

UNIVERSITÀ DEGLI STUDI DI PADOVA



FACOLTÀ DI INGEGNERIA  
Corso di Laurea in Bioingegneria

**RICOSTRUZIONE DI MODELLO 3D DA  
SCANSIONI MRI DELL'ENCEFALO PER SISTEMA  
DI NEURONAVIGAZIONE COMPUTERIZZATA  
PER TMS**

**Relatore: Ch.ma Prof.ssa Toffolo Gianna Maria**

**Correlatori: Ing. Rossi Marco  
Ing. Brunelli Cristian**

**Laureando: Elia Oliviero**

**Anno Accademico 2010/2011**



Innanzitutto ringrazio la Prof.ssa Gianna Maria Toffolo per l'attenzione, la professionalità ed i preziosi consigli, non solo relativi alla tesi, ma molto utili anche per una futura esperienza professionale.

Ringrazio l'Ing. Marco Rossi e l'Ing. Cristian Brunelli per l'aiuto e l'assistenza nel corso della stesura di questo elaborato e per l'ottima esperienza vissuta presso la loro azienda.

Ringrazio i miei genitori, per il loro supporto, il loro sostegno, la loro comprensione, la loro capacità di capirmi e soprattutto per l'affetto e per la possibilità che mi hanno dato di arrivare fin qui, di frequentare l'università e di studiare ciò che desidero e li ringrazio per essermi stati accanto in tutte le scelte che ho fatto fino ad ora.



# Indice

<b>1 INTRODUZIONE</b>	<b>1</b>
<b>2 SEGMENTAZIONE CON SOGLIA E CON CONTORNI ATTIVI</b>	<b>7</b>
2.1 La segmentazione .....	7
2.2 Thresholding .....	8
2.2.1 Formulazione matematica .....	9
2.2.2 Standard DICOM .....	10
2.2.3 Implementazione .....	11
2.2.4 Risultati .....	16
2.3 Contorni attivi .....	17
2.3.1 Formulazione matematica .....	19
2.3.2 Implementazione .....	22
2.3.3 Risultati .....	24
2.3.4 Limitazioni e problematiche dei contorni attivi .....	26
<b>3 LA SEGMENTAZIONE UNIFICATA</b>	<b>29</b>
3.1 Metodo .....	29
3.1.1 Mix di gaussiane .....	30
3.1.2 Bias .....	32
3.1.3 Mappe probabilistiche .....	34
3.1.4 Regolarizzazione .....	36
3.1.5 Ottimizzazione e stima dei parametri .....	37

3.1.5.1 Stima dei parametri $\mu, \sigma^2, \gamma$ .....	38
3.1.5.2 Stima dei parametri $\beta$ .....	39
3.1.5.3 Stima dei parametri $\alpha$ .....	40
3.2 Implementazione .....	41
3.3 Plot dei risultati .....	50
3.4 Analisi qualitativa dei risultati .....	52
3.5 Prestazioni computazionali .....	54

## **4 VALIDAZIONE DEL METODO DI SEGMENTAZIONE UNIFICATA 59**

4.1 Testing set .....	59
4.2 Indici di validazione .....	64
4.3 Procedura di validazione ed implementazione .....	66
4.4 Misure ed indici calcolati: metodo 'sd' .....	72
4.5 Conclusioni .....	75

## **5 CONCLUSIONI 77**

## **Appendice 83**

A1 .....	83
A2 .....	89

## **Bibliografia 96**







# Capitolo 1

## Introduzione

La stimolazione magnetica transcranica (TMS) è stata introdotta per la prima volta nel 1985 da un gruppo di ricerca di Sheffield, che, messo a punto lo strumento, ha avviato le prime ricerche stimolando aree cerebrali senza provocare dolore. È diventata oggi un utile strumento diagnostico ed investigativo nella pratica clinica neurofisiologica e recentemente alcuni studi hanno suggerito la sua efficacia nel trattamento di alcuni disturbi psichici come la depressione. La TMS è uno strumento di esplorazione nell'uomo della fisiologia del cervello sano e dei correlati tra l'attività e il comportamento; è un potente strumento di indagine delle neuroscienze.

Dal punto di vista clinico e di ricerca si spazia dallo studio delle aree corticali primarie, alla plasticità neurale, alle neuroscienze cognitive. La mappatura funzionale, tramite la stimolazione magnetica transcranica, permette di correlare le differenti aree cerebrali motorie, sensoriali e cognitive, con le relative proprietà funzionali e, conseguentemente, di prevedere quali danni facciano seguito a determinate lesioni e di osservare le modificazioni neurali che si hanno in seguito a determinate patologie (plasticità neurale). Vengono così stimulate le differenti aree che si ipotizza siano deputate alle diverse funzioni e si misurano le attività sensoriali e motorie che vengono generate dalla stimolazione. La stimolazione delle aree motorie primarie genera una contrazione involontaria dei muscoli contralaterali, la sollecitazione della corteccia somatosensoriale primaria interferisce negativamente con la percezione tattile, mentre scotomi e fosfeni possono essere percepiti con la stimolazione dell'area visiva primaria. La TMS viene utilizzata anche in combinazione con l'elettroencefalogramma (EEG). La

stimolazione magnetica è in grado di sollecitare i neuroni superficiali della materia grigia mentre l'EEG è in grado di registrare l'attività elettrica cerebrale. Nell'ambito delle neuroscienze cognitive, la tecnica è stata utilizzata allo scopo di studiare i processi neurali legati alle attività delle aree corticali superiori, all'interno di studi sulla percezione, attenzione, apprendimento, linguaggio e consapevolezza. La stimolazione magnetica transcranica ripetitiva ad alta frequenza ( $> 1$  Hz) può generare un'interruzione transitoria e focale della normale attività cerebrale, dando origine ad una sorta di lesione temporanea e reversibile, riproducendo così le condizioni che determinano il deficit. In clinica psichiatrica il principale campo di utilizzo della TMS è la terapia della depressione, ma si sta sperimentando la sua utilità anche nel disturbo ossessivo compulsivo (DOC), nei disturbi allucinatori e nella dipendenza da cocaina.

La Stimolazione Magnetica Transcranica viene effettuata tramite un dispositivo elettronico detto Stimolatore Magnetico. Concettualmente, tale apparecchiatura è composta da un generatore di tensione di carica, un condensatore, alcuni interruttori che permettono la carica o la scarica del condensatore, e da una bobina (coil) stimolante posta a diretto contatto con il cranio del paziente. La TMS funziona secondo il principio dell'Induzione elettromagnetica. Se vi è un passaggio di corrente entro una bobina metallica (coil), si genera un campo elettromagnetico perpendicolare al flusso di corrente presente nella bobina. Posizionando un secondo conduttore (scalpo) entro il campo magnetico, viene indotta corrente in questo secondo mezzo. Si determina, pertanto, una corrente indotta, in grado di produrre dei potenziali di azione in neuroni eccitabili della corteccia. È importante notare come gli effetti della tecnica non dipendano direttamente dal campo magnetico, bensì dal campo elettrico indotto, il quale provoca la depolarizzazione neuronale. Il campo elettrico generato dipende da diversi fattori. Per quanto riguarda il coil, questo può avere forme e dimensioni differenti, anche se attualmente viene maggiormente impiegato il coil con la forma a "8", che genera un campo elettrico più focalizzato, il quale permette di stimolare aree cerebrali più precise. Lo stimolatore inoltre implementa un'elettronica di controllo che permette di controllare l'intensità ed il numero di impulsi inviati. I metodi di stimolazione sono sostanzialmente due: il metodo a "singolo impulso" (Single Pulse TMS) dove l'impulso consiste in una singola scarica conferita al soggetto, ed i "treni di impulsi" (Repetitive TMS) dove

l'impulso consiste in n scariche in un periodo di tempo prestabilito. È da notare che, contrariamente all'elettroshock, dov'è prodotta una depolarizzazione neuronale massiva, in cui la teca cranica funge da resistenza, i campi magnetici non sono deflessi o attenuati dai tessuti e ciò rende questa tecnica più focale e quindi mirata, oltre che indolore, non essendo stimolati i recettori nocicettivi.

Nell'uso della TMS per lo studio del cervello umano, la conoscenza della corrispondenza spaziale esistente tra la posizione del coil stimolante posto sullo scalpo del soggetto in esame e la sottostante corteccia cerebrale, riveste un'importanza fondamentale. L'esatto posizionamento del coil è fondamentale per una corretta interpretazione degli effetti dello stimolo.

L'obiettivo di questa tesi è quello di sviluppare un algoritmo di ricostruzione di modelli 3D dello scalpo e del cervello a partire da scansioni MRI dell'encefalo: tale obiettivo si inserisce nell'ambito di un progetto per un sistema di neuronavigazione robotizzata specifico per l'utilizzo con la stimolazione magnetica transcranica, al fine di facilitare il corretto posizionamento del coil stimolante in corrispondenza di specifiche aree del cervello del soggetto in esame. L'algoritmo, sviluppato in Matlab R2010b, deve essere ottimizzato in modo da poter essere eseguito in tempi ragionevoli su PC di fascia consumer, in modo tale da garantire l'applicabilità dell'intero sistema in ambulatorio medico. Il progetto è nato ed è stato sviluppato presso l'azienda ATES Medica Device.

Il lavoro si articola per capitoli nel seguente modo:

- **Capitolo 2**

Descrizione delle tecniche di segmentazione basate sulla soglia e sui contorni attivi: sono illustrate brevemente alcune tecniche di segmentazione che sono state prese in esame all'inizio del progetto al fine di individuare la tecnica più adatta per l'applicazione specifica, evidenziandone pregi e difetti.

- **Capitolo 3**

Metodo della segmentazione unificata e implementazione dell'algoritmo in tutti i suoi step: in questo capitolo è descritta la soluzione che è stata

---

adottata evidenziandone i vantaggi rispetto alle altre tecniche, nonché le problematiche affrontate.

- **Capitolo 4**

Validazione del metodo della segmentazione unificata: in questo capitolo viene valutata la bontà dei risultati ottenuti; sono definiti degli indici di validazione che permettono di affermare o meno se la segmentazione e quindi i modelli ricostruiti sono corretti.

- **Capitolo 5**

Conclusioni.

Segue un'appendice in cui si visualizzano i modelli creati con un semplice software che permette una visualizzazione 3D migliore di quella del Matlab.



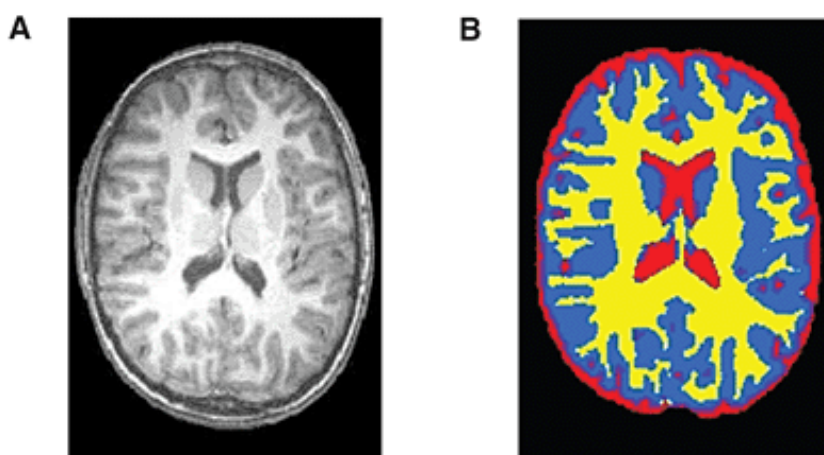


## Capitolo 2

# Segmentazione con soglia e con contorni attivi

### 2.1 La segmentazione

L'obiettivo della segmentazione è partizionare automaticamente un'immagine in regioni in modo tale da creare zone in cui i pixel vengono ad avere una particolare proprietà in comune: è quindi un'operazione di classificazione automatica dei pixel; consiste in un'estrazione di strutture di interesse dallo sfondo o da altre strutture. La segmentazione è uno strumento molto importante nell'imaging medico e utile in molteplici applicazioni quali la pianificazione di interventi, la classificazione in regioni anatomiche, la quantificazione di lesioni.



- **Figura 2.1.** Rappresentazione di un'immagine iniziale (A) e rappresentazione dell'immagine segmentata (B).

Le metodologie allo stato dell'arte per la segmentazione di immagini sono svariate e molto differenti tra loro: si va dalla semplice applicazione di filtri, al clustering non supervisionato fino a metodi variazionali basati sulla minimizzazione di funzionali (energia). Infatti, a seconda della particolare applicazione cambiano sia la definizione di segmento, ovvero la definizione di "omogeneità" delle regioni da segmentare, sia i requisiti dell'algoritmo in termini di accuratezza e, soprattutto, di efficienza computazionale; non si può dunque affermare che esista un'unica tecnica applicabile universalmente e che produca risultati soddisfacenti in tutti i campi di applicazione: a seconda dell'obiettivo è possibile individuare un algoritmo più efficiente di altri per realizzarlo.

L'obiettivo è quindi quello di effettuare la suddivisione di immagini MRI bidimensionali dell'encefalo in sottoregioni, materia bianca, materia grigia, tessuto osseo per poi ricostruire modelli 3d di cervello e scalpo. Una rappresentazione compatta di queste strutture è difficile per la variabilità e complessità delle forme anatomiche e a seconda delle tecniche impiegate si può arrivare alla definizione di contorni e regioni indistinguibili o disconnesse.

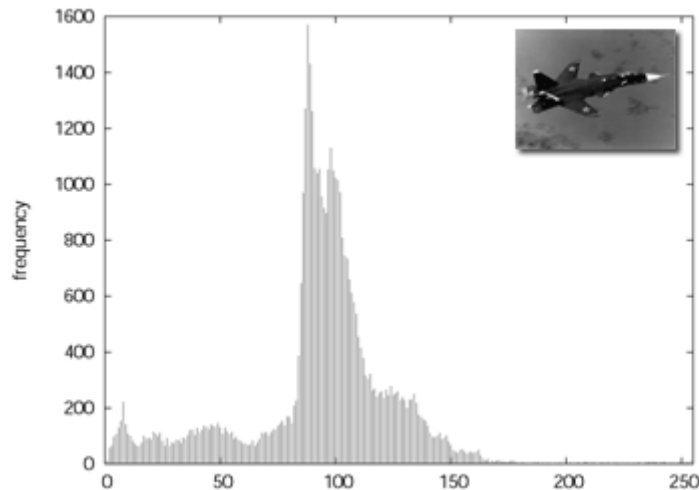
Di seguito sono illustrate brevemente alcune tecniche di segmentazione che sono state prese in esame all'inizio del progetto al fine di individuare la tecnica più adatta per l'applicazione specifica.

## 2.2 Thresholding

L'algoritmo *thresholding* (sogliatura) ha lo scopo di dividere l'immagine in due zone: la figura e lo sfondo (background). Tale tecnica procede secondo l'algoritmo di seguito descritto. Dato un valore di soglia  $k$  associata ad un parametro scelto, divide i pixel dell'immagine in due insiemi, quelli con valore inferiore a  $k$  e quelli con valore superiore. Molto spesso per garantire la robustezza dell'algoritmo, si lascia all'algoritmo stesso la scelta del valore di soglia  $k$ , sfruttando le informazioni dell'istogramma dell'immagine, una rappresentazione grafica della quantità di pixel presenti nell'immagine per ciascun livello di grigio  $r$ . Ricavato l'istogramma dell'immagine, se ha un unico picco si impone come soglia  $k$  il valore medio che divide l'istogramma in due parti. Quando non si riesce a segmentare l'immagine con una singola soglia globale, ovvero quando



l'istogramma presenta più picchi, la si partiziona in sottoimmagini (di dimensioni tali da poter ipotizzare un'illuminazione uniforme) i cui istogrammi siano composti da un solo picco, per cui è possibile individuare una sola soglia per ogni sottoimmagine.



➤ **Figura 2.2.** Esempio di istogramma di un'immagine.

### 2.2.1 Formulazione matematica

Possiamo definire l'operatore di thresholding nel seguente modo:

$$T = T [x, y, p(x,y), f(x,y)]$$

Ove  $(x,y)$  sono le coordinate di un punto dell'immagine;  $f(x,y)$  è l'immagine;  $p(x,y)$  denota una proprietà locale del punto.

Se  $T$  dipende solo da  $f(x,y)$  → **soglia globale**

Se  $T$  dipende da  $f(x,y)$  e  $p(x,y)$  → **soglia locale**

Se  $T$  dipende dalle coordinate spaziali  $x$  e  $y$  → **soglia adattativa**

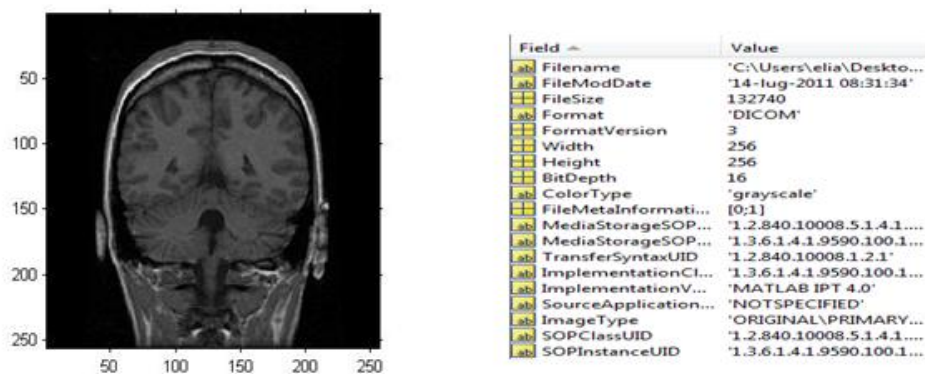
E l'immagine  $g(x,y)$  elaborata è definita come segue:

$$g(x,y) = 1 \quad \text{se } f(x,y) > T$$

$$g(x,y) = 0 \quad \text{se } f(x,y) \leq T$$

### 2.2.2 Standard DICOM

Lo studio alla ricerca della soluzione più adatta per ottenere immagini segmentate di buona qualità per ricostruire successivamente i modelli, inizia con l'applicazione della tecnica thresholding. Il linguaggio di programmazione utilizzato è MATLAB ver 2010b. È stato utilizzato un database costituito da 60 file che corrispondono a 60 slice coronali ottenute da risonanza magnetica. Il formato **DICOM** (**D**igital **I**maging and **C**OMmunications in **M**edicine) è uno standard (cioè un insieme di regole) per la codifica e la trasmissione tra apparecchiature diverse (macchine per acquisizione, computer, stampanti, archivi digitali, etc.) di dati digitali come immagini biomediche e di informazioni medico-sanitarie, mediante la definizione di un formato di dati e di un protocollo di comunicazione. Un file DICOM nella sua essenza può essere visto come un contenitore in cui i dati contenuti non rappresentano solamente i pixel e i colori dell'immagine, ma anche un'insieme strutturato di dati che descrivono tutto il procedimento che ha portato alla costruzione dell'immagine stessa. Sostanzialmente un file DICOM consiste di un'intestazione (*header*) e da un corpo dati atto a contenere una o più immagini. In particolare, l'intestazione è costituita da un insieme di attributi contenenti informazioni di varia natura: tali attributi possono venire raggruppati in base alla relazione che esiste tra di loro, formando quindi l'entità conosciuta come oggetto informativo (*Information Object Modules*); con ciascun gruppo può essere identificato come l'astrazione di un'entità reale, quali ad esempio: Paziente, Studio, Serie di dati, Immagine e così via.



➤ **Figura 2.2.** Esempio di file dicom: corpo dati (sinistra) ed intestazione con alcuni attributi (destra).

### 2.2.3 Implementazione

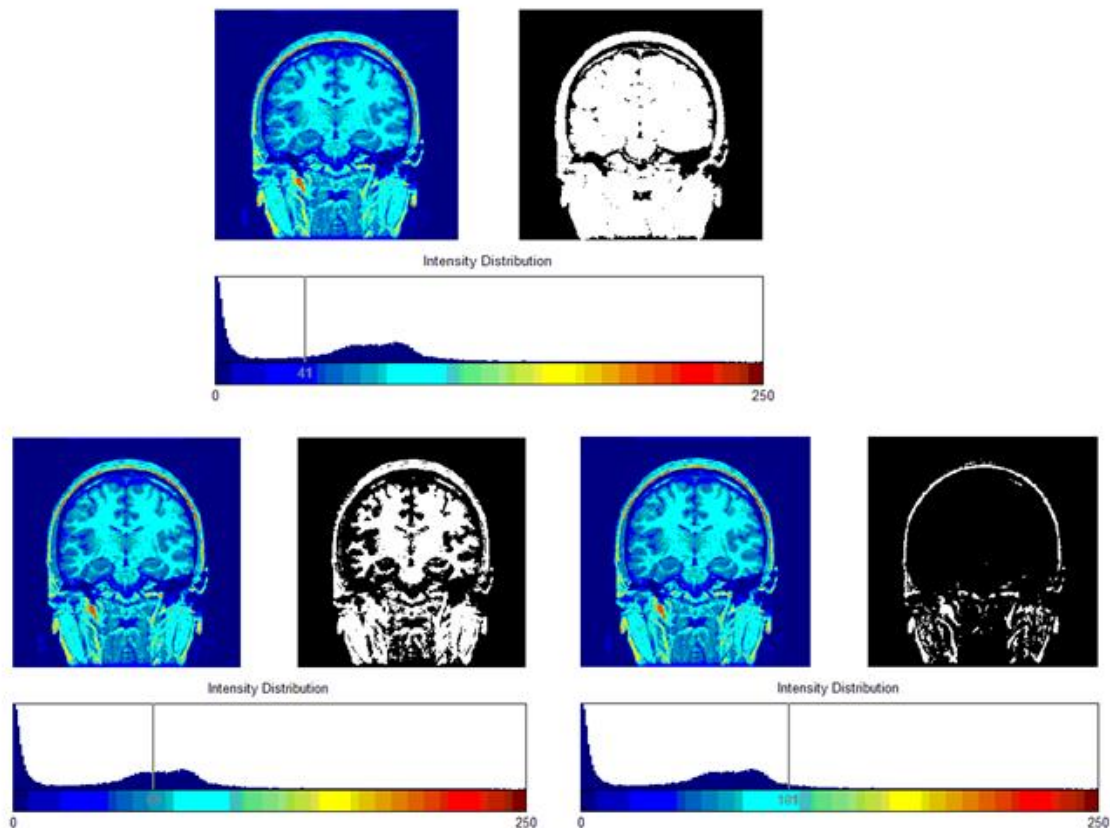
Inizialmente sono stati importati in matlab tutti i 60 file dicom e, tramite il comando **dicomread**, sono stati estratti i dati immagine da ciascun file (si crea una matrice di dimensioni 256x256) ed è stato ricreato un volume 3D di dati (di dimensioni 256x256x60) semplicemente affiancando l'una all'altra le matrici bidimensionali relative ad ogni slice.

```
for i=length(fnum):-1:1
    fname = [prefix num2str(fnum(i)) ext];
    D(:,:,i) = uint16(dicomread(fname));
end
```

#### ❖ Frammento di codice 2.1

È richiamata poi la funzione **thresh\_seg** che sfruttando la funzione di matlab **graythresh** seleziona in modo automatico la soglia globale ottima dando in output l'immagine binaria segmentata; la soglia ottima è calcolata sfruttando il metodo di Otsu: questo metodo si basa sull'analisi del discriminante. I pixel dell'immagine vengono associati a due classi:  $C_0$  e  $C_1$ , con  $C_0=(0,1,\dots,t)$  e  $C_1=(t+1,t+2,\dots,l-1)$ , dove  $t$  rappresenta il livello di sogliatura, ed  $l$  il numero di livelli di grigio dell'immagine. Se  $\sigma_w^2$  è la varianza della prima classe,  $\sigma_t^2$  è la varianza della seconda classe, e  $\sigma_T^2$  è la varianza totale, la sogliatura ottima si ottiene massimizzando una delle seguenti funzioni:  $\sigma_b^2/\sigma_w^2, \sigma_T^2/\sigma_w^2, \sigma_b^2/\sigma_T^2$ .

L'utente può anche scegliere in maniera interattiva un diverso valore di soglia: come è possibile vedere dalle figure che seguono, questa operazione fa sì che non vengano considerate sulle immagini parte delle strutture.



➤ **Figura 2.3.** Immagine segmentata con soglia ottima (in alto); immagini segmentate con diversi valori di soglia scelti dall'utente.

Segmentate tutte le immagini, sono state imposte ulteriori condizioni di soglia per ignorare e partizionare certe parti di dati: ad esempio per ignorare fluido cerebrospinale e corteccia cerebrale ottenendo immagini che tengono conto solo della massa cerebrale oppure per individuare materi bianca e grigia. Sogliature accompagnate da operazioni di erosione dell'immagine per mezzo della funzione di matlab `imerode`. L'erosione di un'immagine consiste in un'elaborazione morfologica dell'immagine stessa al fine di migliorarla ed estrarre caratteristiche. Per le immagini in scala di grigi può essere così definita:

$$(f \ominus b)(s, t) = \min\{f(s+x, t+y) - b(x,y) \mid (s+x), (t+y) \in D; (x,y) \in D_b\}$$

ove  $f$  (immagine) e  $b$  (elemento strutturante: bulk di determinate forma e dimensione) sono funzioni che assegnano ad ogni punto di coordinate  $(x,y)$  un livello di grigio. Si sceglie il valore minimo di  $(f-b)$  in un intorno di  $f$  definito dal dominio di  $b$  con le condizioni che  $(s+x)$  e  $(t+y)$  debbano stare nel dominio di  $f$  e

$(x,y)$  nel dominio di  $b$  : ovvero l'elemento strutturante debba essere completamente contenuto in  $f$ .

```
thresh_seg(im)

% applicazione di soglie per ignorare parte di dati
D(D<=40) = 0;           %ignora bassi livelli (CSF & aria)
D(D>=100) = 0;        %ignore high levels (cranio e altro)

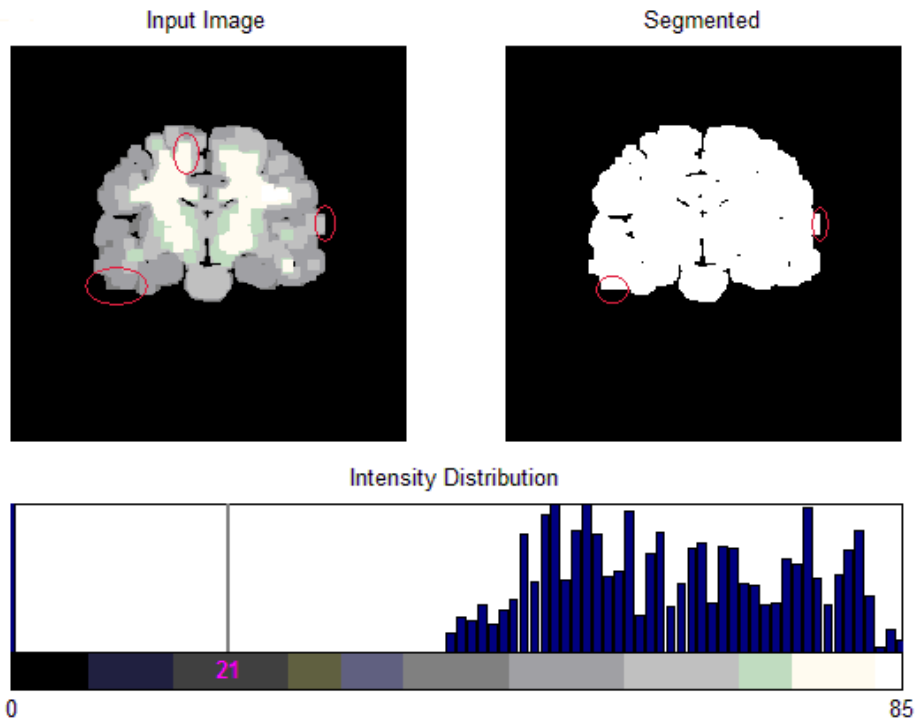
%erosione
blk = ones([2 5 5]);
D = imerode(D,blk);

%isolamento massa cerebrale
lev = graythresh(double(im)/max_level) * max_level;
bw = (D>=lev);
L = bwlabeln(bw);

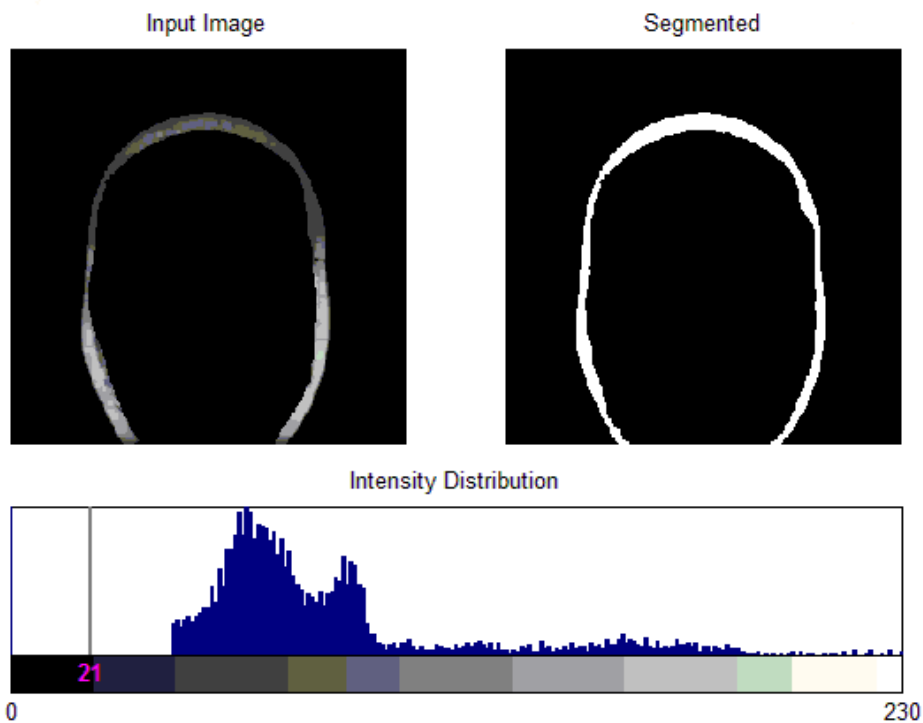
%partizionamento massa cerebrale
lev2 = 67;
L = zeros(size(D));    %0=no massa cerebrale
L(D<lev2 & D>0) = 2;  %2=material grigia
L(D>=lev2) = 3;       %3=materia bianca
```

❖ **Frammento di codice 2.2: funzione thresh\_seg.**

Il risultato di tutte le seguenti operazioni è mostrato nelle figure 2.4 e 2.5: osservando l'immagine relativa alla massa cerebrale, si può notare che la qualità della segmentazione, soprattutto nelle zone di confine (materia bianca/materia grigia e materia grigia/aria), non è accurata e dettagliata ma grossolana, con numerosi e lunghi tratti verticali ed orizzontali che non rispecchiano la realtà morfologica del cervello. Osservando invece l'immagine dello scalpo si nota come la segmentazione abbia un esito più definito: lo scalpo presenta infatti una struttura più regolare, a contrario della massa cerebrale che presenta un'anatomia assai più complessa, ricca di setti ed insenature.



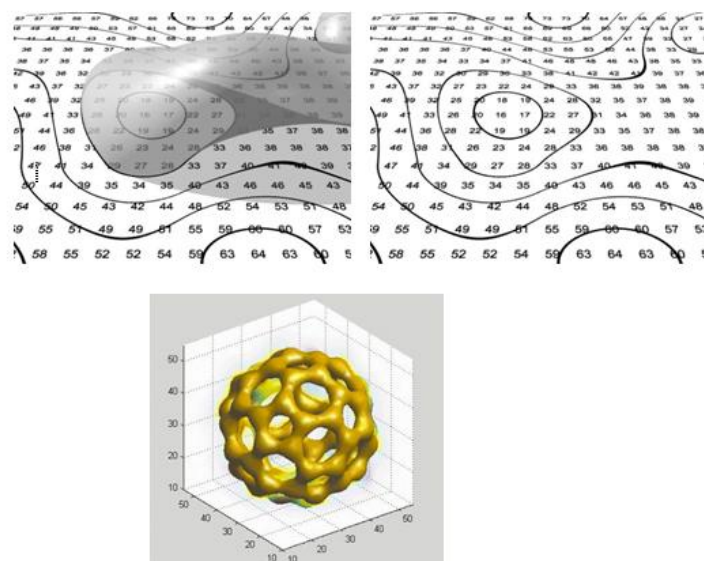
➤ **Figura 2.4.** Immagine segmentata della massa cerebrale



➤ **Figura 2.5.** Immagine segmentata dello scalpo

A partire poi dalle immagini segmentate sono stati ricostruiti i modelli 3D di cervello e scalpo utilizzando la funzione di matlab **isosurface**: tale funzione permette di realizzare superfici nello spazio tridimensionale data la descrizione di una funzione dei punti dello spazio da  $R^3$  a  $R$ . In particolare, per ogni punto dello spazio tale funzione associa un valore in  $R$ : una isosuperficie è definita come il luogo dei punti che hanno lo stesso valore.

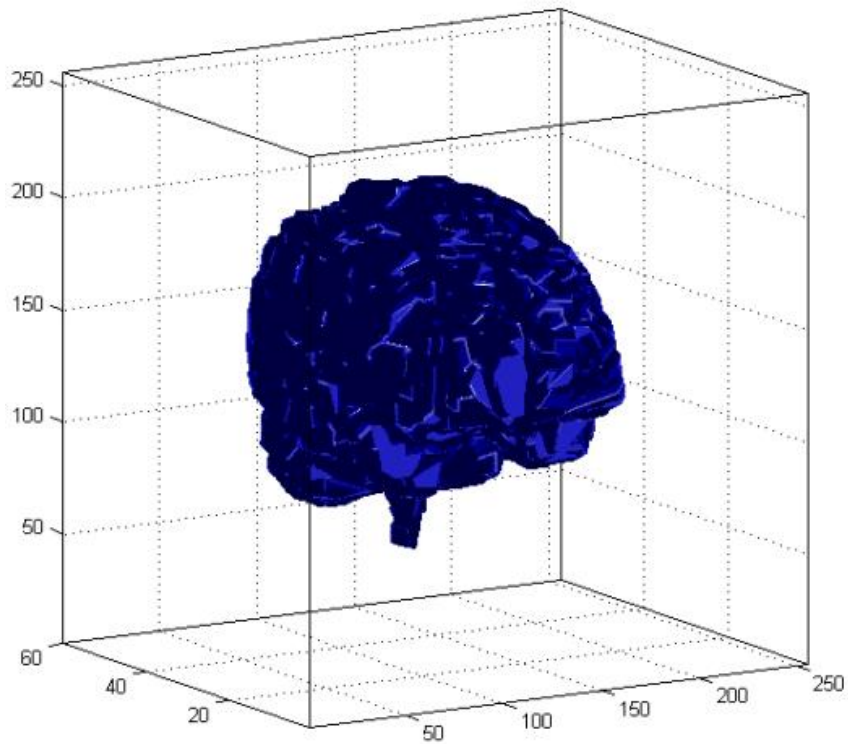
Un esempio concreto per una funzione continua nello spazio tridimensionale potrebbe essere la descrizione della pressione atmosferica: facendo riferimento alla immagine di destra, si descrive la pressione atmosferica misurata a terra tramite un valore reale per ogni punto della superficie terrestre sulla quale è stata effettuata la misura. Nella prima immagine (a destra) sono riportate linee che uniscono i punti in cui i valori numerici sono 20, 30, 40, 50 e 60. Se ora, con uno sforzo di immaginazione, si potessero avere i dati di pressione per ogni livello di altitudine dal suolo, si potrebbero sovrapporre piani di misurazioni analoghi al piano che riporta le misurazioni eseguite a terra: tramite tale sovrapposizione si avrebbe una funzione che mappa i punti dello spazio in valori di pressione atmosferica. Nella seconda immagine (a sinistra), si suppone quindi che ci siano valori numerici in ogni punto dello spazio. Ora è possibile unire i punti che hanno lo stesso valore numerico con superfici tridimensionali. In questo caso la superficie mostrata collega tutti i punti in cui il valore numerico è 30.



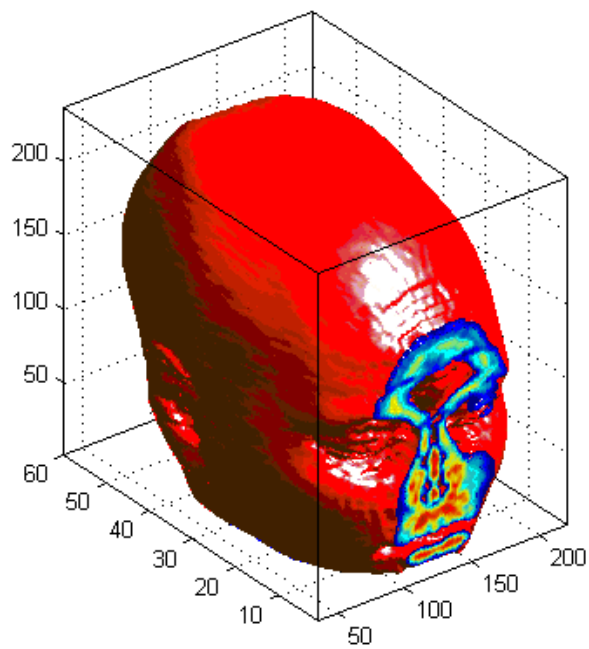
➤ **Figura 2.6. Esempi di isosuperfici**

### 2.2.4 Risultati

Riportiamo ora qui di seguito i modelli di cervello e scalpo ricostruiti:



➤ **Figura 2.6. Modello 3D del cervello**



➤ **Figura 2.7. Modello 3D dello scalpo**



```
p1 = patch(isosurface(volume,.5), ...  
          'FaceColor','red','EdgeColor','none');  
view(3); axis vis3d tight  
camlight; lighting phong;grid on;box on;
```

### ❖ Frammento di codice 2.3

La segmentazione a soglia è un metodo molto semplice e veloce dal punto di vista computazionale, in pochi secondi infatti (~ 10s) riusciamo a segmentare 60 immagini. I risultati ottenuti non sono però accettabili: osservando il modello del cervello, possiamo affermare che le sembianze sono effettivamente quelle di un cervello reale ma guardando al dettaglio non c'è alcuna traccia ad esempio delle tipiche insenature della superficie cerebrale: la grossolana segmentazione che è stata fatta (Figura 2.4) fa sì che non vengano presi in considerazione dettagli rilevanti nella ricostruzione poi dei modelli, ed in particolar modo proprio in quello del cervello ove la superficie risulta particolarmente spigolosa e formata da superfici più o meno grandi unite l'una all'altra. Meno problematica è invece la ricostruzione del modello dello scalpo, soprattutto grazie al fatto che, come detto in precedenza, esso presenta per sua natura un'anatomia meno complessa: osservando la figura 2.7, trascurando l'artefatto presente frontalmente dovuto al fatto che non ci sono ulteriori slices, possiamo affermare che il modello è sufficientemente esatto.

## 2.3 Contorni attivi

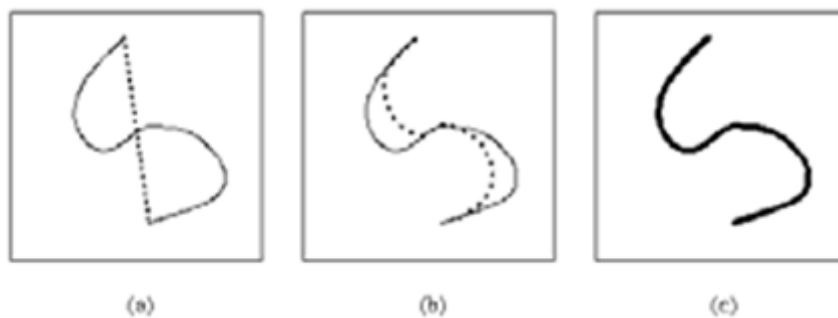
Descrivere ed analizzare contorni e strutture presenti in un'immagine anatomica con accuratezza ed efficienza può presentare dei problemi dovuti alla limitatezza dei dati e la variabilità e complessità delle forme anatomiche di interesse. La segmentazione modello-indipendente estrae spesso bordi indistinguibili e disconnessi ed in generale per una loro corretta interpretazione si devono fare ulteriori elaborazioni.

I modelli deformabili (*deformable models*), costituiscono un potente approccio, **modello-dipendente**, per la segmentazione e l'analisi delle immagini di strutture anatomiche, includendo sia i vincoli forniti dai dati dell'immagine sia l'informazione nota a priori sulla posizione, misura e forma della struttura di

interesse. Matematicamente i modelli deformabili si basano su concetti di (1) geometria: per rappresentare la forma dell'oggetto usano splines con un certo numero di gradi di libertà (una *spline di ordine  $d$*  è una funzione polinomiale costante a tratti di ordine  $d$ , che consiste di segmenti polinomiali concatenati (*span*), ognuno di ordine polinomiale  $d$ , uniti insieme in punti chiamati *breakpoint*); (2) fisica: ci si rifà alla teoria dell'elasticità per vincolare le possibili variazioni della forma dell'oggetto nel tempo e nello spazio (si limitano così i gradi di libertà delle splines); (3) teoria dell'ottimizzazione: i modelli vengono usati per fittare i dati misurati.

L'interpretazione fisica dei modelli deformabili è quella di corpi elastici che rispondono naturalmente a forze e vincoli ad essi applicati; ad essi sono associate funzioni di energia di deformazione.

Gli *snakes* (o contorni attivi) rappresentano un caso particolare di modelli deformabili: sono curve continue (aperte o chiuse) che cominciano a deformarsi da un punto iniziale per adattarsi alla forma dell'oggetto (Figura 2.8).



➤ **Figura 2.8: (a) posizione iniziale dello snake definita vicino al contorno vero (linea tratteggiata); (b) e (c) iterazioni dello snake che minimizza l'energia per raggiungere il contorno vero.**

Si pone sull'immagine una spline (iniziale) e si minimizza la sua energia, che dipende dalla sua forma e posizione nell'immagine. I minimi locali di questa energia corrispondono alle proprietà desiderate dell'oggetto che si vuole segmentare. A differenza di molti altri modelli di immagini, lo snake è attivo, cioè esibisce un comportamento dinamico, sempre minimizzando il suo funzionale di energia.

### 2.3.1 Formulazione matematica

Lo *snake* è una curva parametrica nel piano immagine  $(x,y)$ :

$$\mathbf{u}(s)=(x(s),y(s))$$

ove  $x(s)$  e  $y(s)$  sono le coordinate lungo il contorno e  $s \in [0,1]$  è la lunghezza normalizzata dell'arco lungo il contorno.

Alla curva è associato un funzionale:

$$E_{snake}(u) = \int_0^1 E_{snake}(u(s)) ds = \int_0^1 E_{int}(u(s)) ds + \int_0^1 E_{est}(u(s)) ds \quad (1)$$

Ove  $E_{int}$  è l'energia interna (dello snake) ed  $E_{est}$  è l'energia esterna (dell'immagine e dei vincoli). L'energia interna deformabile è responsabile principalmente della flessibilità e dell'allungamento del contorno; per forze interne si intendono forze di tensione che fanno muovere lo snake come una serie di molle e forze di rigidità che fanno resistere lo snake ai piegamenti. L'energia interna penalizza dunque l'allungarsi ed il raggomitolarsi della curva e la possiamo riscrivere nel seguente modo:

$$E_{int} = \alpha(s) \frac{|\partial u|^2}{|\partial s|^2} + \beta(s) \frac{|\partial^2 u|^2}{|\partial^2 s|^2} \quad (2)$$

Ove  $\alpha$  controlla l'elasticità del contorno mentre  $\beta$  la sua rigidità.

Per quel che riguarda l'energia esterna (o energia dell'immagine) questa cambierà a seconda delle varie applicazioni. Le forze esterne dell'immagine sono derivate dalle caratteristiche dell'oggetto che si vuole descrivere con lo snake: potenziali esterni che forzano lo snake verso particolari caratteristiche dell'oggetto vengono imposti o dall'utente o da altri processi ad alto livello. Un funzionale di immagine semplice è ad esempio l'intensità:

$$E_{linea} = I(x,y)$$

con questo funzionale lo snake viene attratto dalle aree scure dell'immagine.

Un funzionale basato invece sugli edges è:

$$E_{\text{edge}} = -|\Delta I(x,y)|^2$$

questo funzionale invece attrae lo snake a contorni con un alto valore di gradiente, ovvero verso quei pixel che sono sicuramente un edge.

Il contorno attivo può essere attratto da entrambe le caratteristiche linee e bordi e viene allora definito il seguente funzionale:

$$E_{\text{immagine}} = \omega_{\text{linea}} E_{\text{linea}} + \omega_{\text{edge}} E_{\text{edge}}$$

in questo caso il contorno attivo viene attratto a linee scure o chiare in dipendenza da  $\omega_{\text{linea}}$ .

Un potenziale che viene usato sempre per far tendere lo snake ai bordi dell'immagine è il GVF, che implementa il flusso del vettore gradiente:

$$E_{\text{immagine}} = -|\nabla(G_{\sigma}(x,y) * I(x,y))|^2$$

Dove  $I(x,y)$  rappresenta la luminosità dell'immagine e  $G_{\sigma}$  è il kernel di un filtro gaussiano di deviazione standard  $\sigma$ ; il filtraggio con il filtro gaussiano rende i contorni più sfuocati ma aumenta notevolmente la capacità dello snake di catturare tali contorni.

Il contorno dell'oggetto si trova nella posizione in cui lo snake raggiunge un minimo locale di energia e si dovrà dunque minimizzare il seguente funzionale:

$$E_{\text{snake}}(\mathbf{u}) = E_{\text{int}}(\mathbf{u}) + E_{\text{ext}}(\mathbf{u});$$

E lo snake che minimizza questo funzionale deve soddisfare alla seguente equazione:

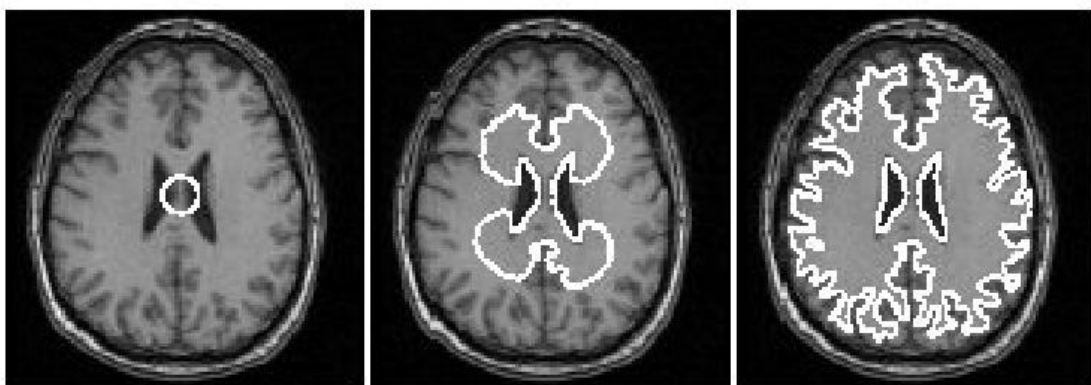
$$-\frac{\partial}{\partial s} \left( \alpha(s) \frac{\partial \mathbf{u}}{\partial s} \right) + \frac{\partial^2}{\partial s^2} \left( \beta(s) \frac{\partial^2 \mathbf{u}}{\partial s^2} \right) + \nabla E_{\text{ext}}(\mathbf{u}(s)) = 0 \quad (2)$$

Possiamo allora dire che la segmentazione con i contorni attivi è una procedura di minimizzazione dell'energia. Esistono diverse tecniche per minimizzare

l'energia dello snake, tra le più note citiamo la programmazione dinamica e gli approcci greedy. La programmazione dinamica è una tecnica algoritmica che consente di affrontare problemi dotati di un'apparente complessità esponenziale: si riduce il problema originale a sottoproblemi e si calcolano le soluzioni di tutti i sottoproblemi possibili, e a partire da sottosoluzioni si ricavano nuove sottosoluzioni, fino a risolvere il problema originale. La fase di suddivisione del problema iniziale è molto delicata, perché una frantumazione eccessiva può condurre a un numero di sottoproblemi troppo grande, ma d'altra parte senza un numero sufficiente di sottosoluzioni non sarà possibile ricostruire quella desiderata. Una volta fatta questa analisi, viene costruita una tabella che contiene in ogni suo elemento una sottosoluzione (o i dati sufficienti a ricostruirla). Una volta che la tabella è riempita, è facile desumere la soluzione del problema originale.

Negli approcci greedy si stima direttamente la distanza e la curvatura tra i vari punti dello snake: con algoritmi greedy si intende **una classe di algoritmi utilizzati per problemi di ricerca di una soluzione ottima**. L'idea è quella di risolvere un problema in modo ciclico scegliendo ogni volta la soluzione migliore in quel momento; questi tipi di algoritmi euristici forniscono spesso soluzioni sub-ottime ma vengono scelti per la loro ridotta complessità computazionale.

La procedura è interattiva: l'utente disegna un contorno approssimato sull'immagine e inizia una simulazione dinamica dove il contorno viene deformato fino a che raggiunge uno stato di equilibrio.



➤ **Figura 2.9: Esempio di snake che si sviluppano all'interno di sezioni cerebrali per evidenziare la materia grigia.**

### 2.3.2 Implementazione

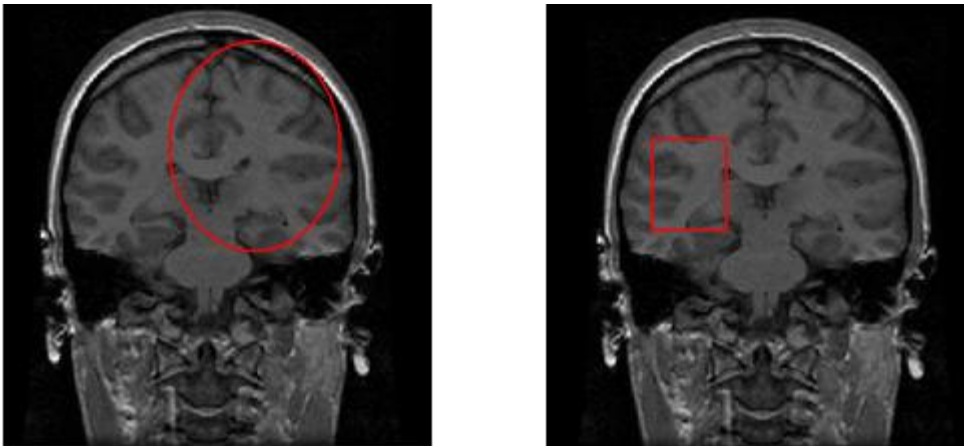
Illustreremo ora i risultati ottenuti applicando questa tecnica di segmentazione: il data set utilizzato è quello impiegato nel test della tecnica precedente, costituito da 60 file dicom.

Innanzitutto l'utente deve disegnare un contorno iniziale sull'immagine:

```
...  
get(ud.imageId);hold on;  
[c,h] = contour(phi,[0 0],cl{1},'Linewidth',2)
```

❖ **Frammento di codice 2.6: funzione contour.**

Gli snakes sono di solito inizializzati come linee chiuse che circondano le regioni di interesse. Per tracciare il contorno sull'immagine ci sono comunque diverse possibilità: può essere plottato automaticamente quando è caricata l'immagine in una data posizione, oppure è l'utente che lo disegna nella posizione (al centro, in alto a destra, in basso a sinistra, . . .) e delle forma e dimensioni desiderate (rettangolare, ellissoidale, . . .).



➤ **Figura 2.10: Esempi di contorni iniziali disegnati sull'immagine.**

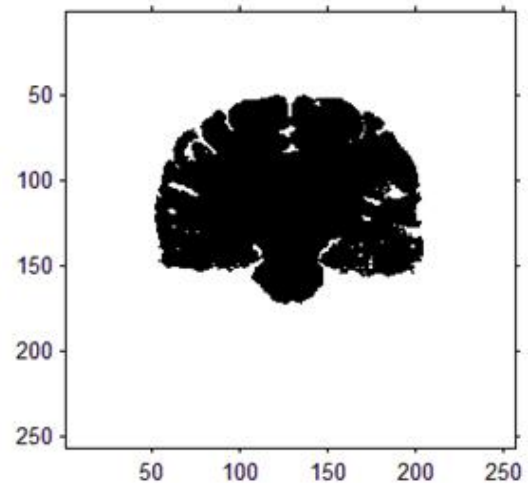
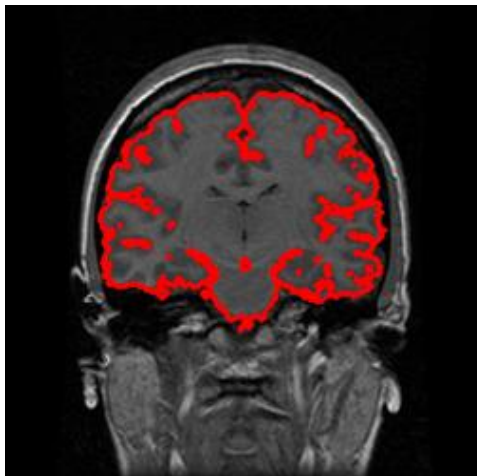
Essi poi si muovono contraendosi finché l'effetto dell'energia di trasporto non li fa avvicinare e poi fermare sulle linee desiderate secondo una serie di criteri e condizioni stabilite: ovvero è soddisfatto un criterio di stop.

```
...
sc = 1; i = 1;
while( i<size_out && sc )
    x = Lo(1,i);y = Lo(2,i);
    if F(x,y)>0
        sc = 0;
    end
    i = i+1;
end
...

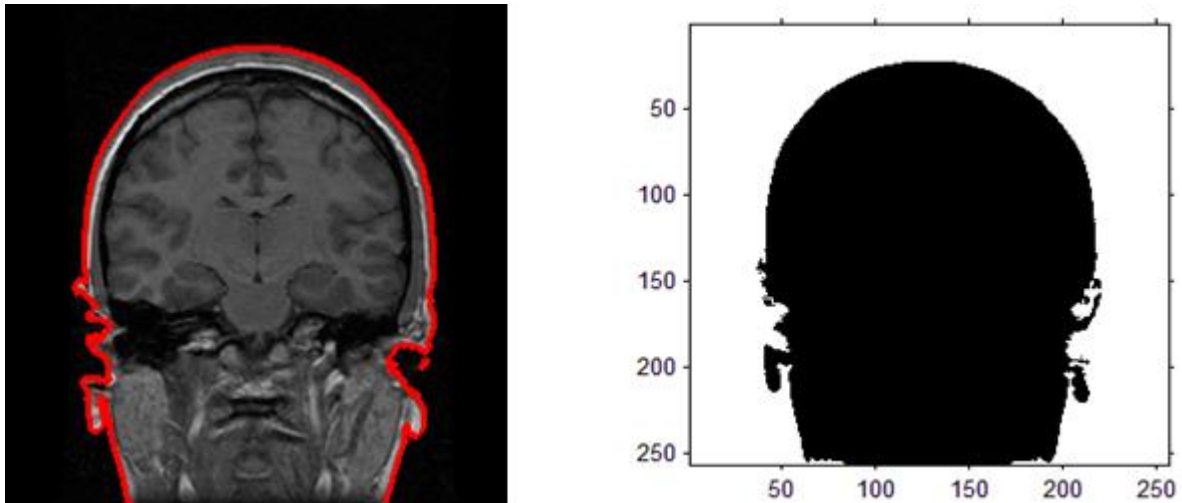
if (stop_cond==0)
    feature = zeros(size(phi));
    [x,y] = find(abs(phi) < 2);
    feature(x,y) = -(img(x,y) - u).^2 + (img(x,y) - v).^2;
    feature = feature./max(abs(feature(:)));
end
```

❖ **Frammento di codice 2.7: funzione stop\_crit.**

Nelle seguenti immagini sono riportati i risultati ottenuti: le figure 2.11 e 2.12 si riferiscono alla medesima slice da cui è stata estratta la massa cerebrale prima e lo scalpo poi; le figure 2.13 e 2.14 riportano i modelli 3D ricostruiti da tutte le slice segmentate del dataset.



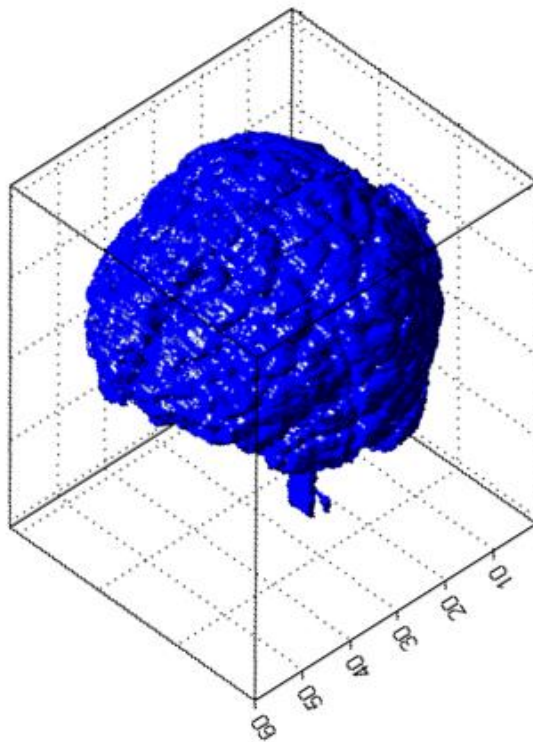
➤ **Figura 2.11. Segmentazione della massa cerebrale.**



➤ **Figura 2.12. Segmentazione dello scalpo.**

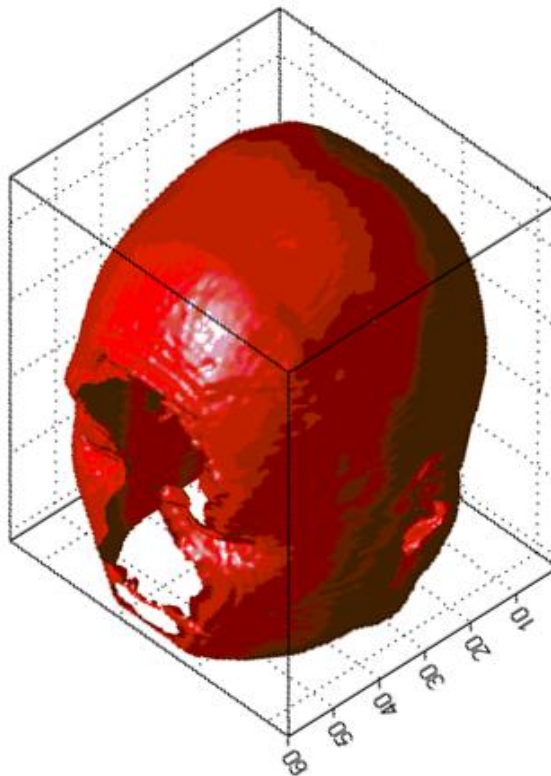
### 2.3.3 Risultati

I modelli 3D sono stati ricostruiti sempre utilizzando la funzione di matlab `isosurface`.



➤ **Figura 2.13. Modello del cervello ricostruito da immagini segmentate con contorni attivi**





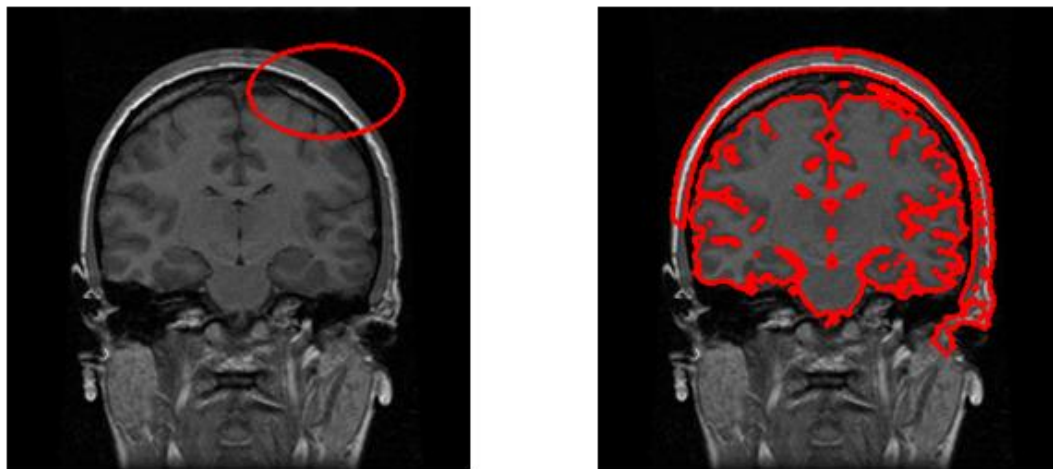
- **Figura 2.13. Modello dello scalpo ricostruito da immagini segmentate con contorni attivi.**

Confrontando questi modelli con quelli ottenuti con la tecnica del thresholding (Fig. 2.6 e 2.7) si possono fare le seguenti considerazioni: il modello del cervello ottenuto dall'applicazione dei contorni attivi (Fig. 2.12) è migliore di quello ottenuto applicando il thresholding (Fig. 2.6); in quest'ultimo infatti come detto in precedenza non c'è traccia delle tipiche insenature ed è particolarmente spigoloso, ora invece si è ottenuto un modello dalle forme più arrotondate e con traccia, seppur in modo confuso delle insenature, quindi più vicino all'anatomia reale. Per quanto riguarda invece i modelli di scalpo ricostruiti con le due tecniche (Fig. 2.7 e 2.13), non ci sono differenze rilevanti: entrambi sono sufficientemente dettagliati, indipendentemente dalla tecnica usata è semplice estrarre dall'immagine il contorno esterno dello scalpo (di forma più regolare) perché c'è una forte variazione dei livelli di grigio man mano che ci si sposta dallo sfondo ai pixel del contorno ed è semplice individuarlo e catturarlo. I contorni attivi permettono di ricostruire modelli migliori, perché segmentano più

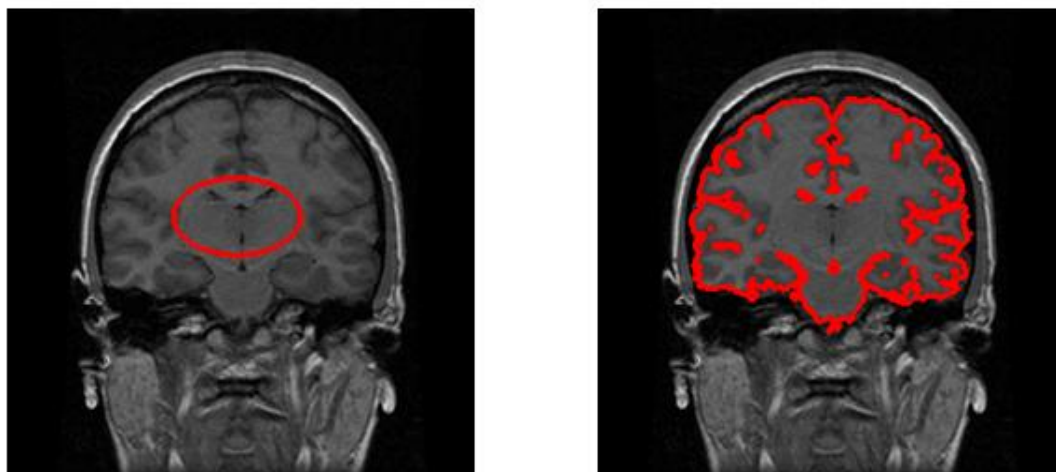
accuratamente; le maggiori difficoltà si incontrano però anche per essi nella segmentazione della massa cerebrale.

### 2.3.4 Limitazioni e problematiche dei contorni attivi

Nelle varie prove effettuate, si sono riscontrati diversi fattori di incertezza e instabilità; in alcuni casi ad esempio si è riscontrata una certa variabilità delle strutture segmentate a seconda della posizione in cui viene definito il contorno iniziale:

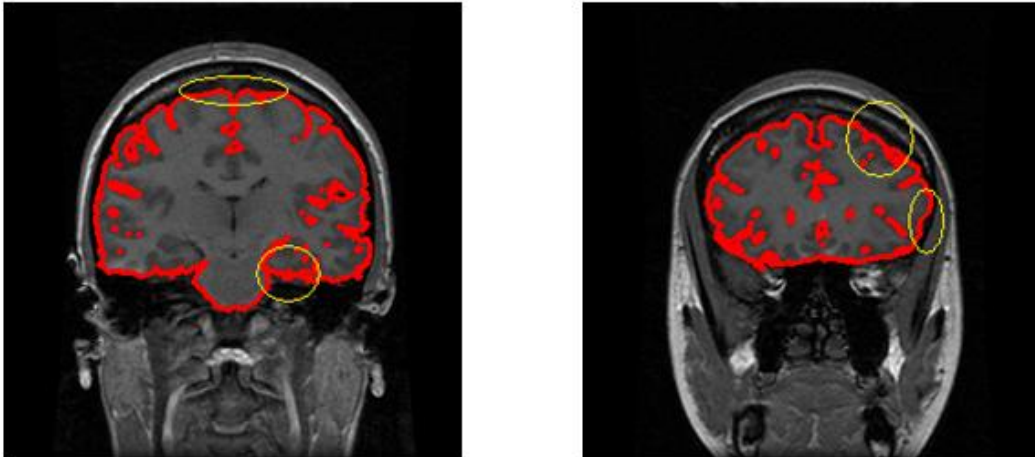


- **Figura 2.14:** Inizializzazione del contorno (sinistra); strutture segmentate con il dato contorno iniziale (destra).



- **Figura 2.15:** Inizializzazione del contorno (sinistra); strutture segmentate con il dato contorno iniziale (destra).

Osservando poi alcune immagini segmentate si riscontra l'incapacità del contorno attivo di modellare adeguatamente e in modo completo parte delle strutture d'interesse: in figura 2.16 si vede ad esempio che la massa cerebrale nelle due slice non viene catturata totalmente.



➤ **Figura 2.16: esempi di irregolarità della segmentazione.**

La segmentazione e la ricostruzione dei modelli, come detto nel capitolo introduttivo devono avvenire in tempi ragionevolmente brevi e ridurre al minimo l'interattività del medico. Per questo dataset di 60 immagini, ipotizzando che nel caso peggiore il medico deve caricare la singola immagine, disegnare su di essa il contorno iniziale, avviare la simulazione dinamica, il tempo necessario si aggira intorno ai 65 minuti; se il load dell'immagine e l'inizializzazione del contorno avvengono in automatico il tempo necessario è di circa 30 minuti. Ovviamente per dataset più ampi, i tempi di elaborazione aumentano notevolmente (ad esempio con un dataset di 120 immagini si oltrepassa l'ora).



# Capitolo 3

## La Segmentazione Unificata

### 3.1 Metodo

La segmentazione di immagini anatomiche del cervello in materia grigia (GM), materia bianca (WM), e fluido cerebrospinale (CSF), può esser fatta secondo due diversi approcci:

- Il primo approccio consiste in una *tissue classification* ove ciascun voxel è assegnato ad una classe di tessuto conformemente alla sua intensità. Per fare un'assegnazione corretta, la distribuzione di intensità di ciascuna classe di tessuto necessita essere caratterizzata da voxels rappresentativi di ogni classe. Una selezione automatica dei voxel rappresentativi di ogni classe viene effettuata attraverso una prima registrazione del cervello in uno spazio standard e successivamente la classificazione dell'intensità di tutti i voxel viene modellata nell'istogramma delle intensità con un mix di Gaussiane e pesata utilizzando una mappa di probabilità a priori riferita allo spazio standard.
- Il secondo approccio consiste in una registrazione: un cervello standard (*template*) che è già stato segmentato e in cui ogni struttura interna è stata identificata viene deformato per allinearlo al cervello da segmentare permettendo l'identificazione delle diverse strutture.

Con la segmentazione unificata si vuole accorpate questi due approcci in un unico impianto. Si fa una iniziale normalizzazione spaziale seguita dalla segmentazione nelle differenti classi di tessuti, smoothing e uso di una probabilità a priori per calcolare la probabilità statistica di appartenenza di ogni voxel ad ogni classe. Il cervello di ogni singolo soggetto viene normalizzato con un cervello standard affinché le materie grigie dei due cervelli combacino perfettamente; questo per eliminare strutture come lo scalpo che possono introdurre effetti contraddittori nella registrazione. Successivamente vengono utilizzate le mappe probabilistiche a priori, che, combinate con l'intensità di ogni voxel attraverso la regola di Bayes, restituiscono la probabilità a posteriori.

Questa procedura sembrerebbe di primo acchito circolare: una registrazione iniziale richiede una classificazione iniziale (la segmentazione deve essere nota per poter allineare i due cervelli) e una classificazione iniziale richiede a sua volta una registrazione iniziale (il prior deve essere allineato con il cervello in esame). Il ragionamento circolare è stato quindi risolto combinando entrambe le metodologie con un singolo modello di tipo generativo. Modello che include anche parametri che tengono conto della non uniforme intensità delle immagini dovuto al bias del campo magnetico. Massimizzando la probabilità a posteriori, vengono calcolati i parametri del modello alternando i tre step: classificazione, correzione del bias e registrazione. Questo approccio risulta più efficace rispetto alla singola ottimizzazione di ogni singolo step.

Approcci integrati come quello appena descritto sono molto complessi e difficili con tempi di esecuzione a volte lunghi; d'altro canto però si ottengono risultati molto accurati.

### 3.1.1 Mix di Gaussiane

La distribuzione di intensità di un'immagine è modellata da un mix di  $K$  Gaussiane (*mixture of Gaussians*, MOG), che corrispondono ai cluster a cui apparterrà ciascun voxel. Ciascuna delle gaussiane viene descritta da una media ( $\mu_k$ ), una varianza ( $\sigma_k^2$ ) e da un coefficiente detto *mixing proportion* che rappresenta il rapporto sul totale dei voxel che appartengono alla  $k$ -esima gaussiana ( $\gamma_k$ , ove  $\sum_{k=1}^K \gamma_k = 1$  con  $\gamma_k \geq 0$ ); si deve determinare il fit del MOG, ovvero massimizzare la probabilità di osservare gli  $I$  elementi

dell'immagine  $y$  data la parametrizzazione della gaussiana. La probabilità di ottenere un dato con intensità  $y_i$  sapendo che appartiene alla  $k$ -esima gaussiana ( $c_i = k$ ) parametrizzata da  $(\mu_k)$  e  $(\sigma_k^2)$  è:

$$P(y_i | c_i = k, \mu_k, \sigma_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{(y_i - \mu_k)^2}{2\sigma_k^2}\right) \quad (3.1)$$

La probabilità a priori di ciascun voxel, a prescindere dalla sua intensità, di appartenere alla  $k$ -esima gaussiana dato il coefficiente di mix dei voxel che appartengono ad essa è:

$$P(c_i = k | \gamma_k) = \gamma_k \quad (3.2)$$

Applicando la regola di Bayes per la probabilità congiunta, la probabilità di un voxel avente intensità  $y_i$  di appartenere al cluster  $k$ , chiamata anche *probabilità a posteriori* è:

$$P(y_i, c_i = k | \mu_k, \sigma_k, \gamma_k) = P(y_i | c_i = k, \mu_k, \sigma_k) P(c_i = k | \gamma_k) \quad (3.3)$$

Dati i parametri integrando tutti i  $K$  cluster si ottiene la probabilità del voxel  $y_i$  :

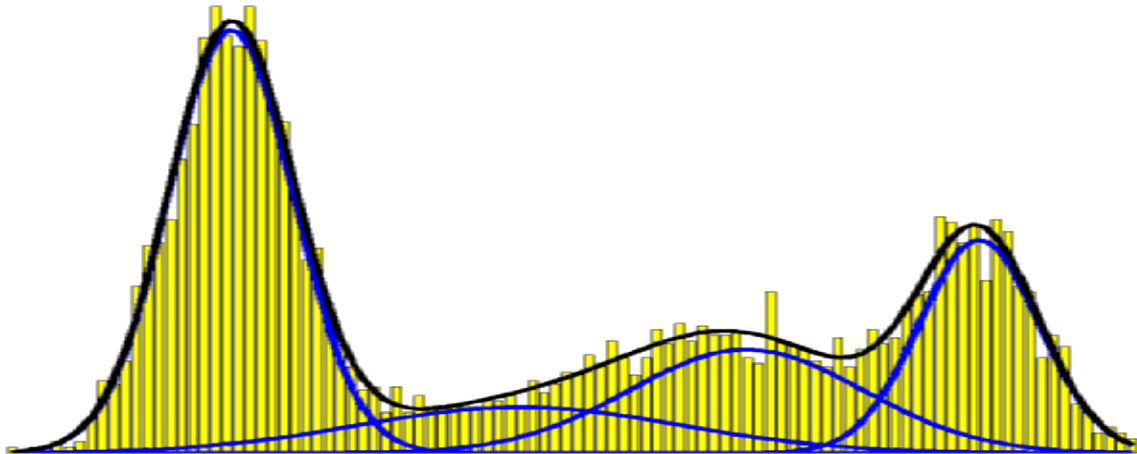
$$P(y_i | \mu, \sigma, \gamma) = \sum_{k=1}^K P(y_i, c_i = k | \mu_k, \sigma_k, \gamma_k) \quad (3.4)$$

La probabilità dell'intero data-set  $y$  si ottiene, assumendo che tutti gli elementi sono indipendenti, moltiplicando la probabilità di ciascun voxel  $y_i$  del dataset (cioè dalla probabilità congiunta di tutti i voxel):

$$P(y | \mu, \sigma, \gamma) = \prod_{i=1}^I P(y_i | \mu, \sigma, \gamma) = \prod_{i=1}^I \left( \sum_{k=1}^K \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{(y_i - \mu_k)^2}{2\sigma_k^2}\right) \right) \quad (3.5)$$

Tale probabilità è massimizzata quando la seguente funzione costo è minimizzata:

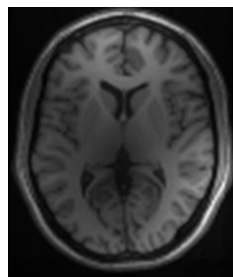
$$\mathcal{E} = -\log P(y|\mu, \sigma, \gamma) = -\sum_{i=1}^I \log \left( \sum_{k=1}^K \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp \left( -\frac{(y_i - \mu_k)^2}{2\sigma_k^2} \right) \right) \quad (3.6)$$



➤ **Figura 3.1:** Istogramma delle intensità dei voxel (in giallo) sovrapposto alle gaussiane di ciascuna classe (in blu) e le intensità modellizzate dalla somma di gaussiane (linea nera).

### 3.1.2 Bias

Le immagini MR sono molto spesso alterate nell'intensità a causa di artefatti dovuti soprattutto alla disomogeneità del campo magnetico all'interno della macchina. Questi artefatti non costituiscono un problema per l'ispezione visiva dell'immagine ma ovviamente impediscono una corretta segmentazione.



➤ **Figura 3.2:** disomogeneità dell'intensità causata da imperfezioni del campo magnetico.



I modelli parametrici per la correzione del bias sono spesso parte integrante della classificazione. Tra i numerosi modelli disponibili, qui è impiegato un modello parametrico in cui l'intensità del segnale osservato  $y_i$  è costituito dal segnale vero  $\mu_i$  e da un rumore gaussiano aggiuntivo  $n_i$  ridotti ad un certo fattore di bias  $\rho_i$

$$y_i = \frac{\mu_i + n_i}{\rho_i} \quad (3.7)$$

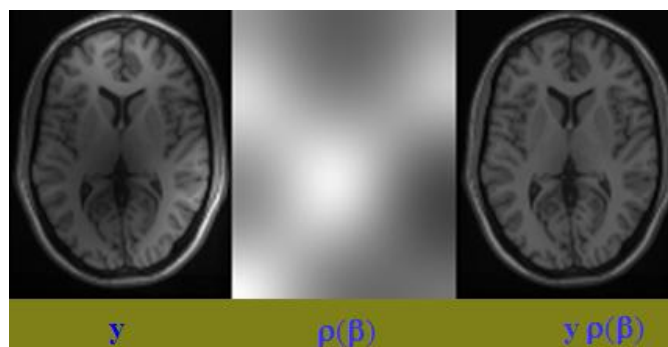
In questo modello la correzione del bias è inclusa nel MOG da ulteriori parametri che tengono conto della variazione dell'intensità dello smooth; il parametro che modella questa variazione è  $\rho_i(\beta)$  ove  $\beta$  è un vettore di parametri non noti. Le intensità del  $k$ -esimo cluster sono assunte essere distribuite normalmente con media  $\mu_k/\rho_i(\beta)$  e varianza  $(\sigma_k/\rho_i(\beta))^2$ . Quindi la probabilità di ottenere l'intensità  $y_i$  dal cluster  $k$ -esimo, data la sua parametrizzazione, è:

$$P(y_i | c_i = k, \mu_k, \sigma_k, \beta) = \rho_i(\beta) \frac{1}{(2\pi\sigma_k^2)^{\frac{1}{2}}} \exp\left(-\frac{(\rho_i(\beta)y_i - \mu_k)^2}{2\sigma_k^2}\right) \quad (3.8)$$

La funzione obiettivo (3.6) della classificazione diventa ora:

$$\mathcal{E} = -\sum_{i=1}^I \log\left(\rho_i(\beta) \sum_{k=1}^K \frac{\gamma_k}{(2\pi\sigma_k^2)^{\frac{1}{2}}} \exp\left(-\frac{(\rho_i(\beta)y_i - \mu_k)^2}{2\sigma_k^2}\right)\right) \quad (3.9)$$

Il modello parametrizza il bias come un'esponenziale di una combinazione lineare di funzioni di base a bassa frequenza.



➤ **Figura 3.3: Correzione del bias**

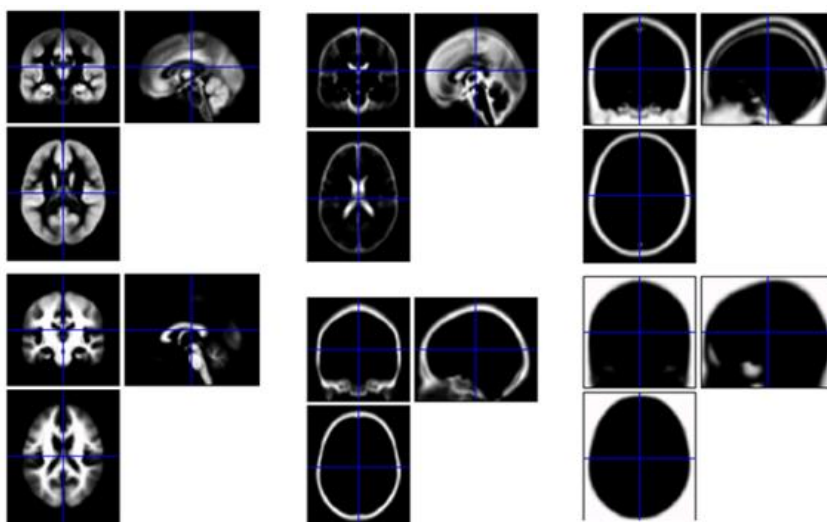
Modelli alternativi a quello appena visto sono i seguenti:

$$y_i = \frac{\mu_i}{\rho_i} + n_i \quad (1) \quad \text{logy}_i = \log\mu_i - \log\rho_i + n_i \quad (2) \quad (3.10)$$

Il primo è simile a quello visto in precedenza con l'unica differenza che qui il rumore è aggiunto dopo che il segnale è stato scalato; nel secondo viene calcolata prima la trasformata logaritmica dei dati.

### 3.1.3 Mappe probabilistiche

Anziché considerare stazionarie le probabilità a priori basate sul coefficiente di mix, sono usate informazioni addizionali basate su immagini di cervello di altri soggetti. Le mappe probabilistiche a priori (*prior*) sono generate registrando insieme un grande numero di soggetti, assegnando i voxels alle differenti classi di tessuto e mediandole tra di esse. Le mappe consistono in una versione modificata dell' ICBM Tissue Probabilistic Atlas costituito da 452 immagini T1-pesate. Per mezzo di esse si ottiene la probabilità di appartenenza dei voxel dell'immagine a ciascuna classe indipendentemente dal valore delle intensità. Sono usate sei mappe: materia grigia, materia bianca, fluido cerebrospinale, osso, tessuto molle, e una sesta classe ottenuta come 1 meno la somma delle altre 5 (aria).



➤ **Figura 3.4:** Mappe probabilistiche a priori di materia grigia, materia bianca, fluido cerebrospinale, osso, tessuto molle, aria.

È possibile ora modificare l'equazione (3.9) tenendo conto dei prior: non si usa un fattore di mix stazionario ( $P(c_i = k|\gamma) = \gamma_k$ ) e la probabilità del voxel  $i$ -esimo di appartenere al cluster  $k$ -esimo è:

$$P(c_i = k|\gamma) = \frac{\gamma_k b_{ik}}{\sum_{j=1}^K \gamma_j b_{ij}} \quad (3.11)$$

con  $b_{ik}$  che rappresenta la probabilità tessutale per la classe  $k$  del voxel  $i$ -esimo. Il numero di Gaussiane usate per rappresentare la distribuzione di intensità di ogni classe è in genere maggiore di uno: in altre parole una mappa può far parte di cluster diversi. Le ragioni di questa scelta sono diverse: infatti una sola Gaussiana non riesce a modellare la distribuzione di intensità dei voxel di ogni classe perché solitamente un voxel non contiene solamente un solo tipo di tessuto (*partial volume effect*), soprattutto in quelle zone in cui si ha un graduale passaggio da una classe ad un'altra (la distinzione tra GM e WM non può essere rappresentata attraverso una linea netta) oppure in certe strutture interne, come il talamo, costituito sia da WM, che da GM. La maggior parte dei modelli utilizza una gaussiana per ogni classe più una gaussiana per ogni intervallo tra classi diverse, che contiene l'insieme dei voxel che non appartengono né all'una, né all'altra classe, con un rapporto di 50/50. Qui si assume semplicemente che la distribuzione di ogni classe non sia Gaussiana e, in accordo con questo, viene usata una combinazione di queste: il numero di gaussiane che vengono utilizzate per la GM sono 2, per la WM sono 2, per il CSF sono 2 per il tessuto molle sono 2, per l'osso ancora 2 e per la rimanente sesta classe (ambiente esterno, l'aria) sono 4.

L'equazione (3.11) è migliorata permettendo alle mappe originali, registrate prima in uno spazio standard, di deformarsi in accordo ad un set di parametri  $\alpha$ :

$$P(c_i = k|\gamma, \alpha) = \frac{\gamma_k b_{ik}(\alpha)}{\sum_{j=1}^K \gamma_j b_{ij}(\alpha)} \quad (3.12)$$

La registrazione in uno spazio standard avviene attraverso la determinazione dei parametri di una trasformazione Affine ottimizzando una funzione obiettivo

somma di log-likelihood e di termini a priori: il risultato è una matrice di trasformazione 4x4 con parametri di traslazione, rotazione, shear e zoom.

Presi in considerazione questi fattori, la funzione obiettivo (3.9) diventa:

$$\mathcal{E} = -\sum_{i=1}^I \log \left( \frac{\rho_i(\beta)}{\sum_{k=1}^K y_k b_{ik}(\alpha)} \sum_{k=1}^K \frac{y_k b_{ik}(\alpha)}{\sqrt{2\pi\sigma_k^2}} \exp \left( -(\rho_i(\beta)y_i - \mu_k)^2 / 2\sigma_k^2 \right) \right) \quad (3.13)$$

Le deformazioni delle mappe sono parametrizzate da una combinazione lineare di dodici funzioni cosenosoidali di base: questa non è una via molto precisa per codificare le deformazioni, ma si riesce a modellare la variabilità della forma di tutto il cervello.

### 3.1.4 Regolarizzazione

Per la classificazione delle immagini un fattore importante è la distinzione tra la variazione di intensità a causa di artefatti intrinseche delle immagini MR, e quelle invece dovute all'appartenenza a diverse tipologie di tessuto. La prima infatti è una variazione che tende essere di bassa frequenza (smooth) nello spazio considerato, la seconda invece tende ad avere delle componenti in alta frequenza. Una più accurata stima del bias del campo magnetico può essere ottenuta includendo conoscenze a priori sulla distribuzione del bias stesso. Se per esempio è noto che non c'è non uniformità di intensità, o ce né in minima quantità, è preferibile penalizzare per i parametri grandi valori di disomogeneità di intensità. Questa regolarizzazione può essere inserita in un contesto Bayesiano. Similmente è possibile modellare la disomogeneità di intensità a causa di una registrazione scorretta: se si ha a disposizione conoscenza a priori della probabilità di distribuzione della forma del cervello, allora questa informazione può essere usata per regolarizzare le deformazioni. Non è però possibile avere una completa specificazione della probabilità di distribuzione.

Inclusi i termini di regolarizzazione, la funzione obiettivo likelihood da massimizzare al variare dei parametri che devono essere stimati (*fitting* del modello), è:

$$P(y, \beta, \alpha | \gamma, \mu, \sigma^2) = P(y | \beta, \alpha, \gamma, \mu, \sigma^2) P(\beta) P(\alpha) \quad (3.14)$$

Passando alla trasformata logaritmo Minimizzare la (3.14) equivale a minimizzarne il logaritmo, della (3.14), ciò quindi equivale a minimizzare:

$$\mathcal{F} = -\log P(\mathbf{y}, \boldsymbol{\beta}, \boldsymbol{\alpha} | \gamma, \mu, \sigma^2) = \mathcal{E} - \log P(\boldsymbol{\beta}) - \log P(\boldsymbol{\alpha}) \quad (3.15)$$

con la densità di probabilità dei parametri spaziali che è assunta essere una gaussiana multivariata a media zero ( $P(\boldsymbol{\alpha}) = N(\mathbf{0}, C_{\alpha})$ ;  $P(\boldsymbol{\beta}) = N(\mathbf{0}, C_{\beta})$ ). Il prior sul bias del campo magnetico è generato da uno smoothing di un rumore gaussiano random a media zero, da un ampio kernel gaussiano (di circa 70 mm FWHM) e successivamente convertito ad esponenziale.

### 3.1.5 Ottimizzazione e stima dei parametri

Definita la funzione obiettivo, lo scopo è minimizzarla al variare dei parametri: i parametri ottimi stimati saranno quelli in cui la funzione ha un minimo ricavati nel punto minimo della funzione. Non c'è una soluzione in forma chiusa per determinare i parametri e il valore ottimo di ogni parametro dipende dal valore di altri. Viene utilizzato un approccio Iterated Conditional Models (ICM): si inizia assegnando dei valori iniziale ai parametri e iterando la procedura fino a trovare una soluzione ottima; ciascuna iterazione alterna la stima di un parametro tenendo fissi gli altri al loro valore ottimo precedentemente trovato. I parametri sono aggiornati attraverso un algoritmo di Expectation Maximisation (EM), mentre il bias e le deformazioni sono tenuti fissi alle loro modalità.

Poichè l'intensità del bias è smooth, lo si può descrivere da un ristretto numero di parametri, utilizzando un approccio LM (Levenberg-Marquardt) per l'ottimizzazione. Fissati i parametri ed il bias, le deformazioni delle mappe sono ri-stimate.

La procedura converge ad un ottimo minimo locale, pertanto è necessario assegnare dei buoni valori iniziali dei parametri: i parametri per la clusterizzazione sono assegnati in maniera random mentre per i coefficienti per il bias e la deformazione non lineare sono settati a zero, ma viene stimata una iniziale registrazione Affine per allineare in maniera approssimativa il cervello con le mappe probabilistiche a priori.

### 3.1.5.1 Stima dei parametri $\mu, \sigma^2$ e $\gamma$

Per la stima dei parametri di clustering ( $\mu, \sigma^2$  e  $\gamma$ ) è sufficiente minimizzare la funzione obiettivo rispetto ad essi dato che non hanno effetti sui prior o sulla regolarizzazione. Definito il vettore dei parametri  $\theta = \{\mu, \sigma, \gamma, \alpha, \beta\}$ , sia  $q_{ij}$  una qualche distribuzione, la funzione obiettivo da minimizzare è:

$$\mathcal{E} \leq \mathcal{E}_{EM} = -\sum_{i=1}^I \log P(y_i | \theta) + \sum_{i=1}^I \sum_{k=1}^K q_{ik} \log \left( \frac{q_{ik}}{P(c_i=k|y_i, \theta)} \right) \quad (3.16)$$

Si utilizza l'approccio iterativo EM in cui si alterna un E-step (che minimizza  $\mathcal{E}_{EM}$  rispetto a  $q_{ik}$ ) ad un M-step (che minimizza  $\mathcal{E}_{EM}$  rispetto a  $\theta$ ). Il secondo termine dell'equazione (3.16) è minimo, cioè vale zero quando  $q_{ik} = P(c_i = k | y_i, \theta)$  e in questo caso si ottiene l'uguaglianza  $\mathcal{E} = \mathcal{E}_{EM}$ . Poiché  $q_{ik}$  non è considerato nel primo termine della (3.16), l'E-step dell' $n$ -esima iterazione consiste del seguenti setting:

$$q_{ik}^{(n)} = P(c_i = k | y_i, \theta^{(n)}) = \frac{P(y_i, c_i=k | \theta^{(n)})}{P(y_i | \theta^{(n)})} = \frac{p_{ik}}{\sum_{j=1}^K p_{ij}} \quad (3.17)$$

Dove  $p_{ik}$  è:

$$p_{ik} = \frac{\gamma_k b_{ik}(\alpha)}{\sum_{j=1}^K \gamma_j b_{ij}(\alpha)} (2\pi\sigma_k^2)^{-\frac{1}{2}} \exp\left(-\frac{(y_i(\beta) - \mu_k)^2}{2\sigma_k^2}\right) \quad (3.18)$$

$q_{ik}$  rappresenta la probabilità a posteriori di  $c_i = k$  a cui si è interessati.

L'M-step utilizza i valori aggiornati al passo  $n$  di  $q_{ik}$  per minimizzare  $\mathcal{E}$  rispetto a  $\theta$ .

L'equazione (3.16) può essere riformulata come:

$$\mathcal{E} = \mathcal{E}_{EM} = -\sum_{i=1}^I \sum_{k=1}^K q_{ik} \log P(y_i, c_i = k | \theta) + \sum_{i=1}^I \sum_{k=1}^K q_{ik} \log q_{ik} \quad (3.19)$$

Essendo il secondo termine indipendente da  $\theta$  l'M-step procede assegnando

nuovi valori ai parametri in modo che le derivate seguenti si annullino:

$$\frac{\partial \mathcal{F}}{\partial \mu_k} = \frac{\partial \mathcal{E}}{\partial \mu_k} = 0 \rightarrow \mu_k^{(n+1)} = \frac{\sum_{i=1}^I q_{ik}^{(n)} \rho_i(\beta) y_i}{\sum_{i=1}^I q_{ik}^{(n)}}$$

$$\frac{\partial \mathcal{F}}{\partial \sigma_k^2} = \frac{\partial \mathcal{E}}{\partial \sigma_k^2} = 0 \rightarrow (\sigma_k^2)^{(n+1)} = \frac{\sum_{i=1}^I q_{ik}^{(n)} \left( \mu_k^{(n+1)} - \rho_i(\beta) y_i \right)^2}{\sum_{i=1}^I q_{ik}^{(n)}}$$

$$\frac{\partial \mathcal{F}}{\partial \gamma_k} = \frac{\partial \mathcal{E}}{\partial \gamma_k} = 0 \rightarrow \frac{\sum_{i=1}^I q_{ik}^n}{\sum_{i=1}^I \frac{b_{ik}(\alpha)}{\sum_{j=1}^K \gamma_j^{(n)} b_{ij}(\alpha)}}$$

Con la  $\mathcal{F}$  ottenuta a partire dall'equazione (3.15).

### 3.1.5.2 Stima dei parametri $\beta$

La stima del bias viene fatta tenendo fissi gli altri parametri e migliorando la stima di  $\beta$  con un approccio di ottimizzazione LM (Newton-Rapson o Gauss-Newton). Per ogni iterazione sono richieste le derivate prima e seconda della funzione obiettivo rispetto ai parametri; viene utilizzato un fattore di scala  $\lambda$  che funge da trade-off tra velocità di convergenza e stabilità: bassi valori di  $\lambda$  (prossimi allo 0) aumentano la velocità di convergenza ma l'algoritmo può diventare instabile, viceversa per elevati valori di  $\lambda$ . Il valore di  $\lambda$  viene in genere diminuito gradualmente al diminuire della funzione costo; se la funzione costo aumenta, viene tenuto il valore di  $\beta$  precedentemente stimato e si aumenta  $\lambda$  per incrementare la stabilità dell'algoritmo. In questo caso vengono poste a zero le derivate prime e seconde della funzione costo rispetto al parametro  $\beta$  per ottenere la sua stima:

$$\beta^{(n+1)} = \beta^{(n)} - \left( \frac{\partial^2 \mathcal{F}}{\partial \beta^2} \Big|_{\beta^{(n)}} + \lambda I \right)^{-1} \frac{\partial \mathcal{F}}{\partial \beta} \Big|_{\beta^{(n)}} \quad (3.20)$$

La probabilità a priori dei parametri è modellata da una distribuzione gaussiana multivariata con media  $\beta_0$  e covarianza  $C_\beta$ :

$$-\log P(\beta) = \frac{1}{2} (\beta - \beta_0) C_\beta^{-1} (\beta - \beta_0) + \text{const} \quad (3.21)$$

E le derivate prima e seconda sono quindi:

$$\frac{\partial \mathcal{F}}{\partial \beta} = \frac{\partial \mathcal{E}}{\partial \beta} + C_\beta^{-1} (\beta - \beta_0) \quad \frac{\partial^2 \mathcal{F}}{\partial \beta^2} = \frac{\partial^2 \mathcal{E}}{\partial \beta^2} + C_\beta^{-1} \quad (3.22)$$

Il bias del campo magnetico è parametrizzato dall'esponenziale di una combinazione lineare di funzioni di base smooth:

$$\rho_i(\beta) = \exp(\sum_{m=1}^M \beta_m \psi_{im}); \quad \frac{\partial \rho_i(\beta)}{\partial \beta_m} = \psi_{im} \rho_i(\beta); \quad \frac{\partial^2 \rho_i(\beta)}{\partial \beta_m \partial \beta_n} = \psi_{im} \psi_{in} \rho_i(\beta) \quad (3.23)$$

### 3.1.5.3 Stima dei parametri $\alpha$

Identica strategia è usata per aggiornare i parametri riferiti alla registrazione  $\alpha$ ; l'algoritmo di Gauss-Newton tiene conto solo della differenza quadratica media della funzione obiettivo tra le immagini da registrare e di riferimento.

Sono necessarie le derivate prima e seconda della funzione costo rispetto ai parametri che definiscono la deformazione. Per semplificare il calcolo delle derivate la componente *likelihood* della funzione obiettivo è ri-espressa come:

$$\mathcal{E} = \sum_{i=1}^I \log(\sum_{k=1}^K f_{ik} l_{ik}) - \sum_{i=1}^I \log \rho_i(\beta) \quad (3.24)$$

con:

$$f_{ik} = \frac{b_{ik}(\alpha)}{\sum_{j=1}^K \gamma_j b_{ij}(\alpha)} \quad l_{ik} = \gamma_k (2\pi\sigma_k^2)^{-\frac{1}{2}} \exp\left(-\frac{(\rho_i(\beta) y_i - \mu_k)^2}{2\sigma_k^2}\right) \quad (3.25)$$



Le derivate prima e seconda di  $\mathcal{E}$  rispetto ad  $\alpha$  sono:

$$\frac{\partial \mathcal{E}}{\partial \alpha_m} = - \sum_{i=1}^I \frac{\sum_{k=1}^K \frac{\partial f_{ik} l_{ik}}{\partial \alpha_m}}{\sum_{k=1}^K f_{ik} l_{ik}} \quad (3.26)$$

$$\frac{\partial^2 \mathcal{E}}{\partial \alpha_m \partial \alpha_n} = \sum_{i=1}^I \frac{\left( \sum_{k=1}^K \frac{\partial f_{ik} l_{ik}}{\partial \alpha_m} \right) \left( \sum_{k=1}^K \frac{\partial f_{ik} l_{ik}}{\partial \alpha_n} \right)}{\left( \sum_{k=1}^K f_{ik} l_{ik} \right)^2} - \sum_{i=1}^I \frac{\sum_{k=1}^K \frac{\partial^2 f_{ik} l_{ik}}{\partial \alpha_m \partial \alpha_n}}{\sum_{k=1}^K f_{ik} l_{ik}} \quad (3.27)$$

Per la derivata di  $\mathcal{E}$  rispetto ad  $\alpha$  è definita la seguente :

$$\frac{\partial f_{ik}}{\partial \alpha_m} = \frac{\frac{\partial b_{ik}(\alpha)}{\partial \alpha_m}}{\sum_{j=1}^K \gamma_j b_{ij}(\alpha)} - \frac{b_{ik}(\alpha) \sum_{j=1}^K \gamma_j \frac{\partial b_{ij}(\alpha)}{\partial \alpha_m}}{\left( \sum_{j=1}^K \gamma_j b_{ij}(\alpha) \right)^2} \quad (3.28)$$

usata per calcolare le derivate di  $f_{ik}$  basate su una percentuale del campo di deformazione rispetto ai parametri e al gradiente delle mappe di probabilità

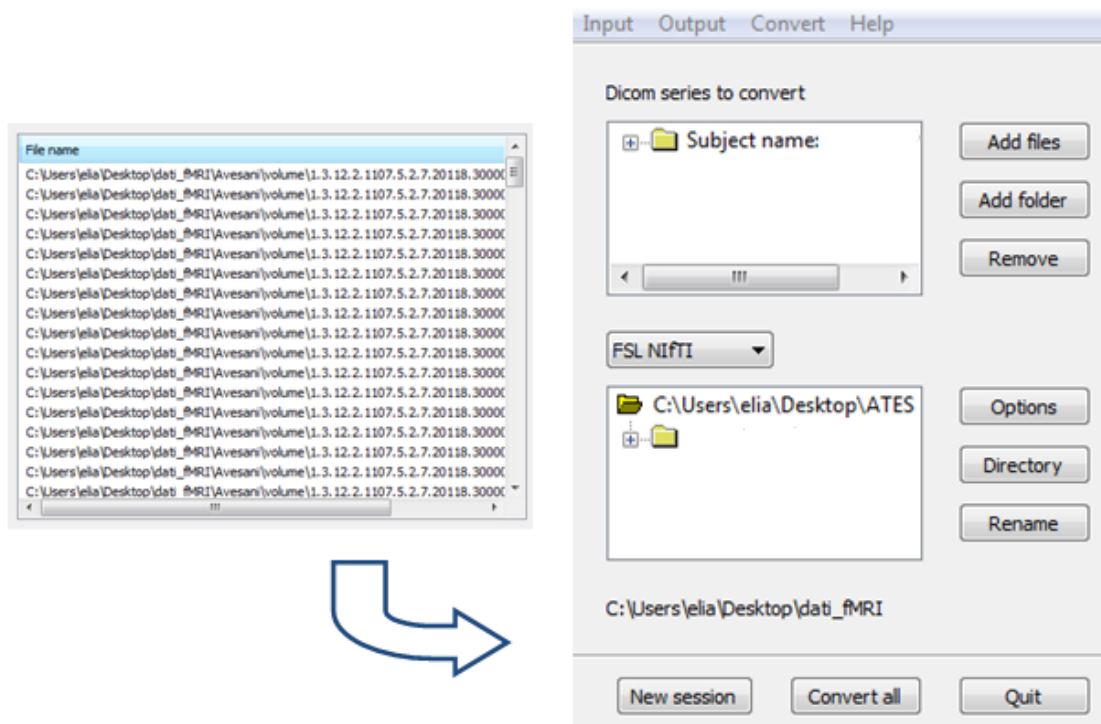
campionate in appropriati punti. Per interpolare i dati si può scegliere tra un'interpolazione trilineare o un'interpolazione con B-spline; dato che le mappe tissutali devono avere valori compresi tra 0 e 1, è preferita la prima perché le B-spline possono produrre valori negativi o fuori dal range ammissibile per mantenere una certa regolarità. Le B-spline possono comunque essere usate attraverso strategie di deconvoluzione che applicano una restrizione sulla non negatività dei nuovi valori stimati.

### 3.2 Implementazione

Sono stati utilizzati 2 dataset ciascuno costituito da 160 immagini MR T1-pesate di due soggetti sani (un maschio e una femmina) di età compresa tra i 25 e 35 anni. Le immagini consistono in slice sagittali ottenute utilizzando una macchina Siemens, secondo le seguenti modalità di scansione:

Parametro	Valore
TR	23 msec.
TE	3.9 msec.
Flip-angle	12°
Slice thickness	1 mm
n° slice	160
matrice d'immagine	256x224

La cartella di ciascun soggetto è costituita da 160 file Dicom, e come operazione preliminare questi sono convertiti in formato Nifti (*Neuroimaging Informatics Technology Initiative*) con la creazione di una coppia di file (.hdr e .img) che contengono le informazioni relative alle immagini.



➤ **Figura 3.5:** finestra grafica per la creazione dei file hdr e img.

È stato scritto il file **main.m** che richiama le funzioni principali che eseguono tutti gli step descritti nel paragrafo 3.1 e le funzioni per la creazione dei modelli 3D:

```
clear all
clc

tpm = load_maps(tpm);
images = obj.image = volume(images);
obj.fudge = 5;
obj.biasreg = 0.0001;
obj.biasfwhm = 60;
obj.tpm = tpm;
obj.lkp = [1,1,2,2,3,3,4,4,4,5,5,5,5,6,6];
obj.reg = 4;
obj.samp = 4;
obj.affreg = 'mni';
Affine = eye(4);
if ~isempty(obj.affreg),
    Affine=
    affine_reg(obj.image,obj.samp,obj.fudge,tpm,Affine,obj.affreg);
end;

res = segment(obj);
save('res','res');

cls = save_results(res);
e3 = cputime-t3

% creo modello del cervello
model_brain;

%creo modello dello scalpo
model_head;
```

### ❖ Frammento di codice 3.1: script main.m

Sono caricate le mappe probabilistiche a priori (*prior*) e il file .img che contiene tutte le slice del soggetto in esame a formare un volume, ovvero una matrice tridimensionale di dimensioni 256x224x160.

Dopo il setting dei parametri, è richiamata la funzione **affine\_reg.m** per fare la registrazione delle immagini con i template in uno spazio standard, che da in output una matrice di trasformazione affine di dimensioni 4x4: prima si fa un campionamento delle immagine e poi si applica iterativamente lo schema di ottimizzazione di Gauss-Newton per minimizzare la funzione obbiettivo.

```

function [M,h] = affine_reg(varargin)
. . .

[x1,x2] = ndgrid(1:sk(1):d(1),1:sk(2):d(2));
x3      = 1:sk(3):d(3);
for i=1:d(3)
    g(:,:,i) = sample_volume(V,x1,x2,o*x3(i),0);
end;
. . .

MG = V.mat;
x   = {x1,x2,x3};
. . .

mu   = [zeros(6,1) ; mu];
isig = [zeros(6,12) ; zeros(6,6) isig];
isig   = isig*ff;
Alpha0 = isig;
. . .

for iter=1:200
penalty = 0.5*(sol1-mu)'*isig*(sol1-mu);
T        = tpm.M\P2M(sol1)*MG;
R        = derivs(tpm.M,sol1,MG);
y1a     = T(1,1)*x1 + T(1,2)*x2 + T(1,4);
y2a     = T(2,1)*x1 + T(2,2)*x2 + T(2,4);
y3a     = T(3,1)*x1 + T(3,2)*x2 + T(3,4);
. . .

if (l11-l1)<1e-4, break; end;
    l1   = l11;
    sol  = sol1;
. . .

Alpha = R'*Alpha*R;
Beta  = R'*Beta;

% Gauss-Newton update
sol1  = sol - (Alpha+Alpha0)\(Beta+isig*(sol-mu));
end;

```

❖ **Frammento di codice 3.2: funzione `affine_reg` per la registrazione in un spazio standard.**

Determinata la matrice di trasformazione  $M$ , è richiamata la funzione ***segment.m*** che combina gli step della segmentazione e della normalizzazione spaziale e della correzione del bias, che da in output una struttura che contiene tutti i parametri ottimi stimati  $(\alpha, \beta, \mu, \sigma, \gamma)$ .

```

% Basis functions for bias correction
chan(n).B3 = dctmtx(d0(3),d3(3),z0);
chan(n).B2 = dctmtx(d0(2),d3(2),y0(1,:));
chan(n).B1 = dctmtx(d0(1),d3(1),x0(:,1));
% Initial parameterisation of bias field
if isfield(obj,'Tbias') && ~isempty(obj.Tbias{n}),
    chan(n).T = obj.Tbias{n};
else
    chan(n).T = zeros(d3);
end
. . .

ll = -Inf;
toll = 1e-5; % Stopping criterion
Starting estimates for intensity distribution parameters
for iter1=1:12,
    mg = obj.mg;
    mn = obj.mn;
    vr = obj.vr;
. . .

% Estimate cluster parameters

for subit=1:20
    mom0 = zeros(K,1)+tiny;
    mom1 = zeros(N,K);
    mom2 = zeros(N,N,K);
    ...
    q = likelihoods(buf(z).f,buf(z).bf,mg,mn,vr);
    ll = ll + sum(log(sq + tiny));
    for k=1:K
        q(:,k) = q(:,k)./sq;
        mom0(k) = mom0(k) + sum(q(:,k));
        mom1(:,k) = mom1(:,k) + (q(:,k) '*cr)';
        mom2(:,:,k) = mom2(:,:,k) + (repmat(q(:,k),1,N) .*cr) '*cr;
    end
end
. . .

% Estimate histogram parameters
for n=1:N,
    mom0 = sum(chan(n).hist(:,k1)) + eps;
    mom1 = sum(chan(n).hist(:,k1).*x) + eps;
. . .

% Estimate bias
Alpha = zeros(d3,d3); % Second derivatives
Beta = zeros(d3,1); % First derivatives
...
Beta = Beta + kron(b3,krutil(wt1,chan(n).B1,chan(n).B2,0));
Alpha = Alpha + kron(b3*b3',krutil(wt2,chan(n).B1,chan(n).B2,1));
...
% Gauss-Newton(iteration to update bias field parameters
R=diag(chan(n).lmreg*(abs(Beta)+sqrt(sum(Beta.^2)/numel(Beta)))); % L-
M regularisation
end
end

```

```

results.image = obj.image;
results.tpm   = tpm.V;
results.Affine = Affine;
results.lkp   = lkp;
results.MT    = MT;
results.Twarp = Twarp;
results.Tbias = {chan(:).T};
if use_mog,
    results.mg   = mg;
    results.mn   = mn;
    results.vr   = vr;
else
    for n=1:N,
        results.intensity(n).lik      = chan(n).lik;
        results.intensity(n).interscal = chan(n).interscal;
    end
end
end

```

❖ **Frammento di codice 3.3: funzione `segment.m` per la segmentazione normalizzazione, e correzione del bias.**

È richiamata infine la funzione **`save_result.m`** che trascrive i dati processati e da in output 6 file `.nii` (6 volumi), tanti quante sono le mappe di distribuzione: avremo a disposizione le immagini della materia grigia, della materia bianca, del fluido cerebrospinale, dei tessuti duri (osso), dei tessuti molli (pelle e muscoli) e di tutto ciò che non appartiene alle immagini precedenti (ovvero l'ambiente circostante).

```

. . .
for k1=1:Kb,
    if tc(k1,4) || any(tc(:,3)) || tc(k1,2) || nargout>=1,
        cls{k1} = zeros(d(1:3), 'uint8');
        do_cls = true;
    end
    if tc(k1,1),
        tiss(k1).Nt      = nifti;
        tiss(k1).Nt.dat = file_array(fullfile(pth, ['c',
num2str(k1), nam, '.nii']), ...
        tiss(k1).Nt.mat = res.image(n).mat;
        tiss(k1).Nt.mat0 = res.image(n).mat;
        tiss(k1).Nt.descrip = ['Tissue class ' num2str(k1)];
        create(tiss(k1).Nt);
        do_cls = true;
    end;
end
end

```

```

. . .

if do_defsl,
    [t1,t2,t3] = defsl(Coef,z,res.MT,prm,x1,x2,x3,M);
. . .

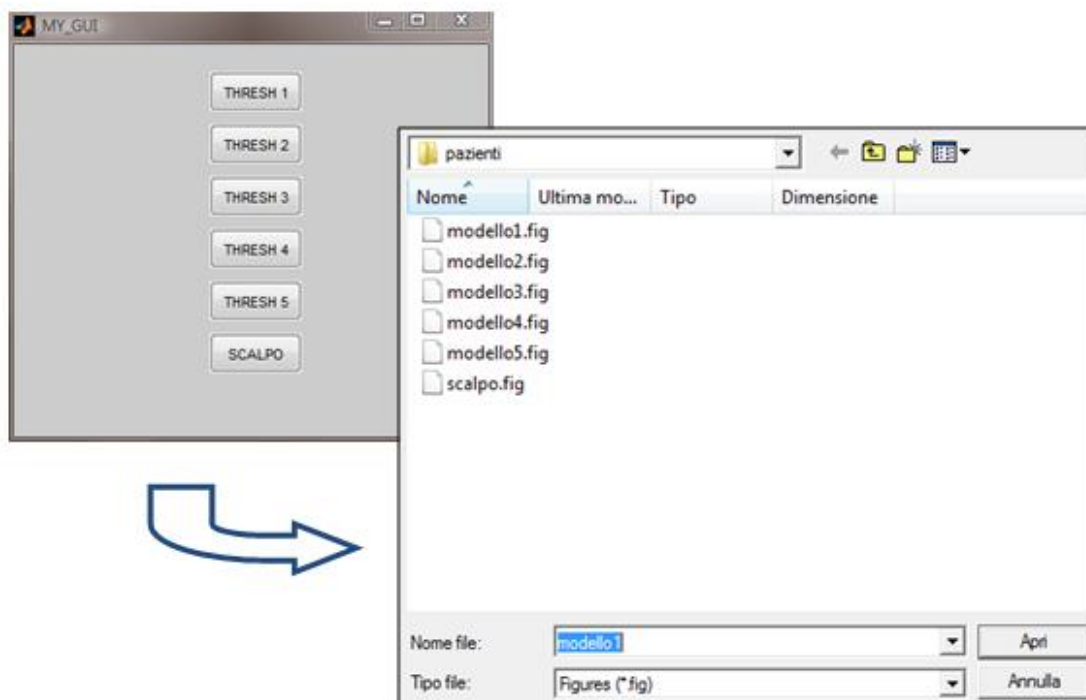
do_cls,
    msk = (f==0) | ~isfinite(f);
    if isfield(res,'mg'),
        q = zeros([d(1:2) Kb]);
        q1 = likelihoods1(cr,[],res.mg,res.mn,res.vr);
        q1 = reshape(q1,[d(1:2),numel(res.mg)]);
        b = sample_maps(tpm,t1,t2,t3);
. . .

for k1=1:Kb,
    tmp = q(:, :, k1);
    tmp(msk) = 0;
    tmp = tmp./sq;
    if ~isempty(cls{k1}),
        cls{k1}(:, :, z) = uint8(round(255 * tmp));
    end
    if ~isempty(tiss(k1).Nt),
        tiss(k1).Nt.dat(:, :, z, ind(1), ind(2)) = tmp;
    end
end
end

```

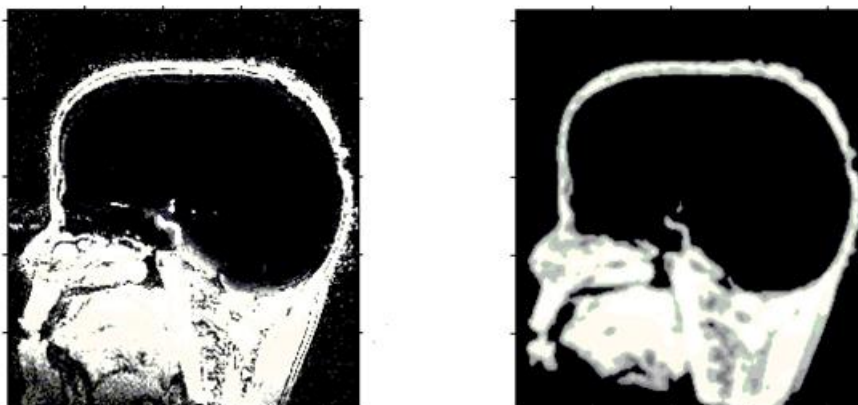
❖ **Frammento di codice 3.4: funzione `save_result.m` che crea i volumi segmentati a partire dai dati processati.**

Arrivati a questo punto si hanno a disposizione le immagini segmentate per creare i modelli 3D di scalpo e cervello e sono richiamate le funzioni ***modello\_head.m*** e ***modello\_brain.m***. Modelli ricostruiti usando la funzione ***isosurface*** di matlab: in particolare per quanta riguarda la creazione del modello del cervello, alla funzione *isosurface* sono passati più *iso-values* (5 diversi valori di soglia) ricostruendo 5 modelli ciascuno determinato da una superficie che racchiude livelli di intensità di tessuto diverse; con un diverso grado di trasparenza, e offrendo ; è poi data all'utente la possibilità di selezionare quello desiderato per mezzo di una semplice interfaccia grafica usata anche per importare e visualizzare il modello dello scalpo.



➤ **Figura 3.6:** interfaccia per la selezione dei modelli creati

Per quanto riguarda invece la ricostruzione del modello dello scalpo si usa un unico *iso-value* e le immagine vengono processate: viene eseguita la funzione ***pulisci.m*** in cui sono iterate operazioni di soglia, smoothing e di controllo dei singoli pixel per eliminare artefatti (quali possono essere piccole protuberanze, oppure vere e proprie macchie) che vanno a compromettere la qualità del modello.



➤ **Figura 3.7:** immagine segmentata dello scalpo (sinistra); immagine risultato dopo la pulizia (destra).



Tale processing non viene eseguito per le immagini segmentate del cervello che risultano meno rumorose e non è necessaria quindi un'operazione preliminare di pulizia; avendo a disposizione più modelli a diversi valori di soglia l'utente può comunque scegliere, se ci sono artefatti, quello in cui sono assenti o presenti in minima misura.

```
function model_brain

thresh = [0.4,0.5,0.6,0.7,0.8];
m = length(thresh);

% calcolo vari modelli quanti sono i valori di soglia
for i = 1:m
    [faces{i},vertices{i}] = isosurface(br,thresh(i));
    p = patch('Vertices', vertices{i}, 'Faces', faces{i} . . .
    set(0, 'CurrentFigure', fg(i));
    set(fg(i), 'CurrentAxes', ax);
    set(fg(i), 'Renderer', 'OpenGL');
    l = camlight('headlight', 'infinite');
    axis image;
end

function model_head

thresh = 0.6;
[n,m,k] = size(br);

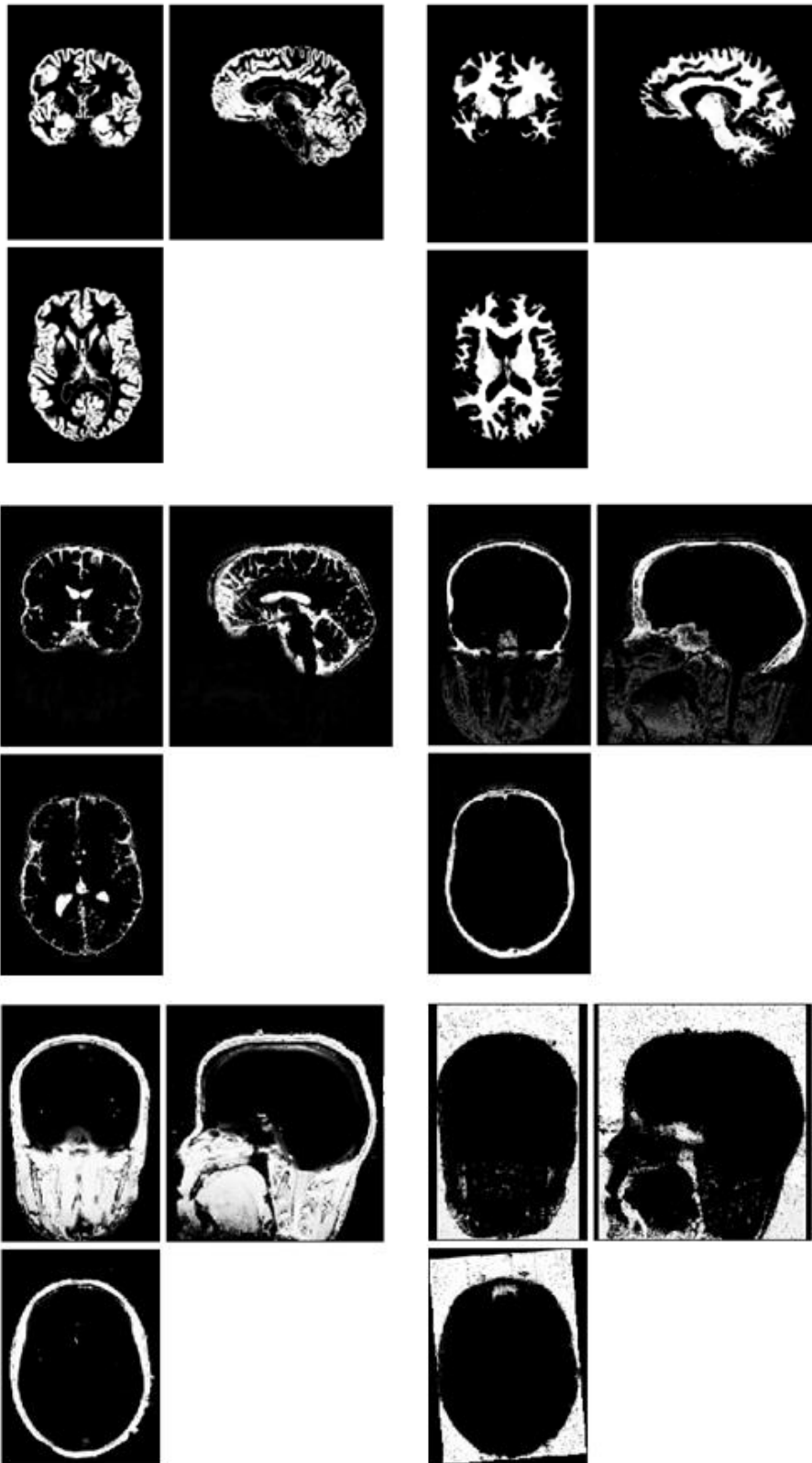
for k = 1:160
    for i = 2:n-1
        for j = 2:m-1
            if (br(i-1,j,k)== 0 && br(i+1,j,k)== 0)
                br(i,j,k) = 0;
            end
            . . .
            A = smooth3(br);
            B = smooth3(A);
            . . .
        end
    end

for i = 1:m
    [faces{i},vertices{i}] = isosurface(br,thresh(i));
    p = patch('Vertices', vertices{i}, 'Faces', faces{i} . . .
    set(0, 'CurrentFigure', fg(i));
    set(fg(i), 'CurrentAxes', ax);
    set(fg(i), 'Renderer', 'OpenGL');
    l = camlight('headlight', 'infinite');
    axis image;
end
```

❖ **Frammento di codice 3.5: ricostruzione dei modelli del cervello e dello scalpo**

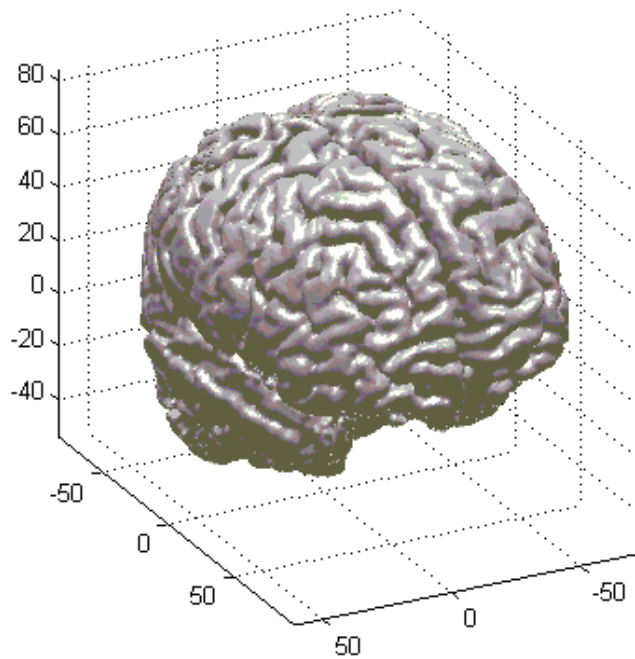
### 3.3 Plot dei risultati

Sono di seguito riportate le immagini risultato della segmentazione:

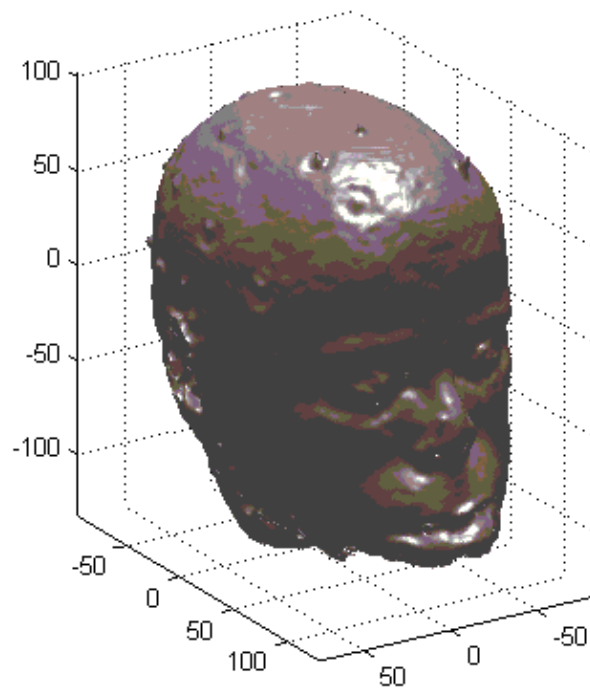


➤ *Figura 3.8: Risultato della segmentazione*

E i modelli 3D di cervello e scalpo ricostruiti:



➤ **Figura 3.9: Modello 3D del cervello**

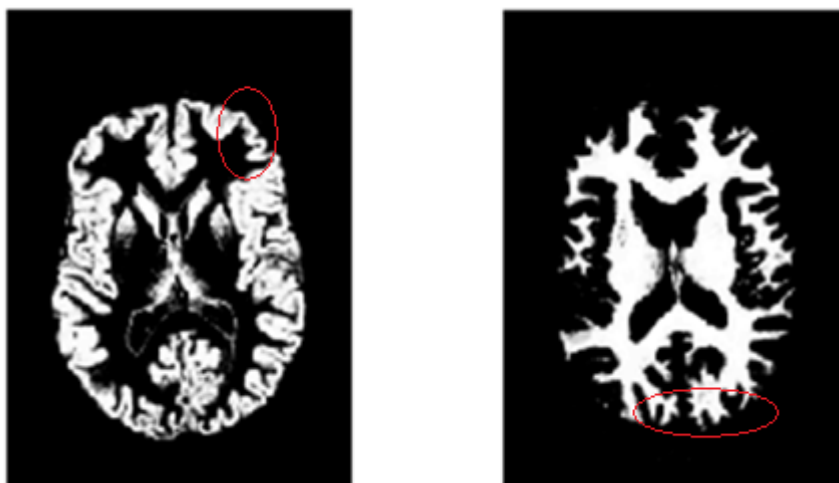


➤ **Figura 3.10: Modello 3D dello scalpo**

### 3.4 Analisi qualitativa dei risultati

Osservando le immagini segmentate (Fig. 3.9) e i modelli ricostruiti (Fig. 3.10) è possibile fare una analisi qualitativa dei risultati ottenuti. Ci si occuperà infatti nel prossimo capitolo di valutare la performance e l'accuratezza della tecnica.

Guardando più in dettaglio le immagini segmentate di materia bianca e materia grigia, per esempio, in (Fig. 3.11) si nota come la segmentazione sia sufficientemente fine e riesce a catturare piccoli dettagli, piccole insenature presenti nella materia grigia e nella materia bianca anche nella realtà. Di primo acchito quindi i risultati sembrano accurati e lo dimostreremo come detto precedentemente nel prossimo capitolo.

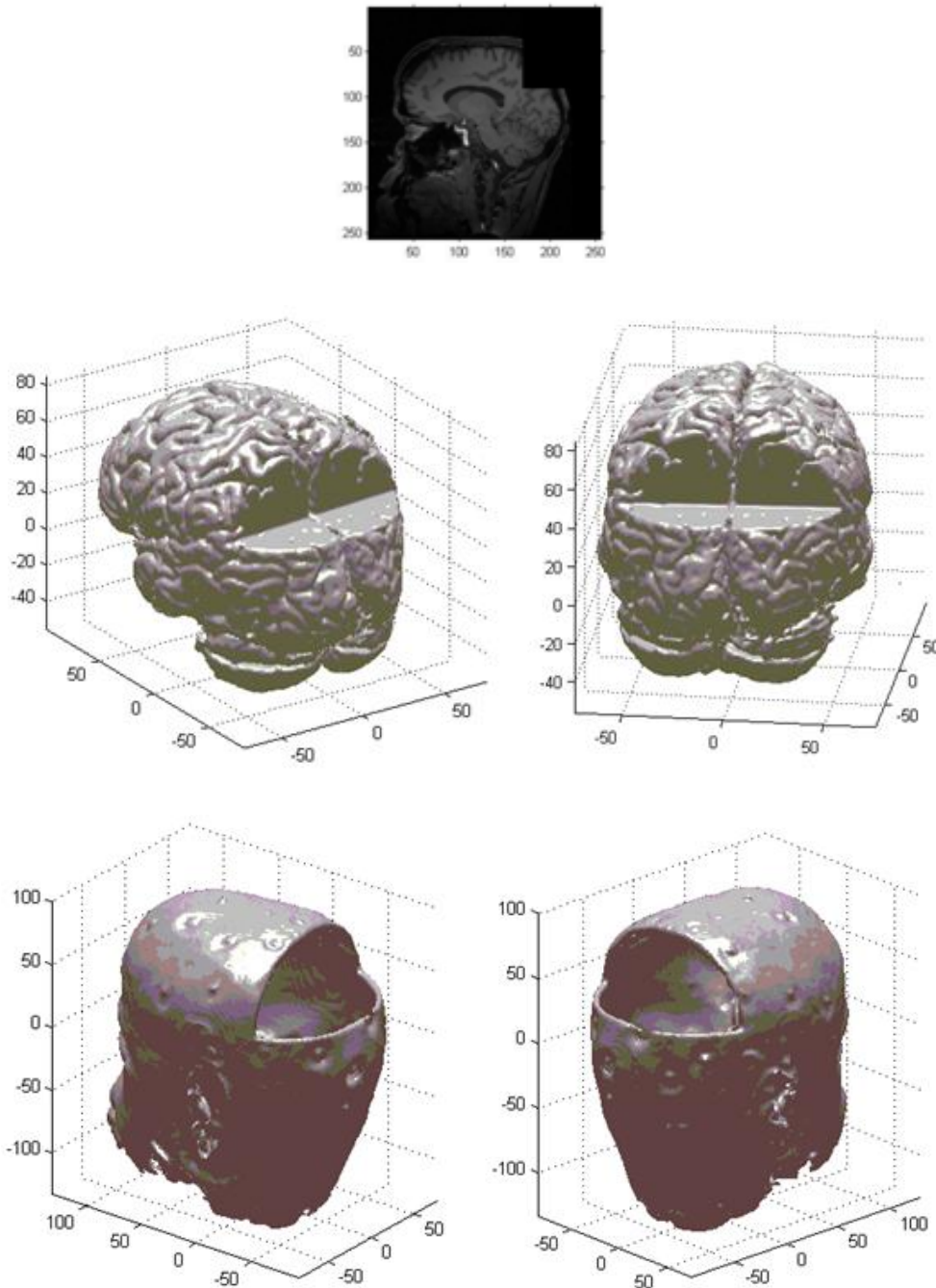


➤ **Figura 3.11: immagini segmentate di materia grigia (sinistra) e materia bianca (destra).**

Anche i modelli ottenuti sono accurati e dettagliati: addirittura nel modello dello scalpo si vedono le posizioni in cui erano fissati gli elettrodi (ove ci sono gli avvallamenti circolari, Fig. (3.10)). Per quanto riguarda il modello del cervello è di gran lunga migliore di quelli ottenuti con le precedenti tecniche ( thresholding e contorni attivi): si vede infatti come sulla superficie cerebrale siano definite dettagliatamente le insenature.

Sono state fatte anche delle prove di segmentazione e poi di ricostruzione dei modelli su immagini MR in cui è stata oscurata una parte dei dati al fine di simulare la mancanza di certe porzioni della struttura cerebrale e dello scalpo che può verificarsi in casi in cui un paziente è sottoposto a particolari interventi

chirurgici: l'obiettivo è valutare la tecnica nei confronti di immagini MR " non standard "; e anche in questo caso come si può vedere nelle seguenti immagini i modelli ricostruiti sono soddisfacenti. In ciascuna delle 160 immagini è stata messa a nero un'area quadrata di dimensioni 90x90 pixel.



➤ **Figura 3.12:** modelli ricostruiti a partire da immagini MR non standard.

I modelli risultati sono corretti, non compare nessun errore di classificazione, soprattutto le zone di confine (in cui si passa dalla massa cerebrale al vuoto) dove ci si aspettava delle complicanze sono invece ben definite ed esatte.

### 3.5 Prestazioni computazionali

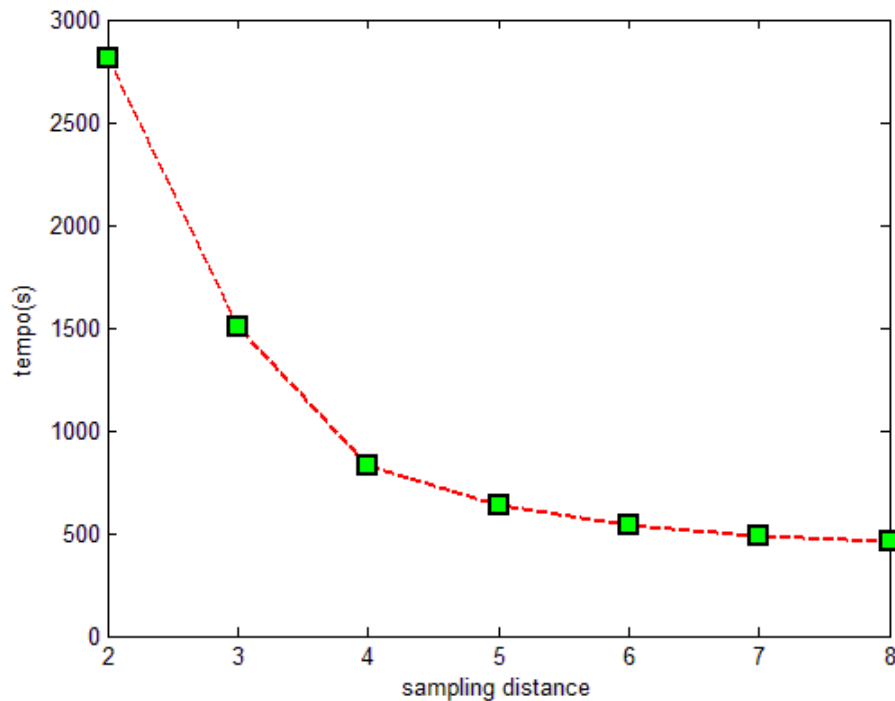
Sono state condotte numerose prove per individuare quali parametri dell'algoritmo incidessero maggiormente sul tempo di esecuzione e scegliere il giusto set di parametri per avere un buon compromesso tra velocità dell'algoritmo e accuratezza dei risultati. Dalle varie prove è risultato che il parametro più significativo è la *distanza di sampling* (in mm) tra i campioni: nei vari step dell'algoritmo non sono ovviamente considerati tutti i voxel dell'immagine perché questo significherebbe avere risultati molto accurati, ma tempi di esecuzione estremamente lenti (circa 2 ore).

Preso il dataset di un soggetto, la seguente tabella riporta i tempi di esecuzione (in secondi) degli step principali dell'algoritmo in funzione di diversi valori della distanza di sampling (in mm), e il tempo totale.

	<b>affine_reg</b>	<b>segment</b>	<b>save_result</b>	<b>model</b>	<b>T. TOT</b>
<b><i>samp = 2</i></b>	<b>1134 s</b>	<b>1376 s</b>	<b>154 s</b>	<b>150 s</b>	<b>2814 s</b>
<b><i>samp = 3</i></b>	<b>505 s</b>	<b>694 s</b>	<b>153 s</b>	<b>151 s</b>	<b>1503 s</b>
<b><i>samp = 4</i></b>	<b>223 s</b>	<b>309 s</b>	<b>152 s</b>	<b>146 s</b>	<b>830 s</b>
<b><i>samp = 5</i></b>	<b>127 s</b>	<b>210 s</b>	<b>149 s</b>	<b>146 s</b>	<b>632 s</b>
<b><i>samp = 6</i></b>	<b>88 s</b>	<b>152 s</b>	<b>150 s</b>	<b>148 s</b>	<b>538 s</b>
<b><i>samp = 7</i></b>	<b>68 s</b>	<b>125 s</b>	<b>148 s</b>	<b>142 s</b>	<b>483 s</b>
<b><i>samp = 8</i></b>	<b>58 s</b>	<b>109 s</b>	<b>152 s</b>	<b>145 s</b>	<b>464 s</b>

Dai risultati ottenuti si osserva che la distanza di sampling incide notevolmente sui tempi di esecuzione della registrazione con i template e della segmentazione: all'aumentare del suo valore ( ovvero considero sempre meno voxels nelle

immagini) si ha un decadimento esponenziale dei tempi di elaborazione che si ripercuote sul tempo di esecuzione totale (Fig. 3.13). Viceversa per la trascrizione dei dati e la ricostruzione dei modelli, come ci si aspettava, l'incidenza è praticamente nulla essendo processi indipendenti dalla distanza di sampling.



➤ **Figura 3.13:** andamento del tempo di esecuzione totale dell'algoritmo in funzione della distanza di sampling.

Tenuto conto quindi dei risultati, un buon compromesso tra velocità di esecuzione ed accuratezza dei modelli ricostruiti è assegnare un valore 4 alla distanza di sampling.

Sono state fatte anche ulteriori prove per cercare di aumentare ulteriormente la velocità di esecuzione della registrazione: anziché considerare tutte le immagini del dataset ne sono state prese 80 (esattamente metà), poi 60, poi 40, poi 20, ed infine 10, tutte in posizione centrale ottenendo i seguenti risultati:

n° immagini	affine_reg
80	223 s
60	222 s
40	223 s
20	124 s
10	100 s

Considerando fino a circa 40 immagini, i tempi di esecuzione sono praticamente identici, come lo è anche la matrice di trasformazione affine che viene data in output; con un numero inferiore di immagini i tempi di esecuzione si riducono anche di molto (con 20 c'è una riduzione di circa 100 secondi) ma la matrice affine è diversa rispetto ai casi precedenti: è evidente che la quantità di dati non è sufficiente per una corretta elaborazione.

Sulla base delle prove condotte, assegnati i valori ottimi dei parametri, le prestazioni dell'algoritmo in termini di velocità di esecuzione sono risultate soddisfacenti: in circa 650 secondi (~ 11 minuti), l'utente ha a disposizione i modelli con un buon livello di accuratezza; naturalmente usando computer di adeguata potenza i tempi si possono ridurre di 3 o 4 minuti. È un tempo accettabile dato che per la calibrazione ed il setting della strumentazione per la neuronavigazione sono necessari circa 10 minuti: il medico quindi in questo lasso di tempo può creare i modelli.







## Capitolo 4

# Validazione del metodo di Segmentazione Unificata

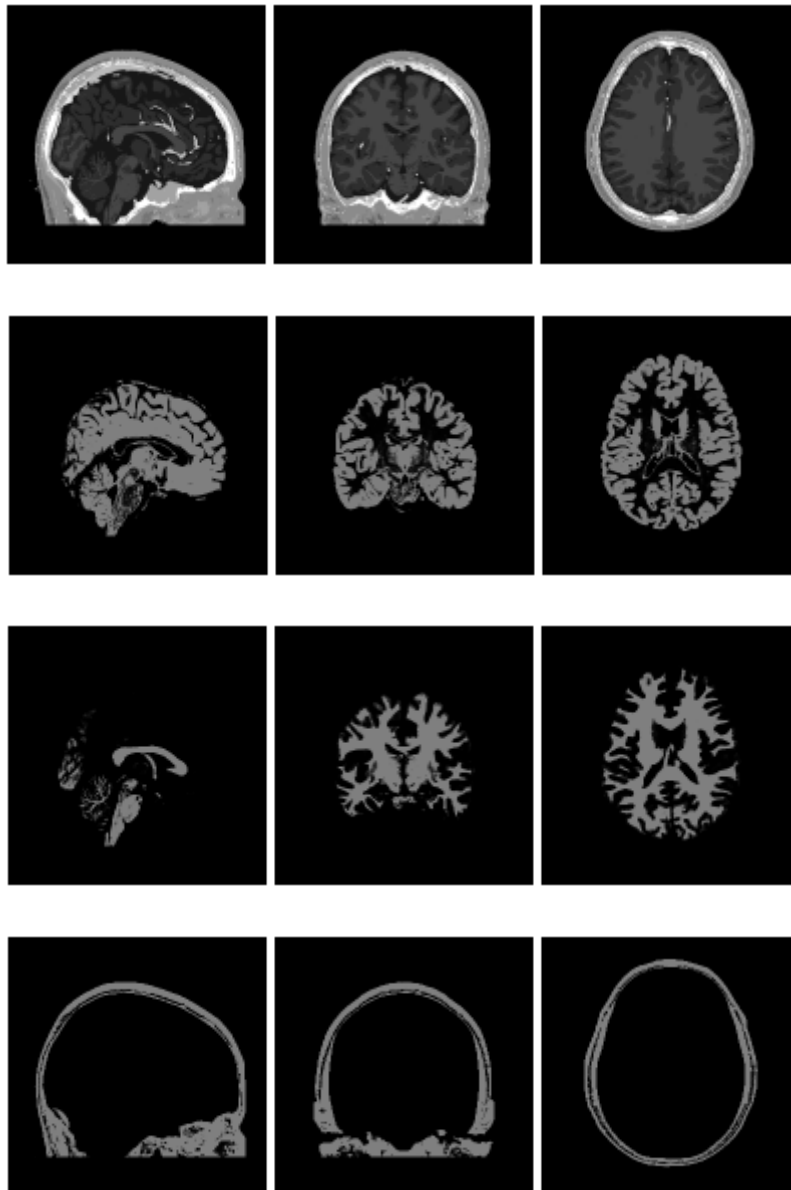
### 4.1 Testing set

Nel capitolo precedente è stata descritta in tutte le sue parti la tecnica della Segmentazione Unificata grazie alla quale si ottengono risultati assai soddisfacenti; le caratteristiche e l'anatomia dei modelli 3D ricostruiti sono di fatto molto vicini alla realtà.

Ci occuperemo ora di validare la tecnica per stabilire il grado di accuratezza della segmentazione.

È stato utilizzato un dataset scaricato dal sito internet BW (BrainWeb <http://www.bic.mni.mcgill.ca/brainweb/>) costituito da un set di dati MRI reali generati da un simulatore: il dataset comprende i dati volumetrici di 15 soggetti e per ciascun soggetto si hanno a disposizione i modelli anatomici di diverse classi di tessuto visualizzati per le tre viste trasversale sagittale e coronale: materia bianca, materia grigia, fluido cerebrospinale, tessuto molle (muscolo e pelle), vasi sanguigni, tessuti duri (osso); i volumi rappresentano le classi di tessuto, e i valori di grigio dei voxels in uno di questi volumi sono proporzionali alla probabilità che il tessuto identificato da quei voxels appartenga alla classe di cui fanno parte i voxels in esame: il nero (valore 0) indica probabilità nulla, il bianco (valore 255) rappresenta la probabilità massima pari a 1. riflettono la proporzione di quanto tipo di un determinato tessuto è presente nel range di valori [0...1].

Le nostre prove sono state fatte solo relativamente alle classi di tessuto materia bianca, materia grigia e scalpo.



**Figura 4.1:** Viste ortogonali dei dati generati dal simulatore MRI. Partendo dall'alto troviamo l'encefalo, la materia grigia, la materia bianca e lo scalpo.

I database simulati (SBD ovvero *Simulated Brain Database*) costituiscono una buona soluzione per valutare l'efficacia delle tecniche di analisi delle immagini non esistendo degli "gold standard" di dati acquisiti in vivo su cui fare affidamento per l'analisi.

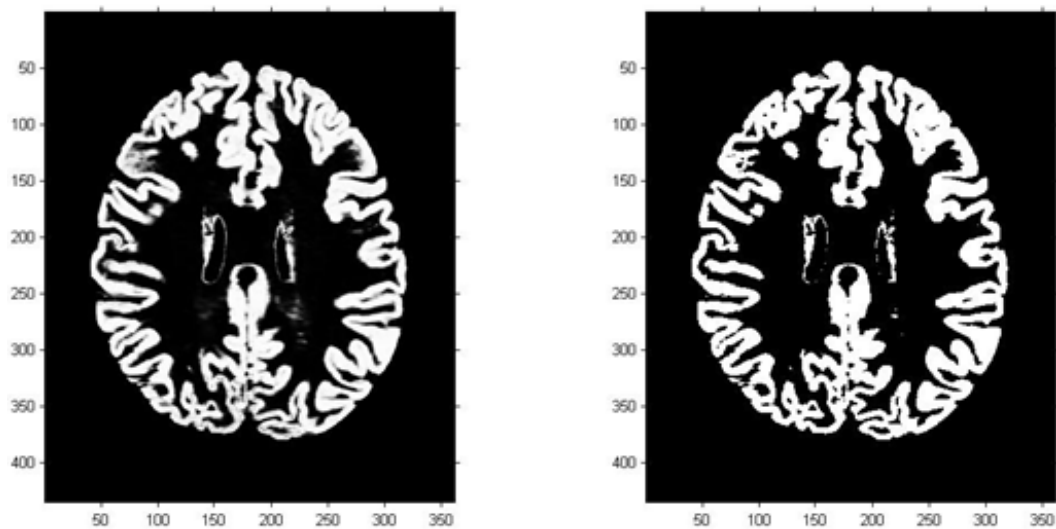
I dati utilizzati per la validazione sono stati generati utilizzando per il simulatore MRI i seguenti valori dei parametri:

<b>Simulation model</b>	normal
<b>MR pulse sequence</b>	
Pulse sequence	T1
Protocol	ICBM
Slice thickness [mm]	1
Scan technique	SFLASH (spoiled FLASH )
Repetition time (TR) [ms]	22
Flip angle [deg]	30
Echo time (TE) [ms]	9.2
<b>Imaging artifacts</b>	
Noise reference tissue	brightest_tissue
Noise level [%]	3
INU ("RF") level [%]	20

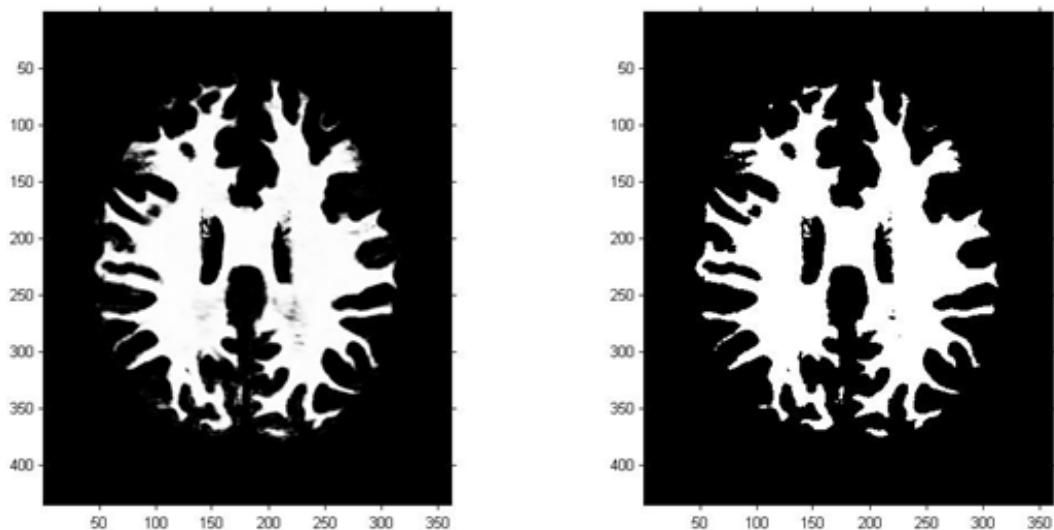
Poiché i dati scaricati sono in formato MINC , ciascun file (volume T1, ovvero il modello discreto completo dell'encefalo, e la segmentazione di riferimento della materia grigia, della materia bianca e dello scalpo) è stato convertito in formato Nifti e poi per le nostre elaborazioni in Matlab in matrici tridimensionali di dimensione 362x434x362.

In primis abbiamo semplicemente confrontato visivamente le immagini della segmentazione di riferimento (quella scaricata) e quelle ottenute con il metodo della segmentazione unificata in modo da osservare immediatamente eventuali grosse differenze riconoscibili anche ad occhio nudo: in particolare in questo primo passaggio si è visivamente valutato se le forme anatomiche fossero mantenute o se comparisse una qualche deformazione; si è osservato inoltre se fossero assenti strutture anatomiche, in tal caso non riconosciute come appartenenti a quella determinata classe di tessuto.

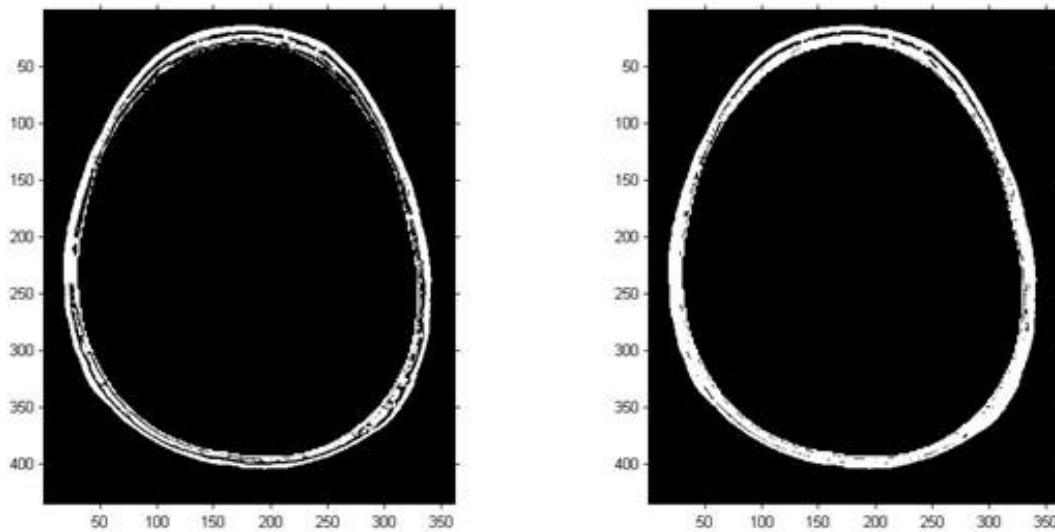
Per queste prime operazioni sono state prese varie immagini campione per più soggetti di cui riportiamo di seguito un esempio:



- **Figura 4.2:** immagini segmentate della materia grigia. Segmentazione di riferimento (sinistra); immagine segmentata ottenuta con la tecnica della segmentazione unificata (destra).

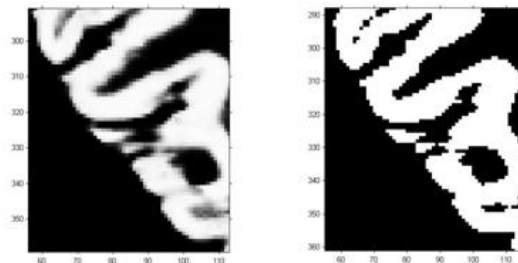


- **Figura 4.3:** immagini segmentate della materia bianca. Segmentazione di riferimento (sinistra); immagine segmentata ottenuta con la tecnica della segmentazione unificata (destra).



➤ **Figura 4.4:** immagini segmentate dello scalpo. Segmentazione di riferimento (sinistra); immagine segmentata ottenuta con la tecnica della segmentazione unificata (destra).

Come nel caso riportato nelle precedenti immagini, dalle numerose coppie di immagini ispezionate non sono state rilevate grosse incongruenze: le immagini ottenute con la Segmentazione Unificata sono molto vicine alla segmentazione di riferimento e in nessun caso nei numerosi confronti effettuati l'algoritmo ha prodotto artefatti rilevanti come deformazioni o mancanza di parti anche minime della struttura. Come si può invece vedere dal dettaglio in figura 4.5, si sono ottenuti risultati accurati anche nella segmentazione di piccole strutture e lobi.



➤ **Figura 4.5:** dettaglio della segmentazione della materia grigia.

## 4.2 Indici di validazione

Dopo questa prima ispezione visiva, attraverso cui è stata valutata grossolanamente l'accuratezza dei risultati, sono stati definiti degli indici di validazione per valutare la bontà dell'algoritmo. Per definire tali indici sono state fatte le seguenti valutazioni, per semplicità ora espresse in termini di slices, ma perfettamente riportabili in termini di volumi:

1. una classificazione perfetta dovrebbe realizzare un'immagine nera, in quanto l'immagine ottenuta dalla classificazione dovrebbe essere identica all'immagine della classe originale; è evidentemente un caso limite, raggiungibile se l'algoritmo fosse specifico per questo soggetto: una qualche generalità dell'algoritmo implica infatti che il riconoscimento abbia una probabilità non unitaria in alcuni punti, in cui gli atlanti non si sovrappongono perfettamente;
2. una immagine differenza che riporti dei bordi sottili e continui (dei contorni di aree anatomiche quindi) è ancora un risultato accettabile, sia perché si tratta in effetti di una differenza spaziale minima (nel nostro caso nell'ordine del millimetro di spessore anatomico), ma anche in quanto il modello che ne deriva presenterebbe un errore paragonabile a quello ottenibile dalla variazione di soglia dell'algoritmo, errore che abbiamo comunque supposto accettabile, preservando come in quel caso le caratteristiche anatomiche corrette;
3. esiste un errore "di presentazione" dei dati, definito come errore dovuto al differente codominio della segmentazione. Mentre nella segmentazione di riferimento i valori di grigio hanno sfumature molto accentuate, con valori di grigio molto bassi soprattutto sui bordi (discontinuità "fuzzy" tra le classi), nella nostra segmentazione i valori di grigio sono 2: bianco o nero (discontinuità "binaria"), e questo perché la tecnica riesce sempre ad assegnare i pixels ad una determinata classe nel caso di segmentazione di modelli simulati che risultano comunque più chiari rispetto ad slices di un esame MRI reale.
4. un'immagine differenza in cui siano presenti solo pixels sparsi sarebbe da considerarsi corretta in quanto sottocaso del caso al punto 2;
5. un'immagine differenza in cui siano visibili delle aree non nere che non siano "sottili" implica che la segmentazione di un'area è errata: nel modello 3D si avrebbe in tal caso una struttura anatomica errata; per le necessità del progetto abbiamo stimato che un errore anatomico di volume circa 2mm cubici



(riconducibile ad un'area di circa 25 pixels quadrati nel caso delle nostre immagini) in tutto il modello possa essere preso come limite di accettabilità.

In base a queste valutazioni si è introdotto un indice  $t$  dell'intensità media dei pixel dell'immagine, il cui valore tra 0 (immagine completamente nera) e 1 (immagine completamente bianca) è proporzionale al numero di pixel non neri ed al loro livello di grigio. Abbiamo poi osservato che tale valore è legato all'anatomia del soggetto esaminato, in quanto a causa dell'errore di rappresentazione il numero di pixel non neri è proporzionale all'estensione anatomica delle aree riportate nell'immagine: questo rende difficilmente correlabili gli indici di diversi soggetti. E' stato quindi necessario applicare una "normalizzazione" rispetto alla massima probabilità di ogni pixel della classe, ottenuta come somma dei livelli di grigio delle immagini confrontate: si è introdotto quindi l'indice relativo  $Tr$ , ottenuto dal rapporto tra l'indice assoluto dell'immagine differenza e l'indice assoluto dell'immagine somma. Tale indice è stato impostato traendo spunto dall'**indice di Jaccard**, un indice statistico utilizzato per confrontare la similarità e la diversità di insiemi campionari ed è definito come la dimensione dell'intersezione divisa per la dimensione dell'unione degli insiemi campionari:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}.$$

Il significato è complementare a  $Tr$  in quanto quest'ultimo è piuttosto un indice di "dissimilarità"; inoltre nel nostro caso abbiamo normalizzato rispetto alla somma e non all'unione, ottenendo un indicatore valido ma con un vantaggio in termini di implementazione.

Da questa prima definizione di  $Tr$  si è poi valutata la capacità dell'indice di evidenziare gli errori di classificazione; con l'obiettivo di massimizzare tale sensibilità abbiamo valutato tre tecniche:

- distanza lineare 'd': l'immagine differenza viene calcolata tramite differenza algebrica pixel a pixel delle immagini originali;
- distanza quadratica 'sd': l'immagine differenza viene calcolata tramite quadrato della differenza pixel a pixel delle immagini originali;

- distanza logaritmica 'ld': l'immagine differenza viene calcolata tramite logaritmo applicato alla differenza algebrica pixel a pixel delle immagini originali;

Dei tre metodi descritti risulta che il metodo più sensibile all'inserimento del quadrettino di errore è il metodo quadratico ('sd'): infatti l'indice risultante ha una sensibilità doppia rispetto alla distanza lineare. Il metodo logaritmico risulta invece avere sensibilità inversamente proporzionale al disturbo.

Inoltre si può fare un'ulteriore valutazione: se la differenza  $U-V$  è negativa (quindi se  $V$  è più "bianca" di  $U$ ) si hanno pixel negativi. Applicando la somma algebrica, i valori di  $t$  potrebbero avere valore nullo anche con immagini differenti e "complementari": infatti per esempio se avessimo in  $U$  metà schermo nero e metà bianco, e se  $V$  fosse l'inverso di  $U$ , le due immagini sarebbero ben diverse, ma  $t$  sarebbe nullo.

### 4.3 Procedura di validazione ed implementazione

Introdotta l'indice  $Tr$ , ora si descrive il procedimento di validazione applicato a ciascuna classe di tessuto:

1. per ogni soggetto viene rilevato l'indice  $t_{\text{differenza}}$  del volume differenza e l'indice  $t_{\text{somma}}$  del volume somma; viene quindi calcolato l'indice  $Tr$  per quel soggetto tramite il rapporto dei due indici precedenti;
2. viene individuato il soggetto che presenta indice  $Tr$  minore, divenendo il soggetto di riferimento;
3. si applica l'errore di classificazione massimo, creando un'area bianca di 25pixel quadrati nell'immagine differenza (quadrato  $5 \times 5$ );
4. si calcola nuovamente  $Tr$ , il cui valore indicherà il  $Tr$  massimo per considerare valida la classificazione:  $Tr_{\text{max}}$ ;
5. si valutano visivamente le slices dei soggetti che hanno  $Tr > Tr_{\text{max}}$  e si osserva se la classificazione sia comunque accettabile;

Tutto ciò è stato implementato in Matlab: è stata scritta la funzione **validazione.m** che restituisce in output un volume differenza  $I1$  con il parametro **t\_differenza**, un volume somma  $I2$  con il parametro **t\_somma**, e il parametro **Tr** ottenuto dividendo il  $t_{\text{differenza}}$  per il  $t_{\text{somma}}$ ; riceve in ingresso due immagini o due

volumi (**U** e **V**) e il parametro "**metodo**", che specifica come sono stati calcolati i **t\_somma**, **t\_differenza**, **I1**, **I2**.

```
function [t,I]= validazione(V,U,metodo)
if(exist('Mask','var')==0), Mask=[]; end
if(exist('MaskNum','var')==0), MaskNum=numel(V); end
switch(type)
case 'd'
    if ( nargin > 1 )
        [t,I]= error_differences(V,U);
    else
        t= error_differences(V,U);
    end
case 'sd'
    if ( nargin > 1 )
        [t,I]= error_squared (V,U);
    else
        t= error_squared (V,U);
    end
case 'ld'
    if ( nargin > 1 )
        [t,I]= error_log_absolute_distance(V,U);
    else
        t= error_log_absolute_distance(V,U);
    end
end
```

❖ **Frammento di codice 4.1: funzione validazione.m per la stima dell'indice t e dell'immagine (o volume) differenza I.**

Si è proceduto quindi con l'analisi dei dati.

In un primo momento si è utilizzata la distanza lineare 'd': con questo metodo **I1** è stato calcolato tramite la differenza (**U-V**) e il parametro **t\_differenza** sommando tutti gli elementi di **I1** diviso la dimensione di **V**; **I2** è stato calcolato tramite la somma (**U+V**) e il parametro **t\_somma** ancora sommando tutti gli elementi di **I2** diviso la dimensione di **V** e **Tr** come **t\_differenza/t\_somma**.

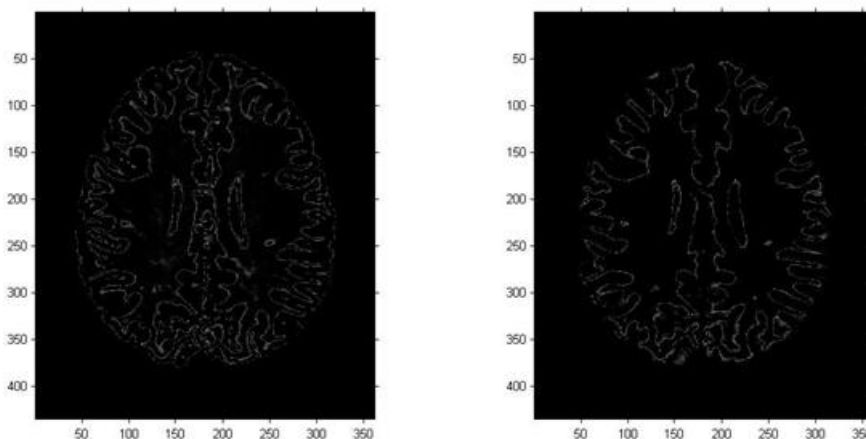
```
function [t,I]=error_differences(V,U,Mask,MaskNum)
I1=(V-U);
I2=(V+U);
if(~isempty(Mask)), I=I.*Mask; end
t_differenza=sum(I(:))/MaskNum;
t_somma=sum(I(:))/MaskNum;
Tr = t_differenza./t_somma;
```

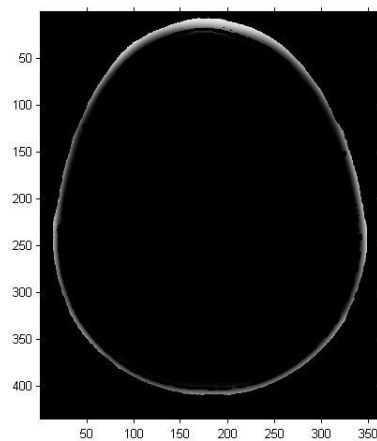
❖ **Frammento di codice 4.2: funzione error\_differences.**

Per ciascuna classe di tessuto (materia bianca, materia grigia e scalpo), sono stati presi volumi parziali di 15 slice: i motivi di questa scelta dipendono esclusivamente dall'incapacità di lavorare con l'intero volume a causa dell'enorme mole di dati e della ridotta velocità computazionale.

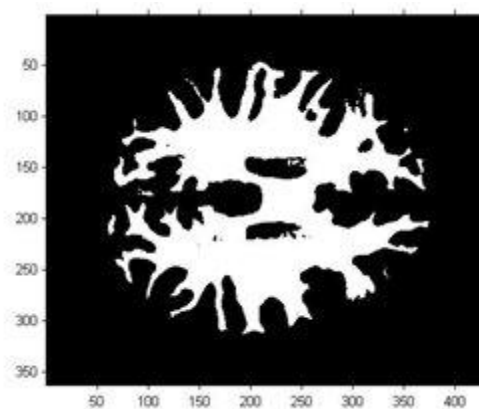
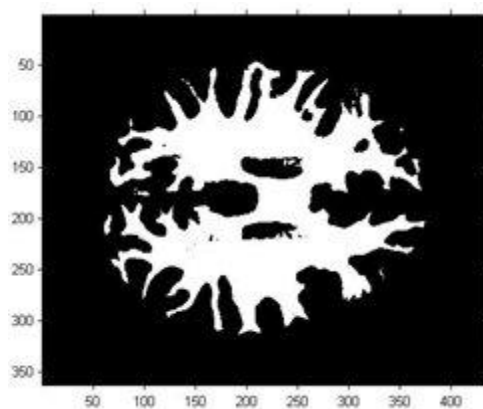
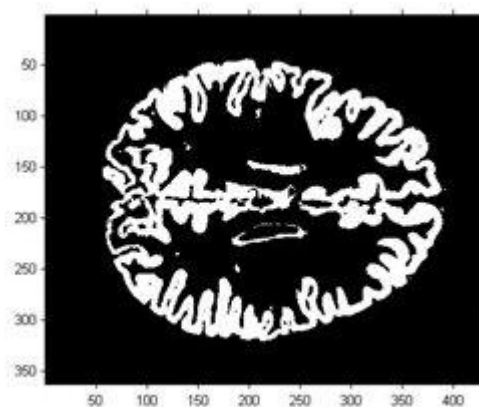
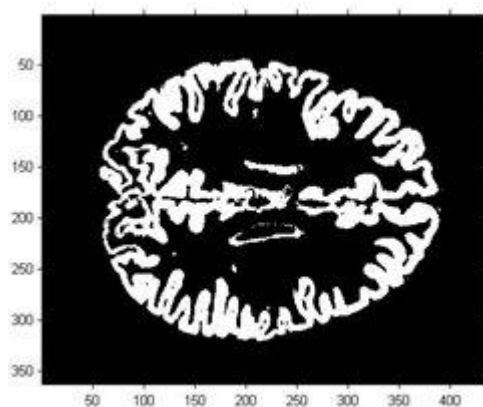
In particolare, se si ispezionano le slice del volume differenza I1 ad esempio per le classi materia bianca e grigia (Figura 4.6), queste presentano contorni continui e "sottili": questa caratteristica è riconducibile all' "errore di presentazione", come si è descritto all'inizio del paragrafo 4.2: le classi di riferimento hanno il bordo sfumato, sono immagini con valori compresi tra zero e uno le classi ottenute con la segmentazione hanno un bordo netto, sono immagini binarie (Figure 4.2 e 4.3).

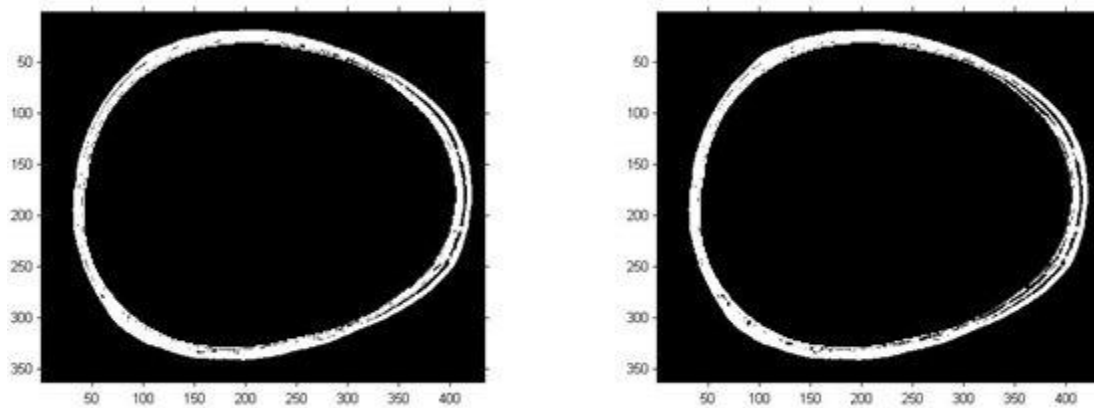
Per ridurre il più possibile l'errore di rappresentazione abbiamo sogliaato le classi di riferimento: i pixel con valori maggiori di 0.5 sono stati messi a 1, quelli con valori minori sono stati messi a zero. Il valore di soglia 0.5 è il valore di soglia che è stato impostato nella nostra segmentazione, valore per il quale otteniamo i modelli 3D migliori.





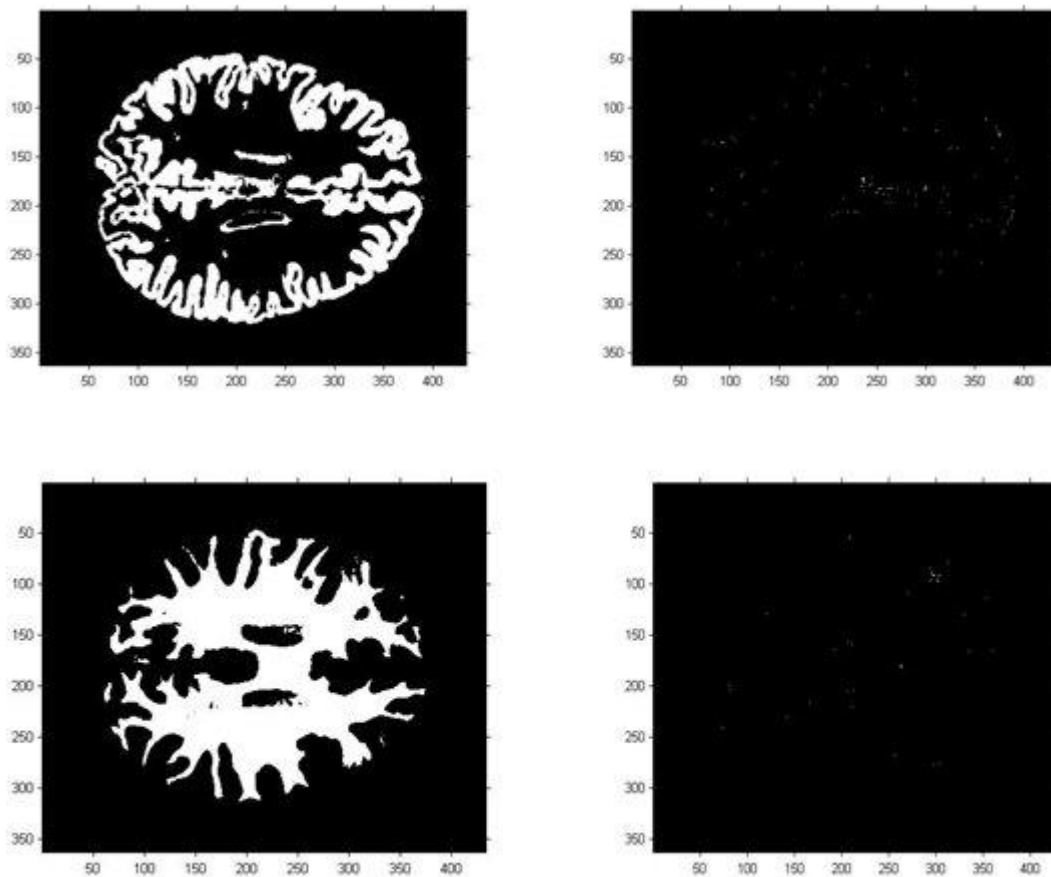
➤ **Figura 4.6:** Immagini differenza per materia grigia, materia bianca e scalpo per un soggetto con il metodo "d".

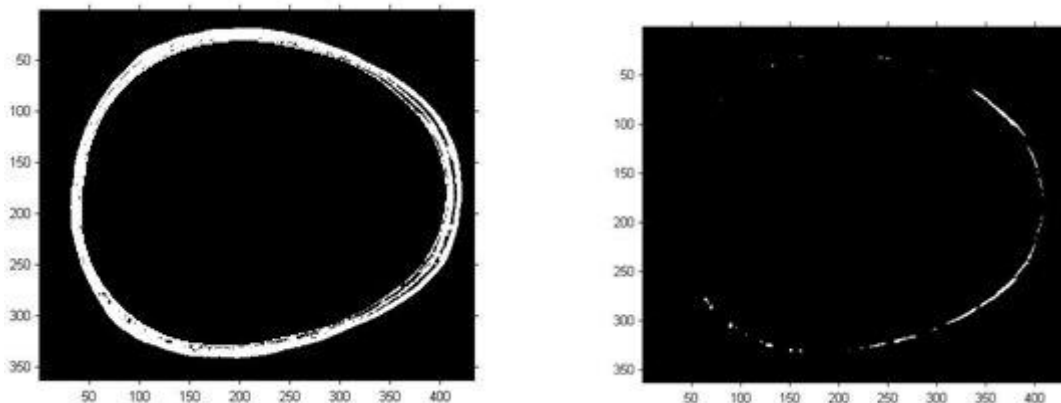




➤ **Figura 4.7:** Immagini web dello scalpo dopo sogliatura (sinistra), Immagine segmentata dello scalpo con la nostra tecnica (destra).

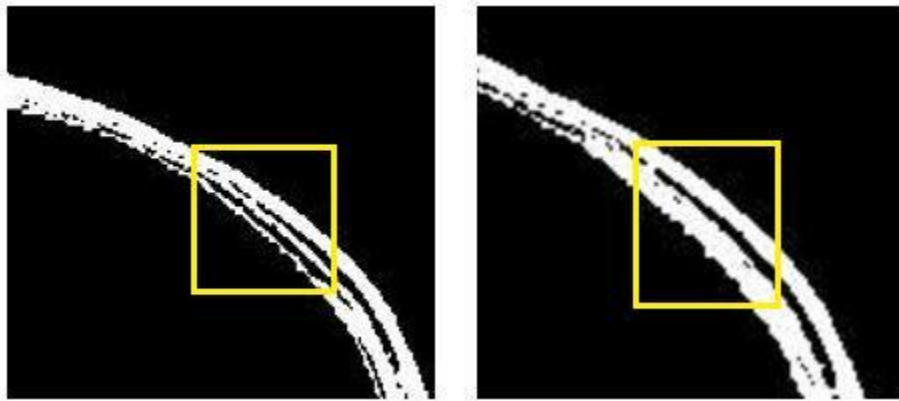
I risultati ottenuti per i volumi somma e differenza sono i seguenti sono i seguenti:





**Figura 4.8:** Immagini somma e differenza per le tre classi

Osservando le immagini differenza si nota come l'errore di rappresentazione sia stato quasi completamente eliminato e le differenze che compaiono sono a questo punto dovute alla sola segmentazione. Per quanto riguarda le immagini differenza di materia grigia e bianca sono quasi completamente nere: la differenza tra la nostra segmentazione e quella di riferimento si riduce a pochi pixel sparsi sull'intera immagine e questo tipo di errore non inficia la qualità del modello 3D. Nelle immagini differenza dello scalpo, non ci sono pochi pixel sparsi ma si riconoscono vaste aree bianche: qui la differenza è più marcata. Abbiamo quindi proceduto con un'analisi più approfondita (Figura 4.9): si vede che la differenza è distribuita sulla parte interna dell'osso: lo spessore dello scalpo nelle due immagini è lo stesso, la differenza sta nel fatto che regioni vuote nell'immagine di riferimento sono invece riempite nelle nostre, effetto dovuto forse alla scarsa sovrapposizione in quest'area degli atlanti dell'algorithmo rispetto ai modelli web, che paiono quindi poco verosimili. Questo tipo di errore può anche dipendere da una parte dalla risoluzione più bassa usata per segmentare, dall'altra dal fatto che i modelli web di riferimento come si è detto sono 'fuzzy'. Abbiamo quindi valutato come accettabile questo errore per l'immagine esaminata visivamente.



- **Figura 4.9:** Particolare dell'immagine web(sinistra) e dell'immagine segmentata con la nostra tecnica (destra).

#### 4.4 Misure ed indici calcolati: metodo 'sd'

Con questo metodo **I1** è stato calcolato facendo la differenza **(U-V)** elevata al quadrato e il parametro **t\_differenza** sommando tutti gli elementi di **I1** diviso la dimensione di **V**; **I2** è stato calcolato invece facendo la somma **(U+V)** elevata al quadrato e il parametro **t\_somma** ancora sommando tutti gli elementi di **I2** diviso la dimensione di **V** e Tr come  $t\_differenza/t\_somma$ .

```
function [t,I]=error_squared_differences(V,U,Mask,MaskNum)
I1=(V-U).^2;
I2=(V+U).^2;
if(~isempty(Mask)), I=I.*Mask; end
t_differenza=sum(I1(:))/MaskNum;
t_somma=sum(I2(:))/MaskNum;
Tr = t_differenza./t_somma;
```

- **Frammento di codice 4.3:** funzione `error_squared_differences`.



I risultati ottenuti per i parametri  $t_{\text{differenza}}$ ,  $t_{\text{somma}}$ ,  $T_r$ , sono i seguenti:

N°Soggetto	td_GM	td_WM	td_SC
1	0.0005415	0.0000412	0.0275
2	0.0004226	0.0001600	0.0078
3	0.0009344	0.0005168	0.0362
4	0.0005228	0.0000373	0.0074
5	0.0005601	0.0001091	0.0066
6	0.0002605	0.0000352	0.0114
7	0.0002168	0.0000068	0.0060
8	0.0001837	0.0000322	0.0083
9	0.0003573	0.0000645	0.0102
10	0.0003641	0.0002109	0.0225
11	0.0008852	0.0003607	0.0106
12	0.0007787	0.0005673	0.0480
13	0.0002597	0.0002699	0.0100
14	0.0002300	0.0000607	0.0108
15	0.0005028	0.0001018	0.0040

- **Tabella 4.1: Valori del parametro t per le tre classi facendo la differenza**

N°Soggetto	ts_GM	ts_WM	ts_SC
1	0.7704	0.8930	0.2284
2	0.7882	0.8550	0.3123
3	0.7915	0.9350	0.4641
4	0.7892	0.8663	0.2238
5	0.7841	0.8539	0.2996
6	0.7833	0.8816	0.2040
7	0.7801	0.8945	0.4107
8	0.8248	0.8645	0.3937
9	0.8018	0.8915	0.4110
10	0.7219	0.8563	0.4737
11	0.7083	0.9610	0.3393
12	0.6946	0.8785	0.3606
13	0.6872	0.9492	0.2433
14	0.7316	0.8528	0.4745
15	0.8001	0.8531	0.1937

- **Tabella 4.2: Valori del parametro t per le tre classi facendo la somma**

N°Soggetto	Tr_GM	Tr_WM	Tr_SC
1	0.0004	0.0000461	0.1204
2	0.0005	0.0001871	0.0250
3	0.0012	0.0005527	0.0780
4	0.0007	0.0000431	0.0331
5	0.0007	0.0001278	0.0220
6	0.0003	0.0000399	0.5588
7	0.0003	0.0000076	0.0146
8	0.0002	0.0000372	0.0111
9	0.0004	0.0000724	0.0248
10	0.0005	0.0002463	0.0475
11	0.0012	0.0003753	0.0312
12	0.0011	0.0006457	0.1331
13	0.0004	0.0002843	0.0411
14	0.0003	0.0000712	0.0222
15	0.0005	0.0001193	0.0206

- **Tabella 4.3: Valori del parametro Tr per le tre classi**

Per quanto riguarda la classe materia grigia (GM), il Tr minore è relativo al soggetto numero 8, per questo soggetto è stato inserito il quadrettino di errore ed è stato calcolato il nuovo valore di Tr, **Tr\_max%** :

$$t(7)_{\text{somma}} = 0.8242;$$

$$t(7)_{\text{differenza}} = 0.0004;$$

da cui ricavo: **Tr\_max% = 0.00049**

Verificando ora la distanza dei Tr da Tr\_max, solo in 5 casi ho valori superiori: soggetto 3, soggetto 4, soggetto 5, soggetto 11, soggetto 12. Per questi tre soggetti l'ispezione visiva delle slice conferma che la segmentazione è comunque buona (non ci sono macchie o parti mancanti) e sono da considerare validi.

Per quanto riguarda la classe materia bianca (WM), il Tr minore è relativo a più soggetti, e prendiamo come riferimento il numero 8; per questo soggetto è stato inserito il quadrettino di errore ed è stato calcolato il nuovo valore di Tr, **Tr\_max%** :

$t(8)_{\text{somma}} = 0.8938;$

$t(8)_{\text{differenza}} = 0.0002359;$

da cui ricavo:  **$Tr_{\text{max}}\% = 0.00026$**

Verificando ora la distanza dei Tr da  $Tr_{\text{max}}$ , solo in 4 casi ho valori superiori: soggetto 3, soggetto 11, soggetto 12, soggetto 13. Per questi tre soggetti l'ispezione visiva delle slices conferma che la segmentazione è comunque buona (non ci sono macchie o parti mancanti) e sono da considerare validi.

Per quanto riguarda la classe scalpo (SC), il Tr minore è relativo al soggetto numero 7, per questo soggetto è stato inserito il quadrettino di errore ed è stato calcolato il nuovo valore di Tr,  **$Tr_{\text{max}}\%$**  :

$t(7)_{\text{somma}} = 0.4109;$

$t(7)_{\text{differenza}} = 0.0062;$

da cui ricavo:  **$Tr_{\text{max}}\% = 0.0150$**

Verificando ora la distanza dei Tr da  $Tr_{\text{max}}$ , solo in 2 casi ho valori inferiori: soggetto 7, soggetto 8.-Nella maggior parte dei casi devo fare l'ispezione visiva: per la presenza degli errori discussi precedentemente infatti i valori di Tr sono molto variabili e distanti tra loro. L'ispezione visiva delle slice conferma però anche per questa classe di tessuto che la segmentazione è da considerarsi buona perché gli "errori" di segmentazione consistono proprio in regioni vuote all'interno dello spessore dello scalpo dell'immagine di riferimento che sono invece riempite nelle nostre.

#### 4.5 Conclusioni

In questo capitolo è stata validata la nostra tecnica di segmentazione; si è cercato di trovare un criterio quantitativo per valutare il grado di accuratezza della tecnica. È necessario fare un'importante considerazione: in quest'ambito l'ispezione visiva delle immagini è molto importante per la valutazione qualitativa della segmentazione; soggetti che i nostri criteri scartano sono invece da considerarsi accettabili dopo un'ispezione visiva.



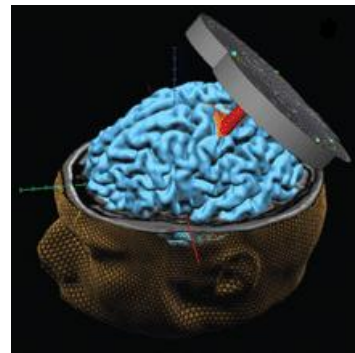
## Capitolo 5

### Conclusioni

La Stimolazione magnetica transcranica (TMS), introdotta negli anni '80 e utilizzata come strumento diagnostico per patologie che alterano le funzionalità di diverse strutture nervose, ha mostrato potenzialità terapeutiche in diversi ambiti e in particolar modo in quello psichiatrico. La TMS è una metodica di stimolazione non invasiva del cervello che, per le sue caratteristiche tecniche, è considerata un importante e promettente strumento d'analisi delle funzioni corticali. Essa utilizza campi magnetici per attivare la corteccia cerebrale e viene per questo utilizzata in clinica neurologica, per diagnosticare patologie che comportano un'alterazione della funzionalità di diverse strutture nervose.

Inoltre si è scoperto che pazienti affetti da patologie neurologiche e disturbi del tono dell'umore, sottoposti a TMS ripetitiva (rTMS) per fini diagnostici, presentavano un miglioramento del quadro depressivo. Da qui, l'avvio dell'utilizzo della Stimolazione magnetica transcranica come trattamento terapeutico in ambito neuropsichiatrico. Se utilizzata in modo ripetitivo, infatti la TMS, ad alte o basse frequenze, può ridurre e modulare i fenomeni di riorganizzazione neuronale, detta plasticità, ed è in grado di facilitare o inibire, in modo relativamente selettivo, circuiti neuronali responsabili di determinate funzioni e determinati sintomi. Recentemente, la TMS sia a singolo stimolo che ripetitiva è stata utilizzata nell'uomo sano anche per studiare le funzioni cognitive. Tale approccio può essere utilizzato non solo per capire se una certa area cerebrale è cruciale per un determinato compito cognitivo, ma anche per tentare di modulare una determinata funzione cerebrale sia in persone sane che con patologia. In base alla funzione cognitiva studiata ed ai parametri di stimolazione utilizzati

(frequenza, durata, intensità di stimolazione e sito), la TMS può sia attivare che disattivare una specifica funzione cognitiva. La stimolazione magnetica transcranica, richiede una apparecchiatura particolare costituita da un generatore di impulsi di corrente e una bobina di stimolazione (coil), composta di solito da cavi di rame arrotolati all'interno di un involucro di plastica, un monitor in cui è visualizzato il modello del cervello ricostruito co-registrato con la testa del paziente. Il soggetto sottoposto ad una seduta con TMS è seduto su una poltrona confortevole, è vigile e, se lo studio lo richiede, deve svolgere un determinato compito durante la stimolazione. Al momento dell'emissione dell'impulso, il coil viene mantenuto in una posizione fissa sulla superficie della testa del soggetto e sul monitor si vede la zona che è stata stimolata.

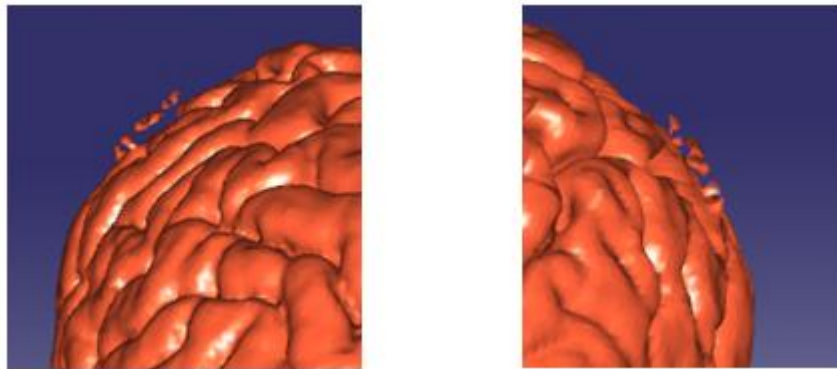


Questo progetto sviluppato in collaborazione con l'azienda **ATES Medica Device** ha avuto come obiettivo quello di ricreare i modelli 3D di cervello e scalpo per un sistema di neuronavigazione in grado di visualizzare in tempo reale la direzione di propagazione del campo magnetico e l'area cerebrale bersaglio, utilizzando informazioni spaziali cerebrali fornite da immagini di risonanza magnetica (MRI). L'obiettivo è stato raggiunto grazie ad un software che comprende la classificazione del tessuto (materia grigia, materia bianca, scalpo), correzione del bias e registrazione all'interno dello stesso modello e con tempi di esecuzione sufficientemente efficienti da garantire l'applicazione in ambulatorio medico.

Il risultato è stato validato con successo tramite confronto tra il risultato prodotto

(volumi relativi alle classi di materia grigia, materia bianca, liquido cerebrospinale, scalpo) ed i reali volumi messi a disposizione da BrainWeb, e osservando che le differenze tra i volumi prodotti dal nostro software ed i volumi reali sono minime ed accettabili.

Restano comunque problematiche da affrontare ed ampi margini di miglioramento. In particolare, in alcuni soggetti, i modelli 3D del cervello ricostruiti presentavano degli artefatti, come si può vedere nella seguenti immagini:



➤ **Figura 5.1: esempi di artefatti apparsi sui modelli ricostruiti.**

questi consistono in piccole porzioni di tessuto che sono staccate dal modello e che dipendono da un'errata classificazione: tessuto che fa parte dello scalpo è classificato facente parte del cervello. Una possibile soluzione adottata, come descritto nel capitolo 3, è stato di calcolare più modelli per diversi valori di soglia e dare la possibilità al medico di scegliere quello in cui sono stati eliminati o ridotti al minimo: si possono fare molte prove su più soggetti (circa un centinaio) e stabilire sulla base dei risultati ottenuti un valore di soglia ottimo (un valore di default), ma offrire comunque al medico una libertà di scelta. Una soluzione alternativa potrebbe essere di ispezionare tutte le immagini ed individuare quelle in cui sono presenti irregolarità che causano gli artefatti dei modelli.

Il software è stato sviluppato in linguaggio Matlab e le sue prestazioni possono essere migliorate ri-implementandolo in un diverso linguaggio come C o C++ guardando anche alle possibilità di parallelizzazione dell'algoritmo: il principio è che i grandi problemi spesso in alcuni casi possono essere suddivisi in quelli più piccoli, che vengono poi risolti simultaneamente ("in parallelo"). Tradizionalmente,

---

il software è scritto per il calcolo seriale; per risolvere un problema, un algoritmo è costruito e implementato come un flusso di una serie di istruzioni che vengono eseguite su un'unità centrale, un computer. Una sola istruzione alla volta può essere eseguita e terminata la precedente è eseguita la successiva. Il calcolo parallelo, al contrario, utilizza elementi di elaborazione simultanea per risolvere un problema. Questo si realizza rompendo il problema in parti indipendenti in modo che ogni elemento di elaborazione sia in grado di eseguire la sua parte dell'algoritmo in contemporanea con gli altri.

Con questi accorgimenti si ottiene di sicuro un algoritmo più performante con un'ulteriore riduzione dei tempi di calcolo arrivando anche a soli 5-6 minuti anziché 10.







# Appendice

## A1

Per la determinazione degli indici sono stati valutati anche gli altri due metodi "d" ed "Id", si riportano i risultati ed i calcoli di seguito.

### Metodo "d"

I risultati ottenuti per i parametri  $t_{\text{differenza}}$ ,  $t_{\text{somma}}$ ,  $T_r$ , sono i seguenti:

N°Soggetto	td_GM	td_WM	td_SC
1	0.00054	0.0000412	0.0099
2	0.00042	0.0001600	0.0022
3	0.00093	0.0005169	0.0038
4	0.00052	0.0000374	0.0033
5	0.00056	0.0001065	0.0025
6	0.00023	0.0000347	0.0074
7	0.00017	0.0000068	0.0013
8	0.00018	0.0000255	0.0043
9	0.00035	0.0000645	0.0035
10	0.00031	0.0002109	0.0014
11	0.00088	0.0003607	0.0096
12	0.00077	0.0005665	0.0091
13	0.00025	0.0002699	0.0054
14	0.00023	0.0000148	0.0005
15	0.00050	0.0001019	0.0035

- **Tabella A1.1: Valori del parametro t per le tre classi facendo la differenza**

<b>N°Soggetto</b>	<b>ts_GM</b>	<b>ts_WM</b>	<b>ts_SC</b>
1	0.3855	0.4465	0.1579
2	0.3943	0.4276	0.1601
3	0.3962	0.4678	0.2502
4	0.3948	0.4332	0.1756
5	0.3923	0.4270	0.1531
6	0.3918	0.4408	0.1588
7	0.3902	0.4473	0.2083
8	0.4125	0.4323	0.2010
9	0.4011	0.4458	0.2106
10	0.3611	0.4283	0.2481
11	0.3546	0.4807	0.1749
12	0.3477	0.4395	0.2043
13	0.3437	0.4748	0.1766
14	0.3659	0.4264	0.2401
15	0.4003	0.4266	0.1479

- **Tabella A1.2: Valori del parametro t per le tre classi facendo la somma**

<b>N°Soggetto</b>	<b>Tr_GM</b>	<b>Tr_WM</b>	<b>Tr_SC</b>
1	0.0014	0.0001	0.0627
2	0.0011	0.0004	0.0137
3	0.0023	0.0011	0.0152
4	0.0013	0.0001	0.0188
5	0.0014	0.0002	0.0163
6	0.0006	0.0001	0.0466
7	0.0004	0.0002	0.0062
8	0.0004	0.0001	0.0214
9	0.0009	0.0001	0.0166
10	0.0009	0.0005	0.0056
11	0.0025	0.0008	0.0549
12	0.0022	0.0013	0.0445
13	0.0007	0.0006	0.0306
14	0.0006	0.0004	0.0021
15	0.0012	0.0002	0.0237

- **Tabella A1.3: Valori del parametro Tr per le tre classi**

I bassi valori dei  $t_{\text{differenza}}$  ottenuti per i volumi differenza di materia bianca e materia grigia (per i volumi dello scalpo sono più alti ma comunque accettabili per le considerazioni fatte precedentemente ) testimoniano che non ci sono grosse differenze tra la nostra segmentazione e quella di riferimento (  $t = 0$ , significa che le due immagini sono completamente uguali,  $t = 1$  che le due immagini sono totalmente diverse): non ci sono grosse macchie bianche (che sarebbero presenti in caso di mancata segmentazione di certe parti anatomiche), ma la differenza è distribuita uniformemente in tutta l'immagine. Per i volumi differenza dello scalpo i valori dei  $t_{\text{differenza}}$  sono invece maggiori e più variabili per le considerazioni fatte precedentemente.

Per quanto riguarda la classe materia grigia (GM), il  $Tr$  minore è relativo al soggetto numero 7, per questo soggetto è stato inserito il quadrettino di errore ed è stato calcolato il nuovo valore di  $Tr$ ,  **$Tr_{\text{max}}\%$**  :

$$t(7)_{\text{somma}} = 0.3901;$$

$$t(7)_{\text{differenza}} = 0.00075;$$

da cui ricavo:  **$Tr_{\text{max}}\% = 0.0019$**

Verificando ora la distanza dei  $Tr$  da  $Tr_{\text{max}}$ , solo in 3 casi ho valori superiori: soggetto 3, soggetto 11, soggetto 12.

Per quanto riguarda la classe materia bianca (WM), il  $Tr$  minore è relativo a più soggetti, e prendiamo come riferimento il numero 8; per questo soggetto è stato inserito il quadrettino di errore ed è stato calcolato il nuovo valore di  $Tr$ ,  **$Tr_{\text{max}}\%$**  :

$$t(8)_{\text{somma}} = 0.4320;$$

$$t(8)_{\text{differenza}} = 0.0002037;$$

da cui ricavo:  **$Tr_{\text{max}}\% = 0.0005$**

Verificando ora la distanza dei  $Tr$  da  $Tr_{\text{max}}$ , solo in 3 casi ho valori superiori: soggetto 3, soggetto 11, soggetto 12, soggetto 13.

Per quanto riguarda la classe scalpo (SC), il Tr minore è relativo al soggetto numero 7, per questo soggetto è stato inserito il quadrettino di errore ed è stato calcolato il nuovo valore di Tr, **Tr\_max%** :

$$t(7)\_somma = 0.2086;$$

$$t(7)\_differenza = 0.0016;$$

$$\text{da cui ricavo: } \mathbf{Tr\_max\% = 0.0076}$$

Verificando ora la distanza dei Tr da Tr\_max, solo in 3 casi ho valori inferiori: soggetto 7, soggetto 10, soggetto 14.

### Metodo 'ld'

I risultati ottenuti per i parametri t\_differenza, t\_somma, Tr, sono i seguenti:

N°Soggetto	td_GM	td_WM	td_SC
1	0.0003753	0.0000285	0.0206
2	0.0002930	0.0001109	0.0054
3	0.0006477	0.0003583	0.0251
4	0.0003624	0.0000259	0.0183
5	0.0003883	0.0000756	0.0046
6	0.0001806	0.0000250	0.0182
7	0.0001503	0.0000047	0.0042
8	0.0001274	0.0000224	0.0057
9	0.0002477	0.0000447	0.0071
10	0.0002524	0.0001462	0.0156
11	0.0006136	0.0002500	0.0074
12	0.0005397	0.0003933	0.0332
13	0.0001800	0.0001871	0.0163
14	0.0001594	0.0000421	0.0047
15	0.0003486	0.0000706	0.0207

- **Tabella A1.4: Valori del parametro t per le tre classi facendo la differenza.**

<b>N°Soggetto</b>	<b>ts_GM</b>	<b>ts_WM</b>	<b>ts_SC</b>
1	0.2118	0.2453	0.0993
2	0.2167	0.2349	0.0890
3	0.2178	0.2570	0.1426
4	0.2170	0.2379	0.1148
5	0.2156	0.2346	0.0850
6	0.2152	0.2422	0.1035
7	0.2144	0.2457	0.1153
8	0.2266	0.2375	0.1116
9	0.2204	0.2449	0.1172
10	0.1984	0.2353	0.1395
11	0.1949	0.2641	0.0976
12	0.1911	0.2415	0.1191
13	0.1888	0.2608	0.1129
14	0.2010	0.2342	0.1327
15	0.2200	0.2344	0.0959

- **Tabella A1.5: Valori del parametro t per le tre classi facendo la somma.**

<b>N°Soggetto</b>	<b>Tr_GM</b>	<b>Tr_WM</b>	<b>Tr_SC</b>
1	0.0018	0.0001	0.2075
2	0.0014	0.0005	0.0607
3	0.0030	0.0014	0.1760
4	0.0017	0.0001	0.1594
5	0.0018	0.0003	0.0541
6	0.0008	0.0001	0.1758
7	0.0007	0.0003	0.0364
8	0.0006	0.0001	0.0511
9	0.0011	0.0002	0.0606
10	0.0013	0.0006	0.1118
11	0.0031	0.0009	0.0758
12	0.0028	0.0016	0.2788
13	0.0010	0.0007	0.1444
14	0.0008	0.0002	0.0371
15	0.0016	0.0003	0.2158

- **Tabella A1.6: Valori del parametro Tr per le tre classi.**

Per quanto riguarda la classe materia grigia (GM), il Tr minore è relativo al soggetto numero 8, per questo soggetto è stato inserito il quadrettino di errore ed è stato calcolato il nuovo valore di Tr, **Tr\_max%** :

$$t(7)\_somma = 0.2265;$$

$$t(7)\_differenza = 0.00053;$$

da cui ricavo: **Tr\_max% = 0.0023**

Verificando ora la distanza dei Tr da Tr\_max, solo in 3 casi ho valori superiori: soggetto 3, soggetto 11, soggetto 12.

Per quanto riguarda la classe materia bianca (WM), il Tr minore è relativo a più soggetti, e prendiamo come riferimento il numero 6; per questo soggetto è stato inserito il quadrettino di errore ed è stato calcolato il nuovo valore di Tr, **Tr\_max%** :

$$t(8)\_somma = 0.2421;$$

$$t(8)\_differenza = 0.0001838;$$

da cui ricavo: **Tr\_max% = 0.00075**

Verificando ora la distanza dei Tr da Tr\_max, solo in 3 casi ho valori superiori: soggetto 3, soggetto 11, soggetto 12.

Per quanto riguarda la classe scalpo (SC), il Tr minore è relativo al soggetto numero 7, per questo soggetto è stato inserito il quadrettino di errore ed è stato calcolato il nuovo valore di Tr, **Tr\_max%** :

$$t(7)\_somma = 0.1155;$$

$$t(7)\_differenza = 0.0043;$$

da cui ricavo: **Tr\_max% = 0.0372**

Verificando ora la distanza dei Tr da Tr\_max, solo in 2 casi ho valori inferiori: soggetto 7, soggetto 8. Nella maggior parte dei casi devo fare l'ispezione visiva:



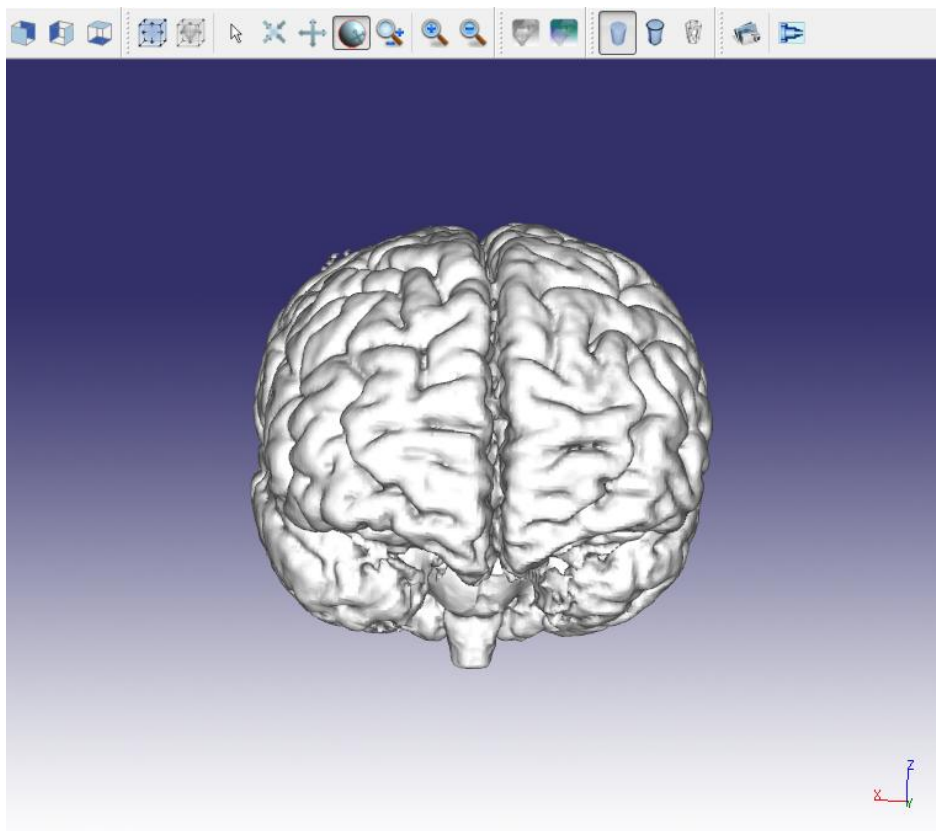
per la presenza degli errori discussi precedentemente infatti i valori di  $T_r$  sono molto variabili e distanti tra loro.

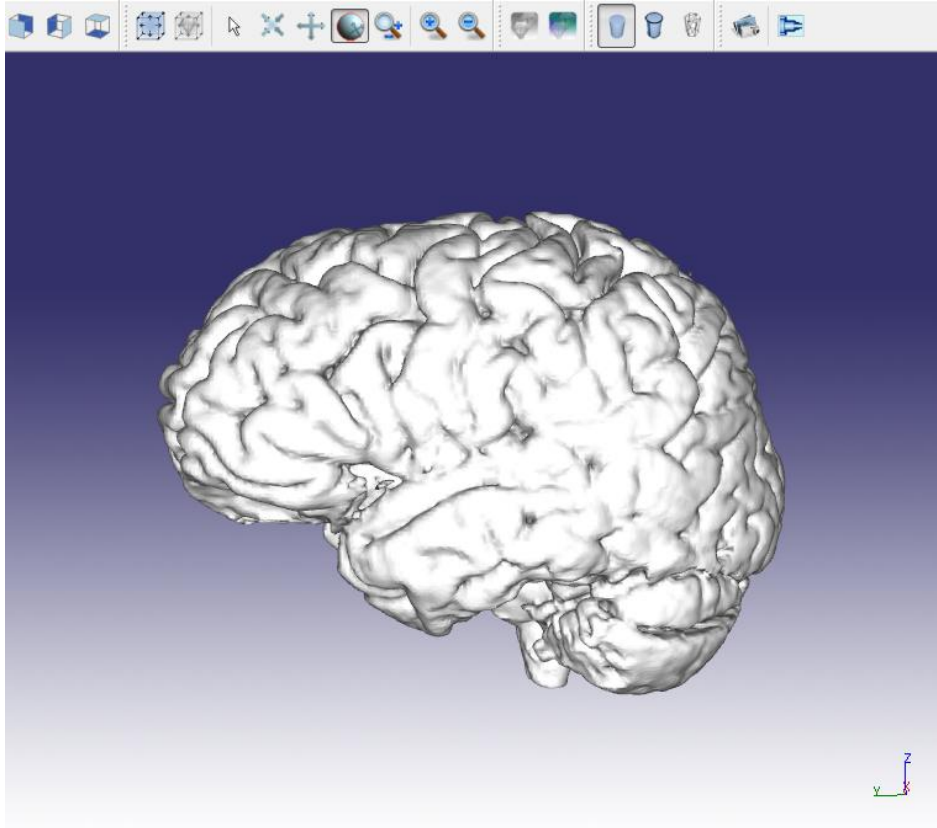
## A2

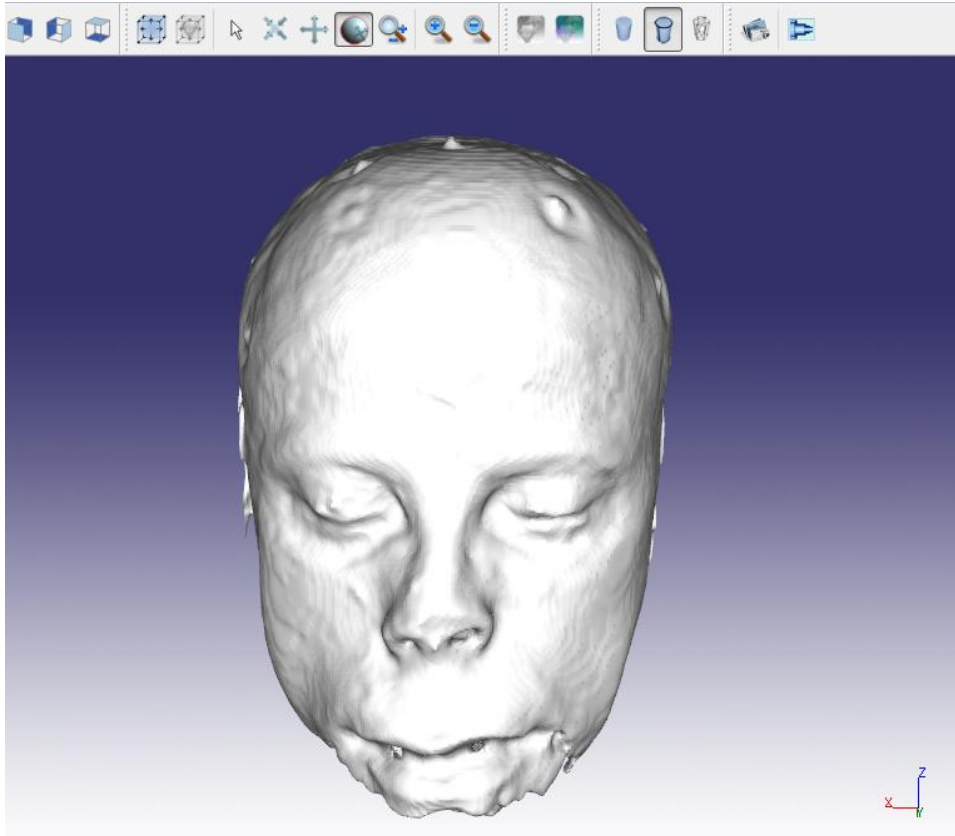
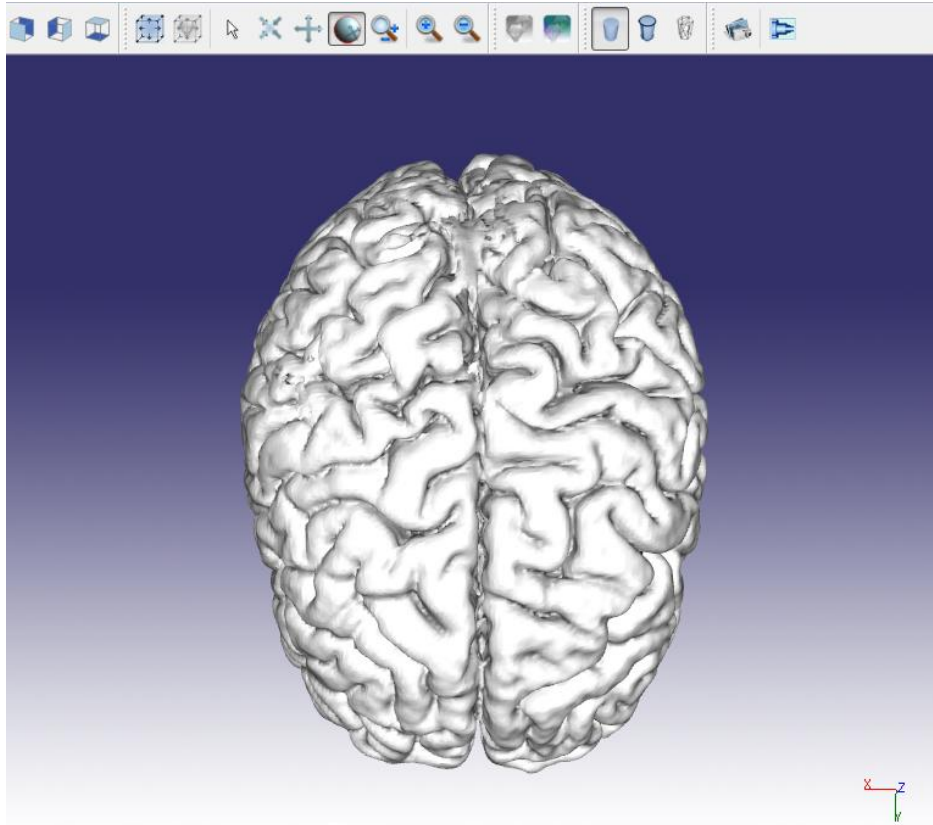
Le prestazioni di matlab nella visualizzazione dei modelli 3D non sono state sufficientemente soddisfacenti; in particolare la rotazione dei modelli per visualizzarli da vari punti di vista sono molto lente.

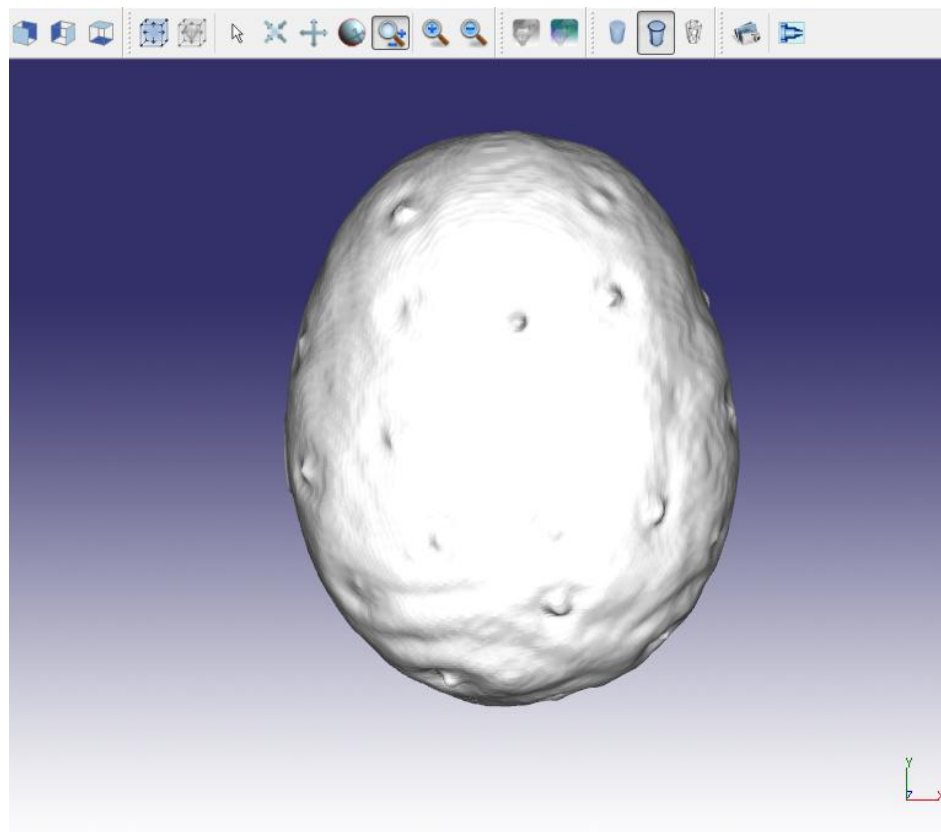
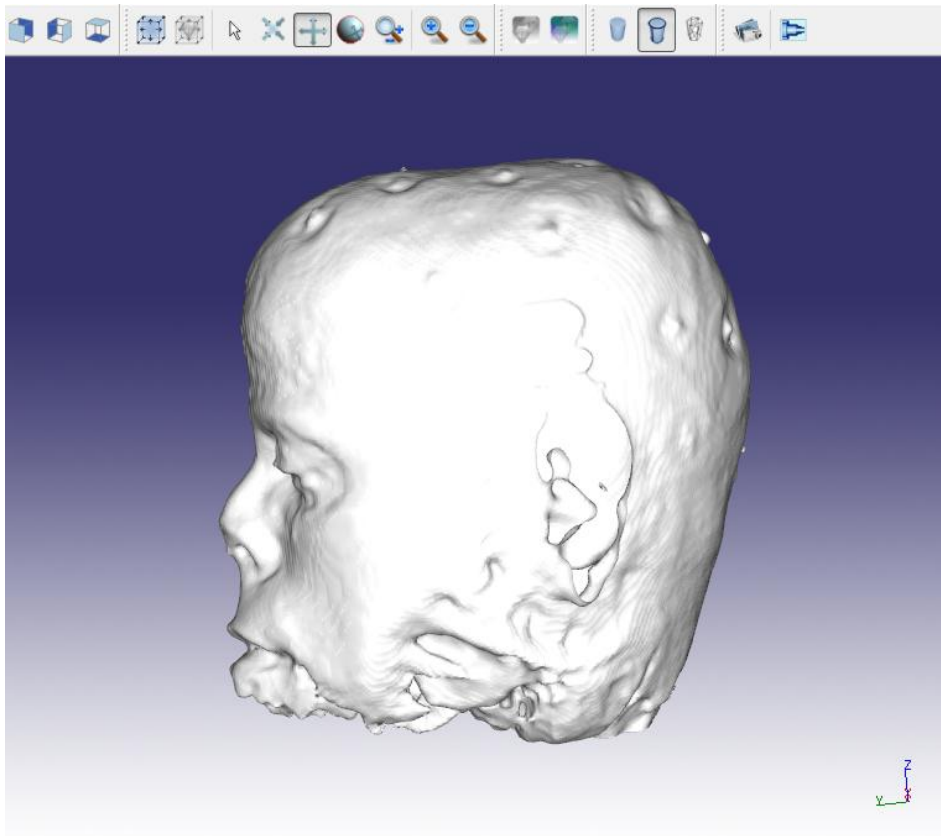
Per questo motivo è stato usato il software **GLC\_Player** che offre delle ottime prestazioni nella visualizzazione di modelli 3D e nella navigazione al loro interno. I dati sono stati convertiti in file **.off** (Object Format file), che rappresentano delle collezioni di poligoni planari con vertici condivisi.

I risultati ottenuti, come si può vedere nelle seguenti immagini, sono ottimi:









---

Le superfici dei modelli ricostruiti sono qualitativamente migliori rispetto alla rappresentazione in Matlab, ma soprattutto si riesce a ruotarli molto velocemente e in modo continuo.



## Bibliografia

R.Sparing, M.D. Hesse, G.R.Fink, *Neuronavigation for transcranial magnetic stimulation (TMS): Where we are and where we are going*. Cortex 46, 118-120, 2010.

U.Hervig, C-Lecuona, A.P. Wunderlich, A.Thielscher, H.Walter, M.Spitzer. *The Navigation of transcranial magnetic stimulation*. Psychiatry Reserch: Neuroimaging Section 108 (2001) 123-131.

Ourselin, S., Roche, A., Subsol, G.,Pennec, X., Ayache, N.,2005. *Reconstructing a 3D structure from serial histological sections*. NeuroImage 6 (3), 209– 217.

Y. Zhang, M. Brady, and S. Smith, *Segmentation of brain MR images through a hidden markov random field model and the Expectation Maximization algorithm*, *IEEE Trans. Med. Imag.*, vol. 20, pp. 45–57, Jan. 2001.

K.V: Leemput, Maes, D. Vandermeulen, and P.Suetens, *A Unifying Framework for Partial Volume Segmentation of Brain MR Images*, *IEEE Trans. Med. Imag.*, vol. 22, Jan. 2003.

Ashburner, J., Friston, K.J., Neelin, P.,Collins, D.L., Evans, A.,2005. *Incorporating Prior Knowledge into Image Registratio*. NeuroImage 6 (3), 209– 217.

Ashburner, J., Friston, K.J., 1997. *Multimodal image coregistration and partitioning—a unified framework*. NeuroImage 6 (3), 209– 217.

Ashburner, J., Friston, K.J., 2005. *Unified Segmentation*. NeuroImage 6 (3), 209– 217.

Ashburner, J., Andersson, J., Friston, K.J., 2000. *Image registration using a symmetric prior—in three-dimensions*. Hum. Brain Mapp. 9 (4), 212– 225.

Ashburner, J., Andersson, J., Friston, K.J., 1999. *High-dimensional nonlinear image registration using symmetric priors*. NeuroImage 9, 619– 628.

H. Greenspan, G. Oza, N. Kiryatia, S. Peled, *MRI inter-slice reconstruction using super-resolution*. Magnetic Resonance Imaging 20 (2002) 437–446.

D’Agostino, E., Maes, F., Vandermeulen, D., Suetens, P., 2004. *Non-rigid atlas-to-image registration by minimization of class-conditional image entropy*. In: Barillot, C., Haynor, D., Hellier, P. (Eds.), Proc. MICCAI 2004. LNCS 3216. Springer-Verlag, Berlin Heidelberg.

Xiaohui Yuan, Xiaojing Yuan, *Fusion of multi-planar images for improved three-dimensional object reconstruction*. Computerized Medical Imaging and Graphics (2010).

Friston, K.J., Ashburner, J., Frith, C.D., Poline, J.-B., Heather, J.D., Frackowiak, R.S.J., 1995. *Spatial registration and normalization of images*. Hum. Brain Mapp. 2, 165– 189.











