COMMUN. MATH. SCI. © 2008 International Press
Vol. 6, No. 4, pp. 881–913

VISIBILITY OF POINT CLOUDS AND EXPLORATORY PATH PLANNING IN UNKNOWN ENVIRONMENTS[*]

YANINA LANDA[†] AND RICHARD TSAI[‡]

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 location and performs essentially non-oscillatory (ENO) interpolation of the projected data. Curvatures of the occluding objects can be approximated and used in many ways. We demonstrate how our visibility formulation can be incorporated into novel algorithms for mapping unknown environments with single or multiple observers, and target finding problems. A convergence proof is provided indicating suitability of our algorithm for some canonical types of environments. Various postprocessing optimization techniques are considered to obtain a more uniform exposure of the region along the path.

Key words. Visibility, interpolation, point clouds, path planning

AMS subject classifications. 65D18, 68W05, 68T40, 68T45

1. Introduction

The problem of visibility involves the determination of regions in space visible to a given observer when obstacles are present. When the observer is replaced by a light source in the simplified geometrical optics setting with a 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 high frequency wave propagation problems and is needed for many computational high frequency wave approaches. 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, [8] studies the connectedness of the surface shadow and the convexity of the occluding surface.

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 vantage location(s) that maximize visibility using a certain predefined metric.

In many visualization applications, (1) is solved by projecting triangles. The question studied in [8] can be viewed as in category (2). The surveillance problems

[*] Received: May 7, 2008; accepted (in revised version): August 25, 2008. Communicated by Shi Jin.

[†] Department of Mathematics, University of California, Los Angeles, CA 90095, USA (ylanda@math.ucla.edu). Landa's research is supported by ONR MURI subcontract from Stanford University and ARO MURI subcontract from University of South Carolina.

[‡] Department of Mathematics, University of Texas at Austin, TX 78704 (ytsai@math.utexas.edu). Tsai's research is supported by NSF DMS-0513394 and ARO MURI subcontract from University of South Carolina, and a Sloan Fellowship. Tsai thanks Center for Theoretical Sciences at Taipei for hosting part of this research.

are related to (3). We will present an algorithm for a problem related to both (1), (2), and (3).

In the following, we summarize the main problems under consideration of this paper.

Problem 1.1. *We are given a vantage point and a set of points (a point cloud) that are evenly distributed over solids with piecewise smooth surfaces.*
Construct a high order accurate representation of the portions of the solid surfaces that are visible to the vantage point and also the corresponding occlusion volume.

Problem 1.2. *We are given a bounded domain with unknown solid obstacles and a vantage point. Assume an evenly distributed set of points can be sampled from the portions of the unknown solids that are visible to a given vantage location.*
Construct a piecewise linear path so that (a) any point on the solid surfaces is seen by at least one vertex of the path, and (b) an accurate representation of the solids is constructed from the point clouds that are collected at the vertices of the path.

In practice this data could be obtained from sensors such as LIDAR or even from triangulated surfaces (here the point cloud would be the set of vertices).

The outline of the paper is as follows. In Section 2 we define the visibility of point clouds along with ENO interpolation for smoother visibility approximation. Error analysis of the resulting interpolant is performed in Appendix A, whereas in Appendix B we derive the dynamics of the visibility function with respect to observer's motion. Section 3 is devoted to application of our visibility formulation to motion planning in an unknown environment. We propose a navigation algorithm for a single and multiple observers. We furthermore present an application of the exploration algorithm to a target-finding problem in an unknown environment, given a target location. Simulation results and statistics demonstrate robustness of our algorithm in different types of environments. A rigorous convergence proof of our single observer algorithm in an environment with an arbitrary number of disjoint convex obstacles is provided Appendix C. Finally, in Section 4 we consider postprocessing of the exploration path via optimization with respect to uniform illumination of the region of interest.

1.1. Representations of visibility. Today computational geometry and combinatorics are the primary tools to solve visibility problems [9, 26, 4]. 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.

We define a representation of visibility in a regime, where visible lights are modeled accurately by rays, and with it, we solve the problems including but not limited to those considered in computational geometry [9], on general environments in two or three dimensions. We require this representation of visibility to utilize minimum information about the environment, be efficient to compute, and to allow for accurate computations of geometrical quantities such as curvature of the occluding surface.

One approach is to introduce a level set representation of the occluding objects and the visibility function, defined in [24]. This formulation can be applied to general types of environments and easily extended to three dimensions and curved lines of sight. However, 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 developed by LaValle *et al.* in [14, 19, 20, 21, 22], and [23]. 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.

Our new model can handle complicated geometries and has been extended to three dimensions [11], and curved lines of sight [13, 11], in a similar fashion as the level set representation [24]. Furthermore, unlike [24], it does not require any *a priori* information about the environment. In contrast to LaValle's representation, we utilize distance and angular information, which, in practice, can be easily provided by the sensor.

1.2. Robotic path planning with visibility considerations.

Consider a bounded region which may contain an unknown number of arbitrary positioned obstacles of unknown general shapes. Our goal is to obtain an algorithm that would utilize visibility information to allow an autonomous observer(s) equipped with a range sensor to fully explore the region and map the obstacles' boundaries. The latter refers to the construction of an accurate mathematical representation of the obstacles.

In [27], an algorithm extracting planar information from point clouds is introduced and is used in mapping an outdoor environment. In [15], depth to the occluding objects is estimated by a trinocular stereo vision system and is then combined with a predetermined "potential" function so that a robot can move to the desired location without crashing into obstacles.

A wall-following curvature-based control algorithm is introduced in [28] and evaluated with real-life robots equipped with range sensors in [29]. Even with high sensor precision, curvature estimates have significant inaccuracies in the absence of filtering. The noise in curvature computations is related to the computation of derivatives of the range data which are prone to noise. To deal with this problem, we employ essentially non-oscillatory interpolation (ENO) from [10] to obtain high order interpolation of the range data, so that derivatives can be easily estimated away from discontinuities.

The motivation for our navigation algorithm comes from work of Tovar *et al.* [19, 20, 21, 22], and [23]. In [23], a single robot (observer) must be able to navigate through an unknown simply or multiply connected piecewise-analytic planar environment. The robot is equipped with a sensor that maps onto a circle relative locations of discontinuities in depth information, i.e., gaps, in the order of their appearance with respect to the robot's heading. Each gap corresponds to a connected portion of space that is not visible to the robot.

To navigate the environment the observer approaches one of the gaps. The visibility map is then updated and the process is repeated until the whole region has been explored. Critical events such as appearance and disappearance of gaps are tracked by the dynamic data structure. As a result of exploration, the region is characterized by the number of gaps and their relative positions. No distance or angular information is accumulated.

In contrast, our algorithm maps the obstacles in cartesian coordinates as the observer proceeds through the environment, and utilizes the recovered information for further path planning. At the termination of the path all the obstacles' boundaries are reconstructed. Thus we obtain a complete representation of the environment. The algorithm is easily scalable to allow for multiple observers. We rigorously prove that for a general class of environments our algorithm maps the entire region in finite number of steps. Furthermore, we present the statistics of the number of discrete steps used by our algorithm for the exploration of the entire given region.

In [12], our algorithm has been validated on a group of autonomous vehicles equipped with range-sensors whose task is to explore an unknown bounded region and construct the map of the explored environment. Satisfactory results have been obtained in mapping an unknown environment using multiple mobile inexpensive sensors where noise is an issue.

2. Visibility of point clouds and surface reconstruction

Our approach is based on the observation that visibility along each ray emanating from the vantage point satisfies a *causality condition*: if a point is occluded, then all other points farther away from the vantage point along the same ray are also occluded.

The first step of our algorithm, in some sense, can be viewed as the reverse action of ray tracing, where discrete rays are sent out from the origin to sample given surfaces. However, instead of assuming a complete explicit or implicit representation of the surfaces, we assume that a set of points is "uniformly" sampled from the occluding surfaces. In practice this data could be obtained from range sensors such as LIDAR or from triangulated surfaces (here, the set of vertices can be regarded as the points sampled from the surfaces).

Given a vantage point, our algorithm retains a subset of visible data points and constructs a piecewise polynomial interpolation of the visible portions of the surfaces. Unlike the level set representation [24], our algorithm can handle open surfaces and does not require *a priori* knowledge of occluding surfaces to construct visibility. Our scheme can be regarded as a surface reconstruction scheme for the portions of surfaces that are visible to the given vantage point.

The algorithm consists of the following steps:

1. Begin with the point cloud P sampled from the occluding surfaces.
2. Project P onto a unit sphere centered at the vantage point x_0.
3. Filter out portions of P visible to the observer at x_0. [1]
4. Interpolate visible data to obtain a piecewise smooth reconstruction.

Details are presented in the following subsections.

2.1. Projection and filtering of data points.

Let $\mathcal{S}^{d-1}$ be the unit sphere in $\mathbb{R}^d$, centered at the origin. We set up a spherical coordinate system centered at x_0 by $y=x_0+r\nu$, where $\nu\in\mathcal{S}^{d-1}$ and $r=|y-x_0|$.

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, by $\pi_{x_0}(x_0+r\nu)=\nu$. Let Ω be a subset of $\mathbb{R}^d$. Define $\rho_{x_0}:\mathcal{S}^{d-1}\mapsto[0,M)$ by

$$\rho_{x_0}(\nu):=\min_{x_0+r\nu\in\bar{\Omega}\bigcup\partial B(x_0,M)} r, \tag{2.1}$$

where $B(y,M)=\{y'\in\mathbb{R}^d:|y-y'|<M\}$ is the unit disc with radius M centered at y.

[1] Note that this step is optional if P has only been sampled from x_0.

The points $\tilde{y} = \tilde{y}(\tilde{r},\nu) = x_0 + \tilde{r}\nu \in \Omega$ are classified as occluded for all $\tilde{r} > \rho_{x_0}(v)$. A point $y(r,\nu) = x_0 + r\nu \in \partial\Omega$ is called a horizon point if and only if $\nu \cdot n(y) = 0$, where $n(y)$ is the outer normal of $\partial\Omega$ at y. A point $y(r,\nu)$ is called a cast horizon point if and only if there exists a horizon point $y(r^*,\nu^*)$ such that $\nu = \nu^*$ and $r^* = r$.

Given the set of data points $\{y_j\}$ and a partition of the unit sphere $\mathcal{S}^{d-1} = \cup_{i=0}^{N} \bar{K}_i$, where K_i are open regions with diameter ϵ. We define a piecewise constant interpolation of ρ_{x_0} by

$$\tilde{\rho}_{x_0}(y) = \min_{y_j} |x_0 - y_j|, \text{ for every } y,\ \pi_{x_0} y \in K_i, i = 0,\cdots,N. \tag{2.2}$$

Consequently, we classify y as occluded if $\tilde{\rho}_{x_0}\big(\pi_{x_0}(y)\big) < |y - x_0|$. Thus we may define the visibility indicator

$$\Xi(y) := \rho_{x_0}\big(\pi_{x_0}(y)\big) - |y - x_0|, \tag{2.3}$$

such that $\{\Xi \geq 0\}$ is the set of visible regions and $\{\Xi < 0\}$ is the set of regions invisible from x_0.

In case the surface normals are available for each data point, we can use an ellipse instead of a ball in the above construction. A similar approach is also used by QSplat in rendering of the digitized data of Michelangelo's statues [17].

In the case where light rays are curved, the term $|x_0 - y_j|$ in (2.2)–(2.3) is replaced by the arc length of the ray connecting x_0 and y_j. This can easily be computed by solving an eikonal equation (see [13] for more details).

2.2. Smoother reconstruction by ENO interpolation. Note that, analytically, the visibility function ρ_{x_0} is piecewise continuous with jumps corresponding to the locations of horizons. Smoothness of ρ_{x_0} in each of its continuous pieces is related to the smoothness of the corresponding visible part of $\partial\Omega$. In the previous section we obtained a piecewise constant approximation $\tilde{\rho}_{x_0}$ to ρ_{x_0} using formula (2.2). Along the way, we also extract a subset of visible data points $\tilde{P} \subseteq P$ serving as "originators" of each constant value of $\tilde{\rho}_{x_0}$.

The edges, or discontinuities in the visibility function ρ typically occur near the locations of horizons. A standard choice in image processing for the edge-detection function $g : \mathbb{R} \mapsto (0,1]$, is $g(s) = 1/\big(1 + s'^2\big)$ [1]. If the value of $g(s)$ is below some threshold value, we get an edge. The threshold value depends on the sampling of s.

We implement a modified version of the edge-detection function. Using the piecewise constant values of the visibility function $\tilde{\rho}_{x_0}$, we substitute a finite difference approximation for the derivative of ρ. The resulting edge-detection function $g : \mathcal{S}^1 \mapsto [0,1)$ maps θ onto

$$g\left(\tilde{\rho}_{x_0}(\theta)\right) = 1 \Bigg/ \left(1 + \left(\frac{\tilde{\rho}_{x_0}(\theta_{i+1}) - \tilde{\rho}_{x_0}(\theta_i)}{\theta_{i+1} - \theta_i}\right)^2\right), \theta_i \leq \theta < \theta_{i+1}. \tag{2.4}$$

Periodic boundary conditions are used in this formulation. A natural choice of the threshold value is the polar grid size $\delta\theta$. The need for Gaussian smoothing of the derivatives is needed when strong noise in present in the data as in [12]. In our applications, where the data came from, *e.g.*, LIDAR, noise is insignificant.

We can then use $\tilde{\rho}_{x_0}$ to construct a piecewise polynomial approximation $\rho_{x_0}^{\text{int}}$ to the visibility function, which preserves the jumps. Essentially non-oscillatory (ENO) interpolation is used to compute such $\rho_{x_0}^{\text{int}}$ away from discontinuities. ENO interpolation is a nonlinear polynomial interpolation that has been widely and successfully used in shock problems of computational fluid dynamics. We refer the readers to the seminal paper [10] for more detail.

Below we describe the interpolation procedure on $\mathcal{S}^1$ for two dimensional problems. Possible extension of the strategies to the three dimensional problems will be described in a forthcoming publication.

2.2.1. The two-dimensional case. Denote the extracted visible data points by $p_i \in \tilde{P}$. Since $\mathcal{S}^1$ can be parameterized by angles $\theta \in [-\pi,\pi)$, we can sort the points in $\tilde{P}$ in the increasing order of the angle they form with respect to our chosen spherical coordinate system; that is, points in $\tilde{P}$ are sorted in the increasing order of $\tilde{\rho}_{x_0}^{-1}(p_i) = \arg(p_i - x_0)$. Note that the restriction of the angle of visualization is typically a consequence of the hardware. In this paper, we assume an idealized situation, parameterizing the entire $\mathcal{S}^1$, similar to previous work in robotics such as LaValle's work [14].

We can then construct $\rho_{x_0}^{(1)}$ by linearly interpolating between each successive pair of p_i and p_{i+1} if $\tilde{\rho}_{x_0}(\theta) \neq M$ for $\theta \in \left[\tilde{\rho}_{x_0}^{-1}(p_i), \tilde{\rho}_{x_0}^{-1}(p_{i+1})\right)$.

Instead of linear interpolation, we use ENO interpolation to construct $\rho_{x_0}^{(p)}$, a piecewise p-th order approximation of ρ_{x_0}. We use the piecewise p-th order approximation $\rho_{x_0}^{(p)}$ to compute derivatives on the occluding surfaces (away from the edges) and to extract various geometric quantities. We only consider uniform discretization of $\mathcal{S}^1$. The ENO interpolation uses Newton's divided differences and does not really require uniform grids. The requirement of uniform grids for ENO schemes for conservation laws is due to the special way the conservative schemes are built up. In our algorithm, we only use ENO interpolations, which are not the schemes for conservation laws. In Figure 2.1 (a) we illustrate visibility from the vantage point at $(-0.2, 0.6)$. A corresponding visibility function ρ, its derivatives, and curvature κ are displayed in Figure 2.1 (b). As one can see, we obtain a high order approximation of the derivatives and, subsequently, curvature along the visible occluding boundaries away from the discontinuities corresponding to horizons in Figure 2.1.

The error of this procedure is analyzed in Appendix A.

We refer the reader to [13] for the details and examples of constructing visibility in curved lines of sight, *i.e.* when the index of refraction varies across the domain. These examples further demonstrate the flexibility of our formulation.

2.2.2. Conversion to Cartesian level set formulation. The piecewise polynomial reconstruction of the visibility function ρ may be used to obtain a smooth level set visibility function ϕ defined on a Cartesian coordinate system. The following construction yields a level set visibility function that is smooth across the discontinuities. Begin by defining

$$G := \{(\theta, r) : r < \rho(\theta)\} \tag{2.5}$$

the set containing the visible points on polar coordinates. We proceed to construct a smooth signed distance function ϕ to the shadow boundary ∂G using redistancing [2]:

$$\begin{cases} \phi(\theta,r) > 0, \text{ if } (\theta,r) \in G, \\ \phi(\theta,r) < 0, \text{ if } (\theta,r) \in G^C, \\ \phi(\theta,r) = 0, \text{ if } (\theta,r) \in \partial G. \end{cases} \tag{2.6}$$

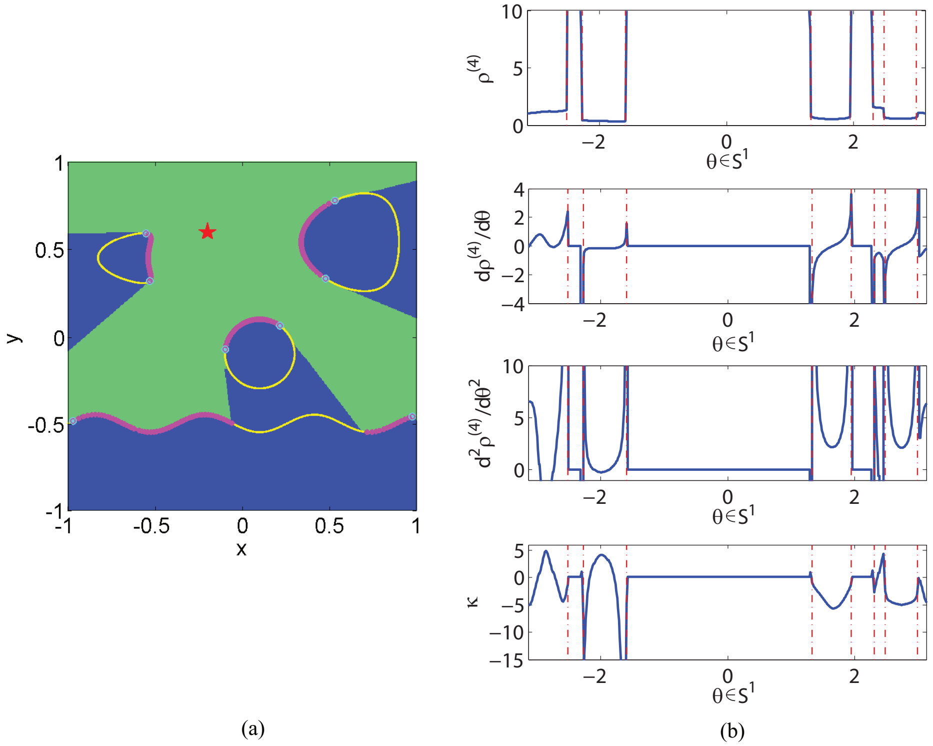


Fig. 2.1. *(a) Visibility map generated from artificial data: dark regions - invisible, light regions – visible, red star – vantage point* $(-0.2, 0.6)$*, magenta circles – visible boundary, yellow outline – actual boundary, cyan circles – horizon points. (b) Fourth order interpolation of the visibility function* ρ *corresponding to (a). Computation of* $\frac{d\rho}{d\theta}, \frac{d^2\rho}{d\theta^2}$ *and the curvature* $\kappa = \frac{\rho^2+2\rho_\theta^2-\rho\rho_{\theta\theta}}{(\rho^2+\rho_\theta^2)^{\frac{3}{2}}}$ *away from discontinuities (vertical dashed lines).*

The resulting signed distance function may then be easily converted from polar to cartesian coordinates

$$\phi(x,y) = \phi(x(\theta,r), y(\theta,r)).$$

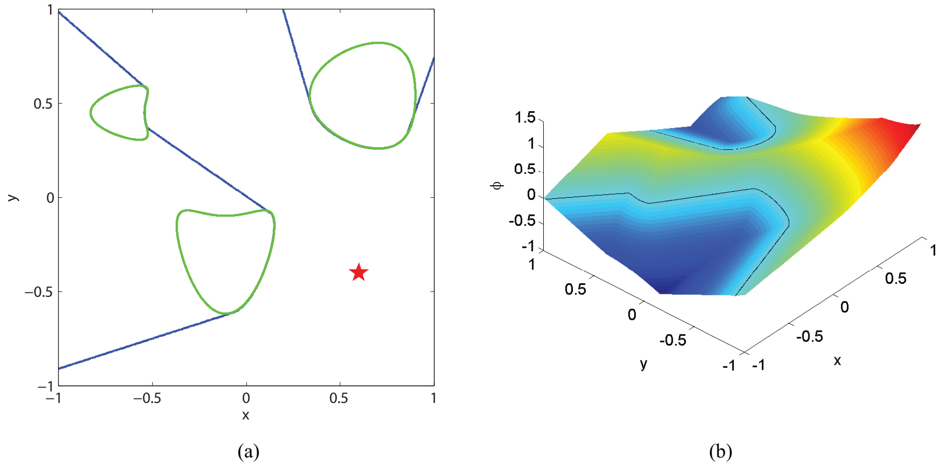
On the grid level, this is done by interpolation. Thus the obtained level set visibility representation is consistent with the one obtained in [24]. In Figure 2.2 we present the smooth level set visibility function corresponding to the vantage point marked by the red star.

The level set formulation on a fixed Cartesian coordinate system allows for easy Boolean operations on visibility of different vantage locations. For example, the joint visibility from two vantage points x_1 and x_2 may be defined on cartesian grid as

$$\phi_{x_1,x_2} = \max\{\phi_{x_1}, \phi_{x_2}\}. \tag{2.7}$$

This property can be used to construct a level set map of the visible region from multiple view points as in Figure 3.1.

2.3. Processing and Denoising. In real-life applications we frequently deal with noisy data. There are different sources of noise. For example, noise may be



(a)

(b)

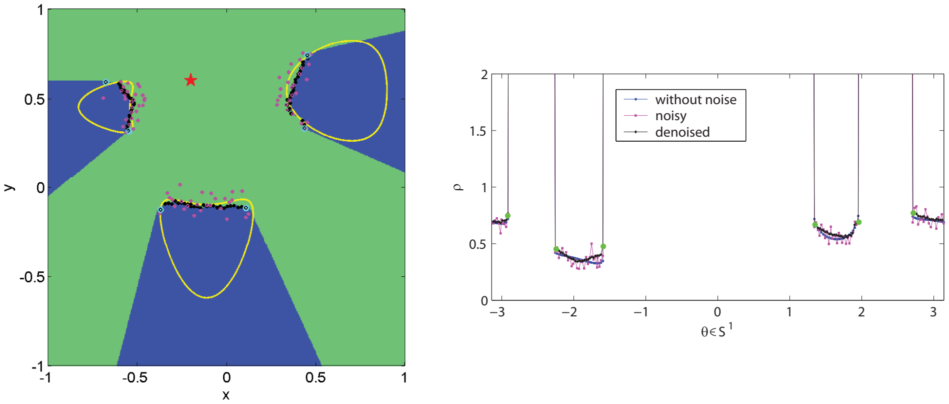
Fig. 2.2. *(a) Environment with obstacles (green contour), observer (red star), and shadow boundary (blue contour). (b) Visibility level set function ϕ.*

introduced by the measuring device as in [12] and [29]. As one can see from [29], even a high accuracy sensor produces significant error in curvature computations. Filtering is used in [29] to clean up the sensor data. In addition to sensor error, noise in the data can be introduced from an uneven terrain and/or presence of foliage, cars, and people in the scene, as in [27]. We consider applications using high precision LIDAR, where virtually no angular error is present.

We propose the use of a simple edge-preserving total variation based noise removal algorithm [16], which can be applied to the interpolated data to reduce the effect of noise in the scene. In Figure 2.3 (a), we plot visibility based on the denoised visibility function ρ depicted with black diamonds in Figure 2.3 (b). Here, an artificial noise of variance $\sigma = 0.05$ is added to the projected point cloud. The obtained data is then filtered and interpolated. Afterwards we apply the denoising algorithm from [16].

We would like to remark that application of ROF denoising may lead to loss of contrast and staircasing, both becoming more pronounced under severe noise in the data. The loss of contrast will result in an under-approximation of the objects that are closer to the observer and an over-approximation of farther obstacles. So, for the objects closer to the observer, the reconstructed obstacle may appear further from the observer than it actually is. And this may cause a collision of observer into the object. As in any image processing problem, one may obtain some estimate of the lost of contrast and modify the algorithm accordingly. We refer the reader to [18] for an example. To be very conservative, one may modify the navigation algorithm presented below (similar to the proof in Appendix C) so that one more observation/recosntruction is made before passing the edge point. The over-approximation for farther objects actually helps avoid the collision.

Staircasing, on the other hand, may result in identification of false edge points for the algorithm to pursue. We recommend enforcing a larger threshold on the edge detection algorithm to identify discontinuities in the function ρ. There are recent developments in image processing techniques that greatly reduce both typical problems related to the ROF algorithm. We point out the nonlocal means techniques of Morel



(a) (b)

Fig. 2.3. *(a) Visibility map generated from noisy data. Magenta circles – noisy visible boundary, black circles – denoised visible boundary, yellow outline – actual boundary, cyan circles – horizon points, dark regions – invisible with respect to the denoised visibility function, light regions – visible. (b) Visibility function* ρ *corresponding to (a). Green circles – edges (horizon points).*

[5], Osher [7], and Bregman iteration techniques [6]. One sees that the problems of loss of contrast and staircasing can be considered almost solved for a wide range of noise.

3. Exploratory path for mapping unknown environments

Here we consider the application of visibility to the problem of exploration of an unknown bounded two-dimensional region which may contain many disconnected obstacles. Our objective is to construct a path for an observer, so that at the termination of the path the observer will have seen the entire domain. In addition, a map (or an accurate description) of the domain representing the boundaries of the obstacles would be constructed. Our algorithm is designed with the consideration of handling general geometries. For practicality, we set the following constraints on the observer's path:

1. The path is continuous and consists of discrete steps.
2. The number of steps is finite.
3. The total distance traveled is finite.

The intuition behind our algorithm is the following. Assume some portions of the obstacles' boundaries are visible to the observer from a given vantage point. Each continuous portion of the visible boundary terminates with the horizon points, or edges on the visibility map, as in Figure 2.1. These horizon points are similar to an edge of the door that is ajar. One must proceed beyond the edge of the door to see more. Similarly, an observer must proceed beyond the horizon point to gain new information about the environment. At this point, we need to decide how far the observer should march beyond the chosen horizon point. Our strategy relies on the geometry of the obstacle near the horizons. Briefly, if the obstacle is a simple circle, then naturally the march distance should depend on the radius of the circle; this corresponds directly to the curvature at horizon points. These simple insights allow us to construct a path consisting of discrete steps.

A related approach is described in [19]. At each step of the navigation algorithm, the observer randomly chooses to approach one of the gaps, *i.e.* depth discontinuities projected onto $\mathcal{S}^1$. However, in contrast to our discrete approach, a practical implementation of the algorithm from [19] requires constant gap tracking. Furthermore, a practical implementation of this algorithm requires a wall-following procedure. Additional modifications of the algorithm are required when dealing with multiply connected environments, *i.e.* markings of once visited gaps. Meanwhile, our algorithm does not require any special treatment of certain types of environments.

In the following subsections we are going to describe our horizon-chasing algorithm for a single observer and its extension to the case of multiple observers. We also provide the results of navigation simulations in sample environments. In Section 3.3.2, we provide the statistics of our algorithm for some general types of environments. Appendix C contains the convergence proof of the algorithm in multiply connected regions. A practical implementation of this algorithm on an economical cooperative control tank-based platform is described in [12].

3.1. Chasing the horizons. In this section we provide the navigation algorithm for a single observer operating in two dimensions. The key idea behind our algorithm is to proceed in the environment Ω by approaching one of the currently visible horizons. Once a new horizon appears, it is stored in a list. Once the horizon has been explored, it is removed from the list. The observer must explore every horizon in the list before the algorithm terminates. The observer is allowed to return and inspect previously skipped horizons if no more new horizons are available. The exploration is complete once there are no more horizons left to approach. The details are provided in Algorithm 1 below. The following discussion in this subsection applies to Algorithm 1.

As a result of the navigation algorithm, we obtain a complete map of the environment, *i.e.* polynomial interpolated boundaries of the obstacles along with the visibility indicator function Ξ which marks the interior and exterior of the obstacles. In addition, we may easily construct the level set representation of the reconstructed environment map via (2.7). The use of the level set maps in postprocessing algorithms will be presented in the next section. Further applications of the level set representation of visibility are described in [24] and [3].

Note that in steps 21 and 25, the observer must proceed beyond the horizon in order to see more new environment. We choose the overshoot step size to be inversely proportional to the curvature κ of the obstacle's boundary near the horizon. High order ENO interpolation allows us to compute curvature at the obstacle's boundary with desired accuracy. Our choice of the overshoot step size is further explained in Appendix C.

A parameter λ introduced in step 20 provides extra buffer space between the observer and the obstacle's boundary. It may depend on the observer's size and mobility. If application allows, λ may change in the process of exploration. For example, a smaller λ would allow the observer to explore narrow regions of high curvature. When the curvature of the occluding surface is large, a bigger λ would be more suitable.

In step 21, we define an intermediate position $x_{k+\frac{1}{2}}$. Motivation for this additional step is to have a homogeneous coverage of the obstacles' boundaries. By keeping the observer a uniform distance away from the boundary, we are able to obtain the same level of detail everywhere in the region. Additionally, the step $x_{k+\frac{1}{2}}$ is motivated by

Algorithm 1 Single observer

```
1:  $k = 0$
2:  $L$: list of unexplored edges, initially empty
3:  repeat
4:      $x_k$: vantage point outside the occluding objects
5:      $\rho_{x_k}$: visibility function corresponding to $x_k$
6:      update the map $\Xi$ of the explored region {$\Xi$ was defined in (2.3)}
7:      find all the edges (horizons) on the $\big(\theta, \rho_{x_k}(\theta)\big)$ map
8:      if an edge is found then
9:          choose the edge to approach, say, in the direction $\theta_e$ {choice depends on
            particular aspects of the problem and will be discussed below}
10:         store the rest of the edges in a list $L$
11:         remove those edges from $L$ that are currently visible
12:     else {no edges found}
13:         pick an edge $\theta_e$ from the list of unexplored edges $L$
14:         backtrack $x_k$ to one of the previous positions corresponding to the chosen
            edge
15:     end if
16:     if $\rho_{x_k}(\theta_e) < \rho_{x_k}(\theta_e + \delta)$ then
17:         choose the direction $\Theta = \theta_e + \delta$
18:     else
19:         choose the direction $\Theta = \theta_e - \delta$
20:     end if{here $\delta$ is chosen so that there is always a buffer of size $\lambda$ between the
        observer and any obstacle boundaries. For example, $\lambda$ may depend on the
        robot's size to avoid collisions.}
21:     $x_{k+\frac{1}{2}}$ is obtained by moving $x_k$ in the chosen direction $\Theta$ by amount

            $$r_1 = \min\{\rho_{x_k}(\Theta), \rho_{x_k}(\theta_e) - \tan\left(\frac{\pi}{3}\right)\frac{1}{\kappa}\}$$

        {$\kappa$ is the curvature near the edge (if $\kappa = 0$ shift $x_k$ by small amount to see the
        next edge)}
22:     $\rho_{x_{k+\frac{1}{2}}}$: visibility function corresponding to $x_{k+\frac{1}{2}}$
23:     update the map $\Xi$ of the explored region
24:     remove those edges from $L$ that are currently visible
25:     $x_{k+1}$ is obtained by overshooting from $x_{k+\frac{1}{2}}$ by

            $$r_2 = \min\{\rho_{x_k}(\Theta) - r_1, 2\tan\left(\frac{\pi}{3}\right)\frac{1}{\kappa_e}\}$$

26: until $L \neq \{\emptyset\}$
27: have explored the entire environment
```

the convergence proof included in the Appendix C. An alternative algorithm is to introduce an intermediate position $x_{k+\frac{1}{2}}$ only when approaching a cast horizon. We use this modified version of Algorithm 1 to obtain all the results in this paper.

Now let us discuss the choice of horizon in step 9. In our experiments the observer approaches the nearest previously unexplored horizon. Intuitively, this choice would minimize the length of the path. The convergence proof in Appendix C is based on the nearest edge approach. However, other choices may be more applicable under different circumstances. For example, one may choose to approach a random horizon as was done in [19] or a horizon with the largest curvature κ (so that the overshoot step size is the smallest). In [12], the choice of the next horizon is dictated by the specifics of sensor design: to minimize the errors produced by the sensors in the experiments, it is always preferable to navigate around the objects in the counterclockwise fashion. Thus, the observer always prefers to approach the right-most edge of the obstacle.

3.2. Multiple observers. The extension of the navigation algorithm for multiple observers is straightforward. Let $\{x_j\}_{j=1}^{n}$ be a set of observing locations. Similar to (2.3), define the visibility indicator $\Xi_j(y) := \rho_{x_j}\big(\nu(y)\big) - |y - x_j|$, such that $\{\Xi_j \geq 0\}$ is the set of visible regions and $\{\Xi_j < 0\}$ is the set of invisible regions from x_j. In addition, let $\Theta_j = \{\theta_{j,1}, \ldots, \theta_{j,k}\}$ be the set of edges visible from the vantage point x_j. The algorithm for multiple observers is as follows.

Algorithm 2 Multiple observers (based on Algorithm 1)

```
 1: $N$: number of observers
 2: $x_j$: vantage points outside the occluding objects, $j = 1, ..., N$
 3: $\rho_{x_j}$: visibility function corresponding to $x_j$
 4: compute $\Xi = \max_j\{\Xi_j\}$
 5: find all the edges (horizons) $\Theta_j$ corresponding to each observer $x_j$
 6: exclude those $\theta_{j,k}$ for which $\Xi \geq 0$
 7: if found an edge then
 8:     proceed as in Algorithm 1 for each individual observer
 9: else {no edges found}
10:     move observer at $x_j$ in the direction orthogonal to the direction of the nearest
          $x_i$ to see new edges;
11: end if
12: proceed as in Algorithm 1 until no more new edges
13: have explored the entire environment
```

Note that in step 6 of Algorithm 2 we exclude those edges corresponding to x_j, which are visible by another observer x_i and thus do not need to be further explored. The orthogonal move in step 10 is chosen to maximize the chance of "seeing" more new area.

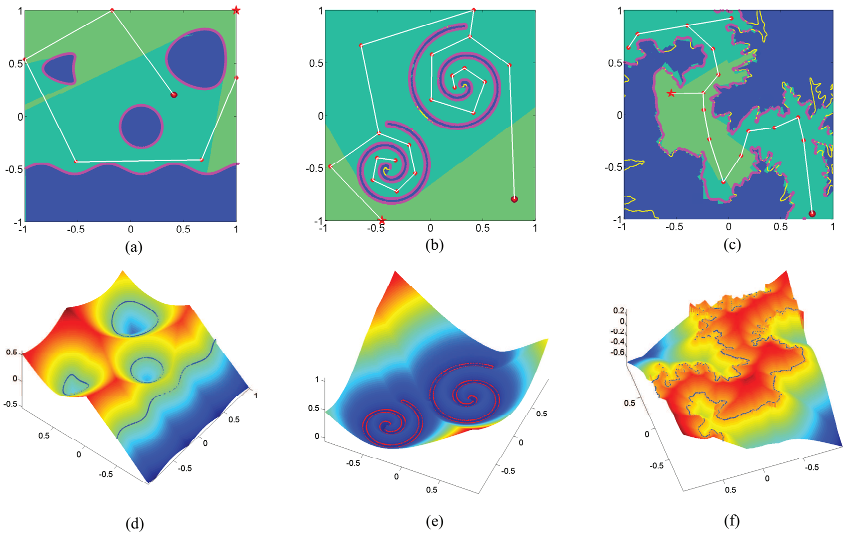
We would like to remark on different modes of execution of Algorithm 2. In *concurrent mode* all observers process sensor data simultaneously. This way, the next vantage point of each observer depends only on their previous positions. In *sequential mode* the observers are ordered as a sequence, and only one may move at a time. In this situation, the position of the next observer depends on new positions of the previous observers. The ordering may change according to the decision to optimize joint visibility. In some applications the concurrent mode would be more desirable since this mode allows for more autonomous maneuvering for each observer. Of course, in practice the usage of one mode or switching from one to the other depends on the data communication model as well as the routing algorithm. In the experiments in [12] the concurrent mode has been implemented.

3.3. Results. In this section we present some simulation results for environment exploration in case of single and multiple observers. We also present the statistics of some extensive simulations which demonstrate the stability of our algorithm in terms of step sizes and path lengths.

3.3.1. Single observer. In Figure 3.1 we demonstrate the paths generated using Algorithm 1 in different environments. In Figure 3.1 (a), the test environment consists of non-convex obstacles. The observer is able to explore the entire environment in seven discrete steps without having to backtrack and clear previously unexplored edges. In contrast, in Figures 3.1 (b) and (c), the observer must return to one of its previous locations to clear the unexplored edges. In the case of two spirals in Figure 3.1 (b), the observer first explores the interior of one spiral, then returns to proceed inside the second spiral.

The environment in Figure 3.1 (c) is generated by taking a level set of a portion of the Grand Canyon elevation data.[2] The non-uniform change of curvature of the visible boundary makes it difficult to explore fractal-like portions of the boundary. A constant parameter λ, which controls how close an observer may approach an obstacle, does not allow for a more detailed exploration of the narrow regions. As this parameter may depend on the physical size of the observer, the result is an illustration of a realistic exploration outcome. If the size and mobility of the observer allows for it, an algorithm with λ varying as a function of curvature may be implemented. Another approach would be to utilize optimization techniques to construct a new path based on the results of the initial "rough" exploration as in Section 4.

The second row in Figure 3.1 depicts the level set representations based on the reconstructed environments from the first row. The zero level set corresponding to the boundaries of the obstacles is marked on each figure.



(a) (b) (c)

(d) (e) (f)

Fig. 3.1. *Top row: results of exploration algorithm with a single observer (dark circle – initial observer position, star – final observer position, white line with circles – observer's path); bottom row: implicit level set reconstruction of the explored regions, zero level set corresponds to the objects' boundaries.*

[2] The terrain data was obtained from `ftp://ftp.research.microsoft.com/users/hhoppe/data/gcanyon/`.

3.3.2. Statistics. Figure 3.3 depicts the statistics of environment exploration simulations using Algorithm 1 and simulations using a random walk strategy, which serve as control experiments. In the random walk strategy, the next position of the observer is randomly chosen from the currently visible region. We compare the distributions of the number of steps required to explore a sample environment containing twelve nonconvex disjoint objects. The statistics are collected from 1000 independent runs with random initial positions of the observer and non-overlapping random placement of the obstacles. A sample obstacle configuration is illustrated in Figure 3.2. The initial positions of the observer are constrained to be outside of the obstacles.

Histograms of the total number of steps required to explore the environment are presented in Figures 3.3 (a) and (b). Figure 3.3 (a) contains the closeup version of the histogram corresponding to the nearest edge approach (Figure 3.3 (c)). From it, one can see that Algorithm 1 requires no more than 26 steps to explore the environment. Furthermore, the exploration most frequently terminates in 19 or 20 steps. The minimum number of steps required to explore this type of environment is 11. We would like to remark that the corresponding path lengths can be inferred taking into account the size of the exploration domain.

In the simulations using the random walk strategy described above, a limit of 400 steps is imposed, regardless of whether the environment has been entirely explored or not. One can see that about a quarter of experiments terminate before the entire region has been explored.

Note that if the curvature changes its sign $2m$ times along the boundary of a single star-shaped obstacle, the number of steps required to explore the entire boundary is $3+m$. Here, 3 steps are needed to see the convex hull of the object and m steps are required to explore each concave part. The complexity estimates for our algorithm are discussed in detail in Appendix C. For now, we would like to remark that if each object is treated independently, the number of steps required to explore a region with n disjoint star-shaped obstacles is $n(3+m)$. For our particular experiment setup, this amounts to 48 steps. Note that Algorithm 1 allows to explore the entire environment in half as many steps.

Clearly, Algorithm 1 provides a superior strategy for environment exploration compared to the random walk strategy. The statistics also provide an estimate on path length and the number of steps required to explore the region with twelve nonconvex obstacles. In Appendix C we prove that our proposed algorithm would always terminate in finite number of steps in any bounded region containing an arbitrary finite number of disjoint convex obstacles.

3.3.3. Multiple observers. In Figures 3.4, 3.5, and 3.6 we illustrate environment exploration with multiple observers in concurrent mode according to Algorithm 2. The experiment with two robots navigating in the environment in sequential mode is described in [12]. The two observers in Figure 3.4 are initially positioned so that Observer 1, which is closer to the obstacle, does not see any new horizons that are invisible to Observer 2, which is farther away. In this situation, Observer 1 makes a move in the direction orthogonal to Observer 2, according to step 4 of Algorithm 2. Thus it takes three steps by each of the two observers to explore the region with a single obstacle.

In Figure 3.5 we have two observers in a more complex environment consisting of three nonconvex shapes. Two observers explore such an environment in four steps. Finally, in Figure 3.6, we have three observers in the environment with four circles. This time it takes only three steps to complete the exploration.

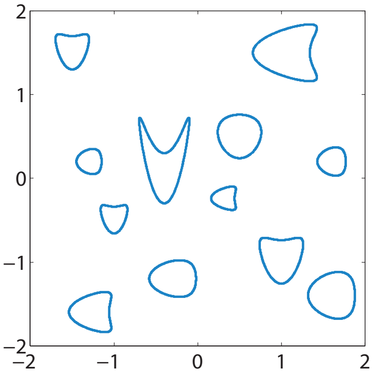


Fig. 3.2. *Sample environment used in statistics experiments using Algorithm 1.*

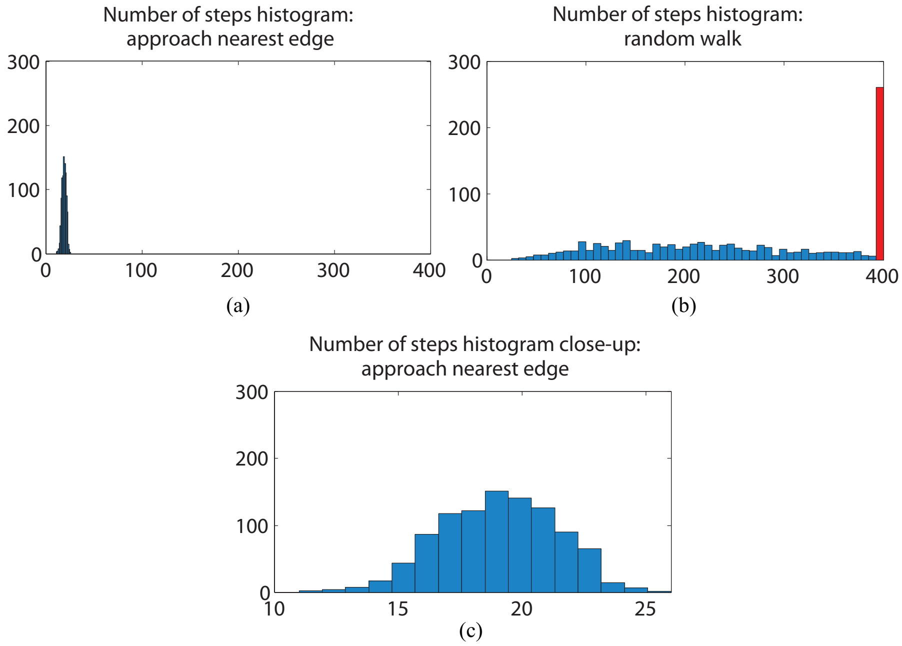


Fig. 3.3. *Statistics experiment using Algorithm 1: (a) Number of steps histogram for the nearest edge approach; (b) number of steps histogram for the random walk; (c) closeup of the number of steps histogram for the nearest edge approach. The random walk simulation is terminated if the step count is greater than 400.*

3.3.4. Target finding. To demonstrate the versatility of the proposed Algorithm 1, we show how it can be applied towards the problem of searching for a target in an unknown environment. Assume the target coordinates are given. The observer is equipped with a range sensor and must find a path through the unknown environment, so that at the termination of the path, the target is visible. If desired, the observer can then march along the line of sight connecting it to the target to reach it.

The proposed discrete strategy is to approach the nearest edge to the target at

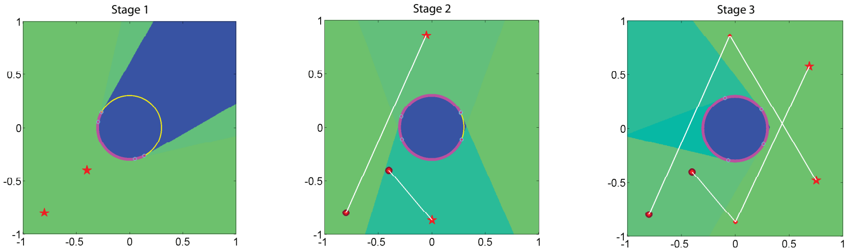


Fig. 3.4. *Stages of environment exploration with two observers. Red stars – current observers' position.*

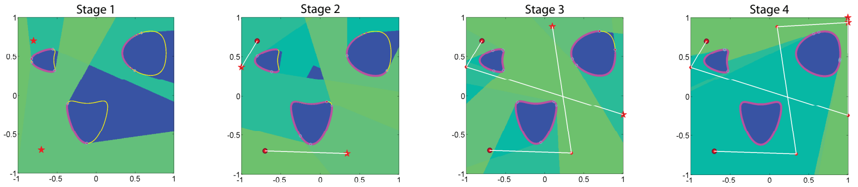


Fig. 3.5. *Stages of environment exploration with two observers. Red stars – current observers' position.*

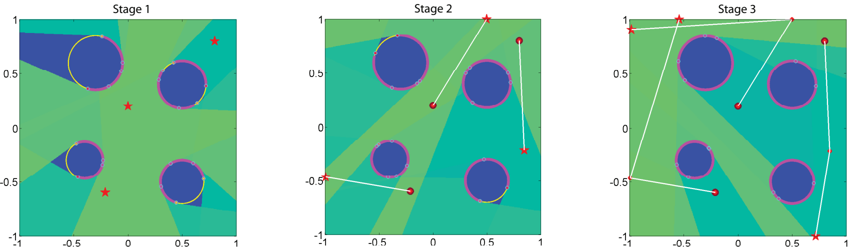
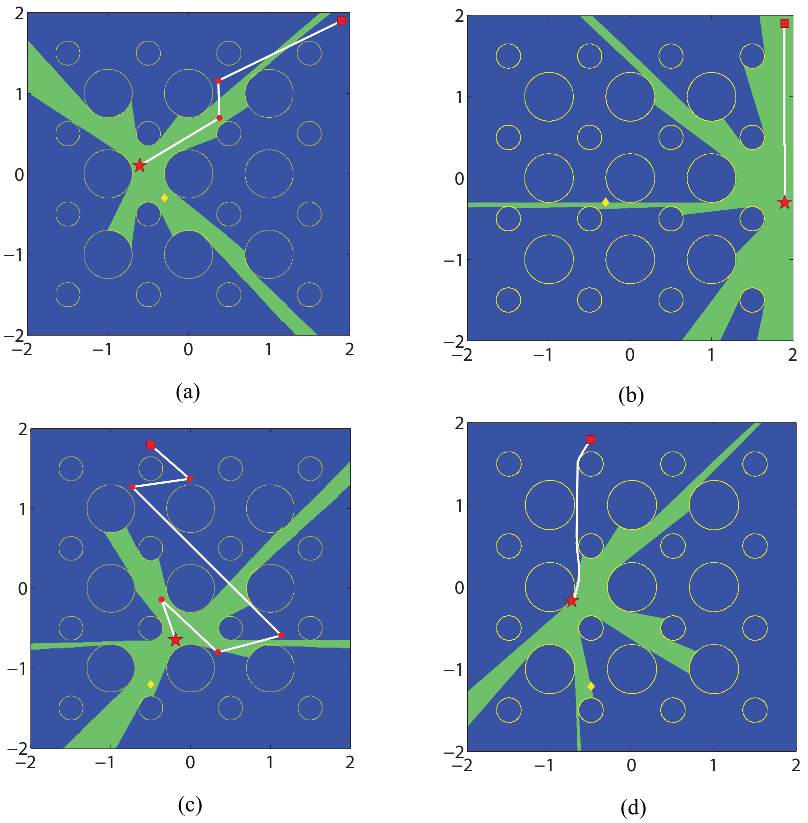


Fig. 3.6. *Stages of environment exploration with three observers. Red stars – current observers' position.*

every step. Figure 3.7 illustrates sample paths to the target (marked by the yellow diamond) in an environment containing multiple disk-shaped obstacles. The initial position of the observer is marked by the red square and the final position by the star. The green regions are visible from the final position of the observer. Figures 3.7 (a) and (c) illustrate the paths resulting from the proposed algorithm, while Figures 3.7 (b) and (d) depict the optimal paths to see the target, computed using the algorithm for known environments introduced in [3].

The length of the path in Figure 3.7 (a) is 2.6215, while the optimal path length in the current configuration is 2.2033. The second row depicts a slightly different observer-target configuration, with the resulting path length 6.0557 and the corresponding optimal path length 2.0349.

A forthcoming publication will address application of visibility to other types of target finding problems, such as looking for diffusive or wave sources in known and unknown environments.



(a)

(b)

(c)

(d)

Fig. 3.7. *(a) and (c): Target finding in an unknown environment. Yellow diamond – target, red square – initial observer's position, red circles – discrete observers' positions along the path, red star – final position. The green region is currently visible to the observer. (b) and (d): Optimal paths to see the target, computed using [3]. Path lengths: (a) 2.6215, (b) 2.2033, (c) 6.0557, (d) 2.0349.*

4. Postprocessing of the path: illumination optimization

Once we have constructed a route to explore an unknown environment, we can apply optimization techniques to post-process the obtained path in order to obtain a more uniform illumination/exposure of the explored region. In what follows, let Ω be the region of exploration and D be the obstacles. Then the region discovered along the path is $\Omega \backslash D$.

Let $\gamma = \{z_0, z_1, \ldots, z_N\}$ be the positions returned by the exploration algorithm. Let $\phi(\cdot; z_k)$ be the visibility level set function corresponding to observer at z_k:

$$\begin{cases} \phi(x; z_k) > 0, \text{ if } x \text{ is visible from } z_k, \\ \phi(x; z_k) < 0, \text{ if } x \text{ is invisible from } z_k, \\ \phi(x; z_k) = 0, \text{ if } x \text{ is on the shadow boundary with respect to } z_k. \end{cases} \tag{4.1}$$

Define the total illumination of a point $x\in\Omega\backslash D$ due to γ by

$$I(x;\gamma):=\sum_{k=0}^{N}H(\phi(x;z_k)), \tag{4.2}$$

where H is the one-dimensional Heaviside function. Note that I records the number of observers that can view x, and so $0\le I(x;\gamma)\le N+1$.

From a given set of observing locations $\{z_k\}$, we seek a set of nearby $\{z_k^*\}$, so that on average points outside obstacles are viewed in a more uniform manner – the deviation of illumination from a prescribed illumination level is small.

In certain applications, a higher priority may be placed on viewing a specific region in space, while lower priority is placed on other regions. In our formulation we simulate this effect through the use of weights. Let $w:\Omega\to\mathbb{R}^+$ be a positive real-valued function defined over Ω. We relate the magnitude of w to the importance of a given point $x\in\Omega$ to be visible, with larger magnitude associated with greater importance. By including w in the measure used in spatial integration, we attach importance weights to the visibility of space.

More precisely, we formulate the variational problem as follows:

Problem 4.1. *Given a positive constant C and a weight function $w(x)$, find $\gamma\in\mathbb{R}^{2N}$ that minimizes*

$$E(\gamma;C)=\frac{1}{2}\sum_{k=0}^{N-1}|z_{k+1}-z_k|^2+\frac{\lambda}{2}\int_{\Omega\backslash D}\big(I(x;\gamma)-C\big)^2w(x)dx \\ +\mu\sum_{k=0}^{N-1}\big(|z_{k+1}-z_k|-\rho_{z_k}(\theta)\big). \tag{4.3}$$

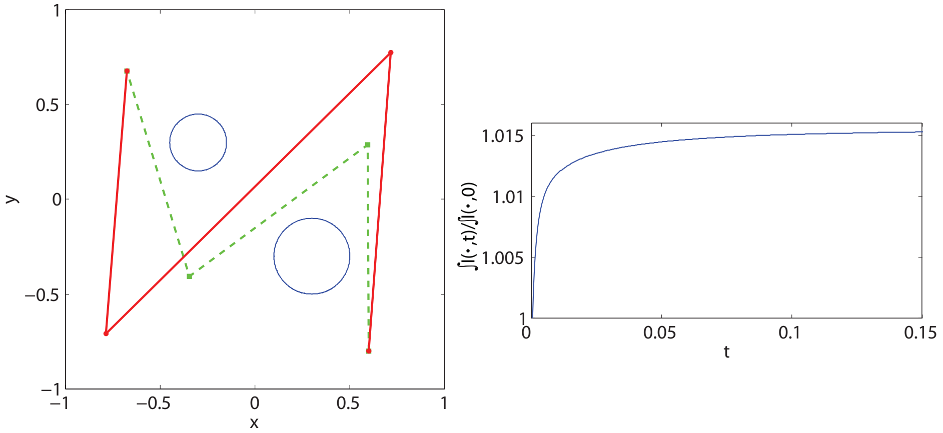
The first term in the above functional seeks to stabilize the problem, by penalizing against fractal or space filling paths. The constant C in the second term is related to the desired exposure of the region. Thus, large C causes the largest possible number of observers to see any given point $x\in\Omega\backslash D$, taking into account other constraints. More discussion on the choice of C can be found in [2]. In the meantime, the last term in (4.3) prevents the continuous path connecting the discrete locations in γ from accidentally crossing the obstacles' boundary. If θ is the direction of z_{k+1} when looking out of z_k, the boundary non-crossing condition is equivalent to keeping $|z_{k+1}-z_k|<\rho(\theta)$ for all k. The coefficients λ and μ serve as parameters for the penalty terms.

Using summation by parts and fixing z_0 and z_N, we arrive at the following Euler-Lagrange equation:

$$\begin{aligned}\dot{z_k}=&(z_{k+1}-2z_k+z_{k-1})\\ &-\lambda\int_{\Omega\backslash D}\Big(\sum_{j=0}^{N}H\big(\phi(x;z_j)\big)-C\Big)\delta\big(\phi(x;z_k)\big)\nabla_{z_k}\phi(x;z_k)w(x)dx\\ &-\mu\Big[\Big(\frac{z_{k+1}-z_k}{|z_{k+1}-z_k|}-\frac{z_k-z_{k-1}}{|z_k-z_{k-1}|}\Big)-\nabla_{z_k}\rho(\theta)\Big],\ 1\le k\le N-1.\end{aligned} \tag{4.4}$$

Using the path γ_0 constructed via Algorithm 1 as an initial guess, Equation (4.4) can be solved by simple integration techniques.

Unless stated otherwise, in the following discussion we take $w\equiv 1$. In Figure 4.1 (a) we have two circular objects. The dashed green line segments join the four vantage points forming initial exploration path γ_0 obtained via Algorithm 1. This path is then deformed using the flow (4.4), resulting in a new path γ represented in solid red. To maximize the total illumination of the region we set constant $C=15$, so that the desired exposure is always greater then maximum possible $C=4$. The flow eventually reaches a steady state. In Figure 4.1 (b) we plot the ratio $\int_{\Omega\setminus D} I(x;\gamma;t)dx / \int_{\Omega\setminus D} I(x;\gamma;0)dx$ of total exposure at time t to the initial total exposure. One can see that the total increase in the exposure for this simple geometry is only roughly 1%.



(a) (b)

Fig. 4.1. *Path postprocessing corresponding to Problem 4.1, obstacles – two circles (blue curves): (a) initial path (dashed green) and optimized path (solid red); (b) $\int_{\Omega\setminus D} I(x;\gamma;t)dx / \int_{\Omega\setminus D} I(x;\gamma;0)dx$. Here $C=15, \lambda=0.1, \mu=1$.*

In contrast to the previous example, the gain in exposure in our second experiment is around 23%, see Figure 4.2 (b). Here the region is constructed from a slice of Grand Canyon elevation data, which has a much more complex geometrical structure compared to the example with two circles. We further increase the complexity of the Grand Canyon terrain by adding holes shaped as circles to the interior of the explored region. The initial and optimized paths are depicted in Figure 4.2 (a). The original exploration path (dashed green) branches out to explore the regions occluded by the circles. Note that the optimized path (solid red) is shorter than the original and has fewer kinks. Here, we again choose the desired exposure constant $C=20$.

Figure 4.3 shows the evolution of the path along the Grand Canyon terrain and the resulting exposure of the region where the Gaussian importance weights are centered at (0.9,0) and (−0.5,0.25). We see an increase of about 30% in total illumination of the region in Figure 4.3 (b). The resulting optimized path (solid red) in Figure 4.3 (a) is shorter than the original path (dashed green), with the observers' positions concentrated near the regions of increased importance (magenta diamonds).

It may be desirable in some applications to increase the number of observing locations along the path. Our goal is to arrange the new observers in an optimal way with respect to total illumination of the region. Starting with an initial path γ obtained by our algorithm, we can insert points along the line segments connecting z_k and z_{k+1} for each k to obtain a new set $\tilde{\gamma}=\{Z_0,Z_1,\ldots,Z_M\}$, such that $\gamma\subset\tilde{\gamma}$. Given

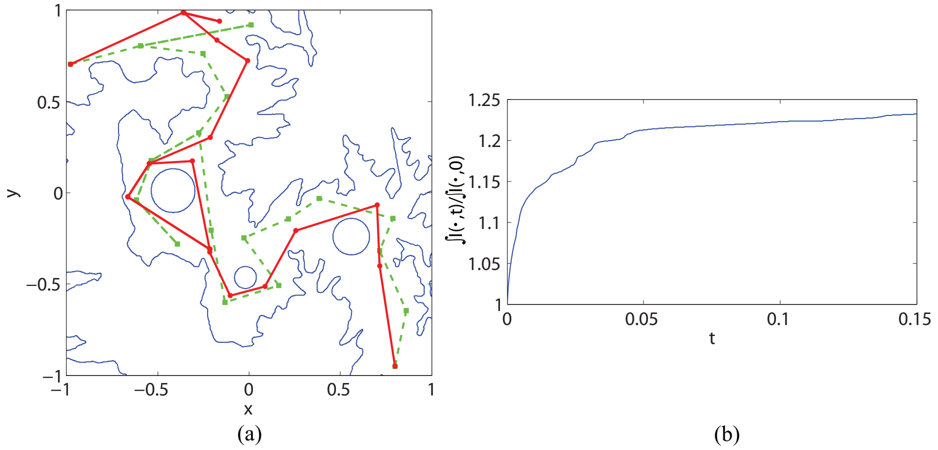


Fig. 4.2. *Path postprocessing corresponding to Problem 4.1, obstacles – Grand Canyon terrain (blue curves): (a) initial path (dashed green) and optimized path (solid red); (b)* $\int_{\Omega\setminus D} I(x;\gamma;t)dx / \int_{\Omega\setminus D} I(x;\gamma;0)dx$. *Here* $C=20, \lambda=0.1, \mu=1$.

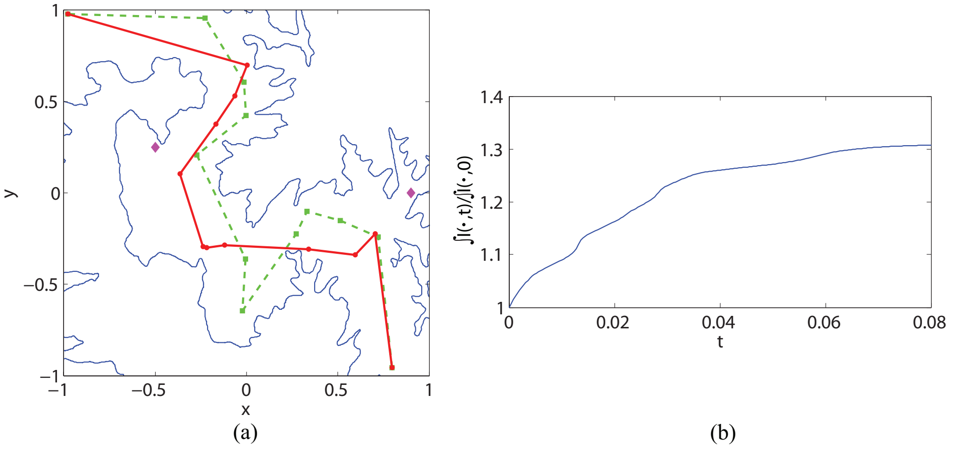


Fig. 4.3. *Path postprocessing corresponding to Problem 4.1, obstacles – Grand Canyon terrain (blue curves), the weights are centered at* $(0.9,0)$ *and* $(-0.5,0.25)$ *(magenta diamonds); (a) initial path (dashed green) and optimized path (solid red); (b)* $\int_{\Omega\setminus D} I(x;\gamma;t)dx / \int_{\Omega\setminus D} I(x;\gamma;0)dx$. *Here* $C=20, \lambda=0.1, \mu=1$.

a parameter $ds>0$, we optimize the positions of thus obtained vantage points with additional constraint $|Z_{k+1}-Z_k|=ds$. Precise variational formulation of the problem is provided below:

Problem 4.2. *Given* $\{z_k\}_{k=0}^{N}$*, a constant* $C>0$*, and a parameter* $ds>0$*, find* $\tilde{\gamma}\in\mathbb{R}^{2M}$

that minimizes

$$
\begin{aligned}
E(\tilde{\gamma};C) = &\frac{1}{2}\sum_{k=0}^{M-1}(|Z_{k+1}-Z_k|-ds)^2 \\
&+\frac{\lambda}{2}\int_{\Omega\setminus D}\big(I(x;\tilde{\gamma})-C\big)^2dx \\
&+\mu\sum_{k=0}^{M-1}\big(|Z_{k+1}-Z_k|-\rho_{Z_k}(\theta)\big).
\end{aligned} \tag{4.5}
$$

Similar to Problem 4.1, the first term in the above functional acts as a regularizer of the path. By setting ds to be the actual arc length along the path, we additionally enforce the uniform distribution of the observing locations along the path. The Euler-Lagrange equation corresponding to Problem 4.2 is

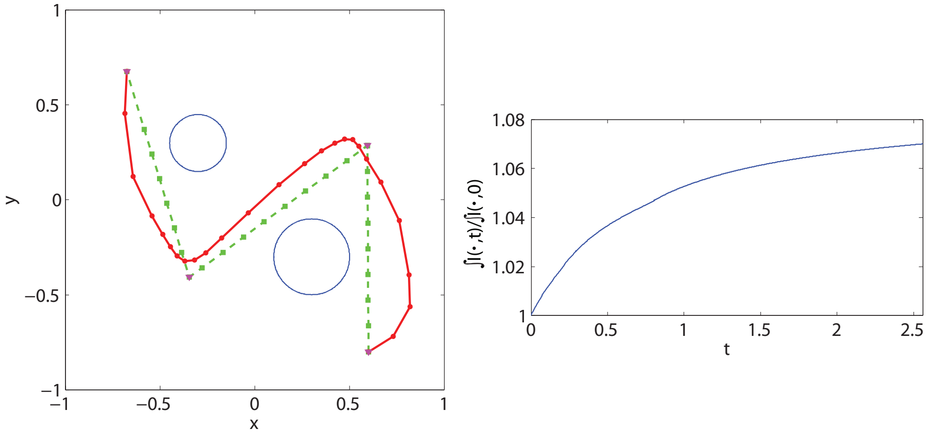
$$
\begin{aligned}
\dot{Z}_k = &\Big(Z_{k+1}-2Z_k+Z_{k-1}+ds\Big[\frac{Z_k-Z_{k-1}}{|Z_k-Z_{k-1}|}-\frac{Z_{k+1}-Z_k}{|Z_{k+1}-Z_k|}\Big]\Big) \\
&-\lambda\int_{\Omega\setminus D}\Big(\sum_{j=0}^{M}H\big(\phi(x;Z_j)\big)-C\Big)\delta\big(\phi(x;Z_k)\big)\nabla_{Z_k}\phi(x;Z_k)dx \\
&-\mu\Big[\Big(\frac{Z_{k+1}-Z_k}{|Z_{k+1}-Z_k|}-\frac{Z_k-Z_{k-1}}{|Z_k-Z_{k-1}|}\Big)-\nabla_{Z_k}\rho(\theta)\Big],\ 1\le k\le M-1.
\end{aligned} \tag{4.6}
$$

In Figures 4.4 and 4.5 we present the results of exploration according to Problem 4.2. In case of two circles, the path expands away from the obstacles' boundaries which provides better illumination of the region. The total gain in illumination as a result of the flow (4.6) is about 7%. The path, in case of Grand Canyon, clearly smoothes out and contracts as a result of postprocessing as can be seen from Figure 4.5 (a). Note that in order to have a continuous path, the observer has to backtrack in places where the path branches out. The exposure of the region keeps increasing with the total gain slightly under 15%. We remark that more complex environments, *e.g.*, the Grand Canyon, allow for greater improvement in illumination through postprocessing of the path as opposed to simpler environments like the one with two circles.

Other types of visibility optimization problems are considered in [3].

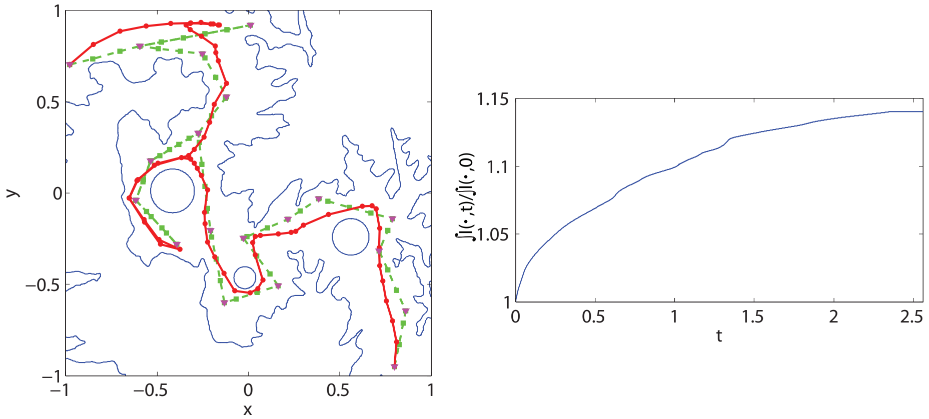
Appendix A. Error analysis. In this section we discuss the accuracy of the visibility function resulting from the projection method that uses ENO interpolation. For simplicity, we consider sample environments containing a finite number of disjoint strictly convex objects. The observer is positioned outside the obstacles. We demonstrate how the error relates to the distance from the observer, the view direction, and the size of the fan $\delta\theta$. In particular, we demonstrate how the quality of interpolation deteriorates as the view direction becomes orthogonal to the outer surface normal near the horizon locations.

Without loss of generality, we may assume that there are no partially occluded objects. The analysis for the case of partially occluded objects is a straightforward generalization. Due to the convexity assumption, there are exactly two horizons corresponding to each object in the scene, as illustrated in Figure A.1. Let the horizon locations correspond to θ_L and θ_R. Suppose $\Theta := \{\theta_i \in [\theta_L,\theta_R], i=0,\cdots,\nu\} \subset [-\pi,\pi)$ are distinct angles sorted in increasing order. The angles in Θ correspond to view



(a) (b)

Fig. 4.4. *Path postprocessing corresponding to Problem 4.2. Obstacles – two circles (blue curves): (a) initial path (dashed green), four original observers' locations (magenta triangles), and optimized path (solid red); (b)* $\int_{\Omega\setminus D} I(x;\gamma;t)dx/\int_{\Omega\setminus D} I(x;\gamma;0)dx$. *Here* $C=100, \lambda=0.001, \mu=1, ds=0.01$, *and the total number of steps along the path is 26.*



(a) (b)

Fig. 4.5. *Path postprocessing corresponding to Problem 4.2. Obstacles – Grand Canyon terrain (blue curves): (a) initial path (dashed green), original observers' locations (magenta triangles), and optimized path (solid red); (b)* $\int_{\Omega\setminus D} I(x;\gamma;t)dx/\int_{\Omega\setminus D} I(x;\gamma;0)dx$. *Here* $C=150, \lambda=0.001, \mu=1, ds=0.01$, *and the total number of steps along the path is 71.*

directions from x_0 to points $p_i \in \tilde{P}$ on a visible region of an occluding surface that is bounded by two horizons. These angles and points are obtained using the projection described in Section 2.1.

Assume the occluding surface between θ_L and θ_R is smooth enough, such that the visibility function $\rho(\theta) \in C^{n+1}(\theta_L, \theta_R)$. Then for each $\theta \in [\theta_L, \theta_R]$, we have the standard error estimate

$$\rho(\theta)=\rho^{\mathrm{ENO}\ n}(\theta)+\frac{\rho^{(n+1)}(\xi(\theta))}{(n+1)!}\Pi_{i=0}^{n}(\theta-\theta_i),\ \text{for some mean value}\ \xi(\theta)\in(\theta_L,\theta_R), \tag{A.1}$$

where $\rho^{\mathrm{ENO}\ n}(\theta)$ is the n-th order ENO polynomial approximation and

$$E^n:=\frac{\rho^{(n+1)}(\xi(\theta))}{(n+1)!}\Pi_{i=0}^{n}(\theta-\theta_i) \tag{A.2}$$

is the error term. Since ρ is smooth in $[\theta_L,\theta_R]$, depending on the order of approximation, the error term E^n is bounded according to the regularity of ρ. For example, with the third order ENO, for any $\theta\in[\theta_3,\theta_{\nu-4}]$, we have

$$\left|E^3\right|\leq\max_{\xi\in(\theta_3,\theta_{\nu-4})}\frac{\left|\rho^{(4)}(\xi)\right|}{4!}(2\delta\theta)^4.$$

Note that in order for the above bound to hold, we have to assume that ENO interpolation would not choose a stencil that goes across the discontinuities of ρ. Otherwise, if ENO stencil includes the jump location, the remainder term (A.2) can be very big. In order to avoid this problem we introduce the following assumption on the size of the fan used in filtering:

Assumption A.1. *$\delta\theta$ is small enough, so that $g(\rho(\theta))<\delta\theta$ implies there is a discontinuity in the visibility function ρ at θ.*

Above, g is the edge-detector function defined in (2.4). Such $\delta\theta$ can always be found in the asymptotic limit. However, it may not always exist in practical applications, as can be seen from Figure 3.1 (c), where small concave regions of the Grand Canyon terrain are not fully resolved because the above assumption on $\delta\theta$ is not satisfied.

Furthermore, we require that $\delta\theta$ is small enough that the arc connecting p_i and p_{i+1} may be approximated by a straight line segment. This translates into the following assumption:

Assumption A.2. *$\delta\theta^2|\kappa|<\epsilon$ if $\kappa\neq0$. Here $0<\epsilon\ll1$ is a small constant.*

The above assumption can be easily derived using Taylor's expansion: for any $\theta\in[\theta_i,\theta_{i+1}]$ we can write $\rho(\theta)=\rho(\theta_i)+(\theta-\theta_i)\rho'(\theta_i)+\frac{1}{2}(\theta-\theta_i)^2\rho''(\xi)$ for some ξ between θ and θ_i. A linear approximation of ρ is obtained by setting the second order term in Taylor's approximation to 0. Precisely, we have $\delta\theta^2\kappa=0$, where κ is the curvature of the occluding surface. Then, Assumption A.2 follows.

Thus we have derived the two conditions on the size of the fan $\delta\theta$ which guarantee a bounded error term in the estimate (A.1).

Note that the accuracy of ENO polynomial approximation of ρ is not uniform along the occluding surface. It depends on the view direction and the proximity to the observer. We would like to find an upper bound on the derivatives of ρ, and thus obtain an expression for the remainder term E^n which relies on the properties of the projection and filtering method.

Without loss of generality, assume $|p_{i+1}-x_0|\geq|p_i-x_0|$. Denote the outer normal to the surface at p_{i+1} by $\vec{n}$ and let the angle between $\vec{n}$ and the view direction θ_{i+1} be

$\pi-\phi$. Let $M=\max_{p_i\in\tilde{P}}|x_0-p_i|$. Using simple trigonometry, we obtain the following bounds

$$|p_i-p_{i+1}|=\frac{\sin(\theta_{i+1}-\theta_i)\,|x_0-p_i|}{\cos\phi}\leq\frac{M\sin(2\delta\theta)}{\cos\phi}, \tag{A.3}$$

$$|\rho(\theta_{i+1})-\rho(\theta_i)|=\frac{\sin\left(\phi-\frac{\theta_{i+1}-\theta_i}{2}\right)|p_{i+1}-p_i|}{\cos\left(\frac{\theta_{i+1}-\theta_i}{2}\right)}\leq\frac{2M\tan\phi}{\cos\delta\theta}:=K. \tag{A.4}$$

From the above estimates we see that the shortest distance between the two neighboring sample points p_i and p_{i+1} as well as the smallest difference in their corresponding visibility values is obtained when $\phi=0$, *i.e.*, when the view direction is parallel to the outer normal $\vec{n}$. As ϕ approaches $\pi/2$, which happens near the horizon locations, both $|p_{i+1}-p_i|$ and $|\rho(\theta_{i+1})-\rho(\theta_i)|$ tend to infinity (see Figure A.1). Also, as $\delta\theta$ decreases to 0, the difference $|\rho(\theta_{i+1})-\rho(\theta_i)|$ decreases to $2M\tan\phi$. Denote the minimum distance to a point in $\tilde{P}$ by m. From the relation (A.3),

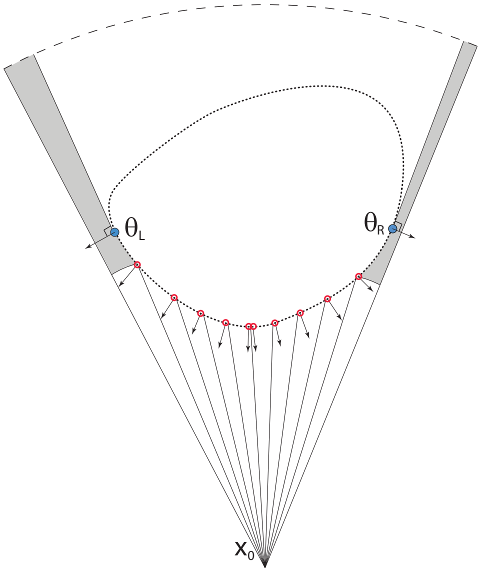


Fig. A.1. *Filtered out visible data $p_i\in\tilde{P}$ along with surface normals. Error in the approximation of horizon locations.*

$$|p_i-p_{i+1}|\geq\frac{m\sin(2\delta\theta)}{\cos\phi}$$

for any points p_i and p_{i+1} in $\tilde{P}$. Then

$$\theta_{i+1}-\theta_i\geq\psi:=2\sin^{-1}\left(\frac{m\sin(2\delta\theta)}{M\cos\phi}\right)$$

for any angles $\theta_i, \theta_{i+1} \in \Theta$. Using the estimate on divided differences

$$\rho[\theta_0, \theta_1, ..., \theta_n] = \frac{\rho^{(n)}(\xi)}{n!}. \tag{A.5}$$

and the relation (A.4), we obtain the following bounds on derivatives of the visibility function

$$\begin{aligned}
|\rho'(\xi)| &= \frac{|\rho(\theta_{i+1}) - \rho(\theta_i)|}{\theta_{i+1} - \theta_i} \leq \frac{1}{\psi} K, \\
\frac{|\rho''(\xi)|}{2!} &= \min\{|\rho[\theta_i, \theta_{i+1}, \theta_{i+2}]|, |\rho[\theta_{i-1}, \theta_i, \theta_{i+1}]|\} \leq \frac{2}{2!\psi^2} K, \\
\frac{\left|\rho^{(3)}(\xi)\right|}{3!} &\leq \frac{2^2}{3!\psi^3} K, \\
&\vdots \\
\frac{\left|\rho^{(n)}(\xi)\right|}{n!} &\leq \frac{2^{n-1}}{n!\psi^n} K.
\end{aligned} \tag{A.6}$$

Then the error term (A.2) can be bounded by

$$\begin{aligned}
|E^n| &= \left| \frac{\rho^{(n+1)}(\xi(\theta))}{(n+1)!} \Pi_{i=0}^n (\theta - \theta_i) \right| \\
&\leq \frac{2^n K \Pi_{i=0}^n |\theta - \theta_i|}{(n+1)!\psi^{n+1}} \\
&\leq \frac{2^n K (\theta_R - \theta_L)^{n+1}}{(n+1)!\psi^{n+1}}.
\end{aligned} \tag{A.7}$$

Furthermore, in a bounded domain $\Omega = B_R(x_0)$, we can estimate the error in the shadow boundary location near the horizon (as θ approaches $\pi/2$), *i.e.*, the area of the grey regions in Figure A.1. The maximum angle of the resulting fan is $\delta\theta$. Then the error in the shadow boundary corresponding to a given horizon is only linear

$$A_{\text{horizon}} \leq \frac{\delta\theta}{2}(R^2 - m^2). \tag{A.8}$$

Appendix B. Dynamics. Below we derive the dynamics equations of the visibility function and horizon points with respect to the moving vantage point. In two dimensions let us consider a coordinate system centered at x_0 with the visible portions of the occluding surfaces parameterized by polar coordinates. A point 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 $\theta = \varphi + \psi$, see

Figure B.1 (a). Then we can compute

$$\begin{aligned}\frac{d}{dt}(\rho^2) &= \lim_{\Delta t\to 0}\frac{\tilde{\rho}^2-\rho^2}{\Delta t}\\ &= \lim_{\Delta t\to 0}\frac{|z-(x_0+v\Delta t)|^2-|z-x_0|^2}{\Delta t}\\ &= -2v\cdot(z-x_0)\\ &= -2\rho v\cdot\begin{pmatrix}\cos\theta\\ \sin\theta\end{pmatrix}.\end{aligned} \tag{B.1}$$

On the other hand,

$$\frac{d}{dt}(\rho^2)=2\rho\frac{d\rho}{dt}=2\rho(\dot{\rho}+\rho_\theta\dot{\theta}). \tag{B.2}$$

Therefore,

$$\dot{\rho}+\rho_\theta\dot{\theta}=-v\cdot\begin{pmatrix}\cos\theta\\ \sin\theta\end{pmatrix}. \tag{B.3}$$

To find an expression for $\dot{\theta}$, note from Figure B.1 (a) that

$$\rho\sin\psi=\tilde{\rho}\sin\tilde{\psi}=L.$$

Since L is the distance from z to x_0+vt, it is independent of the motion of x_0 once the direction v is fixed. Therefore,

$$\frac{dL}{dt}=\frac{d\rho}{dt}\sin\psi+\dot{\psi}\rho\cos\psi=0. \tag{B.4}$$

Then

$$\dot{\theta}=\dot{\psi}=-\frac{\frac{d\rho}{dt}\sin\psi}{\rho\cos\psi}=\frac{v}{\rho}\cdot\begin{pmatrix}\sin\theta\\ -\cos\theta\end{pmatrix}. \tag{B.5}$$

Combining (B.5) with (B.3) we finally obtain

$$\dot{\rho}=-\frac{\rho_\theta}{\rho}v\cdot\begin{pmatrix}\sin\theta+\cos\theta\\ \sin\theta-\cos\theta\end{pmatrix}. \tag{B.6}$$

The above Equation (B.6) describes the change of the visible portion of the occluding surface, *i.e.* between the horizons. In order to have a complete description of the visibility we must derive the motion of horizons e_1 and e_2 on Figure B.1 (b) with respect to the observer.

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, the vector e_i-x_0 is tangent to the occluding surface at the horizon point. Without loss of generality, in all the computations below we will consider just e_1.

In the coordinate system defined as above, $\theta=\varphi+\psi$ 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.* $\dot{e_1}$.

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

$$\kappa=\frac{\rho^2+2\rho_\theta^2-\rho\rho_{\theta\theta}}{(\rho^2+\rho_\theta^2)^{\frac{3}{2}}}. \tag{B.7}$$

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

$$\begin{aligned} n^\perp(e_1)&=\frac{e_1-x_0}{|e_1-x_0|}\\ n(e_1)&=\Big(n^\perp(e_1)\Big)^\perp=\Big(\frac{e_1-x_0}{|e_1-x_0|}\Big)^\perp. \end{aligned} \tag{B.8}$$

Now we can plug in the above into the formula for horizon dynamics from [24] to get

$$\dot{e_1}=\frac{v\cdot n(e_1)}{\kappa\rho}n^\perp(e_1). \tag{B.9}$$

Therefore, from (B.6) and (B.9) 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 [24].

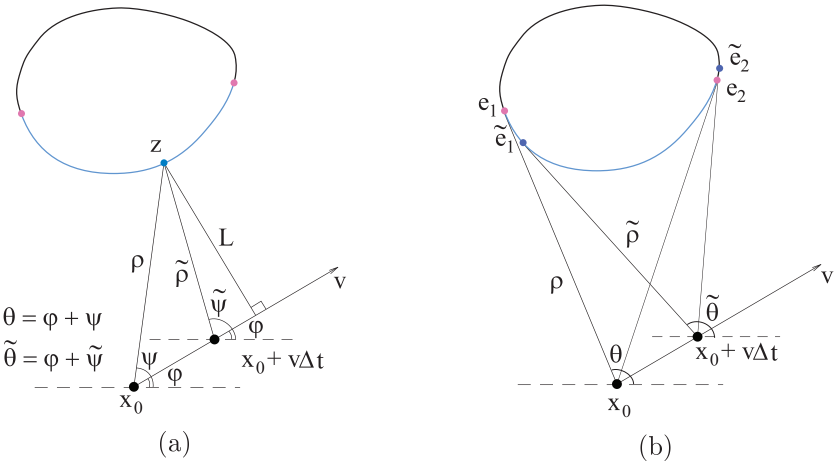


FIG. B.1. *Derivation of the dynamics equations for the visibility function (a) and the horizons (edges) (b).*

Appendix C. Complexity estimates for 2D cases. Let us begin by considering a single obstacle Ω bounded by a simple C^2 regular convex curve Γ.

PROPOSITION C.1. *The Gauss map $S:\Gamma\mapsto\mathcal{S}^1$ is monotone.*

Proof. Note that a simple closed C^2 regular curve is convex if and only if its signed curvature κ does not change its sign, in particular, if it is never zero. But the curvature κ is the derivative of the tangent vector parameterized by the arc length of a given curve. Since the normal vector is just a tangent vector rotated by $\frac{\pi}{2}$ we conclude that the angle of the normal vector should be monotone along the curve, otherwise κ would change its sign. Thus the Gauss map of Γ is monotone. □

Next we are going to use monotonicity of the Gauss Map to construct a path for the observer to see whole region, *i.e.* the boundary of the obstacle Ω.

Claim C.2. *At least three steps are required to explore a simply connected region containing a single convex obstacle.*

Proof. Let x_0 be the center of mass of Ω. Since the Gauss map $S:\Gamma\mapsto\mathcal{S}^1$ is monotone, "seeing" Γ is equivalent to seeing the boundary of a disk C centered at x_0 that encloses Ω. Let z_0,z_1, and z_2 be the vertices of triangle that encloses C and is tangent to C at e_0,e_1, and e_2. The observer placed at z_0 is able to see the portion of the boundary of C between e_0 and e_1. Similarly, from z_1 the observer is able to see an arc between e_1 and e_2, and from z_2 the remaining portion of the boundary of C and, correspondingly, Γ. Since the observer's path must be finite, we exclude the case of exploring the entire boundary in just two steps. □

We may assume that the observer may not approach the obstacle nearer than $\lambda>0$ and may not depart from the obstacle further than η. Then the minimum number of steps required to explore Γ is $\lfloor \frac{\pi}{\cos^{-1}\frac{r+\lambda}{r+\eta}} \rfloor$. Here, r is the radius of the smallest circle enclosing Ω, centered at the center of mass of Ω (see Figure C.1).

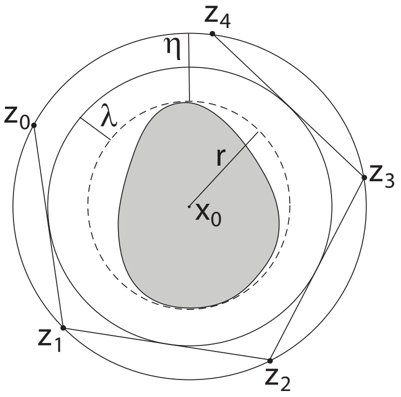


Fig. C.1. *Constructing a path around a convex obstacle under restrictions.*

Now we consider the scenario where the obstacles consist of finite number of disjoint, closed, strictly convex sets as in Figure C.2. Denote each convex component by Ω and denote by C the smallest disk centered at Ω's center of mass x_0 that encloses Ω. Let r be the radius of C. Let C' be the smallest disk that contains an equilateral triangle enclosing C. Let r' be the radius of C'. The length of a side of the equilateral triangle inscribed in C' is $L=2\sqrt{3}r=\sqrt{3}r'$. Let C_0 be the smallest circle of radius r_0 and corresponding triangle side length L_0. Let $R=\max_{j,k}\operatorname{dist}\{C'_j,C'_k\}$ be the largest distance between any two disks in $\{C_k\}$. Then all the disks must be contained in some bounded domain B_R.

Proposition C.3. *Start at z_0 on some $C'_1\supset C_1$ and overshoot the horizon e_1 by $\frac{L}{2}$ to arrive at z_1. Then will "see" the entire C_1, i.e. the remaining arc between the horizons e_0 and e_2 on Figure C.3, in finite number of steps.*

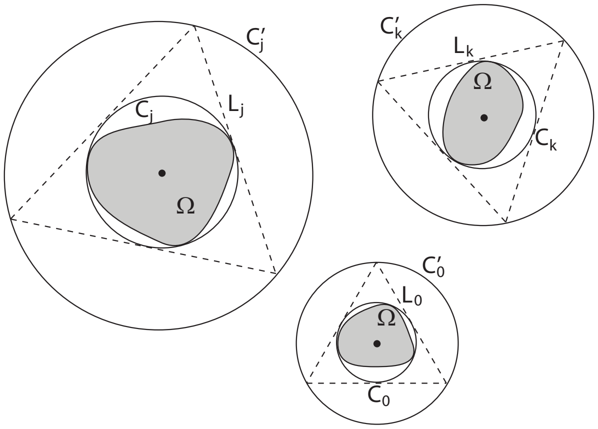


FIG. C.2. *Sample environment with closed, convex, disjoint sets. C_0 is the smallest disk possible.*

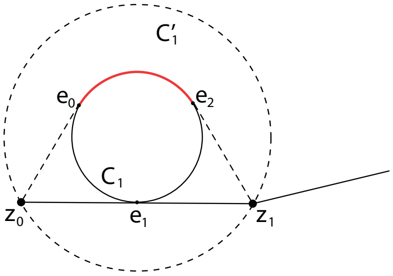


FIG. C.3. *Setup for Prop. C.1. The unexplored portion of C_1 is marked red.*

Proof. From z_1 an observer may

1. Proceed to see e_2, so that $z_2 \in C'_1$ and all of C_1 has been explored in just three steps: z_0, z_1, and z_2.
2. Proceed to some $e_3 \in C_2$ such that $\|z_1 - e_3\| \leq \|z_1 - e_2\|$. In this case z_2 is on some $C'_2 \neq C'_1$.

In case (2) the following is true:

CLAIM C.4. *If there exists $k \geq 1$, such that $z_{k+1} \in C'_1$ then C_1 is entirely seen from $z_0, z_1, z_{k+\frac{1}{2}}$ and z_{k+1}.*

Proof. If $z_{k+1} \in C'_1$ then there is a horizon $e_k \in (e_0, e_2)$ on C_1 which is the nearest to $z_k \in C'_j$ for some j. Let $z_{k+\frac{1}{2}} \in C'_1$ be the point of intersection of the ray (z_k, z_{k+1}) and C'_1. Thus, $[e_k, e_2]$ is entirely visible from $z_{k+\frac{1}{2}}$ and $[e_0, e_k]$ is entirely visible from z_{k+1}. Hence, the entire arc of C_1 between e_0 and e_2 has been explored. □

Suppose towards a contradiction that the observer does not return to C_1 at all. Since the collection $\{C_k\}$ is finite, the observer must be then stuck in a loop, *i.e.* there exist C_j and C_k such that C_j is approached from C_k infinitely many times. But this is impossible according to the following claim.

CLAIM C.5. *C_j may be approached from C_k at most twice.*

Proof. Note that given two disks, there exist four bitangents: ℓ_1, ℓ_2 and symmetric ℓ'_1 and ℓ'_2 as in Figure C.4. We will only consider ℓ_1 and ℓ_2 below unless we indicate otherwise. If there exists $C'_k \neq C'_j$ such that C'_k and C_j are bitangent, then there is a ray from the center of C'_k that is perpendicular to some θ as in Figure C.5. Then C'_k tangents the ray perpendicular to $\theta \pm \delta\theta$. By construction it is obvious that all possible return angles on C_j from $z_m \in C'_k$ are in $\theta \pm \delta\theta$, *i.e.* the observer may only approach horizons on C_j that lie on an arc of size $2\delta\theta$. Refer to Figure C.5 for an illustration.

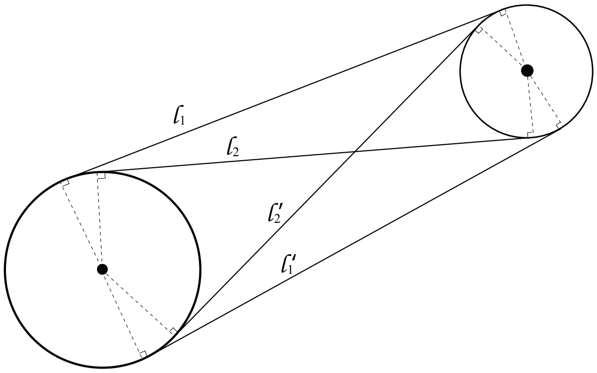


Fig. C.4. *Four bitangents to two disks.*

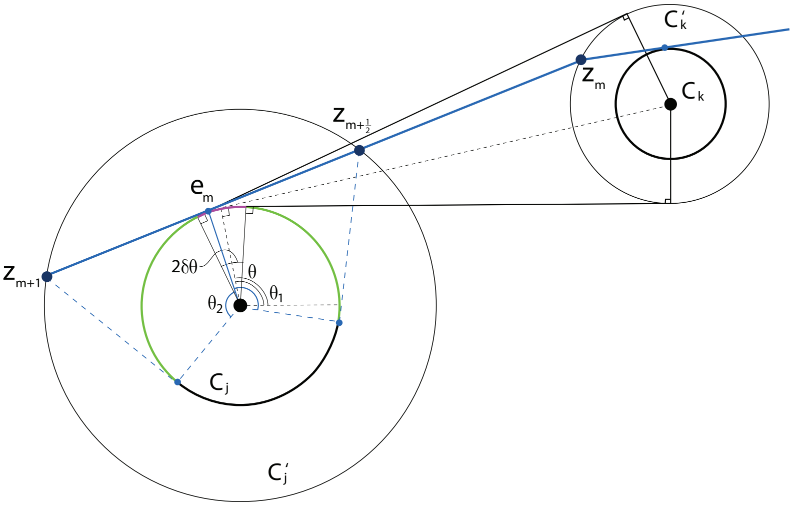


Fig. C.5. *Portions of C_j visible from $z_{m+\frac{1}{2}} \in C'_j$ and $z_{m+1} \in C'_j$.*

Let $z_{m+\frac{1}{2}} = (z_m, z_{m+1}) \cap C'_j$. Then let θ_1 be the angle visible from $z_{m+\frac{1}{2}} \in C'_j$ and θ_2 be the angle visible from $z_{m+1} \in C'_j$ depicted in Figure C.5. Note that $\theta_1 = \theta_2 = \frac{2\pi}{3}$. Hence, $2\delta\theta$ is entirely visible from $z_{m+\frac{1}{2}}$ and z_{m+1}.

Thus an observer is able to see the entire arc where possible horizons visible from C'_k are located in a single approach. The symmetrical case with another pair of bitangents provides the possibility for the second approach from C'_k. Hence, an observer may approach C_j from C'_k at most twice. □

We have shown that it is not possible for an observer to approach a single disk infinitely many times. Therefore an observer must return to C_1 in finite number of steps. This completes the proof of Prop. C.3. ∎

PROPOSITION C.6. *The entire environment B_R has been explored at the termination of the exploration algorithm. In other words, the observer has seen the boundary of every obstacle at the termination.*

Proof. We begin the proof with the definition:

DEFINITION C.1. *Two disks C_j and C_k are neighbors if there is no other obstacle in the region bounded by arcs of C_j' and C_k' between the outermost bitangents to C_j' and C_k' (ℓ_1 and ℓ_1' in Figure C.4).*

Then we make the following observations:

CLAIM C.7. *In the process of exploring C_j the observer must detect at least one horizon/edge on every neighbor of C_j.*

Proof. By the definition of neighbors there are no other objects obstructing the neighbors from each other. Assume, without loss of generality, that the observer visits vantage points z_0 and z_1 on C_j' during the exploration of C_j. Consider the lines ℓ_1,ℓ_2, and ℓ_3 which are tangent to C_j at the horizon points e_0,e_1, and e_2, as in Figure C.6. Then every neighbor of C_j must be in one of the half-planes $\Omega_1, \Omega_2, \Omega_3$, or their intersection. The observer at z_0 is able to see all of $\Omega_1 \cup \Omega_2$, whereas $\Omega_2 \cup \Omega_3$ is visible from z_1. Therefore, the observer is able to see at least one horizon/edge on each neighbor from just z_0 and z_1. ∎

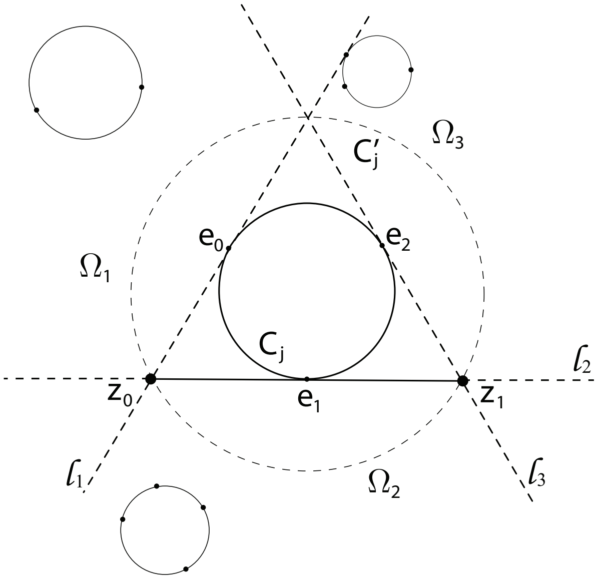


FIG. C.6. *Labeling horizons on the neighbors of C_j.*

Once a horizon has been labeled on C_k, the entire C_k will be seen completely later on, according to Prop. C.3.

Claim C.8. *Every disk in the given configuration will have at least one horizon/edge labeled on it at some stage of the algorithm.*

Proof. The following proof is by induction. Suppose we start the exploration at some disk C_1. Then all the neighbors $\{C_{1_j}\}_{j=1}^{M}$ of C_1 will have at least one edge marked on them according to Claim C.7. Suppose at some stage of the exploration, all the disks have at least one edge labeled on them but C_k. In the given configuration C_k has at least one neighbor C_{k_1}. By induction assumption, C_{k_1} also has some edges labeled on it. Then, at some point of the algorithm the observer must come to explore C_{k_1}. At that time it will label an edge on C_k, since C_k and C_{k_1} are neighbors. □

Hence, the entire environment will be explored at the termination of the algorithm. This completes the proof of Prop. C.6. □

Props. C.3 and C.6 imply that the Algorithm 1 will terminate in finitely many steps. At the termination of the algorithm the entire environment consisting of closed, disjoint, strictly convex sets will be explored.

Acknowledgement. The authors would like to thank Li-Tien Cheng for the very useful conversations on this topic.

REFERENCES

[1] G. Aubert and P. Kornprobst, *Mathematical Problems In Image Processing. Partial Differential Equations And The Calculus of Variations*, Springer, 2002.

[2] L.T. Cheng and Y.H. Tsai, *Redistancing by flow of time dependent eikonal equation*, UCLA CAM, 07-16, 2007.

[3] L.T. Cheng and Y.H.R. Tsai, *Visibility optimization using variational approaches*, Commun. Math. Sci., 3, 425-451, 2005.

[4] W.P. Chin and S. Ntafos, *Shortest watchman routes in simple polygons*, Discrete Computational Geometry, 6, 9-31, 1991.

[5] A. Buades, B. Coll and J. M Morel, *A review of image denoising algorithms, with a new one*, SIAM Multiscale Modeling and Simulation (MMS), 4(2), 490-530, 2005.

[6] M. Burger, G. Gilboa, S. Osher and J. Xu, *Nonlinear inverse scale Space methods*, Commun. Math. Sci., 4(1), 175-208, 2006.

[7] G. Gilboa and S. Osher, *Nonlocal linear image regularization and supervised segmentation*, SIAM Multiscale Modeling and Simulation (MMS), 6(2), 595-630, 2007.

[8] M. Ghomi, *Shadows and convexity of surfaces*, Annals of Mathematics, 155, 281-293, 2002.

[9] J.E. Goodman and J. O'Rourke, editors, *Handbook of discrete and computational geometry*, CRC Press LLC, Boca Raton, FL, second edition, April, 2004.

[10] A. Harten, B. Engquist, S. Osher and S.R. Chakravarthy, *Uniformly high order accurate essentially nonoscillatory schemes, III*, J. Comput. Phys., 71, 231-303, 1987.

[11] Y. Landa, *Visibility of point clouds and exploratory path planning in unknown environments*, Ph. D. thesis, 2008.

[12] Y. Landa, D. Galkowski, Y.R. Huang, A. Joshi, C. Lee, K.K. Leung, G. Malla, J. Treanor, V. Voroninski, A.L. Bertozzi and Y.H.R. Tsai, *Robotic path planning and visibility with limited sensor data*, Proc. American Control Conference, ACC 2007, 5425-5430, 2007.

[13] Y. Landa, R. Tsai and L.T. Cheng, *Visibility of point clouds and mapping of unknown environments*, Springer Lecture Notes, Computational Science and Engineering, 1014-1025, 2006.

[14] S.M. LaValle, *Planning Algorithms*, Cambridge University Press, 2006.

[15] D. Murray and C. Jennings, *Stereo vision based mapping for a mobile robot*, IEEE Conf., Robotics and Automation, 1997.

[16] L.I. Rudin, S. Osher and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D, 60, 259-268, 1992.

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

[18] D.M. Strong, J.F. Aujol and T.F. Chan, *Scale recognition, regularization parameter selection, and Meyer's G-norm in total variation regularization*, SIAM Multiscale Modeling and Simulation (MMS), 5(1), 273-303, 2006.

[19] B. Tovar, L. Guilamo and S.M. LaValle, *Gap navigation trees: minimal representation for visibility-based tasks*, Workshop on the Algorithmic Foundations of Robotics, 2004.

[20] B. Tovar, S.M. LaValle and R. Murrieta, *Optimal navigation and object finding without geometric maps or localization*, IEEE International Conference on Intelligent Robots and Systems, 2003.

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

[22] B. Tovar, S.M. LaValle and R. Murrieta, *Optimal navigation and object finding without geometric maps or localization*, IEEE International Conference on Robotics and Automation, 2003.

[23] B. Tovar, R. Murrieta-Cid and S.M. LaValle, *Distance-optimal navigation in an unknown environment without sensing distances*, IEEE Transactions on Robotics, 2007.

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

[25] Y.H. Tsai, L.T. Cheng, S. Osher and H.K. Zhao, *Fast sweeping algorithms for a class of Hamilton-Jacobi equations*, SIAM J. Numer. Anal., 2003.

[26] J. Urrutia, *Art gallery and illumination problems*, in J.R. Sack and J.Urrutia, editors, Handbook of Computational Geometry, 973-1027, 2000.

[27] F. Wolf, A. Howard and G.S. Sukhatme, *Towards geometric 3D mapping of outdoor environments using mobile robots*, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 1258-1263, 2005.

[28] F. Zhang, E. Justh and P.S. Krishnaprasad, *Steering control, curvature and Lyapunov navigation*, preprint, 2003.

[29] F. Zhang, A. O'Connor, D. Luebke and P.S. Krishnaprasad, *Experimental study of curvature-based control laws for obstacle avoidance*, IEEE International Conference on Robotics and Automation, New Orleans, LA, April, 2004.