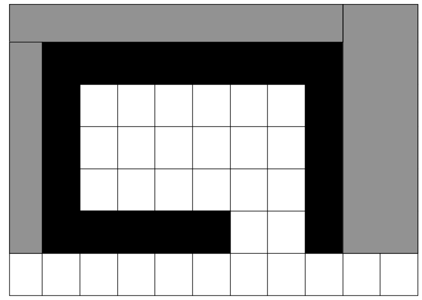
Lecture 13: Occupancy Grids

CS 344R/393R: Robotics

Benjamin Kuipers

Occupancy Grid Map



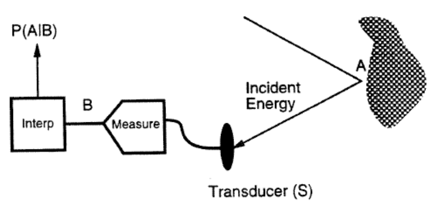
Occupancy Grid Map

- Maps the environment as an array of cells.
  - Cell sizes range from 5 to 50 cm.
- Each cell holds a probability value
  - that the cell is occupied.
- Useful for combining different sensor scans, and even different sensor modalities.
  - Sonar, laser, IR, bump, etc.
- No assumption about type of features.
  - Static world, but with frequent updates.

A Bit of History

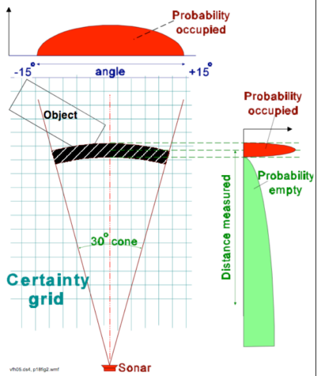
- Occupancy grids were first popularized by *Hans Moravec* and *Alberto Elfes* at CMU.
- *Kurt Konolige* at SRI made a number of valuable contributions.
  - Konolige's Erratic robot is the ancestor to the Amigobot. Konolige developed Saphira, too.
- *Hugh Durrant-Whyte* and *John Leonard* (then at Oxford) used landmarks and Kalman filters as an alternative.
- *Sebastian Thrun* (then CMU, now Stanford) has done very impressive metrical mapping work, which we will study.

Sonar Sensors Give Evidence of Obstacles



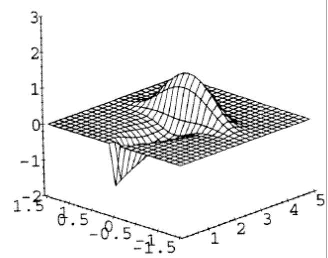
Sonar Sweeps a Wide Cone

- Obstacle could be anywhere on the arc at distance D.
- The space closer than D is likely to be free.



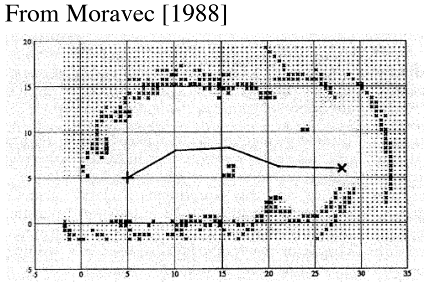
Occupancy from Sonar Return

- One 2D Gaussian for information about occupancy.
- Another for free space.



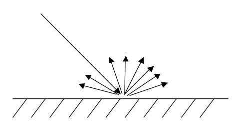
Wide Sonar Cone Creates a Noisy Map

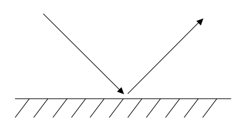
- From Moravec [1988]



Diffuse and Specular Reflections

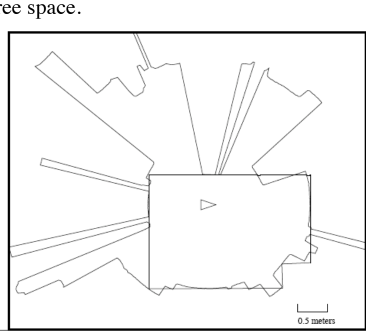
- Diffuse
- Specular





Specular Reflections in Sonar

- Specular (multi-path) reflections hallucinate free space.

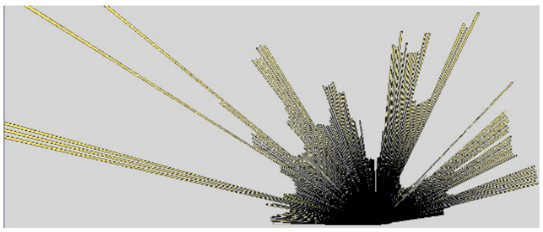


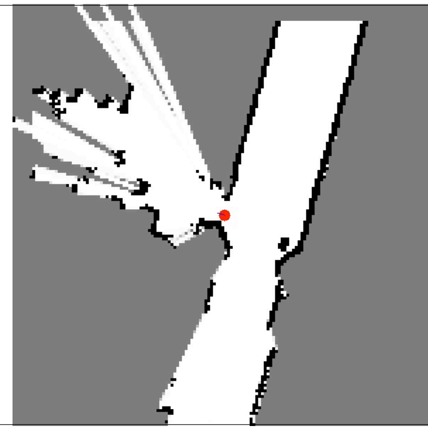
Laser Range Finder

- 180 ranges over 180° planar field of view
- 10-12 scans/second
- 4 cm range resolution
- Max range 50-80 m.
- Problems with mirrors, glass, and matte black.
- Much better than sonar!

Laser Rangefinder Image

- 180 narrow beams at 1° intervals.





Occupancy Grid Cells C_{ij}

- The proposition $occ(i,j)$ means:
  - The cell C_{ij} is occupied.
- **Probability**: $p(occ(i,j))$ has range [0,1].
- **Odds**: $o(occ(i,j))$ has range $[0,+\infty)$.

$$o(A) = \frac{p(A)}{p(\neg A)}$$

- **Log odds**: $\log o(occ(i,j))$ has range $(-\infty,+\infty)$
- Each cell C_{ij} holds the value $\log o(occ(i,j))$
  - $C_{ij} = 0$ corresponds to $p(occ(i,j)) = 0.5$

Probabilistic Occupancy Grids

- We will apply Bayes Law

$$p(A \mid B) = \frac{p(B \mid A) * p(A)}{p(B)}$$

  - where A is $occ(i,j)$
  - and B is an observation $r=D$

- We can simplify this by using the log odds representation.

Bayes Law Using Odds

- Bayes Law: $$p(A \mid B) = \frac{p(B \mid A) * p(A)}{p(B)}$$
- Likewise: $$p(\neg A \mid B) = \frac{p(B \mid \neg A) * p(\neg A)}{p(B)}$$
- so: $$o(A \mid B) = \frac{p(A \mid B)}{p(\neg A \mid B)} = \frac{p(B \mid A) * p(A)}{p(B \mid \neg A) * p(\neg A)}$$ $$= \lambda(B \mid A) * o(A)$$
- where:

$$o(A \mid B) = \frac{p(A \mid B)}{p(\neg A \mid B)} \qquad \lambda(B \mid A) = \frac{p(B \mid A)}{p(B \mid \neg A)}$$

Easy Update Using Bayes Law

- Bayes' Law can be written:

$$o(A \mid B) = \lambda(B \mid A) * o(A)$$

- Take log odds to make multiplication into addition.

$$\log o(A \mid B) = \log \lambda(B \mid A) + \log o(A)$$

- Easy update for cell contents.

Occupancy Grid Cell Update

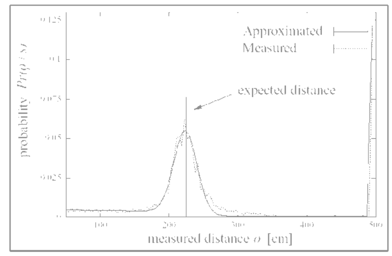
- Cell C_{ij} holds $\log o(occ(i,j))$.
- Evidence $r=D$ means sensor r returns D.
- For each cell C_{ij} accumulate evidence from each sensor reading.

$$\log o(A \mid B) = \log \lambda(B \mid A) + \log o(A)$$

$$\log o(occ(i,j)) + \log \lambda(r = D \mid occ(i,j)) = \log o(occ(i,j) \mid r = D)$$

Sensor Model $p(r=D \mid occ(i,j))$

- Probability of range-reading given known occupancy at a known distance.



Update Values for λ

- If the laser terminates at C_{ij} at distance D

$$\lambda(r = D \mid occ(i,j)) = \frac{p(r = D \mid occ(i,j))}{p(r = D \mid \neg occ(i,j))} \approx \frac{.06}{.005} = 12$$

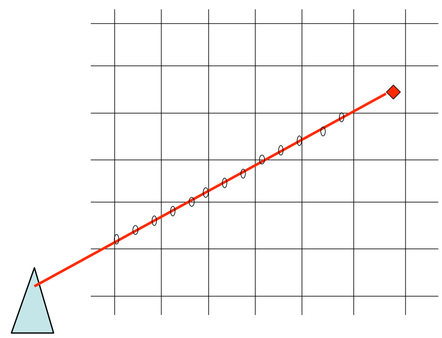
  – so $\log_2 \lambda \approx +3.5$

- If the laser passes through C_{ij}.

$$\lambda(r > D \mid occ(i,j)) = \frac{p(r > D \mid occ(i,j))}{p(r > D \mid \neg occ(i,j))} \approx \frac{.45}{.90} = .5$$

  – so $\log_2 \lambda \approx -1.0$

Mapping One Laser Scan



Future Attraction: SLAM

- To build an accurate map, we assume that robot pose (x,y,θ) is known accurately.
  - This is usually not true.
- *Localization* means using sensor input to estimate the robot pose (x,y,θ).
- Simultaneous Localization and Mapping (SLAM) uses the existing map and current sensor input for localization.
  - Once localized, use sensors to update the map

Mapping Assignments (4 and 5)

- We will give you laser range-sensor traces.
  - Few specular reflections; no spreading cone.
  - Off-line computation; no physical control.
- For **Assignment 4**, you will have accurate pose information (x,y,θ).
  - You build an accurate occupancy grid map.
- For **Assignment 5**, you will do simultaneous localization and mapping (SLAM).
  - You are given laser and odometry sensor values.

Implementation Hints

- Use 10×10 cm^2 grid cells.
  - But make cell size a parameter and try others.
- To display the grid:
  - Black means occupied
  - White means free
  - Grey means unknown
- Experiment with different shade mappings.
  - Make it both useful and attractive.

Implementation Hints

- Robot pose (x,y,θ) and laser endpoints (p,q) are high-resolution values.
  - Grid cells correspond to extended regions.
- Put cell centers at integer coordinates so *rounding* quickly gives cell coordinates.
- Increment C_{ij} for endpoint of laser beam.
- Step regularly along free part of the beam, decrementing C_{ij} for free cells.