ECE 6560 : Final Project

Geodesic Active Contours

Amandine GOUT

05/2017

Contents

1 Problem statement

In this project, we intend to perform the segmentation of an object in an image. We want to be able to find the contour of this object, partitioning the image in two parts : the object and the rest of the image, or background. The goal of segmentation is to change the representation of an image into something more meaningful and easy to analyze.

In medical imaging for example, the resulting contours from applying segmentation to a stack of images can be used to create 3D reconstruction of an organ for example. However our objective is simpler here as we only want to find the contour of a brain in a medical image. The main difficulty we want to deal with is the presence of surrounding elements from the cranium in the image, which could interfere in the segmentation of the brain only. Indeed, if our method is not parametrized well enough, the contour will finally surround the cranium and no fit the brain precisely.

The method we use in this project is also called Geodesic Active Contours. This method consists in starting from a standard shape like a circle and make it evolve until it fits the boundaries of the desired object.

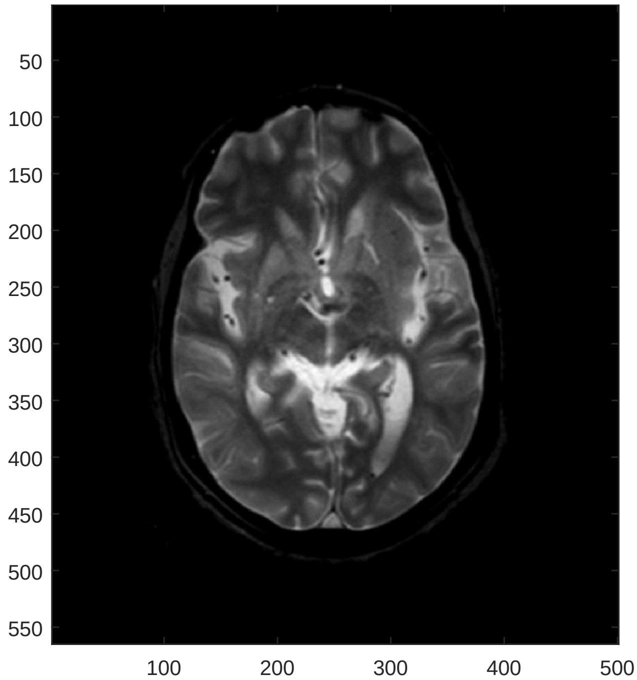

Figure 1: Biomedical image of a brain

2 Mathematical concepts

Geodesic Active Contours is an edge-based method. Indeed, the strength of the edges we are interested in should guide the evolution of the contour across time. We can formulate an energy function which minimization will lead to find the appropriate contour. As we will be using a gradient descent, we need the energy to be high when the contour is far from the edges and low when the contour fits the edges. Thus we can formulate the energy as the integral of the following potential function Φ, C representing the curve, s the arclength parameter, and Φ the potential function :

$$
\begin{aligned}
E(c) &= \int_C \Phi \, \mathrm{d}s \\
\Phi &= \frac{1}{1+\|\nabla I\|^2}
\end{aligned}
$$

This energy is an edge-based energy function. The potential function is small when the contour (C) is close to edges, corresponding to high gradient values of $\|\nabla I\|$, and it is high in flat areas where $\|\nabla I\| = 0$.

3 Deriving the Partial Differential Equation

3.1 Gradient Descent Flow

We need to compute the Gradient Descent Flow to obtain the curve evolution :

$$C_t = -\nabla_C E$$

To compute the gradient of the energy function on the curve, we compute the directional derivative of the energy in the direction of the perturbation on the curve C_t, as it is known that :

$$\frac{dE(C(t))}{dt} = \langle C_t, \nabla_C E \rangle$$

Taking the derivative of the energy function, we get :

$$\begin{aligned}
\frac{dE(C(t))}{dt} &= \frac{d}{dt}\int_C \Phi(C(s,t))\,\mathrm{d}s \\
&= \frac{d}{dt}\int_0^1 \Phi(C(p,t))\,\|C_p\|\ \mathrm{d}p \ \text{, given that } \|C_p\|\ \mathrm{d}p = \|C_s\|\ \mathrm{d}s \text{ and that } \|C_s\| = 1
\end{aligned}$$

As p and t are independent, we have :

$$\begin{aligned}
\frac{dE(C(t))}{dt} &= \int_0^1 \frac{d}{dt}(\Phi(C(p,t))\,\|C_p\|)\,\mathrm{d}p \\
&= \int_0^1 \nabla\Phi\cdot C_t\,\|C_p\| + \Phi\,\|C_p\|_t\ \mathrm{d}p \\
&= \int_0^1 \nabla\Phi\cdot C_t\,\|C_p\| + \Phi\frac{C_{pt}C_p}{\|C_p\|}\ \mathrm{d}p \\
&= \int_0^1 (\nabla\Phi\cdot C_t + \Phi\frac{C_{tp}C_p}{\|C_p\|^2})\,\|C_p\|\ \mathrm{d}p \\
&= \int_C (\nabla\Phi\cdot C_t + \Phi\frac{C_{ts}C_s}{\|C_s\|^2})\,\mathrm{d}s \\
&= \int_C (\nabla\Phi\cdot C_t + \Phi C_{ts}C_s)\,\mathrm{d}s \\
&= \int_C \nabla\Phi\cdot C_t + (\Phi C_s)C_{ts}\,\mathrm{d}s \\
&= \int_C \nabla\Phi\cdot C_t + (\Phi T)C_{ts}\,\mathrm{d}s \quad \text{given that } C_s = T \text{ as T is the tangential component of C}
\end{aligned}$$

According to integration by parts :

$$\begin{aligned}
\frac{dE(C(t))}{dt} &= \int_C \nabla\Phi\cdot C_t - (\Phi T)_s C_t\,\mathrm{d}s \quad \text{as } [(\Phi T)C_t]_C = 0 \\
&= \int_C \nabla\Phi\cdot C_t - (\Phi T)_s C_t\,\mathrm{d}s \\
&= \int_C \nabla\Phi\cdot C_t - (\Phi_s T + \Phi T_s)C_t\,\mathrm{d}s
\end{aligned}$$

We also know that :

$$\begin{aligned}
T_s &= -KN \quad \text{according to Frenet equation, with N the normal to C, and K its curvature} \\
\Phi_s T &= (\nabla\Phi\cdot T)T
\end{aligned}$$

Consequently :

$$\begin{aligned}
\frac{dE(C(t))}{dt} &= \int_C \nabla\Phi\cdot C_t - ((\nabla\Phi\cdot T)T - \Phi KN)C_t\,\mathrm{d}s \\
&= \int_C C_t\cdot(\nabla\Phi - (\nabla\Phi\cdot T)T + \Phi KN)\,\mathrm{d}s \\
&= \int_C C_t\cdot((\nabla\Phi\cdot N)N + \Phi KN)\,\mathrm{d}s
\end{aligned}$$

We can deduce that :

$$
\begin{aligned}
\frac{dE(C(t))}{dt} &= \langle C_t, (\nabla\Phi \cdot N)N + \Phi K N \rangle \\
\nabla_{\mathbf{C}}\mathbf{E} &= (\nabla\mathbf{\Phi} \cdot \mathbf{N})\mathbf{N} + \mathbf{\Phi}\mathbf{K}\mathbf{N}
\end{aligned}
$$

Gradient Descent Flow :

$$
\begin{aligned}
C_t &= -\nabla_C E \\
\mathbf{C_t} &= -(\nabla\mathbf{\Phi} \cdot \mathbf{N})\mathbf{N} - \mathbf{\Phi}\mathbf{K}\mathbf{N} = -(\nabla\mathbf{\Phi} \cdot \mathbf{N} + \mathbf{\Phi}\mathbf{K})\mathbf{N}
\end{aligned}
$$

This equation defines the curve evolution for the Geodesic Active Contour. First, the term $-\Phi K N$ deals with smoothing and shrinking the curve. When close to the boundary, the term $-(\nabla\Phi \cdot N)N$ deals with dragging the curve to smaller values of Φ, in the normal direction of Φ, to find the optimal edge location.

3.2 Level Set Framework

As we are going to use the level set method, we need to derive the level set flow :

$$
\begin{aligned}
\frac{d}{dt}(\Psi(C(p,t),t) &= 0 \\
\nabla\Psi \cdot C_t + \Psi_t &= 0 \\
\Psi_t &= -\nabla\Psi \cdot C_t \\
\Psi_t &= (\nabla\Phi \cdot N + \Phi K)\nabla\Psi \cdot N \text{ and } N = \frac{\nabla\Psi}{\|\nabla\Psi\|} \\
\Psi_t &= (\nabla\Phi \cdot \frac{\nabla\Psi}{\|\nabla\Psi\|} + \Phi K)\,\|\nabla\Psi\| \text{ and } K = \nabla \cdot (\frac{\nabla\Psi}{\|\nabla\Psi\|})
\end{aligned}
$$

As a result the level set framework is :

$$
\mathbf{\Psi_t} = \nabla\mathbf{\Phi} \cdot \nabla\mathbf{\Psi} + \mathbf{\Phi}\,\|\nabla\mathbf{\Psi}\| \cdot \nabla \cdot (\frac{\nabla\mathbf{\Psi}}{\|\nabla\mathbf{\Psi}\|})
$$

3.3 Erosion

In order to perform Geodesic Active Contours, the choice is made to initialize the contour outside the object and to perform erosion until the contour of the object is found. The balloon force can be added to the energy function, and it will prefer smaller surfaces to bigger ones. The α coefficient weights the strength of the edges that the contour can overpass during erosion. The energy can now be written as :

$$
E(c) = \int_C \Phi\,\mathrm{d}s + \alpha \int\int_{C_{inside}} \Phi\,\mathrm{d}x\mathrm{d}y
$$

We also know from class results that the gradient of $E(c) = \int\int_R f\,dx\,dy$ is simply $f\,N$. This leads to the following Gradient Descent Flow :

$$
\begin{aligned}
\mathbf{C_t} &= -(\nabla\mathbf{\Phi} \cdot \mathbf{N} + \mathbf{\Phi}\mathbf{K} + \alpha\mathbf{\Phi})\mathbf{N} \\
\mathbf{\Psi_t} &= \nabla\mathbf{\Phi} \cdot \nabla\mathbf{\Psi} + \mathbf{\Phi}\,\|\nabla\mathbf{\Psi}\| \cdot \nabla \cdot (\frac{\nabla\mathbf{\Psi}}{\|\nabla\mathbf{\Psi}\|}) + \alpha\mathbf{\Phi}\,\|\nabla\mathbf{\Psi}\|
\end{aligned}
$$

Here we choose to perform erosion and α will be a positive coefficient. In order to weight the balloon force more accurately, we also choose to weight α with the potential values Φ, which are at a maximum of 1 in flat and constant areas of the image and which gets smaller around edges. This allows the balloon force to be stronger when further from the edges and smaller around them.

4 Discretization of the Partial Differential Equation

In order to implement the PDE that was previously derived, we need to find the right choice of discretization for the different derivatives.

When discretizing the derivatives, we have to make sure that the chosen discretization leads to stability. This stability can be studied by computing the corresponding CFL (Courant–Friedrichs–Lewy) condition for the chosen scheme with the Von Neumann Stability Analysis. It can either result in showing that the scheme is always unstable, or that it can be stable under a certain condition. Indeed the CFL condition will then give the constraints on the choice of the temporal and spatial discretization steps.

Our PDE contains 3 different terms to study separately :

$$\mathbf{\Psi_t} \quad = \quad \nabla\mathbf{\Phi} \cdot \nabla\mathbf{\Psi} + \mathbf{\Phi} \left\|\nabla\mathbf{\Psi}\right\| \cdot \nabla \cdot (\tfrac{\nabla\mathbf{\Psi}}{\|\nabla\mathbf{\Psi}\|}) + \alpha \left\|\nabla\mathbf{\Psi}\right\|$$

4.1 Diffusion term

The first term to study is the diffusion term leading to the following level set evolution :

$$\begin{aligned}\mathbf{\Psi_t} &= \mathbf{\Phi} \left\|\nabla\mathbf{\Psi}\right\| \cdot \nabla \cdot (\tfrac{\nabla\mathbf{\Psi}}{\|\nabla\mathbf{\Psi}\|}) \\ \Psi_t &= \Phi \tfrac{\Psi_x^2\Psi_{yy} - 2\Psi_x\Psi_y\Psi_{xy} + \Psi_y^2\Psi_{xx}}{\Psi_x^2 + \Psi_y^2}\end{aligned}$$

This 2D non-linear diffusion equation happens to be locally equivalent to a 1D linear diffusion equation. Indeed, when deriving the Geometric Heat Equation, we saw that :

$$\begin{aligned}\Psi_t &= \Phi \tfrac{\Psi_x^2\Psi_{yy} - 2\Psi_x\Psi_y\Psi_{xy} + \Psi_y^2\Psi_{xx}}{\Psi_x^2 + \Psi_y^2} \\ &= \Phi\, \Psi_{\epsilon\epsilon} \\ \text{with} \qquad & \Psi_{\epsilon\epsilon} \ \text{ the diffusion along the edge}\end{aligned}$$

Consequently, the diffusion equation is locally equivalent to a 1D linear diffusion equation in the direction along the edge, which changes at each time t and point in space. Let us derive the CFL condition for a 1D linear diffusion equation with the Von Neumann analysis :

$$\begin{aligned}u_t &= b\, u_{xx} \ \text{ with b the speed of the diffusion equation} \\ \tfrac{u(x,t+\Delta t)-u(x,t)}{\Delta t} &= b\, \tfrac{u(x+\Delta x,t)-2u(x,t)+u(x-\Delta x,t)}{\Delta x^2} \\ u(x,t+\Delta t) &= u(x,t) + \tfrac{b\Delta t}{\Delta x^2}(u(x+\Delta x,t) - 2u(x,t) + u(x-\Delta x,t)) \\ u(x,t+\Delta t) &= (1 - \tfrac{2b\Delta t}{\Delta x^2})u(x,t) + \tfrac{b\Delta t}{\Delta x^2}(u(x+\Delta x,t) + u(x-\Delta x,t))\end{aligned}$$

Using the Discrete Fourier Transform :

$$\begin{aligned}U(w,t+\Delta t) &= (1 - \tfrac{2b\Delta t}{\Delta x^2})U(w,t) + \tfrac{b\Delta t}{\Delta x^2}(U(w,t)e^{jw\Delta x} + U(w,t)e^{-jw\Delta x}) \\ U(w,t+\Delta t) &= (1 - \tfrac{2b\Delta t}{\Delta x^2} + \tfrac{2b\Delta t}{\Delta x^2}(\tfrac{e^{jw\Delta x}+e^{-jw\Delta x}}{2}))U(w,t) \\ U(w,t+\Delta t) &= (1 - \tfrac{2b\Delta t}{\Delta x^2} + \tfrac{2b\Delta t}{\Delta x^2}cos(w\Delta x))U(w,t) \\ U(w,t+\Delta t) &= \alpha(w)U(w,t) \\ \text{with } \alpha(w) &= 1 - \tfrac{2b\Delta t}{\Delta x^2}(1 - cos(w\Delta x)) \ \text{ the amplification factor}\end{aligned}$$

The scheme is stable if $|\alpha(w)| \leq 1$

First it can be noticed that $\forall w \;\; 1 - cos(w\Delta x) \geq 0$, therefore $\alpha(w) \leq 1$ is always true.

On the other side of the inequality we have :

$$\begin{aligned}
-1 &\leq \alpha(w) \\
-1 &\leq 1 - \frac{2b\Delta t}{\Delta x^2}(1 - cos(w\Delta x)) \\
\frac{b\Delta t}{\Delta x^2}(1 - cos(w\Delta x)) &\leq 1 \\
\text{In the worst case, } 1 - cos(w\Delta x) &= 2 \\
\text{Leading to } \frac{2b\Delta t}{\Delta x^2} &\leq 1 \\
\textbf{CFL condition for diffusion in 1D : } \quad \mathbf{b\,\Delta t} &\leq \mathbf{\frac{\Delta x^2}{2}}
\end{aligned}$$

In our case, we choose the same value for Δx and Δy, called q. As the speed depends on the Φ coefficients, we would have :

$$\begin{aligned}
max(|\Phi|)\,\Delta t &\leq \frac{q^2}{2} \text{ and } max(|\Phi|) = 1 \text{ given the potential function} \\
\textbf{CFL condition for the diffusion term : } \quad \mathbf{\Delta t} &\leq \mathbf{\frac{q^2}{2}}
\end{aligned}$$

4.2 Advection term

The second term to study is the advection term leading to the following equation :

$$\begin{aligned}
\mathbf{\Psi_t} &= \mathbf{\nabla\Phi \cdot \nabla\Psi} \\
\Psi_t &= \Phi_x \cdot \Psi_x + \Phi_y \cdot \Psi_y
\end{aligned}$$

This is a 2D linear Transport PDE, which can be discretized with an upwind scheme based on the Von Neumann analysis on 1D Transport PDEs :

In the case of a 1D transport PDE :

$$ut = b\,u_x \text{ with the speed b}$$

If b is positive, we use a Forward Time Forward Space discretization and if b is negative, we use a Forward Time Backward Space discretization, which ensures the stability of the PDE. This scheme is obtained by applying the Von Neumann analysis with both discretizations :

Forward Time Forward Space

$$\begin{aligned}
u_t &= b\,u_x \\
\frac{u(x,t+\Delta t) - u(x,t)}{\Delta t} &= b\,\frac{u(x+\Delta x,t) - u(x,t)}{\Delta x} \\
u(x,t+\Delta t) &= u(x,t) + \frac{b\Delta t}{\Delta x}(u(x+\Delta x,t) - u(x,t)) \\
u(x,t+\Delta t) &= (1 - \frac{b\Delta t}{\Delta x})u(x,t) + \frac{b\Delta t}{\Delta x}u(x+\Delta x,t)
\end{aligned}$$

Using the Discrete Fourier Transform :

$$\begin{aligned}
U(w,t+\Delta t) &= (1-\tfrac{b\Delta t}{\Delta x}+\tfrac{b\Delta t}{\Delta x}e^{jw\Delta x})U(w,t)\\
U(w,t+\Delta t) &= \alpha(w)U(w,t)\\
\text{with }\ \alpha(w) &= 1-\tfrac{b\Delta t}{\Delta x}+\tfrac{b\Delta t}{\Delta x}e^{jw\Delta x}\ \text{ , the amplification factor}\\
&= 1-\tfrac{b\Delta t}{\Delta x}+\tfrac{b\Delta t}{\Delta x}cos(w\Delta x)+j\tfrac{b\Delta t}{\Delta x}\,sin(w\Delta x)\\
&= 1-\tfrac{b\Delta t}{\Delta x}(1-cos(w\Delta x))+j\tfrac{b\Delta t}{\Delta x}\,sin(w\Delta x)
\end{aligned}$$

The scheme is stable if $|\alpha(w)|^2 \leq 1$

$$\begin{aligned}
|\alpha(w)|^2 &\leq 1\\
(1-\tfrac{b\Delta t}{\Delta x}+\tfrac{b\Delta t}{\Delta x}cos(w\Delta x))^2+(\tfrac{b\Delta t}{\Delta x}\,sin(w\Delta x))^2 &\leq 1\\
1-\tfrac{2b\Delta t}{\Delta x}(1-cos(w\Delta x))+(1-cos(w\Delta x))^2(\tfrac{b\Delta t}{\Delta x})^2+(\tfrac{b\Delta t}{\Delta x})^2 sin(w\Delta x))^2 &\leq 1\\
-\tfrac{2b\Delta t}{\Delta x}(1-cos(w\Delta x))+(1-cos(w\Delta x))^2(\tfrac{b\Delta t}{\Delta x})^2+(\tfrac{b\Delta t}{\Delta x})^2 sin(w\Delta x))^2 &\leq 0\\
-\tfrac{2b\Delta t}{\Delta x}(1-cos(w\Delta x))+(1-2cos(w\Delta x))(\tfrac{b\Delta t}{\Delta x})^2+(\tfrac{b\Delta t}{\Delta x})^2 &\leq 0
\end{aligned}$$

if $b<0$:

$$\begin{aligned}
-(1-cos(w\Delta x))+(1-cos(w\Delta x))(\tfrac{b\Delta t}{\Delta x}) &\geq 0\\
(1-cos(w\Delta x))(\tfrac{b\Delta t}{\Delta x}-1) &\geq 0\ \text{ and } \forall w,\ \ 1-cos(w\Delta x))\geq 0\\
\tfrac{b\Delta t}{\Delta x}-1 &\geq 0\\
\tfrac{b\Delta t}{\Delta x} &\geq 1
\end{aligned}$$

Seen that $b<0$ this is impossible and this is unstable

if $b>0$:

$$\begin{aligned}
-(1-cos(w\Delta x))+(1-cos(w\Delta x))(\tfrac{b\Delta t}{\Delta x}) &\leq 0\\
(1-cos(w\Delta x))(\tfrac{b\Delta t}{\Delta x}-1) &\leq 0\ \text{ and } \forall w,\ \ 1-cos(w\Delta x))\geq 0\\
\tfrac{b\Delta t}{\Delta x}-1 &\leq 0\\
\mathbf{b\Delta t} &\leq \mathbf{\Delta x}\ \ \textbf{CFL condition}
\end{aligned}$$

Seen that $b>0$, this is stable if Δt is small enough and verifies this CFL condition

Backward Time Forward Space

$$
\begin{aligned}
u_t &= b\, u_x \\
\frac{u(x,t+\Delta t)-u(x,t)}{\Delta t} &= b\,\frac{u(x,t)-u(x-\Delta x,t)}{\Delta x} \\
u(x,t+\Delta t) &= u(x,t)+\frac{b\Delta t}{\Delta x}(u(x,t)-u(x-\Delta x,t)) \\
u(x,t+\Delta t) &= (1+\frac{b\Delta t}{\Delta x})u(x,t)-\frac{b\Delta t}{\Delta x}u(x-\Delta x,t)
\end{aligned}
$$

Using the Discrete Fourier Transform :

$$
\begin{aligned}
U(w,t+\Delta t) &= (1+\frac{b\Delta t}{\Delta x}-\frac{b\Delta t}{\Delta x}e^{-jw\Delta x})U(w,t) \\
U(w,t+\Delta t) &= \alpha(w)U(w,t) \\
\text{with } \alpha(w) &= 1+\frac{b\Delta t}{\Delta x}-\frac{b\Delta t}{\Delta x}e^{jw\Delta x} \quad \text{, the amplification factor} \\
&= 1+\frac{b\Delta t}{\Delta x}-\frac{b\Delta t}{\Delta x}cos(w\Delta x)-j\,\frac{b\Delta t}{\Delta x}\,sin(w\Delta x) \\
&= 1+\frac{b\Delta t}{\Delta x}(1-cos(w\Delta x))-j\,\frac{b\Delta t}{\Delta x}\,sin(w\Delta x)
\end{aligned}
$$

The scheme is stable if $|\alpha(w)|^2 \leq 1$

$$
\begin{aligned}
|\alpha(w)|^2 &\leq 1 \\
(1+\frac{b\Delta t}{\Delta x}-\frac{b\Delta t}{\Delta x}cos(w\Delta x))^2+(\frac{b\Delta t}{\Delta x}\,sin(w\Delta x))^2 &\leq 1 \\
1+\frac{2b\Delta t}{\Delta x}(1-cos(w\Delta x))+(1-cos(w\Delta x))^2(\frac{b\Delta t}{\Delta x})^2+(\frac{b\Delta t}{\Delta x})^2 sin(w\Delta x))^2 &\leq 1 \\
\frac{2b\Delta t}{\Delta x}(1-cos(w\Delta x))+(1-cos(w\Delta x))^2(\frac{b\Delta t}{\Delta x})^2+(\frac{b\Delta t}{\Delta x})^2(1-cos(w\Delta x))^2) &\leq 0 \\
\frac{2b\Delta t}{\Delta x}(1-cos(w\Delta x))+2(1-cos(w\Delta x))(\frac{b\Delta t}{\Delta x})^2 &\leq 0 \\
\frac{2b\Delta t}{\Delta x}(1-cos(w\Delta x))(1+\frac{b\Delta t}{\Delta x}) &\leq 0 \ \text{ and } \forall w,\ \ 1-cos(w\Delta x))\geq 0 \\
\frac{2b\Delta t}{\Delta x}(1+\frac{b\Delta t}{\Delta x}) &\leq 0
\end{aligned}
$$

if $b > 0$:

$$
1+\frac{b\Delta t}{\Delta x} \leq 0
$$

Impossible. It is unstable.

if $b < 0$:

$$
\begin{aligned}
1+\frac{b\Delta t}{\Delta x} &\geq 0 \\
\frac{-b\Delta t}{\Delta x} &\leq 1 \\
\mathbf{-b\Delta t} &\leq \mathbf{\Delta x} \ \ \textbf{CFL condition}
\end{aligned}
$$

Seen that $b < 0$, this is stable under the derived CFL condition

More generally, for a 1D Transport PDE, we use a Forwart Time Forward Space discretization when b is postive and a Forward Time Backward Space discretization when b is negative. For both cases, the CFL condition of the 1D transport equation is :

$$\mathbf{|b|\Delta t} \quad \leq \quad \mathbf{\Delta x}$$

Discretization for the advection term in our specific case

As a result, the following scheme will be applied in our case : **if $\Phi_x \geq 0$, we apply Forward Time Forward Space discretization for Ψ_x. If $\Phi_x \leq 0$, we apply Forward Time Backward Space discretization for Ψ_x. The same discretization scheme is applied for the y term $\Phi_y \cdot \Psi_y$. The two derivatives Φ_x and Φ_y will simply be obtained with a Forward Time Central Space derivative as they are not evolving. The time discretization step will also have to follow a CFL condition for the 2D case.**

To obtain this condition, we look at the Von Neumann analysis in the worst 2D case, where :

$$\begin{aligned} \Psi_t \quad &= \quad a\,\Psi_x + b\,\Psi_y \\ \text{with} \qquad & \qquad a = \Phi_x, \ \ b = \Phi_y \\ \text{and} \qquad & \qquad \Delta x = \Delta y = q \end{aligned}$$

The **2D CFL condition** is obtained with a new speed of $\sqrt{a^2+b^2}$ instead of $|b| = \sqrt{b^2}$ in 1D :

$$\sqrt{\mathbf{\Phi_x^2 + \Phi_y^2}}\mathbf{\Delta t} \quad \leq \quad \mathbf{q}$$

4.3 Erosion term

The third term to study is the erosion term leading to the following level set evolution :

$$\begin{aligned} \mathbf{\Psi_t} \quad &= \quad \alpha\mathbf{\Phi}\,\|\nabla\mathbf{\Psi}\| \\ \Psi_t \quad &= \quad \alpha\Phi\sqrt{\Psi_x^2 + \Psi_y^2} \end{aligned}$$

In order to discretize this term we use an entropy upwind scheme in 2D :

$$\Psi_t \quad = \quad \alpha\Phi\sqrt{min^2(D_x^-\Psi,0) + max^2(D_x^+\Psi,0) + min^2(D_y^-\Psi,0) + max^2(D_y^+\Psi,0)} \ \text{ with } \alpha \geq 0$$

This 2D entropy upwind scheme is an extension of the 1D case :

$$\begin{aligned} \Psi_t \quad &= \quad \alpha\Phi|\Psi_x| \\ \Psi_t \quad &= \quad \alpha\Phi\sqrt{min^2(D_x^-\Psi,0) + max^2(D_x^+\Psi,0)} \ \text{ with } \alpha\Phi \geq 0 \end{aligned}$$

The 1D entropy upwind scheme is derived from the equivalent transport PDE depending on the sign of $\alpha\Phi$ and Ψ_x.

If $\alpha\Phi > 0$:

$$\Psi_t = \alpha\Phi|\Psi_x|$$

$$\Psi_t = \alpha\Phi \;\; sign(\Psi_x)\Psi_x$$

If $D_x^+\Psi > 0$ and $D_x^-\Psi > 0$: Use Forwart Time Forward Space
If $D_x^+\Psi < 0$ and $D_x^-\Psi < 0$: Use Forwart Time Backward Space
If $D_x^+\Psi > 0$ and $D_x^-\Psi < 0$: Use 0
If $D_x^+\Psi < 0$ and $D_x^-\Psi > 0$: Use the mean of $D_x^+\Psi$ and $D_x^-\Psi$

If $\alpha\Phi < 0$: Reverse the signs above and min and max in the general formula :

$$\Psi_t = -|\alpha\Phi| \; |\Psi_x|$$

$$\Psi_t = -|\alpha\Phi|\sqrt{max^2(D_x^-\Psi, 0) + min^2(D_x^+\Psi, 0)} \text{ with } \alpha\Phi \leq 0$$

For the CFL condition, we generally obtain the same CFL condition as seen for the Transport PDE :

$$|\alpha\mathbf{\Phi}|\,\mathbf{\Delta t} \leq \mathbf{\Delta x}$$

$$|\alpha||\mathbf{\Phi}|_{\mathbf{max}}\,\mathbf{\Delta t} \leq \mathbf{\Delta x}$$

However, we also need to find the corresponding CFL condition in 2D. In the same way as we derived the CFL condition for the advection term, we are going to consider a new speed for the 2D equation, knowing that $\Delta x = \Delta y = q$. According to the 2D CFL condition for the advection term, the speed becomes is :

$$\sqrt{a^2 + b^2} = \sqrt{(\alpha\Phi_1)^2 + (\alpha\Phi_2)^2}$$

In the worst case :

$$\begin{aligned}
\sqrt{a^2 + b^2} &= \sqrt{(\alpha|\Phi|_{max})^2 + (\alpha|\Phi|_{max})^2} \\
&= |\alpha|\sqrt{|\Phi|_{max}^2 + |\Phi|_{max}^2} \\
&= |\alpha|\sqrt{2|\Phi|_{max}^2} \\
&= |\alpha||\Phi|_{max}\sqrt{2} \\
&= |\alpha|\sqrt{2} \text{ as } |\Phi|_{max} = 1
\end{aligned}$$

The **CFL condition for the 2D entropy upwind scheme** is :

$$|\alpha|\sqrt{\mathbf{2}}\,\mathbf{\Delta t} \leq \mathbf{q}$$

Discretization for the erosion term in our specific case

Here we choose to perform erosion : $\alpha\Phi$ will be a positive coefficient. As Φ is always positive, α is a positive coefficient. As an extension of the 1D case derived above, the following entropy upwind scheme will be applied in our 2D case :

$$\mathbf{\Psi_t} = \alpha\,\mathbf{\Phi}\sqrt{\mathbf{min^2(D_x^-\Psi, 0) + max^2(D_x^+\Psi, 0) + min^2(D_y^-\Psi, 0) + max^2(D_y^+\Psi, 0)}} \text{ with } \alpha\,\Phi \geq 0$$

The time discretization step will also have to follow the **CFL condition** :

$$|\alpha|\sqrt{\mathbf{2}}\,\mathbf{\Delta t} \leq \mathbf{q}$$

5 Redistancing

5.1 Level set implementation

The PDE of the level set flow we have to implement is :

$$\mathbf{\Psi_t} \quad = \quad \nabla\mathbf{\Phi}\cdot\nabla\mathbf{\Psi} + \mathbf{\Phi}\,\|\nabla\mathbf{\Psi}\|\cdot\nabla\cdot(\tfrac{\nabla\mathbf{\Psi}}{\|\nabla\mathbf{\Psi}\|}) + \alpha\,\mathbf{\Phi}\,\|\nabla\mathbf{\Psi}\|$$

Level set implementation requires also implementing what is called Extension and Reinitialization (or Redistancing). Indeed, to move the level set around the interface, corresponding to the 0 level set, we need the potential function Φ in the neighborhood of the interface. If we do not want the level sets to overlap as they would be influenced by different values of Φ, we can compute the Extension of Φ and extend the value on the interface in the normal direction to the level set and therefore avoid overlapping by making all level sets evolve accordingly.

As we run the PDE, we can also observe flat or steep regions develop as the interface moves, resulting in inaccurate computation and contour plotting at those locations. Consequently, we need to make sure that the level set does not become ill-conditioned by keeping it close to a distance function from time to time. This is what we call Redistancing or Reinitialization.

Nevertheless, performing Extension can be avoided if already doing Redistancing. Indeed, by making sure to kepp the level set close to a signed distance function, we also make sure it is not ill-formed with overlapping and we correct the level set at the locations where new interfaces arise by mistake, still conserving the main one around the object. Consequently, it was chosen to only perform Redistancing for this project.

Redistancing is performed by solving the following PDE :

$$\Psi_t \quad = \quad S(\Psi)(1-|\nabla\Psi|) \text{ with } S(\Psi) = \tfrac{\Psi}{\sqrt{\Psi^2+1}}$$

Indeed, we want to reshape the level set function to be close to a signed distance function. The function S gives the sign of Ψ, given that Ψ is negative inside the contour, 0 on the contour (0 level set) and positive outside of the contour. We want Ψ to evolve according to its sign, in order to have it solving the following equation, which solution is a signed distance function :

$$|\nabla\Psi| \quad = \quad 1$$

5.2 Discretization of the Redistancing PDE

$$\Psi_t \quad = \quad S(\Psi)(1-|\nabla\Psi|)$$

We need to find a scheme such that the discretization of the following equation is stable :

$$\Psi_t \quad = \quad -S(\Psi)|\nabla\Psi| = \pm|\nabla\Psi|$$

As previously showed for the erosion term in the level set method discretization, we can use an entropy upwind scheme such that :

$$\Psi_t \quad = \quad -S(\Psi)\sqrt{min^2(D_x^-\Psi,0)+max^2(D_x^+\Psi,0)+min^2(D_y^-\Psi,0)+max^2(D_y^+\Psi,0)} \text{ if } -S(\Psi)\geq 0$$

$$\Psi_t \quad = \quad -S(\Psi)\sqrt{max^2(D_x^-\Psi,0)+min^2(D_x^+\Psi,0)+max^2(D_y^-\Psi,0)+min^2(D_y^+\Psi,0)} \text{ if } -S(\Psi)\leq 0$$

In this case, it was previously proved that the CFL condition is :

$$|\alpha|\sqrt{2}\,\Delta t \quad \leq \quad q \text{ with } q = \Delta x = \Delta y$$

$$|1|\sqrt{2}\,\Delta t \quad \leq \quad q$$

$$\mathbf{\sqrt{2}\,\Delta t} \quad \leq \quad \mathbf{q}$$

6 Experiments

Now we are going to test the derived PDE with the above discretization for a segmentation task. Our test image is a medical image of a brain. The difficulty in this image is the presence of the surrounding cranium which can also be considered as a contour. Our main goal here is to find which parameters will allow us to match best the expected contour of the brain part only, excluding the cranium.

6.1 Initialization

At the beginning of the level set method, the contour is initialized with a circular contour, with the corresponding signed distance function. This circle being outside of the brain, the PDE should perform erosion in order to find the expected contour.

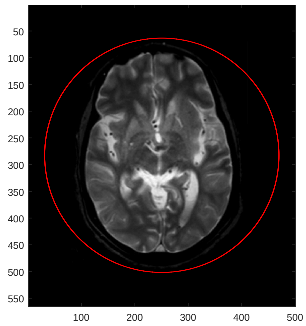

Figure 2: Initialization of the contour : circle surrounding the brain and cranium

6.2 Quality of contour : Error function

In order to measure the quality of the resulting contour, we choose to compute an error function measuring the differences between the true contour and the result from our test. The first step is to manually build a ground truth of the contour (left of Figure 3). We build a mask of the brain by filling the pixels inside the contour in red (center of Figure 3). Finally we threshold this image in order to obtain a binary image where the cranium does not appear anymore, where we have ones inside the brain and zeros outside (right of Figure 3).

Figure 3: Building the ground truth binary mask. On the left, ground truth of the contour. In the center, mask built from filling the inside of the true contour. On the right, binary mask after thresholding the image.

The error of the resulting contour is computed with the following formula :

$$Error \quad = \quad \sum_i \sum_j Mask^{test}(i,j) - Mask^{truth}(i,j)$$

The error function will be positive if the contour is larger than the ground truth. On the other hand, the error function will be negative if the contour is smaller than the ground truth. We will also use a relative error with respect to the surface of the brain. Thus the error function is :

$$Error \quad = \quad \frac{\sum_i \sum_j Mask^{test}(i,j) - Mask^{truth}(i,j)}{\sum_i \sum_j Mask^{truth}(i,j)}$$

On Figure 4, we can observe the evolution of the relative error of an evolving contour with the derived PDE during 4326 iterations. It can be observed that the error first decreases very quickly in flat areas, where the main goal is to shrink the curve. When the contour gets closer to the edges, the error decreases slowly as the contour is locally ajusting itself. The error finally saturates around 0 percent and remains to a steady state, meaning we found the optimal contour.

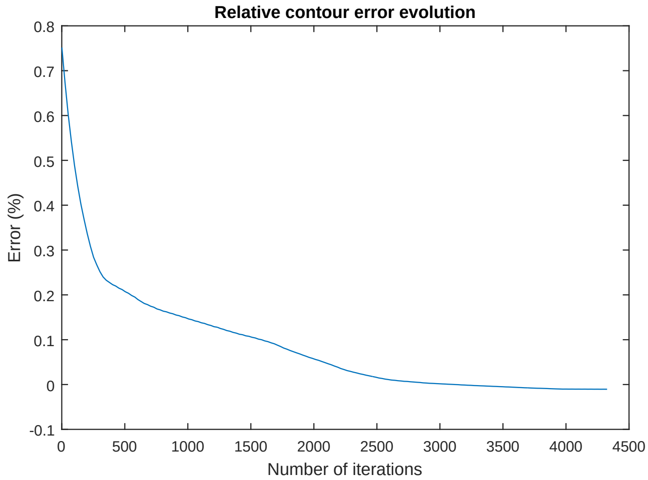

Figure 4: Contour error evolution during 4326 iterations

6.3 Redistancing

When not performing redistancing, the level sets become ill-conditioned around the 0 level set. Indeed, in Figure 5 we can observe the evolution of the contour during the first 250 iterations without redistancing. The level sets around the contour are influenced by local edges and emerge as small additional contours.

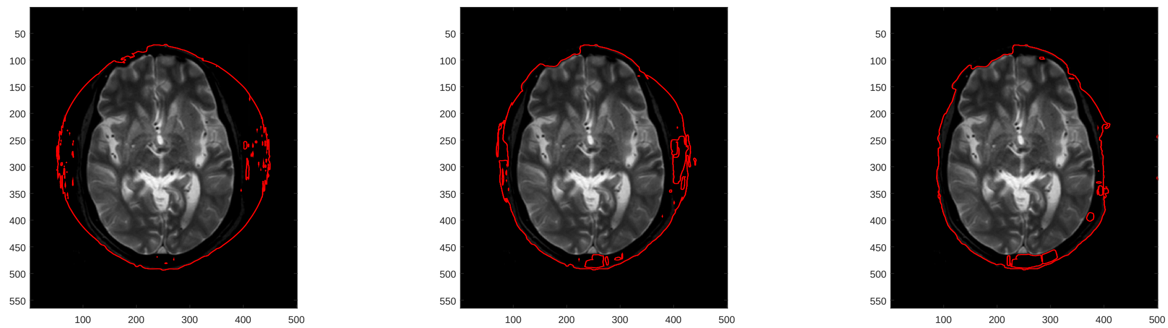

Figure 5: Active contour without redistancing at iterations 100, 175, 250, from left to right

Using redistancing will prevent the levels sets from evolving independently. It will homogenize the evolution of all levels sets accordingly and avoid them to overlap.

6.4 Evolution of the energy function

When updating the level sets according to the PDE, the energy function should be decreasing as the contour fits more and more edges and even stronger ones. In order to verify this property, the energy function was recorded during 4326 iterations (Figure 6). As a reminder, the formula of the energy is :

$$E(c) \quad = \quad \int_C \Phi \, \mathrm{d}s + \alpha \int \int_{C_{inside}} \Phi \, \mathrm{d}x\mathrm{d}y$$

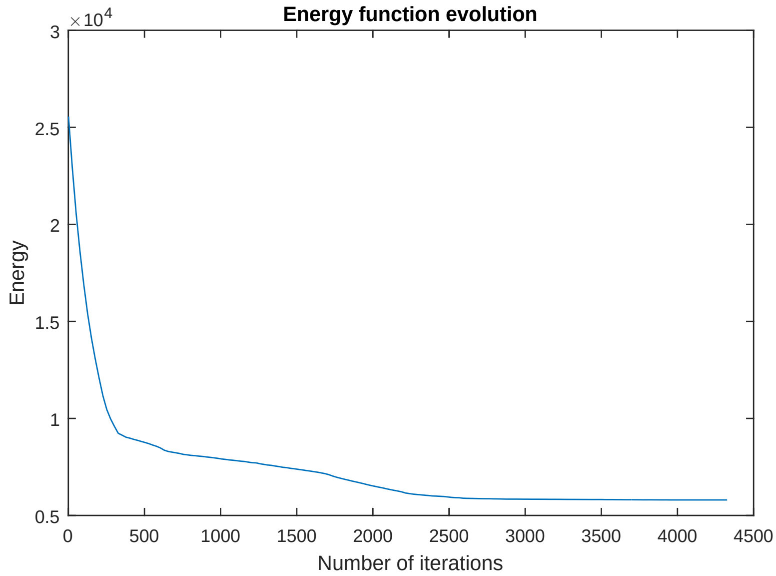

Figure 6: Evolution of the energy function during 4326 iterations with erosion coefficient α set to 0.4

As expected, the energy decreases very strongly during the first iterations while the contour is still in flat areas far from the edges, and decreases more gently when the contour gets closer to edges and adjusts to the shape of the brain. The energy also reaches a steady state after 3000 iterations, similarly to the error term seen in the previous section in Figure 4. Those results demonstrate a consistent behaviour with what could be expected from the implementation of this energy-based PDE.

The detection of the steady state of the energy also defines a stopping criteria for the evolution of the level sets. It can be used in order to have an optimal number of iterations and to stop the processing when the 0 level set stabilizes. This stopping criteria is only valid if the contour stabilizes at the end.

6.5 Erosion

When deriving the level set evolution PDE, we added a balloon force to perform erosion :

$$\mathbf{\Psi_t} \quad = \quad \alpha \mathbf{\Phi} \left\| \nabla \mathbf{\Psi} \right\|$$

Now we need to find an optimal value for the coefficient α to best fit the expected contour of the brain.

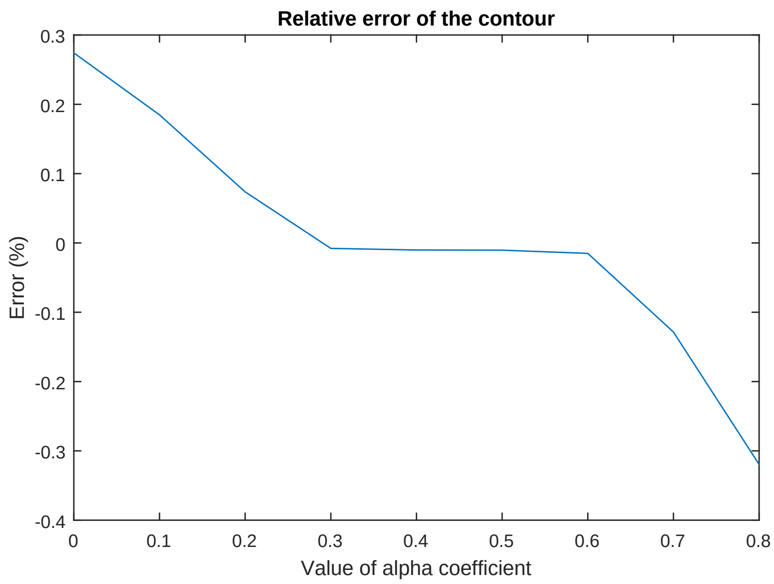

Figure 7: Relative error of the contour for a range of α values from 0.1 to 0.8

The value we choose for α deals with the strength of the edges we want the active contour to be able to overpass. The higher α is, the stronger is the erosion effect. The critical case is when α becomes too strong and the contour even shrinks beyond the boundaries of our object (right of Figure 9). On the other hand, if we do not use any erosion term, setting α to 0, or if α is too small, the contour will get stuck on undesired edges before the contour gets to the actual boundaries of the object (Figure 8).

As Figure 7 shows above, the lower α is, the higher is the relative error in the positive values. Symetrically, the higher α is, the lower is the relative error in the negative values. Thus the higher is the absolute value of the error for high values of α. If we look at the best trade-off between too high and too small values of α, we should choose α in the range [0.3,0.5], where the relative error is close to 0 percent.

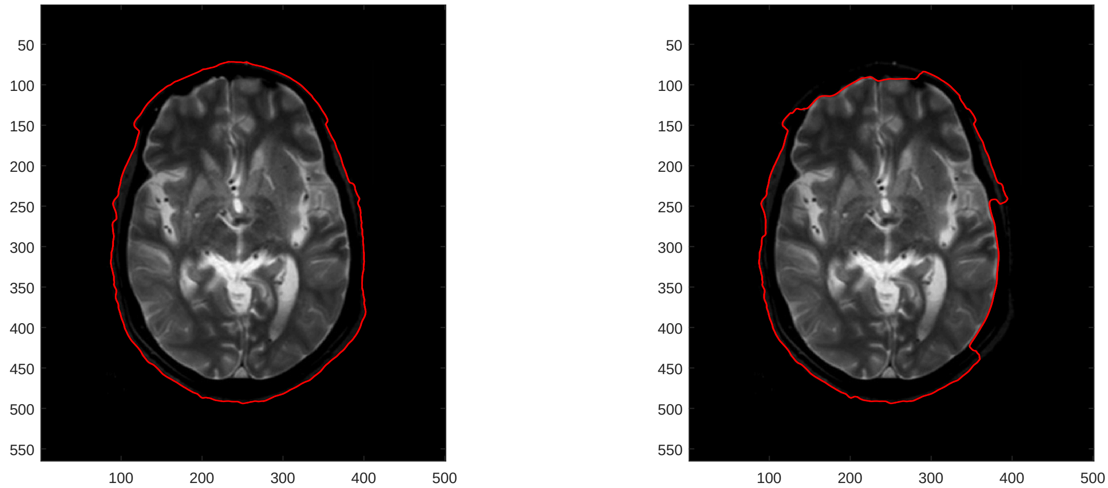

Figure 8: Resulting contour for α =0 (No erosion) and 0.01 from left to right. The corresponding relative error is positive as the contour is larger than the expected contour

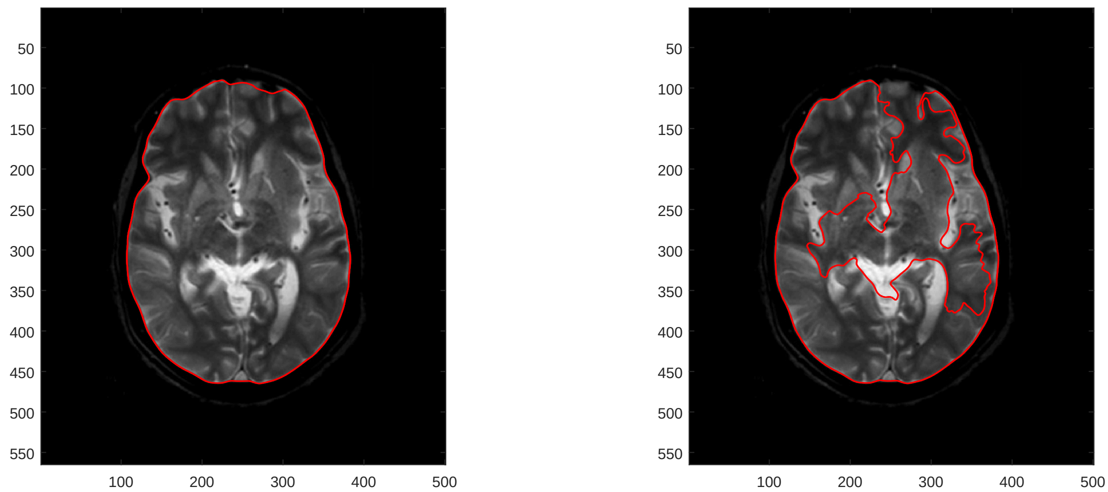

Figure 9: Resulting contour for α =0.04 and 0.08 from left to right. The corresponding relative error is almost nul for α=0.04 as the expected and resulting contours almost match perfectly. However the relative error is negative for α=0.08 as the resulting contour is inside the object boundaries

Despite the importance of the final relative error, we must also find a value of α which leads to a stable contour. This means that we must find a value that leads to a steady state when the contour matches the expected contour. Our 0 level set must converge to the solution. It should not keep on eroding until nothing is left of it. We must also choose α such that we minimize the number of required iterations to get to this steady state.

The number of required iterations was recorded for each α value on Figure 10. Values of α greater than 0.6 are not represented here as the contour does not stabilize in a steady state for those values. The contour keeps on eroding until it completely vanishes. For $\alpha = 0.5$, the relative error (1.416 percent) is slightly worse than for $\alpha = 0.4$ (1.01 percent) as the contour starts eroding indide the brain in the upper right corner. Consequently, the best choices we are left with with respect to the relative error are 0.3 and 0.4. It can also be seen on Figure 10 that choosing $\alpha = 0.3$ required 7500 iterations whereas α=0.4 required 4050 iterations.

As a result, α=0.4 is the most optimal value as it is the best trade-off between the relative error and the number of iterations.

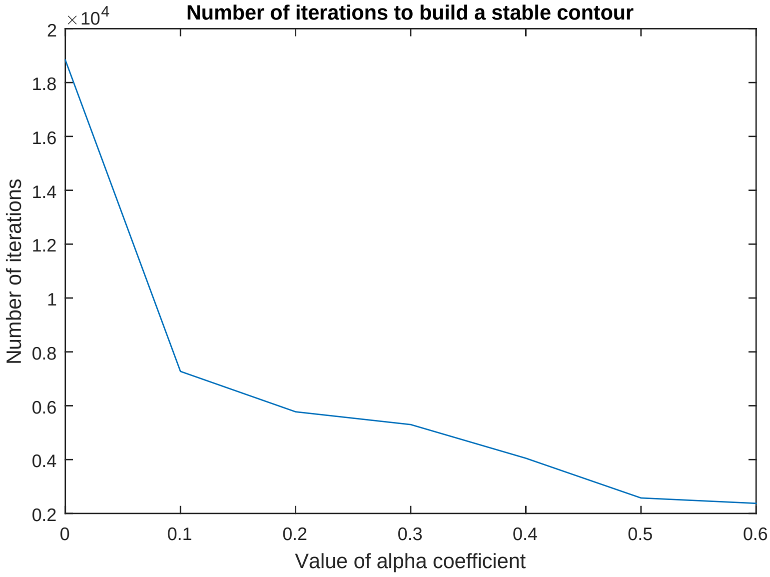

Figure 10: Number of iterations before obtaining a stable contour, for a range of α values from 0.1 to 0.8. The number of iterations decreases as alpha increases because the erosion is faster due to the greater strength of the balloon force.

7 Conclusion

The best result from this project is the contour obtained by using a number of 4326 iterations and a value of 0.4 for the erosion coefficient α.

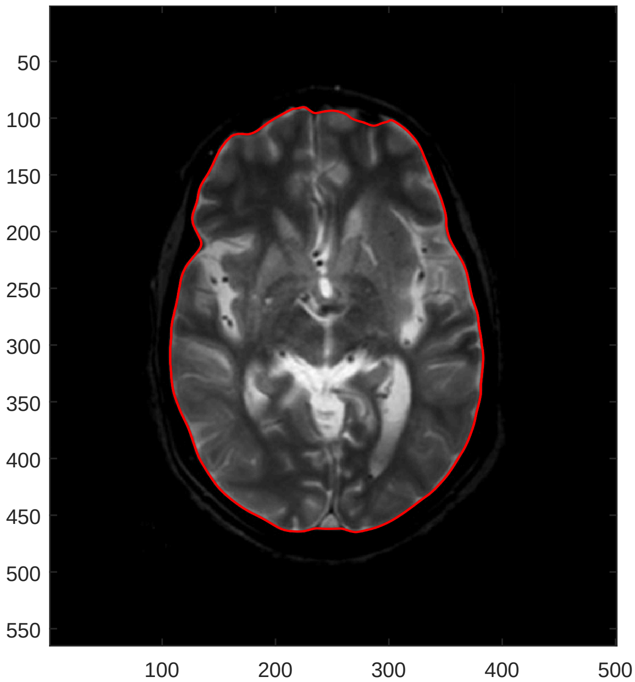

Figure 11: Final contour : $\alpha = 0.4$ and 4326 iterations

Still this contour is not perfect. Due to the smooth aspect of diffusion, the sharpest transitions are not followed properly by the contour. Indeed on Figure 12, the difference between the reference binary mask and the resulting mask is plotted. In black are the regions which were wrongly excluded by the contour and the white areas are the regions the contour took as being part of the object whereas it should have been associated with the background. The biggest missed region is the most complicated one on the top right of the brain. Because of its smooth aspect, the contour did not follow all the sharp transitions. However, the final relative error being of 1 percent, this result is very much satisfying for this project.

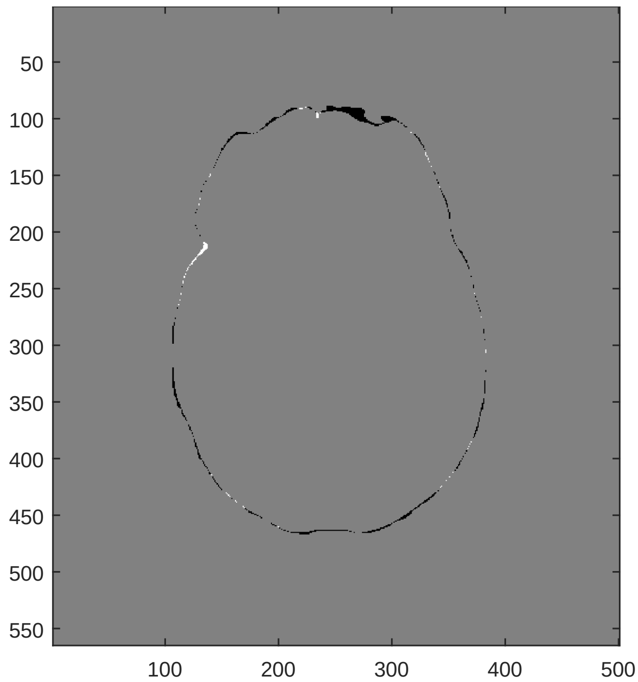

Figure 12: Difference between the reference binary mask and the resulting binary mask from the test

The parametrization of the erosion term was also determining for making the PDE work in an optimal way. Indeed, choosing the best value for α is the result of a trade-off between stability of the contour, speed in terms of number of iterations, and relative error. The instability of the contour was especially hard to detect and there may be more systematic ways to detect it instead of observing the contour vanish. A first way to detect it would be to look at negative errors but the problem is that we obtain small negative errors of 1 percent in the best cases due to smoothing effects. The limit between a negative error due to smoothing and a negative error due to small over-shrinking is very small and very dependent on the image. Using a threshold in this case is not robust at all and we should try to find a more elegant method to solve this issue.

This project especially showed than an erosion term derived from a non constant potential function could be used in Geodesic Active Contours in order to improve the results and robustness regarding secondary contours such as the cranium. It was proved to better deal with the unwanted cranium contour by helping discriminating against weaker edges. The results without erosion clearly showed that the contour did not perform this discrimination and finally stabilized around the cranium. In order to adapt even more to edges strength, we could also try to use another potential function and lead a comparative study between different functions.