Robot Teknolojisine Giriş BLM4830

Öğr. Grv. Furkan ÇAKMAK

Ders Tanıtım Formu ve Konular

BLM4830 Robot Teknolojisine Giriş Hafta 4

Hafta	Tarih	Konular
1	2.03.2022	Ders Tanıtımı, ROS ve Platform Tanıtımı, Robot Çeşitleri ve Robotik Konuları Başlangıcı
2	9.03.2022	Kinematik - Genel Tanımlar - Diferansiyel Sürüşlü Robot İçin Hesaplama Örnekleri
3	16.03.2022	Sensörler - Çeşitleri ve Çalışma Sistematikleri ve Uygulamaları
4	23.03.2022	Odometri ve Lokalizasyon Kavramları
5	30.03.2022	Uygulama 1 (Laboratuvar)
6	6.04.2022	Haritalama Yöntemleri ve Uygulamaları
7	13.04.2022	Navigasyon ve Keşif Yaklaşımları ve Uygulamaları (Ödev Teslimi)
8	20.04.2022	Ara Sınav
9	27.04.2022	Uygulama 2 (Laboratuvar)
10	4.05.2022	Tatil - Ramazan Bayramı Arifesi
11	11.05.2022	Robot Üzerinden Görüntü İşleme Teknikleri
12	18.05.2022	Robot Üzerinden Görüntü İşleme Teknikleri (Devam)
13	25.05.2022	3B Haritalama Yöntemleri
14	1.06.2022	Proje Sunumları

Localization:

Determination of the pose (= position + orientation) of a mobile robot in a known environment in order to succesfully perform a given task

Localization as an estimation problem

The robot must infer its pose from available data

Data (noisy):

- **Motion information:**

  → Proprioceptive sensors (e.g. encoders, accelerometers, etc.)

- **Environment Measurements**

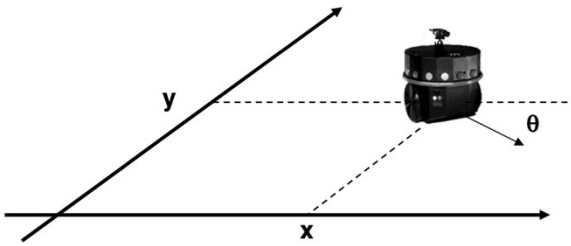
  → Exteroceptive sensors (e.g. laser, sonar, IR, GPS, camera, RFID, etc.)

A filtering approach is required to fuse all information

Notation

$$x_r = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

Robot pose



$x_{r,0:t} = \{x_{r,0}, x_{r,1}, \ldots, x_{r,t}\}$ Robot poses from time 0 to time t

$z_{1:t} = \{z_1, z_2, \ldots, z_t\}$ Robot exteroceptive measurements from time 1 to time t

$u_{0:t} = \{u_0, u_1, \ldots, u_t\}$ Motion commands (or proprioceptive measurements) from time 0 to time t

Belief of the robot at time t: probability density function (pdf) describing the information the robot has regarding its pose at time t, based on all available data (exteroceptive measurements and motion commands):

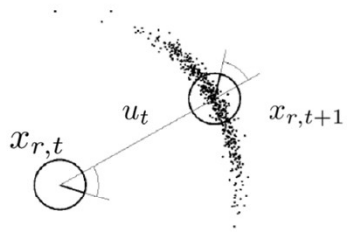
$$bel_t(x_r) = p(x_{r,t} = x_r | z_{1:t}, u_{0:t-1})$$

Prior belief of the robot at time t: pdf before acquiring the last measurement z_t:

$$\overline{bel}_t(x_r) = p(x_{r,t} = x_r | z_{1:t-1}, u_{0:t-1})$$

The robot motion model is the pdf of the robot pose at time t+1 given the robot pose and the motion action at time t. It takes into account the noise characterizing the proprioceptive sensors:

$$p(x_{r,t+1}|x_{r,t}, u_t)$$



The measurement model describes the probability of observing at time t a given measurement z_t when the robot pose is $x_{r,t}$. It takes into account the noise characterizing the exteroceptive sensors:

$$p(z_t|x_{r,t})$$

Robot Localization Algorithms

Genel olarak lokalizasyon algoritmaları Bayes Filtresine dayanıyor:

1. Adım Tahmin (t anında x state inde iken u kontrol işareti uygulandığında t+1 anında yeni state olasılığı)

Motion model

$$\overline{bel}_t(x_r) = \int_\Omega \boxed{p(x_r|x_{r,t-1} = y, u_{t-1})} bel_{t-1}(y)dy$$

2. Adım Güncelleme (t anında state x iken bir z ölçümünü alma olasılığı)

Measurement model

$$bel_t(x_r) = \eta \boxed{p(z_t|x_{r,t} = x_r)} \overline{bel}_t(x_r)$$

Typical Motion Models

- In practice, one often finds two types of motion models:
  - **Odometry-based**
  - **Velocity-based (dead reckoning)**
- Odometry-based models are used when systems are equipped with wheel encoders.
- Velocity-based models have to be applied when no wheel encoders are given.
- They calculate the new pose based on the velocities and the time elapsed.

Dead Reckoning

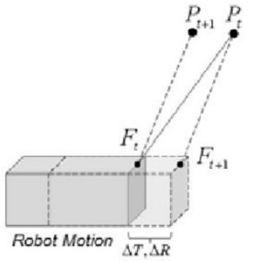
- Derived from "deduced reckoning."
- Mathematical procedure for determining the present location of a vehicle.
- Achieved by calculating the current pose of the vehicle based on its velocities and the time elapsed.

- Odometry tends to be more accurate than velocity model,
- But, Odometry is only available after executing a motion command, cannot be used for motion planning

Wheel Odometry

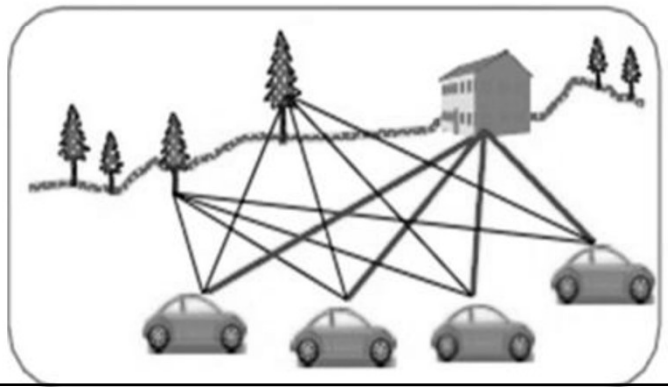
- **Odometry** is the use of data from motion sensors to estimate change in position over time. → Calculate the resulting robot position and orientation from wheel encoder measurements (Use the forward kinematics equations)

Visual Odometry

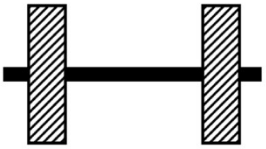


The visual odometry concept. The vehicle ego-motion is estimated from the apparent motion of the features in the image space

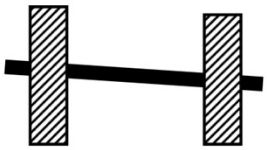
** LSM laser scan matching kamera yerine laser verilerini ilişkilendirme



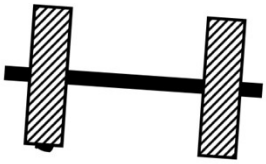
Possible sources of noise



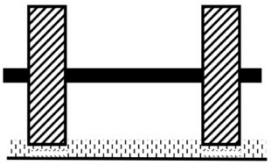
Ideal case



Different wheel diameters (systematic source of noise)



Bump (casual source of noise)



Carpet (casual source of noise)

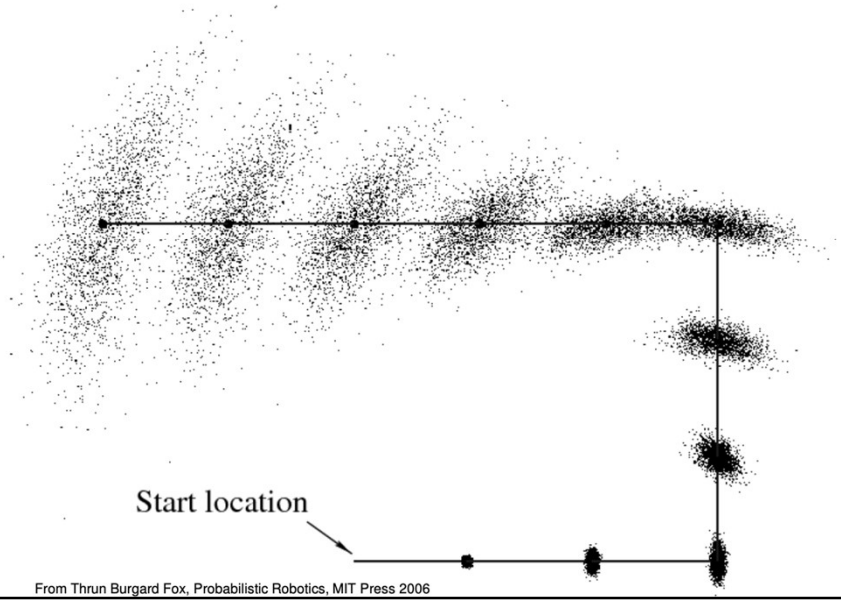
and many more (e.g. distance between the wheels) …

Deterministic (systematic) errors can be eliminated through proper **calibration**

Non-deterministic errors have to be **described by error models** and will always lead to uncertain position estimate.

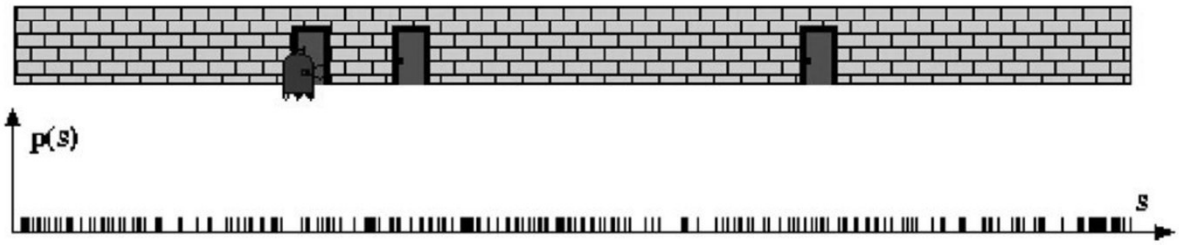
Errors are integrated, unbounded !!

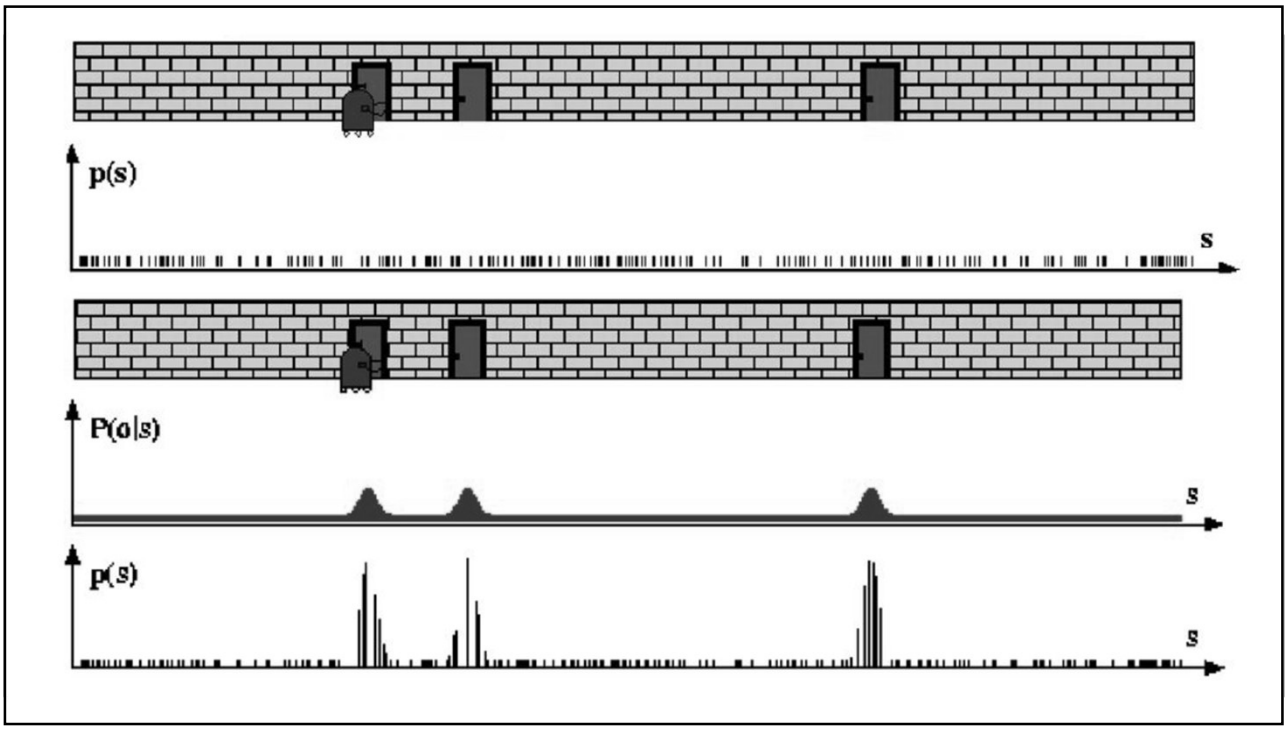
Accumulation of the pose estimation error under the robot motion
(only proprioceptive measurements)

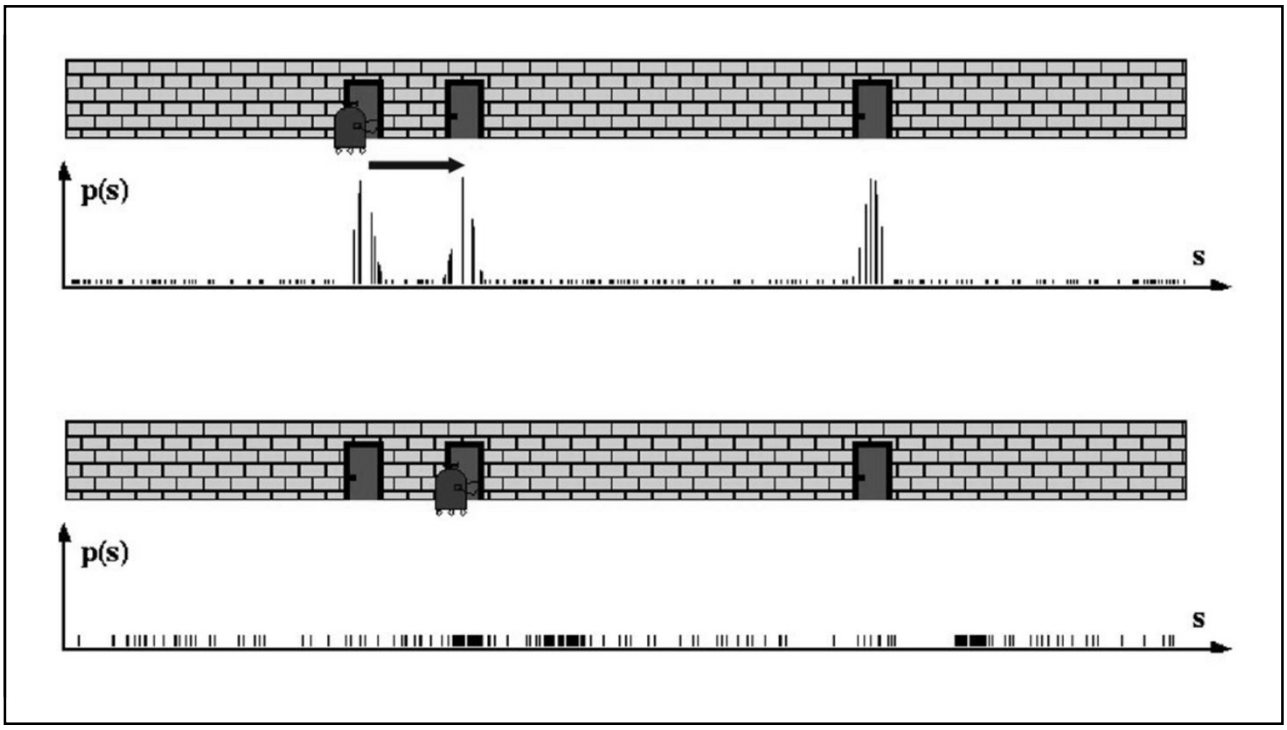


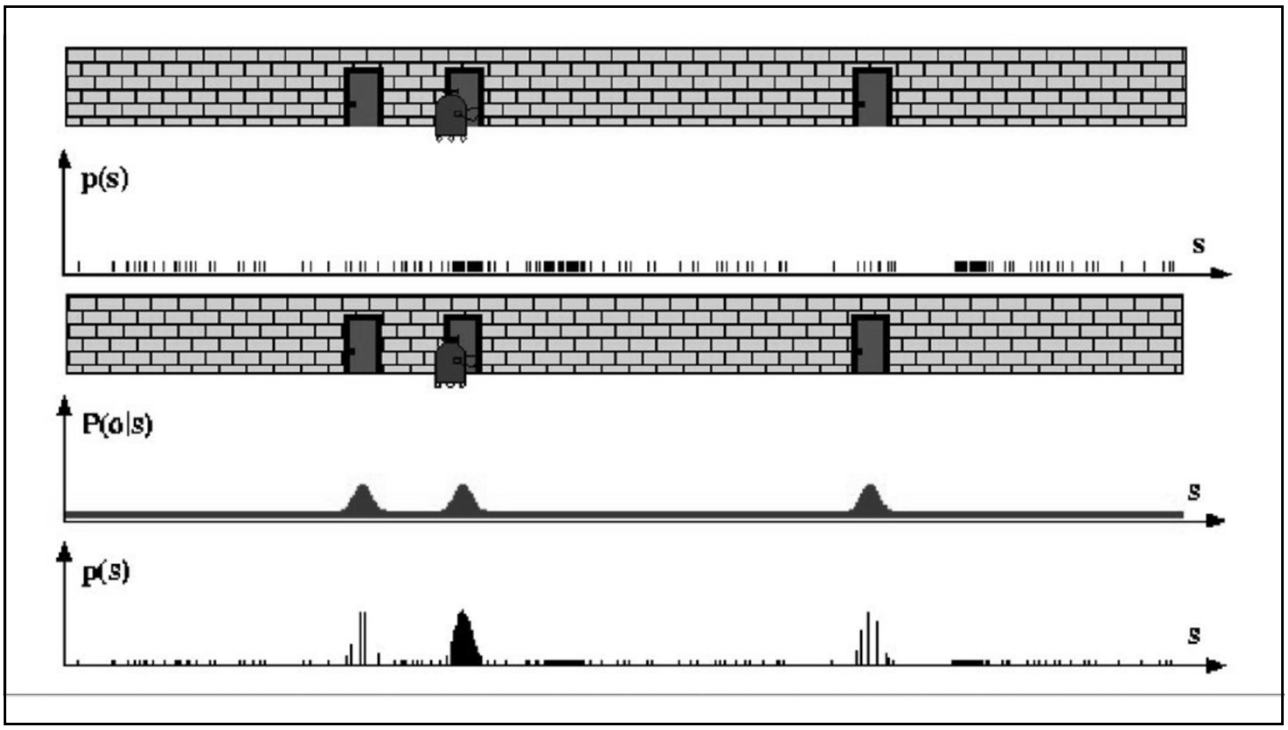
From Thrun Burgard Fox, Probabilistic Robotics, MIT Press 2006

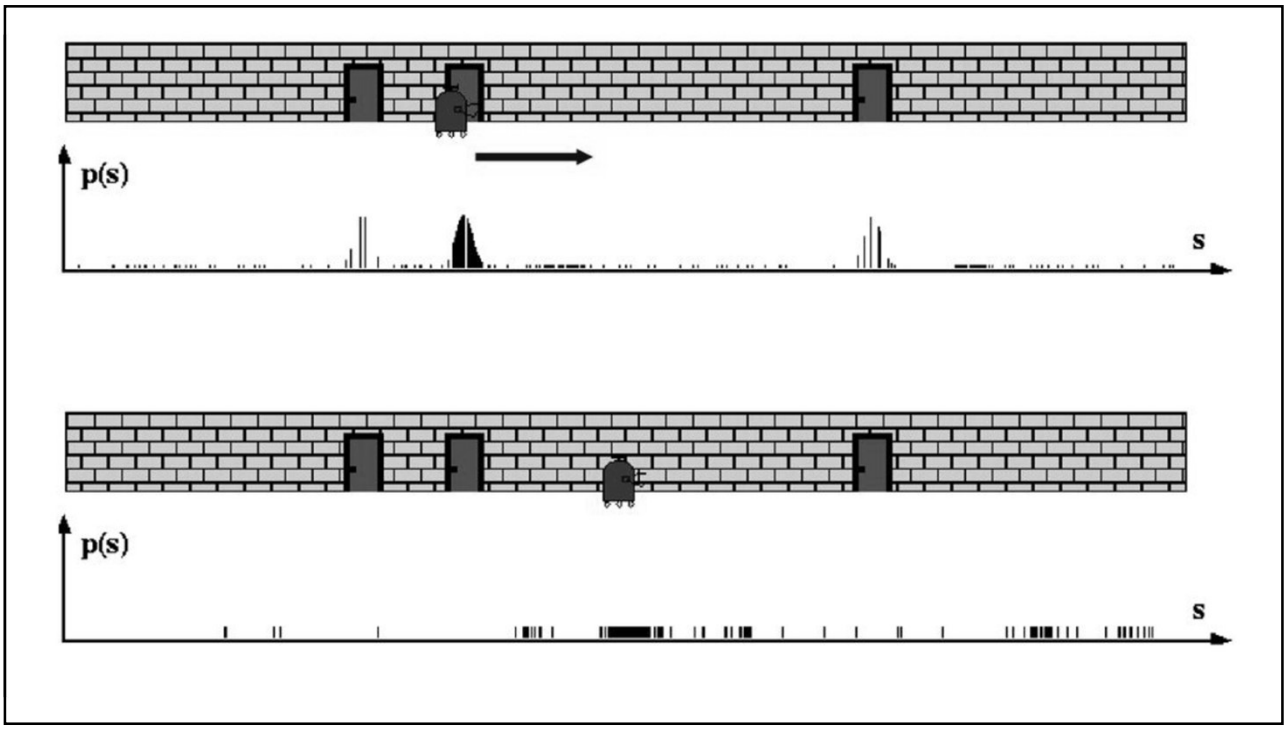
Particle Filters

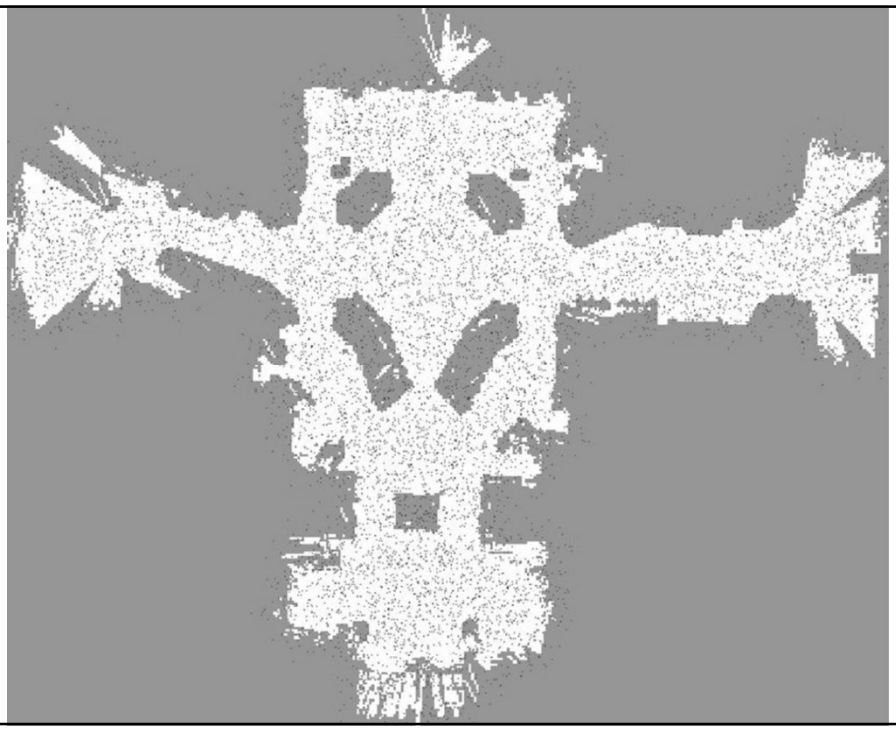


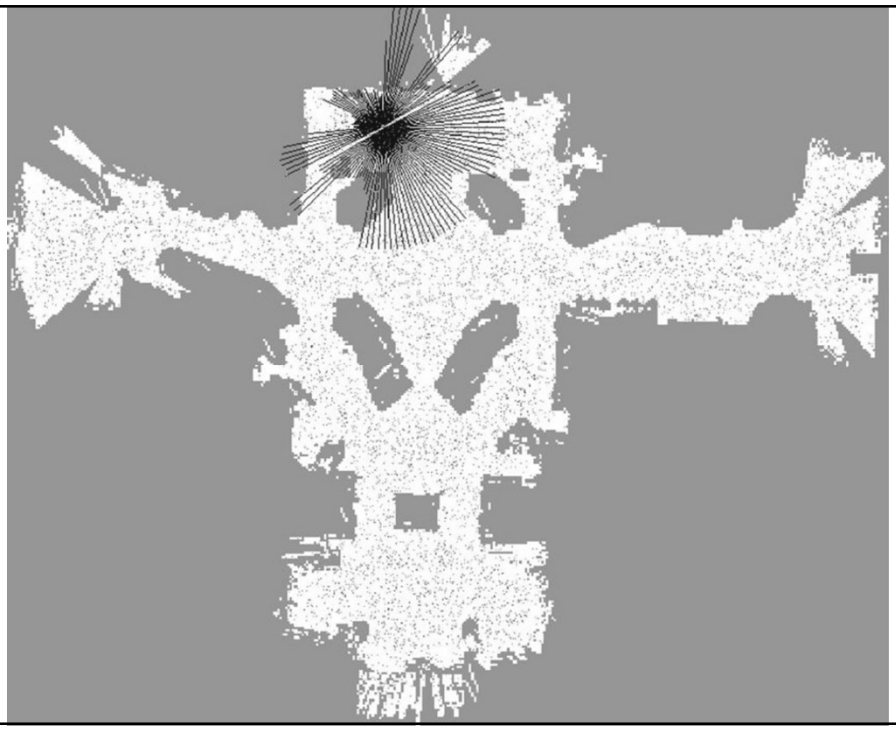

p(s)
s
P(o|s)
s
p(s)
s

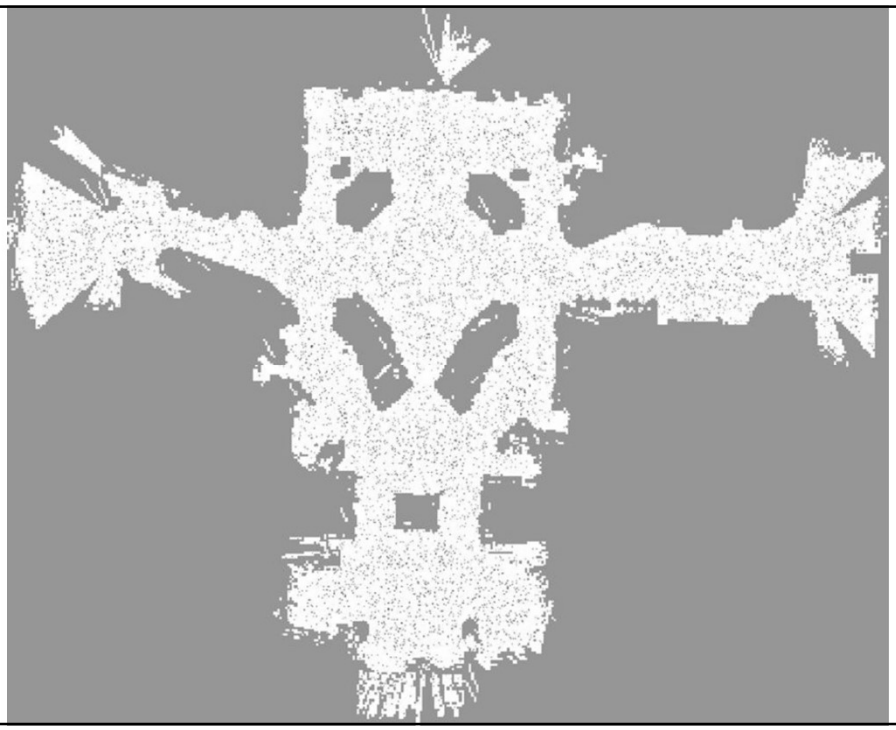

p(s)
s
p(s)
s

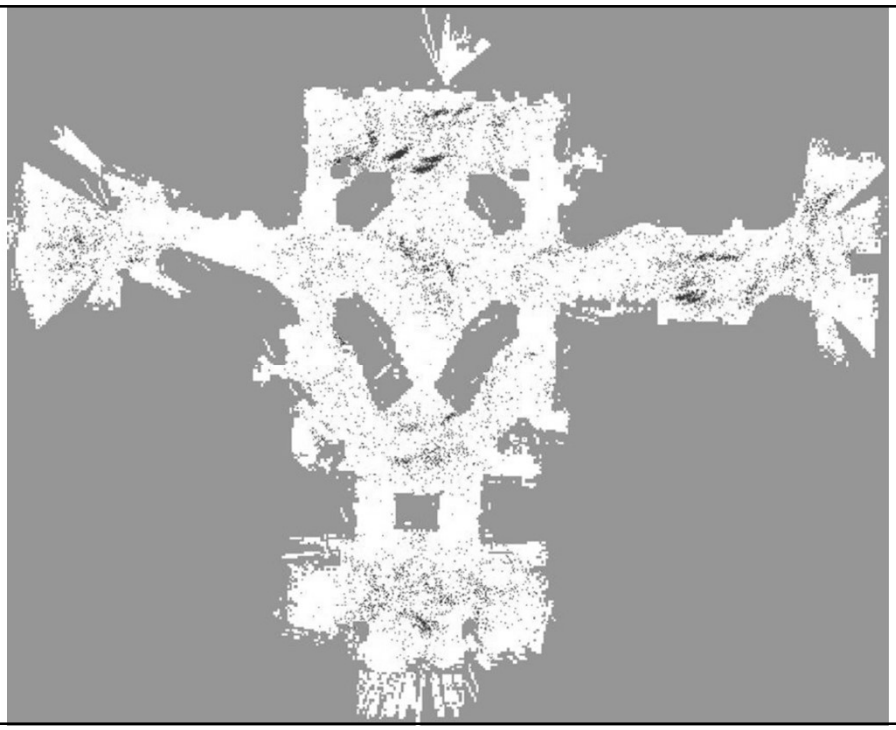

p(s)
s
P(o|s)
s
p(s)
s

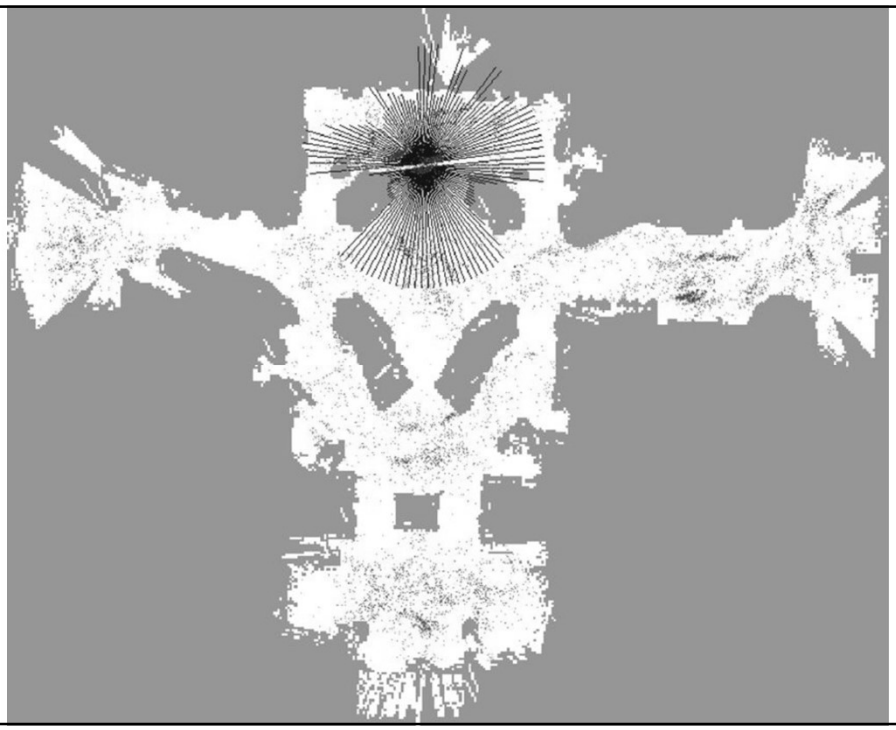

p(s)
s
p(s)
s

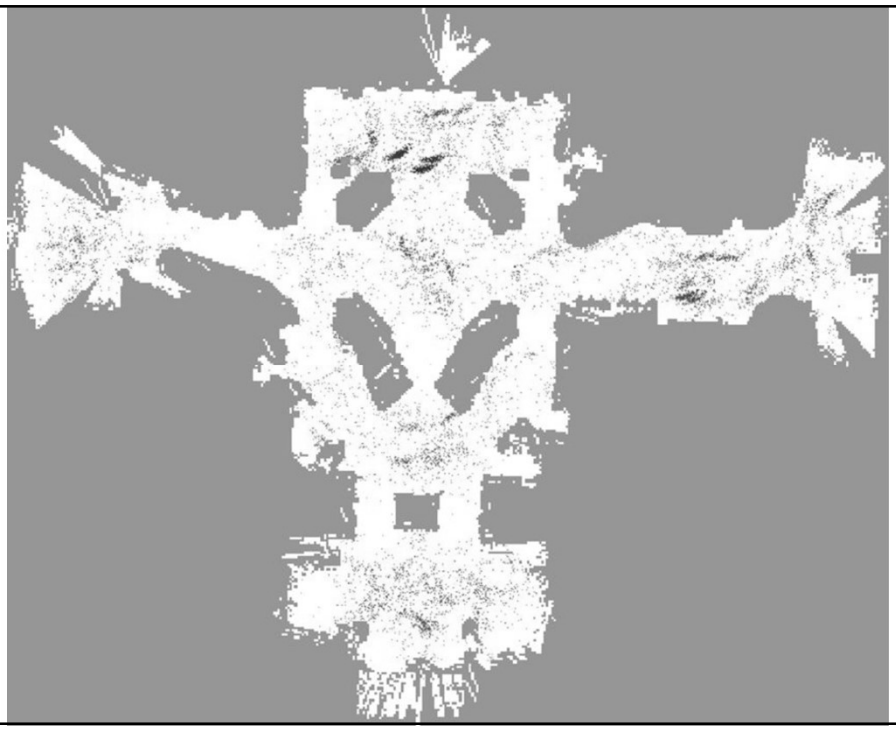


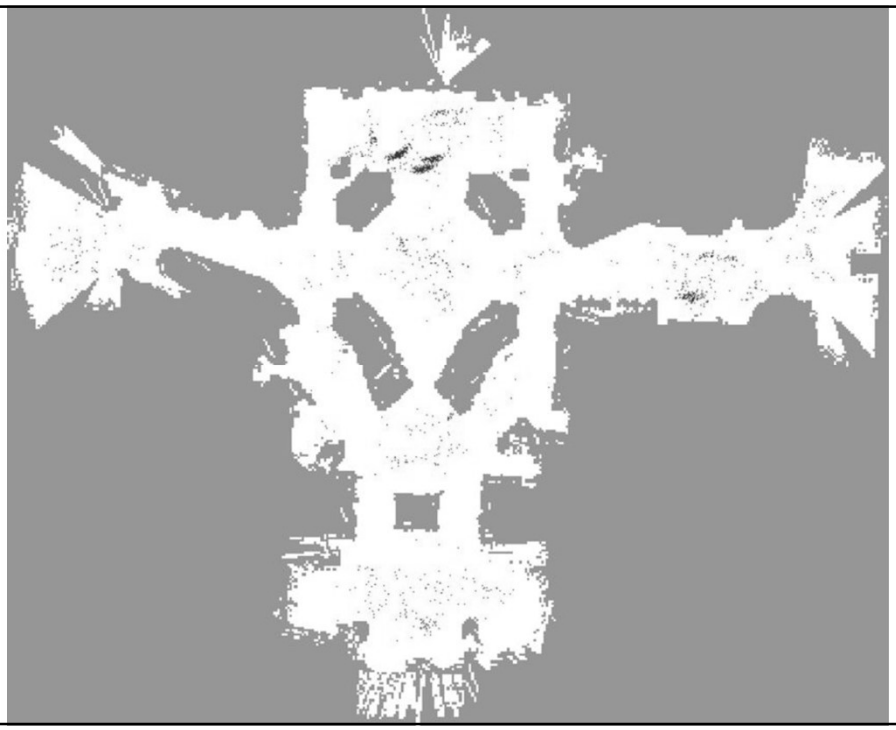


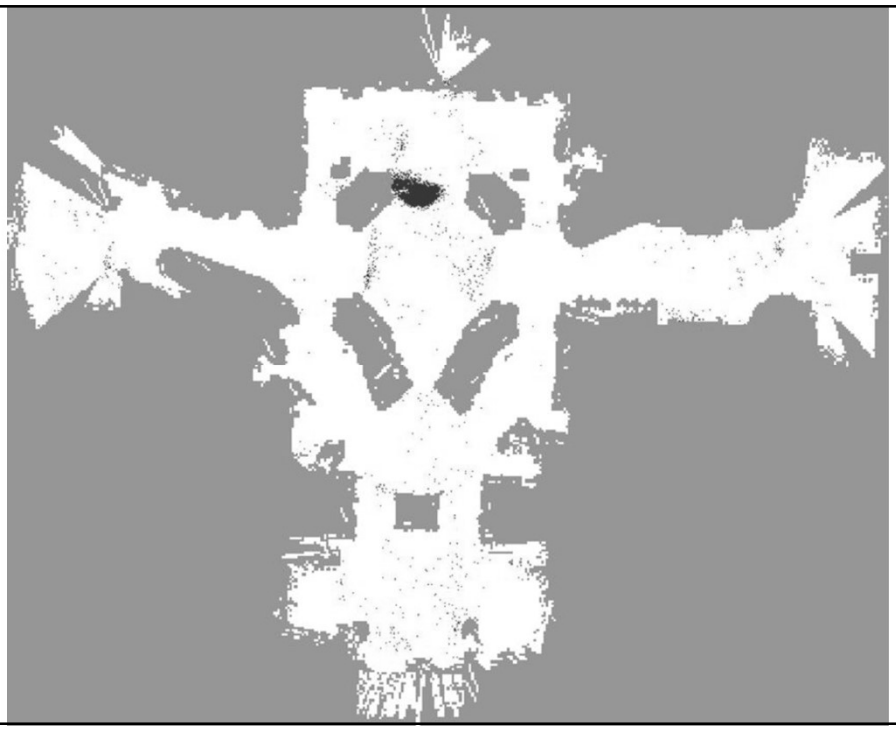


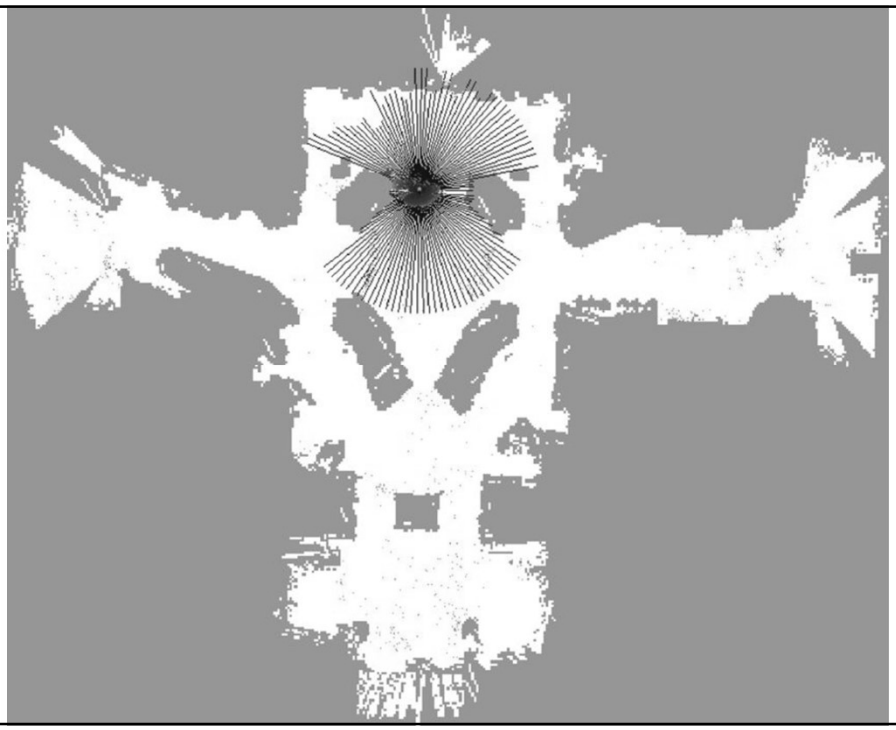


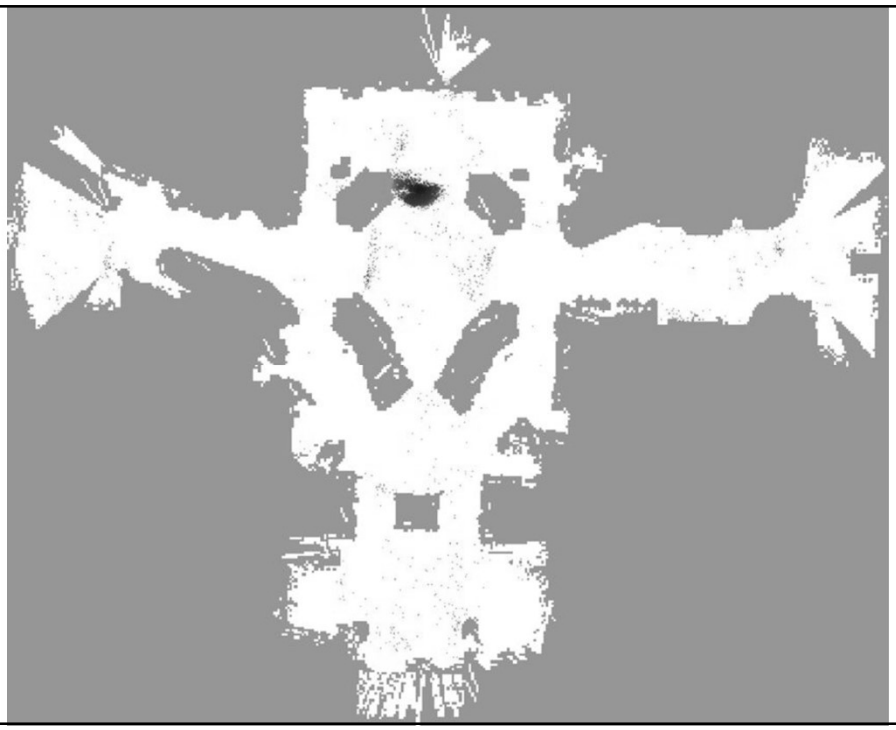


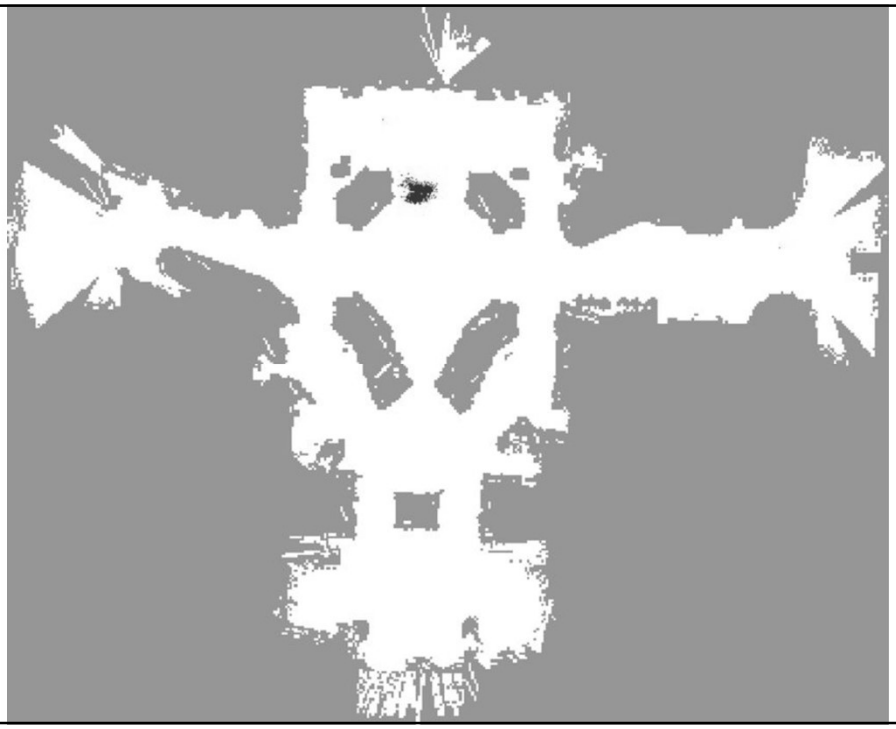


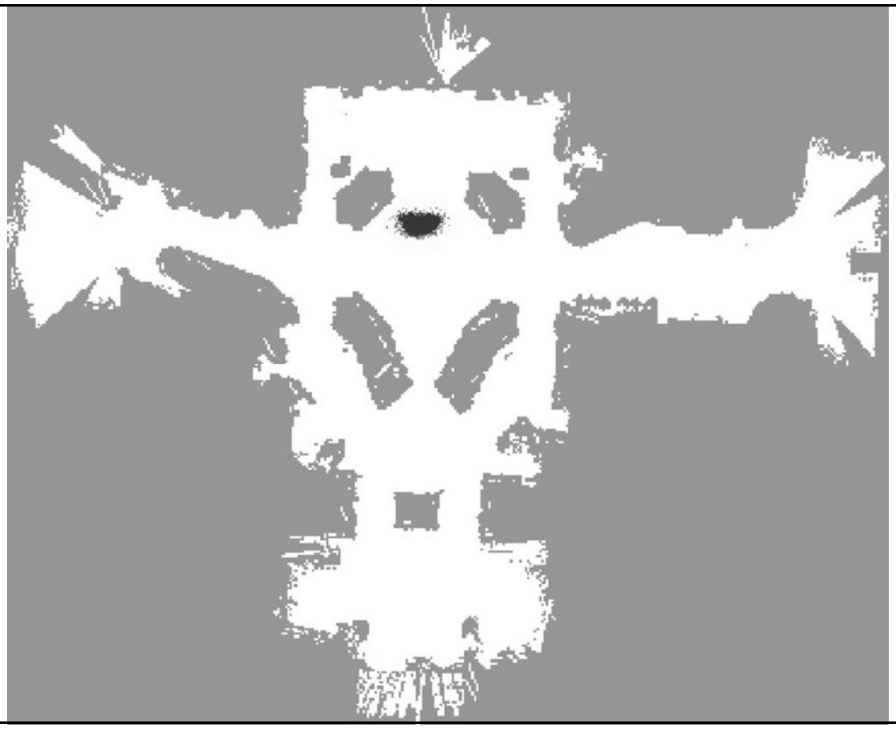


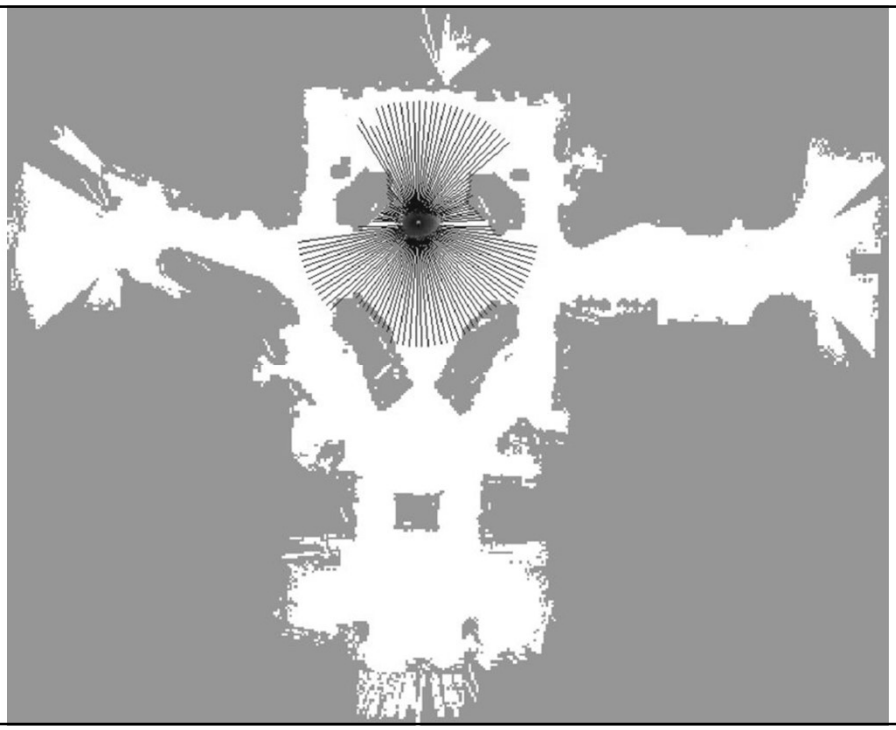


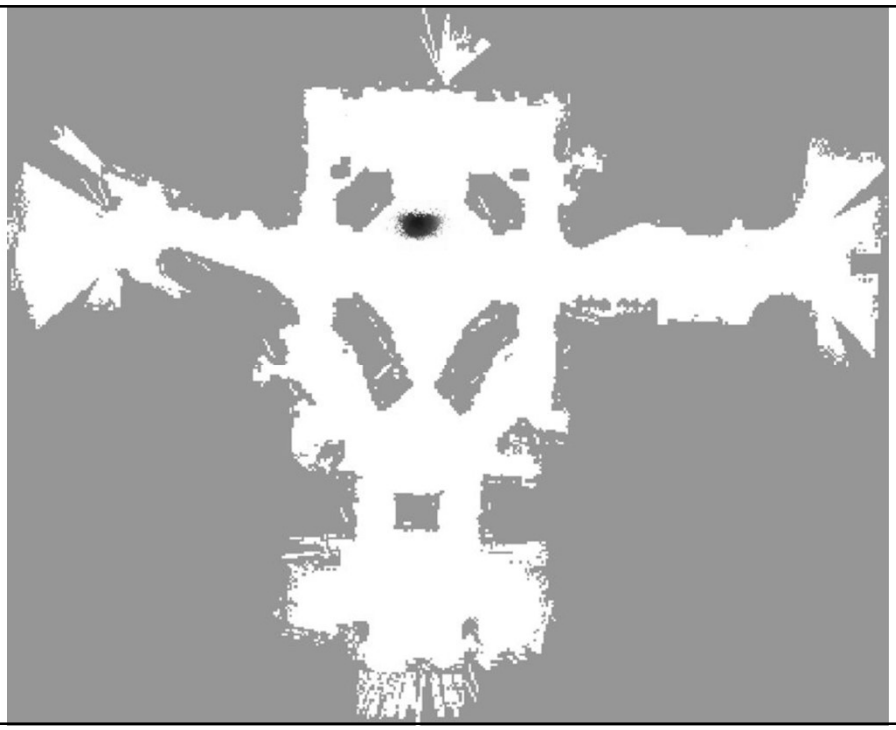


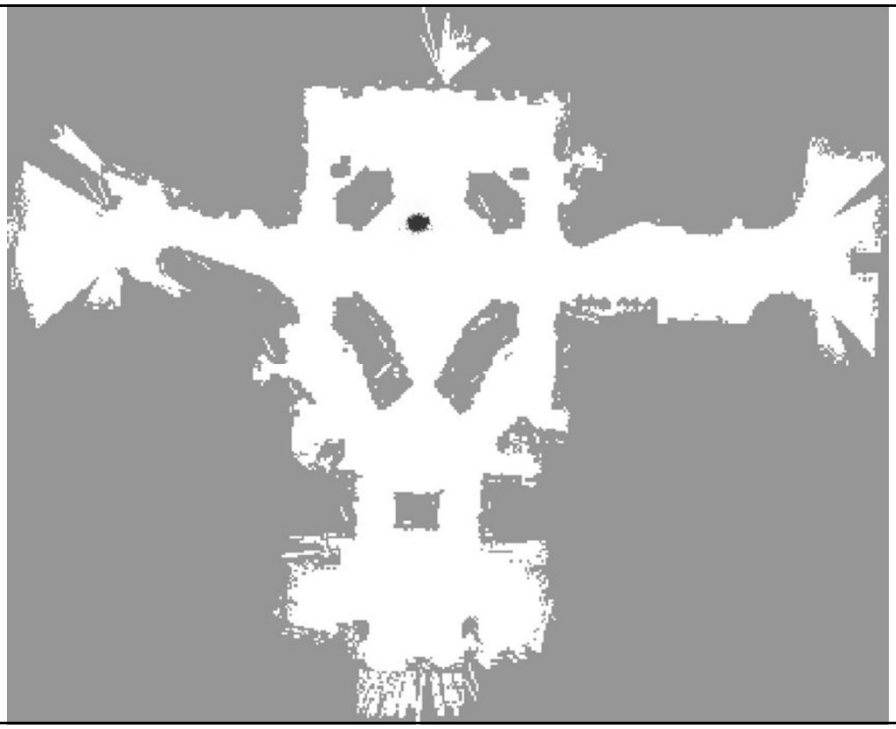


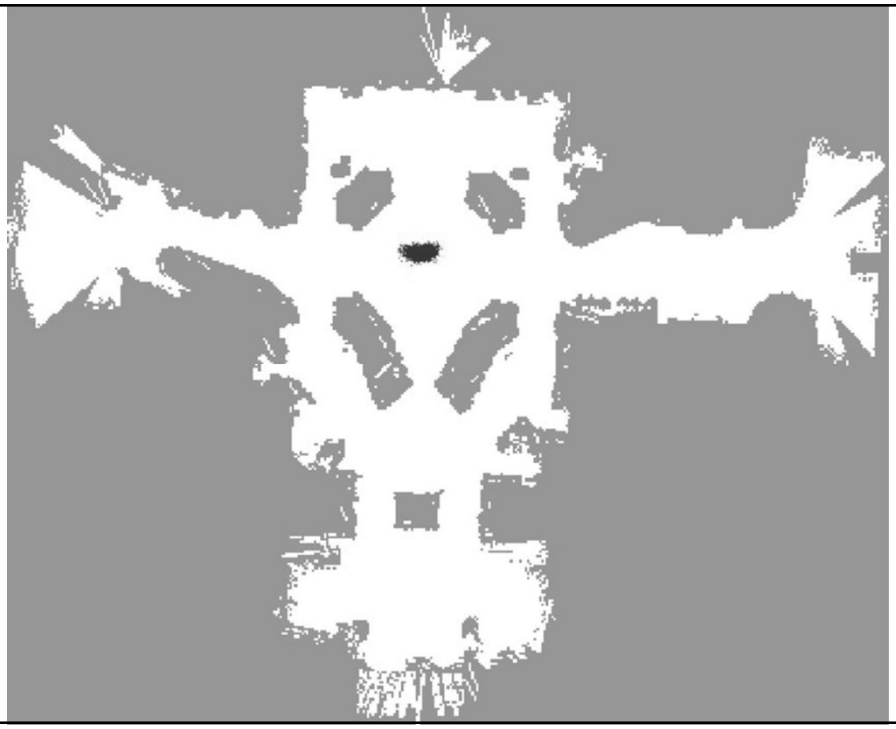












the ROS robot_pose_ekf Package

- *The Robot Pose EKF package is used to estimate the 3D pose of a robot, based on (partial) pose measurements coming from different sources.*
- *It uses an extended Kalman filter with a 6D model (3D position and 3D orientation) to combine measurements from wheel odometry, IMU sensor and visual odometry.*
- *The basic idea is to offer loosely coupled integration with different sensors, where sensor signals are received as ROS messages*

Robot_Pose_EKF (RPE)

Kombine bir odom üretir
  - LSM odometrisi
  - IMU odometrisi
  - Görsel odometri
  - Tekerlek odometrisi
  - GPS
- Robotun pozu en son t_0 anında güncellendiyse;
- Abone olunan her bir sensörden t_0 anından sonra en az 1 ölçüm gelene kadar beklenir.
- t_0'dan sonra gelen bir t_1 anında odometri bilgisi alınırsa ve bundan sonra t_2 anında IMU bilgisi ulaşırsa (yani her bir sensörden bilgi almak mümkün olduğunda) kombine odom bilgisi güncellenebilmektedir.
- Her bir mesajın t_2 anında içerdiği bilgi farklı time_stamp barındırıyorsa, lineer interpolasyon ile dönüşüm yapılır.

Laser_Scan_Matcher (LSM)

- /scan mesajına abone olur
- /pose2D mesajı üretir
- Iterative Closest Point – ICP temelli PLICP kullanır.
- /odom üretilmektedir

Algoritma PLICP

Girdi: y_{t-1}, y_t, q_0

$S^{ref} \leftarrow y_{t-1}$ den oluşturulan parçalı doğru yüzeyi,

$k \leftarrow 0$

repeat

$p_i^w \leftarrow p_i \oplus q_k$

$j_1^i, j_2^i \leftarrow p_i^w$'ye en yakın iki nokta $(j_1^i, j_2^i \in y_{t-1})$

$C_k \leftarrow$ tüm $\langle i, j_1^i, j_2^i \rangle$ üçlüleri

C_k'dan aykırı değerleri temizle

$J(q_{k+1}, C_k) \leftarrow \sum_i \left(n_i^T \left[R(\theta_{k+1}) p_i + t_{k+1} - p_{j_1^i} \right] \right)^2$

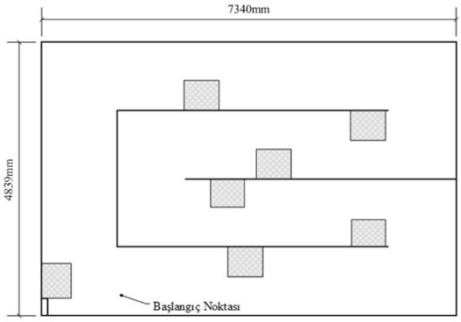
J'yi en küçükleyen q_{k+1} değerini bul

$k \leftarrow k+1$

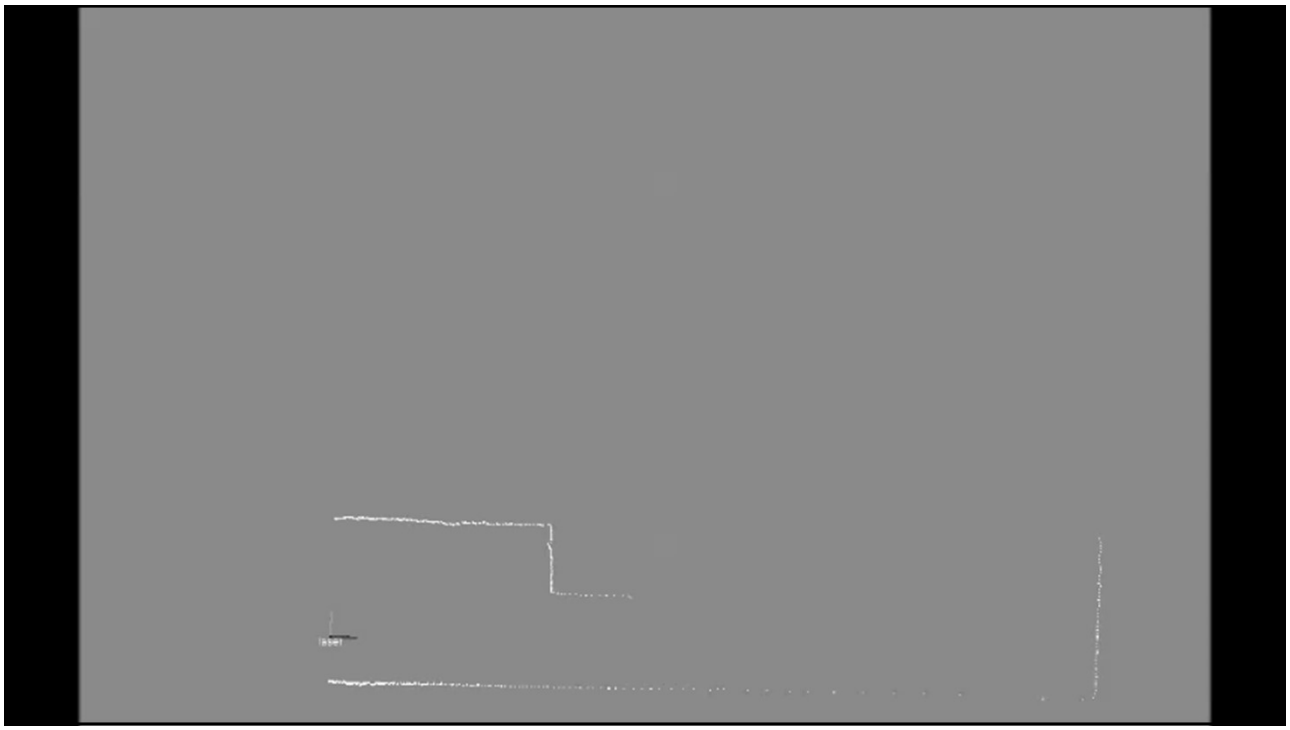
until (max _iterasyon_sayısı) or (yakınsama)

Çıktı: q

LSM Deneyleri – Deney Ortamı



laser


laser

Sabırla Dinlediğiniz İçin Teşekkürler

BLM4830
Robot
Teknolojisine
Giriş
Hafta 4

Öğr. Grv. Furkan Çakmak