

# Adaptive level set evolution starting with a constant function

Yan Wang, Chuanjiang He

Gaetan Bahl

February 10, 2017

# Outline

## 1 Introduction

## 2 Method

- Notations
- Partial Differential Equation

## 3 Results

- New level set evolution method
- No need to re-initialize the level set function to a signed distance function every few iterations
- No need for a good initial estimation of the level set function
- Easy to implement and very fast

# Outline

## 1 Introduction

## 2 Method

- Notations
- Partial Differential Equation

## 3 Results

We denote:

- $I(x, y)$  is the intensity of the image
- $I_\sigma$  is the image blurred with gaussian noise of variance  $\sigma$
- $g(x) = \exp(-\frac{x}{20})$
- $H$  is the Heaviside function
- $c_1(\phi) = \frac{\int_{\Omega} I(x,y)H(\phi(x,y))dxdy}{\int_{\Omega} H(\phi(x,y))dxdy}$  and  
 $c_2(\phi) = \frac{\int_{\Omega} I(x,y)H(-\phi(x,y))dxdy}{\int_{\Omega} H(-\phi(x,y))dxdy}$

The evolution of the level set function  $\phi$  is controlled by two forces:

- an adaptive driving force,  $F_{adp}$ , that moves the level set function up or down depending on whether or not the point is part of the object

$$F_{adp} = g(|\nabla I_\sigma|) \text{sign}(I(x, y) - \frac{c_1 + c_2}{2})$$

- a total-variation based regularisation force,  $F_{reg}$  that smoothes the level set function

$$F_{reg} = g(|\nabla I_\sigma|) \text{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right)$$

It is then written as a PDE:

$$\frac{d\phi}{dt} = \alpha F_{adp} + \beta F_{reg}$$

With Neumann boundary conditions.  $\alpha$  and  $\beta$  are parameters to be set by the user.  $\phi$  can be initialized to any constant function.

# Outline

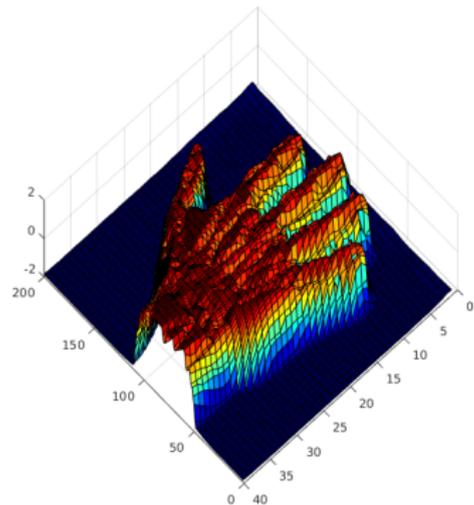
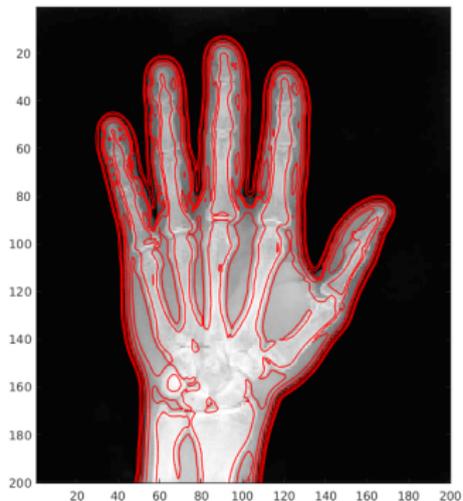
## 1 Introduction

## 2 Method

- Notations
- Partial Differential Equation

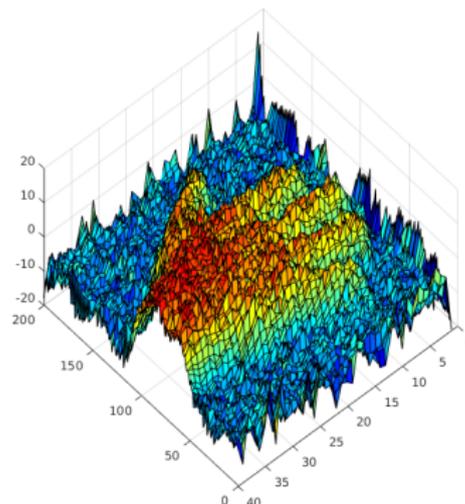
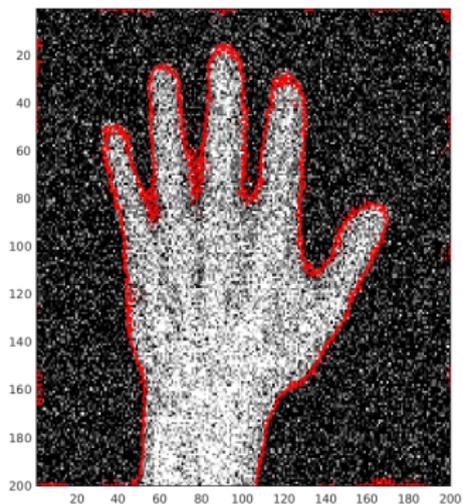
## 3 Results

I implemented the method and tried it on an X-Ray image of a hand. The method is very fast indeed and gives excellent results.



Original image with level set isovalues and level set function

This method works great even with very noisy images.



Same image with a lot of noise