Computer Vision

Project Report

Ziyong Cheong

2025-01-12

1 Recap of Basic Computer Vision Concepts

Scale-invariant feature transform (SIFT) (Lowe 2004) is an algorithm which detects local features in images. These can then be *matched* by their *descriptors*, vectors which describe the local image patch around the feature.

Random sample consensus (RANSAC) (Fischler and Bolles 1981) is an iterative method to fit a model to a dataset which contains many outliers. It works by randomly sampling a subset of the data, fitting the model to this subset, and then evaluating how many points in the entire dataset are consistent with the model, dubbed *inliers*. The model with the most inliers is then chosen.

Direct linear transform (DLT) is a method to solve a system of linear equations in homogeneous coordinates, i.e., equations of the form

$$\lambda_i \mathbf{x}_i = P\mathbf{X}_i$$

where λ_i is an unknown scalar, $\mathbf{x}_i$ and $\mathbf{X}_i$ are vectors, and P is an unknown (camera projection) matrix. Depending on the variables known, this can be rearranged to the matrix-vector equation

$$Mv = 0$$

which can be solved using singular value decomposition (SVD), since v is the last column of the matrix V in the SVD $USV^\top = M$.

2 Algorithm Details

2.1 Robustly Estimating T_i

RANSAC was used to robustly estimate T_i using the reprojection error as the evaluation function. Deriving the DLT equations was rather difficult; a naive approach, shown below, results in a non-homogeneous linear system of equations, which cannot be solved using DLT.

Naive Approach. Consider a 2D-3D correspondence $(x, y, 1)^\top = \mathbf{x}_j \leftrightarrow \mathbf{X}_j = (X_j^\top, 1)^\top$ for a calibrated camera $P = [R\ T]$. Using the camera equations results in

$$\mathbf{0} = \mathbf{x}_j \times P\mathbf{X}_j = \mathbf{x}_j \times (RX_j + T) = \underbrace{\mathbf{x}_j \times RX_j}_{c_j \in \mathbb{R}^3} + [\mathbf{x}_j]_\times T \implies [\mathbf{x}_j]_\times T = -c_j,$$

a non-homogeneous linear system of equations.

Instead, the trick is to represent T in homogeneous coordinates as $\mathbf{T} = (\mathbf{T}_{1:3}, t_4)^\top \in \mathbb{R}^4$ where $T = \mathbf{T}_{1:3}/t_4$. Doing this and multiplying both sides of the camera equations by t_4 results in

$$\mathbf{0} = t_4(\mathbf{x}_j \times P\mathbf{X}_j) = t_4 c_j + [\mathbf{x}_j]_\times t_4 T = t_4 c_j + [\mathbf{x}_j]_\times \mathbf{T}_{1:3}$$

which can now be written for multiple 2D-3D correspondences as

$$\begin{bmatrix} [\mathbf{x}_1]_\times & c_1 \\ [\mathbf{x}_2]_\times & c_2 \\ \vdots & \vdots \end{bmatrix} \mathbf{T} = Mv = 0.$$

3 Results

The results for each dataset $i = 3, \dots, 9$ are given in the files `results[i].mat` and are also visualised below. Surprisingly, datasets 8 and 9 were not as difficult as the other datasets, with the only catch being the dominant planes (the paper below the dinosaur) not exactly coinciding when triangulated. By far, the most difficult dataset was dataset 7, whose final triangulation has a large prominent distortion as well as several trailing points.

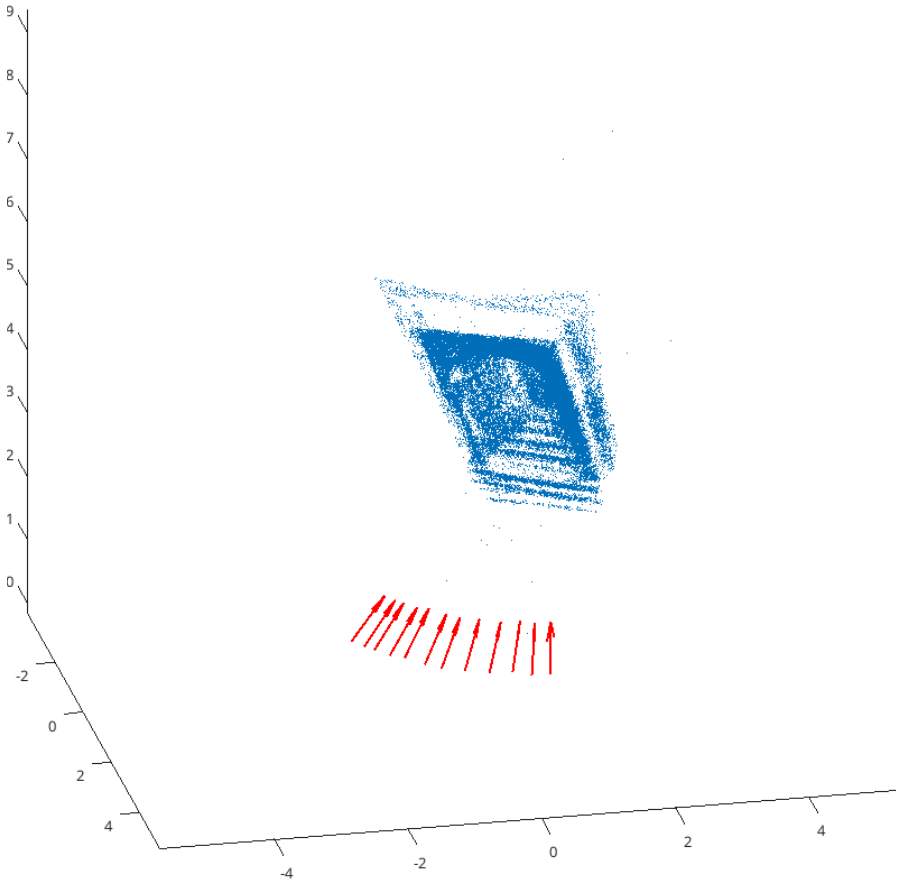

Figure 1: Results for dataset 3.

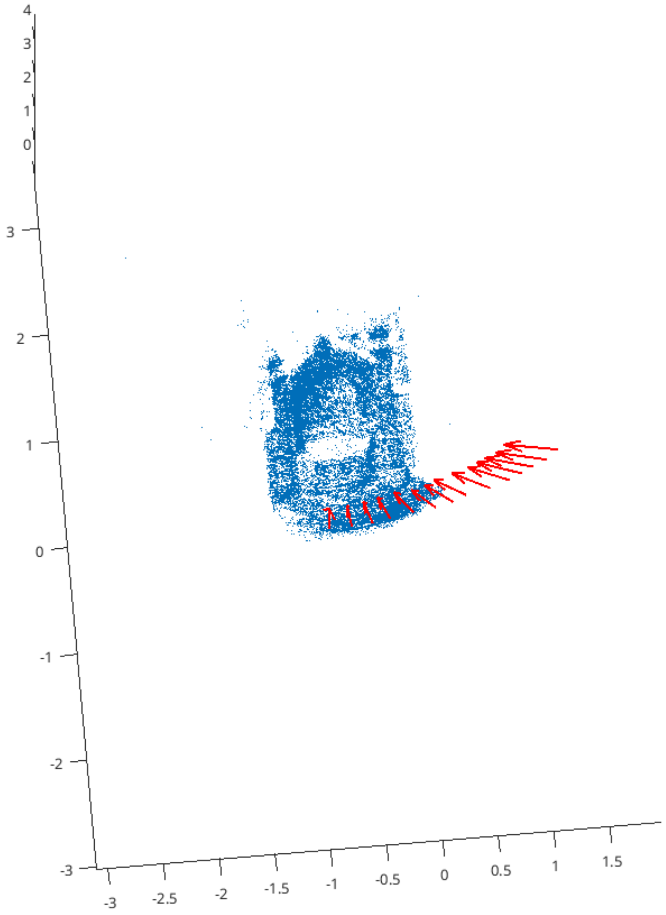

Figure 2: Results for dataset 4.

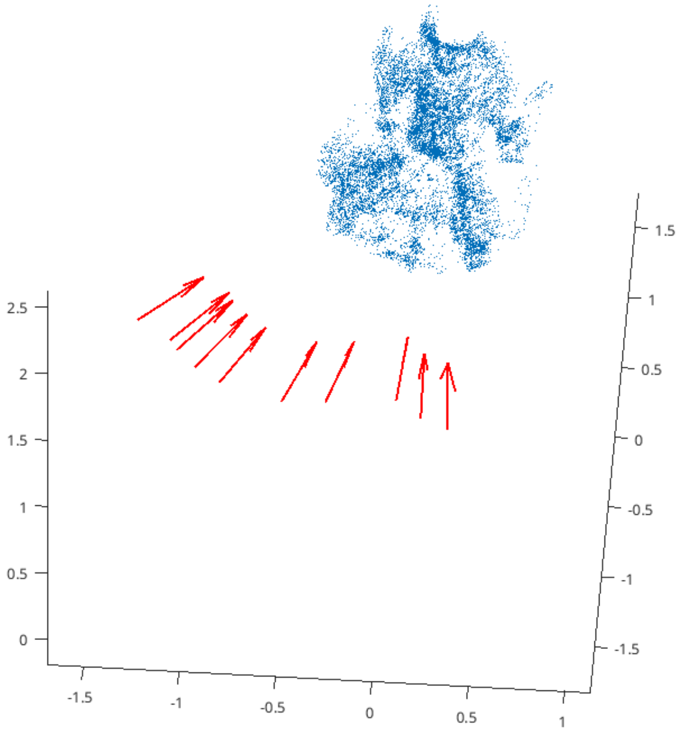

Figure 3: Results for dataset 5.

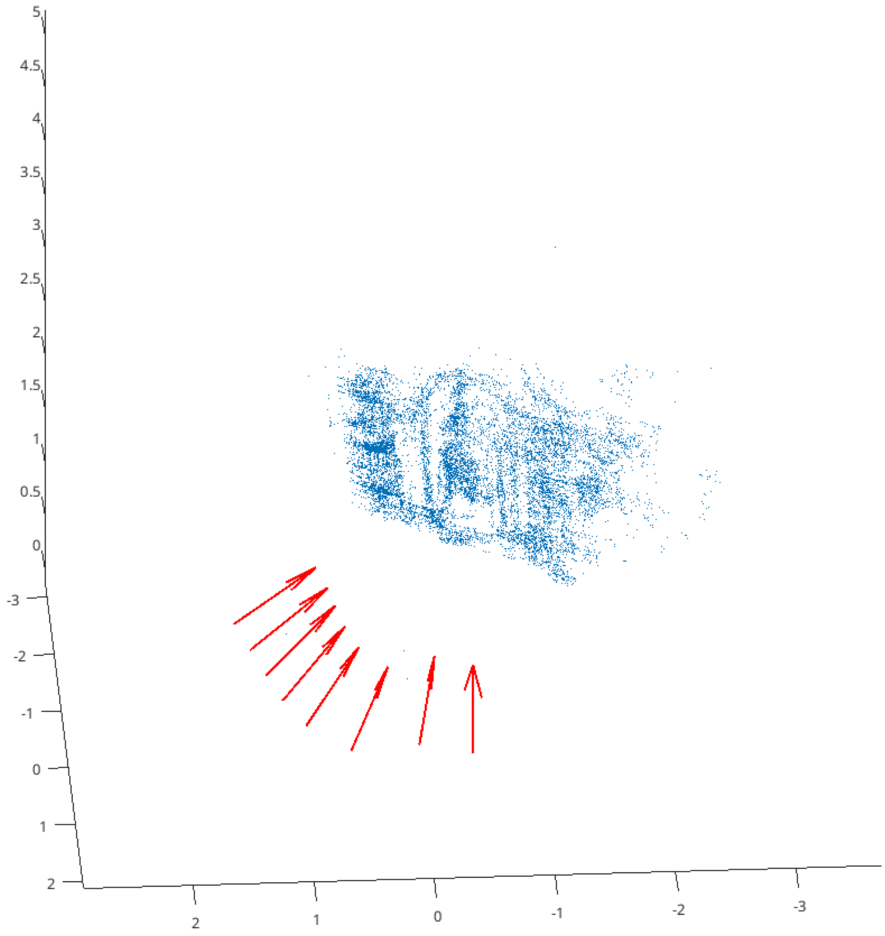

Figure 4: Results for dataset 6.

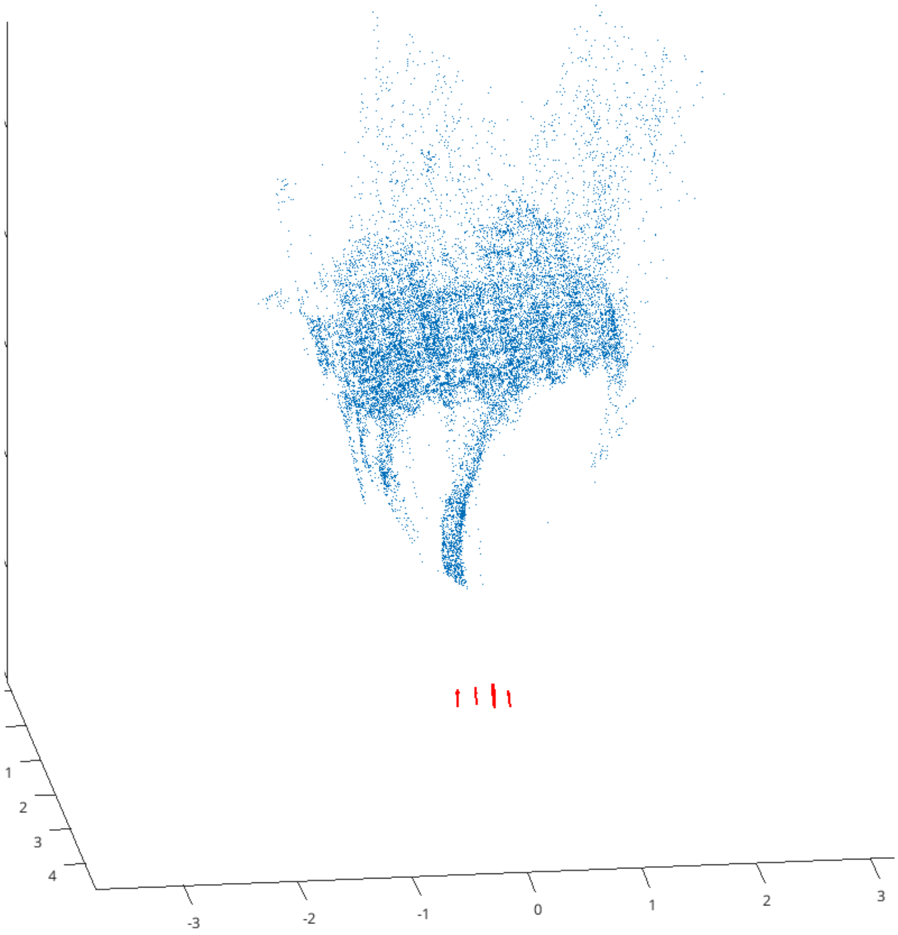

Figure 5: Results for dataset 7.

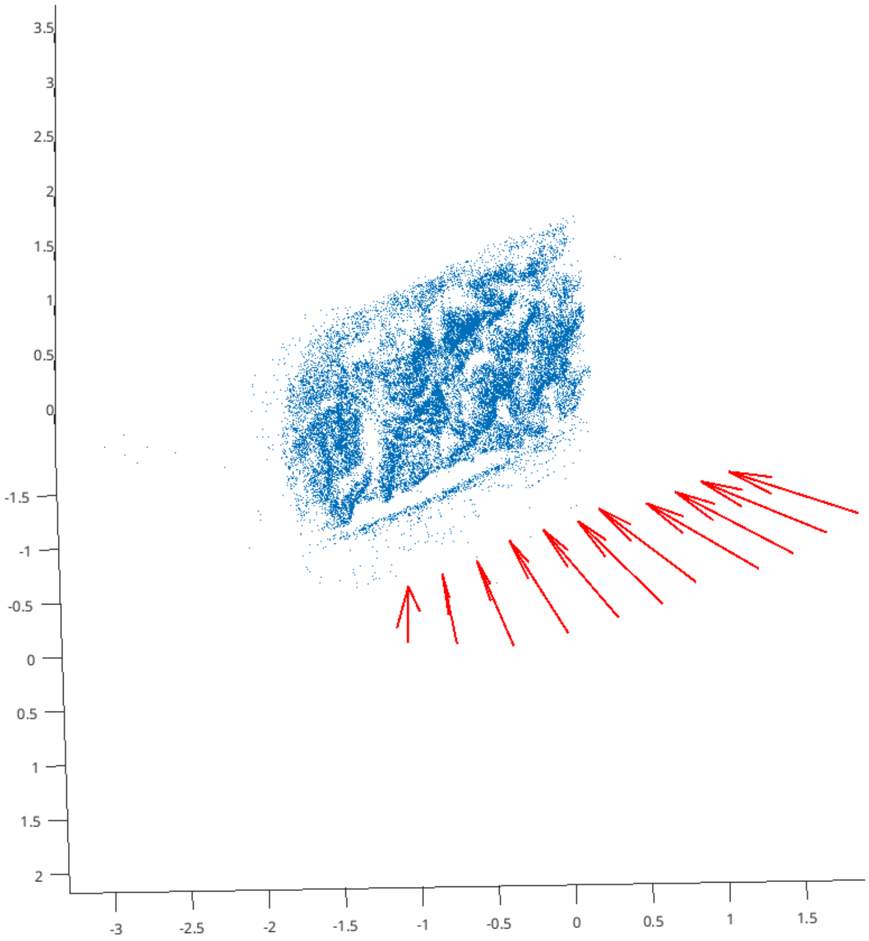

Figure 6: Results for dataset 8.

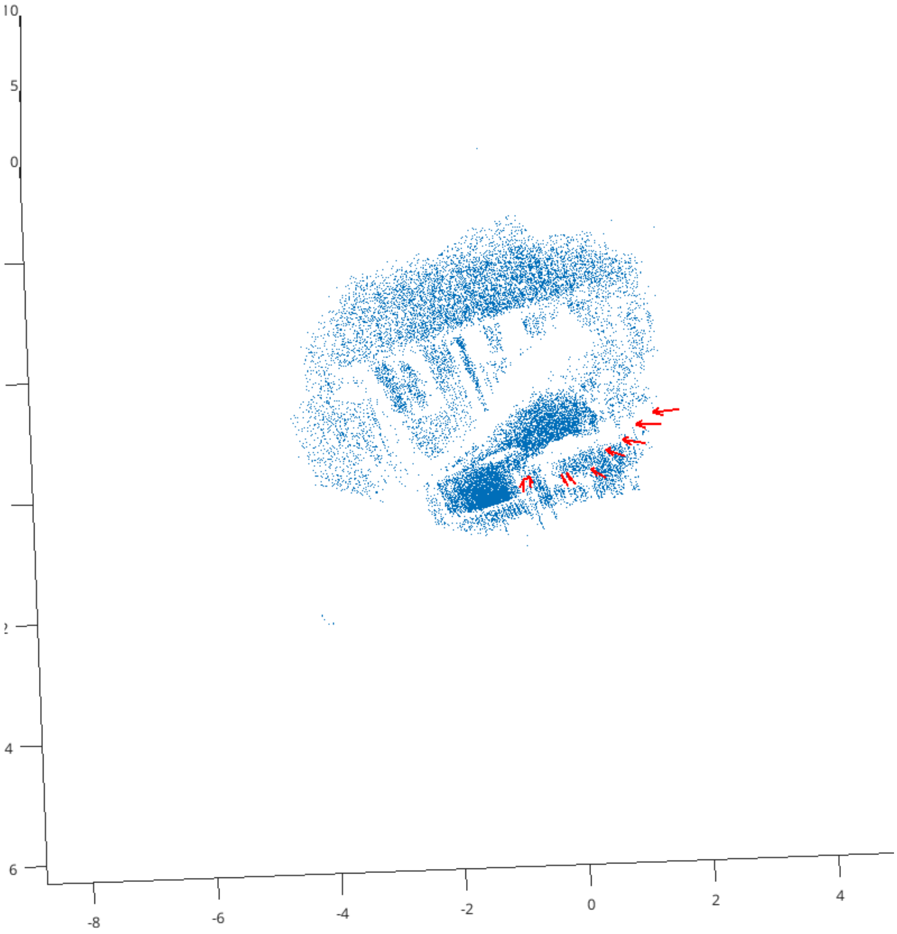

Figure 7: Results for dataset 9.