ROB GY-6213 Robotic Localization and Navigation project II

Yash Khanolkar yak9424 N12169803

7 April 2024

1 Introduction

Projective geometry is a pivotal part of Robot Localization and Navigation, which was developed to explain perspective changes of the three-dimensional objects when projected to a plane. Another important concept is Optical Flow and Motion field, which is primarily applied in the area of computer vision. This gives an ideal representation of motion of an image or object in 3-D space as it is projected onto a camera image. We can estimate the position and orientation of a point or an image(quadrotor in our case) using these concepts which is final aim of our project. This is done based on April tags, which are a set of unique ids to provide low accuracy, high accuracy localization for many different applications in the field of Robotics.

2 Problem Statement

Our initial task is to implement a 3-D vision based Pose estimation system. Here, the data provided is of a Nano+ quadrotor flown over a mat of April Tags. We have to perform calibration between the camera frame and the robot frame to transform the pose estimates so as to compare them against the ground truth vicon data.Using this camera calibrated data, the corners of the tags and it's world frame location, we are supposed to calculate the measured pose of the quadrotor for each packet of data.

This task must be commenced by extracting and tracking the corners of each image. The sparse optical flow between two images and the timestamps will give the velocity in the calibrated image frame. Making use of this information, the linear and angular velocities must be estimated.

Now, it is possible that there are inconsistencies in the optical flow computation. We need to implement RANSAC to reject outliers to make our estimates more accurate. A 3-point RANSAC is advised as, in our case, three sets of constraints are required to solve the system.

3 Procedure

In the part 1 of the problem, we have to draw the graph of the position and orientation values of a quadrotor with the help of a mat of April Tags using Projective geometry. Firstly, we need to find out the coordinates for the corners for each April Tags with the help of the IDs provided to us. This is called in the main function where the position and orientation is derived with the help of these coordinates.

The point coordinates which were calculation by us is denoted by $\begin{pmatrix} x_i & y_i & 1 \end{pmatrix}^T$ are the values in terms of distances which are derived by us from the given mat of April Tags.

Consequently, there are a group of pixel values which are returned to us by the camera sensor on the quadrotor which can be given as $\begin{pmatrix} x_i^{'} & y_i^{'} & 1 \end{pmatrix}^T$.A few algebraic tricks can be performed to find the matrix providing all the information about these values

$$A = \begin{pmatrix} x_i & y_i & 1 & 0 & 0 & 0 & -x_i^{'}x_i & -x_i^{'}y_i & -x_i^{'} \\ 0 & 0 & 0 & x_i & y_i & 1 & -y_i^{'}x_i & -y_i^{'}y_i & -y_i^{'} \end{pmatrix}$$

These values are arranged in the form of the matrix for containing information for all 5 corner coordinates, including the center of the April Tag. We take the singular Valued Decomposition in the camera frame to get the U, S and V matrices , where the 9th column of V gives us the H matrix which is useful to solve the step of projective transformation where,

$$\begin{pmatrix} \hat{R}_1 & \hat{R}_2 & \hat{T} \end{pmatrix} = K^{-1}H$$

These element vectors of the matrix can be applied to find the SVD with respect to the Robot frame which can be denoted by R.

$$\begin{pmatrix} \hat{R}_1 & \hat{R}_2 & \hat{R}1X\hat{R}2 \end{pmatrix} = USV^T$$

where,

$$R = U \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & det(UV^T) \end{pmatrix} V^T$$

The diagonal of the above matrix guarantees that it is a rotation matrix. Now, we need to make sure that our estimate of translation is in the right scale:

$$T = \hat{T}/||\hat{R}_1||$$

After finding the values of the H, R and T matrices, we have to deduce the transformation matrix form the figures given to us.The transformation matrix given in the robot frame is

$$\begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix}$$

Inversed with the transformation matrix calculated by us, i.e,

$$\begin{pmatrix} 0.707 & -0.707 & 0 & 0.04 \\ -0.707 & -0.707 & 1 & 0 \\ 0 & 0 & -1 & -0.03 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

this matrix gives us the pose of the quadrotor which contains the final position and orientation of the robot which is our result for part 1.

In part 2 of the problem, we have to find the translation and angular velocity in the body frame expressed in the camera frame when the camera is moving and the rest of the world is static. This can be expressed in the camera frame as $\begin{pmatrix} v_x & v_y & v_z \end{pmatrix}^T$ and $\begin{pmatrix} \omega_x & \omega_y & \omega_z \end{pmatrix}^T$. The position of 2D point on the image is given by p = $\begin{pmatrix} x & y & 1 \end{pmatrix}^T$ where p = P/Z; Z referring to the depth of the image.

Now, this can be calculated based on the motion field equations which is given by

$$\dot{p} = \frac{1}{Z} A(p)V + B(p)\Omega$$

which can be further elaborately expressed as

$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{1}{Z} \begin{pmatrix} -1 & 0 & x \\ 0 & -1 & y \end{pmatrix} \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix} + \begin{pmatrix} xy & -(1+x^2) & y \\ 1+y^2 & -xy & x \end{pmatrix} \begin{pmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{pmatrix}$$

We need the solve this motion field equation to get our final solution. There are 4 cases possible for such an equation - known depth, no translation velocity, no angular velocity and both translation and angular velocity with unknown depth which are increasingly harder to find, in the same order. Fortunately, in our case, we have known depth depth, hence the velocities can be found out by using least square method.

Case 1: Known Depth

With known depth the equations reduce to:

$$\dot{\mathbf{p}} = \frac{1}{Z} A(\mathbf{p})\mathbf{V} + B(\mathbf{p})\mathbf{\Omega} = \begin{pmatrix} \frac{1}{Z} A(\mathbf{p}) & B(\mathbf{p}) \end{pmatrix} \begin{pmatrix} \mathbf{V} \\ \mathbf{\Omega} \end{pmatrix}$$

And this can be solved with a least squares problem:

$$\mathbf{V}^*, \mathbf{\Omega}^* = \arg\min_{\mathbf{V},\mathbf{\Omega}} \sum_{i=1}^{n} \left\| \begin{pmatrix} \frac{1}{Z_i} A(\mathbf{p}_i) & B(\mathbf{p}_i) \end{pmatrix} \begin{pmatrix} \mathbf{V} \\ \mathbf{\Omega} \end{pmatrix} - \dot{\mathbf{p}}_i \right\|^2$$

$$\mathbf{V}^*,\boldsymbol{\Omega}^* = \boxed{\arg\min_{\mathbf{V},\boldsymbol{\Omega}} \boxed{\sum_{i=1}^{n} \boxed{\left\| \begin{pmatrix}\frac{1}{Z_i}A(\mathbf{p}_i) & B(\mathbf{p}_i)\end{pmatrix}\begin{pmatrix}\mathbf{V}\\ \boldsymbol{\Omega}\end{pmatrix} - \dot{\mathbf{p}}_i \right\|}^2}}$$

$$\boxed{\left(\boldsymbol{H}_i\begin{pmatrix}\mathbf{V}\\ \boldsymbol{\Omega}\end{pmatrix} - \dot{\boldsymbol{p}}_i\right)^T\left(\boldsymbol{H}_i\begin{pmatrix}\mathbf{V}\\ \boldsymbol{\Omega}\end{pmatrix} - \dot{\boldsymbol{p}}_i\right)} \qquad \boxed{\left(\mathbf{H}\begin{pmatrix}\mathbf{V}\\ \boldsymbol{\Omega}\end{pmatrix} - \dot{\boldsymbol{p}}\right)^T\left(\mathbf{H}\begin{pmatrix}\mathbf{V}\\ \boldsymbol{\Omega}\end{pmatrix} - \dot{\boldsymbol{p}}\right)}$$

$$\boxed{\left(\begin{pmatrix}\frac{1}{Z_1}\boldsymbol{A}(\boldsymbol{p}_1) & \boldsymbol{B}(\boldsymbol{p}_1)\\ \vdots & \\ \frac{1}{Z_n}\boldsymbol{A}(\boldsymbol{p}_n) & \boldsymbol{B}(\boldsymbol{p}_n)\end{pmatrix}\begin{pmatrix}\mathbf{V}\\ \boldsymbol{\Omega}\end{pmatrix} - \begin{pmatrix}\dot{\boldsymbol{p}}_1\\ \vdots\\ \dot{\boldsymbol{p}}_n\end{pmatrix}\right)^T\left(\begin{pmatrix}\frac{1}{Z_1}\boldsymbol{A}(\boldsymbol{p}_1) & \boldsymbol{B}(\boldsymbol{p}_1)\\ \vdots & \\ \frac{1}{Z_n}\boldsymbol{A}(\boldsymbol{p}_n) & \boldsymbol{B}(\boldsymbol{p}_n)\end{pmatrix}\begin{pmatrix}\mathbf{V}\\ \boldsymbol{\Omega}\end{pmatrix} - \begin{pmatrix}\dot{\boldsymbol{p}}_1\\ \vdots\\ \dot{\boldsymbol{p}}_n\end{pmatrix}\right)}$$

$$\boxed{\begin{pmatrix}\mathbf{V}^*\\ \boldsymbol{\Omega}^*\end{pmatrix} = \mathbf{H}^{\dagger}\dot{\mathbf{p}}}$$

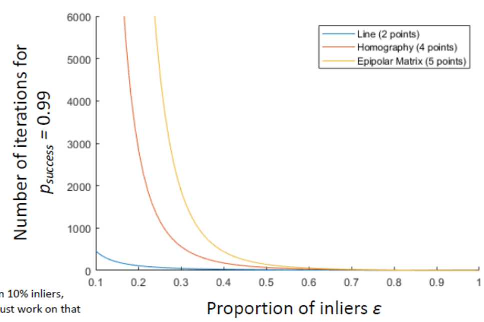
This is the translation and angular velocity found by applying optical flow and projective geometry where we consider all the points of an image which results in there being some inaccuracies in our final output graphs.

This is solved in part 3 of the project where we have to perform RANSAC, which means, to select random points in the image and check how many inliers it contains. The point with the highest inliers is considered. This increases the accuracy of our results and gives us a refined solution. In our cases we are selecting 3 points for outlier rejection because we have a set of 3 constraints with us.

Probability of Hitting Inliers

$$k = \frac{log(1 - p_{success})}{log(1 - \epsilon^M)}$$

For $p_{success}$ = 0.99 the k needed for various models:



If you have less than 10% inliers, maybe you should just work on that

RANSAC: Random Sample Consensus

The algorithm (in pseudocode):

```
Repeat for k iterations
1.  Choose a minimal sample set
2.  Count the inliers for this set
3.  Keep maximum, if it exceeds a desired number of inliers
stop.
```

Step 1: $\begin{pmatrix}\mathbf{V}^*\\ \mathbf{\Omega}^*\end{pmatrix} = \mathbf{H}^\dagger\dot{\mathbf{p}}$

Step 2: $\left\| \left(\frac{1}{Z_i}A(\boldsymbol{p}_i) \quad B(\boldsymbol{p}_i)\right)\begin{pmatrix}\mathbf{V}^*\\ \mathbf{\Omega}^*\end{pmatrix} - \dot{\boldsymbol{p}}_i \right\|^2 \leq \beta$

Recompute your solution with the optimal set

$$\begin{pmatrix}\mathbf{V}^*\\ \mathbf{\Omega}^*\end{pmatrix} = \mathbf{H}^\dagger\dot{\mathbf{p}}$$

4 Code

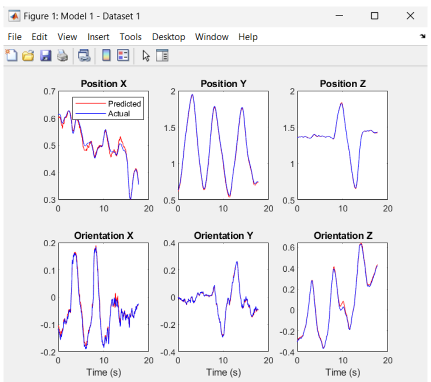
To implement any of the data provided to us, we need to find the corner coordinates of all the April Tags given to us. We use the April Tag Ids to pinpoint the rows and columns where the April Tag corners are present and accordingly calculate the location of each corner with respect to the origin,i.e, the top-left corner. We call this in the main function and store this theoretical and pixel-based corner information in a matrix on which we operate to find out our results. The SVD of this matrix in the camera and robot frame gives the transformation of the camera matrix with respect to the world frame. Consequently, we also find the transformation matrix in the camera frame with respect to the Robot by the images given to us. Inversing both of them gives us the pose matrix which contains the position and orientation information of the quadrotor.

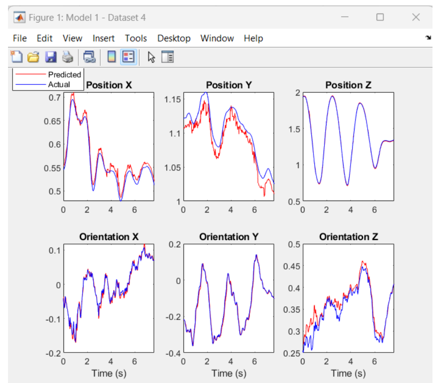
Next, we perform the Optical flow where need to take a current and next image and compare them to identify what changes occur in that time instant. We have to detect the edges of these images and track the points of the next image in the current image. We calculate V and Z values which can important to apply the motion field formula in order to calculate Optical flow without the help of RANSAC. This will give us a slightly skewed output even when a filter is applied, due to the huge amount of points considered.

To prevent this , RANSAC is applied, where we take only 3 points in our case of kind the value of K, which is the probability of hitting inliers. Then the same method as optical flow is followed but at the end we take the norm of our values to compare them with the range of the inliers and find out which line has the more inliers to increase the accuracy of the code.

5 Results

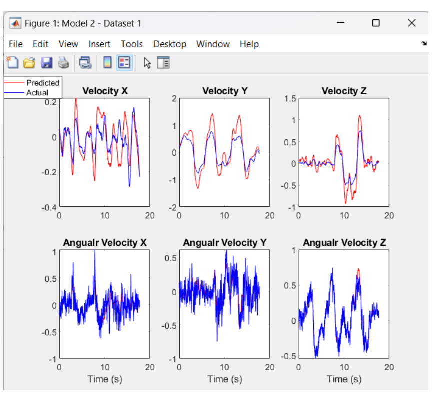
5.1 Part I

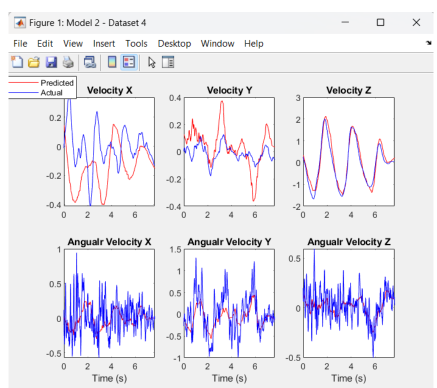




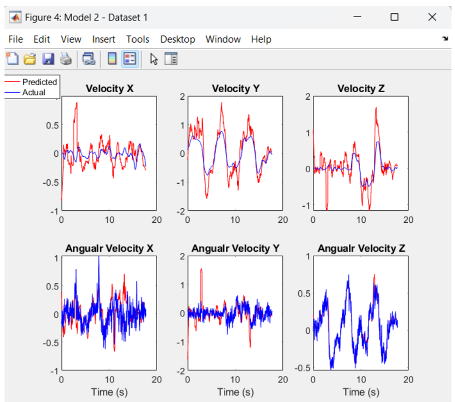
We can observe how the position and orientation information of the expected values and the values derived by implementing Projective Geometry is similar in all aspects for both the datasets which denotes that the method is consistent of any given vicon data. The blue line nearly overlaps the red line in this case. If the getCorners.m file has some error or the transformation matrix is incorrect, then there will be some deviation in our graph, indicating that the correct position and orientation is not being stored in such a case.

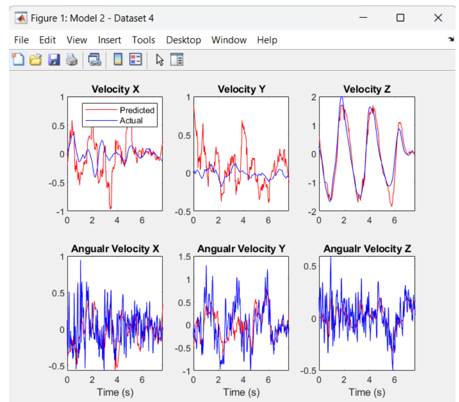
5.2 Part II





What we can see from the translation and angular velocity of both the datasets is that the predicted plot is very much in sync with the expected graph. But as with any other method, some points may be inconsistent and may decrease the overall accuracy of the plot. This problem can be reduced by some margin by implementing RANSAC, which increases the smoothness of the output graphs and reduces most of the issues we encountered when only solely relying on Optical flow.





6 Conclusion

We have demonstrated the practical implementation of Projective geometry and Optical flow and how it is useful in Robotic Localization and Navigation. April Tags is a powerful information storage tool from the fact that it can store all the vicon information in a single Map. It has the scope to store even complex information for robotic implementation. Also, this project highlights the advantages of Matlab in effortless decoding, running and displaying accurately, all the aspects and details of the path and pattern of quadrotor motion.

Finally, we learnt that the larger the data included in our estimation, the more erratic and inconsistent our predicted output will be from the expected output which puts a light on the advantages of RANSAC and under which conditions and circumstances can it be to accurately map the data.

7 References

1) *Richard Szeliski, Computer Vision: Algorithms and Applications, 2nd Edition, final draft, Sept. 2021, Springer*
2) *Siciliano, L. Sciavicco, L. Villani, G. Oriolo, "Robotics: Modelling, Planning and Control", Springer*
3) *Y. Ma, S. Soatto, J. Kosecka, S. Shankar Sastry, "An Invitation to 3D Vision", Springer*
4) *Professor Loianno's Lecture notes*
5) *https://www.mathworks.com/*