EECE 5354 Computer Vision

Assignment 1

Shrey Shah

27th January 2019

This report documents the work performed and modifications made to the program testcv_mt_vid.py for assignment 01 of Computer Vision. The modifications enabled the program to perform the following OpenCV operations:

1) Gaussian Filter: The function sepFilter2D() with a Gaussian kernel performed the gaussian filtering operation on the video frame. The function getGaussianKernel() with aperture size = 11 was to generate the Gaussian Kernel. The sigma for the gaussian kernel is selected by the user with a trackbar ranging from 0 to 64. This operation is performed by pressing 'g' key.

2) Colormap: The function applyColorMap() was used to apply various colormaps to the video frame. The colormap to be applied is selected by the user with a trackbar. For instance, a 0 on the trackbar applied the Autumn colormap, a 1 on the trackbar applied the colormap Bone and so on. This operation is performed by pressing 'c' key.

3) Morphological Erosion: The function erode() was used to perform morphological erosion on the frame. The getStructuringElement() function was used to generate a structuring element of size (7, 7) for performing the erosion. The type of the structuring element is selected by the user with a trackbar. Trackbar value of 0 would create a rectangular structuring element, value of 1 would create a cross-shaped structuring element, and a value of 2 would create an elliptic structuring element. This operation is performed by pressing 'm' key.

4) Downsampling: The function pyrDown() was used to blur and downsample the frame. The size of the output image is computed as Size((src.cols+1)/2, (src.rows+1)/2)[1]. This operation is performed by pressing 's' key.

5) Region of Interest: Clicking the left mouse button and dragging the cursor without releasing button lets the users create a rectangular region of interest in the video frame, such that all the processing operations are restricted to the region of interest. This was achieved by using the functions setMOUSECALLBACK(), EVENT_LBUTTONDOWN() and EVENT_LBUTTONUP(), while the rectangle was created by using the function rectangle()[2]. The following processing operations can be performed in the ROI:
   i) Display canny edges with or without the raw frames. This operation is performed by pressing 'e' key, while the raw frames are toggled by pressing 'f' key.
   ii) Apply various colormaps to the ROI using a trackbar. This operation is performed by pressing 'c' key.

[1] Formula from: https://docs.opencv.org/3.4.4/d4/d86/group__imgproc__filter.html#gaf9bba239dfca11654cb7f50f889fc2ff

[2] Tutorial for Capturing Mouse click events from: https://www.pyimagesearch.com/2015/03/09/capturing-mouse-click-events-with-python-and-opencv/

iii) Apply Gaussian Filtering to the ROI. This is done by the sepFilter2d() function with a Gaussian kernel of aperture size 11 and sigma value selected by the trackbar. This operation is performed by pressing ‘g’ key.