CS9645b – Computer Vision Techniques

Assignment 4

By,

Rajaraman Ganesan – 251056279

Trajectory is a path that object with mass in motion follows through space as a function of time. Here, from the video the two successive frames are extracted. The extracted image from the video given below:

From the frames of the image, the matrix estimated and given as input. Initially the two frames extracted from video converted to gray scale from the camera. The frames that captured at successive time intervals extracted with all intrinsic parameters, obtained via one of the numerous calibration available.

For each consecutive frame, the rotation matrix and translation vector found which describes the motion of the object between the two frames. Now since we have described the two images, now time to find the matches by using FLANN based Matcher technique. FLANN stands for Fast Library for Approximate Nearest Neighbors. It contains a collection of algorithms optimized for fast nearest neighbour search in large datasets and for high dimensional features.

SIFT is Scale Invariant Feature Transform. For any object, there are many features, interesting points on the object that can be extracted to provide a "feature" description of the object. This description can then be used when attempting to locate the object in an image containing many other objects. There are many considerations when extracting these features and how to record them. SIFT image features provide a set of features of an object that are not affected by many of the complications experienced in other methods, such as object scaling and rotation.

With the above technique, we have estimated the matrix and RANSAC used for estimating geometric transforms. It is an iterative method for estimating a mathematical model from a data set that contains outliers. The RANSAC algorithm works by identifying the outliers in a data set and estimating the desired model using data that does not contain outliers. , RANSAC used as a robust approach to estimate the fundamental matrix in stereo vision, for finding the commonality between two sets of points for feature-based object detection, and registering sequential video frames for video stabilization.

Translation is the shifting of object location. In case of we know the information of the shift in (x,y) direction we can create transformation matrix as given below:

$$M = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \end{bmatrix}$$

Rotation of an image for an angle achieved by transformation matrix of as given below:

$$M = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

When acting on a matrix, each column of the matrix represents a different vector. For the rotation matrix R and vector v, the rotated vector is given by R*v. Rotation angle specified as a real-valued scalar. The rotation angle is positive if the rotation is in the counter-clockwise direction when viewed by an observer looking along the x-axis towards the origin. Angle units are in degrees.

A cameraParameters object that contains the intrinsic, extrinsic, and lens distortion parameters of a camera. It sets properties of the cameraParameters object by using one or more Name, Value pair arguments. Unspecified properties use default values. Also creates an identical cameraParameters object from an existing cameraParameters object with parameters stored in paramStruct.

The intrinsic parameters obtained in assignment 3 given below:

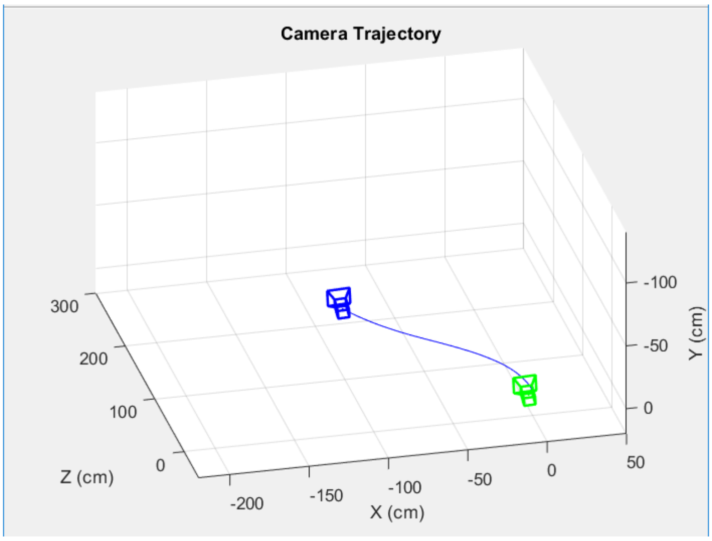
```
-- CAMERA INTRINSIC PARAMETERS --

Focal length in x:
2832.83781786116
Focal length in y:
2824.5861370559783
Pixel skew is:
0.0
Camera origin:
1654.756086788289
 Camera origin 2:
1218.3565198615536
```

Once the essential matrix E estimated with rotation matrix and translation vector, eliminate the outliners from the feature matches. An outlier is an observation that lies outside the overall pattern of a distribution. Next is to estimate fundamental matrix. EstimateFundamentalMatrix estimates the fundamental matrix from corresponding points in stereo images. Configure this function to use all corresponding points or to exclude outliers. You can exclude outliers by using a robust estimation technique such as random-sample consensus (RANSAC). When you use robust estimation, results may not be identical between runs because of the randomized nature of the algorithm.

Structure from motion (SfM) is the process of estimating the 3-D structure of a scene from a set of 2-D views. It is used in many applications, such as robot navigation, autonomous driving, and augmented reality. Next is to bundle the camera poses using bundle adjustments. It returns the refined 3-D points and camera poses that minimize re-projection errors. The refinement procedure is a variant of the Levenberg-Marquardt algorithm. Here, all parameters are inputted and an updated view is set. With the help of re-projection error and some random values, the current view of the camera estimated.

Finally, the camera pose obtained on doing above factors and with help of rotation matrix and translation, vector camera trajectory graph is plotted. GNUPLOT does a very good job of displaying 3D surfaces. It sketches them out with lines, or some simple solid surfaces. In some cases, MATLAB can take the same data and make a much more attractive image. This set of programs tries to read a few typical GPL data files, and display them appropriately. It is a MATLAB program, which illustrate how a program can write data, and command files so that gnuplot can create plots of the program results.

The output of camera trajectory 3D plot is given below:



References:

1. https://www.researchgate.net/publication/220669319_Monocular_Camera_Trajectory_Optimization_using_LiDAR_data
2. https://in.mathworks.com/help/vision/examples/monocular-visual-odometry.html
3. https://in.mathworks.com/help/mpc/ug/lane-following-control-with-monocular-camera-perception.html
4. https://in.mathworks.com/help/vision/ref/estimateworldcamerapose.html
5. https://in.mathworks.com/help/vision/ref/bundleadjustment.html