NA 568 Mobile Robotics: Methods & Algorithms Winter 2022 – Homework 3 – Nonlinear Filtering

Maani Ghaffari
University of Michigan

January 29, 2022

This is a reminder that no late HW is accepted. We drop your lowest grade from HW 1-6. It is perfectly fine to drop a zero as a HW grade. We are using Gradescope for turning in HW; see relevant information on the course Canvas site.

This problem set counts for about 7% of your course grade. You are encouraged to talk at the conceptual level with other students, but you must complete all work individually and may not share any non-trivial code or solution steps. See the syllabus for the full collaboration policy.

Submission Instructions

Your assignment must be received by 11:55 pm on Friday, February 4 (Anywhere on Earth Time). This corresponds to 6:55 AM on February 5 in Eastern Time. This is selected out of fairness to all our students, including those who take the course remotely. You are to upload your assignment directly to the Gradescope website as two attachments:

1. A `.tar.gz` or `.zip` file *containing a directory* named after your uniqname with the structure shown below.

   ```
   alincoln_hw3.tgz:
   alincoln_hw3/
   alincoln_hw3/task1a.m
   alincoln_hw3/task1b.m
   alincoln_hw3/task2a.m
   alincoln_hw3/task2b.m
   ```

   Or a Jupyter notebook per task using Python or Julia kernels for your programming. Follow the same naming convention for the notebooks .

2. A PDF with the written portion of your write-up. Scanned versions of hand-written documents, converted to PDFs, are perfectly acceptable. No other formats (e.g., .doc) are acceptable. Your PDF file should adhere to the following naming convention: `alincoln_hw3.pdf`.

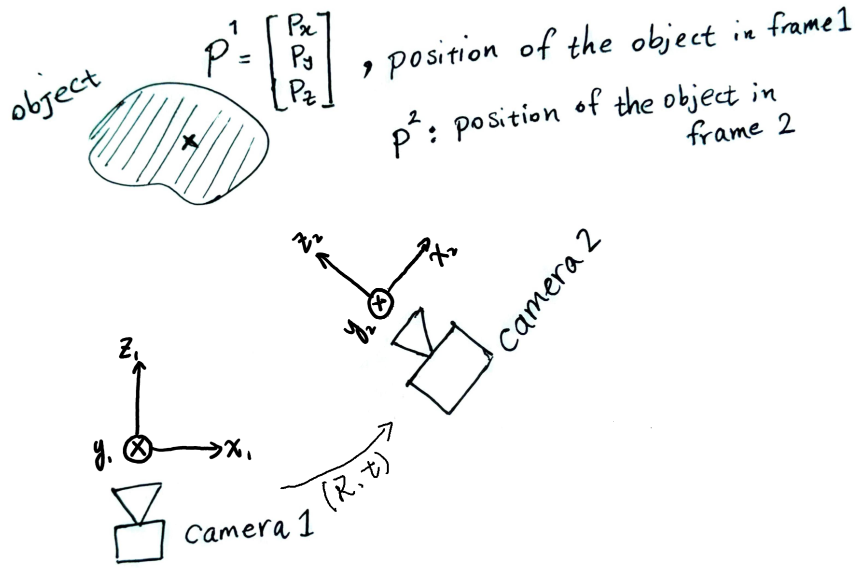

Figure 1: The problem setup for the object's position estimation using two monocular cameras.

Problem Setup

We are given a setup, shown in Figure 1, to estimate the 3D position of an object's center. The object is stationary, and there are also two fixed monocular cameras in the room. A previously developed object detection algorithm can provide the object's center in the 2D image coordinates. However, the depth is not observable by a single RGB image. Therefore, our goal is to combine measurements from two cameras to estimate the object's 3D position in the frame of camera 1. The relative pose (orientation and translation) between the two cameras is accurately known and given.

One of your colleagues (who is good at math and modeling!) has already formulated the problem. But he has very poor knowledge of state estimation, which is why he reached out to you for the rest of the solution. Luckily, you are taking the Mobile Robotics class, and you can help him out this time, so he doesn't lose his job! So far, we have the following.

Measurement Models

From an elementary computer vision knowledge, we know that a monocular pinhole camera model takes the following form

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} f_x & 0 \\ 0 & f_y \end{bmatrix} \cdot \frac{1}{p_z} \cdot \begin{bmatrix} p_x \\ p_y \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix} = K_f \pi(p) + c,$$

where $K_f = \begin{bmatrix} f_x & 0 \\ 0 & f_y \end{bmatrix}$ is called intrinsic camera matrix, $\begin{bmatrix} c_x \\ c_y \end{bmatrix}$ is the coordinates of the optical center of the image, and $\pi(\cdot)$ is the projection function (it takes a point and divides its coordinates by z, i.e., the last coordinate).

We denote the object's center in frame 1 by ${}^1p$ and ${}^2p$ as the object's center in frame 2. Given the rotation matrix, R, and the translation vector, t of frame 2 with respect to frame 1, we have the following relationship for the object's center between the two frames.

$${}^1p = R \cdot {}^2p + t,$$

and

$$ {}^2p = R^\mathsf{T} \cdot {}^1p - R^\mathsf{T} t. $$

Combining everything so far, we have the following measurement models for camera 1

$$ z = \begin{bmatrix} u \\ v \end{bmatrix} = {}^1K_f \pi({}^1p) + {}^1c := h_1({}^1p), $$

and camera 2

$$ z = \begin{bmatrix} u \\ v \end{bmatrix} = {}^2K_f \pi({}^2p) + {}^2c = {}^2K_f \pi(R^\mathsf{T} \cdot {}^1p - R^\mathsf{T} t) + {}^2c := h_2({}^1p). $$

We also assume each measurement model is corrupted by a zero-mean white Gaussian noise.

$$ \begin{aligned} z_1 &= h_1({}^1p) + v_1, \quad v_1 \sim \mathcal{N}(0, \Sigma_{v_1}), \\ z_2 &= h_2({}^1p) + v_2, \quad v_2 \sim \mathcal{N}(0, \Sigma_{v_2}). \end{aligned} $$

we may also stack two synchronized observations to form a stacked observation model as follows.

$$ z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}_{4\times 1} = \begin{bmatrix} h_1({}^1p) \\ h_2({}^1p) \end{bmatrix}_{4\times 1} + \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}_{4\times 1} := h({}^1p) + v, $$

where now $v \sim \mathcal{N}\left(0_{4\times 1}, \mathrm{blkdiag}(\Sigma_{v_1}, \Sigma_{v_2})\right)$, and $\mathrm{blkdiag}(\cdot)$ forms a block diagonal matrix.

Motion Model

Furthermore, camera 1 is attached to a structure that can vibrate. The vibration is not so severe that we assume the camera moves, but to account for inaccuracies caused by the fixture vibration we use a discrete-time random walk process as its motion model.

$$ {}^1p_{k+1} = {}^1p_k + w, \quad w \sim \mathcal{N}(0, \Sigma_w). $$

Jacobians

The Jacobians are also given using the chain rule and the fact that if $q = Rp + t$, then $\frac{\partial q}{\partial p} = R$ (you may use a symbolic math software to compute the Jacobians as well). In the following we use ${}^1p = \begin{bmatrix} p_x & p_y & p_z \end{bmatrix}^\mathsf{T}$ and ${}^2p = \begin{bmatrix} q_x & q_y & q_z \end{bmatrix}^\mathsf{T}$.

$$ H_1 = \frac{\partial h_1}{\partial\, {}^1p} = K_f \frac{\partial \pi}{\partial\, {}^1p} = K_f \begin{bmatrix} \frac{1}{p_z} & 0 & -\frac{p_x}{p_z^2} \\ 0 & \frac{1}{p_z} & -\frac{p_y}{p_z^2} \end{bmatrix}, $$

$$ H_2 = \frac{\partial h_2}{\partial\, {}^1p} = K_f \frac{\partial \pi}{\partial\, {}^1p} = K_f \frac{\partial \pi}{\partial\, {}^2p} \cdot \frac{\partial\, {}^2p}{\partial\, {}^1p} = K_f \begin{bmatrix} \frac{1}{q_z} & 0 & -\frac{q_x}{q_z^2} \\ 0 & \frac{1}{q_z} & -\frac{q_y}{q_z^2} \end{bmatrix} R^\mathsf{T}, $$

and the stacked measurement Jacobian can also be constrcuted as

$$ H_{4\times 3} = \begin{bmatrix} H_1 \\ H_2 \end{bmatrix}. $$

Provided Data Format

The processed measurements are provided in .mat and .csv files. Both files contain the same data. Unfortunately, there is no knowledge about the noise covariances, and tuning these parameters must be done manually as part of your implementation.

In the provided data,

- Kf_1 and Kf_2 are the intrinsic camera matrices for two cameras.
- C_1 and C_2 are the coordinates of the optical center of the image for two cameras.
- z_1 and z_2 are the 20 observations in the form of (u, v) for two cameras.
- R and t are the rotation matrix and the translation vector of camera 2 with respect to camera 1.

Note: We don't provide code template for this homework. You may use existing examples (`Canvas > Files > code_examples > MATLAB > target_tracking` or `Canvas > Files > code_examples > Python > target_tracking`) as you see fit.
Remark: Generally speaking, the batch update is expected to be more accurate than the sequential update, especially for EKF. This is because in a sequential update, after each correction, the linearization point is changed. The Jacobian of the next correction is not evaluated at the same point as the previous correction. This process might cause unexpected outcomes, depending on the problem. In the batch case, all information is incorporated into the filter at once.

1 EKF (40 points)

Develop an EKF to solve the problem using

A. (20 pts) A sequential measurement update that applies camera 1's correction first followed by camera 2's correction.

B. (20 pts) A batch measurement update using the stacked measurement model.

2 Particle Filter (40 points)

Develop a particle filter to solve the problem using

A. (20 pts) A sequential measurement update that applies camera 1's correction first followed by camera 2's correction.

B. (20 pts) A batch measurement update using the stacked measurement model.

3 Comparisons and Conclusion

Compile a summary of your results and report

A. (10 pts) For each of the four cases, plot the object position (the state) trajectory vs. the time steps (Figure 2 is a sample plot). Also, report the final estimated object position for each filter. Include this information in your PDF submission with appropriate labels.

B. (10 pts) Which filter performs the best and we should recommend for deployment? Explain why and discuss this in your PDF submission.

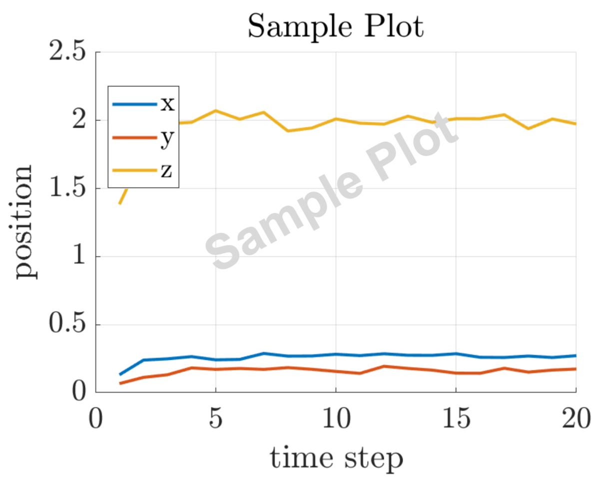

Figure 2: A sample plot of object position (the state) trajectory vs. the time steps.