



Politecnico di Torino

FACOLTÀ DI INGEGNERIA MECCANICA
Corso di Laurea Magistrale in Ingegneria Meccanica indirizzo Automazione

TESI DI LAUREA MAGISTRALE

**Sviluppo di strumenti software e componenti hardware
per la valutazione pre e post operatoria
in ambito maxillo-facciale.**



Candidato:
Cecilia Pontin
Matricola 244379

Relatore:
Prof. Sandro Moos

Ai miei genitori
e
ai miei fratelli

Abstract

Il presente elaborato si pone due obiettivi: il primo consiste nello sviluppo di strumenti software per l'ausilio in interventi maxillo-facciali, il secondo riguarda la progettazione di componenti e programmi per il miglioramento delle funzionalità di una stampante 3D impiegata durante tali interventi.

L'iter delle operazioni, alcune delle quali prevedono la sostituzione di parti di osso con placche metalliche fissate mediante viti, si divide in una fase di pianificazione ed una fase post-operatoria di confronto tra il risultato ottenuto e quello programmato. Nella prima fase, uno dei problemi connessi all'inserimento dei perni è la valutazione dello spessore dell'osso nel punto di foratura. Si è quindi sviluppato un programma che, data una scansione del cranio, simuli la foratura, permettendo così di individuare il punto di applicazione più idoneo per ciascun perno. La fase post operatoria prevede invece il confronto tra i referti degli esami di controllo eseguiti prima e dopo l'operazione. Questo raffronto, definito registrazione, prevede l'allineamento delle scansioni dei crani e può essere eseguito mediante diversi software. Al fine di valutare vantaggi e limiti di ciascuno, nel Capitolo3 si è fornita una classificazione dei differenti metodi di allineamento e si sono confrontate alcune sovrapposizioni di crani ottenute mediante tre differenti programmi. Il capitolo si conclude con una valutazione quantitativa della bontà della registrazione ottenuta mediante il calcolo dello scostamento tra le superfici, eseguito dal programma Geomagic Wrap.

L'ultima parte dell'elaborato è invece dedicata al miglioramento del funzionamento della stampante 3D, con tecnologia fused deposition modeling (FDM), utilizzata per la stampa di ricostruzioni del cranio. I limiti riscontrati durante l'utilizzo consistono nella determinazione della quantità di filo consumata, nella carbonizzazione del filo nell'estrusore al termine della bobina o in caso di bloccaggio della stessa.

Si sono quindi progettati una serie di dispositivi allo scopo di risolvere tali

problemi. Il capitolo presenta i componenti progettati, nonché la selezione a catalogo dei sensori. L'integrazione con la stampante, ottenuta mediante un Raspberry Pi, ha previsto lo sviluppo di un programma python in grado di raccogliere i dati dai sensori ed effettuare le dovute misure. Il software comunica a sua volta con un'interfaccia grafica che permette all'utente di modificare i parametri di funzionamento e di controllare lo stato del sistema in tempo reale.

Indice

Introduzione	1
1 Stato dell'arte	4
2 Modulo 3D Slicer per la valutazione dello spessore osseo	7
3 Confronto tra referti pre e post operatori	18
3.1 Metodi di registrazione	18
3.1.1 Classificazione basata sulle tecniche di registrazione . .	19
3.1.2 Classificazione basata sul tipo di Trasformazione	21
3.2 Confronto di metodi di registrazione di due sequenze di im- magini mediche digitali	26
3.3 Confronto di software per la registrazione di ricostruzioni tri- dimensionali pre e post operatorie	62
3.3.1 3D Slicer	62
3.3.2 Allineamento mediante SolidWorks	85
3.3.3 Registrazione mediante ProPlan	101
3.3.4 Conclusioni	111
4 Progettazione dei componenti di supporto per la stampante	
FDM	120
4.1 Progettazione del sistema per la misura del consumo di filamento	122
4.1.1 L'encoder	122
4.1.2 Analisi del sistema di misurazione del consumo di filo .	125
4.2 Progettazione del sistema di rilevazione dell'esaurimento del filo	127
4.2.1 Il fotointerruttore	127
4.2.2 Progettazione del supporto	137
4.3 Dispositivo per il rilevamento del bloccaggio del filo	138
4.3.1 Il sensore reed	139
4.3.2 Analisi del sistema di rilevamento del bloccaggio del filo	143
4.4 Selezione del relè per lo spegnimento della stampante	147

4.5	Programma per la lettura dello stato dei sensori	149
4.5.1	Il main e le variabili globali	149
4.5.2	La funzione setup	150
4.5.3	Il loop	152
4.5.4	Le interrupt service routine	154
4.5.5	L'interfaccia grafica	157
Appendice A		159
A.1	Anatomia del cranio	159
Appendice B		162
B.1	Disegni Tecnici	162
B.1.1	Sistema per la misura del consumo del filamento	162
B.1.2	Sistema di rilevazione dell'esaurimento del filo	172
B.1.3	Dispositivo per il rilevamento del bloccaggio del filo	174
Appendice C		179
C.1	Programma per la lettura dello stato dei sensori	179
C.2	Programma per la realizzazione dell'interfaccia grafica	183
C.2.1	Listato per l'aggiornamento dei parametri riguardanti il filamento	187
Appendice D		189
D.1	Programma per la valutazione dello spessore osseo	189
Appendice E		204
E.1	Programma per l'esecuzione della registrazione basata sull'in- dividuazione dei landmarks	204
Appendice F		210
F.1	Programma per la stima della variazione dell'inclinazione del capo nei referti confrontati	210
Ringraziamenti		214
Bibliografia		215

Elenco delle figure

1	Taglio sagittale della TAC e ricostruzione tridimensionale della TAC.	2
2	Ultimaker3 [16]	3
1.1	Classificazione dei metodi di registrazione proposta nello studio di Aristeidis Sotiras.	5
1.2	Classificazione dei metodi di registrazione proposta nello studio di Ahmed El-Gamal [11].	6
2.1	Interfaccia grafica del modulo <i>3D Slicer</i> sviluppato per coadiuvare la scelta del posizionamento dei fori.	7
2.2	Immagini raffiguranti tre sezioni del cranio e le corrispondenti sezioni del primo foro da eseguire sul paziente A.	10
2.3	Vista 3D del cilindro che rappresenta il primo foro da eseguire sulla TAC del paziente A.	10
2.4	Immagini raffiguranti tre sezioni del cranio e le corrispondenti sezioni del secondo foro sul paziente A.	11
2.5	Vista 3D del cilindro che rappresenta il secondo foro da eseguire sulla TAC del paziente A.	12
2.6	SIImmagine raffigurante tre sezioni del cranio e le corrispondenti sezioni del terzo foro sul paziente A.	13
2.7	Vista 3D del cilindro che rappresenta il terzo foro da eseguire sulla TAC del paziente A.	13
2.8	Ricostruzione 3D del cranio del paziente A, comprensiva dei tre fori da realizzare.	14
2.9	Confronto tra la posizione pianificata del primo foro e la collocazione effettiva della vite inserita.	15
2.10	Confronto tra la posizione pianificata del secondo foro e la posizione effettiva della vite inserita.	15
2.11	Confronto tra la posizione pianificata del terzo foro e la collocazione effettiva della vite inserita.	16

3.1	Interfaccia grafica del modulo <i>Transforms</i> del software <i>3D Slicer</i> mediante la quale può essere eseguita la sovrapposizione manuale.	19
3.2	Rappresentazione grafica della rotazione, pari all'angolo θ , intorno all'asse z di un generico punto P.	23
3.3	Raffigurazione delle tre rotazioni principali effettuabili dal capo[15].	27
3.4	Immagine descrittiva del metodo di individuazione del Piano di Francoforte, fonte: <i>www.researchgate.net</i>	28
3.5	Ricostruzione della conformazione del cranio prima dell'operazione con indicati i punti utilizzati per creare il Piano di Francoforte e il piano stesso.	29
3.6	Ricostruzione della conformazione del cranio dopo l'operazione con indicati i punti utilizzati per creare il piano auricolare-orbitario e il piano stesso.	30
3.7	Misura grafica dell'angolo generato tra i piani auricolare-orbitari.	32
3.8	Sovrapposizione nella vista assiale di un taglio delle TAC del paziente A ottenuta mediante registrazione automatica.	34
3.9	Sovrapposizione nella vista sagittale di un taglio delle TAC del paziente A ottenuta mediante registrazione automatica.	35
3.10	Sovrapposizione nella vista coronale di un taglio delle TAC del paziente A ottenuta mediante registrazione automatica.	36
3.11	Vista frontale della ricostruzione 3d della sovrapposizione dei due crani ottenuta mediante metodo automatico.	37
3.12	Vista laterale della ricostruzione 3d della sovrapposizione dei due crani ottenuta mediante metodo automatico.	37
3.13	Viste assiale e sagittale della posizione del fiducial F1 sulla TAC pre operatoria del paziente A.	38
3.14	Viste assiale e coronale della posizione del fiducial F2 sulla TAC pre operatoria del paziente A.	39
3.15	Viste assiale e sagittale della posizione del fiducial F3 sulla TAC pre operatoria del paziente A.	40
3.16	Viste assiale e sagittale della posizione del fiducial F4 sulla TAC pre operatoria del paziente A.	41
3.17	Sovrapposizione di un taglio, nella vista assiale, ottenuta mediante registrazione basata sull'individuazione di landmarks sulle TAC del paziente A.	42
3.18	Sovrapposizione di un taglio, nella vista sagittale, ottenuta mediante registrazione basata sull'individuazione di landmarks sulle TAC del paziente A.	43

3.19	Sovrapposizione di un taglio, nella vista coronale, ottenuta mediante registrazione basata sull'individuazione di landmarks sulle TAC del paziente A.	44
3.20	Ricostruzione del cranio del paziente B prima dell'operazione con indicati i punti utilizzati per creare il Piano di Francoforte e il piano stesso.	46
3.21	Ricostruzione del cranio del paziente B dopo l'operazione con indicati i punti utilizzati per creare il Piano di Francoforte e il piano stesso.	46
3.22	Misura grafica dell'angolo generato tra i piani auricolo-orbitari.	47
3.23	Immagine della sovrapposizione manuale di due tagli corrispondenti, dei referti del paziente B, nella vista assiale.	48
3.24	Immagine della sovrapposizione manuale di due tagli corrispondenti, dei referti del paziente B, nella vista sagittale.	49
3.25	Immagine della sovrapposizione manuale di due tagli corrispondenti, dei referti del paziente B, nella vista coronale.	50
3.26	Immagine della sovrapposizione automatica di due tagli corrispondenti, dei referti del paziente B, nella vista assiale.	52
3.27	Immagine della sovrapposizione automatica di due tagli corrispondenti, dei referti del paziente B, nella vista sagittale.	53
3.28	Immagine della sovrapposizione automatica di due tagli corrispondenti, dei referti del paziente B, nella vista coronale.	54
3.29	Viste assiale e sagittale della posizione del fiducial F1 sulla TAC pre operatoria del paziente B.	55
3.30	Viste assiale e sagittale della posizione del fiducial F2 sulla TAC pre operatoria del paziente B.	56
3.31	Viste assiale e coronale della posizione del fiducial F3 sulla TAC pre operatoria del paziente B.	57
3.32	Viste assiale e coronale della posizione del fiducial F4 sulla TAC pre operatoria del paziente B.	58
3.33	Immagine della sovrapposizione semiautomatica di due tagli corrispondenti, dei referti del paziente B, nella vista assiale.	59
3.34	Immagine della sovrapposizione semiautomatica di due tagli corrispondenti, dei referti del paziente B, nella vista sagittale.	60
3.35	Immagine della sovrapposizione semiautomatica di due tagli corrispondenti, dei referti del paziente B, nella vista coronale.	61
3.36	Interfaccia grafica del modulo di <i>3D Slicer</i> sviluppato per l'esecuzione della registrazione basata sui landmarks.	63
3.37	Ricostruzione delle TAC pre operatoria del paziente A con indicata la posizione dei marcatori usati per l'allineamento.	64

3.38 Ricostruzione delle TAC post operatoria del paziente A con indicata la posizione dei marcatori usati per la sovrapposizione.	65
3.39 Immagine raffigurante la sovrapposizione, ottenuta mediante il modulo di 3D Slicer, delle ricostruzioni tridimensionali pre e post operatorie del cranio del paziente A.	66
3.40 Vista laterale sinistra della mappatura a colori del cranio del paziente A.	67
3.41 Vista frontale della mappatura a colori del cranio del paziente A.	68
3.42 Vista laterale destra della mappatura a colori del cranio del paziente A.	68
3.43 Vista frontale della mappa vettoriale del cranio del paziente A.	69
3.44 Ricostruzione delle TAC pre operatoria del paziente B con indicata la posizione dei marcatori usati per l'allineamento. . .	70
3.45 Ricostruzione delle TAC post operatoria del paziente B con indicata la posizione dei marcatori usati per la sovrapposizione.	71
3.46 Immagine raffigurante la sovrapposizione, ottenuta mediante il modulo di 3D Slicer, delle ricostruzioni tridimensionali pre e post operatorie del cranio del paziente B.	71
3.47 Vista laterale destra della mappatura a colori del cranio del paziente B.	72
3.48 Vista frontale della mappatura a colori del cranio del paziente B.	73
3.49 Vista laterale sinistra della mappatura a colori del cranio del paziente B.	73
3.50 Vista laterale della mappa a vettori del cranio del paziente B.	74
3.51 Ricostruzione delle TAC pre operatoria del paziente C con indicata la posizione dei marcatori usati per la sovrapposizione.	75
3.52 Ricostruzione delle TAC post operatoria del paziente C con indicata la posizione dei marcatori usati per la registrazione. .	76
3.53 Immagine raffigurante la sovrapposizione, ottenuta mediante il modulo di 3D Slicer, delle ricostruzioni tridimensionali pre e post operatorie del cranio del paziente C.	76
3.54 Vista laterale destra della mappatura a colori del cranio del paziente C.	77
3.55 Vista frontale della mappatura a colori del cranio del paziente C.	77
3.56 Vista laterale sinistra della mappatura a colori del cranio del paziente C.	78
3.57 Vista laterale della mappa a vettori del cranio del paziente C.	78

3.58 Ricostruzione delle TAC pre operatoria del paziente D con indicata la posizione dei marcatori usati per la sovrapposizione.	80
3.59 Ricostruzione delle TAC post operatoria del paziente D con indicata la posizione dei marcatori usati per la registrazione.	80
3.60 Immagine raffigurante l'allineamento, ottenuto mediante il modulo di 3D Slicer, delle ricostruzioni tridimensionali pre e post operatorie del cranio del paziente D.	81
3.61 Vista laterale destra della mappatura a colori del cranio del paziente D.	82
3.62 Vista frontale della mappatura a colori del cranio del paziente D.	82
3.63 Vista laterale sinistra della mappatura a colori del cranio del paziente D.	83
3.64 Vista frontale della mappa a vettori del cranio del paziente D.	83
3.65 Modello pre operatorio del cranio del paziente A con indicati i fiducial posizionati, la retta e il piano creati.	86
3.66 Ricostruzione tridimensionale post operatoria del cranio del paziente A con indicati i fiducial posizionati.	86
3.67 Sovrapposizione mediante <i>SolidWorks</i> delle ricostruzioni ricavate dai due referti del paziente A.	87
3.68 Vista laterale destra della mappa a colori della sovrapposizione dei crani del paziente A.	88
3.69 Vista frontale della mappa a colori della sovrapposizione dei crani del paziente A.	88
3.70 Vista laterale sinistra della mappa a colori della sovrapposizione dei crani del paziente A.	89
3.71 Modello pre operatorio del paziente B con indicati i marcatori inseriti, la retta e il piano generati.	90
3.72 Modello post operatorio del paziente B con indicati i marcatori inseriti.	90
3.73 Immagine raffigurante la sovrapposizione, ottenuta mediante <i>SolidWorks</i> , delle ricostruzioni tridimensionali pre e post operatorie del cranio del paziente B.	91
3.74 Vista laterale destra della mappa a colori della sovrapposizione dei crani del paziente B.	91
3.75 Vista frontale della mappa a colori della sovrapposizione dei crani del paziente B.	92
3.76 Vista laterale sinistra della mappa a colori della sovrapposizione dei crani del paziente B.	92

3.77 Ricostruzione tridimensionale ricavata dalla TAC pre operatoria del paziente C con indicati i fiducial inseriti, la retta e il piano generati.	93
3.78 Ricostruzione tridimensionale ricavata dalla TAC post operatoria del paziente C con indicati i fiducial posizionati.	94
3.79 Sovrapposizione mediante <i>SolidWorks</i> delle ricostruzioni ricavate dai due referti del paziente C.	94
3.80 Vista laterale destra della mappa a colori dell'allineamento dei modelli del paziente C.	95
3.81 Vista frontale della mappa a colori dell'allineamento dei modelli del paziente C.	95
3.82 Vista laterale sinistra della mappa a colori dell'allineamento dei modelli del paziente C.	96
3.83 Ricostruzione tridimensionale ricavata dalla TAC pre operatoria del paziente D con indicati i fiducial inseriti, la retta e il piano generati.	97
3.84 Ricostruzione tridimensionale ricavata dalla TAC post operatoria del paziente D con indicati i fiducial inseriti.	97
3.85 Sovrapposizione mediante <i>SolidWorks</i> delle ricostruzioni ricavate dai due referti del paziente D.	98
3.86 Vista laterale destra della mappa a colori dell'allineamento dei modelli del paziente D.	98
3.87 Vista frontale della mappa a colori dell'allineamento dei modelli del paziente D.	99
3.88 Vista laterale sinistra della mappa a colori dell'allineamento dei modelli del paziente D.	99
3.89 Sovrapposizione mediante <i>ProPlan</i> delle ricostruzioni ricavate dai referti del paziente A.	101
3.90 Vista laterale destra della mappa a colori dell'allineamento, ottenuto mediante <i>ProPlan</i> , dei modelli del paziente A.	102
3.91 Vista frontale della mappa a colori della sovrapposizione, ottenuta mediante <i>ProPlan</i> , dei modelli del paziente A.	102
3.92 Vista laterale sinistra della mappa a colori dell'allineamento, ottenuto mediante <i>ProPlan</i> , dei modelli del paziente A.	103
3.93 Immagine della sovrapposizione, ricavata mediante <i>ProPlan</i> , dei modelli pre e post operatori del cranio del paziente B.	104
3.94 Vista laterale destra della mappa a colori dell'allineamento ottenuto mediante <i>ProPlan</i> dei modelli del paziente B.	105
3.95 Vista frontale della mappa a colori dell'allineamento ottenuto mediante <i>ProPlan</i> dei modelli del paziente B.	105

3.96	Vista laterale sinistra della mappa a colori dell'allineamento ottenuto mediante <i>ProPlan</i> dei modelli del paziente B.	106
3.97	Risultato della sovrapposizione eseguita, per il paziente C.	107
3.98	Vista laterale destra della mappa a colori dell'allineamento ottenuto mediante <i>ProPlan</i> dei modelli del paziente C.	107
3.99	Vista frontale della mappa a colori dell'allineamento ottenuto mediante <i>ProPlan</i> dei modelli del paziente C.	108
3.100	Vista laterale sinistra della mappa a colori dell'allineamento ottenuto mediante <i>ProPlan</i> dei modelli del paziente C.	108
3.101	Risultato della sovrapposizione eseguita, per il paziente D.	109
3.102	Vista laterale destra della mappa a colori dell'allineamento ottenuto mediante <i>ProPlan</i> dei modelli del paziente D.	110
3.103	Vista frontale della mappa a colori dell'allineamento ottenuto mediante <i>ProPlan</i> dei modelli del paziente D.	110
3.104	Vista laterale sinistra della mappa a colori dell'allineamento ottenuto mediante <i>ProPlan</i> dei modelli del paziente D.	111
3.105	Immagine del modello pianificato in cui sono visibili i due centri di rotazione e le ossa zigomatiche, colorate di rosa, ruotate.	113
3.106	Rappresentazione sul modello pianificato della posizione dei fiducial, della retta e del piano.	114
3.107	Rappresentazione sul modello post operatorio della posizione dei marcatori.	115
3.108	Risultato della sovrapposizione eseguita mediante <i>SolidWorks</i> , per il paziente D.	115
3.109	Indicazione delle distanze tra le superfici nei punti scelti.	116
3.110	Valori delle distanze tra le superfici nei punti scelti.	116
3.111	Risultato della sovrapposizione eseguita con <i>ProPlan</i> , per il paziente D.	117
3.112	Indicazione delle distanze tra le superfici nei punti scelti.	117
3.113	Valori delle distanze tra le superfici nei punti scelti.	118
3.114	Risultato della sovrapposizione eseguita mediante <i>3DSlicer</i> , per il paziente D.	118
4.1	La stampante Ultimaker3.[16]	120
4.2	Vista complessiva del sistema di misurazione del consumo di filo.	122
4.3	Immagine di esempio della struttura interna dell'encoder[19].	123
4.4	Schema del circuito elettrico dell'encoder C14D32N-D3A3	124
4.5	Vista in sezione del sistema di regolazione del precarico tra la ruota ed il cuscinetto a sfere.	126

4.6	Sezione dell'assieme modellato in corrispondenza del piano medio della struttura.	126
4.7	Staffa di supporto dell'encoder.	126
4.8	Modello e sezione del componente inferiore del meccanismo di misura del consumo di filo.	127
4.9	Schema circuitale del fotointerruttore OPB-618.	128
4.10	Circuito elettrico completo per il collegamento del fotointerruttore alla scheda di controllo e al relè.	130
4.11	Circuito elettrico completo del fotointerruttore con indicate le sigle dei componenti usate nei calcoli.	132
4.12	Immagine del circuito stampato realizzato.	136
4.13	Disegno meccanico del fotointerruttore OPB618	137
4.14	Supporto per il fotointerruttore che rileva la presenza o assenza del filo.	138
4.15	Sistema per il rilevamento del bloccaggio del filo.	139
4.16	Schema circuitale del sistema per il rilevamento del bloccaggio del filo.	141
4.17	Vista in sezione del sistema per il rilevamento del bloccaggio del filo.	144
4.18	Vista isometrica e sezione della base.	144
4.19	Viste del corpo centrale mobile.	145
4.20	Disegno (ricavato dal datasheet) riportante le dimensioni meccaniche del sensore.	146
4.21	Raffigurazione del coperchio modellato.	146
4.22	Viste dei supporti ad L per la guida del filo.	147
4.23	Schema di un relè con contatti NA e NC[17].	147
4.24	Schema del circuito elettrico del relè.	148
4.25	Schermata principale dell'interfaccia grafica, riportante lo stato dei sensori e le informazioni sul consumo di filo.	157
4.26	Segnalazione dell'allarme nel caso dell'esaurimento del filo.	158
4.27	Segnalazione dell'allarme nel caso del bloccaggio del filo.	158
4.28	Form per l'aggiornamento della lunghezza della bobina e l'azzeramento del contatore associato alla quantità di filo consumata.	158
A.1	Vista frontale del cranio[20].	160
A.2	Vista laterale del cranio[20].	161

Elenco delle tabelle

2.1	Tabella riportante colore, etichetta e versore assegnati a ciascun foro.	9
2.2	Coordinate del punto di foratura e del punto che identifica la profondità e componenti del versore che definisce la direzione per il primo foro.	9
2.3	Coordinate del punto di foratura, del punto che identifica la profondità e componenti del versore che definisce la direzione di foratura per il secondo foro.	11
2.4	Coordinate del punto di foratura, del fiducial che identifica la profondità e componenti del versore che definisce la direzione di foratura per il terzo foro.	12
3.1	Coordinate dei punti che definiscono il Piano di Francoforte nel referto pre-operatorio.	28
3.2	Coordinate dei punti che definiscono il piano auricolo-orbitario nel referto post-operatorio.	29
3.3	Valori calcolati per le rotazioni intorno agli assi del sistema di riferimento.	31
3.4	Tabella indicante le scelte attuate per la definizione del riferimento, del test e del tipo di trasformazione da applicare.	33
3.5	Coordinate dei tre punti che definiscono il Piano di Francoforte nel referto pre-operatorio del paziente B.	45
3.6	Coordinate dei punti che definiscono il Piano di Francoforte nel referto post-operatorio del paziente B.	45
3.7	Valori calcolati per le rotazioni intorno agli assi del sistema di riferimento.	46
3.8	Tabella riportante le scelte attuate per eseguire la registrazione automatica dei referti del paziente B.	51
3.9	Tabella riportante l'indicazione della posizione dei marcatori individuati sui modelli 3D, ottenuti dai referti pre e post operatori del paziente A.	64

3.10	Scelte attuate, riguardo il volume di riferimento e quello di test, per l'esecuzione dell'analisi mediante <i>Geomagic Wrap</i> e valide per tutti i pazienti.	66
3.11	Scelte attuate nella definizione dell'intervallo di tolleranza degli scostamenti, per l'esecuzione dell'analisi mediante <i>Geomagic Wrap</i> e valide per tutti i pazienti.	67
3.12	Tabella riportante i valori degli indici statistici forniti da <i>Geomagic Wrap</i> sul paziente A.	69
3.13	Tabella riportante l'indicazione della posizione dei marcatori individuati sui modelli 3D, ottenuti dai referti pre e post operatori del paziente B.	70
3.14	Tabella riportante i valori degli indici statistici forniti da <i>Geomagic Wrap</i> sul paziente B.	74
3.15	Tabella riportante l'indicazione della posizione dei marcatori individuati sui modelli 3D, ottenuti dai referti pre e post operatori del paziente C.	75
3.16	Tabella riportante i valori degli indici statistici forniti da <i>Geomagic Wrap</i> sul paziente C.	79
3.17	Tabella riportante l'indicazione della posizione dei marcatori sui modelli 3D, ottenuti dai referti pre e post operatori del paziente D.	79
3.18	Tabella riportante i valori degli indici statistici forniti da <i>Geomagic Wrap</i> sul paziente D.	84
3.19	Tabella riportante l'indicazione della posizione dei marcatori sui modelli 3D, ottenuti dai referti pre e post operatori del paziente A.	85
3.20	Scelte attuate, riguardo il volume di riferimento e di test, per l'esecuzione dell'analisi mediante <i>Geomagic Wrap</i> e valide per tutti i pazienti.	87
3.21	Scelte attuate per la definizione dell'intervallo di tolleranza degli scostamenti, per l'esecuzione dell'analisi mediante <i>Geomagic Wrap</i> e valide per tutti i pazienti.	87
3.22	Tabella riportante i valori degli indici statistici forniti da <i>Geomagic Wrap</i> sul paziente A.	89
3.23	Tabella riportante i valori degli indici statistici forniti da <i>Geomagic Wrap</i> sul paziente B.	93
3.24	ITabella riportante i valori degli indici statistici forniti da <i>Geomagic Wrap</i> sul paziente C.	96
3.25	Tabella riportante i valori degli indici statistici forniti da <i>Geomagic Wrap</i> sul paziente D.	100

3.26	Tabella riportante i valori degli indici statistici forniti da <i>Geomagic Wrap</i> sul paziente A.	103
3.27	Tabella riportante i valori degli indici statistici forniti da <i>Geomagic Wrap</i> sul paziente B.	106
3.28	Tabella riportante i valori degli indici statistici forniti da <i>Geomagic Wrap</i> sul paziente C.	109
3.29	Tabella riportante i valori degli indici statistici forniti da <i>Geomagic Wrap</i> sul paziente D.	111
3.30	Tabella riportante per ogni paziente e per ogni software la distanza media e la deviazione standard ad essa associata. . .	112
4.1	Tabella sinottica riportante le principali caratteristiche della stampante Ultimaker 3.	121
4.2	Caratteristiche dell'encoder C14D32N-D3A3.	123
4.3	Caratteristiche elettriche del fotointerruttore OPB-616.	128
4.4	Caratteristiche elettriche del fotointerruttore OPB-618 per il dimensionamento delle resistenze.	131
4.5	Caratteristiche elettriche del sensore reed MK06-6-A.	140
4.6	Caratteristiche elettriche del relè G5LE-1.	148
4.7	Tabella riassuntiva degli stati dei canali dell'encoder e corrispondente valore per il calcolo del consumo di filo.	155

Introduzione

Negli ultimi anni si è assistito ad un sempre crescente diffondersi di tecnologie di prototipazione rapida in diversi settori. Tale diffusione è da imputarsi alla netta riduzione dei costi di queste tecnologie, nonché alle loro grandi potenzialità. Tra tutte, quelle relative all'additive manufacturing sono oggi tra le più studiate e in molti casi comportano bassi costi di adozione.

Uno dei settori di maggiore interesse è quello medico, dove la necessità di prodotti e procedure definiti ad hoc per ciascun paziente è rilevante: si assiste oggi alla realizzazione di protesi in stampa additiva a partire da scansioni 3D dell'arto del paziente. Le potenzialità di queste tecnologie si sono però dimostrate molto maggiori: alcuni gruppi stanno infatti indagando la possibilità di stampare bio-materiali, con l'obiettivo ultimo di produrre dei veri e propri organi viventi basati sulle caratteristiche desiderate [1].

Parallelamente all'aspetto hardware, grandi sforzi si sono fatti sul lato software per cercare di sfruttare al meglio le diverse tecniche di produzione. In particolar modo si è rivolto l'interesse ai software in grado di postprocessare le scansioni ottenute tramite tomografia computerizzata (TAC). A partire dai dati ottenuti dalla scansione, è possibile isolare le regioni di interesse e ottenere così una versione tridimensionale della stessa (fig.1), che può poi essere stampata, o può servire come base di partenza per modelli CAD più complessi e realizzati su misura per il paziente in questione.

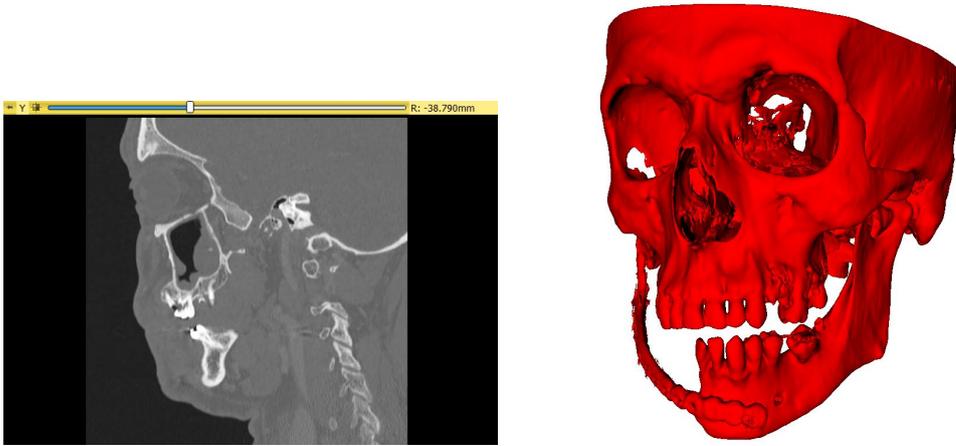


Figura 1: Taglio sagittale della TAC e ricostruzione tridimensionale della TAC.

Uno dei software utilizzati a questo scopo è *3D Slicer*, programma Open Source che, oltre a mettere a disposizione notevoli strumenti di postprocessing, consente agli utenti di integrare nuove funzionalità, utilizzando i linguaggi di programmazione C++ o python. Proprio la natura “aperta” del codice ha consentito un notevole sviluppo delle funzionalità del programma in tempi relativamente brevi, nonché la costituzione di un’ampia rete di supporto online in diversi forum. Questo lavoro prende le mosse da un’applicazione della stampa 3D di tipo fused deposition modeling (FDM) in campo maxillofaciale. In questo settore, le operazioni possono prevedere la rimozione di parti di osso e la loro sostituzione con placche metalliche fissate in posizione mediante viti. Il software *3D Slicer* viene impiegato per diverse valutazioni pre e post operatorie e per la realizzazione di modelli CAD che, prodotti mediante stampa 3D, servono da guida durante l’intervento stesso. Il presente lavoro di tesi si propone di inserirsi all’interno di questo contesto su due fronti: dal lato software, aggiungendo alcune funzionalità richieste dai chirurghi dell’ospedale Molinette di Torino a *3D Slicer*; dal lato hardware, progettando alcuni dispositivi per il miglioramento delle funzionalità della stampante 3D impiegata. Con riferimento a *3D Slicer*, un primo contributo consiste in un modulo con interfaccia utente per la valutazione dello spessore osseo, al fine di definire i punti di foratura ottimali per l’inserimento di perni di fissaggio a testa magnetica. Un secondo contributo riguarda invece la programmazione di un modulo per la registrazione delle scansioni TAC pre e post operatorie ed il confronto dei risultati ottenuti con quelli restituiti da alcuni software commerciali. Per quanto riguarda la stampante 3D, una Ultimaker 3 della casa Ultimaker (fig.2), lo studio ha portato alla realizzazione di tre diversi

dispositivi, ciascuno legato ad un particolare aspetto del processo di stampa.

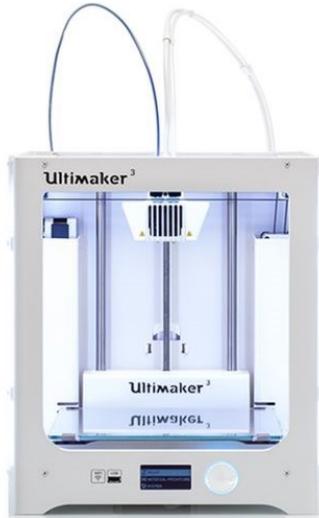


Figura 2: Ultimaker3 [16]

Il primo meccanismo riguarda il calcolo del materiale consumato dalla stampante, il secondo permette di spegnere la macchina quando il materiale disponibile, rappresentato dal filo avvolto in bobina, giunge al termine. In modo analogo, il terzo ed ultimo dispositivo, ha il compito di spegnere la stampante quando, per un non corretto avvolgimento del filo, questo rimane bloccato, interrompendo la stampa. Il primo aspetto è importante per limitare gli sprechi di materiale e consente di accertarsi che il materiale disponibile sia sufficiente per completare la stampa di un nuovo componente. Il secondo ed il terzo svolgono compiti di sicurezza, volti al prolungamento della vita utile dell'estrusore della Ultimaker 3. Quando il filo termina o rimane bloccato, il filamento presente nell'estrusore non può più essere espulso e carbonizza in loco. Ciò comporta la necessità di disassemblare l'estrusore per pulirlo, o, nel peggiore dei casi, la sostituzione dell'estrusore stesso. Al fine di fornire all'utente una chiara visione dello stato dei dispositivi, nonché del consumo di materiale, i diversi sensori sono stati connessi ad un Raspberry Pi, dove un programma python appositamente sviluppato ha il compito di raccogliere le informazioni ed analizzarle a fini diagnostici e di visualizzazione.

Capitolo 1

Stato dell'arte

Negli ultimi anni, i chirurghi hanno iniziato ad adottare la progettazione assistita da computer e software di modellazione (CAD/CAM) come ausilio alla pianificazione e all'esecuzione di complesse procedure cranio-maxillo-facciali (CMF). I diversi sistemi di pianificazione informatica, sviluppati, permettono sia la manipolazione virtuale delle regioni anatomiche deformate sia l'inserimento di nuove strutture anatomiche per correggere i difetti visibili nelle ricostruzioni 3D e nelle TAC [2]. Durante la fase di pianificazione, vengono quindi definite le procedure da attuare durante le operazioni, le quali possono essere scomposte in due fasi: la chirurgia demolitiva e quella ricostruttiva. La prima si occupa della rimozione della neoplastica, cioè delle cellule tumorali, mediante l'esecuzione di osteotomie simulate precedentemente al computer e la sostituzione della parte patologica con tessuti autologhi oppure protesi[3]. La chirurgia ricostruttiva, invece, comporta il fissaggio di tessuti autologhi, non compromessi dal tumore, alla parte sana delle ossa del cranio tramite placche in titanio. Le aree di maggiore interesse negli interventi maxillo-facciali riguardano il volume orbitale e la mandibola, riguardo a quest'ultima in letteratura sono presenti molteplici casi di ricostruzioni mandibolari eseguiti con lembi liberi di perone[4][5] o lembi liberi di cresta iliaca[6].

In questo tipo di intervento, svolgono un ruolo fondamentale le guide di taglio, le quali, progettate in precedenza mediante appositi software, permettono di ridurre i tempi dell'intervento e aumentare la precisione durante la realizzazione delle osteotomie. I benefici derivanti dalla programmazione virtuale sono riportati in molteplici studi [8][9], tra cui quello effettuato da Ganry et al[7]. Il quale valutò una differenza di precisione inferiore ad un millimetro nel confronto effettuato tra la ricostruzione mandibolare ottenuta nell'intervento, a seguito dell'utilizzo di guide di taglio progettata al computer, e quella programmata durante la fase di pianificazione dell'operazione.

Tra gli strumenti utilizzati nella fase di pianificazione dell'intervento, ma anche in quella post operatoria di confronto del risultato ottenuto con quello programmato, è possibile annoverare anche la registrazione. Essa permette infatti di considerare simultaneamente informazioni derivanti sia da differenti dispositivi di acquisizione di immagini mediche (TAC, MRI,...) sia dall'utilizzo di diversi protocolli. Uno dei problemi più dibattuti risulta essere quindi l'individuazione di efficaci metodi di registrazione, che permettono di stabilire corrispondenze spaziali tra le immagini acquisite durante gli esami. I metodi esistenti sono molteplici, un'esempio è riportato nello studio condotto da Aristeidis Sotiras[10], visibile nella Figura 1.1, il quale ha effettuato una classificazione basata sugli algoritmi adottati dai metodi per il confronto delle immagini.

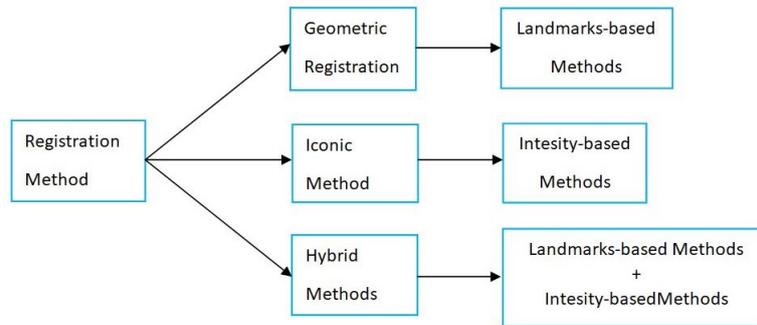


Figura 1.1: Classificazione dei metodi di registrazione proposta nello studio di Aristeidis Sotiras.

Il **Geometric Method** è basato sull'individuazione delle corrispondenze tra i landmarks, i quali devono essere disposti in posizioni anatomiche significative. Tuttavia, l'identificazione, sui referti medici, di punti di riferimento affidabili è un compito complesso e per questo è un argomento attivo di ricerca.[10]

L'**Iconic Method** valuta l'allineamento tra le immagini mediante criteri basati sull'intensità dei pixel. Rispetto al metodo geometrico, questo approccio permette di quantificare meglio e rappresentare con maggiore accuratezza il campo di deformazione.[10]

L'**Hybrid Method**, essendo una combinazione dei due metodi precedenti, sfrutta i vantaggi di entrambi per giungere ad una migliore soluzione del problema.[10]

Una classificazione molto più articolata dei metodi di registrazione, ripo-

ratta nella Figura 1.2, è invece descritta nello studio effettuato da Ahmed El-Gamal [11].

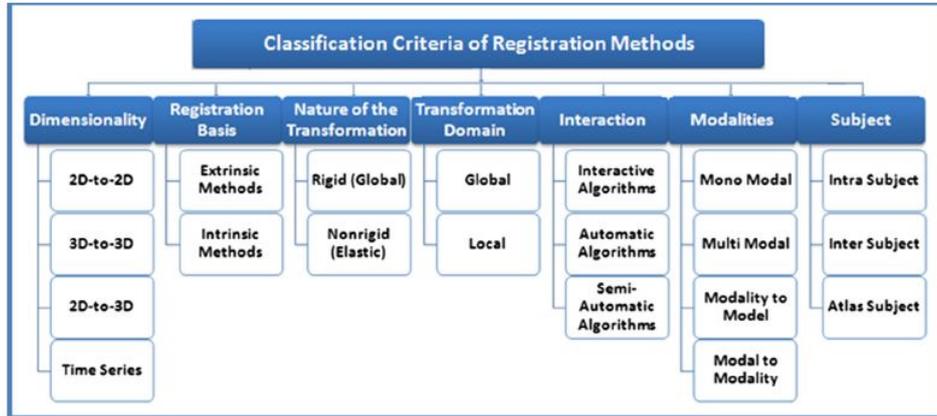


Figura 1.2: Classificazione dei metodi di registrazione proposta nello studio di Ahmed El-Gamal [11].

L'elevato numero di criteri per la scelta del metodo mostra quanto sia complessa l'individuazione della soluzione più efficace, dal momento che essa dipende da diversi fattori tra cui il livello di deformazione dell'immagine, le dimensioni spaziali degli elementi confrontati oppure l'algoritmo su cui si basa il processo di confronto. Sono stati quindi sviluppati, negli anni, diversi toolkit, i quali permettono di scegliere tra i differenti metodi di registrazione e di effettuare un confronto tra i risultati così ottenuti. Tra questi software di supporto, riportati anche da Ahmed El-Gamal nel suo elaborato, troviamo: ITK, l'open source più diffuso, Elastix, ANTs e Nifty Reg.

Nonostante le numerose ricerche compiute in questo ambito, gli sviluppi che si possono ancora effettuare nel campo della registrazione sono molteplici e funzione: delle tecnologie di acquisizione delle immagini mediche, delle tecniche usate, delle caratteristiche anatomiche delle zone rilevate e dei movimenti effettuati durante l'esecuzione dell'esame.

Capitolo 2

Modulo 3D Slicer per la valutazione dello spessore osseo

Le persone sottoposte ad interventi maxillo-facciali presentano delle deformità la cui eliminazione può comportare la sostituzione di parti del viso con placche in titanio, fissate all'osso mediante viti a testa magnetica. Queste devono alloggiare in fori il cui posizionamento è di fondamentale importanza e comporta l'individuazione di regioni dalla densità ossea sufficiente.

Per coadiuvare la pianificazione degli interventi chirurgici, è stato quindi sviluppato un programma, estensione del software gratuito *3D Slicer*, che permette di valutare, scelto il punto di foratura, se lo spessore dell'osso è adeguato per l'alloggiamento del perno. Il risultato dell'algoritmo consiste nella generazione di un cilindro colorato, quale rappresentazione 3D di un foro di pari dimensioni. L'interfaccia grafica del modulo sviluppato è riportata nella Figura 2.1.

The image shows a software interface for a 3D Slicer module. It is organized into several sections:

- Parameters:** Contains dropdown menus for 'Source points' (set to 'F') and 'Curve model' (set to 'VERDE2'). Below these are two sliders: 'Radius (mm)' with a value of 2.50 and 'Height (mm)' with a value of -6.00.
- Buttons:** A 'Generate Curve' button is located below the sliders.
- Coordinates:** A section labeled 'Point 1' with three input fields for x (37.35), y (48.55), and z (-8.88) in millimeters.
- Array:** A section labeled 'Elemnet_Array' with three input fields for x (0.10), y (-0.33), and z (0.94).
- Coordinates Second Point:** A label at the bottom of the array section.

Figura 2.1: Interfaccia grafica del modulo *3D Slicer* sviluppato per coadiuvare la scelta del posizionamento dei fori.

I parametri di ingresso dell'algoritmo, definiti dall'utente, sono le coordinate del foro e le componenti del vettore che fornisce direzione e verso di foratura dell'osso. Appositi campi permettono di definire il raggio e la profondità del foro. Per la rappresentazione del cilindro nella vista 3D, è necessario definire un secondo punto, ottenuto cliccando il pulsante "*Coordinates second point*", la cui generazione è ottenuta con la seguente procedura. Fissati il generico punto di applicazione P_1 , ossia il primo punto inserito dall'utente, di coordinate (x_1, y_1, z_1) , il vettore \vec{n} di componenti (n_x, n_y, n_z) e l'altezza h del cilindro, le coordinate del secondo punto sono:

$$x_2 = x_1 + \frac{n_x h}{\sqrt{n_x^2 + n_y^2 + n_z^2}}$$

$$y_2 = y_1 + \frac{n_y h}{\sqrt{n_x^2 + n_y^2 + n_z^2}}$$

$$z_2 = z_1 + \frac{n_z h}{\sqrt{n_x^2 + n_y^2 + n_z^2}}$$

L'operazione $n_i/|\vec{n}|$ (i=x,y,z) viene eseguita per assicurare la correttezza del risultato anche nel caso in cui il vettore inserito non abbia norma unitaria.

Come accennato, il risultato dello script consiste nella visualizzazione, sulla ricostruzione del cranio, di un cilindro la cui geometria rispecchia i parametri inseriti dall'utente. Sulle viste sagittale, coronale ed assiale vengono infine mostrate le corrispondenti viste in sezione del cilindro.

Per testare il modulo è stata utilizzata la TAC di un paziente, definito d'ora in avanti paziente A, sottoposto ad un intervento per la rimozione e sostituzione di una porzione dell'osso zigomatico destro con una placca in titanio, fissata mediante l'uso di viti a testa magnetica. La procedura standard richiederebbe al chirurgo di valutare, analizzando il referto, dove forare, senza nessun ausilio particolare. Utilizzando il modulo qui presentato invece, basta seguire la procedura di cui sopra perchè venga visualizzata la sezione del cilindro sulle immagini della TAC. Dato che il cilindro è la rappresentazione del foro, è sufficiente accertarsi che, sezione per sezione, esso sia sempre contenuto all'interno dell'osso. Il vantaggio è quindi rappresentato dall'immediatezza con cui si riesce a valutare se si possa forare o meno nel punto prescelto.

Per il paziente A, è stato necessario posizionare tre fori, con diametro di 5 mm, lungo l'orbita dell'occhio destro, due in corrispondenza della parte inferiore e uno situato nell'arco superiore. Per semplificare la loro individuazione, nelle immagini qui riportate, è stato assegnato ad ognuno di essi un colore differente, nella Tabella 2.1 è quindi riportata la nomenclatura utilizzata nell'analisi.

Fori	Colore	Punto di foratura	versore
1	Verde	V_1	\vec{v}
2	Giallo	G_1	\vec{g}
3	Rosa	R_1	\vec{r}

Tabella 2.1: Tabella riportante colore, etichetta e versore assegnati a ciascun foro.

Per ogni foro, sono quindi presentati, sotto forma di tabelle e nelle figure inserite, i risultati derivanti dall'analisi del referto pre operatorio.

Foro 1

Con il numero 1 viene indicato uno dei fori posizionati nella parte inferiore dell'orbita dell'occhio destro. I valori individuati per il versore \vec{v} e i punti V_1 e V_2 , mediante il quale viene definita la profondità del foro, sono riportati nella Tabella 2.2. La posizione di V_2 lungo l'asse z è determinata dal parametro *altezza*, modificabile da parte dell'utente. Nel caso considerato il valore scelto è 2.3 mm .

	X	Y	Z
Coordinate V_1	36.96 (mm)	49.82 (mm)	-12.48 (mm)
Coordinate V_2	36.73 (mm)	50.58 (mm)	-14.64(mm)
versore \vec{v}	-0.1	0.33	-0.94

Tabella 2.2: Coordinate del punto di foratura e del punto che identifica la profondità e componenti del versore che definisce la direzione per il primo foro.

La Figura 2.2 mostra alcuni ingrandimenti delle sezioni del primo foro individuato. Le viste assiali riportate sono in tutto tre e rappresentano rispettivamente il primo punto di contatto della fresa con l'osso (finestra rossa), la base del foro, indicata con V_2 (finestra verde), e una sezione intermedia a quelle precedenti (finestra gialla). Per far risaltare maggiormente le ossa del cranio, il contorno è stato rappresentato in rosso.

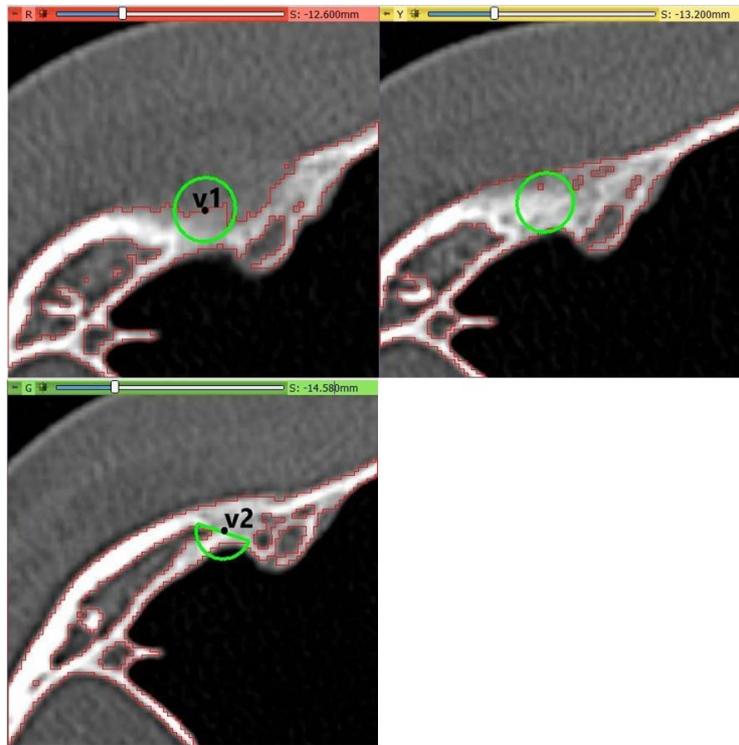


Figura 2.2: Immagini raffiguranti tre sezioni del cranio e le corrispondenti sezioni del primo foro da eseguire sul paziente A.

Come detto in precedenza, è possibile visualizzare il foro anche nel modello tridimensionale. Per facilitarne l'individuazione, nelle immagini l'altezza del cilindro è stata settata in modo tale da farlo sporgere dall'osso.

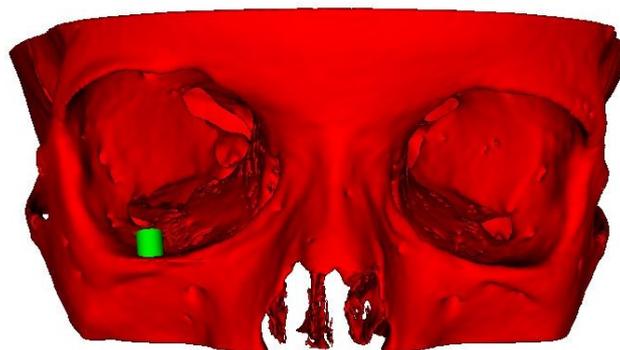


Figura 2.3: Vista 3D del cilindro che rappresenta il primo foro da eseguire sulla TAC del paziente A.

Foro 2

Il secondo foro, posizionato come il primo sull'arco inferiore dell'orbita è rappresentato in giallo. I valori ricavati considerando l'altezza del cilindro uguale a 5.70 mm sono riportati nella Tabella 2.3.

	X	Y	Z
Coordinate G_1	42.668 (mm)	45.191 (mm)	-10.499 (mm)
Coordinate G_2	42.154 (mm)	42.380 (mm)	-15.492 (mm)
versore \vec{g}	-0.09	-0.49	-0.86

Tabella 2.3: Coordinate del punto di foratura, del punto che identifica la profondità e componenti del versore che definisce la direzione di foratura per il secondo foro.

Anche in questo caso, le sezioni mostrate nella Figura 2.4 sono quelle corrispondenti al punto di inizio e fine foratura (G_1 e G_2) e ad un punto a profondità intermedia.

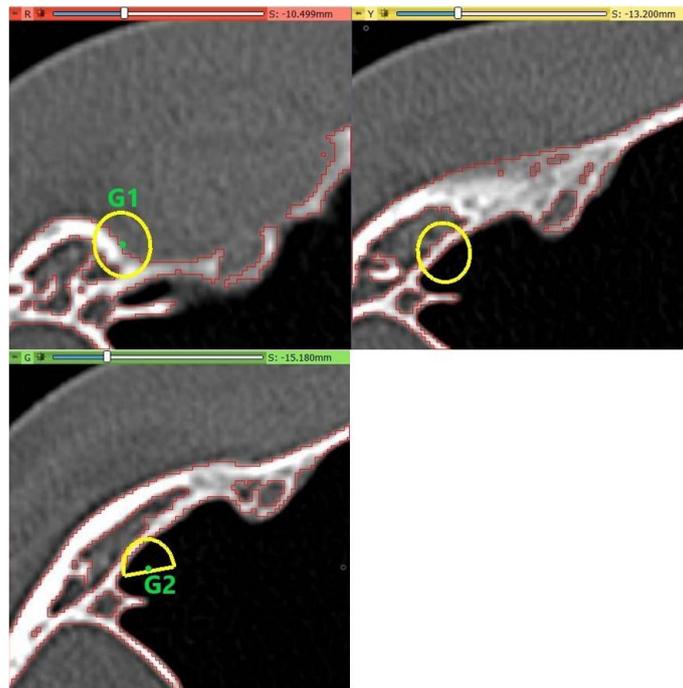


Figura 2.4: Immagini raffiguranti tre sezioni del cranio e le corrispondenti sezioni del secondo foro sul paziente A.

La rappresentazione nella ricostruzione tridimensionale di figura 2.5 facilita l'individuazione della posizione del cilindro, in questo caso di colore giallo.

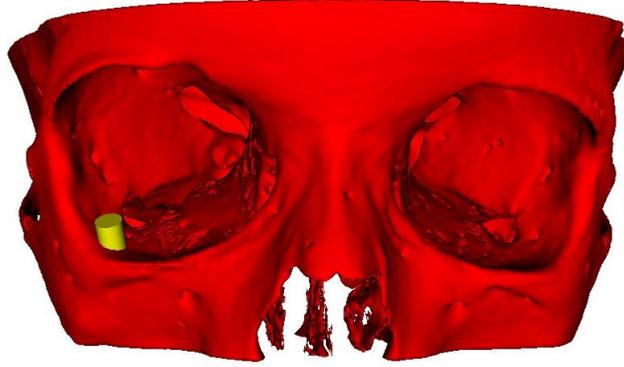


Figura 2.5: Vista 3D del cilindro che rappresenta il secondo foro da eseguire sulla TAC del paziente A.

Foro 3

Da ultimo, il terzo foro è stato collocato nell'arcata superiore dell'orbita dell'occhio. Il valore individuato questa volta per l'altezza è 3.7 mm , mentre le grandezze che definiscono il cilindro, di colore rosa, sono riportate nella Tabella 2.4.

	X	Y	Z
Coordinate R_1	36.658 (mm)	54.789 (mm)	26.713 (mm)
Coordinate R_2	36.215 (mm)	54.938 (mm)	30.380 (mm)
versore \vec{r}	-0.12	0.04	0.99

Tabella 2.4: Coordinate del punto di foratura, del fiducial che identifica la profondità e componenti del versore che definisce la direzione di foratura per il terzo foro.

Anche per il terzo foro sono riportati alcuni esempi raffiguranti le sezioni individuate (fig.2.6), mentre il modello tridimensionale posizionato nella ricostruzione del cranio è visibile nella Figura 2.7.

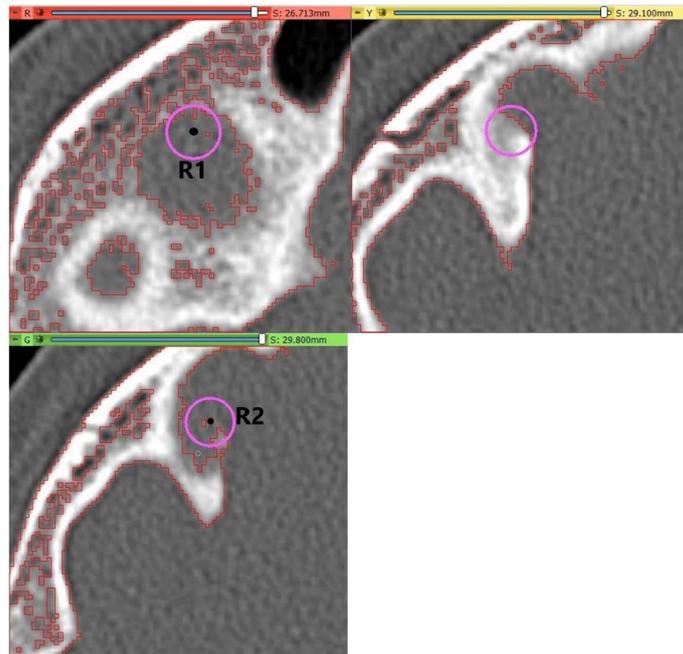


Figura 2.6: SImmagine raffigurante tre sezioni del cranio e le corrispondenti sezioni del terzo foro sul paziente A.



Figura 2.7: Vista 3D del cilindro che rappresenta il terzo foro da eseguire sulla TAC del paziente A.

Un ultimo controllo visivo, che permette di verificare l'assenza di interferenze tra i fori posizionati, può essere effettuato visualizzando contemporaneamente i cilindri sul modello 3D del cranio, come riportato nella Figura2.8. Con i risultati ottenuti, si può quindi realizzare un modello tridimensionale del cranio, comprensivo di fori, mediante additive manufacturing. Tale mo-

dello può servire da riferimento in sala operatoria, o può costituire la base per ulteriori analisi.

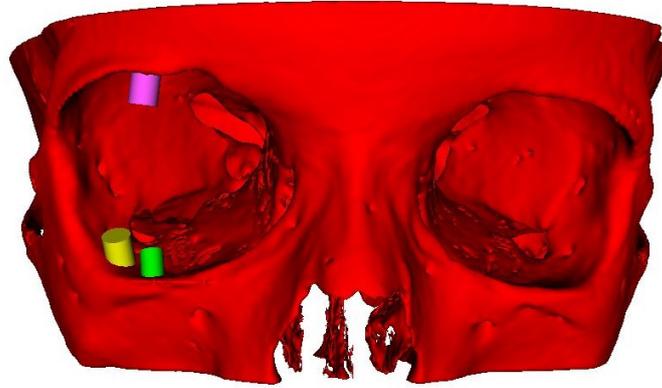


Figura 2.8: Ricostruzione 3D del cranio del paziente A, comprensiva dei tre fori da realizzare.

L'uso del programma realizzato non è limitato alla sola fase di pianificazione dell'intervento, ma può essere impiegato anche come strumento di verifica della corrispondenza tra il risultato programmato e quello effettivamente ottenuto. Per effettuare il raffronto, i pazienti vengono sottoposti ad un ulteriore esame di controllo, alcuni mesi dopo l'intervento. Prendendo in considerazione la TAC post operatoria del paziente A, eseguita ad una distanza temporale di due anni, è stato effettuato un allineamento dei due referti. La sovrapposizione così ottenuta ha permesso di confrontare la posizione delle viti inserite (le quali sono di colore bianco nella TAC) con quella individuata in precedenza durante la fase di pianificazione.

Per ogni foro, sono stati riportati quattro ingrandimenti: tre contenenti i tagli assiali visti in precedenza e uno raffigurante un taglio sagittale, che permette di confrontare la direzione di foratura con quella di inserimento della vite. Nelle seguenti immagini il referto visibile è quello effettuato dopo l'intervento, mentre in rosso sono rappresentate le ossa del cranio della TAC pre operatoria.

Foro 1

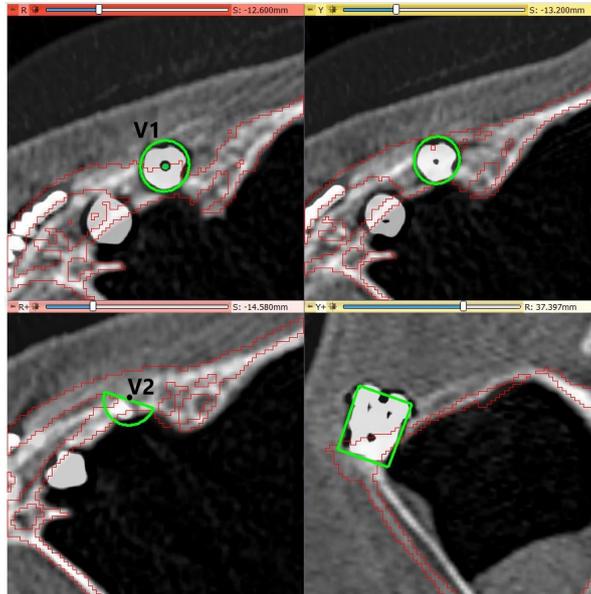


Figura 2.9: Confronto tra la posizione pianificata del primo foro e la collocazione effettiva della vite inserita.

Foro 2

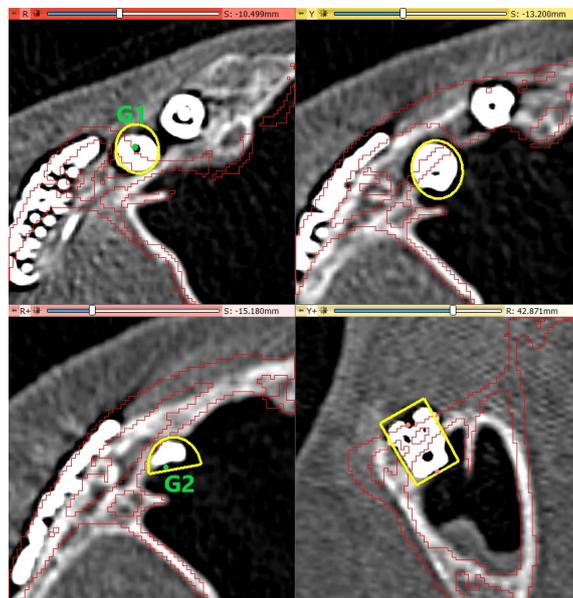


Figura 2.10: Confronto tra la posizione pianificata del secondo foro e la posizione effettiva della vite inserita.

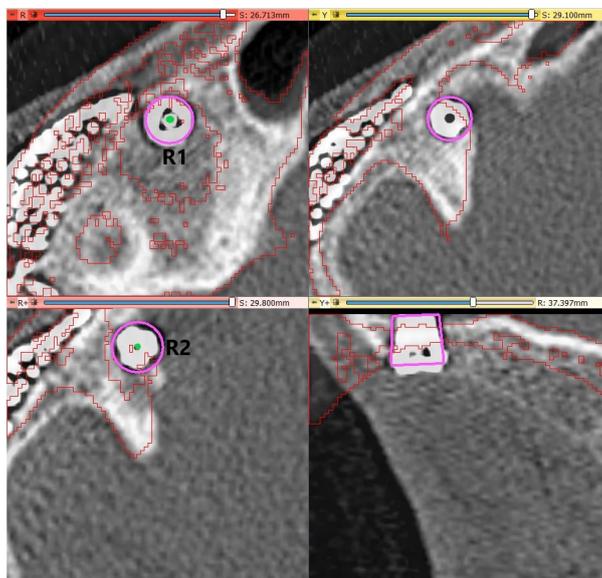
Foro 3

Figura 2.11: Confronto tra la posizione pianificata del terzo foro e la collocazione effettiva della vite inserita.

Come ausilio alla programmazione dell'intervento è stato sviluppato un modulo, la cui interfaccia grafica è visibile nella Figura 2.1, il quale permette di valutare se lo spessore, in corrispondenza del punto di foratura, è sufficiente per l'alloggiamento della vite. I parametri che vengono definiti dall'utente sono la posizione, la profondità e il raggio del foro. L'algoritmo del modulo restituisce invece come risultato un cilindro colorato, quale rappresentazione 3D di un foro di pari dimensioni, mentre sulle viste sagittale, coronale ed assiale vengono mostrate le corrispondenti viste in sezione del cilindro.

Il programma è stato prima testato sul referto pre operatorio del paziente A, per il quale sono stati collocati tre fori per l'inserimento delle viti per il fissaggio di una placca in titanio. Note le posizioni dei fori e le loro dimensioni, è stato generato per ognuno di essi il cilindro corrispondente, del quale è stata mostrata sia la rappresentazione tridimensionale nel modello, sia la

vista in sezione nei tagli assiali della TAC.

L'uso del modulo sviluppato non è limitato alla sola fase di pianificazione, ma permette anche il confronto tra il risultato ricavato dall'operazione e quello programmato. Nel caso in esame è stata quindi raffrontata, dopo aver effettuato la registrazione delle TAC pre operatoria e post operatoria, la posizione dei fori pianificata con quella ottenuta in sala operatoria.

Capitolo 3

Confronto tra referti pre e post operatori

Come già asserito nel capitolo 2, il paziente, generalmente, viene sottoposto a due esami di controllo, uno prima e uno dopo l'operazione, effettuati a distanza di mesi l'uno dall'altro. Il confronto tra i due referti acquisiti può avvenire mediante una sovrapposizione di due sequenze di immagini digitali, prese in momenti diversi, oppure attraverso l'allineamento delle ricostruzioni tridimensionali generate.

Questa sovrapposizione viene anche detta registrazione e permette di ricavare informazioni che un'analisi separata dei dati non fornirebbe, tra cui: la corrispondenza tra l'intervento eseguito e quello programmato, oppure la variazione dell'anatomia della zona interessata dall'intervento rispetto a quella originale.

3.1 Metodi di registrazione

Il problema della registrazione si verifica quando due immagini presentano lo stesso soggetto, ma le posizioni assunte sono differenti, generalmente questo succede se l'immagine viene scattata in tempi o con prospettive differenti o mediante dispositivi diversi. In ambito medico, generalmente, vengono confrontate due ricostruzioni tridimensionali oppure due successioni di immagini, una delle quali è definita Reference, o riferimento, mentre quella sottoposta a deformazione è detta Template, o test. I metodi di allineamento che possono essere adottati, sia per le immagini sia per i modelli tridimensionali, sono basati su algoritmi che possono essere catalogati in base alla tecnica usata (manuale, basata sull'intensità dei pixel, sull'individuazione dei landmarks, sulla superficie) oppure al tipo di trasformazione geometrica applicata.

3.1.1 Classificazione basata sulle tecniche di registrazione

Registrazione manuale

In questo tipo di allineamento l'utente cerca di effettuare una sovrapposizione delle immagini in modo visivo, il software utilizzato presenta quindi un'interfaccia grafica che permette di agevolare la manipolazione dei dati. Un esempio è riportato nella Figura 3.1, dove l'utente, modificando i valori dei parametri, può traslare e ruotare il Template cercando di sovrapporlo al Reference.

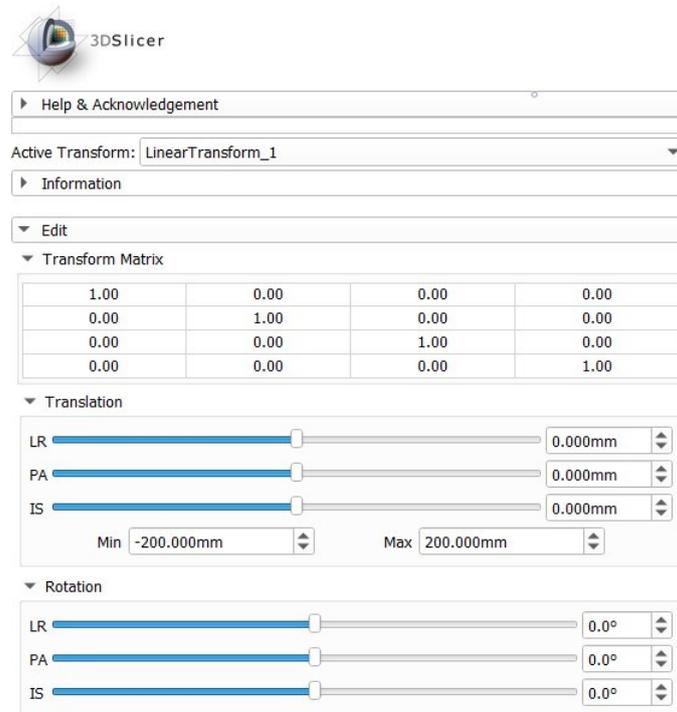


Figura 3.1: Interfaccia grafica del modulo *Transforms* del software *3D Slicer* mediante la quale può essere eseguita la sovrapposizione manuale.

Questa tecnica presenta però dei forti limiti tra cui l'accuratezza e il tempo necessario per ottenere un allineamento soddisfacente, entrambi dipendono dall'abilità dell'utente. La capacità dell'utente può essere però condizionata da fattori esterni, nella Sezione 3.2, è infatti osservato come, all'aumentare della differenza tra le inclinazioni del capo nei due referti acquisiti, la precisione ottenibile nella sovrapposizione si riduca sensibilmente.

Il vantaggio risiede invece nella capacità dell'utente di escludere automaticamente la zona che ha subito modifiche a causa dell'intervento.

Registrazione basata sull'intensità

L'allineamento ottenuto considerando l'intensità dei pixel è basato "sulla misura della similarità (o metrica) delle sezioni da confrontare e sull'ottimizzazione di questa misura, modificando i parametri della trasformazione. Per similarità si intende una grandezza che indica il grado di allineamento delle immagini"[12], questa grandezza può essere misurata come media della somma dei quadrati delle differenze delle intensità tra coppie corrispondenti di punti delle immagini. Date due immagini A e B il valore della Squared Sum Difference (SSD) è ottenuto mediante la seguente relazione:

$$MS(A, B) = \frac{1}{N} \sum_{i=1}^N (A_i - B_i^T)^2$$

con:

A_i intensità dell'i-esimo pixel dell'immagine A,
 B_i^T intensità dell'i-esimo pixel dell'immagine B trasformata,

Un limite di questa tecnica è legato alla necessità di utilizzo dello stesso dispositivo per acquisire entrambe le successioni di immagini; in campo medico, quindi, il paziente dovrebbe eseguire entrambi gli esami con lo stesso macchinario.

Registrazione effettuata mediante landmarks

La registrazione basata sui landmarks può essere definita nel modo seguente. Presi due insiemi di N punti corrispondenti e appartenenti una al riferimento e l'altra al test, viene definita una trasformazione T che minimizza la la distanza tra punti omologhi:

$$D[\varphi] = \sum_{i=1}^N \|x^{R,i} - \varphi(x^{T,i})\|_{R^d}^2$$

con D distanza tra due punti corrispondenti, associata alla trasformazione applicata, $x^{R,i}$ i-esimo punto del riferimento, $x^{T,i}$ i-esimo punto del test.

Questa tecnica prevede di fissare dei marcatori, definiti anche fiducial o landmarks, che possono essere di due tipi:

- Hard: posizionati sul paziente prima che venga effettuato l'esame[13],
- Soft: individuati sui referti dopo la loro acquisizione[13].

Il procedimento, per effettuare questo tipo di allineamento, prevede quindi l'accoppiamento dei fiducial individuati e il calcolo della trasformazione che

minimizza la distanza tra i punti scelti.

Essendo un metodo semiautomatico, il tempo necessario per conseguire un risultato si riduce sensibilmente, rispetto a quello richiesto nel caso di una registrazione manuale.

Anche per questa tecnica però uno dei limiti è l'accuratezza della sovrapposizione ottenuta, poichè la precisione nel posizionamento dei landmarks dipende dall'abilità dell'utente, è infatti difficile caratterizzare matematicamente questi punti che variano da paziente a paziente.

Registrazione basata sulla superficie

La registrazione basata sulle superfici anatomiche comporta la definizione delle stesse a partire dalle sequenze di immagini e la determinazione delle trasformazioni che minimizzano la distanza tra di esse. A differenza dell'allineamento mediante marcatori, i quali vengono identificati manualmente, questa tecnica richiede la ricostruzione del volume da un insieme di contorni segmentati nelle varie immagini. La riuscita di questo metodo dipende essenzialmente dalla fase di segmentazione e dal range di sogliatura scelto. In campo medico il range di sogliatura definisce lo spessore delle ossa.

3.1.2 Classificazione basata sul tipo di Trasformazione

Le sezioni dei referti confrontati presentano un dis-allineamento che può essere eliminato mediante tecniche di trasformazione rigida (detta anche lineare) o non-rigida (indicata anche con il termine elastica)[13].

Trasformazioni rigide

Le trasformazioni lineari sono globali poichè vengono applicate all'intera immagine e possono essere suddivise in:

Modelli rigidi: i quali permettono solo la traslazione e la rotazione, sono caratterizzati quindi da sei gradi di libertà, tre di rotazione e tre di traslazione. Possono essere definiti nel seguente modo:

$$\varphi(x) = Qx + b$$

con

$Q \in R^{d \times d}$ ortogonale con $\det|Q| = 1$

$b \in R^d$

Modelli affini: i quali consentono la traslazione, la rotazione,

la scalatura. Una possibile definizione delle trasformazioni affini è la seguente:

$$\varphi(x) = Ax + b$$

con

$A \in R^{d \times d}$ ortogonale con $\det|A| > 0$

$b \in R^d$

Trasformazioni elastiche

Le trasformazioni non rigide, su cui si basa il metodo di registrazione elastico, sono caratterizzate da un elevato numero di gradi di libertà. La loro applicazione risulta però necessaria nel caso in cui vengano riscontrate, tra le immagini o i volumi confrontati, deformazioni spaziali elevate che non possono essere eliminate mediante l'adozione di modelli rigidi. Queste trasformazioni generano quindi deformazioni locali del Template al fine di allinearle con il riferimento.

Tra i metodi di registrazioni appena descritti, quello basato sull'applicazione di trasformazioni rigide è il più semplice da attuare, questo comporta che le trasformazioni geometriche più usate siano:

- la scalatura,
- la rotazione,
- la traslazione.

Scalatura

Preso una generica figura geometrica e scelto un punto \mathbf{B} appartenente ad essa, effettuare una scalatura significa riposizionare i restanti punti rispetto a \mathbf{B} secondo i fattori di scala individuati, è possibile scrivere quindi in forma matriciale:

$$\{X_1\} = [S] \cdot \{X_0\}$$

dove \vec{X}_0 e \vec{X}_1 sono due vettori colonna indicanti rispettivamente la posizione iniziale e finale del punto in questione, mentre $[S]$ è la matrice di scalatura. Considerando uno spazio 3D la relazione precedente può essere riscritta nel seguente modo:

$$\begin{Bmatrix} x_1 \\ y_1 \\ z_1 \end{Bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix} \cdot \begin{Bmatrix} x_0 \\ y_0 \\ z_0 \end{Bmatrix}$$

Se s_x, s_y ed s_z presentano valori diversi tra loro la scalatura è detta non uniforme, in caso contrario viene definita uniforme.

Rotazione

La rotazione risulta essere funzione solo dell'angolo generico θ e, in campo bidimensionale, permette la trasformazione del vettore di componenti (x_0, y_0) nel vettore (x_1, y_1) .

Di seguito è riportato il procedimento per la determinazione delle relazioni che correlano la posizione finale (indicata con il pedice 1) a quella iniziale (contrassegnata con il pedice 0) di un generico punto P, nel caso di una rotazione intorno all'asse Z.

Nella condizione iniziale, la posizione del punto P può essere espressa mediante le coordinate polari ρ e α :

$$\begin{aligned}x_0 &= \rho \cdot \cos(\alpha) \\y_0 &= \rho \cdot \sin(\alpha)\end{aligned}$$

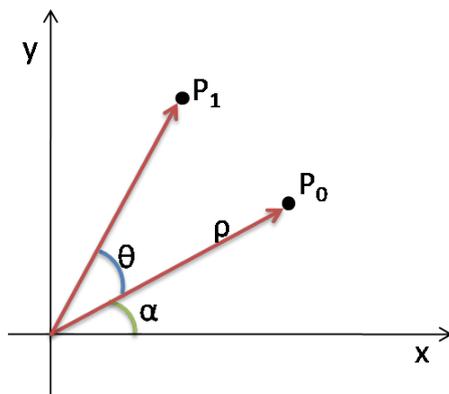


Figura 3.2: Rappresentazione grafica della rotazione, pari all'angolo θ , intorno all'asse z di un generico punto P.

Dopo una rotazione, in senso antiorario, uguale all'angolo θ , P si troverà in una nuova posizione definita dalle coordinate polari $(\alpha + \theta)$ e ρ . Apportando le dovute sostituzioni, il punto presenterà quindi le seguenti coordinate cartesiane:

$$\begin{aligned}x_1 &= x_0 \cdot \cos \theta - y_0 \cdot \sin \theta \\y_1 &= x_0 \cdot \sin \theta + y_0 \cdot \cos \theta\end{aligned}$$

Traslando il risultato ottenuto in campo tridimensionale è possibile scrivere la relazione appena vista come prodotto matrice per vettore:

$$\{X_1\} = [R_z] \cdot \{X_0\}$$

$$\begin{Bmatrix} x_1 \\ y_1 \\ z_1 \end{Bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{Bmatrix} x_0 \\ y_0 \\ z_0 \end{Bmatrix}$$

In egual modo possono essere definite le rotazioni rispettivamente intorno all'asse Y e all'asse X:

$$\{X_1\} = [R_y] \cdot \{X_0\}$$

$$\begin{Bmatrix} x_1 \\ y_1 \\ z_1 \end{Bmatrix} = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \cdot \begin{Bmatrix} x_0 \\ y_0 \\ z_0 \end{Bmatrix}$$

$$\{X_1\} = [R_x] \cdot \{X_0\}$$

$$\begin{Bmatrix} x_1 \\ y_1 \\ z_1 \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \sin \varphi & \cos \varphi \end{bmatrix} \cdot \begin{Bmatrix} x_0 \\ y_0 \\ z_0 \end{Bmatrix}$$

Nella realtà la posizione di un punto può essere espressa come risultato della combinazione di più rotazioni, l'ordine di applicazione delle matrici di rotazione è importante poiché esse non godono della proprietà commutativa.

Traslazione

La traslazione generica di un punto P in una nuova posizione P_1 può essere espressa nel modo seguente:

$$x_1 = x_0 + dx$$

$$y_1 = y_0 + dy$$

$$z_1 = z_0 + dz$$

Visto il largo utilizzo delle coordinate omogenee nel campo della computer grafica, è utile riscrivere tutte le relazioni viste in precedenza mediante queste coordinate.

Dato un punto P, nello spazio, di coordinate cartesiane (x, y, z) , si definiscono coordinate omogenee una qualsiasi quaterna di numeri x_1, x_2, x_3, x_4 tale per cui $\frac{x_1}{x_4} = x$, $\frac{x_2}{x_4} = y$ e $\frac{x_3}{x_4} = z$ con $x_4 \neq 0$.

La posizione di un generico punto sarà quindi definita dal vettore $\{x, y, z, 1\}$. Anche per questa trasformazione, come per la rotazione e la scalatura, è possibile ricavare la posizione finale assunta dal generico punto P mediante il prodotto tra la matrice traslazione, in questo caso di dimensioni (4x4), e il

vettore colonna, contenete le coordinate iniziali del punto:

$$\{X_1\} = [T] \cdot \{X_0\}$$

$$\begin{pmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{pmatrix}$$

L'utilizzo delle coordinate omogenee ha comportato, quindi, la modifica delle matrici di rotazione e di scalatura già presentate:

Matrice di scalatura

$$\begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Matrice di rotazione intorno all'asse Z

$$\begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Matrice di rotazione intorno all'asse Y

$$\begin{bmatrix} \cos \beta & 0 & \sin \beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \beta & 0 & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Matrice di rotazione intorno all'asse X

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi & 0 \\ 0 & \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Normalmente, il risultato della registrazione deriva dall'applicazione di una trasformazione composta, indicata con $[T_c]$, e dovuta alla combinazione delle trasformazioni elementari. Come visto in precedenza per la rotazione, è importante l'ordine di applicazione delle trasformazioni poichè non godono della proprietà commutativa:

$$[T_c] = [R_z] \cdot [R_y] \cdot [R_x] \cdot [S] \cdot [T]$$

$[T_c]$ è dunque una matrice di dimensione (4x4) di componenti $a_{i,j}$ e presenta la seguente forma:

$$[T_c] = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \end{bmatrix}$$

3.2 Confronto di metodi di registrazione di due sequenze di immagini mediche digitali

Prima di effettuare la sovrapposizione delle immagini bidimensionali dei due referti, è stata effettuata una stima della variazione dell'inclinazione del capo assunta dal paziente durante le due TAC. E' logico infatti immaginare che l'orientazione della testa, durante i due esami, non sia la stessa e la loro differenza viene espressa sotto forma di angolo generato tra due piani scelti. Per la determinazione di questa grandezza è stato sviluppato un programma in linguaggio python, il cui risultato è stato confrontato con il valore ottenuto mediante l'uso del software *SolidWorks*. La finalità di questo studio è, quindi, quella di verificare se esiste una relazione tra l'angolo individuato e i metodi di registrazione delle immagini.

Per definizione, l'angolo tra due piani è pari all'angolo compreso tra i due vettori perpendicolari ad essi; questa grandezza è dovuta alla combinazione di tre rotazioni del capo intorno ai tre assi del sistema di riferimento. Come è possibile osservare nella Figura 3.3 queste rotazioni vengono indicate con:

- Pitch o beccheggio: rotazione intorno all'asse LR (asse X del sistema di riferimento)
- Yaw o imbardata: rotazione intorno all'asse IS (asse Z del sistema di riferimento)
- Roll o rollio: rotazione intorno all'asse PA (asse Y del sistema di riferimento)

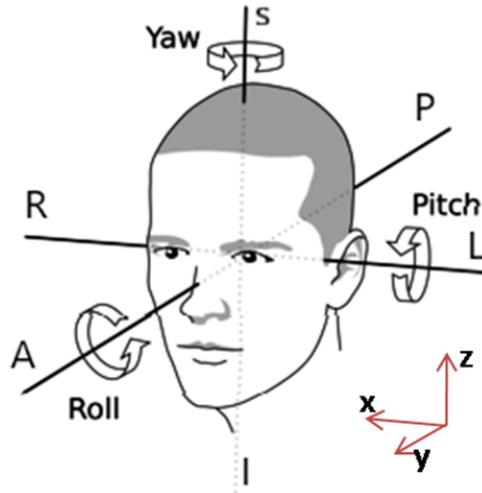


Figura 3.3: Rappresentazione delle tre rotazioni principali effettuabili dal capo[15].

Le formule inserite nelle pagine seguenti richiamano il procedimento analitico seguito per il calcolo dell'angolo e delle rotazioni ad esso associate. Dal punto di vista geometrico, per definire un piano è sufficiente conoscere le coordinate di tre punti appartenenti ad esso; posizionati tre fiduciali sul modello 3D, $A(z_1, y_1, x_1)$, $B(z_2, y_2, x_2)$ e $C(z_3, y_3, x_3)$, vengono individuati i termini noti (a,b,c) dell'equazione del piano, calcolando il determinante della matrice $[M]$:

$$[M] = \begin{bmatrix} x - x_1 & y - y_1 & z - z_1 \\ x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 \end{bmatrix}$$

Sono quindi definite le componenti del vettore \vec{v} normale ad un generico piano:

$$\begin{aligned} a &= (x_2 - x_1) \cdot (y_3 - y_1) - (y_2 - y_1) \cdot (x_3 - x_1) \\ b &= -(x_2 - x_1) \cdot (z_3 - z_1) + (z_2 - z_1) \cdot (x_3 - x_1) \\ c &= (y_2 - y_1) \cdot (z_3 - z_1) - (z_2 - z_1) \cdot (y_3 - y_1) \end{aligned}$$

Noti i vettori normali \vec{v}_1 e \vec{v}_2 viene calcolato l'angolo presente tra di essi nel seguente modo:

$$\theta = \arccos \left(\frac{\vec{v}_1 \cdot \vec{v}_2}{\|\vec{v}_1\| \cdot \|\vec{v}_2\|} \right)$$

Per effettuare questo studio sono state prese in considerazione le TAC appartenenti al paziente A. Il primo passo dell'analisi consiste nel definire un piano che sia di facile individuazione su entrambi i referti e la cui rotazione sia solidale a quella del cranio, la scelta è quindi ricaduta sul Piano di Francoforte.

Il *Piano di Francoforte*:" detto anche auricolo-orbitario è un piano ideale che passa per il punto sottoorbitale e la parte superiore del condotto uditivo"[14]. Di seguito, è possibile osservare un'immagine esemplificativa della posizione dei punti che lo definiscono.

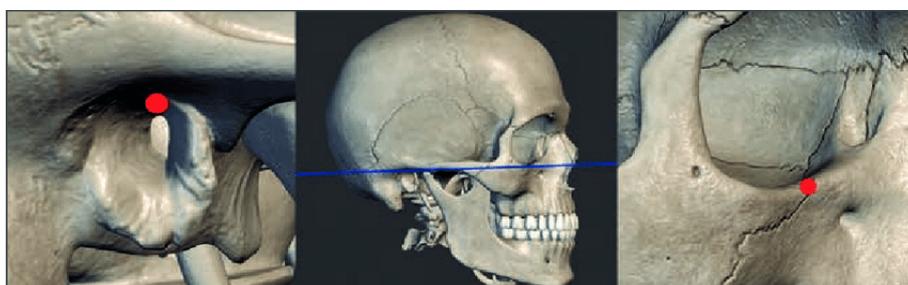


Figura 3.4: Immagine descrittiva del metodo di individuazione del Piano di Francoforte, fonte: *www.researchgate.net*.

Le coordinate dei marcatori individuati, che identificano i due piani auricolo-orbitari nei referti analizzati, sono riportate nelle Tabelle 3.1 e 3.2 mentre nelle Figure 3.5 e 3.6 è possibile osservare la loro collocazione sulle ricostruzioni tridimensionali:

Referto pre-operatorio

PUNTI	Z (mm)	Y (mm)	X (mm)
A1	-15.063	47.902	-30.230
A2	2.714	-28.592	-47.333
A3	4.522	-21.151	62.520

Tabella 3.1: Coordinate dei punti che definiscono il Piano di Francoforte nel referto pre-operatorio.

Referto post-operatorio

PUNTI	Z (mm)	Y (mm)	X (mm)
B1	-18.243	34.321	-42.679
B2	8.866	-42.219	-57.265
B3	-1.391	-37.578	55.159

Tabella 3.2: Coordinate dei punti che definiscono il piano auricolo-orbitario nel referto post-operatorio.

Dalle immagini è possibile notare che i punti utilizzati per creare i piani sono il punto sottoorbitale dell'occhio sinistro e i due condotti uditivi. Nel caso analizzato, non è stato possibile prendere in considerazione l'occhio destro poiché l'intervento effettuato ha portato alla modifica della sua orbita. Per distinguere più facilmente le due TAC del paziente A è stato assegnato ad ognuna di esse un colore differente: rosso per quella eseguita prima dell'intervento, verde per il referto post operatorio.

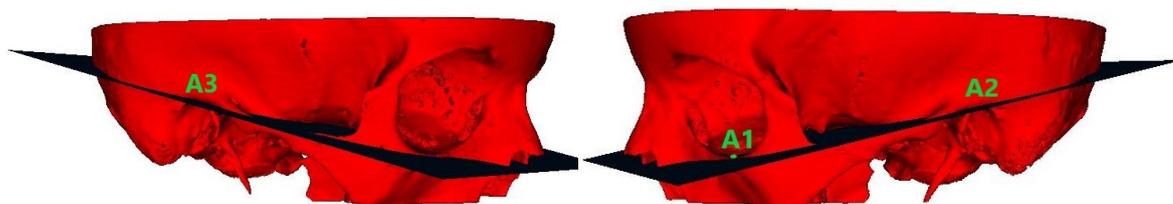


Figura 3.5: Ricostruzione della conformazione del cranio prima dell'operazione con indicati i punti utilizzati per creare il Piano di Francoforte e il piano stesso.

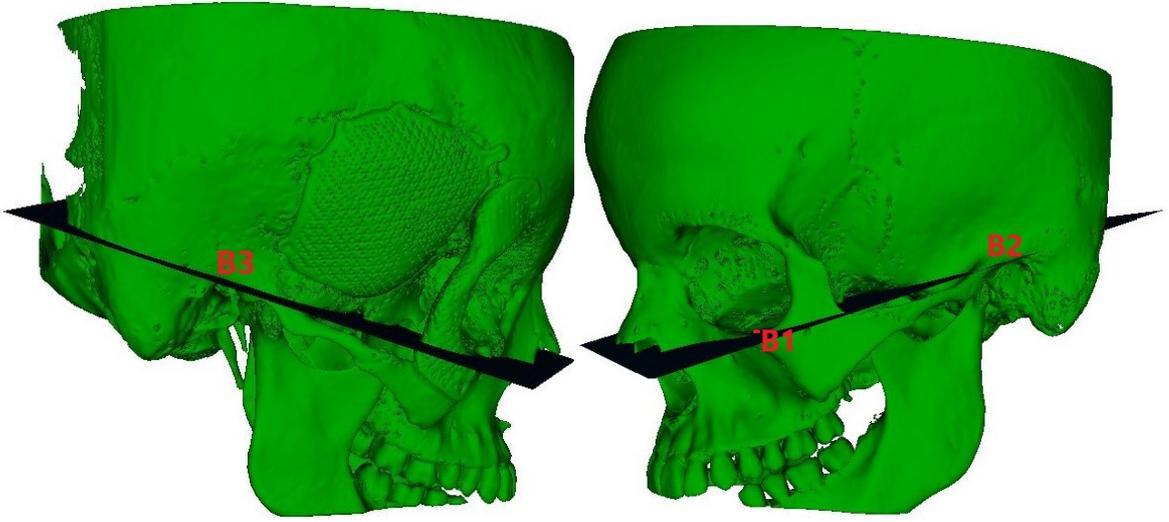


Figura 3.6: Ricostruzione della conformazione del cranio dopo l'operazione con indicati i punti utilizzati per creare il piano auricolare-orbitario e il piano stesso.

Applicando le formule viste in precedenza, i vettori normali ai piani presentano le seguenti componenti espresse in (mm):

$$\vec{v}_1 = (8275.83, 1983.77, -270.58)$$

$$\vec{v}_2 = (8537.24, 2898.09, 659.26)$$

quindi note le loro norme è possibile determinare l'angolo θ tra i due Piani di Francoforte:

$$\|\vec{v}_1\| = 8514.57 \text{ mm}$$

$$\|\vec{v}_2\| = 9039.80 \text{ mm}$$

$$\theta = \text{acos} \left(\frac{\vec{v}_1 \cdot \vec{v}_2}{\|\vec{v}_1\| \cdot \|\vec{v}_2\|} \right) = 7.99^\circ$$

La sola conoscenza dell'angolo calcolato non è sufficiente a comprendere a pieno la rotazione compiuta dal capo tra le due TAC, è quindi necessario determinare anche l'asse intorno al quale ha ruotato il piano auricolare-orbitario; esso coincide con la retta intersezione dei due piani, definita dal versore \vec{n} perpendicolare ai vettori \vec{v}_1 e \vec{v}_2 individuati.

Anche in questo caso, le componenti del versore possono essere calcolate mediante il determinante della matrice $[N]$, che ha per righe le componenti dei

vettori \vec{v}_1 e \vec{v}_2 :

$$[N] = \begin{bmatrix} i & j & k \\ v_{1x} & v_{1y} & v_{1z} \\ v_{2x} & v_{2y} & v_{2z} \end{bmatrix}$$

ottenendo così il seguente risultato:

$$\vec{n} = (0.66, -0.73, 0.19)$$

L'angolo tra due piani, come già asserito, può essere immaginato come il risultato della combinazione delle rotazioni intorno agli assi del sistema di riferimento (X, Y, Z) , il quale è stato indicato nella Figura 3.3. Le entità di queste rotazioni sono definite dagli angoli compresi tra le proiezioni dei vettori normali \vec{v}_1 e \vec{v}_2 sui tre piani del sistema di riferimento. A titolo di esempio viene riportato solo il procedimento seguito per il piano XY, mentre i valori ottenuti per le tre rotazioni (Pitch, Yaw e Roll) sono stati inseriti nella Tabella 3.3.

Nel piano XY i vettori proiezione di \vec{v}_1 e \vec{v}_2 avranno solo componenti y e x:

$$\vec{v}_{1,xy} = (1983.77, -270.58)$$

$$\vec{v}_{2,xy} = (2898.09, 659.26)$$

e norme:

$$\|\vec{v}_{1,xy}\| = 2002.15$$

$$\|\vec{v}_{2,xy}\| = 2972.13$$

L'angolo calcolato corrisponde alla rotazione, definita Yaw, intorno all'asse Z:

$$Yaw = \arccos\left(\frac{\vec{v}_{1,xy} \cdot \vec{v}_{2,xy}}{\|\vec{v}_{1,xy}\| \cdot \|\vec{v}_{2,xy}\|}\right) = 20.58^\circ$$

ROTAZIONI	ANGOLI [°]	ASSE DI ROTAZIONE
Yaw	20.58	Z
Pitch	5.27	X
Roll	6.29	Y

Tabella 3.3: Valori calcolati per le rotazioni intorno agli assi del sistema di riferimento.

Il valore dell'angolo tra i due piani può essere anche ricavato graficamente, mediante l'uso del software *SolidWorks*. Una volta importate le ricostruzioni tridimensionali dei due crani, in formato STL, sono stati posizionati i punti

che definiscono i due piani e, mediante l'uso dello strumento misura, è stato visualizzato l'angolo tra di essi. Come è possibile notare dalla Figura 3.7 il valore ottenuto è 8.07° ed è prossimo a quello calcolato ($7,99^\circ$); l'errore presente è dovuto all'incertezza nel posizionamento dei punti, dal momento che ciò è stato effettuato manualmente.

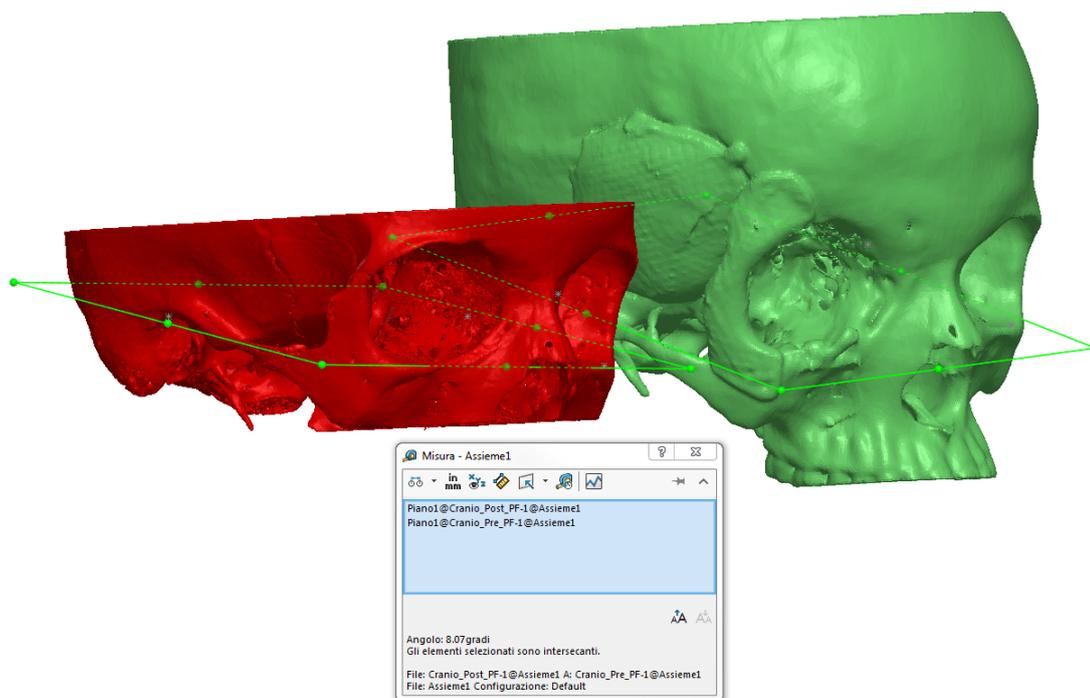


Figura 3.7: Misura grafica dell'angolo generato tra i piani auricolo-orbitari.

Per comprendere la relazione tra l'angolo calcolato e i diversi metodi di allineamento, sono stati testati alcuni moduli a disposizione nel programma *3D Slicer*. Le registrazioni delle immagini dei referti sono state così ottenute mediante i metodi: manuale, automatico o semiautomatico; giungendo alla conclusione che, per il paziente A, l'allineamento semiautomatico, basato sui landmarks è il metodo che fornisce il risultato migliore. E' importante notare come, nonostante l'entità dell'inclinazione calcolata non sia trascurabile, sia comunque possibile effettuare le sovrapposizioni delle sezioni dei due esami, dal momento che i tagli nei due referti sono eseguiti a distanze di 0.3 mm .

Registrazione manuale:

Questa registrazione può essere effettuata mediante il modulo *Transforms* (fig.3.1), non è stata inserita un'immagine della sovrapposizione poiché, nonostante un elevato numero di tentativi, non è stato ottenuto un esito sod-

disfacente. La difficoltà di questo allineamento può essere imputata alla forte variazione dell'inclinazione del capo, la quale condiziona il valore dei parametri che caratterizzano la trasformazione. Inoltre è stato osservato, dal confronto degli allineamenti effettuati per due differenti pazienti, come l'influenza dell'ordine di applicazione delle trasformazioni sul risultato sia proporzionale all'angolo calcolato. I vari tentativi di sovrapposizione sono stati effettuati applicando la trasformazione rigida la quale, anche se risulta la più semplice da attuare, presenta sei gradi di libertà (tre di traslazione e tre di rotazione).

Allineamento automatico:

Il modulo utilizzato, chiamato *General Registration*, richiede di definire alcuni parametri tra cui: la trasformazione da applicare, la serie di immagini usata come riferimento e quella adoperata come test. Il programma restituisce in uscita la successione di immagini del test trasformata spazialmente.

Le scelte effettuate sono riportate nella Tabella 3.4.

Reference	TAC pre operatoria
Template	TAC post operatoria
Trasformazione	Rigida

Tabella 3.4: Tabella indicante le scelte attuate per la definizione del riferimento, del test e del tipo di trasformazione da applicare.

Questo metodo, non permettendo di escludere la zona alterata dall'intervento, la quale nel caso studiato interessa un'area molto estesa, ha fornito il risultato visibile nelle Figure 3.8, 3.9 e 3.10. Esse mostrano per ogni vista la sovrapposizione di due tagli corrispondenti (uno della serie di riferimento, in rosso, e uno del test trasformato spazialmente, in verde):

- finestra rossa : vista assiale
- finestra gialla : vista sagittale
- finestra verde : vista coronale

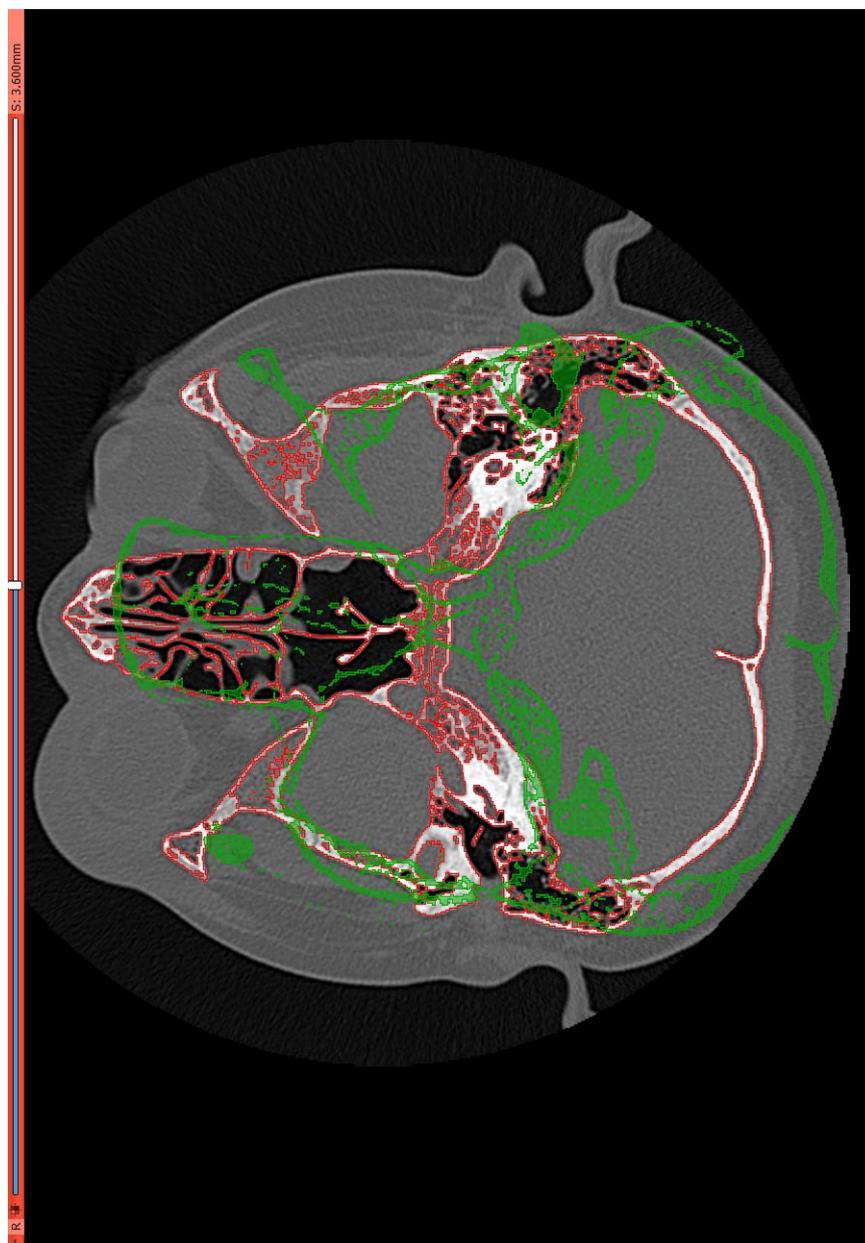


Figura 3.8: Sovrapposizione nella vista assiale di un taglio delle TAC del paziente A ottenuta mediante registrazione automatica.

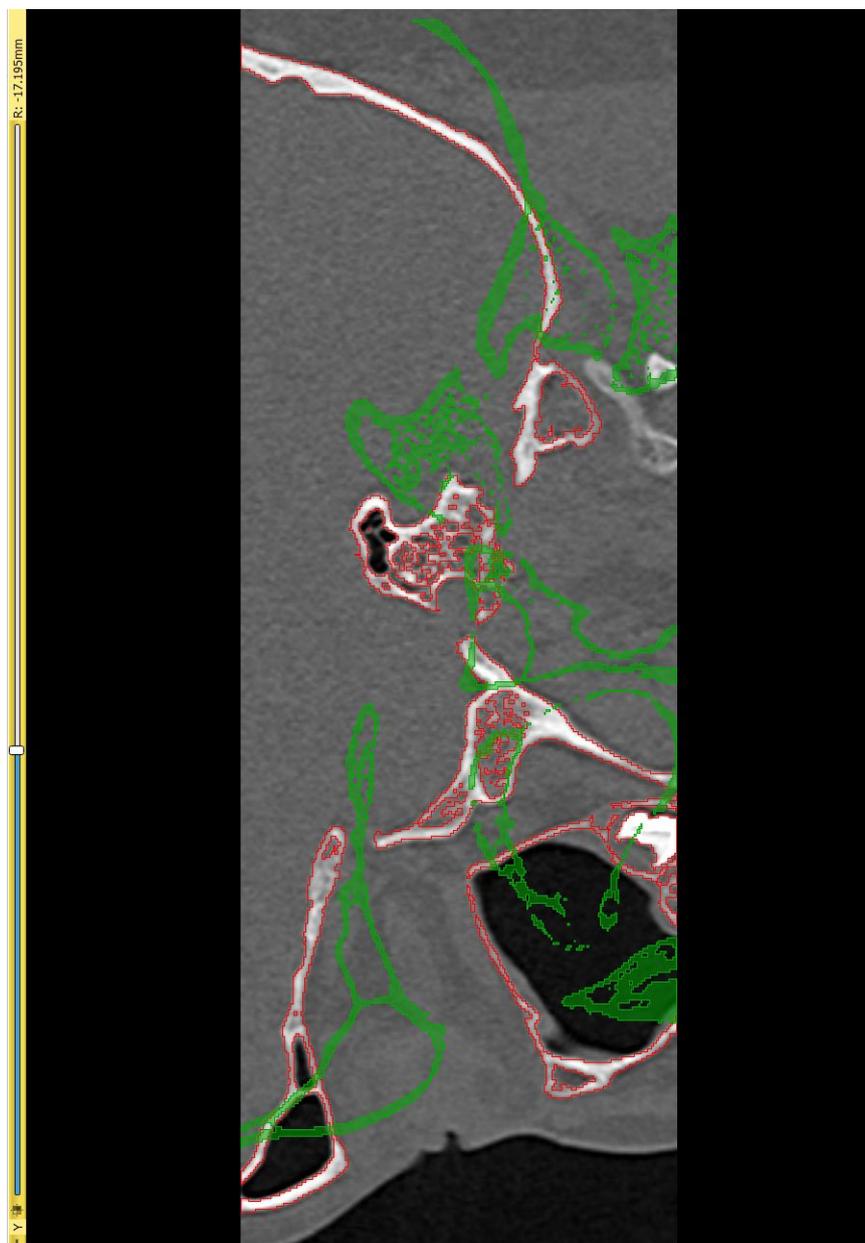


Figura 3.9: Sovrapposizione nella vista sagittale di un taglio delle TAC del paziente A ottenuta mediante registrazione automatica.

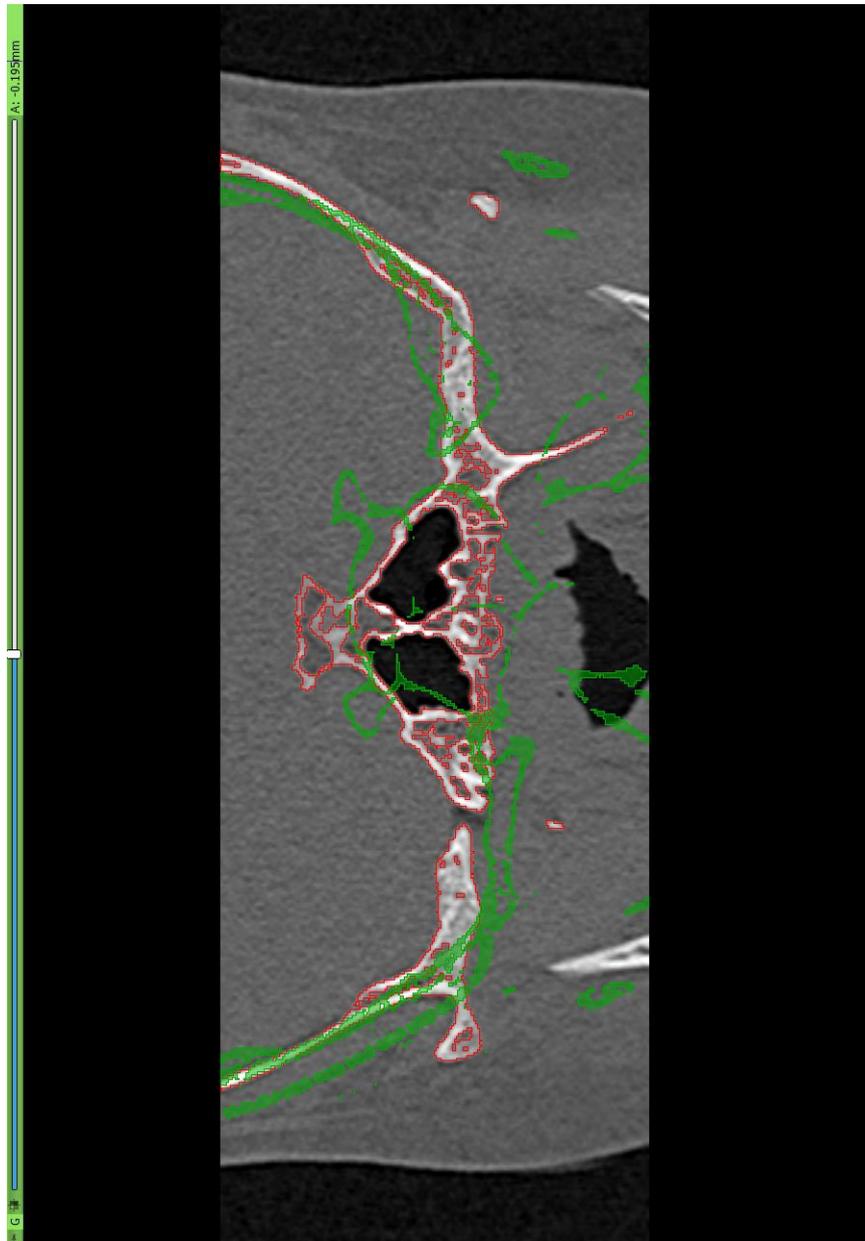


Figura 3.10: Sovrapposizione nella vista coronale di un taglio delle TAC del paziente A ottenuta mediante registrazione automatica.

La pessima sovrapposizione ottenuta, probabilmente dovuta alla concomitanza nell'analisi di un'inclinazione non trascurabile e della zona modificata dall'intervento, la cui estensione interessa la maggior parte del volume, è ancora più evidente se si osserva la sua ricostruzione tridimensionale.

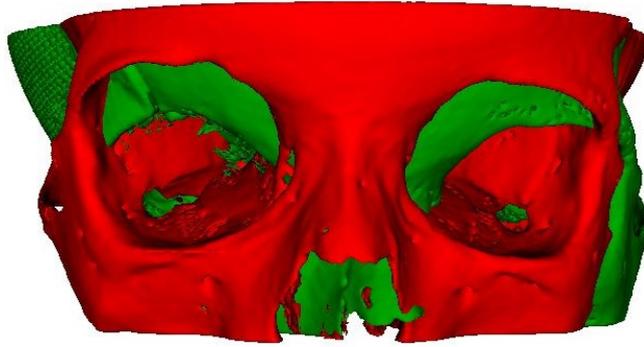


Figura 3.11: Vista frontale della ricostruzione 3d della sovrapposizione dei due crani ottenuta mediante metodo automatico.



Figura 3.12: Vista laterale della ricostruzione 3d della sovrapposizione dei due crani ottenuta mediante metodo automatico.

Registrazione basata sui landmarks

Questo tipo di allineamento è semiautomatico poichè presenta una componente manuale, corrispondente al posizionamento dei fiducial, e una componente automatica, legata alla determinazione della trasformazione spaziale del Template da parte di un programma. Come visto in precedenza la scelta dei marcatori permette di escludere dal calcolo la zona operata fornendo così un risultato migliore e visibile nelle Figure 3.17, 3.18 e 3.19.

Il programma utilizzato per ottenere la sovrapposizione è stato scritto durante lo studio eseguito a causa degli scarsi risultati ottenuti mediante i metodi gratuiti precedentemente testati. Il funzionamento del programma è descritto più nel dettaglio nella sezione 3.3, riguardante il confronto di software utilizzati per la registrazione di modelli tridimensionali.

Il modulo realizzato richiede l'individuazione su entrambi i referti di quattro punti che presentano posizioni corrispondenti. I fiducial definiti sulla TAC pre operatoria, usata come riferimento, sono indicati con la classe F, mentre

quelli situati sul referto post operatorio sono definiti con la lettera M. Nelle immagini seguenti sono mostrate le posizioni dei marcatori della classe F nelle viste assiale (finestra rossa), sagittale(riquadro giallo), coronale (finestra verde):

Fiducial F_1 :

F_1 e M_1 sono situati in corrispondenza della giunzione tra le ossa nasali e le due cartilagini nasali laterali.



Figura 3.13: Viste assiale e sagittale della posizione del fiducial F1 sulla TAC pre operatoria del paziente A.

Fiducial F_2

F_2 e M_2 sono posizionati in corrispondenza del foro zigomatico faciale sinistro.



Figura 3.14: Viste assiale e coronale della posizione del fiducial F2 sulla TAC pre operatoria del paziente A.

Fiducial F_3 :

F_3 e M_3 sono situati sulla sutura fronto-zigomatica sinistra.

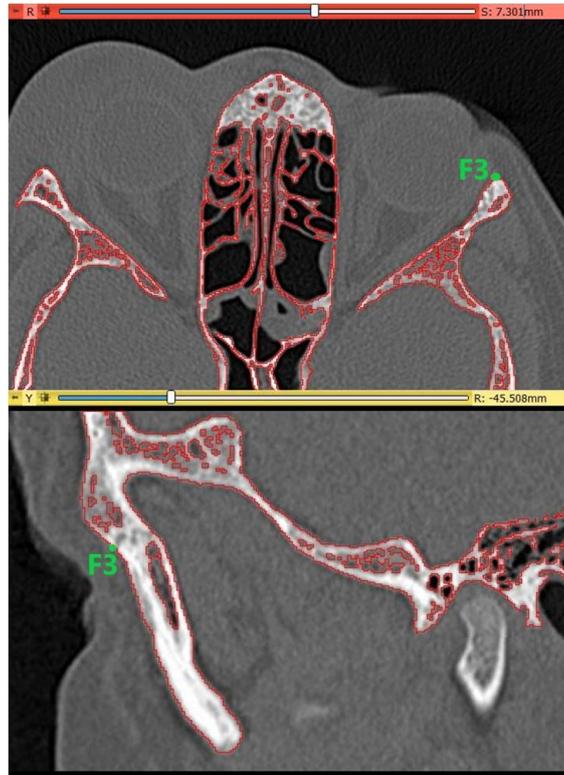


Figura 3.15: Viste assiale e sagittale della posizione del fiducial F3 sulla TAC pre operatoria del paziente A.

Fiducial F_4 :

F_4 e M_4 sono situati su una sporgenza dell'osso temporale destro.

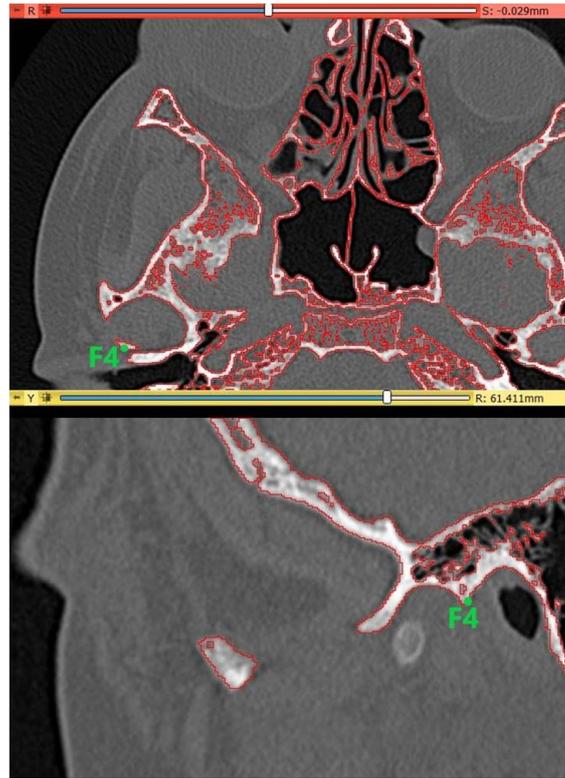


Figura 3.16: Viste assiale e sagittale della posizione del fiducial F_4 sulla TAC pre operatoria del paziente A.

La sovrapposizione ottenuta dall'identificazione di questi marcatori è visibile nelle Figure 3.17, 3.18 e 3.19. Dal momento che l'emisfero destro ha subito modifiche a causa dell'intervento il taglio sagittale riportato individua una sezione della parte sinistra del cranio. Se si confronta il risultato ottenuto con quello del metodo automatico è evidente come, in questo caso, i tagli sovrapposti siano i medesimi.

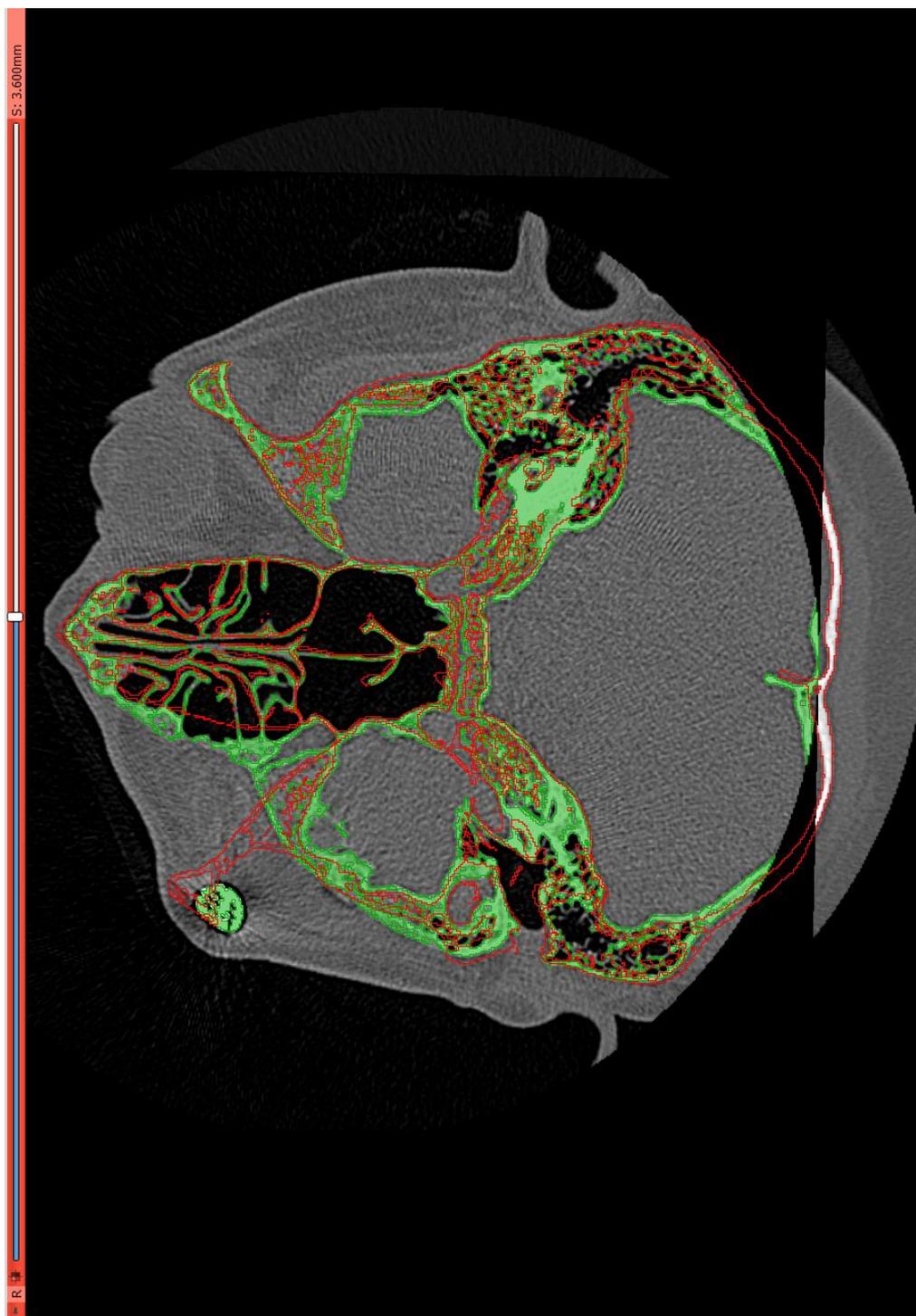


Figura 3.17: Sovrapposizione di un taglio, nella vista assiale, ottenuta mediante registrazione basata sull'individuazione di landmarks sulle TAC del paziente A.

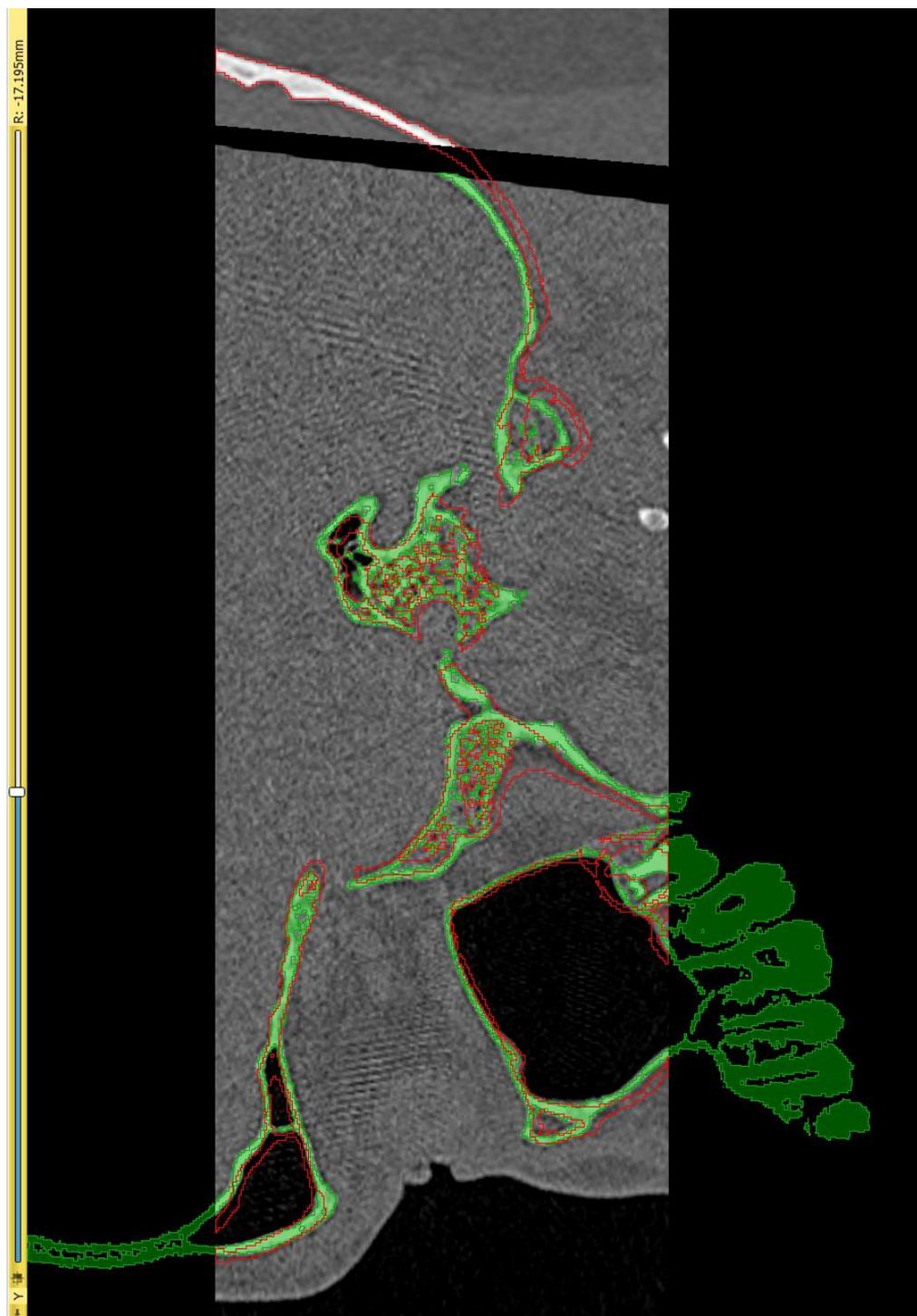


Figura 3.18: Sovrapposizione di un taglio, nella vista sagittale, ottenuta mediante registrazione basata sull'individuazione di landmarks sulle TAC del paziente A.

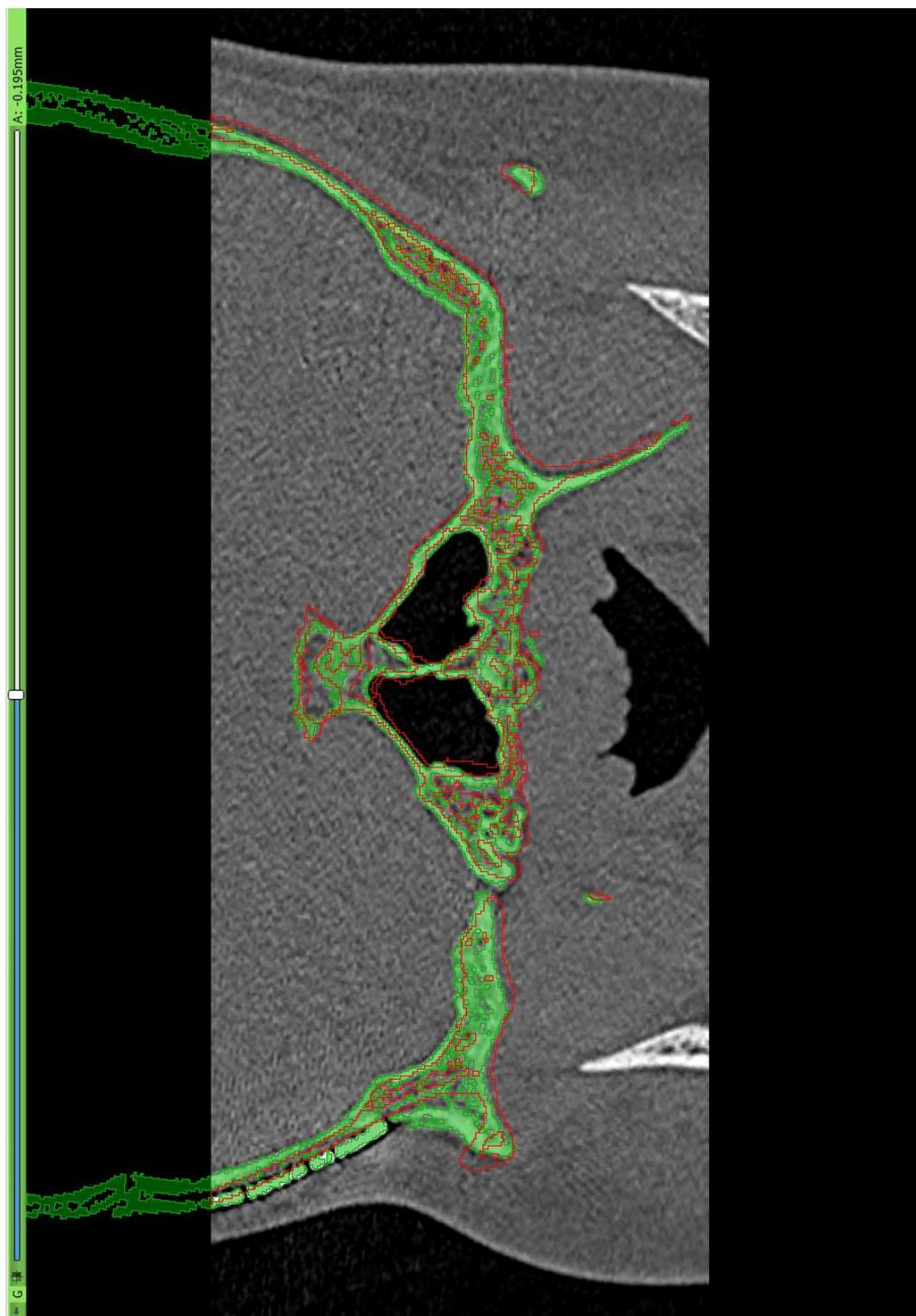


Figura 3.19: Sovrapposizione di un taglio, nella vista coronale, ottenuta mediante registrazione basata sull'individuazione di landmarks sulle TAC del paziente A.

Dal momento che il volume a disposizione del paziente A è limitato e parzialmente utilizzabile per effettuare la registrazione, sono stati testati gli stessi metodi sui referti di un secondo paziente, chiamato paziente B, e del quale si hanno a disposizione le TAC dell'intero cranio. Prima di effettuare il confronto tra le varie registrazioni, per osservare la differenza rispetto al caso precedente, è stato determinato, anche in questo caso, l'angolo tra i due Piani di Francoforte.

I punti individuati per definire i due piani nei referti analizzati sono riportati nelle Tabelle 3.5 e 3.6 e nelle Figure 3.20 e 3.21:

Referto pre-operatorio

PUNTI	Z (mm)	Y (mm)	X (mm)
A1	-14.65	55.98	25.32
A2	-10.73	58.16	-45.84
A3	-5.36	-14.04	-67.31

Tabella 3.5: Coordinate dei tre punti che definiscono il Piano di Francoforte nel referto pre-operatorio del paziente B.

Referto post-operatorio

PUNTI	Z (mm)	Y (mm)	X (mm)
B1	-7.36	63.48	31.91
B2	-2.745	67.82	-39.32
B3	7.99	-5.36	-60.88

Tabella 3.6: Coordinate dei punti che definiscono il Piano di Francoforte nel referto post-operatorio del paziente B.

Per il paziente B la convenzione di colori utilizzata è rosso per la ricostruzione del referto pre operatorio e azzurro per quella della TAC effettuata dopo l'operazione.

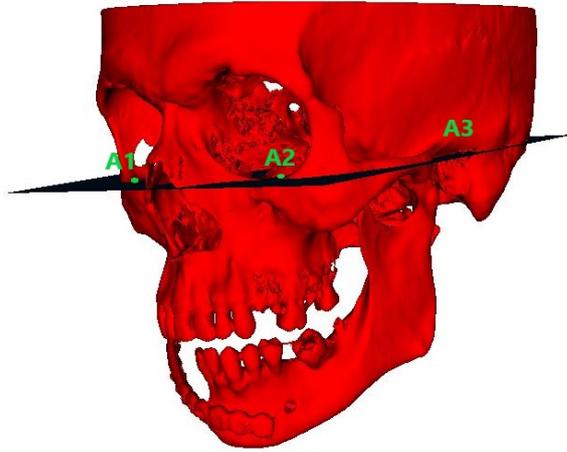


Figura 3.20: Ricostruzione del cranio del paziente B prima dell'operazione con indicati i punti utilizzati per creare il Piano di Francoforte e il piano stesso.

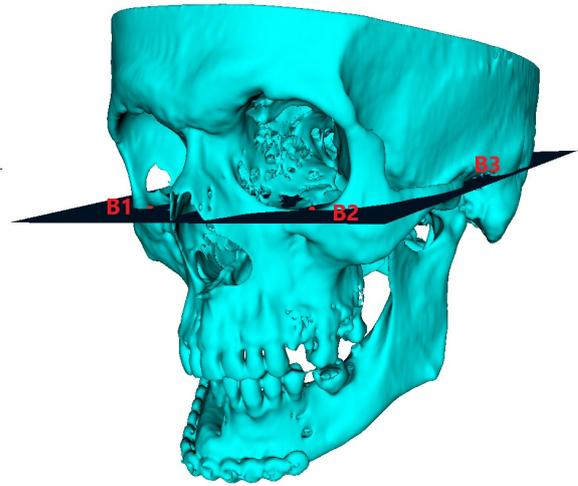


Figura 3.21: Ricostruzione del cranio del paziente B dopo l'operazione con indicati i punti utilizzati per creare il Piano di Francoforte e il piano stesso.

I valori ottenuti dall'analisi eseguita sono:

Angolo tra i Piani di Francoforte: 3.95°

Vettore dell'asse di rotazione: $\vec{n} = (-0.97, 0.22, 0.04)$

ROTAZIONI	ANGOLI [°]	ASSE DI ROTAZIONE
Yaw	14.65	Z
Pitch	3.86	X
Roll	0.89	Y

Tabella 3.7: Valori calcolati per le rotazioni intorno agli assi del sistema di riferimento.

Mediante *SolidWorks* è stato invece rilevato un angolo di 4.15° , come è già stato osservato per il paziente A, la differenza di valore va imputata al posizionamento manuale dei marcatori.

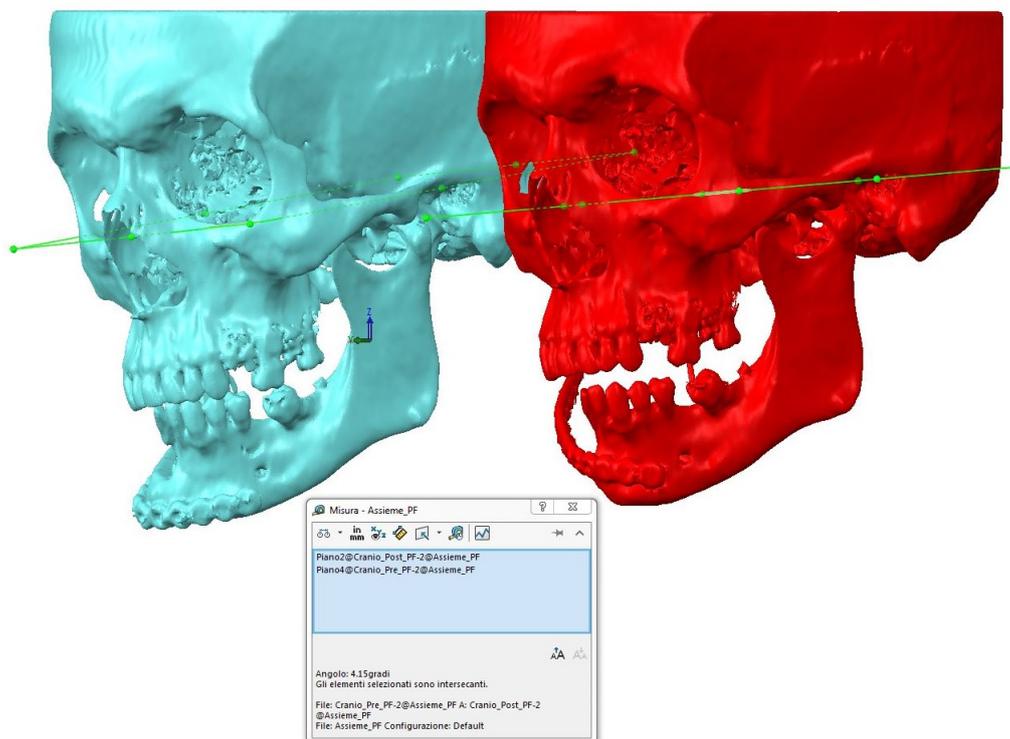


Figura 3.22: Misura grafica dell'angolo generato tra i piani auricolo-orbitari.

Una volta definito l'angolo, sono state effettuate nuovamente le sovrapposizioni manuali, automatiche e semiautomatiche.

Registrazione manuale:

Contrariamente a quanto visto nel caso precedente, per il paziente B è stato possibile ottenere un risultato soddisfacente anche dalla registrazione manuale. E' stata osservata infatti una limitata influenza sul risultato da parte delle rotazioni, questo può essere attribuito alla variazione contenuta dell'inclinazione del capo.

Il tempo impiegato per effettuare la sovrapposizione manuale delle due TAC, riportata nelle Figure 3.23, 3.24 e 3.25, è stata all'incirca di 10 minuti.

Nell'analisi deve essere esclusa la mandibola, sia perché è la parte di volume interessata dall'intervento sia perché, essendo collegata al resto del cranio mediante l'articolazione temporo-mandibolare, la sua posizione risulta essere indipendente da quella delle altre ossa del cranio.

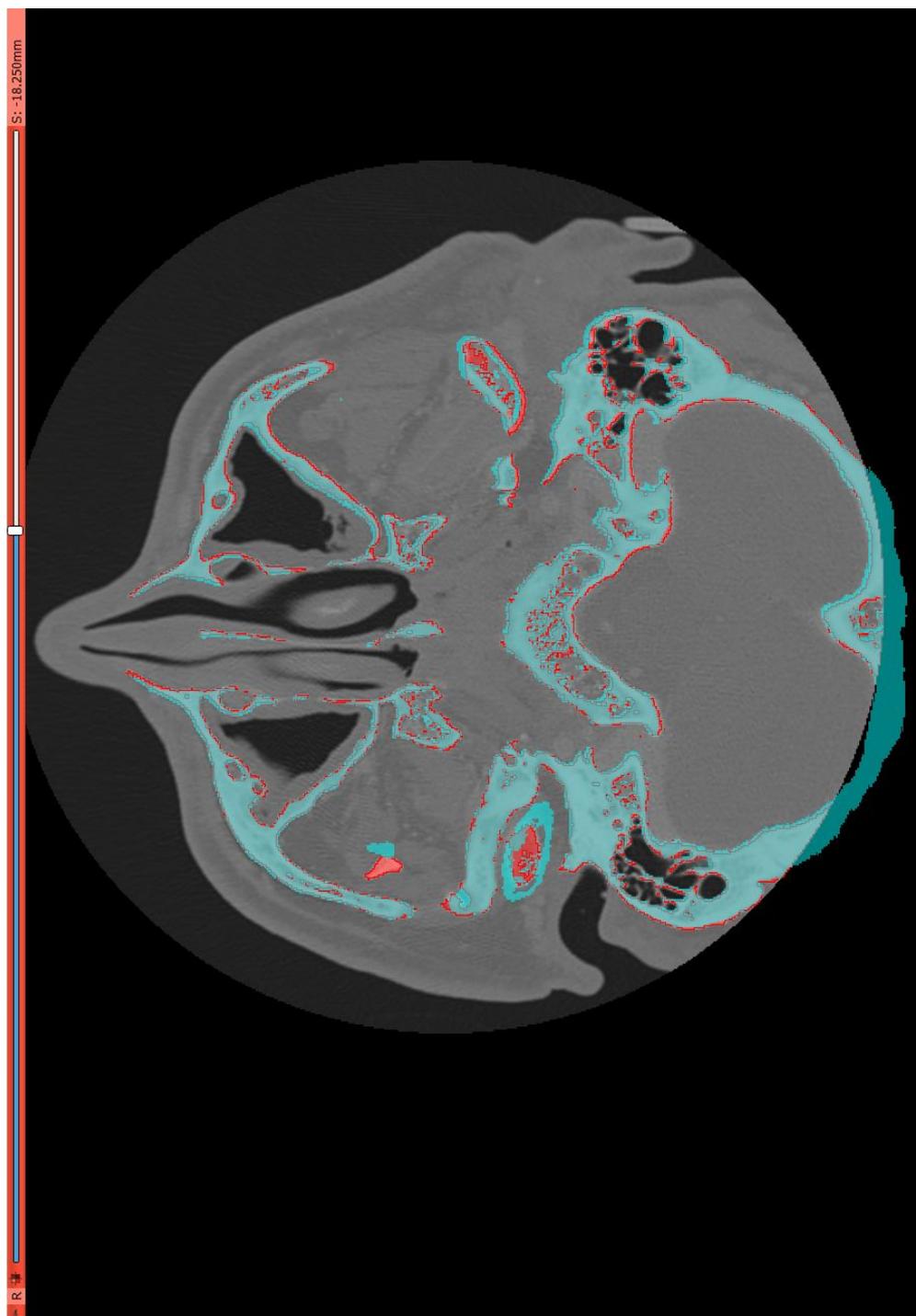


Figura 3.23: Immagine della sovrapposizione manuale di due tagli corrispondenti, dei referti del paziente B, nella vista assiale.

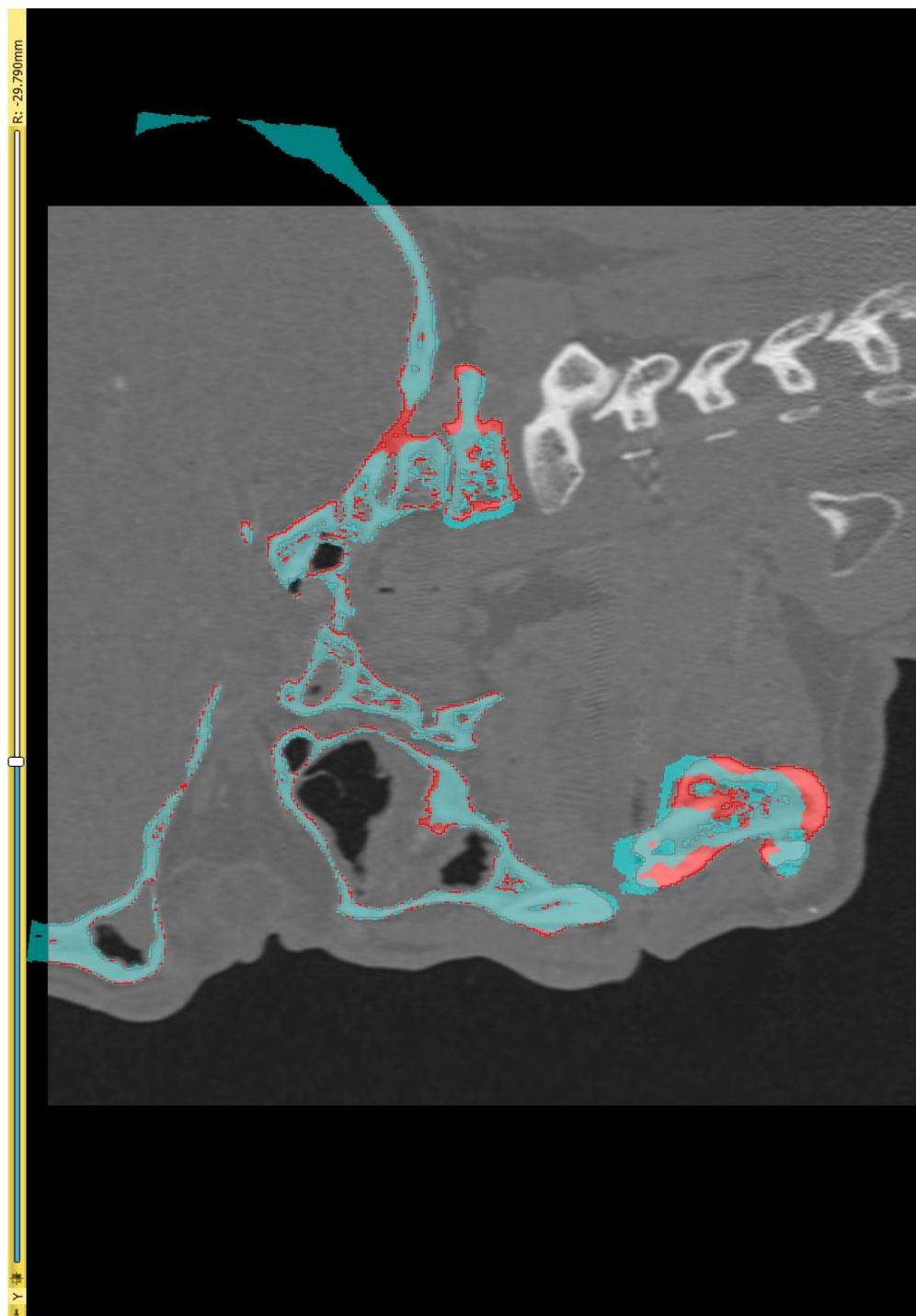


Figura 3.24: Immagine della sovrapposizione manuale di due tagli corrispondenti, dei referti del paziente B, nella vista sagittale.

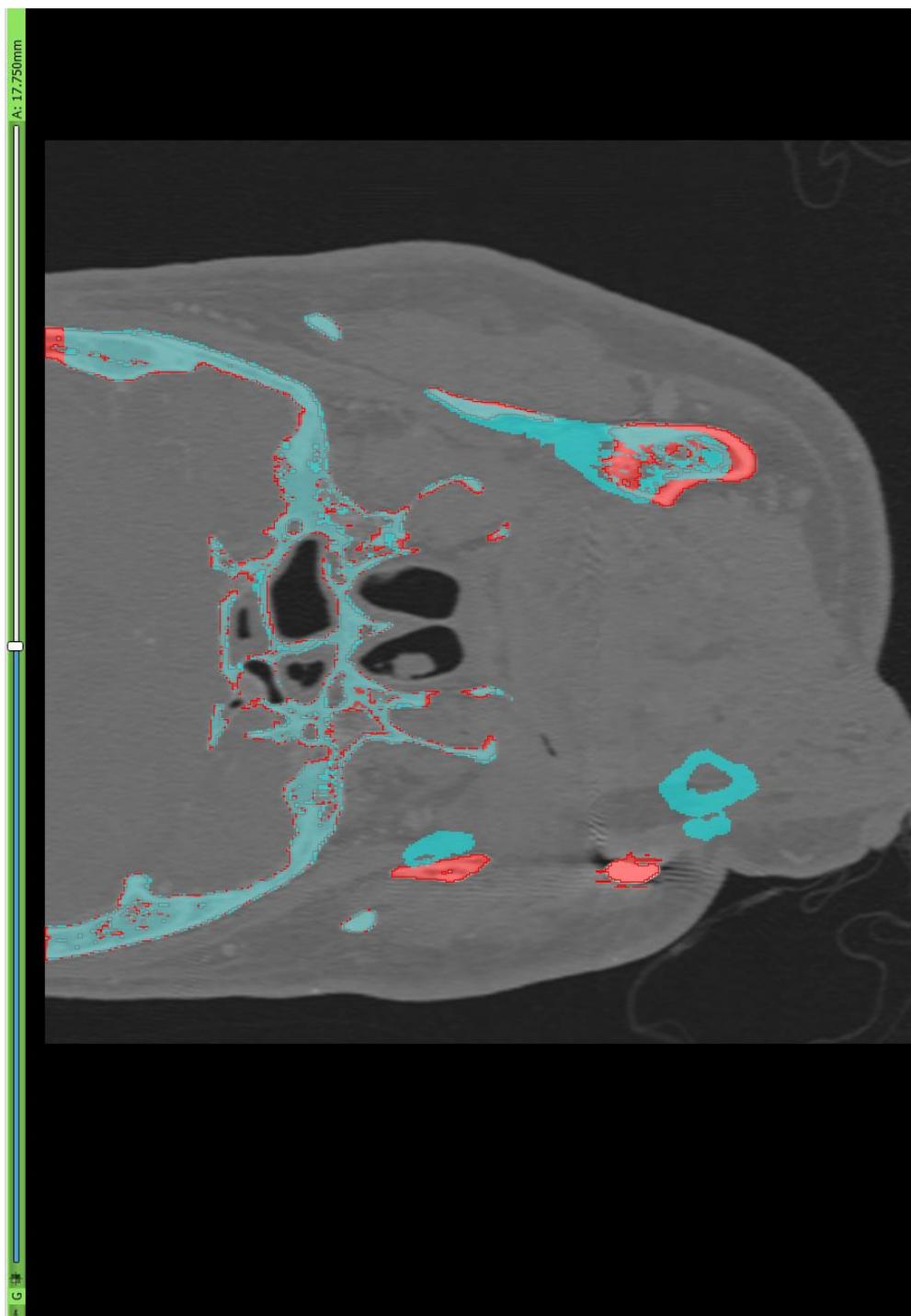


Figura 3.25: Immagine della sovrapposizione manuale di due tagli corrispondenti, dei referti del paziente B, nella vista coronale.

Allineamento automatico:

Le scelte effettuate (Tab.3.8) per l'esecuzione dell'allineamento automatico delle sezioni dei referti del paziente B sono le medesime del paziente A.

Reference	TAC pre operatoria
Template	TAC post operatoria
trasformazione	Rigida

Tabella 3.8: Tabella riportante le scelte attuate per eseguire la registrazione automatica dei referti del paziente B.

Una semplice osservazione dei referti ha mostrato come i due crani fossero solo ruotati e traslati tra di loro, per questo motivo si è deciso di adottare, per la registrazione, una trasformazione rigida. La scelta di una trasformazione elastica avrebbe richiesto infatti un tempo di calcolo maggiore e avrebbe prodotto un risultato non corretto, deformando localmente l'immagine. La sovrapposizione ottenuta è riportato nelle Figure 3.26, 3.27 e 3.28, ad essa è associata la matrice di trasformazione lineare $[T_c]$:

$$[T_c] = \begin{bmatrix} 1 & 0 & -0.02 & -5.46 \\ 0 & 1 & -0.07 & -8.30 \\ 0.02 & 0.07 & 1 & -11.90 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

E' quindi visibile dalle immagini riportate come l'allineamento ottenuto dalle scelte effettuate sia perfetto; nell'analisi va sempre esclusa la mandibola per gli stessi motivi visti nel metodo manuale. La differenza di comportamento del modulo usato rispetto al paziente A può essere dovuta sia al volume esiguo utilizzabile sia al valore non piccolo dell'angolo calcolato nel primo caso analizzato.

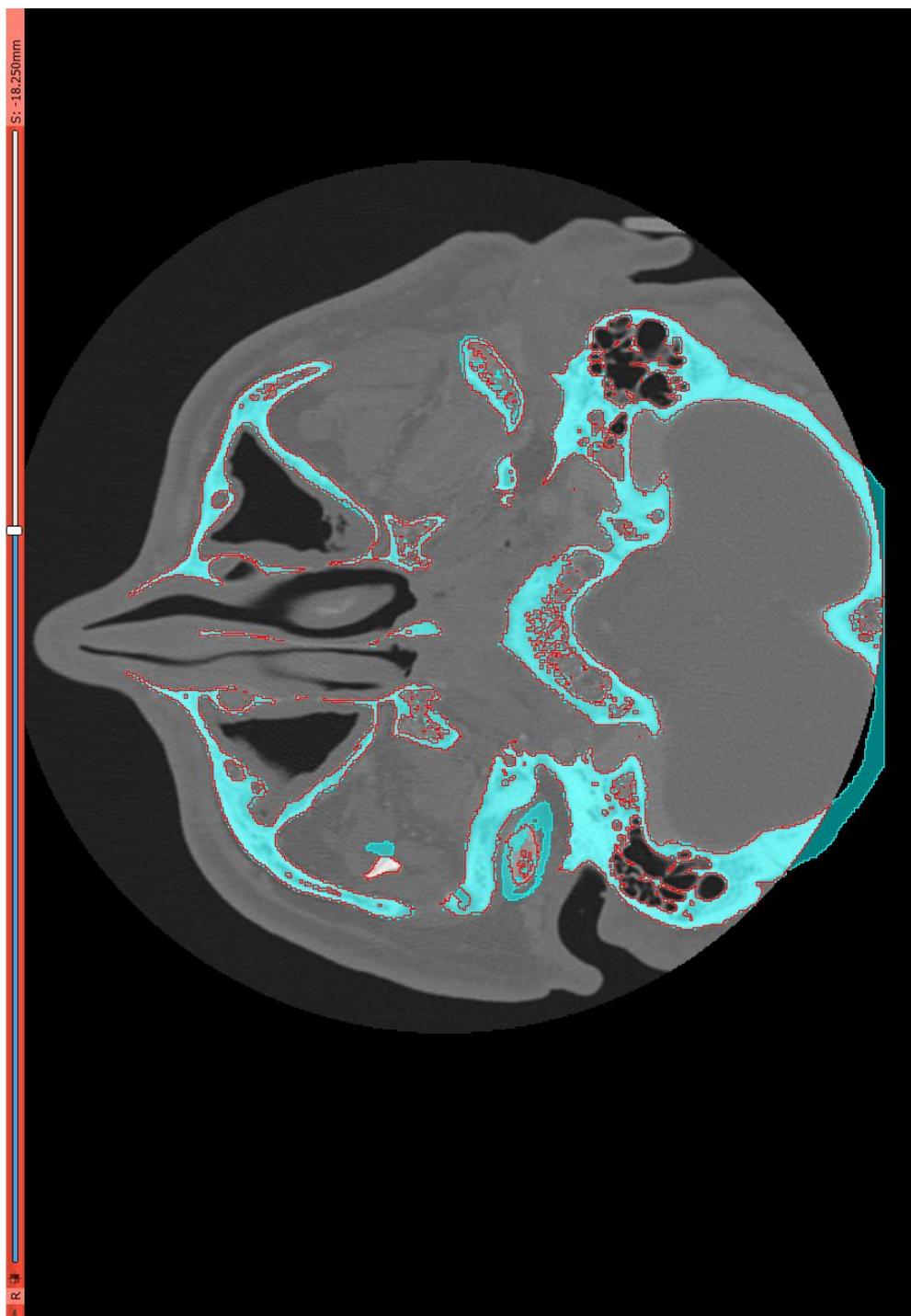


Figura 3.26: Immagine della sovrapposizione automatica di due tagli corrispondenti, dei referti del paziente B, nella vista assiale.

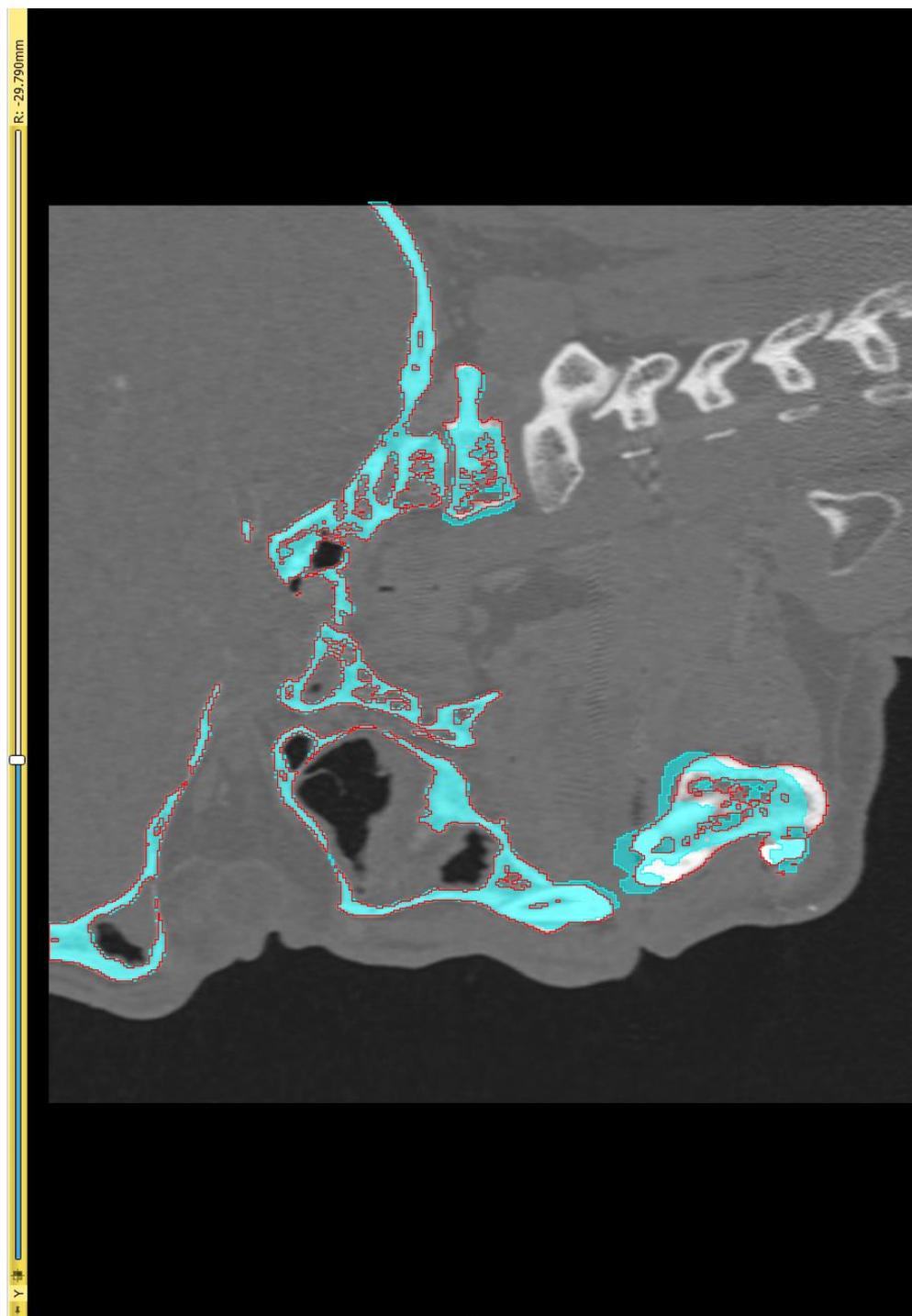


Figura 3.27: Immagine della sovrapposizione automatica di due tagli corrispondenti, dei referti del paziente B, nella vista sagittale.

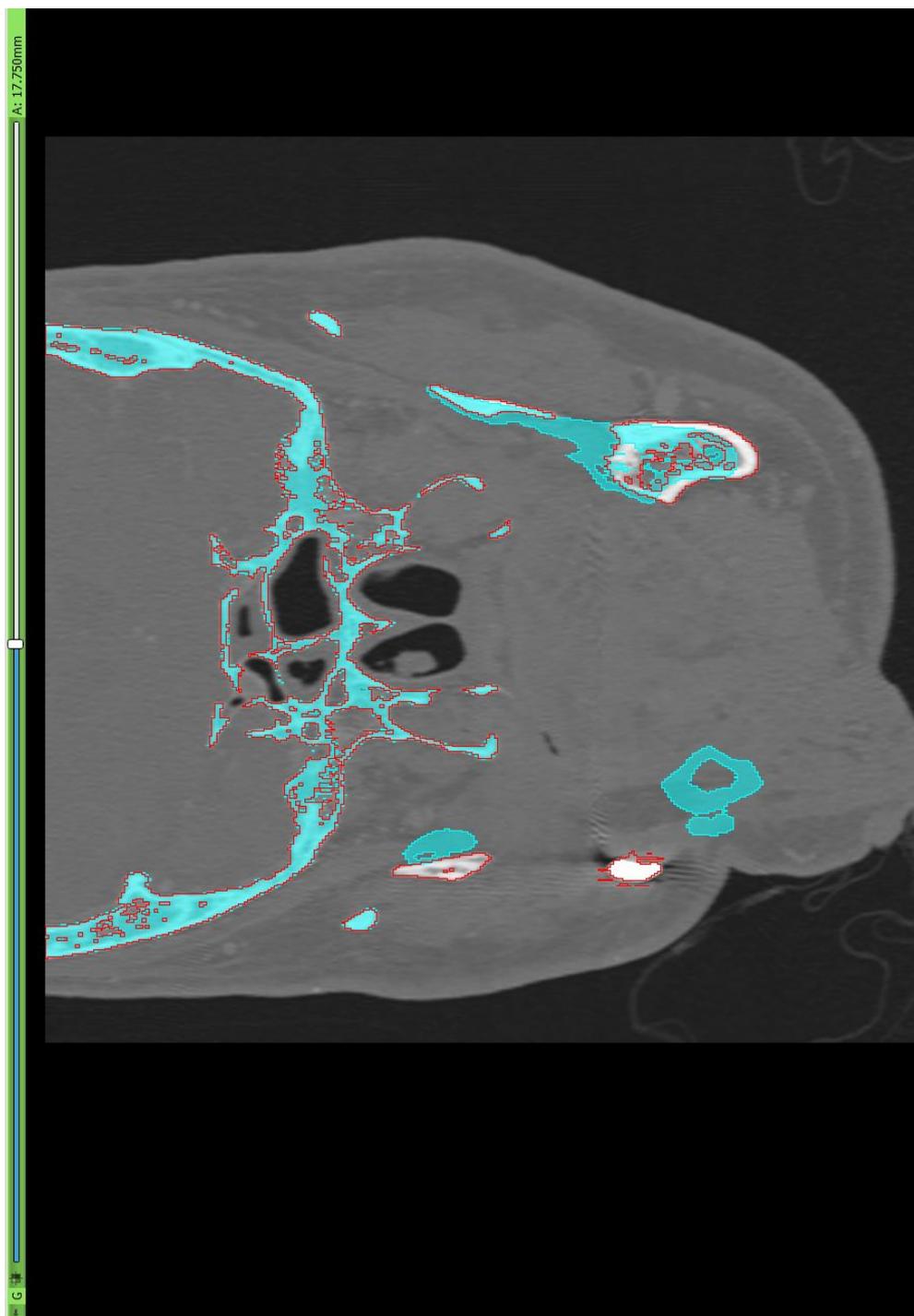


Figura 3.28: Immagine della sovrapposizione automatica di due tagli corrispondenti, dei referti del paziente B, nella vista coronale.

Registrazione basata sui landmarks:

Il procedimento seguito è quello visto per il primo paziente, i marcatori inseriti sono sempre quattro per ogni serie di tagli. I landmarks posizionati sulla TAC pre operatoria, usata come riferimento, sono stati indicati con la classe F, mentre sul referto post operatoria, usato come Template, sono definiti con la lettera M. Nelle immagini seguenti sono riportate, sul referto acquisito prima dell'intervento, le posizioni dei fiducial nelle viste assiale (finestra rossa), sagittale (finestra gialla), coronale(finestra verde).

Fiducial F_1 :

F_1 e M_1 in coincidenza della spina nasale anteriore.

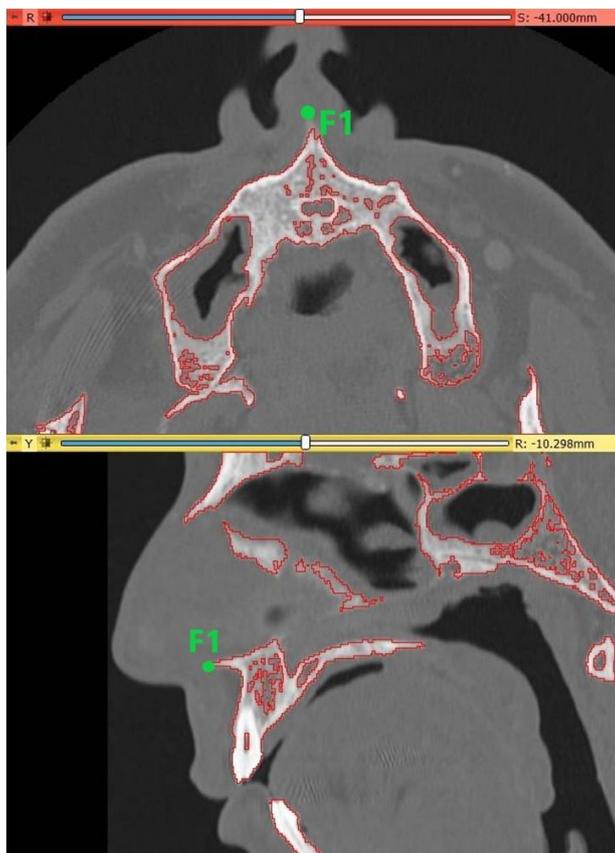


Figura 3.29: Viste assiale e sagittale della posizione del fiducial F1 sulla TAC pre operatoria del paziente B.

Fiducial F_2 :

F_2 e M_2 in corrispondenza della sutura fronto-zigomatica destra.

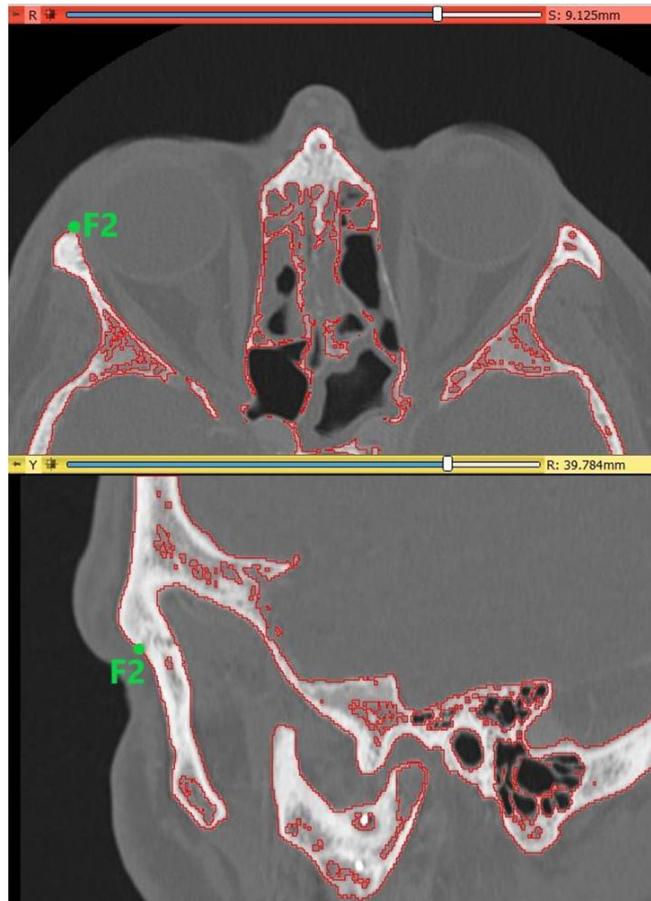


Figura 3.30: Viste assiale e sagittale della posizione del fiducial F2 sulla TAC pre operatoria del paziente B.

Fiducial F_3 :

F_3 e M_3 su una sporgenza dell'osso zigomatico sinistro.



Figura 3.31: Viste assiale e coronale della posizione del fiducial F3 sulla TAC pre operatoria del paziente B.

Fiducial F_4 :

F_4 e M_4 su una piccola protuberanza dell'osso temporale destro.

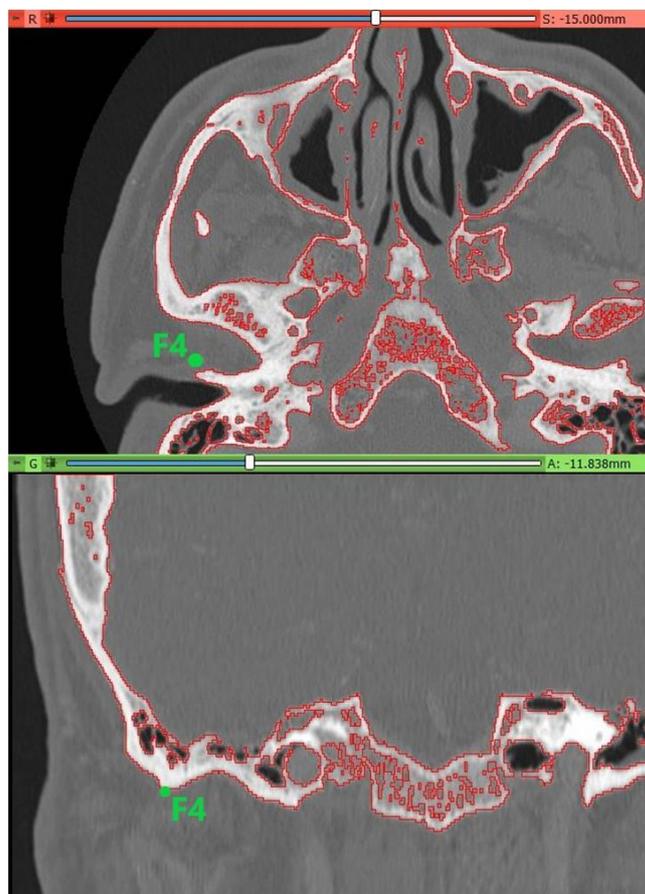


Figura 3.32: Viste assiale e coronale della posizione del fiducial F_4 sulla TAC pre operatoria del paziente B.

Per poter confrontare il risultato ottenuto mediante questo metodo di allineamento, visibile nelle Figure 3.33, 3.34 e 3.35, è stata riportata anche in questo caso la matrice di trasformazione lineare le cui componenti sono funzione dei marcatori individuati:

$$[T_c] = \begin{bmatrix} 1 & 0 & -0.02 & -5.41 \\ 0 & 0.99 & -0.07 & -7.56 \\ 0.01 & 0.06 & 1 & -11.54 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Come è possibile osservare i valori calcolati sono simili a quelli ricavati dalla registrazione automatica. E' quindi possibile considerare corretto il metodo

individuato e il programma scritto, inoltre, come è stato dimostrato per il paziente A, esso permette di ottenere sempre un risultato in tempi brevi, anche quando il materiale a disposizione è limitato.

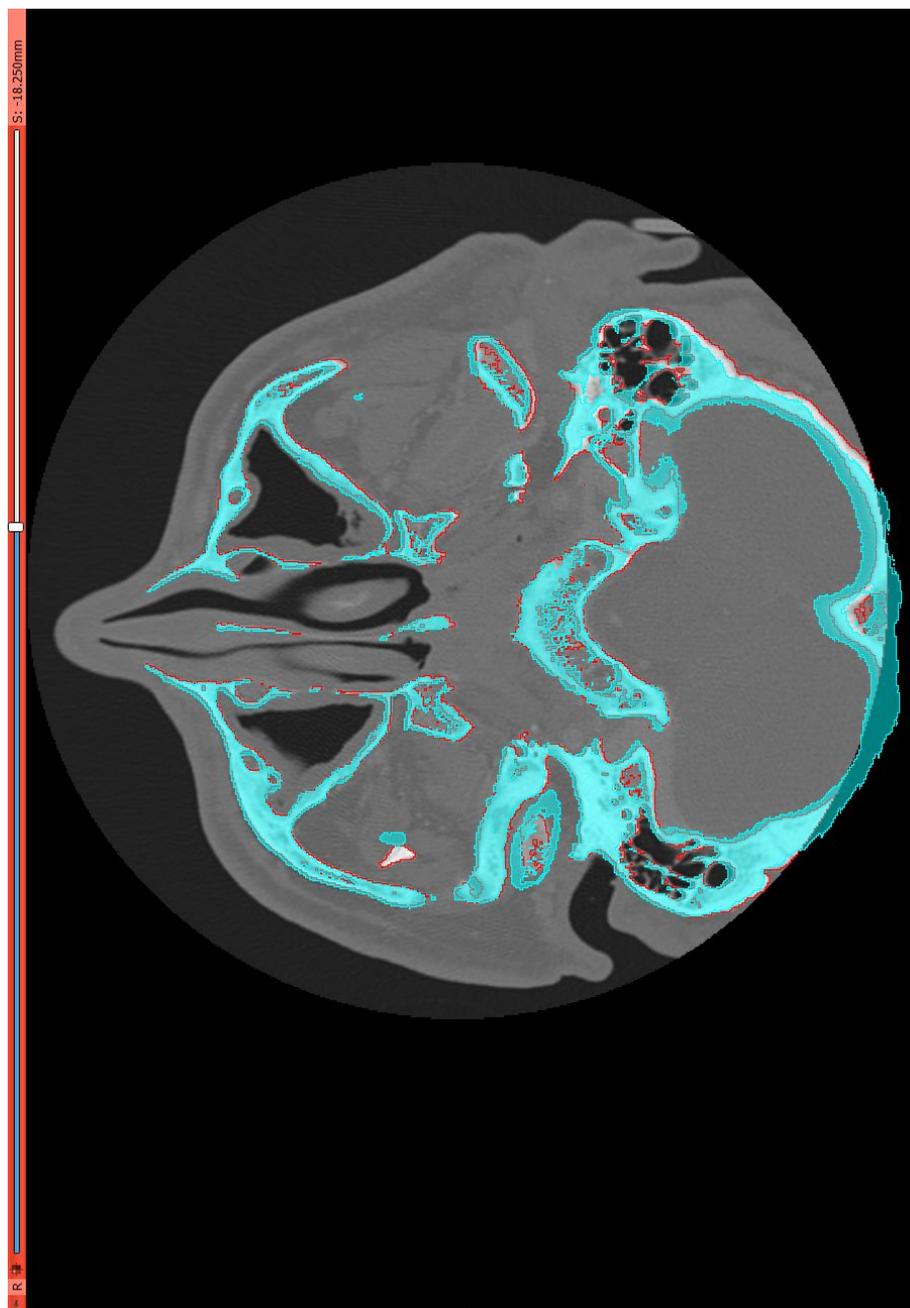


Figura 3.33: Immagine della sovrapposizione semiautomatica di due tagli corrispondenti, dei referti del paziente B, nella vista assiale.

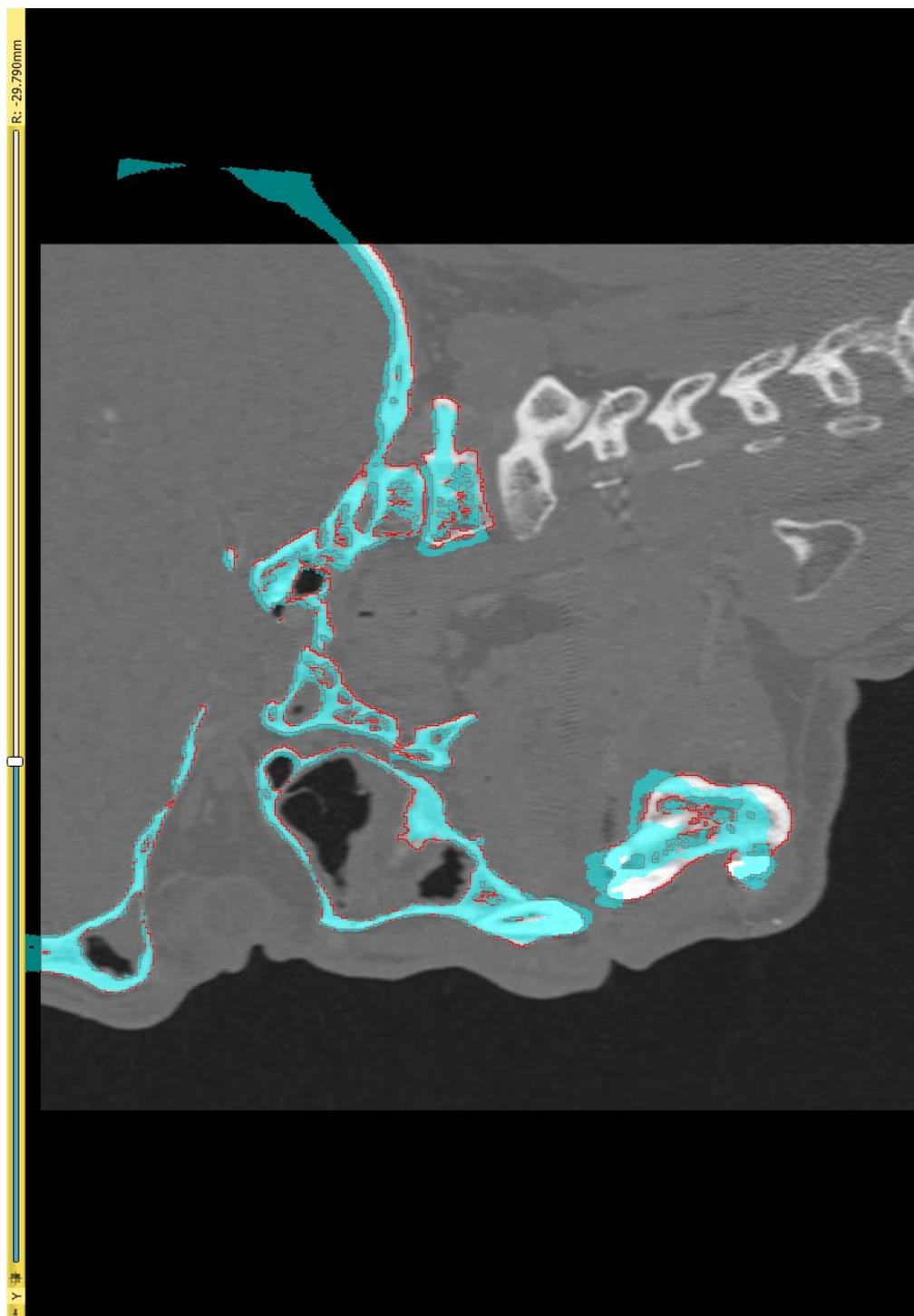


Figura 3.34: Immagine della sovrapposizione semiautomatica di due tagli corrispondenti, dei referti del paziente B, nella vista sagittale.

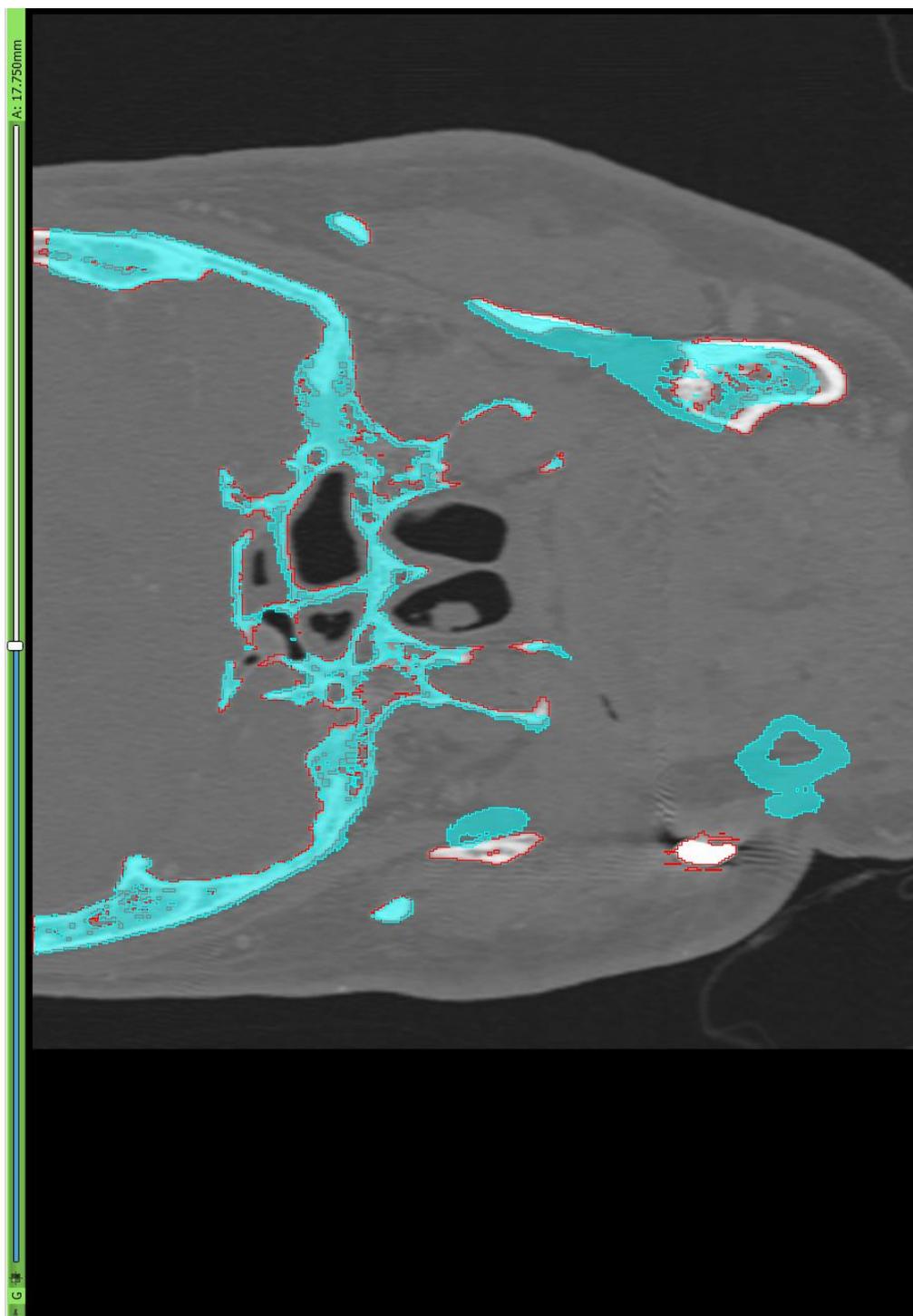


Figura 3.35: Immagine della sovrapposizione semiautomatica di due tagli corrispondenti, dei referti del paziente B, nella vista coronale.

3.3 Confronto di software per la registrazione di ricostruzioni tridimensionali pre e post operatorie

La sovrapposizione dei modelli 3D, ottenuti dalle TAC effettuate prima e dopo l'intervento, è stata eseguita mediante tre differenti software (*3D Slicer*, *SolidWorks* e *ProPlan*), per poter però valutare la bontà dei risultati così ottenuti è stato stimato lo scostamento tra le superfici mediante il programma *Geomagic Wrap*.

Lo studio eseguito, effettuato su quattro pazienti, due dei quali già utilizzati per l'esecuzione della registrazione delle immagini digitali, permette di attuare un confronto tra i software adoperati.

3.3.1 3D Slicer

Il modulo impiegato, come è stato anticipato nella sezione 3.2, è stato appositamente scritto per poter escludere automaticamente la zona operata quando essa occupa una parte considerevole del volume acquisito durante la TAC. L'algoritmo utilizzato nel programma è basato sul posizionamento, su entrambi i modelli, di quattro punti omologhi, questa corrispondenza è essenziale per la riuscita della sovrapposizione. I fiducial scelti vengono così utilizzati dal programma per calcolare la matrice trasformazione lineare $[T_c]$, ottenuta come combinazione delle trasformazioni di rotazione, scalatura e traslazione. La correlazione presente tra i landmarks e la matrice può essere espressa nel seguente modo:

$$[X_M] = [T_c] \cdot [X_F]$$

Note quindi le matrici $[X_M]$ e $[X_F]$, mediante le quali sono indicate le coordinate dei fiducial individuati rispettivamente sul volume sottoposto a deformazione e su quello di riferimento, vengono calcolate le componenti di $[T_c]$ post moltiplicando ambo i membri della relazione per la matrice inversa $[X_F]^{-1}$:

$$[T_c] = [X_M] \cdot [X_F]^{-1}$$

La trasformazione lineare ottenuta è solo funzione della posizione dei marcatori, la cui individuazione costituisce la fase più critica del processo di allineamento e richiede che i punti scelti rispecchino alcune proprietà:

- devono essere il più possibile delle caratteristiche concentrate (sporgenze, piccole cavità)
- se possibile devono essere posizionati su entrambi i lati del cranio

- devono essere presenti su entrambi i modelli
- la successione con cui vengono individuati sulle ricostruzioni tridimensionali deve essere la stessa.

I punti, suddivisi in due classi F e M , vengono utilizzati come dati in ingresso per il programma, la cui interfaccia grafica è riportata nella Figura 3.36. Il modulo restituirà invece in uscita la matrice trasformazione composta, la quale importata all'interno del modulo *Transforms* modifica spazialmente il volume assunto come Template. La matrice ottenuta può essere inoltre visualizzata in un apposito riquadro e salvata in un file di testo.

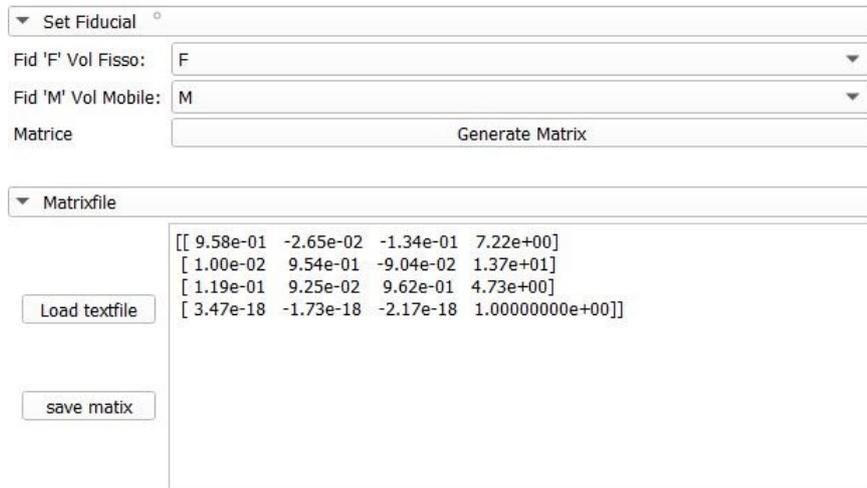


Figura 3.36: Interfaccia grafica del modulo di *3D Slicer* sviluppato per l'esecuzione della registrazione basata sui landmarks.

Di seguito sono riportati i quattro casi analizzati, per ognuno dei quali il modello ottenuto dalla TAC acquisita prima dell'intervento è usato come riferimento, mentre la ricostruzione ricavata dal referto post operatorio è usata come test.

Paziente A

Come già accennato, il primo passo consiste nell'individuazione delle caratteristiche anatomiche più idonee da utilizzare come landmarks. Nel caso del paziente A la loro posizione è riportata nella Tabella 3.9 ed è visibile nelle Figure 3.37 e 3.38:

<i>Fiducial</i>	<i>Posizione</i>
F_1 e M_1	in corrispondenza della giunzione tra le ossa nasali e le due cartilagini nasali laterali
F_2 e M_2	all'interno del foro zigomatico sinistro
F_3 e M_3	sulla sutura fronto-zigomatica sinistra
F_4 e M_4	su una sporgenza dell'osso temporale destro

Tabella 3.9: Tabella riportante l'indicazione della posizione dei marcatori individuati sui modelli 3D, ottenuti dai referti pre e post operatori del paziente A.

La convenzione di colori usata per il paziente A è: rosso per la ricostruzione tridimensionale della TAC acquisita prima dell'intervento, usata come riferimento, verde per quella ottenuta dal referto post operatorio, usata come Template.

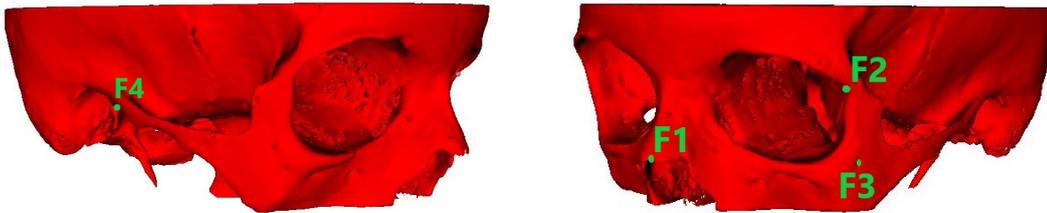


Figura 3.37: Ricostruzione delle TAC pre operatoria del paziente A con indicata la posizione dei marcatori usati per l'allineamento.

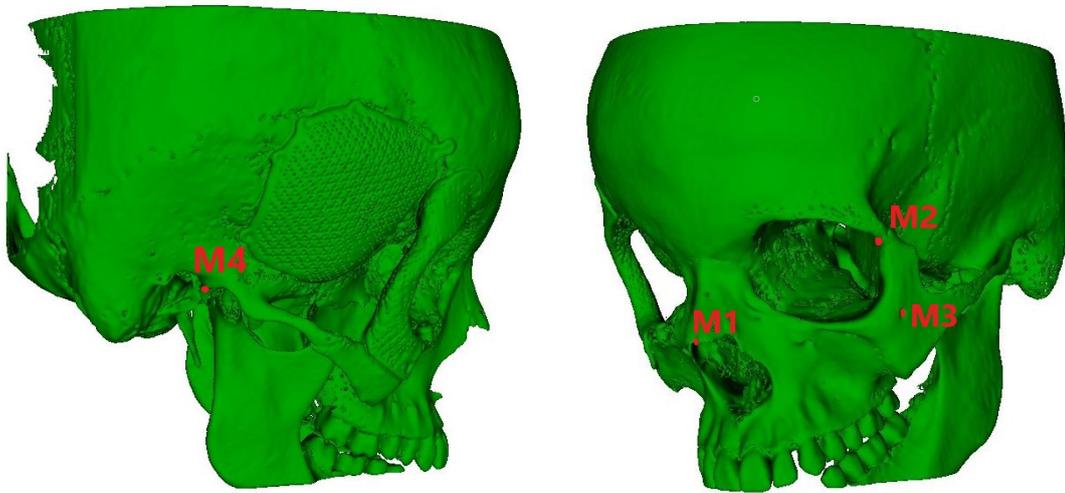


Figura 3.38: Ricostruzione delle TAC post operatoria del paziente A con indicata la posizione dei marcatori usati per la sovrapposizione.

Una volta individuati e posizionati i punti, cliccando sul pulsante "*Generate matrix*", il programma restituisce la matrice di trasformazione lineare che, importata all'interno del modulo *Transforms*, modifica il volume Template per adattarlo al riferimento. Nella Figura 3.39 è riportata la sovrapposizione ottenuta la quale è associata alla seguente matrice di trasformazione lineare:

$$[T_c] = \begin{bmatrix} 0.96 & -0.03 & -0.13 & 7.22 \\ 0.01 & 0.95 & -0.09 & 13.74 \\ 0.12 & 0.09 & 0.96 & 4.73 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

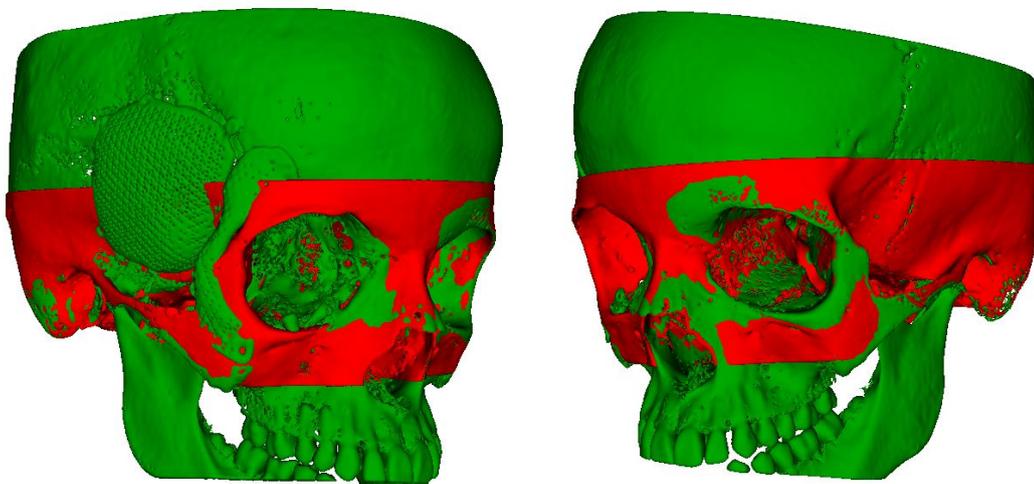


Figura 3.39: Immagine raffigurante la sovrapposizione, ottenuta mediante il modulo di 3D Slicer, delle ricostruzioni tridimensionali pre e post operatorie del cranio del paziente A.

Per valutare la riuscita dell'allineamento vengono importati i due modelli sovrapposti nell'ambiente di lavoro di *Geomagic Wrap*, software che permette di quantificare lo scostamento presente tra le superfici dei due crani. Questa distanza può essere visualizzata sia attraverso una mappatura a colori sia mediante l'utilizzo di vettori.

Per eseguire lo studio è necessario definire il modello di riferimento, rispetto al quale viene misurato lo scostamento, il volume usato come test e i valori che definiscono i limiti degli intervalli di analisi. Le scelte effettuate e mantenute costanti per tutti i casi sono riportate nelle Tabelle 3.10 e 3.11 .

Scelta della convenzioni per i volumi	
Volume di riferimento	modello pre operatorio
Volume di test	modello post operatorio

Tabella 3.10: Scelte attuate, riguardo il volume di riferimento e quello di test, per l'esecuzione dell'analisi mediante *Geomagic Wrap* e valide per tutti i pazienti.

Limiti	Valori (mm)
Critico max	2
Nominale max	0.25
Nominale min	-0.25
Critico min	-2

Tabella 3.11: Scelte attuate nella definizione dell'intervallo di tolleranza degli scostamenti, per l'esecuzione dell'analisi mediante *Geomagic Wrap* e valide per tutti i pazienti.

Scostamenti rappresentati mediante mappatura a colori

Il cranio che viene raffigurato è quello assunto come riferimento, la sua superficie viene colorata secondo uno spettro i cui limiti vengono imposti dall'utente e coincidono con quelli della Tabella 3.11: in verde le aree dei due modelli che presentano uno scostamento inferiore a 0.25 mm e possono quindi essere considerate idealmente sovrapposte, in giallo e azzurro le zone con un allineamento non perfetto, ma ancora accettabile, mentre in rosso e in blu le parti del cranio fuori tolleranza, ossia situate a distanze maggiori di 2 mm . Dalla figura 3.40 è possibile osservare come nell'emisfero sinistro i colori preponderanti siano il verde e l'azzurro, quindi la sovrapposizione in questa regione può essere considerata buona. La maggiore presenza di aree verdi nella parte anteriore del cranio può essere ricondotta alla distribuzione in questa zona di tre marcatori su quattro.

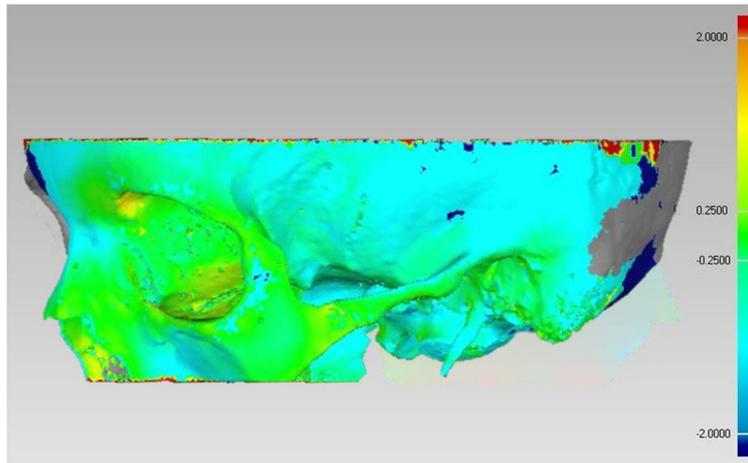


Figura 3.40: Vista laterale sinistra della mappatura a colori del cranio del paziente A.

Come ci si aspetta, le Figure 3.41 e 3.42 mostrano invece come le regioni

che si trovano al di fuori dell'intervallo di tolleranza, in colore rosso e blu, siano situate nell'emisfero destro, ossia nella parte del cranio sottoposta ad intervento.

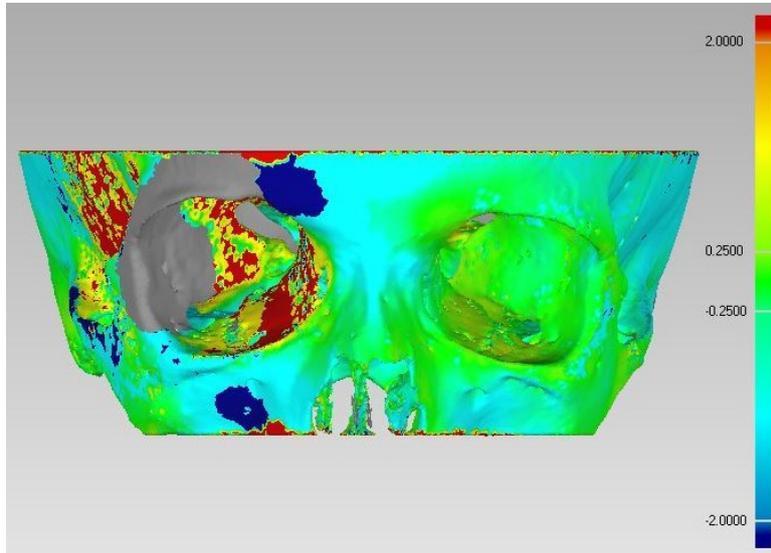


Figura 3.41: Vista frontale della mappatura a colori del cranio del paziente A.

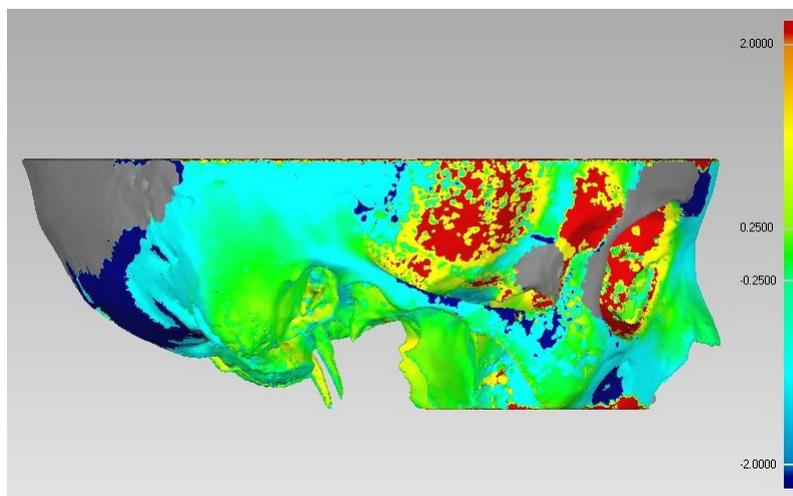


Figura 3.42: Vista laterale destra della mappatura a colori del cranio del paziente A.

Rappresentazione delle distanze mediante mappa vettoriale

Lo scostamento tra le superfici può essere rappresentato anche mediante vet-

tori la cui lunghezza e colore è funzione della distanza del punto considerato della mesh del modello Template dalla superficie del riferimento.

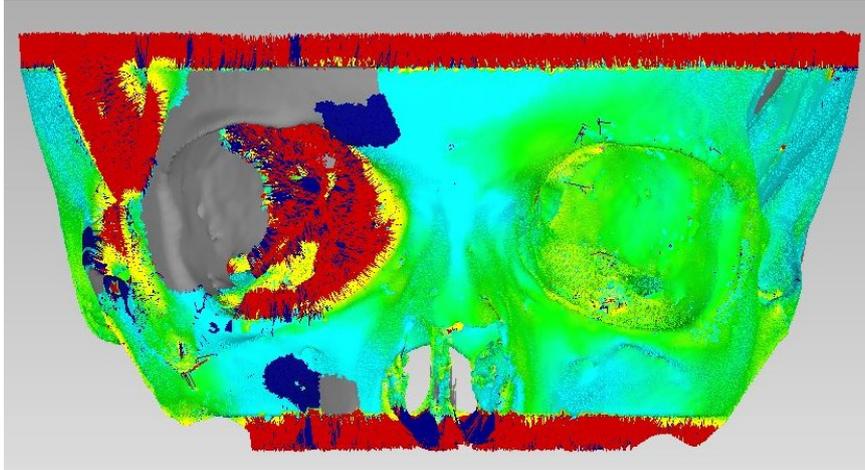


Figura 3.43: Vista frontale della mappa vettoriale del cranio del paziente A.

Geomagic Wrap fornisce inoltre ulteriori informazioni ricavate in modo statistico e riportate nella Tabella 3.12.

		VALORE (mm)
Distanza media		0.05
Deviazione standard		1.54
Distanza massima	Positiva	5
	Negativa	-5

Tabella 3.12: Tabella riportante i valori degli indici statistici forniti da *Geomagic Wrap* sul paziente A.

La Deviazione standard è un indice di dispersione statistico il quale dà un'indicazione della distribuzione dei dati intorno ad un valore medio. Può essere definito mediante la seguente formula:

$$\bar{s}_X = \sqrt{\frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N - 1}}$$

con N numero di unità statistiche che costituisce la popolazione, \bar{x} valore medio, x_i i -esimo componente della popolazione.

Paziente B

Durante l'esecuzione di entrambe le TAC è stato rilevato l'intero volume del cranio. Essendo però la mandibola sede dell'intervento chirurgico in questo soggetto, essa non può essere utilizzata per effettuare l'allineamento ed è stata perciò eliminata dal modello. Al contrario del paziente A, è stato quindi possibile collocare i marcatori su entrambi i lati della testa e in particolare:

<i>Fiducial</i>	<i>Posizione</i>
F_1 e M_1	sulla spina nasale anteriore
F_2 e M_2	in corrispondenza della sutura fronto-zigomatica destra
F_3 e M_3	su una sporgenza dell'osso zigomatico sinistro
F_4 e M_4	su una piccola protuberanza dell'osso temporale destro

Tabella 3.13: Tabella riportante l'indicazione della posizione dei marcatori individuati sui modelli 3D, ottenuti dai referti pre e post operatori del paziente B.

Mediante le immagini riportate di seguito è inoltre possibile osservare la loro posizione sulla ricostruzione tridimensionale. Dal momento che il paziente B è già stato studiato nella Sezione 3.2 è stata mantenuta la stessa convenzione di colori, ossia rosso per il modello pre operatorio e azzurro per quello post operatorio.

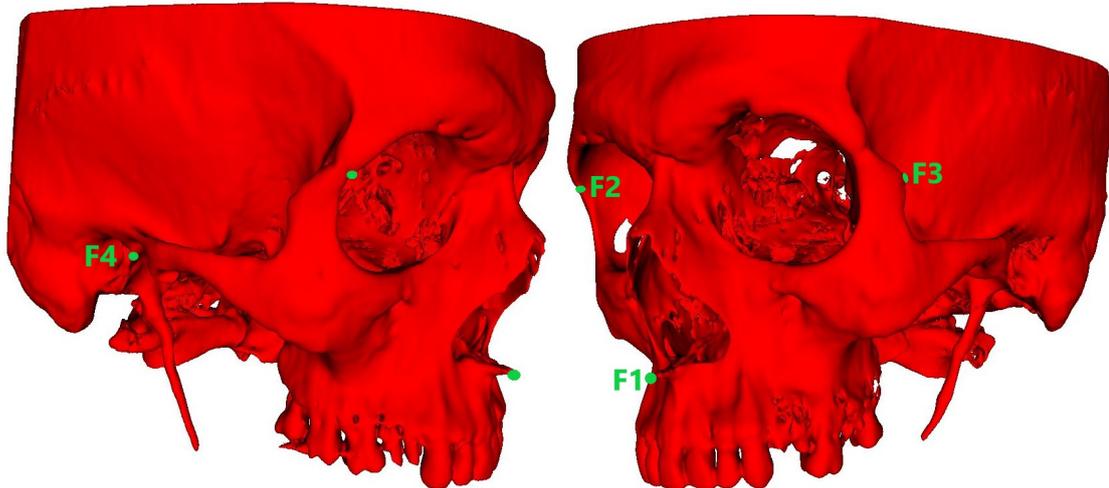


Figura 3.44: Ricostruzione delle TAC pre operatoria del paziente B con indicata la posizione dei marcatori usati per l'allineamento.

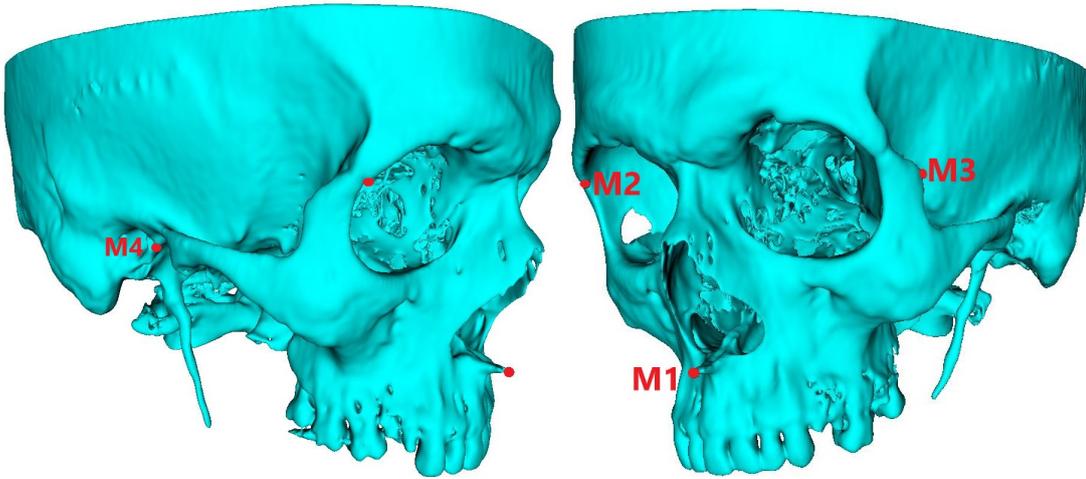


Figura 3.45: Ricostruzione delle TAC post operatoria del paziente B con indicata la posizione dei marcatori usati per la sovrapposizione.

La sovrapposizione così ottenuta è mostrata nella Figura 3.46 e come visto in precedenza essa è associata alla matrice di trasformazione $[T_c]$:

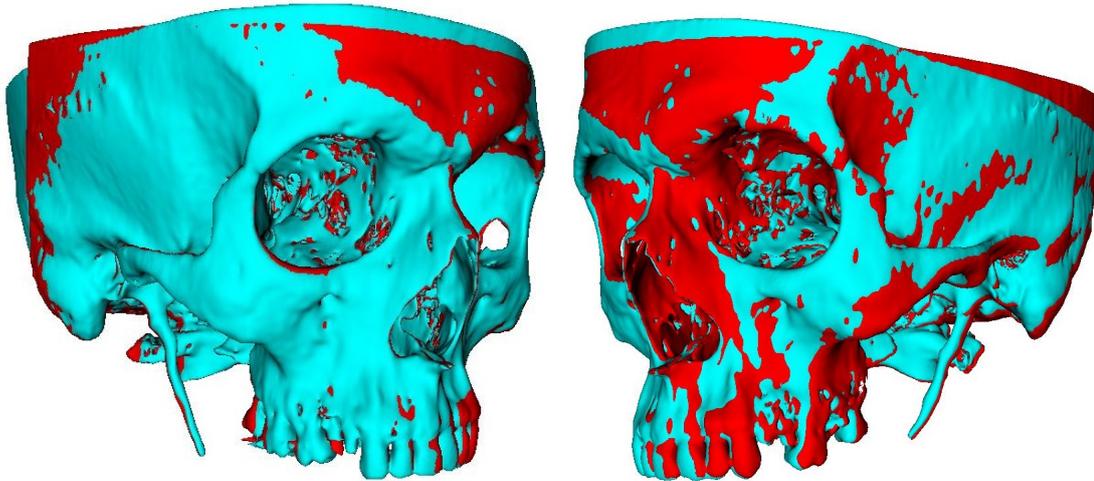


Figura 3.46: Immagine raffigurante la sovrapposizione, ottenuta mediante il modulo di 3D Slicer, delle ricostruzioni tridimensionali pre e post operatorie del cranio del paziente B.

$$[T_c] = \begin{bmatrix} 1 & 0 & -0.02 & -5.46 \\ 0 & 1 & -0.07 & -8.30 \\ 0.02 & 0.07 & 1 & -11.90 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Nell'immagine raffigurante l'allineamento è visibile un'alternanza dei colori rosso e azzurro, questo risultato, che indica una copresenza delle due superfici, preannuncia una buona riuscita della sovrapposizione. Quindi ci aspetteremo, nell'analisi effettuata mediante *Geomagic Wrap*, una maggiore presenza del colore verde rispetto agli altri.

Scostamento raffigurato mediante mappatura a colori

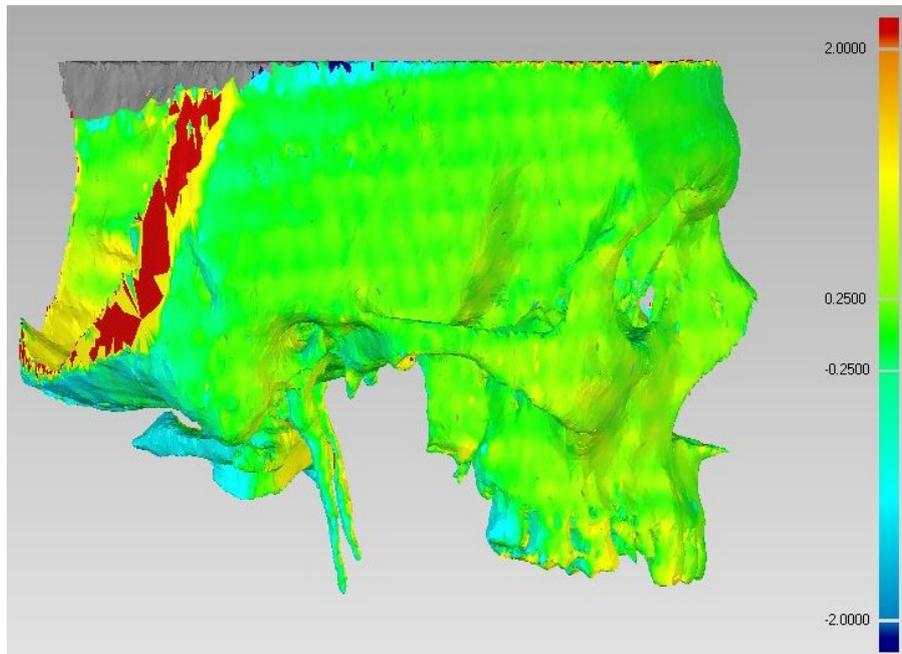


Figura 3.47: Vista laterale destra della mappatura a colori del cranio del paziente B.

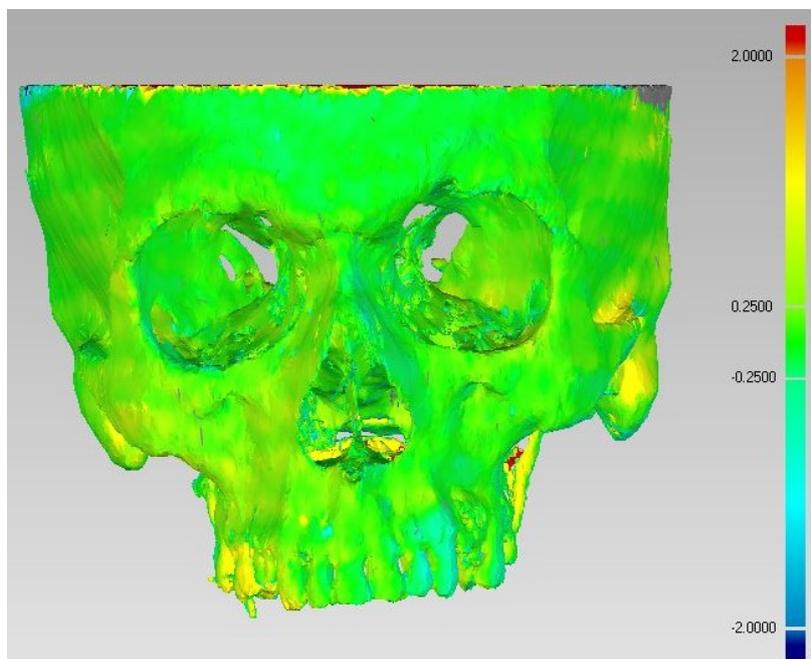


Figura 3.48: Vista frontale della mappatura a colori del cranio del paziente B.

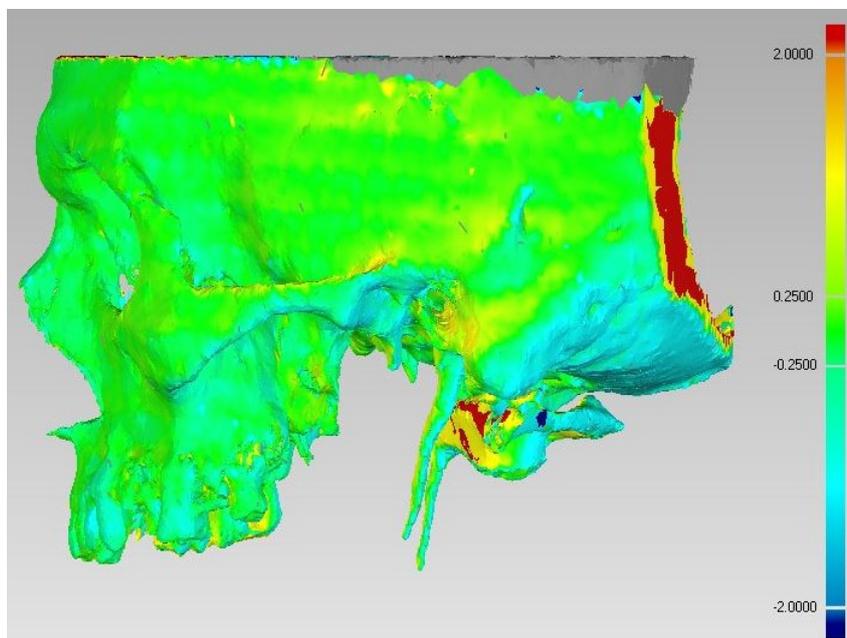


Figura 3.49: Vista laterale sinistra della mappatura a colori del cranio del paziente B.

Come accennato, tutte le immagini presentano una preponderanza del colore verde, questo ad indicare che le distanze tra le superfici dei due crani sono inferiori a 0.25 mm . Piccoli scostamenti sono invece rilevabili in corrispondenza dell'osso occipitale, dall'analisi effettuata deve però essere esclusa la prima vertebra, dal momento che il suo movimento è indipendente da quello del cranio.

Distanze raffigurate mediante mappa vettoriale

La rappresentazione con i vettori, Figura 3.50, mostra come i punti, ai quali corrisponde una distanza maggiore dell'intervallo di tolleranza, in questo caso $d \geq 2 \text{ mm}$, siano situati lungo i contorni della ricostruzione tridimensionali.

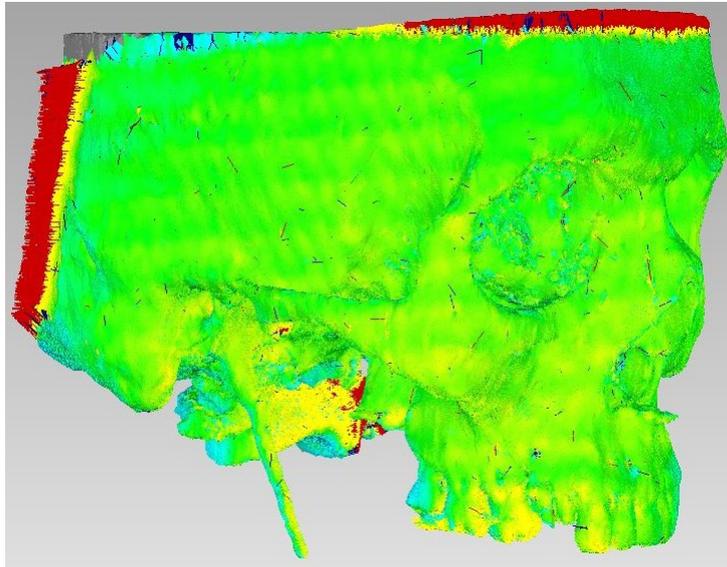


Figura 3.50: Vista laterale della mappa a vettori del cranio del paziente B.

Ulteriori indicazioni dell'ottima riuscita dell'allineamento sono la distanza media e la deviazione standard riportati nella Tabella 3.14.

		VALORE (mm)
Distanza media		0.1
Deviazione standard		0.82
Distanza massima	Positiva	6
	Negativa	-6

Tabella 3.14: Tabella riportante i valori degli indici statistici forniti da *Geomagic Wrap* sul paziente B.

Paziente C

Il volume rilevato nella TAC pre operatoria del paziente C è esiguo e comprende la mascella, parte delle ossa zigomatiche e temporali. Visto le zone del cranio rilevate e la necessità di disporre quattro marcatori in un volume concentrato, sono stati riscontrati non pochi problemi nell'identificazione delle caratteristiche anatomiche ideali per l'esecuzione della registrazione. Le posizioni individuate sono riportate nella Tabella 3.15 e sono visibili nelle Figure 3.51 e 3.52.

<i>Fiducial</i>	<i>Posizione</i>
F_1 e M_1	sulla spina nasale anteriore
F_2 e M_2	sull'incisivo centrale destro
F_3 e M_3	sul secondo molare di sinistra
F_4 e M_4	su una piccola rientranza dell'osso temporale destro

Tabella 3.15: Tabella riportante l'indicazione della posizione dei marcatori individuati sui modelli 3D, ottenuti dai referti pre e post operatori del paziente C.

Il cranio raffigurato con il colore rosso è quello ottenuto dal referto pre operatorio, mentre in verde è rappresentata la ricostruzione della tac acquisita dopo l'intervento.

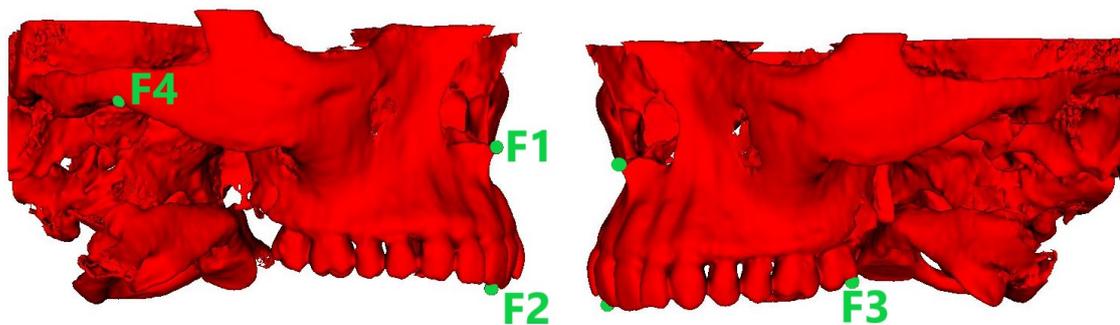


Figura 3.51: Ricostruzione delle TAC pre operatoria del paziente C con indicata la posizione dei marcatori usati per la sovrapposizione.

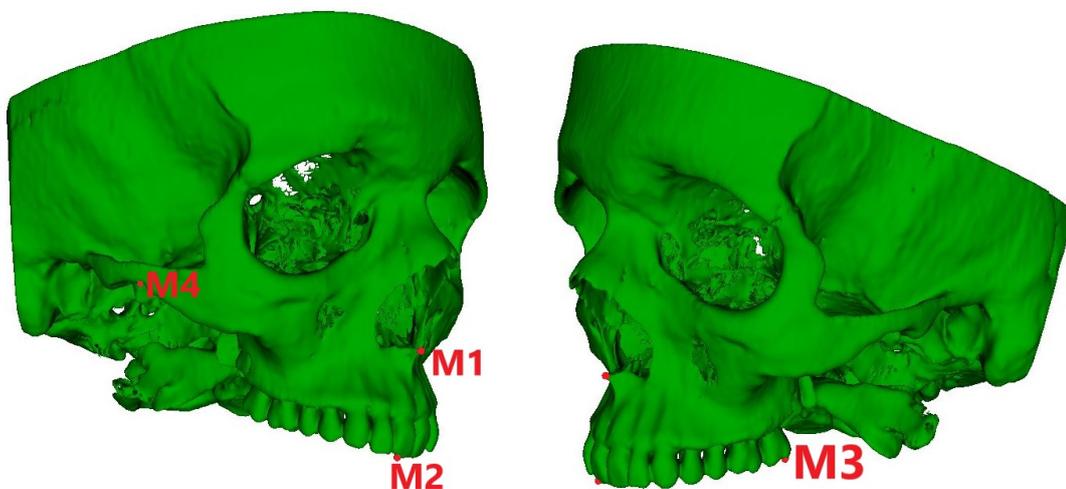


Figura 3.52: Ricostruzione delle TAC post operatoria del paziente C con indicata la posizione dei marcatori usati per la registrazione.

Applicando quindi la matrice di trasformazione lineare $[T_c]$ derivante dai landmarks posizionati viene ottenuta la sovrapposizione visibile nella Figura 3.53.

$$[T_c] = \begin{bmatrix} 0.98 & 0.09 & -0.06 & 4.10 \\ -0.10 & 0.91 & -0.38 & -31.00 \\ 0.04 & 0.37 & 0.92 & 38.27 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

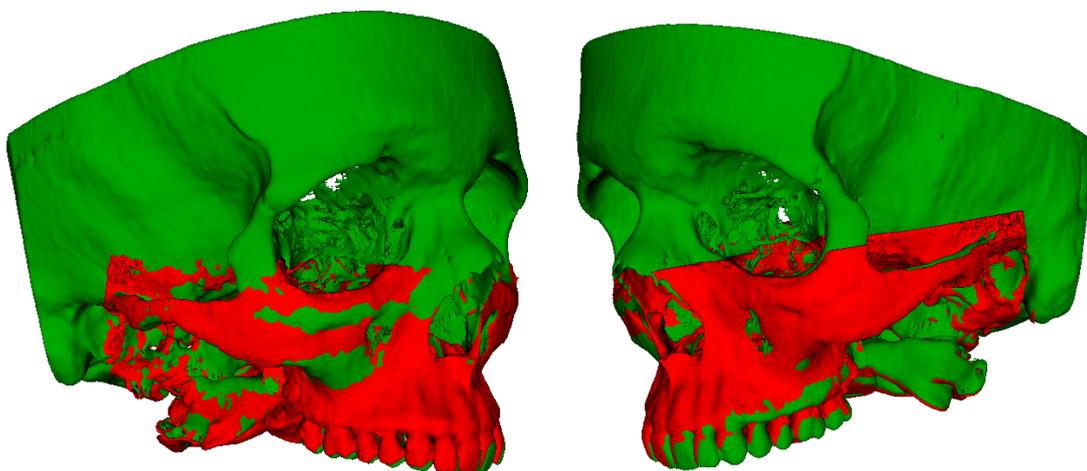


Figura 3.53: Immagine raffigurante la sovrapposizione, ottenuta mediante il modulo di 3D Slicer, delle ricostruzioni tridimensionali pre e post operatorie del cranio del paziente C.

Considerando le immagini soprastanti è possibile ipotizzare che l'emisfero destro sia perfettamente allineato mentre quello sinistro presenti uno scostamento maggiore di 0.25 mm ; Per valutare la bontà della registrazione ottenuta si è ricorsi anche in questo caso al software *Geomagic Wrap*.

Scostamenti rappresentati mediante mappatura a colori

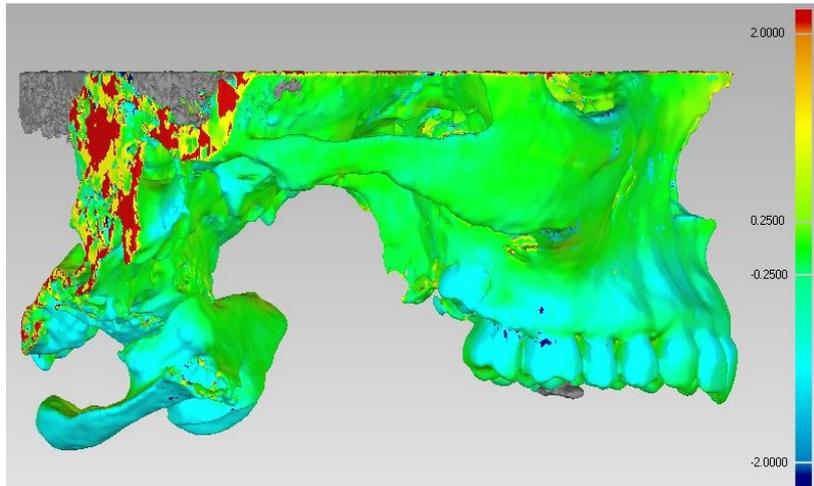


Figura 3.54: Vista laterale destra della mappatura a colori del cranio del paziente C.

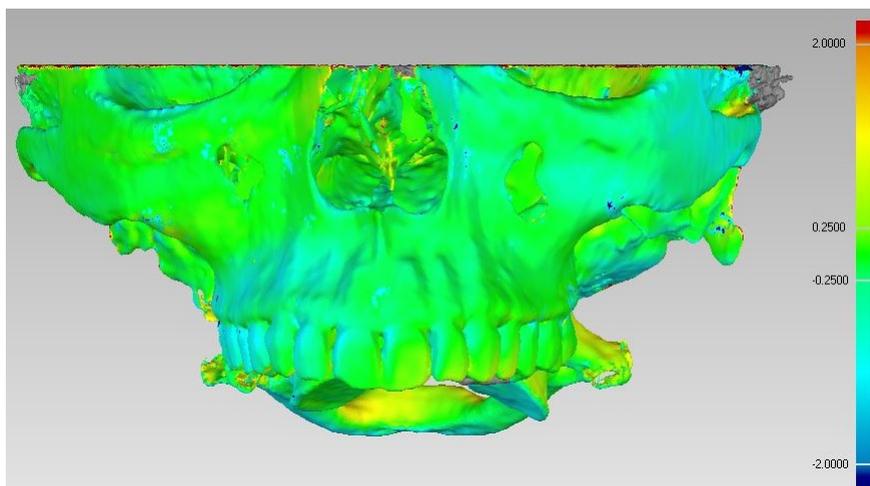


Figura 3.55: Vista frontale della mappatura a colori del cranio del paziente C.

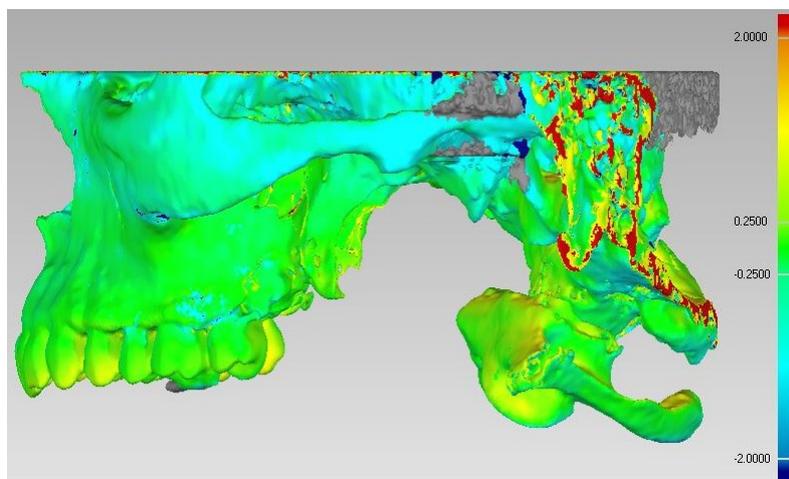


Figura 3.56: Vista laterale sinistra della mappatura a colori del cranio del paziente C.

La mappa a colori riportata mostra che lo scostamento tra le superfici nella parte destra del cranio appartiene all'intervallo $[-0.25 \text{ mm}; 0.25 \text{ mm}]$. Al contrario, come previsto dall'osservazione delle sovrapposizioni (fig.3.53), le ossa temporali sinistre si trovano ad una distanza maggiore, ma comunque all'interno dell'intervallo di tolleranza. Inoltre, il colore azzurro della superficie indica che il cranio post operatorio è situato leggermente a destra rispetto a quello pre operatorio, usato come riferimento.

Distante rappresentate mediante mappa vettoriale

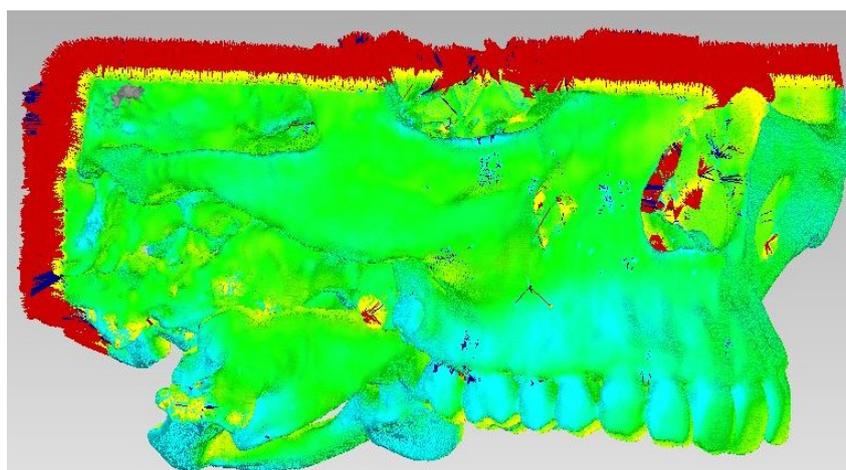


Figura 3.57: Vista laterale della mappa a vettori del cranio del paziente C.

Anche per il paziente C i punti fuori tolleranza sono collocati lungo il contorno del modello.

Gli ulteriori dati forniti dal programma di analisi sono riportati nella Tabella 3.16.

		VALORE (mm)
Distanza media		0.24
Deviazione standard		1.53
Distanza massima	Positiva	6.4
	Negativa	-6.4

Tabella 3.16: Tabella riportante i valori degli indici statistici forniti da *Geomagic Wrap* sul paziente C.

Paziente D

Per il posizionamento dei fiducial nel caso del paziente D è stato necessario escludere, a causa dell'intervento, le ossa zigomatiche e il processo orbitale dell'osso mascellare. I fiducial sono quindi stati situati sull'osso frontale, su quello temporale destro e nella regione dell'osso mascellare non interessata dall'operazione; la disposizione precisa è riportata nella Tabella 3.17 ed è visibile nelle Figure 3.58 e 3.59:

<i>Fiducial</i>	<i>Posizione</i>
F₁ e M₁	sulla spina nasale anteriore
F₂ e M₂	nel foro sopraorbitale destro
F₃ e M₃	nell'incisura sopraorbitale sinistra
F₄ e M₄	in corrispondenza di una rientranza dell'osso temporale destro

Tabella 3.17: Tabella riportante l'indicazione della posizione dei marcatori sui modelli 3D, ottenuti dai referti pre e post operatori del paziente D.

La convenzione di colori adottata è rosso per il cranio ottenuto dalla TAC pre operatoria, e verde per quello ricavato della TAC acquisita dopo l'intervento.

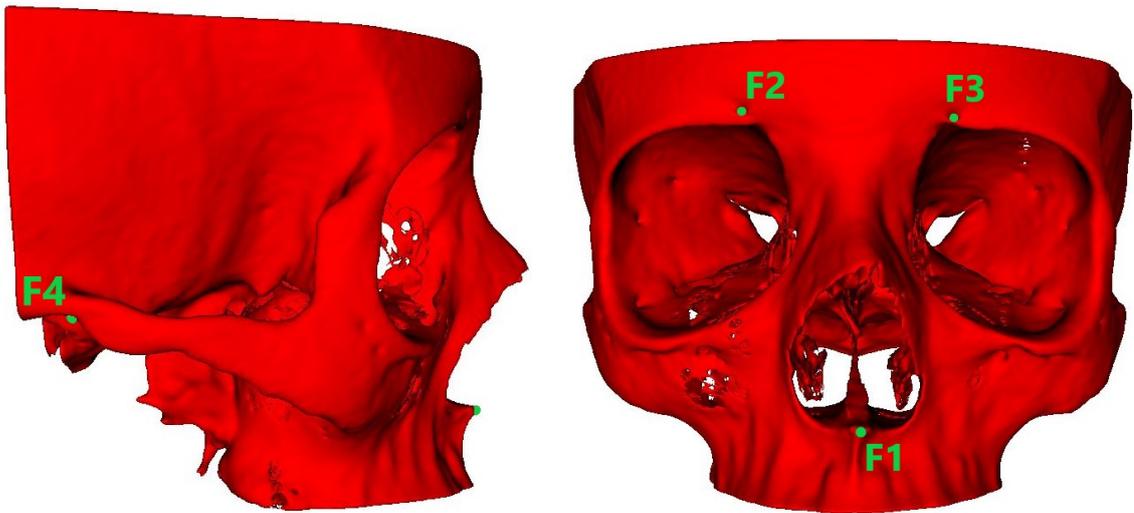


Figura 3.58: Ricostruzione delle TAC pre operatoria del paziente D con indicata la posizione dei marcatori usati per la sovrapposizione.

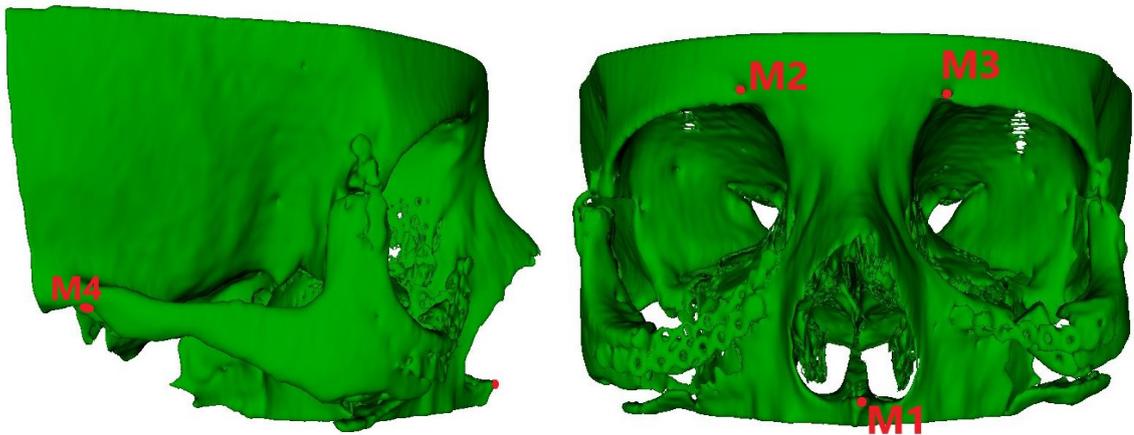


Figura 3.59: Ricostruzione delle TAC post operatoria del paziente D con indicata la posizione dei marcatori usati per la registrazione.

La matrice di trasformazione lineare restituita dal programma fornisce la sovrapposizione visibile nella figura 3.60

$$[T_c] = \begin{bmatrix} 1 & 0 & -0.02 & -0.79 \\ 0.03 & 1.01 & 0.03 & 15.17 \\ 0.01 & -0.03 & 0.99 & -6.97 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

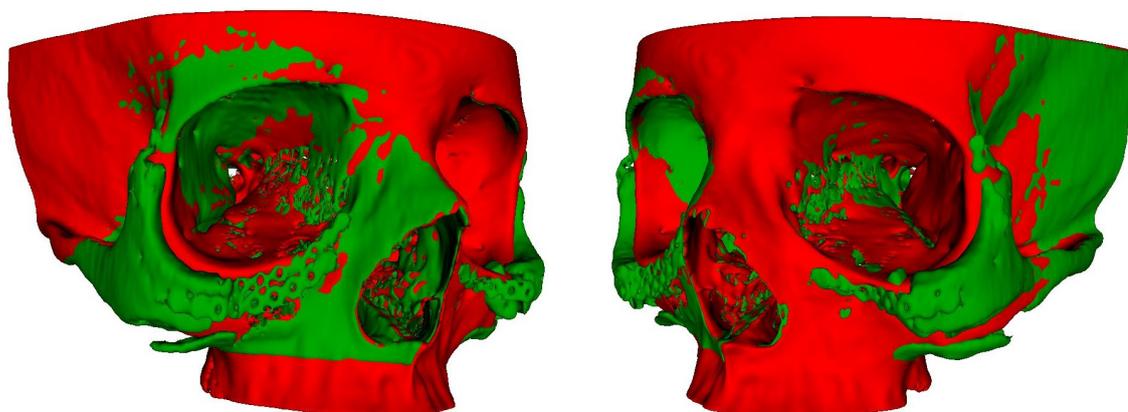


Figura 3.60: Immagine raffigurante l'allineamento, ottenuto mediante il modulo di 3D Slicer, delle ricostruzioni tridimensionali pre e post operatorie del cranio del paziente D.

Dal momento che l'operazione ha interessato un'area molto estesa del viso, l'unico modo per capire se la sovrapposizione eseguita ha prodotto un risultato accettabile è quello di condurre un'analisi mediante *Geomagic Wrap*.

Rappresentazione degli scostamenti mediante mappatura a colori

Confrontando le Figure 3.61, 3.62 e 3.63 è facile osservare come le aree di colore rosso siano situate sull'osso zigomatico e in corrispondenza delle placche in titanio applicate durante l'intervento. La preponderanza del colore verde sul viso dimostra un buon allineamento in questa regione, dovuto molto probabilmente al posizionamento di tre marcatori su quattro nella parte anteriore del cranio. Mentre la presenza di piccole zone di colore azzurro sulla superficie destra del cranio e gialle su quella sinistra può essere dovuto sia al contributo della scalatura, anche se minimo, sia alla fase di segmentazione che ha portato ad ottenere spessori delle ossa leggermente differenti.

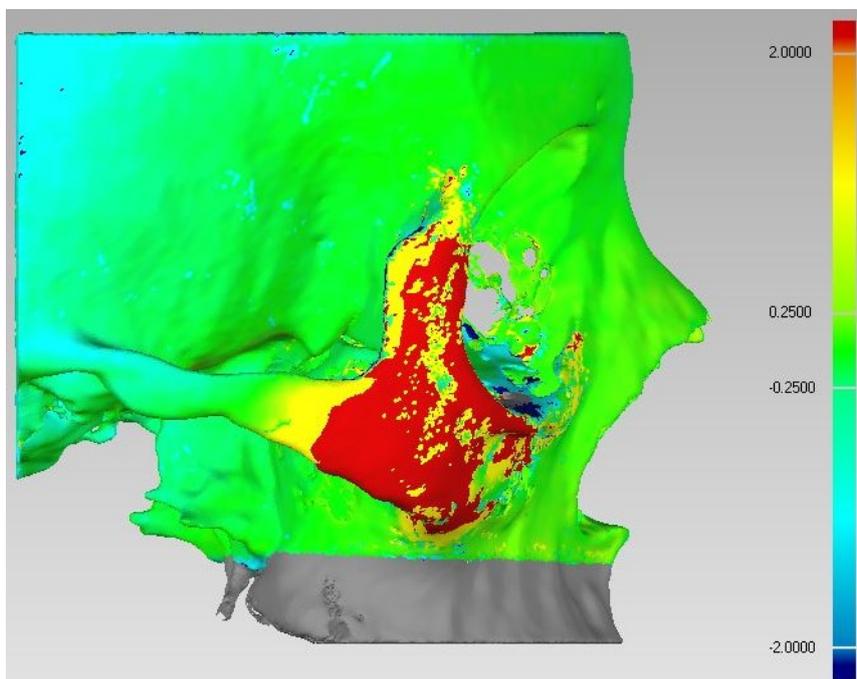


Figura 3.61: Vista laterale destra della mappatura a colori del cranio del paziente D.

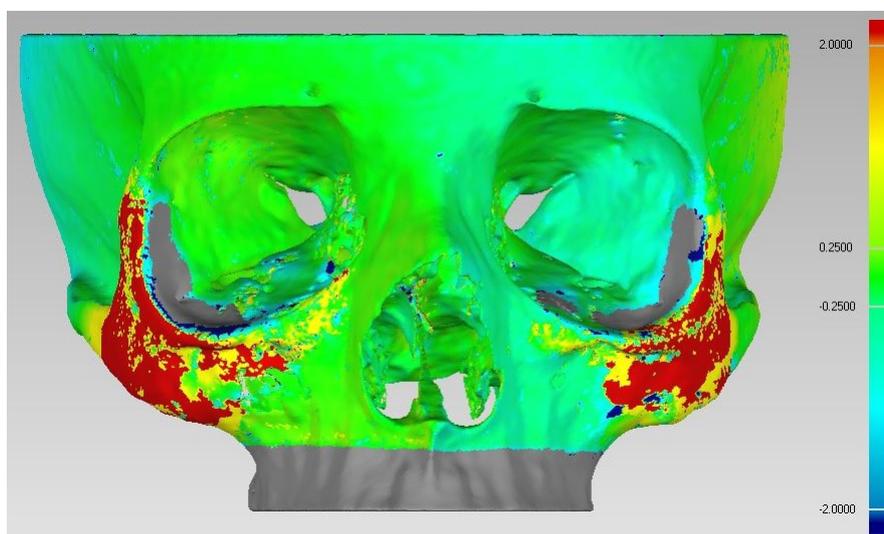


Figura 3.62: Vista frontale della mappatura a colori del cranio del paziente D.

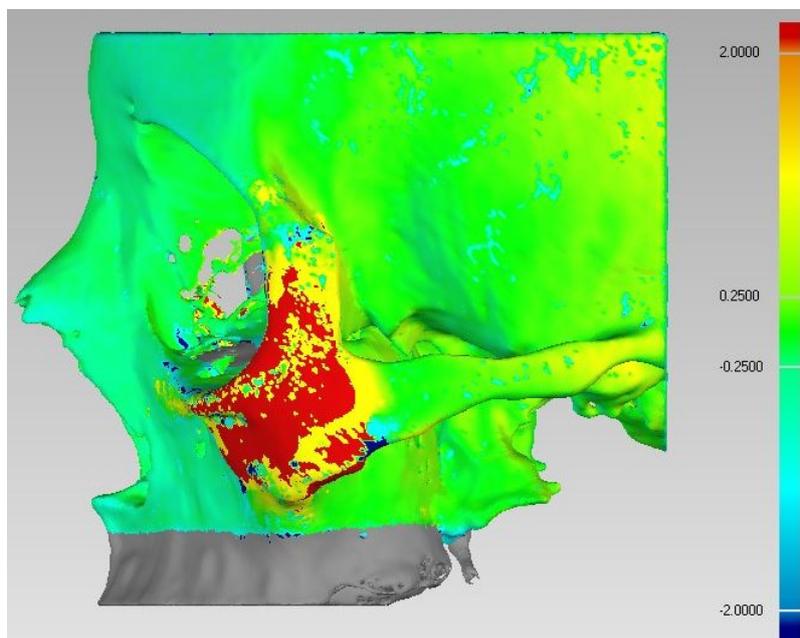


Figura 3.63: Vista laterale sinistra della mappatura a colori del cranio del paziente D.

Raffigurazione delle distanze mediante mappa vettoriale

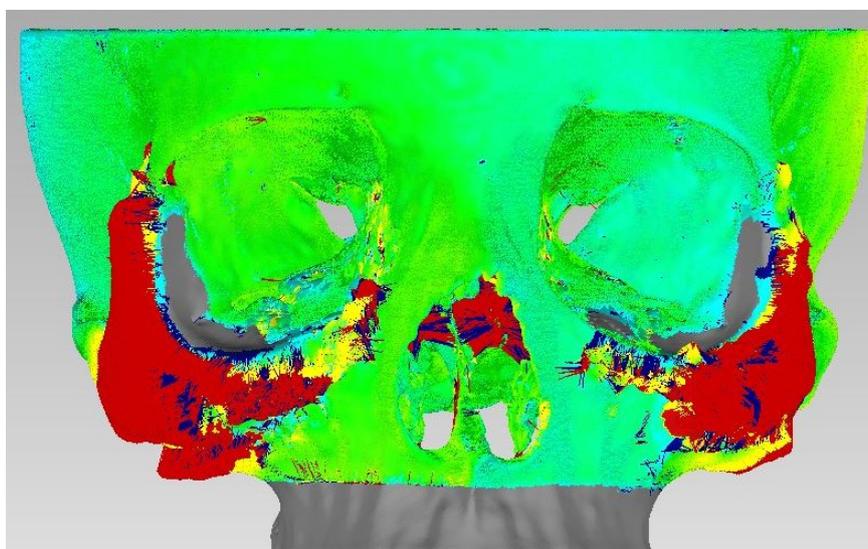


Figura 3.64: Vista frontale della mappa a vettori del cranio del paziente D.

Come già visto per gli altri pazienti, le informazioni fornite da *Geomagic Wrap* sono riportate nella Tabella 3.18.

		VALORE (mm)
Distanza media		0.04
Deviazione standard		1.22
Distanza massima	Positiva	6
	Negativa	-6

Tabella 3.18: Tabella riportante i valori degli indici statistici forniti da *Geomagic Wrap* sul paziente D.

3.3.2 Allineamento mediante SolidWorks

Il secondo programma utilizzato per l'esecuzione della registrazione di modelli tridimensionali è *SolidWorks*. Questo software presenta delle similarità con il modulo di *3D Slicer* tra cui l'identificazione, tra i modelli importati nell'ambiente di lavoro, del cranio di riferimento e quello usato come test e l'inserimento, su ognuno di essi, di tre fiducial. L'allineamento effettuato è quindi basato sui landmarks, ma a differenza di *3D Slicer* il posizionamento viene attuato mediante l'ausilio di una mesh.

Il procedimento prevede l'inserimento, sul modello di riferimento, di tre punti (chiamati A,B,C), i quali vengono usati per creare un piano e una retta che congiunge due dei tre punti, A e B. Mentre sulla ricostruzione post operatoria vengono collocati, in posizione corrispondente, altri tre punti (A_1, B_1 e C_1). Infine, per effettuare la sovrapposizione, vengono stabiliti degli accoppiamenti di coincidenza tra le grandezze geometriche create; in particolare vengono imposte: la coincidenza tra i punti A e A_1 e l'appartenenza dei fiducial B_1 e C_1 rispettivamente alla retta AB e al piano.

Di seguito sono riportati i quattro pazienti studiati per i quali la convenzione di colori adottata è la stessa della Sottosezione 3.3.1. Anche in questo caso i modelli pre operatori sono stati considerati come riferimenti sia nell'ambiente di lavoro di *SolidWorks* sia in quello di *Geomagic Wrap*, mediante il quale sono stati valutati i risultati ottenuti nei vari allineamenti.

Paziente A

Le posizioni individuate per i marcatori, riportate nella Tabella 3.19, sono visibili nelle Figure 3.65 e 3.66.

<i>Fiducial</i>	<i>Posizione</i>
A e A_1	in corrispondenza della giunzione tra le ossa nasali e le due cartilagini nasali laterali
B e B_1	su una piccola protuberanza dell'osso temporale sinistro
C e C_1	su una sporgenza dell'osso zigomatico sinistro

Tabella 3.19: Tabella riportante l'indicazione della posizione dei marcatori sui modelli 3D, ottenuti dai referti pre e post operatori del paziente A.

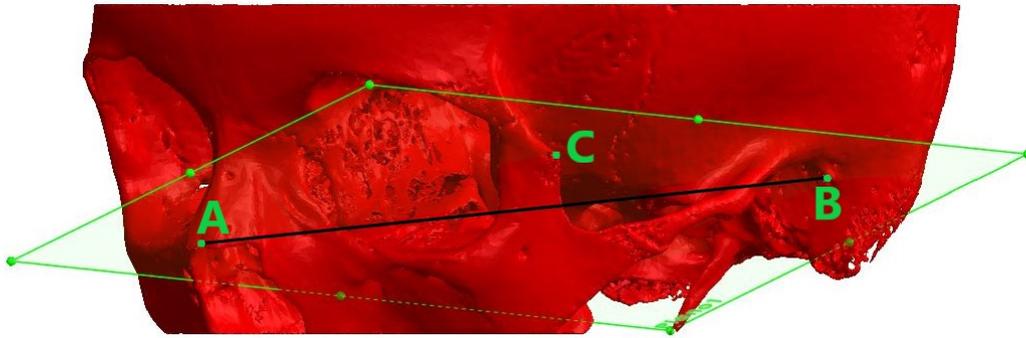


Figura 3.65: Modello pre operatorio del cranio del paziente A con indicati i fiducial posizionati, la retta e il piano creati.

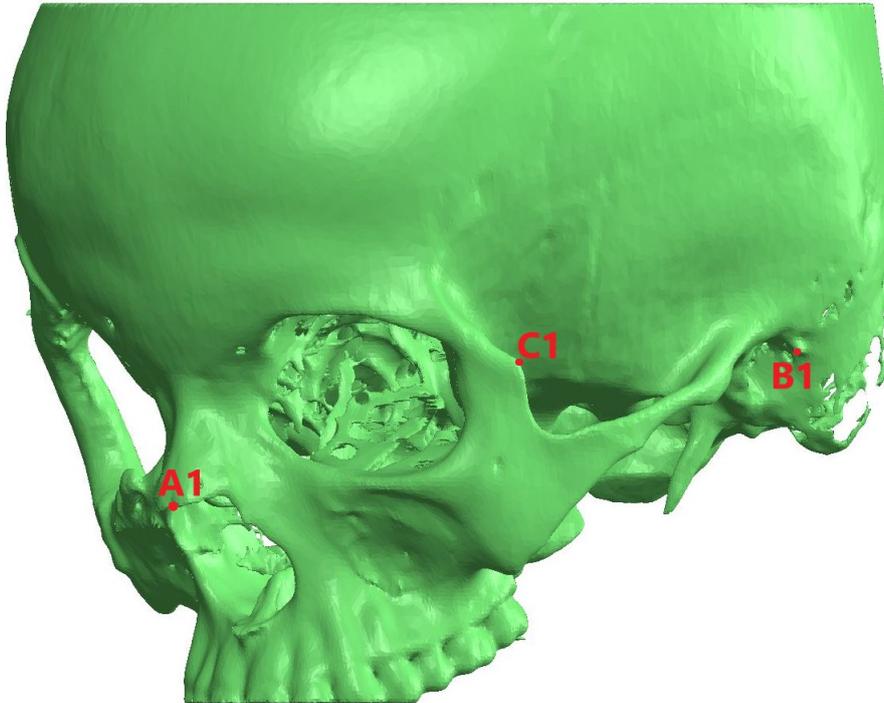


Figura 3.66: Ricostruzione tridimensionale post operatoria del cranio del paziente A con indicati i fiducial posizionati.

Imponendo gli accoppiamenti tra le grandezze geometriche create è stato ottenuto l'allineamento riportato in figura.

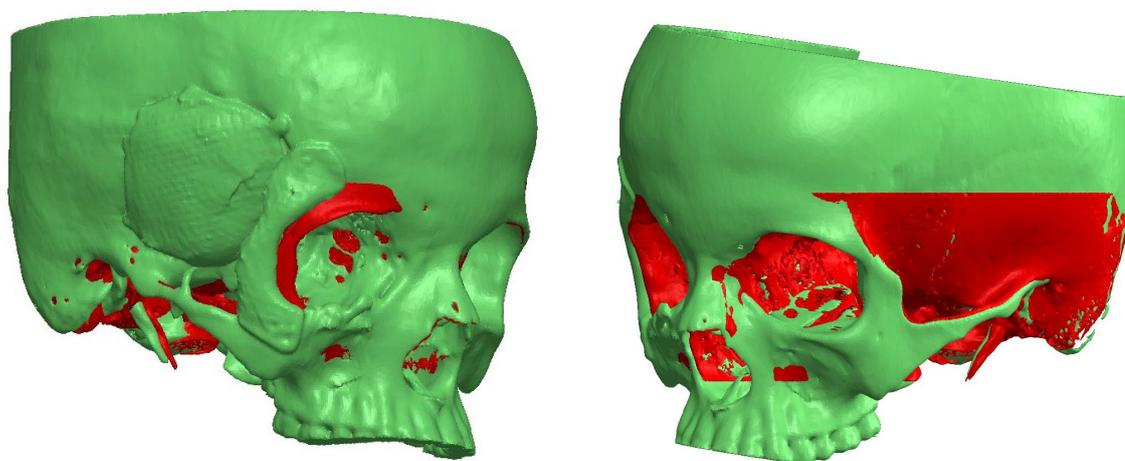


Figura 3.67: Sovrapposizione mediante *SolidWorks* delle ricostruzioni ricavate dai due referti del paziente A.

Il risultato è stato analizzato mediante *Geomagic Wrap* ottenendo una mappatura a colori della superficie. Le scelte effettuate per l'esecuzione dello studio, mantenute invariate per ogni paziente, sono le medesime adottate nella Sottosezione 3.3.1, ossia:

Scelta delle convenzioni sui volumi	
Volume di riferimento	modello pre operatorio
Volume di test	modello post operatorio

Tabella 3.20: Scelte attuate, riguardo il volume di riferimento e di test, per l'esecuzione dell'analisi mediante *Geomagic Wrap* e valide per tutti i pazienti.

Limiti (mm)	Valori (mm)
Critico max	2
Nominale min	0.25
Nominale min	-0.25
Critico min	-2

Tabella 3.21: Scelte attuate per la definizione dell'intervallo di tolleranza degli scostamenti, per l'esecuzione dell'analisi mediante *Geomagic Wrap* e valide per tutti i pazienti.

Raffigurazione della distanza mediante mappatura a colori

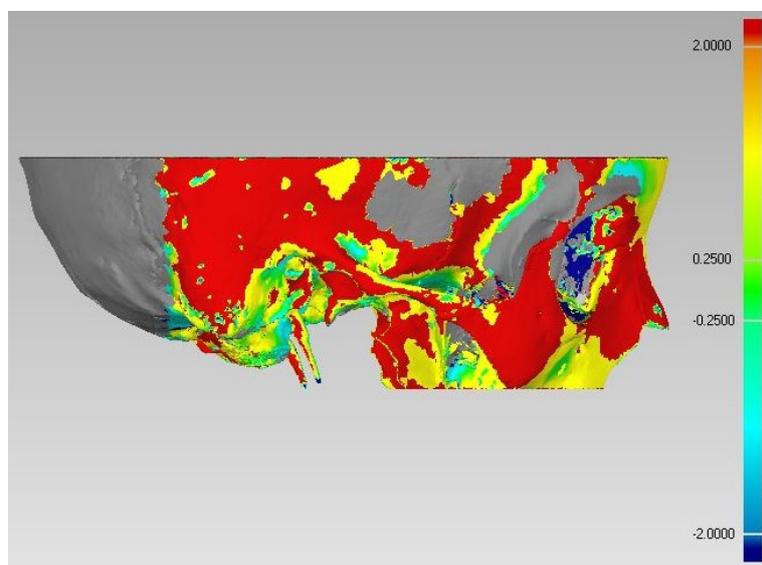


Figura 3.68: Vista laterale destra della mappa a colori della sovrapposizione dei crani del paziente A.

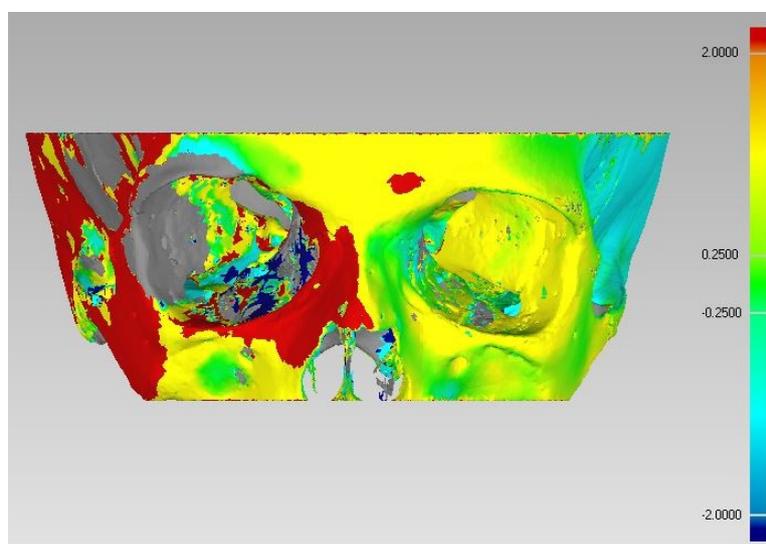


Figura 3.69: Vista frontale della mappa a colori della sovrapposizione dei crani del paziente A.

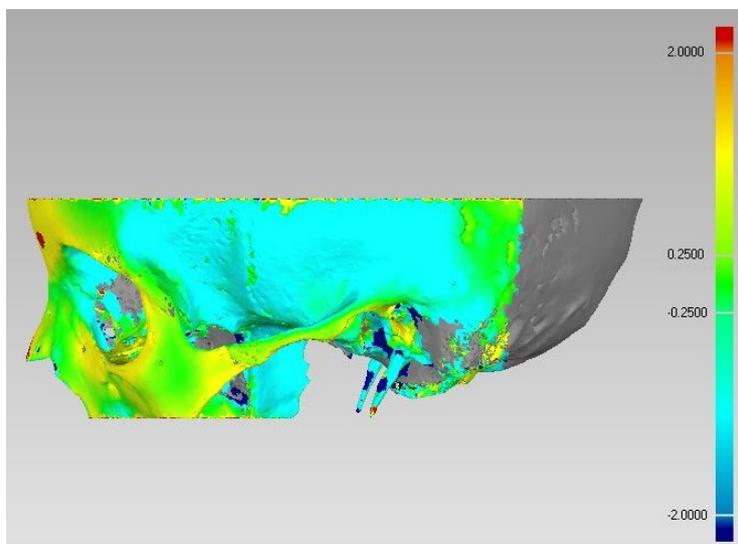


Figura 3.70: Vista laterale sinistra della mappa a colori della sovrapposizione dei crani del paziente A.

Le immagini mostrano un allineamento buono degli emisferi sinistri della testa e pessimo di quelli destri. Questo può essere ricondotto a diversi motivi tra cui la presenza nella parte destra delle aree sottoposte ad intervento chirurgico, la disposizione dei tre marcatori nella zona sinistra e l'incremento volumetrico, anche se minimo, del cranio poiché le TAC, eseguite a distanza di anni, appartengono ad un bambino.

Per poter effettuare in seguito un confronto con le registrazioni ottenute mediante i software *3D Slicer* e *ProPlan* vengono riportati nella Tabella gli ulteriori dati forniti dal software.

		VALORE (mm)
Distanza media		0.53
Deviazione standard		1.89
Distanza massima	Positiva	5
	Negativa	-5

Tabella 3.22: Tabella riportante i valori degli indici statistici forniti da *Geomagic Wrap* sul paziente A.

Paziente B

La sovrapposizione delle due ricostruzioni tridimensionali del paziente B,

riportata nella Figura 3.73, è ottenuta dall'inserimento dei landmarks sulla spina nasale anteriore (A e A_1) sulle suture fronto-zigomatiche destra (B e B_1) e sinistra (C e C_1) in corrispondenza del margine orbitario, come mostrato nelle Figure 3.71 e 3.72.

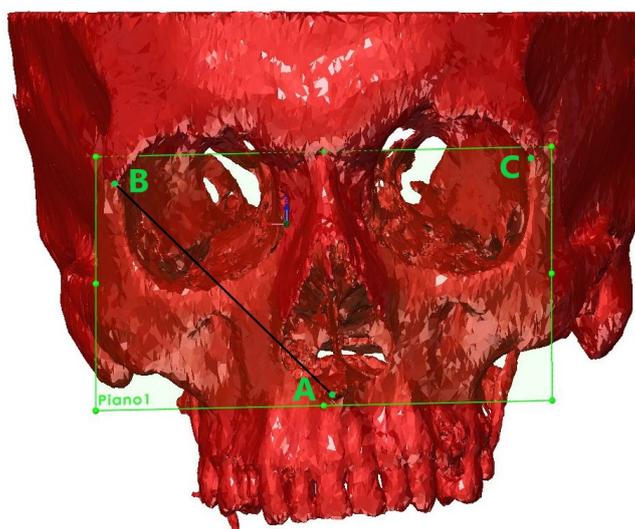


Figura 3.71: Modello pre operatorio del paziente B con indicati i marcatori inseriti, la retta e il piano generati.

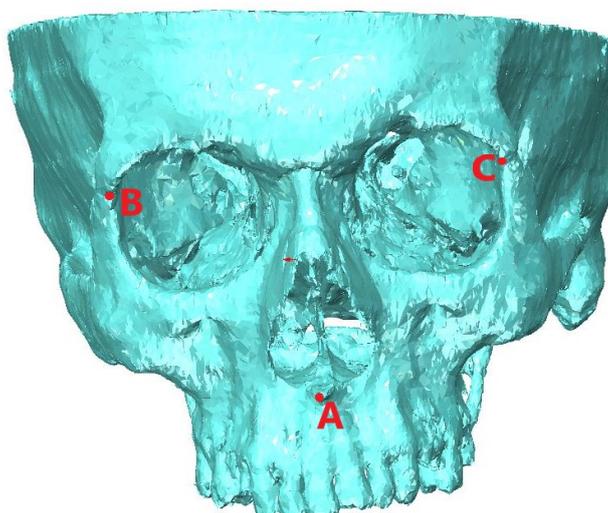


Figura 3.72: Modello post operatorio del paziente B con indicati i marcatori inseriti.

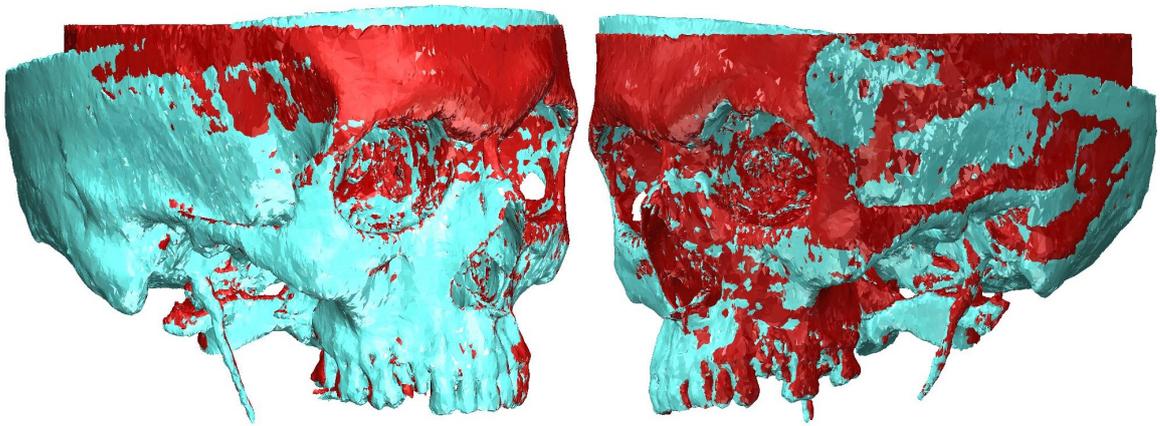


Figura 3.73: Immagine raffigurante la sovrapposizione, ottenuta mediante *SolidWorks*, delle ricostruzioni tridimensionali pre e post operatorie del cranio del paziente B.

L'alternanza di colori preannuncia un ottimo allineamento, affermazione che può essere verificata facilmente mediante un'analisi dello scostamento presente tra le superfici dei due crani.

Rappresentazione degli scostamenti mediante mappatura a colori

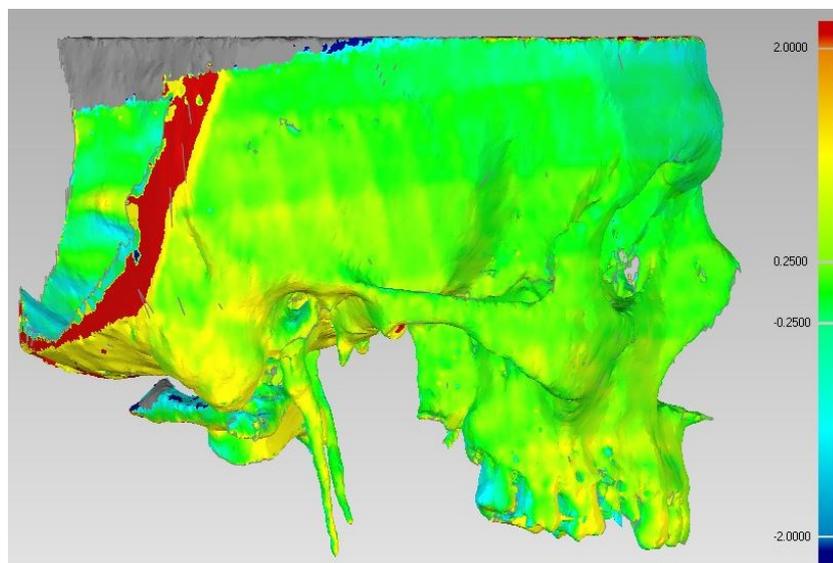


Figura 3.74: Vista laterale destra della mappa a colori della sovrapposizione dei crani del paziente B.

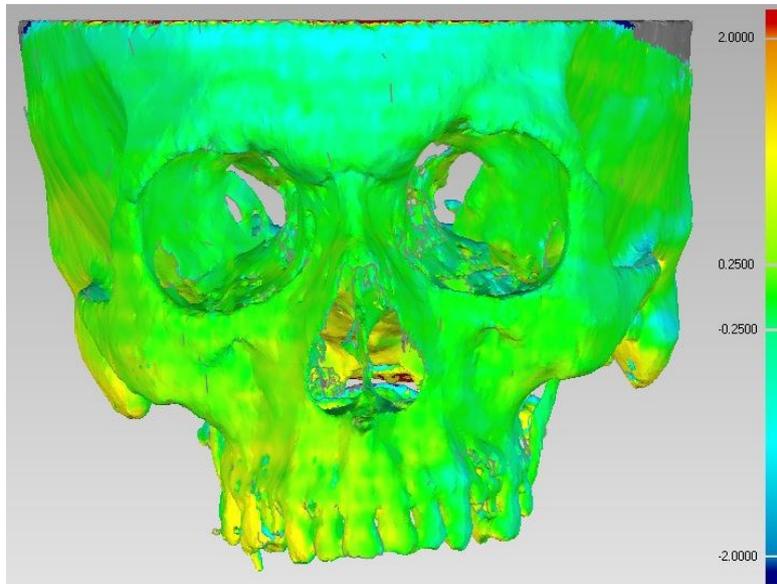


Figura 3.75: Vista frontale della mappa a colori della sovrapposizione dei crani del paziente B.

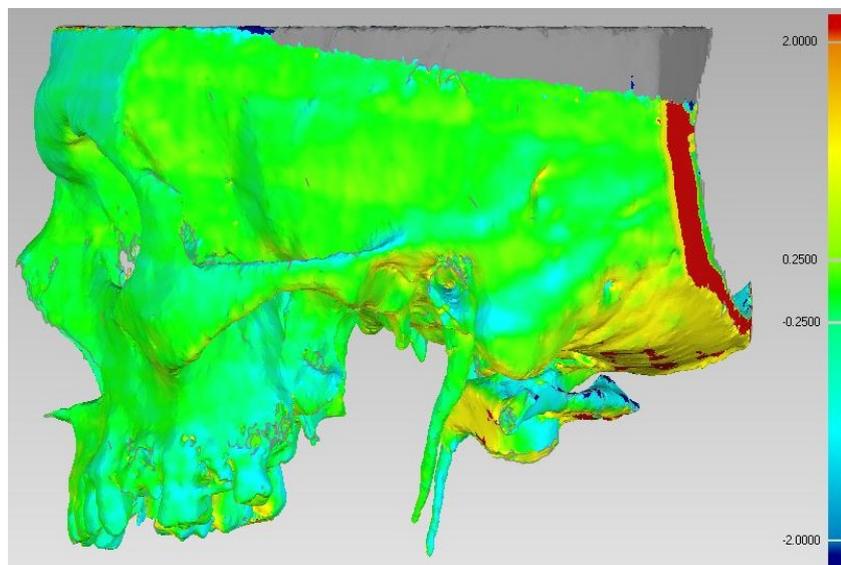


Figura 3.76: Vista laterale sinistra della mappa a colori della sovrapposizione dei crani del paziente B.

L'analisi effettuata mostra come i landmarks scelti siano stati posizionati correttamente, infatti la maggior parte dei punti della superficie Template si

trova a meno di 0.25 mm dalla superficie riferimento. Gli indici più significativi della buona riuscita della registrazione sono la distanza media e la deviazione standard, riportati nella Tabella 3.23.

		VALORE (mm)
Distanza media		0.10
Deviazione standard		0.93
Distanza massima	Positiva	6
	Negativa	-6

Tabella 3.23: Tabella riportante i valori degli indici statistici forniti da *Geomagic Wrap* sul paziente B.

Paziente C

Per il paziente C i punti anatomici individuati per il posizionamento dei marcatori sono la spina nasale anteriore (A e A_1) e i punti sottorbitali destro (B e B_1) e sinistro (C e C_1), come mostrato nelle immagini.

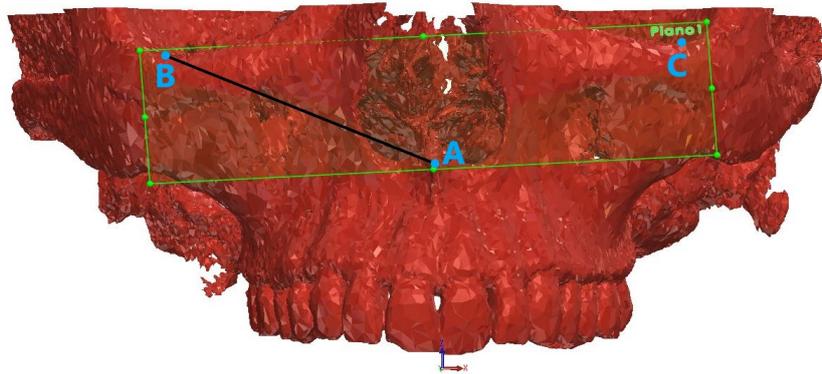


Figura 3.77: Ricostruzione tridimensionale ricavata dalla TAC pre operatoria del paziente C con indicati i fiducial inseriti, la retta e il piano generati.

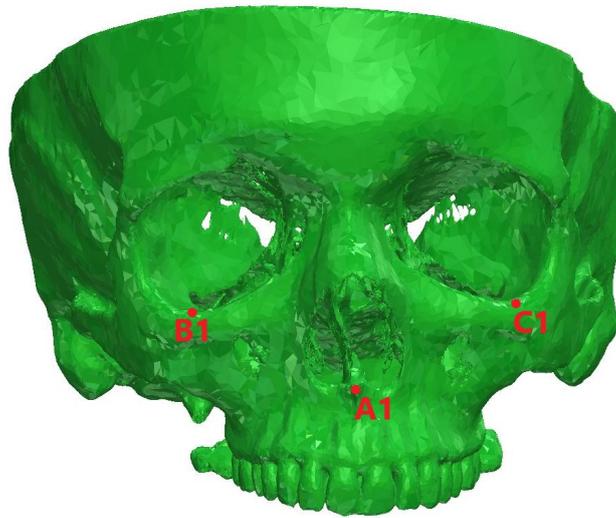


Figura 3.78: Ricostruzione tridimensionale ricavata dalla TAC post operatoria del paziente C con indicati i fiducial posizionati.

L'allineamento ottenuto è riportato nella Figura 3.79. Benchè esso non possa definirsi perfetto, non risultano, dall'analisi degli scostamenti riportata di seguito, grandi aree anatomiche fuori range di tolleranza.

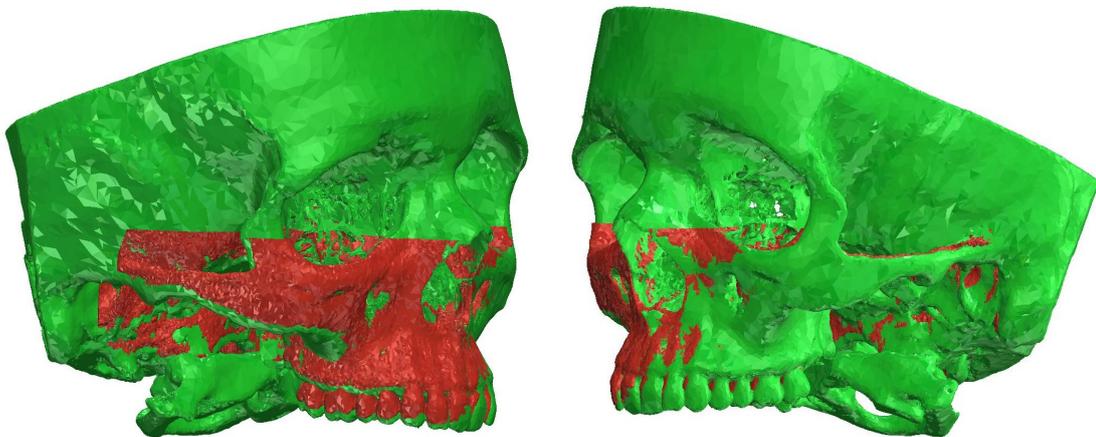


Figura 3.79: Sovrapposizione mediante *SolidWorks* delle ricostruzioni ricavate dai due referti del paziente C.

Raffigurazione delle distanze mediante mappatura a colori

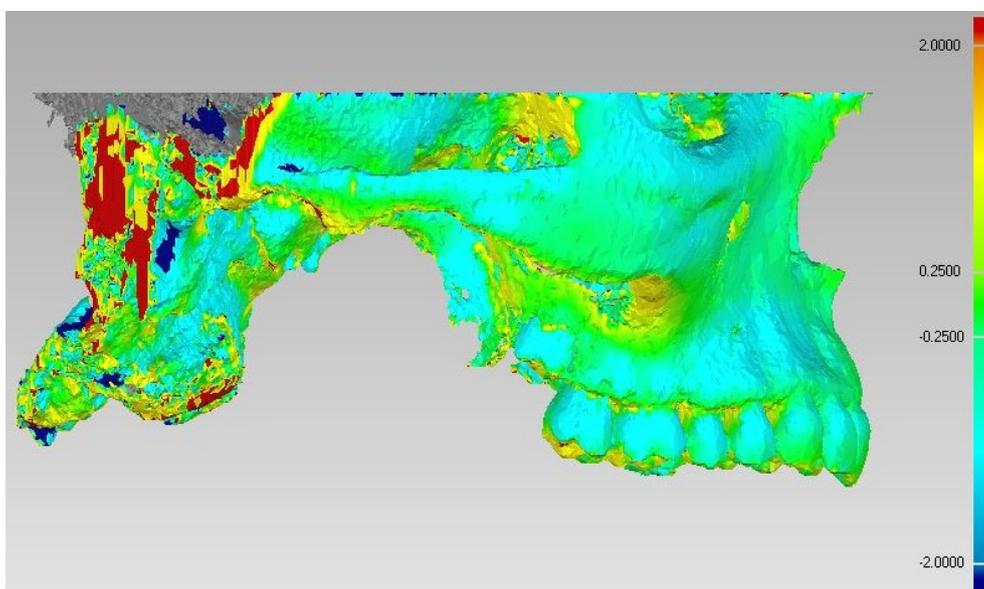


Figura 3.80: Vista laterale destra della mappa a colori dell'allineamento dei modelli del paziente C.

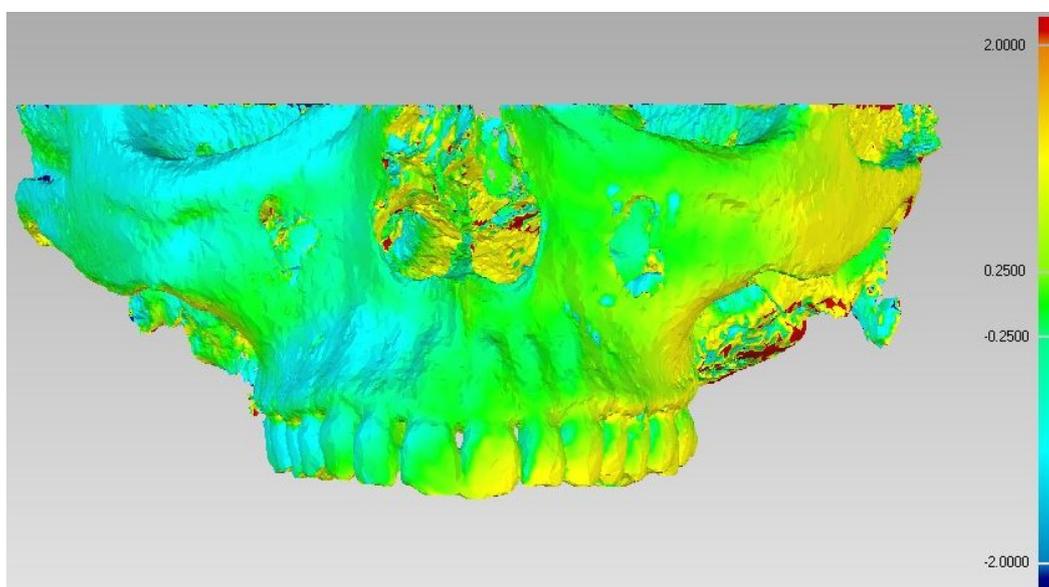


Figura 3.81: Vista frontale della mappa a colori dell'allineamento dei modelli del paziente C.

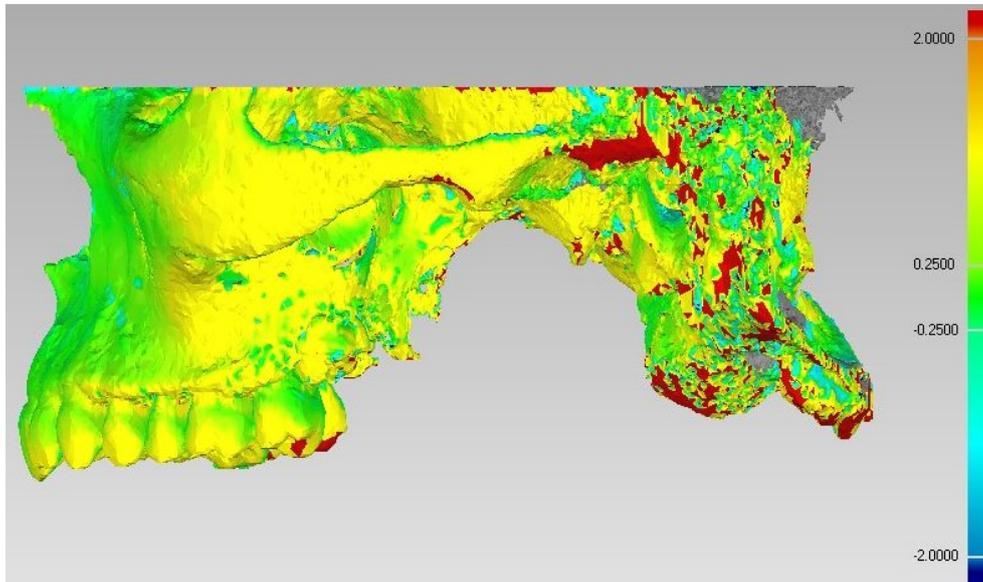


Figura 3.82: Vista laterale sinistra della mappa a colori dell'allineamento dei modelli del paziente C.

La registrazione ottenuta può essere considerata buona. Le zone a cui corrisponde una distanza $|d| \geq 2$ sono piccole e connesse al differente volume rilevato nei due esami, esse risultano infatti localizzate lungo i contorni della ricostruzione.

		VALORE (mm)
Distanza media		0.41
Deviazione standard		1.85
Distanza massima	Positiva	6
	Negativa	-6

Tabella 3.24: ITabella riportante i valori degli indici statistici forniti da *Geomagic Wrap* sul paziente C.

Paziente D

Come era già stato osservato per la registrazione effettuata con *3D Slicer*, nel caso del paziente D, per l'individuazione dei landmarks devono essere escluse le ossa zigomatiche e parte della mascella. Questo ha portato all'identificazione dei seguenti fiducial: sulla spina nasale anteriore (C e C_1) e nel foro sopraorbitale destro (A e A_1) e sinistro (B e B_1).

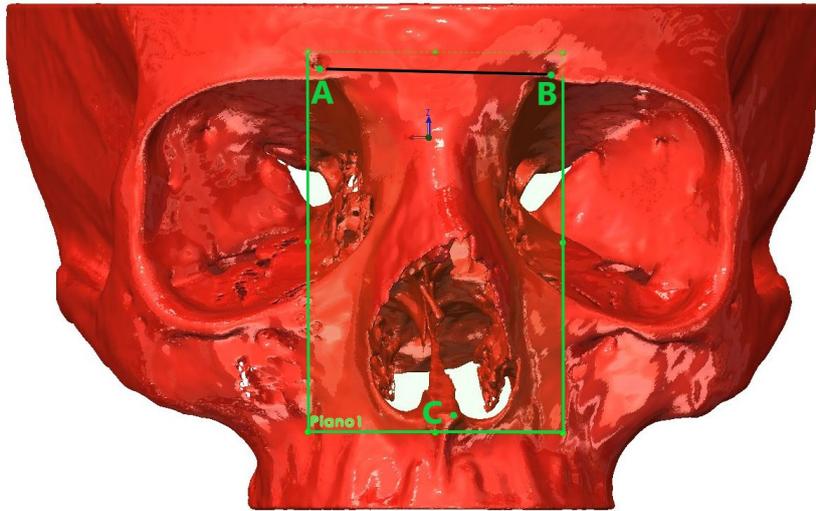


Figura 3.83: Ricostruzione tridimensionale ricavata dalla TAC pre operatoria del paziente D con indicati i fiducial inseriti, la retta e il piano generati.

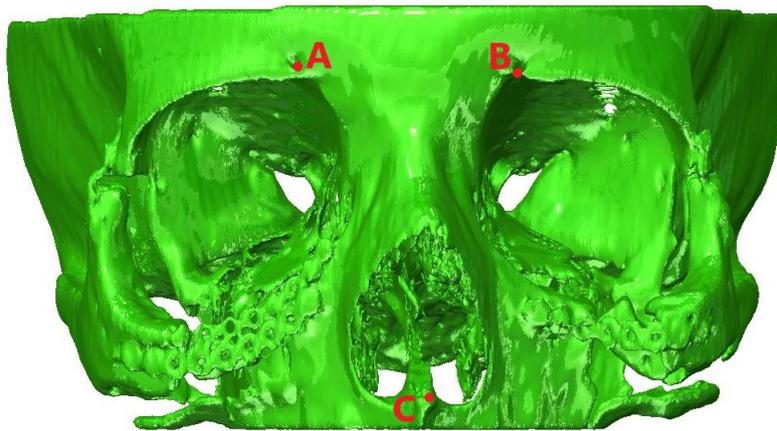


Figura 3.84: Ricostruzione tridimensionale ricavata dalla TAC post operatoria del paziente D con indicati i fiducial inseriti.

Dalla sovrapposizione ricavata e riportata nella Figura 3.85 è possibile ipotizzare che lo studio dello scostamento tra i due crani mostrerà: una colorazione verde del massiccio facciale, con la sola esclusione delle ossa zigomatiche che appariranno rosse.

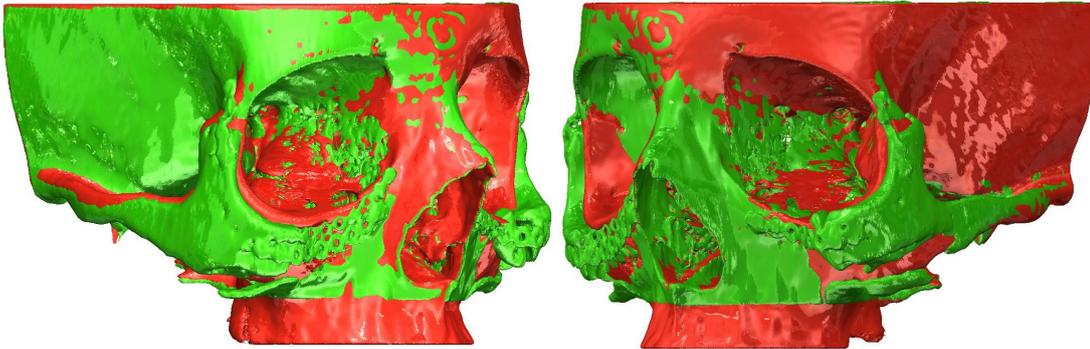


Figura 3.85: Sovrapposizione mediante *SolidWorks* delle ricostruzioni ricavate dai due referti del paziente D.

Raffigurazione degli scostamenti mediante mappatura a colori

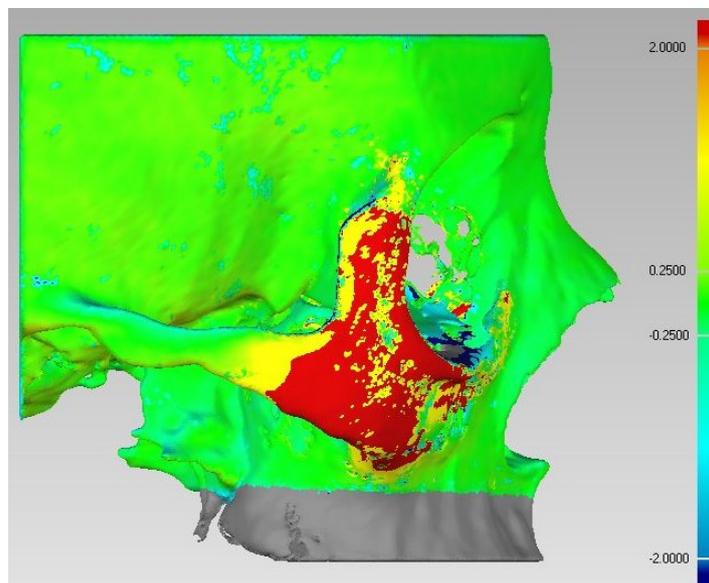


Figura 3.86: Vista laterale destra della mappa a colori dell'allineamento dei modelli del paziente D.

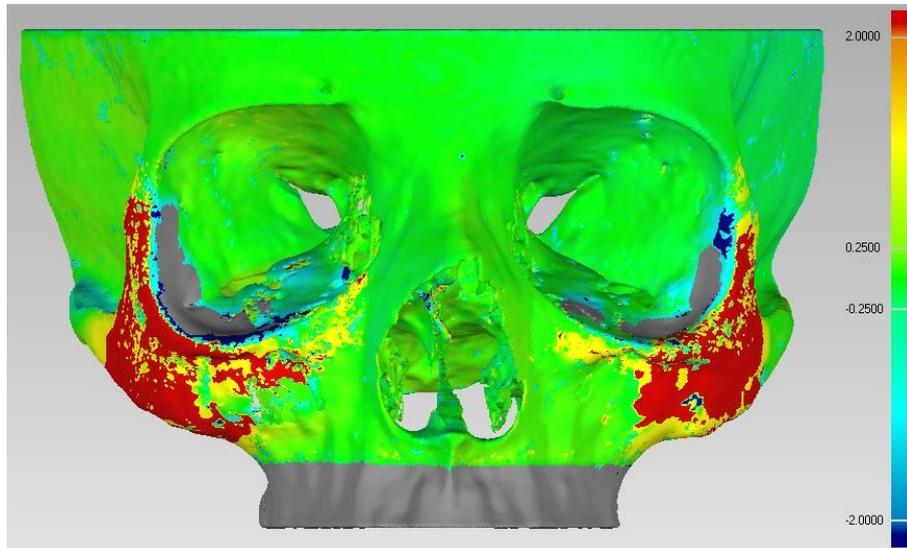


Figura 3.87: Vista frontale della mappa a colori dell'allineamento dei modelli del paziente D.

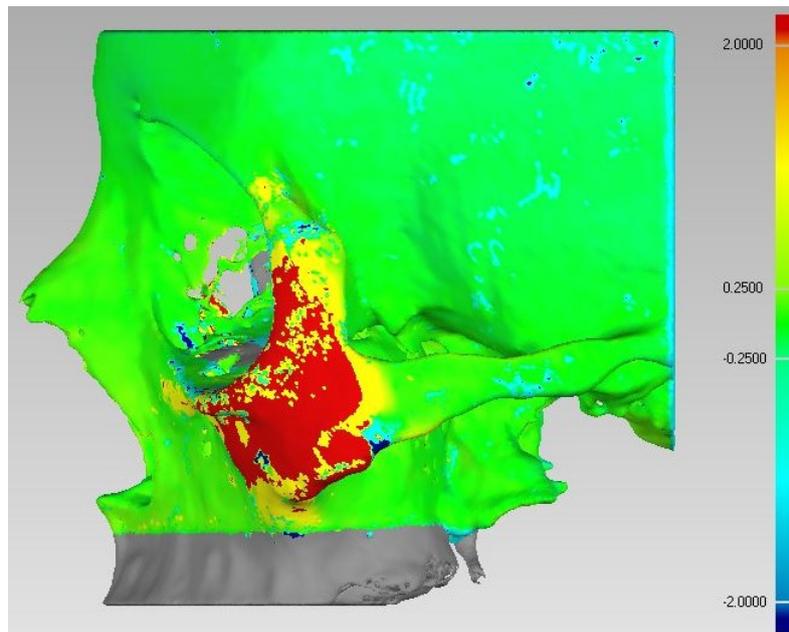


Figura 3.88: Vista laterale sinistra della mappa a colori dell'allineamento dei modelli del paziente D.

Dall'osservazione delle immagini riportate sopra, è possibile notare una leggera colorazione azzurra, a sinistra, e gialla, a destra, delle superfici laterali

ad indicare che la distanza tra i due crani, in queste regioni, è prossima a 0.25 mm.

		VALORE (mm)
Distanza media		-0.03
Deviazione standard		1.18
Distanza massima	Positiva	6
	Negativa	-6

Tabella 3.25: Tabella riportante i valori degli indici statistici forniti da *Geomagic Wrap* sul paziente D.

3.3.3 Registrazione mediante ProPlan

"ProPlan è usato come software di interfaccia e come sistema di segmentazione di immagini ottenute mediante TAC o MRI. E' inoltre adoperato per la pianificazione delle operazioni, simulando e valutando la posizione degli impianti e le varie opzioni di trattamento chirurgico".[18]

L'iter seguito per effettuare la registrazione prevede l'evidenziazione delle superfici dei due crani, escludendo le aree modificate dall'intervento e quelle che presentano delle anomalie come fori dovuti ad una cattiva segmentazione. Il processo di segmentazione permette di ricavare il modello tridimensionale a partire dalle viste bidimensionali della TAC, impostando un range di intensità dei pixel.

Il tempo richiesto e la precisione che deve essere adottata dall'utente nella selezione sono decisamente inferiori rispetto a quelli necessari nel caso degli altri due software testati.

Per ogni paziente sono riportate, di seguito, le sovrapposizioni dei modelli, per i quali è stata mantenuta la convenzione di colori usata in precedenza, e l'analisi dei rispettivi scostamenti.

Paziente A

Una volta selezionata la superficie viene effettuata la sovrapposizione dando il comando *Allinea*, il risultato, ricavato per il paziente A, è visibile nella Figura 3.89.

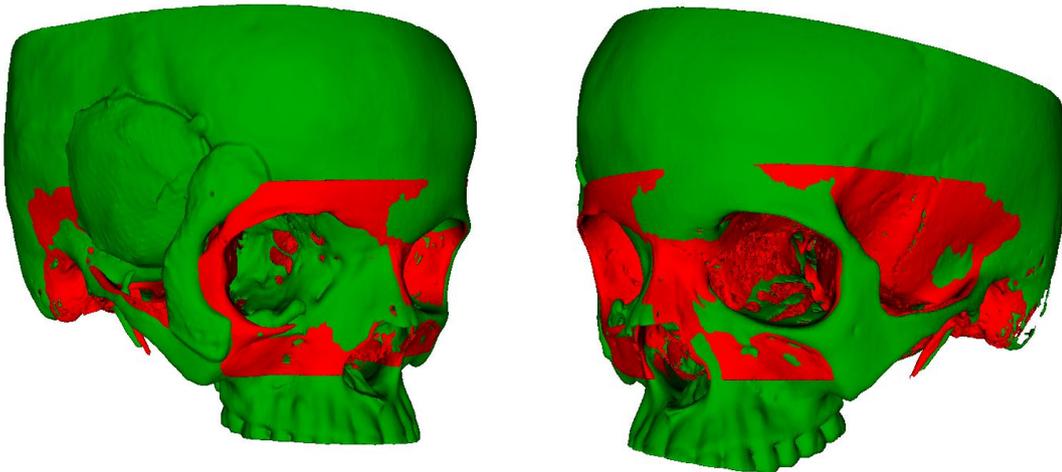


Figura 3.89: Sovrapposizione mediante ProPlan delle ricostruzioni ricavate dai referti del paziente A.

La valutazione della distanza, tra i punti della mesh del modello di riferimento e la superficie del cranio usato come test, è determinata sempre mediante *Geomagic Wrap*, per il quale sono mantenute invariate sia le scelte effettuate riguardo i valori che definiscono gli intervalli di accettazione degli scostamenti sia le convenzioni sui volumi di riferimento e Template. Il risultato dell'analisi eseguita è visibile nelle immagini sottostanti.

Raffigurazione delle distanze mediante mappatura a colori

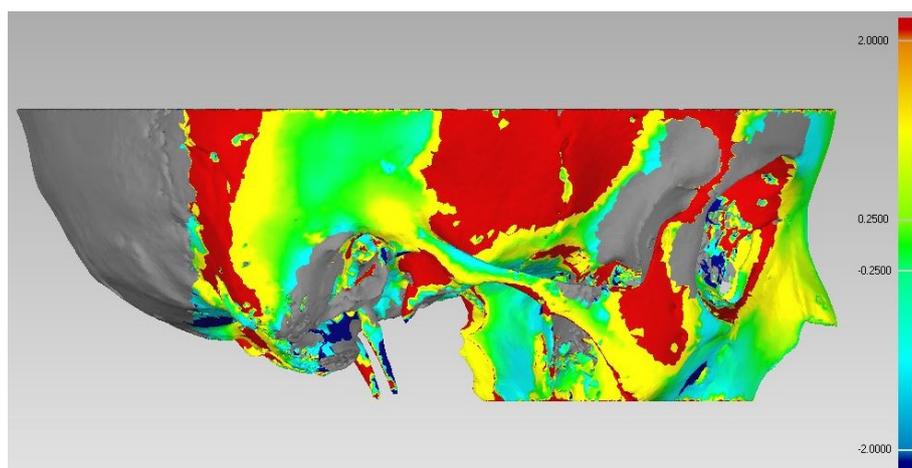


Figura 3.90: Vista laterale destra della mappa a colori dell'allineamento, ottenuta mediante *ProPlan*, dei modelli del paziente A.

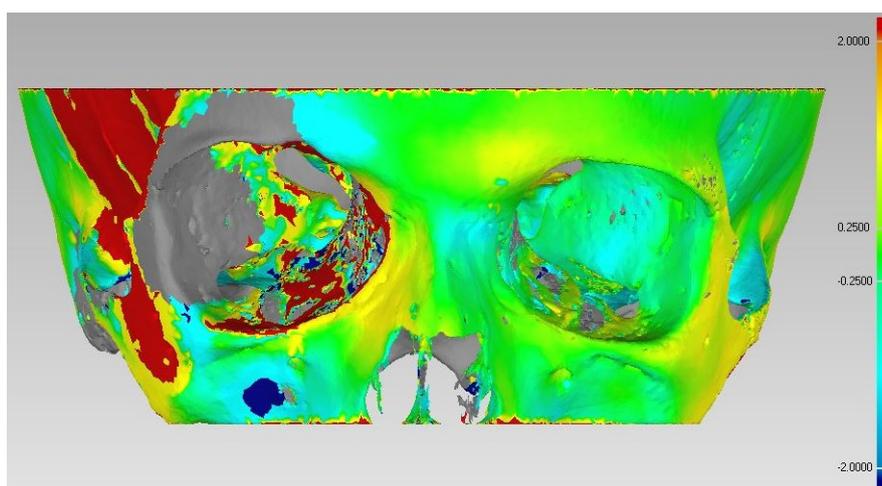


Figura 3.91: Vista frontale della mappa a colori della sovrapposizione, ottenuta mediante *ProPlan*, dei modelli del paziente A.

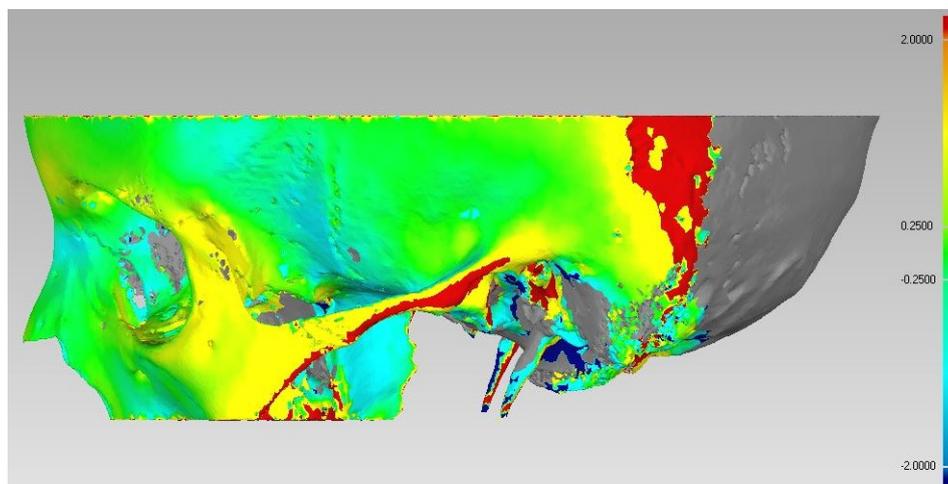


Figura 3.92: Vista laterale sinistra della mappa a colori dell'allineamento, ottenuto mediante *ProPlan*, dei modelli del paziente A.

In rosso sono colorate le aree fuori tolleranza, esse sono situate soprattutto nell'emisfero destro del cranio a causa dell'intervento effettuato, mentre nella restante parte del cranio è possibile osservare una prevalenza del colore verde, e la presenza di alcune zone gialle localizzate in corrispondenza dell'osso zigomatico e di quello temporale. Le estese aree fuori tolleranza, visibili nella Figura 3.90, sono la causa del valore elevato assunto dal parametro *distanza media* riportato nella Tabella 3.26, insieme ad altri indici statistici ricavati dal software.

		VALORE (mm)
Distanza media		0.58
Deviazione standard		1.91
Distanza massima	Positiva	6
	Negativa	-6

Tabella 3.26: Tabella riportante i valori degli indici statistici forniti da *Geomagic Wrap* sul paziente A.

Paziente B

Per questo paziente la selezione risulta estremamente semplice, infatti, dell'intero volume a disposizione deve essere esclusa solo la mandibola, che è stata sottoposta ad intervento. La Figura 3.93 mostra un allineamento perfetto dei crani, caratterizzato da un'alternanza dei colori, in questo caso rosso e azzurro.

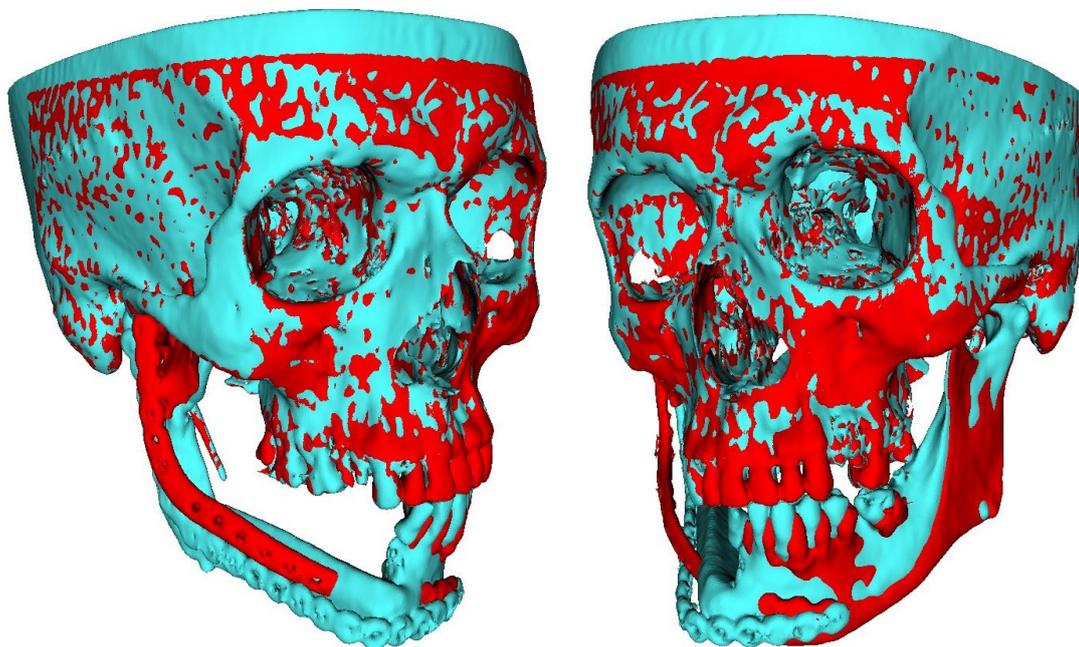


Figura 3.93: Immagine della sovrapposizione, ricavata mediante *ProPlan*, dei modelli pre e post operatori del cranio del paziente B.

Per la valutazione della bontà dell'allineamento non devono essere considerate le ossa mandibolari poiché la loro posizione risulta indipendente da quella delle altre ossa del cranio. Per evitare che la presenza di questa parte anatomica falsi i risultati ottenuti mediante *Geomagic Wrap*, essa è stata esclusa dai modelli importati nell'ambiente di lavori del software.

Raffigurazione degli scostamenti mediante mappatura a colori

E' evidente dall'osservazione delle immagini come la registrazione sia perfetta, infatti più del 90% dei punti presentano uno scostamento inferiore agli 0.25 mm .

Le zone fuori tolleranza $|d| > 2\text{ mm}$ sono invece situate in corrispondenza dei contorni delle ricostruzioni tridimensionali.

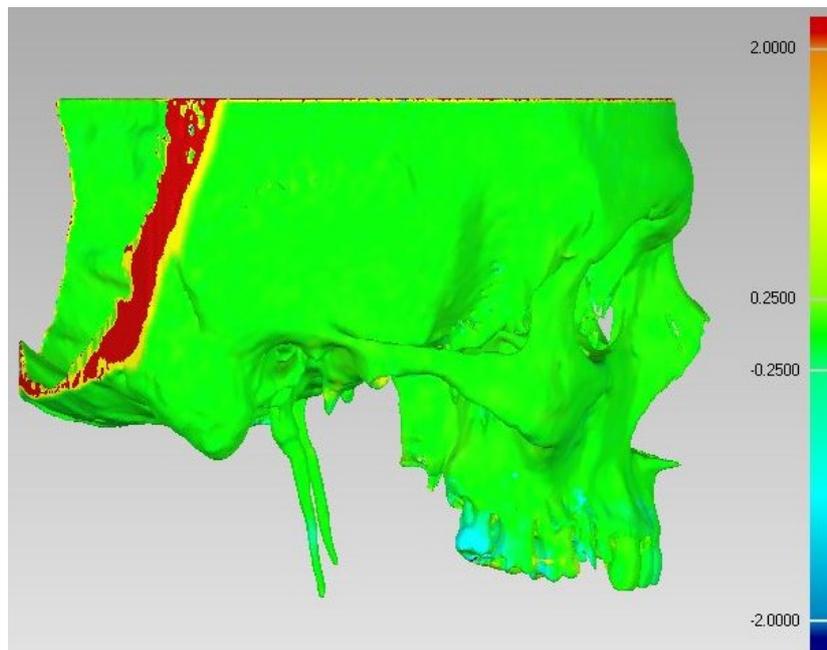


Figura 3.94: Vista laterale destra della mappa a colori dell'allineamento ottenuto mediante *ProPlan* dei modelli del paziente B.

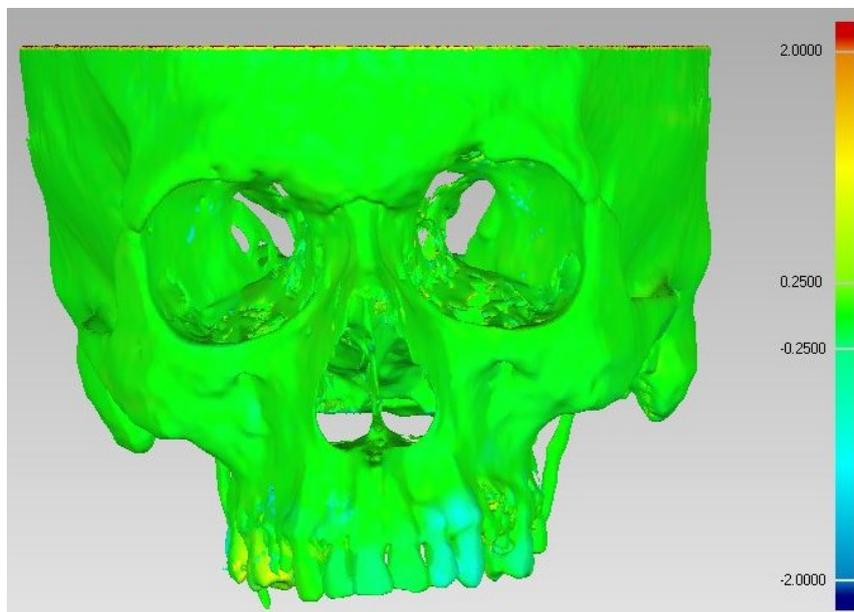


Figura 3.95: Vista frontale della mappa a colori dell'allineamento ottenuto mediante *ProPlan* dei modelli del paziente B.

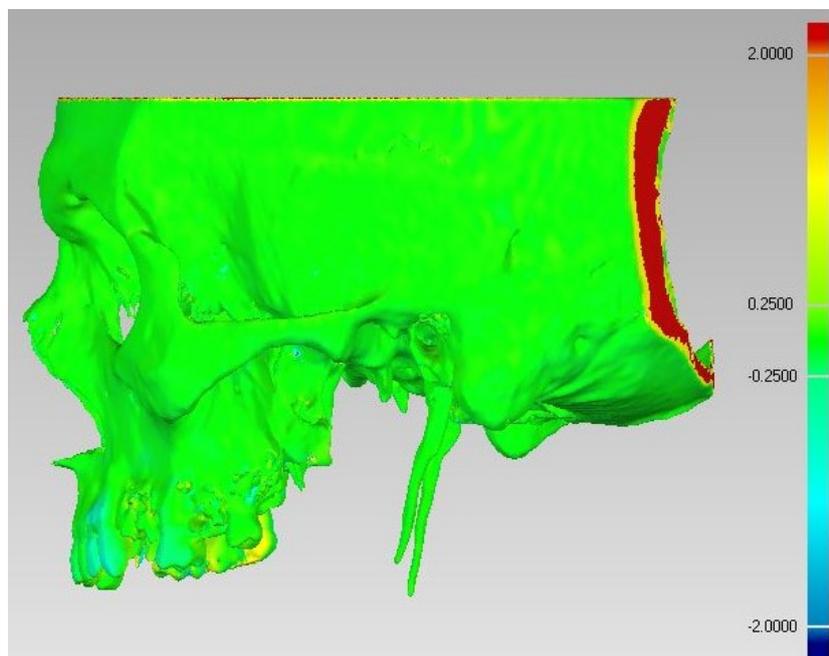


Figura 3.96: Vista laterale sinistra della mappa a colori dell'allineamento ottenuto mediante *ProPlan* dei modelli del paziente B.

		VALORE (mm)
Distanza media		0.20
Deviazione standard		0.81
Distanza massima	Positiva	6
	Negativa	-6

Tabella 3.27: Tabella riportante i valori degli indici statistici forniti da *Geomagic Wrap* sul paziente B.

Paziente C

Le immagini raffiguranti l'allineamento (fig.3.97) e la sua valutazione (fig.3.98, fig.3.99, fig.3.100) mostrano come la sola selezione della mascella (unica parte evidenziabile) permetta di ottenere un allineamento perfetto.

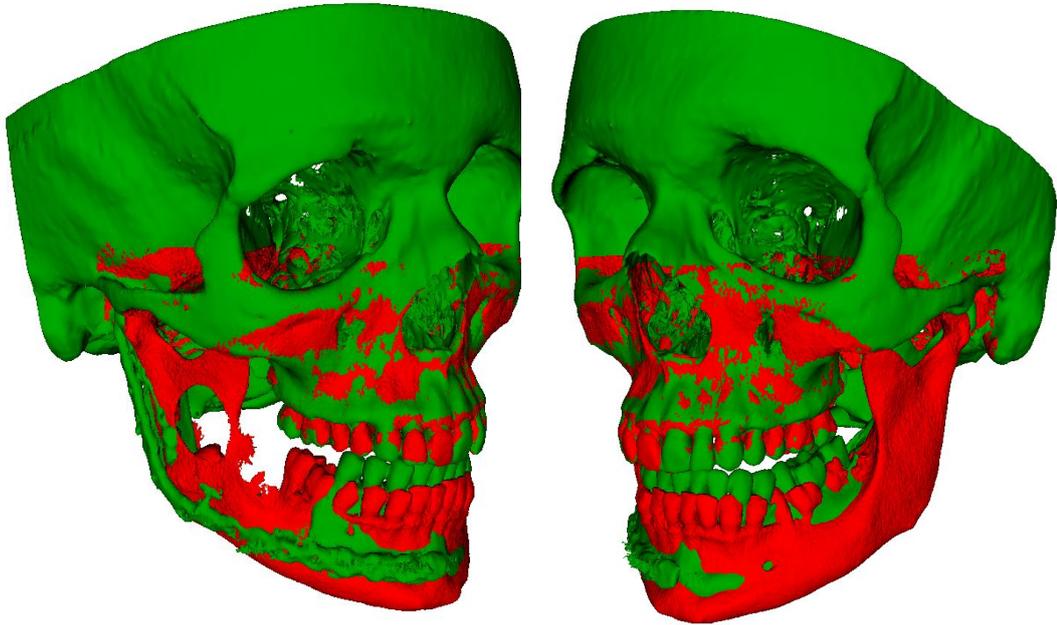


Figura 3.97: Risultato della sovrapposizione eseguita, per il paziente C.

Raffigurazione delle distanze mediante mappatura a colori

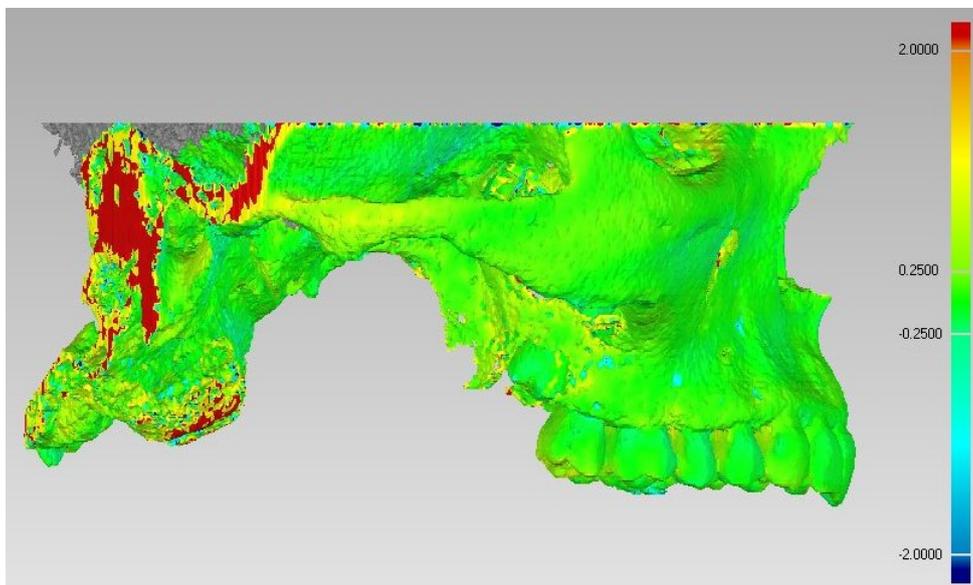


Figura 3.98: Vista laterale destra della mappa a colori dell'allineamento ottenuto mediante *ProPlan* dei modelli del paziente C.

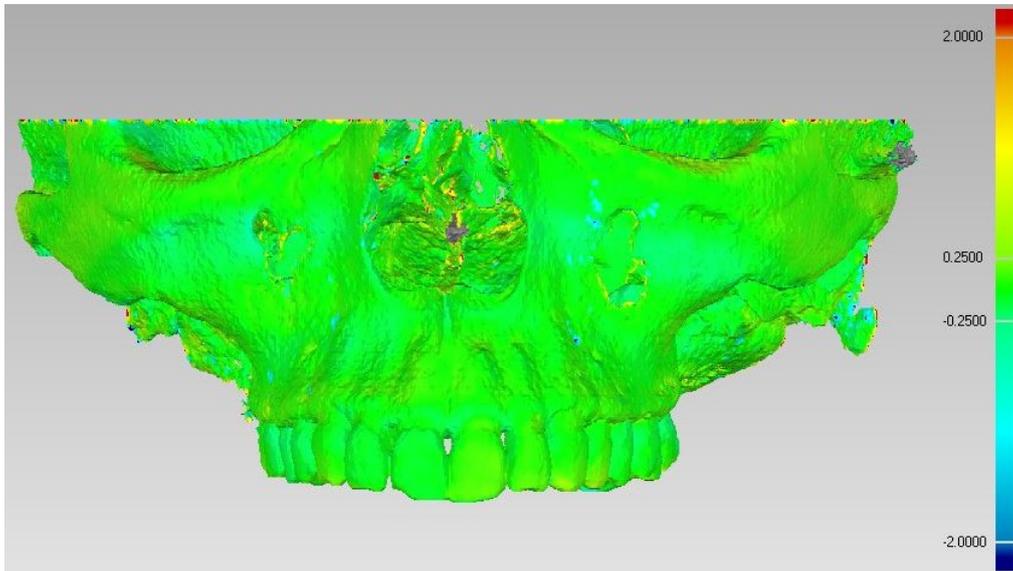


Figura 3.99: Vista frontale della mappa a colori dell'allineamento ottenuto mediante *ProPlan* dei modelli del paziente C.

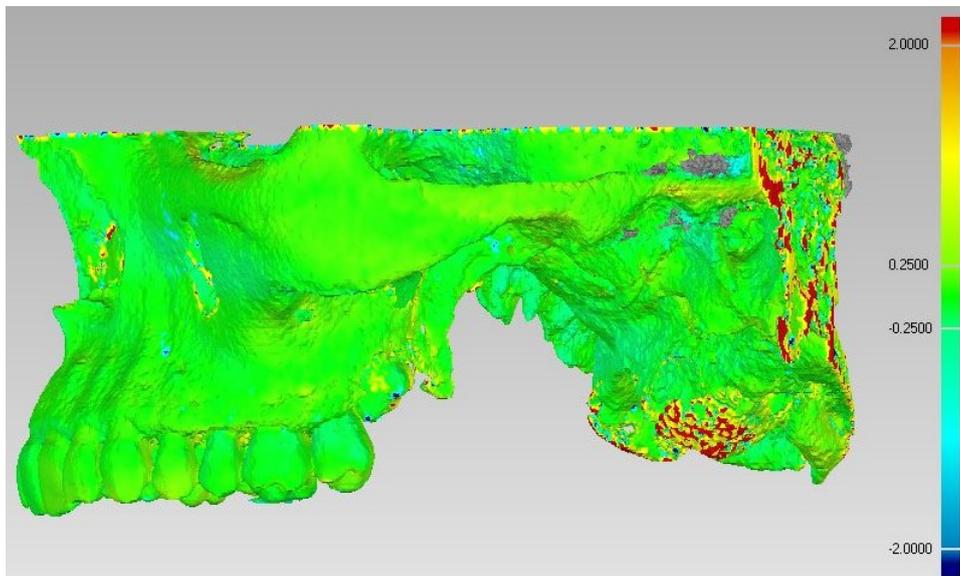


Figura 3.100: Vista laterale sinistra della mappa a colori dell'allineamento ottenuto mediante *ProPlan* dei modelli del paziente C.

		VALORE (mm)
Distanza media		0.5
Deviazione standard		1.77
Distanza massima	Positiva	6.3
	Negativa	-6.3

Tabella 3.28: Tabella riportante i valori degli indici statistici forniti da *Geomagic Wrap* sul paziente C.

Paziente D

Lo stesso procedimento è stato seguito anche per il quarto e ultimo paziente, ricavando il seguente risultato.

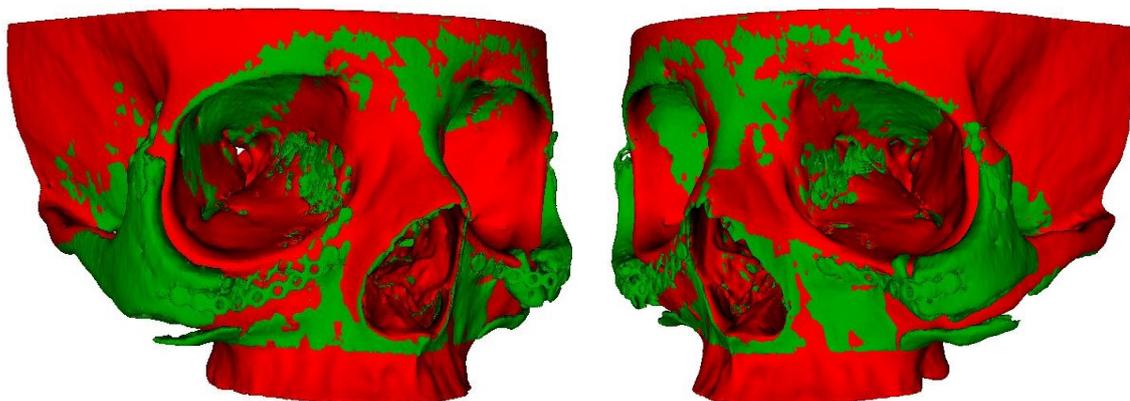


Figura 3.101: Risultato della sovrapposizione eseguita, per il paziente D.

Nella valutazione dello studio effettuato con *Geomagic Wrap* è necessario ricordare che le regioni modificate dall'intervento sono le ossa zigomatiche e parte della mascella.

Raffigurazione delle distanze mediante mappatura a colori

La sovrapposizione ottenuta può essere considerata solamente buona poichè, pur escludendo le aree interessate dall'operazione, essa presenta zone fuori tolleranza situate prevalentemente sull'osso frontale e sulle ossa sfenoidi e temporali.

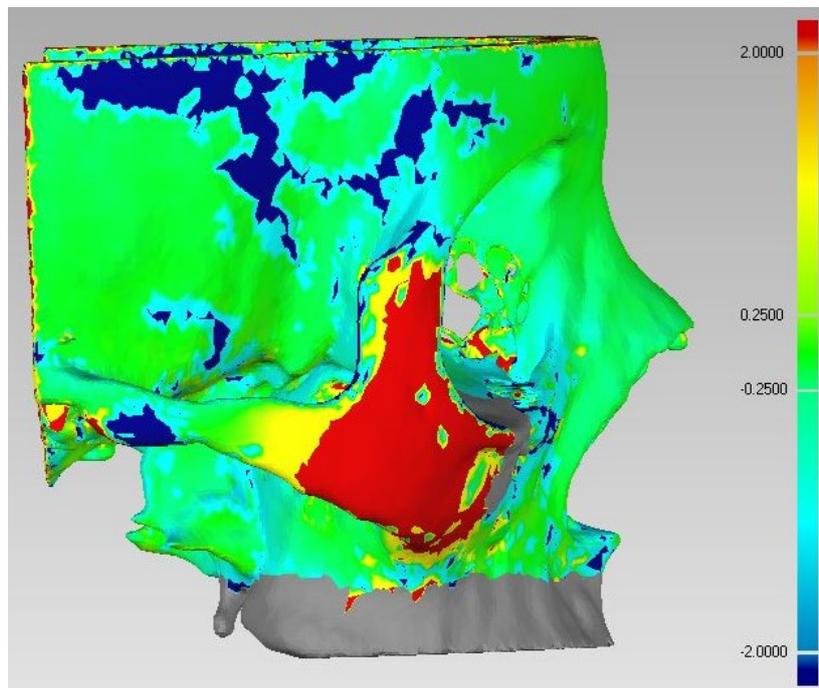


Figura 3.102: Vista laterale destra della mappa a colori dell'allineamento ottenuto mediante *ProPlan* dei modelli del paziente D.

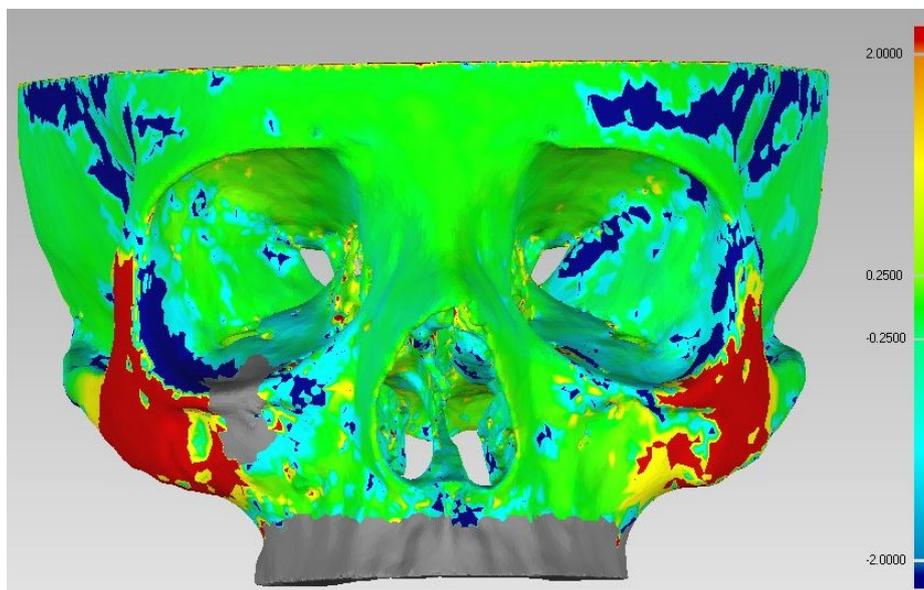


Figura 3.103: Vista frontale della mappa a colori dell'allineamento ottenuto mediante *ProPlan* dei modelli del paziente D.

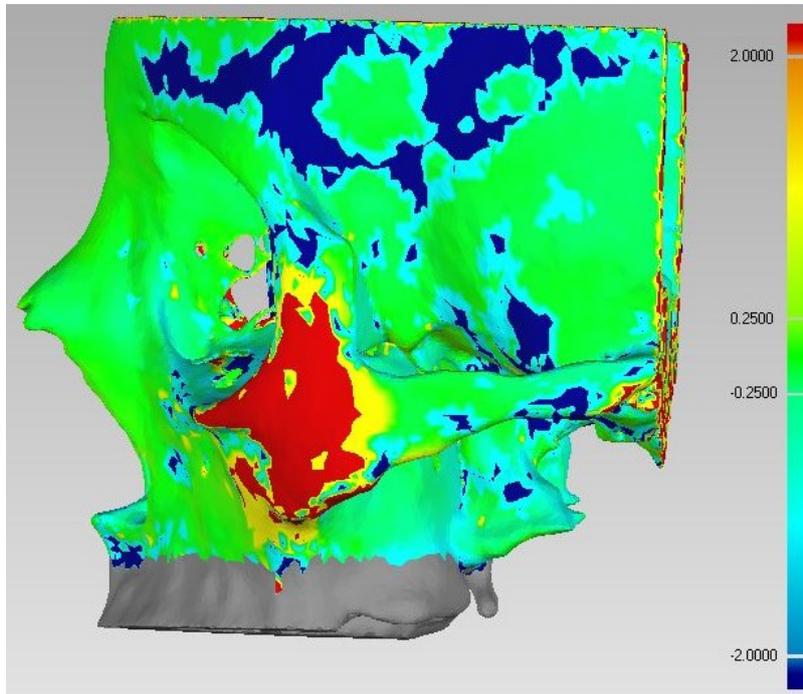


Figura 3.104: Vista laterale sinistra della mappa a colori dell'allineamento ottenuto mediante *ProPlan* dei modelli del paziente D.

		VALORE (mm)
Distanza media		-0.59
Deviazione standard		1.65
Distanza massima	Positiva	6
	Negativa	-6

Tabella 3.29: Tabella riportante i valori degli indici statistici forniti da *Geomagic Wrap* sul paziente D.

3.3.4 Conclusioni

I software disponibili per l'esecuzione della registrazione sono molteplici, essi possono permettere il confronto delle sole immagini mediche digitali, delle sole ricostruzioni tridimensionali oppure di entrambe. Gli algoritmi su cui si basano questi programmi sono funzione del metodo di registrazione adottato e delle dimensioni spaziali (2D-3D) degli elementi confrontati. Nell'elaborato sono stati confrontati sia differenti metodi di sovrapposizione dei tagli delle TAC pre e post operatorie, sia diversi metodi di registrazione ottenuti mediante l'uso di tre software, dei quali sono stati valutati pregi e difetti.

Tra i vantaggi di *ProPlan* si annoverano la semplicità del metodo di registrazione, basato sull'evidenziazione delle superfici, la quale richiede una bassa precisione e tempi ridotti, e la possibilità di poter visualizzare contemporaneamente il modello tridimensionale e i tagli della TAC, conservando in questo modo le informazioni derivanti da una analisi di queste ultime. Il principale svantaggio risiede invece nel costo elevato della licenza.

3D Slicer, come *ProPlan*, permette di visionare sia la ricostruzione tridimensionale sia quella bidimensionale dei referti acquisiti, ma il metodo di registrazione basato sui landmarks richiede un tempo e una precisione maggiore da parte dell'utente. Tra i vantaggi rientra senza dubbio il fatto di essere un software open-source.

SolidWorks è il secondo metodo utilizzato basato sul posizionamento dei fiducial. La disposizione di questi ultimi viene facilitata dall'utilizzo di una mesh. Tuttavia questo software, permettendo di visionare solo i modelli tridimensionali, non conservare le informazioni derivanti dall'osservazione delle immagini bidimensionali del referto.

Una caratteristica importante, che accomuna tutti e tre i programmi testati, è la loro capacità di escludere dall'algoritmo, che determina l'allineamento, la zona alterata dall'intervento .

Per eseguire un ulteriore confronto tra i vari programmi utilizzati possono essere considerati i valori, presentati in precedenza e inseriti nella Tabella 3.30, riguardanti la distanza media e la deviazione standard.

PAZIENTI	GRANDEZZE (mm)	SOFTWARE		
		3D Slicer	SolidWorks	ProPlan
A	Distanza media	0.05	0.53	0.58
	Deviazione Standard	1.54	1.89	1.91
B	Distanza media	0.1	0.10	0.20
	Deviazione Standard	0.82	0.93	0.81
C	Distanza media	0.24	0.41	0.50
	Deviazione Standard	1.53	1.85	1.77
D	Distanza media	0.04	-0.03	-0.59
	Deviazione Standard	1.22	1.18	1.65

Tabella 3.30: Tabella riportante per ogni paziente e per ogni software la distanza media e la deviazione standard ad essa associata.

Dall'analisi dei valori, inseriti nella tabella, risulta che il software che permette di ottenere la sovrapposizione migliore è *3D Slicer*. Gli indici riportati non possono essere considerati come entità separate dalle immagini raffiguranti la mappatura a colori degli scostamenti tra le superfici. Nel calcolo della distanza media e della deviazione standard, infatti, vengono presi in considerazione anche i punti delle superfici interne i quali possono provoca-

re una discrepanza tra il valore ottenuto e quello previsto dall'osservazione dell'immagine. Va ricordato inoltre che lo studio è stato eseguito su un campione molto ristretto di pazienti, quindi le considerazioni che possono essere ricavate hanno solo validità descrittiva ma non possono avere alcuna significatività statistica.

Per concludere lo studio eseguito è stato confrontato il modello tridimensionale del risultato pianificato e quello ottenuto dalla segmentazione della TAC post operatoria del paziente D. Il risultato pianificato è modellato mediante *SolidWorks* a partire dalla ricostruzione pre operatoria; esso, visibile nella Figura 3.105, mostra le ossa zigomatiche tagliate nei punti definiti dalle guide di taglio e ruotate intorno al proprio centro di rotazione. Questo punto è considerato l'origine di un sistema di riferimento in cui il piano XY è parallelo all'osteotomia orizzontale mentre il vettore \vec{z} coincide con l'asse di rotazione. Le due grandezze su cui si basa questa modellazione, ossia la posizione del centro di rotazione e la direzione dell'asse di rotazione, variando da paziente a paziente, non possono essere definite con certezza. Inoltre, le guide di taglio non definiscono l'osteotomia interna, quindi lo spessore effettivo dell'osso, determinato durante l'intervento, risulterà essere differente da quello visibile nel modello pianificato.

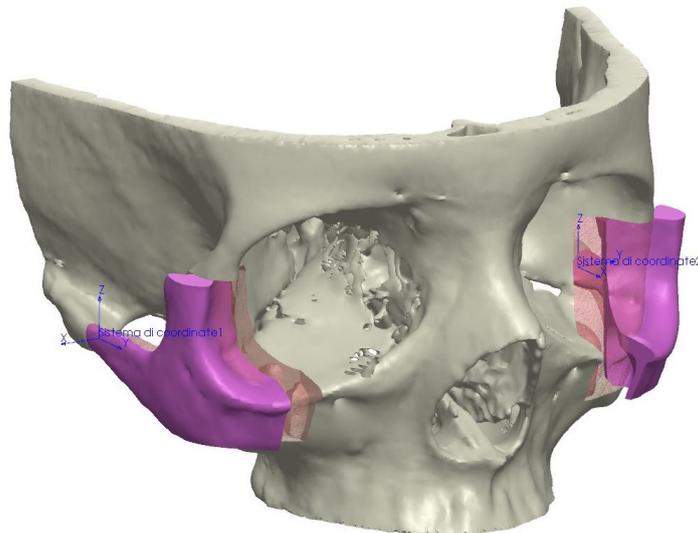


Figura 3.105: Immagine del modello pianificato in cui sono visibili i due centri di rotazione e le ossa zigomatiche, colorate di rosa, ruotate.

I confronti, riportati di seguito, sono stati effettuati valutando gli scostamenti presenti tra le ossa zigomatiche, dopo che le ricostruzioni sono state allineate mediante i software *Solidworks* e *ProPlan*. Non è stato possibile,

in questo caso, valutare l'allineamento ottenuto mediante *3D Slicer*, poichè questo software permette l'importazione del CAD raffigurante la pianificazione, ma non l'esportazione della sovrapposizione realizzata. I procedimenti seguiti e le convenzioni utilizzate, sia per quanto riguarda i colori dei modelli sia per i limiti degli intervalli di studio degli scostamenti, sono i medesimi utilizzati in precedenza.

SolidWorks

Le Figure 3.106 e 3.107 mostrano i punti anatomici scelti per l'allineamento: A e A_1 , B e B_1 sono posizionati nei fori sopraorbitali rispettivamente destro e sinistro, mentre C e C_1 sono situati sulla spina nasale anteriore. La sovrapposizione ottenuta dalla definizione degli accoppiamenti è visibile nella Figura 3.108.

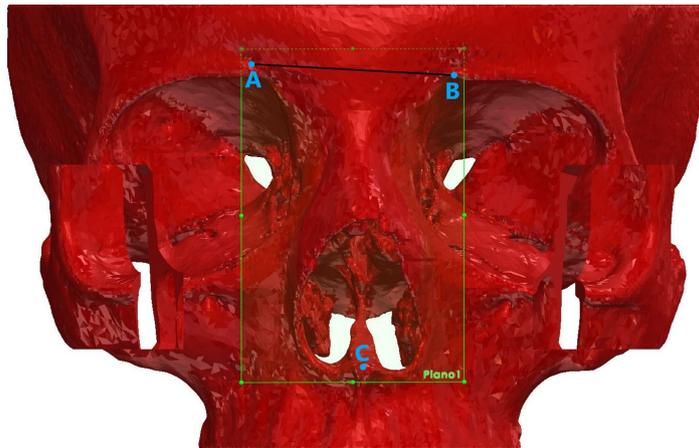


Figura 3.106: Rappresentazione sul modello pianificato della posizione dei fiducial, della retta e del piano.

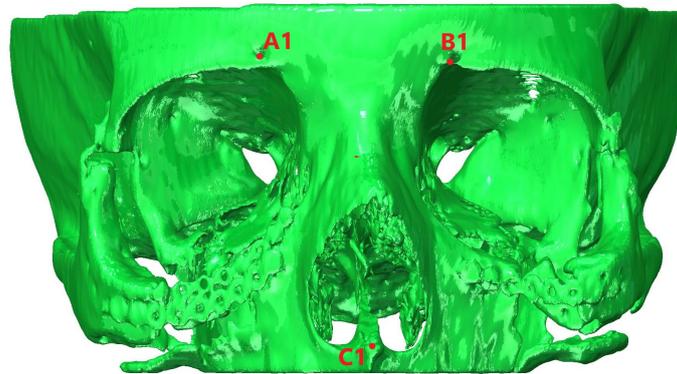


Figura 3.107: Rappresentazione sul modello post operatorio della posizione dei marcatori.



Figura 3.108: Risultato della sovrapposizione eseguita mediante *SolidWorks*, per il paziente D.

La valutazione della registrazione è stata eseguita, questa volta, con *Geomagic Qualify* il quale non solo permette di ottenere una mappatura a colori rappresentante lo scostamento tra le superfici, come *Geomagic Wrap*, ma anche di visualizzare il valore della distanza associata al punto selezionato, come mostrato nella Figura 3.109.

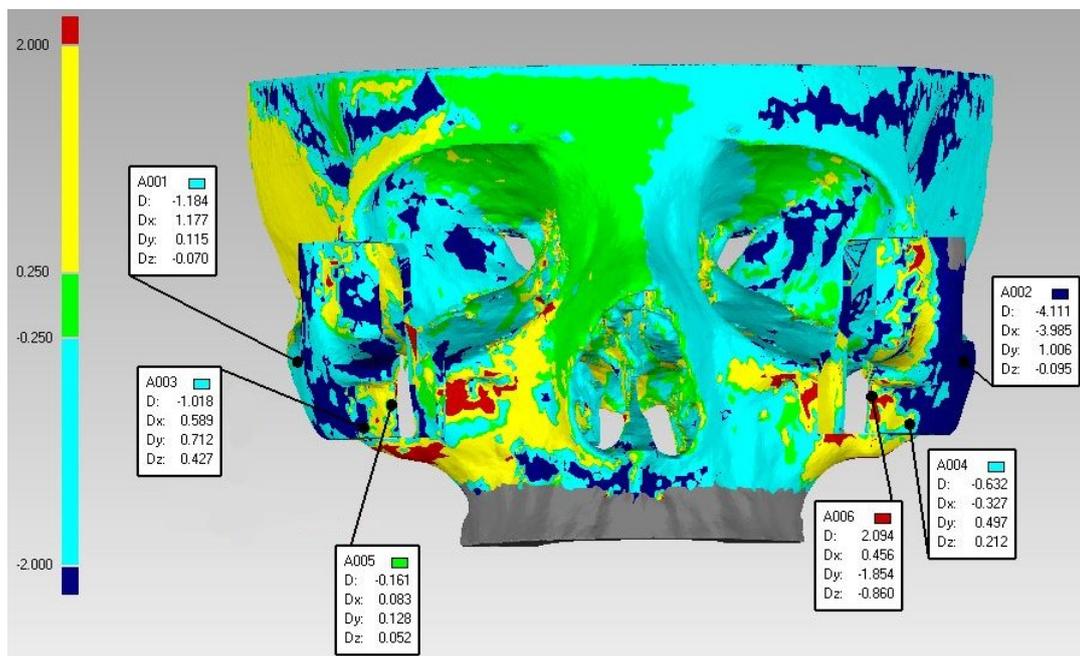


Figura 3.109: Indicazione delle distanze tra le superfici nei punti scelti.

Nella Tabella 3.110 sono riportati i valori degli scostamenti (D) visibili nell'immagine 3.109. I punti sono stati disposti in posizione simmetrica, in modo tale da poter effettuare un confronto tra la parte sinistra e quella destra.

PUNTI	DISTANZA (mm)
A001	-1.184
A002	-4.11
A003	-1.018
A004	-0.632
A005	-0.161
A006	2.094

Figura 3.110: Valori delle distanze tra le superfici nei punti scelti.

E' possibile osservare sia dalla figura 3.109 sia dalla tabella come l'emisfero destro presenti un'area fuori range di tolleranza di dimensioni inferiori rispetto all'emisfero sinistro. Se si considerano per esempio i primi due punti è possibile notare come la distanza A002 sia quasi il triplo dello scostamento tra i punti omologhi sullo zigomo sinistro.

ProPlan

La sovrapposizione eseguita mediante *ProPlan* è riportata nell'immagine sottostante. Per poterla confrontare con quella ottenuta in precedenza è stata eseguita l'analisi degli scostamenti su *Geomagic Qualify*, visibile nella Figura 3.112.

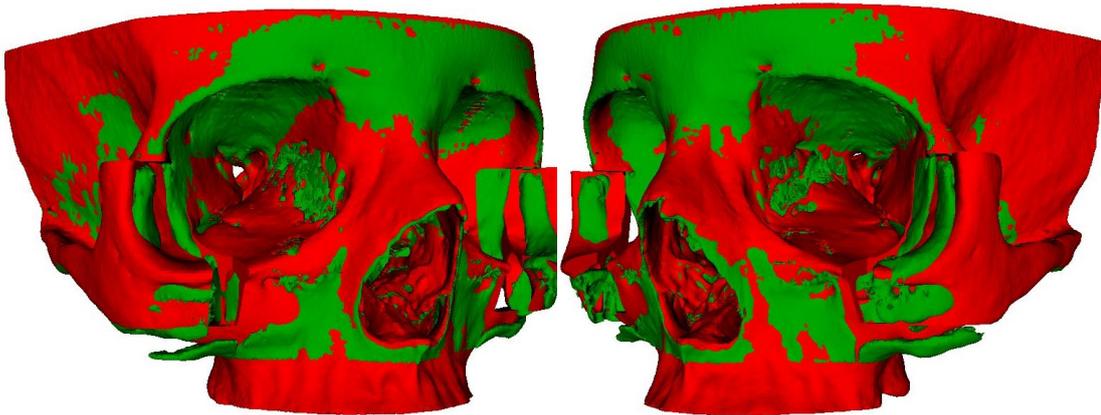


Figura 3.111: Risultato della sovrapposizione eseguita con *ProPlan*, per il paziente D.

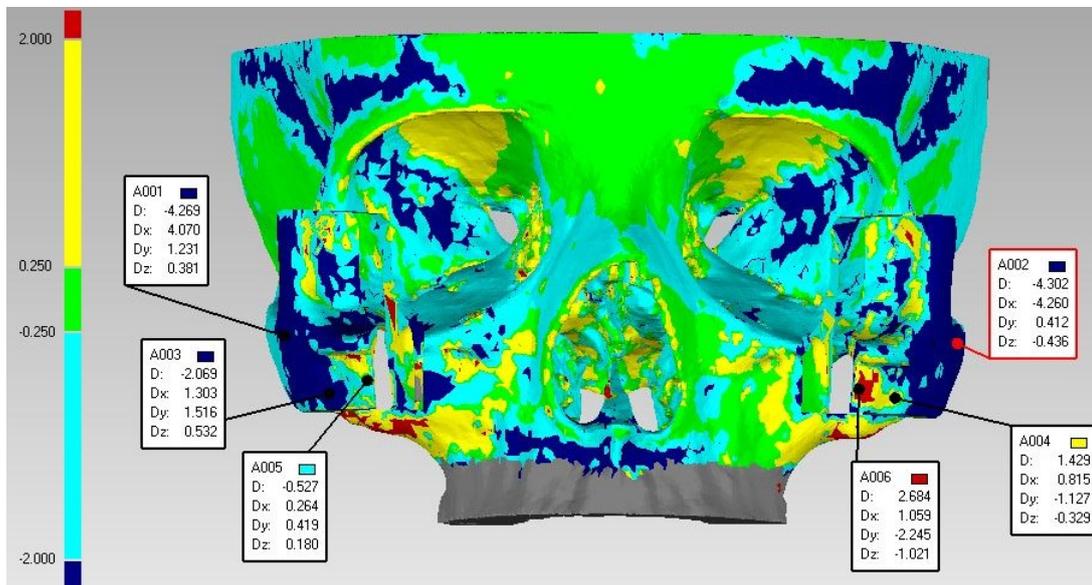


Figura 3.112: Indicazione delle distanze tra le superfici nei punti scelti.

Nella tabella sono riportati i valori degli scostamenti (D) visibili nell'immagine soprastante. La posizione dei punti è la medesima di quelli del caso precedente.

PUNTI	DISTANZA (mm)
A001	-4.269
A002	-4.302
A003	-2.069
A004	-1.429
A005	-0.527
A006	2.684

Figura 3.113: Valori delle distanze tra le superfici nei punti scelti.

Una semplice osservazione delle immagini mostra come la registrazione effettuata mediante *ProPlan* abbia fornito un risultato peggiore per quanto riguarda l'osso zigomatico destro, mentre per quello sinistro non si notano grandi differenze.

3D Slicer

Anche se l'allineamento ottenuto mediante questo Software non può essere valutato e quindi confrontato con quelli precedenti, per completezza è stata comunque riportata un'immagine della sovrapposizione effettuata con *3D Slicer*.

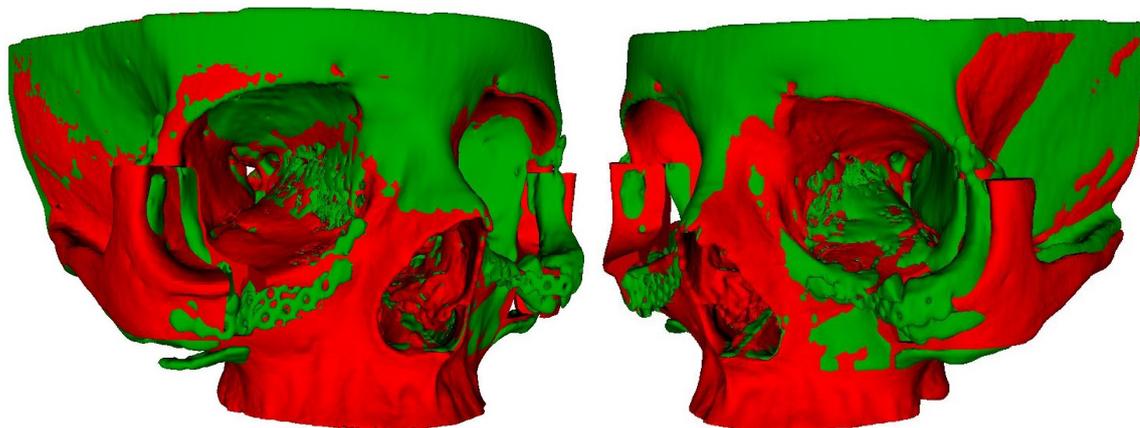


Figura 3.114: Risultato della sovrapposizione eseguita mediante *3DSlicer*, per il paziente D.

In ambito medico, la registrazione viene utilizzata per confrontare sia i referti pre e post operatori, sia la ricostruzione 3D post operatoria con il modello tridimensionale del risultato programmato. In quest'ultimo caso è stato eseguito un raffronto della ricostruzione post operatoria con quella programmata del paziente D, concentrandosi sulle ossa zigomatiche. La sovrapposizione è stata effettuata utilizzando tre software differenti (*3D Slicer*, *ProPlan* e *SolidWorks*) e i risultati ottenuti, eccetto quello di *3D Slicer*, sono stati valutati mediante *Geomagic Qualify*. Quest'ultimo programma permette di visionare sia graficamente, attraverso una mappatura a colori, sia numericamente gli scostamenti presenti tra le superfici confrontate. Dalle Figure 3.109 e 3.112 è possibile osservare come l'allineamento ottenuto con *SolidWorks*, basato sull'individuazione dei landmarks, fornisca un risultato migliore rispetto a quello generato da *ProPlan*, basato sulla definizione delle superfici. Le distanze, riguardanti lo zigomo destro, riportate nelle Tabelle 3.110 e 3.113 mostrano infatti scostamenti maggiori nel caso dell'allineamento effettuato con *ProPlan*.

Capitolo 4

Progettazione dei componenti di supporto per la stampante FDM

In questo capitolo sono descritti i componenti meccanici progettati, con i sensori individuati e il programma scritto al fine di migliorare il funzionamento della stampante Ultimaker a tecnologia FDM (Fused Deposition Modelling) utilizzata. Con l'acronimo FDM si intende una modellazione ottenuta mediante la deposizione di materiale fuso. Il filo viene trascinato dentro ad un estrusore riscaldato dove viene portato a fusione per poi fuoriuscire da un ugello calibrato. La forza motrice è fornita dall'utilizzo di un meccanismo a pignone-ruota condotta. Il pignone, costituito da una ruota godronata, è messo in rotazione mediante un motore elettrico, la ruota condotta è invece un cuscinetto che ha la funzione di generare la pressione di estrusione. Nella Tabella 4.1 sono riportate alcune caratteristiche della stampante Ultimaker 3 (fig. 4.1), utilizzata per la realizzazione di modelli tridimensionali di parti anatomiche e di guide di taglio.



Figura 4.1: La stampante Ultimaker3.[16]

Caratteristiche della stampante	
Tecnologia	FDM
Testina di stampa	a doppio ugello
Diametro del filamento	2.85 mm
Diametro ugello	0.4 mm
Velocità di spostamento della testina di stampa	30 - 300 mm/s
Materiali stampabili	PLA, PVA, ABS, CPE, Nylon
Tensione di alimentazione	100-240 VAC
Corrente di alimentazione	4 A

Tabella 4.1: Tabella sinottica riportante le principali caratteristiche della stampante Ultimaker 3.

Lo studio eseguito può essere scomposto in quattro fasi:

- a. Identificazione, per ogni problema connesso alla tecnologia di stampa utilizzata, della corrispondente azione correttiva:
 - Misura della lunghezza del filo consumata, per evitare l'interruzione della stampa del modello successivo a causa dell'esaurimento del filo.
 - Spegnimento della stampante in caso di esaurimento del materiale, per evitare la carbonizzazione del filo residuo all'interno dell'ugello.
 - Spegnimento dell'alimentazione della stampante a seguito del bloccaggio del filo della bobina.
- b. Scelta dei sensori, necessari per il corretto svolgimento delle azioni sopra presentate:
 - L'*encoder* per la valutazione del filo consumato.
 - Il *fotointerruttore* che vede cambiare lo stato logico della propria uscita a causa dell'esaurimento del filo della bobina.
 - Il *sensore reed* il quale modifica lo stato logico della propria uscita quando il filo, bloccandosi, porta un magnete in contatto con esso.
- c. Progettazione dei componenti meccanici per l'alloggiamento dei sensori e il corretto accoppiamento degli stessi con le diverse parti della stampante.
- d. Scrittura del programma per il controllo dello stato dei sensori.

4.1 Progettazione del sistema per la misura del consumo di filamento

Come precedentemente spiegato, il primo problema da risolvere consiste nello sviluppo di un sistema per la misura del filo consumato, al fine di fornire all'utente una stima precisa del materiale rimanente.

A tale scopo, si è progettato il dispositivo di Figura 4.2, la cui analisi dettagliata viene presentata nella sezione 4.1.2. I componenti principali sono l'encoder, componente 1 nell'immagine, e la ruota 2. La ruota ha il compito di trasformare il moto lineare del filo in moto rotatorio, e di trasferirlo all'encoder, il quale a sua volta trasferisce le informazioni sotto forma di impulsi alla scheda di controllo.

Prima di analizzare il sistema, si presenta di seguito il funzionamento dell'encoder e si ripercorrono i criteri seguiti per la scelta a catalogo.

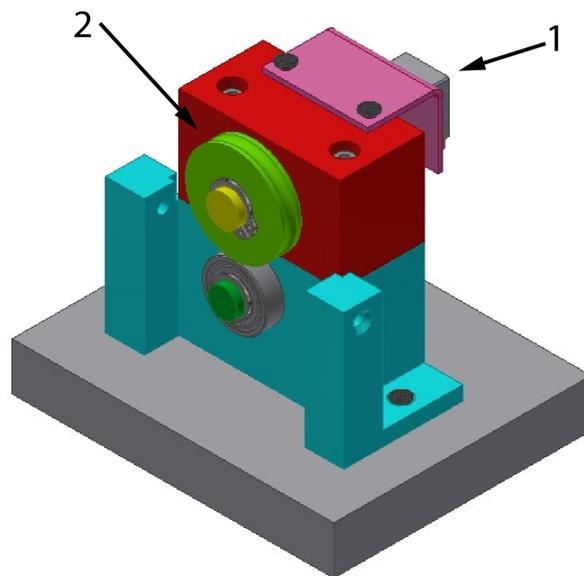


Figura 4.2: Vista complessiva del sistema di misurazione del consumo di filo.

4.1.1 L'encoder

L'encoder è un trasduttore di posizione rotativo di tipo digitale e può essere classificato come incrementale o assoluto. Un trasduttore è assoluto se esiste una relazione univoca tra segnale in uscita e valore misurato; è invece definito digitale se il valore è ottenuto dal conteggio della serie di impulsi in uscita

dal sensore. Come mostra la Figura 4.3, gli elementi principali che formano un encoder sono: il fotodiode, la lente, il regolo, il disco e le fotocellule. Le fotocellule utilizzate sono almeno due e permettono di identificare il verso di rotazione del disco graduato, una terza può essere adottata invece per rilevare la posizione di zero. La lente e il regolo hanno, invece, la funzione di rendere più netto il passaggio tra il tratto opaco e quello chiaro.

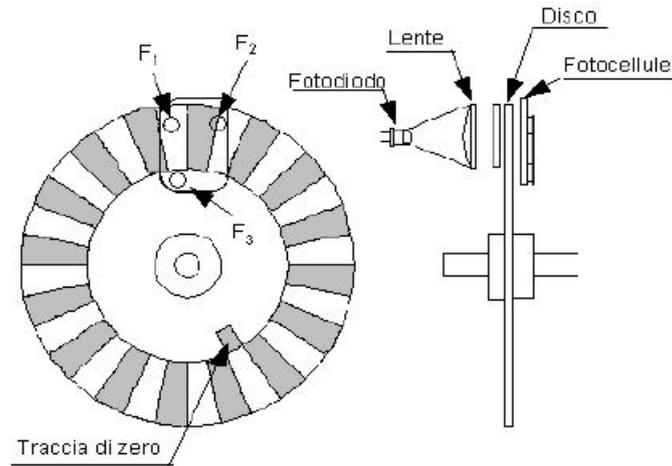


Figura 4.3: Immagine di esempio della struttura interna dell'encoder[19].

Le caratteristiche da tenere in considerazione per la scelta del sensore sono il numero di impulsi, che definisce la risoluzione, e la tensione di alimentazione. Per quanto riguarda quest'ultima grandezza, è stata scelta come valore 3 V, per ragioni di compatibilità con i GPIO pins del Raspberry Pi, che operano a 3.3 V. L'encoder individuato è il **C14D32N-D3A3** della *CUI Devices*, alcune delle sue specifiche sono riportate nella Tabella 4.2.

Proprietà dell'encoder	
Tipologia	Incrementale
n° di impulsi	32
n° di canali	2
Tensione di alimentazione	3.3 V
Tipo di uscita	In quadratura

Tabella 4.2: Caratteristiche dell'encoder C14D32N-D3A3.

Noto il numero di impulsi al giro, è possibile calcolare la risoluzione, ossia l'angolo di rotazione minimo che può essere misurato, come il rapporto tra

l'angolo giro e il numero totale di impulsi:

$$\theta_{min} = \frac{360}{32} = 11.25^\circ$$

Come accennato in precedenza, la rotazione dell'albero dell'encoder è solidale a quella di una ruota di raggio $R = 15 \text{ mm}$, contro la quale scorre il filo di cui bisogna determinare la quantità consumata durante il funzionamento della stampante. La lunghezza di filo associata a θ_{min} può essere calcolata nel seguente modo:

$$L = \frac{2\pi R \cdot \theta_{min}}{360^\circ} = 2.94 \text{ mm}$$

Nella Figura 4.4, è rappresentato lo schema del circuito elettrico dell'encoder scelto.

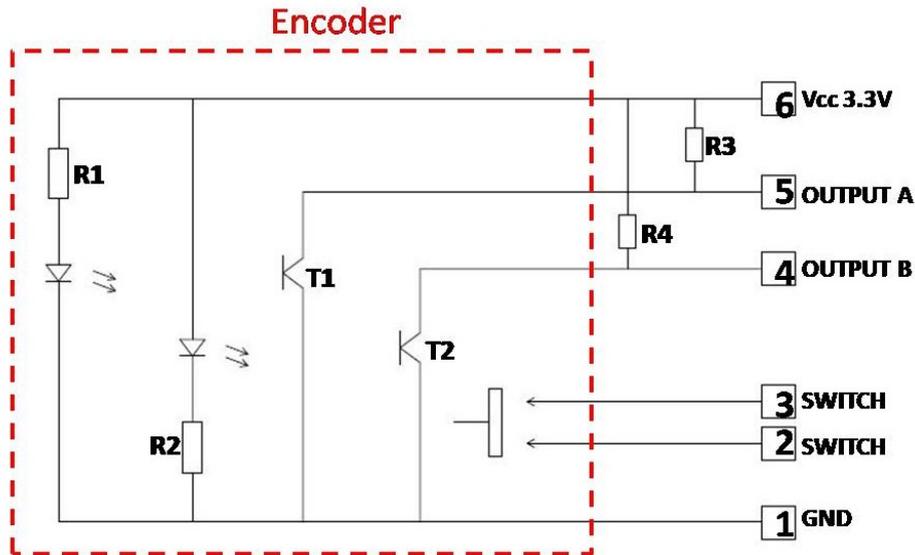


Figura 4.4: Schema del circuito elettrico dell'encoder C14D32N-D3A3.

Il sensore presenta due canali di uscita, indicati con *OUTPUT A* e *OUTPUT B*, ognuno dei quali è collegato ad un fototransistor. Questo componente, in coppia con il LED, agisce come un foto-interruttore, la cui chiusura avviene quando tra emettitore e ricevitore si interpone un tratto chiaro del disco graduato: il fascio di luce generato dal LED, non essendo ostacolato, raggiunge il transistor provocando la commutazione del suo contatto. Di conseguenza, lo stato logico dell'uscita connessa ad esso passa da 0 a 1. L'utilizzo di due fotointerruttori sfasati di 90° permette di determinare il verso di rotazione del disco. Come riportato nel circuito ricavato dal datasheet del

componente, esternamente al sensore è necessario inserire due resistenze (R_3 e R_4) con valore pari a 5.1Ω .

Nello schema è inoltre possibile notare inoltre la presenza di un *Switch* (non utilizzato) il quale, azionato dalla forza peso, può fungere da comando di conteggio del tempo.

4.1.2 Analisi del sistema di misurazione del consumo di filo

Come detto, il meccanismo progettato permette di convertire il movimento lineare del filo in un movimento rotatorio trasmesso all'encoder. Gli impulsi d'uscita vengono quindi inviati al Raspberry Pi per il conteggio.

Come visibile in Figura 4.5 e 4.6, il sistema è costituito da due metà o unità: l'unità superiore mobile, di colore rosso, e l'unità fissa inferiore, di colore azzurro. Il filo, trainato dall'estrusore, scorre tra il cuscinetto a sfere grigio e la ruota di colore verde. Quest'ultima è calettata sull'albero giallo, a sua volta solidale al perno dell'encoder.

Per garantire una corretta valutazione del consumo di materiale, l'aderenza tra il filamento e la ruota va sempre garantito: un moto di scorrimento relativo, infatti, non verrebbe rilevato dall'encoder, con inevitabili ripercussioni sui calcoli. Al fine di garantire tale condizione di aderenza, il filo è premuto contro la ruota dal cuscinetto, con una forza pari al precarico applicato alle due molle presenti nell'unità superiore. Tale forza di precarico può inoltre essere variata agendo sulle viti che collegano le due unità ed agiscono da guide. La Figura 4.5 mostra la molla nella condizione di compressione e la disposizione dei componenti della parte superiore della struttura a causa del passaggio del filo.

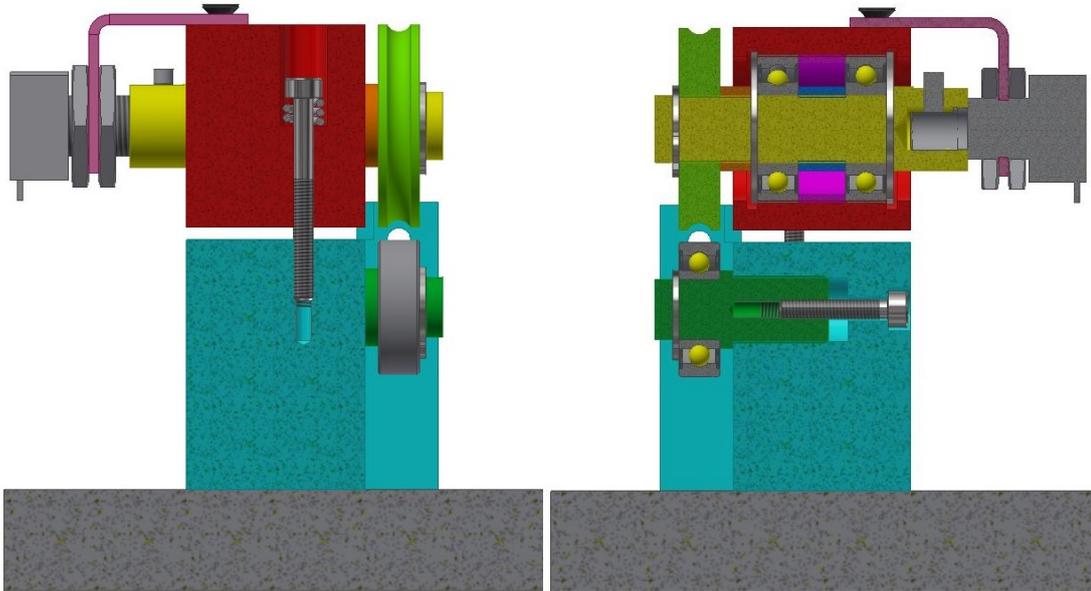


Figura 4.5: Vista in sezione del sistema di regolazione del precarico tra la ruota ed il cuscinetto a sfere.

Figura 4.6: Sezione dell'assieme modellato in corrispondenza del piano medio della struttura.

Il sensore è fissato in posizione mediante il supporto ad L di Figura fig.4.7 ed è collegato all'albero della ruota mediante un grano filettato, come visibile in Figura 4.6. Data la natura rigida di questo tipo di collegamento e non potendo inserire un giunto elastico per ragioni di ingombro, il foro sul supporto ad L ha un diametro maggiorato così da garantire gioco tra il componente stesso e il perno dell'encoder. Questo viene prima collegato all'albero e poi viene fissato alla staffa serrando il dado. Tale sequenza di montaggio garantisce che non ci siano momenti flettenti agenti sull'albero del sensore.

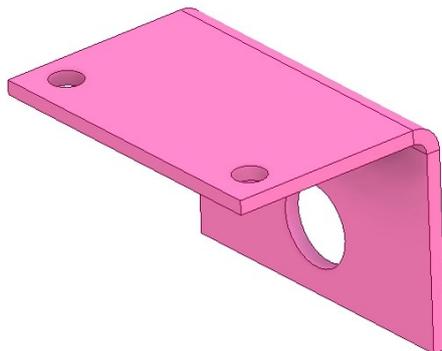


Figura 4.7: Staffa di supporto dell'encoder.

L'unità inferiore, oltre a svolgere le funzioni descritte, presenta i fori per le viti di fissaggio al basamento, nonché due fori di guida per il filo, al fine di evitare eccessive sollecitazioni sulla ruota.

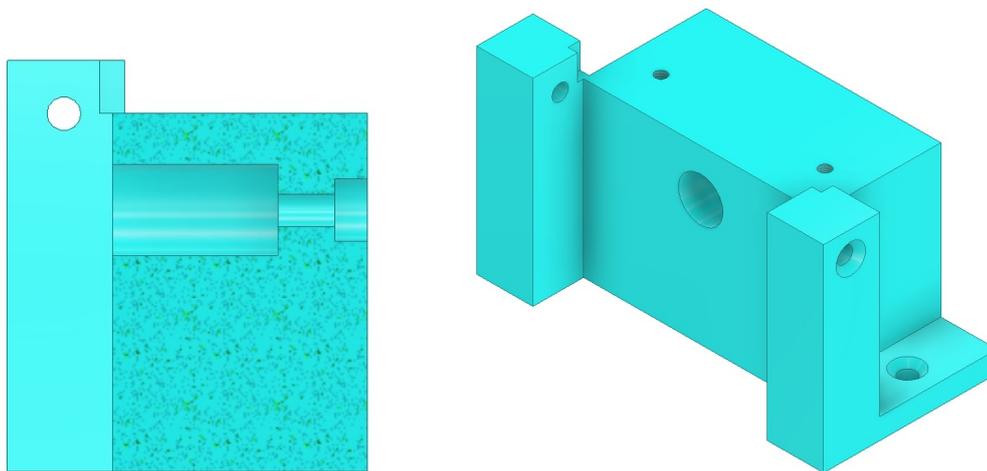


Figura 4.8: Modello e sezione del componente inferiore del meccanismo di misura del consumo di filo.

4.2 Progettazione del sistema di rilevazione dell'esaurimento del filo

Il secondo problema da affrontare consiste nel rilevare quando il materiale disponibile giunge ad esaurimento. Questa condizione infatti, se non viene rilevata prontamente e risolta, comporta la carbonizzazione del filamento residuo nell'estrusore della stampante.

La soluzione individuata consiste nell'impiego di un foto interruttore per rilevare la presenza o assenza di filo nel cammino verso l'estrusore. Tali informazioni vengono inviate al Raspberry Pi a fini diagnostici. Contemporaneamente, il dispositivo progettato controlla un relé che, in caso di assenza del filo, commuta e spegne la stampante 3D.

Come fatto nella sezione precedente, si presenta di seguito il funzionamento del fotointerruttore selezionato e si procede poi con la descrizione del dispositivo progettato e del circuito elettrico ad esso associato.

4.2.1 Il fotointerruttore

Il fotointerruttore è un sensore fotoelettrico, detto anche fotocellula, formato da una sorgente emettitrice e da un ricevitore. Questi possono essere

posizionati su due dispositivi separati, nel caso debbano trovarsi a grande distanza l'uno dall'altro, oppure su un supporto con forma ad U, se lavorano a piccole distanze. L'interposizione di un oggetto, nel caso in esame il filo, muta lo stato logico dell'uscita del sensore, permettendone così il rilevamento.

I fattori che hanno influenzato la scelta della fotocellula sono sia di natura elettrica sia di carattere meccanico. Le caratteristiche elettriche, cioè la tensione di alimentazione e quella del segnale in uscita, dipendono dal dispositivo adoperato per controllare lo stato dei sensori scelti. Come per l'encoder, è importante che la tensione d'uscita non superi i 3.3V e che la tensione di alimentazione sia di 3.3V o di 5V.

La ricerca ha portato all'individuazione del fotointerruttore OPB-618 della TT Electronics, le cui caratteristiche sono riportate in Tabella 4.3.

Caratteristiche del fotointerruttore	
Porta logica	Open collector Inverted output
Tensione di alimentazione min	4.5 V
Diodo	Luce infrarossa

Tabella 4.3: Caratteristiche elettriche del fotointerruttore OPB-616.

La Figura 4.9, rappresentante lo schema elettrico della fotocellula scelta, mostra come i componenti principali siano cinque: un LED, componente 1 nello schema, un ricevitore fototransistor 2, un amplificatore 3, uno Schmitt Trigger 4 e un'uscita a open collector 5.

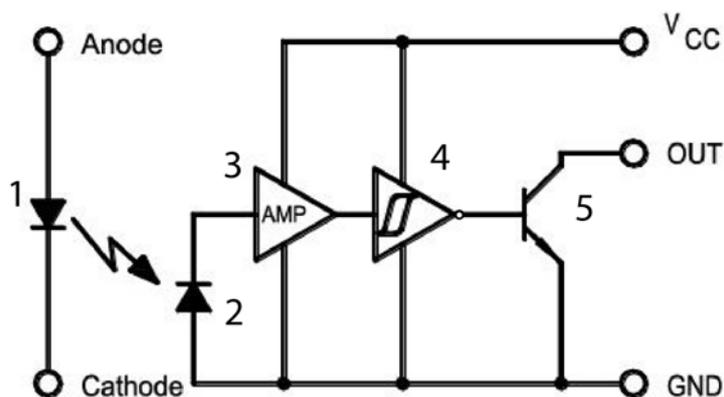


Figura 4.9: Schema circuitale del fotointerruttore OPB-618.

L'amplificatore è un circuito integrato con la funzione di amplificare il segnale in corrente in ingresso e di trasformarlo in un segnale in tensione,

con valore variabile tra GND e V_{cc} .

Lo **Schmitt Trigger** è un elemento di soglia che cambia lo stato logico della propria uscita tutte le volte che il segnale in entrata supera un determinato livello. La caratteristica dello Schmitt trigger è di possedere due livelli di soglia distinti per la transizione da 0 ad 1 e viceversa. Tale caratteristica è importante soprattutto quando il segnale in ingresso contiene del rumore che ne causa l'oscillazione nell'intorno del valore limite.

Per evitare una variazione ripetuta dello stato dell'uscita, quando il segnale in ingresso si trova al di sotto del limite inferiore, l'uscita assume il valore 0, quando al contrario si trova al di sopra della soglia superiore, il valore dell'uscita diventa 1. Nel caso invece sia compresa tra i due limiti, lo stato dell'uscita non cambia rispetto a quello immediatamente precedente.

L'**uscita a open collector** è utilizzata nel caso in cui debba essere pilotato un dispositivo che richiede una corrente o una tensione differente da quella di alimentazione della restante parte di circuito. Nel caso considerato, il circuito viene alimentato ad una tensione di 5 V, mentre l'uscita (indicata con *OUT*), essendo collegata ad un pin di input del Raspberry Pi, può ricevere solo segnali con tensione compresa tra 0 e 3.3 V.

Per poter perciò collegare il fotointerruttore alla scheda di controllo è necessario ampliare il circuito ed inserire ulteriori componenti. Inoltre, come evidenziato, il fotointerruttore deve comandare un relè per lo spegnimento della stampante.

Schema del circuito elettrico

Lo schema completo del circuito realizzato è riportato nella Figura 4.10.

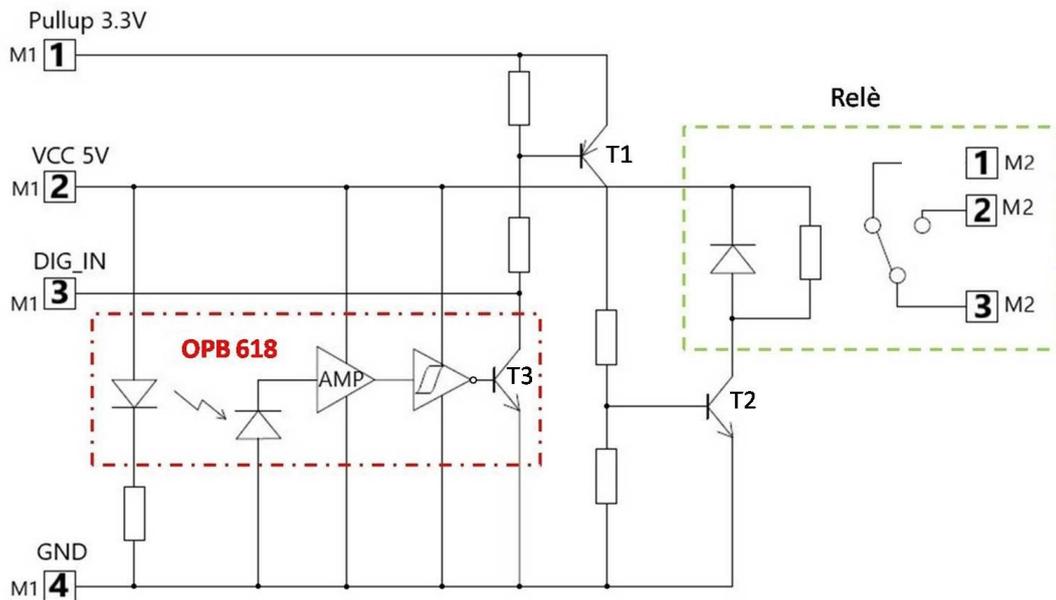


Figura 4.10: Circuito elettrico completo per il collegamento del fotointerruttore alla scheda di controllo e al relè.

Con M_1 e M_2 sono indicate due morsettiere collegate rispettivamente ai pin del fotointerruttore, riquadrato in rosso, e a quelli del relè, riquadrato in verde.

E' possibile distinguere due condizioni di funzionamento del circuito: quella "normale" corrispondente alla presenza del filo e quella "anomala" legata al suo esaurimento.

Funzionamento in presenza del filo

Il filo di PLA/ABS blocca la trasmissione del segnale luminoso tra l'emettitore e il ricevitore. Il circuito interno al fotosensore non risulta quindi percorso da corrente; tuttavia il contatto dell'open collector, ottenuto mediante uno stadio BJT (Bipolar Junction Transistor) NPN, risulta chiudersi dal momento che l'uscita dello Schmitt Trigger è negata. La chiusura del contatto del transistor T_3 mette in comunicazione il morsetto 3 di M_1 con GND e contemporaneamente genera una corrente all'interno del circuito esterno al sensore. Da un lato quindi, si ha l'invio di un segnale 0V ad uno degli ingressi digitali del Raspberry Pi; dall'altro, il passaggio di corrente comporta la chiusura dei transistor T_1 e T_2 e la conseguente attivazione del relè. La commutazione di questi due componenti si verifica quando nel circuito scorre una corrente tale da produrre una caduta tensione tra i rami della base e dell'emettitore

del transistor uguale a $0.7 V$. Data la differente tensione di alimentazione tra il fotointerruttore ($3.3V$) e il relè ($5V$), è stato necessario l'utilizzo di due transistor: T1, pilotato dal sensore, provoca la commutazione di T2 che a sua volta controlla il relè.

L'immagine riportata mostra inoltre come il collettore del transistor della fotocellula sia collegata ad un ramo alimentato dal Raspberry Pi a $3.3 V$.

Funzionamento in assenza del filo

Quando tra emettitore e ricevitore viene a mancare il filo, si genera un segnale in corrente che viene amplificato e trasformato in una variazione di tensione mediante l'amplificatore. Il segnale in uscita da quest'ultimo viene inviato allo Schmitt Trigger, il quale, come già evidenziato, ha la funzione di trasformare un segnale da analogico a digitale. Se il valore in uscita da questo componente è 1 (presenza del filo) si ha la chiusura del contatto del transistor con conseguente generazione di corrente, se invece è 0, condizione corrispondente alla mancanza del filo, l'uscita del fotointerruttore rimane aperta. La conseguente assenza di corrente nel circuito comporta due effetti: l'invio di un segnale a $3.3V$ al morsetto 3 di *M1* e l'apertura di T1 e T2, con risultante commutazione del relè e spegnimento della stampante.

Una volta definito il layout dello schema elettrico e scelti i due componenti principali, il fotointerruttore e il relè (Sezione 4.4), è importante individuare i valori da assegnare ad ogni resistenza, affinché il circuito funzioni correttamente.

Calcolo delle resistenze

Per il dimensionamento degli elementi resistivi presenti nel circuito, è necessaria la conoscenza dei valori limite delle caratteristiche elettriche riportate in Tabella 4.4.

	Electric properties	Value
INPUT DIODE	Forward DC Current	$50 mA$
	Forward Voltage V_F	$1.6 V$
OUTPUT PHOTOLOGIC	Max Voltage at Output	$30 V$
	Low level Output Current	$16 mA$
	Operating DC Supply Voltage min V_{CC}	$4.5 V$
	Low Level Output Voltage V_{OL}	$0.4 V$

Tabella 4.4: Caratteristiche elettriche del fotointerruttore OPB-618 per il dimensionamento delle resistenze.

Nella Figura 4.11 viene ripresentato il circuito completo con indicata, per

ogni componente, la nomenclatura usata nei calcoli.

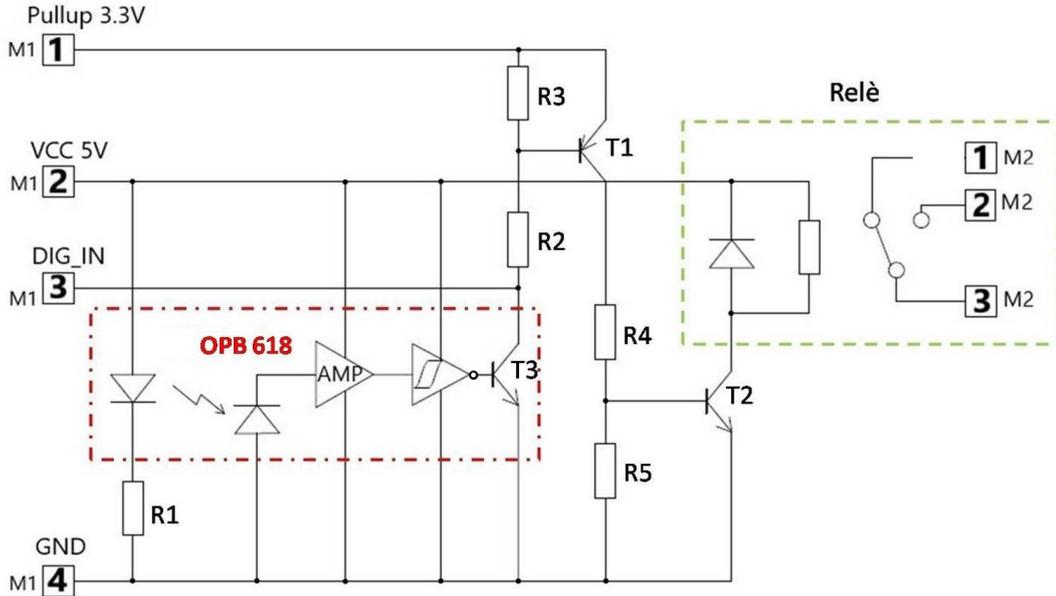
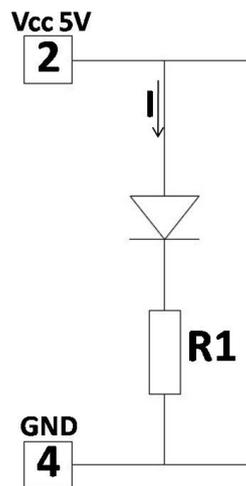


Figura 4.11: Circuito elettrico completo del fotointerruttore con indicate le sigle dei componenti usate nei calcoli.

Resistenza R_1 :



R_1 è collegata in serie al diodo del fotointerruttore. Per la sua determinazione sono stati considerati i valori della corrente massima che può scorrere

attraverso l'emettitore e la caduta di tensione prodotta ai capi dello stesso, entrambi riportati in Tabella 4.4. Posto quindi $i = 50 \text{ mA}$ e $\Delta V = 1.6 \text{ V}$ e considerando una tensione di alimentazione pari a 5.5 V , dovuta ad una tolleranza del 10%, risulta:

$$V_{cc} - \Delta V - R_1 \cdot i = 0$$

E quindi:

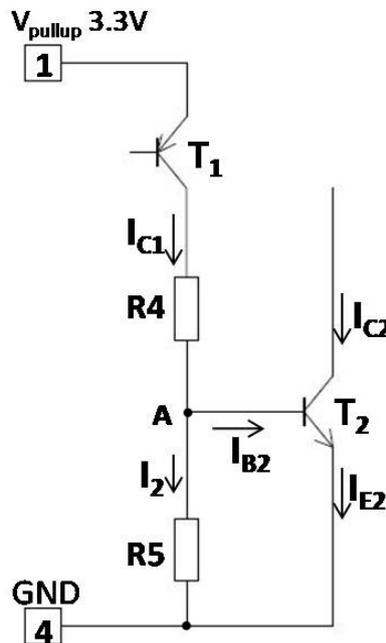
$$R_1 = \frac{V_{cc} - \Delta V}{i} = 78 \Omega$$

Dal momento che il filo utilizzato può essere sia opaco sia presentare un certo grado di trasparenza, il valore assegnato alla corrente è stato ridotto fino a 4 mA , ottenendo così:

$$R_1 = 975 \Omega$$

Una volta calcolato il valore, esso viene confrontato con quelli in commercio, in questo caso la scelta è ricaduta sulla resistenza standard da 1000Ω .

Resistenze R_4 e R_5 :



Per la determinazione delle resistenze R_2 e R_3 , è necessario prima definire i componenti resistivi R_4 e R_5 . Considerando le caratteristiche elettriche dell'utilizzatore, in questo caso il relè, riportate nella Tabella 4.6, la corrente

che attraversa la bobina del relè definisce quella del collettore del transistor T_2 , quindi $i_{C2} = 79.4 \text{ mA}$.

Ogni transistor è caratterizzato da tre correnti, quella di base i_B , quella del collettore i_C e quella dell'emettitore i_E , e da un guadagno indicato con H_{fe} , ricavato mediante un grafico e assunto, per T_2 , uguale a 100. Le relazioni che correlano tra loro queste grandezze sono:

$$i_E = i_C + i_B$$

$$i_C = i_B \cdot H_{fe}$$

Da quest'ultima equazione si ricava:

$$i_{B2,min} = \frac{i_{C2}}{H_{fe}} = 0.794 \text{ mA}$$

La corrente del collettore del transistor T_1 , indicata con i_{C1} deve quindi essere maggiore di i_{B2} . Applicando infatti la legge di Kirchhoff delle correnti al nodo A si ottiene:

$$i_{C1} = i_{B2} + i_2$$

Posta $i_{C1} = 1 \text{ mA}$ si ottiene un valore per i_2 pari a 0.206 mA .

Per il dimensionamento delle resistenze è necessario conoscere le cadute di tensione generate tra emettitore-collettore per T_1 e tra emettitore-base per T_2 . Quest'ultima essendo in parallelo alla resistenza R_5 definisce la caduta di tensione $V_5 = 0.7 \text{ V}$, ricavata da datasheet; mentre la differenza di tensione tra emettitore e collettore di T_1 , indicata con ΔV e pari a 0.3 V . Applicando la legge di Kirchhoff delle tensioni e ricordando che $V_{pullup} = 3.3 \text{ V}$, l'equazione di equilibrio risulta:

$$V_{pullup} - \Delta V - V_4 - V_5 = 0$$

Quindi

$$V_4 = V_{pullup} - \Delta V - V_5 = 2.3 \text{ V}$$

Note le cadute di tensione ai capi dei componenti resistivi e le correnti, è possibile effettuare una stima del valore delle resistenze applicando la legge di *Ohm*:

$$R_{4max} = \frac{V_4}{i_{C1}} = 2.4 \text{ k}\Omega$$

$$R_5 = \frac{V_5}{i_2} = 3.39 \text{ k}\Omega$$

Scegliendo per entrambe le resistenze il valore di $1 \text{ k}\Omega$, la corrente che effettivamente giungerà alla base del transistor sarà

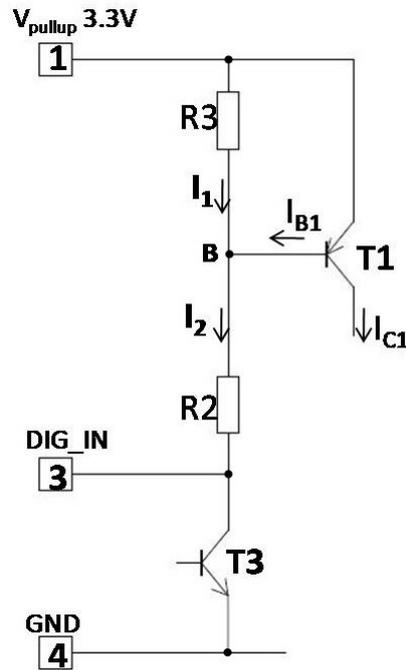
$$i_{B2} = i_{C1} - i_2 = 1.6 \text{ mA}$$

Con:

$$i_{C1} = \frac{V_4}{R_4} = 2.3 \text{ mA}$$

$$i_2 = \frac{V_5}{R_5} = 0.7 \text{ mA}$$

Resistenze R_2 e R_3 :



Definita nello studio precedente la corrente di collettore i_{C1} , si può calcolare la corrente di base $i_{B1 \min}$ che deve essere garantita sul ramo di base di T_1 :

$$i_{B1 \min} = \frac{i_{C1}}{H_{fe}} = 0.023 \text{ mA}$$

con H_{fe} guadagno ricavato dal datasheet del componente transistor e assunto pari a 100.

Per effettuare una stima della resistenza R_2 , si applica la legge di Kirchhoff delle tensioni e si ottiene la seguente equazione di equilibrio:

$$V_{pullup} - V_3 - V_2 - \Delta V = 0$$

con $\Delta V = 0.4 \text{ V}$ caduta di tensione tra collettore ed emettitore del transistor T_3 . Posto $V_{pullup} = 3.3 \text{ V}$ e $V_3 = 0.7 \text{ V}$, definita dalla caduta di tensione

presente tra emettitore e base di T_1 , poichè collegati in parallelo a R_3 , si ottiene la tensione V_2 :

$$V_2 = V_{pullup} - V_3 - \Delta V = 2.2 \text{ V}$$

Trascurando inizialmente la presenza di R_3 e considerando perciò come valore minimo della corrente i_2 0.023 mA , la resistenza R_2 risulta:

$$R_{2\max} = \frac{V_2}{i_2} = 95.65 \text{ k}\Omega$$

Assumendo anche in questo caso l'uguaglianza tra le resistenze, ossia $R_2 = R_3 = 2200 \Omega$, e applicando nuovamente la legge di Kirchhoff delle correnti al nodo B , è possibile calcolare i valori delle correnti i_1 e i_2 e verificare che i_{B1} sia maggiore di 0.024 mA :

$$i_1 = \frac{V_3}{R_3} = 0.3 \text{ mA}$$

$$i_2 = \frac{V_2}{R_2} = 1 \text{ mA}$$

$$i_{B1} = i_2 - i_1 = 0.7 \text{ mA}$$

Identificati quindi tutti i componenti, si è proceduto alla produzione del circuito stampato riportato in Figura 4.12.

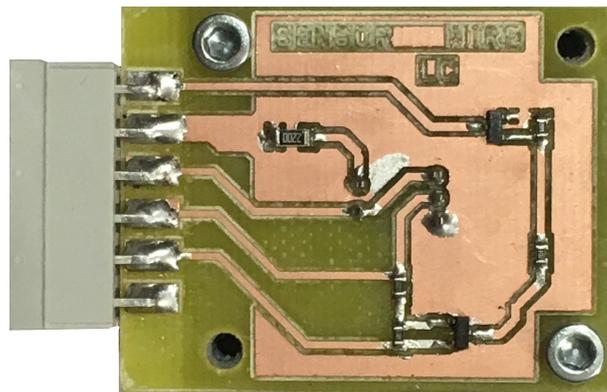


Figura 4.12: Immagine del circuito stampato realizzato.

Come accennato nel primo paragrafo della Sottosezione 4.2.1, la scelta del fotointerruttore ha richiesto la valutazione sia delle caratteristiche elettriche sia di quelle meccaniche, nello specifico gli ingombri e la distanza

tra emettitore e ricevitore, che deve essere maggiore del diametro del filo (2.85 mm). Nella Figura 4.13, raffigurante il disegno meccanico della fotocellula scelta, è possibile osservare come il diodo e il ricevitore si trovino ad una distanza di $3.81 \pm 0.25 \text{ mm}$, mentre il volume massimo occupato è $9.14 \text{ mm} \times 5.58 \text{ mm} \times 8.38 \text{ mm}$.

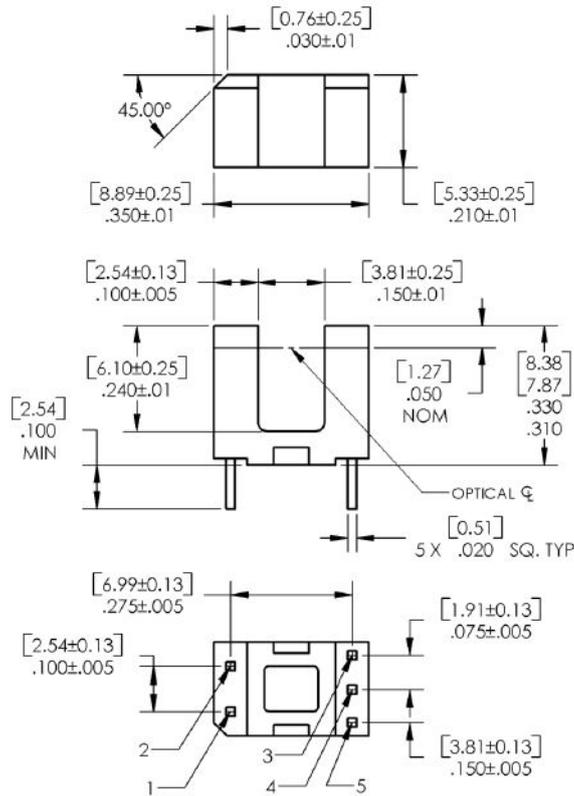


Figura 4.13: Disegno meccanico del fotointerruttore OPB618

4.2.2 Progettazione del supporto

Note le dimensioni del sensore, è stato progettato il supporto di Figura 4.14. Questo, realizzato in PLA, presenta i fori di ingresso ed uscita del filo e la sede per l'alloggiamento del fotointerruttore, rosso nell'immagine. Tale sede è stata realizzata con una profondità maggiore dell'altezza della fotocellula, in modo tale che il peso del circuito stampato gravi sulla superficie del supporto, a cui è fissato mediante l'utilizzo di quattro viti, e non sul sensore. Il supporto è a sua volta fissato in posizione mediante due viti a testa svasata.

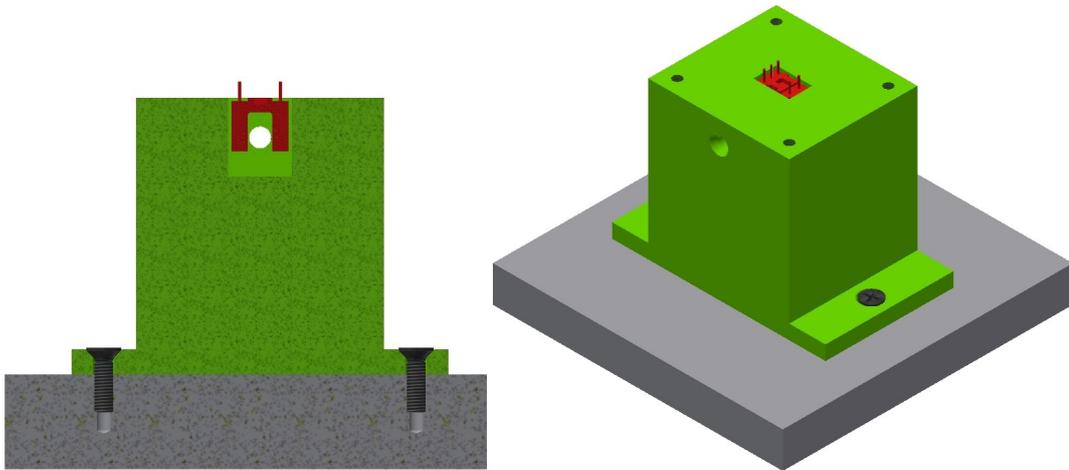


Figura 4.14: Supporto per il fotointerruttore che rileva la presenza o assenza del filo.

4.3 Dispositivo per il rilevamento del bloccaggio del filo

Il terzo ed ultimo problema da affrontare consiste nel rilevare quando, a causa di un avvolgimento non corretto sulla bobina, il filo rimanga bloccato. A seguito di questo bloccaggio infatti, l'estrusore non è più in grado di trascinare il filamento, con conseguente carbonizzazione dello stesso nell'ugello. Per rilevare il bloccaggio, si è progettato un sistema, visibile in Figura 4.15, il cui elemento sensoriale è costituito da un finecorsa di tipo reed. Quando lo svolgimento del filo è impedito, un magnete, presente all'interno del componente azzurro, si avvicina al sensore reed, provocandone la commutazione. L'analisi dettagliata del funzionamento viene eseguita nella Sottosezione 4.3.2. Di seguito si presentano le caratteristiche dei finecorsa reed, la procedura seguita per la scelta del sensore e il circuito realizzato per il suo utilizzo. Come nel caso del fotointerruttore, nel caso in cui lo scorrimento del filo sia impedito, si deve inviare un segnale di errore al Raspberry Pi a si deve procedere allo spegnimento della stampante.

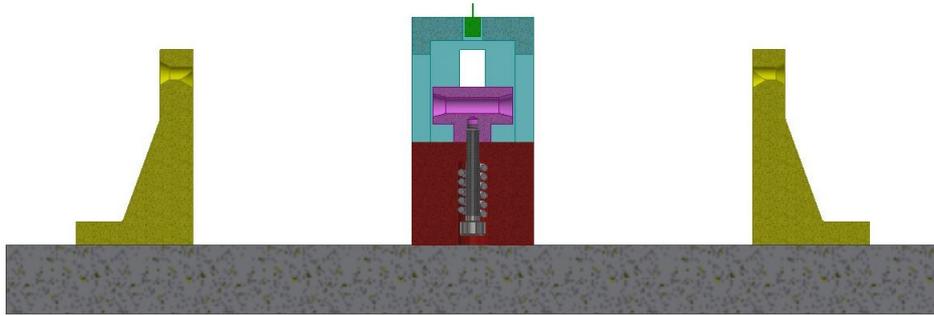


Figura 4.15: Sistema per il rilevamento del bloccaggio del filo.

4.3.1 Il sensore reed

I sensori reed, o finecorsa reed, rientrano nella classe dei sensori di prossimità ed il loro funzionamento è basato sull'azionamento di contatti, formati da lamelle e definiti *reed*, mediante l'uso di un campo magnetico esterno. Con riferimento al libro "Automazione dei sistemi meccanici"[17], i finecorsa reed possono essere classificati in base al tipo di contatto che presentano:

Sensore in configurazione NA

L'attrazione tra le due lamine è dovuta all'azione di un campo magnetico esterno, mentre l'allontanamento di quest'ultimo ripristina la condizione di riposo.

Finecorsa in configurazione NC

Le lamelle sono inizialmente mantenute chiuse mediante l'utilizzo di un magnete solidale all'ampolla reed. La commutazione del contatto si verifica quando un magnete esterno, con polarità opposta a quella del magnete ausiliario, riesce ad annullare il campo magnetico preesistente.

Sensore con contatti in scambio

La barretta, collegata al terminale comune, è in materiale ferromagnetico e può muoversi tra due terminali provvisti di lamina, quello NC è in materiale amagnetico, mentre quello NA è in materiale ferromagnetico. La commutazione del contatto è sempre dovuta alla presenza di un campo magnetico esterno.

Finecorsa con contatto bistabile

L'ampolla reed presenta un magnete interno che premagnetizza le barrette senza chiudere il contatto. L'avvicinamento di un magnete con polarità ugua-

le a quella del magnete ausiliario rinforza il campo magnetico già esistente, provocando la chiusura del contatto che rimane tale anche dopo l'allontanamento del magnete esterno. Se invece ad avvicinarsi è un magnete con polarità opposta a quella pre esistente, allora le due lamelle si allontanano tra di loro.

I criteri guida per la scelta sono gli stessi presentati per il fotointerruttore della sezione precedente. Il sensore reed individuato, del quale sono riportate alcune caratteristiche nella Tabella 4.5, è l'*MK06-6-A* della Standex Meder.

Modello	Form A
Tipo di contatto	Contatto NA SPST (Single pole single throw)
Sensitivity AT Range	05-10

Tabella 4.5: Caratteristiche elettriche del sensore reed MK06-6-A.

Come riportato sul datasheet del sensore, la struttura del codice identificativo è:

MK06 - **0** - **X**

Con **0** viene definita la classe di dimensione. E' stato necessario scegliere la **6**, dal momento che in commercio non è disponibile una classe inferiore con la sensibilità magnetica desiderata (**A**), corrispondente a **X** nel codice.

Come accennato, il fincorsa individuato svolge due funzioni: trasmettere il segnale ad uno dei pin di ingresso del Raspberry Pi a fini diagnostici e comandare l'attivazione di un relè per lo spegnimento della macchina. La Figura4.23 riporta lo schema del circuito realizzato a tale scopo.

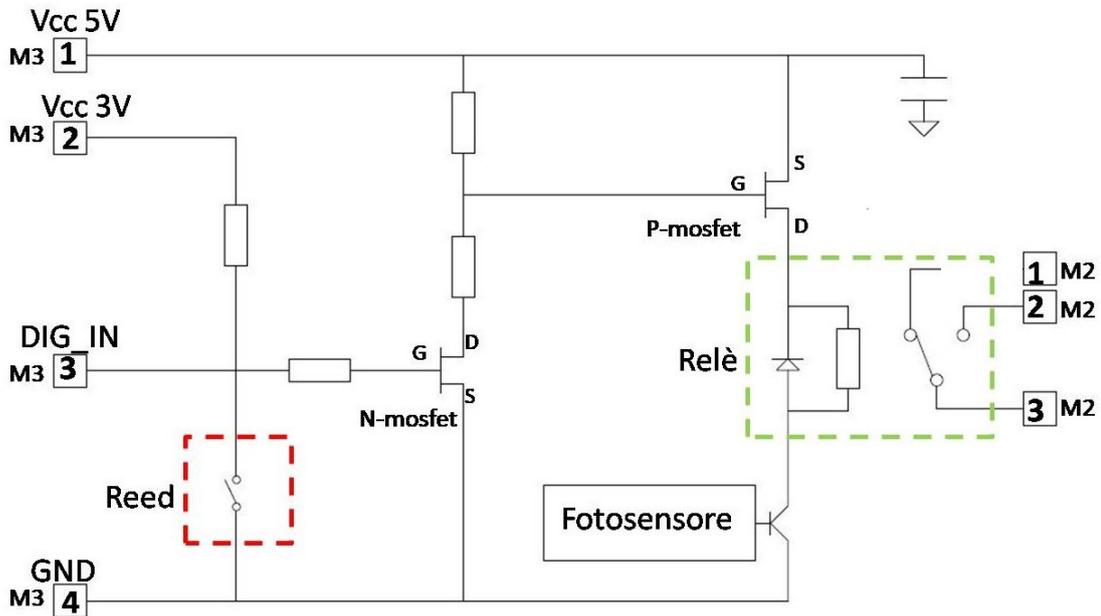


Figura 4.16: Schema circuitale del sistema per il rilevamento del bloccaggio del filo.

In analogia con quanto fatto per il fotointerruttore, anche in questo caso si possono distinguere un funzionamento "normale", che corrisponde ad uno scorrimento uniforme del filo, e un comportamento "anomalo", il quale si verifica nel caso del bloccaggio dello stesso.

Funzionamento con scorrimento omogeneo del filo

Nel caso preso in considerazione, il magnete, trovandosi lontano dal sensore, non causa la commutazione del contatto, che quindi rimane aperto. Non scorrendo corrente in questo ramo del circuito, viene inviato al pin di ingresso del Raspberry Pi un segnale a 3.3 V, pari quindi alla tensione di alimentazione del sensore reed. Contemporaneamente il contatto del N-Mosfet commuta e, affinché ciò avvenga, la caduta di tensione tra Gate e Source V_{gs} deve essere maggiore o uguale a 2.4 V. La chiusura del primo Mosfet comporta la commutazione del P-Mosfet, dovuta ad una caduta di tensione tra Gate e Source maggiore o uguale a $-2.4 V$, e la conseguente creazione di una corrente nel circuito che alimenta il relè. Il passaggio di corrente nel circuito del relè è permesso dalla contemporanea chiusura del transistor del circuito del fotointerruttore: rientriamo infatti nel caso di *funzionamenti in presenza del filo*.

Funzionamento a filo bloccato

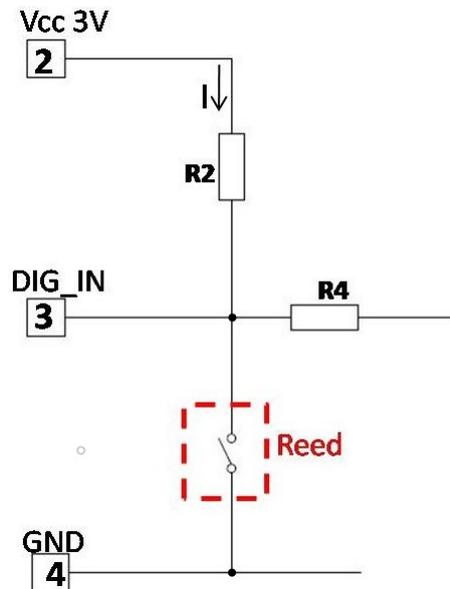
Il bloccaggio del filo comporta l'avvicinamento di un magnete esterno al

senore reed e la conseguente chiusura del contatto. La commutazione del sensore genera un segnale in ingresso al Raspberry Pi pari a 0 V e l'apertura dei due Mosfet, poichè non risultano più verificate le due relazioni di maggioranza ($V_{gs} \geq 2.4 V$ nel caso del N-Mosfet e $V_{gs} \geq -2.4 V$ nel caso del P-Mosfet). L'apertura dei contatti di questi componenti comporta la commutazione del relè e il conseguente spegnimento della stampante.

Calcolo delle resistenze

Come già fatto per il circuito del fotointerruttore, si procede di seguito alla definizione dei valori degli elementi resistivi all'interno del circuito, al fine di garantire il funzionemante appena descritto.

Dimensionamento di R2 e R4

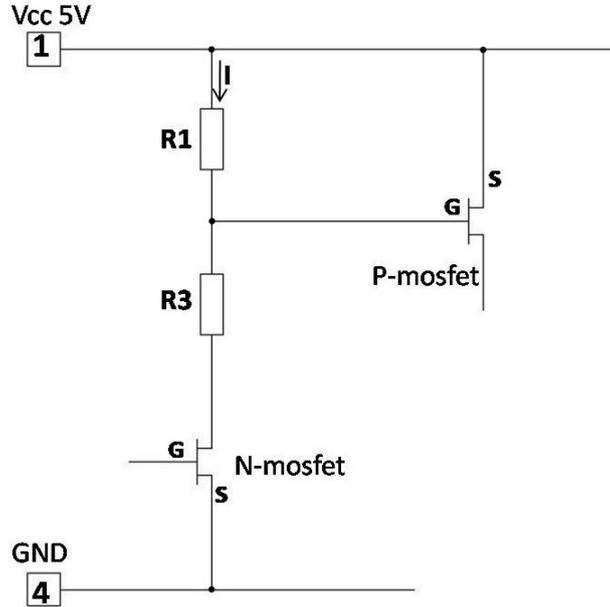


Considerando un valore di corrente pari a 5 mA e una tensione di alimentazione $V_{cc} = 3.3 V$, è stata calcolata la resistenza R_2 .

$$R_{2max} = \frac{V_{cc}}{i} = 660 \Omega$$

Noto il limite massimo, è stato scelto come valore 620 Ω , mentre il componente resistivo R_4 , posto pari a 1 Ω , è stato inserito per ragioni di sicurezza.

Dimensionamento di R1 e R3



Ai capi della resistenza R_1 , montata in parallelo al collegamento tra gate e source del mosfet, è necessaria una caduta di tensione V_1 pari a 2.4 V affinché il mosfet si chiuda. Considerando una corrente pari a 0.5 mA e una tensione di alimentazione di 5.5 V , è possibile stimare il valore minimo per R_1 :

$$R_{1min} = \frac{V_1}{i} = 4.8\text{ k}\Omega$$

Noto R_{1min} è stato scelto come valore $10\text{ k}\Omega$. Mentre per R_3 viene stimato il valore massimo:

$$R_{3max} = \frac{V_{3max}}{i} = 6.2\text{ k}\Omega$$

con $V_{3max} = V_{cc} - V_1 = 5.5 - 2.4 = 3.1\text{ V}$

Ad R_3 è stato quindi assegnato il valore $1\text{ k}\Omega$.

4.3.2 Analisi del sistema di rilevamento del bloccaggio del filo

Il sistema sviluppato, visibile in Figura 4.17, è realizzato in PLA ed è composto da sei elementi: la base in rosso, il coperchio superiore in azzurro, il corpo centrale mobile in viola, il sensore reed in verde e le due staffe in giallo. Come visibile nell'immagine, il filo in ingresso e in uscita scorre attraverso le

due staffe ed attraverso il corpo centrale, solidale al quale si trova un piccolo magnete. In condizioni normali, questo elemento mobile viene mantenuto nella posizione in figura, lontano dal reed, da un sistema a molla. In caso resti bloccato, il filamento si tende e solleva l'elemento viola, portando così in contatto il magnete ed il reed.

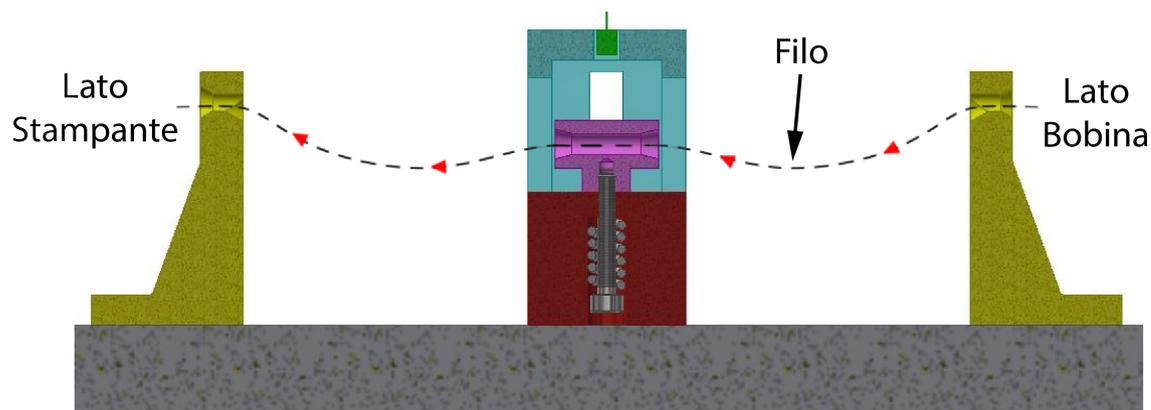


Figura 4.17: Vista in sezione del sistema per il rilevamento del bloccaggio del filo.

La base

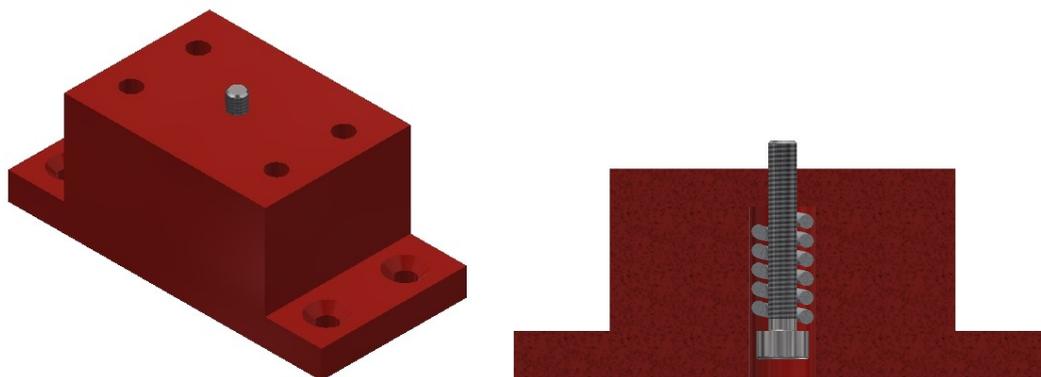


Figura 4.18: Vista isometrica e sezione della base.

Come visibile in Figura 4.18, il componente, che costituisce la base del dispositivo, presenta quattro fori laterali per il fissaggio dell'intero supporto al basamento. Altri quattro fori permettono invece il bloccaggio del coperchio superiore azzurro e un foro centrale consente l'inserimento di una vite, il cui movimento di traslazione verticale è solidale a quello del corpo centrale mobile. Come evidenziato nella vista in sezione di figura 4.18, quando questo

trasla verso l'alto, la testa della vite comprime una molla, la quale svolge tre funzioni: generare la forza elastica necessaria a mantenere l'elemento mobile a contatto con la base nella condizione di riposo (Figura 4.17), evitare l'impatto tra il corpo centrale e il reed quando il filo si blocca e, infine, impedire che piccole tensioni del filo provochino uno spostamento del componente e la consecutiva commutazione indesiderata del sensore.

Il corpo centrale mobile

Il corpo centrale mobile è in grado di traslare in direzione verticale, verso il sensore reed. Il suo movimento è guidato dalla presenza di due sporgenze laterali che scorrono all'interno di guide realizzate nel componente azzurro. In Figura 4.19 è possibile osservare, inoltre, la presenza di un foro per il passaggio del filo e un foro cieco per il fissaggio della vite di collegamento con la base.

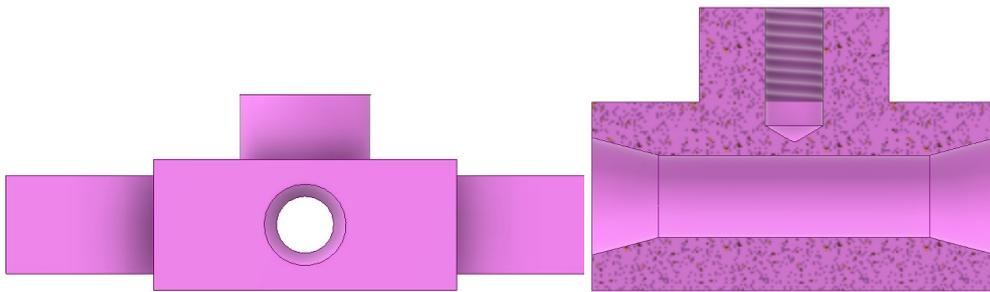


Figura 4.19: Viste del corpo centrale mobile.

Il coperchio superiore

Il coperchio superiore, visibile in Figura 4.21 ha il duplice scopo di alloggiare il sensore reed e di servire da guida per la traslazione del componente centrale mobile. Dal disegno meccanico del sensore, riportato in Figura 4.20, si sono ricavate le quote per il foro rettangolare sulla faccia superiore del coperchio. Le scanalature sulle facce laterali consentono da un lato lo scorrimento del filo, dall'altro fungono da guida prismatica per le ali laterali del corpo centrale mobile.

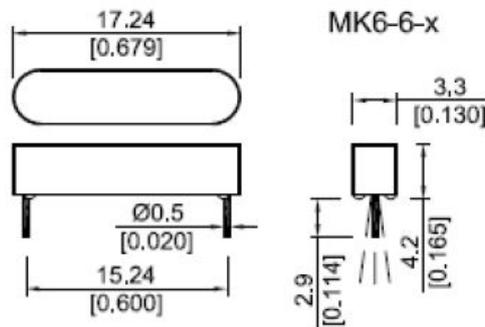


Figura 4.20: Disegno (ricavato dal datasheet) riportante le dimensioni meccaniche del sensore.

Il sensore reed è montato su una scheda appositamente realizzata e fissata al Coperchio mediante quattro viti.

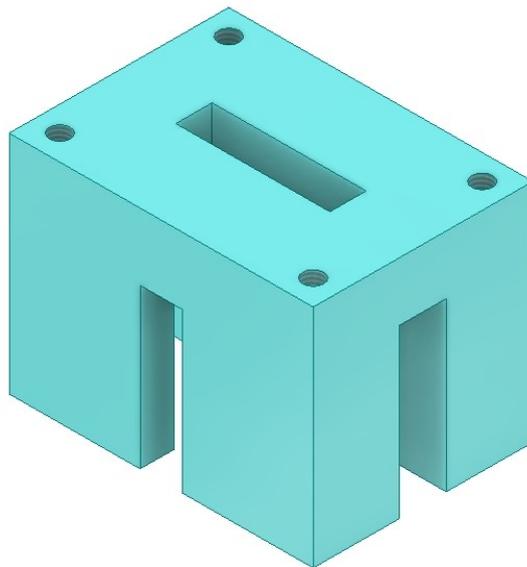


Figura 4.21: Raffigurazione del coperchio modellato.

Le staffe ad L

Per evitare sollecitazioni sugli altri due dispositivi progettati dovute allo spostamento del filo, questo viene guidato dai fori orizzontali presenti nei supporti ad L di Figura 4.22. Come evidenziato nella vista in sezione, questi fori presentano degli smussi in ingresso e in uscita, al fine di favorire la

posizione di riposo del corpo centrale mobile. Due fori sulla base consentono infine il fissaggio delle staffe in posizione.

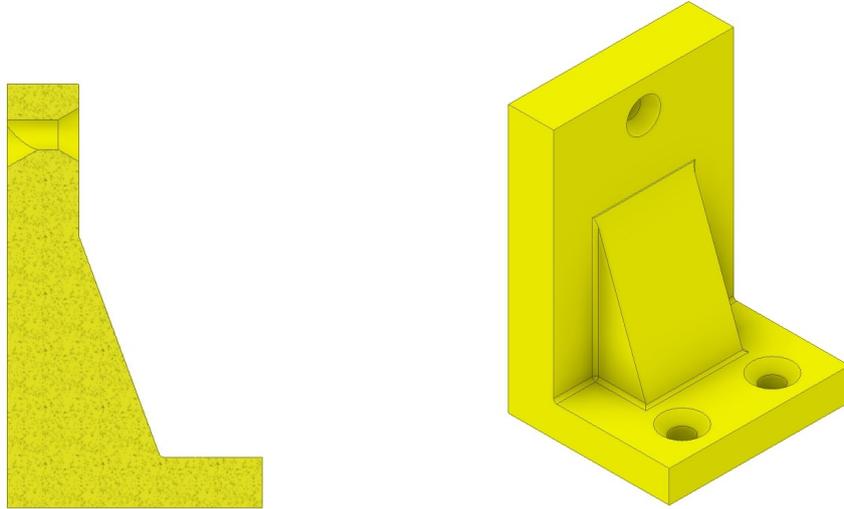


Figura 4.22: Viste dei supporti ad L per la guida del filo.

4.4 Selezione del relè per lo spegnimento della stampante

I relè possono essere principalmente di due tipi: statici, formati solo da componenti elettrici e quindi con una vita utile maggiore, o elettromeccanici. I componenti che caratterizzano questi ultimi sono: la bobina, l'elemento mobile e i contatti NC e NA. Un esempio dello schema di un relè elettromeccanico è riportato in Figura 4.23.

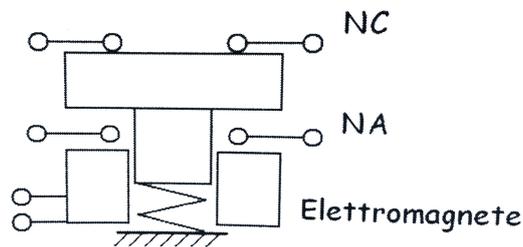


Figura 4.23: Schema di un relè con contatti NA e NC[17].

La posizione dell'elemento mobile nella condizione di riposo viene mantenuta da una molla. Una volta eccitato l'elettromagnete i contatti commutano

per via dello spostamento del corpo mobile, il quale provoca la chiusura del contatto NA e l'apertura di quello NC.

Il relè elettromeccanico scelto per l'applicazione in esame è il *G5LE-1* della Omron, le cui principali caratteristiche sono riportate in Tabella 4.6.

Caratteristiche elettriche del relè		
COIL	Rated Voltage	5 VDC
	Rated Current	79.4 mA
CONTACTS	Contact type	SPDT Single Pole Double Throw
	Rated carry current	10 A

Tabella 4.6: Caratteristiche elettriche del relè G5LE-1.

Questo componente, pilotato sia dal fotointerruttore sia dal sensore reed, è usato per interrompere l'alimentazione alla stampante 3D, provocandone così lo spegnimento.

Nella Figura 4.24 è riportato lo schema del circuito della scheda su cui è montato il relè. Ricordando quanto detto nelle sezioni precedenti, la presenza del filo e il suo scorrimento regolare porta alla commutazione del relè, mettendo in comunicazione i morsetti 2 e 3 della morsettiera M2 e permettendo così l'alimentazione della stampante. Al contrario, nel caso in cui il filo risulti bloccato o mancante, il contatto del relè, non essendo più alimentato, commuta interrompendo l'alimentazione alla macchina.

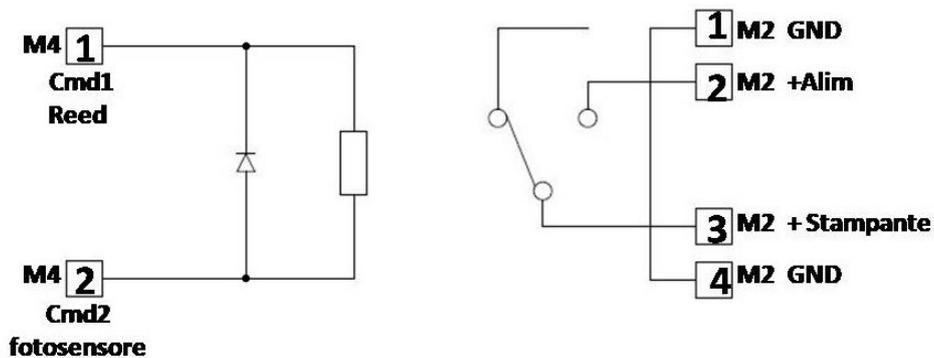


Figura 4.24: Schema del circuito elettrico del relè.

4.5 Programma per la lettura dello stato dei sensori

Per fornire all'utente una pronta visualizzazione dello stato logico dei sensori precedentemente descritti, nonché del consumo del filo, è stato sviluppato un programma in linguaggio python in grado di elaborare i dati raccolti e di trasmetterli ad un'interfaccia grafica.

Il programma nel complesso può essere suddiviso in due macro blocchi: il setup ed il loop. Il setup è una funzione contenente la parte di codice che viene eseguita solo una volta, all'avvio del programma; il loop è invece una funzione eseguita ciclicamente fino allo spegnimento del Raspberry Pi. Nel seguito il programma viene suddiviso in sezioni e ciascuna viene analizzata in dettaglio.

4.5.1 Il main e le variabili globali

Il programma comincia con le dichiarazioni di *import*, con le quali si importano le librerie necessarie. Vengono poi definite diverse variabili globali, per poi arrivare alla funzione *main*, la prima eseguita all'avvio del programma, che contiene le chiamate a *setup()* e *loop()*. E' importante notare come la chiamata a quest'ultima sia inserita all'interno di un ciclo *while* infinito da cui la natura ciclica precedentemente evidenziata.

```
1 import RPi.GPIO as GPIO
2 import numpy as np
3 from time import sleep
4
5
6 Channel_A=8
7 Channel_B=10
8
9 Btn_Reset=19 #Bottone Giallo
10 fotoInt=6 #Bottone Verde
11 fineCorsa=18
12 sftwReset=23
13 Led_G=12
14 errorLed=21 #Led_R
15
16 enc_val=np.uint8(0)
17 enc_count=np.float32(0)
18 last_save_count=np.float32(0)
19
20 flagFotoint=1
21 flagFinecorsa=1
22
```

```
23 reset=0
24 stateChangeFotoInt=0
25 stateChangeFinecorsa=0
26 enc2m=0.0029 #l=pi*D
27
28 globalState=['0','0']
29
30 def main():
31     setup()
32     while 1:
33         loop()
```

4.5.2 La funzione setup

Si riporta di seguito la prima parte del codice della funzione *setup()*.

```
77 def setup():
78
79     global enc_val
80     global enc_count
81     global last_save_count
82
83     file1=open("/home/pi/Desktop/ConfigFiles/consumo_filo.
84               txt", "r")
85     enc_count=float(file1.readline())
86     last_save_count=enc_count
87     print(enc_count)
88     file1.close()
89     GPIO.setwarnings(False)
90     GPIO.setmode(GPIO.BCM)
91     GPIO.setup(Channel_A, GPIO.IN)
92     GPIO.setup(Channel_B, GPIO.IN)
93     GPIO.setup(Btn_Reset, GPIO.IN)
94     GPIO.setup(sftwReset, GPIO.IN)
95     GPIO.setup(fotoInt, GPIO.IN)
96     GPIO.setup(fineCorsa, GPIO.IN)
97     GPIO.setup(Led_G, GPIO.OUT)
98     GPIO.setup(errorLed, GPIO.OUT)
```

Questa parte di codice, eseguita una sola volta all'avvio del programma, serve per inizializzare le variabili globali e le varie componenti hardware. Le righe dalla 83 alla 87 servono per aprire il file di testo in cui è stato memorizzato il consumo di filo e trasferire tale valore nella variabile *enc_count*. Viene anche inizializzata la variabile *last_enc_count*, pari ad *enc_count*.

```
88     GPIO.setwarnings(False)
89     GPIO.setmode(GPIO.BCM)
90     GPIO.setup(Channel_A, GPIO.IN)
```

```

91  GPIO.setup(Channel_B, GPIO.IN)
92  GPIO.setup(Btn_Reset, GPIO.IN)
93  GPIO.setup(sftwReset, GPIO.IN)
94  GPIO.setup(fotoInt, GPIO.IN)
95  GPIO.setup(fineCorsa, GPIO.IN)
96  GPIO.setup(Led_G, GPIO.OUT)
97  GPIO.setup(errorLed, GPIO.OUT)
98  GPIO.add_event_detect(Channel_A, GPIO.BOTH, handleEnc)
99  GPIO.add_event_detect(Channel_B, GPIO.BOTH, handleEnc)
100 GPIO.add_event_detect(Btn_Reset, GPIO.FALLING, handleBtn
    )
101 GPIO.add_event_detect(fotoInt, GPIO.BOTH, handleFotoInt)
102 GPIO.add_event_detect(fineCorsa, GPIO.BOTH,
    handleFineCorsa)
103 GPIO.add_event_detect(sftwReset, GPIO.RISING, handleBtn)
104 GPIO.output(Led_G, 0)
105 GPIO.output(errorLed, 0)
106 enc_val=(GPIO.input(Channel_A)<<1)|GPIO.input(Channel_B)

```

Le righe dalla 88 alla 97 permettono di inizializzare i pin del Raspberry Pi, suddividendoli tra ingressi ed uscite. Dalla riga 98 in avanti, a ciascun pin di ingresso viene associata una interrupt service routine. L'utilizzo degli interrupt, oltre ad aumentare l'efficienza del codice, garantisce che tutti i cambiamenti di stato su tali pin siano rilevati correttamente, garanzia che non si ha impigando la tecnica polling.

Analizzando la chiamata alla funzione `GPIO.add_event_detect()`, si nota che i parametri di input sono tre: il pin da monitorare, l'evento di interesse (transitorio da 0 a 1 detto "RISING", da 1 a 0 o "FALLING", o entrambi "BOTH") e la funzione di callback, eseguita quando si verifica tale evento. Per l'encoder, gli interrupt sono due, uno per ciascun canale d'uscita, e i transitori di interesse sono entrambi, quello di salita e quello di discesa. Per il pulsante di reset e i sensori, gli interrupt sono invece singoli.

```

108  file2=open("/home/pi/Desktop/ConfigFiles/stato_sensori.
    txt", "r+")
109  text=file2.readline()
110  states=text.split(",")
111  states[0]=int(states[0]) #lunghezza tot filo
112  states[1]=int(states[1]) #stato fotointerruttore
113  states[2]=int(states[2]) #stato sensore di prox
114  flagFotoint=GPIO.input(fotoInt)
115  flagFinecorsa=GPIO.input(fineCorsa)
116  file2.seek(0)
117  file2.write('%d, %d, %d' % (states[0], flagFotoint,
    flagFinecorsa))
118  file2.truncate()
119  file2.close()

```

```
120 globalState[0]=str(flagFotoint)
121 globalState[1]=str(flagFinecorsa)
122 sleep(1)
123 GPIO.output(Led_G, 1)
124 print("".join(globalState))
```

La funzione *setup()* si chiude con l'inizializzazione delle variabili corrispondenti allo stato dei vari sensori e l'accensione di un LED che informa l'utente del passaggio alla funzione *loop()*. A questo punto il programma è pronto per iniziare a gestire gli eventi esterni come il conteggio degli impulsi dell'encoder e la stampante può quindi essere avviata.

4.5.3 Il loop

Come accennato, la funzione *loop()* contiene il codice da eseguire ciclicamente. Questo riguarda sostanzialmente l'aggiornamento dei file di testo contenenti i valori degli stati dei sensori e del consumo del filo.

```
126 def loop():
127
128     global enc_count
129     global last_save_count
130     global reset
131     global enc2m
132     global stateChangeFotoint
133     global stateChangeFinecorsa
134     global flagFotoint
135     global flagFinecorsa
136     global globalState
137
138     if reset==1:
139         enc_count=0.0
140         file1=open("/home/pi/Desktop/ConfigFiles/consumo_filo.
141         txt", "w",0)
142         file1.write('%.3f' % enc_count)
143         file1.close()
144         last_save_count=enc_count
145         print(enc_count)
146         reset=0
147
148     if stateChangeFotoint==1:
149         file2=open("/home/pi/Desktop/ConfigFiles/stato_sensori
150         .txt", "r+")
151         text=file2.readline()
152         states=text.split(",")
153         states[0]=int(states[0])
154         states[1]=int(states[1])
155         states[2]=int(states[2])
```

```

154     file2.seek(0)
155     file2.write('%d, %d, %d' % (states[0], flagFotoint,
states[2]))
156     file2.truncate()
157     file2.close()
158     globalState[0]=str(flagFotoint)
159     stateChangeFotoint=0
160     print("".join(globalState))

```

Le righe dalla 138 alla 145 consentono di azzerare il consumo del filo salvato in memoria, quando viene premuto il tasto di reset hardware o quando viene premuto il tasto di reset presente nell'interfaccia grafica. Le linee di codice dalla 147 alla 160 aggiornano invece lo stato del fotointerruttore.

```

162     if stateChangeFinecorsa==1:
163         file2=open("/home/pi/Desktop/ConfigFiles/stato_sensori
.txt", "r+")
164         text=file2.readline()
165         states=text.split(",")
166         states[0]=int(states[0])
167         states[1]=int(states[1])
168         states[2]=int(states[2])
169         file2.seek(0)
170         file2.write('%d, %d, %d' % (states[0], states[1],
flagFinecorsa))
171         file2.truncate()
172         file2.close()
173         globalState[1]=str(flagFinecorsa)
174         stateChangeFinecorsa=0
175         print("".join(globalState))
176
177     if int("".join(globalState))==11:
178         GPIO.output(errorLed, 0)
179         if abs(enc_count-last_save_count)>=2*enc2m:
180             file1=open("/home/pi/Desktop/ConfigFiles/
consumo_filo.txt", "w",0)
181             file1.write('%0.3f' % enc_count)
182             file1.close()
183             last_save_count=enc_count
184     else:
185         #spegni macchina
186         GPIO.output(errorLed,1)
187         pass

```

Analogamente a partire dalla riga 162 fino alla 175, si aggiorna lo stato del finecorsa reed. Il loop conclude con un controllo dello stato congiunto dei sensori. Se tutto funziona correttamente, `int("".join(globalState))==11`, il consumo di filo aggiornato viene salvato nel file corrispondente. In caso

contrario, si accende un LED per indicare all'utente lo spegnimento della stampante.

4.5.4 Le interrupt service routine

Le funzioni associate agli interrupt del pulsante di reset, del fotointerruttore e del finecorsa, di seguito riportate, sono tutte analoghe e contengono solo l'aggiornamento di variabili di flag che vengono poi processate nel *loop()*.

```
35 def handleBtn(pin):
36     global reset
37     global Btn_Reset
38     global sftwReset
39
40     if pin==Btn_Reset or pin==sftwReset:
41         reset=1
42     if pin==sftwReset:
43         print("SOFTWARE RESET!")
44
45 def handleFotoInt(pin):
46     global flagFotoint
47     global fotoInt
48     global stateChangeFotoint
49
50     flagFotoint=GPIO.input(fotoInt)
51     stateChangeFotoint=1
52
53 def handleFineCorsa(pin):
54     global flagFinecorsa
55     global fineCorsa
56     global stateChangeFinecorsa
57
58     flagFinecorsa=GPIO.input(fineCorsa)
59     stateChangeFinecorsa=1
```

La funzione di callback che gestisce i segnali in ingresso dall'encoder merita invece un'analisi più attenta.

```
61 def handleEnc(pin):
62     global enc_val
63     global enc_count
64     global enc2m
65     global globalState
66
67     enc_dict=[0, -1, 1, 0, 1, 0, 0, -1, -1, 0, 0, 1, 0, 1,
68             -1, 0]
69
70     enc_val=enc_val<<2
71     val=np.uint8(0)
```

CAPITOLO 4

```

71 val=(GPIO.input(Channel_A)<<1)|GPIO.input(Channel_B)
72 enc_val=(enc_val | val)&0b1111
73 if int("".join(globalState))==11:
74     enc_count=enc_count+enc_dict[enc_val]*enc2m
75     print(enc_count)

```

Dato l'elevato numero di impulsi al secondo in arrivo dall'encoder, è importante che il loro conteggio avvenga in modo efficiente. Come evidenziato nelle sezioni precedenti infatti, il mancato conteggio di alcuni di essi porterebbe ad una valutazione erronea del consumo di filo con le conseguenze discusse. Dati l'istante t_1 di esecuzione della funzione *handleEnc* e l'istante t_0 corrispondente all'esecuzione precedente, la strategia adottata consiste nel codificare lo stato dei due ingressi a t_1 e t_0 in una stringa binaria e poi di usare la stessa come indice per determinare se incrementare il consumo di filo.

La generazione del vettore `enc_dict` parte dall'analisi del funzionamento dell'encoder stesso. Dati due canali in quadratura, le permutazioni possibili, generate combinando lo stato attuale e quello precedente delle uscite, sono 16. Ad ognuna di queste, come mostrato in Tabella 4.7, viene associato un valore +1, -1 o 0.

Number	$t = t_0$		$t = t_1$		Binary Number	Direction
	A	B	A	B		
0	0	0	0	0	0000	0
1	0	0	0	1	0001	-1
2	0	0	1	0	0010	+1
3	0	0	1	1	0011	0
4	0	1	0	0	0100	+1
5	0	1	0	1	0101	0
6	0	1	1	0	0110	0
7	0	1	1	1	0111	-1
8	1	0	0	0	1000	-1
9	1	0	0	1	1001	0
10	1	0	1	0	1010	0
11	1	0	1	1	1011	+1
12	1	1	0	0	1100	0
13	1	1	0	1	1101	+1
14	1	1	1	0	1110	-1
15	1	1	1	1	1111	0

Tabella 4.7: Tabella riassuntiva degli stati dei canali dell'encoder e corrispondente valore per il calcolo del consumo di filo.

Il valore +1 corrisponde ad un verso di rotazione, orario o antiorario dell'encoder (informazione deperibile dal datasheet del componente), il -1 al verso di rotazione opposto, mentre il valore 0 è associato alla permanenza delle uscite nello stesso stato, o alle combinazioni fisicamente impossibili perchè richiederebbero il cambiamento contemporaneo dello stato di entrambi gli ingressi. Si prenda ad esempio la riga 0 della tabella. I due bit più significativi e i due meno significativi, rappresentanti rispettivamente lo stato degli ingressi a t_0 e a t_1 , sono uguali; questo comporta che non vi sia stata alcuna rotazione dell'albero dell'encoder tra i due istanti di tempo. Analogamente alla riga 4 si trova il codice 0011, che corrisponde ad un passaggio contemporaneo di entrambi gli ingressi da 0 a 1 tra gli istanti t_0 e t_1 , situazione impossibile se le uscite dell'encoder sono in quadratura. Va evidenziato come questi casi siano stati contemplati non per ragioni di completezza, ma per motivi funzionali. Si può infatti notare che la Tabella 4.7 non è stata compilata in ordine casuale, ma bensì in ordine crescente dei codici binari della quarta colonna. Il primo codice 0000 corrisponde al valore decimale 0, il secondo codice 0001 al valore 1, il terzo al valore 2 e via via fino al codice 1111, l'ultimo, coincidente con il valore decimale 15. Il vettore `enc_dict` raccoglie in ordine i valori della quinta colonna. Nella sua definizione, non potendo lasciare dei campi vuoti, si assegna un valore invariante di 0 alle situazioni descritte in precedenza. Analizzando ora il codice più nel dettaglio è possibile capire la ragione di tale scelta.

Alla riga 69, i bit della variabile `enc_val`, che contiene lo stato degli ingressi all'istante precedente, vengono shiftati verso sinistra di due posizioni, per fare spazio ai due bit che descrivono lo stato attuale. Questi vengono salvati in `enc_val` alla riga 72. Da notare come l'operazione di bitwise OR tra `enc_val` e `val`, consenta di preservare lo stato dei bit più significativi di `enc_val`, e di copiare nelle due posizioni meno significative il valore di `val`. Il successivo bitwise AND serve a mascherare i quattro bit più significativi di `enc_val`, variabile a otto bit. Si desidera infatti ottenere una stringa binaria di quattro bit, mentre i quattro bit più significativi contengono gli stati degli ingressi nei momenti precedenti t_{-1} e t_{-2} , di alcun interesse.

Alla riga 74, si aggiorna quindi il consumo di filo mediante l'operazione:

```
74 enc_count=enc_count+enc_dict[enc_val]*enc2m
```

Alla variabile `enc_count` viene sommato l'elemento di `enc_dict` corrispondente al codice binario salvato in `enc_val`, moltiplicato per la costante `enc2m`, che definisce il consumo di filo in metri associato ad ogni impulso dell'encoder. Il risultato dell'operazione viene infine sovrascritto in `enc_count`.

L'aver tenuto in considerazione anche i casi a cui è associato un valore 0

permette quindi di rendere la procedura automatizzata ed efficiente, a costo di un piccolissimo spreco in termini di memoria.

4.5.5 L'interfaccia grafica

L'interfaccia grafica sviluppata ha lo scopo di mostrare in modo chiaro lo stato dei sensori e di informare l'utente sul consumo del filo. Tale interfaccia è stata realizzata in linguaggio PHP/HTML e può essere visualizzata in qualunque browser. Tale scelta è stata fatta per garantire la possibilità di accedervi in remoto, connettendo il Raspberry Pi ad una rete internet.

La Figura 4.25 mostra la pagina principale che riporta lo stato dei sensori, il consumo di filo in m e la percentuale di filo rimanente.



Figura 4.25: Schermata principale dell'interfaccia grafica, riportante lo stato dei sensori e le informazioni sul consumo di filo.

Le Figure 4.26 e 4.27 mostrano invece la stessa pagina nel caso esaurimento o di bloccaggio del filo.

Stato sensori stampante 3D

FILO CONSUMATO	0.48 m
FILO RIMANENTE	52.00%
PRESENZA FILO	ALLARME
FILO BLOCCATO	OK
SETTINGS	

Figura 4.26: Segnalazione dell'allarme nel caso dell'esaurimento del filo.

Stato sensori stampante 3D

FILO CONSUMATO	0.0 m
FILO RIMANENTE	100.00%
PRESENZA FILO	OK
FILO BLOCCATO	ALLARME
SETTINGS	

Figura 4.27: Segnalazione dell'allarme nel caso del bloccaggio del filo.

L'ultima schermata, mostrata in Figura 4.28, è quella che appare cliccando sul pulsante SETTINGS. Come visibile, in questa pagina è possibile resettare i dati riguardanti il consumo del filo e modificare la lunghezza iniziale, nel caso in cui si sia sostituita la bobina.

Stato sensori stampante 3D

Parametri
×

Lunghezza bobina:

 m

Reset encoder

Figura 4.28: Form per l'aggiornamento della lunghezza della bobina e l'azzeramento del contatore associato alla quantità di filo consumata.

Appendice A

A.1 Anatomia del cranio

"Il cranio è formato da ventidue ossa, esclusi gli ossicini delle orecchie. Eccetto per la mandibola, le ossa del cranio sono collegate tra di loro mediante suture, le quali sono immobili."[20]

Di seguito sono riportate la vista frontale e quella laterale del cranio, con indicati i nomi, alcuni dei quali ripresi nella trattazione dell'elaborato, delle ossa e delle varie caratteristiche anatomiche del cranio.

Vista Frontale

"La vista anteriore del capo comprende la fronte superiormente e inferiormente le orbite, la regione nasale, la parte del volto tra le orbite e la mascella, la mascella e la mandibola." [20]

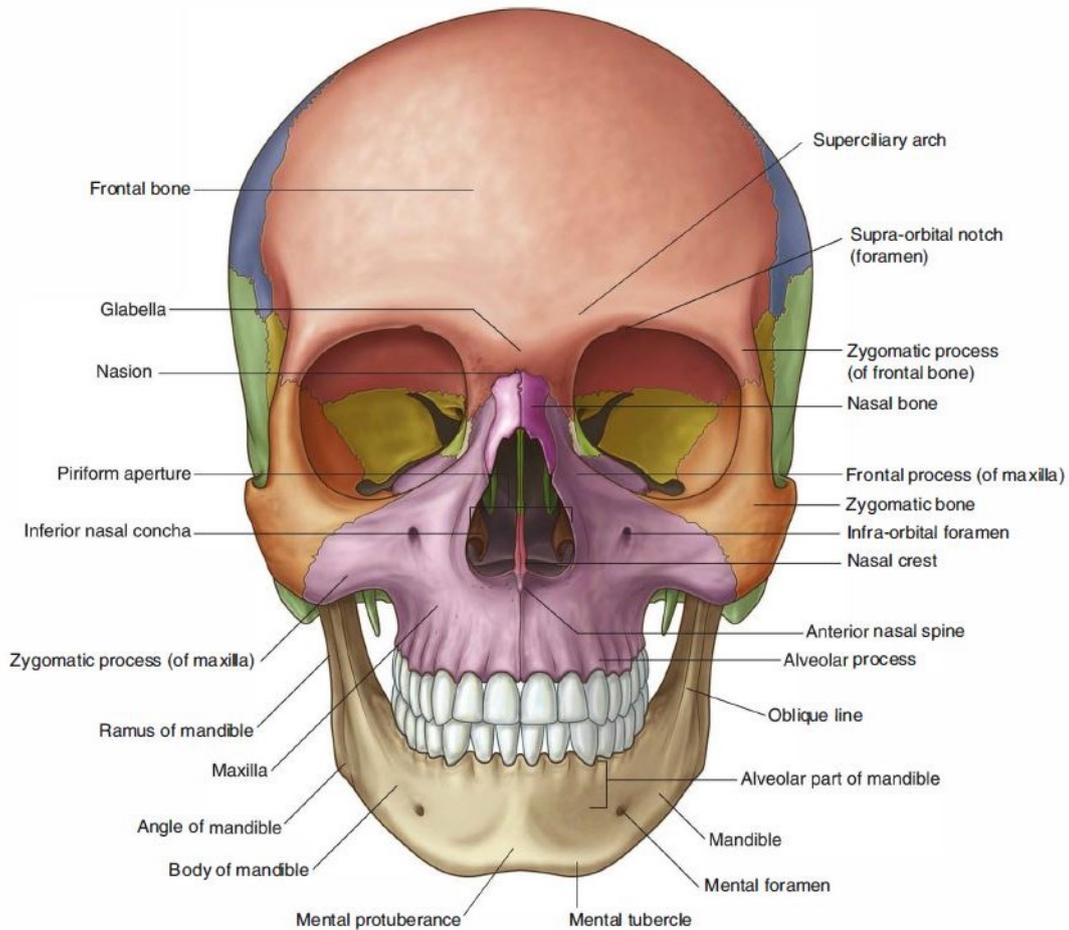


Figura A.1: Vista frontale del cranio[20].

Vista laterale

"La vista laterale del capo consiste nella parte laterale del cranio che comprende le porzioni laterali della volta cranica e del massiccio facciale e metà della mandibola."[20]

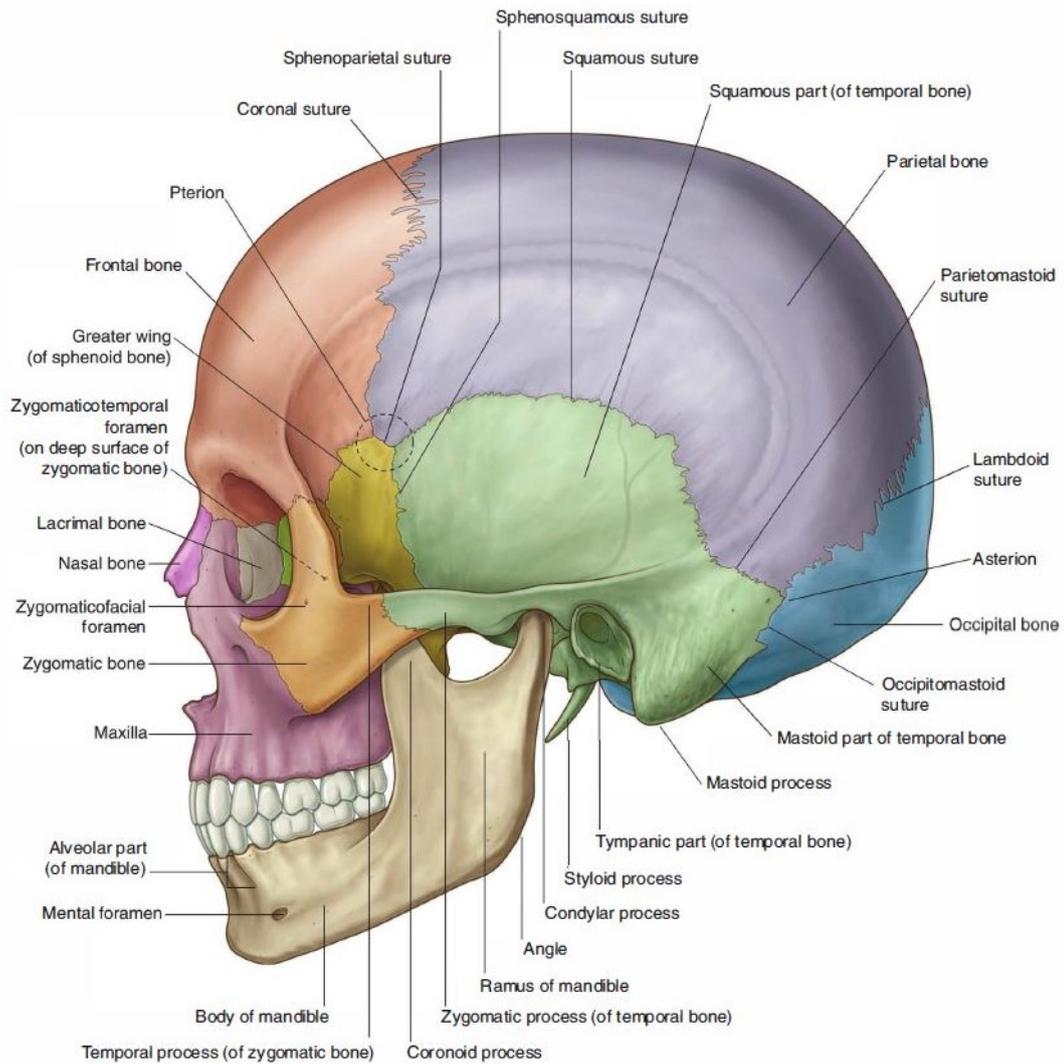


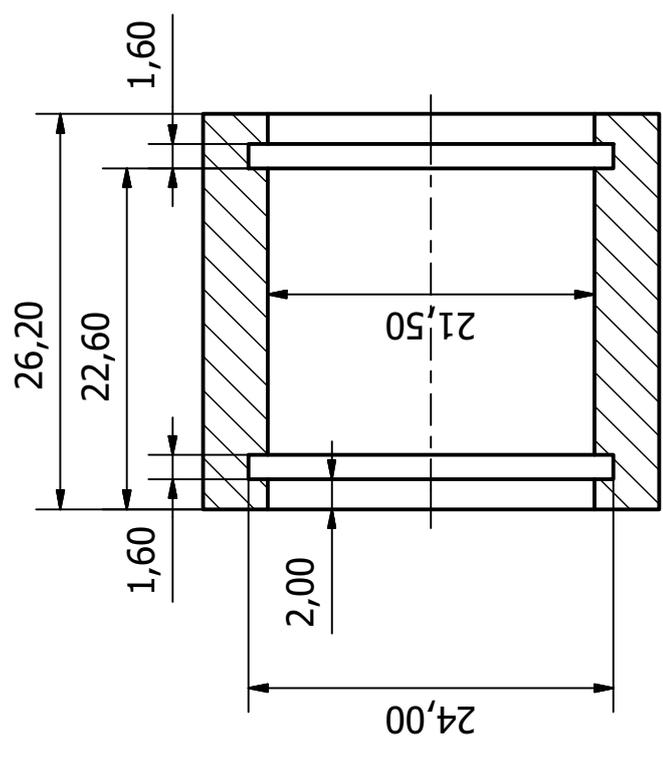
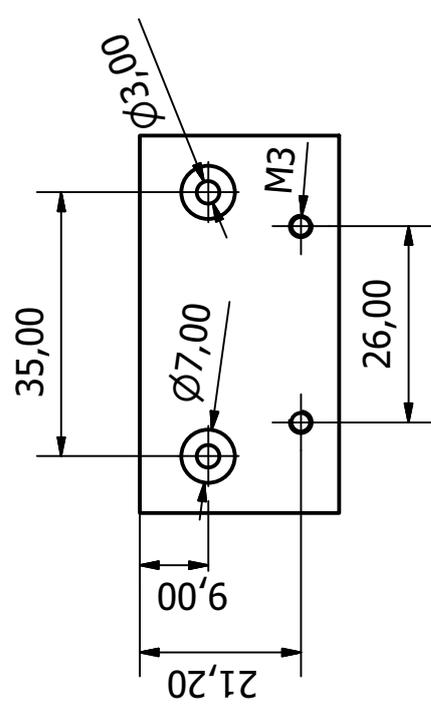
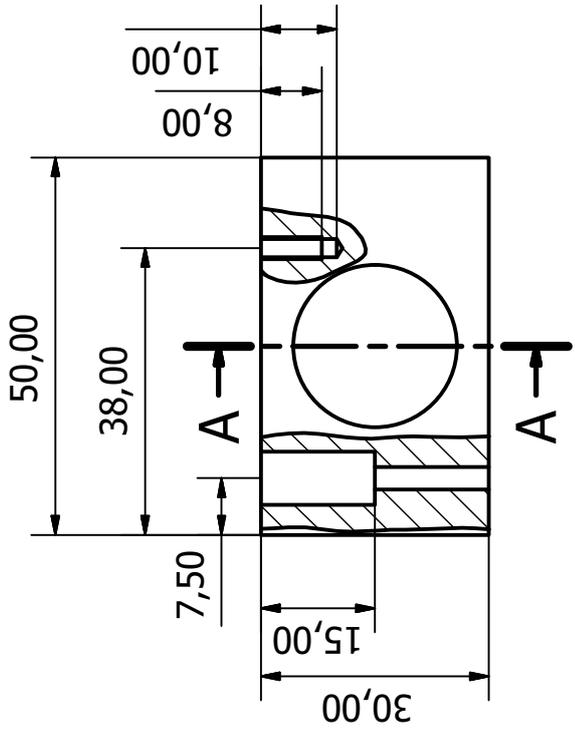
Figura A.2: Vista laterale del cranio[20].

Appendice B

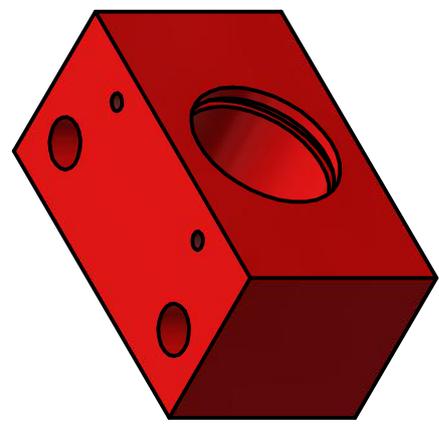
B.1 Disegni Tecnici

Nelle pagine seguenti vengono allegati i disegni tecnici utilizzati per la produzione dei componenti progettati.

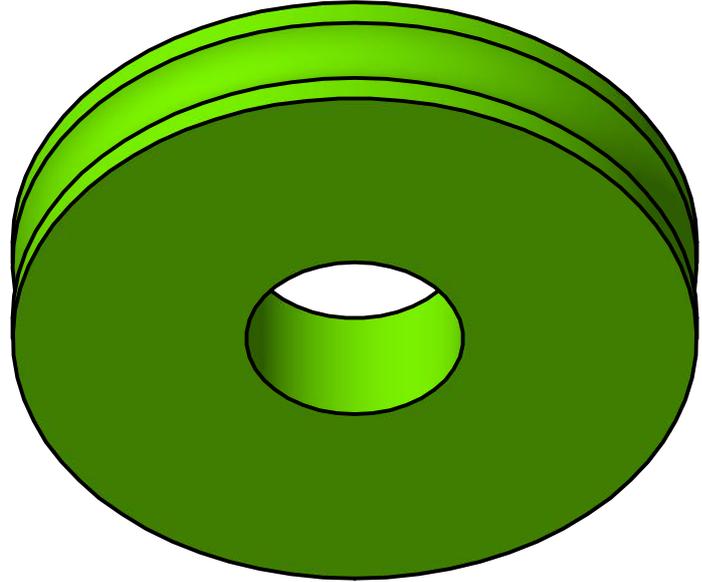
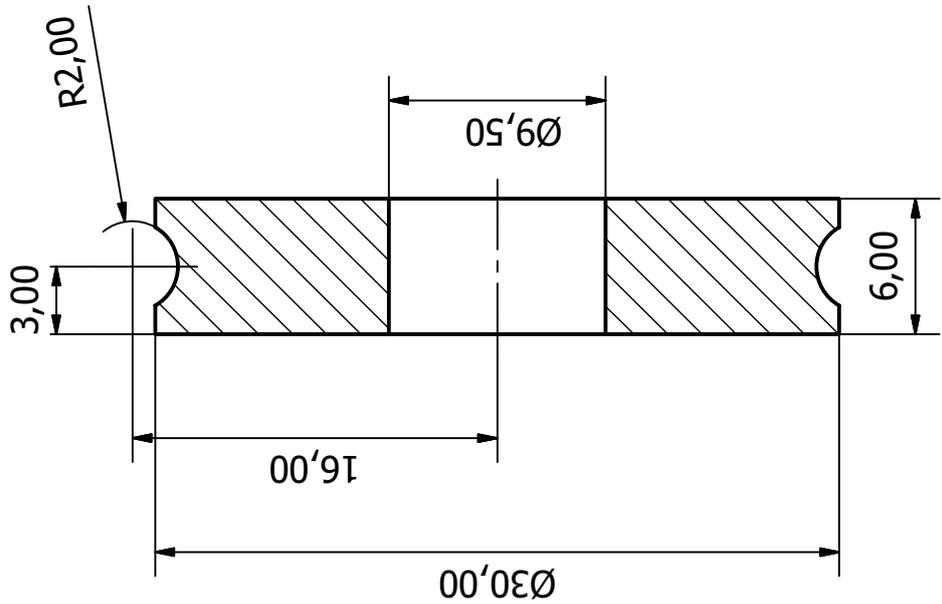
B.1.1 Sistema per la misura del consumo del filamento



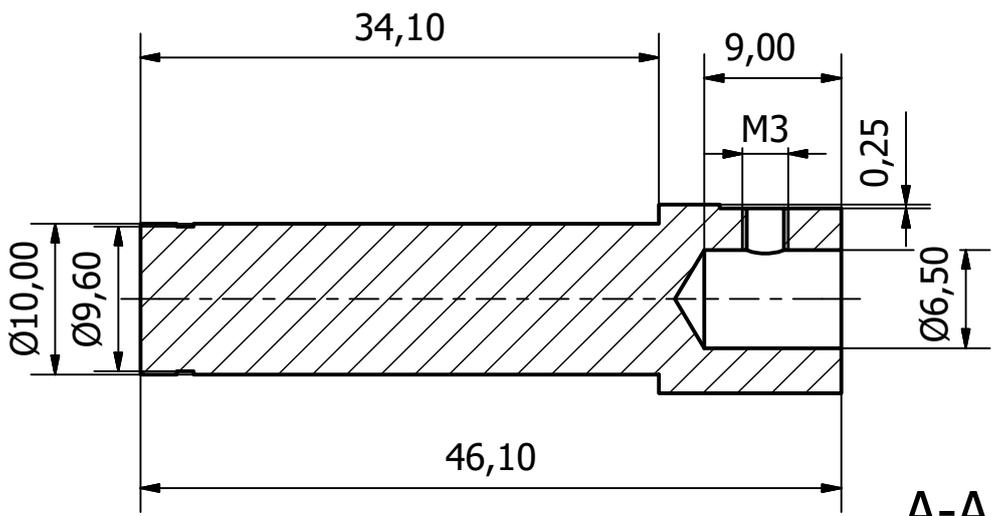
A-A
SCALE 2 : 1



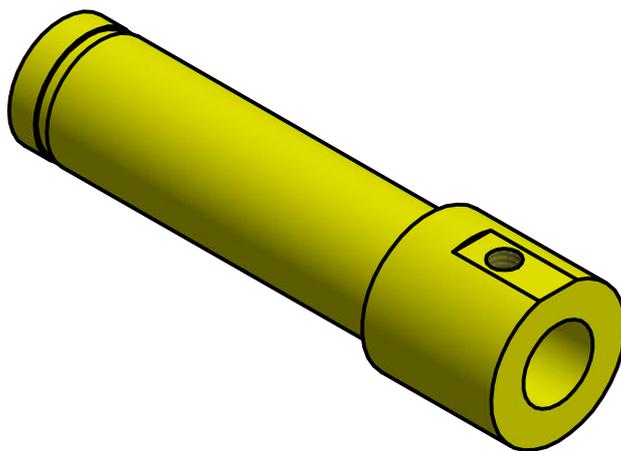
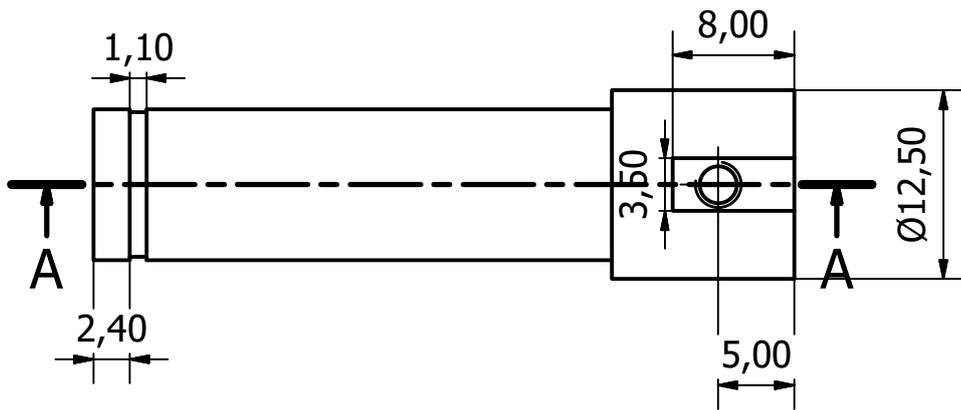
Designed by Cecilia	Checked by	Approved by	Date 17/03/2020	Sheet 1 / 1
Supporto_Superiore			Scale 1:1	



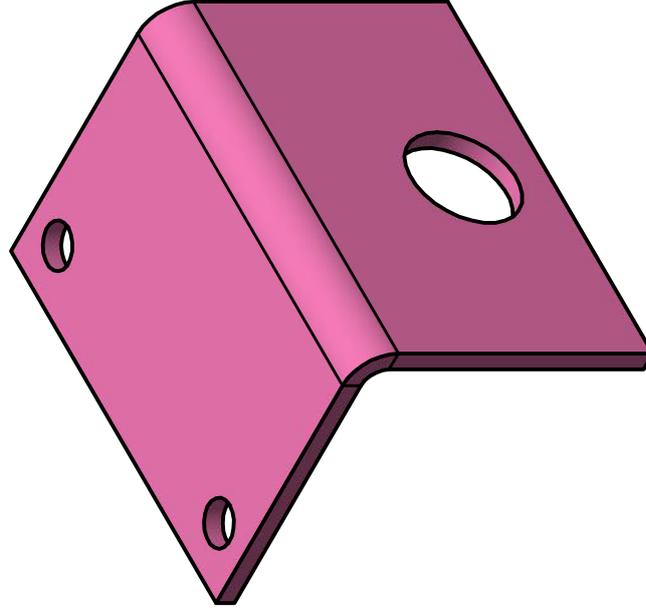
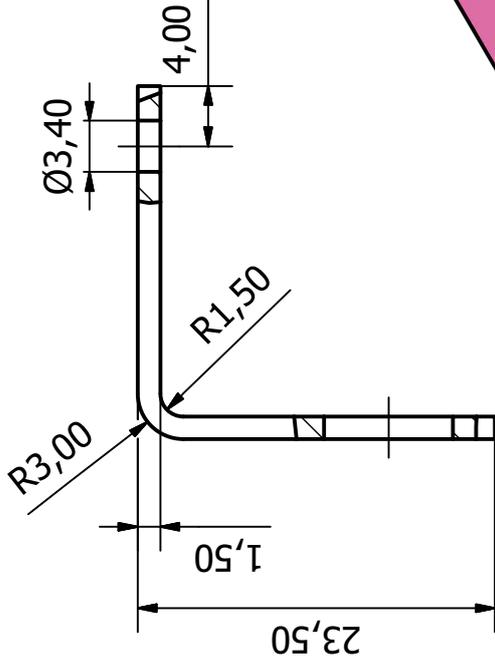
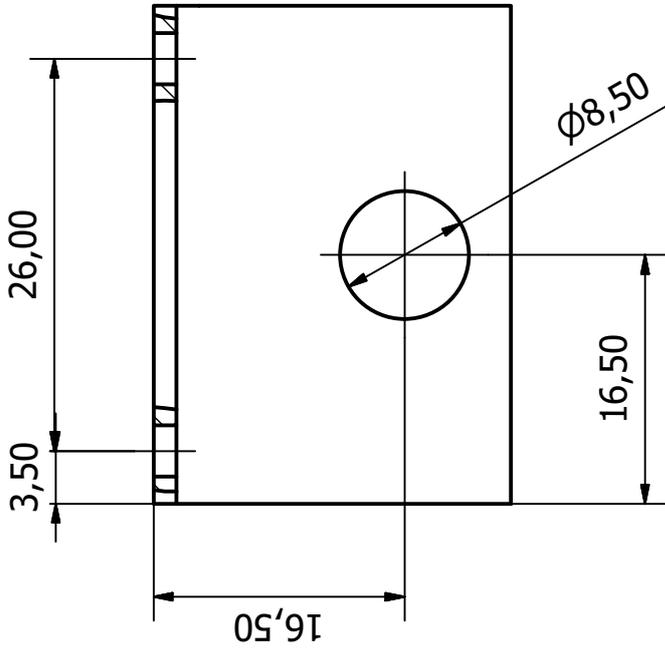
Designed by Cecilia	Checked by	Approved by	Date	Date	17/03/2020
Scale 3:1			Edition		
			Ruota		
			Sheet 1 / 1		



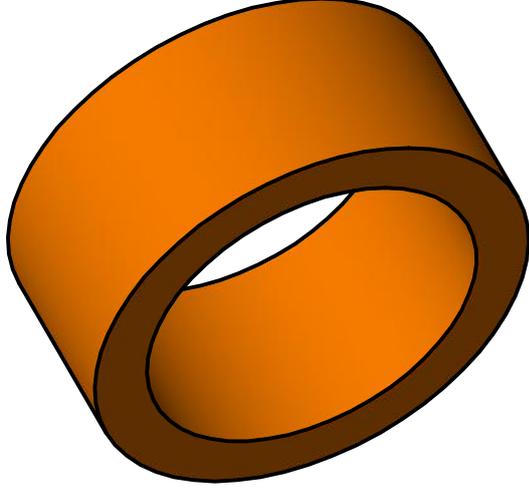
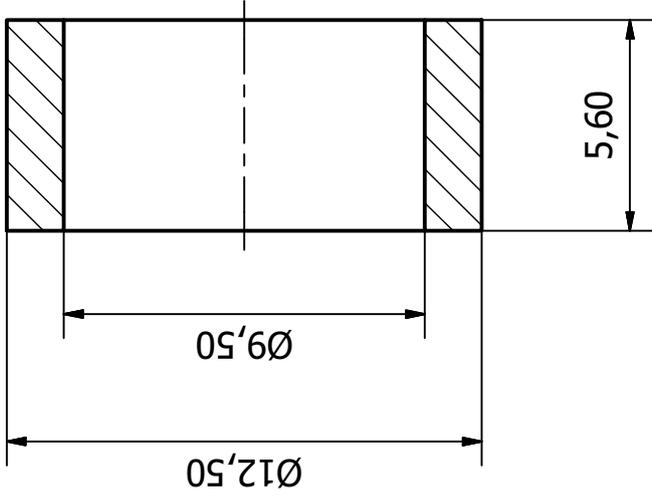
A-A
SCALE 2 : 1



Designed by Cecilia	Checked by	Approved by	Date	Date 17/03/2020
			Scale 2:1	
Albero			Edition	Sheet 1 / 1



Designed by Cecilia	Checked by	Approved by	Date	Date	17/03/2020
Supporto_L			Scale 2:1		
			Edition	Sheet	
			1 / 1		



Designed by
Cecilia

Checked by

Approved by

Date

Date

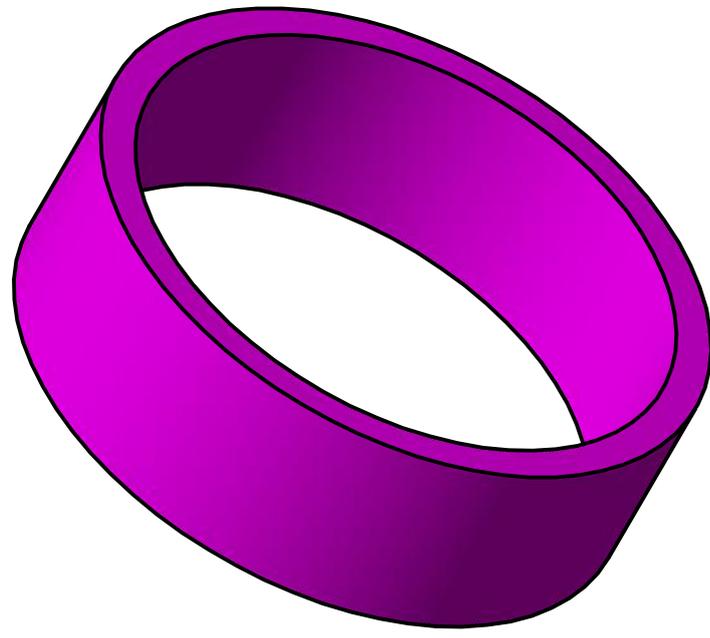
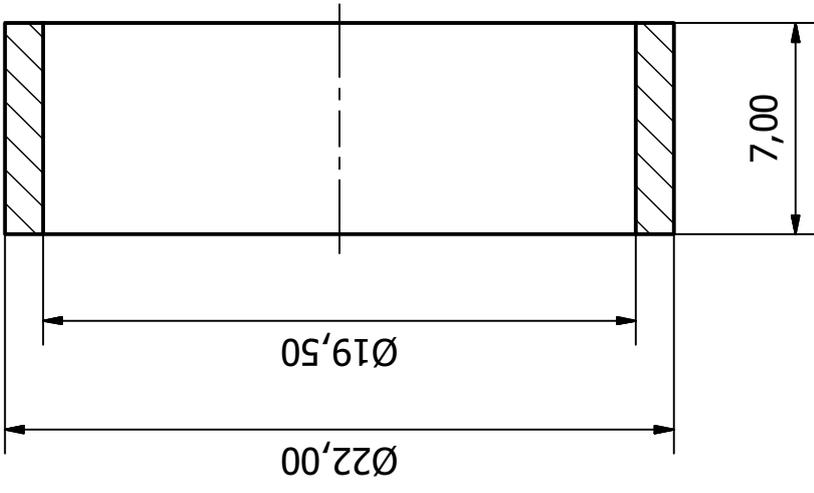
17/03/2020

Scale 5:1

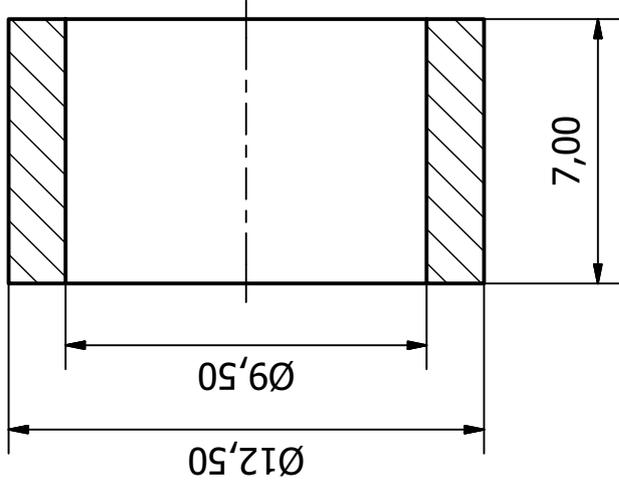
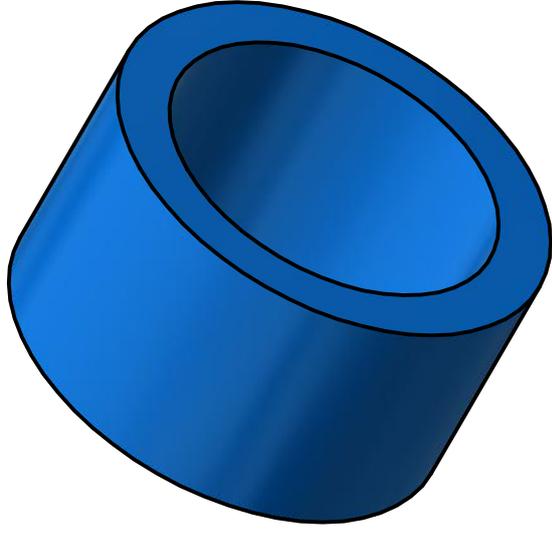
Distanziale_RuotaCuscinetto

Edition

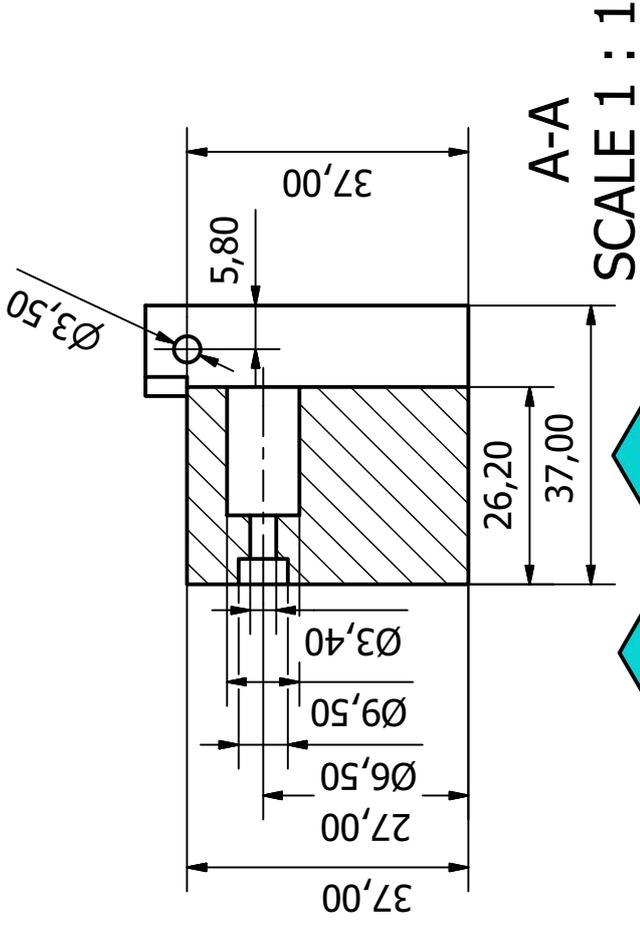
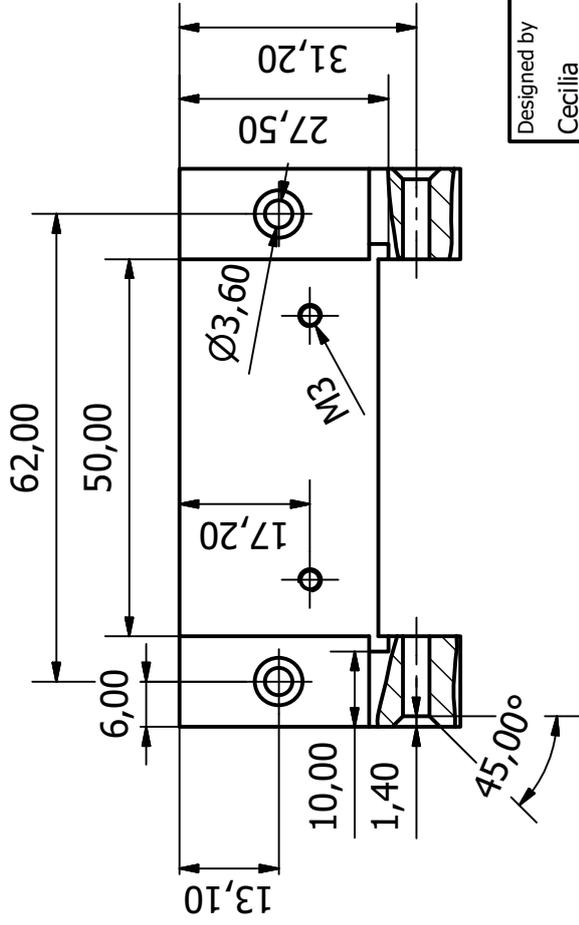
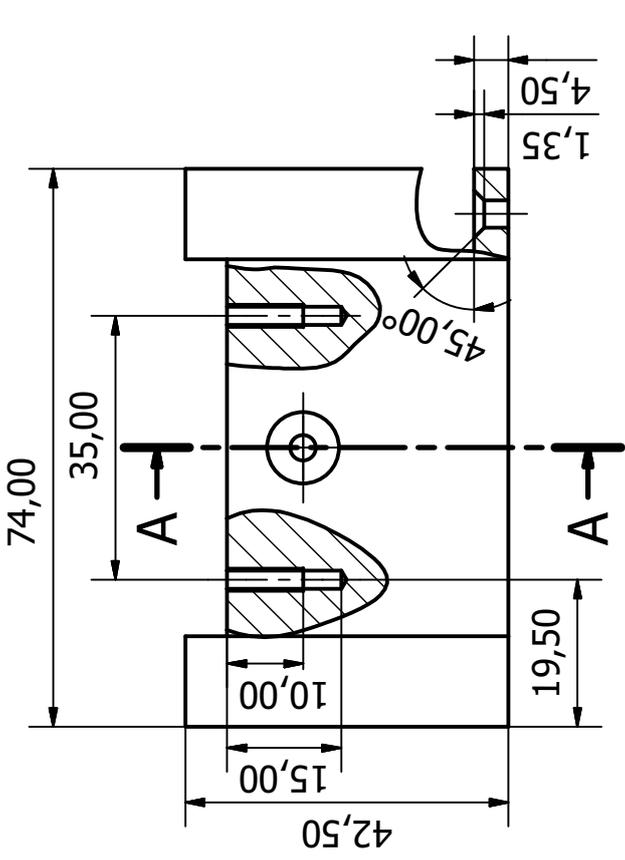
Sheet
1 / 1



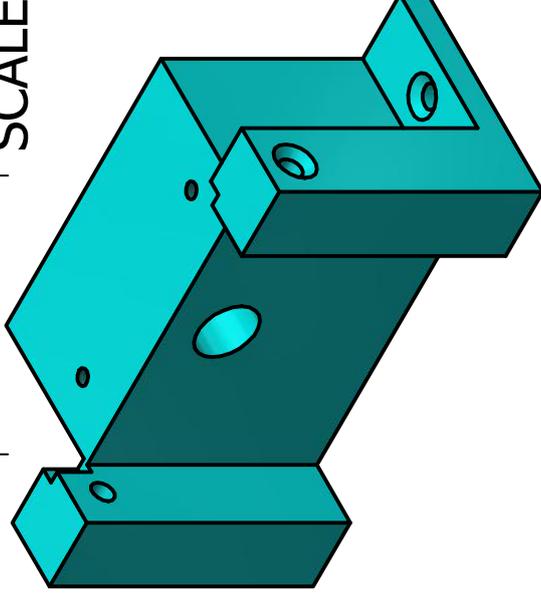
Designed by Cecilia	Checked by	Approved by	Date	Date	17/03/2020
Scale 4:1			Edition		
			Distanziale_Esterno		
					Sheet 1 / 1



Designed by Cecilia	Checked by	Approved by	Date	Date	17/03/2020
Scale 5:1			Distanziale_Interno		
			Edition		
			Sheet 1 / 1		



A-A
SCALE 1 : 1



Designed by
Cecilia

Checked by

Approved by

Date

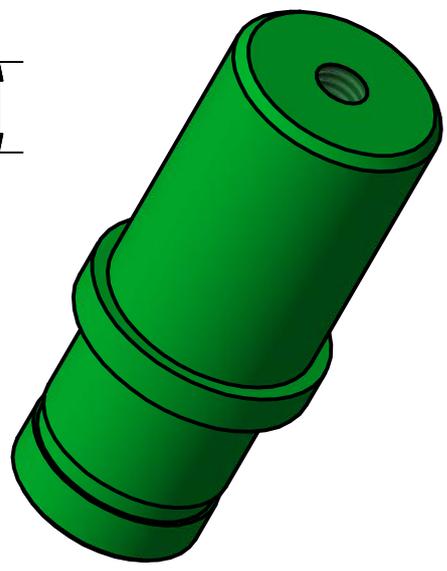
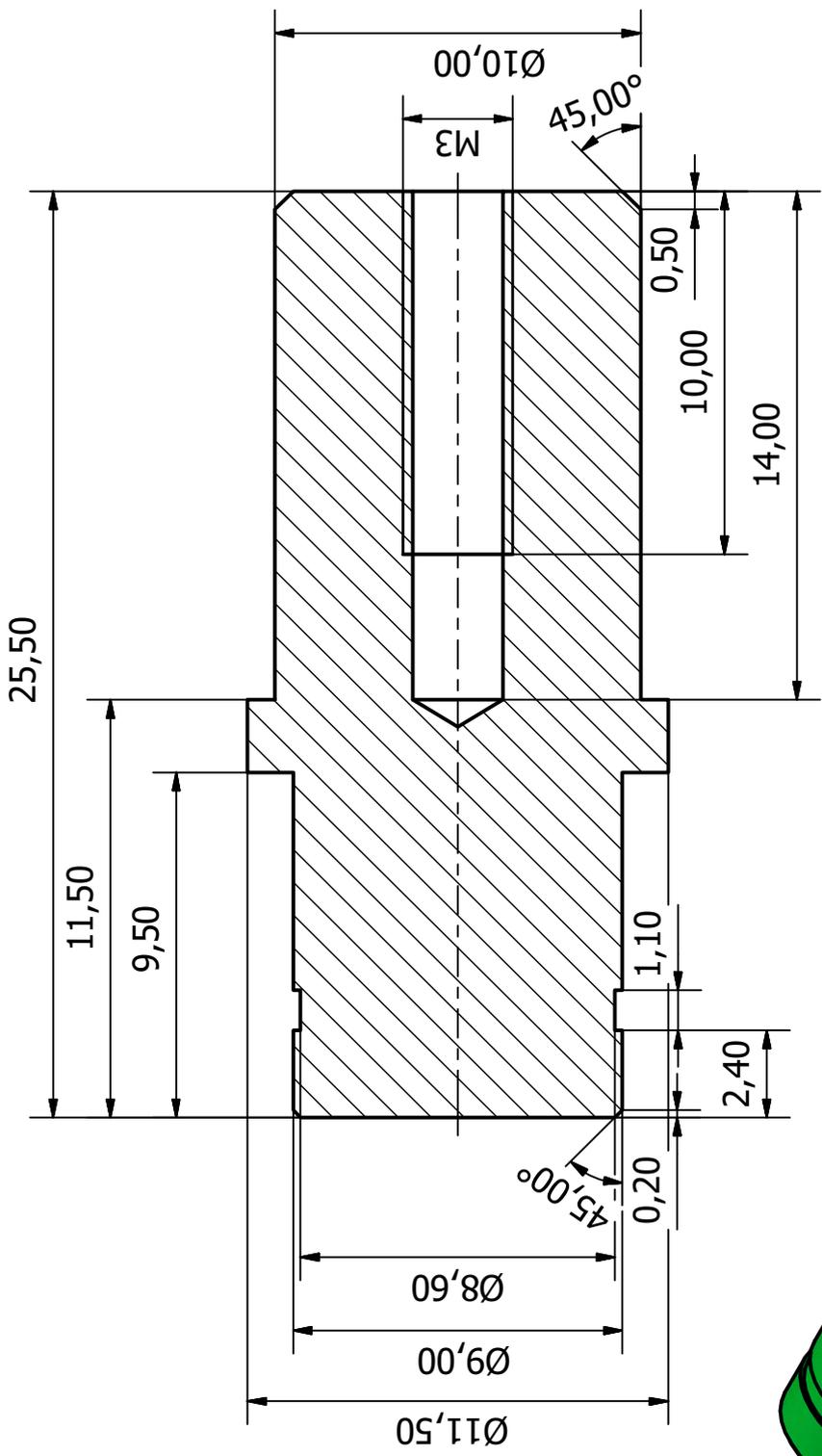
17/03/2020

Scale 1:1

Supporto_Inferiore

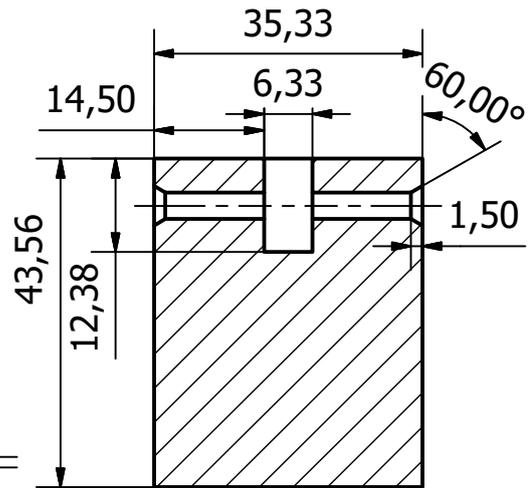
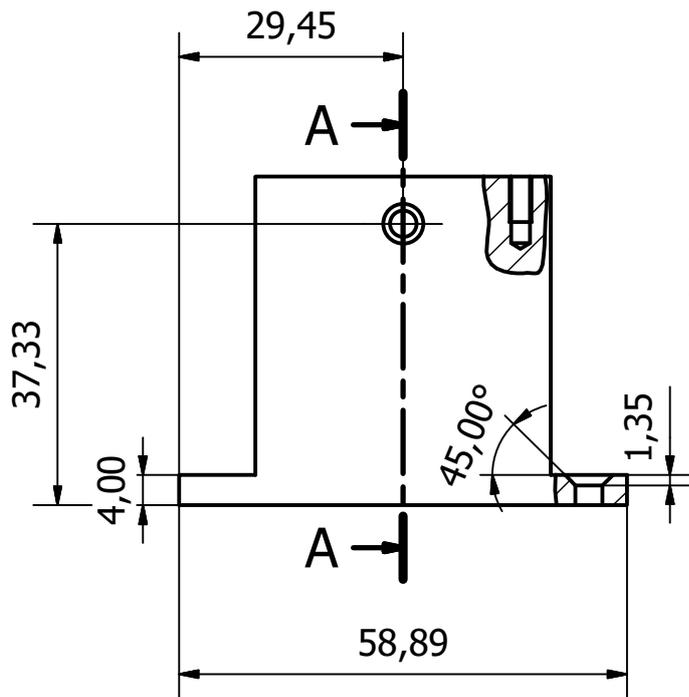
Edition

Sheet
1 / 1

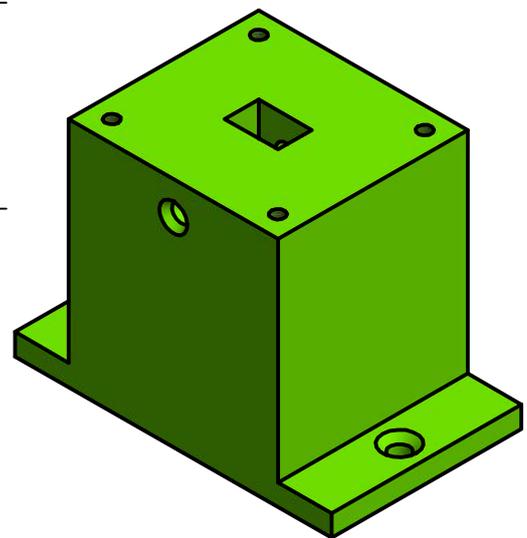
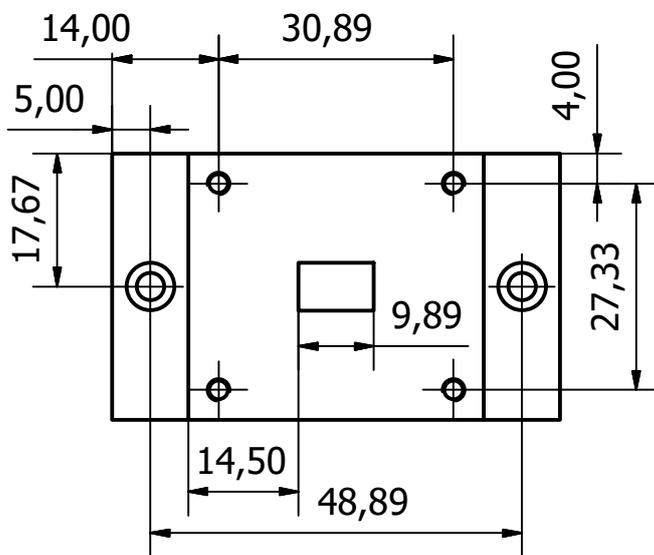


Designed by Cecilia	Checked by	Approved by	Date 17/03/2020
Scale 5:1			Sheet 1 / 1
Perno			Edition

B.1.2 Sistema di rilevazione dell'esaurimento del filo

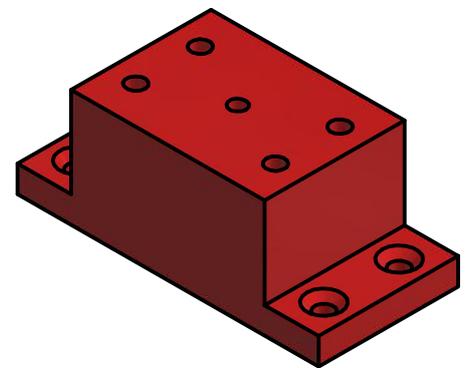
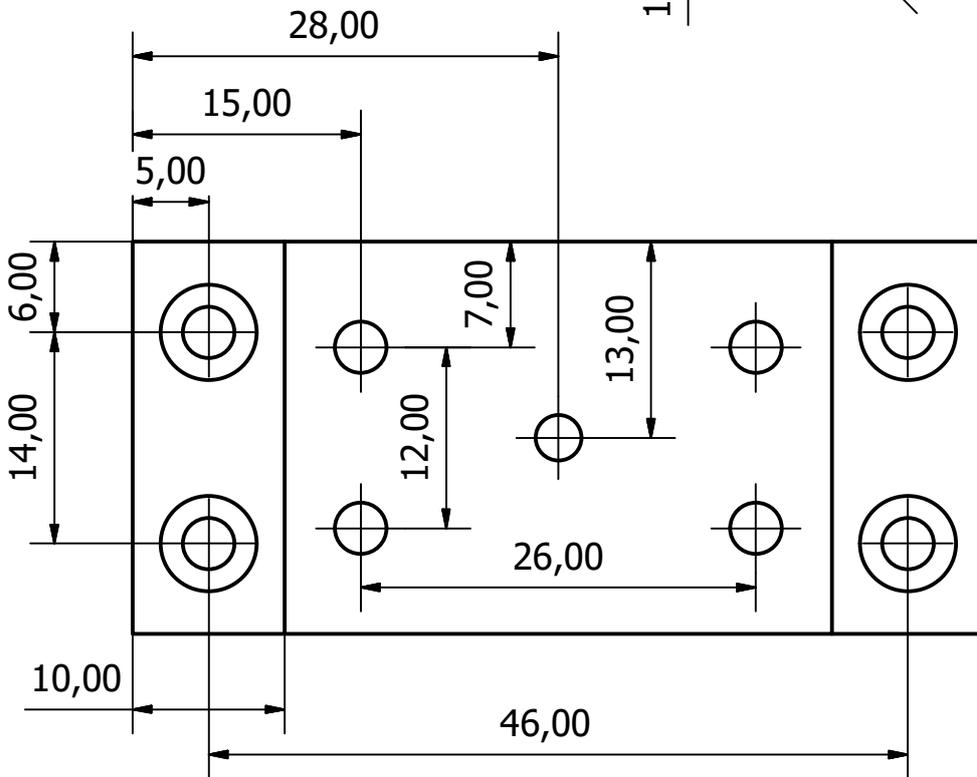
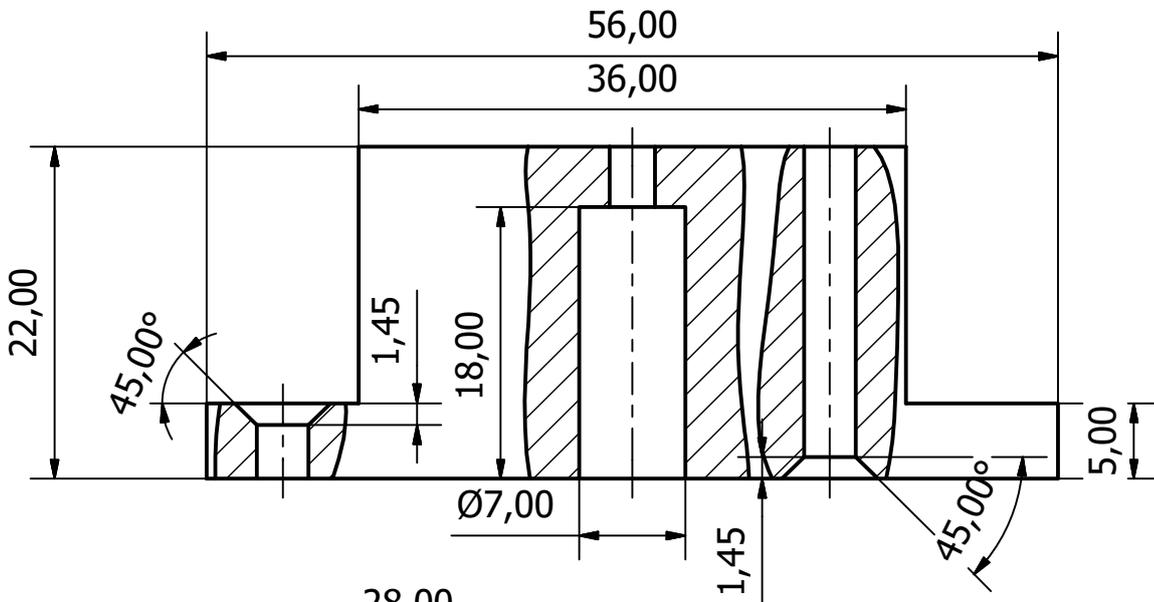


A-A
SCALE 1 : 1

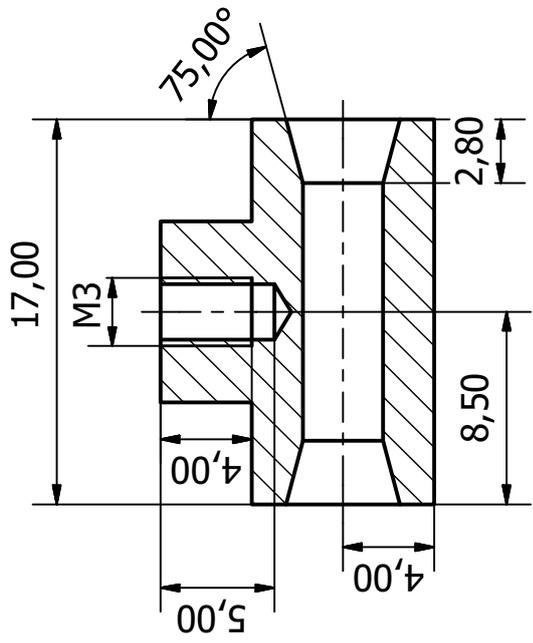
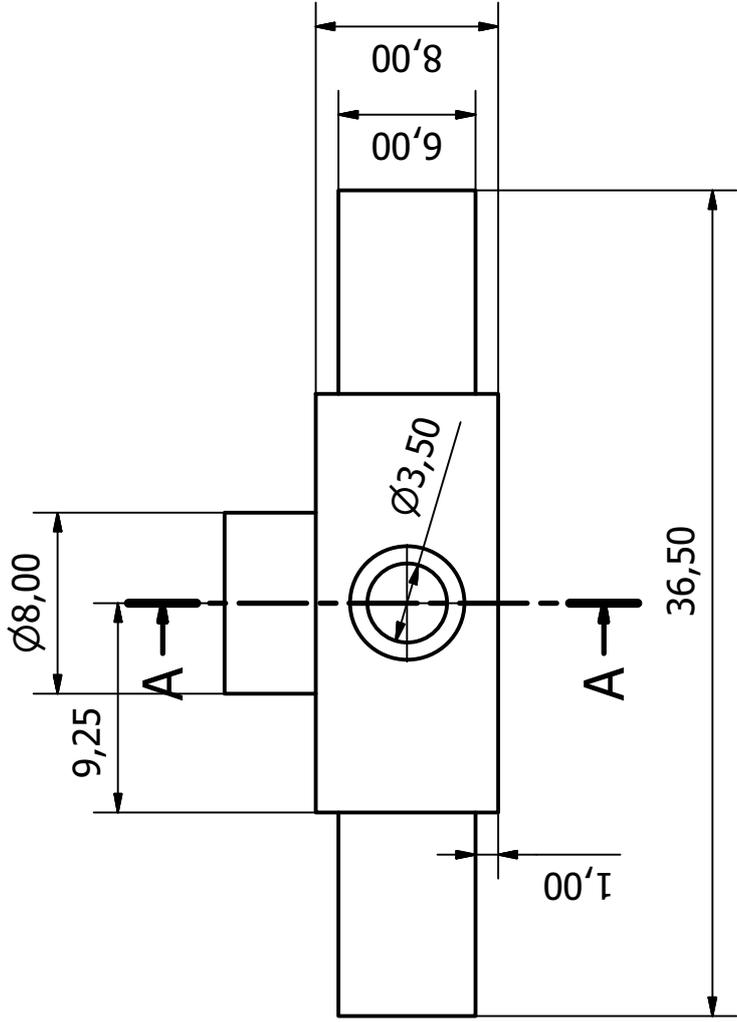


Designed by Cecilia	Checked by	Approved by	Date	Date 17/03/2020
			Scale 1:1	
			Supporto_Fotointerruttore	Edition Sheet 1 / 1

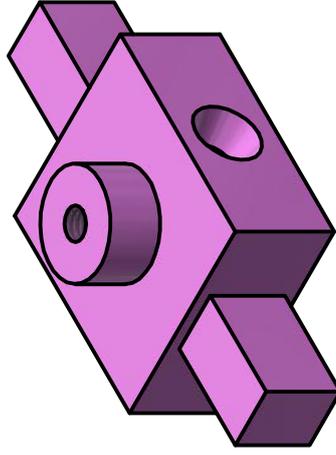
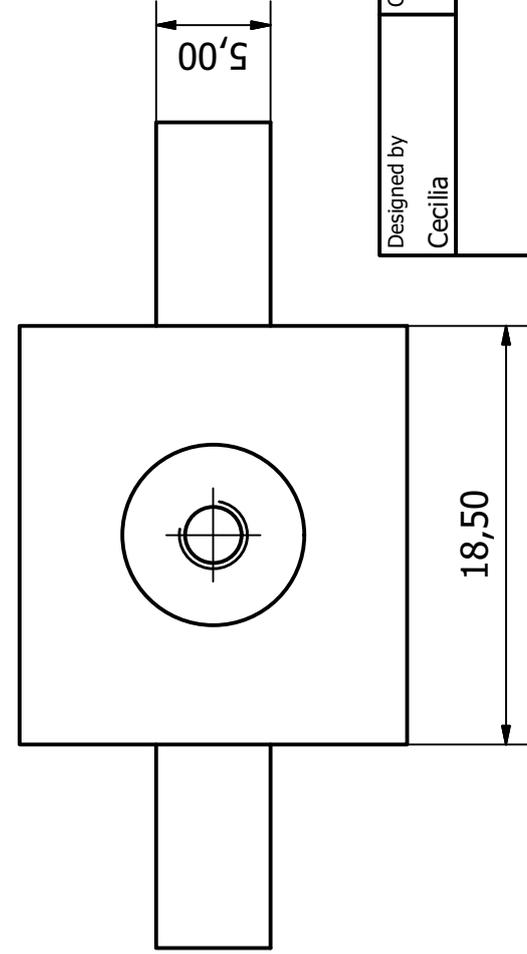
B.1.3 Dispositivo per il rilevamento del bloccaggio del
filo



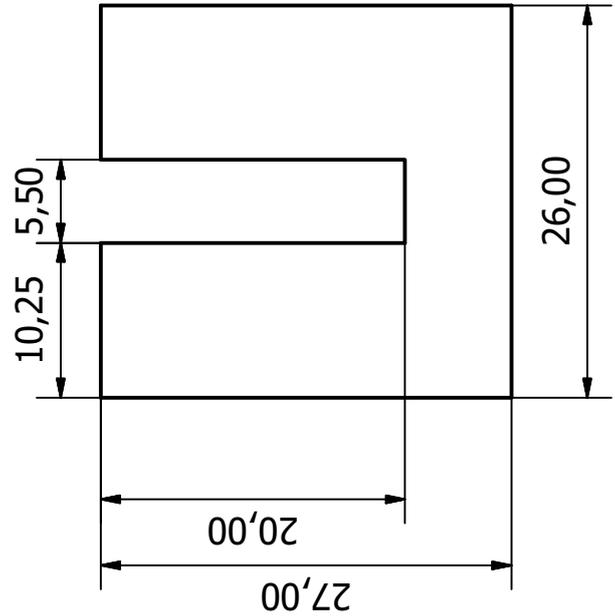
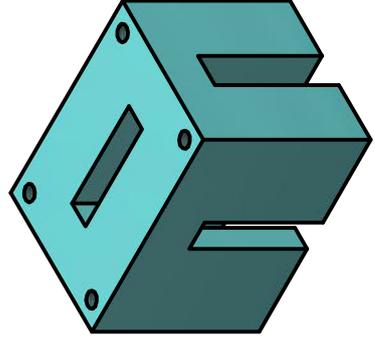
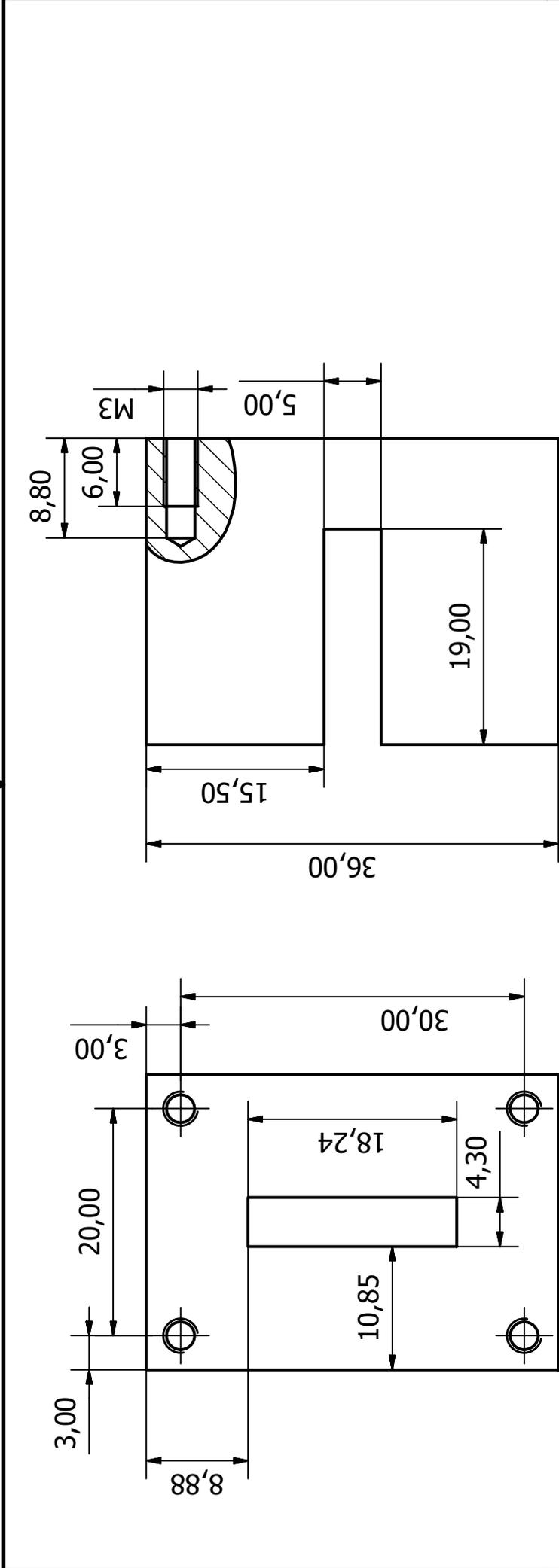
Designed by Cecilia	Checked by	Approved by	Date	Date 17/03/2020
			Scale 2:1	
Base			Edition	Sheet 1 / 1



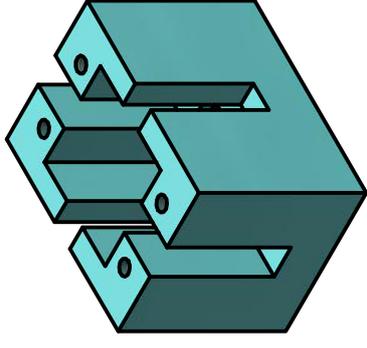
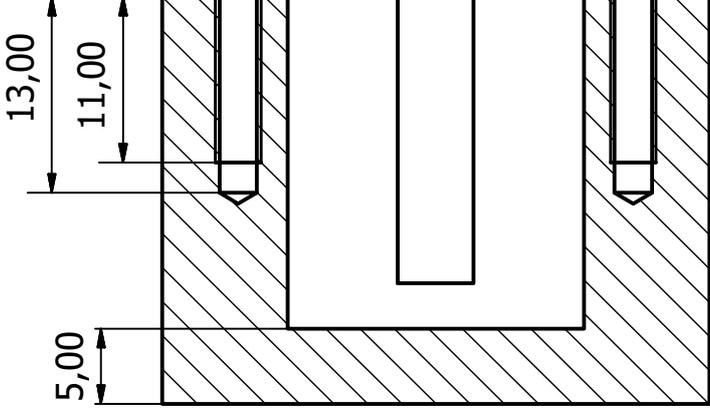
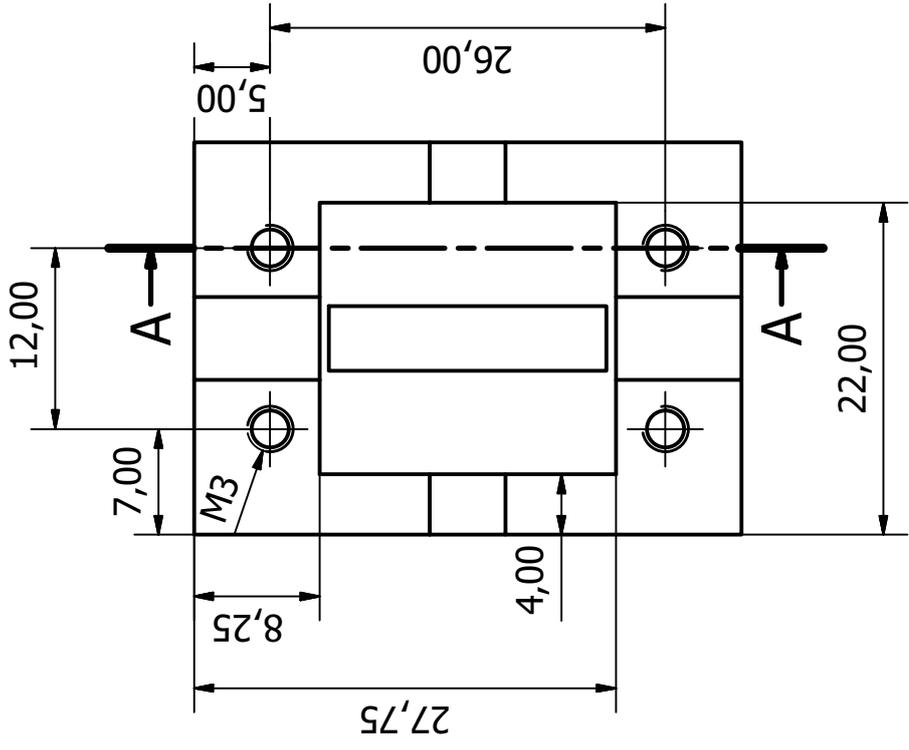
A-A
SCALE 3 : 1



Designed by Cecilia	Checked by	Approved by	Date 17/03/2020
Scale 3:1			Sheet 1 / 1
Corpo_Centrale_Mobile			



Designed by Cecilia	Checked by	Approved by	Date 17/03/2020	Sheet 1 / 2
Scale 2:1			Edition	
Coperchio_Superiore				



A-A
SCALE 2 : 1

Designed by Cecilia	Checked by	Approved by	Date	Date	17/03/2020
Scale 2:1			Edition		
			Coperchio_Superiore		
			Sheet		
			2 / 2		

Appendice C

C.1 Programma per la lettura dello stato dei sensori

Si riporta di seguito il listato Python scritto ed utilizzato per la lettura dello stato dei sensori nel Capitolo 4.

```
1 import RPi.GPIO as GPIO
2 import numpy as np
3 from time import sleep
4
5
6 Channel_A=8
7 Channel_B=10
8
9 Btn_Reset=19 #Bottone Giallo
10 fotoInt=6 #Bottone Verde
11 fineCorsa=18
12 sftwReset=23
13 Led_G=12
14 errorLed=21 #Led_R
15
16 enc_val=np.uint8(0)
17 enc_count=np.float32(0)
18 last_save_count=np.float32(0)
19
20 flagFotoint=1
21 flagFinecorsa=1
22
23 reset=0
24 stateChangeFotoint=0
25 stateChangeFinecorsa=0
26 enc2m=0.0029 #l=pi*D
27
28 globalState=['0','0']
29
```

```
30 def main():
31     setup()
32     while 1:
33         loop()
34
35 def handleBtn(pin):
36     global reset
37     global Btn_Reset
38     global sftwReset
39
40     if pin==Btn_Reset or pin==sftwReset:
41         reset=1
42     if pin==sftwReset:
43         print("SOFTWARE RESET!")
44
45 def handleFotoInt(pin):
46     global flagFotoint
47     global fotoInt
48     global stateChangeFotoint
49
50     flagFotoint=GPIO.input(fotoInt)
51     stateChangeFotoint=1
52
53 def handleFineCorsa(pin):
54     global flagFinecorsa
55     global fineCorsa
56     global stateChangeFinecorsa
57
58     flagFinecorsa=GPIO.input(fineCorsa)
59     stateChangeFinecorsa=1
60
61 def handleEnc(pin):
62     global enc_val
63     global enc_count
64     global enc2m
65     global globalState
66
67     enc_dict=[0, -1, 1, 0, 1, 0, 0, -1, -1, 0, 0, 1, 0, 1,
68              -1, 0]
69
70     enc_val=enc_val<<2
71     val=np.uint8(0)
72     val=(GPIO.input(Channel_A)<<1)|GPIO.input(Channel_B)
73     enc_val=(enc_val | val)&0b1111
74     if int("".join(globalState))==11:
75         enc_count=enc_count+enc_dict[enc_val]*enc2m
76         print(enc_count)
77
78 def setup():
```

CAPITOLO C

```
78
79 global enc_val
80 global enc_count
81 global last_save_count
82
83 file1=open("/home/pi/Desktop/ConfigFiles/consumo_filo.
      txt", "r")
84 enc_count=float(file1.readline())
85 last_save_count=enc_count
86 print(enc_count)
87 file1.close()
88 GPIO.setwarnings(False)
89 GPIO.setmode(GPIO.BCM)
90 GPIO.setup(Channel_A, GPIO.IN)
91 GPIO.setup(Channel_B, GPIO.IN)
92 GPIO.setup(Btn_Reset, GPIO.IN)
93 GPIO.setup(sftwReset, GPIO.IN)
94 GPIO.setup(fotoInt, GPIO.IN)
95 GPIO.setup(fineCorsa, GPIO.IN)
96 GPIO.setup(Led_G, GPIO.OUT)
97 GPIO.setup(errorLed, GPIO.OUT)
98 GPIO.add_event_detect(Channel_A, GPIO.BOTH, handleEnc)
99 GPIO.add_event_detect(Channel_B, GPIO.BOTH, handleEnc)
100 GPIO.add_event_detect(Btn_Reset, GPIO.FALLING, handleBtn
      )
101 GPIO.add_event_detect(fotoInt, GPIO.BOTH, handleFotoInt)
102 GPIO.add_event_detect(fineCorsa, GPIO.BOTH,
      handleFineCorsa)
103 GPIO.add_event_detect(sftwReset, GPIO.RISING, handleBtn)
104 GPIO.output(Led_G, 0)
105 GPIO.output(errorLed, 0)
106 enc_val=(GPIO.input(Channel_A)<<1)|GPIO.input(Channel_B)
107
108 file2=open("/home/pi/Desktop/ConfigFiles/stato_sensori.
      txt", "r+")
109 text=file2.readline()
110 states=text.split(",")
111 states[0]=int(states[0]) #lunghezza tot filo
112 states[1]=int(states[1]) #stato fotointerruptore
113 states[2]=int(states[2]) #stato sensore di proæ
114 flagFotoint=GPIO.input(fotoInt)
115 flagFinecorsa=GPIO.input(fineCorsa)
116 file2.seek(0)
117 file2.write('%d, %d, %d' % (states[0], flagFotoint,
      flagFinecorsa))
118 file2.truncate()
119 file2.close()
120 globalState[0]=str(flagFotoint)
121 globalState[1]=str(flagFinecorsa)
```

```
122     sleep(1)
123     GPIO.output(Led_G, 1)
124     print("".join(globalState))
125
126 def loop():
127
128     global enc_count
129     global last_save_count
130     global reset
131     global enc2m
132     global stateChangeFotoint
133     global stateChangeFinecorsa
134     global flagFotoint
135     global flagFinecorsa
136     global globalState
137
138     if reset==1:
139         enc_count=0.0
140         file1=open("/home/pi/Desktop/ConfigFiles/consumo_filo.
141         txt", "w",0)
142         file1.write('%.3f' % enc_count)
143         file1.close()
144         last_save_count=enc_count
145         print(enc_count)
146         reset=0
147
148     if stateChangeFotoint==1:
149         file2=open("/home/pi/Desktop/ConfigFiles/stato_sensori
150         .txt", "r+")
151         text=file2.readline()
152         states=text.split(",")
153         states[0]=int(states[0])
154         states[1]=int(states[1])
155         states[2]=int(states[2])
156         file2.seek(0)
157         file2.write('%d, %d, %d' % (states[0], flagFotoint,
158         states[2]))
159         file2.truncate()
160         file2.close()
161         globalState[0]=str(flagFotoint)
162         stateChangeFotoint=0
163         print("".join(globalState))
164
165     if stateChangeFinecorsa==1:
166         file2=open("/home/pi/Desktop/ConfigFiles/stato_sensori
167         .txt", "r+")
168         text=file2.readline()
169         states=text.split(",")
170         states[0]=int(states[0])
```

```

167     states[1]=int(states[1])
168     states[2]=int(states[2])
169     file2.seek(0)
170     file2.write('%d, %d, %d' % (states[0], states[1],
flagFinecorsa))
171     file2.truncate()
172     file2.close()
173     globalState[1]=str(flagFinecorsa)
174     stateChangeFinecorsa=0
175     print("".join(globalState))
176
177     if int("".join(globalState))==11:
178         GPIO.output(errorLed, 0)
179         if abs(enc_count-last_save_count)>=2*enc2m:
180             file1=open("/home/pi/Desktop/ConfigFiles/
consumo_filo.txt", "w",0)
181             file1.write('%%.3f' % enc_count)
182             file1.close()
183             last_save_count=enc_count
184         else:
185             #spegni macchina
186             GPIO.output(errorLed,1)
187             pass
188
189     if __name__=="__main__":
190         main()

```

C.2 Programma per la realizzazione dell'interfaccia grafica

Si riporta di seguito il listato in linguaggio PHP/HTML scritto per la definizione dell'interfaccia grafica, visibile nel Capitolo 4, sviluppata per mostrare in modo chiaro lo stato dei sensori.

```

1 <?php
2 $f1=file("/home/pi/Desktop/ConfigFiles/stato_sensori.txt")
3 ;
4 $f2=file("/home/pi/Desktop/ConfigFiles/consumo_filo.txt");
5 foreach ($f1 as $i => $line) {
6     $arr=explode(",", $line);
7 }
8 foreach ($arr as $value) {
9     $states_arr[] = trim($value);
10 }
11 foreach ($f2 as $i => $line) {

```

CAPITOLO C

```
12     $consumo=trim($line);
13 }
14 ?>
15
16 <!DOCTYPE html>
17 <html lang="en">
18 <head>
19     <title>Stato sensori stampante 3D</title>
20     <meta charset="utf-8">
21     <meta name="viewport" content="width=device-width,
22         initial-scale=1">
23     <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
24     <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
25     <script src="https://cdn.jsdelivr.net/npm/popper.js/1.16.0/umd/popper.min.js"></script>
26     <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"></script>
27 </head>
28 <body>
29     <div class="jumbotron text-center">
30         <h1>Stato sensori stampante 3D</h1>
31     </div>
32     <div class="container">
33         <div class="row justify-content-center">
34             <div class="col-sm-8 mb-4 mt-4 justify-content-center text-center">
35                 <div class="card" style="width: 65%; position: relative; display: inline-block; margin-left: 1em;">
36                     <div class="card-body">
37                         <ul class="list-group list-group-flush">
38                             <li class="list-group-item d-flex justify-content-between align-items-center" style="padding: 1 rem 1.25rem; font-size: large;">
39                                 FILO CONSUMATO
40                                 <span class="badge badge-light" style="font-size: large;"><?php echo $consumo;?> m</span>
41                             </li>
42                             <li class="list-group-item d-flex justify-content-between align-items-center" style="padding: 1 rem 1.25rem; font-size: large;">
43                                 FILO RIMANENTE
44                                 <span class="badge badge-light" style="font-size: large;"><?php echo number_format((float)(( $states_arr[0]- $consumo)/ $states_arr[0]*100), 2, '.', '' );?>%</span>
45                             </li>
46                         </ul>
47                     </div>
48                 </div>
49             </div>
50         </div>
51     </div>
52 </body>
53 </html>
```

CAPITOLO C

```
45     <li class="list-group-item d-flex justify-
content-between align-items-center" style="padding: 1
rem 1.25rem; font-size: large;">
46         PRESENZA FILO
47         <?php if($states_arr[1]==1): ?>
48             <span class="badge badge-success" style="
font-size: large;">OK</span>
49         <?php else: ?>
50             <span class="badge badge-danger" style="
font-size: large;">ALLARME</span>
51         <?php endif; ?>
52     </li>
53     <li class="list-group-item d-flex justify-
content-between align-items-center" style="padding: 1
rem 1.25rem; font-size: large;">
54         FILO BLOCCATO
55         <?php if($states_arr[2]==1): ?>
56             <span class="badge badge-success" style="
font-size: large;">OK</span>
57         <?php else: ?>
58             <span class="badge badge-danger" style="
font-size: large;">ALLARME</span>
59         <?php endif; ?>
60     </li>
61     <li class="list-group-item d-flex justify-
content-center" style="padding: 1.25rem 1.25rem; font-
size: large;">
62         <button type="button" class="btn btn-
outline-primary btn-lg btn-block" data-toggle="modal"
data-target="#settingsModal" style="font-size: large;">
SETTINGS </button>
63     </li>
64 </ul>
65 </div>
66 </div>
67 </div>
68 </div>
69 </div>
70
71 <div class="modal fade" id="settingsModal" tabindex="-1"
role="dialog" aria-labelledby="exampleModalCenterTitle"
aria-hidden="true">
72 <div class="modal-dialog modal-dialog-centered" role="
document">
73     <div class="modal-content">
74         <div class="modal-header">
75             <h5 class="modal-title" id="exampleModalLongTitle
">Parametri </h5>
```

CAPITOLO C

```
76     <button type="button" class="close" data-dismiss="
modal" aria-label="Close">
77         <span aria-hidden="true">&times;</span>
78     </button>
79 </div>
80 <div class="modal-body">
81     <div style="text-align: center;" >
82         <div style="display: inline-block; padding: 30
30; position: relative; margin:0;">
83             <form class="form" method="POST" style="display:
inline-block; text-align: left;">
84                 <div class="form-group">
85                     <label><strong>Lunghezza bobina:</strong
></label>
86
87                     <br>
88                     <input type="number" class="form-control"
id="length" value="<?php echo $states_arr[0]?>" style="
padding: .375rem .75rem; margin: 0 0.5em 0 0; width:
70%; display: inline-block; vertical-align: middle;"><
label style="font-size: 1.0em; width: 10%; margin:0;
display: inline-block; vertical-align: middle;">m</
label>
88
89                     </div>
90                     <div class="custom-control custom-checkbox"
style="width:100%; text-align: center; margin-top: 2em
;">
91                         <input type="checkbox" class="custom-control
-input" name="resetCountCheck" id="resetCountCheckBox">
92                         <label class="custom-control-label" for="
resetCountCheckBox"><strong>Reset encoder</strong></
label>
92
93                     </div>
94                 </form>
95             </div>
96         </div>
97     <div class="modal-footer d-flex justify-content-
center" style="width: 100%; text-align: center; margin-
top: 1.5em;">
98         <input type="button" class="form-control btn btn-
outline-primary btn-lg" name="submitbtn" value="
AGGIORNA" onclick="updateParam()" style="width: 60%;">
99     </div>
100 </div>
101 </div>
102 </div>
103 <script>
104 function updateParam(){
105     var l=document.getElementById("length").value;
```

```

106     var reset=document.getElementById("resetCountCheckBox
107     ").checked ? 1 : 0;
108
109     if(format.test(l) || l==" " || l=="0"){
110         alert("La lunghezza deve essere un intero positivo!
111         Per favore correggi.");
112     }
113     else{
114         $.ajax({
115             url: 'updateParameters.php',
116             type: 'POST',
117             dataType: "json",
118             data: {
119                 length: l,
120                 resetNow: reset
121             },
122             success: function(res){
123                 if(res==1){
124                     location.reload();
125                 }
126                 else{
127                     if(!alert("An error occurred! Please try again
128                     !")){
129                         location.reload();
130                     }
131                 }
132             },
133         });
134     }
135 }
136 </script>
137 </body>
138 </html>

```

C.2.1 Listato per l'aggiornamento dei parametri riguardanti il filamento

```

1 <?php
2 $resetPin=18;
3 $length = $_POST['length'];
4 $resetNow=$_POST['resetNow'];
5
6 $f1=file("/home/pi/Desktop/ConfigFiles/stato_sensori.txt")
7 ;
8 foreach ($f1 as $i => $line) {
9     $arr=explode(", ", $line);

```

```
10 foreach ($arr as $value) {
11     $states_arr[] = trim($value);
12 }
13
14 $text = "";
15 $text .=$length.", ".$states_arr[1].", ".$states_arr[2]."\
    r\n";
16 file_put_contents("/home/pi/Desktop/ConfigFiles/
    stato_sensori.txt", $text, LOCK_EX);
17
18 if($resetNow){
19     $text = "";
20     $text .="0.0";
21     file_put_contents("/home/pi/Desktop/ConfigFiles/
    consumo_filo.txt", $text, LOCK_EX);
22     system("gpio -g mode 24 out");
23     system("gpio -g write 24 1");
24     usleep(5000);
25     system("gpio -g write 24 0");
26 }
27 echo 1;
28
29 exit();
30
31 ?>
```

Appendice D

D.1 Programma per la valutazione dello spessore osseo

Si riporta di seguito il listato Python scritto ed utilizzato per la valutazione dello spessore osseo nel Capitolo 2.

```
1 from __main__ import vtk, qt, ctk, slicer
2 import math
3 import numpy
4
5 #
6 # Cilindri
7 #
8
9 class Cilindri:
10     def __init__(self, parent):
11         parent.title= "Cilindri"
12         parent.categories= ["Examples"]
13         parent.dependencies=[]
14         parent.contributors=[]
15         parent.helpText= """
16         """
17         parent.acknowledgementText = """
18         Pippo
19         """
20         self.parent = parent
21 class CilindriWidget:
22     def __init__(self, parent = None):
23         if not parent:
24             self.parent = slicer.qMRMLWidget()
25             self.parent.setLayout(qt.QVBoxLayout())
26             self.parent.setMRMLScene(slicer.mrmlScene)
27         else:
28             self.parent = parent
29         self.layout = self.parent.layout()
```

```
30     if not parent:
31         self.setup()
32         self.parent.show()
33     self.logic = CilindriLogic()
34     #self.tag = 0
35     self.Height=20.0
36     self.Coordinatesx=0.0
37     self.Coordinatesy=0.0
38     self.Coordinatesz=0.0
39     self.ComponentX=1.0
40     self.ComponentY=1.0
41     self.ComponentZ=1.0
42     self.NormX= 0.0
43     self.NormY= 0.0
44     self.NormZ= 0.0
45
46     def setup(self):
47
48         # Tags to manage event observers
49         self.tagSourceNode = None
50         self.tagDestinationNode = None
51         # self.ViewR = None
52
53         #
54         # Parameters Area
55         #
56         parametersCollapsibleButton = ctk.ctkCollapsibleButton()
57         parametersCollapsibleButton.text = "Parameters"
58         self.layout.addWidget(parametersCollapsibleButton)
59
60         # Layout within the dummy collapsible button
61         parametersFormLayout = qt.QFormLayout(
62             parametersCollapsibleButton)
63
64         #
65         #Source points (vtkMRMLMarkupsFiducialNode)
66         #
67         self.SourceSelector = slicer.qMRMLNodeComboBox()
68         self.SourceSelector.nodeTypeNames = ( ("
69             vtkMRMLMarkupsFiducialNode"), "" )
70         self.SourceSelector.addEnabled = True
71         self.SourceSelector.removeEnabled = False
72         self.SourceSelector.noneEnabled = True
73         self.SourceSelector.showHidden = True
74         self.SourceSelector.renameEnabled = True
75         self.SourceSelector.showChildNodeTypes = False
76         self.SourceSelector.setMRMLScene( slicer.mrmlScene )
77         self.SourceSelector.setToolTip( "Pick up a Markups node
78             listing fiducials." )
```

CAPITOLO D

```
76 parametersFormLayout.addRow("Source points: ", self.  
    SourceSelector)  
77  
78 #  
79 # Target point (vtkMRMLMarkupsFiducialNode)  
80 #  
81 # Convert the cylinder into a model  
82 #  
83 self.DestinationSelector = slicer.qMRMLNodeComboBox()  
84 self.DestinationSelector.nodeTypeNames = ( "  
    vtkMRMLModelNode"), "" )  
85 self.DestinationSelector.addEnabled = True  
86 self.DestinationSelector.removeEnabled = False  
87 self.DestinationSelector.noneEnabled = True  
88 self.DestinationSelector.showHidden = False  
89 self.DestinationSelector.renameEnabled = True  
90 self.DestinationSelector.selectNodeUponCreation = True  
91 self.DestinationSelector.showChildNodeTypes = False  
92 self.DestinationSelector.setMRMLScene( slicer.mrmlScene  
    )  
93 self.DestinationSelector.setToolTip( "Pick up or create  
    a Model node." )  
94 parametersFormLayout.addRow("Curve model: ", self.  
    DestinationSelector)  
95  
96  
97 #  
98 # Radius for the tube  
99 #  
100 self.RadiusSliderWidget = ctk.ctkSliderWidget()  
101 self.RadiusSliderWidget.singleStep = 1.0  
102 self.RadiusSliderWidget.minimum = 1.0  
103 self.RadiusSliderWidget.maximum = 50.0  
104 self.RadiusSliderWidget.value = 2.5 # nominal value  
105 self.RadiusSliderWidget.setToolTip("Set the radius of  
    the tube.")  
106 parametersFormLayout.addRow("Radius (mm): ", self.  
    RadiusSliderWidget)  
107  
108 #  
109 # height for the tube  
110 #  
111 self.HeightSliderWidget = ctk.ctkSliderWidget()  
112 self.HeightSliderWidget.connect('valueChanged(double)',  
    self.onTubeUpdatedHeight)  
113 self.HeightSliderWidget.singleStep = 1.0  
114 self.HeightSliderWidget.minimum = -1000.0  
115 self.HeightSliderWidget.maximum = 1000.0  
116 self.HeightSliderWidget.value = 20.0 # nominal value
```

CAPITOLO D

```
117 self.HeightSliderWidget.setToolTip("Set the height of
    the tube.")
118 parametersFormLayout.addRow("Height (mm): ", self.
    HeightSliderWidget)
119 self.HeightSliderWidget.setEnabled(False)
120
121
122     # Coordinates Point1
123 self.CoordinatesCollapsibleButton = ctk.
    ctkCollapsibleButton()
124 self.CoordinatesCollapsibleButton.setProperty('
    collapsedHeight', 8)
125 self.CoordinatesCollapsibleButton.text = "Coordinates"
126 self.layout.addWidget(self.CoordinatesCollapsibleButton)
127 CoordinatesFormLayout2 = qt.QFormLayout(self.
    CoordinatesCollapsibleButton)
128
129 self.CoordinatesCollapsibleButtonLayout = qt.QFormLayout
    (self.CoordinatesCollapsibleButton)
130 self.CoordinatesLayout = qt.QHBoxLayout(self.
    CoordinatesCollapsibleButton)
131 self.CoordinatesLabelPoint1 = qt.QLabel('Point 1')
132 self.CoordinatesxSpinBox = qt.QDoubleSpinBox()
133 self.CoordinatesxSpinBox.minimumWidth=2.5
134 self.CoordinatesxSpinBox.setValue(0.0)
135 self.CoordinatesxSpinBox.minimum=-1000
136 self.CoordinatesxSpinBox.maximum=1000
137 self.CoordinatesLabelx = qt.QLabel('x (mm)')
138 self.CoordinatesLabely = qt.QLabel('y (mm)')
139 self.CoordinatesySpinBox = qt.QDoubleSpinBox()
140 self.CoordinatesySpinBox.minimumWidth=5
141 self.CoordinatesySpinBox.setValue(0.0)
142 self.CoordinatesySpinBox.minimum=-1000
143 self.CoordinatesySpinBox.maximum=1000
144 self.CoordinatesLabelz = qt.QLabel('z (mm)')
145 self.CoordinateszSpinBox = qt.QDoubleSpinBox()
146 self.CoordinateszSpinBox.minimumWidth=20
147 self.CoordinateszSpinBox.setValue(0.0)
148 self.CoordinateszSpinBox.minimum=-1000
149 self.CoordinateszSpinBox.maximum=1000
150 self.CoordinatesLayout.addWidget(self.
    CoordinatesLabelPoint1)
151 self.CoordinatesLayout.addWidget(self.
    CoordinatesxSpinBox)
152 self.CoordinatesLayout.addWidget(self.CoordinatesLabelx)
153 self.CoordinatesLayout.addWidget(self.
    CoordinatesySpinBox)
154 self.CoordinatesLayout.addWidget(self.CoordinatesLabely)
```

```
155 self.CoordinatesLayout.addWidget(self.
    CoordinateszSpinBox)
156 self.CoordinatesLayout.addWidget(self.CoordinatesLabelz)
157 CoordinatesFormLayout2.addRow(self.CoordinatesLayout)
158
159 #Array
160 self.ArrayCollapsibleButton = ctk.ctkCollapsibleButton()
161 self.ArrayCollapsibleButton.setProperty('collapsedHeight
    ', 8)
162 self.ArrayCollapsibleButton.text = "Array"
163 self.layout.addWidget(self.ArrayCollapsibleButton)
164 ArrayFormLayout2 = qt.QFormLayout(self.
    ArrayCollapsibleButton)
165
166 self.ArrayCollapsibleButtonLayout = qt.QFormLayout(self.
    ArrayCollapsibleButton)
167 #self.Coordinates2CollapsibleButtonLayout.
    setContentsMargins(1,2,5,4)
168 #self.Coordinates2CollapsibleButtonLayout.setSpacing(2)
169 self.ArrayLayout = qt.QHBoxLayout(self.
    ArrayCollapsibleButton)
170 self.ArrayLabel= qt.QLabel('Elemnet_Array')
171 self.ComponentXSpinBox = qt.QDoubleSpinBox()
172 self.ComponentXSpinBox.minimumWidth=5
173 self.ComponentXSpinBox.setValue(1.0)
174 self.ComponentXSpinBox.minimum=-1000
175 self.ComponentXSpinBox.maximum=1000
176 self.ComponentLabelX = qt.QLabel('x')
177 self.ComponentLabelY = qt.QLabel('y')
178 self.ComponentYSpinBox = qt.QDoubleSpinBox()
179 self.ComponentYSpinBox.minimumWidth=5
180 self.ComponentYSpinBox.minimum=-1000
181 self.ComponentYSpinBox.maximum=1000
182 self.ComponentYSpinBox.setValue(1.0)
183 self.ComponentLabelZ = qt.QLabel('z')
184 self.ComponentZSpinBox = qt.QDoubleSpinBox()
185 self.ComponentZSpinBox.minimumWidth=20
186 self.ComponentZSpinBox.minimum=-1000
187 self.ComponentZSpinBox.maximum=1000
188 self.ComponentZSpinBox.setValue(1.0)
189 self.ArrayLayout.addWidget(self.ArrayLabel)
190 self.ArrayLayout.addWidget(self.ComponentXSpinBox)
191 self.ArrayLayout.addWidget(self.ComponentLabelX)
192 self.ArrayLayout.addWidget(self.ComponentYSpinBox)
193 self.ArrayLayout.addWidget(self.ComponentLabelY)
194 self.ArrayLayout.addWidget(self.ComponentZSpinBox)
195 self.ArrayLayout.addWidget(self.ComponentLabelZ)
196 ArrayFormLayout2.addRow(self.ArrayLayout)
197 self.ComponentXSpinBox.setEnabled(False)
```

CAPITOLO D

```
198 self.ComponentYSpinBox.setEnabled(False)
199 self.ComponentZSpinBox.setEnabled(False)
200
201
202 #
203 # Interpolation Resolution
204 #
205 self.InterpResolutionSliderWidget = ctk.ctkSliderWidget
    ()
206 self.InterpResolutionSliderWidget.singleStep = 1.0
207 self.InterpResolutionSliderWidget.minimum = 5.0
208 self.InterpResolutionSliderWidget.maximum = 50.0
209 self.InterpResolutionSliderWidget.value = 25.0 # nominal
    value
210 self.InterpResolutionSliderWidget.setToolTip("Number of
    interpolation points between control points. Default is
    25.")
211 parametersFormLayout.addRow("Resolution: ", self.
    InterpResolutionSliderWidget)
212
213 #
214 # Button to generate a curve
215 #
216 self.GenerateButton = qt.QPushButton("Generate Curve")
217 self.GenerateButton.setToolTip("Generate Curve")
218 self.GenerateButton.enabled = True
219 parametersFormLayout.addRow("", self.GenerateButton)
220
221 #
222 #Button Second Point coordinates
223 #
224 self.SecondPointButton = qt.QPushButton("Coordinates
    Second Point")
225 ArrayFormLayout2.addWidget(self.SecondPointButton)
226 # Add vertical space
227 #self.layout.addStretch(1)
228 # Set local var as instance attribute
229
230 #
231 # Connections between button and function
232 #
233 self.SourceSelector.connect("currentNodeChanged(
    vtkMRMLNode*)", self.onSourceSelected)
234 self.DestinationSelector.connect("currentNodeChanged(
    vtkMRMLNode*)", self.onDestinationSelected)
235 self.RadiusSliderWidget.connect("valueChanged(double)",
    self.onTubeUpdated)
236 self.HeightSliderWidget.connect("valueChanged(double)",
    self.onTubeUpdatedHeight)
```

CAPITOLO D

```
237 self.InterpResolutionSliderWidget.connect("valueChanged(  
    double)", self.onInterpResolutionUpdated)  
238 self.GenerateButton.connect('clicked(bool)', self.  
    onGenerateCurve)  
239 self.CoordinatesxSpinBox.connect('valueChanged(double)',  
    self.onCoordinatesxSpinBoxValueChanged)  
240 self.CoordinatesySpinBox.connect('valueChanged(double)',  
    self.onCoordinatesySpinBoxValueChanged)  
241 self.CoordinateszSpinBox.connect('valueChanged(double)',  
    self.onCoordinateszSpinBoxValueChanged)  
242 self.ComponentXSpinBox.connect('valueChanged(double)',  
    self.onComponentXSpinBoxValueChanged)  
243 self.ComponentYSpinBox.connect('valueChanged(double)',  
    self.onComponentYSpinBoxValueChanged)  
244 self.ComponentZSpinBox.connect('valueChanged(double)',  
    self.onComponentZSpinBoxValueChanged)  
245 self.SecondPointButton.connect('clicked(bool)', self.  
    onSecondPointButtonClicked)  
  
246  
247 # Add vertical space  
248 self.layout.addStretch(5)  
249 #another collapsible button  
250 textMenu = ctk.ctkCollapsibleButton()  
251 textMenu.text="Textfile"  
252 self.layout.addWidget(textMenu)  
253 textMenu.collapsed = False  
254 #new layout within textMenu  
255 textMenuLayout = qt.QHBoxLayout(textMenu)  
256 #new frame for layout  
257 textButtonFrame = qt.QFrame(textMenu)  
258 textButtonFrame.setLayout(qt.QVBoxLayout())  
259 textMenuLayout.addWidget(textButtonFrame)  
260  
261 #Load button  
262 loadTextButton = qt.QPushButton("Load textfile")  
263 loadTextButton.setToolTip="Load a textfile from your  
    filesystem"  
264 loadTextButton.connect("clicked(bool)",self.  
    loadTextButtonClicked)  
265 textButtonFrame.layout().addWidget(loadTextButton)  
266  
267 #save button  
268 saveTextButton = qt.QPushButton("save textfile")  
269 saveTextButton.setToolTip = "save text in the textfile open  
    "  
270 saveTextButton.connect("clicked(bool)", self.  
    saveTextButtonClicked)  
271 textButtonFrame.layout().addWidget(saveTextButton)  
272
```

CAPITOLO D

```
273 #text field
274 self.textfield = qt.QTextEdit()
275 self.textfield.setMaximumWidth(300)
276 textMenuLayout.layout().addWidget(self.textfield)
277
278 # f=slicer.util.getNode('F')
279 # centerPointCoord = [0.0,0.0,0.0]
280 # f.GetNthFiducialPosition(0,centerPointCoord)
281
282 def loadTextButtonClicked (self):
283     #at first we clear textfield every time we load a new
    file
284     self.textfield.clear()
285     fileDialog = ctk.ctkFileDialog()
286     #get your path to the file
287     self.filepath = fileDialog.getOpenFileName()
288
289     with open (self.filepath, "r") as file:
290         data = file.readlines()
291     for i in data:
292         self.textfield.insertPlainText(i)
293
294 def saveTextButtonClicked (self):
295     with open (self.filepath,"w") as file:
296         file.write(str(self.textfield.plainText))
297
298 def onGenerateCurve(self):
299     self.logic.generateCurveOnce()
300
301 def onSourceSelected(self):
302     # Remove observer if previous node exists
303     if self.logic.SourceNode and self.tagSourceNode:
304         self.logic.SourceNode.RemoveObserver(self.
tagSourceNode)
305
306     # Update selected node, add observer, and update
control points
307     if self.SourceSelector.currentNode():
308         self.logic.SourceNode = self.SourceSelector.
currentNode()
309
310     # Check if model has already been generated with for
this fiducial list
311     tubeModelID = self.logic.SourceNode.GetAttribute('
CurveMaker.CurveModel')
312     self.DestinationSelector.setCurrentNodeID(
tubeModelID)
313     self.tagSourceNode = self.logic.SourceNode.
AddObserver(slicer.vtkMRMLMarkupsNode.
```

CAPITOLO D

```
PointModifiedEvent, self.logic.controlPointsUpdated, 2)
314 def onDestinationSelected(self):
315     if self.logic.DestinationNode and self.
tagDestinationNode:
316         self.logic.DestinationNode.RemoveObserver(self.
tagDestinationNode)
317         if self.logic.DestinationNode.GetDisplayNode() and
self.tagDestinationDispNode:
318             self.logic.DestinationNode.GetDisplayNode().
RemoveObserver(self.tagDestinationDispNode)
319
320     # Update destination node
321     if self.DestinationSelector.currentNode():
322         self.logic.DestinationNode = self.
DestinationSelector.currentNode()
323
324
325 def onTubeUpdated(self):
326     #print "Hello!"
327     self.logic.setTubeRadius(self.RadiusSliderWidget.value)
328
329 def onTubeUpdatedHeight(self, newValue):
330     self.Height = float(newValue)
331     self.setSecondPoint()
332
333 def onCoordinatesxSpinBoxValueChanged(self, newValue):
334     self.Coordinatesx = newValue
335     #print Coordinatesx
336     self.setCoordinates()
337     self.setSecondPoint()
338
339 def onCoordinatesySpinBoxValueChanged(self, newValue):
340     self.Coordinatesy= newValue
341     self.setCoordinates()
342     self.setSecondPoint()
343
344 def onCoordinateszSpinBoxValueChanged(self, newValue):
345     self.Coordinatesz= newValue
346     self.setCoordinates()
347     self.setSecondPoint()
348
349 def setCoordinates(self):
350     self.HeightSliderWidget.setEnabled(True)
351     self.ComponentXSpinBox.setEnabled(True)
352     self.ComponentYSpinBox.setEnabled(True)
353     self.ComponentZSpinBox.setEnabled(True)
354     try:
355         f = slicer.util.getNode('F')
356         numFids = f.GetNumberOfFiducials()
```

```
357
358     except:
359         markupsNode = slicer.mrmlScene.AddNewNodeByClass("
vtkMRMLMarkupsFiducialNode", "F")
360         markupsNode.AddFiducial(self.Coordinatesx, self.
Coordinatesy, self.Coordinatesz)
361
362     else:
363         if numFids==0:
364             slicer.modules.markups.logic().AddFiducial(self.
Coordinatesx, self.Coordinatesy, self.Coordinatesz)
365         else:
366             f.SetNthFiducialPosition(0, self.Coordinatesx,
self.Coordinatesy, self.Coordinatesz)
367
368     def onComponentXSpinBoxValueChanged(self, newValue2):
369         self.ComponentX = newValue2
370         self.setSecondPoint()
371
372     def onComponentYSpinBoxValueChanged(self, newValue2):
373         self.ComponentY = newValue2
374         self.setSecondPoint()
375
376     def onComponentZSpinBoxValueChanged(self, newValue2):
377         self.ComponentZ = newValue2
378         self.setSecondPoint()
379
380     def setSecondPoint(self):
381         #self.HeightSliderWidget.setEnabled(True)
382         self.NormX= (self.ComponentX/(math.sqrt(self.
ComponentX**2+self.ComponentY**2+self.ComponentZ**2)))*
self.Height+self.Coordinatesx
383         self.NormY= (self.ComponentY/(math.sqrt(self.
ComponentX**2+self.ComponentY**2+self.ComponentZ**2)))*
self.Height+self.Coordinatesy
384         self.NormZ= (self.ComponentZ/(math.sqrt(self.
ComponentX**2+self.ComponentY**2+self.ComponentZ**2)))*
self.Height+self.Coordinatesz
385
386         try:
387             f = slicer.util.getNode('F')
388             numFids = f.GetNumberOfFiducials()
389
390         except:
391             print "Insert base node first"
392
393     else:
394         if numFids==0:
395             print "Insert base node first"
```

CAPITOLO D

```
396         elif numFids==1:
397
398             f.AddFiducial(self.NormX, self.NormY, self.NormZ)
399         else:
400             f.SetNthFiducialPosition(1, self.NormX, self.
NormY, self.NormZ)
401
402     def onSecondPointButtonClicked(self):
403         print "Second Point!"
404         # (Insert Section B text here)
405         # (be sure to match indentation of the rest of this
406         # code)
407         qt.QMessageBox.information(slicer.util.mainWindow(), '
Slicer Python', 'Coordinate secondo punto (mm):\n X %f\t
Y %f\t Z %f' %(self.NormX, self.NormY, self.NormZ))
408
409     def onInterpResolutionUpdated(self):
410         self.logic.setInterpResolution(self.
InterpResolutionSliderWidget.value)
411
412     def onSelectViewR(self, s):
413         self.setViewColor(0)
414         #self.InterpResolutionSliderWidget.enabled = True
415
416
417     def onSelectViewY(self, s):
418         self.setViewColor(1)
419         #self.InterpResolutionSliderWidget.enabled = True
420
421
422     def onSelectViewG(self, s):
423         self.setViewColor(2)
424         #self.InterpResolutionSliderWidget.enabled = False
425
426     def setViewColor(self, color):
427         if color > 3 or color < 0:
428             self.chosenView = 0
429         else:
430             self.ChosenView = color
431
432
433
434
435     #
436     # CilindriLogic
437     #
438
439     class CilindriLogic:
440
```

CAPITOLO D

```
441 def __init__(self):
442     self.SourceNode = None
443     self.DestinationNode = None
444     self.TubeRadius = 5.0
445     self.TubeHeight = 25.0
446
447     selfAutomaticUpdate = False
448     self.NumberOfIntermediatePoints = 20
449     self.ModelColor = [0.0, 1.0, 0.0]
450
451
452     self.CurvePoly = None
453     self.interpResolution = 25
454
455     # Interpolation method:
456     # 0: None
457     # 1: Cardinal Spline (VTK default)
458     # 2: Hermite Spline (Endoscopy module default)
459     self.InterpolationMethod = 0
460
461 def setNumberOfIntermediatePoints(self, npts):
462     if npts > 0:
463         self.NumberOfIntermediatePoints = npts
464         self.updateCurve()
465
466 def setTubeRadius(self, radius):
467     self.TubeRadius = radius
468     self.updateCurve()
469
470
471 def setTubeHeight(self, Height):
472     self.TubeHeight = Height
473     self.updateCurve()
474
475 def generateCurveOnce(self):
476     prevAutomaticUpdate = selfAutomaticUpdate
477     selfAutomaticUpdate = True
478     self.updateCurve()
479     selfAutomaticUpdate = prevAutomaticUpdate
480
481 def controlPointsUpdated(self, caller, event):
482     self.updateCurve()
483
484 def nodeToPoly(self, sourceNode, outputPoly, closed=
485               False):
486     points = vtk.vtkPoints()
487     cellArray = vtk.vtkCellArray()
488
489     nOfControlPoints = sourceNode.GetNumberOfFiducials()
```

```
489     pos = [0.0, 0.0, 0.0]
490     posStartEnd = [0.0, 0.0, 0.0]
491
492     offset = 0
493
494     if not closed:
495         points.SetNumberOfPoints(nOfControlPoints)
496         cellArray.InsertNextCell(nOfControlPoints)
497     else:
498         posStart = [0.0, 0.0, 0.0]
499         posEnd = [0.0, 0.0, 0.0]
500         sourceNode.GetNthFiducialPosition(0, posStart)
501         sourceNode.GetNthFiducialPosition(nOfControlPoints
-1, posEnd)
502         posStartEnd[0] = (posStart[0]+posEnd[0])/2.0
503         posStartEnd[1] = (posStart[1]+posEnd[1])/2.0
504         posStartEnd[2] = (posStart[2]+posEnd[2])/2.0
505         points.SetNumberOfPoints(nOfControlPoints+2)
506         cellArray.InsertNextCell(nOfControlPoints+2)
507
508         points.SetPoint(0, posStartEnd)
509         cellArray.InsertCellPoint(0)
510
511         offset = 1
512
513         for i in range(nOfControlPoints):
514             sourceNode.GetNthFiducialPosition(i, pos)
515             points.SetPoint(offset+i, pos)
516             cellArray.InsertCellPoint(offset+i)
517
518         offset = offset + nOfControlPoints
519
520         if closed:
521             points.SetPoint(offset, posStartEnd)
522             cellArray.InsertCellPoint(offset)
523
524         outputPoly.Initialize()
525         outputPoly.SetPoints(points)
526         outputPoly.SetLines(cellArray)
527
528     def pathToPoly(self, path, poly):
529         points = vtk.vtkPoints()
530         cellArray = vtk.vtkCellArray()
531
532         points = vtk.vtkPoints()
533         poly.SetPoints(points)
534
535         lines = vtk.vtkCellArray()
536         poly.SetLines(lines)
```

```

537
538     linesIDArray = lines.GetData()
539     linesIDArray.Reset()
540     linesIDArray.InsertNextTuple1(0)
541
542     polygons = vtk.vtkCellArray()
543     poly.SetPolys( polygons )
544     idArray = polygons.GetData()
545     idArray.Reset()
546     idArray.InsertNextTuple1(0)
547
548     for point in path:
549         pointIndex = points.InsertNextPoint(*point)
550         linesIDArray.InsertNextTuple1(pointIndex)
551         linesIDArray.SetTuple1( 0, linesIDArray.
GetNumberOfTuples() - 1 )
552         lines.SetNumberOfCells(1)
553
554     def updateCurve(self):
555         if self.AutomaticUpdate == False:
556             return
557         if self.SourceNode and self.DestinationNode:
558             print self.SourceNode.GetNumberOfFiducials()
559             if self.SourceNode.GetNumberOfFiducials() < 2:
560                 if self.CurvePoly != None:
561                     self.CurvePoly.Initialize()
562                     self.CurveLength = 0.0
563             else:
564                 if self.CurvePoly == None:
565                     self.CurvePoly = vtk.vtkPolyData()
566                 if self.DestinationNode.GetDisplayNodeID() == None
:
567                 modelDisplayNode = slicer.
vtkMRMLModelDisplayNode()
568                 modelDisplayNode.SetSliceIntersectionVisibility(
True)
569                 modelDisplayNode.SetColor(self.ModelColor)
570                 modelDisplayNode.SetOpacity(0.7)
571                 modelDisplayNode.SetSliceIntersectionThickness
(2)
572                 #model.GetDisplayNode().
SetSliceIntersectionThickness(10)
573                 slicer.mrmlScene.AddNode(modelDisplayNode)
574                 self.DestinationNode.SetAndObserveDisplayNodeID(
modelDisplayNode.GetID())
575                 if self.InterpolationMethod == 0:
576                     self.nodeToPoly(self.SourceNode, self.CurvePoly,
False)
577                 tubeFilter = vtk.vtkTubeFilter()

```

```
578     curvatureValues = vtk.vtkDoubleArray()
579     tubeFilter.SetInputData(self.CurvePoly)
580     tubeFilter.SetRadius(self.TubeRadius)
581     tubeFilter.SetNumberOfSides(20)
582     tubeFilter.CappingOn()
583     tubeFilter.Update()
584
585     self.DestinationNode.SetAndObservePolyData(
tubeFilter.GetOutput())
586     self.DestinationNode.Modified()
587     if self.DestinationNode.GetScene() == None:
588         slicer.mrmlScene.AddNode(self.DestinationNode)
589     displayNode = self.DestinationNode.GetDisplayNode()
590     if displayNode:
591         displayNode.SetActiveScalarName('')
592
593
594     def setInterpResolution(self, res):
595         ## Resoution is specified as the number of
interpolation points between two consecutive control
points
596         self.interpResolution = res
597         self.updateCurve()
```

Appendice E

E.1 Programma per l'esecuzione della registrazione basata sull'individuazione dei landmarks

Si riporta di seguito il listato python scritto ed utilizzato per l'esecuzione della registrazione delle immagini mediche digitali (Sezione 3.2) e delle ricostruzioni tridimensionali (Sezione 3.3.1), nell'ambiente di lavoro di *3D Slicer*.

```
1 from __main__ import vtk, qt, ctk, slicer
2 import math
3 import numpy as np
4
5 #
6 # Matrice TrasformazioneLineare
7 #
8
9 class MatriceTrasformazioneLineare:
10     def __init__(self, parent):
11         parent.title= "MatriceTrasformazioneLineare"
12         parent.categories= ["Matrice"]
13         parent.dependencies=[]
14         parent.contributors=[]
15         parent.helpText= ""
16         """
17         parent.acknowledgementText = ""
18         Pippo
19         """
20         self.parent = parent
21 class MatriceTrasformazioneLineareWidget:
22     def __init__(self, parent = None):
23         if not parent:
24             self.parent = slicer.qMRMLWidget()
25             self.parent.setLayout(qt.QVBoxLayout())
```

```

26     self.parent.setMRMLScene(slicer.mrmlScene)
27     else:
28         self.parent = parent
29     self.layout = self.parent.layout()
30     if not parent:
31         self.setup()
32         self.parent.show()
33     #self.logic = MatriceTrasformazioneLineareLogic()
34     #self.tag = 0
35
36
37     def setup(self):
38
39         # Tags to manage event observers
40         self.tagSourceNode = None
41         self.tagDestinationNode = None
42         # self.ViewR = None
43         self.nFidsF = 0
44         self.nFidsM = 0
45         self.X0=np.ones((4,4))
46         self.X1=np.ones((4,4))
47         self.res=np.ones((4,4))
48
49         #
50         # Parameters Area
51         #
52         FiducialCollapsibleButton = ctk.ctkCollapsibleButton()
53         FiducialCollapsibleButton.text = "Set Fiducial"
54         self.layout.addWidget(FiducialCollapsibleButton)
55
56         # Layout within the dummy collapsible button
57         FiducialFormLayout = qt.QFormLayout(
58             FiducialCollapsibleButton)
59
60         #
61         # Set Fiducial on Reference
62         #
63         self.FiducialVolFisso= slicer.qMRMLNodeComboBox()
64         self.FiducialVolFisso.nodeTypeNames = ( ("
65             vtkMRMLMarkupsFiducialNode"), "" )
66         self.FiducialVolFisso.addEnabled = True
67         self.FiducialVolFisso.removeEnabled = False
68         self.FiducialVolFisso.noneEnabled = True
69         self.FiducialVolFisso.showHidden = True
70         self.FiducialVolFisso.renameEnabled = True
71         self.FiducialVolFisso.showChildNodeTypes = False
72         self.FiducialVolFisso.setMRMLScene( slicer.mrmlScene )
73         self.FiducialVolFisso.setToolTip( "Individua 4 Markup
74             corrispondenti a quelli del volume fisso" )

```

CAPITOLO E

```
72 FiducialFormLayout.addRow("Fid 'F' Vol Fisso: ", self.  
    FiducialVolFisso)  
73  
74 #  
75 # Set Fiducial on Template  
76 #  
77 self.FiducialVolMobile= slicer.qMRMLNodeComboBox()  
78 self.FiducialVolMobile.nodeTypes = ( ("  
    vtkMRMLMarkupsFiducialNode"), "" )  
79 self.FiducialVolMobile.setEnabled = True  
80 self.FiducialVolMobile.removeEnabled = False  
81 self.FiducialVolMobile.noneEnabled = True  
82 self.FiducialVolMobile.showHidden = True  
83 self.FiducialVolMobile.renameEnabled = True  
84 self.FiducialVolMobile.showChildNodeTypes = False  
85 self.FiducialVolMobile.setMRMLScene( slicer.mrmlScene )  
86 self.FiducialVolMobile.setToolTip( "Individua 4 Markup  
    corrispondenti a quelli del volume mobile." )  
87 FiducialFormLayout.addRow("Fid 'M' Vol Mobile: ", self.  
    FiducialVolMobile)  
88  
89 #  
90 # Matrix Button  
91 #  
92 self.GenerateButton = qt.QPushButton("Generate Matrix")  
93 self.GenerateButton.setToolTip( "Genera Matrice di  
    rototraslazione" )  
94 self.GenerateButton.setEnabled = True  
95 FiducialFormLayout.addRow("Matrice", self.GenerateButton  
    )  
96  
97 # Add vertical space  
98 self.layout.addStretch(1)  
99 #another collapsible button  
100 textMenu = ctk.ctkCollapsibleButton()  
101 textMenu.text="Matrixfile"  
102 self.layout.addWidget(textMenu)  
103 textMenu.collapsed = False  
104 #new layout within textMenu  
105 textMenuLayout = qt.QHBoxLayout(textMenu)  
106 #new frame for layout  
107 textButtonFrame = qt.QFrame(textMenu)  
108 textButtonFrame.setLayout(qt.QVBoxLayout())  
109 textMenuLayout.addWidget(textButtonFrame)  
110  
111 #  
112 # Load Button  
113 #  
114 loadTextButton = qt.QPushButton("Load textfile")
```

CAPITOLO E

```
115 loadTextButton.setToolTip="Load a textfile from your
    filesystem"
116 loadTextButton.connect("clicked(bool)",self.
    loadTextButtonClicked)
117 textButtonFrame.layout().addWidget(loadTextButton)
118
119 #
120 # Save Button
121 #
122 saveTextButton = qt.QPushButton ("save matix")
123 saveTextButton.setToolTip = "save the matrix"
124 saveTextButton.connect("clicked(bool)", self.
    saveTextButtonClicked)
125 textButtonFrame.layout().addWidget(saveTextButton)
126
127 #
128 # Text Field
129 #
130 self.textfield = qt.QTextEdit()
131 self.textfield.setMaximumWidth(500)
132 textMenuLayout.layout().addWidget(self.textfield)
133
134 #
135 # Connection
136 #
137 self.FiducialVolFisso.connect("currentNodeChanged(
    vtkMRMLNode*)", self.onFiducialVolFisso)
138 self.FiducialVolMobile.connect("currentNodeChanged(
    vtkMRMLNode*)", self.onFiducialVolMobile)
139 self.GenerateButton.connect('clicked(bool)', self.
    onApplyButton)
140
141 def loadTextButtonClicked (self):
142     #at first we clear textfield every time we load a new
    file
143     self.textfield.clear()
144     fileDialog = ctk.ctkFileDialog()
145     #get your path to the file
146     self.filepath = fileDialog.getOpenFileName()
147
148     with open (self.filepath, "r") as file:
149         data = file.readlines()
150         for i in data:
151             self.textfield.insertPlainText(i)
152 def saveTextButtonClicked (self):
153     with open (self.filepath,"w") as file:
154         file.write(str(self.res))
155     self.textfield.clear()
156     with open (self.filepath, "r") as file:
```

```

157     data = file.readlines()
158     for i in data:
159         self.textfield.insertPlainText(i)
160
161     #
162     # Creation class F
163     #
164     def onFiducialVolFisso (self):
165         f=slicer.util.getNode('F')
166         self.nFidsF = f.GetNumberOfFiducials()
167         print self.nFidsF
168         self.X0=np.ones((4,4))
169         for n in range(0,4):
170             f.GetNthFiducialPosition(n,self.X0[n,0:3])
171
172     #
173     # Creaztion classe M
174     #
175     def onFiducialVolMobile(self):
176         m=slicer.util.getNode('M')
177         self.nFidsM = m.GetNumberOfFiducials()
178         print self.nFidsM
179         self.X1=np.ones((4,4))
180         for t in range(0,4):
181             m.GetNthFiducialPosition(t,self.X1[t,0:3])
182
183     def onApplyButton(self):
184         self.onFiducialVolFisso ()
185         self.onFiducialVolMobile()
186         print self.nFidsM
187         print self.nFidsF
188         print self.X0
189         print self.X1
190         nFids = self.nFidsF + self.nFidsM
191         print nFids
192         if nFids>=8 :
193             X0_trasp=np.transpose(self.X0)
194             print "X0 trasp"
195             print X0_trasp
196             X1_trasp=np.transpose(self.X1)
197             print "X1 trasp"
198             print X1_trasp
199             X1i=np.linalg.inv(X1_trasp)
200             print "Inv"
201             print X1i
202             self.res =np.matmul(X0_trasp, X1i)
203             print "res"
204             print self.res
205

```

CAPITOLO E

```
206     else :  
207         print "Numero Fiducial insufficienti"
```

Appendice F

F.1 Programma per la stima della variazione dell'inclinazione del capo nei referti confrontati

Si riporta di seguito il listato python scritto ed utilizzato per la stima della variazione dell'inclinazione del capo nei referti confrontati (Sezione 3.2).

```
1 print("#####")
2 import numpy as np
3 import math
4
5 ### PRIMO PIANO ###
6
7 # 3 punti appartenenti al primo piano
8
9 # PazB_Referto_Pre_Operatorio
10 # A=np.array([-14.650, 55.980, 25.315])
11 # B=np.array([-10.734, 58.159, -45.840])
12 # C=np.array([-5.359, -14.038, -67.314])
13
14 # PazA_Referto_Pre_Operatorio
15 A=np.array([-15.063, 47.902, -30.230])
16 B=np.array([2.714, -28.592, -47.333])
17 C=np.array([4.522, -21.151, 62.520])
18
19 ## Calcolo componenti vettore v1 ##
20
21 a=(B[2]-A[2])*(C[1]-A[1])-(B[1]-A[1])*(C[2]-A[2])
22 b=- (B[2]-A[2])*(C[0]-A[0])+(B[0]-A[0])*(C[2]-A[2])
23 c=(B[1]-A[1])*(C[0]-A[0])-(B[0]-A[0])*(C[1]-A[1])
24
25 ## v1 vettore normale ##
26 v1_old=[a,b,c]
27 print ("v1 old",v1_old)
```

```

28 # if a<0:
29     # v1=[-a, -b, -c]
30 # else:
31     # v1=[a, b, c]
32 v1=np.array([a, b, c])
33 print ("v1",v1)
34
35 norm_v1=math.sqrt(v1[0]**2+v1[1]**2+v1[2]**2) #calcolo
        della norma di v1
36 print ("norma v1",norm_v1)
37
38 vers_v1=v1/norm_v1
39 print ("versore v1",vers_v1) #calcolo versore v1
40 print()
41
42 ### SECONDO PIANO ###
43
44 # 3 Punti apparteneti al secondo piano
45
46 #PazB_Referto_Post_operatorio
47 # A2=np.array([-7.360, 63.484, 31.906])
48 # B2=np.array([-2.748, 67.815, -39.322])
49 # C2=np.array([7.997, -5.362, -60.883])
50
51
52 #PazA_Referto_Post_operatorio
53 A2=np.array([-18.243, 34.321, -42.679])
54 B2=np.array([8.866, -42.219, -57.265])
55 C2=np.array([-1.391, -37.578, 55.159])
56
57 ## Calcolo componenti vettore v2 ##
58
59 a2=(B2[2]-A2[2])*(C2[1]-A2[1])-(B2[1]-A2[1])*(C2[2]-A2[2])
60 b2=-(B2[2]-A2[2])*(C2[0]-A2[0])+(B2[0]-A2[0])*(C2[2]-A2
        [2])
61 c2=(B2[1]-A2[1])*(C2[0]-A2[0])-(B2[0]-A2[0])*(C2[1]-A2[1])
62
63 ## v2 vettore normale ##
64 v2_old=[a2,b2,c2]
65
66 print ("v2 old",v2_old)
67 # if a2<0:
68     # v2=[-a2, -b2, -c2]
69 # else:
70     # v2=[a2, b2, c2]
71 v2=np.array([a2, b2, c2])
72 print ("v2",v2)
73

```

CAPITOLO F

```
74 norm_v2=math.sqrt(v2[0]**2+v2[1]**2+v2[2]**2) #calcolo
    della norma di v2
75 print ("norma v2",norm_v2)
76
77 vers_v2=v2/norm_v2
78 print ("versore v2",vers_v2) #calcolo versore v2
79
80 ### CALCOLO ANGOLO TRA PIANI ###
81 Theta_rad= np.arccos(np.dot(v1,v2)/(norm_v1*norm_v2)) #rad
82 Theta=Theta_rad*180/np.pi
83 print ("Theta",Theta)
84 print()
85
86 ### CALCOLO ASSE DI ROTAZIONE ###
87 a3=(v1[0]*v2[1]-v1[1]*v2[0])
88 b3=-((v1[0]*v2[2]-v1[2]*v2[0]))
89 c3=v1[1]*v2[2]-v1[2]*v2[1]
90
91 v3=np.array([a3,b3,c3])
92 norm_v3=math.sqrt(v3[0]**2+v3[1]**2+v3[2]**2)
93 v3_norm=v3/norm_v3
94 print("v3",v3)
95 print("Norma v3",norm_v3)
96 print("versore v3",v3_norm)
97
98 print ()
99 print("#####")
100
101 ### ANGOLI DI ROTAZIONE ###
102 # Per calcolare gli angoli di rot intorno ai tre assi del
    sistema di
103 # riferimento proietto i due vettori normali sui piani
    del sistema di
104 # RIF (x,y,z)
105
106 ## proiezione sul piano XY ##
107 v1_XY= np.array([v1[1],v1[2]])
108 norm_v1_XY=math.sqrt(v1_XY[0]**2+v1_XY[1]**2)
109 v2_XY= np.array([v2[1],v2[2]])
110 norm_v2_XY=math.sqrt(v2_XY[0]**2+v2_XY[1]**2)
111 print ("norma v1 XY",norm_v1_XY)
112 print ("norma v2 XY",norm_v2_XY)
113 print()
114
115 Yaw_rad=np.arccos(np.dot(v1_XY,v2_XY)/(norm_v1_XY*
    norm_v2_XY))
116 YAW=Yaw_rad*180/np.pi #Rot attorno all'asse Z
117 print ("YAW",YAW)
118
```

```

119 ## proiezione sul piano ZX ##
120 v1_ZX= np.array([v1[0],v1[2]])
121 norm_v1_ZX=math.sqrt(v1_ZX[0]**2+v1_ZX[1]**2)
122 v2_ZX= np.array([v2[0],v2[2]])
123 norm_v2_ZX=math.sqrt(v2_ZX[0]**2+v2_ZX[1]**2)
124
125 Roll_rad=np.arccos(np.dot(v1_ZX,v2_ZX)/(norm_v1_ZX*
      norm_v2_ZX))
126 ROLL=Roll_rad*180/np.pi #Rot attorno all'asse Y
127 print ("ROLL",ROLL)
128
129 ## proiezione sul piano ZY ##
130 v1_ZY= np.array([v1[0],v1[1]])
131 norm_v1_ZY=math.sqrt(v1_ZY[0]**2+v1_ZY[1]**2)
132 v2_ZY= np.array([v2[0],v2[1]])
133 norm_v2_ZY=math.sqrt(v2_ZY[0]**2+v2_ZY[1]**2)
134
135 Pitch_rad=np.arccos(np.dot(v1_ZY,v2_ZY)/(norm_v1_ZY*
      norm_v2_ZY))
136 PITCH=Pitch_rad*180/np.pi #Rot attorno all'asse X
137 print ("PITCH",PITCH)

```

Ringraziamenti

Desidero ringraziare innanzitutto il Professore Sandro Moos, relatore della tesi, per la disponibilità e l'attenzione dimostrate e Andrea Novaresio per il supporto fornito.

Un ringraziamento particolare va ai miei fratelli e ai miei genitori che, in tutti questi anni di studio, mi hanno sopportata e supportata, aiutandomi a raggiungere il traguardo della laurea magistrale.

Bibliografia

- [1] S. Murphy, A. Atala, "3D bioprinting of tissues and organs". *Nat Biotechnol* 32, pp.773–785 (2014).
- [2] R.B. Bell, "Computer planning and intraoperative navigation in cranio-maxillofacial surgery", *Oral Maxillofac Surg Clin North Am*, pp. 135-136,(2010).
- [3] A. Tarsitano, S. Battaglia, V. Ramieri, P. Cascone, L. Ciocca, R. Scotti e C. Marchetti, "Short-term outcomes of mandibular reconstruction in oncological patients using a CAD/CAM prosthesis including supporting a fibula free flap", *Journal of Cranio-Maxillo-Facial-Surgery*, n. 45, pp. 330-337,(2017).
- [4] Y. Y. Wang, H. Q. Zhand, S. Fan, D. M. Zhang, Z. Q. Huang, W. L. Chen, J. T. Ye e L. J. S., "Mandibular reconstruction with the vascularized fibula flap: comparison of virtual planning surgery and conventional surgery", *Oral & Maxillofacial Surgery*, n. 45, pp. 1400-1405, (2016).
- [5] J. Li, W. Chen, Z. Huang e D. Zhang, "Pediatric mandibular reconstruction after benign tumor ablation using a vascularized fibular flap", *J Craniofac Surg*, n. 20, pp. 431-434,(2009).
- [6] Y. Shen, J. Sun, J. Li, T. Ji, M. Li, W. Huang e M. Hu, "Using computer simulation and stereomodel for accurate mandibula reconstruction with vasculatrized iliac crest flap", *Oral & Maxillofacial surgery*, vol. 114, n. 2, pp. 175-182, (2012).
- [7] L. Ganry, J. Quilichini, C. Bandini, P. Leyder, B. Hersant e J. Menin-guad, "Three-dimensional surgical modelling with an open-surce software protocol: study of precision and reproducibility in mandibular reconstruction with the fibula free flap", *Oral & Maxillofacial Surgery*, n. 46, pp.946-957,(2017).

BIBLIOGRAFIA

- [8] E. Zavattero, P. Garzino Demo, M. Fasolis e G. Ramieri, «To computer-aided design and manufacturing or not to computer-aided design and manufacturing? Free fibula flap with comuter - aided technique for mandibula reconstruction,» *Journal of Craniofacial Surgery*, n. 26,pp. e206-e209,(2015).
- [9] S. Roser, S. Ramachandra, H. Blair, W. Grisl, G. Carlson, C. A., K. Weimer e M. Steed, «The Accuracy of Virtual Surgical Planning in Free Fibula Mandibular Reconstruction: Comparison of Planned and Final Results,» *Journal of oral maxillofacial surgery*, n. 68, pp. 2824-28232,(2010).
- [10] A. Sotiras, C. Davatzikos, N. Paragios,"Deformable medical image registration: a survey", *IEEE Trans. Med. Imaging*,pp. 1153-1166,(2013).
- [11] F. E.-Z. A. El-Gamal, M. Elmogy, & A. Atwan,"Current trends in medical image registration and fusion". *Egypt. Informat. J.*,pp.99-121,(2016).
- [12] M. Esposito, L. M. Riccardi. "Il problema della co-registrazione di immagini biomediche e sue applicazioni nell'ambito radiologico e diagnostico".
- [13] J. Modersitzki,"Numerical methods for image registration" ,(2004).
- [14] Sandro Barbone,"Il corpo umano. Igiene, Anatomia, Fisiologia e Patologia". Franco Lucisano Editore,(2010).
- [15] https://www.researchgate.net/figure/The-yaw-pitch-and-roll-angles-in-the-human-head-motion-11_fig1_305684696
- [16] <https://ultimaker.com/it/3d-printers/ultimaker-3>
- [17] V. Viktorov, F. Colombo ,"Automazione dei sistemi meccanici". quarta edizione. C.L.U.T Editrice,(2016).
- [18] <https://www.materialise.com/en/medical/software/proplan-cmf/regulatory-information>
- [19] E.Atzeni,"Il controllo numerico delle macchine utensili- I trasduttori"
- [20] Drake R.L., Vogl W., Mitchell A.W.M., "Gray's anatomy for students.", Philadelphia: Elsevier Inc; third edition, p. 855-857,(2015).