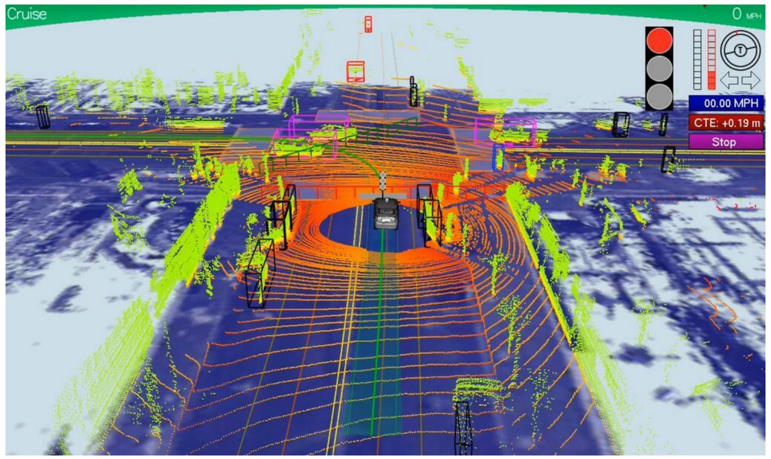
Localization and Bayesian Filters

Nacho Sañudo
University of Modena and Reggio Emilia
Ignacio.sanudoolmedo@unimore.it

Credits to Udacity self-driving course

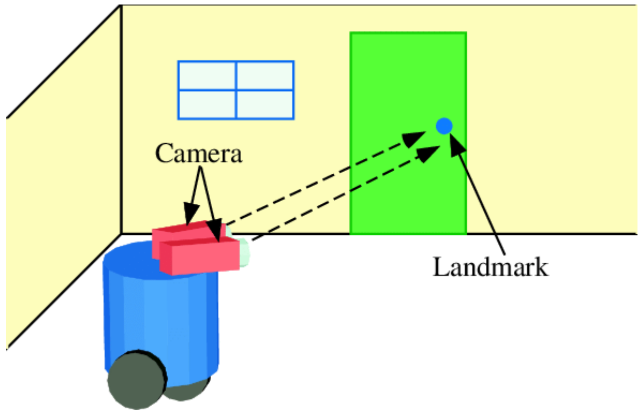
Localization

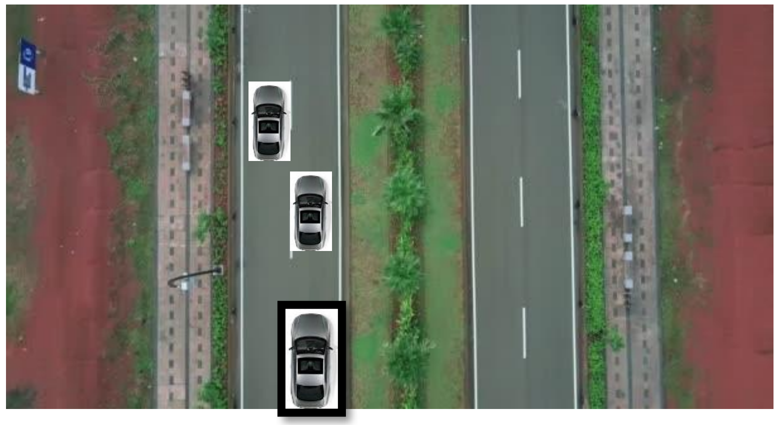
- Let’s suppose that we have a car driving in a highway
- The traditional way to localize the vehicle is using GPS
  - Unfortunately the GPS is not accurate enough
    - 2-10 meters of error
- Localization answers this question:
  - Where am I?
    - With an accuracy of 10cm



Localization

- To localize the vehicle/robot, the we can use "world" features
  - Landmark-based localization, is the most common method used
    - identify landmarks and measure the vehicle's distance from each, to estimate the ego vehicle position
    - lanes, traffic lights, signals, etc…

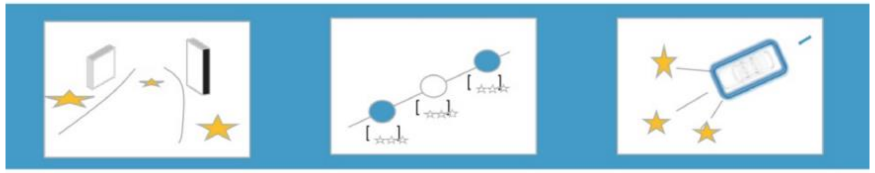




[1] Levinson, J., & Thrun, S. (2010). Robust vehicle localization in urban environments using probabilistic maps. In 2010 IEEE International Conference on Robotics and Automation (pp. 4372–4378). IEEE.

Localization

- The localization software component is in charge of estimating the ego vehicle pose (position and orientation) relative to a map or road
  - **LIDAR-Based Localization**
    - Extract road features and compare to the registered map
    - We will implement [1]
  - **Camera-Based Localization**
    - Based on extracting visual odometry and road maps
  - **LIDAR plus Camera-Based Localization**
    - LIDAR data is used to build a map, and camera data to estimate the localization of the self-driving car relative to the map
    - Most of these methods matches stereo images to 3D point-cloud maps
  - **GPS is also used!**



1. **Landmarks** for localization are defined.

2. A **map** is created with such localization landmarks.

3. A **camera** detects landmarks. Those are matched with landmark maps to compute the **actual pose** of the vehicle.

Source: Daimler

Google VS Tesla (localization)

- **Google uses LiDARs**
  - They rely on all kind of road features (lane lines, curbs, signals...) and prior maps
    - HD maps
    - Expensive and weak to climate changes or road changes
- Tesla uses vision
  - They mostly rely on lane lines + features
    - They also use maps
      - Less detailed since they use cameras+radars
      - Cheap but inaccurate

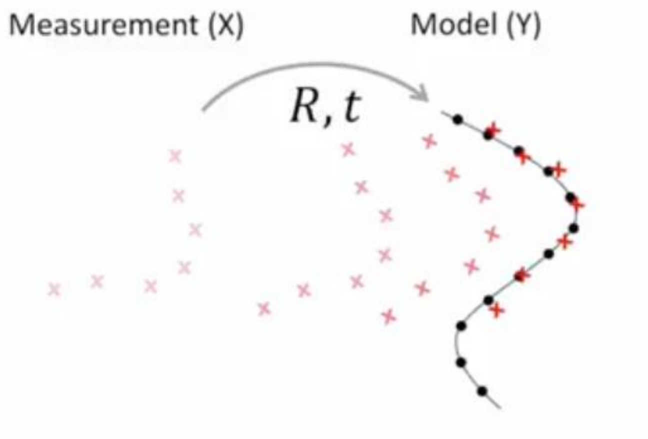
Localization

- Localization is what allows an autonomous car to know precisely where it is
- **Traditional way**: use global navigation satellite (GPS) to find the car with respect to the map → GPS is not precise enough
  - GPS, 2 to 10 meters of error, **we need something like 2 to 10 CM of error** → you can't trust on GPS
  - "the GPS signal cannot be guaranteed in occluded areas, such as under trees, in urban canyons (roads surrounded by large buildings) or in tunnels"
- **Common practice**: use onboard sensor data (lidar, radar) with our global map to measure distance to static obstacles, then you map the obstacles in the global map
  - **To estimate where the car is in the map you have to match the observations with the map information, i.e, transformation between local car coordinate system and global coordinate system of the map**

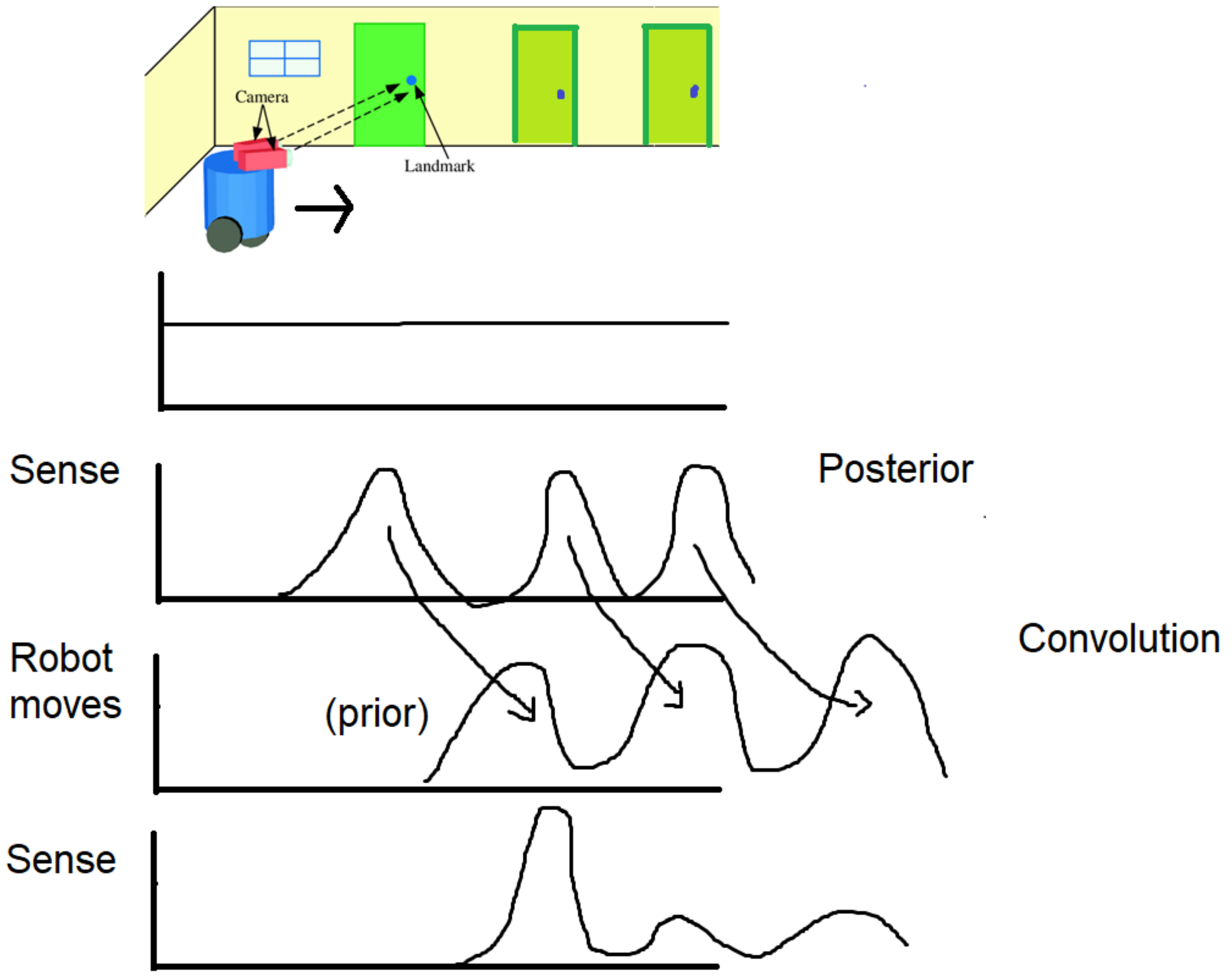
Self-Driving Cars: A Survey. Claudine Badue et al.

Mapping

- Maps are used to localize the vehicle with high precision
- Cars containing all the sensors are used to map the **features** of the environment
  - High resolution LiDARs, radar, GPS, IMUs, cameras…
  - HD maps
    - Point cloud: “estimate the transformation to move one cloud so that it is aligned with the other one”
      - ICP, LOAM, GICP…
- A “cheaper” vehicle can used later to localize the vehicle
  - comparing the features of the pre-computed maps with the current sensor measurements



Localization (basics)



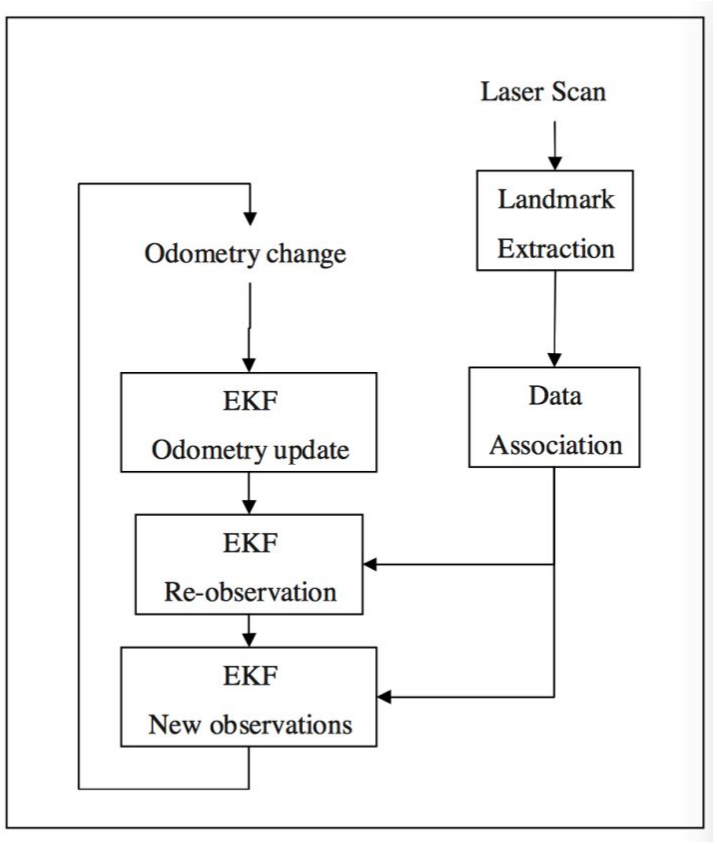
Localization techniques

- **Odometry** — This first technique, odometry, uses **a starting position** and **a wheel displacement calculation** to estimate a position at a time t. This technique is generally very inaccurate and leads to an accumulation of errors due to measurement inaccuracies, wheel slip, ...
- **Kalman filter** —this technique is used to estimate the state of the vehicles around us. We can also implement this to **define the state of our own vehicle**.
- **Particle Filter —** The Bayesian filters can also have a variant called particle filters. This technique compares the observations of our sensors with the environmental map. **We then create particles around areas where the observations are similar to the map.**
- **SLAM** — A very popular technique if we also want to estimate the map exists. It is called SLAM (Simultaneous Localization And Mapping). In this technique, we estimate **our position** but **also the position of landmarks**. A traffic light can be a landmark
  - The absence of an initial map in the SLAM problem makes it impossible to localize the robot during mapping using algorithms like MCL

Source: https://towardsdatascience.com/self-driving-car-localization-f800d4d8da49

EKF-SLAM

- SLAM (Simultaneous Localization And Mapping) is a technique used to build a map while at the same time estimating the position of the vehicle
  - use the environment to update the position of the robot
- EKF-SLAM: EKF used in SLAM
  1. State prediction
  2. Measurement prediction
  3. Measurement
  4. Data association
  5. Update



Particle filter

- **Particle filter**
  - technique that uses a set of **particles** to represent the posterior distribution of some stochastic process given noisy and/or partial observations
  - A particle represents a discrete guess of where the robot might be
  - Estimate the state of an object with a given observations
- Algorithm composed of 4 steps
  - Initialize
  - Prediction
  - Update
  - Resampling