Project 4a: Stereo Visual Odometry

Team 6: Noob Quaternions

using 2 late days

Prateek Arora
Masters of Engineering in Robotics
University of Maryland, College Park
Email: pratique@terpmail.umd.edu

Abhinav Modi
Masters of Engineering in Robotics
University of Maryland, College Park
Email: abhi1625@umd.edu

I. Introduction

The aim of the project is to compute the 3D trajectory of the quadrotor using a images from downward facing stereo camera.

II. Data Collection

Images from downward facing stereo camera of the quadrotor moving in helix trajectory were bagged. We have used the bag provided by team BRZ for this project.

III. Implementation details

The pipeline is described in detail in the following subsections:

A. *Stereo Visual Odometry using optical flow and depth*

In this approach, linear and angular velocities are computed using the optical flow equation. These velocities are then integrated to get pose of the quadrotor. Another variable necessary to compute the velocities is the depth which is computed from the stereo camera.

B. *Feature matching and tracking using Optical flow*

First step towards calculating optical flow between current frame $\mathcal{I}_t$ and next frame $\mathcal{I}_{t+1}$ is to get reliable features. We use Opencv function *goodFeaturesToTrack()* to extract Shi-Tomasi features in frame $\mathcal{I}_t$. GoodFeaturesToTrack() works best for out case considering trade-off between speed and accuracy.

Let (x_{t_i}, y_{t_i}) and ($\mathrm{x}_{(t+1)_i}$, $\mathrm{y}_{(t+1)_i}$) denote the coordinates of features in frame $\mathcal{I}_t$ and $\mathcal{I}_{t+1}$ respectively. Since the tracker is not perfect there is some drift in the coordinates of features in the frame $\mathcal{I}_{t+1}$ which results in poor 3D velocity estimates. To ensure that we get accurate coordinates of features we use **RANSAC** to prune poor matches. The model chosen for RANSAC is homography between $\mathcal{I}_t$ and $\mathcal{I}_{t+1}$ and the model-specific loss function is the difference between the (x_{t_i},y_{t_i}) and warped ($\mathrm{x}_{(t+1)_i}$,$\mathrm{y}_{(t+1)_i}$). Now, that we have better feature matches, all that is needed to get linear and angluar velocities is depth.

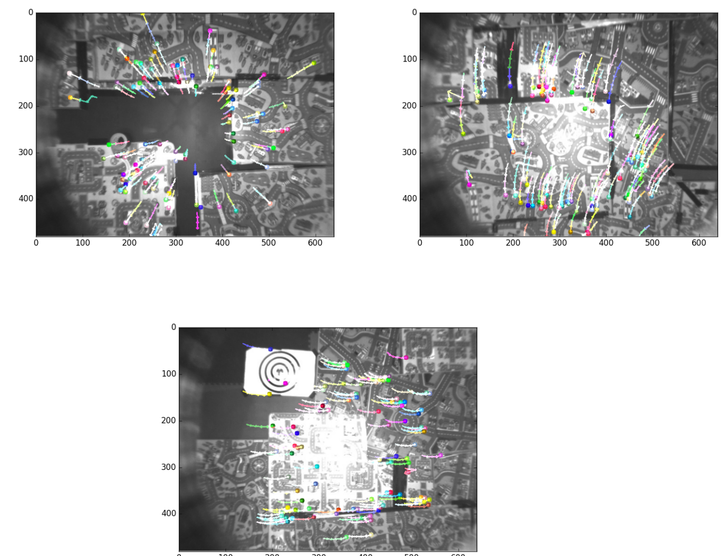

Figure 1: Klt tracker with reset after 5 frames

1) Stereo depth: Opencv function $StereoBM\_create()$ is used to estimate pixel-wise disparity from stereo camera. Using equation 1, depth can be easily estimated from disparity.

$$d = \frac{B \times f}{Z} \tag{1}$$

where d,B,f,Z are the disparity, baseline, focal length and depth respectively.

Getting depth from disparity map is not sufficient to compute robust VO since features with very small values results in absurdly large depth values. To remedy this, mean of the disparity values is computed for those pixels where the disparity is greater than a certain threshold. This assumption hold true because the downward facing camera is looking at a plane and this gives us particularly robust results. Now that we have both depth and optical flow, we compute 3D

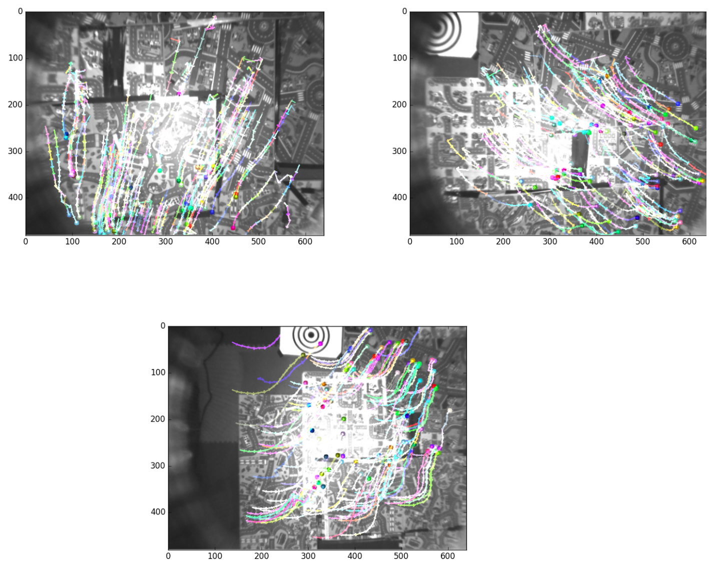

Figure 2: Klt tracker with reset after 20 frames

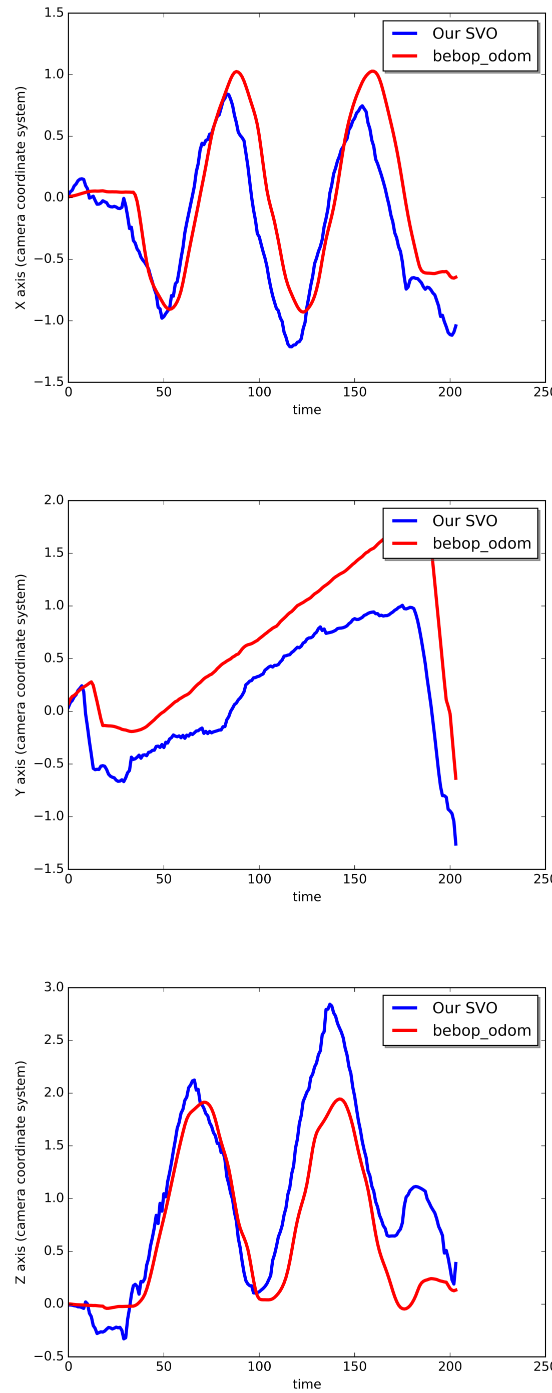

Figure 3: Comparision of x axis, y axis and z axis vs time graph of the Helix trajectory computed using SVO and bebop's odometry

velocities.

2) Pose Using Dead Reckoning: After obtaining optical flow and depth we use linear least squares to compute the velocities as given in the equation 2

$$\begin{bmatrix}\dot{x}\\ \dot{y}\end{bmatrix} = \begin{bmatrix} f_1(x,y,Z) \\ f_2(x,y,Z)\end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ V_z \\ \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix} \tag{2}$$

where $f_1 = \begin{bmatrix}\frac{-1}{Z} & 0 & \frac{x}{Z} & xy & (-1+x^2) & y\end{bmatrix}$ and $f_2 = \begin{bmatrix}0 & \frac{-1}{Z} & \frac{y}{Z} & (-1+y^2) & -xy & -x\end{bmatrix}$ These velocities are finally integrated to get 6 DOF pose as seen in figure 3.

IV. Discussion and Conclusion

A. KLT optical flow

Optical flow estimation using KLT tracker is prone to drift due to abrupt illumination change. To counter this, the features are tracked for a couple of frames (5 frames in our case as seen in figure 2) and then reset. After the reset new features are extracted from next frame and these new features are tracked.

B. Disparity

Since disparity is inversely proportional to depth, small values of disparity results in insanely large values of depth. To rectify this, mean of the disparity values is computed for those pixels where the disparity is greater than a certain threshold (threshold=0.001)

V. Outputs and Results

Here is the video output with feature detection in both left and right stereo images and trajectory of the quadrotor. StereoVO-rviz.mp4

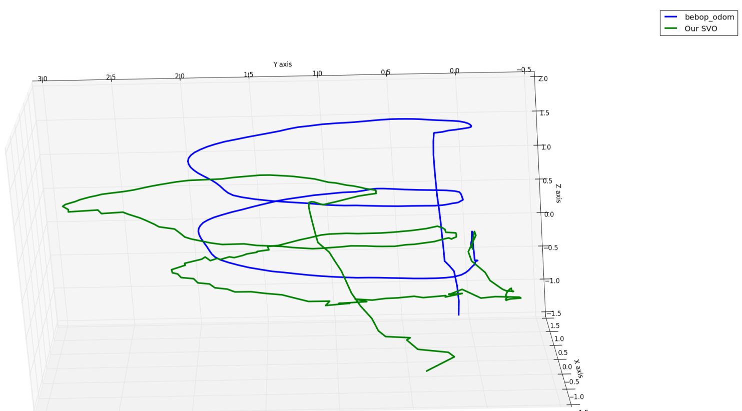

Figure 4: Final Helix trajectory plot. Blue is bebop's odometry data and green is pose from SVO