Read frame

Detect objects

Process object

Check if fit to previous object (

mTracks.neigh = NearestNeighbors(n\_neighbors = 1)

check = NearestNeighbors(n\_neighbors = 1)

check.fit(zList)

\_, idx = check.kneighbors([z])

d, relativeIndex = self.neigh.kneighbors(self.trackers[key].position()[np.newaxis, :])

mtracks.neigh.fit(zList))

if new, make track

new architecture:

run on frame\_dt/2

for each track: predict.

in frame\_cycle:

read frame.

Detect objects.

If associated, update.

If new, open track.

If more than 3 frames wasn’t updated, close.

Process frame annotations

Kalman model: below.

\\ to much work. Let’s leave it currently in the notes to the notebook example. And

\\ just set dynamically the box when assignment is valid.

|  |  |  |
| --- | --- | --- |
| neigh = NearestNeighbors(n\_neighbors = 1) | mTracks constructor | N\_neighbors = 1  find the single nearest neighbor for each query point |
| neigh.fit(zList)) | Main loop | Train with existing data on each loop if points length > 1 |
| d, relativeIndex = neigh.kneighbors(trackers[key].position()[np.newaxis, :]) | mTracks.refresh | If fit is the train, kneigbours is the test. Check this current track.  Iterates on each one of the trackes, |
| check = NearestNeighbors(n\_neighbors = 1) | mTracks.trackerMaker(zList) | Prepare object for assignment with a single beighbour per query. |
| check.fit(zList) | mTrack.fit(zList) | Train the model based on the imput data. |
| \_, idx = check.kneighbors([z]) |  |  |

Diff between neigh and check:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Neigh | Check 1 | Check 2 |
| Member | a member of the mTracks class. | locally within the mTracks .trackerMaker method |  |
| Use | for associating object measurements with existing trackers. |  |  |
| purpose | nearest neighbors of trackers based on object measurements. |  |  |
| Training dataset |  | Zlist array of all detected objects. | Only the associated detected objects in zlist. |
| Test\samples data |  | Every one of the exisiting trackers. | Each one of the non-associated detections |
| Purpose |  | identify singles to open new tracks. | Identify far associations and open also for them a new track. |
| Implementation with c4d. |  |  |  |

Old:

July 7:

Run the yolo program with c4dynamics

Show traj of all the cars that detected during th run.

Make a jupyter notbook in c4dynamics of the yolo tracker example

create a beautiful carousel shwing how I utilized c4dynamics to run the same program easily.

יתרונות:

לא היה צריך לכתוב מילה אחת של קלמן./רק המשואות.

ניהול הנקודות במרחב ושמירת דגימות בזמן מוכנים לשמוש.

Theoretical background

Where:

A is the system matrix

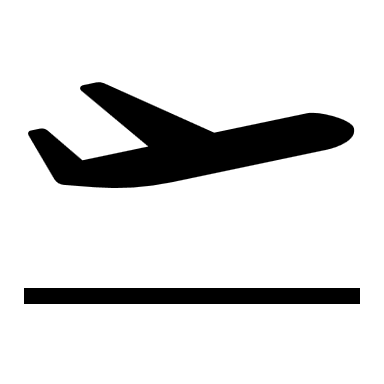
b is the input parameters vector

vp is the process noise, zero mean, normal distributed, with covariance Q: vp=0+randn\*Q

y is the observation of the measurement

c is the output params vector

vm is the measure noise zero mean normal distributed with noise covariance R: vm=0+randn\*R



Sample system Kalman filter equaions

<https://www.mathworks.com/help/vision/ug/using-kalman-filter-for-object-tracking.html>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Steady state | Linear | EKF | Detect track params |
| State |  |  |  |  |
| Initial conditions |  |  |  |  |
| System |  |  |  | For :The process noise cannot depend on the video width!the measurement noise has larger values for a less accurate detector.\* it’s actually not clear why to use this form of Kalman where the optimal gains here may be found in the form of the steady state case and every new measure contributes to the correction of the state with:Where K is vcalculated once at the beginning by solving ARE problem \\ with scipy? \\:המערכת לא ניתנת לחיזוי. להערכה.אולי אפשר להפריד אותה לשתי מערכת:גם כנראה שלו. זה נגזרת אחת של השניה.אבל יש לי טעות בחישוב מטריצת השערוך. |
| X predict |  |  |  | אז בכלל שוה לנסבות לעשות ככה עם חישוב אחד אפריורי ולמצוא Q כזה גדול שמבוסס על המסלולים של המטוסים כדי שכל הטיסה בקו ישר שלהם תהיה תחומה בפנים אבל לתת למדידות R מאוד קטן |
| P predict |  |  |  |  |
| X correct |  | \\?\\ |  |  |
| P correct |  |  |  |  |
|  |  | State vector |  |  |
|  |  | Measures \ output vector |  |  |
| \ |  | Linear system matrix | nonlinear equations |  |
|  |  | Linear input vector |  |  |
|  |  |  | Nonlinear measure equationsto is like to |  |
|  |  | Output matrix | Linearized measures matrix |  |
|  |  | Input signal |  |  |
|  |  | Process noise variance |  |  |
|  |  | Measures noise variance |  |  |
|  |  | System covariance matrix |  |  |
|  |  | State transition matrix |  |  |
|  |  | Process noise covariance, |  |  |
|  |  | Kalman gain |  |  |
|  |  | Measurement noise covariance, |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Appendix: all Kalmans’ convention

<https://www.mathworks.com/help/control/ref/ss.kalman.html>

covariances

1 Linear equations, steady state covariance

In the steady state case we can rely on the constinuos model (wheres that from? Do the gains match the discrete system if they were calc with the cont approach? ) and calculate the optimal gains in advance:

“the steady state Kalman filter: discrete-time case.”

Predict + correct

2 Linear transitional.

3 ekf.

The velicty doesn’t suffice to reach at itsfinal value.

The covariances increase rather than decrease.

A screenshot of a graph

Description automatically generated

Trk.X = [.52, .6, .37, -.0007]

Trk.Xpixels = [674, 435, 479, 0]

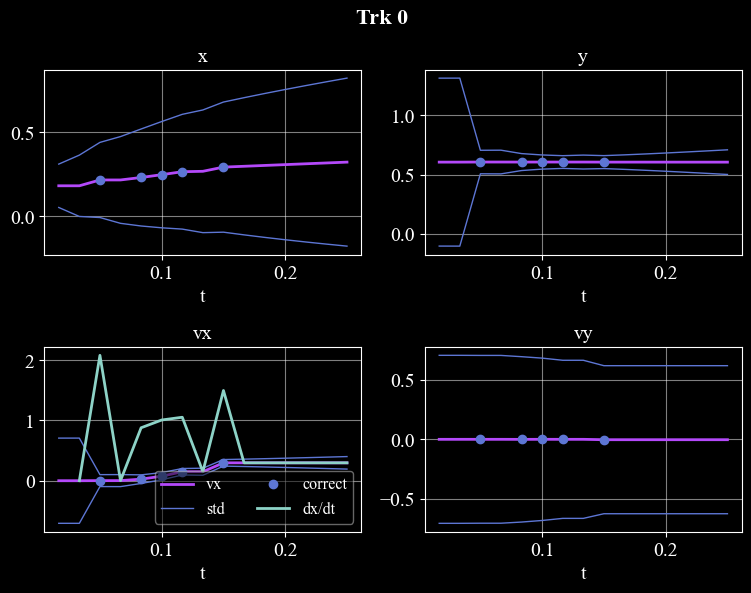
Max velocity 1/10 screen.

Max velocity = screen diagonal (sqrt(2)) per second

Screen diagonal = sqrt(width^2 + height^2)

Arrow size = min(velicty\_pixels, screen\_diagonal) / screen\_diagonal \* .1

Opened too much tracks.



A graph of a graph of a graph

Description automatically generated with medium confidence

A graph of different types of graphs

Description automatically generated with medium confidence

A graph of a graph of a graph

Description automatically generated with medium confidence

Almost the same thing for

Same also for

And also for P0=0.001

A graph of different types of graphs

Description automatically generated with medium confidence

…

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| P0 | Q | R | Ntrks | Estimations | Decreasing covariances |
|  |  |  | 1 | X close to the raw inputs but with bias (why? Maybe initial condition)  Y zero.  Vx raw is 0 but from  velocity doesn’t suffice to reach at itsfinal value. | In x increase  Y almost const.  Vx too tight. Const.  Vy increase. |
|  |  |  | 4 |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| [0.5 0.5 0.1 0.1] | [0.1 0.1 0.01 0.01] | [0.1 0.1] |  |  |  |