**from** **PIL** **import** Image, ImageDraw

**import** **matplotlib.pyplot** **as** **plt**

**import** **numpy** **as** **np**

**import** **math**

**from** **scipy** **import** signal

**import** **numpy.linalg** **as** **lin**

**from** **datetime** **import** datetime

**from** **Hmk2** **import** boxfilter

**from** **Hmk2** **import** gaussconvolve2d

**def** boxconvolve2d(image, n):

*"""*

*Convolve the image with a boxfilter of size n.*

*Input:*

*image the image to be convolve2d*

*n the size of the box filter*

*Output:*

*return convolved image*

*"""*

**return** signal.convolve2d(image, boxfilter(n))

**def** Estimate\_Derivatives(im1, im2, sigma=1.5, n=3):

#No changes

**def** Optical\_Flow(im1, im2, x, y, window\_size, sigma=1.5, n=3):

*"""*

*Calculates the new coordinates of the point due to position changes in*

*the scene.*

*[I]:*

*im1 original image*

*im2 is im1 under position changes*

*x,y x,y original coordinates*

*window\_size size of the window where the optical flow equation is computed*

*sigma sigma parameter for the gaussian filter*

*n size of the boxfilter*

*[O]:*

*V[1],V[0] new x,y coordinates*

*"""*

**assert**((window\_size % 2) == 1), "Window size must be odd"

*# UNCOMMENT THE NEXT LINE WHEN YOU HAVE COMPLETED Estimate\_Derivatives*

Ix, Iy, It = Estimate\_Derivatives(im1, im2, sigma, n)

half = np.floor(window\_size/2)

*# # select the three local windows of interest*

*# # UNCOMMENT THE NEXT LINE WHEN YOU HAVE COMPLETED Estimate\_Derivatives*

win\_Ix = Ix[y-half-1:y+half, x-half-1:x+half].T.flatten()

*# #*

*# # PROVIDE THE REST OF THE IMPLEMENTATION HERE (BASED ON THE WIKIPEDIA ARTICLE)*

*# #*

win\_Iy = Iy[y-half-1:y+half, x-half-1:x+half].T.flatten()

win\_It = It[y-half-1:y+half, x-half-1:x+half].T.flatten()

A = np.vstack([win\_Ix, win\_Iy])

At = A.T

V = np.dot(At, lin.pinv(np.dot(A, At)))

V = np.dot((-1)\*win\_It, V)

**print** V[1], V[0]

**return** V[1], V[0]

**def** AppendImages(im1, im2):

#No changes

**def** DisplayFlow(im1, im2, x, y, uarr, varr):

#No changes

**def** HitContinue(Prompt='Hit any key to continue'):

#No changes

Q5)

Window Size = 5

Sigma = 0.9

frame 8 to 9

-2.29509819451 -0.0036688268899

frame 9 to 10

-2.19472946224 -0.131779512392

frame 10 to 11

-2.13637901953 -0.134549469841

frame 11 to 12

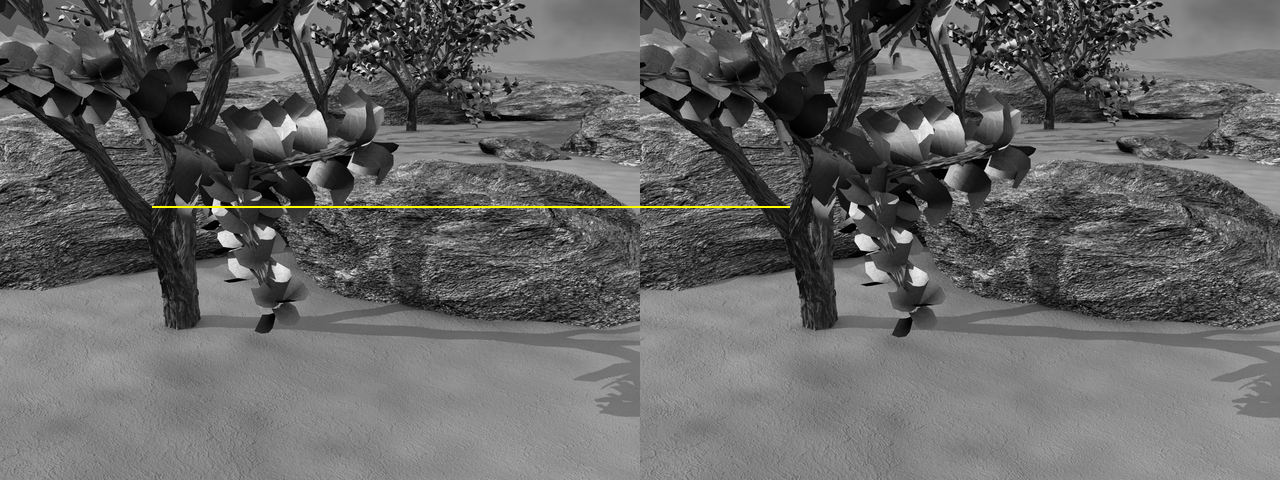
-1.5602610228 -0.377457722549

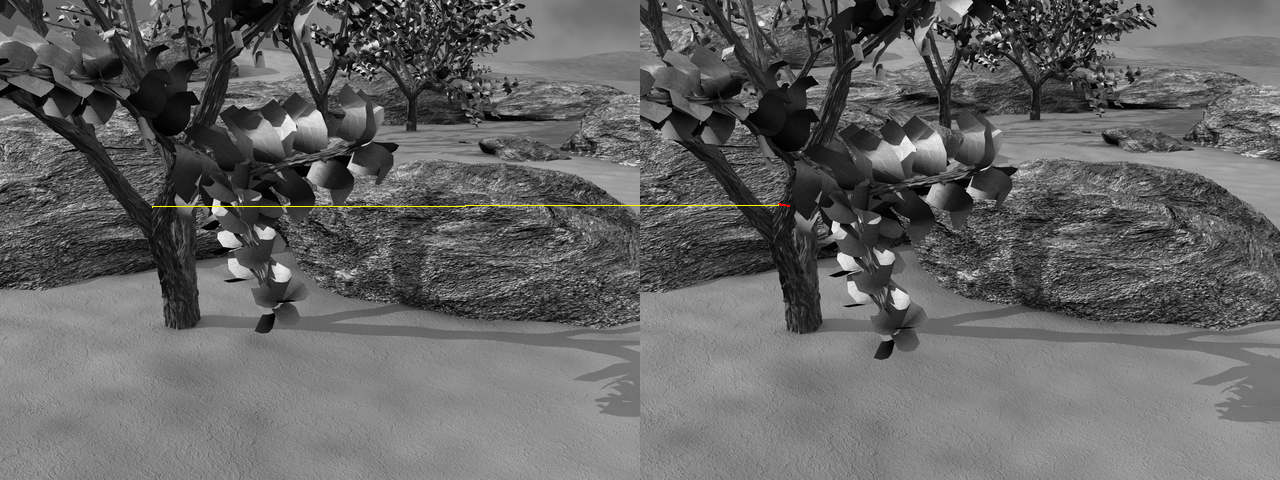
frame 12 to 13

-1.23863905496 -0.475137414346

frame 13 to 14

-1.41270286792 -0.646444473595





Q6) i) Situations of a flat region where all the derivatives are zero (locally constant), then there are problems. ii) Linearly correlated derivatives of the x and y axis may cause problems as well, representing lines, however, am not too sure.