MATH473 HW5

Jiasen Zhang

1 Problem 1

Given

$$\mathbf{x} = (\cdots, x_{-2}, x_{-1}, x_0, x_1, x_2, \cdots)$$

Compute $X_l = D(\mathbf{x} * l_a)$:

$$\begin{aligned}
&\text{With} \quad (\mathbf{x} * l_a)_k = x_k (l_a)_0 + x_{k-1}(l_a)_1 = \frac{x_k + x_{k-1}}{2} \\
&\Rightarrow \quad \mathbf{x} * l_a = (\cdots, \frac{x_{-2}+x_{-1}}{2}, \frac{x_{-1}+x_0}{2}, \frac{x_0+x_1}{2}, \cdots) \qquad (\mathbf{x} * l_a)_0 = \frac{x_{-1}+x_0}{2} \\
&\Rightarrow \quad X_l = D(\mathbf{x} * l_a) = (\cdots, \frac{x_{-3}+x_{-2}}{2}, \frac{x_{-1}+x_0}{2}, \frac{x_1+x_2}{2}, \cdots) \qquad (X_l)_0 = \frac{x_{-1}+x_0}{2}
\end{aligned}$$

Compute $X_h = D(\mathbf{x} * h_a)$:

$$\begin{aligned}
&\text{With} \quad (\mathbf{x} * h_a)_k = x_k (h_a)_0 + x_{k-1}(h_a)_1 = \frac{x_k - x_{k-1}}{2} \\
&\Rightarrow \quad \mathbf{x} * h_a = (\cdots, \frac{x_{-1}-x_{-2}}{2}, \frac{x_0-x_{-1}}{2}, \frac{x_1-x_0}{2}, \cdots) \qquad (\mathbf{x} * h_a)_0 = \frac{x_0-x_{-1}}{2} \\
&\Rightarrow \quad X_h = D(\mathbf{x} * h_a) = (\cdots, \frac{x_{-2}-x_{-3}}{2}, \frac{x_0-x_{-1}}{2}, \frac{x_2-x_1}{2}, \cdots) \qquad (X_h)_0 = \frac{x_0-x_{-1}}{2}
\end{aligned}$$

Now to get $\tilde{\mathbf{x}}$, compute $V_l = l_s * U(X_l)$:

$$\begin{aligned}
&U(X_l) = (\cdots, \frac{x_{-3}+x_{-2}}{2}, 0, \frac{x_{-1}+x_0}{2}, 0, \frac{x_1+x_2}{2}, \cdots) \qquad U(X_l)_0 = \frac{x_{-1}+x_0}{2} \\
&\text{With} \quad (l_s * U(X_l))_k = U(X_l)_k (l_s)_0 + U(X_l)_{k-1}(l_s)_1 = U(X_l)_k + U(X_l)_{k-1} \\
&\Rightarrow \quad V_l = l_s * U(X_l) = (\cdots, \frac{x_{-3}+x_{-2}}{2}, \frac{x_{-3}+x_{-2}}{2}, \frac{x_{-1}+x_0}{2}, \frac{x_{-1}+x_0}{2}, \frac{x_1+x_2}{2}, \cdots) \\
&\qquad (V_l)_0 \quad \text{is the first} \quad \frac{x_{-1}+x_0}{2}
\end{aligned}$$

Compute $V_h = h_s * U(X_h)$:

$$\begin{aligned}
&U(X_h) = (\cdots, \frac{x_{-2}-x_{-3}}{2}, 0, \frac{x_0-x_{-1}}{2}, 0, \frac{x_2-x_1}{2}, \cdots) \qquad U(X_h)_0 = \frac{x_0-x_{-1}}{2} \\
&\text{With} \quad (h_s * U(X_h))_k = U(X_h)_k (h_s)_0 + U(X_h)_{k-1}(h_s)_1 = -U(X_l)_k + U(X_l)_{k-1} \\
&\Rightarrow \quad V_h = h_s * U(X_h) = (\cdots, \frac{x_{-3}-x_{-2}}{2}, \frac{x_{-2}-x_{-3}}{2}, \frac{x_{-1}-x_0}{0}, \frac{x_0-x_{-1}}{2}, \frac{x_2-x_1}{2}, \cdots) \\
&\qquad (V_h)_0 = \frac{x_{-1}-x_0}{2}
\end{aligned}$$

Compute $\tilde{\mathbf{x}} = V_l + V_h$ and compare it with $\mathbf{x}$:

$$\tilde{\mathbf{x}} = (\cdots, x_{-3}, x_{-2}, x_{-1}, x_0, x_1, \cdots) \qquad \tilde{x}_0 = x_{-1}$$
$$\mathbf{x} = (\cdots, x_{-2}, x_{-1}, x_0, x_1, x_2, \cdots) \qquad x_0 = x_0$$

By comparing $\tilde{\mathbf{x}}$ and $\mathbf{x}$, we can find $\tilde{x}_k = x_{k-1}$. So the output $\tilde{\mathbf{x}}$ is a "delay 1" of $\mathbf{x}$.

2 Problem 2

Given

$$x = (\cdots, x_{-2}, x_{-1}, x_0, x_1, x_2, \cdots)$$

- Compute $X_l = D(x * l_a)$ and $X_h = D(x * h_a)$:

$$\text{With} \quad (x * l_a)_k = x_k (l_a)_0 = x_k$$
$$\Rightarrow \quad x * l_a = (\cdots, x_{-2}, x_{-1}, x_0, x_1, x_2, \cdots) = x$$
$$\Rightarrow \quad X_l = D(x * l_a) = (\cdots, x_{-2}, x_0, x_2, \cdots)$$

$$\text{With} \quad (x * h_a)_k = x_{k-1} (h_a)_1 = x_{k-1}$$
$$\Rightarrow \quad x * h_a = (\cdots, x_{-2}, x_{-1}, x_0, \cdots)$$
$$\Rightarrow \quad X_h = D(x * h_a) = (\cdots, x_{-3}, x_{-1}, x_1, \cdots)$$

- Compute $U(X_l)$ and $U(X_h)$:

$$U(X_l) = (\cdots, x_{-2}, 0, x_0, 0, x_2, \cdots) \qquad U(X_l)_0 = x_0$$
$$U(X_h) = (\cdots, x_{-3}, 0, x_{-1}, 0, x_1, \cdots) \qquad U(X_h)_0 = x_{-1}$$

- To make sure $l_s * U(X_l) + h_s * U(X_h) = x$, l_s should be an identify filter and h_s should be a delay whose latency is -1. So we have $(l_s)_0 = 1$ and $(h_s)_{-1} = 1$ with all other coefficients zero.

$$(l_s * U(X_l))_k = (l_s)_0 U(X_l)_k = U(X_l)_k$$
$$(h_s * U(X_h))_k = (h_s)_{-1} U(X_h)_{k+1} = U(X_h)_{k+1}$$

$$\Rightarrow \quad (l_s * U(X_l) + h_s * U(X_h))_k = U(X_l)_k + U(X_h)_{k+1}$$

By observing $U(X_l)$ and $U(X_h)$, we know $U(X_l)_k + U(X_h)_{k+1} = x_k$, so we have

$$l_s * U(X_l) + h_s * U(X_h) = x$$

3 Problem 3

The level set formulation of the active contour model of Chan Vese method is:

$$
\begin{aligned}
F(c_1, c_2, \varphi) =& \mu \int_\Omega \delta_\varepsilon(\varphi)|\nabla\varphi| dxdy + \nu \int_\Omega H_\varepsilon(\varphi) dxdy + \lambda \int_\Omega |u_0 - c_1|^2 H_\varepsilon(\varphi) dxdy \\
&+ \lambda \int_\Omega |u_0 - c_2|^2 (1 - H_\varepsilon(\varphi)) dxdy \\
&c_1 = \frac{\int_\Omega u_0 H_\varepsilon(\varphi) dxdy}{\int_\Omega H_\varepsilon(\varphi) dxdy} \qquad c_2 = \frac{\int_\Omega u_0 (1 - H_\varepsilon(\varphi)) dxdy}{\int_\Omega (1 - H_\varepsilon(\varphi)) dxdy}
\end{aligned}
$$

Here H_ε and δ_ε are smoothed Heaviside and Delta functions. We use Euler-Lagrange equations, define $i(\tau) = F(\varphi + \tau\psi)$ and solve for $i'(0) = F'(\varphi) = 0$. Then we will get the weak solution and convert the problem $\min_\varphi F(\varphi)$ to a PDE problem.

$$
\begin{aligned}
\frac{\partial\varphi}{\partial t} &= -F'(\varphi) = -i'(0) \\
&= \delta_\varepsilon(\varphi) \left(\mu \cdot div(\frac{\nabla\varphi}{|\nabla\varphi|}) - \nu - \lambda(u_0 - c_1)^2 + \lambda(u_0 - c_2)^2 \right)
\end{aligned}
$$

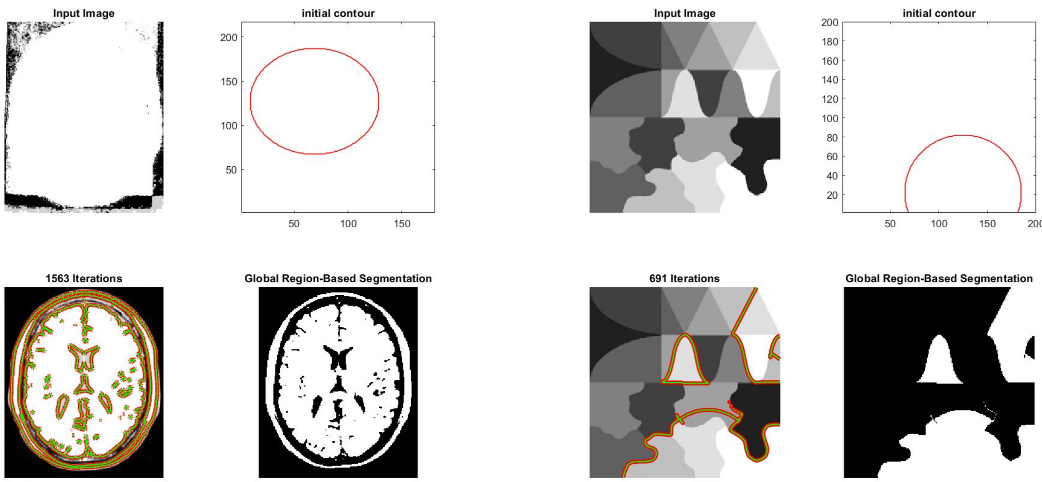
That is what we got in class and implemented in the codes.

I use the second code "chenvese.m", whose parameters include μ, φ_0 and whether using multiphase level set. Other parameters in this code are $\nu = 0$,$\lambda = 1$, time step $dt = 0.5$. The smoothed Heaviside function is $H_{1,\varepsilon}$ introduced in class. Besides, it provides some initial curves.

When initialization, the initial curve is represented by corresponding level set. For each time step, the code computes c_1, c_2, the term $-(u_0 - c_1)^2 + (u_0 - c_2)^2$, the term $\mu \cdot div(\frac{\nabla\varphi}{|\nabla\varphi|})$ and normalize it. Then it adds them together and normalize it again to get the external force. Finally, it computes φ for the next step and check whether to stop.

The case of multiphase is similar. When there are two level sets there will be four phases. There are two PDEs regarding φ_1 and φ_2 and each level set function is affected by four image forces in the four phases. The difference is that multiphase case may need reinitialization.

First I use single level set, set $\mu = 0.2$, use at most 3000 iterations and large circular initial mask. The results are shown in Figure 1. Both of them stop iteration before reaching to 3000. The result of T1Web.mat is good since most details of brain are recognized. But the result of texmos3.s512.tiff is bad. Single level set is not enough to seperate this image.

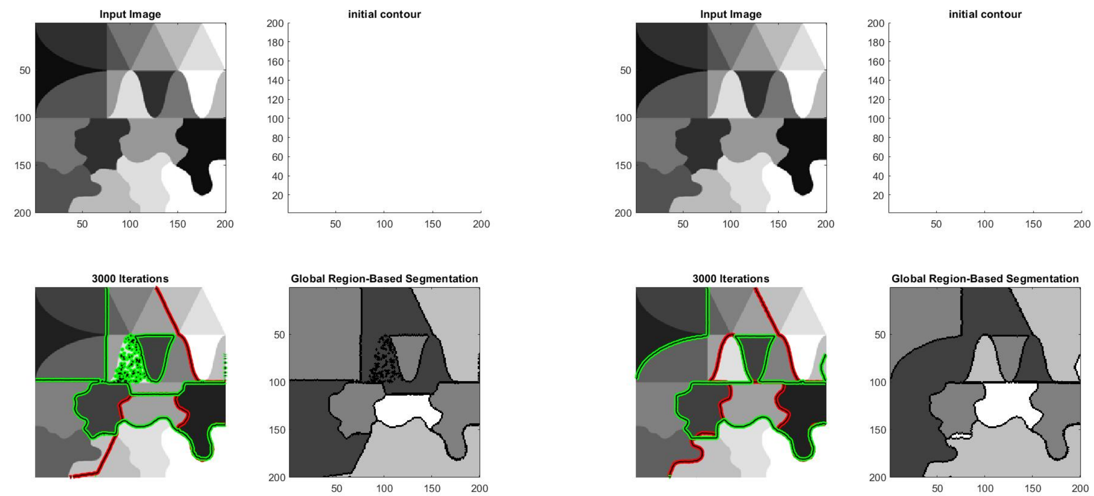


A : T1Web.mat

B : texmos3.s512.tiff

Figure 1: Results with single level set and $\mu = 0.2$.

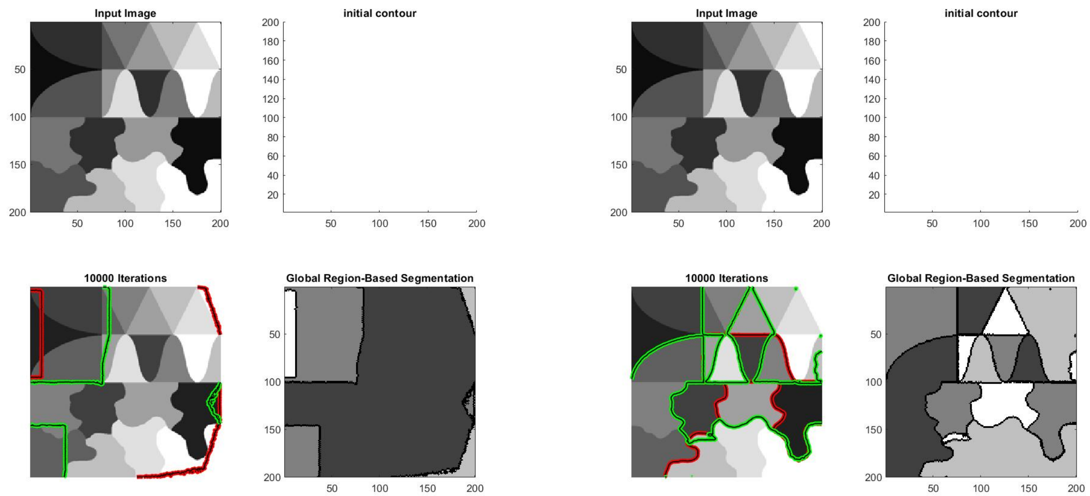
To improve the result of texmos3.s512.tiff, I use two level sets. As shown in Figure 2 A, the result is better than Figure 1 B but still improvable. In Figure 2 B I change μ to 0.07 and get much better result. This difference indicates the effect of the parameter μ. Because it means the weight of length term, if μ is too large the length of curve will be shorter and straighter. In most cases it's bad for image segmentation.



A : $\mu = 0.2$ B : $\mu = 0.07$

Figure 2: Results of texmos3.s512.tiff with two level sets.

To show the effect of μ, I get another two results with $\mu = 2$ and 0.002 for texmos3.s512.tiff (effect of μ is not obvious for T1Web.mat). As shown in Figure 3, when $\mu = 2$, the curves are very short and straight and can't segment the image well. The result with $\mu = 0.002$, with longer curves, is even better than that with $\mu = 0.07$ in Figure 2 B.



A : $\mu = 2$ B : $\mu = 0.002$

Figure 3: Results of texmos3.s512.tiff with two level sets.

4 Matlab Code

4.1 Problem 2

```
clear all;clc;

load T1Web.mat; % I
I2 = imread('texmos3.s512.tiff'); % I2

%u = demo_acwe(I2, 100);
seg = chenvese(I2,'large',10000,0.002,'multiphase');
```