Chan-Vese Segmentation in 2D and 3D Medical Images

Project Report

Sourabh Daptardar (108209783)

Ahsanul Karim (109017329)

Abstract- In this report, we review and explain the implementation of the Chan-Vese model for image segmentation in 2D and 3D medical images (for example: brain MRI data from 3D slicer).

Introduction:

Image segmentation is one of the most important and well-studied problems in the domain of image processing. Segmentation is the process of dividing an image into a set of sub-regions meaningfully and used to locate objects and boundaries in images. Image segmentation is not same as edge detection. Image segmentation is a subjective task and its goal is to simplify or change the image into some representation which can make image data easy to analyze.

In medical imaging, image segmentation is very widely used. Some of the applications are:

- Locating tumor
- Tissue volume measurement
- Study various anatomical features for diagnosis of diseases etc.

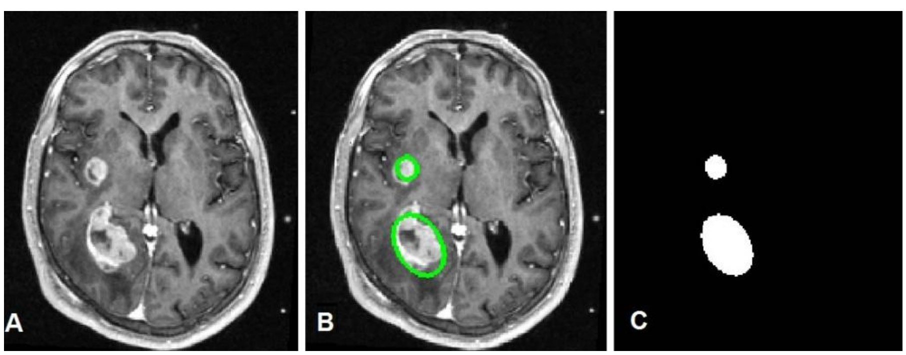

Figure 1 Image Segmentation in Medical Imaging

Therefore, in medical imaging it is very important to get accurate segmentation which can really help the process of medical treatment. In computer vision, there are a number of image segmentation methods. Some of them are:

- Thresholding
- Clustering

- Region Growing
- Graph Partitions
- Machine Learning based approaches
- PDE and Contour based methods

In this current project we have studied and implemented a PDE and active contour based method named Chan-Vese method.

The rest of the report is organized as follows. First we introduce the Chan-Vese model along with the mathematical framework in section 2. Then we explain the level set formulation of the model. In section 3 we provide implementation details. Finally, section 4 shows our results.

2. Chan-Vese Method

Chan-Vese method is a very powerful and flexible method. This method is able to segment many types of images including some that are difficult to segment using classical thresholding or gradient based methods. This is based on Mumford-Shah functional and widely used in medical imaging field (especially for the segmentation of brain, hear and trachea). Unlike other active contour based method, Chan-Vese model does not depend on edge boundary of object.

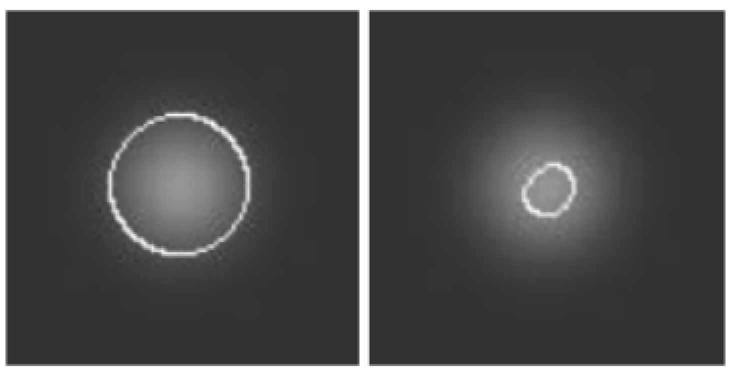

Figure 2 Left: Chan-Vese Method Right: gradient based method

Like other Active contour based algorithms, it begins with a contour in the image plane defining some initial segmentation. The initial contour is evolved according to some evolution equation. The goal is to evolve the contour such that it stops on the boundaries of the foreground region.

The most popular approaches to implement Chan-Vese model include:

- Level Set method
- Graph-cut based method etc

We focus on the Level Set based implementation.

2.1 Model

The objective is to minimize the energy functional $F(c_1, c_2, C)$:

$$F(c_1,c_2,C)=\mu\cdot Length(C)+\nu\cdot Area(inside(C))$$
$$+\lambda_1\int_{inside(C)}|u_0(x,y)-c_1|^2dxdy+\lambda_2\int_{outside(C)}|u_0(x,y)-c_2|^2dxdy$$

Where, $\mu \geq 0$, $\nu \geq 0$, λ_1, $\lambda_2 > 0$ are fixed parameters.

Therefore, the minimization problem is:

$$\inf_{c_1,c_2,C} F(c_1,c_2,C)$$

2.2 Level Set Formulation

Redefining the problem in level set formalism, C is represented by zero level set of some Lipschitz function, $\Phi : \Omega \longrightarrow \mathbf{R}$ such that:

$$\begin{cases} C=\partial\omega=\{(x,y)\in\Omega:\Phi(x,y)=0\} \\ inside(C)=\omega=\{(x,y)\in\Omega:\Phi(x,y)>0\} \\ outside(C)=\Omega\setminus\bar{\omega}=\{(x,y)\in\Omega:\Phi(x,y)<0\} \end{cases}$$

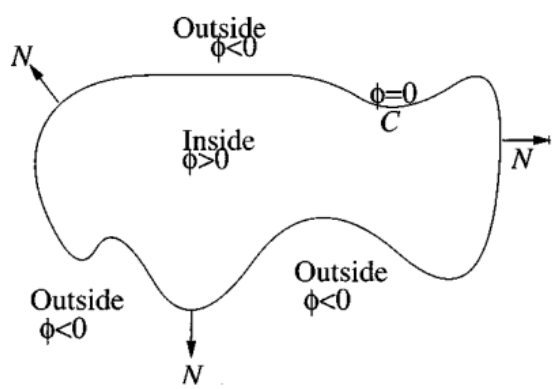

Figure 3 Curve C = {(x,y): , Φ(x,y) = 0} propagating in normal direction

If we want functional $F(c_1, c_2, C)$ in terms of $\Phi(x, y)$:

1. *Length(C)* calculated as length of the zero level set $\Phi(x, y) = 0$:

$$Length(C)=\int_{\Omega}|\nabla H(\Phi(x,y))|dxdy=\int_{\Omega}\delta_0(\Phi(x,y))|\nabla\Phi(x,y)|dxdy$$

Where, $H(z)$ is the Heaviside function:

$$H(z) = \begin{cases} 1, & \text{if } z \geq 0 \\ 0, & \text{if } z < 0 \end{cases}$$

And, Dirac measure:

$$\delta_0(z) = \frac{d}{dz} H(z)$$

2. *Area(inside(C))* calculated as the area of the region where $\Phi(x, y) \geq 0$:

$$Area(inside(C)) = \int_\Omega H(\Phi(x,y))dxdy$$

3. Considering only the region in which $\Phi(x, y) > 0$:

$$\int_{inside(C)} |u_0(x,y) - c_1|^2 dxdy =$$
$$\int_{(x,y):\Phi(x,y)>0} |u_0(x,y) - c_1|^2 dxdy = \int_\Omega |u_0(x,y) - c_1|^2 H(\Phi(x,y))dxdy$$

4. Similarly:

$$\int_{outside(C)} |u_0(x,y) - c_2|^2 dxdy =$$
$$\int_{(x,y):\Phi(x,y)<0} |u_0(x,y) - c_2|^2 dxdy = \int_\Omega |u_0(x,y) - c_2|^2 H(1-\Phi(x,y))dxdy$$

5. The average intensities:

$$c_1 = \frac{\int_\Omega u_0(x,y) H(\Phi(x,y))dxdy}{\int_\Omega H(\Phi(x,y))dxdy},\ c_2 = \frac{\int_\Omega u_0(x,y) H(1-\Phi(x,y))dxdy}{\int_\Omega H(1-\Phi(x,y))dxdy}$$

So, the energy functional becomes:

$$F(c_1, c_2, \Phi) = \mu \int_\Omega \delta_0(\Phi(x,y)) |\nabla\Phi(x,y)| dxdy + \nu \int_\Omega H(\Phi(x,y))dxdy$$
$$+ \lambda_1 \int_\Omega |u_0(x,y) - c_1|^2 H(\Phi(x,y))dxdy + \lambda_2 \int_\Omega |u_0(x,y) - c_2|^2 H(1-\Phi(x,y))dxdy$$

Following region term affects the motion of the curve:

$$-\lambda_1 (u_0 - c_1)^2 + \lambda_2 (u_0 - c_2)^2$$

And the following term is the penalty on the total length of C:

$$\mu \int_{\Omega} \delta_0\left(\Phi(x, y)\right)\left|\nabla \Phi(x, y)\right| dxdy$$

The Euler-Lagrange equation and the gradient-descent method are used to derive equation for the level set function Φ that minimizes fitting energy *F(Φ)*:

$$\begin{aligned}
\frac{\partial \Phi}{\partial t} &= \delta(\Phi)\left(\mu \cdot p \cdot \left(\int_{\Omega} \delta(\Phi)\left|\nabla \Phi\right| \mathrm{dxdy} \right)^{p-1} \operatorname{div}\left(\frac{\nabla \Phi}{\left|\nabla \Phi\right|} \right) - \nu - \lambda_1 \left(u_0 - c_1\right)^2 + \lambda_2 \left(u_0 - c_2\right)^2 \right) \\
&= \delta(\Phi)\left(\mu \cdot p \cdot \left(\int_{\Omega} \delta(\Phi)\left|\nabla \Phi\right| \mathrm{dxdy} \right)^{p-1} \cdot \kappa(\Phi) - \nu - \lambda_1 \left(u_0 - c_1\right)^2 + \lambda_2 \left(u_0 - c_2\right)^2 \right)
\end{aligned}$$

$$\Phi(x, y, 0) = \Phi_o(x, y), \ \frac{\delta(\Phi)}{\left|\nabla \Phi\right|} \frac{\partial \Phi}{\partial \vec{n}} = 0 \text{ on } \partial\Omega$$

Here, κ (*Φ*) is the curvature of evolving curve. For 3D case, we have used Gaussian curvature of surface.

2.3 Numerical Approximation

Analytic approximations of Heaviside and Dirac delta function are given below:

$$H_{\varepsilon}(x) = \frac{1}{2}\left(1 + \frac{2}{\pi} \arctan\left(\frac{x}{\varepsilon}\right)\right)$$

$$\delta_{\varepsilon}(x) = \frac{1}{\pi} \frac{\varepsilon}{\varepsilon^2 + x^2}$$

Analytic approximations of curvature are:

$$Curvature\ K = div\left(\frac{\nabla\phi}{|\nabla\phi|}\right) = \frac{\phi_{xx}{\phi_y}^2 - 2\phi_{xy}\phi_x\phi_y + \phi_{yy}{\phi_x}^2}{(\phi_x + \phi_y)^{3/2}}$$

Where ϕ_x , ϕ_y and ϕ_{xy} are derivatives of phase ϕ.

Here we use forward finite difference and we get:

$$\begin{cases} \phi_x = \dfrac{\phi_{x+\Delta x,y} - \phi_{x,y}}{\Delta x} \\ \phi_y = \dfrac{\phi_{x,y+\Delta y} - \phi_{x,y}}{\Delta y} \\ \phi_{xy} = \dfrac{(\phi_{x+\Delta x,y+\Delta y} - \phi_{x,y+\Delta y}) - (\phi_{x+\Delta x,y} - \phi_{x,y})}{\Delta x \Delta y} \end{cases}$$

Let $h = \Delta x = \Delta y = 1$ be the space steps, then:

$$\begin{cases} \phi_x = \phi_{x+1,y} - \phi_{x,y} \\ \phi_y = \phi_{x,y+1} - \phi_{x,y} \\ \phi_{xy} = \phi_{x+1,y+1} - \phi_{x,y+1} - (\phi_{x+1,y} - \phi_{x,y}) \end{cases}$$

Approximation of Φ is given by:

$$\frac{\phi_{i,j}^{n+1} - \phi_{i,j}^{n}}{\triangle t} = \delta_h(\phi_{i,j}^n)\Big[\frac{\mu}{h^2}(p \cdot L(\phi^n)^{p-1})$$

$$\triangle_-^x\Big(\frac{\triangle_+^x \phi_{i,j}^{n+1}}{\sqrt{(\triangle_+^x \phi_{i,j}^n)^2/(h^2) + (\phi_{i,j+1}^n - \phi_{i,j-1}^n)^2/(2h)^2}}\Big)$$

$$+\frac{\mu}{h^2}(p \cdot L(\phi^n)^{p-1})$$

$$\triangle_-^y\Big(\frac{\triangle_+^y \phi_{i,j}^{n+1}}{\sqrt{(\phi_{i+1,j}^n - \phi_{i-1,j}^n)^2/(2h)^2 + (\triangle_+^y \phi_{i,j}^n)^2/(h^2)}}\Big)$$

$$-\nu - \lambda_1(u_{0,i,j} - c_1(\phi^n))^2 + \lambda_2(u_{0,i,j} - c_2(\phi^n))^2\Big].$$

We have used the following form of the equation for 2D case where *ν=0*:

$$\frac{\partial\varphi}{\partial t} = \delta(\varphi)\left[\frac{\varphi_{xx}{\varphi_y}^2 - 2\varphi_x\varphi_y\varphi_{xy} + \varphi_{yy}{\varphi_x}^2}{\left({\varphi_x}^2 + {\varphi_y}^2\right)^{3/2}} - \lambda_{in}(f - c_{in})^2 + \lambda_{out}(f - c_{out})^2\right]$$

With:

$$\delta_\varepsilon(t) = \frac{\varepsilon}{\pi(\varepsilon^2 + t^2)}$$

$$\kappa(\Phi)=\frac{\Phi_{xx}\Phi_y^2-2\Phi_{xy}\Phi_x\Phi_y+\Phi_{yy}\Phi_x^2}{\left(\Phi_x^2+\Phi_y^2\right)^{3/2}}$$

$\lambda_1 = \lambda_2 = 0.1$ and $\varepsilon = 0.1$

For 3D Slicer data, we have calculated Gaussian Curvature of surface:

$$\kappa(x,y,z) = \{[\Phi_z(\Phi_{xx}\,\Phi_z - 2\,\Phi_x\,\Phi_{xz}) + \Phi_x^2\,\Phi_{zz}][\Phi_z(\Phi_{yy}\,\Phi_z - 2\,\Phi_y\,\Phi_{yz}) + \Phi_y^2\,\Phi_{zz}] - (\Phi_z(-\Phi_x\,\Phi_{yz} + \Phi_{xy}\,\Phi_z - \Phi_{xz}\,\Phi_y) + \Phi_x\,\Phi_y\,\Phi_{zz})^2\}\,[\Phi_z^2(\Phi_x^2 + \Phi_y^2 + \Phi_z^2)^2]^{-1}$$

3. Implementation

We have used Matlab for implementation and done experiments with data from 3D Slicer.

The evolution is calculated as follows:

```
for t = 0:dt:T
    %Approximate derivatives
    F_x = (F([2:m,m],:,:) - F(1:m,:,:))/2;
    F_y = (F(:,[2:n,n],:) - F(:,1:n,:))/2;
    F_z = (F(:,:,[2:p,p]) - F(:,:,1:p))/2;

    F_xx = (F_x([2:m,m],:,:) - F_x(1:m,:,:))/2;
    F_yy = (F_y(:,[2:n,n],:) - F_y(:,1:n,:))/2;
    F_zz = (F_z(:,:,[2:p,p]) - F_z(:,:,1:p))/2;

    F_xy = (F_x(:,[2:n,n],:) - F_x(:,1:n,:))/2;
    F_yz = (F_y(:,:,[2:p,p]) - F_y(:,:,1:p))/2;
    F_xz = (F_x(:,:,[2:p,p]) - F_x(:,:,1:p))/2;

    % Calculating curvature
    Num = (F_z.*(F_xx.*F_z- 2.*F_x.*F_xz) +(F_x.^2).*
F_zz).*(F_z.*(F_yy.*F_z- 2.*F_y.*F_yz) +(F_y.^2) .* F_zz)-((F_z.*(-
1.*F_x.*F_yz- F_xy.*F_z - F_xz.*F_y) +F_x.* F_y.* F_zz).^2);
    Den = F_z.^2.*(F_x.^2+F_y.^2+F_z.^2) + a;

    c_in = sum(sum(sum([F>0].*X)))./(a+sum(sum(sum([F>0]))));
    c_out = sum(sum(sum([F<0].*X)))./(a+sum(sum(sum([F<0]))));

    F = F + dt*epsilon./(pi*(epsilon^2+F.^2)).*( Num./Den  - lambda_in*(X-
c_in).^2 + lambda_out*(X-c_out).^2);

end;
```

4. Results

4.1 Shapes with Smooth Edges

We ran several experiments. First, we tried to see the segmentation on a 3D shape which has smooth edge.

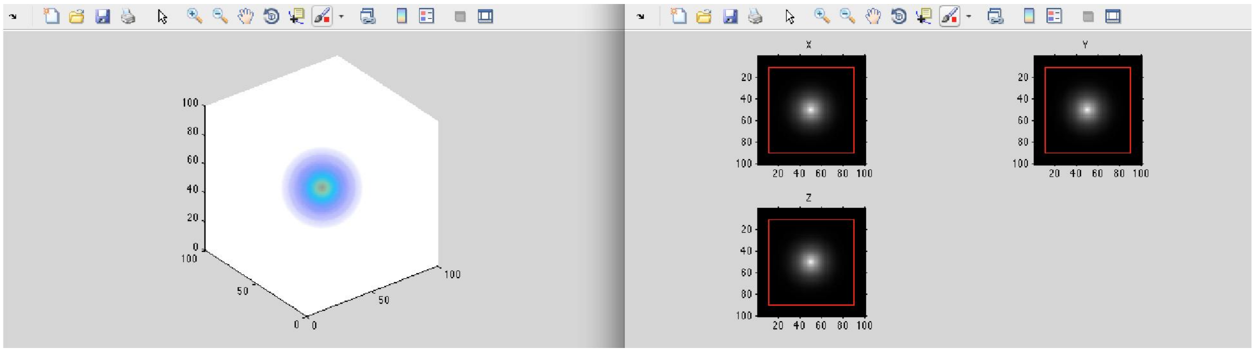

Figure 4 Iteration of Chan-Vese Algorithm on 3D sphere with smooth edge

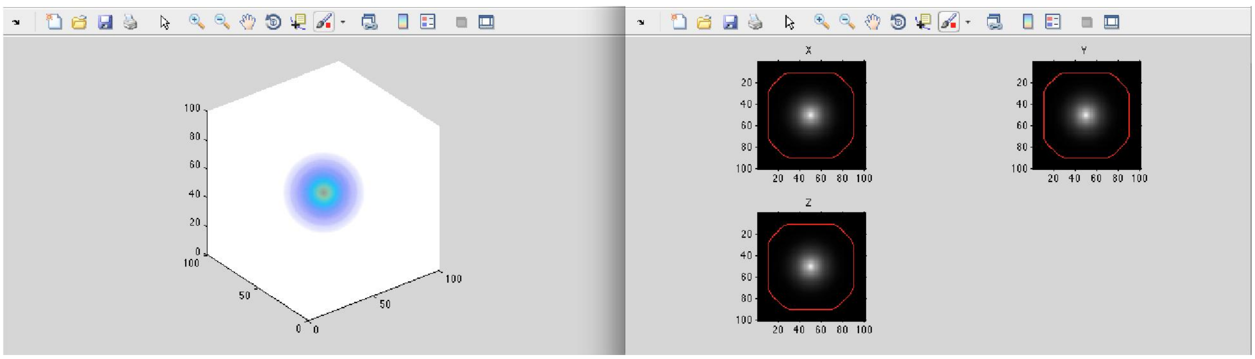

Figure 5 Iteration of Chan-Vese Algorithm on 3D sphere with smooth edge

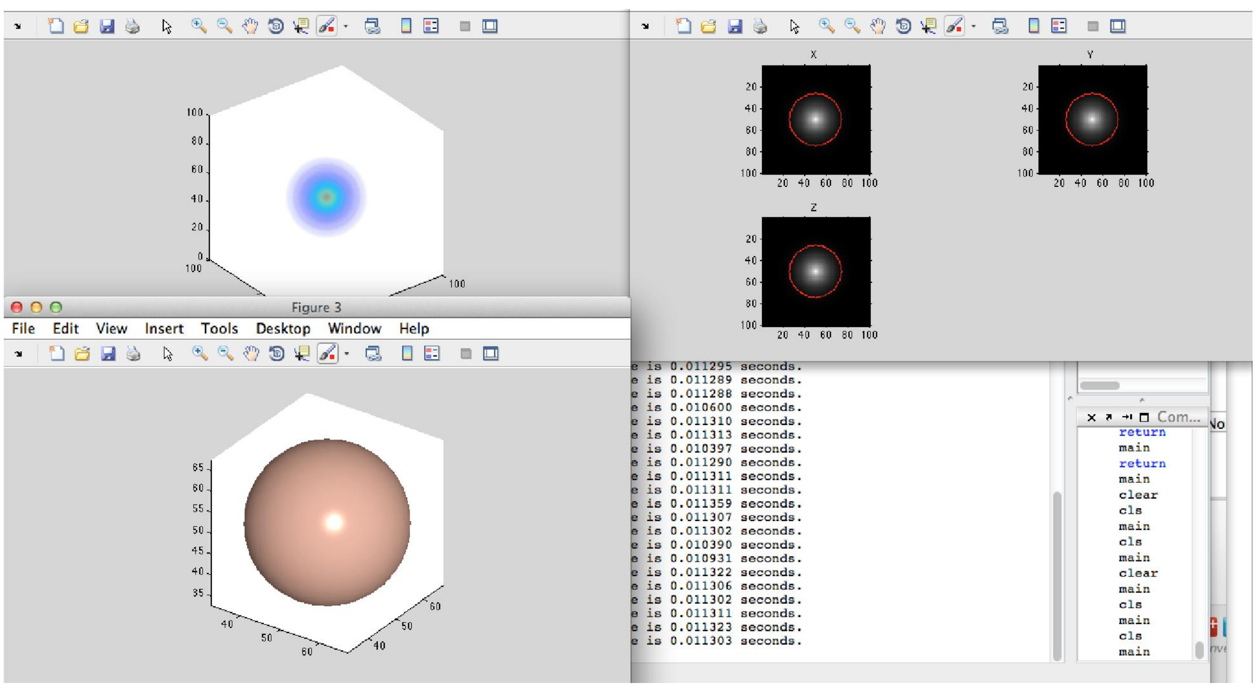

Figure 6 Segmentation Result on 3D sphere with smooth edge

4.2 Image Data from 3D Slicer

We also ran experiments on several image data (.nrrd format) from slicer.

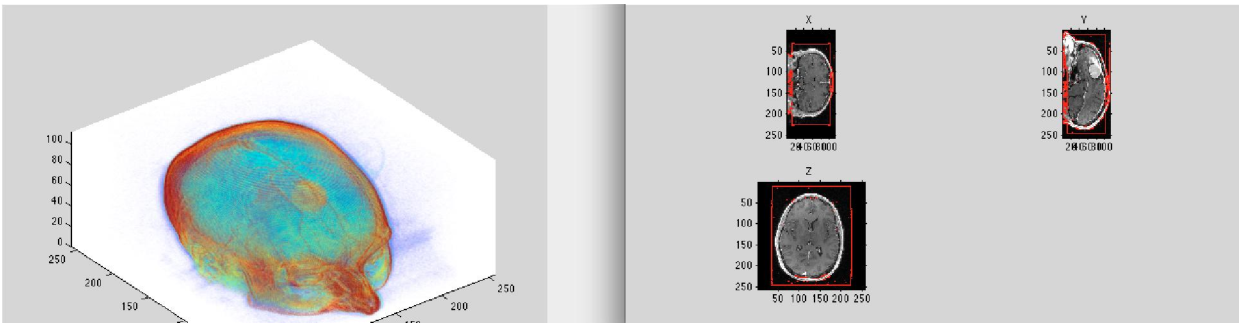

Figure 7 Iteration of Chan-Vese Algorithm on MR Image

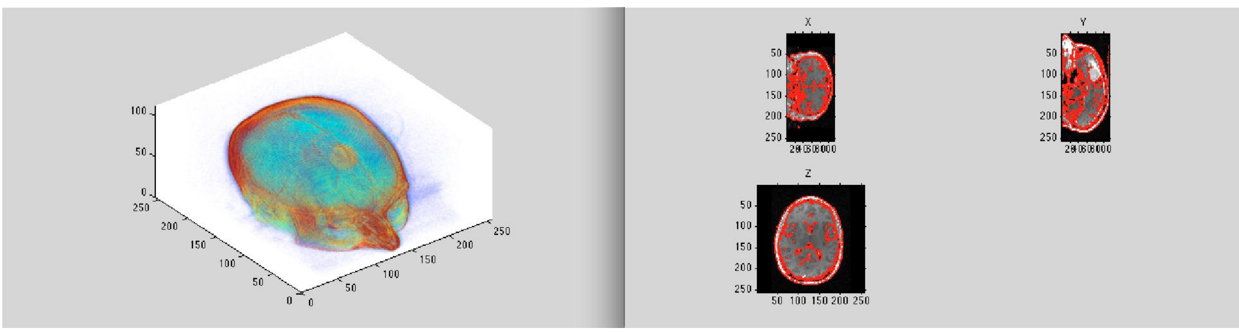

Figure 8Iterations of Chan-Vese Algorithm on MR Image with evolving contour

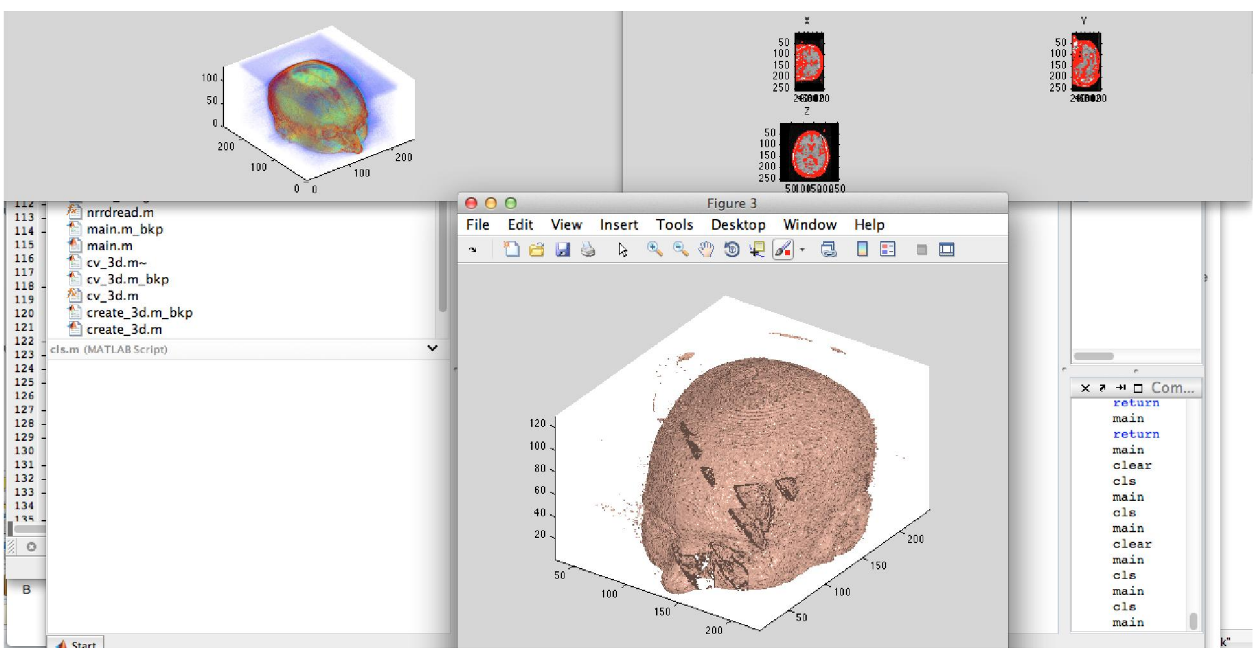

Figure 9 Segmentation result on MR Image

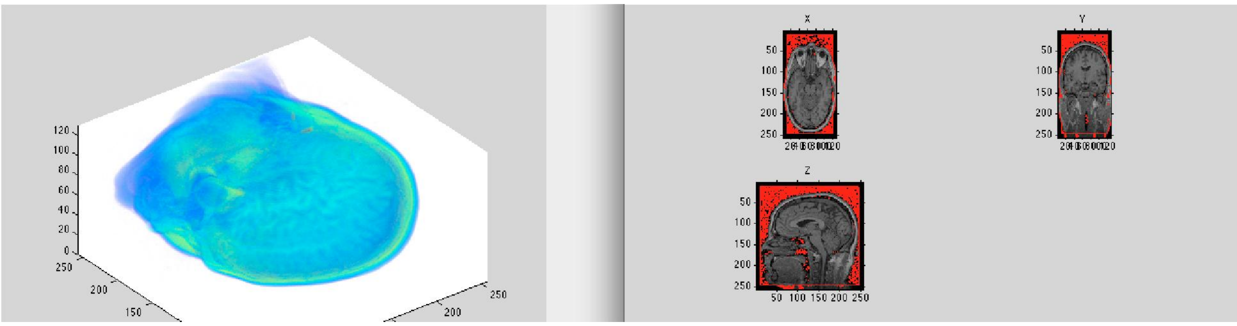

Figure 10Iteration of Chan-Vese Algorithm on MR Image

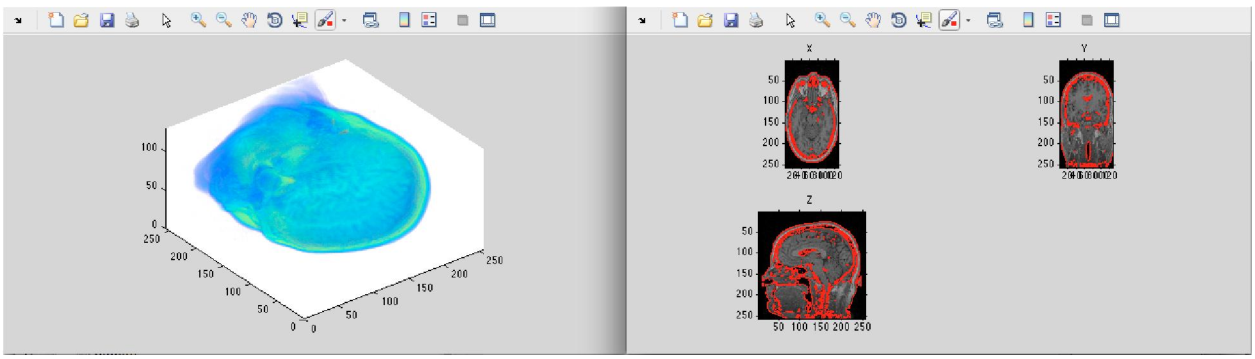

Figure 11Iteration of Chan-Vese Algorithm on MR Image with evolving contour

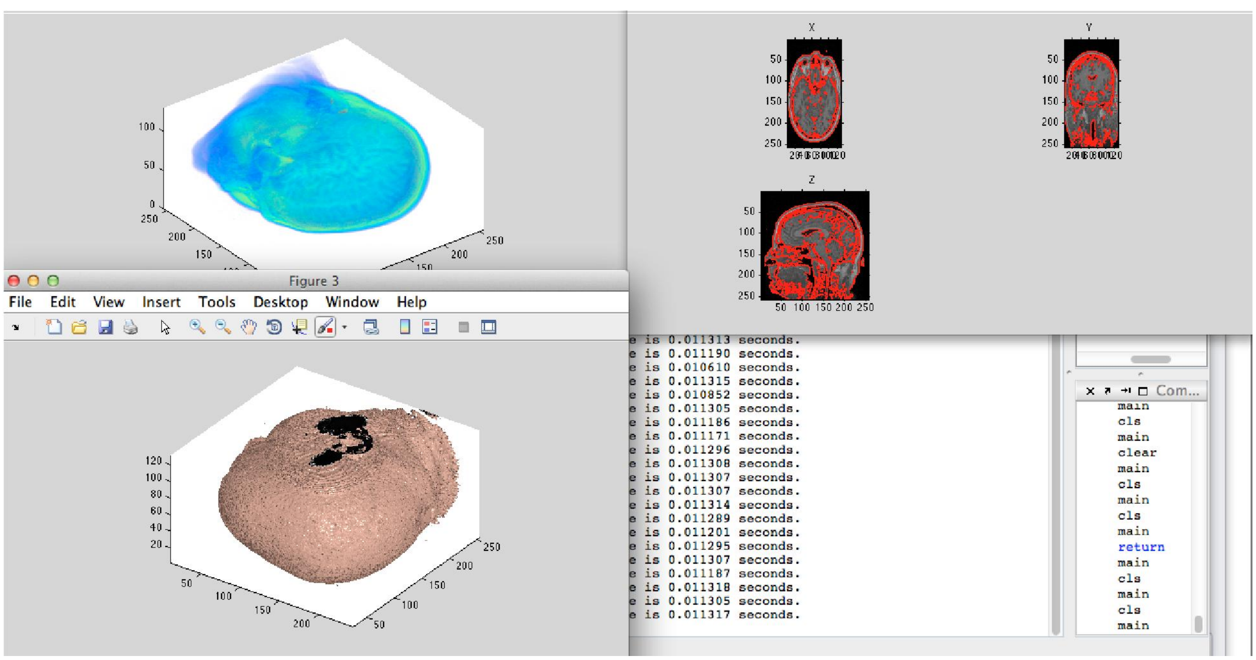

Figure 12 Segmentation result on MR Image

5. Discussion

From the results we can see that the algorithm works quite well even on images where objects do not have sharp edges. The application on 3D also provides good results for MRI images. The algorithm converges quickly. This implementation differs from the Chan-Vese algorithm in the following ways:

1. Instead of mean curvature it uses Gaussian curvature which does not guarantee convergence but works well in practice.
2. The implementation is targeted for 3D objects rather than 2D images as in the original paper.

References

[1] Tony F. Chan & Luminita A. Vese, "Active Contours without Edges" in IEEE transactions on image processing 10(2), 2001, pp. 266-277.

[2] Image Processing Course by Todd Wittman http://www.math.ucla.edu/~wittman/Fields/

[3] "Level Set and PDE Methods for Computer Graphics". Notes for SIGGRAPH 2004. [http://www.museth.org/Ken/Publications_files/Breen-etal_SIG04.pdf]