

## VISIBILITY OPTIMIZATION USING VARIATIONAL APPROACHES\*

LI-TIEN CHENG<sup>†</sup> AND YEN-HSI TSAI<sup>‡</sup>

**Abstract.** Constructing the visible and invisible regions of an observer due to the presence of obstacles in the environment has played a central role in many applications. It can also be a first step. In this paper, we adopt a visibility algorithm that can produce a variety of general information to handle the optimization of visibility information. Through the use of level set tools, gradient flow, finite differencing, and solvers for ordinary differential equations, we introduce a set of distinct algorithms for several model problems involving the optimization of visibility information.

**Key words.** Visibility, level set method, variational approach and path planning.

**AMS subject classifications.** 65, 49, 35

### 1. Introduction

**1.1. Statement of problem.** 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 a 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 through application of the approach introduced in [12] which, though replaceable, provides enough information to handle each of the problems.

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

- *What is the optimal location  $x_0$  for an observer such that the volume of the visible region in  $\Omega$  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 adds to the first:

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

One can interpret the last question as an extension of the second with an 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, 6, 7]).

In many situations, it is useful to think of the observer as a light source. Consequently, our attempt to solve the three central questions amounts to maximizing the illuminated regions in  $\Omega$ , or the average illumination (exposure) in  $\Omega$ . The core of this work revolves around adaptivity in observer locations and optimization related to the corresponding visibility. Thus, it does not detract from the essence of our study

---

\*Received: April 6, 2005; accepted (in revised version): July 12, 2005. Communicated by Shi Jin.

<sup>†</sup>Department of Mathematics, University of California, San Diego, La Jolla, CA 92093.

<sup>‡</sup>Department of Mathematics, University of Texas at Austin, Texas, TX 78712.

to disregard global optimality and accept local maxima as suitable solutions, with gradient flow a valid process.

**1.2. Visibility information.** In order to optimize visibility information, one must first be able to calculate that information. A plethora of options are available, most arising from computational geometry and some from partial differential equations. For a chosen application, certain ones may be preferable; for example, a user may be interested only in a limited amount of visibility information, or have hardware constraints, or emphasize speed over accuracy, or vice versa. Since we are not concentrating on a specific application, we seek one that provides sufficient information that allows for accurate numerical differentiation and integration in the whole space. This may be non-optimal for some specialized graphics applications but will allow us to use the same algorithm independent of application and especially in situations outside of the usual context of computer graphics and vision. Other approaches can be substituted when specific applications are of interest. Thus, we choose to use the algorithm introduced in [12] to generate our visibility information.

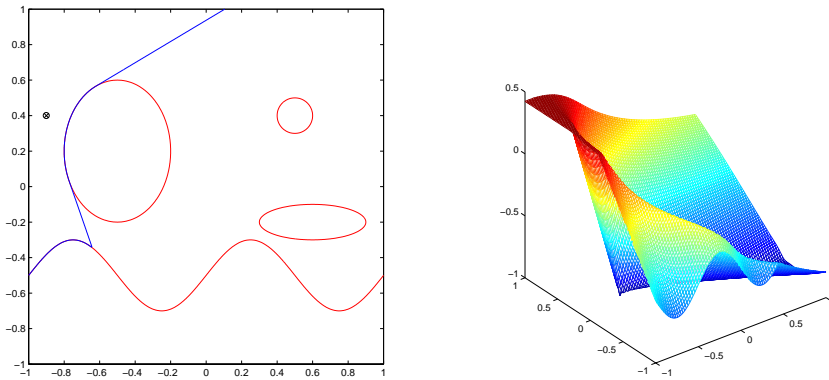


FIG. 1.1. For the obstacle environment, given observer location, and shadow boundary of plot on the left, the visibility level set function is shown in the plot on the right.

This algorithm works with an implicit representation of obstacle surfaces and generates an implicit description of the visibility information with a PDE-based method (see Figure 1.1). The advantage is, as mentioned above, in the substantial amount of information that can be extracted from this description. Under simple operations, one can obtain accurate location of shadow boundaries, including their normal vectors, curvatures, surface areas, and even the distance of points away from them; accurate location of visible and invisible regions, including their volumes; and a partial differential equation viewpoint for analysis. The disadvantage is in the inefficiency of the approach in typical computer graphics applications such as rendering, where only visibility of the two dimensional obstacle surfaces, as opposed to the three dimensional space, is of interest; or when existing hardware allows for fast polygon manipulations as opposed to current implicit surface operations; or when geometry, especially quantities involving derivatives, are not used. However, for the purposes of this paper, the algorithm of [12] is adequate.

We will in addition be using the notation of [12]. Let  $D$  represent given obstructions in the space  $\Omega$  and  $x_0$  the location of the observer. Let  $\psi$  be the level set function representing  $D$ , negative inside  $D$ . Working along rays emanating from the observer,

but in an implicit manner, the algorithm generates the level set function  $\phi(x)$  that encodes the visibility of  $x$  from the observer  $x_0$ . In fact, this function, which we call the visibility level set function, takes the analytic form

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

where  $\mathcal{L}(x, x_0)$  is the integral curve of the vector field  $r(x)$  of rays connecting  $x$  and  $x_0$ .

As an outline, Section 2 introduces an algorithm that answers our first question involving one observer and Euclidean space. Section 3 includes several subsections devoted to questions in the flavor of our second one. Section 4 presents a few visibility optimization problems that require different techniques than those considered up to that point. Included here will be our attempt at answering the third question. Finally, Section 5 acknowledges the contributions of others in this investigation.

## 2. Single point visibility optimization

To facilitate our discussions of adaptivity in the vantage position, we augment the visibility function  $\phi$  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.  $\phi(\cdot; x_0)$  is the level set function representing visibility in a bounded domain  $\Omega$  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. Consider the visible volume function  $V(x_0)$  defined as the volume of the visible region for the observer at  $x_0$ . We have the equality

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

where  $H$  denotes the one-dimensional heaviside function. We shall show later that the values of this volume in fact change continuously with the vantage point. With this, the problem of interest becomes that of finding, using calculus, 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  $\nabla_{x_0} V(x_0)$ , and we note that  $V$  is Lipschitz continuous under certain conditions that we present later in the paper. Thus the gradient ascent flow of the observer location is described by

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

The  $\nabla_{x_0}$  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 [12] 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 [12] to obtain the  $2d$  visibility level set functions  $\phi_{x_0 \pm h e_i}$  over a grid in  $\Omega$  for each  $i = 1, \dots, d$ .
3. Evaluate  $V(x_0 \pm h e_i)$  for each  $i = 1, \dots, d$  using a smoothed-out approximate heaviside function and numerical integration techniques such as the trapezoidal rule over the grid. We note the ideas of [5] 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 considered, we notice that  $D_0^h H(\phi(y; x_0))$  happens to be zero in a large portion of  $\Omega$ . In other words, under small perturbations of the position of the observer, the visible and invisible regions, and, in fact, the visibility level set function, will undergo similarly small changes. This can be seen mathematically in the expansion

$$\nabla_{x_0} V(x_0) = \int_{\Omega} \nabla_{x_0} H(\phi(y; x_0)) dy = \int_{\Omega} \delta(\phi(y; x_0)) \nabla_{x_0} \phi(y; x_0) dy, \quad (2.1)$$

where  $\delta$  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.

At this point, we must remark that  $V$ , though Lipschitz continuous, as shall be shown later, is in general not differentiable everywhere. Care must be taken in defining and choosing an adequate gradient of the function at such locations. In this work, we use central differencing for simplicity; however, we note that at certain, though not all, locations where the visible volume function is not differentiable, central differencing on the gradient may not point in the direction of steepest ascent. To give an example of what is involved in fixing this problem, one can define the gradient on a smooth version of the visible volume function; however, it would be inefficient to

always perform smoothing since the gradient is well-defined almost everywhere. We would like to fully remedy this issue in a future work and continue with standard finite differencing techniques in this paper.

The visibility level set function  $\phi(y; x_0)$  takes the form

$$\phi(y; x_0) = \min_{z \in \mathcal{L}(y, x_0)} \psi(z), \tag{2.2}$$

as stated previously. In the cases considered in this paper, the vector field  $r(x)$  is simply  $(x - x_0)/|x - x_0|$ , corresponding to straight line segments for  $\mathcal{L}(y, x_0)$ . The following Lemma, with this vector field, shows that  $\phi(y; x_0)$  is Lipschitz continuous in  $x_0$  and so the central differencing approximation for  $\nabla_{x_0} \phi(y; x_0)$  in (2.1) will remain a bounded quantity.

LEMMA 2.1. *Let  $K$  be a Lipschitz constant for  $\psi$  in  $\Omega$ , and  $\phi$  be defined as in (2.2). 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  $\phi$  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  $|\phi(y; x_0) - \phi(y; \tilde{x}_0)| \leq K|x_0 - \tilde{x}_0|$ . □

This in fact implies  $V(x_0)$  is Lipschitz continuous in  $x_0$ , under conditions that there is no fattening in the level set function  $\phi$  near its zero level set. In detail, the Lipschitz continuity of  $\phi$  in  $x_0$  says

$$-K|\epsilon| \leq \phi(x; x_0 + \epsilon) - \phi(x; x_0) \leq K|\epsilon|.$$

In the case where  $x$  is on the zero level set of  $\phi(\cdot; x_0 + \epsilon)$ , this reduces to

$$-K|\epsilon| \leq \phi(x; x_0) \leq K|\epsilon|,$$

which means the zero level set of  $\phi(\cdot; x_0 + \epsilon)$  is a subset of  $\{x \in \Omega \mid |\phi(x; x_0)| \leq K|\epsilon|\}$ . Thus

$$\begin{aligned} |V(x_0 + \epsilon) - V(x_0)| &\leq \int_{\{x \in \Omega \mid |\phi(x; x_0)| \leq K|\epsilon|\}} dx \\ &= \int_{-K\epsilon}^{K\epsilon} \int_{\{x \in \Omega \mid \phi(x; x_0) = \eta\}} \frac{1}{|\nabla_x \phi(x; x_0)|} dx d\eta, \end{aligned}$$

using the coarea formula. We may then conclude the Lipschitz continuity of  $V$ ,

$$|V(x_0 + \epsilon) - V(x_0)| \leq C|\epsilon|$$

if a condition such as

$$\text{ess inf}_{x \in W} |\nabla_x \phi(x; x_0)| > 0,$$

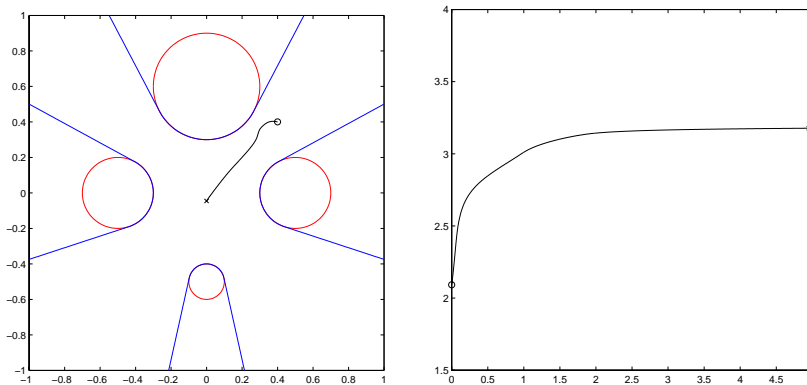


FIG. 2.1. The figure on the left shows the path the observer travels to optimize for the final position visibility. This path originates at 'o' and ends at 'x'. The obstacles to the vision of the observer are four disks of various radii. Also drawn are the shadow boundaries separating the regions of 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.

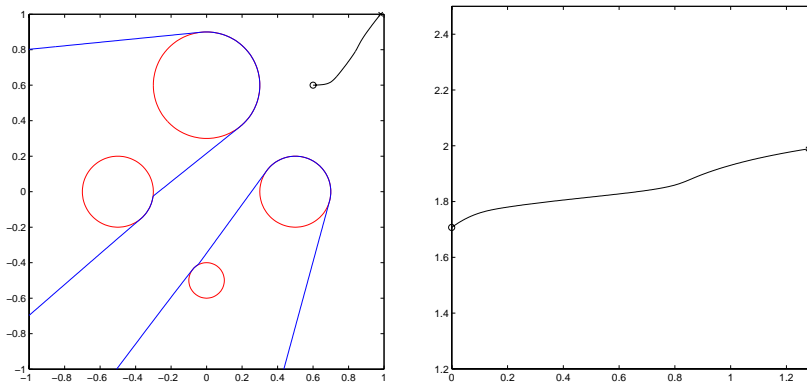


FIG. 2.2. This figure shows has the same setting as in Figure 2.1 with a different initial location for the observer.

is satisfied for all  $x_0 \in \{x \in \Omega | \psi(x) > \delta\}$ , for any constant  $\delta > 0$ , and some open set  $W$  around the zero level set of  $\phi$  where  $|\nabla_x \phi| \neq 0$ . This is related to the level sets of  $\phi$  being lower dimensional than the ambient space, and is thus a no fattening condition.

We present two examples of single point visibility optimization using different starting locations for the observer. Figure 2.1 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 2.2 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 and we chose to stop the computation when it hit the boundary of  $\Omega$ . Finally, Figure 2.3 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. This explains the observer

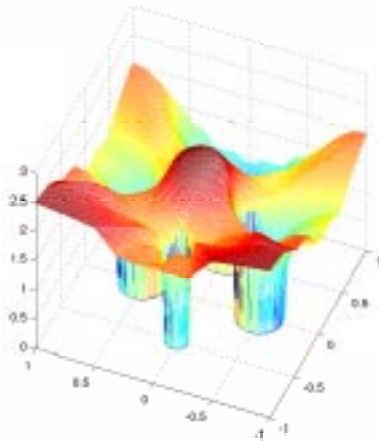


FIG. 2.3. The area of visible regions plotted with respect to different observer locations in the settings of Figures 2.1 and 2.2. Note when the observer is located inside a solid obstacle, the area is zero.

of Figure 2.1 converging to the local maximum at the origin and the observer of Figure 2.2 going away toward the boundary.

We would like to note at this point that the obstacle environment considered in these examples is non-trivial for two reasons. One reason is that though all the obstacles are circular, the presence of multiple obstacles presents topologically complicated visible and invisible regions as well as shadow boundaries. In fact, note the invisible region shown in Figure 2.1 has four connected components while that shown in Figure 2.2 has only two. On top of that, this topology changes with the dynamics of the observer. The second reason is that the paths travelled by the observer and the observer's final position are impossible to guess a priori. Thus even though only trivial circular obstacles appear, the environment produces non-trivial solutions, in a non-trivial process, to visibility optimization. We use the same obstacle environment for our two dimensional illustrative examples for easier comparison of results and since even the more complicated shapes such as that considered in Figure 3.6 produce shadow boundaries consisting of just obstacle boundaries and straight lines. Three dimensional examples such as those of Figure 3.9 and 4.6 are included to answer questions pertaining to our algorithms' abilities in truly complicated and physically realistic environments.

### 3. Extensions for visibility optimization problems

The previous section provided an answer for our first and most basic question concerning optimization and visibility. From its answer, we may consider extensions. In this section, we look at visibility problems involving multiple observers, weighted regions of importance in space, accumulating visibility information, and weighted observer distances.

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. 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.

**3.1. 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 locations 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 using 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.

In the level set framework, there is a simple analogy to unions and intersections. For two level set functions  $\phi_1$  and  $\phi_2$ , 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  $\phi_1$  and  $\phi_2$ . Similarly, the intersection of the negative regions of  $\phi_1$  and  $\phi_2$  is implicitly captured as the negative region of  $\max\{\phi_1, \phi_2\}$ , and its positive region is the union of the positive regions of  $\phi_1$  and  $\phi_2$ .

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  $\Omega$  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  $\nabla_{x_i}$  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 vantage positions for  $x_0, x_1, \dots, x_m$ .

In Figure 3.1, 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 the inevitable presence of 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. Thus, this is an example where the visibility level set function is only continuous, and not



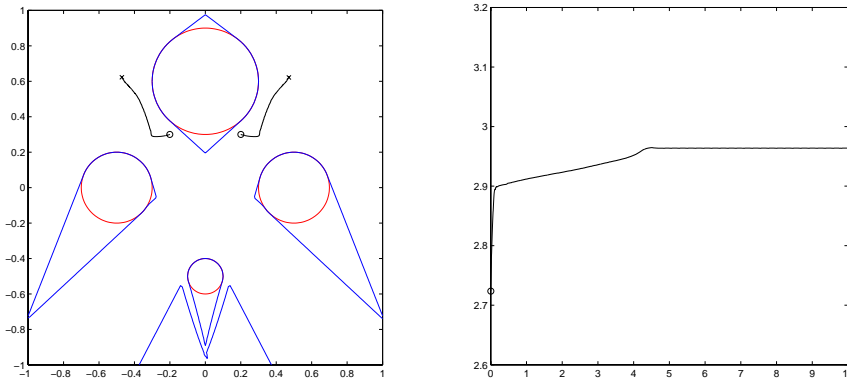


FIG. 3.1. Two observers following gradient flow converge to positions for optimal viewing of the space. Note the figure on the right shows a slight decrease in the area of the visible region calculated along the path of the observer.

differentiable everywhere, near its shadow boundary. Figure 3.2 shows the result of different initial observer placement. The final positions are different and the area of the region of visibility is smaller. Finally, Figure 3.3 shows results with four and five differently situated observers in the obstacle environment.

In Figure 3.4, three robots are placed randomly on a circular orbit at initial time and confined to stay on that circle. 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 more the ensemble of the three robots can obtain. 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.

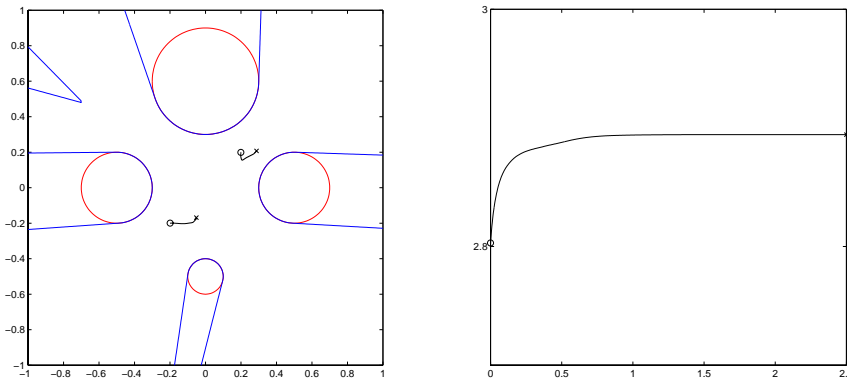


FIG. 3.2. Different starting locations for the observers lead to different solutions. In this case, less of the space is visible to the observers than for the final positions found in Figure 3.1.

**3.2. 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  $\Omega$ . 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.

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 leads to large values for 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 3.5 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 prefer to move toward the origin, as indicated in Figure 2.1. The chosen weight leads to substantial differences in both observer dynamics and optimal position. Figure 3.6 shows a complicated obstacle shape taken from an image. A Gaussian importance weight is centered at the left wall of the square  $\Omega$ . 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.

We further note a more extreme case of the use of weights. Certain computer graphics applications are solely interested in visibility of obstacle surfaces. Consider the surface area function

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

where  $S$  denotes the obstacle surfaces. Maximization of this function maximizes visibility of  $S$ , the obstacle surfaces. In practice, we may 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 [5] to satisfy accuracy requirements.

**3.3. 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 the dynamics 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.

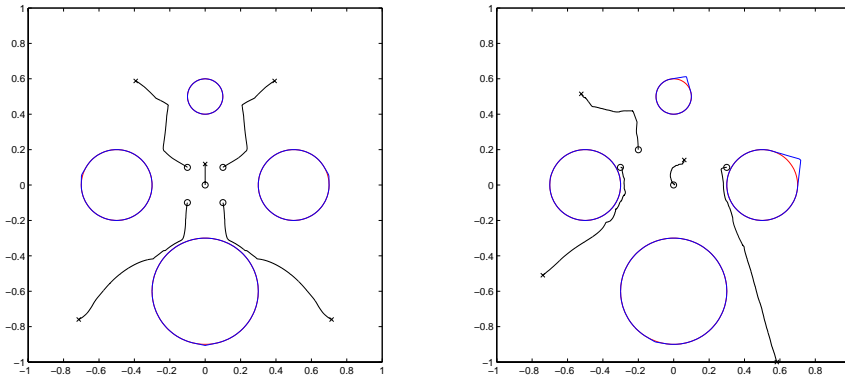


FIG. 3.3. Two more results of multiple observer visibility are shown. In the left plot, five observers move to a position where almost everything can be seen. In the right plot, four observers maximize their visibility until one tries to escape  $\Omega$ .

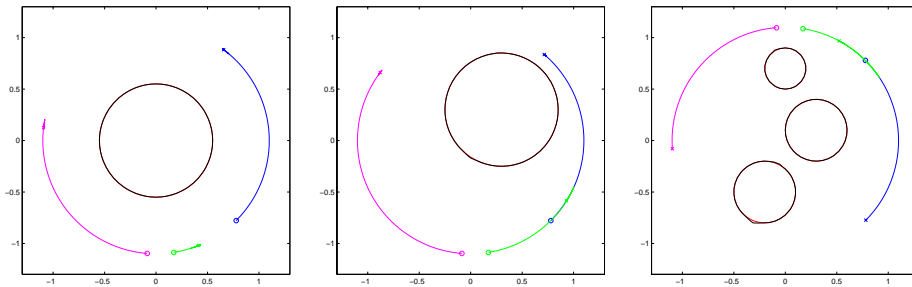


FIG. 3.4. Three observers confined to a circular orbit around given obstacles, different for each of the three figures.

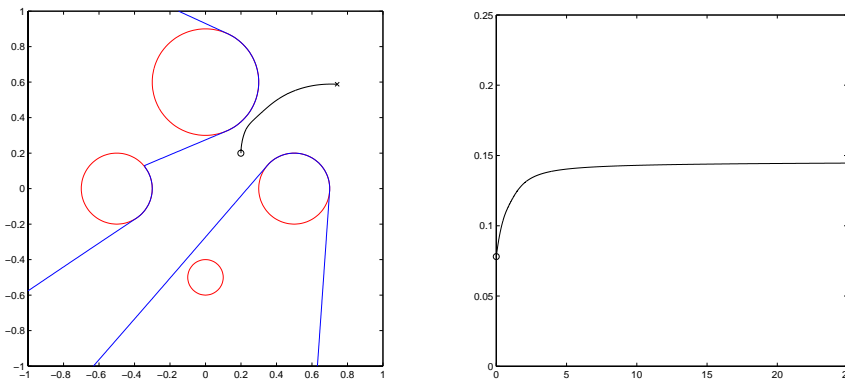


FIG. 3.5. The figure on the left shows the path of an observer that places particular importance in viewing the area surrounding the point  $(1,0.2)$ . The values calculated in the figure on the right represent values of the weighted visible volume function and hence are only related to area.

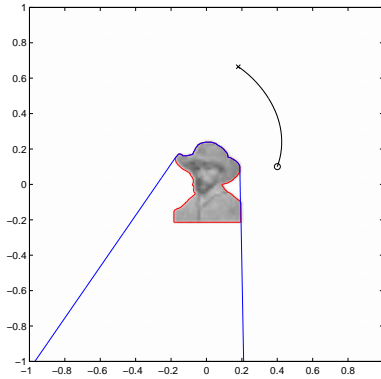


FIG. 3.6. In this figure, an image serves as the obstacle and visibility of the left wall of the square  $\Omega$  is of importance. The observer moves from behind the obstacle to maximize its visibility of the wall, coming to a halt when the wall is completely visible.

To create the path and flow of interest, we record the accumulated visibility information at each time and use it to determine the maximizing direction and speed the observer should take at that time. Then, as the observer moves in this direction, visibility information is updated accordingly.

Let  $\gamma$  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  $\phi_\gamma$ , whose visible regions are the union of visible regions for individual observers located along  $\gamma$ . Note, thus the accumulated visibility information comes from visibility information gathered by multiple, in this case infinite, observers along  $\gamma$ . In the following, we may refer to 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 observers on  $\gamma$  and at  $x_0$ . If we consider  $\nabla_{x_0} V_\gamma(x_0)$ , it tells us the direction and speed an observer located at  $x_0$  should take to maximize visibility when observers located along  $\gamma$  are present. If  $x_0$  is specifically chosen to be the later endpoint of  $\gamma$ ,  $\nabla_{x_0} V_\gamma(x_0)$  then gives the direction and speed for an observer continuing along the path  $\gamma$  to maximize its accumulated visibility.

Note, exact evaluation of  $\phi(\cdot; \gamma)$  is difficult due to the fact that  $\gamma$  is composed of an infinite number of points. We handle this by taking instead a discrete sampling of points of  $\gamma$ ,  $\{z_0, z_1, \dots, z_N\}$ . The visibility level set function of interest can then be approximated by the 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 4.1.

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

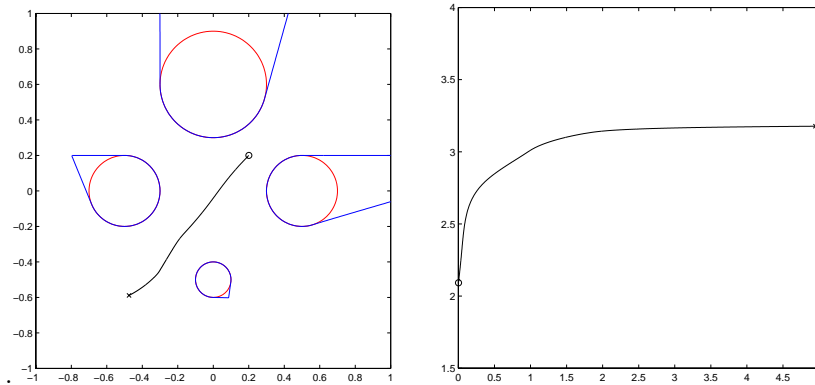


FIG. 3.7. This figure shows the optimizing path of an observer that records visibility information as it moves. The shadow boundaries now those from visibility information culled from the memory of the observer.

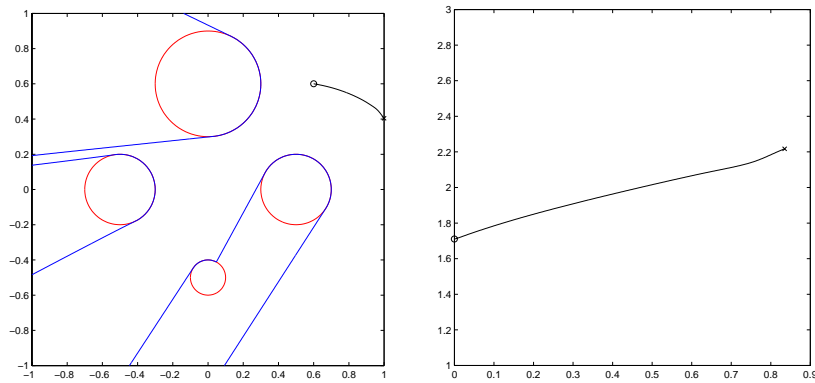


FIG. 3.8. This figure shows a different initial placement of the observer. In order to optimize visibility, in this case it chooses to run towards infinity, even with memory effects present.

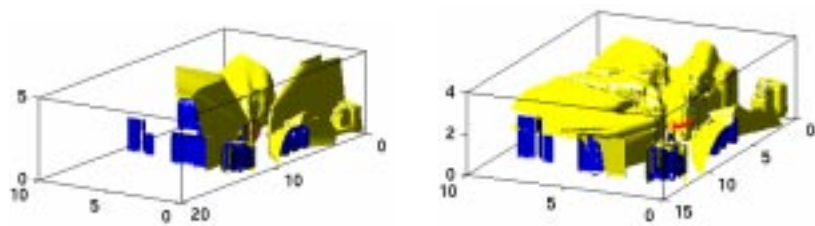


FIG. 3.9. This figure shows different views of the observer path for maximizing accumulated visibility among the buildings of a city in three dimensions.

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 3.7 shows the effects of memory on an observer trying to maximize what it sees during its motion. Note most of space becomes visible to the observer at one time or another during the course of its flow. Figure 3.8 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  $\Omega$ . Figure 3.9 shows different views of a computation involving an observer among obstacles in three dimensional space. The obstacles used here are actually the buildings of a digital city.

**3.4. Note on weights and the observer.** One perhaps undesirable phenomenon we observe in our examples so far is that occasionally the farther away the observer is, the more it can see (see, e.g., Figures 2.2 and 3.8). Thus, in many situations, the position of the observer giving a local maximum for visibility will lie on the boundary of  $\Omega$ . However,  $\Omega$  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  $\Omega$  to be the square computational domain  $[-1, 1] \times [-1, 1]$  in our two dimensional examples and rectangular boxes in our three dimensional ones. Thus, in many situations, if the computational domain is expanded, the optimal location of the observer changes with it, preferring to head off to infinity for a local maximum. In terms of human visibility, as well as in many other situations, such a concept is not natural because distance obscures visual detail. To model this, we can 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 use it in the form  $w_{x_0}(|x - x_0|)$  to prescribe 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. Incorporating such a weight into the visible volume function penalizes the observer for being too far from what it wishes to see, represented by another weight function  $w$ , rendering distant regions virtually invisible. Thus the weighted single point visible volume function can be changed 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 have restricted 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 enters this problem and, though this issue may not be difficult and is probably application dependent, we will not consider it 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 see in detail everything in its visual field.

**4. Other types of visibility optimization problems**

In the previous section, we considered extensions of visibility optimization for a single point observer. All of these 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 other approaches for finding paths that allow a more uniform viewing of space as well as shortest paths to visibility.

**4.1. 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  $\tau$ ,

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

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

**4.1.1. Uniform exposure.** Points outside of obstacles can be said to be viewed in a more uniform manner by an observer moving along  $\gamma$  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 4.1.** *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*

$$E(\gamma, C) = \frac{1}{2} \int_{\Omega \setminus D} (\mathcal{X}(x; \gamma) - C)^2 dx + \lambda \int_0^1 |\gamma'(\tau)| d\tau \tag{4.1}$$

*is minimized.*

Notice that the last term in (4.1) is  $\lambda$  times the length of the curve,  $L(\gamma)$ , and seeks to stabilize the problem, when  $\lambda$  is chosen large enough, by penalizing against fractal or space-filling paths. This, however, is at the expense of minimizing the first term.

**PROPOSITION 4.2.** *The Euler-Lagrange equation for Problem 4.1 can be written as*

$$\int_{\Omega \setminus D} (\mathcal{X}(x; \gamma(\tau)) - C) [\delta \circ \phi(x; \gamma(\tau)) \mathcal{P}_{\gamma'(\tau)} \nabla_{x_0} \phi_{\gamma(\tau)} - H \circ \phi(x; \gamma(\tau)) \kappa(\tau) n(\tau)] dx - \lambda \kappa(\tau) n(\tau) = 0,$$

for every  $\tau \in (0, 1)$ , and any given  $C$ . Here,  $\mathcal{P}_w v$  denotes the projection of  $v$  onto the plane with normal vector  $w$  and  $\kappa(\tau)n(\tau)$  denotes the curvature vector.

*Proof.* For the path  $\gamma$ , we consider the formal derivation through variational calculus in parts. First,

$$\begin{aligned} & \left. \frac{d}{ds} \right|_{s=0} \mathcal{X}(x; \gamma + s\eta) \\ &= \left. \frac{d}{ds} \right|_{s=0} \int_0^1 H \circ \phi(x; \gamma(\tau) + s\eta(\tau)) |\gamma'(\tau) + s\eta'(\tau)| d\tau \\ &= \int_0^1 \delta \circ \phi(x; \gamma(\tau)) \nabla_{x_0} \phi(x; \gamma(\tau)) \cdot \eta(\tau) |\gamma'(\tau)| d\tau + \int_0^1 H \circ \phi(x; \gamma(\tau)) \frac{\gamma'(\tau)}{|\gamma'(\tau)|} \cdot \eta'(\tau) d\tau. \end{aligned}$$

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

$$- \int_0^1 \left[ \delta \circ \phi(x; \gamma(\tau)) \nabla_{x_0} \phi(x; \gamma(\tau)) \gamma'(\tau) + H \circ \phi(x; \gamma(\tau)) \left( \frac{\gamma'(\tau)}{|\gamma'(\tau)|} \right)' \right] \cdot \eta(\tau) d\tau.$$

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

$$\begin{aligned} & \int_0^1 \left[ \delta \circ \phi(x; \gamma(\tau)) \left( \nabla_{x_0} \phi(x; \gamma(\tau)) - \nabla_{x_0} \phi(x; \gamma(\tau)) \cdot \frac{\gamma'(\tau)}{|\gamma'(\tau)|} \frac{\gamma'(\tau)}{|\gamma'(\tau)|} \right) \right. \\ & \quad \left. - H \circ \phi(x; \gamma(\tau)) \frac{1}{|\gamma'(\tau)|} \left( \frac{\gamma'(\tau)}{|\gamma'(\tau)|} \right)' \right] \cdot \eta(\tau) |\gamma'(\tau)| d\tau, \end{aligned}$$

which can be simplified to arrive at

$$\begin{aligned} & \left. \frac{d}{ds} \right|_{s=0} \mathcal{X}(x; \gamma + s\eta) \\ &= \int_{\Omega \setminus D} (\mathcal{X}(x; \gamma(\tau)) - C) [\delta \circ \phi(x; \gamma(\tau)) \mathcal{P}_{\gamma'(\tau)} \nabla_{x_0} \phi(x; \gamma(\tau)) \\ & \quad - H \circ \phi(x; \gamma(\tau)) \kappa(\tau) n(\tau)] dx - \lambda \kappa(\tau) n(\tau), \end{aligned}$$

where  $\kappa(\tau)n(\tau)$  refers to the curvature vector and  $n(\tau)$  the normal to the curve. In two dimensions, this takes the form

$$\begin{aligned} & \left. \frac{d}{ds} \right|_{s=0} \mathcal{X}(x; \gamma + s\eta) \\ &= \int_0^1 [\delta \circ \phi(x; \gamma(\tau)) (\nabla_{x_0} \phi(x; \gamma(\tau)) \cdot n(\tau)) n(\tau) \\ & \quad + H \circ \phi(x; \gamma(\tau)) \kappa(\tau) n(\tau)] \cdot \eta(\tau) |\gamma'(\tau)| d\tau. \end{aligned}$$



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

$$\begin{aligned} \int_{\Omega \setminus D} (\mathcal{X}(x; \gamma(\tau)) - C) [\delta \circ \phi(x; \gamma(\tau)) \mathcal{P}_{\gamma'(\tau)} \nabla_{x_0} \phi(x; \gamma(\tau)) \\ - H \circ \phi(x; \gamma(\tau)) \kappa(\tau) n(\tau)] dx - \lambda \kappa(\tau) n(\tau) = 0, \end{aligned}$$

for all  $\tau$ . □

We may now perform gradient descent on the Euler-Lagrange equation. This introduces the time variable  $t$  into  $\gamma$  to form  $\gamma(\tau, t)$ , and flows the path by

$$\partial_t \gamma(\tau, t) = \left( - \int_{\Omega \setminus D} (\mathcal{X}(x; \gamma(\tau)) - C) \delta \circ \phi(x; \gamma(\tau)) \mathcal{P}_{\gamma'(\tau)} \nabla_{x_0} \phi(x; \gamma(\tau)) dx \right) + \tag{4.2}$$

$$\left( \int_{\Omega \setminus D} (\mathcal{X}(x; \gamma(\tau)) - C) H \circ \phi(x; \gamma(\tau)) dx + \lambda \right) \kappa(\tau) n(\tau) \tag{4.3}$$

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

$$\begin{aligned} \partial_t \gamma(\tau, t) &= \left( - \int_{\Omega \setminus D} (\mathcal{X}(x; \gamma(\tau)) - C) \delta \circ \phi(x; \gamma(\tau)) (\nabla_{x_0} \phi(x; \gamma(\tau)) \cdot n(\tau)) dx \right) n(\tau) + \\ &\quad \left( \int_{\Omega \setminus D} (\mathcal{X}(x; \gamma(\tau)) - C) H \circ \phi(x; \gamma(\tau)) dx + \lambda \right) \kappa(\tau) n(\tau). \end{aligned}$$

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 for  $C$ , it is easy to see that the optimal  $C$  should be chosen to be the average exposure,

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

We now discuss the parabolicity of this gradient flow. Note the last term  $\lambda \kappa(\tau) n(\tau)$  of (4.3) acts as regularization by flowing the path by  $\lambda$  times curvature. In fact, collecting  $\kappa(\tau) n(\tau)$  terms, we get

$$\int_{\Omega \setminus D} (\mathcal{X}(x; \gamma) - C) H(\phi(x; \gamma)) dx + \lambda. \tag{4.4}$$

This helps us determine what values we can choose for  $\lambda$ . 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 for parabolicity would be

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

**4.1.2. Numerical considerations.** As in Section 3.3, 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 4.3.** *Given  $\gamma: [0, 1] \mapsto \mathbb{R}^d$  a  $C^2$  simple curve, let  $\psi$  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  $\phi$  as defined in Formula (2.2). 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*

$$0 \leq \phi(x; \gamma) - \phi(x; \{g_j\}) \leq \Delta\tau K \|\gamma'\|_\infty. \tag{4.5}$$

*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  $\phi(x; \gamma) = \max_{\tau \in [0, 1]} \phi(x; \gamma(\tau)) = \phi(x; \gamma(\tau^*))$ . Then applying Lemma 2.1, 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

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

□

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 (4.3). 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 [12] with a single observer  $x_0 = \gamma(\tau_j, t_k)$ .

In Figure 4.1, we have two circular obstacles in the first and third quadrants. A straight line joining  $(-1, 0)$  and  $(1, 0)$ , forming  $\gamma_0$ , is then deformed using the flow (4.3) with  $C = 5$ . The flow eventually reaches a steady state and the resulting path depicted in Figure 4.1 matches with our intuition of an 'S' shaped curve. In addition, we plot  $\mathcal{X}(x; \gamma_0) / \|\mathcal{X}(\cdot; \gamma)\|_\infty$  and  $\mathcal{X}(x; \gamma) / \|\mathcal{X}(\cdot; \gamma)\|_\infty$ . Note, in order to improve the efficiency of the algorithm, instead of just choosing a constant value for  $C$ ,  $C = 5$ , it may be possible to increase  $C$  gradually along the flow by  $C = \max(2 + r_0 L(\gamma), 5)$ , where  $L(\gamma)$  is the length of the path  $\gamma$ .

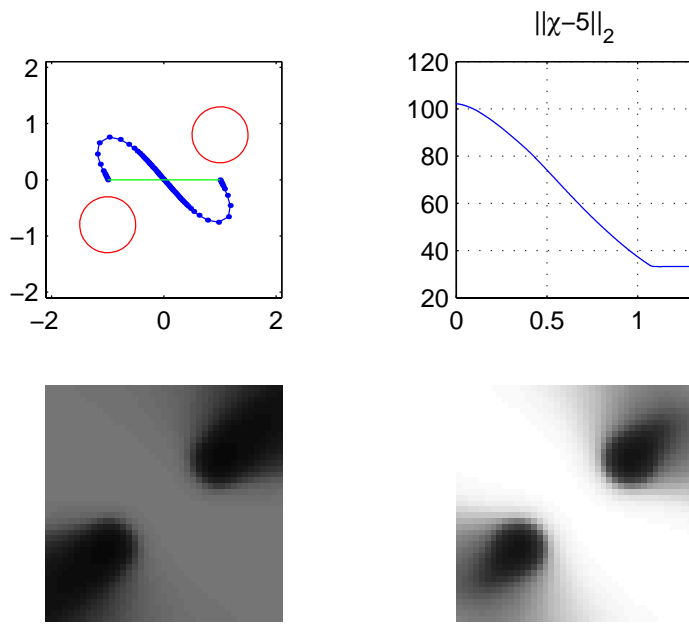


FIG. 4.1. The upper left figure shows the obstacles (red circles), the initial curve (green circle), and the optimized curve (blue dotted curve). The constant  $C$  is chosen to be 5, the curvature regularization term is 5. The images in the second row show the exposure of the initial and the optimized paths.

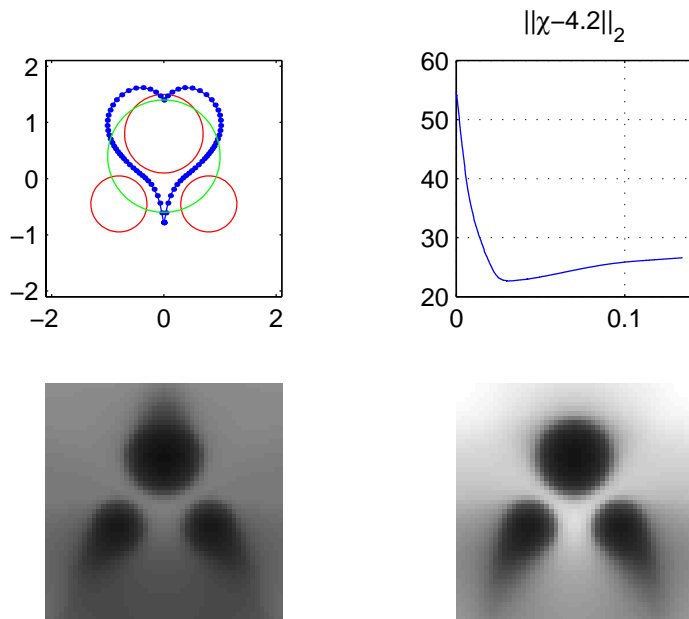


FIG. 4.2. The upper left figure shows the obstacles (red circles), the initial curve (green circle), and the optimized curve (blue dotted curve). The constant  $C$  is chosen to be 4.2, the curvature regularization term is 0.05. The images in the second row show the exposure of the initial and the optimized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

Figure 4.2 shows an example in which we have as the initial path  $\gamma_0$  a closed curve passing through the obstacles. Note the portions of  $\gamma$  away from its fixed end points at  $(0,0.9)$  eventually pull out of the obstacles. The tip at the bottom of the path first reaches further down during the descent and is then pulled back due to the regularization term. One can see a corresponding effect in the change in  $\|\mathcal{X}(\cdot;\gamma) - 4.2\|_2$ , where regularization affects the minimization of this quantity.

**4.1.3. Weights and memory.** We can extend our approach for uniform exposure to include weights that measure the importance of 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: defining

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

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  $\delta(t - \tau)$ , 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  $\eta$ , 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  $\eta$  duration, thus mimicking forgetfulness. Correspondingly, the optimization problem can be formulated as in Problem 4.1, with  $E(\gamma, C)$  replaced by

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

or even

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

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

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

to further maximize, with weight  $\alpha$ , the total exposure.

4. Include a multiplicative penalty term to increase the length of the path:

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

In terms of numerical representation, we would 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 [3, 11] or the segment projection method [4].

**4.2. 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, especially the shape of the tail, needs to be carefully chosen to correctly influence the visible volume.

Let  $x_0$  be the location of an observer and let  $y$  be the location of the point object that we want to make visible. Suppose the non-trivial case of  $y$  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  $\phi(\cdot; y)$ , 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  $\{\phi(\cdot; y) = 0\}$ . This can be written as searching for the minimizer of the problem

$$\min_{\{\gamma: [0,1] \rightarrow \mathbb{R}^d - D, \gamma(0) = x_0, \phi(\gamma(1); y) = 0\}} \int_0^1 |\gamma'(\alpha)| d\alpha.$$

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  $\infty$  inside obstacles. This distance function can be easily constructed using a variety of techniques, including PDE, fast marching, and fast sweeping methods [8, 9, 10, 13, 14] that solve

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

with boundary conditions  $T_y = 0$  where  $\phi(\cdot; y) = 0$  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  $\nabla T_y$  drawn from  $x_0$  to the zero level set of  $\phi(\cdot; y)$ . An ODE solver

combined with finite differencing and interpolation in a method-of-lines approach can be used to solve for this flow,

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

starting from the initial observer position. The  $\max\{-T_y(x_0), 0\}$  term ensures that  $x_0$  will slow down and stop at the zero level set and, further, that  $x_0$  will not move if  $y$  is already visible. Note, the time it takes to travel along this shortest path is given by  $\min_{x \in \{\phi(\cdot; y) = 0\}} T_y(x)$ .

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 forcing slow movement. 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  $\nabla T_y$ .

Figure 4.3 shows a step-by-step slightly simplified example and results of our approach to the shortest path problem.

**4.3. Tracking an object.** One can easily find a situation in which the observer wishes to keep a moving object in sight for as long as possible. We interpret this objective as keeping the target away from occlusion as much as possible. Certainly, a modification of results of the previous section can be helpful here. However, this is a subject of future work. Instead, we take a different approach. 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  $\phi$ , 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  $\psi$  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:

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

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

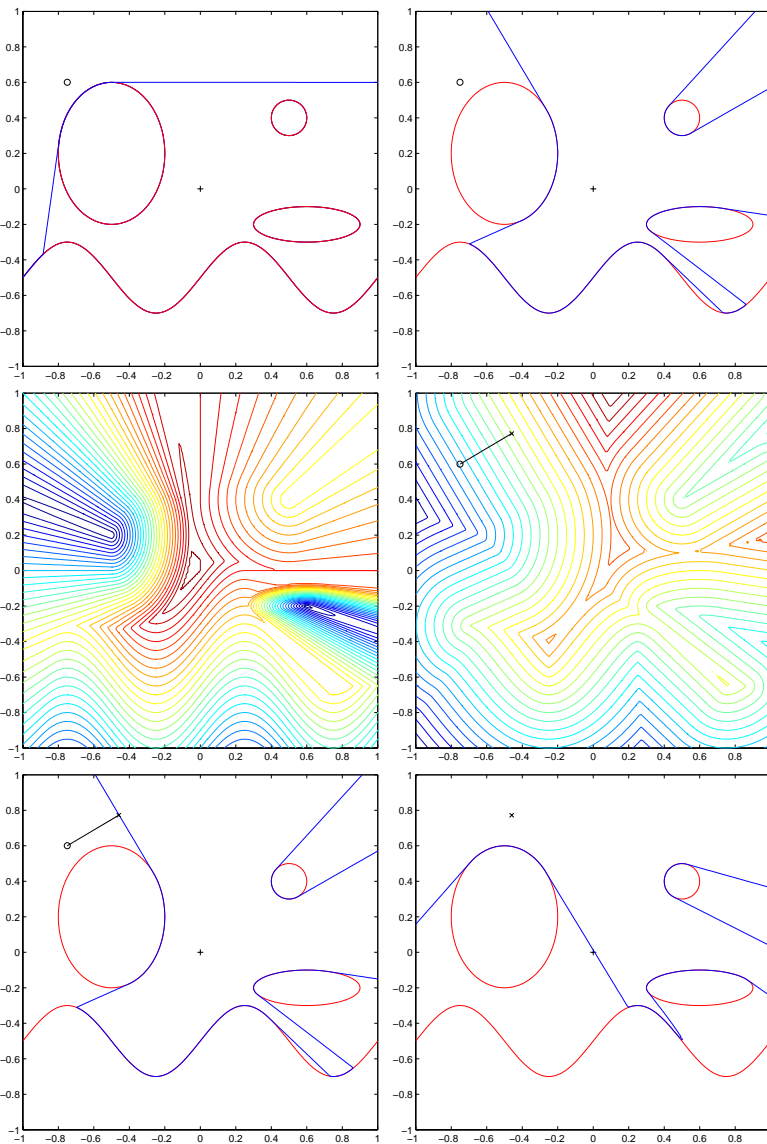


FIG. 4.3. The figures in the first row show the observer, obstacles, shadow boundary, and a point at the origin that is invisible, and then the first step of our procedure, namely treating the origin as the observer. The figures of the second row show the contours of the visibility level set function and its redistancing into signed traveltime, with obstacle effects absent for simplicity, 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.

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  $\lambda$  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  $U(d_D(x_0, y))$ ,

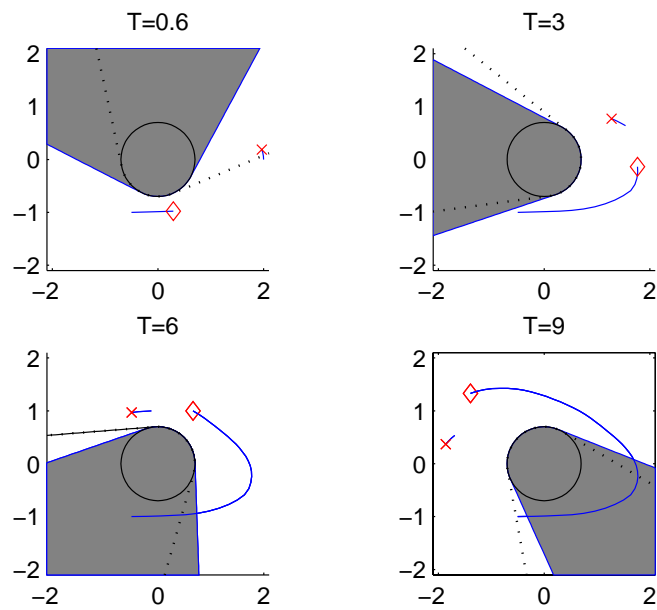


FIG. 4.4. Past trajectories of observer and evader are shown. The diamonds and crosses indicate the current locations of the observer and the evader, respectively. Thus, the observer circles around the obstacle chasing the evader.

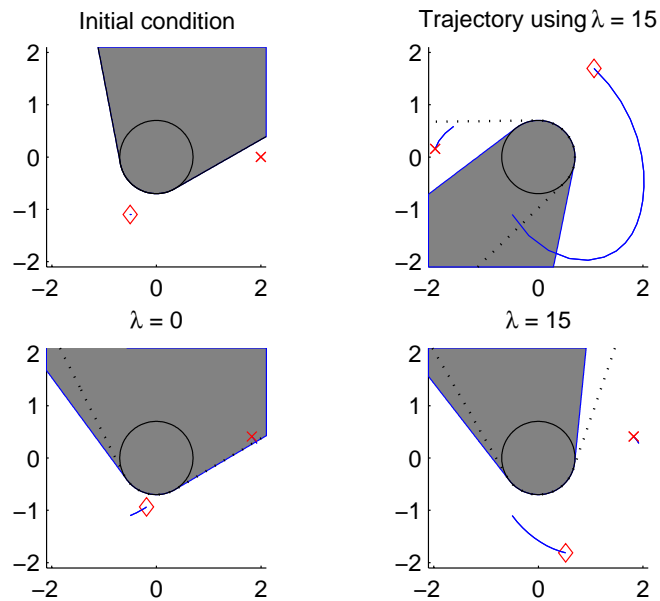


FIG. 4.5. Past trajectories due to the absence and presence of the visibility gradient term are shown. of observer and evader. The diamonds and crosses indicate the current locations of the observer and the evader, respectively. The lower left plot is with the absence of the gradient and should be compared 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.



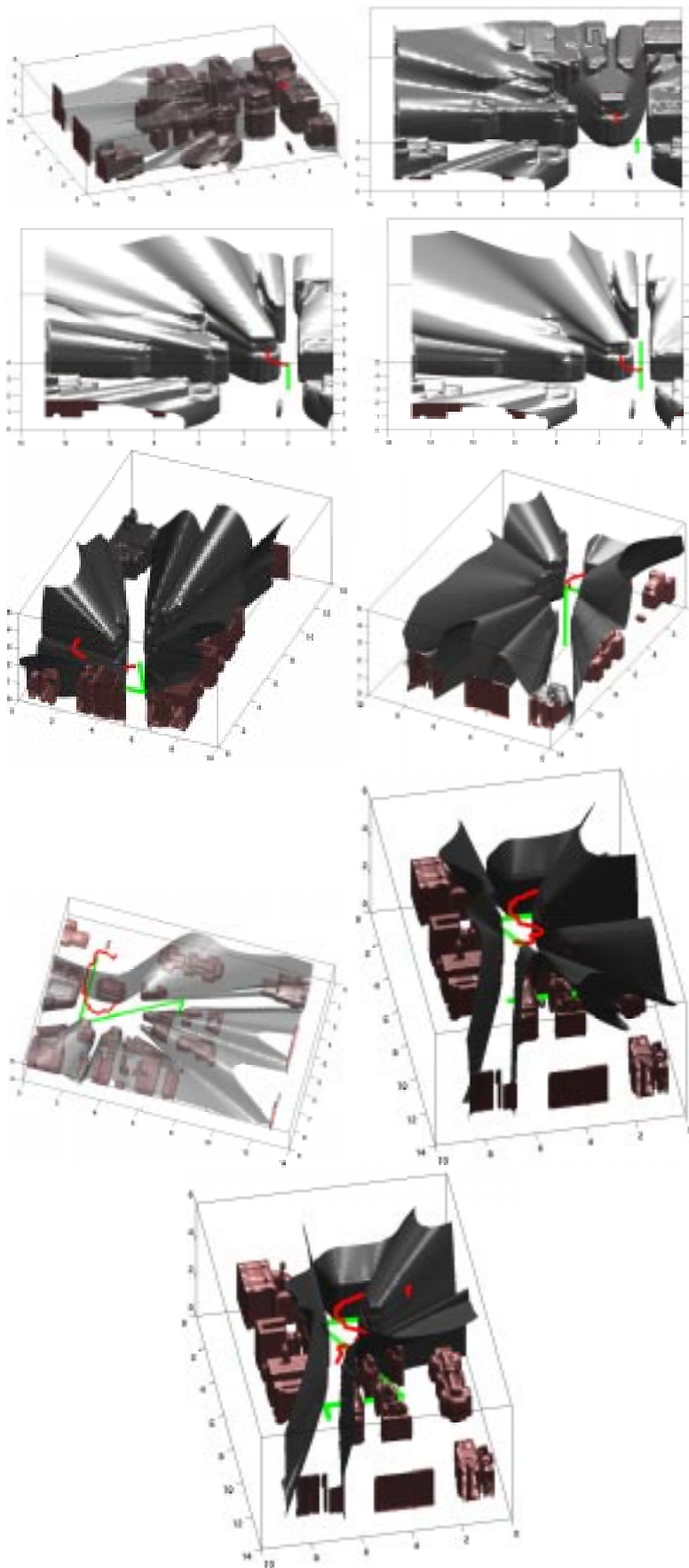


FIG. 4.6. Three dimensional example among buildings of a digital city. The observer is still able, following our algorithm, to keep the evader in sight in this environment.

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 [8, 9, 10, 13, 14].

We present simulations in the selected snapshots of Figures 4.4, 4.5, and 4.6. In Figure 4.4, the dotted lines show the shadow boundary 0.3 time units in the past. The shaded region is the current invisible region for 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 4.5 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. Finally, Figure 4.6 shows a three dimensional example using the buildings of a digital city as obstacles. The observer can handle the complexities of the dimension and the obstacles to keep the evader in sight.

## 5. 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.

**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 and DMS-0513394.

## 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, February, 10, 1, 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] 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.

- [5] B. Engquist, A. K. Tornberg and Y. H. Tsai, *Discretization of dirac delta functions in level set methods*, Journal of Computational Physics, 207, 1, 28-51, 2005.
- [6] 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*, Springer, 95-101, 1998.
- [7] 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, 71-494, 1999.
- [8] J. Helmsen, E. Puckett, P. Colella and M. Dorr, *Two new methods for simulating photolithography development in 3D*, Proc. SPIE, 2726, 253-261, 1996.
- [9] 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.
- [10] J. A. Sethian, *Fast marching level set methods for three dimensional photolithography development*, Proc. SPIE, 2726, 261-272, 1996.
- [11] C. W. Shu and S. Osher, *Efficient implementation of essentially nonoscillatory shock-capturing schemes*, J. Comput. Phys., 77, 2, 439-471, 1988.
- [12] Y. H. Tsai, L. T. Cheng, P. Burchard, S. Osher and G. Sapiro, *Visibility and its dynamics in a PDE based implicit framework*, J. Comput. Phys., 199, 206-290, 2004.
- [13] 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, 673-694, 2003.
- [14] J. N. Tsitsiklis, *Efficient algorithms for globally optimal trajectories*, IEEE Transactions on Automatic Control, 50, 1528-1538, 1995.