# Exercise #5 – Extended Lab Proposal and Exploratory Coding

**DUE: AS INDICATED on Canvas**

Please read OpenCV tutorials, from http://docs.opencv.org/trunk/. Also read "Learning OpenCV" and the ER Davies text with emphasis on an advanced topic areas including:

1. 3D passive vision (disparity maps, depth maps, point cloud generation)
2. 3D active RGB depth mapping (here you will want to look at OpenNI and use one of our RGB-depth cameras – Asus Xtion) - http://www.openni.org/, depth maps and tracking in 3D
3. SIFT and SURF applications with the idea to provide digital image stabilization, mosaic processing (to stitch images together) or to find features for registration between left/right camera for stereopsis and passive depth mapping
4. Object and facial detection and recognition using for example Haar Cascades and Ada Boost for facial detection and recognition
5. Continuous skeletal transformation used for gesture recognition, gate analysis or other behavioral analysis or animation capture.

This exercise also includes exploration of driving assistance application development, which helps students explore intelligent automation that integrates machine vision. The exercise can serve to motivate ideas for a final extended lab project in the course.

If you have a specific project objective you prefer to explore instead of the self-driving exercises, you can substitute your own exploratory programming as noted in requirements below.

## Exercise #5 Requirements:

1) [10 points] Read Sebastian Thrun's design paper on the Stanley vehicle which won the DARPA challenge – "Stanley: The Robot that Won the DARPA Grand Challenge", Journal of Field Robotics 23(9), 661–692 (2006). Describe in good detail how machine vision was used and why it was necessary. Could the LIDAR have been fully replaced by a more sophisticated machine vision system with multiple cameras?

2) [40 points] Research, write code and demonstrate processing of road test video to support a driving assistant. Or choose the most challenging and compute intensive algorithms for your specific interest.

   **For a creative project** of your own interest create an exploratory program to assess feasibility of your application as follows:

a) Select one or more key algorithms and verify that there is an OpenCV implementation you can re-use or that you can implement from the ground up and test on a single or set of still images (e.g. stitch together 2 or more images using feature point correspondence).
b) Modify your example above to work on a continuous stream of images from pre-recorded video that you capture which represents a challenge set for your project of interest (e.g. stitches together a panorama taken from 2 or more cameras as continuous video).
c) Provide processed video and input video to show that your exploratory prototype works or explain why it does not

**For self-driving car**, select one of the following applications (**choose just one**):

a) Lane departure warning system (https://github.com/tomazas/opencv-lane-vehicle-track ). Using the algorithms you have learned in the previous labs try to develop a lane departure warning system. Use the video that has been uploaded in the course website [Self-driving-test-data-1-raw, Self-driving-test-data-2-raw].
   i) Hints – Consider use of Hough lines to find lines on the road and Haar feature extractor for vehicle detection.
   ii) Make sure you draw the detected lines on the image for feedback (this can either be enabled or disabled using command line arguments or interactive key stroke inputs)
   iii) Your code should take arguments as
      (1) Video file name
      (2) Show intermediate results (--show), name the image window according to the intermediate results
      (3) Store the results back in a video (--store followed by the filename)

b) Pedestrian detection (http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/ ). For this question you will have to use Caltech dataset, which has been given in the reference link.
   i) Hints – Histogram of Gradients (HOG descriptor) for pedestrian detection
   ii) Make sure you draw a bounding box on the image for feedback (This can either be enabled or disabled using command line arguments)
   iii) Your code should take arguments as
      (1) Video file name
      (2) Show intermediate results (--show), name the image window according to the intermediate results
      (3) Store the results back in a video (--store followed by the filename)
      (4) Log the number of pedestrians detected and the co-ordinates of the detected pedestrian in a log file.

c) Basic self-driving car first-cut auto-pilot (https://www.theverge.com/2016/6/6/11866868/comma-ai-george-hotz-interview-self-driving-cars ). Consider that you are designing a self-driving car, You are given with two cameras (front view and rear view).
   i) You have the control of the car using commands to control
      (1) Accelerator(0-100%)
      (2) Break(0-100%)

        (3) Steering(rotate clockwise or anticlockwise -n degree)
    ii) The user will give the destination, you have the access to Google maps API (You can download the directions offline before starting the car or Assume you have the direction information available to you. [We can add a log file where it says at what frame you have to take a right/left or straight – to make students life easier])
    iii) Your code should be able to slow down according to speed limits(Sign detection – Don't know whether we need this)
        (1) Your code should detect the pedestrians/cars in front of you and stop before 2 meters of any obstacle and according to the direction information it should be able to take left/right or go straight on a junction. (Lane departure – hard without sideview/blindspot)
        (2) Have a feedback system which maintains a pointer at the steering wheel video, and compare the angle difference between the marker in the video and your algorithms prediction (Log the values if possible).

3) [10 points] Read texts, lecture notes, and supporting materials, consider example code, and choose an advanced Computer Vision application that is either:
    a) Improve work from self-driving car work to create a design and prototype for a self-driving car autopilot (specific functions like lane following, braking, cruise, pedestrian and hazard detection)
    b) 3D passive vision application (using two webcams)
    c) 3D active vision application (using OpenNI and Asus Xtion camera)
    d) SIFT/SURF application
    e) Object detection and recognition (facial or other)
    f) Real-time continuous skeletal transformation
    g) **Other project you propose**

4) [20 points] Write a 3 paragraph proposal for your application that describes it in detail with block diagrams if needed or other figures so that it's clear what you will implement and demo for Lab #6. Define your project goals with at least three features at each of the following levels of challenge:
    a) Optimal – this should include processing, I/O and/or storage optimization to increase throughput and any real-time or interactive requirements you have as well as performance related to accuracy
    b) Target – goals should include the three main features of your design (e.g. frame rates, detection, tracking, recognition, depth estimation, transformation from image to point cloud, etc.)
    c) Minimum – simple objectives that define basic success (e.g. continuous skeletonization of a hand and recognition of a specific set of gestures).

**[20 points]** Overall, provide a well-documented professional report of your findings, output, and tests so that it is easy for a colleague (or instructor) to understand what you've done. Include any

C/C++ source code you write (or modify) and Makefiles needed to build your code and make sure your code is well commented, documented and follows coding style guidelines. I will look at your report first, so it must be well written and clearly address each problem providing clear and concise responses to receive credit.

In this class, you'll be expected to consult the Linux and OpenCV manual pages and to do some reading and research on your own, so practice this in this first lab and try to answer as many of your own questions as possible, but do come to office hours and ask for help if you get stuck.

Upload all code and your report completed using MS Word or as a PDF to Canvas and include all source code (ideally example output should be integrated into the report directly, but if not, clearly label in the report and by filename if test and example output is not pasted directly into the report). ***Your code must include a Makefile so I can build your solution on an embedded Linux system (R-Pi 3b+ or Jetson). Please zip or tar.gz your solution with your first and last name embedded in the directory name and/or provide a GitHub public or private repository link. Note that I may ask you or SA graders may ask you to walk-through and explain your code. Any code that you present as your own that is "re-used" and not cited with the original source is plagiarism. So, be sure to cite code you did not author and be sure you can explain it in good detail if you do re-use, you must provide a proper citation and prove that you understand the code you are using.***

**Grading Rubric**

[10 points] Stanley paper analysis:

[5 pts] How machine vision was used________________________________________

[5 pts] How was LIDAR used? _____________________________________________

[40 points] Driving assistant application or exploratory programming for creative project:

[10 pts] Design___________________________________________________________

[10 pts] Code ____________________________________________________________

[10 pts] Build and test ____________________________________________________

[10 pts] Processing and modification of example video to demonstrate continuous transformation_________________________________________________________

[10 points] Selection of project and proposal:

[5 pts] Relevance of selection and challenge __________________________________

[5 pts] Rationale for selection or proposal in terms of application and learning

_______________________________________________________________________

[20 points] Proposal description and outline of goals:

[10 pts] Minimum_________________________________________________________

[5 pts] Target _____________________________________________________________

[5 pts] Optimal ____________________________________________________________

[20 points] Quality of reporting and code quality and originality:

[10 pts] Professional quality of reporting, testing and analysis (0…6 is below average, 7 is average, 8 is good, 9 excellent, and 10 is best overall.)_________________________________________________________________

[10 pts] Code quality including style, commenting, originality, proper citation for re-used code, modified code, etc.______________________________________________________