ROS2: SLAM (Simultaneous Localization And Mapping)

운영체제의 실제

안인규 (Inkyu An)

What is SLAM?

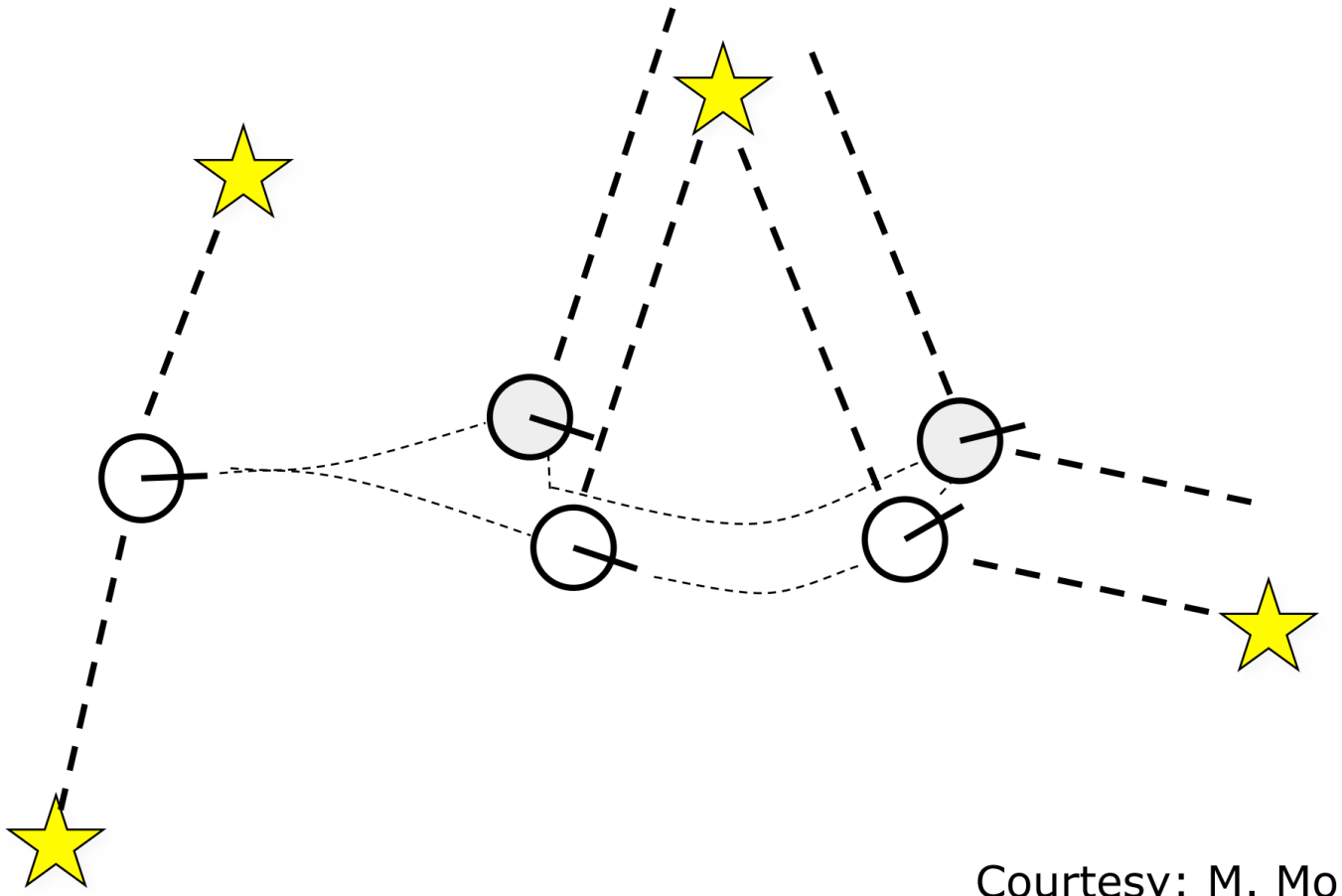
- Compute the robot's poses and the map of the environment at the same time

- **Localization:** estimating the robot's location
- **Mapping:** build a map
- **SLAM:** building a map and localizing the robot simultaneously

referred from "Introduction to SLAM, Cyrill Stachniss"

Localization Example

- Estimate the robot's poses given landmarks

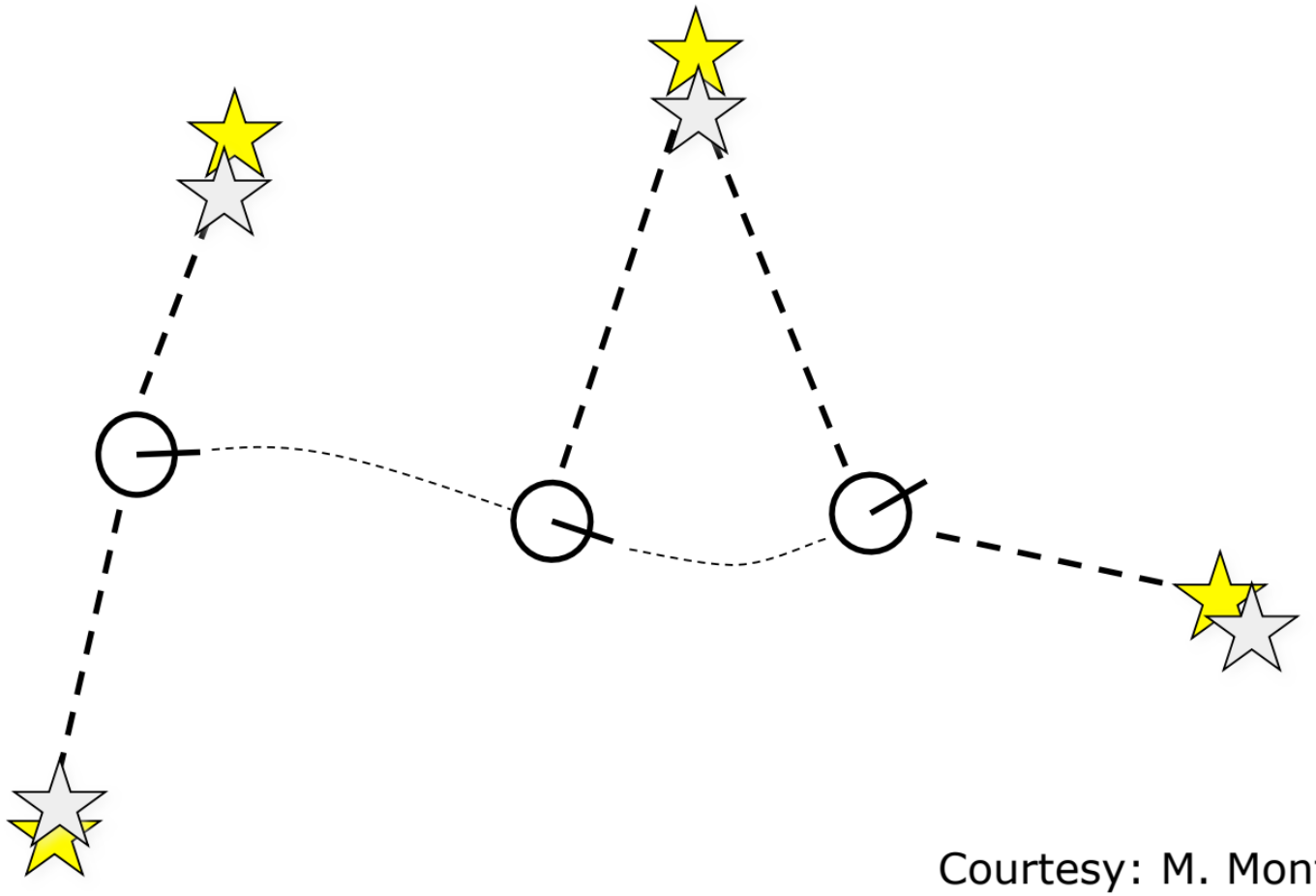


Courtesy: M. Montemerlo

referred from "Introduction to SLAM, Cyrill Stachniss"

Mapping Example

- Estimate the landmarks given the robot's poses

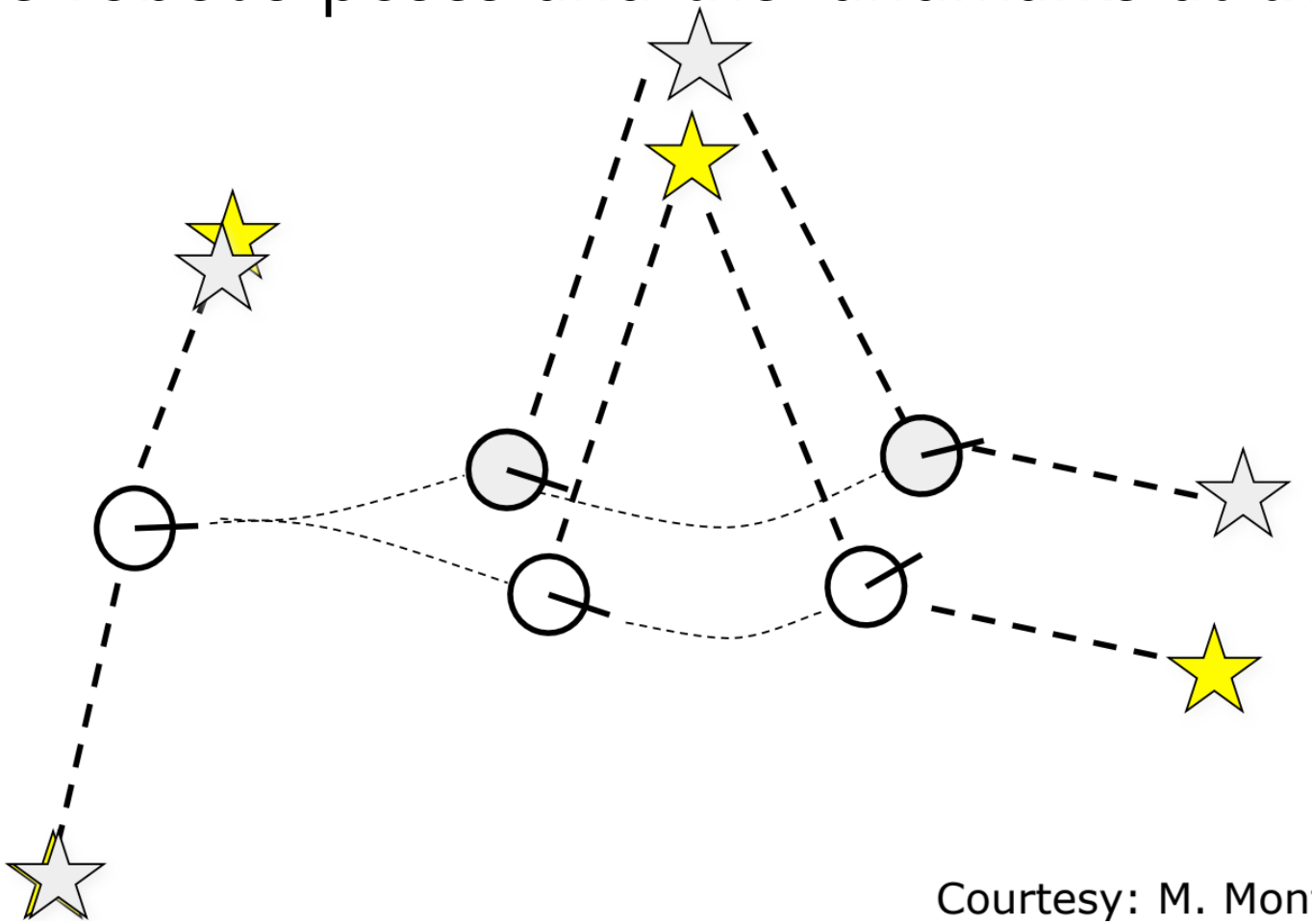


Courtesy: M. Montemerlo

referred from "Introduction to SLAM, Cyrill Stachniss"

SLAM Example

- Estimate the robot's poses and the landmarks at the same time



Courtesy: M. Montemerlo

referred from "Introduction to SLAM, Cyrill Stachniss"

The SLAM Problem

- SLAM is **a chicken-or-egg** problem:
  - A map is needed for localization and
  - A pose estimate is needed for mapping

referred from “Introduction to SLAM, Cyrill Stachniss”

SLAM is Relevant

- It is considered a fundamental problem for truly autonomous robots
- SLAM is the basis for most navigation systems

Autonomous navigation

referred from “Introduction to SLAM, Cyrill Stachniss”

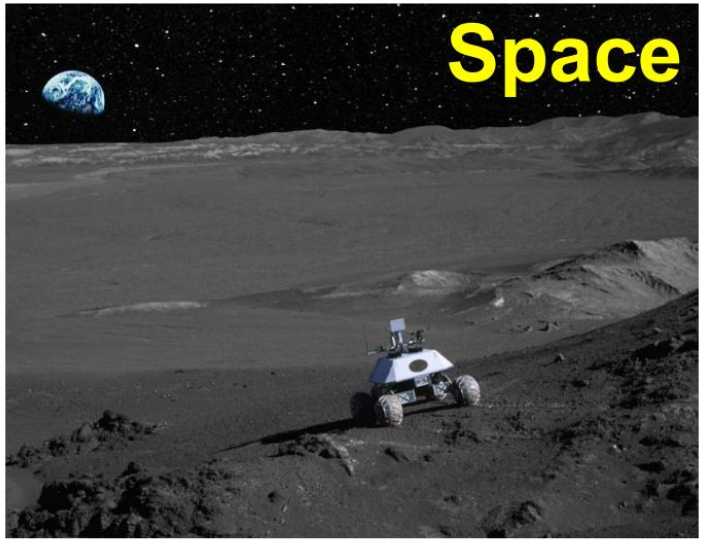
SLAM Applications

- SLAM is central to a range of indoor, outdoor, air and underwater applications for both manned and autonomous vehicles

- Examples:
  - **At home**: vacuum cleaner, lawn mower
  - **Air**: surveillance with unmanned air vehicles
  - **Under water**: reef monitoring
  - **Underground**: exploration of mines
  - **Space**: terrain mapping for localization

referred from “Introduction to SLAM, Cyrill Stachniss”

SLAM Applications

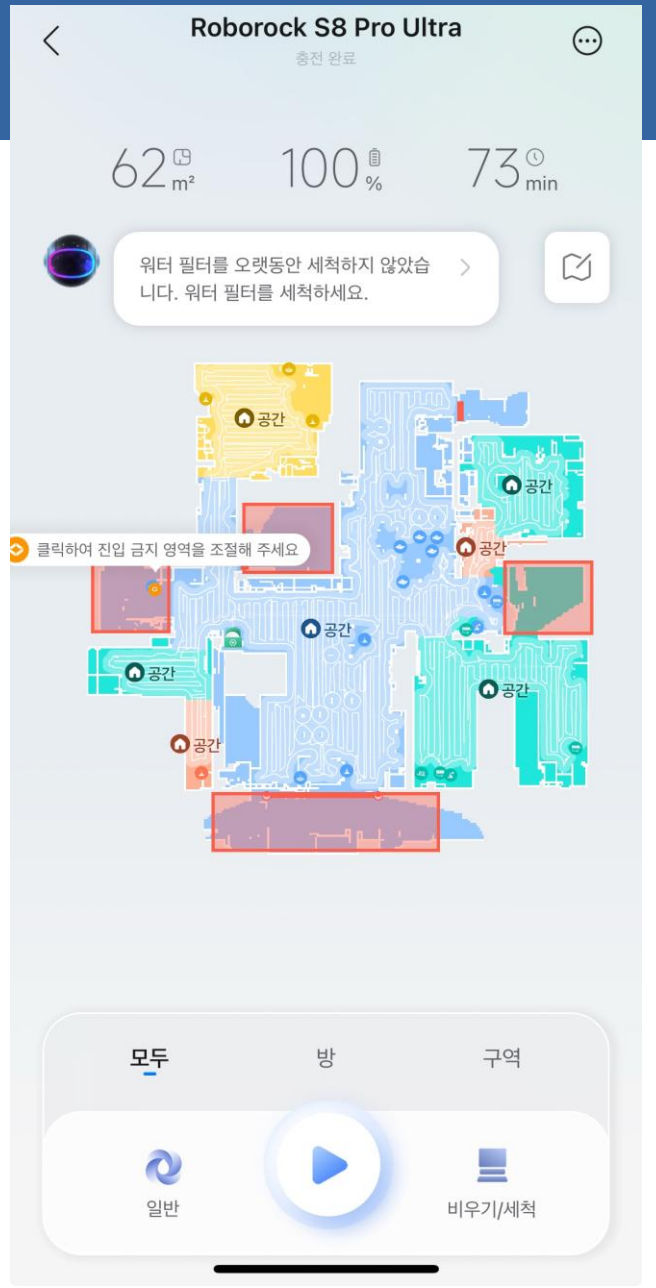


Courtesy: Evolution Robotics, H. Durrant-Whyte, NASA, S. Thrun

referred from “Introduction to SLAM, Cyrill Stachniss”

SLAM Applications

https://youtu.be/bQPj0QQS-f8?si=4l27UF8AkcKP_MMi



SLAM Applications

ORB-SLAM3

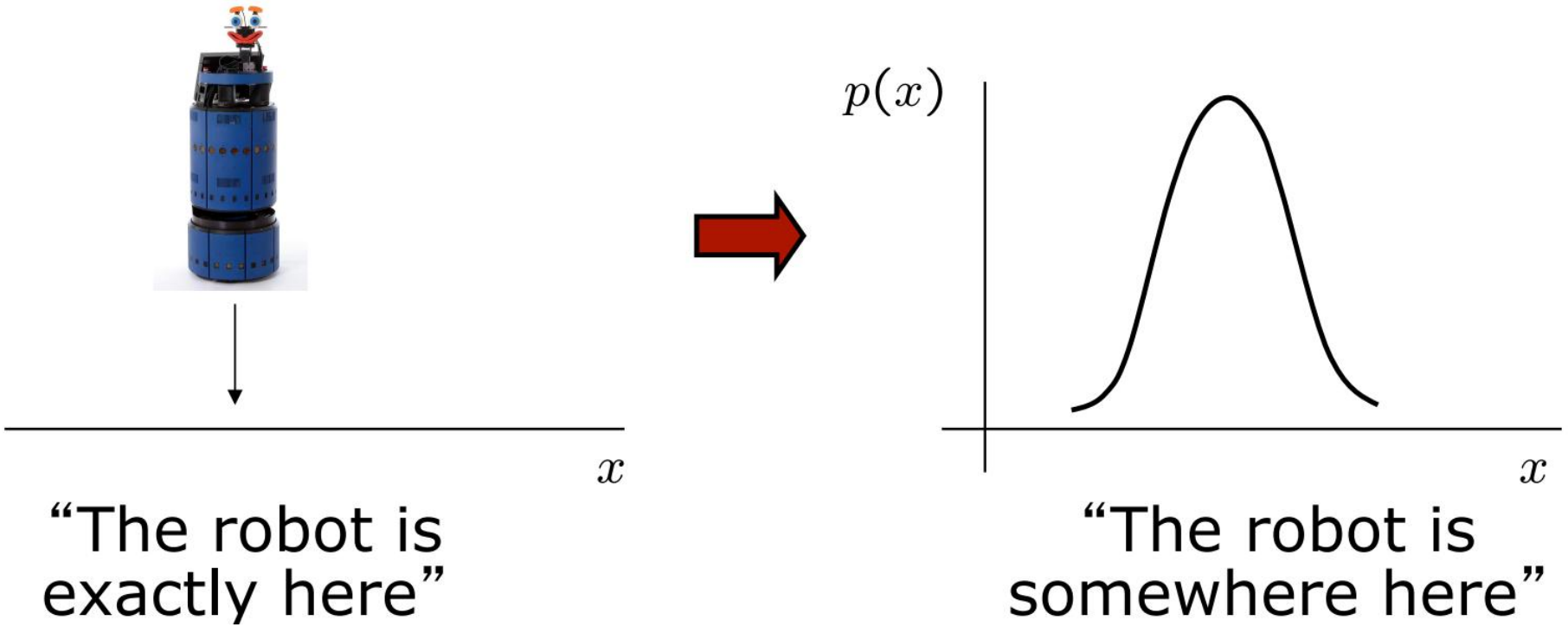
Definition of the SLAM Problem

- **Given**
  - The robot's controls
    - $u_{1:T} = \{u_1, u_2, u_3, \cdots, u_T\}$
  - Observations
    - $z_{1:T} = \{z_1, z_2, z_3, \cdots, z_T\}$
- **Wanted**
  - Map of the environment
    - m
  - Path of the robot
    - $x_{0:T} = \{x_0, x_1, x_2, \cdots, x_T\}$

referred from "Introduction to SLAM, Cyrill Stachniss"

Probabilistic Approaches

- Uncertainty in the robot's motions and observations
- Use the probability theory to explicitly represent the uncertainty



referred from "Introduction to SLAM, Cyrill Stachniss"

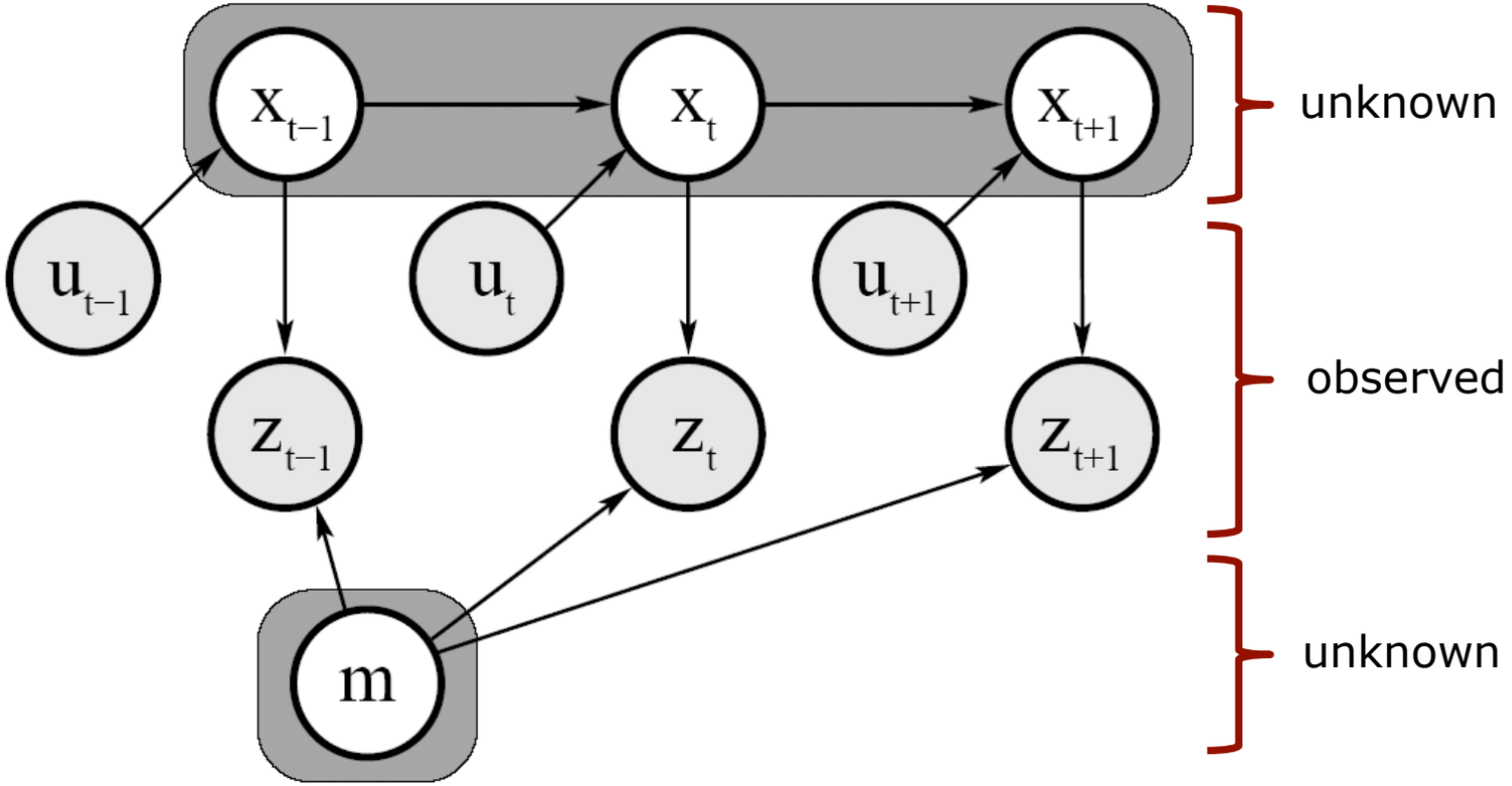
In the Probabilistic World

- Estimate the robot's path and the map

$$p(x_{0:T}, m \mid z_{1:T}, u_{1:T})$$

distribution path map given observations controls

Graphical Model



$$p(x_{0:T}, m \mid z_{1:T}, u_{1:T})$$

referred from “Introduction to SLAM, Cyrill Stachniss”

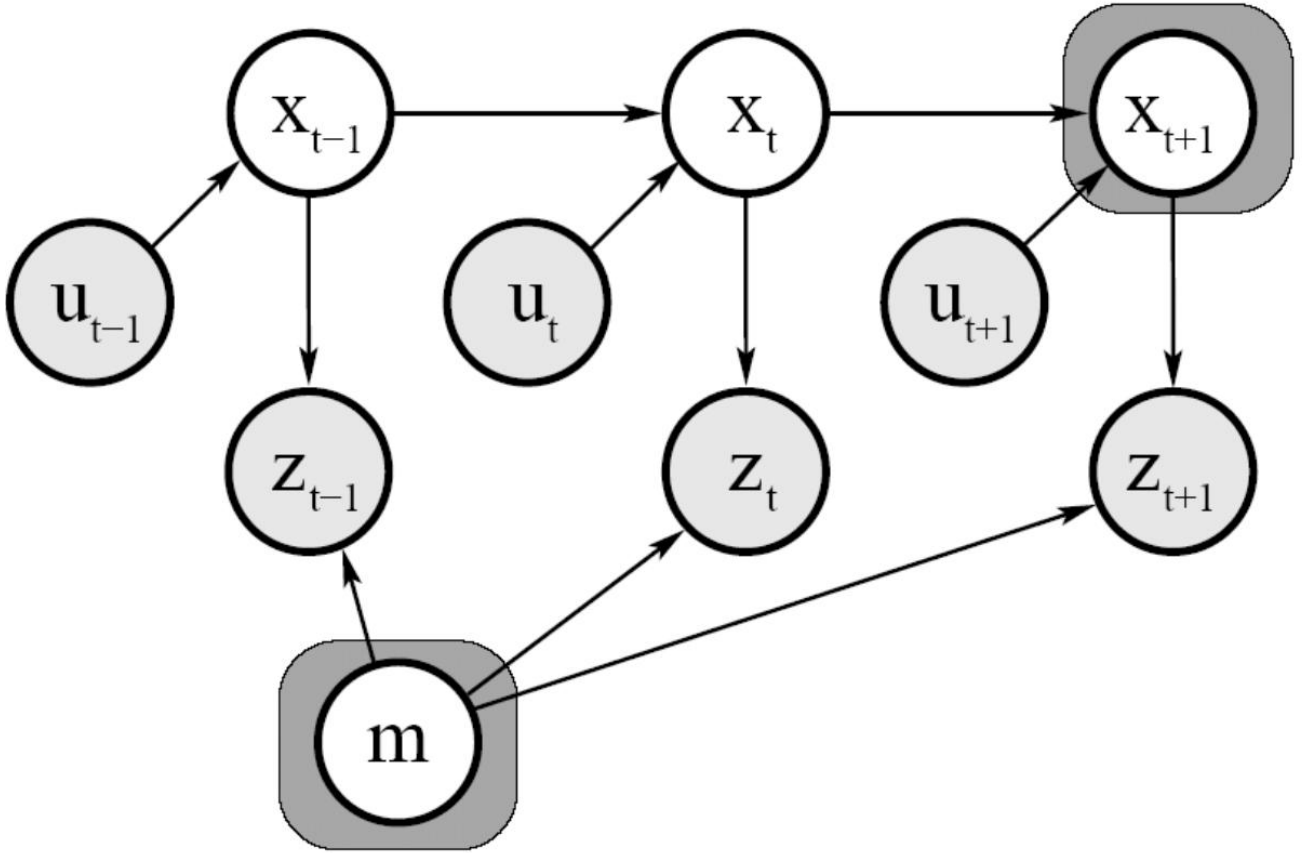
Full SLAM vs. Online SLAM

- Full SLAM estimates the entire path
  - $p(x_{0:T}, m | z_{1:T}, u_{1:T})$

- Online SLAM seeks to recover only the most recent pose
  - $p(x_t, m | z_{1:t}, u_{1:t})$

referred from “Introduction to SLAM, Cyrill Stachniss”

Graphical Model of Online SLAM



$$p(x_{t+1}, m \mid z_{1:t+1}, u_{1:t+1})$$

referred from “Introduction to SLAM, Cyrill Stachniss”

Online SLAM

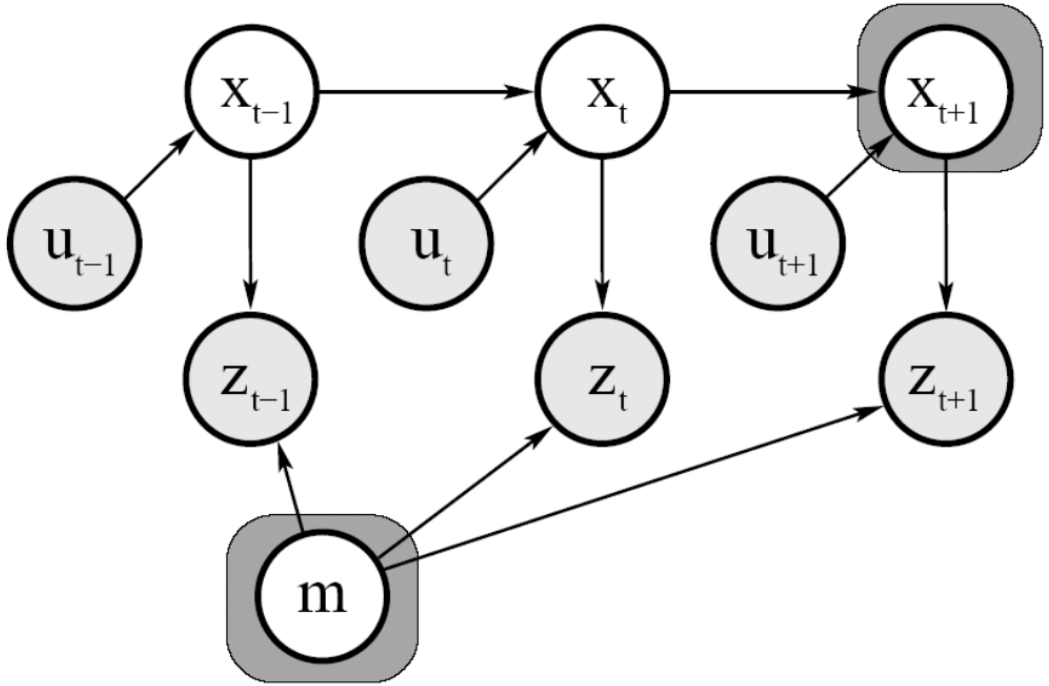
- Online SLAM means marginalizing out the previous poses
  - $p(x_t, m|z_{1:t}, u_{1:t}) = \int \cdots \int \underline{p(x_{0:t}, m|z_{1:t}, u_{1:t})}\, dx_{t-1} \cdots dx_0$

Full SLAM

- Integrals all typically solved recursively, one at a time

referred from “Introduction to SLAM, Cyrill Stachniss”

Graphical Model of Online SLAM

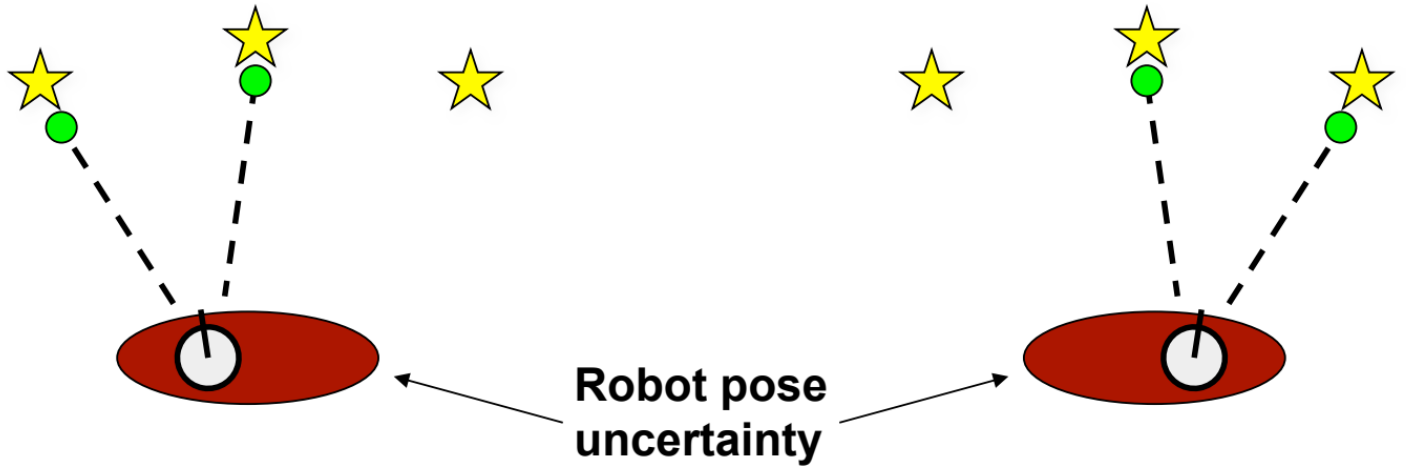


$$p(x_{t+1}, m \mid z_{1:t+1}, u_{1:t+1}) =$$
$$\int \dots \int p(x_{0:t+1}, m \mid z_{1:t+1}, u_{1:t+1})\, dx_t \dots dx_0$$

referred from “Introduction to SLAM, Cyrill Stachniss”

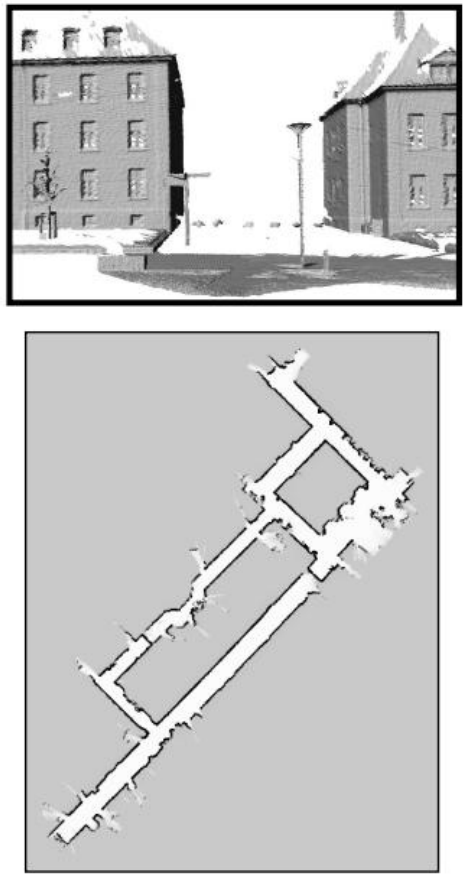
Why is SLAM a Hard Problem?

- The mapping between **observations** and the **map** is **unknown**
- Picking **wrong** data associations can have **catastrophic** consequences (divergence)

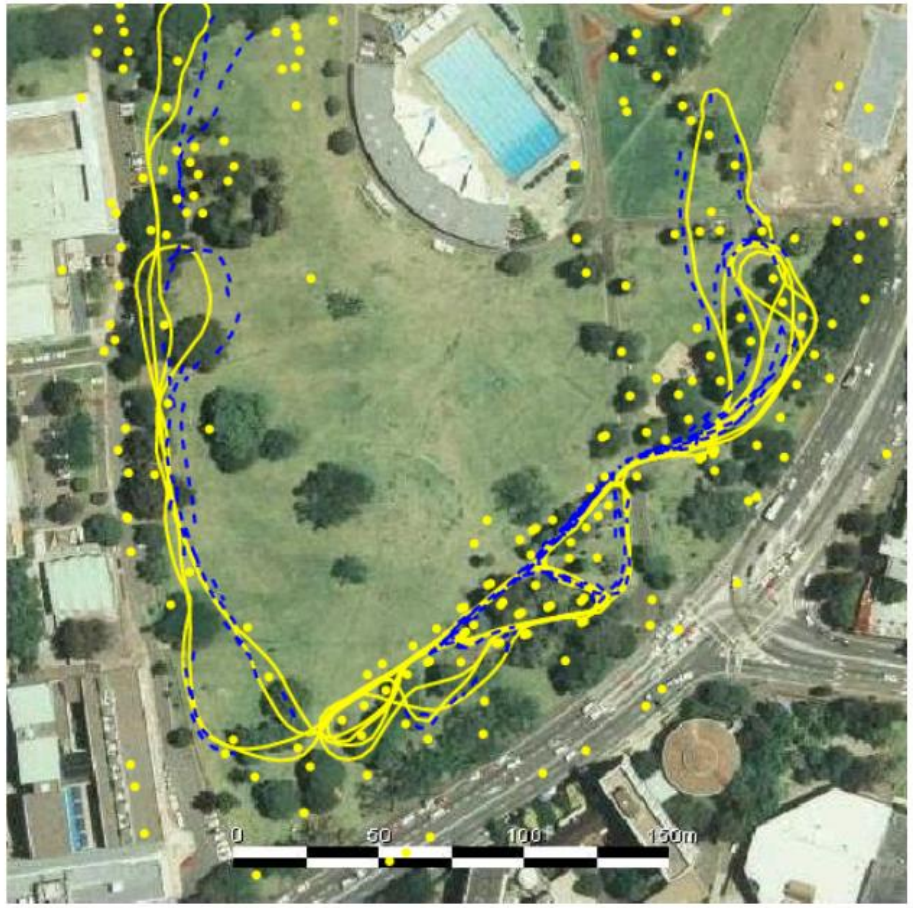


referred from “Introduction to SLAM, Cyrill Stachniss”

Volumetric vs. Feature-Based SLAM



Courtesy: D. Hähnel



Courtesy: E. Nebot

referred from “Introduction to SLAM, Cyrill Stachniss”

Three Traditional Paradigms

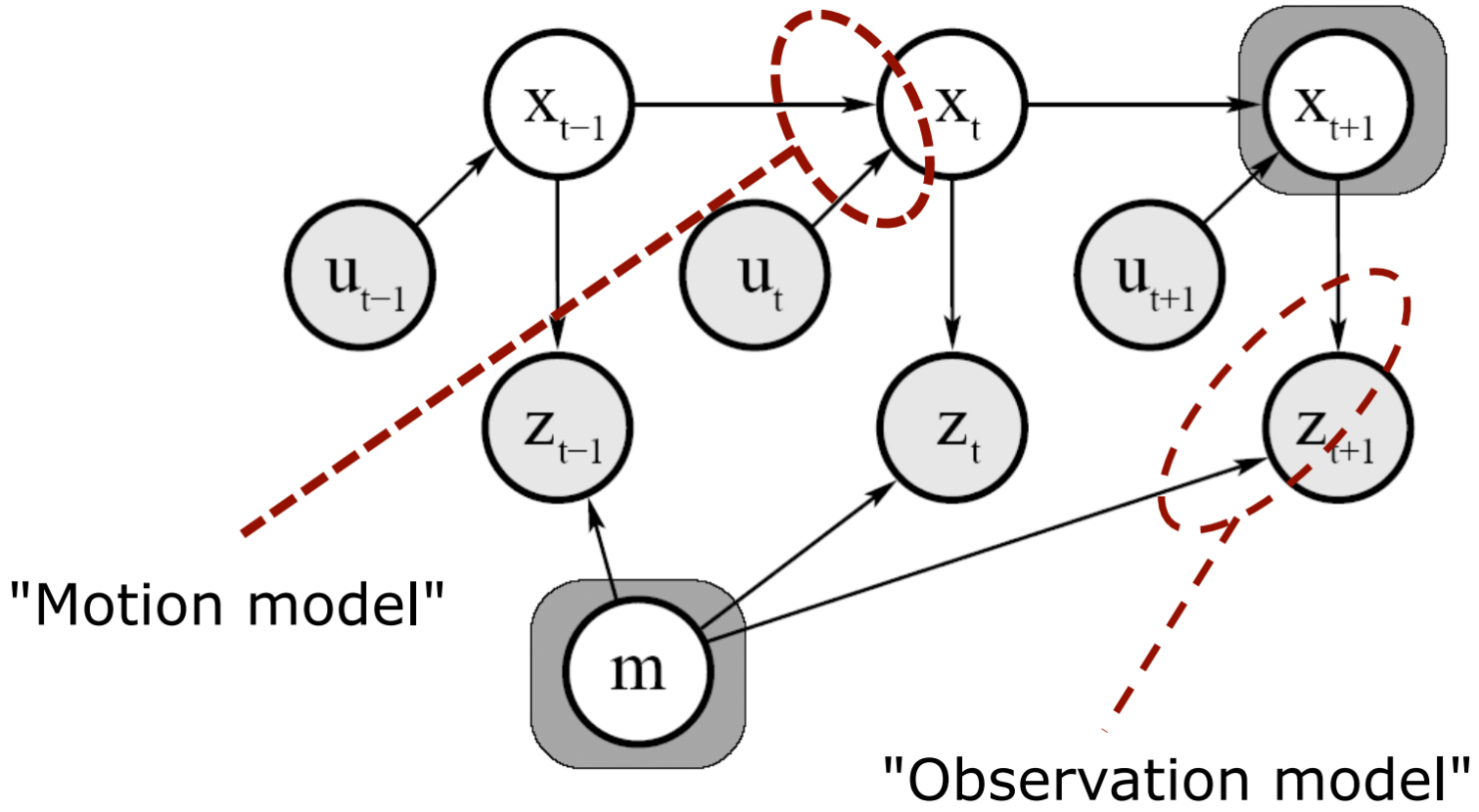
Kalman filter

Particle filter

Graph-based

referred from “Introduction to SLAM, Cyrill Stachniss”

Motion and Observation Model



referred from “Introduction to SLAM, Cyrill Stachniss”

Motion Model

- The motion model describes the relative motion of the robot

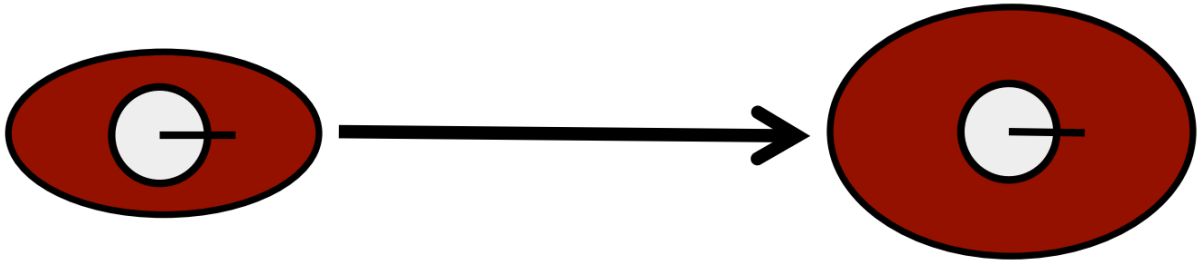
$$p(x_t \mid x_{t-1}, u_t)$$

distribution new pose given old pose control

referred from “Introduction to SLAM, Cyrill Stachniss”

Motion Model Examples

- Gaussian model



- Non-Gaussian model

referred from “Introduction to SLAM, Cyrill Stachniss”

Observation Model

- The observation or sensor model related measurements with the robot's pose

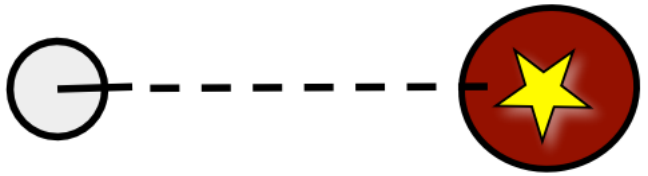
$$p(z_t \mid x_t)$$

distribution observation given pose

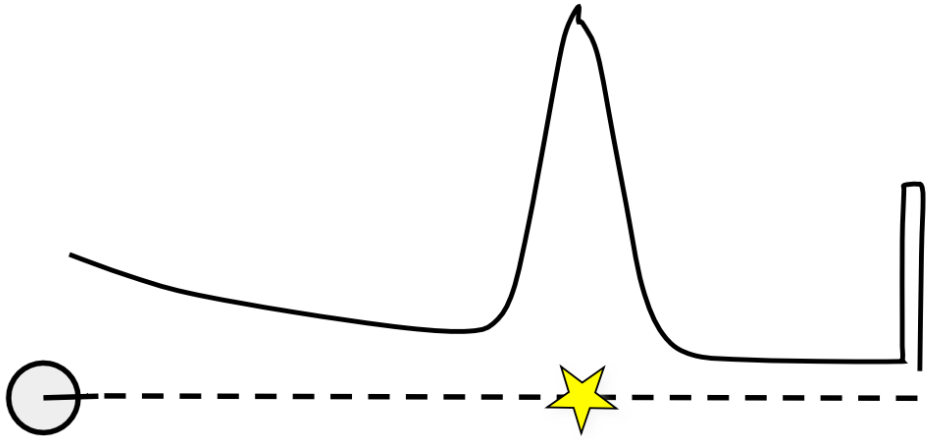
referred from "Introduction to SLAM, Cyrill Stachniss"

Motion Model Examples

- Gaussian model



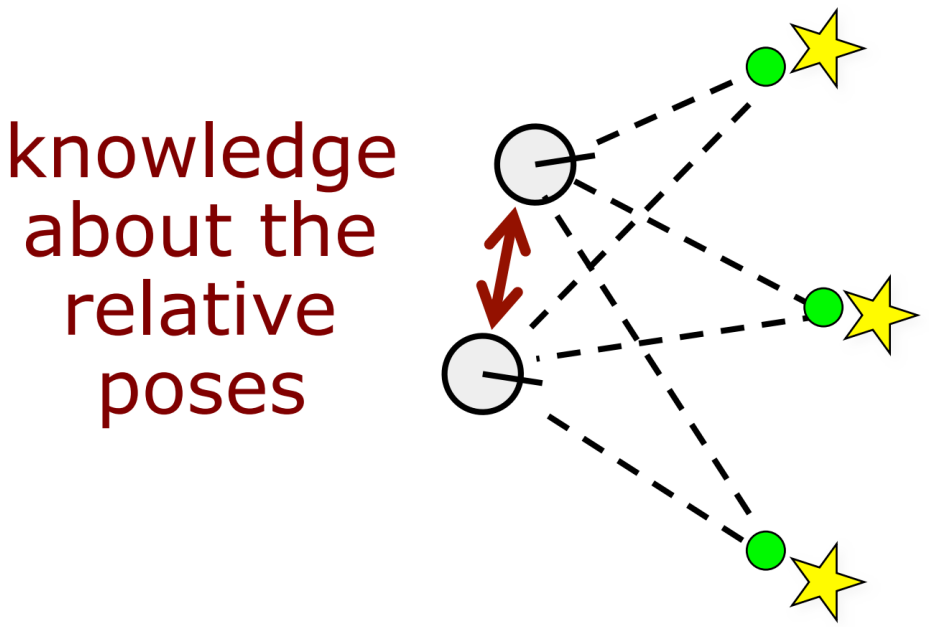
- Non-Gaussian model



referred from “Introduction to SLAM, Cyrill Stachniss”

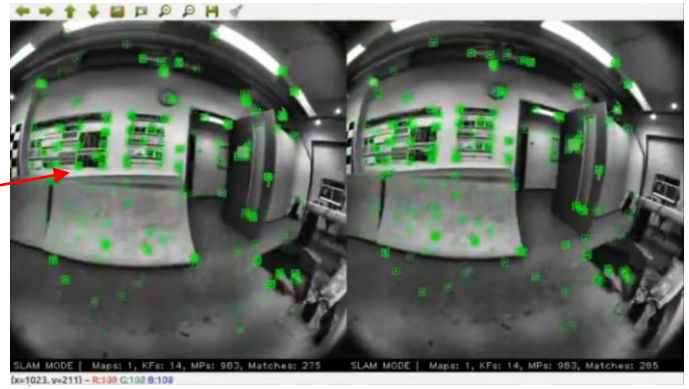
Model for Virtual Observations

- Relate pairs of poses from which observations have been recorded

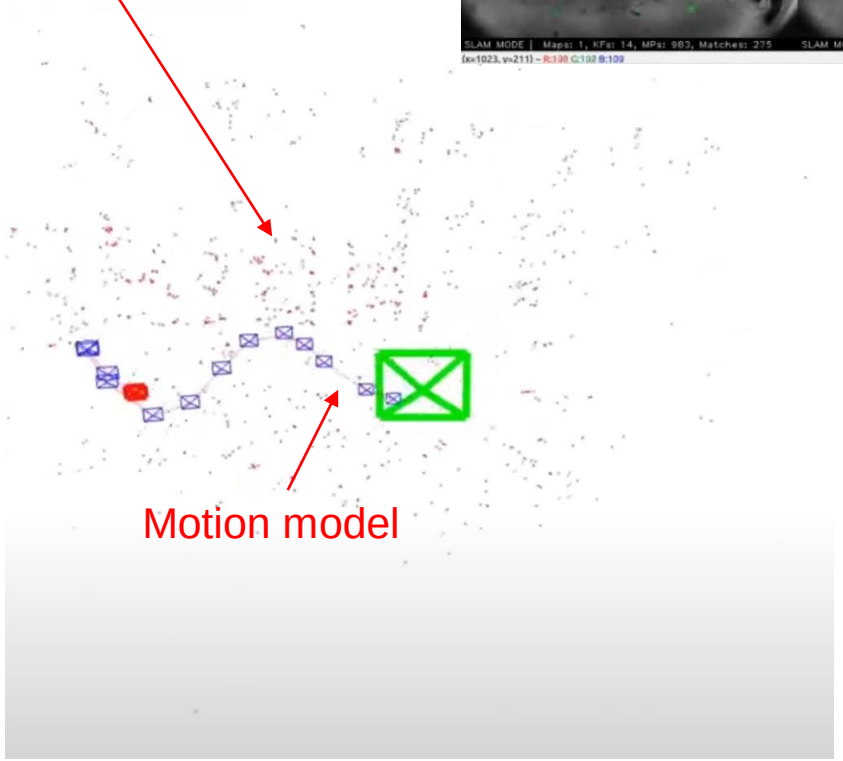


referred from “Introduction to SLAM, Cyrill Stachniss”

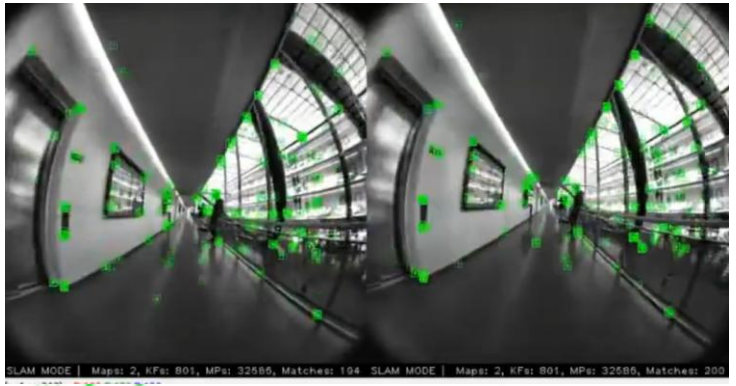
Model for Virtual Observations



Observation model



Motion model



Relative poses

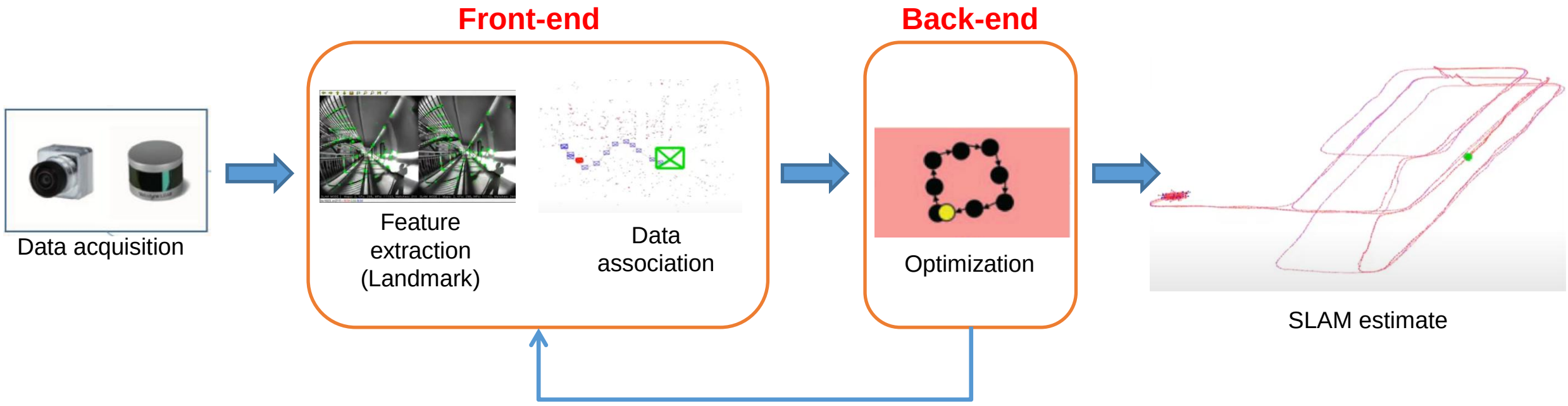
referred from “Introduction to SLAM, Cyrill Stachniss”

SLAM Applications

ORB-SLAM3

SLAM system

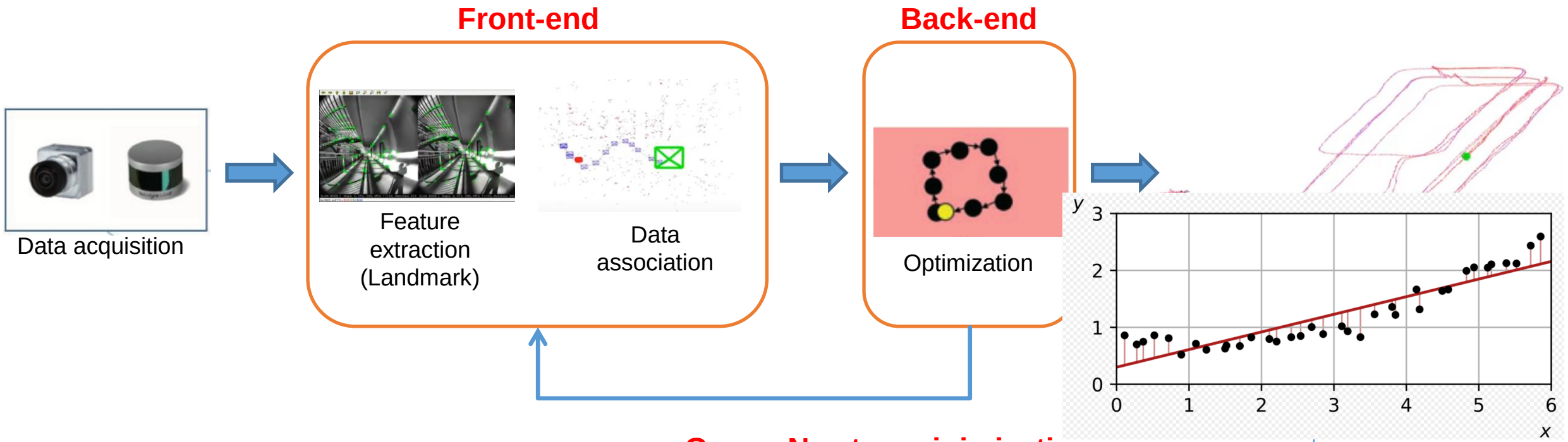
- Graph-based SLAM system



referred from “ORB-SLAM3”

SLAM system

- Graph-based SLAM system



Gauss-Newton minimization:
Non-linear system을 Δx를 이용해 선형화 후,
Least squares problem을 푸는 것

referred from “Gauss-Newton, Wikipedia”

2D Least Squares Example

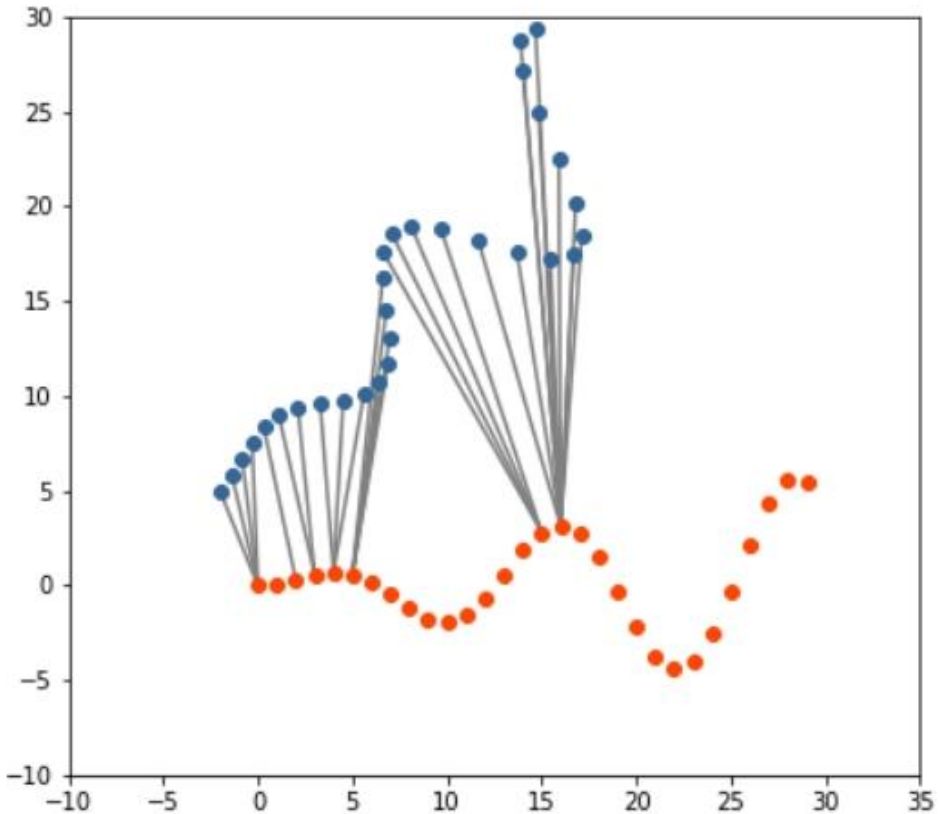


Image courtesy: Bogoslavskyi 23

referred from “Least Squares, Cyrill Stachniss”

2D Least Squares Example

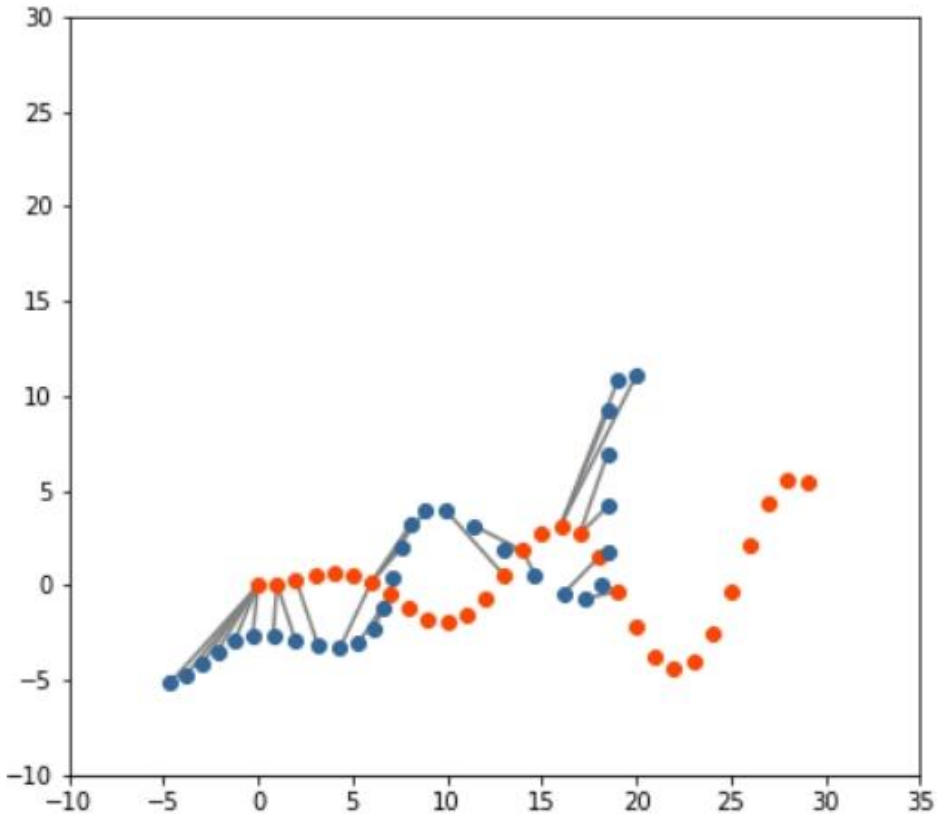


Image courtesy: Bogoslavskyi 24

2D Least Squares Example

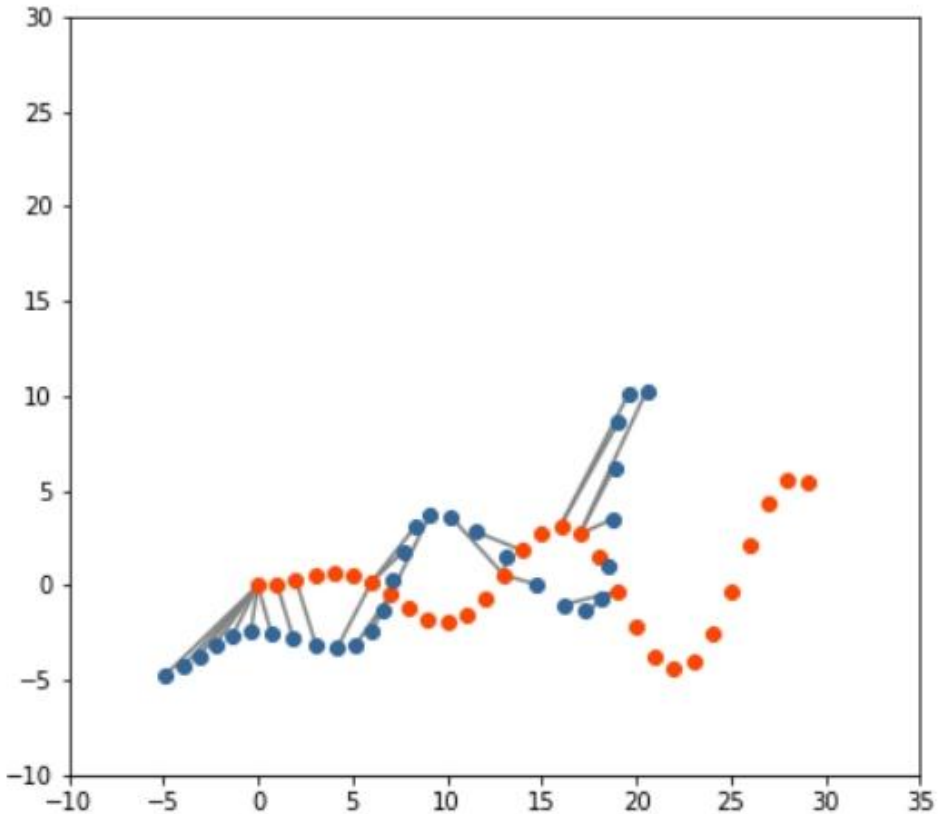


Image courtesy: Bogoslavskyi 25

2D Least Squares Example

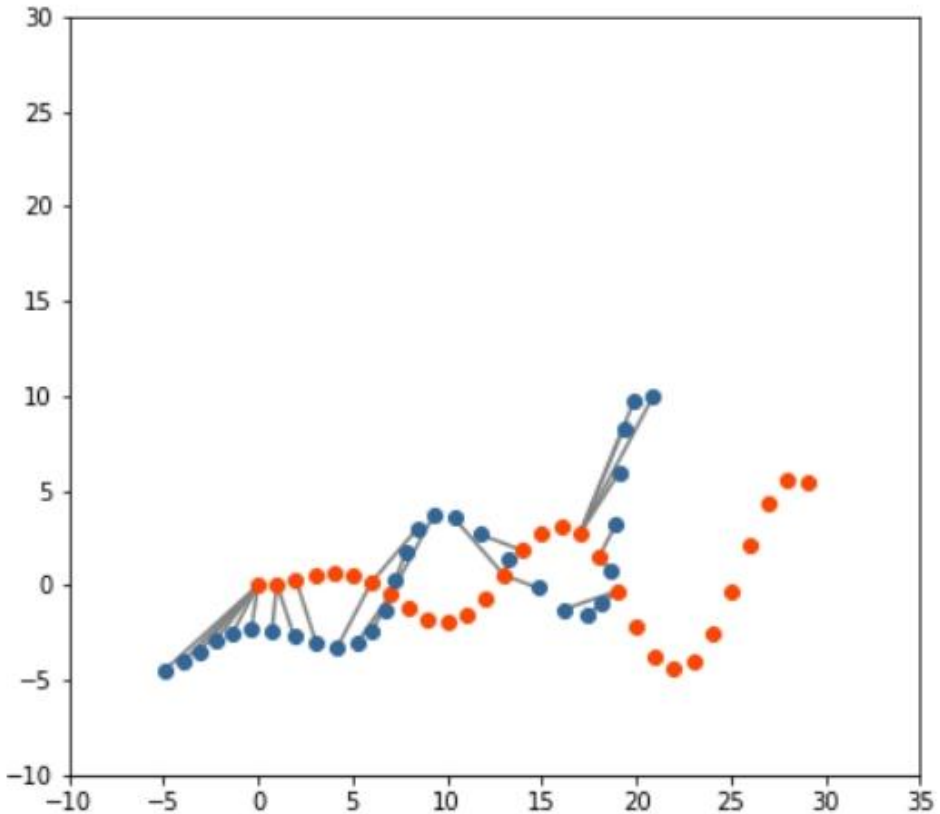


Image courtesy: Bogoslavskyi 26

2D Least Squares Example

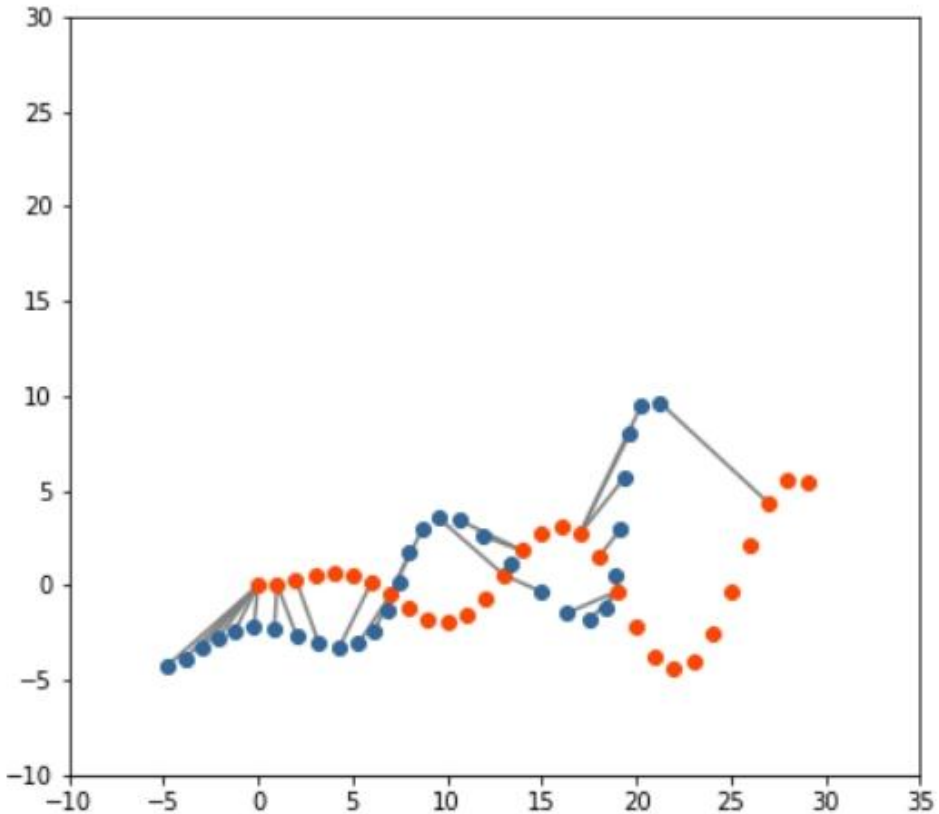


Image courtesy: Bogoslavskyi 27

2D Least Squares Example

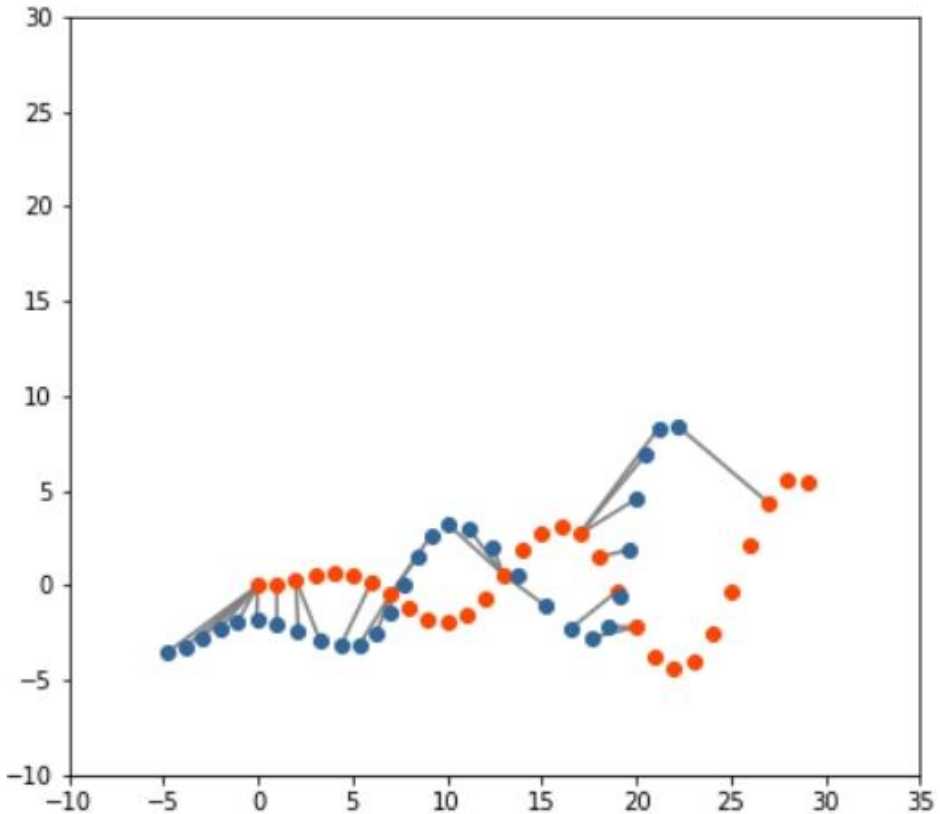


Image courtesy: Bogoslavskyi 28

2D Least Squares Example

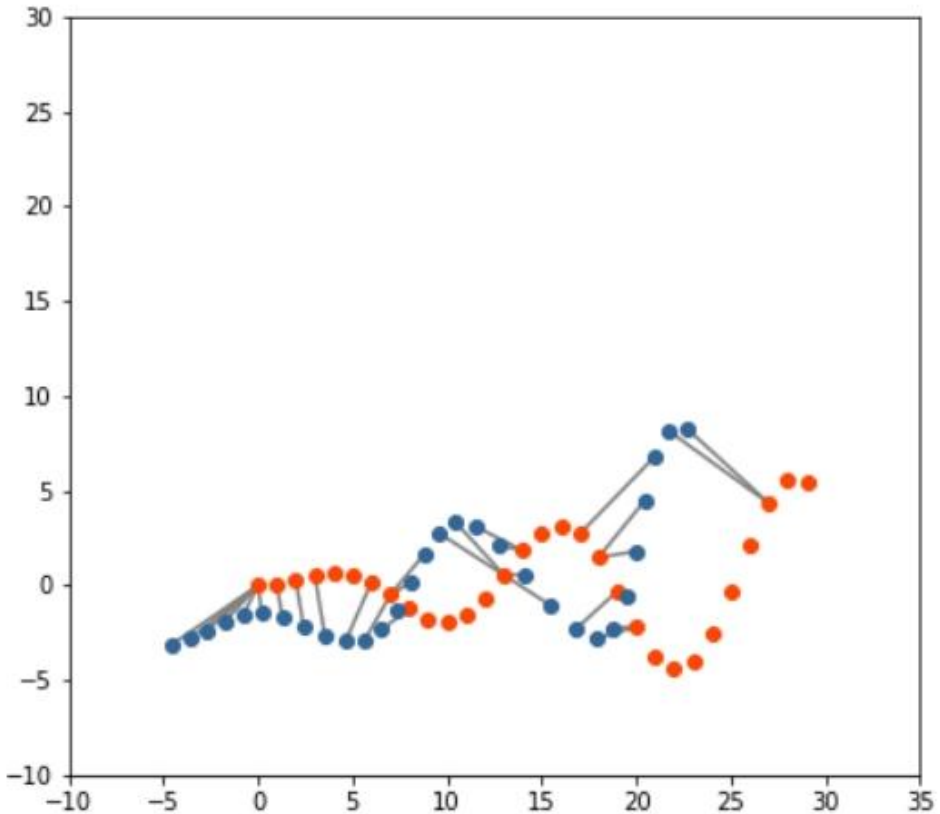


Image courtesy: Bogoslavskyi 29

2D Least Squares Example

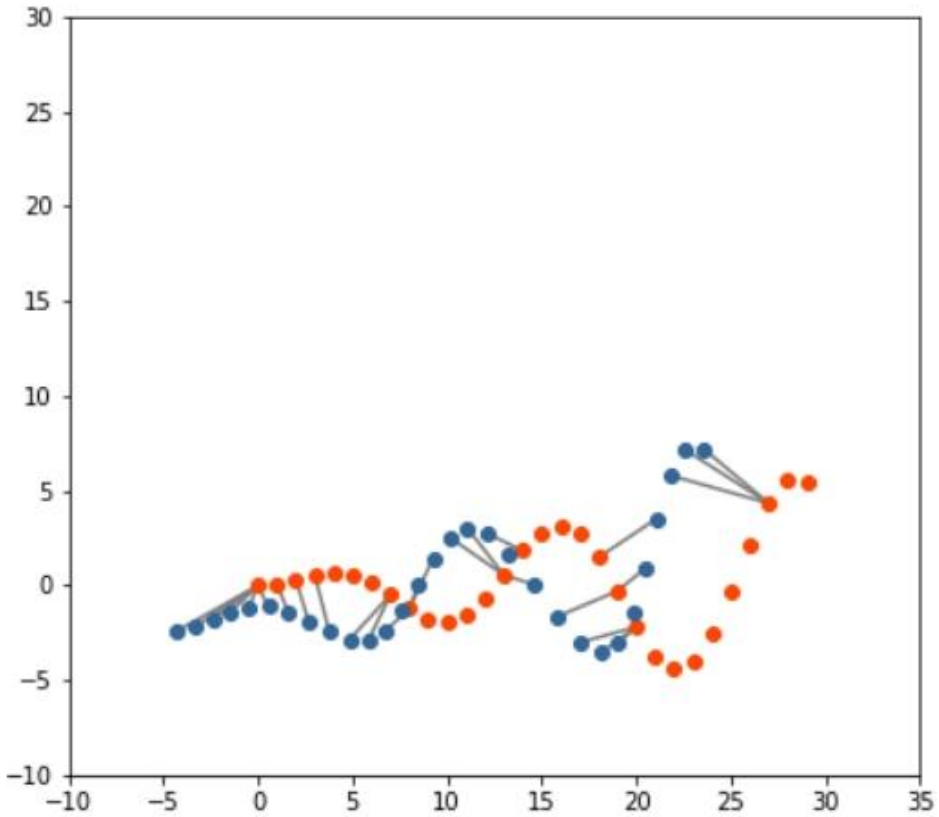


Image courtesy: Bogoslavskyi 30

2D Least Squares Example

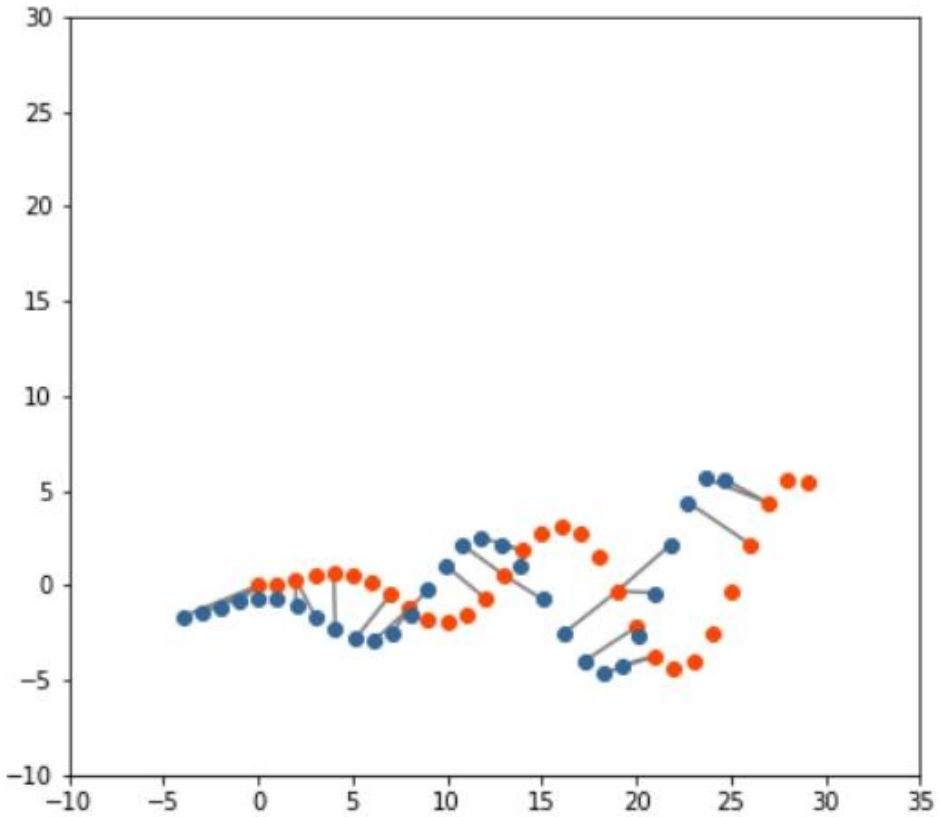


Image courtesy: Bogoslavskyi 31

2D Least Squares Example

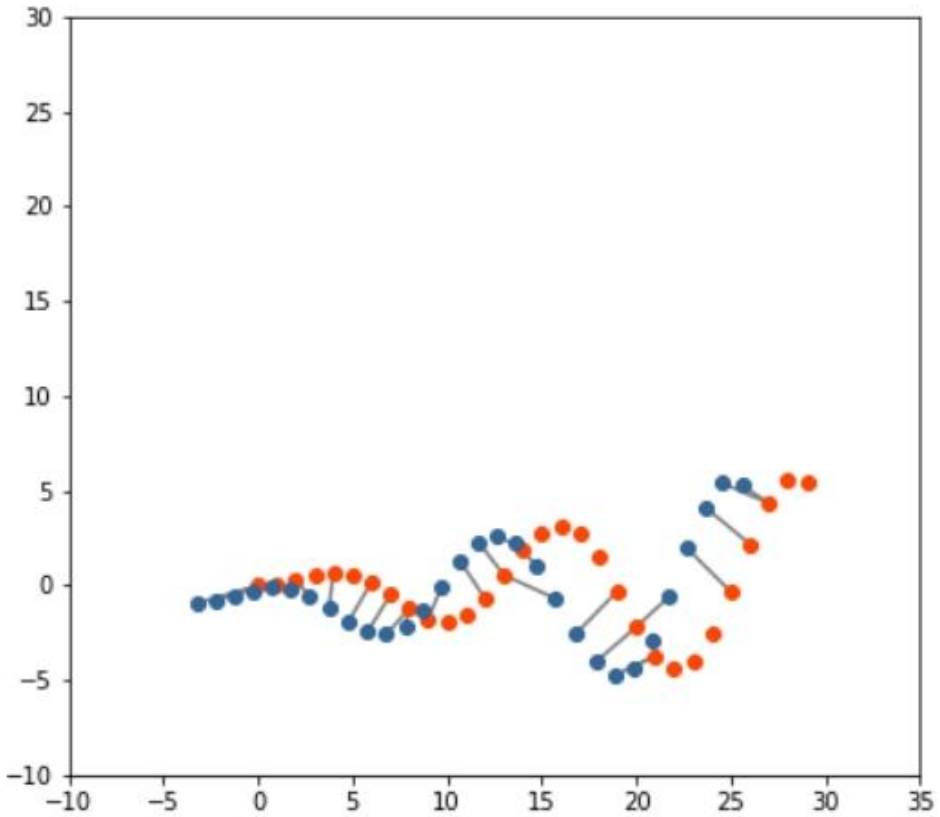


Image courtesy: Bogoslavskyi 32

2D Least Squares Example

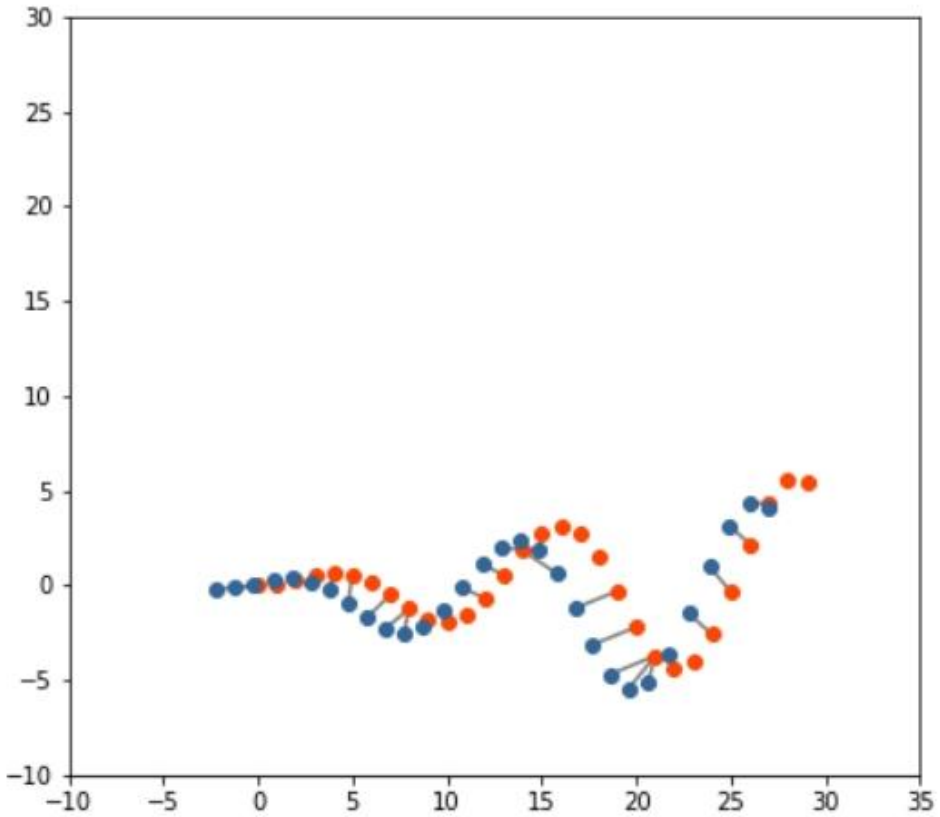


Image courtesy: Bogoslavskyi 33

2D Least Squares Example

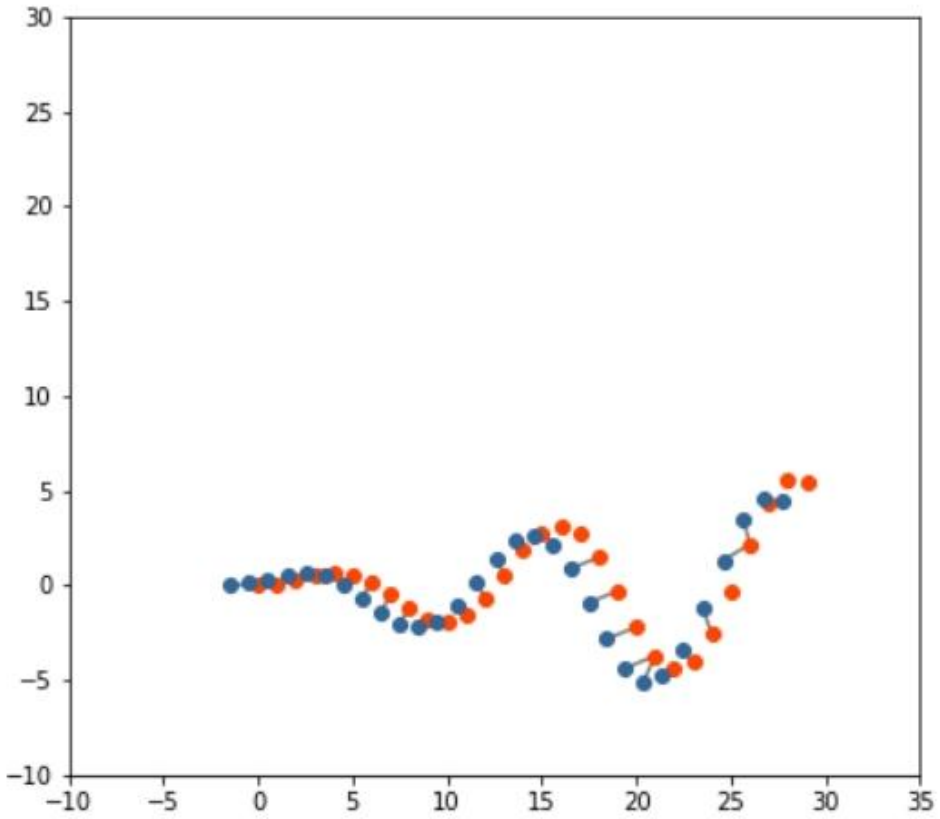


Image courtesy: Bogoslavskyi 34

2D Least Squares Example

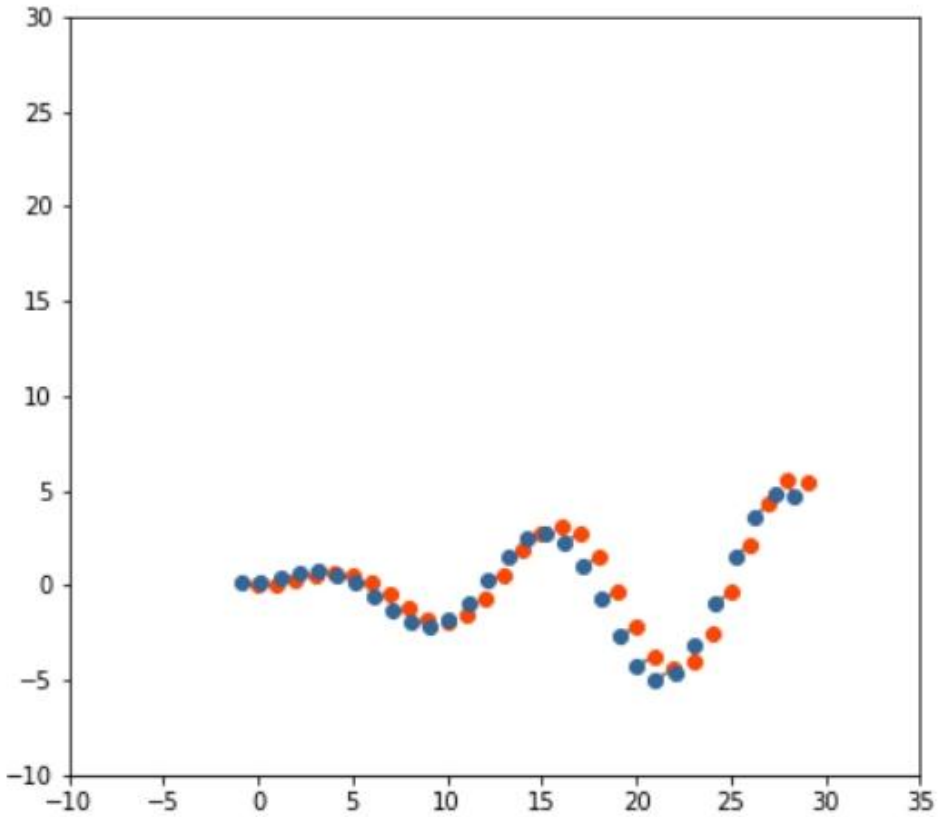


Image courtesy: Bogoslavskyi 35

2D Least Squares Example

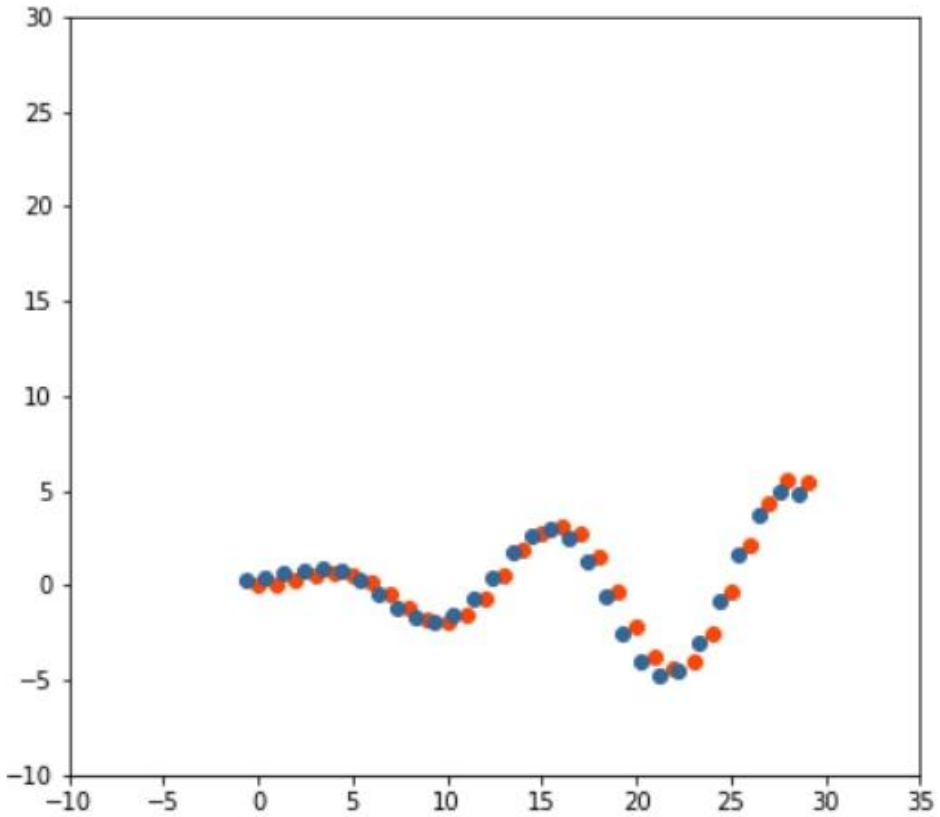


Image courtesy: Bogoslavskyi 36

2D Least Squares Example

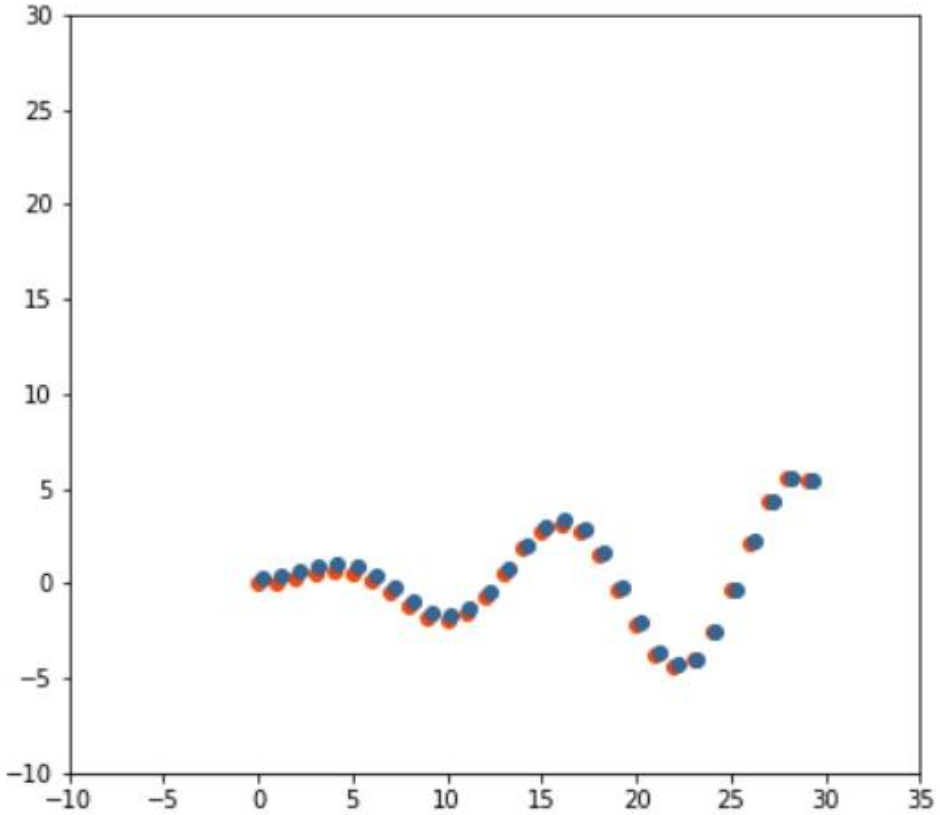


Image courtesy: Bogoslavskyi 37

2D Least Squares Example

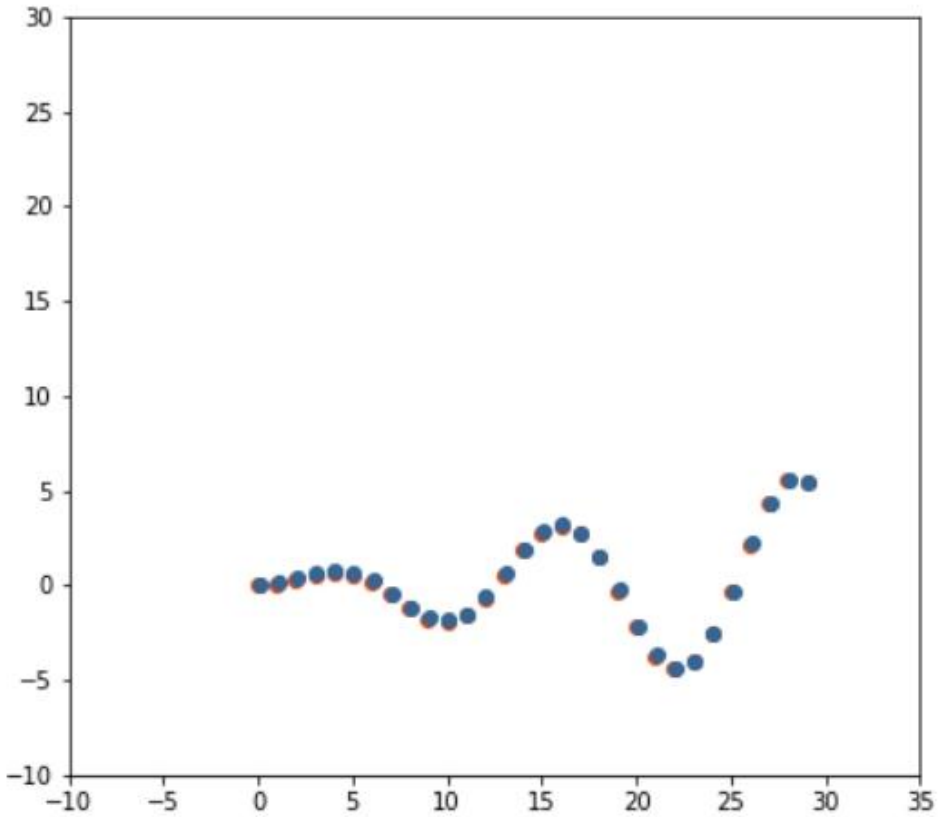


Image courtesy: Bogoslavskyi 38

2D Least Squares Example

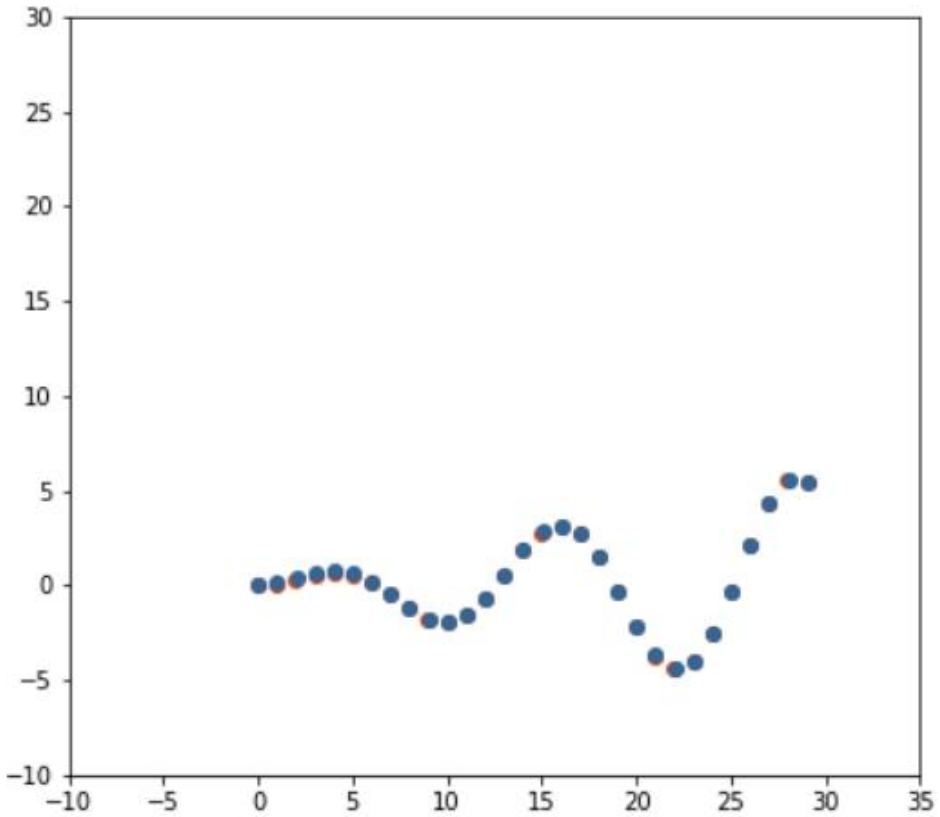


Image courtesy: Bogoslavskyi 39

2D Least Squares Example

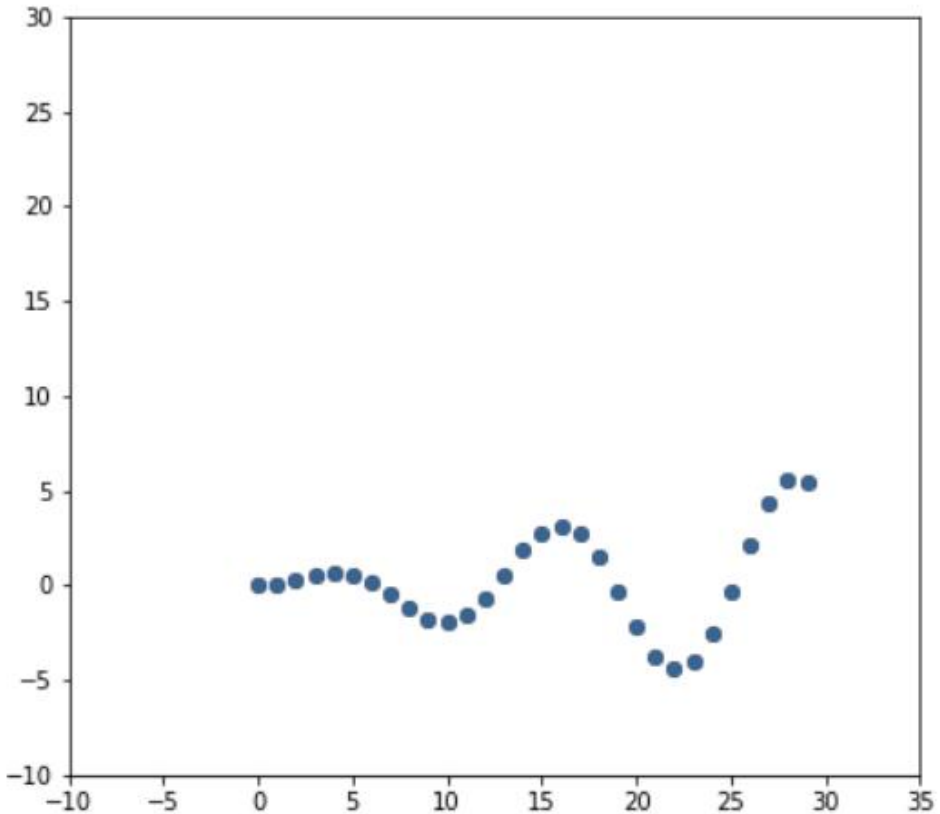


Image courtesy: Bogoslavskyi 40

2D Least Squares Example

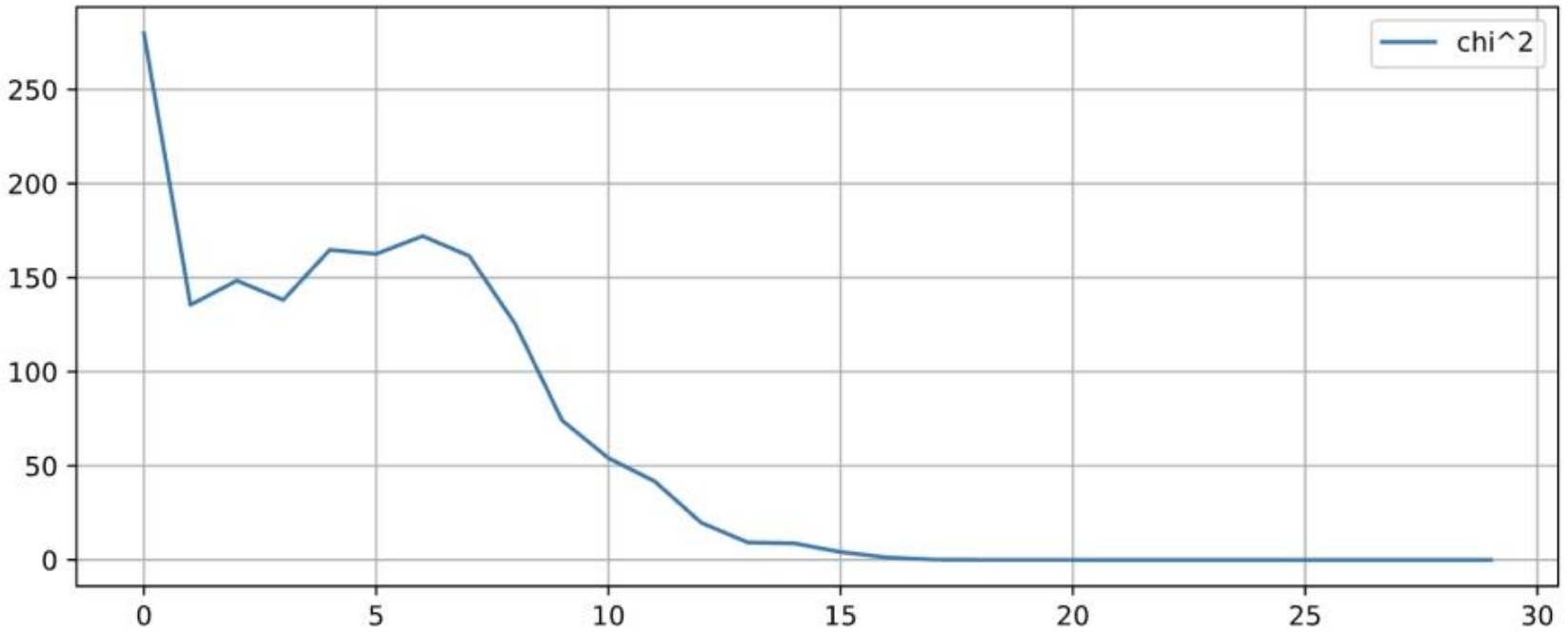


Image courtesy: Bogoslavskyi 41

Turtlebot4 - SLAM

- **How can we build a map using TurtleBot4 (Simulation)?**
  1. Run TurtleBot4 simulator:
     - ros2 launch turtlebot4_ignition_bringup turtlebot4_ignition.launch.py slam:=true nav2:=true rviz:=true
  2. Run SLAM:
     - ros2 launch turtlebot4_navigation slam.launch.py
  3. Drive the TurtleBot4:
     - ros2 run teleop_twist_keyboard teleop_twist_keyboard

referred from “https://turtlebot.github.io/turtlebot4-user-manual/”