Segmentation of Piecewise Constant Grayscale Images

APMA 935	Final Project
Submitted by:	Nathan King
Student #:	301222376
Submitted to:	Dr. Steve Ruuth
	Dr. Faisal Beg
Date:	April 22, 2014

Contents

1 Introduction **2**

2 Description of Segmentation Methods **3**
2.1 Active Contours Without Edges 3
2.2 Threshold Dynamics 6
2.2.1 Diffusion Generated Motion by Mean Curvature 6
2.2.2 Threshold Dynamics for Piecewise Constant Mumford–Shah 8

3 Numerical Results **9**
3.1 Chan–Vese Method 10
3.2 Threshold Dynamics 12
3.2.1 Merriman–Bence–Osher Method 12
3.2.2 Esedoḡlu–Tsai Method 17

4 Discussion **21**

References **22**

Appendix **23**

1 Introduction

Image segmentation is a crucial step in many computer vision applications. Some applications, such as face detection [1], medical image detection of tumors [2] and brake light detection [3], deal with object recognition and interpretation. A popular form of image segmentation assumes that distinct objects in an image have different and approximately constant colors or intensities. Naturally one decomposes an image domain, Ω, into approximately homogeneous regions that are separated by sharp changes in image features. Boundaries of the homogeneous regions are called "edges".

This project explores two different image segmentation techniques. The methods are based on variational models, which determine the segmentation by minimizing an appropriately chosen energy. Here we only consider the segmentation of grayscale images. Let $I_0(\mathbf{x}) : \Omega \to [0, 1]$ be an initial grayscale image and assume $I_0(\mathbf{x})$ is a bounded measurable function. To acquire a segmentation of $I_0(\mathbf{x})$, look for the minimum of the *Mumford–Shah energy* [4],

$$\min_{\substack{I(\mathbf{x}):\,\Omega\to\mathbb{R}\\ K\subseteq\Omega}} \left\{ \mathrm{MS}(I,K) := \int_{\Omega\backslash K} |\nabla I|^2 d\mathbf{x} + \mu\,\mathrm{Length}(K) + \lambda \int_{\Omega} (I - I_0)^2 d\mathbf{x} \right\}. \tag{1}$$

The minimization of the Mumford–Shah energy looks for the best "cartoon like" approximation of minimal complexity for I_0. The set K is a closed subset of Ω given by the union of a finite number of curves that represent the set of "edges" (i.e. boundaries of homogeneous regions) in the image I_0. The function I is a piecewise smooth approximation of I_0. Jumps across curves in K are allowed, but I is forced to be smooth over each region due to the Dirichlet integral over $\Omega\backslash K$.

Equation (1) is the minimization of the full Mumford–Shah energy functional, however in many image processing applications solving (1) is unnecessary. It may be known that objects in an image are not only smoothly varying, but approximately constant in color or grayscale intensity. Separating gray matter from white matter in a brain MRI image is an example where there are two approximately homogenous regions. This supplies a useful simplification of (1) where the function I will only take on two values. For such cases $I(\mathbf{x}) = c_1\mathbf{1}_\omega + c_2\mathbf{1}_{\Omega\backslash\bar{\omega}}$ and we minimize the *piecewise constant Mumford–Shah energy*

$$\min_{\substack{\omega\subseteq\Omega\\ c_1,c_2\in\mathbb{R}}} \left\{ \mathrm{E}(\omega,c_1,c_2) := \mathrm{Per}(\omega;\Omega) + \lambda \int_{\Omega} (c_1 - I_0)^2 d\mathbf{x} + \lambda \int_{\Omega\backslash\bar{\omega}} (c_2 - I_0)^2 d\mathbf{x} \right\}, \tag{2}$$

where ω defines one homogeneous region in Ω and $\Omega\backslash\bar{\omega}$ therefore gives the second region. $\mathrm{Per}(\omega;\Omega)$ denotes the perimeter around the region ω.

The energy minimized by both methods here is the piecewise constant Mumford–Shah energy (2). The methods differ however in their approach to minimize the latter energy. The first method uses a level set approach to determine a segmentation contour. The second method is inspired by work of Merriman et al. [5] on diffusion generated motion by mean curvature. Both methods are quite simple for the case of two regions, but can also be extended for more than two regions.

The project is organized as follows. A description of each method is given in section 2. The first method, detailed in section 2.1, is that of Chan and Vese [6]. A zero level set is

used to represent the boundary between two objects. A method using the entire level set function is then implemented to minimize the piecewise constant Mumford–Shah energy (2). Section 2.2 will discuss the method by Esedoḡlu and Tsai [7], which minimizes the piecewise constant Mumford–Shah energy (2) using threshold dynamics. A gradient descent flow is derived and solved by splitting the equation and determine one part via thresholding. The start of section 2.2 will discuss the Merriman–Bence–Osher (MBO) method [5] that motivates the thresholding method of Esedoḡlu and Tsai. Numerical experiments for both methods are examined in section 3. These experiments aim to illustrate the strengths and weaknesses of each method. We conclude with some practical remarks and behaviours of each method in section 4.

2 Description of Segmentation Methods

Here we detail each of the segmentation methods explored in this project. Both methods minimize the piecewise constant Mumford–Shah energy (2). We introduce the level set approach by Chan and Vese [6] first. The threshold dynamics method by Esedoḡlu and Tsai [7] is given thereafter. The methods will be referred to as the Chan–Vese method and the Esedoḡlu–Tsai method. A brief description of the MBO method will also be given inspiration for the Esedoḡlu–Tsai method. Each of the methods will be illustrated for the case of two approximately homogenous regions.

2.1 Active Contours Without Edges

Classically, image segmentation uses active contour models (or snakes) with edge detection based on $|\nabla I_0|$. The basic idea of active contour models is to evolve a contour in Ω until edges of an object are detected. Edge detection is usually imposed in terms of the size of the gradient in the intensity I_0. This assumes that there is an edge where $|\nabla I_0|$ is infinite. The latter assumption is, however, not appropriate for all images. In practice the discrete gradients are bounded, which can cause curves to pass through edges. Instead the Chan–Vese method uses an active contour model without edge detection using $|\nabla I_0|$.

For the Chan–Vese method denote $C = \partial\omega$ where ω is an open subset of I_0. Let $inside(C)$ and $outside(C)$ denote the regions ω and $\Omega\backslash\bar{\omega}$, respectively. The method is based on the minimization of the piecewise constant Mumford–Shah energy (2). The image I_0 is assumed to consist of two approximately piecewise constant intensities, where C_0 gives the boundary curve between the regions. The distinct intensities are denoted I_0^i and I_0^o, which are $inside(C_0)$ and $outside(C_0)$ respectively. Assume also that the desired object to be segmented is given by the region with approximate intensity I_0^i.

To give intuition for the Chan–Vese method consider the following "fitting" energy

$$F_1(C) + F_2(C) = \int_{inside(C)} |I_0(\mathbf{x}) - c_1|^2 \, d\mathbf{x} + \int_{outside(C)} |I_0(\mathbf{x}) - c_2|^2 \, d\mathbf{x}, \tag{3}$$

where c_1, c_2 depend on C as the averages of I_0 $inside(C)$ and $outside(C)$, respectively. Notice that the fitting energy is minimized by the boundary of the object C_0,

$$\inf_C \{F_1(C) + F_2(C)\} \approx 0 \approx F_1(C_0) + F_2(C_0). \tag{4}$$

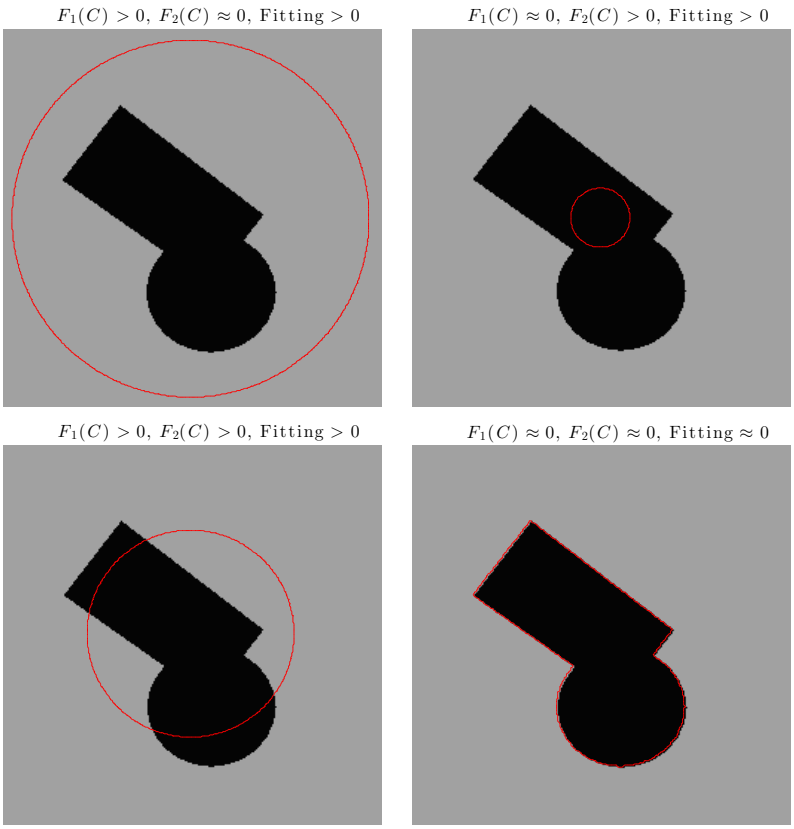


Figure 1: Illustration that the boundary of an object minimizes the fitting energy.

This can be easily seen from figure 1. If the curve C is outside the entire object then $F_1(C) > 0$ and $F_2(C) \approx 0$. The opposite occurs if C is entirely inside the object, that is $F_1(C) \approx 0$ and $F_2(C) > 0$. If the curve C is simultaneous inside and outside the object then both $F_1(C) > 0$ and $F_2(C) > 0$. Therefore the fitting energy is minimized when $C = C_0$, the boundary of the object, since there $F_1(C) \approx 0$ and $F_2(C) \approx 0$.

The energy minimized by Chan and Vese is the fitting energy (3) plus a term for regularization of the curve C. To determine the segmentation of the image I_0 we solve

$$\inf_{c_1,c_2,C} \left\{ \mu \text{ Length}(C) + \lambda_1 \int_{inside(C)} |I_0 - c_1|^2 \, d\mathbf{x} + \lambda_2 \int_{outside(C)} |I_0 - c_2|^2 \, d\mathbf{x} \right\}, \tag{5}$$

where $\mu \geq 0$, $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$ are tunable parameters. A regularization term dependant on the area inside C could also be incorporated, but is not explored here. The method by Chan and Vese formulates the energy (5) in terms of a level set function.

The level set method represents the curve $C \subset I_0$ by the zero level set of a function $\phi(\mathbf{x}) : I_0(\mathbf{x}) \to \mathbb{R}$. The level set function is defined to be negative $outside(C)$ and positive $inside(C)$. So in full we have that

$$\begin{cases} C = \partial\omega = \{\mathbf{x} \in \Omega | \phi(\mathbf{x}) = 0\}, \\ inside(C) = \omega = \{\mathbf{x} \in \Omega | \phi(\mathbf{x}) > 0\}, \\ outside(C) = \Omega \backslash \bar{\omega} = \{\mathbf{x} \in \Omega | \phi(\mathbf{x}) < 0\}. \end{cases} \tag{6}$$

Now the minimization problem (5) can be formulated using the level set function (6). The level set method is desirable here because it easily tracks the curve C.

The perimeter of a set $\mathcal{S}$ is given by $\text{Per}(\mathcal{S}) = \int |\nabla \chi(\mathcal{S})|$, where χ is the characteristic function of $\mathcal{S}$. Since the curve C is determined by the set where $\phi < 0$, the length of $C = \{\phi = 0\}$ is given by

$$\begin{aligned}\text{Length}\{\phi = 0\} &= \int_\Omega |\nabla H(\phi(\mathbf{x}))| \, d\mathbf{x}, \\ &= \int_\Omega \delta_0(\phi(\mathbf{x}))|\nabla \phi(\mathbf{x})| \, d\mathbf{x},\end{aligned}$$

where

$$H(z) = \begin{cases} 1 & \text{if } z \geq 0, \\ 0 & \text{if } z < 0, \end{cases} \qquad \delta_0(z) = \frac{d}{dz} H(z),$$

are the Heaviside function and Dirac delta measure, respectively. The other two terms in (5) can be expressed in terms of the level set function as

$$\int_{\phi > 0} |I_0(\mathbf{x}) - c_1|^2 \, d\mathbf{x} = \int_\Omega |I_0(\mathbf{x}) - c_1|^2 H(\phi(\mathbf{x})) \, d\mathbf{x},$$

and

$$\int_{\phi < 0} |I_0(\mathbf{x}) - c_2|^2 \, d\mathbf{x} = \int_\Omega |I_0(\mathbf{x}) - c_2|^2 (1 - H(\phi(\mathbf{x}))) \, d\mathbf{x}.$$

The energy to be minimize for the Chan–Vese method is therefore

$$\begin{aligned}F(c_1, c_2, \phi) := \mu \int_\Omega \delta_0(\phi(\mathbf{x}))|\nabla \phi(\mathbf{x})| \, d\mathbf{x} + \lambda_1 \int_\Omega |I_0(\mathbf{x}) - c_1|^2 H(\phi(\mathbf{x})) \, d\mathbf{x}, \\ + \lambda_2 \int_\Omega |I_0(\mathbf{x}) - c_2|^2 (1 - H(\phi(\mathbf{x}))) \, d\mathbf{x}.\end{aligned}$$

Now keep $\phi(\mathbf{x})$ fixed and minimize the energy $F(c_1, c_2, \phi)$ with respect to constants c_1 or c_2. One can then show that if C has nonempty interior in Ω (i.e. $\int_\Omega H(\phi(\mathbf{x})) \, d\mathbf{x} \neq 0$), then the optimal value of c_1 is

$$c_1(\phi) = \frac{\int_\Omega I_0(\mathbf{x}) H(\phi(\mathbf{x})) \, d\mathbf{x}}{\int_\Omega H(\phi(\mathbf{x})) \, d\mathbf{x}}.$$

Also, if C has nonempty exterior in Ω (i.e. $\int_\Omega (1 - H(\phi(\mathbf{x}))) \, d\mathbf{x} \neq 0$), then the optimal value of c_2 is

$$c_2(\phi) = \frac{\int_\Omega I_0(\mathbf{x})(1 - H(\phi(\mathbf{x}))) \, d\mathbf{x}}{\int_\Omega (1 - H(\phi(\mathbf{x}))) \, d\mathbf{x}}.$$

The latter formulas for c_1 and c_2 amount to

$$\begin{cases} c_1(\phi) = \text{mean}(I_0) & \text{in } \{\phi \geq 0\}, \\ c_2(\phi) = \text{mean}(I_0) & \text{in } \{\phi < 0\}. \end{cases} \tag{7}$$

To solve the entire minimization problem,

$$\inf_{c_1, c_2, \phi} F(c_1, c_2, \phi),$$

we minimize with respect to ϕ with the constants c_1, c_2 fixed as in (7). The corresponding Euler–Lagrange equations are derived for ϕ and gradient descent flow is used to minimize $F(c_1, c_2, \phi)$. Let $t \geq 0$ be artificial time and $\phi(t, \mathbf{x})$ define the level set function at time t. The initial contour is given by $\{\phi(0, \mathbf{x}) = 0\} = \{\phi_0(\mathbf{x}) = 0\}$ and the gradient descent flow becomes

$$\begin{aligned} &\frac{\partial \phi}{\partial t} = \delta_0(\phi) \left[\mu \, \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \lambda_1 (I_0 - c_1)^2 + \lambda_2 (I_0 - c_2)^2 \right] = 0 \quad \text{in } (0, \infty) \times \Omega, \\ &\phi(0, \mathbf{x}) = \phi_0(\mathbf{x}) \quad \text{in } \Omega, \\ &\frac{\partial \phi}{\partial \vec{n}} = 0 \quad \text{on } \partial\Omega, \end{aligned} \tag{8}$$

where $\vec{n}$ denotes the exterior normal to $\partial\Omega$ and $\partial\phi/\partial\vec{n}$ is the normal derivative of ϕ at the boundary $\partial\Omega$. Section 3 discusses the numerical implementation of (8).

2.2 Threshold Dynamics

The method proposed by Esedoḡlu and Tsai [7] is based on threshold dynamics of the piecewise constant Mumford–Shah energy (2). The piecewise constant Mumford–Shah energy (2) is minimized by gradient descent flow. The resulting gradient descent flow is split into a linear parabolic equation and an equation which is solved by thresholding. The Esedoḡlu–Tsai method was inspired by the MBO method [5]. The MBO method is recalled here first then we describe the Esedoḡlu–Tsai method.

2.2.1 Diffusion Generated Motion by Mean Curvature

The MBO method was discovered from a simple intuition. Imagine that a set of points in a plane is allowed to "diffuse". Diffusion will rapidly blunt sharp points on the boundary, but it has little impart on the flatter parts. The latter behaviour seems to correspond to some sort of curvature motion. One could expect that it is possible to create an algorithm, which repeatedly diffuses a set, "recovers" a new set and thereby generates curvature dependent motion of the boundary. Formal analysis can be done to clarify the intuition given above.

To "diffuse the set" we mean that diffusion is applied to the characteristic function of the set, χ. The goal is to determine how the set evolves when we diffuse the set. That is, analyze the diffusion equation

$$\frac{\partial \chi}{\partial t} = D \Delta \chi.$$

Now consider a point, P, of interest on the boundary of the set (see figure 2). Construct a local polar coordinate system whose origin is at the center of curvature of P. Diffusion of the set can be written in (r, θ)–coordinates as

$$\frac{\partial \chi}{\partial t} = \frac{D}{r} \frac{\partial \chi}{\partial r} + D \frac{\partial^2 \chi}{\partial^2 r}, \tag{9}$$

where the term involving $\frac{\partial^2 \chi}{\partial \theta^2} = 0$ due to local circular symmetry near P.

In the radial direction (9) is an advection–diffusion equation, with advection velocity $V = D/r$. At the point P the radius r is equal to the radius of curvature ρ, so $V = D\kappa$

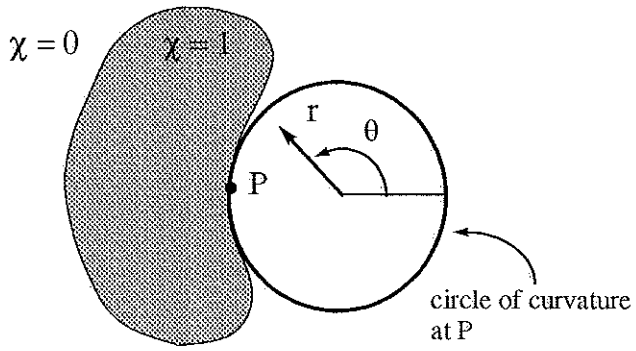


Figure 2: A point P on the boundary of a set (shaded gray region) and the circle of curvature at P.

where $\kappa = 1/\rho$ is the curvature. Hence the effect of diffusing the set is to move the boundary radially at speed $D\kappa$, while simultaneously diffusing χ radially. The diffusion does not affect the motion of the $\chi = 1/2$ level set; the $\chi = 1/2$ level set only moves radially with the advection velocity. Diffusion will therefore generate mean curvature motion at the $\chi = 1/2$ level set. One must ensure to redefine a new set before the local analysis breaks down. The new set is taken to have $\chi = 1/2$ as its boundary and diffusion is repeated to generate new motion.

The above analysis suggest algorithm 1 for evolving an intial curve σ. The only parameter

Algorithm 1 : Diffusion Generated Curvature Motion

1: Define a set with boundary σ and characteristic function χ
2: Determine $\chi(\tau)$ at time τ from

$$\frac{\partial \chi}{\partial t} = D\Delta\chi, \quad \frac{\partial \chi}{\partial \vec{n}} = 0 \text{ on } \Omega$$

3: Define the new characteristic function as

$$\chi = \begin{cases} 1 & \text{if } \chi(\tau) \geq 1/2, \\ 0 & \text{if } \chi(\tau) < 1/2, \end{cases}$$

4: Define σ as the boundary of χ.
5: Repeat steps 2-4.

to be determined is the time step τ. The step size τ must be short enough for the local analysis about to hold, but long enough for the boundary curve to move at least one spatial grid. The local analysis breaks down when information travels on the order of the local radius of curvature ρ. Therefore the requirement on the time step τ is

$$\frac{\rho}{h} \ll \frac{\tau D}{h^2} \ll \left(\frac{\rho}{h}\right)^2. \tag{10}$$

One can see that the basic idea of the MBO method is to alternate the solution of diffusion with thresholding. The MBO method can be phrased in terms of another well known technique for approximating mean curvature motion. The Allen–Cahn equation (sometimes referred to as the phase field method) can be used to interpret the MBO method. In the following formulation the MBO method resembles the Esedoḡlu–Tsai method. Consider the equation

$$I_t = 2\epsilon\Delta I - \frac{1}{\epsilon}W'(I), \tag{11}$$

where $W(\xi) : \mathbb{R} \to \mathbb{R}$ is a double well potential with equidepth wells at 0 and 1. Here we choose $W(\xi) = \xi^2(1-\xi)^2$.

If a time splitting numerical method [8] is applied to (11), then for a time step $\Delta t > 0$ the scheme alternates between two steps. First let $v(\mathbf{x}) = S_{\Delta t}(I^n(\mathbf{x}))$ where $S_{\Delta t}(\xi)$ is the solution to

$$w_t = 2\epsilon\Delta w, \quad w|_{t=0} = \xi$$

and appropriate boundary conditions. Then set $I^{n+1}(\mathbf{x}) = T_{\Delta t}(v(\mathbf{x}))$ is the solution to

$$w_t = -\frac{1}{\epsilon}W'(w), \quad w|_{t=0} = v(\mathbf{x}).$$

The second step of the time splitting is a stiff, non–linear ODE, which has two stable solutions $w = 0$, $w = 1$ and one unstable solution at $w = \frac{1}{2}$. The MBO method uses this observation, for fixed $\Delta t > 0$ and $\epsilon \to 0^+$ the solutions to the ODE can be done by thresholding. That is, for every point $\mathbf{x}$ the value of w converges to one of the stable equilibrium values depending on which basin of attraction $\mathbf{x}$ belongs to

$$\lim_{\epsilon\to 0^+} T_{\Delta t}(\xi) = \begin{cases} 0 & \text{if } \xi \in (-\infty, \frac{1}{2}), \\ 1 & \text{if } \xi \in (\frac{1}{2}, \infty). \end{cases}$$

The above time splitting scheme with this thresholding for the ODE part is the same as algorithm 1.

2.2.2 Threshold Dynamics for Piecewise Constant Mumford–Shah

Esedoḡlu and Tsai [7] proposed a thresholding scheme inspired by the MBO method. The scheme approximates the gradient descent flow for the piecewise constant Mumford–Shah energy (2). To realize the threshold dynamics for the gradient descent flow of (2) consider a phase-field approximation. The idea is to find a diffuse interface approximation of (2) to motivate the proposed dynamics (similar to how the Allen–Cahn equation motivates the MBO method).

The following sequence of energies gives a diffuse interface approximation for (2),

$$MS_\varepsilon(c_1, c_2, I) := \int_\Omega \varepsilon|\nabla I|^2 + \frac{1}{\varepsilon}W(I) + \lambda[I^2(I_0 - c_1)^2 + (1-I)^2(I_0 - c_2)^2]d\mathbf{x},$$

where $\varepsilon > 0$ and $W(\xi) = \xi^2(1-\xi)^2$. Variation of the energy $MS_\varepsilon(c_1, c_2, I)$ with respect to I provides the gradient descent flow

$$I_t = 2\varepsilon\Delta I - \frac{1}{\varepsilon}W'(I) - 2\lambda[I(I_0 - c_1)^2 + (I-1)(I_0 + c_2)^2]. \tag{12}$$

A candidate for threshold dynamics is found by time splitting (12). There are multiple ways in which (12) can be split. Here we split such that the thresholding step is the same as in the MBO method.

The time splitting results in a linear heat equation with a forcing term to be solved and thresholding to either 0 or 1. Explicitly the Esedoḡlu–Tsai method is

1. Let $v(\mathbf{x}) = S_\tau(I^n(\mathbf{x}))$, where $S_\tau(I^n(\mathbf{x}))$ is the solution to

$$\begin{aligned} &w_t = \Delta w - \frac{\lambda}{\sqrt{\pi t}}[w(I_0 - c_1)^2 + (w-1)(I_0 - c_2)^2], \\ &w|_{t=0} = I^n(\mathbf{x}), \\ &\frac{\partial w}{\partial \vec{n}} = 0 \quad \text{on} \quad \partial\Omega, \end{aligned} \tag{13}$$

   at time τ.

2. Set

$$I^{n+1}(\mathbf{x}) = \begin{cases} 0 & \text{if } v(\mathbf{x}) \in (-\infty, \frac{1}{2}], \\ 1 & \text{if } v(\mathbf{x}) \in (\frac{1}{2}, \infty). \end{cases}$$

This method is identical to the MBO method except in the forcing term of the linear heat equation. Also notice the parameter λ has been scaled in the above method as $\lambda/\sqrt{\pi t}$. Derivation of the scaling of λ is given in [7], which is done so that curvature motion moves with desired normal velocity.

The constants c_1, c_2 are determine by the mean of the set where $\{I(\mathbf{x}) = 1\}$ and $\{I(\mathbf{x}) = 0\}$, respectively. That is,

$$c_1 = \frac{\int_\Omega I(\mathbf{x}) I_0(\mathbf{x}) d\mathbf{x}}{\int_\Omega I(\mathbf{x}) d\mathbf{x}} \quad \text{and} \quad c_2 = \frac{\int_\Omega (1 - I(\mathbf{x})) I_0(\mathbf{x}) d\mathbf{x}}{\int_\Omega (1 - I(\mathbf{x})) d\mathbf{x}}.$$

The first step in the Esedoḡlu–Tsai method is a linear diffusion equation, which can be solved with many efficient numerical methods. Esedoḡlu and Tsai suggest discretizing the heat equation implicitly and solve the resulting elliptic equation with a fast method (for example plain vanilla multigrid). The Chan–Vese method results in PDEs that are singular and degenerate. Standard fast methods have trouble achieving their expected convergence rates with the Chan–Vese method. The linear diffusion equation (13) also becomes easier to solve for small Δt, since it becomes closer to being diagonally dominant. Numerical implementation in this paper does not utilize the most efficient techniques for either method. Therefore efficiency of each method will not be investigated in depth here. Numerical methods used here are described next in section 3.

3 Numerical Results

Some numerical experiments for different images are given here. Four different test images are used to compare the relative merits of the Chan–Vese method and Esedoḡlu–Tsai method. The parameter λ is the same for both methods (take $\lambda_1 = \lambda_2 = \lambda$ for Chan–Vese). Section 3.1

details the numerics for the Chan–Vese method, while section 3.2 describes numerics for the Esedoḡlu–Tsai method. Some interesting images are also segmented in the Appendix using the Esedoḡlu–Tsai method.

3.1 Chan–Vese Method

In practice, the energy and corresponding gradient descent flow for the Chan–Vese method needs to be regularized. The regularized Heaviside function, denoted H_ε, has many possibilities. The most common regularization used is

$$H_\varepsilon(z) = \begin{cases} 1 & \text{if } z \geq \varepsilon, \\ 0 & \text{if } z < -\varepsilon, \\ \frac{1}{2}\left[1 + \frac{z}{\varepsilon} + \frac{1}{\pi}\sin\left(\frac{\pi z}{\varepsilon}\right)\right] & \text{if } |z| \leq \varepsilon, \end{cases} \tag{14}$$

however another choice for H_ε is found more useful in [6]. The regularization of H used by Chan and Vese is

$$H_\varepsilon(z) = \frac{1}{2}\left[1 + \frac{2}{\pi}\arctan\left(\frac{z}{\varepsilon}\right)\right], \tag{15}$$

and the Dirac delta is then given by $\delta_\varepsilon(z) = \frac{d}{dz}H_\varepsilon(z)$. Both regularizations converge to the true H and δ_0 as $\varepsilon \to 0$.

The energy to be minimized is however non–convex, which allows many local minima. The energy is the regularized version of (2.1) given by

$$\begin{aligned} F_\varepsilon(c_1, c_2, \phi) := \mu \int_\Omega \delta_\varepsilon(\phi(\mathbf{x}))|\nabla\phi(\mathbf{x})|\, d\mathbf{x} + \lambda_1 \int_\Omega |I_0(\mathbf{x}) - c_1|^2 H_\varepsilon(\phi(\mathbf{x}))\, d\mathbf{x}, \\ + \lambda_2 \int_\Omega |I_0(\mathbf{x}) - c_2|^2 (1 - H_\varepsilon(\phi(\mathbf{x})))\, d\mathbf{x}. \end{aligned}$$

The common regularization (14) tends to find local minima, while the regularization (15) tends to find a global minimizer. Equation (15) can find a global minimizer since it is non–compactly supported, which gives Euler–Lagrange equations that act globally. This allows the model to detect interior contours or contours far from the initial contour.

The regularization of the gradient descent flow is

$$\begin{aligned} &\frac{\partial\phi}{\partial t} = \delta_\varepsilon(\phi)\left[\mu\,\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right) - \lambda_1(I_0 - c_1)^2 + \lambda_2(I_0 - c_2)^2\right] = 0 \quad \text{in } (0,\infty)\times\Omega, \\ &\phi(0,\mathbf{x}) = \phi_0(\mathbf{x}) \quad \text{in } \Omega, \\ &\frac{\partial\phi}{\partial\vec{n}} = 0 \quad \text{on } \partial\Omega \end{aligned} \tag{16}$$

and is discretized as follows. In 2D let $\mathbf{x} = (x, y)$, denote the spatial step size by h and Δt as the time step size. The grid points $(x_i, y_j) = (ih, jh)$, $1 \leq i \leq M$ and $1 \leq j \leq N$, and time is partitioned as $t^n = n\Delta t$. The forward and backward finite differences are defined as

$$\begin{aligned} \Delta^x_- \phi_{i,j} &= \phi_{i,j} - \phi_{i-1,j}, \quad & \Delta^x_+ \phi_{i+1,j} &= \phi_{i,j} - \phi_{i,j}, \\ \Delta^y_- \phi_{i,j} &= \phi_{i,j} - \phi_{i,j-1}, \quad & \Delta^y_+ \phi_{i,j+1} &= \phi_{i,j} - \phi_{i,j}. \end{aligned}$$

Approximate $\phi(t,x,y)$ at t_n, x_i and y_j as $\phi^n_{i,j} \approx \phi(n\Delta t, ih, jh)$, with $n \geq 0$ and $\phi^0 = \phi_0$ is the initial contour. At the n-th step of the algorithm, given ϕ^n from the previous step, we compute $c_1(\phi^n)$ and $c_2(\phi^n)$. The level set function ϕ^{n+1} at t^{n+1} is then computed from the following discretization of (16)

$$\begin{aligned}\frac{\phi^{n+1}_{i,j} - \phi^n_{i,j}}{\Delta t} = \delta_\varepsilon(\phi^n_{i,j}) \Bigg[& \frac{\mu}{h^2}\Delta^x_- \left(\frac{\Delta^x_+ \phi^{n+1}_{i,j}}{\sqrt{(\Delta^x_+\phi^n_{i,j})^2/(h)^2 + (\phi^n_{i,j+1} - \phi^n_{i,j-1})^2/(2h)^2}} \right) \\ & + \frac{\mu}{h^2}\Delta^y_- \left(\frac{\Delta^y_+ \phi^{n+1}_{i,j}}{\sqrt{(\phi^n_{i+1,j} - \phi^n_{i-1,j})^2/(2h)^2 + (\Delta^y_+\phi^n_{i,j})^2/(h)^2}} \right) \\ & - \lambda_1((I_0)_{i,j} - c_1(\phi^n))^2 - \lambda_2((I_0)_{i,j} - c_2(\phi^n))^2 \Bigg].\end{aligned}$$

For superior efficiency here we do not solve the gradient descent flow (16). Instead a Gauss–Seidel iteration was applied directly to the corresponding Euler–Lagrange equation discretization

$$\begin{aligned}0 = -\delta_\varepsilon(\phi^n_{i,j}) \Bigg[& \frac{\mu}{h^2}\Delta^x_- \left(\frac{\Delta^x_+ \phi^{n+1}_{i,j}}{\sqrt{(\Delta^x_+\phi^n_{i,j})^2/(h)^2 + (\phi^n_{i,j+1} - \phi^n_{i,j-1})^2/(2h)^2}} \right) \\ & + \frac{\mu}{h^2}\Delta^y_- \left(\frac{\Delta^y_+ \phi^{n+1}_{i,j}}{\sqrt{(\phi^n_{i+1,j} - \phi^n_{i-1,j})^2/(2h)^2 + (\Delta^y_+\phi^n_{i,j})^2/(h)^2}} \right) \\ & - \lambda_1((I_0)_{i,j} - c_1(\phi^n))^2 - \lambda_2((I_0)_{i,j} - c_2(\phi^n))^2 \Bigg].\end{aligned}$$

The Chan–Vese method is experimented on two artificial images and two real images. In all experiments we take $\mu = 1$, $\lambda_1 = \lambda 2 = \lambda$, $h = 1$ and initial zero level set of a circle with radius R_0. For the regularization of the Dirac delta and Heavyside function $\epsilon = 1$. Only the parameter λ is varied for each test image. Initial image I_0 is given with binary images of the segmentation at different number of iterations. The exact values of λ and ϕ_0 used will be given in each figure caption. Computation time (CT) is also given in the captions and is computed using MATLAB's tic/toc command. This command is not overly accurate, however it is used for comparison purposes.

Figure 3a is a binary image of three shapes, a triangle, donut and rectangle with an ellipse removed. Gaussian noise, with mean zero and variance 0.008, and blurring was added. The Chan–Vese method quickly wraps around the shapes after one iteration (figure 3b). The edges are bumpy due to the noise in the image and are smoothed slightly after another iteration in figure 3c.

Figure 4a is another binary image, this time consisting of just lines. The contours from the Chan–Vese method wrap around the objects in just one iteration again. The lines are completely enclosed by the contours in figure 4b, producing the binary segmentation image. There is a small dot in the center of figure 4b where the initial contour resides and two other

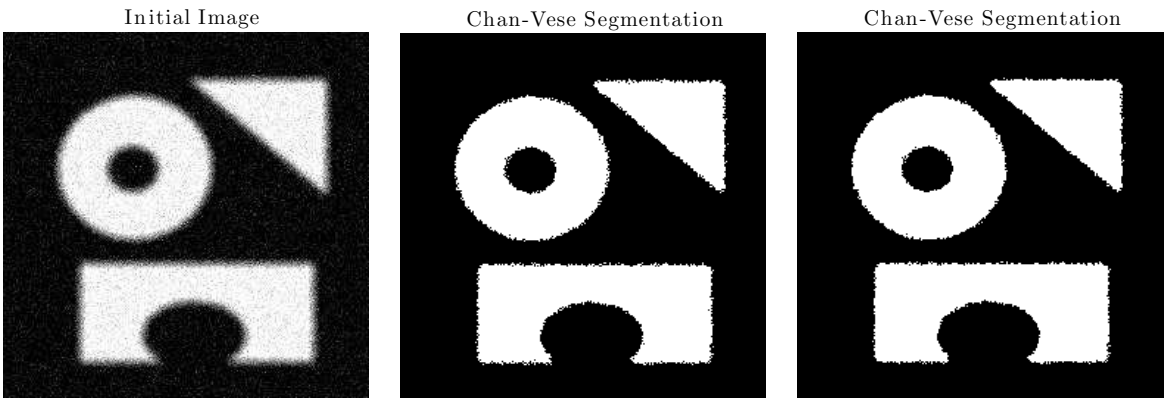


Figure 3: A 249×249 grayscale image of three noisy and blurred shapes in a). Images b) and c) are segmentations after one and two iterations, respectively. Parameter $\lambda = 0.0001$, initial contour $\phi_0(x, y) = -\sqrt{(x - 124.5)^2 + (y - 124.5)^2} + 10$ and CT = 4.87 s.

dots near the quadrilateral. After one more iteration in figure 4c the center dot has shrunk to be nearly unnoticeable and the other two have disappeared.

Figure 5a is a grayscale image of a cluster of stars. The Chan–Vese method in this case takes many more iterations to obtain the segmentation. Figure 5b is after the first iteration and there is no noticeable change from the initial guess. It then takes until the fifth iteration for stars far from the center to be segmented (figure 5c). After seven iterations, in figure 5d, the middle star cluster begins to be detailed. In figure 5e the entire star cluster can be seen after nine iterations and more details are added after eleven iterations (figure 5f).

The final image, in figure 6a, is that of a camera man seen in many image processing tests. The Chan–Vese method obtains a good segmentation after one iteration in figure 6b. The method then continues to pick up on noise–like darkness in the grass in figures 6c and d. The method is impressive in the amount of detail of the camera and the facial region. Comparison of segmentation, for each image here, using the Esedoḡlu–Tsai method will be given next.

3.2 Threshold Dynamics

Since the Esedoḡlu–Tsai method was inspired by the Merriman–Bence–Osher method, some numerical results are given first. Once the behaviour of the MBO method is realized it is easier to understand the Esedoḡlu–Tsai method. Numerical experiments of the same images in section 3.1 are given thereafter for the Esedoḡlu–Tsai method.

3.2.1 Merriman–Bence–Osher Method

We partition $\mathbf{x}$ and t the same as the Chan–Vese method. In 2D let $\mathbf{x} = (x, y)$, denote the spatial step size by h and Δt gives the time step size. For grid points $(x_i, y_j) = (ih, jh)$, $1 \leq i \leq M$ and $1 \leq j \leq N$, and for time $t^n = n\Delta t$. That is, the computed value $\chi^n_{i,j}$ is an approximation to $\chi(t^n, x_i, y_j)$. The MBO method given in algorithm 1 is implemented with

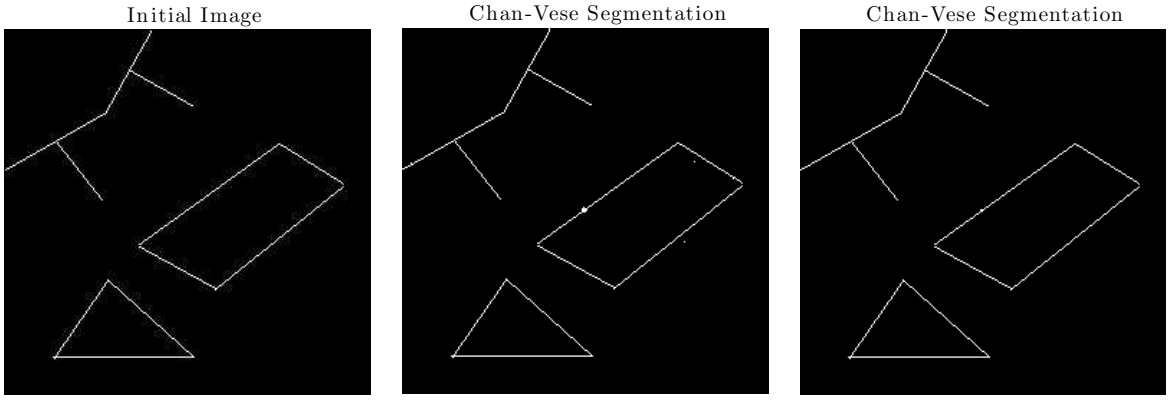


Figure 4: A 256×256 binary image of different shapes created from lines in a). Images b) and c) are segmentations after one and two iterations, respectively. Parameter value $\lambda = 0.00002$, initial contour $\phi_0(x, y) = -\sqrt{(x-128)^2 + (y-128)^2} + 2$ and CT = 5.32 s.

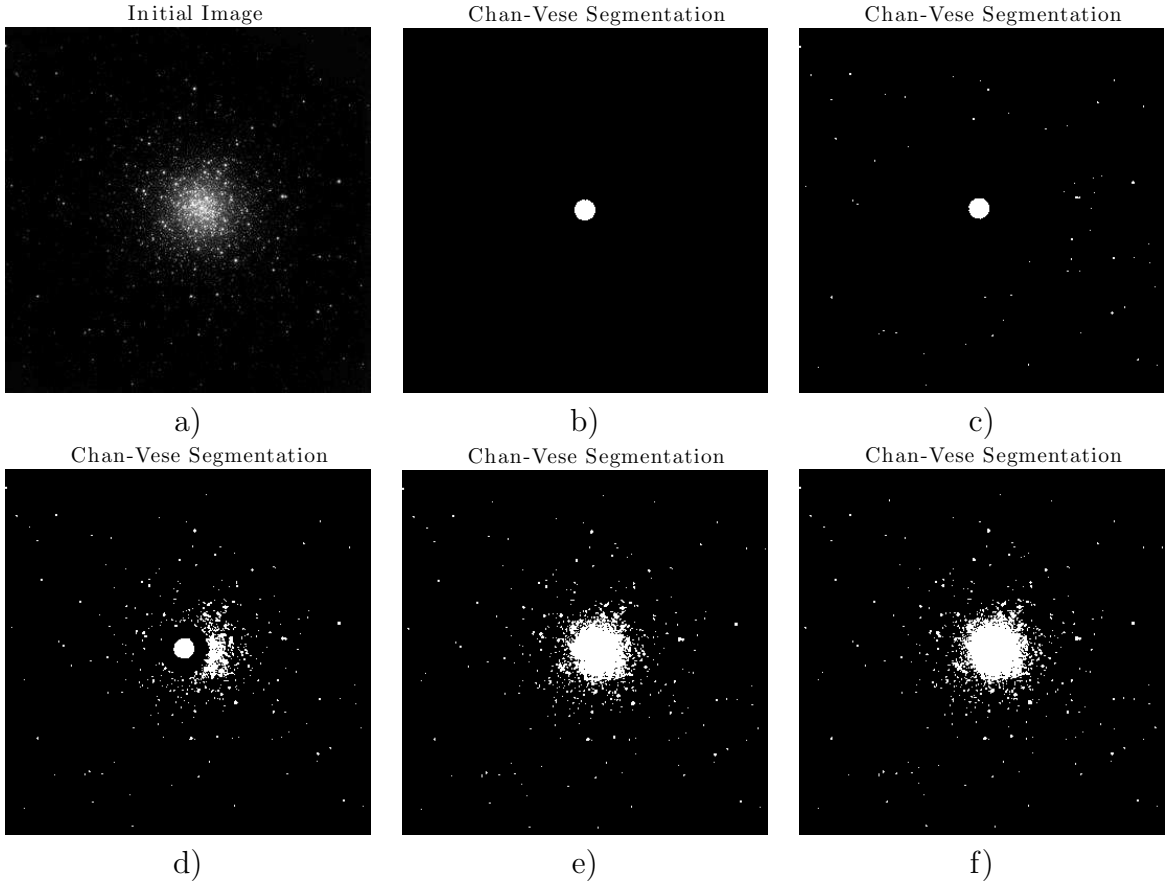


Figure 5: A 350×350 grayscale image of a star cluster in a). Images from b)–f) are after one, five, seven, nine and eleven iterations, respectively. Parameter value $\lambda = 0.00005$, initial contour $\phi_0(x, y) = -\sqrt{(x-175)^2 + (y-175)^2} + 10$ and CT = 41.01 s. Source: www.astro.caltech.edu/palomar/images/jarrett_M3_med.jpg

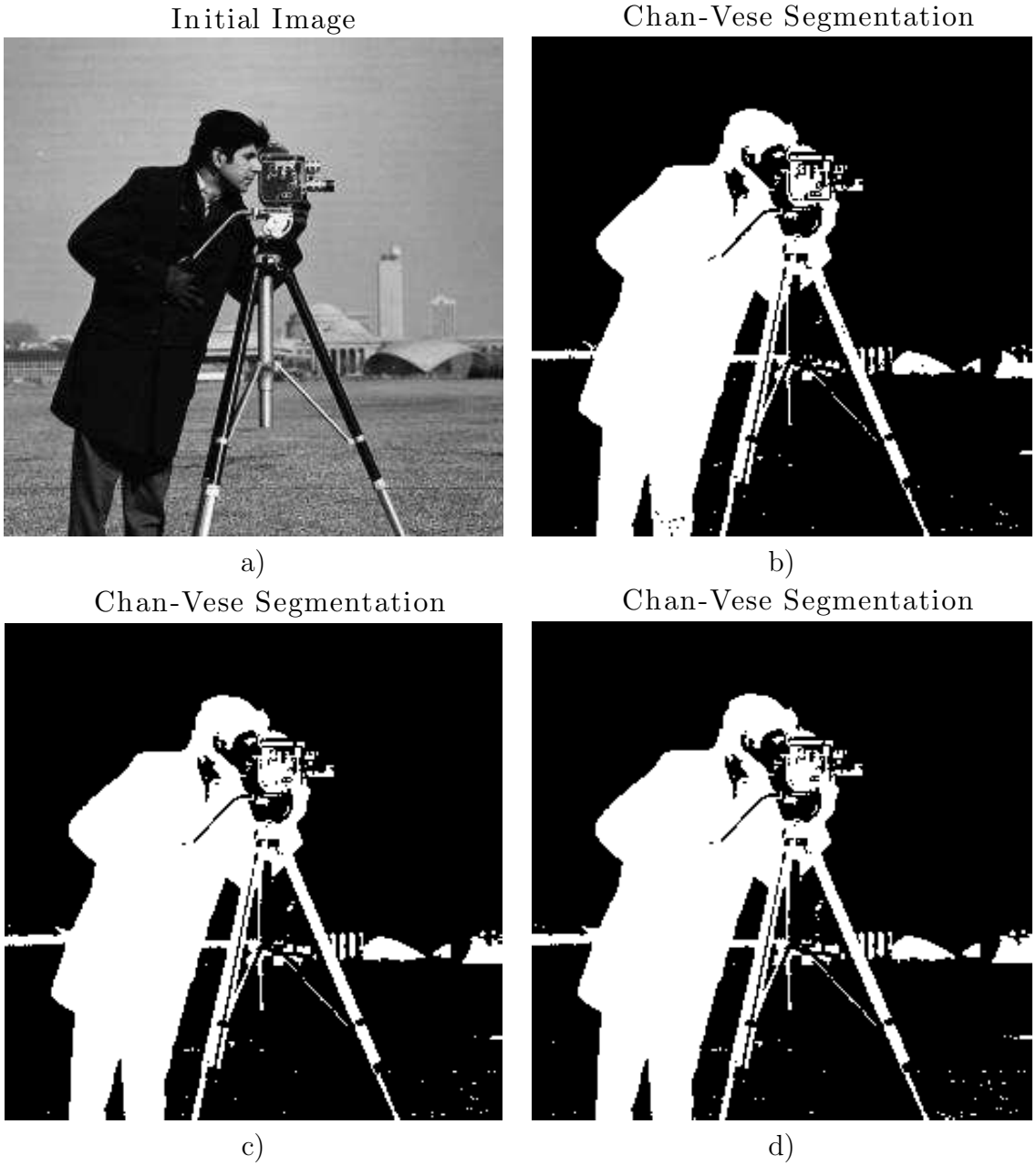


Figure 6: A 256×256 grayscale image of a camera man in a). Images b), c) and d) are segmentations after one, two and three iterations, respectively. Parameter value $\lambda = 0.00025$, initial contour $\phi_0(x, y) = -\sqrt{(x-128)^2 + (y-128)^2} + 10$ and CT = 9.07 s. Source: www.math.ucla.edu/∼yanovsky/Research/IPCG/Deblurring/ROF/cameraman_clean.jpg

the Crank–Nicolson method for the diffusion equation portion. The simple thresholding step is done after diffusion is performed up to a time τ, satisfying (10).

The Crank–Nicolson method is second order in time and space. At the boundary $\partial\Omega$, a second order centered difference scheme is used to incorporate zero flux conditions. That is we approximate $\partial\chi/\partial\vec{n} = 0$ on $\partial\Omega$ as

$$\frac{\partial\chi^n_{i,1}}{\partial n_x} = \frac{\chi^n_{i,2} - \chi^n_{i,0}}{2h} = 0, \quad \frac{\chi^n_{i,N}}{\partial n_x} = \frac{\chi^n_{i,N+1} - \chi^n_{i,N-1}}{2h} = 0, \quad \text{for all } i = 1, \ldots, M \tag{17}$$

and

$$\frac{\partial\chi^n_{1,j}}{\partial n_y} = \frac{\chi^n_{2,j} - \chi^n_{0,j}}{2h} = 0, \quad \frac{\chi^n_{M,j}}{\partial n_y} = \frac{\chi^n_{M+1,j} - \chi^n_{M-1,j}}{2h} = 0, \quad \text{for all } j = 1, \ldots, N. \tag{18}$$

The values $\chi^n_{i,0}$, $\chi^n_{i,N+1}$, $\chi^n_{0,j}$ and $\chi^n_{M+1,j}$ are ghost points.

For exposition we assume a square domain Ω, i.e. $M = N$. The Crank–Nicolson method, with zero flux BCs approximated using (17) and (18), can be written as

$$\left[I_{N^2} - \frac{\Delta t}{2h^2}W\right]\vec{\chi}^{n+1} = \left[I_{N^2} + \frac{\Delta t}{2h^2}W\right]\vec{\chi}^n, \tag{19}$$

where $\vec{\chi}^n$ (or $\vec{\chi}^{n+1}$) is a reshaped $N^2 \times 1$ vector of $\chi^n_{i,j}$ values. The vector $\vec{\chi}^n$ (or $\vec{\chi}^{n+1}$) can be constructed by stacking columns of χ^n (or χ^{n+1}). The matrix I_ν is the $\nu \times \nu$ identity matrix and W is an $N^2 \times N^2$ block tridiagonal matrix

$$W = \begin{bmatrix} D & 2I_N & & & \\ I_N & D & I_N & & \\ & \ddots & \ddots & \ddots & \\ & & I_N & D & I_N \\ & & & 2I_N & D \end{bmatrix} \quad \text{with} \quad D = \begin{bmatrix} -4 & 2 & & & \\ 1 & -4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -4 & 1 \\ & & & 2 & -4 \end{bmatrix}, \tag{20}$$

where D is an $N \times N$ matrix. The matrix equation (19) is solved using MATLAB's built-in "backslash" command here. Note that all $N^2 \times N^2$ components are not stored, since all matrices I_ν, W and D are sparse.

To test this method we examine how a circle evolves by mean curvature motion. The evolution of the circle moves with speed κ towards the center. The ordinary differential equation $dr/dt = -1/r$ can be used to determine the exact solution as

$$r(t) = \sqrt{\frac{1}{16} - 2t}.$$

Figure 7 shows a circle of initial radius $R_0 = 0.25$ shrink by mean curvature motion. The circle evolves until $T = 3/256$ with four equal time steps of $\Delta t = T/4$. The exact radius at time $t = 3/256$ is $r = \sqrt{5/128}$.

The evolution of the circle was computed for different time and space step sizes. Table 1 gives the image resolution, number of time steps taken, relative error in the radius and the order of convergence. The image domain is $[0, 1] \times [0, 1]$ and thus the resolution relates to

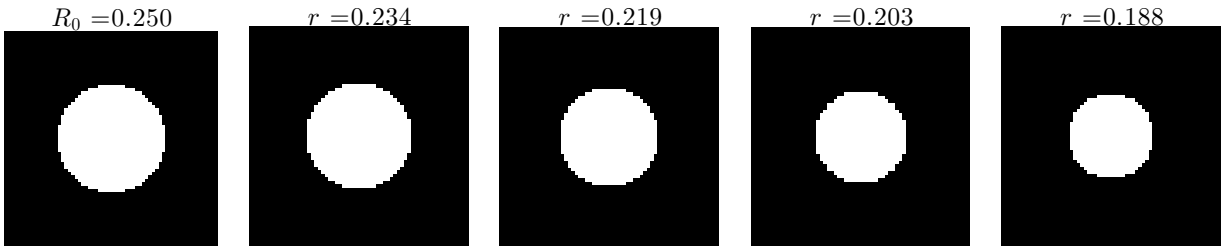


Figure 7: Circle shrinking by mean curvature motion until $T = 3/256$. Initial radius as $R_0 = 0.25$ and $\Delta t = T/4$.

Resolution	# of time steps	Relative error	Order
32×32	2	0.2094	—
64×64	4	0.1304	0.684
128×128	8	0.0908	0.521
256×256	16	0.0513	0.824
512×512	32	0.0316	0.699

Table 1: Relative errors in the radius of circle evolving by mean curvature motion. Results are given for five different resolutions and increasing number of time steps. Order of convergence is also given in the last column.

the space step size. The resolution and time step size are doubled together and a order of convergence near 1 is seen. The resolution and time step size have to change together to ensure (10) is satisfied. When Δt to too small compared to h, the circle does not move. When Δt is too large compared to h, the boundary of the circle becomes sporadic and circle does not shrink as much.

It is also know that motion by mean curvature causes Gaussian noise to disappear, straightens edges and rounds corners. Figure 8 shows the MBO method used on a square that has wavy sides, sharp corners and Gaussian noise added. Noise disappears after the first time step in figure 8b. In figure 8e, the wavy sides are completely straightened after four time steps. The corners round more and more throughout all the time steps.

Figure 8: A 256×256 image in a) of a square with two wavy sides and Gaussian noise added. Images b)–e) show mean curvature motion with the MBO method using $\Delta t = 20$.

3.2.2 Esedoḡlu–Tsai Method

The same notation for numerics of the Chan–Vese method and MBO method apply here. The diffusion part (13) of the Esedoḡlu–Tsai method can be written as

$$w_t = \Delta w - A(\mathbf{x})w + B(\mathbf{x}),$$

with

$$A(\mathbf{x}) = \frac{\lambda}{\sqrt{\pi\Delta t}}[(I_0(\mathbf{x}) - c_1)^2 + (I_0(\mathbf{x}) - c_2)^2] \quad \text{and} \quad B(\mathbf{x}) = \frac{\lambda}{\sqrt{\pi\Delta t}}(I_0(\mathbf{x}) - c_2)^2.$$

Here we use the implicit discretization in time of

$$\frac{w^{n+1} - w^n}{\Delta t} = \Delta w^{n+1} - Cw^{n+1} + (C - A(\mathbf{x}))w^n + B(\mathbf{x}),$$

where C is a constant chosen to be large enough compared to $A(\mathbf{x})$. In all our numerical experiments the value of C was taken to be $C = \frac{\lambda}{\sqrt{\pi\Delta t}}\text{mean}(I_0(\mathbf{x}))$.

The Laplacian of w is discretized using the five point Laplacian. Zero flux boundary conditions are incorporated the same as the MBO method with (17) and (18). The full discretization in space and time can then be written as the system of linear equations

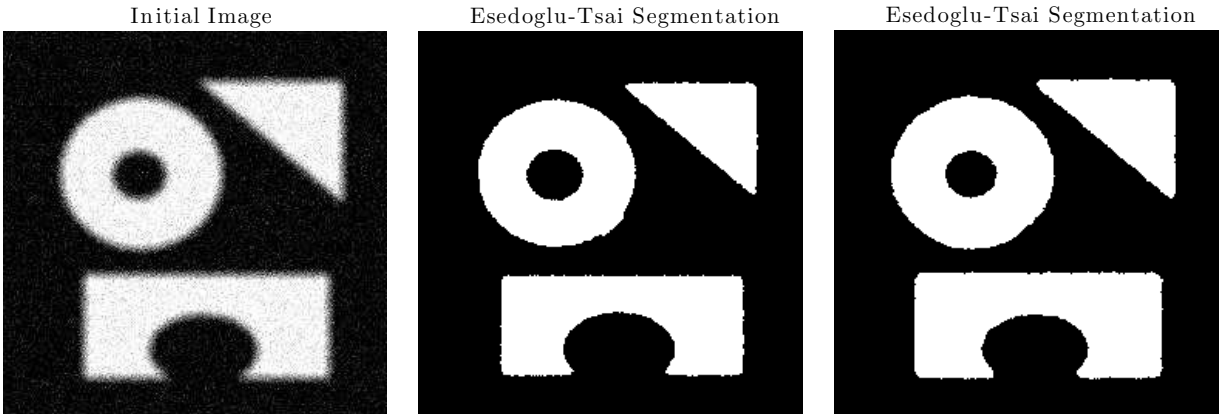
$$[(1 + \Delta t\, C)I_{N^2} + W]\, w^{n+1} = [(1 + \Delta t\, C)I_{N^2}]\, w^n + \vec{A}w^n + \vec{B}, \tag{21}$$

where W is defined in (20). The vectors w^n, w^{n+1}, $\vec{A}$ and $\vec{B}$ are of size $N^2 \times 1$ reshaped like χ^n in (19). The matrix equation (21) is solved using MATLAB's built-in solver. Note that the the diffusion equation is solved until time $t^n = \tau$. The thresholding step is then applied to the approximate solution of $w(\tau, \mathbf{x})$. That is, thresholding does not have to be done after every diffusion step.

Experiments here are done on the same four images as in section 3.1. Identical parameters values λ are taken. The value of $\mu = 1$ in the Chan–Vese method was taken since the corresponding term in the Esedoḡlu–Tsai method has no parameter. In all experiments $h = 1$ and Δt is a multiple of h^2. Multiple diffusion steps can be taken for each thresholding step of Δt. Figure 10a is the only image that uses multiple diffusion steps; three diffusion steps of size $\Delta t/3$ are taken. Initial segmentation guess is a circle centered in the image with radius R_0. Inside the circle the intensity is one and outside it is zero. Initial image I_0 is shown with binary images of the segmentation at different time steps. The exact values of λ, R_0 and computation time are given in each figure caption.

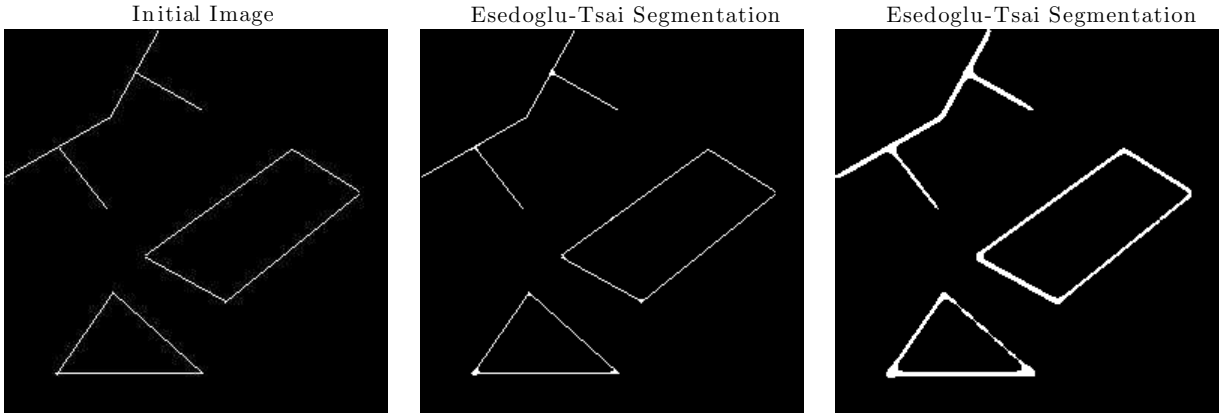
Figure 9a is the binary image of three shapes with Gaussian noise and blurring added. Similar to the result of Chan–Vese, after one time step the method reveals the three shapes with bumpy edges (figure 9b). Then after one more time step in figure 9c the edges are smoothed slightly. Comparing figure 9 with the Chan–Vese result, in figure 3, it can be seen that the Esedoḡlu–Tsai method produces less jagged edges but slightly more round corners.

Figure 10a is the binary image of shapes constructed from lines. The Esedoḡlu–Tsai method gives a good segmentation after just one time step in figure 10b. It can be seen in figure 10b however that the method is not as good as the Chan–Vese method in corner regions. It was also noticed that the method becomes worse if segmentation is continued



a) b) c)

Figure 9: A 249×249 binary image of three noisy and blurred shapes in a). Images b) and c) are segmentations after one and two time steps, respectively. Parameter $\lambda = 0.0001$, initial guess radius $R_0 = 10$, time step size $\Delta t = h^2$ and CT = 2.32 s.



a) b) c)

Figure 10: A 256×256 binary image of different shapes created from lines in a). Images b) and c) are segmentations after one and five time steps, respectively. Parameter value $\lambda = 0.00002$, initial guess radius $R_0 = 2$, time step size $\Delta t = h^2$ and diffusion done with three steps of $\Delta t/3$. Computation time for the desired segmentation in b) is CT = 2.61 s.

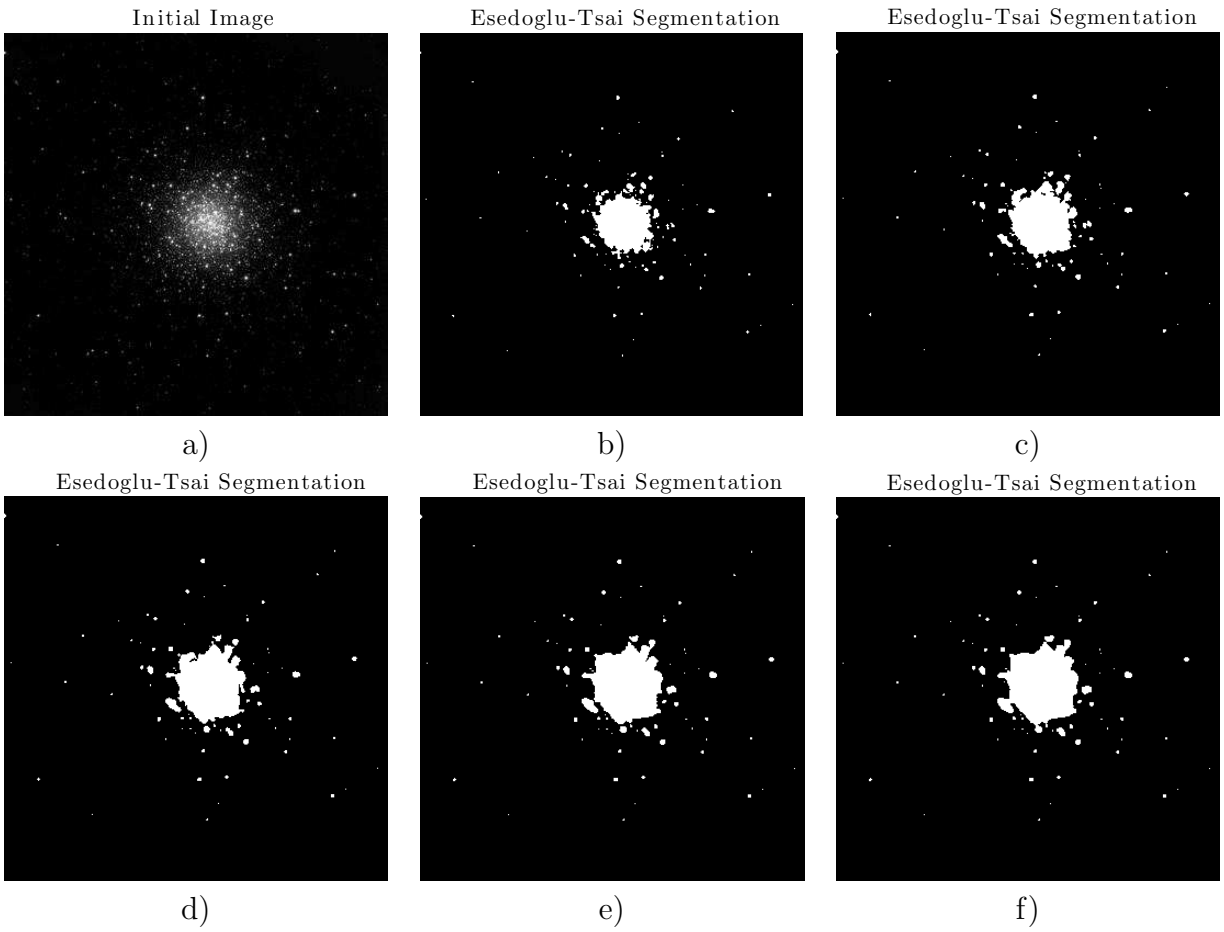


Figure 11: A 350 × 350 grayscale image of a star cluster in a). Images from b)–f) are after one to five time steps, respectively. Parameter value $\lambda = 0.00005$, initial guess radius $R_0 = 10$, time step size $\Delta t = 0.1h^2$ and CT = 6.40 s. Source: www.astro.caltech.edu/palomar/images/jarrett_M3_med.jpg.

for longer. After five time steps, figure 10c shows that the method actually increases the thickness of the lines. It seems that the Esedoḡlu–Tsai method continues diffusing after an appropriate segmentation is found.

Figure 11a is the grayscale image of a star cluster. The Esedoḡlu–Tsai method obtains a reasonable segmentation much faster than the Chan–Vese method. After just one time step the star cluster and stars out further are segmented (in figure 11b). The method then continues to obtain more of the star cluster in figure 11c to figure 11f. The segmentation is not as detailed as the Chan–Vese method for the star cluster.

The final test image, figure 12a, is the grayscale image of a camera man. After one time step the shape of the camera man's body and tripod is apparent, but the facial area and background not segmented correctly (see figure 12b). After another time step in figure 12c the segmentation is satisfactory. One more time step and more details are show in figure 12d. Comparing to the Chan–Vese segmentation (figure 6) it can be seen that many details are blurred together when using the Esedoḡlu–Tsai method. The Chan–Vese method only took

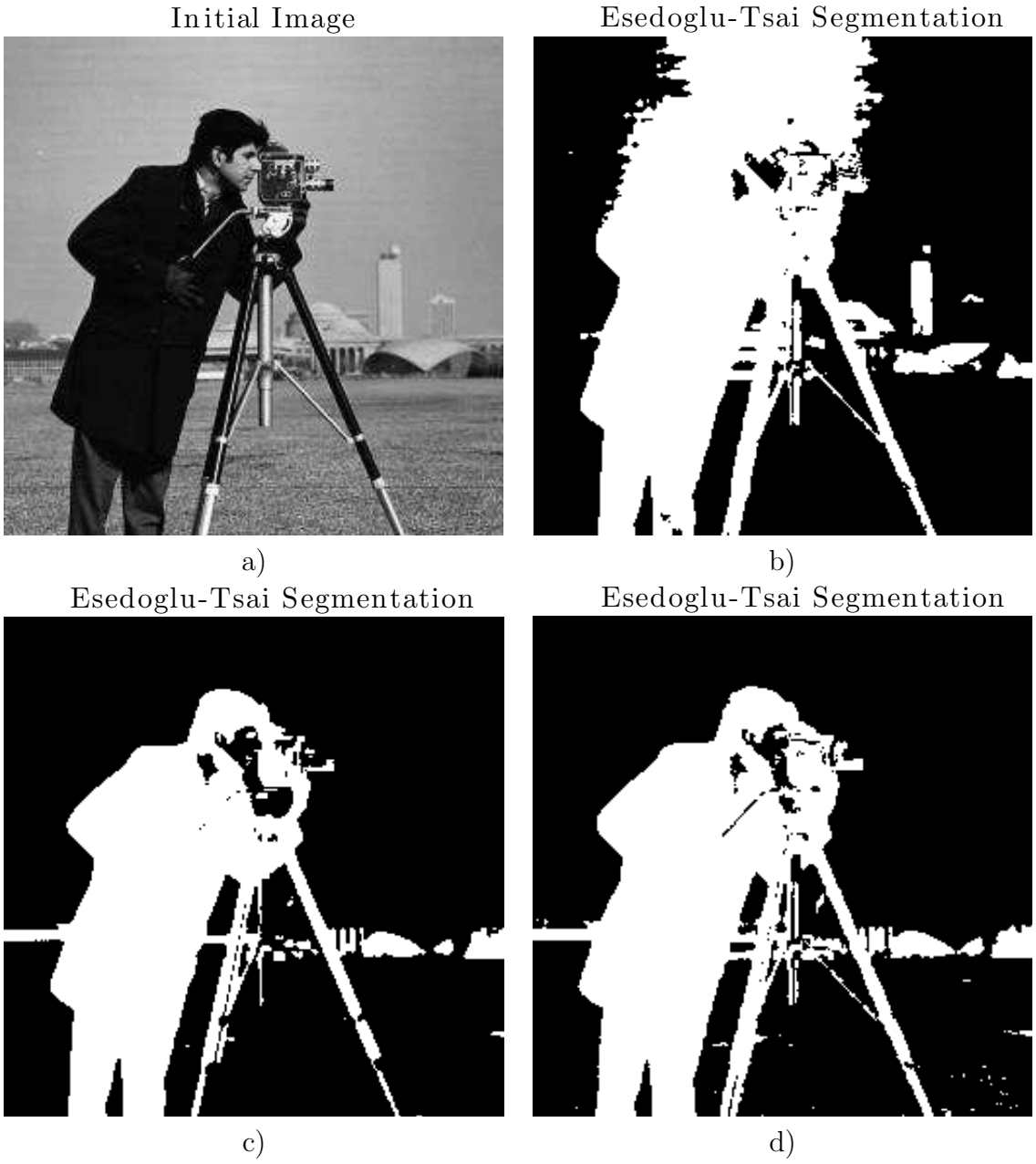


Figure 12: A 256×256 grayscale image of a camera man in a). Images b), c) and d) are segmentations after one, two and three time steps, respectively. Parameter value $\lambda = 0.00025$, initial guess radius $R_0 = 10$, time step size $\Delta t = 0.8h^2$ and CT = 4.26 s. Source: www.math.ucla.edu/∼yanovsky/Research/IPCG/Deblurring/ROF/cameraman_clean.jpg.

one iteration to determine an appropriate segmentation, however the Esedoḡlu–Tsai method has less than half the total computation time.

4 Discussion

Image segmentation is an important initial step in applications of object recognition and interpretation [9]. In this project we explored two types of segmentation methods based on the piecewise constant Mumford–Shah energy [4]. The Chan–Vese method used a level set approach, while the Esedoḡlu–Tsai method implemented threshold dynamics. Both methods return similar segmentations but have certain distinct characteristics. Note that all images here have approximately piecewise constant regions. Both methods would have difficulty for objects with intensities that vary drastically.

The Chan–Vese method allows for easy access to edges of the objects since it uses a level set method. The Esedoḡlu–Tsai method would require additional processing to determine the boundary of the binary segmentation. Dependent on the application object edges may be more important than entire object, so Chan–Vese is probably desired. Chan–Vese method seems to segments more details while Esedoḡlu–Tsai method tends to produce smoother edges. That is, the Esedoḡlu–Tsai method tends to produce segmentations that look more like a cartoon. It is also difficult to determine appropriate time step sizes for the Esedoḡlu–Tsai method. Like the MBO method, diffusion can become stuck if the time step is too small or give inaccurate results if the time step is too large.

The Esedoḡlu–Tsai method has a faster computation time and usually has adequate segmentation after the first time step. The Chan–Vese method takes more than double the computation time for examples in section 3. Here the Esedoḡlu–Tsai method is implemented using a simple method and still has smaller computation times than the Chan–Vese method. The Esedoḡlu–Tsai method requires a relatively simple PDE to be solve, therefore is amendable to even faster numerical techniques. The Esedoḡlu–Tsai method is therefore more desirable in industrial applications where image sizes are large.

References

[1] G. Shakhnarovich, L. Lee, and T. Darrell, "Integrated face and gait recognition from multiple views," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1. IEEE, 2001, pp. I–439.

[2] D. L. Pham, C. Xu, and J. L. Prince, "Current methods in medical image segmentation 1," *Annual review of biomedical engineering*, vol. 2, no. 1, pp. 315–337, 2000.

[3] D.-Y. Chen, Y.-H. Lin, and H.-P. Kuo, "Scattering-based brake-light modeling and detection," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*. IEEE, 2011, pp. 295–300.

[4] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Communications on pure and applied mathematics*, vol. 42, no. 5, pp. 577–685, 1989.

[5] B. Merriman, J. K. Bence, and S. Osher, *Diffusion generated motion by mean curvature*. Department of Mathematics, University of California, Los Angeles, 1992.

[6] T. F. Chan and L. A. Vese, "Active contours without edges," *Image processing, IEEE transactions on*, vol. 10, no. 2, pp. 266–277, 2001.

[7] Y.-H. R. Tsai *et al.*, "Threshold dynamics for the piecewise constant mumford–shah functional," *Journal of Computational Physics*, vol. 211, no. 1, pp. 367–384, 2006.

[8] R. J. LeVeque, *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*. SIAM, 2007.

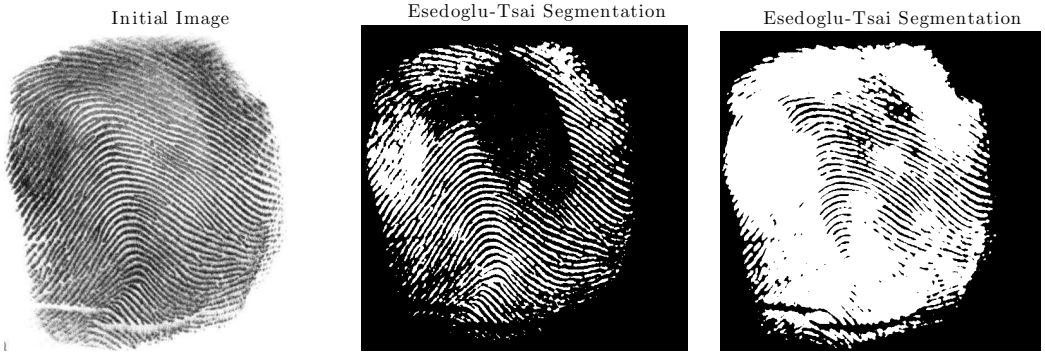
[9] J. M. Morel and S. Solimini, *Variational methods in image segmentation*. Birkhauser Boston Inc., 1995.

Appendix

Here the Esedoḡlu–Tsai method was used on more real images. These tests were done just for curiosity of possible real world applications. The first example, given in figure 13, shows how the shape of a gun could be detected using segmentation. We see that segmentation could be used to help repair smudged fingerprints in figure 14. After $2\Delta t$, in figure 14b, the smudged section can be identified. Further segmentation can then be used to fill in the smudged section (figure 14c). Figure 15 shows how segmentation could help determine licence plate numbers. The last example, in figure 16, is a fingerprint with a watermark over the image. Segmentation does a decent job in removing the watermark. This could be used as a quick method to remove a watermark if complete removable is unnecessary.

a) b)

Figure 13: A 351×351 grayscale image of a boy with a gun in a). Segmentation in b) is after 3 times steps of $\Delta t = 0.005h^2$ with $h = 1$. Parameter value $\lambda = 0.000008$ and initial guess radius of $R_0 = 1$. Source: http://news.bbcimg.co.uk/media/images/61719000/jpg/_61719840_82039610.jpg.

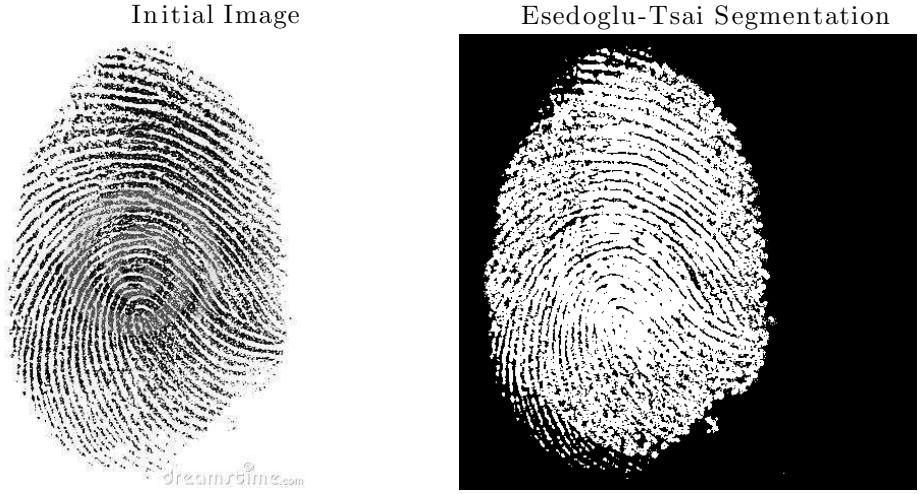


a) b) c)

Figure 14: A 508×508 grayscale image of a smudged fingerprint in a). Segmentation in b) and c) are after one and eight time steps $\Delta t = 0.6h^2$ with $h = 1$, respectively. Parameter value $\lambda = 0.000095$ and initial guess radius $R_0 = 10$. Source: http://shs2.westport.k12.ct.us/forensics/04-fingerprints/arch_sample.gif

a) b)

Figure 15: A 256×256 grayscale image in a) of two licence plates. Segmentation in b) is after 6 times steps of $\Delta t = 0.7h^2$ with $h = 1$. Parameter value $\lambda = 0.0008$ and initial guess radius of $R_0 = 1$. Source: created from two separate licence plate images on google.



a) b)

Figure 16: A 450×450 grayscale image of a fingerprint with a watermark in a). Segmentation in b) is after 2 times steps of $\Delta t = 0.5h^2$ with $h = 1$. Parameter value $\lambda = 0.01$ and initial guess radius of $R_0 = 200$. Source: http://thumbs.dreamstime.com/x/real-thumb-fingerprint-pattern-isolated-26587196.jpg