Report Computer Vision:

Calibration:

We use the camera calibration to estimate the position of the camera. To do that, we need a number of points and their coordinates (n>6). Using these points, we calculate the position of the camera in multiple steps

Data Normalisation:

Data Normalisation is used to reduce noise and improve overall accuracy. Let’s say we have points in our coordinates in a large scale system (high values), however the points are very close together. This could lead to larger errors if the data would not be normalized.

DLT:

For each 2D-3D correspondence we derive 3 equations, which can be simplified to two constraints because the linear system only has degree 2. We need at least 11 since we have 11 degrees of freedom to compute P which is achieved to have 6 or more points. To achieve this, we use the singular value decomposition and picking the smallest singular vector.

Optimizing reprojection errors

As stated in the exercise sheet with the DLT, we minimize the algebraic error. However we want to optimize for geometric error. The Algebraic error minimizes the distance between the known 3D location and the backprojected value. This increases the importance of points further away which is a problem with the algebraic error. However we prefer the geometric solution which minimizes the measured distance in the 2D observation.

Denormalizing the projection matrix

This can be done using the Projection matrices created in the data normalisation. We multiply the normalization / denormalization directly to the projection matrix.

Decomposing the projection matrix

Reprojection error before optimization: 0.0006316426059796243
Reprojection error after optimization: 0.0006253538899291337

```
K=
[[2.713e+03 3.313e+00 1.481e+03]
 [0.000e+00 2.710e+03 9.654e+02]
 [0.000e+00 0.000e+00 1.000e+00]]
R =
[[-0.774  0.633 -0.007]
 [ 0.309  0.369 -0.877]
 [-0.552 -0.681 -0.481]]
t = [[0.047 0.054 3.441]]
```

The decomposition of P lead to the upper triangular matrix K, the valid rotation matrix R. The error after the reprojection decreases a bit. In this specific example the algebraic error would have been enough since all of the points are quite close together. However if there were points further away, the influence of those would have changed the error more significantly.

Structure from Motion

What we do here basically is we have multiple pictures and points on these pictures. We know which points correspond to each other in the pictures. From there we try to find the exact position of the points and location of the camera.

Essential matrix estimation

To estimate the essential Matrix I used the correspondence of matching points in the different images. I created a constraint for each correspondence by finding out each factor of each element in the matrix is multiplied. We need 8 correspondences since that is the amount of degrees of freedom. Finding the best estimate is done similarly to the estimation of the projection matrix in the calibration task. We also make a SVD with the constraints and take the smallest singular vector.

Point triangulation

In the point triangulation, we find out the 3D position of the different points. Since we already have the K matrix this can be easily done.

Finding the correct decomposition

We have four possibilities that can be derived from the points. From these correspondences we choose the one with most points in front of it. This makes sense since the total number of points with corresponding values should be visible from camera perspective, as they are registered in the correspondence.

Map extension

Now that we have all the points in 3D, we can find out the camera pose with the code that we used in calibration and then add further points that were not visible before.