INTELLIGENT ROBOTS

CHAPTER 0: INTRODUCTION

Contact Information

Instructor: Qi Hao

E-mail: hao.q@sustc.edu.cn

Office: Nanshan iPark A7 Room 906

Office Hours: M 2:00-4:00pm

Available other times by appointment or the open door policy

Office Phone: (0755) 8801-8537

QQ: 132297436 智能机器人2018

Web: *http://hqlab.sustc.science/teaching/*

Class Schedule

- **Lectures**: T 8:00 am - 9:50 am Lychee Hills Building 1 Room 208
- **Computer Lab**: T 4:20 am – 6:10 pm Lychee Hills Building 6 Room 406
- **Grading policy**

Assignment (8~10 times):	10%	Quiz (5~10 times):	15%
Midterm Exam :	15%	Final Exam	30%
Projects (2 per group):	10%	Final Projects:	20%

90~93: A-	94~97: A	98~100: A+
80~82: B-	83~86: B	87~89: B+
70~72: C-	73~76: C	77~79: C+
60~62: D-	63~66: D	67~69: D+

Textbook and Lecture Notes

Textbooks:

[1]Introduction to Autonomous Mobile Robots, by Roland Siegwart

Other books:

[1] Robotics: Control, Sensing, Vision, and Intelligence, by Fu King Sun

[2] Probabilistic Robotics, by Sebastian Thrun, Dieter Fox, and Wolfram Burgard

[3] Robot Programming, Cameron Hughes and Kacey Hughes

Lecture notes:

http://hqlab.sustc.science/teaching/

Other Resources

Assignment platform: sakai.sustc.edu.cn

Textbook pdf: http://www.doc88.com/p-3951558082918.html

Teaching Objectives

- Fundamental knowledge of many of the basic tasks of intelligent mobile robots: locomotion, estimation, localization, planning, reconstruction and exploration.

- Intelligent robot system development methods with C/C++/Python through labs and projects

- Intelligent robot system design, integration, and verification skills through the final project, literature surveys and reports

Lecture Schedule

Section 00	Course Introduction
Section 01	Mobile Robots, Control and Decision Architecture
Section 02	Locomotion and Sensors
Section 03	Probabilistic Motion and Sensor Models
Section 04	Motion Planning and Collision Avoidance
Section 05	Occupancy and Mapping
Section 06	Particle and Kalman Filters
Section 07	Error Propagation and Feature Estimation
Section 08	Localization
Section 09	SLAM
Section 10	ROS
Section 11	Path Planning and Navigation
Section 12	Reconstruction
Section 13	Exploration

Lab Schedule

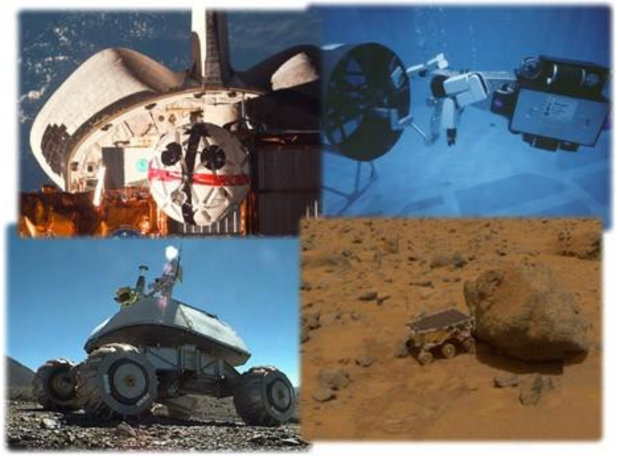
Section 00	Lab Tour
Section 01	Simulation Platform
Section 02	Experiment Platform
Section 03	Robot Perception
Section 04	Robot Control and Decision
Section 05	Robot Communications
Section 06	Motion Planning and Collision Avoidance
Section 08	Occupancy and Mapping
Section 09	Robot Localization
Section 10	SLAM
Section 11	Path Planning and Navigation

Outline

- Robot Types and Framework
- Problem Statement
- Related Areas
- History
- Optimization for Intelligent Robots
- Algorithms
- Platforms

Robot Types

- Robot Manipulators
  - Assembly, automation
- Field robots
  - Military applications
  - Space exploration
- Service robots
  - Cleaning robots
  - Medical robots
- Entertainment (education) robots



Robot Manipulators

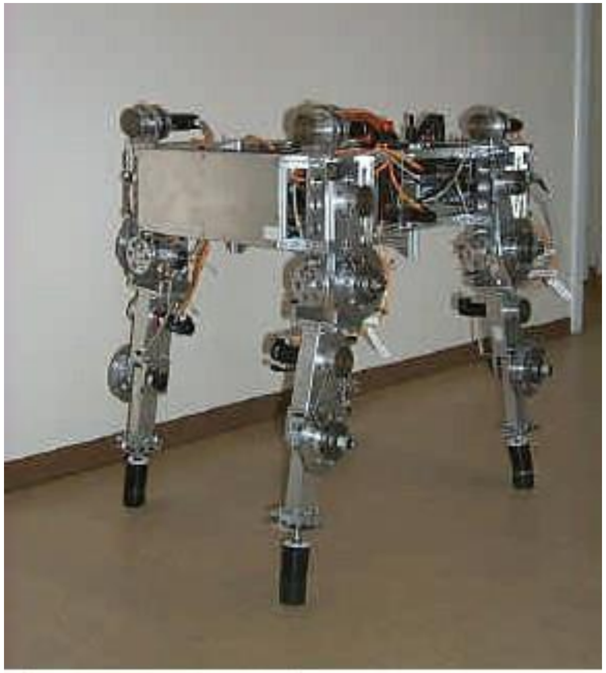
- Static Manipulators

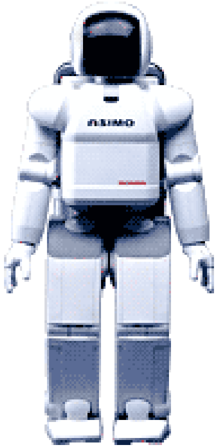
- Mobile Manipulators

Field Robots: Locomotion

Aerial Robots

Wheeled mobile robots





Humanoid

Underwater robots

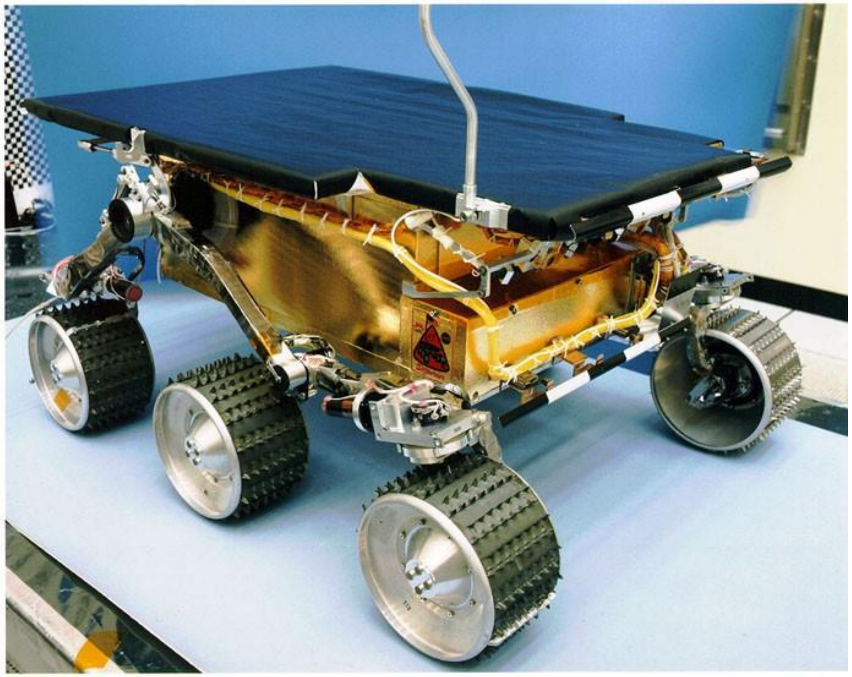
Legged robots

Field Robots: Mobile

Hilare II

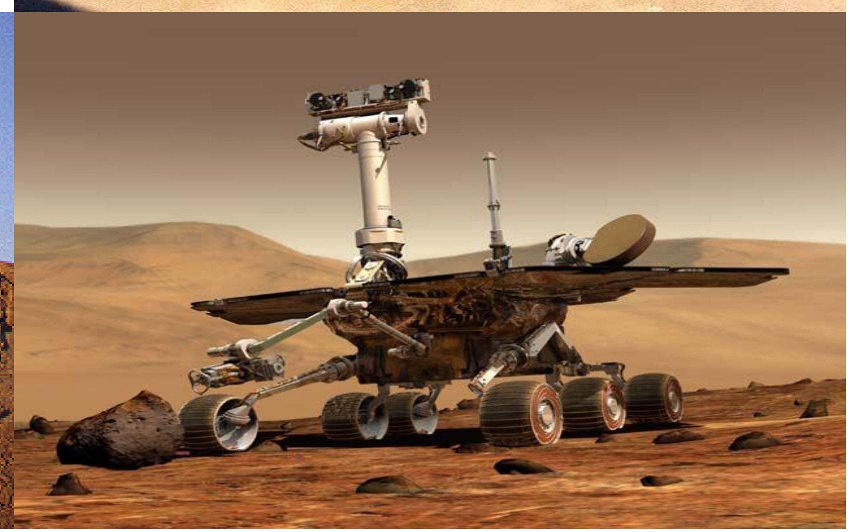
http://www.laas.fr/~matthieu/robots/

Sojourner Rover



NASA and JPL, Mars exploration

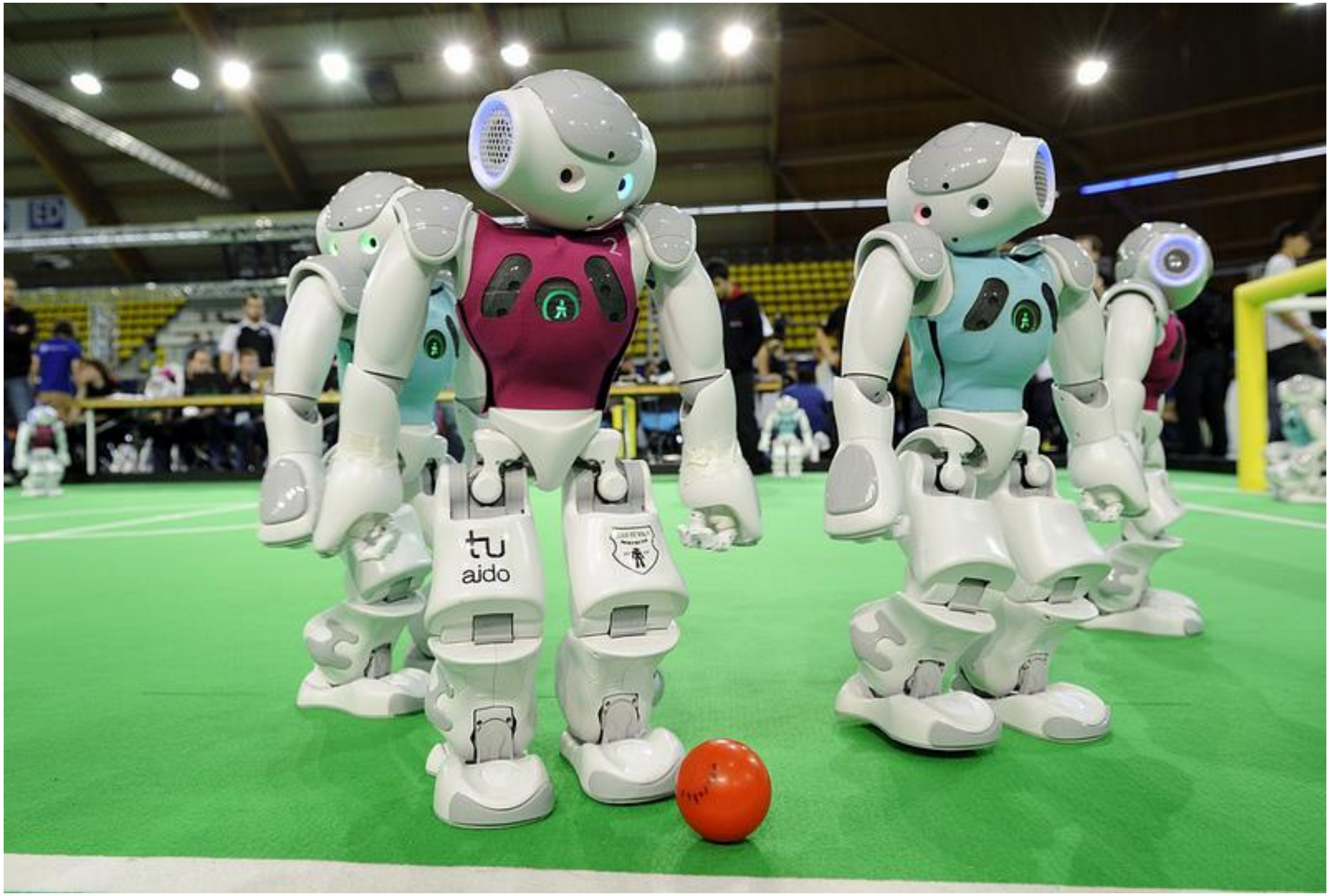
Field Robots: Autonomous



Service Robots

iRobot Scooba Robot

Entertainment Robots

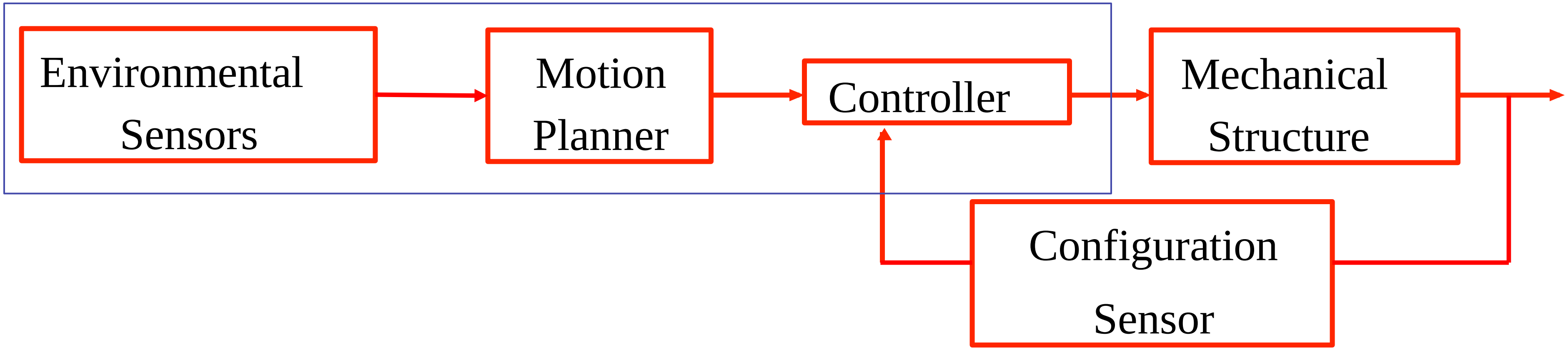


Why Use Robots?

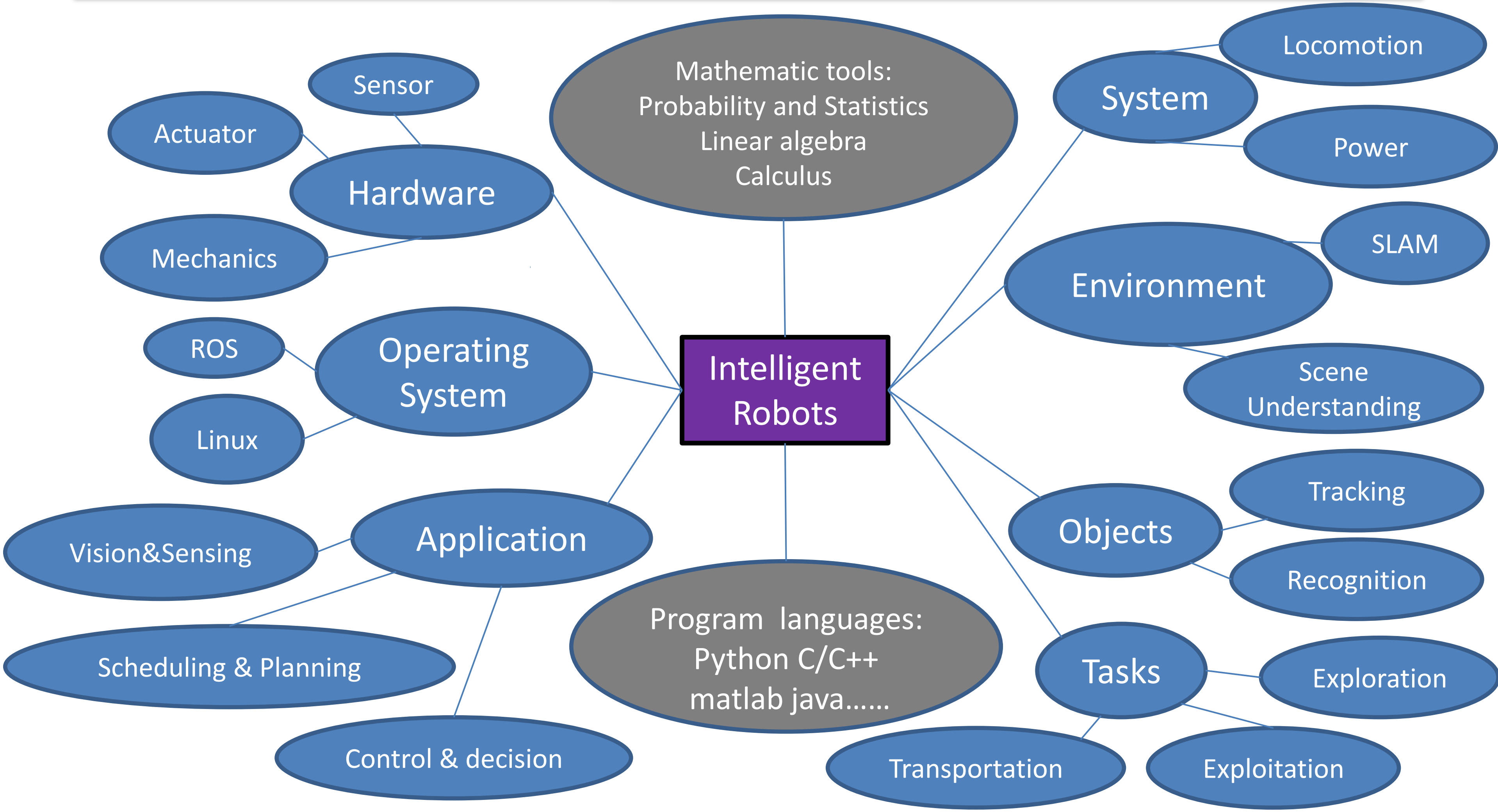
- Application in 4D environments
  - Dangerous
  - Dirty
  - Dull
  - Difficult
- 4A tasks
  - Automation
  - Augmentation
  - Assistance
  - Autonomous

Architecture of Robotic Systems

- Mechanical Structure
  - ➢ Kinematics and dynamics
- Actuators:
  - ➢ Electrical, hydraulic, pneumatic, artificial muscle
- Sensors
  - ➢ Passive：optical，infrared, electro-magnetic，acoustic，IMU，GPS
  - ➢ Active: laser, ultrasound, microwave
- Computation，communication, and control
- User interface and power unit



Framework


Mathematic tools:
Probability and Statistics
Linear algebra
Calculus
Sensor
Actuator
Hardware
Mechanics
ROS
Operating System
Linux
Vision&Sensing
Application
Scheduling & Planning
Control & decision
Intelligent Robots
Program languages:
Python C/C++
matlab java……
Locomotion
System
Power
SLAM
Environment
Scene Understanding
Tracking
Objects
Recognition
Tasks
Exploration
Transportation
Exploitation

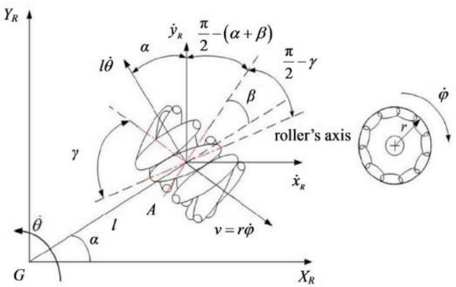
Outline

- Robot Types and Framework
- Problem Statement
- Related Areas
- History
- Optimization for Robots
- Algorithms
- Platforms

Problem Statement

- Problems：

  - **Motion Control**



$$\begin{bmatrix}\dot{x}_I\\ \dot{y}_I\\ \dot{\theta}\end{bmatrix}=-\left(\sqrt{2}/2\right)r\boldsymbol{J}^{+}\begin{bmatrix}\dot{\varphi}_1\\ \dot{\varphi}_2\\ \dot{\varphi}_3\\ \dot{\varphi}_4\end{bmatrix}$$

  - **Range Sensing**
  - **Mapping**
  - **Stereo Vision**
  - **Reconstruction**

$$\underbrace{\begin{pmatrix}P_x(k+1)\\ P_y(k+1)\\ V_x(k+1)\\ V_y(k+1)\end{pmatrix}}_{X(k+1)}=\underbrace{\begin{pmatrix}1&0&T&0\\0&1&0&T\\0&0&1&0\\0&0&0&1\end{pmatrix}}_{A}\underbrace{\begin{pmatrix}P_x(k)\\ P_y(k)\\ V_x(k)\\ V_y(k)\end{pmatrix}}_{X(k)}+\underbrace{\begin{pmatrix}0\\0\\ \sigma_v\\ \sigma_v\end{pmatrix}}_{w(k)}$$

$$\underbrace{\begin{pmatrix}P_x(k)\\ P_y(k)\end{pmatrix}}_{Y(k)}=\underbrace{\begin{pmatrix}1&0&0&0\\0&1&0&0\end{pmatrix}}_{C}\underbrace{\begin{pmatrix}P_x(k)\\ P_y(k)\\ V_x(k)\\ V_y(k)\end{pmatrix}}_{X(k)}+\underbrace{\begin{pmatrix}\sigma_p\\ \sigma_p\end{pmatrix}}_{v(k)}$$

  - **Tracking**
  - **Recognition**
  - **SLAM**

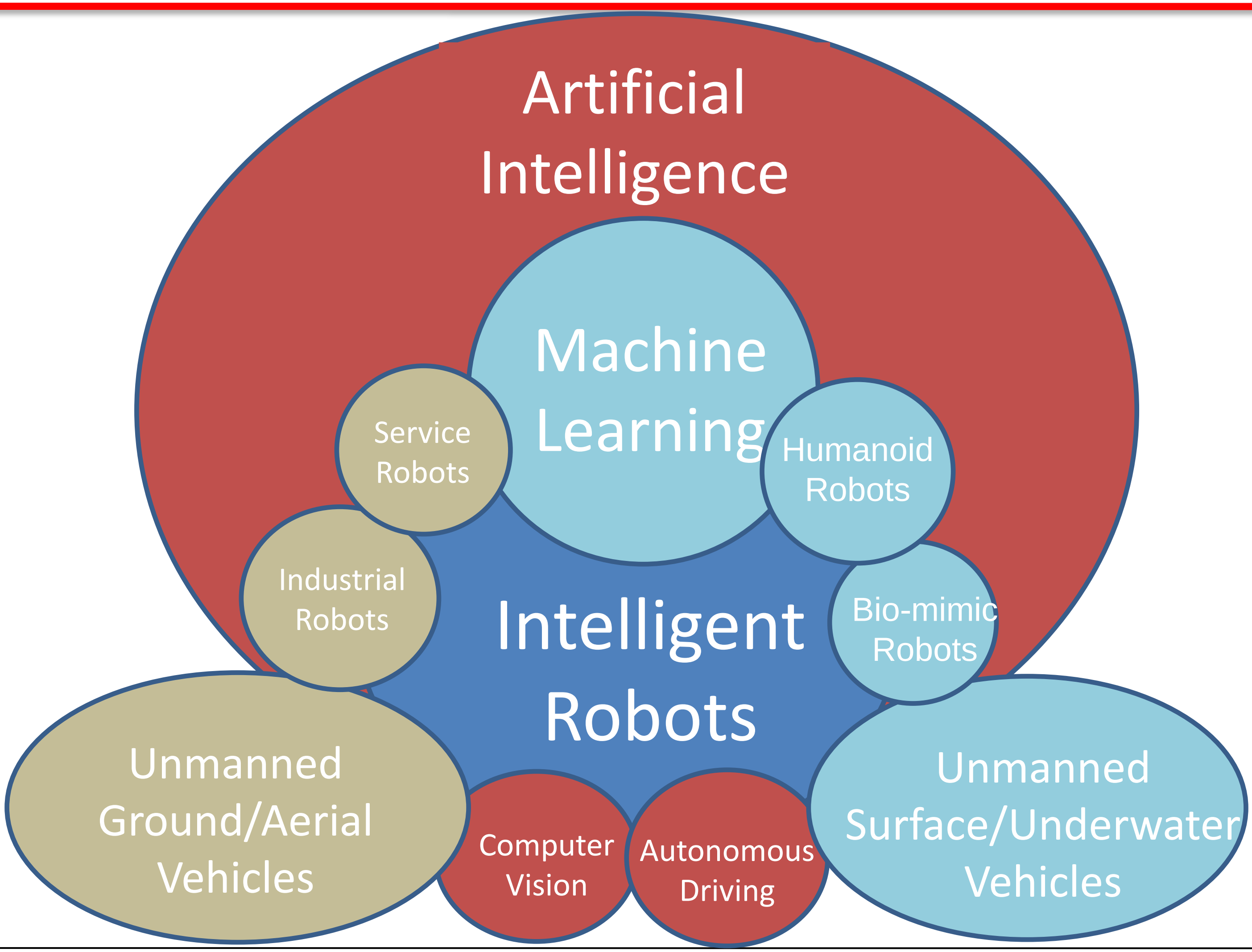
  - **Motion Planning**
  - **Path Planning**

$$f(n)=g(n)+h(n)$$

Outline

- Robot Types and Framework
- Problem Statement
- Related Areas
- History
- Optimization for Intelligent Robots
- Algorithms
- Examples

Related Areas



What is AI

- Knowledge representation
- Understanding natural language
- Learning
- Planning and problem solving
- Inference
- Search
- Vision

Learning and Evolution

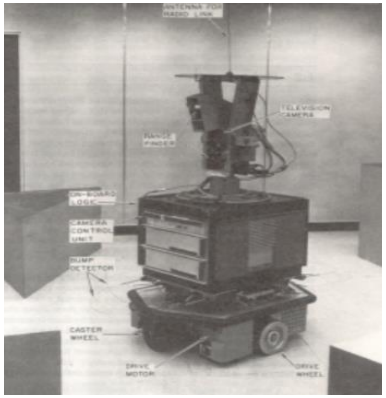
- Learning
  - Skills vs Task (Map acquisition)
- Learning Methods
  - Learning by instruction
  - Learning by imitation
  - Learning by skill transfer
  - Learning by trial-and-error
- Evolution and adaptation

Outline

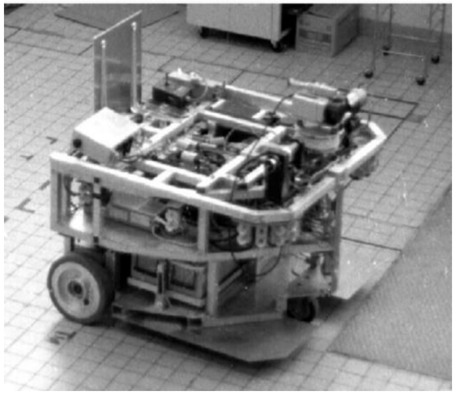
- Robot Types and Framework
- Problem Statement
- Related Areas
- History
- Optimization for Intelligent Robots
- Algorithms
- Examples

History

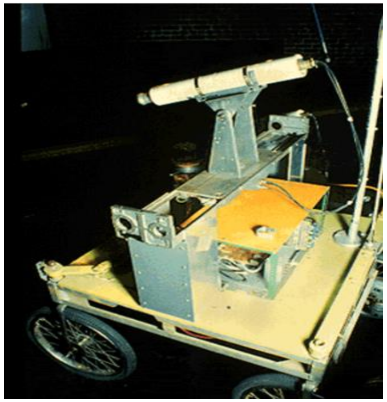
1950 Grey Walter



1969 Shakey



1977 Hilare 1



1979 Stanford Car

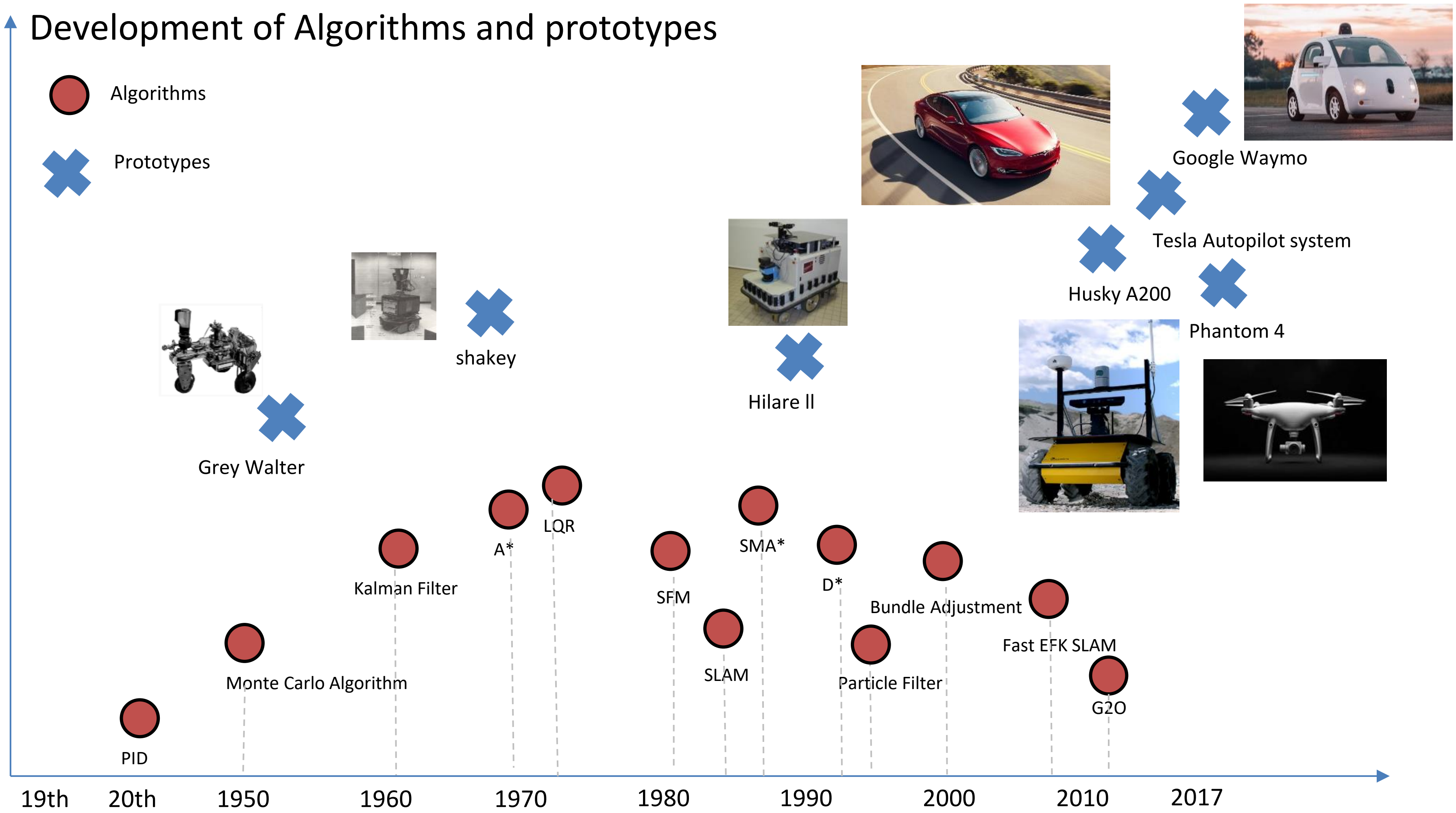
1990 Hilare ll

1998-2000 RAMS

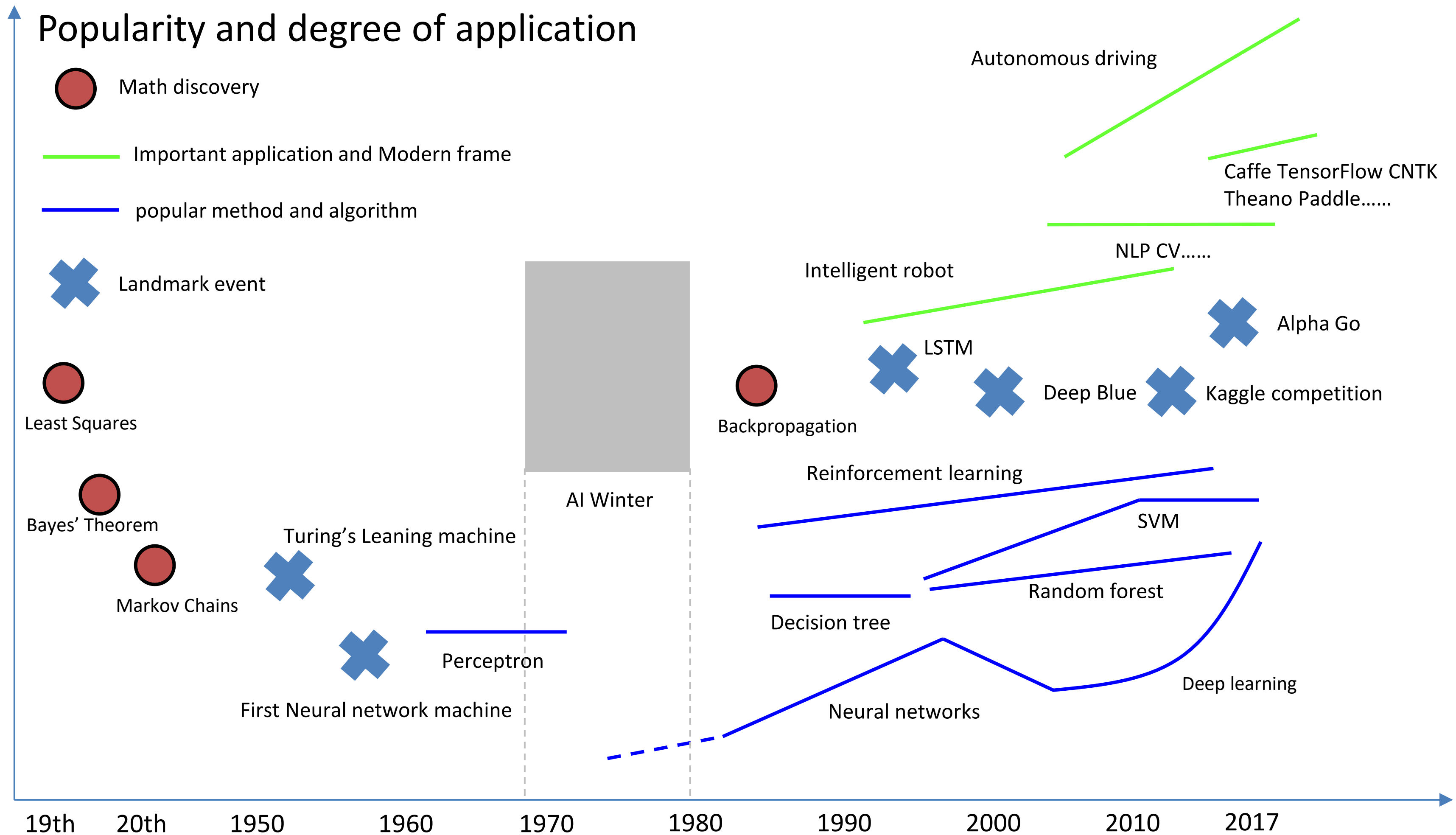
2011 husky a200

2017 Waymo

History


Development of Algorithms and prototypes
Algorithms
Prototypes
Google Waymo
Tesla Autopilot system
Husky A200
Phantom 4
shakey
Hilare II
Grey Walter
LQR
A*
Kalman Filter
SMA*
SFM
D*
Bundle Adjustment
SLAM
Fast EFK SLAM
Particle Filter
Monte Carlo Algorithm
G2O
PID
19th
20th
1950
1960
1970
1980
1990
2000
2010
2017

History



Popularity and degree of application
Math discovery
Important application and Modern frame
popular method and algorithm
Landmark event
Least Squares
Bayes' Theorem
Markov Chains
Turing's Leaning machine
First Neural network machine
Perceptron
AI Winter
Backpropagation
LSTM
Deep Blue
Kaggle competition
Alpha Go
Intelligent robot
NLP CV……
Caffe TensorFlow CNTK
Theano Paddle……
Autonomous driving
Reinforcement learning
SVM
Random forest
Decision tree
Neural networks
Deep learning
19th
20th
1950
1960
1970
1980
1990
2000
2010
2017

Outline

- Robot Types and Framework
- Problem Statement
- Related Areas
- History
- Optimization for Intelligent Robot
- Algorithms
- Platforms

Optimization for Intelligent Robots

- **Perception**
  - Estimation/Prediction/Classification
  - Fusion/Mapping /Reconstruction
  - Scene understanding
- **Control**
  - Time optimal control
  - Optimal/robust control
  - Adaptive control
- **Decision**
  - Utility Function
  - Q-value
- **Planning**
  - Multi-objectives
  - Constraints
- **Scheduling**
  - Multi-tasks
  - Resource allocation

What is optimization?

- Finding (one or more) minimizer of a function subject to constraints

$$\arg\min_{x} f_0\,(x)$$

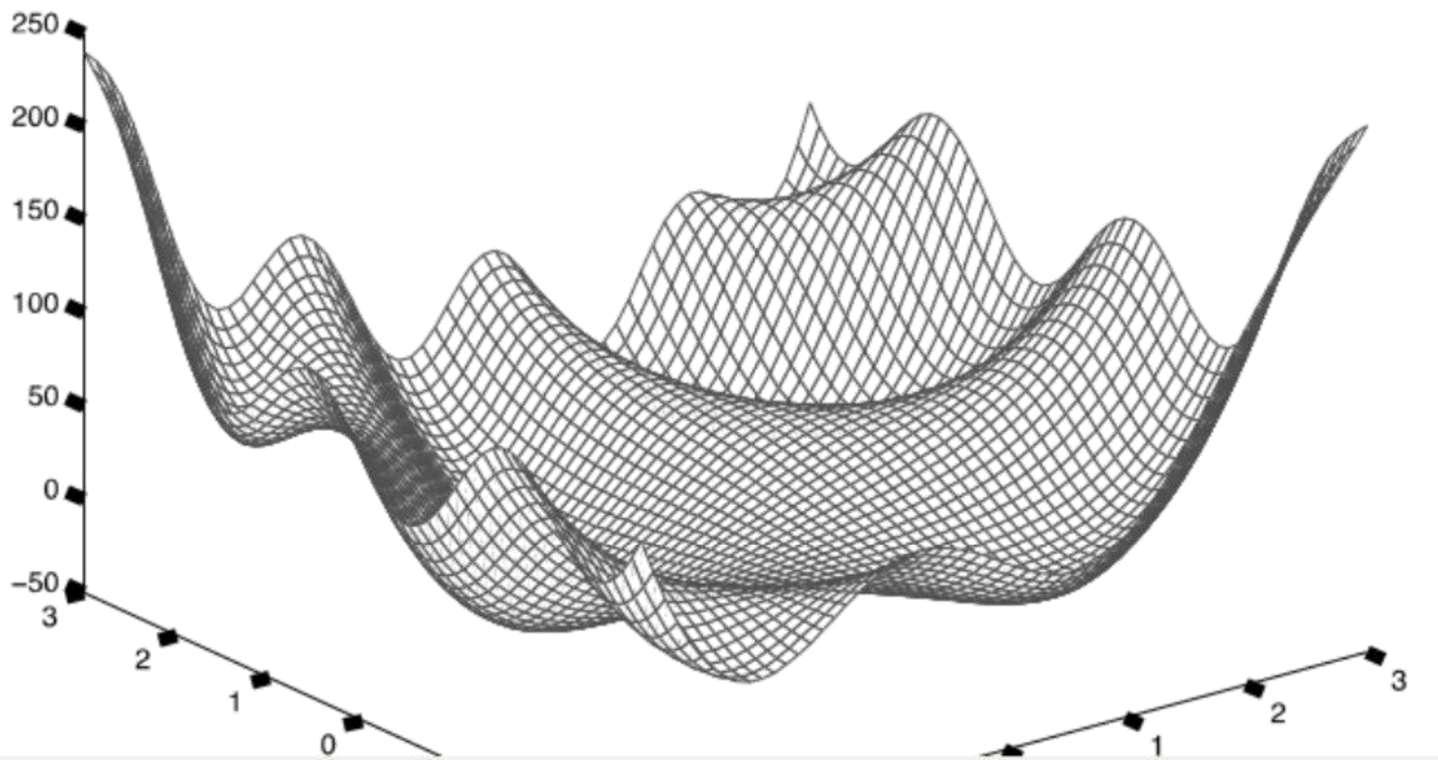
$$s.t.\, f_i(x) \leq 0, i = \{1, \ldots, k\}$$

$$f_j(x) = 0, j = \{1, \ldots, l\}$$

- Most of the machine learning problems are, in the end, optimization problems

General Problem

- Minimize f(x)



Optimization

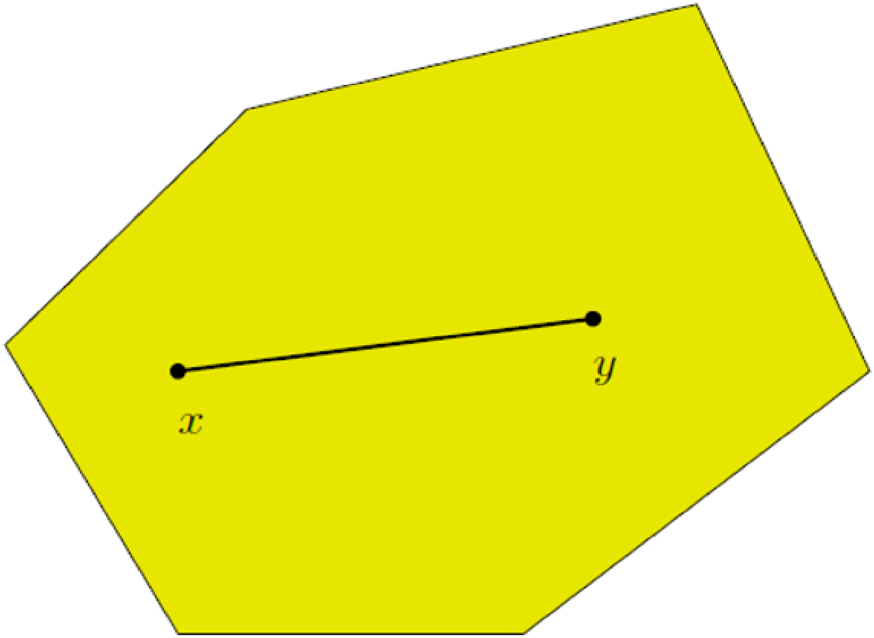
- Convex
  - Unconstrained optimization
  - Constrained optimization
- Non-convex
  - Heuristic algorithms
  - Approximate methods

What is Convex?

- Convex sets

Def: A set C ⊆ R is convex if for x,y ∈ C; a ∈ [0,1]

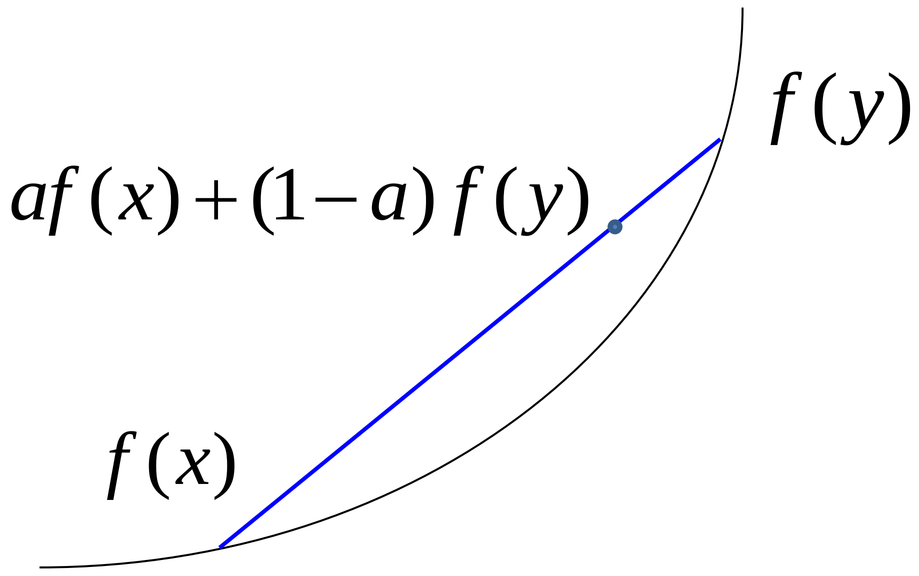
$$ax + (1-a)y \in C$$



What is Convex?

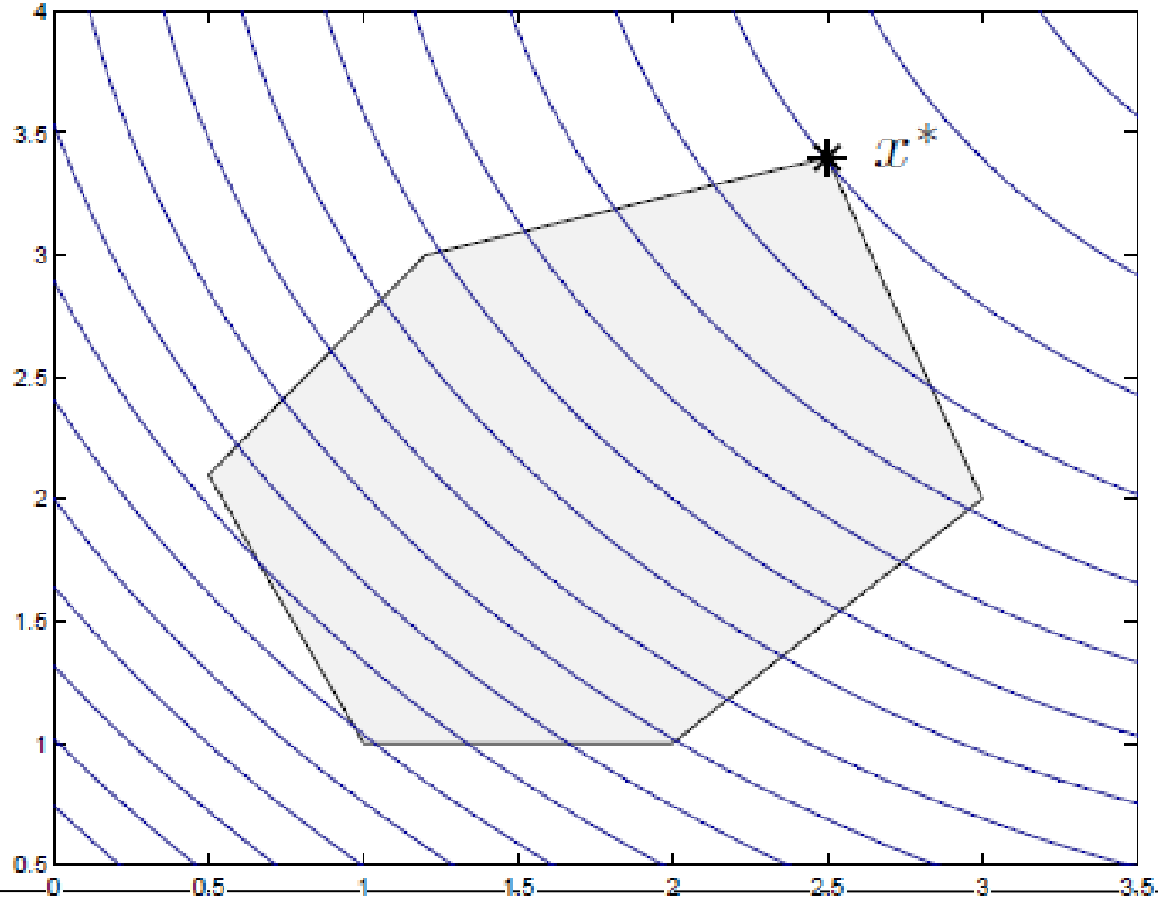
- Convex functions

$$f(ax+(1-a)y) \le af(x)+(1-a)f(y)$$



Convex Optimization

- Local minimizer = Global minimizer



Convex Optimization

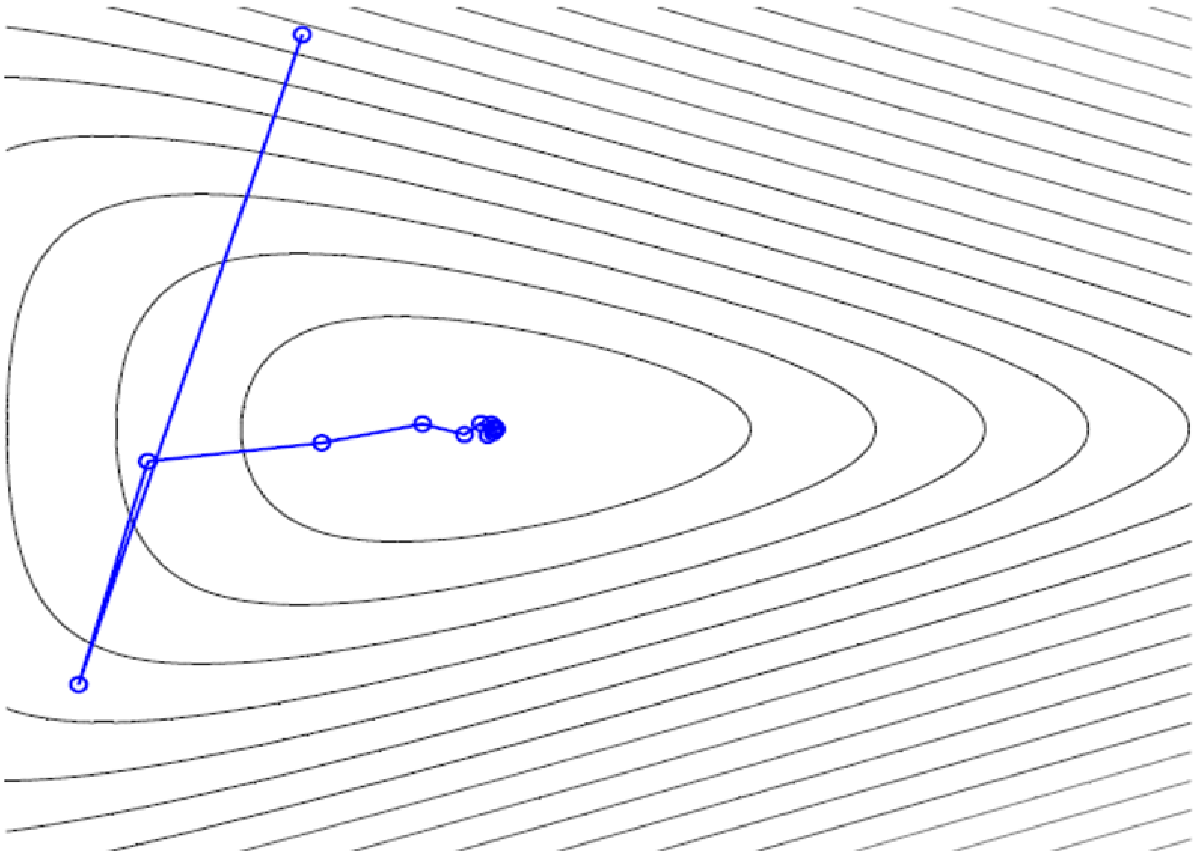
- Unconstrained optimization
  - Gradient descent
  - Newton's method
- Constrained optimization
  - Lagrange function
  - General methods

Convex optimization

- Unconstrained optimization
  - Gradient descent
  - Newton’s method
  - Batch learning
  - Stochastic Gradient Descent
- Constrained optimization
  - Lagrange function
  - General methods

Gradient Descent

$$f(x_{t+1}) = f(x_t) - \eta \nabla f(x_t)^T (x - x_t)$$



Newton's Method

- Idea: use a second-order approximation to function

$$f(x+\Delta x) \approx f(x) + \nabla f(x)^T \Delta x + \frac{1}{2}\Delta x^T \nabla^2 f(x) \Delta x$$

- Choose Δx to minimize above:

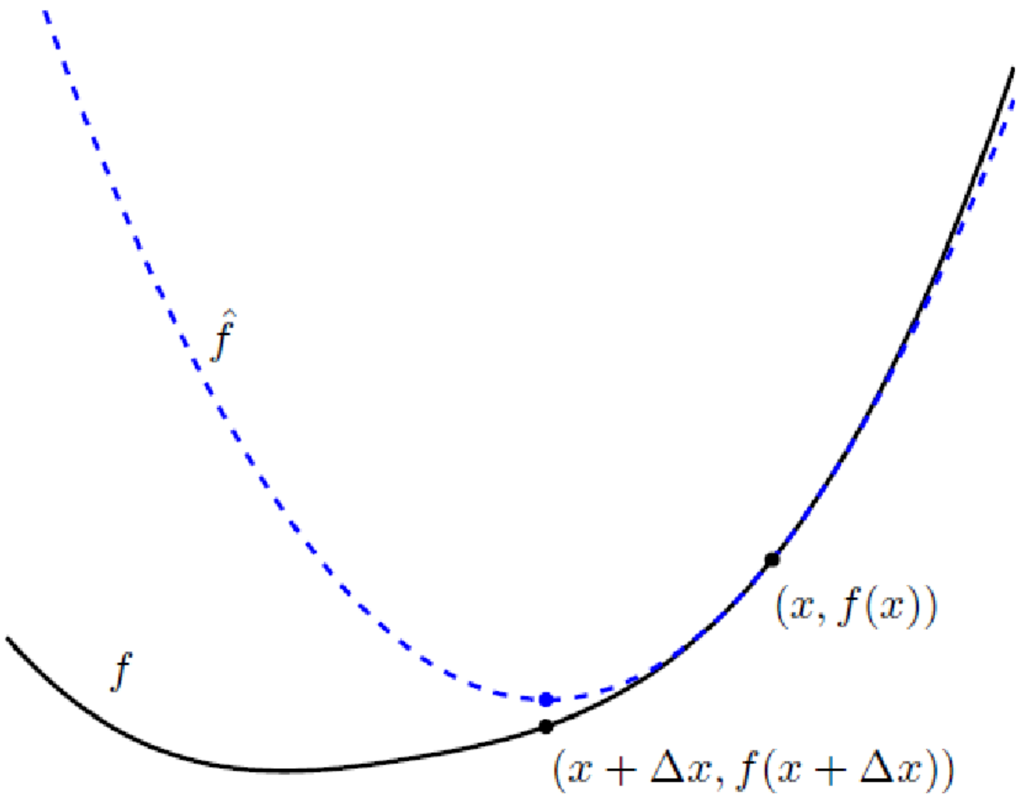
$$\Delta x = -[\nabla^2 f(x)]^{-1} \nabla f(x)$$

- This is descent direction:

$$\nabla f(x)^T \Delta x = -\nabla f(x)^T [\nabla^2 f(x)]^{-1} \nabla f(x) < 0$$

Newton's Method

$\hat{f}$ is 2-order approximation, f is true function.



Convex Optimization

- Unconstrained optimization
  - Gradient descent
  - Newton's method
  - Batch learning
  - Stochastic Gradient Descent
- Constrained optimization
  - Lagrange Function
  - General methods

Lagrange Function

- Start with optimization problem:

$$\arg\min_{x} f_0(x)$$

$$s.t. f_i(x) \le 0, i = \{1, \dots, k\}$$

$$f_j(x) = 0, j = \{1, \dots, l\}$$

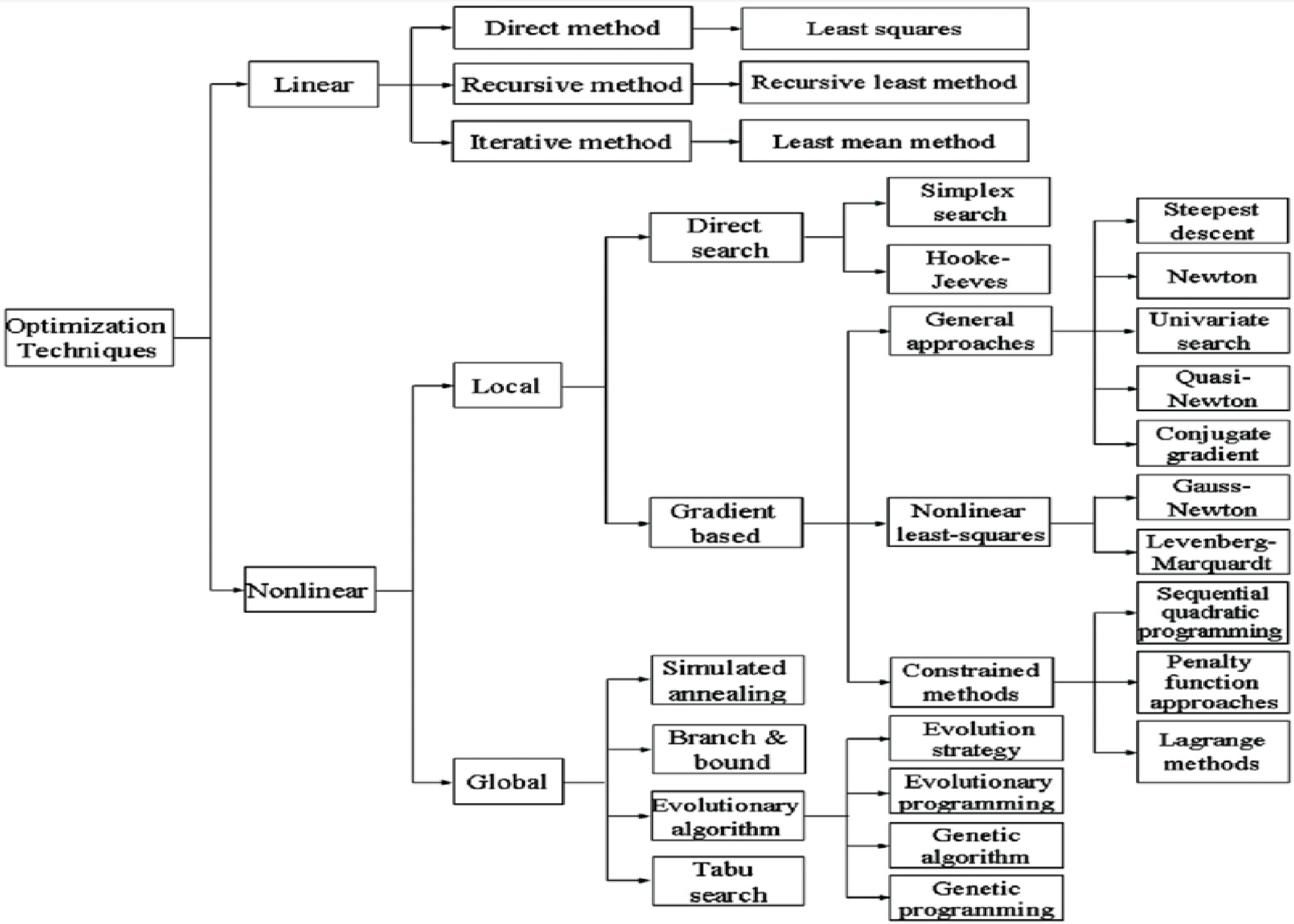
- Is equivalent to min-max optimization:

$$\arg\min_{x}[\sup_{\lambda>0,\nu}(f_0(x) + \sum_{i=1}^{k} \lambda_i f_i(x) + \sum_{j=1}^{l} \nu_j h_j(x))]$$

Optimization

- Convex
  - Unconstrained optimization
  - Constrained optimization
- Non-convex
  - Heuristic algorithms
  - Approximate methods

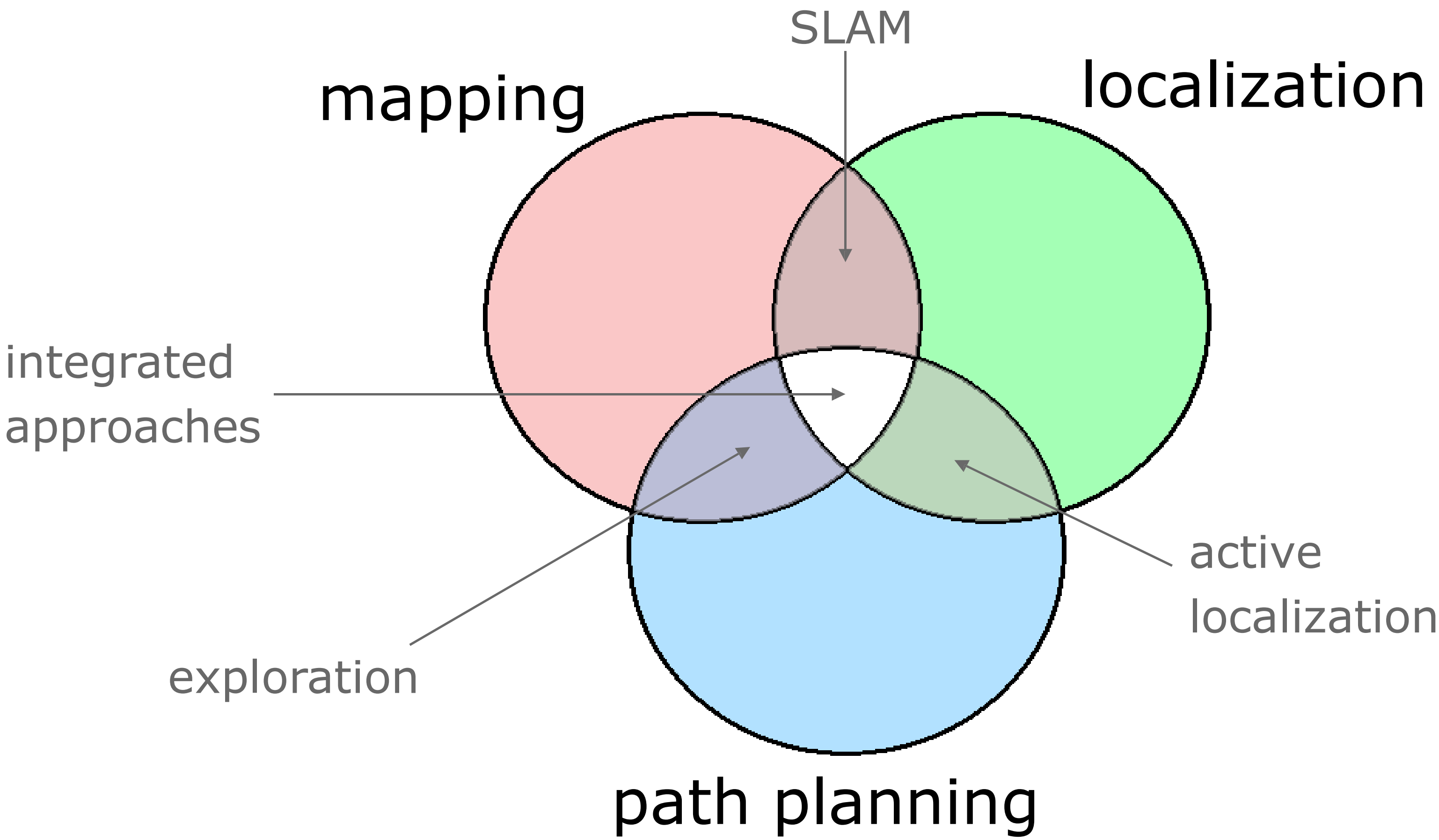
Optimization Methods



Outline

- Robot Types and Framework
- Problem Statement
- Related Areas
- History
- Optimization for Intelligent Robots
- Algorithms
- Platforms

Tasks of Mobile Robots



Algorithms

Control & Perception

- PID, LQR, Sliding Mode
- Kalman Filter
- Stereo Vision, Structure from Motion

Motion and Path Planning

- A*
- SMA*
- D*

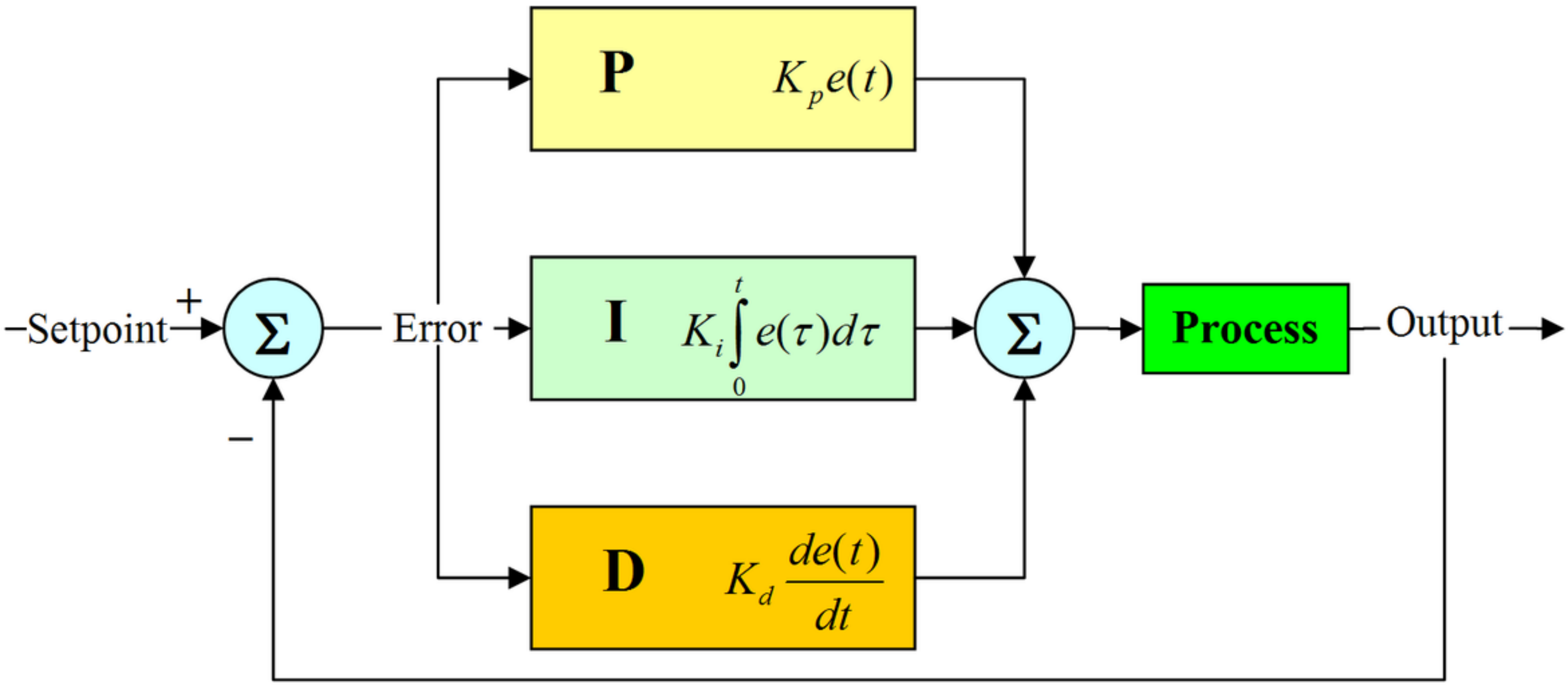
Localization & Mapping

- Recursive Bayesian Estimation
- Gaussian Particle Filter, Monte Carlo localization
- Bundle Adjustment, Graph Optimization

Reconstruction & Exploration

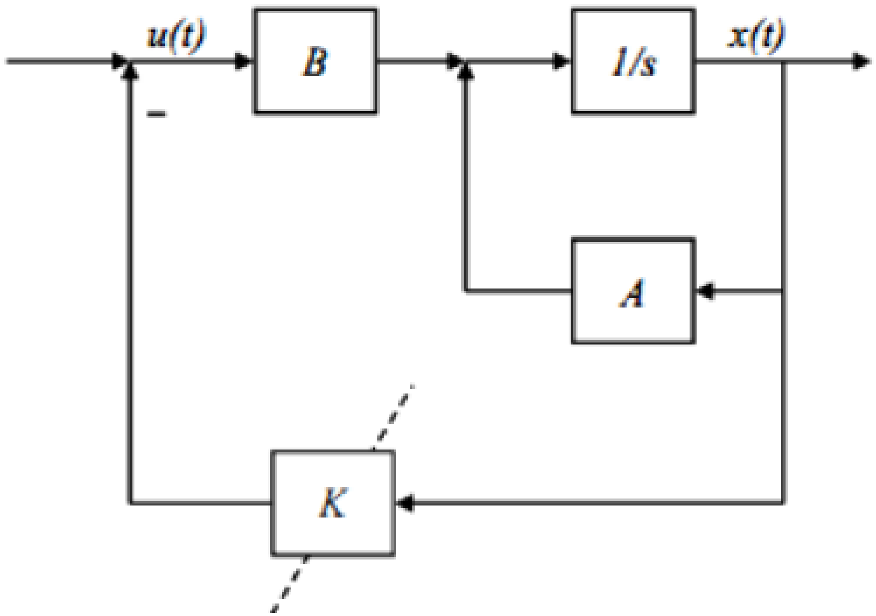
- ICP
- igExplore

PID



P
$K_p e(t)$
I
$K_i \int_0^t e(\tau)d\tau$
D
$K_d \frac{de(t)}{dt}$
Setpoint
+
−
Σ
Error
Σ
Process
Output

LQR



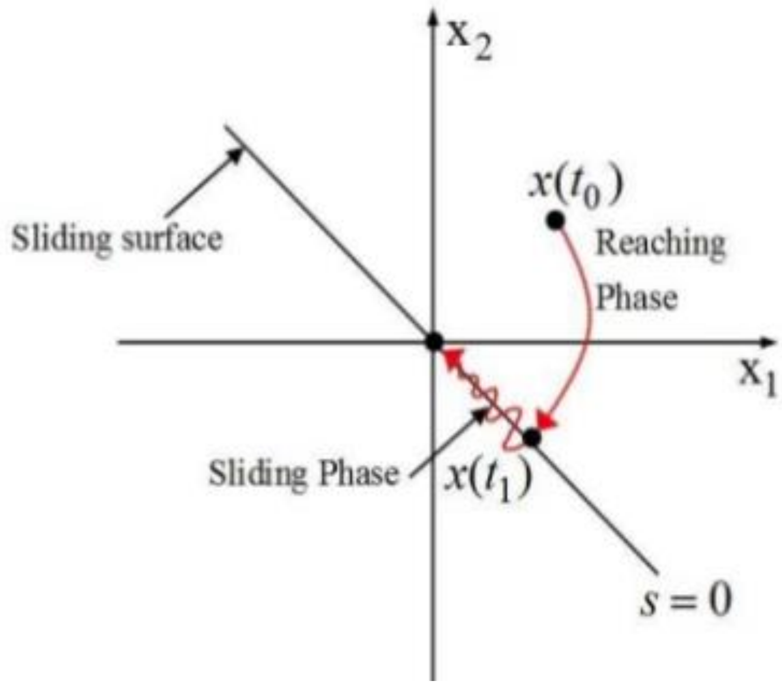
RL in Machine Learning

State space function: $\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}$

$u = -Kx$

Optimize function: $J = \frac{1}{2} \int_0^\infty x^T Q x + u^T R u \, dt$

Sliding Mode



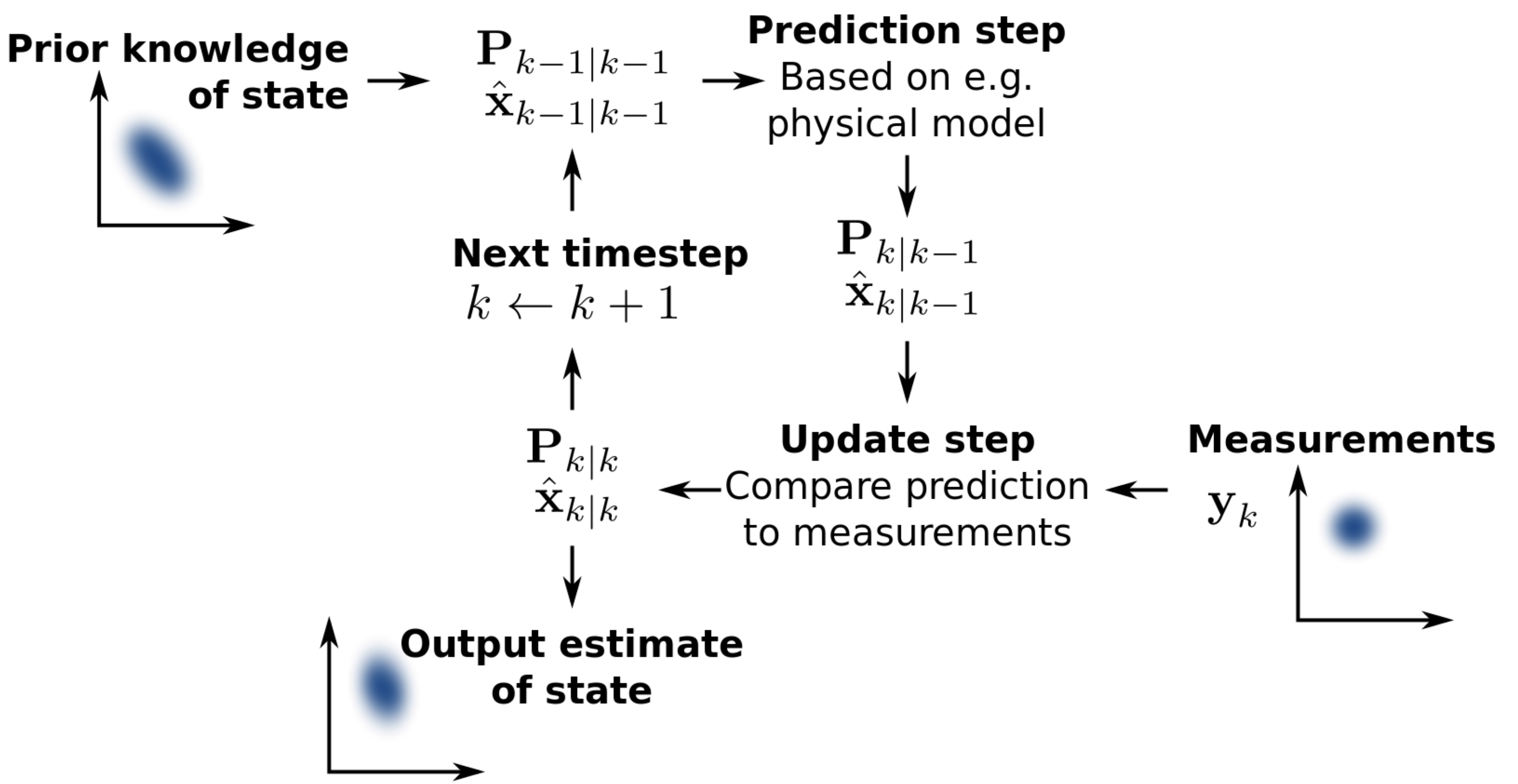
System state equation: $\dot{x} = f(x, u, t)$

Determine switching function: $S(x), S \in R$

Control law:

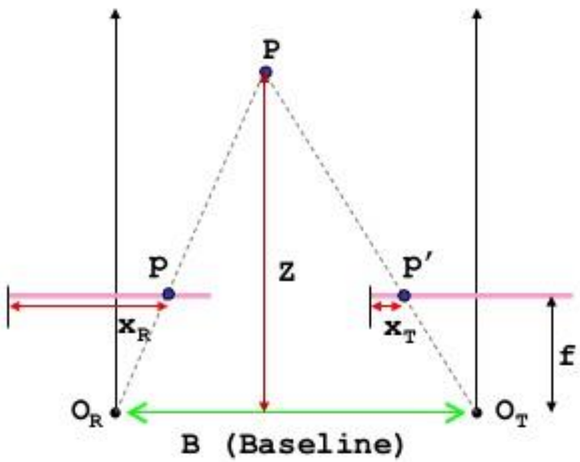
$$u = \begin{cases} u^+(x), S(x) > 0 \\ u^-(x), S(x) < 0 \end{cases}$$

Kalman Filter



Stereo Vision

Disparity and depth



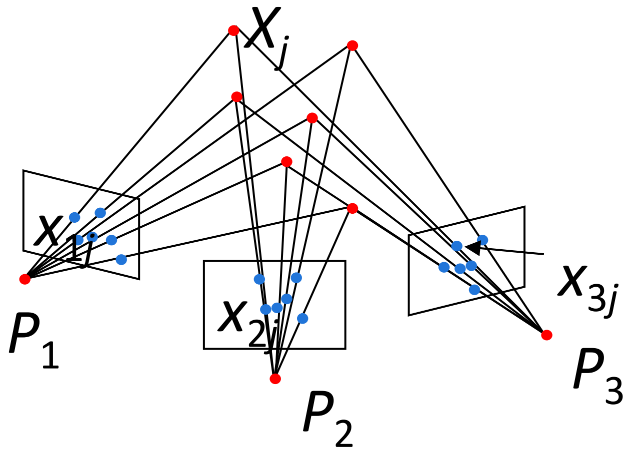
With the stereo rig in standard form and by considering similar triangles (PO_RO_T and Ppp'):

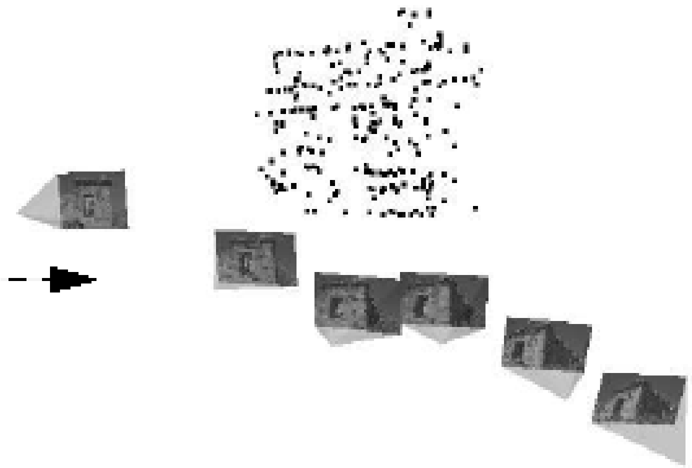
$$\frac{b}{Z} = \frac{(b + x_T) - x_R}{Z - f} \Rightarrow Z = \frac{b \cdot f}{x_R - x_T} = \frac{b \cdot f}{d}$$

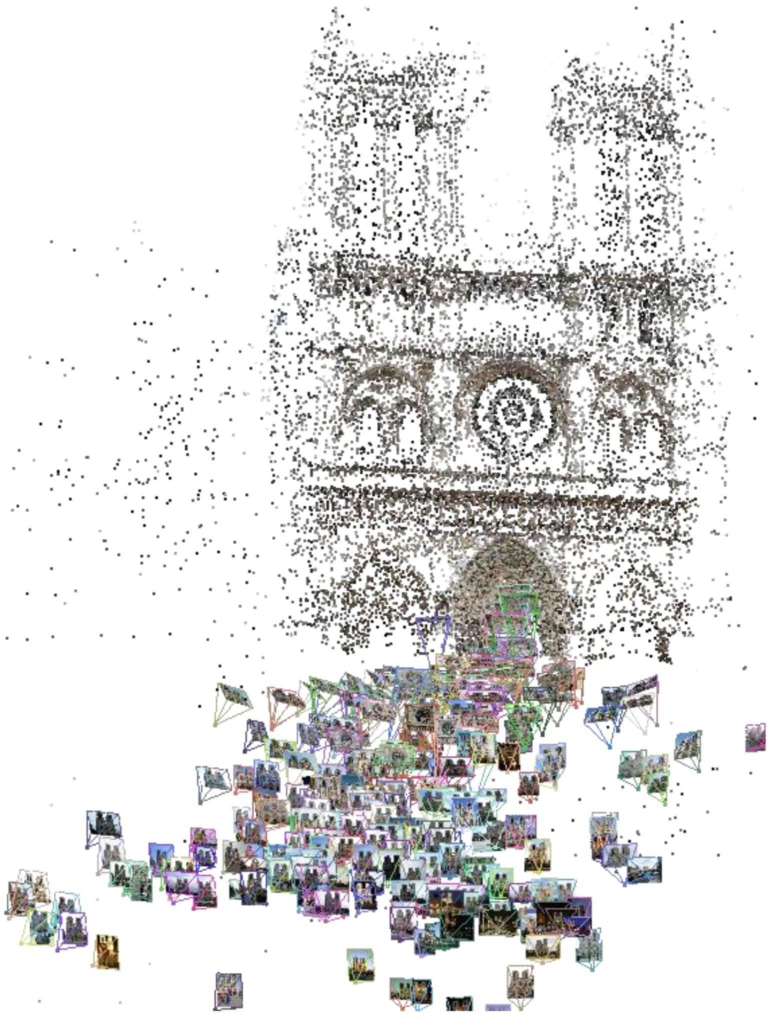
$x_R - x_T$ is the **disparity**

Stefano Mattoccia

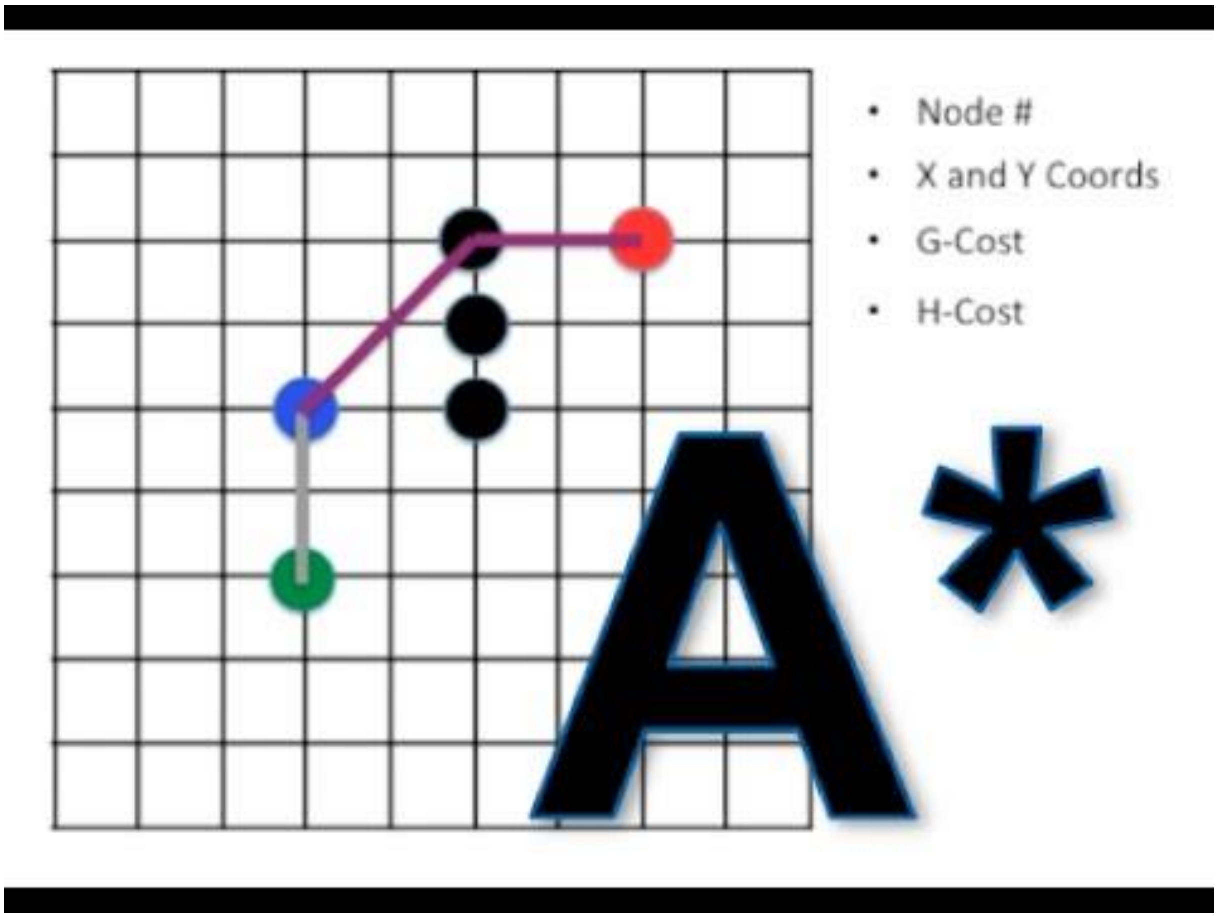
Structure from Motion



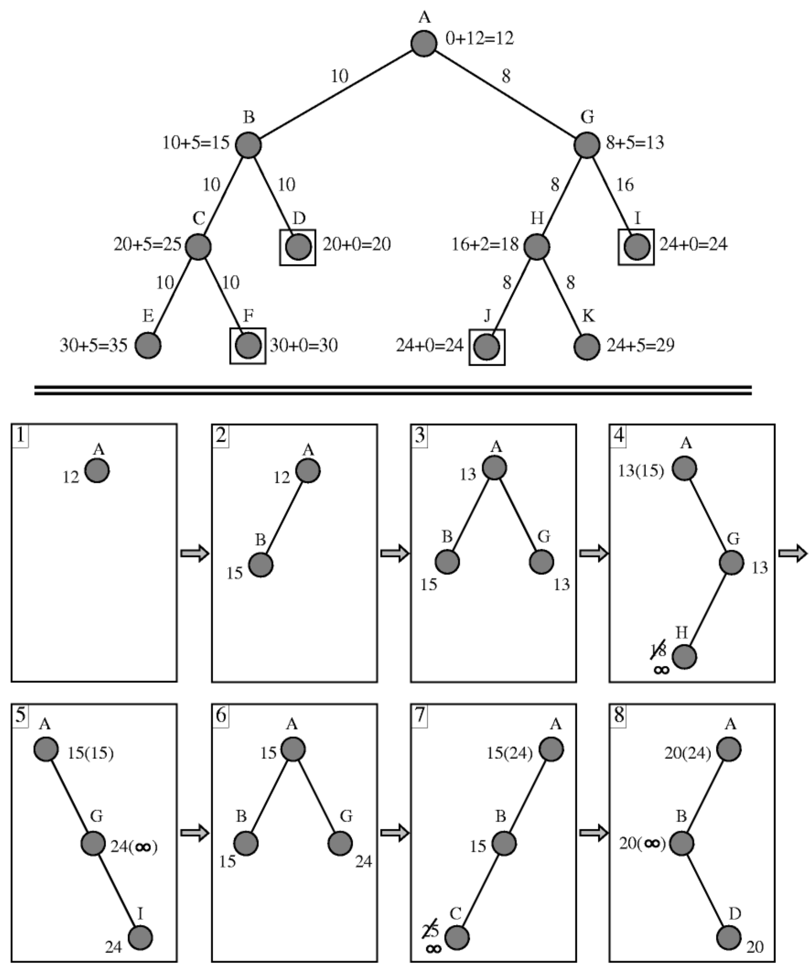




A*



SMA*



A
0+12=12
10
8
B
10+5=15
G
8+5=13
10
10
8
16
C
D
H
I
20+5=25
20+0=20
16+2=18
24+0=24
10
10
8
8
E
F
J
K
30+5=35
30+0=30
24+0=24
24+5=29
1
A
12
2
A
12
B
15
3
A
13
B
G
15
13
4
A
13(15)
G
13
H
18
∞
5
A
15(15)
G
24(∞)
I
24
6
A
15
B
G
15
24
7
A
15(24)
B
15
C
25
∞
8
A
20(24)
B
20(∞)
D
20

D* Lite

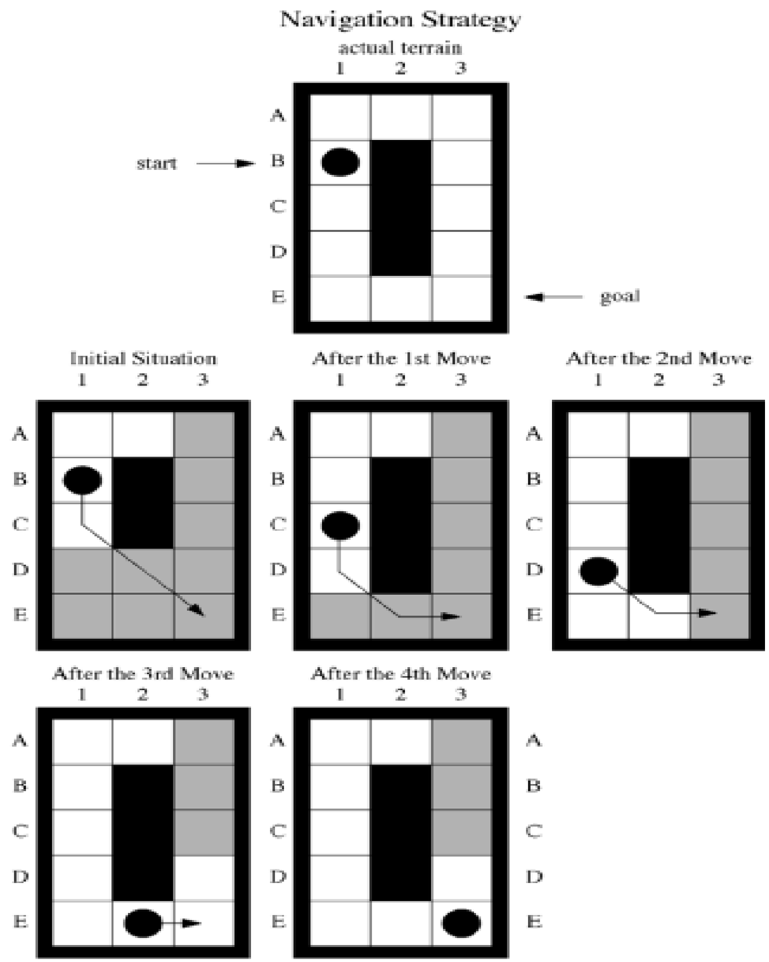
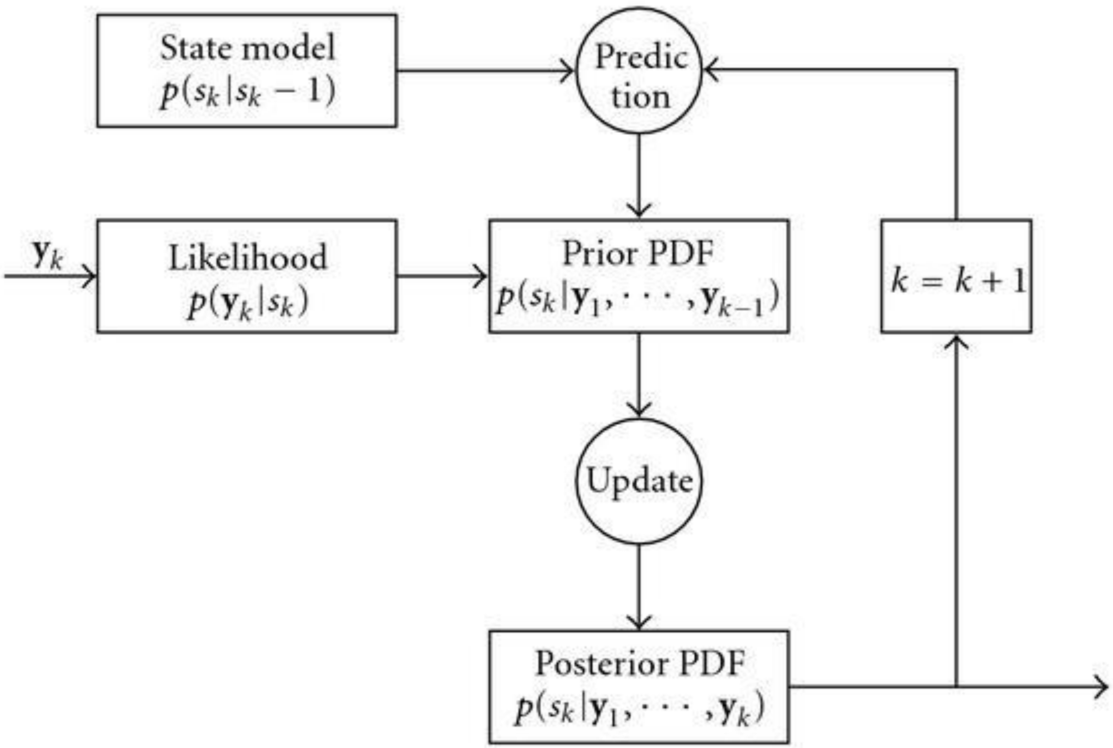
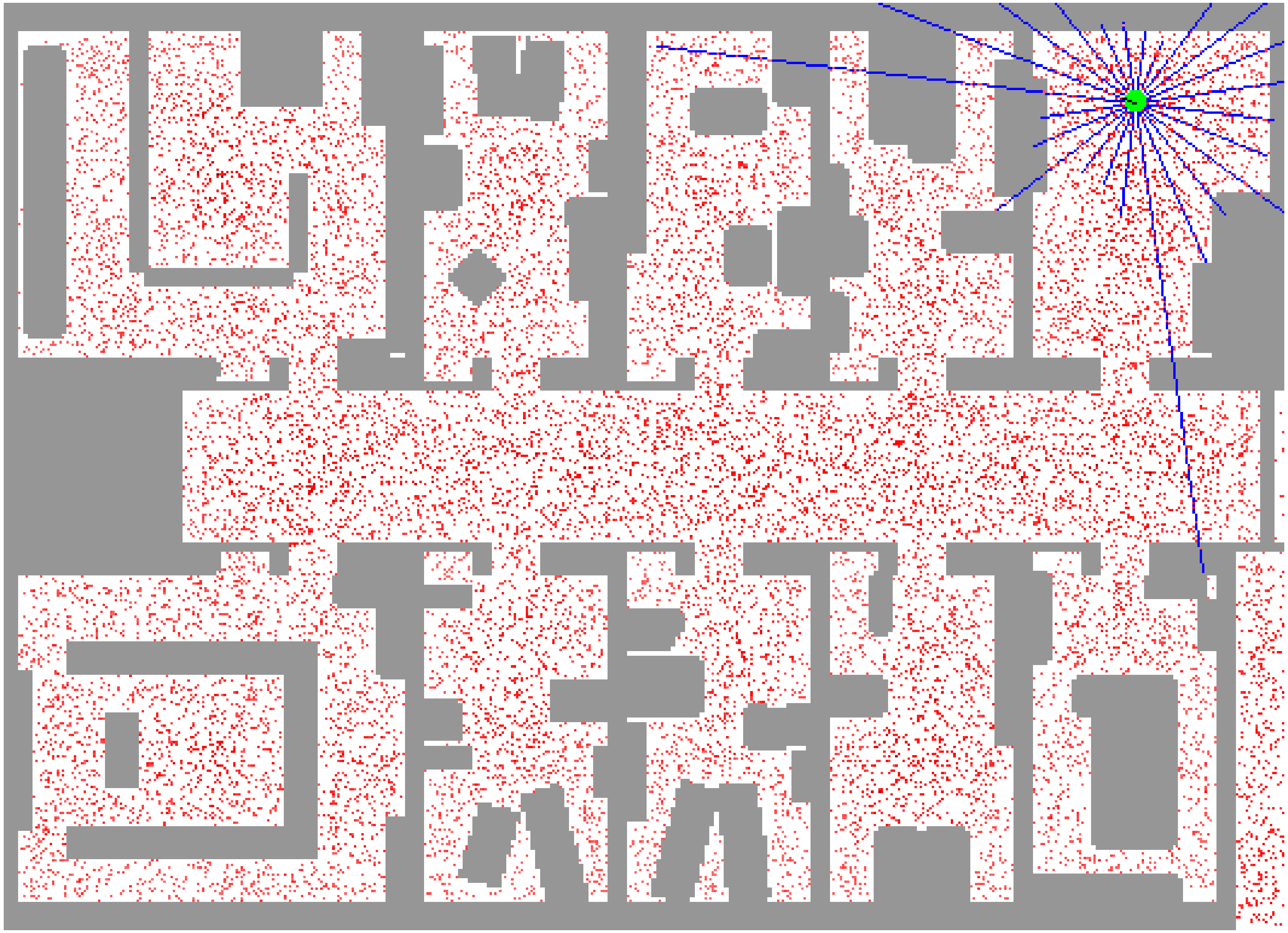


Fig. 1. Illustration of the navigation strategy. Gray cells are cells with unknown blockage status.

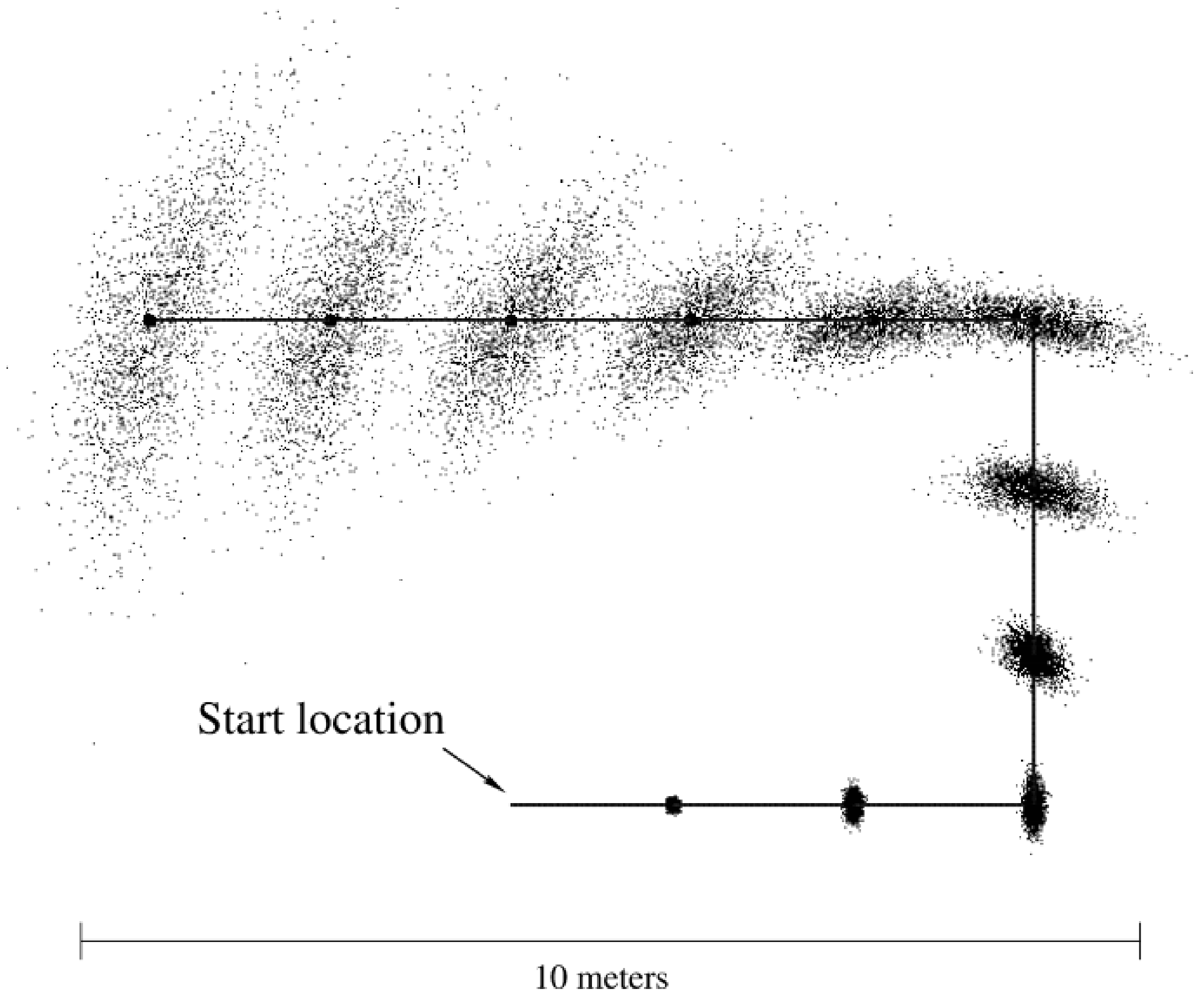
Recursive Bayesian estimation



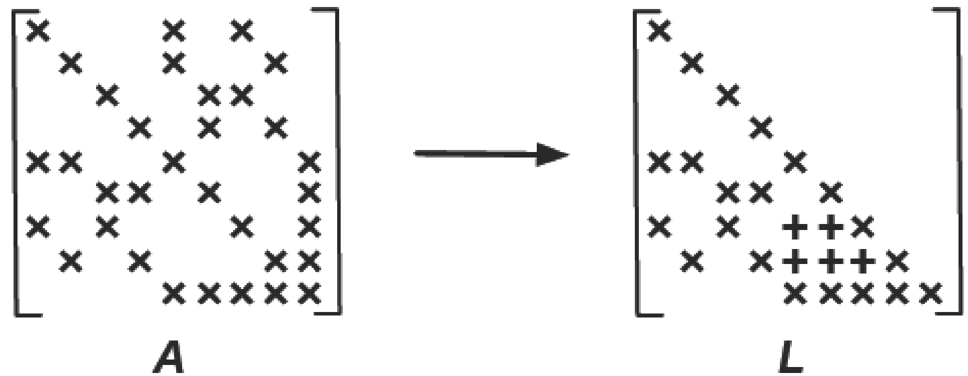
Gaussian Particle Filter

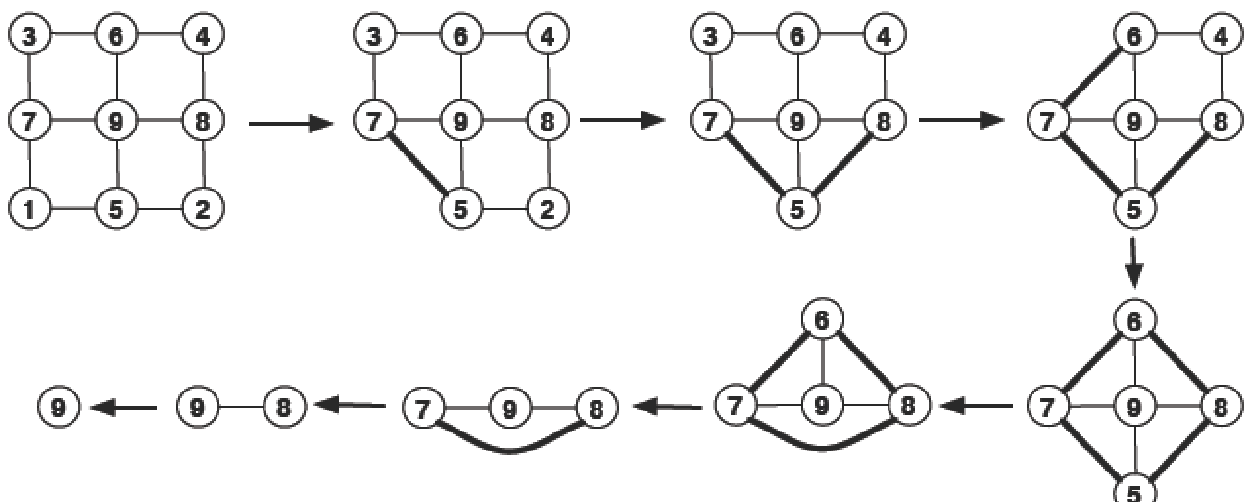


Monte Carlo localization

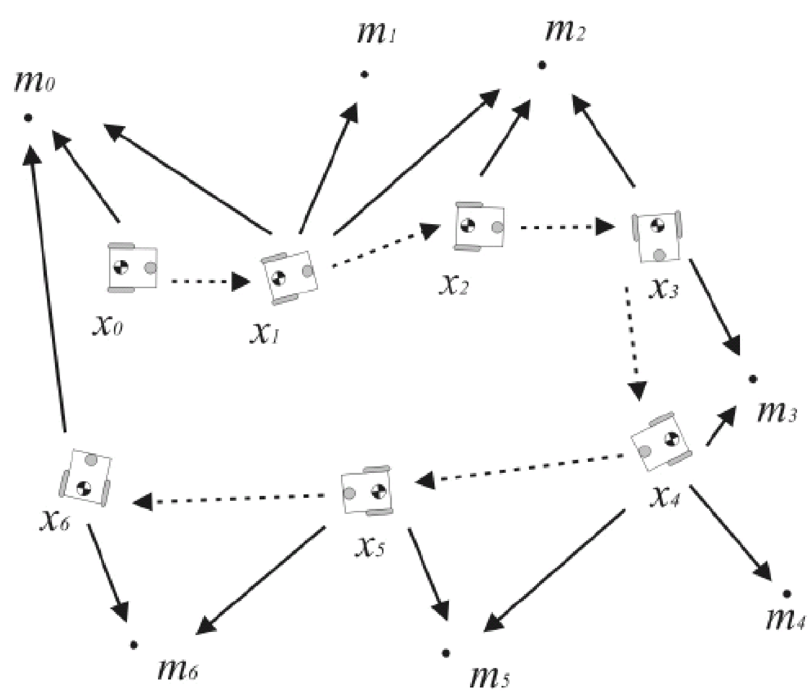


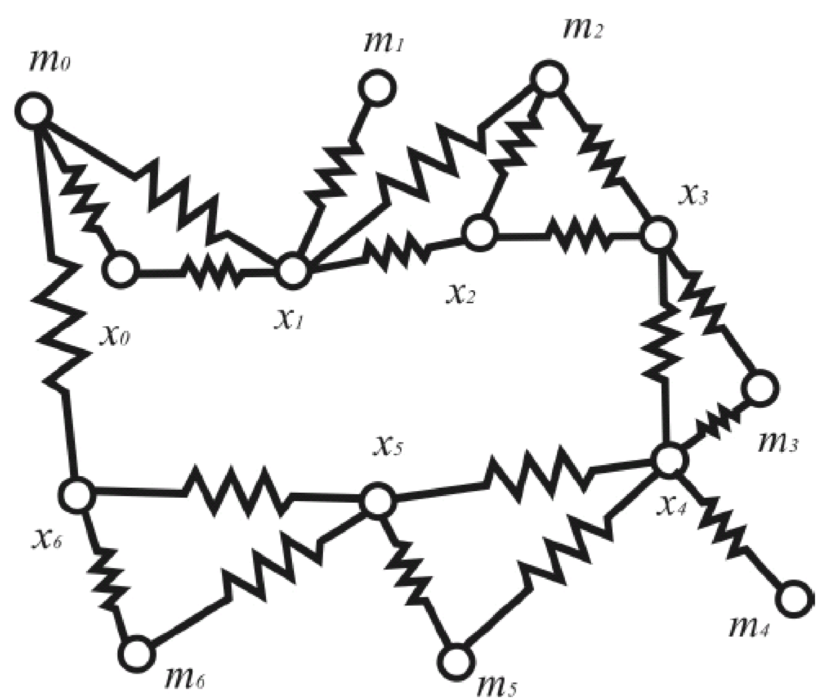
Bundle Adjustment



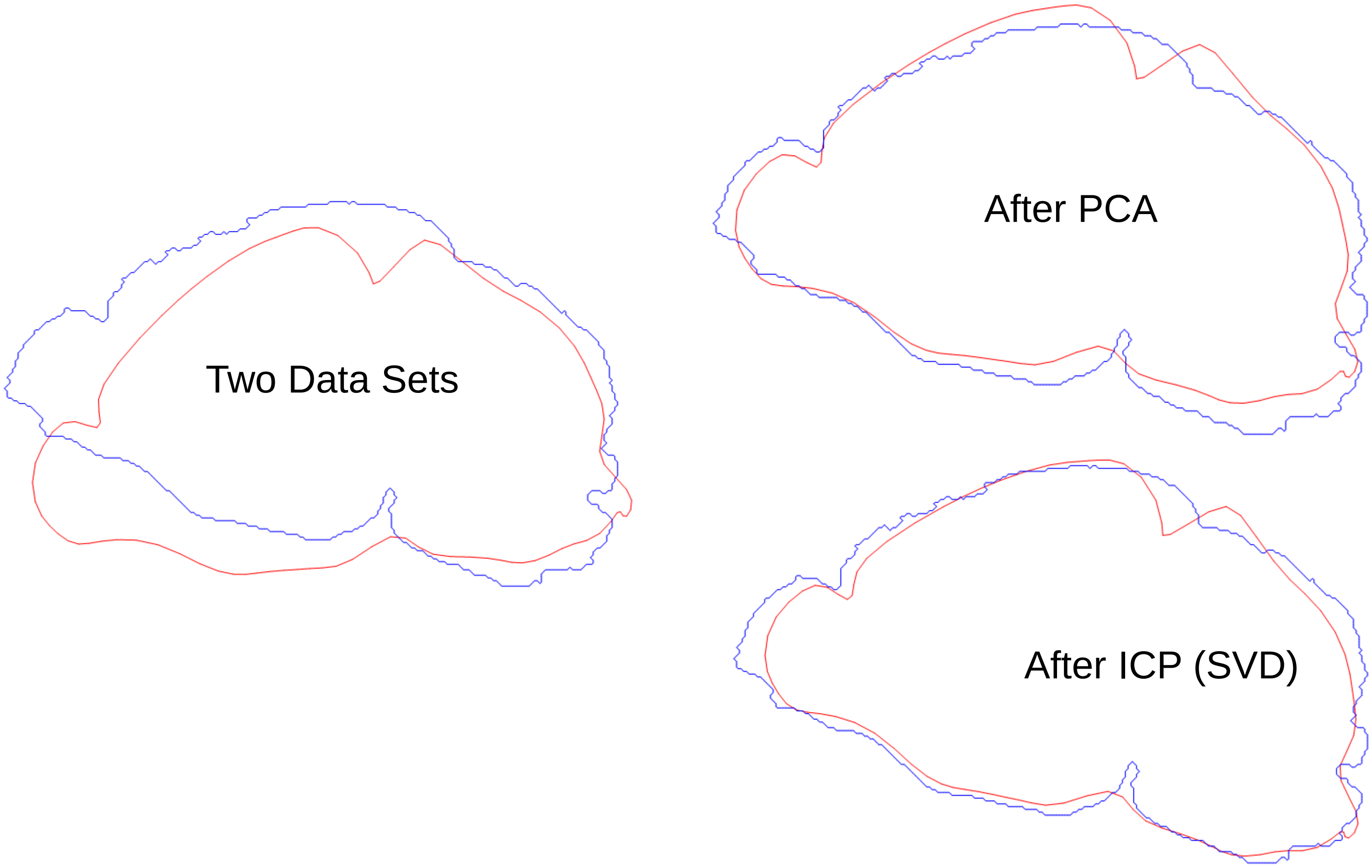


Graph Optimization

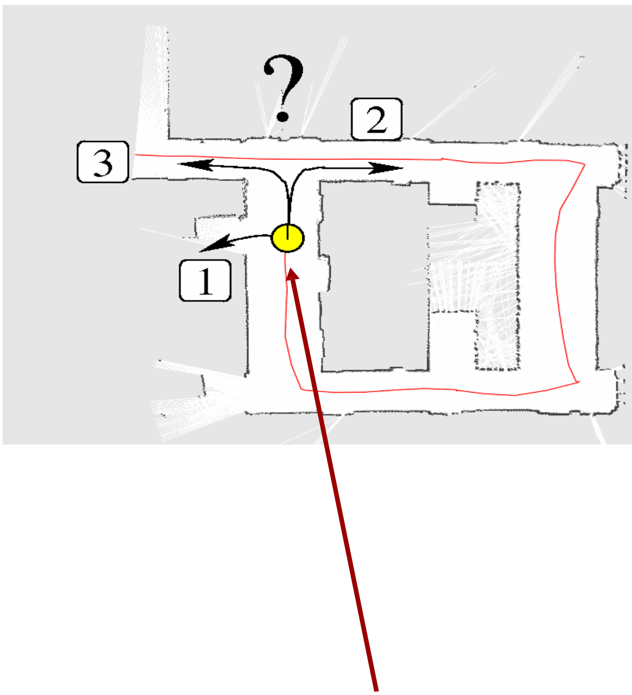




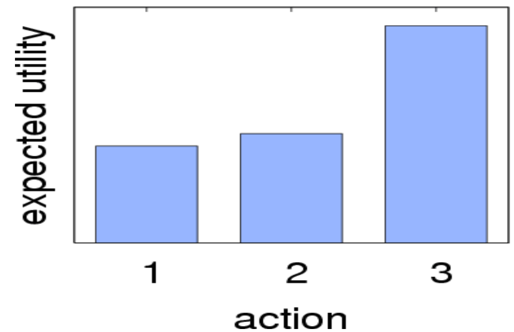
Iterative Closest Points

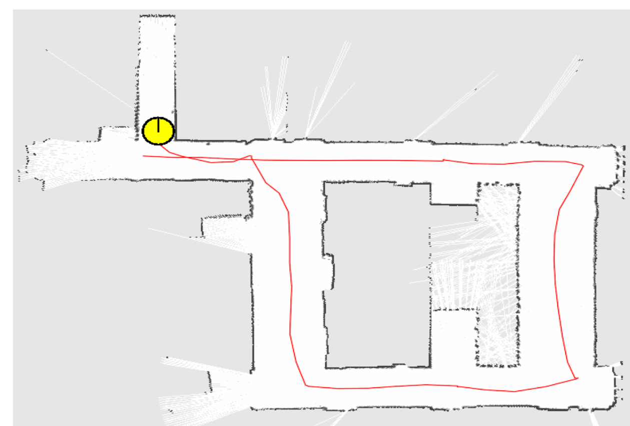


Information Gain Based Exploration



high pose
uncertainty





Outline

- Robot Types and Framework
- Problem Statement
- Related Areas
- History
- Optimization for Intelligent Robots
- Algorithms
- Platforms

Husky A200

Turtlebot2

Turtlebot3

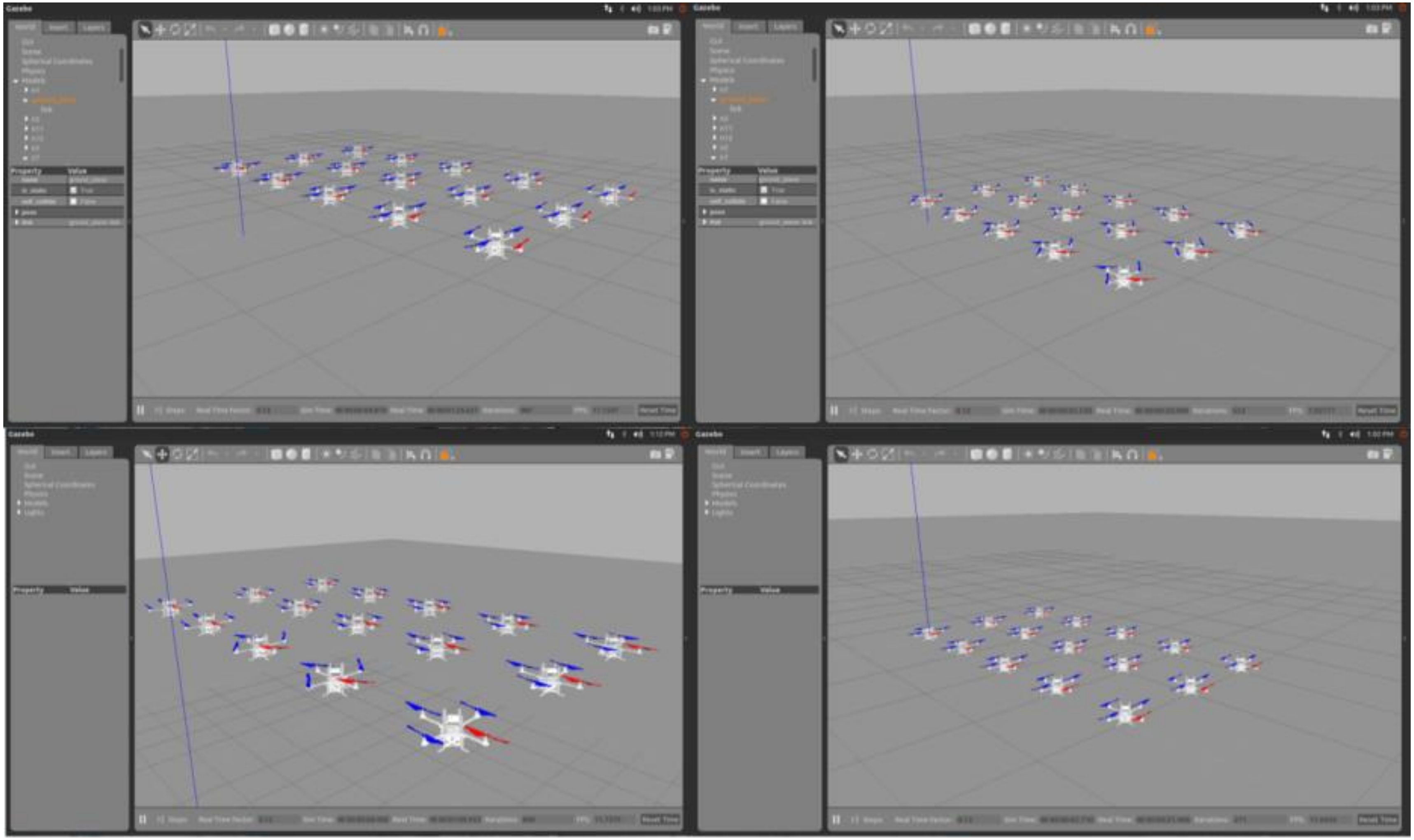
OpenROV


OPENROV
TrustFire
TR-008
CHARGE THREE
18650/25500
26650/26700
32650
NEW

DJI M100

BYD: QIN2018

Simulation Environment: Gazebo



Indoor Environment

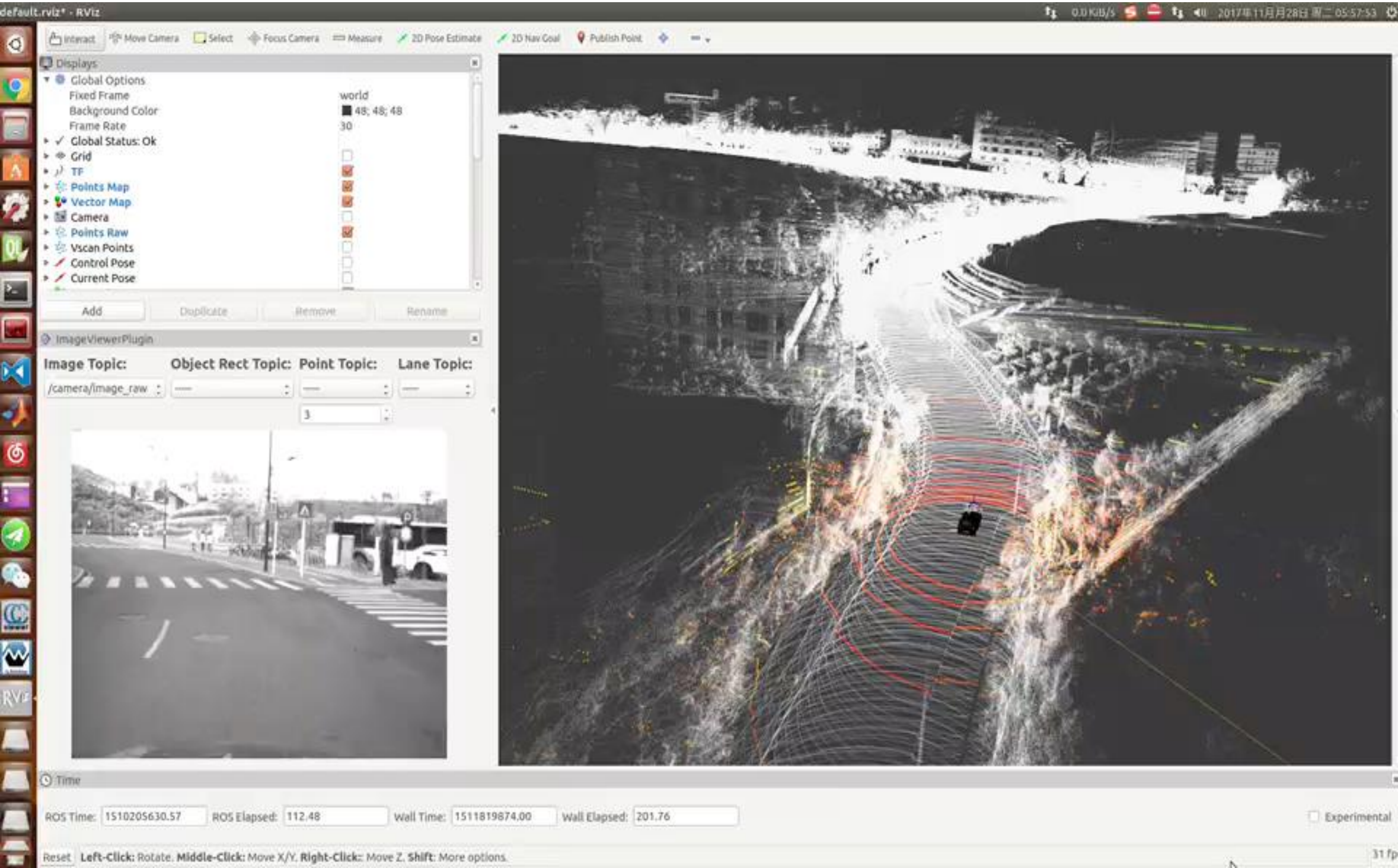
Outdoor Environment

3D Reconstruction

Simulation for Reinforcement Learning

Indoor SLAM Experiment

Outdoor SLAM Experiment



Outdoor Autonomous Driving

Thanks !