Project 2: Visual Odometry.

ENPM 673, Robotics Perception

Gunjan K Khut
UID – 115813083

The following steps were involved in the Visual Odometry project:

1. **Check for the camera intrinsic parameters** – The first step in the pipeline was to check for the camera intrinsic parameters. This was done by using the provided function "ReadCameraModel". This generated the focal length, the principal point of the camera and the undistortion lookup table.

2. **Converting the image from bayer to RGB format –** The given image were in a bayer image format, so the first step was to demosaic (with GBRG alignment) the images to standard RGB format.

Fig (2a) – Input Image (Bayer Format)

Fig (2b) – Demosaic Image (RGB Format)

3. **Undistort the image** – After converting the images to standard RGB format. I undistorted the images using the LUT parameters obtained by "ReadCameraModel" function, which were further used as input for "undistortImage" function.

Fig (3) – Undistorted Image

4. **Denoising the image using Gaussian Filter** – Applied Gaussian filter to remove the noise which smoothen the edges in the image. This allowed better detection of the features.

Fig (4) – Denoised Image with Gaussian Filter

5. **Feature detection and Matching** – I used SURF to detect features in the current image and the next image and then matched the corresponding points using matched features across all the frames. This information is used to estimate the fundamental matrix.

Fig (5) - SURF Matched features

6. **Estimate Fundamental Matrix** – To estimate the fundamental matrix, I used normalized 8 points to find the fundamental matrix which is defined in the function (norm2d.m). To calculate the fundamental matrix, we need to remove the unwanted points and normalize the points.

    - Normalization of points - The centroid of all the points are found and then by appropriate translation centroid is translated to the origin. Then mean distance of all the points from the origin was calculated. And then by applying scaling factor to the points mean distance of all the points from the origin was converted to √2.

    - RANSAC - From all the normalized corresponding points, 8 inlier points which satisfies the fundamental matrix were found. In this randomly 8 points were selected from the set of corresponding points and for each 8 points fundamental matrix is satisfaction is decided. Then the best 8 points are selected as the inliers.

    After estimating the fundamental matrix, it was de-normalized by using the following equation.

    $$F = T2' * Fnorm * T1$$

7. **Essential Matrix** – I used fundamental matrix to estimate the essential matrix.

    - After estimating the fundamental Matrix, K matrix is known to us, So it becomes easy to estimate the essential matrix.

    $$E = K' * F * K$$

- The rotation and translation between matched features were determined from the estimated essential matrix. It was done by applying singular value decomposition to the essential matrix and enforcing E to be:

$$E \sim U\ dia(1,1,0)\ V'$$

- W-matrix = [0 −1 0, 1 0 0, 0 0 1]

- R and T obtained is as:

$$R1 = U\ W\ V'$$
$$R2 = U\ W'\ V'$$
$$t1 = (:,3)$$
$$t2 = -(:,3)$$

- We get 4 solutions for the rotation and translation of the camera pose, but we need only one therefore we use triangulation method to solve it.

8. **<u>Triangulation Method</u>** – Triangulation is the process of determining a point in 3D space given its projections onto two, or more, images.

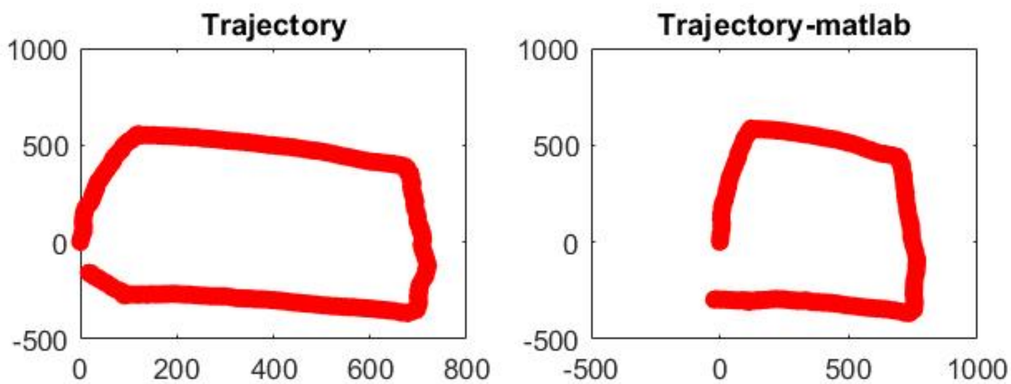
   - The Richard Hartley's triangulation algorithm relates the intrinsic parameters (world frame) and the current frame (given through the rotation and translation matrices).
   - It passes all the matched points through the algorithm and the four point ratio between the two matched point co-ordinates in the two frames and the two camera centers in those frames.
   - This ratio allows us to find the angle between the perpendiculars of the two frames and hence the direction between two frames can be analyzed.
   - The combination of rotation and translation that relates these set of points is then the solution for the camera poses.

9. **<u>Identifying the turn</u>** – I obtained the rotation and translation for the pair of frames, and then updated the location and rotation of the camera as follows:

$$orientation = R * prev\_orientation$$
$$location = prev\_location\ +\ t * orientation$$

10. **<u>Output Plot</u> -** I plotted the camera pose to see how the algorithm estimates it. The output can be seen in Output figure on the next page

Extra Credit – To compare the results, I plotted two different versions of the trajectories: One is the calculated trajectory and the other one is the trajectory given by the MATLAB function "estimateFundamentalMatrix". Triangulation was applied to all of the versions. It can be seen that the different trajectories start at the same point but they drift with time.



Output Figure – Final Output