[CV21] Assignment 3 - Di Zhuang

Approach to calibration task

This section only covers a brief discussion, content related to the questions mentioned in the tasks are written in the next section.

2.2 (a) Data normalization

We normalize the 2D points and the 3D points so that all entries in the constraint matrix have similar magnitude, which ensures numerical stability in computation of the solution. In particular, x_normalized = Tx_original, X_normalized = UX_original.

2.2 (b) DLT

We apply the DLT algorithm to the given correspondences, which makes use of the SVD method, to obtain an estimate of the projection matrix P. This estimate minimizes the algebraic error because the SVD method minimizes norm(Ah) given norm(h) = 1.

2.2 © Optimizing reprojection error

Given the estimate of the DLT algorithm, we apply iterative optimization techniques to the estimate to minimize the geometric errors.

2.2 (d) Denormalizing the projection matrix

Derivation of the denormalized projection matrix is as follows:

$$\mathrm{T}x = \hat{P}UX$$

$$x = T^{-1}\hat{P}UX$$

$$x = (T^{-1}\hat{P}U)X$$

$$P = T^{-1}\hat{P}U$$

, where P_hat is the estimate we obtained from DLT and optimization, while P is the denormalized projection matrix.

2.2 (e) Decompose the projection matrix

Finally, we decompose P into K, R and t, adjust K and R so that K has a positive diagonal (because focal length is positive) and R has a determinant 1 (because a valid rotation matrix has to have a determinant of 1).

Answers to questions in calibration task

Given the camera models we saw in the lecture, the non linear part, such as the radial distortion effect, is not calibrated in our approach.

2.2 (a) Data normalization

The potential problem of skipping data normalization in the DLT algorithm is that some entries in the constraint matrix A might be very large, while some other entries might be very small. In that case, when we apply SVD method to solve for the h that minimizes norm(Ah) (given norm(h) = 1), the errors of entries in h that are multiplied with large entries in A will dominate the overall error that SVD is going to minimize, leading to the estimate of h being unstable.

2.2 (b) DLT

Two independent constraints can be derived from a single 2D-3D correspondence.

2.2 © Optimizing reprojecting errors

The reported error was reduced during optimization from 0.0006316426059796196 to 0.0006253538899291151.
The algebraic error is 0 if the constraint is fulfilled perfectly (i.e. exactly 5.5 constraints are enforced), otherwise it does not have any meaning. On the contrary, geometric errors have geometric meaning. Geometric errors arise from the inaccuracy of measurement of either the image points or the world points. In this subtask, we assume that the world points are known far more accurately than the image points, so we try to minimize the error

$$\sum_i ||\boldsymbol{x}_i - P\,\boldsymbol{X}_i||^2$$

The problem with the error measure e = x x PX is that its norm is the product of norm(x), norm(PX) and sin(theta). This makes it unstable because it is possible that although PX and x are very different with respect to directions, e.g. the angle between is 90 degrees, their norms are small. In that case, the norm of e is still small, but PX and x are not close enough.

2.2 (e) Decomposing the projection matrix

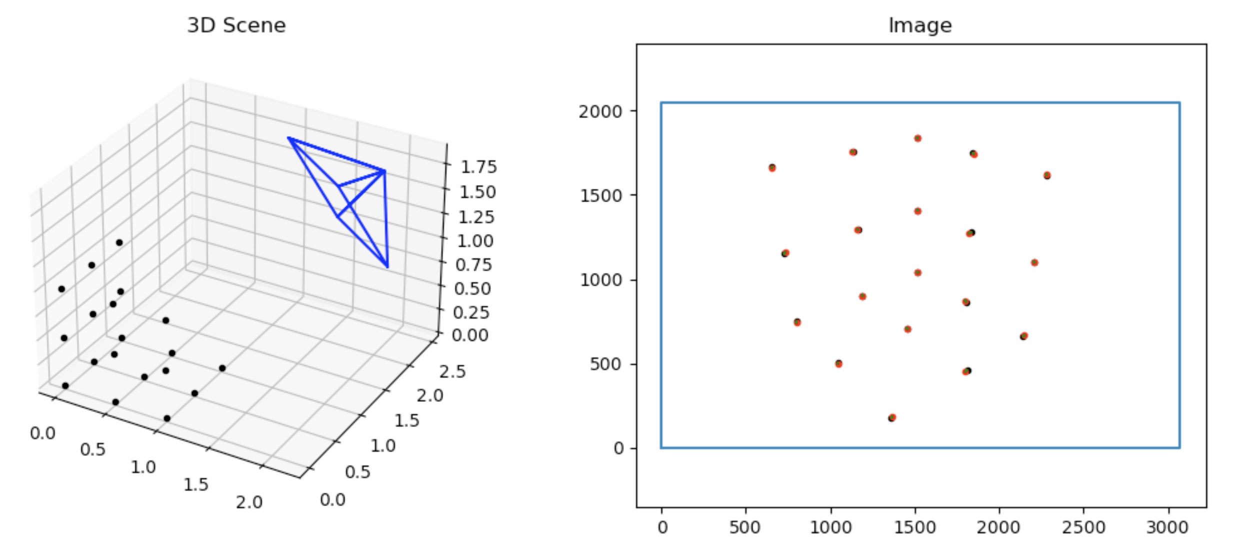
Values of K, R, t:

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 reported error was reduced during optimization from 0.0006316426059796196 to 0.0006253538899291151. This is not a big improvement, suggesting that the linear estimate is accurate enough. The estimated values seem reasonable since the visualization of the results looks nice:



Approach to sfm task

2.3 (a) Essential matrix estimation

First, given a set of correspondences between the two initial images, we use DLT to reformulate the constraints. Derivation of a row in the constraint matrix given correspondence p1 to p2 is as follows:

$$p_1 E p_2 = 0.$$

$$p_1 = (x_1, y_1, 1)$$

$$p_2 = (x_2, y_2, 1)$$

$$E = \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix}$$

$$Ep_2 = (e_{11}x_2 + e_{12}y_2 + e_{13},\ e_{21}x_2 + e_{22}y_2 + e_{23},\ e_{31}x_2 + e_{32}y_2 + e_{33})$$

$$p_1 E p_2 = e_{11}x_1x_2 + e_{12}x_1y_2 + e_{13}x_1 + e_{21}y_1x_2 + e_{22}y_1y_2 + e_{23}y_1 + e_{31}x_2 + e_{32}y_2 + e_{33}$$

$$= \underbrace{(x_1x_2,\ x_1y_2,\ x_1,\ y_1x_2,\ y_1y_2,\ y_1,\ x_2,\ y_2,\ 1)}_{\text{row in constraint matrix given correspondence } p_1 \Leftrightarrow p_2} \cdot e$$

where e is the vectorized $E = (e_{11}, e_{12}, e_{13}, e_{21}, e_{22}, e_{23}, e_{31}, e_{32}, e_{33})$.

By stacking all rows derived from all correspondences, we obtain a constraint matrix. We then apply the SVD method to solve for an estimate of e, the vectorized essential matrix. Reshaping e to a 3 by 3 matrix gives an estimate of E.

In order to fulfill the internal constraints of the essential matrix, we apply singular value decomposition to the above estimate of E, set the singular values to (1, 1, 0), and multiply the three matrices back with the new singular values. This gives us a valid estimate of E, namely E_hat.

2.3 (b) Point triangulation

Multiplying the homogeneous coordinates of the 3D points by projection matrices P1 and P2 respectively, we obtain the coordinates of the 3D points in the space of camera 1 and the space of camera 2. In particular, the third coordinate (z) of a point in the camera space indicates the depth of the point in that camera space. If this coordinate is negative, then the point falls behind the camera, which is wrong. These 3D points with negative (z) and their 2D correspondences are removed from points3D, im1_corrs and im2_corrs.

2.3 © Finding the correct decomposition

We first set the pose of image 2 to R = I and t = (0, 0, 0). Given the four possible relative poses, for each possible relative pose, we set the pose of image 1 to it, and triangulate the points. We choose the relative pose that gives the most points in front of both cameras.

2.3 (d) Absolute pose estimation

We now have a cloud of 3D points, and we are allowed to compute the 2D-3D correspondences between a new image and these 3D points. By calibration, we can compute an absolute pose for the camera of the new image.

2.3 (e) Map extension

Given the absolute pose for the camera of the new image and the correspondences between the new image and the registered images, we can triangulate the new image with respect to all registered images, which gives us new 3D points correspondences for each image. Repeat this process until all images are registered, we get a 3D reconstruction of the scene.