U09M11109 Visual Navigation for Unmanned Aerial Vehicle (VisNav) **Spring 2023**

Lecture 7: Noise Models, Optimization, and Estimation

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor(s).*

This lecture discusses:

- how to model uncertainty and noise on positions, rotations, and poses.
- the use of optimization to mitigate the presence of measurement noise and the fact that one can choose different objective functions in these optimization problems, and
- introduces Maximum Likelihood and Maximum a Posteriori estimation which provide a probabilistically grounded way to select objective functions given assumptions on the measurement noise.

An introduction to (nonlinear) estimation is given in [1, Chapter 4]. A good introduction to uncertainty modeling for poses and rotations is [1, p. 255-283].

7.1 Representing uncertainty on positions, rotations, and poses

7.1.1 Representing uncertainty on positions

As we saw, positions and translations live in a vector space, hence we can use any multivariate distribution from traditional statistics to model an uncertain vector.

A particularly popular choice is to use a multivariate Gaussian or Normal distribution to model an uncertain vector in $\mathbb{R}^d$:

$$\boldsymbol{t} \sim \mathcal{N}(\bar{\boldsymbol{t}}, \boldsymbol{\Sigma}) \doteq \frac{1}{\sqrt{(2\pi)^d \det(\boldsymbol{\Sigma})}} \exp\left(-\frac{1}{2}(\boldsymbol{t}-\bar{\boldsymbol{t}})^{\mathsf{T}} \boldsymbol{\Sigma}^{-1} (\boldsymbol{t}-\bar{\boldsymbol{t}})\right) \tag{7.1}$$

where $\bar{\boldsymbol{t}}$ is the *mean* of the distribution and $\boldsymbol{\Sigma} \in \mathbb{R}^{d\times d}$ is the *covariance* matrix. The inverse of $\boldsymbol{\Sigma}$, is often called the *information matrix* $\boldsymbol{\Omega} \doteq \boldsymbol{\Sigma}^{-1}$.

Theorem 1 (Sum of uncertain vectors)**.** *Given two Normally distributed vectors $\boldsymbol{t}_1 \sim \mathcal{N}(\bar{\boldsymbol{t}}_1, \boldsymbol{\Sigma}_1)$ and $\boldsymbol{t}_2 \sim \mathcal{N}(\bar{\boldsymbol{t}}_2, \boldsymbol{\Sigma}_2)$, it holds:*

$$\boldsymbol{t}_1 + \boldsymbol{t}_2 \sim \mathcal{N}(\bar{\boldsymbol{t}}_1 + \bar{\boldsymbol{t}}_2, \boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2) \tag{7.2}$$

As a consequence of Theorem 1, for a fixed $\bar{\boldsymbol{t}}$, we can write $\boldsymbol{t} \sim \mathcal{N}(\bar{\boldsymbol{t}}, \boldsymbol{\Sigma})$ equivalently as:

$$\boldsymbol{t} = \bar{\boldsymbol{t}} + \boldsymbol{\epsilon}_t \qquad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\boldsymbol{0}_d, \boldsymbol{\Sigma}_t) \tag{7.3}$$

7.1.2 Representing uncertainty on rotations

The Gaussian distribution is a popular choice to represent uncertainty on positions. Unfortunately, the Gaussian is defined on a vector space and we know that rotations (and poses) are not vector spaces, but

Lie groups. Moreover, there are multiple potential choices of distributions that "behave" similarly to the Gaussian distribution. We review two of them in the following.

7.1.2.1 Wrapped Gaussian distribution

A fairly natural way to define the equivalent of a Gaussian distribution on a rotation is:

$$\boldsymbol{R} = \bar{\boldsymbol{R}} \exp(\boldsymbol{\epsilon}_r^\wedge) \qquad \boldsymbol{\epsilon}_r \sim \mathcal{N}(\mathbf{0}_d, \boldsymbol{\Sigma}_r) \tag{7.4}$$

The previous relation implies a distribution over the rotation $\boldsymbol{R}$:

$$\boldsymbol{R} \sim \mathcal{W}(\bar{\boldsymbol{R}}, \boldsymbol{\Sigma}_r) \doteq \frac{1}{\boldsymbol{J}_r(\log(\bar{\boldsymbol{R}}^\mathsf{T}\boldsymbol{R})^\vee)} \frac{1}{\sqrt{(2\pi)^d \det(\boldsymbol{\Sigma}_r)}} \exp\left(-\frac{1}{2}(\log(\bar{\boldsymbol{R}}^\mathsf{T}\boldsymbol{R})^\vee)^\mathsf{T}\boldsymbol{\Sigma}_r^{-1}(\log(\bar{\boldsymbol{R}}^\mathsf{T}\boldsymbol{R})^\vee)\right) \tag{7.5}$$

where $\bar{\boldsymbol{R}}$ is the *mean* rotation and $\boldsymbol{\Sigma}_r$ is the covariance of the distribution.

When $\boldsymbol{\Sigma}_r$ is "small", $\boldsymbol{\epsilon}_r = \log(\bar{\boldsymbol{R}}^\mathsf{T}\boldsymbol{R})^\vee$ is also small (with high probability) and $\boldsymbol{J}_r(\boldsymbol{\epsilon}_r) \approx \mathbf{I}_d$ and the distribution (7.6) is approximated as:

$$\mathcal{W}(\bar{\boldsymbol{R}}, \boldsymbol{\Sigma}_r) \approx \frac{1}{\sqrt{(2\pi)^d \det(\boldsymbol{\Sigma}_r)}} \exp\left(-\frac{1}{2}(\log(\bar{\boldsymbol{R}}^\mathsf{T}\boldsymbol{R})^\vee)^\mathsf{T}\boldsymbol{\Sigma}_r^{-1}(\log(\bar{\boldsymbol{R}}^\mathsf{T}\boldsymbol{R})^\vee)\right) \tag{7.6}$$

16.1.2.2 Isotropic Langevin distribution

The Isotropic Langevin distribution is defined as [3]:

$$\mathcal{L}(\bar{\boldsymbol{R}}, \kappa) = \frac{1}{c_d(\kappa)} \exp(\kappa \operatorname{tr}\left(\bar{\boldsymbol{R}}^\mathsf{T}\boldsymbol{R}\right)) \tag{7.7}$$

where $\bar{\boldsymbol{R}}$ and κ are called the mode and the concentration parameter of the distribution, and $c_d(\kappa)$ is a normalization constant.

In the 2D case, the Langevin distribution is known as the Von Mises distribution:

$$\mathcal{V}(\bar{\theta}, \kappa) = \frac{1}{c_2(\kappa)} \exp(\kappa \ \cos(\theta - \bar{\theta})) \tag{7.8}$$

where the equivalence can be seen from $\operatorname{tr}\left(\bar{\boldsymbol{R}}^\mathsf{T}\boldsymbol{R}\right) = \operatorname{tr}\left(\boldsymbol{R}(\bar{\theta})^\mathsf{T}\boldsymbol{R}(\theta)\right) = 2\cos(\theta - \bar{\theta})$, where the factor "2" is included in the normalization constant $c_2(\kappa)$.

7.1.3 Representing uncertainty on poses

A fairly natural way to define the equivalent of a Gaussian distribution on a pose is:

$$\boldsymbol{T} = \bar{\boldsymbol{T}} \exp(\boldsymbol{\epsilon}_T^\wedge) \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_T) \tag{7.9}$$

where $\bar{\boldsymbol{T}}$ is the *mean* pose and $\boldsymbol{\Sigma}_T$ is the covariance of the distribution.

In alternative, one can consider a pose at a pair of rotation and translation and assume (uncorrelated) distributions on each, e.g., use a Langevin distribution for the rotation and a Gaussian distribution on the translation. A longer discussion on how to represent uncertainty on poses is given by [2].

16.2 Estimating unknown quantities via optimization

Similarly to what we often suggested in the previous lectures, here we discuss how to use optimization to mitigate the presence of noise in the measurements, when estimating some unknown quantity. In particular, we consider 2 examples: point triangulation, and bundle adjustment.

16.2.1 Example 1: Point triangulation

Consider the case where we have two cameras at known poses $(\boldsymbol{R}_{\mathrm{w}}^{\mathrm{c}_1}, \boldsymbol{t}_{\mathrm{w}}^{\mathrm{c}_1})$ and $(\boldsymbol{R}_{\mathrm{w}}^{\mathrm{c}_2}, \boldsymbol{t}_{\mathrm{w}}^{\mathrm{c}_2})$ and known calibration matrices $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$. Assume that we measure the pixel projection $\tilde{\boldsymbol{x}}_1$ and $\tilde{\boldsymbol{x}}_2$ of an unknown 3D point $\boldsymbol{p}^{\mathrm{w}}$ in both cameras. The triangulation problem (or "structure reconstruction") is: estimate the 3D position of the point given the pixel measurements.

From basic perspective projection, we know that:

$$\lambda_1 \tilde{\boldsymbol{x}}_1 = \boldsymbol{K}_1 [\boldsymbol{R}_{\mathrm{w}}^{\mathrm{c}_1} \ \boldsymbol{t}_{\mathrm{w}}^{\mathrm{c}_1}] \tilde{\boldsymbol{p}}^{\mathrm{w}} \tag{7.10}$$

$$\lambda_2 \tilde{\boldsymbol{x}}_2 = \boldsymbol{K}_2 [\boldsymbol{R}_{\mathrm{w}}^{\mathrm{c}_2} \ \boldsymbol{t}_{\mathrm{w}}^{\mathrm{c}_2}] \tilde{\boldsymbol{p}}^{\mathrm{w}} \tag{16.11}$$

Using a more compact notation:

$$\lambda_1 \tilde{\boldsymbol{x}}_1 = \boldsymbol{\Pi}_1 \tilde{\boldsymbol{p}}^{\mathrm{w}} \tag{7.12}$$

$$\lambda_2 \tilde{\boldsymbol{x}}_2 = \boldsymbol{\Pi}_2 \tilde{\boldsymbol{p}}^{\mathrm{w}} \tag{7.13}$$

where $\boldsymbol{\Pi}_1 = \boldsymbol{K}_1 [\boldsymbol{R}_{\mathrm{w}}^{\mathrm{c}_1} \ \boldsymbol{t}_{\mathrm{w}}^{\mathrm{c}_1}]$ and $\boldsymbol{\Pi}_2 = \boldsymbol{K}_2 [\boldsymbol{R}_{\mathrm{w}}^{\mathrm{c}_2} \ \boldsymbol{t}_{\mathrm{w}}^{\mathrm{c}_2}]$.

To get rid of the scale λ_1 and λ_2, let us multiply the first equation by $[\tilde{\boldsymbol{x}}_1]_\times$ and the second equation by $[\tilde{\boldsymbol{x}}_2]_\times$, from which we get:

$$[\tilde{\boldsymbol{x}}_1]_\times \boldsymbol{\Pi}_1 \tilde{\boldsymbol{p}}^{\mathrm{w}} = \mathbf{0} \tag{7.14}$$

$$[\tilde{\boldsymbol{x}}_2]_\times \boldsymbol{\Pi}_2 \tilde{\boldsymbol{p}}^{\mathrm{w}} = \mathbf{0} \tag{7.15}$$

or more succinctly:

$$\begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_2 \end{bmatrix} \tilde{\boldsymbol{p}}^{\mathrm{w}} = \mathbf{0} \tag{7.16}$$

where $\boldsymbol{A}_1 = [\tilde{\boldsymbol{x}}_1]_\times \boldsymbol{\Pi}_1$ and $\boldsymbol{A}_2 = [\tilde{\boldsymbol{x}}_2]_\times \boldsymbol{\Pi}_2$.

In the presence of noise the equation above (6 equalities in 3 variables) will have no solution, hence we rather look for a least square solution:

$$\min_{\|\tilde{\boldsymbol{p}}^{\mathrm{w}}\|=1} \left\| \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_2 \end{bmatrix} \tilde{\boldsymbol{p}}^{\mathrm{w}} \right\|^2 = \min_{\|\tilde{\boldsymbol{p}}^{\mathrm{w}}\|=1} \|\boldsymbol{A}\tilde{\boldsymbol{p}}^{\mathrm{w}}\|^2 \qquad \text{with } \boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_2 \end{bmatrix} \tag{7.17}$$

which can be solved in closed form (solution is the eigenvector corresponding to the smallest eigenvalue of $\boldsymbol{A}^\mathsf{T}\boldsymbol{A}$). The objective in (7.17) in said to minimize the "algebraic" error.

An alternative objective. Recall that the pixel measurements can be written as:

$$\boldsymbol{x}_1 = \begin{bmatrix} u_1 \\ v_1 \end{bmatrix} = \begin{bmatrix} \frac{[\boldsymbol{\Pi}_1 \tilde{\boldsymbol{p}}^{\mathrm{w}}]_1}{[\boldsymbol{\Pi}_1 \tilde{\boldsymbol{p}}^{\mathrm{w}}]_3} \\ \frac{[\boldsymbol{\Pi}_1 \tilde{\boldsymbol{p}}^{\mathrm{w}}]_2}{[\boldsymbol{\Pi}_1 \tilde{\boldsymbol{p}}^{\mathrm{w}}]_3} \end{bmatrix} = \pi(\boldsymbol{R}_{\mathrm{c}_1}^{\mathrm{w}}, \boldsymbol{t}_{\mathrm{c}_1}^{\mathrm{w}}, \boldsymbol{p}^{\mathrm{w}}) \tag{7.18}$$

$$\boldsymbol{x}_2 = \begin{bmatrix} u_2 \\ v_2 \end{bmatrix} = \begin{bmatrix} \frac{[\boldsymbol{\Pi}_2 \tilde{\boldsymbol{p}}^{\mathrm{w}}]_1}{[\boldsymbol{\Pi}_2 \tilde{\boldsymbol{p}}^{\mathrm{w}}]_3} \\ \frac{[\boldsymbol{\Pi}_2 \tilde{\boldsymbol{p}}^{\mathrm{w}}]_2}{[\boldsymbol{\Pi}_2 \tilde{\boldsymbol{p}}^{\mathrm{w}}]_3} \end{bmatrix} = \pi(\boldsymbol{R}_{\mathrm{c}_2}^{\mathrm{w}}, \boldsymbol{t}_{\mathrm{c}_2}^{\mathrm{w}}, \boldsymbol{p}^{\mathrm{w}}) \tag{7.19}$$

where on the right-hand-side we introduced a more compact notation for the perspective projection function $\pi(\cdot)$. Therefore, an alternative objective can be obtained as follows:

$$\min_{\boldsymbol{p}^{\mathrm{w}}} \|\boldsymbol{x}_1 - \pi(\boldsymbol{R}^{\mathrm{w}}_{\mathrm{c}_1}, \boldsymbol{t}^{\mathrm{w}}_{\mathrm{c}_1}, \boldsymbol{p}^{\mathrm{w}})\|^2 + \|\boldsymbol{x}_2 - \pi(\boldsymbol{R}^{\mathrm{w}}_{\mathrm{c}_2}, \boldsymbol{t}^{\mathrm{w}}_{\mathrm{c}_2}, \boldsymbol{p}^{\mathrm{w}})\|^2 \tag{7.20}$$

In general, the solution of (7.20) will be different from the solution of (7.17). The objective in (7.20) in said to minimize the "geometric" error.

Invariance. The formulation (7.20) is invariant to a rigid-body transformation of all the two cameras. Numerically, one can show that the formulation (7.17), instead, is not invariant to rigid-body transformations.

7.2.2 Example 2: Bundle Adjustment

In the previous lecture we have seen how to estimate the relative pose between two cameras given N pixel correspondences (let us assume for now that there are no outliers). Now let's assume that we can track the same N features across 3 consecutive images, i.e., we can detect and match features in image 1 and 2, and then we can match features in image 2 against features extracted in image 3.

As done in the previous lecture we can compute the motion between camera 1 and 2 and between 2 and 3 by estimating the corresponding essential matrices:

$$\boldsymbol{E}_{12} = \arg\min_{\boldsymbol{E}_{12} \in \mathcal{S}_E} \sum_{k=1}^{N} |\tilde{\boldsymbol{y}}_{k,2}^{\mathsf{T}} \boldsymbol{E}_{12} \tilde{\boldsymbol{y}}_{k,1}|^2 \tag{7.21}$$

$$\boldsymbol{E}_{23} = \arg\min_{\boldsymbol{E}_{23} \in \mathcal{S}_E} \sum_{k=1}^{N} |\tilde{\boldsymbol{y}}_{k,3}^{\mathsf{T}} \boldsymbol{E}_{23} \tilde{\boldsymbol{y}}_{k,2}|^2 \tag{7.22}$$

from which one can compute the relative poses (up to scale).

An alternative objective. An alternative approach to estimate the poses between the three cameras is to formulate a single optimization problem:

$$\min_{\substack{(\boldsymbol{R}^{\mathrm{w}}_{\mathrm{c}_i}, \boldsymbol{t}^{\mathrm{w}}_{\mathrm{c}_i}), i=1,2,3 \\ \boldsymbol{p}^{\mathrm{w}}_k, k=1,\ldots,N}} \sum_{k=1}^{N} \sum_{i=1}^{3} \|\boldsymbol{x}_{k,i} - \pi(\boldsymbol{R}^{\mathrm{w}}_{\mathrm{c}_i}, \boldsymbol{t}^{\mathrm{w}}_{\mathrm{c}_i}, \boldsymbol{p}^{\mathrm{w}}_k)\|^2 \tag{7.23}$$

where we minimize the *reprojection error* of the N points in each camera. Note that in (7.23), besides optimizing for the camera poses, we also optimize for the unknown positions of the 3D points $\boldsymbol{p}^{\mathrm{w}}_k$, $k = 1, \ldots, N$.

The previous formulation, which easily extends to the case of K cameras is known as *bundle adjustment*. Since it does not admit a closed-form solution in general, in the next lecture we will discuss numerical optimization methods to obtain a (local) minimizer of (7.23).

Invariance. The formulation (7.23) is invariant to a rigid-body transformation of all the cameras.

16.3 Maximum Likelihood and Maximum a Posteriori estimation

The discussion in the previous section highlights a basic question: what is the "right" objective function to minimize in order to estimate a variable of interest? Each minimization provides a different solution so is there a grounded way to pick an objective function?

This question finds a satisfactory answer in estimation theory, where there are two frameworks (*Maximum Likelihood* and *Maximum a Posteriori* estimation, discussed below) that discuss how the assumption we make on the measurement noise dictates a meaningful objective function to minimize.

16.3.1 Maximum Likelihood Estimation (MLE)

Assume we are given N measurements $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N$ (e.g., pixel measurements) that are function of a variable we want to estimate $\boldsymbol{x}$ (e.g., camera poses, points). Assume that we are also given the conditional distributions:

$$\mathbb{P}\left(\boldsymbol{z}_j | \boldsymbol{x}\right) \tag{7.24}$$

Than the *maximum likelihood* estimator (MLE) is defined as:

$$\boldsymbol{x}_{\text{MLE}} = \arg\max_{\boldsymbol{x}} \mathbb{P}\left(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N | \boldsymbol{x}\right) \tag{7.25}$$

where $\mathbb{P}\left(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N | \boldsymbol{x}\right)$ is also called the *measurement likelihood*. Equivalently:

$$\boldsymbol{x}_{\text{MLE}} = \arg\min_{\boldsymbol{x}} - \log \mathbb{P}\left(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N | \boldsymbol{x}\right) \tag{7.26}$$

Assuming conditional independence between the measurements, the MLE estimator becomes:

$$\boldsymbol{x}_{\text{MLE}} = \arg\max \prod_{j=1}^{N} \mathbb{P}\left(\boldsymbol{z}_j | \boldsymbol{x}\right) \tag{7.27}$$

or equivalently:

$$\boldsymbol{x}_{\text{MLE}} = \arg\min - \log \prod_{j=1}^{N} \mathbb{P}\left(\boldsymbol{z}_j | \boldsymbol{x}\right) = \arg\min - \sum_{j=1}^{N} \log \mathbb{P}\left(\boldsymbol{z}_j | \boldsymbol{x}\right) \tag{7.28}$$

Example: Linear measurements with Gaussian noise. Assume we have N random measurements of an unknown variable $\boldsymbol{x}$ we want to estimate:

$$\boldsymbol{z}_j = \boldsymbol{A}_j \boldsymbol{x} + \boldsymbol{\epsilon}_j \tag{7.29}$$

where $\boldsymbol{A}_j$ are known matrices of suitable dimensions and $\boldsymbol{\epsilon}_j \in \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_j)$.

Now we note that if $\boldsymbol{\epsilon}_j \in \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_j)$, then $\boldsymbol{z}_j \in \mathcal{N}(\boldsymbol{A}_j \boldsymbol{x}, \boldsymbol{\Sigma}_j)$ or equivalently:

$$\mathbb{P}\left(\boldsymbol{z}_i | \boldsymbol{x}\right) = \frac{1}{\kappa_j} \exp -\frac{1}{2} (\boldsymbol{z}_j - \boldsymbol{A}_j \boldsymbol{x})^{\mathsf{T}} \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{z}_j - \boldsymbol{A}_j \boldsymbol{x}) \tag{7.30}$$

where κ_i is the normalization constant of the Normal distribution. Then the MLE is:

$$\boldsymbol{x}_{\text{MLE}} = \arg\min - \sum_{j=1}^{N} \log \mathbb{P}\left(\boldsymbol{z}_j | \boldsymbol{x}\right) = \arg\min - \sum_{j=1}^{N} \log \frac{1}{\kappa_j} \exp -\frac{1}{2} (\boldsymbol{z}_j - \boldsymbol{A}_j \boldsymbol{x})^{\mathsf{T}} \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{z}_j - \boldsymbol{A}_j \boldsymbol{x}) = \tag{7.31}$$

$$\arg\min - \sum_{j=1}^{N} \log \frac{1}{\kappa_j} - \sum_{j=1}^{N} \log \exp -\frac{1}{2} (\boldsymbol{z}_j - \boldsymbol{A}_j \boldsymbol{x})^{\mathsf{T}} \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{z}_j - \boldsymbol{A}_j \boldsymbol{x}) = \tag{7.32}$$

$$\arg\min \sum_{j=1}^{N} \frac{1}{2} (\boldsymbol{z}_j - \boldsymbol{A}_j \boldsymbol{x})^{\mathsf{T}} \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{z}_j - \boldsymbol{A}_j \boldsymbol{x}) \tag{7.33}$$

when $\mathbf{\Sigma}_j = \mathbf{I}$ for all j, we obtain standard least squares:

$$\boldsymbol{x}_{\text{MLE}} = \arg\min \sum_{j=1}^{N} \frac{1}{2} \|\boldsymbol{z}_j - \boldsymbol{A}_j \boldsymbol{x}\|^2 \tag{7.34}$$

Example: Nonlinear measurements with additive Gaussian noise. Assume we have N random measurements of an unknown variable $\boldsymbol{x}$ we want to estimate:

$$\boldsymbol{z}_j = f_j(\boldsymbol{x}) + \boldsymbol{\epsilon}_j \tag{7.35}$$

Repeating the same derivation of the linear case, we get that the MLE is:

$$\boldsymbol{x}_{\text{MLE}} = \arg\min \sum_{j=1}^{N} \frac{1}{2} (\boldsymbol{z}_j - f_j(\boldsymbol{x}))^{\mathsf{T}} \mathbf{\Sigma}_j^{-1} (\boldsymbol{z}_j - f_j(\boldsymbol{x})) \tag{7.36}$$

which is a *nonlinear least squares* problem.

Example: Translation averaging. Let us assume we are given N measurements of an unknown translation $\boldsymbol{t}$. Assume that each measurement is affected by zero mean Gaussian noise with identity covariance:

$$\bar{\boldsymbol{t}}_i = \boldsymbol{t} + \boldsymbol{\epsilon}_i, \quad i = 1, \ldots, N \quad \text{with } \boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{7.37}$$

Following the same derivation of the linear case with Gaussian noise in the previous example, we obtain the MLE estimator:

$$\boldsymbol{t}_{\text{MLE}} = \arg\min_{\boldsymbol{t}} \sum_{i=1}^{N} \|\boldsymbol{t} - \bar{\boldsymbol{t}}_i\|^2 \tag{7.38}$$

whose solution can be computed in closed form as:

$$\boldsymbol{t}_{\text{MLE}} = \frac{1}{N} \sum_{i=1}^{N} \bar{\boldsymbol{t}}_i \tag{7.39}$$

Example: Rotation averaging. Let us assume we are given N measurements of an unknown rotation $\boldsymbol{R}$. Assume that each measurement $\bar{\boldsymbol{R}}_i$ is modeled as:

$$\bar{\boldsymbol{R}}_i = \boldsymbol{R}\boldsymbol{R}_\epsilon, \quad i = 1, \ldots, N \quad \text{with } \boldsymbol{R}_\epsilon \sim \mathcal{L}(\mathbf{I}, \kappa) \tag{7.40}$$

where $\mathcal{L}(\mathbf{I}, \kappa)$ is the Langevin distribution (introduced in Section 7.1.2.2), with the identity mode and concentration parameter κ. Let us assume $\kappa = 1$.

Applying Maximum Likelihood Estimation, we obtain the following estimator (you will prove this as part of Lab 7):

$$\boldsymbol{R}_{\text{MLE}} = \arg\min_{\boldsymbol{R} \in \text{SO}(d)} \sum_{i=1}^{N} \|\boldsymbol{R} - \bar{\boldsymbol{R}}_i\|_F^2 \tag{7.41}$$

One would be tempted to compute $\boldsymbol{R}_{\text{MLE}}$ as the average of $\bar{\boldsymbol{R}}_i$ as we did in the previous examples, but unfortunately, that would not be a valid rotation, and hence violate the constraint $\boldsymbol{R} \in \text{SO}(d)$.[1] Therefore, we need to compute the correct "average" rotation by solving (7.41) (including the constraint $\boldsymbol{R} \in \text{SO}(d)$).

[1]Such approach would be incorrect even in the case of 2D rotations, where each rotation $\bar{\boldsymbol{R}}_i$ can be identified by a single angle $\bar{\theta}_i$. In this case, one would suggest $\theta_{\text{MLE}} = \frac{1}{N}\sum_{i=1}^{N} \bar{\theta}_i$, but it is easy to see that this does not lead to a meaningful average (e.g., pick $\bar{\theta}_1 = \frac{3}{4}\pi$ and $\bar{\theta}_2 = -\frac{3}{4}\pi$: the arithmetic average is zero, while one would expect an average rotation of π).

It turns out this optimization problem admits a closed-form solution:

$$\arg\min_{\boldsymbol{R}\in\mathrm{SO}(d)} \sum_{i=1}^{N} \|\boldsymbol{R} - \bar{\boldsymbol{R}}_i\|_F^2 = \arg\min_{\boldsymbol{R}\in\mathrm{SO}(d)} \sum_{i=1}^{N} \|\boldsymbol{R}\|_F^2 + \|\bar{\boldsymbol{R}}_i\|_F^2 - 2\mathrm{tr}\left(\bar{\boldsymbol{R}}_i^\mathsf{T}\boldsymbol{R}\right) = \arg\min_{\boldsymbol{R}\in\mathrm{SO}(d)} \sum_{i=1}^{N} 2d - 2\mathrm{tr}\left(\bar{\boldsymbol{R}}_i^\mathsf{T}\boldsymbol{R}\right) \quad (7.42)$$

Dropping constants (irrelevant for optimization) and calling $\boldsymbol{N} \doteq \sum_{i=1}^{N} \bar{\boldsymbol{R}}_i$:

$$\arg\min_{\boldsymbol{R}\in\mathrm{SO}(d)} \sum_{i=1}^{N} 2d - 2\mathrm{tr}\left(\bar{\boldsymbol{R}}_i^\mathsf{T}\boldsymbol{R}\right) = \arg\min_{\boldsymbol{R}\in\mathrm{SO}(d)} \sum_{i=1}^{N} -\mathrm{tr}\left(\bar{\boldsymbol{R}}_i^\mathsf{T}\boldsymbol{R}\right) = \arg\min_{\boldsymbol{R}\in\mathrm{SO}(d)} -\mathrm{tr}\left((\sum_{i=1}^{N} \bar{\boldsymbol{R}}_i)^\mathsf{T}\boldsymbol{R}\right) = \arg\max_{\boldsymbol{R}\in\mathrm{SO}(d)}\ \mathrm{tr}\left(\boldsymbol{N}^\mathsf{T}\boldsymbol{R}\right)$$

Now, once again note that:

$$\mathrm{tr}\left(\boldsymbol{N}^\mathsf{T}\boldsymbol{R}\right) = \frac{1}{2}\left(\|\boldsymbol{N}\|_F^2 + \|\boldsymbol{R}\|_F^2 - \|\boldsymbol{N}-\boldsymbol{R}\|_F^2\right) = \frac{1}{2}\left(\|\boldsymbol{N}\|_F^2 + d - \|\boldsymbol{N}-\boldsymbol{R}\|_F^2\right). \quad (7.43)$$

Since $\boldsymbol{N}$ is constant we have:

$$\arg\max_{\boldsymbol{R}\in\mathrm{SO}(d)}\ \mathrm{tr}\left(\boldsymbol{N}^\mathsf{T}\boldsymbol{R}\right) = \arg\min_{\boldsymbol{R}\in\mathrm{SO}(d)}\ \|\boldsymbol{N}-\boldsymbol{R}\|_F \quad (7.44)$$

which is simply a projection of $\boldsymbol{N}$ onto SO(3). Such a projection can be computed as follows [4] (we also saw this in Lecture 15):

$$\arg\min_{\boldsymbol{R}\in\mathrm{SO}(d)}\ \|\boldsymbol{N}-\boldsymbol{R}\|_F = \begin{cases} \boldsymbol{U}\boldsymbol{V}^\mathsf{T} & \text{if } \det(\boldsymbol{U}\boldsymbol{V}^\mathsf{T}) \geq 0 \\ \boldsymbol{U}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}\boldsymbol{V}^\mathsf{T} & \text{otherwise} \end{cases} \quad (7.45)$$

where $\boldsymbol{N} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\mathsf{T}$ is a singular value decomposition of the matrix $\boldsymbol{N}$.

Therefore, the optimal solution to the rotation "averaging" problem (7.41) is to compute the Euclidean average $\boldsymbol{N} \doteq \sum_{i=1}^{N} \bar{\boldsymbol{R}}_i$ and then project this matrix to SO(3).

16.3.2 Maximum a Posteriori Estimation (MAP)

Assume we are given N measurements $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N$ (e.g., pixel measurements) that are function of a variable we want to estimate $\boldsymbol{x}$ (e.g., camera poses, points). *Maximum a Posteriori Estimation* (MAP) is a generalization of MLE. The MAP estimator is:

$$\boldsymbol{x}_{\mathrm{MAP}} = \arg\max_{\boldsymbol{x}} \mathbb{P}\left(\boldsymbol{x}|\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N\right) \quad (7.46)$$

Using Bayes rule:

$$\boldsymbol{x}_{\mathrm{MAP}} = \arg\max_{\boldsymbol{x}} \mathbb{P}\left(\boldsymbol{x}|\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N\right) = \quad (7.47)$$

$$\arg\max_{\boldsymbol{x}} \frac{\mathbb{P}\left(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N|\boldsymbol{x}\right)\mathbb{P}\left(\boldsymbol{x}\right)}{\mathbb{P}\left(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N\right)} = \quad (7.48)$$

$$\arg\max_{\boldsymbol{x}} \mathbb{P}\left(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N|\boldsymbol{x}\right)\mathbb{P}\left(\boldsymbol{x}\right) \quad (7.49)$$

where again $\mathbb{P}\left(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N|\boldsymbol{x}\right)$ is the *likelihood* of the measurements given $\boldsymbol{x}$, and $\mathbb{P}\left(\boldsymbol{x}\right)$ is a *prior* probability over $\boldsymbol{x}$. It is easy to see that MAP reduces to MLE when the prior is uniform ($\mathbb{P}\left(\boldsymbol{x}\right)$ is constant).

Assuming independence between the measurements, and repeating the same derivation of the previous section:

$$\boldsymbol{x}_{\mathrm{MAP}} = \arg\min_{\boldsymbol{x}} -\sum_{j=1}^{N} \log\mathbb{P}\left(\boldsymbol{z}_j|\boldsymbol{x}\right) - \log\mathbb{P}\left(\boldsymbol{x}\right) \quad (7.50)$$

MLE Example: triangulation. Assume that a 3D point is observed by two known cameras and that pixel measurements are affected by Gaussian noise:

$$\boldsymbol{x}_1 = \pi(\boldsymbol{R}^{\mathrm{w}}_{\mathrm{c}_1}, \boldsymbol{t}^{\mathrm{w}}_{\mathrm{c}_1}, \boldsymbol{p}^{\mathrm{w}}) + \boldsymbol{\epsilon}_1 \tag{7.51}$$

$$\boldsymbol{x}_2 = \pi(\boldsymbol{R}^{\mathrm{w}}_{\mathrm{c}_2}, \boldsymbol{t}^{\mathrm{w}}_{\mathrm{c}_2}, \boldsymbol{p}^{\mathrm{w}}) + \boldsymbol{\epsilon}_2 \tag{7.52}$$

with $\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_2)$. Then the MLE is:

$$(\boldsymbol{p}^{\mathrm{w}})^\star = \arg\min_{\boldsymbol{p}^{\mathrm{w}}} \|\boldsymbol{x}_1 - \pi(\boldsymbol{R}^{\mathrm{w}}_{\mathrm{c}_1}, \boldsymbol{t}^{\mathrm{w}}_{\mathrm{c}_1}, \boldsymbol{p}^{\mathrm{w}})\|^2 + \|\boldsymbol{x}_2 - \pi(\boldsymbol{R}^{\mathrm{w}}_{\mathrm{c}_2}, \boldsymbol{t}^{\mathrm{w}}_{\mathrm{c}_2}, \boldsymbol{p}^{\mathrm{w}}) \tag{7.53}$$

MAP Example: bundle adjustment. Assume that N 3D points are observed by K cameras and that pixel measurements are affected by Gaussian noise. Moreover, assume we have a prior on the first camera being at the origin of the reference frame.

$$\min_{(\boldsymbol{R}^{\mathrm{w}}_{\mathrm{c}_i}, \boldsymbol{t}^{\mathrm{w}}_{\mathrm{c}_i}), i=1,\ldots,K} \sum_{k=1}^{N} \sum_{i=1}^{K} \|u_{k,i} - \pi(\boldsymbol{R}^{\mathrm{w}}_{\mathrm{c}_i}, \boldsymbol{t}^{\mathrm{w}}_{\mathrm{c}_i}, \boldsymbol{p}_k)\|^2 + \rho\|\boldsymbol{t}^{\mathrm{w}}_{\mathrm{c}_1}\|^2 + \omega\|\boldsymbol{R}^{\mathrm{w}}_{\mathrm{c}_1} - \mathbf{I}_3\|_F^2 \tag{7.54}$$

where we assumed the prior:

$$\mathbb{P}\left(\boldsymbol{t}^{\mathrm{w}}_{\mathrm{c}_1}\right) = \mathcal{N}(\mathbf{0}, \frac{1}{\rho}\mathbf{I}_3) \tag{7.55}$$

$$\mathbb{P}\left(\boldsymbol{R}^{\mathrm{w}}_{\mathrm{c}_1}\right) = \mathrm{Langevin}(\mathbf{I}_3, \omega) \tag{7.56}$$

References

[1] T. Barfoot. *State Estimation for Robotics*. Cambridge University Press, 2017.

[2] T. D. Barfoot and P. T. Furgale. Associating uncertainty with three-dimensional poses for use in estimation problems. *IEEE Trans. Robotics*, 30(3):679–693, 2014.

[3] A. Chiuso, G. Picci, and S. Soatto. Wide-sense estimation on the special orthogonal group. *Commun. Inf. Syst*, 8:185–200, 2008.

[4] R. Hartley, J. Trumpf, Y. Dai, and H. Li. Rotation averaging. *IJCV*, 103(3):267–305, 2013.