Image segmentation

Segmentation of fluorescent microscopy images of living cells

Christian V. Hansen
University of southern Denmark

Bachelor Project
Advisor: Achim Schroll
Co-Advisor: Niels Christian Overgaard
Department of Mathematics and Computer Science
At University of southern Denmark, Odense.
May 31, 2012

Abstract

This Bachelor project is concerned to Image segmentation. There are several ways to segment images but in this project we focused on the "Active contours without edges" also called the Chan-Vese model. The images that are considered are all transformed to grey scale images and considered mathematically where each grey tone for each pixel are connected to a corresponding number. In contradiction to other models the Chan-Vese model does not depend on the gradients on the image, therefore it would be able to detect objects on more blurred images. The so called Chan-Vese model is derived and presented from the principle of minimizing an energy function which contains different fitting and regularization terms.

In the last half of the report it is explained how the Chan-Vese method is implemented by the use of a level set function and numerical approximations, and in the end the implementation was tested on different images. The test results show that the method works best on images where the object and the background are in contrast colour and the edge of the object is strong.

The whole motivation for studying image segmentation comes from the interest of developing an automated computer analysis for FLIP images of cells. And to do this we should first be able to identify and describe the cell in the FLIP images. In the end part of this project the Chan-Vese method was tested on such an image of a cell. A test which seems to be successful since the Chan-Vese method can be forced to make a smooth contour who fits nicely around the cell.

Acknowledgements

In connection with the creation of this project I would like to send a special thanks to my advisor Achim Schroll for his great support, advices and especially his great involvement, I am really thankful for that. Thanks to Achim we got in touch with Niels Chr. Overgaard from Lund university, who I would also like to thank for having introduced us to Image segmentation, a material that was new to me, and thank you so much for the help through this project. I would also like to thank Martin Svensson for providing us with useful material and for helping and advising us in the last challenging part of the project.
Without the help from these people, especially my advisor Achim Schroll who has spend a lot of time working on this topic, this project would not have been possible.

Contents

1 Motivation

Fluorescence loss in photobleaching FLIP is a modern microscopy method to visualize transport processes in living cells. To develop an automated computer analysis for FLIP images, the first step is to identify the geometry of the cell from the image.

To take this first step into making the automated computer analysis of the FLIP images, this Bachelor project would identify and describe a functioning segmentation method of FLIP images provided by BMB at SDU. In the beginning of this project we visited Niels Chr. Overgaard from the mathematical imaging group at Lund University, where we were introduced to two different image segmentation methods. The first method is the so called "Snake" method that works like a balloon or a rubber band that winds around the object on an image. The other method which is the one we will consider in this project is the "Active contour without edges" also called the Chan-Vese method. Different from the Snake method the Chan-Vese active contour can split up into several contours and the initial contour does not necessarily have to be outside the object on the image. The goal with this report is to derive and describe the Chan-Vese model, such that it can be implemented and used on the FLIP images.

2 Active Contours Without Edges

2.1 Chan-Vese method

The classical image segmentation models using active contours or snakes often starts by setting a curve around the object we want to detect. The curve then moves towards the object, and it stops at the edges of the object. These edges are usually detected by use of the gradients of the image. The Chan-Vese method is able to detect objects whose boundaries are not necessarily defined by gradients. Like the snake method this method will minimize an energy until it reaches the desired boundary, but in the Chan-Vese method the stopping term will not be dependent on the gradients.

In the following section we will discuss the Chan-Vese method and consider the model that evolves its contour by use of a level set function.

2.2 Level set method

The level set method is a very useful tool to describe time varying objects. It is a numerical technique that first time was described in 1988 by Osher, S. and Sethian, J. A. see [2], today it is used to solve a lot of problems. An example of a level set function is seen in Figure 1. The level set method is represented such that topological changes such as a shape which splits up in two are handled efficiently. Furthermore an advantage of the level set method is that we don't need to parametrize curves, but just do numerical operations on the fixed Cartesian mesh.

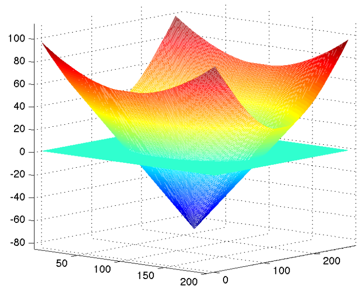


Figure 1: Level set function ϕ

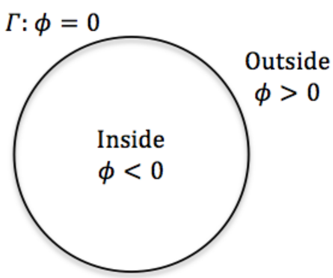


Figure 2: Zero level contour

At Figure 1 we see the surface plot of the level set function $\phi(\mathbf{x})$ $\forall \mathbf{x} \in \mathbb{R}^2$ and at Figure 2 the interface $\Gamma = \{\mathbf{x} = (x, y) \in \mathbb{R}^2 | \phi(x, y) = 0\}$ where Γ is a closed contour, can be seen. The contour is then the zero level of the level set function. Since we define Γ as a closed contour Γ divides the domain Ω in to two sub domains such that inside Γ we have that $\phi < 0$ and $\phi > 0$ outside the contour. When using the method in practise we normally initialize the level set function from a given contour Γ as follows:

$$\phi(\mathbf{x}) = \begin{cases} -distance(\mathbf{x}, \Gamma), & \mathbf{x} \in \text{inside } \Gamma, \\ distance(\mathbf{x}, \Gamma), & \mathbf{x} \in \text{outside } \Gamma. \end{cases} \tag{1}$$

The evolving curve $\Gamma(t)$ actuate in this method concurrently with the minimization of an energy based segmentation at time t. Let's now consider the basic idea behind this model.

2.3 Description of the Chan-Vese model

Let I be the image, such that $I(\mathbf{x})$, for $\mathbf{x} \in \mathbb{R}^2$, is a function in the domain Ω and $I(\mathbf{x})$ gives the intensity in a grey scale image. We now consider the case where the image has two regions, each region has approximatively the same values everywhere inside itself, like a homogeneous black object on a white background. With this consideration in mind, let's now consider the following fitting term from the Chan-Vese method:

$$E_{fit}(\Gamma) = F_1(\Gamma) + F_2(\Gamma) = \int_{inside(\Gamma)} |I(\mathbf{x}) - c_-|^2 d\mathbf{x} + \int_{outside(\Gamma)} |I(\mathbf{x}) - c_+|^2 d\mathbf{x} \tag{2}$$

Here Γ is any variable closed contour, c_- is the average inside Γ and c_+ is the average outside. Now let Γ_0 be the contour on the edge of the object on a given image. We then minimize the fitting term above, such that the greatest lower bound for the set is:

$$\inf_{\Gamma}\{F_1(\Gamma) + F_2(\Gamma)\} \approx 0 \approx F_1(\Gamma_0) + F_2(\Gamma_0) \tag{3}$$

From (2) we can see that if the curve is outside the object, then there is a wide span of points/pixels with different values, so $F_1(\Gamma) > 0$ and $F_2(\Gamma) \approx 0$, This is because we expect the points outside the curve to have an approximatively uniform value, such that each value minus the average is ≈ 0. Opposite if the curve is inside the object, then $F_1(\Gamma) \approx 0$ and $F_2(\Gamma) > 0$. This is illustrated below:

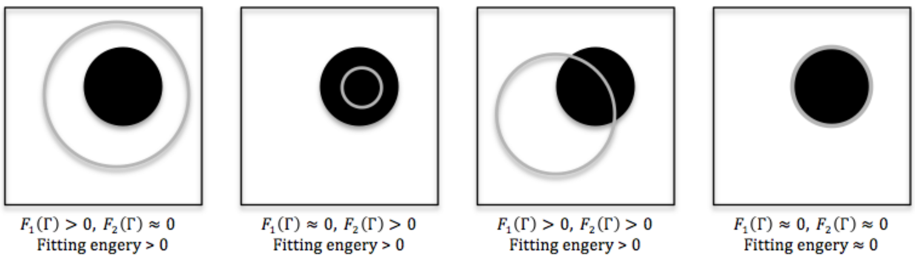


Figure 3: Four different fitting cases.

From this we see that the fitting term E_{fit} is only minimized when Γ lies on the edge of the object such that $\Gamma = \Gamma_0$.

In the original paper [1] published by T. Chan and L. Vese they present some extra regularization terms, in addition to the fitting term, in form of the length of the curve Γ and the area inside Γ, therefore it ends up with the following fitting energy functional:

$$\begin{aligned} E(c_-, c_+, \Gamma) =& \mu \cdot \text{Length}(\Gamma) + v \cdot \text{Area}(inside(\Gamma)) \\ &+ \lambda_1 \int_{inside(\Gamma)} |I(\mathbf{x}) - c_-|^2 d\mathbf{x} \\ &+ \lambda_2 \int_{outside(\Gamma)} |I(\mathbf{x}) - c_+|^2 d\mathbf{x} \end{aligned} \tag{4}$$

where $\mu, v \geq 0$ and $\lambda_1, \lambda_2 > 0$ are fixed parameters. Generally v is set to zero, so we will not discuss this area regularizing term that much in this project. However if we expect the object to have a very smooth boundary we consider the term that concerns the length of the curve and set μ to be large weighted. The parameters λ_1, λ_2 are generally fixed to be one, but if we want to have a more uniform foreground (inside Γ) than background (outside Γ), then we set λ_1 to have a higher weight than λ_2, and if we for example consider an image with an object on a black background we then might want to set $\lambda_2 > \lambda_1$.

2.4 Model described with the level set method

In Section 2.2 we introduced the level set function ϕ which will now be used to describe the evolving curve $\Gamma(t)$ at time t since it is unknown. So we set $\phi(\mathbf{x}, t)$ to be the evolving level set function such that $\Gamma(t) = \{\mathbf{x} \in \mathbb{R}^2 | \phi(\mathbf{x}, t) = 0\}$. In (4) we integrate over the area inside Γ which correspond to the area where $\phi \leq 0$ and outside Γ where $\phi > 0$ (See Figure 2). To make this easier we now introduce the Heaviside function H and also the Dirac measure δ is presented here and defined as follows:

$$H(g(x,y)) = \begin{cases} 1, & \text{if } g(x,y) \geq 0, \\ 0, & \text{if } g(x,y) < 0. \end{cases} \quad \text{and} \quad \delta(g(x,y)) = \nabla H(g(x,y)) \tag{5}$$

In our case g is the level set function $\phi(\mathbf{x})$ with $\mathbf{x} \in \mathbb{R}^2$, therefore it is given that the Dirac delta defines the curve, such that by integrating over this region we get the length of the curve. Since the Dirac delta is equal to the gradient of the Heaviside function we then express the curve length as follows:

$$\text{Length}(\Gamma) = \text{Length}\{\phi = 0\} = \int_\Omega |\nabla H(\phi(\mathbf{x}))| \, d\mathbf{x} \tag{6}$$

The area regularization term is simply the integral of one minus the Heaviside function for the level set function ϕ over the entire domain Ω, since the Heaviside function is zero inside the contour and one outside.

$$Area\{\phi \leq 0\} = \int_\Omega (1 - H(\phi(\mathbf{x}))) \, d\mathbf{x} \tag{7}$$

We now consider the fitting terms and describe these with the level set method too, so given a closed curve Γ we get:

$$\begin{aligned}\int_{inside(\Gamma)} |I(\mathbf{x}) - c_-|^2 \, d\mathbf{x} &= \int_{\phi \leq 0} |I(\mathbf{x}) - c_-|^2 \, d\mathbf{x} \\ &= \int_\Omega |I(\mathbf{x}) - c_-|^2 (1 - H(\phi(\mathbf{x}))) \, d\mathbf{x} \end{aligned} \tag{8}$$

and

$$\begin{aligned}\int_{outside(\Gamma)} |I(\mathbf{x}) - c_+|^2 \, d\mathbf{x} &= \int_{\phi > 0} |I(\mathbf{x}) - c_+|^2 \, d\mathbf{x} \\ &= \int_\Omega |I(\mathbf{x}) - c_+|^2 H(\phi(\mathbf{x})) \, d\mathbf{x} \end{aligned} \tag{9}$$

We can then write the fitting energy as follows:

$$\begin{aligned} E(c_-, c_+, \Gamma) =& \mu \int_\Omega |\nabla H(\phi(\mathbf{x}))| \, d\mathbf{x} + v \int_\Omega (1 - H(\phi(\mathbf{x}))) \, d\mathbf{x} \\ &+ \lambda_1 \int_\Omega |I(\mathbf{x}) - c_-|^2 (1 - H(\phi(\mathbf{x}))) \, d\mathbf{x} \\ &+ \lambda_2 \int_\Omega |I(\mathbf{x}) - c_+|^2 H(\phi(\mathbf{x})) \, d\mathbf{x} \end{aligned} \tag{10}$$

where $\mu, v \geq 0$ and $\lambda_1, \lambda_2 > 0$ are still fixed parameters, the constant c_+ will follow from the optimal grey level, that is when the energy is minimized. So now given that Γ is fixed we set:

$$0 = \frac{\partial}{\partial c_+} \left(\int_\Omega H(\phi(\mathbf{x}))(I(\mathbf{x}) - c_+)^2 d\mathbf{x} \right) = -2 \int_\Omega H(\mathbf{x}))(I(\mathbf{x}) - c_+)^2 d\mathbf{x} \tag{11}$$

Since $H(\phi(\mathbf{x}))(I(\mathbf{x}) - c_+)^2$ is positive, the energy is for sure minimal in that equation. So by rearranging the equation we get:

$$c_+(\phi) = \frac{\int_\Omega I(\mathbf{x}) H(\phi(\mathbf{x})) \, d\mathbf{x}}{\int_\Omega H(\phi(\mathbf{x})) \, d\mathbf{x}} \tag{12}$$

Such that the optimal c_+ can be interpreted as the average of $I(\mathbf{x})$ over the area where $\phi > 0$.

The same optimization can be made for c_- such that c_- becomes:

$$c_-(\phi) = \frac{\int_\Omega I(\mathbf{x})(1 - H(\phi(\mathbf{x}))) \, d\mathbf{x}}{\int_\Omega (1 - H(\phi(\mathbf{x}))) \, d\mathbf{x}} \tag{13}$$

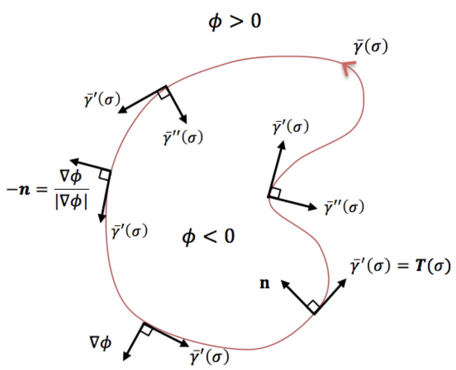
Then the optimal c_- will be the average inside the contour where $\phi \leq 0$.

Later on we will use these average intensities c_- and c_+ to implement the method.

2.5 Energy minimization and the variational PDE

Before we start this section let's first introduce the notation that will be used here.

We will later on in this section consider a parametrized closed curve $\gamma(s)$ with parameter s such that γ is periodic. For this we also assume mathematical positive orientation such that the normal vector points inside the contour. In the case of the arc length parametrization σ the unit normal vector to the closed curve $\bar{\gamma}(\sigma)$ is defined as: $\mathbf{n} := R(\frac{\pi}{2})\bar{\gamma}'(\sigma)$, where $R(\frac{\pi}{2})$ means rotation in mathematical positive direction by $\frac{\pi}{2}$ and $\bar{\gamma}(\sigma)$ is the unit tangent, such that $\mathbf{n}$ is the inward normal vector which can be seen in the figure.



In the previous sections we discussed the level set equation, where the zero level was considered as the contour Γ such that $\Gamma = \{\mathbf{x} \in \mathbb{R}^2 | \phi(\mathbf{x}) = 0\}$. By assuming $\phi < 0$ in the interior part of Γ the gradient of ϕ would have outward direction, such that we can set $\frac{\nabla\phi(\mathbf{x})}{|\nabla\phi|} = -\mathbf{n}(\mathbf{x})$ which is the outward unit normal.

To find the optimal level set equation for the image, we first have to find a contour and average grey levels c_- and c_+ such that they are the arguments that minimizes the fitting energy:

$$(\Gamma^*, c_-^*, c_+^*) = \arg\min_{\Gamma, c_-, c_+} E(\Gamma, c_-, c_+) \tag{14}$$

From equation (4) we now get the following fitting energy by setting $v = 0$ and thereby leaving the area parameter out:

$$\begin{aligned} E(c_-, c_+, \Gamma) =& \mu \int_\Gamma d\sigma \\ &+ \lambda_1 \int_{\phi<0} (I(\mathbf{x}) - c_-)^2 d\mathbf{x} \\ &+ \lambda_2 \int_{\phi>0} (I(\mathbf{x}) - c_+)^2 d\mathbf{x} \end{aligned} \tag{15}$$

note that $d\sigma = |\gamma'(s)|\; ds$ for the parametric closed and smooth curve $\gamma(s) \in C^1([a,b]\ ,\ \mathbb{R}^2)\ ,\ \gamma(a) = \gamma(b)$ thus that $\phi(\gamma(s), t) = 0$ and Γ is intended to be a time dependent contour $\Gamma(t)$ such that $\Gamma(t) = \{\mathbf{x} \in \mathbb{R}^2 | \phi(\mathbf{x}, t) = 0\}$.

Normal velocity:

From Figure 3 we see that the fitting term is minimal when the contour lies at the edge of the object on the image. Therefore we want to minimize the fitting energy from (15) to get the best contour to the image object.

Since it is the level set equation ϕ we want to change such that it minimizes the energy we now want to determine the PDE for the level set function. Let us first consider the normal velocity, this we do by first considering a particle $P(t)$, where $P(t)$ moves with the contour at time t, such that $P(t) \in \Gamma(t)$ and the level set function is therefore zero in that point: $\phi(P(t), t) = 0 \quad \forall t$. Based on this we now compute $\frac{d}{dt}\phi(P(t), t)$ by use of the chain rule:

$$\frac{\mathrm{d}}{\mathrm{dt}}\phi(P(t), t) = \nabla\phi(P(t), t) \cdot P'(t) + \phi_t(P(t), t) = 0$$

Which is also equal to zero since $\phi(P(t), t) = 0$.
We define the velocity of the particle as $\frac{d}{dt}P(t) = P'(t) = (P_1'(t), P_2'(t))$, and $P'(t) \in \mathbb{R}^2$

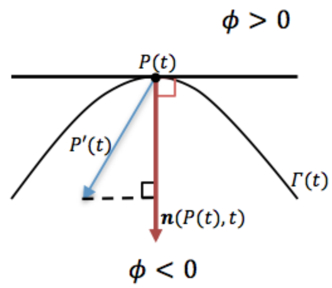
By rearranging and multiplying the previous equation with $\frac{1}{|\nabla\phi|}$ we get:

$$P'(t)\frac{\nabla\phi(P(t), t)}{|\nabla\phi(P(t), t)|} = -\frac{\phi_t(P(t), t)}{|\nabla\phi(P(t), t)|} \tag{16}$$

Which is simply $P'(t) \cdot (-\mathbf{n}(P(t), t))$ since:

$$\mathbf{n}(P(t), t) = -\frac{\nabla\phi(P(t), t)}{|\nabla\phi(P(t), t)|} \tag{17}$$

The inward unit normal is illustrated together with the curve $\Gamma(t)$ and the particle $P(t)$. On the illustration here, we see a projection of $P'(t) \in \mathbb{R}^2$ into a given direction $\mathbf{d}$, where $|\mathbf{d}| = 1$ such that the projection of $P'(t)$ into the direction $\mathbf{d}$ is $P'(t) \cdot \mathbf{d} \in \mathbb{R}$, which is simply the length of $P'(t)$ in the direction $\mathbf{d}$. In this case the direction vector $\mathbf{d}$ is replaced by the inward unit normal $\mathbf{n}(\mathbf{x}, t)$ from equation (17) where $|\mathbf{n}(\mathbf{x}, t)| = 1$.



With all this in mind we have now found that the normal velocity its defined by:

$$\frac{\mathrm{d}}{\mathrm{dt}}\Gamma(t) := P'(t) \cdot \mathbf{n}(P(t), t) = \frac{\phi_t(P(t), t)}{|\nabla\phi(P(t), t)|} \tag{18}$$

Note here that $\frac{\mathrm{d}}{\mathrm{dt}}\Gamma(t) = P'(t) \cdot \mathbf{n} \in \mathbb{R}$ is a scalar as discussed before.
Back to the minimization of the fitting energy for the Chan-Vese method, we now want to evolve the curve in direction of the decreasing energy. Per definition

∇E is the direction of increase for the energy E, so the decreasing energy is $-\nabla E$. Since we want an evolution equation in direction of normal-velocity and with a decreasing energy we consider:

$$\frac{\mathrm{d}}{\mathrm{dt}}\Gamma(t) = -\nabla_{\Gamma} E(\Gamma(t)) \cdot \mathbf{n} \qquad , \quad \text{where } \Gamma(0) \text{ is given} \tag{19}$$

By use of this and equation (18) we now see that the following must hold for the level set equation:

$$\phi_t = -\nabla E(\phi) \cdot \mathbf{n} \cdot |\nabla \phi| \tag{20}$$

where $\phi = \phi(\mathbf{x}, t)$, $\forall \mathbf{x} \in \Gamma(t)$ to the time t.

This means that we want to determine the gradient of the fitting energy from (15). Let's consider it in two parts, first consider the regularization term in form of the length l from the fitting energy function, that can be written as follows:

$$l = \int_a^b |\gamma'(s)| ds \tag{21}$$

where $\gamma'(s)$ is the tangent vector to the parametric closed and smooth curve $\gamma(s)$, $s \in [a, b]$. Instead of the parameter s, we want to use the natural parameter σ such that $\sigma \in [0, l]$ and defined as:

$$\sigma(s) = \int_a^s |\gamma'(q)| \; dq \tag{22}$$

Note that $l = \sigma(b)$ and $d\sigma = |\gamma'(q)| \; dq$.

We now consider $s(\sigma) = \sigma^{-1}(s)$ such that $s'(\sigma) = \frac{1}{\sigma'(s(\sigma))} = \frac{1}{|\gamma'(s(\sigma))|}$.
Thus the arc length parametrization of the curve $\gamma(s)$ becomes:

$$\bar{\gamma}(\sigma) = \gamma(s(\sigma)) \; : \; [0, l] \to \mathbb{R}^2 \tag{23}$$

For the tangent vector we see

$$\bar{\gamma}'(\sigma) = \gamma'(s(\sigma)) s'(\sigma) = \frac{\gamma'(s(\sigma))}{|\gamma'(s(\sigma))|} \tag{24}$$

such that $\mathbf{T}(\sigma) = \bar{\gamma}'(\sigma)$ is the unit tangent vector. Finally we can see that the length of the contour can be written in the following way:

$$l = \sigma(b) = \int_a^b |\gamma'(s)| ds = \int_0^l d\sigma = \int_0^l |\bar{\gamma}'(\sigma)|^2 d\sigma \tag{25}$$

since $|\bar{\gamma}'(\sigma)| = 1$.

So let's set $L[\bar{\gamma}]$ as the length of the curve such that:

$$L[\bar{\gamma}] = \int_0^l |\bar{\gamma}'(\sigma)|^2 d\sigma \tag{26}$$

We now consider an ε-variation to derive the shape gradient of the length of the contour. The ε-variation is given by $\bar{\gamma}_\varepsilon(\sigma) = \bar{\gamma}(\sigma) + \varepsilon \cdot w(\sigma)$, where $w(\sigma) : [0, l] \to \mathbb{R}^2$ is a periodic parametrized curve, such that $w(0) = w(l)$ and $w'(0) = w'(l)$. Then the variation of $L[\bar{\gamma}]$ is given by:

$$
\begin{aligned}
\delta L[\bar{\gamma}, w] &= \frac{d}{d\varepsilon} L[\bar{\gamma}_\varepsilon]\Big|_{\varepsilon=0} \\
&= \frac{d}{d\varepsilon} L[\bar{\gamma} + \varepsilon w]\Big|_{\varepsilon=0} \\
&= \frac{d}{d\varepsilon} \int_0^l |\bar{\gamma}'(\sigma) + \varepsilon w'(\sigma)|^2 \; d\sigma\Big|_{\varepsilon=0} \\
&= 2 \int_0^l (\bar{\gamma}'(\sigma) + \varepsilon w'(\sigma)) \cdot w'(\sigma) \; d\sigma\Big|_{\varepsilon=0} \\
&= 2 \int_0^l \bar{\gamma}'(\sigma) \cdot w'(\sigma) \; d\sigma \\
&= 2[\bar{\gamma}'(\sigma) \cdot w(\sigma)]_0^l - 2 \int_0^l \bar{\gamma}''(\sigma) \cdot w(\sigma) \; d\sigma \qquad (27) \\
&= -2 \int_0^l \bar{\gamma}''(\sigma) \cdot w(\sigma) \; d\sigma \qquad (28)
\end{aligned}
$$

To get to (27) we use partial integration, it can also be seen that the left term in (27) is equal to zero because of periodicity.

By definition the corresponding variational gradient is:

$$\nabla L[\bar{\gamma}] = -2\gamma''(\sigma) \qquad (29)$$

From Frenet equations which are also found in [4], saying that the derivative of the tangent with respect to the arclength is the curvature times the normal, we get:

$$\mathbf{T}'(\sigma) = \bar{\gamma}''(\sigma) = \kappa(\bar{\gamma}(\sigma))\mathbf{n}(\bar{\gamma}(\sigma)) = \kappa\mathbf{n}$$

Where κ is the signed curvature and $\mathbf{n}$ is the inward unit normal vector such that $|\mathbf{n}(\sigma)| = 1$. The curvature κ is here given by the divergence formula which is also derived in [4]:

$$\kappa = -\mathrm{div}(\mathbf{n}) = -\nabla \cdot (\mathbf{n}) = \nabla \cdot \left(\frac{\nabla\phi}{|\nabla\phi|}\right)$$

Finally we get that the variational gradient for the length of the contour is:

$$\nabla L[\bar{\gamma}] = -2\gamma''(\sigma) = -2\kappa(\sigma)\mathbf{n}(\sigma) = -2\nabla \cdot \left(\frac{\nabla\phi}{|\nabla\phi|}\right) \cdot \mathbf{n} \qquad (30)$$

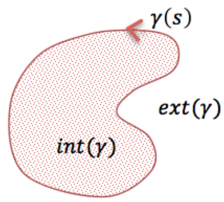
We will use this gradient later to derive the level set equation.

But first we have to find the gradient of the last term from the fitting energy function, which is the fitting term from (15):

$$E_{fit}[\gamma] = \int_{int(\gamma)} (I(\mathbf{x}) - c_-)^2 \; d\mathbf{x} + \int_{ext(\gamma)} (I(\mathbf{x}) - c_+)^2 \; d\mathbf{x} \qquad (31)$$

where $int(\gamma)$ and $ext(\gamma)$ are the internal and external parts of the image, respectively (with respect to the contour γ).

To rewrite this expression first consider that we can write the integral over the external part as follows:



$$\int_{ext(\gamma)} (I - c_+)^2 \, d\mathbf{x} = \int_{\Omega} (I - c_+)^2 \, d\mathbf{x} - \int_{int(\gamma)} (I - c_+)^2 \, d\mathbf{x} \tag{32}$$

By inserting this to the previous equation we get:

$$E_{fit}[\gamma] = \int_{int(\gamma)} (I - c_-)^2 - (I - c_+)^2 \, d\mathbf{x} + \int_{\Omega} (I - c_+)^2 \, d\mathbf{x} \tag{33}$$

Theorem 1: Let $\gamma(s)$ be a smooth closed curve, $\gamma \in C^1([a,b],\mathbb{R}^2)$, $\gamma(a) = \gamma(b)$ and $\gamma'(a) = \gamma'(b)$.
Consider the energy $E[\gamma] = \int_{int(\gamma)} V(\mathbf{x}) \, d\mathbf{x}$, where $V : \mathbb{R}^2 \to \mathbb{R}$ is an integrable energy potential.
The variation of this energy with respect to the curve γ is given by:

$$\delta E[\gamma, w] = -\int_a^b V(\gamma(s))\mathbf{n} \cdot w(s) \, ds \tag{34}$$

Where $\mathbf{n}$ is the inward unit normal.

This theorem can be found proved in Section 2.5.1.
Thus the L^2 shape gradient is:

$$\nabla E[\gamma] = -V(\gamma(s))\mathbf{n}(s) \tag{35}$$

By combining the last three equations, where $V(\mathbf{x}) = (I(\mathbf{x}) - c_-)^2 - (I(\mathbf{x}) - c_+)^2$, we get:

$$\nabla E_{fit}[\gamma] = -\left((I - c_-)^2 - (I - c_+)^2\right)\mathbf{n} + \nabla \int_{\Omega} (I - c_+)^2 \, dx \tag{36}$$

For clarity let's just summarize what we have done so far: At time t, we first calculate c_- and c_+, we then fix c_- and c_+ and then try to optimize Γ by minimizing the energy. Since $I(\mathbf{x})$ and c_+ is fixed $\nabla \int_{\Omega} (I - c_+)^2 \, dx = 0$ such that:

$$\nabla E_{fit}[\gamma] = -\left((I - c_-)^2 + (I - c_+)^2\right)\mathbf{n} \tag{37}$$

Together with the gradient of the length we will use this term to describe the level set equation from (20) later on. But let's first consider the proof of the theorem that we just used.

2.5.1 Proof of Theorem 1

. The proof of the theorem will be based on Green's formula:

$$\int_{int(\gamma)} \frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y}\ dxdy = \int_\gamma P\ dx + \int_\gamma Q\ dy \tag{38}$$

By choosing $P(x,y) = 0$ and $Q(x,y) = \int_{x_0}^{x} V(\xi, y)\ d\xi$, we rewrite the energy as follows:

$$E[\gamma] = \int_{int(\gamma)} V(\mathbf{x})\ d\mathbf{x} = \int_{int(\gamma)} \frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y}\ dxdy = \int_\gamma Q\ dy \tag{39}$$

$$= \int_a^b Q(\gamma(s)) \cdot \gamma_2'(s)\ ds \tag{40}$$

Note $\gamma(s) = (x,y) = (\gamma_1(s), \gamma_2(s))$, $dy = \gamma_2'(s)ds$ and $\frac{\partial Q(x,y)}{\partial x} = V(x,y)$

Once again we consider an ε-variation of γ:

$$\gamma_\varepsilon(s) = \gamma(s) + \varepsilon w(s) \tag{41}$$

Here $w(s)\ :\ [a,b] \to \mathbb{R}^2$ is a parametrized curve, such that $w(a) = w(b)$ and $w'(a) = w'(b)$. Let $\gamma_\varepsilon = ((\gamma_\varepsilon')_1, (\gamma_\varepsilon')_2)$. Then the variation of $E[\gamma]$ from (40) is given by:

$$\begin{aligned}
\delta E[\gamma, w] &= \frac{d}{d\varepsilon} E[\gamma_\varepsilon]\Big|_{\varepsilon=0} \\
&= \frac{d}{d\varepsilon} \int_a^b Q(\gamma_\varepsilon) \cdot (\gamma_\varepsilon')_2\ ds\Big|_{\varepsilon=0} \\
&= \int_a^b \nabla Q(\gamma_\varepsilon \cdot w) \cdot (\gamma_\varepsilon')_2 + Q(\gamma_\varepsilon) \cdot w_2'\ ds\Big|_{\varepsilon=0} \\
&= \int_a^b \nabla Q(\gamma) \cdot w \cdot \gamma_2' + Q(\gamma) \cdot w_2'\ ds
\end{aligned} \tag{42}$$

We now want to rewrite the last part of the last equation namely $Q(\gamma) \cdot w_2'$. For this we have to consider following derivation for two smooth curves u and v in $\mathbb{R}^2$, who have periodic parametrizations such that $u(a) = u(b)$ and $u'(a) = u'(b)$ which also holds for v. By this we now consider the product rule:

$$(uv)' = u'v + uv' \tag{43}$$

By integration we get:

$$\int_a^b (uv)'\ dx = \int_a^b u'v\ dx + \int_a^b uv'\ dx \tag{44}$$

Thus we get:

$$u(b)v(b) - u(a)v(a) = \int_a^b u'v\ dx + \int_a^b uv'\ dx \tag{45}$$

Where the left hand side $u(b)v(b) - u(a)v(a) = 0$ caused by periodicity. This finally gives:

$$-\int_a^b u'v \; dx = \int_a^b uv' \; dx \tag{46}$$

So by use of this on the right term in equation (42) we get:

$$\begin{aligned}
\delta I[\gamma, w] &= \int_a^b \nabla Q(\gamma) \cdot w \cdot \gamma_2' - \frac{d}{ds} Q(\gamma) \cdot w_2 \; ds \\
&= \int_a^b \nabla Q(\gamma) \cdot w \cdot \gamma_2' - \nabla Q(\gamma) \cdot \gamma' \cdot w_2 \; ds \\
&= \int_a^b \frac{\partial Q}{\partial x} w_1 \gamma_2' + \frac{\partial Q}{\partial y} w_2 \gamma_2' - \left(\frac{\partial Q}{\partial x} \gamma_1' w_2 + \frac{\partial Q}{\partial y} \gamma_2' w_2 \right) \; ds \\
&= -\int_a^b \frac{\partial Q}{\partial x} \cdot (-\gamma_2' w_1 + \gamma_1' w_2) \; ds \\
&= -\int_a^b \frac{\partial Q}{\partial x} \cdot \mathbf{n} \cdot w \; ds \qquad (47) \\
&= -\int_a^b V(x, y) \cdot \mathbf{n} \cdot w \; ds \qquad (48)
\end{aligned}$$

Just as we claimed in the theorem.

□

2.5.2 Summary and presentation of the PDE

Let's once again consider the fitting energy for the Chan-Vese method:

$$\begin{aligned}
E(c_-, c_+, \Gamma) =& \mu \int_\Gamma d\sigma \\
&+ \lambda_1 \int_{inside(\Gamma)} (I(\mathbf{x}) - c_-)^2 d\mathbf{x} \\
&+ \lambda_2 \int_{outside(\Gamma)} (I(\mathbf{x}) - c_+)^2 d\mathbf{x}
\end{aligned} \tag{49}$$

By considering the level set equation, where $\Gamma(t) = \{\mathbf{x} \in \mathbb{R}^2 | \phi(\mathbf{x}, t) = 0\}$ we then got equation (20) from the normal velocity, which is represented here:

$$\phi_t(\mathbf{x}, t) = -\nabla E(\phi(\mathbf{x}, t)) \cdot \mathbf{n} \cdot |\nabla \phi(\mathbf{x}, t)| \tag{50}$$

To insert in this equation, we have already derived the main parts of the shape gradient $\nabla E(\phi(\mathbf{x}, t))$ so let's put them together now. From (30) we get the gradient of the length term, and from (37) we see the gradient for the fitting term. So from the fitting energy for Chan-Vese we get that the shape gradient is:

$$\nabla E[\phi(\mathbf{x}, t)] \cdot \mathbf{n}(\mathbf{x}, t) = -2\mu\kappa(\mathbf{x}, t) - \lambda_1 (I(\mathbf{x}) - c_-)^2 + \lambda_2 (I(\mathbf{x}) - c_+)^2 \tag{51}$$

Thus the following evolving equation will minimize the Chan-Vese fitting energy $E(\phi)$ for the level set equation $\phi(\mathbf{x}, t)$. Given $\phi(\mathbf{x}, 0) = \phi_0(\mathbf{x})$ for the initial contour, and the artificial time $t \geq 0$ equation (50) can be written as follows:

$$\frac{\partial}{\partial t}\phi(\mathbf{x}, t) = |\nabla\phi(\mathbf{x}, t)| \left[2\mu\kappa(\mathbf{x}, t) + \lambda_1(I(\mathbf{x}) - c_-)^2 - \lambda_2(I(\mathbf{x}) - c_+)^2\right] \tag{52}$$

With $\kappa(\mathbf{x}, t) = \nabla \cdot \left(\frac{\nabla\phi(\mathbf{x},t)}{|\nabla\phi(\mathbf{x},t)|}\right)$. And $\mu \geq 0$, $\lambda_1, \lambda_2 > 0$ are fixed parameters, c_+ and c_- are the averages of the image $I(\mathbf{x})$ outside and inside the contour respectively.

The Original Active Contours Without Edges
In the original article "Active Contours Without Edges" published by Tony F. Chan and Luminita A. Vese they present the following equation by keeping c_+ and c_- fixed, and minimizing the energy with respect to ϕ, they uses the Euler-Lagrange equation for ϕ and find:

$$\frac{\partial\phi}{\partial t} = \delta_\varepsilon(\phi)\left[\mu \text{ div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) - v + \lambda_1(I_0 - c_-)^2 - \lambda_2(I_0 - c_+)^2\right] \tag{53}$$

The main difference between this equation and the one we derived in equation (52) is that $|\nabla\phi|$ is replaced by the Dirac delta function. If we look at $|\nabla\phi|$ it is a scalar that depends only on how the level set function looks. As it could be seen on Figure 1 the $|\nabla\phi|$ is the same everywhere in the image, so it will weight the whole image equal initially. The Dirac delta is more focused on the area around the contour Γ, but since $|\nabla\phi|$ is just a scalar we decide to use the Dirac delta instead, such that we only focus on the area around the contour and not the whole image.

As it also can be seen the areal of the area inside the contour disappears and only the constant factor v remain. Therefore it is not so interesting to minimize the fitting energy with respect to the area inside the contour.

3 Numerical Approximation of Chan-Vese

To be able to use this Chan-Vese method on an image, we now have to find a numerical approximation for the model. The implementation of the Chan-Vese model in Matlab has been made with the Dirac delta function, therefore we will now consider a regularization of the Heaviside function H to be:

$$H_\varepsilon(z) = \frac{1}{2}\left(1 + \frac{2}{\pi}\arctan\left(\frac{z}{\varepsilon}\right)\right) \tag{54}$$

Since $\delta_\varepsilon = H'_\varepsilon$ it follows that the regularization of δ is:

$$\delta_\varepsilon(z) = \frac{1}{\pi}\frac{\varepsilon}{\varepsilon^2 + z^2} \tag{55}$$

for some small constant $\varepsilon > 0$. As $\varepsilon \to 0$ the approximation converges to H and δ.

The proposed Heaviside- and Dirac delta functions are here plotted with $\varepsilon = \frac{1}{\pi}$.

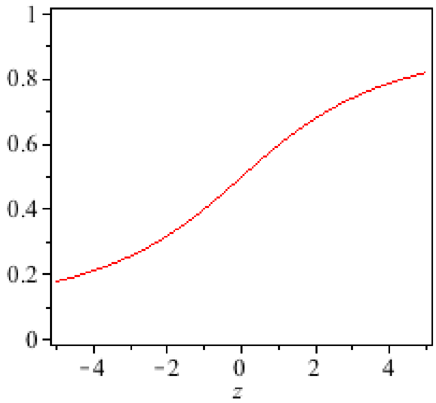


Figure 4: Regularization of the Heaviside function

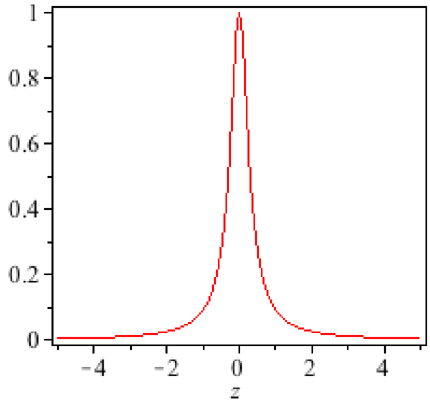


Figure 5: Regularization of the Dirac delta function

Besides the Heaviside function and the Dirac delta function we also have to discretize κ. To discretize the PDE we use finite differences. For the notation we use h as the space step and Δt as the time step. The finite difference method will be used on the curvature, which can be written in the following way:

$$\kappa(\mathbf{x}, t) = \nabla \cdot \left(\frac{\nabla \phi(\mathbf{x}, t)}{|\nabla \phi(\mathbf{x}, t)|} \right) = \left(\frac{\phi_{xx}\phi_y^2 - 2\phi_x\phi_y\phi_{xy} + \phi_{yy}\phi_x^2}{\sqrt{(\phi_x^2 + \phi_y^2)^3}} \right) \tag{56}$$

To approximate ϕ_x and ϕ_y we use the three point centred difference formula:

$$\frac{\partial \phi_{i,j}^n}{\partial x} \approx \frac{\phi_{i+1,j}^n - \phi_{i-1,j}^n}{2h_1} \tag{57}$$

$$\frac{\partial \phi_{i,j}^n}{\partial y} \approx \frac{\phi_{i,j+1}^n - \phi_{i,j-1}^n}{2h_2} \tag{58}$$

Also the three point centred difference formula for the double derivative is used:

$$\frac{\partial^2 \phi_{i,j}^n}{\partial x^2} \approx \frac{\phi_{i-1,j}^n - 2\phi_{i,j}^n + \phi_{i+1,j}^n}{(h_1)^2} \tag{59}$$

$$\frac{\partial^2 \phi_{i,j}^n}{\partial y^2} \approx \frac{\phi_{i,j-1}^n - 2\phi_{i,j}^n + \phi_{i,j+1}^n}{(h_2)^2} \tag{60}$$

The only part of the approximation we are missing so far is the mixed derivative $\frac{\partial^2 \phi^n_{i,j}}{\partial x \partial y}$, which will be derived here. Let's for convenience introduce $g(x_i) = \frac{\partial f(x_i, y_j)}{\partial y}$, such that we want to find $\frac{\partial^2 f(x_i,y_j)}{\partial x \partial y}$.

At first we use the three point centred difference on $g(x_i)$:

$$g'(x_i) \approx \frac{g(x_i + h_1) - g(x_i - h_1)}{2h_1} = \frac{\frac{\partial}{\partial y} f(x_i + h_1, y_j) - \frac{\partial}{\partial y} f(x_i - h_1, y_j)}{2h_1} \tag{61}$$

Once again we use the three point centred difference formula for the two first order derivatives:

$$\frac{\partial f(x_i + h_1, y_j)}{\partial y} = \frac{\frac{\partial}{\partial y} f(x_i + h_1, y_j + h_2) - \frac{\partial}{\partial y} f(x_i + h_1, y_j - h_2)}{2h_2} \tag{62}$$

$$\frac{\partial f(x_i - h_1, y_j)}{\partial y} = \frac{\frac{\partial}{\partial y} f(x_i - h_1, y_j + h_2) - \frac{\partial}{\partial y} f(x_i - h_1, y_j - h_2)}{2h_2} \tag{63}$$

By inserting these two equations into (61) and reintroducing the notation from earlier we then get:

$$\frac{\partial^2 f(x_i, y_j)}{\partial x \partial y} \approx \frac{f_{i+1,j+1} - f_{i+1,j-1} - f_{i-1,j+1} + f_{i-1,j-1}}{4h_1 h_2} \tag{64}$$

For the approximation on ϕ we set $h_1 = h_2 = h$ such that the mixed derivative for the level set equation becomes:

$$\frac{\partial^2 \phi^n_{i,j}}{\partial x \partial y} \approx \frac{\phi^n_{i+1,j+1} - \phi^n_{i+1,j-1} - \phi^n_{i-1,j+1} + \phi^n_{i-1,j-1}}{4h^2} \tag{65}$$

At the edges of the level set function ϕ these approximations cannot be calculated. Therefore we will only use one sided finite difference instead of the two sided to approximate here. For the mixed derivative of ϕ we have 8 special cases, one for each of the four edges and one for each corner.

Let's denote the resulting finite-difference approximation to the curvature by κ_h. Then the discretization of the PDE will be written as:

$$\frac{\phi^{n+1}_{i,j} - \phi^n_{i,j}}{\Delta t} = \delta_h(\phi^n_{i,j}) \left[\mu \kappa_h - \lambda_1 (I_{i,j} - c_-(\phi^n))^2 + \lambda_2 (I_{i,j} - c_+(\phi^n))^2 \right] \tag{66}$$

This system is then solved iteratively by computing ϕ^{n+1} from the known ϕ^n.

3.1 Reinitialization of the level set function

The first program written in this project with the Chan-Vese model didn't have this reinitialization of the level set function $\phi(\mathbf{x})$, which meant that the speed the contour was moving towards the object with was decreasing very much after a few calculations. Another thing to notice is that the model without the reinitialization of the level set function did make many more interior contours than the one without. The reinitializing of the level set function ϕ will keep ϕ from being too flat. If ϕ is too flat the Dirac delta function would then be able to blur the method by calculating on some maybe unwanted areas. The procedure for this reinitialization of ϕ that is used here can be found in [5]. The reinitialization can be done by solving the following evolution equation:

$$\begin{cases} \psi_\tau = \text{sign}(\phi(t)) \left(1 - \sqrt{\psi_x^2 + \psi_y^2}\right) \\ \psi(0) = \phi(t) \end{cases} \tag{67}$$

We now use the scheme proposed in [5] to solve the following equation numerically:

$$\psi_{i,j}^{n+1} = \psi_{i,j}^n - \Delta\tau \ \text{sign}(\phi(t)) G(\psi_{i,j}^n) \tag{68}$$

Where

$$G(\psi_{i,j}^n) = \begin{cases} \sqrt{\max((a_+)^2, (b_-)^2) + \max((c_+)^2, (d_-)^2} - 1 & \phi(i,j,t) > 0 \\ \sqrt{\max((a_-)^2, (b_+)^2) + \max((c_-)^2, (d_+)^2} - 1 & \phi(i,j,t) < 0 \\ 0 & \phi(i,j,t) = 0 \end{cases}$$

With a, b, c and d defined as:

$$\begin{aligned} a &= \frac{\psi_{i,j} - \psi_{i-1,j}}{h} \quad , \quad a_+ = \max(a, 0) \ , \ a_- = \min(a, 0) \\ b &= \frac{\psi_{i+1,j} - \psi_{i,j}}{h} \quad , \quad b_+ = \max(b, 0) \ , \ a_- = \min(b, 0) \\ c &= \frac{\psi_{i,j} - \psi_{i,j-1}}{h} \quad , \quad c_+ = \max(c, 0) \ , \ c_- = \min(c, 0) \\ a &= \frac{\psi_{i,j+1} - \psi_{i,j}}{h} \quad , \quad d_+ = \max(d, 0) \ , \ d_- = \min(d, 0) \end{aligned}$$

The stopping criterion that is used for this iteration is:

$$Q = \frac{\sum_{|\phi_{i,j}^m| < \alpha} |\phi_{i,j}^{m+1} - \phi_{i,j}^m|}{M} < \Delta\tau h^2 \tag{69}$$

where M is the number of grid points where $|\phi_{i,j}^m| < \alpha$. In our implementation we have $h = 1$ and $\Delta\tau = 0.5$.

3.2 The algorithm

Let's now summarize this text by considering the principal steps of the Chan-Vese method which is also given in [1]:

1. Initialize $\phi(\mathbf{x}, 0)$
2. Compute c_- and c_+
3. Solve the level set equation from (53) which is discretized in (66).
4. Reinitialize ϕ
5. Check if the solution is stationary, if it is then stop, else run again from point 2.

This algorithm is in this project implemented in Matlab version 7.9.0 (R2009b), so lets now focus a bit more on this implementation.

3.3 The implementation of Chan-Vese method

One of the more challenging parts of the project has been to implement the Chan-Vese method. As mentioned the implementation of the method has been done in Matlab and together with this report a CD with the program is attached. The program consists of the following files:

```
Demo.m
chanvese.m
curvature.m
reinitialization.m
stop.m
```

Inside the `Demo.m` file there can be found two different methods to use the Chan-Vese method. The first is taking a `.jpg` file as input, the other takes a sequence of images saved as an `.tif` file as input. The file `chanvese.m` is the main file, with the algorithm from Section 3.2 and can be called by:

$\phi = \text{chanvese}(I, n, \mu, \lambda_1, \lambda_2, ImageSize)$

The `chanvese.m` returns the level set function after n iterations or whenever the user decides to stop the iteration by pressing the "OK" button in the pop-up window. The inputs to `chanvese.m` is:

- I is the input image in RGB colour or grey colour (If it is RGB it would be automatically changed to grey colour).
- n is the maximum number of iterations of the algorithm.
- $\mu, \lambda_1, \lambda_2$ are the parameters for the Chan-Vese method.
- ImageSize is the size of the longest edge the image should be rescaled to (recommended to be below 500 pixels)

If `chanvese.m` is called only with the image I as input the other parameters are initially set to be: $n = 300$, $\mu = 1$, $\lambda_1 = 1$, $\lambda_2 = 1$, $ImageSize = 250$.

When the `chanvese.m` is called the first it asks for is an initial contour. This is manually selected by selecting some points on the image by use of the cursor, when pressing the enter button on the keyboard, the program proceeds the initial contour from the points and starts iterating. As mentioned before the iteration procedure can be stopped using the "OK" button, when the user thinks the contour fits well to the image.

The recommendation of the image size to be below 500×500 pixels occur since the complexity of the method in asymptotic notation is $\mathrm{O}(N \cdot M)$, where $N \times M$ is the size of the image. But it would of course be possible to run it with larger images, the calculations would just take more time.

The rest of the files follows here by a short description:

curvature.m Calculates the curvature for the level set function by use of the numerical approximation in Section 3

reinitialization.m Reinitialize the level set function using the method in Section 3.1

stop.m Makes the stop box that one could press to stop the iterations.

4 Test results

Now that we have considered how the method is implemented in Matlab it is time to check how the algorithm performs. All the tests are performed on a MacBook Pro with 2.4 GHz Intel Core 2 Duo processor and 4 GB DDR3 RAM.

So the first test run was to test if the program works at all. Therefore we consider this simple 200×200 pixel image with a sharp edged black dot on a white background. We set μ , λ_1 , λ_2 all equal to one.

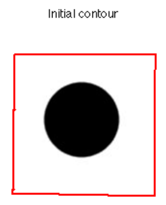


Figure 6: Initial curve on the image

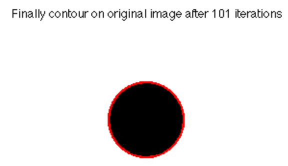


Figure 7: After 101 iterations

On Figure 6 we see the initial contour which is a bit away from the object, it took the algorithm 101 iterations to give the contour at Figure 7 and in real time it took only 4,6 seconds to calculate and show the plots (one plot for very fifth iteration).

The second test that we will consider is to test how the method interacts with blurred areas. Together with the test for blurred objects we test how the change of the parameters λ_1 and λ_2 effects the final contour. For all the three tests bellow the parameter μ is set to one.

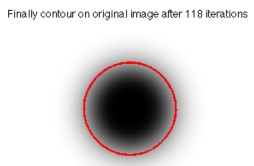


Figure 8: $\lambda_1 = 1$, $\lambda_2 = 1$

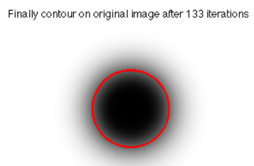


Figure 9: $\lambda_1 = 70$, $\lambda_2 = 1$

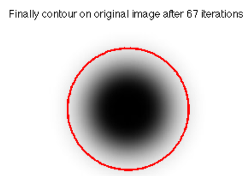


Figure 10: $\lambda_1 = 1$, $\lambda_2 = 70$

Figure 8 shows the final contour after 118 iterations from an initial contour as in Figure 6 and with parameters $\lambda_1 = 1$, $\lambda_2 = 1$. The final contour lies more or less in the middle of the blurred area. In Section 2.3 we discussed that if we wanted a more uniform foreground (the object) than background we should set $\lambda_1 > \lambda_2$, so by setting $\lambda_1 = 70$, $\lambda_2 = 1$ we get the final contour as shown in Figure 9. So on this figure we can actually see that it worked like we expected it to do, the object inside the contour is a lot more uniformly distributed than the area outside the contour.

If we on the other hand would rather have a uniform background and a more noisy object inside the contour we set $\lambda_1 < \lambda_2$. That is done in Figure 10 where

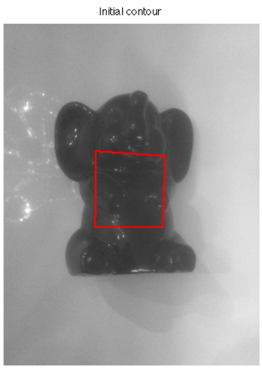
$\lambda_1 = 1,\ \lambda_2 = 70$. The figure shows that there is nearly nothing else than a white area outside the contour. So it can be seen that the theory about changing the weights λ_1 and λ_2 for the fitting terms actually works in practice.

The next thing that will be considered is the consequence of where we place the initial contour. In the previous examples we placed the initial contour outside the object, but it is actually possible to place the contour inside the object too. To see what difference it would make, a test where the initial contour was outside the object and one where the initial contour was inside the object has been made on the same picture. At the same time we test the method on a "real" photo. The parameters are: $\mu = 1\ ,\ \lambda_1 = 1\ ,\ \lambda_2 = 1$

$\Rightarrow$

Figure 11: Initial contour outside the object

Figure 12: Final contour



$\Rightarrow$

Figure 13: Initial contour inside the object

Figure 14: Final contour

As it can be seen the main difference in the two final contours is that the last one with the initial contour inside the object achieve more than one contour. When the contour evolves from the inside of the object and out, it goes over some light spots on the object and make a new closed curve around that spot as it pass by. Where on the other hand the first case where the initial contour is outside the object, these white dots are not detected. In this case we are not interested in the white dots, so to get rid of them we increased μ from 1 to 50, the method then still detected the white dots but after some time the curve around them would disappear.

This test actually showed that the contour is able to split into several contours, so let's now test that on an image with two objects:

The fixed parameters are set as follows: $\mu = 1$, $\lambda_1 = 1$, $\lambda_2 = 1$.

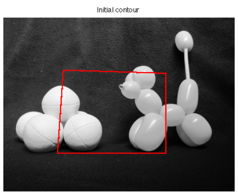


Figure 15

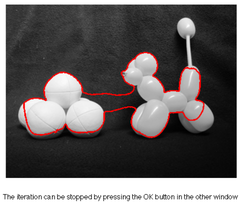


Figure 16

Figure 17

On photos with two objects it is important that the initial contour contains a part of each object. In Figure 15 such an example of an initial contour is presented. In Figure 16 we see that the evolving curve are ready to split into two in the middle of the image. In Figure 17 we see the final contour on the original image in RGB-colours. Even though the final contour is printed on the colour image, the contour was originally modelled to the grey image.

On the image it can be seen that one leg of the dog and the button of the balls that has a shadow are not contained in the final contour. By increasing λ_2 to 10 we can make a bit better fit but it is not possible to get the whole object inside the contour.

So what we see is that images with shadows can be problematic to segmentation. Let's try to use the method on an image with the same scene but this time without shadows, simply by changing the angle from which the photo is taken:

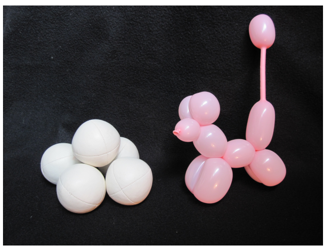


Figure 18: Image without shadows

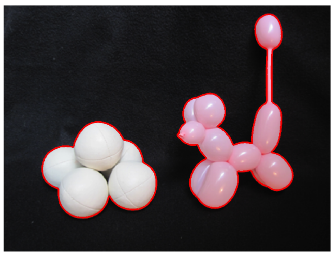


Figure 19: Final contour

As it can be seen the contour fits the objects really good on this image. So by the knowledge that we have from all our examples and experiments so far, we are now equipped to consider the problem that was the basis for considering the Chan-Vese method, namely the segmentation of fluorescent microscopy images of living cells.

4.1 Segmentation of fluorescent microscopy images of living cells

As written in the motivation the overall purpose with this project is to identify the geometry of the cell from the FLIP images. BMB at SDU has provided us with a series of images, the images are all photos of the same cell but at different times. The first image in the series looks as follows:

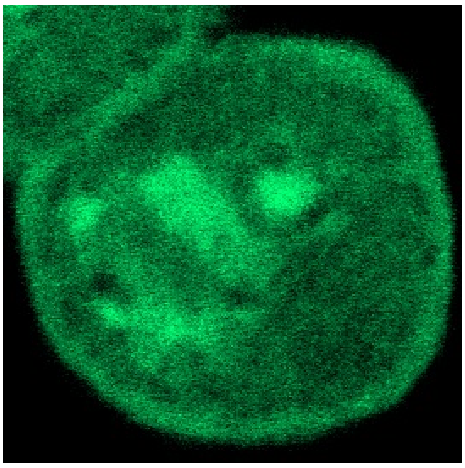


Figure 20: FLIP image

It can be seen that the background is black everywhere on the image, therefore we might want to set $\lambda_2 > \lambda_1$ as discussed in the end of Section 2.3. Furthermore it can be seen that the edge of the cell seems a bit bumpy and we want the contour as smooth as possible, so we will set the length regularization term μ to be high. So by setting the fixed parameters to be $\mu = 100$, $\lambda_1 = 1$ and $\lambda_2 = 5$ we get the following contour on the image:

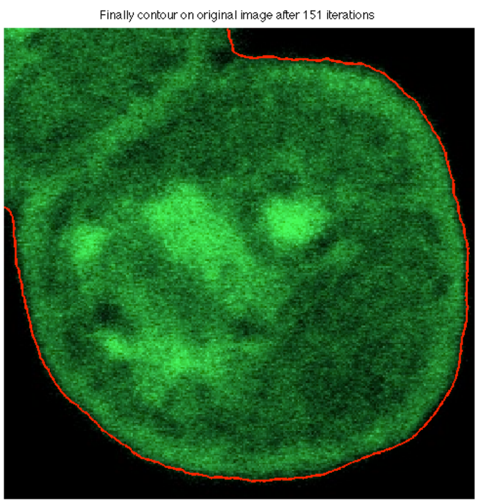


Figure 21: FLIP image with contour

As it can be seen the contour lies elegantly along the edge of the cell. Thereby we have actually achieved the purpose with this project, since the purpose was to identify and describe a functioning segmentation method of FLIP images. So by using the Chan-Vese method on the FLIP image we have got what is shown in Figure 21.

5 Discussion

As it can be seen in the previous sections most of the tests with use of the Chan-Vese method gave a successful result. As expected the method worked best on objects whose colour was in contrast to the background. There was actually performed one test, where the edges of the objects wasn't clear:

Figure 22: Grayscale image

Figure 23: Final contour

Since we wanted the contour to lie around the cars, we would consider this segmentation as bad. The bad segmentation was expected since the objects did contain a lot of the same tone as the background and does not have a clear edge.

Let's now recall the example with the black elephant on a white background, here we run the Chan-Vese model from two different initial contours and we get two different results. In the case where the initial contour was placed inside the object, the evolving contour did find different areas inside the object that was light as the background and it did then set contours around these dots. But it did only find these small dots inside the object when the contour passed over the dots, in the case where the initial contour was placed outside the object the dots wasn't spotted. That would be because of the Dirac delta function the method is implemented with, if the method was implemented with the $|\nabla\phi|$ parameter instead, it might have been able to find the white spots inside the object even though the curve did not evolve through the point.

Another thing that has been observed is that if there are two objects on the image, the initial contour has to contain a part from both objects. Because of the use of the Dirac delta function, the method only considers the part closest to the contour, but if the $|\nabla\phi|$ were used instead, then the contour might be able to split itself into two, even though the initial contour did not contain a part from both objects because $|\nabla\phi|$ kind of weigh the areas with large gradient higher.

6 Conclusion

Overall the Chan-Vese method seems to make really good segmentation of images that contain objects whose colour is in contrast to the background. So we can conclude from our tests that the method isn't good for real images where shadows can occur, but it works really good on more simple photos with approximately uniform background and/or foreground. As it could be seen on the example with the cell, the cell is itself not that uniform but because of the black background we can easily use the Chan-Vese method.

From all these examples we also see that the regularization parameters μ , λ_1 and λ_2 actually work as expected, even though the μ parameter has to be very high before something really visible happens.

We have chosen to use the Dirac delta instead of $|\nabla\phi|$ since we in the FLIP images were only interested in one contour. The implementation with this version of Chan-Vese worked very well on the FLIP images. So we have now made it possible to identify and describe the geometry of the cell by the contour from the zero level of the level set function $\phi(\mathbf{x})$.

7 Appendix

Files on the CD or in the attached .zip file. The Demo.m file is a Matlab file with examples on how to run the code:

```
Demo.m
chanvese.m
curvature.m
reinitialization.m
stop.m

Images
 -cars.jpg
 -cell.jpg
 -cells.tif
 -circle.jpg
 -dog.jpg
 -dog2.jpg
 -dot-blurred.jpg
 -dot.jpg
 -elephant.jpg
 -nutcracker.jpg
```

References

[1] T. Chan and L. Vese, "Active Contours Without Edges," IEEE transactions on image processing vol. 10, no. 2, feb. 2001, pp. 266-277

[2] Osher, S.; Sethian, J. A. (1988), "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations", J. Comput. Phys. 79: 12âĂŞ49.

[3] Siqi Chen, Guillaume Charpiat and Richard J. Radke, "Converting Level Set Gradients to Shape Gradients," Proceedings of the 11th European Conference on Computer Vision (ECCV) 2010, September 2010.

[4] Ron Goldman, "Curvature formulas for implicit curves and surfaces", Computer Aided Geometric Design, Volume 22, Issue 7, October 2005, Pages 632-658, Elsevier B.V.

[5] M. Sussman, P. Smereka and S. Osher, "A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow," Journal of Computational Physics, vol. 114 (1994), pp. 146-159.