COMP9517 Assignment 2 Report

Jingshi Yang z5110579

Objectives:

Familiar with techniques and implementation of feature detection and matching in OpenCV.

Summary:

In this assignment, I implement object detection by surf or orb descriptor, and based on these detected features, implement object tracking by using MultiTracker and for each object using KCF tracker.

During tracking, for each object there is one bounding rectangle (each rectangle has one different color) to display the estimated location and one tracing line to describe the path of the motion of the object (each line has one different color).

Details:

1. Use parser to pass argument, input video is one command argument.

2. Use cv2 built-in function to create orb or surf detector. ORB is an efficient alternative o SIFT or SURF, full name is Oriented FAST and Rotated BRIEF, as the name, orb is basically a fusion of FAST keypoint detector and BRIEF descriptor with many modifications to enhance the performance.

The advantages of ORB are: 1. ORB is affected little by Gaussian noise by compared with SIFT; 2. ORB performance is superior to SIFT and SURF in outdoor environments.

How to apply ORB: First it uses FAST to find keypoints, then apply Harris corner measure to find top N points among them. Second ORB uses BRIEF descriptor, due to poor performance on rotation for BRIEF, so ORB improves this according to the orientation of keypoints.

3. Pre-process images. When extract features by using ORB, first I preprocess the image by cv2.cvtColor and cv2.GaussianBlur. The reason is that it can effectively remove noise and eliminate the influence by sunshine or other natural factors.

4. The function to select objects which we want to track is cv2.selectROI.

5. Tracker. Because we trace multiple objects, so first we use cv2.MultiTracker, then for each object, I use KCF tracker. The full name of KCF tracker is Kernel Correlation Filter,

this algorithm has good performance on tracking speed, but one big issue of KCF tracker is that it cannot solve the problem when the target is occluded during the tracking process.

6. Operating on gray images but display original images. Can get better matching output and images are not boring.

7. Tracing. For each object, we trace the object by appending center of each motion, because we know the corners of the object, we can get coordinate of center by simple calculation. Then after append the center with its corresponding cluster(object), remember to reshape it, the reason is that transform it to 2D array (center.reshape(-1, 1, 2)), here 1,2 means transform it to 2D, -1 means that it is unknown dimension and we want numpy to figure is out.

8. Draw Tracing lines. cv2.Polylines is used for drawing tracing lines, it takes 7 arguments, but some arguments are optional, like thickness, lineType, shift.

9. Draw Rectangle. cv2.rectangle is used for drawing rectangles, it takes 7 arguments, image, vertex of the rectangle pt1, and one vertex of the rectangle opposite to pt1 and color, thickness, lineType and shift are optional.

10. Draw matched points and connect them. I use drawMatchesKnn, cv2.drawMatches helps us to draw matches. It stacks two images horizontally and draw lines from first image to second image showing best matches. cv2.drawMatchesKnn draws all the k best matches.

Results: