Intelligent Systems Group
Universität Duisburg-Essen
Prof. Dr. Josef Pauli / Stefan Lörcks

WS 18/19
2019-01-11

Exercises Computer/Robot Vision - 10

PROGRAMMING (7 points)

In the exercise sheet we will make use of the Kalman Filter. In the first exercise we will implement the Kalman Filter equations to estimate a trajectory in 2D points. The true state (which is unknown in real-world applications) is created for the simulation. In the second exercise we will use the Kalman Filter equations to estimate the position of a robot in an image sequence. In the third exercise we will also work on this image sequence but make use of the Kalman Filter functions, which are part of the MATLAB computer vision toolbox.

23. **Kalman Filter - I** (3 points)
    In this exercise we will implement the Kalman Filter equations estimate a 2D set of points. We create the points as the true states and create noisy measurements manually.

    a) Create a script named

    `CRV_23.m`

    The first eight lines should look like:

```
%% CRV_23_KalmanFilterParabola
% name: John Doe
% student number: 11235813

%% clean up
clear all;
close all;
clc;
```

    Of course again, fill in your name and student number!

    b) Create the ground truth of the position. Evaluate the function $f : \mathbb{R} \to \mathbb{R}, x \mapsto y := f(x) := a \cdot (x - b)^2 + c$ for the parameters $a = -0.43, b = 19, c = 150$ on the domain $X = \{1, 1.1, 1.2, \ldots, 39.8, 39.9, 40\}$.

    c) Create the state transition matrix

$$\Phi_k = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

the measurement model matrix

$$H_k = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix},$$

the system noise covariance matrix

$$Q = 10^{-6} \cdot \begin{pmatrix} 100 & 1 & 1 & 1 \\ 1 & 100 & 1 & 1 \\ 1 & 1 & 100 & 1 \\ 1 & 1 & 1 & 100 \end{pmatrix},$$

the measurement noise covariance matrix

$$R = 10^{-3} \cdot \begin{pmatrix} 500 & 1 \\ 1 & 500 \end{pmatrix}$$

and the estimated process noise covariance matrix

$$\overline{C_{k-1}} = 10^{-9} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

d) Set the initial state estimation to

$$\overline{s_{k-1}} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

e) Simulate noisy measurements for all the true positions (x_i, y_i). The noise added to the true positions should follow a normal distribution with zero mean vector and the covariance matrix R.

Generating normally distributed random vectors for a given covariance matrix C can be done using the Cholesky decomposition. Let $C = L \cdot L^T$ be the Cholesky decomposition of C, where L is the lower triangle matrix. If X is a standard normal distributed random vector, $L \cdot X$ is normal distributed with zero mean and covariance C. In Matlab `chol(C)` returns L^T. Therefore, `chol(C)'*randn(2,1)` returns a random vector with two entries, if C is a 2×2 covariance matrix.

f) Based on these noisy measurements estimate the positions using the Kalman Filter. Therefore, calculate the prediction of the state, the prediction of the state covariance matrix, the Gain-matrix, the correction of the predicted state, which gives the estimation of the new state, and the correction of the predicted state covariance matrix for every position.

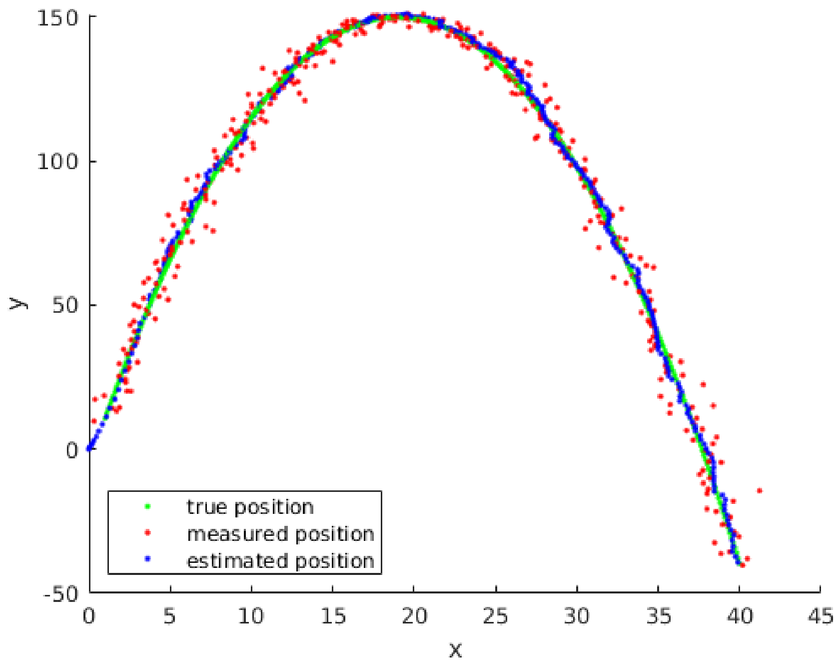

Figure 1: Example of results of Kalman Filtering for the 2D problem in exercise 23

g) Plot the results in one figure showing the true (unknown) positions, the measurements and the estimated positions. This can look like fig. 1.

h) Try out different covariance matrixes, initial state estimations and different states (e.g. set $a = -0.05$ to get a more linear movement) and examine the effect on the estimations of the position. You can also implement a more complicated model, which suits the data better.

24. **Kalman Filter - II** (2 points)
In this exercise we will implement the Kalman Filter equations to track the robot in an image sequence and estimates the robots position, when its hidden behind a wall. To get measurements of the position of the robot we will use a simple pattern matching approach, which is based on cross-correlation. The pattern and a script containing the calculation of the cross-correlation are provided.

a) Download the image pattern.png, the image sequence contained in imgSeqRobWall.zip (unzip it!) and the script

`CRV_24.m`

and fill in your name and student number!

b) Execute the script. It calculates the cross-correlation between the pattern and every image of the sequence and shows the highest correlation value and its position. Find a threshold ϑ, which allows you to distinguish between images, where the robot has been found via pattern matching and those, where the approach failed (e.g. because the robot was not visible).

c) Create the state transition matrix

$$\Phi_k = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

the measurement model matrix

$$H_k = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix},$$

the system noise covariance matrix

$$Q = 10^{-6} \cdot \begin{pmatrix} 100 & 1 & 1 & 1 \\ 1 & 100 & 1 & 1 \\ 1 & 1 & 100 & 1 \\ 1 & 1 & 1 & 100 \end{pmatrix},$$

and the estimated process noise covariance matrix

$$\overline{C_{k-1}} = 10^{-6} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

d) The measurement noise covariance matrix should depend on whether the pattern matching approach succeeded or failed (current max. correlation above or below threshold). If it succeeded (so actual measurements for the position are available) set

$$R_k = 10^{-3} \cdot \begin{pmatrix} 500 & 1 \\ 1 & 500 \end{pmatrix}$$

otherwise set

$$R_k = 10^{-3} \cdot \begin{pmatrix} 50000 & 1 \\ 1 & 50000 \end{pmatrix}$$

e) Create a matrix to save the state estimations (similiar than the matrix for the measurements). Set the initial state estimation to

$$\overline{s_1} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

f) For every image except the first one: Calculate the prediction of the state, the prediction of the state covariance matrix, the Gain-matrix, the correction of the predicted state, which gives the estimation of the new state, and the correction of the predicted state covariance matrix for every position.

g) Plot the results in one figure showing current image, the measurements and the estimated positions.

h) Try a different initialization of the first state estimation, e.g.

$$\overline{s_1} = \begin{pmatrix} 1470 \\ 700 \\ 0 \\ 0 \end{pmatrix}$$

and different parameters/matrices or even a different system model. Write down your observations as a comment in the script.

25. **Kalman Filter - III** (2 points)
In this exercise we will use the Kalman Filter functions contained in the MATLAB Computer Vision Toolbox. Therefore, its necessary to do this exercise in one of the computer centers provided by ZIM or in the exercise class. A script is provided which makes use of the Kalman Filter functions and salient points to track the robot in the fifth image sequence of exercise 25.

a) Download the script

`CRV_25.m`

and fill in your name and student number!

b) Copy the directory with the fifth image sequence from exercise 25 into your current working directory.

c) Execute the script and try out its function.

d) Try to understand the code of the script and adapt the parameters defined in the third section. Save the resulting final figure (which contains the parameter set in the title) as a png image. This can look like fig. 2.

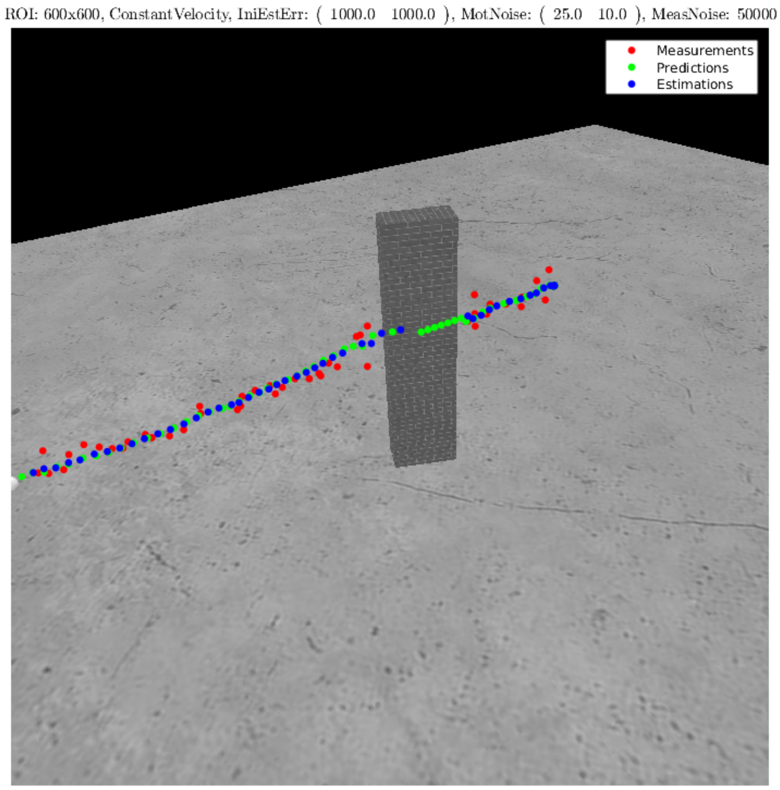

Figure 2: Example of results of exercise 25.