LA-UR- 11-01223

Approved for public release; distribution is unlimited.

Title: Determination of Minimum Time Rendezvous Points for Multiple Agents Via Level Set Methods

Author(s): Travis Brown, University of Notre Dame
Tariq Aslam, LANL WX-9
James Schmiedeler, University of Notre Dame

Intended for: ASME 2011 International Design Engineering Technical Conferences (IDETC) and Computers and Information in Engineering Conference (CIE)
Aug 28-31, 2011
Washington, DC

Los Alamos
NATIONAL LABORATORY
EST. 1943

Determination of Minimum Time Rendezvous Points for Multiple Agents Via Level Set Methods

T.L. Brown, T.D. Aslam, J.P. Schmiedeler

February 18, 2011

Abstract

This paper addresses the problem of finding the minimum time rendezvous point for a geographically distributed group of heterogeneous mobile agents. This would be useful in any situation where agents must regroup quickly to achieve some objective. In contrast to the traditional treatment of the multi-agent rendezvous problem, focus is given mainly to the identification of the globally optimal solution rather than the behavior of the system based on a given control policy. Level sets are introduced as a tool to solve this problem by first computing an arrival time map for each agent, subject to speed, terrain, and dynamic constraints. The computation is parallelizable by requiring each agent to generate its own arrival time map. The arrival time maps can be easily combined to give the overall minimum time rendezvous point. Despite the apparent simplicity of this approach, it is capable of accommodating numerous complicating factors with minimal modification while simultaneously generating a target path trajectory for each agent through the state-space. Examples involving ground, sea, and air robots are used to illustrate the power of this technique.

1 Introduction

The need for a group of autonomous vehicles to come together at some location may arise from a broad range of practical mission requirements. In a military context, this may be for protection or fuel dispersement; in a scientific survey, it could be to return samples to a cargo robot; in service robots it may permit smaller robots to dock with larger robots for transportation. This so called rendezvous problem is a classic problem in the field of distributed control[1]. Distributed control generally deals with decision making and consensus within groups of autonomous agents given restrictions on global information and communication bandwidth, reliability, and latency. For the rendezvous problem, the existence and stability of a consensus-based rendezvous point are typically examined[5][4], but little attention is given to the actual location of the rendezvous point. This is consistent with the control of UAV's for which the environment is simple and obstacles are rare. Furthermore, communications restrictions are consistent with small air vehicles that must be light, thus limiting on-board computation and communication equipment. These assumptions do not hold when applied to ground vehicles. In the complicated practical environments in which ground robots operate, the location of the rendezvous point becomes significant, and the availability of global information and high bandwidth communication is more likely. The existence of rough terrain and fixed obstacles like buildings and walls may impose serious restrictions on the movement of ground robots that must be addressed in any decision making process. Furthermore, the slow speed of most ground robots requires that attention be paid to optimizing the path of each robot and the path of the group as it moves on toward the next objective. Thus, traditional approaches to rendezvous are, in some sense, ill-suited for practical vehicles operating in more complex conditions.

2 Level Set Methods

With access to global environment data and adequate communications, finding the optimal rendezvous point becomes a question of path planning. Using path planning techniques, the feasibility of every possible rendezvous point may be evaluated by each robot and the optimal meeting point may be found. The problem of robot path planning has been well studied from both theoretical and practical perspectives[8]. Common techniques include grid-based graph search algorithms, sampled graph search (e.g. Probabilistic Roadmap), and gradient or potential field approaches. Level set methods are somewhat analogous to a breadth first search over a rectangular grid. For the simplest static point-to-point navigation problems, level set methods are at a computational disadvantage relative to many other techniques, but have the advantage of providing solutions that are both correct and complete under resolution. This means that given a fine enough grid, these technique will find all possible paths and correctly predict the actual cost of each one. Thus, while point-to-point algorithms must be repeated for each possible destination at great computational cost, level set methods require practically no modification. Thus, level set methods are intrinsically well suited to problems that, like the rendezvous problem, take advantage of their complete nature.

Level set methods have recently become prevalent in several fields. Their most common application is in high fidelity simulators for fluid flows, and thermodynamics systems, however they have also found application in computer vision and robotic path planning. Hassouna et. al.[6] use the flexibility of level set methods to make 2D path planning more robust. Kimmel[2] et al use level sets to find shortest paths over 3D surfaces. A full treatment of the subject is given by Osher[7].The basic principle underlying the level set method is that many problems can be framed in terms of an interface function that satisfies a hyperbolic PDE; fast PDE solvers can then be used to yield solutions to these problems. To adapt this to robot path planning, a propagation PDE that correctly moves the reachable frontier of the robot must be created. For robots with negligible dynamics, this corresponds to normal wave propagation emanating from the starting point. By careful choice of the governing PDE, the dynamics of a given robot can be embedded into the problem. Despite the potential benefits of this approach it has yet to be adopted as a mainstream path planning technique.

The main method to find rendezvous points discussed in this paper is the min-max method. In this method the arrival time for each agent is computed and the maximum of all the arrival time maps is computed at each point. This parallelism makes these techniques more suitable to multi-agent problems, assuming these agents can be fitted with reasonably powerful processing equipment. The location with the smallest maximum is the optimal rendezvous point. A more detailed discussion of the method is provided in the next section. To illustrate the flexibility and applicability of this technique, three example problems are discussed. The first problem deals with navigation in complex terrain, the second deals with the integration of dynamic effects into the level set method, and the third deals with applying level set in dynamic and non-rectangular environments.

3 Ground Terrain Navigation

This problem gives an example of the level set technique applied to a simple 2D environment with static obstacles and non-uniform terrain. The assumption taken here is that the time scales of the dynamics of the agent are negligible compared to the time scales involved with traversing the environment. This would be valid for most ground robots since they are typically designed to move slow enough to ignore dynamic effects. This would break down for faster moving robotic vehicles like those seen in the DARPA Grand Challenge. If applied to human ground units this restriction is also of minimal consequence since humans have no trouble navigating within local dynamic environments.

3.1 Problem Description

A simple rendezvous problem that is stated as follows. Two agents start at locations A and B respectively, find the rendezvous point that minimizes the rendezvous time and thus allows them to meet in the shortest possible time. Route each must take to achieve this optimal rendezvous time should also be determined. In this case, the agent at location A is defined to be a legged robot and the agent at location B will be a wheeled robot. It is assumed that the wheeled vehicle is faster on defined paths but is slowed considerably when deviating from defined paths. For added complexity, the walking robot and the wheeled robot must proceeding to the final destination at C, limited by the speed of the slowest robot.

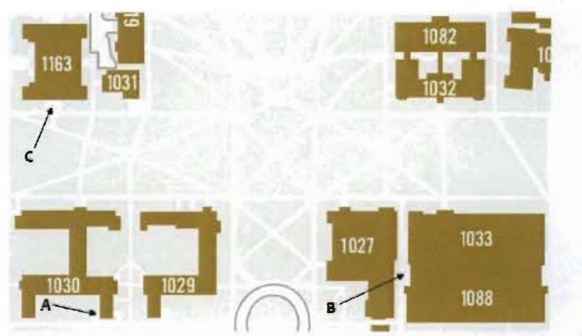

Figure 1: Starting and Ending Locations on Campus

It is easy to see that the rendezvous constraint will increase the time it takes for the agents to arrive at the final destination. This is because the agents will be forced to deviate from the optimal path in order to first proceed to the intermediate destination. By moving the rendezvous point closer to the final destination this effect can be reduced but at the cost of increasing the rendezvous time. It will be shown that the level set method lends itself to a convenient way of balancing these two competing goals of minimizing the rendezvous time and minimizing the overall transit time.

3.2 Method

In the level set approach requires the existence of $\psi(x, y, t)$, where the $\psi = 0$ level set defines the time dependent interface between reachable and unreachable regions of the map for a given agent. $\psi < 0$ represents points that are reachable and $\psi > 0$ represents points that are unreachable at a given time step. The ψ function then evolves subject to a hyperbolic partial differential equation that describes the movement abilities of the agent. The main computational cost of the level set approach is the construction of the numerical solution of the psi function over space and time. The PDE for this example is

$$\frac{\partial \psi}{\partial t} + u\frac{\partial \psi}{\partial x} + v\frac{\partial \psi}{\partial y} = 0 \qquad (1)$$

where u and v define the speeds in the x and y directions respectively. For the purposes of this example it is assumed that both the walking robot and the wheeled robot can move in any direction at anytime with speed subject to a navigability constraint that limits the maximum speed based on a given terrain.

$$\sqrt{u^2 + v^2} <= s_{max} * c(x, y) \tag{2}$$

The navigability of the grid is represented at all points by a navigability index c ranging from 0 to 1. High navigability indexes represent locations at which an agent can move at or near it's maximum speed. Low indexes represent terrain which severely impedes an agent's progress. An index of zero represents a completely impassible terrain. In practice different agents may have different navigability maps, for example wheeled vehicles are impeded by rough terrain much more than a person on foot so the navigability index for a vehicle in these areas would be comparatively lower. In this example a subset of the Notre Dame campus map (Figure 7) is taken to be the operating region of interest. The navigability indices are taken from a published map of campus that represents buildings, grass, and sidewalks as dark, light, and white respectively. Taking the grayscale values of this map at each point provides a basis for generating navigability maps for this area. For the walking robot, thresholding is used to provide a map that is zero for buildings and 1 for everything else. In the wheeled case, this map could be used directly in its current form but to make the problem more interesting the navigability values have been modified to increase the penalty of non-sidewalk paths.

To minimize the ψ function and thus include the largest amount of terrain inside the reachable region u and v are chosen such that the total velocity is normal to the level set. This causes the largest increase in the reachable region and thus defines the maximal reachable set at a given time as desired. This can be accomplished by choosing

$$u = \frac{c(x,y) \times \psi_x}{\sqrt{\psi_x^2 + \psi_y^2}}, v = \frac{c(x,y) \times \psi_y}{\sqrt{\psi_x^2 + \psi_y^2}} \tag{3}$$

The initial conditions for ψ are given by the signed distance function with a small radius r around the starting point.

$$\psi_0(x, y) = \sqrt{(x - x_0)^2 + (x - x_0)^2} - r \tag{4}$$

The solution of (1) can be found using various numerical integration techniques which are beyond the scope of this work. For this example a first order upwind scheme with Euler time integration was used. At each iteration in time a check is performed to see if ψ has become negative for a given grid point and the first such time is recorded in a separate grid. This yields a first crossing time function which is the quantity of interest. The gradient of the crossing time map can be followed back to the starting location to determine the optimal route. At this point it is important to stress the power of this technique. Not only does it yield the transit times from the start location to any point on the map, but it also provides an easy method for calculating the optimal route necessary to achieve those transit times. The solution is complete in that it covers every point in the domain.

Figures 2 and 3 show contour plots for the walking and wheeled robot crossing times at 10 second intervals with the calculated optimal path shown in red. For the case of the wheeled robot, the method for penalizing non-sidewalk movement has led to a path that predominantly favors the sidewalks as desired.

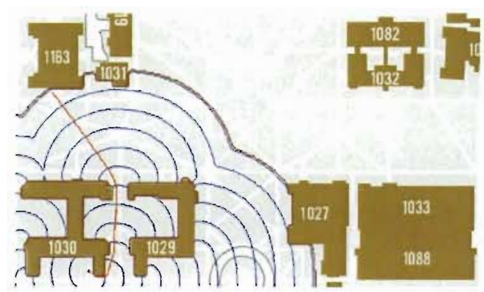

Figure 2: Legged Robot (A) crossing time map

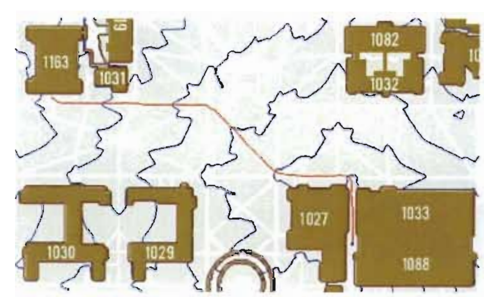

Figure 3: Wheeled Robot (B) crossing time map

Once the level set solution has been generated, many useful properties of the system may be calculated with practically no computational cost. The minimum time rendezvous point is one such property. The first step is to generate the rendezvous time map for the two robots as given by

$$t_r(x, y) = max(t_A(x, y), t_B(x, y)), \tag{5}$$

where $t_A(x, y)$ and $t_B(x, y)$ are the arrival times of the two robots are a given location. The minimum rendezvous point is the values of x and y that minimizes t_r. The overall transit time to any destination is given by the sum of the rendezvous time and the transit time from that rendezvous point to the destination.

$$t_{transit}(x,y) = t_r(x,y) + t_C(x,y) \tag{6}$$

As stated earlier it may be desirable to relax the rendezvous constraint in favor of minimizing the overall transit time. This can be accomplished simply by finding the x and y that minimize a weighted transit time as given by

$$t_{transit}(x,y) = max(t_r(x,y), t_r') + \epsilon t_C(x,y), \tag{7}$$

where $t_r' \in [t_r, t(x_C, y_C)]$ is a new specified rendezvous time and epsilon is a very small value. The max operation results in a flat floor in the rendezvous time map which is then slightly weighted toward the destination by the ϵ term. For values of t_r' close to t_r, the rendezvous time map is the dominant factor but as t_r' approaches $t(x_C, y_C)$, the rendezvous map becomes flatter and the small effect of $t(x_C, y_C)$ becomes significant. Thus the level of rendezvous relaxation can be controlled by specifying a less ideal rendezvous time.

3.3 Results

The following figure shows the rendezvous time map as calculated using equation 5. From the figure it can be seen that there are 2 minimum areas in two geographically separated locations. The lower location has a slightly lower rendezvous time so it is chosen as the optimal solution. By applying the rendezvous relaxation discussed above the rendezvous points can be moved to the upper location. It can be seen that the rendezvous points do not always vary continuously with t_r' but will occasionally jump from one optimal area to another as necessary.

Figure 4: Rendezvous time map

Figures 6 and 7 illustrate the dramatic difference in the final solution that the rendezvous relaxation method provides. In the pure min-max method the path is fairly complicated and seems unnecessarily long. With the relaxation method, the path is straight forward and intuitive.

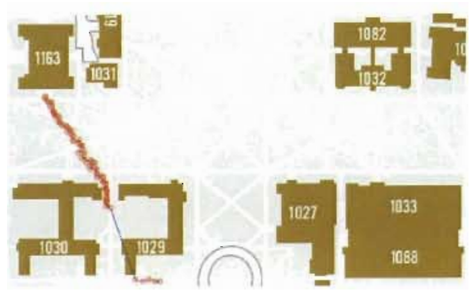

Figure 5: Rendezvous Locations with relaxation

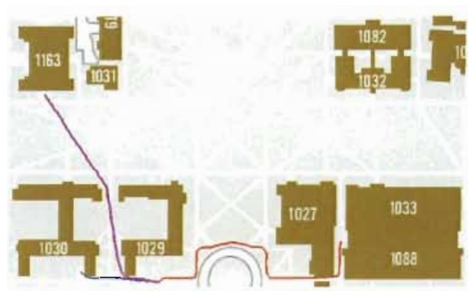

Figure 6: Optimal paths with min-max

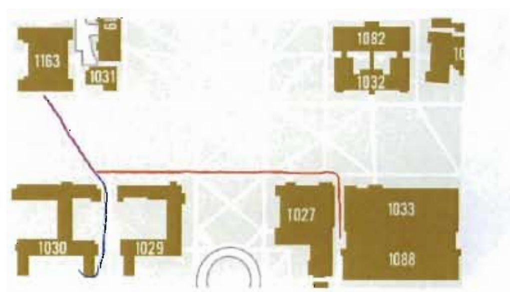

Figure 7: Optimal paths with relaxation

4 Ocean Navigation with Dynamics

The previous example showed a problem for which many solution techniques could be adapted. In the following problem, the dynamics of the system cannot be ignored. In this case, the algorithmic choices are much more restricted, and it is here that the advantages of level set methods are apparent. It is very difficult to modify most graph search techniques to account for the dynamics of the system, but such modifications are trivial for level set methods. The only disadvantage arises from the necessary exponential increase in time and space complexity incurred by adding an additional dimension.

4.1 Problem Descriptions

The example problems that will be addressed in this section concern a large robot carrier ship that moves through the ocean at speed s on a heading of θ. Heading and speed can be adjusted by setting the rudder 'r' and speed 's'. A reasonable approximation for the dynamics of the ship might be

$$\frac{dx}{dt} = s * cos(\theta) \quad (8)$$

$$\frac{dy}{dt} = s * sin(\theta) \quad (9)$$

$$\frac{d\theta}{dt} = r * s \quad (10)$$

where $s \in [-s_{max}, s_{max}]$ and $r \in [-r_{max}, r_{max}]$. The ship is initially located at $(x_0, y_0) = (5, 5)$ moving on a heading of $\theta = \pi/4$. Additionally, there is a small unmanned underwater vehicle (UUV) off the starboard side of the ship that can move in any direction at a maximum speed of 2. The goal is to maneuver both the ship and the UUV in order to minimize the rendezvous time. A rendezvous in this case is taken to be a coincidence in x and y since the scout does not have a heading. It is also assumed that the ship can drop anchor at any point and wait for the UUV to catch up.

The solution of a second similar problem is also shown in this section. It consists of the robot carrier and a robotic escort craft that follows the same dynamics equations as the carrier but starts at a different set of initial conditions, $(x_0, y_0) = (7, 2)$ on a heading of $\theta = 3\pi/4$. The escort must maneuver along side the ship by matching its location and heading. The problem is to minimize the amount of time it takes to get into formation with the ship and identify the path that both vessels should follow. This is a rendezvous problem in 3 dimensions. Here again it is assumed that either ship can drop anchor and wait for the other ship to catch up.

4.2 Method

The solution of this problem will use the same level set method used in the campus rendezvous problem, but with some special considerations. The PDE for this problem is

$$\frac{\partial\psi}{\partial t} + s * cos(\theta)\frac{\partial\psi}{\partial x} + s * cos(\theta)\frac{\partial\psi}{\partial y} + s * r\frac{\partial\psi}{\partial\theta} = 0 \quad (11)$$

With the assumption of constant speed, the only way of affecting ψ is through control of the rudder. To maximize the propagation of the $\psi = 0$ level curve, $s * r\frac{\partial\psi}{\partial\theta}$ must be maximized. Thus, the control becomes

$$r = r_{max} * sign(\frac{\partial\psi}{\partial\theta}). \quad (12)$$

Before proceeding, note that the equations specify the change in θ as a function of s and r. Consider the case of a ship moving in a circular path centered at a point.

$$\dot{\theta} = \frac{V}{radius} \quad (13)$$

$$= V\kappa \quad (14)$$

where κ is the curvature of the circle. This matches the dynamic equation for θ. The rudder angle is therefore equivalent to the instantaneous curvature of the path on which the ship will move.

The ψ function for the UUV is governed by a PDE that is identical to that discussed in the campus navigation problem. The control for the UUV is therefore chosen to be equal to equations 3.

In the case of rendezvous with a second ship, max maps at each heading level are computed to find intersection points for all three dimensions

4.3 Results

For this problem values of $s = 10$ and $r_{max} = .5$ were chosen. Figure 8 shows a composite crossing time map for the carrier at a resolution of $Nx = 200, Ny = 200, N\theta = 200$. This map was created by taking the minimum values of the crossing time for a given value of x and y over all headings. It is essentially a 2D projection of the 3D crossing time array. A visual inspection confirms that the maximal circular paths have a radius approximately equal to 2, which is consistent with our choice of r_{max}.

Solutions for this problem were computed with a variety of solvers. The best results were obtained using a 1st order Lax-Friedrichs scheme, but 2nd order RK with WENO5 and 3rd order RK with WENO 5 were performed. Figures 9 and 10 show the results of the 1st order LLF scheme and the 3rd order RK with WENO5 for N=50

To test the accuracy and convergence properties of the 1st order LLF scheme, solutions were generated at different levels of resolution. Figure 11 shows a table L1 error norms computed against the $N = 200$ resolution level. Convergence rates are better than first order which is surprising for a first order scheme.

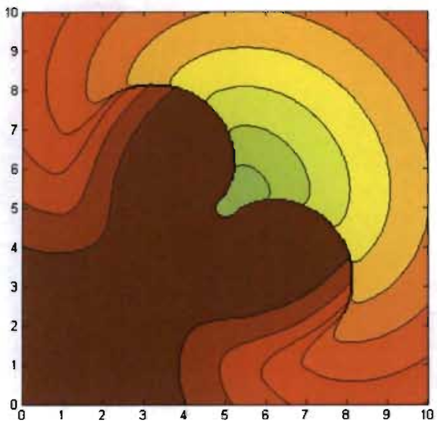

Figure 8: Composite crossing time map for ship starting at (5,5, pi/4)

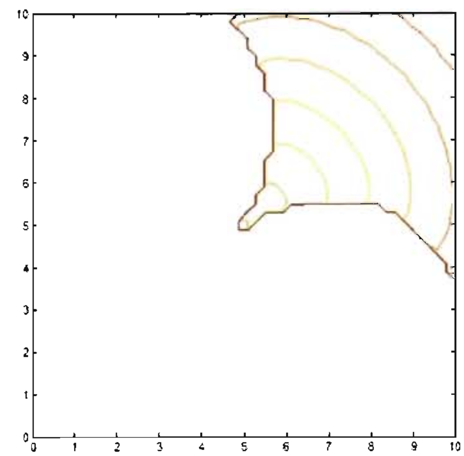

Figure 10: Composite crossing time map generated with RK3 and WENO5

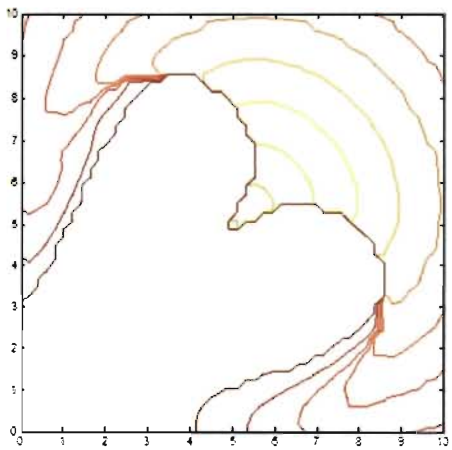

Figure 9: Composite crossing time map generated with RK1 and LLF

Nx	Ny	Nh	Nt	L1 Error	rc
25	25	25	100	0.8255	1.141459
50	50	50	200	0.3742	1.381754
100	100	100	400	0.1436	
200	200	200	800		

Figure 11: Table of L1 errors of crossing time map

Figure 12 shows the crossing time map for the UUV and the robot carrier overlaid to see the intersection of the two maps. From these maps, the technique used for the campus map can be followed to determine the minimum time rendezvous point.

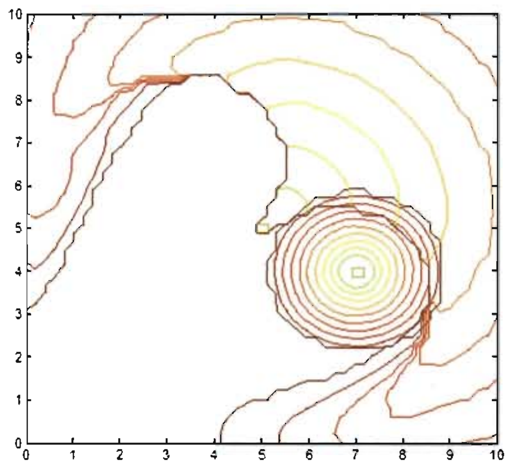

Figure 12: Crossing time paths for the UUV and the robot carrier

Figure 13 shows the composite crossing time map for the two ship case. To generate a path through state space, the gradient climbing approach can be extended to 3D and applied to find the path to any intersection point.

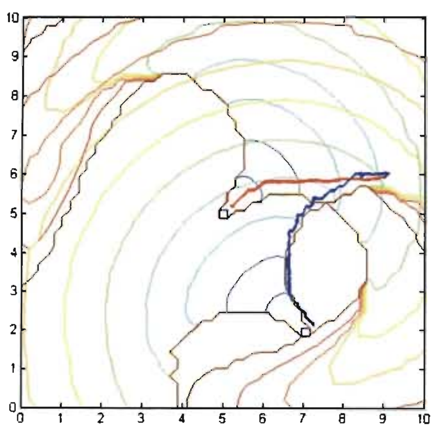

Figure 13: 2D projection of the optimal paths

5 Global Flight Navigation

The following problem shows that level set methods easily handle path planning in environments with moving obstacles. This problem also demonstrates the algorithm's ability to operate in spaces defined by non-rectangular grids. Both of these abilities make level set methods unique if not superior to other path planning methods in terms of flexibility.

5.1 Problem Description

Consider the task of aircraft navigation over large distances. At these distances the curvature of the earth becomes a significant effect on the path planning problem. For this problem, it is assumed that there are two aircraft, a UAV, and a refueling robot. The fighter moves at a maximum speed of mach 2 while the tanker moves well under the speed of sound. Now also assume the existence of a large volcanic ash cloud through which neither aircraft may pass. The ash cloud is subject to advection forces and thus moves with time. The fighter and the tanker must rendezvous in the minimum time possible while avoiding the ash cloud. This is a rendezvous problem in a dynamic environment defined over a spherical geometry. Here again the flexibility of the level set method makes solving this problem relatively easy.

5.2 Methods

The two main differences for this problem are the geometry and the dynamic environment. The method can be adapted to spherical geometry with minimal effort as described below.

The space must first be discretized into a grid of angles that uniquely describes every point on the sphere. The angles that will be used are α and β, where alpha defines the longitude, and beta defines the latitude. These angles will replace x and y as state space variables. The PDE for this configuration is

$$\frac{\partial \psi}{\partial t} + A\frac{\partial \psi}{\partial \alpha} + B\frac{\partial \psi}{\partial \beta} = 0 \qquad (15)$$

where A and B are the alpha and beta speeds of the aircraft. We would like to replicate the expansion of ψ in the normal direction that was used in the past two examples, but the form needed to accomplish this is not immediately clear. To solve this problem, we introduce a linear Cartesian approximation at every grid point.

$$\begin{aligned} x &= r\cos(\beta)(\alpha - \alpha_0) \qquad (16) \\ y &= r(\beta - \beta_0) \qquad (17) \end{aligned}$$

A Jacobian matrix can be found that allows transfer of information into and out of Cartesian space.

$$J = \begin{bmatrix} \frac{\partial x}{\partial \alpha} & \frac{\partial x}{\partial \beta} \\ \frac{\partial y}{\partial \alpha} & \frac{\partial y}{\partial \beta} \end{bmatrix} = \begin{bmatrix} r\cos(\beta) & -r\sin(\beta)(\alpha - \alpha_0) \\ 0 & r \end{bmatrix} \qquad (18)$$

Since the linear approximation is always centered on the grid point of interest the upper right term can be ignored, leaving

$$J = \begin{bmatrix} r\cos(\beta) & 0 \\ 0 & r \end{bmatrix} \quad (19)$$

The spatial derivatives can be transformed through the transpose of the Jacobian.

$$\begin{bmatrix} \psi_\alpha \\ \psi_\beta \end{bmatrix} = \begin{bmatrix} r\cos(\beta) & 0 \\ 0 & r \end{bmatrix} \begin{bmatrix} \psi_x \\ \psi_y \end{bmatrix} \quad (20)$$

$$\begin{bmatrix} \psi_x \\ \psi_y \end{bmatrix} = \begin{bmatrix} \frac{1}{r\cos(\beta)} & 0 \\ 0 & \frac{1}{r} \end{bmatrix} \begin{bmatrix} \psi_\alpha \\ \psi_\beta \end{bmatrix} \quad (21)$$

The transformed spatial derivatives can be used to generate u and v values that ensure normal propagation of the ψ function. The u and v values can then be transformed back into spherical space through the inverse Jacobian, yielding correct A and B values. If there are advection terms, they can be introduced in Cartesian space or spherical space. Because level set methods work by solving linear approximations to problems at very small length and time scales, the local linear transformation is a natural choice.

The second challenge with this problem is the simulation of moving objects in the environment. Level set methods can easily handle this by defining another function to describe the boundary of the ash cloud. For this problem the cloud will be described by the χ function.

$$\frac{\partial \chi}{\partial t} + C\frac{\partial \chi}{\partial \alpha} + D\frac{\partial \chi}{\partial \beta} = 0 \quad (22)$$

The governing PDE can then be configured to evolve the boundary of the ash cloud according to any given flow field. When calculating the evolution of the ψ function for the fighter and the tanker, the value of χ may be checked to determine whether or not the aircraft should be allowed to move at that point. For example the equation for u would be

$$u = \begin{cases} \frac{\psi_x}{\sqrt{\psi_x^2 + \psi_y^2}} & \chi > 0 \\ 0 & \chi <= 0 \end{cases} \quad (23)$$

This provides a simple method for dealing with objects in the environment. This method is not restricted to clouds and other advecting objects. Coquerelle[3] et al show that very complicated environments, including colliding solid bodies, may be simulated with level set methods.

5.3 Results

As a first step the accuracy of this new method in approximating a non-rectangular grid must be verified. To do this, the system is made to simulate a system that is simple enough to be verified by hand. This is the case of normal propagation. The level sets of ψ should proceed away from the initial conditions in a circular fashion when plotted on a sphere. The propagation speed can be verified by measuring the location of the boundary at a fixed time. Figure 14 shows the circular simulation used to validate this approach.

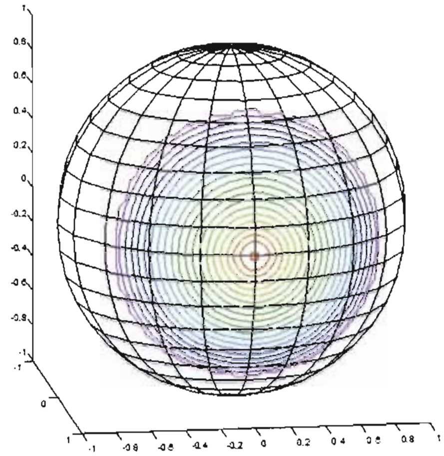

Figure 14: The ψ function correctly propagates in a radial direction

The following figures show examples of different initial conditions and different storm advection velocities. Here again the min-max technique can be applied to find the shortest time rendezvous point. This step is identical to previous applications of the min max principle and is omitted here for brevity.

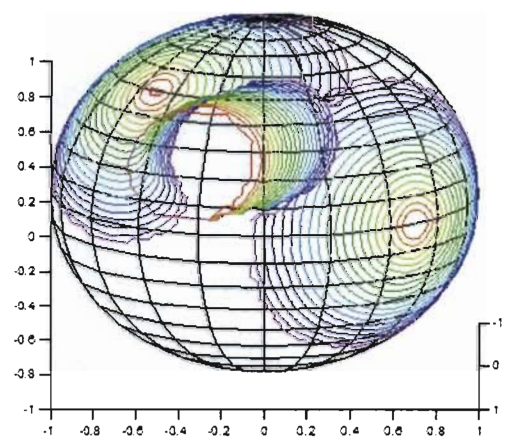

Figure 15: Here the optimal rendezvous in front of the cloud

6 Discussion

Level set methods represent an important class of powerful path finding methods. The global data provided by these

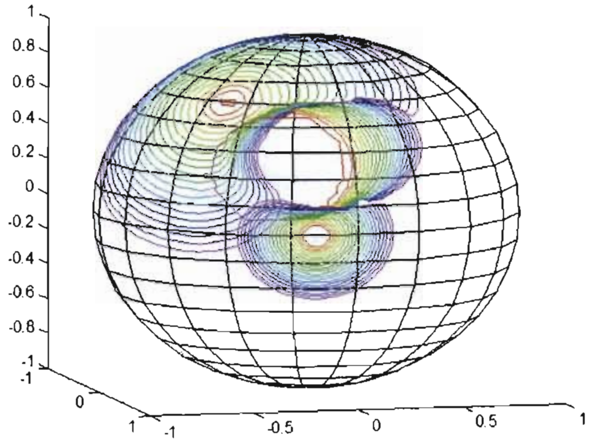

Figure 16: Here the rendezvous point lies in the wake of the cloud

methods make it relatively easy to apply post processing techniques to solve other problems with low computational cost. This paper mainly focused on the rendezvous problem but the underlying level set methods are flexible and general enough to do many other path finding tasks. To show some of this flexibility, 3 example problems were introduced. The first problem showed the ability of level set methods to deal with complex terrain. The ship navigation problem showed the ability of level set methods to deal with the dynamics of the agent. The final problem showed that level sets readily deal with moving obstacles and non-rectangular grids.

References

[1] H. Ando, Y. Oasa, 1. Suzuki, and M. Yamashita. "*Distributed memoryless point convergence algorithm for mobile robots with limited visibility.*" IEEE Transactions on Robotics and Automation, pages 818-828, oct 1999.

[2] Kimmel, R, Arnon, A, Bruckstein, A "*Finding Shortest Paths on Surfaces Using Level Sets Propagation,*" IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 17, No 6, (1995)

[3] Coquerelle, M, Cottet, G-H "*A vortex level set method for the two-way coupling of an incompressible fluid with colliding rigid bodies,*" Journal of Computational Physics, 227, (2008)

[4] S. Hara, H, Shimizu, and T. Kim, "*Consensus in Hierarchical Multi-agent Dynamical Systems with Low-rank Interconnections: Analysis of stability and convergence rates*", Proceedings of the American Control Conference, 2009

[5] Lin, J, Morse, A.S., Anderson, B.D.O "*The Multi-Agent Rendezvous Problem,*" proceedings of the 42nd JEEE Conference on Decision and Control, 1508, (2003)

[6] Hassouna, M, Abdel-Hakim, A, Farag, A. "*Robust Robotic Path Planning Using Level Sets,*" 2005 International Conference on Image Processing, (2005)

[7] S. Osher, R, Fedkiw, "*Level Set Methods and Dynamic Implicit Surfaces*", Springer (2002)

[8] M Sharir, "*Robot motion planning*", Communications on pure and applied mathematics, 1995 vol:48 iss:9-10 pg:1173 -1186