Statistical Density Estimation using Threshold Dynamics for Geometric Motion

Tijana Kostić · **Andrea Bertozzi**

Abstract Our goal is to estimate a probability density based on discrete point data via segmentation techniques. Since point data may represent certain activities, such as crime, our method can be successfully used for detecting regions of high activity. In this work we design a binary segmentation version of the well-known Maximum Penalized Likelihood Estimation (MPLE) model, as well as a minimization algorithm based on thresholding dynamics originally proposed by Merriman, Bence and Osher [21]. We also present some computational examples, including one with actual residential burglary data from the San Fernando Valley.

1 Introduction

The idea to use binary segmentation techniques for density estimates arises from the need to quickly and accurately locate the regions of higher activity, as well as to calculate the local density. Instead of obtaining a complete density function estimate, such as seen in [23] and [29], our goal is to segment only one region at a time. Maximum penalized likelihood estimation models have become standard for estimating a probability density function $d(\mathbf{x})$ based on point data $\mathbf{x}_1, \mathbf{x}_2 \ldots \mathbf{x}_N \in \mathbb{R}^n$, and they take a form

$$\hat{d}(\mathbf{x}) = \arg \max_{\substack{d(\mathbf{x}) \geq 0 \\ \int d = 1}} \left\{ \sum_{i=1}^{N} \log(d(\mathbf{x}_i)) - \alpha R(d) \right\} \tag{1}$$

Tijana Kostić
Mathematics Department, UCLA, Box 951555, Los Angeles, CA 90095-1555, USA
E-mail: kostict@math.ucla.edu

Andrea Bertozzi
E-mail: bertozzi@math.ucla.edu

where $R(d)$ is a penalty function. A wide variety of different penalty functions are proposed in the literature, and many of them are designed to impose additional smoothness on the density function. Single variable density estimation with a non-smooth density function is discussed in [28] and [18] where the authors of both works introduced the TV semi-norm as a penalty function. In their work [23], Mohler et al. construct the first two dimensional TV norm based MPLE:

$$\hat{d}(\mathbf{x}) = \arg \max_{\substack{d(\mathbf{x}) \geq 0 \\ \int d = 1}} \left\{ \sum_{i=1}^{N} \log(d(\mathbf{x}_i)) - \alpha \int |\nabla d| \right\}. \tag{2}$$

The authors also create an efficient computational method for minimization of (2) based on the Split Bregman algorithm [15]. Smith et al. in [29] propose a modification of the model (2), as well as the model that uses the H^1 norm as a penalty function, and they successfully integrated geographic information of the observed region to get a more accurate density estimate. The authors use city maps, census data as well as other types of geographical data to determine the region where the events typically occur, the valid region D. Knowing that the density function is zero anywhere outside the valid region, the authors align the zero level sets of the valid region and the density function. In their modified MPLE model aligning is achieved through addition of the alignment term, and the model they propose is

$$\hat{d}(\mathbf{x}) = \arg \max_{\substack{d(\mathbf{x}) \geq 0 \\ \int d = 1}} \left\{ \sum_{i=1}^{N} \log(d(\mathbf{x}_i)) + \lambda \int d \nabla \cdot \frac{\nabla 1_D}{|\nabla 1_D|} - \alpha \int |\nabla d| \right\} \tag{3}$$

where 1_D is a characteristic function of the valid region. In the case of the weighted H^1 MPLE model, in order to allow the density function d to have sharp jumps on the border of the valid region, the H^1 penalty functional was forced away from the boundary of the valid region. Thus, the proposed model becomes:

$$\hat{d}(\mathbf{x}) = \arg \max_{\substack{d(\mathbf{x}) \geq 0 \\ \int d = 1}} \left\{ \sum_{i=1}^{N} \log(d(\mathbf{x}_i)) - \alpha \int z_\varepsilon^2 |\nabla d|^2 \right\}, \tag{4}$$

where z_ε is a continuous function such that

$$z_\varepsilon = \begin{cases} 1 & \text{if } distance(\mathbf{x}, \partial D) > \varepsilon \\ 0 & \text{if } \mathbf{x} \in \partial D. \end{cases}$$

In this work we propose the variational MPLE segmentation model using the Ginzburg-Landau functional as a regularizer to segment the geography and calculate the density function:

$$\arg \min_{u(x)} \int \varepsilon |\nabla u|^2 + \frac{1}{\varepsilon} W(u) dx - \mu \int w \log(d(u)) dx \tag{5}$$

where w is a function that represents the given data, u is a segmentation function, and $d(u)$ is a two phase density function that can be determined from u. We define w as a sum of Dirac δ functions $w = \sum_{i}^{N} \delta(\mathbf{x}_i)$, where where $\mathbf{x}_i$ are data points. This function is introduced for the purpose of more compact notation. Note that in equation

(5) the maximum likelihood term $\int w \log(d(u))dx$ involves the data function w, while in equations (4),(3),(2) and (1) we used the sum $\sum_{i=1}^{N} \log(d(\mathbf{x}_i))$. The dense region Σ is located via segmentation function u, i.e. $u = \chi_\Sigma$. Using the information on the data set we obtain through function w, as well as the assumption that density $d(u)$ is piecewise constant, we calculate the values of $d(u)$ in Σ and Σ^C.

This paper is organized as follows, in Section 2 we give some background on variational methods for image segmentation, in Section 3 we describe the MBO scheme, in Section 4 we present some background on MPLE models. In Section 5 we discuss the proposed model in more details, and calculate the time dependent Euler-Lagrange equation to minimize the functional (5). Section 6 explains the thresholding dynamics for minimization of our energy functional. Details on the numerical implementation are presented in Section 7.

2 Background on variational methods in image segmentation and MBO scheme

Variational segmentation models are widely successful in image processing. In their pioneering work [24], Mumford and Shah propose the following segmentation model:

$$\min_{\substack{K \subseteq D \\ u:D \to \mathbf{R}}} MS(u,K) = \int_{D \setminus K} |\nabla u|^2 dx + \mu Length(K) + \lambda \int_D (u-f)^2 dx \qquad (6)$$

where f is an image that should be segmented, and u is a segmentation function. One of the major results in image segmentation is Chan-Vese algorithm presented in [5], where authors, inspired by the level set method by Osher and Sethian [25], propose a method for minimizing the piecewise constant version of (6):

$$\min_{\substack{\Sigma \subseteq D \\ c_1, c_2 \in \mathbf{R}}} E(\Sigma, c_1, c_2) = Per(\Sigma; D) + \lambda \int_\Sigma (c_1 - f)^2 dx + \lambda \int_{D \setminus \Sigma} (c_2 - f)^2 dx \qquad (7)$$

where Σ is a segmented region. The level set version of (7) proposed in [5]

$$\min_{\substack{\phi:D \to \mathbf{R} \\ c_1, c_2 \in \mathbf{R}}} \int_D |\nabla H(\phi)| + \lambda \left\{ H(\phi)(c_1 - f)^2 + (1 - H(\phi))(c_2 - f)^2 \right\} dx \qquad (8)$$

is obtained by the boundary of Σ being represented as a 0-level set of the function $\phi(x) : D \to \mathbf{R}$. After the optimal values for constants c_1 and c_2 are determined, the time-dependent Euler-Lagrange equation for ϕ is found:

$$\phi_t = H'_\varepsilon(\phi) \left\{ \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \lambda \left\{ (c_1 - f)^2 - (c_2 - f)^2 \right\} \right\} \qquad (9)$$

where H_ε is an approximation of the Heaviside function, and a semi-implicit numerical scheme is created to solve (9). Due to the computational complexity of the numerical algorithm that solves equation (9), Gibou and Fedkiw created algorithms that do not explicitly solve (9). In their work [16], they designed the hybrid k-means level

set method and applied it to images that were previously processed using the Perona-Malik diffusion. Another algorithm that successfully minimizes the piecewise constant Mumford-Shah functional without solving the gradient descent equation (9) is proposed in [31]. Numerous modifications of the piecewise constant Mumford-Shah model, and different ways to minimize it appeared in the literature (see [2], [6],[9] [17], [14], [4] and [30]).

The regularization based on the Total Variation semi-norm (TV) is widely used and established as a standard procedure in numerous image processing applications. It has been proven in [19] that the Ginzburg-Landau functional converges to the TV semi-norm in the sense of Γ-convergence. This property justifies the substitution of the TV norm for the Ginzburg-Landau functional. The reason one might prefer the GL functional over the TV semi-norm is the simplicity of the minimization numerical scheme that the GL functional yields. Indeed, the L^2 gradient descent of the GL functional gives us the Allen-Cahn equation

$$u_t = 2\varepsilon\Delta u - \frac{1}{\varepsilon}W'(u), \tag{10}$$

as opposed to having a nonlinear curvature term obtained through minimization of the TV semi-norm, although the Split Bregman technique [15] successfully resolves that issue. Models with the GL functional are typically referred to as diffuse interface models, because the double well potential causes the minimizer of the functional to have two phases with a diffuse interface between them. The diffuse interface scale is $O(\varepsilon)$. The Ginzurg-Landau functional has appeared in many image processing applications, such as inpainting [3,7] and segmentation [9,10].

In the case of motion by mean curvature a planar curve moves with a normal velocity that is proportional to the curvature at any given point. In their work [21], Merriman, Bence and Osher introduced an intuitive and efficient level set method to approximate the motion of an interface by its mean curvature flow. The authors analyzed the motion by mean curvature of the curve C by studying the diffusion equation $\chi_t = \Delta\chi$, where χ is the characteristic function of the set Σ and $\partial\Sigma = C$, i.e. C represents a sharp front between two phases of the characteristic function. The appropriate change of coordinates suggested in [21] reveals that, when diffusion is applied, any point on the front moves with the normal velocity that is equal to the mean curvature at that point. Simultaneously, the front is radially blurred. However, the $\chi = \frac{1}{2}$ level set is invariant to blurring. From there it follows that, the $\chi = \frac{1}{2}$ level set yields motion by mean curvature. Based on the previous observation the following numerical scheme is proposed:

– **Step 1** Let $v(x) = S(\delta t)u_n(x)$ where $S(\delta t)$ is a propagator by time δt of the equation:

$$v_t = \Delta v$$

with appropriate boundary conditions.

– **Step 2** Threshold

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

The reason we are interested in motion by mean curvature flow is due to is its close relation to the Allen-Cahn equation (10).

It was shown in [26] that limit of the rescaled solutions $u_\varepsilon(x, \frac{t}{\varepsilon}), \varepsilon \to 0^+$ of the Allen-Cahn equation yields motion by mean curvature. The modification of the MBO scheme proposed in [9] by Esedoglu and Tsai particularly interesting. In their paper [9], the authors proposed a diffuse interface version of (7)

$$MS_\varepsilon(u, c_1, c_2) = \int_D \varepsilon|\nabla u|^2 + \frac{1}{\varepsilon}W(u) + \lambda\{u^2(c_1 - f)^2 + (1 - u)^2(c_2 - f)^2\}dx \quad (11)$$

for segmenting the image f. The first variation of the model (11) yields the following gradient descent equation:

$$u_t = 2\varepsilon\Delta u - \frac{1}{\varepsilon}W'(u) + 2\lambda\{u(c_1 - f)^2 + (1 - u)(c_2 - f)^2\} \quad (12)$$

and the adaptation of the MBO scheme was used to solve it. Similarly to the MBO scheme where the propagation step based on the heat equation is combined with thresholding, Esedoglu and Tsai proposed the following scheme:

– **Step 1** Let $v(x) = S(\delta t)u_n(x)$ where $S(\delta t)$ is a propagator by time δt of the equation:
$$w_t = \Delta w - 2\tilde{\lambda}\left(w(c_1 - f)^2 + (1 - w)(c_2 - f)^2\right)$$
with appropriate boundary conditions.
– **Step 2** Set
$$u_{n+1}(x) = \begin{cases} 0 & \text{if } v(x) \in (-\infty, \frac{1}{2}] \\ 1 & \text{if } v(x) \in (\frac{1}{2}, \infty) \end{cases}$$

Based on the computational experiments, the authors presented their conclusions about the choice of δt timestep in the step **1.** of the algorithm. If δt is chosen too large comparing to the parameter λ^{-1}, the interface tend to be overly smooth. As the advantage of choosing larger values for δt they mention faster convergence. The segmentation could also benefit from the larger values for the parameter λ, as there is less penalty on high curvature of the interface in this case. The size of the spatial resolution has to be taken into account when the value for δt is chosen, as the values much smaller than the spatial resolution could lead to the stillness of interface. These observations can also be used as guidance in parameter selection for our algorithm. The authors also show that MBO thresholding type methods for binary segmentation can easily be generalized to multi-phase segmentation methods. To accomplish that, the authors propose the four phase model based on the modified version of (11) with the sum of the two Ginzburg-Landau functionals built around two different segmentation functions, u_1 and u_2. In this case, the fidelity term naturally depends on both u_1 and u_2. After they find the gradient descent equations with respect to u_1 and u_2, they construct the thresholding numerical scheme to solve the obtained system of parabolic equations. We will adapt ideas by Esedoglu and Tsai to solve our MPLE problem and illustrate the usefulness of this simple method. Some extension of the MBO algorithms appeared in [11, 12, 22] An efficient algorithm for motion by mean curvature using adaptive grids was proposed in [27].

3 MPLE methods and proposed model

Let us assume our data consists of n points $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$ and is a sample of n independent random variables with common density d_0. Maximum likelihood estimation is a standard method for estimating a density function d_0 based on given data. In the case the of the parametric model, we know that d_0 belongs to the family of density functions $D = \{d(\cdot, \theta) : \theta \in \Theta\}$, and our goal is to find a parameter θ_0 such that $d(\cdot, \theta_0) = d_0(\cdot)$. This class of problems are known as parametric density estimates. In 1922, According to Fisher's model [13], an optimal parameter $\theta_0 \in \Theta$ (where Θ is a set of all parameters) satisfies the following:

$$\theta_0 = -\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \log d(\mathbf{x}_i, \theta) \tag{13}$$

However, in many applications, a parametric model may not be available, or information on the family of density functions d_0 belongs to may be unknown, in which case we are dealing with a nonparametric density estimate. The analog of the model (13) is an ill-posed problem, i.e. finding a probability density function d_0 such that

$$d_0 = -\min_{\substack{d: \Omega \to \mathbb{R} \\ \int d = 1}} \frac{1}{n} \sum_{i=1}^{n} \log d(\mathbf{x}_i) \tag{14}$$

has no solution, and thus can not be directly applied. To avoid "rough" density functions, with large values concentrated around data points and small values elsewhere, the roughness penalization function $R(d)$ was introduced, and this method is known as maximum penalized likelihood estimation:

$$d(\mathbf{x}) = \max_{\substack{d: \Omega \to \mathbb{R} \\ \int d = 1}} \left\{ \sum_{i=1}^{N} \log(d(\mathbf{x}_i)) - \alpha R(d) \right\} \tag{15}$$

Different penalty functionals appeared in literature, such as $R(d) = \int_{\Omega} |\nabla \sqrt{d}|^2$, or $R(d) = \int_{\Omega} |\nabla^3 \log d|^2$ from [8]. These, and many other standard penalty functional enforce smoothness on density function, but do not perform well when the density function has sharp gradients, i.e. is piecewise constant. To resolve this issue, Koenker and Mizera in [18] as well as Sardy and Tseng in [28] propose the penalty functional to be the TV semi-norm. This approach was also successfully used in [29] and [23]. In our work, since we assume the density is a step function, choosing a penalty functional that can successfully handle sharp gradients is crucial. As previously mentioned , instead of the TV semi-norm, we chose the Ginzburg-Landau functional to be the penalty functional.

3.1 General Model

For now are going to focus on the segmentation function u. We assume our segmentation function is the characteristic function of the region Σ, where Σ is an area with a larger density. For any given data and any given segmentation function there is a

unique density function corresponding to them. With w being the function that approximates the data, the total number of events is approximately equal to $\int w$, while the number of events inside and the number of events outside of the region Σ are approximated by $\int wu$ and $\int w(1-u)$, respectively. According to that, the density $c_1(u)$ inside the region Σ is equal to $\frac{\int wu}{\int u \int w}$ and the density $c_2(u)$ in the region Σ^C is equal to $\frac{\int w(1-u)}{\int w \int (1-u)}$. Finally, we write the density function as $c_1(u)u + c_2(u)(1-u)$. The established correspondence between the segmentation and the density function suggests that building a diffuse interface MPLE model around the segmentation function is possible. As the segmentation function takes only 0 and 1 values, the Ginzburg-Landau functional is a natural choice. As the density is a rescaled segmentation function, using the Ginzburg-Landau functional for u, as opposed to the Ginzburg-Landau functional for the density seems both reasonable and convenient.

Here we propose the diffuse interface Maximum Penalized Likelihood Estimation model.

$$\arg\min_{u(x)} \int \varepsilon |\nabla u|^2 + \frac{1}{\varepsilon} W(u) dx - \mu \int w \log(c_1(u)u + c_2(u)(1-u)) dx \quad (16)$$

where $c_1(u) = \frac{\int wu}{\int u \int w}$ and $c_2(u) = \frac{\int w(1-u)}{\int w \int (1-u)}$, w represents the given data, $W(u) = u^2(1-u)^2$ and μ is a parameter. As we already mentioned, the Ginzburg-Landau functional converges to the TV norm in the sense of Γ convergence, as $\varepsilon \to 0^+$. As a consequence, the diffuse interface model we propose here converges to the TV-based MPLE that was used in [23],[29], when $\varepsilon \to 0^+$. Now, variation of energy of (16) gives us the following cases for the L^2 gradient descent equation:

– If both and $c_1(u)$ and $c_2(u)$ (further we use c_1 and c_2 instead for simplicity of the notation) are non-zero:

$$u_t = 2\varepsilon \Delta u - \frac{1}{\varepsilon} W'(u) + \mu w \left[\frac{c_1 - c_2}{c_1} u + \frac{c_1 - c_2}{c_2} (1-u) + \left(\frac{\int (1-u)w}{\int 1-u} - \frac{\int uw}{\int u} \right) \right] \quad (17)$$

– If c_1 is equal to zero:

$$u_t = 2\varepsilon \Delta u - \frac{1}{\varepsilon} W'(u) + \mu \left[w(u-1) + \left(\frac{\int (1-u)w}{\int 1-u} - w \right) \right] \quad (18)$$

– If c_2 is equal to zero

$$u_t = 2\varepsilon \Delta u - \frac{1}{\varepsilon} W'(u) + \mu \left[wu + \left(w - \frac{\int uw}{\int u} \right) \right] \quad (19)$$

3.2 Special Case Model

For the special case problem the density in the region Σ^C is zero, so the density function is just a rescaled segmentation function. Thus, replacing the density function

$c_1(u)u + c_2(u)(1-u)$ in the MPLE term of our general model by the segmentation u seems reasonable. Now, the model we propose is:

$$\underset{u(x)}{\arg\min} \int \varepsilon |\nabla u|^2 + \frac{1}{\varepsilon} W(u) dx - \mu \sum_{i,j} w_{i,j} \log(u_{i,j}) \qquad (20)$$

Once again, assuming w is a function that approximates the data, in order to make everything well-defined we introduce the model with a small constant v:

$$\underset{u(x)}{\arg\min} \int \varepsilon |\nabla u|^2 + \frac{1}{\varepsilon} W(u) dx - \mu \int w \log((1-v)u + v(1-u)) dx \qquad (21)$$

The Euler-Lagrange equation for the energy functional (21) is:

$$u_t = 2\varepsilon \Delta u - \frac{1}{\varepsilon} W'(u) + \mu w \left(\frac{1}{1-v} u + \frac{1}{v} (1-u) \right)(1-2v) \qquad (22)$$

4 Proposed dynamics

We use the following gradient descent equation

$$u_t = 2\varepsilon \Delta u - \frac{1}{\varepsilon} W'(u) + A(u(\cdot,t))u + B(u(\cdot,t)) \qquad (23)$$

with $A(u(\cdot,t))$ and $B(u(\cdot,t))$ being non-linear functions. Each of the gradient descent equations, (17), (18), (19) and (22) can be given in the form (23), where $A(u(\cdot,t))$ and $B(u(\cdot,t))$ take different vales in different cases.

$A(u(\cdot,t)) = -\mu w \dfrac{(c_1 - c_2)^2}{c_1 c_2}$ $\quad B(u(\cdot,t)) = \mu w \dfrac{c_1 - c_2}{c_2} + (c_2 - c_1) \int w \quad$ in the equation (17)

$A(u(\cdot,t)) = \mu w$ $\qquad\qquad\quad B(u(\cdot,t)) = \mu (c_2 \int w - 2w) \qquad\qquad$ in the equation (18)

$A(u(\cdot,t)) = \mu w$ $\qquad\qquad\quad B(u(\cdot,t)) = \mu (w - c_1 \int w) \qquad\qquad$ in the equation (19)

$A(u(\cdot,t)) = -\mu w \dfrac{(2v - 1)^2}{(1-v)v}$ $\quad B(u(\cdot,t)) = \mu w \dfrac{1 - 2v}{v} \qquad\qquad$ in the equation (22)

Motivated by the MBO scheme for solving the Allen-Cahn equation, we propose a thresholding scheme to approximate the solution of the equation (23). Esedoglu and Tsai used a similar approach to minimize the Mumford-Shah segmentation functional in [9]. The first step we need to take toward generating a thresholding scheme is finding a good way to split the equation (23) into two steps analogous to those proposed in the MBO scheme. In that regard, finding a way that successfully deals with the non-linear forcing term of the equation (23) is critical. Inspired by the algorithm presented in [9] we propose the following dynamics:

– **Step 1.** Let $v(x) = S(\delta t)u_n(x)$ where $S(\delta t)$ is a propagator by time δt of the equation:

$$y_t = \Delta y - A(y(\cdot,t))y + B(y(\cdot,t))$$

with appropriate boundary conditions.

– **Step 2.** Set

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

5 Numerical implementation

In the propagation phase of the algorithm we solve the following PDE:

$$v_t = \Delta v - A(v(\cdot, t))v + B(v(\cdot, t))$$

with the u_n being an initial condition. To generate the numerical results we denote the timestep by $\delta \tau$ is a timestep, and approximate the Laplacian by its five-point stencil, which gives us the following scheme:

$$\frac{v^{n+1} - v^n}{\delta \tau} = \Delta v^{n+1} + A(v^n(\cdot, t))v^n + B(v^n(\cdot, t))$$

As the linear linear system described above is circulant we find the Fast Fourier transform (FFT) a convenient way solve it. We are aware that other solvers with even better computational complexity than the FFT, such as multigrid algorithm, could be used. In the propagation phase our goal is not to achieve steady state, and iterations are repeated only until the total propagation time reached δt. After the propagation phase, a thresholding step is necessary to complete the iteration:

$$u_{i,j}^{n+1} = \begin{cases} 0 & \text{if } v_{i,j}^l \in (-\infty, \frac{1}{2}] \\ 1 & \text{if } v_{i,j}^l \in (\frac{1}{2}, \infty) \end{cases}$$

where l is a total number of iterations we made in the propagation phase. The small relative change of the L^2 norm between two consecutive iterations was used as a stopping criterion.

In this implementation, the data function w is used as an initial condition, along with Dirichlet or Neumann boundary conditions.

5.1 Adaptive timestepping

The choice of timestep in the propagation phase, a "sub-timestep", can be chosen to optimize performance. In the early stage of computation, it is important to keep the sub-timestep small in order to obtain a good estimate in the propagation phase. However, as our algorithm is approaching steady state, a large number of iterations in the propagation phase pose a burden on the computational time. To successfully speed up the convergence of our algorithm, we used adaptive timestepping, a modified form of the scheme proposed in [1].

The scheme uses a dimensionless local truncation error calculated in every iteration, and the timestep is increased when the error is smaller then a chosen tolerance. The error at time t^n uses solution at three consecutive timesteps t^{n-1}, t^n and t^{n+1}. Let us

define $e^{n+1} = \frac{(v^{n+1}-v^n)}{v^n}$ and $e^n = \frac{(v^n-v^{n-1})}{v^n}$, as well as $\Delta t_{old} = t^n - t^{n-1}$. The previous definitions allow us to define a dimensionless estimate of the local truncation error:

$$\left\| e^{n+1} - \frac{\Delta t}{\Delta t_{old}} e^n \right\|_{L^2}. \tag{24}$$

This algorithm is both quick and practical. Even though other more accurate ways to estimate the measure of error are available, such as step doubling, they are often much more computationally expensive.

We used adaptive timestepping at two different levels, in the propagation phase of the algorithm we adapt the sub-timestep, as well as adapting an initial sub-timestep for the future iterations. In the propagation phase of any iteration, we calculate a dimensionless truncation error estimate for different propagation times. Once an error is smaller than a given tolerance Tol_1 for a certain number of the consecutive iterations, we increase the timestep by 10%. We also estimate the dimensionless error in every iteration of the algorithm, and if we find an error to be smaller than Tol_2 the initial sub-timestep in the propagation phase of the next iteration will be increased be 10%. However, we never allow the initial sub-timestep to be larger than $\frac{1}{8}$ of the timestep. Notice that we are not adapting the timestep, the total propagation time in each iteration is the same.

5.2 Adaptive resolution

Another way to improve the computational time is to use adaptive resolution. As we mentioned before, we use the data function w as an initial condition when solving the equation (23). It is reasonable to assume that the more the initial condition "resembles" the solution, the less iterations the algorithm would take to obtain the solution. The main idea is to generate a lower resolution form of the data set, then use a low resolution solution to create a good initial guess for the high resolution solution. Providing a good initial guess for the higher resolution problem is particularly useful as the iterations when the algorithm is applied to the higher resolution versions of the data set tend to be slower. In this implementation, we typically applied this procedure several times on some sparse data sets. At each step we create the coarser form of the given data set, until we reach the version of the data set that has a satisfying density. Our experiments show that data sets with the total density between 0.05 and 0.2 are optimal for this algorithm. Once a sufficiently dense low resolution version of the data set is obtained, we run our algorithm to get the low resolution solution, and start working our way up from there. The higher resolution approximation of the solution is then generated, and used as an initial condition in the next step. In the next step, we are solving the problem on the data set that has a higher resolution. It is important to mention that this process does not alter the original data set. We call this process *n-step adaptive resolution* where n is the total number of times we reduced the resolution of the original data set. The number of steps, n, is closely related to our choice of timestep. In case we are segmenting the region of higher density in our data, we noticed, through multiple experiments, that the timestep often can be given as $\omega 2^n$,

where n is the number of levels in adaptive resolution, and $\omega \in [0.15, 0.2]$. In case we are locating the valid region, we usually allow a smaller timestep, but also a larger number of levels in adaptive resolution. However, starting with a problem that has a significantly lower resolution comparing to the original one, we might run into some problems. Decreasing resolution significantly may result in a very different looking data set, thus segmentation would not perform in an expected way, i.e. this first initial guess would not be a good approximation of the solution we are trying to find.

6 Computational Examples

6.1 Test Shapes

To verify the performance of our algorithm, we constructed three examples of the probability density maps featuring three shapes and they are shown in the figures 1(a), 2(a) and 3(a). Each of the densities were sampled, and the toy data sets are obtained, and presented in the figures 1(b), 2(b) and 3(b) respectively. We applied the general version of our algorithm to solve these problems, and the results are shown in images 1(c), 2(c) and 3(c). Adaptive timestepping was used in all of the examples with three shapes, while the adaptive resolution was not used, given the small scale of these examples.
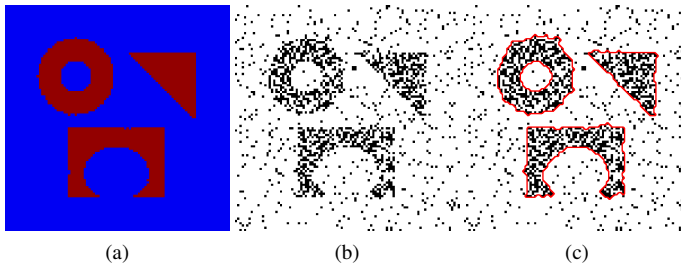

(a) (b) (c)

Fig. 1: Segmentation of three shapes: Image (a) presents a plot of the density map. Assuming the size of (a) is 1×1, the densities of the dense region and the sparse region are 3.3943 and 0.3927, respectively. Image (b) is a plot of the data set, in a 100×100 pixel image, with 1449 events. Image (c) shows the contours of the segmented dense region. In the process of generating this result we used the timestep 1.6 along with the parameter $\mu = 0.13$. The densities of our estimated dense and sparse regions are 3.4811 and 0.3649.

6.2 Orange County Coastline

In their work [29] Smith et al. constructed several test data sets using the spatial information of the Orange County coastline obtained from Google Earth. The authors
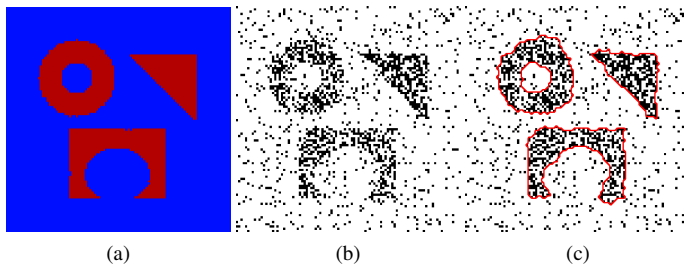
(a)   (b)   (c)

Fig. 2: Segmentation of three shapes: Image (a) presents a plot of the density map. Assuming the size of (a) is 1×1, the densities of the dense region and the sparse region are 3.145 and 0.456, respectively. Image (b) is a plot of the data set, in a 100×100 pixel image, with 1539 events. Image (c) shows the contours of the segmented dense region. The choice of parameters was 1.6 for the timestep, and $\mu = 0.15$. The densities of our estimated dense and sparse regions are 3.311 and 0.422.
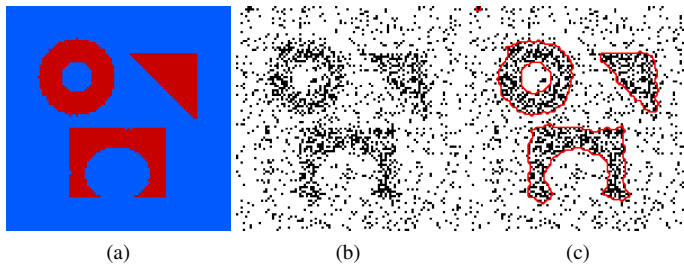

(a)   (b)   (c)

Fig. 3: Segmentation of three shapes: Image (a) presents a plot of the density map. Assuming the size of (a) is 1×1, the densities of the dense region and the sparse region are 2.605 and 0.592, respectively. Image (b) is a plot of the data set, in a 100×100 pixel image, with 1696 events. Image (c) shows the contours of the segmented dense region. The choice of parameters was 1.6 for the timestep, and $\mu = 0.1$. The densities of our estimated dense and sparse regions are 2.780 and 0.573.

determined the boundary between the valid and the invalid region by locating the ocean, rivers, parks and other features the valid region of a residential crimes map naturally excludes. Then, inside the valid region, they constructed three other regions and assigned different densities to them, which is shown on the density map in Figure 10, and we also, inspired by their examples, constructed the density maps shown in Figure 4 as well as in Figure 7. Sampling from these density functions we generated data sets shown in Figures 5(a), 6(a), 8(a), 9(a) and 11(a). A segmentation of the corresponding dense region is shown next to each data set. Note that there are four different levels of densities in Figure 10. The solution in Figure 11 shows that the different choice of the parameter μ can lead to segmentation of the dense regions at different levels. All images in this section have resolution of 600×1000 pixels.

Fig. 4: The toy density map of the Orange County Coastline, featuring two disconnected dense regions.

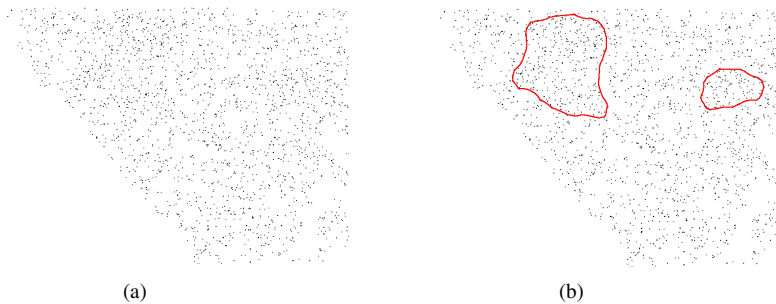(a)   (b)

Fig. 5: A 2000 event sample of the density function from Figure 4 is shown in (a). The contour of the segmented dense region is shown in (b). The choice of parameters is: $\mu = 0.1$, timestep is 13.0 with the 3-step adaptive resolution. The data points in this figure are manually enhanced for the purpose of more clear display of the image.

(a)   (b)

Fig. 6: An 8000 event sample of the density function from Figure 4 is shown in (a). The contour of the segmented dense region is shown in (b). The choice of parameters is: $\mu = 0.092$, timestep is 8.0 with the 2-step adaptive resolution. The data points in this figure are manually enhanced for the purpose of more clear display of the image.

Fig. 7: The toy density map of the Orange County Coastline, featuring three disconnected dense regions.

(a)   (b)

Fig. 8: A 2500 event sample of the density function from Figure 7 is shown in (a). The contour of the segmented dense region is shown in (b). The choice of parameters is: $\mu = 0.11$, timestep is 11.0 with the 3-step adaptive resolution. The data points in this figure are manually enhanced for the purpose of more clear display of the image.

(a)   (b)

Fig. 9: A 10000 event sample of the density function from Figure 7 is shown in (a). The contour of the segmented dense region is shown in (b). The choice of parameters is: $\mu = 0.075$, timestep is 7.5 with the 2-step adaptive resolution. The data points in this figure are manually enhanced for the purpose of more clear display of the image.

Fig. 10: The toy density map of the Orange County Coastline, featuring two discon-
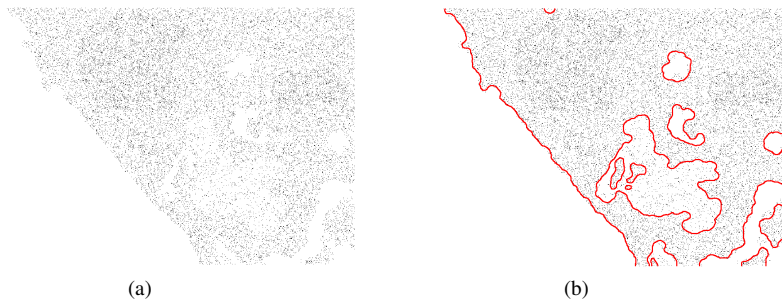nected dense regions, and one sparse region.

Fig. 11: A 20000 event sample of the density function from Figure 10 is shown in (a).
The contour of the segmented dense region is shown in (b). The choice of parameters
is: $\mu = 0.03$, timestep is 2.2 with the 2-step adaptive resolution.

6.3 San Fernando Valley Residential Burglary Data

We may also consider a special case of our problem, the case when we assume all
events are located inside the region Σ, and our goal is to segment the region. The
region Σ would represent a valid region of the given data set. In absence of geo-
graphic data that describe the location of the valid region, an accurate estimate of it
can dramatically improve accuracy of the density estimation, see [29] and [23]. In
the following example, our goal was to, without using any spatial information, seg-
ment the valid region from the San Fernando Valley residential burglary data. The
events in Figure 12(a) represent locations where burglaries took place during 2004
and 2005. The contour of our valid region estimate obtained by applying the special
case model is also shown in Figure 12(a). Smith et al. in [29] performed the valid
region estimate using census and other types of data to locate the residential area in
the region of interest, and their result is Figure 12(b). They incorporated the valid
region estimate from Figure 12(b) in their Weighted H^1 MPLE model to obtain the
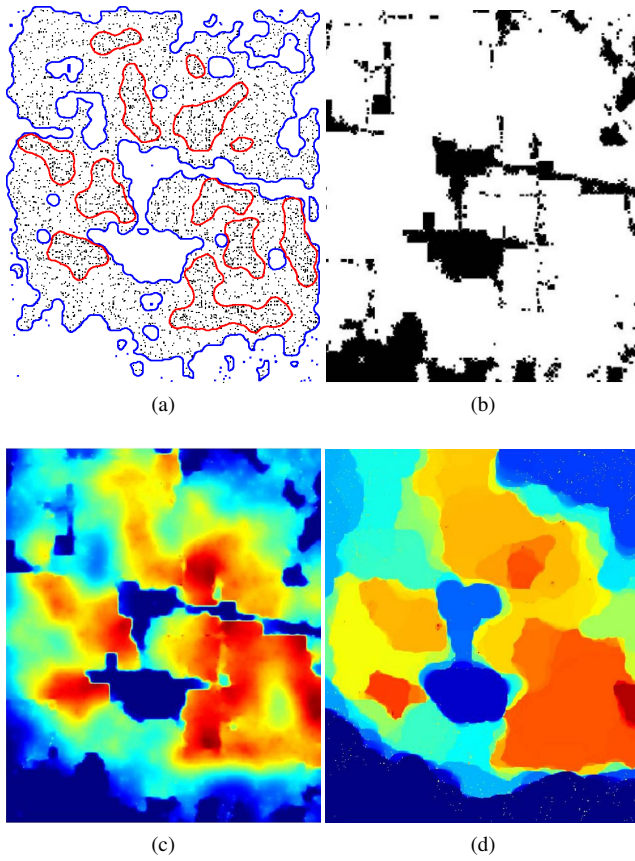density estimate results from 12(c). The TV MPLE algorithm developed by Mohler

Fig. 12: San Fernando Valley residential burglary: (a) A plot of the data set with the contours of the solution, the red contour bounds the dense region, while the blue one bounds the valid region . The size of the original data set is 605×525 pixels. We used a timestep 2.0 with $\mu = 1.1$ and 3-step adaptive resolution to find the valid region, and $\mu = 0.08$, a timestep 3.5 with 1-step adaptive resolution to locate the dense region. The data points in this figure are manually enhanced for the purpose of more clear display of the image. (b) A valid region estimate from [29]. (c) and (d) are density estimates form [29] and [23] respectively.

et al. in [23], was used to generate the density estimate in 12(b). This method did not use any additional spatial information to locate the valid region.

7 *V*-fold cross validation

The choice of the smoothing parameter μ and the timestep can affect the performance of this algorithm. Our experiments show that in case we are segmenting the high den-

sity region the optimal value of the parameter μ is not larger than 0.2. When our goal is to estimate the valid region, we typically assign larger values to the parameter μ. To estimate the value of the smoothing parameter we implemented a version of the V-fold cross validation algorithm. In their work [28] Sardy and Tseng proposed the V-fold cross validation based on the Kullback-Leibler information. In the V-fold cross validation, the original data set is partitioned into V disjoint subsets $\mathbf{x}_v = \{x_i, i \in S_v\}$ where S_v consists of all indexes of data points from the partition $v = 1, \ldots V$. Set $\mathbf{x}_{-v} = \{x_i, i \notin S_v\}$ is used as a *training set*, i.e. the algorithm is applied on $\mathbf{x}_{-v}$ with some particular value $\hat{\mu}$, and the density $\hat{d}_{\hat{\mu}, -v}$ is estimated. Set $\mathbf{x}_v$ is a *validating set*, which means that $\{\hat{d}_{\hat{\mu}, -v}(x_i)\}_{i \in S_v}$ is used to estimate the density on $\mathbf{x}_v$. Following these observations, the authors of [28] proposed the following estimate of Kullback-Leibler information $CV(\hat{\mu}) = -\sum_{v=1}^{V} \sum_{i \in S_v} \hat{d}_{\hat{\mu}, -v}(x_i)$, and after the search of the set of parameters $\hat{\mu}$ the one that minimizes this quantity is selected. However, $CV(\hat{\mu})$ uses only the log-likelihood to predict the performance of the model for some value $\hat{\mu}$ of the smoothing parameter, but does not take the H^1 norm of the segmentation function of the estimated density into account. We denote the segmentation function that corresponds to the density $\hat{d}_{\hat{\mu}, -v}$ by $\hat{u}_{\hat{\mu}, -v}$. Since the segmentation function is a binary function, the descrete H^1 and the descrete TV norm are equivalent, thus the H^1 norm also measures the length of the front between two phases. In some applications, such as the case when the dense and the surrounding sparse region have similar densities, the values of $CV(\mu)$ for the different values of μ tend to be similar. It is useful in those cases to also measure the H^1 norm of the obtained segmentation $\hat{u}_{\hat{\mu}, -v}$, and incorporate that information in the V-fold cross validation. Because of that, we propose a slightly different technique, where we evaluate $CV_{H^1}(\hat{\mu}) = -\sum_{v=1}^{V}(\sum_{i \in S_v} \hat{d}_{\hat{\mu}, -v}(x_i) - \xi \int |\nabla \hat{u}_{\hat{\mu}, -v}|^2)$ for each value of $\hat{\mu}$ from some proposed set of parameters, and select the value that minimizes it. The results do not appear to be very sensitive to ξ, we used small values, comparable to those of $\hat{\mu}$. The evaluation of $CV_{H^1}(\mu)$ for a single value of parameter μ requires V different density estimates, which could cause the V-fold cross validation to be very computationally intense. However, all density estimates that have to be performed are independent of each other, which makes the V-fold cross validation a perfect candidate for parallelization. In this implementation, we used 10-fold cross validation, and the process of calculating CV_{H^1} is parallelized using 10 threads, which reduces the computational time of one evaluation of CV_{H^1} down to the computational time needed for one density estimate. The computational time one density estimate takes varies from $0.2s$ in the small scale examples (100×100 pixels) to around one minute in the large scale examples (600×1000 pixels). To demonstrate the performance of the proposed algorithm, Figures 13 and 14 show some computational examples with segmentations generated using a model with the smoothing parameter obtained through 10-fold cross validation. To find the parameter μ we performed the linear search of intervals.
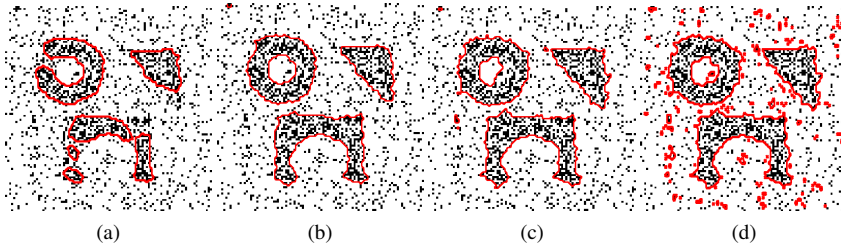
Fig. 13: The data set shown in 3 and the results obtained using different values of μ:(a) Smoothing parameter is too small $\mu = 0.12$. (b) We used μ obtained through our 10-fold cross validation with the H^1 norm, we evaulated CV_{H^1}. (c) 10-fold cross validation from [28], $CV(\mu)$, was applied to obatain μ used in this example. (d) Smoothing parameter $\mu = 0.3$, the value is larger than optimal.

Fig. 14: 10-fold cross validation was applied on the data sets displayed in 6(a) and 9(a) to select the parameter μ. The data sets with the countours of the respective dense regions obtained using those values of μ are shown.

8 Conclusion

This work demonstrates that threshold dynamics methods for image segmentation are a powerful tool for statistical density estimation for problems involving two dimensional geographic information. The efficiency of the method, especially when combined with multi-resolution techiques makes this a practical choice for parameter estimation involving V-fold cross validation, especially when parallel platforms are available. The method is a binary segmentation method that also determines density values. However, it can be naturally generalized to multi-level segmentation. One way to achieve that may include representing the segmentation function as a linear combination of the multiple binary components, similarly to the idea used for generalizing binary to grayscale inpainting in [7].However, this requires sufficient data to warrant a multi-level segmentation.

9 Aknowlegdments

References

1. A.L. Bertozzi, M.P. Brenner, T.F. Dupont, and L.P. Kadanoff, *Singularities and Similarities in Interface Flows*, Trends and perspectives in applied mathematics, Springer-Verlag New York, Inc. New York, NY, USA, (1994).
2. E.S. Brown, T.F. Chan, and X. Bresson, *A Convex Relaxation Method for a Class of Vector-valued Minimization Problems with Applications to Mumford-Shah Segmentation*, UCLA CAM Report 10-43, (2010).
3. Andrea L. Bertozzi, Selim Esedoglu, and Alan Gillette, *Inpainting of binary images using the Cahn-Hilliard equation*, IEEE Transactions on Image Processing, 16(1):285-291, (2007).
4. T.F. Chan, M. Nikolova, and S. Esedoglu, *Algorithms for finding global minimizers of image segmentation and denoising models*, SIAM Journal on Applied Mathematics , 66, 16-32 (2006).
5. T.F. Chan and L.A. Vese *Active contours without edges* IEEE Transactions on Image Processing, (10)2:266-277, (2001).
6. G. Chung and L.A. Vese, *Image segmentation using a multilayer level-set approach*, Computing and visualization in science 12 , no. 6, 267285, (2009).
7. J.A. Dobrosotskaya and A.L. Bertozzi, *A wavelet-Laplace variational technique for image deconvolution and inpainting*, IEEE Transactions on Image Processing, 17(5):657-663, (2008).
8. P.P.B. Eggermont and N.V. LaRiccia, *Maximum Penalized Likelihood Estimation*, Springer, (2001).
9. S. Esedoglu and Y.R. Tsai, *Threshold Dynamics for the Piecewise Constant Mumford-Shah Functional*, Journal of Computational Physics, 211(1), 367-384, (2006).
10. S. Esedoglu and R. March, *Segmentation with depth but without detecting junctions*, J. Math. Imaging Vision, 18(1):7-15, 2003. Special issue on imaging science, (2002).
11. S. Esedoglu, S.J. Ruuth and R. Tsai *Diffusion generated motion using signed distance functions*, Journal of Computational Physics, 229(4), 1017-1042, (2010).
12. S. Esedoglu, S.J. Ruuth and R. Tsai, *Threshold Dynamics for High Order Geometric Motions*, Interfaces and Free Boundaries, 10(3): 263-282, (2008).
13. R.A. Fisher, *On the mathematical foundations of theoretical statistics*, Philos. Trans. Royal. Soc. London A 222, 309-360, (1922).
14. S. Gao and T.D. Bui, *Image segmentation and selective smoothing by using Mumford-Shah model*, Image Processing, IEEE Transactions on 14(10), 15371549, (2005).
15. T. Goldstein and S. Osher, *The Split Bregman method for L^1-regularized problems*, SIAM J. Imaging Sci., 2(2), 323-343, (2009).
16. F. Gibou and R. Fedkiw, *A Fast Hybrid k-Means Level Set Algorithm For Segmentation*, Preceedings of the 4th Annual Hawaii International Conference on Statistics, (2005).
17. G. Chung and L.A. Vese, *Image segmentation using a multilayer level-set approach*, Computing and Visualization in Science, 12(6), 267285, (2009).
18. Koenker, R. and Mizera, I. *Density estimation by total variation regularization*, Advances in Statistical Modeling and Inference, Essays in Honor of Kjell A. Doksum, World Scientific, 613-634, (2006).
19. R. V. Kohn and P. Sternberg, *Local minimisers and singular perturbations*, Proc. Roy. Soc. Edinburgh Sect. A, 111(1-2),69-84, (1989).
20. J. Lie, M. Lysaker and X.C. Tai, *A binary level set model and some applications to Mumford-Shah image segmentation*, Image Processing, IEEE Transactions, 15(5), 11711181, (2006).
21. B. Merriman, J. Bence and S. Osher, *Diffusion Generated Motion by Mean Curvature*, In AMS Selection in Math.,The Computational Crystal Growers, 73-83 , (1992).
22. B. Merriman and S.J. Ruuth, *Diffusion Generated Motion of Curves on Surfaces*, J. Comput. Phys., 225(2), 2267-2282, (2007).

23. G. Mohler, A. Bertozzi, T. Goldstein and S. Osher, *Fast TV Regularization for 2D Maximum Penalized Likelihood Estimation*, Journal of Computational and Graphical Statistics, 20(2), 479, (2011).

24. D. Mumford and J. Shah, *Optimal approximations by piecewise smooth functions and associated variational problems*, Comm. Pure Appl. Math., 42, 577-658, (1989).

25. S. Osher and J. A. Sethian, *Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations*, J. Comput. Phys., 79, 1249,(1988).

26. J. Rubinstein, P. Sternberg and J. B. Keller,*Fast reaction, slow diffusion, and curve shortening*, SIAM J. Appl. Math. 49(1), 116-133, (1989).

27. S.J. Ruuth, *Efficient algorithms for diffusion-generated motion by mean curvature* Journal of Computational Physics, 144, 603-625, (1998).

28. S. Sardy, P. Tseng, *Density estimation by total variation penalized likelihood driven by the sparsity l1 information criterion* Scandinavian Journal of Statistics, 37(2), 321-337, (2010).

29. L. Smith, M. Keegan, T. Wittman, G. Mohler and A. Bertozzi, *Improving Density Estimation by Incorporating Spatial Information* URASIP Journal on Advances in Signal Processing Volume 2010.

30. L.A. Vese and T.F. Chan, *A multiphase level set framework for image segmentation using the Mumford and Shah model*, International Journal of Computer Vision 50(3), 271293, (2002).

31. B. Song,T. Chan, *A fast algorithm for level set based optimization*. UCLA