

# Fast Generation and Tracking of GNSS Visibility and Dilution-of-Precision Regions Using Level Set Methods

Nora E.H. Stack, *University of California Los Angeles*  
Walter Z. Cai, *University of California Los Angeles*  
Scott Manifold, *University of California Los Angeles*  
Ana C. Perez-Gea, *University of California Los Angeles*  
Sante R. Scuro, *The Aerospace Corporation*  
Joseph E. Papac, *The Aerospace Corporation*  
Yanina Landa, *The Aerospace Corporation*  
Jaime Y. Cruz, *The Aerospace Corporation*  
Paul D. Massatt, *The Aerospace Corporation*  
Frederick E. Fritzen, *The Aerospace Corporation*

## BIOGRAPHIES

*Nora E.H. Stack* is a recent graduate in Mathematics from St. Mary's College of Maryland. She will be pursuing a PhD in Applied Mathematics at the Colorado School of Mines beginning in the Fall of 2014.

*Walter Z. Cai* is a recent graduate in Mathematics from Cornell University and will be pursuing an M.S. degree in computer science at UW-Madison in the Fall of 2014.

*Scott H. Manifold* is a recent graduate in Mathematics from the University of California, Riverside and will be pursuing a PhD in Engineering Sciences and Applied Mathematics at Northwestern University in the Fall 2014.

*Ana C. Perez-Gea* is a recent graduate in Applied Mathematics from the Mexico Autonomous Institute of Technology in Mexico City. She is currently a research assistant at the Center for Research and Teaching in Economics also in Mexico City.

*Sante R. Scuro* is a Senior Member of the Technical Staff in the Navigation and Geopositioning Systems Department at The Aerospace Corporation. He has an M.S. degree in High Energy Physics from Texas A&M University.

*Joseph E. Papac* is a Senior Member of the Technical Staff in the Structural Dynamics Department at The Aerospace Corporation. He has a doctorate in Mechanical Engineering from UC Santa Barbara.

*Yanina Landa* is a Member of the Technical Staff in the Imagery Products and Exploitation Department at The

Aerospace Corporation. She has a doctorate in Mathematics from UCLA.

*Jaime Y. Cruz* is the Associate Director of the Navigation and Geopositioning Systems Department at The Aerospace Corporation. He has a doctorate in Geodesy from The Ohio State University.

*Paul D. Massatt* is a Senior Engineering Specialist in the Navigation and Geopositioning Systems Department at The Aerospace Corporation. He has a doctorate in Applied Mathematics from Brown University.

*Frederick E. Fritzen* is a Senior Engineering Specialist in the Navigation and Geopositioning Systems Department at The Aerospace Corporation. He has a doctorate in Mathematics from the University of Southern California.

## ABSTRACT

With the advent of multi-Global Navigation Satellite Systems (GNSS), many organizations will need to be able to evaluate GNSS coverage efficiently. Satellite visibility regions and Dilution-of-Precision (DOP) calculations represent a natural application of level set interface representation and tracking methods. Nevertheless, level set analysis has yet to be examined with respect to computational satellite performance simulations. Undergraduate students from the UCLA Research in Industrial Projects for Students (RIPS) program combined their efforts with engineers from The Aerospace Corporation (Aerospace) to develop computationally efficient visibility regions and DOP displays using Level

Set Methods (LSMs); to the benefit of GNSS systems engineering and modeling & simulation efforts.

For this study, the Aerospace RIPS student team developed, implemented, and assessed two methods. The first approach is a static method in which visibility and DOP are calculated from satellite data at discrete time steps. This approach was implemented and optimized in several ways in order to improve accuracy and computational efficiency. A second approach is a dynamic method for visibility. The problem is initialized from satellite data, then the visibility information is evolved forward in time in a level set framework. Two implementations of the dynamic approach were studied, one following a Semi-Lagrangian advection scheme and the other making use of the Essentially Non-Oscillatory (ENO) finite difference discretization scheme; each with merits and drawbacks. Analysis of these methods was conducted and the current verdict is that the static approach is superior to both dynamic approaches in terms of efficiency, grid scaling, iteration scaling, and ease of implementation. In addition to visibility and DOP displays, the static LSM has shown promise in tracking the merging and separation events of visibility zones and three methods were developed to approximate these event times.

## INTRODUCTION

The effort to maximize GNSS coverage includes the evaluation of several performance-based metrics that depend on the geometry of in-view satellites relative to each user. When evaluating coverage, GNSS analysts consider all potential satellite failures because constellation management (and sustainment) efforts must provide high assurance that sudden failures will not cause operational interruptions. GNSS performance studies include high precision positioning, high assurance flight and maritime safety, and navigation in difficult terrain (terrain-masking of satellites). Even with utilization of a cluster of multi-node computers, this effort requires (at its heart) very efficient calculations of geometry-based performance metrics and their derivatives. Current efficient algorithms developed by Aerospace have focused on grid evaluations and are CPU-intensive. New metrics and analysis capabilities need to be developed to address new receivers that use all satellites in-view.

The current methodology to determine the visibility of GNSS satellites is accurate and useful; however, it is also computationally intensive. Similarly, the study of DOP through space and time is a large computational challenge. For example, a  $3^\circ$  grid on the Earth's surface evolved through a ten-minute time interval requires the calculation of approximately 300,000,  $4 \times 4$  matrix inverses. Additionally, one must take into account satellite failures among other tasks, which increase the computational cost by a factor of  $10^{12}$ . Faced with these issues, Massatt and Rudnick [1] derived an

efficient formula for calculating PDOP with four satellites in view.

Level Set Methods (LSMs), introduced by Osher and Sethian in [2], are efficient numerical techniques for the representation and tracking of implicitly defined interfaces, curves, shapes, and surfaces on an Eulerian grid. They have flourished in a variety of fields such as fluid mechanics, materials science, computer graphics and animation, and imaging science. LSMs have the unique ability to automatically handle topology changes such as merging and separation of evolving interfaces. The potential for a dramatic reduction in processing time is especially important in constellation design, optimization, and maintenance applications where repeated evaluations of performance measures are necessary in an iterative process (e.g., the number and variety of failures for a thirty-satellite constellation or in a combined GNSS constellation). It is the objective of this research collaboration (between Aerospace and UCLA's Institute for Pure and Applied Mathematics) to study the application of LSMs to manage space-based geopositioning systems and geolocation technologies. LSMs could prove to be more time efficient, while maintaining the accuracy of systems currently employed by analysts at Aerospace. It is for this reason that the Aerospace RIPS student team implemented LSMs to measure visibility and DOP, and assessed computational benefits.

The methodology of the research (and outline for the report) includes an introduction to the mathematics for determining user position with GNSS systems and DOP. The application of LSMs to the satellite geometry and GNSS performance domains will be described in detail. For this study, the Aerospace RIPS student team successfully developed, implemented, and assessed two methods: a static approach and a dynamic approach. Further efforts have been dedicated towards the optimization of the computational complexity involved with these strategies and their implementation using Matlab. In addition to visibility zones and DOP calculations, the static LSM has shown promise in tracking the merging and separation events of visibility zones and three methods were developed to approximate these event times.

The results of this study will be used to: (1) Determine when and where users can count on sufficient coverage, (2) Optimize the GNSS satellite configuration, (3) Determine how coverage changes when satellites fail, (4) Determine how to best alter the GNSS satellite configuration to best use the reduced number of satellites resulting from a satellite failure, and (5) Determine where new satellites should be placed in the satellite constellation to produce the best possible coverage.

## VISIBILITY REGIONS

Calculations for visibility and DOP incorporate the Cartesian, *Earth Center Fixed* (ECF) frame, and *East North Up* (ENU) frame approaches to three-dimensional positioning around the earth. In this work, the earth is represented by an ellipsoid of revolution of given equatorial radius and flattening. Points on this mathematical surface are specified by their latitude ( $\varphi$ ) and longitude ( $\lambda$ ) coordinates taken from the WGS 84 model.

Visibility determination primarily uses the ENU frame. In order for a satellite to be considered visible from a position on the surface of the earth it must share a direct line of sight with that location. Furthermore, the satellite's elevation must exceed a  $5^\circ$  *mask angle*; a constant chosen in order to avoid common obstructions from trees, buildings, and general topographical variation.

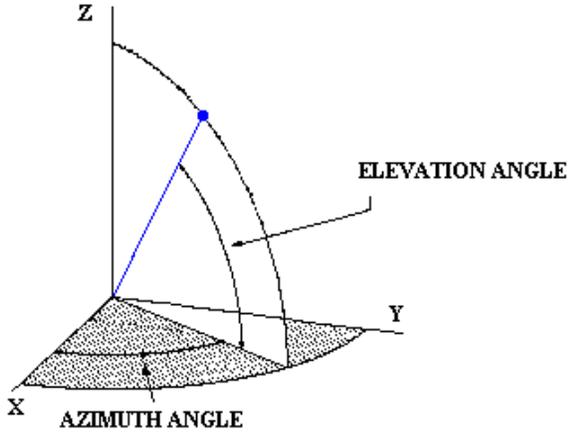


Figure 1: Elevation Angle, as defined from surface tangent plane up towards the satellite in the plane spanned by the zenith and the vector between surface location and satellite. Image sourced from [3].

A user's position is determined by measuring the time it takes for a signal to travel between the user and multiple satellites and from this deriving relative distances. Hence, one satellite will generate a sphere of possible locations. A pair of satellites will generate the intersection of two spheres; a circle. Three satellites will further narrow down the number of possibilities to only 2 points, and a fourth satellite will narrow down the position to a single location. With 31 operational satellites in the constellation, at any given time there will be at least 6 satellites present overhead. Because of this fact, for testing purposes, we utilized a reduced 24 satellite constellation in order to more easily identify errors and areas of interest on our plots.

## DILUTION OF PRECISION

Even with multiple visible satellites overhead, GNSS user error can still be appreciable. To manage error potential, we construct Dilution of Precision (DOP) values. These are metrics that relate the geometry of visible satellites to the likelihood and magnitude of user error.

We define DOP with the following formulations: Let index  $i$  denote the  $i^{\text{th}}$  satellite. Define vectors  $s_i$ , from the center of the earth to the satellite; and  $u$ , from the center of the earth to user position on the earth's surface, as illustrated in Fig. 2.

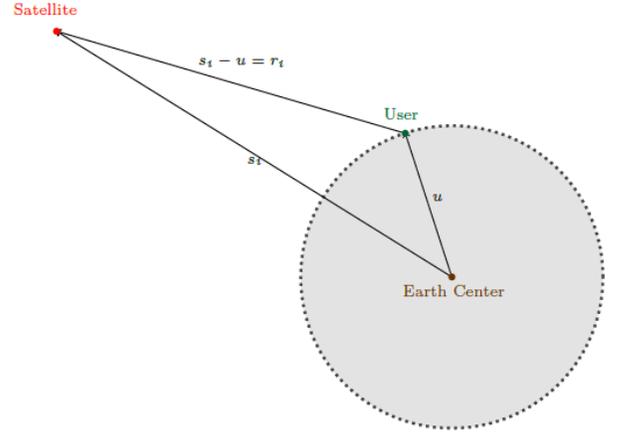


Figure 2: Pseudoranges.

Define  $r_i = s_i - u$  as the vector from the user to the satellite. We may define the observed and model *pseudoranges*, an approximate distance measurement between the satellite and user with the following equations. We use the term pseudorange in this case since  $r_i$  is a measurement from a clock-biased range.

$$\begin{cases} \rho_{i_{obs}} = |r_{i_{obs}}| + c[\Delta T - \Delta t] + [\Delta_{Ion} + \Delta_{Tropo} + \Delta_{\epsilon_i}] \\ \quad = |r_{i_{obs}}| + c[\Delta T - \Delta t] + [\Delta_{\epsilon_i}] \\ \rho_{i_{mod}} = |r_{i_{mod}}|. \end{cases}$$

In the above expressions,  $\rho_i$  denotes the  $i^{\text{th}}$  satellite's pseudorange measurement;  $r_i$  the geometric range between the satellite and user. The values  $\Delta T$  and  $\Delta t$  define the offsets of the user and satellite clocks respectively from GNSS time, while  $c$  is the velocity of the signal in a vacuum. Terms  $\Delta_{Ion}$ ,  $\Delta_{Tropo}$ ,  $\Delta_{\epsilon_i}$  represent distortion from signals propagating through the Ionosphere, Troposphere, and unmodeled noise, respectively. These three terms are compacted into the single term  $\Delta_{\epsilon_i}$ .

The pseudorange can be linearized by expanding  $\rho_{i_{mod}}$  in a Taylor's series expansion around an approximate user position  $\hat{u}$ , then ignoring higher order terms.

$$\left\{ \begin{array}{l} \hat{r} = s_i - \hat{u} \\ \Delta u = u - \hat{u} \\ a_i = \frac{\hat{r}}{|\hat{r}_i|} \\ \Delta \rho_i = \rho_{i_{obs}} - \rho_{i_{mod}} \approx a_i \cdot \Delta u + c\Delta T + \Delta \xi_i \end{array} \right.$$

where  $\Delta \xi_i$  also includes  $\Delta e_i$ . These equations may be equivalently written in matrix form:

$$\begin{bmatrix} \Delta \rho_1 \\ \Delta \rho_2 \\ \Delta \rho_3 \\ \vdots \\ \Delta \rho_n \end{bmatrix} = \begin{bmatrix} a_{x_1} & a_{y_1} & a_{z_1} & 1 \\ a_{x_2} & a_{y_2} & a_{z_2} & 1 \\ a_{x_3} & a_{y_3} & a_{z_3} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ a_{x_n} & a_{y_n} & a_{z_n} & 1 \end{bmatrix} \begin{bmatrix} \Delta u_x \\ \Delta u_y \\ \Delta u_z \\ c\Delta T \end{bmatrix} + \begin{bmatrix} \Delta \xi_1 \\ \Delta \xi_2 \\ \Delta \xi_3 \\ \vdots \\ \Delta \xi_n \end{bmatrix}$$

$$\Delta \rho = H\Delta u + \Delta \xi.$$

We may solve for the above equation computationally using least squares to minimize  $|\Delta \xi|^2$ . Let  $\Delta u^*$  denote the solution while  $\tilde{H}$  is the pseudo-inverse matrix:

$$\Delta u^* = (H^T H)^{-1} H \Delta \rho = \tilde{H} \Delta \rho.$$

Assuming pseudorange covariance is symmetric over rotations, we may define user covariance as follows:

$$\begin{aligned} cov(\Delta u) &= E[\Delta u^* \Delta u^{*T}] \\ &= E[\tilde{H}(\Delta \rho \Delta \rho) \tilde{H}^T] \\ &= \tilde{H} cov(\Delta \rho) \tilde{H}^T \\ &= \sigma^2 \tilde{H} \tilde{H}^T \\ &= \sigma^2 (H^T H)^{-1} \\ &= \sigma^2 \begin{bmatrix} d_x^2 & d_{xy} & d_{xz} & d_{xt} \\ d_{xy} & d_y^2 & d_{yz} & d_{yt} \\ d_{xz} & d_{yz} & d_z^2 & d_{zt} \\ d_{xt} & d_{yt} & d_{zt} & d_t^2 \end{bmatrix}. \end{aligned}$$

Lastly, we may define the various DOP values (Geometric DOP, Positional DOP, Horizontal DOP, Vertical DOP, and Time DOP):

$$\begin{aligned} GDOP &= \text{tr}[(H^T H)^{-1}] = \sqrt{d_x^2 + d_y^2 + d_z^2 + d_t^2} \\ PDOP &= \sqrt{d_x^2 + d_y^2 + d_z^2} \\ HDOP &= \sqrt{d_x^2 + d_y^2} \\ VDOP &= \sqrt{d_z^2} \\ TDOP &= \sqrt{d_t^2}. \end{aligned}$$

Generally, GDOP measurements at or below 1 represent highly accurate readings, whereas high values such as 20 denote extremely poor accuracy with errors as high as 300 meters.

## LEVEL SET METHOD

The Level Set Method is a numerical technique for representing sharp interfaces, curves, shapes, and surfaces as they move over a fixed Eulerian grid. An advantage of LSMs in comparison to explicit interface tracking techniques is that there is no need to parameterize the object(s), thus complex topological changes are handled implicitly. The main drawback of LSMs is that they are not conservative, therefore a loss of mass can occur, particularly with coarse numerical resolutions.

To illustrate the level-set approach, consider the domain presented in Figure 3. We describe  $\Omega$  by the set of points,  $x$ , such that  $\psi(x) < 0$ . Likewise, we describe  $\Omega^+$  by the set of points such that  $\psi(x) > 0$ . The interface  $\Gamma$  is implicitly defined by the zero level set,  $\psi(x) = 0$ . In the context of this work,  $\Omega$  corresponds to the regions where a satellite is visible and  $\Omega^+$  corresponds to the regions where it is not. The evolution of the interface is then given by the evolution of the level set function,  $\psi$ , and obeys the following Hamilton-Jacobi equation,

$$\psi_t + V \cdot \nabla \psi = 0,$$

where  $V$  is an externally generated velocity field. Since the interface is implicitly defined, as the interface evolves in time, merging and/or separation of regions are handled automatically without any special logic or handling. Additionally, geometric quantities such as the normal to the interface and the interface mean curvature can be easily calculated from the level set function. For more details about the level set method, the interested reader is referred to [4].

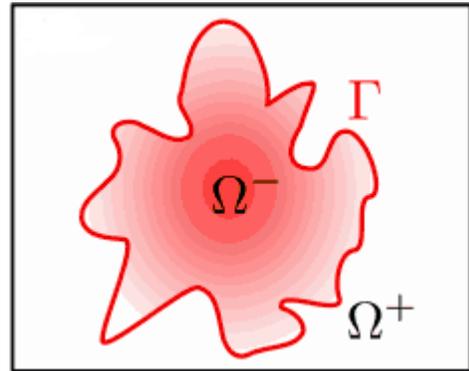


Figure 3: A schematic of the level-set representation of the domain.

## STATIC APPROACH FOR VISIBILITY AND DOP

In general, a level set function,  $\psi$ , is not unique, but rather can be any arbitrary function that is smooth and differentiable and satisfies the interface constraint. The traditional choice in the level set community is to use a signed-distance function. In fact, many algorithms periodically re-initialize the level set function to a signed

distance function in order to maintain good numerical properties. For this work, we have chosen a level set function which is defined as follows:

$$\psi(\varphi, \lambda, time) = MaskAngle - ElevAng(\varphi, \lambda, time),$$

where ‘*ElevAng*’ stands for *Elevation Angle* as defined as the angle from the plane tangent to the earth's surface up towards the vector connecting the surface location and satellite (see Fig. 1). Although this choice of  $\psi$  is not a signed distance function, it is very convenient for the application at hand because it is not overly steep, it is smooth and differentiable, and it provides a measure of the distance of any particular grid node to the interface which we can utilize in adaptive grid meshing strategies. In this model, visibility is considered binary; a satellite is visible if and only if  $\psi < 0$ .

The static approach for visibility regions uses discrete time steps to track the interface as it moves, as opposed to the dynamic approach which utilizes information from previous time steps. For the static approach, we construct a two-dimensional uniform grid to represent the surface of the earth. To be more precise, we implement an *Equirectangular Projection* defined as

$$\begin{aligned} x &= \lambda \cos \phi_1 \\ y &= \phi \end{aligned}$$

where  $\lambda$  is longitude,  $\phi$  is latitude,  $\phi_1$  are the standard parallels (north and south of the equator) where the scale of the projection is true,  $x$  is the horizontal position along the map, and  $y$  is the vertical position along the map. Using this mapping, we get the most direct visualization of our problem over the earth, though we see distortion along the north and south poles in terms of area and curvature. Once we have this uniform grid, we use our level set formulation to calculate level set functions for each satellite in orbit. Operations on the individual level set functions allow us to visualize regions of interest. For example, consider the visible regions of from two satellites,  $\psi_A$  and  $\psi_B$ . The boundary of the region visible by both satellites can be calculated from the zero isocontour of  $\psi_{AB} = \min(\psi_A, \psi_B)$ . Figure 4 is an illustration of the number of satellites that are visible at time  $t=14400s$ , considering only the 24 satellite constellation. The horizontal axis represents the latitude and the vertical is the longitude, which encompasses the entire globe in the Cartesian coordinate system.

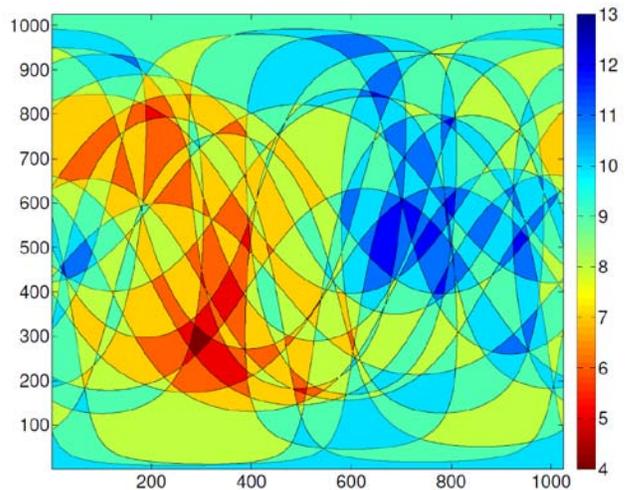


Figure 4: Visibility regions for a 24 satellite constellation at time  $t = 14400s$ .

## PERFORMANCE OPTIMIZATION

Uniform grids, while convenient to implement, can be computationally expensive at high resolutions. For this reason, we have investigated several techniques to improve efficiency. The Narrow Band Level Set Method [5] and the use of nonuniform, adaptive grids such as quadtree grids [6, 7] are two such approaches which we have investigated. Matlab's optimal performance involves pre-allocating matrices and using matrix operations instead of loops. Thus when implementing optimization methods, we had to take into account Matlab's structure.

The Narrow Band Method is a technique in which a narrow band is placed around the interface and only the points of the mesh inside that band are stored so that the computation is restricted locally near the interface. This approach has been shown to reduce computation from an initial operation count of  $O(N^2)$  to a count of  $O(kN)$ , where  $N$  is the size of each grid dimension and  $k$  is the width of the narrow band. We used the idea of the Narrow Band Method to construct our Pseudo-Narrow Band Method, taking into consideration Matlab's optimization. The algorithm that we implement contains two different mesh matrices. The first matrix,  $A$ , stores the initial values of the broad grid. Once the regions that have a change of sign in the level set function are detected, the points for the narrow band are selected.  $B$  will then be the matrix of a much finer mesh grid that will store the values of  $\psi$  inside the narrow band.

Figure 5 illustrates the implementation of the algorithm. We start by having a broad uniform grid of size  $n \times n$  which is the map at different latitudes and longitudes. The green dots,  $g_i$  with  $i = 1, 2, 3, 4$ , represent points within  $\mathcal{Q}$  near the interface. The red stars are the neighboring points above, below, and to both sides of the green. If  $r_j$  are these red points, since  $\psi(r_j) > 0$ , then there must be points of the interface in between the green and

red dots. The blue dots in the figure represent grid nodes on the resulting fine mesh. There is a trade-off between accuracy and computation time in the selection of the width of the narrow band. The problem that may be encountered is that if the geometry of the interface involves sharp corners and the width of the band is too narrow, then these corners may not be captured by the algorithm. The final step is defining the fine grid. Each cell has a side length of  $1/n$ , and thus they can be broken into  $m^2$  different cells of side length  $1/mn$ . Figure 5 shows the example of  $m=2$ , which makes the resolution of the mesh have four times as much information. This new mesh will have a resolution of  $nm \times nm$ , but will have made fewer calculations compared to a uniform mesh map of the same resolution.

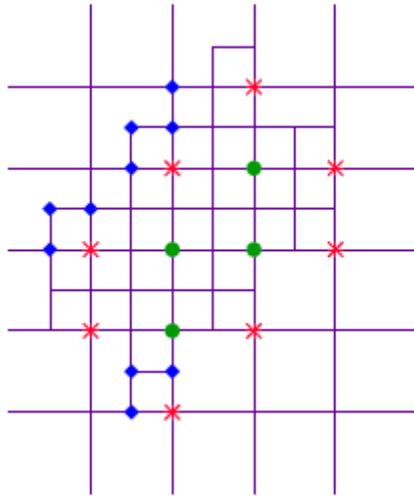


Figure 5: A schematic of the Pseudo Narrow Band algorithm.

The quadtree approach [6, 7] is a tree-based data structure in which the entire domain is encompassed within a root cell. Cells near the interface are recursively divided into child cells until a specified minimum cell size is reached. This has the effect of concentrating the grid nodes near the interface. This approach requires a more complex implementation. Two objects will be required for the quadtree construction, a *cell* and a *node*. A *node* is simply a point on the grid with the latitude and longitude coordinates as well as its interface value,  $N_i = (\theta, \lambda, \psi)$ . A *cell* has the form of a square in the grid and will be defined as an object with two properties: the length of the side of the cell (referred to as the *size*) and the four nodes that it has at each corner. Figure 6 demonstrates the process of cell creation. A cell will split if one of its vertices meets the following Whitney decomposition criteria [7], which is a measure of distance to the interface:

$$\min_{v \in \text{nodes}(C)} |\psi(v)| \leq L \cdot \text{diag}(C),$$

where  $L$  is a Lipschitz constant taken to be approximately one. The splitting will proceed until it gets to a certain level where the cells' size reaches a tolerance parameter.

Figure 7 illustrates the evaluated points of a quadtree grid for GDOP values over a domain that ranges from 0 to 180 for the latitude and -90 to 90 for the longitude. The color of the points in Figure 7 are different levels of GDOP that range from 1.419 to 10. The actual maximum of GDOP is around 20 for a few points, but for display reasons they were set to a GDOP of 10. We can observe that as we get closer to the interface, the amount of points greatly increases.

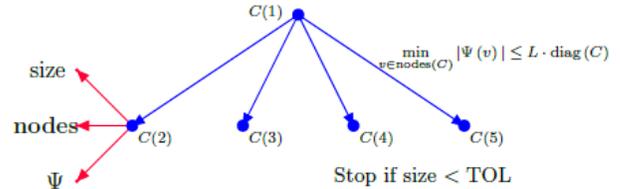


Figure 6: Quadtree data structure and cell design.

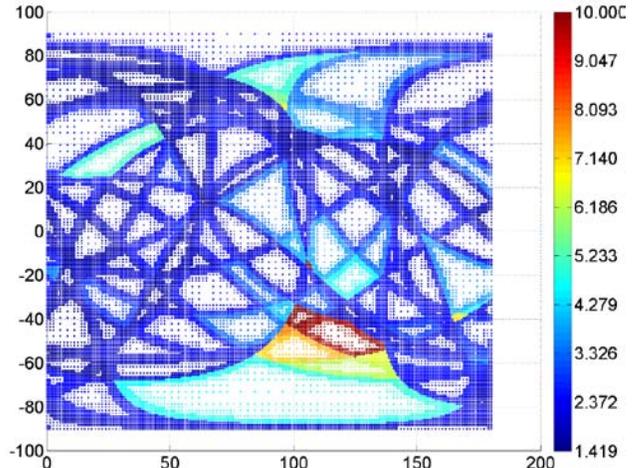


Figure 7: GDOP plot displaying the evaluated nodes of a quadtree grid.

## DYNAMIC APPROACH FOR VISIBILITY REGIONS

In the dynamic approach, the level set function,  $\psi(x, t)$ , is initialized at time  $t=0$  and evolved forward in time by means of solving the Hamilton-Jacobi level set equation,

$$\begin{cases} \psi_t + V(x, t) \cdot \nabla \psi = 0 \\ \psi(x, 0) = \psi_0. \end{cases}$$

The velocity field,  $V$ , is unique to the physics of the problem, and in our case is determined by the motion of the satellite. Since this equation cannot be explicitly solved, we approximate the solution using finite difference schemes.

As we evolve  $\psi$  over time according to our problem parameters we observe that the level set function

may develop steep gradients or other undesirable properties. Rather than the traditional approach of re-initializing the level set function to a signed distance function [8], which involves solving a computationally expensive nonlinear PDE, we instead call our static level set algorithm to reinitialize  $\psi$ . One thing to note is that a three-dimensional description of our problem is necessary in the dynamic approach in order to relate satellite motion to the interface velocity field, whereas we were able to reduce the static level set approach to a two-dimensional map projection of the earth's surface.

In this work, we utilize two approaches to solving the level set equation. The first approach is a finite difference discretization that uses an implicit Euler time discretization along with the third-order accurate Essentially Non-Oscillatory (ENO) [9] spatial discretization scheme. The advantage of this approach is that implementation is straightforward and the numerical accuracy is very good. The disadvantages of this approach are that the time step is limited by a CFL condition for numerical stability and the ENO algorithm requires evenly-spaced grid nodes so we are limited to a uniform grid.

The second implementation uses a Semi-Lagrangian approach [10] for the level set advection equation. This approach is unconditionally stable, so the time step is only constrained by accuracy requirements. Additionally, this approach is well-suited for nonuniform grids such as a quadtree/octree or narrow-band approaches. The primary disadvantage of the Semi-Lagrangian scheme is that it has a propensity for numerical diffusion.

## RAYCASTING AND DYNAMIC HORIZONS

While the dynamics of the satellites and the earth are well-studied, it is a more challenging task to track the dynamics of our visibility interfaces as they are projected onto the earth. In determining the velocity field which is responsible for advecting the level set function, we draw heavily upon the ideas used in Tsai, et al. [11] and their raycasting approach to visibility.

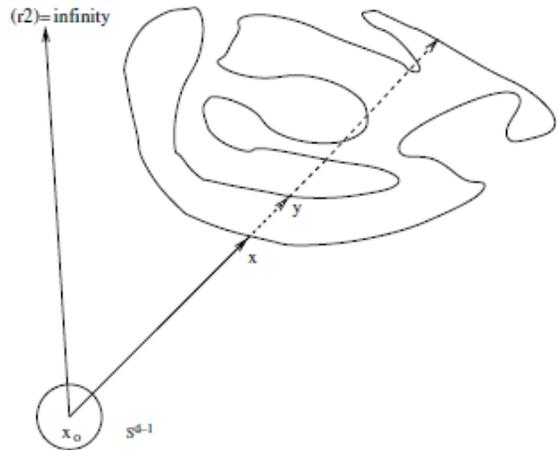


Figure 8: A diagram displaying the basis behind the raycasting technique. We see that since  $y$  comes after  $x$  it is considered blocked. Figure is from Tsai et al. [11].

We begin with some point  $x_0$  in our domain which we consider a source or observer, which in all directions casts out visibility rays. Also embedded in our domain are a series of obstacles or occluders, which will be represented by an occluding level set function  $\Phi$ . Let us say we are interested in the direction from our source,  $x_0$ , to some point  $x$  in our domain. We define the following

$$r(x, x_0) = \frac{(x - x_0)}{|x - x_0|}.$$

We also adopt the convention in saying that some point  $x_1$  is before  $x_2$  if  $r(x_0, x_1) = r(x_0, x_2)$  and  $|x_0 - x_1| \leq |x_0 - x_2|$ . Furthermore we can make the relation strict by simply replacing  $\leq$  with  $<$ . As an example if we refer to Figure 8, we would then say that  $x$  is before  $y$ . The reason this concept is introduced is because we need to make it clear how these visibility rays interact with the occluding objects in our domain. Referring again to Figure 8 and keeping in mind the sphere around  $x_0$ , define the following

$$\rho(\theta) = \begin{cases} \min_{x \in \mathbb{R}^d} \{|x - x_0| \mid r(x, x_0) = \theta, \Phi(x) \leq 0\} \\ \infty, & \text{if does not exist} \end{cases}$$

Thus our implementation must be the requirement that  $\rho(r(x, x_0)) \leq |x - x_0|$ . For clarity sake, it is helpful to think of  $r(x, x_0)$  as referring to the direction of our rays and  $\rho(\theta)$  referring to the length of these rays. In our particular problem we can very naturally consider  $x_0$  to be a GNSS satellite with the visibility rays being the range of the signal it sends. Our domain would then have one large occluding object, the earth, which blocks these signals being sent by our satellites. Very naturally, due to the size of the earth, outlines of these visibility regions will become apparent on the earth and shall be referred to as the *horizon points*.

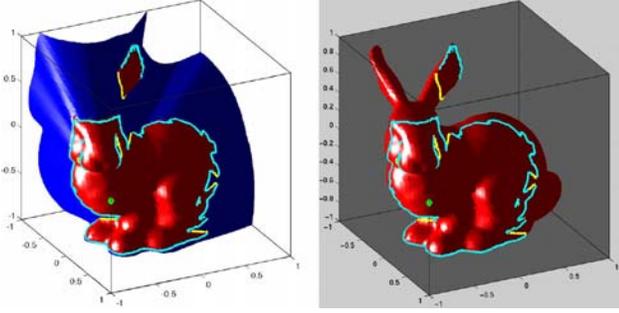


Figure 9: A diagram displaying how visibility zones (outlined by cyan) would be displayed. The green dot represents the satellite and the bunny would be the earth. Image is from Tsai et al. [11].

Mathematically speaking we define the horizon points as the following

$$H = \left\{ x \in \mathbb{R}^3 \mid \Phi(x) = 0, (x - x_0) \cdot \nabla \Phi = -\sin\left(\frac{\pi}{36}\right) \right\}$$

where the  $-\sin(\pi/36)$  term comes from the  $5^\circ$  mask angle we are working with. Since this is a dynamic problem, the velocity field also changes in time. We propose the following candidate for a 3-D vector field we can use to advect our horizon, given a satellite location,  $S(t)$ :

$$\left[ (x - S(t)) D^2 \Phi \right] \left( \frac{dx}{dt} \right)^T = \begin{bmatrix} 0 \\ \frac{d}{dt} S(t) \cdot \nabla \Phi \end{bmatrix}$$

While this is an accurate description, there are difficulties with this approach due to the need to solve a linear system for each point we want to evaluate. In addition to this efficiency problem, there is a free variable each time we solve this system which is difficult to track for each point. The current conjecture for this free variable is that it accounts for movement along the interface as we shift our horizon in time.

While this lays down the necessary groundwork for calculating how our horizon moves with respect to a satellite through time, there is still the initial task of defining the horizon at some initial time. The horizon can be defined as the intersection of the edge of the visible regions and the earth. Mathematically we can write this as:

$$H = \{x \in \mathbb{R}^3 \mid \Phi(x) = 0\} \cap \{x \in \mathbb{R}^3 \mid \psi(t, x) = 0\}.$$

Thus we define our 3D level set function to be:

$$\psi(t, x) = |x - S(t)| \sin(5^\circ) + (x - S(t)) \cdot \nabla \Phi(x).$$

When implementing this dynamic approach our goal was to find the most natural coordinate system and method such that we could easily extract the level set information

we were interested in while minimizing the necessary computations. Thus our initial approach was to use a Cartesian grid that was indexed by longitude, latitude, and height off the surface of the earth. The ease in using this pseudo-spherical indexing to calculate information needed, along with the simplicity in converting into Cartesian coordinates where necessary made this approach very enticing. However, one initial challenge with this sort of coordinate indexing is that it results in uneven spacing, meaning we can not use the usual ENO scheme to solve our Hamilton-Jacobi equation. However with a nonuniform grid we can still utilize the Semi-Lagrangian approach.

In order to solve for the velocity field in our chosen system, which gives us information how the horizon corresponding to one satellite moves, we define a function  $up(\theta, \lambda, h)$ , which corresponds to

$$\nabla \Phi|_{\theta, \lambda, h} \approx \frac{\nabla \Phi}{|\nabla \Phi|} \Big|_{(\theta, \lambda, h)} = up(\theta, \lambda, h)$$

where  $\Phi$  is a level set function that defines the earth's surface. Using this information along with first order differencing to approximate  $S'(t)$ , we can easily get the information for  $x$ ,  $S(t)$ ,  $S'(t)$ ,  $grad(\Phi)$ . However when dealing with the Hessian,  $D^2\Phi$ , we need to take the following derivatives  $up_x$ ,  $up_y$ ,  $up_z$  which is where difficulty arises. Since our coordinates are indexed in longitude, latitude, and distance from the surface of the earth, implementing first order differencing with respect to Cartesian coordinates poses a problem. The current workaround this issue is to use the  $up$  information calculated over our pseudo-spherical grid and interpolate it to get the desired information when shifting in  $x$ ,  $y$ , and  $z$  directions. Once all this information is calculated, Matlab's `linsolve` function is used to solve the linear system at each point over our grid, giving us  $u$ ,  $v$ ,  $w$  the corresponding  $x$ ,  $y$ ,  $z$  components of our vector field.

Once we have the vector field for one satellite, we implement the Semi-Lagrangian algorithm as discussed in [7]. As the simulation is run, we periodically reinitialize the level set functions with the static method in order to preventing numerical noise and distortion.

Overall, the main advantages of this Semi-Lagrangian approach are the stability in time that lets us choose the time step, the ease in calculating  $grad(\Phi)$ , and how naturally our grid corresponds to the surface of the earth. After solving for our new shifted level set function,  $\psi$ , we want to isolate the information corresponding to the surface of the earth. Since our grid is constructed such that it contains isocontours of the earth level set function  $\Phi$ , then we can easily pick out the contour of the earth's surface. One large drawback to this method is that we need a 3 dimensional band around the earth's surface to ensure our vector field remains intact, requiring a minimum of 3 times the calculation for a similar 2 dimensional calculation. For example, with the static

method in order to get a 120x120 plot of the earth's surface,  $120^2 = 14400$  grid points are required, while this comparable Semi-Lagrangian method would require  $(120^2)(3) = 43200$  points of computation. Another large drawback to this Semi-Lagrangian method is the number of interpolations needed in the algorithm along with calculating the Hessian, as these 3 dimensional interpolations scale very poorly with grid size. So while for low grid sizes, the Semi-Lagrangian method seems enticing due to its natural benefits, there are many drawbacks in computational time when it comes to larger and more accurate calculations.

To avoid the drawbacks in grid size scaling resulting from excessive interpolation, we choose a uniform Cartesian grid that lets us easily implement an Essential Nonoscillatory Polynomial Interpolation of data (ENO) algorithm and shift  $up(\theta, \lambda, h)$  in the  $x, y, z$  directions. These two changes allow us to forgo the multiple interpolations needed. In addition, it lets our dynamic implementation become much faster and efficient at larger grid sizes, which is needed to reach the threshold levels of accuracy in a feasible time. There are some drawbacks to this approach, in that calculating the gradient,  $grad(\Phi) = up(\theta, \lambda, h)$ , at each  $(x, y, z)$  coordinate requires utilizing Matlab's *fsolve* function in its *Optimization toolbox* to get

$$pinverse: [-90,90) \times [0,360) \times \mathbb{R} \rightarrow \mathbb{R}^3 \\ (\theta, \lambda, h) \mapsto (x, y, z).$$

While we only calculate *pinverse* once over point set representation, this inverse operation can add a considerable amount of time as we scale up our grid size. Another additional problem is that the grid configuration does not directly provide information about visibility on the earth's surface. Thus if want to use our data to get the corresponding information on the surface of the earth, we run a tricubic interpolation, which is another costly operation.

Since the implementation uses third-order ENO, or ENO3, we see much less distortion versus Semi-Lagrangian approach, which allows us to reinitialize with the static method less frequently. Together with the Narrow Band construction to save on evaluation time, we see improvements of this ENO approach. The main deterrent to the Dynamic ENO method is utilizing the 3-D information we generate and translating it to the surface of the earth. We can do a cubic interpolation of the earth's surface, but this drastically increases computation time. Also, due to the CFL condition when working with this ENO approach, the time step restriction can be severe, especially as the grid resolution is increased.

## GRID SCALING AND TIMING RESULTS

Table 1 contains computation time and grid size scaling results for a 3-D Static and both developed Dynamic

approaches for calculating visibility zones. We see that the Static approach has the best performance and scaling with grid size. Between the two Dynamic methods, the Dynamic ENO approach has better performance than the Semi-Lagrangian. Also, as we would expect, the use of the Narrow Band for the Static Method substantially improves computation time over using the full grid.

Grid Size	Static (Narrow Band)	Static	Dynamic ENO (Narrow Band)	Dynamic Semi-Lagrangian
125	1.93s	4.06s	5.64s	9.08s
1000	13.10s	10.05s	24.11s	50.22s
8000	89.22s	136.9s	169.8s	416.7s
27000	273.5s	472.6s	579.0s	1722.13s

Table 1: Grid Size Scaling.

However of practical interest to us is not 3-D Visibility information, but the 2-D Visibility on the earth's surface. Thus we are interested in the time and accuracy of generating and interpolating our 3-D information into 2-D information. Figure 10 is a plot of the computation time versus grid resolution for the Static 2D approach and the Dynamic ENO approach including interpolation onto the surface of the earth. As can be expected, the Static 2D approach is more computationally efficient, particularly at high resolutions.

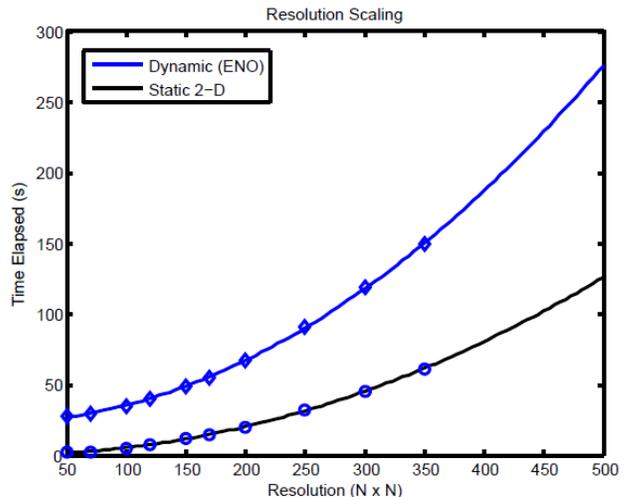


Figure 10: Resolution scaling for the Dynamic ENO approach and the 2D Static approach for Visibility Zones.

## DYNAMIC DILUTION OF PRECISION

Due to the nature of Dilution of Precision values, the dynamic approach presents very difficult problems as a simulation method. For visibility, it is possible to track visible region boundaries for each satellite with an independent vector field operating on every interface and then overlaying the graphs using a sum value for each time step. In contrast, DOP values require simultaneous input from every satellite in the constellation. Hence, each

satellite would need to be tracked concurrently in order to devise a feasible vector field.

Moreover, unlike visibility, DOP yields a continuum of possible values and hence there do not exist precise interfaces to track, making level set representation cumbersome. The difficulty in locating interfaces also limits the feasibility of employing optimization techniques on adaptive grids since every grid point will display nontrivial changes in value at all times. These issues are further complicated by discontinuous jumps in DOP that occur at the boundaries of satellite visibility regions. Therefore, in order to fully encode an accurate approximation for the transition of DOP values through time, one would have to derive a velocity field that correlates the motion of the satellites as well as their orientation with respect to each other in space and time.

Due to the difficulty in constructing a suitable vector field, we have not developed a true dynamic DOP implementation. However, we can achieve a semi-dynamic DOP. Since calculations of DOP values heavily rely on visibility calculations, we can evolve the visibility calculations with the dynamic approach then statically calculate the Dilution of Precision values.

## MERGING OF VISIBILITY REGIONS

Knowing when two visibility regions merge or separate, or rather simply when the regions begin or end overlapping periods, is vital since it can mark the exact time when positioning calculations may begin operation or fail for a given location on earth's surface. The static level set approach may be used to estimate these event times.

Consider arbitrary satellites  $i, j$  and their associated level set functions, visibility regions, and interfaces;  $\psi_i, \psi_j, \Omega_i^-, \Omega_j^-, \Gamma_i, \Gamma_j$ , respectively, where  $\psi_i, \psi_j$  are the level set functions employed in the static method.

Taking an area tracking approach, we define the function  $\psi_{max} = \max\{\psi_i, \psi_j\}$  which will have negative values only in regions where both satellites are visible. Our purpose is to detect the area of the intersection region, or simply, the number of grid points in  $\Omega_i \cap \Omega_j$ .

We may detect mergers by beginning at a particular time step where the number of grid elements in  $\Omega_i \cap \Omega_j$  is zero and iteratively checking the area on a coarse time step. When the value becomes positive, we backtrack to evaluate from the last zero-valued time and then re-evaluate on a finer grid and time step. Separation may be detected analogously by investigating when the number of points in  $\Omega_i \cap \Omega_j$  switches from positive to zero. Note that this method provides an upper bound on merger times and a lower bound on separation times.

A second method is similar to the area tracking approach except that it uses narrow bands built outward from  $\Gamma_i, \Gamma_j$ . In merger detection, while the narrow bands remain disjoint the interfaces do not intersect. Furthermore, the narrow bands must intersect before the

interfaces come into contact with each other. The associated opposite process will occur in visibility region separation. A variation of this technique, which seems to improve accuracy, is to build narrow bands for the sum,  $\psi_{sum} = \psi_i + \psi_j$ , and difference,  $\psi_{diff} = \psi_i - \psi_j$ , level set functions. The sum and difference functions will approach each other at higher rates than  $\psi_i, \psi_j$  and will begin with more cushion between their respective interfaces, forcing the narrow bands to remain disjoint for a longer time before the interfaces actually first come in contact, see Figure 11.

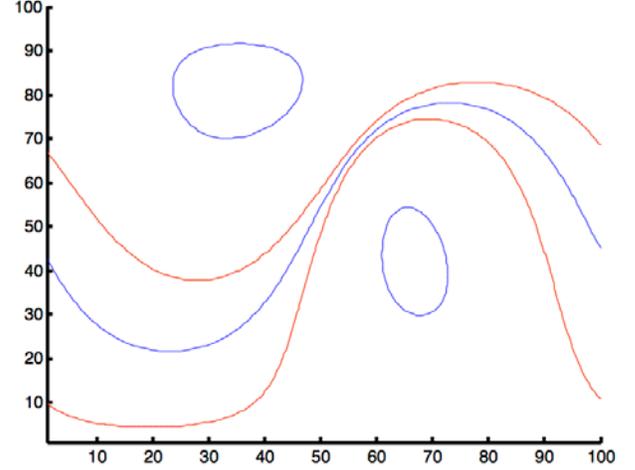


Figure 11: We see that  $\Gamma_i, \Gamma_j$  (red) are very close while  $\Gamma_{sum}, \Gamma_{diff}$  (blue) remain far apart. This helps prevent the Narrow Bands from intersecting before a true merge happens.

We may note that this process will necessarily provide a lower bound on merger times and an upper bound on separation times. Thus, the Narrow Band method and Area Tracking may be used in tandem to supply an accurate and refineable range for merger and separation times.

A third method detects the merger and separation of visibility regions by investigating the boundary of the level set function defined by  $\psi_{sum} = \psi_i + \psi_j$ . While visibility regions  $\Omega_i^-, \Omega_j^-$  are disjoint,  $\psi_{sum}$  yields a pair of negative valued regions that are necessarily subsets of  $\Omega_i^-$  and  $\Omega_j^-$ .

As  $\Omega_i^-$  and  $\Omega_j^-$  draw closer through time, the negative regions of  $\psi_{sum}$  begin expanding towards the future intersection point on the  $\psi_i, \psi_j$  interfaces;  $\Gamma_i, \Gamma_j$ . At the moment of intersection, the two disjoint negative regions of  $\psi_{sum}$  will also first meet at the same point as  $\Omega_i \cap \Omega_j$ . They cannot meet earlier since this would imply there exists an element outside of  $\Omega_i^-, \Omega_j^-$  where  $\psi_{sum}$  takes a negative value. Moreover, for all  $x \in \Omega_i^- \cap \Omega_j^-$ , we must necessarily have  $\psi_{sum}(x) < 0$ .

There is an observed topological change to the negative region at the time of merger for the  $\psi_{sum}$  function. This topological change can be detected using

the *contourc* function in Matlab. *Contourc* returns the boundaries of negative regions in the form of lists, each occupying one cell of an array. Hence, each cell represents an independent connected component of the  $\psi_{sum}$  negative region. When the number of cell drops from 2 to 1, there has been a merger of two previously disjoint negative regions of  $\psi_{sum}$  implying  $\Omega_i^-$  and  $\Omega_j^-$  have come into contact. Similarly, separation of intersecting visibility regions is detected by increases in the number of cells. An illustration of the underlying topological change may be found in Figure 12.

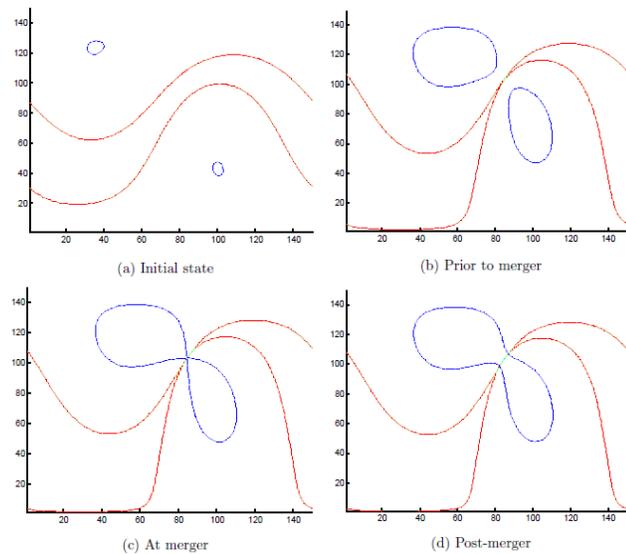


Figure 12: Diagrams of the  $\psi_{sum}$  function (in blue) close to merger time.

While the methods have been addressed and conceived with pairwise satellite configurations in mind, the aforementioned methods can easily be extended to any size satellite constellation. We simply define a symmetric matrix, with each entry  $(i,j)$  corresponding to the relevant information about satellites  $i,j$ . This matrix can be generated alongside with the visibility data with little extra computation, giving us information about the constellation at each point in time.

## CONCLUSIONS

This paper is a first work in applying state of the art Level Set techniques to the evaluation and visualization of multi-Global Navigation Satellite Systems coverage. A static level set approach to visualization and Dilution of Precision was implemented and evaluated along with several optimization techniques to reduce computational expense. Additionally, two implementations of a dynamic level set method for satellite visibility were implemented, one utilizing a Semi-Lagrangian evolution and the other making use of an ENO finite difference method. Analysis of these methods suggested that the static approach is superior to both dynamic approaches in terms of

efficiency, grid scaling, iteration scaling, and ease of implementation. A dynamic approach for calculating DOP was investigated and does not seem to be a strong candidate in terms of ease of implementation and computational efficiency. In addition to displays of visibility zones and DOP values, the static level set approach has shown promise as a tool for tracking the locations and times of merging and separation events of visibility zones.

## ACKNOWLEDGMENTS

The authors would like to thank the staff of the Institute of Pure and Applied Mathematics (IPAM) at UCLA for hosting the students for the duration of the RIPS program, James Gidney from the Aerospace Corporation for all he has done in support of this project, and NSF Grant #0931852 for providing funding.

## REFERENCES

1. Massatt, P.D. and Rudnick, K.M., Geometric formulas for dilution of precision calculations, *Navigation*, 37(4):379-391, 1990.
2. Osher, S. and Sethian, J. A., Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations, *Journal of Computational Physics*, 79(1):12-49, 1988.
3. Basic antenna theory. <http://www.ycars.org/EFRA/Module/%20C/AntBasics.htm>, 2013.
4. S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, Springer-Verlag, Paris, 2003.
5. David Adalsteinsson and James Sethian. A fast level set method for propagating interfaces. *Journal of Computational Physics*, 118(2):269–277, 1994.
6. John Strain. Fast tree-based redistancing for level set computations. *Journal of Computational Physics*, 152(2):664–686, 1999.
7. Chohong Min and Frederic Gibou. A second order accurate level set method on non-graded adaptive Cartesian grids. *Journal of Computational Physics*, 225:300–321, July 2007.
8. G. Russo and P. Smereka, *A Remark on Computing Distance Functions*, 2000.
9. Stanley Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics*, 77(2):439-471, August 1988.
10. Andrew Staniforth and Jean Cote. Semi-lagrangian integration schemes for atmospheric models a review. *Monthly Weather Review*, 119(9):2206{2223, 1991.
11. Y.-H.R. Tsai, L.-T. Cheng, S. Osher, P. Burchard, and G. Sapiro. Visibility and its dynamics in a PDE based implicit framework. *Journal of Computational Physics*, 199:206-290, September 2004.