ECE 634: Digital Video Systems
Spring 2017
Instructor: Prof. A. R. Reibman

PURDUE UNIVERSITY

Programming Assignment 0

Video playback and debugging environment
(Last update: January 9, 2017)

The purpose of this exercise for you to create yourself an environment to write software for processing videos. One essential programming requirement is that you have a method to debug your code. Matlab offers powerful visual debugging capabilities, but is quite weak otherwise when processing videos. OpenCV has many capabilities for processing videos, but you have to create your own tools for visual debugging.
Use Python, or C/C++.

- Download and install OpenCV with ffmpeg capabilities on your computer. I recommend any version between 2.4.9 and 2.4.13, although if you're familiar with version 3.0+, you are also welcome to use that.
  - Mac: I recommend homebrew.
    `brew tap homebrew/science`
    `brew install opencv --with-ffmpeg`
  - Windows, this web-site may be useful to install OpenCV with pre-built ffmpeg binaries:
    `http://kronoskoders.logdown.com/posts/256664-installing-opencv-and-ffmpeg-on-windows`
  - Linux, this web-site may be useful:
    `http://www.samontab.com/web/2014/06/installing-opencv-2-4-9-in-ubuntu-14-04-lts/`
- Write a program that reads video in one of 3 formats, and displays it at one of 3 speeds: as fast as possible, 30 fps, and one frame at a time. For your program to be a flexible start to future assignments, it should not have hard-wired input but instead take inputs from the command line. Required formats:
  - YUV format, one byte per pixel.
  - A compressed video file like *.mp4
  - A series of JPEG images with names "file%4d.jpg", where the "%4d" means a 4-digit integer (or bigger) to denote frame number, like tennis0010.jpg is the 11-th image (because time starts at 0).
- Augment your program to compute and display the frame-difference image, which is the difference between two adjacent frames.
- Augment your program to respond to keyboard and mouse commands. You should include the following set of operations:
  - Stop/Start (when the space bar is pressed).
  - Step One Frame (when the "." period key is pressed).
  - Print (x,y) (when the left mouse is pressed).

– Print (R,G,B) when the right mouse is pressed).

Note that there are ample sites on the web that can provide tutorials and even sample code to point you on your way. For mouse inputs, see for example:
`http://opencv-srf.blogspot.com/2011/11/mouse-events.html`

Give credit where credit is due.

- Sample video files can be found on the course web page.

Extra credit:

- Come up with one or two of your own debugging ideas. Implement. Explain in clear English why you think this would be a useful complement for you going through this semester.

Questions or comments concerning this assignment should be directed to Prof. Amy Reibman, reibman@purdue.edu.