The American University in Cairo

Computer Science and Engineering Department

Fundamentals of computer vision

Dr. Mohamed Mostafa

Final project

Detection of car crashes in videos

Hager Radi Mahmoud 900123209

First Approach using Motion detection:

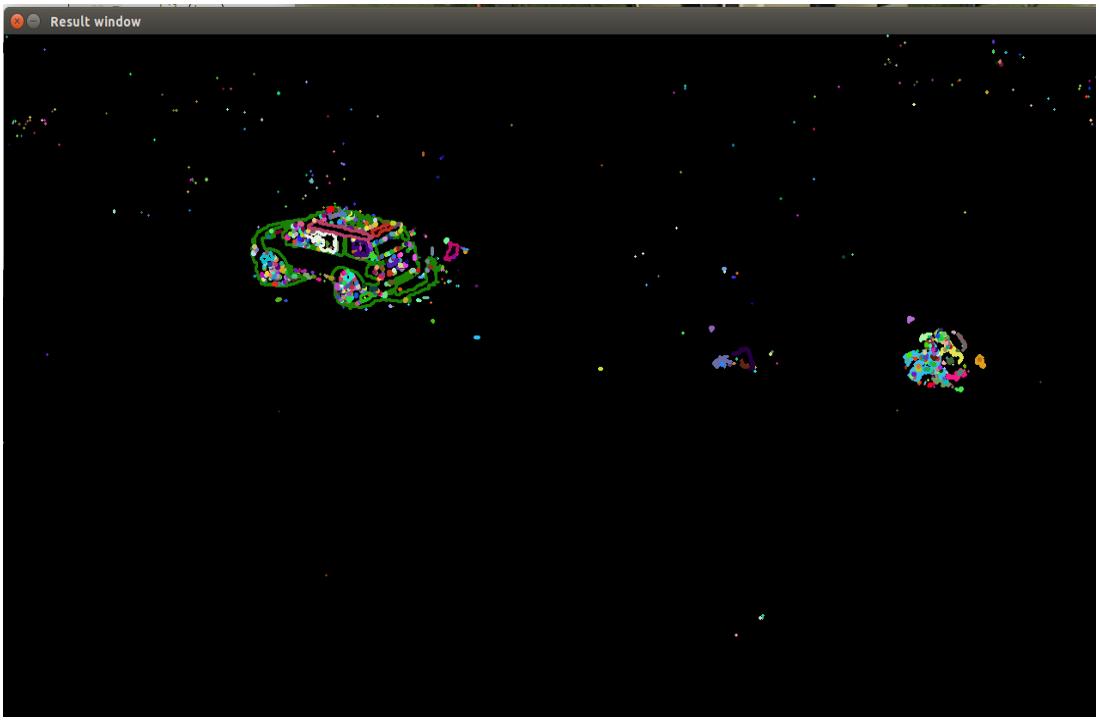
In an infinite loop, we start reading the frames; we convert them to grayscale and then do the following:

1. Detect the moving objects using **background subtraction** by getting the absolute difference between every two consecutive frames.
2. Convert the difference image to a binary image using a certain threshold.
3. Draw a rectangle that encloses the moving objects to detect the overall motion in each frame.
4. Outline the moving objects by calling the function (**FindContours**) on the difference image. And then draw these contours on the original frame to highlight the moving objects.
5. Draw the **bounding rectangle** for each moving object after excluding the contours with bounded rectangle less that a certain threshold to eliminate the small moving objects and focus only on cars/vehicle
6. Then, we **detect the crash** between two cars by checking in every frame if any two rectangles intersected at any point of time which will probably be an accident.

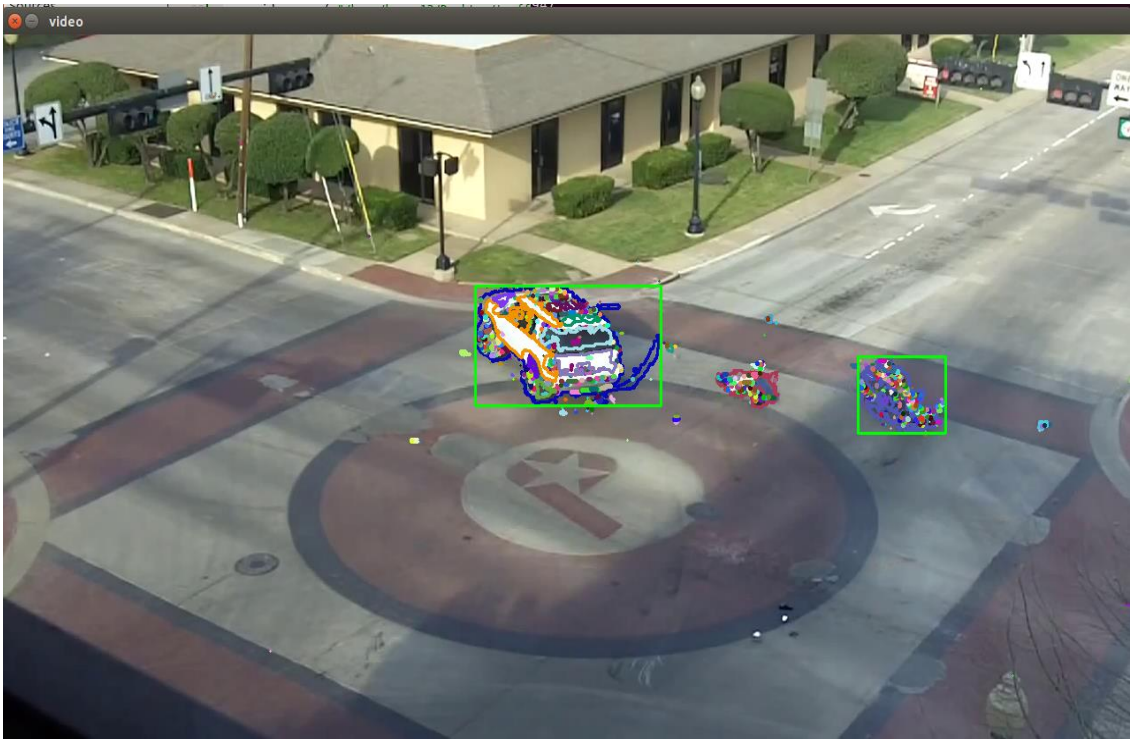
This approach was **not perfectly correct** because of:

1. We may have false positives. A certain moving object may have more than one bounding rectangle due to lighting problems that make (FindContours) function outline one object as more than one. So if we have two bounding rectangles around one object and they intersect at any point of time, because of the out-of-plane rotation of cars, it will be detected as an accident.
2. We have false negatives because of occlusion over the moving objects and dark background so the moving objects are not perfectly detected from the beginning.
3. The program does not actually track the moving object. So in a certain frame, the bounded rectangle over a moving object does not cover the same object in the next frame.

Result of background subtraction:



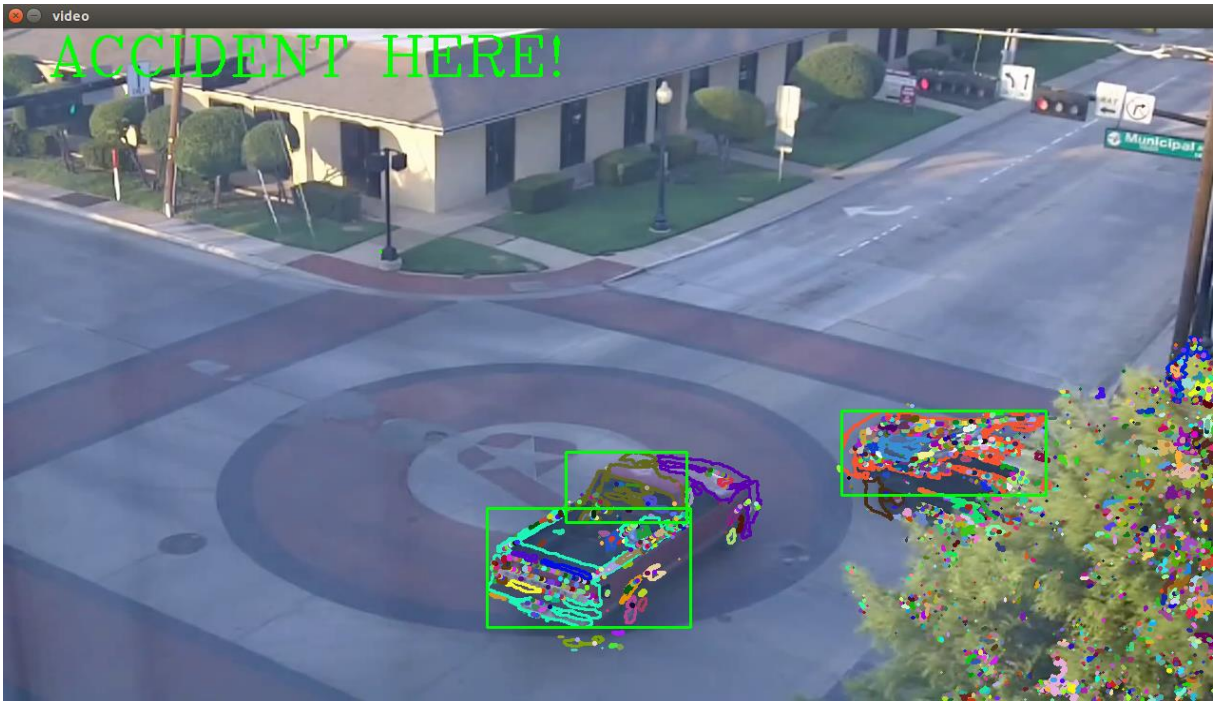
Result of Moving objects detection (findContours)



Bounding rectangles around moving objects (No accident)

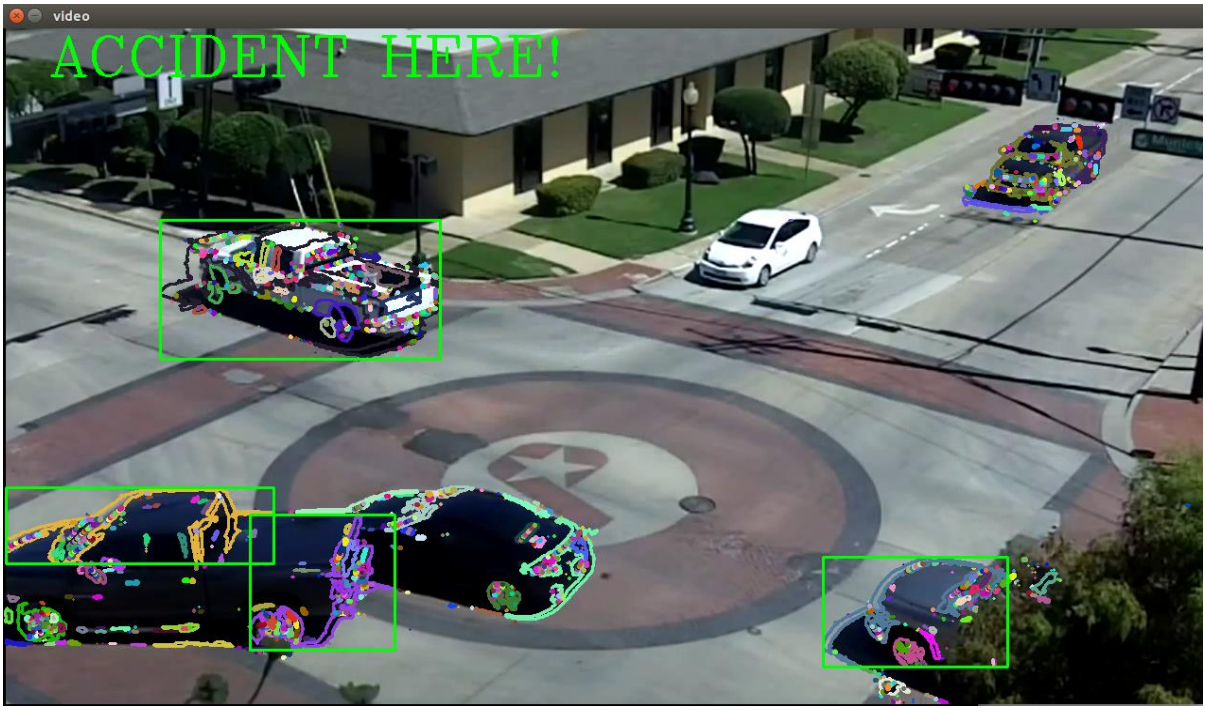
Accident detection (true positive)

Bounding rectangles around moving objects (No accident)

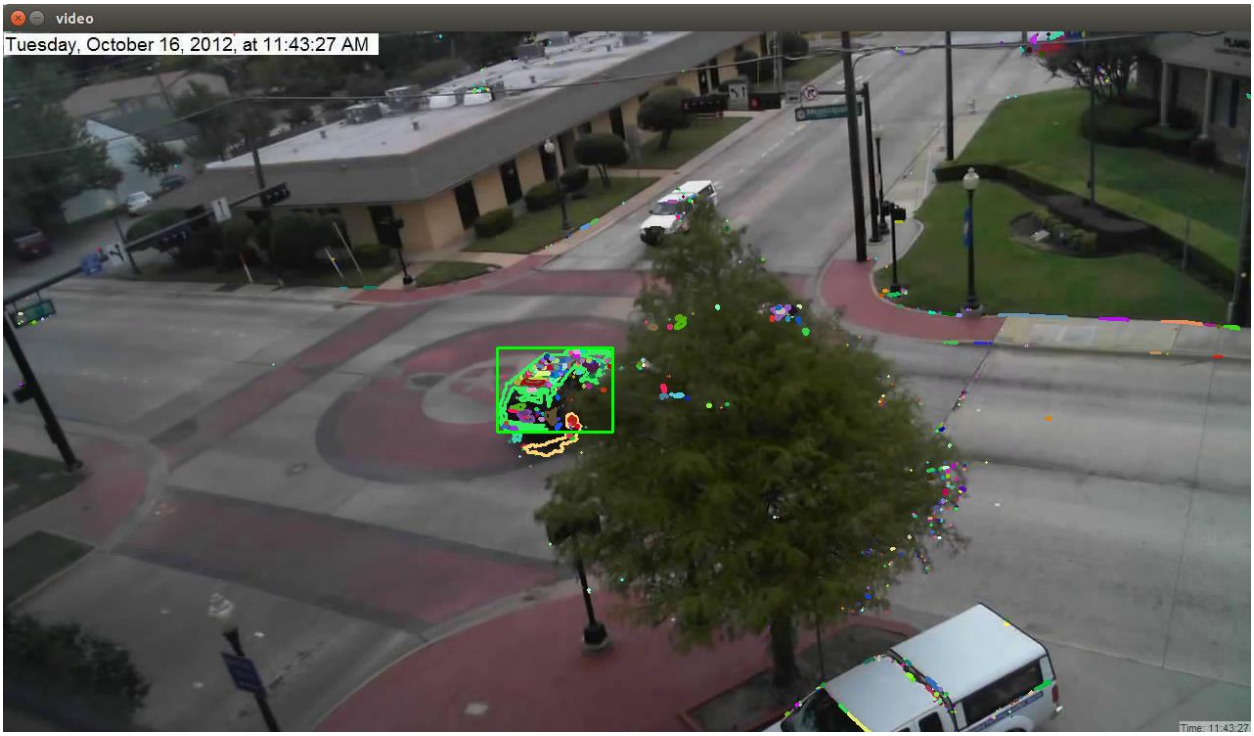


Definitely not accident prediction; It Is a false positive because of having two rectangles on the same moving object

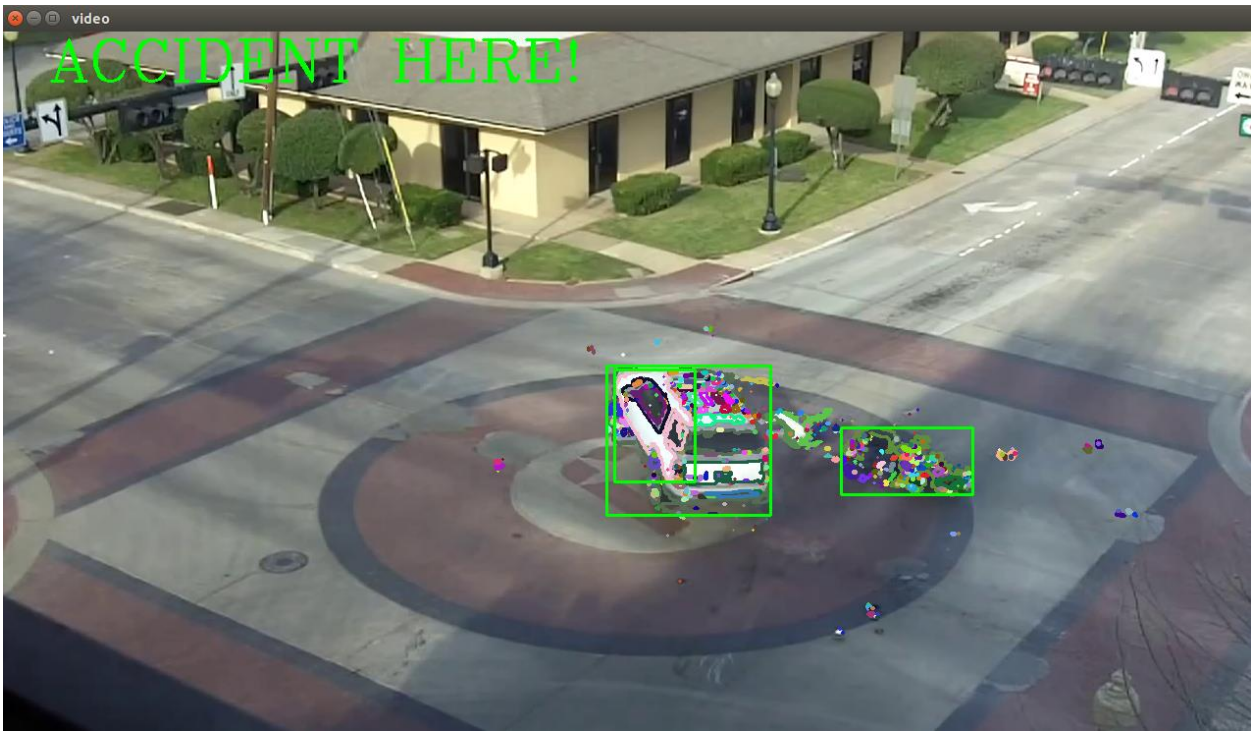
Accident detection (true positive)



It Is a false positive because of having two rectangles on the same moving object.



The other car is not detected as a moving object because of occlusion.



Another false positive.

A random frame with moving objects.

For the above drawbacks, I tried a second approach:

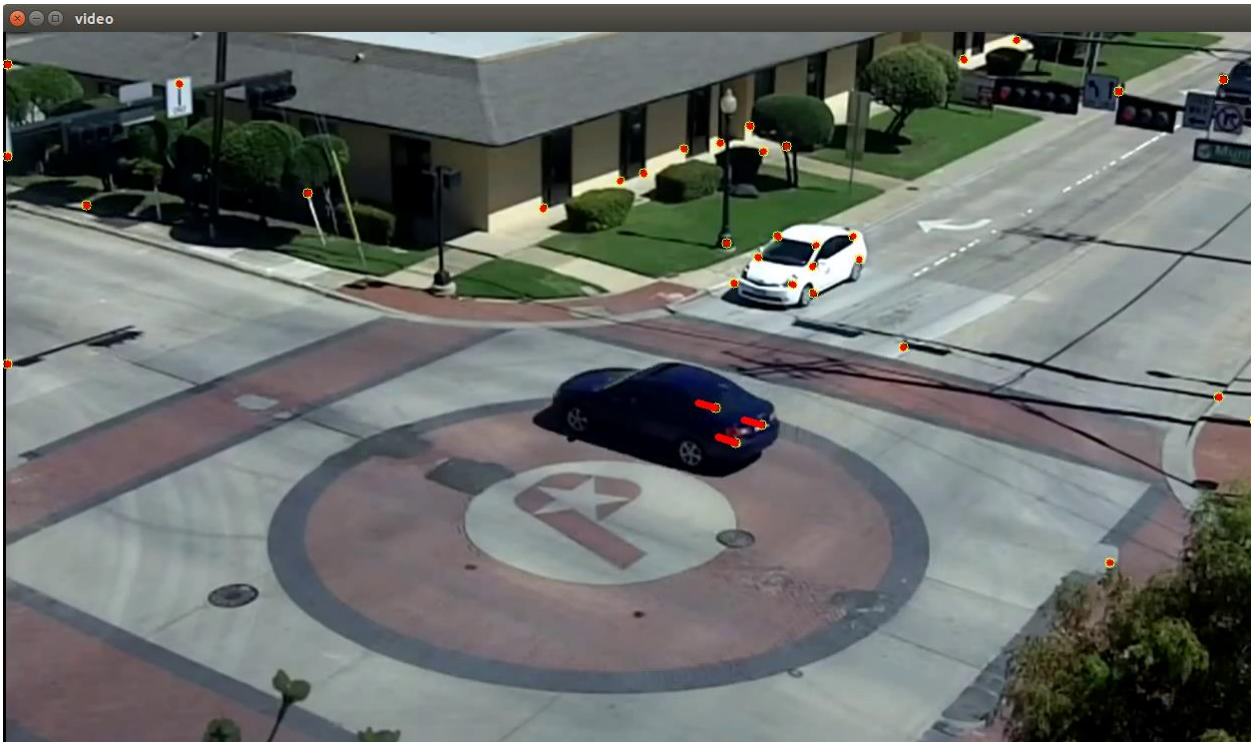
Motion Tracking:

after a lot of research and looking for tracking functions/classes in Opencv, i found a tracker demo from OpenCV that tracks the motion in videos.
Then I reached that we can do motion tracking using optical flow.

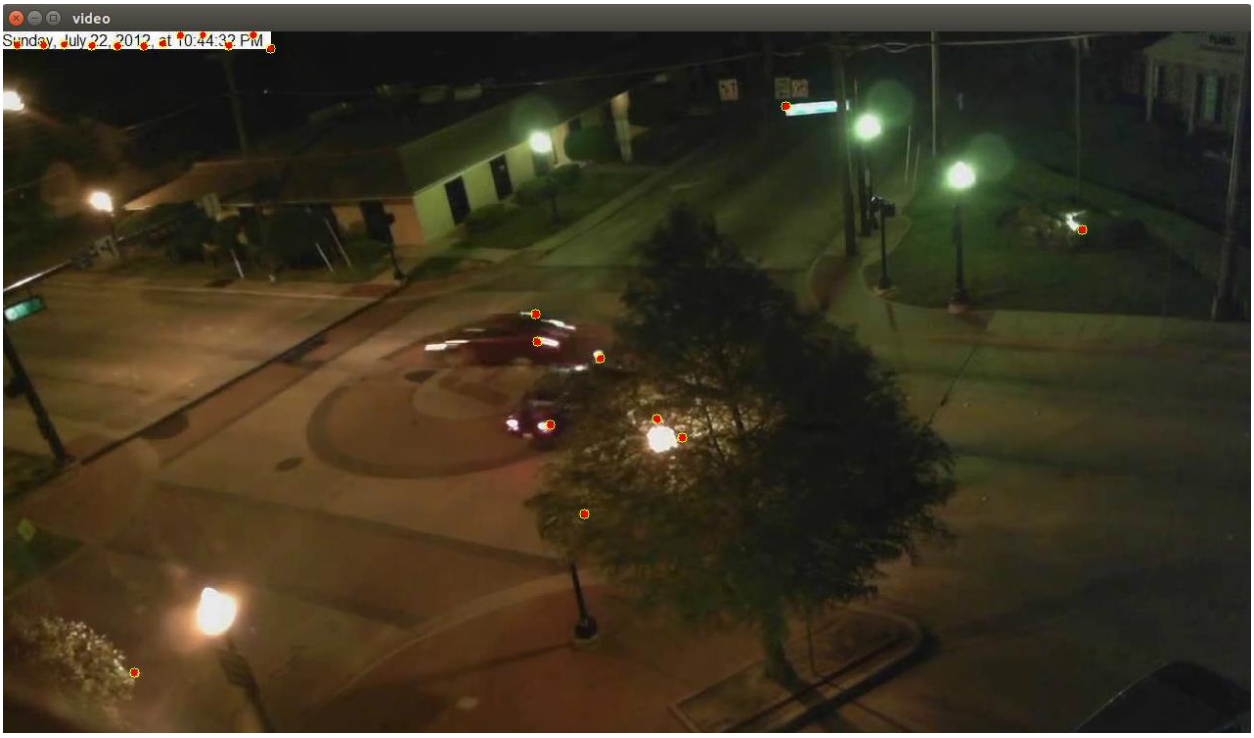
In an infinite loop, we start reading the frames; we convert them to grayscale; we initialize two vectors of points to save the previous and current features-to-track in any two consecutive frames.

1. We call the function "**GoodFeaturesTotrack**" that determines strong corners/features in an image. This function is based on Harris-corner detection algorithm. After specifying the maximum number of features we need, the quality level and the minimum distance between any two features, the function returns a vector of points with the best features to track.
2. We draw these detected features on the frame.
3. We call the function "**calcOpticalFlowPyrLK**" that Calculates an optical flow for a sparse feature set using the iterative Lucas-Kanade method with pyramids. We send the previous features we receive from step 2 and expect a vector of the position of the same features in the next frame.
4. Then we draw the new position of the same previous features and draw a line between them to show/track the actual motion of an object.

However, I was not able to detect the car crashes with this method.




video
video



Other unsuccessful approaches:

1. The mean shift and Camshift methods: these are used for motion tracking but they need the initial motion for the object to be tracked so it wouldn't be practical to initialize the moving objects in each frame because cars go in and out of the frames with no specific pattern.
2. Tracking API in opencv 3.0: spent some time trying to understand it but didn't actually have enough time to test it.

Test Video:
https://www.youtube.com/watch?v=INMiXqP8vI0

References:
https://github.com/Itseez/opencv/blob/master/samples/cpp/camshiftdemo.cpp
http://opencv-python-tutroals.readthedocs.org/en/latest/py_tutorials/py_video/py_meanshift/py_meanshift.html
https://github.com/abhi-kumar/OPENCV_MISC/blob/master/tracker.cpp
http://docs.opencv.org/master/d7/d8b/tutorial_py_lucas_kanade.html#gsc.tab=0
http://docs.opencv.org/3.0-beta/modules/tracking/doc/tracking.html