ROBOT PERCEPTION

ENPM 673 – PROJECT 2

NAME: AKSHAY RAJARAMAN

UID: 115625136

CONTENTS:

VISUAL ODOMETRY

Visual odometry is the process of determining the position and orientation of a robot by analyzing the associated camera images.

The types of visual odometry include:

1. Monocular visual odometry: Only one camera is used to obtain visual data. Using monocular odometry, one can only obtain a scaled version of the 3-dimensional space around the camera.
2. Stereo visual odometry: Two cameras are placed in stereo setup to obtain the visual data. Stereo visual odometry can be used to obtain an accurate representation of the 3-dimensional space around the cameras.
3. Feature based and Direct method
4. Visual Inertial Odometry

HARRIS CORNER DETECTION:

Motion estimation requires the finding of corresponding features across two or more views. Often, the elements to be matched are image patches of fixed size. The corner points in an image are the junction of the contours of the objects in the image.

Since corner points have a large variations of intensity levels for a small displacement in the image, they can be used to obtain the common features between two or more images.

The basic idea behind the Harris Corner detector is to obtain the pixel intensity for a window across the image. If the point the window is located at was a corner, nudging the window in any direction would cause a large change in the intensity values of the window. This would not be so in the case of edges and plain patches.

The change in intensity is measured as a sum of squared differences among the pixels present in a window when it is nudged by a small value.

If the window moves by u and v in the x and y direction of the image respectively, then the change in intensity E for a movement of (u,v) is given by

$$E(u,v) = \Sigma[I(x+u,y+v)-I(x,y)]^2$$

Using Taylors series, the above equation can be reduced to:

$$E(u,v) = [u\ v]\ M\ [u\ v]';$$

Where,

$$M = [I_x^2\ I_xI_y;\ I_xI_y\ I_y^2]$$

And I_x and I_y are the derivatives of the window in the image with respect to x and y directions and is given by:

$$I_x = 1/3 *[-1\ 0\ 1;-1\ 0\ 1;-1\ 0\ 1]*I$$

And

$$I_y = 1/3[1\ 1\ 1;0\ 0\ 0;-1\ -1\ -1]*I$$

We then calculate R as:

$$R = \det(M)/(I_x+I_y)$$

If R was very large, then the part of the image in the window is said to be a corner.

RANSAC ALGORITHM:

RANSAC stands for RANdom Sampling and ACquisition. Once we obtain the features based on Harris Corner detection, We check two consecutive images for matched feature points, This is done by obtaining the SSD of the pixel intensities of all the feature points in the second image that in within a particular radius of the feature point in image one. The window with least SSD is matched to the feature in image one and the coordinates of both features are stored.

8 random points are then selected from these matched feature points, and the fundamental matrix is calculated using the 8 point algorithm and the cost is computed as the squares of the values of the vector x' F x where x' and x are image coordinates for the eight features in both the images.

The above process is iterated for n times and the F matrix with least cost is chosen as the Fundamental Matrix.

The inliner points are then calculated as the points that have least value for the equation x' F x where x' and x are now the entire set of matched points in image1 and image 2.

8 POINT ALGORITHM:

The solution to the equation x’ F x = 0 where x’ and x are the 8 randomly sampled points is given by the reshaped matrix formed from the values in the last column of the V matrix from the singular value decomposition of a matrix A that is represented by

A = [x'x x'y x' y'x y'y y’ x y 1]

i.e if

[U,D,V] = svd(A);

F = reshape(V(:,9),3,3)’;

The Fundamental Matrix obtained is then normalized and passed on to the RANSAC algorithm to determine the least cost matrix with maximum number of outliers.

ESSENTIAL MATRIX:

The essential matrix is the cross product of the rotation matrix and the translation vector and can be obtained from the Fundamental Matrix using the cameras intrinsic parameters.

We obtain the intrinsic parameters of the camera using the ReadCameraModel method that has been provided as follows:

[fx,fy,cx,cy,~,LUT]=ReadCameraModel('.\stereo\centre','.\model');

The cameras intrinsic parameter matrix is given by

K = [fx 0 cx;0 fy cy;0 0 1]

The essential matrix is then given as:

$$E = K^T F K$$

The rank of the essential matrix is then reduced to 2 using the SVD as follows:

$$[U,V,D] = svd(E)$$

$$E = U*(diag([1\ 1\ 0]))*V^T$$

SOLVING FOR R AND T:

The Rotation Matrix R and translation vector T are obtained from the Essential matrix using the equations:

$$[U,D,V] = svd(E)$$

$$\text{Rotation matrix} = U*\pm W*V^T$$

The skew symmetric matrix of translations is given by

$$T = (U*\pm W*diag([1\ 1\ 0]))*U'$$

As it can be observed, it is possible to obtain 4 different combinations of R and T matrices which implies that we obtain 4 different 3 dimensional points when we use the triangulation technique.

We obtain the 3 Dimensional coordinates of the features from the image coordinates using the equation

$$X = -R^T T + RK^{-1}x$$

Where X is the 3D coordinates of x.

The corresponding 3D coordinates for the four possible combinations of R and T are obtained and the combination for which the 3D point lies in front of the camera is chosen as the final rotation and translation matrix.

The camera coordinates can then be calculated as

Coord(currentFrame+1) = Coord(currentFrame)+RotationMatrix(currentFrame+1)*translationMatrix(currentFrame+1)

Here, the RotationMatrix considered is with respect to the global coordinates. i.e.

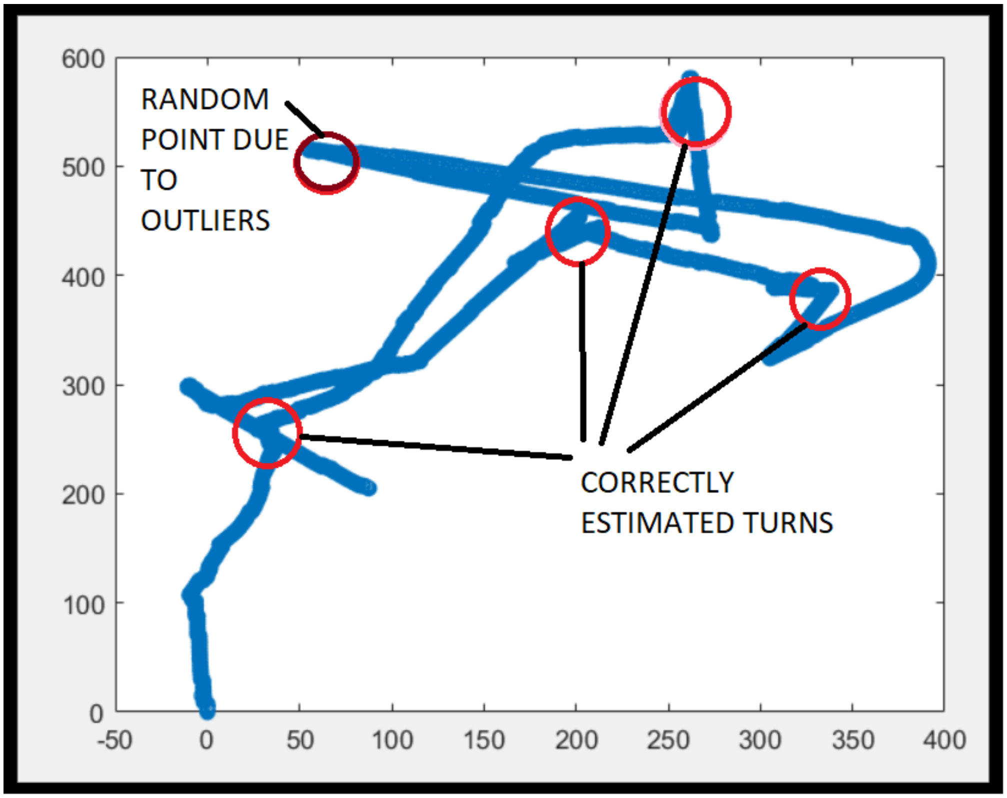
RotationMatrix(currentFrame+1) = RotationMatrix(currentFrame+1)*RotationMatrix(currentFrame)

The coordinates of the camera can then be plotted to obtain the visual path of the camera.

RESULTS

Since the image from the sensors can contain noise, It is possible to obtain a perfect path of the robot only with a few optimization algorithms that reduce the noise from the data.

The presence of outliers, though can be reduced using the RANSAC algorithm, cannot be eliminated completely. The presence of outliers in greatly reduce the performance of the system.



As it can be seen from the image above, the turns of the camera have been captured correctly, but the path taken by the camera is not estimated correctly. This could be due to the outlier errors. The presence of even one outlier during computation can affect the position of the points calculated after and hence affects the performance of the system exponentially.