Lab 4
Mapping

Learning an Environment

- **With a sensor and some memory, a robot can generate a map of an environment**

- **What does a robot need to know to make a map?**
  - Where obstacles are relative to the robot
    - LIDAR sensor readings!
  - Where the robot and obstacles are relative to some reference frame
    - Odometry and homogenous transforms

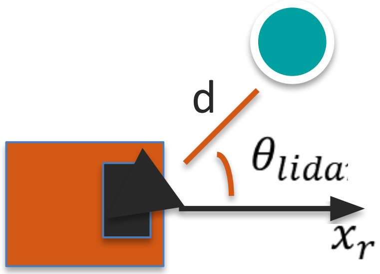
Recall: Homogenous Transform

- Instead of ${}^{A}Q = {}^{A}_{B}R * {}^{B}Q + {}^{A}P$, we can express the transformation as a single matrix multiplication

$$\begin{bmatrix} {}^{A}Q \\ 1 \end{bmatrix} = \left[\begin{array}{ccc|c} & {}^{A}_{B}R & & {}^{A}P \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \begin{bmatrix} {}^{B}Q \\ 1 \end{bmatrix}$$

Making a Map

- When the robot sees an object, it needs to localize it within its own reference frame:



The LIDAR sensor gives us distance-to-object (d)

The pose of the sensor gives us the angle to the object (θ_{lidar}) off the robot's x axis (x_r)

With d and θ_{lidar} you can compute the object's pose (x,y) in the robot's reference frame!

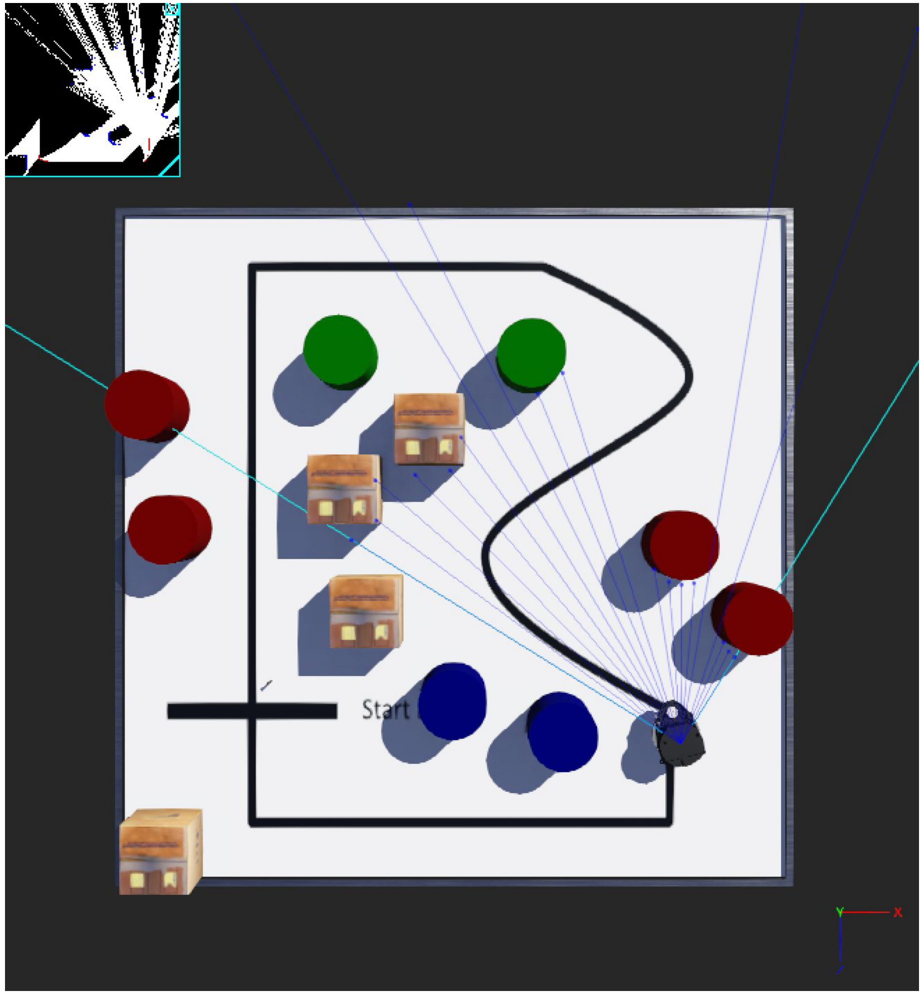
- Once the robot has the object's coordinates relative to itself, it needs to transform them to world coordinates:

$$^{A}Q = {}^{A}_{B}R * {}^{B}Q + {}^{A}P$$

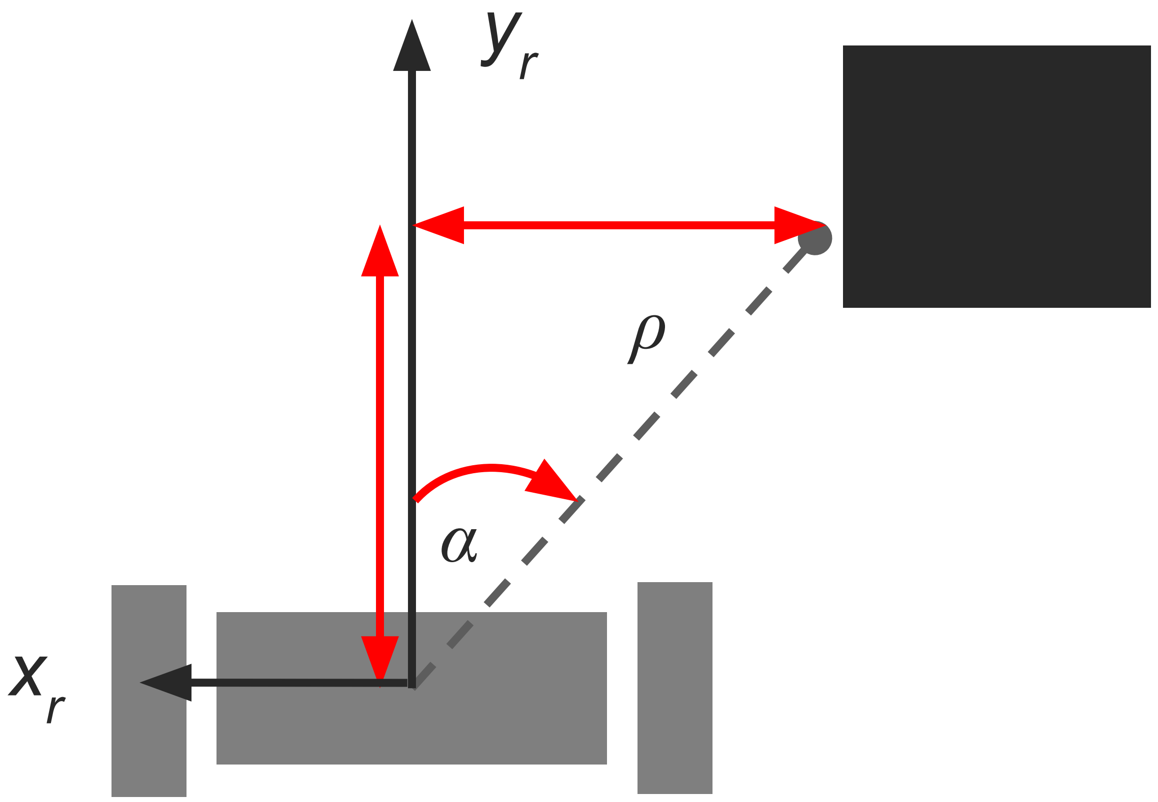
Representing the Map

- Once we know where obstacles are, we need to store them in a representation that allows for planning around them.
- For Lab 4, we'll implement a simple 4-connected grid representation as a 2D array of Boolean values. The display can do the job for this grid in this lab's case.
  - For row j and column i, **grid[j][i] = 0** if occupied, **1** if free space
- Since we're using a grid, each cell will represent a region of the world space instead of a single point.
  - If any obstacle is within the cell, we mark it as occupied.
  - To figure out where the robot is, we will use design and use functions that map world pose coordinates (x,y) to grid coordinates (i,j).

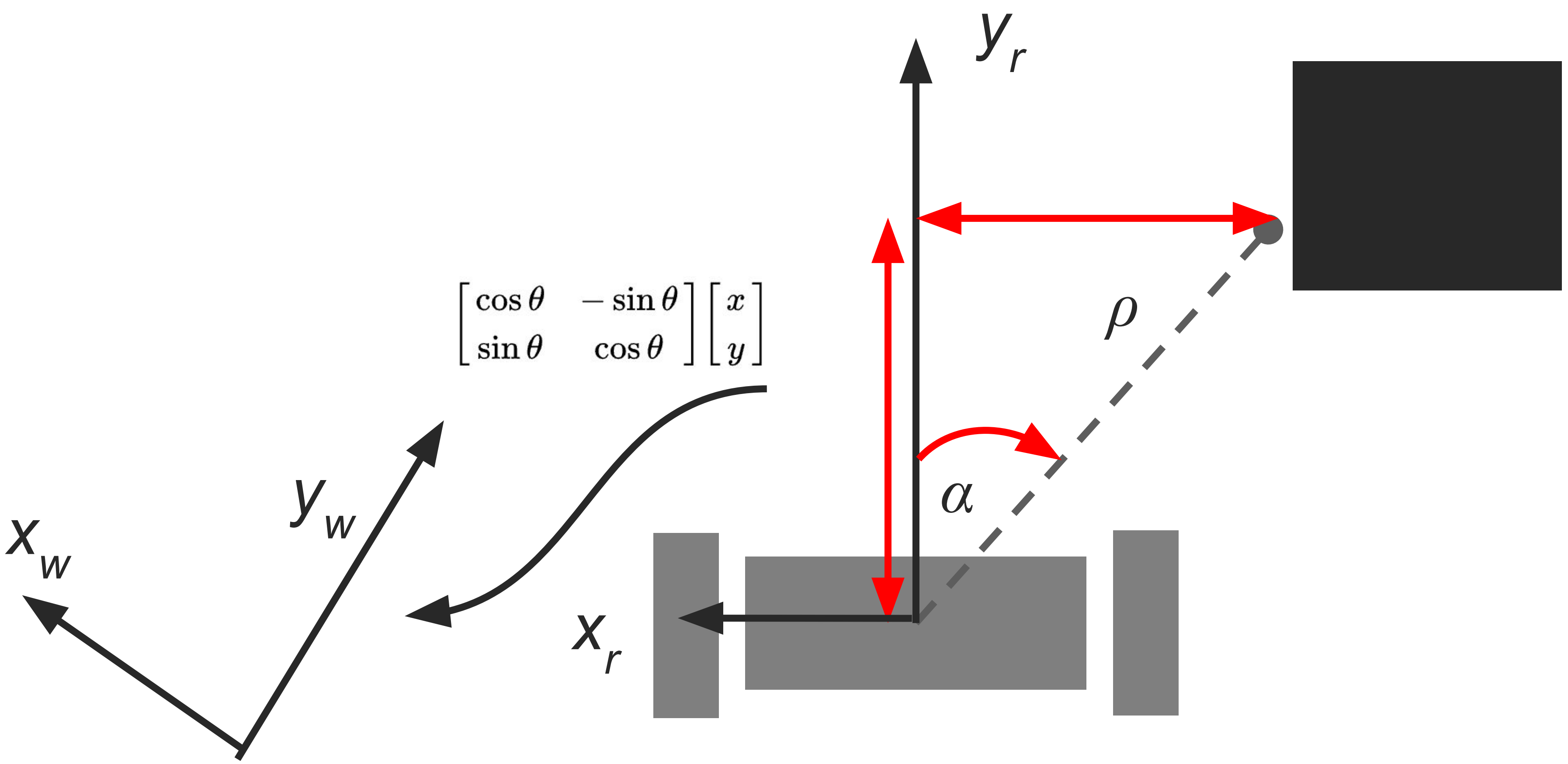
Lab Setup



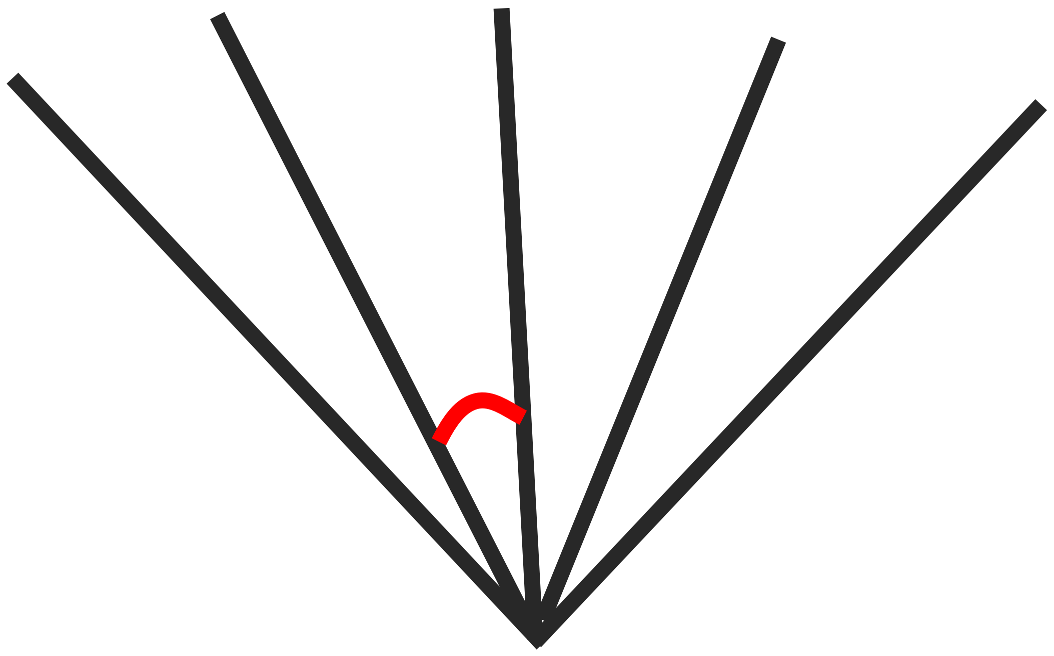
Turn measurements into robot coordinates



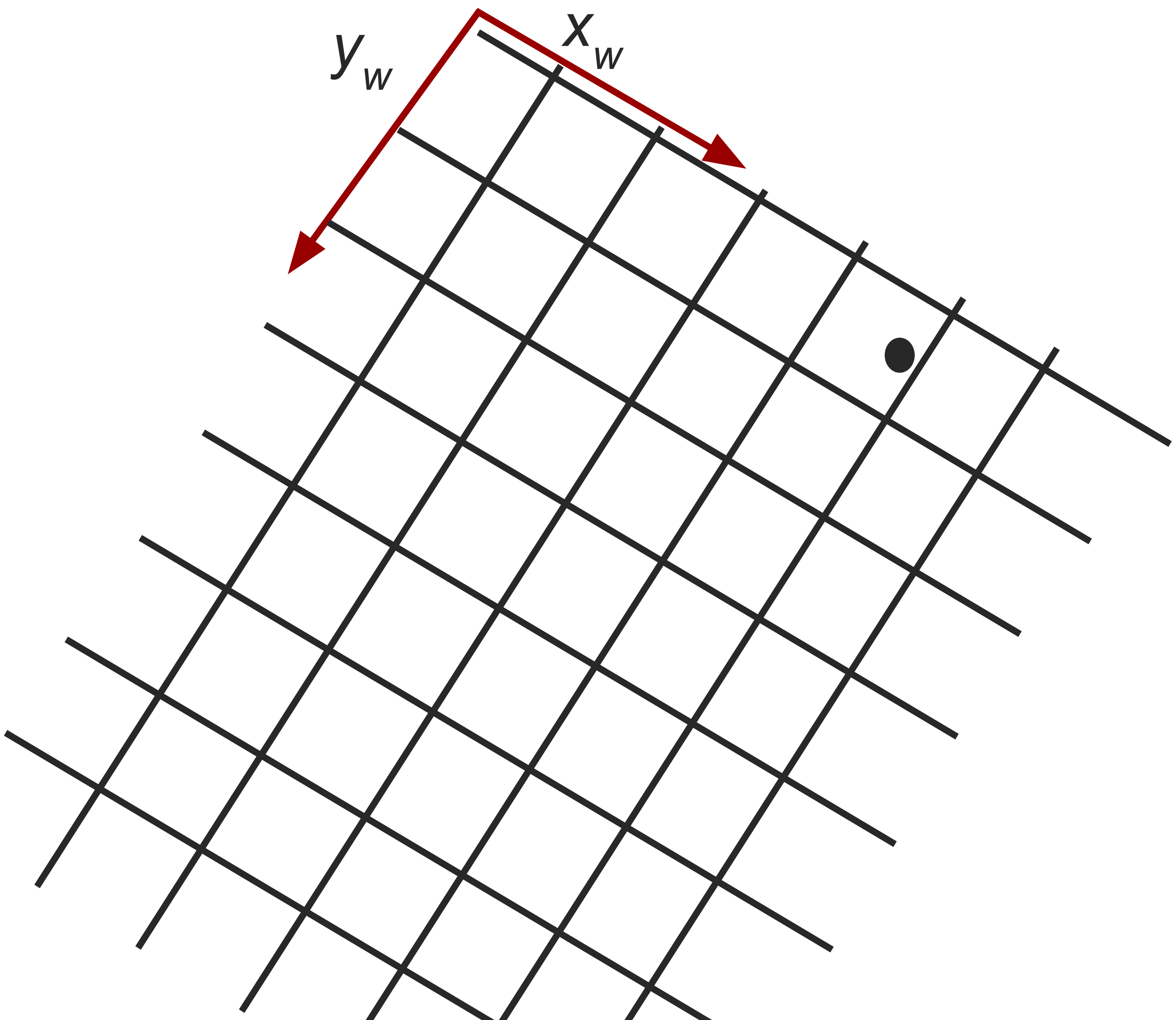
Turn robot coordinates into world coordinates



LiDAR rays are discreet and at uniform interval



Turn into world coordinates into map pixels



Algorithm

1. Calculate robot coordinates for each ray that is not ***inf***
2. Turn robot coordinates into world coordinates
3. Draw obstacle (pixel) and free space onto the map
4. Draw robot location onto the map
5. Move forward

- **This is a 1-week lab. Due Tuesday 3/1 at 11:59pm.**
- In case, you don’t want to use the homogenous transform matrices but just basic sin/cos projections then I have made a Piazza hints post with an arbitrarily chosen robot local frame axes.


Questions?