



PROGETTAZIONE DI UN SISTEMA BIOMETRICO INNOVATIVO BASATO SU PALMVEIN

Tesina progetto Sensori Rilevatori e Dispositivi
Elettronici

Docente: Antonio Iula

UNIVERSITÀ DEGLI STUDI DELLA BASILICATA

Manuele Capece – Gianfranco Manfreda – Chiara Caporale
67468 - 68856 - 67463

Anno accademico: 2023/2024

SOMMARIO

1 ABSTRACT	3
2 INTRODUZIONE AI SISTEMI BIOMETRICI.....	3
2.1 PROCESSO DI RICONOSCIMENTO BIOMETRICO	4
2.2 METRICHE E PRESTAZIONI	6
2.3 IL PALMVEIN	8
2.4 IL PALMPrint.....	9
2.4.1 <i>Caratteristiche</i>	9
2.4.2 <i>Sistema di acquisizione.</i>	10
3 GENERAZIONE DI ULTRASUONI	15
3.1 ULTRASUONI	15
3.1.1 <i>Emissione di ultrasuoni</i>	16
3.1.2 <i>Propagazione degli ultrasuoni.</i>	17
3.2 SONDA ECOGRAFICA	22
3.2.1 <i>Tipologie di sonde</i>	23
3.2.2 <i>Generazione del fascio.</i>	25
3.2.3 <i>Formazione dell'immagine ecografica</i>	26
4 DESCRIZIONE DEL PROGETTO	29
4.1 SCOPO DEL PROGETTO.....	29
4.2 BIOMETRIC DATA REPRESENTATION (BDR)	30
4.3 OBIETTIVI.....	30
4.4 MOTIVAZIONI.....	30
4.4.1 <i>Palmprint</i>	30
4.4.2 <i>Palmvein</i>	31
4.4.3 <i>Sistemi multimodali</i>	31
4.4.4 <i>Acquisizione del palmpoint e del palmvein</i>	32
4.4.5 <i>Tecnologie ad ultrasuoni</i>	32
4.5 ARCHITETTURA DEL SISTEMA DI ACQUISIZIONE	33
4.5.1 <i>Sonda ecografica</i>	33
4.6 PANTOGRAFO A CONTROLLO NUMERICO	34
4.7 SISTEMA ULA-OP	34
4.8 FILE DI CONFIGURAZIONE (.CFG)	35
4.9 FILE DI STRATEGIA Tx/Rx (.ULA).....	37
4.10 SET-UP SPERIMENTALE	38
4.10.1 <i>Processo di acquisizione del palmprint</i>	39
5 ANALISI DEL DATABASE LUONGO – SCAVONE	40
5.1 GENERAZIONE DELLA MATRICE VOLUMETRICA	40
5.1.1 <i>Acquisizione dei dati</i>	41
5.1.2 <i>Pre-processing dei dati</i>	42
5.1.3 <i>Rappresentazione</i>	42
5.1.4 <i>Visualizzazione pattern</i>	44
6 ESTRAZIONE PATTERN VENOSO	45
6.1 FEATURE EXTRACTION	45
6.1.1 <i>Estrazione dei Bscan</i>	46
6.1.2 <i>Filtraggio e binarizzazione</i>	46
6.1.3 <i>Binarizzazione</i>	47
6.1.4 <i>Ricerca delle componenti connesse</i>	49

6.1.5	<i>Estrazione dei pattern venosi</i>	49
6.2	TEMPLATE RISULTANTI	50
6.2.1	<i>Considerazioni sui template estratti</i>	52
7	IL METODO CCM	53
7.1	INTRODUZIONE	53
7.1.1	<i>Features</i>	53
7.1.2	<i>Assunzioni</i>	53
7.1.3	<i>Premessa</i>	54
7.1.4	<i>Database utilizzato</i>	54
7.1.5	<i>Codice realizzato</i>	54
7.2	CONFRONTO TRA VOLSHOW E BDR	55
7.3	CROPPING.....	62
7.4	ESTRAZIONE	63
7.5	BINARIZZAZIONE.....	66
7.5.1	<i>Calcolo del vettore fine binarizzazione</i>	67
7.5.2	<i>Prima binarizzazione del volume</i>	71
7.5.3	<i>Calcolo vettore distanza minima tra le vene e il palmo</i>	72
7.5.4	<i>Calcolo soglia iniziale</i>	74
7.5.5	<i>Binarizzazione finale</i>	77
7.6	OPERAZIONI MORFOLOGICHE	80
7.7	FILTRAGGIO	85
7.7.1	<i>Addestramento di un modello di classificazione binaria</i>	85
7.7.2	<i>Filtraggio ricorsivo</i>	90
7.8	AFFINAMENTO	93
7.8.1	<i>Affina disturbi</i>	94
7.8.2	<i>Elimina rumore</i>	100
7.8.3	<i>Filling vene</i>	107
7.9	VISUALIZZAZIONE DEL TEMPLATE FINALE.....	109
7.9.1	<i>Utente 1</i>	110
7.9.2	<i>Utente 2</i>	113
7.9.3	<i>Altri utenti</i>	114
8	MATCHING 3D	115
8.1	PROCEDURA	116
8.2	RISULTATI	120
8.2.1	<i>Considerazioni sul tempo di esecuzione</i>	121
8.2.2	<i>Scelta delle acquisizioni</i>	122
8.2.3	<i>Esempi</i>	123
8.2.4	<i>Valutazione dei risultati</i>	123
9	CONCLUSIONI	129
9.1	LIMITI DEL METODO DI ESTRAZIONE	129
9.1.1	<i>Assunzioni</i>	129
9.1.2	<i>Distanza minima vene-palmo</i>	129
9.1.3	<i>Elimina rumore</i>	130
9.1.4	<i>Affina vene</i>	131
9.2	CRITICITÀ NEL MATCHING	131
9.3	POSSIBILI SVILUPPI FUTURI	132
BIBLIOGRAFIA	133	

1 ABSTRACT

Questa tesina affronta il tema dei **sistemi biometrici**, con un focus particolare sulle tecnologie di riconoscimento basate sulle caratteristiche del palmo della mano, come il **palmprint** e il **palmvein**. Dopo una panoramica introduttiva sui principi fondamentali del riconoscimento biometrico e le metriche di valutazione, il lavoro esplora il concetto di **biometric data representation**, ovvero la rappresentazione dei dati biometrici in tre sezioni ortogonali per facilitare il confronto e l'identificazione. Il contributo di questo lavoro comprende inizialmente l'**analisi del database Luongo-Scavone**, un set di dati contenente acquisizioni palmari di diverse persone. A seguito di questa analisi, l'obiettivo iniziale era di verificare la possibilità di estrarre pattern venosi utilizzando il **metodo Vizzuso**, al fine di stabilire se il database fosse adatto a tale scopo. Dopo aver confermato la possibilità di estrarre pattern venosi, si è deciso di sviluppare un sistema personalizzato per il **riconoscimento delle vene del palmo**, migliorando l'approccio esistente. Il nuovo metodo proposto, denominato **metodo CCM**, offre una soluzione più mirata e precisa per l'identificazione biometrica delle vene palmari, rappresentando un'evoluzione significativa rispetto ai metodi precedenti.

2 INTRODUZIONE AI SISTEMI BIOMETRICI

Il termine biometria deriva dalle parole greche *bìos* che significa vita e *metros* che significa conteggio o misura ed è la disciplina che studia la misurazione delle variabili fisiologiche o comportamentali tipiche degli individui, attraverso metodologie matematiche e statistiche. La biometria studia la misurazione dei tratti fisici o comportamentali tipici degli esseri umani. Le misurazioni avvengono su delle grandezze dette caratteristiche biometriche che devono rispettare diversi parametri per essere considerate tali. In particolare, una caratteristica è biometrica se è:

- **Universale**: ogni persona dovrebbe possedere tale caratteristica.
- **Distintiva**: due persone qualsiasi dovrebbero essere sufficientemente differenti per quanto concerne la caratteristica in esame, quindi la caratteristica deve essere unica.
- **Permanente**: la caratteristica dovrebbe rimanere pressoché invariata per un determinato periodo di tempo anche se soggetta a piccole variazioni, non deve mutare (entro limiti accettabili).
- **Accettabile**: la metodologia di acquisizione non deve essere invasiva, l'acquisizione della caratteristica deve essere comoda e tollerabile per l'utente.
- **Catturabile**: deve essere possibile misurare la caratteristica attraverso dispositivi e/o sensori biometrici.
- **Collezionabile**: la caratteristica è misurabile quantitativamente.
- **Inimitabile**: non deve essere imitabile o sostituibile da terzi.

Le caratteristiche biometriche possono essere divise in due classi principali:

1. **Fisiologiche**: legate alla forma del corpo, tra cui ci sono:

- a. Impronte digitali
- b. Volto

- c. Geometria della mano
- d. Palmo della mano
- e. Vene della mano
- f. Iride
- g. Vene del polso ecc...

2. **Comportamentali:** legate al comportamento di una persona, tra cui:

- a. Firma
- b. Stile di battitura
- c. Camminata
- d. Voce

In generale le caratteristiche fisiologiche sono grandezze stabili e poco variabili nel tempo, a differenza di quelle comportamentali che sono dinamiche e possono essere influenzate anche da fattori emotivi. Motivo per cui sono considerate più affidabili le grandezze fisiologiche rispetto a quelle comportamentali.

I sistemi di riconoscimento **biometrico** si stanno rapidamente diffondendo a livello globale poiché sono in grado di offrire maggiore sicurezza rispetto ai sistemi di autenticazione tradizionali. Questo perché i sistemi di riconoscimento basati su ID e password verificano l'oggetto e non il possessore, di conseguenza ci possono essere diverse problematiche come: i furti d'identità, le frodi, la pirateria informatica etc.

Mentre i sistemi di riconoscimento biometrico sono basati su caratteristiche biometriche che autenticano il possessore, rendendo impossibile o alquanto complesso le fughe di dati di autenticazione.

2.1 PROCESSO DI RICONOSCIMENTO BIOMETRICO

Un sistema di riconoscimento biometrico vuole garantire l'unicità della persona, infatti, è un sistema informatico che ha la funzionalità e lo scopo di riconoscere una persona sulla base di una o più caratteristiche fisiologiche e/o comportamentali, confrontandole con i dati, precedentemente acquisiti e presenti nel database del sistema, tramite algoritmi e sensori di acquisizione.

Il principio di funzionamento di un sistema di riconoscimento biometrico è basato su tre fasi:

1. Fase di **registrazione** (Enrollment): il sistema identifica le caratteristiche biometriche dell'utente e lo memorizza nel database del sistema in forma di **templates**.
 - i. **Templates:** sono dati caratteristici e codificati ottenuti dalle features uniche di un **esempio** biometrico, in sostanza sono i valori tipici di indicatori biometrici di una persona. Un template “non dovrebbe permettere di ricostruire” un esempio valido. Per ogni individuo sono generalmente memorizzati più template al fine di prendere in considerazione possibili variazioni della caratteristica in esame, inoltre vengono aggiornati periodicamente.
 - ii. **Esempio:** campione biometrico grezzo acquisito dal sistema durante una sessione di scansione.

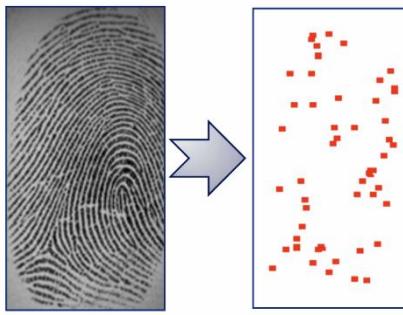


Figura 2.1 – Passaggio da esempio a template

2. Fase di **Verifica**: si confronta la nuova impronta biometrica (il nuovo esempio) con un'altra di riferimento presente nel database (template), si esegue dunque un confronto uno a uno (1:1) per determinare se l'identità dichiarata sia vera o meno, risponde quindi alla domanda: “Sono chi dichiaro di essere?”.
3. Fase di **Identificazione**: si confronta la nuova impronta biometrica con tutte quelle presenti nel database, in questo caso si esegue un confronto uno a molti (1: molti) per stabilire l'identità dell'individuo, risponde alla domanda: “Chi sono?”.

La modalità **verifica** viene impiegata esclusivamente per il riconoscimento *positivo*, mentre la modalità **identificazione** sia per il riconoscimento positivo che *negativo*. Vediamo in dettaglio cosa significano i due termini:

- *Positivo*: indica una modalità di riconoscimento che è finalizzata ad impedire che più persone usino la stessa identità. Stabilisce se la persona è chi dichiara di essere.
- *Negativo*: indica una modalità di riconoscimento che ha lo scopo di impedire che una singola persona utilizzi più identità. Stabilisce se l'individuo è chi nega di essere.

Di seguito vediamo uno schema che riassume il funzionamento di un sistema biometrico.

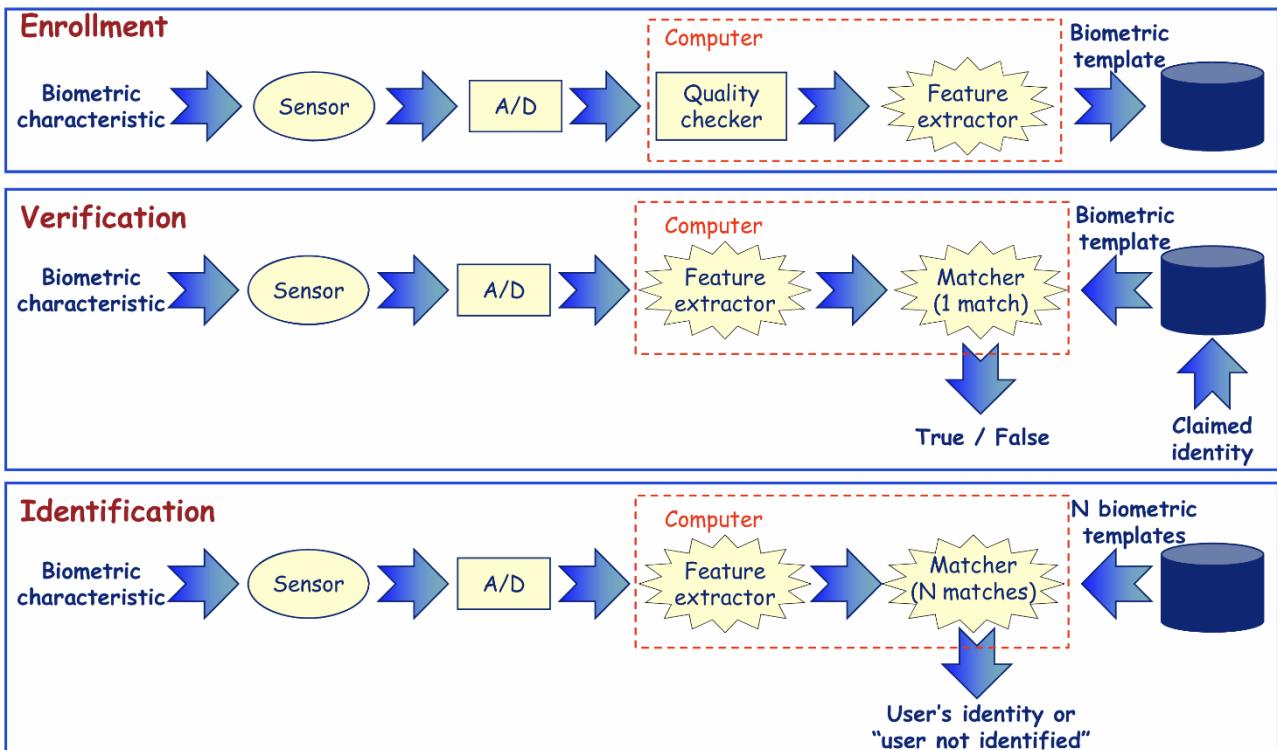


Figura 2.2 – Architettura di un sistema biometrico

L'operazione di estrazione delle features (*features extraction*) comune a tutte e tre le fasi è un'operazione in cui il dato acquisito viene processato per estrarre i valori delle features, i dati biometrici vengono convertiti in template digitali. L'operazione di *matching* invece avviene solo nella fase di verifica e identificazione e prevede che i valori presenti nelle features estratte siano confrontati con quelli presenti nel template al fine di generare un **punteggio** (score) finale. Lo **score** finale del matching viene confrontato con un valore di soglia prefissata in modo da avere un solo risultato finale tra i due possibili: match o non match. I sistemi biometrici non forniscono mai un match al 100%. Il punteggio può essere:

- **Genuino:** se è il risultato del matching di due esempi (template salvato con il nuovo template) della stessa feature di un individuo.
 - Uno score genuine inferiore alla soglia prefissata determina una falsa reiezione (*false non-match*)
- **Impostore:** se nasce dal confronto tra due esempi di individui diversi.
 - Un impostore che supera la soglia causa una falsa accettazione (*false match*)
- **False Match:** termine con la quale si fa riferimento a misurazioni biometriche di persone diverse che vengono erroneamente valutate come appartenenti allo stesso individuo.
- **False Non-Match:** termine con la quale si fa riferimento a misurazioni biometriche della stessa persona le quali sono erroneamente valutate appartenenti ad individui diversi.

2.2 METRICHE E PRESTAZIONI

Sulla base di quanti *false match* e *false non-match* si verificano è possibile definire le seguenti metriche:

- **False Acceptance Rate** o False Match Rate (**FAR** o **FMR**): indica la percentuale di *false match* verificati, una metrica derivata da questa è la ZeroFMR che corrisponde al valore FMR = 0;
- **False Rejection Rate** o False Non-Match Rate (**FRR** o **FNMR**): indica la percentuale di *false non-match* che si verificano, analogamente al caso precedente il valore ZeroFNMR si ottiene quando FNMR = 0;

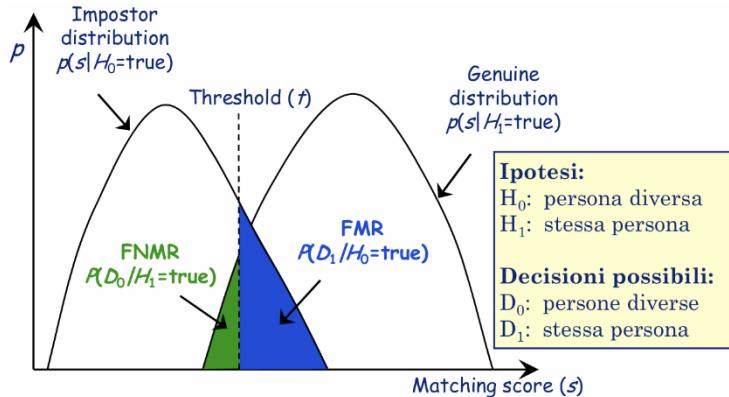


Figura 2.3- Distribuzione degli score di impostori e genuini

La relazione che intercorre tra le due è inversamente proporzionale infatti: all'aumentare della FAR si riduce la metrica FRR; quindi, all'aumentare della FRR si verifica un decremento della FAR. A partire dalla FAR e FRR si definiscono altre metriche importanti come:

- **Genuine Match Rate (GMR)**: ossia la probabilità che non si verifichino false non-match, analiticamente può essere espresso come: $1 - \text{FNMR}$;
- **Equal Error Rate (EER)**: indica la percentuale di errore che si ottiene quando FMR è pari a FNMR, graficamente interpretabile come il punto di intersezione delle due curve (FMR, FNMR), minore sarà l'EER migliore sarà la performance del sistema. La metrica EER valuta il margine di errore del sistema biometrico.

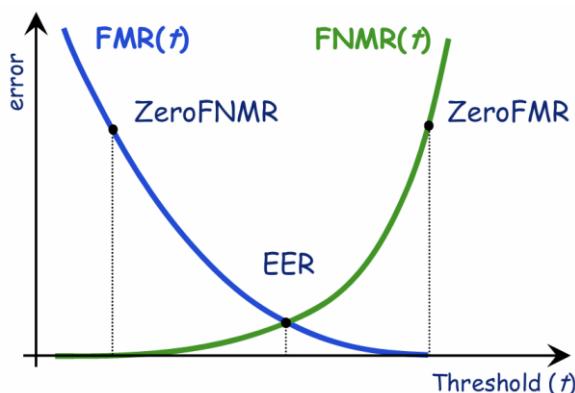


Figura 2.4- Rappresentazione grafica delle metriche di un sistema di riconoscimento biometrico

Un parametro fondamentale, accennato precedentemente è la soglia (t) la quale determina la tolleranza di errore, all'aumentare della soglia si ha una diminuzione della FAR e conseguentemente un aumento della FRR, per via del rapporto di proporzionalità indiretta che lega le due. Aumentando "t" si rende più difficile per gli impostori superare la soglia, ma si aumenta il rischio di respingere utenti legittimi, e viceversa. Riducendo "t" si semplifica notevolmente l'accesso degli utenti legittimi, ma si può aumentare il rischio di accettare impostori. FRR e FAR sono rappresentati come funzioni della soglia "t" con FRR come una curva monotona crescente e FAR come una curva monotona decrescente.

Un altro strumento molto importante per valutare le prestazioni di un sistema di riconoscimento biometrico sono:

- Curve **DET** (*Detection Error Tradeoff*): in cui viene riportata la FMR in funzione della FNMR. In questo caso il valore di EER è ottenuto dall'intersezione delle curve con la prima bisettrice. Pertanto, quanto più la curva è vicina agli assi, tanto migliori sono le prestazioni del sistema biometrico.
- Curve **ROC** (*Receiver Operating Characteristic*): in cui viene riportato il GMR in funzione della FMR.

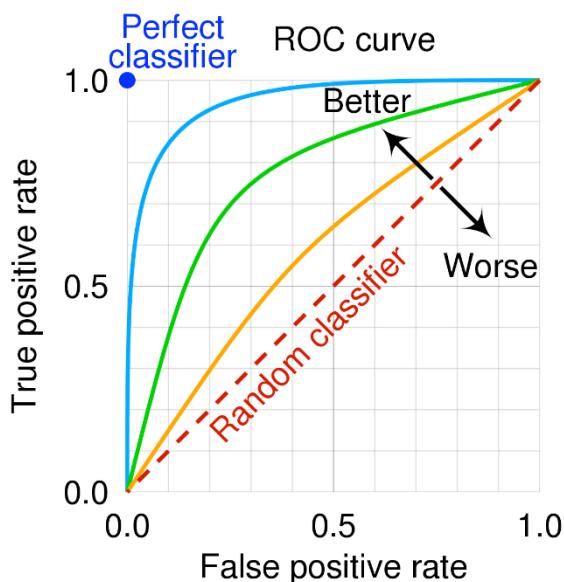


Figura 2.5- Esempio curva ROC

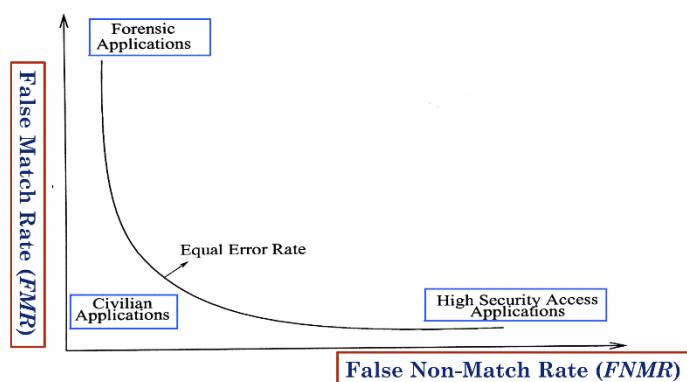


Figura 2.6 - Esempio curva DET

2.3 IL PALMVEIN

La tecnica di autenticazione basata sulla rilevazione della configurazione venosa all'interno del palmo è considerata una delle più moderne, sicure ed igieniche. Questo approccio si basa sull'utilizzo di luce infrarossa generata da un emettitore con una frequenza che consente alla luce di penetrare attraverso gli strati superficiali della pelle del palmo senza richiedere un contatto diretto. La luce infrarossa permette di creare un'immagine estremamente dettagliata della rete venosa del palmo della mano. Questa immagine catturata viene elaborata fino a generare una stringa di dati, nota come "template." La configurazione delle vene all'interno del palmo è distintiva per ciascun individuo e rimane stabile per

un periodo di tempo considerevole. Una delle caratteristiche distintive di questa tecnica biometrica è che, poiché le vene si trovano al di sotto della superficie della pelle, l'identificazione è virtualmente immune da errori dovuti a sporco o tagli sulle dita. A differenza, ad esempio, delle impronte digitali, in cui l'identificazione può essere influenzata negativamente da dita sporche o danneggiate, la scansione delle vene offre una solida autenticazione in ogni situazione. In sintesi, il riconoscimento basato sulla configurazione venosa del palmo è un approccio altamente affidabile, sicuro e igienico per l'autenticazione delle persone, che si basa su caratteristiche biometriche uniche e difficilmente alterabili.

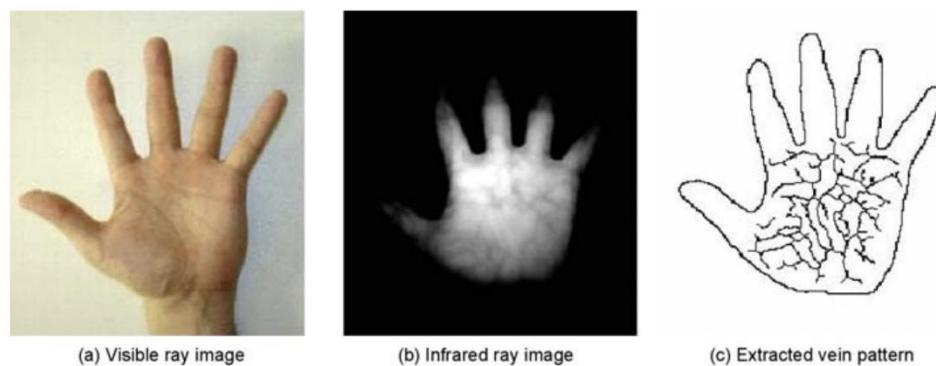


Figura 2.7 - Estrazione dei pattern venosi della mano

2.4 IL PALMPRINT

Questa tecnica di identificazione si basa sulla superficie interna del palmo della mano per estrarre caratteristiche come le linee principali e le rughe. È stato stimato che circa il 30% delle impronte latenti rinvenute sulla scena del crimine sono impronte del palmo di una mano. Ne consegue che i sistemi di riconoscimento del palmprint sono necessari, soprattutto in ambito forense. Vediamo alcuni vantaggi e svantaggi di questa caratteristica.

Vantaggi:

- Permette l'estrazione di un gran numero di features trattando un'area ampia.
- Accuratezza elevata e tempi di elaborazione ridotti.
- Unicità elevata.

Svantaggi

- Caratteristica ancora in fase sperimentale
- Necessita di dispositivi di acquisizione di grandi dimensioni

2.4.1 Caratteristiche

Il palmo presenta tre tratti essenziali

- Linee principali (pieghe o *line*)
- Linee secondarie (rughe o *wrinkles*)
- Creste (*ridges*)
- Valli (*valleys*)

- Pori (pores)
- Minuzie (minutiae)

L'area palmare è ricca di informazioni e, oltre ai tratti principali, presenta numerose features quali pori, minuzie, creste e valli. Le linee principali e le rughe sono caratteristiche dominanti nelle immagini del palmpoint a bassa risoluzione, al contrario minuzie e creste sono visibili solo in immagini ad alta risoluzione.

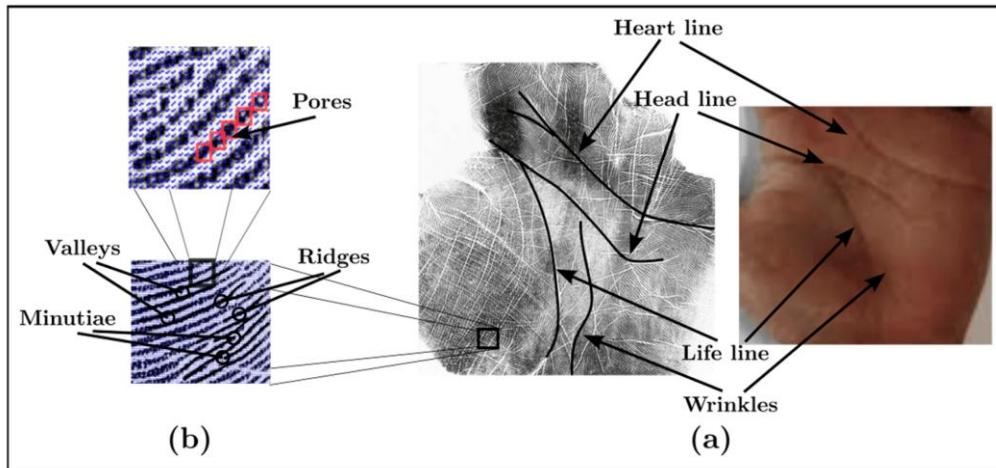


Figura 2.8 – Caratteristiche del palmpoint

L'estrazione di minuzie, ridge line e singolarità richiede immagini a risoluzione elevata (superiore a 400 dpi). Da immagini a 100 dpi si possono estrarre informazioni relative alle linee principali e alle pieghe che però non risultano sufficienti ai fini dell'identificazione.

2.4.2 Sistema di acquisizione

L'analisi del palmpoint è composta da 4 fasi principali come:

- **Acquisizione**
- **Pre-processing**
- **Features extraction**
- **Matching**

Vediamo più in dettagli le fasi.

2.4.2.1 Acquisizione

In questa fase viene acquisita l'immagine, il palmpoint viene scansionato per mezzo di un dispositivo di input.

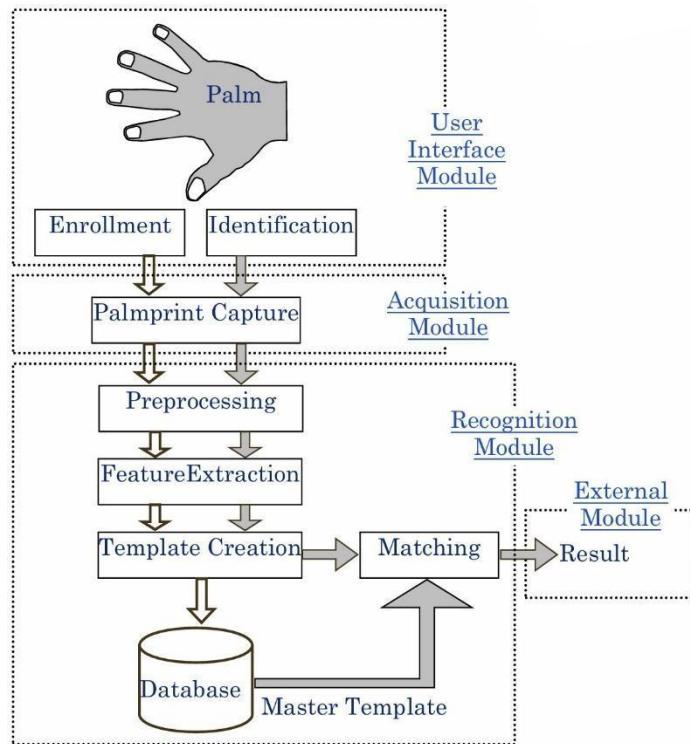


Figura 2.9 - Schema del sistema di acquisizione

2.4.2.2 Pre-processing

Nel pre-processing l'immagine acquisita viene preparata per l'elaborazione. Tipicamente in questa fase si cerca di migliorare la qualità dell'immagine attraverso l'applicazione di filtri di smoothing. Fase necessaria per facilitare il corretto allineamento di palmprint diversi, si localizza la regione centrale del palmo per l'estrazione delle feature.

- **Step 1:** Si procede alla *binarizzazione* dell'immagine a livelli di grigio e si applica successivamente un filtro di smoothing passa-basso.

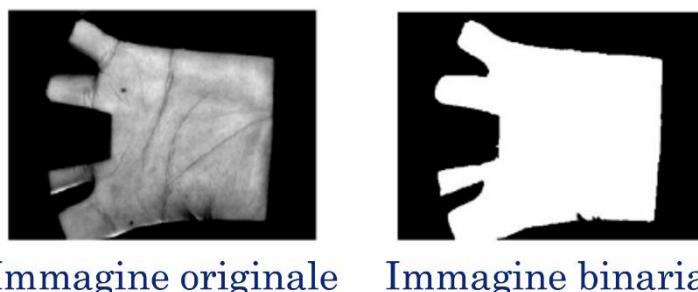


Immagine originale Immagine binaria

Figura 2.10 -Risultato step di binarizzazione

- **Step 2:** Si individuano due zone:
 - H_1 : zona tra il dito indice e il medio
 - H_2 : zona tra il mignolo e l'anulare
- **Step 3:** Si determinano i punti T_1 e T_2 per le due regioni.

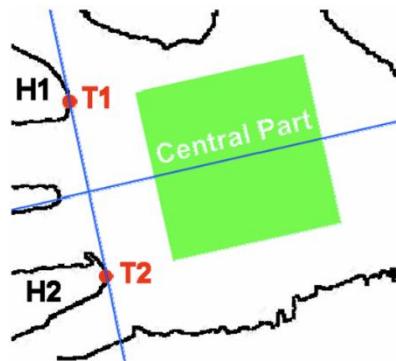


Figura 2.11 - Risultato step individuazione keypoints

- **Step 4:** Si definisce un nuovo sistema di coordinate disegnando due linee: una collega T_1 e T_2 (assey) l'altra è perpendicolare all'assey e passante per il punto medio di T_1 e T_2 (assex).

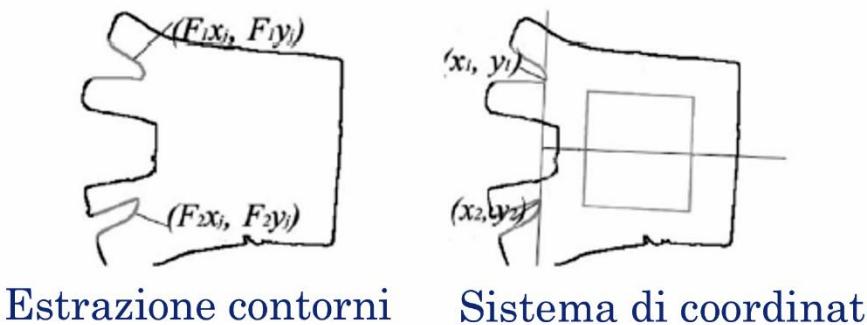
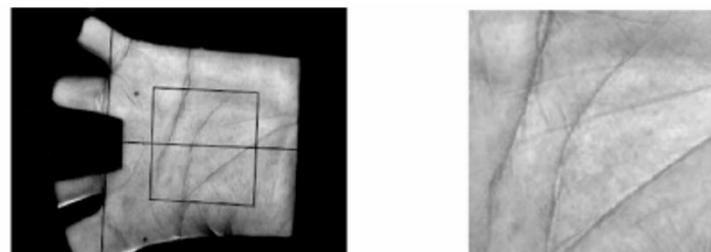


Figura 2.12 -Risultato step individuazione sistema di coordinate

- **Step 5:** Nel nuovo sistema di coordinate si estrae una sotto immagine di dimensione prefissata, dalla quale saranno poi estratte le feature per il riconoscimento.



Regione centrale Risultato pre-processing

Figura 2.13 -Risultato step estrazione regione centrale

2.4.2.3 Features extraction

In questa fase, attraverso opportuni algoritmi, vengono estratte le caratteristiche principali come linee, rughe e valli creando i template corrispondenti. L'estrazione delle features quali avviene attraverso vari step di elaborazione dell'immagine acquisita nella fase precedente. In particolare vengono applicate procedure come:

- Regolazione del Contrast
- Resize
- Filtraggio

- Riduzione del rumore

Alla fine di questa fase verrà ottenuta un'immagine con le features messe in risalto come l'immagine seguente:

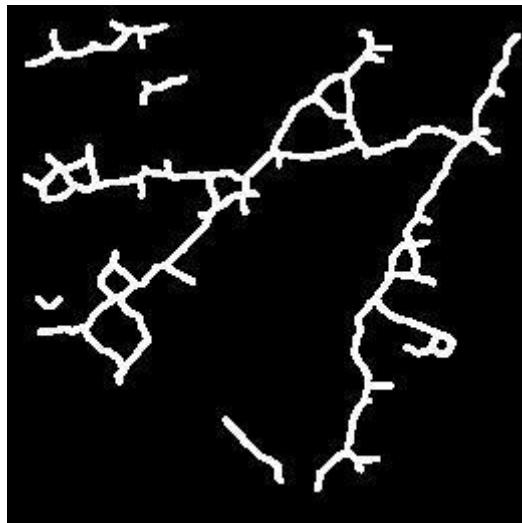


Figura 2.14 -Risultato fase di estrazione delle features

2.4.2.4 Matching

Nell'ultima fase vengono confrontati i due templates attraverso opportune metriche come la distanza di *Hamming*, il *pixel to area* o il *voxel to area* per valutare la similità tra i due template (2D o 3D) tramite lo score. Un valore numerico compreso nell'intervallo [0, 1], dove il valore 0 sta ad indicare discordanza totale, mentre il valore 1 indica assoluta affinità.

A seconda della tipologia dei templates (2D o 3D) da valutare, è possibile distinguere due tipologie di matching:

2.4.2.5 Matching 2D

Il processo di matching 2D tra due templates è il cosiddetto confronto “pixel to area”. Questo approccio implica l'esecuzione di un'operazione AND logico tra i pixel corrispondenti delle due immagini seguita da una normalizzazione, che prende in considerazione la media aritmetica del totale dei pixel bianchi presenti in ciascuna immagine. Formalmente, è definito come:

$$Score_{2D} = \frac{2 \sum_{x=1}^n \sum_{y=1}^m (T_{1d}(X, Y) > 0) \oplus (T_{1t}(X, Y) > 0)}{\sum_{x=1}^n \sum_{y=1}^m T_{1d}(X, Y) + \sum_{x=1}^n \sum_{y=1}^m T_{1t}(X, Y)}$$

Equazione 2.1 - Formula score 2D

La formula calcola un punteggio chiamato Score2D basato su due templates 2D: T_{1d} e T_{1t} . Spieghiamo un attimo il significato dei vari componenti.

- Numeratore:

- La funzione $T_{1d}(X, Y) > 0$ è una funzione indicatrice che restituisce 1 se il valore di T_{1d} in un punto specifico (X, Y) è maggiore di 0 e 0 altrimenti.
- Analogamente $(T_{1t}(X, Y) > 0)$, è una funzione indicatrice per T_{1t} .
- L'operatore \oplus rappresenta l'operazione AND logico (spesso simboleggiata con \wedge) tra i due valori risultanti dalle funzioni indicatori.
- Il doppio sommatorio $\sum_{x=1}^n \sum_{y=1}^m (\dots)$ implica che stiamo sommando su tutti i valori di X da 1 a n e Y da 1 a m.

- Denominatore:
 - La somma $\sum_{x=1}^n \sum_{y=1}^m T_{1d}(X, Y)$ rappresenta l'aggregazione di tutti i valori nel template T_{1d} attraverso le sue dimensioni.
 - Analogamente, $\sum_{x=1}^n \sum_{y=1}^m T_{1t}(X, Y)$ rappresenta l'aggregazione di tutti i valori nel template T_{1t} attraverso le sue dimensioni.

Il punteggio finale $Score_{2D}$ sarà un valore tra 0 e 1, rappresentando un indice normalizzato basato sulla somiglianza tra i due templates T_{1d} e T_{1t} attraverso un confronto diretto tra i loro pixel.

2.4.2.6 Matching 3D

La procedura utilizzata per i matchings 3D è simile a quella dei matchings 2D, dove formula adoperata, viene leggermente modificata come segue:

$$Score_{3D} = \frac{2 \sum_{x=1}^n \sum_{y=1}^m |T_d(X, Y) - T_t(X, Y)| < \alpha \oplus (T_d(X, Y) > 0) \oplus (T_t(X, Y) > 0)}{\sum_{x=1}^n \sum_{y=1}^m T_d(X, Y) + \sum_{x=1}^n \sum_{y=1}^m T_t(X, Y)}$$

Equazione 2.2 – Formula score 3D

Il punteggio $Score_{3D}$ viene calcolato utilizzando due templates: T_d e T_t . Osservando l'equazione fornita, possiamo analizzare i componenti principali:

- Numeratore:
 - La funzione $|T_d(X, Y) - T_t(X, Y)| < \alpha$ indica che stiamo calcolando la differenza assoluta tra i valori dei due templates in un punto specifico (X, Y) e confrontando se questa differenza è minore di un certo valore α .
 - $T_d(X, Y) > 0$ e $T_t(X, Y) > 0$ sono funzioni indicatrici che restituiscono 1 se i rispettivi valori nei templates in un punto specifico sono positivi.
 - L'operatore \oplus rappresenta l'operazione AND logico tra i due valori risultanti dalle funzioni indicatrici. Il doppio sommatorio $\sum_{x=1}^n \sum_{y=1}^m (\dots)$ implica una somma su tutti i valori di X da 1 a n e Y da 1 a m.
- Denominatore:
 - La somma $\sum_{x=1}^n \sum_{y=1}^m T_d(X, Y)$ implica una somma su tutti i valori di X da 1 a n e Y da 1 a m, rappresenta l'aggregazione di tutti i valori nel template $T_d(X, Y)$ attraverso le sue dimensioni.
 - Analogamente, $\sum_{x=1}^n \sum_{y=1}^m T_t(X, Y)$ rappresenta l'aggregazione di tutti i valori nel template $T_t(X, Y)$ attraverso le sue dimensioni.

Questo punteggio $Score_{3D}$ rappresenta un indice di somiglianza tra i due templates T_d e T_t , ma con una condizione aggiuntiva: si tiene conto solo delle differenze tra i valori dei templates che sono inferiori al valore soglia α .

Quindi il risultato di questa fase è la produzione dello **score** che può assumere un andamento tipo quello in figura

Threshold	1 template image		3 template images	
	FAR (%)	FRR (%)	FAR (%)	FRR (%)
0.32	0.000027	8.15	0.000012	5.12
0.34	0.00094	4.02	0.0016	2.18
0.36	0.011	1.94	0.017	0.86
0.38	0.096	1.05	0.15	0.43
0.40	0.68	0.59	1.03	0.19

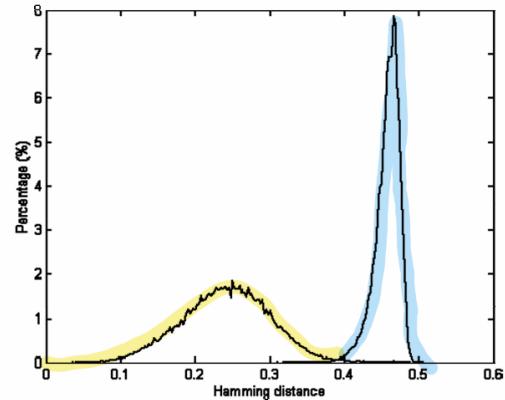


Figura 2.15 - Esempio risultati della fase di matching

Una volta terminata la fase di Matching (2D o 3D), è possibile valutare lo score ottenuto. In particolare, se lo score associato al matching supera una certa soglia prefissata t , l'identificazione avrà successo; in caso contrario, fallirà.

Per valutare le prestazioni del sistema biometrico, si utilizzano i parametri FAR ed FRR definiti come:

- **FAR** = frequenza con cui il sistema biometrico accetta erroneamente utenti non legittimi.

$$FAR = \frac{FP}{T}$$

Equazione 2.3 - Formula del FAR

- **FRR** = frequenza con cui il sistema respinge erroneamente utenti legittimi.

$$FRR = \frac{FR}{T}$$

Equazione 2.4 - Formula FRR

- Dove
 - **FA** è il numero di false accettazioni (falsi positivi)
 - **FR** è il numero di false reiezioni (falsi negativi)
 - **T** è il numero di processi di riconoscimento

3 GENERAZIONE DI ULTRASUONI

L'acquisizione del palmprint e dei pattern venosi viene effettuata tramite una sonda ecografica ad ultrasuoni, quindi vediamo la metodologia e i dettagli tecnici di tale tecnica di scansione.

3.1 ULTRASUONI

Gli ultrasuoni sono onde acustiche generate dalla trasmissione di vibrazioni meccaniche attraverso il mezzo (come aria o i tessuti del corpo umano). Trasferiscono al mezzo ai quali vengono applicate solamente energia meccanica eccitando le molecole del mezzo stesso. Il numero di vibrazioni per unità di tempo, detta **frequenza di vibrazione**, permette di determinare la qualità del suono prodotto. In genere, lo spettro sonoro può essere suddiviso in tre parti:

- Frequenze udibili:** sono le frequenze percepibili dall'udito umano, il cui intervallo di frequenze è compreso tra 20 Hz e 20 KHz.
- Infrasuoni:** sono caratterizzati dall'avere una frequenza inferiore rispetto a quella delle frequenze udibili.
- Ultrasuoni:** sono caratterizzati dall'avere una frequenza maggiore rispetto a quella delle frequenze udibili.

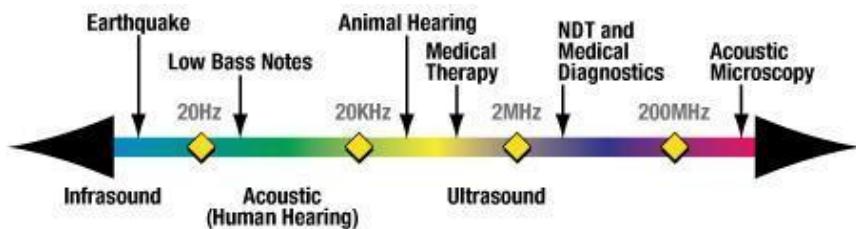


Figura 3.1 -Spettro delle frequenze sonore

3.1.1 Emissione di ultrasuoni

Una sonda ecografica è un dispositivo progettato per generare e rilevare segnali ad ultrasuoni. Questi segnali possono essere prodotti in due modi:

- Attraverso un'emissione continua di ultrasuoni, se il segnale ad ultrasuoni viene generato in modo continuo. È il metodo più comunemente utilizzato nella maggior parte delle applicazioni pratiche.
- Attraverso impulsi successivi, noti come emissione pulsata, se invece il segnale ad ultrasuoni viene generato attraverso impulsi successivi.

La sorgente tipica per generare gli ultrasuoni è un cristallo (piezoelettrico) pilotato elettricamente a vibrare (effetto piezoelettrico) e posto in contatto con la superficie esterna del mezzo, che nel nostro caso è rappresentata dal corpo umano.

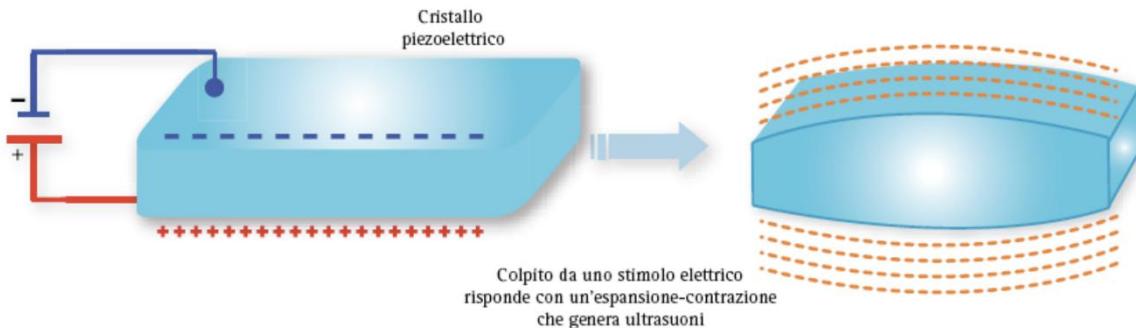


Figura 3.2 -Principio della piezoelettricità

La sonda è composta da numerosi cristalli piezoelettrici che sono in grado di emettere onde ultrasoniche. Questi cristalli possono essere sottoposti a due fenomeni:

- Effetto piezoelettrico **diretto** si verifica quando i cristalli piezoelettrici subiscono una compressione o un'estensione a causa di uno stress meccanico. L'energia meccanica applicata viene convertita in carica elettrica superficiale. In genere, viene utilizzato per la rilevazione degli ultrasuoni.

- Effetto piezoelettrico **inverso** si verifica quando viene applicata una differenza di potenziale alle estremità di un cristallo piezoelettrico. Il cristallo risponde espandendosi o contraendosi, convertendo l'energia elettrica applicata in energia meccanica. In genere, viene utilizzato per la produzione di ultrasuoni.

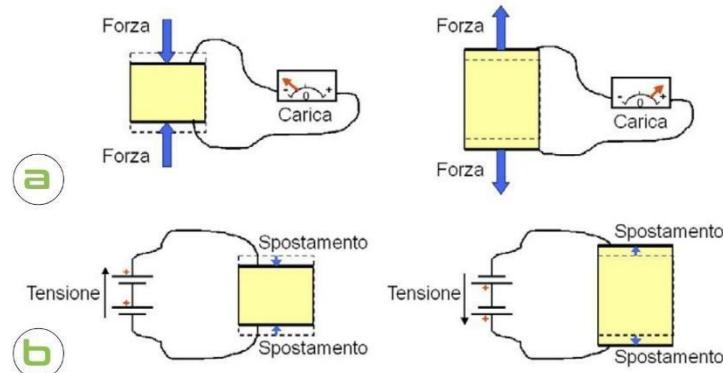


Figura 3.3 - a) effetto piezoelettrico diretto - b) effetto piezoelettrico inverso

Grazie alla reversibilità di questo fenomeno, gli stessi cristalli possono essere utilizzati sia per generare che per rilevare gli ultrasuoni.

3.1.2 Propagazione degli ultrasuoni

Un segnale ad ultrasuoni può essere generato da una sonda, ovvero da un dispositivo ecografico, costituito da tanti materiali cristallini piezoelettrici. Gli ultrasuoni per potersi propagare hanno bisogno di un mezzo materiale. La sonda trasferisce il disturbo meccanico al mezzo con il quale viene a contatto, innescando nelle sue particelle delle vibrazioni, che, oltre ad influenzare le particelle circostanti, innescano anche un trasferimento di energia. Nel caso preso in esame, le onde che si vengono a generare sono di tipo longitudinali, in quanto lo spostamento delle particelle avviene nella stessa direzione in cui si verifica l'oscillazione della sorgente ultrasonica.

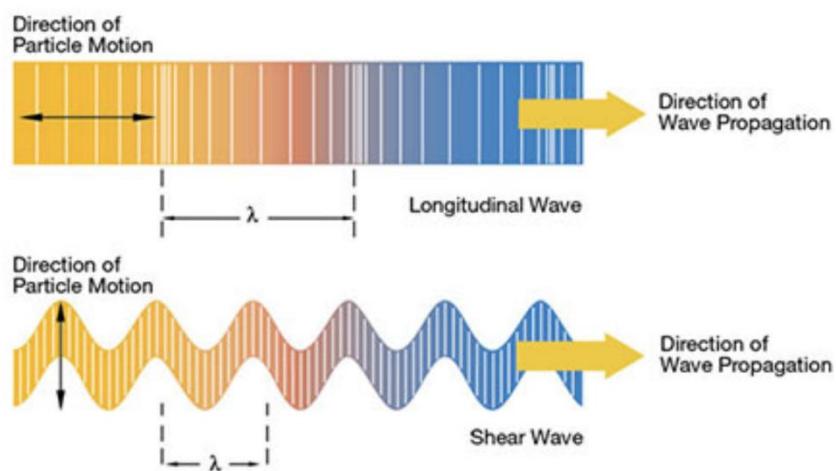


Figura 3.4 - Propagazione di onde longitudinali e trasversali

I movimenti periodici delle particelle possono essere rappresentati come una curva sinusoidale aventi determinate caratteristiche.

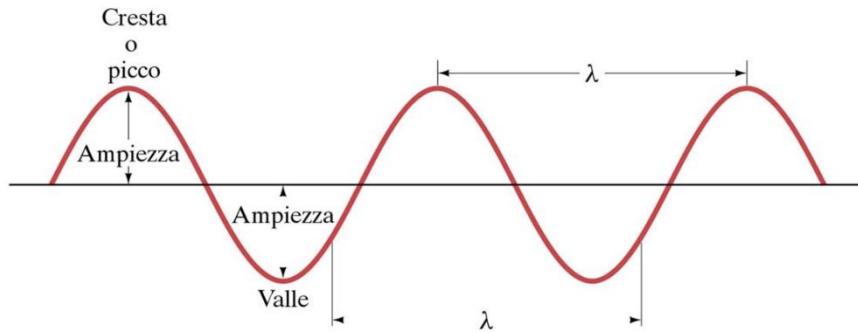


Figura 3.5 - Onda sinusoidale

In particolare:

1. **Periodo T:** viene espresso in secondi e indica il tempo che impiega un'onda per compiere un ciclo completo.
2. **Frequenza f:** viene misurata in Hz ed è l'inverso del periodo

$$f = \frac{1}{T}$$

Equazione 3.1

3. **Velocità di propagazione c:** viene espressa in m/s ed indica lo spazio percorso dal fronte d'onda nell'unità di tempo. Essa dipende dal mezzo in cui si propaga l'onda, quindi dal suo modulo di Young (E), dalla sua densità (ρ) e dal suo coefficiente di Poisson (ν); Nell'acqua e nei tessuti ad alto contenuto di H₂O è di 1540 m/s.
4. **Lunghezza d'onda λ :** viene misurata in metri e rappresenta la distanza percorsa dal fronte d'onda nel tempo T a velocità c

$$\lambda = cT$$

Equazione 3.2

5. **Pulsazione ω :** indica il numero di radianti nell'unità di tempo, viene quindi misurata in rad/s

$$\omega = 2\pi f$$

Equazione 3.3

6. **Impedenza acustica Z:** è la capacità dei mezzi di opporsi al passaggio delle onde ultrasoniche, diminuendone così l'intensità. Si misura in *rayl* dove ρ è la densità del mezzo attraversato dagli ultrasuoni.

$$Z = \rho c$$

Equazione 3.4

7. **Pressione acustica p:** è rappresentata da una sinusoide in funzione della distanza di propagazione. La pressione sonora ha importanza fondamentale poiché è il parametro dell'onda direttamente rilevabile a livello strumentale. L'unità di misura è il Pascal (Pa).
8. **Intensità acustica I:** rappresenta la quantità d'energia che attraversa l'unità di superficie nell'unità di tempo, viene misurata in [W/m²]

$$I = \frac{p^2}{2Z}$$

Equazione 3.5

Nelle tecniche diagnostiche che utilizzano le immagini, gli ultrasuoni sono impiegati per le loro alte frequenze. Queste ultime provocano una riduzione della lunghezza d'onda, permettendo quindi un'immagine di maggiore qualità grazie ad una miglior risoluzione spaziale.

3.1.2.1 Integrazioni con i tessuti

Gli ultrasuoni emessi dalla sonda possono attraversare diverse tipologie di tessuto, caratterizzati dall'avere diverse densità e diversa impedenza acustica. La definizione dell'impedenza acustica (Equazione 3.4) evidenzia che materiali diversi rispondono in modo diverso con gli ultrasuoni. In particolare, le aree del tessuto, in cui i valori dell'impedenza acustica variano, sono chiamate confini acustici o interfacce di tessuto.

MEZZO	VELOCITÀ PROPAGAZIONE (m/s)	IMPEDENZA (Rayl x 10 ⁻⁵)
Vuoto	0	-
Aria	330	0.0004
Grasso	1450	1.38
Acqua	1480	1.48
Fegato	1550	1.65
Rene	1560	1.62
Sangue	1570	1.61
Muscoli	1580	1.70
Osso	4080	7.8
Media tessuti molli	1540	1.63

Figura 3.6 - Velocità di propagazione degli US nell'aria, nell'acqua e nei tessuti biologici.

Nei punti delle interfacce, in corrispondenza delle quali il valore dell'impedenza acustica varia, si possono verificare tre fenomeni differenti:

- **Riflessione**

- Quando un'onda acustica colpisce la superficie di separazione di due mezzi materiali, parte dell'onda viene trasmessa, quindi attraversa la superficie di separazione, mentre la parte rimanente viene riflessa con lo stesso angolo incidente e si propaga nella superficie in direzione della sonda.
- Il coefficiente di riflessione è definito come il rapporto tra l'intensità dell'onda riflessa e l'intensità dell'onda incidente:

$$R = \frac{I_r}{I_i} = \left(\frac{Z_1 - Z_2}{Z_1 + Z_2} \right)^2$$

Equazione 3.6 - coefficiente di riflessione

- I_r = intensità dell'onda riflessa
- I_i = intensità dell'onda incidente
- Z_1 = impedenza del mezzo 1

- Z_2 = impedenza del mezzo 2
- Se $R = 1$, si ha che l'onda incidente coincide con l'onda riflessa e, di conseguenza, l'onda viene completamente riflessa.
- Se $R = 0$, si ha che l'onda incidente non subisce alcuna riflessione.

● Trasmissione

- La trasmissione si verifica quando l'onda acustica viene trasmessa e, quindi, si propaga nel tessuto, con un angolo differente rispetto a quello incidente.
- Il coefficiente di trasmissione è definito come il rapporto tra l'intensità dell'onda trasmessa e l'intensità dell'onda incidente.

$$T = \frac{I_t}{I_i} = \frac{4Z_1Z_2}{(Z_1 + Z_2)^2}$$

Equazione 3.7 - coefficiente di trasmissione

- I_t = intensità dell'onda trasmessa
- Se $T = 1$, si ha che l'onda incidente coincide con l'onda trasmessa e, di conseguenza, l'onda viene completamente trasmessa.
- Se $T = 0$, si ha che l'onda incidente non subisce alcuna trasmissione.

● Rifrazione

- Il fenomeno della rifrazione si verifica quando la direzione dell'onda acustica cambia completamente direzione nel momento in cui entra in contatto con la superficie di separazione di due mezzi materiali aventi differenti velocità di propagazione.

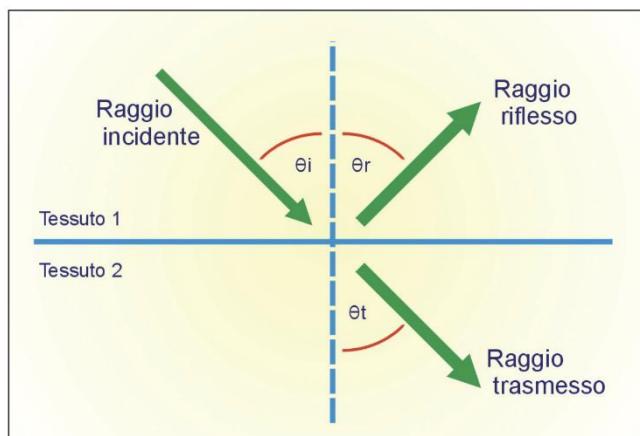


Figura 3.7 - Riflessione diffusa e speculare. L'energia acustica riflessa è tanto più intensa quanto più l'angolo di incidenza degli US si avvicina a 90°

L'**eco** è un'onda riflessa che arriva alla sonda a causa del fenomeno di riflessione. A causa delle irregolarità delle interfacce, si verifica anche il fenomeno di **diffusione**, che è caratterizzato da onde multiple riflesse con angoli diversi da quelli d'incidenza e a bassa intensità. Tuttavia, per scopi diagnostici, solo l'onda riflessa rappresenta un'informazione utile per la sonda.

L'immagine ecografica viene costruita a partire da due informazioni:

- Il *tempo di arrivo* degli echi, che ci dice quanto è profonda la sorgente;
- La *posizione degli echi* rispetto all'asse, che ci dà la posizione laterale nello spazio.

Da qui si comprende perché le immagini ecografiche vengono acquisite in **acqua** o **gel** e mai in **aria**. Quando l'impedenza acustica tra due mezzi è molto diversa, gran parte dell'energia degli ultrasuoni viene **riflessa** alla superficie di contatto (come nel caso dell'aria e della pelle). Invece, se le impedenze sono più simili (come tra acqua e tessuti), gli ultrasuoni riescono a penetrare e trasmettersi efficacemente nel mezzo successivo. Con la sonda in aria le onde sonore verrebbero quasi totalmente riflesse e l'immagine risulterebbe completamente bianca. Effetto che è possibile vedere anche quando sono presenti delle ossa durante la scansione ecografica, gli ultrasuoni si riflettono quasi completamente quando colpiscono una superficie ossea, poiché l'osso ha un'impedenza acustica molto più alta rispetto ai tessuti molli circostanti. Questo causa un riflesso forte che appare come una **zona bianca** sull'immagine ecografica, con una perdita di dettagli dietro l'osso (effetto "ombra acustica").

La **risoluzione spaziale** della sonda è la distanza minima tra due sorgenti vicine che permette di distinguere gli echi prodotti. Questa risoluzione è composta da una componente assiale e una componente laterale. Se queste componenti non sono sufficienti, gli echi risulteranno sovrapposti e sarà impossibile distinguere i tempi di arrivo, perdendo qualità nell'immagine finale.

Un'altra importante relazione tra velocità e frequenza degli ultrasuoni è la seguente:

$$v = \lambda f$$

Equazione 3.8 - velocità di un'onda ultrasonica

dove λ è la lunghezza d'onda, che è anche una misura della massima risoluzione spaziale. Infatti, le strutture con dimensioni inferiori alla lunghezza d'onda non possono interagire con gli ultrasuoni e, quindi, non possono essere rilevate. Di conseguenza, per migliorare la risoluzione spaziale è sufficiente aumentare la frequenza. Tuttavia le onde con frequenze più elevate sono anche più soggette ad attenuazione, di conseguenza la profondità massima raggiunta dagli ultrasuoni diminuisce.

A seconda del tipo di applicazione, è necessario trovare un compromesso per poter meglio immaginare gli organi e i tessuti a diverse profondità. Più un soggetto di studio è profondo, più bassa dovrebbe essere la frequenza.

Inoltre, un altro fattore contribuisce a diminuire la risoluzione massima. Infatti, le onde vengono normalmente inviate a pacchetti, comprendendo quindi più di un ciclo, cioè più di una λ . Se due oggetti lungo il percorso degli ultrasuoni si trovano a una distanza inferiore alla lunghezza del pacchetto, i loro echi non sono distinguibili. Una soluzione sarebbe quindi quella di utilizzare meno cicli in una trasmissione. Tuttavia, quando si inviano pacchetti più corti, gli echi ricevuti presentano larghezze di banda maggiori e non permettono di ricostruire la frequenza corretta delle onde. D'altro canto, pacchetti più lunghi permettono di distinguere la frequenza prevista, ma la risoluzione peggiora inevitabilmente.

Un vantaggio molto importante degli ultrasuoni è che possono essere facilmente focalizzati attraverso l'uso di lenti posizionate sopra i trasduttori, oppure attraverso una focalizzazione dinamica, applicando ritardi diversi all'**array di trasduttori**. L'utilizzo di fasci focalizzati è una pratica molto comune, in quanto permette di concentrare l'energia degli ultrasuoni in zone specifiche. Di conseguenza, la risoluzione spaziale può essere aumentata nel punto in cui viene posizionato il fuoco. Di solito, lungo una linea di scansione, vengono applicati più punti di focalizzazione, per migliorare la risoluzione anche a diverse profondità.

3.1.2.2 Attenuazione

Il fenomeno dell'attenuazione acustica è quel fenomeno che causa una perdita di intensità durante la propagazione dovuta a due cause principali:

1) Scattering:

È causato dall'interazione dell'onda con eventuali discontinuità e disomogeneità presenti nel mezzo che provoca una parziale diffusione di energia in direzioni diverse dalla direzione principale in cui l'onda si propaga. Si verifica per la differenza di impedenza acustica tra i due mezzi, in prossimità dell'interfaccia tra discontinuità e materiale.

2) Assorbimento: è dovuto al fatto che una parte dell'energia dell'onda viene trasferita al mezzo sotto forma di calore, tale contributo è tanto più elevato quanto più è elevata la frequenza. La diminuzione di intensità dovuta all'attenuazione è espressa dalla legge di Lambert-Beer:

$$I(z) = I_0 e^{-\alpha z}$$

Equazione 3.9 - Legge di Lambert-Beer

- Dove:

- z è lo spessore di materiale attraversato dal fascio.
- I_0 è l'intensità del fascio incidente, cioè per $z = 0$.
- α è detto coefficiente di attenuazione lineare, e dipende dal mezzo attraversato;

3.2 SONDA ECOGRAFICA

Gli ecografi sono apparecchiature di imaging medicale che si basano sull'emissione di una serie di impulsi ad ultrasuoni e sulla conseguente rilevazione degli echi di ritorno dovuti al principio della riflessione. Concettualmente sono costituiti da una sonda, un generatore a ultrasuoni che pilota i piezoelettrici e che deve essere in grado di generare la frequenza e l'ampiezza della sinusoide di cui abbiamo bisogno e infine un circuito di ricezione costituito da un demodulatore acustico che va a ricercare un impulso di frequenza nota all'interno del rumore; tipicamente la sonda svolge un doppio ruolo, non solo l'emissione degli impulsi ma anche la ricezione degli echi. Come noto, per ottenere l'immagine, si sfrutta il fenomeno della riflessione, in pratica l'impulso emesso dalla sonda si propaga lungo il tessuto sino ad incontrare una discontinuità, a questo punto una parte dell'impulso viene riflesso e viene ricevuto dalla sonda che ne calcola l'ampiezza. L'ampiezza dell'eco che torna alla sonda è minore dell'ampiezza dell'impulso inizialmente emesso, a causa dei fenomeni di attenuazione visti in precedenza.

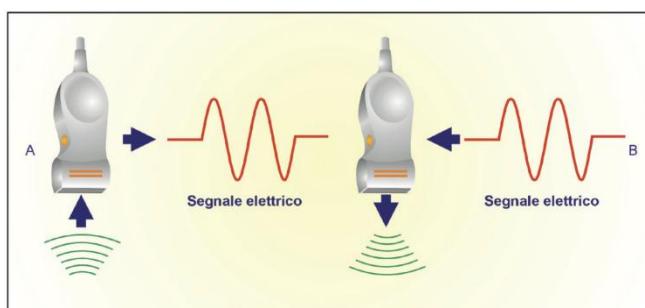


Figura 3.8 - Effetto piezoelettrico diretto e indiretto. Gli elementi di ceramica del trasduttore convertono l'energia acustica in segnale elettrico e l'energia elettrica in impulso US

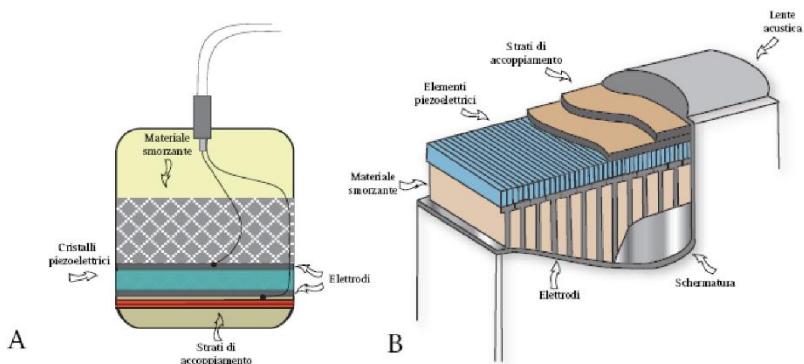


Figura 3.9 - Struttura di una sonda

Dietro i cristalli si trova un materiale di backing che ha il doppio compito di smorzare le vibrazioni del cristallo e di assorbire energia sonora che potrebbe creare interferenza. Tra i cristalli piezoelettrici e la faccia della sonda che si appoggia sul paziente si trovano: un adattatore di impedenza e uno strato di gomma. L'adattatore di impedenza viene utilizzato per evitare di avere un coefficiente di riflessione troppo elevato all'interfaccia cristallo/tessuti in quanto l'impedenza acustica dei cristalli è molto maggiore rispetto all'impedenza acustica dei tessuti molli; lo strato di gomma, che costituisce la parte più esterna della sonda funge da lente acustica per focalizzare il fascio di ultrasuoni riducendo la distanza focale.

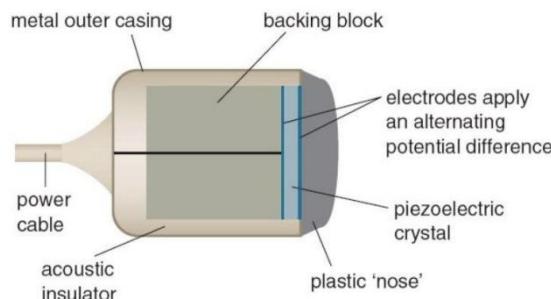


Figura 3.10 - Struttura di una sonda

3.2.1 Tipologie di sonde

Sulla base di come sono disposti i cristalli nella sonda possiamo identificare due tipologie di sonde:

- **Sonda linear array:** i cristalli (in numero fino a 512) sono disposti in linea. Il trasduttore fornisce quindi un'immagine di forma rettangolare. Un trasduttore linear array è più utile per esaminare oggetti di piccole dimensioni (spesso in associazione a sistemi Doppler: es. sonde per vasi).



Figura 3.11 - Sonda linear array

- Sonda **convex array**: gli elementi piezoelettrici sono disposti su un supporto curvilineo e vengono eccitati in simultaneo o in sequenza. Permette di ottenere una sezione “a ventaglio”.

Sulla base del tipo di stimolazione che ricevono i cristalli possiamo distinguere due classi di sonde:

- Sonde **non phased-array**: i cristalli vengono eccitati per lo più in contemporanea.
- Sonde **phased-array**: sono costituite da una schiera di elementi piezoelettrici che vengono attivati in sequenza, ognuno con un opportuno ritardo. Il principale vantaggio di questo tipo di sonde risiede nel fatto di avere la possibilità di variare l’angolo di incidenza del fascio senza però modificare la posizione della sonda, per questo motivo sono le più utilizzate in ecocardiografia.
 - In questo modo, il fronte d’onda che si forma è inclinato (Figura 3.12 A) in modo diverso rispetto alla superficie del trasduttore a seconda del ritardo con cui gli elementi piezoelettrici vengono successivamente eccitati.
 - Con un ritardo pari a 0, l’onda prosegue parallela alla superficie del trasduttore
 - Con ritardi progressivamente differenti, il fascio ultrasonoro può essere variamente orientato, e anche fatto oscillare (Figura 3.12 B), ottenendo anche in questo caso una scansione “a ventaglio”.
 - Se il ritardo viene variato come indicato in Figura 3.12 (in modo “curvilineo”), si può ottenere l’effetto di “focalizzare” gli ultrasuoni su un punto prescelto. L’entità del ritardo fra i vari elementi determina la profondità del fuoco del fascio ultrasonoro trasmesso, che può essere anche variata durante l’esecuzione della scansione.

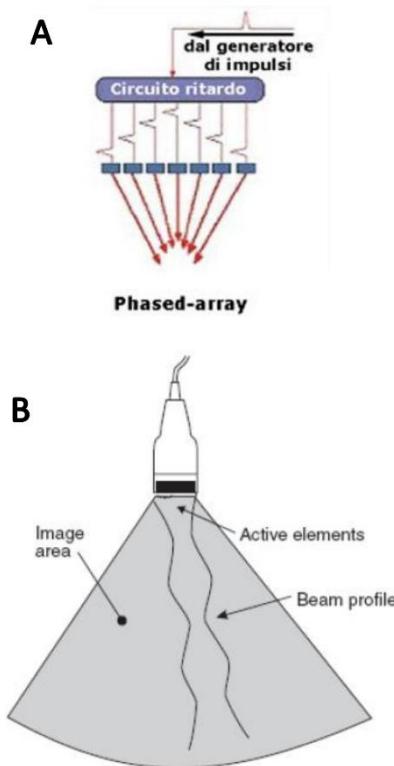


Figura 3.12- Rappresentazione sonda phased array

3.2.2 Generazione del fascio

Il profilo del fascio è costituito da gradienti di pressione in direzione trasversale e assiale. Nell'illustrazione del profilo del fascio riportata di seguito, il rosso rappresenta la zona ad alta energia, mentre il verde e il blu rappresenta la zona a bassa energia.

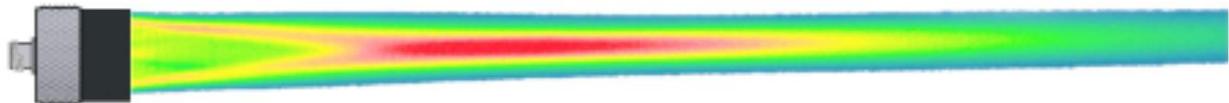


Figura 3.13 – Rappresentazione del fascio generato da una sonda

La regione nella quale il fascio possiede una maggiore intensità e una minore larghezza è chiamata zona focale. Il campo acustico di una sonda è diviso in due zone, il campo vicino e il campo lontano. Il campo vicino è la zona in prossimità della sonda dove la pressione acustica passa per una serie di massimi e di minimi. Termina all'ultimo massimo sull'asse a una distanza N dalla superficie della sonda. La distanza N del campo vicino rappresenta il centro naturale della sonda.

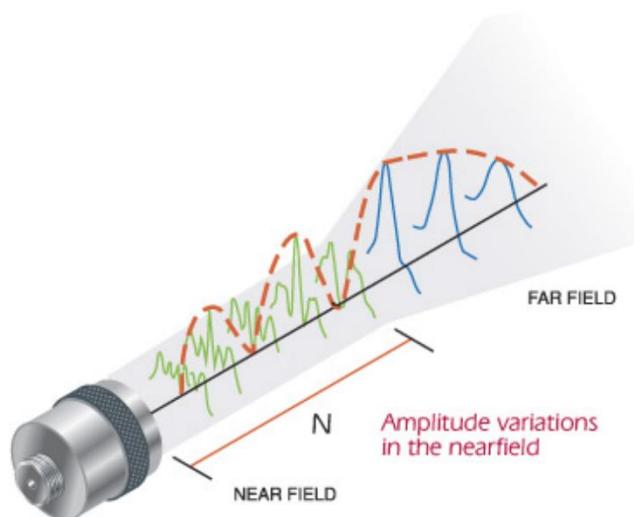


Figura 3.14 - Campo vicino e campo lontano di una sonda

Assumendo la zona focale come riferimento possiamo distinguere due parti all'interno del fascio:

1. **Campo vicino** (o zona Fresnel): situata tra la sonda e la zona focale, nella quale il fascio ha forma cilindrica e la massima capacità di risoluzione laterale, la sua estensione è direttamente proporzionale al quadrato del raggio del trasduttore e inversamente proporzionale alla lunghezza d'onda dell'ultrasuono che si propaga nel mezzo;
2. **Campo lontano** (o zona di Fraunhofer): situata aldilà della zona focale, che presenta una geometria divergente e nella quale la risoluzione laterale degrada in maniera importante.

Esistono inoltre dei raggi ultrasonori aventi un'intensità molto più bassa rispetto al fascio principale, detti lobi laterali, che si generano a causa dell'espansione radiale dei cristalli piezoelettrici.

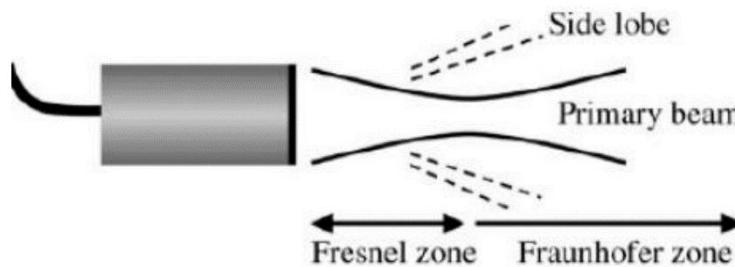


Figura 3.15 - Zone di un fascio ad ultrasuoni

3.2.3 Formazione dell'immagine ecografica

Siccome lo stesso trasduttore deve funzionare sia da emittente del fascio ultrasonoro che da ricevente dell'eco, dovrà alternare in modo congruo le due funzioni.

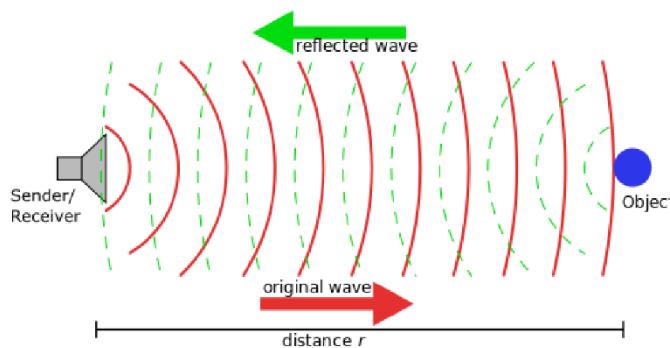


Figura 3.16 – Trasmissione e ricezione delle onde

A seconda della distanza dell'oggetto che vogliamo esplorare, il trasduttore emetterà quindi un primo treno di ultrasuoni, e dovrà poi attendere prima di emettere un nuovo impulso, rimanendo in ascolto per un tempo adeguato. Più le strutture da esplorare saranno profonde, maggiore sarà il tempo di ascolto e quindi minore il numero di impulsi inviati per unità di tempo (di conseguenza sarà minore la qualità dell'immagine prodotta).

Per ottenere l'immagine ecografica, il mezzo viene insonificato utilizzando più fasci focalizzati in sequenza e l'immagine viene ricostruita affiancando ogni linea di scansione. Senza considerare i potenziali tempi di elaborazione per ricostruire l'immagine o per migliorarne la qualità, questa procedura permette di ottenere un solo fotogramma, dopo che i canali del trasduttore hanno successivamente trasmesso i fasci e ricevuto gli echi riflessi. Pertanto, il tempo richiesto per produrre un'immagine è:

$$T_{frame} = \frac{N_{lines} \cdot 2 \cdot z}{c}$$

Dove:

- N_{lines} = numero di linee di scansione
- z = profondità massima dell'immagine
- c = velocità di propagazione nel mezzo

Le immagini ecografiche utilizzate in questo lavoro sono state acquisite utilizzando la tecnica del **pulse-echo imaging**, che utilizza le onde sonore emesse da una sonda trasduttore a contatto con la pelle. Queste onde, viaggiando attraverso il corpo, interagiscono con strutture di diversa densità e

composizione, come ossa, muscoli e fluidi. In particolare, quando un'onda sonora incontra un confine tra due tipi di tessuto, parte di essa viene riflessa indietro verso la sonda, creando un "eco". Questo fenomeno, noto come riflessione, avviene, ad esempio, quando l'onda sonora incontra una superficie che non può attraversare, come l'osso. Allo stesso tempo, una parte dell'onda può cambiare direzione e continuare a viaggiare attraverso la struttura, dando vita al fenomeno della rifrazione, che si verifica quando l'onda passa da un mezzo a un altro con diversa densità, come dall'acqua al grasso. La sonda rileva gli echì, sia riflessi che rifratti, e registra il tempo che impiegano per ritornare alla sonda. Poiché la velocità del suono nei tessuti del corpo è nota, questo tempo può essere utilizzato per calcolare la distanza tra la sonda e il confine del tessuto che ha riflettuto l'onda sonora. Questo processo viene ripetuto per diversi angoli e posizioni, creando una serie di misurazioni che possono essere utilizzate per creare un'immagine bidimensionale o tridimensionale dell'area di interesse. Infine, queste misurazioni vengono elaborate da un'unità di elaborazione, che le converte in un'immagine che viene visualizzata su uno schermo, permettendo di esaminare le strutture interne del corpo in tempo reale. Esistono tre modalità per generare un'immagine ad ultrasuoni.

3.2.3.1 A-SCAN

Durante la scansione il trasduttore può rimanere fisso in una posizione, o, registrando la stessa vista con gli echì posizionati negli stessi punti, formando un'immagine A-Scan (**Amplitude mode**) in cui ogni eco è mostrato come un picco di ampiezza sul monitor indicando sia la sua intensità che la profondità da cui proviene.

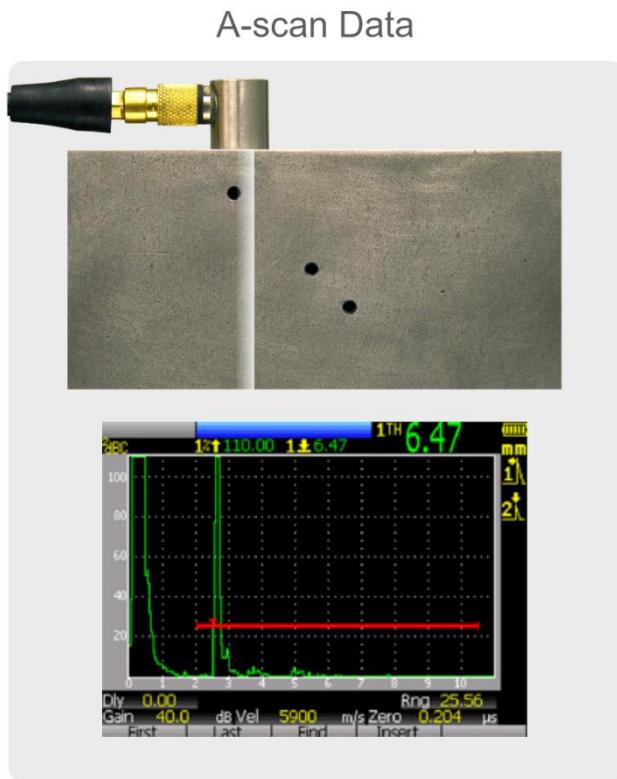


Figura 3.17 – Immagine A-SCAN

Dall'immagine Figura 3.17 possiamo vedere come ogni eco è un picco di cui si rappresenta intensità e profondità di provenienza. L'intensità si trova sull'asse verticale, la profondità sull'orizzontale. La modalità A-scan quindi permette di generare un'immagine ad ultrasuoni **1D**.

3.2.3.2 B-SCAN

In questa modalità di scansione, durante il movimento del trasduttore lungo il corpo del paziente, si registra la posizione a cui si riceve l'eco per cui si riesce a realizzare un'immagine bidimensionale su uno schermo: modalità B-Mode. L'immagine è considerando per ogni eco, la sua posizione, che coincide con la posizione della sonda al momento della sua cattura, la sua profondità e la sua intensità. In base all'intensità l'eco è colorato con un diverso gradiente di grigio (perciò si parla di **Brightness Mode**). In questo modo si riesce ad ottenere un'immagine che ricostruisce il profilo dei riflettori. La modalità B-Mode migliora nettamente la qualità delle immagini. Per mantenere alta la risoluzione, è fondamentale che il trasduttore sia mantenuto sullo stesso piano di scansione durante l'intero processo e che vengano fornite sempre le medesime coordinate x, y e θ. Questo assicura che ogni scansione produce un'immagine coerente e dettagliata.

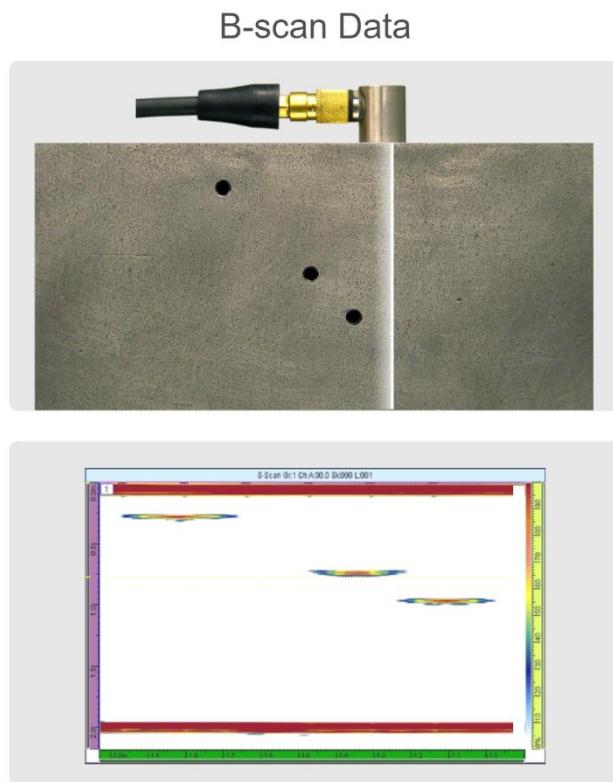


Figura 3.18 – Immagine B-SCAN

La modalità B-scan quindi prevede la realizzazione di immagini **2D** costituite da differenti immagini A-SCAN ottenute spostando la sorgente in direzioni diverse durante l'acquisizione. In particolare, l'immagine B-SCAN è un'immagine in scala di grigi, dove la luminosità di ciascun pixel dipende dall'intensità dell'eco di ritorno.

3.2.3.3 C-SCAN

La modalità C-MODE (**Constant-Range Mode**) produce come risultato un'immagine **2D**, chiamata CSCAN, ottenuta usando un piano normale a quello dell'immagine B-MODE a una specifica profondità. Anche in questo caso, l'immagine 2D è un'immagine in scala di grigi.

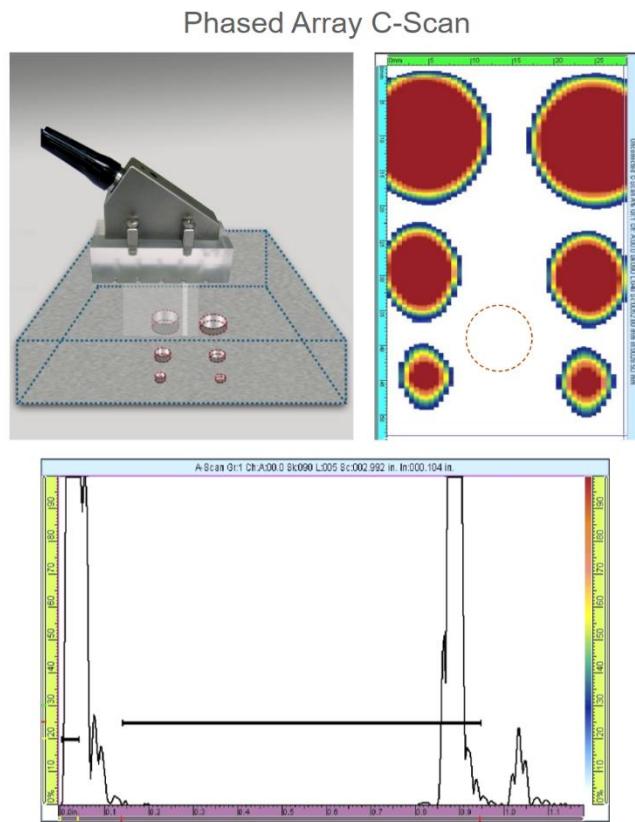


Figura 3.19 – immagine C-SCAN

4 DESCRIZIONE DEL PROGETTO

4.1 SCOPO DEL PROGETTO

Il nostro progetto parte dall’analisi del database Luongo-Scavone, con l’obiettivo iniziale di valutare la possibilità di distinguere in modo preciso e chiaro i pattern venosi. Questa necessità nasce dall’esigenza di superare le limitazioni del database fornito, che non ha permesso ai nostri predecessori di estrarre entrambe le caratteristiche biometriche con un livello di precisione elevato, dando priorità al **palmpoint**. Una delle principali cause di questa scarsa riconoscibilità dei pattern venosi è da ricercarsi nella scelta di un guadagno inadeguato durante la fase di acquisizione delle immagini, probabilmente impostato troppo basso, generando immagini eccessivamente scure. Tuttavia l’idoneità del database verrà confermata (6.2.1) mediante l’utilizzo del metodo di estrazione a singola vena sviluppato dalla Dott.ssa Vizzuso. Sulla base di queste considerazioni, l’obiettivo finale del progetto è stato lo sviluppo di un sistema personalizzato per il riconoscimento delle vene palmari. Questo sistema si propone di superare le limitazioni del metodo Vizzuso, consentendo l’estrazione dell’intero **pattern venoso**, preservando la struttura morfologica naturale delle vene e garantendo, al contempo, un sistema il più possibile replicabile.

- **Guadagno**

- La sonda spara un segnale con una certa energia, l’onda si propaga nel mezzo (corpo umano) e quando incontra una differenza di impedenza una parte viene riflessa e una parte viene trasmessa. Le onde che penetrano di più nel tessuto restituiranno un’onda riflessa con intensità molto bassa, in fase di ricezione quindi saranno molto deboli. È

necessario amplificare queste onde, in particolare bisogna amplificare maggiormente le onde che vengono dai tessuti più in profondità rispetto a quelle che provengono da tessuti superficiali. Nel file di configurazione si trovano due guadagni, uno fisso e uno variabile; quello variabile viene indicato con una pendenza e si misura in dB/decade.

- **Immagini del database**

- Luongo e Scavone, probabilmente, hanno impostato un guadagno troppo lieve e le immagini risultanti sembrano essere troppo scure. Per cui è opportuno prendere confidenza con queste immagini e utilizzare *Biometric data representation* (BDR) per vedere se è possibile riconoscere i pattern venosi anche se le immagini sono scure. Eventualmente modificare qualche parametro di *BDR* per adattarla al tipo di qualità di immagine che abbiamo.

4.2 BIOMETRIC DATA REPRESENTATION (BDR)

BDR, detto anche ‘ula op’, fornisce in uscita un file .uob ad alta frequenza che viene riconvertito a bassa frequenza restituendo delle immagini punto per punto. I dati poi vengono processati per ricostruire il volume acquisito e vengono fatte immediatamente delle operazioni preliminari (filtraggi, binarizzazioni, ecc...). I parametri da impostare per queste operazioni dipendono dal tipo di immagini. Se l’immagine risulta scura bisogna dover variare qualche parametro per farla risultare più chiara e così via.

4.3 OBIETTIVI

Il preprocessing deve fare in modo che i pattern venosi siano meglio individuabili attraverso BDR. Se sono individuabili allora si può pensare ad un algoritmo che estragga questi pattern venosi.

La prima fase del nostro progetto consiste nell’esplorare il database e analizzare le acquisizioni per controllare se è possibile distinguere i pattern venosi. Analizzare i pattern e decidere se le immagini sono scure o meno. Successivamente bisogna analizzare il codice di BDR per comprendere tutti i suoi step (filtraggio, binarizzazione, interpolazione).

4.4 MOTIVAZIONI

4.4.1 Palmprint

La caratteristica unica del palmprint è la sua area molto più ampia rispetto all’impronta digitale, il che permette di estrarre numerose caratteristiche distintive. La sua accettazione elevata, il basso grado di intrusione rispetto ad altre modalità biometriche e la precisione acquisita nel corso degli anni lo rendono una scelta eccellente per il riconoscimento individuale. Un altro vantaggio notevole risiede nella capacità di utilizzare immagini a bassa risoluzione, poiché le feature possono essere estratte con precisione anche da immagini di questo tipo, riducendo così la complessità tecnologica. Fino a tempi recenti, i sistemi di riconoscimento del palmprint si basavano principalmente su immagini palmari 2D, che hanno dimostrato ottimi risultati in termini di precisione e velocità nel riconoscimento individuale. Tuttavia, le impronte palmari 2D presentano svantaggi intrinseci. L’area dell’impronta palmare 2D, sebbene piuttosto ampia, può essere compromessa da contaminazioni e abrasioni eccessive nella superficie del palmo, il che può invalidare il processo di riconoscimento. Inoltre, le immagini 2D sono sensibili alle variazioni dell’illuminazione ambientale, aumentando il rischio di falsi riconoscimenti. Infine, le impronte palmari 2D possono essere soggette a contraffazione e perdita di informazioni.

strutturali presenti nel palmo 3D. Di conseguenza, al fine di affrontare queste problematiche di sicurezza, sono state sviluppate diverse metodologie basate sul palmprint 3D, le quali registrano informazioni di profondità relative alla superficie del palmo osservata. Integrando queste informazioni tridimensionali alle caratteristiche 2D, è possibile ottenere una scansione 3D in grado di ricreare l'intera superficie del palmo osservato. Il riconoscimento basato sull'impronta palmare 3D è inoltre più robusto rispetto alle variazioni di illuminazione e alle lesioni gravi sulla superficie del palmo.

4.4.2 Palmvein

La caratteristica distintiva del palmvein è l'utilizzo del pattern delle vene all'interno del palmo, il che permette di estrarre numerose caratteristiche uniche e difficilmente replicabili. La sua alta accettazione, il basso grado di intrusione rispetto ad altre modalità biometriche e l'elevata precisione ottenuta negli ultimi anni lo rendono una scelta eccellente per il riconoscimento individuale. Un altro vantaggio significativo risiede nella capacità di utilizzare immagini non invasive, poiché il pattern venoso può essere rilevato attraverso la pelle con tecniche di imaging a infrarossi, riducendo così la complessità tecnologica e aumentando la sicurezza. Tradizionalmente, i sistemi di riconoscimento palmvein si sono basati su immagini bidimensionali (2D) del pattern venoso, che hanno dimostrato ottimi risultati in termini di precisione e velocità nel riconoscimento individuale. Tuttavia, le immagini palmvein 2D presentano alcuni svantaggi intrinseci. Le vene del palmo possono variare a causa di fattori fisiologici come la temperatura, la pressione sanguigna e la condizione della pelle, che possono influire sulla qualità dell'immagine e sul processo di riconoscimento. Inoltre, le immagini 2D sono sensibili alle variazioni ambientali, come la temperatura e l'umidità, aumentando il rischio di falsi riconoscimenti.

Per affrontare queste problematiche, sono state sviluppate metodologie basate sul palmvein tridimensionale (3D), che registrano informazioni di profondità relative alla struttura venosa del palmo osservato. Integrando queste informazioni tridimensionali alle caratteristiche 2D, è possibile ottenere una rappresentazione più accurata e dettagliata del pattern venoso, migliorando la robustezza del sistema contro variazioni fisiologiche e ambientali. Il riconoscimento basato sul palmvein 3D è inoltre più resistente alle lesioni e ai cambiamenti superficiali del palmo, poiché la struttura venosa interna è meno soggetta a danneggiamenti rispetto alla superficie esterna della pelle. L'utilizzo di tecniche 3D riduce significativamente il rischio di contraffazione, poiché replicare il pattern venoso tridimensionale è estremamente complesso. In conclusione, l'adozione di tecnologie di riconoscimento basate sul palmvein 3D rappresenta un significativo passo avanti nel campo della biometria, offrendo un metodo sicuro, preciso e non invasivo per il riconoscimento individuale.

Un esempio è la tecnologia di autenticazione contactless basata sul pattern delle vene del palmo sviluppata da Fujitsu che utilizza i pattern vascolari come dati di identificazione personale. La tecnologia di riconoscimento delle vene è sicura perché i dati di autenticazione esistono all'interno del corpo e quindi sono molto difficili da falsificare. È anche altamente accurata: nei test condotti su 140.000 profili palmari di 70.000 individui, ha riscontrato un tasso di falsa accettazione inferiore allo 0,00008% e un tasso di falso rifiuto dello 0,01%. [1]

4.4.3 Sistemi multimodali

Negli ultimi anni, c'è stata un'attenzione crescente verso l'analisi di una singola caratteristica biometrica, ma le prestazioni spesso si sono rivelate insoddisfacenti. Le metodologie monomodali affrontano problematiche come la presenza di rumore nei dati, limitazioni nei gradi di libertà, mancanza di universalità e la possibilità di attacchi di spoofing. Per ottenere sistemi di riconoscimento più robusti, è spesso necessaria la fusione di diverse caratteristiche biometriche, dando vita ai cosiddetti sistemi multimodali. Una delle grandezze più utilizzate nei sistemi multimodali è il palmprint, che grazie alla

sua versatilità può essere facilmente fuso con altre caratteristiche, come la geometria della mano (hand geometry) e le vene del palmo (palm vein). Ad esempio, un sistema di identificazione multimodale è quello proposto da J.G. Wang [2], il quale prevede la fusione delle immagini del palmprint e delle vene del palmo in un'unica immagine.

4.4.4 Acquisizione del palmprint e del palmvein

L'acquisizione del palmprint e del palmvein rappresenta la fase iniziale del processo di riconoscimento biometrico. I sistemi di acquisizione si suddividono principalmente in due categorie:

- Offline: In questo approccio, l'acquisizione e la digitalizzazione delle impronte avvengono separatamente.
- Online: In questo caso, l'acquisizione e la digitalizzazione delle impronte avvengono simultaneamente. L'impronta palmare viene catturata direttamente da un dispositivo di input come uno scanner, una fotocamera o sensori ottici. Questo metodo rappresenta attualmente il principale approccio per l'acquisizione delle impronte palmari.

Le tecnologie più utilizzate nei sistemi biometrici per l'acquisizione di immagini palmari sono principalmente basate su sensori ottici, come i CCD scanner e scanner digitali. Tuttavia, negli ultimi anni, le tecnologie ad ultrasuoni stanno diventando sempre più rilevanti nell'ambito del riconoscimento biometrico. Questo è dovuto alle numerose peculiarità e vantaggi offerti dalle onde ultrasoniche.

In particolare le tecnologie ottiche o a infrarossi fanno una fotografia della mano quindi sono in grado di estrarre solamente un pattern 2D delle vene. Invece con gli ultrasuoni è possibile esplorare l'intero volume acquistato e quindi estrarre le vene come delle features 3D.

In seguito, ci concentreremo sulla tecnologia ad ultrasuoni per l'acquisizione delle impronte palmari e dei pattern venosi.

4.4.5 Tecnologie ad ultrasuoni

La tecnologia ad ultrasuoni è ampiamente utilizzata nel campo della medicina diagnostica, grazie alla sua capacità di penetrare nei tessuti con diverse impedenze acustiche. Questa caratteristica consente la generazione di echi all'interno del corpo, permettendo la scansione di intere sezioni degli organi. Negli ultimi anni, le tecnologie ad ultrasuoni sono diventate predominanti nel riconoscimento biometrico, specialmente per l'identificazione delle impronte digitali e delle impronte palmari. La capacità degli ultrasuoni di penetrare nei tessuti permette di ottenere informazioni a diversi livelli di profondità, fornendo dati non solo sulla superficie esterna della pelle ma anche sulla struttura interna. Questo rappresenta un vantaggio significativo rispetto alle tecnologie ottiche, che forniscono solo informazioni superficiali. Di conseguenza, i sistemi ad ultrasuoni possono generare rappresentazioni tridimensionali dettagliate delle impronte palmari, contribuendo a migliorare le prestazioni complessive dei sistemi biometrici. Ulteriori vantaggi degli ultrasuoni includono la loro insensibilità a potenziali contaminazioni come olio o inchiostro sulla pelle. Inoltre, le immagini ad ultrasuoni possono verificare la vitalità del campione, ad esempio rilevando il battito di una vena, rendendo più difficile il ricorso a tecniche di falsificazione. Tuttavia, va notato che l'acquisizione tramite ultrasuoni presenta alcune sfide, tra cui costi elevati e la necessità di utilizzare un mezzo di accoppiamento poiché le onde ultrasoniche si riflettono fortemente quando incontrano l'aria. [3]

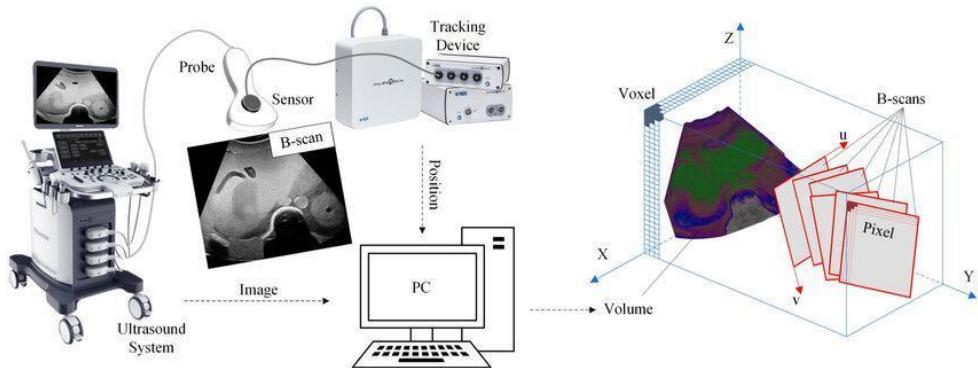


Figura 4.1 - Sistema di acquisizione ad ultrasuoni

4.5 ARCHITETTURA DEL SISTEMA DI ACQUISIZIONE

L’acquisizione del volume del palmo della mano avviene mediante un sistema di imaging costituito principalmente da:

- **Sonda ecografica** (LA435 by Esaote S.p.A., Genova, Italy) usata come trasduttore ad ultrasuoni
- **Pantografo a controllo numerico** (by Delta Macchine CNC, Vazia (RI) - Italy) usato come sistema automatico di manipolazione della sonda.
- **Sistema di imaging ULA-OP, Ultrasound Advanced Open Platform**, una piattaforma aperta avanzata per la ricerca ecografica.

4.5.1 Sonda ecografica

Per acquisire le informazioni del palmo della mano, si è utilizzata la sonda ecografica (Esaote LA435 in modalità B) che permette di ottenere immagini bidimensionali in scala di grigi.

La sonda LA435 è lineare ad alta frequenza (da 6 a 18 MHz) e permette di ottenere immagini, delle strutture superficiali del corpo, ad alta risoluzione e contrasto come, ad esempio, il pattern cutaneo e il pattern venoso. È composta da 192 elementi ed è realizzata con la tecnologia piezocomposita. Ha una frequenza centrale di circa 12 MHz, una larghezza di banda frazionale di circa 75%, e un’apertura totale di 38,4 mm che divisa per il numero di elementi si ha una distanza fra essi di 0.2mm.

La sonda è pilotata da uno scanner ad ultrasuoni, collegata a un pantografo numerico.



Figura 4.2 - Sonda ecografica LA435

4.6 PANTOGRAFO A CONTROLLO NUMERICO

Il pantografo a controllo numerico è uno strumento utilizzato come sistema di movimento automatico della sonda ed è in grado di eseguire pattern complessi 3D con una precisione migliore di 20 µm, può muoversi lungo 3 direzione ed effettuare una rotazione.

Non dispone di un sistema di riferimento assoluto garantendo la libertà di impostare arbitrariamente il valore delle coordinate della posizione del braccio, indipendentemente da dove si trovi localizzato.

il movimento del pantografo è eseguibile tramite semplici operazioni MATLAB.

Nel file **pantografo_acqua** è definita una classe pantografo che si compone delle seguenti proprietà:

- Coordinata dell'origine
- Tipo di movimento
- Velocità (espressa in millimetri al secondo)
- Coordinate di percorso da effettuare



Figura 4.3 - Pantografo a controllo numerico

4.7 SISTEMA ULA-OP

Il sistema ULA-OP, uno strumento completamente sviluppato nel laboratorio universitario di Firenze. Grazie all'integrazione di tutta l'elettronica necessaria per il controllo simultaneo fino a 64 elementi di una sonda array di 192 elementi. Le due schede che costituiscono il sistema possono essere facilmente connesse a qualsiasi PC commerciale tramite la porta USB 2.0, garantendo una semplice interoperabilità e accessibilità.

Questo sistema si presenta come un'importante piattaforma di ricerca ecografica ed offre un terreno fertile per lo sviluppo e la sperimentazione di nuovi metodi di imaging a ultrasuoni. Le sue funzionalità avanzate permettono di testare strategie originali di beamforming, elaborazione delle immagini in tempo reale e tecniche pulsate Doppler e Vector Doppler.

Il sistema ULA-OP può essere programmato per funzionare in una varietà di modalità, ognuna delle quali dipende dalle impostazioni fornite nei seguenti file:

- **File di configurazione (.cfg)**
- **File di strategia TX/RX (.ula)**

Inoltre, è necessario che sia presente un valido **file di configurazione della sonda (.wks)** nella cartella appropriata.

4.8 FILE DI CONFIGURAZIONE (.CFG)

Si evidenziano 11 sezioni principali:

1. **[NAME]:** Questa sezione definisce il nome della configurazione. Il nome viene visualizzato all'interno dell'applicazione "ULA-OP Starter"
2. **[COMMENT]:** Questa sezione contiene commenti sulla configurazione. Ad esempio, "One window with B-Mode only" indica che la configurazione utilizza una sola finestra con la modalità B-Mode
3. **[TREE]:** Questa sezione definisce l'aspetto e il rapporto della configurazione.
 - a. **aspect** è impostato su "1H0", descrive la disposizione delle finestre nella visualizzazione principale del software in tempo reale
 - b. **ratio = "100"** elenco di numeri interi che descrive la disposizione delle finestre nella visualizzazione principale del software in tempo reale.
4. **[SEQUENCER]:** Questa sezione specifica l'ordine degli elementi da acquisire durante la scansione. In questo caso
 - a. **Item0:** Elenco dei file .ula. Ogni file .ula descrive un fascio o un gruppo di fasci impiegati nel sistema.
 - b. **DefItemKey:** Definisce i tasti di scelta rapida di un elemento. Si compone di 4 parametri:
 - c. **iItem** è l'indice dell'elemento corrispondente.
 - d. **eControl** è il nome del parametro che deve essere controllato dai tasti di scelta rapida. Potrebbe essere: "FBurst" (0), "TgcA" (1), "TgcB" (2), "Focus" (3), "TxAmp" (4), "TxNC" (5), "StartingDepth" (6).
 - e. **sKKIncrement** è il riferimento al tasto della tastiera per incrementare il valore del parametro dichiarato in eControl.
 - f. **sKKDecrement** è il riferimento al tasto per decrementare il valore del parametro dichiarato in eControl.
 - g. **iStep** è il valore del passo incrementale. L'unità di misura del passo dipende dal parametro specificato nel campo eControl.
5. **[WORKINGSET]:** Questa sezione definisce le impostazioni di lavoro. Il parametro **SoundSpeed = 1540** indica la velocità dell'onda ultrasonica nel mezzo di propagazione (che nel nostro caso è l'acqua)
6. **[SSG]:** Questa sezione definisce la Mappa delle Frequenze di Ripetizione dell'Impulso
 - a. **PrfsMap = 1000, 2000, 3000,, 12000:** Elenco dei valori della frequenza di ripetizione degli impulsi (PRF), espressi in Hertz [Hz]. L'utente può scegliere la frequenza di ripetizione degli impulsi desiderata
 - b. **Prf** è composto da 4 parametri:
 - i. **12000** è il valore della frequenza di ripetizione degli impulsi (PRF) con cui l'applicazione in tempo reale parte.
 - ii. **z** è il riferimento al tasto per passare tra la modalità "Freeze" (congelamento) e "Unfreeze" (non congelato).

- iii. **x** è il riferimento al tasto per incrementare il valore della frequenza di ripetizione degli impulsi (PRF).
 - iv. **c** è il riferimento al tasto per decrementare il valore della frequenza di ripetizione degli impulsi (PRF).
- 7. [SAVEOPT]:** Questa sezione specifica le opzioni di salvataggio predefinite per le immagini acquisite.
- 8. [ACQUIREIQ]:** Questa sezione definisce le impostazioni per l'acquisizione di dati IQ (In phase and Quadrature-phase), dell'immagine. **slice0 = 256, 48000, 0, SliceIQ, 1, 1:** risiedono nella memoria del DSP e contengono campioni demodulati (IQ) forniti dall'FPGA. La dimensione della slice è determinata dai campi
- a. **iGates (256)** e **iPris (48000)** che corrisponde al numero di intervalli di ripetizione degli impulsi.
 - b. **iStartGate (0)** è l'indice (impostato a zero di default) nel buffer della FPGA da cui recuperare i campioni in arrivo.
 - c. **sName (SliceIQ)** assegna un nome alla slice. Il nome è principalmente utilizzato come suffisso per i nomi dei file quando la slice viene salvata su disco. Se non viene specificato alcun nome, ULA-OP Modula utilizza un nome predefinito (ad esempio, "SliceIQ<i>"). Poiché l'applicazione utilizza tale etichetta per marcare i file durante il salvataggio dei dati della slice su disco, il nome della slice non deve contenere caratteri vietati nei nomi di file, così come gli underscore.
 - d. **iPrfDivisor (1)** specifica il rapporto tra la frequenza di ripetizione degli impulsi (PRF) del sistema e la PRF equivalente (impostata di default iPrfDivisor = 0).
 - e. **eSave (1)** determina se la slice dovrebbe essere salvata per impostazione predefinita o meno: "NoSave" (0) (impostazione predefinita), "Save" (1).
- 9. [BLOCKSEQUENCER]:** questa sezione elenca le sequenze ordinate delle PRI e per ogni PRI specifica il pattern di trasmissione/ricezione e le slice che devono raccogliere i campioni in ingresso.**pri0 = 0, 1, 0, 0, 0:**
- a. **iItem (0)** determina il pattern di trasmissione/ricezione attraverso il suo numero di elemento.
 - b. **iCount (1)** determina la quantità da aggiungere o sottrarre al contatore del pattern.
 - c. **iSliceIQ (0):** le slice che devono raccogliere i campioni provenienti dalla fase di ricezione nella PRI specificata, sono elencate tramite il campo iSliceIQ per i campioni demodulati.
 - d. **iMerge (0)** specifica se la PRI viene fusa con quella precedente.
 - e. **iHDStore (0)** abilita/disabilita la registrazione HD per la PRI corrente. Se abilitato, i dati che scorrono nella Slice IQ selezionata vengono anche registrati nell'HDD incorporato.
- 10. [MODULES]:** Questa sezione contiene l'elenco delle istanze dei moduli di elaborazione. Per ogni chiave di questo elenco, l'applicazione crea un'istanza del modulo e il contenuto del campo sName diventa il nome di tale istanza. Per ogni istanza del modulo, nel file di configurazione deve essere presente la "sezione dell'istanza del modulo di elaborazione": cioè una sezione con lo stesso nome dell'istanza, le cui chiavi devono specificare il tipo di modulo di elaborazione e tutti i suoi parametri
- 11. [IMAGE1]:** Questa sezione si occupa di configurare le impostazioni dell'immagine, come la soglia, la gamma dinamica, la finestra di visualizzazione, la fetta di dati da utilizzare e l'elemento da visualizzare. Questi parametri dovrebbero essere presenti o all'interno di una sezione del file di configurazione o all'interno di un file esterno.
- 12. [StdBMode.mod]:** Questa sezione definisce le impostazioni per il modulo Standard B-Mode.

- a. **ModuleName, SizeX, SizeY, VideoFilter:** Queste chiavi sono utilizzate per configurare le impostazioni del modulo, come il nome del modulo, le dimensioni dell'immagine, etc.

4.9 FILE DI STRATEGIA Tx/Rx (.ula)

Nel file di configurazione .cfg, nella sezione "SEQUENCER", sono specificati uno o più file con estensione .ula I file ".ula" permettono di settare le caratteristiche della sonda e le varie modalità di beamforming.

Vengono analizzate le principali sezioni:

1. **[GLOBAL]:** Questa sezione contiene le impostazioni globali per la configurazione.
 - a. **Name:** stringa che identifica il nome del pattern.
 - b. **NLines:** il numero di raggi/linee che compongono il modello. Nel caso di un'immagine BMode, questo numero intero corrisponde al numero di linee che compongono la larghezza dell'immagine. (nel nostro caso NLines = 192 che rappresenta il numero di elementi della sonda)
 - c. **ScanType:** Tipo di scansione:
 - i. "Linear" (0), in caso di scansioni lineari con sonde lineari
 - ii. "Phasing" (1), per scansioni angolari
 - iii. "Sector" (2), per scansioni settoriali con sonde convesse.
 - d. **ScanStep:** Determina la fase di scansione. La produzione di un'immagine (BMode) spesso richiede che la linea di scansione si muova e/o si sposti lungo gli elementi della sonda. Per le scansioni lineari e settoriali, questo parametro specifica di quanti elementi alla volta deve essere spostata la linea di scansione. Per le scansioni fasiche, determina di quanti gradi alla volta deve essere orientata la linea.
 - e. **YPhasing:** Determina la distanza tra la sonda e il pivot (punto centrale di rotazione). Questa chiave è necessaria solo per le scansioni di fase. Un valore positivo posiziona il pivot nel campo di scansione, mentre un valore negativo posiziona il pivot "prima" della sonda.
 - f. **LineOffset:** Imposta l'offset tra l'origine e le linee Tx e Rx definite. Se omesso, il valore predefinito di questo parametro è (-(Probe_Elements -1)/2) per una sonda lineare e 0 per una sonda convessa.
2. **[TXSETTINGS]:** Questa sezione contiene le impostazioni di trasmissione.
 - a. **NTX:** numero di elementi usati in trasmissione. Questo numero intero non deve mai superare il numero totale di elementi della sonda.
 - b. **TXFocus:** Coordinata X del punto di fuoco della trasmissione, espressa in millimetri [mm].
 - c. **TYFocus:** Coordinata Y del punto di fuoco della trasmissione, espressa in millimetri [mm].
 - d. **TXAngle:** Angolo di steering della trasmissione, espresso in gradi [°]
 - e. **TXApod:** Finestra di apodizzazione spaziale per la trasmissione: "Rect" (0), "Hanning" (1), "Hamming" (2), "Blackman" (3).
 - f. **TXFreq:** Frequenza degli impulsi di trasmissione, espressa in Hertz [Hz].
 - g. **TXNC:**
 - i. **iTXNC:** numero di cicli dell'impulso di trasmissione.

- ii. **iMaxTXNC:** numero massimo di cicli dell'impulso di trasmissione. Se questo campo non è specificato, il numero massimo di cicli è uguale al numero di cicli iTXNC.
 - h. **TXAmp:**
 - i. **fTXAmp:** ampiezza del segnale. Questo valore deve essere compreso tra 0 e 1.
 - ii. **fMaxTXAmp:** ampiezza massima del segnale. Se questo campo viene lasciato vuoto, l'ampiezza massima del segnale equivale all'ampiezza del segnale fTXAmp.
 - i. **TXWin:** Finestra di apodizzazione temporale per la trasmissione ad impulsi: "Rect" (0), "Hanning" (1), "Hamming" (2), "Blackman" (3)
- 3. [RXSETTINGS]:** Questa sezione contiene le impostazioni di ricezione.
- a. **NRX:** numero degli elementi usati in ricezione. Questo numero intero non deve mai superare il numero totale di elementi della sonda.
 - b. **RXPass:** Coordinata X del punto di passaggio, espressa in millimetri [mm]. Il punto di passaggio si trova sul percorso di ricezione.
 - c. **RYPass:** Coordinata Y del punto di passaggio, espressa in millimetri [mm]. Il punto di passaggio si trova lungo il percorso di ricezione.
 - d. **RYMin:** Profondità minima di interesse, espressa in millimetri [mm]. Nota: questo parametro deve essere impostato tenendo conto della profondità di messa a fuoco della trasmissione e della lunghezza del segnale. Più bassa è la profondità di messa a fuoco, più alto è il valore RYMin. Più lungo è il segnale, più alto è il valore RYMin.
 - e. **RYMax:** Profondità massima di interesse, espressa in millimetri [mm]. Nota: questo parametro influenza sul PRF (frequenza di ripetizioni di impulsi) massimo raggiungibile. Più alto è il valore di RYMax, più basso è il PRF.
 - f. **RXAngle:** Angolo di steering della ricezione, espresso in gradi [°].
 - g. **RXApod:** Finestra di apodizzazione per la ricezione: "Rect" (0), "Sinc" (1).
 - h. **RXApodFactor:** Fattore di proporzionalità dell'apodizzazione. Il valore predefinito è 1, un valore più alto apre l'apertura più lentamente.
 - i. **RXApodType:** Seleziona il tipo di apodizzazione della ricezione: "Dynamic" (0), "Static" (0).
- 4. [RXANALOG]:** Questa sezione contiene le impostazioni analogiche di ricezione.
- a. **Tgc:** Il controllo del guadagno temporizzato. Ha due fattori. Il primo *iA* è il valore iniziale del TGC espresso in dB. Il secondo parametro *fB* è il valore della pendenza del TGC espresso in dB per centimetro.
- 5. [RXELAB]:** Questa sezione contiene le impostazioni di elaborazione della ricezione.
- a. **DemodulationFrequency:** frequenza di demodulazione, espressa in Hertz [Hz]. A causa della sua implementazione, il demodulatore non può gestire tutte le frequenze, quindi l'applicazione cerca di utilizzare la frequenza più vicina a quella desiderata.
 - b. **Ncpm:** questi campi rappresentano il ritardo del primo, secondo, terzo e quarto filtro a pettine (o equivalentemente la lunghezza dei corrispondenti filtri a media mobile).
 - c. **NDec:** questi campi determinano i fattori di ri-campionamento applicati dopo il secondo, il terzo e il quarto filtro CIC (Cascaded Integrator-Comb)

4.10 SET-UP Sperimentale

Come esposto precedentemente, vi è la necessità di utilizzare un mezzo di accoppiamento tra la sonda e il palmo della mano poiché le onde ultrasoniche si riflettono fortemente quando incontrano l'aria. Per creare tale accoppiamento, si è utilizzata l'acqua come mezzo di trasmissione degli ultrasuoni. L'acqua ha una bassa attenuazione e una bassa impedenza acustica, che favoriscono la propagazione degli ultrasuoni e la formazione di immagini di buona qualità. Si preferisce usare l'acqua e non il gel, perché l'acquisizione viene fatta in modo automatico, e il gel usato da solo non va bene. Bisognerebbe usare dei gel pad specifici realizzati con materiale sintetico.

Il primo passo consiste nell' immergere sia la sonda che il palmo della mano in una vaschetta d'acqua. Per favorire il corretto posizionamento della mano sono state inserite, nella piastra, delle guide all'altezza delle dita, in modo da evitare acquisizioni con grandi differenze di posizione. Per aumentare la comodità e la stabilità, ci serviamo di una guida per il posizionamento del braccio che serve anche a istruire l'utente su come posizionarsi correttamente. Questa configurazione verrà mantenuta per tutta la durata del processo di acquisizione.

Tra la sonda e il centro della mano vi è una distanza di 20 mm, tale scelta scaturisce dal fatto che la tipologia di sonda da noi utilizzata ha una focalizzazione fissa tra 15-20 mm, motivo per cui è bene scegliere una distanza in questo intorno.

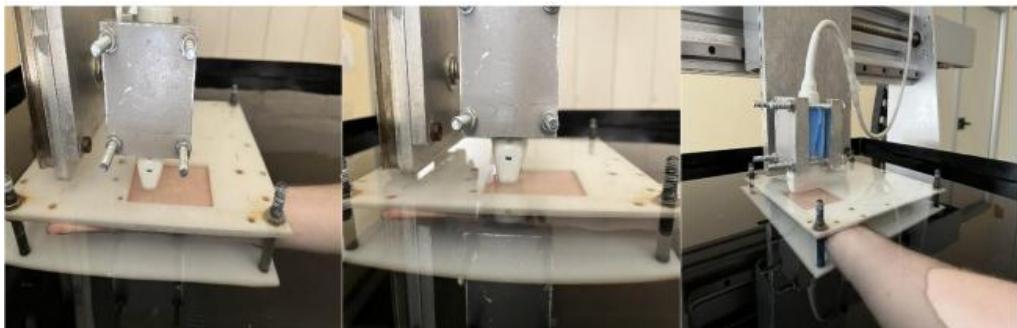


Figura 4.4 - Set up sperimentale

Prima di avviare il processo di acquisizione è necessario accertarsi che l'utente abbia posizionato correttamente la mano e che non vi sia formazione di bolle d'aria sul palmo durante il posizionamento, poiché potrebbero interferire con la qualità dei dati. È necessario verificare che l'area di interesse, roi, risulti chiaramente visibile attraverso una finestra presente nella piastra e che rimanga completamente immersa, così come la sonda, durante tutto il processo, in modo da ottenere dati di alta qualità.

La calibrazione del sistema è un passaggio fondamentale per ottenere acquisizioni accurate e coerenti. L'origine della sonda è tarata lungo la direzione z portandola prima a contatto con il centro del palmo fino a raggiungere la distanza desiderata tra la sonda e la mano. Questa procedura ha permesso di stabilire un punto di riferimento comune per tutte le acquisizioni, garantendo un corretto allineamento spaziale dei dati raccolti.

4.10.1 Processo di acquisizione del palmprint

Nel seguente paragrafo, verrà descritta l'intera procedura di acquisizione del volume. Questa procedura è composta da una serie di operazioni eseguite in ambiente Matlab, utilizzando un insieme di funzioni e procedure.

La prima fase consiste nell' eseguire lo script "**acquisisci.m**" che permette di settare:

- numero di acquisizioni da effettuare consecutivamente;
- il nome dell'utente che si sta sottponendo all'acquisizione;

- l'istante d'inizio dello slot di acquisizione: questo permette di poter acquisire lo stesso utente in più momenti diversi, mantenendo le acquisizioni nella stessa cartella, senza il rischio di sovrascritture indesiderate

All'interno di tale script viene creata l'istanza della classe pantografo attraverso la chiamata del file: "**pantografo_acqua**". Questo consente di configurare il movimento del pantografo

Nel ciclo for viene chiamata la function "**palmoIstanteN(n, nomeUtente)**" che prende in input l'istante dell'acquisizione e il nome dell'utente ed è qui che avviene l'acquisizione vera e propria. Lo script si compone di comandi standard che si occupano di:

- Attivare un collegamento tra Matlab e ULA-OP attraverso "**UOLink**"
- far ritornare il pantografo nell'origine in quanto una volta effettuata un'acquisizione il pantografo torna indietro senza acquisire
- freeze: al termine dell'acquisizione la finestra si freeza e in quell'istante viene salvata l'acquisizione

lo script "**configurazioniBMode**" permette di caricare il file di configurazione .cfg che a sua volta richiama il file .ula

5 ANALISI DEL DATABASE LUONGO – SCAVONE

In questo capitolo andiamo a illustrare il processo di analisi del database e di valutazione della qualità delle immagini acquisite. Per prima cosa è necessario prendere dei campioni dal database originale per poterli visualizzare tramite *Biometric data representation*. Una volta scelti i campioni è possibile esplorare il volume acquisito convertendo il file **.uob** generato dal sistema di imaging in un matrice 3D con estensione **.mat**

5.1 GENERAZIONE DELLA MATRICE VOLUMETRICA

Per generare la matrice volumetrica bisogna eseguire lo script **fromUOBtoMAT_singoloUtente.m**, questo converte le scansioni **.uob** di un singolo utente in matrici 3D **.mat**. Mentre se si vuole convertire un database con più utenti (cartella contenente le cartelle degli utenti, ad esempio "Database" all'interno della struttura sopra presentata) bisogna eseguire lo script **fromUOBtoMAT_dbCompleto.m**.

Dopo aver settato tutti i percorsi per l'acquisizione e il salvataggio dei dati la funzione effettua le operazioni di acquisizione dei dati che vedremo di seguito, ed infine verranno effettuate le operazioni di preprocessing. La struttura dello script per la generazione della matrice è la seguente:

```
fromUOBtoMAT_dbCompleto.m.

passaggio_UOB_BSCAN;
eliminaResidui;
convert2bmp;
preprocessing_bio;
```

5.1.1 Acquisizione dei dati

La prima fase consiste nell'acquisire i dati grezzi prodotti dalla scansione della sonda e processati dal sistema di imaging **UOB 64**.

Inizialmente, la prima fase consiste nella generazione di 300 immagini B-scan, che sono immagini B-mode acquisite durante lo spostamento della sonda lungo la direzione di elevazione, corrispondente alla direzione Y. In Matlab, questa operazione è eseguita tramite il file "**passaggio_UOB_BSCAN.m**", il quale prende in input un file con estensione "**.uob**" che rappresenta l'acquisizione 3D senza spaziatura nelle tre direzioni. All'interno di questa funzione, vengono definiti i parametri relativi al tipo di sonda, "**phys pitch**" ovvero il passo fisico della sonda e "**repr pitch**" che rappresenta la dimensione del passo tra gli spostamenti lungo la direzione di scansione laterale meccanica. Nel nostro caso, la sonda ESAOTE LA435 ha un "**phys pitch**" e un "**repr pitch**" pari a 0.2 mm. Successivamente vengono eseguite due procedure: "**eliminaResidui.m**," che rimuove l'acqua dai potenziali punti luminosi indesiderati, e "**convert2bmp.m**," la procedura effettiva per generare i B-scan.

Con questi dati è ora possibile ricostruire il volume acquisito.

Struttura:

```
passaggio_UOB_BSCAN;  
eliminaResidui;  
convert2bmp;
```

Per caricare i file salvati dall'ecografo bisogna eseguire il codice **fromUOBtoMAT_dbCompleto.m**, che caricherà il file **.uob** salvato precedentemente dall'ecografo. Lo script quindi che fa funzione di *main* prende in input il file **.uob** li converte in banda base e genera in output i 300 B-SCAN.

Lo script prima di generare i B-SCAN chiama la procedura **eliminaResidui** che ha lo scopo di pulire l'immagine dagli eventuali spot luminosi che si manifestano nei B-SCAN e di separare il palmo della mano dall'acqua che viene rappresentata di colore nero intenso.

Per capire come ricostruire i dati dobbiamo prima capire come vengono acquisiti e salvati i dati. Questo dipende da come sono stati settati i parametri di acquisizione del metodo Luongo-Scavone, in particolare dimX, dimY e dimZ.

- dimZ: regola il numero di punti sparati dal fascio ultrasonico lungo la direzione z, pari a 320.
- dimX: è pari al numero di elementi della sonda quindi ha un valore fisso di 192.
- dimY: è direttamente correlato al numero di immagini BSCAN acquisite, pari a 300.

La sonda fa 192 spari in sequenza, inizialmente spara lungo Z acquisendo 320 punti fino a completare il B-SCAN sparando 192 volte lungo X. Successivamente la sonda si muove lungo Y acquisendo un altro B-SCAN. Alla fine otterrò una matrice di dimensioni pari a $192 \times 300 \times 320$ salvata nel file **.uob**. Su matlab quindi andiamo a leggere i punti acquisiti e ricostruiamo la matrice di pixel 3D.

L'acqua quindi verrà eliminata dall'immagine dato che non porta contributo ai fini del riconoscimento e questo viene fatto impostando una soglia ed eliminando i pixel che la superano. In questo caso la soglia è stata impostata a -50, in output la procedura restituisce una matrice logica che separa il palmo dall'acqua. Successivamente viene applicata una procedura di filtraggio con un filtro mediano per rendere più omogenea l'immagine e smussare il segnale.

Infine i B-SCAN vengono generati dalla funzione `convert2bmp` che prende la matrice 3D ricostruita e ne estrae 300 B-SCAN.

Lo script `fromUOBtoMAT_singoloUtente.m` comprende al suo interno tutte le operazioni di **pre-processing** che partecipano alla creazione del volume finale da cui verranno estratti i template.

5.1.2 Pre-processing dei dati

In questa seconda fase andiamo ad effettuare un “crop” grafico dei dati di scansione. Lo script che andiamo ad analizzare è quindi `preprocessing_bio.m` che contiene varie chiamate ad altre funzioni.

Struttura:

```
preprocessing_bio.m  
  init_bio.m  
  probes_bio.m  
  depths_bio.m  
  data_bio.m  
  crop_bio.m  
  update_positions.m  
  interp_bio.m
```

Il primo script che viene eseguito è `init_bio.m` la quale setta i parametri che danno l'informazione sulla spaziatura nelle 3 dimensioni del volume acquisito. Quindi si procede a definire i parametri della sonda, in particolare la sonda utilizzata ha una spaziatura lungo x pari a 0.2mm (pitch) e la profondità di campo. Parametri settati delle funzioni `probes_bio.m` e `depths_bio.m`.

Passiamo ora allo script `data_bio.m` che raggruppa i 300 b-scan in una matrice 3D assegnando la spaziatura lungo X,Y e Z con i parametri settati precedentemente.

Successivamente con lo script `crop_bio.m` vengono tagliate le parti laterali dell'immagine che hanno una qualità inferiore questo perché sul bordo il numero di elementi della finestra diminuisce e la qualità non è ottimale. Selezioniamo nel volume la parte più pulita.

Infine viene eseguita la procedura `interp_bio.m` che ha lo scopo di eseguire un'interpolazione. Lungo l'asse Z la risoluzione dell'immagine è di 64 micron mentre lungo gli altri due assi è di 200 micron, quello che bisogna fare è rendere uniforme la spaziatura lungo i 3 assi.

Dopo l'interpolazione, si ottiene uno step di 46,2 micron (0.0462 mm), simile a quanto avviene per i vettori X e Z. Questo valore definisce la lunghezza in pixel legata alla profondità di scansione dove quindi ogni pixel avrà la dimensione di 0.0462 mm. Al termine di queste operazioni, si ottiene una matrice 3D delle dimensioni $835 \times 820 \times 320$ voxel, che corrisponde a un volume di $38.4 \times 38 \times 15$ mm³ e che è acquisito in circa 5 secondi.

Alla fine di questa fase abbiamo come risultato un file `.mat`

5.1.3 Rappresentazione

In questa fase andiamo a ricostruire il volume acquisito effettuando i render. Ci sono 2 tipi di render che possono essere effettuati: render 3D, render 2D

5.1.3.1 Rappresentazione 3D

Permette di avere una visione d'insieme del volume acquisito.

Struttura:

render3d_bio.m

vol3dd.m

La funzione **render3d_bio.m** permette di ricostruire il volume dal file **.mat**. Il risultato è il seguente:

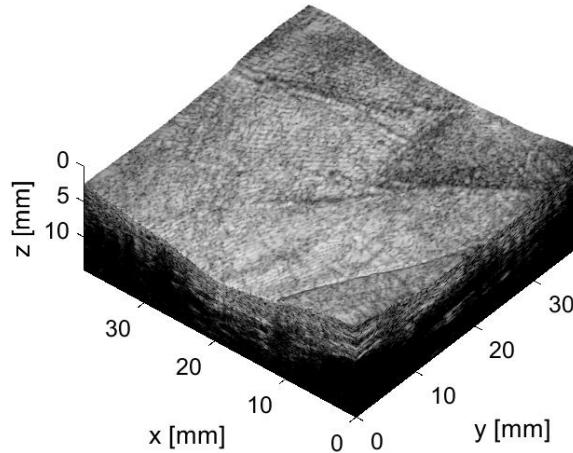


Figura 5.1- Immagine volumetrica del palmo della mano

5.1.3.2 Rappresentazione 2D

Questo modulo consente di rappresentare i dati mediante tre sezioni ortogonali tra loro (X-BSCAN, Y-BSCAN e Z-CSCAN). Una semplice interfaccia consente di modificare le coordinate di rappresentazione in tempo reale.

Struttura:

render2d_bio.m

measure_2drender.m

update_2drender.m

update_pos2drenderer.m

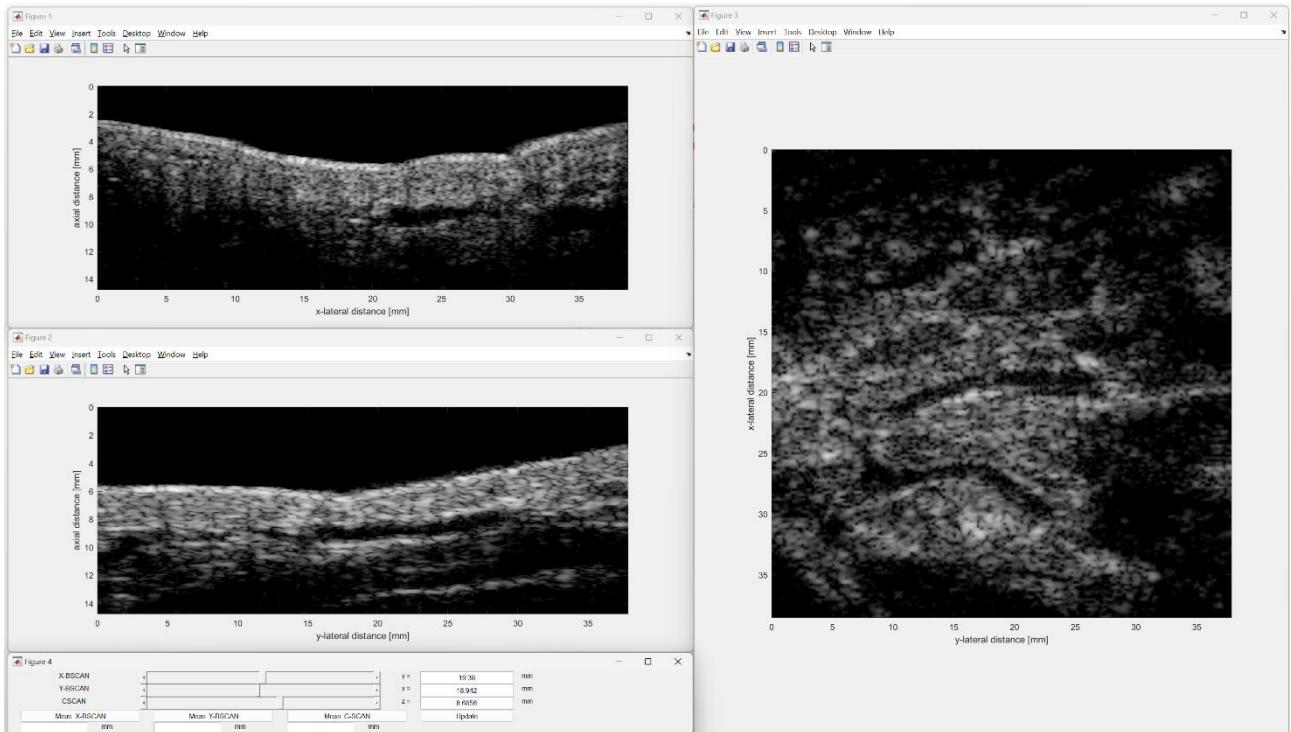


Figura 5.2 - Esempio rappresentazione 2D

Il vantaggio di questa visualizzazione è che possiamo esplorare il volume e risulta molto utile per la ricerca delle vene. Inoltre può essere utilizzato anche come strumento di verifica. Una volta estratti pattern venosi è opportuno andarli a proiettare su questi piani per verificare la correttezza delle features estratte.

5.1.4 Visualizzazione pattern

In questa ultima fase andiamo a renderizzare il pattern del palmo acquisito, visualizzando i pattern alle diverse profondità.

Struttura:

```
renderpattern_bio.m
update_interp.m
update_surfdetect.m
update_pattern.m
```

Con lo script `renderpattern_bio.m` mandiamo a visualizzare la proiezione della superficie del palmo sul piano fatta a varie profondità, da non confondere con il C-SCAN. Anche in questo caso possiamo esplorare il render a varie profondità tramite una gui. Possiamo notare come nelle prime immagini i tratti del palmo ancora non si contraddistinguono dato che sulla superficie persiste ancora il rumore dell'acqua. Avanzando con il cursore e scendendo di qualche pixel in profondità si iniziano a scorgere le features fino a scomparire completamente verso il fondo.

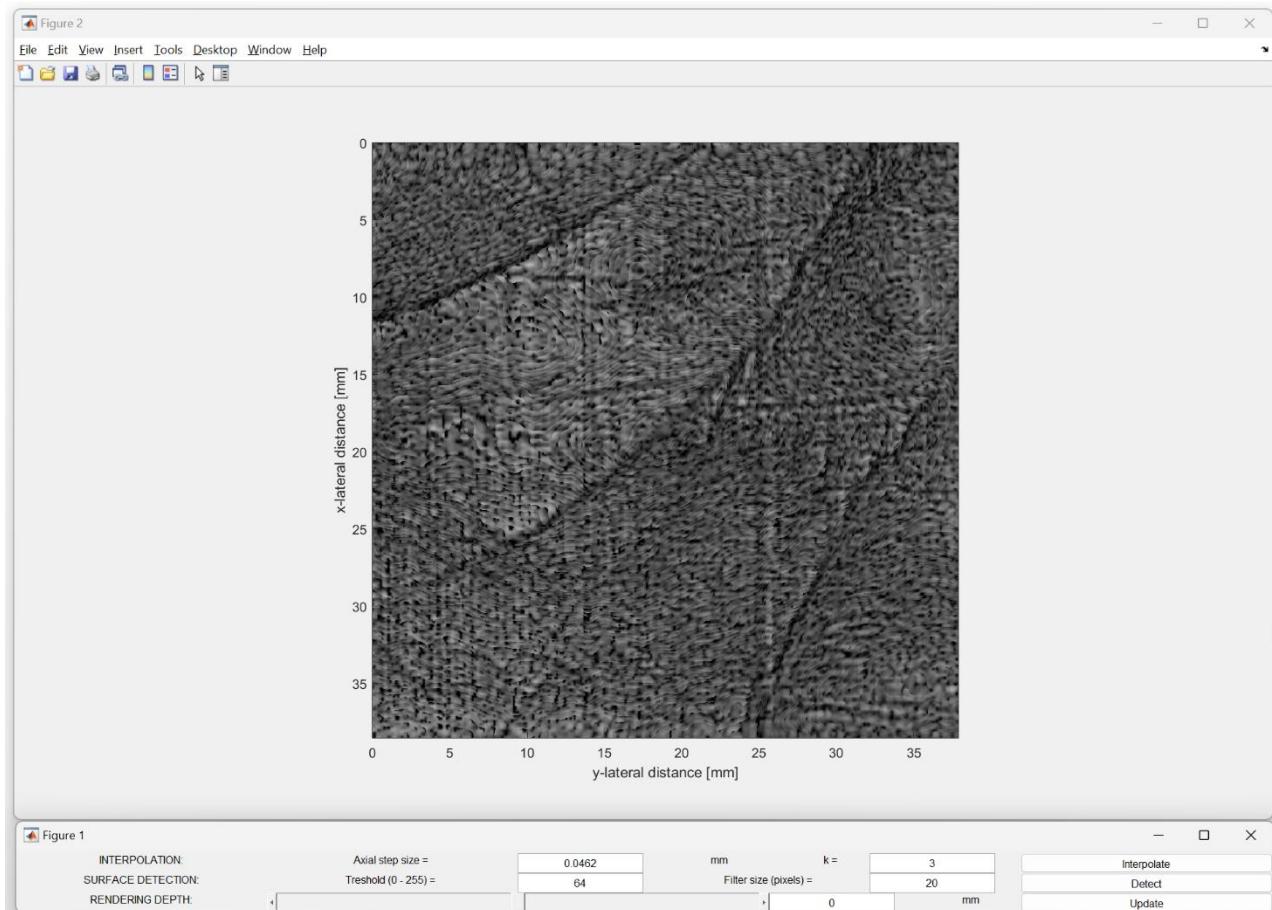


Figura 5.3 - Visualizzazione pattern a varie profondità

6 ESTRAZIONE PATTERN VENOSO

Analizzati i data set a nostra disposizione, nel seguente capitolo verranno illustrate le varie tecniche adoperate per estrarre il pattern venoso palmare a partire da immagini ad ultrasuoni, seguendo quanto effettuato dalla dott.ssa Vizzuso.

Prima di procedere è necessario precisare che le immagini volumetriche utilizzate sono state acquisite per estrarre il palm print, pertanto con questo studio si vuole innanzitutto dimostrare se da esse è possibile estrarre il pattern venoso palmare.

La trattazione di questo capitolo è basata su [4]

6.1 FEATURE EXTRACTION

La fase di **features extraction** è il processo in cui, a partire dall'immagine volumetrica acquisita nella fase precedente, viene creato un template che contiene esclusivamente le informazioni numeriche necessarie per caratterizzare il pattern venoso.

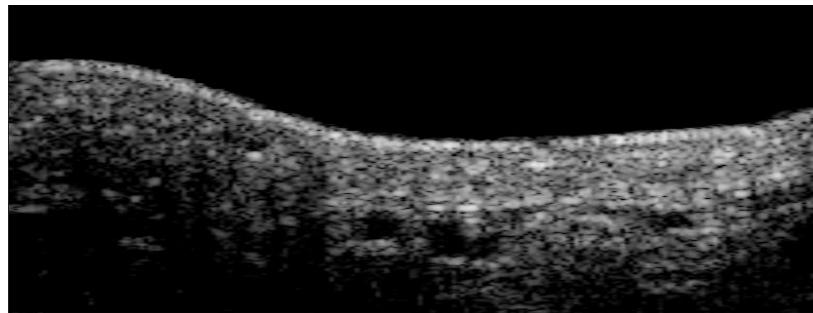
La **features extraction** si articola in varie fasi:

1. Estrazione dei BSCAN
2. Filtraggio (opzionale)
3. Binarizzazione

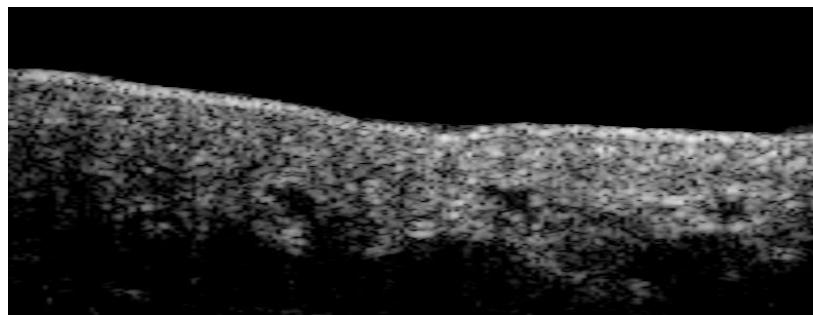
4. Ricerca delle componenti connesse
5. Estrazione del pattern venoso.

6.1.1 Estrazione dei Bscan

A partire dall'immagine volumetrica delle dimensioni $835 \times 820 \times 320$ voxel, che corrisponde a un volume di $38.4 \times 38 \times 15 \text{ mm}^3$, di cui si è parlato nel paragrafo precedente, è possibile estrarre 820 immagini B-SCAN in scala di grigi.



a)



b)

Figura 6.1 - (a) Primo BSCAN estratto; (b) Ultimo BSCAN estratto dalla stessa acquisizione volumetrica.

6.1.2 Filtraggio e binarizzazione

Prima di procedere alla fase di binarizzazione si verifica se l'utilizzo di un filtro può migliorare o meno le estrazioni del pattern venoso. Tutto ciò è possibile inserendo una fase di pre-processing realizzata con il filtro SRAD.

6.1.2.1 Filtro SRAD

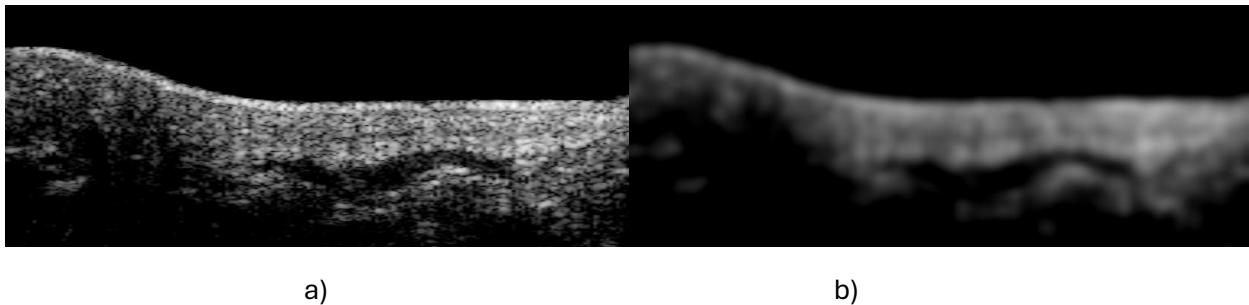
Il filtro SRAD, *Speckle Reducing Anisotropic Diffusion*, consente di ridurre lo speckle noise, un rumore additivo tipico delle immagini ad ultrasuoni scaturito dall'interferenza tra due o più onde che formano punti scuri e luminosi compromettendo quindi la qualità dell'immagine.

Il funzionamento dell'algoritmo è garantito dalla presenza di una finestra che si fa scorrere sull'intera immagine in modo tale che, di volta in volta, sia centrata nel pixel considerato.

Si settano:

- Dimensione della finestra
- Numero di iterazioni
- Passo di campionamento

Si osserva come una finestra eccessivamente ridotta con un numero di iterazioni e passo di campionamento bassi, non permetteranno di ridurre lo speckle noise (Figura 6.2 a). Al contrario, se le dimensioni di tali parametri sono troppo elevate, si verificherà una levigatura eccessiva, compromettendo i dettagli specifici dell'immagine infatti risulterà molto più sfocata (Figura 6.2 b).



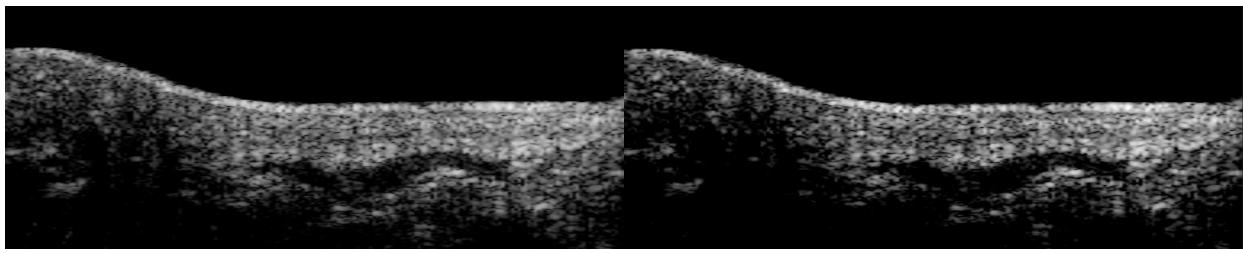
a)

b)

Figura 6.2 - a) finestra di dimensione 1 x 1 pixels, numero di iterazioni pari a 10, passo di dimensione 0.01.

b) finestra pari a 100 x 100 pixels, numero di iterazioni pari a 100, passo di dimensione pari a 1

Un buon compromesso si raggiunge utilizzando una finestra di dimensione 50 x 50 pixels, con numero di iterazioni impiegato pari a 40 e passo di campionamento pari a 0.1. Osservando le immagini sottostante si nota come l'utilizzo del filtro esalta le vene infatti appaiono molto più scure e con meno pixel bianchi rispetto all'immagine B-SCAN di partenza



a)

b)

Figura 6.3 - a) B-SCAN di partenza, b) B-SCAN con filtro usato.

6.1.3 Binarizzazione

La fase successiva prevede l'uso della binarizzazione. Si tratta di un processo che consente di convertire un'immagine in scala di grigi in un'immagine binaria (in bianco e nero). Per farlo è necessario stabilire una soglia e classificare i pixel dell'immagine impostando sul nero i pixel con un livello di grigio inferiore a tale soglia e sul bianco i pixel che, invece, presentano un livello di grigio superiore.

La scelta della soglia può avvenire in due modi:

- Utilizzare un valore globale, che risulta essere sempre lo stesso a prescindere dall'immagine che si sta esaminando;
- Selezionare una soglia adattiva che varia a seconda delle caratteristiche dell'immagine su cui è applicata.

Vengono testate 2 tipologie di binarizzazione con soglia adattiva differenti. il primo prevede l'utilizzo della media, il secondo utilizza la soglia di Ridler.

6.1.3.1 Binarizzazione per mezzo della media

Il primo tipo di binarizzazione applicata è la binarizzazione per mezzo della media. Essa prevede che per ogni B-SCAN venga calcolato il livello di grigio considerando la media dei pixel contenuti. Questo valore, moltiplicato per un fattore di scala, determina la soglia utilizzata nell'algoritmo.

La maschera viene applicata tenendo conto che:

- Il pixel, che ha un livello di grigio inferiore alla soglia, viene classificato come nero ed assume valore 0;
- In caso contrario è catalogato come bianco e presenta valore 1.

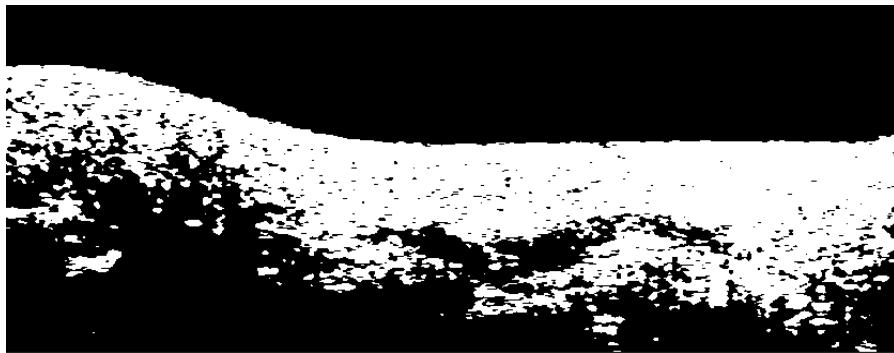


Figura 6.4 - Risultato della binarizzazione con media

6.1.3.2 Binarizzazione con soglia di Ridler

La binarizzazione con soglia di Ridler - Calvard è un metodo iterativo che può essere considerato un raffinamento della binarizzazione tramite media.

L'algoritmo alla base di tale binarizzazione si articola in vari punti:

1. Per ciascun BSCAN viene definito l'istogramma tramite `[counts,N]=imhist(I)`; dove `I` rappresenta l'immagine, `N` sono i valori assunti da ciascun pixel, compresi tra [0, 255] mentre l'insieme `counts` contiene le frequenze con cui ogni `N` si manifesta.
2. Determinare la soglia iniziale T_0 a partire dall'intensità media dell'immagine. Questo valore viene calcolato applicando la media tra i valori assunti dall'istogramma e le frequenze con cui si manifestano.
3. Tutti i pixel al di sotto della soglia T_0 vengono inseriti in un insieme A, i restanti in un insieme B.
4. Per ciascun insieme A e B vengono calcolate rispettivamente le soglie indicate come T_A e T_B calcolando la media dei pixel contenuti in ciascun insieme.
5. La nuova soglia di riferimento risulta $T_1 = (T_A + T_B)/2$.

A questo punto la soglia di riferimento è T_1 . L'algoritmo termina se T_1 risulta identica a T_0 altrimenti si procede in maniera iterativa eseguendo i passi 3, 4, 5 fin quando non si ottengono due soglie uguali.

Come nel caso della binarizzazione per mezzo della media, la soglia che andrà ad essere utilizzata sarà determinata da un fattore di scala opportunamente scelto. Una volta definita la soglia, si procede analogamente alla binarizzazione con media in modo da convertire i BSCAN da immagini in scala di grigi in immagini binarie (in bianco e nero).

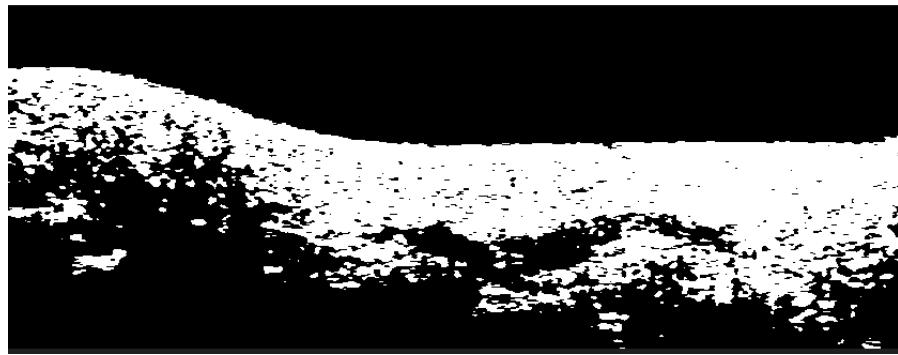


Figura 6.5 - Risultato della binarizzazione con soglia di Ridler

6.1.4 Ricerca delle componenti connesse

L'immagine logica viene esaminata per ricercare le coordinate dei pixels appartenenti alle regioni connesse. Affinché la ricerca vada a buon fine è necessario che i pixel delle regioni connesse siano impostati ad 1. È necessario calcolare per tutte le immagini logiche le corrispettive immagini complementari.

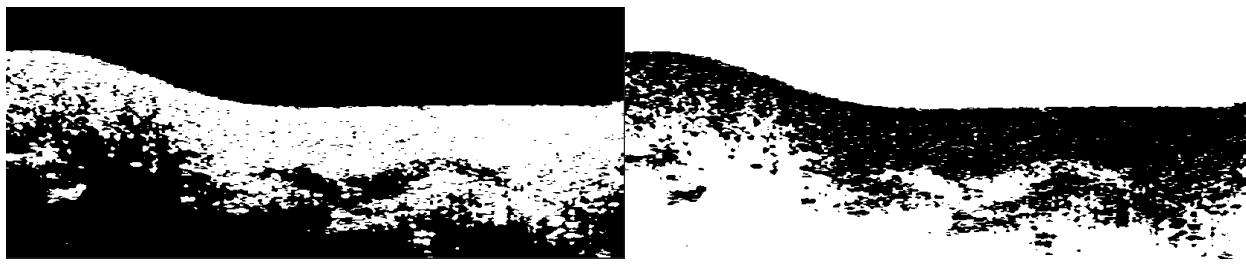


Figura 6.6 -(a) Immagine binaria; (b) Immagine complementare corrispondente.

L'algoritmo, utilizzato per la ricerca delle componenti connesse, si articola in varie fasi:

1. CASO A: Si consideri la prima immagine complementare. Su di essa viene applicata un'operazione di cleaning che elimina le componenti connesse aventi un'area maggiore di 1600 pixels e minori di 129 pixels.
2. CASO B: Se non viene individuata alcuna componente连nessa, si amplia il range di ricerca delle componenti connesse: si considerano le componenti connesse la cui area si estende da 50 pixels fino ad un massimo 1500.
3. Per i restanti BSCAN, la procedura di selezione delle componenti connesse è la medesima: vengono selezionate quelle che presentano un'area compresa tra 50 e 1500 pixels.

Le componenti connesse sono regioni bianche circondate da pixel neri. Per ciascuna di esse vengono individuati i centri di massa, o semplicemente centroidi, fondamentali per ricostruire il pattern venoso. Il centroide, infatti, viene identificato da due coordinate: la sua posizione lungo l'asse x e lungo l'asse z in modo da poter ricostruire la vena sia in modalità bidimensionale sia tridimensionale.

6.1.5 Estrazione dei pattern venosi

Una volta individuate tutte le componenti connesse ed i relativi centroidi, occorre stabilire la posizione del centroide di riferimento che varia in base al caso utilizzato per la scelta delle componenti connesse infatti:

- Nel caso A, il centroide di riferimento viene scelto come punto medio della sottomatrice avente una profondità compresa nell’intervallo [6,10.9] mm ed asse x che varia da 0 a 18 mm.
- Nel caso B, il centroide di riferimento rappresenta il punto medio della sottomatrice con asse x di partenza ed avente una profondità che varia da 6 a 10.9 mm.

A questo punto si può procedere con l’algoritmo di ricostruzione del pattern venoso articolato come segue:

- Per la prima immagine, viene calcolata la distanza euclidea tra il centroide di ciascuna componente connessa ed il centroide di riferimento. La componente connessa, avente minore distanza con il centroide di riferimento, rappresenta il punto di partenza per l’estrazione della vena.
- Per ogni immagine successiva viene individuato il centroide che presenta la minore distanza euclidea rispetto al centroide individuato nell’immagine precedente, a patto che il nuovo non disti più di 15 pixels lungo l’asse x e 12 pixels lungo l’asse z rispetto al precedente.

Per estrarre il pattern venoso, utilizzare il file “*automatico.m*”. Si eseguono le seguenti operazioni:

1. FASE DI CONFIGURAZIONE:

- a. È preferibile scaricare la cartella ‘Codice estrazione’ che contiene i file .m per evitare problemi di sincronizzazione;
- b. Specificare la posizione della cartella scaricata nella variabile **base_path**;
- c. Specificare la posizione della cartella contenente i file “.mat” in **dataset_path**.

Inserire il numero di core fisici della macchina su cui viene eseguito il codice tramite la variabile **num_core**.

TIPI DI STUDIO POSSIBILI:

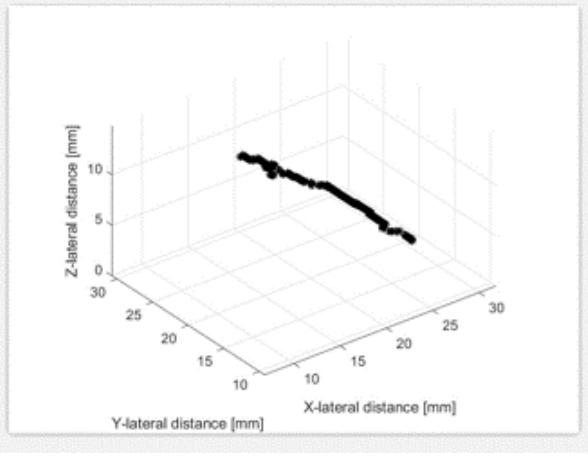
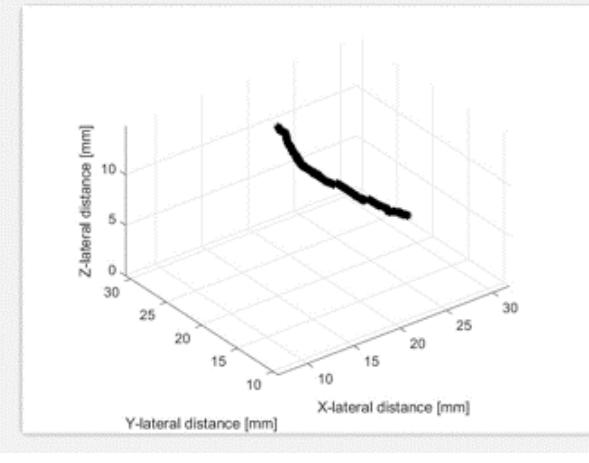
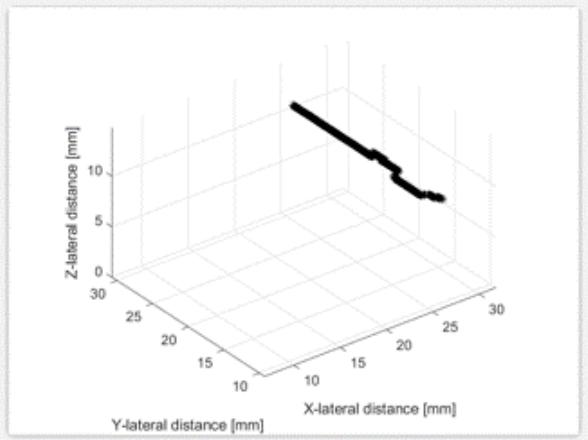
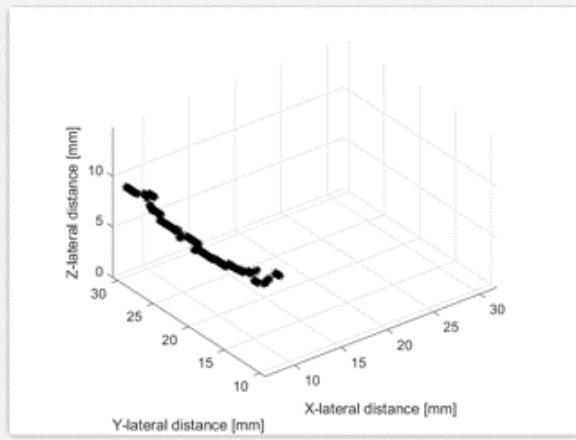
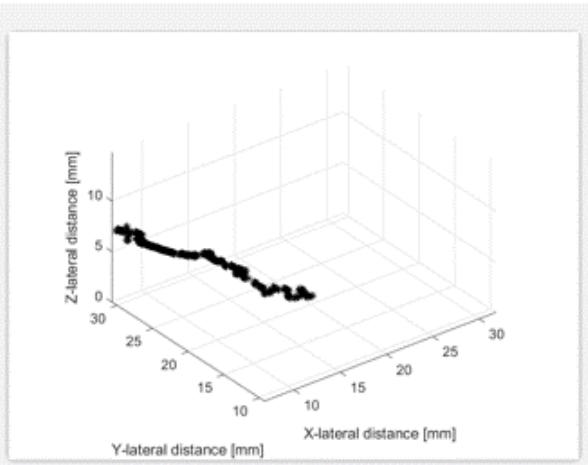
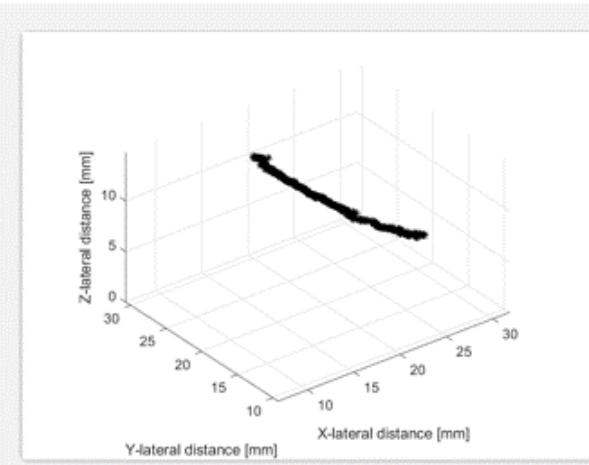
- **filtered** stabilisce se lo script utilizza un filtro in pre-processing: 0 nessun filtro, 1 il filtro SRAD.
- **Tipo Binarizzazione** consente di scegliere il tipo di binarizzazione da usare: 0 corrisponde alla binarizzazione con media mentre 1 con soglia di Ridler.

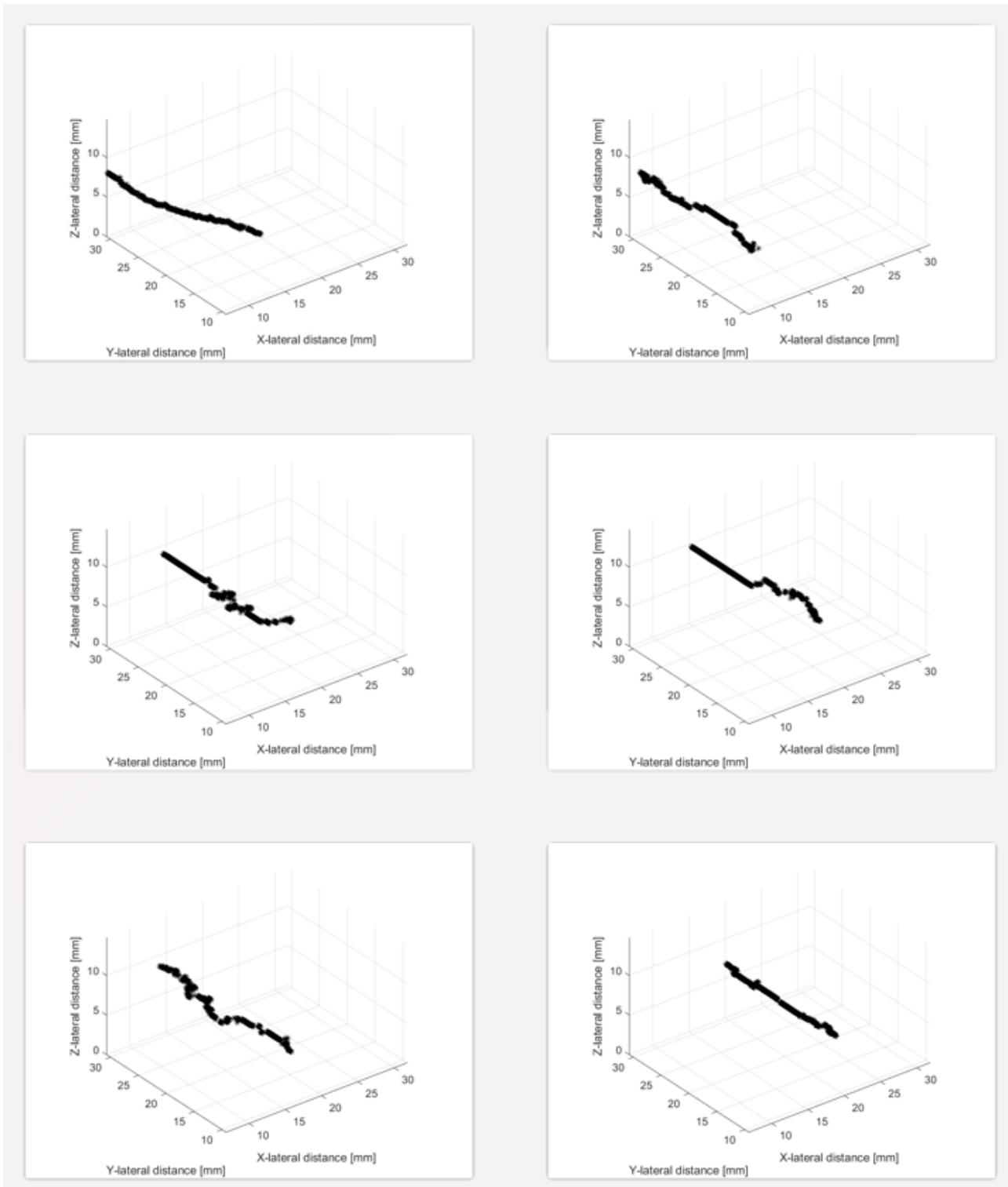
Una volta effettuate queste modifiche è possibile eseguire lo script, al termine del quale si generano 9 directory posizionate nel percorso specificato. Nel dettaglio si ha:

- **b-scan estratti:** contiene le 820 immagini B-SCAN in scala di grigi;
- **srad filtered b-scan:** contiene le 820 immagini B-SCAN filtrate con filtro SRAD. Tale cartella sarà vuota se si setta la variabile filtered pari a 0;
- **mask b-scan:** contiene le 820 immagini B-SCAN convertite in immagini binarie (fase di binarizzazione);
- **immagini_complementari:** contiene le 820 immagini complementari;
- **regioniconnesse b-scan:** contiene le 820 immagini che rappresentano le componenti connesse;
- **grafici2D e grafici 3D:** conterranno una proiezione del pattern venoso sul piano XY e sul piano XYZ;
- **template2D e template3D:** contengono le informazioni numeriche necessarie per caratterizzare il pattern venoso sia 2D che 3D;

6.2 TEMPLATE RISULTANTI

Di seguito riportiamo alcuni template che si ricavano dall’applicazione del metodo di estrazione Vizzuso al database Luongo-Scavone





6.2.1 Considerazioni sui template estratti

Dati i template estratti, è evidente che dal database in questione è possibile identificare e estrarre le vene palmari, confermando la validità dell'approccio iniziale. Tuttavia, l'applicazione del metodo Vizzuso ha messo in luce due limitazioni significative: da un lato, l'estrazione si concentra su una singola vena e non dell'intero pattern venoso; dall'altro, risultando unica, essa spesso condivide caratteristiche simili con le vene di altri utenti, compromettendo l'affidabilità del matching. Per superare queste limitazioni, abbiamo deciso di sviluppare un nuovo metodo di feature extraction, con l'obiettivo di migliorare la precisione e l'affidabilità del riconoscimento biometrico delle vene palmari.

7 IL METODO CCM

7.1 INTRODUZIONE

Il metodo di *Feature Extraction* che è stato sviluppato e che verrà proposto in questo capitolo, denominato ‘**metodo CCM**’ (dai suoi creatori Capece, Caporale, Manfreda), risulta essere innovativo nell’estrazione del pattern venoso del palmo di una mano, discostandosi notevolmente dall’approccio del sistema delineato dal ‘*metodo Vizzuso*’, sottopostoci all’attenzione dal prof. Iula, in termini di obiettivi prefissati e tecniche adottate.

Mentre il ‘*metodo Vizzuso*’ propone l’estrazione di una singola vena palmare, il ‘**metodo CCM**’ presenta l’estrazione dell’intero pattern venoso palmare, con le vene mantenute nella loro naturale struttura morfologica.

La procedura proposta basa il suo funzionamento sull’adozione di tecniche di *analisi dei dati* al fine di conservare solo i voxel del volume associati a strutture venose. A differenza di altri metodi non si va a ricostruire l’andamento della vena bensì si isolano le vene da quello che non è di interesse, denominato *rumore o sporco*.

Nello specifico l’obiettivo del metodo è quello di adattare l’estrazione dei template per tutti gli utenti, ottenendo risultati ripetibili che si regolano alle caratteristiche univoche di ogni individuo.

7.1.1 Features

Per raggiungere l’obiettivo prefissato è stato necessario creare un metodo che facendo uso di tecniche di analisi dei dati possa ricavare delle *features* che siano in grado di caratterizzare al meglio le strutture venose di ogni utente. Queste *features* devono essere univoche per ogni utente al fine di generalizzare il più possibile il procedimento e quindi rendere l’estrazione adattabile a vari tipi di utenti e acquisizioni. Riassumendo questo metodo permette di adattarsi al maggior numero possibile di utenti calcolando in modo automatico delle *features* che permettano di estrarre il template nel modo più chiaro possibile conservando le diverse strutture venose presenti nel palmo della mano.

Le principali features ingegnerizzate e che permettono al metodo di raggiungere un’estrazione efficace sono:

1. **Fine binarizzazione**
2. **Distanza vene-palmo**
3. **Soglia iniziale e finale**

Ognuna di queste permette di aggiungere un tassello in più nel processo di isolamento del pattern venoso e quindi di guidare il metodo alla formazione di un template quanto più realistico possibile. Ognuna delle features nominate verrà illustrata nella sua interezza attraverso i vari capitoli di questa relazione.

7.1.2 Assunzioni

Per raggiungere l’obiettivo sopra preposto è stato necessario fare delle assunzioni in modo tale da specializzare il metodo verso un’estrazione corretta sotto delineate ipotesi. Le assunzioni vengono riportate nei capitoli ma in questa introduzione viene dato un cenno su quelli che sono gli ambiti di funzionamento del modello. Il modello assume che:

1. La vena principale dell'acquisizione si trova alla profondità massima tra tutte le vene ed ha dei valori di intensità molto scuri tendenti a 0.
2. I primi voxel con intensità prossima a 0 che si osservano scansionando il volume nella direzione Z a partire dal palmo della mano e dirigendosi verso il fondo rappresentano delle vene.
3. Le vene sottili sono costituite da valori di intensità maggiori rispetto a quelle massicce profonde e si trovano più in superficie.

7.1.2.1 Perché limitarsi a queste assunzioni?

Come detto in precedenza lo scopo del metodo è quello di essere più generale possibile e quindi estrarre correttamente il più grande numero di template del database. Queste assunzioni permettono di percorrere una strada che porta a estrarre dei template veritieri che cadono sotto queste assunzioni. A seguito di una prima fase di analisi del database Luongo-Scavone si è visto come la stragrande maggioranza degli utenti ricade sotto queste ipotesi. Per cui adattare l'algoritmo a analizzare i dati in questo formato consente di estrarre numerosi template corretti a discapito di pochi risultati deludenti. Difatti ogni metodo consiste nell'adottare un compromesso per far sì che i ragionamenti fatti vadano bene in specifici casi e meno in altri, conseguentemente nel processo di sviluppo si è visto che queste assunzioni portavano i benefici migliori.

7.1.3 Premessa

Il metodo proposto è molto complicato, consiste di circa 60 script e svariate iterazioni lungo le 3 dimensioni del volume che portano i tempi di calcolo ad aumentare vertiginosamente. Per questo è stato necessario salvare gli step intermedi del processo in modo tale da rendere il processo di esecuzione più efficiente. Si è fatto largo uso di calcolo parallelo e di esplorazione 3D tramite la visualizzazione volshow di matlab. Nella seguente tesina si cercherà di trasmettere i ragionamenti che si celano dietro certe determinate scelte per mostrare come l'evidenza guidi il processo. Per cui non ci si soffermerà molto sui dettagli tecnici del codice sorgente, a questo scopo si rimanda il lettore alla visione degli script MATLAB.

7.1.4 Database utilizzato

Il metodo discussso è stato sviluppato su un sotto-database derivato dal database completo di Luongo-Scavone, che conta un totale di **73** utenti. Durante una prima fase di analisi delle acquisizioni, si è scelto di catalogare gli utenti in tre classi, utilizzando il metodo BDR nelle tre sezioni ortogonali X-BSCAN, Y-BSCAN e Z-CSCAN:

- S = strong, le vene si riconoscono chiaramente.
- L = low, le vene si riconoscono in maniera limitata.
- N = none, non si riesce a individuare delle vene.

Per ottimizzare il processo e ridurre i tempi di elaborazione, si è deciso di concentrare l'analisi sugli utenti catalogati come **S**, che rappresentano **43** utenti. Ogni utente contiene 20 acquisizioni, e, escludendo alcuni casi di acquisizioni corrotte (file .mat non leggibili), sono stati generati circa **800** template. La scelta di limitarsi agli utenti della classe S deriva dalla consapevolezza che la generazione dei template è un processo computazionalmente oneroso e richiede tempi significativi. Utilizzando questo approccio mirato, è stato possibile focalizzare le risorse computazionali sugli utenti con dati di qualità sufficiente a garantire risultati significativi, riducendo al contempo il carico di elaborazione.

7.1.5 Codice realizzato

Il codice realizzato si trova nella cartella **metodoCCM** e si suddivide in tre sottocartelle.

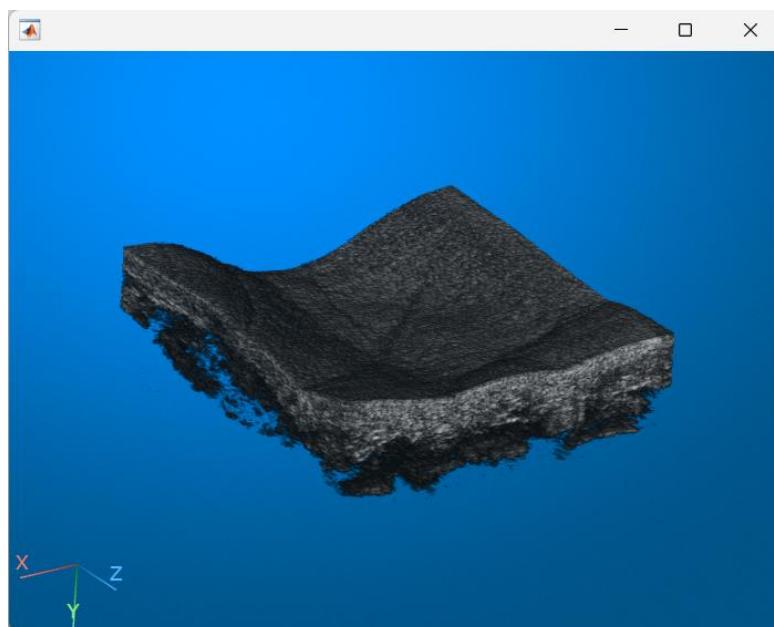
- **estrazioneTemplate**
 - Contiene il codice relativo all'estrazione dei template. Per estrarre un singolo template bisogna eseguire lo script **estraiTemplate**. Per estrarre i template di tutto il database o di una serie di utenti selezionati bisogna eseguire lo script **main**.
 - I risultati dei vari step del processo di estrazione vengono salvati nella cartella **metodoCCM/estrazioneTemplate/stepIntermedi**
 - Il template finale per ogni acquisizione di ogni utente si trova nel percorso **metodoCCM/estrazioneTemplate/stepIntermedi/utente/acquisizione/volAff.mat**
- **addestramento**
 - Contiene i codici relativi alla creazione del dataset di addestramento e al training del modello di classificazione binaria.
- **matching**
 - Contiene i codici per effettuare il matching. Per eseguire i confronti bisogna avviare lo script **identificazioneMatching3D**

7.2 CONFRONTO TRA VOLSHOW E BDR

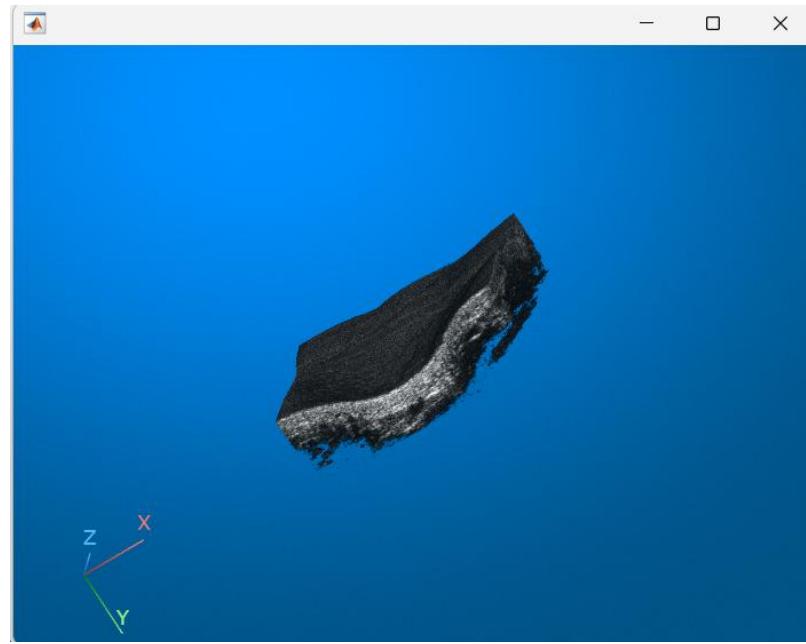
È utile ricordare che il codice di ‘*Biometric Data Representation*’ permette di ricostruire il volume palmare acquisito e contenuto nei file .mat attraverso due rappresentazioni; la rappresentazione 3D permette di avere una visione d’insieme e statica del volume acquisito mentre la rappresentazione 2D consente di visualizzare i dati mediante tre sezioni ortogonalni tra loro (X-BSCAN, Y-BSCAN e Z-CSCAN), avallandosi di una semplice interfaccia per modificare le coordinate di rappresentazione con conseguente possibilità di visionare le tre sezioni ispezionate in tempo reale.

Per il nostro studio, ci siamo serviti di un metodo nativo di matlab per tener conto della visualizzazione dei volumi palmari degli utenti: ‘**volshow()**’.

Attraverso il metodo ‘*volshow*’ è possibile ruotare, ingrandire e rimpicciolire il pattern 3D del palmo della mano, superando i limiti di staticità della rappresentazione 3D di ‘*Biometric Data Representation*’.



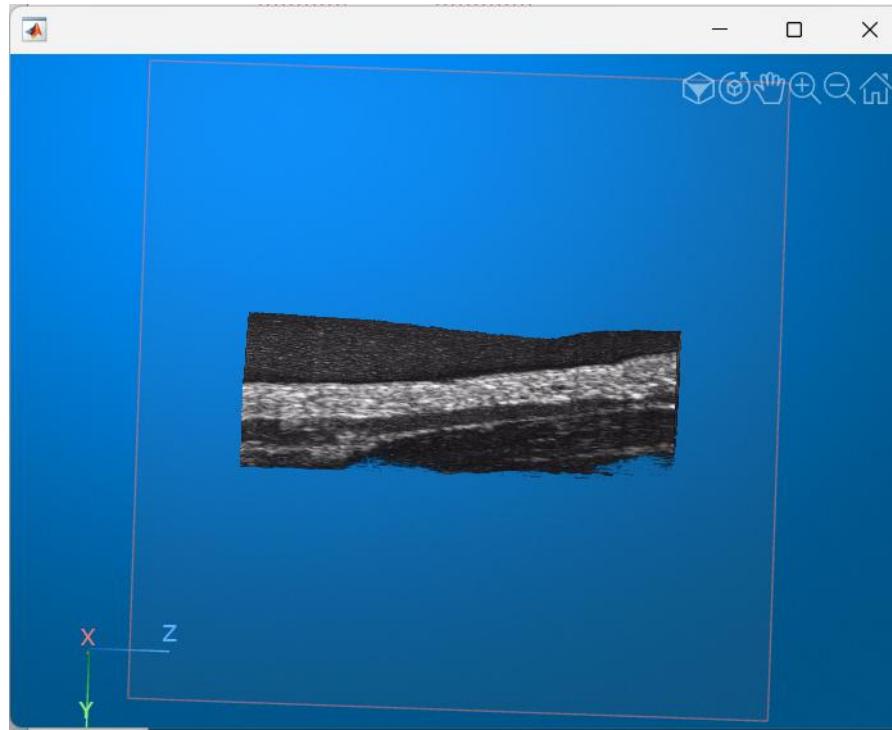
a)



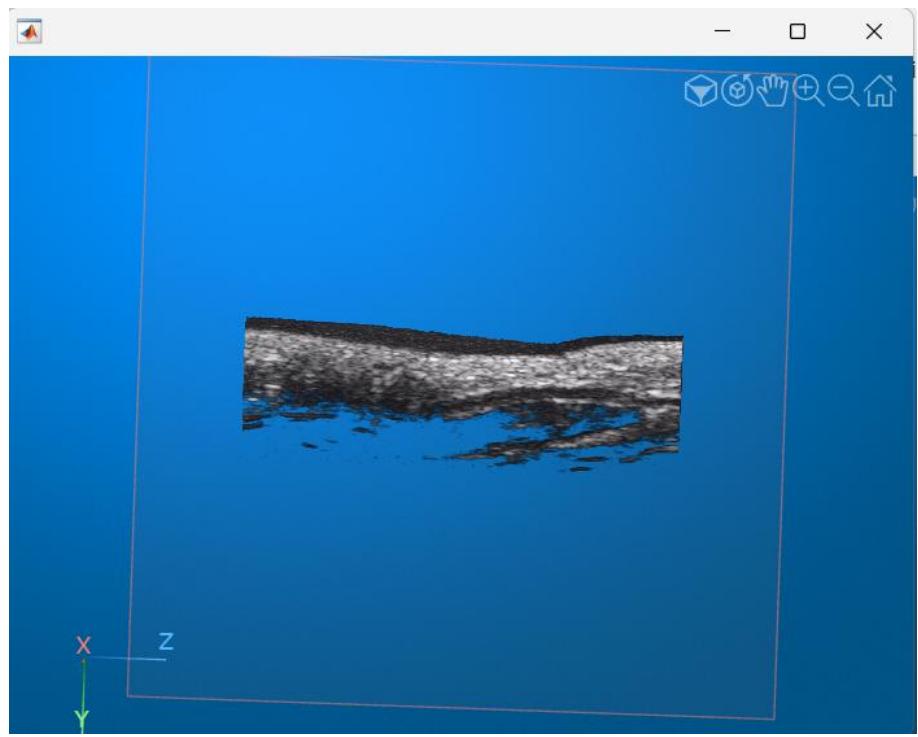
b)

Figura 7.1 (a) immagine volumetrica con volshow; (b) immagine volumetrica con volshow ruotata e rimpicciolita.

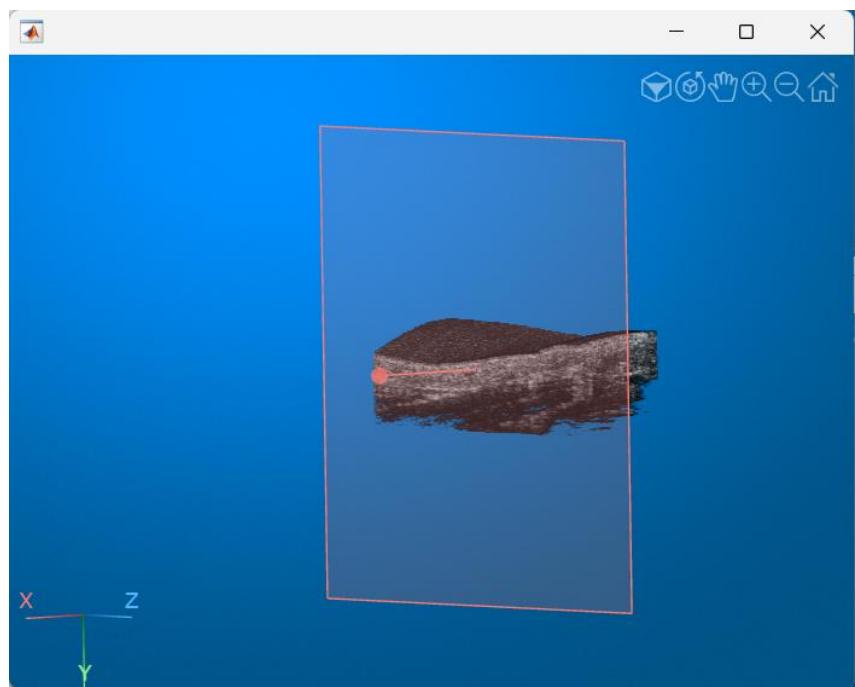
Il metodo ‘volshow’ offre la possibilità di esplorare interattivamente le sezioni X-BSCAN, Y-BSCAN e Z-CSCAN in profondità attraverso un piano di ritaglio:



a)

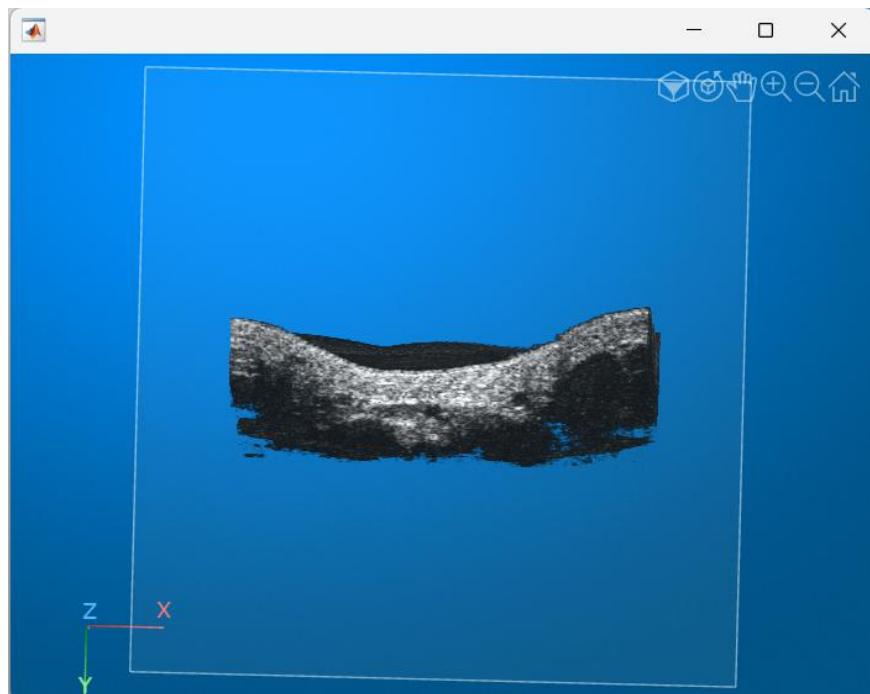


b)

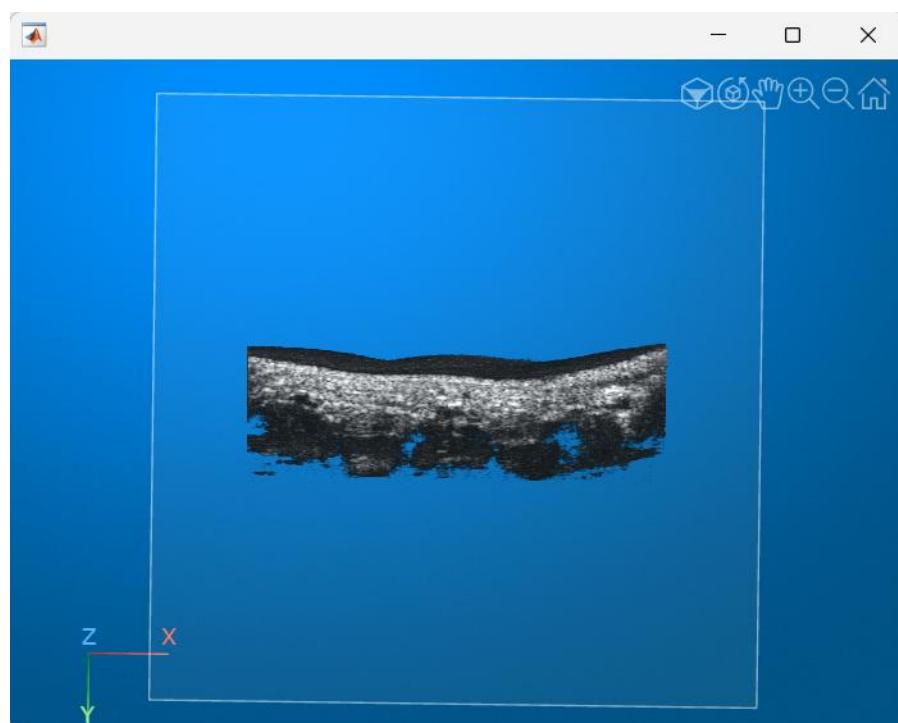


c)

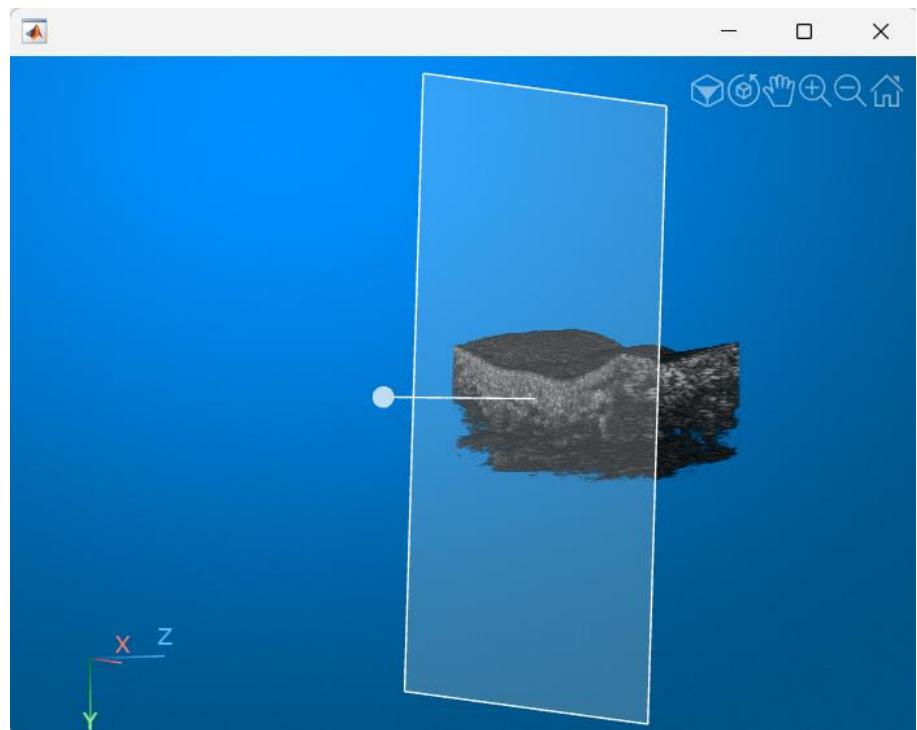
Figura 7.2 -(a), (b). Esempi di Y-BSCAN interattivi con piano di ritaglio a diverse profondità; (c) visualizzazione del piano di ritaglio;



a)

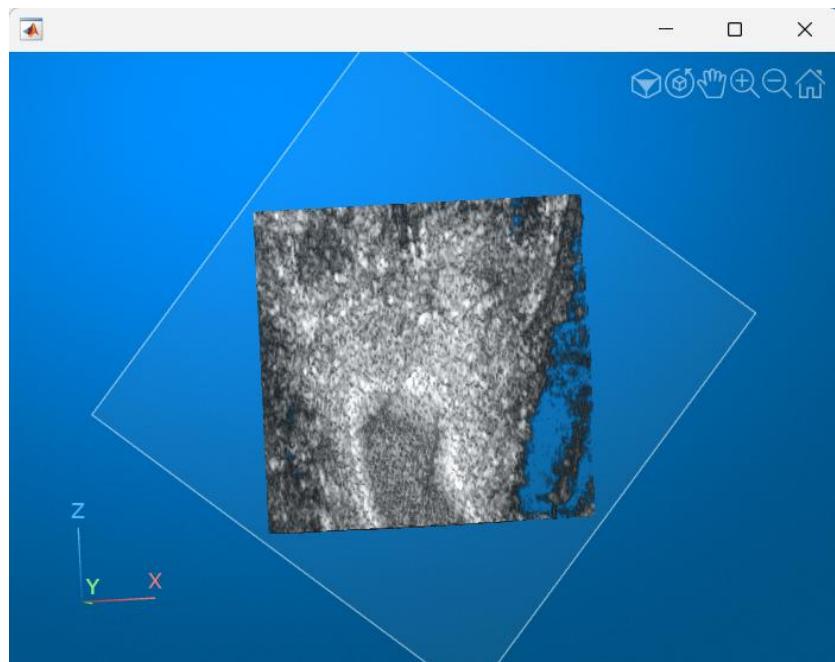


b)

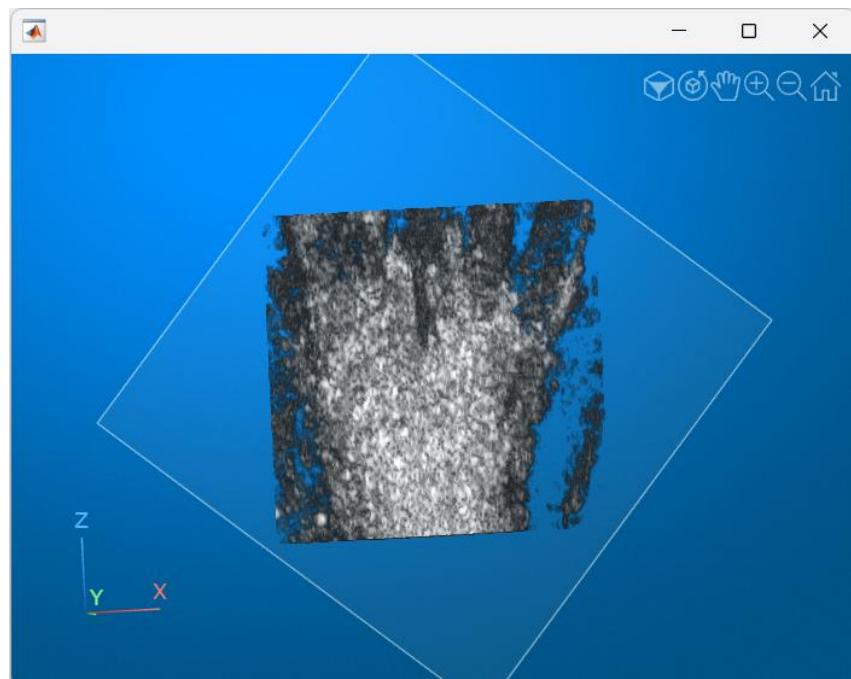


c)

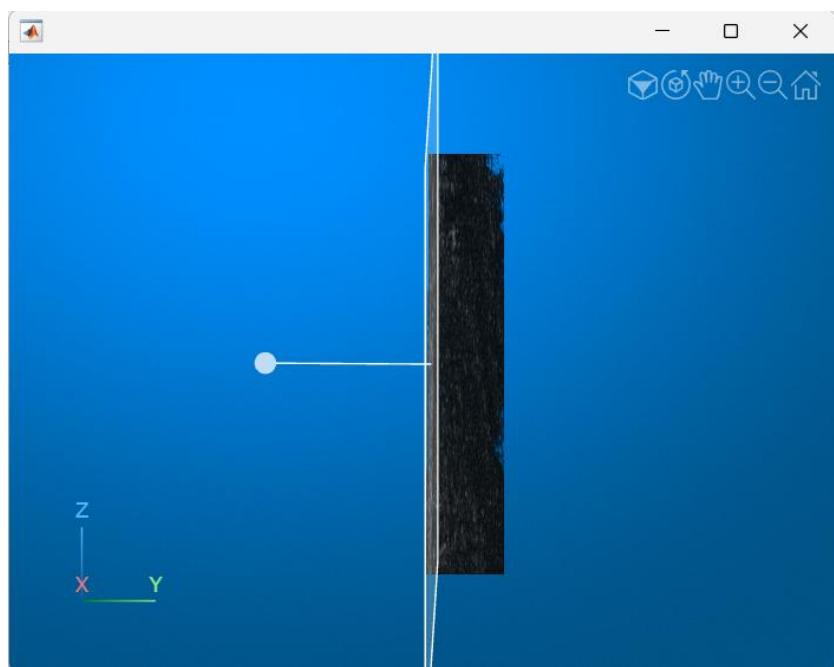
Figura 7.3 - (a), (b). Esempi di X-BSCAN interattivi con piano di ritaglio a diverse profondità; (c) visualizzazione del piano di ritaglio;



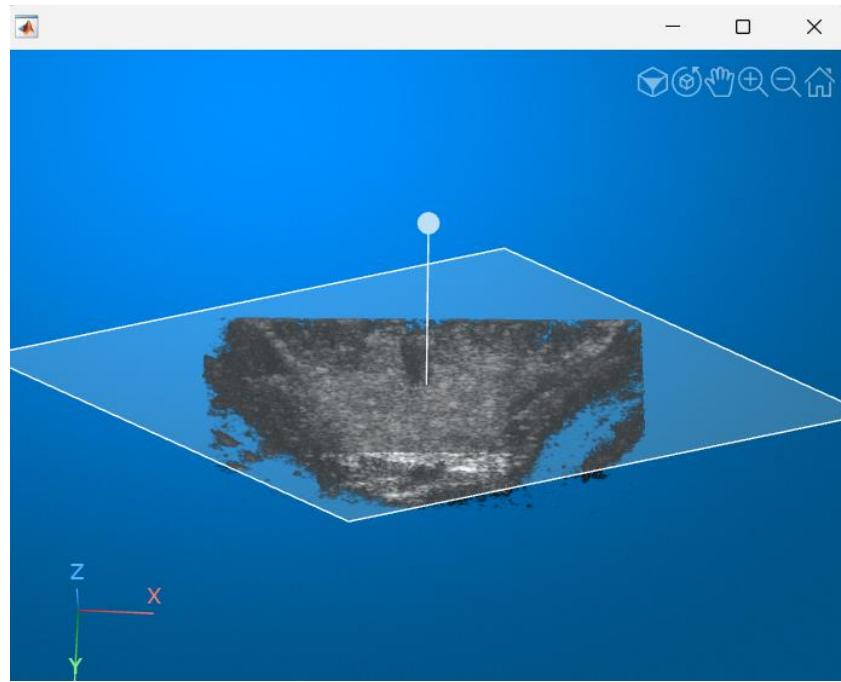
a)



b)



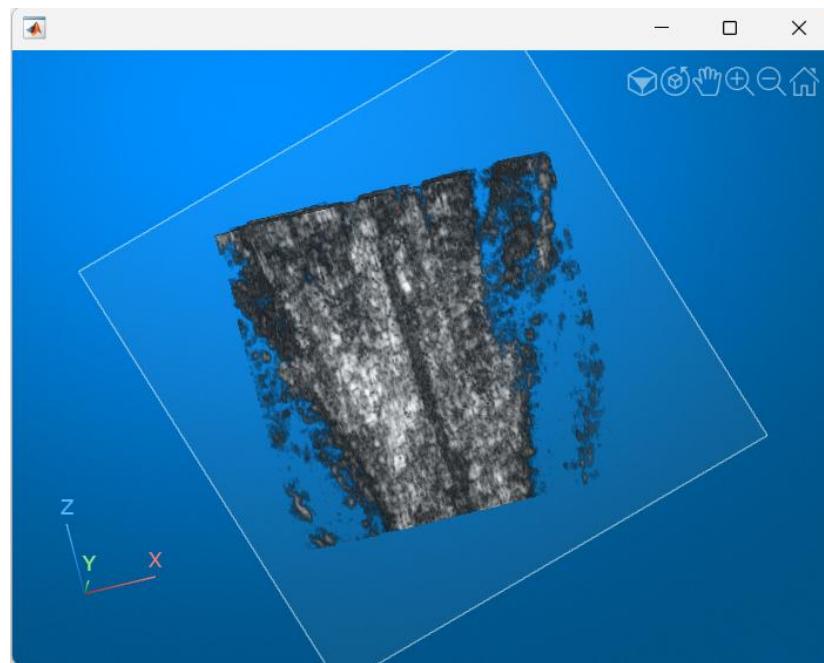
c)



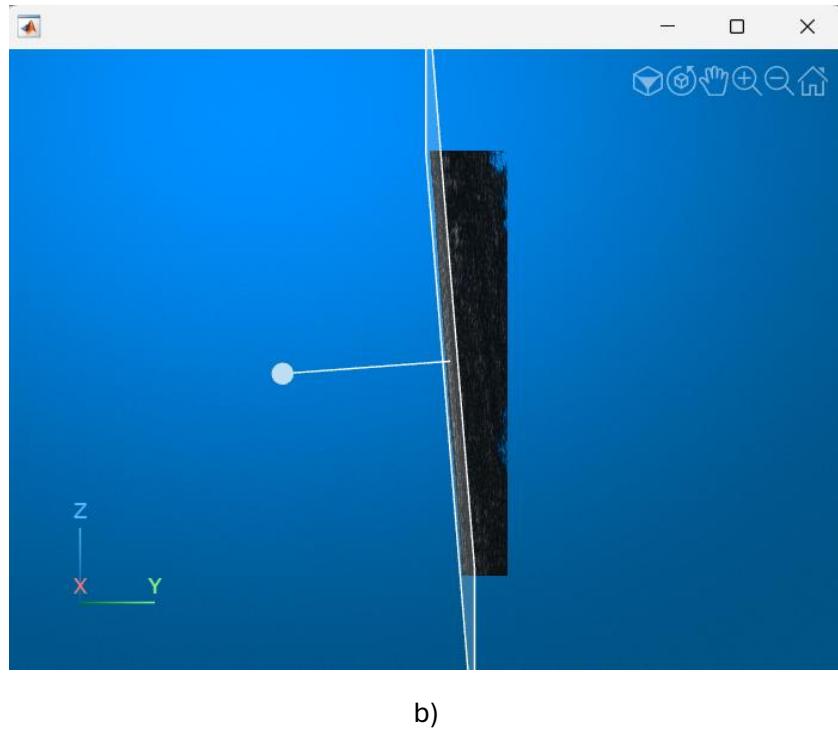
d)

Figura 7.4 - (a), (b). Esempi di CSCAN interattivi con piano di ritaglio a diverse profondità; (c), (d) visualizzazione del piano di ritaglio;

Grazie alla funzionalità ‘clipping plane’, è possibile anche inclinare il piano di ritaglio, funzione non disponibile in ‘Biometric Data Representation’, permettendo così di riconoscere chiaramente l’intero sviluppo del tratto venoso, anche quando si dirama in **profondità**:



a)



b)

Figura 7.5 – (a), (b). Piano di ritaglio inclinato con visualizzazione intera della vena.

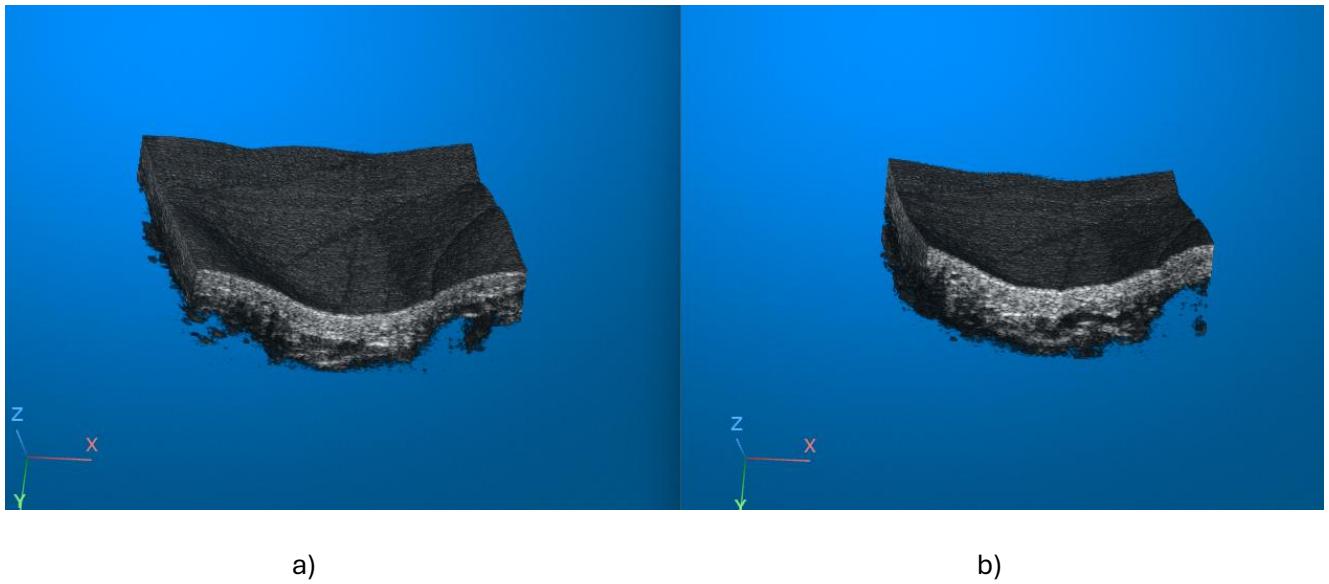
In merito alla praticità e velocità di visualizzazione del volume 3D del palmo della mano, sia lungo le tre sezioni ortogonali X-BSCAN, Y-BSCAN e Z-CSCAN che lungo qualsiasi altra direzione trasversale, è stato accantonato il metodo di visualizzazione limitato di ‘*Biometric Data Representation*’ per il più concreto metodo di Matlab ‘*volshow*’. L’utilizzo di quest’ultimo è stato fondamentale per avere un riscontro visivo sull’effettiva estrazione del template venoso in maniera corretta durante le fasi successive dello studio.

7.3 CROPPING

La funzione **cropMatrice.m** ridimensiona la matrice volumetrica M (contenuto informativo dei file .mat), di dimensioni originali $835 \times 820 \times 320$ voxel, estraendone una porzione centrale di **650×650×320** voxel.

La **zona centrale** del volume contiene le informazioni più rilevanti per l’analisi, poiché qui le vene risultano maggiormente visibili e distinguibili. Al contrario, i bordi del volume presentano un’elevata quantità di “rumore” e artefatti indesiderati che complicano il processo di feature extraction; questi disturbi, essendo troppo attaccati alle vene, sono difficili da separare senza compromettere la qualità d’insieme del pattern venoso. Il cropping centrale permette di isolare la parte interna della matrice, riducendo l’influenza del rumore periferico e preservando dettagli rilevanti per l’estrazione delle feature, senza “sporco” eccessivo.

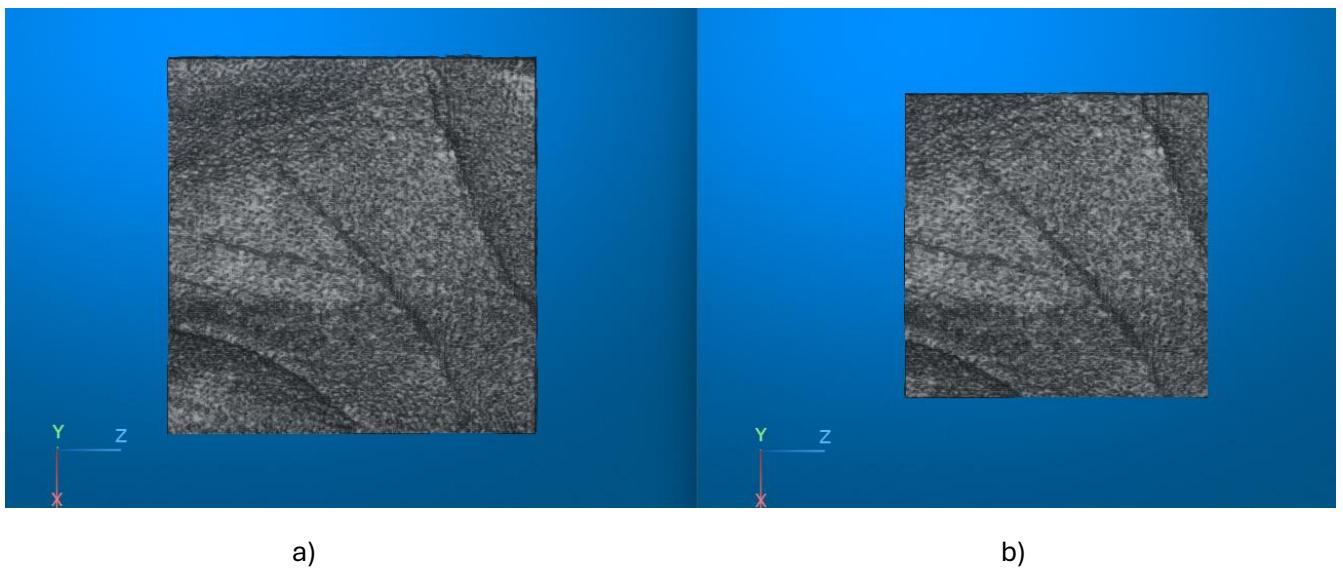
Inoltre, la riduzione della matrice M a una risoluzione più contenuta riduce significativamente il numero di voxel da elaborare, con un conseguente risparmio nei tempi di calcolo. Questo è particolarmente vantaggioso nelle successive fasi di estrazione delle feature, che altrimenti richiederebbero un elevato carico computazionale per l’analisi dell’intero volume originario.



a)

b)

Figura 7.6 - a) volume $835 \times 820 \times 320$; b) volume $650 \times 650 \times 320$.



a)

b)

Figura 7.7 - (a) volume $835 \times 820 \times 320$; (b) volume $650 \times 650 \times 320$.

7.4 ESTRAPOLAZIONE

Il codice di `estrapolaVolumeVene.m` è strutturato in maniera tale da effettuare diversi filtraggi della matrice M servendosi di apposite maschere logical con lo scopo di estrarre il volume 3D del palmo della mano privo sia della componente ‘acqua’ che delle componenti di rumore in eccesso alla base del palmo, con lo scopo di diminuire l’onere computazionale delle fasi successive.

All’inizio è importante definire una matrice di trasparenza A (utile alla creazione di `mascheraAcqua`), la quale viene impostata a 0 (completamente trasparente) per i voxel che sono al di sopra della superficie del palmo e a 255 (completamente opaco) per quelli che si trovano sotto la superficie, entro un certo spessore. Successivamente, se la variabile `trans_flag` è impostato a 1, viene applicata una “rampa” di trasparenza attorno alla superficie del palmo. Questo gradiente consente una transizione

graduale da trasparente a opaco nei voxel vicini alla superficie, invece di un cambiamento netto, migliorando la visualizzazione del pattern palmare.

```
Mstart = M;
mascheraAcqua = (A <= thresh);
Mstart(mascheraAcqua) = 0;
```

Successivamente, viene creata una maschera logica, **mascheraAcqua**, che identifica quali voxel nella matrice A hanno un valore inferiore o uguale a una certa soglia (**thresh**). Questa soglia serve a individuare le aree che rappresentano l'acqua o altre parti sopra il palmo, settando tali voxel a **true** nella maschera.

Questa maschera **mascheraAcqua** viene applicata alla matrice **Mstart**, una copia del volume di partenza **M**. Usando la maschera, tutti i voxel di **Mstart** che corrispondono a valori true in **mascheraAcqua** vengono impostati a zero, rimuovendoli dal volume visualizzato. In questo modo, viene eliminata dal volume la parte **superiore contenente l'acqua**, lasciando solo la porzione che rappresenta la superficie del palmo e le aree sottostanti contenenti il pattern venoso d'interesse.

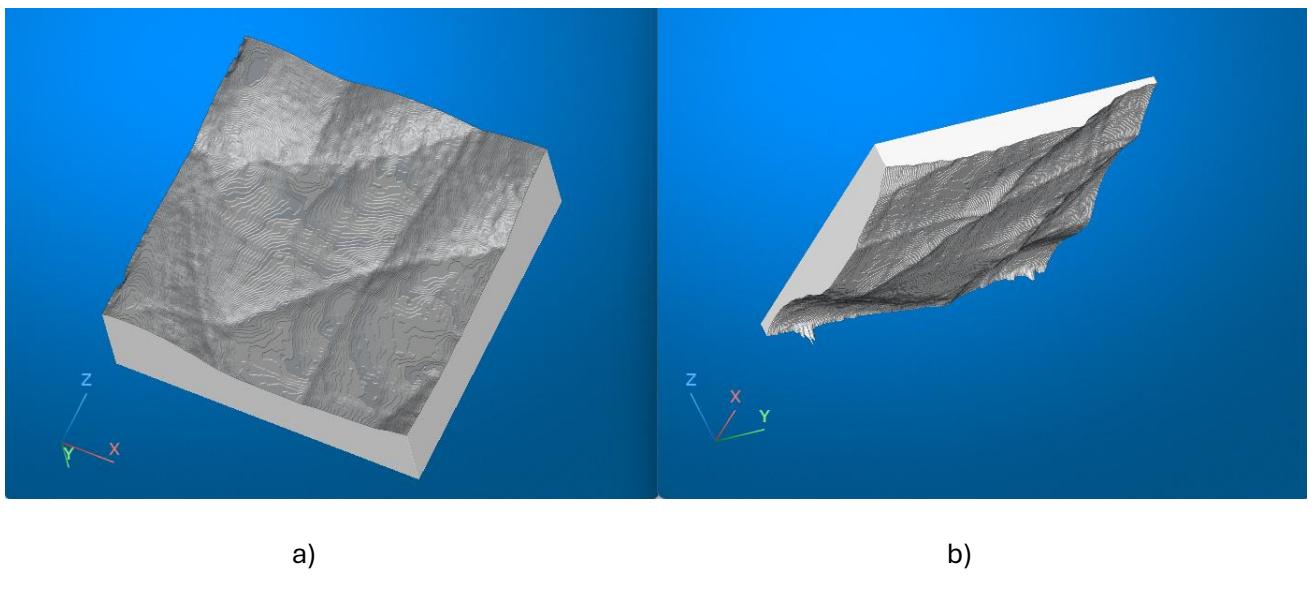


Figura 7.8 - (a) matrice di trasparenza A con rampa di trasparenza abilitata; (b) mascheraAcqua.

Dopodiché, nel metodo **calcolaMaschere.m** viene creata la maschera **mascheraNeroPalmo**.

Per ogni colonna (x , y) del volume, viene individuata la posizione **altezzaPalmo**, l'indice in cui termina il palmo (ovvero, il primo voxel non nullo dall'alto verso il basso). Tutti i voxel sotto (**altezzaPalmo** - **offsetPalmo**) vengono impostati a **true** in **mascheraNeroPalmo**.

```
mascheraNeroPalmo(y,x,1:altezzaPalmo-offsetPalmo) = true;
```

offsetPalmo è utilizzato per spostare la maschera più in basso rispetto alla superficie del palmo, quindi dare un offset dalla superficie palmare identificata da **altezzaPalmo**, eliminando i voxel sotto questo offset. Abbiamo deciso di impostare un valore che riuscisse a rappresentare un buon compromesso tra il mantenimento delle vene più profonde e l'esclusione dei voxel associati al rumore

in fondo al volume. Dopo numerose applicazioni su acquisizioni di più utenti, il valore, stabilito empiricamente e che potrebbe anche cambiare con sviluppi futuri del metodo CCM, è stato di 200.

Quindi, questa maschera identifica la parte del volume sotto il palmo considerato rumore superfluo. Serve per escludere questa porzione dal volume totale.

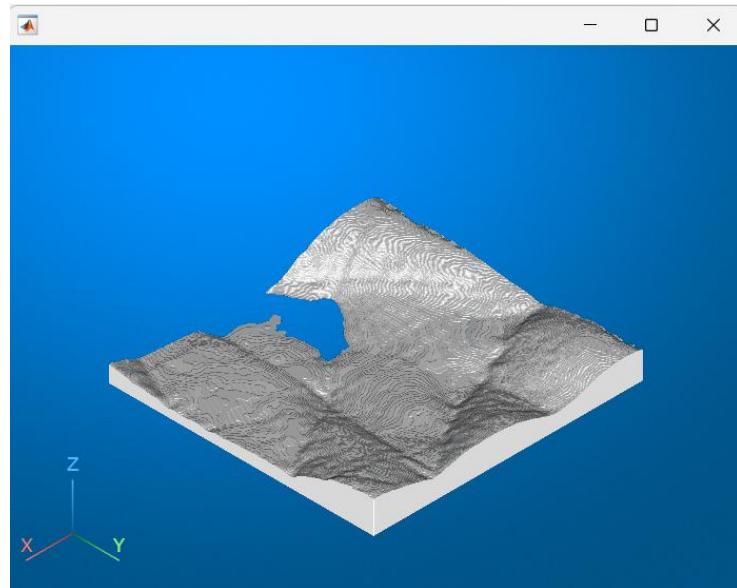


Figura 7.9 - mascheraNeroPalmo.

Infine si determina la maschera **mascheraNeroTotale**; questa maschera combina **mascheraNeroPalmo** e **mascheraAcqua**, rappresentando tutte le aree da escludere nel volume finale **Mnp**.

```
mascheraNeroPalmoAcqua = mascheraNeroPalmo | mascheraAcqua;
```

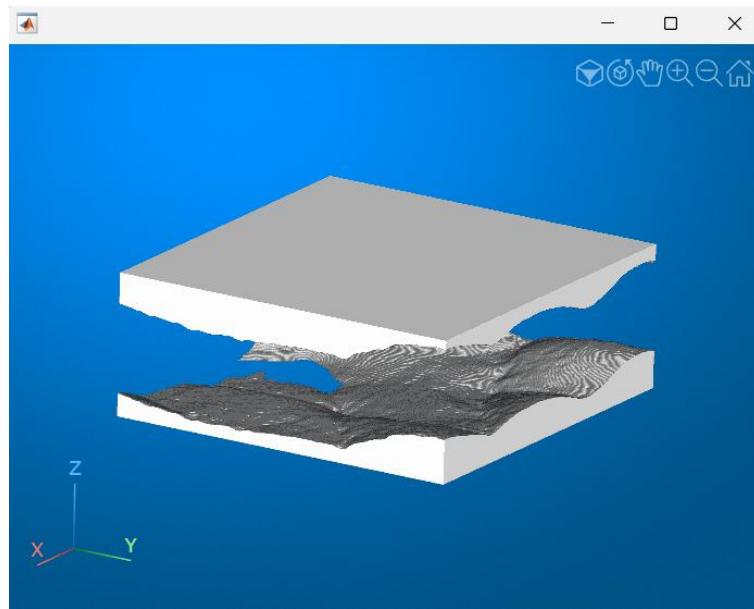


Figura 7.10 - mascheraNeroPalmoAcqua.

Di seguito il volume 3D di partenza Mstart ed il volume Mnp in output da *estrapolaVolumeVene.m* su cui è stata applicata la maschera **mascheraNeroPalmoAcqua**

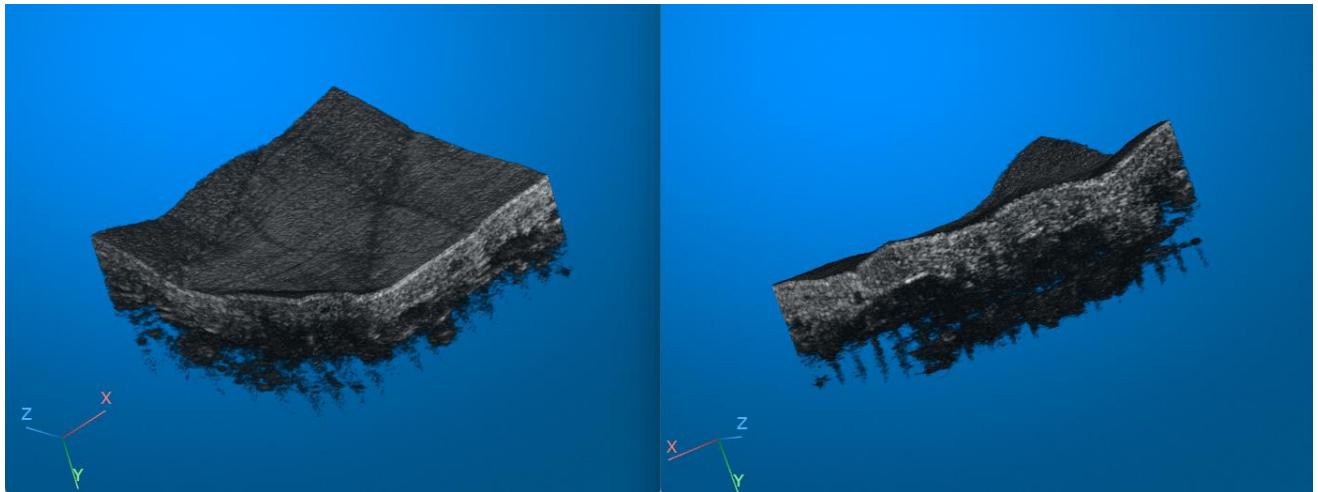


Figura 7.11- Mstart.

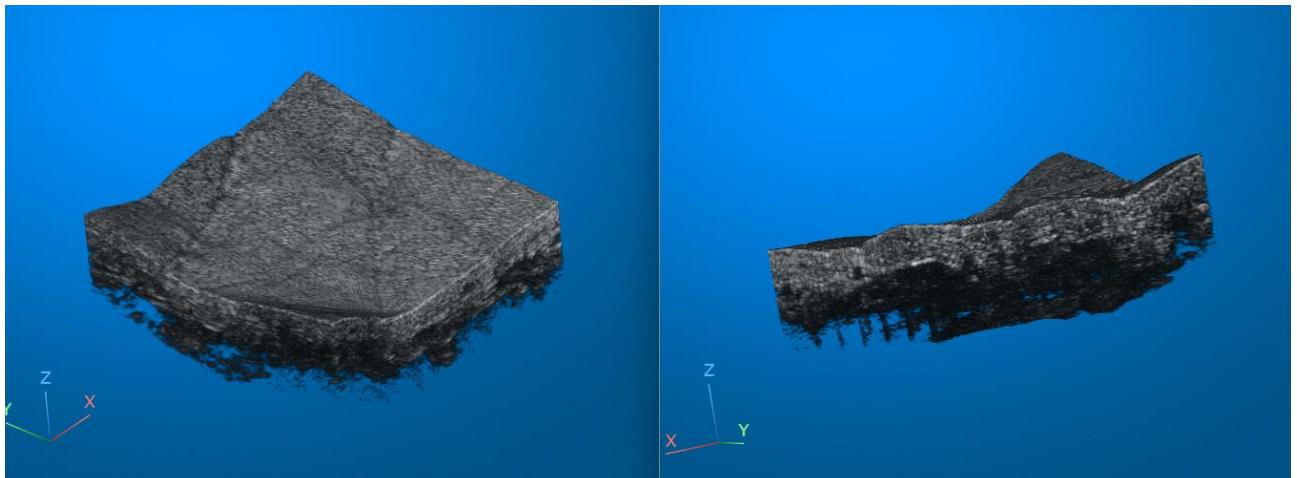


Figura 7.12 - Mnp

7.5 BINARIZZAZIONE

Il processo di binarizzazione consiste nel convertire il volume di partenza in formato uint8 con valori da 0 a 255 in un volume binario con valori logici di 0 e 1. Nel passaggio dal tradizionale range di valori utilizzato per rappresentare le immagini in scala di grigio a una visualizzazione binaria è importante mantenere la maggior quantità di informazioni e allo stesso tempo escludere tutti quei voxel che non sono di interesse nel processo di estrazione del pattern venoso. Il risultato finale di questa fase sarà il volume con le dimensioni croppate ($650 \times 650 \times 320$) contenente il maggior numero possibile di voxel che raffigurano un collegamento venoso. Effettuare questo procedimento per utenti diversi risulta essere particolarmente difficoltoso in quanto un valore di soglia stabilito empiricamente non sarà mai adattabile a diverse situazioni e quindi varie forme e dimensioni delle vene associate a utenti diversi.

Per questo è stato scelto di utilizzare una binarizzazione che si riesca ad adattare al singolo utente in modo tale da estrapolare più informazione possibile. Questa procedura viene denominata **binarizzazione incrementale adattiva**.

La binarizzazione incrementale è un tipo di binarizzazione che si basa su 4 informazioni:

1. Coordinata lungo Z in cui parte la binarizzazione
2. Coordinata lungo Z in cui finisce la binarizzazione
3. Soglia iniziale di binarizzazione
4. Soglia finale di binarizzazione

Con queste informazioni riesco a realizzare una binarizzazione che varia al variare di Z, quindi la soglia dipende dalla coordinata Z in cui effettuo la binarizzazione. È una procedura adattiva perché al variare di queste informazioni posso regolare quanti voxel prendere e quali scartare in base alle caratteristiche univoci degli utenti.

Al fine di realizzare la binarizzazione adattiva è necessario ricavare le 4 informazioni menzionate, per farlo è stato scelto il *processo iterativo*.

L'intera fase di binarizzazione prende luogo nello script **effettuaBinarizzazione**, che dato in input le informazioni provenienti dallo step precedente (**Mnp**, **volumePalmo**, **mascheraAcqua**, **mascheraNeroTotale**) restituisce il volume binarizzato **volBinFinal**.

7.5.1 Calcolo del vettore fine binarizzazione

Per realizzare una binarizzazione che tenesse conto della pendenza e quindi di come si sviluppa la vena lungo la dimensione Z è stato necessario effettuare una binarizzazione incrementale del volume per piani XZ. Quindi viene scansionata la matrice di partenza nei suoi 650 piani XZ e per ognuno di essi viene stabilita la prima informazione utile ad effettuare una binarizzazione adattabile ad ogni utente, ovvero il valore di *fine binarizzazione*. Questo valore rappresenta la coordinata lungo Z nella quale andrà a terminare la binarizzazione e quindi determina il range utile all'interno del quale il piano XZ contiene le vene di interesse. Questo range viene determinato partendo dalla coordinata alla quale si trova il palmo della mano e procedendo per tentativi successivi si calcola un valore soglia oltre il quale verranno presi solo i pixel maggiori di 255 e quindi non verrà preso più nessun pixel.

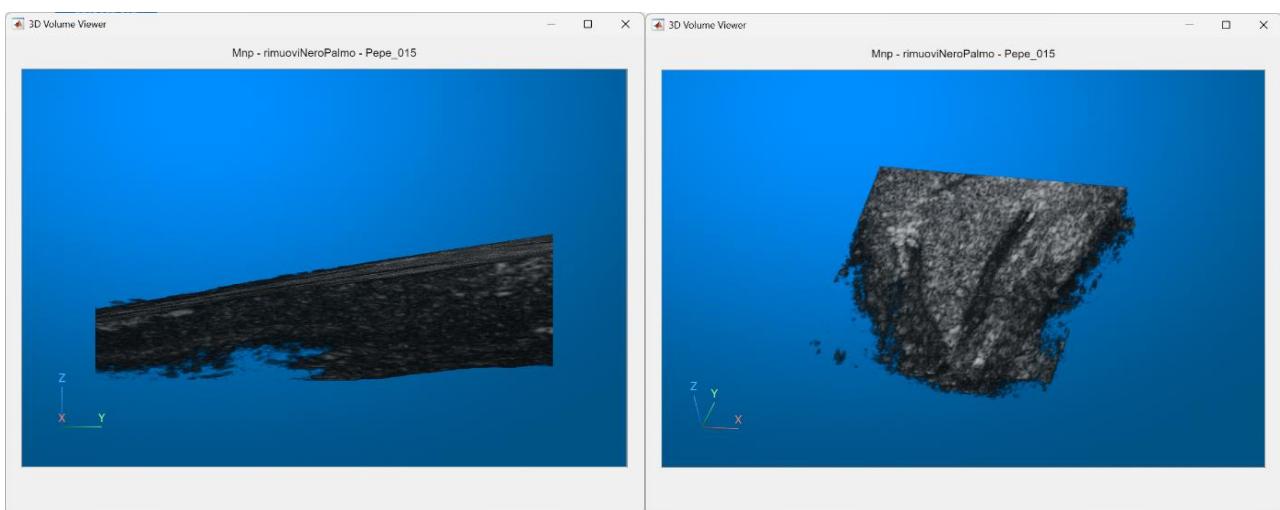


Figura 7.13 - Esempio di un utente con vene molto pendenti

Il codice quindi funziona in questo modo: dato ogni piano XZ questo viene binarizzato a partire dagli indici Z contenenti il palmo della mano fino agli indici Z di fine scelti pari alla minima coordinata Z del palmo. Quindi il piano(X-BSCAN) viene binarizzato a partire dal palmo fino a un indice soglia aumentando la profondità di 1 ad ogni iterazione. Per ogni ciclo viene salvato il numero di componenti connesse del piano in un vettore di appoggio. In questa fase di binarizzazione del piano vogliamo

individuare quella che è la vena che si trova più in profondità dato che stiamo impostando un limite di fine binarizzazione. Qui andiamo a fare la nostra **prima assunzione**:

- La vena che si trova più in basso è anche quella più consistente e quindi sarà caratterizzata da molti voxel pari a 0.

La conseguenza di questa assunzione è che per caratterizzare la vena di fine binarizzazione è necessario impostare un valore di soglia pari a 254 e quindi andremo a prendere solo i pixel del piano che hanno valore 0. Appurato questo c'è da illustrare come contiamo le componenti connesse del piano.

Vediamo cosa succede binarizzando il primo BSCAN ($y = 1$) alle varie iterazioni

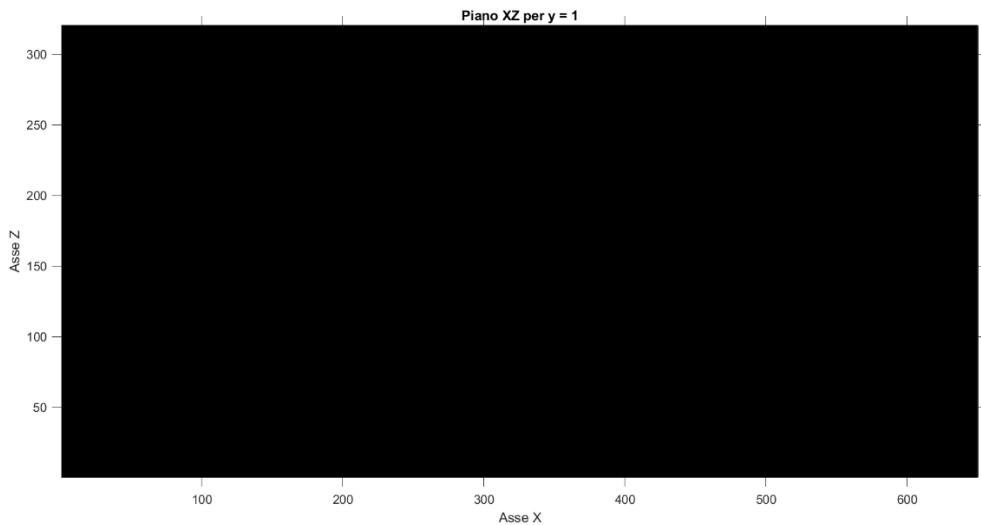


Figura 7.14 - Prima iterazione binarizzazione piano XZ per $Y = 1$

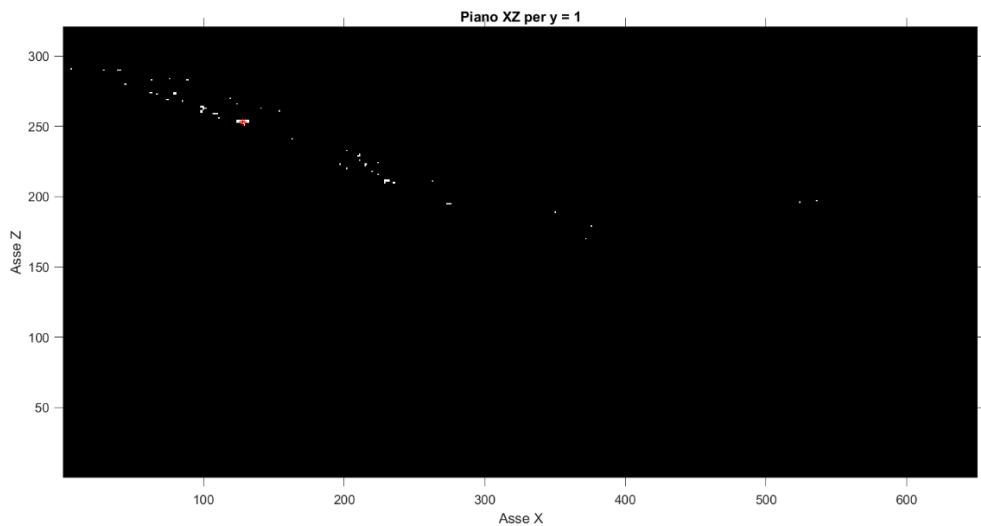


Figura 7.15 – Iterazione 23 binarizzazione piano XZ per $Y = 1$

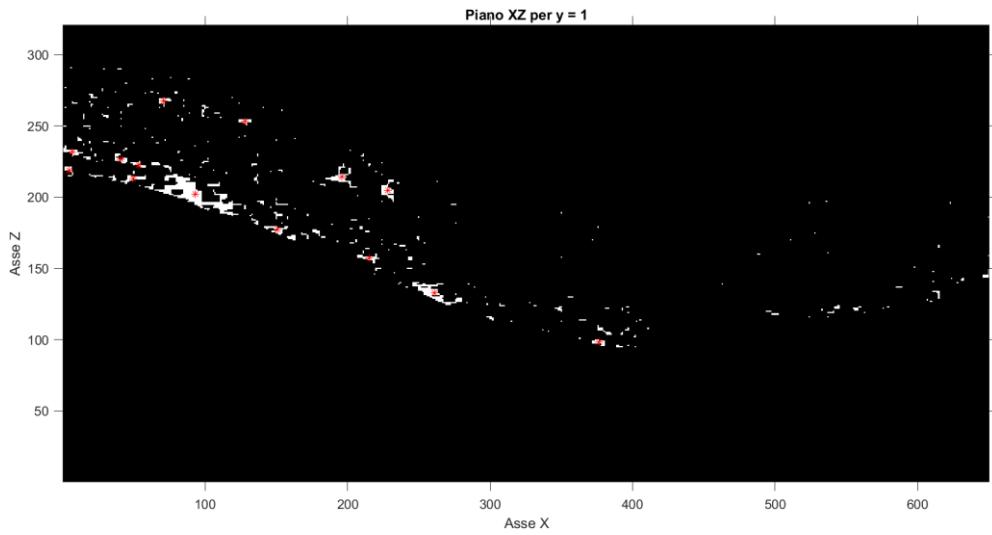


Figura 7.16 - Iterazione 90 binarizzazione XZ per Y = 1

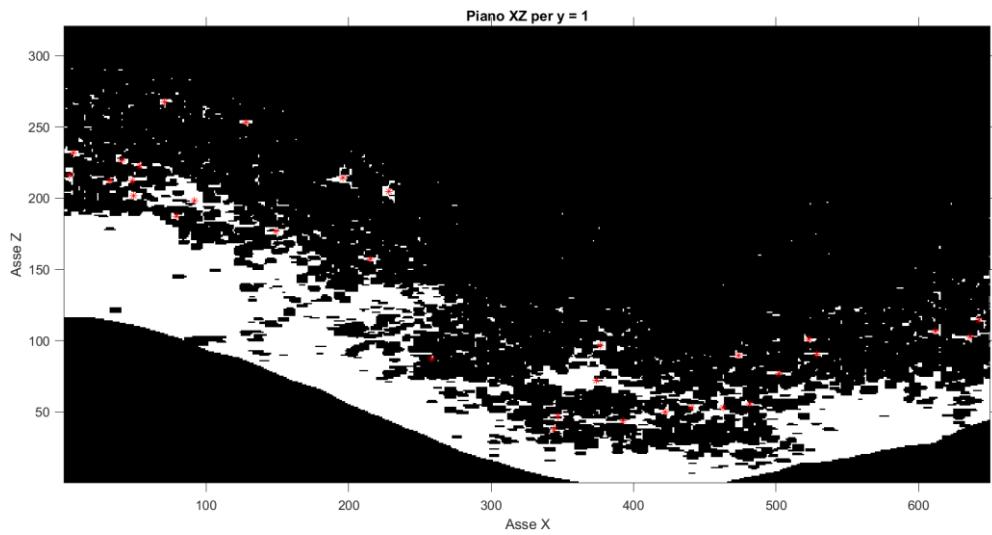


Figura 7.17 - Ultima iterazione binarizzazione XZ per Y = 1

Per ogni iterazione viene contato il numero di centroidi presenti nel piano (visibili in rosso * nella Figura 7.16), per evitare di falsare il calcolo viene imposto un filtraggio sulla dimensione delle componenti connesse considerando solo quelle aventi un'area maggiore di 20 in modo tale da non considerare tutti quei puntini che non sono di interesse.

Notiamo che alla prima iterazione non compare nessun centroide, questo perché ci troviamo nella zona vicino al palmo e quindi è difficile trovare voxel così scuri. Più si decrementa la fine della binarizzazione e più si trovano voxel scuri di conseguenza aumenta il numero delle componenti connesse trovate.

Procedendo per tutte le iterazioni possiamo visualizzare il grafico del vettore contenente il numero delle cc (componenti connesse) ad ogni iterazione.

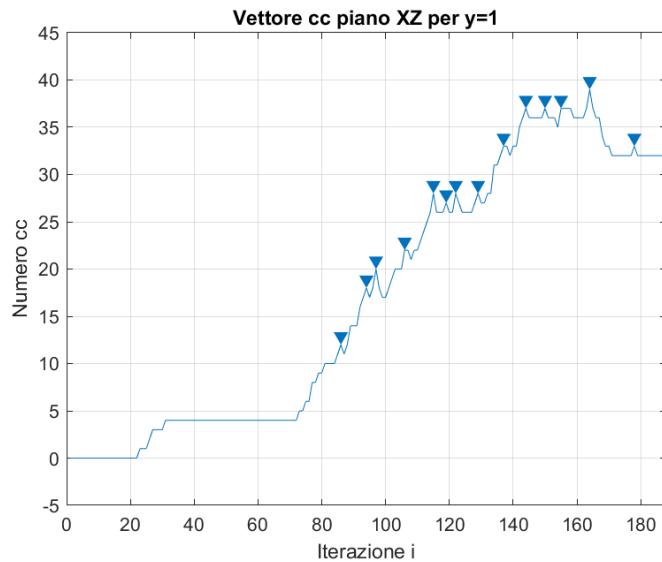


Figura 7.18 - Plot del numero delle cc per ogni iterazione del piano XZ

A questo punto dobbiamo scegliere un valore di fine binarizzazione per questo piano e per farlo andiamo a ricavare l'indice in corrispondenza dei primi due massimi locali del vettore e ne facciamo una media.

Come si può osservare dal grafico in Figura 7.18 il numero delle componenti connesse tende a crescere con l'aumentare delle iterazioni e quindi con l'aumentare della coordinata di fine. Scansionando il piano dal palmo alla coordinata Z=0 si prendono sempre più componenti ma durante il variare delle iterazioni si incontrano dei **picchi**. Questi punti, corrispondenti a dei massimi locali, indicano che il numero delle cc cresce e poi decresce mostrando come la connessione delle componenti nel piano fa decrescere il numero di centroidi presenti.

Questo indica che probabilmente siamo ad una coordinata che contiene una vena perché le varie sfumature che la compongono compaiono nel piano e dopo qualche iterazione si vanno a connettere formando la sezione della vena, di conseguenza il numero delle cc decresce.

Osserviamo che procedendo con le iterazioni il numero delle cc tende a crescere fino a raggiungere un plateau dopo l'iterazione 160 e tra i vari picchi ci sono delle zone in cui il numero delle cc rimane costante. Conseguenza del fatto che non vengono aggiunte altre strutture e quelle presenti non si connettono tra di loro. Per evitare di considerare componenti non di interesse ma conservare solo le vene è stato provato che la media dei primi due massimi riesce a considerare bene la vena principale, obiettivo di questa fase.

Il valore scelto viene salvato nel vettore **vecFineBin** che quindi contiene il valore ottimo di fine binarizzazione per ogni piano XZ dell'acquisizione. Una volta completati tutti i piani è possibile visualizzare il vettore ricavato. Prima di discutere il vettore fine è importante notare che questa fase è molto dispendiosa in termini di calcolo per cui è assolutamente fondamentale realizzare il primo ciclo for su tutti i piani XZ(650) in modo parallelo, questo velocizzerà di molto l'esecuzione che richiederà comunque un tempo ragionevolmente lungo.

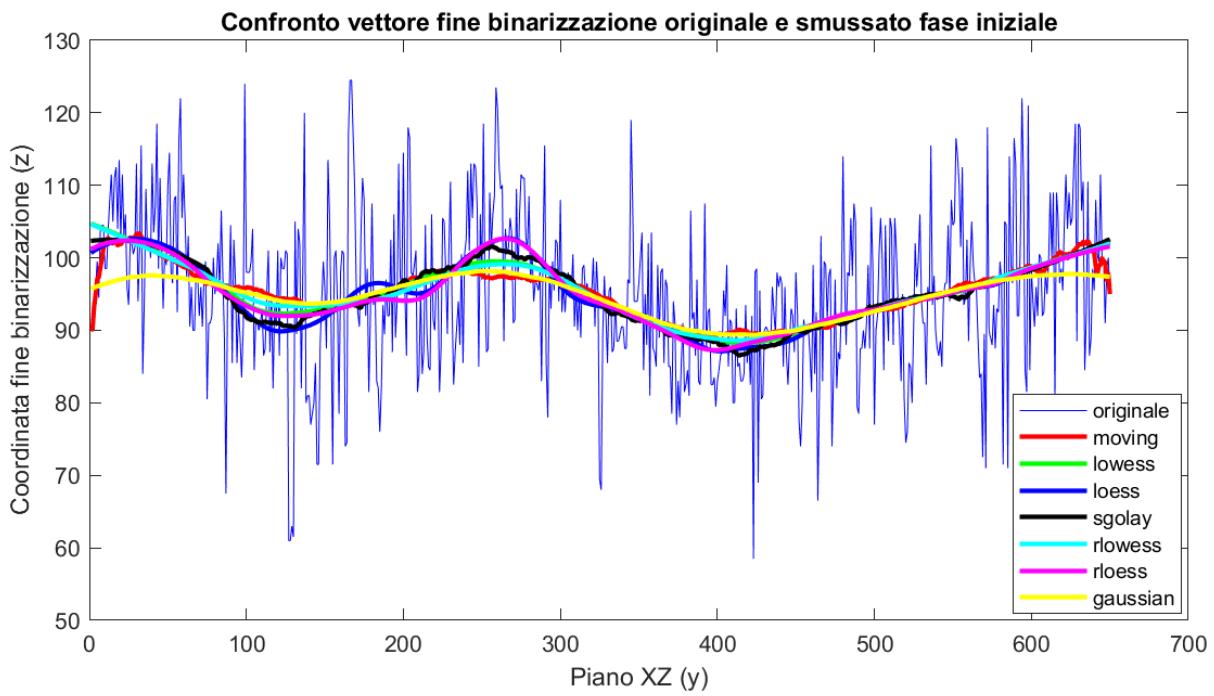


Figura 7.19 - Fine binarizzazione calcolata per ogni piano XZ del volume

Nella Figura 7.19 possiamo vedere il valore di fine ottimo calcolato per ogni piano XZ il quale rappresenterà l'andamento della binarizzazione lungo la dimensione Z e quindi sarà in grado di ricavare sia le vene più pendenti che quelle costanti. Per rendere la caratterizzazione delle vene più omogenea è stato deciso di applicare al vettore in questione una procedura di smoothing. Lo smussamento va a regolare i picchi non dando peso a possibili outlier ma conservando l'andamento generale delle vene profonde. Sono stati provati diversi operatori di smoothing ma alla fine si è visto che l'operatore **rlowess** funziona meglio per questo scopo (tratto verde acqua).

L'operatore rlowess applica una regressione locale pesata ai dati che ricadono nella finestra di smoothing, ovvero il numero di dati sui cui si calcola la curva smussata. I punti più vicini al valore da stimare hanno pesi maggiori rispetto ai punti più lontani. Per approssimare localmente i dati utilizza una semplice funzione lineare, agli outlier viene assegnato peso pari a zero se si trovano al di fuori di sei deviazioni medie assolute.

Da questo vettore possiamo intuire come l'utente in questione abbia nella zona del BSCAN che va da 0 a 200 la vena principale ad una profondità nell'intorno di 100 per poi risalire più verso la superficie da 250 in poi fino a 500 per poi scendere di nuovo.

7.5.2 Prima binarizzazione del volume

Una volta ricavato il vettore di fine binarizzazione possiamo costruire il primo volume binarizzato che ci servirà per ricavare le altre caratteristiche con cui andare a realizzare la binarizzazione finale. Con questa informazione possiamo visualizzare quella che è la vena principale profonda dell'acquisizione e quindi procediamo a binarizzare tutti i BSCAN del volume con il vettore precedentemente calcolato. In questa fase andiamo ad effettuare una binarizzazione dell'intero piano con un valore soglia di 250 dato che ci interessano solo le vene profonde e consistenti fermandoci al valore di fine ricavato. Possiamo visualizzare il risultato nella figura che segue.

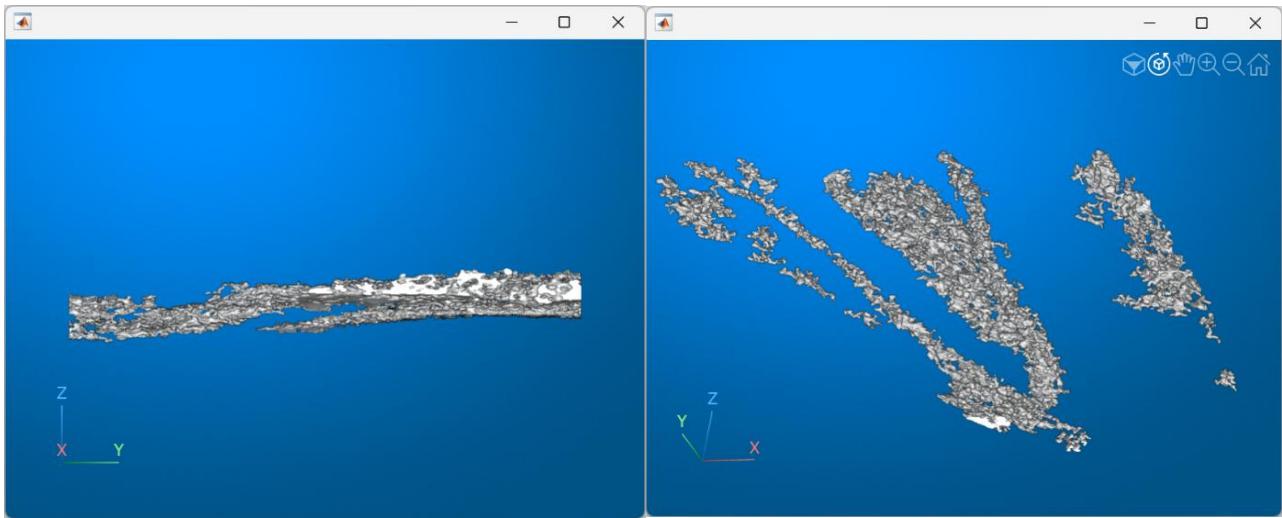


Figura 7.20 - Volume binarizzato fase 1

Possiamo vedere come la vena profonda segue l'andamento della fine calcolata, inoltre dalla figura si nota come non siano presenti altre componenti al di sopra della vena questo perché oltre alla binarizzazione è stato effettuato un taglio lungo Z del singolo piano in modo da conservare solo le strutture profonde. Il taglio è stato effettuato con un offset costante pari a 60.

7.5.3 Calcolo vettore distanza minima tra le vene e il palmo

La seconda informazione utile per realizzare una binarizzazione che si adatti in modo automatico a più utenti possibili è la distanza che intercorre tra la vena e il palmo della mano. In questa fase andiamo a illustrare la nostra **seconda assunzione**:

- Partendo dal palmo della mano e dirigendosi verso il basso i primi voxel che si incontrano con intensità molto bassa rappresentano delle vene.

Questa features ci permetterà di calcolare la coordinata lungo Z definitiva in cui termina la binarizzazione, vediamo ora come viene ricavata questa seconda feature.

Il procedimento segue quanto visto per il vettore fine binarizzazione solo che non adottiamo una procedura iterativa ma scansionando i vari BSCAN calcoliamo la distanza fisica che intercorre tra il palmo e la cc che si trova più in superficie (che come assunto è la vena). Per fare questo andiamo prima a costruire il volume combinato tra la binarizzazione che abbiamo appena visto e il palmo della mano che possiamo vedere nella figura seguente.

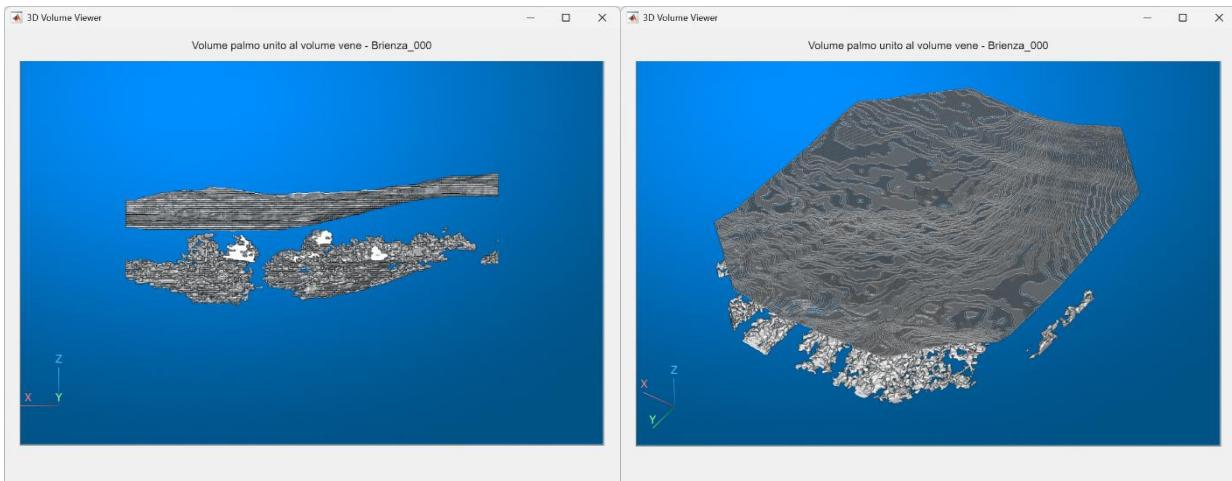


Figura 7.21 - Volume combinato palmo della mano e binarizzazione fase 1

Con questo volume adesso possiamo scansionare i diversi piani XZ e calcolare la distanza dalla vena al palmo per ogni BSCAN. Per farlo andiamo sempre a filtrare le cc del piano in modo tale da escludere il rumore in base alla dimensione dell'area e successivamente scansioniamo ogni colonna del piano. Quindi per ogni X e per ogni Y definiamo la distanza come:

- La coordinata Z del palmo meno la prima coordinata a partire dal palmo che contiene 3 pixel consecutivi pari a 1.

Una volta calcolate tutte le distanze del piano andiamo a scegliere il minimo, la distanza minore di quel piano e quella sarà la coordinata di fine binarizzazione per il BSCAN in questione. Così facendo siamo sicuri che effettuando una binarizzazione che parte dal palmo e arriva al valore calcolato andiamo a considerare nel volume binarizzato soltanto la porzione di matrice contenente dei voxel associati a delle vene.

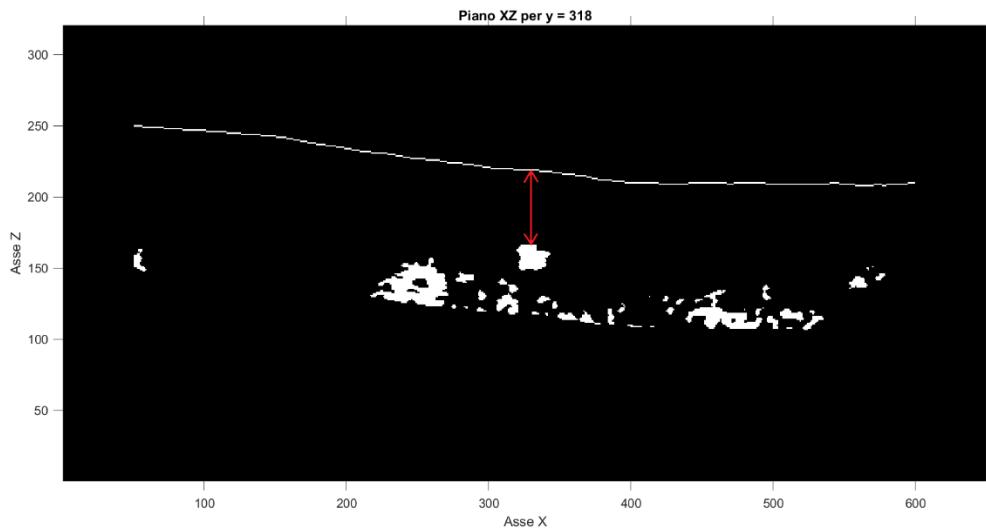


Figura 7.22 - Esempio distanza scelta per un dato piano XZ

Procedendo per ogni piano otteniamo la features desiderata per ogni y che possiamo vedere nella figura seguente.

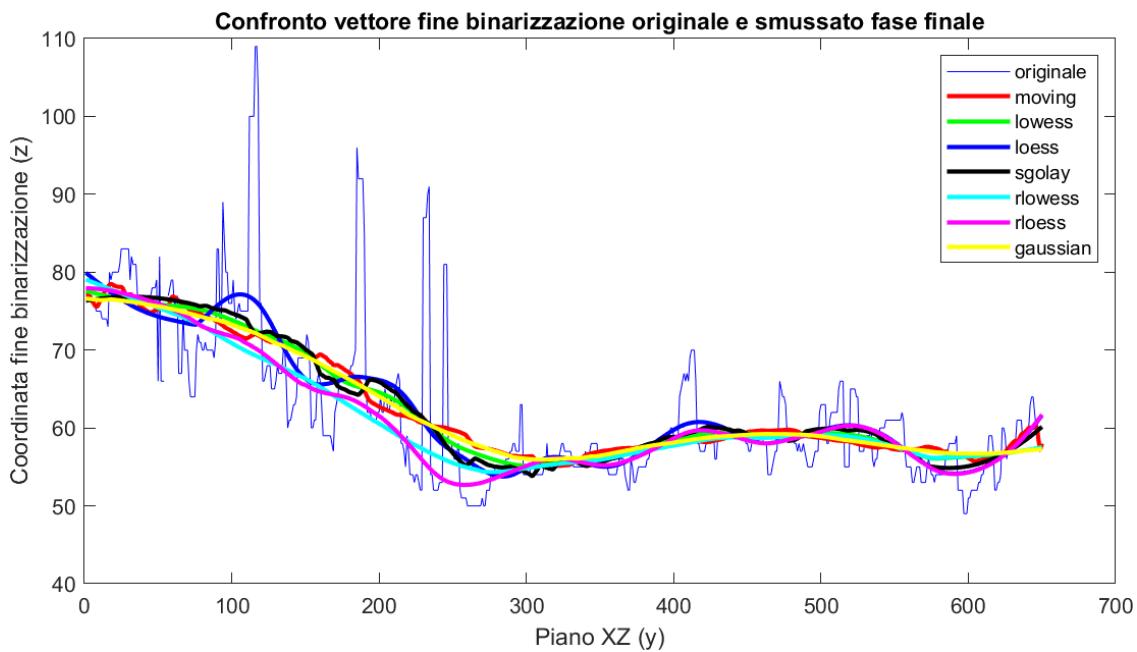


Figura 7.23 - Distanze minime tra le cc e il palmo per ogni BSCAN

Bisogna precisare quindi che il calcolo del vettore fine binarizzazione serve solo per creare un primo volume binarizzato su cui poter effettuare il calcolo delle distanze minime. Il vettore di fine utile ad effettuare la binarizzazione definitiva sarà quello contenente le distanze minime tra le vene e il palmo.

7.5.4 Calcolo soglia iniziale

L'ultima feature necessaria per effettuare la binarizzazione finale del volume è la *soglia iniziale*.

In questa fase andiamo ad enunciare la nostra ultima assunzione:

- *Le vene sottili sono costituite da valori di intensità maggiori rispetto a quelle massicce profonde e si trovano più in superficie.*

La soglia iniziale rappresenta il valore oltre il quale andiamo a considerare i voxel della matrice inversa. Questo valore non può essere assegnato in modo fisso in quanto ogni utente ha delle caratteristiche diverse e quindi il risultato finale sarebbe insoddisfacente. È un valore che va a regolare quante vene sottili il metodo si pone di estrarre. Un valore minore della soglia iniziale sarà in grado di conservare più vene, anche quelle più sottili e superficiali ma al contempo introdurrà parecchio rumore che deve poi essere eliminato. Un valore maggiore andrà a conservare solo le vene principali riducendo il rumore introdotto.

La soglia iniziale permette di raccogliere più o meno voxel nella zona vicina al palmo, quindi dove si trovano le vene più superficiali. Così facendo il metodo è in grado di ricavare sia le vene robuste che si trovano in profondità e sia quelle più esili che si trovano in superficie.

Anche in questo caso il metodo che si è rivelato più efficace per fissare questo parametro è stato quello *iterativo*.

Per procedere con questo metodo innanzitutto bisogna fissare una **soglia iniziale** di partenza e una **soglia iniziale** di arrivo, scelte rispettivamente pari a 210 e 250, il numero di iterazioni sarà quindi di 40. Il prossimo step consiste nell'effettuare una binarizzazione incrementale del volume partendo dalla soglia iniziale di 250 fino a raggiungere il valore di 210.

7.5.4.1 Binarizzazione incrementale

La binarizzazione incrementale progettata per questo metodo di estrazione consiste nel variare dinamicamente la soglia tra un valore di partenza e uno di arrivo con l'interare delle coordinate Z. La funzione consiste nel far variare questa soglia con legge quadratica, quindi la soglia varierà di poco nelle prime iterazioni (ovvero nella zona vicina al palmo della mano) e invece varierà di molto nelle ultime iterazioni avvicinandosi rapidamente al valore di fine, scelto sempre pari a 255 in modo tale da interrompere la raccolta dei voxel. Quindi la coordinata di inizio sarà sempre quella del palmo della mano mentre la coordinata di fine sarà quella del vettore contenente le distanze minime ricavato precedentemente. Nelle figure seguenti vediamo due esempi di binarizzazione incrementale. A sinistra la binarizzazione con valore di soglia iniziale scelta pari a 250 e a destra la binarizzazione con valore di soglia iniziale scelto pari a 210.

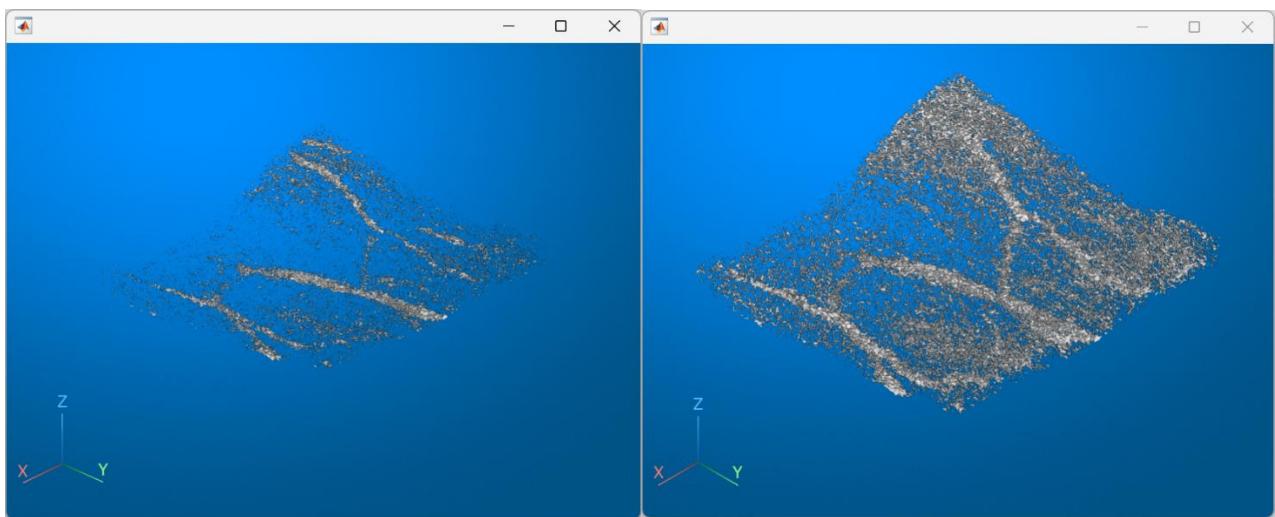


Figura 7.24 - Confronto tra due volumi binarizzati con soglia iniziale differente

Riassumendo, la binarizzazione incrementale si basa su 4 parametri:

1. Valore di soglia iniziale (da adattare)
2. Valore di soglia finale (sempre 255)
3. Coordinata Z di inizio (sempre quella del palmo)
4. Coordinata Z di fine (data dalla distanza minima vene-palmo)

La binarizzazione incrementale quindi consiste nel far variare il valore di soglia in un intervallo $[a,b]$ in cui a è la soglia iniziale e b è la soglia finale. Il valore quindi da determinare è a dato che b sarà sempre 255 perché arrivati alla fine vogliamo terminare la raccolta dei voxel.

7.5.4.2 Scelta del valore ottimo

Per calcolare il valore ottimo di soglia iniziale per ogni acquisizione seguiamo un ragionamento analogo a quanto detto per il valore di fine binarizzazione. Partendo da un valore di soglia iniziale di 250 andiamo a decrementarla fino a raggiungere 210, per un totale di 41 iterazioni. In particolare andiamo a contare il numero delle componenti connesse per ogni iterazione, salvando i valori in un vettore. Successivamente possiamo approssimarne la derivata attraverso la funzione `diff` di MATLAB. Plotando il vettore e la sua derivata notiamo che:

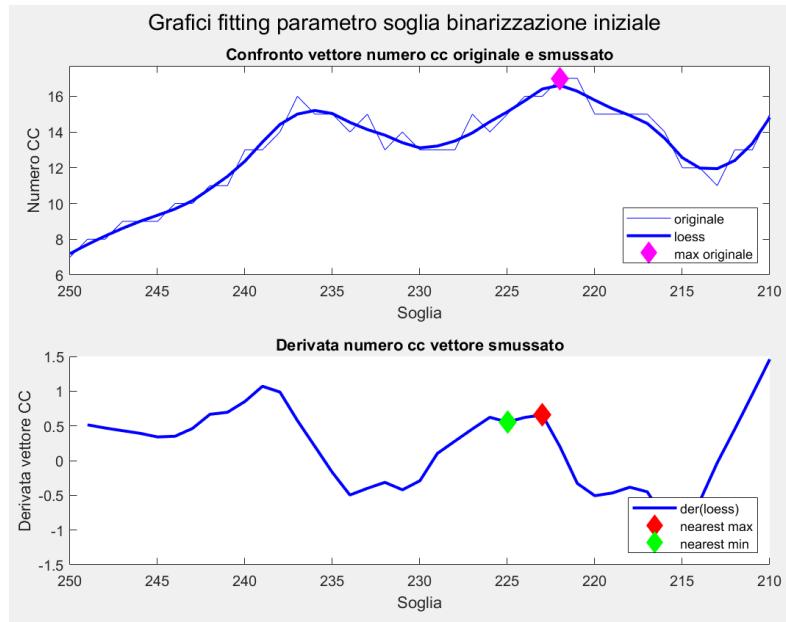


Figura 7.25 - Plot numero cc in funzione del valore di soglia iniziale e sua derivata

- Il numero delle componenti connesse cresce diminuendo la soglia
- Dopo ogni massimo c'è una zona di minimo locale

Anche in questo caso si può osservare che variando la soglia varia il numero di componenti connesse del volume. Al decrescere della soglia il numero delle componenti aumenta perché prendo in considerazione sempre più voxel. Con un valore alto (prime iterazioni) sto considerando una soglia troppo selettiva e quindi non raccolgo tutte le peculiarità del pattern venoso. Con una soglia troppo bassa (ultime iterazioni) considero tutte le vene ma queste saranno collegate con il rumore infatti non raggiungo il picco di componenti connesse.

Dato che l'obbiettivo della soglia iniziale è quello di recuperare le vene sottili in prossimità del palmo, bisogna trovare un compromesso nella variazione del numero delle cc che indicherà la comparsa delle vene superficiali.

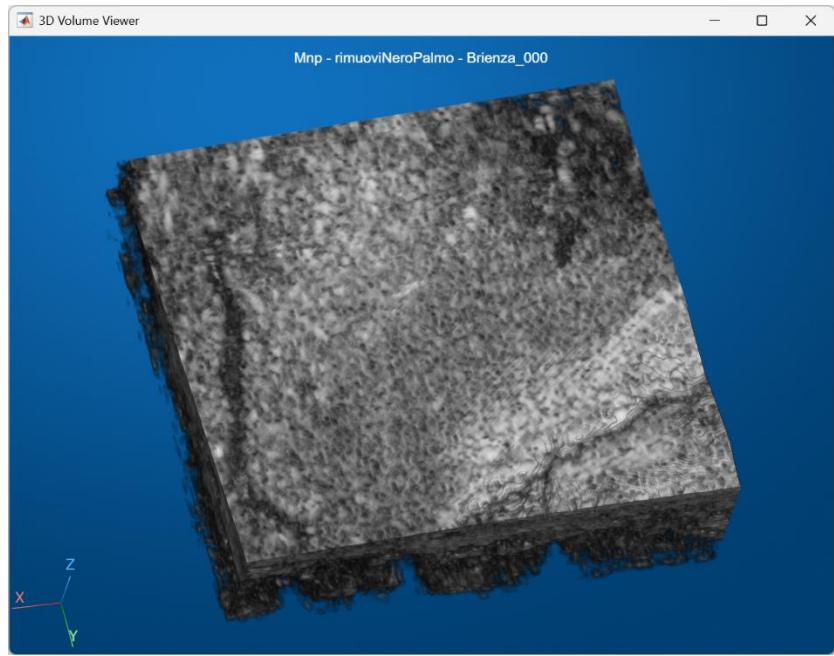


Figura 7.26 - Esempio di vene superficiali

Dato che vogliamo conservare quante più strutture possibili andiamo a posizionarci nella zona in corrispondenza della soglia a cui si ottiene il massimo numero delle componenti connesse. In seguito andiamo a selezionare un'intorno di questa zona in cui il numero delle cc si sia stabilizzato, condizione che va ad indicare la comparsa delle vene e il loro consolidamento prima che si vadano a connettere con il rumore. Quindi scegliere la soglia in questa zona significa conservare nel volume sia le vene sottili superficiali che quelle consistenti profonde.

In funzione di queste considerazioni si è scelto il valore di soglia come segue:

1. Trovo il valore della soglia in cui ho raggiunto il **massimo assoluto** del numero delle componenti connesse (marker color magenta di Figura 7.25)
2. Applico un operatore di smoothing al vettore contenente il numero delle cc per iterazione
3. Calcolo la derivata del vettore smussato contenenti il numero delle cc
4. Trovo gli indici dei massimi locali della derivata
5. Trovo gli indici dei minimi locali della derivata
6. Individuo la soglia in corrispondenza del **massimo** locale della derivata più vicina alla soglia trovata al punto 1 (marker color rosso di Figura 7.25)
7. Individuo la soglia in corrispondenza del **minimo** locale della derivata più vicina alla soglia trovata al punto 6 (marker color verde di Figura 7.25)
8. Imposto la mia soglia iniziale pari a quella individuata nel punto 7

7.5.5 Binarizzazione finale

Una volta ricavati i valori di fine binarizzazione e il valore di soglia iniziale si procedere alla binarizzazione finale del volume. La binarizzazione viene effettuata dalla funzione **binarizzaVolume** che dato in input le caratteristiche calcolate e i dati relativi al volume da binarizzare restituisce una matrice 3D logica contenente il volume binarizzato caratterizzato dalla presenza delle strutture venose.

La funzione va a binarizzare in modo incrementale ogni BSCAN utilizzando il valore di inizio binarizzazione pari agli indici del palmo della mano e come valore di fine binarizzazione il vettore di distanze calcolato al passo precedente. Infine la soglia iniziale sarà quella adattata di cui sopra.

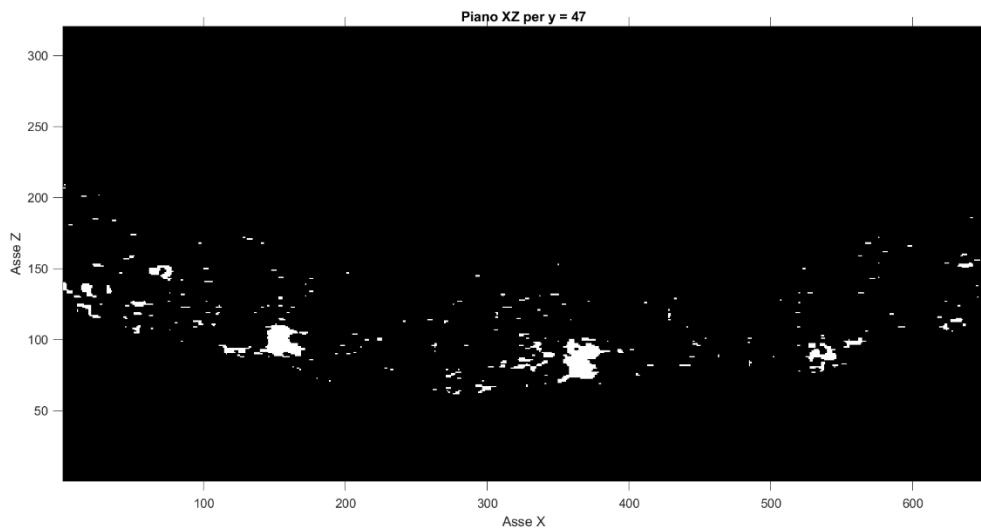


Figura 7.27 - Esempio di piano binarizzato finale in modo incrementale

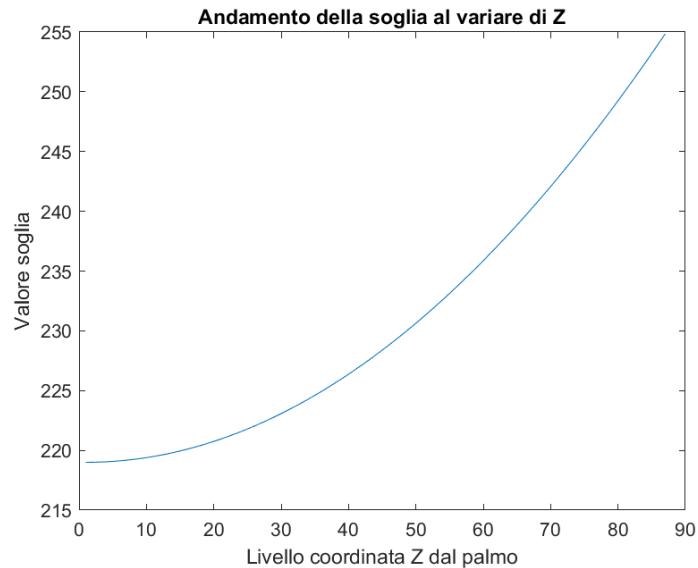


Figura 7.28 - Variazione del valore di soglia per le varie coordinate Z del BSCAN

$$soglia = sogliaIniziale + \left(\frac{z - 1}{fineBinarizzazione} \right)^2 (255 - sogliaIniziale)$$

Equazione 7.1 – Legge con cui varia la soglia

La binarizzazione quindi partirà dal palmo della mano e si fermerà arrivando al valore soglia di 255 in corrispondenza della minima distanza tra le vene e il palmo per il singolo piano XZ. Come riportato in Figura 7.28 la soglia cambierà ad ogni iterazione, sarà pari al valore della **sogliaIniziale** quando andremo a binarizzare la porzione di piano in corrispondenza degli indici Z in cui si trova il palmo e sarà pari a 255 una volta raggiunto il valore di fine binarizzazione.

Per migliorare la visualizzazione delle vene (in particolare quelle molto spesse) si è scelto di introdurre un termine di correzione chiamato **costante diametro vene** calcolato pari al diametro medio delle vene presenti nel volume binarizzato fase 1. Questo termine viene poi validato tramite una procedura iterativa analoga a quella discussa per la soglia iniziale in modo tale da determinare il valore ottimo per ogni specifico caso. Il termine costante va quindi ad aumentare il livello di fine binarizzazione, questo è necessario perché la fine della binarizzazione si basa sulla coordinata in cui si trova la sommità della vena. Quindi per raccogliere meglio la vena in tutto il suo diametro è buona norma estendere la fine al

livello in cui si trova la parte inferiore del vaso sanguigno. Rimane necessario validare la costante con una procedura iterativa dato che aumentando troppo il livello di fine si rischia di raccogliere artefatti che saranno poi difficili da eliminare, soprattutto (come vedremo) se si trovano al di sotto delle vene.

Vediamo alcuni esempi di volume binarizzato finale.

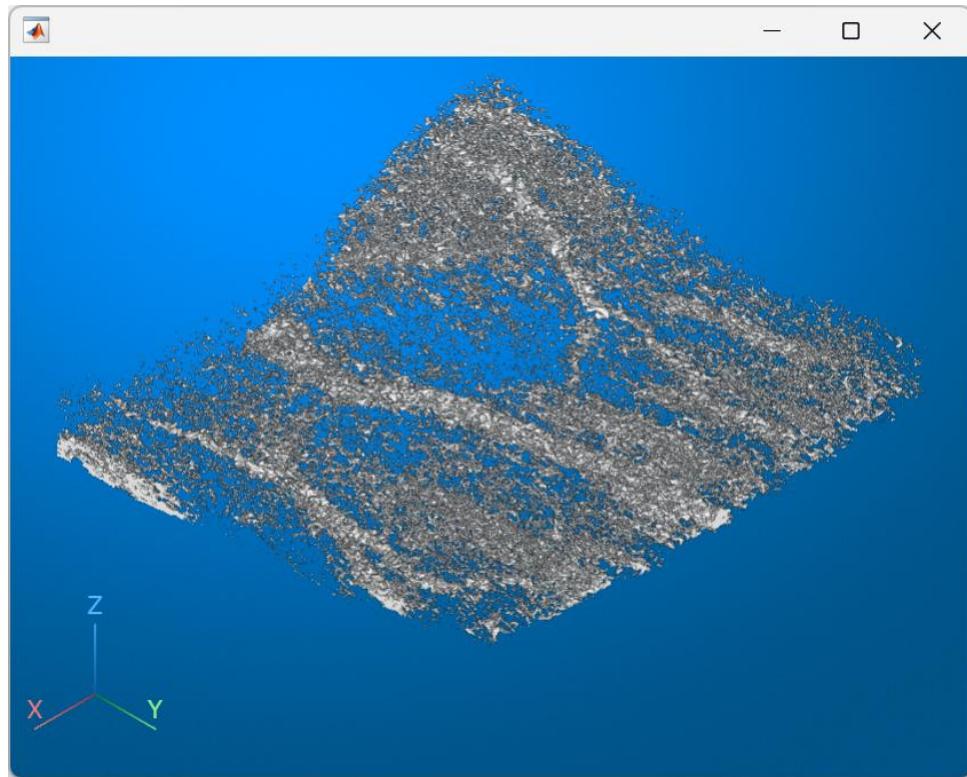


Figura 7.29 - Volume binarizzato finale per utente con vene sottili superficiali e vena principale profonda

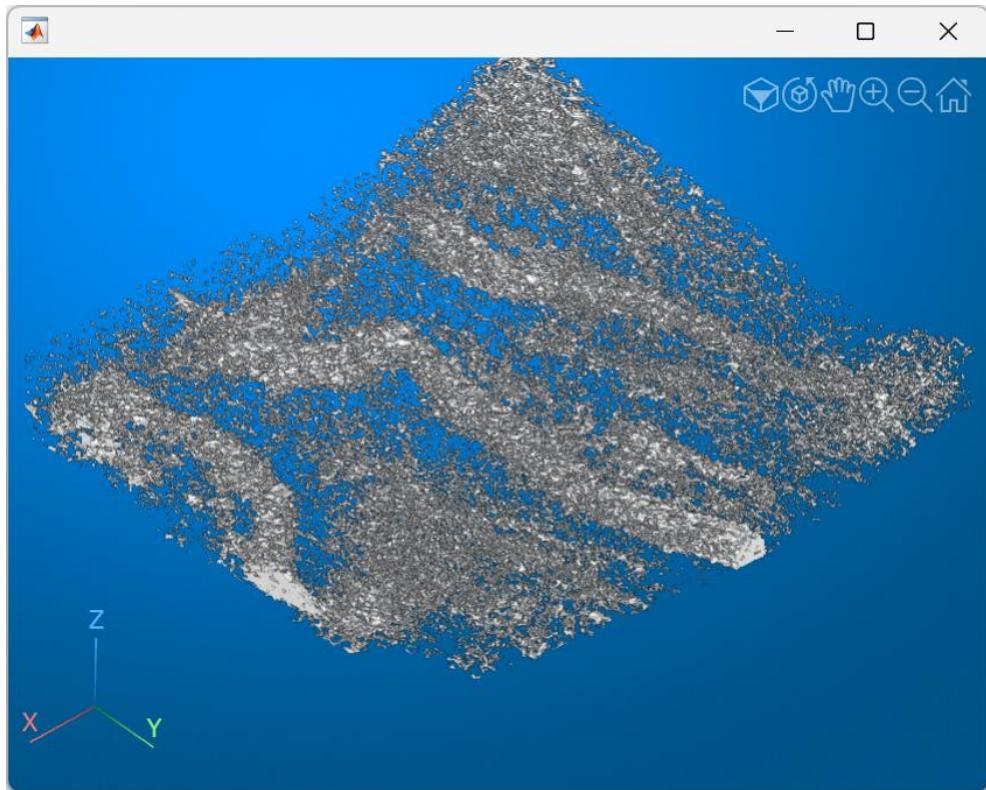


Figura 7.30 - Volume binarizzato finale per utente con vene molto corpose

7.6 OPERAZIONI MORFOLOGICHE

In questa fase viene richiamata la funzione `isolaPatternVenoso.m` che effettua una serie di filtri su componenti piccole e separate sul volume binarizzato `volumeBin`, oltre che operazioni morfologiche così da distinguere più chiaramente il pattern venoso dalle componenti di rumore.

```
volumeCleaned = bwareaopen(volume, 500, 6);
```

Viene utilizzata la funzione `bwareaopen`, che rimuove tutte le componenti connesse nel volume binarizzato di dimensione inferiore a 500 voxel assumendo che queste componenti siano rumore; **6** indica che due voxel sono considerati connessi se condividono una **faccia**. Questo significa che ogni voxel può essere collegato a un massimo di 6 voxel adiacenti. È il valore più restrittivo che permette la funzione '`bwareaopen`'.

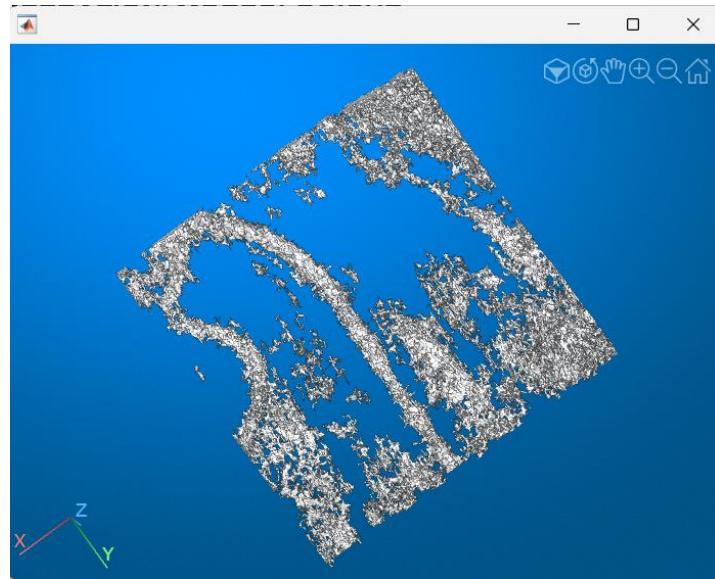


Figura 7.31 – Volume binarizzato filtrato

```
volumeOpen = imopen(volumeCleaned, strel('sphere', 1));
```

Dopo il primo filtraggio viene applicata un'operazione di **apertura morfologica** al volume `volumeCleaned`, usando come **elemento strutturante** una sfera di raggio 1 voxel. L'apertura consiste in due passaggi:

1. **Erosione:** Prima, vengono "corrosi" o ridotti i bordi degli oggetti nel volume, usando l'elemento strutturante.
2. **Diluizione:** Successivamente, i bordi vengono "espansi" o ripristinati, riportando gli oggetti alla loro dimensione originale, ma senza il rumore isolato e le piccole sporgenze.

L'elemento strutturante `strel('sphere', 1)` crea una sfera che include solo i voxel entro una distanza di 1 voxel dal centro, agendo come filtro per rimuovere dettagli piccoli o rumore. Questo processo lascia intatte le strutture più grandi, come le vene, ma le rende più omogenee e pulite, senza artefatti.

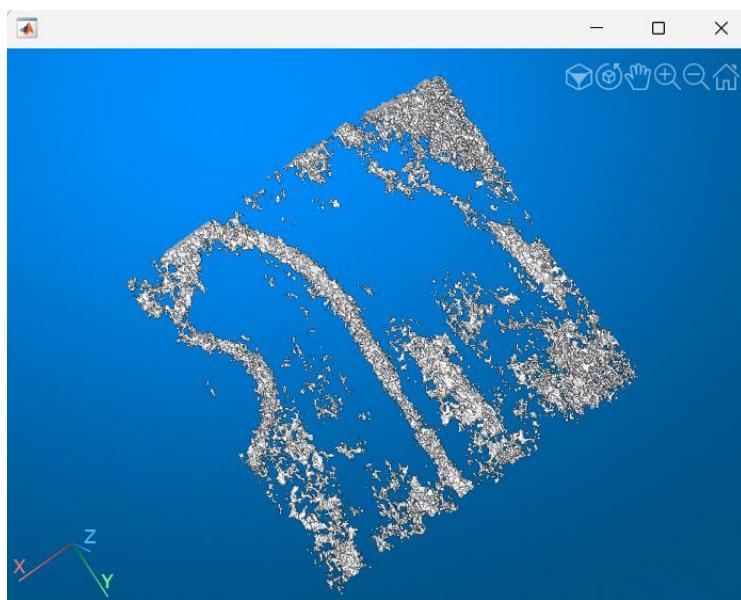


Figura 7.32 – Volume filtrato dopo apertura

```
volumeCleaned = bwareaopen(volumeOpen, 1000, 26);
```

Dopo l'operazione morfologica, viene nuovamente applicata **bwareaopen** per rimuovere componenti connesse di dimensioni inferiori a 1000 voxel. **26** indica che i voxel sono connessi se condividono una faccia, un bordo o un angolo, permettendo a ciascun voxel di avere fino a 26 vicini (la **connettività 26** è la più permissiva e permette di includere anche i voxel diagonali o angolari, quindi si usa quando si vuole che le componenti connesse siano considerate più "ampie" e includano tutti i possibili voxel vicini).

Questo ulteriore filtro rimuove eventuali frammenti di rumore residuo.

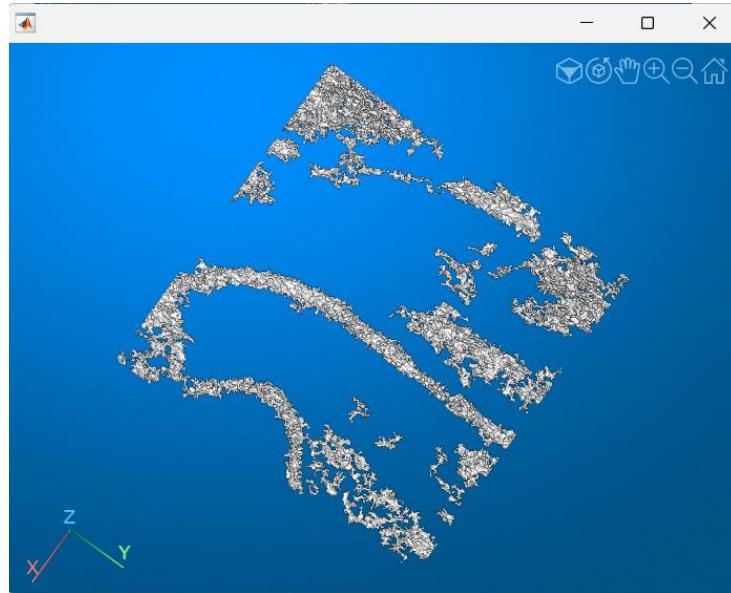


Figura 7.33 – Volume aperto dopo secondo filtraggio

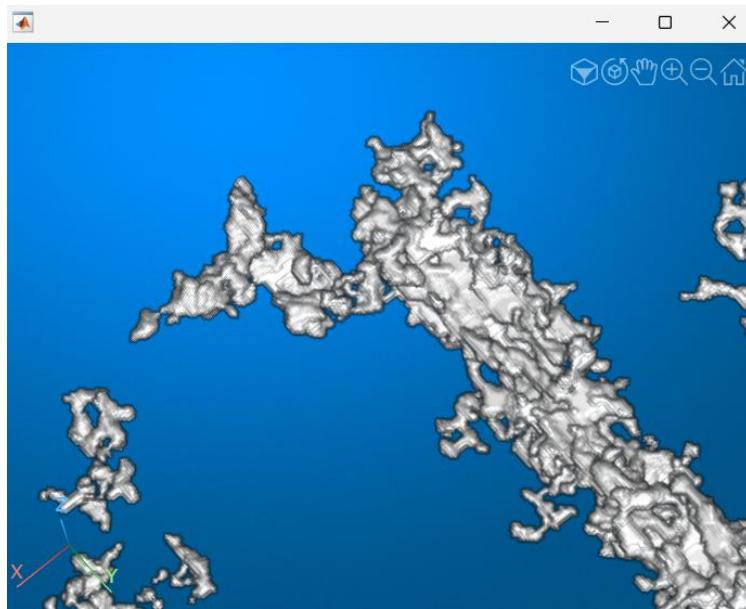


Figura 7.34 – Esempio connessione vena e sporco in output da isolaPatternVenoso.m

La funzione **inspesamento.m** si occupa di **rafforzare e rendere più spesse e piene** le vene nel volume, oltre che migliorare la separazione tra le strutture venose e il rumore, con l'obiettivo di **migliorare la loro visibilità** per facilitare la classificazione e l'analisi.

```
erodedVolume = imerode(volume, strel('sphere', 1));
```

Il primo passo inizia con **l'erosione** del volume usando una struttura sferica, riducendo il volume delle aree più piccole, eliminando il rumore e separando le strutture venose. In altre parole, l'erosione aiuta a "pulire" il volume rimuovendo le piccole connessioni che non appartengono alle vene principali.

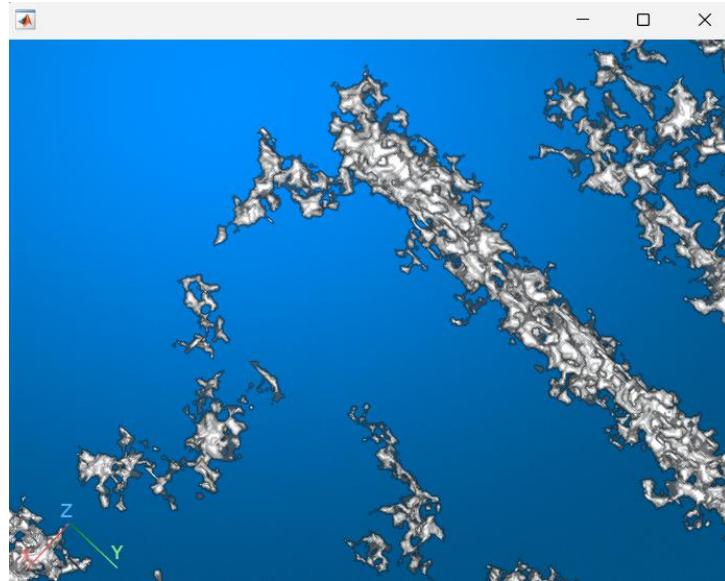


Figura 7.35 – Esempio connessione vena e sporco dopo erosione

```
volumeGauss = imgaussfilt3(erodedVolumeint, 1);
```

Dopo l'erosione, viene applicato un **filtro gaussiano 3D** (con `imgaussfilt3`) al volume per 'levigare' le immagini e connettere le componenti venose, quindi a ridurre eventuali artefatti e migliorare la loro visibilità, rendendo le strutture venose più continue e meno frastagliate.

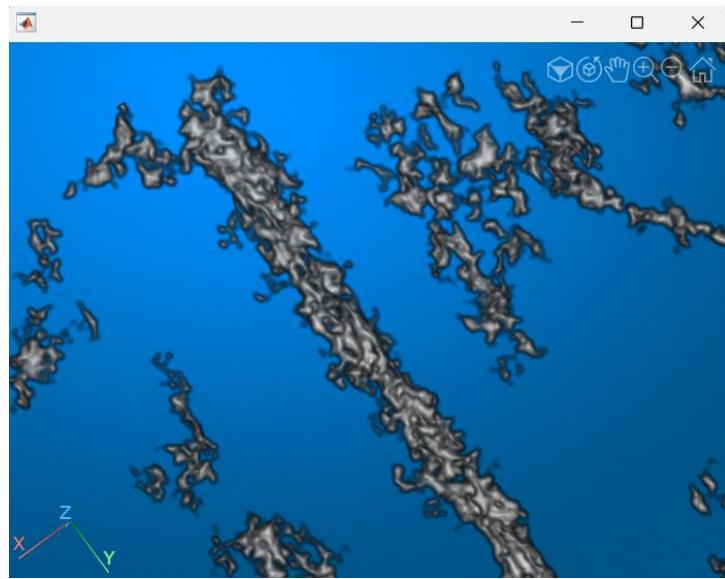


Figura 7.36 – Applicazione filtro gaussiano

```
volSpesso = binarizza(volumeGauss,'manuale','soglia');
```

Ora l'idea è quella di trovare una soglia che ottimizzi la separazione tra le vene e il rumore, minimizzando il numero di componenti non venose. Per farlo è stata usata una procedura iterativa che va a salvare il numero di componenti connesse che si ottengono con diversi valori di soglia.

Quindi provando diversi valori di soglia di binarizzazione che vanno da 1 a 250, per ogni iterazione contiamo il numero di componenti connesse del volume e salviamo questa informazione in un vettore d'appoggio. Una volta terminate le iterazioni possiamo approssimare la derivata di questo vettore attraverso la funzione **diff** di MATLAB e plottare i due grafici. In particolare nella figura che segue vediamo in alto il valore di soglia in funzione del numero delle cc e in basso la derivata del vettore a cui è stato applicato un operatore di smoothing.

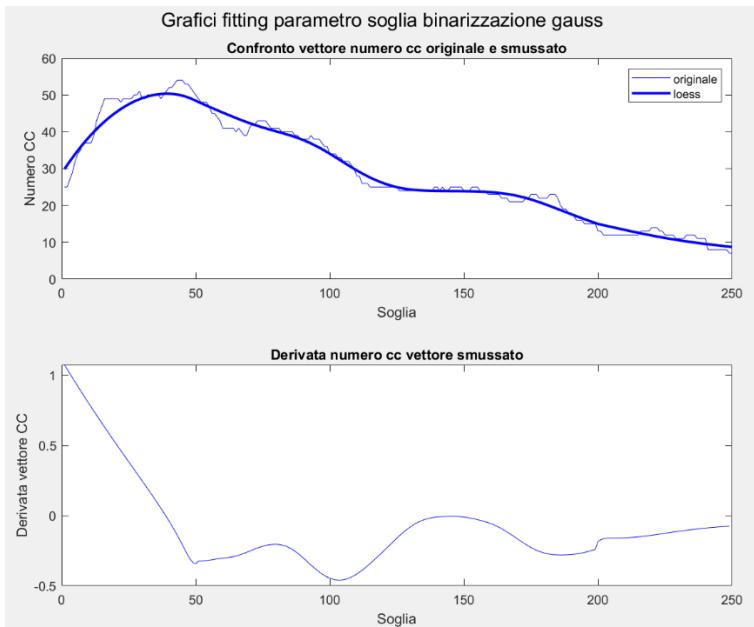


Figura 7.37 - Plot del vettore cc e sua derivata

A questo punto scegliamo come soglia l'indice in corrispondenza del primo massimo locale della derivata del vettore. Che nella figura di riferimento si vede essere tra 80 e 90.

Una volta che la soglia è calcolata, il volume viene binarizzato utilizzando questa soglia. L'output è un volume in cui le strutture venose sono chiaramente evidenziate e separate da connessioni con rumori.

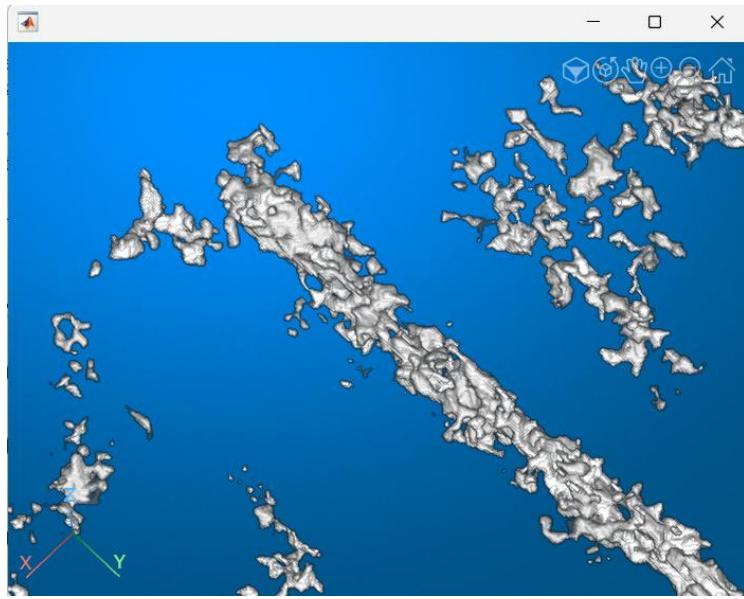


Figura 7.38 – volume binarizzato con soglia calcolata

Dopo aver eseguito tutti questi passaggi, il volume finale avrà vene più evidenti, più spesse, più coese e più facili da distinguere rispetto al rumore.

7.7 FILTRAGGIO

Lo scopo di questa fondamentale fase è quello di eliminare le componenti connesse che non sono delle vene e mantenere invece le vene. Per realizzare questo scopo si è pensato di addestrare un modello di Machine Learning, in particolare un classificatore binario per classificare le componenti connesse.

Come si vede nelle figure risultanti dagli step di **isolaPatternVenoso** ed **inspessimento** nel volume inspessito sono presenti varie componenti connesse come le vene ed altre strutture che non sono vene, considerate *rumore*.

7.7.1 Addestramento di un modello di classificazione binaria

Per addestrare un modello di classificazione è necessario realizzare un dataset e per realizzare un dataset sono a sua volta necessarie delle **features** (caratteristiche) da cui il modello deve imparare per fare le previsioni in base al valore delle etichette a loro assegnate.

7.7.1.1 Creazione delle features

Per insegnare al modello a riconoscere le vene da altre strutture si è pensato di creare delle features basate sulla geometria delle componenti connesse. Nello specifico le features verranno calcolate in base alle componenti connesse che si trovano nel volume risultato dalla fase di **inspessimento**. In particolare le features realizzate tengono conto delle 3 dimensioni lungo gli assi principali, delle coordinate dei centroidi, della superficie e di altre features messe a disposizione da **regionprops3**. Funzione presente in MATLAB che permette di calcolare proprietà geometriche e statistiche di regioni connesse in un volume 3D

Le features utilizzate si possono raggruppare in due classi:

- 1) Feature messe a disposizione da **regionprops3** e sono:
 - **Volume**: Volume della componente in voxel;

- **PrincipalAxisLength**: Lunghezze dei tre assi principali;
- **EquivDiameter**: Diametro di un cerchio con la stessa area della
- **Extent**: Rapporto tra il volume della componente e il volume del suo bounding box;
- **Solidity**: Rapporto tra il volume della componente e il volume del suo convesso;
- **SurfaceArea**: Superficie totale della componente;
- **Centroid**: Coordinate del centro di massa della componente.

2) Features derivate combinando o elaborando le feature fornite da regionprops3 e sono:

- **Rapporti tra le dimensioni principali (rapportoAssiYZ, rapportoAssiXZ, rapportoAssiXY)**: Valutano le dimensioni delle componenti connesse lungo i tre assi principali;
- **Rapporti tra volume e dimensioni principali (rapportoVolumeY, rapportoVolumeX, rapportoVolumeZ)**: Indicano come il volume si distribuisce lungo ciascuna dimensione principale;
- **Rapporti tra superficie e dimensioni principali (rapportoSurfaceYDim, rapportoSurfaceXDim, rapportoSurfaceZDim)**: Indicano la densità superficiale rispetto ai diversi assi;
- **Rapporto superficie Volume (rapportoSurfaceVolume)**: Misura utile per distinguere forme compatte da forme più allungate o frammentate;
- **Rapporto tra centroide e dimensioni (rappDimXCentX, rappDimYCentY, rappDimZCentZ)**: Analizzano la posizione relativa del centroide rispetto alle dimensioni principali;
- **Distanza Centroide-Pelle (DistCentroidPelle)**;
- **Rapporto Fine Binarizzazione e Distanza Centroide-Pelle (RappFineBinDistCP)**.

7.7.1.2 Etichettatura delle componenti

Una volta definite le features si può passare ad etichettare manualmente le cc in modo tale da creare il dataset definitivo. Per fare questo è stato realizzato uno script che si chiama **etichettaCC** il quale prende in input il volume restituito da inspessimento e crea il dataSet per quella singola acquisizione. È importante sottolineare che nella fase di etichettatura il volume è stato filtrato in base alla dimensione delle CC, in particolare sono state etichettate solo le componenti connesse che mantengono una dimensione del volume maggiore o uguale di 5000. Questo per evitare di far confondere il modello dato che strutture di dimensioni ridotte sono geometricamente simili quindi il rumore può essere confuso con le vene.

Una volta filtrate le componenti queste vengono iterate una ad una e tramite un'apposita interfaccia è possibile etichettare la componente corrente come vena o non vena. Questa funzione viene svolta dallo script **iteraCC**.

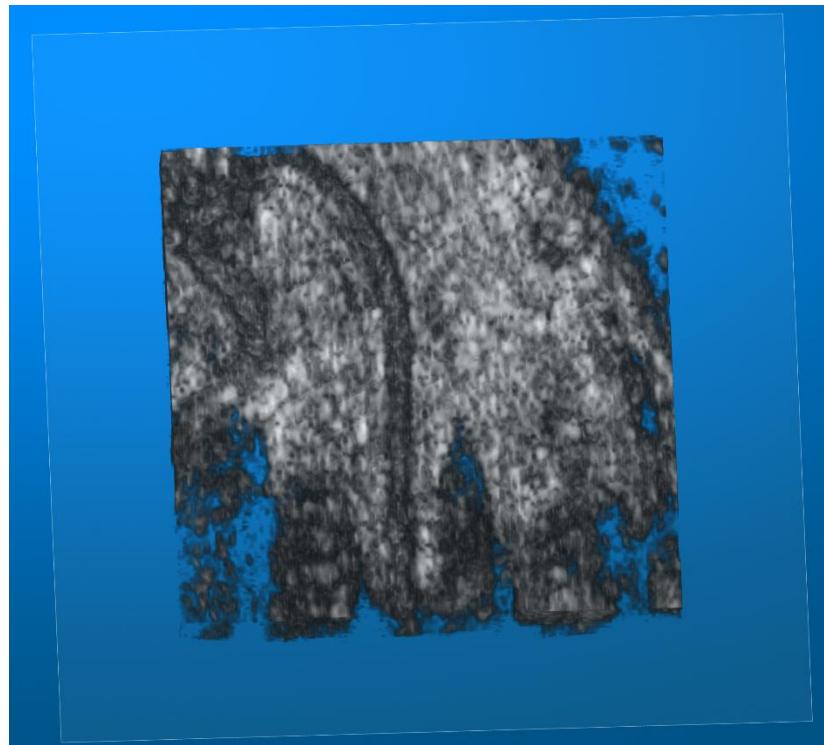


Figura 7.39 - Vene evidenziate nel volume di partenza

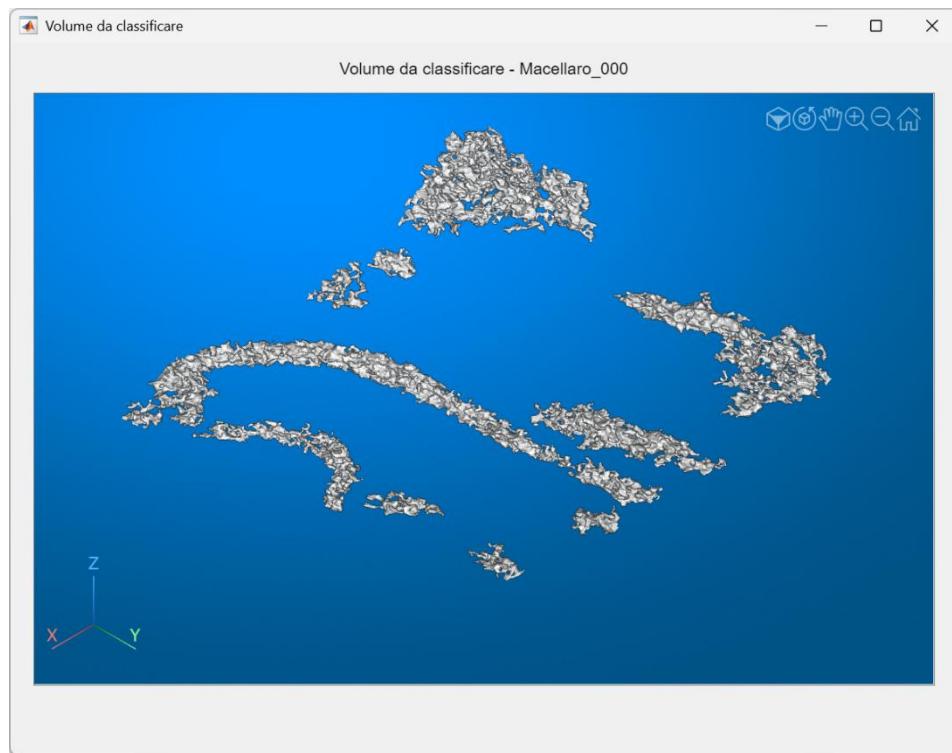


Figura 7.40 - Volume con le cc da etichettare

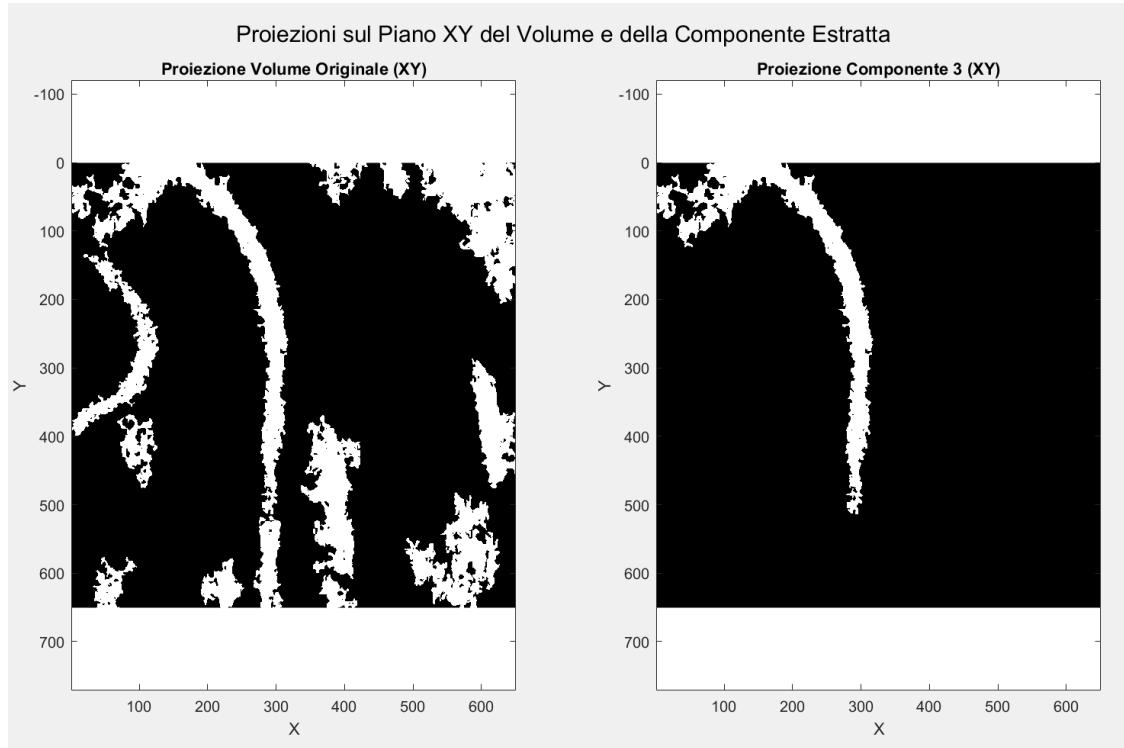


Figura 7.41 - Etichettatura della singola componente

Si è scelto di etichettare un valore che va da 2 a 4 acquisizioni per ogni utente su un totale di 43 utenti in modo tale da addestrare il classificatore a riconoscere quanti più casi possibili, evitando di ricadere nel problema dell'overfitting. Invece considerando molte acquisizioni per un unico utente il modello sarà meno accurato nel generalizzare su esempi mai visti, inoltre quegli utenti andrebbero scartati dal database perché presi interamente come dataset di training. Con poche acquisizioni per utente posso conservare nella procedura di matching tutti gli individui in modo tale da avere curve più realistiche.

Il dataset risultante possiede un totale di 16 features e 1544 esempi di training.

Figura 7.42 - Porzione del dataset realizzato

7.7.1.3 Addestramento

Una volta creato il dataset si può procedere con l'addestramento del modello, che viene effettuato nello script **addestraModello** il quale restituisce in output il modello con cui poter effettuare le

previsioni. Il dataset è stato diviso in due porzioni, 80% dati di training e 20% dati di test per verificare l'accuratezza. Il fitting è stato effettuato attraverso la funzione **`fitcensemble`** di MATLAB che permette di creare modelli di **ensemble** per la classificazione, combinando diversi **classificatori deboli** (come alberi di decisione) in un classificatore robusto e accurato. La potenza dei modelli ensemble risiede nella capacità di aggregare le previsioni di più classificatori, riducendo l'errore complessivo rispetto a singoli modelli.

Nello specifico il modello è stato realizzato attraverso un metodo di **bagging** che prevede di addestrare diversi alberi di decisione per ridurre il più possibile il problema dell'overfitting. Gli altri iperparametri sono stati ottimizzati automaticamente dalla funzione come:

- **NumLearningCycles** (Numero di alberi o cicli di apprendimento)
- **MinLeafSize** (Numero minimo di osservazioni per foglia)
- **LearnRate** (Tasso di apprendimento)

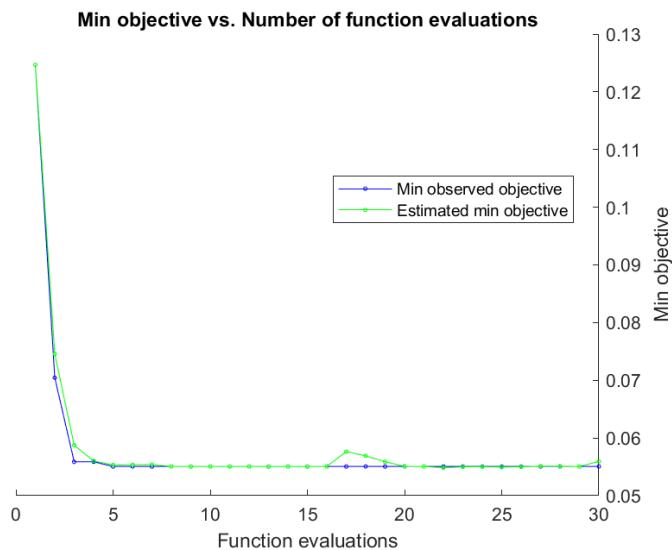


Figura 7.43 - Funzione di costo del modello

Una volta completato l'addestramento possiamo testare l'accuracy del modello sui dati di testi, in particolare risulta un accuracy del **93.85%** con la seguente matrice di confusione:

Matrice di Confusione:

204	5
14	86

Accuratezza: 93.85%

Figura 7.44 - Statistiche del modello addestrato

Risultano quindi

- 204 TN (True Negative)
- 86 TP (True Positive)
- 14 FP (False positive)
- 5 FN (False negative)

7.7.2 Filtraggio ricorsivo

Una volta addestrato il modello è possibile utilizzarlo per fare le previsioni e quindi eliminare tutto il rumore inutile ai fine della creazione del template. Prendiamo per esempio il volume in Figura 7.40 e applichiamo il filtraggio passando le features delle cc al modello addestrato per farne le previsioni, il risultato sarà il seguente:

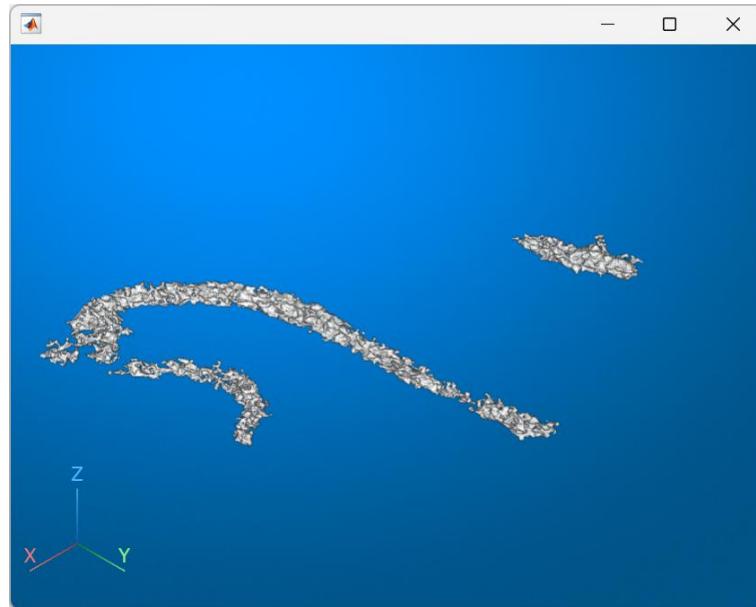


Figura 7.45 - Volume filtrato

Come si può vedere tutte le componenti che non erano vene sono state correttamente eliminate quindi in questo caso il modello ha svolto egregiamente il suo lavoro.

Ora consideriamo un altro utente ad esempio il seguente:

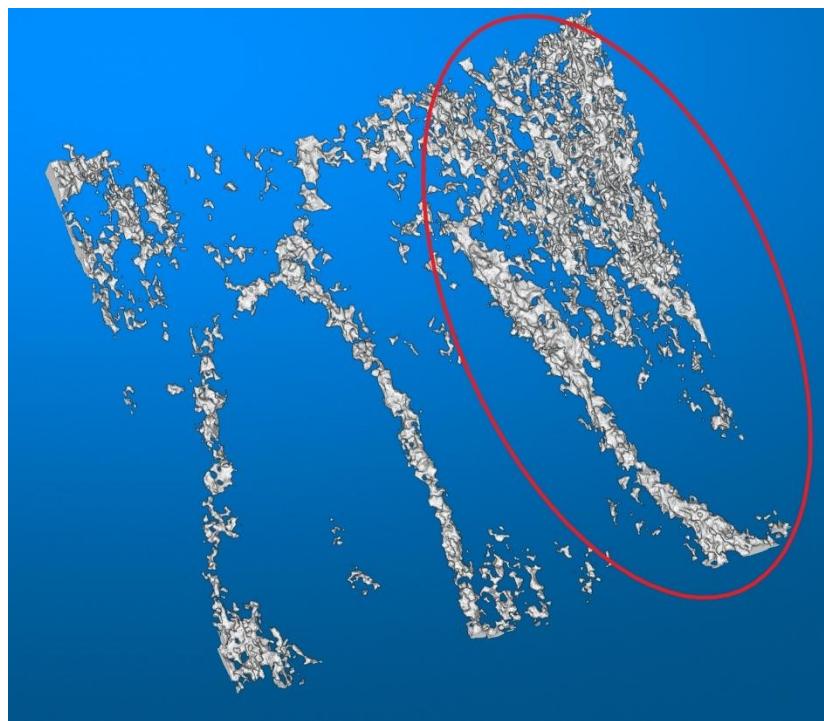


Figura 7.46- Esempio di utente con vena collegata al rumore

Come si nota dal volume inspessito, per questa specifica acquisizione, il metodo per separare e riconnettere le cc non ha funzionato a dovere. Questo a causa della forte connessione del rumore con la vena, come evidenziato in rosso. A questo punto passando la suddetta componente al classificatore possono accadere due casi:

1. Il classificatore predice 1, quindi la cc viene etichettata come vena e non viene persa nel filtraggio.
2. Il classificatore predice 0, la cc viene etichettata come rumore e viene eliminata.

È necessario gestire questi due casi dato che il modello non è stato addestrato su cc contenenti vene collegate al rumore non possiamo sapere a priori quale sarà il suo comportamento. Nell'attuale fase di filtraggio andremo a gestire il caso 1, la gestione del caso 2 verrà discussa nel capitolo successivo.

Quindi per gestire un eventuale vena eliminata ‘erroneamente’ dal filtraggio si è pensato di adottare un filtraggio **ricorsivo**. Questo significa che la procedura di filtraggio sarà una funzione ricorsiva che chiamerà sé stessa finché successive procedure di erosione e connessione applicata alla singola componente vanno a separare il rumore dalla vena. Una volta separate possono essere correttamente classificate dal modello.

La procedura che effettua il filtraggio ricorsivo, denominata **filtraCC**, prende in input il volume da filtrare, il modello e il parametro **stopRicorsione** che regola la fine della ricorsione. Il funzionamento è il seguente:

1. Creo le features della singola componenente
2. Effettuo la previsione con il modello
 - a. Il modello predice 1: la cc è stata classificata come vena, passo a quella successiva.
 - b. Il modello predice 0: la cc è stata classificata come rumore. Bisogna verificare se è il caso di effettuare un recupero.
 - i. Se la dimensione del volume è maggiore di 20000 e la cc è stata classificata come rumore si passa all'esecuzione della funzione **recuperaVena**

7.7.2.1 Recupero vena

La funzione **recuperaVena** prende in input la cc classificata come rumore e il parametro **stopRicorsione** (parametro che viene decrementato ad ogni esecuzione di **filtraCC**).

All'inizio della funzione viene inserito il codice per bloccare la ricorsione, quindi se **stopRicorsione** è negativo la funzione restituisce un volume vuoto. Nel caso contrario viene eseguita la procedura **inspessimentoRecuperoAdatt** sulla cc in questione. La funzione segue quanto illustrato per inspessimento, quindi va ad effettuare due operazioni morfologiche che sono erosione e filtraggio di gauss per poi restituire il volume filtrato da gauss binarizzato secondo una soglia ottimale. A differenza di inspessimento per ottenere dei tempi minori la soglia viene restituita appena la procedura iterativa trova un massimo nella derivata del vettore contenenti il numero delle cc per iterazione.

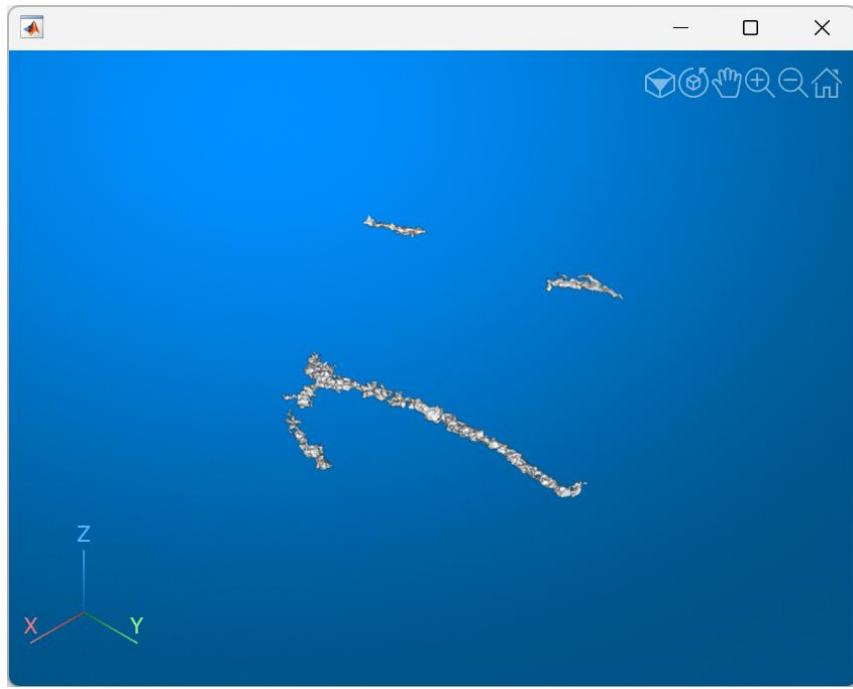


Figura 7.47 - Risultato della classificazione su cc con vena collegata al rumore

Come si evince dalla Figura 7.47 il modello ha classificato la cc evidenziata in rosso di Figura 7.46 come rumore eliminandola dal volume. Questo è un perfetto esempio di caso in cui è necessaria la procedura di ripristino.

Andando ad abilitare la ricorsione nella funzione `filtraCC` (passando un valore di `stopRicorsione > 0`) la vena si dovrebbe separare dal rumore consentendo al modello di identificare correttamente le due componenti. Esaminiamo meglio la componente incriminata.

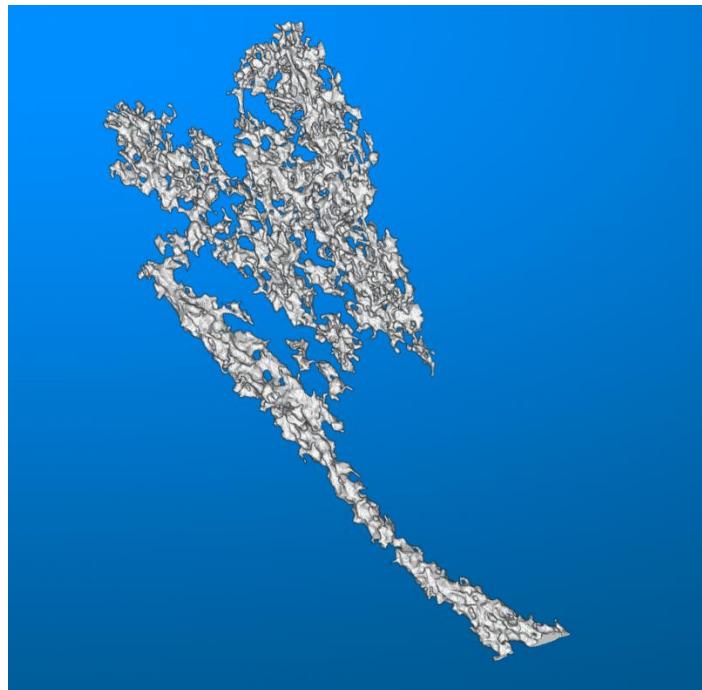


Figura 7.48 - Componente connessa contenente una vena collegata al rumore

Andando ad applicare ricorsivamente la procedura si nota come i collegamenti tra il rumore e la vena iniziano a disgregarsi e il classificatore dopo un massimo di 3 chiamate è in grado di individuare

correttamente la vena. Nel caso dell'esempio riportato dopo una chiamata ricorsiva otteniamo la vena correttamente separata dal rumore come illustrato in Figura 7.49



Figura 7.49 - Risultato `filtraCC` con ricorsione abilitata

Successivamente a `filtraCC` viene effettuata un'operazione di connessione per riconnettere eventuali frammenti di vena che si sono disgregati e viene applicato il filtraggio (non ricorsivo) anche alle componenti con `volumeSize < 5000`. Ne risulta che il volume filtrato finale è il seguente:



Figura 7.50 - Risultato del volume filtrato finale

7.8 AFFINAMENTO

In questo capitolo andremo ad illustrare l'ultima procedura che viene applicata al volume contenente il pattern venoso. La procedura di affinamento consiste nello smussare le strutture rimaste dopo il filtraggio in modo tale da separare eventuali artefatti dalle vene e rendere la visualizzazione del

template più omogena. Inoltre ha lo scopo di gestire il caso 2 come spiegato in precedenza. Prendiamo un utente di esempio e seguiamo la procedura step by step.

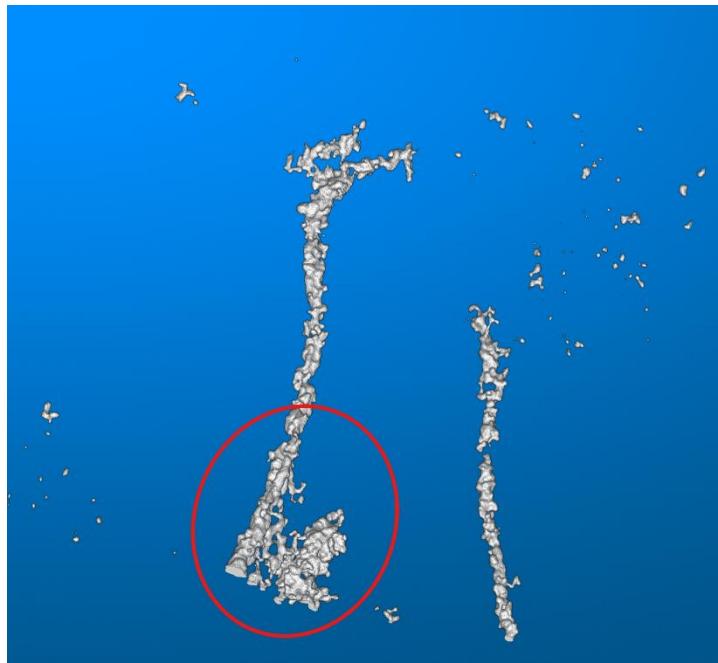


Figura 7.51 - Esempio di risultato proveniente da filtraCC contenente del rumore

Considerando la Figura 7.51 , notiamo come il volume filtrato abbia conservato la vena ma questa presenta del rumore collegato ad essa. In questo caso è fondamentale applicare una procedura di affinamento per scollegare le componenti ed ottenere un template finale il più pulito possibile.

La procedura di affinamento viene realizzata dalla funzione **affinaVene** che prende in input il volume filtrato. La procedura consiste nello scansionare una ad una tutta le cc del volume e applicare ad ognuna di esso la funzione **affinaDisturbi**.

7.8.1 Affina disturbi

Questa funzione va ad iterare tutti i BSCAN della componente e individua **n** gruppi di cluster contenenti cc appartenenti a un singolo ramo della vena. Questo perché la vena può essere biforcuta e quindi avere più rami (**n** = numero di rami). Il codice segue questi step:

1. Scansioni ogni BSCAN
2. Calcolo la matrice di distanza
 - a. Conserva le distanza tra ogni coppia di centroide (i,j)
3. Trovo un valore di distanza per indicare la presenza di più rami
 - a. La soglia è calcolata come il diametro massimo delle cc moltiplicato per 2
4. Trovo le cc vicine tra loro
 - a. Due componenti sono considerate vicine se la loro distanza è inferiore alla soglia e diversa da zero.
5. Costruzione del grafo delle vicinanze
 - a. I nodi rappresentano le componenti connesse e gli archi rappresentano la loro vicinanza.
6. Trovo le componenti connesse del grafo
 - a. Gruppi di componenti connesse tra loro tramite archi nel grafo
7. Per ogni gruppo trovato effettuo delle operazioni morfologiche

Vediamo come funziona nel caso di vena biforcuta.



Figura 7.52 - Esempio di cc rappresentante una vena biforcuta

Ottengo i cluster per ogni BSCAN appartenenti a rami differenti

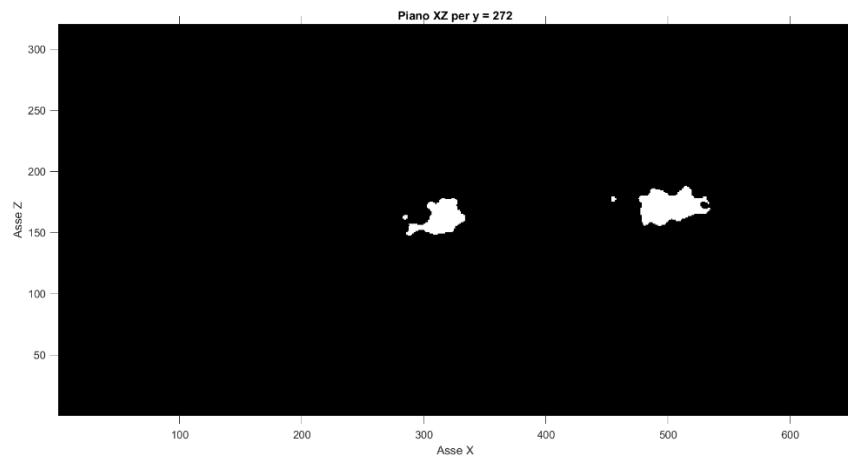


Figura 7.53 - Individuazione di 2 gruppi di cluster appartenenti a due rami differenti

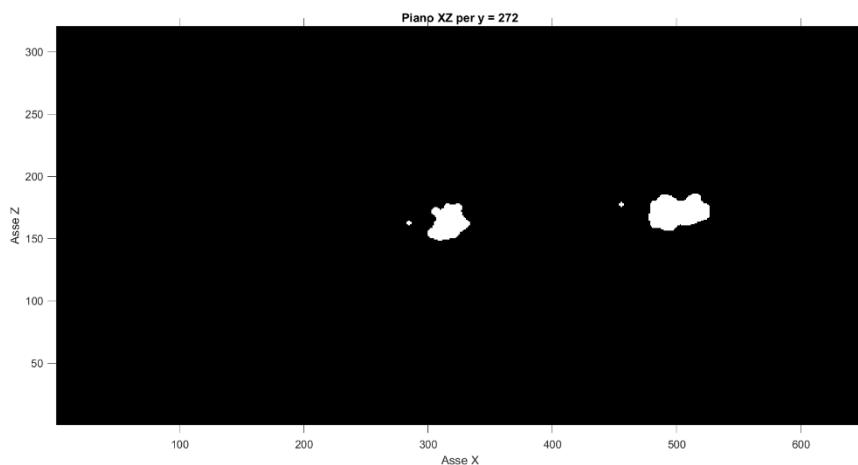


Figura 7.54 - Risultato operazioni morfologiche a più gruppi di cluster

Il risultato finale sulla vena biforcuta è rappresentato nella figura che segue

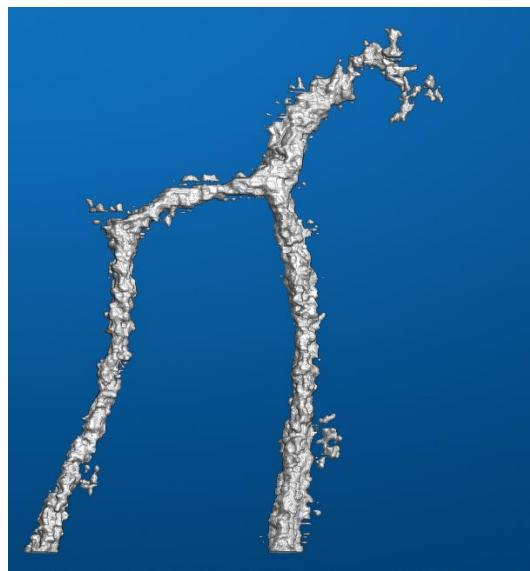


Figura 7.55 - Risultato affinamento vena biforcuta

Torniamo al caso in cui è presente una vena connessa con il rumore.

Riassumendo, per ogni cluster si individua la cc con il diametro maggiore (che si assume essere la vena) e si applica un'operazione di apertura con una geometria a disco di diametro calcolato in base al diametro della cc più grande. A questo punto si trasferisce il piano filtrato e affinato in un nuovo volume che prende il nome di **volumeAffinato**. Questa procedura ha lo scopo di rendere la vena di forma più circolare e rompere le connessioni che si presentano tra vena e rumore. Vediamo il risultato della procedura applicata sulla cc evidenziata di Figura 7.51



Figura 7.56 - Risultato procedura affinaRumore

Guardando attentamente la figura che precede possiamo osservare come la componente di rumore è stata correttamente staccata dalla vena lasciando il pattern venoso perfettamente isolato. Giunti a questo step è ora possibile eseguire su questo volume separato una nuova classificazione per lasciare solo la vena.

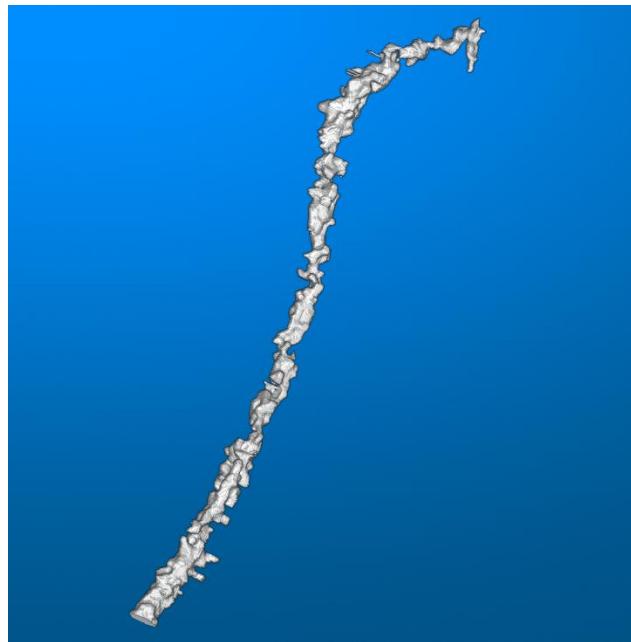


Figura 7.57 - Risultato filtraggio del volume affinato

Come possiamo vedere dalla applicando il filtraggio sul volume affinato la vena e il rumore vengono classificati correttamente dal modello ottenendo la struttura candidata a diventare il nostro template.

7.8.1.1 Altri casi possibili

Le possibilità che incorrono eseguendo **affinaVene** solo innumerevoli data l'unicità del pattern venoso di ogni utente. Questo significa che nel caso 2 possono ricadere tanti sotto casi che se non gestiti portano all'estrazione di un template fallato. In questo sottocapitolo andremo a spiegare in sommi capì come sono stati gestiti questi sotto casi con alcuni esempi visivi.

La procedura **affinaVene** è in grado sì di separare le vene dal rumore ma in qualche caso particolare si potrebbe verificare anche una separazione delle vene. Questo accade nel caso di una vena biforcuta in cui il ramo principale ha un diametro molto maggiore del ramo secondario come illustrato nella figura che segue.



Figura 7.58 - Esempio di vena biforcuta con rami di dimensioni di diametri diversi

Vediamo cosa accade andando ad applicare affinaVene su questa specifica cc.

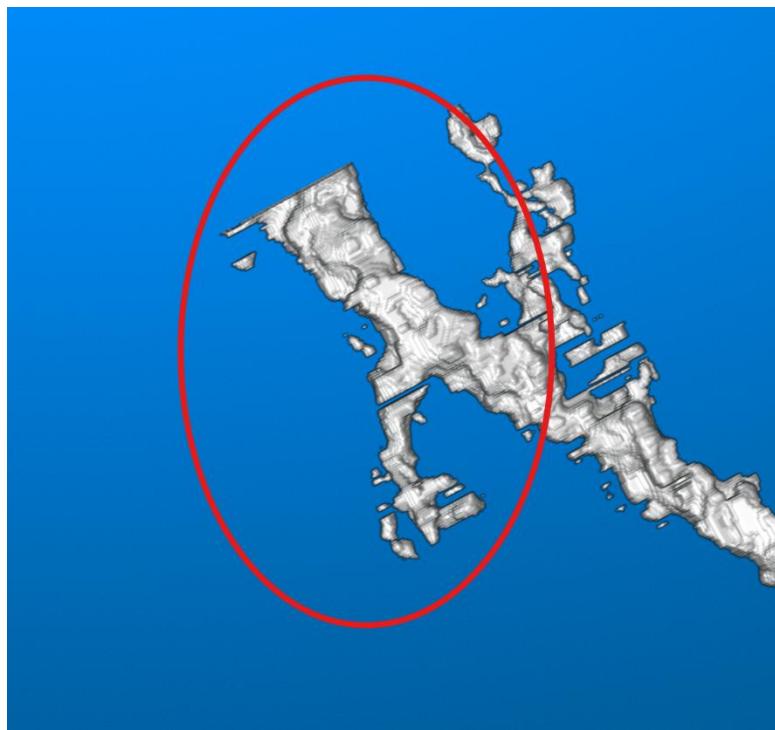


Figura 7.59 - Risultato della procedura di affinamento sul volume della figura che precede

Come si vede dalla figura soprastante il pezzo di vena sulla sx è stato scollegato dal ramo principale. A questo punto andando ad applicare il filtraggio su questa componente possono accadere due casi:

1. Il pezzo di vena viene classificato come vena. Non ci sono ulteriori passaggi da fare.
2. Il pezzo di vena viene classificato come rumore. In questo caso la cc in questione deve essere recuperata perché fondamentale alla creazione di un template più completo possibile.

Ma come faccio a capire se la cc che è stata eliminata mi serve o meno ai fini della realizzazione del template?

Il ragionamento che è stato seguito è il seguente:

- Osservando la Figura 7.58 si nota come il pezzo incriminato sia in realtà una porzione di un ramo molto più lungo.
- Alla luce di queste osservazioni il pezzo sarebbe in grado di collegarmi due vene quindi passare da 3 componenti connesse a 1.
- Bisogna realizzare una procedura che controlli se la porzione di vena che è stata eliminata dal classificatore è in realtà una componente connessa che mi serve nella creazione di un template completo.

Per procedere con quanto affermato sopra si è deciso di realizzare la funzione **recuperaPezzi**. Questa funzione consiste nel controllare uno ad uno i pezzi eliminati per verificare se questi collegano delle vene tra loro. Il codice della funzione è abbastanza lungo quindi per una comprensione dettagliata si invita il lettore a visionare il sorgente. In questa tesina ci limiteremo alle considerazioni appena fatte.

Il risultato della funzione **recuperaPezzi** è un volume contenente le cc che servono alla realizzazione del template.

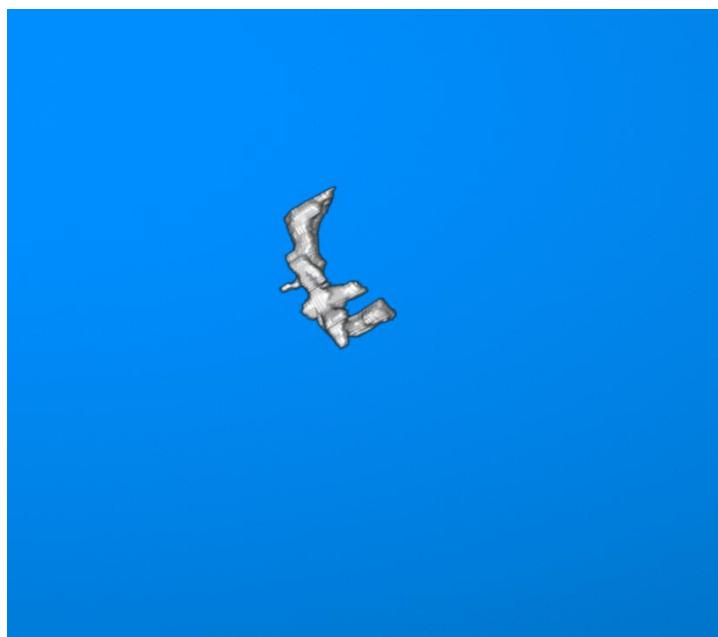


Figura 7.60 - Risultato funzione recuperapZetti

Quindi combinando il volume affinato con il volume recuperato ed applicando ad esso delle operazioni morfologiche otteniamo il volume affinato finale:



Figura 7.61 - Volume affinato finale

Si sottolinea come il risultato delle due procedure sia il volume perfettamente isolato e contenente solo la vena ricomposto di tutte le sue parti.

7.8.2 Elimina rumore

In questa ulteriore fase di filtraggio vengono mostrati i limiti di un approccio di filtraggio basato su un modello di Machine Learning. Per quanto il modello sia bravo ad imparare ed esente da problemi come l'overfitting e l'underfitting questo non può essere perfetto e i suoi limiti derivano principalmente da quanto le features sono rappresentative della realtà. Le features descritte nel paragrafo 7.7.1.1 si basano sulla distinzione delle cc in base alle proprie peculiarità geometriche, ma cosa succede se una componente di rumore è geometricamente molto simile al rumore? In questo caso è molto probabile che il modello si confonda nel classificare il rumore e quindi si possono generare dei FP (false positive).

Prendiamo come esempio la seguente acquisizione:

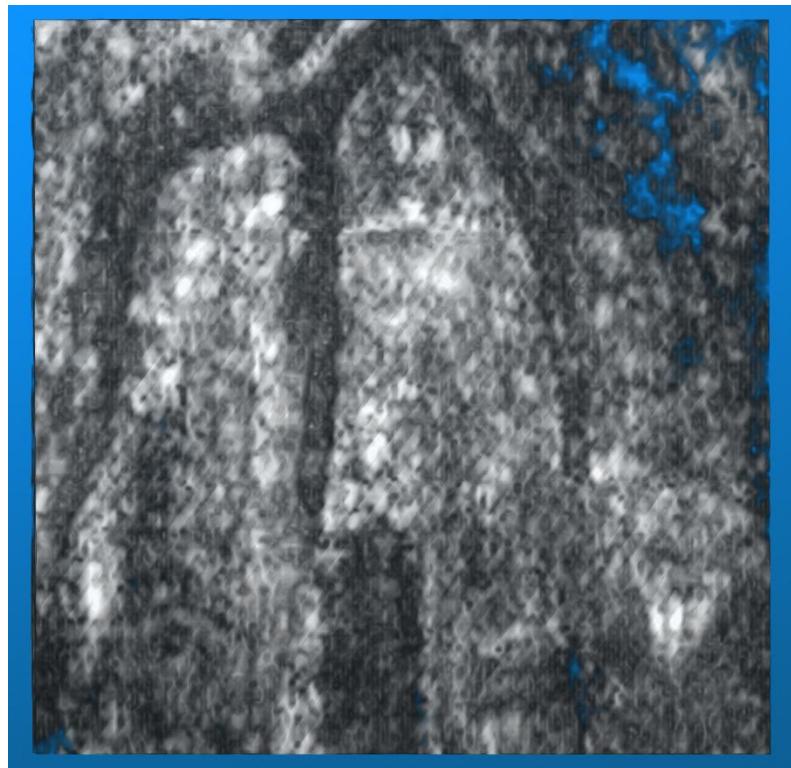


Figura 7.62 - CSCAN alla profondità delle vene

Notiamo come sia ben visibile una vena con 3 diramazioni. Proviamo a vedere cosa troviamo ad una profondità maggiore.

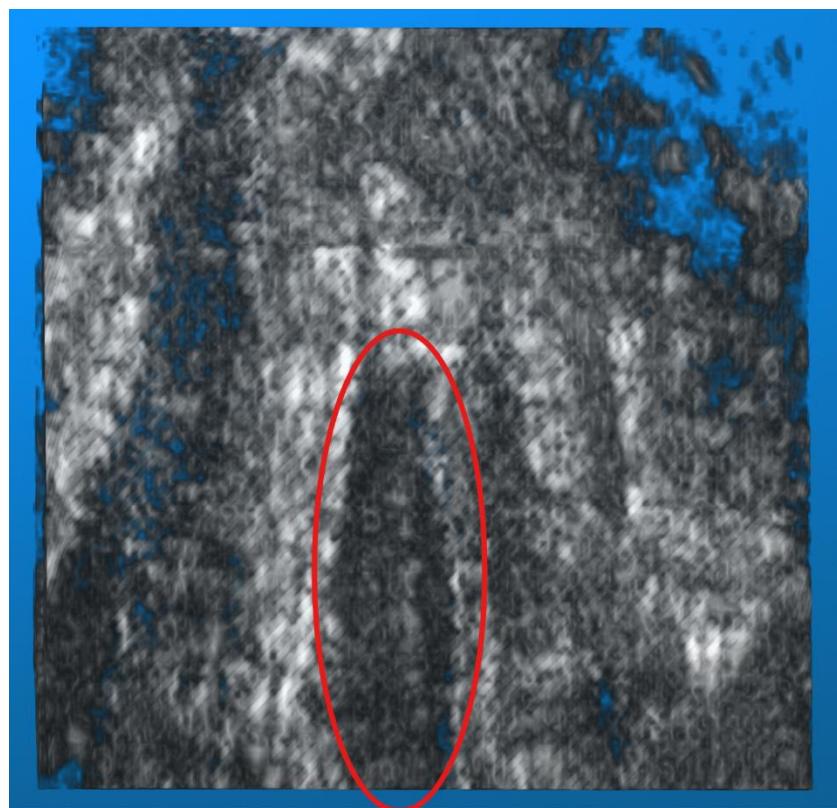


Figura 7.63 - CSCAN in cui si evidenzia del rumore simile a una vena

Dalla Figura 7.63 si evidenzia come proprio sotto la vena ci sia del rumore che assume la forma di una vena vera e propria. Questo accade perché la binarizzazione termina appena sotto la vena e quindi prende in considerazione anche parte della superficie del rumore. La parte superficiale assomiglia molto alla vena ma si capisce che non si tratta di una struttura venosa dal fatto che i voxel continuano ad avere intensità molto basse per tutta la profondità dell'acquisizione e quindi si tratta della sommità del rumore (nero) profondo.

A questo punto vediamo il volume inspessito e poi filtrato.

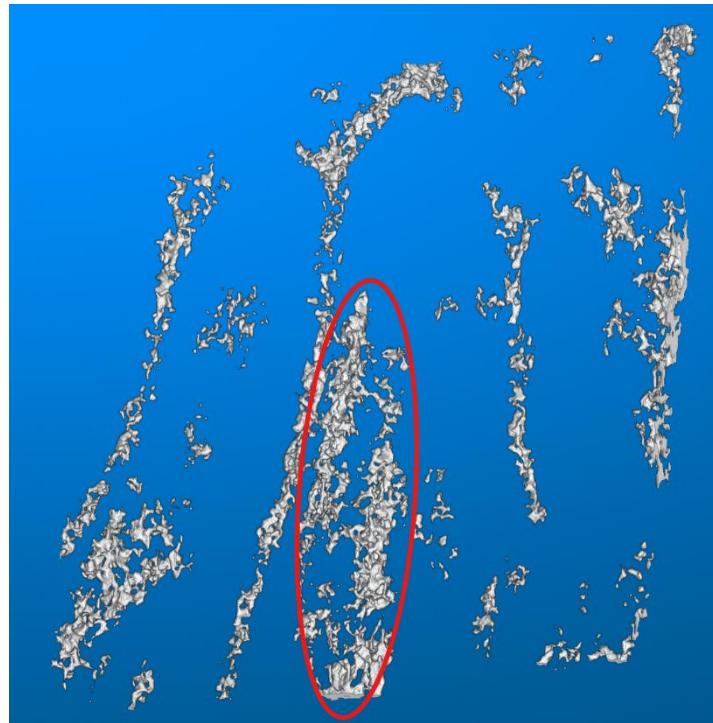


Figura 7.64 - Volume inspessito della figura precedente

Dal volume inspessito si conferma quanto affermato prima. È possibile notare la sommità del rumore che si trova sotto la vena e che è stato recuperato dalla binarizzazione.

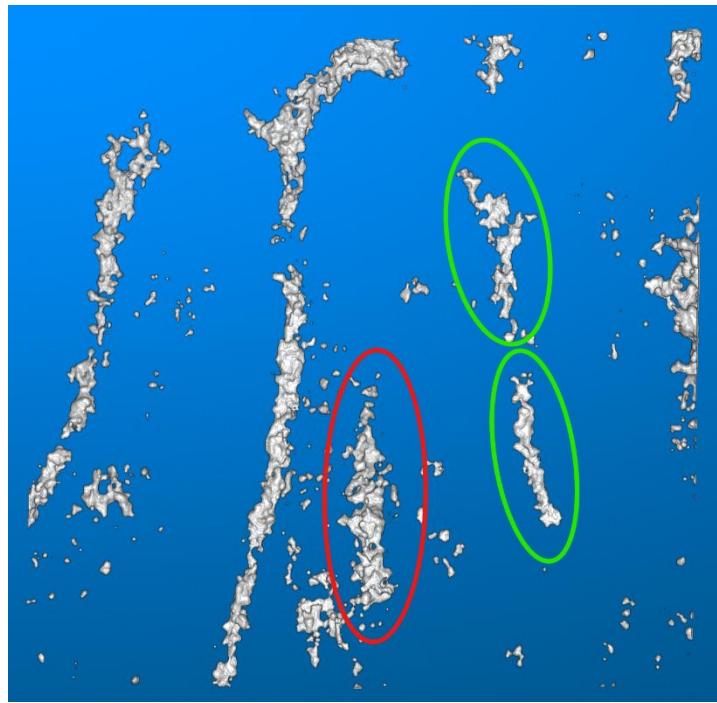


Figura 7.65 - Volume filtrato con del rumore simile alle vene presente

Applicando il classificatore emergono i limiti appena discussi. Come rappresentato in Figura 7.65 emerge che il classificatore abbia erroneamente etichettato il rumore sottostante come vena, quindi siamo nel caso di un falso positivo. D'altronde il rumore (evidenziato in **rosso**) è geometricamente molto simile ad una vena, si può anche notare il confronto con il ramo di destra (gruppo di vene evidenziate in **verde**), che visto così è indistinguibile dal rumore.

Per risolvere queste situazioni è stata progettata la funzione **eliminaRumore**. Funzione che prende in input il volume affinato e cerca di rimuovere il rumore che rimane dalla fase di classificazione e da quella di affinamento. Elimina rumore effettua un'analisi dei dati delle componenti rimaste cercando di capire dalla distribuzione dei dati se è presente qualche outlier da eliminare, come nel caso sopra descritto. Quindi valuta elementi come:

1. Distanza tra il centroide del piano XZ e la relativa coordinata del palmo, per ogni BSCAN della componente
2. Statistiche quali media, mediana, varianza, ecc... della feature di cui sopra
3. Pendenza media della vena
4. Numero di piani XZ della componente
5. Numero di piano XZ collegati con il bordo del volume.

Analizzando questi dati e stabilendo delle soglie adattive per i singoli casi è possibile distinguere dei possibili outlier rimasti. La funzione è stata inserita subito dopo il completamento di **affinaDisturbi** e un'attimo prima di **recuperaPezzi**, esaminiamone il risultato.

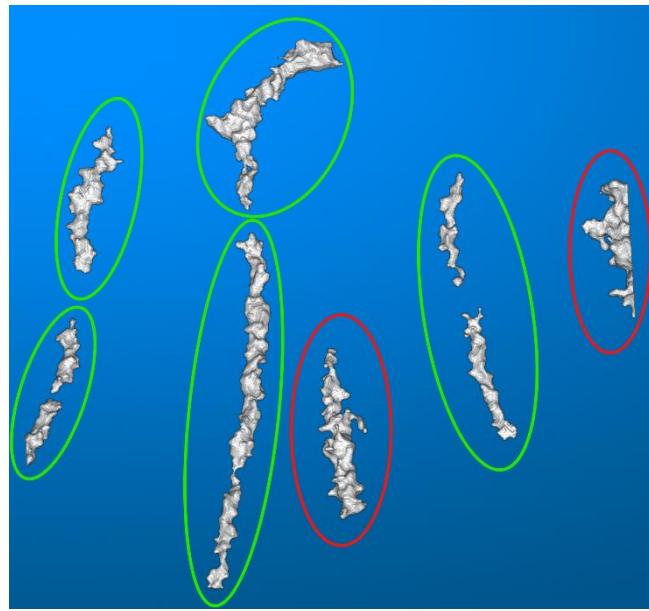


Figura 7.66 - Volume affinato con rumore molto simile a una vena

Dalla fase di **affinaDisturbi** risulta ancora più evidente come il rumore abbia assunto sembianze molto simili a una vena. A questo punto diamo in pasto questo risultato a **eliminaRumore**.

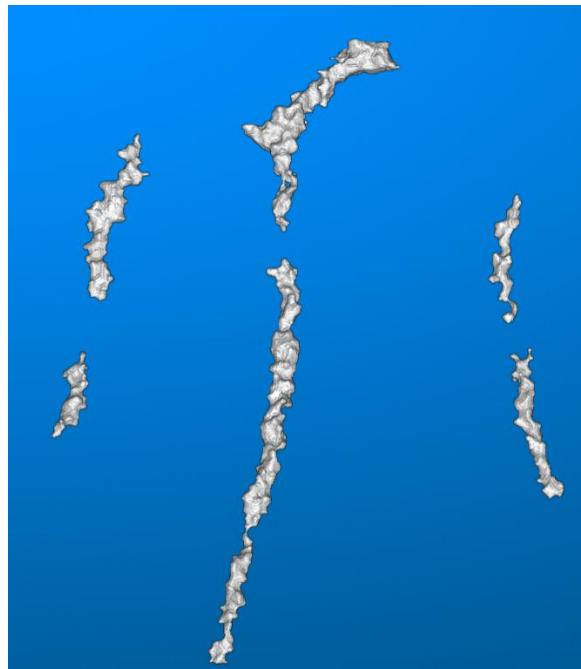


Figura 7.67 - Risultato di **eliminaRumore**

La funzione è riuscita ad individuare gli outlier e a rimuoverli dal volume. Spieghiamo brevemente come. Nella funzione **eliminaRumore** non sono state prese in considerazione features geometriche che avrebbero portato agli stessi problemi discussi in precedenza ma come già detto sono state prese in considerazione features basate sulla distribuzione delle componenti nello spazio. Riportiamo i grafici di decisione e mostriamo perché le componenti sono state eliminate.

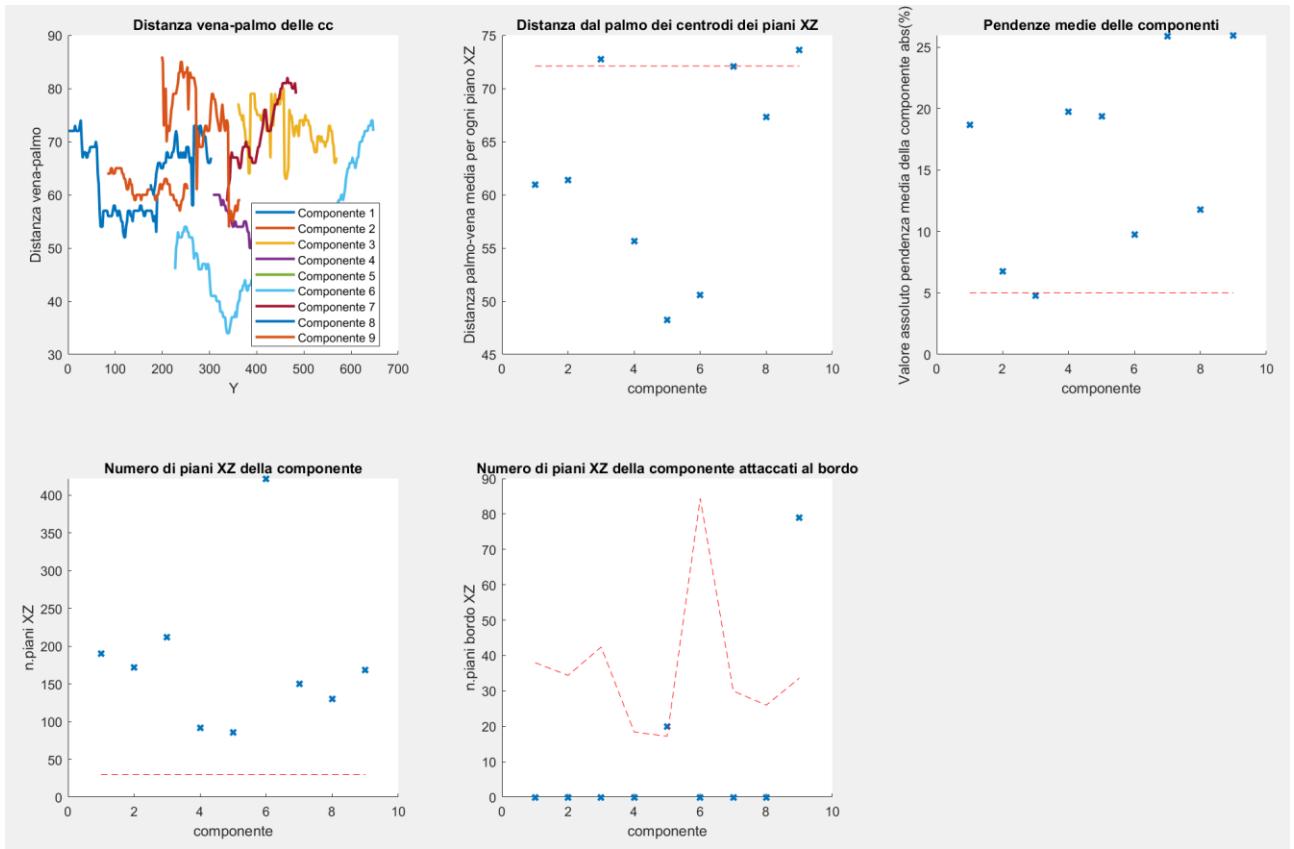


Figura 7.68 - Grafici decisionBoundary eliminaRumore

Il filtraggio statistico è basato su 3 criteri:

1. Distanza centroide-palmo per ogni BSCAN superiore alla soglia e pendenza media assoluta della componente inferiore al 5 %
 - a. Soglia calcolata come la media+deviazione standard delle medie delle distanze per ogni componente
 - b. (grafico in posizione [1,2]) AND (grafico in posizione [1,3])
2. Numero di BSCAN della componente inferiore a 30
 - a. (grafico in posizione [2,1])
3. Numero di BSCAN della componente connessi con il bordo superiore al 20% del numero dei BSCAN da cui è costituita
 - a. (grafico in posizione [2,2])

Dai grafici si vede come la componente 3 supera la soglia calcolata in merito alla distanza media dei centroidi di ogni BSCAN dal palmo (grafico in posizione [1,2]). Condizione che deve essere combinata con una pendenza media assoluta inferiore al 5% (grafico in posizione [1,3]), questa soglia è stata impostata in modo empirico. Dato che il rumore simile alle vene, come visto prima, è costituito dalla superficie del nero sottostante si assume che questo non sia pendente ma pressoché piatto. Queste due condizioni unite da un AND fanno considerare la componente 3 come rumore. Per quanto riguarda la componente 9 si vede dalla Figura 7.66 che essa è un rumore derivante dal bordo del volume, difatti supera la soglia impostata al 20% della lunghezza totale connessa con il bordo (grafico in posizione [2,2]).

Vediamo un esempio del filtraggio statistico nel caso del criterio 2. Nel caso di seguito si nota come la procedura di affinamento abbia prodotto degli artefatti geometricamente simili a delle vene. Ma questi

sono chiaramente delle componenti da eliminare dato il numero irrisorio di piani XZ da cui sono costituiti.

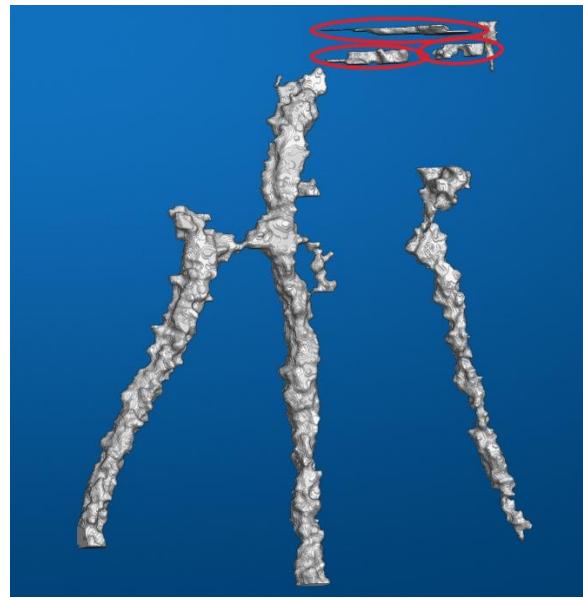


Figura 7.69 - Esempio di artefatti prodotti dalla procedura di affinamento

Questo caso ricade nel criterio 3. Le componenti cerchiate di rosso sono inferiori alla soglia minima fissata a 30.

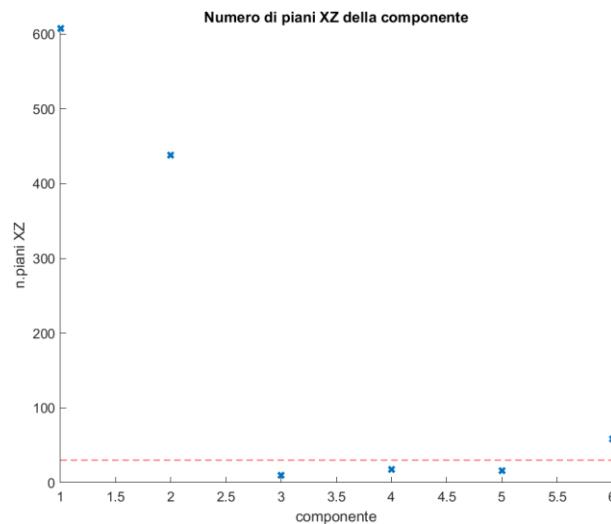


Figura 7.70 - Decision boundary relativi al criterio 2

Difatti notiamo come queste vengano eliminate da **eliminaRumore**



Figura 7.71 - Risultato procedura *eliminaRumore*

7.8.3 Filling vene

Dopo tutti questi procedimenti l'ultima fase del processo di affinamento è il ***filling***. In questa fase andiamo ad effettuare una procedura per migliorare la resa grafica del template e renderlo più omogeneo. In seguito alle operazioni precedenti alcuni segmenti venosi possono essersi assottigliati e quindi per migliorarne l'andamento nello spazio e renderlo più realistico si va a rendere più consistenti le porzioni sottili tramite un'operazione di dilatazione mirata. Il procedimento si costituisce delle seguenti fasi:

1. Si scansionano tutte le componenti rimaste
2. Si applica la procedura di affinamento dei disturbi
3. Si calcola la dimensione del raggio di dilatazione in funzione delle proprietà geometriche della vena
4. Si skeletonizza il volume per ottenere una guida su cui applicare la dilatazione
5. Si calcola la mappa di distanza della componente
6. Si dilatano le porzioni sottili della componente

Vediamo come funziona il procedimento analizzando un'acquisizione di esempio.

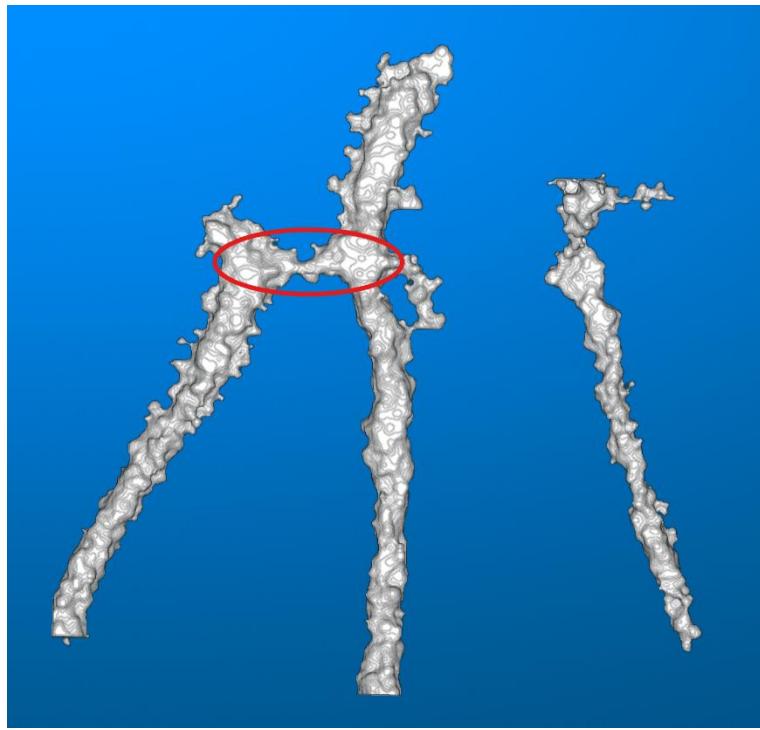


Figura 7.72 - Risultato della fase di affinamento che necessita di un filling

Come si vede nella zona evidenziata in rosso le due vene risultano essere collegate da una porzione molto sottile rispetto al diametro delle due.

Calcoliamo quindi il volume skeletonizzato e la mappa di distanza

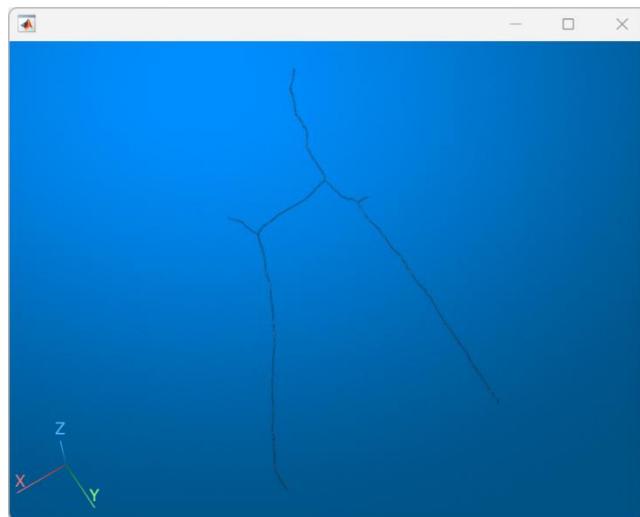


Figura 7.73 - Volume skeletonizzato



Figura 7.74 - Mappa di distanza

A questo possiamo procedere con il filling che quindi verrà effettuato solo nelle zone guidate dal volume skeletonizzato e che presentano un valore della mappa di distanza inferiore alla soglia impostata. Precisiamo che il filling non va ad alterare il numero di componenti connesse, quindi le vene che erano singole rimangono singole le vene che erano biforcute rimangono biforcute. Questo perché vado ad applicare la procedura singolarmente su tutte le componenti connesse andando a dilatare le zone sottili rendendo la vena più uniforme.

Di seguito vediamo il risultato finale della fase di affinamento completo di filling che costituisce il **template 3D finale**.

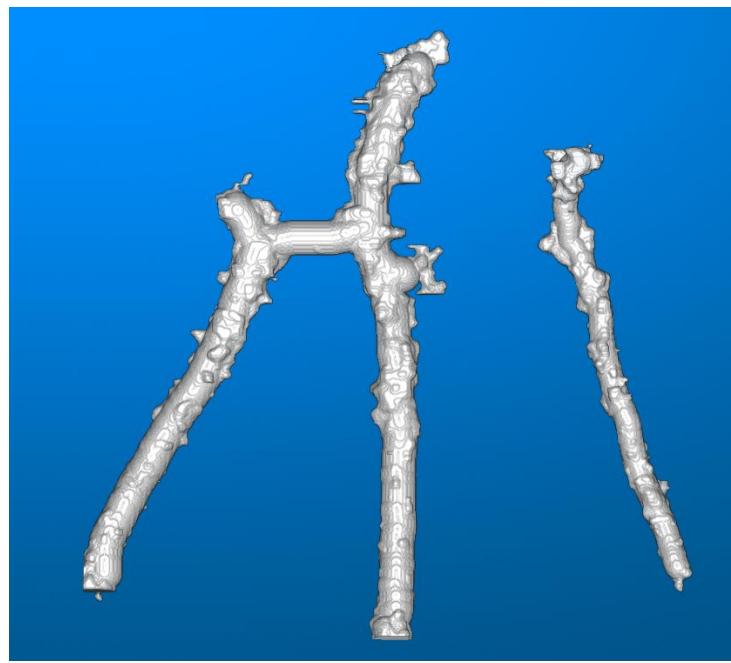


Figura 7.75 - Template finale 3D

7.9 VISUALIZZAZIONE DEL TEMPLATE FINALE

In questo capitolo andiamo a confrontare alcuni template estratti con il volume originale per *dimostrare* la correttezza dei template estratti, nonché i suoi limiti.

Per fare un confronto tra il volume originale e il template è stato creato un volume unito pari alla somma dei due. In questo modo il template estratto si va ad inserire perfettamente all'interno del volume originale così da avere un riscontro obiettivo sulle potenzialità del metodo.

7.9.1 Utente 1

Iniziamo con il primo utente. In questo caso andiamo a riportare anche varie sezioni del volume di partenza.

Sull'immagine di sinistra riportiamo il volume originale, sull'immagine di destra il template estratto

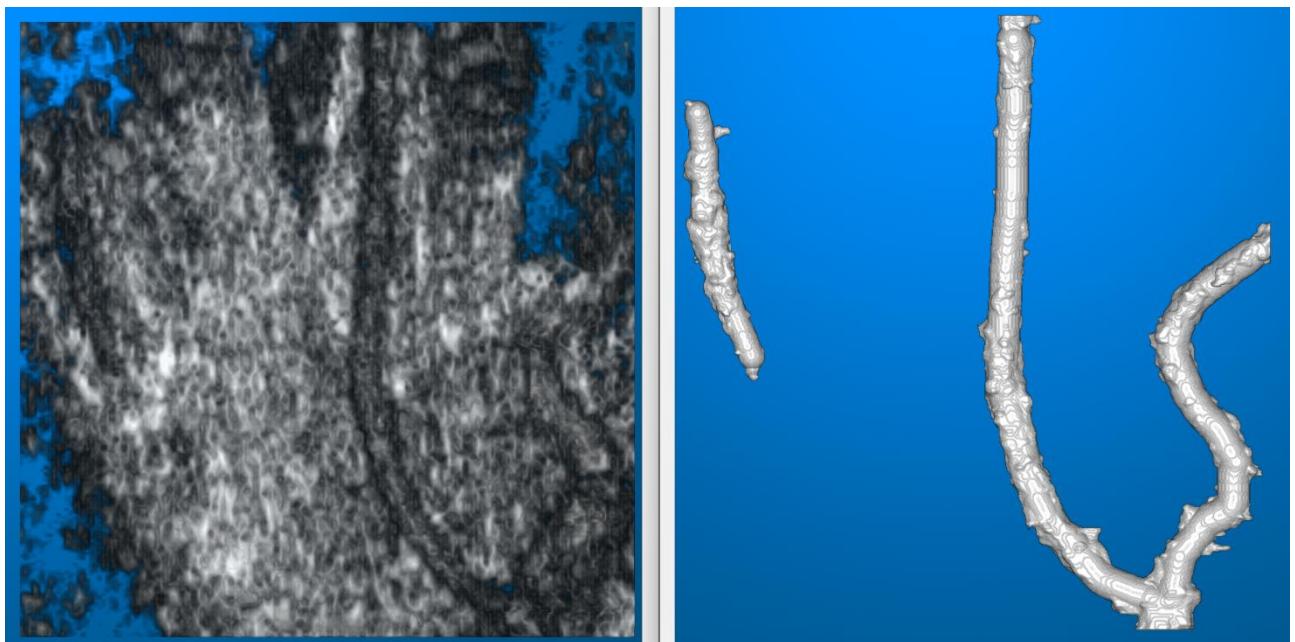


Figura 7.76 - Confronto Z-CSCAN tra originale e template

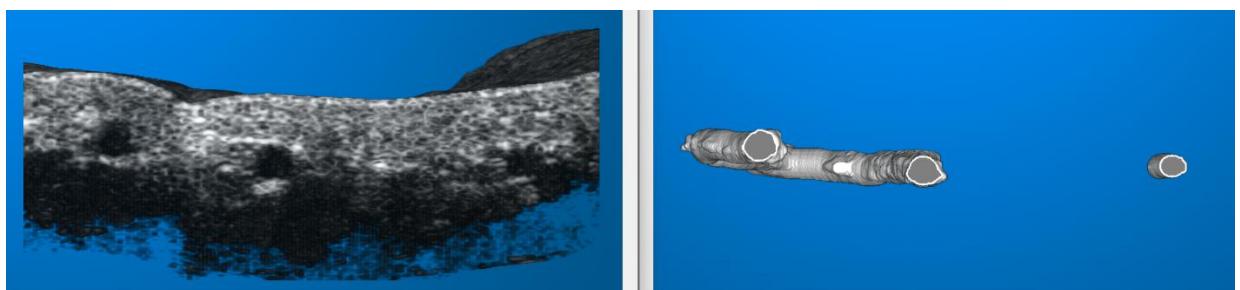


Figura 7.77 - Confronto X-BSCAN tra originale e template

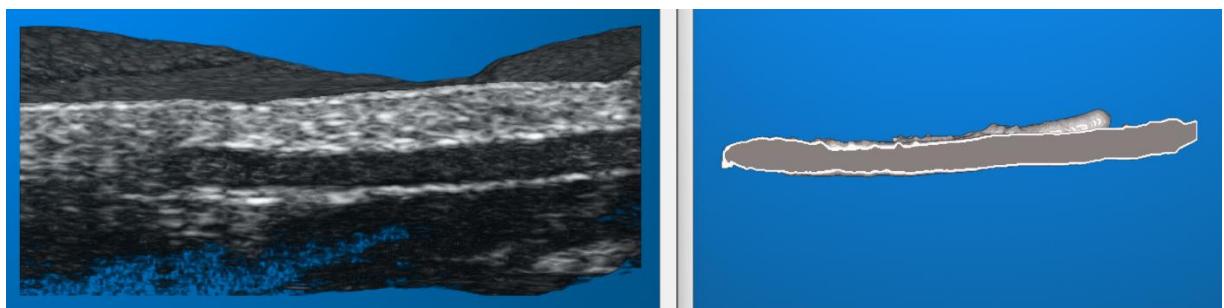
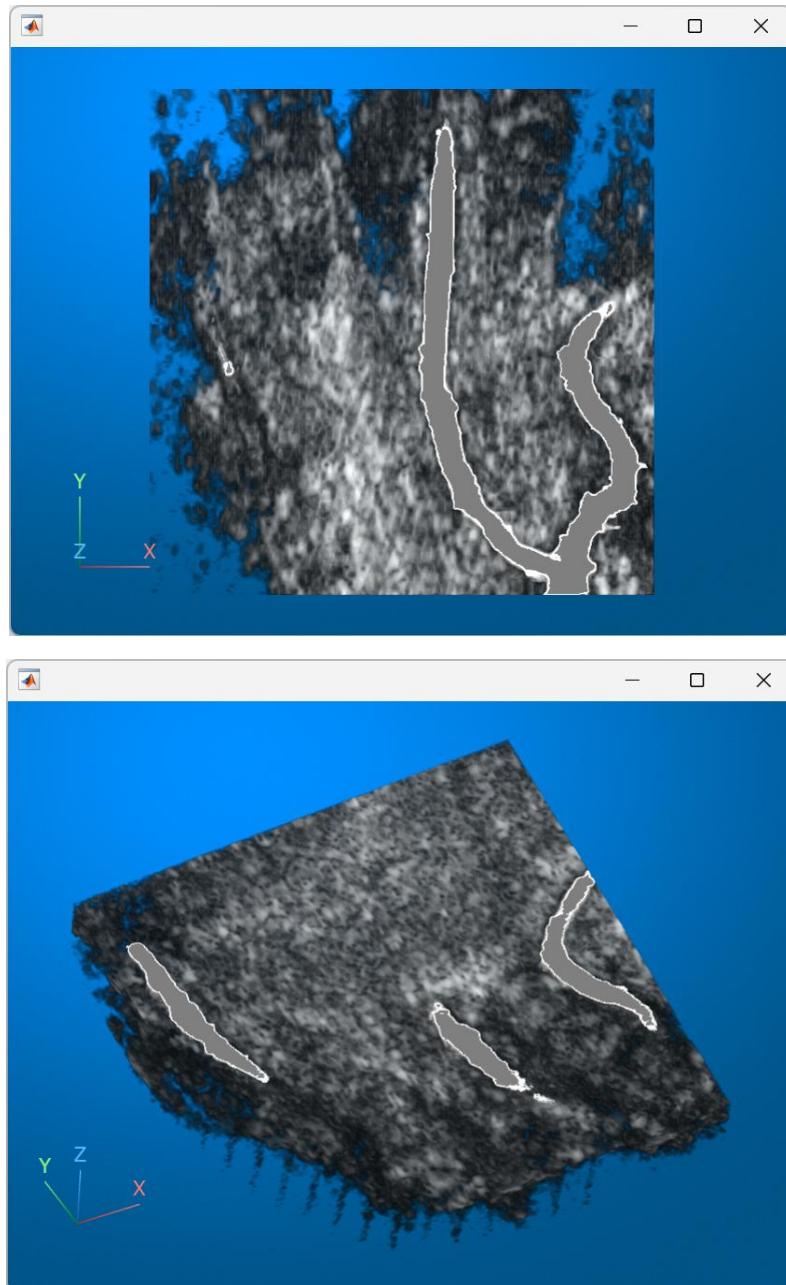
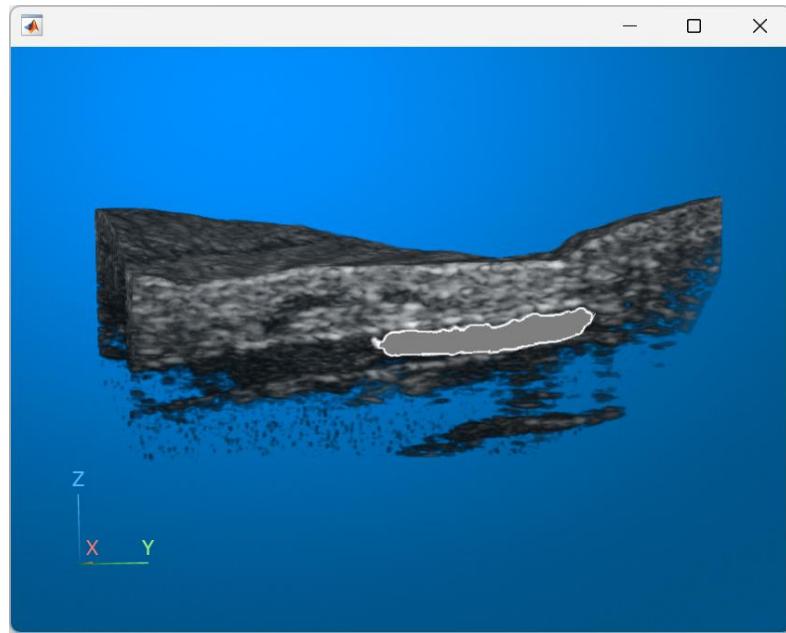


Figura 7.78 - Confronto Y-BSCAN tra originale e template

Una volta appurato che il template rimane spazialmente fedele alla posizione delle vene del volume originale riportiamo il volume unito per avere un confronto più esplicativo. Di seguito riportiamo diversi clipping plane non ortogonali alle 3 dimensioni in modo tale da evidenziare lo sviluppo delle vene nelle 3 direzioni. I voxel di colore grigio chiaro rappresentano il template estratto.



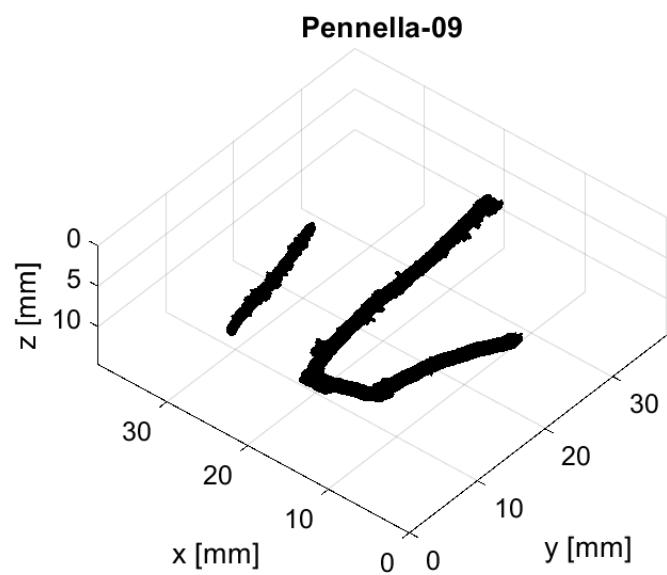
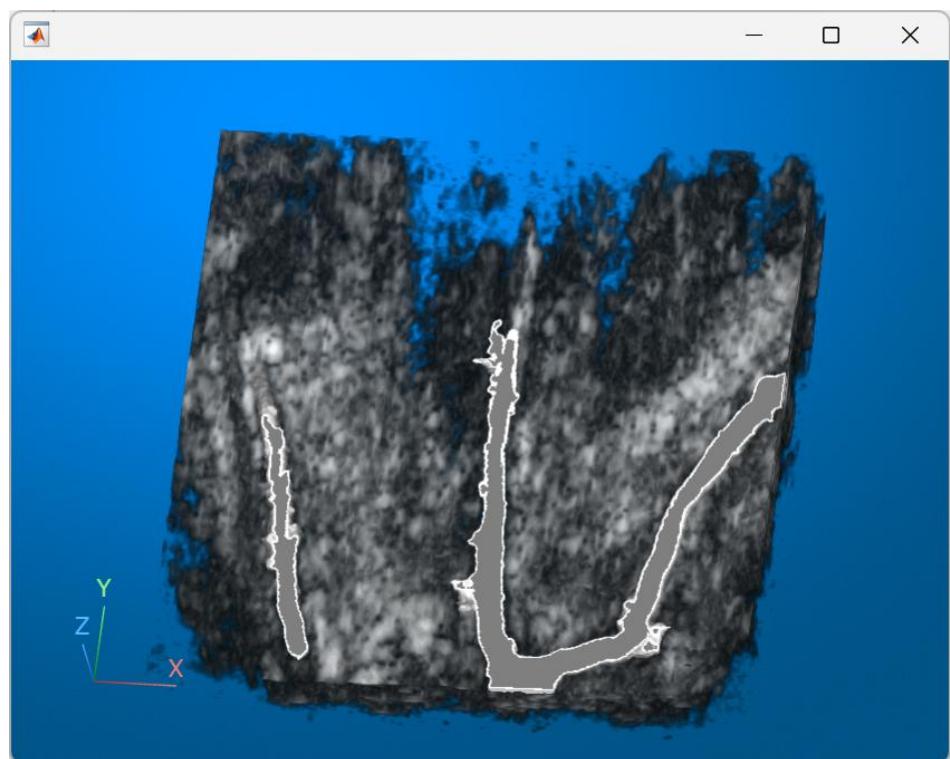


Come si può vedere dalle figure riportate il template segue l'andamento in modo molto preciso della vena originale conservando orientamento, spessore, forma e sviluppo nello spazio.

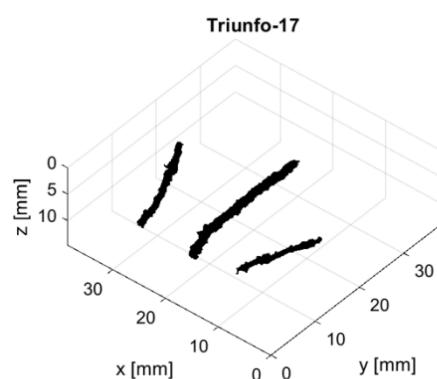
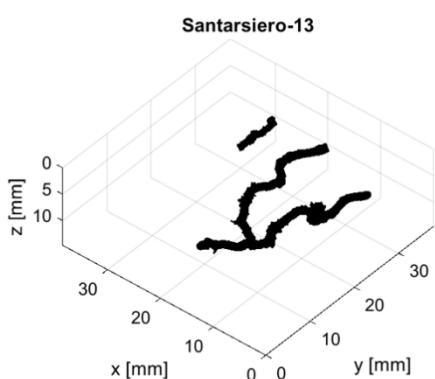
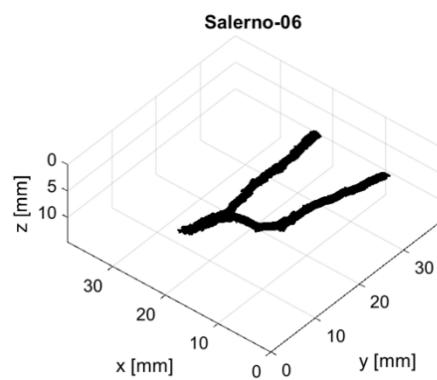
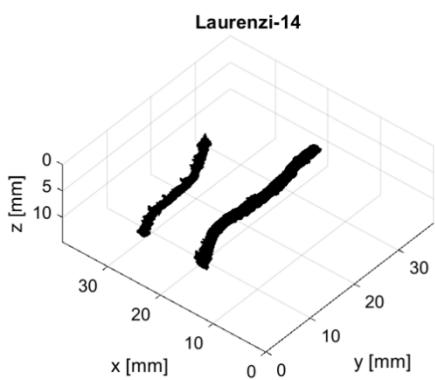
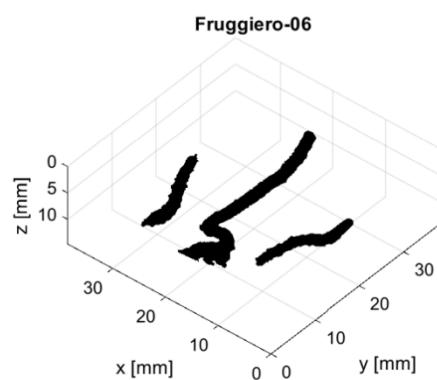
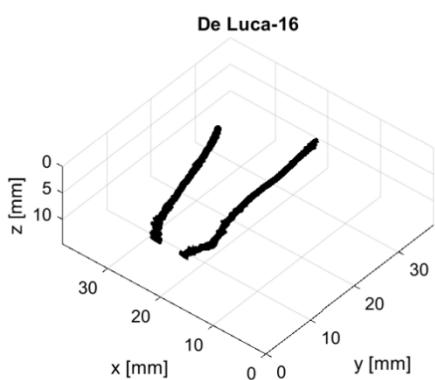
Difatti dimostrano come questo metodo non sia una ricostruzione delle vene bensì un isolamento dei voxel interessati da tutto il resto. Il metodo CCM si pone lo scopo di analizzare i voxel presenti in un volume lasciando soltanto quelli appartenenti a delle vene e nient'altro. Questi sono i vantaggi di un metodo basato sull'analisi dei dati, rispetto a un metodo basato sulla ricostruzione (vedi il metodo Vizzuso).

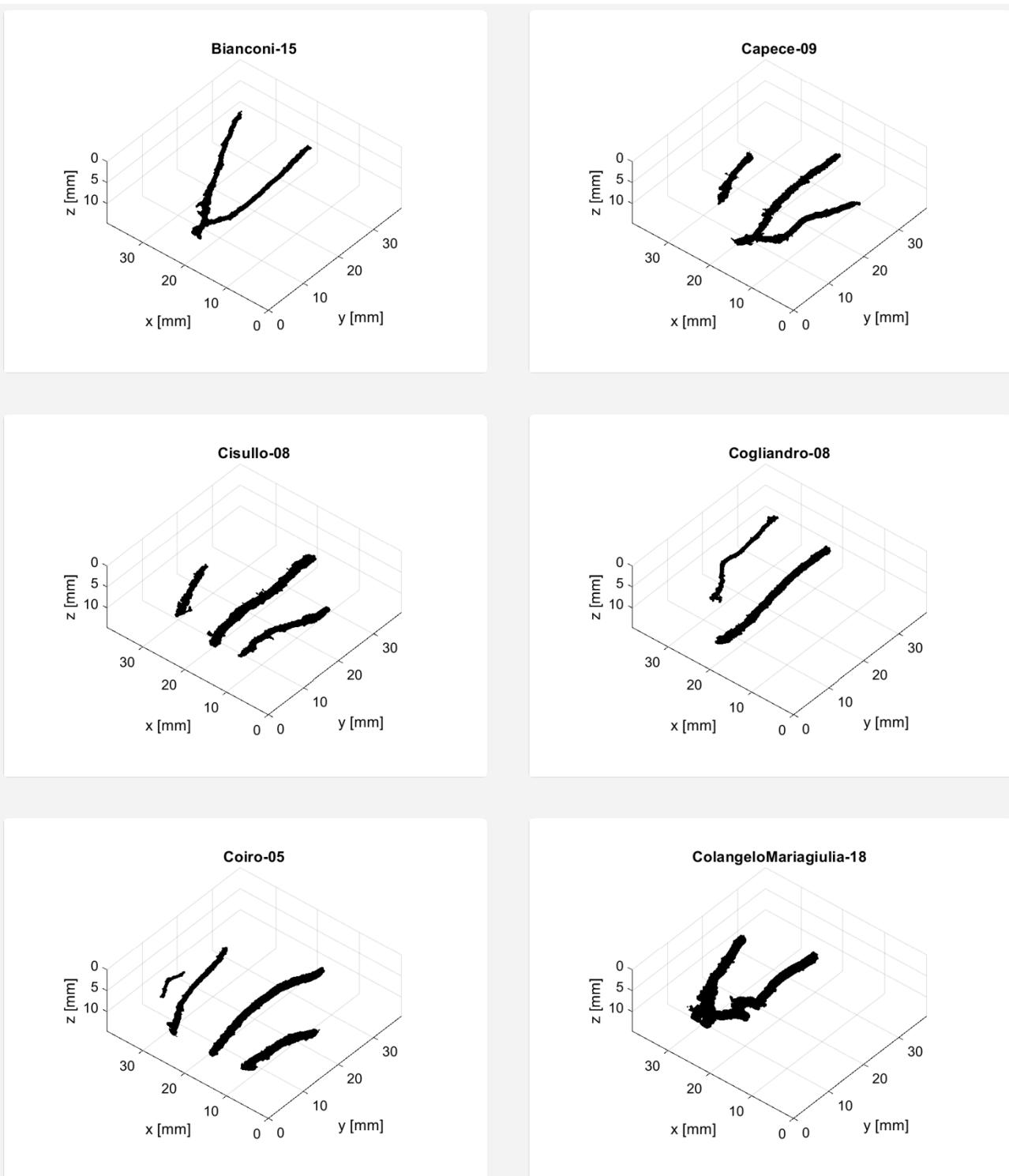
Di seguito riportiamo altri esempi di template graficati con gli assi di riferimento per avere un riscontro dimensionale.

7.9.2 Utente 2



7.9.3 Altri utenti





8 MATCHING 3D

Nel capitolo seguente illustreremo l'ultima fase del processo di riconoscimento del palm-vein. In questa fase si effettua il matching, un processo che misura la similarità tra due template3D permettendo di stabilire se un individuo risulta effettivamente colui che dichiara di essere.

8.1 PROCEDURA

La procedura utilizzata per il matching 3D si basa su un approccio **voxel-to-area** descritto dall'equazione:

$$SCORE(A, B) = \frac{2}{M_A + M_B} \sum_{i=1}^m \sum_{j=1}^n \sum_{z=1}^o [A(i, j, z) \wedge B(i, j, z)]$$

Equazione 8.1 - Matching 3D voxel-to-area

In cui

- $A(i, j, z)$ è il template di riferimento
- $B(i, j, z)$ è il template preso in esame
- M_A è la somma di tutti i voxel diversi da 0 presenti nel template A
- M_B è la somma di tutti i voxel diversi da 0 presenti nel template B

Il confronto avviene tra due volumi sovrapposti, applicando un AND logico, per identificare e conteggiare i voxel corrispondenti posizionati nella medesima area. Il risultato ottenuto è detto **score**, un valore numerico compreso nell'intervallo [0, 1], dove 0 indica una totale discordanza e 1 rappresenta una perfetta corrispondenza.

Il calcolo dello score, da solo, non è sufficiente a garantire un risultato accurato. Errori di traslazione e rotazione, dovuti a un posizionamento impreciso della mano, possono compromettere l'efficacia del matching.

A dimostrazione di ciò, in Figura 8.1, sono riportati i pattern venosi estratti da due acquisizioni differenti, relative ad uno stesso utente. Utilizzando lo script **confrontaTemplate**, è possibile osservare come una collocazione errata del palmo, durante la fase di acquisizione, abbia influenzato la posizione della vena centrale, con conseguente generazione di uno score pari a 0 ad indicare totale discordanza.

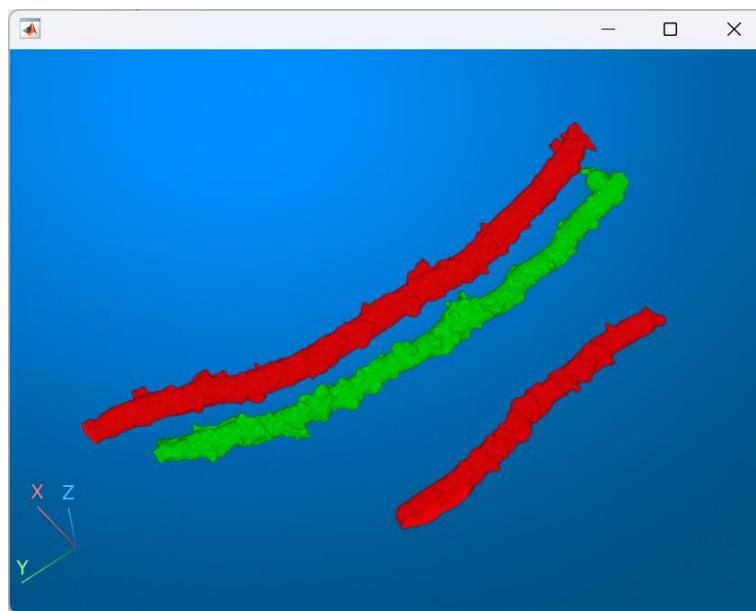


Figura 8.1 - Sovrapposizione e confronto fra template di uno stesso utente. Nella specifico in rosso è raffigurato il primo template 'D'Onofrio_02' e in giallo il secondo 'D'Onofrio_018'

Per ovviare a tale problema la procedura **matching3Dtr**, da noi proposta, tenta di applicare opportune traslazioni e rotazioni ai template B, mantenendo fissa la posizione e l'orientamento del template A, al fine di allineare correttamente i volumi confrontati.

I template utilizzati per il matching, derivati dalla fase di filling, possiedono un proprio spessore e, pertanto, non vengono ulteriormente dilatati. Per garantire che i movimenti e le trasformazioni non generino errori di bordo, si applica un'operazione di padding ai due volumi.

Il metodo **matching3Dtr** utilizza due approcci distinti per l'allineamento e il confronto di volumi 3D, selezionando di volta in volta l'approccio più adatto in base allo score ottenuto dalla prima trasformazione rigida. Questa scelta strategica nasce dall'esigenza di ridurre al minimo i tempi di calcolo, come già evidenziato in precedenza. Di seguito riportiamo una descrizione più dettagliata dei metodi utilizzati.

8.1.1.1 Allineamento calcolato

Il primo metodo consiste nell'allineare i due volumi calcolando la discordanza tra posizione e orientamento e in seguito applicare una determinata trasformazione rigida per centrarli.

Viene eseguita una trasformazione rigida composta da:

1. Rotazione sui 3 assi principali di un angolo calcolato come la differenza degli angoli di Eulero delle rispettive componenti connesse. Step che permette di allineare l'orientamento dei due volumi.
2. Traslazione sui 3 assi principali di una dimensione pari alla differenza tra le coordinate dei centroidi delle componenti principali. Step che permetti di allineare la posizione dei due volumi.

Questo approccio viene adottato perché i volumi in esame sono di dimensioni notevoli (650x650x320), tentare tutte le possibili combinazioni di rotazioni e traslazioni in modo iterativo risulterebbe computazionalmente molto oneroso. Grazie a questa procedura, invece, è possibile allineare i volumi utilizzando soltanto 4 trasformazioni: **3 rotazioni e 1 traslazione**.

Dopo l'applicazione della trasformazione rigida al template B, viene calcolato lo score. Se questo valore è maggiore di 0.4, il processo si interrompe, considerando il confronto tra i due template sufficientemente accettabile. Questa metodologia, quindi, utilizza un meccanismo a “**corto circuito**”, interrompe il procedimento non appena si raggiunge uno score soddisfacente superiore alla soglia scelta, ottimizzando così ulteriormente il carico computazionale.

La soglia per lo score pari a 0.4 è un valore non limitante è stato scelto in quanto, nel nostro caso, ha dimostrato essere un valido compromesso per ridurre efficacemente sia i falsi negativi che i falsi positivi.

Per evitare allineamenti errati, abbiamo definito, un limite massimo entro il quale i due volumi possono essere considerati compatibili. Questo impedisce, ad esempio, che componenti simili ma distanti (template appartenenti ad utenti diversi) possano essere erroneamente allineate, ottenendo uno score alto ma non rappresentativo generando falsi positivi.

In Figura 8.2 è riportato un esempio di allineamento calcolato per template appartenenti ad uno stesso utente. Si può notare come, con l'applicazione di sole 4 trasformazioni, le vene risultino perfettamente sovrapposte con score pari a 0.51.

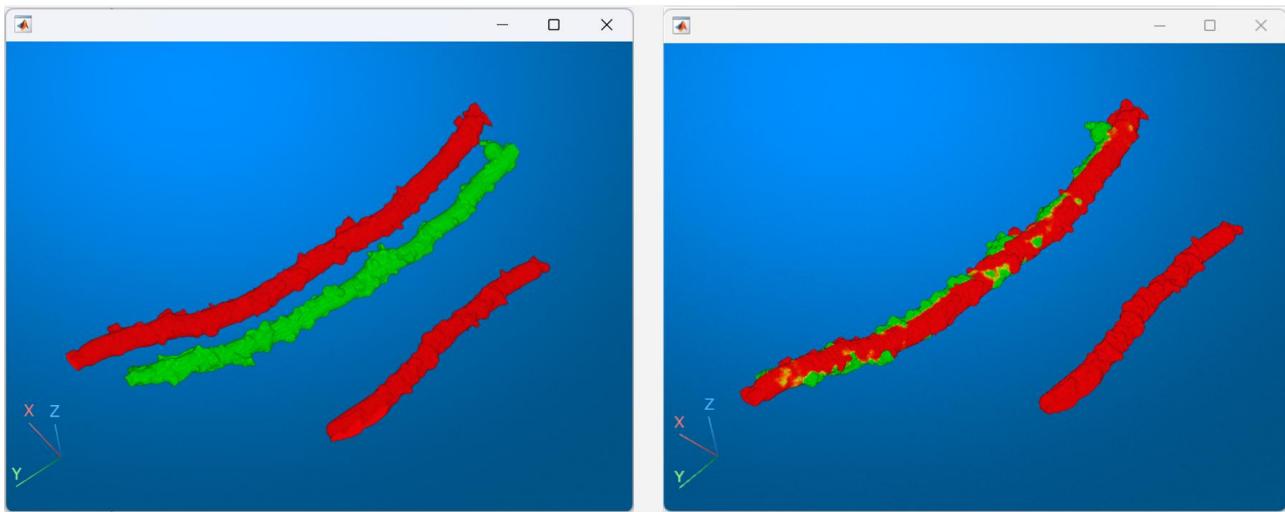


Figura 8.2 - Sovrapposizione e confronto fra template di uno stesso utente. A sinistra vi è il confronto PRE matching a destra vi è il confronto POST allineamento calcolato

8.1.1.2 Allineamento iterativo

Se lo score ottenuto dalla trasformazione rigida è inferiore a 0.4, si passa a una procedura iterativa che viene applicata sempre ai due volumi nelle loro posizioni iniziali. La procedura consiste di due fasi:

1. Traslazione iterativa sui 3 assi principali.

- Vengono inizializzate in una matrice tutte le combinazioni possibili di traslazione sui 3 assi di una quantità che si adatta al diametro delle vene e che varia con uno step pari al diametro/3.
- Calcolo lo score per ogni combinazione.
- Ricavo la combinazione che mi ha fatto ottenere lo score maggiore
- Traslo il volume originale con la combinazione trovata

2. Rotazione iterativa sui 3 assi principali

- Calcolo l'angolo in gradi con cui voglio ruotare il volume, pari al diametro/10.
- Effettuo delle rotazioni fisse sui 3 assi in senso orario e antiorario
- Calcolo lo score per ogni combinazione
- Restituisco lo score massimo

Notiamo che con questa procedura eseguiamo:

- $3^7 = 343$ traslazioni
 - Tutte le disposizioni con ripetizione del vettore composto da 7 elementi pari a **b. [-diametro : diametro/3 : diametro]**
- 7 rotazioni sui 3 assi descritte dalle combinazioni:
 - a. R = [0,0,0; 1,0,0; 0,1,0; 0,0,1; -1,0,0; 0,-1,0; 0,0,-1]**

Per un totale di **350 trasformazioni** eseguite sulla GPU. Tutte queste operazioni per l'allineamento iterativo sono necessarie perché piccole differenze strutturali nei template causano importanti disallineamenti negli angoli di Eulero. Quindi in caso di template leggermente diversi l'allineamento calcolato produrrà uno score molto basso, di conseguenza non ci resta che procedere per iterazioni successive.

Per giustificare il numero elevato di traslazioni, è importante sottolineare che ogni asse viene esplorato separatamente in 7 posizioni, un numero del tutto ragionevole. Sebbene il numero di traslazioni per ciascun asse sia contenuto, la combinazione di tutte le possibili traslazioni sui tre assi porta a un numero complessivo di operazioni significativo. Questo accade perché il nostro metodo di matching

lavora su tutto il volume tridimensionale, senza fare riferimento a una proiezione bidimensionale. Dai vari test condotti ne è emerso che, ridurre il numero di posizioni per ogni asse non comporta miglioramenti sostanziali rispetto all'utilizzo esclusivo della procedura di allineamento calcolato.

Nelle figure sottostanti viene riportato un esempio di allineamento iterativo, utilizzando due template appartenenti allo stesso utente. Nella Figura 8.3 si osserva che, applicando il matching tramite la procedura di allineamento calcolato su template già originariamente allineati, comporta un disallineamento, con uno score di 0.16. In questo caso, infatti, le sole 4 trasformazioni non sono sufficienti a garantire una sovrapposizione adeguata. Tuttavia, questo problema viene risolto tramite la procedura iterativa. Nella Figura 8.4 è mostrato il confronto tra i due template dopo aver applicato l'approccio iterativo. Si può notare che, grazie ad un elevato numero di trasformazioni, i template si sovrappongono perfettamente.

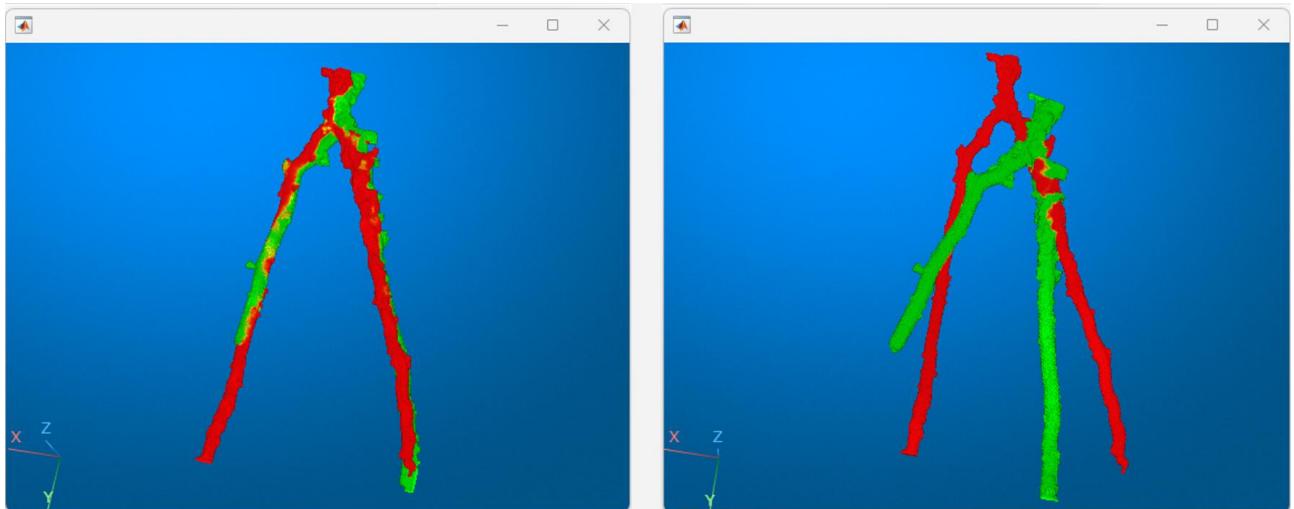


Figura 8.3 - Sovrapposizione e confronto fra template di uno stesso utente. A sinistra vi è il confronto PRE matching a destra vi è il confronto POST allineamento calcolato

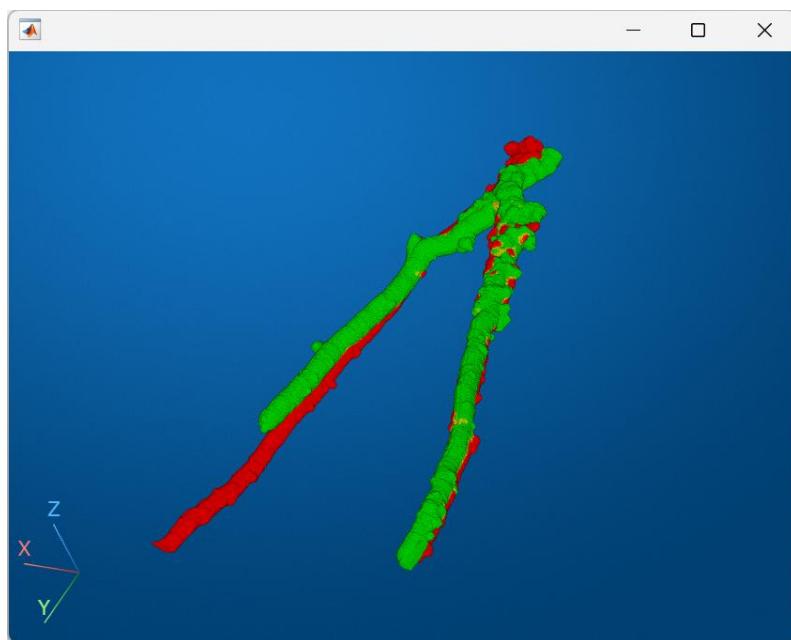


Figura 8.4 - Sovrapposizione e confronto fra template di uno stesso utente POST allineamento iterativo.

Nonostante l'impiego di due approcci distinti, il confronto di set di dati di dimensioni considerevoli fa sì che i tempi di calcolo risultino comunque elevati. Per ottimizzare al massimo la velocità di

esecuzione, abbiamo sfruttato la potenza di calcolo della scheda grafica. Tuttavia, questa soluzione presenta una limitazione importante: non supporta l'utilizzo di cicli paralleli.

8.2 RISULTATI

Un primo matching 3D di prova è stato realizzato su un sotto database di quello di partenza, in modo da poter valutare le prestazioni e i tempi di calcolo del metodo di matching sopra descritto utilizzando il seguente hardware:

- Processore: Intel® i5 13600KF @5,1 GHz 14 cores e 20 threads.
- Memoria RAM: 32 GB DDR5-6000Mhz.
- Scheda Video: Nvidia RTX 3070 con 8GB di memoria video.

Precisamente il primo matching 3D è stato effettuato su un campione di 26 utenti con circa 3 acquisizioni per utente. La lista delle acquisizioni scelta per questo primo matching di prova si trova nella cartella **matching/listaTemplate** e prende il nome di **listaTemplateProva.xlsx**

Utilizzando la formula combinatoria

$$\frac{n(n - 1)}{2}$$

Equazione 8.2 - formula per calcolare il numero di confronti da effettuare

Possiamo calcolare il numero di confronti che sono stati effettuati in cui n rappresenta il totale di acquisizioni scelte con cui effettuare il matching. Dato $n = 76$ il numero totale di confronti risulta essere 2850, il tempo necessario per completare il matching su questo set di prova è risultato essere 49487 secondi. Con questi dati possiamo stimare il tempo medio di calcolo per confronto pari a **17,36** secondi per confronto.

I risultati ottenuti performando il matching sul database di prova sono i seguenti

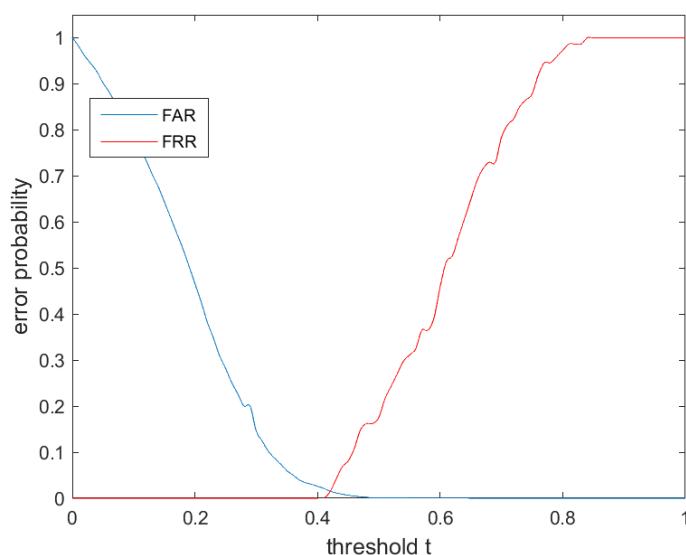


Figura 8.5 – Curva FAR/FRR database di prova

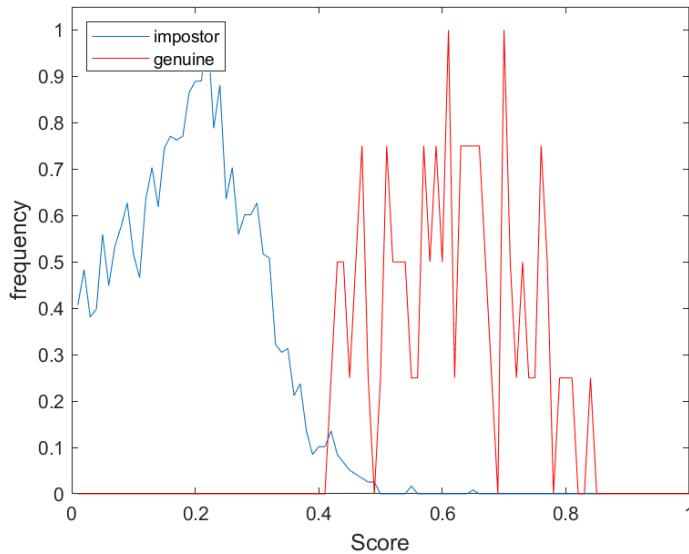


Figura 8.6 - Distribuzione impostori - genuini database di prova

Notiamo dalle figure di cui sopra come i risultati siano abbastanza buoni, con un **EER** contenuto pari a solo 1,5%. Comunque è presente una certa sovrapposizione tra impostori e genuini con score massimi impostori ben allineati con gli score medi genuini.

Lista acquisizioni utilizzata	EER(%)
listaTemplateProva.xlsx	1,5

8.2.1 Considerazioni sul tempo di esecuzione

Come accennato per velocizzare le operazioni di trasformazione dei volumi 3D abbiamo optato per computare le rotazioni e le traslazioni utilizzando la scheda video. Questo è possibile convertendo i volumi 3D in **gpuArray**, chiamando le funzioni su essi e recuperando il risultato con il metodo **gather**. Questo approccio costringe l'esecuzione sequenziale dei confronti, in poche parole non è possibile sfruttare i cicli for paralleli. Per questo l'esecuzione del matching sul pc del laboratorio non porta a grandi vantaggi rispetto al pc di riferimento per i benchmark riportati. In particolare il PC del laboratorio ha seguenti specifiche:

- Processore: Intel® Xeon® Gold 6238R CPU @2.20Ghz con 56 cores e 112 threads
- Memoria RAM: 512 GB DDR4-3000Mhz.
- Scheda Video: Nvidia RTX A5000 con 24GB di memoria dedicata

La differenza non è tantissima perché non usando il calcolo parallelo non andiamo a sfruttare la potenza dei numerosi core logici di cui dispone il pc del laboratorio. Difatti in un ciclo for sequenziale è più rilevante la frequenza del processore e vediamo come in single core l'I5 abbia un vantaggio prestazionale. Per questo nei continui trasferimenti di memoria da CPU a GPU e viceversa è la frequenza che conta e questo spiega la poca differenza tra una workstation server e un computer fisso con hardware moderno. Per quanto riguarda la scheda video la RTXA5000 e la RTX3070 hanno prestazioni tutto sommato paragonabili.

Stimiamo adesso il tempo necessario per effettuare il matching sull'intero database. Nel nostro caso specifico, il database è composto da 73 utenti con 20 acquisizioni ciascuno, per un totale di 1460 template da confrontare. Il numero totale di confronti è pari a:

$$\frac{1460(1460 - 1)}{2} = 1065070 \text{ confronti}$$

Per cui il tempo necessario stimato per completare il database è circa:

$$1065070 \times 17,36 = 18489615,2 \text{ secondi}$$

Equivalenti a 308160 minuti o 5136 ore o 214 giorni. È chiaro che con questo metodo il database completo non può essere utilizzato per effettuare il matching.

Sulla base di queste considerazioni abbiamo scelto di calcolare il matching su un database ridotto di 43 utenti per un totale di 159 template da confrontare. Il numero totale di confronti è pari a 12561. Considerando sempre un tempo medio di 17,36 secondi, per effettuare il matching su questo database sono necessarie circa 60 ore che è un tempo molto più ragionevole seppur lunghissimo.

8.2.1.1 Considerazioni sul tempo di esecuzione per l'estrazione dei template

Per quanto riguarda l'estrazione dei template è stato rilevato un tempo medio necessario di 303 secondi. Il tempo può variare leggermente a seconda delle varie acquisizioni. In questo caso viene fatto uso di cicli for paralleli per ridurre i tempi di calcolo. Quindi eseguendo la procedura di estrazione su hardware con molti core e thread si risparmia parecchio tempo, ad esempio usando il PC del laboratorio.

8.2.2 Scelta delle acquisizioni

Abbiamo scelto di selezionare queste 159 acquisizioni basandoci sulla presenza di due caratteristiche:

1. Assenza di rumore

- a. Alcuni template estratti contengono elementi di rumore che, a causa della loro somiglianza strutturale con le vene, non siamo riusciti a eliminare completamente. L'utilizzo di questi template risulta rischioso, in quanto può generare falsi negativi o falsi positivi, influendo negativamente l'efficienza del sistema portando a un valore di EER (Equal Error Rate) elevato.

2. Presenza di più vene

- a. Il metodo da noi elaborato è un multi-vein il che significa che il nostro obiettivo era riuscire a estrarre il maggior numero possibile di vene visibili per ciascun utente. Per questo motivo, abbiamo deciso di considerare i template che riportano il maggior numero di vene estratte per singolo utente.

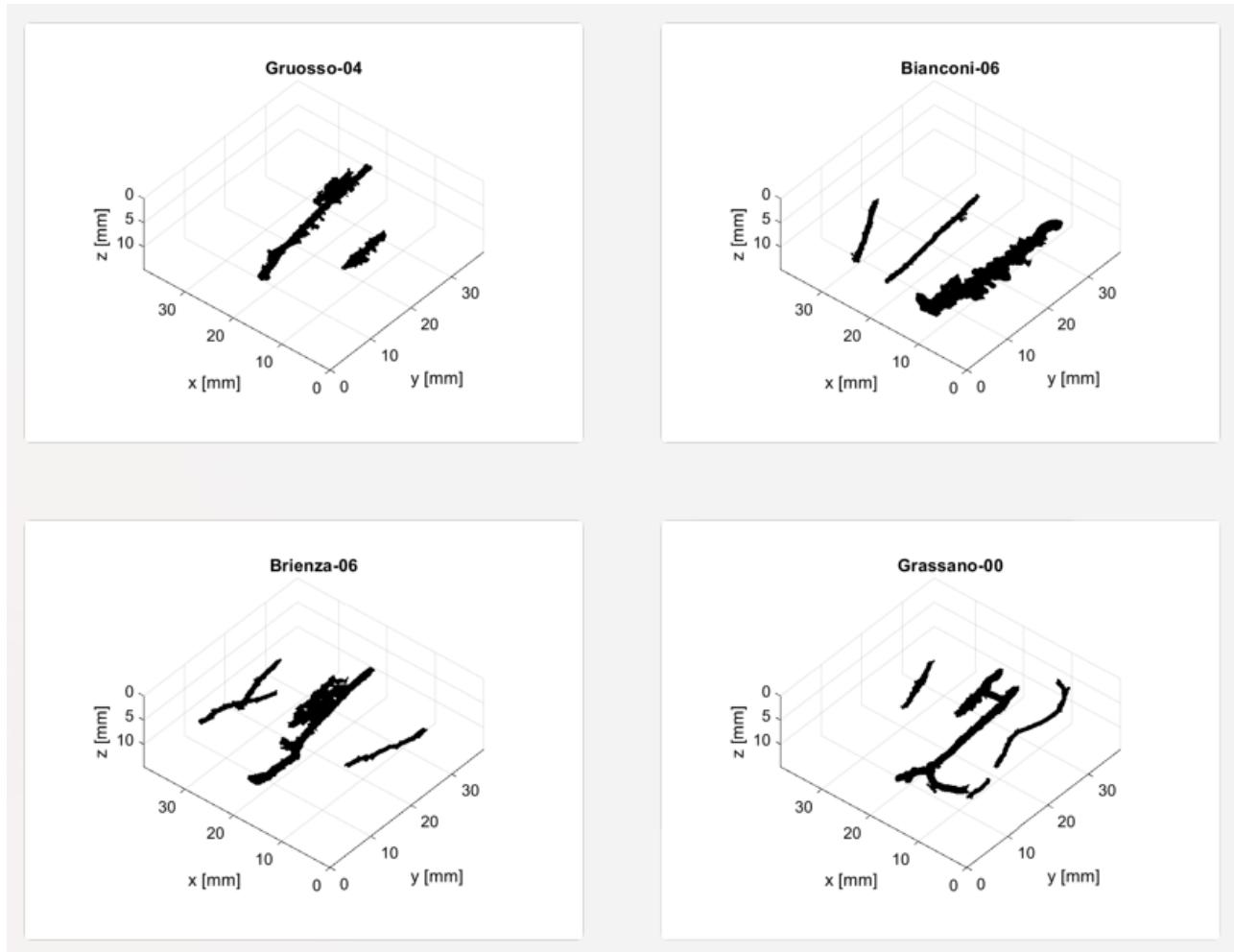
In questo modo abbiamo garantito che i template di un medesimo utente fossero più simili possibile tra di loro per il matching, così da far risultare sufficiente il solo allineamento calcolato per ottenere uno score genuino ed un tempo di elaborazione ottimizzati. Inoltre utilizzando dei template contenenti il maggior numero di vene si ottiene un campione che risulta essere più univoco possibile diminuendo lo score in caso di utenti con vena principale analoga.

Infine ci siamo limitati a considerare al massimo 6 acquisizioni per utente per i motivi relativi ai tempi di esecuzione, così da evitare un numero troppo elevato di confronti. La tabella contenente le acquisizioni scelte si trova nella cartella **matching/listaTemplate** e prende il nome di **listaTemplateFinale.xlsx**

Tali scelte portano ad avere una distribuzione sbilanciata di impostori e genuini, difatti è preferibile avere tante acquisizioni per un utente. Dovendoci comunque limitare nei confronti abbiamo optato per conservare le peculiarità di tutti gli individui.

8.2.3 Esempi

Di seguito sono riportati alcuni esempi di template esclusi prima della fase di matching, a causa della presenza di rumore, sia isolato che adiacente alla struttura venosa.



8.2.4 Valutazione dei risultati

In questo capitolo vengono presentati i risultati ottenuti che ci hanno permesso di valutare l'efficacia del sistema.

In generale, per valutare le prestazioni di un sistema biometrico, si fa riferimento ai seguenti parametri:

- Distribuzione *Impostori/Genuini*, permette di stabilire lo score minimo nei confronti tra genuini e la distribuzione degli scores impostori.
 - *Genuino*: rappresenta un risultato positivo di un confronto tra due template di un medesimo individuo;
 - *Impostore*: indica il risultato di un confronto tra le caratteristiche biometriche di due individui diversi.
- **FAR (“False Acceptance Rate”)** è la frequenza con cui il sistema biometrico accetta erroneamente utenti non legittimi.
- **FRR (“False Rejection Rate”)** è la frequenza con cui il sistema respinge erroneamente utenti legittimi.

- **EER (“Equal Error Rate”)** rappresenta l’errore intrinseco del sistema biometrico, ed è l’ordinata del punto di intersezione tra la curva FAR e la curva FRR. L’obiettivo è quello di minimizzare il più possibile tale errore: più il valore è piccolo, più elevata sarà l’efficienza del sistema di riconoscimento.

Per generare le statistiche si esegue lo script **statics3D.m** che richiede in input la tabella con il punteggio di matching.

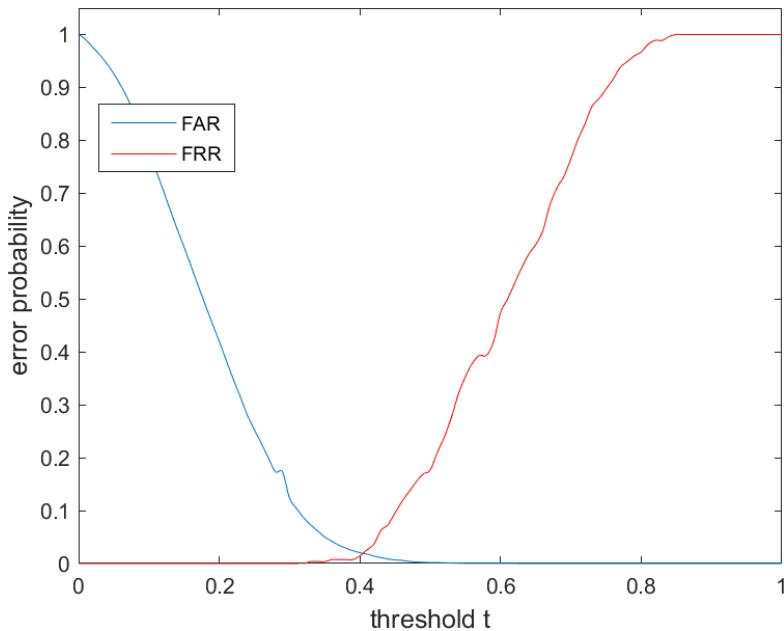


Figura 8.7 - Curva FAR/FRR database finale

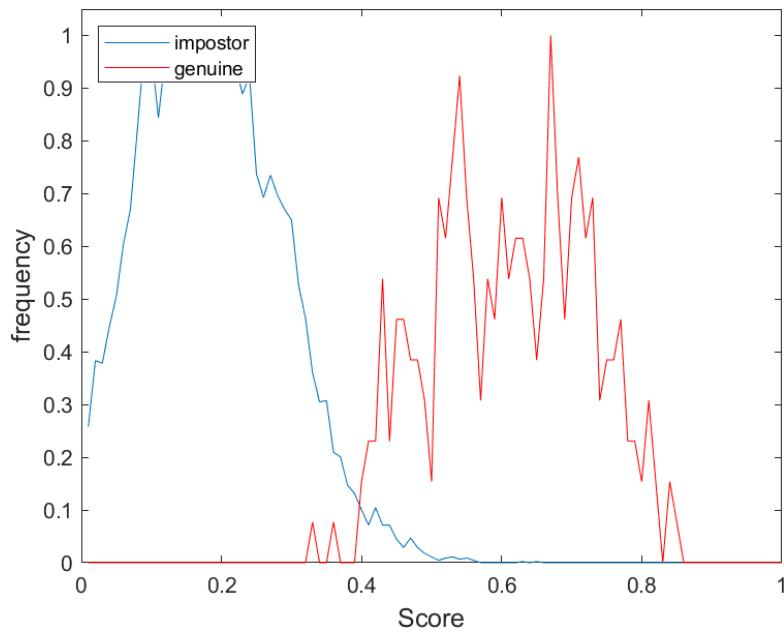


Figura 8.8 - Distribuzione impostori - genuini database finale

Lista acquisizioni utilizzata	EER(%)
listaTemplateFinale.xlsx	1,9

Riportiamo l'**EER** pari a 0,019 o 1,9%.

Dall'analisi dei grafici emergono i seguenti limiti:

1. Score impostori elevati:

Questo accade perché, con l'approccio voxel-to-area, un minimo allineamento di alcuni voxel è sufficiente per ottenere uno score superiore a una determinata soglia. Di conseguenza, nel caso di utenti diversi ma con vene simili in posizioni vicine, lo score può facilmente raggiungere valori elevati. In Figura 8.9, è riportata una parte della tabella contenente gli score ottenuti dal matching tra template impostori.

	1 Utente1	2 Utente2	3 Score
1	'Gruosso_010'	'Martiello_019'	65
2	'Gruosso_011'	'Martiello_017'	55
3	'Larotonda_007'	'Laurenzi_010'	55
4	'Cogliandro_006'	'Martiello_014'	49
5	'Larotonda_008'	'Macellaro_014'	49
6	'Larotonda_013'	'Laurenzi_010'	49
7	'Cogliandro_006'	'Martiello_017'	48
8	'D'Onofrio_001'	'Larotonda_007'	48
9	'Gruosso_010'	'Martiello_017'	48
10	'Cisullo_017'	'Laurenzi_010'	47
11	'Grassano_015'	'Larotonda_007'	47
12	'Grassano_017'	'Laurenzi_007'	47
13	'Larotonda_013'	'Macellaro_014'	47
14	'Cogliandro_005'	'LuongoAntonio_013'	46
15	'Cogliandro_005'	'LuongoAntonio_014'	46
16	'Cogliandro_005'	'Martiello_017'	46
17	'Larotonda_013'	'Laurenzi_008'	46
18	'Laurenzi_010'	'Macellaro_014'	46
19	'Cisullo_009'	'Molinaro_004'	45
20	'Cogliandro_005'	'Martiello_014'	45
21	'Giordano_000'	'Larotonda_008'	45
22	'Giordano_001'	'Laurenzi_010'	45
23	'Giordano_001'	'Macellaro_014'	45
24	'Grassano_017'	'Laurenzi_008'	45
25	'Celi_002'	'Grassano_007'	44
26	'Cisullo_008'	'Molinaro_007'	44
27	'Cogliandro_005'	'Martiello_019'	44
28	'Cogliandro_008'	'Martiello_014'	44
29	'Cogliandro_008'	'Martiello_017'	44

Figura 8.9 – Sezione della tabella impostori ordinata per score decrescente

Nello specifico prendiamo in esame i template ‘Grassano_010’ e ‘Martiello_019’ e focalizziamoci sulla loro sovrapposizione prima e dopo il processo di allineamento.

La Figura 8.10 illustra la sovrapposizione dei due template prima del processo di matching, si osserva come essi presentino pattern venosi simili e orientati nello stesso modo. Successivamente, in Figura 8.11, che rappresenta il risultato post-matching, si nota una sovrapposizione completa dei due template, causando uno score di 0.65.

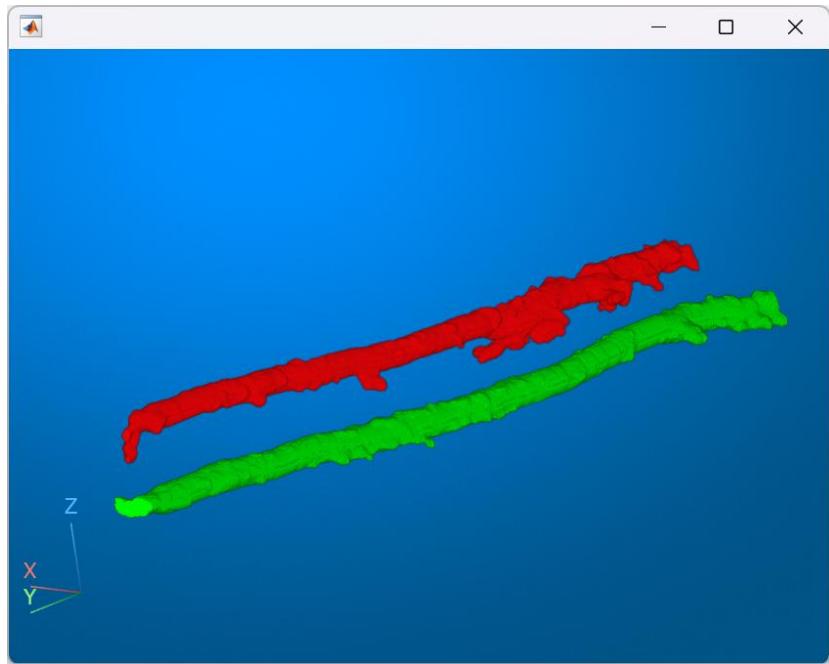


Figura 8.10 – Confronto PRE-matching fra i template ‘Gruosso_010’ in rosso e ‘Martielo_019’ in verde.

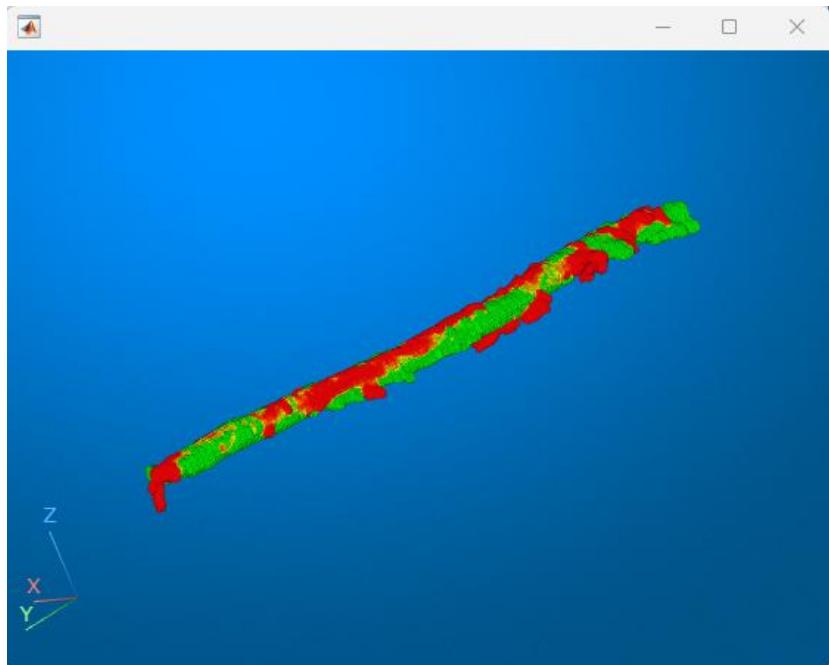


Figura 8.11 – Confronto POST-matching fra i template ‘Gruosso_010’ in rosso e ‘Martielo_019’ in verde.

2. Difficoltà nel raggiungere score genuini elevati:

È complesso ottenere score genuini superiori a 0.8, in quanto risulta difficile allineare perfettamente i volumi in tutte le direzioni, data la loro complessità e variabilità. Questa limitazione influenza sulla possibilità di raggiungere score genuini elevati. In Figura 8.12, è riportata parte della tabella contenente gli score ottenuti dal matching tra template genuini.

	1 Utente1	2 Utente2	3 Score
1	'D"Onofrio_001'	'D"Onofrio_019'	33
2	'Fruggiero_007'	'Fruggiero_019'	36
3	'De Luca_016'	'De Luca_017'	40
4	'Triunfo_006'	'Triunfo_012'	40
5	'D"Onofrio_001'	'D"Onofrio_008'	41
6	'De Luca_008'	'De Luca_016'	41
7	'Salerno_009'	'Salerno_014'	41
8	'Fruggiero_012'	'Fruggiero_019'	42
9	'Macellaro_014'	'Macellaro_019'	42
10	'Sprovera_015'	'Sprovera_018'	42
11	'Cogliandro_001'	'Cogliandro_005'	43
12	'ColangeloGabriele_010'	'ColangeloGabriele_013'	43
13	'D"Onofrio_001'	'D"Onofrio_003'	43
14	'Di Giacomo_012'	'Di Giacomo_014'	43
15	'Grassano_007'	'Grassano_015'	43
16	'Sprovera_006'	'Sprovera_015'	43
17	'Triunfo_006'	'Triunfo_011'	43
18	'Capece_002'	'Capece_009'	44
19	'Celi_002'	'Celi_019'	44
20	'Fruggiero_006'	'Fruggiero_007'	44
21	'Cogliandro_001'	'Cogliandro_006'	45
22	'Coiro_004'	'Coiro_012'	45
23	'ColangeloMariagiulia_0...'	'ColangeloMariagiulia_01...	45
24	'Fruggiero_006'	'Fruggiero_019'	45
25	'Pepe_007'	'Pepe_017'	45
26	'Salerno_009'	'Salerno_015'	45
27	'Celi_008'	'Celi_019'	46

Figura 8.12 – Sezione della tabella genuini ordinata per Score crescente

Nello specifico prendiamo in esame i template ‘Grassano_07’ e ‘Grassano_015’ e focalizziamoci sulla loro sovrapposizione prima e dopo il processo di allineamento.

La Figura 8.13 mostra il confronto tra i due template prima dell'applicazione delle trasformazioni derivanti dal processo di matching. Nella Figura 8.14, invece, è rappresentato il risultato dopo il matching. Si può osservare come la procedura abbia sovrapposto con precisione le vene principali, mentre per le vene secondarie, che già in partenza presentano alcune differenze strutturali, ha permesso solo un corretto orientamento ma non una sovrapposizione completa. Questo spiega il punteggio finale di 0.43.

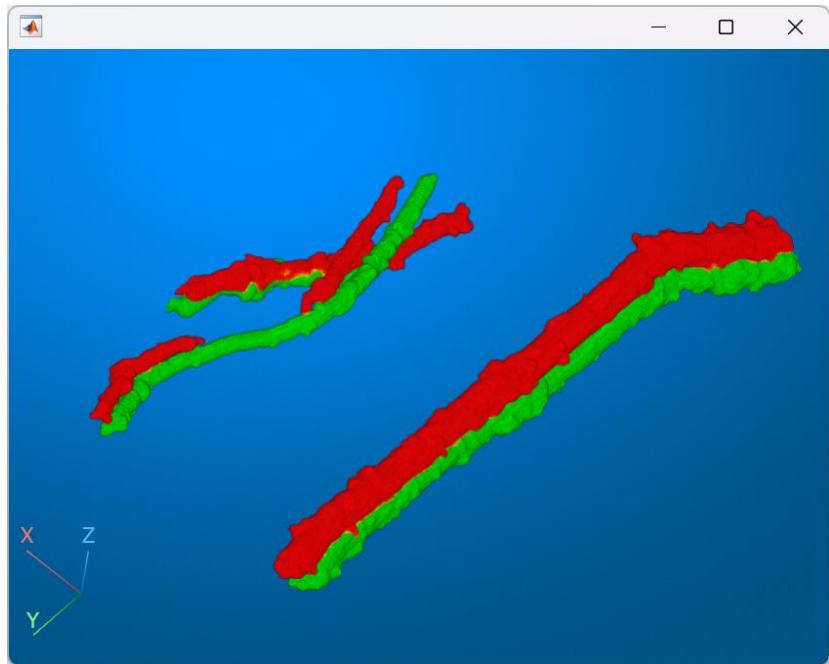


Figura 8.13 – Confronto PRE-matching fra i template ‘Grassano_07’ in rosso e ‘Grassano_015’ in verde.

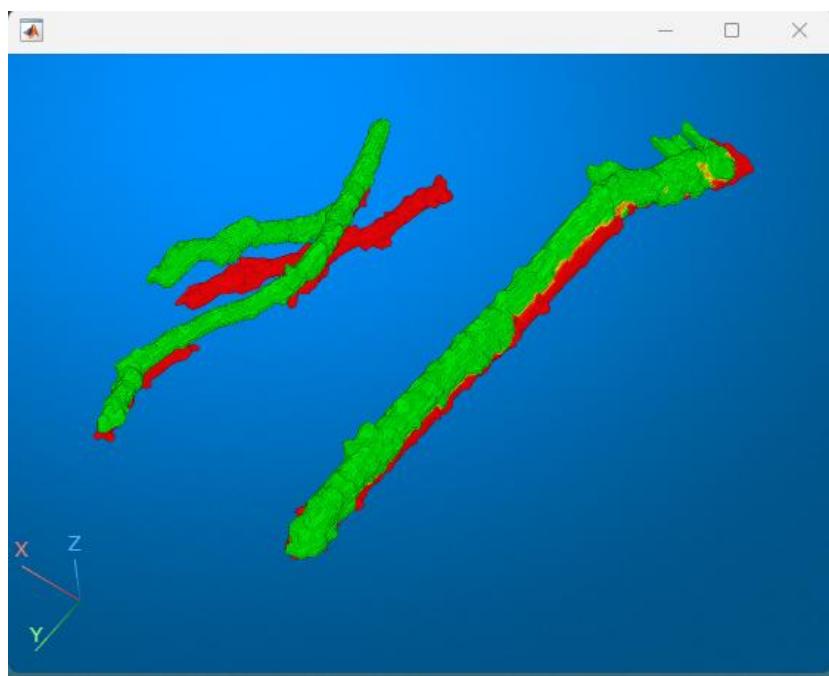


Figura 8.14 – Confronto POST-matching fra i template ‘Grassano_07’ in rosso e ‘Grassano_015’ in verde.

I limiti, appena descritti, comportano un avvicinamento delle curve di confronto tra score genuini e impostori, con un conseguente aumento dell'Equal Error Rate (EER) e quindi una riduzione dell'efficacia complessiva del sistema.

Nonostante questo un errore minore del 2% rimane piuttosto basso ed indica buone performance del sistema biometrico a conferma della bontà del template estratto.

9 CONCLUSIONI

9.1 LIMITI DEL METODO DI ESTRAZIONE

Il metodo proposto soffre di alcune limitazioni dovute soprattutto alle assunzioni fatte all'interno della trattazione.

9.1.1 Assunzioni

Considerando il fatto che il metodo è in grado di ricavare le vene esili in superficie e quelle robuste in profondità purtroppo non si comporta bene quando ciò non accade. Nel caso in cui un utente abbia delle vene profonde con valori di intensità più alti della media l'estrazione del template non sarà ottimale.

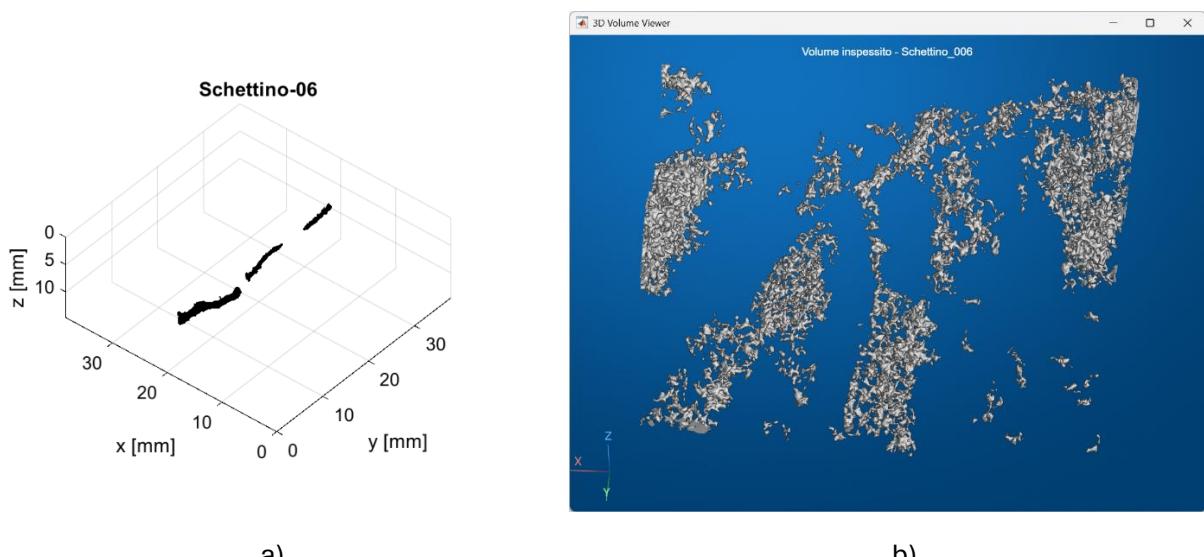


Figura 9.1 – a) Esempio template finale incompleto b) Risultato volume binarizzato per un template incompleto

Come possiamo vedere dalla Figura 9.1 (a) per questo utente il template estratto è di scarsa qualità, Il problema nasce dal fatto che l'utente in questione non ricade nella prima assunzione. Esplorando il volume originale si nota che la vena principale profonda ha delle intensità chiare tipiche delle vene superficiali. È evidente che durante la fase di binarizzazione la soglia elevata non ha raccolto i voxel venosi pregiudicando il template finale come si può vedere dalla Figura 9.1 (b)

9.1.2 Distanza minima vene-palmo

Un altro problema deriva dal calcolo della distanza minima tra le vene e il palmo. Seppur vero che la distanza minima che intercorre tra il palmo e i primi voxel pari a zero corrisponde a una vena questo non permette di individuare delle vene che si trovano al di sotto di questa distanza. Quindi se ci sono più vene profonde a livelli diversi prenderemo in considerazione solo quella più in superficie tagliando le altre. Riportiamo un esempio di questo caso nella figura che segue.

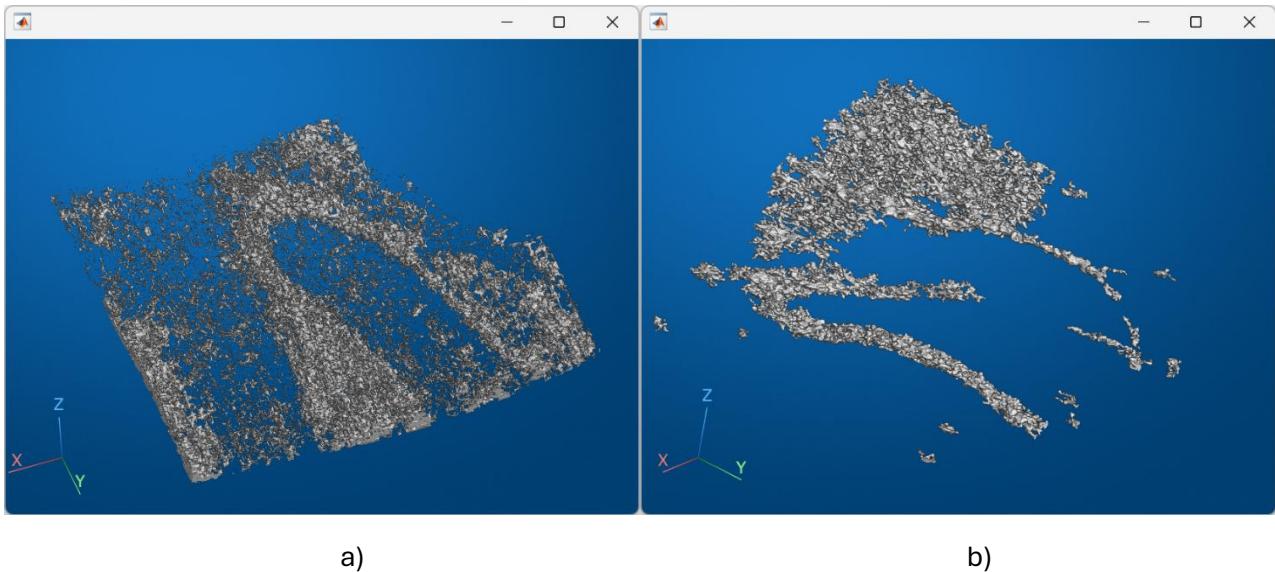


Figura 9.2 - a) Volume post calcolo vecFineBin b) volume post calcolo vecMinDistVP

Dal primo volume binarizzato (a) si nota come la vena sulla destra abbia uno sviluppo maggiore lungo l'asse y ma questa viene tagliata (b) perché si prolunga sotto la vena principale quindi il livello di fine binarizzazione la esclude.

9.1.3 Elimina rumore

Come detto in precedenza il classificatore tende ad avere più falsi positivi che falsi negativi infatti capita spesso che alcune componenti di rumore non vengano eliminate per poi propagarsi nella fase di estrazione e pregiudicare la qualità del template finale. Nonostante la funzione `eliminaRumore`, capita in alcuni casi che le cc di rumore permangono e il template finale risulta fallato. Di seguito vengono riportati alcuni esempi di template con delle componenti di rumore.

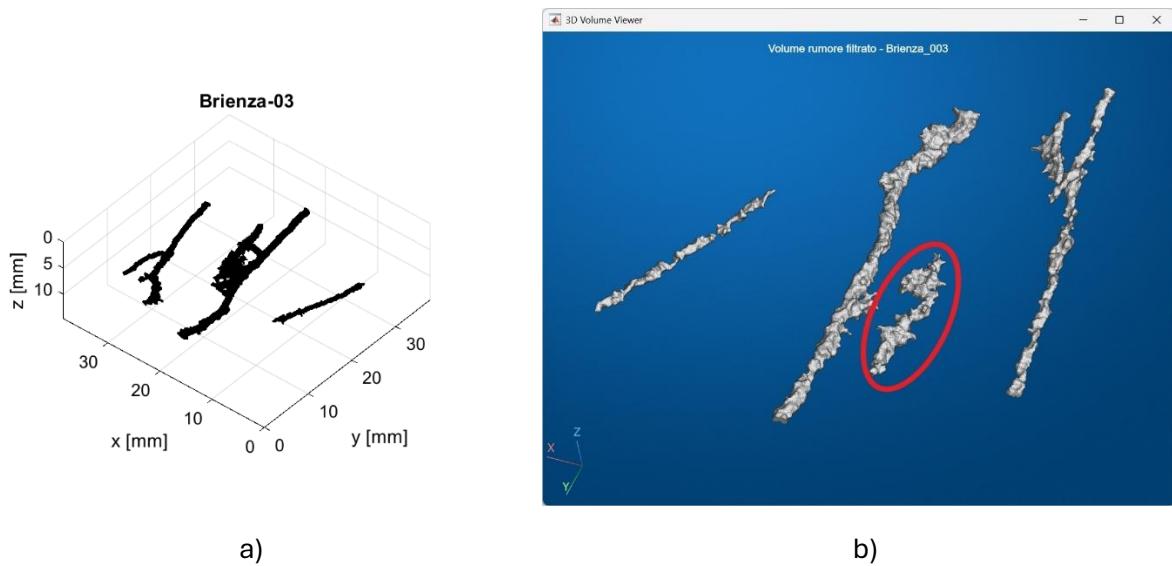


Figura 9.3 - a) Esempio template non valido b) Esempio di cc che non è stata eliminata da `eliminaRumore`

In Figura 9.3 (b) viene riportato un esempio in cui il rumore non viene eliminato. Come si può vedere dalla Figura 9.3 (a) questo rumore rimane nel template finale facendolo risultare non valido ai fini del matching. Questi tipi di template sono purtroppo da scartare difatti il template incriminato è presente nella lista ‘listaTemplateEsclusi.xlsx’

Inoltre il filtraggio iterativo non sempre riesce a separare le vene dal rumore, questo perché la procedura ricorsiva viene eseguita per un massimo di 3 volte. Quindi nel caso di un rumore connesso fortemente con la vena 3 tentativi non saranno sufficienti a separarla, di conseguenza comparirà anche nel template finale.

9.1.4 Affina vene

Un'ultima criticità risiede nella procedura di affinamento. Tale procedura è sicuramente necessaria ma a volte porta anche a degli errori. Se è vero che le aperture e i filtri delle cc dei vari BSCAN vanno a separare le vene dal rumore è anche vero che portano a distorcere il rumore stesso e fargli assumere, in rari casi, le sembianze di una vena. Componente che poi non può essere più rimossa dal template. Ovviamente anche questo tipo di template è da scartare.

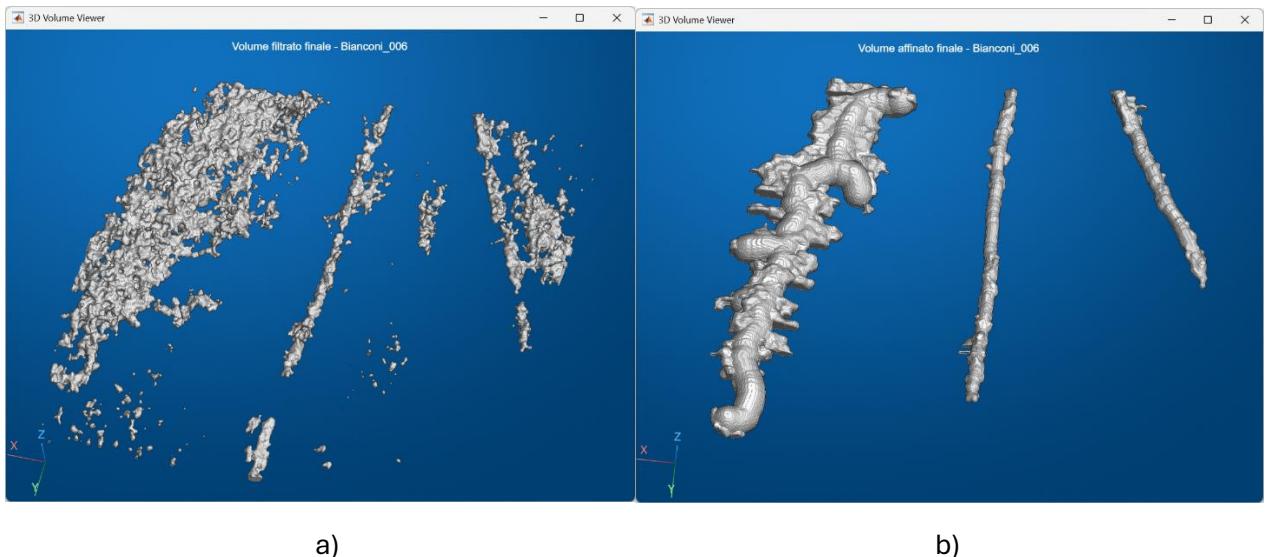


Figura 9.4 – a) Risultato post filtraggio b) Risultato post affinamento

9.2 CRITICITÀ NEL MATCHING

Il metodo di matching proposto soffre di alcune limitazioni, principalmente legate ai tempi computazionali e all'utilizzo di volumi di dimensioni significative. Le principali criticità possono essere riassunte come segue:

- 1. Alto consumo computazionale:** Il metodo di `matching3Dtr` richiede in input l'intero volume tridimensionale dei template, evitando l'uso di proiezioni 2D. Sebbene questa scelta consenta di ottenere allineamenti più precisi, l'elaborazione di volumi 3D di grandi dimensioni, come nel nostro caso (650x650x320), comporta un elevato consumo di risorse computazionali. Di conseguenza, non è stato possibile eseguire il matching su tutto il database né su tutte le 20 acquisizioni di un singolo utente, a causa dei significativi tempi di calcolo richiesti.
- 2. Sensibilità alle piccole differenze strutturali:** Il modello risulta sensibile anche a minime differenze strutturali tra i template. Anche lievi disallineamenti possono causare imprecisioni negli angoli di Eulero rendendo poco efficace l'allineamento calcolato. Sebbene la procedura iterativa aiuti a migliorare l'allineamento, la necessità di molteplici iterazioni comporta un aumento significativo del costo computazionale.
- 3. Score impostori elevati:** L'approccio di matching tenta il più possibile di allineare template simili per aumentare lo score dei genuini ma ciò porta a falsi positivi, dove template

appartenenti a utenti diversi, ma con vene simili o in posizioni vicine, ottengono score elevati non rappresentativi della vera somiglianza.

4. **Difficoltà nel raggiungere score genuini elevati:** Il raggiungimento di score genuini superiori a 0.8 è complesso, in quanto allineare perfettamente i volumi in tutte le direzioni risulta difficile a causa della loro variabilità e complessità.

9.3 POSSIBILI SVILUPPI FUTURI

Per migliorare il sistema biometrico proposto si potrebbe provare a:

- Provare diversi valori per i parametri e le soglie impostate durante le varie fasi di estrazione del template.
- Cambiare la logica dietro il calcolo del livello di fine binarizzazione. In merito a questo si potrebbe condensare il doppio passaggio del calcolo della fine binarizzazione in un unico step magari combinando le due tecniche o usando solo la procedura iterativa.
- Aggiungere uno step di preprocessing in cui si va ad applicare una tecnica di filtraggio avanzata come filtro SRAD o filtro di LEE per ridurre il rumore e migliorare il riconoscimento dei pattern venosi.
- Migliorare la procedura di matching sostituendo la logica presentata per allineare i template con un processo di *image registration*. Questo eviterebbe le tante trasformazioni iterative ma potrebbe addirittura peggiorare i tempi di esecuzione.
- Infine uno sviluppo necessario è sicuramente un'ottimizzazione del codice. Il metodo così realizzato è molto complesso e questo porta a tempi elevati sia per estrarre i template che per calcolare lo score.

BIBLIOGRAFIA

- [1] Fujitsu Computer Products of America, «Palm Vein Pattern Authentication,» 2006.
- [2] J.-G. Wang, W.-Y. Yau, A. Suwandy e E. Sung, «Conferences >2007 IEEE Conference on Compu...,» *IEEE*, 2007.
- [3] R. Scavone e A. Luongo, «Fusion Palm-Vein,» *Unibas*, 2022.
- [4] A. Vizzuso, «Tecniche di estrazione del pattern venoso palmare,» *Unibas*, 2021.