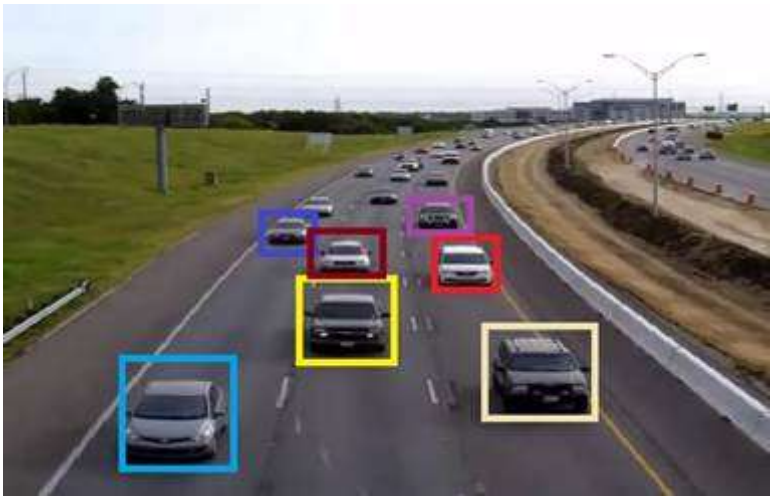
# CV Homework Set #3 – Multiple Object tracking

# Due: 31/5/16

Submission: 1) HW will be tested interactively with the checker. Times will be posted.

2) HW code should be submitted to HW checker either via DOK or by email to: imageprocessinghaifau@gmail.com

---



**Goal**: Write a multiple object tracking system using Kalman Filtering.

**General**: You will be using the Computer Vision Open Source library **OpenCV**.
You may choose the programming code of your choice (C++, Java, Python, Matlab) to interface with the OpenCV code.
Unfortunately, no technical support will be given in this course. A discussion forum has been opened in the Moodle site for this course.

Grading: Grade will be given based on actively running the program on images. Some example images you will be tested on will be available for you to test on. Additional new images will be used for testing as well.
Grade will be based on quality of segmentation, as well as convenient interfacing, organization and documentation of code, ext.

**Submission**:

Submit all necessary files to run the code.
Submit a **Readme file** explaining how to compile, make, and prepare the code (including
Makefile if necessary).
Submit **Instructions file** explaining how to run the program, (explain inputs, interface,
howto, etc).
**Do not forget to put Names and Student I.D. in all files.**

## HW3 Instructions:

Implement a multiple object tracking system using Kalman Filtering.

Write code that implements the following:

1) Given an input video, tracks the moving objects. Specifically given a highway scene
tracks the cars moving towards the camera. Cars are marked with different colored
bounding boxes (as in image above).

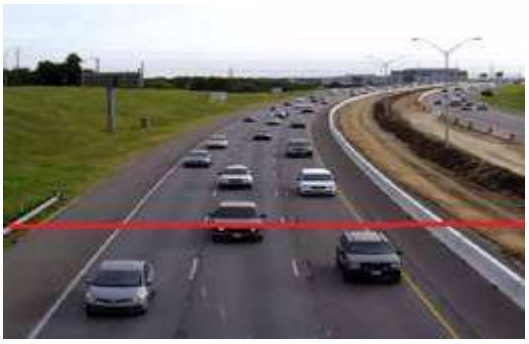
   Use the Video files found here.
   Use OpenCV functions (Kalman Filtering, and also BGsubtraction if you want).

   Suggested Steps:
   1) First Create a BG image. You can try simple averaging all the frames or you can use
   manual inputs to assist to create a god bg images. You can try to use BG Subtraction
   function of Open CV (but this might be overkill).
   You may share the BG image amongst the students.

   2) Create simple one object tracker.
   Start by tracking a single object. You can start by manually clicking in the image window on the
   car to track. So as not to confuse the tracker you can select a car that is unique in color
   so as to distinguish it from all other FG objects.
   You can try with a simpler video sequence (with one moving object).

3) Expand to multiple objects.
This might require matching between tracked objects and objects in previous frame.
Usually, closest object (center of mass of objects) works well enough since motion is
relatively slow between frames. Start by tracking in sequences where cars are not too
close to each other.
You must decide when a new object appears that did not exist in previous frames and
start tracking it.

4) Bonus Points:
Count the number of cars passing line. Given a y coordinate, mark a line in the video
frame and count the number of cars passing the line. Show the count interactively
(either on video or on screen). It is OK if multiple cars tracked as 1 car are counted as 1.



Simplifications:
* If the car is too small (far away or a motorcycle) you may disregard it. (Thus FG objects/Blobs
that are too small can be disregarded.
* If 2 (or more) cars are close they might be tracked as a single object (FG blob sees them as
one object) – that is fine. However if the blobs separate – tracking should split into 2 cars.
* The bounding box does not need to be exactly bounding the car.
* Since you must decide when a new object appears that did not exist in previous frames, you may assume

the cars all move towards the bottom of the image. You may also start tracking at specific y coordinate.
* if the video frames are too close (takes too long to track etc) you may skip frames.

2) Testing – Your system will be tested on the videos supplied to you + additional videos of the sequence.