Problem Set 2 — 2.086 Spring 2012

Released: *Tuesday, 28 February 2012, at 4 PM.*

Due: *Friday, 9 March 2012, at 5:00 PM by hardcopy.*

Please also upload to Stellar any MATLAB *function/script files you were required to supply by hardcopy in your problem set document.*

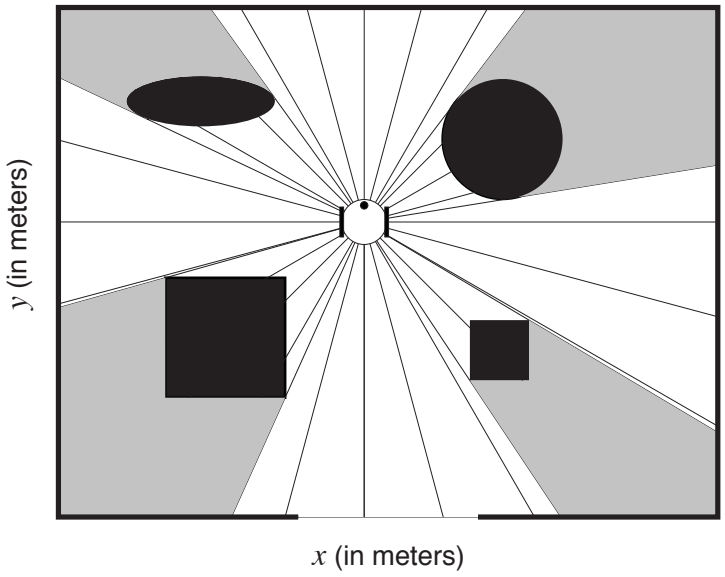


Figure 1: A surveillance robot scanning a room. Obstructions (in black) divide the space into a visible region (the white region) and a non-visible region (the gray and black regions).

Introduction

A museum has enlisted a camera-equipped mobile robot for surveillance purposes. The robot will navigate the museum's premises, pausing to take one or more 360 degree scans in each room. Figure 1 shows a typical room filled with various stationary obstructions (in black). We wish to determine the vantage point in each room from which the robot will have the most unobstructed view for its scan by *estimating the area of the visible region (in white) for each candidate vantage point*. (Note the obstructions are considered part of the non-visible region.) Considering that the robot might want to calculate a rough estimate of this result very quickly for real-time path modification purposes, we decide to use a Monte Carlo integration approach which deals easily with the complex "implicit" definition of the visible region and also will allow us to stop calculating as soon as the relative error of our estimate has reached an acceptable level.

We first define, for any vantage point $\boldsymbol{x}_V$ and any surveillance point (to be watched) in the room $\boldsymbol{x}_W$, the line segment $S(\boldsymbol{x}_V, \boldsymbol{x}_W)$ that connects $\boldsymbol{x}_V$ and $\boldsymbol{x}_W$. We can then express the area

visible from a vantage point $\boldsymbol{x}_V$ as the integral

$$A(\boldsymbol{x}_V) = \int_{\boldsymbol{x}_W \text{ in } R \text{ such that } S(\boldsymbol{x}_V,\boldsymbol{x}_W)\cap O=\varnothing} \mathrm{d}\boldsymbol{x}_W, \tag{1}$$

where R is the (rectangular) room and O is the collection of obstructions. The visible area is thus defined as the integral over all points in the room such that the line segment $S(\boldsymbol{x}_V, \boldsymbol{x}_W)$ between $\boldsymbol{x}_V$ and $\boldsymbol{x}_W$ does not intersect an obstruction (or, equivalently, such that the intersection of sets S and O is the null set).

There are many ways to do the visibility test $S(\boldsymbol{x}_V, \boldsymbol{x}_W) \cap O \stackrel{?}{=} \varnothing$, but perhaps the method most amenable to mobile robotics is to use an "occupancy grid," a discretization of the map in which each cell's value corresponds to the likelihood that the cell is empty or occupied. See the appendix for a description of how to use the occupancy-grid-based "visibility test" code we provide for your programming convenience.

Instructions

Download the two files `assignment2.mat` and `isvisible.m` from Stellar. Loading `assignment2.mat` in MATLAB will load all the data for this assignment.

Questions

1. (20 pts) We first consider the simple test case shown in Figure 2, where exactly one quarter of a 10 meters by 10 meters room R has been completely walled off by obstruction O. Calculate by hand the area visible from vantage point $\boldsymbol{x}_V = (x, y) = [2; 2]$. (Recall that points within the obstruction also count as non-visible.) Describe a very simple visibility test

   $$S(\boldsymbol{x}_V, \boldsymbol{x}_W) \cap O \stackrel{?}{=} \varnothing$$

   for this particular test room (Figure 2) and vantage point $\boldsymbol{x}_V = [2; 2]$, and then implement this test as a MATLAB function `test_isvisible.m` which takes as its single argument a surveillance point $\boldsymbol{x}_W$ and returns a logical 1 if the point is visible from vantage point $\boldsymbol{x}_V = [2; 2]$ and a logical 0 otherwise.

   The deliverables here are

   (*i*) $A(\boldsymbol{x}_V)$ (a numerical value) for the data given; and

   (*ii*) your very simple visibility test for this particular test room (Figure 2) and vantage point $\boldsymbol{x}_V = [2; 2]$: a mathematical/logical expression (not yet MATLAB code) in terms of $\boldsymbol{x}_W$ (and constants) which is true if and only if $\boldsymbol{x}_W$ is within the visible region; and

   (*iii*) your MATLAB function `test_isvisible.m` (which you should copy and paste into your problem set document and also upload to Stellar) which implements your test of (*ii*) above.

2. (40 pts) Construct a Monte Carlo code — based on Sections 10.2.2, 10.3, and 11.1.4 of the text — to estimate $A(\boldsymbol{x}_V)$ to within a relative error of 4%[1] for a 95% (normal-approximation

[1] Note by a relative error of 4% here (and elsewhere in the problem set) we mean RelErr = 0.04 (and not RelErr = 0.0004).

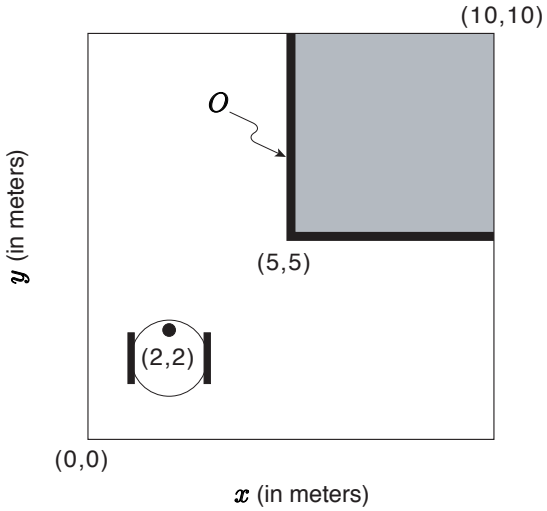


Figure 2: Test room.

binomial) confidence level. Note you are asked for the visible area, not the fraction of the room which is visible.

Test your Monte Carlo code for the test case of Question 1 using first, your own room-specific `test_isvisible.m` function of Question 1 and second, the provided general, occupancy-grid based `isvisible.m` function and occupancy grid `testmap` described in the appendix. Note you should run your Monte-Carlo code twice: once with your code `test_isvisible.m`, and once with "our" code `isvisible.m`.

The deliverables here are first,

(i) A Monte Carlo convergence plot for the case in which you use `test_isvisible.m` to determine visibility; and

(ii) A Monte Carlo convergence plot for the case in which you use `isvisible.m` to determine visibility.

Note in each case the plot should include title, axis labels (the abscissa label should be n (of Section 11.1.4) and the ordinate label should be "Visible Area (in meters squared)"), and legend. The plot should contain the information of Figure 11.2(a) but of course updated to reflect your particular calculation: you should include the visible area estimate, the visible area confidence interval, and the expected (exact) result for the visible area — all clearly labeled in the legend. Note you should not plot the confidence interval if $n\hat{\theta}_n \leq 5$ or $n(1-\hat{\theta}_n) \leq 5$.

Also please provide

(iii) A table which indicates, for both cases (in which you use `test_isvisible.m` and `isvisible.m`, respectively), (a) the value of n (for $n\hat{\theta}_n > 5$ and $n(1-\hat{\theta}_n) > 5$) at which the relative error first falls below 4%, and for that value of n the associated values of (b) the Monte Carlo visible area estimate, and (c) the visible area confidence interval (as lower and upper visible area bounds). Note the data in your table should be for the particular "runs" reported in your plots (i) and (ii) above.

3. (10 pts) For the provided occupancy grid `museummap` (corresponding to the map of Figure 3 and also Figure 4(a)) and using the provided `isvisible.m` function, use your Monte Carlo code to estimate (again, within a relative error of 4% for a 95% confidence level) the visible area from the vantage point $\boldsymbol{x}_V^{\text{large}} = [27.5; 21.5]$. Note: the required number of samples should be less than 5,000.

   The deliverables here are

   (*i*) A Monte Carlo convergence plot (with all the same attributes as in Question 2. except that you can no longer provide the exact result); and

   (*ii*) (*a*) the value of n at which the relative error first falls below 4%, and for that value of n the associated values of (*b*) the Monte Carlo visible area estimate, and (*c*) the visible area confidence interval (as lower and upper visible area bounds).

4. (15 pts) Repeat the analysis of Question 3, this time for the vantage point $\boldsymbol{x}_V^{\text{small}} = [7; 36.5]$. Explain why $\boldsymbol{x}_V^{\text{large}}$ and $\boldsymbol{x}_V^{\text{small}}$ require such a different number of samples to provide the same 4% relative accuracy. Note: the required number of samples should be less than 500,000.

   The deliverables here are

   (*i*) A Monte Carlo convergence plot with all the same attributes as in Question 3. (to avoid slowing down your code, make sure you do not plot inside the loop); and

   (*ii*) (*a*) the value of n at which the relative error first falls below 4%, and for that value of n the associated values of (*b*) the Monte Carlo visible area estimate, and (*c*) the visible area confidence interval (as lower and upper visible area bounds); and

   (*iii*) a short but quantitative explanation, in terms of relative error, for the larger Monte Carlo sample size required by the vantage point of Question 4. relative to the Monte Carlo sample size required by the vantage point of Question 3.

   We provide a comment which requires no action on your part: in this question, it would be much better to first eliminate a large part of the museum room which by inspection is clearly not visible and then perform Monte Carlo over the remaining part of the museum room (of course being careful to now interpret the "fraction of darts in" as the ratio of the visible area to the area of the remaining part of the museum room). This is known as "importance sampling" within the Monte Carlo literature.

5. (15 pts) Now try to find a candidate "best" vantage point for the robot in the museum room. Using Figure 4(a), choose 4 potentially good points for $\boldsymbol{x}_V$ and determine which one yields the largest visible area (again, within a relative error of 4% for a 95% confidence level). Provide the visible area estimate and confidence interval for each of your candidate vantage points, making sure to specify the one you found to be best.

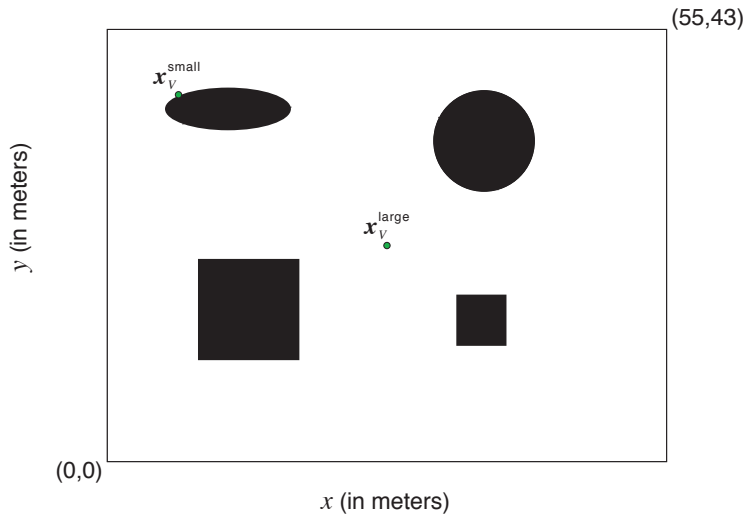


Figure 3: Museum room.

The deliverables here are

(i) a table which reports for each of your 4 vantage points the corresponding visible area estimate and confidence interval (as lower and upper visible area bounds); and

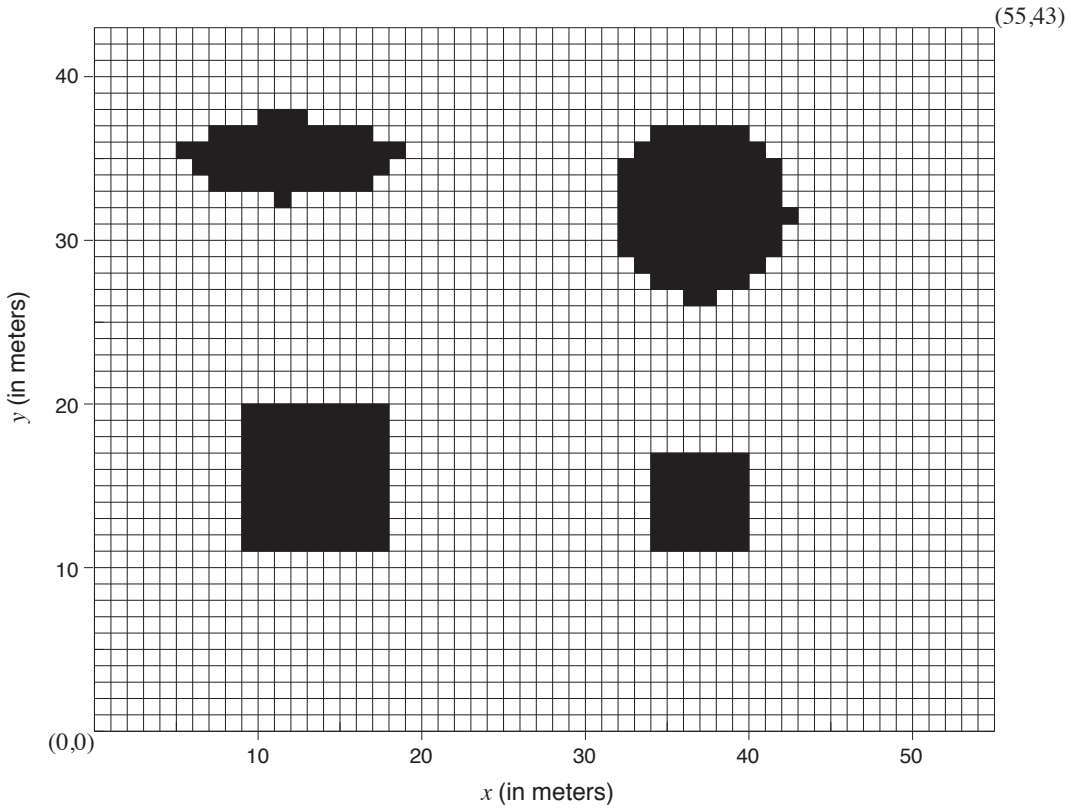
(ii) the vantage point which provides the best result (largest visible area); and

(iii) a short discussion of what the confidence intervals in your table say about your ability to distinguish the best vantage point (amongst your 4 candidates) based on your Monte Carlo calculations.

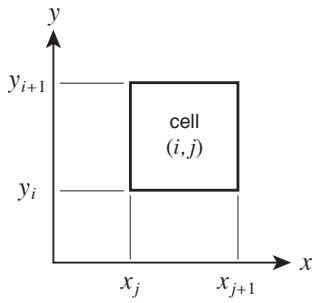
Appendix

We begin by converting our map of the room to an "occupancy grid," a discretization of the map in which each cell's value corresponds to the likelihood that the cell is empty or occupied. In our case, because we know ahead of time the layout of the room, a given cell contains either a zero if the cell is empty, or a one if the cell is occupied. Figure 4(a) shows a visualization of a fairly low-resolution occupancy grid for our map, where occupied cells are shown in black. Figure 4(b) shows the discretization convention we have adopted for our grid, in which the cell in row i and column j is defined by grid points x_j, x_{j+1} and y_i, y_{i+1}, where $1 \leq i \leq 43$, $1 \leq j \leq 55$; note the mesh spacing is 1 meter in each direction.

We can use the occupancy grid to determine the visibility of a point $\boldsymbol{x}_W$ in the room from a given vantage point $\boldsymbol{x}_V$. To do this, we draw a line between the two points, determine through which cells the line passes, and then check the occupancy condition of each of the intervening cells. If all of the cells are empty, the point is visible. If any of the cells are occupied, the point is not visible. The provided function `isvisible.m` takes as arguments $\boldsymbol{x}_V$, $\boldsymbol{x}_W$ (both 2×1 column vectors) and an occupancy grid of the room and performs this visibility check for you, returning a logical 1 if the specified point is visible from the specified vantage point and a logical 0 otherwise. For example, for



(a) Occupancy grid for `museummap`.



(b) Discretization scheme.

Figure 4: Occupancy grid and its discretization scheme. Note in Figure 4(a) each tick mark is one meter.

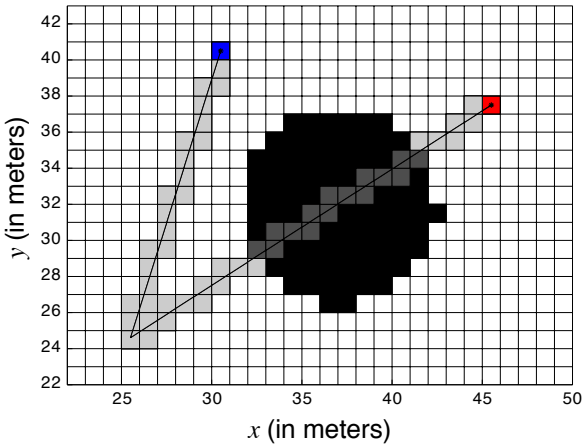


Figure 5: Visibility checking of two points from a single vantage point for the room `mapname`. Cell corresponding to visible point marked in blue; cell corresponding to non-visible point marked in red.

the two points in Figure 5, a call to `isvisible()` would return a logical 1 for the surveillance point at $[30.5; 40.5]$ (i.e., `isvisible([25.5; 24.5], [30.5; 40.5], mapname)` returns 1) and a logical 0 for the surveillance point at $[45.5; 37.5]$ (i.e., `isvisible([25.5; 24.5], [45.5; 37.5], mapname)` returns 0). Note the units of length are specific to each map; for the maps in this problem set length is measured in meters.

2.086 Numerical Computation for Mechanical Engineers
Fall 2012