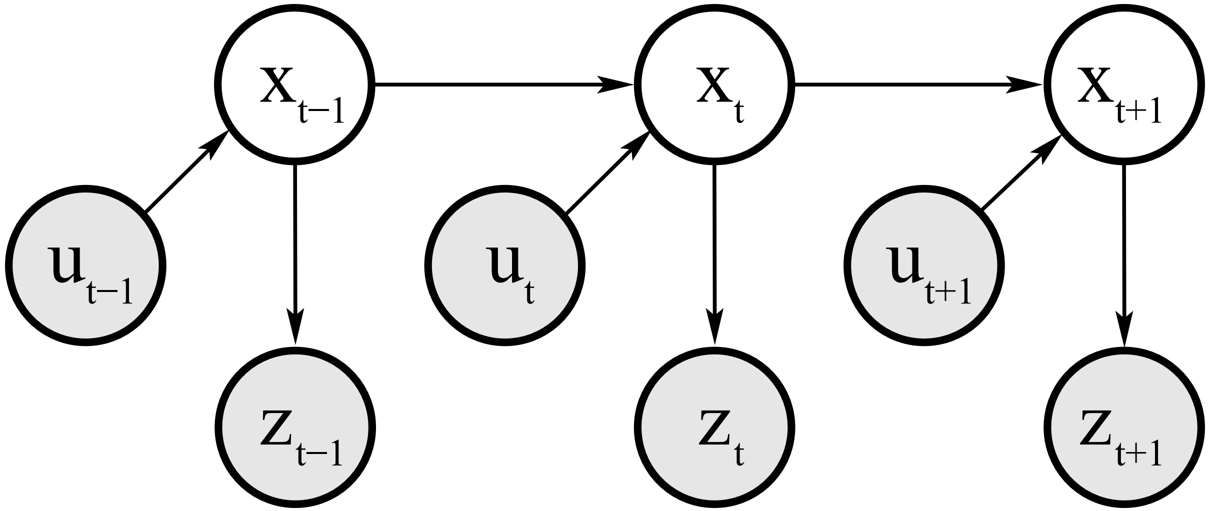
Robot Perception

Dynamic Bayes Network for Robot Pose



Dynamic Bayes Network for Robot Pose

Under the Markov assumption, we are required to model

$$p(x_t|x_{t-1}, u_t)$$

which is called the **state transition probability** and shows how the robot's pose changes in time due to input actions.

And,

$$p(z_t|x_t)$$

which is called the **measurement probability** and shows the probability of sensor measurements given the robot's pose

Sensor modeling

We do not have to model the phenomena that generate the measurements

We aim to model the **typical noise** observed by the sensor.

`Note` **We may use any insights based on the specific phenomena that generate the sensor measurements, but we can treat is a black box**

Laser rangefinder models

Laser rangefinder

Sensor commonly used in robotics uses a narrow laser beam to determine the distance to objects using the time it takes the beam to be reflected from the object (`time of flight`)

Pobabilistic Model

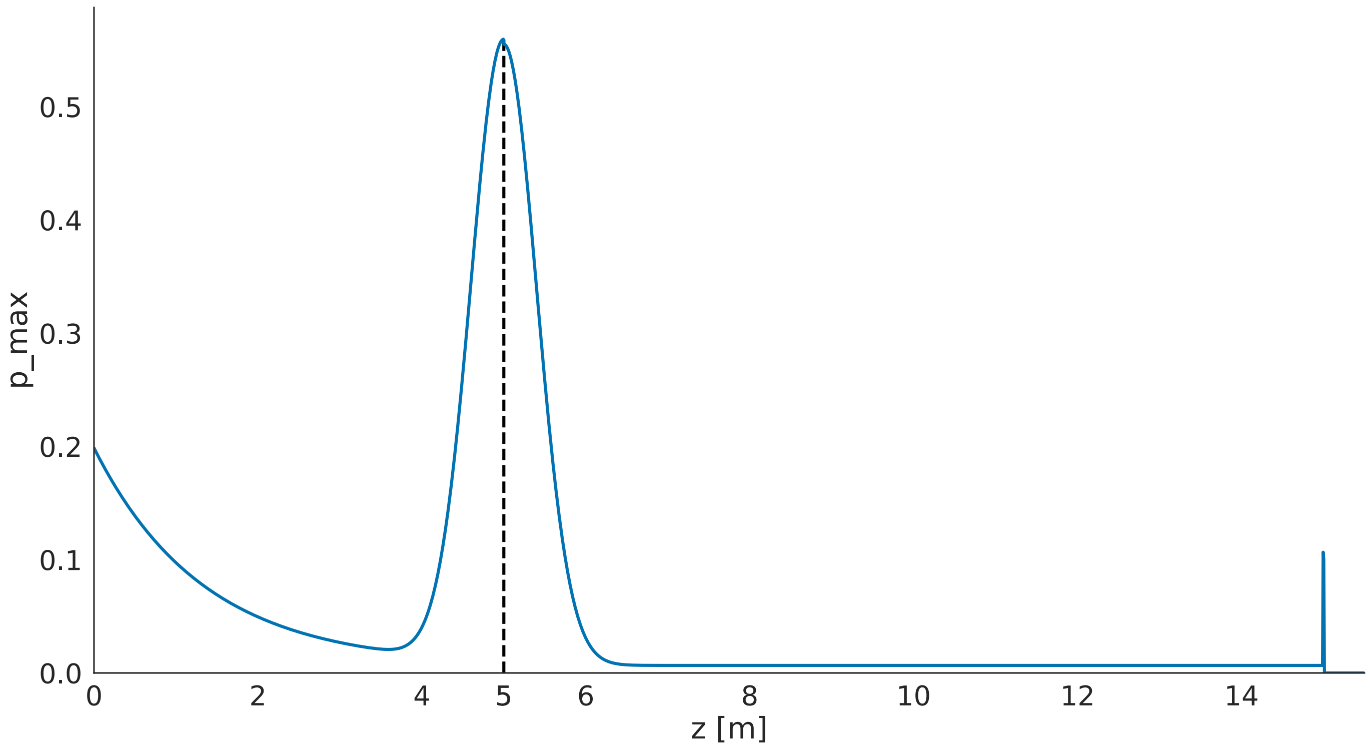
4 error types are used:

1. Measurement noise - p_{hit}
2. Occlusions - p_{short}
3. Missrecognitions - p_{max}
4. Random errors - p_{rand}

Beam model

We combine the 4 errors using weight constants $\alpha_{0:3}$, with $\sum \alpha_{0:3} = 1$

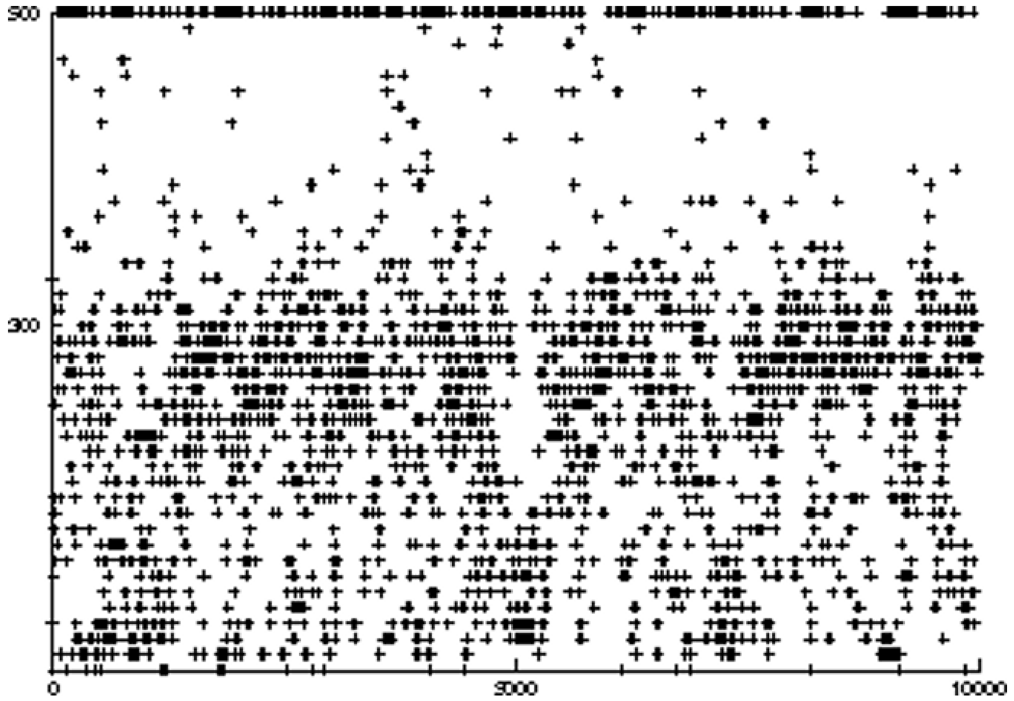
$$p_{tot} = \alpha_0 p_{hist} + \alpha_1 p_{short} + \alpha_2 p_{zmax} + \alpha_3 p_{rand}$$



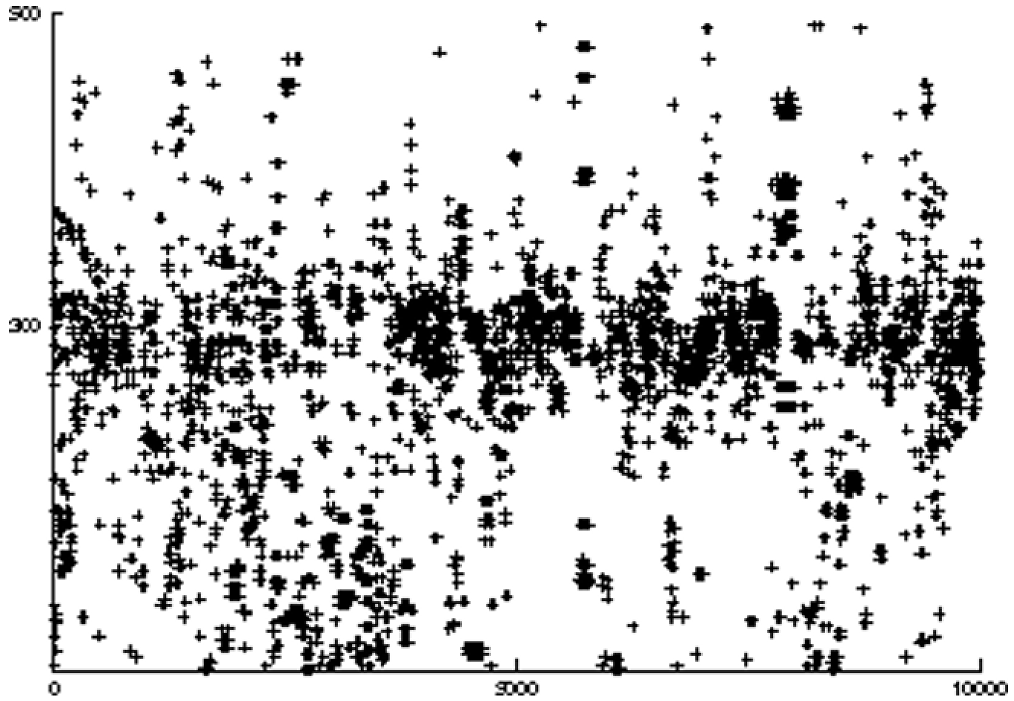
Ultrasonic Sensors

Ultrasonic Sensors

We can use the same Beam model used for rangefinders, with different weights



Sonar

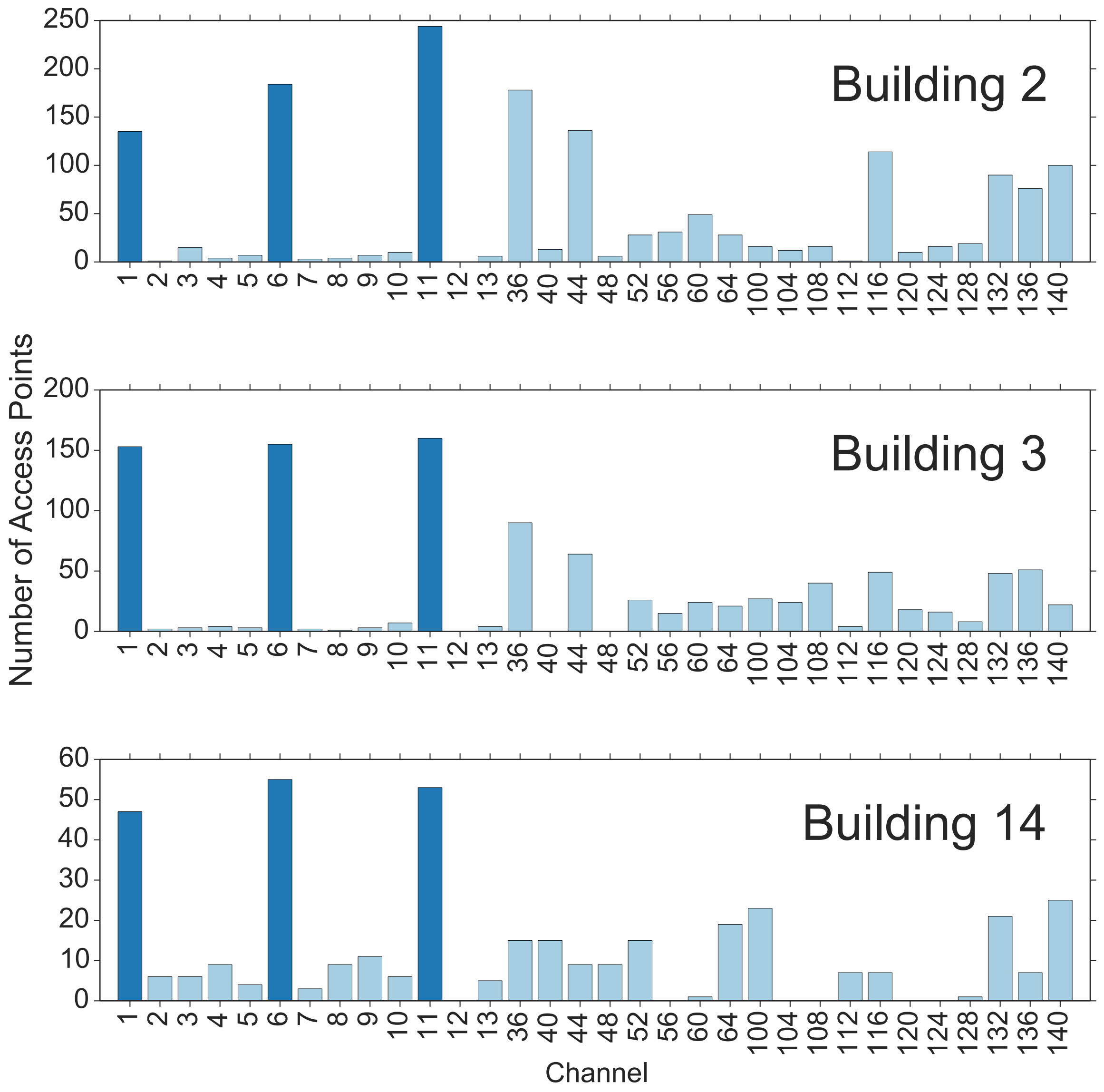


Laser

Wireless Signal Strength

Why use WiFi?

- Ubiquity
- No data association problem



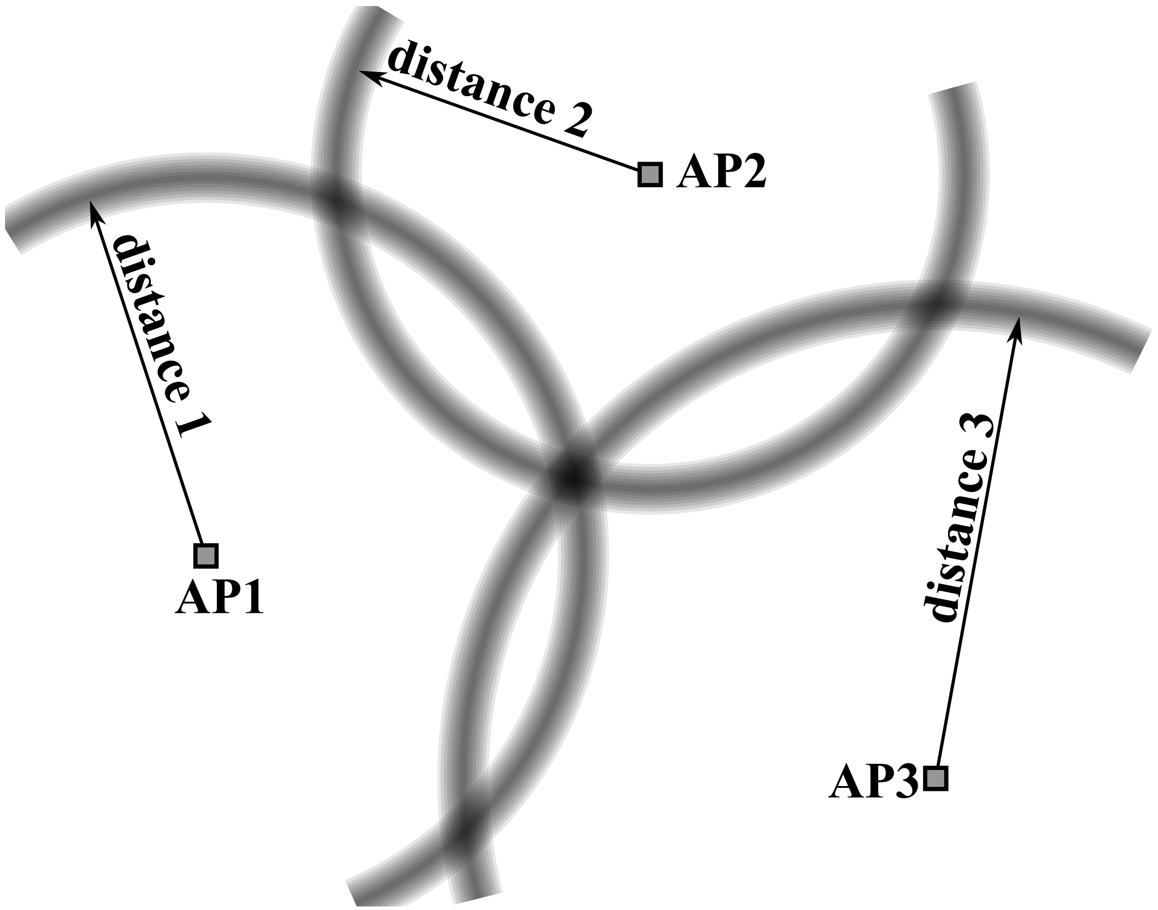
Wireless Signals Propagation Through Space

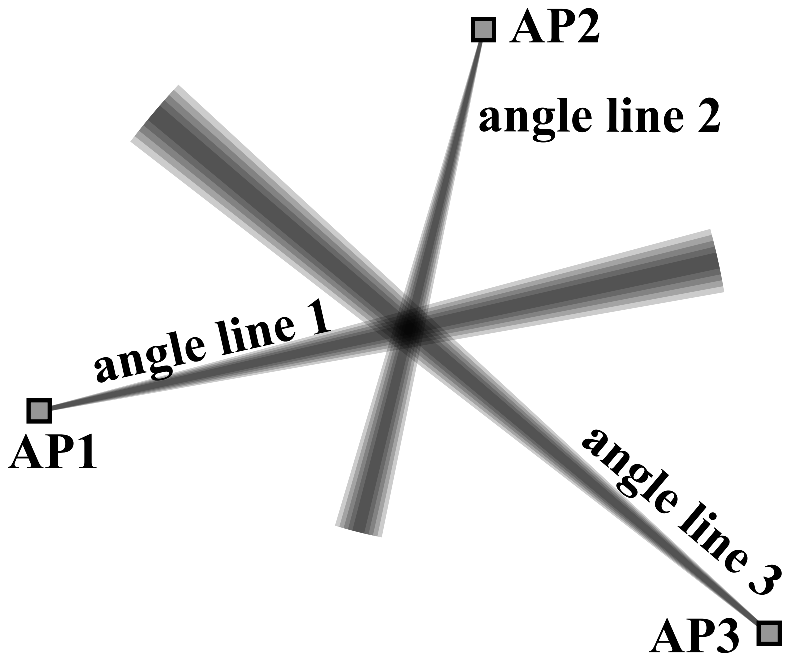
Signal strength lowers as the receiver gets further from the transceiver
If multiple antennas are installed, the direction of the signal can also be computed

Therefore, signal strength can be used to compute the distance and/or bearing to the transceiver.

Wireless Signals Propagation Through Space

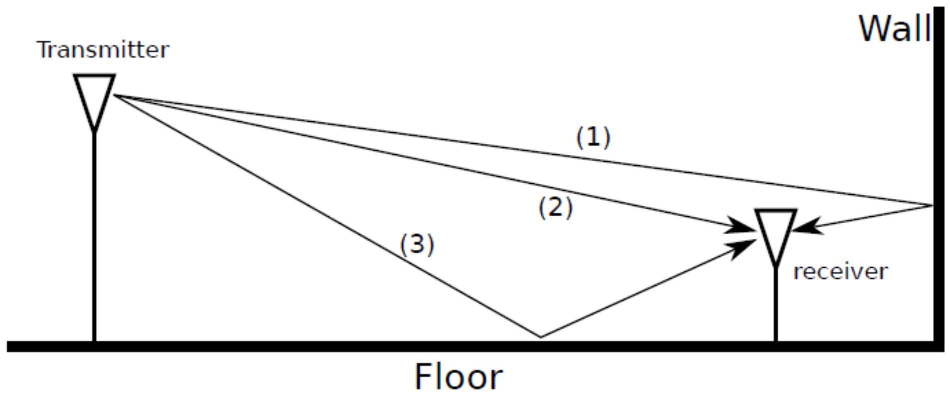
Distance and Bearing can be modeled probabilistically by adding sensor noise

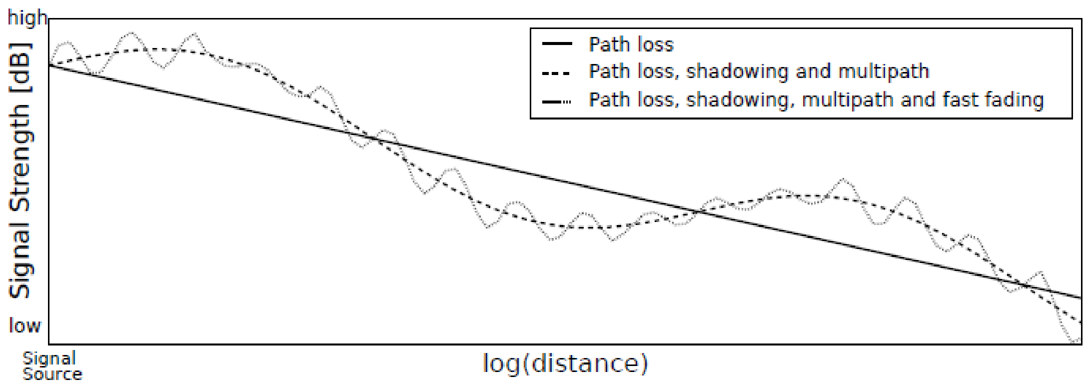




Wireless Signals Propagation Through Space

However, signals travel in all directions, reflecting/refracting and being absorbed, making the behaviour quite complex

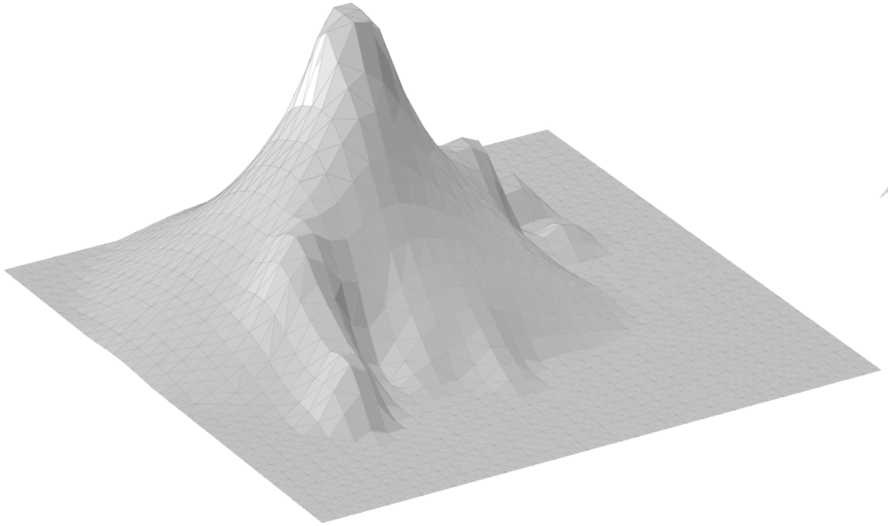




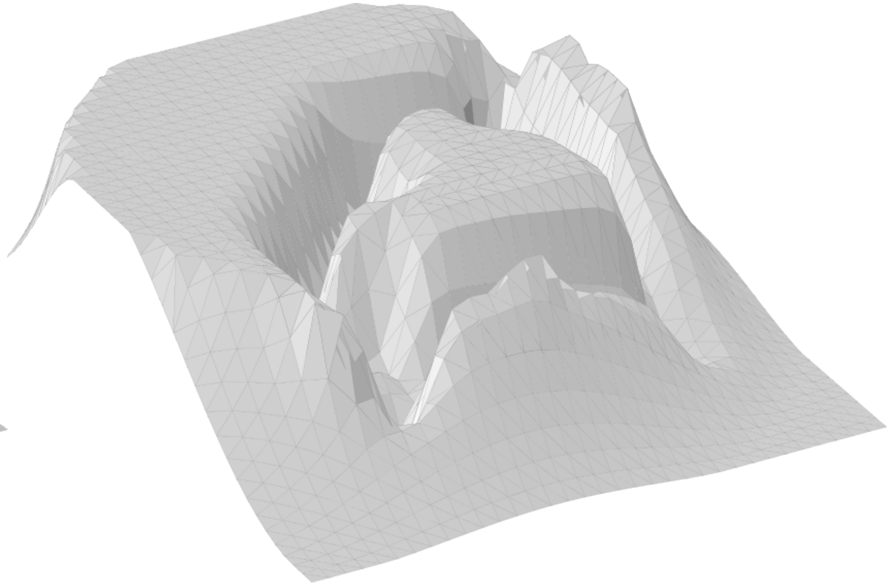
Wireless Signal Strength

Instead, we can build **intensity maps**

Intensity maps are similar to grid maps, where each cell stores the intensity value of a signal at that point, and variance that expresses the confidence in that prediction



Mean



Variance

WiFi measurement model

Measurement noise (p_{hit})

The noise around the correct signal strength is typically modeled as a Gaussian around the true intensity z_m (from the intensity map)with standard deviation σ_m (from the intensity map) and support $[0, 1]$ (we normalize signal strength from 0 ~ -95 dBm to 0 ~ 1)

$$p_{hit}(z) = \begin{cases} \eta \mathcal{N}(z | \mu = z_m, \sigma = \sigma_m) & \forall z \in [0, 1] \\ 0 & \text{otherwise} \end{cases}$$

with η being a normalization constant

$$\eta = \frac{1}{CDF(1) - CDF(0)}$$

WiFi measurement model

Missrecognitions (p_{min})

Probability error to account signals not sensed

Modeled as a probability mass at 0

$$p_{min}(z) = \begin{cases} 1 & \text{if} \quad z = 0 \\ 0 & \text{otherwise} \end{cases}$$

WiFi measurement model

Random errors (p_{rand})

Probability error to account for electronic noise

$$p_{rand} = 1$$

WiFi measurement model

The previous errors are combined using weight constants $\alpha_{0:2}$, with $\sum \alpha_{0:2} = 1$

$$p_{tot} = \alpha_0 p_{hit} + \alpha_1 p_{min} + \alpha_2 p_{rand}$$

Landmark Models

Landmarks

Locally distinct points that can be used as a reference for localization

Landmarks

Example of sensors used

- GPS
- Cameras

Landmarks

Data obtained

- Range, r
- Bearing, ϕ
- Range and Bearing

And may also generate a

- Signature, s

Signatures

a.k.a Descriptors

Signatures are a numerical value (typically a scalar or vector) which identifies the landmark

Signatures can be unique (no repeating values) or similar/same for many landmarks

Probabilistic Model

Given the sensor data, we can define a landmark-based on its feature vector

$$f = \begin{bmatrix} r \\ \phi \\ s \end{bmatrix}$$

Probabilistic Model

Considering a single measurement z can give us multiple features, we can compute the measurement model using iid

$$p(\mathbf{z}|\mathbf{x}, m) = \prod_{f_i} p(f_i|\mathbf{x}, m)$$

or as discussed with rangefinders a smoothed version

$$p(\mathbf{z}|\mathbf{x}, m) = \prod_{f_i} (p(f_i|\mathbf{x}, m))^{\lambda}$$

Feature Maps

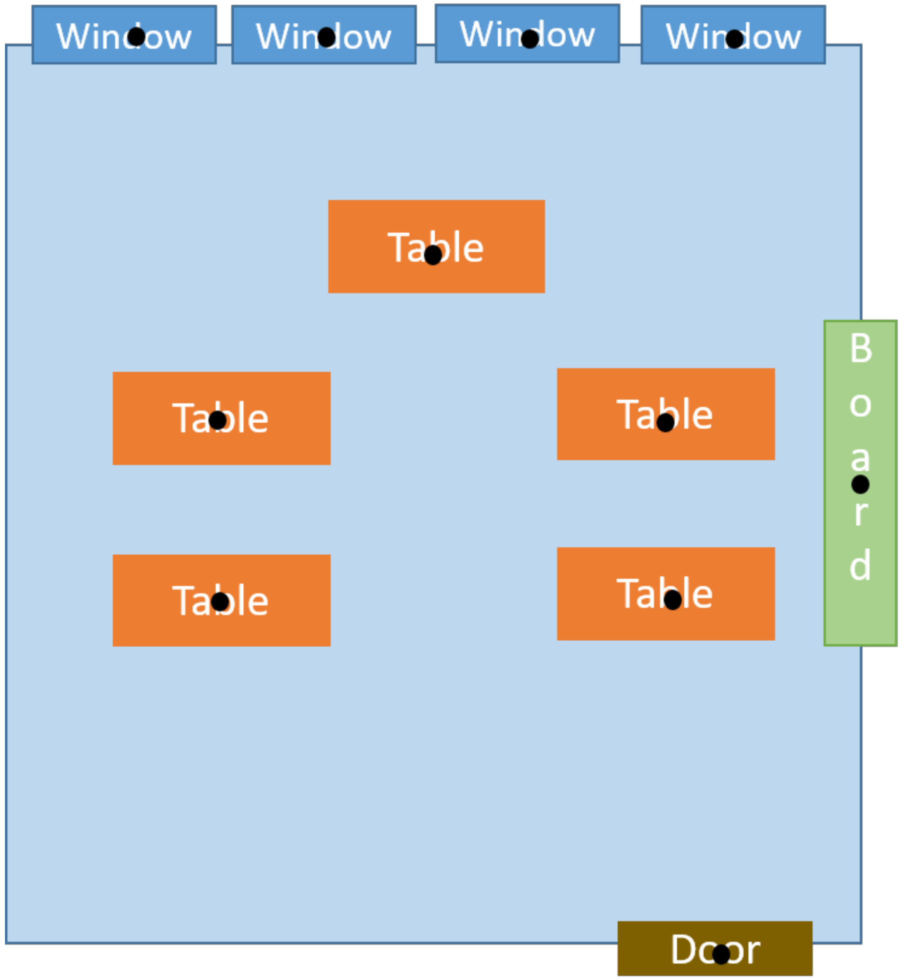
Opposed to rangefinders that typically use grid maps, landmark-based localization uses **feature maps**

Feature maps are a list of graphs, encoding the location of landmarks (with their signature if available)

Feature Maps

A feature map representing the environment on the right could be:

```
7,0 Door      2,6 Table
2,3 Table     1,7 Window
6,3 Table     3,7 Window
8,4 Board     5,7 Window
6,5 Table     7,7 Window
4,5 Table
```



Measurement probability

Depends on the type of data obtained

Measurement model for Range Only

Measurement noise

The noise around the correct distance is typically modeled as a Gaussian around the true distance r_t with standard deviation σ_{hit} and support $[0, r_{max}]$

$$p_{hit}(r) = \begin{cases} \eta\mathcal{N}(r|\mu = r_d, \sigma = \sigma_{hit}) & \forall z \in [0, r_{max}] \\ 0 & \text{otherwise} \end{cases}$$

with η being a normalization constant (due to the modified support) of r.

$$\eta = \frac{1}{CDF(r_{max}) - CDF(0)}$$

r_{max} depends on the sensor (it may not be considered)

Measurement model for Range Only

Random errors

Probability error to account for electronic noise or failure to compute depth.

$$p_{rand} = \frac{1}{r_{max}}$$

Measurement model for Range Only

The two errors are combined using weight constants $\alpha_{0:1}$, with $\sum \alpha_{0:1} = 1$

$$p_{tot}(r) = \alpha_0 p_{hit} + \alpha_1 p_{rand}$$

Measurement model for Bearing Only

Measurement noise

The noise around the correct distance is typically modeled as a Gaussian around the true distance ϕ_t with standard deviation σ_{hit} and support $[-\pi, \pi]$. Alternatively, a VonMisses distribution may be used.

$$p_{hit}(\phi) = \eta \mathcal{N}(\phi | \mu = \phi_d, \sigma = \sigma_{hit}) \quad \forall \phi \in [-\pi, \pi]$$

with η being a normalization constant (due to the modified support)

$$\eta = \frac{1}{CDF(\pi) - CDF(-\pi)}$$

Measurement model for Bearing Only

Random errors (p_{rand})

Probability error to account for electronic noise or failure to compute bearing.

$$p_{rand}(phi) = \frac{1}{2\pi}$$

Measurement model for Bearing Only

The two errors are combined using weight constants $\alpha_{0:1}$, with $\sum \alpha_{0:1} = 1$

$$p_{tot}(phi) = \alpha_0 p_{hit} + \alpha_1 p_{rand}$$

Measurement model for Range and Bearing

Combine the errors for both previous models assuming distance and bearing are independent

$$p(f) = p_{tot}(r)p_{tot}(\phi)$$

Landmark Correspondance

a.k.a Data association problem

Landmarks may not be completely differentiable from one another.
I.e., you may confuse one landmark with another

This is known as the data association problem

Probabilistic modeling considering Data association

To model data association, you simply add one more term to your model, which represents your certainty that you are observing the correct landmark.

$$p(f) = p_{tot}(r)p_{tot}(\phi)p(s)$$

Cameras

Types of Cameras

- Monocular
- Stereo
- RGB-D

Monocular

- FoV Horizontal(Vertical): 70.42 (43.3)
- Resolutions
  - 160x90, 160x120, 176x144, 320x180, 320x240, 352x288, 432x240, 640x360, 800x448, 800x600, 864x480, 960x720, 1024x576, 1280x720. 1600x896, 1920x1080, 2304x1296, 2304x1536
- Frames per Second (fps)
  - 60*, 30, 24, 20, 15, 10, 7.5, 5, 2*
- USB2.0

60Hz only available at 720p

Monocular

- FoV Horizontal(Vertical): Camera lens options available
- Resolution: 1920x1200 to 4096 x 3000
- Frames per Second (fps): 163 to 30 fps
- Global Shutter
- USB3.1 (Ethernet option available)

Monocular

Fish-Eye (Lens)

- FoV: 180 solid

Monocular

Spherical Camera

Two fish-eye lens

- FoV: 360 solid

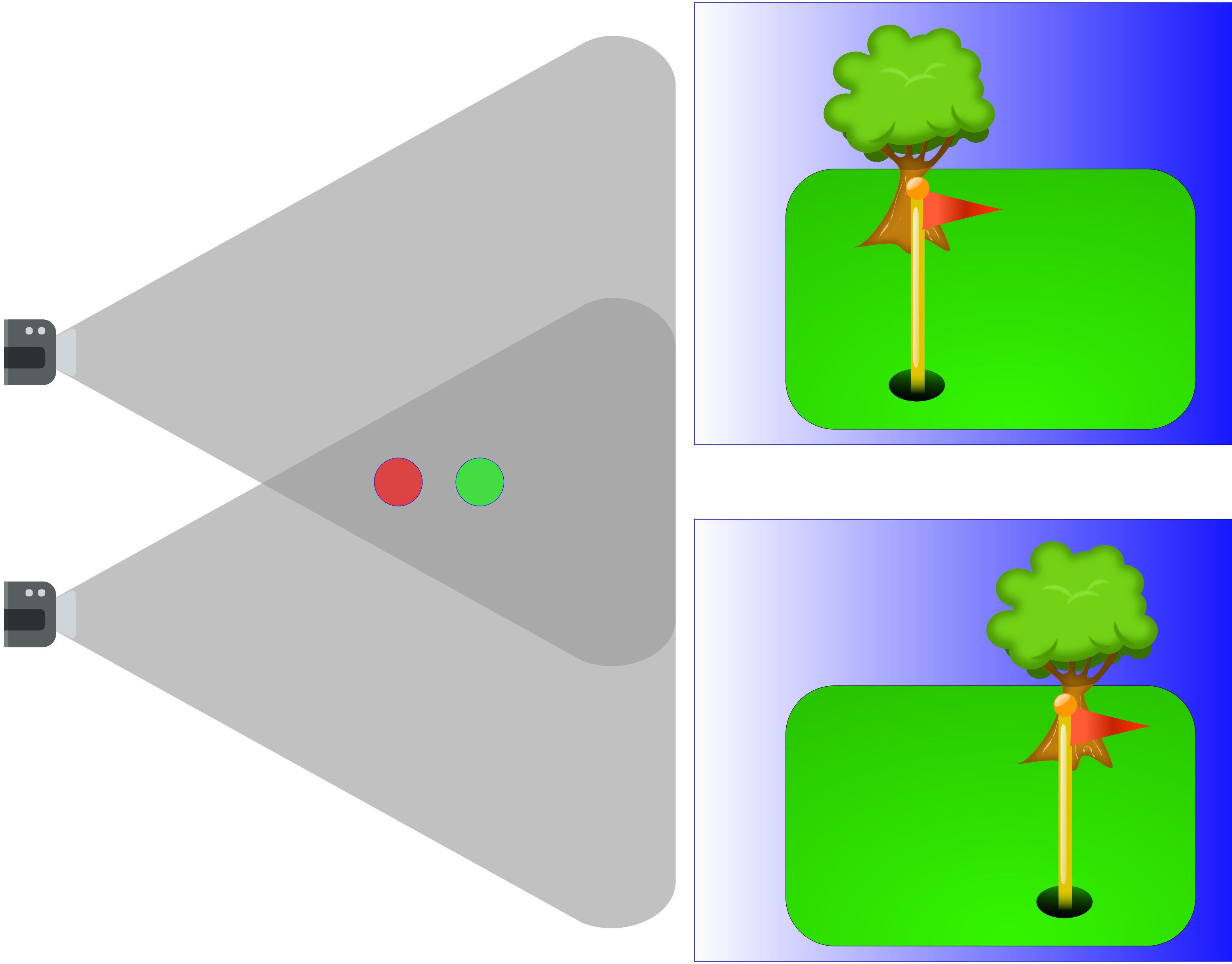
Goto, Tsubasa, Sarthak Pathak, Yonghoon Ji, Hiromitsu Fujii, Atsushi Yamashita, and Hajime Asama. "Spherical Camera Localization in Man-made Environment Using 3 D-2 D Matching of Line Information." (2016).

Stereo Camera

Consists of two cameras mounted at a known distance and relative angle

If the same point is observed by each camera, its relative angle with respect to each reference frame becomes known

Using simple geometry, the xyz location of the point can then be calculated



RGB-D Camera

How to get Landmarks

1. Keypoint Selection
2. Feature Descriptor

Keypoint Selection

Recognizing Corners

Corners can be recognized by computing the gradients of the image

$$f(u, v) = (I(u, v) - I(u + \delta u, v + \delta v))^2$$

Using Taylor Expansion

$$I(u + \delta u, v + \delta v) \approx I(u, v) + \begin{bmatrix} J_u & J_v \end{bmatrix} \begin{bmatrix} \delta u \\ \delta v \end{bmatrix}$$

Keypoint Selection

From the previous equations, we then have

$$f(u,v) = \begin{bmatrix} \delta u & \delta v \end{bmatrix} \begin{bmatrix} J_u^2 & J_{uv} \\ J_{uv} & J_v^2 \end{bmatrix} \begin{bmatrix} \delta u \\ \delta v \end{bmatrix}$$

where $\begin{bmatrix} J_u^2 & J_{uv} \\ J_{uv} & J_v^2 \end{bmatrix}$ is called the *structure matrix*

Keypoint Selection

Edges have a structure matrix with one dominant eigenvalue

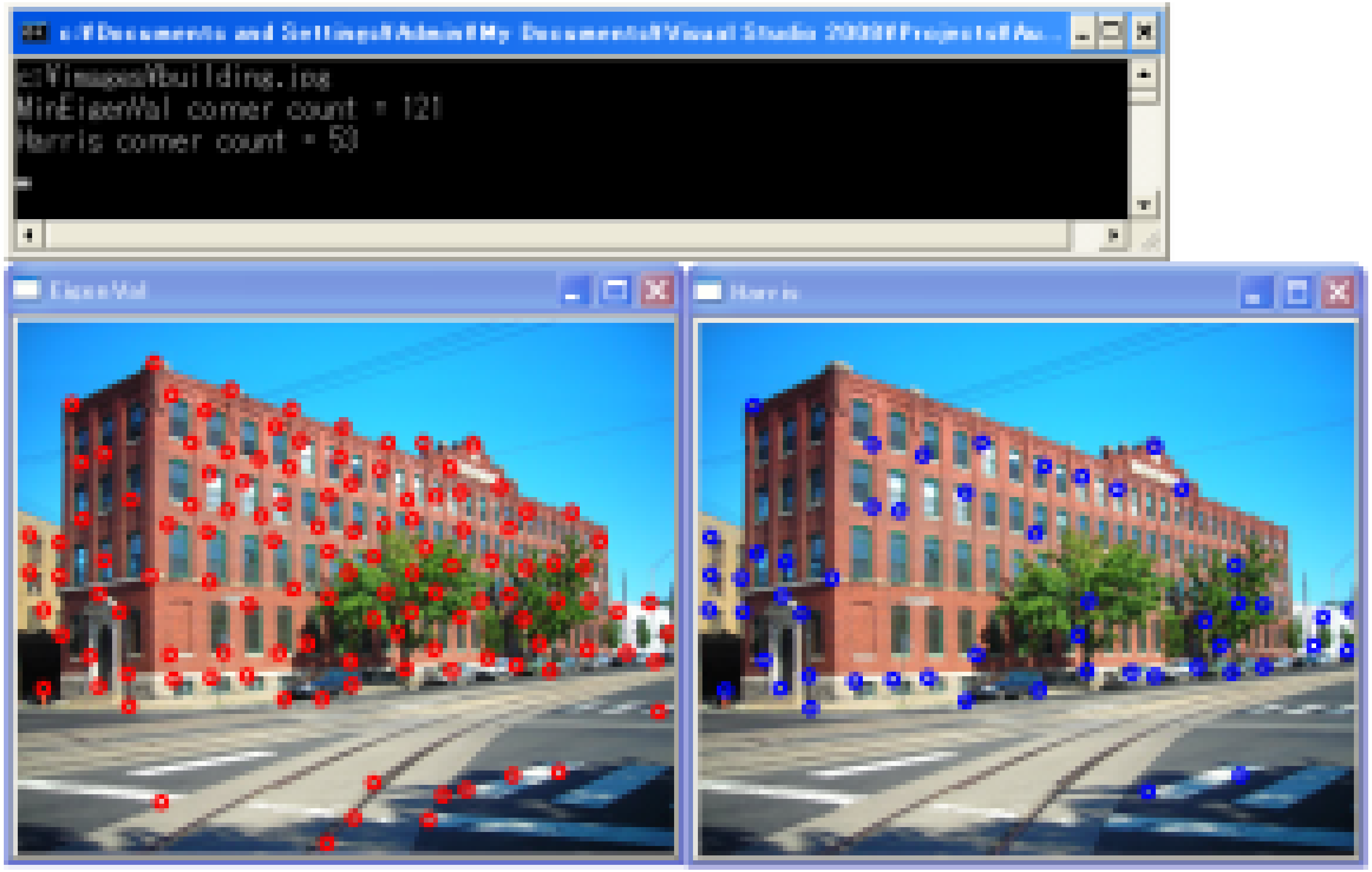
Corners have a structure matrix with two dominant, mostly perpendicular, eigenvalues

Keypoint Selection

In practice, we use filters to compute the structure matrix, and criteria to select corners

Criterions

- Förstner
- Harris
- Shi-Tomasi



https://fahmifahim.com/2010/10/22/opencv-corner-detection

Point (feature) descriptors

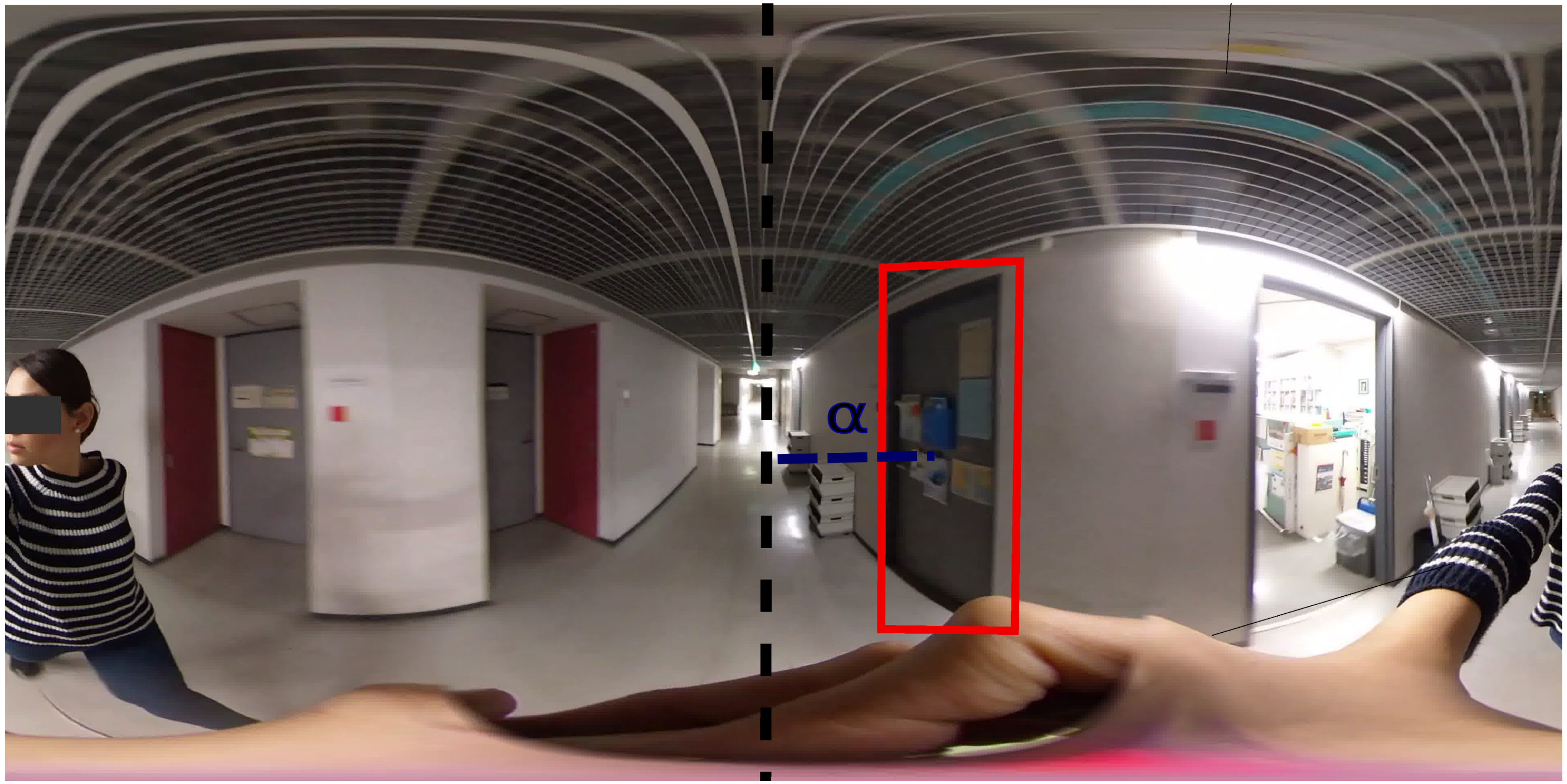
Describe the local surroundings of the keyframes

1. BRIEF: Binary Robust Independent Elementary Feature
2. SURF: Speed-Up Robust Features
3. SIFT: Scale Invariant Feature Transform
4. AKAZE: Accelerated KAZE (wind)
   among many others

They encode a patch of pixels into a vector, trying for the encoding to be invariant with respect to rotations, scaling, and rotations.

Using High-level Features

Example: Semantic Localization (YOLO)



Uygur, Irem, Renato Miyagusuku, Sarthak Pathak, Alessandro Moro, Atsushi Yamashita, and Hajime Asama. 2020. "Robust and Efficient Indoor Localization Using Sparse Semantic Information from a Spherical Camera" Sensors 20, no. 15: 4128. https://doi.org/10.3390/s20154128

Sensors Comparison

	GPS	LRF	Ultrasonic	Camera	WiFi	Magnetic
Distance/Bearing	NA	D&B	D&△B	B/D&B	D/B/D&B	NA
Indoors and Outdoors	X	O/X	O	△	O	O
Accurate	O	O	X	O	X	△
Low Cost ($)	O/X	X/O	O	O	O	O
Low Computation	O	O	O	X	O	O
No line of sight	X	X	X	X	O	O
No data association problem	O	X	X	X	O	O