

Università degli Studi di Roma Tor Vergata

MACROAREA DI INGEGNERIA
Corso di Laurea Triennale in Ingegneria Meccanica

TESI DI LAUREA:

**Caratterizzazione delle proprietà meccaniche
del corpo vertebrale di L1 mediante
Statistical Shape and Appearance Modelling**

Candidato:
Francesco Pais

Relatore:
Prof. Ing. M. E. Biancolini

Correlatore:
Dott. Ing. M. Sensale

*A mamma e papà,
per essere sempre stati sinceri con me
e per avermi insegnato a dare il meglio.*

Se qualcosa può andar male, lo farà.

Legge di Murphy

Sommario

Per migliorare la qualità della progettazione di dispositivi *patient-specific* o per personalizzare operazioni chirurgiche è importante ottenere una caratterizzazione il più precisa possibile del paziente e della parte anatomica che bisogna operare. Per evitare di sottoporre i pazienti ad operazioni invasive e costose come DXA, CT Scan o MRI, oggi si fa sempre più uso dei metodi statistici di Statistical Shape and Appearance Models (SSAM). A partire da immagini di *training*, in combinazione con analisi che riducono l'ordine di dimensioni dei dati campionati come la Principal Component Analysis (PCA), si arriva alla generazione di modelli bi o tridimensionali noti come *Digital Twins*: fedeli riproduzioni virtuali caratterizzati dalle stesse proprietà meccaniche dell'oggetto fisico che rappresentano su cui si effettuano gli studi *in silico*. L'oggetto di questa analisi è il corpo vertebrale della prima vertebra lombare L1, di cui, per mezzo di mesh isotopologiche ottenute con l'utilizzo di tecniche di *mesh morphing*, sono state caratterizzate le proprietà meccaniche con la creazione di uno *Statistical Appearance Model*.

Indice

1	Introduzione al problema	1
1.1	Generalità sul moderno sistema sanitario	1
1.2	Problematiche legate alla spina dorsale	4
1.3	Il rachide	7
1.4	Le vertebre	9
1.5	Oggetto dello studio: vertebre lombari	12
2	Teoria e metodologie	15
2.1	Generalità dell'analisi agli elementi finiti	15
2.2	Concetto di <i>mesh</i>	16
2.3	Mesh Morphing	20
2.3.1	Radial Basis Functions	23
2.4	Standard Triangulation Language: STL	25
2.5	Approccio Statistico	26
2.5.1	Distribuzioni univariate	27
2.5.2	Distribuzioni multivariate	29

2.5.3	Singular Value Decomposition: SVD	31
2.5.4	Principal Component Analysis: PCA	33
2.5.5	Validazione del modello: LOOCV	37
2.6	Statistical Shape and Appearance Modelling	38
2.6.1	Statistical shape models	38
2.6.2	Statistical appearance models	45
2.6.3	Applicazioni mediche	46
3	Analisi	49
3.1	Raccolta dati	49
3.2	Meshing	51
3.3	Morphing per l'isotopologia	52
3.4	Mappatura delle proprietà meccaniche	54
3.5	PCA e generazione del modello SAM	56
4	Conclusione	63
4.1	Limiti dell'approccio ed analisi dei risultati	63
4.2	Conclusione e commenti finali	68
	Script Matlab	71
	Bibliografia	102

Elenco delle figure

1.1	Pedicle screws	5
1.2	Riferimento dei piani del corpo umano	8
1.3	Colonna vertebrale	10
1.4	Schema generale di una vertebra	11
1.5	Rappresentazione di una vertebra lombare	12
2.1	Esempio di errore di discretizzazione del <i>meshing</i>	17
2.2	Elementi più utilizzati nel metodo agli elementi finiti	19
2.3	Dati sperimentali e modello lineare	26
2.4	La varianza come indicazione della "larghezza" di una distribuzione di dati	29
2.5	Matrici di covarianza e punti di isoprobabilità in 2D	31
2.6	Full SVD	33
2.7	Economy SVD	33
2.8	Esempio di Screen Plot	36
2.9	Landmarks posizionati manualmente	40

3.1	Immagine tomografica del <i>paziente 1</i> in 3D-Slicer	50
3.2	File .stl della mesh di reference	51
3.3	STL dei corpi vertebrali dei pazienti	51
3.4	Reference dopo il meshing in Ansys: vista isometrica e in sezione	52
3.5	Preview del mesh morphing del paziente 13	54
3.6	Corpo vertebrale del paziente 1 allineato all'immagine CT . .	55
3.7	Componenti principali e varianza	57
3.8	Errore medio ottenuto dalla combinazione lineare di tutte le componenti principali	59
3.9	Errore massimo ottenuto dalla combinazione lineare di tutte le componenti principali	60
3.10	Errore medio ottenuto dalla combinazione lineare dei primi 9 modi	61
3.11	Errore massimo ottenuto dalla combinazione lineare dei primi 9 modi	61
4.1	CT Scan e posizionamento virtuale dei cilindri per misurare l'intensità dei voxel	65
4.2	Errore di interpolazione tra gli algoritmi RBF e FFDM	67
4.3	Errore di interpolazione tra gli algoritmi RBF e FFDM rispet- to alla mesh ottimizzata	68

Capitolo 1

Introduzione al problema

1.1 Generalità sul moderno sistema sanitario

Il progresso tecnologico ha raggiunto un passo così veloce che oggi ci è permesso quasi tutto nella vita, e ciò è particolarmente vero nel campo della medicina. Essa ha una storia molto lunga: gli uomini primitivi consideravano le malattie come le rappresentazioni dei capricci degli dèi o degli spiriti e impararono con l'esperienza che certe piante avevano delle proprietà guaritrici, cominciando così a coltivarle. I ritrovamenti in quasi ogni parte del mondo di scheletri di cranio bucati, inoltre, indica il fatto che la pratica della chirurgia era già qualcosa di noto: i fori servivano per accedere al cervello e liberare il paziente che soffriva di mal di testa estremamente dolorosi o di malattie ben più gravi, come l'epilessia, dai demoni maligni che causavano tali patologie. Tra gli antichi Greci, *Ippocrate* insegnava ai suoi allievi che le malattie erano processi naturali che si sviluppavano secondo passaggi logici e che i sintomi non erano altro che le reazioni del corpo ad esse. A cavallo tra

il XV e XVI secolo, si accese il desiderio di conoscere la vera natura di tutti i fenomeni naturali, anche in campo medico. Molti studiosi, come Leonardo da Vinci, Michelangelo o Raphael Durer, approfondirono l'anatomia umana e cominciarono a rappresentarla nella sua vera essenza. Nacquero quindi le prime scuole di medicina a Bologna, Padova, Oxford e Montpellier. Tutte abbracciavano la dottrina di Ippocrate, secondo cui il terapeuta doveva accompagnare il corpo del paziente nella conquista della cura della malattia. Con la nascita del metodo sperimentale introdotto da Galileo Galilei, iniziò uno studio più sistematico della natura umana. La collaborazione tra le leggi della fisica e lo studio dell'anatomia portò alla scoperta del meccanismo dello scorrimento del sangue all'interno del sistema venoso e quindi alla conoscenza del funzionamento del cuore. In Inghilterra, sotto il regime di Enrico VIII, nacque ufficialmente il mestiere del medico: dottori e studenti di medicina sostituirono i ruoli delle suore e dei monaci.

La moderna medicina cominciò a svilupparsi solo nel XX secolo e la tecnologia ha giocato un ruolo fondamentale. Fino al 1900, infatti, il campo della medicina era riservato esclusivamente ai fisici, ma i loro servizi potevano essere ancora praticati da chiunque riusciva a curare le patologie più comuni imparando con l'esperienza. Intorno al 1930, grazie all'introduzione dei Raggi-X, si poté osservare praticamente ogni organo del corpo umano. Inoltre, l'introduzione della penicillina all'inizio degli anni '40 ridusse notevolmente uno dei pericoli legati all'ospedalizzazione: quello dell'infezione tra pazienti.

Grazie alla rivoluzione tecnologica degli anni '60 che portò alla nascita dei primi componenti elettronici, si arrivò alla costruzione dei calcolatori

elettronici che potevano eseguire finalmente tutti i calcoli troppo complessi e, circa dieci anni dopo, comparve la tecnica di diagnostica per immagini della *tomografia computerizzata (CT)* e della *risonanza magnetica (MRI)*.

Nel 1954 ci fu il primo trapianto di reni: il concetto di *organo artificiale* acquistò finalmente ufficialità ed entrò in voga in ambito medico portando allo sviluppo della tecnologia che permise la progettazione di protesi come le valvole cardiache e i vasi sanguigni artificiali.

Grazie al continuo miglioramento e all'avanzamento nel campo delle nanotecnologie e dell'ingegneria dei tessuti e degli organi artificiali, quello che un tempo credevamo fosse fantascienza sta oggi diventando realtà. Ciò è testimoniato dalla nascita di progetti grandiosi come quello di *MeDiTATe* (Medical Digital Twin for Aneurysm prevention and Treatment), che si occupa dello sviluppo di *digital twins* per la progettazione di protesi mediante tecniche di *additive manufacturing* per ridurre l'incidenza di complicazioni tromboemboliche; o di progetti d'eccellenza MSCA come *Spinner-EID* (con la cui collaborazione nasce questa tesi), il cui obiettivo è quello di sviluppare nuovi metodi e dispositivi per aiutare i chirurghi ad effettuare le scelte più ottimali prima e durante un'operazione alla spina dorsale, fornendo un piano di cura personalizzato per il paziente per ottenere il miglior risultato possibile¹.

¹Per maggiori informazioni sul progetto Spinner si rimanda al sito internet ufficiale: <https://www.sheffield.ac.uk/spinner-eid>

1.2 Problematiche legate alla spina dorsale

Le malattie legate alla colonna vertebrale sono un problema che riguarda molte persone in tutto il mondo. Ci sono numerose patologie diverse che colpiscono differenti parti della spina e le più comuni sono:

- Dolore alla schiena
- Degenerazione del disco spinale
- Scogliosi
- Fratture
- Tumori
- Osteoporosi

Una spina dorsale danneggiata è accompagnata da perdita di mobilità e da una diminuzione generale della qualità di vita. Nel 2008, circa 415000 interventi di fusione spinale sono stati eseguiti solo negli Stati Uniti e ogni anno tendono ad aumentare sempre di più. I costi medici legati alle malattie della colonna vertebrale sono molto elevati e le operazioni chirurgiche necessitano di grandi sforzi e tecniche precise per prevenire complicazioni post-operatorie, garantendo al paziente un risultato di successo e un recupero rapido.

Per trattare gli infortuni alla spina dorsale e, più in generale, le fratture toraco-lombari, è molto comune l'utilizzo di *pedicle screws* (viti peduncolari, ovvero applicate nei *peduncoli* della vertebra²).

²Il peduncolo è una parte dell'osso vertebrale. Per un maggiore approfondimento si veda § 1.4.

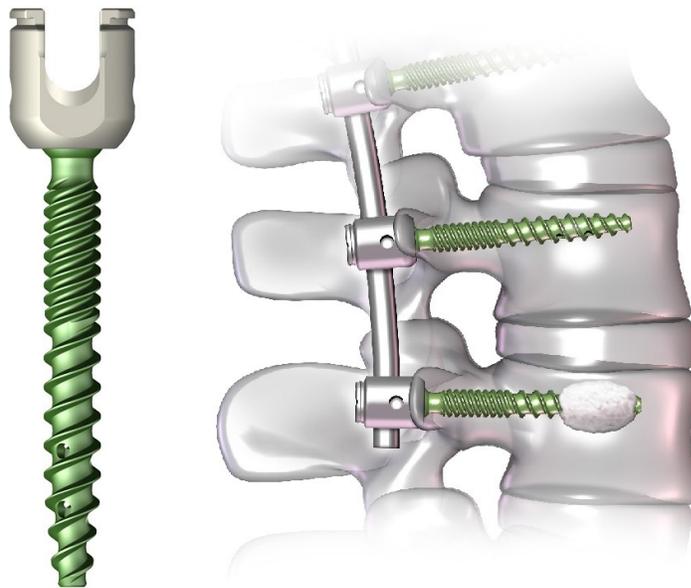


Figura 1.1: Pedicle screws

Esse sono delle viti che si mettono sopra e sotto le vertebre fuse per fornire supporto e forza extra. Una piccola barra, chiamata *rod*, viene utilizzata per connettere le viti e impedire i movimenti in modo da permettere all'innesto osseo di guarire. La fusione spinale è generalmente indicata per stabilizzare la colonna vertebrale:

- dopo incidenti o infortuni
- quando si rimuovono delle vertebre per liberare i nervi compressi
- in casi di scogliosi o cifosi
- quando si soffre di tumori che causano indebolimento delle ossa e quindi instabilità della spina dorsale.

L'impiego di *pedicle screws* nella spina dorsale lombare è la tecnica più diffusa per ottenere fusione spinale e stabilizzazione. Lo svantaggio del loro

utilizzo è l'allentamento nel corso del tempo e in alcuni casi la rottura, fatti che portano a complicazioni meccaniche che implicano nel 6% dei casi a un intervento di revisione. Mentre i chirurghi decidono la grandezza ottimale della vite, il punto di inserimento e l'orientazione della vite basandosi su misure anatomiche che derivano dalle immagini CT, i modelli agli elementi finiti sono efficienti strumenti matematici per valutare la stabilità di differenti configurazioni della spina dorsale operata sotto diverse condizioni di carico.

Secondo lo studio condotto da *Sensale et al.* [23], è emerso che il diametro della vite ha un'influenza significativa sulla deflessione massima totale della testa della stessa. Fissando la lunghezza, la deflessione aumenta al diminuire del diametro, mentre, fissando il diametro, diminuisce all'aumentare della lunghezza. Inoltre è risultato che il diametro della vite ha anche un'enorme impatto sugli stress equivalenti di Von Mises: al diminuire del diametro, fissata la lunghezza, gli stress aumentano; al contrario, fissando il diametro e aumentando la lunghezza della vite, gli stress sono minori. Secondo questo studio, in conclusione, manca un approccio più generale (statistico) per trovare la combinazione ottimale di diametro e lunghezza della vite per ciascun paziente ed è fondamentale riuscire a ottimizzare le dimensioni del diametro delle viti in quanto hanno una grande influenza sugli stress a cui sono soggette.

Un altro tipo di patologie legate alla spina dorsale riguardano le deformità iatrogene³. Il numero di pazienti che soffrono di queste complicazioni sta aumentando sempre di più in quanto aumenta anche il numero di interventi

³Le deformità iatrogene sono deformità che derivano o sono conseguenza di una particolare terapia o intervento chirurgico.

spinali eseguiti ogni anno in tutto il mondo. Correggere questa deformazione è una grande sfida a causa delle complesse irregolarità geometriche e anatomiche causate dagli interventi precedenti. Le forme più comuni di deformità iatrogene sono la perdita della lordosi e lo sviluppo di cifosi segmentaria o globale. Nello studio condotto da *Fayad et al.* [9], una paziente di 71 anni che soffriva di un grave dolore alla schiena non riusciva a camminare per più di 50 metri. Questi dolori derivavano da numerosi interventi a cui ha dovuto sottoporsi nel corso di 39 anni e che hanno portato a un severo disallineamento sagittale e coronale. Grazie a modelli virtuali e tridimensionali, i ricercatori sono riusciti a comprendere l'unico e complesso problema anatomico-geometrico che la caratterizzava e quindi a pianificare correttamente l'operazione chirurgica correttiva, che è andata a buon fine ed ha permesso un riallineamento di circa 30° della lordosi lombare, riducendo l'indice ODI da 80 a 20 e il VAS da 9 a 3⁴.

1.3 Il rachide

Il *rachide* è un complesso formato da uno scheletro - la colonna vertebrale - unito da varie articolazioni e rivestito da muscoli che gli conferiscono stabilità e mobilità. Ha la funzione di sostegno della testa e del tronco e di protezione del midollo spinale contenuto al suo interno.

La *colonna vertebrale* è un complesso osseo costituito da 33 o 34 ossa articolate fra loro: le *vertebre*. Fornisce un sostegno al corpo nella statica

⁴Il VAS (Visual Analog Scale) e l'ODI (Oswestry Disability Index) sono due indici che misurano rispettivamente l'intensità del dolore e l'influenza di esso sulla vita di tutti i giorni del paziente.

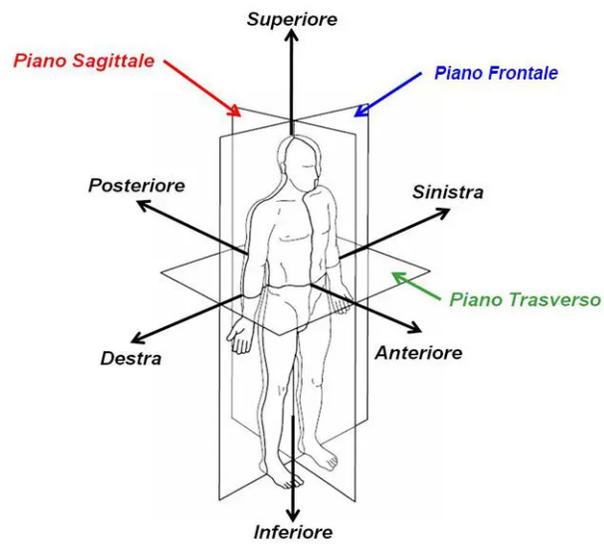


Figura 1.2: Riferimento dei piani del corpo umano

e nella dinamica. Può essere suddivisa in cinque segmenti che formano lo scheletro delle varie parti del corpo:

- Il segmento *cervicale*, che dà sostegno al collo ed è formato da 7 vertebre cervicali;
- Il segmento *toracico*, costituito da 12 vertebre toraciche che si articolano anche con le coste;
- Il segmento *lombare*, composto da 5 vertebre lombari, l'ultima delle quali è articolata con l'osso sacro;
- Il segmento *sacrale*, costituito da 5 vertebre sacrali fuse insieme a formare l'osso sacro;
- Il segmento *coccigeo*, formato da 4 o 5 vertebre coccigee unite a formare il coccige.

La colonna vertebrale non è rettilinea, ma presenta alcune curvature sia sul piano sagittale che su quello frontale. Lateralmente infatti si può osservare una convessità al livello del segmento cervicale chiamata *lordosi cervicale*; una convessità posteriore nel segmento toracico detta *cifosi dorsale*; una convessità anteriore nel segmento lombare (*lordosi lombare*) e una convessità nuovamente posteriore nel segmento sacro-coccigeo nota come *cifosi sacrococcigea*.

In vista anteriore, è visibile una leggera curva concava a sinistra nel tratto toracico dovuta probabilmente alla presenza del cuore e dell'arco aortico, insieme a due curve minori. Queste curvature vengono definite *scoliosi fisiologica*. All'interno della colonna vertebrale è presente il *canale vertebrale*, che si forma per sovrapposizione dei fori vertebrali delle singole vertebre e nel quale è presente il midollo spinale rivestito dalle meningi.

1.4 Le vertebre

Le vertebre hanno caratteristiche morfologiche comuni indipendentemente dal segmento cui appartengono e caratteristiche specifiche che permettono di riconoscere il segmento di appartenenza.

Sono ossa brevi formate prevalentemente da tessuto osseo spugnoso rivestito da un sottile strato di tessuto osseo compatto. Sono costituite da un *corpo* e un *arco*, che insieme delimitano il *foro vertebrale*. Dall'arco si originano delle sporgenze chiamate *processi*, che si dividono a loro volta in *trasversali*, *articolari* e *spinosi*.

Il corpo è la parte più voluminosa della vertebra. Ha una forma cilindrica

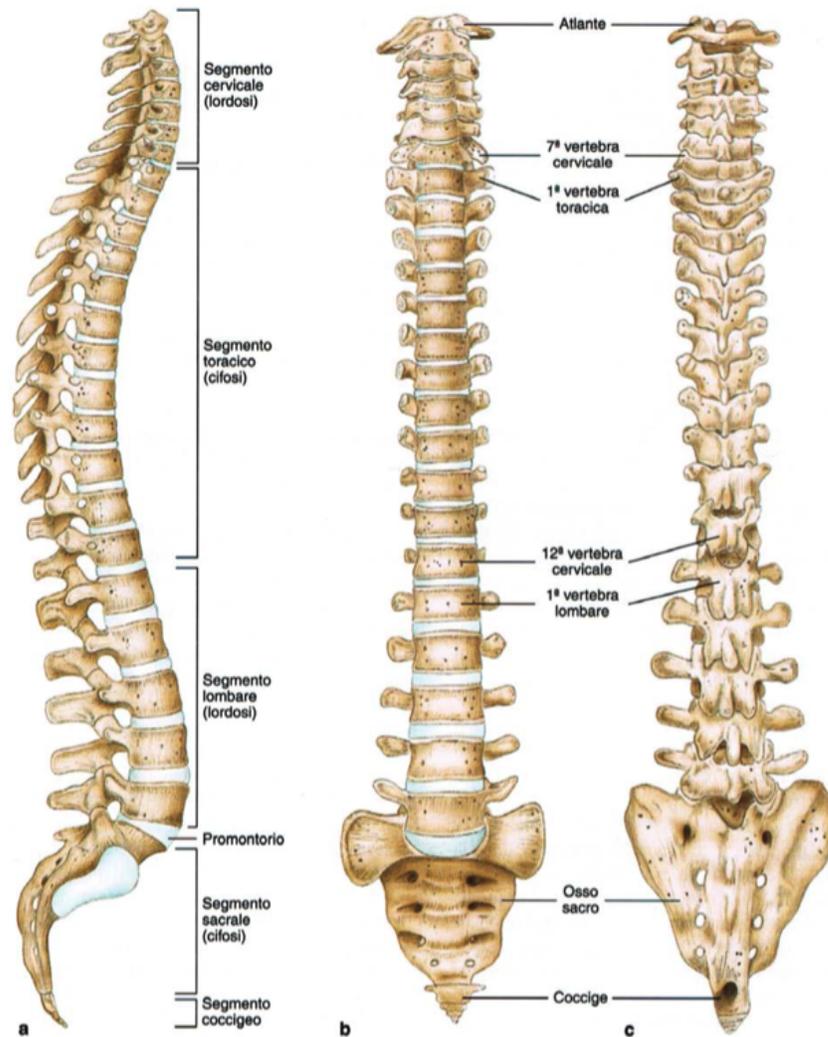


Figura 1.3: Colonna vertebrale

irregolare nella quale si possono descrivere una porzione centrale e un anello periferico. All'interno del corpo sono presenti due facce *invertebrali superiore* e *inferiore* leggermente concave al centro, e un contorno detto *circonferenza*.

L'arco è la parte posterolaterale della vertebra ed è formato dai *peduncoli*, dalle *lamine* e dai processi che originano da esso.

I peduncoli dell'arco vertebrale uniscono il corpo alle *lamine* che hanno a

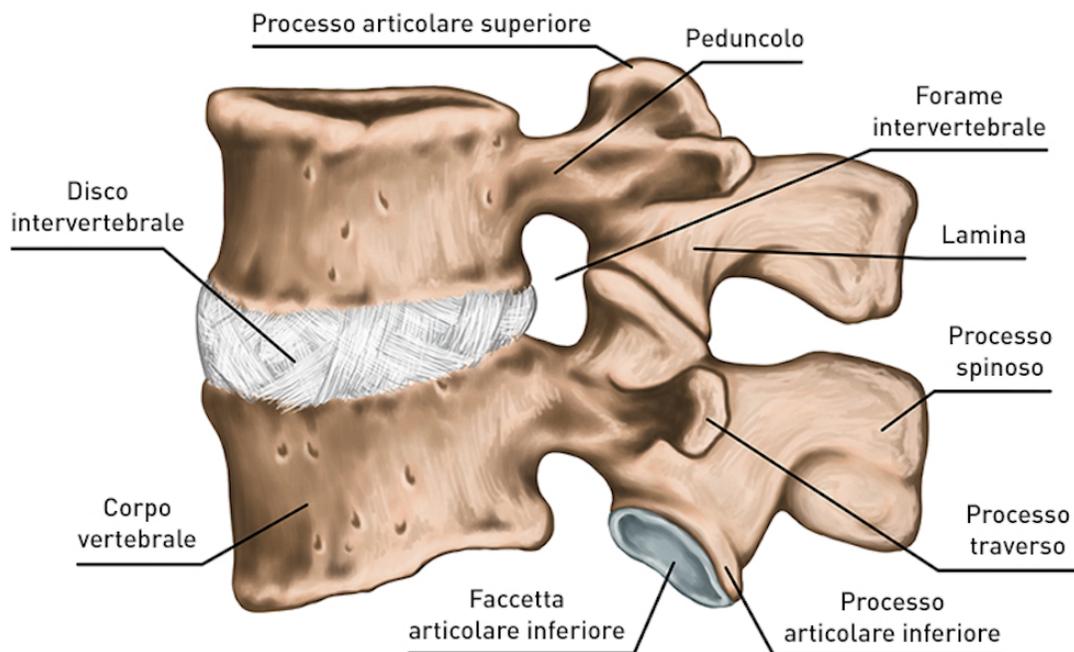


Figura 1.4: Schema generale di una vertebra

loro volta una forma appiattita e sono inclinate verso l'alto e dorsalmente. Le lamine di una vertebra si sovrappongono a quelle della vertebra sottostante come le tegole di un tetto. I margini superiori e inferiori dei peduncoli sono incavati e formano le *incisure vertebrali superiore e inferiore*. Dall'accostamento delle incisure di due vertebre contigue si forma il *foro intervertebrale*, attraverso il quale passa il nervo spinale.

I processi trasversi si originano all'unione dei peduncoli con le lamine e si dirigono in fuori e posteriormente.

I processi articolari sono quattro (due superiori e due inferiori). I processi superiori si dirigono cranialmente e presentano alle loro estremità le faccette articolari superiori per l'articolazione con i processi articolari inferiori della

vertebra soprastante. Il contrario avviene per i processi inferiori.

Fra ogni corpo vertebrale si inserisce il *disco intervertebrale* che, oltre a separare i corpi vertebrali, ha un'importante funzione di ammortizzatore.

1.5 Oggetto dello studio: vertebre lombari

Le vertebre lombari presentano un grosso corpo che va aumentando di volume in senso craniocaudale⁵. Hanno un processo spinoso appiattito e tre processi al posto di quello trasverso. Il corpo delle vertebre lombari è voluminoso e ha la forma di un cuneo, essendo più spesso ventralmente che dorsalmente. I peduncoli sono tozzi e presentano un'incisura vertebrale inferiore più profonda rispetto a quella superiore.

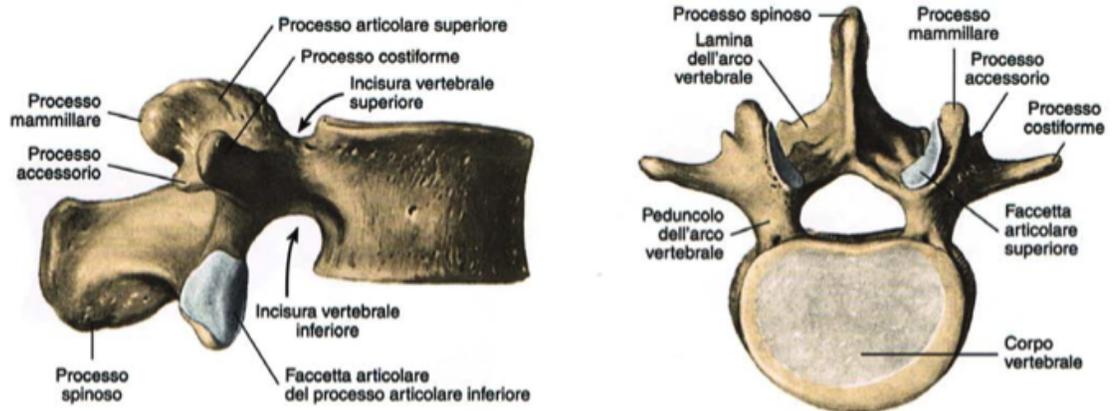


Figura 1.5: Rappresentazione di una vertebra lombare

Le lamine sono spesse e più sviluppate in altezza che in larghezza. I processi trasversi sono sostituiti da tre processi: uno è quello *costiforme*; gli

⁵Termine antropometrico nato dalla crasi di craniale e caudale. Descrive il movimento o l'estensione di una struttura dalla posizione superiore a quella inferiore.

altri due (*mammillare* e *accessorio*) sono dei piccoli rilievi l'uno sopra l'altro. I processi articolari superiori originano subito dietro al processo costiforme e si dirigono in alto, mentre i processi articolari inferiori si portano verso il basso e hanno faccette articolari inferiori a forma di cilindro solido. Il processo spinoso è robusto e ha la forma di una lamina rettangolare. Il foro vertebrale, infine, è triangolare a base anteriore.

L'oggetto dello studio di questa tesi è il corpo vertebrale della prima vertebra lombare L1.

Capitolo 2

Teoria e metodologie

2.1 Generalità dell'analisi agli elementi finiti

Il metodo agli elementi finiti è un metodo numerico per risolvere problemi di fisica e ingegneria. Le equazioni che governano il comportamento meccanico dei corpi reali sono generalmente equazioni differenziali alle derivate parziali spesso non lineari, molto difficili da risolvere in termini matematici. Di fatto, solo per certe situazioni semplificate si possono ottenere soluzioni analitiche. Esistono numerosi metodi numerici utilizzati nelle applicazioni ingegneristiche per cui l'utilizzo di calcolatori elettronici è fondamentale. Nel mondo della progettazione meccanica dove si utilizzano software CAD (Computer-Aided Design), il metodo d'analisi che meglio si integra con essi è il *Metodo agli Elementi Finiti (Finite Element Method, FEM)*.

Il metodo agli elementi finiti è impiegato per diversi scopi: statica, dinamica, problemi lineari e non, analisi delle tensioni e delle deformazioni, vibrazioni libere e forzate, trasferimento di calore, instabilità elastica (buckling),

acustica, elettrostatica, magnetismo, fluidodinamica e molto altro.

2.2 Concetto di *mesh*

Definiamo un generico componente meccanico come un *corpo elastico continuo*. Il metodo agli elementi finiti divide il corpo in piccole e ben definite strutture elastiche chiamate *elementi* creando in questo modo una griglia, chiamata *mesh*. Dunque: le equazioni differenziali vengono lasciate inalterate (relativamente a ciascun elemento finito) mentre il dominio viene discretizzato. In un problema al continuo di qualsivoglia dimensione, cioè in un corpo o in una regione dello spazio in cui abbia luogo un particolare fenomeno, la variabile di campo, come la pressione, lo spostamento o la densità, è funzione di ciascun generico punto del dominio di definizione. Di conseguenza il problema presenta un numero *infinito* di incognite. Discretizzare agli elementi finiti riduce il problema ad un numero *finito* di incognite e, tramite l'impiego di funzioni polinomiali e di operazioni matriciali, il comportamento dell'intero continuo viene sviluppato in funzione delle proprietà geometriche e materiali di ciascun elemento finito. I vari carichi possono essere applicati all'*interno* (come la gravità o la temperatura), sulla *superficie* o ai *nodi* dell'elemento. I nodi sono i componenti fondamentali di ogni elemento, in quanto è tramite di essi che i vari elementi si collegano tra loro e sono quindi il luogo in cui si applicano le condizioni al contorno. Ciascun nodo possiede dei gradi di libertà che consistono nelle traslazioni e rotazioni indipendenti che si possono avere: al più queste consistono in tre traslazioni e tre rotazioni.

Dato che il metodo agli elementi finiti è una tecnica numerica che discre-

tizza il dominio di una struttura continua, gli errori sono inevitabili. Essi possono essere:

- *Computazionali*, dovuti all'arrotondamento del computer a causa dei calcoli a virgola mobile
- Di *discretizzazione*, in quanto la geometria e la distribuzione delle deformazioni di un corpo reale varia continuamente e utilizzare un numero finito di elementi per modellare la struttura introduce errori nel far combaciare le geometrie dovuti a una limitazione matematica degli elementi.

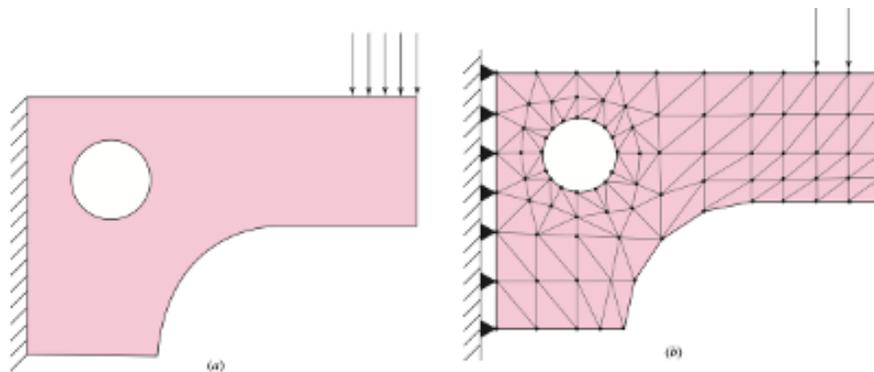


Figura 2.1: Esempio di errore di discretizzazione del *meshing*

Si prenda come esempio la lastra piana sottile a spessore costante della figura 2.1, in cui si può osservare un modello della lastra agli elementi finiti dove sono stati utilizzati elementi a 3 nodi. Questo tipo di elemento, tuttavia, comporta due problemi: il primo riguarda il fatto che ha dei lati rettilinei che rimangono tali anche quando devono approssimare la curvatura del foro o del raccordo; il secondo problema – più importante – è che le sollecitazioni

nei vari punti del modello reale cambiano rapidamente (e.g. in prossimità del foro), mentre il fatto che gli elementi siano costanti induce solo a un'approssimazione della sollecitazione media al centro di ciascun elemento. Pertanto i risultati che si otterranno non saranno del tutto affidabili. Si può migliorare la precisione e l'accuratezza cambiando la grandezza della mesh (riducendo l'element size), oppure usando un elemento migliore (come ad esempio un quadrilatero ad otto nodi).

La scelta della tipologia di elementi da utilizzare dipende dalla struttura fisica del corpo sottoposto alle condizioni di carico e dal livello di accuratezza dei risultati richiesta dalla soluzione. È necessario saper scegliere se utilizzare elementi mono, bi, o tridimensionali. La scelta del tipo di elemento più appropriato è uno dei compiti più importanti che il progettista/modellatore deve effettuare. Ci sono differenti geometrie disponibili per gli elementi che possono essere suddivisi principalmente in lineari, superficiali e solidi. La seguente tabella ne elenca alcuni tra i più utilizzati.

Gli elementi monodimensionali consistono fondamentalmente in un elemento trave. Sono caratterizzati da una sezione retta, ma vengono rappresentati da dei segmenti e sono utilizzati particolarmente per modellare strutture reticolari e travi. L'elemento lineare più semplice ha due nodi posizionati all'estremità dello stesso. Gli elementi bidimensionali (superficiali) sono caricati da forze agenti sullo stesso piano e possono essere elementi triangolari o quadrangolari. Gli elementi più semplici di questo tipo hanno lati dritti con i nodi agli angoli, ma ci sono anche elementi di ordine superiore con i nodi posizionati anche nei punti medi dei vari lati (vengono chiamati elementi *quadratici*) o con lati curvi. Possono avere spessore variabile e sono spes-

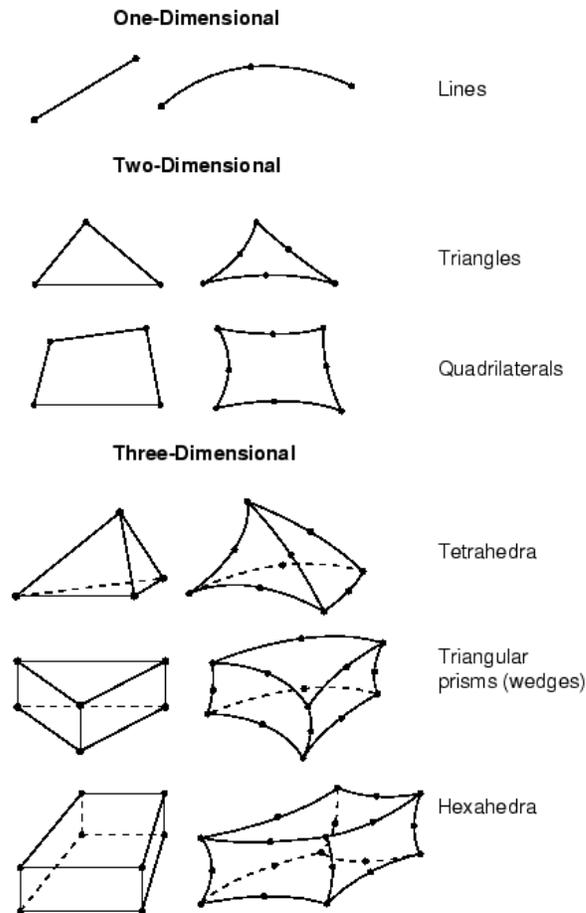


Figura 2.2: Elementi più utilizzati nel metodo agli elementi finiti

so utilizzati per modellare un vasto insieme di problemi ingegneristici. Gli elementi tridimensionali più comuni sono quelli tetraedrici ed esaedrici (chiamati anche *brick*) e vengono particolarmente utilizzati quando è necessario effettuare un'analisi tridimensionale dello stress agente sul corpo.

Si parla di affinamento di mesh (*mesh refinement*) quando la mesh viene modificata in seguito a un'analisi per migliorare i risultati. Essi sono più accurati quando la densità di mesh è più elevata nelle zone di concentrazione delle tensioni e/o quando, nelle zone di variazioni geometriche, le mesh sono

più gradualmente. Ci sono tre modi per generare delle mesh:

- Generazione *manuale*: è il metodo impiegato agli inizi dello sviluppo degli elementi finiti. È un compito laborioso ed è raramente effettuato;
- Generazione *semiautomatica*: il modellatore deve dividere il corpo in regioni e successivamente il calcolatore, grazie a sofisticati algoritmi, riesce a creare la mesh di ciascuna regione;
- Generazione *automatica*: il modellatore non deve fare nulla in quanto il calcolatore riconosce automaticamente i limiti del corpo e crea le mesh (in alcuni casi riesce anche ad effettuare un affinamento della mesh autoadattivo).

2.3 Mesh Morphing

Al giorno d'oggi, in cui i computer sono applicati quasi ovunque nella scienza e nell'ingegneria, diventa sempre più importante implementare funzioni matematiche per una valutazione efficiente nei programmi informatici. A questo scopo è solitamente necessario utilizzare tutti i tipi di *approssimazioni* delle funzioni invece che la loro forma matematica esatta e le ragioni sono varie. Una semplice è che in molti casi non è possibile implementare le funzioni in modo esatto. Inoltre, la funzione che vogliamo utilizzare potrebbe non essere completamente nota, o potrebbe essere troppo costosa e impegnativa in termini di tempo e memoria del computer per poter essere calcolata in anticipo, il che è un'altra tipica e importante ragione per cui sono necessarie le approssimazioni. Questo è vero anche a fronte di una velocità e di una

disponibilità di memoria del computer che sono sempre maggiori, dato che memoria e velocità aggiuntive aumenteranno sempre la richiesta degli utenti e la dimensione dei problemi da risolvere. Infine, i dati che definiscono la funzione possono dover essere calcolati interattivamente o con un approccio graduale, il che rende ancora una volta adatto il calcolo delle approssimazioni.

I prototipi virtuali stanno diventando uno standard in tutti i processi di progettazione ingegneristica. Di solito si basano su una geometria tridimensionale definita combinando caratteristiche e operazioni per produrre la forma finale. Questo approccio presenta alcune limitazioni per le parti estetiche di forma complessa o per le geometrie organiche, per cui si può ricorrere alla scultura virtuale. Il mesh morphing è un'opzione che consente di superare queste complicazioni e di ottenere applicazioni rilevanti per diversi settori industriali. Il *mesh morphing* consiste nell'adattamento di una griglia computazionale adottata dai software CAE (*Computer-Aided Engineering*, ovvero software per *l'ingegneria aiutata dai computer*) come ad esempio una mesh solida di un componente strutturale che deve essere processata da un'analisi agli elementi finiti, o una mesh volumetrica completa di condizioni al contorno per un'analisi di fluidodinamica computazionale (CFD). Entrambe possono essere trasformate in una nuova forma semplicemente modificando e aggiornando la posizione dei nodi. Questo significa che la topologia della mesh (i.e. il numero di nodi, di elementi e la connettività) rimane la stessa. Le uniche a cambiare sono le coordinate X, Y e Z dei nodi di quella sottoparte del modello che si va a modificare.

Il mesh morphing è spesso utilizzato per diversi scopi, come la creazione di parametri di forma (i.e. lunghezza, angolo o spessore), o per spostarsi

su una nuova forma nota (ad esempio come progettato dal CAD), o su una forma prevista dalla soluzione CAE.

La capacità del mesh morphing di introdurre parametri di forma in grado di aggiornare sia le mesh di superficie che quelle di volume, tuttavia, può essere sfruttata anche quando tale rappresentazione CAD non è affatto accessibile, come nel caso delle applicazioni biomeccaniche dove i modelli di base delle morfologie studiate consistono in superfici risultanti dal processo di acquisizione di immagini cliniche.

Di solito il mesh morphing è più veloce del *remeshing* (creazione di una mesh dall'inizio), in quanto richiede un minore impegno computazionale; per di più il modello CAE può essere aggiornato in background mantenendo tutte le impostazioni originali delle condizioni al contorno.

Per le applicazioni CAE la tendenza è quella di utilizzare elementi sempre più piccoli perché oggi abbiamo a disposizione una potenza di calcolo molto elevata. Nel 2009 una mesh di grandi dimensioni per l'aerodinamica esterna nell'industria automobilistica era dell'ordine di 50 milioni di elementi, mentre una da 250 milioni era comune per applicazioni di punta come la *Formula 1*. Oggi è ormai comune superare i 100 milioni per l'uso standard e raggiungere il livello del miliardo per le applicazioni di punta. Il morphing delle mesh è utile per qualsiasi dimensione del modello e la complessità dell'azione di morphing cresce con la dimensione della mesh da aggiornare.

Nel campo delle simulazioni CAE biomeccaniche, il mesh morphing si presta molto bene a lavori di medicina predittiva in cui un modello numerico *patient-specific* viene preso come riferimento per comprendere la fisica di interesse mediante tecniche di simulazione. Una volta creato un modello di

riferimento affidabile, è possibile esplorare l'effetto prodotto dalle modifiche della geometria per prevedere l'evoluzione di una patologia o valutare l'efficacia di una correzione chirurgica. È possibile affrontare anche l'interazione delle parti del corpo umano con la geometria dei componenti meccanici, come nel caso del dimensionamento e del posizionamento di parti protesiche nel sistema o nel caso della simulazione di un'operazione chirurgica.

2.3.1 Radial Basis Functions

Le *Radial Basis Functions (RBF)* sono uno strumento matematico in grado di interpolare campi noti su una nuvola di punti. Il mesh morphing definisce un campo di spostamento su una nuvola di punti sorgente (*source points*) e poi lo propaga su una nuvola di punti target (*target points*). Il costo computazionale di RBF può essere molto elevato, per cui sono necessari algoritmi specifici (Fast Radial Basis Functions) per godere di tali vantaggi nelle applicazioni industriali.

Il problema dell'interpolazione delle RBF nello spazio viene posto assegnando N distinti *source points* ($x_{s1}, x_{s2}, \dots, x_{sN}$) in \mathbb{R}^3 e i loro corrispondenti valori reali ($g_{s1}, g_{s2}, \dots, g_{sN}$) e consiste nel determinare una funzione regolare $s(\mathbf{r})$ tale che:

$$s(\mathbf{x}) = \sum_{i=1}^N \gamma_i \phi(\|\mathbf{x} - \mathbf{x}_{si}\|) - h(\mathbf{x}) \quad (2.1)$$

Dove

- ϕ : funzione radiale: $\mathbb{R} \rightarrow \mathbb{R}$

- $h(\mathbf{x}) = \beta_0 + \beta_1 \mathbf{x}^X + \beta_2 \mathbf{x}^Y + \beta_3 \mathbf{x}^Z$, con \mathbf{x}^i : componente i -esima del vettore \mathbf{x}
- γ_i : pesi a valori reali

La funzione di 2.1 è dunque un polinomio di primo grado. I coefficienti γ_i e β_i possono essere determinati dalle condizioni di interpolazione:

$$s(\mathbf{x}_{si}) = \sum_{j=1}^N \gamma_j (\|\mathbf{x}_{si} - \mathbf{x}_{sj}\|) - h(\mathbf{x}_{si}) = g_{si} \quad (2.2)$$

e dalle condizioni di ortogonalità:

$$\sum_{j=1}^N \gamma_j = \sum_{j=1}^N \gamma_j x_{sj}^X = \sum_{j=1}^N \gamma_j x_{sj}^Y = \sum_{j=1}^N \gamma_j x_{sj}^Z = 0 \quad (2.3)$$

Definendo $\phi_{ij} = \phi(\|\mathbf{x}_{si} - \mathbf{x}_{sj}\|)$ e considerando le equazioni (2.2) e (2.3), si ottiene il seguente sistema lineare:

$$\begin{bmatrix} \phi_{11} & \phi_{12} & \dots & \phi_{1N} & 1 & \mathbf{x}_{s1}^X & \mathbf{x}_{s1}^Y & \mathbf{x}_{s1}^Z \\ \phi_{21} & \phi_{21} & \dots & \phi_{2N} & 1 & \mathbf{x}_{s2}^X & \mathbf{x}_{s2}^Y & \mathbf{x}_{s2}^Z \\ \vdots & \vdots \\ \phi_{N1} & \phi_{N1} & \dots & \phi_{NN} & 1 & \mathbf{x}_{sN}^X & \mathbf{x}_{sN}^Y & \mathbf{x}_{sN}^Z \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ \mathbf{x}_{s1}^X & \mathbf{x}_{s2}^X & \dots & \mathbf{x}_{sN}^X & 0 & 0 & 0 & 0 \\ \mathbf{x}_{s1}^Y & \mathbf{x}_{s2}^Y & \dots & \mathbf{x}_{sN}^Y & 0 & 0 & 0 & 0 \\ \mathbf{x}_{s1}^Z & \mathbf{x}_{s2}^Z & \dots & \mathbf{x}_{sN}^Z & 0 & 0 & 0 & 0 \end{bmatrix} \begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \dots \\ \gamma_N \\ \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix} = \begin{pmatrix} g_{s1} \\ g_{s1} \\ \dots \\ g_{sN} \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (2.4)$$

Risolvendo tale sistema si trovano i coefficienti γ_i e β_i .

Quando le RBF vengono utilizzate per il mesh morphing, le tre componenti del campo di spostamenti assegnato ad una certa nuvola di *control points* vengono interpolate nello spazio secondo (2.1) e utilizzate per modificare la posizione nodale della mesh che deve essere *morfata*. Si ottiene quindi il campo di spostamenti:

$$\mathbf{s}(\mathbf{x}) = \begin{cases} s_X(\mathbf{x}) = \sum_{i=1}^N \gamma_i^X (\|\mathbf{x} - \mathbf{x}_{si}\|) + \beta_0^X + \beta_1^X \mathbf{x}^X + \beta_2^X \mathbf{x}^X + \beta_3^X \mathbf{x}^X \\ s_Y(\mathbf{x}) = \sum_{i=1}^N \gamma_i^Y (\|\mathbf{x} - \mathbf{x}_{si}\|) + \beta_0^Y + \beta_1^Y \mathbf{x}^Y + \beta_2^Y \mathbf{x}^Y + \beta_3^Y \mathbf{x}^Y \\ s_Z(\mathbf{x}) = \sum_{i=1}^N \gamma_i^Z (\|\mathbf{x} - \mathbf{x}_{si}\|) + \beta_0^Z + \beta_1^Z \mathbf{x}^Z + \beta_2^Z \mathbf{x}^Z + \beta_3^Z \mathbf{x}^Z \end{cases} \quad (2.5)$$

Che viene applicato a tutti i punti di cui si vuole modificare la posizione:

$$\mathbf{x}_{new} = \mathbf{x}_{old} + \mathbf{s}(\mathbf{x}_{old}) \quad (2.6)$$

2.4 Standard Triangulation Language: STL

Lo *Standard Triangulation Language (STL)* è un formato comunemente utilizzato per la stampa 3D e la progettazione assistita da computer (CAD). Tale formato è stato creato da *3D Systems* nel 1987 nell'ambito dello sviluppo della tecnologia di stampa stereolitografica per stampanti 3D commerciali. Da allora il formato di file STL non è cambiato molto e attualmente è considerato lo standard per la stampa 3D.

Un file *.stl* rappresenta un solido la cui superficie è stata discretizzata in triangoli. Esso consiste nelle coordinate X, Y e Z ripetute per ciascuno dei

tre vertici di ciascun triangolo e in un vettore per elemento per descrivere l'orientazione della normale alla superficie. Più è complessa la superficie, più sono i triangoli prodotti. Il formato STL risulta molto facile da generare e da processare, ma presenta una geometria approssimata e la sua struttura dati, che pur risultando semplice, può presentare la ripetizione dello stesso vertice più volte.

Gli STL non contengono colori e texture, quindi tendono ad essere di piccole dimensioni garantendo comunque tempi di elaborazione più veloci rispetto ad altri tipi di file. Questa caratteristica è vantaggiosa per la creazione di prototipi, ma rappresenta un enorme svantaggio per la creazione del prodotto finale. Per questo motivo raramente si utilizzano file in formato STL per i prodotti finali [1].

2.5 Approccio Statistico

Un modello è un'approssimazione della realtà. Tramite di esso, inteso come equazione matematica, si può prevedere qualcosa senza doverlo misurare.

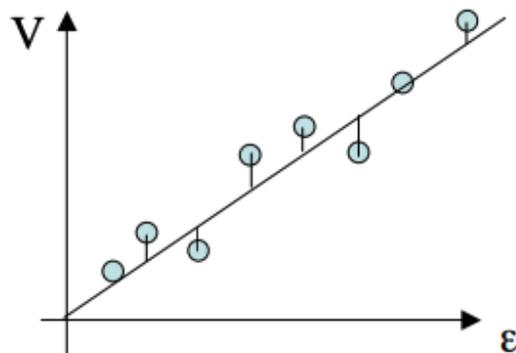


Figura 2.3: Dati sperimentali e modello lineare

Per capire se il modello generato descrive una buona approssimazione del fenomeno reale si effettuano *analisi statistiche* per calcolare quanto un dato sperimentale si discosti da esso.

2.5.1 Distribuzioni univariate

Con il termine *varianza* si definisce quella funzione che fornisce una misura della variabilità dei valori assunti da una variabile. Il termine, introdotto per la prima volta nel 1918 da Ronald Fisher [27], sostituisce *deviazione standard quadratica* e viene spesso indicato con σ . Più precisamente, fornisce la misura di quanto i valori di una distribuzione si discostano quadraticamente dalla media aritmetica o dal valore atteso.

In una distribuzione univariata, data una popolazione N , si definiscono la media:

$$\mu = \frac{1}{N} \sum_{i=1}^N X_i \quad (2.7)$$

La deviazione standard:

$$\sigma_X = \sqrt{\frac{\sum_{i=1}^N (X_i - \mu)^2}{N}} \quad (2.8)$$

E la varianza:

$$\sigma^2_X = (\sigma_X)^2 \quad (2.9)$$

La varianza è invariante per traslazione e cambia quadraticamente per riscaldamento:

$$\sigma^2_{aX+b} = a^2 \sigma^2_X \quad (2.10)$$

La varianza della somma (o della differenza) di due variabili *indipendenti* o *incorrelate* è pari alla *somma* delle loro varianze:

$$\sigma_{X+Y}^2 = \sigma_X^2 + \sigma_Y^2 \quad (2.11)$$

Se X e Y non sono eventi indipendenti, la formula viene corretta utilizzando il concetto di *covarianza*:

$$\sigma_{X+Y}^2 = \sigma_X^2 + \sigma_Y^2 + 2\sigma_{X,Y} \quad (2.12)$$

Dove:

$$\sigma_{X,Y} = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y}) \quad (2.13)$$

Nella *teoria della probabilità*, la funzione di probabilità di una variabile casuale discreta X è una funzione che assegna ad ogni valore possibile di X la probabilità dell'evento elementare. Per variabili casuali continue non ha senso parlare di probabilità di un valore esatto della variabile, ma al più di probabilità che la variabile *cada all'interno di un range di valori*. La teoria delle probabilità introduce quindi sull'insieme A la funzione di *densità di probabilità* $p_x(x)$ tale che $P(X \in A) = \int_A p_x(x)dx$. L'area sottesa dalla funzione densità di probabilità è sempre pari ad 1 (perché è il 100% della probabilità). Bisogna sottolineare che in caso di variabili casuali continue non ha senso parlare di *probabilità di ottenere esattamente un certo valore*, ma occorre necessariamente specificare un intervallo del tipo $[x - \delta < X < x + \delta]$.

Con *distribuzione normale* si intende la distribuzione di Gauss ed è la più

comune dei fenomeni aleatori. La sua densità di probabilità è definita da:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (2.14)$$

Due punti A e B sono a *isoprobabilità* se sono a distanza uguale dalla media.

La larghezza di una distribuzione standard è misurata dalla sua deviazione standard: il 99,7% dei punti di un processo distribuito secondo una gaussiana ricadono all'interno dell'intervallo $[\mu - 3\sigma; \mu + 3\sigma]$:

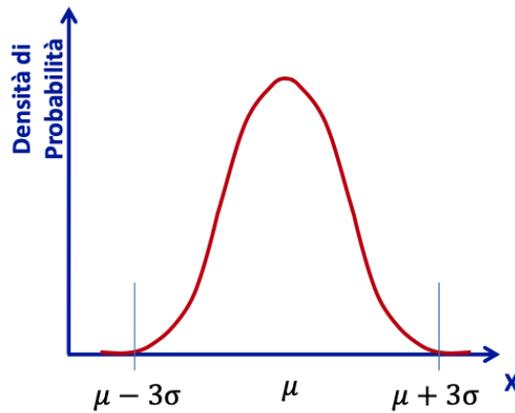


Figura 2.4: La varianza come indicazione della "larghezza" di una distribuzione di dati

2.5.2 Distribuzioni multivariate

Se per distribuzioni univariate i principali descrittori sono degli scalari (media e varianza), per distribuzioni *multivariate* (i.e. distribuzioni normali a più dimensioni) quei descrittori si evolvono in grandezze vettoriali e matriciali.

Otteniamo infatti la media:

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{pmatrix} \quad (2.15)$$

E la *matrice di covarianza*

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \dots \\ \sigma_{21} & \sigma_{22} & \dots \\ \vdots & \vdots & \sigma_{nn} \end{bmatrix} \quad (2.16)$$

In una distribuzione multivariata, la distribuzione di covarianza $\boldsymbol{\Sigma}$ definisce il grado di correlazione fra le variabili. Il luogo dei punti di isoprobabilità è un'ellisse ottenuta come forma quadratica avente come matrice la covarianza.

In un problema di regressione lineare multipla (acr. MLR), la stima delle variabili \mathbf{x} si ottiene invertendo la matrice dei valori osservabili \mathbf{Y} . L'inversione è possibile solo se il rango di \mathbf{Y} è massimo, ovvero se il numero di colonne linearmente indipendenti corrisponde al numero di colonne totali. Questo significa che la matrice è invertibile se e solo se tutti gli osservabili sono indipendenti tra loro.

Si definisce *colinearità* una parziale dipendenza: sono colineari le variabili che dipendono da alcune caratteristiche del campione. Nel caso di non colinearità, la matrice di covarianza è diagonale:

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_n \end{bmatrix}$$

Si può trasformare la matrice di covarianza in una forma diagonale cambiando (adeguatamente) il sistema di riferimento. Rimuovere la colinearità, quindi, significa ridurre la matrice di covarianza in forma diagonale introducendo delle nuove variabili latenti.

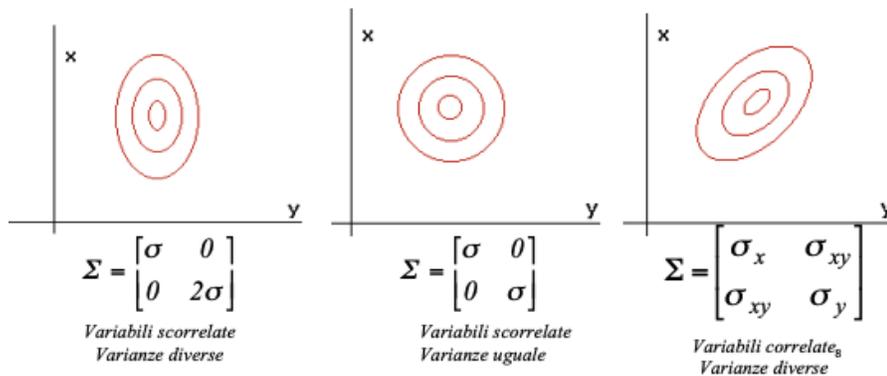


Figura 2.5: Matrici di covarianza e punti di isoprobabilità in 2D

2.5.3 Singular Value Decomposition: SVD

Molti sistemi o fenomeni sono caratterizzati da dati che presentano dei pattern dominanti, i quali possono essere descritti da un sistema a un più basso numero di dimensioni. Un esempio sono le immagini, che possono essere compresse e quindi rappresentate da un numero di pixel più basso dell'originale. La *Singular Value Decomposition (SVD)* fornisce un modo sistematico per

determinare un'approssimazione a dimensione minore di un certo insieme di dati in termini di pattern dominanti

Generalmente in un esperimento si effettuano delle misurazioni, e queste vengono inserite in un vettore \mathbf{x} . Inserendo tutte le misurazioni come vettori colonna in una matrice, si ottiene una matrice $n \times m$ del tipo:

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_m \\ | & | & & | \end{bmatrix}$$

Dove le colonne sono chiamate anche *snapshots* ed hanno dimensione n (che corrisponde al numero di misurazioni), mentre m è il numero di campioni. La SVD è una matrice (unica) tale che¹:

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^* \quad (2.17)$$

Dove:

- \mathbf{U} è una matrice quadrata *unitaria* (ovvero tale che la sua trasposta è uguale all'inversa e che $\mathbf{A}\mathbf{A}^* = \mathbf{A}^*\mathbf{A} = \mathbb{1}$) di dimensione $n \times n$
- \mathbf{V} è una matrice $m \times m$
- $\mathbf{\Sigma}$ è una matrice diagonale a valori non-negativi

Di solito la matrice \mathbf{X} non è quadrata e molto spesso si hanno un numero di misurazioni maggiore del numero di campioni ($n > m$). Quindi al massimo $\mathbf{\Sigma}$ avrà m zeri sulla diagonale. In questo caso conviene rappresentare \mathbf{X} utilizzando il cosiddetto *economy* SVD:

¹Per matrici a valori reali, il simbolo $*$ indica la normale trasposta per cui $X^* = X^T$

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^* = \begin{bmatrix} \hat{\mathbf{U}} & \mathbf{U}^\perp \end{bmatrix} \begin{bmatrix} \hat{\mathbf{\Sigma}} \\ \mathbf{0} \end{bmatrix} \mathbf{V}^* = \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\mathbf{V}^* \quad (2.18)$$

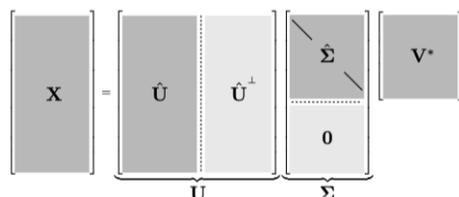


Figura 2.6: Full SVD

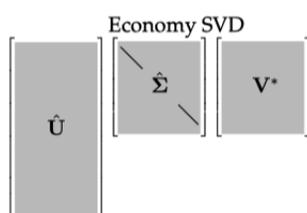


Figura 2.7: Economy SVD

2.5.4 Principal Component Analysis: PCA

Il sistema di riferimento adeguato a scrivere la matrice di covarianza 2.16 in forma diagonale corrisponde agli autovettori della matrice $\mathbf{\Sigma}$, cioè agli assi principali dell'ellisse costruita come forma quadratica della matrice di covarianza stessa. Le nuove variabili non sono più degli osservabili fisici (oggetto di misurazioni), ma sono *combinazioni lineari* di queste. Le nuove variabili prendono il nome di *Componenti Principali* e l'insieme delle procedure di calcolo e di interpretazione delle componenti principali si chiama *Analisi delle Componenti Principali* (PCA), il cui scopo (tra gli altri) è proprio la rimozione della colinearità.

In caso di non colinearità, la probabilità che accada l'evento x insieme all'evento y è uguale al prodotto delle probabilità che accadano i singoli eventi:

$$p(x, y) = p(x) \cdot p(y) \quad (2.19)$$

Mentre ciò non accade in presenza di colinearità:

$$p(x, y) \neq p(x) \cdot p(y) \quad (2.20)$$

Lo scopo dell'analisi delle componenti principali, più in generale, è la rappresentazione di un insieme di dati con colinearità (quindi caratterizzati da una matrice di covarianza non diagonale e di dimensione N) in uno spazio di dimensione minore di N , ma in cui gli stessi dati sono rappresentati da una matrice di covarianza diagonale. La diagonalizzazione si ottiene con una rotazione delle coordinate nella base degli autovettori (i.e. le componenti principali).

Gli autovettori corrispondenti a variabili originariamente correlate in modo parziale tra loro possono essere trascurati (perché i rispettivi autovalori sono molto piccoli). Si può quindi limitare la rappresentazione agli autovettori con autovalori più grandi.

La PCA si basa sull'ipotesi che la variabile \mathbf{x} sia distribuita in modo normale ed è solo uno dei modelli per ridurre le dimensioni dei problemi statistici. A livello geometrico consiste nella proiezione ortogonale dello spazio originale sullo spazio delle componenti principali.

Si definiscono *scores* le nuove coordinate dei vettori corrispondenti alle os-

servazioni nella base delle componenti principali, mentre si chiamano *loadings* i coefficienti delle combinazioni lineari che definiscono le componenti principali. Questi ultimi forniscono una misura del contributo di ogni osservabile alle componenti principali.

Scelta del numero delle componenti principali

Partendo da N variabili (x_1, x_2, \dots, x_N) , a seconda del rango della matrice Σ si potranno trovare un massimo di N componenti principali. Tuttavia il compito dell'analisi delle componenti principali è quello di analizzare un numero di dati inferiore a quello di partenza, quindi bisogna passare dalla dimensione N a k , con $k < N$. I criteri adoperati per la scelta del numero di componenti sono tre:

- Prendere solo quelle componenti che rappresentano più del 90% della variabilità complessiva;
- La *Regola di Kaiser*, che consiste nel prendere solo quelle componenti che hanno un autovalore maggiore o uguale ad uno, oppure, equivalentemente, le componenti che hanno varianza maggiore di quella media;
- Attraverso il grafico degli autovalori ordinati in modo decrescente, chiamato anche *Screen Plot*. All'interno del grafico si sceglie il numero di componenti corrispondente al punto di *gomito* della spezzata. Secondo la figura 2.8, la brusca variazione di pendenza si ha con la terza componente principale. Pertanto, in questo caso, bisogna scegliere proprio le prime tre componenti principali.

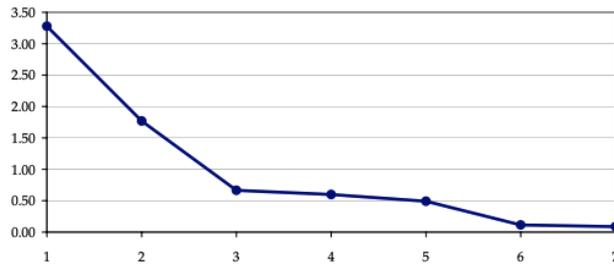


Figura 2.8: Esempio di Screen Plot

Interpretazione delle componenti principali

Interpretare le componenti principali è molto complicato e spesso dipende dalla capacità, dall'esperienza e dalla sensibilità di chi effettua lo studio. Si è già detto che le componenti principali non sono più degli osservabili, ma combinazioni lineari di essi e tali combinazioni lineari sono caratterizzate da coefficienti chiamati *loadings*. Il generico *loading* rappresenta pertanto il peso che ciascuna variabile ha nella determinazione della particolare componente principale. Ciò significa che una componente principale sarà maggiormente caratterizzata da una certa variabile i -esima a cui corrisponde il coefficiente i -esimo maggiore in valore assoluto. L'interpretazione delle *PCs* (*Principal Components*) viene quindi effettuata di solito sulla base dell'osservazione della matrice di correlazione tra le variabili originarie e le componenti stesse.

Nonostante l'elevata potenza (ed utilizzo) dell'analisi PCA, lo svantaggio principale riguarda proprio l'interpretazione dei risultati, in quanto l'analisi dei *loadings* è condotta soggettivamente dal ricercatore che può ritenere significativi alcuni valori arbitrariamente.

2.5.5 Validazione del modello: LOOCV

Per valutare le prestazioni di un modello su un insieme di dati è necessario misurare la corrispondenza tra le previsioni fatte dal modello e i dati osservati.

Il modo più comune per misurarlo è l'errore quadratico medio (*Mean Squared Error*, MSE), calcolato come: $MSE = \frac{1}{N} \sum_{i=1}^N (x_i - f(x_i))^2$, dove N è il numero totale di misurazioni, x_i è il valore della misurazione i -esima e $f(x_i)$ è il valore previsto dal modello della misurazione corrispondente. Maggiore sarà la fedeltà del modello alle osservazioni, minore sarà il valore MSE. Questo test dà un'idea di quanto un modello è capace di prevedere un insieme di campioni sconosciuti (non utilizzati per costruire il modello). Tuttavia questo test può variare molto in funzione di quali osservazioni sono state utilizzate nel *training* e *testing set*². Un modo per evitare questo problema consiste nell'adattare il modello più volte, utilizzando ogni volta un *training* e *testing set* diverso, calcolando successivamente l'MSE del test come la media di tutti gli MSE di ciascun test. Questo metodo è noto come *Leave-One-Out Cross-Validation (LOOCV)*.

Secondo questo particolare test di convalida, il numero di iterazioni è uguale al numero di campioni contenuti nel set di dati. In questo modo l'algoritmo di apprendimento viene applicato una volta per ogni campione utilizzando tutte le altre istanze come *training set* e utilizzando l'istanza selezionata come *test set* a singolo elemento [19].

²Il *training set* è l'insieme dei dati utilizzati per costruire il modello, mentre il *testing set* è l'insieme dei dati utilizzati per testare il modello

2.6 Statistical Shape and Appearance Modeling

Con *Statistical shape models* (SSM) e *Statistical appearance models* (SAM) si intendono due tipologie di modelli statistici che descrivono rispettivamente la forma e la distribuzione di densità media di una certa popolazione di dati. Generalmente gli SSAM sono formati da due componenti: la forma (o la distribuzione di densità) e i principali modi di variazione dai valori medi. La SSA (Statistical Shape Analysis) trova applicazione in diversi campi tra cui l'imaging medico, l'anatomia computazionale, e la profilazione geografica.

2.6.1 Statistical shape models

La metodologia classica implementata per progettare dispositivi *patient-specific* o per personalizzare operazioni chirurgiche si basa su un approccio interattivo di *design* ed è composto da tre passi principali:

1. Ingegneria inversa per ottenere il modello 3D della regione anatomica d'interesse e per preparare un ambiente di lavoro virtuale;
2. Modellazione 3D all'interno dell'ambiente di lavoro virtuale dove è richiesta la collaborazione tra i vari campi di interesse (medico, ingegneristico, biomeccanico) per il settaggio dei parametri;
3. Simulazione e (se necessario) prototipazione tramite tecniche di *additive manufacturing*.

Fondamentale per l'implementazione di questo approccio è la disponibilità del modello 3D che rappresenta la geometria di interesse. SSM e SAM sono generalmente creati sulla base di un set di dati *training* che forniscono al modello una conoscenza *a priori* della forma e della distribuzione della densità delle ossa. La possibilità di applicare SSM e SAM a immagini inedite crea molte opportunità che potrebbero essere sfruttate per facilitare lo studio, la diagnosi e il trattamento delle malattie dello scheletro umano. Ad esempio, la SSM tridimensionale può essere utilizzata per generare modelli ossei tridimensionali *patient-specific* a partire da un insieme di immagini 2D. Quando un modello osseo 3D viene ricostruito correttamente da una serie di immagini 2D utilizzando la SSM tridimensionale, si ottiene un enorme vantaggio: il livello di dettaglio normalmente ottenuto dall'imaging 3D al prezzo dell'imaging 2D. L'uso di modalità di imaging 2D comporta una significativa riduzione dei costi, una minore esposizione dei pazienti alle radiazioni e, in definitiva, molte nuove opportunità per migliorare la diagnosi e il trattamento.

Per gli SSM 2D si utilizzano generalmente immagini a *Raggi-X* o di *assorbimetria a Raggi-X a doppia energia (DXA)*. Gli SSM 3D sono spesso costruiti utilizzando immagini di *tomografia computerizzata (CT)* o *risonanza magnetica (MRI)*.

Il primo passo per ottenere la forma media e i principali modi di variazione è trovare un sistema per rappresentare virtualmente le forme ossee. Il più semplice e allo stesso tempo il più generico consiste nell'utilizzare un set di punti distribuiti sulla superficie. In letteratura questi punti sono chiamati *landmarks*.



Figura 2.9: Landmarks posizionati manualmente

Le coordinate dei k punti vengono organizzate in un unico vettore \mathbf{x} che descrive la forma dell'osso:

$$\mathbf{x} = (x_1, y_1, z_1, \dots, x_k, y_k, z_k)^\top \quad (2.21)$$

I *landmarks* non devono essere necessariamente fissati sui punti notevoli delle caratteristiche dell'osso, come invece è richiesto dalla definizione propria del termine in ambito anatomico, e per sottolineare questa differenza molti autori si riferiscono ad essi come *semi-landmarks* [11]. Spesso inoltre si sfrutta la connettività dei punti la cui informazione è racchiusa nelle mesh. Per l'utilizzo di landmarks quale base di modelli SSM, *Cootes et al. (1992)* hanno coniato l'espressione *Point Distribution Models (PDMs)*, diventato piuttosto popolare in letteratura; oggi la maggior parte dei modelli di forma sono basati proprio su di essi e su questa terminologia³.

Costruire un modello di *statistical shape* consiste fundamentalmente nel-

³Altre metodologie per rappresentare delle forme biologiche sono i *medial models*, l'utilizzo di *superfici di Fourier* (estensioni delle classiche trasformate di Fourier) o la tecnica delle *armoniche sferiche (SPHARMs)*.

l'estrarre la forma media da una quantità di modi di variazione da una collezione di campioni di *training*. Il metodo impiegato dipende fortemente dalla forma da rappresentare.

Si definisce *forma* quella proprietà che non cambia sotto trasformazioni di similarità, ovvero che è invariante alla traslazione, rotazione e *scaling*. Per alcune applicazioni, cambiamenti di grandezza possono essere trattati come parte del processo di variazione biologica, ma generalmente si vogliono rimuovere queste trasformazioni di similarità, quindi il primo passo è quello di allineare tutti i campioni in un comune sistema di riferimento.

Il metodo più popolare per risolvere questo problema di *allineamento* è quello conosciuto come *Generalized Procrustes Alignment (GPA)*. Alla base del processo c'è un'operazione di minimizzazione della distanza media quadratica tra due forme permettendone il calcolo analitico. Per allineare l'intero gruppo di campioni questa procedura viene ripetuta iterativamente, e questo va a definire il GPA.

Eseguito l'allineamento, lo step successivo è quello di ridurre la dimensione del *training set*, i.e. trovare un più piccolo set di modi che meglio descrivono la variazione osservata. Spesso per ottenere questo risultati si effettua un'analisi alle componenti principali. Ricordando (2.21), ogni forma allineata è descritta da $3k$ punti di coordinate nel vettore \mathbf{x}_i . Quindi la forma media può essere trovata semplicemente facendo la media degli N campioni

$$\mathbf{x} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i \quad (2.22)$$

E quindi la matrice di covarianza è data da:

$$\Sigma = \frac{1}{N-1} \sum_{i=1}^N (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top \quad (2.23)$$

L'analisi delle componenti principali viene successivamente eseguita sulla matrice di covarianza per calcolare le modalità di variazione più importanti rispetto alla forma media ϕ . Le modalità principali di variazione sono ordinate in ordine decrescente rispetto alla percentuale di variazione della forma descritta dal modo stesso. Pertanto, l'importanza delle prime componenti nella descrizione delle forme di altri individui all'interno della popolazione è molto maggiore rispetto all'importanza delle altre modalità⁴. Quindi, una volta trovati gli autovettori ϕ_m e i rispettivi autovalori λ_m , si ordinano questi ultimi in ordine decrescente $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{N-1}$. Successivamente è possibile approssimare ogni forma valida mediante una combinazione lineare dei primi c modi:

$$\mathbf{x} = \bar{\mathbf{x}} + \sum_{m=1}^c b_m \phi_m \quad (2.24)$$

Nel vettore \mathbf{b} della (2.24) sono raccolte le informazioni sui parametri di forma. Per non avere variazioni inverosimili e irreali, tale vettore deve essere limitato in un certo intervallo. Un metodo comune è quello di trattare tutti i modi di variazione come distribuzioni indipendenti e definire ciascun b_m nell'intervallo $[-3\lambda_m; 3\lambda_m]$. Se la variabilità della forma non segue una distribuzione gaussiana, i limiti appena dati al vettore \mathbf{b} possono portare a delle forme non valide.

⁴A volte, invece di trovare gli autovalori e rispettivi autovettori della matrice di covarianza, si possono calcolare gli stessi con una SVD sulla matrice dei *landmark* allineati, definita come $\mathbf{L} = ((\mathbf{x}_1 - \bar{\mathbf{x}}) \dots (\mathbf{x}_N - \bar{\mathbf{x}}))$

La potenza di un modello statistico risiede nella quantità di dati disponibili. Nel caso di modelli SSM 3D questa quantità è la maggior parte delle volte insufficiente, in quanto l'ottenimento pratico di immagini e di segmentazione manuale sono onerosi e richiedono molto tempo. Anche quando ci sono sufficienti dati di training, potrebbero non essere utilizzati tutti completamente a causa di una stima troppo scarsa della grandezza dei campioni necessaria. Questo porta alla conclusione che i modelli risultano eccessivamente limitati e quindi le restrizioni imposte sulle deformazioni non gli permettono di adattarsi adeguatamente ai nuovi dati.

Effettuare un modello statistico di un'insieme di forme richiede quindi un training dataset appropriato con una ben definita corrispondenza. Il posizionamento manuale dei landmarks è oggigiorno sempre più evitato sia per modelli 2D che per quelli 3D perché, oltre al fatto che è dispendioso di tempo e richiede un elevato livello di competenza ed esperienza, manca soprattutto di riproducibilità. Quindi si utilizzano degli algoritmi che automaticamente calcolano ed elaborano questa corrispondenza:

- *Mesh-to-mesh registration*, in cui si lavora direttamente sulle mesh di training, alla cui base c'è il metodo dell'*Iterative Closest Point (ICP)*. Fondamentalmente si estraggono delle superfici dalle immagini di training e successivamente un template di mesh con landmarks viene adattato a queste superfici distribuendo i propri landmarks su tutti i campioni;
- *Mesh-to-volume registration*, per cui, dato che le immagini mediche sono spesso ottenute tramite CT o MRI, i dati originali di rappresentazione

non sono delle mesh ma dei volumi segmentati. Pertanto si generano i landmarks con un approccio che adatta un modello di una superficie deformabile a questi volumi. La posizione finale dei landmarks nelle immagini determina infine la corrispondenza;

- *Volume-to-volume registration*, in cui un atlante volumetrico viene fatto corrispondere a tutte le immagini di training generando un campo di spostamento che viene poi utilizzato per trasformare la mesh template di landmarks (estratta dall'atlante stesso) a tutti i campioni, distribuendo così i landmarks e definendo la corrispondenza.
- *Parameterization-to-parameterization registration*, per cui, dopo aver estratto le superfici da tutte le immagini di training, si crea una mappatura uno-ad-uno in un dominio comune. Si definisce quindi la posizione dei landmarks in questa base comune e quindi i rispettivi punti di corrispondenza vengono direttamente trasferiti sulle superfici di training invertendo la funzione di mappatura;
- *Population-based optimization*, in cui, come nel metodo di *parameterization*, si effettua la mappatura e si posizionano i landmarks nel dominio di base. Tramite un processo iterativo di ottimizzazione, si modifica la funzione di mappatura ottenendo il risultato di cambiare le corrispondenze. Questo processo viene ripetuto finché la funzione che descrive la qualità del modello converge al miglior valore ottenibile.

Dati i numerosi metodi per ottenere corrispondenza tra immagine di training e forma ottenuta tramite posizionamento di landmarks, è importante

saper valutare quale sia il migliore. Tuttavia questa valutazione è ostacolata dal fatto che generalmente la reale corrispondenza delle forme biologiche non è nota. Ci sono fondamentalmente tre misure per valutare la qualità generale di un modello di *statistical shape*: capacità di generalizzazione, specificità e compattezza. La prima quantifica la capacità di un modello di rappresentare nuove forme; la seconda descrive la validità delle forme generate dal modello; la terza misura semplicemente la varianza cumulativa del modello.

2.6.2 Statistical appearance models

I modelli SSM forniscono informazioni utili sulla forma delle ossa, ma non contengono informazioni sulla distribuzione della densità all'interno. I modelli SAM possono descrivere la distribuzione media della densità all'interno delle ossa e le principali modalità di variazione di tale distribuzione rispetto ai valori medi.

Come gli SSM, gli SAM devono essere ottenuti da dati campionati. In sostanza, gli *appearance models* si basano su caratteristiche di *bordo* o di *regione*.

Caratteristiche di bordo

L'idea alla base dei primi modelli SAM era quella di campionare i profili perpendicolarmente alla superficie in tutte le immagini di training e - come nei modelli di SSM - estrarre il profilo medio e i principali modi di variazione di ciascun landmark. Tuttavia uno dei problemi più comuni agli *appearance models* basati sulla matrice di covarianza riguardano il fatto che i dati cam-

pionati devono seguire idealmente una distribuzione gaussiana, ma questo non accade quasi mai. Per questo motivo sono state proposte delle soluzioni non lineari.

Caratteristiche di volume

Il metodo più comune che utilizza caratteristiche di volume è quello dell'*Active Appearance Models (AAM)*. Qui, l'intera regione interna della forma di un modello viene utilizzata per costruire un vettore delle caratteristiche \mathbf{g} che nel più semplice dei casi conserva gli elementari valori di intensità dei pixel della regione considerata. Per costruire tale vettore per forme differenti, la rispettiva regione deve essere trasformata in una prima forma standard (di solito è la forma media ottenuta con una SSM). L'*appearance model* è in seguito generato con una PCA sulla matrice di covarianza delle texture delle immagini (i.e. intensità dei pixel). Questo metodo si rivela molto utile quando si lavora con immagini dove la distribuzione dei dati non segue una distribuzione gaussiana (e.g. nelle immagini ad ultrasuoni).

2.6.3 Applicazioni mediche

Le applicazioni più importanti di modelli SSM 3D per l'analisi di immagini medicali sono le seguenti:

- La segmentazione 3D, che consiste nell'analisi e l'identificazione (*labeling*) di immagini volumetriche. È l'applicazione più importante dei modelli statistici e grazie all'avvento del *landmarking* automatico ha subito un enorme sviluppo;

- L'analisi di forme, ovvero la ricerca della correlazione tra le forme di particolari strutture e degli attributi *patient-specific* ottenibili da studi clinici;
- L'estrapolazione di forme da dati 3D non organizzati per ottenere una rappresentazione più dettagliata. Spesso tale compito è richiesto da qualche occasione di telemetria intra-operatoria.

Capitolo 3

Analisi

L'obiettivo di questo capitolo è descrivere l'approccio e le operazioni effettuate per la creazione del modello statistico delle proprietà meccaniche del corpo vertebrale di L1 a partire da uno *statistical shape model* effettuato in collaborazione con il progetto *Spinner-EID* dell'Università di Sheffield. Tutte le analisi sono state condotte in *MATLAB* (v. R2022a, *The MathWorks, Inc., Natick, MA, USA*) utilizzando codici scritti dall'autore.

3.1 Raccolta dati

Per lo studio sono stati utilizzati i corpi vertebrali della prima vertebra lombare di 16 pazienti anonimi. Le immagini cliniche dei pazienti derivano dal dataset pubblico *VerSe2020*¹. Originariamente in formato *.NiFTI*, non supportato da *Bonemat* (il software utilizzato per la mappatura delle caratteristiche meccaniche di ciascun corpo vertebrale), sono state convertite in *.DICOM* mediante l'utilizzo del software gratuito ed open-source *3D-Slicer*.

¹Sito ufficiale di VerSe2020: <https://osf.io/t98fz/>

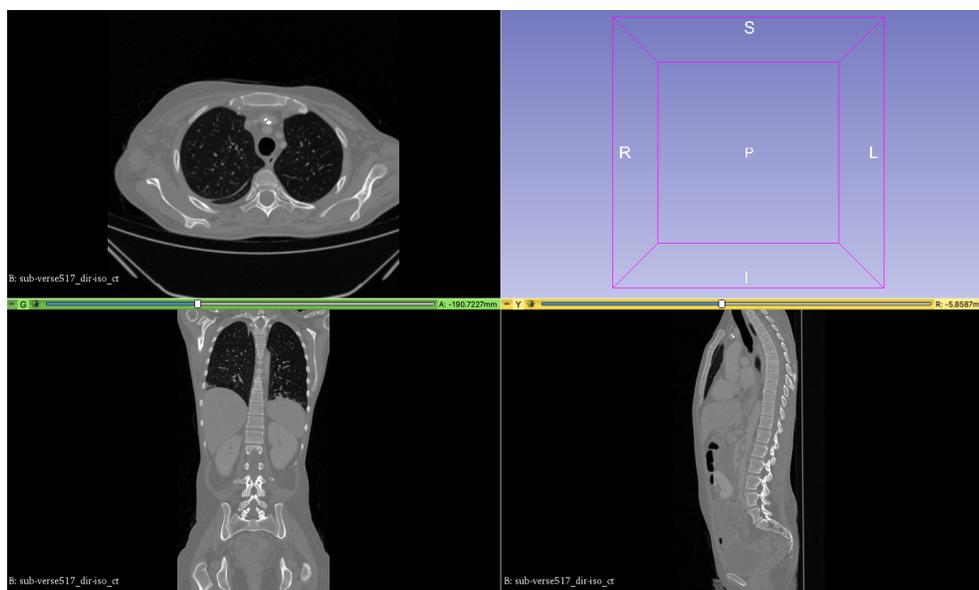


Figura 3.1: Immagine tomografica del *paziente 1* in 3D-Slicer

Questo dataset è stato precedentemente processato dal dott. ing. Marco Sensale del progetto Spinner-EID che ne ha estratto il corpo vertebrale, ne ha effettuato il processo di allineamento ed infine ha effettuato il morphing utilizzando l'algoritmo del *Free Form Deformable Models (FFDM)*. Da queste mesh, in seguito alla creazione di un modello SSM, ne è stata estratta una dalla forma media caratterizzata da 25524 elementi e 12764 nodi ottimizzata per minimizzare l'errore di morphing², a cui ci si riferirà da qui in avanti con *mesh di reference* e dalla quale si è partiti per effettuare la creazione del modello statistico della distribuzione di densità.

²Per approfondire questo e altri limiti del seguente studio si rimanda a § 4.1.

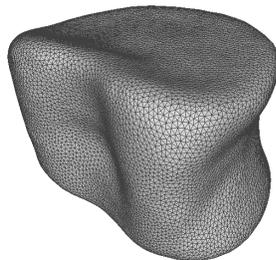


Figura 3.2: File .stl della mesh di reference

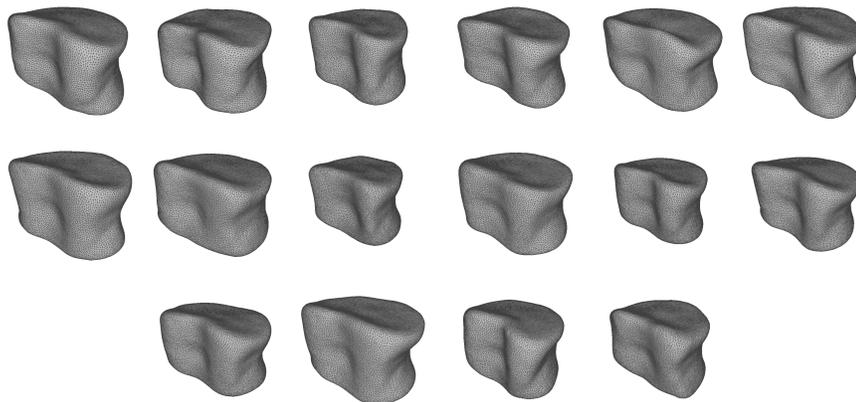


Figura 3.3: STL dei corpi vertebrali dei pazienti

3.2 Meshing

Ai fini della caratterizzazione meccanica, occorre convertire le mesh dal formato .stl (dunque di tipo superficiale) in mesh volumetriche (nel formato .cdb). Ci sono numerosi software disponibili che permettono l'analisi FEM: quello utilizzato in sede dello studio descritto in questa tesi è *ANSYS®2021R1*. La mesh di reference in formato .stl è stata importata in

Space-Claim e ne è stata eseguita una *patch*, la quale è stata successivamente elaborata in *Mechanical* per ottenere una mesh volumetrica tetraedrica di ordine quadratico caratterizzata da un *element size* di 2 mm. La mesh ottenuta è composta da 48122 nodi e 33585 elementi.

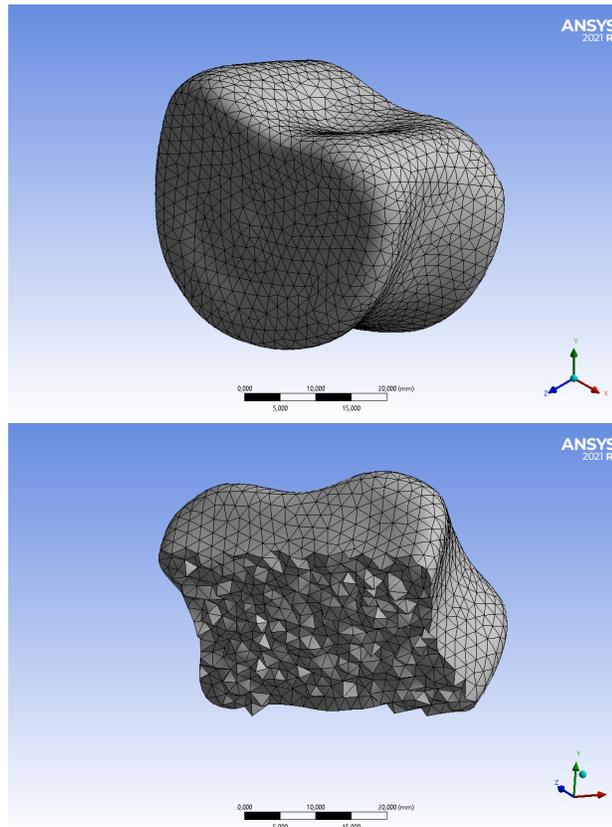


Figura 3.4: Reference dopo il meshing in Ansys: vista isometrica e in sezione

3.3 Morphing per l'isotopologia

Affinché il modello statistico sulle proprietà meccaniche sia correttamente creato tramite l'analisi delle componenti principali, uno degli approcci più

Tabella 3.1: Morph From File

X_1	Y_1	Z_1	ΔX_1	ΔY_1	ΔZ_1
X_2	Y_2	Z_2	ΔX_2	ΔY_2	ΔZ_2
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
X_n	Y_n	Z_n	ΔX_n	ΔY_n	ΔZ_n

utilizzati è quello di rendere preliminarmente le mesh isotopologiche. Il metodo per il morphing utilizzato in questa analisi è quello delle *radial basis functions* spiegato in § 2.3.1: la mesh ottenuta in *Ansys Mechanical* a partire dalla reference è stata adattata a tutti gli altri corpi vertebrali dei pazienti (mesh target) con la tecnologia di *RBF Morph* utilizzando la funzione del *morph from file*. Essa esegue l'adattamento della mesh sorgente a quella target applicando a una nuvola di *control points* uno spostamento in modo tale che essi vadano ad occupare la posizione dei *target points*. In questo caso, la nuvola di *source points* è la stessa per tutti i pazienti ed è l'insieme dei nodi della mesh di reference; i *target points* sono invece i nodi della mesh di ciascun paziente. Il tool del *morph from file* richiede in input una matrice salvata in un file in formato *.txt* composta da tante righe quanti sono i nodi delle mesh (ogni riga corrisponde in questo modo a un *source point*) e sei colonne: le prime tre corrispondono alle coordinate X , Y , Z del punto di sorgente, mentre le altre allo spostamento relativo necessario a spostarsi nella posizione del corrispettivo *target point*:

Ottenuta tale matrice per ciascun paziente si è eseguito il *morphing*. Dalla figura 3.5 si possono osservare i *control points* (rappresentati in rosso) e i *target points* (rappresentati in blu).

Le mesh volumetriche appena ricavate (caratterizzate in conclusione tutte

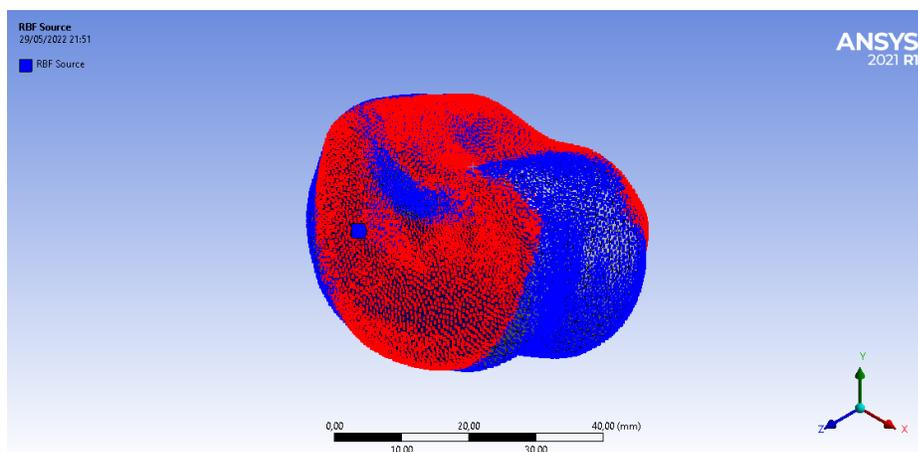


Figura 3.5: Preview del mesh morphing del paziente 13

da 48122 nodi e 33585 elementi) sono state esportate in formato *.cdb* tramite l'*APDL Solver* per la mappatura delle proprietà meccaniche.

3.4 Mappatura delle proprietà meccaniche

Prima di effettuare la mappatura è stato necessario allineare ciascuna mesh alle CT Scan dei rispettivi pazienti. Ai fini della caratterizzazione meccanica è infatti fondamentale che il corpo vertebrale sia perfettamente sovrapposto all'immagine clinica. Grazie alle operazioni preliminari di estrazione del corpo vertebrale del dott. ing. Marco Sensale, è stata ricavata e poi applicata a ciascuna mesh volumetrica una matrice di rototraslazione affinché si spostassero verso la posizione desiderata, andandosi a sovrapporre correttamente alle immagini cliniche. La matrice di rototraslazione è stata ottenuta basandosi sull'orientazione assoluta mediante *metodo di Horn*, che permette di determinare la corrispondenza traslazionale, rotazionale e scalare uniforme

tra due diversi sistemi di coordinate cartesiane, dato un insieme di coppie di punti corrispondenti.

Una volta che le mesh volumetriche sono risultate correttamente sovrapposte alle immagini cliniche dei pazienti, si è passati alla mappatura delle proprietà meccaniche mediante l'utilizzo del software *Bonemat V3.2 (Stable Version)*.

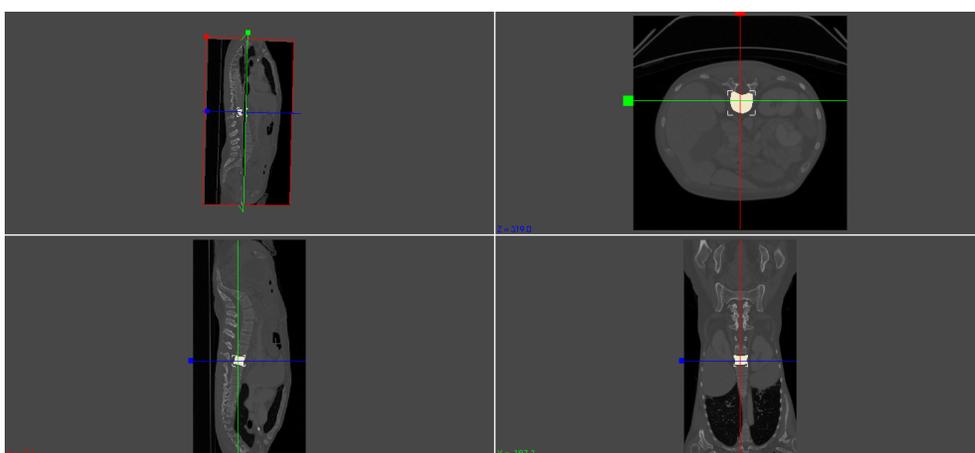


Figura 3.6: Corpo vertebrale del paziente 1 allineato all'immagine CT

L'osso vertebrale è stato considerato un materiale localmente isotropo ed eterogeneo, il cui relativo modulo di Young dipende dalla densità ossea (*Bone Mineral Density, BMD*) locale stimata dalle immagini cliniche. Introduciamo preliminarmente due misure della densità ossea: quella *radiologica* e quella *apparente*. La densità radiologica *QCT (Quantitative Computer Tomography)*, indicata con ρ_{QCT} , deriva direttamente dalla radiodensità degli scan CT; la densità apparente, indicata con ρ_{app} , è calcolata come rapporto tra la massa e il *volume apparente*, ovvero quello ottenuto considerando il volume reale, il volume occupato dai pori e quello occupato dall'acqua

($V_{app} = V + V_p + V_w$). Per lo studio si è supposto che le macchine che hanno effettuato le tomografie computerizzate siano state sottoposte a una calibrazione densitometrica *a-priori*, pertanto le *Hounsfield units*³ sono state considerate equivalenti ai valori della BMD tramite una relazione lineare con un fattore correttivo per la conversione in g/cm^3 . La densità ossea apparente è stata calcolata dalla relazione [21]:

$$\rho_{QCT} = \rho_{app} \times 0.6 \frac{g}{cm^3} \quad (3.1)$$

Il modulo di Young è stato calcolato utilizzando la relazione sperimentale specifica per le vertebre toraco-lombari [16]:

$$E_{bone} = 4730 \rho_{app}^{1.56} \quad (3.2)$$

Il coefficiente di Poisson è stato imposto pari a [29]: $\nu_{bone} = 0.3$.

Infine è stata esportata una mesh volumetrica dove ciascun elemento è caratterizzato, oltre che dalle informazioni sulla geometria e sulla connettività, da un modulo di Young, una densità e un coefficiente di Poisson.

3.5 PCA e generazione del modello SAM

In conclusione dello studio, le mesh volumetriche dei pazienti - contenenti ora le proprietà meccaniche mappate con il procedimento e le relazioni descritte nella sezione precedente - sono state importate in Matlab ed è stata eseguita un'analisi alle componenti principali per trovare i principali modi di

³Con *Hounsfield units (HU)* si intende la scala utilizzata per quantificare la radiodensità. Ci si riferisce anche con *numero CT*

variazione. Dato che la densità ossea è ricavata dalle immagini cliniche e il coefficiente di Poisson è supposto costante per tutti gli elementi della mesh, il proseguimento dello studio è stato condotto caratterizzando meccanicamente ciascun elemento solo con il modulo di Young.

Per ogni corpo vertebrale dei pazienti è stato creato un vettore di dimensione n corrispondente al numero di elementi della relativa mesh. Ogni casella del vettore è occupata dal modulo di Young relativo all' n -esimo elemento. Questi vettori colonna sono stati uniti infine per comporre una matrice $n \times m$, con m pari al numero di pazienti (in questo caso 16) e quest'ultima matrice è stata oggetto dell'analisi PCA.

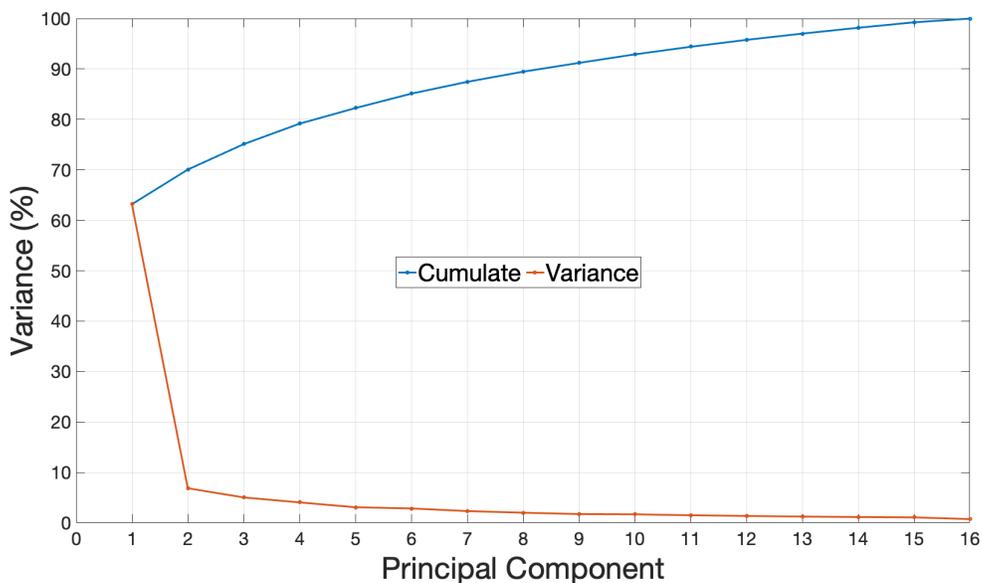


Figura 3.7: Componenti principali e varianza

Dalla figura 3.7 si può osservare la variabilità associata a ciascuna componente principale (indicata con *variance*) e la cumulata (indicata con *cumulate*). Per la scelta dei modi è stato utilizzato il criterio secondo cui si

Tabella 3.2: Varianza cumulata

Componente Principale	Varianza cumulata
1	63.1859
2	70.0542
3	75.0957
4	79.1523
5	82.2469
6	85.1058
7	87.4453
8	89.4580
9	91.2029
10	92.8933
11	94.3992
12	95.7716
13	97.0132
14	98.1706
15	99.2539
16	100.000

utilizzano quelle componenti a cui è associato almeno il 90% della variabilità totale. Il numero di componenti principali è risultato essere dunque pari a *nove*, come evidenziato anche dalla tabella 3.2:

Secondo la teoria alla base dell'analisi delle componenti principali, dato un certo *dataset* iniziale caratterizzato da n misurazioni per m variabili, si otterranno al più m componenti principali. Ogni vettore (di dimensione n) corrispondente a ciascuna variabile può essere scritto come combinazione lineare delle componenti principali: più componenti si utilizzano per questa combinazione lineare, più precisi saranno i risultati. Tuttavia, le singole misurazioni possono essere rappresentate anche (in modo sufficientemente preciso) dai modi principali responsabili della maggiore variabilità. Per questo moti-

vo, avendo 16 pazienti si sono ottenute 16 componenti principali e il primo passo è stato verificare che il risultato della PCA fosse corretto riscrivendo i vettori dei pazienti come combinazione lineare di tutte le componenti principali. Le figure 3.8 e 3.9 mostrano rispettivamente l'errore medio e massimo⁴ ottenuto da questa combinazione lineare.

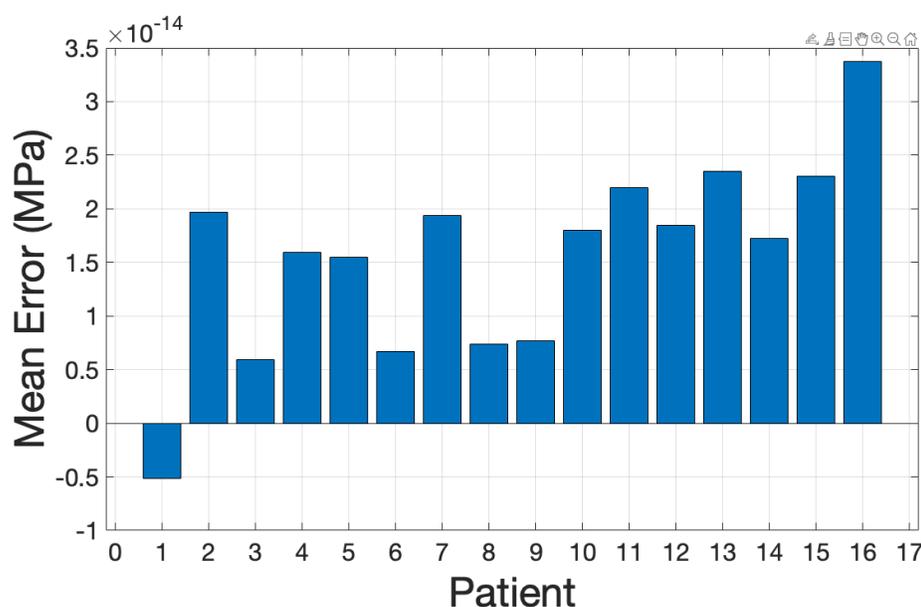


Figura 3.8: Errore medio ottenuto dalla combinazione lineare di tutte le componenti principali

Validazione dell'analisi

Per validare l'analisi effettuata si è utilizzato il criterio del *Leave-One-Out Cross-Validation*. Se il modello statistico ottenuto dall'analisi PCA è valido, significa che è in grado di ottenere con un errore di approssimazione minimo anche le caratteristiche di dati esterni al *dataset di training*. Que-

⁴L'errore è stato calcolato sottraendo ad ogni vettore originale quello ottenuto dalla combinazione lineare delle componenti principali.

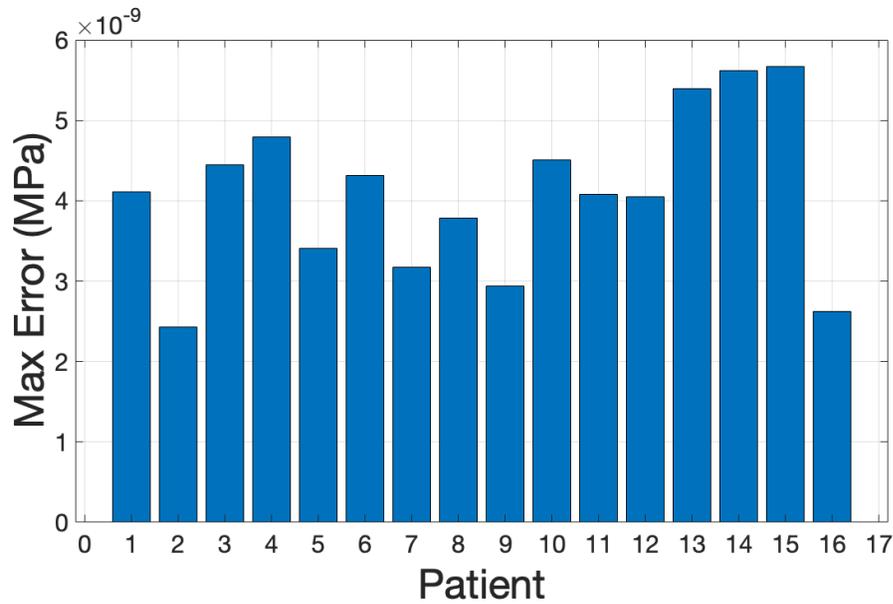


Figura 3.9: Errore massimo ottenuto dalla combinazione lineare di tutte le componenti principali

sto dataset corrisponde, nel nostro caso, ai vettori dei pazienti. Il metodo del *Leave-One-Out* è stato applicato rimuovendo iterativamente ogni paziente dal *dataset* di *training* ed effettuando un'analisi PCA sui rimanenti 15 pazienti. Avendo fissato a priori il numero di modi a nove, ciascun vettore temporaneamente rimosso è stato riscritto come combinazione lineare delle prime nove componenti principali appena trovate ed infine è stato calcolato l'errore risultante. Le figure 3.10 e 3.11 rappresentano gli errori medi e massimi ottenuti in seguito alla combinazione lineare dei nove modi trovati sia utilizzando un dataset ridotto con la validazione della *Leave-One-Out (LOO)* che con l'intero dataset di pazienti.

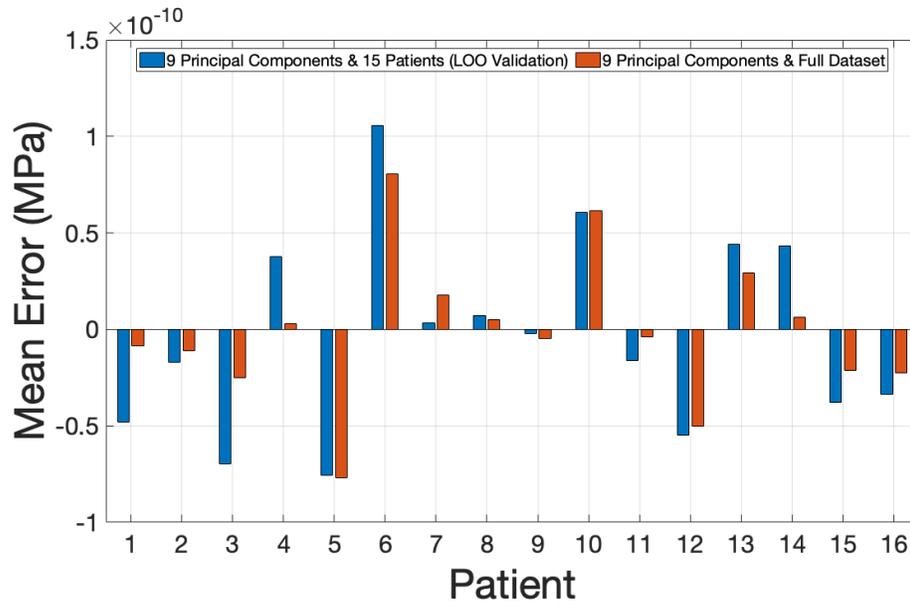


Figura 3.10: Errore medio ottenuto dalla combinazione lineare dei primi 9 modi

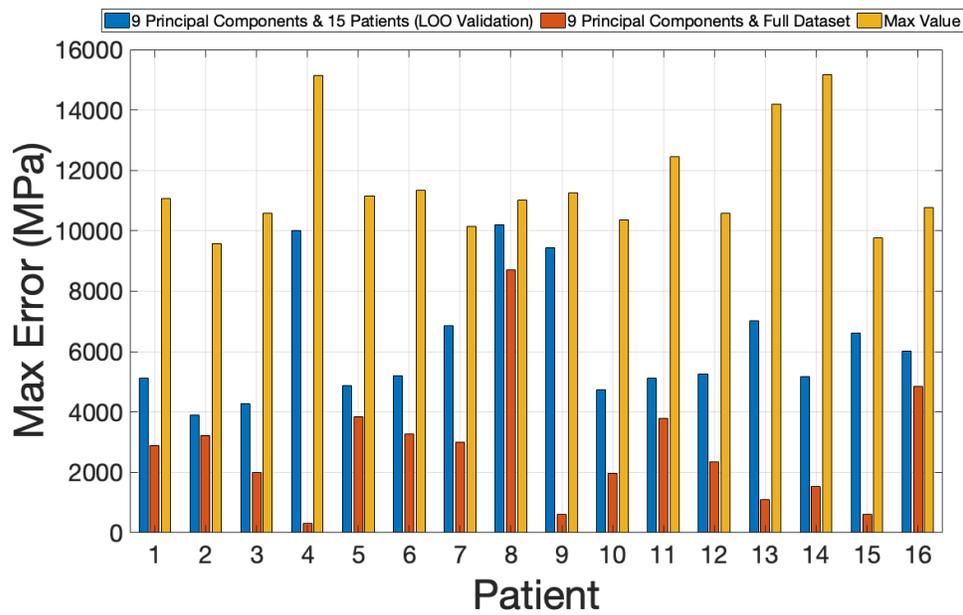


Figura 3.11: Errore massimo ottenuto dalla combinazione lineare dei primi 9 modi

Capitolo 4

Conclusione

4.1 Limiti dell'approccio ed analisi dei risultati

Il limite più grande dell'approccio utilizzato in questo studio riguarda la mappatura delle proprietà meccaniche delle mesh volumetriche effettuata in Bonemat. Per effettuare tale operazione, si sono utilizzate due relazioni: una che lega il modulo di Young con la densità (3.2) e un'altra che lega la densità ossea ai livelli di grigio tramite le *Hansfield units (HU)* (3.1). Quest'ultima relazione dipende fortemente dalla calibrazione che va effettuata sulla macchina che effettua il CT Scan.

Si definisce *coefficiente di attenuazione* la radiodensità di ciascun materiale e si esprime in HU in una scala che va da 0 (per l'acqua distillata in condizioni di temperatura e pressione ambientali) a -1000 (per l'aria). La calibrazione degli scanner CT serve a correggere le variazioni nelle caratteristiche di attenuazione¹. Senza un'appropriata calibrazione, la BMD misurata

¹Ad esempio, le dimensioni del corpo di cui si sta effettuando l'imaging possono alterare tali caratteristiche.

può variare fortemente tra i vari scanner, portando a confusione nell'interpretazione dei dati e quindi riducendo la loro utilità clinica. Generalmente la calibrazione avviene con un fantoccio (*phantom*) di densità nota che va posizionato sotto il paziente durante la scannerizzazione. Questo approccio comporta un aumento dei costi e del peso logistico dell'imaging. Un secondo approccio consiste nel pre-calibrare la macchina utilizzando un phantom e poi usare questa pre-calibrazione per l'imaging dei pazienti. Tuttavia questo metodo presenta un problema: non è patient-specific. Un terzo approccio che si utilizza allora è quello di usare tessuti interni al paziente (spesso quello muscolare) per la calibrazione: è un procedimento *phantomless* ed è patient-specific, ma non è riproducibile, in quanto è difficile scegliere la regione del muscolo in una maniera che sia ripetibile ed è complicato separare in modo coerente i componenti del tessuto puro. Un'altra tecnica, proposta da *Lee et al.* [14], è quella di utilizzare come materiale di calibrazione l'aria esterna (visibile nello scan del paziente) e uno tra il tessuto adiposo o sanguigno adiacente all'osso analizzato. Questa tecnica è risultata molto vantaggiosa in quanto non richiede l'utilizzo di un fantoccio ed è molto più precisa delle precedenti.

Nell'analisi di questa tesi si sono effettuate le stesse ipotesi fatte da *Marco Sensale et al.* in [23], (i.e. che la calibrazione sia stata fatta a priori) e per questo motivo si suppone valida la relazione 3.1. La calibrazione densitometrica delle macchine ha un impatto importante sulla variabilità dei dati e quindi sui risultati della PCA, ma non si ha la certezza di *se e quando*, per il dataset oggetto di questo studio, sia stata effettuata tale calibrazione. Inoltre finora non è stato dimostrato se sia più corretto fare una media su

ogni elemento del campo di valori delle HU e poi ricavare il modulo di Young dell'elemento, oppure, al contrario, trasformare ogni valore HU del voxel in un modulo di Young del relativo voxel e poi fare una media di questo nuovo campo scalare su ogni elemento. Questi due approcci porterebbero a risultati identici se la relazione densità-elasticità adottata fosse lineare. Poiché in letteratura le relazioni più utilizzate tra densità ossea e modulo di Young sono invece esponenziali, le due strategie porterebbero a una diversa distribuzione delle proprietà del materiale. Una possibile modalità di calibrazione potrebbe essere quella descritta negli studi di *Schileo et al.* [21] tramite l'utilizzo dell'*ESP (European Spine Phantom, standard per il controllo di qualità, precisione e riproducibilità delle misurazioni della BMD nelle ossa della colonna vertebrale)* come riportato da *Kalender* [13].

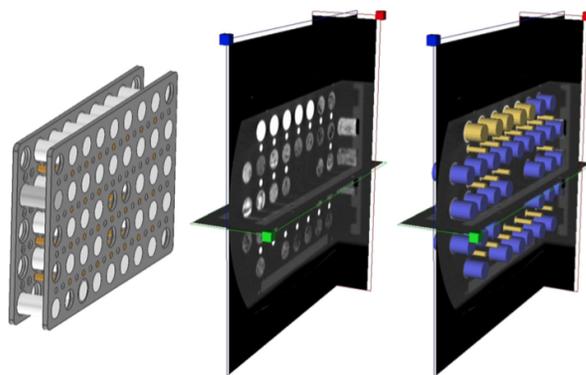


Figura 4.1: CT Scan e posizionamento virtuale dei cilindri per misurare l'intensità dei voxel

In questo caso, l'equazione di calibrazione che lega le Haunsfield units alla densità è stata ricavata come: $\rho_{QCT} = 0.007764HU - 0.056148$, dove ρ_{QCT} è espressa in $\frac{g}{cm^3}$. Per calcolare il valore medio del numero CT in ogni ele-

mento, l'intensità dei voxel contenuti nell'immagine di calibrazione si misura mediante l'inserimento di cilindri virtuali contenenti i voxel d'interesse.

Un altro limite che caratterizza quest'analisi è che 16 pazienti sono un numero troppo basso per un'analisi alle componenti principali. Nonostante non esista formalmente un *minimo* di misurazioni per effettuare una PCA, sicuramente si può affermare che il numero di pazienti studiati non determina efficacemente tutta la variabilità anatomica che esiste tra gli esseri umani. L'analisi alle componenti principali qui condotta è generalizzabile e valida in generale sicuramente per il dataset di pazienti utilizzato, ma non è detto che lo sia anche per pazienti esterni ad esso; lo studio condotto sul *Leave-One-Out* è stato inteso più come un approccio metodologico che conclusivo. A supporto di queste affermazioni si può osservare come, nonostante la figura 3.10 mostri che gli errori medi ottenuti sia con la validazione che con il full dataset siano molto bassi e simili tra loro, gli errori massimi siano troppo elevati - come evidenziato in figura 3.11. In quest'ultimo grafico è stato riportato anche il valore più elevato del modulo di Young di ciascun paziente per una maggiore comprensione e una migliore comparazione dei risultati: è vero che gli errori massimi ottenuti con i due metodi sono simili, ma è anche evidente che siano di ordini di grandezza troppo alti. Questo risultato è riconducibile al numero eccessivamente basso di pazienti nel dataset iniziale.

Il terzo limite è legato al processo di *mesh morphing*. La tabella 3.1 è stata ottenuta utilizzando le informazioni sulla connettività delle mesh superficiali in formato STL ricavate dall'ing. dott. Marco Sensale e le coordinate e gli spostamenti riportati sono riferiti a quei nodi ed elementi. Tuttavia, quando la mesh è stata importata in Ansys ne è stato eseguito un *reme-*

shing generale, quindi i *control points* del file per il *Morph From File* non coincidevano più con i nodi della mesh appena effettuata. Il metodo delle radial basis functions, come riportato in § 2.3.1, si basa su delle interpolazioni e le mesh superficiali originali, come spiegato in § 3.1, sono state morfate con un algoritmo diverso da quello delle radial basis functions. Questo fatto crea un errore di morphing dovuto proprio all'interpolazione. Le figure 4.2 e 4.3 rappresentano gli errori medi e massimi dovuti all'interpolazione nei due algoritmi di morphing utilizzati. Tali errori corrispondono alla *distanza di Hausdorff*² tra le mesh ottenute con i diversi algoritmi.

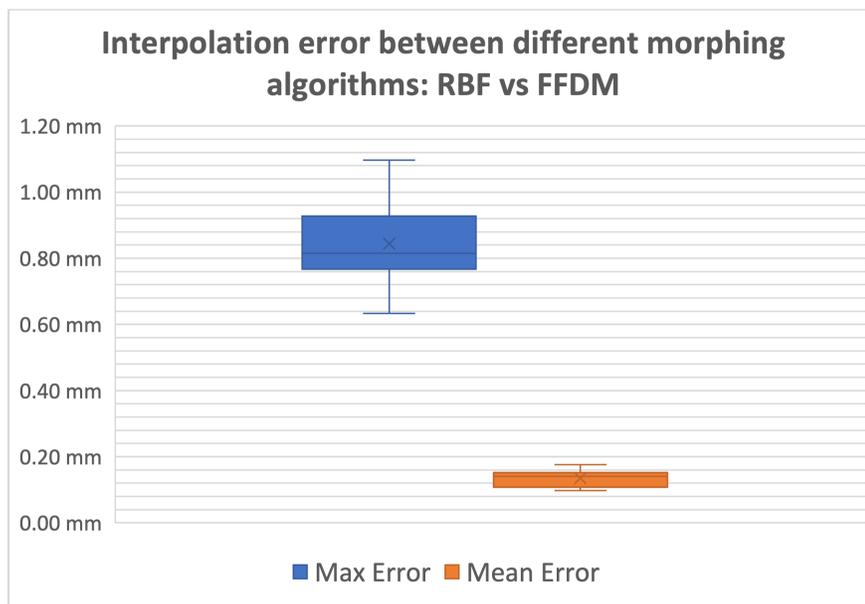


Figura 4.2: Errore di interpolazione tra gli algoritmi RBF e FFDM

L'ultimo limite dell'analisi riguarda l'isotopologia delle mesh per la PCA. Affinché l'analisi alle componenti principali sulle proprietà meccaniche sia valida, occorre controllare che ogni elemento corrisponda alla stessa parte

²La distanza di Hausdorff media e massima è stata calcolata utilizzando il software *MeshLab v2022.02*.

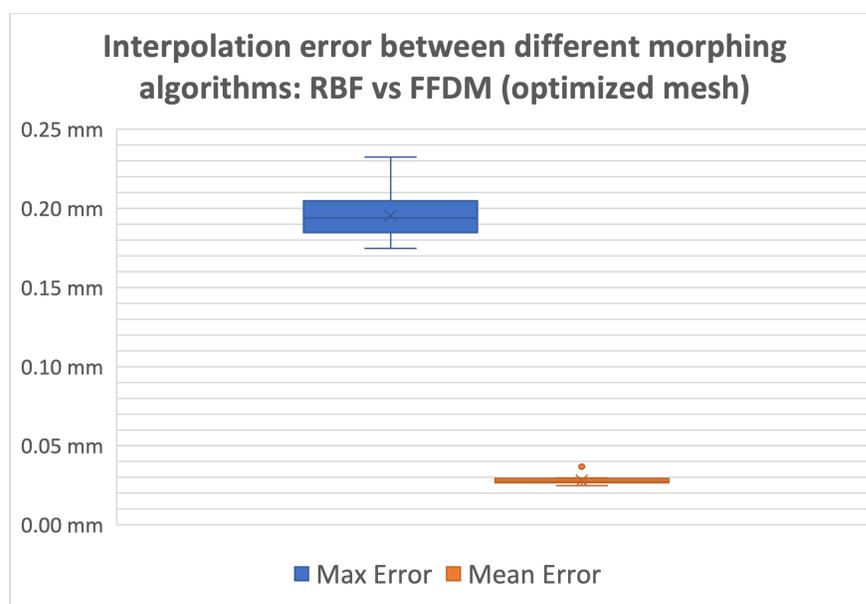


Figura 4.3: Errore di interpolazione tra gli algoritmi RBF e FFDM rispetto alla mesh ottimizzata

anatomica di ogni paziente. L'elemento x di un certo paziente può essere in una posizione relativa diversa rispetto allo stesso elemento x di un altro paziente. Manca dunque il controllo del rispetto all'esterno e soprattutto all'interno del mantenimento della posizione relativa degli elementi. Se ciò non fosse verificato, potrebbe essere fonte di errore e rumore all'interno della PCA, riducendo la validità dei risultati.

4.2 Conclusione e commenti finali

In questo studio si è partiti da uno *statistical shape model* dai corpi vertebrali della prima vertebra lombare di 16 pazienti e si è creato uno *statistical appearance model* con un'analisi alle componenti principali. Per rendere le mesh isotopologiche si è utilizzato l'algoritmo delle radial basis functions e per la

mappatura delle caratteristiche meccaniche si sono utilizzate le relazioni 3.1 e 3.2. Per la validazione del modello statistico, si è utilizzato il metodo del *Leave-One-Out* che ha evidenziato errori medi molto bassi, ma errori massimi troppo elevati. Tali risultati sono dovuti ai limiti dell'approccio utilizzato descritti in § 4.1. Questo studio può essere migliorato verificando in prima istanza che le macchine che hanno prodotto le immagini cliniche siano state sottoposte a una calibrazione densitometrica, in modo da ottenere una relazione tra densità ossea e i livelli di grigio dell'immagine più precisa e accurata. Inoltre è importante ridefinire la variabilità del dataset di training del modello statistico aumentando sicuramente il numero di pazienti studiati e soprattutto non introducendo dati troppo variabili, in quanto la PCA troverebbe l'eccessiva variabilità restituendo troppo rumore nei risultati. Occorre dunque ridurre la variabilità a priori (ad esempio segmentando i pazienti studiando solo quelli con determinate caratteristiche fisiche, anatomiche, di sesso, di età ecc.). L'ultimo passo per migliorare questo studio è quello di trovare un metodo per assicurarsi del mantenimento della posizione relativa degli elementi corrispondenti tra i vari pazienti. Seguendo tutte queste proposte e risolvendo questi problemi, la PCA risulterebbe ottimizzata e restituirebbe risultati più accurati e fedeli.

Script Matlab

Matrice Delta per RBF Morph From File

```
% Creazione matrice Delta per RBF Morph From File

clc;

clear all;

close all;

% Importo i file stl

meanMesh = stlread('MeanMeshIteration4.stl');

ptMesh = stlread('Pt1L1_MeshMorphed.stl');

% Assegno le matrici

meanMesh = meanMesh.Points;

ptMesh = ptMesh.Points;

size = size(meanMesh);

size = size(1,1);

Delta = zeros (size, 6);
```

```
for i = 1:size

    for j = 1:3

        % Calcolo il delta
        delta = (ptMesh(i,j) - meanMesh(i,j))/1000;

        % Assegno il delta alla i-esima riga e (j+3)-esima
        % colonna (quarta, quinta, sesta)
        Delta(i, j+3) = delta;

        % Assegno X,Y,Z dell'elemento i-esimo alla j-esima
        % riga (prima, seconda, terza)
        Delta(i,j) = meanMesh(i,j)/1000;

    end

end

end

% Traspongo la matrice D per stamparla correttamente
% con fprintf
Delta = Delta';

% Stampo il file corretto da utilizzare per
% RBF Morph From File
rbf = fopen('MorphFromFile_Pt1L1.txt', 'w');
```

```
% Formattazione: colonne con tablatura tra di loro e valori
% con 6 cifre decimali
fprintf(rbf, '%.6f\t %.6f\t %.6f\t %.6f\t %.6f\t %.6f\n', Delta);

fclose(rbf);
```

Rototraslazione mesh volumetrica

```
% Rototraslazione mesh volumetrica .cdb
% Lo scopo è di trovare una matrice di rotazione e un vettore traslazione
% da applicare alla mesh volumetrica affinché vada ad allinearsi con la
% mesh allineata alle CT Scan, per poi ricavare
% le proprietà meccaniche dell'osso in Bonemat

clc;
clear all;
close all;

% Importo i file stl sorgente (partenza) e target (arrivo)
% sorgente
fv1 = stlread('Pt1L1_segmented_aligned.stl');
% target
fv2 = stlread('Pt1L1_segmented.stl');

% creo la matrice A con le coordinate degli N nodi della
% mesh sorgente per la funzione absor(A,B)
A = fv1.Points;
% traspongo A affinché sia del tipo 3xN
A = A';

% stesso procedimento per B
B = fv2.Points;
B = B';
```

```
% Calcolo la matrice di rotazione e il vettore di traslazione
% con la funzione absor
[regParams,Bfit,ErrorStats]=absor(A,B);

%% Sezione di verifica corretto funzionamento
% Applico R e t alla matrice di partenza per verificare
% che si sovrapponga, quindi traspongo nuovamente A
% per farla tornare a Nx3
A = A';

% creo la matrice C con un numero di righe pari al
% numero di nodi e 3 colonne per le coordinate X Y Z
s = size(A);
C = zeros(s(:,1), 3);

% Estrapolo la matrice di rotazione
R = regParams.R;

% Estrapolo il vettore traslazione
t = regParams.t;

% applico la rototraslazione a tutti i punti della
% matrice di partenza
for i=1:s(:,1)
    v1=A(i,:);
```

```
C(i,:) = R*(v1)' + t;

end

% verifico che ci sia sovrapposizione tra target e
% mesh appena ruotata

% Traspongo di nuovo B
B = B';
K = zeros(s(:,1), 3);
for i = 1:s(:,1)
    for j=1:3
        d = B(i,j)-C(i,j);
        K(i,j) = d;
    end
end

end

% visualizzo a video l'errore medio
differenza = mean(K)

%% Applico la rototraslazione alla mesh volumetrica

% Prima di continuare, bisogna convertire il file .cdb
% in .txt
% Per farlo è sufficiente rinominare il file cambiando
% l'estensione
```

```
% Creazione matrice P con informazioni nodali e rototraslate
% da sostituire

% Leggo il file txt della mesh da ruotare
% per estrapolare i dati
cdb = fopen('Volum_Mesh_Pt1.txt');

% Creo un array saltando le prime 10 righe, che considero header
A = textscan(cdb, '%f %f %f %f %f %f', 'HeaderLines', 10);

% Creo la matrice C con tante righe quanti sono i nodi e
% 6 colonne (IV, V e VI colonna corrispondono a X, Y e Z del nodo)
C = cell2mat(A);

% Determino il numero di righe di C (i.e. il #nodi)
s = size(C);

% Creo variabile s = numero righe (i.e. #nodi)
s = s(1,1);

% Matrice vuota P da scrivere con ciclo for
P = zeros(s, 6);

% Applico la rototraslazione
for i=1:s
```

```
% Creo un vettore con le coordinate X Y Z dell'i-esima riga
v2=[C(i,4) C(i,5) C(i,6)];

% Creo il vettore ruotato
k = R*(v2)' + t;

% Scrivo la matrice P dove vado a mettere nelle
% prime 3 colonne gli stessi valori di C,
% e sulle altre 3 colonne le coordinate X Y Z
% ruotate
P(i,:) = [C(i,1) C(i,2) C(i,3) k(1) k(2) k(3)];

end

%% Trascrizione nuovo file .txt con le coordinate rototraslate
% Il file .txt della mesh è diviso in 4 sezioni:
% testo - numeri - testo - numeri

% Scrivo la prima sezione di testo creando un nuovo file testo

% Apro il file txt della mesh in lettura e copio le prime 10 righe
fid = fopen('Volum_Mesh_Pt1.txt','r');
righe = 10;
intro = cell(righe,1);
for i = 1:righe
    testo(i) = {fgetl(fid)};
```

```
end
fclose(fid);

% Apro un file vuoto in scrittura e scrivo le righe che ho copiato
% in precedenza
fid = fopen('Volum_Mesh_Pt1_rot.txt','w');
for i = 1:righe
    fprintf(fid,'%s\n',testo{i});
end
fclose(fid);

%% Scrivo seconda parte (coordinate)

% Apro il nuovo file appena creato in modalità "append"
% per scrivere sotto alle righe che ho appena copiato
fid2 = fopen('Volum_Mesh_Pt1_rot.txt', 'a');

% Trascrivo la matrice P
for i = 1:s
    fprintf(fid2,'%u %u %u %.13E %.13E %.13E\n', P(i,:));
end
fclose(fid2);

%% Scrivo terza parte testo
```

```
% Adesso apro il file originale della mesh volumetrica
% per creare un array
fid3 = fopen('Volum_Mesh_Pt1.txt','r');
S = textscan(fid3,'%s','delimiter','\n');
S = S{1};
fclose(fid3);

% Apro il file txt da scrivere in modalità append
% e scrivo la terza parte
% di testo
fid4 = fopen('Volum_Mesh_Pt1_rot.txt','a');
w = size(S);
w = w(1);
g = 11+s;
for i = g:g+2
    B = string(S(i));
    fprintf(fid4, '%s\n', B);
end
fclose(fid4);

%% Scrivo ultima parte

fid5 = fopen('Volum_Mesh_Pt1.txt','r') ;
S = textscan(fid3,'%s', 'delimiter', '\n') ;
S = S{1} ;
fclose(fid5) ;
```

```
fid6 = fopen('Volum_Mesh_Pt1_rot.txt','a');

% In questo caso devo prima convertire l'array
% in una stringa perché devo trascrivere del testo
for i = (g+3):(w)
    C=char(S(i));
    fprintf(fid6, '%s\n', C);
end

%% Ultimo passaggio: convertire nuovamente il nuovo file
%% .txt in un .cdb e caricarlo in Bonemat per proseguire
```

Matrice dei moduli di Young

```
%% Scrittura file proprietà elementi mesh
% A partire dalla mesh esportata da Bonemat, scrivere una matrice di due
% colonne: #elemento | Modulo Young

clc;
clear all;
close all;

% Leggo il file txt della mesh per estrapolare i dati
fid1 = fopen('Volum_Mesh_Pt15_rot_Bonemat.txt','r');
S = textscan(fid1,'%s','delimiter','\n');
S = S{1};
fclose(fid1);

% Ignoro le prime 115304 righe, che sono quelle
% con le informazioni sulla geometria della mesh
% volumetrica
h = 115305;
w = size(S);
w = w(1)-3;      % Le ultime tre righe sono uno spazio
                 % e due stringhe

%Creo un file testo ausiliario da cui
% estrarrò i dati successivamente
fid2 = fopen('CDBL1_AUX.txt','w');
```

```

for i = h:w
    if floor(i/2)*2 == i && i < w
        C=char(S(i));
        fprintf(fid2, '%s\n', C);
    end
end

fclose(fid2);

%% Creazione matrice E dei moduli di Young

% Leggo il file txt ausiliario per estrapolare i dati
mesh = fopen('CDBL1_AUX.txt');

% Creo un array dalla lettura del file testo ausiliario
A = textscan(mesh, '%s %s %d %s %d %d %f', 'delimiter', ',');
m = size (A{1,1});
C = zeros (m(1),2);
C(:,1) = A{1,5};
C(:,2) = A{1,7};
s = size(C);
s = s(1);
t = ceil(s/3);
E = zeros(t, 2);
B = A{1,4};

% Creo la matrice E leggendo solo i valori che

```

```
% corrispondono al modulo di Young (ricercando la stringa "EX")
for i = 1:s

    if B(i) == "EX"
        E(i,1)=C(i,1);
        E(i,2)=C(i,2);
    end
end

% Elimino le righe vuote dalla matrice E
E = E(~all(E == 0, 2),:);

% Disattiva commento per eliminare righe uguali
% E = unique(E,'rows','stable');

% Elimino il file ausiliario, che è ormai inutile
delete CDBL1_AUX.txt;

% Leggo il file txt della mesh per isolare il blocco EBLOCK

fid3 = fopen('Volum_Mesh_Pt15_rot_Bonemat.txt','r');
S = textscan(fid3,'%s','delimiter','\n');
S = S{1};
fclose(fid3);

% Ignoro le prime 48133 righe, che sono quelle
```

```

% con le informazioni sulla geometria della mesh volumetrica
% e mi concentro fino alla fine del blocco EBLOCK

h = 48134;
w = size(S);
j = size(S)-(t*6 +4);
j = j(1);

%Creo un secondo file testo ausiliario da cui
% estrarrò i dati successivamente
fid2 = fopen('CDBL1_AUX.txt','w');
for i = h:j

    if floor(i/2)*2 == i && i < j
        C=char(S(i));
        fprintf(fid2, '%s\n', C);
    end

end

fclose(fid2);

% Apro il nuovo txt ausiliario dove è presente solo il blocco EBLOCK
fid4 = fopen('CDBL1_AUX.txt','r');
Q = textscan(fid4,'%d %d %d\n');

% vettore con numero materiale per elemento
H = Q{1,1};

```

```
h = size(H);
h = h(1);

% vettore con numero elemento
F = Q{1,11};
f = size(F);
f = f(1);
fclose(fid4);

R = zeros(h,2);

for i = 1:h
    R(i,1)=F(i);
    R(i,2) = E(H(i),2);
end

% Ordino la matrice in ordine crescente rispetto l'ID Element
R = sortrows(R);

% Salvo il file .txt con colonne ID Element | Young modulus
writematrix(R, 'ID_E_Pt15.txt', 'delimiter', 'tab');
delete CDBL1_AUX.txt;

MAX = max(E(:,2));
min = min(E(:,2));
```

Analisi PCA

```
% Importo le tabelle dei pazienti

Pt1 = readmatrix("Patient1/ID_E_Pt1.txt");
Pt2 = readmatrix("Patient2/ID_E_Pt2.txt");
Pt3 = readmatrix("Patient3/ID_E_Pt3.txt");
Pt4 = readmatrix("Patient4/ID_E_Pt4.txt");
Pt5 = readmatrix("Patient5/ID_E_Pt5.txt");
Pt6 = readmatrix("Patient6/ID_E_Pt6.txt");
Pt7 = readmatrix("Patient7/ID_E_Pt7.txt");
Pt8 = readmatrix("Patient8/ID_E_Pt8.txt");
Pt9 = readmatrix("Patient12/ID_E_Pt12.txt");
Pt10 = readmatrix("Patient13/ID_E_Pt13.txt");
Pt11 = readmatrix("Patient14/ID_E_Pt14.txt");
Pt12 = readmatrix("Patient15/ID_E_Pt15.txt");
Pt13 = readmatrix("Paziente16/ID_E_Pt16.txt");
Pt14 = readmatrix("Paziente18/ID_E_Pt18.txt");
Pt15 = readmatrix("Paziente19/ID_E_Pt19.txt");
Pt16 = readmatrix("Paziente20/ID_E_Pt20.txt");

% Creo la matrice dei pazienti PtMatrix del tipo nxp,
% con n = osservazioni e p = variabili
% (elementi x pazienti)

s = size(Pt1);

s = s(1);

PtMatrix = zeros(s,16);

PtMatrix(:,1)=Pt1(:,2);
```

```
PtMatrix(:,2)=Pt2(:,2);
PtMatrix(:,3)=Pt3(:,2);
PtMatrix(:,4)=Pt4(:,2);
PtMatrix(:,5)=Pt5(:,2);
PtMatrix(:,6)=Pt6(:,2);
PtMatrix(:,7)=Pt7(:,2);
PtMatrix(:,8)=Pt8(:,2);
PtMatrix(:,9)=Pt9(:,2);
PtMatrix(:,10)=Pt10(:,2);
PtMatrix(:,11)=Pt11(:,2);
PtMatrix(:,12)=Pt12(:,2);
PtMatrix(:,13)=Pt13(:,2);
PtMatrix(:,14)=Pt14(:,2);
PtMatrix(:,15)=Pt15(:,2);
PtMatrix(:,16)=Pt16(:,2);

% Effettuo la PCA
% PC = principal component
% var = varianza relativa alla componente principale
% perc = percentuale di varianza sul totale
% (relativa alla componente
% principale)

[coeff,PC,var,~,perc] = pca(PtMatrix);
k = size(PC);
elements = k(1);
```

```
patients = k(2);

% creo un vettore con la percentuale di varianza cumulata
cumulate = cumsum(perc);

% cerco il numero di PC che mi danno più del 90% di variabilità
c=1;
while cumulate(c) <= 90
    c = c+1;
end

% Scrivo la matrice con il massimo E di ciascun paziente
MAXVALUE = zeros(patients,1);
for i = 1:patients
    MAXVALUE(i,:) = max(PtMatrix(:,i));
end

% Scrivo la matrice con il medio E di ciascun paziente
MEANVALUE = zeros(patients,1);
for i = 1:patients
    MEANVALUE(i,:) = mean(PtMatrix(:,i));
end

%% Sezione plot

% Plot della varianza delle componenti principali
```

```
s = size(PtMatrix);
patients = s(2);
x = linspace(1,patients,patients);

% Plot della cumulata con uno spessore di linea pari a 3
plot(x,cumulate,'-p', 'LineWidth',3)
hold on

% Plot della varianza di ciascuna PC con uno spessore di linea pari a 3
plot(x,perc,'-p', 'LineWidth',3)
hold off

ax = gca;
ax.FontSize = 30;
% Imposto l'etichetta sull'asse X con grandezza di font pari a 30
xlabel('Principal Component', 'fontsize', 50);
% Imposto l'etichetta sull'asse Y con grandezza di font pari a 30
ylabel('Variance', 'fontsize', 50);
grid on
% Imposto la legenda posizionandola nella posizione migliore (best)
% con grandezza di font pari a 20
legend('Cumulate','Variance','Orientation','horizontal', 'fontsize', 30);
xticks(0:patients);
```

Leave-One-Out Cross-Validation

```
clc;
clear all;
close all;

% Importo le tabelle dei pazienti
Pt1 = readmatrix("Patient1/ID_E_Pt1.txt");
Pt2 = readmatrix("Patient2/ID_E_Pt2.txt");
Pt3 = readmatrix("Patient3/ID_E_Pt3.txt");
Pt4 = readmatrix("Patient4/ID_E_Pt4.txt");
Pt5 = readmatrix("Patient5/ID_E_Pt5.txt");
Pt6 = readmatrix("Patient6/ID_E_Pt6.txt");
Pt7 = readmatrix("Patient7/ID_E_Pt7.txt");
Pt8 = readmatrix("Patient8/ID_E_Pt8.txt");
Pt9 = readmatrix("Patient12/ID_E_Pt12.txt");
Pt10 = readmatrix("Patient13/ID_E_Pt13.txt");
Pt11 = readmatrix("Patient14/ID_E_Pt14.txt");
Pt12 = readmatrix("Patient15/ID_E_Pt15.txt");
Pt13 = readmatrix("Paziente16/ID_E_Pt16.txt");
Pt14 = readmatrix("Paziente18/ID_E_Pt18.txt");
Pt15 = readmatrix("Paziente19/ID_E_Pt19.txt");
Pt16 = readmatrix("Paziente20/ID_E_Pt20.txt");

% Creo la matrice dei pazienti PtMatrix del tipo nxp,
% con n = osservazioni e p = variabili (elementi x pazienti)
s = size(Pt1);
```

```
s = s(1);  
PtMatrix = zeros(s,16);  
PtMatrix(:,1)=Pt1(:,2);  
PtMatrix(:,2)=Pt2(:,2);  
PtMatrix(:,3)=Pt3(:,2);  
PtMatrix(:,4)=Pt4(:,2);  
PtMatrix(:,5)=Pt5(:,2);  
PtMatrix(:,6)=Pt6(:,2);  
PtMatrix(:,7)=Pt7(:,2);  
PtMatrix(:,8)=Pt8(:,2);  
PtMatrix(:,9)=Pt9(:,2);  
PtMatrix(:,10)=Pt10(:,2);  
PtMatrix(:,11)=Pt11(:,2);  
PtMatrix(:,12)=Pt12(:,2);  
PtMatrix(:,13)=Pt13(:,2);  
PtMatrix(:,14)=Pt14(:,2);  
PtMatrix(:,15)=Pt15(:,2);  
PtMatrix(:,16)=Pt16(:,2);  
  
% Effettuo la PCA  
% PC = principal component  
% var = varianza relativa alla componente principale  
% perc = percentuale di varianza sul totale (relativa alla componente  
% principale)  
  
[coeff,PC,var,~,perc] = pca(PtMatrix);
```

```
k = size(PC);
elements = k(1);
patients = k(2);

% creo un vettore con la percentuale di varianza cumulata
cumulate = cumsum(perc);

% cerco il numero di PC che mi danno più del 90% di variabilità
c=1;
while cumulate(c) <= 90
    c = c+1;
end

% Creo il vettore medio, dove in ogni colonna inserisco la media
% di ciascun paziente
X_medio = zeros(1, patients);

for i = 1 : patients
    X_medio(:,i) = mean(PtMatrix(:,i));
end

%% Sezione di verifica delle combinazioni lineari (disattivare i commenti)

% Test per controllare che se faccio la combinazione lineare tra tutte le
% componenti principali e ogni paziente mi ritorna proprio la matrice
% PtMatrix
```

```
Test = zeros(elements,patients);

for t = 1:patients

    % Calcolo i coefficienti della combinazione lineare
    x_testcoeff = PC\PtMatrix(:,t);

    % Scrivo la matrice Test dove ogni colonna corrisponde al relativo
    % paziente scritto come combinazione lineare di tutte le componenti
    % principali
    Test(:,t) = PC*x_testcoeff;
end

% Calcolo ora la differenza (errore)
testDiff = PtMatrix - (Test+X_medio);

% Errore medio
mean_testDiff = mean(testDiff);

% Media dell'errore medio
mean_mean = mean(mean_testDiff);

% Errore massimo
max_testDiff = max(testDiff);

% Media dell'errore massimo
mean_mean = mean(max_testDiff);
```

```
%% Combinazione lineare delle prime c componenti principali

% Scrivo ora la matrice PCompMatrix (Principal Component Matrix) le cui
% colonne sono i primi c componenti principali

PCompMatrix = zeros(elements,c);

for i = 1:c
    PCompMatrix(:,i)=PC(:,i);
end

% Creo ora una matrice LCFullData (Linear Combination of Full Dataset)
% che ha le stesse dimensioni di quella che contiene
% le info sui pazienti (i.e. PtMatrix)
LCFullData = zeros(elements,patients);

% Scrivo in ciascuna colonna di LCFullData il relativo paziente
% come combinazione lineare delle prime c componenti principali
for t = 1:patients

    % Trovo i coefficienti della combinazione lineare PCompMatrix*q = PtMatrix(:,t)
    q = PCompMatrix\PtMatrix(:,t);

    % Scrivo la matrice LCFullData con i risultati della combinazione
    % lineare PCompMatrix*q
    LCFullData(:,t) = PCompMatrix*q;
```

```
end
```

```
% Matrice E1 degli errori
```

```
E1 = PtMatrix - (LCFullData + X_medio);
```

```
% Errore medio full dataset
```

```
mean_diff_fd = mean(E1);
```

```
% Errore massimo full dataset
```

```
max_diff_fd = max(E1);
```

```
%% Leave One Out Cross-Validation
```

```
% Creo la matrice PtMat_L00 (Patient Matrix_Leave One Out) da cui vado a
```

```
% togliere ogni volta un paziente durante il Leave One Out
```

```
PtMat_L00 = zeros (elements, patients-1);
```

```
% Creo la matrice nella quale andrò a mettere le prime c componenti
```

```
% principali della nuova PCA (Principal Components_Leave One Out)
```

```
PComp_L00 = zeros (elements, c);
```

```
% Creo la matrice nella quale ogni colonna corrisponderà a ciascun paziente
```

```
% scritto come combinazione lineare delle c componenti principali trovate
```

```
% nel Leave One Out (Patient Linear Combination_Leave One Out)
```

```
PtLinComb_L00 = zeros(elements, patients);
```

```
% h indica il paziente che rimuovo
```

```
for h = 1 : patients
```

```
% Scrivo il vettore corrispondente al paziente h-esimo che salto
jumpedPt = PtMatrix(:,h);

l = 1;
    for j = 1:patients
        if l < (patients-1)
            if j ~= h
                PtMat_L00(:,l)=PtMatrix(:,j);
                l = l+1;
            end
        end
    end

end

% Fin qui ho scritto la matrice PtMat_L00 composta da 15 pazienti,
% avendo tolto l'h-esimo paziente
% Di questa matrice devo fare la PCA e prendere le prime c
% componenti principali
[~,princcomp] = pca(PtMat_L00);

% Trovo le prime c componenti principali e le inserisco nella
% matrice PComp_L00
for y = 1:c
    PComp_L00(:,y)=princcomp(:,y);
end
```

```
% Adesso devo scrivere l'h-imo paziente che ho saltato come
% combinazione lineare delle prime c componenti principali appena
% inserite nella matrice PComp_L00

% Trovo i coefficienti L00coeff (Leave One Out Coeff) della combinazione lineare
L00_coeff = PComp_L00\jumpedPt;

% Riscrivo il paziente saltato come combinazione lineare
jumpedPt = PComp_L00*L00_coeff;

% Inserisco il paziente h-imo saltato nella colonna h-ima di
% PtLinComb_L00
PtLinComb_L00(:,h)=jumpedPt;

end

% Matrice E2 degli errori
E2 = PtMatrix - (PtLinComb_L00+X_medio);

% Scrivo l'errore medio (leave one out)
mean_diff_loo = mean(E2);

% Scrivo l'errore massimo (leave one out)
max_diff_loo = max(E2);

% Creo la matrice meanE degli errori medi calcolati con le c componenti
```

```

% principali in Leave One Out e full dataset
meanE = zeros(16,1);
meanE(:,1)= mean_diff_loo;
meanE(:,2)= mean_diff_fd;

% Scrivo la matrice con il massimo E di ciascun paziente
MAXVALUE = zeros(patients,1);
for i = 1:patients
    MAXVALUE(i,:) = max(PtMatrix(:,i));
end
% Creo la matrice meanE degli errori medi calcolati con le c componenti
% principali in Leave One Out e full dataset + il valore massimo di E di
% ciascun paziente
maxE = zeros(16,3);
maxE(:,1)= max_diff_loo;
maxE(:,2)= max_diff_fd;
maxE(:,3)= MAXVALUE;

% Creo una stringa 1x2 dove inserisco la legenda dei grafici in funzione
% del numero di componenti principali c
str = {strcat(num2str(c), ' Principal Components & 15 Patients (LOO Validation)')};
str = [str , strcat(num2str(c), ' Principal Components & Full Dataset')];

% Stampo il grafico a barre dell'errore medio e massimo con tutte le
% componenti principali
% Errore medio

```

```
figure()
bar(mean_testDiff)
ax = gca;
ax.FontSize = 30;
% Imposto l'etichetta sull'asse X con grandezza di font pari a 30
xlabel('Patient', 'fontsize', 50);
% Imposto l'etichetta sull'asse Y con grandezza di font pari a 30
ylabel('Mean Error (MPa)', 'fontsize', 50);
grid on
xticks(0:s);

% Errore massimo
figure()
bar(max_testDiff)
ax = gca;
ax.FontSize = 30;
% Imposto l'etichetta sull'asse X con grandezza di font pari a 30
xlabel('Patient', 'fontsize', 50);
% Imposto l'etichetta sull'asse Y con grandezza di font pari a 30
ylabel('Max Error (MPa)', 'fontsize', 50);
grid on
xticks(0:s);

% Stampo il grafico a barre dell'errore medio con c componenti principali
figure()
bar(meanE)
```

```

ax = gca;
ax.FontSize = 30;
% Imposto l'etichetta sull'asse X con grandezza di font pari a 30
xlabel('Patient', 'fontsize', 50);
% Imposto l'etichetta sull'asse Y con grandezza di font pari a 30
ylabel('Mean Error (MPa)', 'fontsize', 50);
grid on
% Imposto la legenda posizionandola nella posizione
% migliore (best) con grandezza di font pari a 20
legend(str{:}, 'Location', 'best', 'Orientation', 'horizontal', 'fontsize', 20);
xticks(0:s);

% Stampo il grafico a barre dell'errore massimo con c componenti principali
figure()
bar(maxE)
ax = gca;
ax.FontSize = 30;
% Imposto l'etichetta sull'asse X con grandezza di font pari a 30
xlabel('Patient', 'fontsize', 50);
% Imposto l'etichetta sull'asse Y con grandezza di font pari a 30
ylabel('Max Error (MPa)', 'fontsize', 50);
grid on
% Imposto la legenda posizionandola nella posizione
% migliore (best) con grandezza di font pari a 20
legend(str{:}, 'Max Value', 'Location', 'best', 'Orientation', 'horizontal', 'fontsize', 20);
xticks(0:s);

```


Bibliografia

- [1] Adobe. *File STL*. URL: <https://www.adobe.com/it/creativecloud/file-types/image/vector/stl-file.html>.
- [2] Anjany Sekuboyina et al. “VerSe: A Vertebrae labelling and segmentation benchmark for multi-detector CT images”. In: *Medical Image Analysis* 73 (2021), p. 102166. ISSN: 1361-8415. DOI: <https://doi.org/10.1016/j.media.2021.102166>. URL: <https://www.sciencedirect.com/science/article/pii/S1361841521002127>.
- [3] Giuseppe Anastasi et al. *Anatomia Umana–Atlante–Volume 1*. EdiErmes, 2014.
- [4] Marco Evangelos Biancolini e Pier Paolo Valentini. “Virtual human bone modelling by interactive sculpting, mesh morphing and force-feedback”. In: *International Journal on Interactive Design and Manufacturing (IJIDeM)* 12.4 (2018), pp. 1223–1234.
- [5] Steven L Brunton e J Nathan Kutz. *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2022.

- [6] Martin D Buhmann. *Radial basis functions: theory and implementations*. Vol. 12. Cambridge university press, 2003.
- [7] G.A. Day, A.C. Jones e R.K. Wilcox. “Using Statistical Shape and Appearance Modelling to characterise the 3D shape and material properties of human lumbar vertebrae: A proof of concept study”. In: *Journal of the Mechanical Behavior of Biomedical Materials* 126 (2022), p. 105047. ISSN: 1751-6161. DOI: <https://doi.org/10.1016/j.jmbbm.2021.105047>. URL: <https://www.sciencedirect.com/science/article/pii/S175161612100672X>.
- [8] John Enderle. *Introduction to biomedical engineering*. Academic press, 2012.
- [9] Jennifer Fayad et al. “Complicated Postoperative Flat Back Deformity Correction With the Aid of Virtual and 3D Printed Anatomical Models: Case Report”. In: *Frontiers in Surgery* 8 (2021). ISSN: 2296-875X. DOI: [10.3389/fsurg.2021.662919](https://doi.org/10.3389/fsurg.2021.662919). URL: <https://www.frontiersin.org/article/10.3389/fsurg.2021.662919>.
- [10] Lorenzo Grassi, Sami P Väänänen e Hanna Isaksson. “Statistical Shape and Appearance Models: Development Towards Improved Osteoporosis Care”. In: *Current osteoporosis reports* (2021), pp. 1–12.
- [11] Tobias Heimann e Hans-Peter Meinzer. “Statistical shape models for 3D medical image segmentation: A review”. In: *Medical Image Analysis* 13.4 (2009), pp. 543–563. ISSN: 1361-8415. DOI: <https://doi.org/10.1016/j.media.2009.05.004>. URL: <https://www.sciencedirect.com/science/article/pii/S1361841509000425>.

- [12] *Introduzione alla Analisi dei dati sperimentali*. URL: https://didattica-2000.archived.uniroma2.it/Sensori_ed_applic_2/deposito/dispensa_statistica.pdf.
- [13] Willi A Kalender. “A phantom for standarization and quality control in spinal bone mineral measurements by QCT and DXA: Design considerations and specifications”. In: *Medical physics* 19.3 (1992), pp. 583–586.
- [14] David C Lee et al. “Phantomless calibration of CT scans for measurement of BMD and bone strength—inter-operator reanalysis precision”. In: *Bone* 103 (2017), pp. 325–333.
- [15] C. Lorenz e N. Krahnstover. “3D statistical shape models for medical image segmentation”. In: *Second International Conference on 3-D Digital Imaging and Modeling (Cat. No.PR00062)*. 1999, pp. 414–423. DOI: 10.1109/IM.1999.805372.
- [16] Elise F Morgan, Harun H Bayraktar e Tony M Keaveny. “Trabecular bone modulus–density relationships depend on anatomic site”. In: *Journal of biomechanics* 36.7 (2003), pp. 897–904.
- [17] Elise F Morgan, Harun H Bayraktar e Tony M Keaveny. “Trabecular bone modulus–density relationships depend on anatomic site”. In: *Journal of biomechanics* 36.7 (2003), pp. 897–904.
- [18] Giulia Pascoletti. “Statistical shape modelling of the human mandible: 3D shape predictions based on external morphometric features”. In: *International Journal on Interactive Design and Manufacturing (IJI-*

- DeM*) (mag. 2022), pp. 1–19. DOI: 10.1007/s12008-022-00882-5.
- [19] “Leave-One-Out Cross-Validation”. In: *Encyclopedia of Machine Learning*. A cura di Claude Sammut e Geoffrey I. Webb. Boston, MA: Springer US, 2010, pp. 600–601. ISBN: 978-0-387-30164-8. DOI: 10.1007/978-0-387-30164-8_469. URL: https://doi.org/10.1007/978-0-387-30164-8_469.
- [20] Nazli Sarkalkan, Harrie Weinans e Amir A Zadpoor. “Statistical shape and appearance models of bones”. In: *Bone* 60 (2014), pp. 129–140.
- [21] Enrico Schileo et al. “An accurate estimation of bone density improves the accuracy of subject-specific finite element models”. In: *Journal of biomechanics* 41.11 (2008), pp. 2483–2491.
- [22] Joseph J Schreiber, Paul A Anderson e Wellington K Hsu. “Use of computed tomography for assessing bone mineral density”. In: *Neurosurgical focus* 37.1 (2014), E4.
- [23] Marco Sensale et al. “Patient-Specific Finite Element Models of Posterior Pedicle Screw Fixation: Effect of Screw’s Size and Geometry”. In: *Frontiers in Bioengineering and Biotechnology* 9 (2021). ISSN: 2296-4185. DOI: 10.3389/fbioe.2021.643154. URL: <https://www.frontiersin.org/article/10.3389/fbioe.2021.643154>.
- [24] Joseph Edward Shigley. *Shigley’s mechanical engineering design*. Tata McGraw-Hill Education, 2011.

- [25] Fulvia Taddei et al. “The material mapping strategy influences the accuracy of CT-based finite element models of bones: an evaluation against experimental measurements”. In: *Medical engineering & physics* 29.9 (2007), pp. 973–979.
- [26] Wikipedia. *STL (formato di file)* — *Wikipedia, L’enciclopedia libera*. 2021. URL: [http://it.wikipedia.org/w/index.php?title=STL_\(formato_di_file\)&oldid=122538518](http://it.wikipedia.org/w/index.php?title=STL_(formato_di_file)&oldid=122538518).
- [27] Wikipedia. *Varianza* — *Wikipedia, L’enciclopedia libera*. 2021. URL: <http://it.wikipedia.org/w/index.php?title=Varianza&oldid=121816290>.
- [28] Wikipedia. *Vertebra* — *Wikipedia, L’enciclopedia libera*. 2021. URL: <http://it.wikipedia.org/w/index.php?title=Vertebra&oldid=122538663>.
- [29] Dieter Christian Wirtz et al. “Critical evaluation of known bone material properties to realize anisotropic FE-simulation of the proximal femur”. In: *Journal of biomechanics* 33.10 (2000), pp. 1325–1330.