1. Given an \( m \times n \) matrix \( M \) with entries \( m_{ij} \), the Frobenius norm of \( M \) is defined as follows:

\[
\|M\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} m_{ij}^2}.
\]

For any \( n \times n \) matrix \( C \) with entries \( c_{ij} \) the quantity

\[
\text{tr}(C) = \sum_{i=1}^{n} c_{ii}
\]

is called the trace of \( C \). It is easy to see that for any two square matrices \( A \) and \( B \) the following properties hold:

\[
\text{tr}(AB) = \text{tr}(BA) \quad \text{and} \quad \text{tr}(A + B) = \text{tr}(A) + \text{tr}(B).
\]

This problem takes you through the steps of proving the following statement:

Let \( A \) be a \( 3 \times 3 \) matrix. The orthogonal \( 3 \times 3 \) matrix \( Q \) that minimizes \( \|A - Q\|_F \) is

\[
Q = UV^T
\]

where

\[
A = U\Sigma V^T
\]

is the Singular Value Decomposition (SVD) of \( A \).

While the result above holds for any square matrix with 3 replaced by \( n \), it is ok to limit your answers to the case \( n = 3 \), unless specified otherwise.

(a) Show that for any \( m \times n \) matrix \( M \) the following property holds:

\[
\text{tr}(M^T M) = \|M\|_F^2.
\]

(b) Prove that minimizing \( \|A - Q\|_F \), where \( A \) and \( Q \) are \( 3 \times 3 \) and \( Q \) is orthogonal, is the same as maximizing \( \text{tr}(Q^T A) \).

(c) Use the SVD of \( A \) and properties of the trace advertised earlier to find an orthogonal matrix \( Z \) such that

\[
\text{tr}(Q^T A) = \text{tr}(Z\Sigma).
\]

(d) Spell out \( \text{tr}(Z\Sigma) \) in terms of the entries \( z_{ij} \) of \( Z \) and the singular values \( \sigma_i \) in \( \Sigma \).

(e) Use the expression for \( \text{tr}(Z\Sigma) \) you just derived and what you know about the entries of an orthogonal matrix to show that

\[
\text{tr}(Z\Sigma) \leq \sigma_1 + \sigma_2 + \sigma_3.
\]

Also show that equality is reached when \( Q = UV^T \). [This expression for \( Q \) therefore yields the maximum value of \( \text{tr}(Q^T A) \), that is, the minimum value of \( \|A - Q\|_F \).]
2. A consumer camera has a fixed, thin lens with a focal length of 10 millimeters, and a sensor with 1080 rows and 1920 columns (this is nominal HD resolution). The sensor is 8 millimeters wide and 4.5 millimeters tall. A calibration object is placed at about 1 meter in front of the camera, and the camera is focused at exactly 1 meter. Intrinsic calibration of the camera reveals that lens distortion is negligible, the sensor has no skew, and the principal point is at row 590 and column 925.

In this problem, you will assemble the matrices in the camera model described in Section 3.3.2 of the textbook, and use those matrices and data provided with this assignment to do extrinsic calibration with the procedure described in Section 6.5.2. All notation refers to those sections unless otherwise specified. Report all your numerical values rounded to four significant digits, but use Matlab’s default double floating point numerical resolution for all your calculations. Do not use decimal-exponent notation. That is, say $4.32 \times 10^3$ rather than $4.32 \times 10^3$.

Give units of measure for all quantities.

(a) Give the numerical values of the matrices $K_s$, $K_f$, and $K$, where the focal distance (not focal length!) $f$ in $K_f$ is in millimeters. Show your reasoning.

(b) Are the pixels of this camera square?

(c) The camera described above was placed in front of a calibration box to produce the image shown in the figure below.\(^1\)

The 3D coordinates of the corners of the light-colored squares on the visible faces of the box were measured on the box itself, in a frame of reference whose origin is the one invisible corner of the box, and the axes are parallel to the edges of the box. The corresponding 2D image coordinates were measured in the image. The file data.mat on the homework web page contains this data. Specifically, the command

```
load data
```

will place two matrices $P$ and $p$ in your MATLAB workspace. The matrix $P$ has in each of its columns the $X,Y,Z$ coordinates of one of the corners, in millimeters. The matrix $p$ has in each of its columns the $x',y'$ coordinates of one of the corners, in pixels. Columns in $P$ correspond to columns in $p$.

Write a MATLAB function

```
Pi = projectionMatrix(P, p)
```

that finds the projection matrix $\Pi$ by solving the problem (6.54) in the textbook. Normalize $\Pi$ so that its last entry $\pi_{34}$ is equal to 1. Show your code and the resulting matrix.

(d) Section 6.5.2 of the textbook assumes that the intrinsic calibration matrix $K$ is unknown. However, we know $K$ from the intrinsic calibration stage, so the text after equation (6.54) is irrelevant to this exercise.

Use the calibration matrix and properties of orthogonal matrices to find the rotation matrix $R$ and translation vector $t$ (in the book, the translation vector is called $T$). Show your reasoning, your calculations, and their results. Make sure you specify units of measure where appropriate. [Hint: For an $n \times n$ matrix $A$, we have $\det(cA) = c^n \det(A)$.]

\(^1\)Everything here is a simulation, and the image in the figure was produced synthetically.
(e) Let \( i, j, k \) be unit vectors along the camera’s coordinate axes. Specifically, \( i \) is parallel to the pixel rows of the sensor pointing to the right (when looking at the camera from behind it), \( j \) to the pixel columns pointing down, and \( k \) points toward the scene. What are \( i, j, k \) in terms of \( R \)?

(f) Fill the gaps in the following sentence (the length of the gaps is irrelevant):

The vector \( t \) found above points from _____ to _____, and its entries are expressed in _____ coordinates.

(g) Find the coordinates of the center of projection of the camera in the reference frame in which the 3D coordinates of the squares on the box were measured. Show your calculation.

(h) Is the matrix \( R \) you found orthogonal to within four significant digits? That is, are the entries of \( R^T R \) within four decimal digits of \( I_3 \)?

(i) The data for this problem are synthetic and essentially noise-free, so things work out fine. With noisy data, the matrix \( R \) resulting from the calculations above is unlikely to be orthogonal. How can you use the result you proved in problem 1 to find an orthogonal matrix close to \( R \)? Explain briefly.