optimized_code

The optimized_code folder contains a copy of Andy's histogram filter code as well. This is where you can tweak the code for optimization. These files have extra comments giving some ideas about what to optimize.

instructions.md

This file contains further instructions about how to complete and run the project.

hints.md

The optimized_code files already have ideas about what to optimize. The hints file contains more explicit ideas but does not give the actual code. You are not required to read this file, and we encourage you to try optimizing the code first before reading the hints file.

Explanation of the Code

Here is an explanation of how the histogram filter code works.

In main.cpp, you'll find a vector called grid. Grid is a char vector holding the color values of a 2-dimensional square grid.

The **initialize_beliefs** function takes in the char grid and outputs the initial probabilities for each square on the grid.

Then the **sense** function takes a measurement of the current grid space's color. The measurement is used to update the probabilities of each space on the grid.

Next, the **blur** function passes the grid through a smoothing algorithm.

Then the probabilities are normalized with the **normalize** function.

Finally, the robot moves to a new space on the board, and the probabilities are updated appropriately.

Each function is run ten-thousand times. You can adjust the number of times by changing the value in the iterations variable.

Order

As a suggestion, do the project in the following order: