

On Parameter Estimation Using Level Sets

Jordan M. Berg
Department of Mechanical Engineering
Texas Tech University
Lubbock, TX 79409-1021
jberg@ttu.edu

18 February 1998

Abstract

Consider the problem of selecting the member of a parametrized family of curves that best matches a given curve. This is a key step in determining proper values for adjustable parameters in low-order plasma etching and deposition models. Level set methods offer several attractive features for treating such problems. This paper presents a parameter estimation scheme that exploits the level set formulation. The method is completely geometric; there is no need to introduce an arbitrary coordinate system for the curves.

1 Introduction

This work is motivated by the need for accurate low-order phenomenological models of thin film etching and deposition processes. These processes are central to the manufacture of microelectronic devices. Phenomenological models are necessary because of the complexity of the surface chemistry and the plasma-surface interactions. Typically these models lump together numerous unknown rate constants into relatively few parameters [7, 11, 17]. Reliable use of these models for simulation or control then depends on the ability of the user to choose the values of these parameters correctly. The parameter values may be selected based on surface evolution data from scanning electron micrographs. To our knowledge, only one study has investigated methods to optimize this process [7]. That work, while quite successful, used a non-geometric cost function that requires the user to select points in one-to-one correspondence on the actual and estimated surfaces. This selection introduces an arbitrary component of unknown significance into the procedure, and places an undesirable burden on the user. In the following sections we introduce a completely coordinate free cost function that eliminates this arbitrary element.

In a recent series of papers Sethian and Adalsteinsson apply level set methods to the simulation of feature development in a variety of semiconductor manufacturing applications [1, 15, 16]. Level set methods provide a flexible framework for surface evolution problems. The main alternatives to level set methods are “string” methods. Here the surface is defined

by points, which are advanced according to the surface velocity. Two problems may occur if corners appear in the surface. First, if the surface points are allowed to “bunch up” at the corner, the CFL condition for numerical stability may be violated. Second, points on the surface may move past each other, necessitating “delooping”. The level set framework is inherently immune to these problems. String methods may also become unwieldy when topological transitions occur, as during merging or splitting of surfaces. Again, these transitions are handled automatically using level sets. For a detailed discussion and a survey of the literature see [15]. Both corner formation and topological transition occur in etching and deposition, therefore these are promising areas for the application of level set methods.

Although we are motivated by plasma etching and deposition applications, as described above, the problem we attack here is more narrow in scope. We describe a geometric cost function based on level set descriptions of both the curve to be matched and the parametrized family. We then construct derivatives of the cost function in terms of the parameters. This allows us to apply the gradient descent class of minimization methods. A simple example is presented, and the tasks remaining before the results can be applied to the full problem are discussed. Several important properties of the cost function are discussed. Perhaps most important is that the cost function is not differentiable at the optimum.

Level set methods have been used in identification. Approaches to real-time estimates of evolving features in plasma etching based on level sets have been put forth in [2, 3, 4, 5]. Santosa devises a level set approach to reconstruct the shape of an unknown object from a discrete set of measurements [13]. This is somewhat different from our situation. In this paper we treat the case where the shape of the object is itself the measurement.

2 Level Sets

The approach presented here is general, but in this paper we restrict our attention to curves in the plane. In the level set formulation, a curve \mathcal{C} is represented by the zero level set (ZLS) of a level set function (LSF) $\Phi(x)$, that is, $\mathcal{C} = \{x \in R^2 : \Phi(x) = 0\}$. Clearly the choice of Φ is not unique. To remove this non-uniqueness we will think of Φ as defining an *equivalence class* of continuous functions, where two such functions are defined to be equivalent if they have the same *signature*, that is,

$$\Phi(x) \sim \Psi(x) \Leftrightarrow \text{sign}(\Phi(x)) = \text{sign}(\Psi(x)) \forall x \in R^2 \quad (1)$$

It is easy to show that this is indeed an equivalence relation, and that it preserves both the ZLS itself, and the sense of “inside” and “outside” of the corresponding curve. As usual when dealing with equivalence classes, it is convenient to choose a canonical element. In this case that element will be the *signed distance function*, Ξ . In defining the signed distance function we assume the existence of a norm, denoted $\|\cdot\| : R^2 \rightarrow R$. Then, given a curve \mathcal{C} ,

$$\Xi(x) = \pm \min_{y \in \mathcal{C}} \|y - x\|$$

where $\Xi(x)$ is negative if x is *inside* \mathcal{C} , and positive if x is *outside* \mathcal{C} . Efficient algorithms exist for generating the signed distance function given an arbitrary LSF, particularly when allowed freedom in the choice of norms.

The real importance of the level set approach comes when considering *evolving* curves. Here $\mathcal{C} : [0, 1] \times [0, t_f] \rightarrow R^2$ is a parametrized curve evolving in time according to the equation

$$\frac{\partial \mathcal{C}}{\partial t} = \tilde{\beta}(s, t, \dots) \nu \quad (2)$$

where ν is the *outward pointing normal* to \mathcal{C} . The *speed function* $\tilde{\beta}$ describes the outward normal velocity of \mathcal{C} , and may depend on independent variables, on local properties of the curve, or on global considerations. In Eq. (2), $\tilde{\beta}$ is defined only on the curve. Now the LSF too is time-dependent. The evolution of the LSF according to Eq. (2) is governed by the following PDE:

$$\Phi_t + \beta(x, t, \dots) \|\nabla \Phi\| = 0. \quad (3)$$

This PDE is derived as follows: The curve $\mathcal{C}(s, t)$ is represented by the zero level set of a smooth and Lipschitz continuous function $\Phi : R^2 \times [0, \tau] \rightarrow R$. Assume that Φ is negative in the interior and positive in the exterior of the zero level set. We consider the zero level set, defined by

$$\{X(t) \in R^2 : \Phi(X, t) = 0\}. \quad (4)$$

We have to find an evolution equation of Φ , such that the evolving curve $\mathcal{C}(t)$ is given by the evolving zero level $X(t)$, i.e.,

$$\mathcal{C}(t) \equiv X(t). \quad (5)$$

By differentiating $\Phi(X(t), t) = 0$ we obtain:

$$\nabla \Phi(X, t) \cdot X_t + \Phi_t(X, t) = 0. \quad (6)$$

Note that for the zero level, the following relation holds:

$$\frac{\nabla \Phi}{\|\nabla \Phi\|} = \nu. \quad (7)$$

Combining Eq. (2) and Eq. (4) through Eq. (7) gives Eq. (3).

All LSFs in the equivalence class must satisfy an equation like Eq. (3). Here β is a function defined everywhere in the plane. Different choices for β are possible, but at any instant they must all coincide on the ZLS itself, and there be equal to $\tilde{\beta}$. We will call any β satisfying this condition *admissible*. To remove the non-uniqueness in the definition of Eq. (3) we again turn to the equivalence relation Eq. (1). Any admissible β will map a member of the equivalence class of LSFs at time t_0 into the proper equivalence class at time t . The canonical signed distance function can be recovered from any other member of the class as desired. This process is often referred to as *renormalization*.

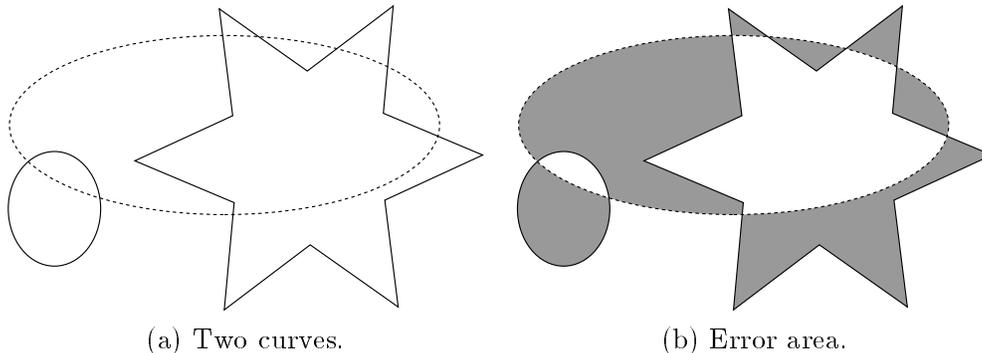


Figure 1: Two curves. One (dashed) is a large ellipse. The other (solid) consists of two simply connected components, a star and a small ellipse. The distance between the curves is found by summing the areas of the shaded regions.

3 A Metric for Level Set Functions

The process of parameter identification will require that we find the parameter values that give, in some sense, the closest match to an observed evolution. To make this rigorous, we must define a metric for LSFs that formalizes the idea of “distance.” We turn now to the construction of a suitable function. Our main objective here is to avoid the need to parametrize the curves. Such parametrizations are intrinsically arbitrary, and place a burden on the experience and expertise of the end user. This is the motivation for defining a geometric or coordinate-free cost function. The idea is presented in Fig. 1. Figure 1(a) shows two closed curves, one consisting of a single connected component, the other consisting of two connected components. In all cases “inside” is the bounded portion of the curves. Since the two curves are not identical, there should be a positive distance between them. We define this distance to be the area of the region between the two curves. This area is shaded light gray in Fig. 1(b). We refer to it as the *error area*.

Formally, given a pair of simple closed curves, \mathcal{C}_1 and \mathcal{C}_2 , let $A(\mathcal{C}_1, \mathcal{C}_2)$ be the total area enclosed by either one curve or the other, but not both. In terms of sets, if I_1 is the interior of the first curve, and I_2 is the interior of the second curve, then A is the Lebesgue measure of the symmetric difference $S(I_1, I_2) := (I_1 \setminus I_2) \cup (I_2 \setminus I_1)$. The symbol \setminus denotes the usual set difference. This is indeed a metric on sets [12], and so on closed curves. In higher dimensions area is replaced by volume, etc. The definition is easily extended to any curve defined by a LSF on a compact domain by including appropriate portions of the boundary.

The example shown in Fig. 1 is abstract. A geometry that might occur in thin film deposition into a trench or via is shown in Fig. 2. Here the points contained inside both curves are shaded dark, the points outside both curves are unshaded, and the points inside one curve but not the other—that is, the points making up the error area—are shaded a light gray.

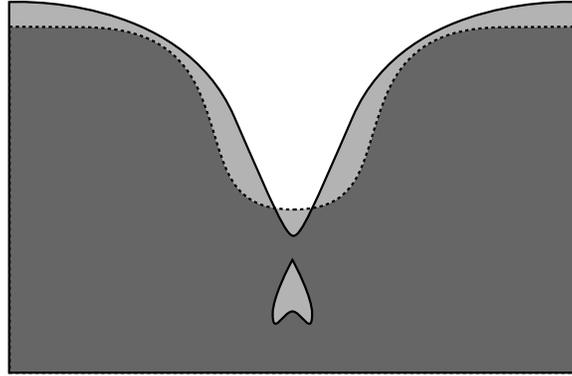


Figure 2: The estimate (dashed) differs from the measurement (solid). The associated cost is then the area between the two curves (light gray).

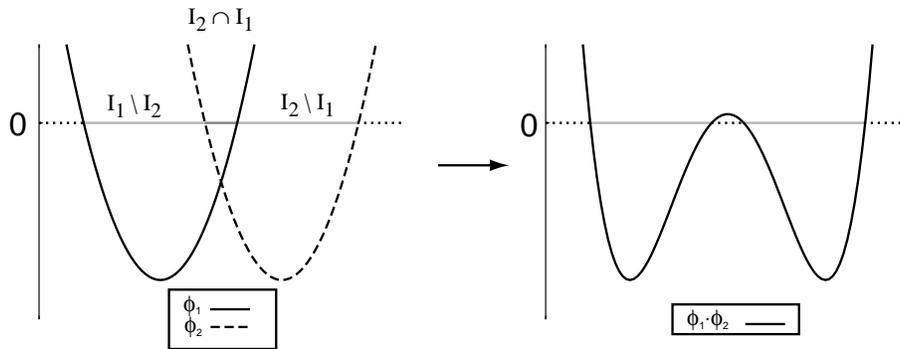


Figure 3: Product method for constructing error area from level sets.

3.1 The Product Method

When the estimate and the measurement curve are both characterized by LSFs, the metric defined above is easy to calculate. If the two LSFs are multiplied pointwise, the result is a new LSF, which we call the *product LSF*. The ZLS of the product LSF defines a curve, and the area of the interior of this curve is exactly the desired value. To see this, let \mathcal{C}_1 be the ZLS of $\Phi(x)$, and \mathcal{C}_2 be the ZLS of $\Psi(x)$. Recall that the distance between \mathcal{C}_1 and \mathcal{C}_2 is the measure of the symmetric difference between their interiors. That is, in terms of the LSFs, the measure of the set of points for which one LSF takes a positive value, while the other takes a negative value. If we form $\Gamma(x) = \Phi(x)\Psi(x)$, we see that $A(\mathcal{C}_1, \mathcal{C}_2)$ is just the area of the interior of the ZLS corresponding to $\Gamma(x)$. Figure 3 is a graphical depiction of this construction in one dimension. To calculate the area we generate \mathcal{C}_3 —the contour corresponding to the ZLS of $\Gamma(x)$ (note that the contour is now composed of *oriented curves* and is no longer just a set)—and apply Green’s Theorem:

$$A = \frac{1}{2} \int_{\mathcal{C}_3} \langle \mathcal{C}_3, \nu \rangle ds \quad (8)$$

where $\langle \cdot, \cdot \rangle$ is the vector inner product. Here, as elsewhere in this paper, s stands for the arclength parameter [8]. Note that it is necessary to assign an LSF to the measurement. There are many ways to do this, but our choice is to use the canonical LSF, that is, the signed distance function.

In the procedure above, generating the contour corresponding to the ZLS of $\Gamma(x)$ is a key step. A geometric algorithm to accomplish this has been developed by Siddiqi, *et al.*, and is presented in [14].

3.2 The Symmetric Difference Method

In the preceding section the product LSF was used to define the error area. In this section we present an alternative that has better numerical properties. Consider the interior of the ZLS as a set, and note the following relationships between operations on LSFs and set operations on the interiors.

1. Given a LSF Φ , and denoting the interior of the ZLS of Φ as $\text{int}(\Phi)$, then $\text{int}(-\Phi) = \text{int}(\Phi)^c$. That is, the interior of the negative of the ZLS is the *complement* of the interior of the ZLS.
2. Given two LSFs, Φ and Ψ , then $\text{int}(\max(\Phi, \Psi)) = \text{int}(\Phi) \cap \text{int}(\Psi)$.
3. Given two LSFs, Φ and Ψ , then $\text{int}(\min(\Phi, \Psi)) = \text{int}(\Phi) \cup \text{int}(\Psi)$.

The symmetric difference between sets A and B is defined as $S(A, B) = (A \cap B^c) \cup (B \cap A^c)$. It can be easily shown that $S(A, B) = (A \cup B) \cap (A \cap B)^c$. Identifying A with $\text{int}(\Phi)$ and B with $\text{int}(\Psi)$, we define

$$\Theta = \max(\underline{\theta}, -\bar{\theta}) \quad (9)$$

where,

$$\underline{\theta} = \min(\Phi, \Psi) \tag{10}$$

$$\bar{\theta} = \max(\Phi, \Psi) \tag{11}$$

And it is easily seen that the error area is the Lebesgue measure of the interior of Θ . We refer to Θ as the symmetric difference LSF.

This formulation is numerically preferable to that of the previous section because if the gradient of the LSFs are near unity everywhere, then the gradient of the symmetric difference LSF will be also. In contrast, the gradients of the product LSF are not preserved. However, even for the symmetric difference method, away from the ZLS the signed distance function is not preserved. To see this, consider the case of the two LSFs, $\Phi(x, y) = x$ and $\Psi(x, y) = y$, whose interiors are the left half-plane and upper half-plane, respectively. These LSFs correspond to the signed distance function in the 1-norm or 2-norm. The union of their interiors is the interior of the LSF $\Theta = \min(\Phi, \Psi)$. The gradient of Θ has magnitude one everywhere on the ZLS, except at the corners. To see that the signed distance function is not preserved, consider the point $(-1, -1)$. Although its distance from the ZLS is 2 (in the 1-norm) or $\sqrt{2}$ (in the 2-norm), the value of Θ at $(-1, -1)$ is -1 .

4 Parametrized Level Sets

Ultimately we wish to parametrize the speed function β . Here however, we consider the simpler case of a parametrized level set. For the moment also we suppress the time dependence of the curves. Our objective then is as follows: We are given a *measured* curve, \mathcal{M} , which is the ZLS of $\Phi(x)$. We also have a parametrized family of level set functions, $\Psi(x; \lambda)$. Let $\mathcal{L}(\lambda)$ be the ZLS of $\Psi(x; \lambda)$. We wish to find the value of the parameter vector λ such that $\mathcal{L}(\lambda)$ is closest to \mathcal{M} .

To proceed, we treat the metric as a cost function, and seek to minimize

$$J(\lambda) = A(\mathcal{M}, \mathcal{L}(\lambda)) \tag{12}$$

Or, in terms of Green's Theorem, to minimize

$$J(\lambda) = \frac{1}{2} \int_{\mathcal{E}} \langle \mathcal{E}, \nu \rangle ds \tag{13}$$

where \mathcal{E} is the ZLS of

$$\Gamma(x; \lambda) := \Phi(x) \Psi(x; \lambda). \tag{14}$$

We wish to apply (at least) gradient descent methods to accomplish this minimization. To do this, we need to calculate the sensitivity derivatives J_i , where we denote partial differentiation with respect to λ_i by a subscript i .

For the smooth segments of the curve we have

$$J_i(\lambda) = \int_{\mathcal{E}} \langle \mathcal{E}_i, \nu \rangle ds \tag{15}$$

Because the derivation of Eq. (15) may not be obvious, here is a brief sketch. Let θ parametrize the curve, and take values from 0 to 2π regardless of the value of λ . Of course, this parameter is, in general, no longer the arclength. Now, writing \mathcal{E} explicitly as (x, y) , the tangent vector explicitly as $(x_\theta, y_\theta)/\sqrt{x_\theta^2 + y_\theta^2}$, and the outward normal as $(-y_\theta, x_\theta)\sqrt{x_\theta^2 + y_\theta^2}$,

$$J(\lambda) = \frac{1}{2} \int_0^{L(t;\lambda)} \langle \mathcal{E}, \nu \rangle ds \quad (16)$$

$$= \frac{1}{2} \int_0^{2\pi} \langle \mathcal{E}, (-y_\theta, x_\theta) \rangle d\theta \quad (17)$$

Differentiating inside the integral, and using the fact that the limits of integration no longer depend on λ ,

$$J_\lambda(\lambda) = \frac{1}{2} \int_{\mathcal{E}} \langle \mathcal{E}_\lambda, \nu \rangle ds + \frac{1}{2} \int_0^{2\pi} \langle \mathcal{E}, (-y_{\theta\lambda}, x_{\theta\lambda}) \rangle d\theta \quad (18)$$

Differentiation by λ and by θ commute. Integration of the second term by parts gives

$$\frac{1}{2} \int_0^{2\pi} \langle \mathcal{E}, (-y_{\theta\lambda}, x_{\theta\lambda}) \rangle d\theta = \langle \mathcal{E}, (-y_\lambda, x_\lambda) \rangle \Big|_0^{2\pi} - \frac{1}{2} \int_0^{2\pi} \langle \mathcal{E}_\theta, (-y_\lambda, x_\lambda) \rangle d\theta \quad (19)$$

The leading term in this expression is zero because the curve is closed. To evaluate the remaining term, write the vectors explicitly.

$$-\frac{1}{2} \int_0^{2\pi} \langle \mathcal{E}_\theta, (-y_\lambda, x_\lambda) \rangle d\theta = -\frac{1}{2} \int_0^{2\pi} \langle (x_\theta, y_\theta), (-y_\lambda, x_\lambda) \rangle d\theta \quad (20)$$

$$= -\frac{1}{2} \int_0^{2\pi} (-x_\theta y_\lambda + x_\lambda y_\theta) d\theta \quad (21)$$

$$= \frac{1}{2} \int_0^{2\pi} \langle (x_\lambda, y_\lambda), (-y_\theta, x_\theta) \rangle d\theta \quad (22)$$

$$= \frac{1}{2} \int_{\mathcal{E}} \langle \mathcal{E}_\lambda, \nu \rangle ds \quad (23)$$

Substituting this result into Eq. (18) gives Eq. (15).

Now consider the change in \mathcal{E} corresponding to a change in the i -th parameter λ_i . We write

$$\Gamma(X(s; \lambda); \lambda) = 0$$

So,

$$\Gamma_i + \langle \nabla \Gamma, \mathcal{E}_i \rangle = 0$$

In fact, differentiation with respect to a parameter gives the same form we found when we derived the evolution equation, namely

$$\Gamma_i + \langle \mathcal{E}_i, \nu \rangle \|\nabla \Gamma\| = 0$$

This time, however, we arrange the terms as follows:

$$\langle \mathcal{E}_i, \nu \rangle = -\frac{\Gamma_i}{\|\nabla \Gamma\|} \quad (24)$$

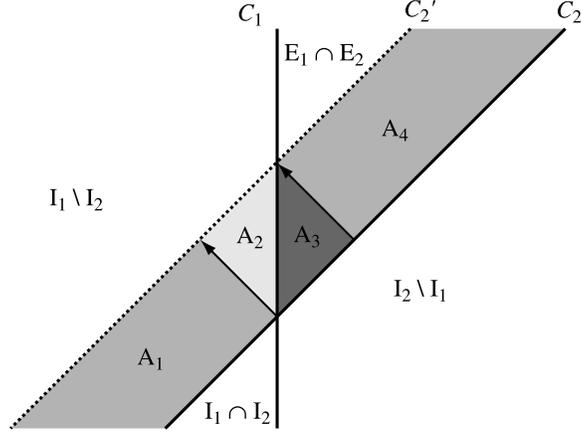


Figure 4: Close-up of corner effects.

Now, substituting Eq. (24) into Eq. (15), we obtain

$$J_i(\lambda) = - \int_{\mathcal{L}} \Gamma_i / \|\nabla \Gamma\| ds \quad (25)$$

The derivatives of Γ are replaced by $\Gamma_i = \Phi \Psi_i$, and $\nabla \Gamma = \Phi \nabla \Psi + \Psi \nabla \Phi$. Since $\Phi = 0$ on \mathcal{M} and $\Psi = 0$ on \mathcal{L} ,

$$J_i = - \int_{\mathcal{M} \cup \mathcal{L}} \Phi \Psi_i / \|(\Phi \nabla \Psi + \Psi \nabla \Phi)\| ds \quad (26)$$

$$= - \int_{\mathcal{L}} \Phi \Psi_i / \|\Phi \nabla \Psi\| ds \quad (27)$$

$$= - \int_{\mathcal{L}} \text{sign}(\Phi) \Psi_i / \|\nabla \Psi\| ds \quad (28)$$

This formula applies only to the smooth portion of the curves, and the contours will fail to be smooth at a finite number of corners. If the original curves are smooth, which we assume here for convenience, then the error contours defined by the ZLS of the product LSF, will have corners only where the original curves intersect. It can be shown that to first order the contribution at each of these corners is zero. To see this, consider Fig. 4. Here, I_1 , the interior of C_1 , lies to the left of C_1 , and I_2 , the interior of C_2 , lies to the right of C_2 . The exteriors are denoted by E_1 and E_2 , respectively. The affect of moving C_2 to C_2' with C_1 stationary is to increase the area of $I_2 \setminus I_1$, decrease the area of $I_1 \setminus I_2$, increase the area of $I_1 \cap I_2$, and decrease the area in $E_1 \cap E_2$. The change in the error area predicted by a first order analysis is as follows: the error area will decrease by A_1 , due to the reduction in $I_1 \setminus I_2$, and increase by $A_2 + A_3 + A_4$, due to the increase in $I_2 \setminus I_1$. However, due to the corner, the actual decrease in $I_1 \setminus I_2$ is greater than expected by A_2 , and at the same time, the actual increase in $I_2 \setminus I_1$ is less than expected by A_2 . The effects cancel, and the net result is that the first order analysis requires no modification at the corner.

4.1 The Symmetric Difference Method

To derive sensitivity expressions for the symmetric difference method for the error area, we replace the product LSF Γ in (14) with the symmetric difference LSF Θ from (9). This substitution does not affect the calculation until Eq. (25). There we find

$$\Theta_i = \begin{cases} \Psi_i; & \Psi = 0, \Phi > 0 \\ -\Psi_i; & \Psi = 0, \Phi < 0 \\ 0 & \text{otherwise} \end{cases} \quad (29)$$

where the case that both Φ and Ψ are zero along a measurable portion of the curve is neglected. This corresponds to a degenerate LSF, and the numerical scheme will typically not detect the contour, since no sign change occurs. To see that this expression is correct, note that along the ZLS of the symmetric difference LSF either Φ or Ψ is zero. As noted, we assume that they are not both zero on any line segment. When Φ is zero then the ZLS is locally determined only by Φ . Since Φ has no dependence on parameters, Φ_i , and so Θ_i , is zero. When $\Psi = 0$ the ZLS is locally determined only by Ψ . Then

$$\underline{\theta} = \min(\Phi, \Psi) = \begin{cases} \Phi; & \Phi < 0 \\ \Psi; & \Phi > 0 \end{cases}$$

$$\bar{\theta} = \max(\Phi, \Psi) = \begin{cases} \Psi; & \Phi < 0 \\ \Phi; & \Phi > 0 \end{cases}$$

And so,

$$\Theta = \max(\underline{\theta}, -\bar{\theta}) = \begin{cases} \max(\Phi, -\Psi); & \Phi < 0 \\ \max(\Psi, -\Phi); & \Phi > 0 \end{cases} = \begin{cases} -\Psi; & \Phi < 0 \\ \Psi; & \Phi > 0 \end{cases}$$

Taking the gradient gives

$$\nabla\Theta = \begin{cases} -\nabla\Psi; & \Phi < 0 \\ \nabla\Psi; & \Phi > 0 \end{cases}$$

So the final expression for $\partial J/\partial\lambda_i$ becomes,

$$J_i = - \int_{\mathcal{L}} \text{sign}(\Phi) \Psi_i / \|\nabla\Psi\| ds \quad (30)$$

That is, the expression for the sensitivity is the same for both the product LSF and the symmetric difference LSF. Since the computed value of the cost function is the same in both methods, and the sensitivity is also the same, it is natural to ask why one differs from the other. The answer is that the contour tracing step will give different results. Since the symmetric difference method results in a LSF whose intersection with any grid line is piecewise linear, the interpolation routines used to locate the zeros should give better results for this case. However the sensitivity calculation is an integral over the ZLS of the estimate only. Therefore the contour corresponding to the product LSF or symmetric difference LSF is used only in the calculation of the cost function itself.

5 Examples

Here we consider two simple examples. In the first, the measured curve \mathcal{M} is a circle of radius R . Choosing the canonical LSF to represent the curve, we write, $\Phi(x, y) = \sqrt{(x - x_0)^2 + (y - y_0)^2} - R$. As a parametrized LSF we choose,

$$\Psi(x, y; x_c, y_c, \Pi) = \sqrt{(x - x_c)^2 + (y - y_c)^2} - \Pi$$

Note that $\|\nabla\Psi\| = \|\nabla\Phi\| = 1$. The sensitivities are calculated exactly, as follows,

$$\Psi_{x_c} = -\frac{(x - x_c)}{\sqrt{(x - x_c)^2 + (y - y_c)^2}} \quad (31)$$

$$\Psi_{y_c} = -\frac{(y - y_c)}{\sqrt{(x - x_c)^2 + (y - y_c)^2}} \quad (32)$$

$$\Psi_{\Pi} = -1 \quad (33)$$

In the optimal solution the curves are the same, and the cost function is zero.

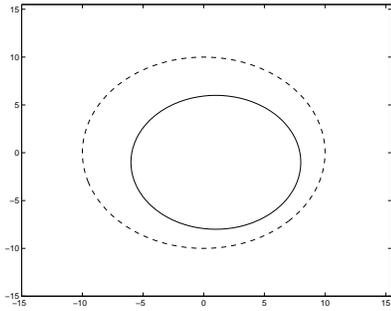
For the second example the parametrized level set is the same, but the measured curve is taken as an ellipse, that is, the ZLS of $\Phi(x, y) = \sqrt{a^2(x - x_0)^2 + b^2(y - y_0)^2} - R$. This leaves the calculations unchanged, but the optimal estimate no longer gives a perfect match. The gridsize in all the following calculations is 0.5 in both the x and y directions.

5.1 Derivative and Error Computation

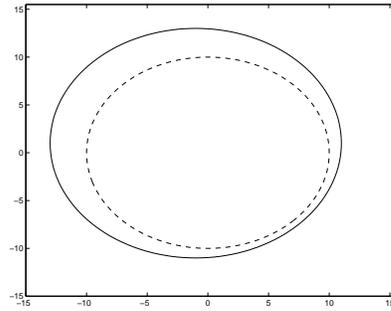
The computation of the error function and its sensitivity to parameter variation was checked using four static circular geometries. In the first, the estimated curve is a circle completely contained within the measured circle. In this case the error is the difference in the areas, the sensitivities J_{x_c} and J_{y_c} are zero, and J_{Π} is the negative of the circumference. In the second case, the positions are reversed, with the measurement contained in the estimate. The only difference between this and Case I is the sign of the radius sensitivity. In Cases III–V the estimated circle lies on the measured circle in such a way that half the boundary of the estimate lies inside the measured circle, and half outside. This should result in $J_{\Pi} = 0$. In Case III the center of the estimated circle is displaced from the center of the measurement only in the x direction, in Case IV, only in the y direction, and in Case V, in both x and y . Figure 5 shows the cases. Table 1 compares the computed results to the theoretical values. Because the sensitivity calculations depend only on the ZLS of the estimated LSF, the sensitivities are the same for the two calculation methods. However, the symmetric difference method appears to be the more accurate way to calculate the cost function itself.

5.2 Contour Tracing

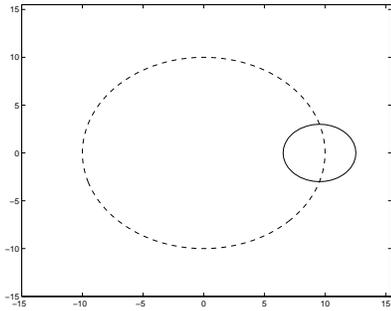
The error calculation becomes more difficult as the estimate approaches the true value. the reason is that the contour defining the error area becomes difficult to trace accurately. We next examine this behavior more closely, and compare the results from the product method



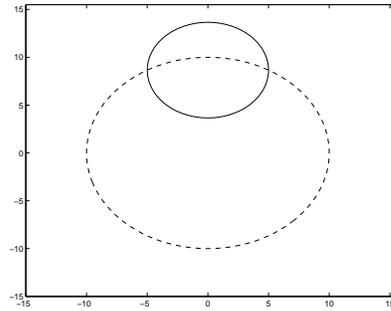
(a) Case I



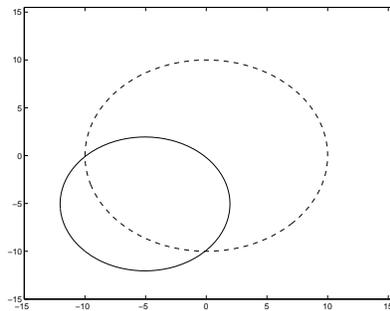
(b) Case II



(c) Case III

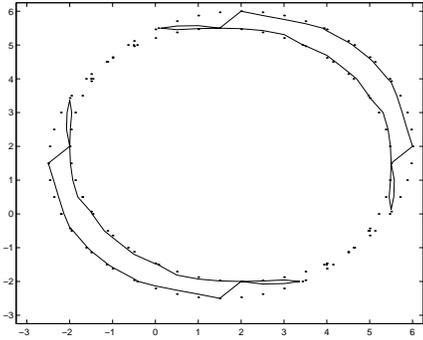


(d) Case IV

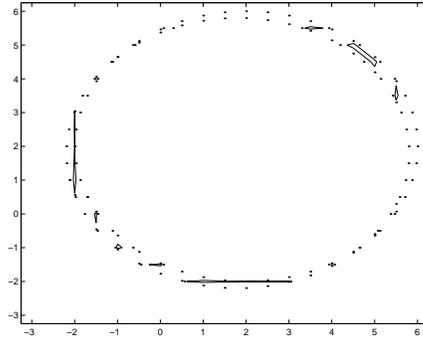


(e) Case V

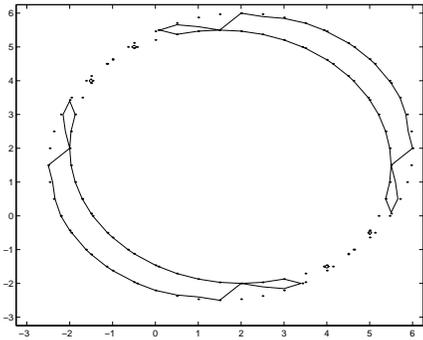
Figure 5: Cases for testing error and sensitivity computations.



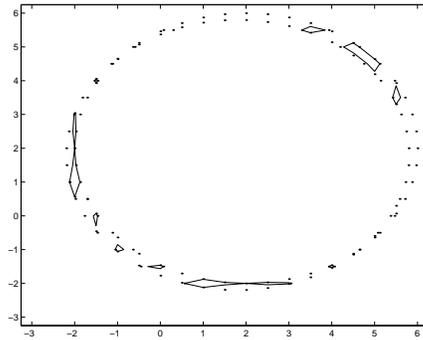
(a) Product method. $(x_c, y_c) = (1.5, 1.5)$, $J = 7.143$



(b) Product method. $(x_c, y_c) = (1.8, 1.8)$, $J = 0.2456$



(c) Symmetric difference method. $(x_c, y_c) = (1.5, 1.5)$, $J = 8.588$



(d) Symmetric difference method. $(x_c, y_c) = (1.8, 1.8)$, $J = 0.7756$

Figure 6: Contour tracing using (a,b) the product method and (c,d) the symmetric difference method. True and estimated curves are dotted. The true curve is a circle of radius 4 centered at $(2, 2)$. The estimated curve is a circle of radius 4. The centers are as noted. The resulting cost functions are also given.

Table 1: Computed Cost Function and Sensitivities vs. Actual Values ($h = 0.5$)

		J	J_{x_c}	J_{y_c}	J_{Π}
Case I	Theory	160.221	0	0	-43.982
	Product	159.083	0.000	0.000	-43.969
	Sym. Diff.	160.216	0.000	0.000	-43.969
Case II	Theory	138.230	0	0	75.398
	Product	135.274	0.000	0.000	75.390
	Sym. Diff.	138.183	0.000	0.000	75.390
Case III	Theory	310.457	12	0	0
	Product	310.119	11.991	0.000	0.160
	Sym. Diff.	310.314	11.991	0.000	0.160
Case IV	Theory	296.042	0	20	0
	Product	295.496	1.000	19.967	-0.359
	Sym. Diff.	295.869	1.000	19.967	-0.359
Case V	Theory	259.060	-19.799	-19.799	0
	Product	257.962	-19.002	-20.532	-1.091
	Sym. Diff.	258.725	-19.002	-20.532	-1.091

with the results from the symmetric difference method. The test case is a circle centered at $(2, 2)$ with a radius of 4. Figure 6(a) shows the true curve compared to a series of progressively closer estimates (both true and estimated curves are dotted) with the computed contours as calculated by the product method. Note that, because the computed contours are completely contained within the actual error area, the computed error area will be less than the actual value. Figure 6(b) shows the same test using the symmetric difference method. Although both methods struggle with the contour as the curves become close, the symmetric difference method does a better job—as is evident from the plots, and the higher computed values of J . The contour tracing algorithm implemented here is quite crude. Linear interpolation is used to locate zero crossings on the computational grid, and those points are connected to form contours. No shock placement logic is currently used. Improving the error calculation further requires either a smaller gridsize or a higher order interpolation method for contour tracing.

5.3 Numerical Properties

The first example, in which both the measured and estimated curves are circles, gives several useful insights into the numerical properties of this problem. Consider the case in which the circles are concentric, and the only free parameter is the estimated radius, Π . Then the cost function is $J = \pi|R^2 - \Pi^2|$, where R is the true radius. From this we see that $J_{\Pi} = (2\pi) \text{sign}(\Pi - R) \Pi$. Figure 7 shows this cost function. The function is nonsmooth, and would not appear to be well approximated by a quadratic at the optimal point. The first

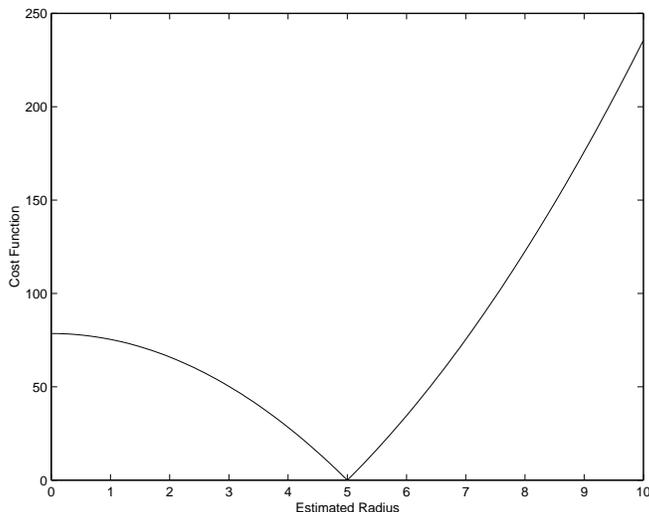


Figure 7: The cost function $J = \pi|R^2 - \Pi^2|$ for $R = 5$.

derivative is defined everywhere except the optimal point. Away from the optimum the sign of the derivative does indicate the correct direction to move in order to find the optimum, however the magnitude of the derivative does not give correct information about how far to step. Also evident is the local maximum at $R = 0$.

Some other properties of this problem are that when the boundaries of the two circles do not intersect, the sensitivity of the cost function to the point (x_c, y_c) is zero. In the special case that the interiors are disjoint as well, we see that the estimated curve will simply decrease in radius until it disappears. The configuration where the estimate is of zero area is a local minimum. This last observation holds true in general, and care must be taken to avoid either starting the minimization with disjoint interiors, or allowing such a situation during the minimization process. This is particularly a concern since the gradients can become quite large, even very near the optimum.

In the case of a circular estimated curve, it is easy to prevent the interior of the initial guess from being disjoint with the interior of the measured curve. To do so it is sufficient to center the estimate on the origin, and then pick the initial radius sufficiently large. This may seem to be special to the example, but it is not. Consider the c -level set, $\{x \in R^2 : \Psi(x, t) = c\}$. If we use the signed distance function for Ψ , when c is sufficiently large, the measured curve must be contained in the interior of the estimate. So we replace $\Psi(x; \lambda)$ by $\tilde{\Psi}(x; \tilde{\lambda})$, where $\tilde{\Psi} = \Psi - c$, and $\tilde{\lambda} = (\lambda, c)$.

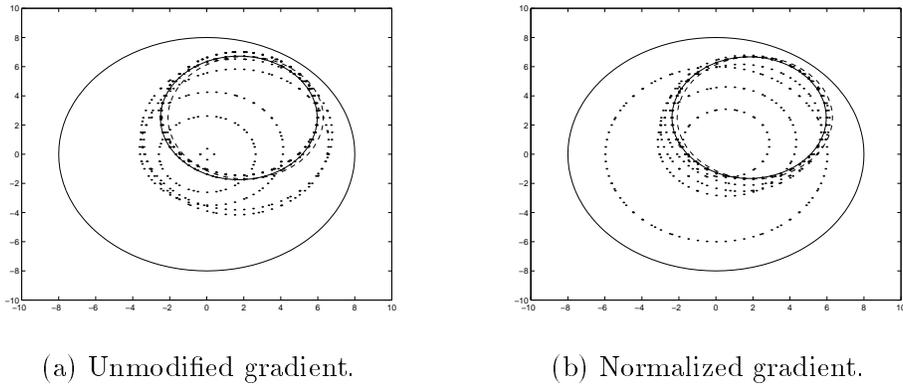


Figure 8: True curve (dashed) is circle of radius 4.2, centered on (2.1, 2.5). Initial estimate (solid) is circle of radius 8 centered on (0, 0). (a) Standard BFGS search converges to final estimate of $R = 4.2402$ and $(x_c, y_c) = (1.7335, 2.4881)$ (also solid) in 10 iterations (dotted). Final cost is $J = 2.1960$. (b) BFGS search with normalized gradient converges to final estimate of $R = 4.1735$ and $(x_c, y_c) = (1.7934, 2.5059)$ (also solid) in 8 iterations (dotted). Final cost is $J = 1.2041$.

5.4 Numerical Minimization Results

Test cases were run using the BFGS algorithm, as implemented in the *TOMLAB* package [9]. The value of Π was constrained to be greater than the gridsize h . All cases used the symmetric difference method to calculate the cost function, and the gridsize was 0.5 in the x and y directions. Because BFGS uses a quadratic approximation to the cost function, it is probably not the best choice for this problem. However, pending a more detailed analysis, it seemed to work well enough for the purposes of this preliminary study. One attempt at improving performance and robustness concerned how the gradient was handled. As noted above, the gradient varies widely in magnitude, and this variation is unrelated to the distance from the optimal point. On the other hand, the direction information contained in the gradient is very useful. Therefore some cases were run using the unmodified gradient, while in others the gradient was normalized to a unit vector. Performance is notably superior in the normalized cases, as shown by the decreased number of iterations required for convergence, and in the lower cost function. Results are shown in Figs. 8–10. Clearly there is room for improvement in implementation and accuracy, but the method seems to be performing as desired.

6 Parametrized Speed Functions

This work is motivated by applications to curve evolution problems. In these problems of interest, the λ dependence of $\Psi(x; \lambda)$ is not explicitly due to the parametrization, but arises

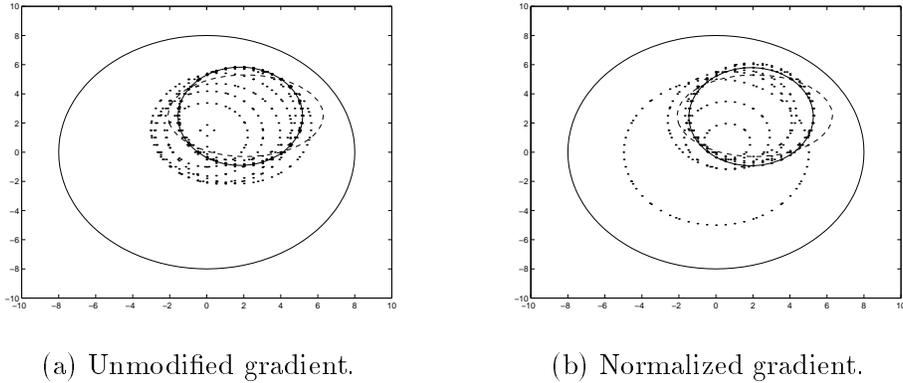


Figure 9: True curve (dashed) is ellipse $(x - 2.1)^2 + 2.25(y - 2.5)^2 = 4.2^2$. Initial estimate (solid) is circle of radius 8 centered on $(0, 0)$. (a) Standard BFGS search converges to final estimate of $R = 3.3729$ and $(x_c, y_c) = (1.8128, 2.4613)$ (also solid) in 14 iterations (dotted). Final cost is $J = 7.5502$. In this example the first step comes very close to being disjoint with the measured curve. This would have resulted in convergence to a local minimum far from the optimal point. (b) BFGS search with normalized gradient converges to final estimate of $R = 3.3686$ and $(x_c, y_c) = (1.9003, 2.4353)$ (also solid) in 9 iterations (dotted). Final cost is $J = 7.4152$. Here there was no danger of convergence to a local minimum.

from the variation of β with λ in the evolution equation

$$\Psi_t + \beta(\cdot; \lambda) \|\nabla \Psi\| = 0 \tag{34}$$

Thus we will need to determine Ψ_i from β_i instead of writing it directly. One way to accomplish this is by differentiating both sides of Eq. (34) with respect to the i -th parameter. The result is a PDE for the sensitivity, coupled to the original evolution equation. This type of approach is described further in [6]. For the level set evolution equation, the result of the differentiation is as follows:

$$\Psi_{t,i} + \beta_i \|\nabla \Psi\| + (\beta / \|\nabla \Psi\|) (\Psi_{x,i} + \Psi_{y,i}) = 0$$

Exchanging the order of differentiation gives,

$$\Psi_{i,t} + \beta_i \|\nabla \Psi\| + (\beta / \|\nabla \Psi\|) (\Psi_{i,x} + \Psi_{i,y}) = 0$$

Finally, denote the sensitivity by $S^{(i)} := \Psi_i$, and write,

$$S_t^{(i)} + \beta_i \|\nabla \Psi\| + (\beta / \|\nabla \Psi\|) (S_x^{(i)} + S_y^{(i)}) = 0 \tag{35}$$

Once Eq. (34) is solved, $\|\nabla \Psi\|$ can be computed. Since β is known, and β_i can be computed, Eq. (35) is a linear first order PDE. Initially the various $S^{(i)}$ are everywhere zero, since the

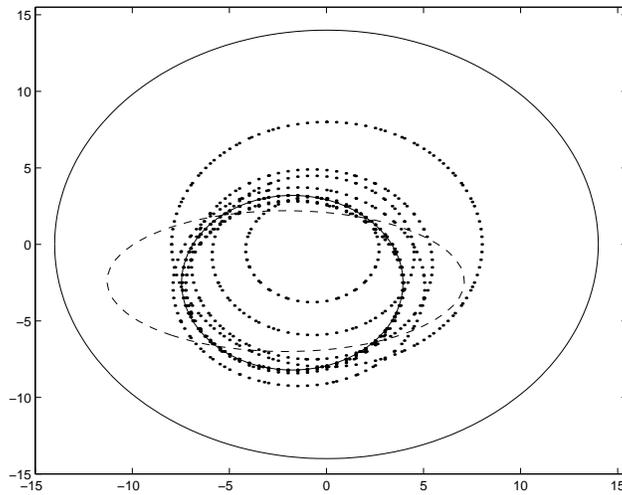


Figure 10: True curve (dashed) is ellipse $(x+2.1)^2+4.0(y+2.4)^2 = 9.2^2$. Initial estimate (solid) is circle of radius 14 centered on $(0,0)$. BFGS search with normalized gradient converges to final estimate of $R = 5.7057$ and $(x_c, y_c) = (-1.7587, -2.5040)$ (also solid) in 10 iterations (dotted). Final cost is $J = 53.267$.

starting curve for the model is exactly the true starting curve, and does not depend on the speed function β . Depending on storage requirements and the details of the solver implementation, Eq. (35) can be advanced at each time step in conjunction with Eq. (34), or it can be solved after the full solution of Eq. (34) is obtained.

7 Acknowledgements

This work has been partially supported by a grant from the Lam Research Corporation. The author would like to thank Prof. Allen Tannenbaum of the University of Minnesota and Dr. Anthony Yezzi of the Massachusetts Institute of Technology for their help on the theoretical and numerical analysis of curve evolution. Thanks also to Prof. Kenneth Holmström, of Mälardalen University, for making available his numerical optimization library *TOMLAB*. Any misuse of the *TOMLAB* routines is solely the responsibility of the author.

References

- [1] D. Adalsteinsson and J. A. Sethian, “A level set approach to a unified model for etching, deposition, and lithography I: algorithms and two-dimensional simulations,” *Journal of Computational Physics* **120**:128–144, 1995.
- [2] J. M. Berg, A. Yezzi, and A. R. Tannenbaum, “Phase transitions, curve evolution, and the control of semiconductor manufacturing processes” *Proc. 35th IEEE CDC*, pp. 3376–3381, Kobe, Japan, 1996.
- [3] J. M. Berg, A. Yezzi, and A. R. Tannenbaum, “Phase transitions, curve evolution, and the control of semiconductor manufacturing processes,” IMA Preprint 1454, Institute for Mathematics and Its Applications, University of Minnesota, 1997.
- [4] J. M. Berg, A. Yezzi, and A. R. Tannenbaum, “Toward Real-Time Estimation of Surface Evolution in Plasma Etching: Isotropy, Anisotropy, and Self-Calibration,” *Proceedings of the 36th IEEE CDC*, San Diego, CA, pp. 860–865, 1997.
- [5] J. M. Berg, A. Yezzi, and A. R. Tannenbaum, “Curve Evolution Models for Real-Time Identification with Application to Plasma Etching,” *IEEE Transactions Automatic Control*, to appear.
- [6] J. Borggaard and J. Burns, “A sensitivity equation approach to shape optimization in fluid,” in *Flow Control*, M.D. Gunzburger, ed., Springer-Verlag, New York, 1995.
- [7] T. S. Cale, M. B. Chaara, and A. Hasper, “Estimating Local Deposition Conditions and Kinetic Parameters Using Film Profiles,” *Mat. Res. Soc. Symp. Proc.*, Vol. 260, pp. 393–398, 1992.
- [8] M. P. do Carmo, *Differential Geometry of Curves and Surfaces*, Prentice-Hall, New Jersey, 1976.

- [9] K. Holmström, *NLPLIB TB 1.0—A MATLAB Toolbox for Nonlinear Optimization and Parameter Estimation*, Research Report in Mathematics/Applied Mathematics, Technical Report IMA-TOM-1997-2, Center of Mathematical Modeling, Department of Mathematics and Physics, Mälardalen University, Sweden, 1997.
- [10] B. B. Kimia, A. Tannenbaum, and S. W. Zucker, “On the evolution of curves via a function of curvature, I: the classical case,” *J. of Math. Analysis and Applications* **163**, pp. 438-458, 1992.
- [11] J. P. McVittie, J. C. Rey, A. J. Bariya, M. M. IslamRaja, L. Y. Cheng, S. Ravi, and K. C. Saraswat, “SPEEDIE: A Profile Simulator for Etching and Deposition,” *Proceedings SPIE Symposium: Advanced Techniques for Integrated Circuits Processing*, Vol. 1392, pp. 126-138, 1990.
- [12] W. Rudin, *Principles of Mathematical Analysis*, third edition, McGraw-Hill, Inc., New York, 1976.
- [13] F. Santosa, “A Level-Set Approach for Inverse Problems Involving Obstacles,” *ESAIM: Control, Optimisation, and Calculus of Variations*, Vol. 1, pp. 17–33, 1996.
- [14] K. Siddiqi, B. Kimia, C-W. Wang, “Geometric Shock-Capturing ENO Schemes for Subpixel interpolation, Computation, and Curve Evolution,” Technical Report LEMS-142, Brown University, February, 1995.
- [15] J. A. Sethian, *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, Cambridge University Press, 1996.
- [16] J. A. Sethian and D. Adalsteinsson, “An Overview of Level Set Methods for Etching, Deposition, and Lithography Development,” *IEEE Trans. Semiconduct. Manufacturing* **10**:1, pp. 167–184, 1997.
- [17] E. Zawaideh and N. S. Kim, “A plasma etching model based on a generalized transport approach,” *J. Appl. Phys.*, Vol. 62, No. 15, pp. 2498–2507, 1987.