

# Watershed segmentation for White Matter extraction

Riccardo Arena - VR096092  
Gianluca Rubinacci - VR093283  
Giuseppe Ballardin - VR096292

# 1 Segmentazione mediante la trasformata Watershed

In geografia topologica *watershed* rappresenta il dislivello con cui un'area viene drenata da diversi sistemi fluviali. Un bacino idrografico è un'area geografica drenata in bacino o riserva idrica.

La *watershed transform* applica l'idea nell'elaborazione di immagini in scala di grigi in modo che possa essere utilizzata per risolvere vari problemi di segmentazione dell'immagine.

Comprensibilmente la *watershed transform* richiede di elaborare immagini in scala di grigio come superfici topologiche, dove il valore della funzione in un punto  $f(x,y)$  è interpretato come altezza. Possiamo, ad esempio, considerare l'immagine in figura 1 come una superficie tridimensionale.

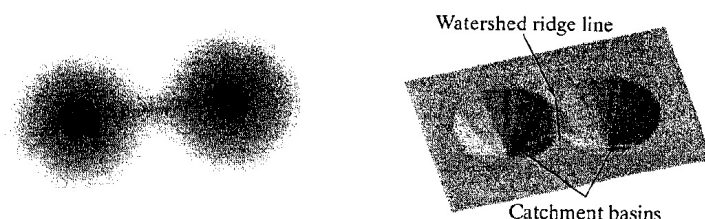


Figura 1: immagini in scala di grigi viste come superfici limitate dalle linee bacino di watershed e relativi bacini idrografici.

Se immaginiamo veder cadere della pioggia su tale superficie, è chiaro che l'acqua verrebbe raccolta in due aree distinte chiamate bacini idrografici. L'acqua caduta esattamente sulla linea definita dalla funzione di watershed si distribuirebbe equamente in entrambi i bacini. In termini di risoluzione dei problemi di segmentazione, il concetto chiave? quello di cambiare l'immagine di partenza in un'altra immagine di cui i bacini idrografici sono gli oggetti o le regioni che vogliamo identificare.

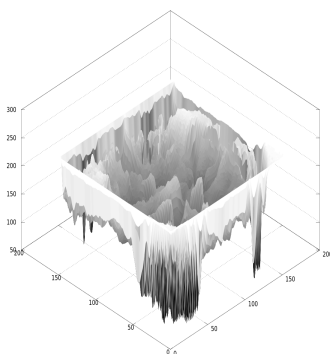


Figura 2: Visualizzazione 3D del negativo dell'immagine

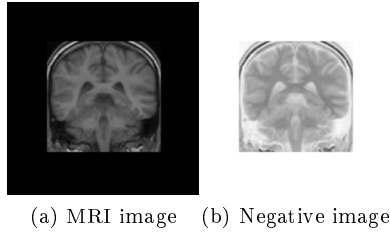


Figura 3

## 2 Algoritmo di Watershed

In Image Processing, *watershed transform* fa parte della più ampia branca di *algoritmi di segmentazione dell'immagine*, area che si occupa della semplificazione e/o della modifica di rappresentazione di un'immagine in qualcosa di più significativo e facile da analizzare.

Il risultato della segmentazione di un'immagine è un set di segmenti che ricoprono collettivamente l'intera immagine, o un set di contorni estratti dall'immagine. Ogni pixel in una regione è simile (secondo alcune caratteristiche come colore, intensità, ecc) ad uno appartenente alla stessa regione. Questa tecnica è basata sull'assunzione che i contorni dell'immagine corrispondano alle linee di margine del suo gradiente.

Data l'immagine  $I$  in scala di grigi, l'algoritmo è il seguente:

- viene calcolata l'immagine gradiente  $I'$ ;
- per ogni oggetto d'interesse viene selezionato il pixel interno con il valore più basso possibile. Questi costituiscono la base per bacini idrografici iniziali;
- a partire da questi punti l'immagine gradiente viene *inondata*, ovvero: per ogni livello di intensità  $k$ :
  - per ciascun gruppo di pixel di intensità  $k$ :
    - \* se adiacenti ad esattamente una regione esistente, aggiungere questi pixel di quella regione;
    - \* altrimenti se adiacenti a più regioni esistenti, viene impostato come confine;
    - \* si passa alla regione successiva.

La forza dell'algoritmo è quella di fornire risultati univoci per ogni immagine sorgente ed essere esteso facilmente a immagini  $n$ -dimensionali.

Lo svantaggio è che in presenza di rumore nell'immagine sorgente, il risultato finale presenterà sovra segmentazione (Figura 4), problema risolvibile ricorrendo allo smoothing preliminare dell'immagine. Altre possibilità per evitare la sovra-segmentazione sono l'impostazione della *minima profondità del bacino* o della *minima dimensione del bacino*.

L'estrazione della Materia Bianca è effettuata secondo l'assunzione di connettività della stessa. Due punti di un'immagine in scala di grigi sono connessi se esiste un percorso di pixel adiacenti a loro luminosi tanto quanto il più scuro tra i due.

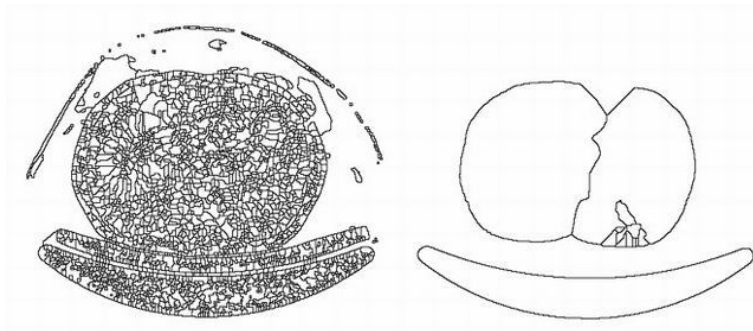


Figura 4: Fenomeno della sovra segmentazione

### 3 Segmentazione della materia bianca

L'algoritmo Watershed é usato per segmentare un immagine MRI del cervello .  
Il codice utilizzato é il seguente:



Figura 5: Immagine originale

```
clc; clear; close all;

p = imread('data/brain2.jpg');

% Converte l'immagine in scala di grigi
I = rgb2gray(p);
subplot(221); imshow(I,[]), title('Original image')

% crea un filtro bidimensionale hy esaltando con l'opzione
```

```

% 'sobel' i bordi orizzontali dell'immagine. (estrazione della ↵
% variazione dell'intensità)
hy = fspecial('sobel');

% hx è la matrice trasposta hy: esalta i bordi verticali
% dell'immagine
hx = hy';

% esaltiamo i bordi orizzontali e verticali in due matrici distinte Iy↵
% e Ix
% filtra l'array multidimensionale double(I) con il filtro hy
Iy = imfilter(double(I), hy, 'replicate');
% filtra l'array multidimensionale double(I) con il filtro hx
Ix = imfilter(double(I), hx, 'replicate');

% estrae l'immagine gradiente dalle immagini filtrate. Ogni punto è la
% magnitudine della variazione di intensità
gradmag = sqrt(Ix.^2 + Iy.^2);
subplot(222); imshow(gradmag,[]), title('Gradient magnitude (gradmag)')↵
)

% crea l'elemento morfologico strutturante di forma discoidale
% di raggio 10
se = strel('disk', 10);

% esegue l'apertura morfologica dell'immagine tramite l'elemento
% morfologico strutturante.
% 'imopen' comprende l'operazione di erosione e dilatazione.
Io = imopen(I, se);

% esegue l'erosione morfologica dell'immagine tramite l'elemento
% morfologico strutturante.
Ie = imerode(I, se);

% effettua la ricostruzione morfologica dell'immagine Ie, attraverso ↵
% la maschera I (immagine in scala di grigi)
Iobr = imreconstruct(Ie, I);
subplot(223); imshow(Iobr), title('Apertura attraverso ricostruzione (↵
Iobr)')

% dilata l'immagine Iobr usando l'elemento morfologico 'se' (forma ↵
% discoidale)
Iobrd = imdilate(Iobr, se);

% ricostruisce l'immagine tramite i rispettivi complementi delle ↵
% elaborazioni
Iobrcbr = imreconstruct(imcomplement(Iobrd), imcomplement(Iobr));

% esegue il nuovamente il complemento per restituire l'immagine con i
% massimi regionali corretti
Iobrcbr = imcomplement(Iobrcbr);
subplot(224); imshow(Iobrcbr), title('Apertura-chiusura attraverso ↵
% newline la ricostruzione (Iobrcbr)')

% estrae i massimi regionali
fgm = imregionalmax(Iobrcbr, 8);
figure, subplot(221); imshow(fgm), title('Massimi regionali di ↵
% apertura-chiusura \newline attraverso ricostruzione (fgm)')

I2 = I;
% i massimi vengono sovrapposti all'immagine I2
I2(fgm) = 255;

% creazione dell'elemento strutturante se2
se2 = strel(ones(2,2));

% dilatazione dei massimi per evitare frammentazione dei marker
fgm2 = imclose(fgm, se2);
fgm3 = imerode(fgm2, se2);
% rimuove dall'immagine i marker più piccoli di 20 pixel
fgm4 = bwareaopen(fgm3, 20);

I3 = I;
% i massimi ora corretti vengono sovrapposti all'immagine I3

```

```

I3(fgm4) = 255;
subplot(222); imshow(I3), title('Massimi regionali modificati ←
    sovrainposti \newline alla immagine originale(fgm4)')

% graytresh: rende binaria l'immagine Iobrcbr tramite la funzione ←
    graytresh
bw = im2bw(Iobrcbr, graythresh(Iobrcbr));
subplot(223); imshow(bw), title('Apertura-chiusura sogliata \newline ←
    tramite ricostruzione (bw)');

% per ogni pixel calcola la sua distanza dal più vicino valore non ←
    zero
% estrae lo scheletro delle zone di influenza degli oggetti in primo ←
    piano.
% serve a limitare l'espansione dell'algoritmo di watershed al passo
% successivo.
D = bwdist(bw);

DL = watershed(D);

% background markers - Linee di confine del Watershed
bgm = DL == 0;

% quelli che prima erano i massimi regionali ora sono i minimi: a ciò ←
    viene
% associato lo scheletro delle zone d'influenza.
gradmag2 = imimposemin(gradmag, bgm | fgm4);
subplot(224); imshow(gradmag2), title('gradmag2s')

% esegue l'algoritmo watershed per estrarre i contorni definitivi dei
% marker
L = watershed(gradmag2);
figure, subplot(221); imshow(label2rgb(L)), title('watershed');
I4 = I;

% ricalca i contorni degli oggetti in primo piano
I4(imdilate(L == 0, ones(3, 3)) | bgm | fgm4) = 255;
subplot(222), imshow(I4), title('Markers e contorni degli oggetti \←
    newline sovrainposti alla immagine originale (I4)')

% le regioni vengono colorate
Lrgb = label2rgb(L, 'jet', 'w', 'shuffle');
subplot(223), imshow(Lrgb)
title('Colazione della matrice \newline di watershed (Lrgb)')

subplot(224), imshow(I), hold on
himage = imshow(Lrgb);
% il canale alfa permette di rendere trasparente le tonalità di colore
% dell'immagine Lrgb
set(himage, 'AlphaData', 0.3);
title('Immagine Lrgb sovrainposta \newline trasparentemente alla ←
    immagine originale')

```

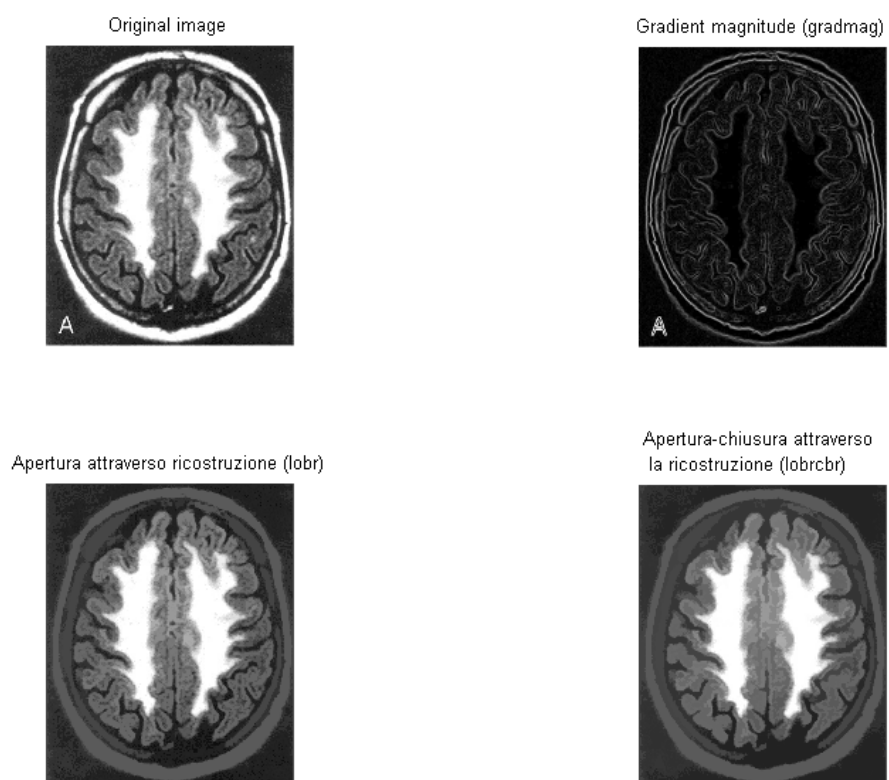


Figura 6: Fasi di Pre-Processing

Massimi regionali di apertura-chiusura  
attraverso ricostruzione (fgm)



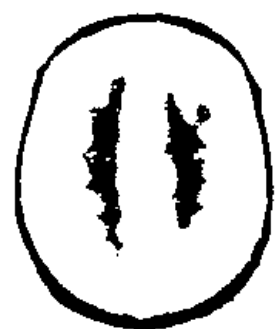
Massimi regionali modificati sovrainposti  
alla immagine originale(fgm4)



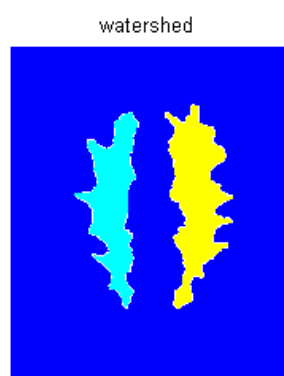
Apertura-chiusura sogliata  
tramite ricostruzione (bw)



gradmag2s







Markers e contorni degli oggetti  
sovraimposti alla immagine originale (14)



Colazione della matrice  
di watershed (Lrgb)



Immagine Lrgb sovrainposta  
trasparentemente alla immagine originale

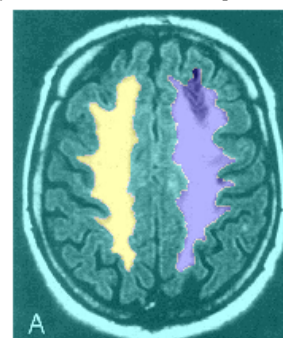


Figura 7: Segmentazione della Materia Bianca

## 4 Illustrazione del codice

L'obiettivo della prima parte del codice è quella di curare l'immagine per creare un affidabile gradiente. Presa l'immagine originale, questa viene subito convertita in scala di grigi.

Vengono definiti due filtri diversi `hy` e `hx` per evidenziare i bordi, rispettivamente, verticali e orizzontali dell'immagine. Con questi due filtri vengono create le matrici `Iy` e `Ix` necessarie per creare il gradiente.

Viene successivamente creato un elemento morfologico strutturante `se` di forma discoidale di raggio 10 pixel. Con `se` vengono effettuate diverse operazioni morfologiche di *apertura*, *erosione*, *dilatazione* e *ricostruzione*, con lo scopo di individuare i massimi regionali che verranno successivamente estratti tramite la funzione `imregionalmax()`. In tali circostanze i massimi regionali possono presentare della frammentazione tra loro, per questo viene effettuato un ulteriore ciclo di chiusura ed erosione per la loro dilatazione utilizzando un altro elemento strutturante `se2`.

Il passo successivo è l'individuazione dei *bacini* entro cui deve essere limitato l'algoritmo di watershed. Usando l'immagine `black-white` e applicandovi la funzione `bwdist()` si ottiene lo scheletro delle zone d'influenza su cui viene successivamente lanciato l'algoritmo di watershed. Ciò che ne risulta viene poi colorato e sovrapposto all'immagine originale per mostrare il lavoro compiuto.

## 5 Metodo alternativo alla segmentazione watershed: Thresholding

Come alternativa all'algoritmo di Watershed è possibile usare l'approccio basato sulla tecnica del thresholding. È il metodo più semplice per la segmentazione dell'immagine ed è utilizzato per creare immagini binarie. I pixel individuali, in un'immagine, sono segnati come *pixel oggetto* se il loro valore è più grande di un *valore soglia* o come *pixel di sfondo* altrimenti. Questa tecnica di thresholding viene definita *threshold above*.

Altre varianti sono *threshold below* che è l'opposto del precedente, *threshold inside* dove i pixel sono etichettati come oggetti se il suo valore è tra due soglie; *threshold outside* è il contrario del *threshold inside*.

Tipicamente ad un *pixel oggetto* viene attribuito il valore 1, mentre al *pixel di sfondo* viene assegnato il valore 0.

Il metodo di Otsu serve ad effettuare la selezione della soglia di segmentazione, ovvero esegue la sogliatura automatica dell'istogramma dell'immagine sorgente.

L'algoritmo presume che nell'immagine da sogliare siano presenti due sole classi, primo piano e sfondo, e quindi calcola la soglia ottima per separare queste due classi minimizzando la varianza intra classe.

### 5.1 La variante

Nella risoluzione dell'elaborato è stata implementata una variante senza far uso dell'algoritmo Watershed che segmenta la materia bianca facendo uso di diverse operazioni morfologiche che hanno portato comunque ad un risultato soddisfacente. Sfruttando il fatto che una proprietà della materia bianca è quella

di essere strettamente connessa e omogenea, la funzione `imextendedmin()`, con una connettività (secondo parametro) molto alta, si é rivelata sufficiente.

```
close all;
clc;
clear;
A = imread('images\256_MRI_z_109.png');
subplot(221); imshow(A); title('Immagine di partenza');

% effettua denoising dell'immagine tramite median filtering a [2,2]
A = medfilt2(A,[2,2]);

% crea l'elemento morfologico strutturante
se = strel('disk',5);

% applica il filtering sull'immagine tramite disk di dimensione 5.
% imtophat esegue l'apertura morfologica e sottrae il risultato
% dall'immagine originale
imc = imtophat(A,se);
% applica il filtering sull'immagine tramite disk di dim. 5
% imbothat esegue la chiusura morfologica e sottrae il risultato
% dall'immagine originale
ima = imbothat(A,se);

% eseguo il miglioramento del contrasto dell'immagine tramite top-hat ←
% e
% bottom-hat filtering
Ienh = imsubtract(imadd(imc,A),ima);

% estraggo l'immagine complementare
ic = imcomplement(Ienh);

% la funzione seguente serve ad estrarre i minimi regionali dall'←
% immagine
% 60 e' il valore minimo delle valli.
iem = imextendedmin(ic, 60);

% impongo i minimi sull'immagine originale
iimp = imimposemin(ic,iem);
iimp = imcomplement(iimp);
subplot(222); imshow(Ienh); title('Immagine a contrasto migliorato');

Lrgb = label2rgb(iem, 'jet', 'k', 'shuffle');
subplot(223); imshow(Lrgb); title('Valli evidenziate da imextendedmin←
()')

subplot(224); imshow(A); hold on
himage = imshow(Lrgb);

% il canale alfa permette di rendere trasparente le tonalit  di ←
% colore
% dell'immagine Lrgb
set(himage, 'AlphaData', 0.2);
title('Immagine Lrgb sovrainposta \newline trasparentemente alla ←
immagine originale')
```

## 5.2 Illustrazione del codice

Per prima cosa viene effettuato il **denoising** dell'immagine tramite **median filtering** al fine di ridurre il rumore presente. Successivamente viene creato l'elemento strutturante discoidale **se** di raggio 5 pixel con il quale vengono effettuate le operazioni morfologiche.

**imtophat** esegue l'apertura morfologica ovvero erosione e dilatazione; il risultato di questa operazione viene sottratto all'immagine originale. L'apertura morfologica viene utilizzata per rimuovere piccoli oggetti in primo piano, preservando forma e dimensione.

`imbothat` esegue la chiusura morfologica, ovvero dilatazione ed erosione; il risultato dell'operazione viene sottratto all'immagine originale.

Eseguendo l'unione dell'immagine sorgente con il risultato dell'immagine elaborata dal filtro top-hat e successivamente sottraendola all'immagine elaborata dal filtro bottom-hat, si ottiene un'immagine a contrasto maggiore, avendo preservati sia dimensione che forma degli oggetti presenti.

Complementando poi il risultato si ottiene un'immagine le cui valli, ovvero bassi valori di intensità, sono composte da ciò che nell'immagine originale era invece rilievo. In questo modo, applicando poi `imextendedmin` sull'immagine ottenuta, e specificando un alto valore di connettività, riusciamo a considerare tutte le valli che compongono la materia bianca.

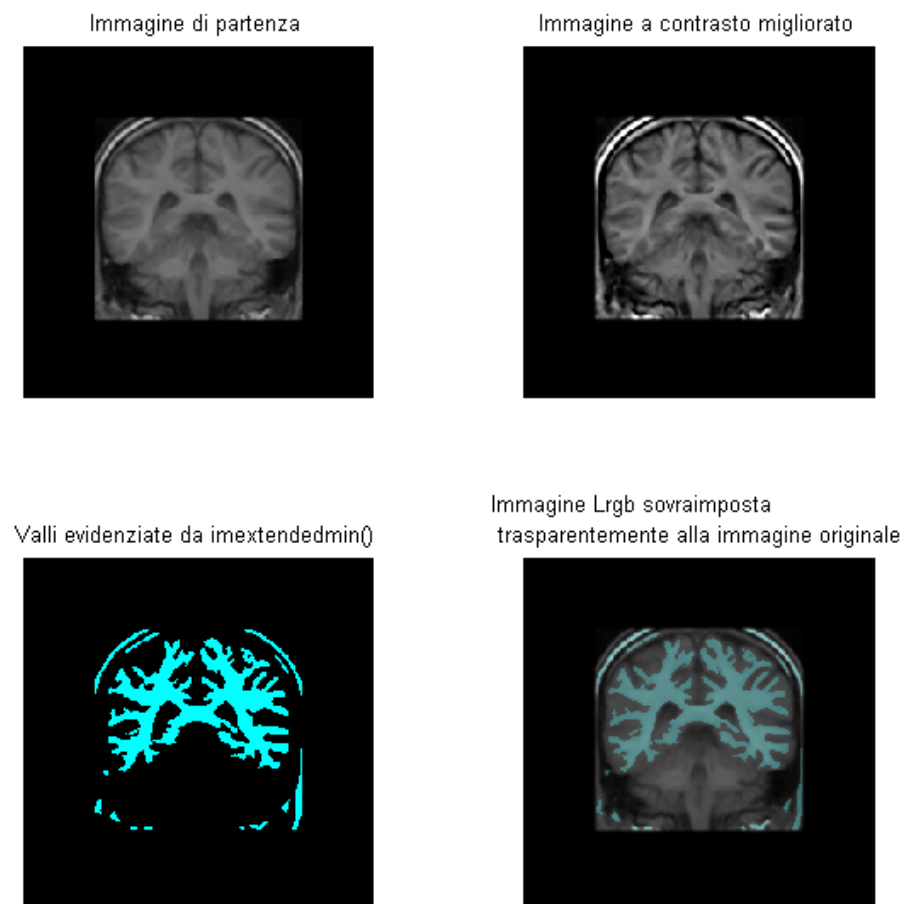


Figura 8: Esecuzione della segmentazione alternativa

## 6 Conclusioni

L'algoritmo watershed, come per tutti gli algoritmi inerenti la segmentazione di immagini, risulta essere di difficile implementazione. I problemi di over-segmentazione da un lato, e quelle di sotto-segmentazione dall'altro, rendono molto importante la fase di pre-processing e di pre-segmentation per rendere efficace la trasformazione. In queste fasi risultano essere decisive le variabili in gioco, quale ad esempio il raggio da scegliere per l'elemento strutturante. Sta quindi alla sensibilità del programmatore adattare a diverse immagini il tipo e il valore delle variabili, in modo da raggiungere il risultato migliore.

La variante illustrata evidenzia come talvolta soluzioni semplici possono essere comunque efficaci, anche se in modo grezzo. La segmentazione di immagini può quindi, in determinati casi, venire implementata solo attraverso l'utilizzo di operazioni morfologiche che agiscano sull'intensità dell'immagine, evidenziandone le parti desiderate. Il tutto può essere ricondotto ad una trasformazione che fa uso di thresholding histogram-based.

### 6.1 Confronto tra le due modalità

Provando ad utilizzare il metodo alternativo sull'immagine usata dall'algoritmo di watershed otteniamo la figura 9. Come si può notare il risultato è diverso da quello ottenuto precedentemente e inoltre segmenta parti che non appartengono alla materia bianca e tutto ciò contribuisce a rendere la segmentazione troppo approssimativa.

Ora analizziamo il comportamento dell'algoritmo di watershed con la figura utilizzata nel *codice alternativo*: figure 10, 11, 12.

Il primo algoritmo risulta essere efficiente fino al momento della ricerca dei massimi regionali, successivamente, però, Watershed fallisce e la risultante segmentazione non è appropriata, nonostante sia stato tenuto conto dell'adattamento alla nuova immagine dei parametri come il raggio degli elementi morfologici strutturanti e la connettività della funzione `imregionalmax`.

Ciò che si può concludere è che non esiste un algoritmo migliore degli altri nei problemi di segmentazione. Sta nuovamente al programmatore capire qual è il miglior metodo da usare per ogni specifica immagine.

Immagine di partenza

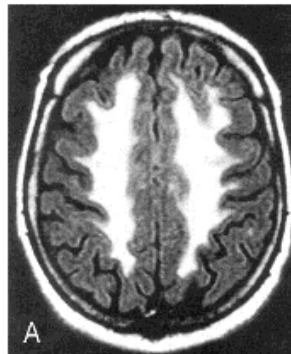
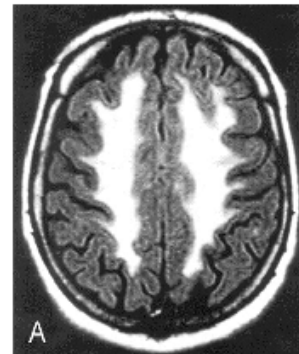


Immagine a contrasto migliorato



Valli evidenziate da `imextendedmin()`

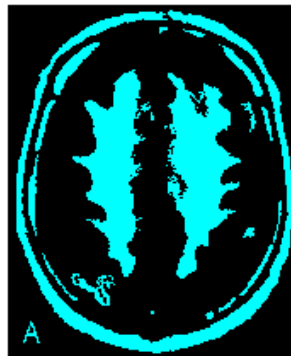


Immagine Lrgb sovrainposta  
trasparentemente alla immagine originale

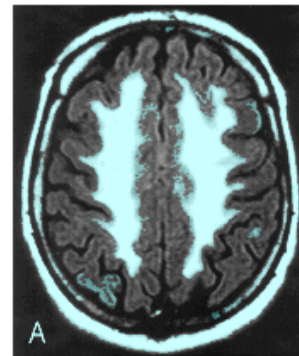


Figura 9: Esecuzione della segmentazione alternativa con la stessa immagine usata nell'algoritmo di Watershed

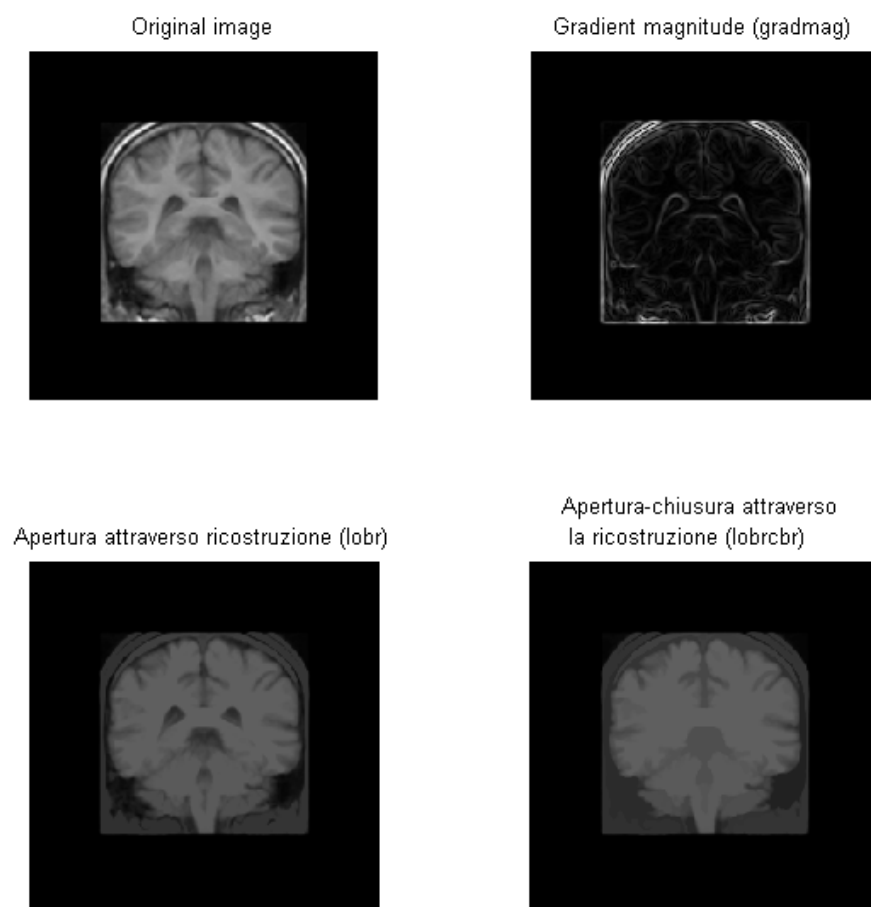
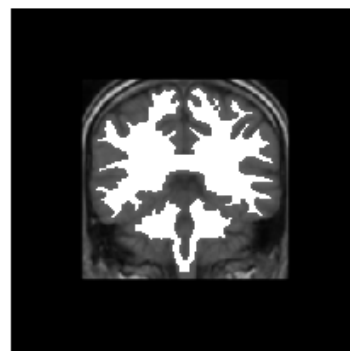


Figura 10: Esecuzione dell'algoritmo di watershed con l'immagine usata nel secondo procedimento

Massimi regionali di apertura-chiusura  
attraverso ricostruzione (fgm)



Massimi regionali modificati sovrapposti  
alla immagine originale(fgm4)



Apertura-chiusura sogliata  
tramite ricostruzione (bw)



gradmag2s



Figura 11: Esecuzione dell'algoritmo di watershed con l'immagine usata nel secondo procedimento



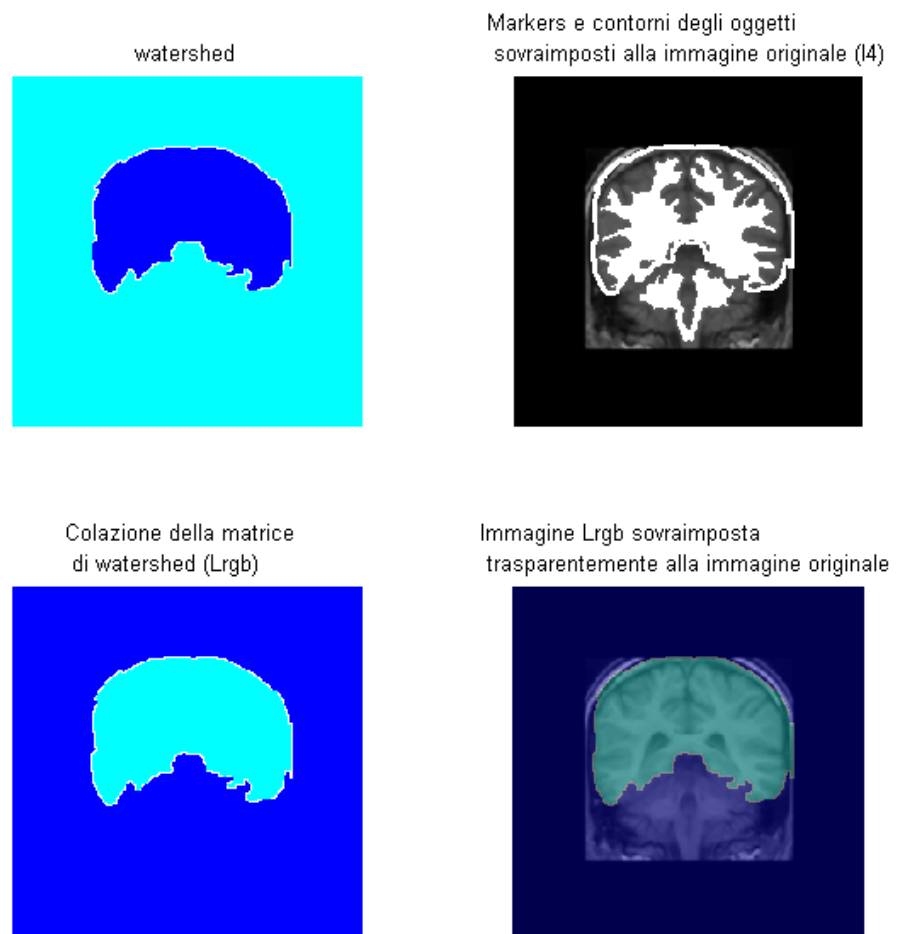


Figura 12: Esecuzione dell'algoritmo di watershed con l'immagine usata nel secondo procedimento