A Variational Approach to Path Planning in Three Dimensions Using Level Set Methods

Thomas Cecil*
ICES, UT Austin. Austin, TX 78712,
and
Daniel E. Marthaler†
ACS-UMS, Northrop Grumman Corp, Rancho Bernardo, CA

December 8, 2004

Abstract

In this paper we extend the two dimensional methods set forth in [4], proposing a variational approach to a path planning problem in three 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. We follow the framework for motion of curves in three dimensions set forth in [2], modified appropriately to take into account that we allow for paths with positive, varying widths. Applications of this method extend to robotic motion and visibility problems, for example. Numerical methods and algorithms are given, and examples are presented.

1 Introduction

This paper extends results seen in [4] to path planning in three dimensions. Path planning in an obstacle-ridden environment while simultaneously attempting to search is an inherently difficult, and well-studied problem. In particular, in the field of unmanned aerial vehicle path planning, many different solution techniques have been studied. Potential field path planning methods have appeared frequently in the literature [22], but are plagued with inherent limitations [17]. Probabilistic road mapping [15] is a technique which uses a heuristic method to generate a road map through the space, then searches to find the lowest cost path. The heuristic nature of the path generation leads to a difficulty in characterizing the algorithms in terms of performance, robustness, complexity and reliability [14]. An optimization-based technique using a mixed integer linear

*tcecil@ices.utexas.edu
†daniel.marthaler@ngc.com

programming (MILP) method has recently been shown to perform quite well in specific instances [26],[10]. This method combines linear programming with the ability of constraining some subset of the state variables to be integers.

In general, path planning methods tend to fall in a spectrum with complex direct optimization approaches on one side and more pragmatic heuristic approaches on the other. Most approaches fall somewhere in between the two extremes and make use of both heuristic and optimization based components. Combining these components can lead to some guarantees on performance and robustness with reduced complexity and computation time. Also, and perhaps most relevant, is the fact that rarely, if ever is the globally optimal path ever required. In practice, most applications require a process that produces a reasonable result with the cost of the solution increasing with longer solve time.

The general problem of finding the optimal path through a domain under some given constraints has many applications. 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 three-dimensional terrain is a function $\omega(x, y, z)$ whose values describe how to best reach a goal point from the location (x, y, z). Optimality in this case could be shortest path, least visibility from above, largest patrol area, etc. Previous work on true optimal-path maps for autonomous robotics have almost always investigated restricted cases.

Another field of relevance is that of robotic motion. In [16] 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 propagate 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 [19],[3],[18]. Also, in the context of manipulators there has been path planning research done within a variational framework [28].

In [31] the framework for studying visibility and its dynamics using level sets was established. In [6] various variational problems were approached using the framework established in [31]. In [6] 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 [33] for a path planning algorithm based on visibility.

In this paper, we investigate the general problem of finding a "search path" through a domain where we know some information about where targets and obstacles 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 while avoiding obstacles. 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, while simultaneously avoiding obstacles, which will be determined via the information

we know about the domain.

Computationally we represent the path, Γ, as in intersection of the sets $\{x|\phi(x) = 0\} \cap \{x|\psi(x) = 0\}$ of two level set functions $\phi, \psi : \mathbb{R}^3 \to \mathbb{R}$. This framework was originally established in [2] and [7]. However, the extension of their method to our case requires that we evolve a tube, not just a codimension-2 curve, whose axis is Γ and whose boundary represents the edge of the "visible" region of the observer located on Γ.

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 $\Omega \in \mathbb{R}^3$, we seek a path $\Gamma : [0, 1] \to \mathbb{R}^3$, 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 our work on this problem was motivated by the task of computing optimal search strategies in the presence of *a priori* knowledge [20]. 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 intersection of the 0 level sets of two functions $\phi, \psi : \Omega \to \mathbb{R}$. This follows the framework established in [2] and [7]. Given initial functions $\phi(t = 0), \psi(t = 0)$, and an energy $E(\phi, \psi)$ to be minimized/maximized, we use the method of gradient descent/ascent to arrive at a set of coupled PDEs of the form

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

where $\frac{\partial E}{\partial \phi}, \frac{\partial E}{\partial \psi}$ are taken from the Euler-Lagrange equations. These PDEs are then evolved to steady state resulting in ϕ, ψ obtaining local minima. Most of our variational problems will be nonconvex, so the initial choices of ϕ, ψ will determine the local minima 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, ψ are weighted signed distance functions, and that they are perpendicular, i.e. $\nabla\phi \cdot \nabla\psi = 0$. Given this assumption, we can see that

$$\chi(S_\Gamma) = H(r - \|(\phi, \psi)\|), \tag{3}$$

where χ is the characteristic function, and H is the Heaviside function, and in practice we use $\|\phi, \psi\| \equiv \sqrt{\phi^2 + \psi^2}$. Here, the width of the tube r is a constant. We note that this results in a cylindrical path whose cross sections are circles of radius r, whose center axis is Γ. See Figure 1 for a sample tube construction. If other norms were used instead of l_2 then we would have tubes whose cross sections would be other objects, such as squares when using the l_1 norm.

Our integral (2) can then be written as

$$\int_{S_\Gamma} P(x)dx = \int_\Omega H(r - \sqrt{\phi^2 + \psi^2})\ P(x)\ dx, \tag{4}$$

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

$$\phi_t = P(x)\delta(r - \sqrt{\phi^2 + \psi^2})\frac{-\phi}{\sqrt{\phi^2 + \psi^2}}, \tag{5a}$$

$$\psi_t = P(x)\delta(r - \sqrt{\phi^2 + \psi^2})\frac{-\psi}{\sqrt{\phi^2 + \psi^2}}, \tag{5b}$$

where $\delta(y) = H'(y)$ denotes the Dirac delta function. Intuitively, solving (5) attempts to move the set $\{x|\sqrt{\phi^2 + \psi^2}\} = r$ away from the 0 level set where $P > 0$, so that $\int_{S_\Gamma} P$ becomes larger as time progresses.

Note that we have made the assumptions that ϕ, ψ are weighted signed distance functions, and that $\nabla\phi \cdot \nabla\psi = 0$. It is necessary for these assumptions to hold for (4) to be valid. Therefore, we need to enforce these conditions during the PDE evolutions. The way we do this is by periodically solving a set of PDEs whose steady state solutions satisfy the necessary criteria.

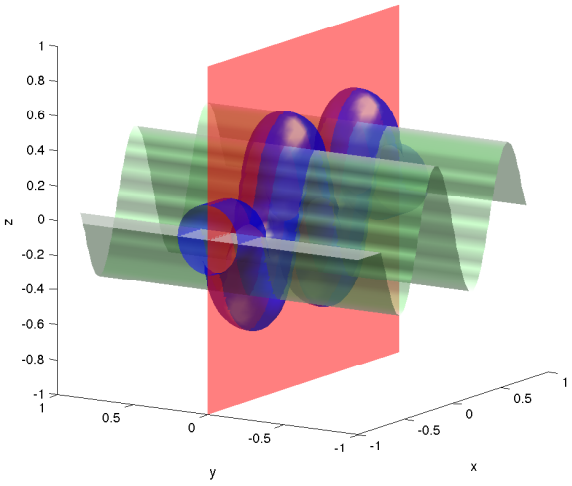

Figure 1: The graphs of $\{\phi = 0\}$ (red xz plane), $\{\psi = 0\}$ (green sinusoidal plane), and $\{\|(\phi, \psi)\|_2 = 0.2\}$ (blue tube).

When we say that ϕ is a signed distance function, we mean that ϕ satisfies

$$|\nabla \phi| = \frac{1}{R(x)}, \tag{6}$$

where $R(x) > 0$, with boundary condition given by $\{\phi(x) = 0 | x \in \Gamma\}$, and that $sign(\phi)$ is specified in Ω. One way of solving this is to solve the PDE

$$\phi_t + S(\phi)\left(|P_{\nabla \psi} \nabla \phi| - \frac{1}{R(x)}\right) = 0, \tag{7}$$

to steady state, where $S(x)$ is a regularized signum function, $\phi(x, t = 0) = \phi_0$ in Ω, and $\phi(x \in \Gamma) = 0$. The solution, ϕ_∞, is a signed distance function with distance measured on the sets $\{\psi = c\}, \forall c \in \mathbb{R}$. Similarly, to enforce that the level sets of ϕ are perpendicular to the 0 level set of ψ we can solve the following PDE to steady state

$$\phi_t + S(\psi)\frac{\nabla \psi}{|\nabla \psi|} \cdot \nabla \phi = 0, \tag{8}$$

with $\phi(x, t = 0) = \phi_0$ in Ω, and $\phi(t \geq 0)|_{\psi=0} = \phi(t = 0)|_{\psi=0}$, thus yielding

$$\frac{\nabla \psi}{|\nabla \psi|} \cdot \nabla \phi = 0, \tag{9}$$

We will discuss numerical solvers for these problems later.

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

$$|\Gamma| = \int_\Omega \delta(\phi)\delta(\psi)|P_{\nabla\psi}\nabla\phi||\nabla\psi| \; dx, \tag{10}$$

where

$$P_v = I - \frac{v \otimes v}{|v|^2} \tag{11}$$

is the orthogonal projection matrix projecting onto the plane with normal v. In standard codimension-1 level set methods the mean curvature motion

$$u_t = |\nabla u|\kappa \tag{12}$$

can be derived by minimizing an integral analogous to (10) by gradient descent, resulting in

$$u_t = \delta(u)\kappa, \tag{13}$$

and then replacing the δ function by $|\nabla u|$ (here κ is the signed mean curvature of the codimension-1 surface). In our framework it was shown in [2] that when minimizing (10) one ends up with a diagonal matrix of δ functions that can be replaced by a matrix serving the analogous role of $|\nabla u|$ in the codimension-1 case, which will result in the PDEs

$$\phi_t + kN \cdot \nabla\phi = 0 \tag{14a}$$
$$\psi_t + kN \cdot \nabla\psi = 0, \tag{14b}$$

where kN is curvature times the normal vector of Γ, i.e. after applying gradient descent we arrive at

$$\begin{pmatrix} \phi_t \\ \psi_t \end{pmatrix} = \begin{pmatrix} \delta(\phi)\delta(\psi) & 0 \\ 0 & \delta(\phi)\delta(\psi) \end{pmatrix} \begin{pmatrix} -\nabla \cdot \left(\frac{P_{\nabla\psi}\nabla\phi}{|P_{\nabla\psi}\nabla\phi|}|\nabla\psi| \right) \\ -\nabla \cdot \left(\frac{P_{\nabla\phi}\nabla\psi}{|P_{\nabla\phi}\nabla\psi|}|\nabla\phi| \right) \end{pmatrix}.$$

Then the matrix of δ functions is replaced by

$$\begin{pmatrix} \frac{|\nabla\phi|}{|P_{\nabla\phi}\nabla\psi|} & \frac{\nabla\phi\cdot\nabla\psi}{|P_{\nabla\psi}\nabla\phi||\nabla\psi|} \\ \frac{\nabla\phi\cdot\nabla\psi}{|P_{\nabla\phi}\nabla\psi||\nabla\phi|} & \frac{|\nabla\psi|}{|P_{\nabla\psi}\nabla\phi|} \end{pmatrix},$$

which is a symmetric positive definite matrix, indicating that we are still following a gradient descent direction minimizing (10).

We note that the vector kN can be found by taking the tangent vector

$$T = \frac{\nabla\psi \times \nabla\phi}{|\nabla\psi \times \nabla\phi|},$$

and deriving

$$kN = \frac{dT}{ds} = \nabla T \cdot T = \begin{pmatrix} \nabla T_1 \cdot T \\ \nabla T_2 \cdot T \\ \nabla T_3 \cdot T \end{pmatrix}, \tag{15}$$

where T_i is the i^{th} component of T and s is an parameterization of Γ, see [1],[2] for more details.

For certain problems one may want to control the magnitude of k along the path. Energies to be minimized in this case could be of the form

$$\int_\Omega \delta(\phi) g(k)\ dx, \tag{16}$$

where g is a non-negative function of k such as $|k|^p, p > 0$. The PDEs resulting from (16) are fourth order involving second partial derivatives of k.

In general we will evolve (5) with its right hand side augmented by adding weighted terms that are found from the additional energy minimizations/maximizations that each particular problem demands.

3 Numerical Methods

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 [9], [12], and numerical methods that converge to the viscosity solution have been implemented [8], [27], [24]. We use these methods to solve our equations. In general they consist of upwind type spatial discretizations and explicit Runge-Kutta time discretizations, and also fast sweeping solutions to find steady state solutions. We note that the notation below will refer to ϕ as the unknown function where ψ is fixed, but all algorithms and PDEs are also applied to ψ with ϕ fixed.

3.1 Advancement of Time Dependent PDEs Resulting from Gradient Flows

The level set problem formulations found in [2],[7] involve only one level set of interest, the set $\Gamma_0 = \{\phi(x) = 0\} \cap \{\psi(x) = 0\}$. Thus the PDEs to be evolved only involve δ functions that have support localized near Γ_0. However, for our problem, (5) includes a δ functions whose support lies near the set $S_\Gamma = \{x | \|(\phi, \psi)\|_2 = r\}$, when we use a numerical approximation to δ. When (5) is combined with a Lagrange multiplier term derived from minimizing $\lambda|\Gamma_0|$, then we have a PDE system with δ functions localized near Γ_0 also.

The goal is simultaneously evolve all the PDEs derived from gradient flow on the energy integrals, as well as from the distance function and perpendicularity requirements. We have chosen to use a splitting technique similar to that in [4]. The general idea introduced for a 2d domain there was to evolve

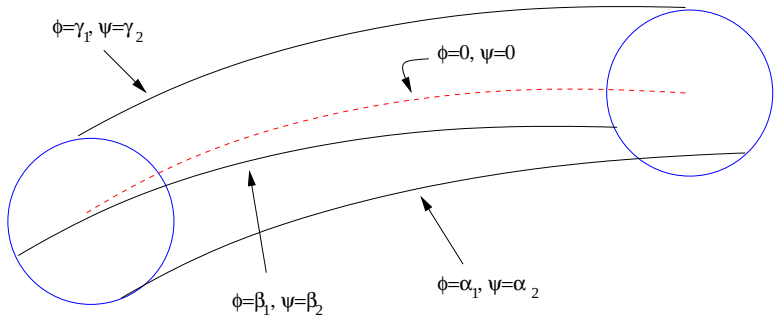

Figure 2: Tube S_Γ around Γ_0 (dotted line), with three curves, $\Gamma_{(\alpha_1,\alpha_2)}, \Gamma_{(\beta_1,\beta_2)}, \Gamma_{(\gamma_1,\gamma_2)}$, that lie on S_Γ, i.e. $\|(\alpha_1,\alpha_2)\|_2 = \|(\beta_1,\beta_2)\|_2 = \|(\gamma_1,\gamma_2)\|_2 = r$.

ϕ near the $-r, 0, r$ level sets separately, while enforcing (or pseudo-enforcing) the distance function requirement in between advancements. In 3d we have not only the added burden of enforcing perpendicularity, but also an infinite number of codimension-2 curves, $\Gamma_{(a,b)} \equiv \{\phi(x) = a\} \cap \{\psi(x) = b\}$, where $\|(a,b)\|_2 = r$, of which the tube S_Γ is composed. Our approach numerically will be to approximate these infinite number of curves by a finite number. Also, the perpendicularity requirement will be met by also solving (9), along with (6), in between advancements near the curves Γ_0 or $\Gamma_{(a,b)}$. See Figure 2.

Thus if we are using the l_2 norm to determine S_Γ, then we discretize a circular cross section of S_Γ (found by taking $T(y) \cdot (x - y)$ for a point $y \in \Gamma_0$) into M points by

$$\{(a_i, b_i)\}_{i=1}^M = \{cos(\theta_0 + 2\pi i/M), sin(\theta_0 + 2\pi i/M)\}_{i=1}^M, \tag{17}$$

where θ_0 can be chosen arbitrarily. Then we split the evolution of Γ_r into M separate evolutions near the curves $\{\phi = a_i\} \cap \{\psi = b_i\}$. After ϕ has been advanced for one timestep using an explicit Runge-Kutta method near $\Gamma_{(a_i,b_i)}$, we perform pseudo-reinitialization (as was done in the 2d case). This pseudo-reinitialization propagates the information from $\Gamma_{(a_i,b_i)}$ to the rest of Ω. We repeat the advancement and pseudo-reinitialization for ψ near $\Gamma_{(a_i,b_i)}$ as well before advancing near another $\Gamma_{(a_j,b_j)}$ for $j \neq i$. In practice we try to choose the (a_j, b_j) so that $(a_i, b_i) \cdot (a_j, b_j)$ is minimized.

In order to avoid the discretization of the δ function we modify the PDE (5a), for example, by replacing the $\delta(r - \|(\phi,\psi)\|)$ with $|\nabla\phi| C_r(r - \|(\phi,\psi)\|)$, where $C_r(x)$ is a cutoff function, as described in [25] with support over points where $|x| < r$. This allows all level sets of ϕ to move with the same speed, and still maintains the local nature (near $\Gamma_{(a_i,b_i)}$) of the evolution. The term $|\nabla\phi|$ is discretized using a Godunov numerical Hamiltonian.

Near Γ_0 we use the same technique except that we note that in (14) the δ functions have already been substituted out, and instead of using pseudo-

reinitialization after each step, we use true reinitialization. To discretize kN we use the formulation (15) with central finite differences used to approximate all derivatives.

3.2 Fast Sweeping Reinitialization

Because of the frequency with which we solve (7) and (9) to steady state, it is imperative that we have a fast solution method. Fortunately, there has been progress made in this area recently. Solution methods known as fast sweeping [36],[27], [32], and fast marching [29] take advantage of the hyperbolic nature of the problems to sweep or march along characteristics to derive the steady state solution with a minimal number of operations per gridpoint. For reinitialization on manifolds, however, the straightforward extension of these fast methods to the projected eikonal equation

$$|P_{\nabla\psi}\nabla\phi| = \frac{1}{R(x)} \tag{18}$$

does not work, as (18) does not satisfy the requirements for the currently designed sweeping and marching methods. This was noted in [21] where a fast method was proposed for solving (18) by instead solving (6) in a small band around the set $\{\psi = 0\}$. As the width of this band goes to 0, the solution converges to that of (18).

The method we use is to use the banded domain method of [21] to define the locations where ϕ will be evolved, and then employ the fast sweeping technique to solve (6). Some modifications are done to the methods to fit in our framework. Firstly, to avoid the overhead of initializing the band (and to keep the framework of the fast sweeping algorithm fixed), we instead solve the problem

$$|\nabla\phi| = \frac{1}{R(x)(C_h(\psi(x)) + \epsilon)}, \tag{19}$$

where again C_h is a piecewise constant cutoff function, taking only values $\{0, 1\}$, with support width h, and $0 < \epsilon \ll dx$. Also, to obtain a signed distance function as our solution we store the sign of ϕ at each point prior to starting the evolution, and then let $\phi = |\phi|$ before starting the sweeping procedure (note that the sweeping procedure then requires an initialization of every point away from a fixed band where $|\phi| \leq \delta = 1.5dx$ to a large positive value). After the sweeping iterations have ended we correct each ϕ by multiplying it by its original sign.

We note that in practice, because of the frequent perpendicularizations and reinitializations, the local coordinate system of ϕ, ψ inside of each cross section of Γ_r resembles that of two perpendicular axes, thus the projected Hamiltonian

$$|P_{\nabla\psi}\nabla\phi| = \sqrt{|\nabla\phi|^2 - |\nabla\phi \cdot \nabla\psi|^2} \approx |\nabla\phi|, \tag{20}$$

as $\nabla\phi \cdot \nabla\psi \approx 0$. Thus reasonable solutions can be obtained with large band widths, h.

The exact formulas used in the Gauss-Seidel sweeping step for 2d are listed in [36], and a procedure for deriving them is higher dimensions is given in [35]. For completeness we list them for 3d here.

Given a uniform grid with spacing dx, with function values indexed by $u_{i,j,k}$, we discretize (19) (with the right hand side forcing term replaced by $f_{i,j,k}$)

$$[(u^n_{i,j,k} - u^n_{xmin})^+]^2 + [(u^n_{i,j,k} - u^n_{ymin})^+]^2 + [(u^n_{i,j,k} - u^n_{zmin})^+]^2 = f^2_{i,j,k} dx^2, \tag{21}$$

where

$$\begin{aligned} u^n_{xmin} &= \min(u^n_{i-1,j,k}, u^n_{i+1,j,k}), \\ u^n_{ymin} &= \min(u^n_{i,j-1,k}, u^n_{i,j+1,k}), \\ u^n_{zmin} &= \min(u^n_{i,j,k-1}, u^n_{i,j,k+1}), \end{aligned}$$

and $(x)^+ = \max(x, 0)$. At $\partial\Omega$ we use one sided differences consisting only of points lying within Ω.

Assuming we have ordered the a_i from smallest to largest by $a_1 \le a_2 \le a_3$, then the solution to

$$[(u - a_1)^+]^2 + [(u - a_2)^+]^2 + [(x - a_3)^+]^2 = f^2 dx^2,$$

which is the same as (21), is given by the following formulas:

$$u = \begin{cases} \mu_1 \equiv a_1 + f dx & \text{if } \mu_1 \le a_2, \\ \mu_2 \equiv \frac{a_1 + a_2 + [2(fdx)^2 - (a_2 - a_1)^2]^{1/2}}{2} & \text{else if } \mu_2 \le a_3, \\ \mu_3 \equiv \frac{(\sum_{i=1}^3 a_i) + [3(fdx)^2 - 2((\sum_{i=1}^3 a_i^2) - a_1 a_2 - a_1 a_3 - a_2 a_3)]^{1/2}}{3} & \text{otherwise.} \end{cases} \tag{22}$$

3.3 Pseudo-reinitialization

As noted in [4] when reinitializing after an advancement near the tube boundary, we cannot use straightforward eikonal equation solvers as we lose interesting portions of solutions that differ from the viscosity solution. To remedy this problem we use the same type of pseudo-reinitialization as was introduced in [4].

The idea is that instead of using the PDE method of reinitialization to a weighted signed distance function [30] which solves the equation

$$\phi_t + S(\phi)\left(|\nabla\phi| - \frac{1}{R(x)}\right) = 0, \tag{23}$$

we instead solve

$$\phi_t + S(\phi)\left(\nabla\phi \cdot \frac{\eta}{|\eta|} - |\eta|\right) = 0, \tag{24}$$

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

To solve (24) we first choose η. This is done in a Godunov type upwind manner. For each gridpoint $x_{i,j}$ we make the following choices:

$$\begin{aligned}
\eta_1 &= maxmod(\max(D_x^- \phi_{i,j,k}, 0), \min(D_x^+ \phi_{i,j,k}, 0))/dx, \\
\eta_2 &= maxmod(\max(D_y^- \phi_{i,j,k}, 0), \min(D_y^+ \phi_{i,j,k}, 0))/dy, \\
\eta_3 &= maxmod(\max(D_z^- \phi_{i,j,k}, 0), \min(D_z^+ \phi_{i,j,k}, 0))/dz,
\end{aligned} \tag{25}$$

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

$$\begin{aligned}
\eta_1 &= maxmod(\min(D_x^- \phi_{i,j,k}, 0), \max(D_x^+ \phi_{i,j,k}, 0))/dx, \\
\eta_2 &= maxmod(\min(D_y^- \phi_{i,j,k}, 0), \max(D_y^+ \phi_{i,j,k}, 0))/dy, \\
\eta_3 &= maxmod(\min(D_z^- \phi_{i,j,k}, 0), \max(D_z^+ \phi_{i,j,k}, 0))/dz,
\end{aligned} \tag{26}$$

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

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

Here

$$\begin{aligned}
D_x^{\pm} \phi_{i,j,k} &= \pm(\phi_{i\pm1,j,k} - \phi_{i,j,k}), \\
D_y^{\pm} \phi_{i,j,k} &= \pm(\phi_{i,j\pm1,k} - \phi_{i,j,k}), \\
D_z^{\pm} \phi_{i,j,k} &= \pm(\phi_{i,j,k\pm1} - \phi_{i,j,k}).
\end{aligned}$$

We note that more accurate W/ENO methods can also be used to construct η. At $\partial\Omega$ we enforce that there will be no incoming characteristics by taking $V_i = 0$ if V_i has the sign of an incoming characteristic, e.g. at the left boundary in x we take $V_1 = \min(V_1, 0)$. These boundary conditions are consistent with those used in (21), which employ an approximation to the Neumann boundary conditions, $\partial\phi/\partial n = 0$, when incoming characteristics are found.

Once η has been chosen we solve (24) using a fast sweeping method adapted to this linear PDE. The basic form of the steady state PDE is

$$V \cdot \nabla\phi = f, \tag{27}$$

where in this case $V = S(\phi)\eta/|\eta|, f = S(\phi)|\eta|$. In this case the Godunov Hamiltonian is found by upwinding depending on the sign of V_i. So we discretize ϕ_x, for example, by

$$\phi_x \approx \begin{cases} (\phi_{i,j,k} - \phi_{i-1,j,k})/dx, & \text{if } V_1 \geq 0, \\ (\phi_{i+1,j,k} - \phi_{i,j,k})/dx, & \text{if } V_1 < 0. \end{cases} \tag{28}$$

Thus we can write the discretized version of (27) as

$$\sum_{i=1}^{3} V_i(a_i \phi_{i,j,k} + b_i \phi_{\text{offset}_i})/dx_i = f, \tag{29}$$

where $a_i, b_i \in \{-1, 1\}$, and ϕ_{offset_i} is found from equations analogous to (28) by taking the indices of the point chosen in the approximation of ϕ_{x_i} that is different from $\phi_{i,j,k}$. If we note the dependence of a_i, b_i on the sign of V_i, then we can write (29) as

$$\sum_{i=1}^{3} |V_i|(\phi_{i,j,k} - \phi_{\text{offset}_i})/dx_i = f. \tag{30}$$

Then solving for $\phi_{i,j,k}$ we find

$$\phi_{i,j,k} = \frac{f + \sum_{i=1}^{3} |V_i|(\phi_{\text{offset}_i})/dx_i}{\sum_{i=1}^{3} |V_i|/dx_i}. \tag{31}$$

If $V_i = 0$ then is does not matter which offset point we choose, and in the cases where $\sum_{i=1}^{3} |V_i| \approx 0$ we set $\phi_{i,j,k}$ to the average of its neighbors with indices (a, b, c) such that $\|(a, b, c) - (i, j, k)\|_{l_1} \leq 1$.

The sweeping directions used are the same as those used for the eikonal equation, however, we do not initialize the grid to large positive values away from the fixed band. Rather outside the fixed band where $|\phi| \leq \delta = 1.5dx$ we use the current values of ϕ as initial values.

3.4 Perpendicularization

For perpendicularization we note that (9) can be discretized as (27) with $f = 0$ and $V = S(\psi)\nabla\psi/|\nabla\psi|$. The solution to this PDE is found by fast sweeping using the same framework and initialization, including Gauss-Seidel sweeps using (31) as was done in the pseudo-reinitialization case. In this case the fixed band is taken where $|\psi| \leq \delta = 1.5dx$. The vector V is found using central differencing, where we regularize the denominator in V, and correct for incoming characteristics as well.

3.5 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 another point $b \in \partial\Omega$ and has fixed topology. For codimension-1 level set dynamics there is a method, outlined in [13], that guarantees topology preservation. This was used in [4] to keep the path from changing topology in 2d. However, for codimension-2 curves in 3d there is no existing algorithm of this type. Therefore, in the spirit of the projected PDEs with which we have been working, we attempt to project the 2d method from [13] onto the surface with normal $\nabla\psi/|\nabla\psi|$.

The idea is that when we are changing the value of $\phi_{i,j,k}$, if the sign is being changed and we are near the set $\{\psi = 0\}$, then we attempt to find a local neighborhood, N, of ϕ values on $\{\psi = 0\}$, and project N to the standard 2d nine point neighborhood where the topology preservation can be enforced. To find N we first calculate the normal vector to ψ: $n_\psi = \nabla\psi(x_{i,j,k})/|\nabla\psi(x_{i,j,k})|$ using

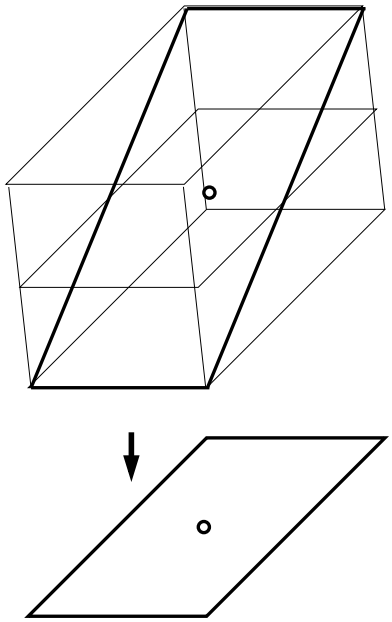

Figure 3: Projection of an 8 point 3d neighborhood onto a 2d plane. The bold lines represent the boundary of the neighborhood.

central finite differences. Then we find the normal vector w that approximates n_ψ such that $w \in W = \{x_{a,b,c} - x_{i,j,k}\}$, where $\|(a,b,c) - (i,j,k)\|_{l_\infty} = 1$. We do this by taking the $w \in W$ that maximizes $n_\psi \cdot w/|w|$. Once this approximate normal is found we fill N with all points such that $(x_{a,b,c} - x_{i,j,k}) \cdot w = 0$, where $\|(a,b,c) - (i,j,k)\|_{l_\infty} = 1$, i.e. we take points from the 26 nearest neighbors to $x_{i,j,k}$ that lie on the plane passing through $x_{i,j,k}$ with normal w. The cardinality of N is 8 unless $\|w\|_{l_1} = dx + dy + dz$, in which case it is 6.

Next, we project N onto a local 2d grid by finding a nonzero component of w and projecting to the plane perpendicular to this component direction. For example, if $w = (1,0,1)$, then we could project any node $x_{a,b,c} \in N$ onto $y_{b,c}$ or $y_{a,b}$. Once we have this 2d neighborhood, all of whose points lie on a regular 2d grid, we can apply the standard topology preservation method from [13]. See Figure 3 for an example of how N is projected onto a 2d plane when $card(N) = 8$.

If $card(N) = 6$ then we need to fill out the projected 2d neighborhood with 2 extra points, or else our algorithm will be too severe in its judgment of whether or not topology has changed. Thus we examine all three symmetric extensions of N that include the points, $x_{a,b,c}$, that are closest to $x_{i,j,k}$, are lying on our grid, and have the property that $(x_{a,b,c} - x_{i,j,k}) \cdot w = 0$. These extra points have the property that exactly one of the indices $\{a,b,c\}$ is offset from the index $\{i,j,k\}$ by ± 2. Figure 4 shows an example of the hexagonal neighborhood in 3d,

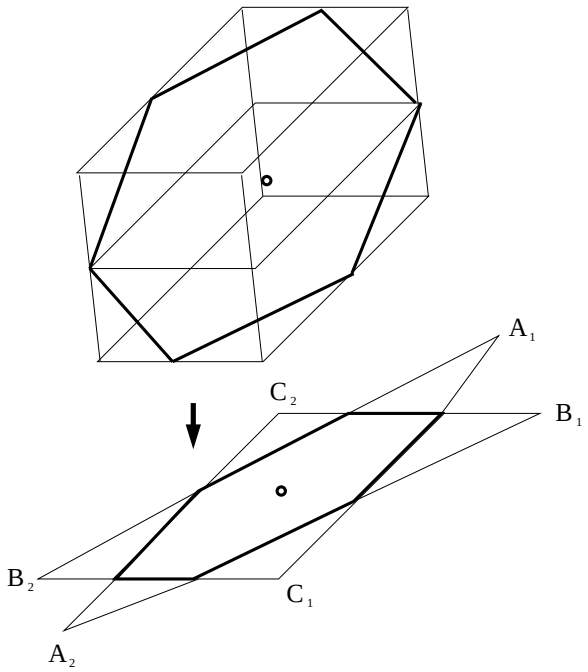

Figure 4: Projection of a 6 point 3d neighborhood onto a 2d plane. The bold lines represent the boundary of the neighborhood. The hexagon along with either $\{A_1, A_2\}, \{B_1, B_2\}$, or $\{C_1, C_2\}$ makes up an 8 point neighborhood that can be mapped to a standard 8 point rectangular neighborhood of the center point.

along with its three possible extended symmetric 8 point projections onto a 2d plane, each of which can be mapped in a one to one manner onto a rectangular 8 point neighborhood where the usual topology preservation algorithm can be run. For points where $card(N) = 6$ we run the topology change test on all three possible extended neighborhoods, and prohibit a sign change in ϕ at $x_{i,j,k}$ if any of the three tests indicates a topology change.

While the work done in constructing N is more significant than what was done in the 2d case, the codimension-2 nature of Γ_0 means that this procedure must be applied an order of magnitude fewer times during the evolution. In practice we require $|\psi| \leq 2dx$ before the topology preservation method is applied.

We also note that if it is necessary to keep the points where Γ_0 intersects the boundary (or any other subset of Ω) fixed, we imposed Dirichlet boundary conditions $\phi = \psi = 0$ at these points. 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.

3.6 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, ψ by adding or subtracting (a_i, b_i) at all points so that identical equations being solved near $\Gamma_{(a_i,b_i)}$ can be solved using the same coded functions for all i. 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 $\phi + a_i$. It is assumed that after the step in question is completed, that ϕ is then shifted back the opposite way by $-a_i$. This is done similarly with ψ.

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 system of the form:

$$\phi_t = P(x)\delta(r - \sqrt{\phi^2 + \psi^2})\frac{-\phi}{\sqrt{\phi^2 + \psi^2}} - \lambda k N \cdot \nabla\phi, \tag{32a}$$

$$\psi_t = P(x)\delta(r - \sqrt{\phi^2 + \psi^2})\frac{-\psi}{\sqrt{\phi^2 + \psi^2}} - \lambda k N \cdot \nabla\psi \tag{32b}$$

where we will use the substitutions

$$\begin{aligned} &\delta(r - \sqrt{\phi^2 + \psi^2}) \to |\nabla\phi| C_r(r - \|(\phi, \psi)\|), \text{ in (32a) or} \\ &\delta(r - \sqrt{\phi^2 + \psi^2}) \to |\nabla\psi| C_r(r - \|(\phi, \psi)\|) \text{ in (32b),} \end{aligned} \tag{33}$$

when they are implemented numerically.

1. Advance functions near $\Gamma_{(a_i,b_i)}$, for $i = 1, \ldots M$ where we choose (a_i, b_i) such that $\|(a_i, b_i)\| = r$.

   (a) Find $\eta \approx \nabla\phi$ based on $\phi - a_i, \psi - b_i$ using (25), (26).

   (b) Evolve ϕ from time t_1 to $t_1 + dt$ the points near where $\phi = a_i, \psi = b_i$, i.e. evolve all ϕ_t PDE terms with $\delta(r - \sqrt{\phi^2 + \psi^2})$ in them, e.g.

   $$\phi_t = P(x)\delta(r - \sqrt{\phi^2 + \psi^2})\frac{-\phi}{\sqrt{\phi^2 + \psi^2}}.$$

   (c) Pseudo-reinitialize ϕ using a banded fast sweeping method within a fixed band around $\{\psi - b_i = 0\}$, using $\{\phi - a_i \approx 0\}$ as the points where ϕ is fixed during the sweeping process.

   (d) Find $\eta \approx \nabla\psi$ based on $\phi - a_i, \psi - b_i$ using equations (25), (26) with ψ substituted in place of ϕ.

(e) Evolve ψ from time t_1 to $t_1 + dt$ the points near where $\phi = a_i, \psi = b_i$, i.e. evolve all ψ_t PDE terms with $\delta(r - \sqrt{\phi^2 + \psi^2})$ in them, e.g.

$$\psi_t = P(x)\delta(r - \sqrt{\phi^2 + \psi^2})\frac{-\psi}{\sqrt{\phi^2 + \psi^2}}.$$

(f) Pseudo-reinitialize ψ using a banded fast sweeping method within a fixed band around with $\{\phi - a_i = 0\}$, using $\{\psi - b_i \approx 0\}$ as the points where ψ is fixed during the sweeping process.

2. Advance functions near Γ_0.

(a) Evolve ϕ from time t_1 to $t_1 + dt$ the points near where $\phi = \psi = 0$, i.e. evolve all ϕ_t PDE terms with $\delta(\sqrt{\phi^2 + \psi^2})$ in them, e.g.

$$\phi_t = -\lambda k N \cdot \nabla\phi.$$

(b) Reinitialize ϕ using a banded fast sweeping method within a fixed band around with $\{\psi = 0\}$, using $\{\phi \approx 0\}$ as the points where ϕ is fixed during the sweeping process.

(c) Perpendicularize ϕ with fast sweeping, using $\{\psi \approx 0\}$ as the points where ϕ is fixed during the sweeping process.

(d) Evolve ψ from time t_1 to $t_1 + dt$ the points near where $\phi = \psi = 0$, i.e. evolve all ψ_t PDE terms with $\delta(\sqrt{\phi^2 + \psi^2})$ in them, e.g.

$$\psi_t = -\lambda k N \cdot \nabla\psi.$$

(e) Reinitialize ψ using a banded fast sweeping method within a fixed band around with $\{\phi = 0\}$, using $\{\psi \approx 0\}$ as the points where ψ is fixed during the sweeping process.

(f) Perpendicularize ψ with fast sweeping, using $\{\phi \approx 0\}$ as the points where ψ is fixed during the sweeping process.

4 Numerical Simulations

In this section we present some numerical simulations. The PDE we evolve to steady state is (32), using the methods mentioned above. The domain Ω is $[-1, 1]^3$ for all problems, discretized in a uniform rectangular grid. A conservative estimate on the CFL condition for the problem is

$$dt \ \max\left\{\frac{|P|}{dx} + \frac{|P|}{dy} + \frac{|P|}{dz}, \lambda k_{\max}\left(\frac{1}{dx} + \frac{1}{dy} + \frac{1}{dz}\right)\right\} \leq 1, \tag{34}$$

where $k_{\max}$ is the magnitude of the maximum curvature that we allow to be discretized on the grid. In practice this is set at $1/dx$. We use the max applied to

the P and k 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 show them in contour plots. The way they are constructed is by determining an initial curve $\Gamma_0(t=0)$ and finding arbitrary functions that have $\{\phi(t=0)=0\}\cap\{\psi(t=0)=0\}=\Gamma_0(t=0)$, and then running reinitializations and perpendicularizations if necessary. More details about the initializations can be found in [2], or in [34] where Clebsch variables (from hydrodynamics) that are constant along particle paths are discussed.

For some figures the energy vs. time steps plots are shown, where the energy to be maximized is defined by (4) summed with $-\lambda|\Gamma|$ using (10). The δ and Heaviside functions used are the compactly supported ones given in [5], with support parameter $\epsilon = 2dx$. It should be noted that a more accurate numerical construction of these singular functions can be found in [11] should a more exact measure of the energy be needed.

In Figure 5 we show an example where S_Γ is a tube of radius 0.2 and there is an obstacle (a sphere with a tunnel through it) where $P(x) = -1$. Outside of the obstacle $P(x) = 0$. Here we fix the boundary of the tube where $x = \{\pm 1\}$. The regularization parameter $\lambda = 0.01$. We use a uniform rectangular discretization with $dx = dy = dz = 2/50$. The number of S_Γ advancements is $M = 5$. At each time step these are chosen using (17), where θ_0 is chosen randomly. In the figure it is seen that the tube locates the tunnel and avoids the areas where $P(x) < 0$.

In Figure 6 we show two examples starting form identical initial conditions, but where topology preservation is enforced in one example but not in the other. Here S_Γ is a tube of radius 0.2, and there are two balls where $P(x) > 0$, a box where $P(x) < 0$, and $P(x) = 0$ elsewhere. The regularization parameter $\lambda = 0.02$. We use a uniform rectangular discretization with $dx = dy = dz = 2/50$. The number of S_Γ advancements is $M = 4$. At each time step these are chosen using (17), where θ_0 is chosen randomly. In the figure we see that the topology is prohibited from changing when the topology preservation is enforced.

In Figure 7 we show an example where pseudo-reinitialization allows for cusped regions to form. Here S_Γ is a tube of radius 0.2, and there is a rectangle where $P(x) > 0$. The regularization parameter $\lambda = 0.01$. We use a uniform rectangular discretization with $dx = dy = dz = 2/50$. The number of S_Γ advancements is $M = 4$. At each time step these are chosen using (17), where θ_0 is chosen randomly. The figure demonstrates how cusped regions are allowed to form when pseudo-reinitialization is used as opposed to standard reinitialization. If standard reinitialization is used instead of pseudo-reinitialization, then we do not capture the cusped region and the result is that of Figure 8.

In Figure 9 we show an example where the path width is spatially varying. In this example $R(x) = 1$ when $x \leq 0$, while $R(x) = 2$ when $x > 0$. Here S_Γ is a tube of radius 0.15, and there is a sphere where $P(x) = -1$ centered near the origin. The regularization parameter $\lambda = 0.02$. We use a uniform rectangular discretization with $dx = dy = dz = 2/50$. The number of S_Γ advancements is $M = 5$. At each time step these are chosen using (17), where θ_0 is chosen

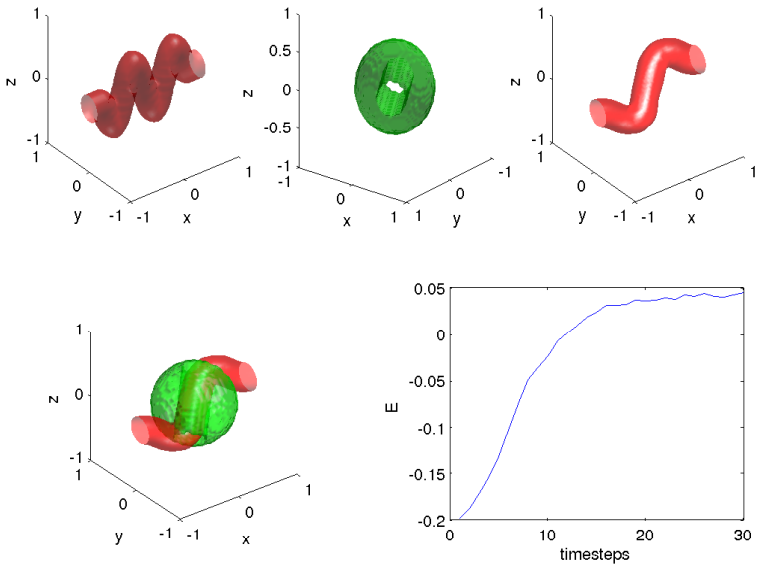

Figure 5: Top, left to right: Initial condition, boundary of region where $P < 0$, final solution. Bottom, left to right: Final solution together with boundary of region where $P < 0$, and energy vs. time steps plot.

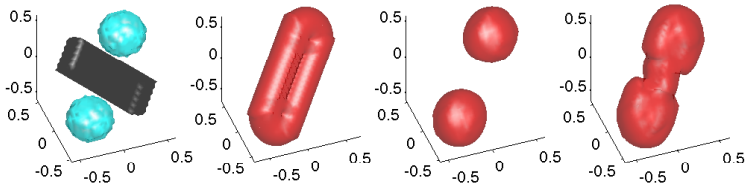

Figure 6: Left to right: Boundary of sets where $P > 0$ (blue spheres) and $P < 0$ (black box), initial condition of tube, final tube with no topology preservation, final tube with topology preservation.

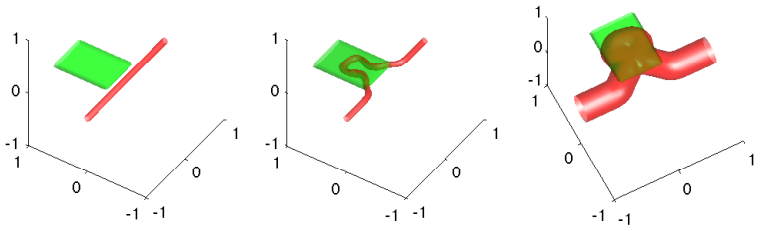

Figure 7: Left to right: Initial Γ and boundary of rectangle where $P > 0$, advanced Γ, and advanced S_Γ. In this example pseudo-reinitialization is used.

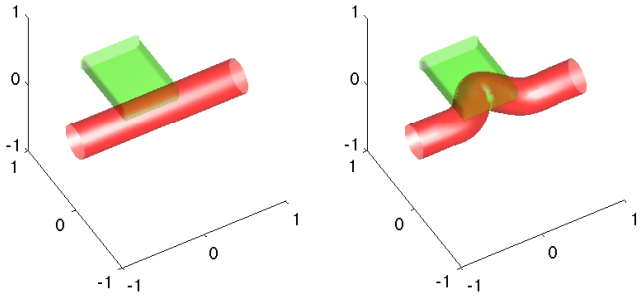

Figure 8: Left to right: Initial S_Γ and boundary of rectangle where $P > 0$, and advanced S_Γ. In this example standard reinitialization is used instead of pseudo-reinitialization.

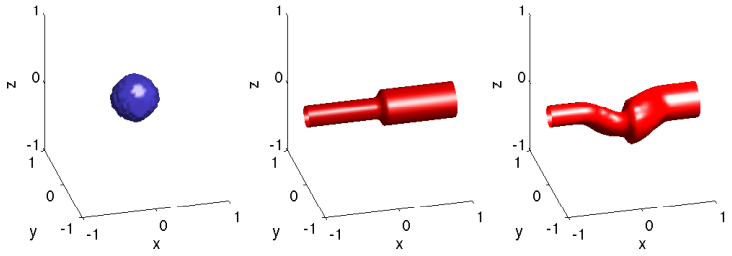

Figure 9: Left to right: Boundary of set where $P < 0$, initial S_Γ, and final S_Γ. In this example $R(x)$ is spatially varying.

randomly.

5 Conclusion

We have presented an extension of the level set based algorithm for solving a variational approach to path planning that was originally proposed in [4]. This involved adapting the codimension-2 level set framework established in [2] for motions of curves in $\mathbb{R}^3$. Some key features of this algorithm are the energy integrals used to define the search criteria, the splitting technique used to advance the PDEs, the fast methods for reinitialization, pseudo-reinitialization, prependicularization, and the topology preservation for curves in $\mathbb{R}^3$.

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, torsion, or other path properties can be constructed.

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. Also, it may be possible to use the ideas from [23] to use this 3d method to construct self-intersecting paths for problems with 2d domains. Level set motion of tubes having small positive width may also have applications in image processing, such as segmentation of filaments or other thin objects such as blood vessels.

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.

We would also like to thank Y.-H. Tsai for helpful discussions regarding the fast sweeping techniques.

References

[1] Steven J. Altschuler. Singularities of the curve shrinking flow for space curves. *J. Differential Geom.*, 34(2):491–514, 1991.

[2] Paul Burchard, Li-Tien Cheng, Barry Merriman, and Stanley Osher. Motion of curves in three spatial dimensions using a level set approach. *J. Comput. Phys.*, 170(2):720–741, 2001.

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

[4] T. Cecil and D. Marthaler. A variational approach to search and path planning using level set methods. *UCLA CAM Report, Submitted to Interfaces and Free Boundaries*, (04-61), 2004.

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

[6] 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.

[7] Li-Tien Cheng, Paul Burchard, Barry Merriman, and Stanley Osher. Motion of curves constrained on surfaces using a level-set approach. *J. Comput. Phys.*, 175(2):604–644, 2002.

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

[9] Michael G. Crandall and Pierre-Louis Lions. Viscosity solutions of Hamilton-Jacobi equations. *Trans. Amer. Math. Soc.*, 277(1):1–42, 1983.

[10] M.G. Earl and R. DAndrea. Modelling and contro of a multi-agent system using mixed integer linear programming. In *Proceedings of the 41st IEEE Conference on Decision and Control, December 2002*.

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

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

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

[14] L. Kavraki, M. Kolountzakis, and J. Latombe. Analysis of probabilistic roadmaps for path planning. *IEEE Transactions on Robotics and Automation*, 14(1), 1998.

[15] L. Kavraki, P. Svestka, J. Latombe, and M. Overmars. Probabilistic roadmaps for path planning in high dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4), 1996.

[16] 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).

[17] Y. Koren and J. Borenstein. Potential field methods and their inherent limitations for mobile robot navigation. In *Proceedings of the IEEE Conference on Robotics and Automation, Sacremento, CA, April 1991.*

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

[19] 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.

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

[21] Facundo Mémoli and Guillermo Sapiro. Fast computation of weighted distance functions and geodesics on implicit hyper-surfaces. *J. Comput. Phys.*, 173(2):730–764, 2001.

[22] B. Q. Nguyen, Y-L Chuang, D. Tung, C. Hsieh, Z. Jin, L. Shi, D. Marthaler, A. L. Bertozzi, and R. M. Murray. Virtual attractive-repulsive potentials for cooperative control of second order dynamic vehicles on the Caltech MVWT. *submitted to the American Control Conference 2005.*

[23] Stanley Osher, Li-Tien Cheng, Myungjoo Kang, Hyeseon Shim, and Yen-Hsi Tsai. Geometric optics in a phase-space-based level set and Eulerian framework. *J. Comput. Phys.*, 179(2):622–648, 2002.

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

[25] 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.

[26] A. Richards and J. How. Aircraft trajectory planning with collision avoidance using mixed integer linear programming. In *IEEE American Control Conference, May 2002.*

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

[28] 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.

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

[30] 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.

[31] 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.

[32] 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.

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

[34] V. E. Zakharov, S. L. Musher, and A. M. Rubenchik. Hamiltonian approach to the description of nonlinear plasma phenomena. *Phys. Rep.*, 129(5):285–366, 1985.

[35] H.K. Zhao. A fast sweeping method for eikonal equations. *to appear in Mathematics of Computation.*

[36] H.K. Zhao, S. Osher, B. Merriman, and M. Kang. Implicit and non-parametric shape reconstruction from unorganized points using variational level set method. *Comp. Vis. and Image Under.*, 80:295–319, 2000.