Visibility of Point Clouds and Mapping of Unknown Environments

Yanina Landa[*], Richard Tsai[†], and Li-Tien Cheng[‡]

April 14, 2006

Abstract

We present an algorithm for interpolating the visible portions of a point cloud that are sampled from opaque objects in the environment. Our algorithm projects point clouds onto a sphere centered at the observing locations and performs essentially non-oscillatory (ENO) interpolation to the projected data. Curvatures of the occluding objects can be approximated and used in many ways. We show how this algorithm can be incorporated into novel algorithms for mapping an unknown environment.

1 Visibility

The problem of visibility involves the determination of regions in space visible to a given observer when obstacles to that sight are present. When the observer is replaced by a light source in the simplified geometrical optics setting with perfectly absorbing boundary condition at the obstacles, the problem translates to that of finding illuminated regions. In this regard, the visibility problem is highly related to the high frequency wave propagation problems and is needed in many computational high frequency wave approaches [2]. We will interchange the term visibility with illumination, and occlusion with shadow freely in this paper.

In visualization, visibility information can be used to make complicated rendering processing more efficient by skipping over occlusion. In robotics mission planning, achieving certain visibility objectives may be part of the mission. Video camera surveillance design is one such example.

Visibility problems have also been studied by geometers. For example, H. Wente asked if connectedness of the on surface shadow is sufficient to imply convexity of the occluding surface [4].

In general, one may consider the following classes of visibility problems:

1. Given occluders, construct shadow volume and its boundary;
2. Given a projection of visible regions, construct the occluders;
3. Find location(s) that maximize visibility using certain predefined metric.

[*]Department of Mathematics, University of California, Los Angeles, CA 90095
[†]The University of Texas at Austin, TX 78704
[‡]Department of Mathematics, University of California, San Diego, CA 92093

In many visualization applications, (1) is solved by projecting triangles. Wente's question can be viewed as in category (2). Problems related to surveillance is related to (2). We will present an algorithm for a problem related to both (1), (2), and (3).

1.1 Representations of Visibility

Today computational geometry and combinatorics are the primary tools to solve visibility problems [5][17],[3]. The combinatorial approach is mainly concerned with defining visibility on polygons and more general planar environments with special structure. All the results are based on an underlying assumption of straight lines of sight. The simplified representation of the environment is a major limitation of this methodology. Furthermore, the extension of these algorithms to three dimensional problems may be extremely complicated.

Our goal is to define such a representation of visibility as to be able to solve the problems considered in computational geometry [5] on general environments in two or three dimensions, independent of the integral field defining the lines of sight, utilizing minimum information about the environment.

One attempt was to introduce the level set representation of the occluding objects and the visibility function, defined in [15]. While this algorithm can be applied to general types of environment, easily extended to three dimensions, and curved lines of sight, it requires a priori knowledge of the occluding objects to construct the level set representation of the environment. This information may not be available in some important real life applications, e.g. navigation in an unknown environment, or if the occluding objects are represented by open surfaces.

Another method for visibility representation was introduced by LaValle et al in [6], [13]. This is a rather minimal framework based on detecting discontinuities in depth information (called gaps) and their topological changes in time (referred to as gap critical events). The "visible" environment is represented by a circle centered at the vantage point, with gaps marked on the circumference in the order of their appearance to the observer. Note that no distance or angular information is provided. As with most combinatorial approaches, LaValle's method works only on regions having special geometries.

In [18], an algorithm extracting planar information from point clouds is introduced and used in mapping outdoor environment. In [10], depth to the occluders is estimated by a trinocular stereo vision system and is then combined with a predetermined "potential" function so that a robot can moved to the desired location without crashing into obstacles.

Here we introduce a new model which, similarly to the level set representation, can handle complicated geometries and curved lines of sight. In contrast to LaValle's representation, we utilize distance and angular information, which, in practice, can be easily provided by the sensor.

2 Visibility Interpolation and Dynamics

Assume we have a set of points P that are "uniformly" sampled from the occluding surfaces. In practice this data could be obtained from sensors such as

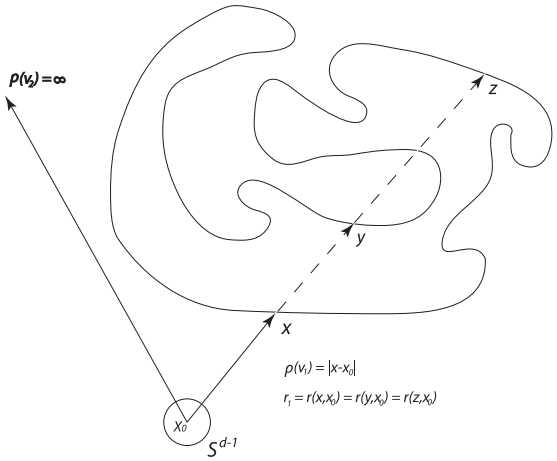

Figure 1: Demonstration of visibility

LIDAR or even from triangulated surfaces (here P would be the set of vertices). Given a vantage point, our algorithm would produce a subset of visible data points and a piecewise polynomial interpolation of the visible portions of the surfaces. Unlike the level set representation [15], our algorithm can handle open surfaces and does not require a priori knowledge of occluding surfaces to construct visibility.

2.1 Basic Formulation

Let us begin by introducing some notations. Let x_0 denote the vantage point (always assume x_0 outside of the objects). Consider $\Omega \subset \mathbb{R}^d$ $(d = 2, 3)$ – a set of objects in question, $\Gamma = \partial\Omega$, and $\Gamma^*_{x_0}$ – visible portion of Γ with respect to x_0. Denote by ϕ the signed distance function to Γ. We define the view direction from x_0 to x by $\nu(x_0, x) := (x - x_0)/|x - x_0|$. For any two points in space x_1 and x_2, we say that $x_1 \leq x_2$ (x_1 is "before" x_2) if $\nu(x_0, x_1) = \nu(x_0, x_2)$ and $|x_1 - x_0| \leq |x_2 - x_0|$. Also, a point $y \in \Gamma$ is called a horizon point if and only if $\nu(x_0, y) \cdot n(y) = 0$, where $n(y)$ is the outer normal of Γ at y. Lastly, a point $y \in \Gamma$ is called a cast horizon point if and only if there is a point y^* such that $y^* \leq y$ and y^* is a horizon point.

Observe that the visibility status of points sharing the same radial direction with respect to the vantage point satisfies a causality condition. That is, if x_1 is occluded and $x_1 \leq x_2$, then x_2 is also occluded. We set

$$\rho_{x_0}(p) := \begin{cases} \min_{x \in \bar{\Omega}} \{|x - x_0| : \nu(x_0, x) = p\}, & \text{if exists} \\ \infty, & \text{otherwise} \end{cases} \tag{1}$$

Define the visibility indicator $\Theta(x, x_0) := \rho(\nu(x, x_0)) - |x - x_0|$ such that $\{\Theta \geq 0\}$ is the set of visible regions and $\{\Theta < 0\}$ is the set of occluded regions. See Fig. 1 for an example.

Assume, in addition, that the sampling of points is "uniform". That is, we can find an $\epsilon > 0$, such that ϵ-balls centered at each sampled point on Γ connect the connected components and do not connect disconnected components of Γ.

Let $P \subset \mathbb{R}^d$ be the sampled data set. Enumerate all the points $y_i \in P$. Define the projection operator $\pi_{x_0} : \mathbb{R}^d \mapsto S^{d-1}$, mapping a point onto the unit sphere centered at x_0. Then we can construct the following piecewise constant approximation to the surface on a sphere:

$$\tilde{\rho}_{x_0}(z) = \min\left(\rho_{x_0}(z), |x_0 - y_i|\right), \text{ for every } z \in \pi_{x_0} B\left(y_i, \epsilon\right). \tag{2}$$

In addition we can define an auxiliary function $R_{x_0} : S^{d-1} \mapsto P$, which records $\tilde{P} \subset P$ – a subset of all points in P visible from x_0:

$$R_{x_0}(z) := \begin{cases} y_i, & \text{if } \rho_{x_0}(z) > |x_0 - y_i| \\ \text{value unchanged}, & \text{otherwise} \end{cases} \tag{3}$$

In case the surface normals are available for each data point, we can use ellipse instead of a ball in the above construction. In [11], a similar projection approach is proposed for rendering purposes.

2.2 Smoother Reconstruction by ENO Interpolation

Note that analytically the visibility function ρ is piecewise continuous with jumps corresponding to the locations of horizons. Smoothness of ρ in each of its continuous pieces relates to the smoothness of the corresponding visible part of Γ, i.e. $\Gamma^*_{x_0}$. In the previous section we obtained a piecewise constant approximation $\tilde{\rho}_{x_0}$ of the visibility function and recorded an auxiliary function R_{x_0} which keeps track of the visible data points serving as "originators" of the constant values of $\tilde{\rho}_{x_0}$. We will use R_{x_0} to construct a piecewise polynomial approximation ρ_{int} to the visibility function which would preserve the jumps. ENO (Essentially Non-Oscillatory) interpolation introduced by Harten et al [7] is used to compute such a ρ_{int}.

For example, consider a two dimensional reconstruction on S^1. First, parameterize S^1 by angles $\theta \in [-\pi, \pi)$. Then sort the visible points $p_i \in \tilde{P}$ in the increasing order of the angle they form with respect to the vantage point: $\rho^{-1}_{x_0}(p_i) = \arg(p_i - x_0)$. To construct a piecewise linear interpolation $\rho^{ENO(1)}_{x_0}$ use the values of $\tilde{\rho_{x_0}}(\theta)$, where $\theta \in I[\tilde{\rho}^{-1}(p_i), \tilde{\rho}^{-1}(p_{i+1}))$. Similarly, we can obtain $\rho^{ENO(p)}_{x_0}$ – a piecewise p-th order interpolation. See Fig.2 for an example.

ENO interpolation can be applied in two steps to compute an approximation on S^2 for organized data clouds. Let θ_1 and θ_2 parameterize S^2. We first ENO-interpolate $\rho^{(0)}_{x_0}(\cdot, \theta_2)$ in the θ_1 direction to obtain $\rho^{ENO(p,*)}_{x_0}$. Then use $\rho^{(0)}_{x_0}(\theta_1, \cdot)$ and $\rho^{ENO(p,*)}_{x_0}$ to interpolate in θ_2 direction to obtain $\rho^{ENO(p,q)}_{x_0}$. Figure 3 is an example in three dimensions.

We shall use the piecewise p-th order approximation $\rho^{ENO(p)}_{x_0}$ to compute derivatives on the occluding surfaces (away from the edges) and easily extract various geometric quantities.

2.3 Curved Lines of Sight

To demonstrate the flexibility of our formulation, consider the case when the lines of sight are no longer straight. Then we can not use the relation $\nu(x, x_0) = (x - x_0)/|x - x_0|$ in the definition of the visibility function (1). As in [15], we consider instead the flow lines connecting x_0 to the data points $p \in P$. The

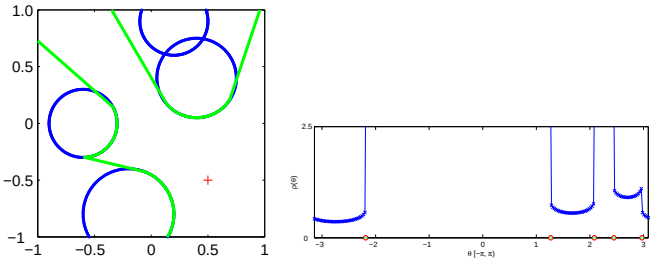

Figure 2: Points visible from $(0.5, -0.5)$, corresponding visibility function $\rho(\theta)$, and the edges (horizon points)

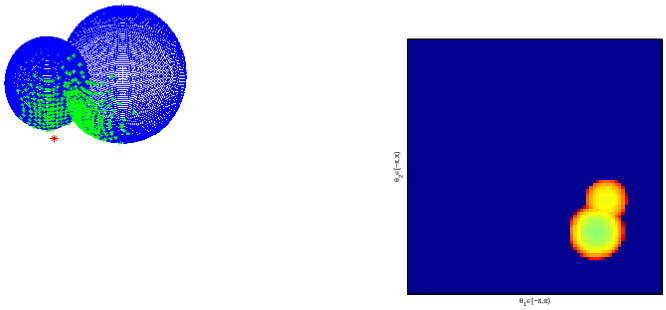

Figure 3: Visible points from a vantage point marked by red star and corresponding visibility function $\rho(\theta_1, \theta_2)$

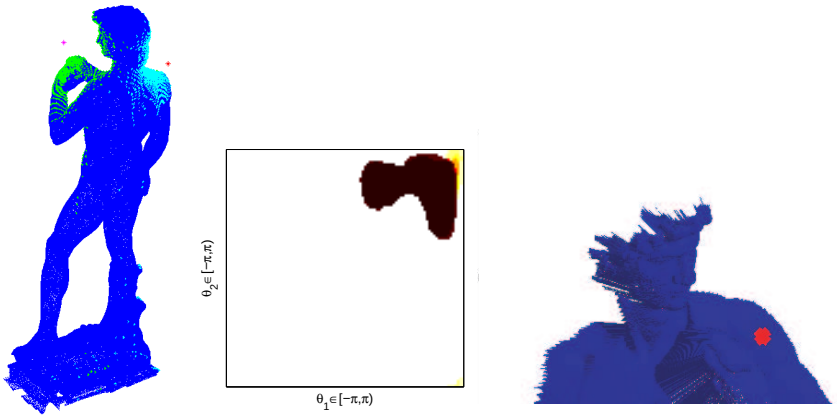

Figure 4: Visible points from two vantage points, one of the corresponding visibility maps $\rho(\theta_1, \theta_2)$, and a reconstruction of the surface.

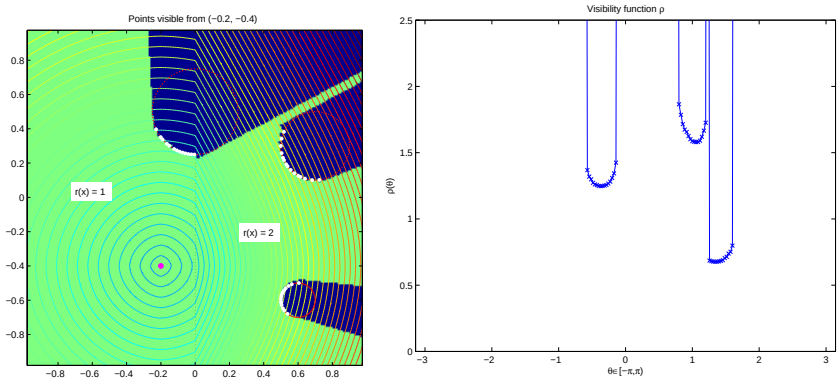

Figure 5: Left: non-straight lines of sight; Right: corresponding visibility function $\rho(\theta)$

construction of the visibility function is done as follows. First, we construct the distance function φ on the whole domain D by solving the eikonal equation

$$|\nabla\varphi(x)| = r(x), \quad \text{in } D,\ \varphi(x_0) = 0, \tag{4}$$

where $r(x) > 0$ is the variable index of refraction. We use the fast sweeping technique from [16] to solve (4). To determine the polar coordinates $(\theta, \rho(\theta))$ corresponding to the point $\mathbf{p}$ on the occluding surface we then solve

$$\begin{aligned} &\frac{\partial x}{\partial t} = -\nabla\varphi(x), \\ &x|_{t=0} = p, \end{aligned} \tag{5}$$

to trace point p back to x_0 along the line of sight connecting them. Then θ is the angle made by $\nabla\varphi$ at x_0, and $\rho(\theta) = \varphi(p)$. The visibility function can be constructed using the causality condition with respect to φ. See Figure 5.

Such computations may be useful when determining visibility in regions with variable refraction such as water or fog, or in anisotropic medium (in this case, one needs to solve more general Hamilton-Jacobi equations as considered in [15]).

2.4 Dynamics

When the lines of sight are straight, we can derive how the visibility changes along with a moving vantage point x_0. In two dimensions let us consider a coordinate system centered at $\mathbf{x_0}$ with the visible portions of the occluding surfaces parameterized by polar coordinates. A point $\mathbf{z}$ on the occluder is visible from x_0. Assume the observer moves with the velocity $v = (v_1, v_2)$. The value of the visibility function is $\rho_{x_0}(\theta) = |z - x_0|$. Suppose during the period of time Δt the observer has moved to a new location $x_0 + v\Delta t$. The corresponding value of the visibility function is $\tilde{\rho}_{x_0+v\Delta t}(\tilde{\theta}) = |z - (x_0 + v\Delta t)|$. The angle between the velocity vector v and the x-axis is $\varphi = \tan^{-1}\frac{v_2}{v_1}$. The angle between $z - x_0$ and the velocity vector v is ψ. Then, the angle between $z - x_0$ and the x-axis is $\theta = \varphi + \psi$.

We can obtain the following expressions:

$$\frac{d\theta}{dt} = |v| \sin\psi, \tag{6}$$

$$\frac{d}{dt}\Big(\rho(\theta(t),t)\Big) = \rho_t + \rho_\theta \theta_t = \frac{d}{dt}|x_0(t) - z|. \tag{7}$$

Now we can put (6) and (7) together to get

$$\rho_t + |v(t)| \sin\psi \rho_\theta = v(t) \cdot \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix}. \tag{8}$$

Now let us consider the motion of horizon points e_1 and e_2. Note that $(e_i - x_0) \cdot n_{e_i} = 0$, where n_{e_i} is the outer unit normal to the occluding surface at the point e_i for $i = 1, 2$. That is, $e_i - x_0$ is tangent to the occluding surface at the horizon point. Without loss of generality, in all future computations we will consider just e_1.

In the coordinate system centered at x_0, $\theta = \varphi + \psi$ is the angle between $e_1 - x_0$ and the x-axis. The value of the visibility function is $\rho_{x_0}(\theta) = |e_1 - x_0|$. Now suppose the observer moves to a new position $x_0 + v\Delta t$, moving with the velocity $v = (v_1, v_2)$. For this new location, the position of the edge has changed to $\tilde{e}_1$ and the corresponding value of the visibility function is $\tilde{\rho}_{x_0+v\Delta t}(\tilde{\theta}) = |\tilde{e}_1 - (x_0 + v\Delta t)|$. Here $\tilde{\theta} = \varphi + \tilde{\psi}$ is the angle between $\tilde{e}_1 - (x_0 + v\Delta t)$ and the x-axis in the coordinate system centered at $x_0 + v\Delta t$. Our goal is to find the change in the position of horizon, i.e. $\frac{d}{dt}e_1$.

First, note that the curvature of the occluding surface at the point $(\rho(\theta), \theta)$ is given by

$$\kappa = \frac{\rho^2 + 2\rho_\theta^2 - \rho\rho_{\theta\theta}}{(\rho^2 + \rho_\theta^2)^{\frac{3}{2}}}. \tag{9}$$

Also, since $e_1 - x_0$ is tangent to the occluder at e_1, we obtain

$$\begin{aligned} n^\perp(e_1) &= \frac{e_1 - x_0}{|e_1 - x_0|} \\ n(e_1) &= \Big(n^\perp(e_1)\Big)^\perp = \Big(\frac{e_1 - x_0}{|e_1 - x_0|}\Big)^\perp. \end{aligned} \tag{10}$$

Now we can plug in the above into the formula for horizon dynamics from [15] to get

$$\frac{de_1}{dt} = \frac{1}{\kappa} \frac{v \cdot n(e_1)}{|e_1 - x_0|} n^\perp(e_1), \tag{11}$$

or, using the fact that $v \cdot n(e_1) = |v| \cos(\psi + \frac{\pi}{2})$,

$$\frac{de_1}{dt} = \frac{|v| \cos(\psi + \frac{\pi}{2})}{\kappa\rho^2}(e_1 - x_0). \tag{12}$$

Remember that in all of the above $\psi = \theta - \varphi = \theta - \tan^{-1}\frac{v_2}{v_1}$.

Therefore, from (8) and (12) we obtain full description of the change in the visible portion of the occluder with respect to the observer's motion.

The corresponding expressions can also be derived in three dimensions, see [15].

3 Applications of Visibility Interpolation to Navigation Problems

Let us consider the application of visibility to navigation in an unknown environment, for example exploring the environment, object finding, and pursuit-evasion. LaValle et al have addressed these problems in [6], [13], [14], [12], [9]. Their algorithms only work on polygonal domains or curved regions whose boundary may be represented as a set of solutions to an implicit polynomial equation of the form $f(x_1, x_2) = 0$ (see [9]). Our algorithms work on general types of environments using point cloud data that is either presampled or sampled in action by some hardware.

3.1 Problem: Seeing the Whole Environment

Here we consider the problem of exploring the unknown bounded region with obstacles. The objective is to map the whole environment. We set the following restrictions on the path traveled by the observer:

1. The path should be continuous and consist of discrete steps;
2. The number of steps should be finite;
3. The total distance traveled must be finite.

These restrictions ensure that the algorithm would be practical in real life applications. Consider first simple, but non-practical examples of navigation in a bounded region with a single occlusion in shape of a circle. One strategy to explore the environment around the occlusion would be to approach the circle's boundary and travel along it until we return to the same point. This strategy does not satisfy our restrictions since it would require an infinite number of steps to travel along the boundary. Another strategy would be to proceed to infinity to see half a circle at once, then jump to infinity at the opposite side of the circle to see the other half. Such a strategy does not satisfy our restrictions either, since the path would be infinite and not continuous.

Our algorithm was inspired by LaValle et al. In this method the observer randomly chooses a gap marked on the visibility plot and approaches it. The visibility map is then updated and the process is repeated until the whole region is explored. Critical events such as appearance and disappearance of gaps are tracked by the dynamic data structure. Since the visibility map has no distance or angular information, the algorithm is not optimal with respect to the total distance traveled. In particular, if this algorithm is applied to cases that contain fine polygonalization of curved objects, the computational cost of this algorithm may become too large.

Our visibility representation includes distance and angular information, and our algorithm is designed with the consideration of handling basic smooth geometries. In essence, the visibility of any bounding disk of a convex object guarantees the visibility of that object. If a set of separated, non-overlapping bounding disks exists for a collection of disjoint convex objects, we may consider the visibility problem of each convex object independently. Furthermore, if a non-convex object can be decomposed by the set difference of a finite number of convex sets, then one can treat it "almost" like a convex object. We see that the

signed curvature, a notion of local convexity, is rather essential in applying the above arguments. We obtained formulas for the upper bounds of the number of observing locations in these situations as functions of the sign changes in the curvatures as well as the number of disconnected components, and would report our finding in a forthcoming paper.

ALGORITHM 1

1. For the given x_0 outside the occluding objects; construct the visibility function $\rho(\theta)$;
2. Find all the edges on the $(\theta, \rho(\theta))$ map and proceed to the nearest edge;
3. Find edges;

   If no edges are found, we are on the boundary of an obstacle at the horizon point. Thus we need to "overshoot" x_0 along the tangent line to see where to proceed next. We choose the following overshooting step size
   $$r = \lambda \tan\left(\frac{\pi}{3}\right) \frac{1}{\kappa}, \tag{13}$$
   where κ is the curvature of an edge defined by (9) and λ is a parameter. This way we have a minimal number of steps to travel around the obstacle, e.g. for a circle, $r = 1/2$ the side of the equilateral triangle enclosing the circle. In case $\kappa = 0$ we shall shift the position by a small amount to see the next edge.

   If the edges are found, move x_0 to the nearest edge. Store the unexplored edges in a list;

4. Finish when the change in total visible area is less then the desired tolerance and all the edges are are "removed" from the list. Otherwise go to 1 with the current location of x_0.

Figure 6 illustrates the steps of the above algorithm with one and two circles as obstacles. The pink arcs correspond to the portions of the circles that are reconstructed. Figure 7 depicts final paths for different test cases. As one can see from the examples, the algorithm handles both convex and concave obstacles. The algorithm always converges, however it does not provide the desired optimality with respect to the total distance traveled.

We remark that ALGORITHM 1 can be applied to the curved lines of sight cases with following modifications. Since the curvature of the occluding surfaces cannot be recovered from the visibility function ρ, the overshoot step-size must be defined by the user in step 3. To proceed further from the edge we follow the line of sight passing through this edge by solving

$$\frac{dx}{dt} = \nabla\phi(x), x(0) = x_e, \tag{14}$$

where x_e is the position of the edge. Consider Fig.8 for a sample step-by-step path.

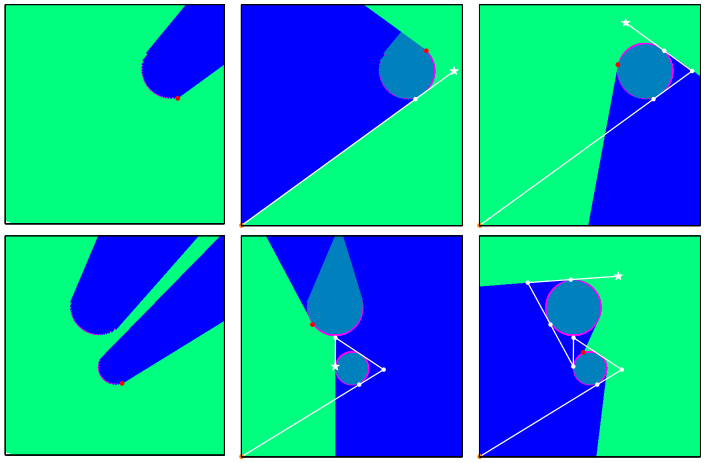

Figure 6: Steps of the exploration algorithm with one and two circles as obstacles

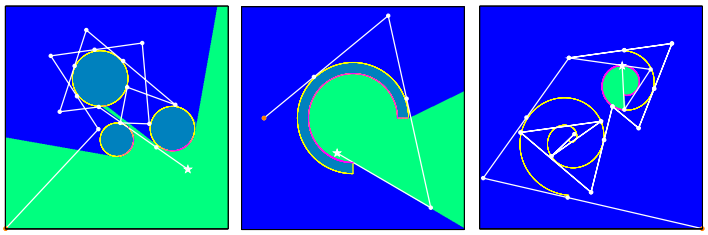

Figure 7: Full paths for different obstacles

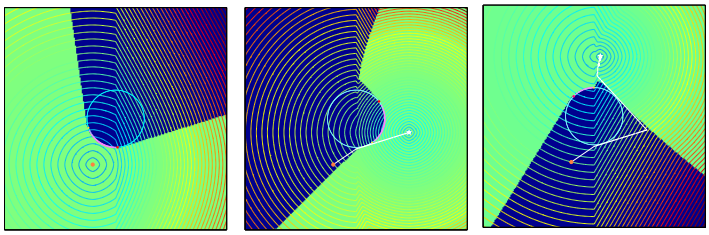

Figure 8: Full path in curved rays

4 Conclusion

We present an essentially non-oscillatory algorithm for interpolating point cloud visibility information in polar coordinates. This algorithm is capable of approximating higher order derivatives of the surface so that curvatures can be computed. We also present a new path planning algorithm using our point cloud visibility interpolation. Our future work lies in optimizing the above algorithm. We desire a better performance with respect to the distance traveled and/or the number of steps.

5 Acknowledgement

Landa's research is supported by ONR MURI Grant N00014-02-1-0720, Tsai's research is supported by NSF DMS-0513394, and Cheng's research is supported by Alfred P. Sloan Fellowship and NSF Grant 0511766.

References

[1] A. Atle and B. Engquist, "On surface radiation conditions for high frequency wave scattering", preprint, (2006).

[2] O. Bruno, C.A. Geuzaine, J.A. Monro Jr., and F. Reitich, "Prescribed error tolerances within fixed computational times for scattering problems of arbitrarily high frequency: the convex case", *Philos. Trans. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci.*, **1816**,(2004), 629-645.

[3] W.-P. Chin and S. Ntafos, "Shortest watchman routes in simple polygons", *Discrete Comput. Geom.*,**6** (1991), 9-31.

[4] M. Ghomi,"Shadows and convexity of surfaces", Annals of Mathematics, **155** (2002), 281-293.

[5] J.E. Goodman, J.O'Rourke, editors "Handbook of discrete and computational geometry", CRC Press LLC, Boca Raton, FL; Second Edition, April 2004

[6] L. Guilamo, B. Tovar, S.M. LaValle, "Pursuit–evasion in an unknown environment using gap navigation graphs" *IEEE International Conference on Robotics and Automation*, (2004), under review.

[7] A. Harten, B. Engquist, S. Osher, S.R. Chakravarthy, "Uniformly high order accurate essentially nonoscillatory schemes, III," *Journal of Computational Physics*, **71**, (1987), 231-303.

[8] H. Jin, A. Yezzi, H.-H. Tsai, L. T. Cheng and S. Soatto, "Estimation of 3D surface shape and smooth radiance from 2D images; a level set approach", To appear, *Journal of Scientific Computing*, **19**, (2003), 267-292.

[9] S.M. LaValle, J. Hinrichsen "Visibility based pursuit-evasion: An extension to curved environments", *Proc. IEEE International Conference on Robotics and Automation*, (1999), 1677-1682.

[10] D. Murray and C. Jennings. "Stereo vision based mapping for a mobile robot", *Proc. IEEE Conf. on Robotics and Automation*, (1997).

[11] S. Rusinkiewicz and M. Levoy, "QSplat: A multiresolution point rendering system for large meshes", *SIGGRAPH*,(2000),343-352

[12] S. Sachs, S. Rajko, S.M. LaValle "Visibility based pursuit-evasion in an unknown planar environment", to appear in *International Journal of Robotics Research*, (2003).

[13] B. Tovar, S.M. LaValle, R. Murrieta, "Locally-optimal navigation in multiply-connected environments without geometric maps", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, (2003).

[14] B. Tovar, S.M. LaValle, R. Murrieta,"Optimal navigation and object finding without geometric maps or localization", *Proc. IEEE/RSJ International Conference on Robotics and Automation*, (2003).

[15] Y.-H.R. Tsai, L.-T. Cheng, S. Osher, P. Burchard, G. Sapiro, "Visibility and its dynamics in a PDE based implicit framework" *Journal of Computational Physics*, **199**, (2004), 260-290.

[16] Y.-H.R. Tsai, L.-T. Cheng, S. Osher, H.-K. Zhao, "Fast sweeping methods for a class of Hamilton-Jacobi equations", *SIAM J. Numer. Anal.*, **41**(2), (2003) 673-694.

[17] J. Urrutia. "Art gallery and illumination problems", In J. R. Sack and J. Urrutia, editors, Handbook of Computational Geometry, pages 973-1027, 2000.

[18] D. F. Wolf, Andrew Howard, and Gaurav S. Sukhatme. "Towards Geometric 3D Mapping of Outdoor Environments Using Mobile Robots", *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*,(2005),1258-1263.