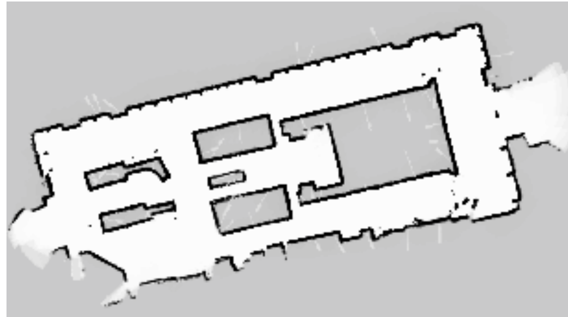
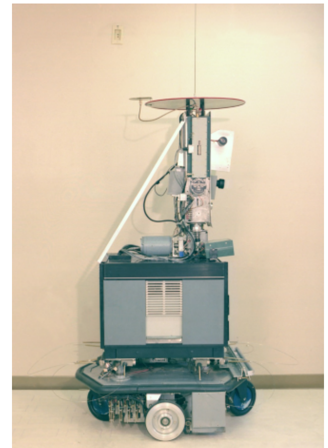
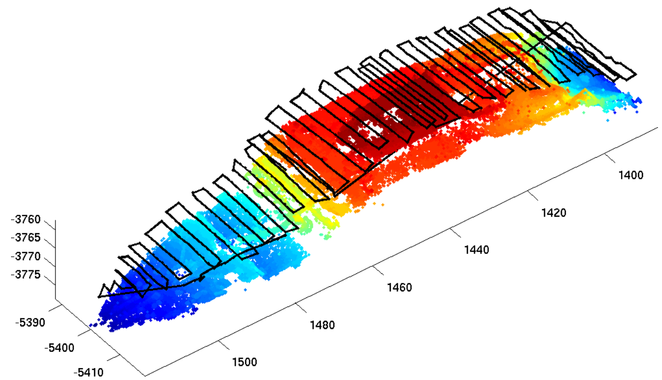
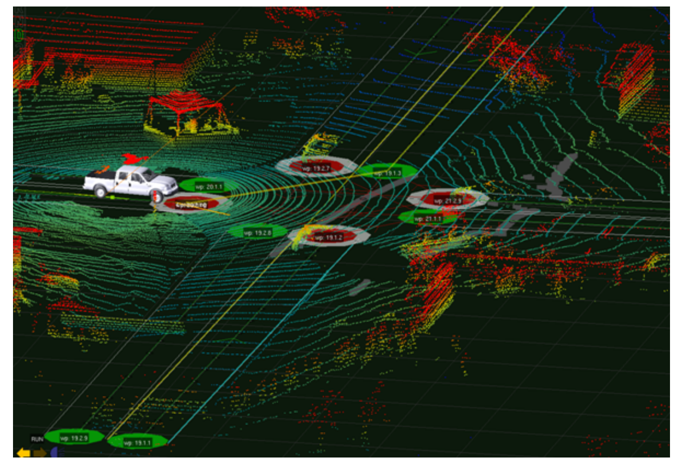








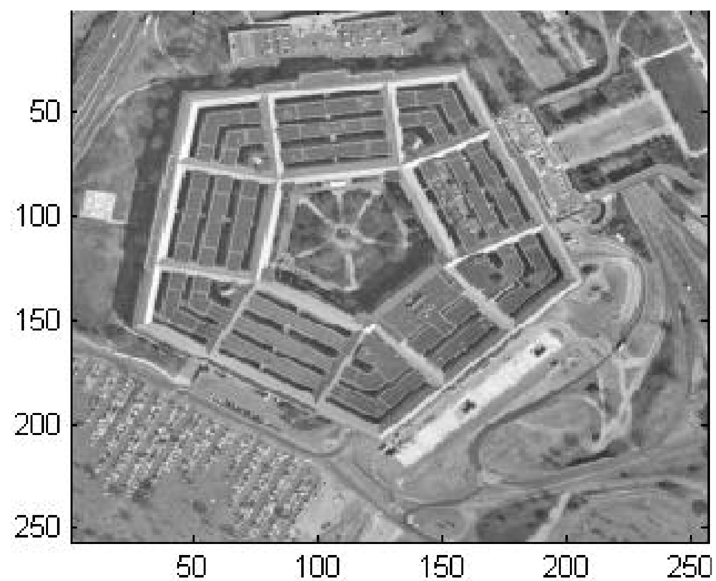
EECS568 Mobile Robotics: Methods and Principles

Prof. Edwin Olson

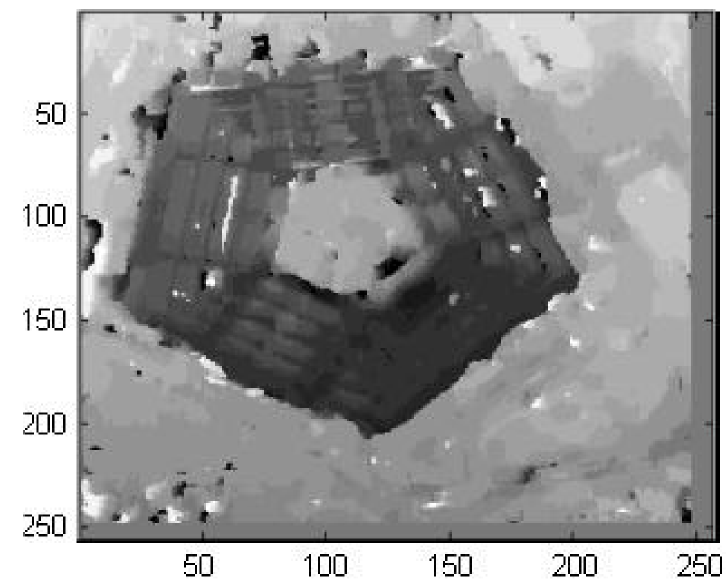
L22. A trip through the sensor zoo

Stereo

- Classic approach to stereo vision: matching pixel patches between left and right.



Left Image



Depth Map (8x8)

- Shortcoming: in low-detail areas, results are erratic. (How would we enforce local consistency?)

Block Matching

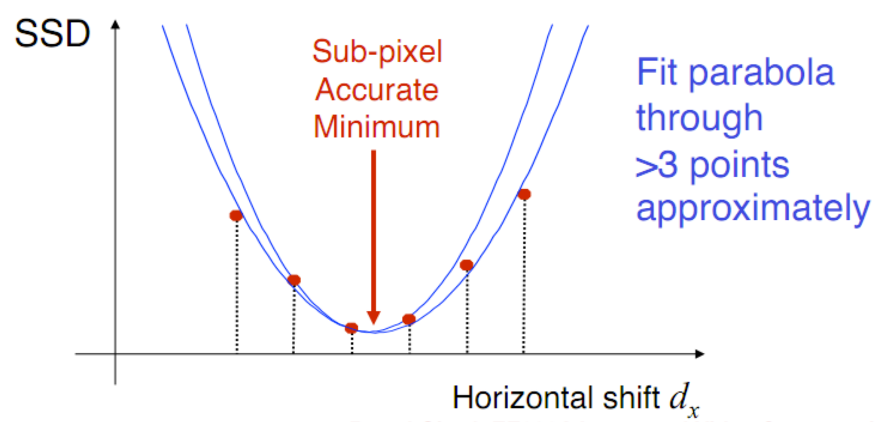
- Exploit epipolar geometry
  - A pixel in the left camera corresponds to a ray.
  - The image of a ray (in the right camera) is a line
  - Thus, if we know the geometry of the cameras, we only need to search for matches along a line.

$$\sum_{(i,j)\in W} |I_1(i,j) - I_2(x+i, y+j)|$$

$$\sum_{(i,j)\in W} \left(I_1(i,j) - I_2(x+i, y+j)\right)^2$$

- Matching procedure
  - Block size (5x5, 7x7, ...)
  - Comparison (SAD, SSE)

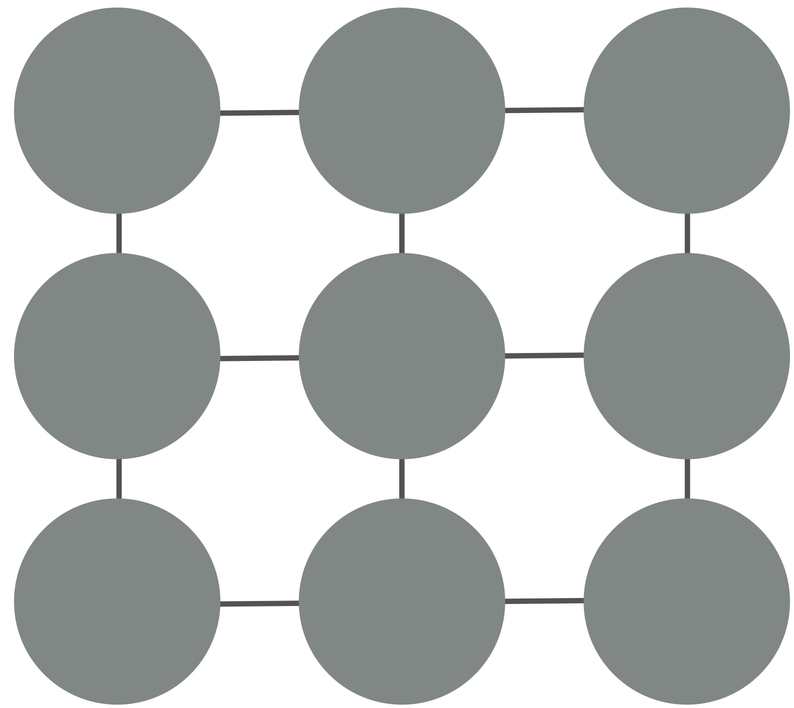
- Sub-pixel matching
  - Fractional translation of reference image
  - Polynomial interpolation of full-pixel data



Bernd Girod: EE398A Image and Video Compression

Stereo Vision: Graphical Model

- Label each pixel with a disparity
  - Maximize agreement between adjacent pixels (“discontinuity cost”)
  - Maximize agreement between left and right pixel (“data cost”)



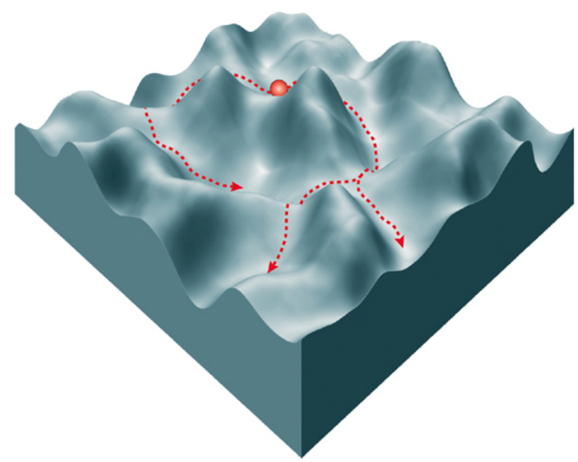
Stereo MRFs

- Could approach as a least-squares problem
  - State: disparity at each node (relax to continuous values)
  - Optimize product of function potentials (or equiv. sum of log of potentials... "log likelihood")

- Very difficult local minimum
  - Least-squares solves a local quadratic problem. If you're not in the right basin, you won't converge.
  - Least squares doesn't work well.

Iterated Conditional Modes

- Simple idea:
  - Consider a single node at a time. (i.e., fix the values of all other nodes)
  - Compute a new disparity for that node that minimizes the log likelihood
    - Only a function of the neighboring factor potentials... cheap!
    - Always reduces global error
- Not much better than least squares--- still get stuck in local minima.
- Need a method that can "look ahead", leaping out of local minima
  - Consider two nodes a=0, b=0. Cost f(a,b) has local minimum at 0,0, but global minimum at 1,1.



Nature Reviews | Molecular Cell Biology

Loopy Belief Propagation

- Each node passes messages to its neighbors:
  - “If you take on value *v*, the cost could be as low as m(v).”
  - All possible values of *v* are evaluated in a best-case sense, allowing the recipient to “teleport” to a new minimum

$$m^t_{p\to q}(f_q) = \min_{f_p}\left(V(f_p, f_q) + D(f_p) + \sum_{s\in N(p)\backslash q} m^{t-1}_{s\to p}(f_p)\right)$$

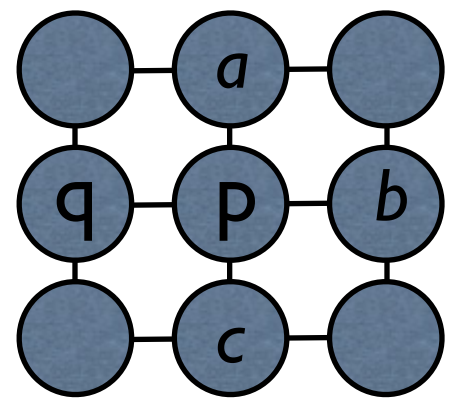
P’s message to q specifies a “cost” for each value that q might take.

For every value fq, we’ll report the cost for the best-case fp.

Cost of neighbors having different labels

Cost of assigning fp to pixel p

How much do *our* neighbors say it would cost for q to have value p



Isn't this fun?

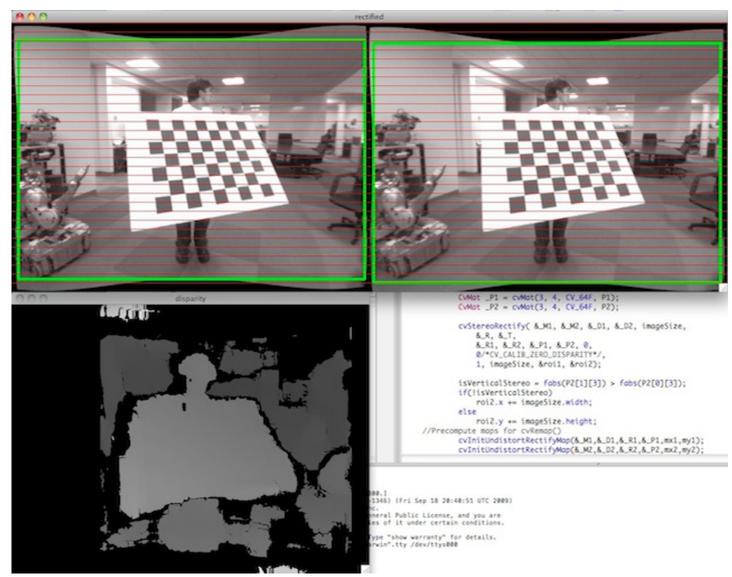
- With an almost trivial model, we can destroy block matching problems.
  - You can be competitive with Middlebury top 100 in a couple days' effort!

SSD+min-filter [scharstein szeliski], rank = 90

LBP [olson], rank* = 60

The disappointment

- MRF approaches are too slow for robots
  - #1. [Wang/Zheng]: 20s
  - #2. [Yang/Nister]: 62s



- Block matching is fast!
  - (unranked) [Konolige], 10ms

Why is LBP slow?

- Short answer: because computing messages is slow

$$m^t_{p\to q}(f_q) = \min_{f_p}\left(V(f_p, f_q) + D(f_p) + \sum_{s\in N(p)\backslash q} m^{t-1}_{s\to p}(f_p)\right)$$

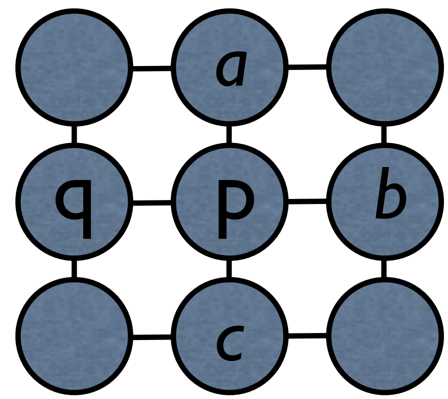
P's message to q specifies a "cost" for each value that q might take.

For every value fq, we'll report the cost for the best-case fp.

Cost of neighbors having different labels

Cost of assigning fp to pixel p

How much do *our* neighbors say it would cost for q to have value p

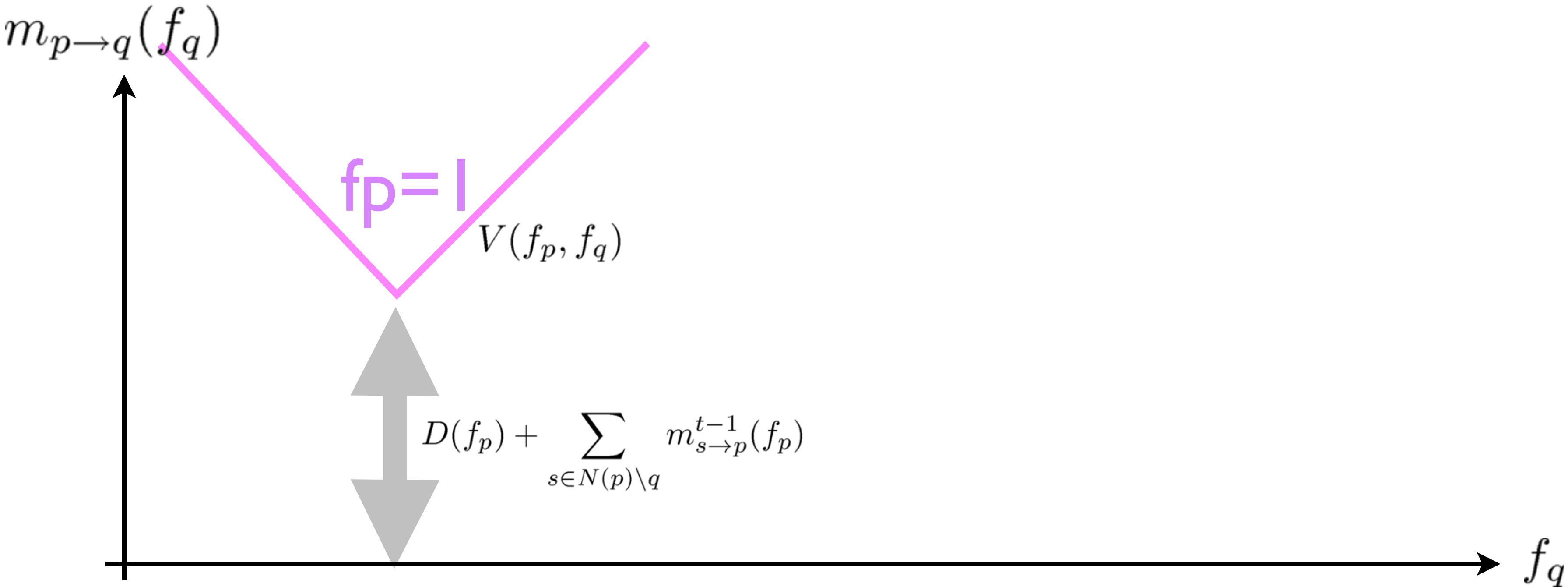



```
for x=1:width
    for y=1:height
        for n=1:neighbors
            for fq=1:labels
                for fp=1:labels
                    ...
```

Cool trick #1: Min Convolution

[Felzenswalb/Huttenlocher 2004]

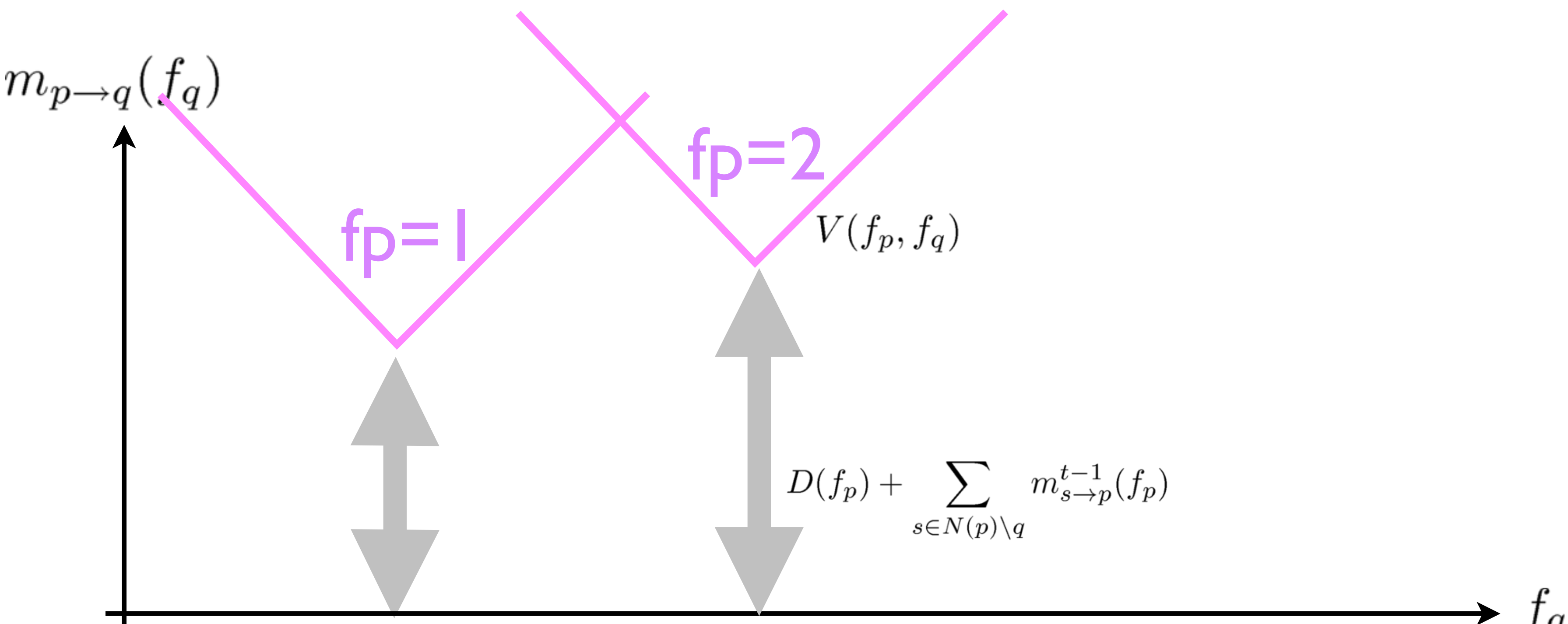
$$m^t_{p\to q}(f_q) = \min_{f_p}\left(V(f_p, f_q) + D(f_p) + \sum_{s\in N(p)\backslash q} m^{t-1}_{s\to p}(f_p)\right)$$



Cool trick #1: Min Convolution

[Felzenswalb/Huttenlocher 2004]

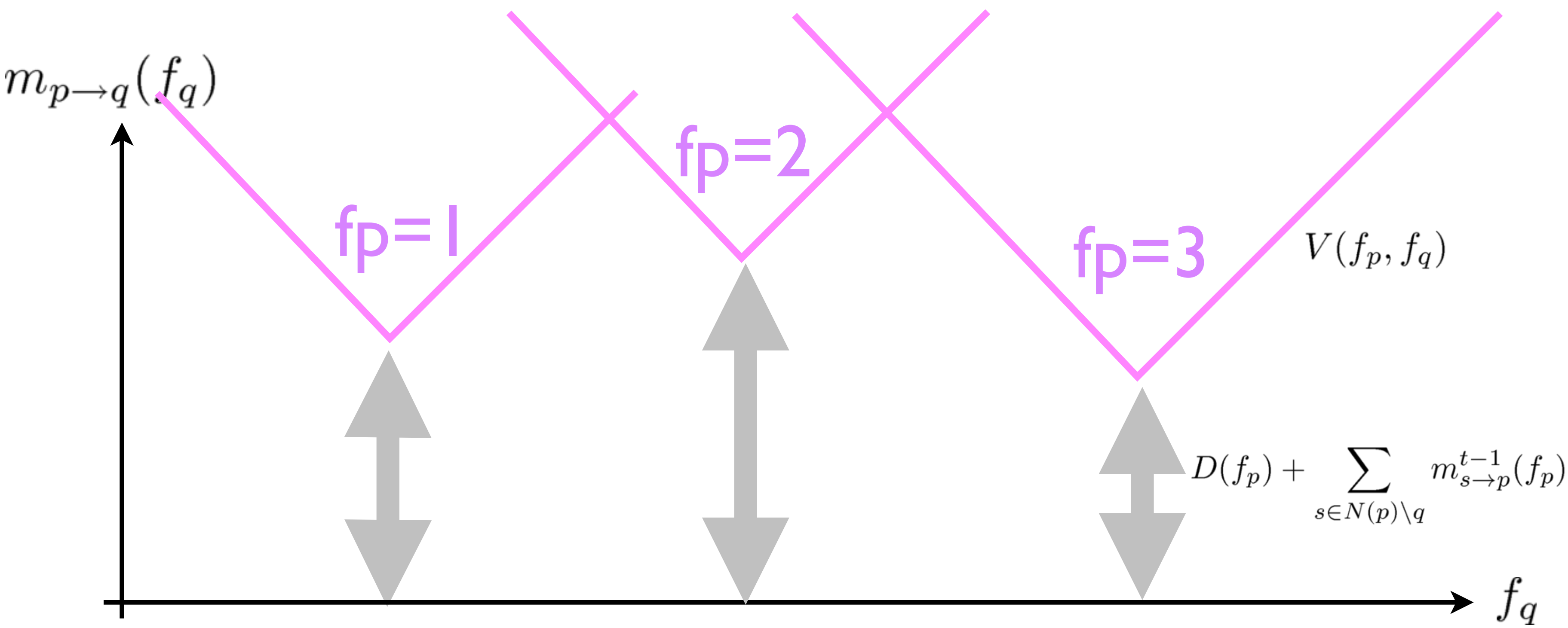
$$m^t_{p\to q}(f_q) = \min_{f_p}\left(V(f_p, f_q) + D(f_p) + \sum_{s\in N(p)\backslash q} m^{t-1}_{s\to p}(f_p)\right)$$



Cool trick #1: Min Convolution

[Felzenswalb/Huttenlocher 2004]

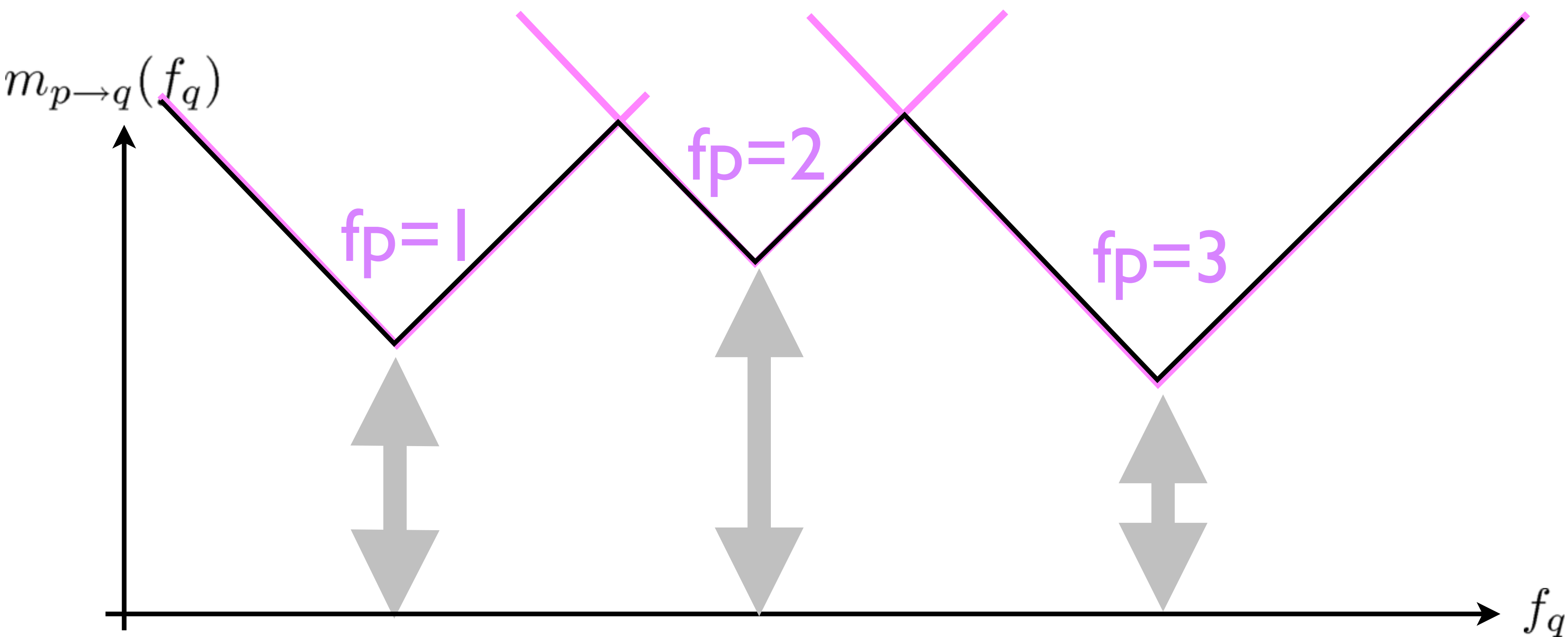
$$m^t_{p\to q}(f_q) = \min_{f_p}\left(V(f_p, f_q) + D(f_p) + \sum_{s\in N(p)\backslash q} m^{t-1}_{s\to p}(f_p) \right)$$



Cool trick #1: Min Convolution

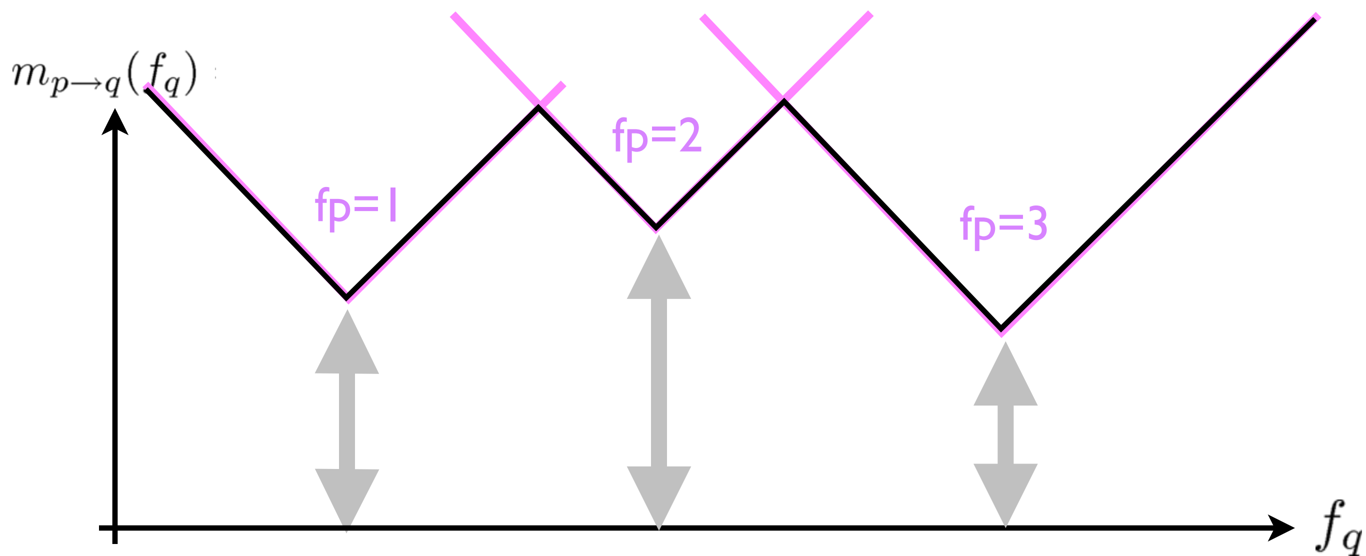
[Felzenswalb/Huttenlocher 2004]

$$m^t_{p\to q}(f_q) = \min_{f_p}\left(V(f_p, f_q) + D(f_p) + \sum_{s\in N(p)\backslash q} m^{t-1}_{s\to p}(f_p)\right)$$

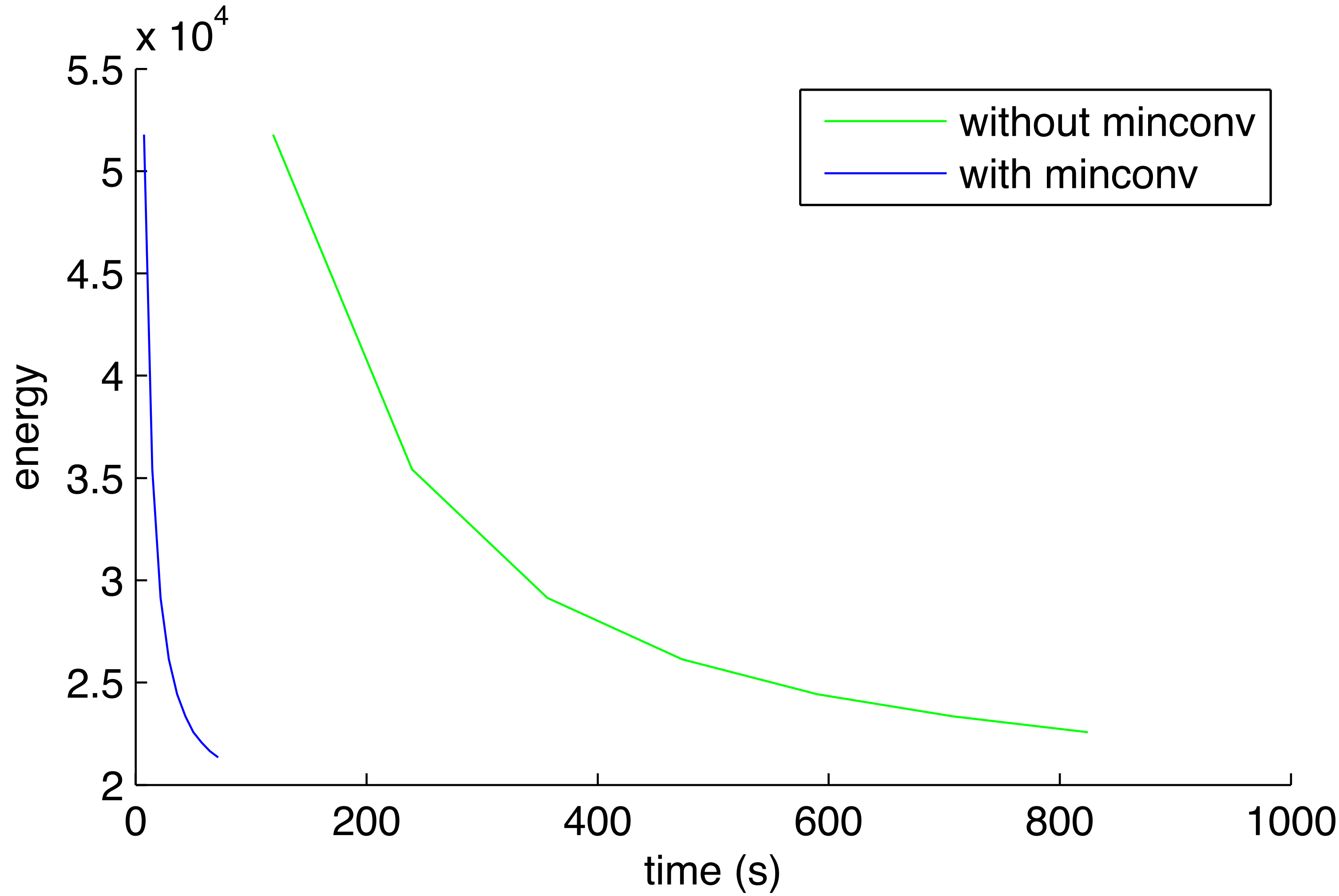


Cool trick #1: Min Convolution

- This is a min-convolution operation
  - Naive implementation is O(k^2)
- Efficient algorithms exist for special cases!
  - In linear case, forward-backwards algorithm O(k)
  - Quadratic case also has a method... a bit messier, but still O(k)
- Exact!



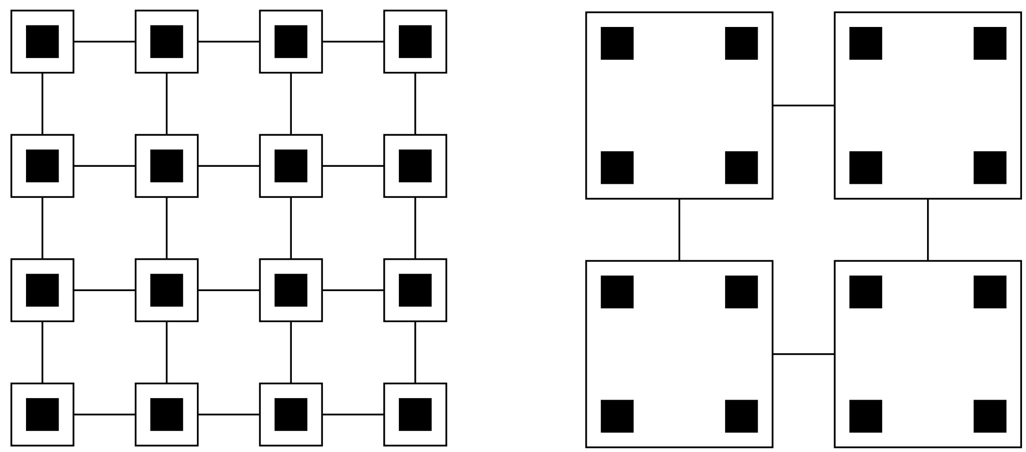
Performance



tsukuba (384 x 288), 16x subpixel, nlabels=256

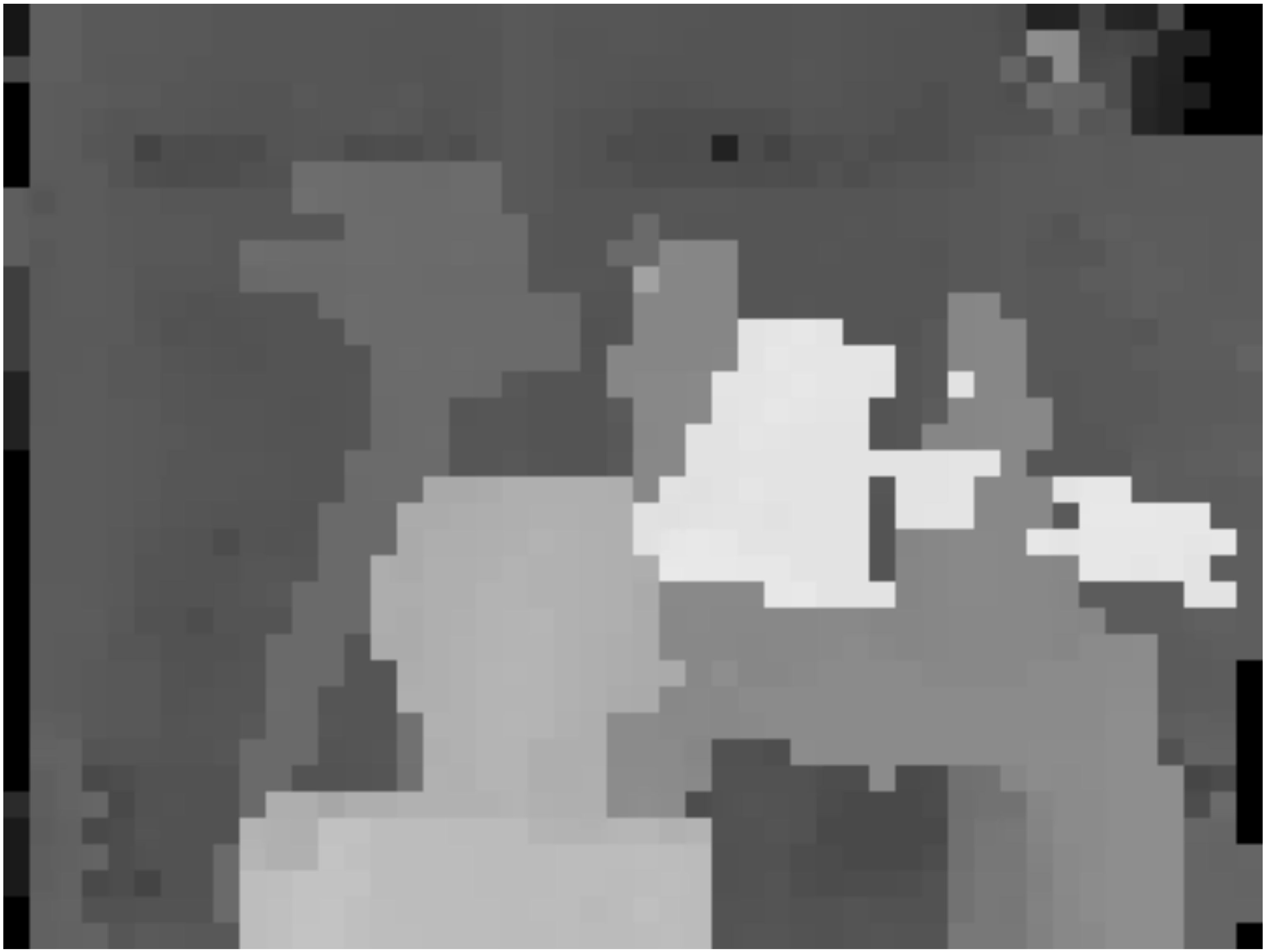
Cool Trick #2: Multi-Grid

[Felzenszwalb/Huttenlocher 2004]



- Advantages:
  - Information spreads rapidly around graph
- Disadvantages:
  - Have to come up with function potentials for other levels of the image pyramid
  - Can lead to artifacts due to the arbitrary alignment of the grid cells

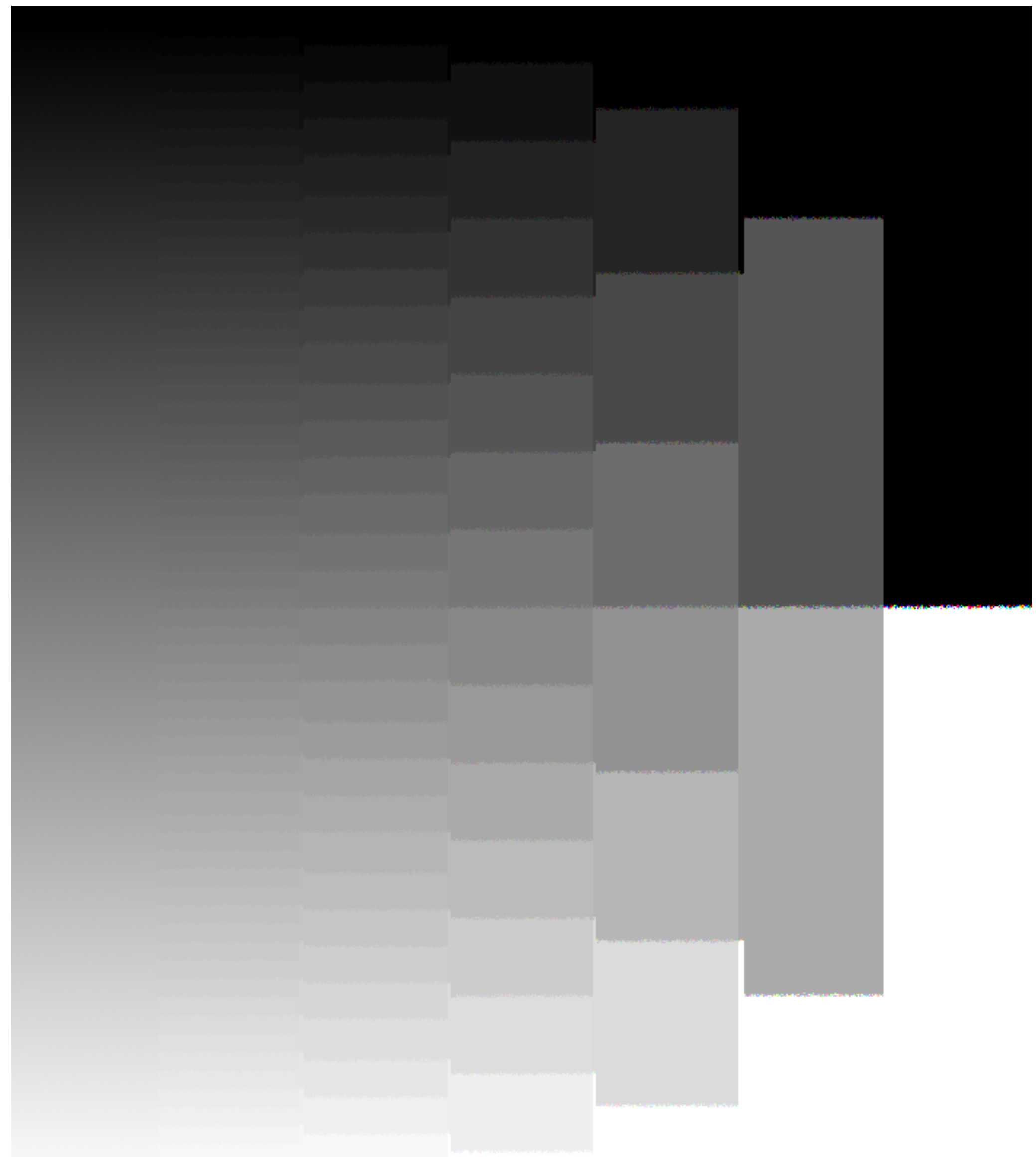
Multi-resolution LBP



Cool Trick #3: Quantized labels

[Strom, Olson 2010*]

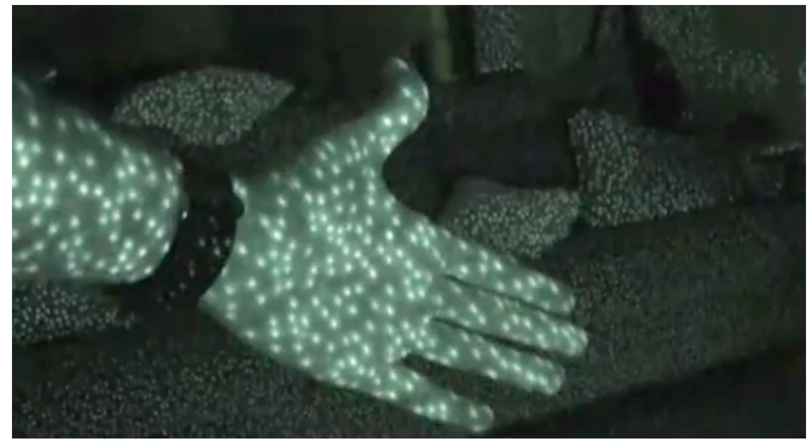
- Idea: Start iterating with fewer labels, slowly increase number of labels
- Advantages:
  - Information spreads rapidly around graph
  - No spatial blocking artifacts
- Disadvantage:
  - Not as fast as multi-grid



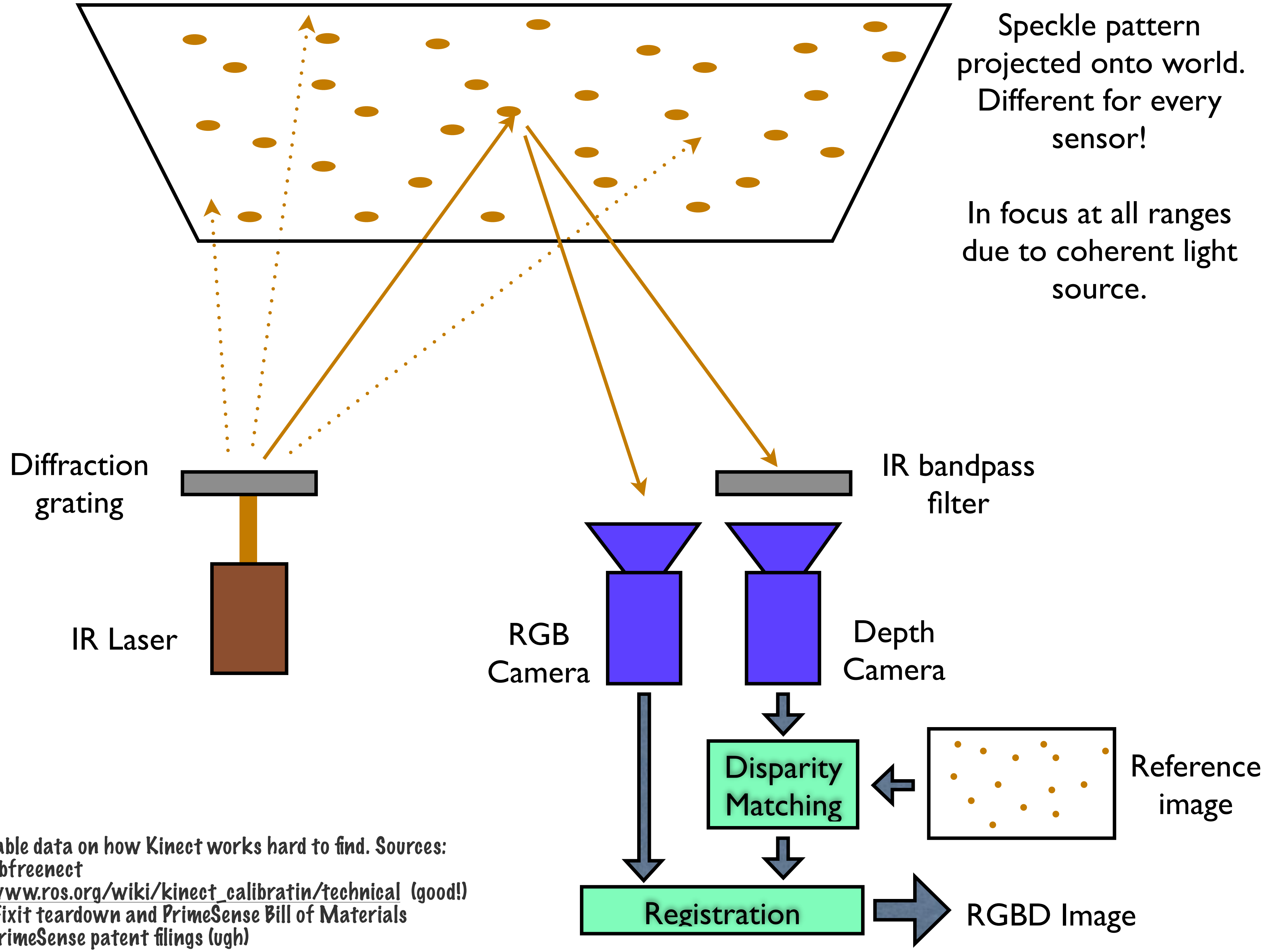
Quantized LBP

PrimeSense/Kinect

- Similar to a stereo camera in concept
  - But replace one camera with a *projector*
  - Second camera detects projected camera.
- Why is this a good idea?
  - It works even when the environment is devoid of distinguishing features (e.g. white walls)
  - Under favorable conditions, very good results
- What are the shortcomings?
  - Brightness of projector limits effectiveness at long ranges and outdoors
  - Power consumption / stealth



graphics.stanford.edu



reliable data on how Kinect works hard to find. Sources:
-- libfreenect
-- www.ros.org/wiki/kinect_calibratin/technical (good!)
-- iFixit teardown and PrimeSense Bill of Materials
-- PrimeSense patent filings (ugh)

Kinect Particulars

- Produces 640x480 RGBD Image
  - IR Camera is 1280x1024 @ 15Hz
    - Uses 2x2 binning to increase sensitivity and frame rate to 30Hz
    - Monochrome... 16 bit?
- Matching
  - Calibration image stored in device at factory
  - Repeatedly "streamed" in sync with acquired IR image, fed into matching engine
  - Block based matching
    - 9x9 blocks
    - 1/8 pixel interpolation
    - 64 (?) pixel search range   (Kinect returns 11 bit range values)
- Registration
  - Corrects for parallax of RGB and depth sensor. (Could be eliminated by using a single sensor with both RGB and IR pixels in an RGBI "Bayer" pattern).

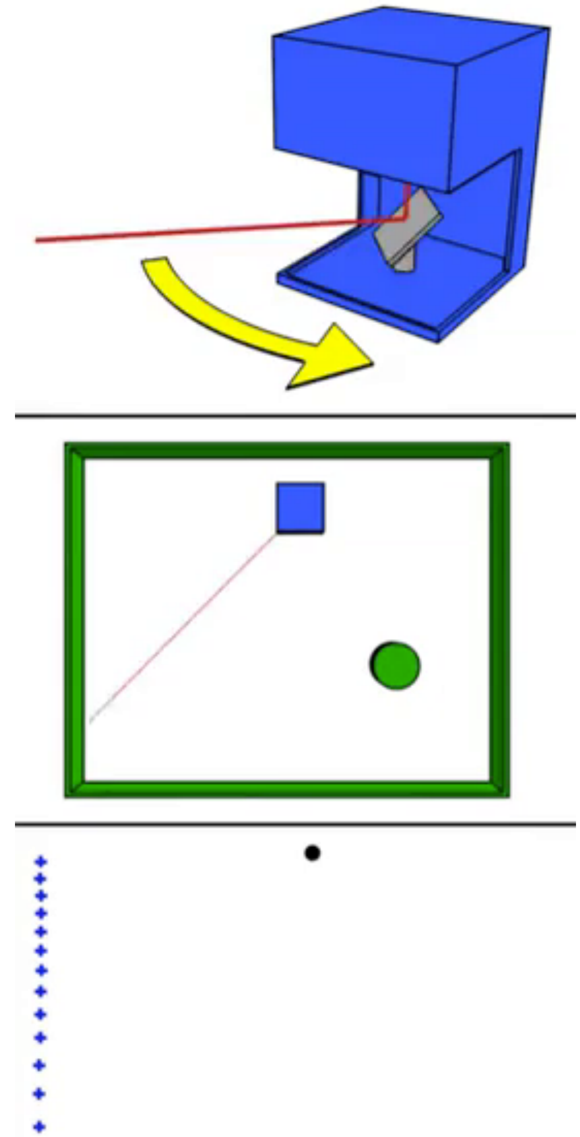
Laser Range Finders

- SICK

- Hokuyo

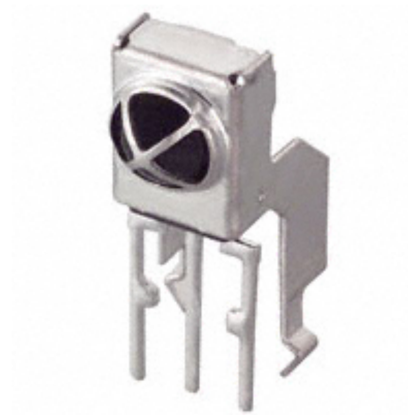
- Velodyne

- Ibeo

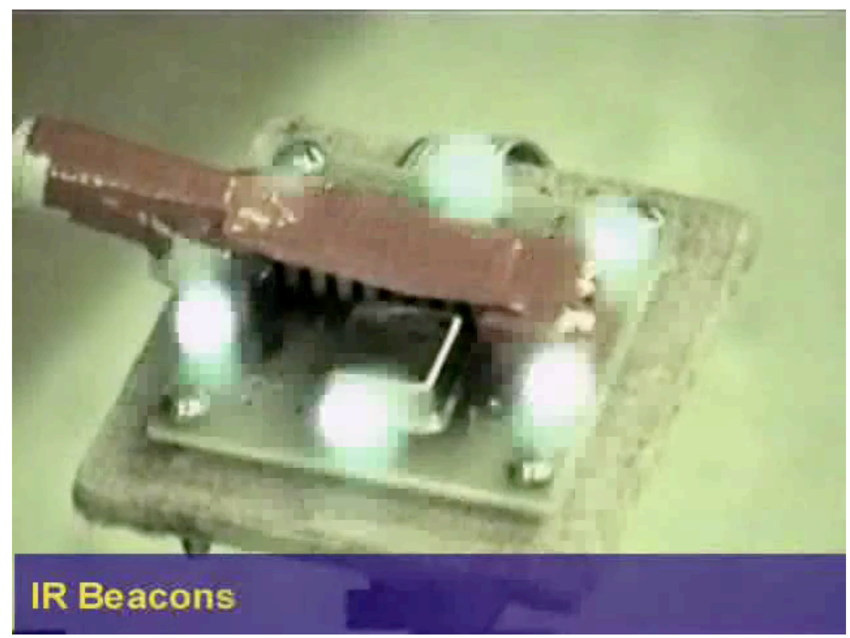


IR Beacons

- LEDs and photo diodes
  - Very cheap
  - Communications
  - (Crude) distance/proximity



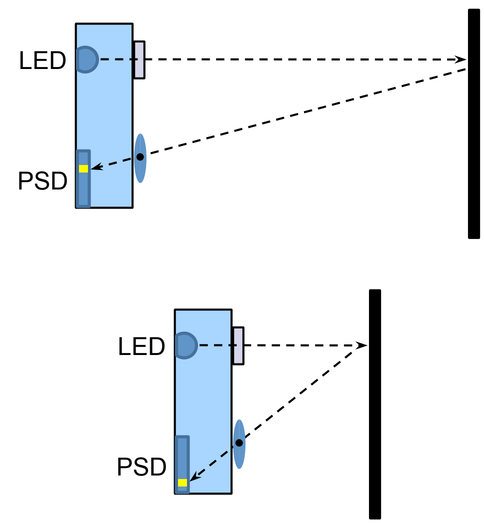
- Remote control demodulators
  - 40kHz, quite robust!
  - Integrated signal conditioning, amplification, demodulation
  - Comms (~5kbps)
  - Proximity
  - Bearing (with baffle)
  - $0.63 each

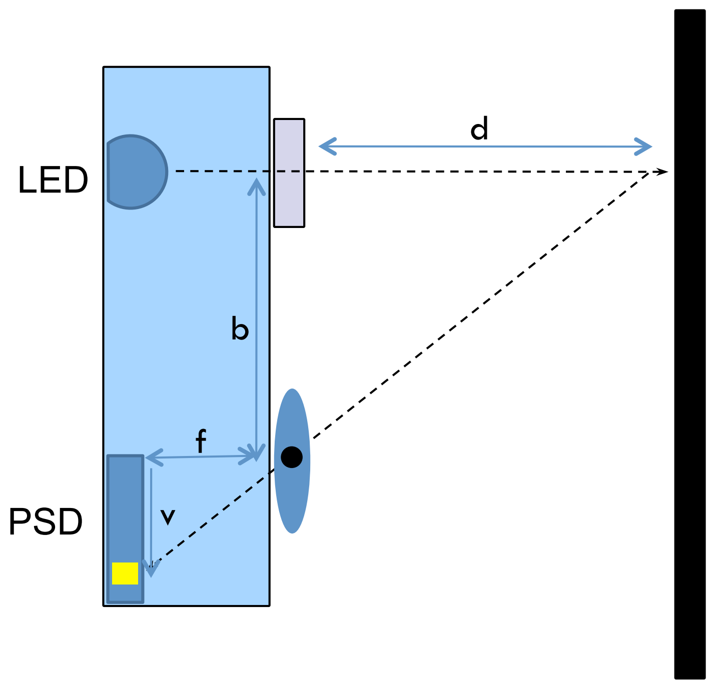


IR Beacons

IR Range Finders

- Range measurement
  - $8





f, b: Properties of device

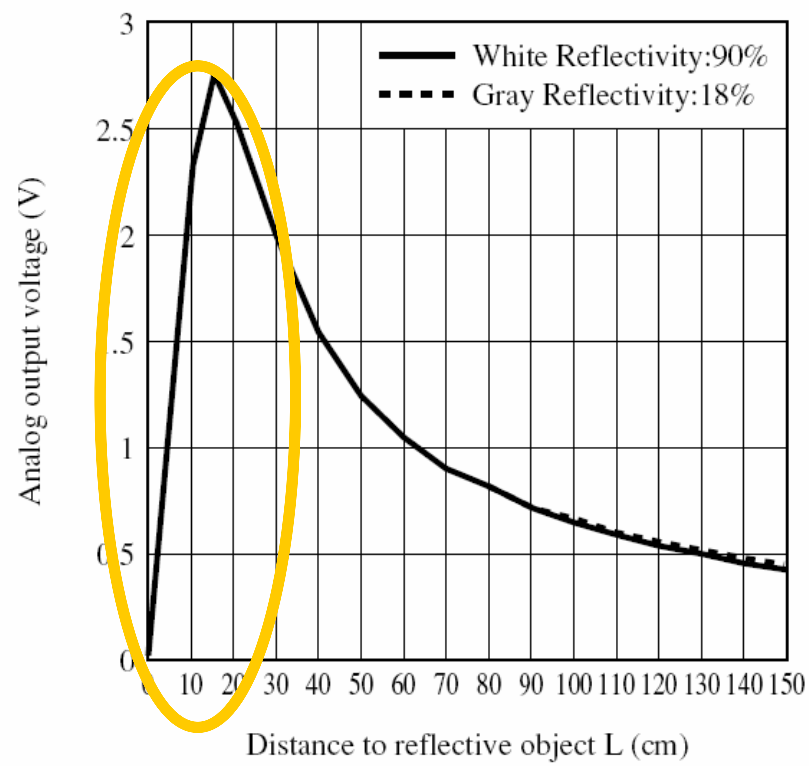
d: quantity (distance) we want to know

v: voltage (proportional to position) that we observe

IR Range Finders

- Add in a few parameters to fit non-idealities of device, we get the observation model:

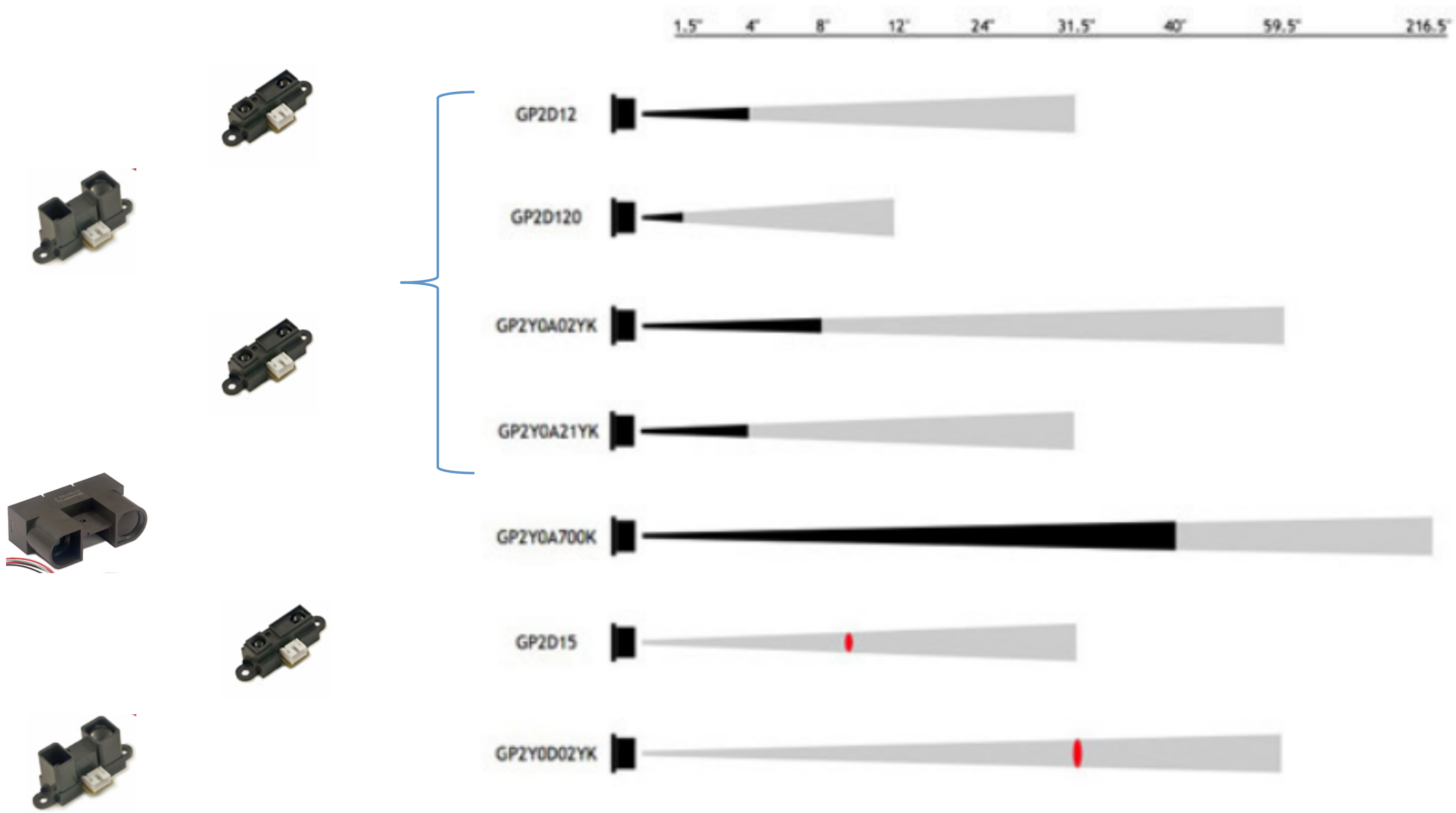
$$V = \frac{K_m}{d + K_d} + K_b$$



- Covariance model?

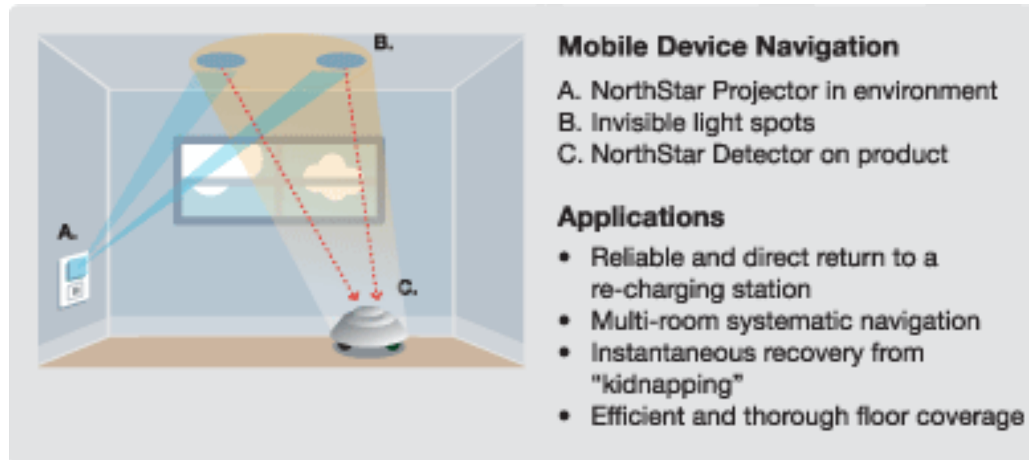
IR Range Finders

IR Range Finders



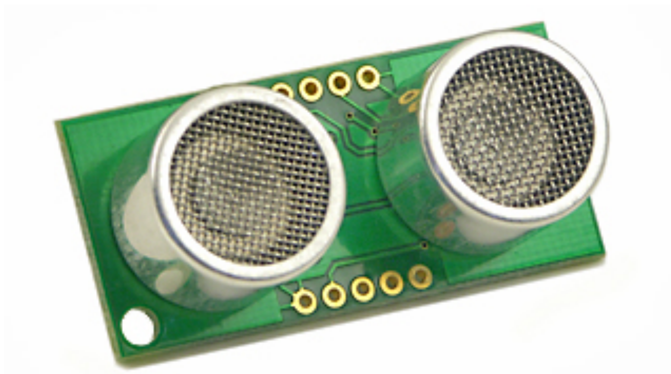
Northstar System

- Evolution Robotics
  - Indoor “GPS”

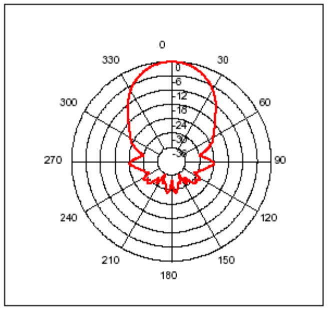


- 10Hz update rate
  - Up to 20 “spots”

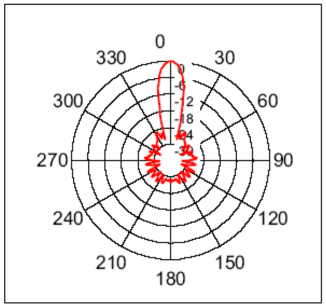
Ultrasound



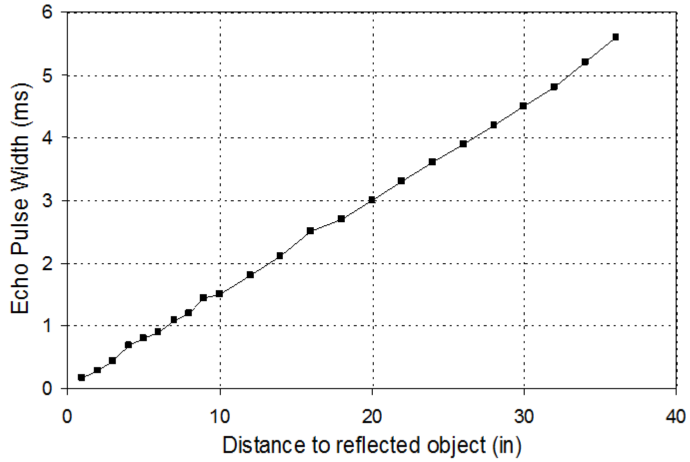
- Time-of-flight of sound
  - Linear response
  - Fairly accurate.
- Wide lobes (and side lobes!)
  - Can't see details
  - Can be either good or bad... why?



40khz ($30-60), range 3m-6m

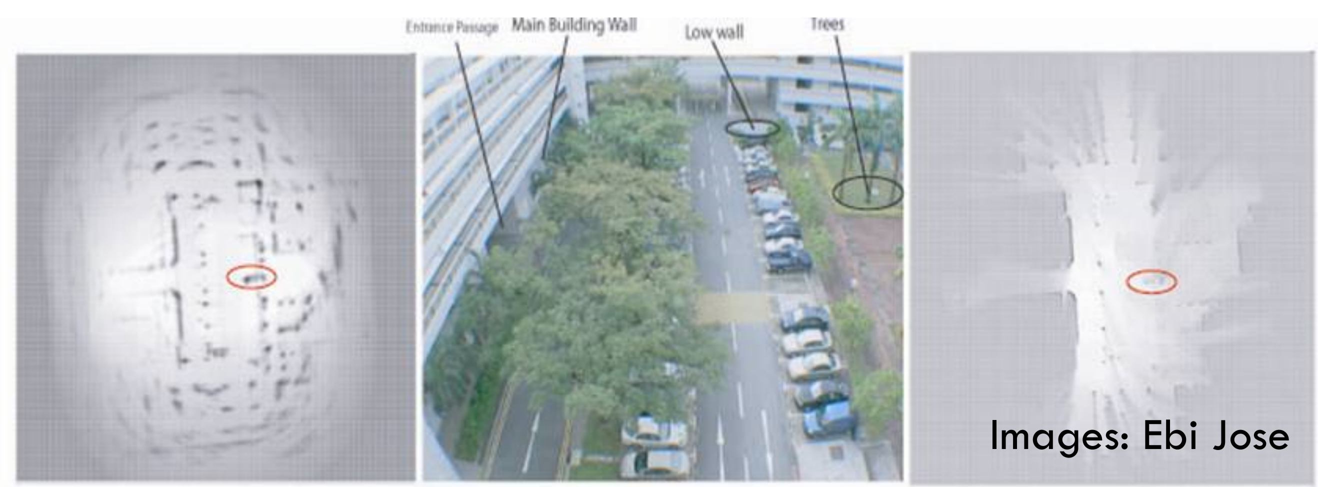


235khz ($140), range 1m



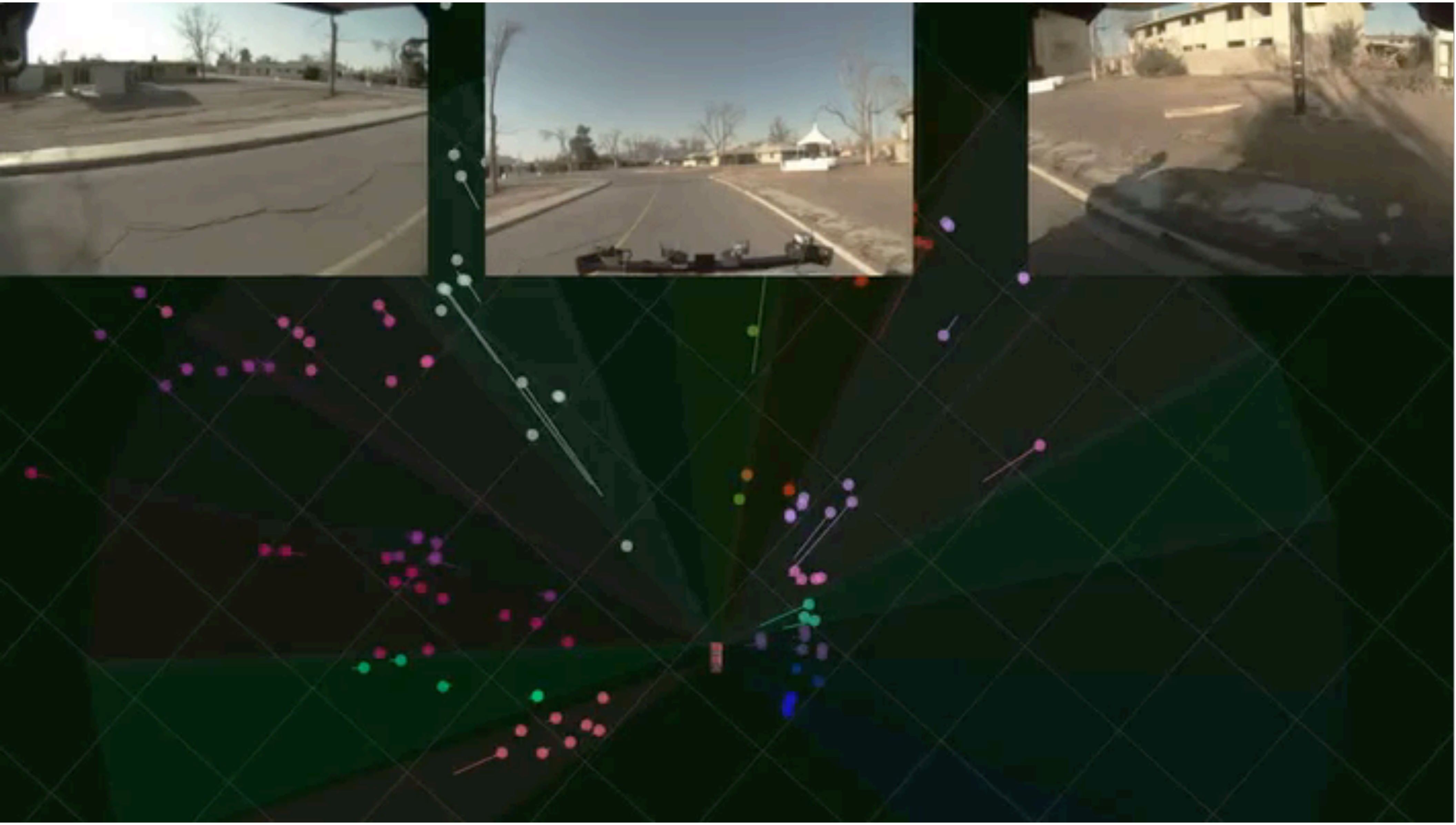
Radar

- 24GHz: Proximity detection
- 75GHz: Millimeter-wave radar
    - Automotive cruise control radars (e.g. Delphi ACC)
    - Mechanically steered
- Can get position and range rate!
    - Great help in data association



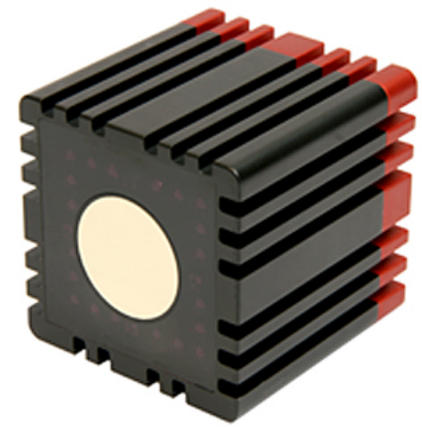
Images: Ebi Jose

Radar

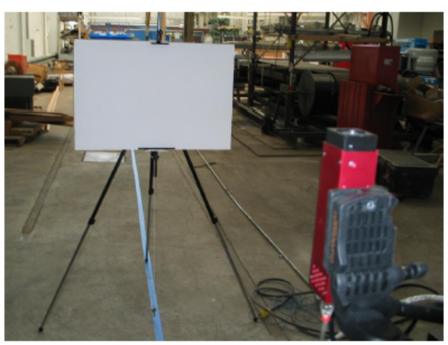


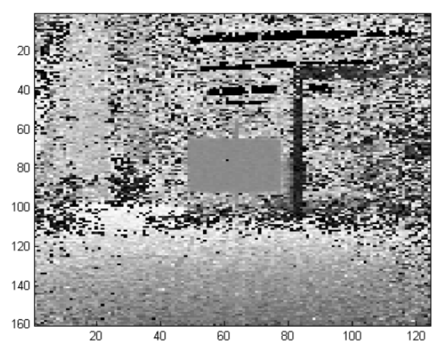
Flash LIDAR

- SwissRanger SR4000
- $9100

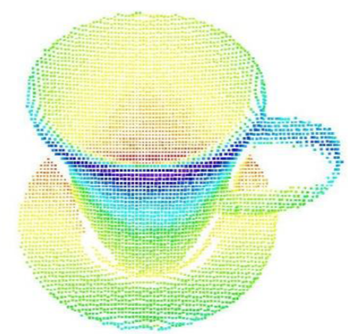


- 174x144 pixels @ (up to) 54fps





real world evaluation (CMU)

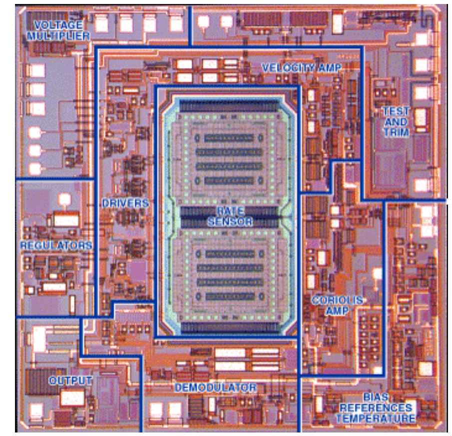


manufacturer's demo image

IMU

- Coriolis effect
  - An object in a rotating coordinate frame experiences a force proportional to its velocity
  - Idea: move a mass around a lot and see if there's a force acting on it.
  - Orientation is integral of angular motion… error accumulates.

- Fiber-optic Gyro (FOG)
  - Shoot photons in a circle. If we're rotating CCW, the CW photons will complete a circuit faster than those moving CCW.
  - Measure arrival times using interferometry.
  - More fiber = more circles = greater sensitivity. (100m - 3km of fiber optic cabling!)



MEMs gyro: degrees per minute. Hard to calibrate.

FOGs: 0.1 deg / hour (FOG200 Northrop Grumman)

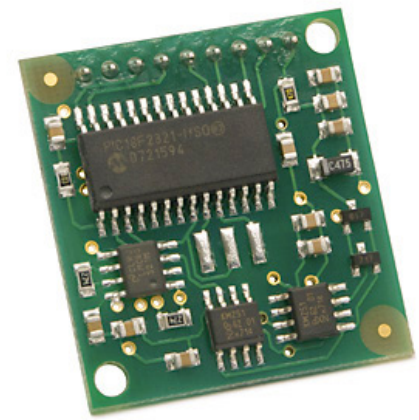
IMU Performance

- Tactical grade
  - 1 deg/h, 1 mg
- Navigation grade
  - 0.01 deg/h, 25 ug
- Strategic grade
  - (classified)

- For comparison
  - Earth rotation rate (at pole)
    - 15 deg /h
  - ITAR Limits
    - 0.5 deg/h, 50 mg (?)

Coriolis Effect

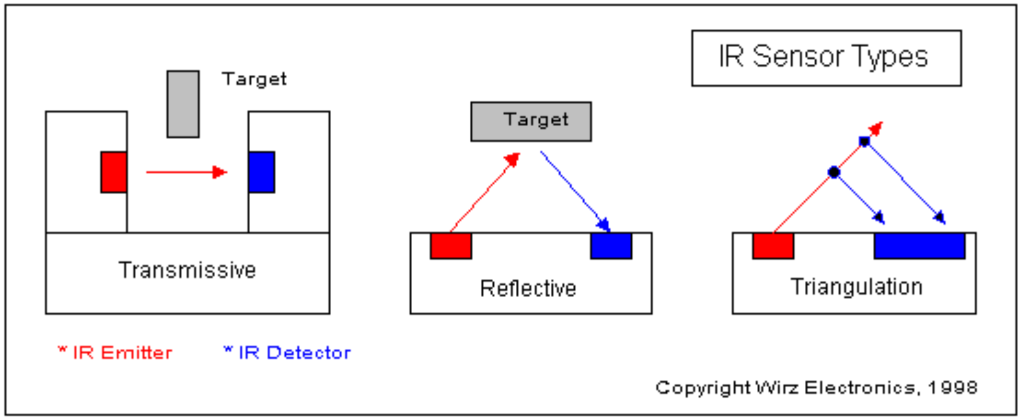
Magnetometer/Compass



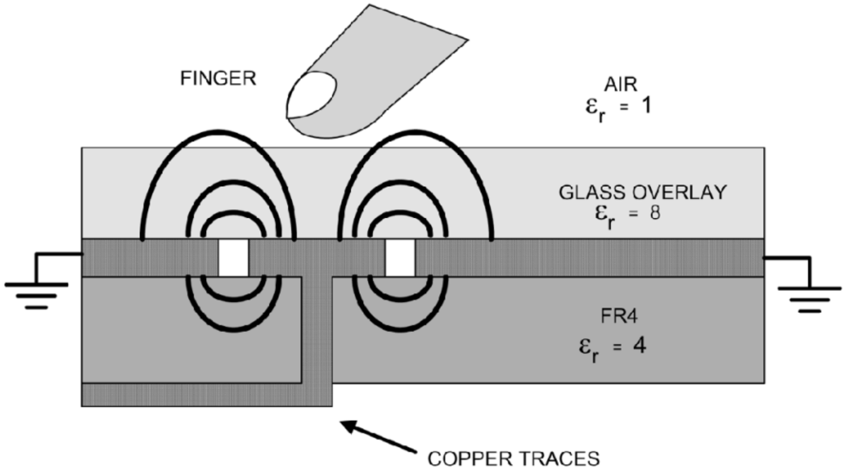
- Always a popular idea
  - Error doesn’t integrate over time.
  - Unfortunately, hard to make work reliably
- Many sources of interference
  - Robot itself
  - Buildings
  - In fact, some have built maps of environments by using the local magnetic flux as a landmark!
- Gyro compassing
  - With an accurate enough gyro, you can measure the Earth spinning beneath you (unless you’re at the equator!)

Even Cheaper Sensors

- Switches
- IR break beam

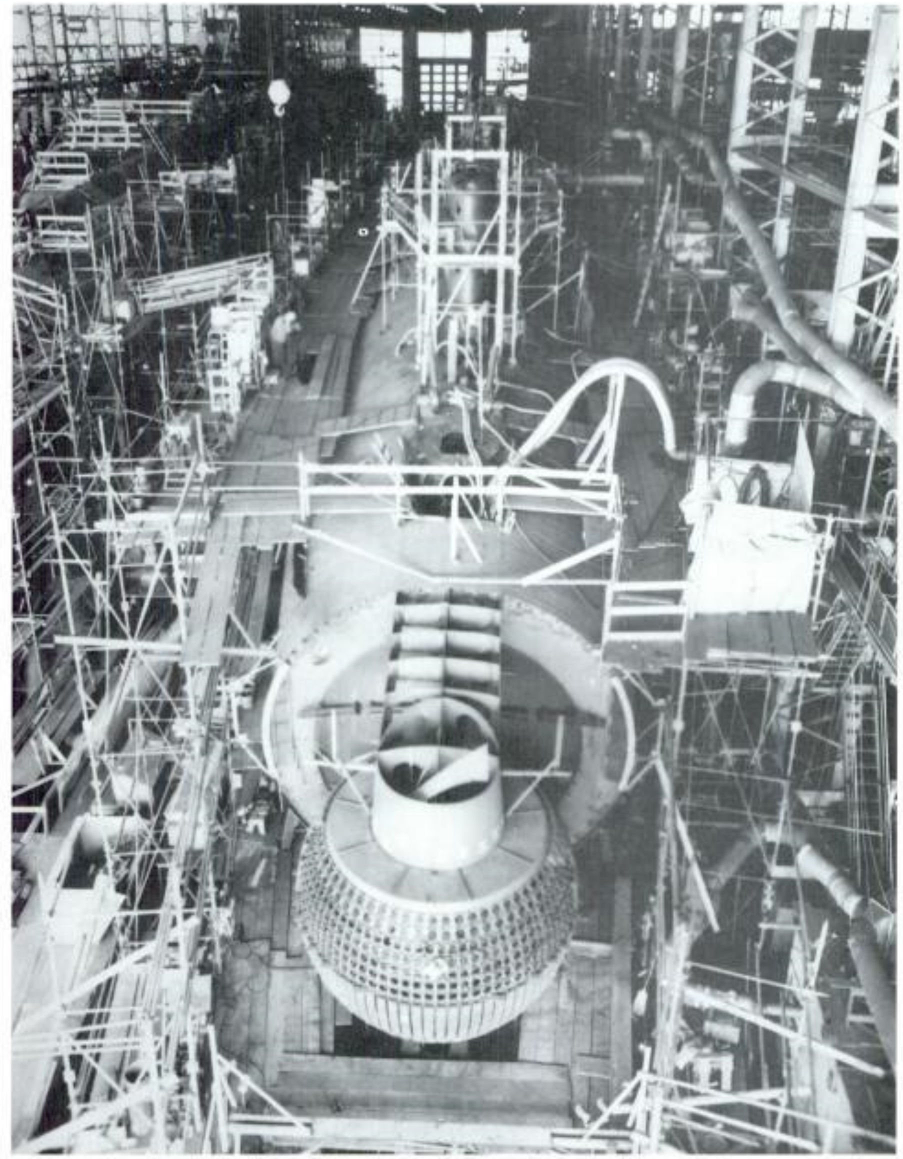


- Capacitive Field Proximity Sensors



Sonars at Sea

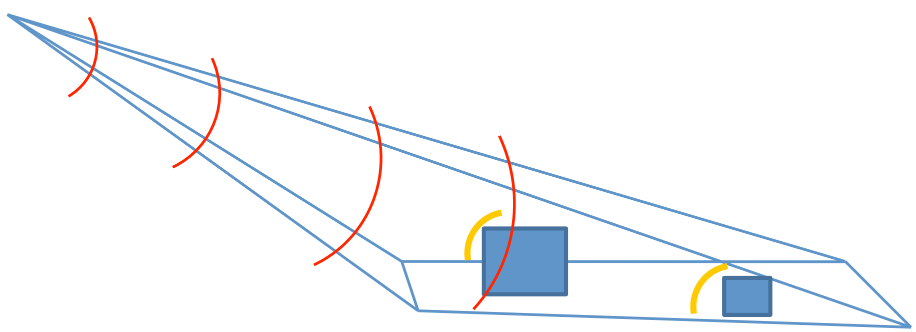
- Many specialized sensors for under water:
  - Radio doesn't propagate very far
  - Murky: optical doesn't work very well.
- Sound, on the other hand…
  - Travels much farther than in air



BQQ-5 uses a spherical bow array directly descended from the BQS-6 which equipped earlier U.S. submarines. The first BQS-6 is shown on board the attack submarine *Thresher*, building at Portsmouth Naval Shipyard, April 1960, with the lower hydrophones in place. The triple row of hydrophones wrapped around the sphere is not visible. (U.S. Navy)

DIDSON Sonar

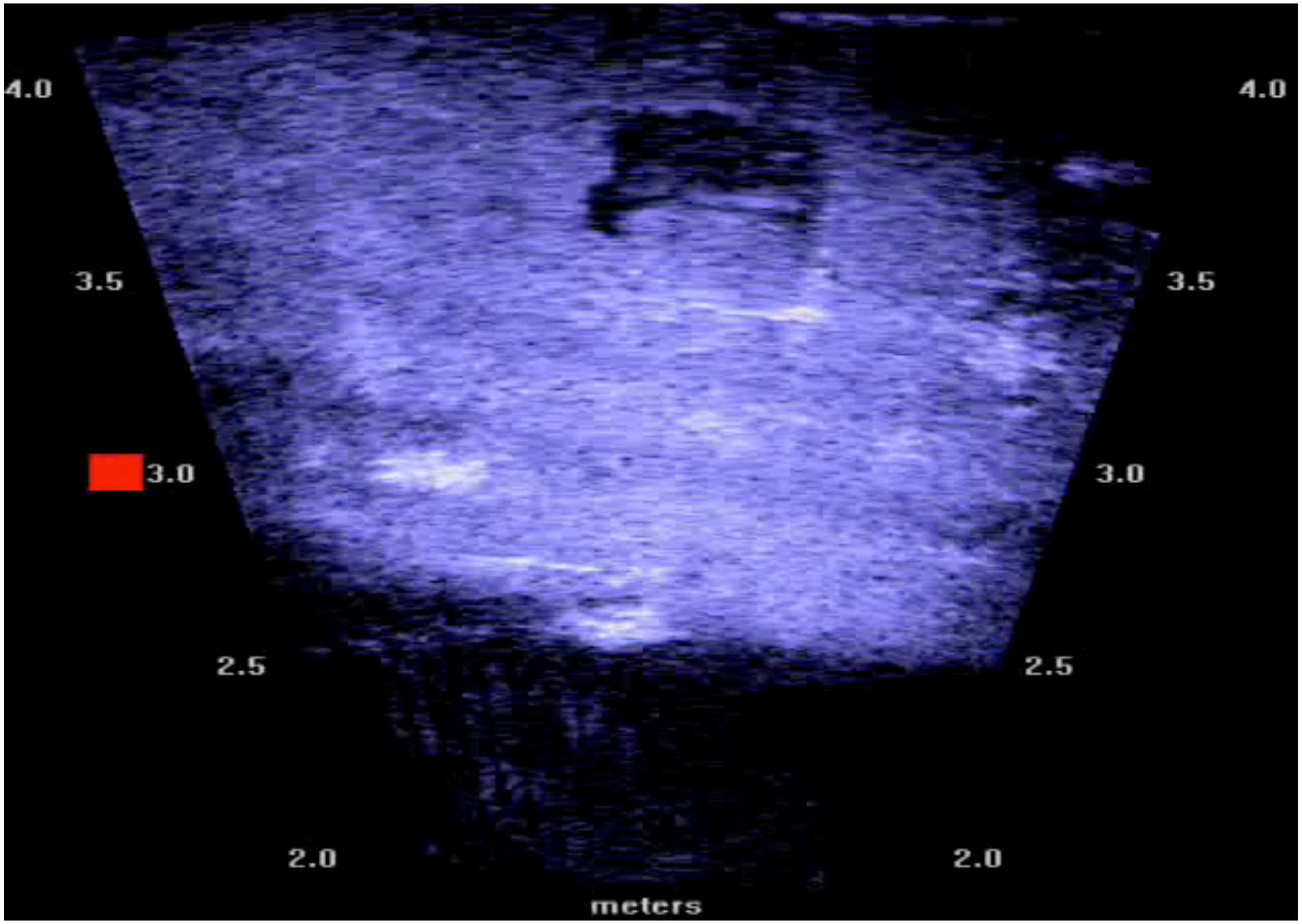
- Underwater sonar "camera"
  - 96x500* pixels
  - Essentially 96 multi-echo, narrow beam sonars that fire together
  - Plot return intensity vs. time for each sonar
- Restricted to shallow grazing angles



One sonar element: narrow horizontal FOV, wide vertical FOV.

DIDSON internals, showing acoustic lenses

DIDSON Sonar



Manta mine: shallow water anti-landing mine. Uses acoustic and magnetic triggering mechanism.

Doppler Velocity Log (DVL)

- Constantly measure Doppler shift of "pings" as they reflect off the ground beneath the AUV.

- Can measure velocity of vehicle in x, y, z.

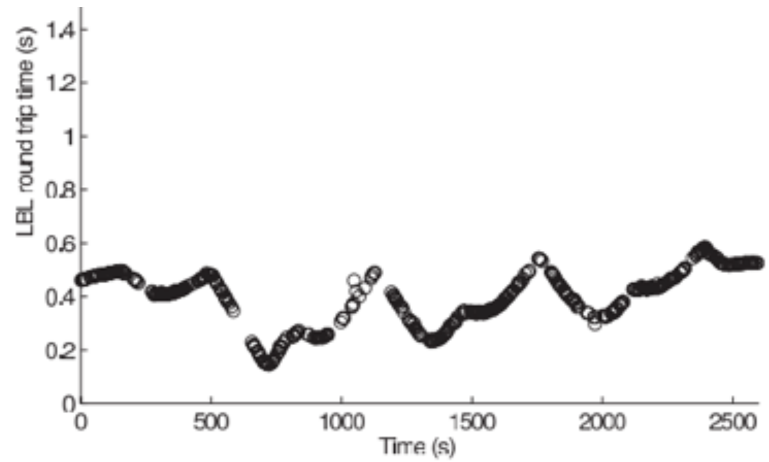
Long Baseline (LBL)

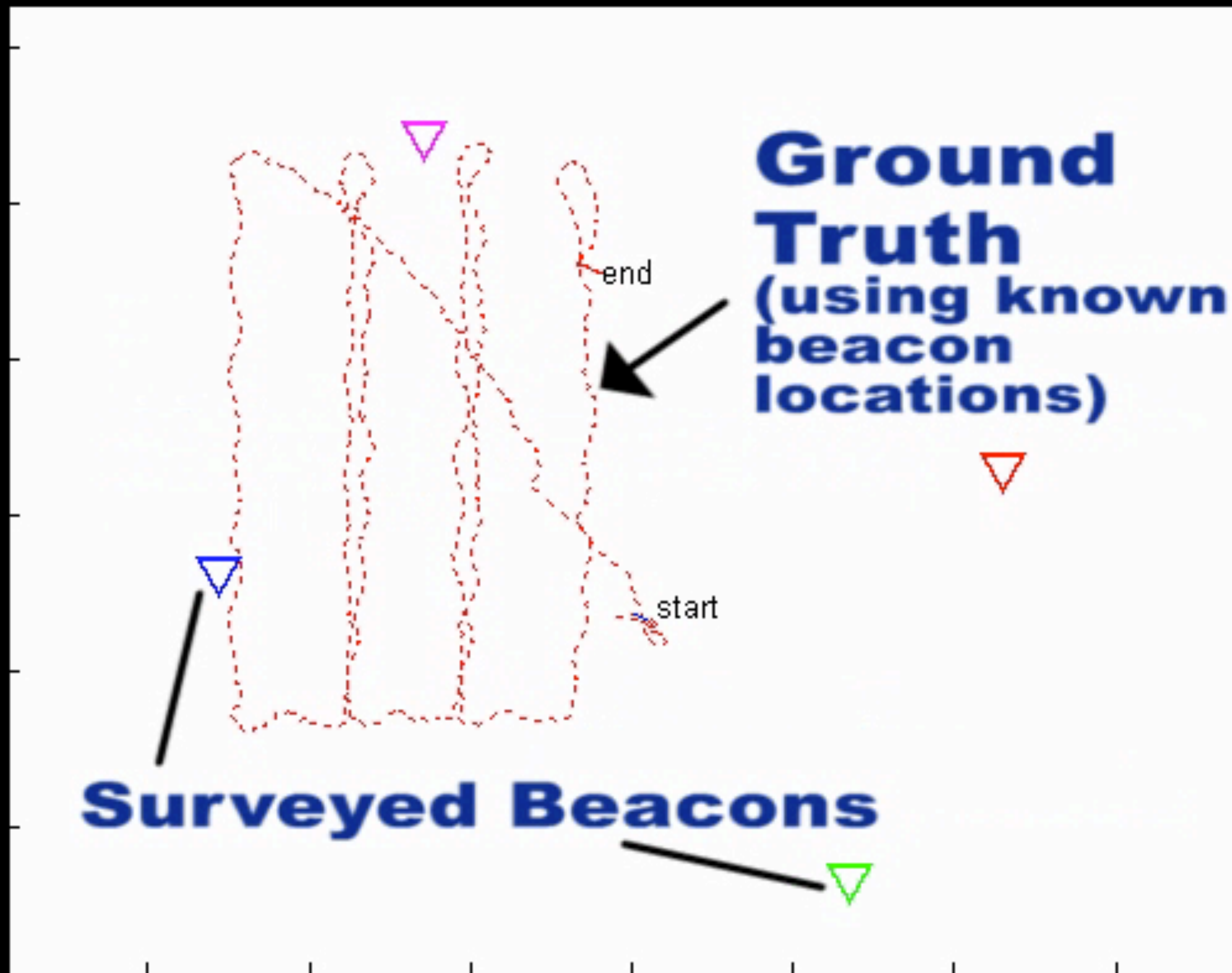
- Deploy beacons whose position is known in advance

- Robot periodically "pings" beacons
  - Beacons respond immediately
  - Robot measures RTT range measurement

- Ranges can extend to kilometers

- How many beacons?
  - One beacon: robot is on a sphere
  - Two beacons: robot is on a circle
  - Three beacons: robot is on a point
  - Assume known depth: one less beacon.





Ground Truth (using known beacon locations)
end
start
Surveyed Beacons

The end