A Variational Approach to Search and Path Planning Using Level Set Methods

Thomas Cecil[*]
ICES, UT Austin. Austin, TX 78712,
and
Daniel Marthaler[†]
Department of Mathematics, UCLA. Los Angeles, CA 90095

September 30, 2004

Abstract

In this paper we propose a variational approach to a path planning problem in 2 dimensions using a level set framework. After defining an energy integral over the path, we use gradient flow on the defined energy and evolve the entire path until a locally optimal steady state is reached. Unlike typical level set implementations where the interface being tracked is a codimension-1 set, we allow for paths with positive, varying widths. Applications of this method extend to robotic motion, tool-path milling, and arial search patterns for example. Numerical methods and algorithms are given, and examples are presented.

1 Introduction

The general problem of finding the optimal path through a domain under some given constraints has multiple applications. From path planning for autonomous vehicles [23], to computing tool path trajectories [8], to maximizing visibility [5], there exist wide applications for the general solutions to this problem. Given that the domain is not homogeneous, i.e. there is an associated cost function to the path in the domain, the general solution begins to increase rapidly in complexity.

A specific instance of the general problem is that of finding an optimal-path map for a known environment. The optimal-path map for a known two-dimensional terrain is a function $\omega(x, y)$ whose values describe how to best reach the goal from the location (x, y). Optimality in this case could be shortest path, least visibility from above, largest patrol area, etc. Previous work on true

[*] `tcecil@ices.utexas.edu`
[†] `daniel@math.ucla.edu`

optimal-path maps for autonomous robotics have almost always investigated restricted cases. While a thorough search is performed, it is usually over a finite set of locations. Searching over an infinite set of locations does not allow the possibility of reducing the problem to graph search.

Robotic Motion In [23], the problem of constructing optimal-path maps for two-dimensional polygonal weighted-region domains was investigated for piecewise-constant solutions. The complexity of the algorithm rendered it incompatible for real-time path generation, but when a large amount of planning time is available (daily routes of a sentry, etc.) it became desirable. In [15] path planning for robots was studied using level sets where there were objects to be avoided in the domain. The method of solution was to construct a weighted distance function over the entire domain and then, from a final position, back propogate the solution perpendicular to the level sets of the distance function, resulting in an optimally shortest path. Path planning algorithms for mobile robots are also described in [17],[2],[16]. Also, in the context of manipulators there has been path planning research done within a variational framework [24].

Tool Paths Other instances of the general problem arise in the generation of tool paths for pocket machining. Here, material is milled from a blank, layer by layer, until the object is machined into a manufactured part. The tool path for a layer of a pocket is the centerline path along which a tool cuts the material. Optimal for this problem involves minimal length and minimal curvature of the path (high curvature induces more wear and tear on the drill). In [1], an approximate solution to the problem of how to best produce tool paths for pocket machining was presented. There, the authors solved an elliptic partial differential equation (PDE) boundary value problem, and used the contours of that solution to construct a solution. In [8] an extensive, categorized, reference list is presented of papers related to numerical control tool path generation. Some of the areas of research are isoparametric paths, non-isoparametric paths, planar pocketing paths, roughing paths, and space-filling curve based tool paths.

Dynamic Visibility In [27] the framework for studying visibility and its dynamics using level sets was established. This included the use of implicit surfaces to represent visible regions and the introduction of PDEs that govern horizon and boundary curves. In [5] various variational problems were approached using the framework established in [27], including single and multiple point visibility optimization, the effects of weighted spatial regions, and the effects of memory. In [5] a parameterized path planning algorithm was introduced that treats the path as a finite union of multiple observers which are evolved so as to maximize the accumulated visibility along the path. See also [29] for a path planning algorithm based on visibility.

Related ideas in image processing having to do with active contours and snakes can be found [14],[3],[11]. These papers introduce curve evolutions resulting from variational frameworks used to segment images. Often these implementations involve the maximization of a quantity defined by the image, such as the norm of the gradient, over a curve, and they often involve regularizing terms resulting in mean curvature flow when in a level set setting.

Something the above examples have in common that it is not only the path that is optimized but also the set that the path carves out that must be considered. For the robot path planning algorithm the width of the set may be dependent upon the size of the robot. In the tool planning problem it can be the width of the drill bit. For searching we can consider either the region that the observer can see to be analogous to the size of the drill bit in the milling example. A reverse case would be if an eluder did not want to be seen, and then the size of the region where the eluder can be detected would be analogous to the drill bit size.

In this paper, we investigate the general problem of finding a "search path" through a domain where we know some information about where targets may be located. An agent searching such a domain would want to have a path that satisfies being shortest with having a high confidence of finding targets. The searcher can only "see" a finite distance about it at any given point, and this distance may vary spatially according to local weather conditions, altitude, etc. Therefore, we wish to generate an optimal path that gives a certain level of confidence of locating targets which will be determined via the information we know about the domain.

The remainder of the paper is organized as follows, in the next section, we formulate the search path problem in a general framework, with general metrics describing the optimization, and constraints stemming from the search path problem. Following that, we introduce the level set method, and then our algorithm. We present simulations of canonical examples demonstrating the method and conclude with some remarks about the generality of the method.

2 Problem Formulation

The general path planning problem has had many formulations. For a given set $\Omega \in \mathbb{R}^n$, we seek a path $\Gamma \in \mathbb{R}$, with the following properties:

1. Optimize some function of Γ (Arclength, Curvature, etc.).
2. Given an *a priori* distribution, P, on Ω, maximize

$$\int_{S_\Gamma} P(x)dx$$

where $S_\Gamma = \{x \in \Omega : |x - \Gamma| \leq c(x)\}$, where $c(x)$ is the radius of the set "cut-out" of the domain by the path Γ.

We note that this problem was motivated by the problem of computing optimal search strategies in the presence of *a priori* knowledge [18]. The function P represents any knowledge of the search domain, Ω. Possible choices for the optimization would include minimal arclength and minimal curvature. Also, we note that this encompasses obstacle avoidance when 2 is minimized or the sign of P is changed.

2.1 Level Set Formulation

The search path Γ will be represented by the 0 level set of a function $\phi : \Omega \to \mathbb{R}$. Given an initial function $\phi(t = 0)$, and an energy $E(\phi)$ to be minimized/maximized, we use the method of gradient descent/ascent to arrive at a PDE of the form

$$\frac{\partial \phi}{\partial t} = -/+ \ \frac{\partial E}{\partial \phi}, \tag{1}$$

where $\frac{\partial E}{\partial \phi}$ is taken from the Euler-Lagrange equation. This PDE is then evolved to steady state resulting in ϕ obtaining a local minimum. Most of our variational problems will be nonconvex, so the initial choice of ϕ will determine the local minimum in which we finish. The numerical methods for solving (1) will be discussed later.

2.2 Examples of Energies and PDEs

The energy representing

$$\int_{S_\Gamma} P(x) dx \tag{2}$$

will always be included in our variational formulation. First we assume ϕ is a weighted signed distance function. One way to construct ϕ is to solve

$$|\nabla \phi| = \frac{1}{R(x)}, \tag{3}$$

where $R(x) > 0$, with boundary condition given by $\{\phi(x) = 0 | x \in \Gamma\}$ [22], [19],[25],[28]. In terms of the search problem $R \approx 0$ implies that visibility is reduced to almost nothing, while when $R \to \infty$ visibility becomes large. Given this assumption, we can see that the set

$$\chi(S_\Gamma) = H(r - \phi) H(r + \phi), \tag{4}$$

where χ is the characteristic function, and H is the Heaviside function. Here r is a constant, and we assume that the information about the radius of visibility function $c(x)$ has been incorporated into the function $R(x)$.

Thus our integral (2) can be written as

$$\int_{S_\Gamma} P(x) dx = \int_\Omega H(r - \phi) H(r + \phi) \ P(x) \ dx, \tag{5}$$

where we have integrated over the entire domain Ω. To maximize this integral we perform gradient ascent and arrive at the PDE

$$\phi_t = P(x)[H(r - \phi)\delta(r + \phi) - H(r + \phi)\delta(r - \phi)], \tag{6}$$

where $\delta(y) = H'(y)$ denotes the Dirac delta function. Intuitively (6) attempts to move the $\pm r$ level sets away from the 0 level set where $P > 0$, so that $\int_{S_\Gamma} P$ becomes larger as time progresses.

Another common energy term to be minimized is the length of Γ. We can write

$$|\Gamma| = \int_\Omega \delta(\phi)|\nabla\phi| \; dx. \tag{7}$$

Gradient descent yields

$$\frac{\partial\phi}{\partial t} = \delta(\phi)\nabla \cdot \frac{\nabla\phi}{|\nabla\phi|} = \delta(\phi)\kappa, \tag{8}$$

where κ is the mean curvature of Γ.

For certain problems such as the tool path problem one may want to control the magnitude of κ along the path. Energies to be minimized in this case could be of the form

$$\int_\Omega \delta(\phi)g(\kappa) \; dx, \tag{9}$$

where g is a non-negative function of κ such as $|\kappa|^p, p > 0$. The PDEs resulting from (9) are fourth order involving second partial derivatives of κ.

In general we will evolve (6) with its right hand side augmented by adding weighted terms that are found from the other energy minimizations/maximizations that each particular problem demands. An example would be adding a term $\lambda\delta(\phi)\kappa$ to the right hand side of (6), where $\lambda > 0$ can be thought of as a Lagrange multiplier.

3 Numerical Methods

3.1 Advancement of Time Dependent PDEs

The PDEs found in section 2.2 are generally Hamilton-Jacobi equations. To discretize them we construct a uniform rectangular grid on Ω. Viscosity solutions for these types of equations have been studied well [6], [10] and numerical methods that converge to the viscosity solution have been implemented [7], [20]. We use these methods to solve our equations. In general they consist of upwind type spatial discretizations and explicit Runge-Kutta time discretizations. On $\partial\Omega$, which is the boundary of Ω, we use Neumann boundary conditions: $\partial\phi/\partial n = 0$.

Most level set variational problem formulations involve only one level set of interest, usually the set $\Gamma_0 = \{x|\phi(x) = 0\}$. Thus the PDEs to be evolved only involve δ functions that have support localized near Γ_0. However, for our problem, (6) includes δ functions whose support lies near the sets $\Gamma_{\pm r} = \{x|\phi(x) = \pm r\}$ when we use a numerical approximation to δ. When (6) is combined with a Lagrange multiplier term derived from (8), then we have a PDE with δ functions localized near the places where $\phi = -r, 0, r$. Therefore three separate level sets of ϕ must be evolved, while ϕ maintains the criterion that it is a weighted signed distance function satisfying (3) with Γ_0 as the boundary condition.

If we try to simultaneously evolve ϕ in the different regions near $\Gamma_0, \Gamma_{\pm r}$, then we see that (3) will not be satisfied. This cannot be quickly repaired by reinitialization of ϕ to a signed distance function because it is unclear which boundary conditions should be used (i.e. Γ_0, Γ_{-r}, or Γ_r) Interesting motion for our variational problem has occurs near all three level sets, and reinitialization with only one of the Γ_z as the boundary condition would disregard motion near the other two $\Gamma_{w \neq z}$.

To remedy this problem we propose to evolve ϕ near one of the Γ_z, $z = -r, 0, r$ for a short period of time, then if $z = \pm r$ we perform a pseudo-reinitialization that approximates the solution to (3) with Γ_z as the boundary condition (we will explain in a moment what we mean by 'pseudo-reinitialization' and why it is necessary), and if $z = 0$ we perform the usual reinitialization to a weighted signed distance function solving (3). Once one of the Γ_z has been advanced and ϕ has been reinitialized, we move to another Γ_z and repeat the process. So prior to each advancement of a particular Γ_z we are forcing ϕ to approximately or exactly satisfy (3).

In order to avoid the discretization of the δ function we modify the PDEs by replacing the $\delta(\phi)$ with $|\nabla\phi|$. This allows all level sets to move with the same speed. However, we would still like to maintain the local nature of the support for the δ function, so we also multiply $|\nabla\phi|$ by a cutoff function, $\rho(\phi)$, as described in [21] with support over points where $|\phi| < r$.

3.1.1 Pseudo-reinitialization

In the method described above, if we were to use true weighted signed distance reinitialization instead of the pseudo-reinitialization procedure, we would encounter problems where the reinitialization with boundary condition Γ_y would pick the viscosity solution to (3) and we would lose interesting parts of ϕ near $\Gamma_{w \neq y}$.

An example of this phenomenon is shown in figure 1. Note how on the right plot Γ_0 has not changed location, but Γ_r has been smoothed as the viscosity solution to (3) has been found, eliminating the part of Γ_r containing a cusp. In both plots Γ_0 is a level set of ϕ that satisfies (3) with boundary condition given by Γ_r (which is a condition that ϕ must satisfy), but one can see that the area between the two curves in much larger in the left plot, and this area could be an energy which we would like to maximize. Thus we would like to be able to maintain cusped areas of the Γ_z while also requiring that (3) holds with boundary condition given by Γ_0.

The method we use to achieve this is denoted pseudo-reinitialization. For notational purposes we will assume Γ_0 is the boundary condition for reinitialization, as using other Γ_z for the boundary only require shifts in the signum function, S. The idea is that instead of using the PDE method of reinitializa-

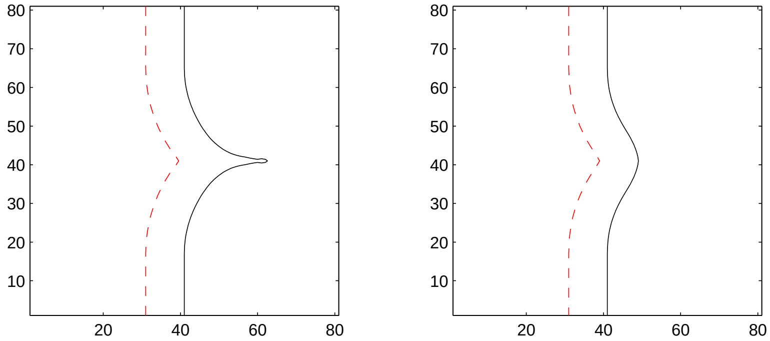

Figure 1: Left: Initial contour plot of ϕ. Dashed line: Γ_0. Solid line: Γ_r. Right: Contour plot of ϕ after true reinitialization with Γ_0 used as boundary condition, i.e. solution of (3).

tion to a weighted signed distance function [26] which solves the equation

$$\phi_t + S(\phi)\left(|\nabla\phi| - \frac{1}{R(x)}\right) = 0, \tag{10}$$

we instead solve

$$\phi_t + S(\phi)\left(\nabla\phi \cdot \frac{\eta}{|\eta|} - |\eta|\right) = 0, \tag{11}$$

where η is a static vector field found by taking $\eta = \nabla\phi$ prior to starting the pseudo-reinitialization.

The difference between (10) and (11) is that we predefine the characteristic flow directions, $\eta/|\eta|$, instead of letting them evolve during the reinitialization, and we also use $|\eta|$ instead of $1/R(x)$ to determine the growth of ϕ along the characteristics. As long as ϕ has not changed too much during an evolution step near Γ_0, then both $\eta/|\eta|$ and $|\eta|$ will be close to the quantities they represent in (10), which are $\nabla\phi/|\nabla\phi|$ and $1/R(x)$, respectively.

The idea behind (11) is that instead of allowing the characteristics to follow a rarefaction that would be found when taking the viscosity solution to (10), as in figure 2, we instead have them follow paths that they would take if reinitialization was being done with Γ_r as the boundary condition and we were finding the viscosity solution to (10) in that case. Figure 3 shows an example of the paths the characteristics might take if Γ_r had a protruding bump. These paths are quite different than those in the viscosity solution to 2, and could be thought of as being the characteristic paths of a nonphysical shock in the solution to (10).

To solve (11) we first choose η. This is done in a Godunov type upwind

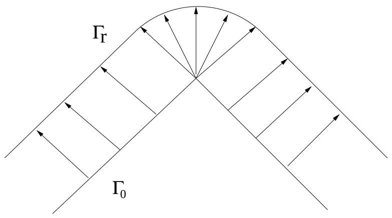

Figure 2: Characteristics of the viscosity solution to reinitialization.

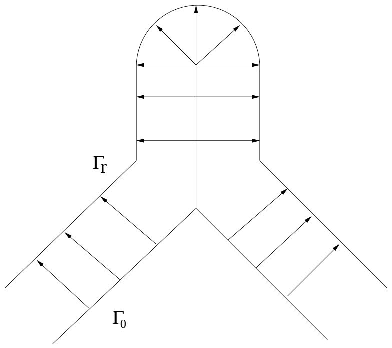

Figure 3: Characteristics in pseudo-reinitialization.

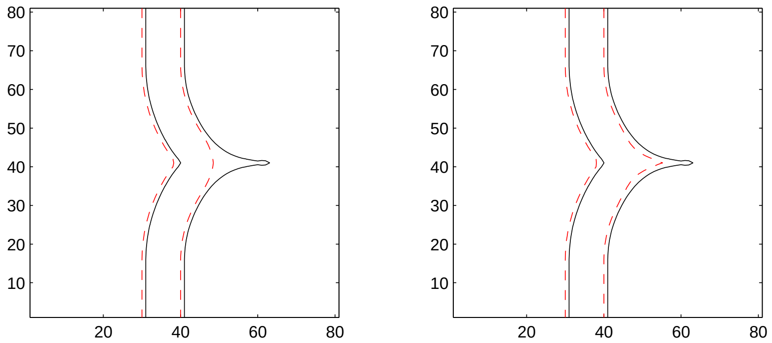

Figure 4: Left: Two contour plots of ϕ at different times. Solid lines are initial contours: Γ_0 (leftmost curve running from top to bottom), Γ_r (rightmost curve running from top to bottom). Dashed lines are contours of ϕ after Γ_0 has been evolved a short period of time (it has moved slightly to the left), and then true reinitialization has been performed with Γ_0 used as boundary condition. Right: Similar contour plots of ϕ, except that the dashed lines now have been calculated using pseudo-reinitialization with Γ_0 as boundary condition.

manner. For each gridpoint $x_{i,j}$ we make the following choices:

$$\begin{aligned} \eta_1 &= maxmod(\max(D_x^- \phi_{i,j}, 0), \min(D_x^+ \phi_{i,j}, 0))/dx, \\ \eta_2 &= maxmod(\max(D_y^- \phi_{i,j}, 0), \min(D_y^+ \phi_{i,j}, 0))/dy, \end{aligned} \tag{12}$$

if $S(\phi_{i,j}) \geq 0$, and

$$\begin{aligned} \eta_1 &= maxmod(\min(D_x^- \phi_{i,j}, 0), \max(D_x^+ \phi_{i,j}, 0))/dx, \\ \eta_2 &= maxmod(\min(D_y^- \phi_{i,j}, 0), \max(D_y^+ \phi_{i,j}, 0))/dy, \end{aligned} \tag{13}$$

if $S(\phi_{i,j}) < 0$, where

$$maxmod(x, y) = \begin{cases} x & \text{if } |x| \geq |y|, \\ y & otherwise. \end{cases}$$

Here $D_x^\pm \phi_{i,j} = \pm(\phi_{i\pm1,j} - \phi_{i,j})$, and $D_y^\pm \phi_{i,j} = \pm(\phi_{i,j\pm1} - \phi_{i,j})$. We note that more accurate W/ENO methods can also be used to construct η.

Once η has been chosen we can advance (11) using upwinding or W/ENO methods [20],[13] in space and TVD Runge-Kutta methods [20] in time as it is basically a linear advection equation with a forcing term.

Figure 4 shows an example of how pseudo-reinitialization retains cusped structures. One can imagine more drastic cases where the point of the cusped region expands to form a larger inlet region which would be removed during typical reinitialization.

3.2 Topology Preservation

For certain problems such as searching it makes physical sense that the search path Γ_0 enters Ω from one point $a \in \partial\Omega$, and leaves through one point $b \in \partial\Omega$ while never changing topology.

When using level set methods, one of the advantages over particle tracking methods is the ability to change topology without user intervention or adjustments to the algorithms being used. However, in our case we may wish to avoid these topological changes. If topology preservation is required for a particular problem it is implemented in the way outlined in [12].

This method of topology preservation in applicable only to uniform rectangular grids and relies on the well studied field of digital topology. Its implementation is local in nature and thus does not add significant complexity to the run time of the algorithm. It consists of a check of the signs of all the neighbors $x_{k,l}$ of a point $x_{i,j}$ with $\max(|k-i|, |l-j|) \leq 1$, along with their connectivities, after all $\phi_{i,j}$ have been advanced a timestep. If a topological change would occur in Γ_0 at $x_{i,j}$ if $\phi_{i,j}$ was allowed to change sign, then this sign change is prohibited. We refer the reader to [12] for more details on implementation and references concerning this technique.

We also note that in order to keep the points where Γ_0 intersects the boundary fixed, we imposed Dirichlet boundary conditions $\phi = 0$ there. If these points do not lie on the uniform grid then we can modify the grid slightly near them so that they are included in the discretization of Ω. If this is done then a local method for advancing the solution on an unstructured grid could be used near the points.

3.3 Outline of Evolution Procedure

In this section we outline the evolution procedure. We give a listing of the steps taken during one iteration. The evolution procedure is repeated until steady state is reached.

In certain steps of the computational procedure we shift ϕ by adding or subtracting r at all points so that equations such as (11) involving signum functions can be solved similarly no matter when they are called. This is explained in this way to emphasize that coding can be done using a smaller number of functions that do identical jobs on shifted versions of the data. When this is done we denote the shifted version of ϕ as $\phi \pm r$. It is assumed that after the step in question is completed ϕ is then shifted back the opposite way by $\mp r$.

The evolution loop advancing the solution from time t_1 to $t_1 + dt$ is given below. We illustrate the steps with an example PDE of the form:

$$\phi_t = P(x)[H(r-\phi)\delta(r+\phi) - H(r+\phi)\delta(r-\phi)] + \mu\kappa\delta(\phi), \tag{14}$$

using the substitutions of $\delta(\phi)$ functions by $|\nabla\phi|\rho(\phi)$, which is how they are implemented numerically.

1. Find η based on $\phi + r$ using (12), (13).

2. Evolve from time t_1 to $t_1 + dt$ the points near where $\phi = -r$, i.e. evolve all PDE terms with $\delta(r + \phi)$ in them, e.g. $\phi_t = P(x)\ H(r - \phi)|\nabla\phi|\rho(r + \phi)$.

3. Pseudo-reinitialize using (11) with $\phi + r$.

4. Find η based on $\phi - r$ using (12), (13).

5. Evolve from time t_1 to $t_1 + dt$ the points near where $\phi = r$, i.e. evolve all PDE terms with $\delta(r - \phi)$ in them, e.g. $\phi_t = P(x)\ H(r + \phi)|\nabla\phi|\rho(r - \phi)$.

6. Pseudo-reinitialize using (11) with $\phi - r$.

7. Evolve from time t_1 to $t_1 + dt$ the points near where $\phi = 0$, i.e. evolve all PDE terms with the $\delta(\phi)$ in them, e.g. $\phi_t = \mu\kappa|\nabla\phi|\rho(\phi)$.

8. Reinitialize to a weighted signed distance function from the 0 level set using (10).

9. Go to step 1.

4 Numerical Simulations

In this section we present some numerical simulations. The PDE we evolve to steady state is

$$\phi_t = P(x)[H(r - \phi)\delta(r + \phi) - H(r + \phi)\delta(r - \phi)] + \mu\kappa\delta(\phi), \tag{15}$$

using the methods mentioned above. The domain Ω is $[-1, 1]^2$ for all problems, discretized in a uniform rectangular grid with $dx = dy = 1/80$. As mentioned above Neumann BCs $\partial\phi/\partial n = 0$, are used. Topology preservation is used in all figures except for figure 12. After replacing the δ functions with $|\nabla\phi|\rho(\phi)$ a conservative estimate on the CFL condition for the problem is

$$dt\ \max\left\{\frac{|P|}{dx} + \frac{|P|}{dy}, \frac{2\mu}{dx^2} + \frac{2\mu}{dy^2}\right\} \leq 1, \tag{16}$$

where we use the max applied to the P and κ terms individually instead of to their sum because we are splitting the evolution procedure.

For the individual examples we do not explicitly write the initial conditions, but rather shown them in contour plots. The way they are constructed is by determining an initial contour $\Gamma_0(t = 0)$ and finding an arbitrary function that takes $\Gamma_0(t = 0)$ as its 0 level set, and then running reinitialization over the entire domain Ω with $\Gamma_0(t = 0)$ as the boundary condition.

For each example we show the initial and steady state solutions as well as a plot of the energy over time. This energy function, defined by taking (5) $-\mu$ times (7), is what we hope to maximize. The δ and Heaviside functions used are the compactly supported ones given in [4], with support parameter $\epsilon = 2dx$. It should be noted that a more accurate numerical construction of these singular

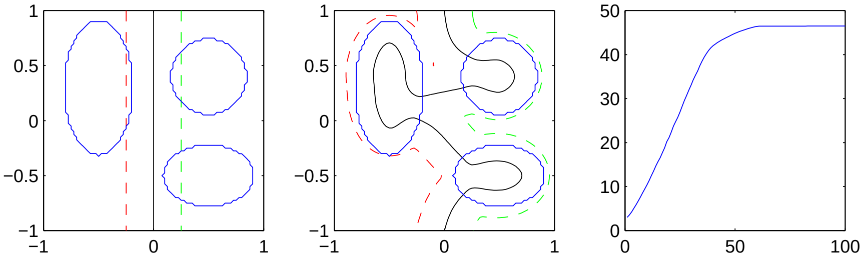

Figure 5: Initial, steady state and energy vs. timesteps plots. Constant visibility function $R(x)$.

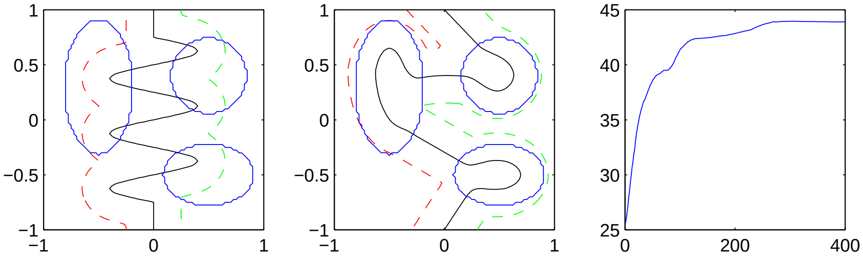

Figure 6: Initial, steady state and energy vs. timesteps plots. Constant visibility function $R(x)$.

functions can be found in [9] should a more exact measure of the energy be needed.

In all examples the path boundary level sets are $r = \pm 0.25$.

In figure 5 we show the initial, steady state and energy plots for an example with constant $R(x) = 1$. The circular objects have $P(x) = 40$, while $P(x) = 0$ otherwise. The regularization coefficient $\mu = 0.5$.

In figure 6 we show the initial, steady state and energy plots for an example with constant $R(x) = 1$, but this time we use a different initial condition. The circular objects have $P(x) = 40$, while $P(x) = 0$ otherwise. The regularization coefficient $\mu = 1$.

In figure 7 we show the initial, steady state and energy plots for an example with varying $R(x)$. In the lower half of the plane $R(x) = 0.6$, and in the upper half, $R(x) = 1$. The circular objects have $P(x) = 40$, while $P(x) = 0$ otherwise. The regularization coefficient $\mu = 0.5$.

In figure 8 we show the initial, steady state and energy plots for an example where $P(x)$ is nonzero in a larger region of interest. Within a circle of radius 0.9 we set $P(x) = 20 + 20\ G(x, y)$, where $G(x, y)$ is a compactly supported approximation to a δ function with radius of support 0.9, see figure 9. In figure 8 the boundary of the region with nonzero support is shown. In this example

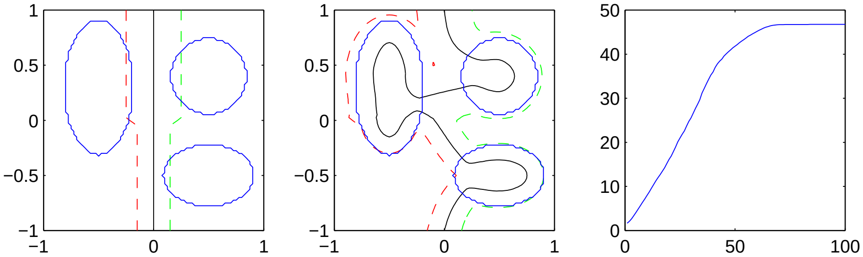

Figure 7: Initial, steady state and energy vs. timesteps plots. Varying visibility function $R(x)$.

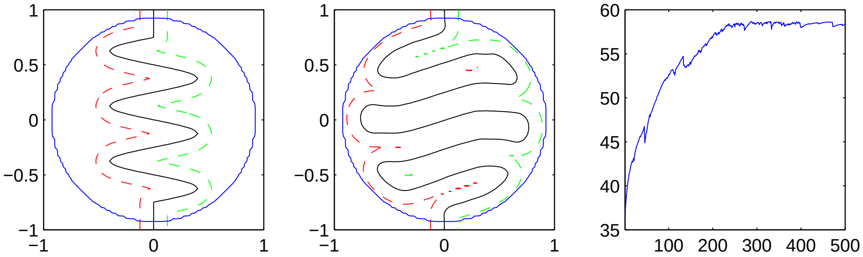

Figure 8: Initial, steady state and energy vs. timesteps plots. Gaussian type $P(x)$.

$R(x) = 0.5$ in the entire domain, and $\mu = 0.5$. In this example the steady state solution is the simple back and forth sweeping pattern.

In figure 10 we show the initial, steady state and energy plots for an example where object avoidance is the objective. Here $P(x) = -40$ in the circular objects and 0 elsewhere. In this example $R(x) = 0.5$ in the entire domain, and $\mu = 1$.

In figure 11 we show what can happen if the initial conditions are too close to a local minimum that is far form the global minimum. The problem parameters are identical to those of figure 10, but a different initial condition is imposed.

In figure 12 we show the initial, intermediate, steady state and energy plots for an example where topology change is allowed. Here $P(x) = 40$ in the circular objects and $P(x) = -40$ elsewhere. In this example $R(x) = 0.5$ in the entire domain, and $\mu = 0.02$.

Note how concentric ellipses have formed within each of the elliptical regions where $P > 0$.

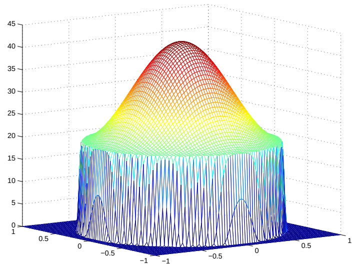

Figure 9: $P(x)$ used for figure 8.

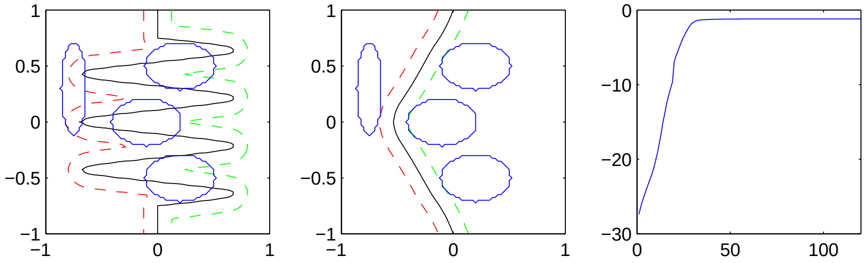

Figure 10: Initial, steady state and energy vs. timesteps plots for object avoidance. Solution converges to global minimum.

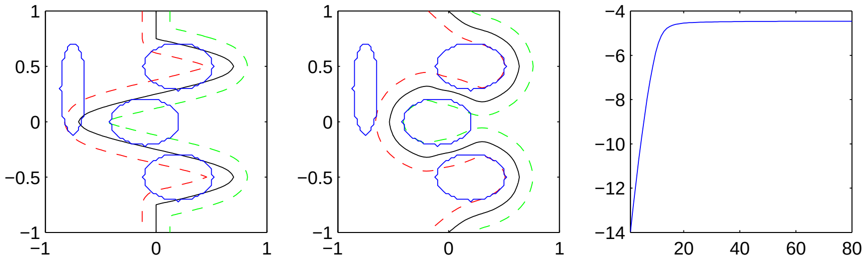

Figure 11: Initial, steady state and energy vs. timesteps plots for object avoidance. Solution converges to local minimum.

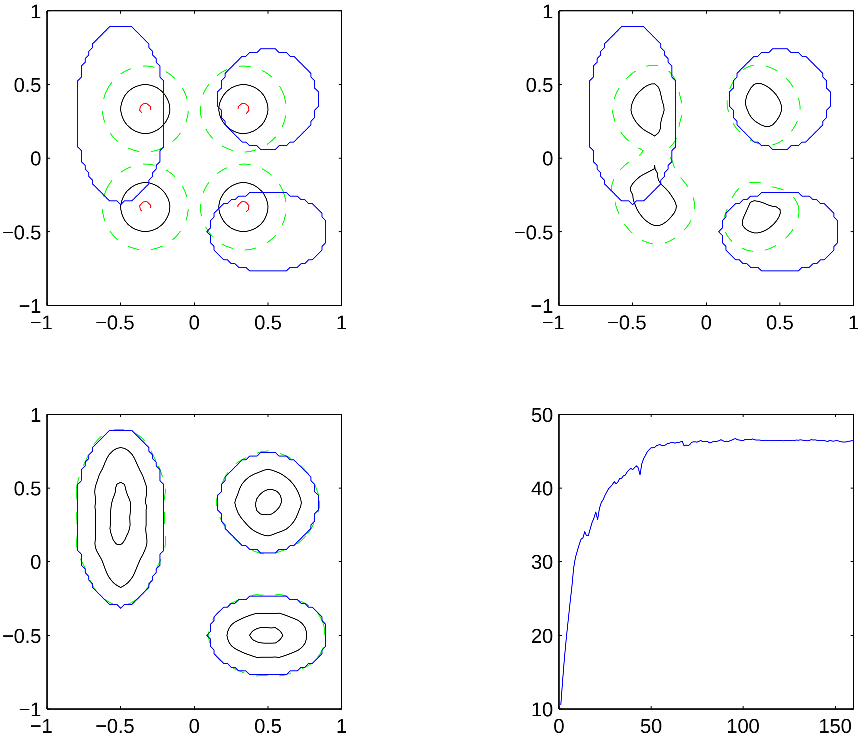

Figure 12: Top row: initial, and intermediate state plots. Bottom row: steady state, and energy vs. timesteps plots. Topology change is allowed in this example.

5 Conclusion

We have presented a level set based algorithm for solving a variational approach to path planning. Some key features of this algorithm are the energy integrals used to define the search criteria, the splitting technique used to advance the PDEs, and the pseudo-reinitialization. The number of timesteps needed to converge to steady state are on the order of hundreds, and for 2 dimensional problems the runtime can be made close to real time for applications.

The energy integrals used are very basic and encompass general properties that are desirable in many path planning problems. However, they are not exhaustive and more complicated energies based on functionals of curvature and other path properties can be constructed. Solving some of these types of variational formulations would be straightforward, but there are definite challenges ahead in this area.

An analytic proof of the numerical convergence of the splitting and pseudo-reinitialization techniques to a true maximization of the energy is also needed. While these techniques give quantitative results that indicate energy maximization, it would be useful to have a better understanding of their numerical properties.

Extensions to 3 dimensions would also be useful, but in 3d the path Γ_0 is a codimension-2 set and therefore either requires the use of 2 level set functions, or a new way of being tracked. Also, the splitting scheme in 3d may become more complicated.

Some other problems which we have not approached but are feasible for future research are: multiple non-intersecting paths, time dependent parameters such as R, P, μ, paths passing through multiple prescribed points, and self-intersecting paths. Level set motion with interfaces having positive width may also have applications in image processing, such as segmentation of thin objects, and also in areas of physics where the dynamics may demand that an interfacial width take nonzero measure.

6 Acknowledgments

The first author was supported by grants: NSF DMS-0312222, NSF ACI-0321917, and an ONR MURI grant subcontracted from Stanford University. The second author was supported by grants: ONR grant N000140410054, and ARO grant DAAD19-02-1-0055.

References

[1] M. Bieterman and D. Sandstrom. A curvilinear tool-path method for pocket machining. *Journal of Manufacturing Science and Engineering*, 125:709–715, 2003.

[2] John F. Canny. *The complexity of robot motion planning.* MIT Press, 1988.

[3] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *Int. J. of Comp. Vision*, 22, 1997.

[4] Tony Chan and Luminita Vese. Active contours without edges. *IEEE Trans. on Imag. Proc.*, 10(2), 2001.

[5] L.-T. Cheng and R. Tsai. A level set framework for visibility related variational problems. *UCLA CAM Report, Submitted to J. Comp. Phys.*, (04-03), 2003.

[6] M. G. Crandall, L. C. Evans, and P.-L. Lions. Some properties of viscosity solutions of Hamilton-Jacobi equations. *Trans. Amer. Math. Soc.*, 282(2):487–502, 1984.

[7] M. G. Crandall and P.-L. Lions. Two approximations of solutions of Hamilton-Jacobi equations. *Math. Comp.*, 43(167):1–19, 1984.

[8] D. Dragomatz and S. Mann. A classified bibliography of literature on NC milling path generation. *Comp.-Aided Design*, 29(3):239–247, 1997.

[9] Bjorn Engquist, Anna-Karin Tornberg, and Richard Tsai. Discretization of dirac delta functions in level set methods. *UCLA CAM Report*, (04-16), 2004.

[10] Lawrence C. Evans. *Partial differential equations*, volume 19 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 1998.

[11] P. Fua and Y.G. Leclerc. Model driven edge detection. *Mach. Vision. and Appl.*, 3(3):45–56, 1990.

[12] X. Han, C. Xu, and J. Prince. A topology preserving deformable model using level sets. *CVPR2001*, December 2001.

[13] Guang-Shan Jiang and Danping Peng. Weighted ENO schemes for Hamilton-Jacobi equations. *SIAM J. Sci. Comput.*, 21(6):2126–2143 (electronic), 2000.

[14] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Int. J. of Comp. Vision*, 1:321–331, 1988.

[15] Ron Kimmel and James A. Sethian. Optimal algorithm for shape from shading and path planning. *J. Math. Imaging Vision*, 14(3):237–244, 2001. Mathematics and image analysis 2000 (Paris).

[16] J.-C. Latombe. *Robot motion planning.* Kluwer Academic Publishers, 1991.

[17] J.-P. Laumond, editor. *Robot motion planning and control*, volume 229 of *Lecture Notes in Control and Information Sciences.* Springer-Verlag London Ltd., London, 1998.

[18] D. Marthaler, A. Bertozzi, and I. Schwartz. Levy searches based on *a priori* information: The biased Levy walk. *UCLA CAM Report*, (04-50), 2004.

[19] Stanley Osher and Ronald P. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Oxford Univ. Press, 2002.

[20] Stanley Osher and Chi-Wang Shu. High-order essentially nonoscillatory schemes for Hamilton-Jacobi equations. *SIAM J. Numer. Anal.*, 28(4):907–922, 1991.

[21] Danping Peng, Barry Merriman, Stanley Osher, Hongkai Zhao, and Myungjoo Kang. A PDE-based fast local level set method. *J. Comput. Phys.*, 155(2):410–438, 1999.

[22] Elisabeth Rouy and Agnès Tourin. A viscosity solutions approach to shape-from-shading. *SIAM J. Numer. Anal.*, 29(3):867–884, 1992.

[23] N. Rowe and R. Alexander. Finding optimal-path maps for path planning across weighted regions. *International Journal of Robotics Research*, 19(2):83–95, 2000.

[24] S. Sen, B. Dasgupta, and A.K. Mallik. Variational approach for singularity-free path-planning of parallel manipulators. *Mechanism and Mach. Theory*, 38:1165–1183, 2003.

[25] J. A. Sethian. Fast marching methods. *SIAM Rev.*, 41(2):199–235 (electronic), 1999.

[26] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *J. Comput. Phys.*, 114:146–159, 1994.

[27] Y.-H. R. Tsai, L.-T. Cheng, S. Osher, P. Burchard, and G. Sapiro. Visibility and its dynamics in an implicit framework. *J. Comput. Phys.*, 199:260–290, 2004.

[28] Yen-Hsi Richard Tsai, Li-Tien Cheng, Stanley Osher, and Hong-Kai Zhao. Fast sweeping algorithms for a class of Hamilton-Jacobi equations. *SIAM J. Numer. Anal.*, 41(2):673–694 (electronic), 2003.

[29] P. K. C. Wang. Optimal path planning based on visibility. *J. Optim. Theory Appl.*, 117(1):157–181, 2003.