Professor Fearing EECS192/Assignment #1 v1.00 Spring 2016

Due at beginning of class Tues Mar. 15.

(One assignment per team.)

The purpose of this assignment is to allow you to prototype track finding algorithms with a high level language before implementing in C on the KL25Z processor. You are free to use signal processing libraries, floating point, etc. One goal is to find the best performance which can be obtained with unlimited processing capability. However, remember that your line finding algorithm will need to work in real-time on a 40 MIPS processor.

Typical strategies to use for finding the track include:

1. Frame subtraction and peak detection
2. smoothing followed by gradient detection (e.g. difference of Gaussians approximation to the Laplacian)
3. curve fitting, e.g. cubic spline or $\tanh^{-1}$.

3 sets of line scan data (for a white stripe on carpet at NATCAR 2015) is provided on Piazza for EE192 under "Resources". A partial iPython notebook `linescan-HW1.ipynb` is provided.

Complete the function `find_track(linescans)` which takes as input n frames of linescans of 128 values in the range 0...255 and returns:

a) `track_center_list` which is a length n list of the index in the range 0...127 corresponding to the center of the track in each frame.

b) `track_found_list` which is a length n list of booleans, **True** if the track is visible for a particular frame.

c) `Cross_found_list` which is a length n list of booleans, **True** if a crossing is present for a particular frame.

Upload your completed iPython notebook to bcourses.

Your python function will be tested against another data set taken under similar conditions on a similar track.

Directions for installing iPython can be found on the EE123 web page at:
`http://www-inst.eecs.berkeley.edu/~ee123/sp15/python.html`.

track?	crossing?	track found	cross found
F	F	F	F
F	T	F	T
T	F	T	F
T	T	T	T