

ACTIVE CONTOURS WITHOUT EDGES

AMALIE THERESE WILLUM HANSEN
 UNIVERSITY OF COPENHAGEN – bmz927@alumni.ku.dk

June 17, 2015

ABSTRACT

In this paper I will look into the model for active contours proposed by T.F. Chan and L.A. Vese in 2001. This model involves techniques of curve evolution, Mumford-Shah functional for segmentation and level sets. The model is able to detect objects in any given image, when the objects boundaries are not defined by a gradient, as earlier methods need them to be. There will be introduced a numerical algorithm using finite differences, implemented as an explicit scheme, whereas the Chan-Vese model is implemented as an implicit scheme. The algorithm is computed in MATLAB. There will be presented results from experiments that will serve as a validation that the algorithm works at it is supposed to.

I. INTRODUCTION

Active contours are used in the domain of image processing to locate the contour of an object. In achieving this, there are different types of models, some of which are not always applicable to any given problem. All classical active contour models rely on an edge function, which is dependent on the gradient of an image to stop the curve evolution. If an edge of an object is not defined by a gradient, these models do not apply to this given problem. One of the models worth mentioning is snakes which is a parametric curve that attempts to move into a position where its energy is minimized and its maxima is acting as an edge detector and therefore dependent on the gradient. [5] The snake model is,

$$\begin{aligned} J_1(C) = & \alpha \int_0^1 |C'(s)|^2 ds + \beta \int_0^1 |C''(s)| ds \\ & - \lambda \int_0^1 |\nabla u_0(C(s))|^2 ds \end{aligned} \quad (1)$$

where α , β and λ are positive parameters. The first two terms describe the internal energy and therefore controls the smoothness of the contour. The third and

last term describes the external energy and is the term that attracts the contour toward the object. [2]

Another model of active contours is the geometric model, which is based on geometric partial differential equations. [1] The geometric model is given by the following evolution equation,

$$\frac{\partial \phi}{\partial t} = g(|\nabla u_0|) |\nabla \phi| \left(\operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) + \nu \right) \quad \text{in } (0, \infty) \times \mathbb{R}^2 \quad (2)$$

with the initial level set function $\phi(0, x, y) = \phi_0(x, y)$ in \mathbb{R}^2 and the edgefunction $g(|\nabla u_0|)$ with $p = 2$. [2] This model is numerically stable, but it cannot retrieve edges with angles. [1]

In 2001 T. F. Chan and L. A. Vese proposed a new and improved model of active contours. The model is based on techniques of curve evolution, Mumford-Shah functional for segmentation and level sets. [2] Their model was new and improved due to the stopping term no longer needed to depend on the gradient of the image. The stopping term is instead based on segmentation of the image achieved by Mumford-Shah segmentation techniques. The model proposed in [2]

$$\begin{aligned} \frac{\partial \phi}{\partial t} = & \delta_\epsilon(\phi) \left[\mu \operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \nu - \lambda_1(u_0 - c_1)^2 \right. \\ & \left. + \lambda_2(u_0 - c_2)^2 \right] = 0 \quad \text{in } (0, \infty) \times \Omega \end{aligned} \quad (3)$$

with the initial contour $\phi(0, x, y) = \phi_0(x, y)$ and the boundary $\frac{\delta_\epsilon(\phi)}{|\nabla \phi|} \frac{\partial \phi}{\partial \vec{n}} = 0$ on $\partial \Omega$. [2] The main advantages of this model is that it can detect contours both with or without an image gradient. It also applies to both objects with smooth or discontinuous boundaries. Another superiority is that the level set formulation makes it possible to automatically detect interior contours, and in addition the initial curve can be anywhere in the given image. [2]

The Chan-Vese model differs significantly from older methods due to the fact that it does not require an image gradient, meaning that the method will also converge towards a desired outcome even with the lack of an image gradient, which is what the previous methods use as their stopping criteria. In the snakes model, knowledges of the desired contour is needed beforehand. This is understood as you cannot choose an arbitrary initial contour whilst getting a nearly perfect result.

In addition to T. F. Chan and L. A. Veses previous work, they have in 2002 drafted a new method, which is a multiphased approach to segmentation, again an improved version of former methods for active contours. For further reading [8].

II. THE CHAN-VESE MODEL

The way that T.F. Chan and L.A. Vese obtains a mathematical model for detect objects in any given image is by using techniques from already proven results. Their goal is to evolve the countour in such a way, that the algorithm stops at the boundaries in the foreground of the obejct. This is done by using the Mumford-Shah functional for segmentation and by the use of a level set method.

IDEALISATION

The energy functional from which Chan and Vese [2] have conducted their mathematical model is a minimizing level set function of ϕ , which is the function that defines the segmentation. They consider regularized versions of the Heaviside function H and the one-dimensional Dirac measure δ_0 to compute the associated Euler-Lagrange equation for the unknown function ϕ . The regularized energy functional is defined by

$$\begin{aligned} F_\epsilon(c_1, c_2, \phi) &= \mu \int_{\Omega} \delta_\epsilon(\phi(x, y)) |\nabla \phi(x, y)| dx dy \\ &+ \nu \int_{\Omega} H_\epsilon(\phi(x, y)) dx dy \\ &+ \lambda_1 \int_{\Omega} |u_0(x, y) - c_1|^2 H_\epsilon(\phi(x, y)) dx dy \\ &+ \lambda_2 \int_{\Omega} |u_0(x, y) - c_2|^2 (1 - H_\epsilon(\phi(x, y))) dx dy \end{aligned} \quad (4)$$

c_1 and c_2 are averages of the image u_0 in the regions $\phi \geq 0$ and $\phi < 0$, $\Omega \subset \mathbb{R}$ and H_ϵ and δ_ϵ are the

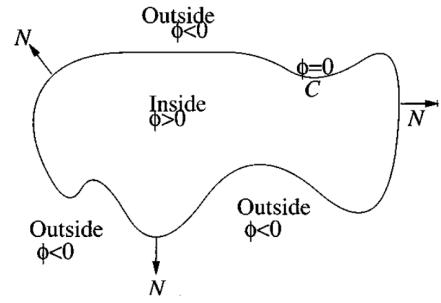


Figure 1: Curve $C = \{(x, y) : \phi(x, y) = 0\}$ propagating in the normal direction

regularized version of H and δ . c_1 and c_2 can be defined as a function of ϕ ,

$$c_1(\phi) = \frac{\int_{\Omega} u_0(x, y) H(\phi(x, y)) dx dy}{\int_{\Omega} H(\phi(x, y)) dx dy} \quad (5)$$

$$c_2(\phi) = \frac{\int_{\Omega} u_0(x, y) (1 - H(\phi(x, y))) dx dy}{\int_{\Omega} (1 - H(\phi(x, y))) dx dy} \quad (6)$$

Where the first term of (4) is the length of the edge contour also described as the mean curvature. The second term is the penalty of the total area of the foreground region. The third and the fourth terms are proportional to the variance of the image in the foreground (inside the curve) and in the background (outside the curve), respectively. The Mumford-Shah functional is achieved by setting $\nu = 0$ and $\lambda_1 = \lambda_2 = 1$.

The Heaviside function H is a step function which values is 0 for a negative argument and 1 for a positive argument. The function is defined by,

$$H(z) = \begin{cases} 1, & \text{if } z \geq 0 \\ 0, & \text{if } z < 0, \end{cases} \quad (7)$$

and the one-dimensional Dirac measure defined by,

$$\delta_0(z) = \frac{d}{dz} H(z). \quad (8)$$

From the deduced associated Euler-Lagrange equation the mathematical model is the following, with an artificial time $t \geq 0$,

$$\begin{aligned} \frac{\partial \phi}{\partial t} = \delta_\epsilon & \left[\mu \operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \nu - \lambda_1 (u_0 - c_1)^2 \right. \\ & \left. + \lambda_2 (u_0 - c_2)^2 \right] = 0 \text{ in } (0, \infty) \times \Omega, \quad (9) \\ \phi(0, x, y) &= \phi_0(x, y) \text{ in } \Omega, \\ \frac{\delta_\epsilon(\phi)}{|\nabla \phi|} \frac{\partial \phi}{\partial \vec{n}} &= 0 \text{ on } \partial \Omega \end{aligned}$$

Where ϕ_0 is the initial contour and $\frac{\partial \phi}{\partial \vec{n}}$ denotes the normal derivative of ϕ at the boundary.

DISCRETIZATION

In order to solve the partial differential equation numerically, the model has to be discretised, this is done with finite differences, where h denotes the spatial difference in pixel spacing and Δt denotes the space in the artificial time.

$$\begin{aligned} \Delta_-^x \phi_{i,j} &= \phi_{i,j} - \phi_{i-1,j}, & \Delta_+^x \phi_{i,j} &= \phi_{i+1,j} - \phi_{i,j}, \\ \Delta_-^y \phi_{i,j} &= \phi_{i,j} - \phi_{i-1,j}, & \Delta_+^y \phi_{i,j} &= \phi_{i+1,j} - \phi_{i,j}. \end{aligned}$$

For the artificial time a forward difference approximation is used. Because we know ϕ^n the first step is to compute $c_1(\phi^n)$ and $c_2(\phi^n)$ using (5) and (6) and then afterwards starting to numerically computing ϕ^{n+1} by the following discretisation and linearization of (9) in ϕ

$$\begin{aligned} \frac{\phi_{i,j}^{n+1} - \phi_{i,j}^n}{\Delta t} &= \delta_h(\phi_{i,j}^n) \left[\frac{\mu}{h^2} \Delta_-^x \right. \\ &\cdot \left(\frac{\Delta_+^x \phi_{i,j}^{n+1}}{\sqrt{(\Delta_+^x \phi_{i,j}^n)^2/(h^2) + (\phi_{i,j+1}^n - \phi_{i,j-1}^n)^2/(2h)^2}} \right) \\ &+ \frac{\mu}{h^2} \Delta_-^y \\ &\cdot \left(\frac{\Delta_+^y \phi_{i,j}^{n+1}}{\sqrt{(\phi_{i+1,j}^n - \phi_{i-1,j}^n)^2/(2h)^2 + (\Delta_+^y \phi_{i,j}^n)^2/(h^2)}} \right) \\ &\left. - \nu - \lambda_1 (u_{0,i,j} - c_1(\phi^n))^2 + \lambda_2 (u_{0,i,j} - c_2(\phi^n))^2 \right]. \end{aligned} \quad (10)$$

with the discretized versions of the Heaviside function H and the one-dimensional Dirac measure δ_0 . The versions used of these functions are regularized, [2]

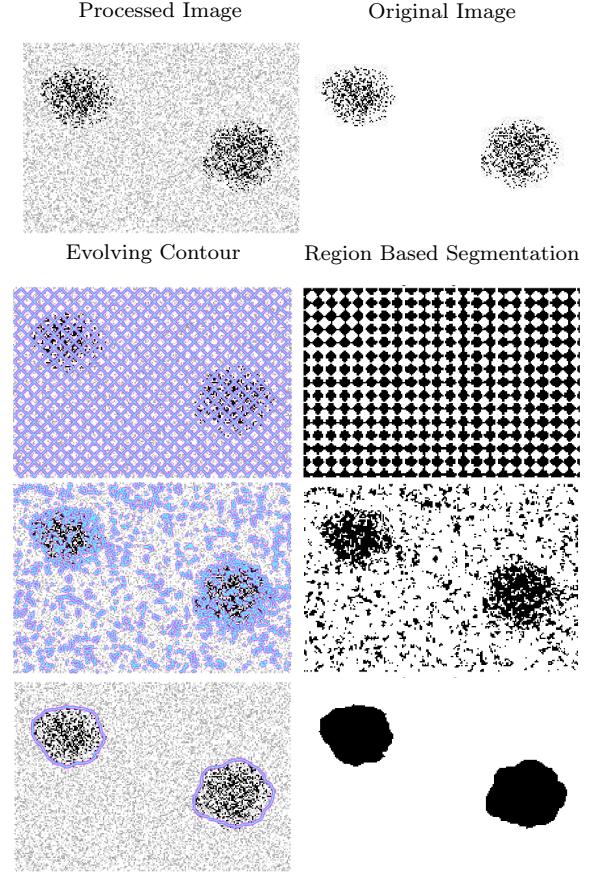


Figure 2: Detecting a cluster formation in an image applied with Gaussian white noise filter. Image size 209×143 pixels, $\phi_0(x, y) = \sin(\frac{x\pi}{5}) + \sin(\frac{y\pi}{5})$, $\mu = 2$, CPU = 131.0 s.

$$H_\epsilon(z) = \frac{1}{2} \left(1 + \frac{2}{\pi} \arctan \left(\frac{z}{\epsilon} \right) \right) \quad (11)$$

and

$$\delta_\epsilon(z) = \frac{1}{\pi \epsilon \left(1 + \frac{z^2}{\epsilon^2} \right)} \quad (12)$$

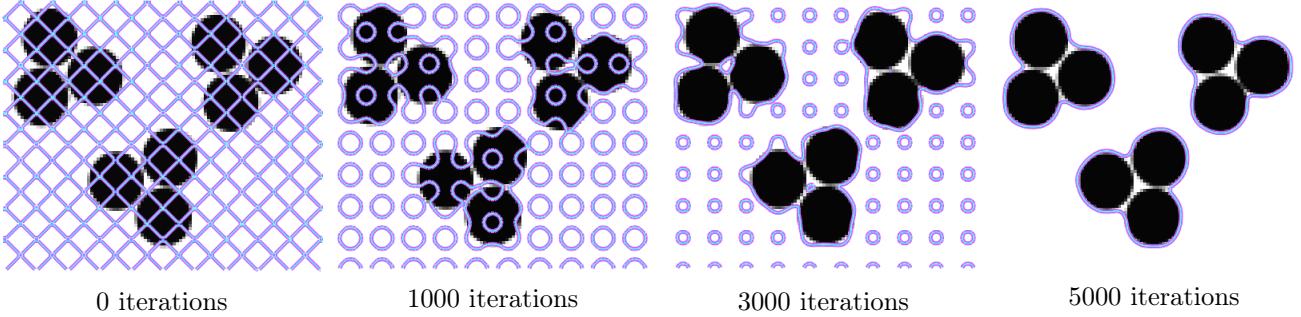
with $\epsilon = 1$.

III. IMPLEMENTATION

In the implementation done by Chan-Vese, they have used an implicit scheme where they solve a system of equations, in this way they achieve a more stable method, however there are more calculations per iteration.

We have implemented an explicit scheme [4]. This scheme is easier to implement. The challenge lies in finding an appropriate numerical approximation of $\nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right)$, which is done with first order centered differences in

Figure 3: Detection of groups of objects. Image size 102×86 pixels, $\phi_0(x, y) = \sin\left(\frac{x\pi}{5}\right) + \sin\left(\frac{y\pi}{5}\right)$, $\mu = 2$, CPU = 13.5 s.



the first derivatives and second order centered differences in the second order derivatives of ϕ – this will be approximated to κ .

The result of the approximation is,

$$\kappa = \frac{\phi_x^2 \phi_{yy} + \phi_y^2 \phi_{xx} - 2\phi_{xy}\phi_x\phi_y}{(\phi_x^2 + \phi_y^2)^{3/2}}$$

Because we have implemented an explicit scheme instead of an implicit one, our method is not as stable, therefore we have to be more careful when choosing the values for Δt , thus we have a CFL condition there must hold at all times,

$$\Delta t \leq \frac{h}{2\kappa_{max}} \quad (13)$$

where $\kappa_{max} = \frac{1}{h}$. In our implementation we have a fixed $h = 0.1$, which trivially implies the following CFL condition,

$$\Delta t \leq \frac{h}{2h} = \frac{0.1}{20} = 0.005. \quad (14)$$

Therefore a valid value for Δt would be $\Delta t = 0.001$.

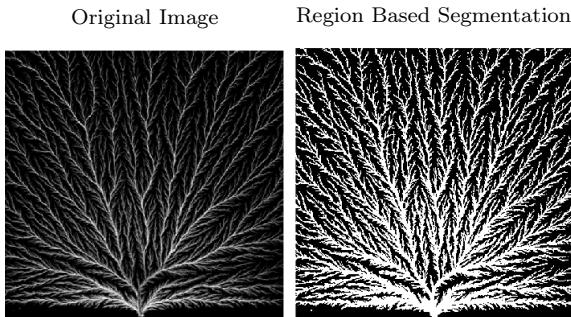


Figure 4: Segmentation of a Lichtenberg fractal. Image size 482×464 pixels, $\phi_0(x, y) = \sin\left(\frac{x\pi}{5}\right) + \sin\left(\frac{y\pi}{5}\right)$, $\mu = 0.2$, CPU = 183.9 s.

IV. EXPERIMENTAL RESULTS

Experiments enable visual overview, facilitating verification and validation of the mathematical and discrete model. Multiple experiments are there appreciated.

VARIABLES AND PARAMETERS USED IN THE EXPERIMENTS

The independent variables are the parameters the algorithm are given. The parameters can be changed according to the given image, this helps in achieving a wanted result.

The independent variable is as follows; the initial contour C_0 , the image u_0 , the weight between the foreground and the background, λ_1 and λ_2 , respectively, and the penalty of the mean curvature, μ , and the penalty of the area, ν .

In all the experiments I will use the values $\lambda_1 = \lambda_2 = \lambda = 1$. The initial contour is determined by one of the two following formulae

$$\phi_0(x, y) = \sqrt{\left(x - \frac{\text{width}(u_0)}{2}\right)^2 + \left(y - \frac{\text{height}(u_0)}{2}\right)^2} - r$$

or

$$\phi_0(x, y) = \sin\left(\frac{x\pi}{5}\right) + \sin\left(\frac{y\pi}{5}\right),$$

where the first formula is a circle defined from pixels of width and height of the picture and with the radius, r . The second formula is a grid over the entire picture. The parameter, μ , can be adjusted to the object we are looking after. e.g. if we want to detect a cluster formation in a noisy image, you should choose a larger μ . If one wants to detect many small objects in an image one should choose a small μ .

The dependent variable for the experiments is $\phi(x, y)$, and this is the value the segmentation is formed by.

Some other factors in the experiments are the constants c_1 and c_2 which describes what are inside or outside the given object, respectively. These constants are repeatedly estimated during the algorithm. If the original image u_0 is only black and white they will have the values $c_1 = 1$ and $c_2 = 0$, depending on which area there is black and white.

TESTS

A good way to start is to run some tests, to see whether or not the algorithm runs as it theoretically should.

Theoretically we should be able to find an object without the regularization term on a black and white image with a fairly good result.

If we only look at the regularization term on a blank picture – this could be both a completely black or a completely white image – the parameter, μ , would affect the contour in such a way that the length of the contour decreases.

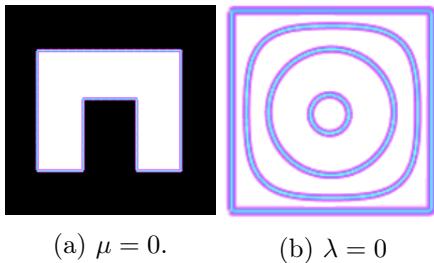


Figure 5: Visualization test of the algorithm with the two significant terms.

Figure 5a is showing a run on an black and white image with $\mu = 0$ (canceling out the regularization term) and fixed c_1 and c_2 , equal to 1 and 0, respectively. The figure shows that the result is as expected. This is done quickly and is easily implemented.

Figure 5b shows how the contour is evolving when only using the regularization term – this is achieved by setting $\lambda = 0$, thus canceling out the last term of the discretization (10). The first curve in the figure is the initial contour, $\phi_0(x, y)$, the curves shown on the inside, is μ attempting to decrease the length of the contour.

EXPERIMENTS

Due to successful tests, we now know that both of the terms in the discretization is working as desired. The next step is to do experiments on how the independent variables work on the outcome – the dependent variable ϕ . Following variables will be fixed throughout all experiments unless other is stated, $\lambda_1 = \lambda_2 = \lambda = 1$, $\nu = 0$, $\Delta t = 0.001$, thus comply with the CFL condition in (13) and (14). Furthermore, there has been made no reinitialization.

Processed Image

Original Image

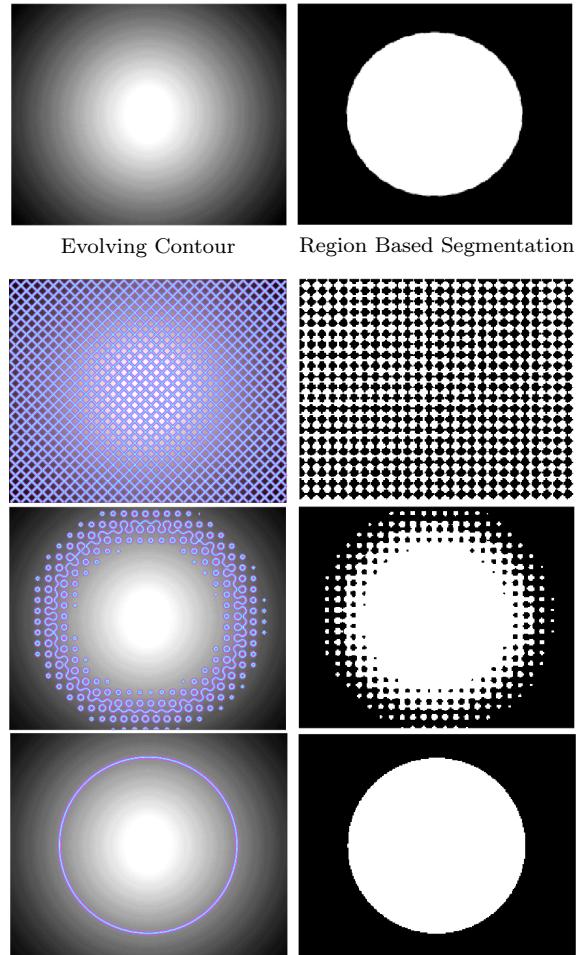


Figure 6: Circle applied with a Gaussian lowpass filter with standard deviation, $\sigma = 10$. Image size 256×206 pixels, $\phi_0(x, y) = \sin\left(\frac{x\pi}{5}\right) + \sin\left(\frac{y\pi}{5}\right)$, $\mu = 0.2$, CPU = 156.2 s.

EXPERIMENT 1

As mentioned in the Test section, a varying μ should produce a different outcome depending on the size of this independent variable.

Also mentioned earlier, if we want to detect many small objects in an image, μ should be relatively small too, e.g. a small $\mu = 0.2$ (because of normalization, otherwise it would be $\mu = 0.2 \cdot 255^2$).

On the contrary, if we wanted to detect e.g. a cluster formation in an image full of tiny objects, μ should be given a relatively high value, $\mu = 2$ (because of normalization, otherwise it would be $\mu = 2 \cdot 255^2$).

The reason for this is that μ controls the length of the contour, and the larger μ is, it is equivalent more powerfull, thus the lenght will be shorter, than if μ is small. And yet there is still a limit of how large or small you can choose μ to be.

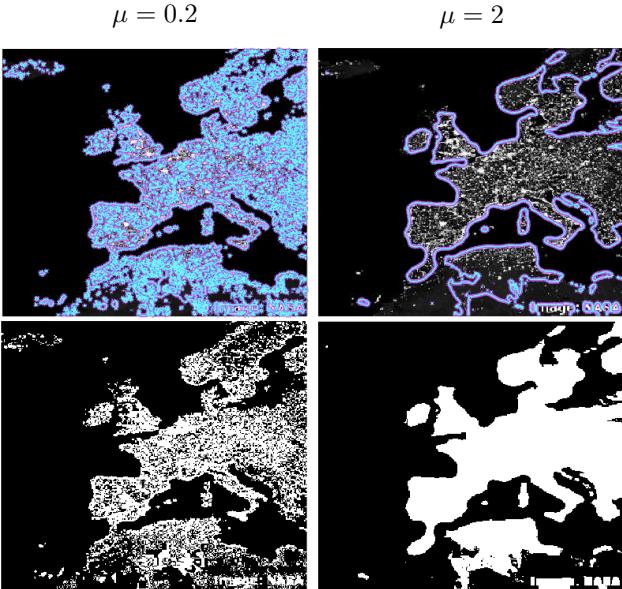


Figure 7: Europe with varying μ . Image size 320×281 pixels, $\phi_0(x, y) = \sin\left(\frac{x\pi}{5}\right) + \sin\left(\frac{y\pi}{5}\right)$.

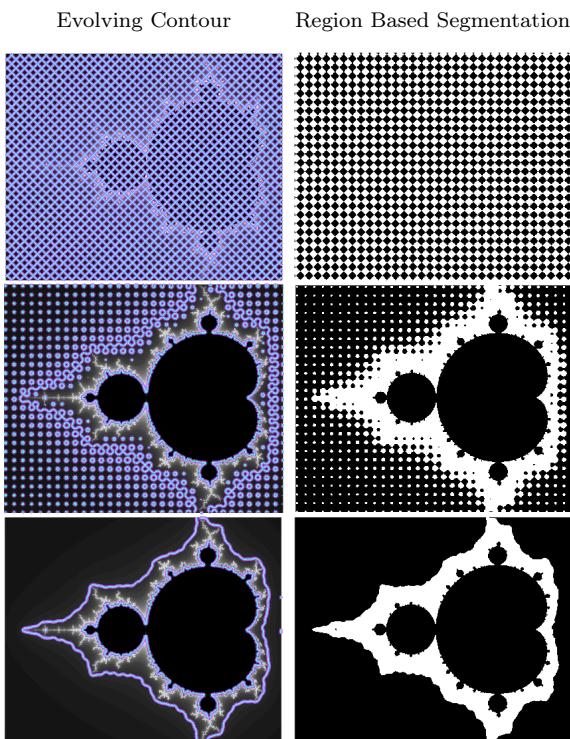


Figure 8: Mandelbrot fractal at 0, 12,000 and 65,000 iterations. Image size 671×549 pixels, $\phi_0(x, y) = \sin\left(\frac{x\pi}{5}\right) + \sin\left(\frac{y\pi}{5}\right)$, $\mu = 1$, CPU ≈ 6000 s

EXPERIMENT 2

In addition from previous methods for region based segmentation, Chan-Vese have implemented an independent variable or parameter in their algorithm, to control the area of the region inside the contour, ν . This parameter is often fixed as $\nu = 0$, but in theory the larger value ν is taking, the smaller the area of the

contour will be. ν should always be zero or a positive value, since theoretically if ν is too small it will make the area of the contour larger, and this is rarely what is desired when detecting an object, as the contour cannot get close enough to the object in the given image.

EXPERIMENT 3

If we wish to challenge the algorithm a bit, we can apply synthetic noise or a smoothing function to a picture with sharp edges.

A Gaussian white noise filter with mean, $\mu = 0$ and variance, $\sigma^2 = 0.01$ can be applied [9]. The Gaussian white noise is a normal distribution, thus a good approximation to "real-world" noise from e.g. background. This filter can be applied to an image with a simple MATLAB function. The implemented algorithm should be able to detect an object in any given image, with this kind of noise added.

A Gaussian smoothing function is likewise a challenge for the algorithm as a result of the blur fewer edges are detected [7]. This can also be applied to an image with MATLAB that takes a standard deviation, σ , as argument. Default value is $\sigma = 0.5$. The value of σ can be changed according to the amount of blur we whishes the given image to have.

Gaussian smoothing function is reducing the high-frequency components in the image, thus it is a lowpass filter.

The reason why we apply a filter instead of just using a tool such as paint to draw free-hand on to the image, is that the experiment would no longer be a quantitative experiment, instead it could be assumed to be both subjective and selective.

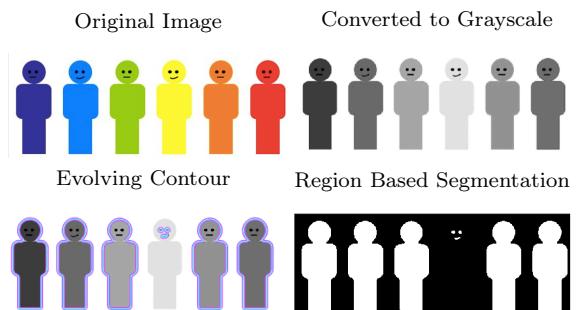


Figure 9: Detecting objects with different intensities. Image size 576×215 pixels, $\phi_0(x, y) = \sin\left(\frac{x\pi}{5}\right) + \sin\left(\frac{y\pi}{5}\right)$, $\mu = 1$, CPU = 29.9 s.

EXPERIMENT 4

The fourth experiments is a experiment revolving all other experiments. This experiment will validate the previous experiments. Although we can determine whether or not a segmentation is close to the original image, visually, there is numerous ways to calculate how similar data set are. This way to validate is a quantitave measure instead of a subjective matter measured visually. If we did not include this experiment, the validation would only rely on empiricism.

The first way to calculate the similarity for our sets – the original image and the segmented image, respectively – is to calculate the Jaccard coefficient [6],

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}, \quad (15)$$

where $J(A, B)$ is the similarity between the original image and the segmented image and lies in $0 \leq J(A, B) \leq 1$. If $A = B \Rightarrow J(A, B) = 1$, meaning the sets A and B are the same with 100%. This is also the case if both A and B are empty sets.

Another way to compare the two images is through the Dice coefficient[3], which resembles the Jaccard coefficient,

$$D(A, B) = \frac{2|A \cap B|}{|A| + |B|} \quad (16)$$

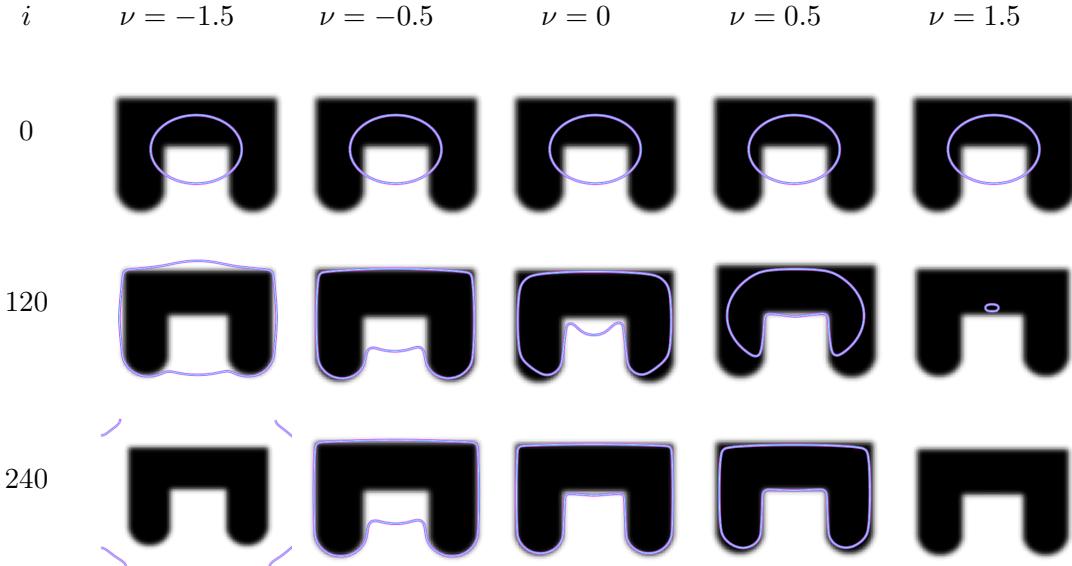


Table 1: The effect of ν , with artificial timestep $\Delta t = 0.01$, picture size 166×124 , $\phi_0(x, y) = \sqrt{(x - 83)^2 + (y - 62)^2} - 0.2$ and $\mu = 2$.

where $D(A, B)$ is the similarity between the original image and the segmented image and lies in $0 \leq D(A, B) \leq 1$.

Equivalence between the Dice and Jaccard coeffiecient is as follows

$$J(A, B) = \frac{D(A, B)}{(2 - D(A, B))}$$

$$D(A, B) = \frac{2 \cdot J(A, B)}{(1 - J(A, B))}.$$

The Dice coefficient tends to give a, somewhat, more satisfying result as it gives twice the weight to the indices both A and B have in common.

RESULTS

This section will function as a conclusion of numerical results obtained from the implemented model.

In Figure 2 I show how the model operates on an image with synthetic noise applied by a gaussian filter with mean, $\mu = 0$ and variance, $\sigma^2 = 0.01$. We see that the algorithm has no problem in detecting the cluster formation with $\mu = 2$. This could not be done by one of the methods that relies on the boundaries to be defined

by a gradient e.g. snakes.

The application of synthetic Gaussian white noise, validate that the Chan-Vese model is applicable in a situation where other active contours based on a gradient would not detect the desired object.

The calculations is done in $\text{CPU} = 131.0\text{ s}$, with $J(A, B) = 0.925$ and $D(A, B) = 0.961$ which is a fairly good result.

In [Figure 3](#) I show that the model can detect objects defined as grouping, when having a large μ , which is making the length of the contour shorter than if μ took a small value. $J(A, B) = 0.974$ and $D(A, B) = 0.987$.

In [Figure 4](#) is shown a Lichtenberg figure, which is an approximate fractal and also a natural phenomena occurring at high-voltage breakdown such as lightning strikes.

Since fractals are nowhere differentiable, I did not expect a particularly good result. However when looking at the region based segmentation, I find that it resembles the original quite well.

Unfortunately, both the Dice and Jaccard index is not applicable to this figure, due to the way it is implemented.

In [Figure 6](#) a processed image of a circle is presented with a Gaussian lowpass filter with standard deviation, $\sigma = 10$, which is a parameter that controls the amount of blur added onto the original image. Just by a simple visual evaluation I can see that the model handles this kind of problem highly plausible.

The Jaccard and Dice indices also supports my visual validation with $J(A, B) = 0.836$ and $D(A, B) = 0.911$.

In [Figure 7](#) there is displayed map over Europe, where you can see the city lights. This is a clearly example of what the impact of the parameter, μ , is in the model.

You can easily see how the final contour differs from a small μ to a large μ . This explains that the model can detect objects with smooth boundaries, even in a noisy image.

[Figure 8](#) depicts a Mandelbrot fractal, which is a self-similarity set, thus approximation the same pattern at different scales. Also, the boundary of the set has fractal dimensions.

By the definition of a Mandelbrot fractal it is trivial that it does not have any smooth edges – this is also implied by that a fractal is nowhere differentiable – therefore the expectations would be as in [Figure 4](#). Although I find the region based segmentation to be preferable good, as the model is attempting to detect an unlimited number of edges¹, whilst the image also contain a smoothing function (not added by me).

[Figure 9](#) shows how the model works on different intensity scales. I find this example quite funny, as there is one guy left out of the region based segmentation. This is obviously because his color is too light relatively to the other intensities.

In [Table 1](#) there is depicted the effect of the regularized parameter, ν which mission functions just as μ 's. Where μ attempts to smooth the contour by shortening the lenght, ν attempts to make a tighter bound by lessening the area.

V. CONCLUSION

In this paper, my group and I have implemented the model for active contour without edges by T. F. Chan and L.A. Vese.

We have done this in MATLAB.

I have hereafter validated that the model is working as opposed to according to [2] by numerical results.

¹No matter how magnified the image of a Mandelbrot fractal is, there will always be 'new' self-similar edges.

REFERENCES

- [1] CASELLES, V., CATTÉ, F., COLL, T., AND DIBOS, F. A geometric model for active contours in image processing. *Numerische mathematik* 66, 1 (1993), 1–31.
- [2] CHAN, T. F., AND VESE, L. A. Active contours without edges. *Image processing, IEEE transactions on* 10, 2 (2001), 266–277.
- [3] DICE, L. R. Measures of the amount of ecologic association between species. *Ecology* 26, 3 (1945), 297–302.
- [4] ERLEBEN, K. Advanced finite difference methods. Lecture Note 11, 2012.
- [5] KASS, M., WITKIN, A., AND TERZOPoulos, D. Snakes: Active contour models. *International journal of computer vision* 1, 4 (1988), 321–331.
- [6] NIWATTANAKUL, S., SINGTHONGCHAI, J., NAENUDORN, E., AND WANAPU, S. Using of jaccard coefficient for keywords similarity. In *Proceedings of the International MultiConference of Engineers and Computer Scientists* (2013), vol. 1, p. 6.
- [7] SAITO, S. Approximate real inversion formulas of the gaussian convolution. *Applicable Analysis* 83, 7 (2004), 727–733.
- [8] VESE, L. A., AND CHAN, T. F. A multiphase level set framework for image segmentation using the mumford and shah model. *International journal of computer vision* 50, 3 (2002), 271–293.
- [9] WELLNER, J. A. Gaussian white noise models: some results for monotone functions. *Lecture Notes-Monograph Series* (2003), 87–104.

Figure 10: Tak for denne gang!

