling*, 2003.
- [20] M. E. Biancolini, A. Chiappa, F. Giorgetti, S. Porziani, and M. Rochette. Radial basis functions mesh morphing for the analysis of cracks propagation. *AIAS 2017 International Conference on Stress Analysis*, 8:433–443, September 2017.
- [21] G. Donato and S. Belongie. Approximate thin plate spline mappings. *Computer Vision. ECCV 2002. Lecture Notes in Computer Science*, 2352:21–31, April 2002.
- [22] R. Sprengel, K. Rohr, and H. Siegfried Stiehl. Thin-plate spline approximation for image registration. *18th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Amsterdam*, 3:1190–1191, 1996.
- [23] S. Masecchia. Metodi statistici per l’analisi ed il riconoscimento della forma. *Tesi di Laurea Magistrale in Informatica, Università degli studi di Napoli Federico II*, 2008/2009.

- [24] A. Ross. Procrustes analysis. *Department of Computer Science and Engineering University of South Carolina, SC 29208*.
- [25] PhD Joan E. Sanders, MSE Michael R. Severance, and CPO Kathryn J. Allyn. Computer-socket manufacturing error: How much before it is clinically apparent? *Journal of Rehabilitation Research and Development*, 49(4):567–582, 2012.
- [26] M. E. Biancolini. RBF morph software - How to reshape the CAE workflows by radial basis functions mesh morphing. *TechNet Alliance Spring Meeting*, 2014.
- [27] M. E. Biancolini and P. P. Valentini. Virtual human bone modelling by interactive sculpting, mesh morphing and force-feedback. *International Journal on Interactive Design and Manufacturing*, 12:1223–1234, 2018.
- [28] Geodesia. <https://www.treccani.it/enciclopedia/geodesia/>.
- [29] F. Facchinei. Cammini minimi. *Sapienza Università di Roma*, 2016.
- [30] T. H. Cormen, C. E. Leiserson, Ronald L. Rivest, and C. Stein. *Introduction to Algorithms, Second Edition*. The MIT Press, McGraw-Hill, 2001.
- [31] M. Luthi, C. Jud, T. Gerig, and T. Vetter. Gaussian process morphable models. 2016.
- [32] M.D. Hanwell, K.M. Martin, A. Chaudhary, and L.S. Avila. The visualization toolkit (VTK): Rewriting the rendering code for modern graphics cards. *SoftwareX*, 1-2:9–12, 2015.
- [33] W.J. Schroeder, L.S. Avila, and W. Hoffman. Visualizing with vtk: A tutorial. *IEEE Computer Graphics and Applications*, 20(5):20–27, 2000.
- [34] C. Quammen. Scientific data analysis and visualization with python, vtk, and paraview. *Python in Science Conference*, pages 31–38, 01 2015.
- [35] VTK and VMTK for image processing and visualization. *IUSS Istituto Universitario di studi Superiori di Padova*.
- [36] A. Tamerus. A visualisation pipeline editing interface to vtk using python introspection. *Tesi di Laurea in Informatica, Universiteit van Amsterdam*, 2013.
- [37] C.B. Sullivan and A.A. Kaszynski. Pyvista: 3d plotting and mesh analysis through a streamlined interface for the visualization toolkit (VTK). *JOSS: Journal of Open Source Software*, 37(4):1450, 2019.

- [38] Kok-Lim Low. Linear least-squares optimization for point-to-plane icp surface registration. *Technical Report TR04-004, Department of Computer Science, University of North Carolina*, 2004.
- [39] J.W. Steer, O. Stocks, J. Parsons, P.R. Worsley, and A.S. Dickinson. ampscan: A lightweight python package for shape analysis of prosthetics and orthotics. *JOSS: Journal of Open Source Software*, 5(48):2060, 2019.
- [40] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia. MeshLab: an Open-Source Mesh Processing Tool. *The Eurographics Association*, 2008.
- [41] G. Sacchetti. Studio e implementazione di un algoritmo per lo skinning automatico di mesh poligonali deformabili. *Tesi di Laurea Magistrale in Informatica, Università di Pisa*, 2013/2014.
- [42] S. Karamizadeh, Shahidan M. Abdullah, Azizah A. Manaf, Mazdak Zamani, and Alireza Hooman. An overview of principal component analysis. *Journal of Signal and Information Processing*, 4:173–175, August 2013.