Computer Vision – HW 5

Javier De Velasco Oriol,
Christian Benz (zneb_naitsirhc@web.de)

July 20, 2016

Problem 1 – Normal Distribution

1. Since we can assume that $\mathbf{y}$ is normally distributed and we know that the mean of such is $\boldsymbol{\mu_y} = \mathbb{E}[\boldsymbol{y}]$ we can conclude that:

$$\begin{aligned}
\boldsymbol{\mu_y} &= \mathbb{E}[\boldsymbol{y}] \\
&= \mathbb{E}[\boldsymbol{Ax} + \boldsymbol{b}] \\
&= \boldsymbol{A}\ \mathbb{E}[\boldsymbol{x}] + \boldsymbol{b} \\
&= \boldsymbol{A\mu_x} + \boldsymbol{b}
\end{aligned}$$

   Furthermore we need to determine the covariance matrix of $\boldsymbol{y}$. It is calculated according to (cf. `https://en.wikipedia.org/wiki/Covariance_matrix`):

$$\begin{aligned}
\boldsymbol{\Sigma_y} &= \mathbb{E}[(\boldsymbol{y} - \mathbb{E}[\boldsymbol{y}])(\boldsymbol{y} - \mathbb{E}[\boldsymbol{y}])^T] \\
&= \mathbb{E}[(\boldsymbol{y} - (\boldsymbol{A\mu_x} + \boldsymbol{b}))(\boldsymbol{y} - (\boldsymbol{A\mu_x} + \boldsymbol{b}))^T] \\
&= \mathbb{E}[(\boldsymbol{Ax} + \boldsymbol{b} - (\boldsymbol{A\mu_x} + \boldsymbol{b}))(\boldsymbol{Ax} + \boldsymbol{b} - (\boldsymbol{A\mu_x} + \boldsymbol{b}))^T] \\
&= \mathbb{E}[(\boldsymbol{Ax} - \boldsymbol{A\mu_x})(\boldsymbol{Ax} - \boldsymbol{A\mu_x})^T] \\
&= \mathbb{E}[\boldsymbol{A}(\boldsymbol{x} - \boldsymbol{\mu_x})(\boldsymbol{x} - \boldsymbol{\mu_x})^T\boldsymbol{A}^T] \\
&= \mathbb{E}[\boldsymbol{A}(\boldsymbol{x} - \mathbb{E}[\boldsymbol{x}])(\boldsymbol{x} - \mathbb{E}[\boldsymbol{x}])^T\boldsymbol{A}^T] \\
&= \boldsymbol{A}\ \mathbb{E}[(\boldsymbol{x} - \mathbb{E}[\boldsymbol{x}])(\boldsymbol{x} - \mathbb{E}[\boldsymbol{x}])^T]\boldsymbol{A}^T \\
&= \boldsymbol{A\Sigma_x A}^T
\end{aligned}$$

2. We made supportive use of `https://www.cl.cam.ac.uk/~rmf25/papers/Understanding%20the%20Basis%20of%20the%20Kalman%20Filter.pdf`:

$$
\begin{aligned}
\mathcal{N}(x|\mu_1,\sigma_1^2)\cdot\mathcal{N}(x|\mu_2,\sigma_2^2) &= \frac{1}{\sqrt{2\sigma_1^2\pi}}e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}}\cdot\frac{1}{\sqrt{2\sigma_2^2\pi}}e^{-\frac{(x-\mu_2)^2}{2\sigma_2^2}} && (1)\\
&= \frac{1}{\sqrt{2\sigma_1^2\pi}}\cdot\frac{1}{\sqrt{2\sigma_2^2\pi}}\cdot e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}}\cdot e^{-\frac{(x-\mu_2)^2}{2\sigma_2^2}} && (2)\\
&= \frac{1}{2\pi\sigma_1^2\sigma_2^2}\cdot e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}-\frac{(x-\mu_2)^2}{2\sigma_2^2}} && (3)\\
&= \frac{1}{2\pi\sigma_1^2\sigma_2^2}\cdot e^{-\left(\frac{(x-\mu_1)^2}{2\sigma_1^2}+\frac{(x-\mu_2)^2}{2\sigma_2^2}\right)} && (4)\\
&= \frac{1}{2\pi\sigma_1^2\sigma_2^2}\cdot e^{-\left(\frac{2\sigma_1^2(x-\mu_2)^2+2\sigma_2^2(x-\mu_1)^2}{2\sigma_1^2 2\sigma_2^2}\right)} && (5)\\
&= \frac{1}{2\pi\sigma_1^2\sigma_2^2}\cdot e^{-\left(\frac{2\sigma_1^2(x^2-2x\mu_2+\mu_2^2)+2\sigma_2^2(x^2-2x\mu_1+\mu_1^2)}{4\sigma_1^2\sigma_2^2}\right)} && (6)\\
&= \frac{1}{2\pi\sigma_1^2\sigma_2^2}\cdot e^{-\left(\frac{2\sigma_1^2x^2-4\sigma_1^2x\mu_2+2\sigma_1^2\mu_2^2+2\sigma_2^2x^2-4\sigma_2^2\mu_1+2\sigma_2^2\mu_1^2}{4\sigma_1^2\sigma_2^2}\right)} && (7)\\
&= \frac{1}{2\pi\sigma_1^2\sigma_2^2}\cdot e^{-\left(\frac{x^2(\sigma_1^2+\sigma_2^2)-2x(\sigma_1^2\mu_2+\sigma_2^2\mu_1)+\sigma_1^2\mu_2^2+\sigma_2^2\mu_1^2)}{2\sigma_1^2\sigma_2^2}\right)} && (8)\\
&= \frac{1}{2\pi\sigma_1^2\sigma_2^2}\cdot e^{-\left(\frac{x^2-2x\frac{\sigma_1^2\mu_2+\sigma_2^2\mu_1}{\sigma_1^2+\sigma_2^2}+\frac{\sigma_1^2\mu_2^2+\sigma_2^2\mu_1^2}{\sigma_1^2+\sigma_2^2}}{2\frac{\sigma_1^2\sigma_2^2}{\sigma_1^2+\sigma_2^2}}\right)} && (9)
\end{aligned}
$$

The nominator of the exponent is in quadratic form; this gives us

$$
\begin{aligned}
\mu_{12} &= \frac{\sigma_1^2\mu_2+\sigma_2^2\mu_1}{\sigma_1^2+\sigma_2^2} && (10)\\
&= \frac{\sigma_1^2\mu_2+\sigma_2^2\mu_1+\mu_1\sigma_1^2-\mu_1\sigma_1^2}{\sigma_1^2+\sigma_2^2} && (11)\\
&= \frac{\sigma_1^2(\mu_2-\mu_1)+(\sigma_1^2+\sigma_2^2)\mu_1}{\sigma_1^2+\sigma_2^2} && (12)\\
&= \frac{\sigma_1^2(\mu_2-\mu_1)}{\sigma_1^2+\sigma_2^2}+\frac{(\sigma_1^2+\sigma_2^2)\mu_1}{\sigma_1^2+\sigma_2^2} && (13)\\
&= \frac{\sigma_1^2(\mu_2-\mu_1)}{\sigma_1^2+\sigma_2^2}+\mu_1 && (14)\\
&= k(\mu_2-\mu_1)+\mu_1 && (15)
\end{aligned}
$$

With $k = \frac{\sigma_1^2}{\sigma_1^2+\sigma_2^2}$, which denotes the Kalman gain. Assuming positive variance, $0 < k < 1$ must be true. Furthermore if $\mu_1 \leq \mu_2$:

$$\mu_1 \leq \mu_{12} \qquad \leq \mu_2 \tag{16}$$
$$\mu_1 \leq k(\mu_2 - \mu_1) + \mu_1 \qquad \leq \mu_2 \tag{17}$$
$$0 \leq k(\mu_2 - \mu_1) \qquad \leq \mu_2 - \mu_1 \tag{18}$$

Due to the prior assumption, $\mu_2 - \mu_1$ must be positive, hence the left condition is true. The right condition becomes true for the above assumed properties of k.

According to above's deduction, the variance must be:

$$\sigma_{12}^2 = \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2} \tag{19}$$
$$= \frac{\sigma_1^2\sigma_2^2 + \sigma_1^4 - \sigma_1^4}{\sigma_1^2 + \sigma_2^2} \tag{20}$$
$$= \frac{\sigma_1^2\sigma_2^2 + \sigma_1^2\sigma_1^2 - \sigma_1^2\sigma_1^2}{\sigma_1^2 + \sigma_2^2} \tag{21}$$
$$= \frac{\sigma_1^2(\sigma_2^2 + \sigma_1^2) - \sigma_1^2\sigma_1^2}{\sigma_1^2 + \sigma_2^2} \tag{22}$$
$$= \frac{\sigma_1^2(\sigma_2^2 + \sigma_1^2)}{\sigma_1^2 + \sigma_2^2} - \frac{\sigma_1^2\sigma_1^2}{\sigma_1^2 + \sigma_2^2} \tag{23}$$
$$= \sigma_1^2 - \frac{\sigma_1^2\sigma_1^2}{\sigma_1^2 + \sigma_2^2} \tag{24}$$
$$= \sigma_1^2 - k\sigma_1^2 \tag{25}$$

Again k denotes for the Kalman gain, $0 < k < 1$. Since $k\sigma_1^2 > 0$ the resulting σ_{12}^2 will be smaller than σ_1^2. Note, that the proof for $\sigma_{12}^2 < \sigma_2^2$ is analogous to the above one.

Problem 2 – Learned Kalman Model

- See Figure 1.
- See code `a5p2.jl`.
- See code `kalman_filter`.
- See Figure 2 and comment in the code.

Problem 3

- See code `a5p3.jl`.
- See code `find_object.jl`.
- The Kalman implementation can be found within the `annotate_video.jl` file.

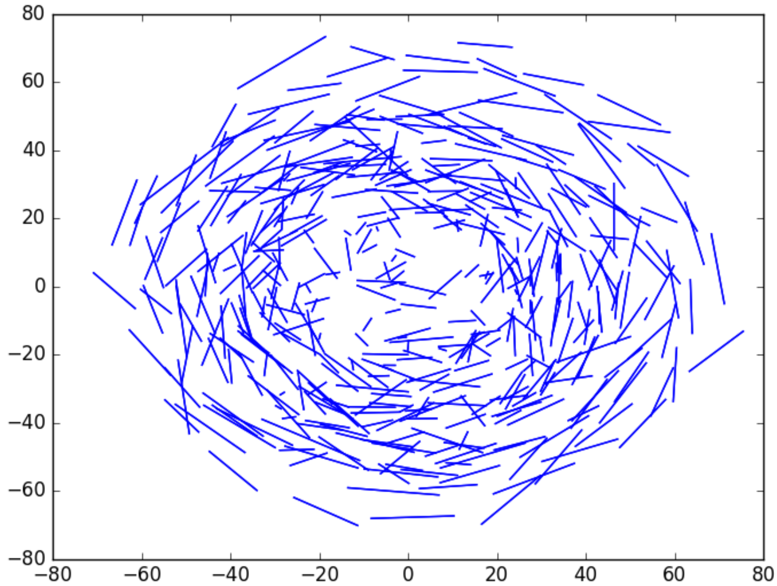

Figure 1: States and Measurements based on the Training Set

- See code `draw_rectangle.jl`.
- We started from frame 7, since the ball is not properly visible in the first few frames, which makes template matching unfeasible.
- The resulting sequence is to be found in the `result` folder. Comment in the code.

Problem 4 – Particle Filters

- Particle filters relax some of the restriction of Kalman filters and thereby form a more general approach. Particle filters, however, require a higher computational effort. The restrictive assumptions of the Kalman filter are the assumption of Gaussian noise and the linearity of the state and output transformations. As we known from optical flow, motion in image sequences is not restricted to linear transformations. Hence, the Kalman filter is incapable of representing e.g. affine transformations. Also with respect to Gaussian noise, Kalman filters have limitations when it comes to representing corruptions resulting from e.g. sensor heating effects.
- Using non-parametric approximations means to use non-Gaussian methods for approximation. Parametric approximations use exactly two parameters (mean and variance) to describe the distribution. Non-parametric methods on the other hand do need many more (possibly infinitely many) parameters in order to describe the distribution of values approriately.

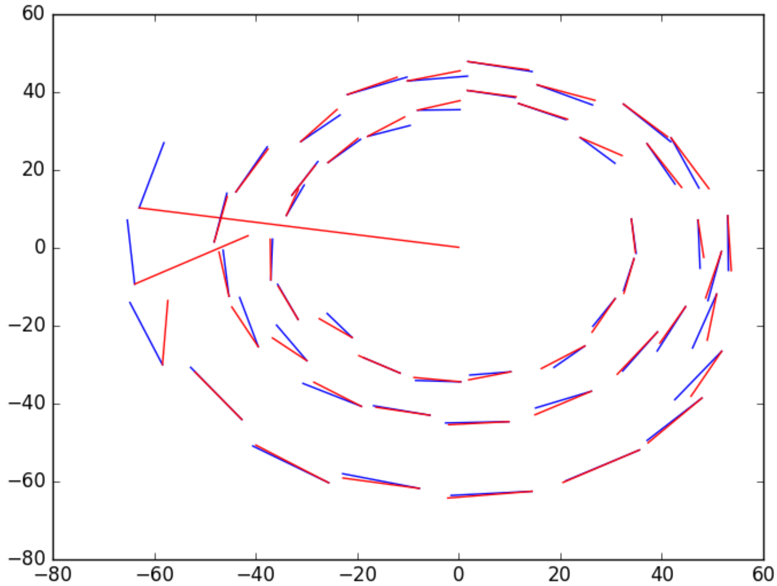

Figure 2: True and Estimated States based on the Test Set

- Particle filtering is actually a Monte Carlo method implemented to be able to determine the hidden truth values of a state-space model through the observation of noisy measurements. In our case we ant to use it to solve a hidden markov model The Monte Carlo method refers to obtaining a high amount of inputs through a random probability distributuion over a specified domain and then agreggating the results. Ideally given an infinite amount of samples it will converge to the hidden underlying truth. We use an approximation through MC method with our values of x as a sample representation taken from our density (Which we can also then weight) and then we iterate through many steps while refining and adjusting both the parameters and the probability distribution based on the observations.

- Due to how the weighting process is done usually there are few points with big weights and many points with very small weights, which leads to a degeneracy problem due to the fact that we lose particles. This can be then solved by eliminating the small particles by resampling a new set of samples that come from the weighted distribution so that these small-weighted particles are removed. This in turn will make the new samples contain less information (so our sample is now impoverished) due to the variance lost through the resampling where the small weighted particles disappear. To fix this problem we can use a continous distribution instead of a discrete one or set it so that two samples drawn are not identical. Thus resampling with a good enough kernel fixes both of our problems.