

```

# CS4243 Lab 3
# =====
# Name: Tay Yang Shun
# Matric: A0073063M

import cv2
import cv2.cv as cv

import numpy as numpy

im = cv2.imread('LabPhoto1.jpg', cv2.CV_LOAD_IMAGE_COLOR)
gr = cv2.imread('LabPhoto1.jpg', cv2.CV_LOAD_IMAGE_GRAYSCALE)

# Pops up a window for displaying the CV_LOAD_IMAGE_GRAYSCALE
winname = 'imageWin'
win = cv.NamedWindow(winname, cv.CV_WINDOW_AUTOSIZE)

cv2.putText(im, 'motion', (20, 20), cv2.FONT_HERSHEY_COMPLEX_SMALL, 1, (255,
255, 255))
cv2.imshow('motion image', im)
cv2.waitKey(1000)
cv.DestroyWindow(winname)

invid = cv2.VideoCapture('LabVideo.MOV')
width = int(invid.get(cv.CV_CAP_PROP_FRAME_WIDTH))
height = int(invid.get(cv.CV_CAP_PROP_FRAME_HEIGHT))
fps = int(invid.get(cv.CV_CAP_PROP_FPS))
length = int(invid.get(cv.CV_CAP_PROP_FRAME_COUNT))

for i in range(length):
    _, im = invid.read()
    if i % 3 == 0:
        cv2.imshow('fastForward', im)
        cv2.waitKey(100)

del invid
cv2.destroyAllWindows()

# Lab 3

IMAGE_1_FILENAME = 'LabPhoto1.jpg'
IMAGE_1_GRAY_FILENAME = 'grayImage1.jpg'
IMAGE_2_FILENAME = 'LabPhoto2.jpg'
IMAGE_2_GRAY_FILENAME = 'grayImage2.jpg'
IMAGE_1_TRACKING_FILENAME = 'LabPhotoTracking1.jpg'
IMAGE_2_TRACKING_FILENAME = 'LabPhotoTracking2.jpg'

TEXT_ORIGIN = (20, 20)
TEXT_FONT = cv2.FONT_HERSHEY_COMPLEX_SMALL
TEXT_FONT_SCALE = 1
TEXT_FONT_COLOR = (255, 255, 255)

TRACKING_CORNER_COUNT = 200
TRACKING_QUALITY_LEVEL = 0.001
TRACKING_MIN_DISTANCE = 9.0

im1 = cv2.imread(IMAGE_1_FILENAME, cv2.CV_LOAD_IMAGE_COLOR)
im2 = cv2.imread(IMAGE_2_FILENAME, cv2.CV_LOAD_IMAGE_COLOR)

im1_height, im1_width, im1_depth = im1.shape
im2_height, im2_width, im2_depth = im2.shape

print 'Dimensions of', IMAGE_1_FILENAME, ': ', im1_height, 'x', im1_width
print 'Dimensions of', IMAGE_2_FILENAME, ': ', im2_height, 'x', im2_width

grImg1 = cv2.cvtColor(im1, cv2.COLOR_BGR2GRAY)
cv2.putText(grImg1, IMAGE_1_FILENAME, TEXT_ORIGIN, TEXT_FONT,
TEXT_FONT_SCALE, TEXT_FONT_COLOR)
cv2.imshow(IMAGE_1_GRAY_FILENAME, grImg1)
cv2.imwrite(IMAGE_1_GRAY_FILENAME, grImg1)

```

```

grImg2 = cv2.cvtColor(im2, cv2.COLOR_BGR2GRAY)
cv2.putText(grImg2, IMAGE_2_FILENAME, TEXT_ORIGIN, TEXT_FONT,
TEXT_FONT_SCALE, TEXT_FONT_COLOR)
cv2.imshow(IMAGE_2_GRAY_FILENAME, grImg2)
cv2.imwrite(IMAGE_2_GRAY_FILENAME, grImg2)

feat1 = cv2.goodFeaturesToTrack(grImg1, TRACKING_CORNER_COUNT,
TRACKING_QUALITY_LEVEL, \
TRACKING_MIN_DISTANCE).reshape((-1, 2))

criteria = (cv.CV_TERMCRIT_ITER | cv.CV_TERMCRIT_EPS, 80, 0.0001)
win = (3, 3) # actual size is 3*2+1 x 3*2+1
zero_zone = (-1, -1) # no dead zone
cv2.cornerSubPix(grImg1, feat1, win, zero_zone, criteria)

feat2 = np.copy(feat1)
feat2, status, err = cv2.calcOpticalFlowPyrLK(grImg1, grImg2, feat1, feat2)

print feat1
print feat2

im1 = cv2.imread(IMAGE_1_FILENAME, cv2.CV_LOAD_IMAGE_COLOR)
cv2.namedWindow('Picture1')
for (x, y) in feat1:
    cv2.circle(im1, (int(x), int(y)), 3, (255, 255, 255), -1)

cv2.imshow("Picture1", im1)
cv2.imwrite(IMAGE_1_TRACKING_FILENAME, im1)

im2 = cv2.imread(IMAGE_2_FILENAME, cv2.CV_LOAD_IMAGE_COLOR)
cv2.namedWindow('Picture2')
for (x, y) in feat2:
    cv2.circle(im2, (int(x), int(y)), 3, (255, 255, 255), -1)

cv2.imshow("Picture2", im2)
cv2.imwrite(IMAGE_2_TRACKING_FILENAME, im2)

if cv2.waitKey(0) == 27:
    cv2.destroyAllWindows() # Save marked images.

```

### Question 7

Picture 1

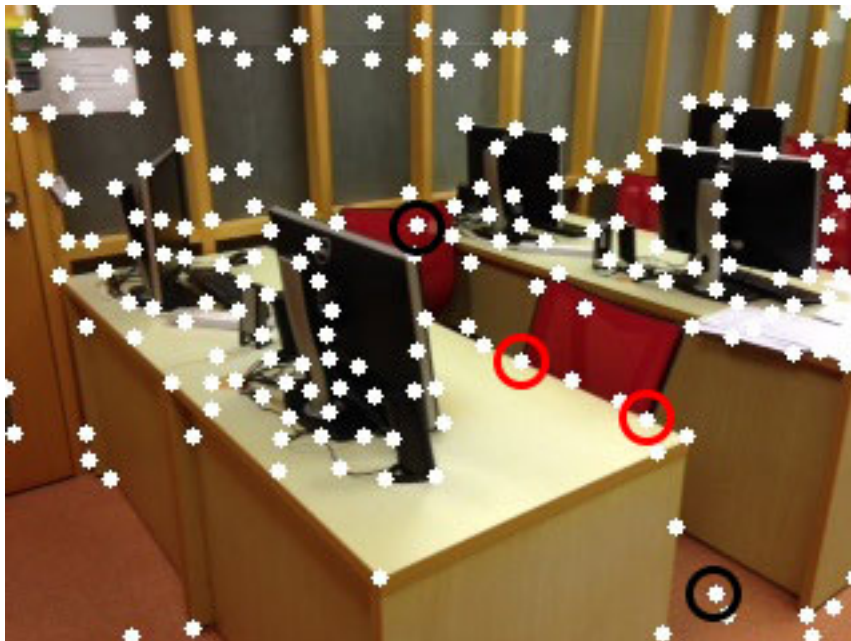


Picture 2



**Question 8**

Picture 1 Marked (Points in black circles)



The tracking was wrong for the points circled in black because they were not very distinct corners.

The points circled in red are good corners but they aren't real features. They are created from the superposition of two objects. Hence they should not be tracked.