Free-Style Object Tracking using OpenCV APIs

Contents

TEAM MEMBERS

- ❑ Team leader :
  - ❖ Akash Ghimire (12194814)

- ❑ Other members:
  - ❖ Thapa pradip (12194940)
  - ❖ Bijay pariyar (12194945)
  - ❖ Keshav Adhikari (12194874)

Team meeting and brainstorming

Date	Purpose	venue
Nov 19	Idea selection and dividing personal task	Kay's cafe
Nov 21	Coding and work progress report	Inha university library
Nov 26	Record the video journal	Zoom meeting

Team Member Contribution:

Team UNO	Team Members
Contribution- Team Leader and project handler.	Akash Ghimire (12194814) – Team Leader
Contribution- Research and data collection.	Bijay Pariyar (12194945)- Team Member
Contribution- Devil’s Advocate and quality controller.	Keshav Adhikari (12194874)- Team Member
Contribution- Arbitrator and facilitator.	Pradip Thapa (12194940)- Facilitator

Fig: Group meeting in Compose Cafe for idea creation and discussion

Fig: General group meeting

Object tracking

- A process of estimating or predicting the positions and other relevant information of moving objects in a video.

The tracking procedure can be viewed as a combination of two models

Motion model : tracks the speed and direction of the object's movement, allowing it to forecast the object's new position based on the received data

Appearance model: responsible for verifying whether the object we've chosen is within the frame or not.

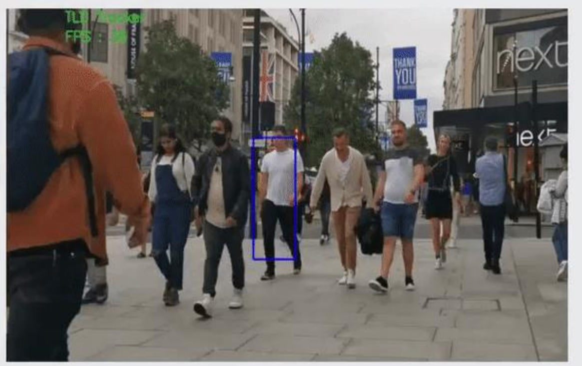
Comparison of different types of object tracking filters

Filters	description	pros	Cons
CSRT	works by training a correlation filter with compressed features (HoG and Color Names).	shows comparatively better accuracy, resistance to overlapping by other objects.	sufficiently low speed, an unstable operation when the object is lost.
GOTURN	It is deep learning based ,offline tracking algorithm.	comparatively good resistance to noise and obstructions.	Loses an object and shifts to another if the speed of the first one is too high.

Sample using csrt tracker

Sample using GOTURN tracker

filters	description	pros	cons
TLD	real-time algorithm for tracking of unknown objects in video streams. TLD simultaneously Tracks the object, Learns its appearance and Detects it whenever it appears in the video.	shows relatively good results in terms of resistance to object scaling and overlapping by other objects.	unpredictable behaviour instability, constant loss of an object, tracking similar objects instead of the selected one.
Median-Flow tracker	This algorithm is based on the Lucas-Kanade method	sufficiently high speed and accuracy	inability to continue tracking after the loss of the object.



Sample using TLD tracker

Sample using median-flow tracker

Sample reference: https://broutonlab.com/blog/opencv-object-tracking

Why kalman filtering

- Efficient “least-squares” implementation
- Past, present and future estimation
- Estimation of missing states
- Measure of estimation quality (variance)
- Convenient form for online real time processing.

Kalman Filter

Why kalman filtering is better than others

- This is a statistical technique that adequately describes the random structure of experimental measurements.

- provides information about the quality of the estimation by providing, in addition to the best estimate, the variance of the estimation error.

- Its recursive structure allows its real-time execution without storing observations or past estimates.

Application Scopes of Kalman Filter:

- Tracking Objects (e.g., missiles, faces, heads, hands).
- Fitting Bezier patches to (noisy, moving,..) point data.
- Economics.
- Navigation.
- Many computer vision applications:
    - Stabilizing depth measurements
    - Feature tracking
    - Cluster tracking
    - Fusing data from radar, laser scanner & stereo-cameras for depth & velocity measurements etc,.

Application Example of Kalman Filter:

- Kalman Filters are widely used in the area of Object Tracking in Artificial Intelligence systems. For example:

- Take a case where a person who was visible for a number of frames gets occluded by a car later.
- Now when the person is occluded, since there aren't pixels occupied by the person, it's impossible for detector to detect the person.
- However, using Kalman Filter, knowing that the person was moving from a certain position in a certain direction, the Kalman Filter will be able to predict the trajectory, and so it's possible to track the person even if the detector can't.

Problems with kalman filter

- It is very difficult to compute the covariance matrix of noise of various sensors and systems

- The filter needs to process several samples in order to get enough iterations to produce meaningful results

Kalman Filter:

The Kalman filter represents all distributions by Gaussians and iterates over two different things: **measurement updates** and **motion updates**.

a) Measurement updates: The Kalman filter estimates the current state of the system given the measurement at that time step.

$$\mu' = \frac{r^2\mu + \sigma^2\nu}{r^2 + \sigma^2}$$

Update the prior mean with the weighted sum of the old means, where the weights are the variances of the other mean. The mean is normalized by the sum of the weighting factors.

$$\sigma^{2'} = \frac{1}{\frac{1}{r^2} + \frac{1}{\sigma^2}}$$

update of the variance uses the previous variance

b) Prediction/Motion update: The Kalman Filter predicts the next state of the system given the previous measurements.

$$\mu' = \mu + u$$

The mean is updated by taking the previous mean and adding the motion of the movement, indicated by variable u.

$$\sigma^{2'} = \sigma^2 + r^2$$

The variance is updated by taking the old sigma and adding onto it the variance of the motion Gaussian

State Matrix

- The state matrix records the object being tracked. It could be another car on the road or a plane in the air.

i) $$X_t^p = AX_{t-1} + Bu_t + w_t$$

State Matrix Prediction

ii) $$X = \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix} \qquad X = \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{bmatrix}$$

The state matrix can represent multi-dimensions multiple ways, but the ordering will affect how the A matrix is constructed.

iii) $$x = x_0 + \dot{x}t + \frac{1}{2}\ddot{x}t^2 \qquad d = d_0 + vt + \frac{1}{2}at^2$$

Updating the position involves determining the displacement of the object given the acceleration and velocity.

iv) $$AX = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} = \begin{bmatrix} x + \Delta T \dot{x} \\ \dot{x} \end{bmatrix}$$

First part of updating matrix X

v) $$Bu = \begin{bmatrix} \frac{1}{2}\Delta T^2 \\ \Delta T \end{bmatrix} \begin{bmatrix} a \end{bmatrix} = \begin{bmatrix} a\frac{1}{2}\Delta T^2 \\ a\Delta T \end{bmatrix}$$

Second part of updating matrix X

iv) $$\sigma_x \sigma_y = \sum_{i=1}^{n} \frac{(\bar{x} - x_i)(\bar{y} - y_i)}{N}$$

Covariance, where x_bar is the mean of x

v) $$\begin{bmatrix} \sigma_x^2 & \sigma_x \sigma_y \\ \sigma_y \sigma_x & \sigma_y^2 \end{bmatrix}$$

Example 2x2 Covariance Matrix

vi) $$\begin{bmatrix} \sigma_x^2 & \sigma_x \sigma_y & \sigma_x \sigma_z \\ \sigma_y \sigma_x & \sigma_y^2 & \sigma_y \sigma_z \\ \sigma_z \sigma_x & \sigma_z \sigma_y & \sigma_z^2 \end{bmatrix}$$

Example 3x3 Covariance Matrix

Working of Kalman Filter

Flowchart of a Kalman Filter Matrix process

Kalman Gain

- The Kalman gain is used to determine how much of the new measurements to use to update the new estimate.

$$K = \frac{E_{est}}{E_{est} + E_{mea}}$$

Kalman Gain is calculated by comparing the error in the estimate relative to the error of the estimate and measurement combined

$$x_t = x_{t-1} + K[p - x_{t-1}]$$

The Kalman gain is used to adjust the update to the state.

$$Est_t = Est_{t-1} + K[M - Est_{t-1}]$$

Updating estimates weighted by Kalman gain

$$E_t = [1 - K]E_{t-1}$$

Updating estimation errors