

Stack Overflow is a community of 4.7 million programmers, just like you, helping each other.

Join them; it only takes a minute:

Sign up

Join the Stack Overflow community to:

Ask programming questions

Answer and help your peers

Get recognized for your expertise

## Kalman filter in computer vision: the choice of Q and R noise covariances

I read some works about Kalman filter for CV object tracking but I can't find some reference about the choice of: 1)the process noise covariance Q; 2)Measurement noise covariance R. So far I have realized that the model is equation of motion (someone uses acceleration as state variable, others use position and speed only) but nobody is clear about Q and R choice including this example by mathworks:

<http://www.mathworks.it/it/help/vision/examples/using-kalman-filter-for-object-tracking.html> Recently I found this page:

<http://blog.cordiner.net/2011/05/03/object-tracking-using-a-kalman-filter-matlab/> but the Q and R assignment is not clear. Does anyone know help me, please ?

matlab computer-vision simulink matlab-cvst kalman-filter

edited Jan 21 '14 at 3:18



Dima

28.1k

10

50

94

asked Jan 20 '14 at 22:08



Orlok

114

2

14

Do you have any real data (measurements)? You could use those to get approximations of R and Q. R depends on your sensor quality and there are no fixed values for it but you could estimate it from real measurements. R expresses how accurate your sensors are. Q is a measure of how accurate your model is - some dynamics are too complicated to be modelled and are assumed as process noise. By comparing your model predictions with real measurements you could estimate Q. Because of this choices of Q and R vary in literature. – [remus](#) Jan 21 '14 at 10:40

Thanks for reply. Can you read comment below ? Thanks. – [Orlok](#) Jan 21 '14 at 21:51

## 1 Answer

R is the covariance matrix of the process noise, assumed to be Gaussian. In the context of tracking objects in video it means your detection error. Let's say you are using a face detector to detect faces, and then you want to track them using the Kalman filter. You run the detector, you get a bounding box for each face, and then you use the Kalman filter to track the centroid of each box. The R matrix must describe how uncertain you are about the location of the centroid. So in this case for the x,y coordinates the corresponding diagonal values of R should be a few pixels. If your state includes velocity, then you need to guess the uncertainty of the velocity measurement, and take the units into account. If your position is measured in pixels and your velocity in pixels per frame, then the diagonal entries of R must reflect that.

Q is the covariance of the process noise. Simply put, Q specify how much the actual motion of the object deviates from your assumed motion model. If you are tracking cars on a road, then the constant velocity model should be reasonably good, and the entries of Q should be small. If you are tracking people's faces, they are not likely to move with a constant velocity, so you need to crank up Q. Again, you need to be aware of the units in which your state variables are expressed.

So this is the intuition. In practice you start with some reasonable initial guess for R and Q, and then you tune them experimentally. So setting R and Q is a bit of an art. Also, in most cases using diagonal matrices for R and Q is sufficient.

Here is an [example](#) that uses the `vision.KalmanFilter` in Matlab for tracking multiple people.

answered Jan 21 '14 at 14:41



Dima

28.1k

10

50

94

Thanks for reply. I know what are Q and R physically but I wonder if there are some advice, procedures or methods to assign, for example, the model noise covariance Q. My application is a simple object tracking and I think R is "how much change" centroid/cursor position (assuming that it could be measured as a state variable only the position) due to acquisition noise; my problem is Q. For example, both linked page above and this: [studentdave.tutorials.weebly.com/...](http://studentdave.tutorials.weebly.com/...) assign same value to Q:  $(0.1)^2$ , so I wonder if there is something that I must consider – [Orlok](#) Jan 21 '14 at 21:49

1 One thing to try could be to plot the predictions and see how much they deviate from the detections. If you

are sure about R, then this will tell you if Q is right. – [Dima](#) Jan 22 '14 at 2:13

---

---