CHANVESE

Active contour without edges.

Syntax

$Segmentation = chenvese(I, mask, num\_iter, mu, method)$

Description

$Segmentation = chenvese(I, mask, num\_iter, mu, method)$ performs active contour without edges on grayscale, binary, or color image I with initial phase $mask$.

Variable $method$ specifies the algorithm of active contour. Its default value is $'chan'$.

Value	Meaning
$'chan'$	Chan-Vese method (active contour without edge)
$'vector'$	Chan-Vese method (active contour without edge for vector image)
$'multiphase'$	Chan-Vese method (A Multiphase Level Set active contour)

Here $mask$ is created by either the user or the built-in mask creation function. In order to define a user $mask$, the user should make sure that the dimensions of $mask$ match those of the input image I, i.e. $size(m, 1) = size(I, 1)$ and $size(m, 2) = size(I, 2)$.

For $method = 'chan'$ *or* $'vector'$, only the top layer $mask$ is used, i.e. $size(m, 3) = 1$.
But for $method = 'multiphase'$, two layers $mask$ is used, i.e. $size(m, 3) = 2$.

For built-in mask creation function, keywords and their meanings are listed as follows. Depending on the value of $method$, the $mask$ may have one layer or two layers.

Value	Meaning
$'small'$	Creating a small circular mask with r = 9
$'medium'$	Creating a medium circular mask
$'large'$	Creating a large circular mask
$'whole'$	Creating a mask with small holes all over the mask
$'whole + small'$	Creating a 'whole' mask on one layer and another 'small' mask on the other layer

num_iter is the iteration number for algorithm.

mu is the same variable defined in the paper, while $lambda1$ and $lambda2$ in the paper are simply defined to 1 . The default value of mu is 0.2.

$Segmentation$ is the output segmentations gotten by the Chan-Vese active contour. $Segmentation$ is a binary image for $method = 'chan'$ *or* $'vector'$ and it is a labeled matrix for different partitions of the original image for $method = 'multiphase'$.

Class Support

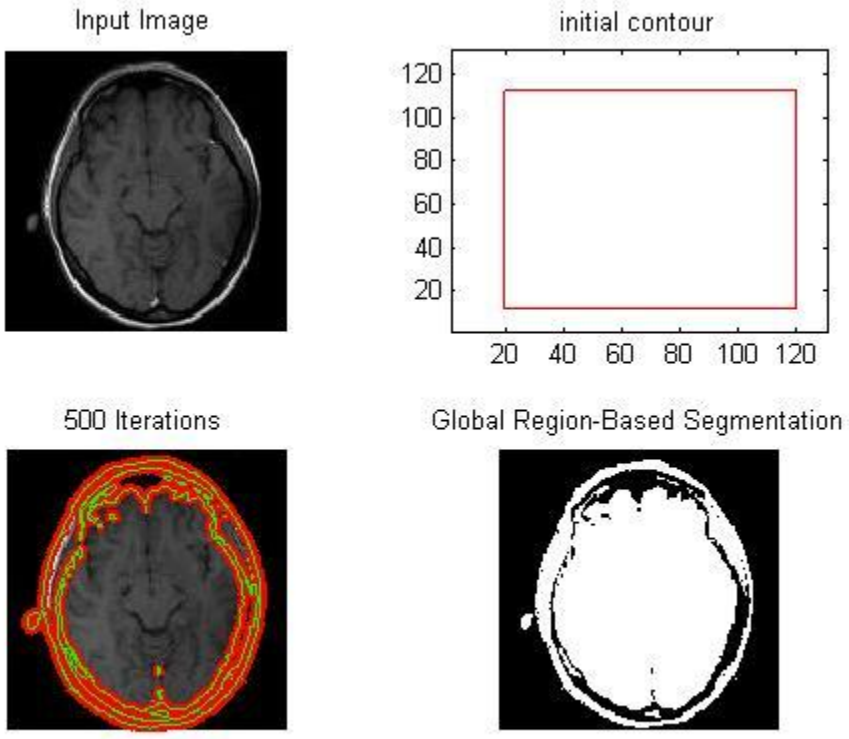
When using a user defined $mask$, the $mask$ can only be logical. Input image I can be double, binary, gray and RGB.

Examples

Example 1: Creating Customerlized Mask in CHANVESE

Find the segmentation for a gray image with $method$ 'chan' and customerlized $mask$.

```
I = imread('brain.jpg'); % read input image
m = zeros(size(I,1),size(I,2)); % initialize the dimensions of the mask
m(20:120,20:120) = 1; % create the mask
seg = chenvese(I,m,500,0.1,'chan'); % get segmentation
```

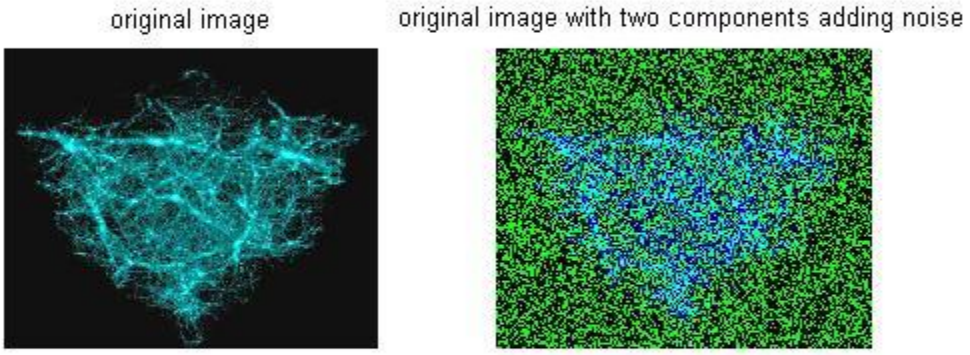


Example 2: Active contour on noisy image

Show the strong anti-noise ability of active contours without edge for vector image.

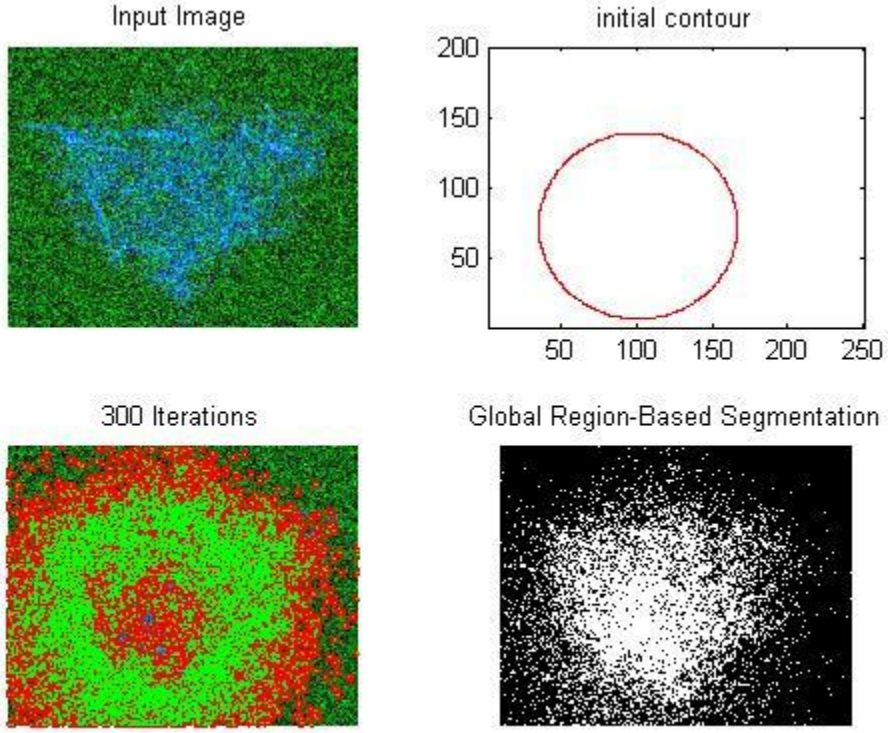
First create a noisy image from Matlab. Two types of noises have been added in the original image.

```
P = imread('anti-mass.jpg'); % Get original image
I = P;
I(:,:,1) = imnoise(I(:,:,1),'speckle'); % add noise on component 1
I(:,:,2) = imnoise(I(:,:,2),'salt & pepper',0.8); % add noise on component 2
figure(),subplot(1,2,1),imshow(P),title('original image'); % show original & noisy image
subplot(1,2,2),imshow(I),title('original image with two components adding noise')
```



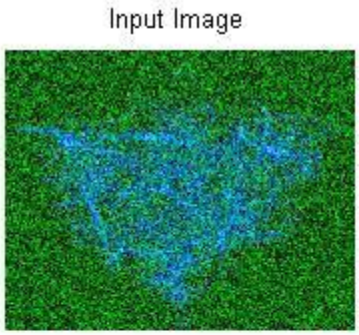
Method 'chan' is not good enough for this noisy image. Its result is showed as follows.

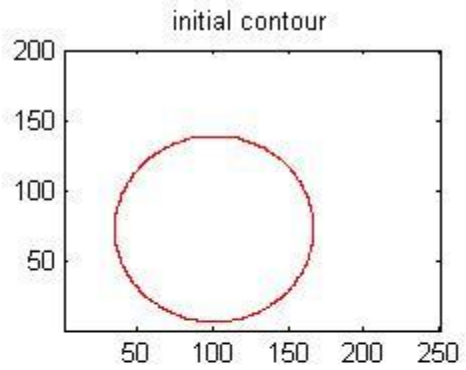
```
seg = chenvese(I,'large',300,0.02,'chan');
```

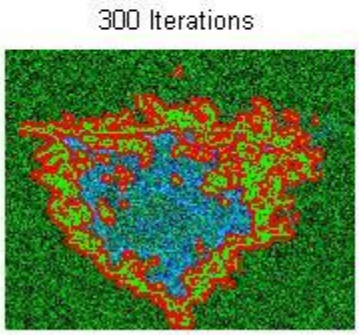


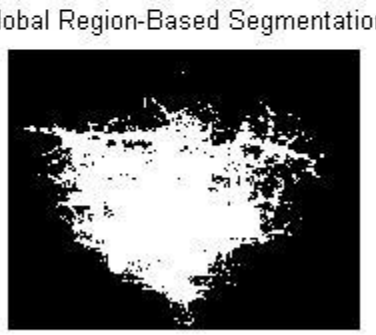
Use method 'vector' i.e. Chan-Vese algorithm for vector image.

```
seg = chenvese(I,'large',300,0.02,'vector');
```



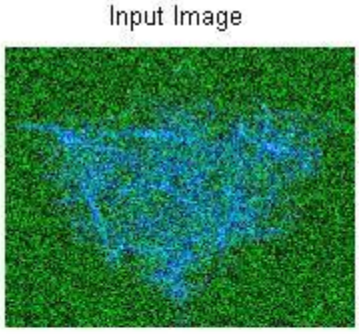


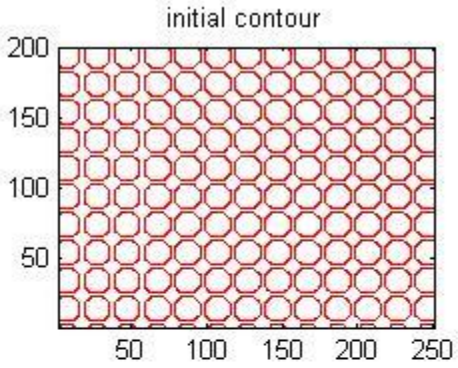


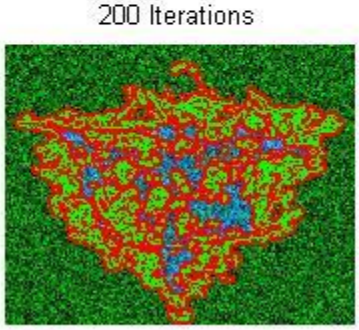


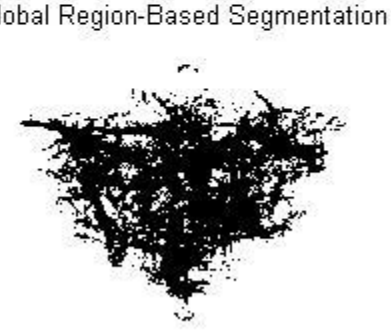
As we know, active contour is sensitive to its initial positions. Thus a different mask may lead to a different segmentation. Here built-in $mask$ = $'whole'$ is used and it leads to better and faster result.

```
seg = chenvese(I,'whole',200,0.02,'vector');
```







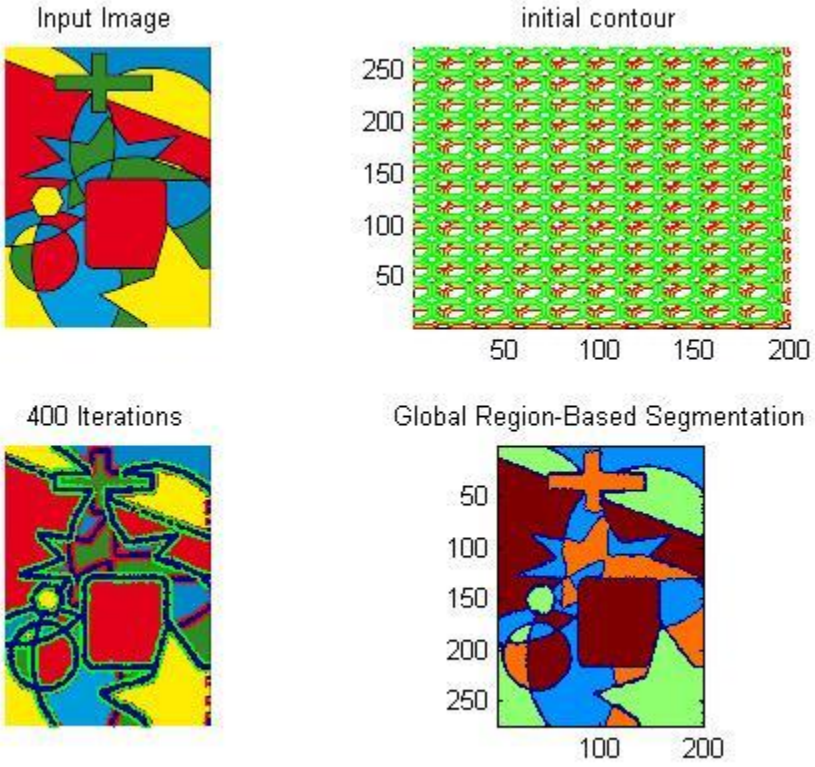


Example 3: Multiphase Active Contours

Get the segmentation via multiphase algorithm. For two phases, we can at most distinguish four partitions, namely four colors.

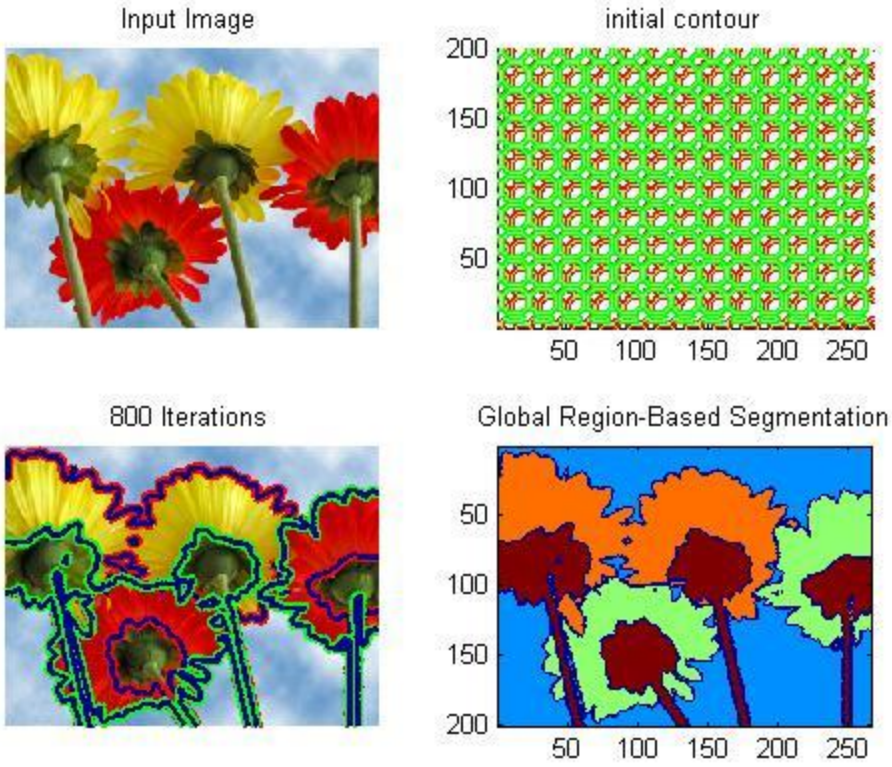
First, we apply this algorithm on a synthesized image.

```
I = imread('4colors.jpg');
seg = chenvese(I,'whole',400,0.1,'multiphase');
```



Now, we apply this algorithm to a real image.

```
I = imread('flowers.jpg');
seg = chenvese(I,'whole',800,0.2,'multiphase');
```



Reference

1. Chan, T. F., & Vese, L. A. (2001). Active contours without edges. IEEE Transactions on Image Processing, 10(2), 266-277.
2. Chan, T.F., & Sandberg Y. B(2000). Active contours without edges for Vector-valued Image. Journal of Visual Communication and Image Representation 11, 130–141 (2000)
3. Chan, T. F., & Vese, L. A. (2002). A Multiphase level set framework for image segmentation using the Mumford and Shah model. International Journal of Computer Vision 50(3), 271–293, (2002)
4. Kass, M., Witkin, A., & Terzopoulos, D. (1988). Snakes: Active contour models. International Journal of Computer Vision, 1(4), 321-331.
5. J. A. Sethian, Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science (Cambridge ... on Applied and Computational Mathematics), Cambridge University Press; 2 edition (1999)
6. Matlab Help, Mathworks Inc.
7. Shawn Lankton, Active Contour Matlab Code Demo, http://www.shawnlankton.com/?s=active+contour