Introduction

For UAVs to determine their location with respect to the environment they fly in, the use of a stereo camera along with mathematical calculations can be implemented. The purpose of stereo cameras is to capture depth information whereby each camera can provide two accurate frame measurements. Combining the two cameras provides 4 measurements for 3 axis and thus allows for the UAV to see depth from lens to object within the image. To get the final pose estimation data which is the motion of the UAV by tracking the surroundings frame by frame, each camera image is first de-warped, and rectification is complete. This is where the image corrected from the resulting fish-eye effect from the lens. Then points within the image are detected using the SIFT feature detection method. Next is where the two images from the right and left come together and basing the key point from each, image are aligned, the difference in distance can be obtained to provide depth information. Lastly using a previous frame and comparing to the current frame, the shift between frames provides a distance measurement which can be used along with time to determine velocity change caused by the motion of either the camera or the object used for key point detection called Euclidean distance feature matching.

Equation 1 gives the position reference of a particular point of interest to a frame b based on an intermediate frame a, whereby the transformation matrix is used to convert the measured distance from frame a location to b. In this case P^j_a is a point of the previous frame and P^j_b is the current frame.

$$P_b^j = C_{ba} P_a^j + r_b^{ab} \tag{1}$$

With the position measurements completed for us, our focus turns to outlier rejection, particularly with a method called RANSAC (Random sample and consensus). Next the use of RANSAC takes 3 arbitrary points and classifies the rest of the points as inliers or outliers. To compute model, estimate rigid body transformation using SVD and point cloud alignment. The model is used to predict points in the current frame and classify which points in the previous frame were inliers vs. outliers. The outlier rejection process is a method used to detect and exclude data points that don’t conform to an expected pattern. Outliers impact computer vision algorithms by skewing results and reducing accuracy, thereby necessitating a rejection outlier technique to improve the quality of data used for object recognition and motion tracking, as well as improving algorithms to be less susceptible to outliers. RANSAC is an iterative algorithm used to estimate model

parameters from data that contains outliers. RANSAC works by selecting small random subsets of a given set of data and fitting a desired model (e.g., a line) to them. The data points that conform to the underlying model, are then categorized as inliers, while those with significant deviations are categorized as outliers. This process continues until a model with sufficient inliers is established, providing a better approximation that is less influenced by outliers.

$$P_a = \frac{1}{w} \sum_{j=1}^{J} w^j P_a^j \tag{2}$$

$$P_b = \frac{1}{w} \sum_{j=1}^{J} w^j P_b^j \tag{3}$$

$$w = \sum_{j=1}^{J} w^j \tag{4}$$

Now with the filtered point cloud data, the centers of each point cloud are found P_a & P_b and used to construct the W_{ba}, cross-covariance matrix. Equation 4 finds the summed scalar weight for all points and used with conjunction of the scalar weights of each point “j”, previous and next frame points can be determined shown with equation 2 and 3.

$$W_{ba} = \frac{1}{w} \sum_{j=1}^{J} w^j \left(P_b^j - P_b\right)\left(P_a^j - P_a\right)^T \tag{5}$$

Equation 5 solves for W_{ba} that helps to find the errors between the centers of the point cloud and the points. It is important to note that W_{ba} needs to remain singular throughout the calculations. Once we have cross-covariance matrix W_{ba} we can apply SVD and compute the unknown rotation matrix.

$$VSU^T = W_{ba} \tag{6}$$

$$U^T U = V^T V = 1 \tag{7}$$

By applying equations 6 and 7, rearranging for U and V, the determinants can be found to construct the transformation matrix C_{ba} seen from the very first equation.

$$C_{ba} = V \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & detUdetV \end{bmatrix} U^T \tag{8}$$

Once the transformation matrix is determined, it can be used to find the unknown translational matrix r_a^{ba}.

$$r_a^{ba} = -C_{ba}^T P_b + P_a \tag{9}$$

$$T_{ba} = \begin{bmatrix} C_{ba} & -C_{ba} r_b^{ab} \\ 0^T & 1 \end{bmatrix} \tag{10}$$

Lastly, the pose T_{ba} is found from the use of both C_{ba} found with equation 8 and r_b^{ab} found at the very beginning with respect to equation 1.

Results

For the code, both previous and current frames are inputted into the pose estimation function which are first converted into point cloud data. Next RANSAC filters the features using at least 3 points required. For the best results from turning the parameters was at 200 iterations and 5 Euclidean points. We could achieve similar results in the same time even by increasing the number of iterations. To find the centroid, the mean of the drawn points is used for both previous and current frames.

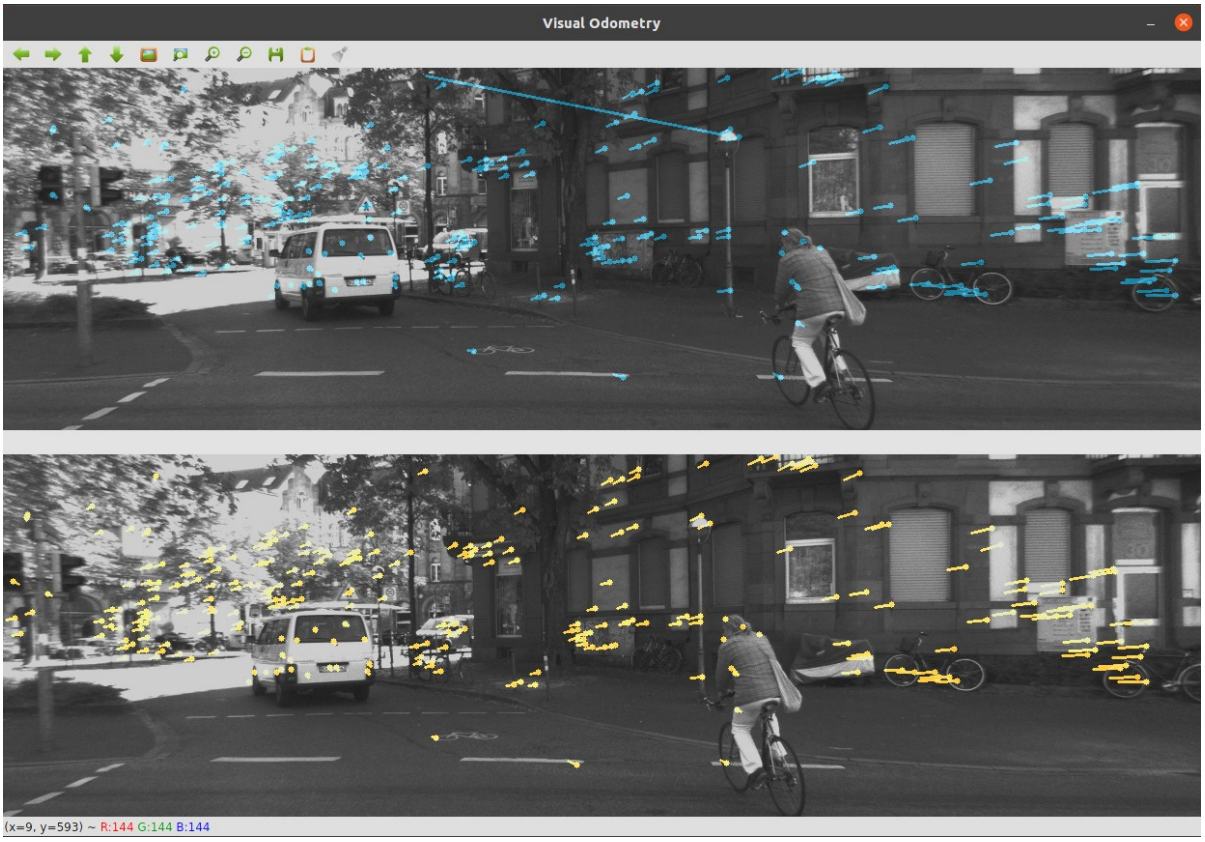

Figure 1: Unfiltered feature correspondences (top) vs. filtered feature correspondence (bottom) using RANSAC

Having access now to the transformation matrix means that we know how the camera is moving with respect to the objects around it. Thus, the translation and rigid body transform is completed based on equation 10. These values need to be converted to homogenous coordinates and ultimately returned for the ground truth comparison seen in figure 2.

As shown in figure 2, the resulting visual odometry trajectory does follow the ground truth at a satisfactory rate but could be improved. It is believed that the reason for the error is due to the accuracy of the cameras' information. RANSAC, being better than without, is only able to provide a result based on the majority of what the data does at a given time (using the inliers and excluding the outliers of a given dataset).

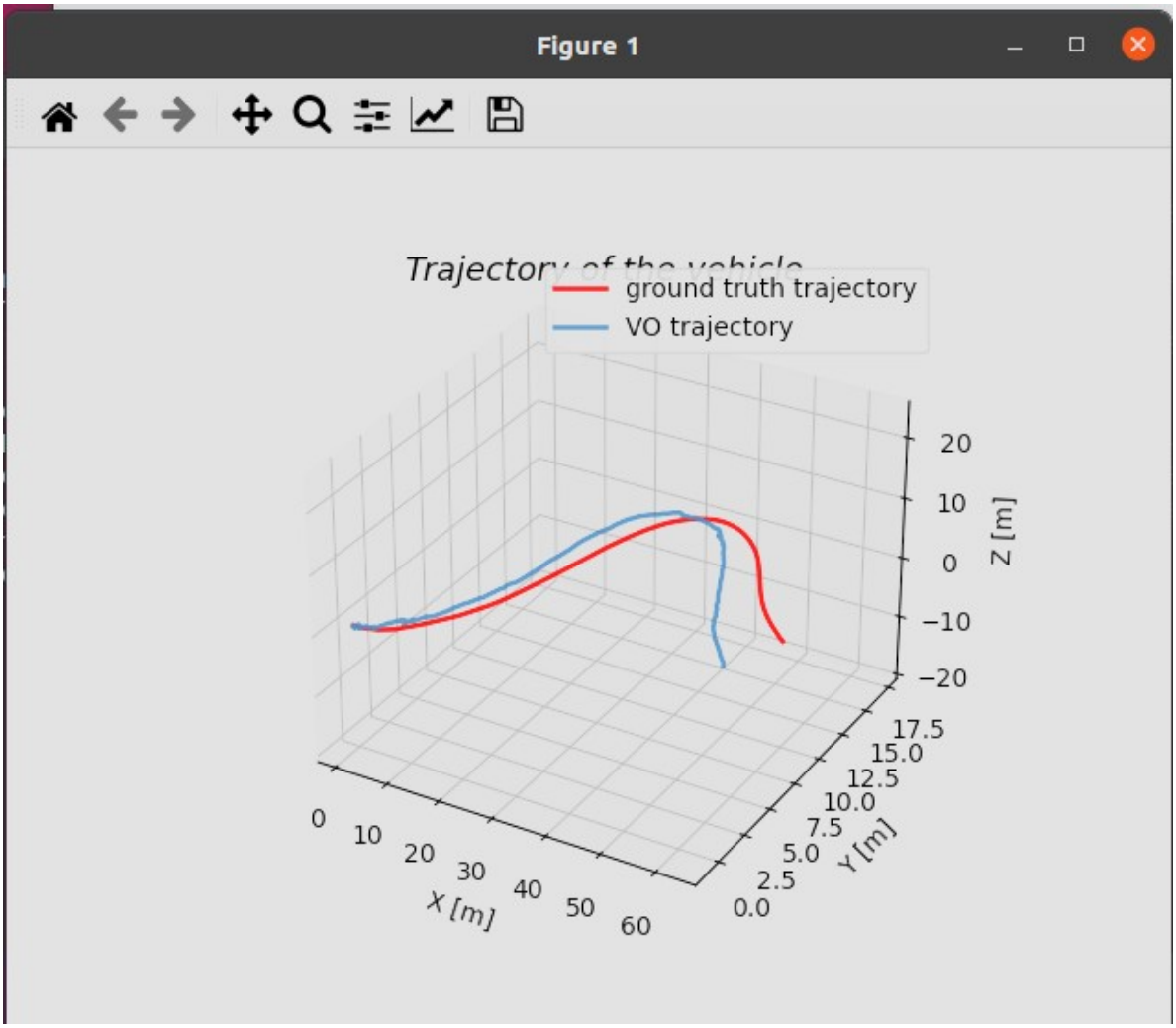

Figure 2: Comparison of Visual Odometry vs. Ground Truth Trajectory

The issue with this, is that even though most of the inliers are relatively accurate, there could still be error overall in the image which RANSAC would not be able to capture due to all inliers having the same error amount. This error overtime could add up and result in a larger displacement away from the ground truth data seen in figure 2. Methods to improve on trajectory accuracy is to implement more accessible datapoints to compare to such as using a secondary stereo camera. Since all measurement devices inherently have error, having measurements from different tools would ensure RANSAC to see the intrinsic error in one camera, which will give RANSAC a better picture of what is truly suppose to be considered an inlier or an outlier.

Another important point to note here is, because the RANSAC uses random points for Visual Odometry. Thus, we get different plots for the same values of iterations and the residual threshold (distance for separating inliers.)