1. Projection matrix to intrinsic and extrinsic parameters.

Answer the following, providing citations to any sources you consult:

a. What is the $RQ$ factorization (also known as the $RQ$ decomposition)\(^1\) of a matrix $A$, and how can it help us recover the intrinsic and extrinsic parameters of a camera calibration matrix $M = [ \begin{bmatrix} A & b \end{bmatrix}$?

b. Why is the standard notation for the $RQ$ factorization especially horrible in this context?

2. Representing 3D rotations.

Although we can represent every rotation in 3D as a $3 \times 3$ matrix, 3D rotations only have three degrees of freedom. There are therefore several alternative parameterizations of 3D rotations, aside from the matrix representation.

- **Euler angles** encode rotations as a triplet of angles to rotate around a sequence of known axes. For instance, the roll-pitch-yaw convention rotates first around the $x$ axis (roll), next about the $y$ axis (pitch), and finally about the $z$ axis (yaw).

- **Unit quaternions** are an extension of the concept of complex numbers into higher dimensions. Just as a single complex number $(a + bi)$ with unit magnitude can represent a rotation in the plane, a unit quaternion $(a + bi + cj + dk)$ can represent rotations in space, given the appropriate multiplicative identities defined on $i$, $j$, and $k$.

- **Rotation vectors** are a compact way to encode rotations based on the fact that any 3D rotation can be represented as a finite rotation of some angle $\alpha$ about a particular axis $a$, with $\|a\| = 1$. The corresponding $3 \times 1$ rotation vector $r$ is then simply given by $r = \alpha a$.

\(^1\)Note: the $RQ$ factorization is related to – but distinct from – the $QR$ factorization.
Read section 2.1.4 of the book, and flip through up to about page 22 of the Diebel paper on the course website. Then, answer the following questions:

a. For each of the four parameterizations mentioned above (matrices, Euler angles, unit quaternions, rotation vectors), explain what the constraints on the representation are. A constraint is any property that prevents some sets of numbers from representing valid rotations for a particular parameterization.

To get you started: even though a rotation matrix has nine elements, it is not the case that any nine numbers form a valid rotation matrix. What must be true about a matrix $R$ in order for it to be a valid rotation matrix?

b. In which parameterizations can we easily compute compositions of rotations – the rotation resulting from applying two arbitrary rotations in sequence?

c. What is gimbal lock, and why is it undesirable?