16.485: Visual Navigation for Autonomous Vehicles (VNAV) **Fall 2021**

Lecture 15: RANSAC and 3D-3D Correspondences

Lecturer: Luca Carlone *Scribes: -*

This lecture covers:

- RANSAC (which we will use to estimate the relative pose from pixel correspondences, possibly including wrong correspondences),
- estimation of the relative pose from 3D-3D correspondences (e.g., the ones given by an RGB-D camera).

For both topics, we follow the presentation of the original papers (ref. [2] for RANSAC, and ref. [1] for pose estimation from 3D-3D correspondences).

15.1 Recap on Essential matrix Estimation

In the last lecture we saw how to estimate the essential matrix from 8 point correspondences. In particular, we observed that, in the absence of noise, (a vectorized version of) the essential matrix can be obtained by solving the linear system:

$$\boldsymbol{A}\boldsymbol{e} = 0 \tag{15.1}$$

where $\boldsymbol{e}$ is assumed to be nonzero (e.g., $\|\boldsymbol{e}\| = 1$) and $\boldsymbol{A}$ is a suitable 8×9 matrix computed from the pixel correspondences. By solving the linear system (15.1) and re-arranging the entries of $\boldsymbol{e}$ into a 3×3 matrix, we obtain the desired essential matrix $\boldsymbol{E}$. Note that since $\boldsymbol{A}\boldsymbol{e} = -\boldsymbol{A}\boldsymbol{e} = 0$ both $\boldsymbol{E}$ and $-\boldsymbol{E}$ are valid solutions to the linear system (15.1), so we need to consider both as potential essential matrices (we discussed how to resolve this ambiguity in the previous lecture).

Noisy pixel measurements. Since the pixels measurements are typically affected by noise, the solution of the linear system may not be an essential matrix. Therefore, it is common to project the solution onto the essential space using eq. (13.15) in Lecture 13.

More than 8 points. If we have more than $N > 8$ point correspondences and in absence of noise, we can still expect to compute the right essential matrix via (15.1) (in general, $\boldsymbol{A} \in \mathbb{R}^{N\times 9}$). However, in presence of noise, the linear equations will not be satisfied exactly and we might look for a solution that is as close as possible to satisfying the linear equalities:

$$\min_{\boldsymbol{E}\in\mathcal{S}_E} \|\boldsymbol{A} \ \ \text{vec}(\boldsymbol{E})\|^2 \tag{15.2}$$

where we remarked that the minimization has to be restricted to valid essential matrices and vec($\boldsymbol{E}$) denotes a vector of 9 elements stacking the entries of $\boldsymbol{E}$. We can follow the same logic of the 8-point algorithm and approximate the solution of (15.2) by the following steps:

- solve the relaxed problem: $\arg\min_{\|\boldsymbol{e}\|=1} \|\boldsymbol{A}\boldsymbol{e}\|^2$ (whose solution can be computed in closed form: it is simply the eigenvector corresponding to the smallest eigenvalue of $\boldsymbol{A}^\mathsf{T}\boldsymbol{A}$, see https://en.wikipedia.org/wiki/Rayleigh_quotient.)

- project the result to the Essential space using eq. (13.15) in Lecture 13.

What if we have wrong correspondences? If some of the point correspondences are wrong, the minimization (15.2) will not produce a meaningful result. Therefore, we introduce RANSAC, a method that will allow computing the essential matrix even when a subset of the correspondences are wrong.

15.2 RANSAC

From the previous section:

- we know how to compute the essential matrix from 8 "good" point correspondences.
- issue: we are given $N >> 8$ point correspondences and some of them may be wrong.

RANSAC (**Ran**dom **Sa**mple **C**onsensus) is an approach to perform estimation (in our case: estimate the essential matrix) in presence of outliers (in our case: wrong pixel correspondences).

The basic idea behind RANSAC is straightforward: say that in general we can compute a model from n points (in the case of the 8-point method, clearly $n = 8$). Then the idea is to sample a subset of n elements from the N available correspondences till we get a "good" set. What is a "good" set? A set for which the corresponding model (i.e., the essential matrix) explains as many remaining correspondences as possible.

Example: Essential matrix estimation RANSAC proceeds as follows, when applied to the estimation of the Essential matrix. Given (i) N point correspondences, possibly including wrong matches, (ii) a minimal solver that can estimate the Essential matrix from n points (e.g., $n = 8$), RANSAC computes an estimate of $\boldsymbol{E}$ as follows:

1. Randomly select a subset $\mathcal{S}$ of n point correspondences out of the N available correspondences.
2. Estimate the Essential matrix $\boldsymbol{E}$ from the correspondences in $\mathcal{S}$
3. Compute the set $\mathcal{S}^\star$ of correspondences (i, j) that are such that:
$$\boldsymbol{y}_j^\top \boldsymbol{E} \boldsymbol{y}_i \approx 0 \tag{15.3}$$
   or more formally $\|\boldsymbol{y}_j^\top \boldsymbol{E} \boldsymbol{y}_i\| < \epsilon$ for some small threshold ϵ. The set $\mathcal{S}^\star$ is called the *consensus set* of $\mathcal{S}$.
4. Case (a): if the cardinality of $\mathcal{S}^\star$ (number of elements of $\mathcal{S}^\star$, denoted as $|\mathcal{S}^\star|$) is larger than a given threshold T, accept $\mathcal{S}^\star$ as the set of inliers and use it to recompute a better estimate of the essential matrix $\boldsymbol{E}$ via (15.2).

   Case (b): if $|\mathcal{S}^\star| < T$ repeat from Step 1. Stop after a maximum number of iterations $k_{\max}$.

General algorithm We can generalize the example above as follows. Given (i) N data points, possibly corrupted by outliers, and (ii) a minimal solver that can estimate the unknown model $\mathcal{P}$ from from n data points, RANSAC proceeds as follows:

1. Randomly select a subset $\mathcal{S}$ of n data points out of the N available points.
2. Estimate the model $\mathcal{P}$ from the data points in $\mathcal{S}$

3. Compute the set $\mathcal{S}^\star$ of correspondences (i, j) that are in agreement with the model $\mathcal{P}$ up to some given tolerance ϵ.
4. Case (a): if $|\mathcal{S}^\star| \geq T$, accept $\mathcal{S}^\star$ as the set of inliers and use it to recompute a better estimate of the model $\mathcal{P}$.

   Case (b): if $|\mathcal{S}^\star| < T$ repeat from Step 1. Stop after a maximum number of iterations $k_{\max}$.

15.2.1 Parameter tuning

15.2.1.1 Error Tolerance ϵ

How to set the tolerance ϵ that is used to decided whether a given correspondence belongs to the consensus set? Let us consider our Essential matrix estimation example, where the condition to satisfy is:

$$\|\boldsymbol{y}_j^\top \boldsymbol{E} \boldsymbol{y}_i\| < \epsilon \tag{15.4}$$

In this case one may select ϵ as a function of the pixel noise, but that would not account for the fact that the matrix $\boldsymbol{E}$ estimated from the 8 points in the set $\mathcal{S}$ is also noisy. In some cases one may do some error propagation, but it is more common to set this tolerance empirically by trial and error.

15.2.1.2 Maximum number of iterations $k_{\max}$

Assume we know the probability w that a given correspondence is an inlier[1] Moreover, call k the number of trials required to select an outlier-free set (note: k is a random variable since we sample at random). Then the expected value of k is:

$$\mathbb{E}[k] = \sum_{i=1}^{\infty} i \;\; \mathbb{P}(k = i) = b + 2ab + 3a^2 b + ... = b[1 + 2a + 3a^2 + ...] \tag{15.5}$$

where $b = w^n$ and $a = (1 - b)$. Intuitively: $\mathbb{P}(k = 1)$ is the probability of extracting all n "good points" at the first iteration, hence $\mathbb{P}(k = 1) = w^n$; $\mathbb{P}(k = 2)$ is the probability of **not** extracting all good points at the first iteration (probability: $1 - w^n$) and extracting all good points at the second (probability: w^n), and so on. The infinite series in the expression above is equal to:

$$1 + 2a + 3a^2 + ... = \frac{1}{(1-a)^2} \tag{15.6}$$

Recalling that $a = (1 - b)$:

$$\mathbb{E}[k] = b\frac{1}{(1-a)^2} = b\frac{1}{b^2} = \frac{1}{b} = w^{-n} \tag{15.7}$$

Typically, one would set the maximum number of iterations $k_{\max}$ equal to $2 - 3$ times the expected value above. Ref. [2] provides a nice justification for that in terms of the variance of k.

15.2.1.3 Admissible size of an acceptable consensus set T

Calling q the probability that a given data point is within the tolerance of a wrong model, then q^{T-n} is the probability that $T - n$ points agree with the incorrect model. Assuming $q < 0.5$ (i.e., it's somehow unlikely

[1] More precisely, this is the probability that a correspondence falls in the consensus set built from a set of inliers.

that a point agrees with an incorrect model by chance), then for $T = n + 5$, it holds $q^{T-n} < 0.5^5 < 0.03$, i.e., it is unlikely to have a consensus set larger than $T = n + 5$ for an incorrect model.

Again, since these theoretical bounds are based on (often unrealistic) simplifying assumptions, it is more common to set T to a larger number, e.g., $T > N/2$.

15.3 Relative Pose Estimation from 3D-3D Correspondences

Assume that we are given N (calibrated) pixel correspondences $(\tilde{\boldsymbol{y}}_{1,k}, \tilde{\boldsymbol{y}}_{2,k})$ for $k = 1, \ldots, N$ picturing corresponding points in 2 cameras. Moreover, assume that we have an RGB-D camera which is also able to measure the depth at each pixel. How can we estimate the relative pose between the cameras?

Since we have the depth $\lambda_{1,k}$ for each pixel $\tilde{\boldsymbol{y}}_{1,k}$, we can reproject the corresponding 3D point as $\boldsymbol{p}_{1,k} = \lambda_{1,k}\tilde{\boldsymbol{y}}_{1,k}$ (and similarly for the points in the second camera). Therefore the problem can be understood as the estimation of the relative pose between 2 sets of 3D points observed by 2 cameras, given correspondences $(\boldsymbol{p}_{1,k}, \boldsymbol{p}_{2,k})$, for $k = 1, \ldots, N$.

In the absence of noise, each point k must satisfy:

$$\boldsymbol{p}_{1,k} = \boldsymbol{R}^{\mathrm{c}_1}_{\mathrm{c}_2}\ \boldsymbol{p}_{2,k} + \boldsymbol{t}^{\mathrm{c}_1}_{\mathrm{c}_2} \qquad\qquad k = 1, \ldots, N \tag{15.8}$$

i.e., the points seen by the two cameras only differ by a rigid-body transformation $(\boldsymbol{R}^{\mathrm{c}_1}_{\mathrm{c}_2}, \boldsymbol{t}^{\mathrm{c}_1}_{\mathrm{c}_2})$ corresponding to the relative pose between the cameras. For simplicity of notation, in the following we simply use $\boldsymbol{R}$ (instead of $\boldsymbol{R}^{\mathrm{c}_1}_{\mathrm{c}_2}$) and $\boldsymbol{t}$ (instead of $\boldsymbol{t}^{\mathrm{c}_1}_{\mathrm{c}_2}$) since this is the only pose of interest in the rest of these notes. Therefore, in absence of noise we can compute $\boldsymbol{R}$ and $\boldsymbol{t}$ by solving the set of linear equations (15.8).

In presence of noise, there is no pose $(\boldsymbol{R}, \boldsymbol{t})$ that exactly satisfies all the equations in (15.8) and we can only look for a pose that makes $\boldsymbol{p}_{1,k} - (\boldsymbol{R}\boldsymbol{p}_{2,k} + \boldsymbol{t})$ close to zero. This suggests estimating the pose $(\boldsymbol{R}, \boldsymbol{t})$ by solving the following minimization problem:

$$\min_{\substack{\boldsymbol{R}\in\mathrm{SO}(3)\\ \boldsymbol{t}\in\mathbb{R}^3}} \sum_{k=1}^{N} \|\boldsymbol{p}_{1,k} - (\boldsymbol{R}\boldsymbol{p}_{2,k} + \boldsymbol{t})\|^2 = \sum_{k=1}^{N} \|\boldsymbol{p}_{1,k} - \boldsymbol{R}\boldsymbol{p}_{2,k} - \boldsymbol{t}\|^2 \tag{15.9}$$

For any choice of $\boldsymbol{R}$, the minimum over $\boldsymbol{t}$ is attained at

$$\boldsymbol{t}^\star = \frac{1}{N}\sum_{k=1}^{N}(\boldsymbol{p}_{1,k} - \boldsymbol{R}\boldsymbol{p}_{2,k}) \tag{15.10}$$

[hint: from "the point of view" of $\boldsymbol{t}$, problem (15.9) is the same as $\min_{\boldsymbol{t}} \sum_{k=1}^{N} \|\boldsymbol{t}_k - \boldsymbol{t}\|^2$, whose solution is the average: $\boldsymbol{t}^\star = \frac{1}{N}\sum_{k=1}^{N}\boldsymbol{t}_k$].

Substituting $\boldsymbol{t}^\star$ back into (15.9) we get:

$$\min_{\boldsymbol{R}\in\mathrm{SO}(3)} \sum_{k=1}^{N} \left\| \boldsymbol{p}_{1,k} - \boldsymbol{R}\boldsymbol{p}_{2,k} - \frac{1}{N}\sum_{k=1}^{N}(\boldsymbol{p}_{1,k} - \boldsymbol{R}\boldsymbol{p}_{2,k}) \right\|^2 = \tag{15.11}$$

$$\min_{\boldsymbol{R}\in\mathrm{SO}(3)} \sum_{k=1}^{N} \left\| \left(\boldsymbol{p}_{1,k} - \frac{1}{N}\sum_{k=1}^{N}\boldsymbol{p}_{1,k}\right) - \boldsymbol{R}\left(\boldsymbol{p}_{2,k} - \frac{1}{N}\sum_{k=1}^{N}\boldsymbol{p}_{2,k}\right) \right\|^2 \tag{15.12}$$

Redefining $\boldsymbol{p}'_{1,k} = \boldsymbol{p}_{1,k} - \frac{1}{N}\sum_{k=1}^{N}\boldsymbol{p}_{1,k}$ and $\boldsymbol{p}'_{2,k} = \boldsymbol{p}_{2,k} - \frac{1}{N}\sum_{k=1}^{N}\boldsymbol{p}_{2,k}$:

$$\min_{\boldsymbol{R}\in\mathrm{SO}(3)} \sum_{k=1}^{N} \left\|\boldsymbol{p}'_{1,k} - \boldsymbol{R}\boldsymbol{p}'_{2,k}\right\|^2 = \min_{\boldsymbol{R}\in\mathrm{SO}(3)} \sum_{k=1}^{N} \|\boldsymbol{p}'_{1,k}\|^2 + \|\boldsymbol{R}\boldsymbol{p}'_{2,k}\|^2 - 2(\boldsymbol{p}'_{1,k})^\mathsf{T}\boldsymbol{R}\boldsymbol{p}'_{2,k} = \tag{15.13}$$

$$\min_{\boldsymbol{R}\in\mathrm{SO}(3)} \sum_{k=1}^{N} -\mathrm{tr}\left(\boldsymbol{p}'_{2,k}(\boldsymbol{p}'_{1,k})^\mathsf{T}\boldsymbol{R}\right) = \min_{\boldsymbol{R}\in\mathrm{SO}(3)} -\mathrm{tr}\left(\boldsymbol{M}^\mathsf{T}\boldsymbol{R}\right) = \max_{\boldsymbol{R}\in\mathrm{SO}(3)} \mathrm{tr}\left(\boldsymbol{M}^\mathsf{T}\boldsymbol{R}\right) \tag{15.14}$$

where $\boldsymbol{M} = \sum_{k=1}^{N}(\boldsymbol{p}'_{2,k}(\boldsymbol{p}'_{1,k})^\mathsf{T})^\mathsf{T}$. Now we note that:

$$\arg\min_{\boldsymbol{R}\in\mathrm{SO}(3)} \|\boldsymbol{R}-\boldsymbol{M}\|_F^2 = \arg\min_{\boldsymbol{R}\in\mathrm{SO}(3)} \mathrm{tr}\left((\boldsymbol{R}-\boldsymbol{M})(\boldsymbol{R}-\boldsymbol{M})^\mathsf{T}\right) \tag{15.15}$$

(recall $\boldsymbol{R}^\mathsf{T}\boldsymbol{R} = \mathbf{I}$ and that constants are irrelevant for the optimization)

$$= \arg\min_{\boldsymbol{R}\in\mathrm{SO}(3)} \mathrm{tr}\left(\mathbf{I} - 2\boldsymbol{M}^\mathsf{T}\boldsymbol{R} + \boldsymbol{M}\boldsymbol{M}^\mathsf{T}\right) = \arg\max_{\boldsymbol{R}\in\mathrm{SO}(3)} \mathrm{tr}\left(\boldsymbol{M}^\mathsf{T}\boldsymbol{R}\right) \tag{15.16}$$

therefore maximizing $\mathrm{tr}\left(\boldsymbol{M}^\mathsf{T}\boldsymbol{R}\right)$ in eq. (15.14) is the same as solving $\arg\min_{\boldsymbol{R}\in\mathrm{SO}(3)} \|\boldsymbol{R}-\boldsymbol{M}\|_F^2$, which is the definition of the *projection* of a matrix $\boldsymbol{M}$ onto the set SO(3). This projection has a closed-form solution (if you are interested, you can find the closed-form solution in the Appendix below).

The solution to relative pose estimation from 3D-3D correspondences outlined in this section is often called "Arun's method" and was presented in [1]. One can also repeat a similar derivation by representing rotations via unit quaternions, and get the so called "Horn's method", which is described in the beautiful paper [3].

15.3.1 Appendix: Projection onto $\mathrm{O}(n)$ and $\mathrm{SO}(n)$

Given an arbitrary square matrix $\boldsymbol{M} \in \mathbb{R}^{n\times n}$, projecting $\boldsymbol{M}$ onto $\mathrm{O}(n)$ (resp. $\mathrm{SO}(n)$) consists in finding the matrix $\boldsymbol{Q} \in \mathrm{O}(n)$ (resp. $\boldsymbol{R} \in \mathrm{SO}(n)$) that is *nearest* to $\boldsymbol{M}$:

$$\Pi_{\mathrm{O}(n)}(\boldsymbol{M}) = \arg\min_{\boldsymbol{Q}\in\mathrm{O}(n)} \|\boldsymbol{Q}-\boldsymbol{M}\|_F^2\,; \quad \Pi_{\mathrm{SO}(n)}(\boldsymbol{M}) = \arg\min_{\boldsymbol{R}\in\mathrm{SO}(n)} \|\boldsymbol{R}-\boldsymbol{M}\|_F^2\,. \tag{15.17}$$

This is a notable example of tractable nonconvex problems. In fact, the two projection problems in eq. (15.17) admit closed-form solutions, using *singular value decomposition* (SVD) [1,4]. Formally, let $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\mathsf{T}$ be the SVD of $\boldsymbol{M}$, where $\boldsymbol{U},\boldsymbol{V} \in \mathrm{O}(n)$, and $\boldsymbol{S}$ contains the singular values of $\boldsymbol{M}$ in *descending* order,[2] i.e., $S_{11} \geq S_{22} \geq \cdots \geq S_{nn} \geq 0$, then the projections in (15.17) admit the following closed-form solutions:

$$\Pi_{\mathrm{O}(n)}(\boldsymbol{M}) = \boldsymbol{U}\boldsymbol{V}^\mathsf{T}, \tag{15.18}$$

$$\Pi_{\mathrm{SO}(n)}(\boldsymbol{M}) = \boldsymbol{U}\mathrm{diag}\left([1,1,\ldots,\det(\boldsymbol{U})\det(\boldsymbol{V})]\right)\boldsymbol{V}^\mathsf{T}. \tag{15.19}$$

Proof. We only prove the solution for projection onto $\mathrm{O}(n)$ (eq. (15.18)), while we leave the proof for projection onto $\mathrm{SO}(n)$ as an exercise. To prove eq. (15.18) is the solution for problem (15.17), we develop the cost function of problem (15.17):

$$\arg\min_{\boldsymbol{Q}\in\mathrm{O}(n)} \|\boldsymbol{Q}-\boldsymbol{M}\|_F^2 = \arg\min_{\boldsymbol{Q}\in\mathrm{O}(n)} \mathrm{tr}\left((\boldsymbol{Q}-\boldsymbol{M})(\boldsymbol{Q}-\boldsymbol{M})^\mathsf{T}\right) \tag{15.20}$$

$$= \arg\min_{\boldsymbol{Q}\in\mathrm{O}(n)} \mathrm{tr}\left(\boldsymbol{Q}\boldsymbol{Q}^\mathsf{T}\right) + \mathrm{tr}\left(\boldsymbol{M}\boldsymbol{M}^\mathsf{T}\right) - 2\mathrm{tr}\left(\boldsymbol{Q}^\mathsf{T}\boldsymbol{M}\right) = \arg\max_{\boldsymbol{Q}\in\mathrm{O}(n)} \mathrm{tr}\left(\boldsymbol{Q}^\mathsf{T}\boldsymbol{M}\right) \tag{15.21}$$

$$= \arg\max_{\boldsymbol{Q}\in\mathrm{O}(n)} \mathrm{tr}\left(\boldsymbol{Q}^\mathsf{T}\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\mathsf{T}\right) = \arg\max_{\boldsymbol{Q}\in\mathrm{O}(n)} \mathrm{tr}\left(\boldsymbol{S}\boldsymbol{V}^\mathsf{T}\boldsymbol{Q}^\mathsf{T}\boldsymbol{U}\right), \tag{15.22}$$

[2]For example, Matlab `svd` returns the singular values in descending order.

and observe that $\boldsymbol{V}^\top\boldsymbol{Q}^\top\boldsymbol{U}$ is an orthogonal matrix because $\boldsymbol{V}, \boldsymbol{Q}, \boldsymbol{U} \in \mathrm{O}(n)$. Therefore, as $\boldsymbol{S}$ is an diagonal matrix with positive entries, the maximum of $\mathrm{tr}\left(\boldsymbol{S}\boldsymbol{V}^\top\boldsymbol{Q}^\top\boldsymbol{U}\right)$ is attained if and only if $\boldsymbol{V}^\top\boldsymbol{Q}^\top\boldsymbol{U} = \mathbf{I}_n$, i.e., $\boldsymbol{Q} = \boldsymbol{U}\boldsymbol{V}^\top$. □

References

[1] K.S. Arun, T.S. Huang, and S.D. Blostein. Least-squares fitting of two 3-D point sets. *IEEE Trans. Pattern Anal. Machine Intell.*, 9(5):698–700, sept. 1987.

[2] R. Bolles and M. Fischler. A RANSAC-based approach to model fitting and its application to finding cylinders in range data. In *Intl. Joint Conf. on AI (IJCAI)*, pages 637–643, Vancouver, BC, Canada, 1981.

[3] Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Amer.*, 4(4):629–642, Apr 1987.

[4] P.H. Schonemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31:1–10, 1966.