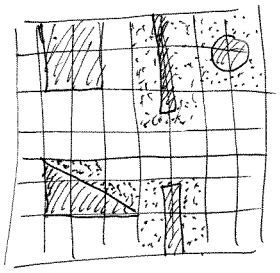
Occupany Grids / Grid Maps

what are some ways a robot can navigate?

- reactive "controls"
  - bug / wall following
  - "follow the leader"
  - etc.
- explicitly planned paths
  - RRT
  - Wavefront
  - ...



Grid maps are useful as output + for path planning

- human consumption
- long term localization (scan matching)
- underlying graph structure

Nice Properties:

- easy to implement
- straightforward to represent spectrum of traversal costs
- easy to extend to arbitrary dimensions (though we're going to stick to 2D)
- cost guarantees for planning w/ easy to understand parameters

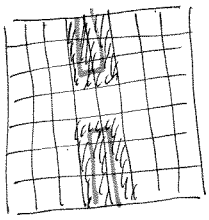
Sad Properties:

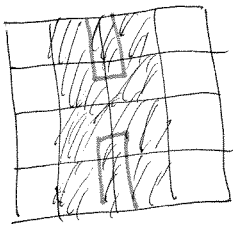
- quantization error
- fine resolutions quickly become cost prohibitive
- issues w/ dynamic environments
- consistency / maintenance problems (localization is very important)

Tradeoffs: how do you pick a scale for your grid?

- higher resolution: more accurate representation of environment (to some extent), but expensive
- lower resolution: fast planning, but go too coarse and traversible routes may disappear!

eg. Doorways





Rule of thumb: use as coarse a resolution as you can afford

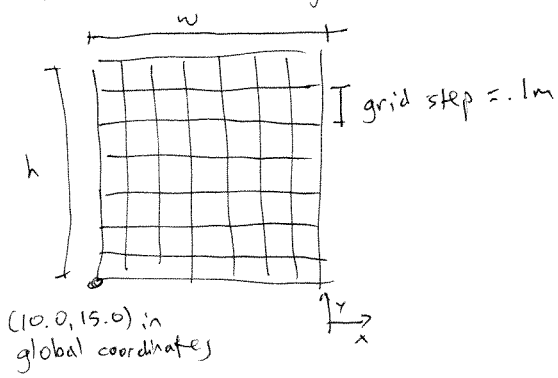
Data Representation

- Typically ternary (occupied, known free space, unknown)

  This allows you to easily write frontier-based exploration planners which send your robot(s) to points where reachable known free space meets unknown

Writing A gridmap

- Need a consistent reference w.r.t. world if you will be adding observations over time
- typically, we use axis-aligned coordinates and designate some corner as our reference point in the "real world". e.g.



- Now you can feed point observations into your map to set grid values, e.g.

```
grid_set(grid, global_x, global_y, OCCUPIED)
```