

Travaux pratiques 6 - IMA203

Filipe Borges De Mattos Zamagna Lauar

Gabriel Barbutti de Lima Baker

Júlia Togashi De Miranda

Vinícius Akira Imaizumi

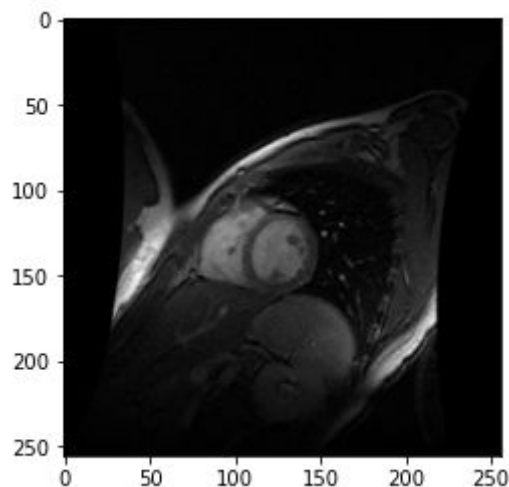
Ramon Galate Baptista Ribeiro

1 Paramètres

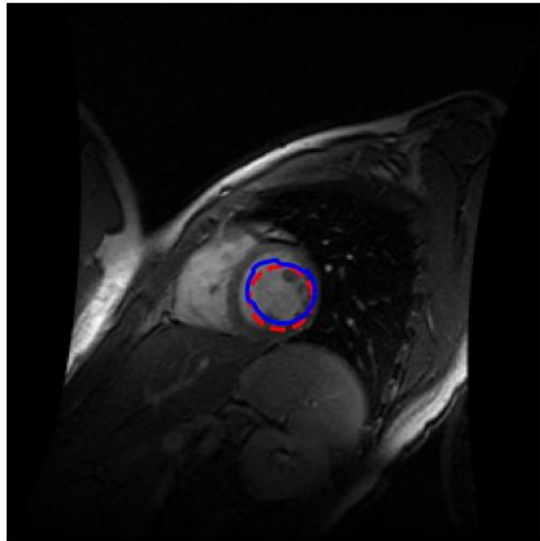
1.1 Méthode des contours actifs paramétriques

- init: L'initialisation des coordonnées du serpent.
- Alpha: Régule la taille de contraction du serpent. Plus grand est alpha, plus vite le serpent va se contracter.
- Beta: Régule le lissage. Plus grand est beta, plus lisse est le serpent.
- w_edge: contrôle l'attraction aux arêtes. Des valeurs négatives vont repousser le serpent des arêtes
- gamma: terme d'inertie pour régulariser la matrice A
- w_line: contrôle l'attraction aux régions plus claires. Des valeurs négatives vont faire l'attraction aux régions noires.

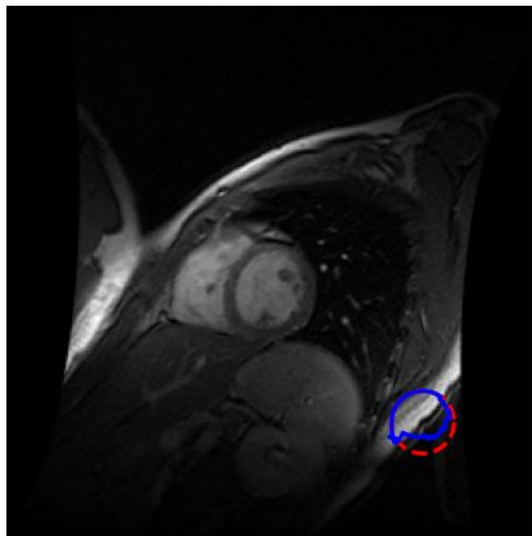
On fera les tests avec l'image 'coeurIRM':



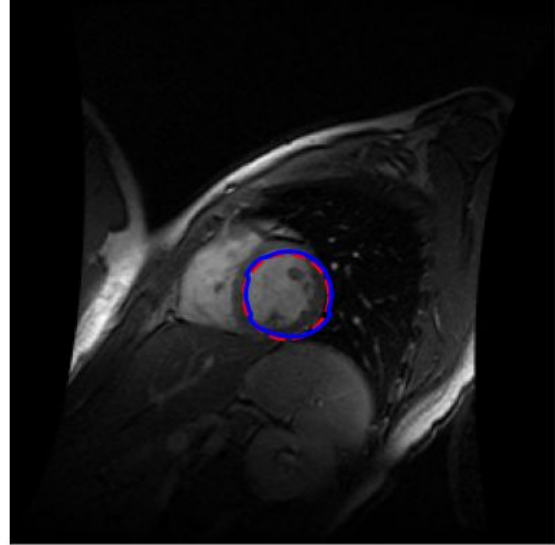
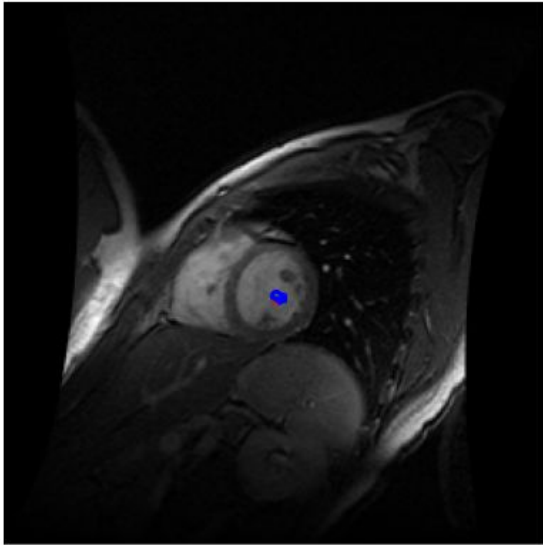
On partira des valeurs mises en défaut par la professeure pour étudier l'influence de chaque paramètre quand on les change. Les valeurs sont: init correspond à des points également espacé échantillonné d'un cercle centré en (140,130) et rayon égal à 15, alpha égal à .5, beta égal à 5, w_edge égal à 20 et gamma égal à .001. Pour ces paramètres on a l'image suivant, où le contour rouge est le contour initiale (paramètre init) et le contour bleu est le contour retourné par l'algorithme (le serpent):



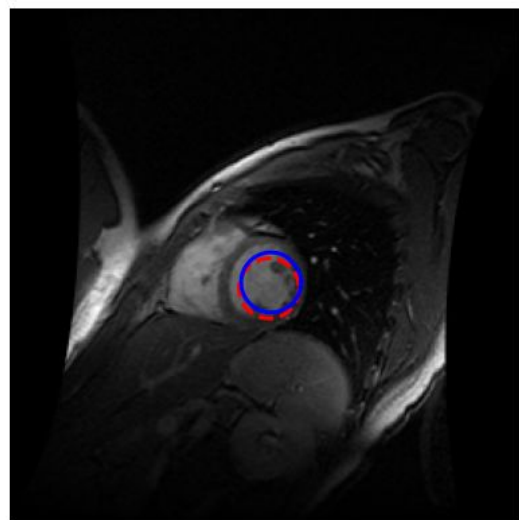
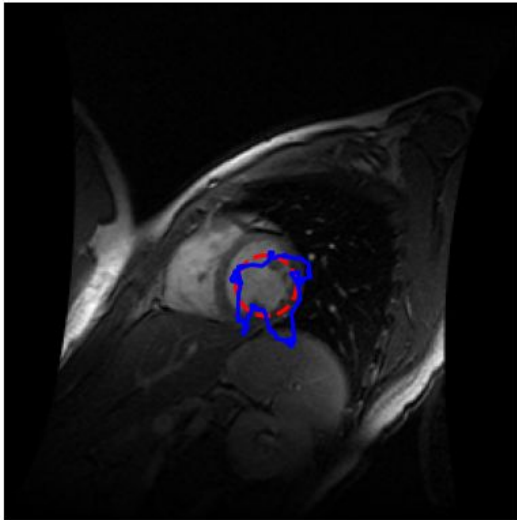
D'abord, si on change la position initiale du contour ou du rayon, le serpent va complètement changer. Par exemple, si on change la position initiale de (140,130) pour (200,200), la méthode va contourner une autre partie de l'image, comme on peut regarder ci-dessous.



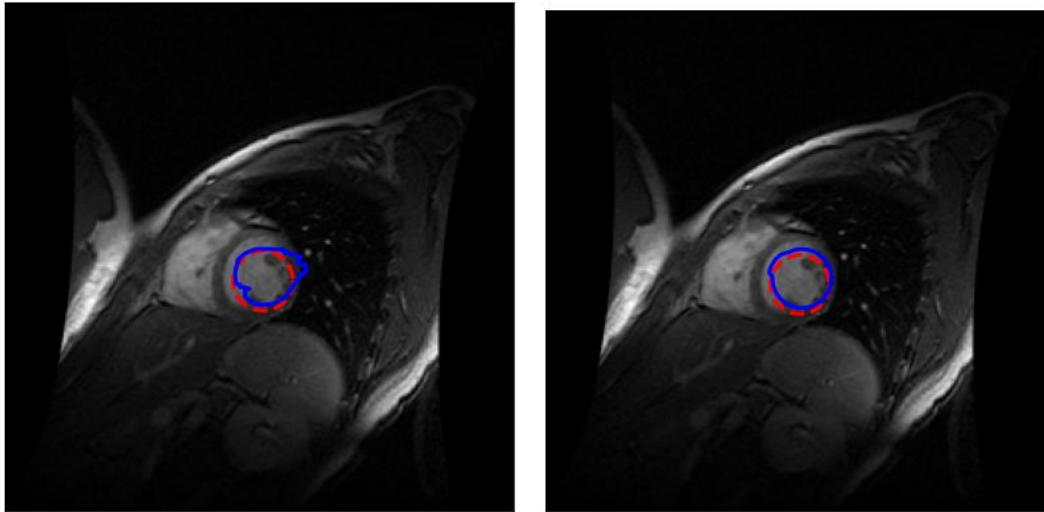
Par conséquent, on peut regarder que l'initialisation correcte est le paramètre le plus important pour que l'algorithme marche bien. On pourrait jamais contourner une partie en haut de l'image si on initialise le contour en bas. La taille du rayon est très importante aussi, il ne vaut rien si on initialise bien le points du contours si on met de rayon trop petit ou trop grand. Ci-dessous un peut regarder en gauche le contour avec un rayon très petit (égal à 2) et une bonne taille de rayon, égal à 20.



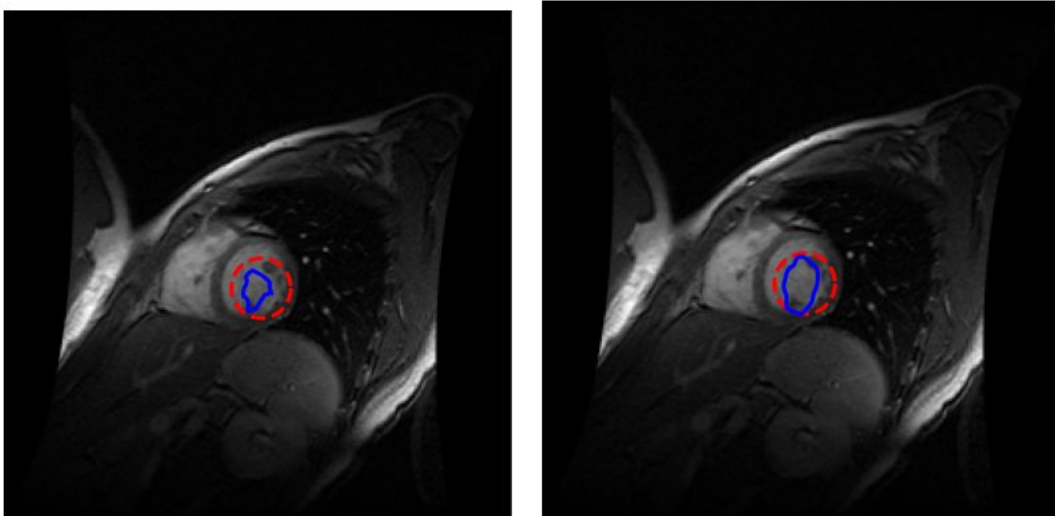
Comme on a dit, le paramètre alpha contrôle la taille de contraction du contour du serpent. Si on met alpha grand, le serpent sera plus contracté et si on met alpha petit le serpent sera moins contracté. Ci-dessous on peut regarder cela, en gauche alpha très petit, égal à .01 et en droite alpha grand, égal à 5.



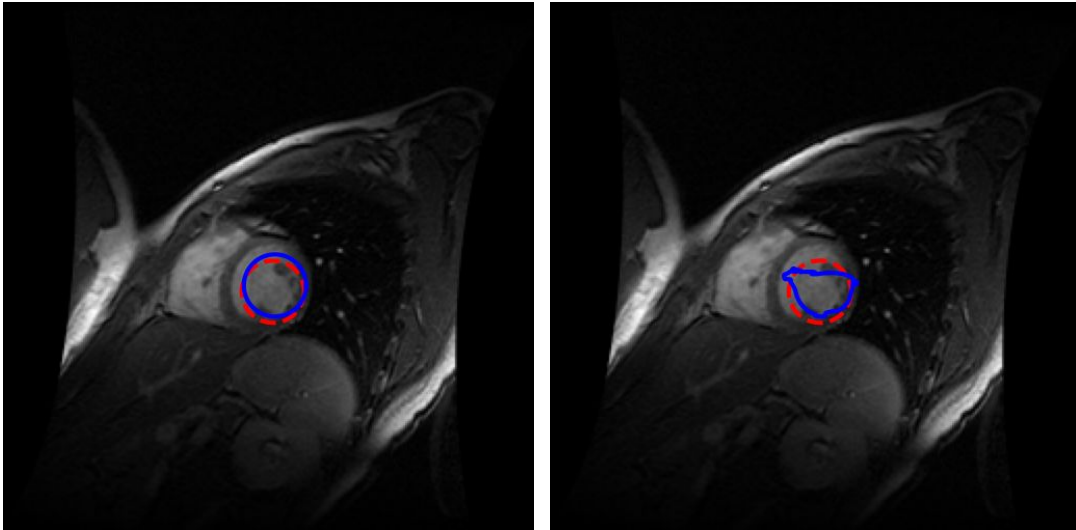
Le paramètre beta va contrôler le lissage du serpent. On peut regarder cela par rapport aux deux images ci-dessous. À gauche beta vaut .5 et à droite il vaut 50. On peut regarder que l'image en droite est beaucoup plus lisse que l'image en gauche.



Pour le paramètre w_edge , si on l'augmente, le serpent sera plus proche des arêtes, si on le diminue, il sera plus loin des arêtes, et si on met des valeurs négatives il sera repoussé des arêtes. On peut regarder cela ci-dessous, en gauche se trouve l'image avec w_edge petitit (égal à 2) et en droite avec w_edge négatif (égal à -2)



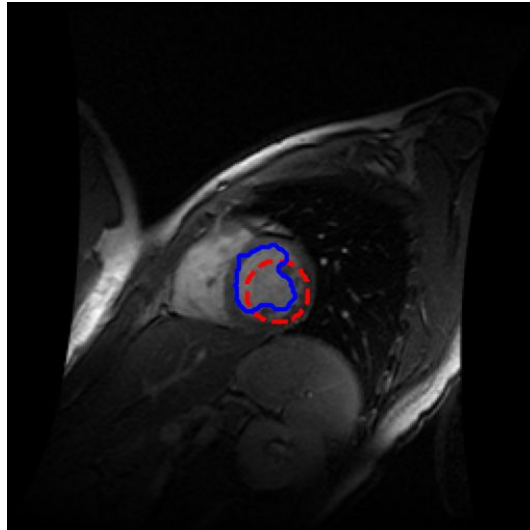
Par rapport à γ , si on l'augment, la segmentation finale s'éloigne du cercle initiale et quand on le diminue le résultat est un cercle plus parfait. Dans l'image à gauche on a utilisé un γ égal à $1e-5$ et dans l'image à droite il est égal à $1e-2$.



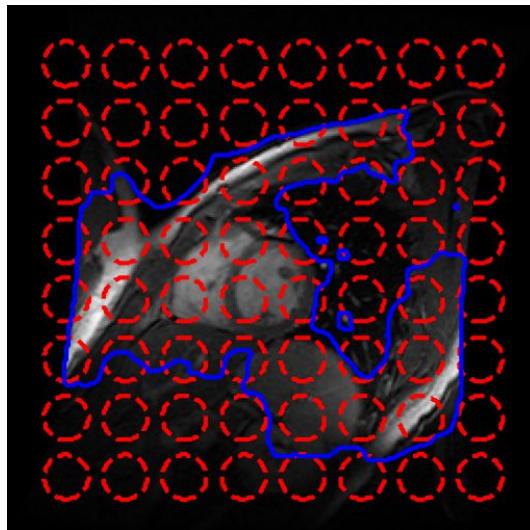
1.2 Méthode de Chan et Vese

- μ : terme de pénalité de la taille du contour. Il est responsable pour le balance entre un contour plus fidèle aux niveaux de gris de l'image (μ petit) et un contour plus compact (μ grand).
- λ_1 : poid pour la région "vraie" dans le résultat. Plus il est grand par rapport au λ_2 , plus cette région est petite.
- λ_2 : poid pour la région "faux" dans le résultat. Plus il est grand par rapport au λ_1 , plus cette région est petite.
- tol : tolérance entre itérations du `level_set` pour déterminer si l'algorithme a convergé. Si la norme L2 entre deux `level_set` de deux itérations successives normalisée par l'area de l'image est plus petite que cette valeur, l'exécution est finie.
- max_iter : maximum nombre d'itérations permis avant que l'algorithme soit fini.
- dt : facteur multiplicatif pour accélérer l'exécution de l'algorithme.
- `init_level_set`: `level_set` initial pour l'algorithme. Il doit être de la même taille de l'image ou une string indiquant un des initialisations standard.
- `extended_output`: variable booléenne pour indiquer si la fonction va retourner seulement la segmentation ou d'autres informations aussi.

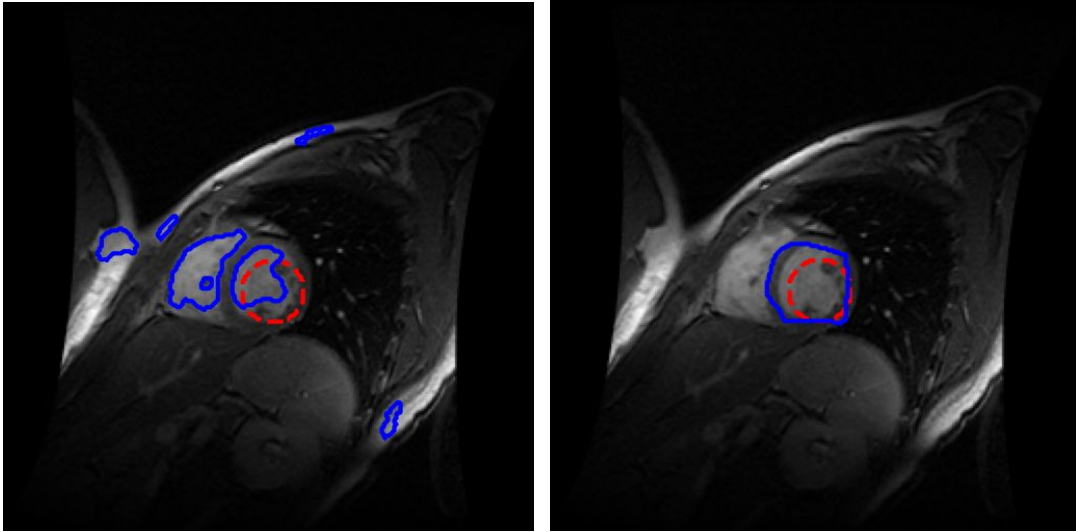
Pour comparer l'influence de chaque paramètre, on a utilisé la même image et le même cercle comme initialisation (paramètre `init_level_set`). Les autres paramètres choisis sont μ égale à .25, λ_1 égale à 10, λ_2 égale à , tol égale à $1e-4$, max_iter égale à 200 et dt égale à .5. Le résultat obtenu a été le suivant:



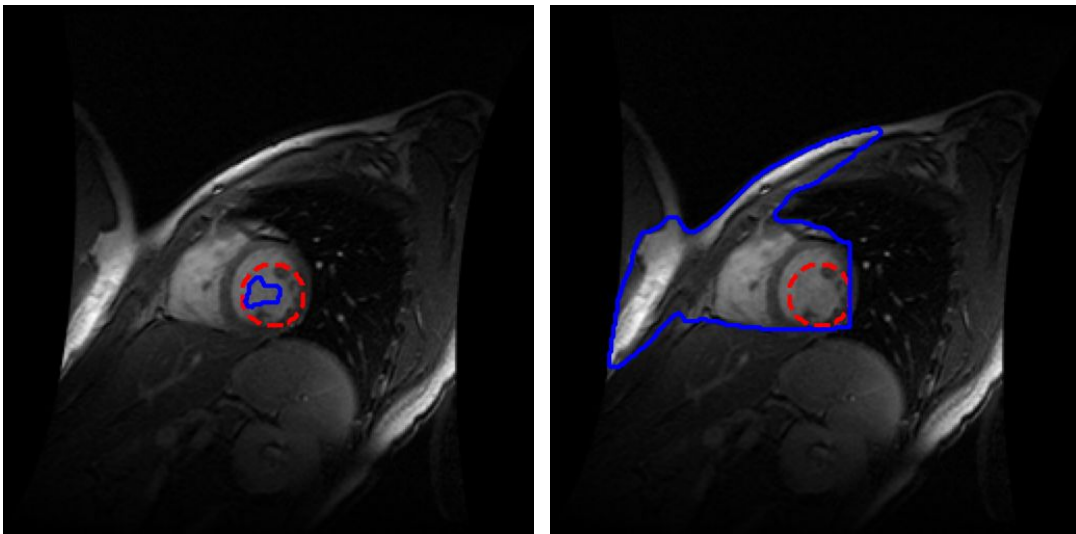
On ne montrera pas la comparaison des résultats pour tous les paramètres parce qu'il y a certains qui influencent seulement dans la convergence de l'algorithme. La première variation présentée est du paramètre d'initialisation, si on prend une grille de cercles au lieu d'un seul, la segmentation va être beaucoup plus grande. De cette façon on peut segmenter à partir des niveaux de gris sans avoir une connaissance à priori de la localisation de l'objet d'intérêt, par contre si l'objet est de la même couleur que d'autres parties de l'image on perd le contrôle précis pour ne segmenter que lui.



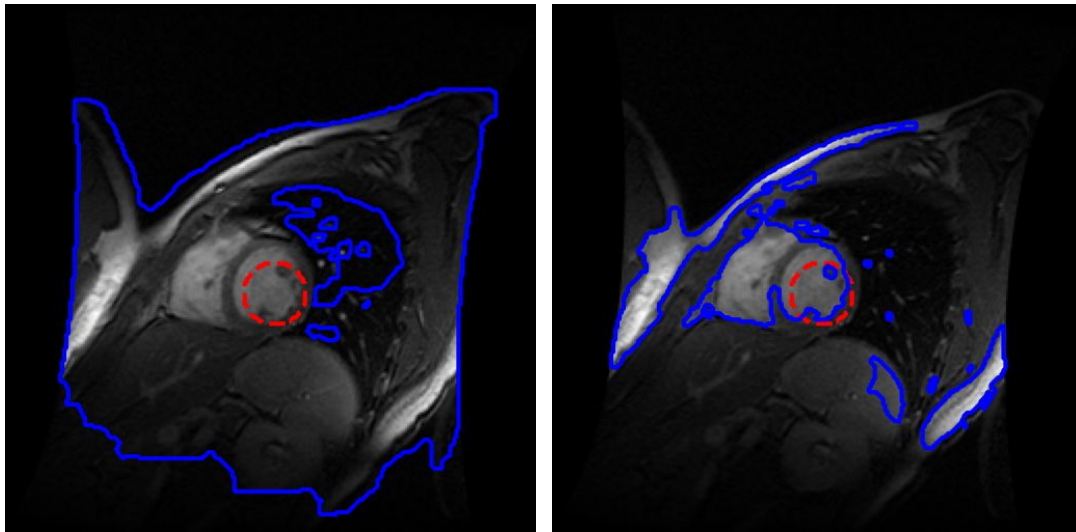
Par rapport au paramètre μ , si on ne pénalise pas grandement la taille du contour (image à gauche avec μ égale à .1), le résultat va avoir beaucoup de zones détachées avec niveaux de gris similaires, mais quand on l'augmente (image à droite avec μ égale à .9), le périmètre du contour est plus petit et il défavorise l'existence de plusieurs groupes.



En regardant l'influence de λ_1 , on a pris une valeur de 100 pour l'image de la gauche et 1 pour l'autre. On voit que la segmentation avec une valeur très grande de λ_1 par rapport à λ_2 , qui est égale à 1, est beaucoup plus petite.



Ici on voit le phénomène inverse qu'avant, parce que quand on augmente la valeur de λ_2 , la zone segmentée est plus grande. Les images ci-dessous montrent cette observation, celle de la gauche a un λ_2 égal à 100 et celle de la droite égal à 10 (λ_1 est égal à 1).



2 Segmentation

2.1 Segmentation de l'IRM 1 du cerveau

Pour la première image du cerveau en IRM (brain.bmp), nous proposons d'essayer de segmenter les *ventriculus lateralis*, qui sont les deux plus gros ventricules du cerveau et contiennent du liquide céphalo-rachidien (LCR). Ils correspondent au corps blanc approximativement au centre de l'image IRM.

Comme nous l'avons vu dans le cours, un exemple de problèmes possibles dans la méthode de contour actif se produit lorsque l'objet a une forte concavité (la forme ressemble à un haricot), dans laquelle la méthode est incapable de segmenter correctement la concavité. Dans notre cas, l'objet que nous voulons segmenter présente des concavités verticalement et horizontalement. Pour cette raison, nous ne nous attendions pas à un bon résultat.

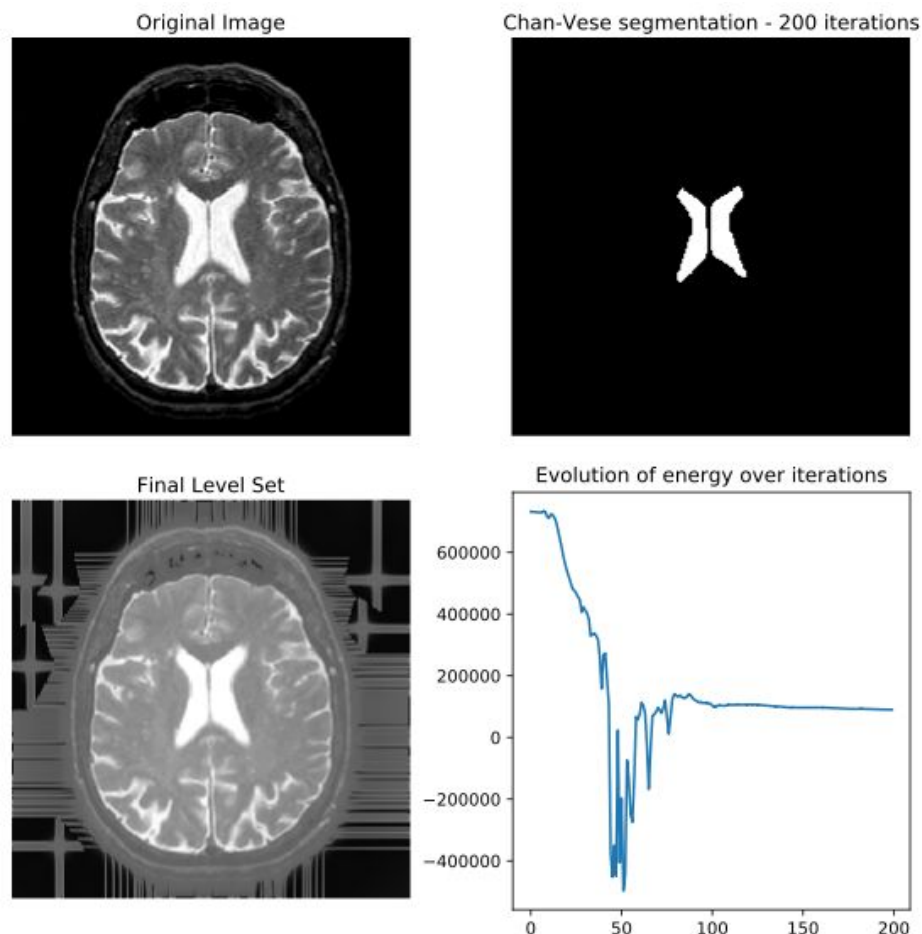
On a effectué les tests pour cette méthode en essayant différentes combinaisons de paramètres (par exemple, réduire les paramètres qui définissent une solution plus lisse), mais aucun n'a obtenu de résultats acceptables.

Pour la Méthode de Chan et Vese nous avons choisi de n'utiliser qu'un seul cercle, car nous savons à peu près que l'objet d'intérêt était au centre de l'image. On s'attend donc à un meilleur résultat qu'avec une grille.

Nous avons choisi un μ élevé, égal à .9, car nous ne voulions pas trouver des zones détachées (la matière grise et le *ventriculus lateralis* ont des niveaux de gris similaires dans cette image), et enfin, nous avons choisi λ 1 un élevé (égal à 100), qui a été choisi par des tests, la valeur qui nous a donné le résultat le plus proche de l'attendu.

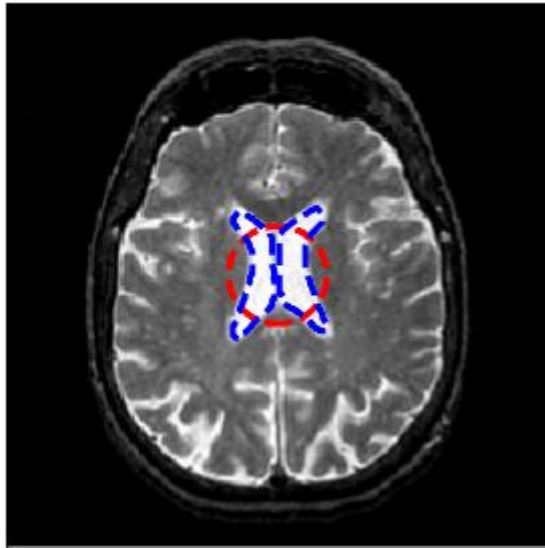
Nous pouvons voir ci-dessous les résultats obtenus pour cet algorithme. Comme décrit dans la figure, nous avons l'image originale ; le résultat obtenu pour la segmentation ; le niveau d'énergie final et le graphique d'énergie pour chaque

interaction. Lors des premières interactions, nous avons une grande variation d'énergie, ce qui explique le nombre élevé d'interactions de l'algorithme. Les paramètres utilisés pour obtenir ce résultat étaient ceux qui donnaient visuellement les meilleurs résultats, mais il est possible qu'il y ait une meilleure combinaison, qui ne compromette pas le résultat et a une convergence plus rapide.



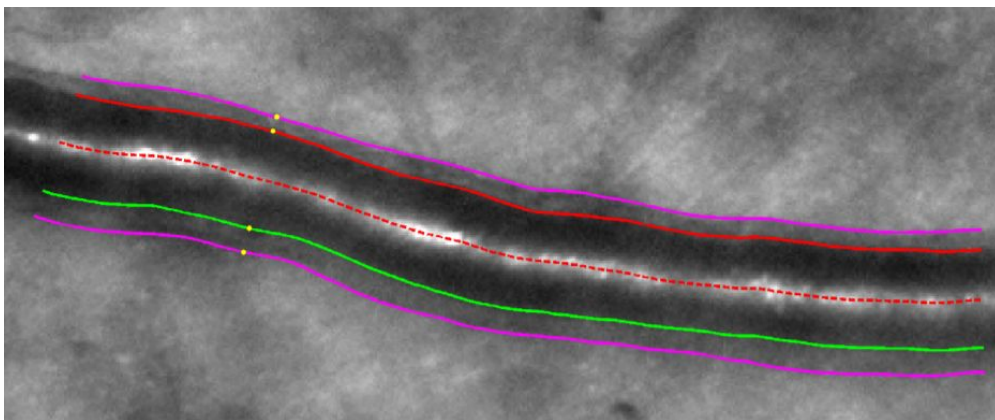
On peut voir ci-dessous la ligne rouge correspondant au cercle utilisé lors de l'initialisation et en bleu, la zone trouvée par l'algorithme après 200 interactions (limite des interactions définie par paramètre de fonction).

Nous considérons le résultat obtenu satisfaisant, étant possible de segmenter la zone souhaitée avec précision et sans détections parasites. Il est à noter qu'avec cette méthode, nous obtenons le contour séparément des ventricules gauche et droit, ce qui peut être utile et efficace pour une application médicale réelle.



2.2 Segmentation des vaisseaux sanguins de la rétine

Comme vu en cours, nous cherchons à segmenter le vaisseau rétinien tout en créant une différenciation entre sa partie la plus interne, qui a principalement un niveau de gris plus clair, sans prendre de valeurs aberrantes qui ont des niveaux proches de son niveau:

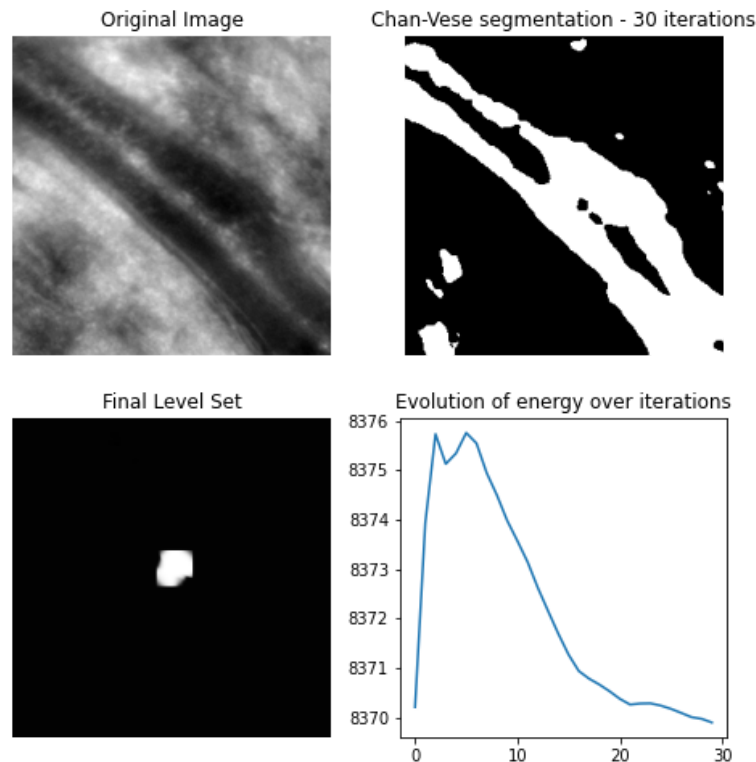


On a effectué les tests pour cette méthode en essayant différentes combinaisons de paramètres, comme le μ et λ , et aussi le type de initiation (plusieurs cercles et damier), mais aucun n'a obtenu de résultats acceptables.

Pour le Méthode de Chan et Vese nous avons choisi d'utiliser un seul cercle de taille 15 au centre de l'image, car nous savons à peu près que l'objet d'intérêt était au centre de l'image, mais la disposition de lui est plus diagonale que centralisée. On s'attend donc à un meilleur résultat qu'avec une grille.

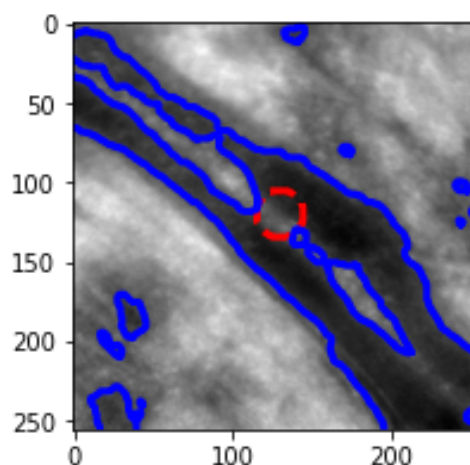
Nous avons choisi un μ bas, égal à .25, car nous voulions trouver des zones détachées (Le filament diagonal de la retina), et enfin, nous avons choisi λ 1 plus petit (5), mais plus haute que λ 2 (égal à 1), qui a été choisi par des tests, la valeur qui nous a donné le résultat le plus proche de l'attendu.

Lors des premières interactions, nous avons une petite variation d'énergie pendant la descente de convergence, ce qui explique le petit nombre d'interactions de l'algorithme.



On peut voir ci-dessous la ligne rouge correspondant au cercle de taille 15 utilisé lors de l'initialisation et en bleu, la zone trouvée par l'algorithme après 30 interactions (moins que la limite des interactions définie par paramètre de fonction).

Nous considérons le résultat obtenu satisfaisant, étant possible de segmenter la zone souhaitée avec précision et de manière efficace, mais avec un peu de détections parasites autour de l'area d'intérêt. Il est à noter qu'avec cette méthode, nous obtenons le contour des vaisseaux sanguins sans les substances plus centrales dans les vaisseaux et un peu de contours parasites qui ont le même niveau de gris que la partie d'intérêt.



2.3 Segmentation de l'IRM 2 du cerveau

Pour la deuxième image du cerveau en IRM (brain2.bmp), nous proposons d'essayer de segmenter une région dense, avec le cortex insulaire, le thalamus, le noyau lenticulaire et le noyau caudé. Elle est la région à gauche du centre d'image, avec une forme qui semble une ellipse.

On a choisi un μ petit, égal à 0.3. Bien que il y a quelques parties du contour avec haut contrast, il y a aussi parties du contour avec un contrast très faible. Si nous avons choisis un μ très grand, le contour serait trop grand, de sorte qu'il touche le lobe temporal et le lobe frontal.

Comme le contour comporte à la fois des régions avec des arêtes et sans arêtes, ainsi que des régions claires et sombres, il n'a pas été possible de trouver une bonne combinaison de paramètres pour la méthode de contour actif.

Après de nombreuses tentatives, les meilleurs paramètres sont seulement un cercle d'initiation (en rouge dans l'image ci-dessous) avec λ_1 égal à 65 et λ_2 égal à 2.3. On a effectué les tests avec plusieurs cercles mais aucun n'a obtenu de résultats acceptables.

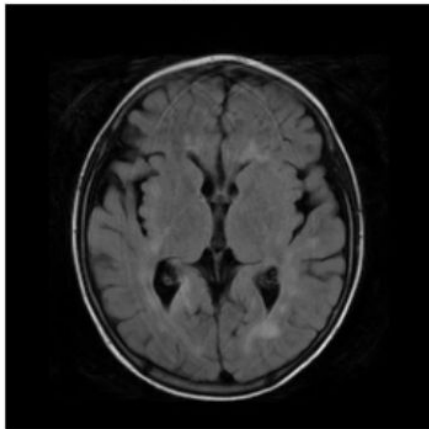
Il faut avoir un équilibre entre λ_1 et λ_2 . Pour un λ_2 grand le contour touche le lobe temporal et le lobe frontal, mais pour un λ_2 petit le contour ne touche pas les arêtes. Pour un λ_1 petit le contour touche le lobe temporal et le lobe frontal aussi, mais pour un λ_2 grand le contour est plus limité. Bien sûr cet équilibre dépend de la valeur du μ aussi.

Pour cette image on a augmenté la quantité d'itérations à 1000, car plusieurs fois après 200 itérations l'image était encore en train de converger.

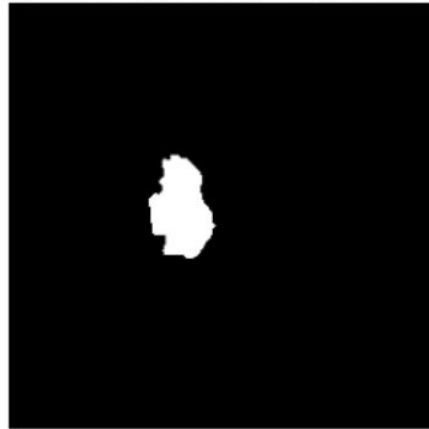
On peut voir ci-dessous la ligne rouge correspondant au cercle de taille 30 utilisé lors de l'initialisation et, en bleu, la zone trouvée par l'algorithme après 1000 itérations. Après 400 itérations l'évolution d'énergie est beaucoup plus petite qu'au début.

Nous considérons le résultat obtenu satisfaisant pour tous les régions à l'exception du cortex insulaire. Il n'a pas été possible de remplir entièrement toutes les arêtes, et le cortex insulaire est dans un arête.

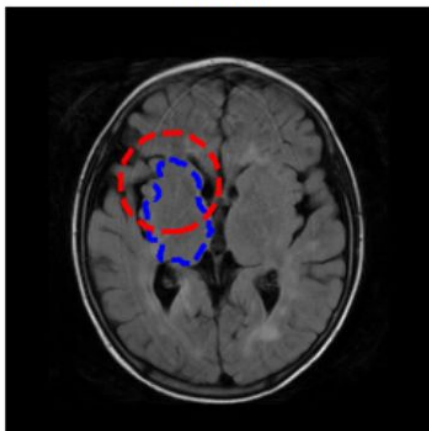
Original Image



Chan-Vese segmentation - 1000 iterations



Contour



Evolution of energy over iterations

