→ Sensor Fusion

Act of combining sensory data from separate sources. The resulting info is 'better' than individually. (more accurate, complete). It's considered a subset of Information Fusion.

Usually based on modeling the sensors and the system being measured. Most common methodologies are probability based. (others are interval calculus, fuzzy logic, theory of evidence).

↳ Bayes Rule

Determine prob. of event given the result of related events.

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

→ conditional probability

P(A) and P(B) are marginal probabilities

↳ Bayesian Filter

Application of the Baye's Rule when:

- x_t is state vector at time t
- u_t is the control vector used to drive from x_{t-1} to x_t
- z_t is the observation of the state at time t.

At an instant t an estimation of the current state can be achieved by application of Baye's Rule:

$$P(x_t | z_t, U_t) = \frac{P(z_t | x_t) P(x_t | z_{t-1}, U_t)}{P(z_t | z_{t-1}, U_t)}$$

↳ Grid world + measurement integration

Measurement integration is performed using the product of sensors model and previous estimate.

→ moving: when the robot moves, belief is updated through convolution.

→ Kalman Filter

Algorithm that uses a series of measurements observed over time, containing noise and other inaccuracies, producing estimations of unknown variables that tend to be more accurate than those based on a single measurement alone.

Assumes linear model, gaussian estimations (mean and variance) and noise is gaussian (mean = 0).

Markov assumption: next estimate only depends on the previous estimate.

- Considers action model and physics (observation model).

- Uses product of gaussian distributions and convolution (motion forecast)

→ Particle Filter (Monte Carlo Filter)

Recursive Bayesian filter that uses a random set of samples (particles) to approximate the posterior distribution of a system's state. The particles represent possible states of the system and the filter uses them to approximate the true state of the system. Non-parametric so it can cope with several types of distributions, not just Gaussian.

2 steps:

- Evolution: evolution of the current step, weighing and creation of a dataset. Weighing computes the likelyhood of the particle getting to the new particle set.

- Resampling: resampling of the dataset for the next evaluation. Resamples particles according to their weights, discarding particles with low weights and creating new particles to be evaluated. It's used to prevent the filter beins stuck at a local minimum.