# Table of Contents

Task 1 .......... 2
Computation of calibration matrix (k1 & k2): .......... 2
Computation of essential matrix: .......... 2
Resolving fourfold ambiguity: .......... 2
Computing & Plotting Epipolar line: .......... 3
Task 2: .......... 3
Computing geometric error: .......... 3
Indirect optimization: .......... 4
Calculation of Optimized Geometric error: .......... 4

# Task 1

## Computation of calibration matrix (k1 & k2):

For this task we need to read the data points from the bh.dat file. We need to first find out the fundamental matrix. Then we find the projection matrix using the fundamental matrix. We assume the left calibration matrix as identity matrix (k1). We get the calibration matrix k2 by performing RQ Decomposition on the projection matrix.
Value that we get for fundamental matrix, k1 and k2 are as follows:

**Fundamental matrix** =

$$\begin{matrix} 0.0000 & 0.0000 & 0.0014 \\ 0.0000 & -0.0000 & 0.0033 \\ -0.0023 & -0.0032 & 0.0800 \end{matrix}$$

**k1** =

$$\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$$

**k2** =

$$\begin{matrix} 0.3203 & -425.5584 & 142.5196 \\ 0 & 321.8556 & -137.4375 \\ 0 & 0 & 1.0000 \end{matrix}$$

## Computation of essential matrix:

In order to estimate the essential matrix, we apply the following formula:
*E = k1 * Fundamental Matrix * k2*
Value we get is as follows:

**Essential matrix** =

$$\begin{matrix} 0.0000 & -0.0004 & 0.0015 \\ 0.0000 & -0.0010 & 0.0037 \\ -0.0007 & -0.0462 & 0.1912 \end{matrix}$$

## Resolving fourfold ambiguity:

The essential matrix has four possible solutions for the relative camera pose from the essential matrix. These solutions are related by a rotation matrix and a translation vector. For resolving the fourfold ambiguity, we need to do the following steps in order to get a suitable rotation matrix and translation vector.

- Perform singular value decomposition on the essential matrix E

- Define the possible rotation and translation matrices
- Define the translation vector
- Define the projection matrices
- Triangulate the 3D points using each projection matrix
- Select the solution with positive z-coordinate

We get the following result:

**Rotation Matrix** =

| | | |
|---|---|---|
| 0.9092 | 0.0484 | 0.4135 |
| -0.3894 | -0.2528 | 0.8857 |
| -0.1474 | 0.9663 | 0.2110 |

**Translation Vector** =

| |
|---|
| -0.9220 |
| 0.3872 |
| -0.0003 |

## Computing & Plotting Epipolar line:

- First, we enforce singularity on the obtained essential matrix by the following:
  **E' =** $\boldsymbol{U\,diag\left(\frac{d1+d2}{2}, \frac{d1+d2}{2}, +0\right)V^T}$
- In order to find out the epipolar line we use the following formula:
  $\boldsymbol{l' = Fx}$
  $\boldsymbol{l = F^T x'}$

For plotting the epipolar line, I have used a blank white image and tried plotting the epipolar lines.
(P.S. : Please minimize the computed figures to view the epipolar lines)

# Task 2:

## Computing geometric error:

Using the singular essential matrix, we compute the value of samson distance

The value of error we get is:
**Geometric Error** = 3.8135e+05

## Indirect optimization:

We have used the built in MATLAB Levenberg-Marquart function to obtain the optimized singular essential matrix.
This function takes in a vector that is to be optimized for us it is the singular essential matrix. The algorithm uses a combination of steepest descent and Gauss-Newton methods to update the parameter. Additionally, it utilizes the Levenberg-Marquardt technique to adjust the step size in order to balance the convergence rate and the optimization stability.
Essential matrix we get after optimization is:

**Optimized Essential Matrix** =

| | | |
|---|---|---|
| -0.0059 | -0.0070 | -0.0080 |
| 0.0225 | -0.0037 | -0.0189 |
| -0.0006 | -0.0209 | 0.0961 |

Resource: *https://de.mathworks.com/help/optim/ug/lsqnonlin.html*

## Calculation of Optimized Geometric error:

In order to calculate the geometric error after optimization we enforce singularity to the optimized essential matrix that we get by using Levenberg-Marquart technique and then we find the geometric error. The error value that we get is:
**Geometric Error after optimization** = 2.0412e+04

As can be clearly seen from the result the error gets minimized after we apply Levenberg-Marquart optimization technique.