OVERVIEW

Andrew Showers

The goal of this program is to read in two source images, determine if the camera is stationary or moving, determine which keypoints are independently moving, and which independently moving objects can be clustered together. The first step was to read in the images in grayscale. Inspired by the OpenCV documentation[1], I used ShiTomasi corner detection (cv2.goodFeaturesToTrack) in order to gather keypoints from the first image. Then I applied Lucas-Kanade Optical Flow (cv2.calcOpticalFlowPyrLK) in order to perform descriptor matching and estimate the motion for the keypoints.

To determine if the camera was moving, I took each keypoint pair and measured the distance between the points. If the pair distance was greater than 1, I considered it a moving pair. If 50% of the keypoint pairs were moving, then I labeled the camera as moving. If it was less than 50%, I labeled the camera as stationary. I then filtered points which did not pass the fundamental estimation.

If the camera was moving, I would find the Focus of Expansion point by using the epilines and finding the intersection of these lines which minimized error (via np.linalg.lstsq). I would then use this center point to determine which keypoint pairs are moving independently when the camera is moving. This was done by first finding the line between the FoE points in image 1 and 2, and then finding the line between the keypoint pair in image 1 and image 2. The a and b coefficients should be roughly the same if the object is not independently moving. I allowed a max difference between the coefficients to be a total of 10 apart before determining the object is independently moving.

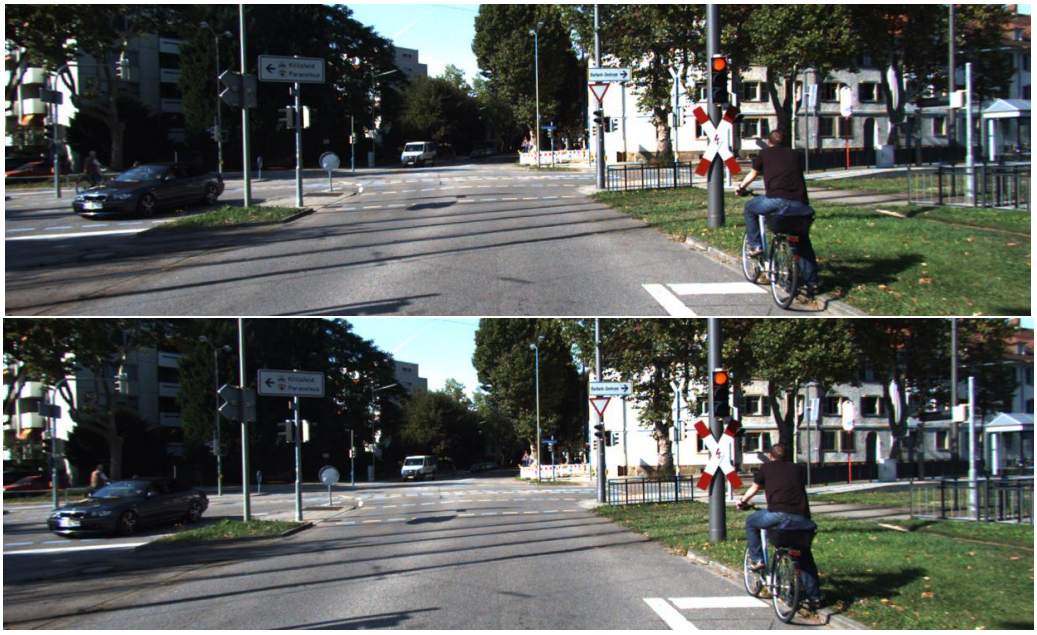
For images in which the camera was not moving, the process in determining independently moving points was much easier. This is because any significant distance between the keypoint locations in a given pair would mean the object is independently moving. As such, I marked all keypoints with a distance greater than 1 pixel as independently moving. I used this small distance since there shouldn't be any significant change given the camera is stationary.

Lastly, clusters were made for independently moving objects. This was done by grouping all keypoints that could be connected by a distance of less than 100 pixels. The groups could be much larger than a 100 pixel span, but would have to contain links that never exceed a length of 100. I found that using a 100 pixel span was ideal as it would give a reasonable number of clusters when necessary while still connecting large objects such as a bus.

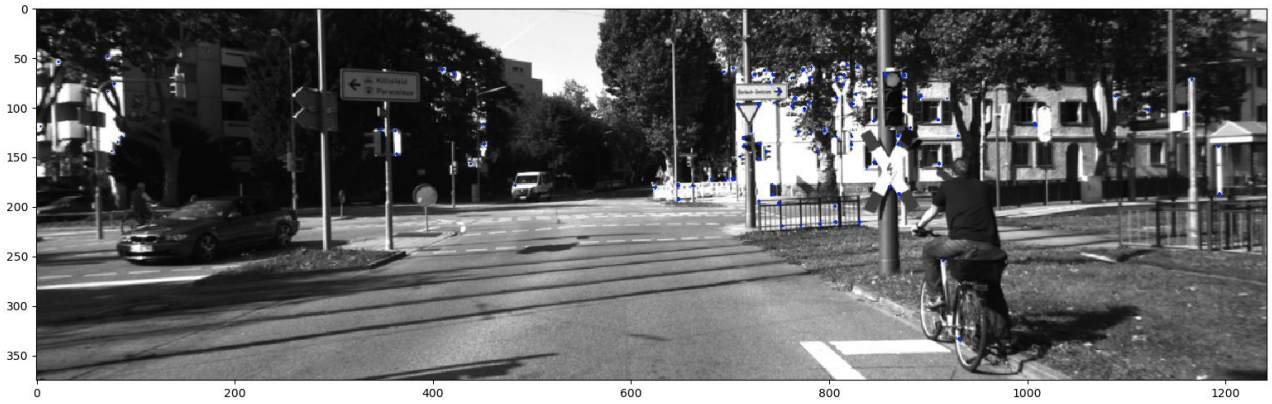
The results varied in performance. If the image had good contrast between moving objects and non-moving objects, the algorithm worked well. However, sometimes the edges would blend together causing the program to classify points as moving when they were in fact stationary (e.g. bike example with decorative sidewalk). Additionally, the results were more accurate with a stable camera. If the camera was moving, it was difficult to tell independently moving objects given how much error can exist in the motion calculations. Also if objects are moving in the same general direction, such as a highway, it can be difficult to classify them as independently moving since their movement is in the same direction as the camera. Under ideal circumstances, the independently moving object is moving in a vastly different direction than the camera.

Results shown on the following page

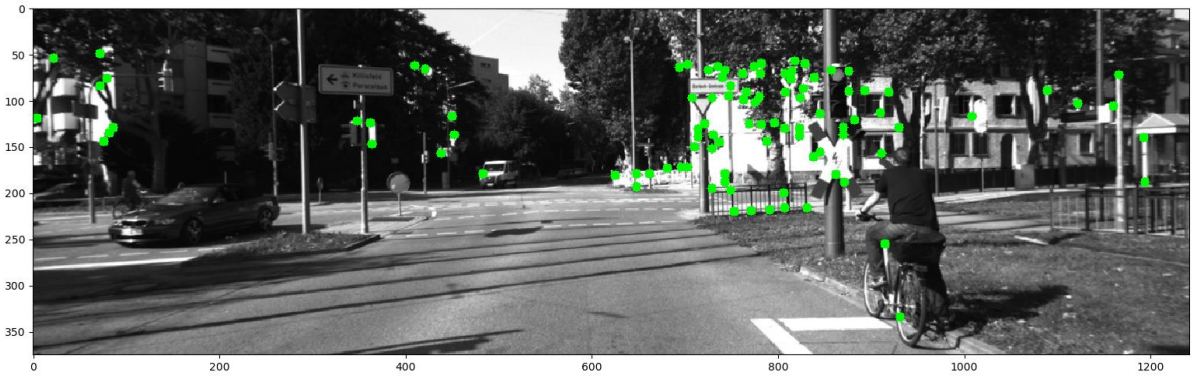
[1] http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_video/py_lucas_kanade/py_lucas_kanade.html



Original images



Motion vectors in blue

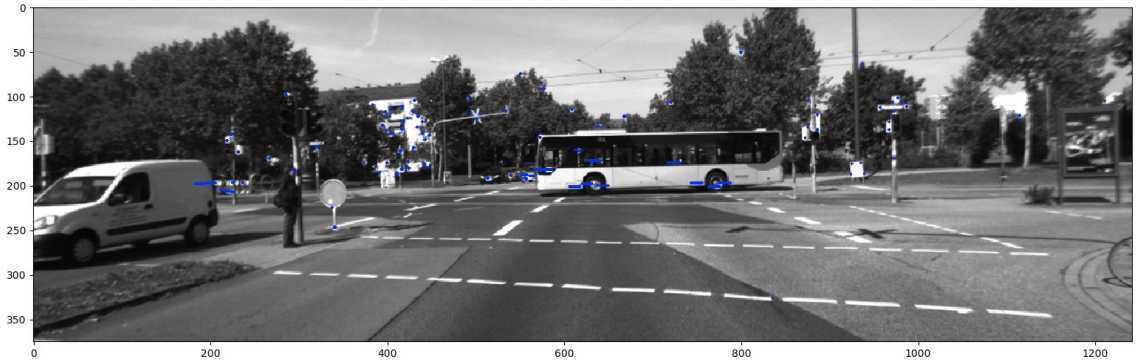


Independently moving points (red) and stationary points (green)

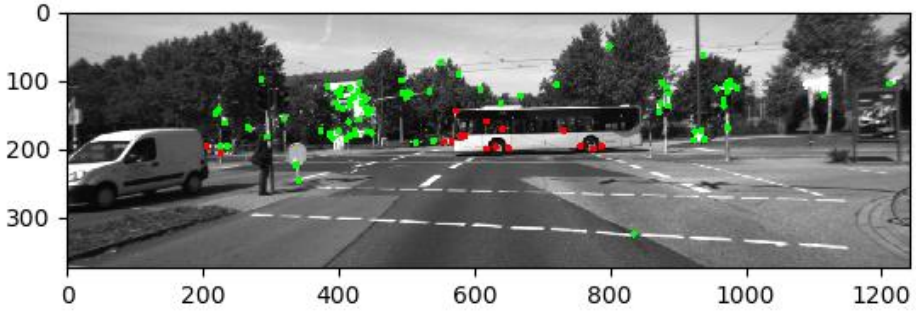
Above we can see two images which had very little movement. This was accurately detected as a stationary camera with all keypoints not moving. Since there were no independently moving keypoints, there was nothing to cluster.

Below is an example of a stationary camera which also detected moving objects:

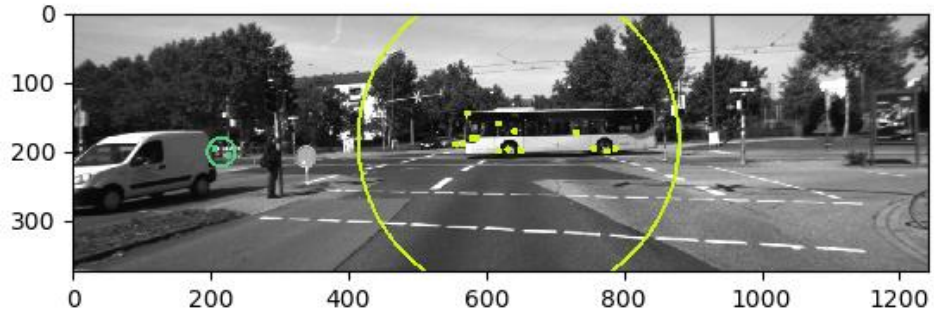
Original images



Motion vectors in blue



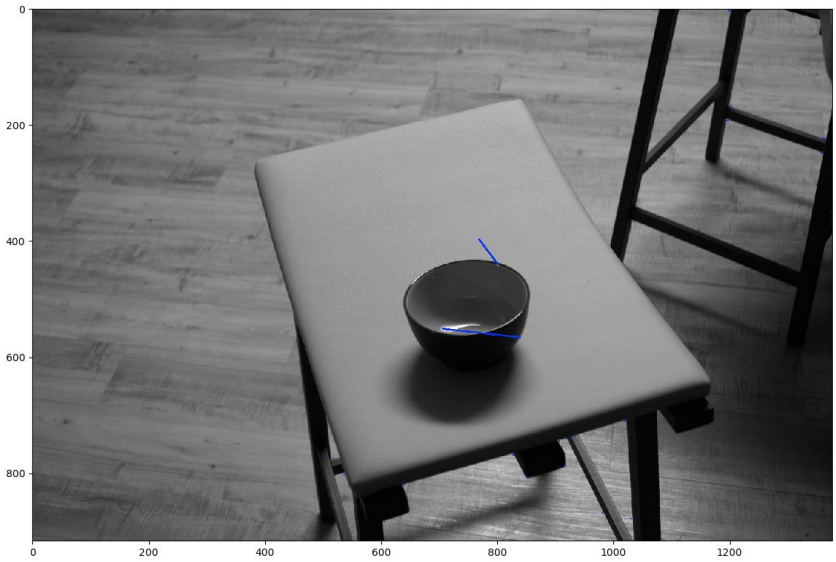
Independently moving points (red) and stationary points (green)



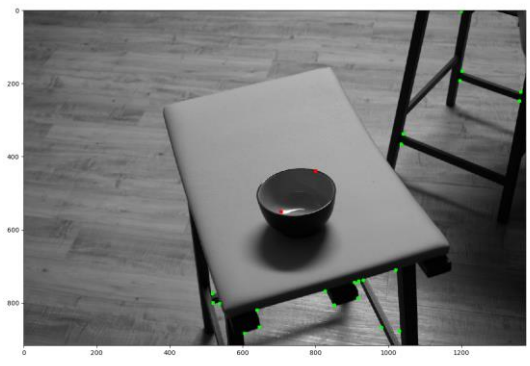
Clusters of moving objects (color coded)

The bus example was able to correctly identify that the camera was stationary, the keypoints which are stationary and which are moving, and clustered them appropriately. This example was difficult for clustering the bus since it was a large object, and required some fine-tuning to get a max keypoint distance to work with this example while still performing well on others. The green keypoints may be erroneously labeled as moving, though it is difficult to tell given they occur near objects that are moving and shadows corresponding with the objects.

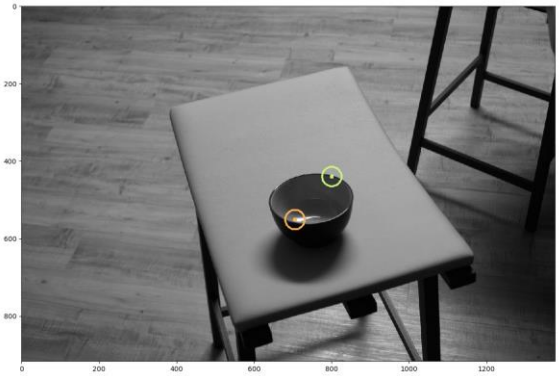

Original images



Motion vectors in blue



Independently moving points (red) and stationary points (green)

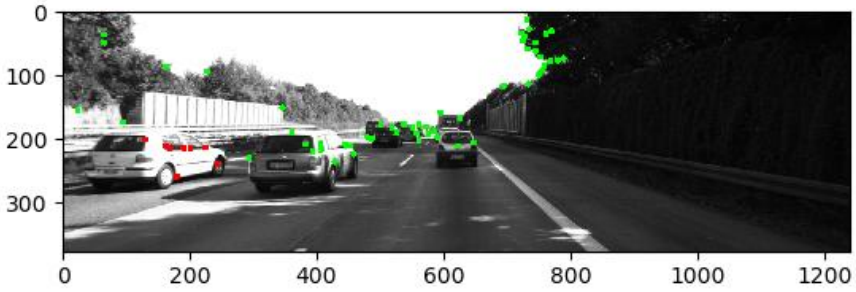


Clusters of moving objects (color coded)

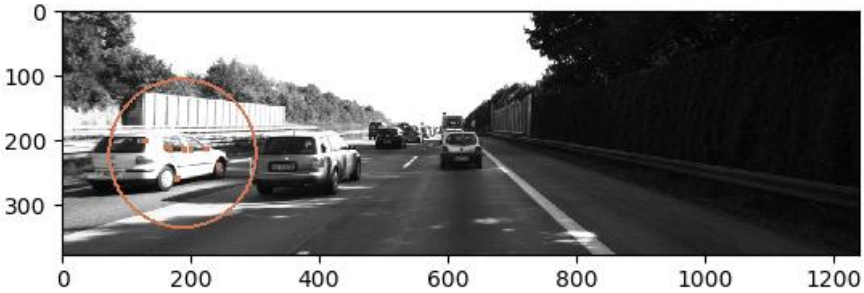
Above we can see two pictures taken of a bowl, moved from one spot to another. The program correctly identified the keypoints which are moving, however, was unable to put them in the same cluster. They just passed the 100 pixel max distance. It would be more reasonable to use a percentage of pixels instead of a constant 100 in order to handle images with differing resolutions.

Original images

Focus of Expansion point in red and motion vectors in blue



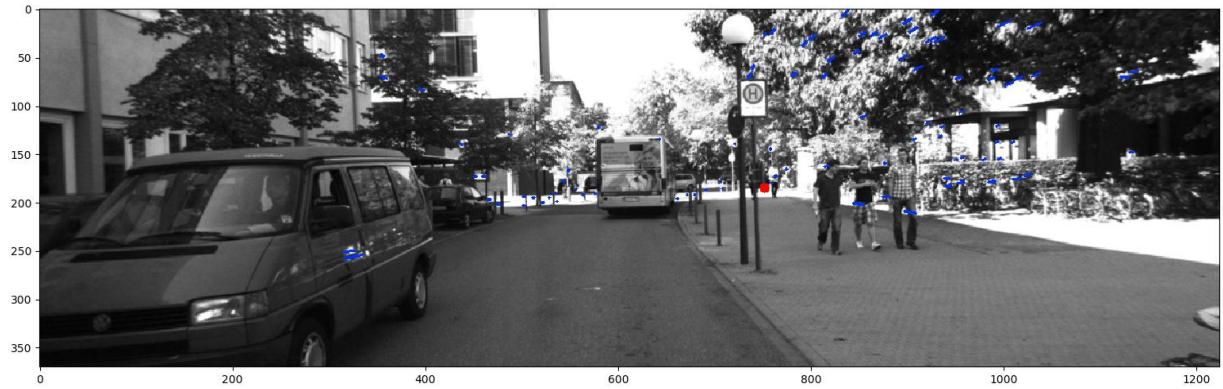
Independently moving points (red) and stationary points (green)



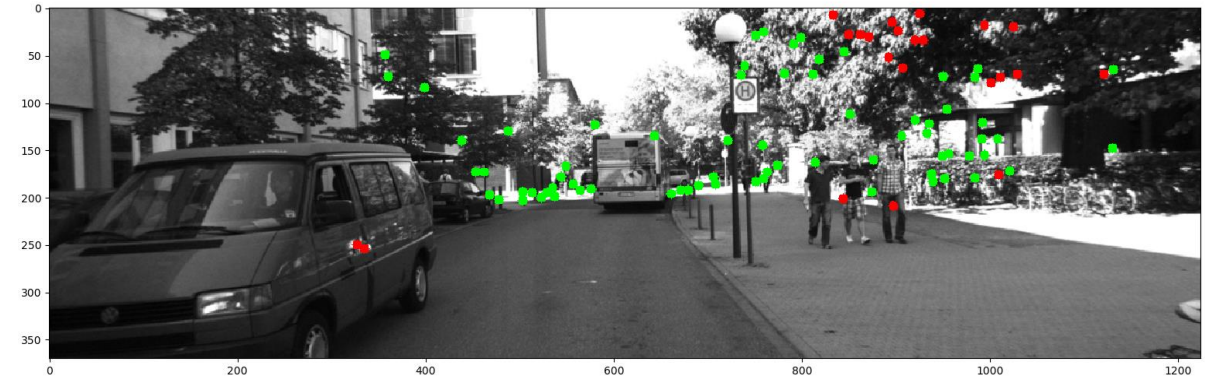
Clusters of moving objects (color coded)

In the example on the previous page, we can see that the focus of expansion was detected and marked in red. Furthermore, the algorithm correctly identified the white moving car. While all the cars were moving, most cars were driving roughly the same speed and direction as the car with the camera. For this reason, they were considered stationary while the white car was considered independently moving.

Original images



Focus of Expansion point in red and motion vectors in blue

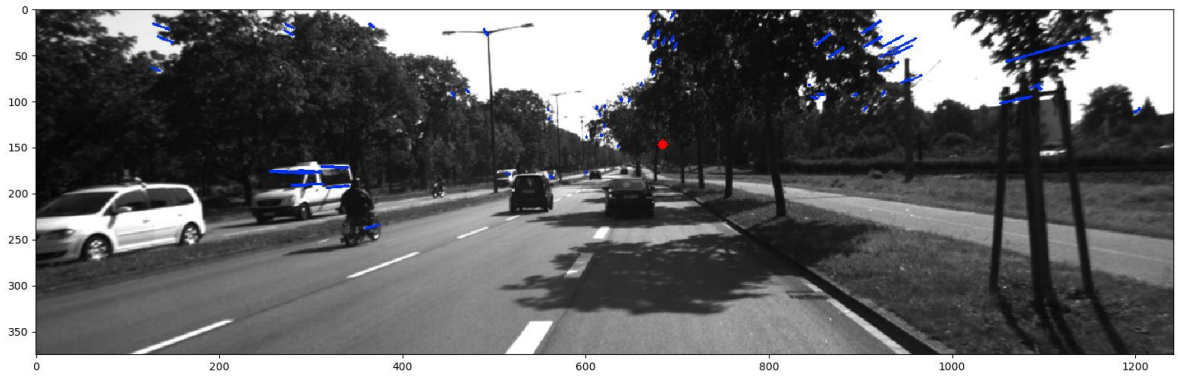


Independently moving points (red) and stationary points (green)

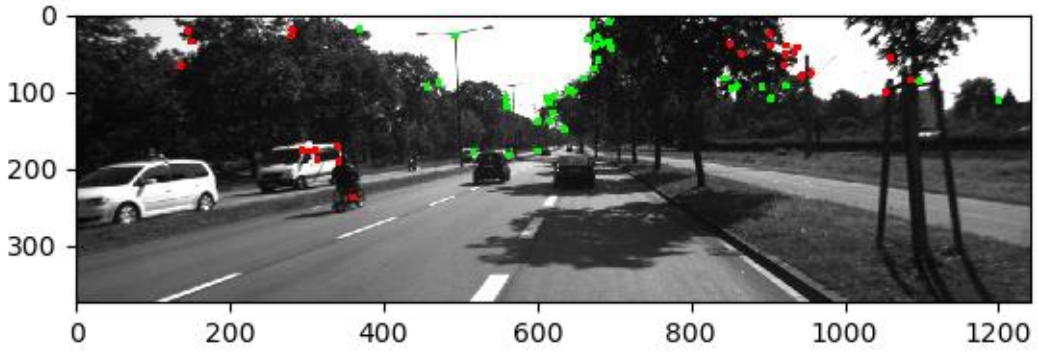
Clusters of moving objects (color coded)

In the example above, the program correctly identified the camera as moving and marked the car and pedestrians as moving objects. However, the leaves in the tree were also marked, possibly due to motion from the wind, or possibly due to error in descriptor matching. The clustering for this problem also performed well by segmenting the car, pedestrians, and leaves into separate clusters.

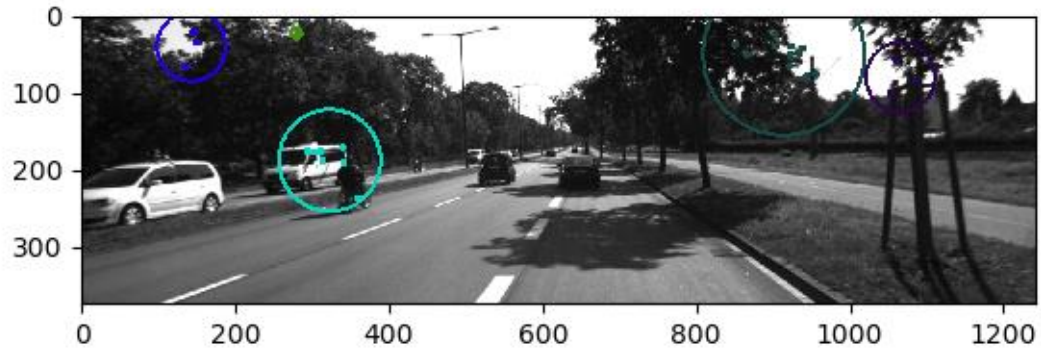
Original images



Focus of Expansion point in red and motion vectors in blue



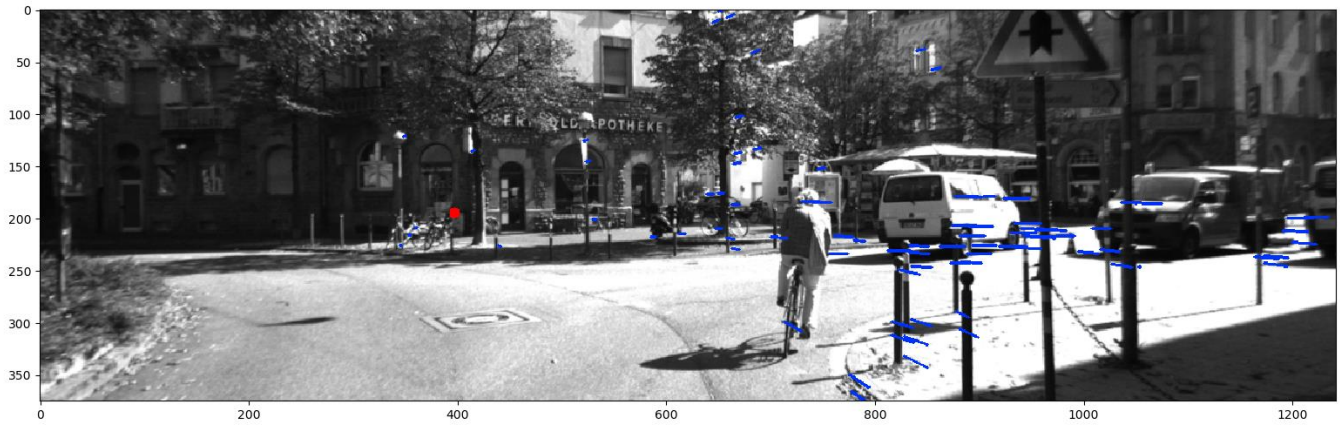
Independently moving points (red) and stationary points (green)



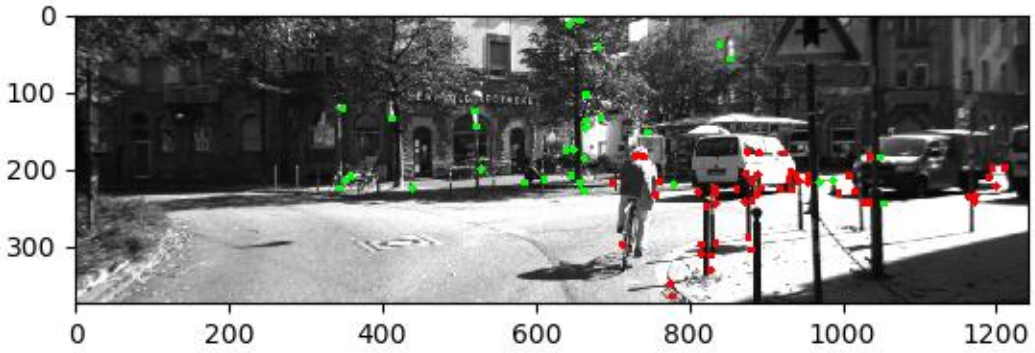
Clusters of moving objects (color coded)

In this example, the camera is detected as moving and the car, motorcycle, and leaves are marked as moving objects. Similar to the previous example, it's difficult to determine if the leaves moved due to wind or if this was due to error. The clustering for this image was not ideal as the car and motorcycle were clustered together due to their close proximity.

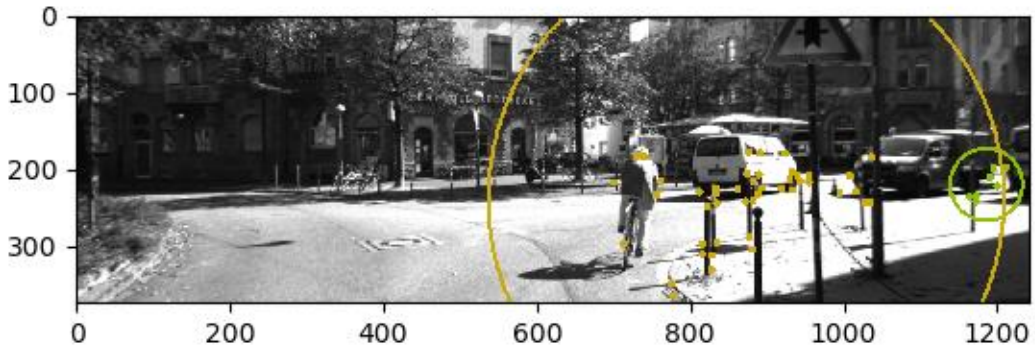
Original images



Focus of Expansion point in red and motion vectors in blue



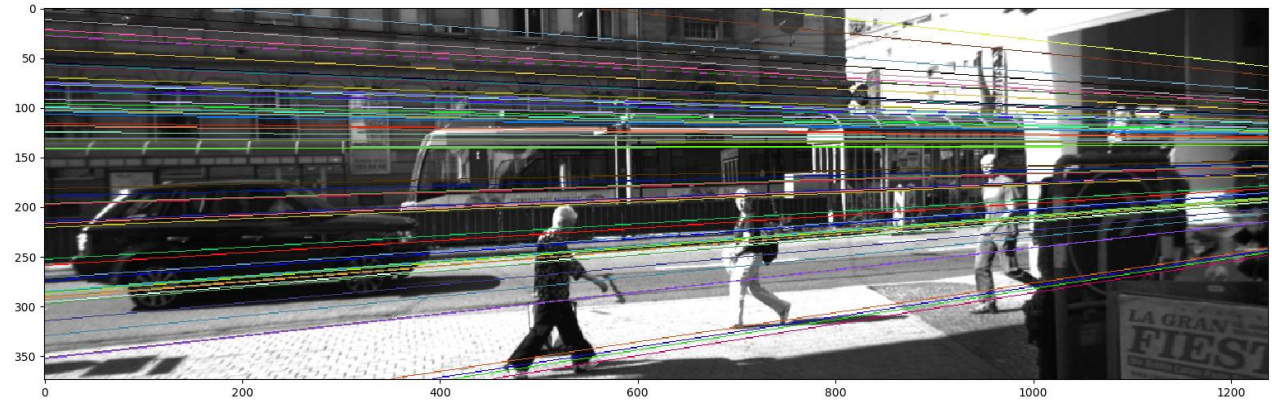
Independently moving points (red) and stationary points (green)



Clusters of moving objects (color coded)

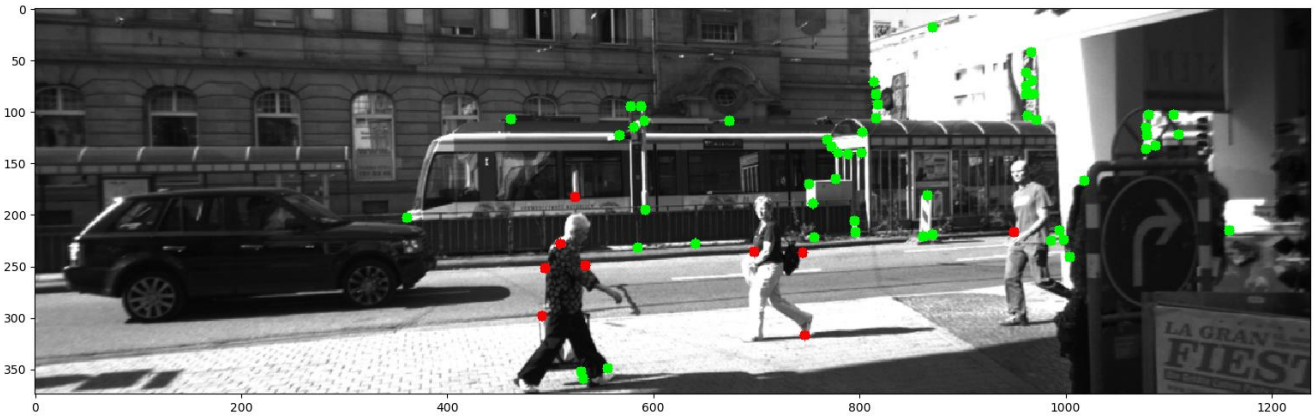
This example was one of the most difficult ones to work with due to the keypoints near sidewalk decorations. Furthermore, the clustering linked the man, van, and sidewalk decorations all in the same cluster. One possible way to address this would be to introduce the direction of motion as a factor in clustering.

Original images

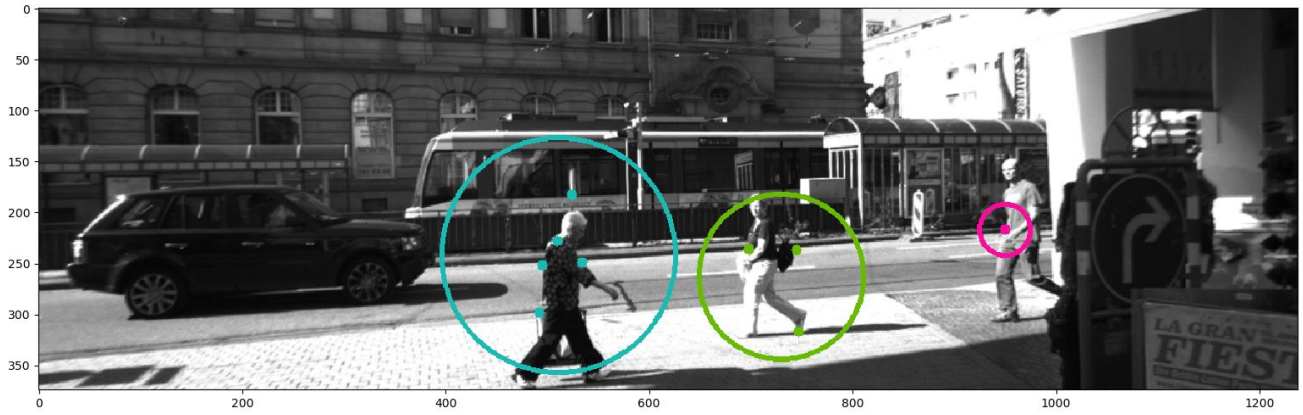


Epilines - Focus of Expansion point is outside the image

Here we can see an issue where the Epilines placed the Focus of Expansion point outside of the image. This is because the camera isn't actually moving. The percentage moving was marked as 52%, slightly above the 50% threshold. Adjusting the threshold to be slightly higher addresses this issue.



Independently moving points (red) and stationary points (green)



Clusters of moving objects (color coded)

The clustering for this image performed very well. It is concerning that some of the keypoints on the left-most pedestrian was marked as stationary, but the majority of the points were considered moving. Overall this image performed well aside from the percentage of moving keypoints being significantly close to the threshold.