Computer Vision II - Homework Assignment 5

Stefan Roth, Tobias Plötz, Uwe Schmidt, GRIS, TU Darmstadt

July 2, 2014

This homework is due on July 15, 2014 at 20:00.
Please read the instructions carefully!

Goals

In this assignment, you will implement Kalman filtering, first for a synthetic dataset using learned parameters, and then for real video data using a simple dynamic model. The data for this assignment originates from Jeffrey Ho.

Problem 1 - Normal Distribution - 16 points

In the following we denote the probability density function of a multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ as

$$p(\mathbf{x}) = \mathcal{N}_{\mathbf{x}}[\boldsymbol{\mu}, \boldsymbol{\Sigma}] = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} \exp[-0.5(\mathbf{x}-\boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})]. \tag{1}$$

1. Consider a multivariate normal distribution in variable $\mathbf{x}$ with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. Show that if we make the linear transformation $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{b}$ then the transformed variable $\mathbf{y}$ is distributed as:
$$p(\mathbf{y}) = \mathcal{N}_{\mathbf{y}}[\mathbf{A}\boldsymbol{\mu} + \mathbf{b}, \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^{\mathsf{T}}].$$
You can assume that $\mathbf{y}$ follows a normal distribution. Hint: Remember the meaning of the two parameters of the normal distribution.

   8 points

2. For the 1D case, show that when we take the product of the two normal distributions with means μ_1, μ_2 and variances σ_1, σ_2 the new mean lies between the original two means and the new variance is smaller than either of the original variances.

   8 points

Note: Please give a clear and understandable derivation!

Problem 2 - Learned Kalman Model - 15 points

- In `a5p1.m`, load the file `train_data.mat`, which contains both the measurements and the hidden states of a system. Both the state vectors and measurements of the system are 2D vectors. Show the measurements and the hidden states of the training data as connected points in the plane.

  2 points

- In order to estimate the parameters $\mathbf{A}, \mathbf{W}, \mathbf{H}, \mathbf{Q}$ of the Kalman filter from the training data, use the closed-form solutions derived in the lecture (this choice of parameters maximizes the joint probability $p(\mathbf{X}_M, \mathbf{Z}_M)$). Do this in the file `learn_model.m`.

  5 points

- In `a5p1.m`, load the file `test_data.mat`, which contains further measurements and hidden states of the system. Using only the parameters obtained above as well as the measurements of the test data, infer the states of the system with a Kalman filter. Implement the filter in `kalman_filter.m`.

  5 points

- Show the measurements and the hidden states of the test data, as well as the states obtained from the Kalman filter. Report on the results with a comment in the code.

  3 points

Problem 3 - Kalman Filter for Tracking - 20 points

In this problem, you will use a Kalman filter to track a ball in a video. To perform tracking, use a linear dynamic model with constant velocity. In this case, $\mathbf{p}_{k+1} = \mathbf{p}_k + \mathbf{v}_k$ and $\mathbf{v}_{k+1} = \mathbf{v}_k$, where $\mathbf{p} = (p_1, p_2)$ gives the position of the object and $\mathbf{v} = (v_1, v_2)$ gives its velocity. A state vector has the general form $\mathbf{x} = (p_1, p_2, v_1, v_2)^\mathsf{T}$. In particular, we set

$$\mathbf{A} = \mathbf{A}_k = \begin{bmatrix} \mathbf{Id} & \mathbf{Id} \\ \mathbf{0} & \mathbf{Id} \end{bmatrix},$$

where $\mathbf{Id}$ and $\mathbf{0}$ are the 2×2 identity and zero matrices, respectively. Since we only measure the position of the object, we can use the following matrix for the measurements:

$$\mathbf{H} = \mathbf{H}_k = \begin{bmatrix} \mathbf{Id} & \mathbf{0} \end{bmatrix}.$$

In order to obtain measurements of the position of the ball, search a small region around the prior position estimate. Use SSD matching with a ball template to determine the most likely location of the ball in the region (you can use the image contained in `ball.png` as a template).

You have to guess the elements of the initial state vector. Furthermore, for the parameters $\mathbf{Q} = \mathbf{Q}_k$ and $\mathbf{W} = \mathbf{W}_k$, it should suffice to choose simple matrices.

Carry out the following items:

- Setup the model in `init_ball_model.m`.

  3 points

- Initialize the model with a state in `a5p2.m`.

  2 points

- Implement the function `find_object` that uses SSD to find an object in an image based on an estimated position. The function should handle arbitrary grayscale images.

  4 points

- The `kalman_filter` has an argument `measureHandle` to allow for measurements based on an estimated state. Implement an anonymous function in `a5p2.m` that calls the function `find_object` and pass its handle to the `kalman_filter` as `measureHandle`.

2 points

- Implement the function `draw_rectangle` that draws a colored rectangle into a grayscale image.

2 points

- Implement the function `annotate_video` that annotates the original video with bounding boxes arund the object tracked by the `kalman_filter`.

2 points

- Experiment with different settings (e.g. different initial velocities or different sizes of search regions for SSD matching). Report on your results with a comment in the code.

5 points

To facilitate this exercise, observe the following:

- We will provide a function `avi2images`, which loads an avi file and converts the frames to grayscale images. You are welcome to use or modify the function.
- For displaying a movie in Matlab, you may find this information useful: It might be necessary to initialize with `mov = moviein(nframes)`, where `nframes` gives the number of frames in the movie. To insert a truecolor image as the i-th frame, you may use `mov(:,i) = im2frame(image)`.