

ECEn 631 Feature Tracking

Objectives:

- Learn to use OpenCV
- Learn to capture live video using OpenCV
- Learn to process live video
- Learn to detect features
- Learning to track a known object

Instructions:

- Use OpenCV and a webcam to complete this assignment.
- You will lose 10 points if any of the following requirements is not met:
 - Submit 2 source code files and 2 videos in one zip file without the folder or directory.
 - Use your first name and last name (e.g., justinsmith.zip) as the file name.
 - Combine the result of all five functions of Task 2 into one video.
- Login to myBYU and submit your work through BYU Learning Suite online submission.

Task 1: Acquire and Display Live Video

- Set up your system to capture and display live video in a window. The following sample code using OpenCV functions can be used to complete this task. Use your last name to name the display window. Nothing to be submitted for this task.

```
VideoCapture video(0);           // get a camera object
Mat frame;                       // allocate an image buffer object
namedWindow("Lee", CV_WINDOW_AUTOSIZE); // initialize a display window
video >> frame;                  // grab a frame from the video feed
imshow("Lee", frame);            // display the grabbed frame
```

Task 2: Real-time Feature Detection 70 points

- Use your code in Task 1 and add five feature detection functions to demonstrate real-time image processing. Run your live image processing code in an infinite loop and use waitKey() to get input from the keyboard to switch among different feature detection functions on the fly.
- Functions to be implemented include thresholding, Canny, corner and line detection, and differencing.
- Submit your code and **one short video**. You can use the following sample code to create and save the video. This code works in Windows 7 (VS2010). You may have to make changes if you use a different operating system or development environment.

```
VideoWriter VOut;                // Create a video write object.
// Initialize video write object (only done once). Change frame size to match your camera resolution.
VOut.open("VideoOut.avi", CV_FOURCC('M', 'P', 'E', 'G'), 30, Size(640, 480), 1);
or VOut.open("VideoOut.avi", -1, 30, Size(640, 480), 1); // use this if you don't have the correct codec
VOut << frame;                  // Use this line in a loop to add one frame at a time to the video write object.
// Video will be saved automatically. Don't save too many frames (60 is good for 2 seconds of video)
```

- (10 points) Point the camera to yourself. Display the binarized video as the result of thresholding (choose your own threshold).
- (10 points) Point the camera to yourself. Display Canny edges (choose your own thresholds).
- (20 points) Download and print a copy of the chessboard image. Move the chessboard paper in front of the camera. Detect corners using the OpenCV sub-pixel corner detection function. Display corners.
- (10 points) Move the chessboard paper in front of the camera. Detect lines using the OpenCV line detection function. Display lines.
- (20 points) Use two image buffers, one for storing the previous frame and one for capturing the current frame. Use an OpenCV absolute difference function to calculate the difference image. Display the difference image.

Task 3: Real-time Tennis Ball Detection 30 points

- Download the tennis ball image sequence from the class website.
- Read one image at a time and process it to locate the tennis ball. Draw a rectangle or superimposed color to ball area (of the original image) to indicate the location of the tennis ball.

- Methods to be considered include, but are not limited to, simple thresholding, differencing, and edge detection.
- Generate a short video from the resulting image.
- Submit your code and the video.

