WHAT??
WHY??
HOW??

SLAM (Simultaneous Localization and Mapping)

WHAT??

SLAM (simultaneous localization and mapping) is a method used for autonomous vehicles that lets you build a map and localize your vehicle in that map at the same time. SLAM algorithms allow the vehicle to map out unknown paths or environments.

The map information is used to carry out tasks such as path planning and obstacle avoidance.

WHY??

SLAM has been the subject of technical research for many years. But with vast improvements in computer processing speed and the availability of low-cost sensors such as cameras and laser range finders, SLAM is now used for practical applications in a growing number of fields.

SLAM is useful in many other applications such as navigating a fleet of mobile robots to arrange shelves in a warehouse, parking a self-driving car in an empty spot, or delivering a package by navigating a drone in an unknown environment or locations.

EXAMPLE OF SLAM IN REAL LIFE

If we consider a home robot vacuum.

Without SLAM: it will just move randomly within a room and may not be able to clean the entire floor surface. In addition, this approach uses more power, so the battery will run out more quickly.

With SLAM: It can use information such as the number of wheel revolutions and data from cameras and other imaging sensors to determine the amount of movement needed. This is called localization. The robot can also simultaneously use the camera and other sensors to create a map of the obstacles in its surroundings and avoid cleaning the same area twice. This is called mapping.
Thus uses SLAM for better efficiency.

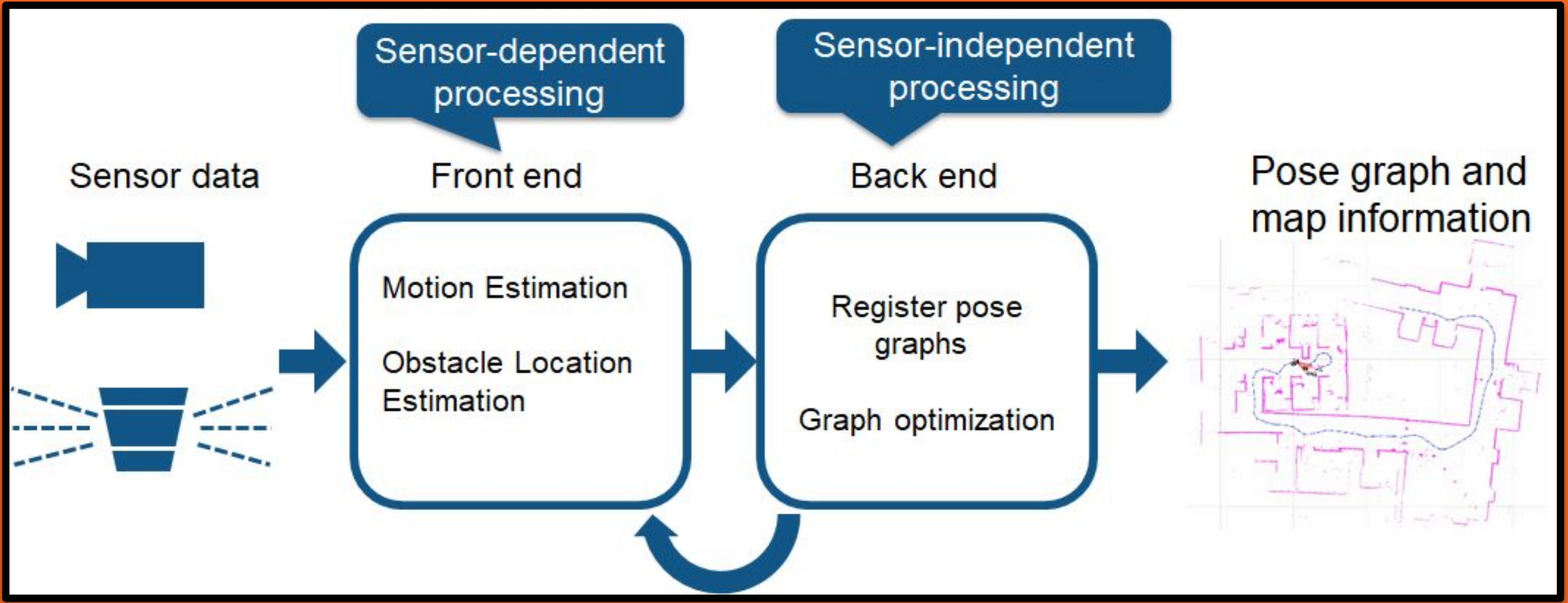
HOW SLAM Works??

There are two types of technology components used to achieve SLAM.

The first type is sensor signal processing, including the front-end processing, which is largely dependent on the sensors used.

The second type is pose-graph optimization, including the back-end processing.

HOW SLAM Works??



Different methods of SLAM

1. **VISUAL SLAM:**

    Visual SLAM uses images acquired from cameras and other image sensors. Visual SLAM can be implemented at low cost with inexpensive cameras. In addition, since cameras provide a large volume of information, they can be used to detect landmarks.

2. **LiDAR SLAM**:

    Light detection and ranging is a method that primarily uses a laser sensor or distance sensor. Lasers are significantly more precise, and are used for applications with high-speed moving vehicles such as self-driving cars and drones.

Common Challenges with SLAM and how to avoid ?

Challenge:

Localization errors accumulate, causing substantial deviation from actual values

How to avoid:

One way to avoid is to remember some characteristics from a previously visited place as a landmark and minimize the localization error. Pose graphs are constructed to help correct the errors. By solving error minimization as an optimization problem, more accurate map data can be generated.

Common Challenges with SLAM and how to avoid

Challenge:

High computational cost for image processing, point cloud processing, and optimization

How to avoid:

One way to solve this is to run different processes in parallel. Processes such as feature extraction, which is preprocessing of the matching process, is relatively suitable for parallelization.

SLAM FOR TURTLEBOT3

The SLAM is a well-known feature of TurtleBot3 from its predecessors.

Gmapping is used as a default SLAM method for turtlebot3.
The gmapping package provides laser-based SLAM, as a ROS node called slam_gmapping. Using slam_gmapping, we can create a 2-D grid map from laser and data collected by a mobile robot.

TurtleBot3 will be exploring unknown area of the map using teleoperation.
It is important to avoid vigorous movements such as changing the linear and angular speed too quickly.

Few parameters which are a part of gmapping package and we are going to use them are:

1. **maxUrange**: This parameter is set the maximum usable range of the lidar sensor.
2. **minimumScore:** This can reduce errors in the expected position of the robot in a large area.
3. **linearUpdate:** When the robot translates longer distance than this value, it will run the scan process.
4. **angularUpdate:** When the robot rotates more than this value, it will run the scan process.

MAPPING in TURTLEBOT3

The map is drawn based on the robot’s scan information. These map data is drawn as the TurtleBot3 travels. After creating a complete map of desired area the map can be saved to the local drive for the later use.

The map uses two-dimensional Occupancy Grid Map (OGM). The saved map will have different coloured areas where white area is collision free area while black area is occupied and inaccessible area, and gray area represents the unknown area. This map is used for the Navigation.