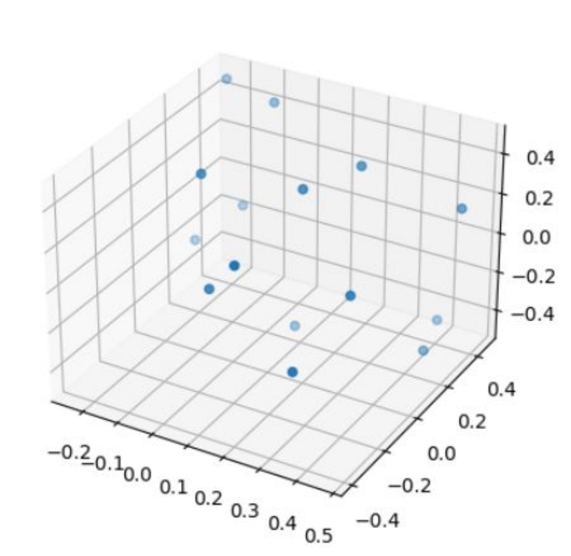
3D reconstruction using direct linear transformation

Inna Larina, Folu Obidare, Andrew Volodichev, Daria Demeshko, Dima Vinichenko

https://github.com/Alpgirl/DLT/tree/main

Problem description



We have some number of points in the space (3D).

But only six of them with known coordinates.

Also we have an image of all points (2D).

Based on this information we would like to define coordinates of all points.

This is Camera Calibration step of 3D reconstruction procedure

Existing solutions

- Monocular cues methods (using one or more images from one view point)
    - Shape-from-shading
    - Photometric stereo (different lightning conditions)
    - Machine learning based solutions
- Stereo vision (obtaining 3D model from multiple images)

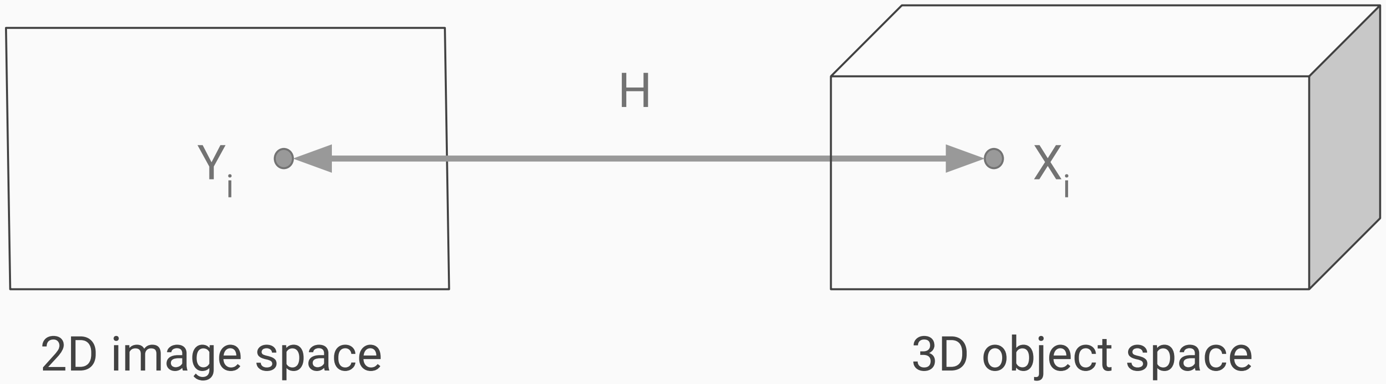
Description of the algorithm

- DLT Camera Calibration
- DLT Data Normalisation
- DLT Gold Standard Algorithm

Description of the algorithm

- DLT Camera Calibration
- DLT Data Normalisation
- DLT Gold Standard Algorithm

DLT Camera Calibration



$$Y_i = H\,X_i$$

DLT Algorithm: 2D-2D

$$x'_i = H x_i$$

$$(x'_i \times H x_i) = 0$$

We denote the j_{th} row of the matrix H by H_j^T:

$$Hx_i = \begin{pmatrix} h^{1T} x_i \\ h^{2T} x_i \\ h^{3T} x_i \end{pmatrix}$$

Denoting X'_i as $(x'_i, y'_i, w'_i)^T$ the cross-product may be given explicitly as:

$$x'_i \times Hx_i = \begin{pmatrix} y'_i h^{3T} x_i - w'_i h^{2T} x_i \\ w'_i h^{1T} x_i - x'_i h^{3T} x_i \\ x'_i h^{2T} x_i - y'_i h^{1T} x_i \end{pmatrix}$$

DLT Algorithm: 2D-2D

We can rewrite it as following:

$$\begin{bmatrix} 0^T & -w'_i x_i^T & y'_i x_i^T \\ w'_i x_i^T & 0^T & -x'_i x_i^T \\ -y'_i x_i^T & x'_i x_i^T & 0^T \end{bmatrix} \begin{pmatrix} h^1 \\ h^2 \\ h^3 \end{pmatrix} = 0$$

$A_i h = 0$, where A is a 3x9 matrix and h is a 9-vector made up of entries to the matrix H. These are only 2 independent equations:

$$\begin{bmatrix} 0^T & -w'_i x_i^T & -y'_i x_i^T \\ w'_i x_i^T & 0^T & -x'_i x_i^T \end{bmatrix} \begin{pmatrix} h^1 \\ h^2 \\ h^3 \end{pmatrix} = 0$$

Next, obtain the SVD of A and take the smallest singular value and corresponding singular vector.

DLT Algorithm: 3D-2D

In the 3D case for each correspondence X_i to x_i we derive the following equation:

$$\begin{bmatrix} 0^T & -w_i X_i^T & y_i X_i^T \\ w_i X_i^T & 0^T & -x_i X_i^T \end{bmatrix} \begin{pmatrix} p^1 \\ p^2 \\ p^3 \end{pmatrix} = 0$$

Description of the algorithm

- DLT Camera Calibration
- DLT Data Normalisation
- DLT Gold Standard Algorithm

DLT Data Normalization

1. Coordinates are translated so their centroid is located at the origin.
2. Coordinates are scaled so that the average distance to the origin is √3 and thus the average point is equal to (1,1,1,1).

The projection matrix for un-normalised coordinates is calculated as follows:

$$P = T^{-1} P' U$$

T - similarity transform which normalises the image space coordinates;
U - similarity transform to normalise the world space coordinates;
P’ - normalised projection matrix.

Description of the algorithm

- DLT Camera Calibration
- DLT Data Normalisation
- DLT Gold Standard Algorithm

Algorithm

1. **Linear Solution**: Compute an initial estimate of H.
   a. **Normalisation**.
   b. **DLT**.

2. **Minimise Geometric error**: Using the linear estimate as starting point minimise the geometric error:

$$\min_P \sum_i d(\widetilde{x}_i, \widetilde{P}\widetilde{X}_i)^2$$

   using an iterative algorithm such as Levenberg-Marquardt.

3. **Denormalisation**:

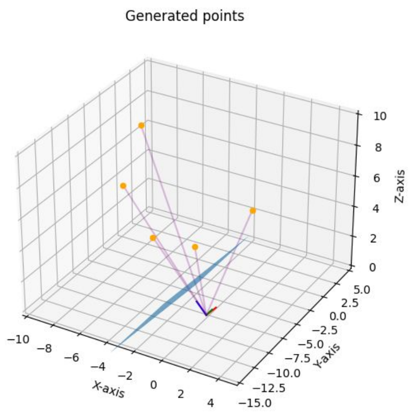
$$P = T^{-1} P' U$$

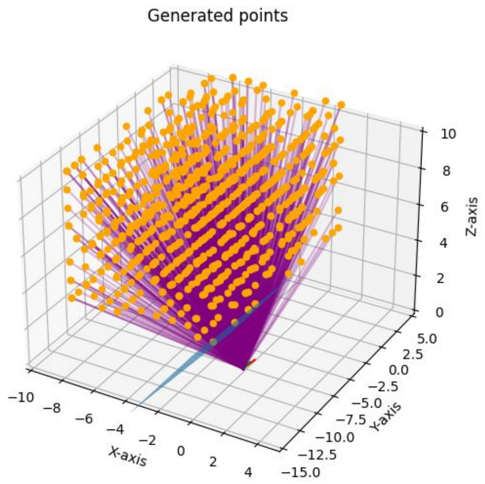
Results

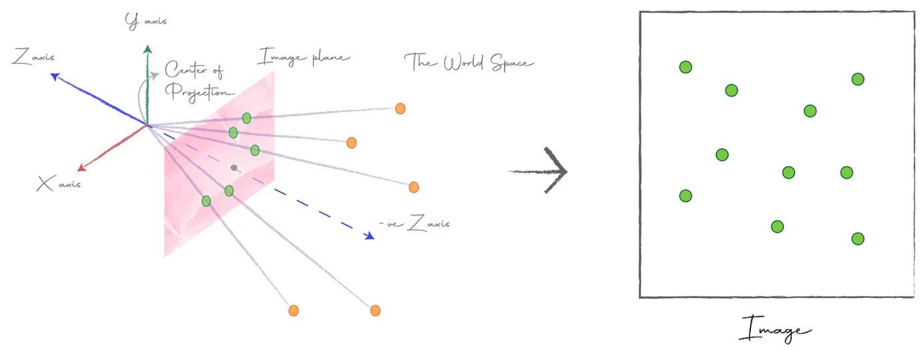
We generate random points in limited space. In our example
x from [-10;0]
y from [-10;10]
z from [f,10]
where f is focal length.

We determine extrinsic and intrinsic camera parameters.

As a result, we obtain 2D projections using camera properties and points' world coordinates.

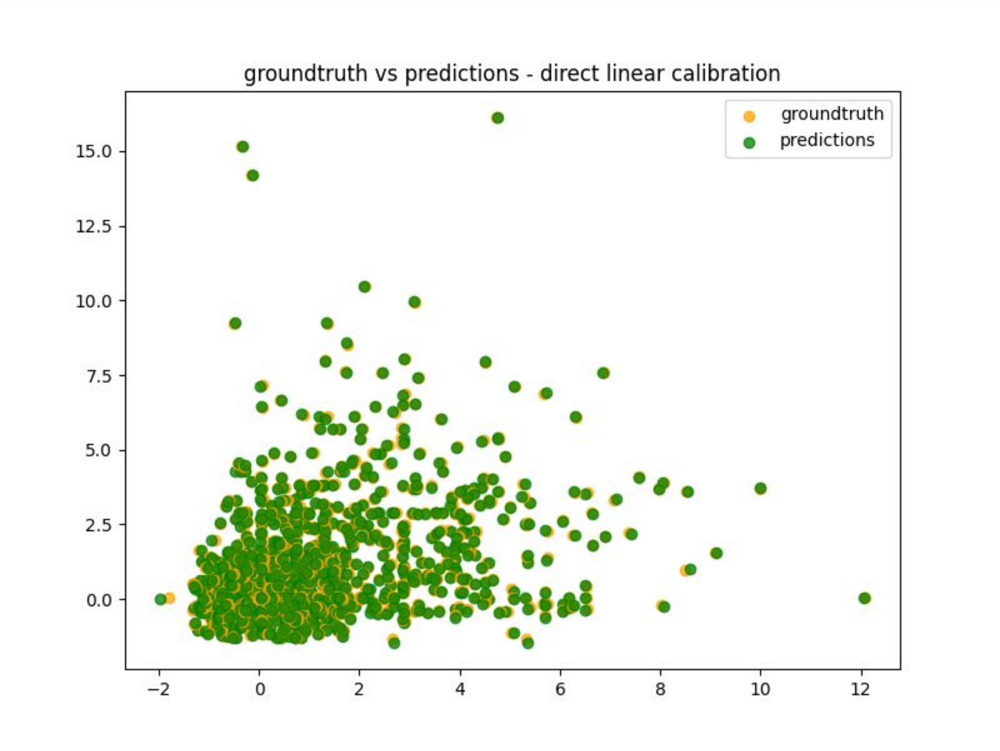




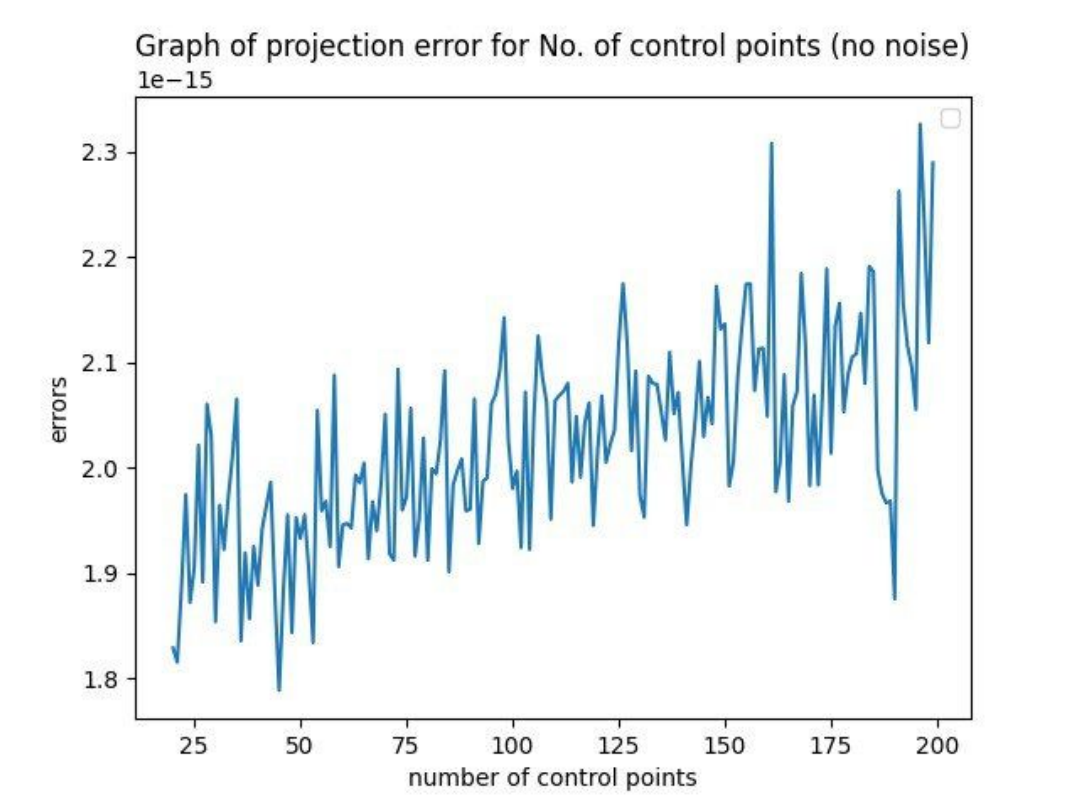


Results

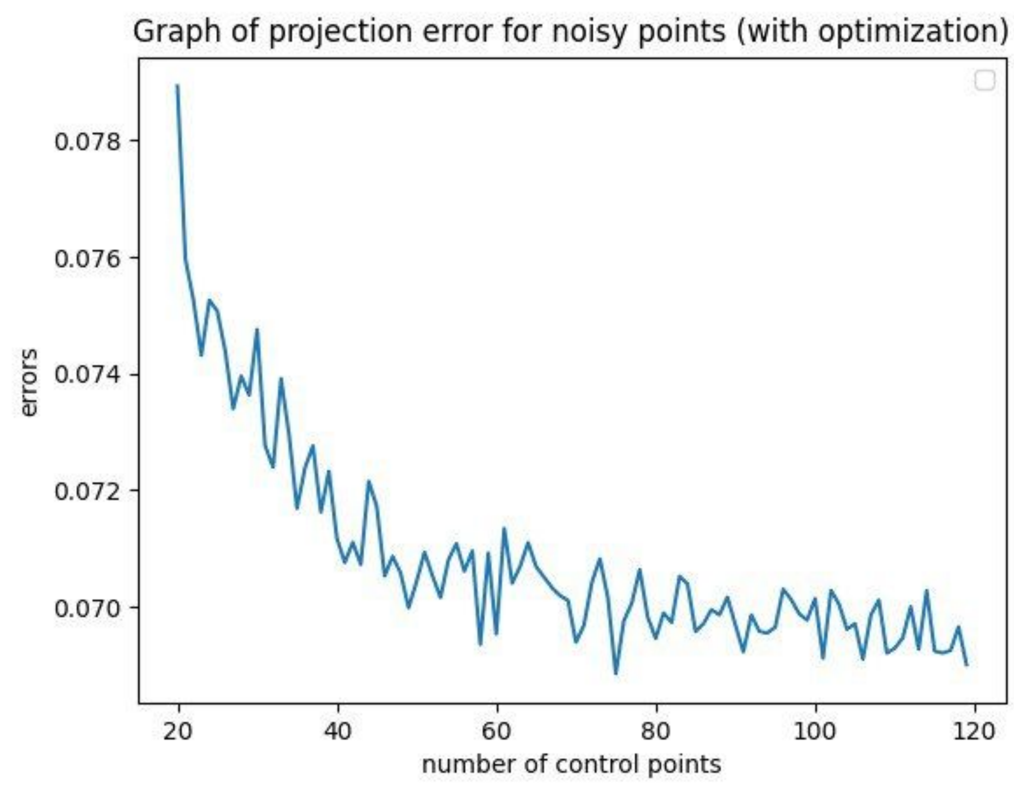
As a result we reconstruct point's position with following precision



Results

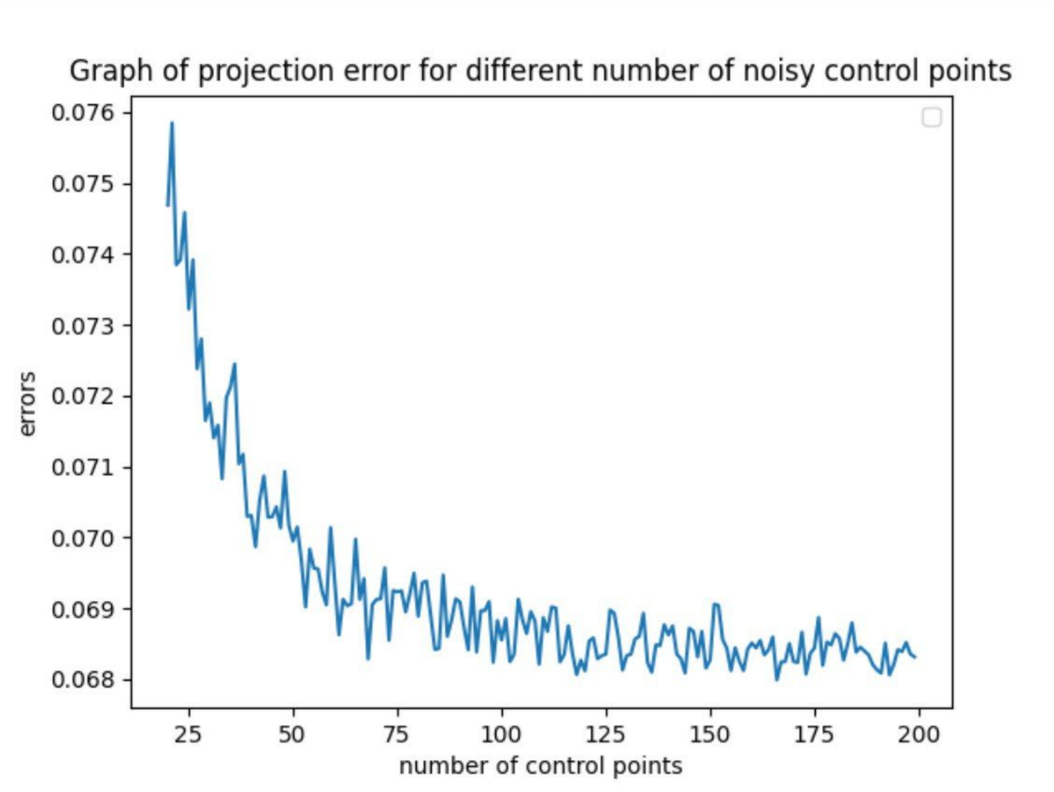

Graph of projection error for No. of control points (no noise)
1e−15
2.3
2.2
2.1
2.0
1.9
1.8
errors
25
50
75
100
125
150
175
200
number of control points

Results



Graph of projection error for noisy points (with optimization)
errors
0.078
0.076
0.074
0.072
0.070
20
40
60
80
100
120
number of control points

Results



Graph of projection error for different number of noisy control points
0.076
0.075
0.074
0.073
0.072
0.071
0.070
0.069
0.068
errors
25
50
75
100
125
150
175
200
number of control points

Tasks distribution

Inna Larina: dataset simulation + dlt programming

Andrei Volodichev: dlt programming + experiments

Folu Obidare: datasets analysing + experiments

Dmitrii Vinichenko: presentation making (results)

Daria Demeshko: presentation making (math) + README