Robotic Perception

Adane Letta (PhD)

Nov 2024

Outline

- Robotic Perception
- Sensors
- Sensor Fusion

What is Robot Perception?

- Making sense of the unstructured real world…

- Incomplete knowledge of objects and scene
- Environment dynamics and other agents
- Imperfect actions may lead to failure

Sensors

- Making contact of the physical world through multimodal senses

- Sensors constitute the perceptual system of a robot

- Sensors are physical devices that measure physical quantities, such as:
    - **Contact -> bump, switch**
    - **Distance -> ultrasound, radar, infra red**
    - **Light level -> photo cells, cameras**
    - **Sound level -> microphones**
    - **Strain -> strain gauges**
    - **Rotation -> encoders**
    - **.............**

- Active and Passive Sensors:
    - Active sensor uses external projecting devices that emit light wavelength, signal or patterns to interact with the scene.
        - 3 types of active sensors used in robotics : Lidar (Light Detection and Ranging), Structured-Light and time-of-flight.
    - Passive sensors gather target data through the detection of vibrations, light, radiation, heat or other phenomena occurring in the environment without external devices.

Sensors

- Sensors range from simple to complex in the amount of information they provide:
  - A switch is a simple on/off sensor
  - A human retina is a complex sensor consisting of more than a hundred million photosensitive elements (rods and cones)
- Sensors provide raw information, which can be treaded in various ways,
- Sensors do not provide state/symbols, just signals
- A great deal of computation may be required to *convert the signal* from a sensor into useful state for the robot.
  - Can be processed to various levels.
- For example, we can simply *react* to the sensor output:
  - If the switch is open, stop, if the switch is closed, go.
- More complex sensors require and allows to do more complex processing

Sensors: Signals

- Example 1. just to figure out if a switch is open or closed, you need to measure voltage going through the circuit; that's electronics
- Example 2. now suppose you have a microphone and you want to recognize a voice and separate it from noise; that's signal processing
- Example 3. now suppose you have a camera, and you want to take the pre-processed image
  - (suppose by some miracle somebody has provided you with all the edges in the image, so you have an "outline" of the objects),
  - and you need to figure out what those objects are,
  - perhaps by comparing them to a large library of drawings;
    - Computation

Sensory data processing is challenging and can be computationally intensive and time consuming. Why does that matter? Because it means your robot needs a brain to do this processing

New and Better Approaches to Perception

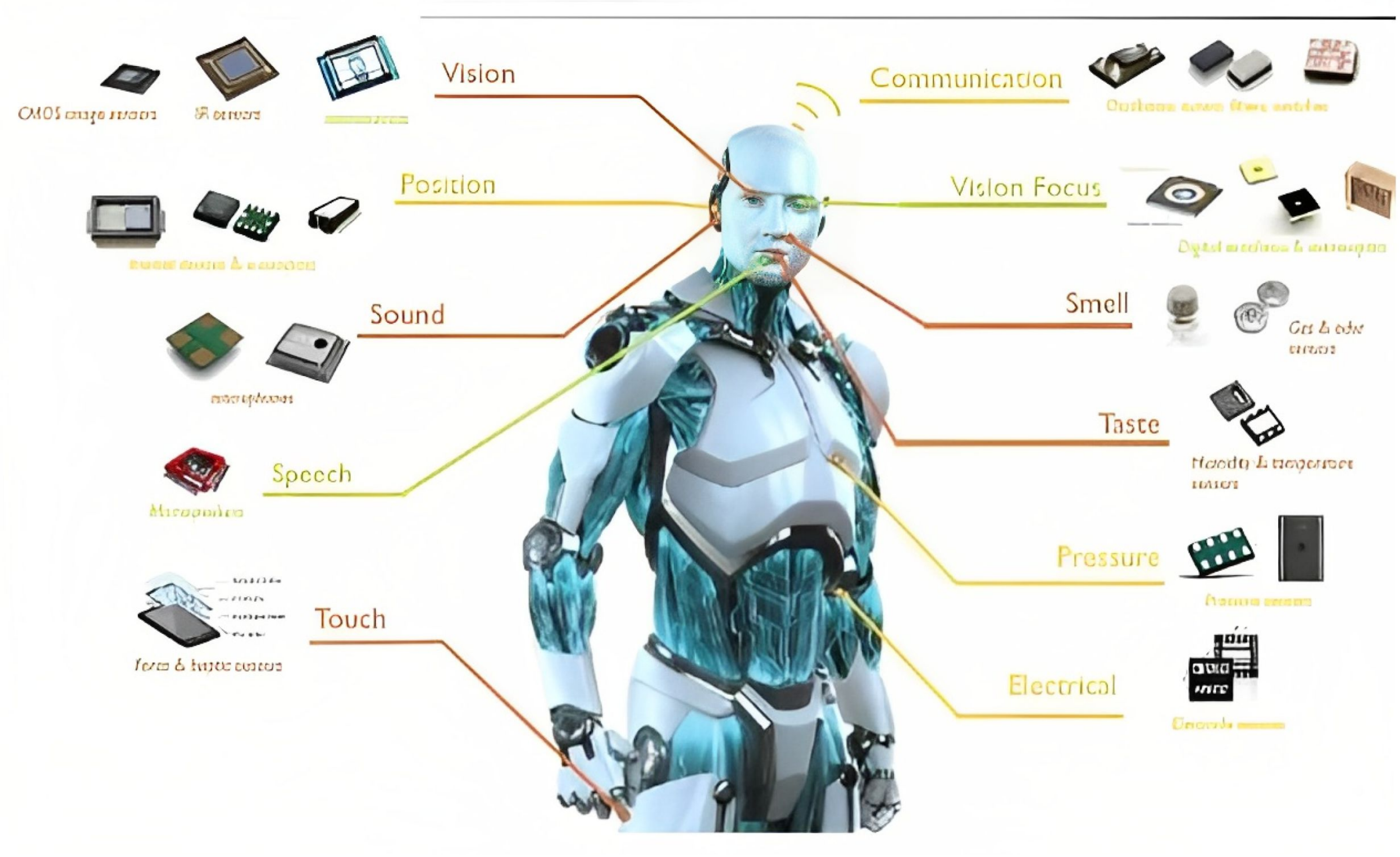
- Perception in the context of action and the task
  - Action-oriented perception
  - Expectation-based perception uses knowledge about the world as constraints on sensor interpretation
  - Focus-of-attention methods provide constraints on where to look
  - Perceptual classes partition the world into useful categories

New and Better Approaches to Perception

- Nature is very clever in the way it solves perception/sensing problem;
  - It evolves special sensors with special geometric and mechanical properties.
    - Facetted eyes of flies, or
    - Polarized light sensors of birds
    - Horizon/line sensors of bugs
    - The shape of the ear, etc.

- All biological sensors are examples of clever mechanical designs that maximize the sensor's properties, i.e., it's range and correctness.

Sensors



Source: Albustanji, R.N.; Elmanaseer, S.; Alkhatib, A.A.A. Robotics: Five Senses plus One—An Overview. *Robotics* **2023**, *12*, 68. https://doi.org/10.3390/robotics12030068

Sensors

- Robots can be equipped with a range of senses to allow them to perceive and interact with the world in a more natural and intuitive way. *Include:*
  - Vision: allows the robot to see and recognize objects and navigate its environment.
  - Hearing: enables the robot to recognize sounds and respond to vocal commands.
  - Touch: allows the robot to perceive information about the texture, shape, and temperature of objects through the sense of touch.
  - Smell: enables the robot to recognize and classify different odors.
  - Taste: enables the robot to identify the chemical composition of materials.
- Many robots use a combination of different senses to perceive and interact with the environment.
- The integration of multiple sensory systems in robots has enabled them to perceive, interact with, and navigate their environment in a way similar to humans.

Robotic Vision

- The major components of a machine vision system include lighting, lens, image sensor, vision processing, and communications.
- Lighting illuminates the part to be inspected
- The lens captures the image and presents it to the sensor in the form of light.
- The sensor in a machine vision camera converts this light into a digital image, which is then sent to the processor for analysis
- Vision processing: is how robotic vision systems obtain data from an image that is used by robots for analysis in order to determine the best course of action for operation.
- Communications connect and coordinate all robotic vision components.
  - This allows all vision components to effectively and quickly interact and communicate for a successful system.

Robotic Vision

- Vision processing
  - The digitized image is subjected to image processing and analysis for data reduction and interpretation of the image
  - Further subdivided into:
    - Prepossessing: It deals with techniques such as noise reduction and enhancement details.
    - Segmentation: It partitions an image into objects of interest.
    - Description: It computes various features, such as size, shape, etc., suitable for differentiating one object from another.
    - Recognition: It defines the object.
    - Interpretation: It assigns meaning to recognized objects in the scene.

Trends and Challenges for Vision Sensing in Robotics

- There are still several research challenges that need to be addressed to further improve the capabilities of robotic vision systems
  - Multi-sensor perception:- was identified as a popular trend in vision sensing in robotics in 2022.
    - Combining data from multiple sensors, such as cameras, LIDAR, and RADAR, to improve the accuracy and robustness of perception in robotics
    - Explainable artificial intelligence (XAI): emerged as a popular trend in vision sensing in robotics, where robots can explain their perceptions and decision-making processes to humans in a transparent and understandable manner
    - Edge computing for real-time perception: Edge computing was identified as a popular trend in vision sensing in robotics
- Robustness to lighting conditions-- key challenges in robotic vision is in developing systems that can work effectively work under varying lighting conditions.

Hearing Sense

- A robotic hearing system, also known as an auditory system, is a type of sensor that allows a robot to detect and interpret sound waves.
- There are several ways that robots can “hear”.
- One common method is to use microphones or other sensors that are able to detect sound waves and convert them into electrical signals.
- These signals can then be processed by the robot’s computer system and used to understand spoken commands or other sounds in the environment.

Hearing Sense

- Component of Robotic Hearing Systems
  - Microphones or other sound sensors: devices that detect sound waves and convert them into electrical signals
  - Amplifiers: These are electronic devices that are used to amplify the electrical signals that are generated by microphones or sound sensors.
  - Analog-to-digital converters (ADCs): These are devices that are used to convert the analog electrical signals from the microphones or sound sensors to digital
  - Computer system: This is the central processing unit of the robot, which is responsible for controlling the various functions and sensors of the robot.
  - Algorithms and software: These are the instructions and programs that are used by the computer system to analyze and interpret digital data from microphones or sound sensors.

Trends and Challenges in the Field of Robotic Hearing

- Sound localization and separation were identified as popular trends in the field of robotic hearing in 2022.
  - Enhancing the capability of robots to accurately locate and separate different sound sources
- Speech recognition and synthesis: continue to be popular trends in the field of robotic hearing, with a focus on improving the ability of robots to understand and produce human speech
- Auditory scene analysis: This has emerged as a popular trend in the field of robotic hearing, where robots can analyze complex sound scenes and identify individual sound sources
- Cross-modal perception is a popular trend in the field of robotic hearing
  - Information from different sensory modalities, such as vision and hearing, is combined to improve the accuracy and robustness of perception in robotics

Tactile Sense

- Tactile sense, also known as the sense of touch
- The tactile sense is often simulated using sensors that are placed on the surface of a robot’s skin or limbs
- These sensors can detect pressure, temperature, and other physical sensations, and send this information to the robot’s control system

Components of Robotic Tactile Sensing

- Components of Robotic Tactile Sensing
  - Sensing: The robot uses sensors to detect physical sensations, such as pressure, temperature, and force.
  - Data collection: The sensor data are collected by the control system and stored in a buffer or memory
  - Data processing: The control system processes the sensor data using algorithms that interpret the data and provide the robot with a sense of touch.
  - Decision-making: The control system uses the processed sensor data to make decisions about how to interact with the environment and how to move the robot's body.
  - Actuation: The control system sends commands to the robot's actuators, which are responsible for moving the robot's body.

Trends and Challenges in Robotic Tactile Sensing

- Soft robotics: Soft robots are robots that are made from flexible materials, such as silicone or rubber, and are designed to be able to deform and adapt to their environment.
- Artificial skin: Researchers are working on developing artificial skin for robots that is capable of detecting and interpreting tactile information
- Grasping and manipulation: Tactile sensing is essential for robots to be able to grasp and manipulate objects, especially those that are delicate or irregularly shaped.
- Prosthetics and rehabilitation: Tactile sensing is also important in the development of prosthetics and rehabilitation devices.
- Perception and learning: Tactile sensing can be used to help robots learn about their environment and develop more sophisticated perception capabilities.

Sensor Fusion

- Another clever thing to do is to combine multiple sensors on a robot to get better information about the world.
- Sensor fusion integrates data from multiple sensors to provide a comprehensive and accurate understanding of the environment or system being monitored or controlled.
- This is called *sensor fusion*.
- Sensor fusion is not simple:
  - Different sensors give different types, accuracy and complexity of information;
  - processing is necessary to put them together in an intelligent and useful way,
  - and in real-time.
- The brain processes information from many sensors (vision, touch, smell, hearing, sound).
- The processing areas are distinct in the brain (and for vision, they are further subdivided into the "what" and "where" pathways).
- Much complex processing is involved in combining the information.

A single sensor cannot adequately sense environmental information

Sensor Fusion:Basic Framework of Multi-sensor Data Fusion

1. Different types of sensors (active or passive) collect the data of the same targets

2. The transformation of feature extraction is performed on the sensor output data to extract the feature vector.

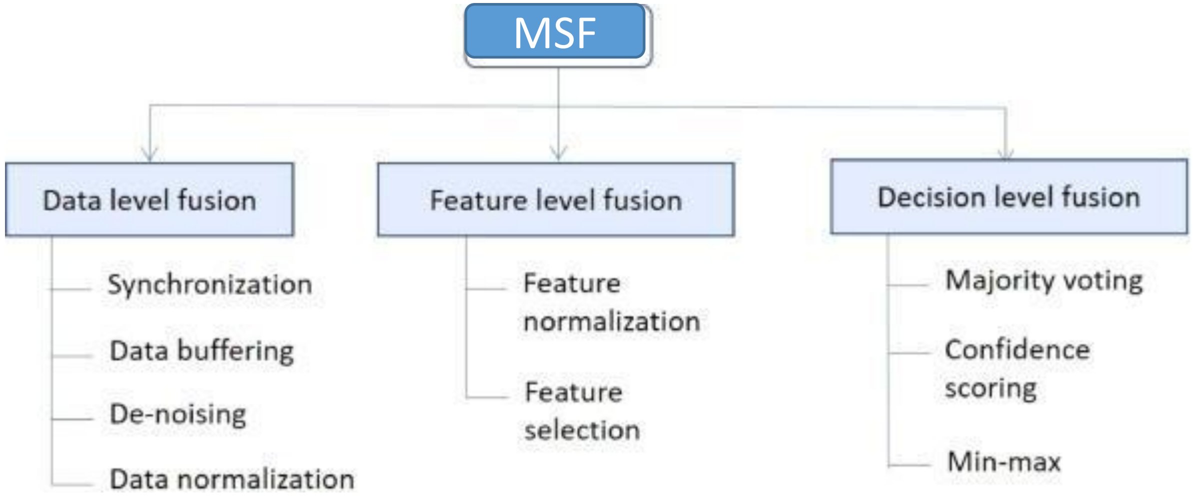
3. Conducting pattern recognition processing for the above obtained feature vectors to achieve the description of all targets in the image by each sensor;

4. The features of each sensor are grouped according to the target category, then the fusion method is used to synthesize the data of each sensor, and the consistent interpretation and description of the target is obtained.

Source: [multi-sensor data fusion - an overview | ScienceDirect Topics]

What are the available multi-sensors data fusion levels?

- Multi-sensors data are fused in three different levels:
  - Data-level fusion: raw data from multiple sensors is combined before any significant processing occurs
  - Feature-level fusion: features (specific patterns or characteristics) are extracted from the raw sensor data, and these features are then combined.
  - Decision-level fusion: each sensor independently processes its data and makes a decision or inference.



Integration and communication

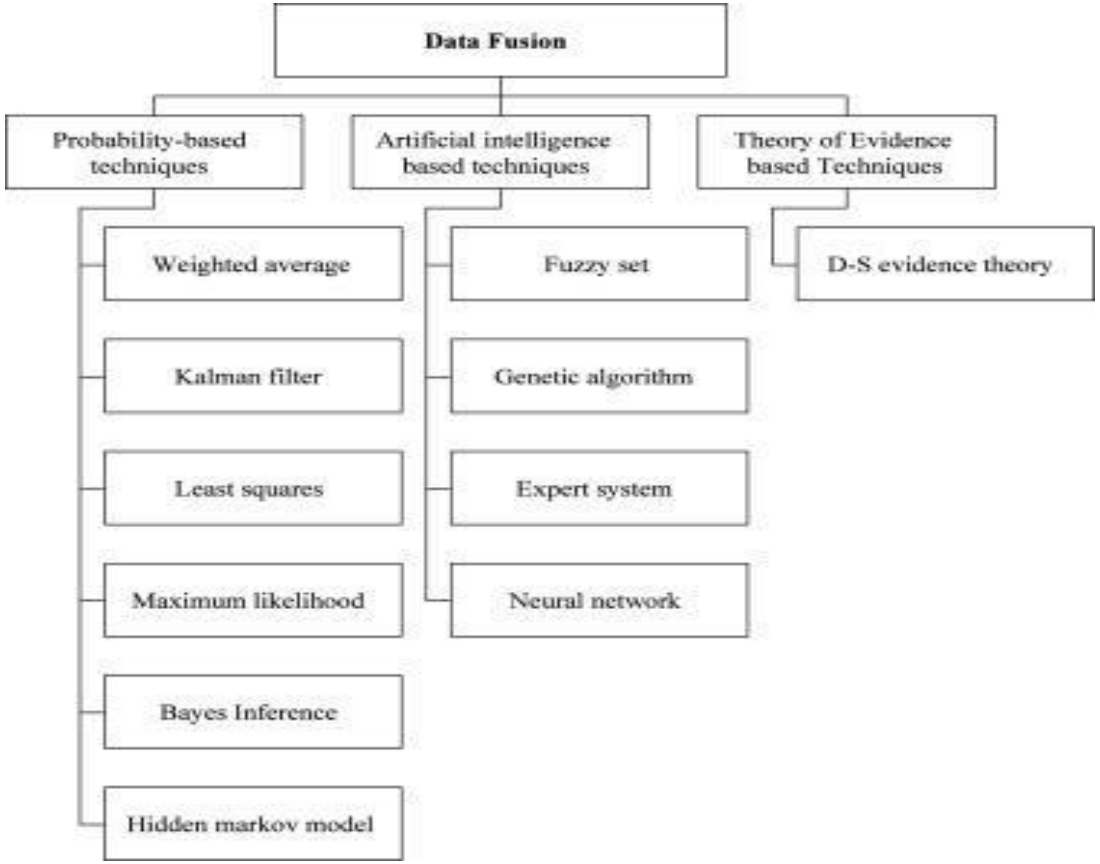
- Sensor data integration often involves:
  - Data formats,
  - Protocols,
  - Time synchronization.

- Integrating heterogeneous sensor data and ensuring seamless communication among sensors and processing units are essential.

- Possible through standardized communication protocols (e.g., CAN bus, Ethernet), middleware for data integration, and data synchronization methods.

Signal processing techniques

- Signal processing algorithms preprocess, filter, and extract useful information from raw data. This includes noise reduction, feature extraction, and data normalization. These algorithms play a crucial role in preparing sensor data for fusion.
  - Kalman Filtering:
  - Bayesian Inference:
  - Wavelet Analysis:
  - Fourier Transforms:
  - Hidden Markov Models:
  - Neural Networks:
  - Consensus Filtering:



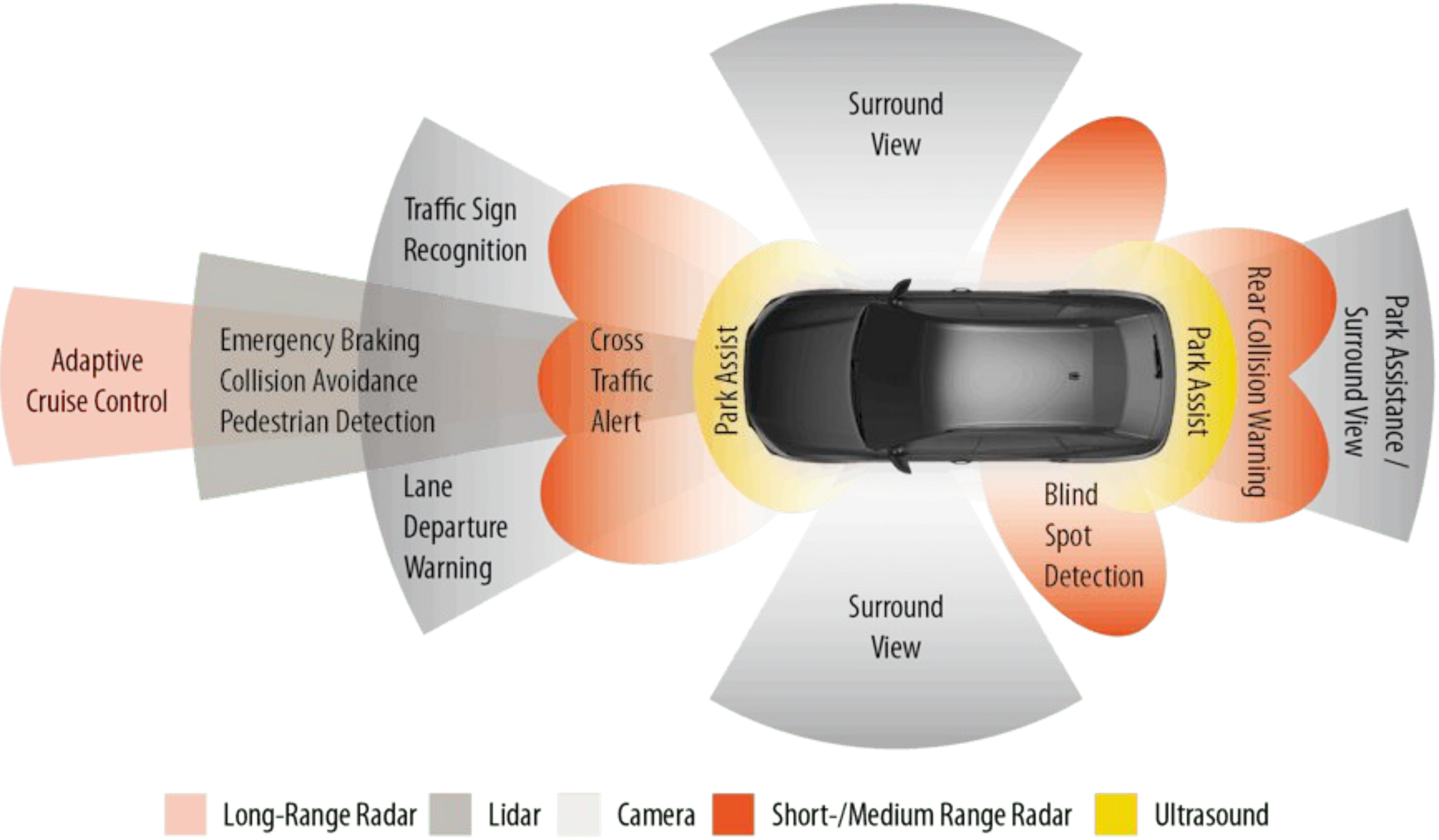
Source: [What is Sensor Fusion? | Dewesoft]

Sensor fusion applications

- Applications for sensor fusion range from medical imaging to robotics and from self-driving cars to industrial automation and control systems.
    - Data Acquisition Systems
    - Self-driving Vehicles
    - Drones
    - Indoor Navigation
    - Industrial Automation and Process Control
    - Medical Imaging
    - Neural Networks
    - Robotics
    - Healthcare
    - Augmented Reality and Virtual Reality

Key sensor fusion technologies

- There critical hardware and software technologies behind sensor fusion.



self-driving vehicle technology

Key sensor fusion trends

- Cross-domain fusion
  - Sensor data from different domains, such as IoT devices, social media, and public databases, can be integrated to provide a more holistic understanding of complex systems and phenomena.
- Leveraging quantum computing
  - Engineers can leverage advances in quantum computing to enhance sensor fusion's speed and capabilities. Real-time sensor fusion is a growing requirement… autonomous vehicle
- Leveraging artificial intelligence
  - Integrating advanced AI and machine learning algorithms will enable more intelligent and adaptive sensor fusion systems to learn from data and improve continuously.
- Protection of privacy rights
  - Ensure that sensor fusion continues to advance while not compromising privacy rights.

Sensor fusion challenges

- Using sensor fusion effectively requires more than combining sensors, algorithms, and signal processing. Challenges include:
  - Data heterogeneity
    - Sensors differ widely, and data can vary significantly in format…. integrating data from different types of sensors (e.g., cameras, LiDAR, RADAR) with diverse characteristics can be challenging.
  - Noise and uncertainty
    - Sensors are prone to noise, inaccuracies, and uncertainties due to environmental factors, hardware limitations, or inherent sensor characteristics.
  - Computational complexity
    - Sensor fusion often involves complex mathematical algorithms and computational processes, especially in real-time applications that require fast data processing.

Localization and mapping

- One of the essential requirements for autonomous robots is the ability to navigate through an unknown environment while avoiding obstacles and reaching their destination safely.

- Localization and mapping technologies estimate objects' position and orientation.

- SLAM (Simultaneous Localization and Mapping) techniques are used in sensor fusion applications, especially in robotics and autonomous vehicles, to build a map of the surroundings with the sensor platform localized within it.

  - A popular approach for robot localization and mapping in unknown environments

SLAM is the process by which a mobile robot can build a map of an environment and at the same time use this map to compute its own location.

Simultaneous Localization and Mapping (SLAM)

- The SLAM problem is generally regarded as one of the most important problems in the pursuit of building truly autonomous mobile robots.
- A topic of active research in robotics for several decades, and many techniques and approaches have been proposed and tested in various scenarios.
- Robustly mapping unstructured, dynamic, and large-scale environments in an on line fashion remains largely an open research problem.

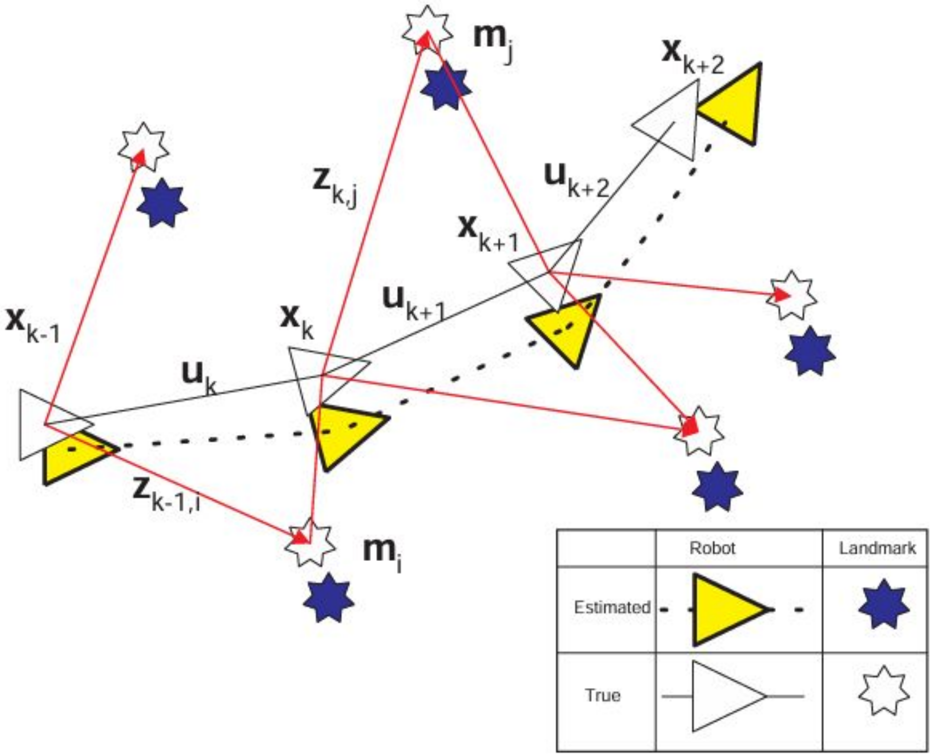
SLAM…….

- SLAM algorithms typically consist of two components: mapping and localization.
- Mapping component involves building a map of the environment using sensor data, such as laser range finders, cameras, or sonar sensors.
- Localization component involves estimating the robot's position within the mapped environment.

SLAM: Problem Definition

- The SLAM problem is defined as follows:

"A mobile robot roams an unknown environment, starting at an initial location x_0. Its motion is uncertain, making it gradually more difficult to determine its current pose in global coordinates. As it roams, the robot can sense its environment with a noisy sensor. The SLAM problem is the problem of building a map of the environment while simultaneously determining the robot's position relative to this map given noisy data."



- X_k: the state vector describing the location and orientation of the vehicle at time k.
- u_k: the control vector applied the time k-1.
- m_i: a vector describing the location of the ith landmark. The landmarks are motionless.
- z_{ik} : an observation taken from the vehicle of the location of the ith landmark at time k.

Source: https://www.cs.columbia.edu/~allen/F19/NOTES/slam_paper.pdf

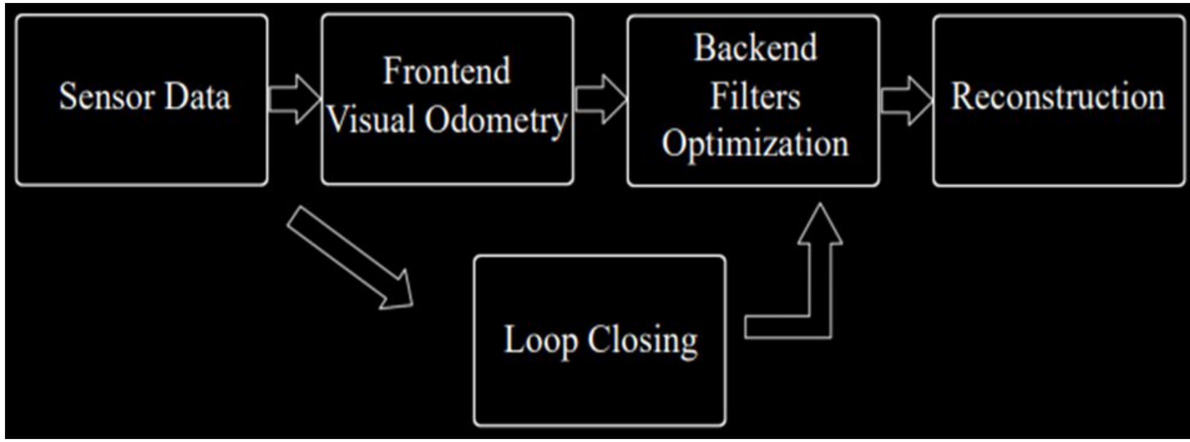
A simultaneous estimate of both robot and landmark locations is required. The true locations are never known or measured directly. Observations are made between true robot and landmark locations.

SLAM: Problem Definition

- In SLAM both the trajectory of the platform and the location of all landmarks are estimated on-line without the need for any a priori knowledge of location.

SLAM framework overview

- The classic visual SLAM framework is composed of:
- Sensor data acquisition: acquisition and preprocessing of camera images,
- Visual Odometry (VO) named as *frontend*: estimate the camera movement between adjacent frames and generate a rough local map,
- Backend filtering/optimization: receives camera poses at different time stamps from Vo and results from loop closing, and then applies optimization to generate a fully optimized trajectory and map.
- Loop Closing: determines whether the robot has returned to its previous position in order to reduce the accumulated drift. If a loop is detected, it will provide information to the backend for further optimization,
- Reconstruction: task-specific map based on the estimated camera trajectory



Loop Closure: Loop closure is the process of detecting and correcting loops in the robot's trajectory.

SLAM: the Maths

- SLAM is best described in probabilistic terminology
- Let us denote time by t, and the robot location by x_t.
- For mobile robots on a flat ground, x_t is usually a three-dimensional vector, comprising its two dimensional (2-D) coordinate in the plane plus a single rotational value for its orientation.
- The sequence of locations, or path, is then given as:

$$X_T = \{x_0, x_1, x_2, \dots, x_T\}$$

  - T-- terminal time (T might be ∞).
  - x_0 refers to the initial location, serves as a point of reference for the estimation algorithm;

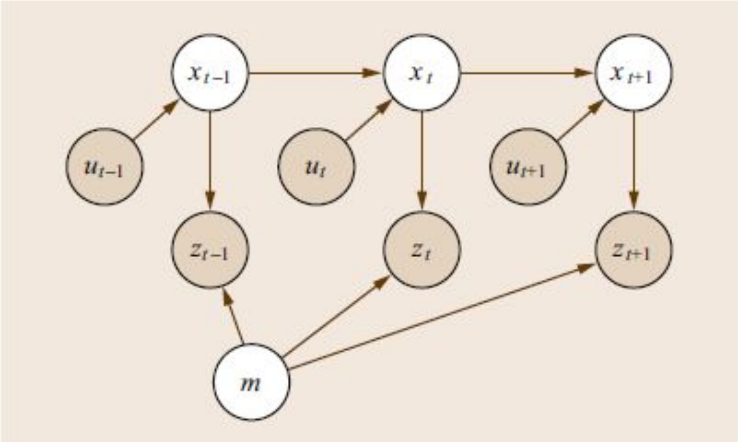
SLAM: the Maths

- Odometry provides relative information between two consecutive locations.
  - Let u_t denote the odometry that characterized the motion between time t_1 and time t
    $U_T = \{u_0, u_1, u_2, \ldots, u_T\}$----characterize the relative motion of the robot.
  - For noise free motion, U_T would be sufficient to recover the poses from the initial location x_0.
- Odometry measurements are noisy, and path integration techniques inevitably diverge from the truth.
- Robot senses objects in the environment.
  - Let *m* denote the *true* map of the environment. The environment may be comprised of landmarks, objects, surfaces, etc., and m describes their locations.
  - The environment map m is often assumed to be time-invariant, i. e., static.
  - The sequence of measurements is given as:

$$Z_T = \{z_0, z_1, z_2, \ldots, z_T\}$$

- Odometry is the process of using data from sensors to estimate changes in position and orientation over time.

SLAM: the Maths



Graphical model of the SLAM problem.

This diagram represents a graphical model for SLAM. It is useful in understanding the dependencies in the problem at hand.

It shows the sequence of locations and sensor measurements, and the causal relationships be tween these variables.

Arcs indicate causal relationships, and shaded nodes are directly observable to the robot. In SLAM, the robot seeks to recover the unobservable variables

SLAM: the Maths

- The SLAM problem is now the problem of recovering a model of the world m and the sequence of robot locations X_T from the odometry and measurement data.

- Mathematical concepts:
  - The SLAM problem can be defined as finding the robot's pose (x,y,θ) and the positions of features (landmarks) in an environment simultaneously.
  - Mathematically, it involves estimating the state X (which includes the robot's position and map features) based on observations Z and actions U.
  - State Vector:

$$X=[x_r,y_r,\theta_r,x_1,y_1,\ldots,x_n,y_n]^T$$

Where (x_r,y_r,θ_r) is the robot's position and orientation, and (x_i,y_i) are the coordinates of the landmarks.

Solutions to SLAM Problem

- There are different approaches to SLAM-based localization and navigation.

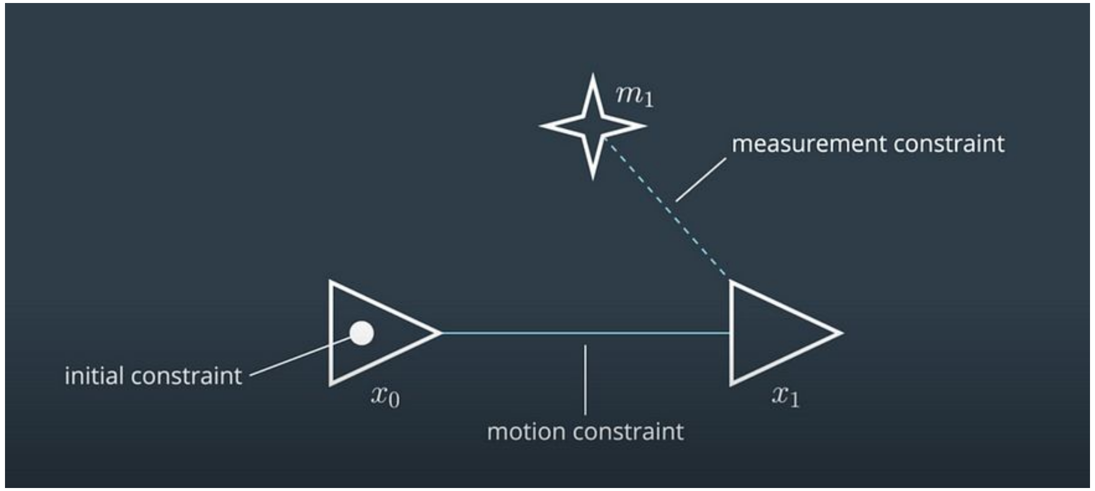
- Three different methods:
  - Graph Slam: Graph-based approaches use graph theory to model the environment
  - EKF-SLAM: Using the Extended Kalman Filter… filter-based approaches use Bayesian filters to estimate the robot’s position.
  - Particle filters: Particle filters use a particle-based representation of the environment to estimate the robot’s position.
    - Rao-Blackwellized particle filter (FastSLAM)

GraphSLAM

- Graph representation of a set of objects where pairs of objects are connected by links encoding relations between the objects
- Use a graph to represent the problem
  - Every node in the graph corresponds to a pose of the robot during mapping
  - Every edge between two nodes corresponds to a spatial constraint between them
- Graph-Based SLAM: Build the graph and find a node configuration that minimize the error introduced by the (noisy) constraints
- Solves the full SLAM problem, i.e, the algorithm recovers the entire path and map instead of just the recent pose and map.
  - Consider dependencies between current and previous poses.

GraphSLAM

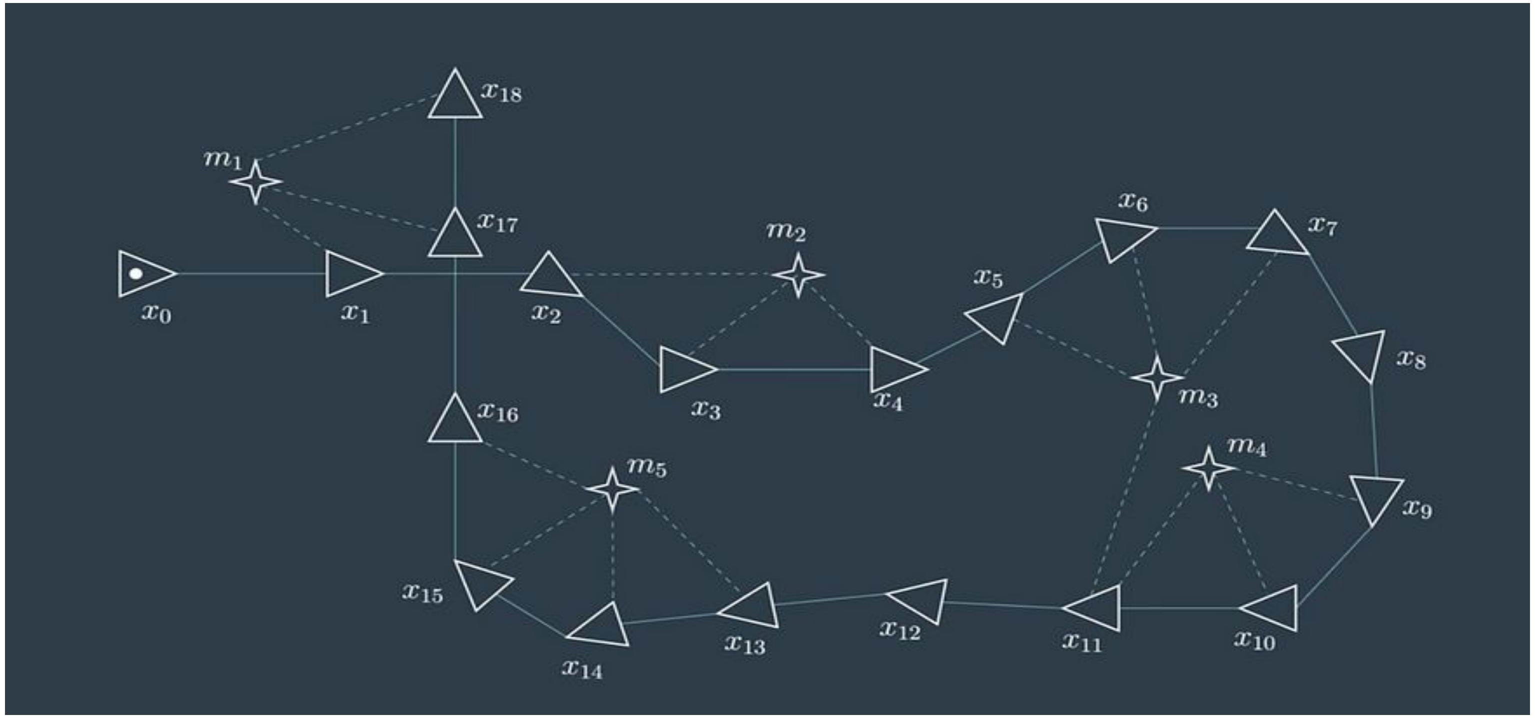
- GraphSLAM works with all of the data at once to find the optimal solution.
- How a graph in constructed:
  - Any two nodes are connected with an edge, and they are called a soft spatial constraint.
  - Soft constraints
    - Motion constraints between two successive robot poses
    - Measurement constraints between a robot pose and a feature in the environment.



- x_0 and x_1 represent robot poses.
- A solid line between the two poses called *motion constraint*.
- If the robot were to sense its environment and encounter a feature *m_1*, a soft *measurement constraint* would be added.
- Measurement constraints are represented in dashed edges.

GraphSLAM

- As the robot moves around, more and more nodes are added to the graph and over time, the graph constructed by the mobile robot becomes very large in size
- GraphSLAM is able to handle large numbers of features.
- The goal is to find the node configuration that minimizes the overall error present in all the constraints



GraphSLAM

- 1-Dimensional Graphs
    - In 1-D graphs the robot's motion and measurements were limited to one dimension —either be performed forwards or backward.

1-D Measurement constraint: $\frac{(z_t - (x_t + m_t))^2}{\sigma^2}$

1-D Motion constraint: $\frac{(x_t - (x_{t-1} + u_t))^2}{\sigma^2}$

GraphSLAM

- n-Dimensional Graphs
  - In multi-dimensional systems, matrices and covariances are used to represent the constraints.

n-D Measurement constraint: $(z_t - h(x_t, m_j))^T Q_t^{-1} (z_t - h(x_t, m_j))$

n-D Motion constraint: $(x_t - g(u_t, x_{t-1}))^T R_t^{-1} (x_t - g(u_t, x_{t-1}))$

Where *h*() and *g*() represent the measurement and motion functions, and *Qt* and *Rt* are the covariances of the measurement and motion noise.

This generalization can be applied to the system of 2-dimensions, 3-dimensions

GraphSLAM

- The multidimensional formula for the sum of all constraints is presented below.

$$J_{GraphSLAM} = x_0^T \Omega x_0 + \sum_t (x_t - g(u_t, x_{t-1}))^T R_t^{-1} (x_t - g(u_t, x_{t-1}))$$
$$+ \sum_t (z_t - h(x_t, m_j))^T Q_t^{-1} (z_t - h(x_t, m_j))$$

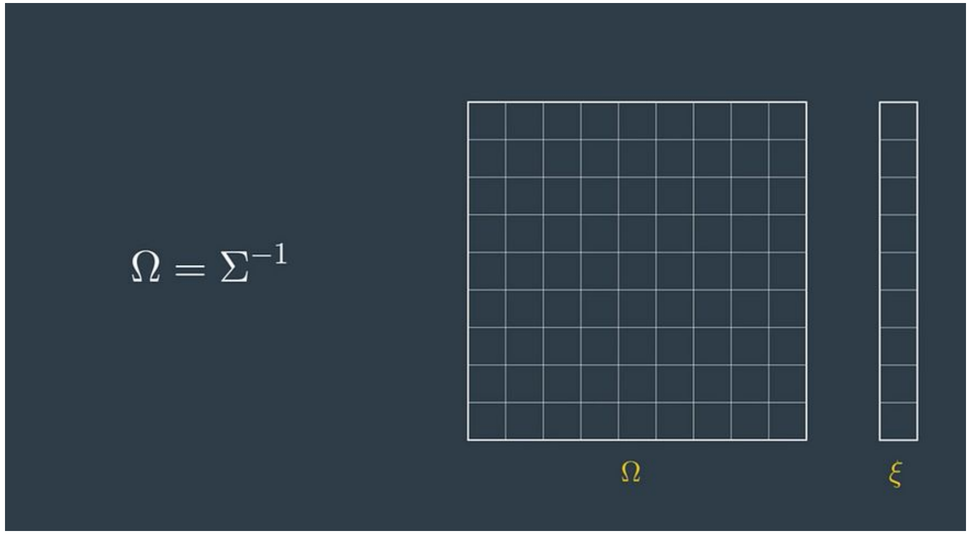
The first element in the sum is the initial constraint — it sets the first robot pose to equal to the origin of the map. The covariance, Ω_0, represents complete confidence.

$$\Omega_0 = \begin{bmatrix} \infty & 0 & 0 \\ 0 & \infty & 0 \\ 0 & 0 & \infty \end{bmatrix}$$

This generalization can be applied to the system of 2-dimensions, 3-dimensions

GraphSLAM

- Information Matrix and Information Vector
  - Are two data structures that can be used to store information from our constraints.

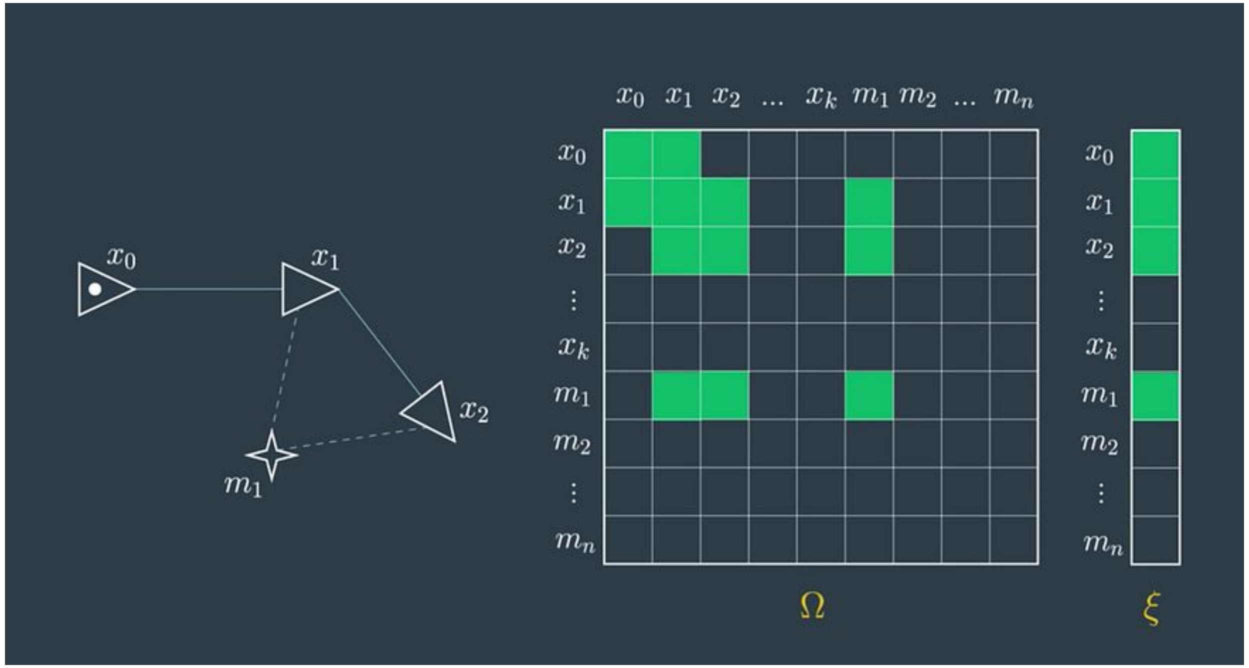


- The information matrix is represented by Ω and the information vector is represented by ξ.
- The information matrix is an inverse of a covariance matrix.

Source: A brief introduction to GraphSLAM (Shiva)

GraphSLAM

- A graph containing five constraints, one initial constraint, two motion constraints, and two measurement constraints.



- The information matrix and vector are populated successively with each constraint.
- A motion constraint, like x0→x1, will tie together two robot poses populating four cells in the matrix and two cells in the vector.
- A measurement constraint will update four cells in the matrix and two in the vector.

GraphSLAM

- Inference
  - Once the information matrix and information vector have been populated, the path and map can be recovered by the following operation:

$$\mu = \Omega^{-1}\xi$$

- The result is a vector, μ defined over all poses and features, containing the best estimate for each.

- Completing the above operation requires solving a system of equations. In small systems, this is an easily realizable task, but as the size of the graph and matrix grows — efficiency becomes a concern.
- The efficiency of this operation, specifically the matrix inversion, depends greatly on the topology of the system.
  - If the robot moves through the environment once, without ever returning to a previously visited location, then the topology is linear.
  - A more common topology is cyclical, in which a robot revisits a location that it has been to before after some time has passed.

Probabilistic SLAM

- The estimation techniques for the robot's pose and map are presented as parts of a probabilistic framework.
- Bayes Rule

$$p(a \mid b) = \frac{p(b \mid a)\; p(a)}{p(b)}$$

$$p(a \mid b, c) = \frac{p(b \mid a, c)\; p(a \mid c)}{p(b \mid c)}$$

- Law of Total Probability

$$p(a) = \sum_i p(a \wedge b_i)$$

Discrete

$$= \sum_i p(a \mid b_i)\, p(b_i)$$

Continuous

$$p(a) = \int p(a \mid b)\, p(b)\, db$$

it follows that:

$$p(a \mid b) = \int p(a \mid b, c)\, p(c \mid b)\, dc$$

Source: https://www.cs.cmu.edu/~motionplanning/lecture/Chap9-Bayesian-Mapping_howie.pdf

Probabilistic SLAM

- In probabilistic form, the Simultaneous Localisation and Map Building (SLAM) problem requires that the probability distribution

$$P(\mathbf{x}_k, \mathbf{m} \mid \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{x}_0)$$

The full SLAM posterior represents the probability of the entire trajectory and map given all sensor measurements (Zo:k) and motion commands (Uo:k)

be computed for all times k.

- This probability distribution describes the joint posterior density of the landmark locations and vehicle state (at time k) given the recorded observations and control inputs up to and including time k together with the initial state of the vehicle.
- A recursive solution to the SLAM problem is desirable.

Probabilistic SLAM

- Starting with an estimate for the distribution $P(x_{k-1}, m \mid Z_{0:k-1}, U_{0:k-1})$ at time k-1, the joint posterior, following a control u_k and observation z_k, is computed using Bayes Theorem.
- This computation requires that a state transition model and an observation model are defined describing the effect of the control input and observation respectively.
- Observation model: describes the probability of making an observation z_k when the vehicle location and landmark locations are known.

$$P(\mathbf{z}_k \mid \mathbf{x}_k, \mathbf{m})$$

- Motion model for the vehicle can be described in terms of a probability distribution on state transitions in the form

$$P(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{u}_k)$$

That is, the state transition is assumed to be a Markov process in which the next state x_k depends only on the immediately proceeding state $x_{k\ 1}$ and the applied control uk, and is independent of both the observations and the map.

Deep Learning Based SLAM

- Visual Simultaneous Localization and Mapping (VSLAM) has been a hot topic of research since the 1990s
- Deep learning has yielded promising results for VSLAM applications such as autonomous driving and navigation, service robots, virtual and augmented reality, and pose estimation.
- VSLAM methods based on classical image processing algorithms consists of six main steps,
  - Initialization (data acquisition), feature extraction, feature matching, pose estimation, map construction, and loop closure
- Three ways are developing with varying degrees of integration of deep learning into traditional VSLAM systems:
  - Adding auxiliary modules based on deep learning,
  - Replacing the original modules of traditional VSLAM with deep learning modules, and
  - Replacing the traditional VSLAM system with end-to-end deep neural networks

Challenges of SLAM

- At a theoretical and conceptual level, SLAM can now be considered a solved problem.
- However, substantial issues remain in practically realizing more general SLAM solutions and notably in building and using perceptually rich maps as part of a SLAM algorithm.

Software

Table 1: Open Source SLAM Software		
Author	**Description**	**Link**
Kai Arras	The *CAS Robot Navigation Toolbox*, a MATLAB simulation toolbox for robot localization and mapping.	`www.cas.kth.se/toolbox/index.html`
Tim Bailey	MATLAB simulators for EKF-SLAM, UKF-SLAM, and FastSLAM 1.0 and 2.0.	`www.acfr.usyd.edu.au/homepages/academic/tbailey/software/index.html`
Mark Paskin	Java library with several SLAM variants, including Kalman filter, information filter, and thin junction tree forms. Includes a MATLAB interface.	`www.stanford.edu/~paskin/slam/`
Andrew Davison	*Scene*, a C++ library for map-building and localisation. Facilitates real-time single camera SLAM.	`www.doc.ic.ac.uk/~ajd/Scene/index.html`
José Neira	MATLAB EKF-SLAM simulator that demonstrates *joint compatibility branch-and-bound* data association.	`http://webdiis.unizar.es/~neira/software/slam/slamsim.htm`
Dirk Hähnel	C language grid-based version of FastSLAM.	`www.informatik.uni-freiburg.de/~haehnel/old/download.html`
Various	MATLAB code from the 2002 SLAM summer school.	`www.cas.kth.se/slam/toc.html`

Dataset

Table 2: Online Datasets		
Author	**Description**	**Link**
Jose Guivant, Juan Nieto and Eduardo Nebot	Numerous large-scale outdoor datasets, notably the popular Victoria Park data.	www.acfr.usyd.edu.au/homepages/academic/enebot/dataset.htm
Chieh-Chih Wang	Three large-scale outdoor datasets collected by the Navlab11 testbed.	www.cs.cmu.edu/~bobwang/datasets.html
Radish (The Robotics Data Set Repository)	Many and varied indoor datasets, including large-area data from the CSU Stanislaus library, the Intel Research Lab in Seattle, the Edmonton Convention Centre, and more.	http://radish.sourceforge.net/
IJRR (The International Journal of Robotics Research)	IJRR maintains a webpage for each article, often containing data and video of results. A good example is a paper by Bosse *et al.* [3], which has data from Killian Court at MIT.	www.ijrr.org/contents/23_12/abstract/1113.html