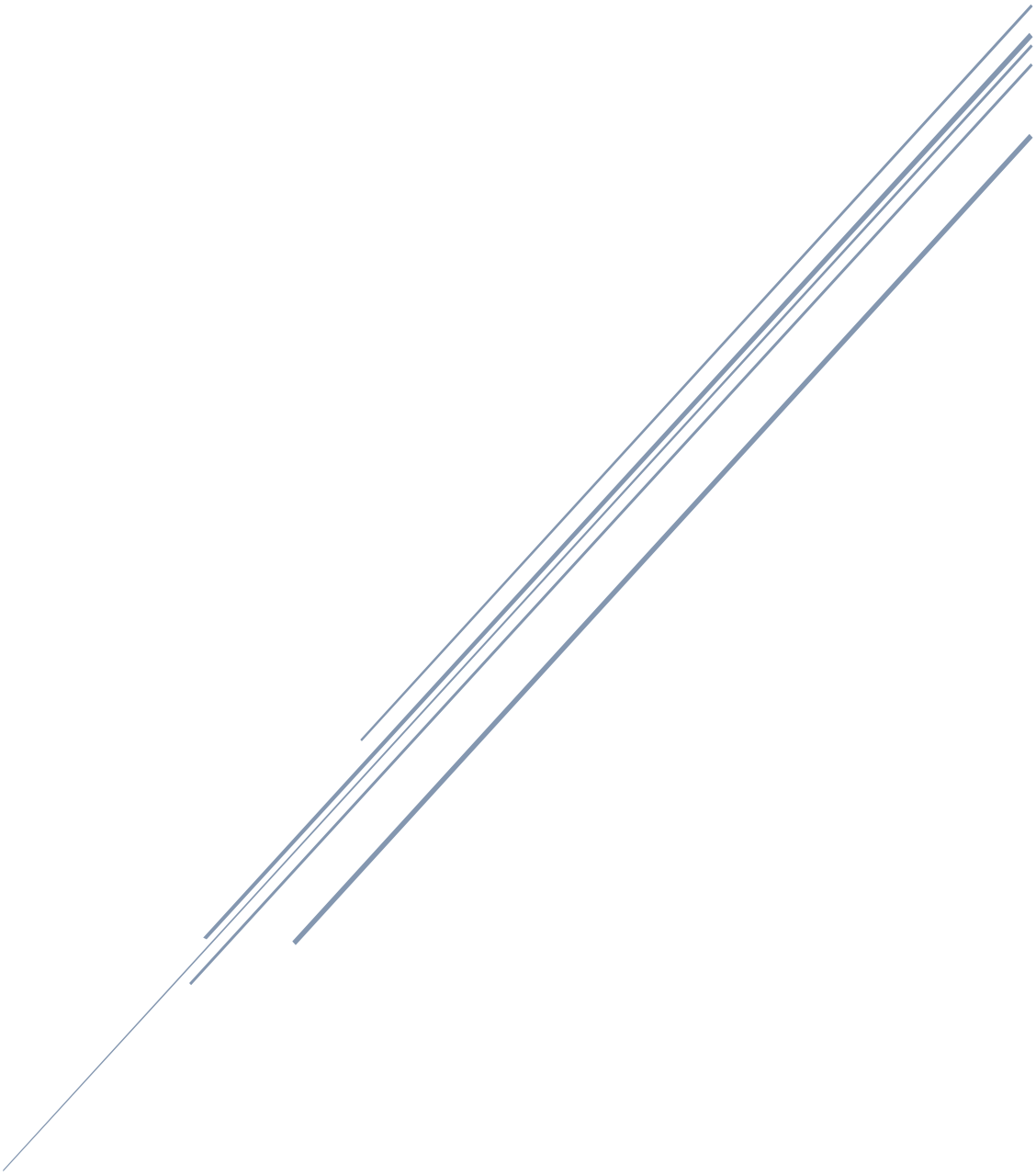
TRAVAUX PRATIQUES -IMA 203

TP4 : Modèles déformables



Télécom Paris
Baccari Mohamed Mootez

I. Paramètres

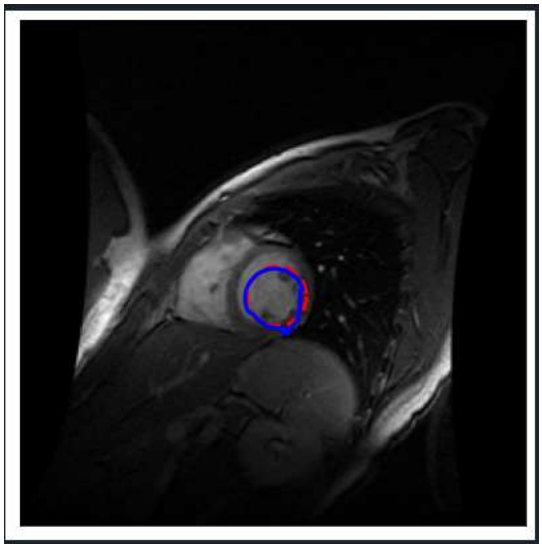
Tout le long de cette partie, nous allons donner, pour la méthode des contours actifs paramétriques et la méthode de chan et Vese, l'interprétation de chaque paramètre, son rôle et son influence sur le résultat.

1. Méthodes de contours actifs paramétriques

a. Init :

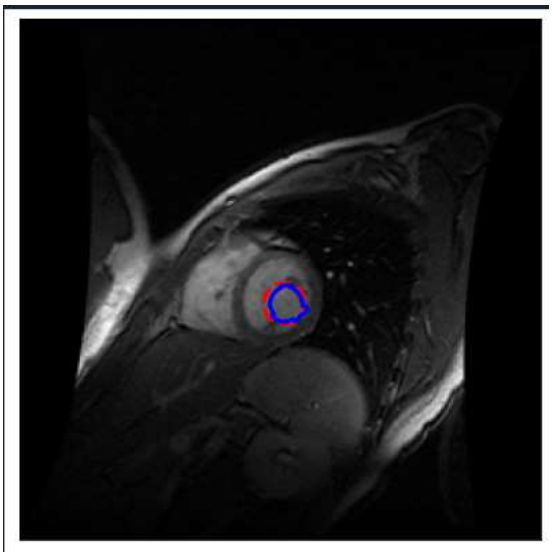
Ce sont les coordonnées initiales du serpent.
Comme l'objet qu'on souhaite segmenter dans ce TP possède une forme circulaire, on prend donc un cercle come contour initiale.

Exemple 1



```
s = np.linspace(0, 2*np.pi, 100)
r = 140 + 15*np.sin(s)
c = 130 + 15*np.cos(s)
init = np.array([r, c]).T
```

Exemple 2



```
s = np.linspace(0, 2*np.pi, 100)
r = 140 + 10*np.sin(s)
c = 130 + 10*np.cos(s)
init = np.array([r, c]).T
```

En exécutant l'algorithme de « Snake » à plusieurs reprises. On constate que la convergence vers la région voulu dépend fortement de l'initialisation des contours. En effet, si on change l'initialisation, tout en fixant les autres paramètres, la solution change et on obtient une segmentation différente.

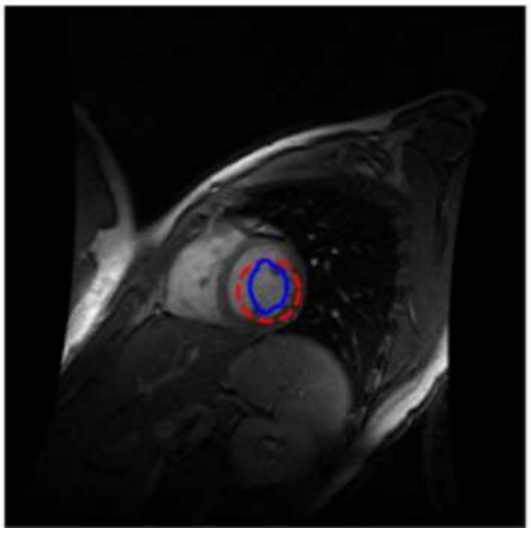
Dans l'exemple 2, j'ai essayé de choisir un contour qui se situe à l'intérieur de la région souhaité, le résultat montre qu'on n'arrive pas à la convergence voulue.

En visualisant la vidéo et en cherchant sur Google, j'ai remarqué qu'on pourra surmonter ce problème en ajoutant, soit une force de ballon, soit la technique de gradient diffusé pour que l'algorithme dépend moins de l'initialisation.

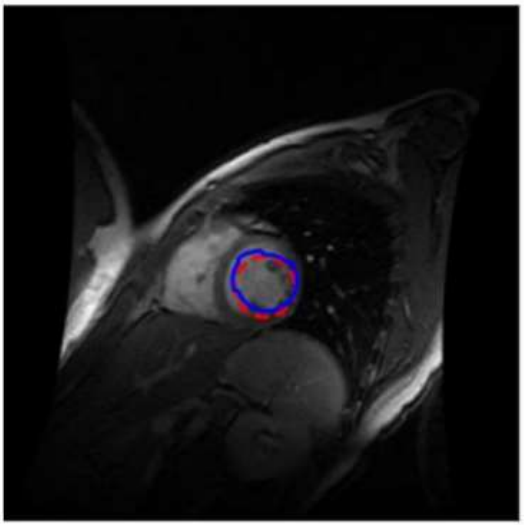
b. Alpha :

La valeur de alpha agit directement à la longueur du « Snake ». En effet, on remarque que plus on augmente la valeur de alpha, plus on a une contraction rapide du serpent.

Résultat alpha =100



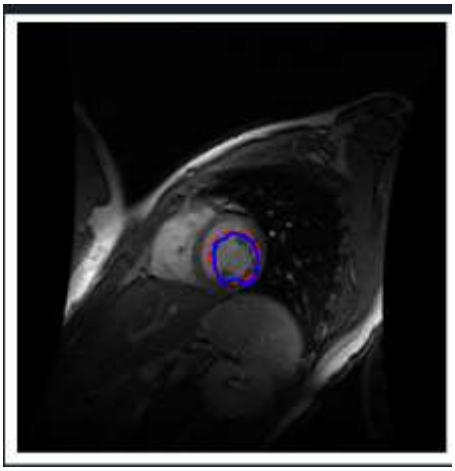
Résultat alpha = 0,5



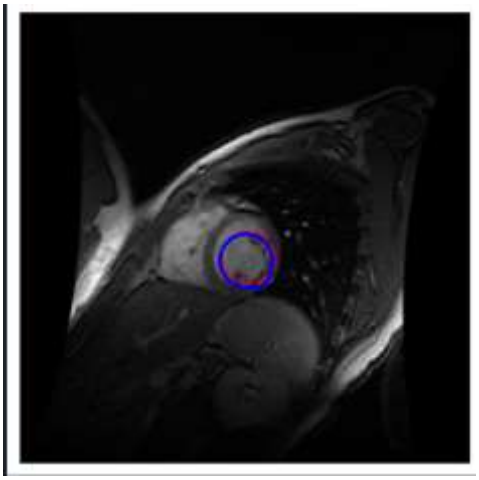
c. Beta :

Il s'agit d'un paramètre qui influence la souplesse du serpent. C'est-à-dire si on a des zone pointus ou lisses.

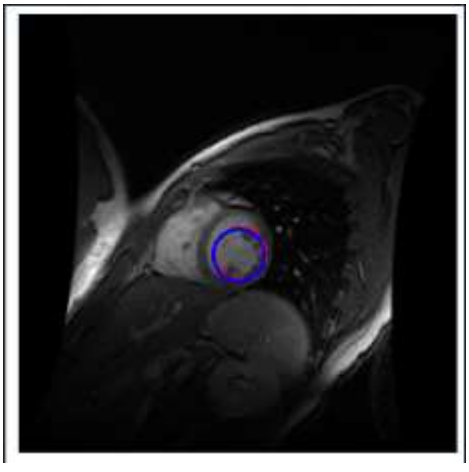
Beta = 0



Beta = 10



Beta = 100

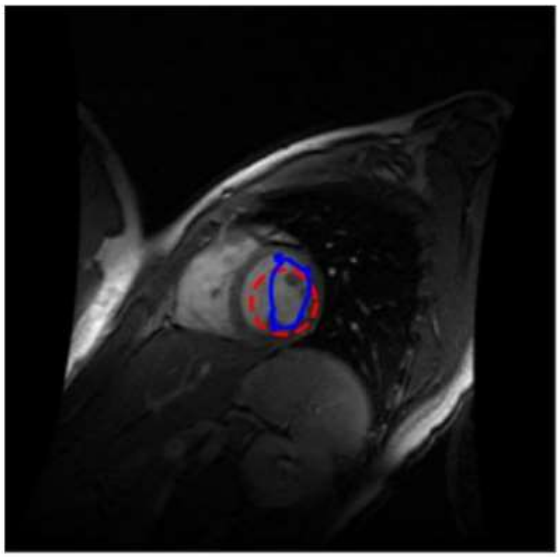


Pour le cas ou beta = 0, on voit l'apparence de contours rugueux. Ensuite, plus on augmente la beta, plus on a un serpent plus lisse. Si beta devient très grand, on se retrouve avec la forme d'un cercle.

d. W_line :

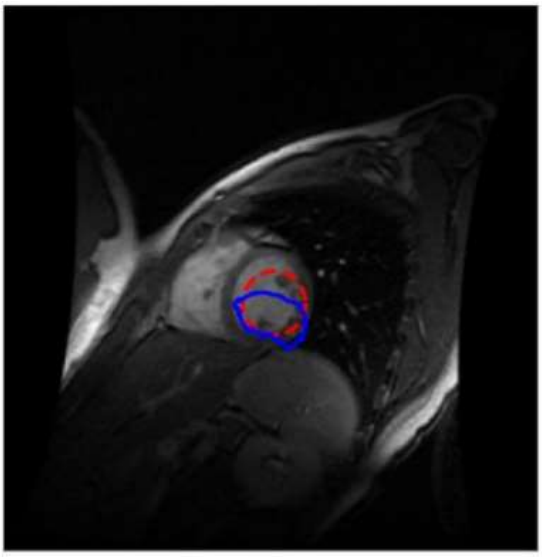
Ce paramètre modélise l'attraction pour la luminosité. En effet, si on utilise des valeurs positives, on est plus proche des zones lumineuses et si on utilise des valeurs négatives, on est plus proche des zones sombres.

W_line = 100



Les contours sont attirés vers une partie claire

W_line = -100



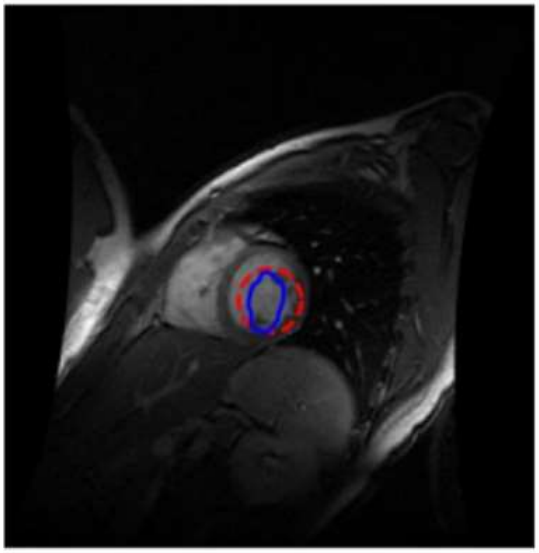
Les contours sont attirés par les lignes sombres

e. W_edge :

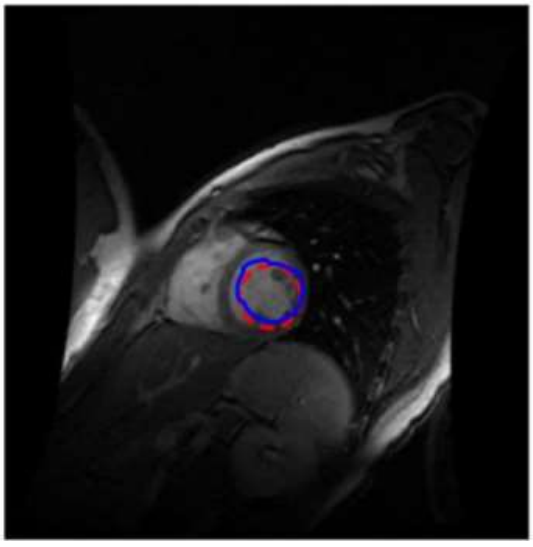
Ce paramètre contrôle l'attraction vers les bords. On distingue l'influence selon le signe.

- Si on prend des valeurs positives, le serpent est attiré par les contours ayant des gradients élevés.
- Si on prend des valeurs négatives, le serpent s'éloigne des bords

W_edge = -1



W_edge = 20

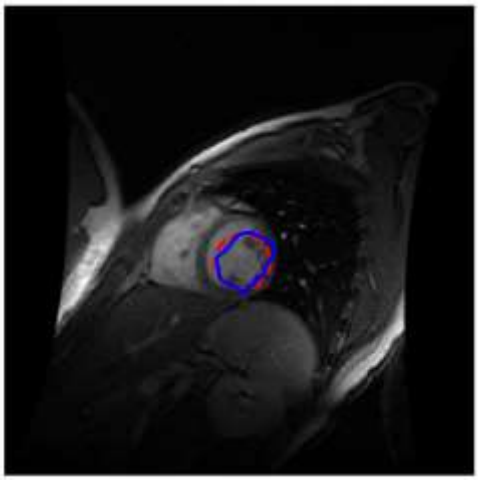


f. Gamma :

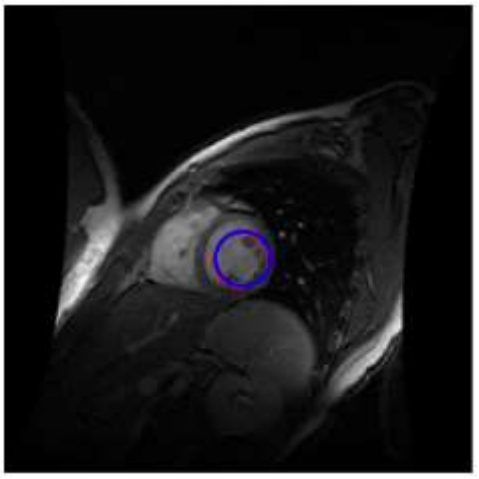
Le paramètre Gamma est celui qui modélise l'effet du gradient. En effet si on met gamma = 0 on désactive tous les effets du gradient. Pour tester ceci, on pourra fixer W_edge a une valeur négative (comme le montre les deux figures suivantes), logiquement, on s'éloignera du bord mais en mettant gamma = 0 rien ne se produit

⇨ On a donc enlevé l'effet du gradient

W_edge = -1 & gamma = 0,001



W_edge = -1 & gamma = 0



2. Méthodes de Chan et Vese

Il s'agit d'un modèle de contour actif en faisant évoluer un jeu de niveaux. Cette méthode pourra être utilisé afin de segmenter des objets sans des limites clairement définies.

a. Mu :

C'est le paramètre de « longueur du bord ». En effet, si on prend des valeurs élevées du Mu on obtient un bord rond. Par contre, si on prend des valeurs faibles, on obtient des objets plus petits.

Mu = 0,05

Mu = 0,25

Mu = 1

On remarque que lorsque mu est petite, on détecte beaucoup plus d'objet de l'image que lorsque mu est grande.

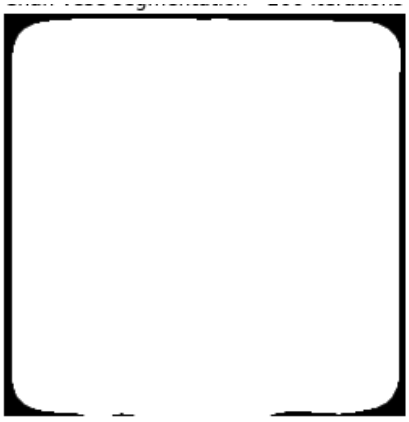
b. Lambda :

Les paramètres sont liés entre eux

- **Lambda 1** : est le paramètre qui modélise la différence « par rapport à la moyenne » pour la région de sortie avec la valeur « True ». Si elle est inférieure à lambda2, cette région aura une plus grande plage de valeurs que l'autre.
- **Lambda 2** : est le paramètre qui modélise la différence « par rapport à la moyenne » pour la région de sortie avec la valeur « False ». Si elle est inférieure à lambda1, cette région aura une plus grande plage de valeurs que l'autre.

Lambda 1 = 5 et lambda 2=1

Lambda 1 = 1 et lambda 2 = 5



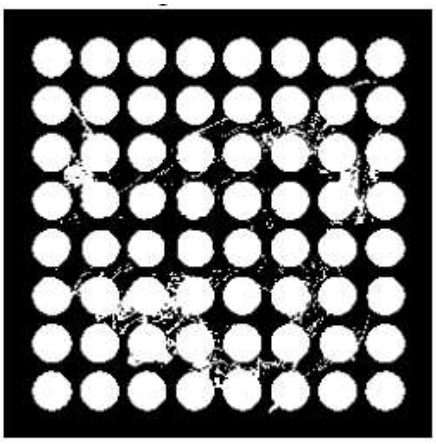
L'utilisation de la méthode de chan et vese nécessite d'effectuer un bon choix de lambda 1 et lambda 2. Comme le montre l'image 2, On a un résultat non raisonnable.

c. TOL :

C’est la tolérance de variation de niveau définie entre les itérations. Si la différence (en
norme L2) entre les ensembles de niveaux d'itérations successives normalisées par la zone de l'image est inférieure à cette valeur, l'algorithme supposera que la solution a été atteinte.

TOL= 10e-3

TOL= 10e-1



On peut considérer TOL comme une condition d’arrêt pour l’algorithme. En effet pour TOL=10-1. L’algorithme donne un résultat non interprétable et il converge en une seule itération

d. Max_iter :

Désigne le nombre maximal d’itération possible à faire avant que l’algorithme s’arrête.

Plus cette valeur est grande, plus on a la possibilité d’effectuer plus d’itération.

e. Dt :

C'est un facteur de multiplication qui sert à accélérer l'algorithme pour les calculs de chaque étape. Bien que des valeurs plus élevées puissent accélérer l'algorithme, des problèmes de convergence peuvent également être induits.

Dt = 0,5

Dt = 100

On remarque que l'exécution de l'algorithme est plus vite, mais les régions segmenter ne sont pas les même. Dans cet exemple, on remarque que les régions segmentées en bas sont plus rapproché lorsque on a augmenté ce paramètre.

f. Init_level_set :

Définit l'ensemble de niveaux de départ initial de l'algorithme.

Init_level_set = np.zeros(image.shape)

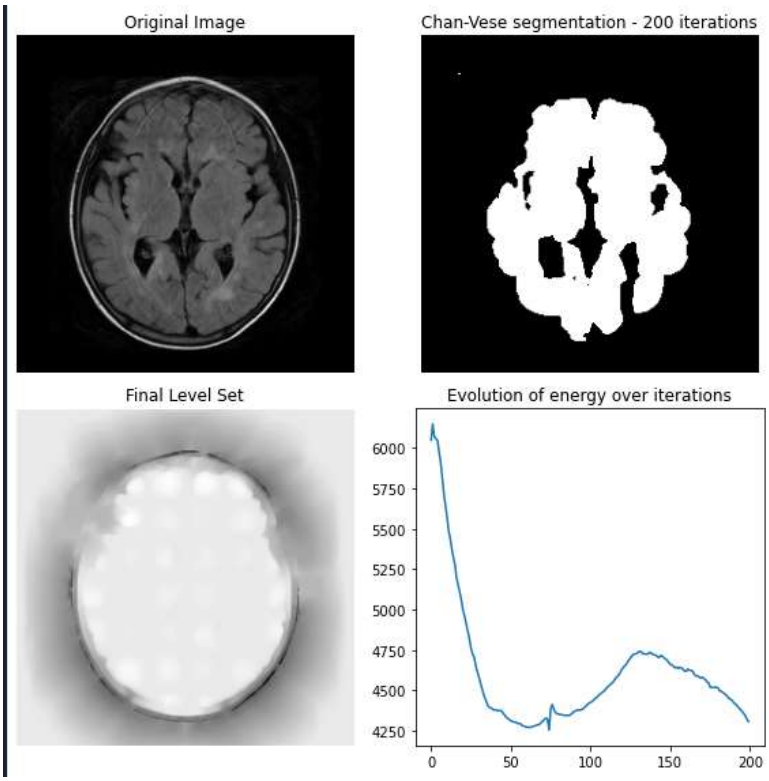
Init_level_set = np.ones(image.shape)

- L'algorithme de Chan et Vese est très sensible à l'initialisation. Pour une initialisation qui est totalement fausse, l'algorithme peut complétement diverger.

II. Segmentation

Tout le long de cette partie, on va détailler les étapes faites pour segmenter le mieux possible les structures anatomiques de l'image « brain2.bmp ».

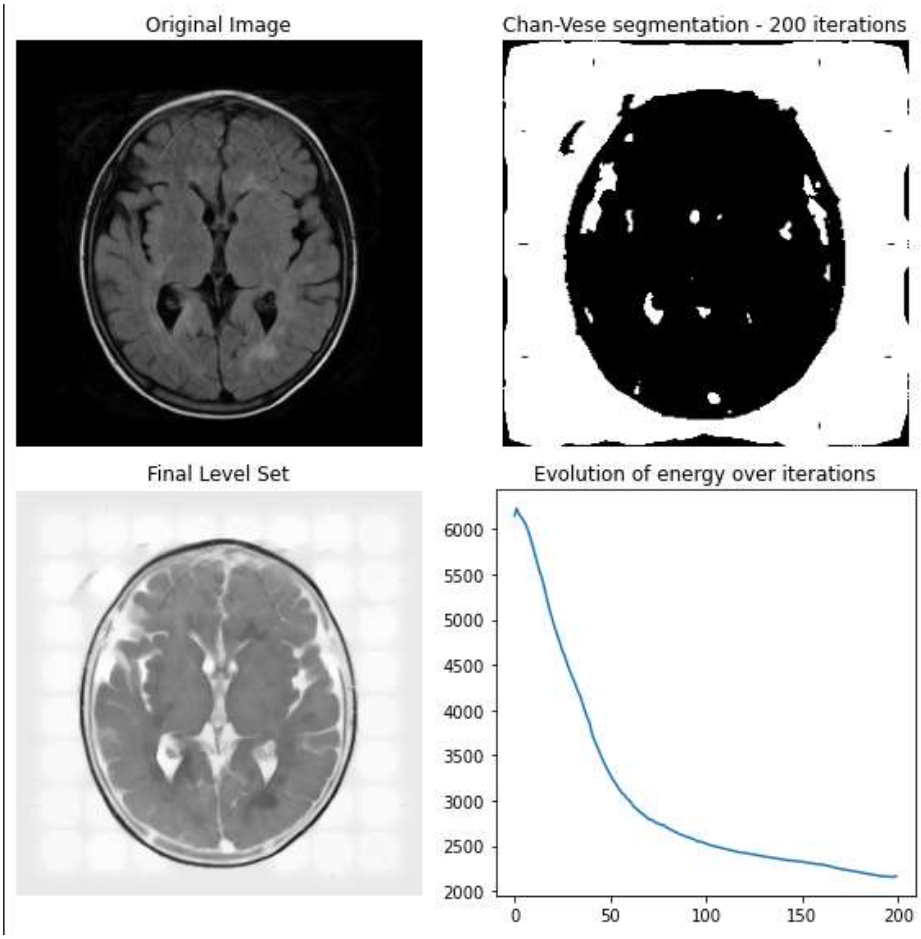
En utilisant l'algorithme de Chan et Vese, notre point de départ est le suivant :



```
cv = chan_vese(image, mu=0.25, lambda1=5, lambda2=1, tol=1e-3, max_iter=200,
               dt=0.5, init_level_set=init_ls, extended_output=True)
```

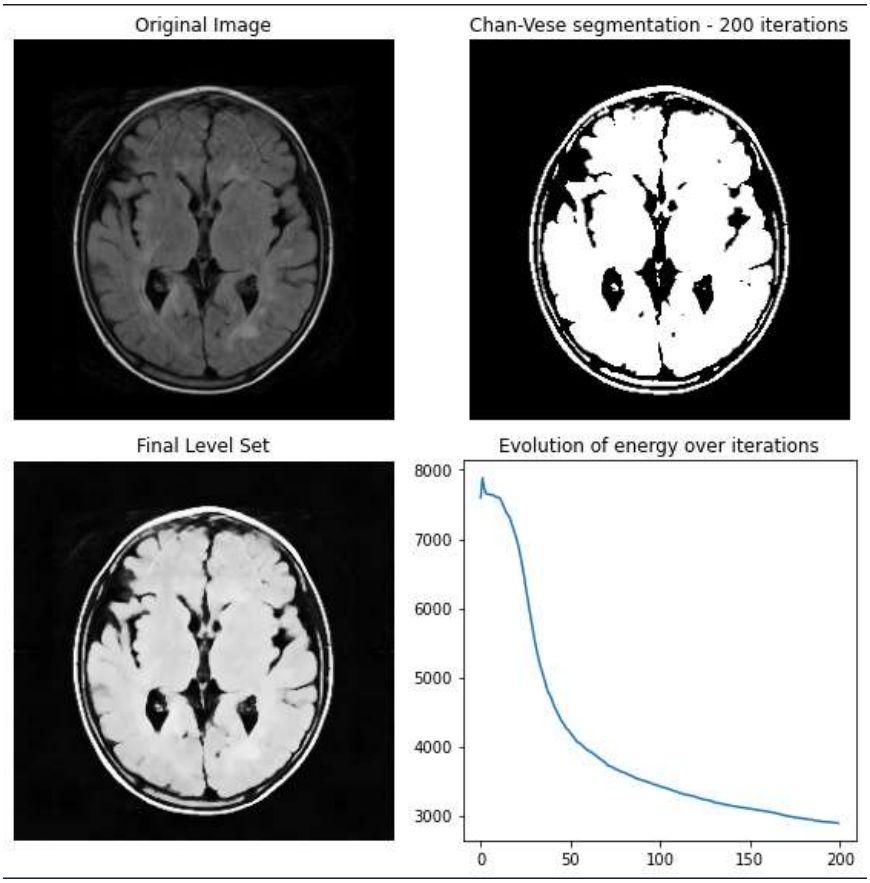
Il est clair que le résultat n'est pas satisfaisant, pour cela on commence à modifier les paramètres.

Au début, j'ai pris un Mu faible (0,025) pour prendre en considération tous les objets de l'image




```
cv = chan_vese(image, mu=0.025, lambda1=5, lambda2=1, tol=1e-3, max_iter=200,
               dt=0.5, init_level_set=init_ls, extended_output=True)
```

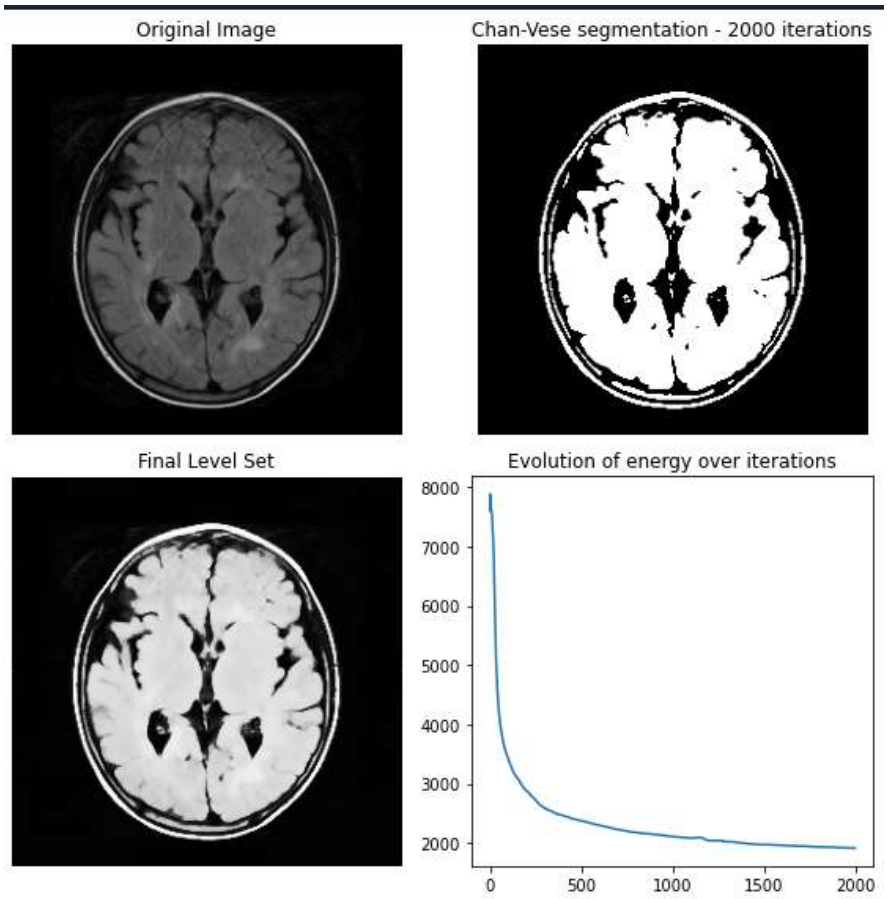
Ensuite, j'ai pris deux valeurs pour lambda1 et lambda 2 tel que lambda 1 > lambda 2 pour aboutir à un résultat raisonnable.




```
cv = chan_vese(image, mu=0.025, lambda1=5, lambda2=3, tol=1e-3, max_iter=200,
               dt=0.5, init_level_set=init_ls, extended_output=True)
```

Après, j'ai gardé la tolérance à 1^{e}-3 vu qu'elle est déjà assez faible.

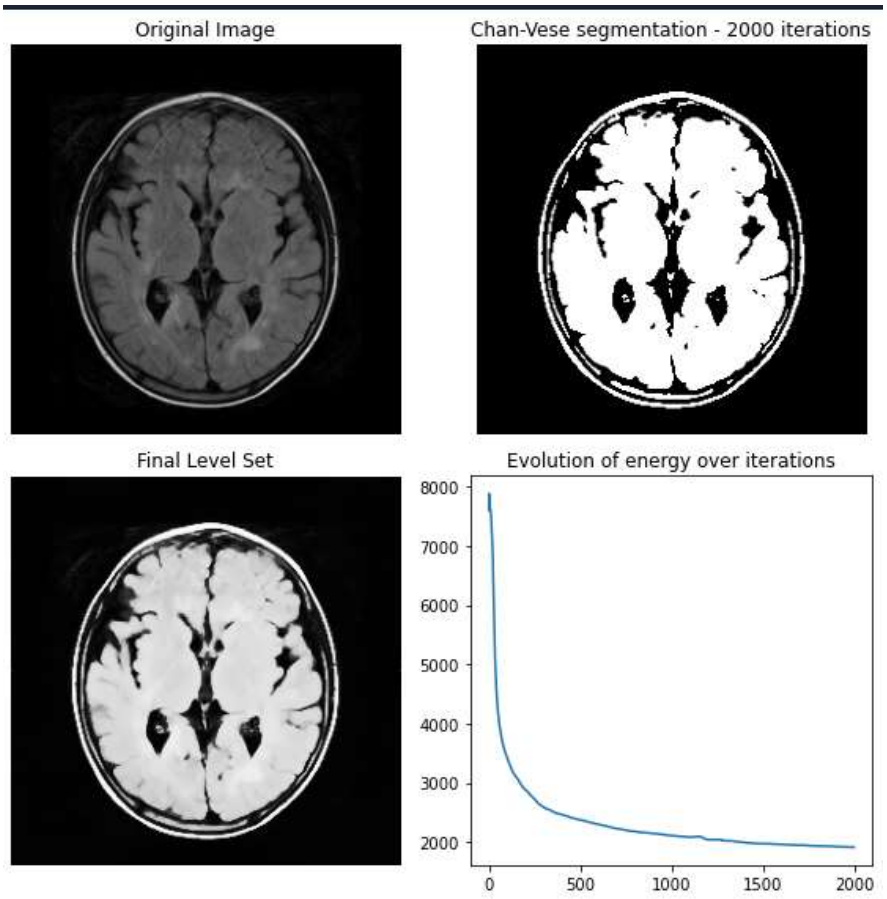
Pour le nombre d'itération, j'ai essayé de prendre une valeur élevée afin de garantir que l'énergie sois faible sans laisser l'algorithme tourner pour une longue durée et bien sur tout en ayant un résultat acceptable.



Ici, en arrivant a l'itération 2000, on remarque bien que l'énergie décroit lentement avec les itérations, on pourra donc dire qu'on est proche du minimum.

Pour éviter les problèmes de convergence, j'ai décidé de garder Dt à 0,5.

Le résultat final est donc :



Dans ce résultat, on voit bien qu'on pourra réaliser une segmentation. L'énergie est presque minimale et on converge dans quelques secondes.

Pour améliorer ses résultats, on pourra penser à des prétraitements.

J'ai voulu vous remerciez pour la réponse a mes questions lors des derniers TPs.

Au cours de ce TP, le choix des paramètres était basé sur le résultat obtenu visuellement. Dans la pratique, selon quels critères on choisit les meilleures valeurs ?

Merci encore une fois pour votre réponse.

Mon mail est : mohamed.baccari@telecom-paris.fr