CIS 581 Project 4.5 Report

Keyan Zhai Zhi Zheng

1 Required

1.1 Show bounding box and features in the first frame (for getFeatures): 10 pts

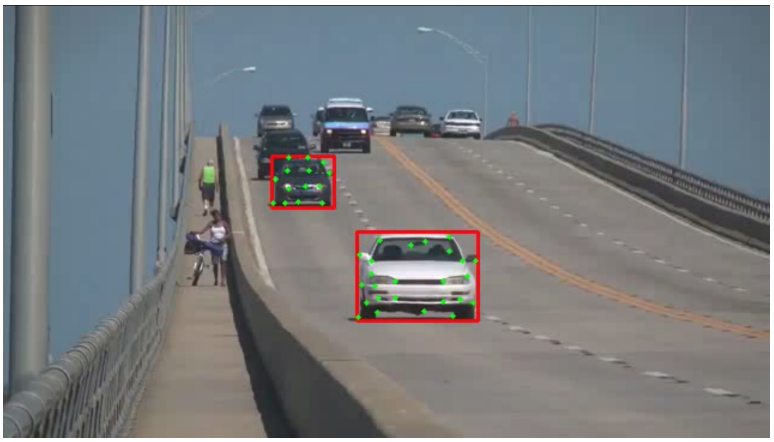

Figure 1: Bounding boxes and features in frame 1

For `getFeatures`, we used the Shi-Tomasi feature detection (`cv2.goodFeaturesToTrack` method). Given the input image and the bounding box, we need to detect the feature points inside the bounding box's range. We took into account the case of multiple objects. We set the maximum number of corners as 100, the minimum quality of corner as 0.03 below which everyone is rejected, and throws away all the nearby corners in the range of minimum distance 7.

1.2 Show features in two frames with noticeable translation of tracked object (for estimateFeatureTranslation): 30 pts

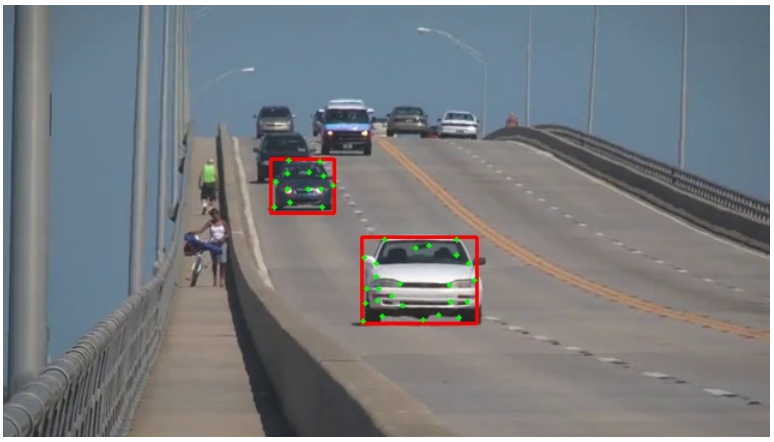

Figure 2: Frame 09

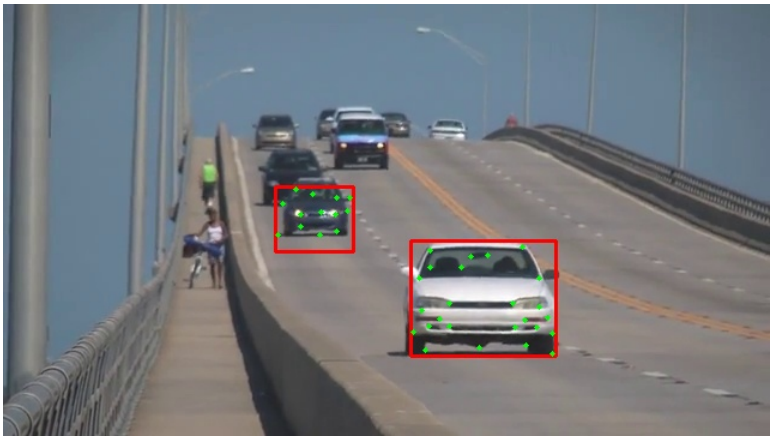

Figure 3: Frame 59

The above figures show the features in frame 09 and frame 59 with noticeable translation of tracked object. For `estimateFeatureTranslation`, we set the Gaussian kernel size as 9, and the Gaussian standard deviation sigma as 1.3. Then we just followed the Kanade-Lucas-Tomasi tracking procedure to track the features. We selected the window with size 15 for each feature point for estimating the translation.

1.3 Show bounding box in two frames with noticeable translation of tracked object (for applyGeometricTransformation): 20 pts

Figure 2 and Figure 3 also show the bounding box in frame 09 and frame 59 with noticeable translation of tracked object. In `applyGeometricTransformation`, we first estimated the transformation matrix between features of the previous frame and the current frame. Then we used this matrix to calculate the new bounding box. Finally we eliminated the outliers in the new features. If a feature point moves more than the threshold (set to 2.5), then eliminate this feature because it probably failed in tracking. If a feature point is outside of the new calculated bounding box, then eliminate this feature.

1.4 Show the case when there are not enough feature points in one frame and you generate enough feature points in the next frame. List a few reasons why features are lost and discuss your strategy to detect and handle this situation. 20 pts

Figure 4 and Figure 5 show the case when there are not enough feature points and we generate enough feature points. One of the reasons why features are lost is that we elminated outliers in `applyGeometricTransformation`. Feature points are lost because they move too much or exceed the new bounding box. For each new frame, we check if there are enough valid feature points for an object. If the number of feature points is less than some threshold (set to 5). we use `getFeatures` to re-detect feature points.

Figure 4: Frame 109

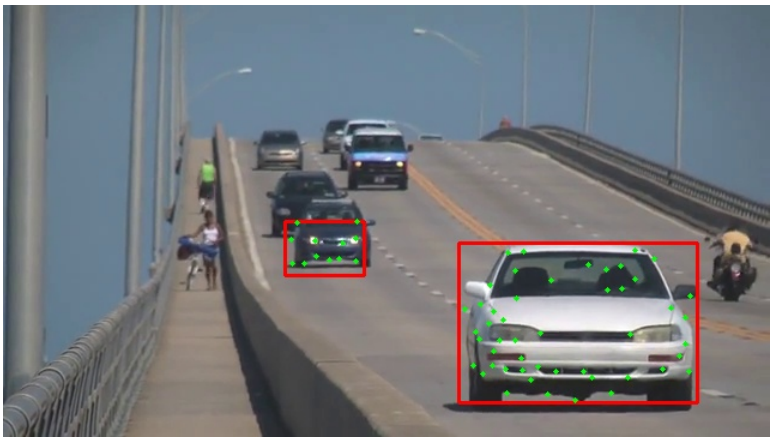

Figure 5: Frame 119

1.5 Result video quality(for objectTracking): 20 pts

In `objectTracking`, we detect the feature points in the first frame, then we use optical flow to track these feature points in the following frames. If there are not enough feature points, we re-detect the feature points. Generally the quality of the output video is good. In the first half, the feature points and the bounding box can track the object very well. However, We don't know how to deal with the feature points and the bounding box when the object is moving out of the image.

2 Extra

2.1 Implement pyramidal KLT tracker and briefly compare it with normal KLT tracker: 20 pts

We did not implement pyramidal KLT tracker.

2.2 Able to track multiple objects: 20 pts

We implemented tracking multiple objects. We just looped for F times and performed the same operation for each object in each function.