Moving Pedestrian and Vehicle Detection and Tracking

Prajyoth Bhandary
CS510 – Computer Vision
03/17/2017

Aim

To detect and track Pedestrian and Vehicle movement from a given sample surveillance video.

Tools Used

- OpenCV 2.4.13 , Python 2.7 , and Python libraries numpy and pygame.

Implementation

The algorithm can be divided into key steps, namely: -

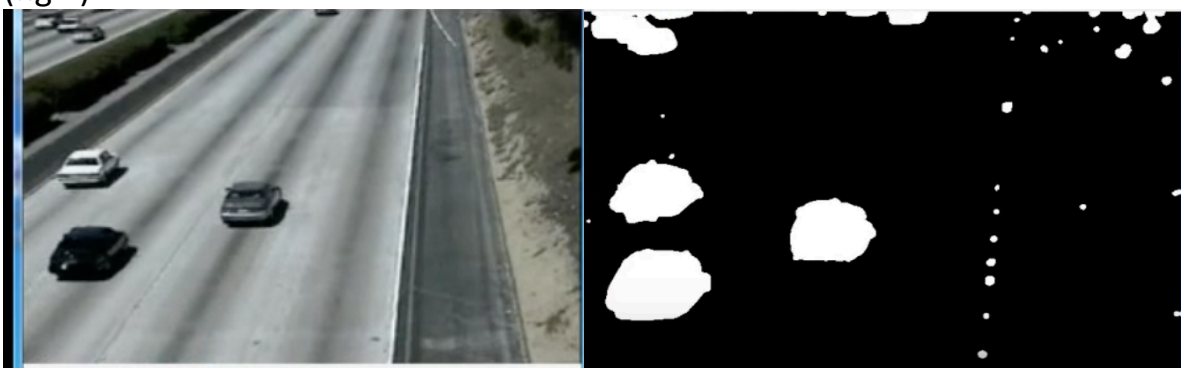
(i) Optical Flow analysis

a. Here we find the apparent motion of objects between two consecutive image frames using calcOpticalFlowPyrLK() implementation of openCV.

(ii) Background Subtraction

a. In this step we focus on removing static background objects and bring to focus the moving dynamic objects. This algorithm is seeded 20% of the initial frames to allow for more accurate background detection.

(fig 1)



Original Image Background Subtraction

(iii) Feature Tracking

a. Once we have identified which objects in the frame are of interest we track features of these objects. This step requires results from optical flow analysis and background subtraction. We also save up to 15 frames worth of feature points.

(iv) Motion Synthesis

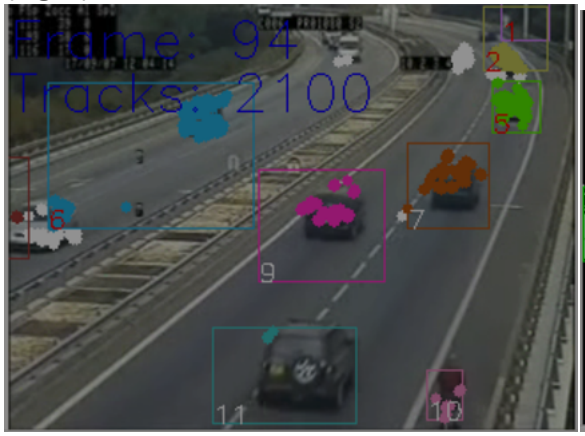
a. This is a prerequisite to the object tracking algorithm implemented in the next step.
b. It consists of an object with two properties – The object block area and an associated direction vector. It gets its input from the outputs of background subtraction and feature tracking.
c. It uses the saved history of frames to generate directional vectors. Although this step does not give very accurate motion tracking details.

(fig 2)

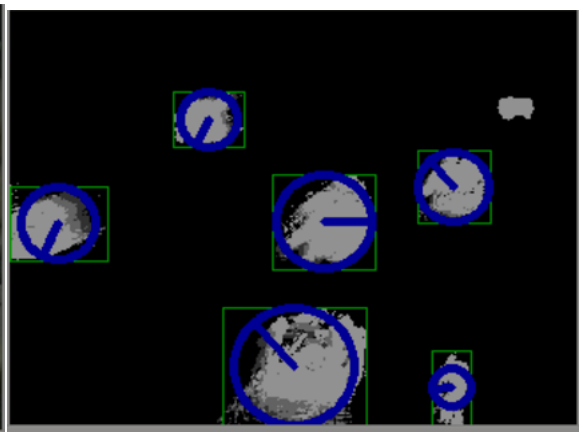
Original Image

Foreground + Optical Flow

(fig 3)



Feature Tracking



Motion Synthesis

(v) Object Tracking

a. Takes the output from motion synthesizer, that is nothing but a synthesizer object and produces a concise and clear representation of object. It achieves this by fine tuning certain parameters such as: -

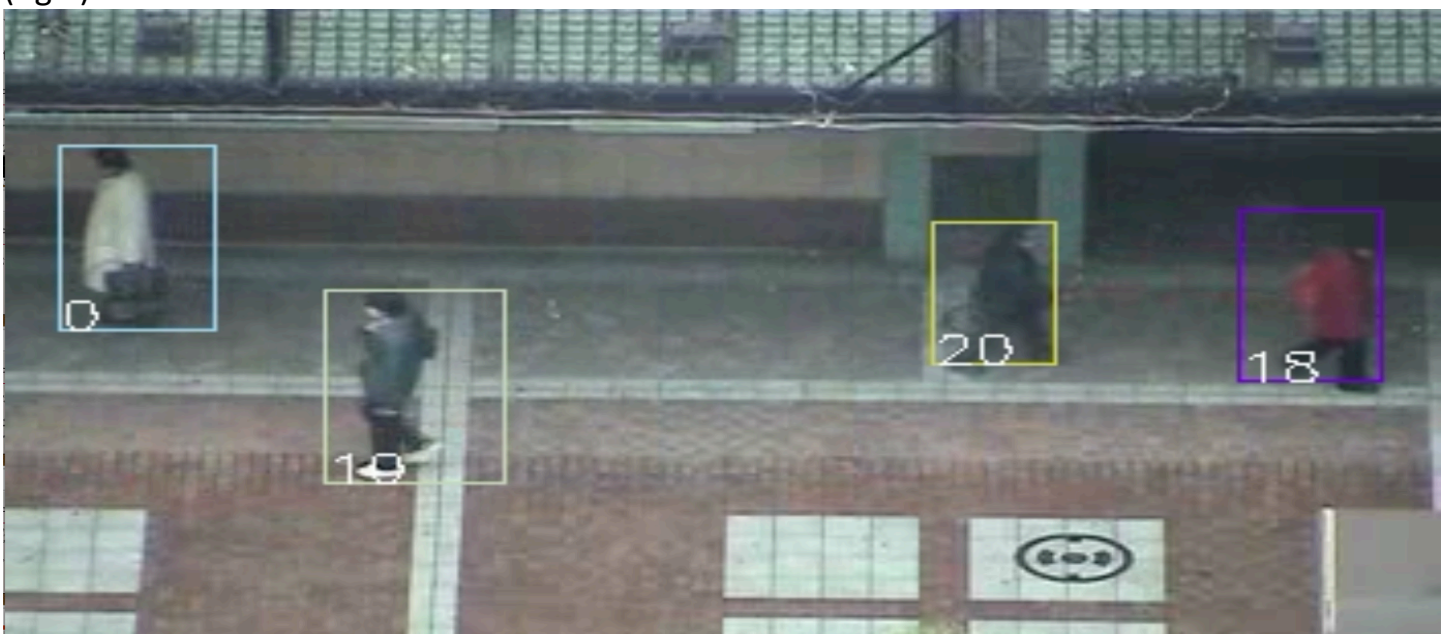
1. Filter objects which overlap. This is done to get a clean detection and tracing of individual distinct subjects without too much clutter.
2. Since each video is different based on image quality and lighting, I use a confidence interval. The confidence interval can be set by user. A high confidence level indicates that even the slightest movement will get captured. A low confidence interval is more conservative in tracking moving objects.
3. Based on lighting conditions, for e.g. in the night, a low confidence interval gives better results because of poor video quality. For better lit scenes we can use a higher interval.
4. If feature points get distorted or lost (e.g. If person goes behind building or a car takes a sharp turn), we use Kalman filtering. Kalman filtering predicts where the object could be in the next frame based on its previous positions.

Results

(fig 4)

In figure, the cars and pedestrian has been detected and is being tracked. You can see the motion synthesizer assigning unique identification to each subject of interest.

(fig 5)



This subway scene depicts the tracking of individual pedestrians.

(fig 6)

This is a more complex scene. You should be able to observe subject no. 2 moving behind the building which shows the kalman prediction in action. Also, vehicles and pedestrians which are parked and stationary the whole video do not show up as they become part of the background.

Conclusion

- Strengths
    - Does a good job tracking and identifying isolated subjects.
    - Is able to predict motion of objects that go out of frame or disappear briefly.
- Weaknesses
    - Kalman prediction sometimes fails to predict paths correctly. This is especially true when a vehicle turns sharply around a corner and disappears.
    - Unable to detect two objects separately much clearly when they overlap with each other.

Future Work

- Should be able to do a much better prediction of paths especially in a complex scene.
- Be able to differentiate between objects even when they overlap. This would require rigorous training of data sets.