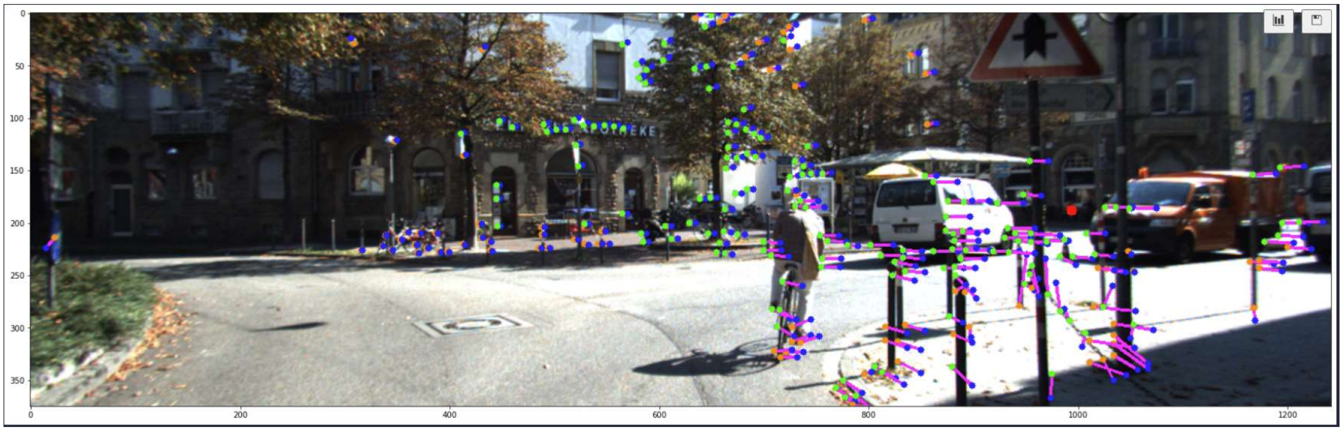
Write up

In terms of corner detection design choice, I decided to go with using Shi-Tomasi corner detection algorithm for finding good feature points and using the parameters for LKT optical flow detection as exemplified by https://docs.opencv.org/3.4/d4/dee/tutorial_optical_flow.html as this was the most guided example to follow. I decided to go with 6000 max corners to mine since the more corners, the better. Blue points represent optical flow corner points of image 2, green represents points of image 1 that are considered inliers after RANSAC, orange points are image 1 outlier points, and the pink lines between points are the motion vector

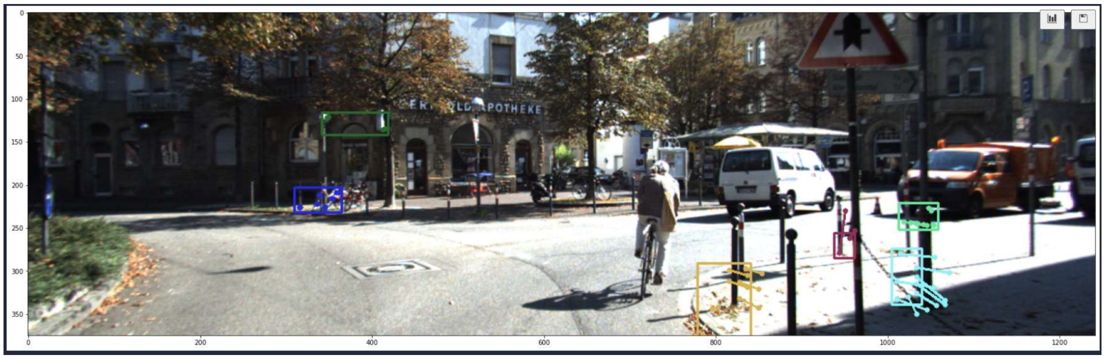
In order to limit the number of faulty motion estimation lines for independent moving objects, I adjusted the RANSAC tau value to 110 as I found this to be the most optimal value.

With tau=50 we get the following result. As we can see, the FOE bears more to the right hand side of the image to support the camera's forward motion causing keypoints to move outward and rightward in this case. But we see that there are still quite a few outliers being generated in the bike stand, poles, and even the ground. The result after the below one is the result of the clustering algorithm being run. It is classifying collections of stationary points on the ground to be moving objects.

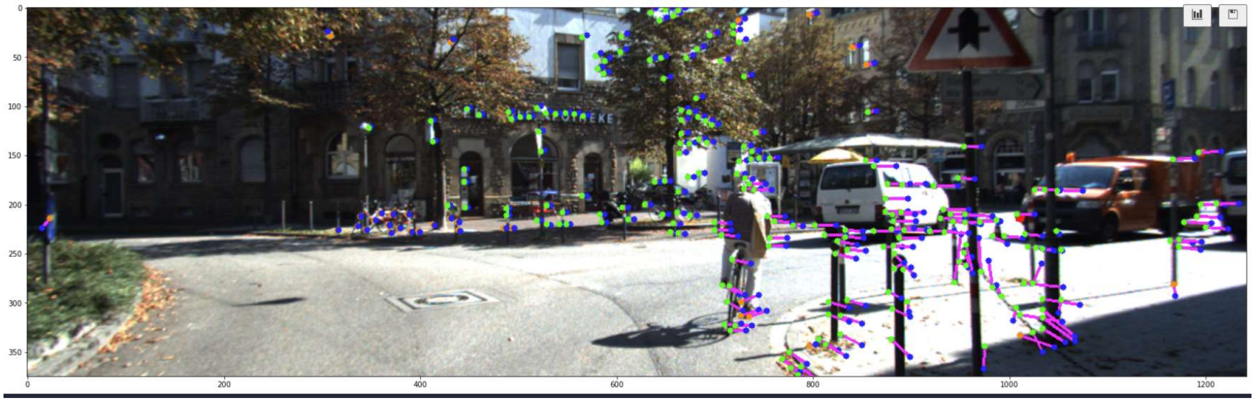
Tau=50

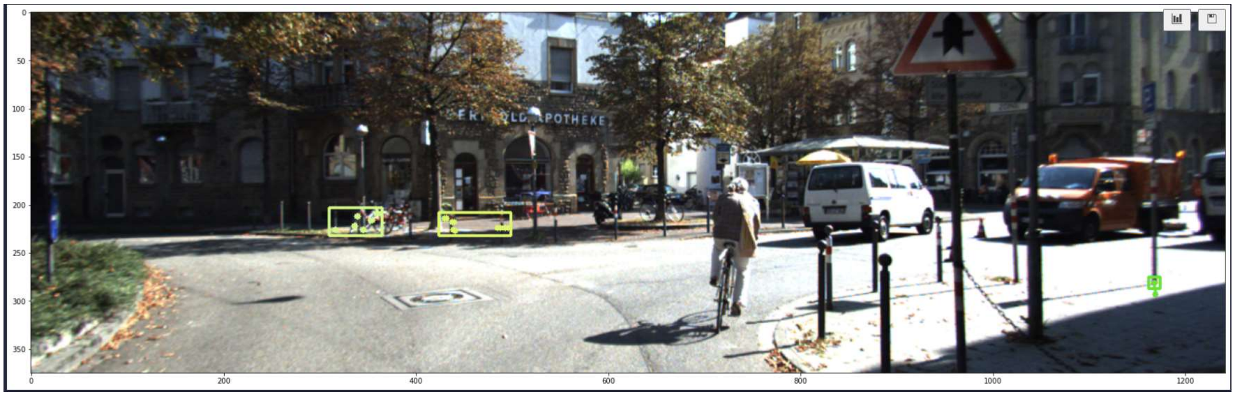


Tau=50 and clustering



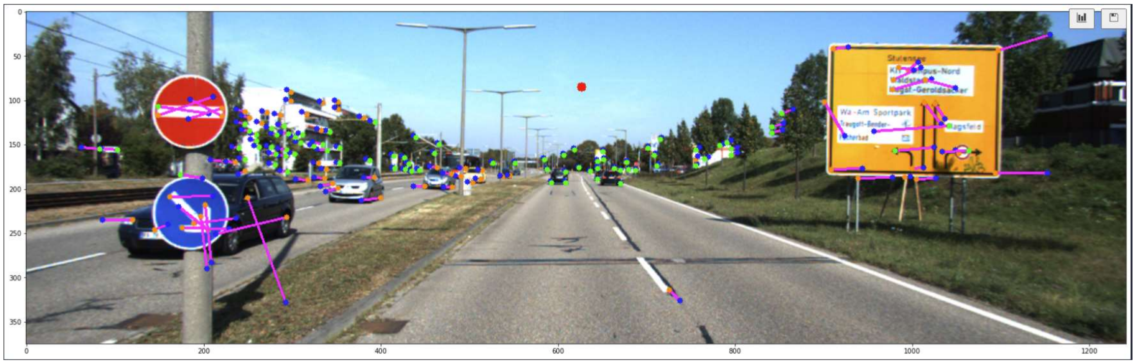
The next set of results are produced from using a tau value of 110, which produces more inlier points that now capture those previous ground outlier points even though the bike stand in the background is still has many outlier points. This might be due to the extremely small variations in images and their lines not being in range of the FOE.

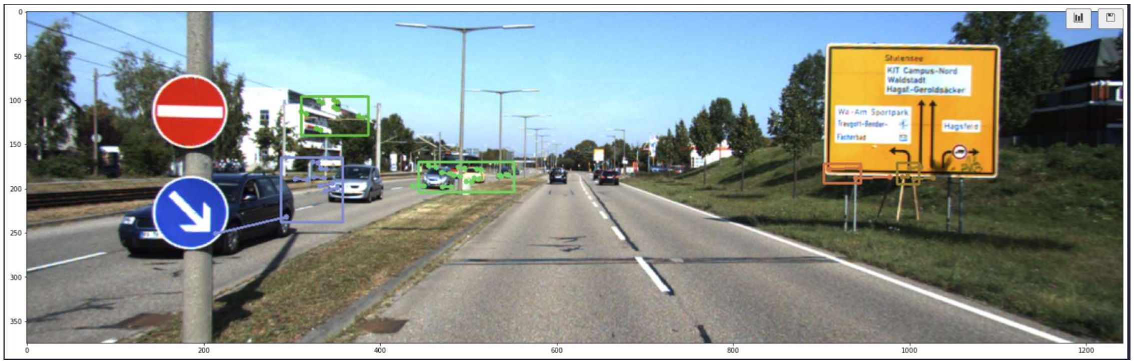




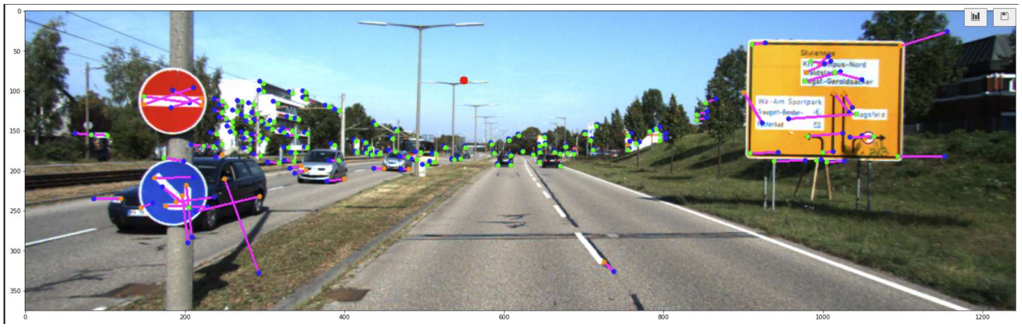
As part of my clustering algorithm, I used Agglomerative Clustering with a clustering threshold of 100 with a cluster filtering method that filters clusters that have 4 or less points, since my selection of corner detection algorithm and selection of tau were capturing too many very small variations in images in random areas.

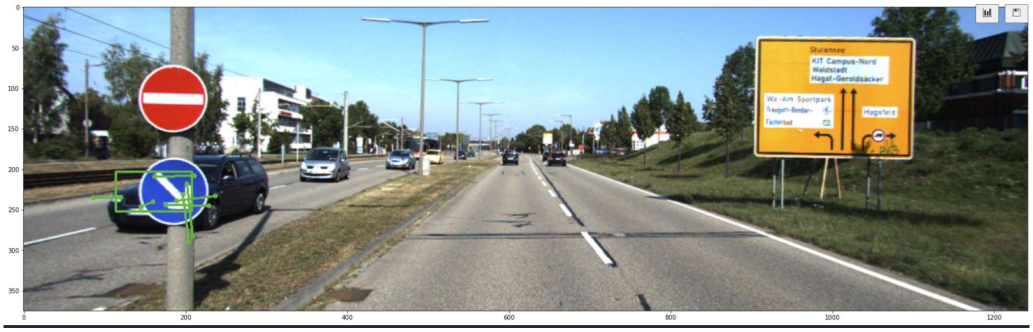
Failure case: This is an example where using tau 50 and keeping the clustering algorithm along with cluster filtering method the same achieves a more accurate result to capture moving cars and also detect that the camera is moving. Even though the white building is classified as a moving object, 2 cars and a section of the gray car are captured as moving objects.





Below result is from setting tau to 110 which filters too many independent moving objects points and misclassifies the blue sign as a moving object.





In the future I think it will be beneficial to implement some kind of filtering method that gets rid of optical flow point corners between images that meet a very small threshold, so that the proceeding steps are not capturing too many outliers that just happen to point to different directions and cause the clustering algorithm to group too many outlier points