

VISIBILITY OPTIMIZATION USING VARIATIONAL APPROACHES

LI-TIEN CHENG AND YEN-HSI TSAI

ABSTRACT. We introduce a framework and construct algorithms based on it to handle optimization problems that deal with the maximization of visibility information for observers when obstacles to vision are present in the environment. This framework uses at its core the approach developed in [17] which adopts the level set framework of [12] to construct a function that encodes visibility information in a continuous way. This continuity allows for powerful techniques to be used in the discrete setting for interpolation, integration, differentiation, and set operations. Thus, through the application of [17], several level set tools, gradient flow, derivative discretizations, and solvers for ordinary differential equations, we produce our visibility framework for optimization and demonstrate its flexibility with algorithms tackling different test problems.

1. INTRODUCTION

The problem of visibility involves the determination of regions in space visible to a given observer when obstacles to 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 paper, we consider a class of optimization problems associated with visibility and solve them under the framework introduced in [17].

Let D be the set of points comprising one or several given solids in a compact subset Ω of \mathbb{R}^d . A solid here refers to an opaque obstruction. We seek solutions to the following three central questions. The first question of our study is:

- *What is the optimal location x_0 for an observer such that the volume of the visible region in Ω is maximized?*

A larger class of problems emerges when variations and extensions involving the observer and the space – multiple observers, moving observers, optimality under different measures – are taken into account. Therefore, our second question extends the first one:

- *What are the optimal locations $\{x_i\}$ for a collection of observers, so that jointly the volume of the visible region in Ω is maximized?*

Lastly, we ask:

- *What is the optimal path $\gamma(t)$ of an observer, traveling from A to B , so that the volume of the visible region in Ω is maximized?*

One can interpret the last question as an extension of the second with uncountable number of observers distributed along the path. Problems related to the three questions above can be found in applications dealing with geometric optics, scattering, path planning, digital surface reconstruction, photolithography, and dynamic games, to name a few (see, e.g., [1, 2, 3, 7, 8]).

In most situations, we find it useful to think of an observers as a light source. Consequently, our attempt in solving the three central questions amounts to maximizing the illuminated regions in Ω , or maximizing the averaged illumination (exposure) in Ω . The main focus is in revealing details of the relationship between optimization and visibility. Thus, it does not detract from the essence of the study for us to disregard global optimality and accept local maxima as suitable solutions, with gradient flow a valid process.

Section 2 briefly describes selected parts of [17], which serves as the underlying framework of our studies. Section 3 introduces an algorithm that answers the central question involving one observer and Euclidean

space. Section 4 includes several subsections devoted to a variety of extensions of the first and second problems. Section 5 presents a few visibility optimization problems that require different techniques than those considered previously. Included in this section will be our attempt at answering the third question. Section 6 summarizes the ideas and techniques developed in the previous sections and ties them together to shed light on the relationship between optimization and visibility. Finally, Section 7 acknowledges the contributions of others in this investigation.

2. VISIBILITY FRAMEWORK

We briefly describe the setup used in the most basic situation considered in [17]. For a more complete description of results, [17] should be consulted. With D representing given obstructions in the space Ω and x_0 the location of the observer, the problem is to identify points of Ω that are visible to the observer. In addition, this should be done in such a way as to facilitate the extraction of visibility-based information.

Let the boundary of D be represented as the zero level set of a function $\psi : \Omega \rightarrow \mathbb{R}$, called a level set function, with $\psi < 0$ on the inside of the solids. When the shape of the objects comprising D excludes small scale features such as infinitely thin projections or point-like masses, this representation is in general stable. In this setting, called the level set framework, we operate on the function ψ instead of the set D . Thus D can be thought of as a binary description of the obstacle – a point is either in D or not in D – while ψ is a continuous one. Numerically, ψ is given as a matrix of values at the lattice locations of a static grid in space, which can be taken to be uniform in each axis direction for ease of use. We also apply the same philosophy and introduce the level set function $\phi : \Omega \rightarrow \mathbb{R}$ to capture the region of invisibility, namely as the set of points where $\phi < 0$. Consequently, the region of visibility is described by $\phi > 0$ and the surface where $\phi = 0$ is the interface, called the shadow boundary, separating the visible from the invisible. Thus, in this setting, the problem becomes that of constructing ϕ from ψ and the observer location x_0 . For more on the level set framework, see, e.g., [11, 12].

The starting point of the algorithm is to first incorporate information that is obviously true, namely marking points inside obstacles as invisible and the observer as visible (or invisible, if located inside an obstacle). Everything else can be temporarily taken to be visible, to be corrected by iterations. In the level set setting, we may thus take $\phi = \psi$ initially, so ϕ has the right sign in D and at x_0 . The type of iteration chosen will involve updating ϕ at points in Ω without changing the sign of ϕ where ϕ is negative and at x_0 . Thus at each step, the negative regions of ϕ are guaranteed to be invisible while the visibility status of the positive regions, excepting x_0 , will be in flux until steady state is reached.

The details of this approach involve working on ϕ along the directional lines-of-sight of the observer, the curves that light emanating from the observer travel along in a space with no visual obstructions. In a space of constant index of refraction, these the directional lines-of-sight are simply rays with endpoint at the observer location, or, mathematically, the integral curves of the vector field $r = x - x_0$ starting at x_0 (more precisely, near x_0). In general, the lines-of-sight can be curves, however, we consider just the case where they are integral curves of a vector field r . One consequence of this is that two lines-of-sight will not intersect unless they are the same one. The main visibility characteristic to notice is that if one were to start out at the observer location and walk along a line-of-sight, the points one travels along will be visible (or invisible, if the observer lies inside an obstacle) until the boundary of an obstacle is reached, after which the points will be invisible. Altogether, this means we may work on each line-of-sight separately to determine the visibility status of its points and the correct order is to work outwards along the line-of-sight from the observer. Furthermore, encountering an obstacle causes invisibility and forces all further points to be invisible.

Thus, a sketch of a simplistic and discrete, bare-boned algorithm works as follows: the points in Ω were previously marked invisible inside D and visible elsewhere as a start. Most notably, x_0 is marked visible if chosen outside of D and invisible if chosen inside. Along each line-of-sight and outwards from x_0 , pass the visibility status of x_0 to its immediate neighbor for comparison. If either the information at x_0 or the

neighbor's current information (alternatively, the neighbor's original information which considers whether it lies in an obstacle) is labeled invisible, then the neighbor relabels itself as invisible. The process is then repeated with the neighbor passing its new visibility information to its immediate neighbor along the line-of-sight for comparison. From this, we see the two main algorithmic aspects for this approach involve a technique for passing values along lines-of-sight and a comparison criterion associated with accepting or rejecting the passed value. Visibility information can be updated in Ω in this manner.

The algorithm ultimately used in [17] follows this basic sketch with choice improvements. Using ϕ to encode visibility information over a grid, as mentioned before, instead of the binary designations of "visible" and "invisible", provides a measure of continuity that realizes subcell resolution for the numerical solution. For example, the shadow boundary can be accurately captured through high order interpolation recovering the zero level set of a locally smooth ϕ . One concern in using a grid is that a given curve in space in general will not pass through any gridpoints. Thus instead of explicitly considering the lines-of-sight, implicitly encoding the lines-of-sight as characteristic directions of a partial differential equation is a better option in the grid-based setting. An example of one such partial differential equation is

$$\partial_t \tilde{\phi} + r \cdot \nabla \tilde{\phi} = 0,$$

with initial condition $\tilde{\phi}(x, 0) = \psi(x)$. We know this as a transport equation, where visibility information given by the value of $\tilde{\phi}$ is transported along the integral curves of r . Note, in practice, applying a finite differencing PDE solver (e.g., [10, 13]) is the most direct way to solve this equation over the grid but not necessarily the best. Instead, efficient grid-based methods such as fast marching, fast sweeping, and other directional sweeping techniques are preferable, especially since this step of the algorithm forms the bulk of the computational workload. For more, see [9, 15, 19] for fast marching and [18] for fast sweeping. Finally the update comparison criterion is added to the passing process. When a gridpoint receives a passed value of ϕ for update, it accepts the minimum between this value and its current value. The effect of this is if either value is negative, then the updated value will be negative and, furthermore, the final ϕ will be a continuous function.

An example of the entire procedure is to start with $\tilde{\phi}(x, 0) = \psi(x)$ and a chosen partition of time steps $0 = t_0 < t_1 < \dots < t_n = T$ to a final time T . At each time step $t_k, k = 0, 1, \dots, n-1$, solve

$$\partial_t \tilde{\phi} + r \cdot \nabla \tilde{\phi} = 0,$$

in Ω , perhaps using standard finite differencing techniques, from time t_k to t_{k+1} to pass and then reset

$$\tilde{\phi}(x, t_{k+1}) = \min\{\tilde{\phi}(x, t_k), \tilde{\phi}(x, t_{k+1})\}$$

for the comparison update. If T is large enough, $\tilde{\phi}$ will reach steady state in Ω and give the correct visibility information for the region. Thus $\phi(x) = \tilde{\phi}(x, T)$ is what we want to construct.

A variation that possibly facilitates analysis is the self-contained partial differential equation description,

$$\partial_t \tilde{\phi} + \max\{r \cdot \nabla \tilde{\phi}, 0\} = 0,$$

with initial condition $\tilde{\phi}(x, 0) = \psi(x)$, to be solved to steady state in Ω . Here, the passing of values is enacted through transport with respect to r and the maximum taken in the equation serves as the update comparison criterion, ensuring each point will only change its value if passed one that is smaller. However, the fast marching, fast sweeping, and other techniques we use in practice lead to more efficient algorithms. Furthermore, we note that the visibility level set function takes the analytic form of

$$\phi(x) = \min_{z \in \mathcal{L}(x, x_0)} \psi(z),$$

where $\mathcal{L}(x, x_0)$ is the integral curve of the vector field r , connecting x and x_0 . This form, however, in many circumstances, is not easy to work with.

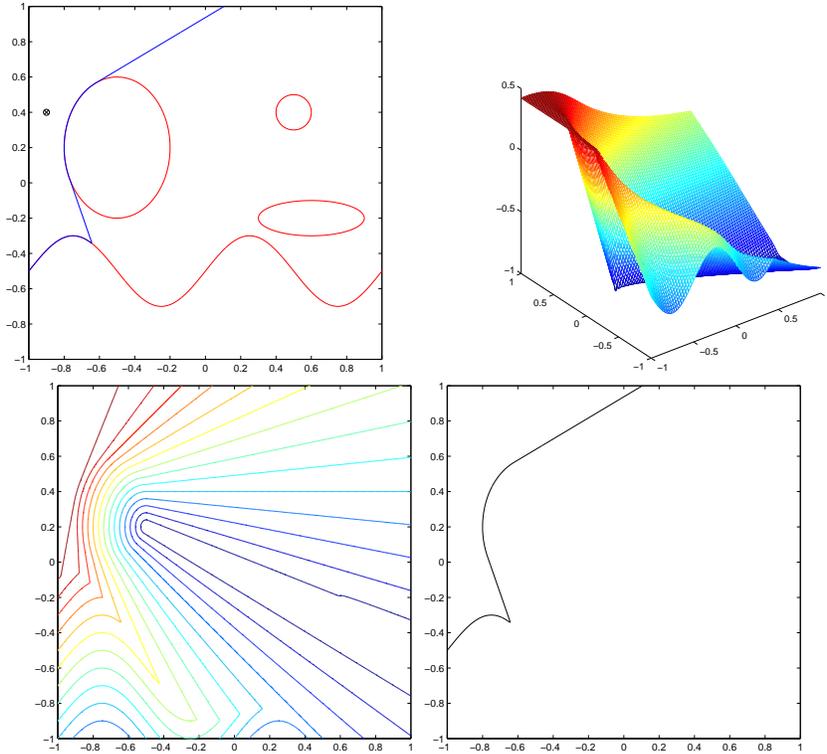


FIGURE 2.1. For the obstacle environment and given observer location of the upper left plot, the visibility level set function is shown along with its contours and, in the right plot, its zero level set, the shadow boundary.

Altogether, we have described a set of algorithms that can find the visible and invisible regions, when iterated to steady state, for given obstructions and an observer location. Figure 2.1 shows a typical visibility level set function generated from the approach, along with its contours and the shadow boundary, for a chosen observer location. We note that in the constant index of refraction case with $r = x - x_0$, the computational complexity is the optimal $O(N^d)$ over a N^d point grid in d dimensions if fast sweeping is used for passing. Moreover, multiresolution techniques can be incorporated to further improve speed and lessen memory requirements. However, the main advantage of the level set representation and PDE description is a framework that emphasizes continuity over the discrete. The information gathered from the solution goes beyond the black and white designations of “visible” and “invisible”, but provides gray levels as well. For example, encoded in the solution is an accurate representation of shadow boundaries that can help measure how visible or invisible a point is. See, e.g., [4] for a review of other visibility algorithms and [1] for another level set-based solution. Thus the approach becomes especially advantageous when applied to problems that need the additional provided information. The optimization of visibility is one such problem.

3. SINGLE POINT VISIBILITY OPTIMIZATION

To facilitate our discussion in adaptivity in the vantage position, we augment the visibility function ϕ described above so that

$$\phi : \Omega^2 \subset \mathbb{R}^d \times \mathbb{R}^d \mapsto \phi(y; x_0) \in \mathbb{R}$$

denotes the visibility function created from a vantage point located at x_0 ; i.e. $\phi(\cdot; x_0)$ the level set function representing visibility in a bounded Ω for an observer at x_0 , with $\phi(\cdot; x_0) > 0$ in the visible regions and

$\phi(\cdot; x_0) < 0$ in the invisible regions. Such a function can be calculated using the visibility solver of [17] described in the previous section. Consider the visible volume function $V(x_0)$ defined as the volume of the visible region for the observer at x_0 . Thus we have the expression

$$V(x_0) = \int_{\Omega} H(\phi(y; x_0)) dy,$$

where H denotes the one-dimensional heaviside function. With this, the problem of interest becomes that of finding the position of the observer x_0 that maximizes this function, thus maximizing the size of the visible region.

If gradient ascent is used, we obtain a flow, potentially of interest itself, of the position of the observer from an initial guess to a local maximum. This can be thought of as a greedy algorithm for a moving observer to maximize its visibility when it is initially located at a non-optimal position. The gradient direction to consider, as easily derived from variational calculus, is $\nabla_{x_0} V(x_0)$. Thus the gradient ascent flow of the observer location is described by

$$\partial_t x_0 = \nabla_{x_0} V(x_0).$$

The ∇_{x_0} used here and later on in the paper is the gradient operator with respect to the observer position.

Analysis of this differential equation is hindered by the lack of convenient analytical forms for $\nabla_{x_0} \phi(\cdot; x_0)$. Thus we take a numerical approach to its solution. The right hand side derivatives can be approximated by central differencing of the values of V at neighboring points, $x_0 \pm h e_i$, $i = 1, \dots, d$, where $\{e_i\}_{i=1}^d$ denotes the standard orthonormal basis of \mathbb{R}^d and h is a chosen stepsize. We denote the resulting approximation of $\nabla_{x_0} V(x_0)$ by $D_0^h V(x_0)$, using standard differencing notation. Note multiple applications of the algorithm of [17] are needed to obtain $\phi(\cdot; x_0 \pm h e_i)$, $i = 1, \dots, d$. In total, this approach leads to the system of ordinary differential equations,

$$\partial_t x_0 = D_0^h V(x_0),$$

approximating gradient ascent flow. A choice of solver for this system completes the algorithm for single point visibility optimization.

We formulate in detail the steps of this algorithm when Euler's method is used as the ordinary differential equation solver:

- (1) Start with a given position x_0 . This can be thought of as an approximation of the optimal location.
- (2) For a chosen stepsize h , use the algorithm of [17] to obtain the $2n$ visibility level set functions $\phi_{x_0 \pm h e_i}$ over a grid in Ω for each $i = 1, \dots, n$.
- (3) Evaluate $V(x_0 \pm h e_i)$ for each $i = 1, \dots, n$ using a smoothed-out approximate heaviside function and numerical integration techniques such as the trapezoidal rule over the grid. We note the ideas of [6] should be used to create the heaviside function for accuracy.
- (4) Form $D_0^h V(x_0)$ through central differencing on $\nabla_{x_0} V(x_0)$. This uses the values calculated in the previous step.
- (5) Using Euler's method, update x_0 by $x_0 + k D_0^h V(x_0)$, where k is a chosen time marching stepsize.
- (6) Repeat from the first step with this new value of x_0 until convergence. Convergence implies a local maximum of the visible volume is reached.

Note the computational workload in each iteration is clearly dominated by the second and third steps. However, the workload of the third step can be reduced by noticing that step is only needed for computing $D_0^h V(x_0)$, in the fourth step. Instead, if the identity

$$D_0^h V(x_0) = \int_{\Omega} D_0^h H(\phi(y; x_0)) dy$$

is used, we notice that $D_0^h H(\phi(y; x_0))$ happens to be zero in a large portion of Ω . In other words, under small perturbations of the position of the observer, the visible and invisible regions will undergo similarly small

changes. This can be seen mathematically in the expansion

$$(3.1) \quad \nabla_{x_0} V(x_0) = \int_{\Omega} \nabla_{x_0} H(\phi(y; x_0)) dy = \int_{\Omega} \delta(\phi(y; x_0)) \nabla_{x_0} \phi(y; x_0) dy,$$

where δ refers to the one-dimensional delta function. Thus, only points with zero or near-zero $\phi(\cdot; x_0)$, need to be considered in the integral. The visibility function $\phi(y; x_0)$ constructed by the algorithm in [17] takes the form

$$(3.2) \quad \phi(y; x_0) = \min_{z \in \mathcal{L}(y, x_0)} \psi(z),$$

as stated previously. In the cases considered in this paper, the vector field $r(x)$ is simply $(x - x_0)/|x - x_0|$. The following Lemma, with this vector field, shows that $\phi(y; x_0)$ thus constructed is Lipschitz continuous and so the central differencing approximation for $\nabla_{x_0} \phi(y; x_0)$ in (3.1) will remain a bounded quantity.

Lemma 3.1. *Let K be a Lipschitz constant for ψ in Ω , and ϕ be defined as in (3.2). Then $|\phi(y; x_0) - \phi(y; \tilde{x}_0)| \leq K|x_0 - \tilde{x}_0|$ for $x_0, \tilde{x}_0, y \in \Omega$.*

Proof. We can rewrite the expression for ϕ as follows:

$$\phi(y; x_0) = \min_{t \in [0, 1]} \psi(tx_0 + (1-t)y).$$

Thus,

$$\begin{aligned} \phi(y; x_0 + \delta x) &= \min_{t \in [0, 1]} \psi(tx_0 + (1-t)y + t\delta x) \\ &\leq \min_{t \in [0, 1]} (\psi(tx_0 + (1-t)y) + tK|\delta x|) \\ &\leq \phi(y; x_0) + K|\delta x|. \end{aligned}$$

Similarly, $\phi(y; x_0 + \delta x) \geq \phi(y; x_0) - K|\delta x|$. So $|\phi(y; x_0) - \phi(y; \tilde{x}_0)| \leq K|x_0 - \tilde{x}_0|$. \square

We also note the continuous representation of visibility information allows for accurate computations using approximate heaviside and delta functions since shadow boundaries can be accurately located. We can in fact use the form in 3.2, with standard numerical integration and differentiation techniques and the approximate delta function or heaviside function of [6], to efficiently compute the value of $\nabla_{x_0} V(x_0)$ in place of the fourth step.

We present two examples of single point visibility optimization using different starting locations for the observer. Figure 3.1 shows the path of an observer originally at (0.4, 0.4) and the area of the region of visibility during the flow in $\Omega = [-1, 1] \times [-1, 1]$. Figure 3.2 shows the path of an observer originally at (0.6, 0.6) and the corresponding visible region areas in the same setting. In this case, the observer prefers to run away towards infinity. Thus, the computation was stopped when it hit the boundary of Ω . Finally, Figure 3.3 shows a graph of the area of visible regions plotted with respect to different observer locations in the same setting. The obstacles can be identified in the plot by the regions where the area is zero. The algorithm essentially employs gradient ascent along this landscape. Thus, the observer of Figure 3.1 converges to the local maximum at the origin and the observer of Figure 3.2 goes away toward the boundary.

4. EXTENSIONS FOR VISIBILITY OPTIMIZATION PROBLEMS

The previous section provided an answer for the central and most basic question concerning optimization and visibility. The reason this particular question is chosen as the central one is due to the multitude of extensions that immediately arise from its answer. We consider in this section visibility problems involving multiple observers, weighted regions of importance in space, accumulating visibility information, and weighted observer distances.

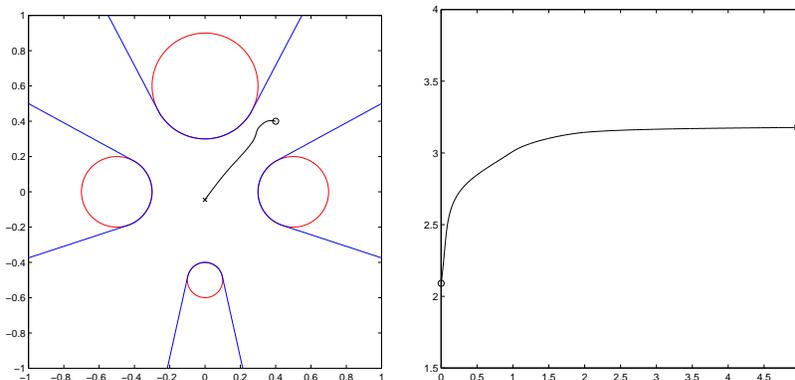


FIGURE 3.1. The figure on the left shows the path the observer travels to optimize visibility. This path originates at ‘o’ and ends at ‘x’. The obstacles to the vision of the observer are four disks of various radii. Also drawn are the shadow boundaries separating the regions of visibility and invisibility. The figure on the right shows the area of the region of visibility plotted with respect to time in the gradient flow of the observer.

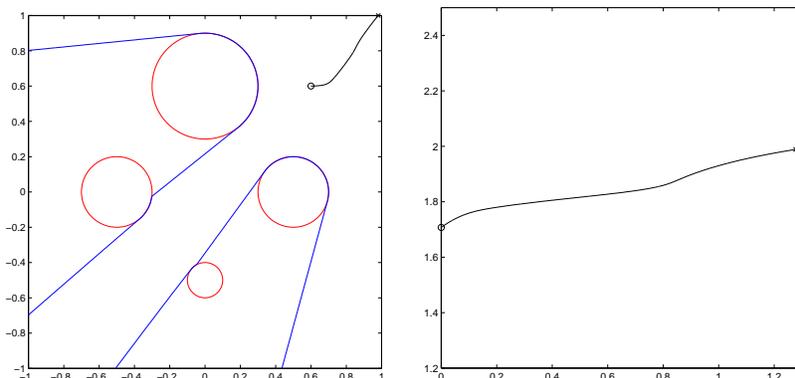


FIGURE 3.2. This figure shows has the same setting as in Figure 3.1 with a different initial location for the observer.

The construction of solutions for these various problems will in general follow the same procedure as listed in the previous section. This implies the construction of a function that is usually associated to volumes of visible regions, different for each problem, and a gradient ascent flow maximizing its value. Numerically, this visible volume function can be accurately evaluated due to continuity in the visibility information provided by the level set framework of [17]. These evaluations are used in a chosen finite differencing approximation of derivatives to approximate the gradient ascent direction, which is then combined with an ODE solver in time for a method-of-lines solution to the flow.

4.1. Multiple Observers. Instead of one observer, we may consider several observers and ask where they should be placed for maximal visibility. Let x_0, x_1, \dots, x_m denote the location of $m + 1$ separate observers. For each $i = 0, 1, \dots, m$, we can construct the visibility level set function $\phi(\cdot; x_i)$ associated to x_i . Visibility information of all the observers can be determined from the visibility information of individual ones due to our definition that a point is visible with respect to multiple observers if it is visible to one of them. Thus the region of visibility for multiple observers is the union of the regions for each individual observer.

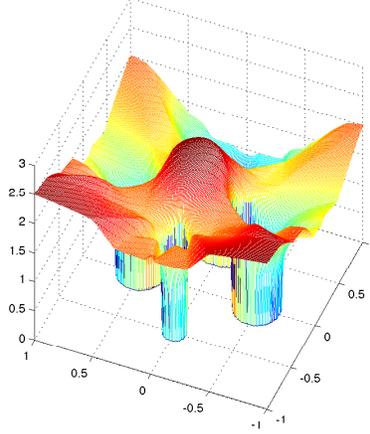


FIGURE 3.3. The area of visible regions plotted with respect to different observer locations in the settings of Figures 3.1 and 3.2. Note when the observer is located inside a solid obstacle, the area is zero.

In the level set framework, there is an analogy to unions and intersections. For two level set functions ϕ_1 and ϕ_2 , the union of their negative regions, $\{\phi_1 < 0\}$ and $\{\phi_2 < 0\}$, is implicitly captured as the negative region of the level set function $\min\{\phi_1, \phi_2\}$. Note, the positive region of this function is thus the intersection of the positive regions of ϕ_1 and ϕ_2 . On the other hand, the intersection of the negative regions of ϕ_1 and ϕ_2 is implicitly captured as the negative region of $\max\{\phi_1, \phi_2\}$. Similarly, the positive region of this function is thus the union of the positive regions of ϕ_1 and ϕ_2 .

From this, we can construct a visibility level set function for multiple observers, which we denote by $\phi(\cdot; x_0, x_1, \dots, x_m)$, by taking the minimum value of the visibility level set functions for individual observers,

$$\phi(y; x_0, x_1, \dots, x_m) = \min_{i=0,1,\dots,m} \phi(y; x_i).$$

We then define a new function corresponding to the volume of the visible part of Ω with respect to the multiple observers,

$$V(x_0, x_1, \dots, x_m) = \int_{\Omega} H(\phi(y; x_0, x_1, \dots, x_m)) dy.$$

The positions x_0, x_1, \dots, x_m maximizing this function will be the desired optimal visibility locations for the multiple observers.

We perform the maximization through gradient ascent on V . This translates to, through calculus of variations, motion of x_i , for each $i = 0, 1, \dots, m$, with direction and speed given by $\nabla_{x_i} V(x_0, x_1, \dots, x_m)$, respectively:

$$\partial_t x_i = \nabla_{x_i} V(x_0, x_1, \dots, x_m),$$

where ∇_{x_i} denotes the gradient in the argument x_i . Using once again the method-of-lines approach, we simulate this motion by approximating the derivatives of the spatial gradients using finite differencing to obtain a system of ordinary differential equations. This system can then be solved with an appropriate solver when given initial observer locations for x_0, x_1, \dots, x_m .

In Figure 4.1, two observers and the path they take to maximize visibility are shown. The observer locations converge to final positions that form a local maximum of the visible volume function. There is a slight, almost invisible, decrease in the area of the region visibility along the gradient ascent path. This, we believe, is simply due to small numerical errors. Also, there is a slight deformation in the shadow boundary at the lower middle of the graph resulting from the plotter's attempt to resolve the kink at that location in the curve. Figure 4.2 shows the result of different initial observer placement. The final positions are

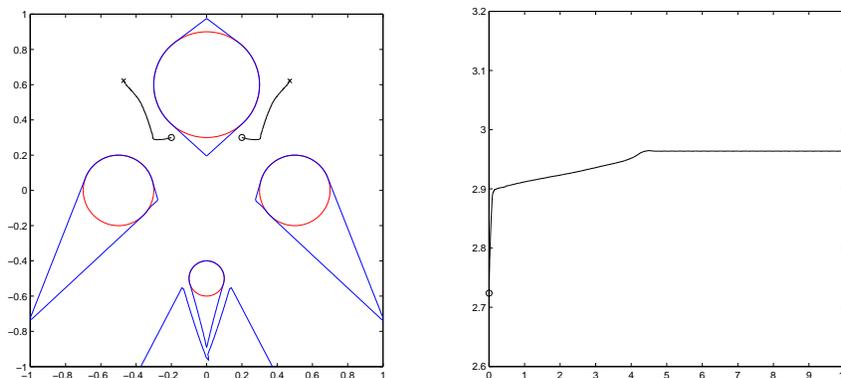


FIGURE 4.1. Two observers following gradient flow converge to positions for optimal viewing of the space. Note the figure on the right shows a slight decrease in the area of the visible region calculated along the path of the observer.

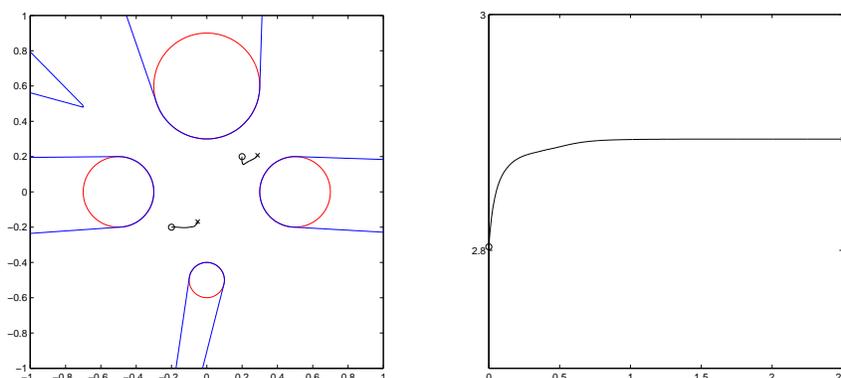


FIGURE 4.2. Different starting locations for the observers lead to different solutions. In this case, less of the space is visible to the observers than for the final positions found in Figure 4.1.

different and the area of the region of visibility is smaller. Finally, Figure 4.3 shows results with four and five differently situated observers in the obstacle environment.

In Figure 4.4, three robots are placed randomly on a circular orbit at initial time. Through optimization on the collective visibility of the robots, we are able to determine the locally optimal search direction of each robot. This means that each search direction is computed to maximize not how much more visibility information a specific robot can obtain, but how much the ensemble of the three robots can. In the simple case of one obstacle centered at the origin, we see that the robots are able to find a globally optimal solution (any right triangle on the circular orbit) based on their initial locations.

4.2. Weights in Space. In certain applications, a higher priority may be placed on viewing a specific region in space, while a lower priority is placed on other regions. The effects of this on the optimal positions of observers, as well as on the motions associated to them, can be simulated through the use of weights.

Let $w : \Omega \rightarrow \mathbb{R}^+$ be a positive real-valued function defined over Ω . Let the magnitude of the value of w at a point relate to how important it is for that point to be visible, with larger magnitude associated with greater importance. By including w in the measure used in spatial integration, we in effect attach importance weights to the visibility of space.

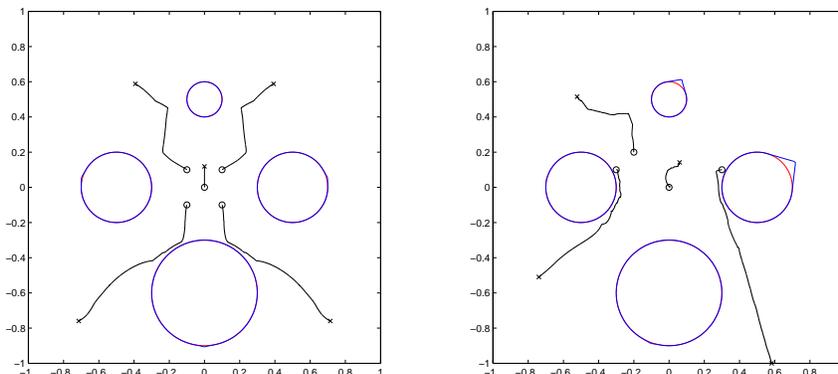


FIGURE 4.3. Two more results of multiple observer visibility are shown. In the left plot, five observers move to a position where almost everything can be seen. In the right plot, four observers maximize their visibility until one tries to escape Ω .

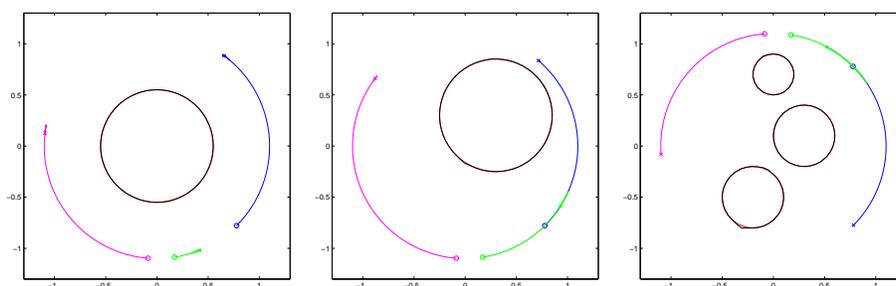


FIGURE 4.4. The three observers are confined on a circular orbit surrounding the obstacles.

For single point visibility optimization, the volume function is modified to

$$V_w(x_0) = \int_{\Omega} H(\phi(y; x_0)) w(y) dy.$$

Thus, having w large in the region of visibility helps increase this function. The gradient flow process then becomes

$$\partial_t x_0 = \nabla_{x_0} V_w(x_0),$$

and we can solve it following our usual steps involving visibility algorithm, finite differencing, and the method-of-lines.

Figure 4.5 shows the motion of an observer initially placed at $(0.2, 0.2)$ for a Gaussian importance weight centered at $(1, 0.2)$. When equally weighted, the observer would instead move toward the origin. Figure 4.6 shows a non-standard obstacle arising from the boundary of an image. A Gaussian importance weight is centered at the left wall of the square Ω . Initially, the observer is situated such that most of the wall is not visible. However, by moving along the plotted path, the observer maximizes its visibility of the wall and, in the end, can view it completely.

In a more extreme case of the use of weights, certain computer graphics applications are solely interested in visibility of the obstacle surfaces. Consider the surface area function

$$V_S(x_0) = \int_S H(\phi(y; x_0)) dA = \int_{\Omega} H(\phi(y; x_0)) \delta(\psi(y)) |\nabla \psi(y)| dy,$$

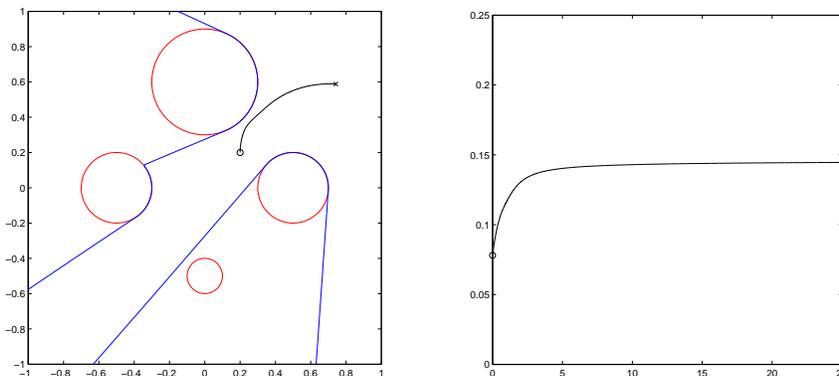


FIGURE 4.5. The figure on the left shows the path of an observer that places particular importance in viewing the area surrounding the point $(1,0.2)$. The values calculated in the figure on the right represent values of the weighted visible volume function and hence are only related to area.

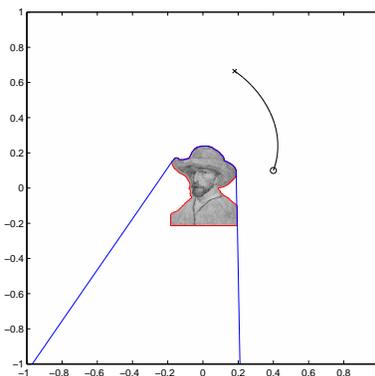


FIGURE 4.6. In this figure, an image serves as the obstacle and visibility of the left wall of the square Ω is of importance. The observer moves from behind the obstacle to maximize its visibility of the wall, coming to a halt when the wall is completely visible.

where S denotes the obstacle surfaces. Maximization of this function maximizes visibility of S , the obstacle surfaces. In practice, we replace the delta function by a smoothed-out approximation that can be considered a weight with values varying from near zero to near infinity. Choice of this approximate delta function and heaviside function should follow the work of [6] to satisfy accuracy requirements.

4.3. Effect of Memory. The algorithm for single point visibility optimization, through the use of gradient flow, leads to a greedy motion for maximizing the visibility of an initially situated observer. However, we may not be exclusively interested in maximizing the visibility of the observer's final position. In the case of an unchanging landscape, the observer may be able to remember what it sees during its motion. With the introduction of memory, the more interesting problem becomes that of finding a motion that, at a given time, attempts to instantaneously maximize visibility in the region of points that, up to that time, have remained invisible. The final path, in general, will not be the same as that constructed by the single point visibility optimization algorithm since information accumulated during motion has a very real effect.

To create the path and motion of interest, we record the accumulated visibility information at each time and use it to determine the maximizing direction and speed for the observer to travel at that time. Then, as the observer travels along this direction, visibility information is updated accordingly.

Let γ be a piece of a curve representing the path traced by the observer up to a certain time. The accumulated visibility information of the observer up to that time can be encoded in the level set function

$$\max\{\phi(y;x)|x \in \gamma\},$$

call it ϕ_γ , whose visible regions are the union of visible regions for individual observers located along γ . Note, thus the accumulated visibility information comes from visibility information gathered by multiple, in this case infinite, observers along γ . In the following, we will often use this multiple observer point of view.

To determine the maximizing direction and speed for visibility at this time, we consider the location of an observer x_0 and the instantaneous visible volume function

$$V_\gamma(x_0) = \int_{\Omega} H(\max\{\phi(y;\gamma), \phi(y,x_0)\})dy.$$

This energy measures the volume of the region visible to either an observer at γ or x_0 . If we consider $\nabla_{x_0} V_\gamma(x_0)$, it tells us the direction and speed for an observer located at x_0 to travel to maximize visibility when observers located along γ are present. If x_0 is specifically chosen to be the later endpoint of γ , $\nabla_{x_0} V_\gamma(x_0)$ then gives the direction and speed for an observer continuing along the path γ to maximize its accumulated visibility.

Note, exact evaluation of $\phi(\cdot;\gamma)$ is difficult due to the fact that γ is composed of an infinite number of points. We handle this by taking instead a discrete sampling of points of γ , $\{z_0, z_1, \dots, z_N\}$. The visibility level set function of interest can then be approximated by a finite multiple observer version,

$$\phi(y; z_0, z_1, \dots, z_N) = \max\{\phi(y;x)|x \in \{z_0, z_1, \dots, z_N\}\}.$$

We will justify the validity of this approximation in Section 5.1.

The main steps of the numerical algorithm are thus, in a condensed format:

- (1) Start with a partition $0 = t_0 < t_1 < \dots < t_n = T$, for a chosen final time T , and an initial location $x_0^{(0)}$ for the observer at t_0 .
- (2) For $k = 0, 1, \dots, n-1$, flow the observer at time step t_k by numerically solving

$$\partial_t x_0 = \nabla_{x_0} V(x_0^{(0)}, x_0^{(1)}, \dots, x_0^{(k)}, x_0),$$

where $x_0(t = t_k) = x_0^{(k)}$, up to the next time step t_{k+1} , and call the result $x_0^{(k+1)}$. This involves computation of the multiple observer visibility level set function $\phi(\cdot; x_0^{(0)}, x_0^{(1)}, \dots, x_0^{(k)})$, finite differencing on the gradient, and a chosen ODE solver such as Euler's method.

Figure 4.7 shows the effects of memory on an observer trying to maximize what it can see. Note most of space has been visible to the observer at one time or another during the course of this flow. Figure 4.8 shows a different initial observer location which causes the observer to run towards infinity. In this case, the computation was halted when the observer touched the boundary of Ω . Figure 4.9 shows different views of a computation involving an observer among obstacles in three dimensional space.

4.4. Note on Weights and the Observer. One perhaps undesirable phenomenon we observe in our examples so far is that in general, the farther away the observer is, the more it can see (see, e.g., Figures 3.2 and 4.8). Thus, in many situations, the position of the observer giving a local maximum for visibility will lie on the boundary of Ω . However, Ω may not be physically relevant, just serving as a device that enforces finite volume regions so that maximizing visibility makes sense. In fact, we have arbitrarily taken Ω to be the square computational domain $[-1, 1] \times [-1, 1]$ in the examples we have considered. Thus, in many situations, if the computational domain is expanded, the optimal location of the observer changes with it,

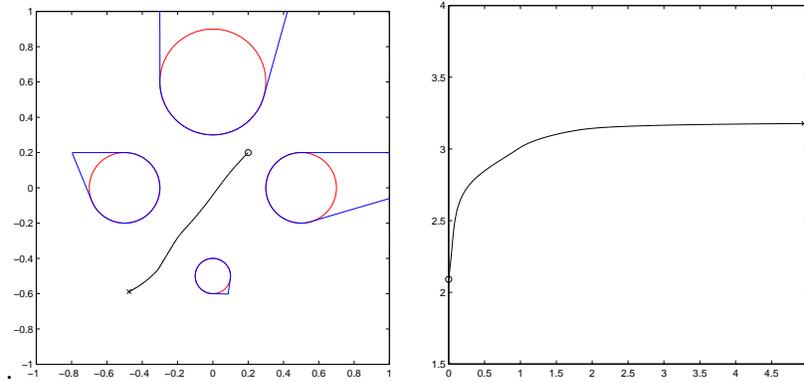


FIGURE 4.7. This figure shows the optimizing path of an observer that records visibility information as it moves. The shadow boundaries now those from visibility information culled from the memory of the observer.

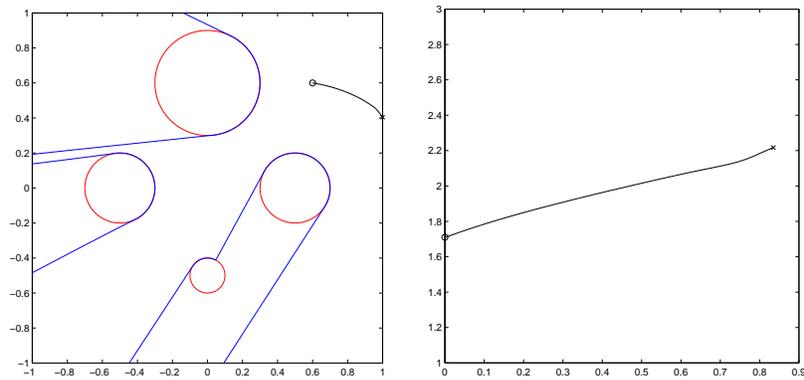


FIGURE 4.8. This figure shows the a different initial placement of the observer. In order to optimize visibility, in this case it chooses to run towards infinity, even with memory effects present.

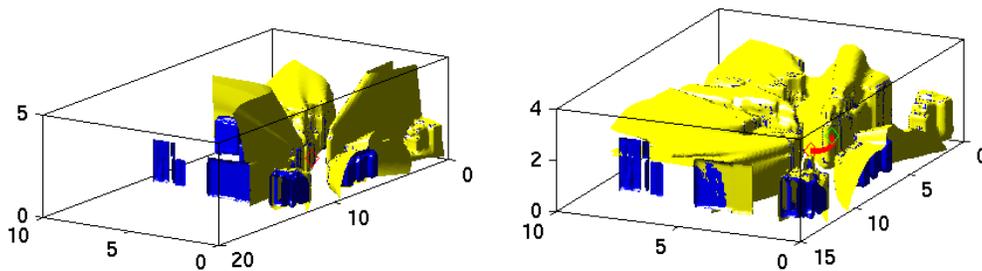


FIGURE 4.9. This figure shows different views of the observer path for maximizing accumulated visibility among obstacles in three dimensional space.

preferring to head off to infinity for a local maximum. In terms of human visibility, such a concept is not natural because distance obscures visual detail. To model this, we introduce weights in space that depend on distance from the observer.

Let $w_{x_0} : \mathbb{R}^+ \cup \{0\} \rightarrow \mathbb{R}^+$ be a decreasing positive function. We will use it solely in conjunction with the distance away from the observer x_0 , in the form $w_{x_0}(|x - x_0|)$, where it prescribes weights on the visibility of points in relation to their distance away from the observer x_0 . If human vision is of interest, the specific form of w_{x_0} can be chosen according to human visual experiments. By incorporating such a weight into the visible volume function, we penalize the observer for being too far from what it wishes to see, represented by another weight function w , rendering distant regions virtually invisible. For the weighted single point visibility optimization problem, the visible volume function changes to

$$V_{w,w_{x_0}}(x_0) = \int_{\Omega} H(\phi(y;x_0))w_{x_0}(|y - x_0|)w(y)dy,$$

with the gradient ascent flow

$$\partial_t x_0 = \nabla_{x_0} V_{w,w_{x_0}}(x_0),$$

maximizing its value.

We restrict our formulation here to the single observer case. For multiple observers, a point may be visible to more than one observer, leading to two different weights placed on it related to its distance from each of those observers. The correct weight to choose becomes an issue which, though not difficult, we will not consider at the present. In fact, we defer studies and results in the presence of visual resolution to a future paper, along with the closely related topic of partial visibility, where an observer cannot perfectly make out everything in its visual field.

5. OTHER TYPES OF VISIBILITY OPTIMIZATION PROBLEMS

In the previous section, we considered extensions of the approach for single point observer visibility optimization. Thus all the problems were solved using appropriately chosen visible volume functions and spatial and temporal discretizations. However, not all optimization problems dealing with visibility can or should be solved in this manner. In this section, we consider different approaches for finding paths that allow a more uniform viewing of the space and shortest paths to visibility as well.

5.1. Exposure. In the accumulated visibility problem, a point in space may be seen, on one hand, for the duration of an observer's path or, on the other hand, for just a split second. The former case represents too much attention perhaps needlessly paid to that point and the latter represents not enough attention. We consider here the construction of a path where the observer has a more uniform viewing habit.

Consider the amount of time a point x is exposed to an observer travelling at unit speed along a path $\gamma : [0, 1] \rightarrow \mathbb{R}^d$, parametrized by τ ,

$$\mathcal{X}(x;\gamma) = \int_0^1 H \circ \phi(x;\gamma(\tau))|\gamma'(\tau)|d\tau,$$

which we will refer to as the exposure due to γ on x . Here, unit speed for the observer is considered for a more geometric and parametrization-independent solution. Thus, we can alternatively think of γ as an infinite set of observers instead, and \mathcal{X} simply counts how many times observers on γ can see x . It is worth noting that $\mathcal{X}(x;\gamma)$ is bounded above by the length of γ ; i.e. $\mathcal{X}(x;\gamma) \leq L(\gamma)$.

5.1.1. Uniform Exposure. Points outside of obstacles can be said to be viewed in a more uniform manner by an observer moving along γ if the deviation of the exposure from being constant,

$$\int_{\Omega \setminus D} (\mathcal{X}(x;\gamma) - C)^2 dx,$$

is small for some constant C . Thus, we formulate a boundary value problem as follows:

Problem 5.1. Given $p_0, p_1 \in \mathbb{R}^d$, and a constant C , find $\gamma : [0, 1] \mapsto \mathbb{R}^d$ with $\gamma(0) = p_0$ and $\gamma(1) = p_1$ such that the energy

$$(5.1) \quad E(\gamma, C) = \frac{1}{2} \int_{\Omega \setminus D} (X(x; \gamma) - C)^2 dx + \lambda \int_0^1 |\gamma'(\tau)| d\tau$$

is minimized.

Notice that the last term in (5.1) is λ times the length of the curve, $L(\gamma)$, and seeks to stabilize the problem, when λ is chosen large enough, by penalizing against fractal or space-filling paths.

Proposition 5.2. *The Euler-Lagrange Equation for Problem 5.1 is*

$$\int_{\Omega \setminus D} (X(x; \gamma(\tau)) - C) [\delta \circ \phi(x; \gamma(\tau)) \mathcal{P}_{\gamma'(\tau)} \nabla_{\gamma(\tau)} \phi_{\gamma(\tau)} - H \circ \phi(x; \gamma(\tau)) \kappa(\tau) n(\tau)] dx - \lambda \kappa(\tau) n(\tau) = 0,$$

for every $\tau \in (0, 1)$, and any given C . Here, $\mathcal{P}_w v$ is the normal projection of v from vector w .

Proof. Performing variational calculus on this, we can arrive at the Euler-Lagrange equation and a gradient descent flow for minimization. For the path γ , we consider the formal derivation in parts. First,

$$\begin{aligned} \frac{d}{ds} \Big|_{s=0} X(x; \gamma + s\eta) &= \frac{d}{ds} \Big|_{s=0} \int_0^1 H \circ \phi(x; \gamma(\tau) + s\eta(\tau)) |\gamma'(\tau) + s\eta'(\tau)| d\tau \\ &= \int_0^1 \delta \circ \phi(x; \gamma(\tau)) \nabla_{\gamma(\tau)} \phi(x; \gamma(\tau)) \cdot \eta(\tau) |\gamma'(\tau)| d\tau + \int_0^1 H \circ \phi(x; \gamma(\tau)) \frac{\gamma'(\tau)}{|\gamma'(\tau)|} \cdot \eta'(\tau) d\tau. \end{aligned}$$

Integration by parts, along with the fact that $\eta(0) = \eta(1) = 0$ due to the Dirichlet boundary conditions, transforms the second integral to

$$- \int_0^1 \left[\delta \circ \phi(x; \gamma(\tau)) \nabla_{\gamma(\tau)} \phi(x; \gamma(\tau)) \gamma'(\tau) + H \circ \phi(x; \gamma(\tau)) \left(\frac{\gamma'(\tau)}{|\gamma'(\tau)|} \right)' \right] \cdot \eta(\tau) d\tau.$$

Inputting this result back into the equation gives the variational derivative as

$$\int_0^1 \left[\delta \circ \phi(x; \gamma(\tau)) \left(\nabla_{\gamma(\tau)} \phi(x; \gamma(\tau)) - \nabla_{\gamma(\tau)} \phi(x; \gamma(\tau)) \cdot \frac{\gamma'(\tau)}{|\gamma'(\tau)|} \frac{\gamma'(\tau)}{|\gamma'(\tau)|} \right) - H \circ \phi(x; \gamma(\tau)) \frac{1}{|\gamma'(\tau)|} \left(\frac{\gamma'(\tau)}{|\gamma'(\tau)|} \right)' \right] \cdot \eta(\tau) |\gamma'(\tau)| d\tau,$$

which can be simplified to the final result in 2 dimensions,

$$\frac{d}{ds} \Big|_{s=0} X(x; \gamma + s\eta) = \int_0^1 [\delta \circ \phi(x; \gamma(\tau)) (\nabla_{\gamma(\tau)} \phi(x; \gamma(\tau)) \cdot n(\tau)) n(\tau) + H \circ \phi(x; \gamma(\tau)) \kappa(\tau) n(\tau)] \cdot \eta(\tau) |\gamma'(\tau)| d\tau,$$

where $\kappa(\tau) n(\tau)$ refers to the curvature vector that points in the normal direction $n(\tau)$. Also,

$$\begin{aligned} \frac{d}{ds} \Big|_{s=0} L(\gamma + s\eta) &= \frac{d}{ds} \Big|_{s=0} \int_0^1 |\gamma'(\tau) + s\eta'(\tau)| d\tau \\ &= \int_0^1 \frac{\gamma'(\tau)}{|\gamma'(\tau)|} \cdot \eta'(\tau) d\tau \\ &= - \int_0^1 \frac{1}{|\gamma'(\tau)|} \left(\frac{\gamma'(\tau)}{|\gamma'(\tau)|} \right)' \cdot \eta(\tau) |\gamma'(\tau)| d\tau \\ &= - \int_0^1 \kappa(\tau) n(\tau) \cdot \eta(\tau) |\gamma'(\tau)| d\tau, \end{aligned}$$

once again using integration by parts. With these identities, we can conclude that the Euler-Lagrange equation takes the form

$$\int_{\Omega \setminus D} (X(x; \gamma(\tau)) - C) [\delta \circ \phi(x; \gamma(\tau)) \mathcal{P}_{\gamma'(\tau)} \nabla_{\gamma(\tau)} \phi(x; \gamma(\tau)) - H \circ \phi(x; \gamma(\tau)) \kappa(\tau) n(\tau)] dx - \lambda \kappa(\tau) n(\tau) = 0,$$

for all τ . □

Thus, the gradient descent procedure introduces the time variable t into γ to form $\gamma(\tau, t)$, and flows it by

$$\begin{aligned} \partial_t \gamma(\tau, t) = & \left(- \int_{\Omega \setminus D} (X(x; \gamma(\tau)) - C) \delta \circ \phi(x; \gamma(\tau)) \mathcal{P}_{\gamma(\tau)} \nabla_{\gamma(\tau)} \phi(x; \gamma(\tau)) dx \right) + \\ & \left(\int_{\Omega \setminus D} (X(x; \gamma(\tau)) - C) H \circ \phi(x; \gamma(\tau)) dx + \lambda \right) \kappa(\tau) n(\tau) \end{aligned}$$

from an initial guess of the path outside obstacles and through the given endpoints. In particular, in two dimensions,

$$(5.2) \quad \begin{aligned} \partial_t \gamma(\tau, t) = & \left(- \int_{\Omega \setminus D} (X(x; \gamma(\tau)) - C) \delta \circ \phi(x; \gamma(\tau)) (\nabla_{\gamma(\tau)} \phi(x; \gamma(\tau)) \cdot n(\tau)) dx \right) n(\tau) + \\ & \left(\int_{\Omega \setminus D} (X(x; \gamma(\tau)) - C) H \circ \phi(x; \gamma(\tau)) dx + \lambda \right) \kappa(\tau) n(\tau). \end{aligned}$$

We may also multiply the right hand side of the flow by $H(\psi)$ to ensure that the path respects the fact that obstacles are impenetrable. If in addition, we optimize also for C , it is easy to see that the optimal C should be chosen to be the average exposure,

$$C = \frac{\int_{\Omega \setminus D} X(x; \gamma) dx}{\int_{\Omega \setminus D} dx}.$$

5.1.2. Numerical Considerations. As in Section 4.3, the path $\gamma(\tau)$ is discretized by a set of points g_j sampled from it. Thus it is important to justify that the visibility of a continuous path can be approximated by the visibility of a finite sampling of this path.

Theorem 5.3. *Given $\gamma: [0, 1] \mapsto \mathbb{R}^d$ a C^2 simple curve, let ψ be the level set function, with Lipschitz constant K , for the obstacles. Furthermore, let m be a positive integer and set $\tau_j = j/m = j\Delta\tau$ and $g_j = \gamma(\tau_j)$ and ϕ as defined in Formula (3.2). If $\phi(x; \gamma^{(m)}) := \phi(x; \{g_j\}) = \max_{0 \leq j \leq m} \phi(x; g_j)$ and $\phi(x; \gamma) = \max_{t \in [0, 1]} \phi(x; \gamma(t))$, then*

$$(5.3) \quad 0 \leq \phi(x; \gamma) - \phi(x; \{g_j\}) \leq \Delta\tau K \|\gamma'\|_\infty.$$

Proof. Fix x and let k be such that $\phi(x; \{g_j\}) = \max_{0 \leq j \leq m} \phi(x; g_j) = \phi(x; g_k)$, and $\tau^* \in [\tau_l, \tau_{l+1}]$ be such that $\phi(x; \gamma) = \max_{t \in [0, 1]} \phi(x; \gamma(t)) = \phi(x; \gamma(\tau^*))$. Then applying Lemma 3.1, we have

$$0 \leq \phi(x; \gamma(\tau^*)) - \phi(x; \gamma(\tau_l)) \leq (\tau^* - \tau_l) K \|\gamma'\|_\infty.$$

By construction, $\phi(x; \gamma(\tau_l)) \leq \phi(x; g_k) \leq \phi(x; \gamma(\tau^*))$, so

$$0 \leq \phi(x; \gamma) - \phi(x; \{g_j\}) \leq \Delta\tau K \|\gamma'\|_\infty. \quad \square$$

The main steps of the numerical algorithm using a straight forward front tracking approach are thus, in a condensed format:

- (1) Start with a partition $0 = t_0 < t_1 < \dots < t_n = T$, for a chosen final time T , an initial path $\gamma(\tau, t_0)$, and a fixed constant C .
- (2) Discretize this path by placing a grid $\{\tau_j = j\Delta\tau : 0 \leq j \leq m\}$ over $[0, 1]$.
- (3) For $k = 0, 1, \dots, n-1$, and for $j = 1, 2, \dots, m-1$, advance the path by Equation (5.1.1). This involves computation of $\phi(x; \gamma(\tau_j, t_k))$, finite differencing on the gradient, and a chosen ODE solver such as Euler's method. We remark that $\phi(x; \gamma(\tau_j, t_k))$ is the standard visibility function of [17] with a single observer $x_0 = \gamma(\tau_j, t_k)$.

We now discuss the stability of this gradient flow. Note the last term $\lambda\kappa(\tau)n(\tau)$ acts as regularization by flowing the path by λ times curvature. In fact, in 2 dimensions, collecting $\kappa(\tau)n(\tau)$ terms, we get

$$(5.4) \quad \int_{\Omega \setminus D} (\mathcal{X}(x; \gamma) - C)H(\phi(x; \gamma))dx + \lambda.$$

This helps us determine what values we can choose for λ . Since $0 \leq \mathcal{X}(x; \gamma) \leq L(\gamma)$,

$$-C\text{Area}(\Omega \setminus D) \leq \int_{\Omega \setminus D} (\mathcal{X}(x; \gamma) - C)H(\phi(x; \gamma))dx \leq (L(\gamma) - C)\text{Area}(\Omega \setminus D),$$

and so a valid condition would be

$$\int_{\Omega \setminus D} -(\mathcal{X}(x; \gamma) - C)H(\phi(x; \gamma))dx \leq C\text{Area}(\Omega \setminus D) \leq \lambda.$$

If explicit time stepping is used for time integration,

$$\Delta t \leq c_0(\lambda + C\text{Area}(\Omega \setminus D))\Delta \tau^2$$

guarantees numerical stability.

In Figure 5.1, we have two circular obstacles in the first and third quadrants. A straight line joining $(-1, 0)$ and $(1, 0)$, forming γ_0 , is then deformed using the flow (5.1.1). In order to improve efficiency of the algorithm, instead of using a constant $C = 5$ that is our objective average exposure, we increase C gradually along the flow by $C = \max(2 + r_0L(\gamma), 5)$, where $L(\gamma)$ is the length of the path γ . The flow eventually reaches a steady state and the resulting path depicted in Figure 5.1 matches with our intuition of an 'S' shaped curve. In addition, we plotted $\mathcal{X}(x; \gamma_0)/\|\mathcal{X}(\cdot; \gamma)\|_\infty$ and $\mathcal{X}(x; \gamma)/\|\mathcal{X}(\cdot; \gamma)\|_\infty$

Figure 5.2 shows an example in which we have the initial path γ_0 a closed curve passing through the obstacles. We see that the portions of γ away from its fixed end points at $(0, 0.9)$ eventually pull out of the obstacles. The tip in the middle of the path first reaches further down during the descent and is then pulled back due to the regularization term. One can see the corresponding effect in the change in $\|\mathcal{X}(\cdot; \gamma) - 4.2\|_2$.

5.1.3. Weights and Memory. We can modify our approach for acquiring uniform exposure to include weights that measure the importance for the visibility of certain regions of space. A slight modification taking this into account leads to the energy

$$\int_{\Omega \setminus D} (\mathcal{X}(x; \gamma) - C)^2 w(x) dx,$$

where w represents the weight. One extreme example of this is when the surfaces S of the obstacles D are the only regions of interest for visibility. The energy in this case becomes

$$\int_S (\mathcal{X}(x; \gamma) - C)^2 dA = \int_\Omega (\mathcal{X}(x; \gamma) - C)^2 \delta(\psi(x)) |\nabla \psi(x)| dx.$$

We can also consider the more interesting situation of variable weighting in time:

$$\tilde{\mathcal{X}}(t, x; \gamma) = \int_0^t K_\eta(x, t, \tau) H(\phi(x; \gamma(\tau))) |\gamma'(\tau)| d\tau,$$

and

$$\bar{\mathcal{X}}(x; \gamma) = \int_0^1 \tilde{\mathcal{X}}(t, x; \gamma) |\gamma'(t)| dt.$$

Notice that when $K_\eta(t, \tau)$ is replaced by $\delta(t - \tau)$, A by 1, then $\tilde{\mathcal{X}}(t, x; \gamma)$ reduces to $H(\phi(x; \gamma))$ and $\bar{\mathcal{X}}(x; \gamma)$ becomes identical to $\mathcal{X}(x; \gamma)$. If $K_\eta(t, \tau)$ is a function of $t - \tau$ with support size η , e.g. $K_\eta(t, \tau) = 1_{\{0 \leq t - \tau \leq \eta\}}$, we can interpret $\tilde{\mathcal{X}}(t, x; \gamma)$ as the exposure of location x at time t from an observer with finite memory of η duration, mimicking forgetfulness. Correspondingly, the optimization problems can be formulated as in Problem 5.1, with $E(\gamma, C)$ replaced by

$$\tilde{E}(t; \gamma, C) = \|\tilde{\mathcal{X}}(t, \cdot; \gamma) - C\| \text{ or } \bar{E}(\gamma, C) = \|\bar{\mathcal{X}}(\cdot; \gamma) - C\|$$

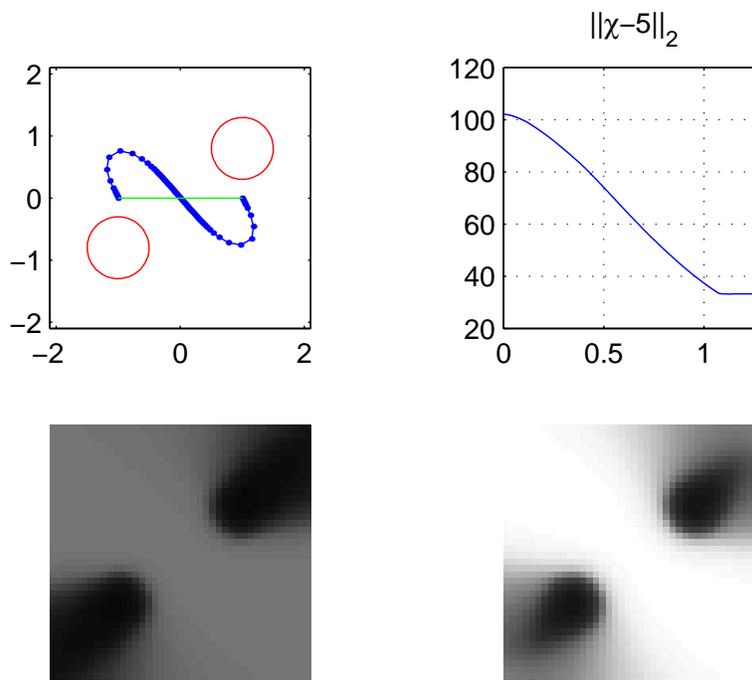


FIGURE 5.1. The upper left figure shows the obstacles (red circles), the initial curve (green circle), and the optimized curve (blue dotted curve). The constant C is chosen to be 5.0, the curvature regularization term is 5. The images in the second row show the exposure of the initial and the optimized paths.

or

$$\int_0^1 \tilde{E}(t; \gamma, C) dt,$$

where $\|\cdot\|$ is some norm.

However, so far we have only considered uniform viewing, ignoring greater visibility. If in addition to uniform viewing, we would like the path to also be balanced with the prevailing desire to maximize visibility, we can enact one the following changes:

- (1) Minimize subject to a constraint of $L(\gamma) = C_0$ or $L(\gamma) \leq C_0$.
- (2) Replace C by a bounded, increasing function of $L(\gamma)$ to force higher exposure levels.
- (3) Add a term such as

$$\alpha \int_{\Omega \setminus D} \left(1 - \frac{X(x; \gamma)}{L(\gamma)} \right) dx = \frac{\alpha}{L(\gamma)} \int_{\Omega \setminus D} (L(\gamma) - X(x; \gamma)) dx$$

to further maximize, with weight α , the total exposure.

- (4) Include a multiplicative penalty term to increase the length of the path:

$$\min_{\gamma, C} \frac{1}{L(\gamma)} \int_{\Omega \setminus D} |X(x; \gamma) - C|^2 dx.$$

In terms of numerical representation, we like to note that we are currently representing the paths using parametrization, with front tracking for its motion. Problems with maintaining an adequate parametrization do occur during the motion. In the future, we would like to use instead an implicit representation such as the level set method [16, 3] or the segment projection method [5].

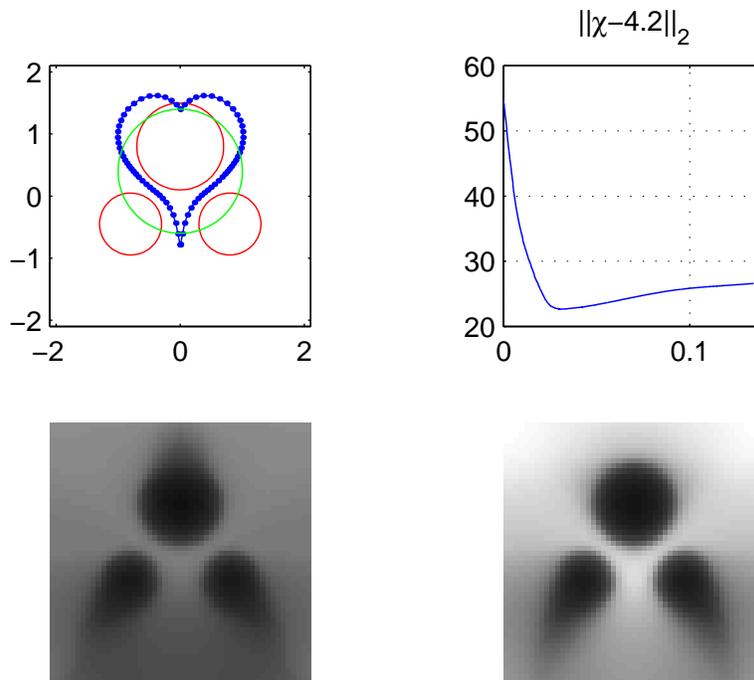


FIGURE 5.2. The upper left figure shows the obstacles (red circles), the initial curve (green circle), and the optimized curve (blue dotted curve). The constant C is chosen to be 4.2, the curvature regularization term is 0.05. The images in the second row show the exposure of the initial and the optimized paths.

5.2. Shortest Path. Maximization of a visible volume function or minimization of a suitably constructed energy are not the only approaches to solving a visibility optimization problem. We consider here the problem of finding the shortest paths observers located at different points can travel to see a chosen point object. It is possible to solve this problem by maximizing visibility with a weight in space that is a smoothed-out approximate delta function centered at the chosen point. However, this is not the best strategy since the observer may be trapped in a local maximum, unacceptable for this problem, and the form of the smoothed-out function needs to be carefully chosen.

Let x_0 be the location of an observer and let y be the location of the point object that we want to be visible. Suppose the nontrivial case where y is initially invisible to x_0 . The ability for x_0 to see y is the same as the ability for y to see x_0 . Thus we consider ϕ_y , the visibility level set function associated to y . If x_0 moves into the $\{\phi(\cdot; y) > 0\}$ region, then y can see x_0 and vice versa. From this, we see that the solution to our problem is the shortest path from x_0 , respecting obstacles, to the region $\{\phi(\cdot; y) > 0\}$, or alternatively the shadow boundary $\{\phi(\cdot; y) = 0\}$. A distance function can be used to find this path.

Let T_y denote the signed distance, or signed traveltime, function to the zero level set of $\phi(\cdot; y)$ in the presence of obstacles. This means T_y evaluated at a point in space is the signed distance of the shortest path from that point to the zero level set of $\phi(\cdot; y)$. This path, of course, must go around obstacles and, in fact, T_y can be taken, as a fixed condition, to be ∞ inside obstacles. This distance function can be easily constructed using a variety of techniques, including PDE, fast marching, and fast sweeping methods [9, 14, 15, 18, 19] that solve

$$|\nabla T_y| = 1,$$

with boundary conditions $T_y = 0$ where $\phi_y = 0$ and $T_y = \infty$ inside obstacles.

With this distance function, the shortest path x_0 needs to take to see y is just the integral curve of ∇T_y drawn from x_0 to the zero level set of ϕ_y . An ODE solver combined with finite differencing and interpolation in a method-of-lines approach can be used to solve for this flow,

$$\partial_t x_0 = \max\{-T_y(x_0), 0\} \nabla T_y(x_0).$$

The $\max\{-T_y(x_0), 0\}$ term ensures that x_0 will slow down and stop at the zero level set and that x_0 will not move if y is already visible.

Note, the shortest paths from many different initial positions x_0 can be determined using the same T_y . Also, even though these paths are all straight lines, our approach applies to more general situations, where the allowable speed is not uniform in space and the paths will bend around regions of slow moving. In this case, a different T_y is constructed from either PDE, fast marching, or fast sweeping methods, solving

$$|\nabla T_y| = f,$$

with the same boundary conditions and a given $f : \Omega \rightarrow \mathbb{R}^+$, where $1/f$ evaluated at a point denotes the allowable speed at that point. The shortest paths are once again the integral curves of ∇T_y .

Figure 5.3 shows a step-by-step example and results of our approach to the shortest path problem.

5.3. Tracking an Object. One could easily concoct a situation in which the observer wishes to keep a moving object in sight for as long as possible. One interpretation of such an objective is to keep the target away from occlusion as much as possible. Let $\phi(y; x_0)$ be the specific level set function for shadow boundaries such that its positive values denote inescapability from the observer's view. Thus, at a point in space, the larger the value of ϕ , the less chance it has to disappear from view when the observer is perturbed. Such a level set function obviously depends on a weighting of the distance a point is away from the shadow boundary with the distance it is away from the observer. It is not our goal in this paper to derive the exact nature and form of this function. Thus we consider here the simplification of having the inescapability of an object at y approximated by

$$I(y; x_0) = -\frac{1}{2}|x_0 - y|^2 + \lambda\phi(y; x_0),$$

where $\phi(y; x_0)$ is computed from our visibility algorithm with the obstacle level set functions ψ as signed distance functions. Thus, maximizing this expression for x_0 will give the safest position for the observer x_0 to be in to keep y in sight of x_0 . Our interest is actually in a moving object, $y(t)$, and the determination of how an initially placed observer should move with it to keep it safely in sight.

Adding the time variable into the expression for inescapability and performing gradient ascent on $I(y; x_0)$ with respect to x_0 , we arrive at a dynamics for the observer:

$$\dot{x}_0 = (y - x_0) + \lambda \nabla_{x_0} \phi(y; x_0).$$

In this formulation, $-I(y; x_0) = |x_0 - y|^2 - \lambda\phi(y; x_0)$ is the potential energy of the system and $\phi(y; x_0)$ is analogous to gravitation field. The twist is that the potential field is driven by the target location $y(t)$. One can see that the observer must balance a desire to get closer to y with the influence of how much $\phi(y; x_0)$ changes with respect to any motion. Note λ should be chosen to vary such that the object is always in the visible region of the observer.

Another term to consider instead of, or in addition to, $\frac{1}{2}|x_0 - y|^2$ is $U(d_D(x_0, y))$, where U is a bounded increasing function and $d_D(x_0, y)$ is the distance in the presence of the obstacles D of y from x_0 . This term contributes the term

$$-U'(d_D(x_0, y)) \nabla_{x_0} d_D(x_0, y)$$

to the velocity of the observer. Note $d_D(x_0, y)$ can be computed efficiently by solving $|\nabla_{x_0} d_D(x_0, y)| = 1$ in $\Omega \setminus D$ with boundary condition $d_D(x_0 = y, y) = 0$, as in [9, 14, 15, 18, 19].

We present simulations in the selected snapshots of Figures 5.4 and 5.5. In Figure 5.4, the dotted lines show the shadow boundary 0.3 time units in the past. The shaded region is the current invisible region for

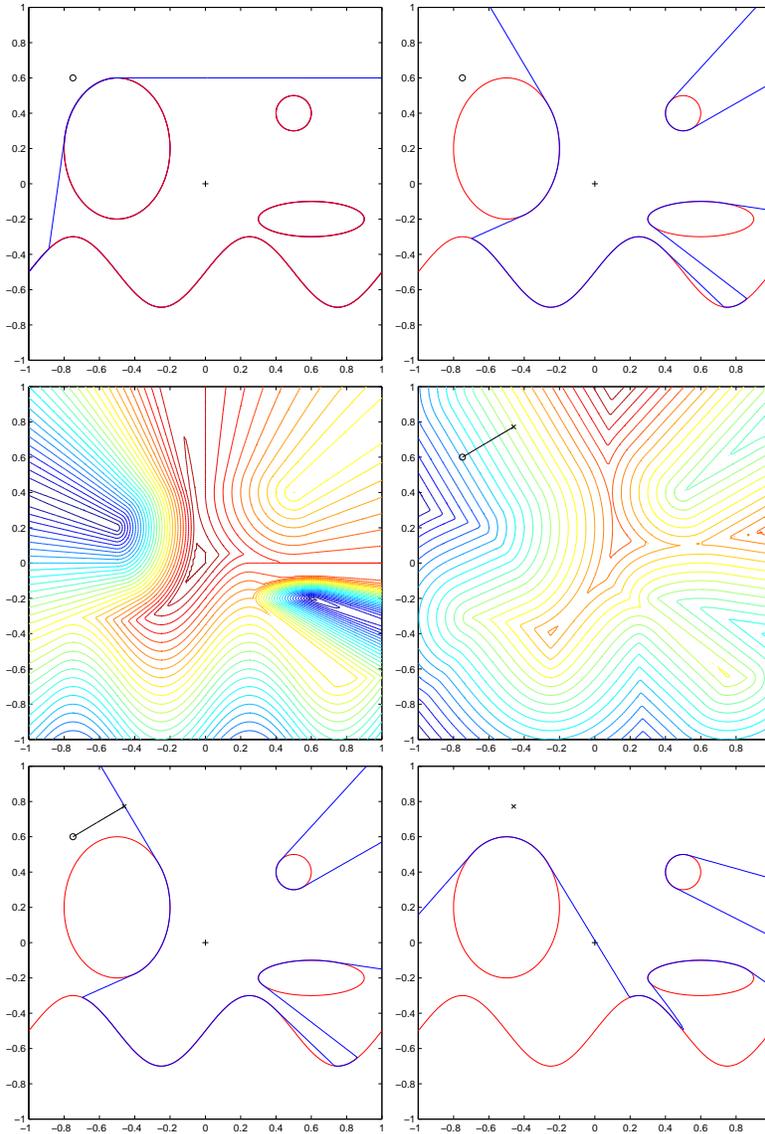


FIGURE 5.3. The figures in the first row show the observer, obstacles, shadow boundary, and a point at the origin that is invisible, and then the first step of our procedure, namely treating the origin as the observer. The figures of the second row show the contours of the visibility level set function and its redistancing into signed traveltime, along whose gradient directions the observer will travel. The figures of the third row show the shortest path of the observer to the shadow boundary and the visibility regions of the observer at the final location where the origin is visible on the shadow boundary.

the observer. The past trajectory of the evader is shown and the past non-trivial trajectory of the observer is computed. Note the observer follows the evader around the circular obstacle to get a clear view. Figure 5.5 tests the importance of the gradient of the visibility level set function. When the visibility gradient term is removed, the evader vanishes behind an obstacle. On the other hand, when the term is included, the evader is always kept in sight by the observer.

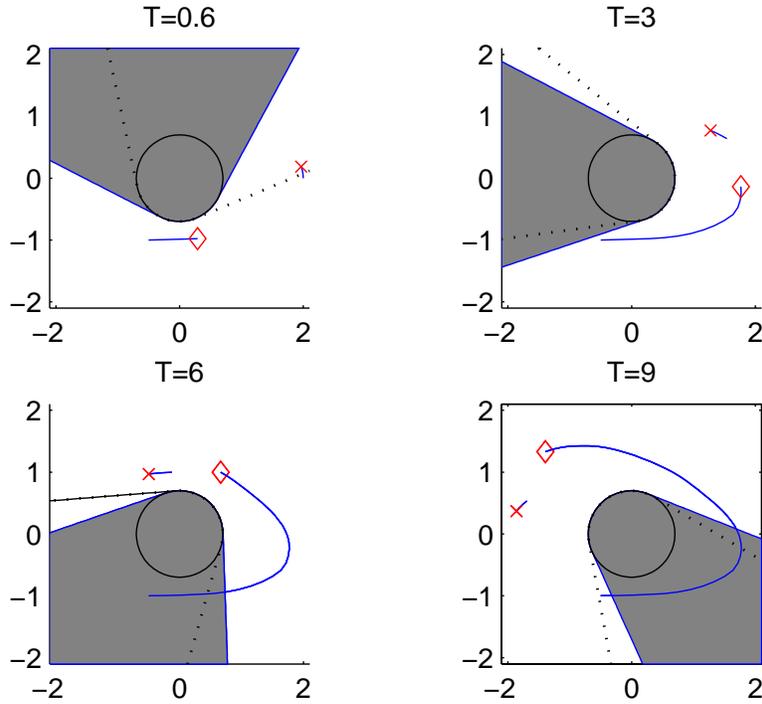


FIGURE 5.4. Past trajectories of observer and evader are shown. The diamonds and crosses indicate the current locations of the observer and the evader, respectively. Thus, the observer circles around the obstacle chasing the evader.

6. CONCLUSION

In this work, we have outlined two basic strategies that apply to several optimization problems involving visibility. One strategy involves producing a visible volume function whose maxima are the desired locations for our observers to maximize visibility. To determine local maxima of this function, we advocated the use of visibility level set functions, level set volume formulations, and gradient flows. These effectively combine, due in no small part to the continuity of visibility information afforded by the visibility level set framework, to create numerical algorithms for a variety of optimization visibility test cases dealing with multiple observers, spatial regions of interest, memory effects, and human visual detail. The other strategy involves the construction of an energy whose minimum achieves the desired effect. This strategy was used to allow for more uniform viewings of space, and the energies can be modified to fit into situations mimicking forgetfulness and other constraints. In future work, we target pursuer-evader games and other more complicated, more realistic applications based on visibility.

7. ACKNOWLEDGEMENTS

The authors would like to thank Stanley Osher for his encouragement in pursuing this direction of research. The first author is partially supported by NSF grant #0208449. The work of the second author is partially supported by the National Science Foundation under agreement No. DMS-0111298.

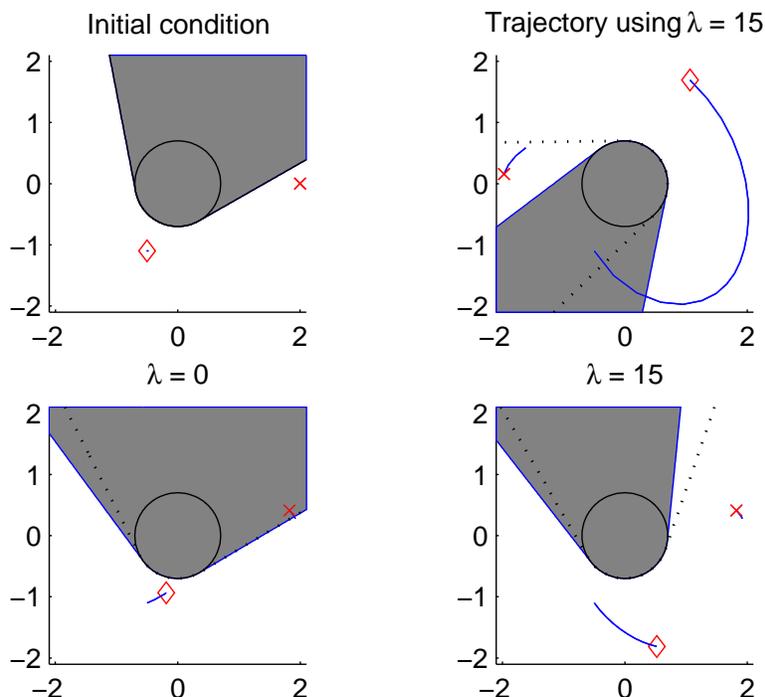


FIGURE 5.5. Past trajectories due to the absence and presence of the visibility gradient term are shown. of observer and evader. The diamonds and crosses indicate the current locations of the observer and the evader, respectively. The lower left plot is with the absence of the gradient and should be compared to the lower right plot which contains the gradient term. The upper right plot is a longer time simulation when the gradient term is present.

REFERENCES

- [1] D. Adalsteinsson and J.A. Sethian. An Overview of Level Set Methods for Etching, Deposition, and Lithography Development. *IEEE Transactions on Semiconductor Devices*, 10(1), February 1997.
- [2] O.P. Bruno and F. Reitich. A New Approach to the Solution of Problems of Scattering by Bounded Obstacles. *SPIE*, 2192:20–28, 1994.
- [3] T. Cecil and D. Marthaler. A Variational Approach to Search and Path Planning Using Level Set Methods. CAM Report, UCLA, 2004.
- [4] F. Durand. *3D Visibility: Analytical Study and Applications*. PhD thesis, Universite Grenoble I-Joseph Fourier Sciences et Geographie, 1999.
- [5] B. Engquist, O. Runborg, and A.-K. Tornberg. High Frequency Wave Propagation by the Segment Projection Method. *J. Comput. Phys.*, 178(2):373–390, 2002.
- [6] B. Engquist, A.-K. Tornberg, and Y.-H. Tsai. Discretization of Dirac Delta Functions in Level Set Methods. CAM Report 04-15, UCLA, 2004.
- [7] H.H. Gonzalez-Banos, L.J. Guibas, J.C. Latombe, S.M. LaValle, D. Lin, R. Motwani, and C. Tomasi. Motion Planning with Visibility Constraints: Building Autonomous Observers. In Y. Shirai and S. Hirose, editors, *Robotics Research – The Eighth International Symposium*, pages 95–101. Springer, 1998.
- [8] L.J. Guibas, J.C. Latombe, S.M. LaValle, D. Lin, and R. Motwani. A Visibility-Based Pursuit-Evasion Problem. *International Journal of Computational Geometry and Applications*, 9(5):471–494, 1999.
- [9] J. Helmsen, E. Puckett, P. Colella, and M. Dorr. Two New Methods for Simulating Photolithography Development in 3D. *Proc. SPIE*, 2726:253–261, 1996.
- [10] G.S. Jiang and D. Peng. Weighted ENO Schemes for Hamilton Jacobi Equations. *SIAM J. Scient. Comput.*, 21(6):2126–2143, 2000.
- [11] S. Osher and R. Fedkiw. *Level Sets and Dynamic Implicit Surfaces*. Springer-Verlag, 2002.

- [12] S. Osher and J.A. Sethian. Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations. *J. Comput. Phys.*, 169(1):12–49, 1988.
- [13] S. Osher and C.-W. Shu. High order essentially non-oscillatory schemes for Hamilton-Jacobi equations. *SINUM*, 28:907–922, 1991.
- [14] D. Peng, B. Merriman, S. Osher, H.K. Zhao, and M. Kang. A PDE-Based Fast Local Level Set Method. *J. Comput. Phys.*, 155(2):410–438, 1999.
- [15] J.A. Sethian. Fast Marching Level Set Methods for Three Dimensional Photolithography Development. *Proc. SPIE*, 2726:261–272, 1996.
- [16] C.-W. Shu and S. Osher. Efficient Implementation of Essentially Nonoscillatory Shock-Capturing Schemes. *J. Comput. Phys.*, 77(2):439–471, 1988.
- [17] Y.-H. Tsai, L.-T. Cheng, P. Burchard, S. Osher, and G. Sapiro. Dynamic Visibility in an Implicit Framework. To appear in *J. Comp. Phys.*
- [18] 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.*, 41(02), 2003.
- [19] J.N. Tsitsiklis. Efficient Algorithms for Globally Optimal Trajectories. *IEEE Transactions on Automatic Control*, 50:1528–1538, 1995.

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF CALIFORNIA SAN DIEGO, LA JOLLA, CA 92093

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF TEXAS AT AUSTIN, TEXAS, TX 78712