

Final Project Report

CS510

Yu Yang

Moving Pedestrian and Vehicle Detection and Tracking

I tried both background subtraction and optical flow (commented out in the code) to implement motion detection.

Background Subtraction

Approach: Detect the foreground objects by comparing the difference between the current frame and the previous frame (as background)

Basically, I use BackgroundSubtractorMog2 provided in OpenCV to implement background subtraction and extract the foreground image.

Code Snippet:

```
cv::BackgroundSubtractorMOG2 bg(5,3,false);
bg.operator ()(frame,foreground);
bg.getBackgroundImage(background);
```



And then, I use cvFindContours to find all contours in the foreground image and draw rectangle for each contours.

Code Snippet:

```
cvFindContours(foreground,storage,&contours,sizeof(CvContour),CV_RETR_EXTERNAL
,CV_CHAIN_APPROX_NONE, cvPoint(0,0));
```

```

for( ; contours != 0; contours = contours->h_next )
{
    rect = cvBoundingRect(contours, 0);
    cvRectangle(frame, rect, CV_RGB(255,0,0), 1, 8, 0);
}

```



Optical Flow

Idea: Specify a set of points beforehand in the frame (usually in the first frame) and track those points in the next frames.

To specify a set of points, I use `cvGoodFeaturesToTrack()` function provided in OpenCV.

```

cvGoodFeaturesToTrack(gray,eig_image, tmp_image,featuresA, &corner_count, 0.01,
10, 0, 3, 0, 0.04);
(In this function, gray is the gray image of the first frame. featuresA will hold
a set of feature points as result.)

```

And then, I use `cvFindCornerSubPix()` to find more accurate corner points.

```

cvFindCornerSubPix(gray, featuresA, corner_count, cvSize(win_size, win_size),
cvSize(-1, -1), cvTermCriteria(CV_TERMCRIT_ITER|CV_TERMCRIT_EPS, 20, 0.03));

```

Finally, I use `cvCalcOpticalFlowPyrLK()` that implements Lukas-Kanade algorithm to track and compute the new position of those points specified in the last frame, and store the results in `featuresB`.

```

cvCalcOpticalFlowPyrLK(prev_gray, gray, prev_pyramid, pyramid, featuresA,
featuresB, corner_count, cvSize(win_size, win_size), 5, status, 0,

```

```
cvTermCriteria(CV_TERMCRIT_ITER|CV_TERMCRIT_EPS,20,0.03), 0);
```



Input :

