Filtering and State Estimation

Graduate Course INTR-6000P

Week 6 - Lecture 11

Changhao Chen

Assistant Professor

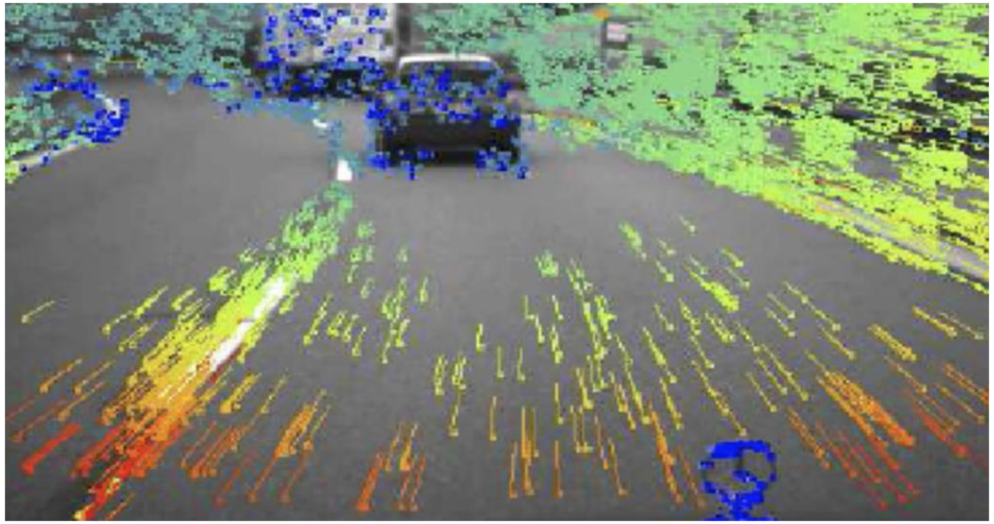
HKUST (GZ)

Recap: Feature-Based Methods vs. Direct Methods

Feature-Based Methods	**Direct Methods**
1. **Detect** distinctive features (e.g., corners).	1. **Use all or most pixel intensities directly.**
2. **Match** features across images.	2. **Assume brightness constancy** for a world point.
3. **Reconstruct** 3D points via triangulation.	3. **Minimize photometric error** to find camera motion.
4. **Motion from 3D-2D** correspondences (PnP).	4. **Geometry and motion are solved jointly.**
"Geometry-first"	**"Photometry-first"**

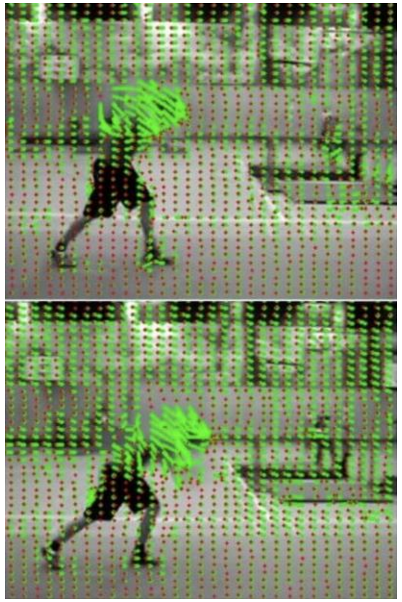
Recap: 2D Optical Flow

Definition: The apparent motion of brightness patterns in the image plane between two consecutive frames.



•**It is a 2D Vector Field:** For each pixel (or region), optical flow is a vector (u, v) representing:

- u: horizontal displacement (pixels/frame)
- v: vertical displacement (pixels/frame)



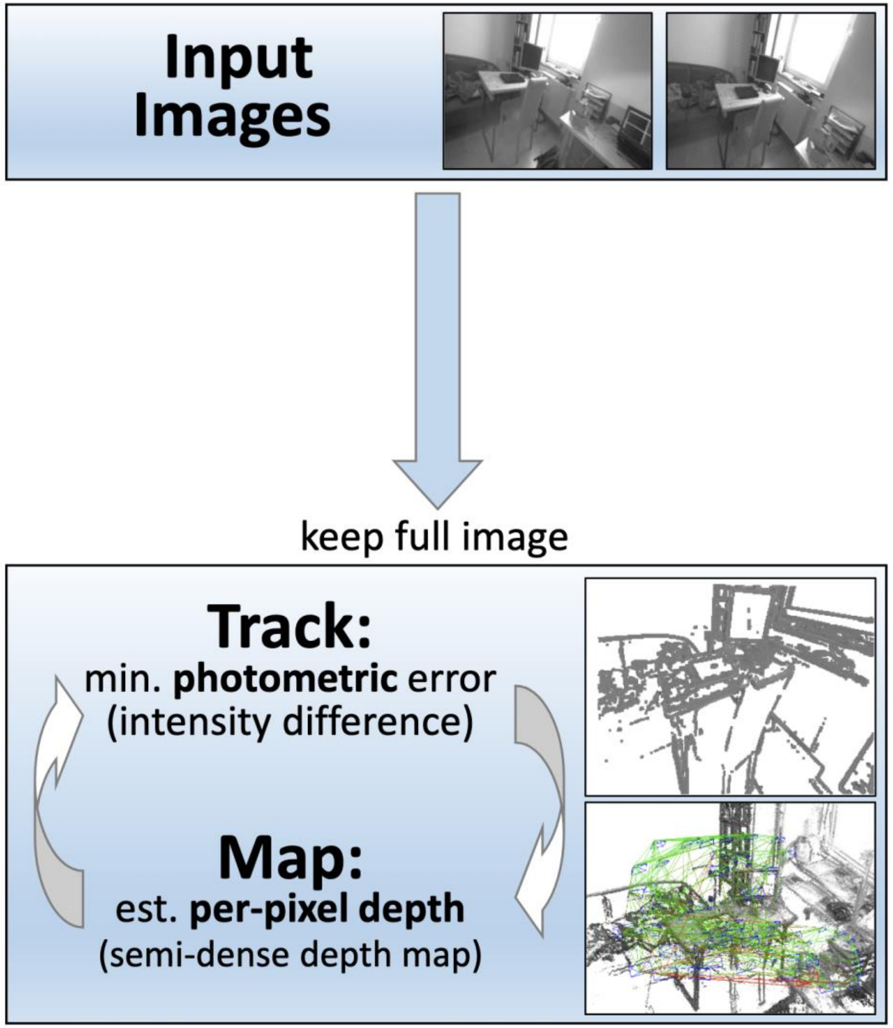
Recap: Direct Visual Odometry

Photometric Consistency

Core Idea: The brightness of a world point remains constant across successive frames.

Assumption: $I(p, t) \approx I(p', t+1)$
- p is a pixel in the first image.
- p' is the *corresponding* pixel in the second image, projected from the same 3D point.

This is a **strong assumption!** It can be violated by:
- Non-Lambertian surfaces (specular reflections)
- Auto-exposure and auto-white balance
- Lighting changes



Recap: Direct Visual Odometry

The Pipeline of a Direct VO System

Input Image -> Pre-processing -> Pixel Selection -> Motion Estimation (Tracking) -> Depth Estimation (Mapping) -> Map Management -> Output Pose

Step 1: Image Pre-processing

- **Why?** The photometric consistency assumption is fragile.
- **Key Calibrations:**
  - **Vignette:** Compensate for darker corners of the image.
  - **Gamma Correction:** Account for non-linear camera response.
  - **Photometric Calibration:** Pre-compute a camera response function.
- **Rolling Shutter Compensation:** Model and correct for distortion from sequential row exposure.

Step 2: Selecting Good Pixels

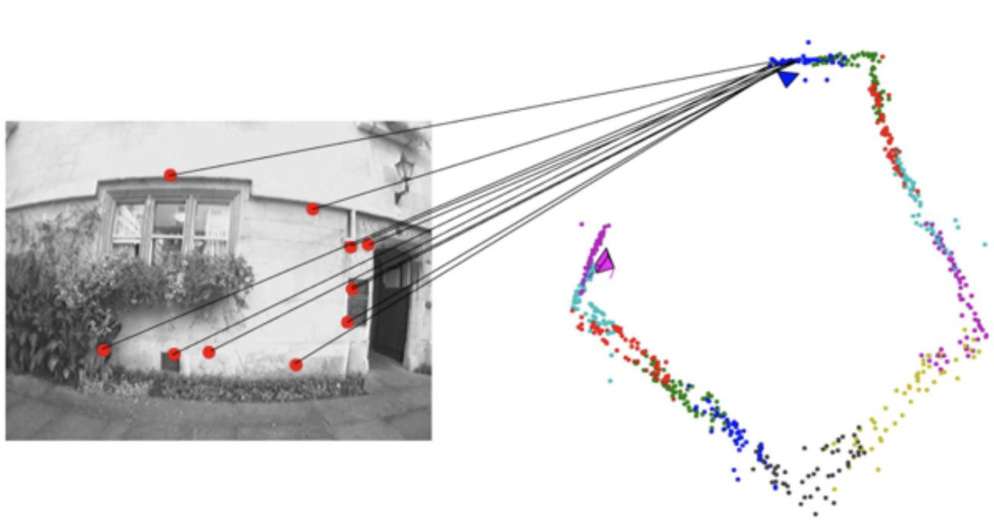
- We want pixels where we can reliably estimate depth.
- **Ideal Pixels:** Have a high image gradient (e.g., along edges).
- **Bad Pixels:** No gradient (completely uniform regions).
- **Strategy:** Sample pixels from across the image, prioritizing those with high gradient magnitude.

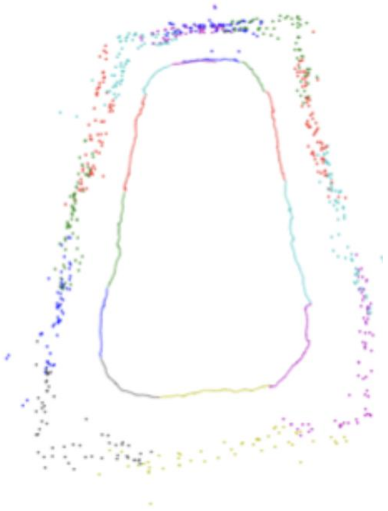
Recap: Loop Closing

Loop Closing - The process of recognizing a previously visited location and correcting the accumulated drift.

Functions:

- **Drift Correction:** Significantly reduces long-term error.
- **Map Consistency:** Produces a globally consistent map.
- **Enables Long-Term Autonomy:** A vehicle can operate for hours/days without getting lost.





Recap: Place Recognition

Definition: **The task of determining where an image was taken by matching it against a database of geo-referenced images.**

It's not just image retrieval! It's about *appearance-invariant* recognition.

Input: A query image from the vehicle's current view.
Output: A binary decision ("Is this a loop?") and/or a match to a previous location in the map.

(a)

(b)

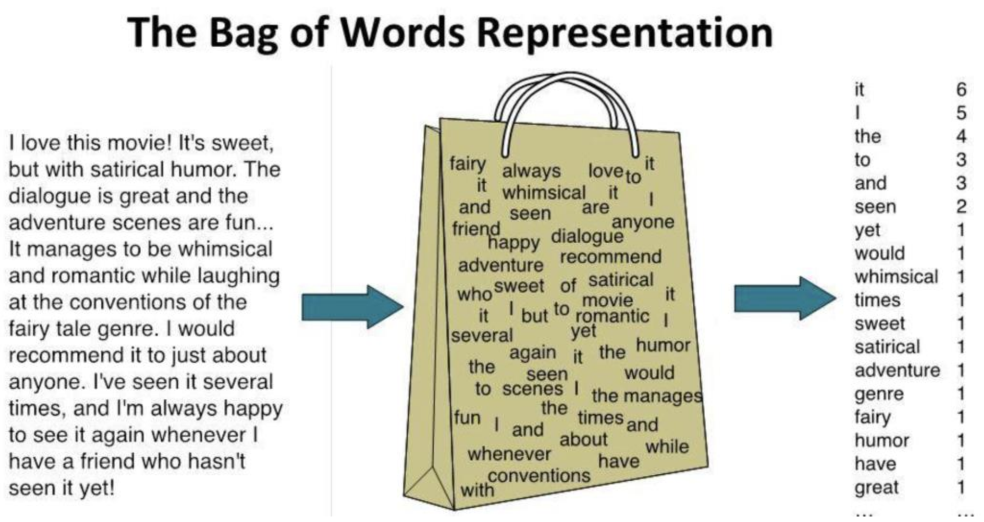
Recap: Bag-of-Words (BoW)

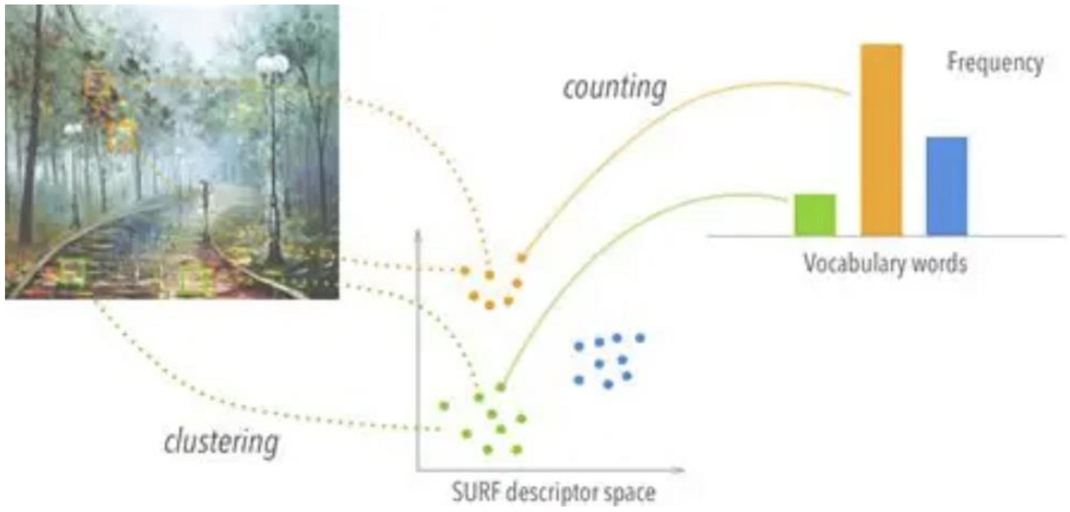
Borrowed from text retrieval. Treat an image as a "bag" of visual words, ignoring their spatial arrangement.

Pipeline:

- **Feature Extraction:** Detect and describe keypoints (e.g., with SIFT).
- **Vocabulary Building:** Cluster all descriptors from the training dataset to create a "visual vocabulary."
- **Quantization:** Assign each new feature to its nearest visual word.
- **Image Representation:** Create a histogram of visual word frequencies for each image.

Matching: Compare histograms using a distance metric (e.g., L1, L2). Fast and scalable!





State Estimation

“State” = the minimal set of variables describing the system at a given time.

For a vehicle:
- Position (x, y, z)
- Velocity (v_x, v_y, v_z)
- Orientation (roll, pitch, yaw)

State evolves over time and is observed indirectly through sensors.

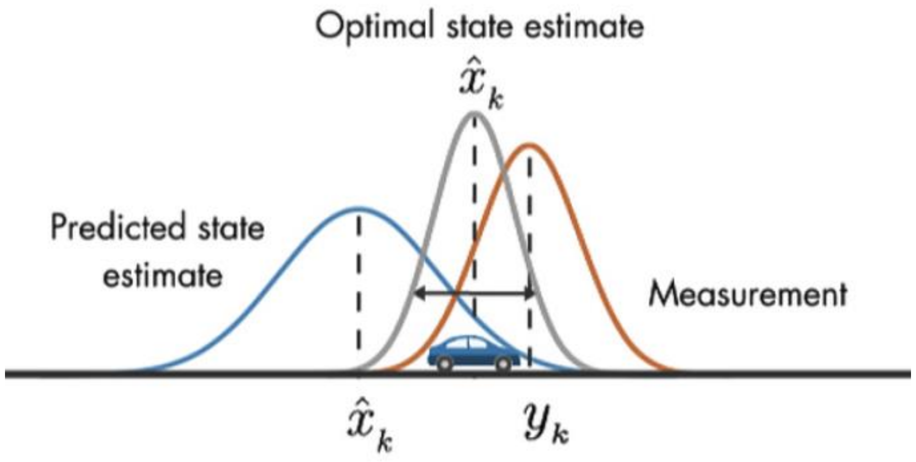
State Estimation

All sensors and models have **uncertainties**.

Motion noise: modeling error.
Measurement noise: GPS accuracy, IMU bias, camera errors.
Represented as random variables.

Target: Maintain an **accurate belief** of the vehicle's state.
Continuously refine this belief as new data arrives.

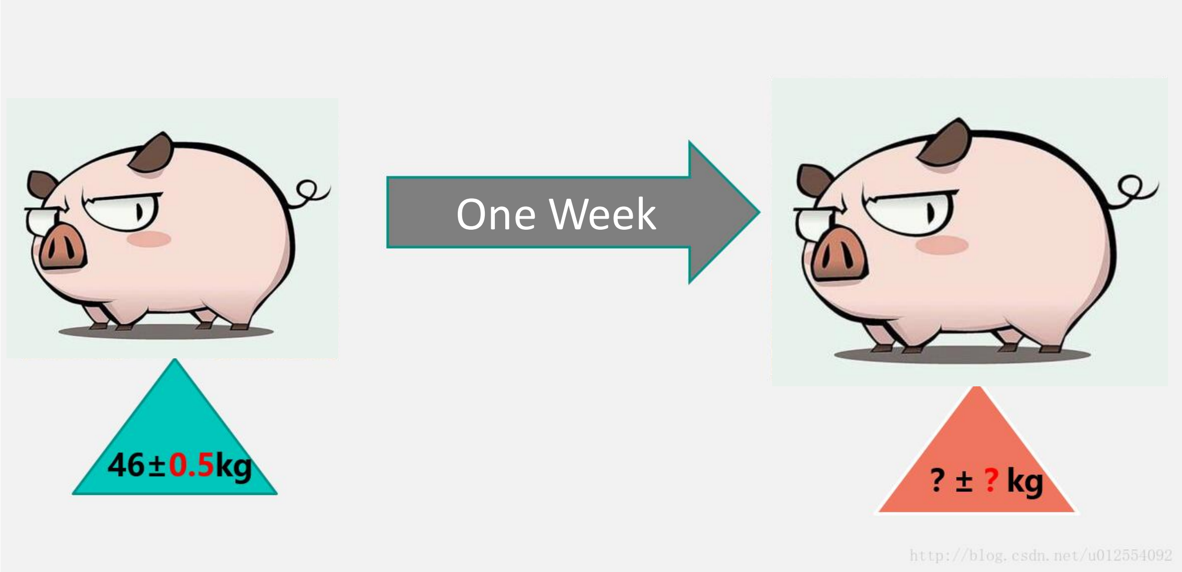
Core question: "*Given noisy sensor data, what is the most probable current state?*"



Motivation

Suppose I have a pig that currently weighs 46 ± 0.5 kg.

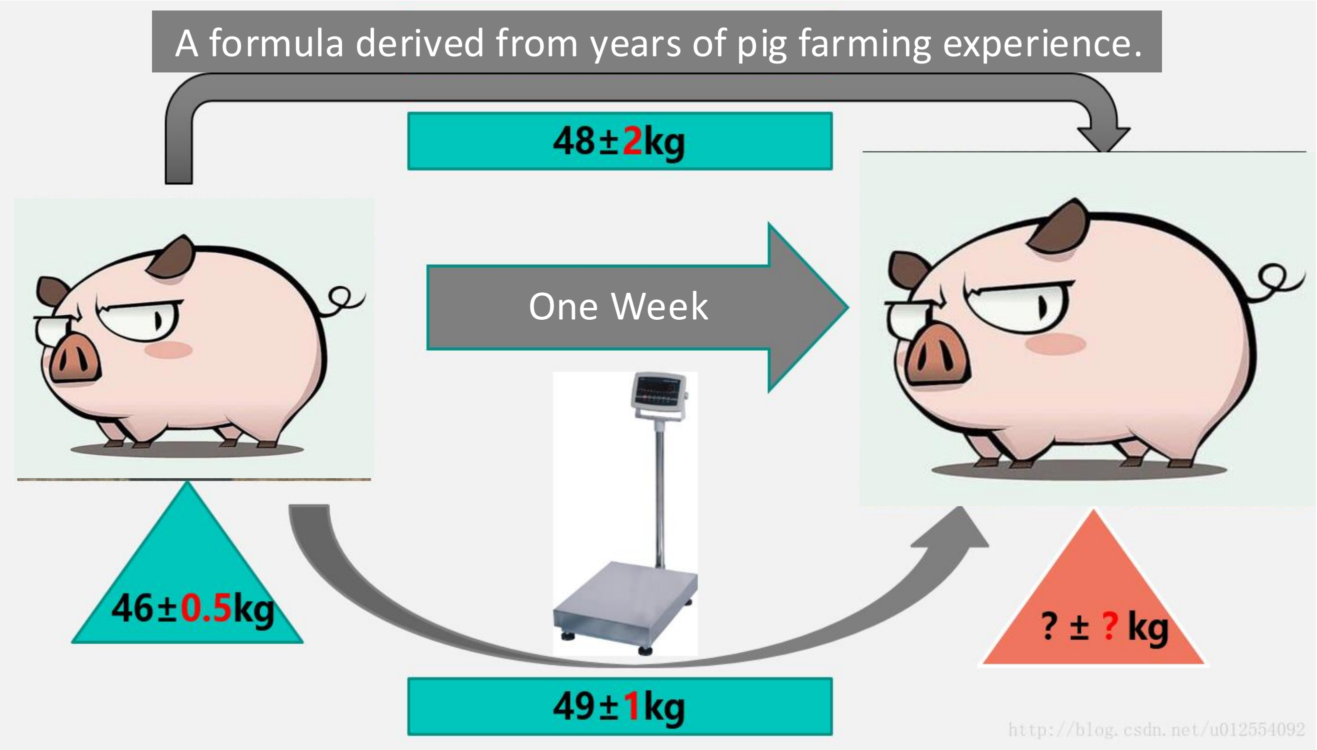
How much heavier is the pig after a week?



Motivation

Method One: Estimation Based on Years of Pig Farming Experience

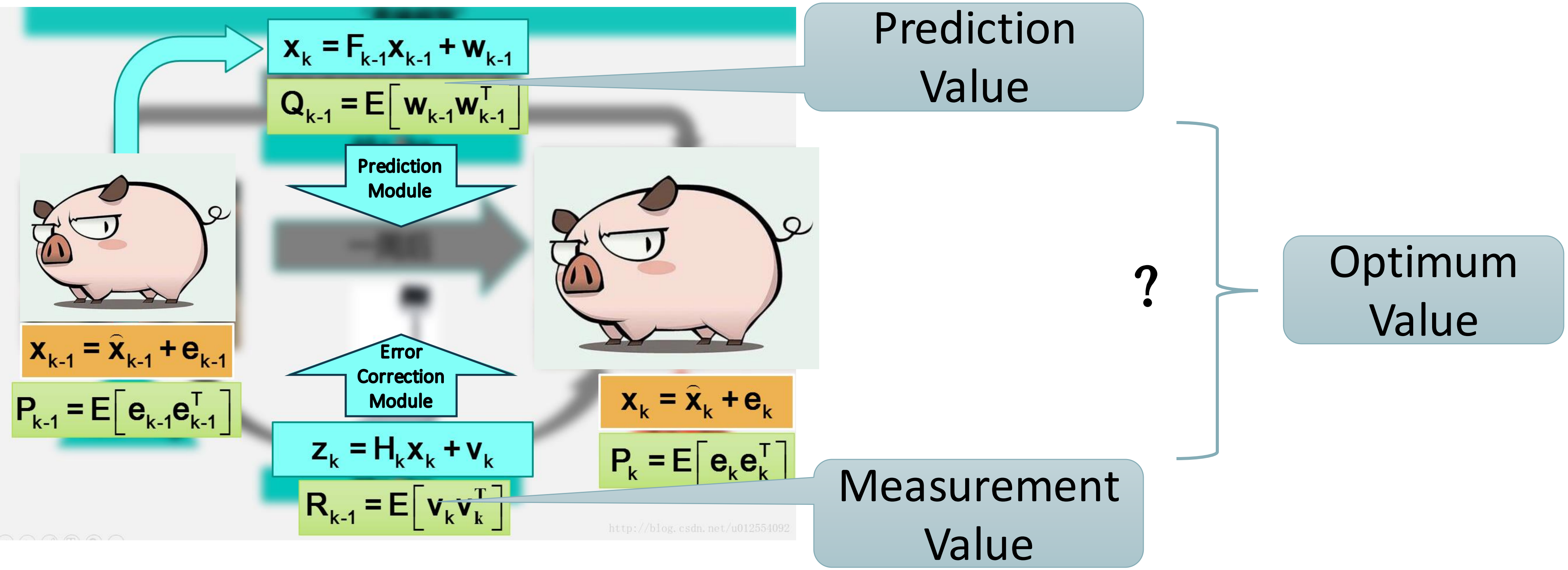
Method Two: Use a scale to weigh



Both methods have errors. How can we obtain more accurate results?

Motivation

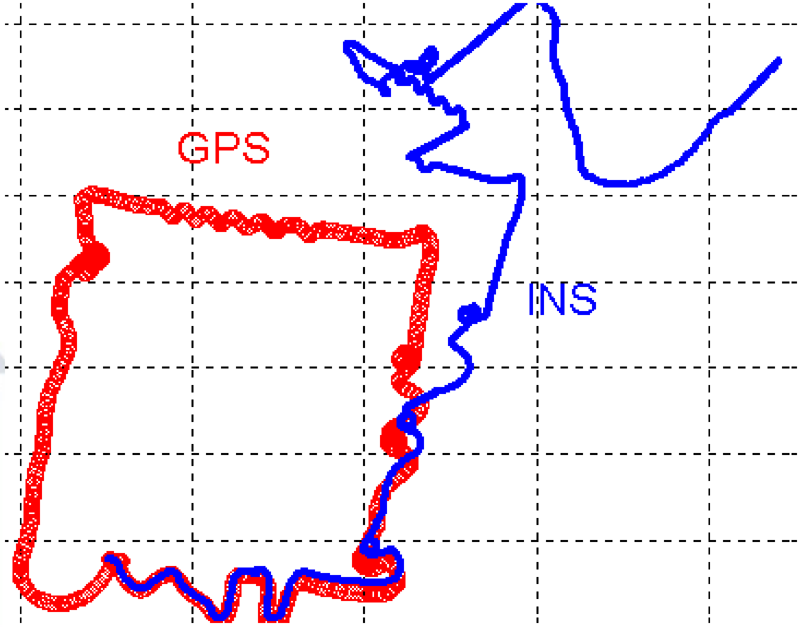
Question: How can I obtain the most accurate weight measurement?



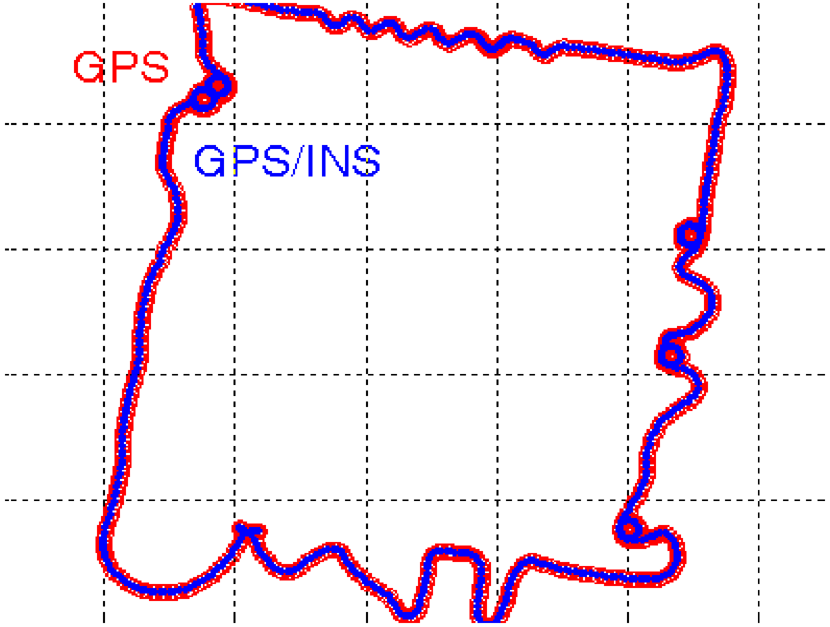
This is precisely the problem that Kalman filtering aims to solve!

Background

- Satellite/Inertial Information Fusion for Integrated Navigation.
- Both inertial navigation and satellite navigation systems produce localization results with errors. How can these be fused?



Localization results obtained solely through inertial navigation or satellite navigation



Fused localization results

Background

Estimating the optimal state from different measurements

In 1794, Gauss proposed the method of least squares to solve the problem of estimating planetary orbital movements.

Without considering the statistical properties of the signal, only the variance of the measurement error is minimized.

In 1942, Wiener proposed the Wiener filter to address the issue of precision tracking in fire control systems.

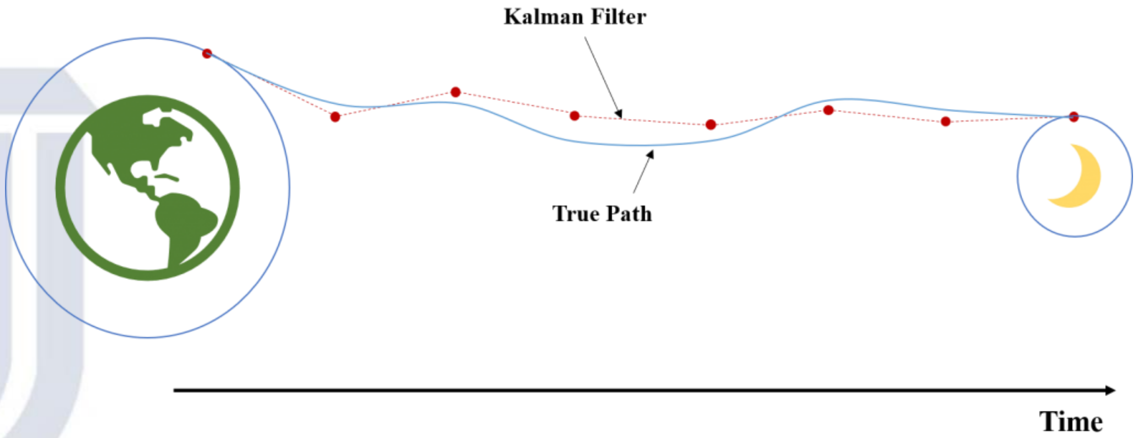
By fully leveraging statistical properties of input and measurement signals, the Wiener filter is a frequency-domain method and is non-recursive, making it inconvenient for real-time applications.

In the 1960s, Kalman proposed the Kalman filter to address the problem of integrated navigation for spacecraft.

Design optimal filters directly in the time domain to obtain recursive minimum mean square error estimates of the system state.

Background

The Kalman filter is an autoregressive filter capable of estimating the state of a dynamic system within composite information characterized by multiple uncertainties.



In 1963, NASA employed a 21-dimensional Kalman filter in the navigation system of the Apollo spacecraft to achieve humanity's first lunar landing.

Rudolf E.Kalman
(1930-2016)

Visited China in the 1980s
Received the National Medal of Science in 2009

2.1 One-Dimensional Case

For a physical quantity x, we obtain two independent measurements x_1, x_2 and σ_1^2, σ_2^2 with corresponding variances

How can two independent estimates of a variable be optimally combined?

$$\boxed{\hat{x} = w_1 x_1 + w_2 x_2} \qquad w_1 + w_2 = 1$$

Calculate the weighted average.

Fundamental Principles

Calculate the variance of the estimate:

$$\sigma^2 = E\left[(\hat{x} - E\hat{x})^2\right] = E\left[(w_1 x_1 + w_2 x_2 - \bar{x})^2\right]$$

$$\overset{w_2=\omega, w_1=1-\omega}{=} (1-\omega)^2 \sigma_1^2 + \omega^2 \sigma_2^2$$

Obtain the optimal estimate:

$$\hat{x} = (1-\omega)x_1 + \omega x_2 == \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} x_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} x_2$$

$$\sigma^2 = \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}$$

Differentiate of the variance:

$$\frac{d}{d\omega}\sigma^2 = -2(1-\omega)\sigma_1^2 + 2\omega\sigma_2^2 = 0$$

$$\Rightarrow \quad \omega = \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}$$

It can also be written as:

$$\boxed{\begin{aligned} \hat{x} &= x_1 - \omega(x_1 - x_2) \\ \sigma^2 &= \sigma_1^2(1-\omega) \end{aligned}}$$

2.2 Mathematical Description of Discrete Dynamic Systems

System Equations: $X_k = \Phi_{\mathrm{k,k-1}} X_{k-1} + \Gamma_{\mathrm{k-1,k}} W_{k-1}$ $\quad W \in N(0, Q)$

State vector

State Transition Matrix

System Noise Matrix

System Noise

Observation equation: $Z_k = H_k X_k + V_k$ $\quad V \in N(0, R)$

Observation vector

Observation Matrix

Observation Noise

Fundamental Principles

2.3 Kalman Filter Equation

Prerequisites for optimal estimation: 1) Linear systems; 2) System noise and observation noise follow a normal distribution.

Prediction Process:

State-by-State Prediction:

$$\hat{X}_{k|k-1} = \Phi_{\mathrm{k,k-1}} \hat{X}_{k-1}$$

One-step Variance Prediction:

$$P_{k|k-1} = \Phi_{k,k-1} P_{k-1} \Phi_{k,k-1}^{T} + \Gamma_{k-1} Q_{k-1} \Gamma_{k-1}^{T}$$

Mean Squared Error Matrix of the Previous Moment Optimal Estimate $\hat{X}_{k-1}$

Update Process:

Filter gain coefficient: $K_k = P_{k|k-1} H_k^T \left(H_k P_{k|k-1} H_k^T + R_k \right)^{-1}$

Optimum estimate: $\hat{X}_k = \hat{X}_{k|k-1} + K_k \left(\underline{Z_k - H_k \hat{X}_{k|k-1}} \right)$

One-step prediction

New Information (One-Step Forecast Error of Observations)

Optimal Estimator Variance Matrix:

$$P_k = (\mathrm{I} - \mathrm{K}_k \mathrm{H}_k) \mathrm{P}_{k|k-1} (\mathrm{I} - \mathrm{K}_k \mathrm{H}_k)^T + K_k R_k K_k^T$$

From Linear to Nonlinear Systems

Real vehicles → nonlinear motion and observation models.
Example: turning, orientation changes, camera projection.

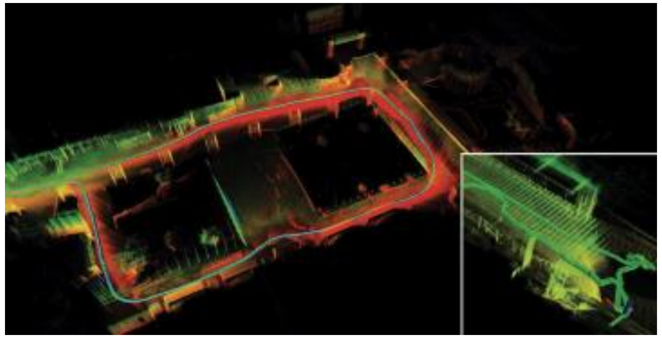
KF needs adaptation → **Extended Kalman Filter (EKF)**.

Linearizes nonlinear models around the current estimate.
Still uses prediction–update steps.
Works well for “mildly” nonlinear systems.

Application

1) Robotics and Autonomous Driving

- Visual/Inertial Information Fusion: MSCKF （2012）
- LiDAR/Inertial Information Fusion: FAST-LIO2 （2021）
- Drone Control: Drone Racer （2023）



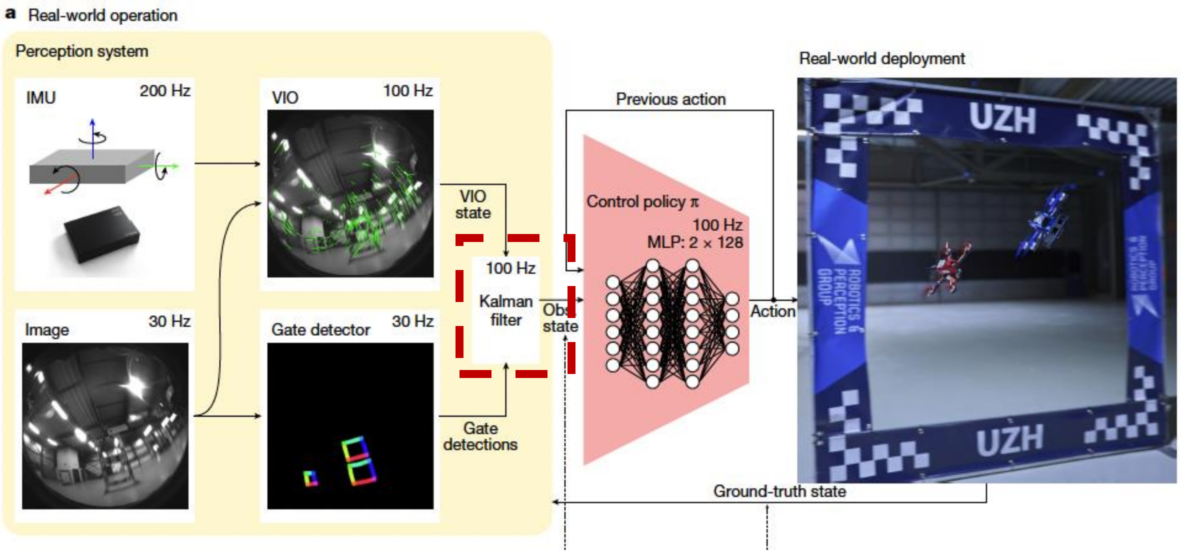
2) Computer Vision

3) Econometrics

4) Process Control

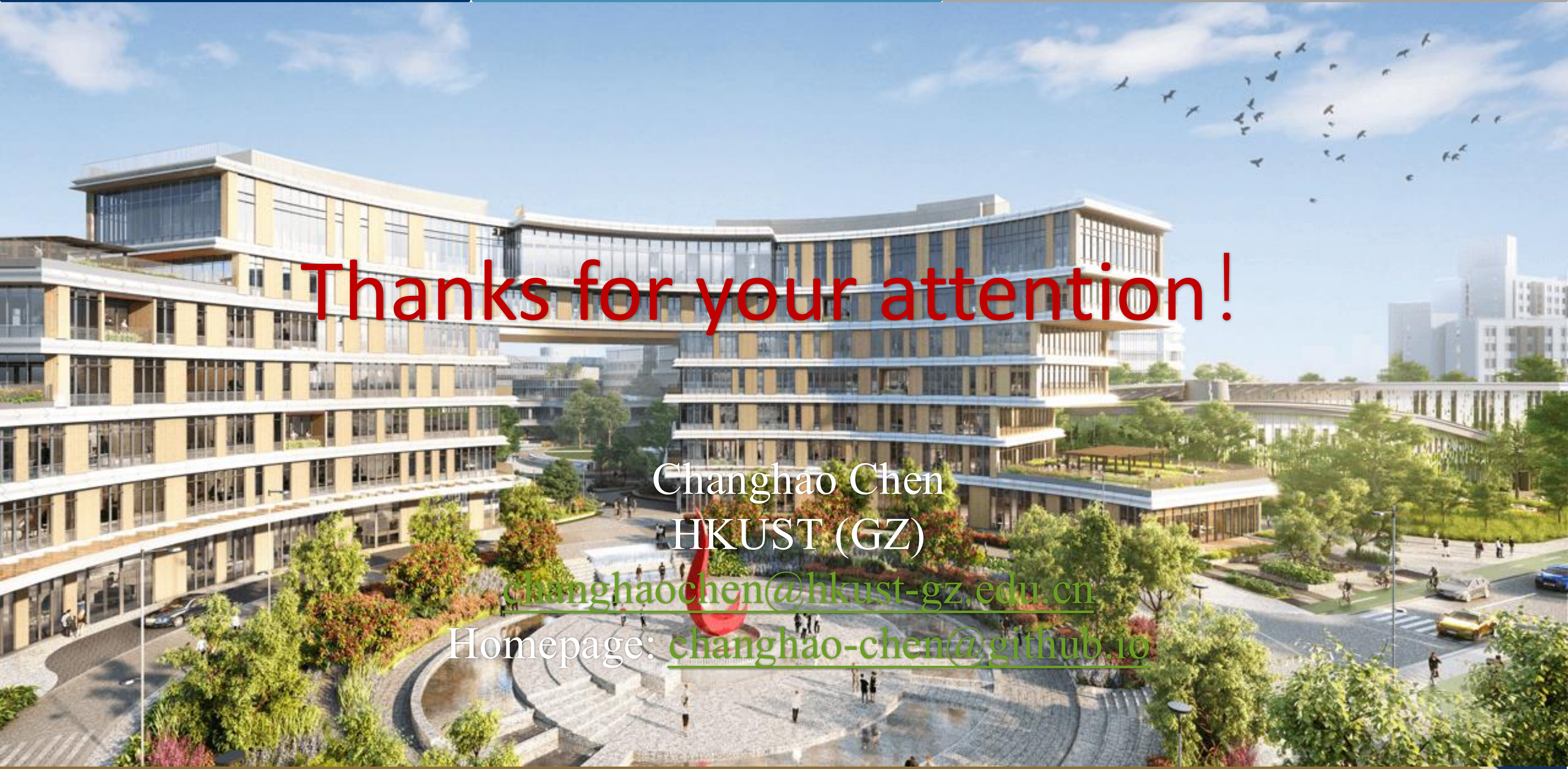
5) Weather Forecast

6) Health Screening



Summary

- The Kalman filter is a general-purpose method for fusing data from multiple sources, estimating system states based on multiple measurements and uncertainties within the system.
- It performs recursive linear minimum variance estimation through prediction and update processes.
- Under the conditions of a linear system and normally distributed noise, it serves as an optimal estimation technique.



Thanks for your attention！

Changhao Chen
HKUST (GZ)
changhaochen@hkust-gz.edu.cn
Homepage: changhao-chen@github.io