AKAASH NIDHISS 2K19/IT/008
ANASUYA MITHRA 2K19/IT/018
ANSHUMAN ROY 2K19/CO/068
AHMAD HABIB KHAN 2K19/SE/003

Vehicle Tracking using Kalman Filter

Computer Vision Project

Index

- Aim Of The Project
- Introduction
  - Kalman Filter
  - Equations
- Code Presentation
- Output Screenshots
- Output Video
- Learning Outcomes
- Conclusion
- Future Aspects

AIM OF PROJECT

This project aims to track a vehicle using Kalman filter, and display the measured, estimated and predicted positions of the vehicle at all instances

Kalman Filter

Kalman filtering is an algorithm that allows us to estimate the states of a system given the observations or measurements. It is a useful tool for a variety of different applications including object tracking and autonomous navigation systems, economics prediction, etc.

The basic idea of the Kalman filter is by using the prior knowledge of the state, the filter makes a forward projection state or predicts the next state.

Equations

The Kalman Filter is intended to estimate the state of a system at time k using the linear stochastic difference equation assuming that the state of a system at a time k evolved from the prior state at time k-1 as written in the following form:

$$\mathbf{x}_k = A\mathbf{x}_{k-1} + B\mathbf{u}_{k-1} + \mathbf{w}_{k-1}$$

It is always paired with the measurement model zk that describes a relation between the state and measurement at the current step k. It is written as:

$$\mathbf{z}_k = H\mathbf{x}_k + \mathbf{v}_k$$

where:

- A, a matrix nxn, is the state transition matrix relating the previous time step k-1 to the current state k
- B, a matrix nxl, is a control input matrix applied to the optional control input uk-1
- H, a matrix m x n, is a transformation matrix that transforms the state into the measurement domain
- wk and vk represent the process noise vector with the covariance Q and the measurement noise vector with the covariance R, respectively. They are assumed statistically independence Gaussian noise with the normal probability distribution.

Code Presentation

as shown in the report

OUTPUT SCREENSHOTS

OUTPUT SCREENSHOTS

OUTPUT SCREENSHOTS

OUTPUT SCREENSHOTS

OUTPUT SCREENSHOTS

OUTPUT SCREENSHOTS

Output Video

A video featuring a sample output of our project code can be found here in the following google drive link:
https://drive.google.com/file/d/1W9r0hor1ZZUcZeaO5FhJ9IyRSe92XLgf/view?usp=sharing

Learning Outcome

- Reviewed various research papers to gain insight on the topic at hand.
- Studied and extended the Kalman Filter with new features.
- Learnt how to evaluate the results obtained from applying the Kalman Filter.
- Tested the project out on a real life model, to obtain accurate results.

Future Aspects

In order to create an algorithm which reacted quickly to changes (Process Noise) in the object being detected, the Kalman Gain coefficient was accordingly increased in order to properly and quickly react to these changes. However, during the work on this project, it was realised that the car detector algo (contained in Detect.py) identified non-car objects and mistyped them as cars. Thus, the performance of the algorithm suffers when faced with high levels of measurement noise.

In order to enhance the working of the project code, necessary modifications should be performed to enhance the car detection algorithm and thus decrease overall measurement noise.

Conclusion

In conclusion, Kalman filtering is an algorithm that allows us to estimate the states of a system given the observations or measurements. It is a useful tool for a variety of different applications including object tracking and autonomous navigation systems, economics prediction, etc.

We implement the same in python, with the project having three files, namely: Kalman.py, Detect.py, and main.py.

Kalman.py contains one class called KalmanFilter consisting of three functions, __init__(), predict(), and update().

Detect.py file plays a role as an object detector and main.py is the main file of this project that we'll execute to track an object. At the beginning of this file, we import the function detect() from the file Detector.py, and class KalmanFilter from the file Kalman.py.

The End.