Visibility Optimizations

Richard Tsai

University of Texas at Austin

July 22, 2005

Joint work with Li-Tien Cheng

Museum v.s. Tourists

Museum:

- Place exhibition items under surveillance cameras
- Place as few cameras as possible

Tourist:

- Shortest path to see Mona-Lisa (La Jaconde)
- Shortest path through the museum
- Not missing any exhibition item along the way

Complication:

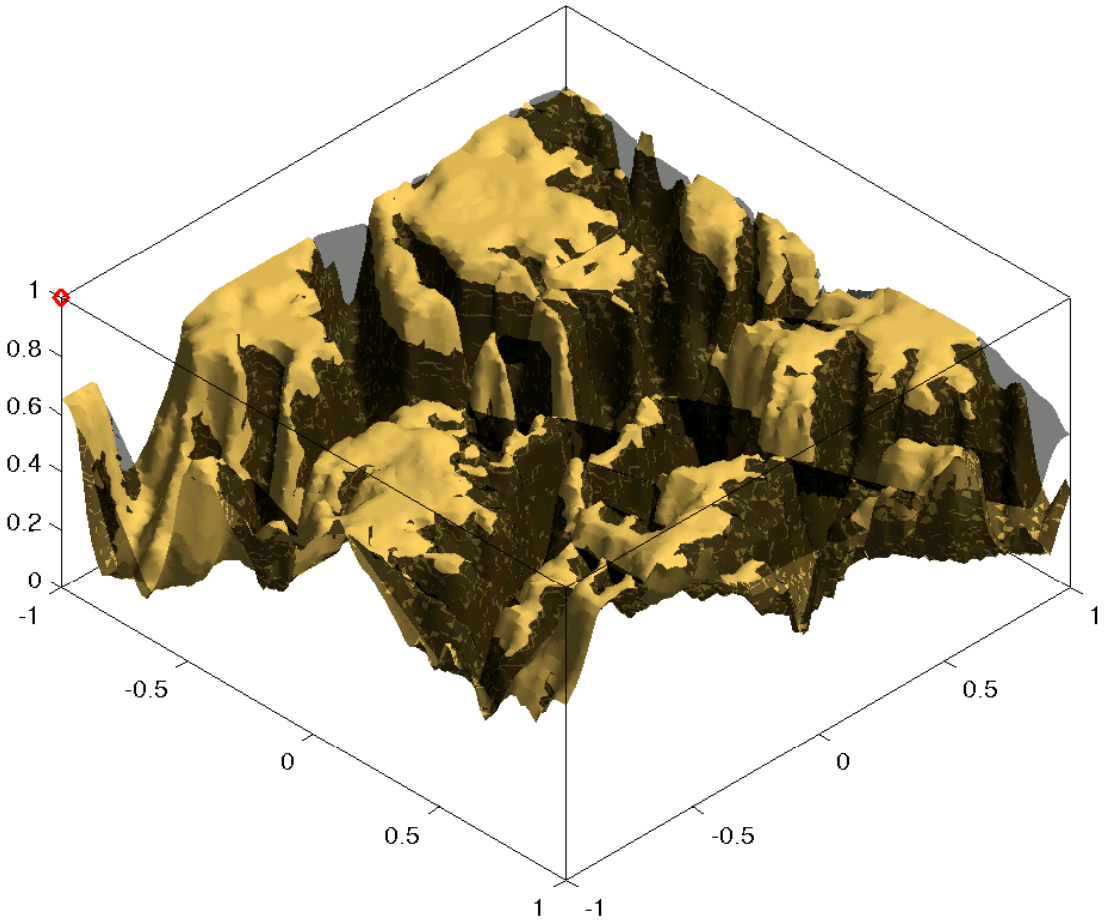
- The tourist does not have a map.
- The “tourist” wants to go from A to B, hiding from the cameras as much as possible.
- Guard following the “tourist”.

Problem 1: Find visibility

Given implicit surfaces (occluders) $\Gamma = \partial\Omega$.

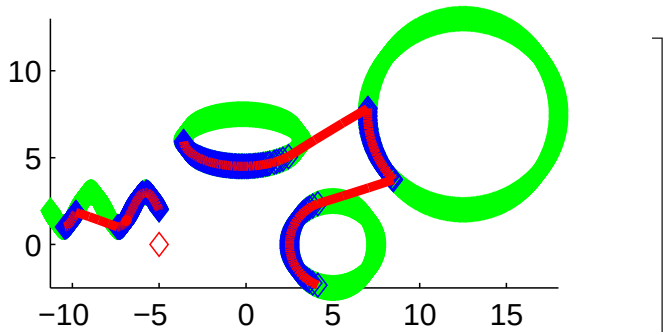
Light source (vantage point) at x_o.

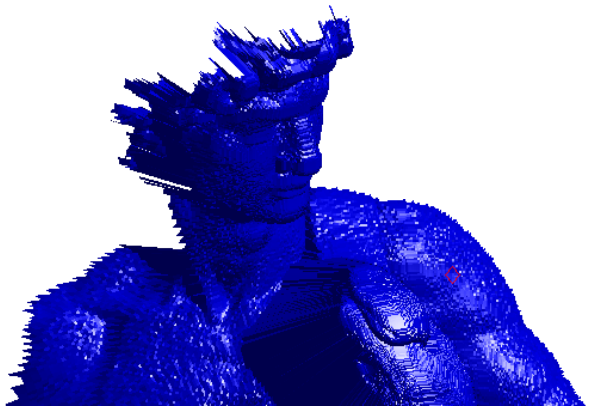
Construct: $u(y; x_0)$ s.t. $\{y : u(y; x_0) < 0\}$ = the invisible regions.



Construct visibility “on the fly”

Visibility from point clouds:





Problem 2: Optimizing for visibility

Known occluder configuration:

1. Place vantage point(s) maximizing visibility.
   (vantage points could form a path.)
2. Path planning under visibility constraints: $H(\nabla v) = r(x, u, \phi)$.
   (possibly with time dependence.)

Unknown occluder configuration:

1. Do the above using surfaces reconstructed from some remote sensing mechanism.
2. Place/Shape occluders optimizing certain visibility objective.

Problem 2: Optimizing for visibility

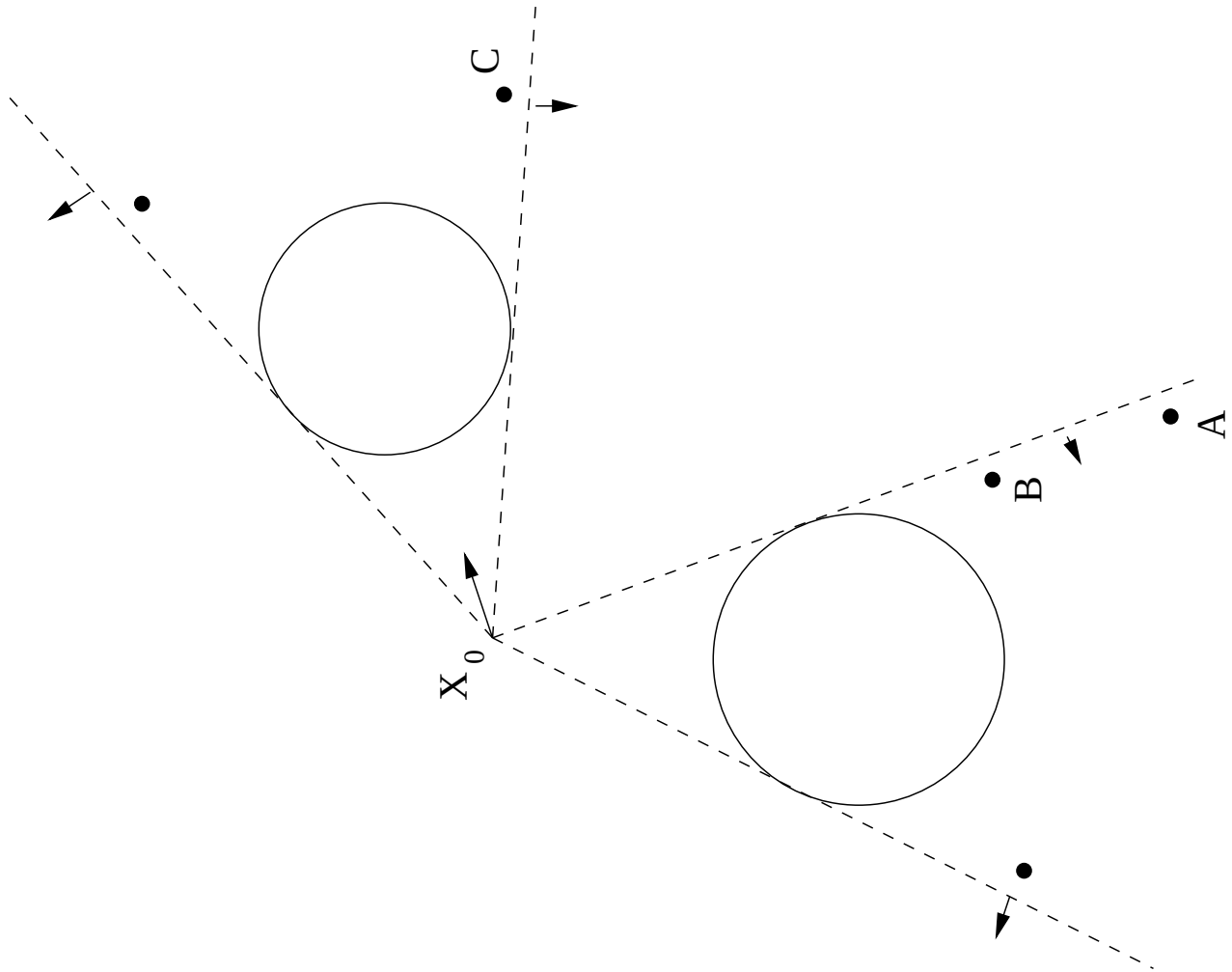
Known occluder configuration:

1. Place vantage point(s) maximizing visibility.
   (vantage points could form a path.)
2. Path planning under visibility constraints: $H(\nabla v) = r(x, u, \phi)$.
   (possibly with time dependence.)

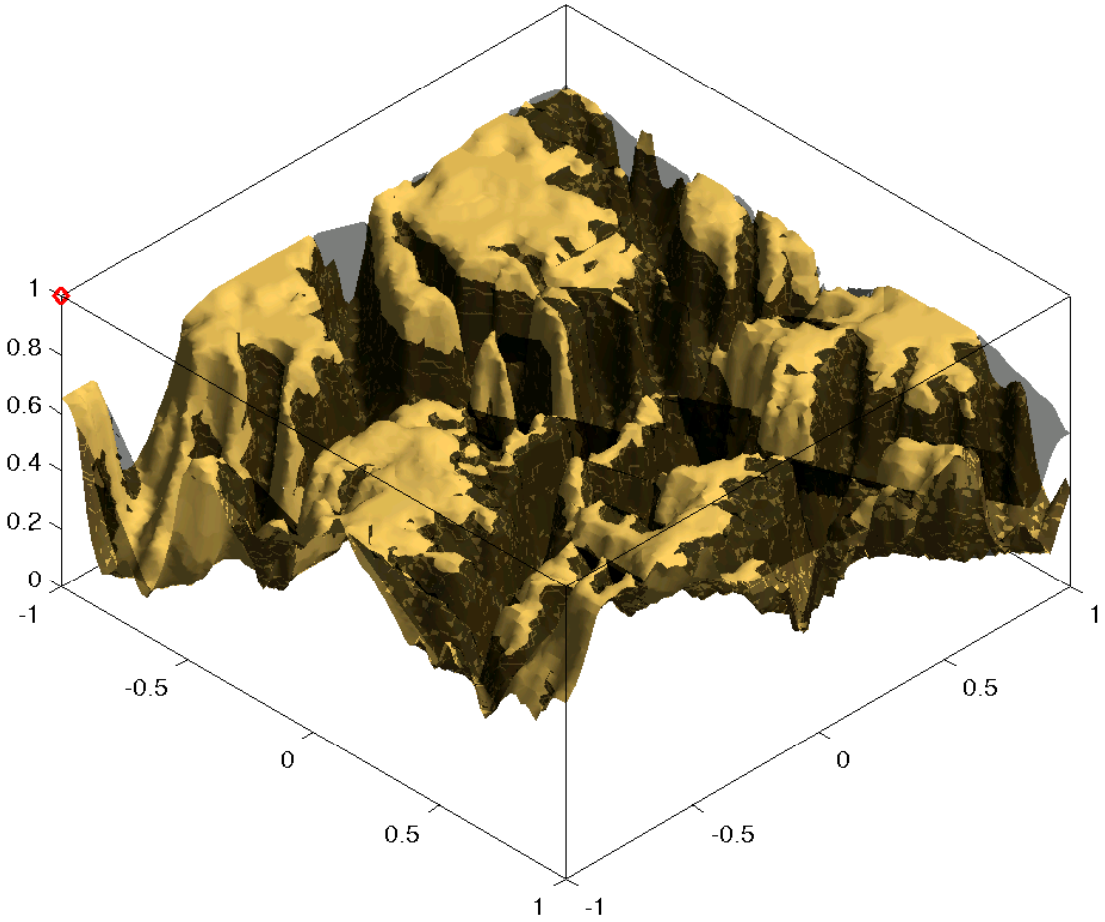
Unknown occluder configuration:

1. Do the above using surfaces reconstructed from some remote sensing mechanism.
2. Place/Shape occluders optimizing certain visibility objective.

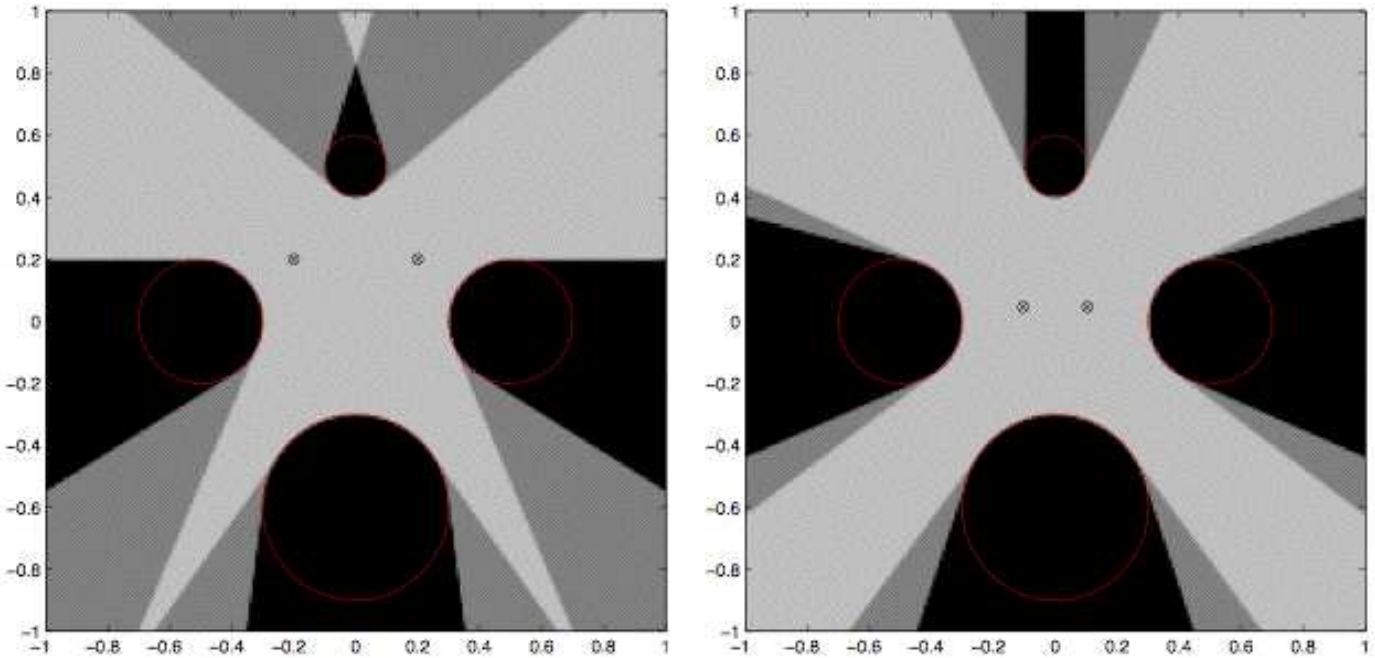
Maximize distance to occlusion



Maximize visible volume



Maximize “averaged exposure”



Related work

- Numerous work in the computer graphics community for computing visibility in polygonal environments.
  E.g. The z-buffer, [Coorg et al.], [F. Durand] etc.
- From the robotics community: certain robotic path planning with visibility penalty in polygonal environments.
  E.g. [Lavalle et al.], [Latombe et al]
  Complexity scales with the number of edges
- Visibility of implicit surfaces: [Adalsteinson]

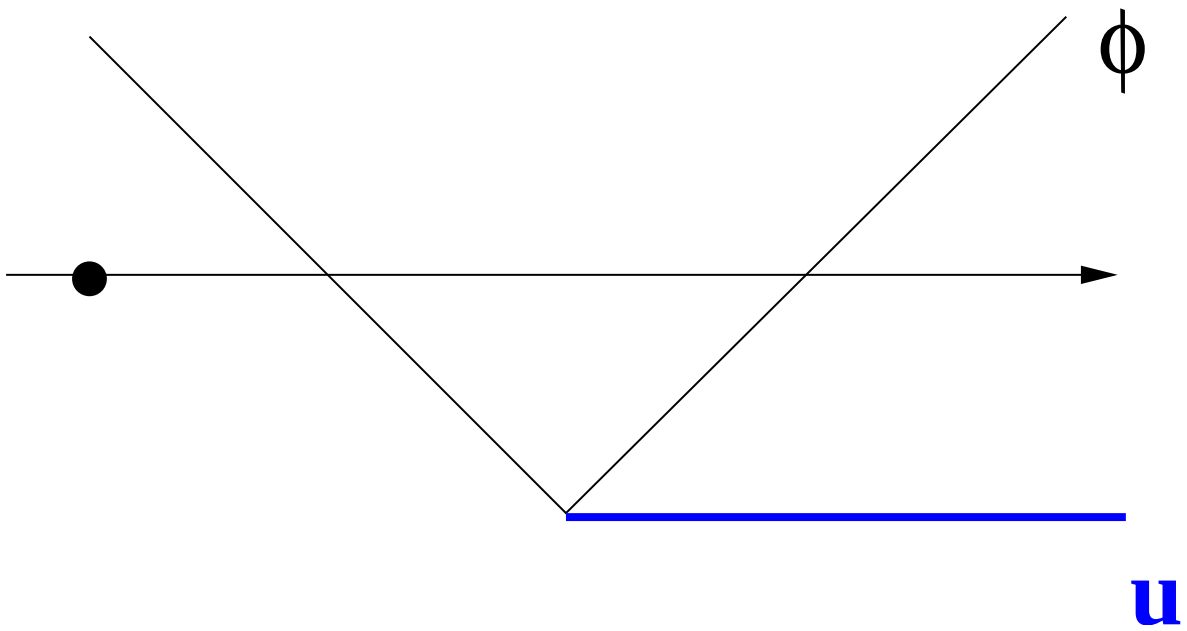
Our contribution:

- Continuous visibility representation and fast algorithm
- A variety of models of visibility optimization and the corresponding numerical algorithms.

Finding visibility

Basic algorithm

$$u(y, \mathbf{x}_o) := \min_{\xi \in L(x_0, y)} \phi(\xi).$$



Integral and differential Eqns

$$u(y, \mathbf{x}_o) := \min_{\xi \in L(x_0, y)} \phi(\xi).$$

$$\frac{\partial}{\partial r} u = H(u - \phi)(\frac{\partial}{\partial r} \phi(x_+))^-$$

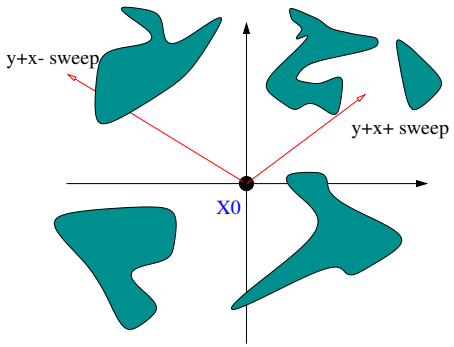
or

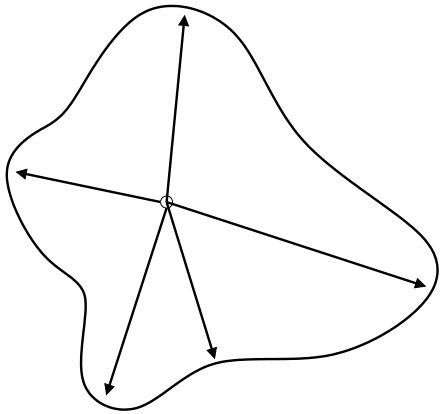
$$u(x) = \phi(x_0) + \int H(u - \phi)(\frac{\partial}{\partial r} \phi(x_+))^-.$$

[C.Y. Kao and Tsai, 2005]

Grid-based algorithm

One-pass upwinding algorithm through each grid point in a predefined order (independent of the occluders): e.g.



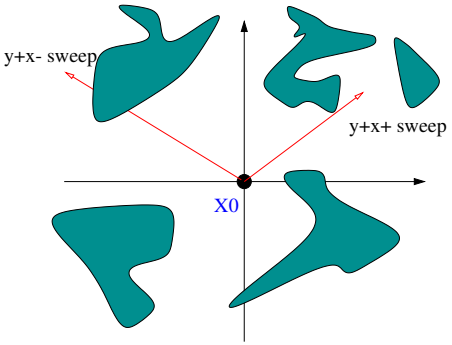


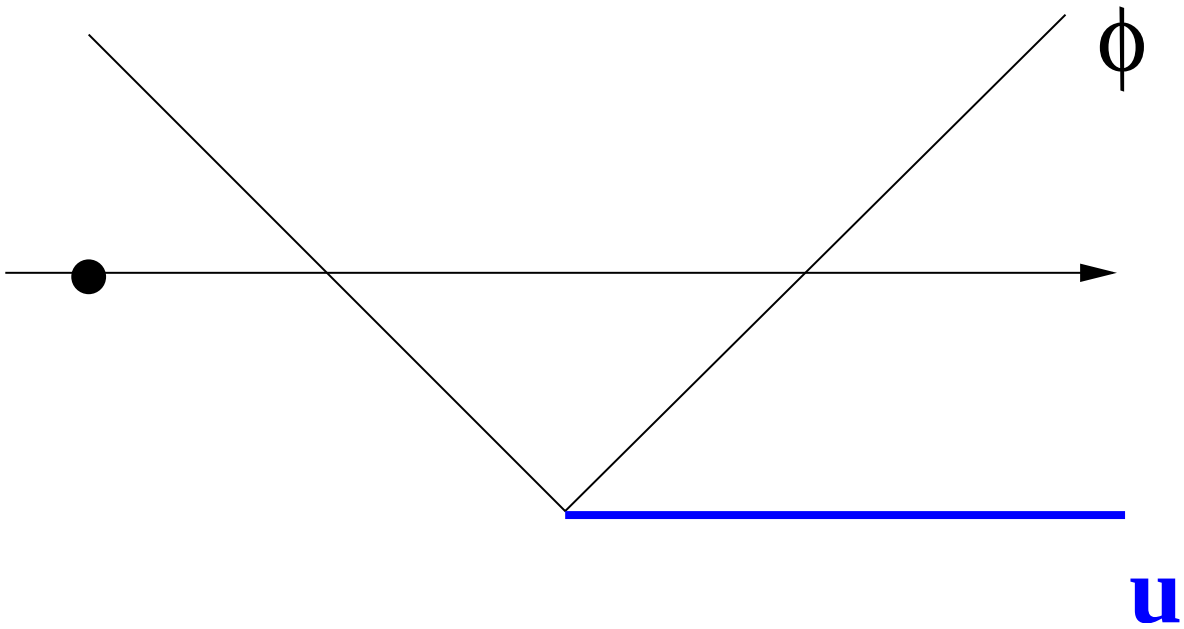
Operations performed at each grid point:

1) Explicit solution of a 3 by 3 system

2) Taking minimum of two values

Algorithm detail



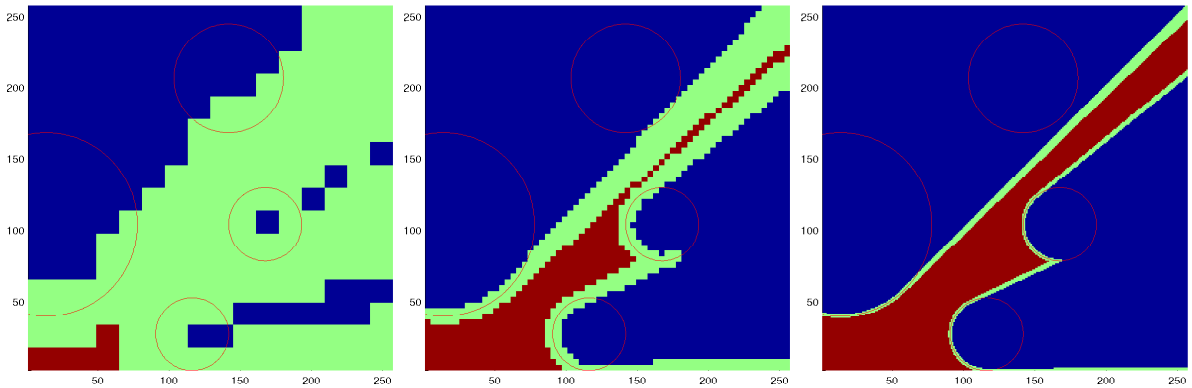


For each grid point x, $r(x) = (x - x_0)/|x - x_0|$:

1. Solve $\nabla u \cdot r(x) = 0$ at x;
2. Update: $u(x) = \min(\phi(x), u(x))$.

Variant: a optimal complexity version for DTED data.

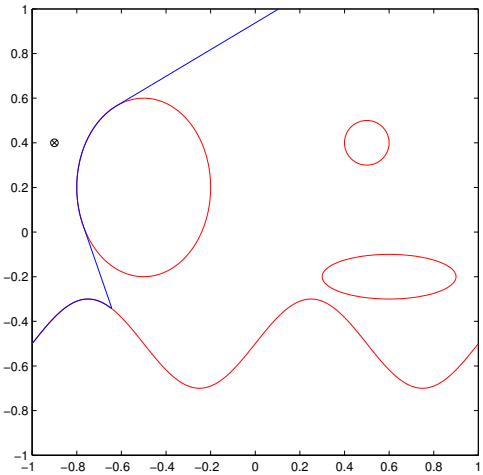
Multi-level algorithm

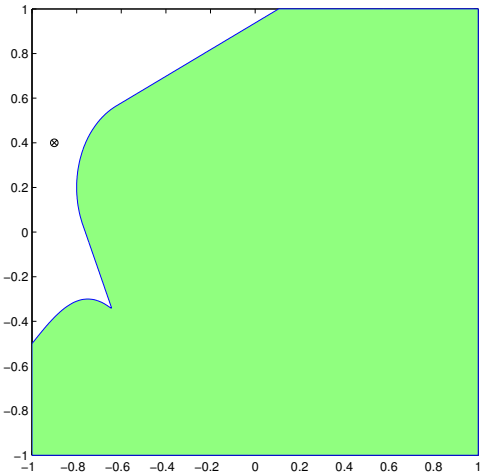


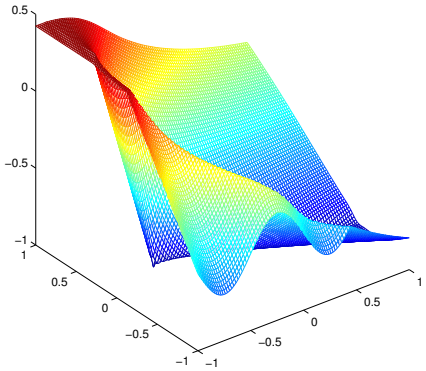
Skipping regions on which ψ does not change signs!

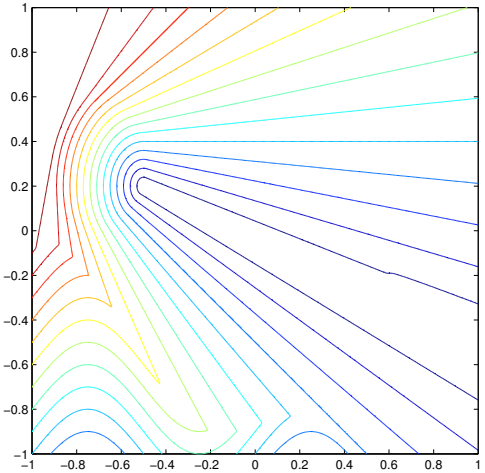
Based on Lipschitz estimates.

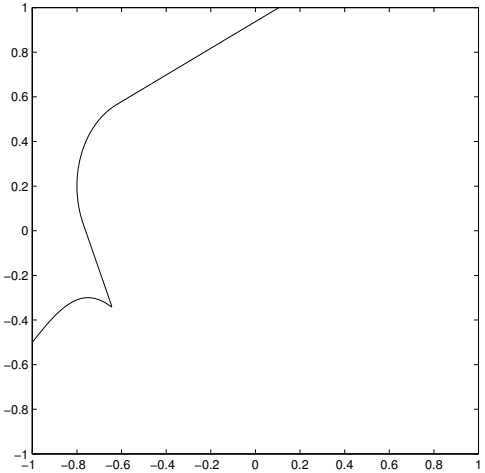
Visibility function





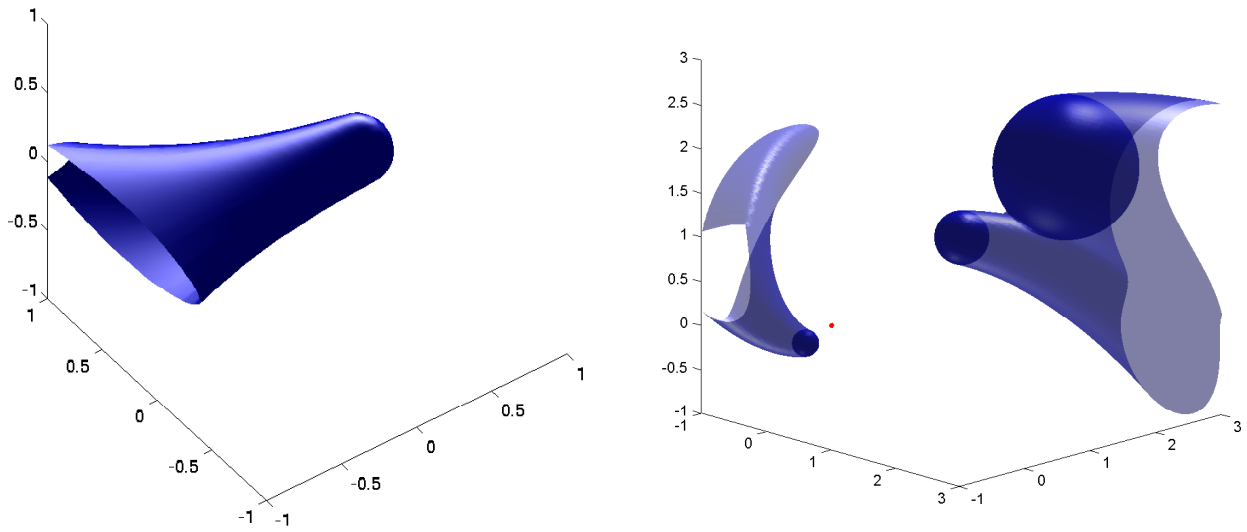






Visibility in more general ray fields

$$\nabla u \cdot r(x) = H(u - \phi) \min(\nabla \phi \cdot r(x), 0) + B.C.$$

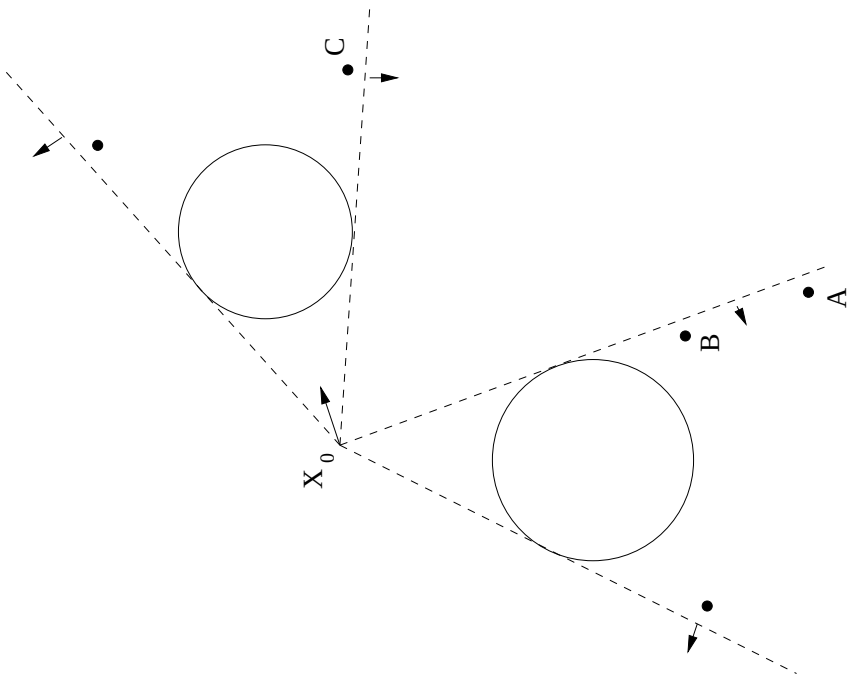


Viscosity solution theory and discretizations. [C. Kao and Tsai]

Efficient algorithm.

Advantages

- Extremely easy algorithm
- Sharp, implicit, representation of occlusion boundaries
- Reciprocity of Solution: $u(x;y) = u(y;x)$.
- Continuity of solution: $\nabla_x u(y;x)$ and $\nabla_y u(y;x)$ make sense. Notion of "how much occluded"?



Advantages

- Algorithm independent of the topology and geometry of the occluders
- Extensions: many vantage points, moving surfaces.

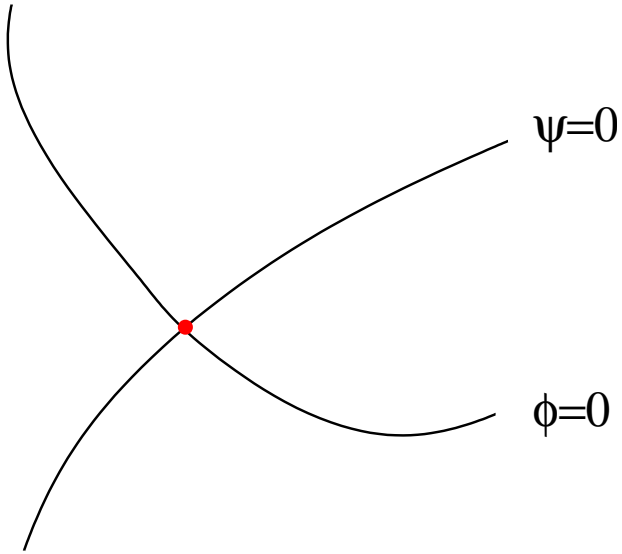
$$\max_{\alpha \in A} F(x, u, \nabla u, r_\alpha) = 0,$$

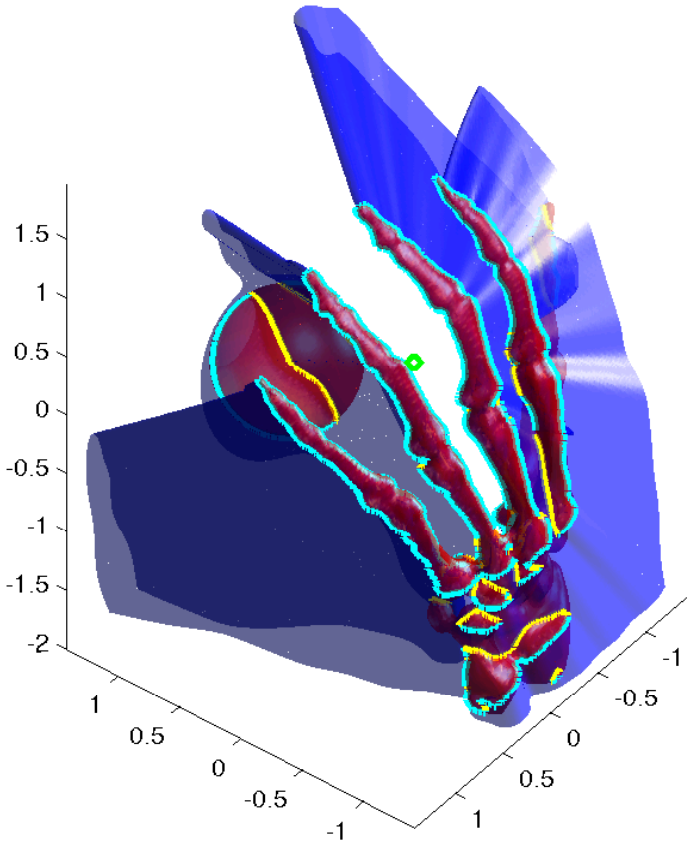
$$F(x, u, \nabla u, r_\alpha) = \nabla u \cdot r_\alpha - H(u - \phi) \min(\nabla \phi, 0).$$

- Approximation of visibility from a path:
  $0 \leq u(x;\gamma) - u(x;\{\gamma_j\}) \leq \Delta\tau K \, ||\gamma'||_\infty.$

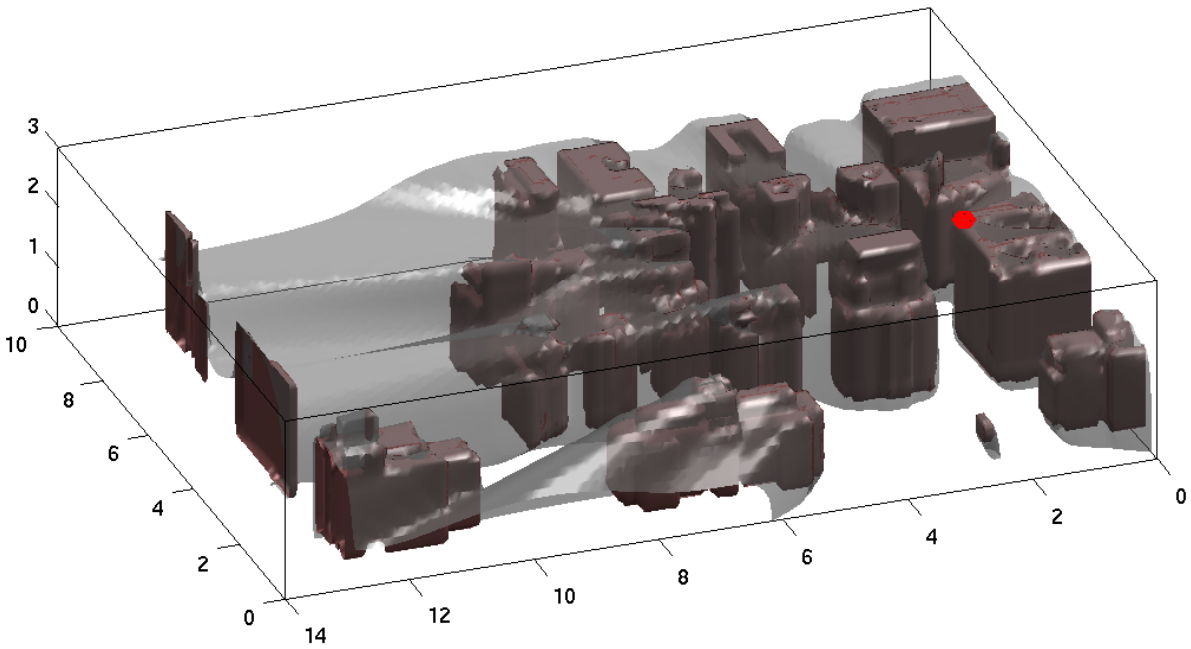
Characterizing horizons and terminators

We characterize the horizon and its terminator by the intersections of level set functions.





Visibility in DC.



Acknowledgement: DC grid data obtained from Flow Analysis, Inc.

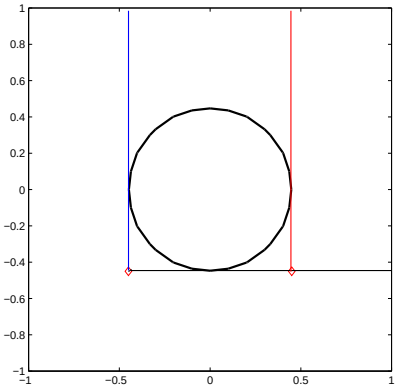
Joint visibility

Given two observers x_0 and x_1:

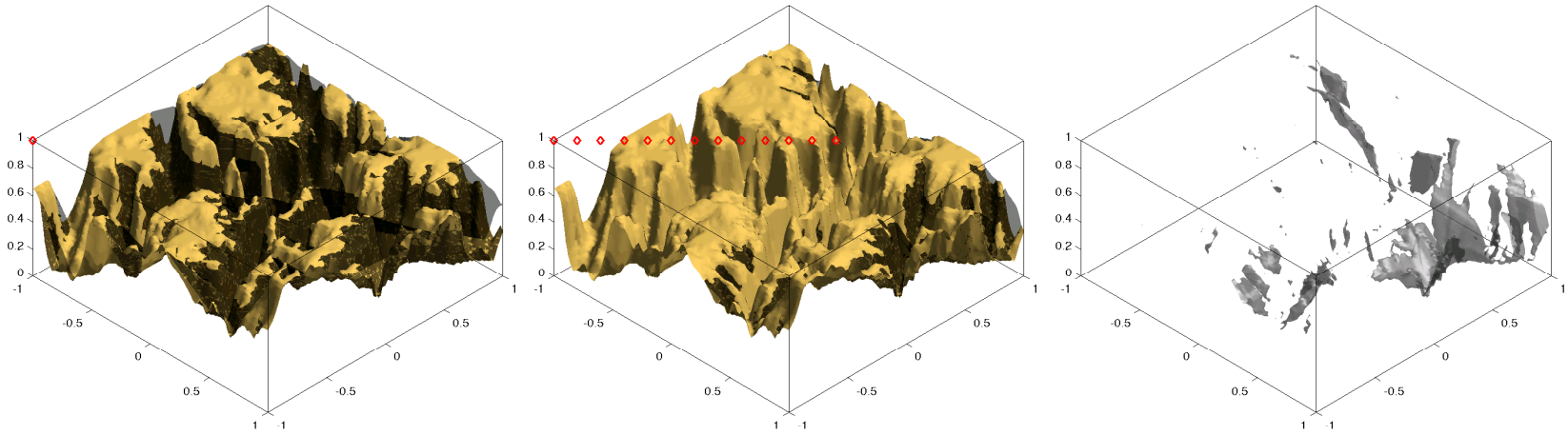
$$u(y;x_0,x_1) := \max(u(y;x_0),u(y;x_1)).$$

$\{u(y;x_0,x_1) < 0\}$ is the region occluded from both x_0 and x_1.

Other Boolean operations possible.



Accumulative result (memory)



Joint visbility of a set of points $\leftrightarrow$visibility of a path $\leftrightarrow$

visibility with memory of a trajectory

Maximizing visibility

Variational problems

$$\min_{x_0} F(u; x_0) \quad \text{or} \quad \min_{\phi} F(\phi; u_{x_0})$$

Known occluder configuration:

1. Place vantage point(s) maximizing visibility:
   - Shortest path to see a given location
   - The “I want to see everything” problem (complete visibility)
   - Uniform viewing habit (exposure)
2. Path planning under visibility constraints: $H(\nabla v) = r(x, \phi, u)$. (with time dependence.)
   - The “I don’t want to be seen” problem.

Unknown occluder configuration:

1. Do the above.
2. Place/Shape occluders maximizing certain visibility objective.

Variational problems

$$\min_{x_0} F(u; x_0) \quad \text{or} \quad \min_{\phi} F(\phi; u_{x_0})$$

Known occluder configuration:

1. Place vantage point(s) maximizing visibility:
   - Shortest path to see a given location
   - The “I want to see everything” problem (complete visibility)
   - Uniform viewing habit (exposure)
2. Path planning under visibility constraints: $H(\nabla v) = r(x, \phi, u)$. (with time dependence.)
   - The “I don’t want to be seen” problem.

Unknown occluder configuration:

1. Do the above.
2. Place/Shape occluders maximizing certain visibility objective.

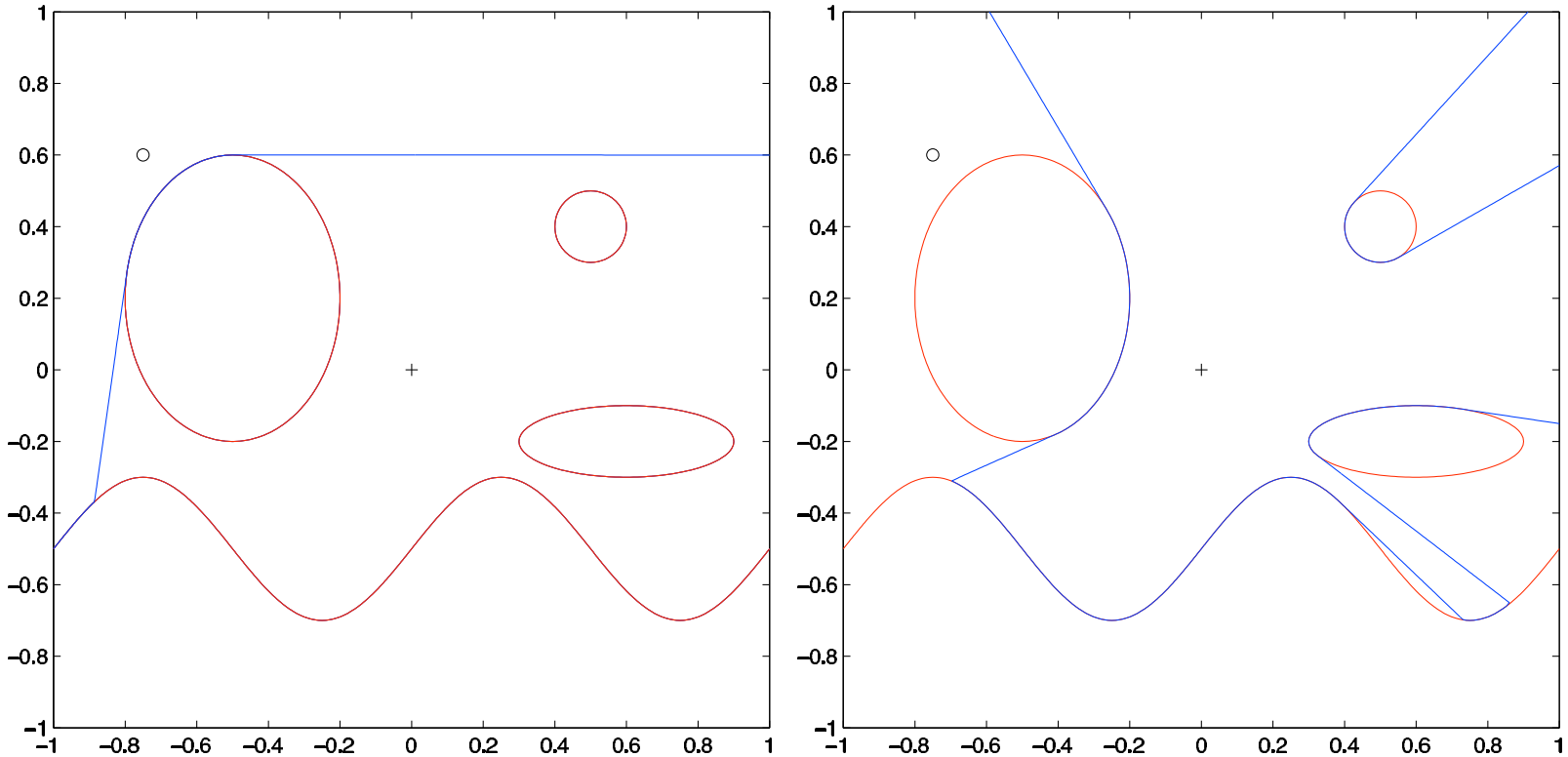
An example with unknown occluder

Shape reconstruction from gray scale images: $\min_{\phi} F(\phi; u_{x_0})$

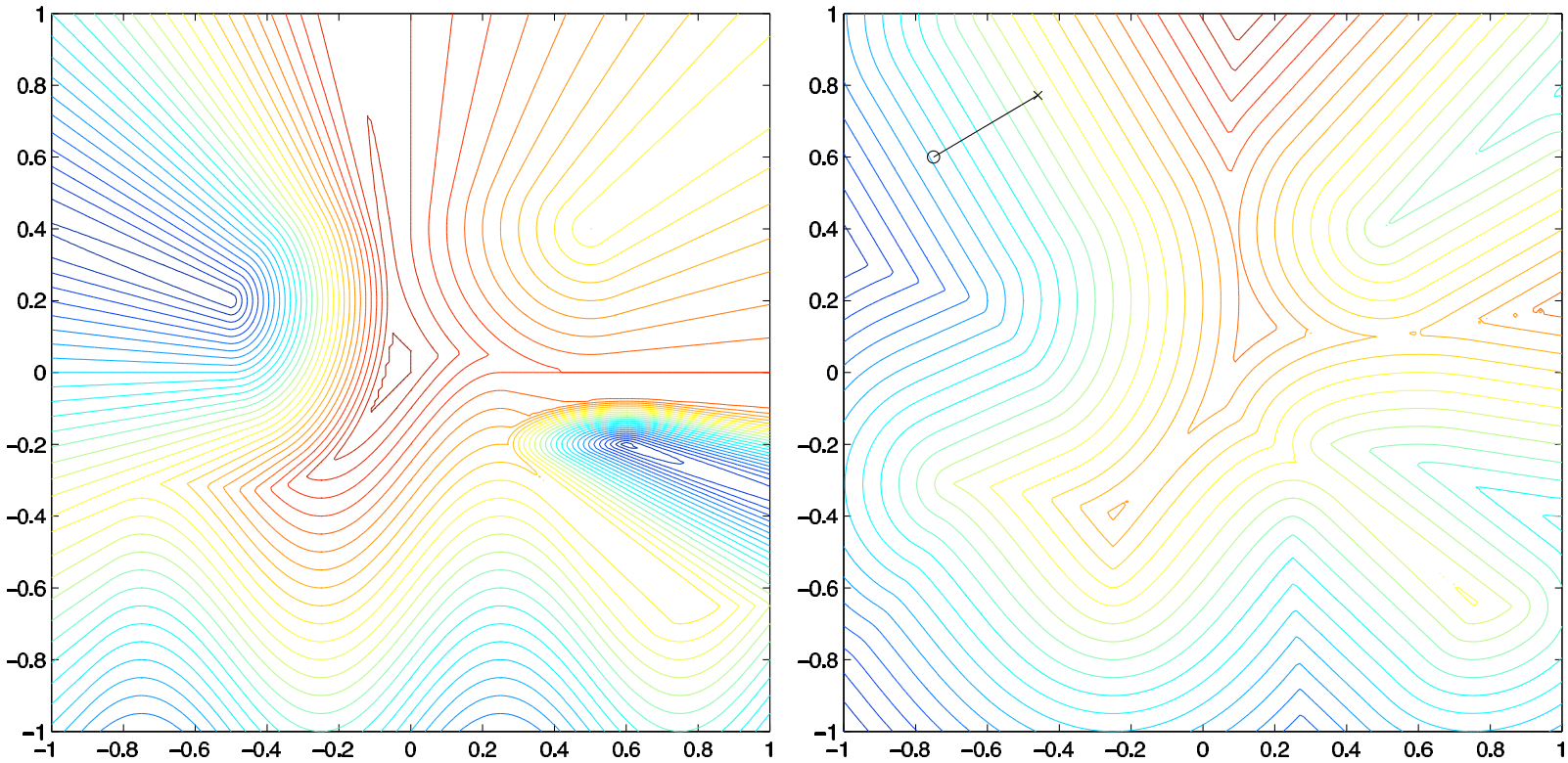
Joint work with Soatto et al.

Taking advantage of the reciprocity

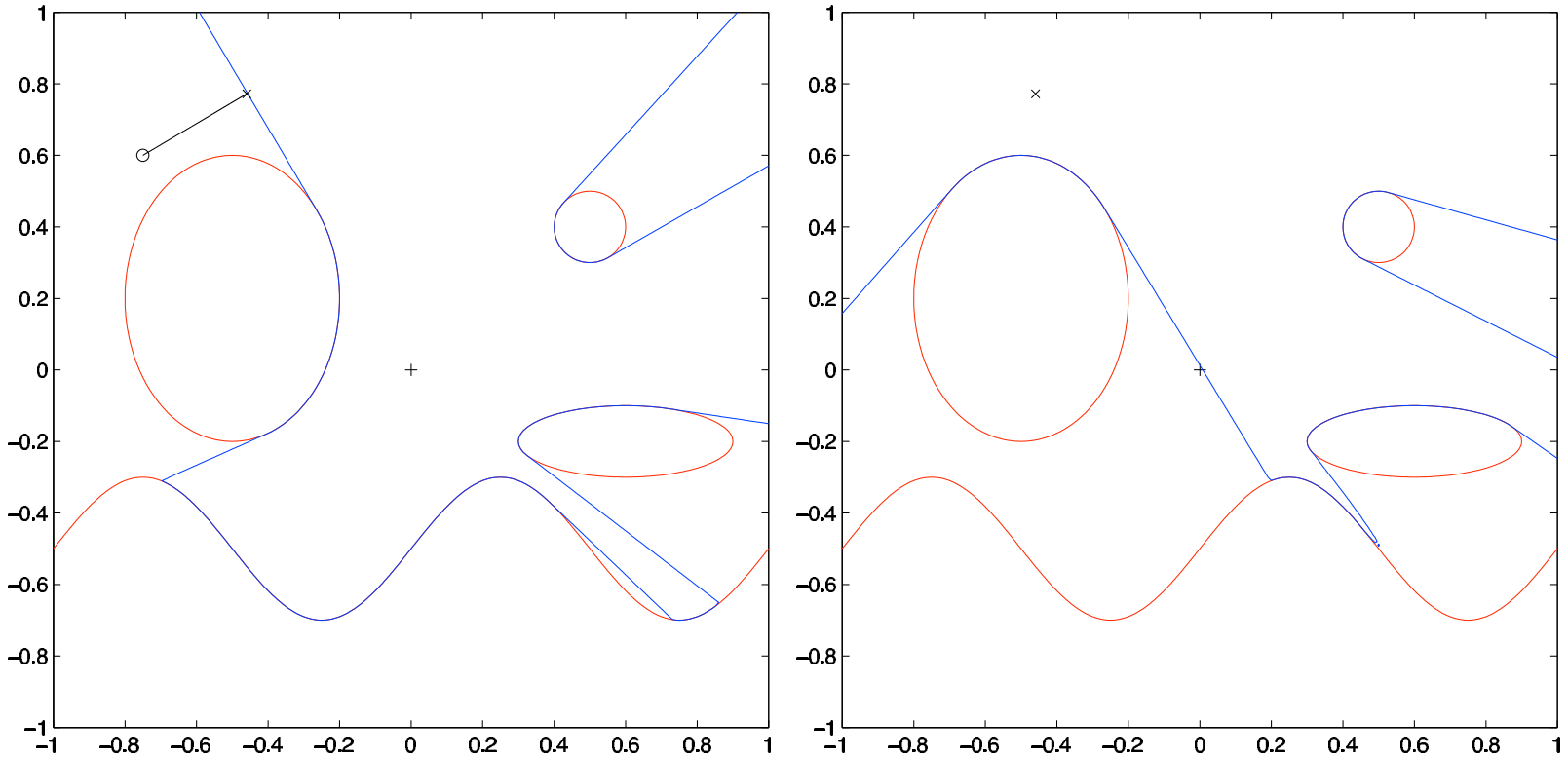
$$u(y, x_0) = u(x_0; y).$$



Taking advantage of the reciprocity

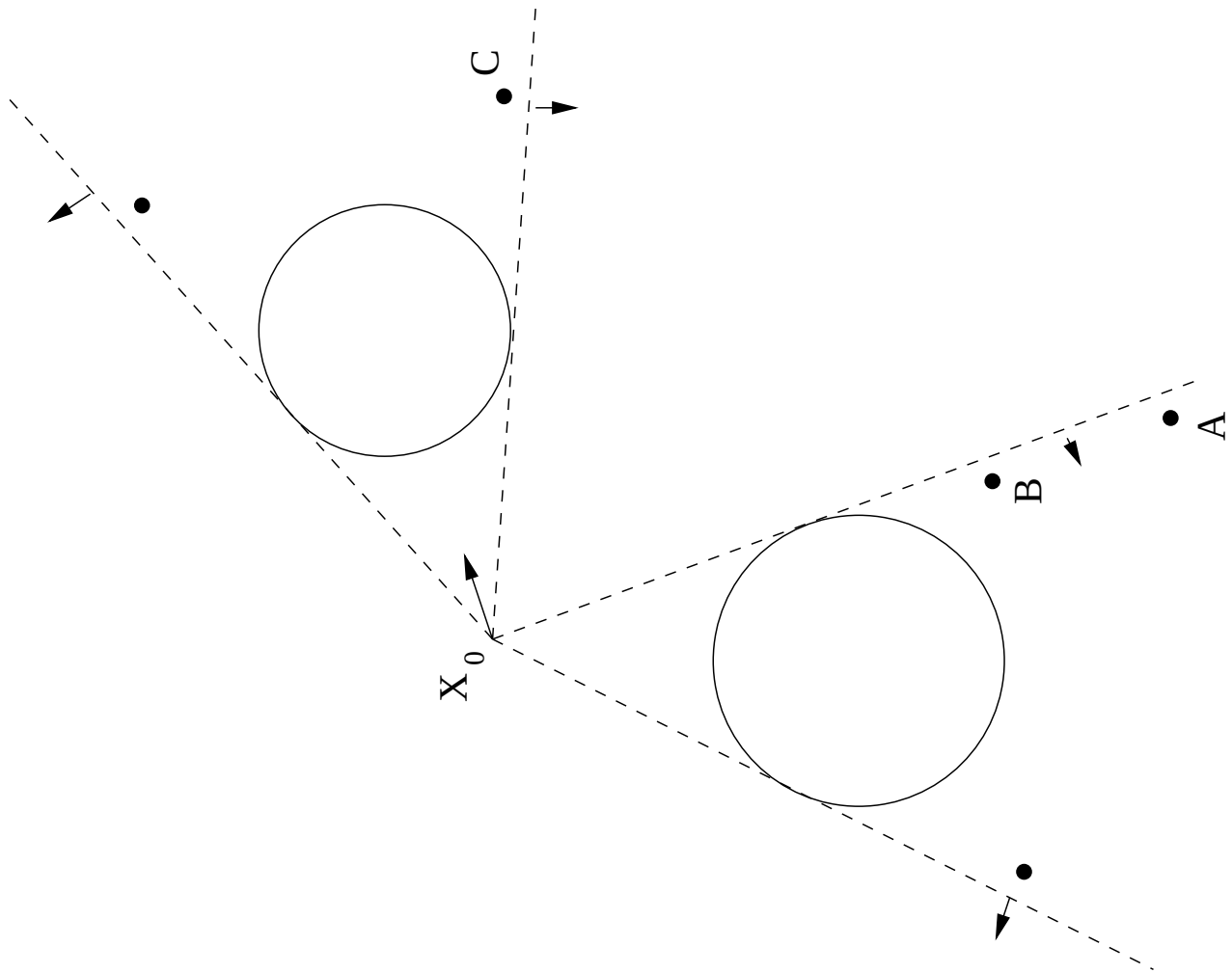


Taking advantage of the reciprocity



Same approach to non-trivial multiple targets.

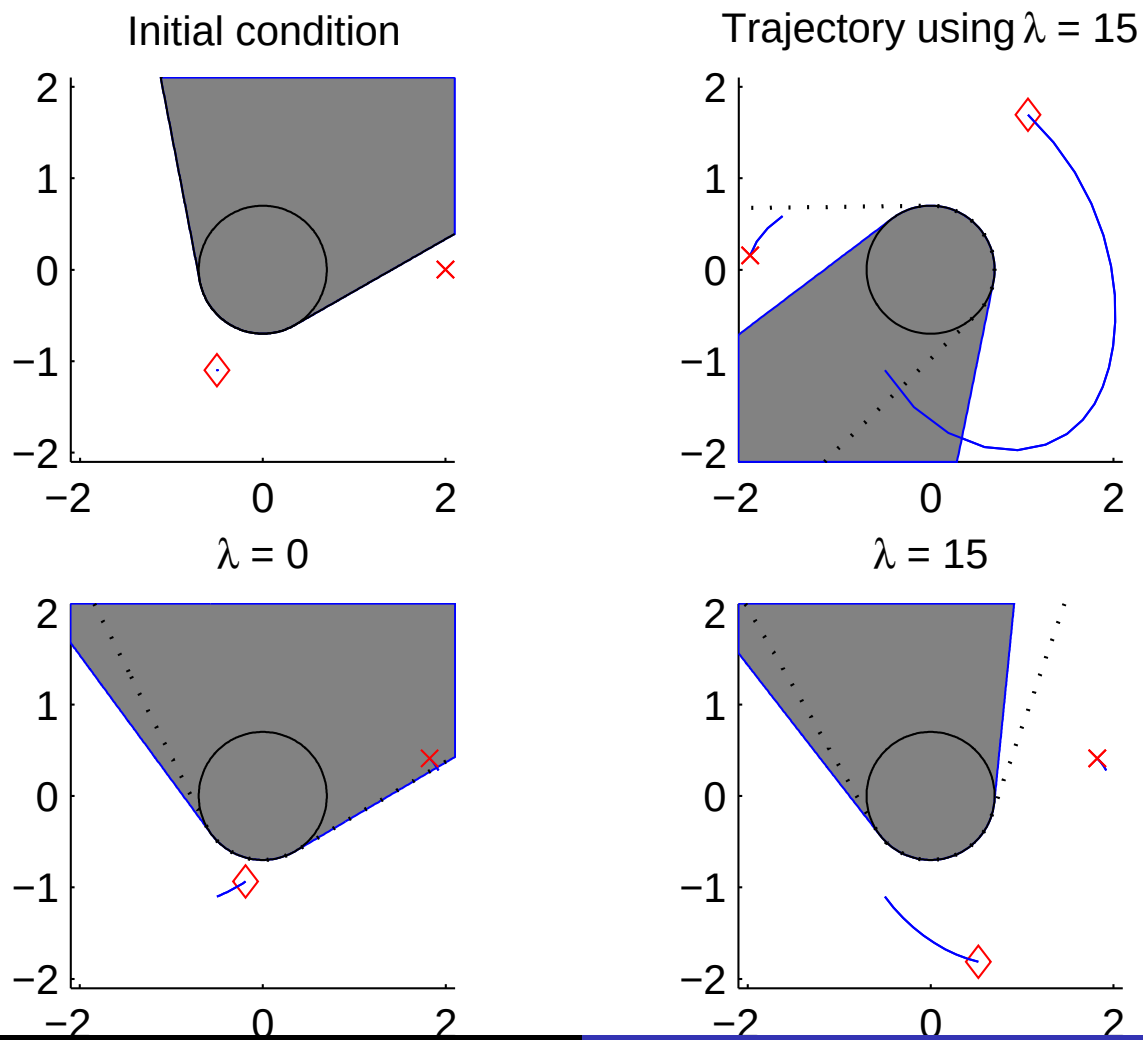
Taking advantage of the continuity



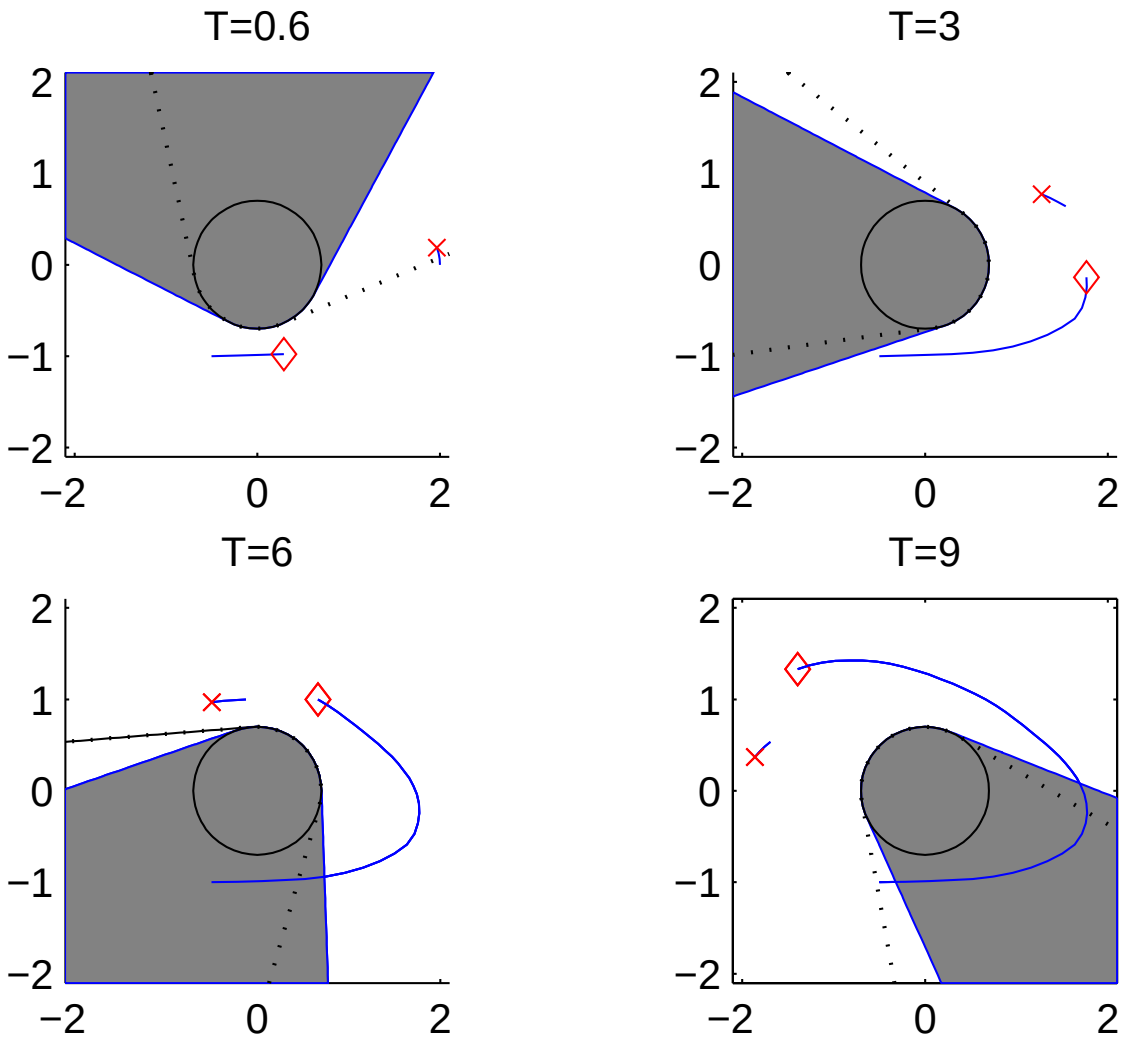
E.g. Move x_0 such that $u(B, x_0)$ is as "positive"as posible.

Keeping a moving object in sight

$$\max_{x(t;\alpha),\alpha\in\mathcal{A}} u(y(t);x(t)) \ s.t. \ |y-x| \sim d_0.$$



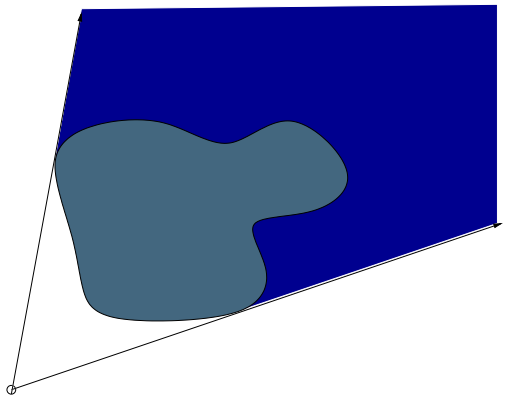
Keeping a moving object in sight



Volume based visibility optimization

Find x minimizing the area of invisible region outside of the occluders:

$$\min_x A(x) = \min_x \int_\Omega H(-u(y;x))H(\phi)dy$$



Continuity of u + compact $\Omega \implies$ continuity of A

Gradient descent:

$$\frac{dx}{dt} = -\nabla_x A(x) = \int_\Omega H(\phi)\frac{\nabla_x u(y;x)}{|\nabla_y u(y;x)|}\delta(u(y;x))|\nabla_y u(y;x)|dy.$$

Volume based visibility optimization

Generalization:

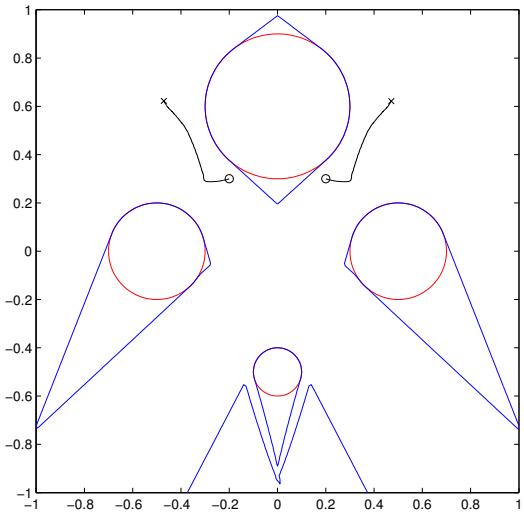
$$\min_{\{x_j\}} A(x) = \min_{\{x_j\}} \int_\Omega w(\phi, u, \Psi, y, \{x_j\}) H(-u(y; \{x_j\})) dy,$$

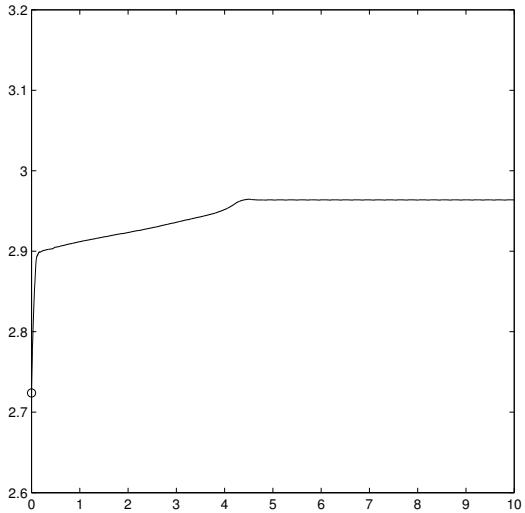
$w(\phi, u, \Psi, y, \{x_j\})$ is a weight function; e.g. near sighted — $w(d)$, where d =distance to the occluder.

In particular, maximize "illumination" on Γ:

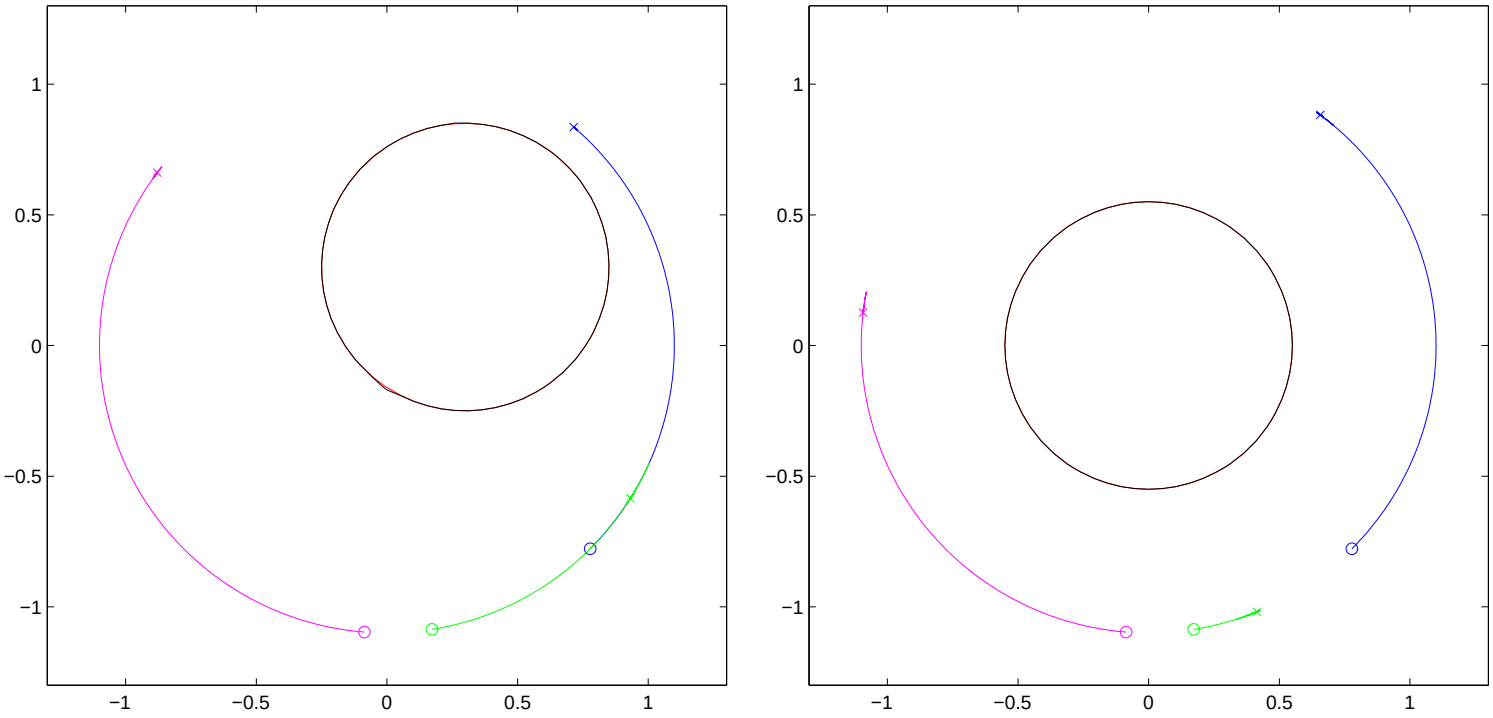
$$\min_x \int_\Omega \delta(\phi(y)) |\nabla \phi(y)| H(-u(y; x)) dy.$$

More examples

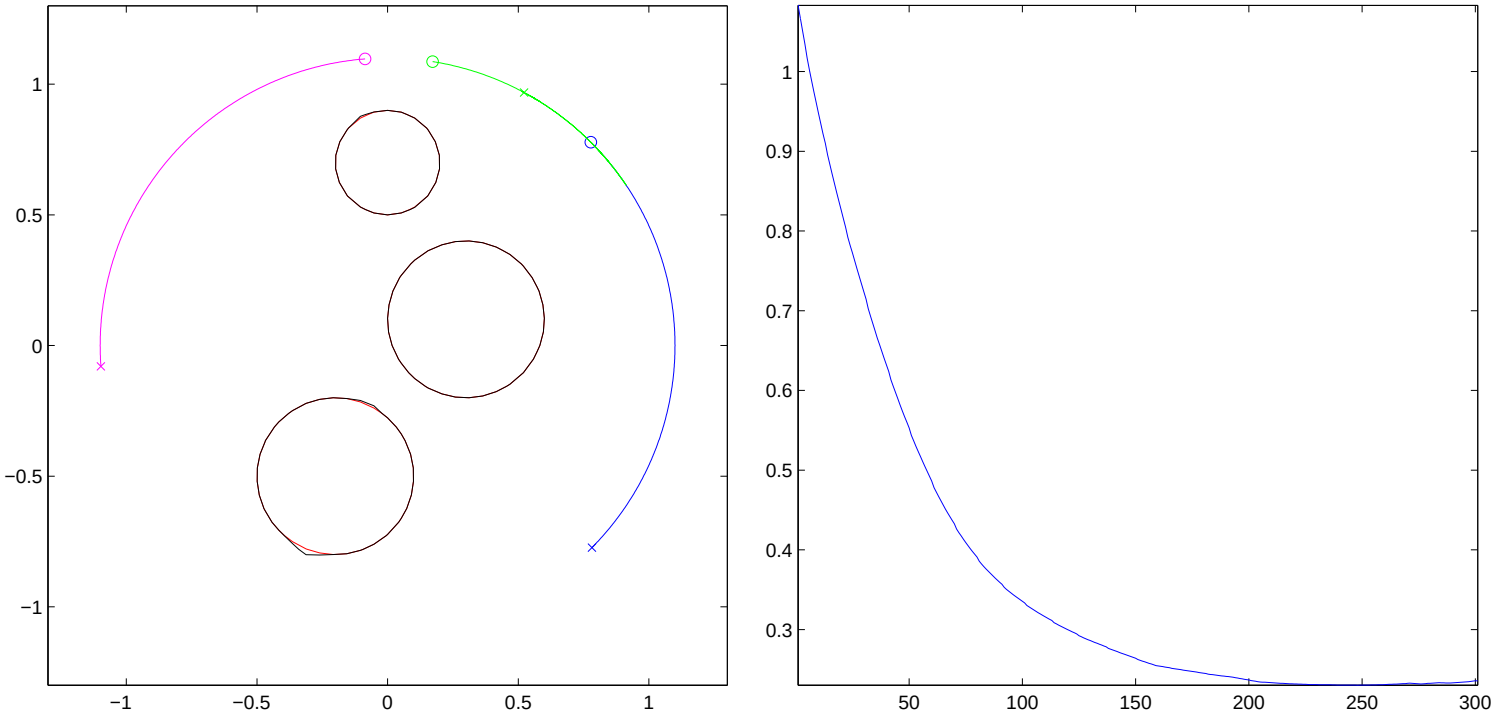




Guarding cameras



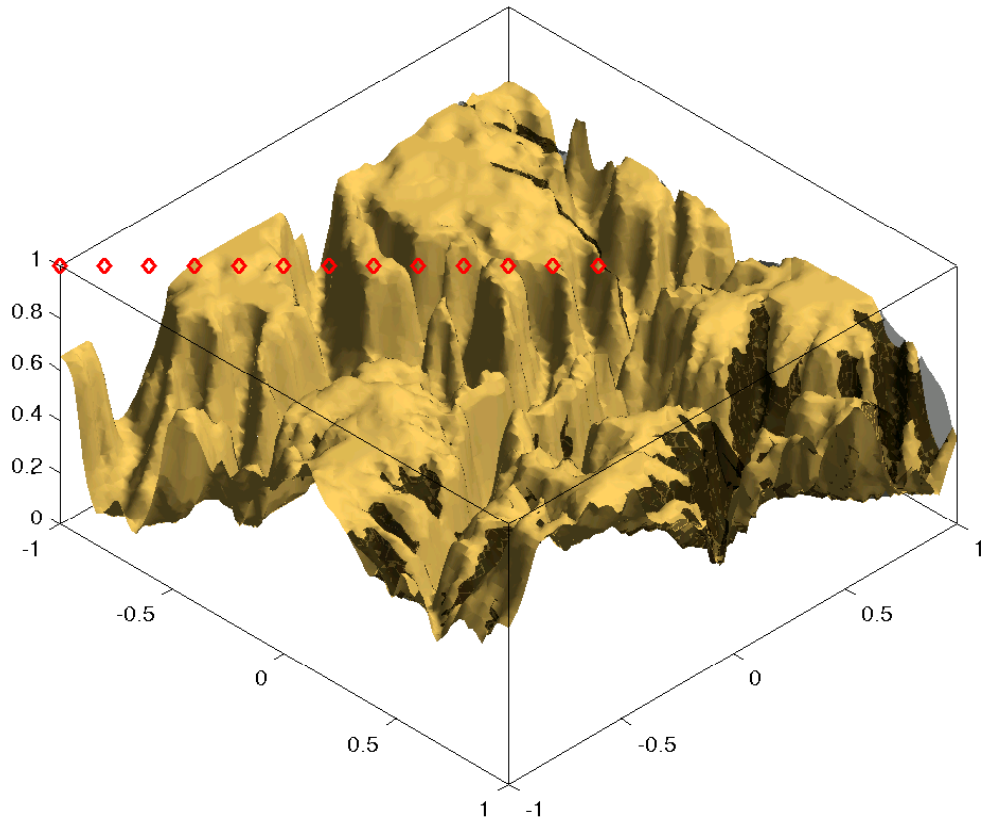
Guarding cameras



Visibility with memory

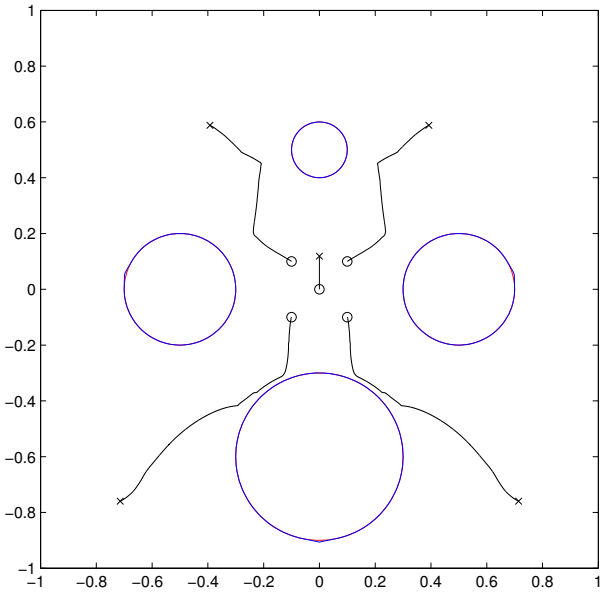
Consider the past trajectory as the vantage points:

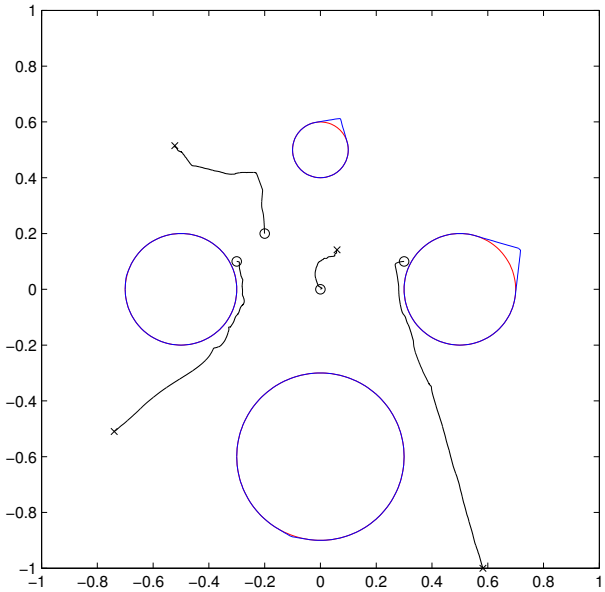
$$\min_{x(t)} A(x) = \min_{x(t)} \int_{\Omega} \omega(\mathbf{x}, \phi, \cdots) H(-\max_{s\in[0,t]} (u(y; x(s)))) H(\phi) dy.$$



Moving surfaces: $\phi(x, t)$ $\rightarrow$ $\dot{x}$, $\ddot{x}$?

Visibility with memory



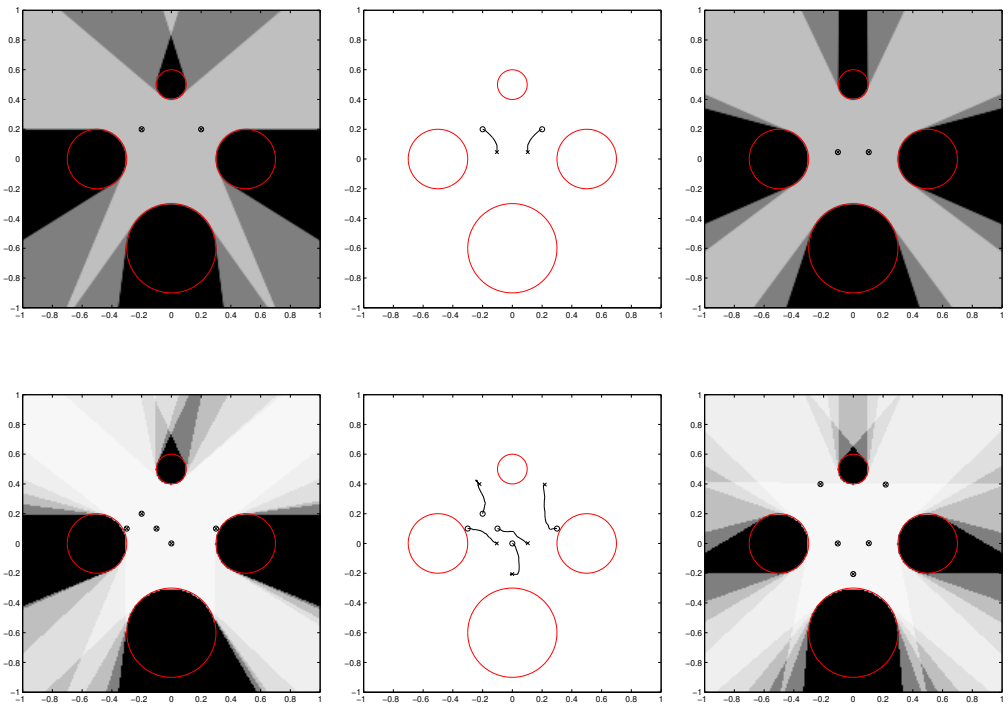


Average exposure

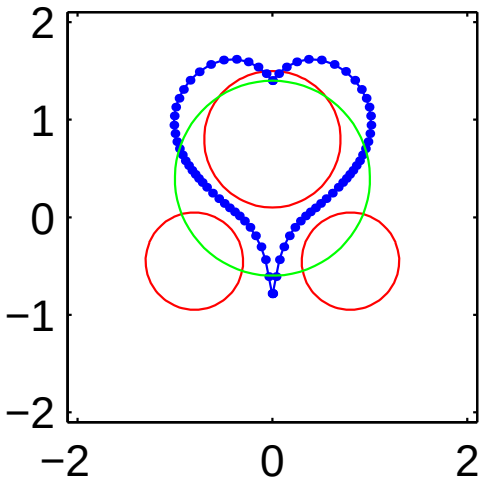
Exposure function: $\mathcal{X}(y;\{x_j\}) = \sum_j H(u(y;x_j))$.

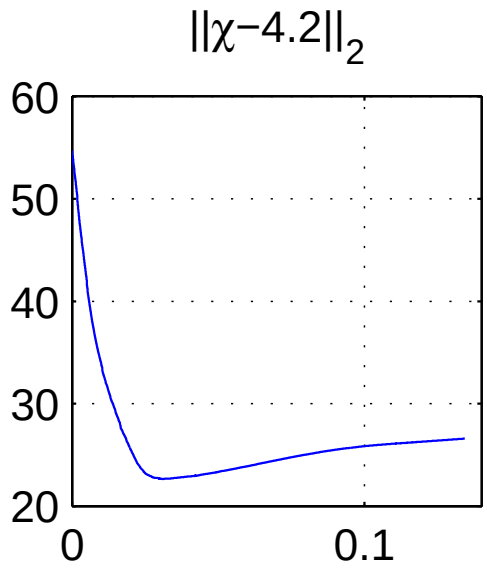
$$\mathcal{X}(y;\gamma) = \int H(u(y;\gamma(s))ds$$

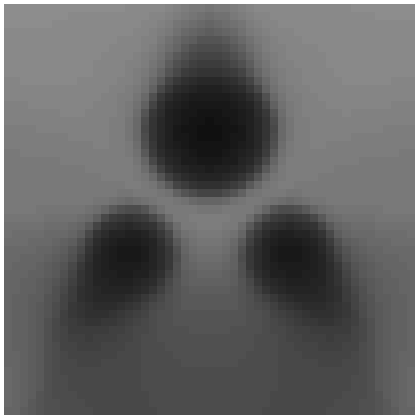
$$E(\gamma, C) = \frac{1}{2}||\mathcal{X}(\cdot;\gamma) - C||^2_{L^2} + \lambda \text{length}(\gamma).$$

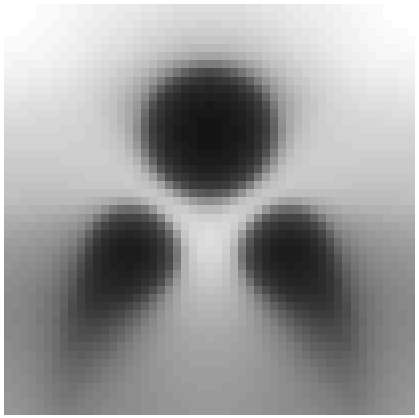


Path example 1

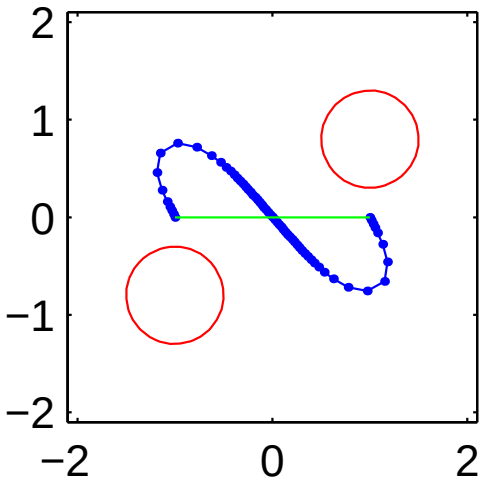


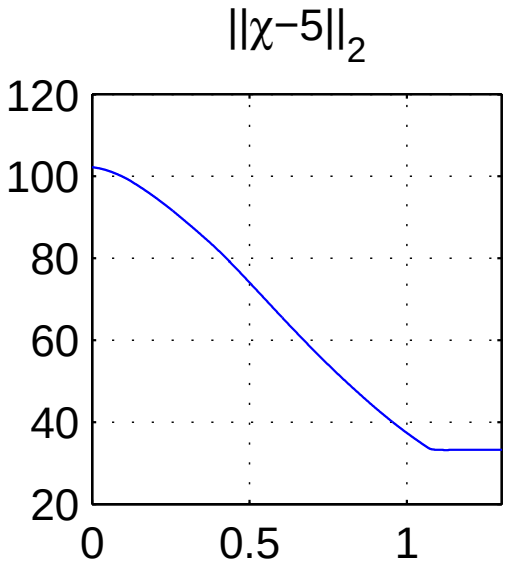


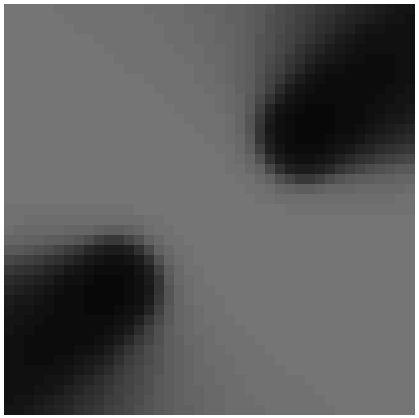


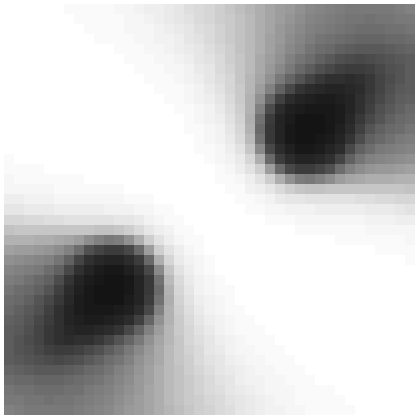


Path example 2









Spatial and temporal averaging

Optimal path $\gamma : [0,1] \mapsto R^d, \gamma(0) = A, \gamma(1) = B$.

Introduce weighting:

$$\tilde{\mathcal{X}}(t,x;\gamma) = \int K(x,t,\tau) H(u(x;\gamma(\tau)) |\gamma'(\tau)| d\tau,$$

and

$$\tilde{E}(\gamma, C) = \frac{1}{2} \int ||\tilde{\mathcal{X}}(t,\cdot,\gamma) - C||^2_{L^2} dt + G(\gamma)$$

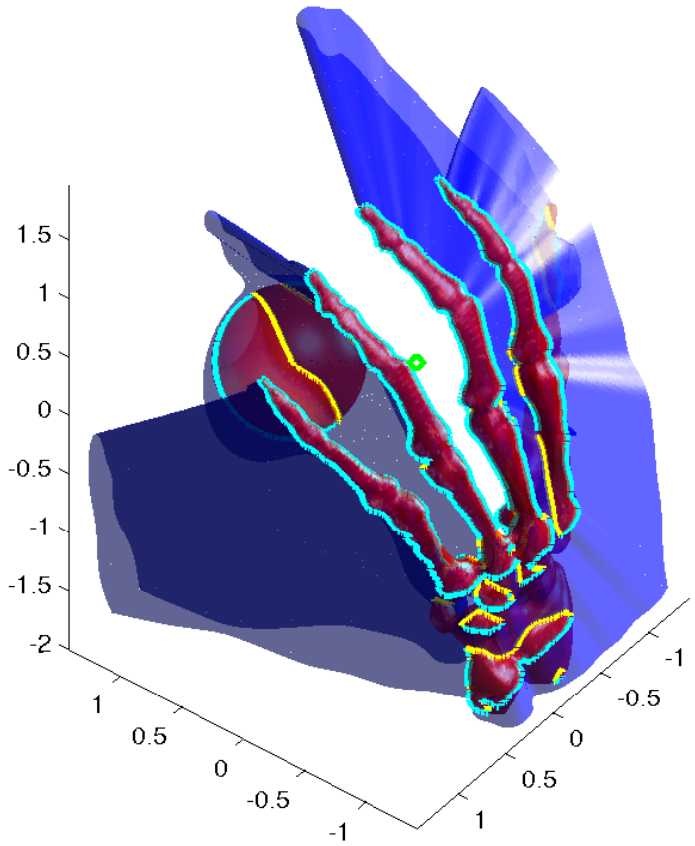
E.g. Temporal averaging = Memory:

$$\tilde{\mathcal{X}}(t,x;\gamma) = \int_{t-\delta t}^{t} H(u(x;\gamma(\tau)) |\gamma'(\tau)| d\tau.$$

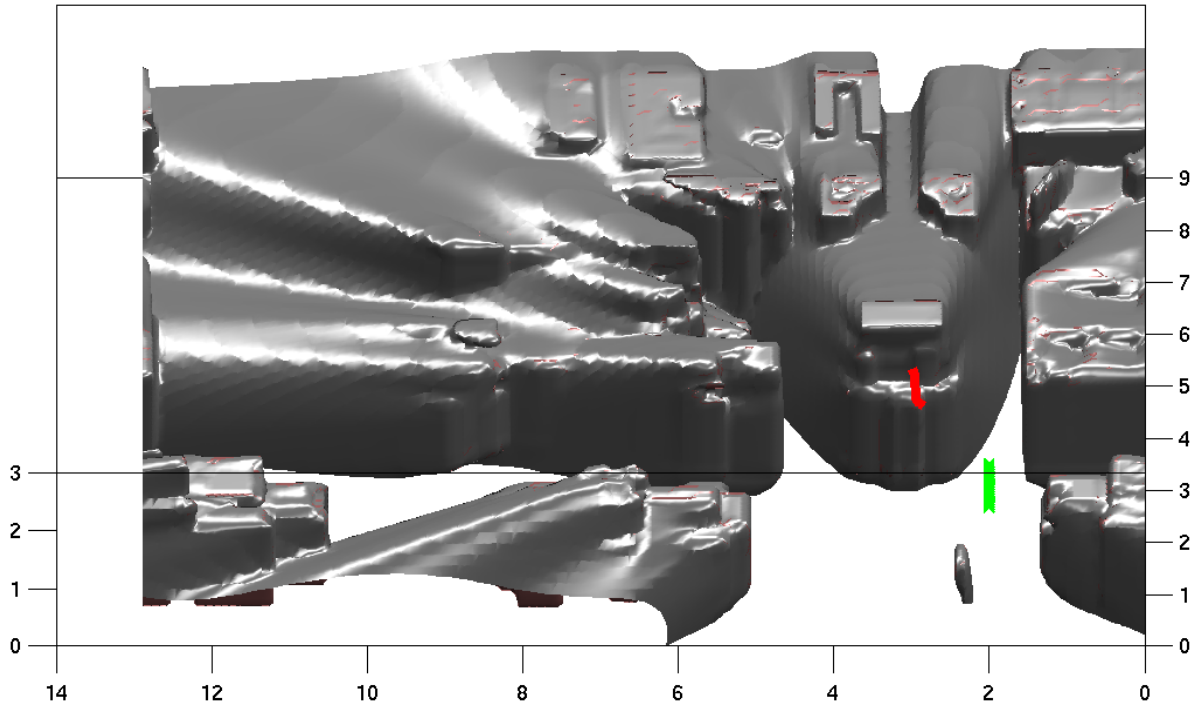
Visibility of occluder surface

$$\tilde{\mathcal{X}}(t,x;\gamma) = \int K(x,t,\tau) H(u(x;\gamma(\tau)) |\gamma'(\tau)| d\tau,$$

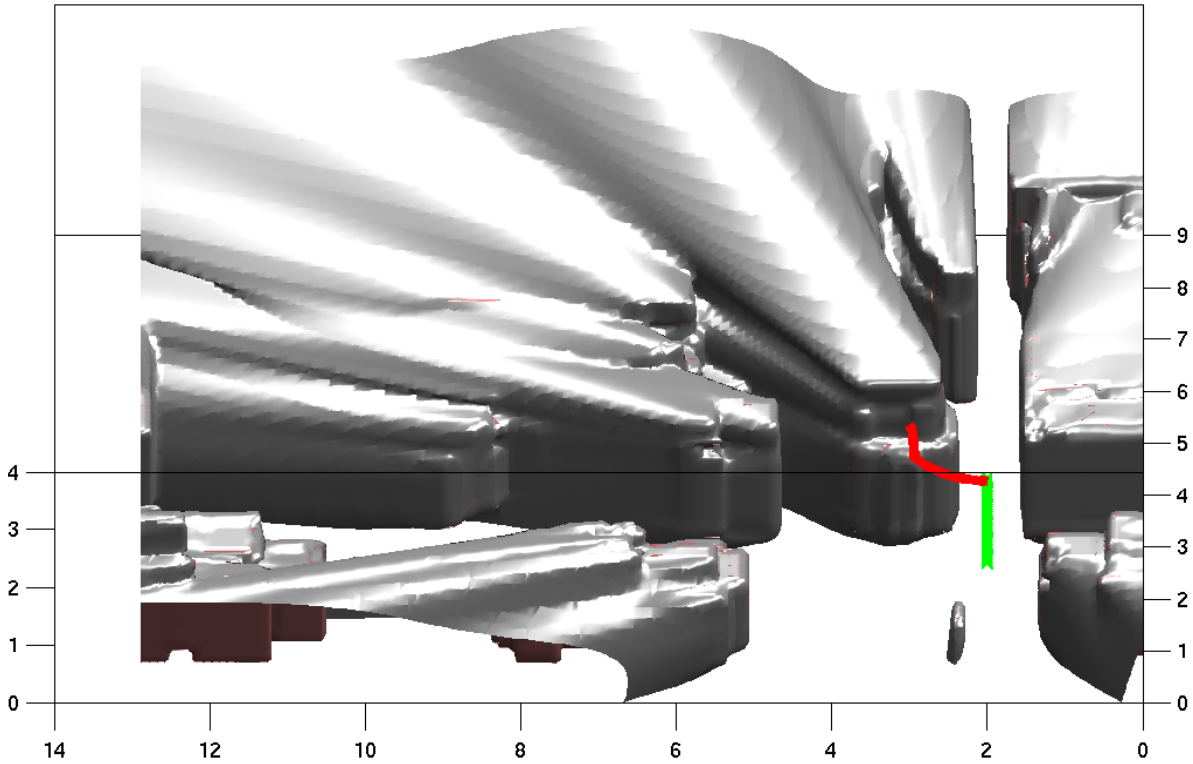
E.g. Visibility of occluder surface $K(x,t,\tau) = \delta(\phi(x))|\nabla\phi|$.



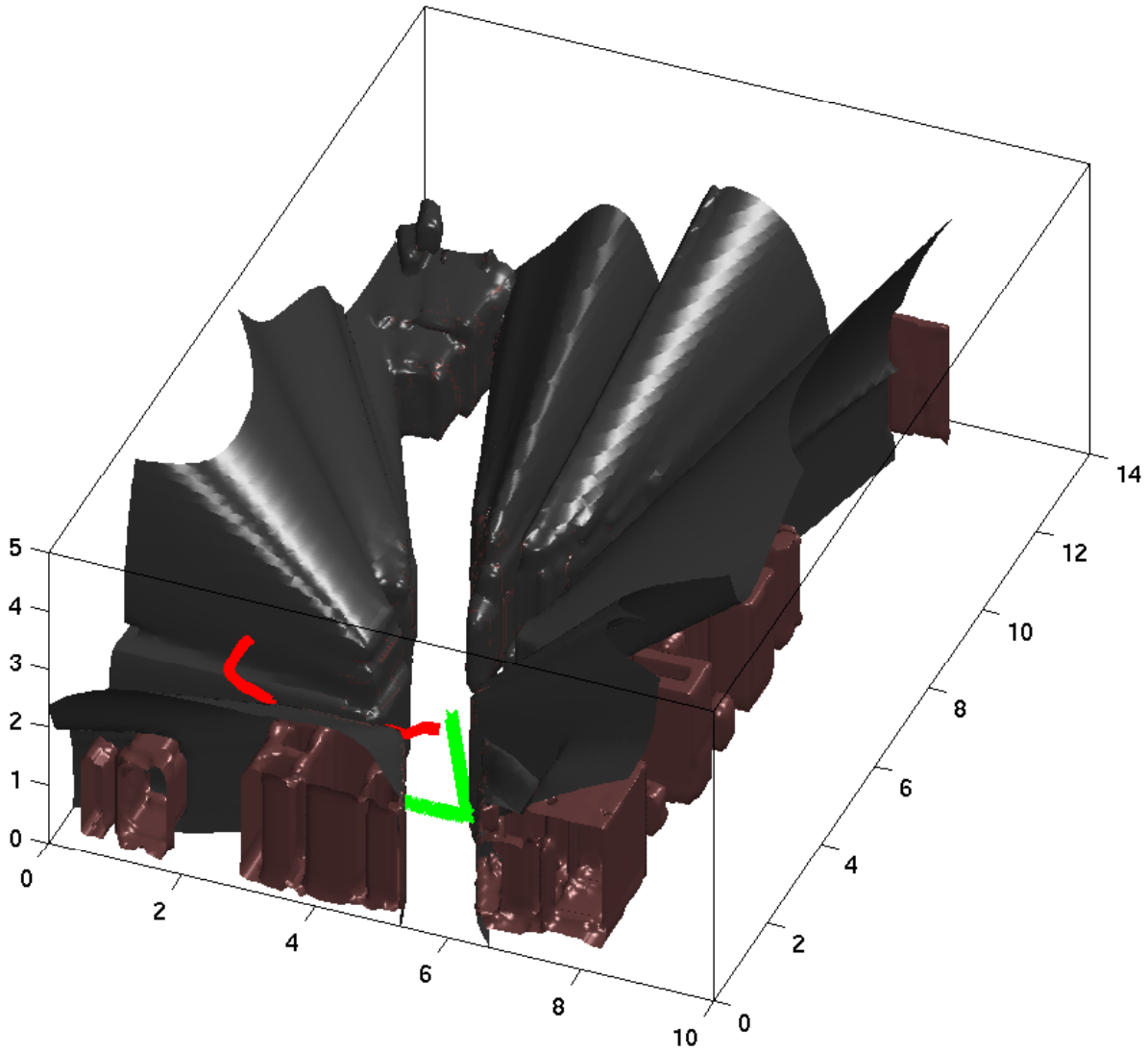
A Chase in DC



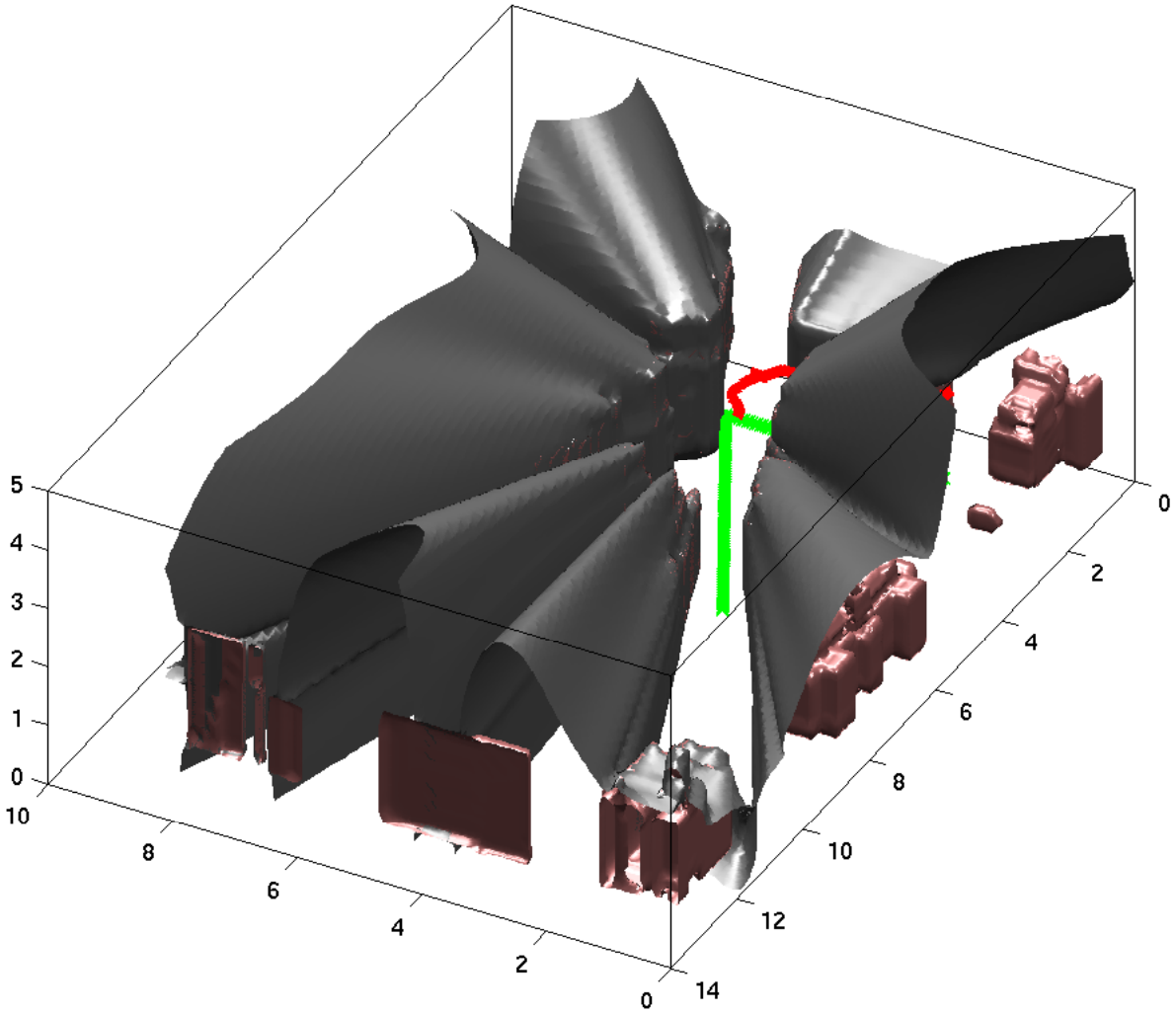
A Chase in DC



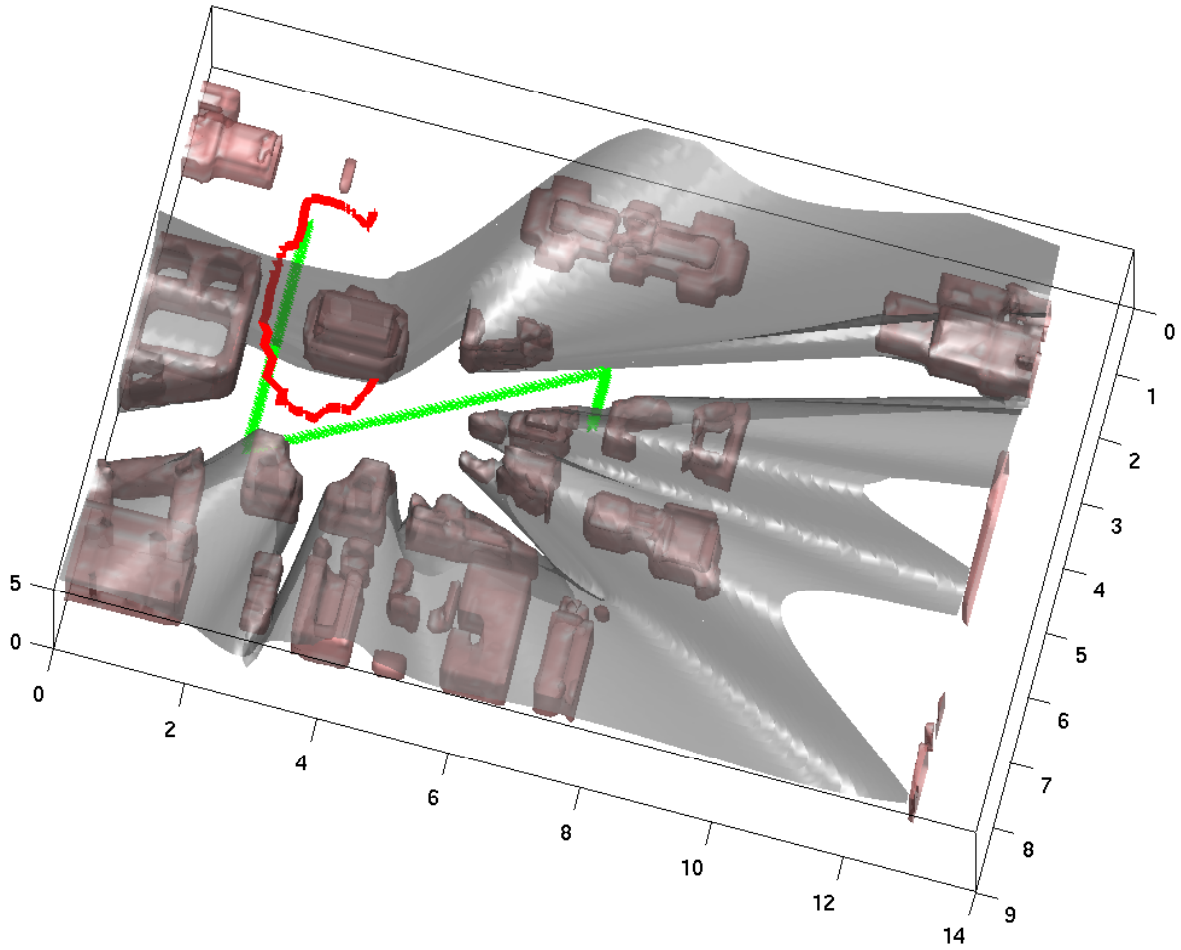
A Chase in DC



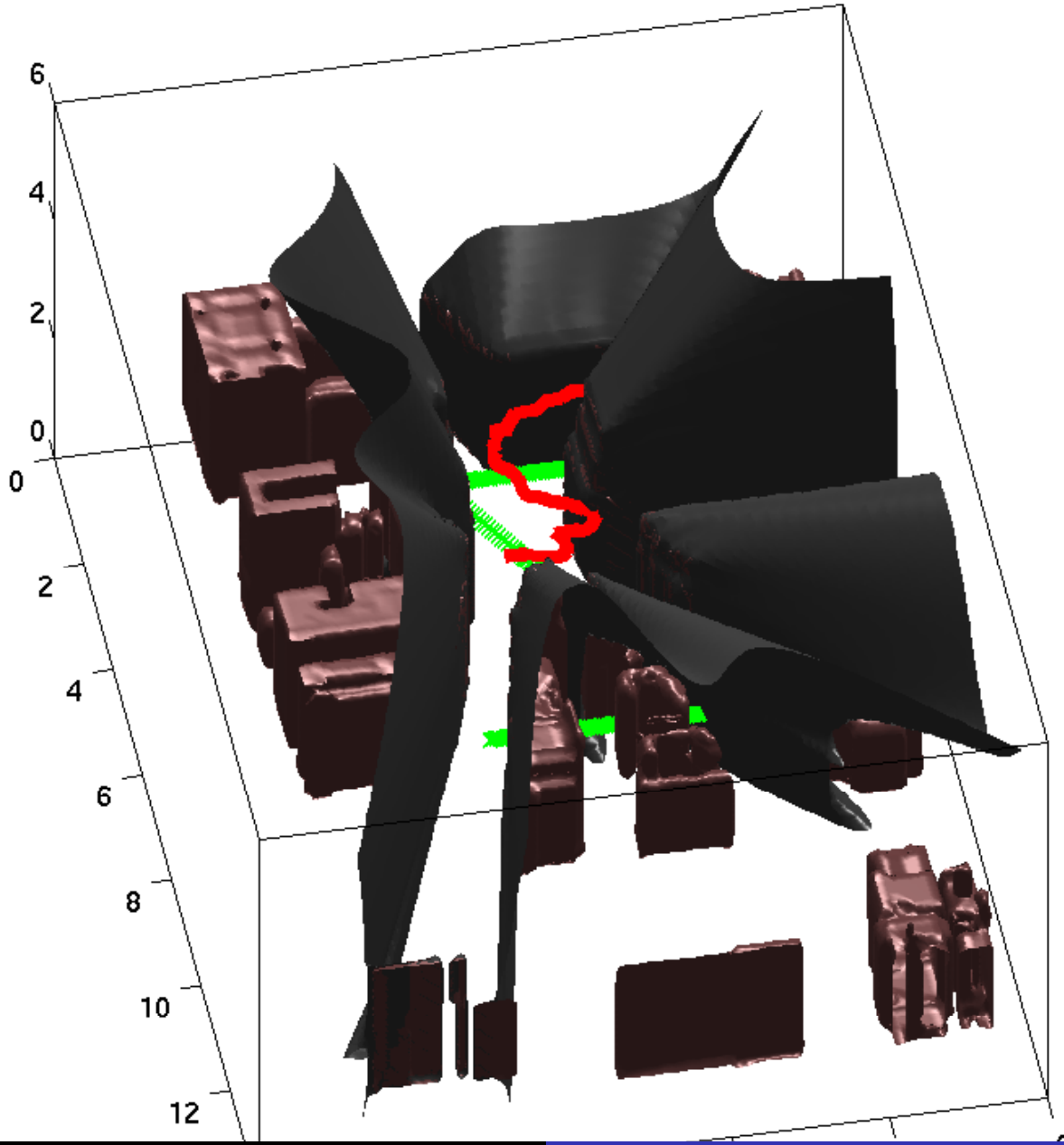
A Chase in DC



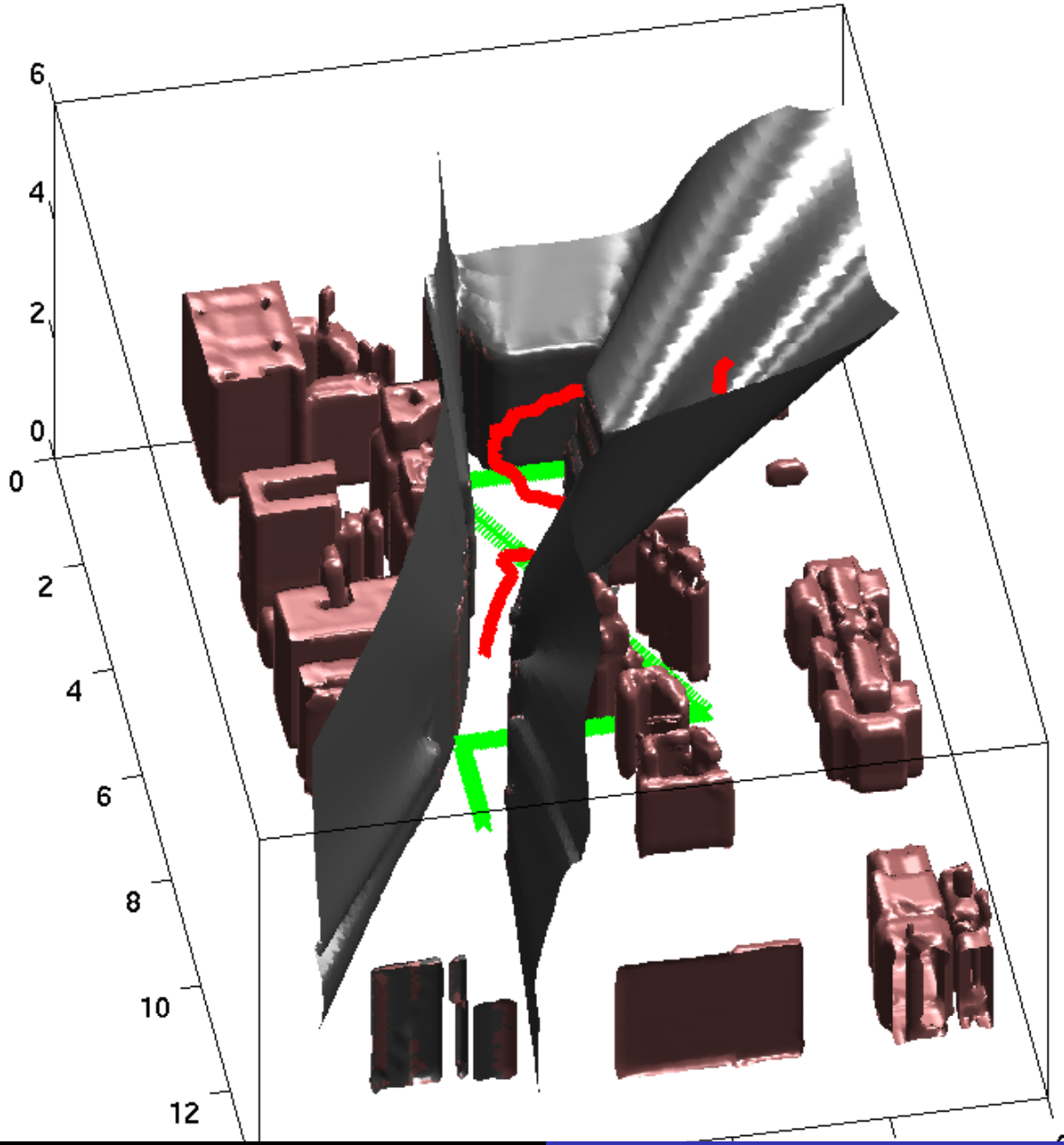
A Chase in DC



A Chase in DC



A Chase in DC



Summary

Under much simplified settings:

- A fast algorithm for constructing shadow boundaries (visibility)
- Adaptivity in the location of vantage points (sensors)

Challenges, current work, and wish list:

- Control and Game formulations [Kao-Navasca-Osher-Tsai]
- Bridging the shadow dynamics derived earlier with $\nabla_x u(y;x)$
- Analogues in other imaging mechanisms?
- Stochastic settings