機器視覺作業報告

放頸鬆-肩頸放鬆小遊戲

姓名:朱冠穎
學號:00457043
日期:06/23

方法

想法:

在一個資訊發達的年代，人們常常因為工作需要長時間坐在螢幕前面，許多上班族需要一天 8 小時都在電腦前，更不用說工程師之類的職業更是可怕。這現象造成許多現代的肩頸問題，而只要能跟著我這套「放頸鬆」做舒緩，簡單的兩分鐘讓你的脖子得到舒緩，可以更有效率的工作且避免肩頸痠痛!

演算步驟:

說明如註解

```
from __future__ import print_function
import numpy as np
from numpy import pi, sin, cos
import cv2
# built-in modules
from time import clock
# local modules
#import common
import copy
import cv2
#import video
from matplotlib import pyplot as plt
from PIL import Image
#載入引導方向的圖片(上,下,左,右)
global imUp #(50,78,3)
imUp= cv2.imread("up.png")
global imDown #(50,78,3)
imDown= cv2.imread("down.png")
global imLeft #(78,50,3)
imLeft= cv2.imread("Left.png")
global imRight #(78,50,3)
imRight= cv2.imread("Right.png")
```

```
#宣告整份檔案需要的常用變數
global remaintime #遊戲剩餘時間
remaintime = 200
global showString #遊戲引導敘述
showString  = ["See left and right","See up and down"]
#目前模式 模式主要有二:
#1.左右放鬆肩頸
#2.上下方鬆肩頸
global nowMode
nowMode = 0
#目前顯示引導方向(0,1,2,3) - (左,右,上,下)
global nowForward
nowForward = 0
#該輪已經執行的時間
global forwardTime
forwardTime = 0
#計算遊戲的分數
global score
score = 0
#遊戲的執行輪數 - (左,右) (上,下)
global loop
loop = 0
```

```
#使用者所看到的畫面-呈現畫面(不含運算線條)
def user_flow(img2):
    #根據不同模式有不同圖片對應位置
    if nowForward == 0:
        img2[220:270,50:128] = imLeft
    elif nowForward ==1:
        img2[220:270,500:578] = imRight
    elif nowForward ==2:
        img2[70:148,290:340] = imUp
    elif nowForward ==3:
        img2[350:428,290:340] = imDown
    cv2.putText(img2,str(remaintime//10) + "   score:" + str(score),(0,30),cv2.FONT_HERSHEY_PLAIN,2,(255,255,255),3)
    cv2.putText(img2,showString[nowMode],(0,70),cv2.FONT_HERSHEY_PLAIN,2,(100,100,255),3)
    return img2
```

```
#使用者看到的運算畫面 - 技術畫面(有曲線的運算線條可以看到運動方向)
def draw_flow(img, flow, step=16):
```

```
#分析亮度變化分布
def analyze_flow(flow,flow_len):
```

```
#主程式
if __name__ == '__main__':
    #抓取攝影頭
    cap  = cv2.VideoCapture(0)
    #抓取攝影頭影像的高,寬和fps,現在的shot編號
    height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT));
    width  = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH));
    fps    = int(cap.get(cv2.CAP_PROP_FPS));
    shot_idx = 0
```

```
    #若攝影頭是打開狀態
    while cap.isOpened():
        #用變數接收boolean ret,和 這一個frame img
        ret, img = cap.read()
        #複製一個img 另一個是拿來放運算frame
        img2 = copy.copy(img)
        #呈現視窗設定為 480,640且位置從(0,0,0,0)開始
        resultPicture = Image.new('RGBA', (480,640), (0, 0, 0, 0))
```

```
        if ret == True:
            #第一張frame
            if shot_idx == 0:
                #轉換成gray level做影響處理
                pregray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
                imgf    = img
                res     = False
            else:
                gray    = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
                #利用亮度計算運動量 Function
                #(前一張圖,後一張圖,光流,兩層之間的尺度關係,層數,均值範圍,迭代次數,像素範圍大小,高斯標準差)
                flow    = cv2.calcOpticalFlowFarneback(pregray, gray, None, 0.5, 3, 15, 3, 5, 1.2, 0)
                pregray = gray
                #分析亮度變化得出來的分布狀態存在hist
                res,hist,pt1,pt2 = analyze_flow(flow,10)
                #用flow的結果繪製在結果test windows上
                imgf    = draw_flow(img, flow)
                #變化有大於等於一個格子的變化量
                if res and  pt2[1]-pt1[1] >= img.shape[0]/6 and pt2[0]-pt1[0]>=img.shape[1]/5:
                    cv2.rectangle(imgf,pt1,pt2,(0,0,255),2);
                    #找出變化量最多的方向
                    mv = np.max(hist)
                    h  = imgf.shape[0]
                    for i in range(0,8):
                        cv2.rectangle(imgf,(i*10,h-hist[i]/mv*100),(i*10+10,h),(0,255,255),-2)
                    cv2.rectangle(imgf,pt1,pt2,(0,0,255),2);
                    #計算方向的分布大小
                    up    = hist[1]+hist[2]
                    down  = hist[5]+hist[6]
                    right = hist[3]+hist[4]
                    left  = hist[0]+hist[7]
                    #遊戲加分判斷依據,印出方向變化量最多的方向
                    #遊戲加分判斷依據,印出方向變化量最多的方向
                    if left/2 > right and left/2 > down and left/2 > up:
                        if(nowForward == 1):
                            score += 50
                        elif(nowForward == 0):
                            score += 10
                        cv2.putText(imgf,'right',(0,50),cv2.FONT_HERSHEY_PLAIN,4,(255,255,255),3)
                    elif right > left and right > down and right > up:
                        if(nowForward == 0):
                            score += 50
                        elif(nowForward == 1):
                            score += 10
                        cv2.putText(imgf,'Left',(0,50),cv2.FONT_HERSHEY_PLAIN,4,(255,255,255),3)
                    elif down/2 > left and down/2 > right and down/2 > up:
                        if(nowForward == 3):
                            score += 50
                        elif(nowForward == 2):
                            score += 10
                        cv2.putText(imgf,'Down',(0,50),cv2.FONT_HERSHEY_PLAIN,4,(255,255,255),3)
                    elif up/2 > left and up/2 > right and up/2 > down:
                        if(nowForward == 2):
                            score += 50
                        elif(nowForward == 3):
                            score += 10
                        cv2.putText(imgf,'Up',(0,50),cv2.FONT_HERSHEY_PLAIN,4,(255,255,255),3)
            #將圖片呈現
            cv2.imshow('cap Test',imgf);
            #將原圖做使用者介面呈現
            user_flow(img2)
            cv2.imshow('Relax shoulder and neck',img2);
```

```
        cv2.waitKey(1)
        shot_idx = 1
        #計算每20秒變換模式(左右)(上下)
        if(remaintime < 10):
            remaintime = 200
            nowMode = (nowMode+1)%2
            loop +=1
            #使用者分數結果呈現
            if loop == 4:
                if score > 10000:
                    cv2.putText(img2,"Health",(100,200),cv2.FONT_HERSHEY_PLAIN,5,(255,50,50),3)
                else:
                    cv2.putText(img2,"Not Bad",(100,200),cv2.FONT_HERSHEY_PLAIN,5,(255,50,50),3)
                cv2.imshow('userShow',img2);
                break
        #計算每1秒讓圖片變換(左,右)(上,下)
        else:
            remaintime -= 1
            forwardTime += 1
            #see left 0 and right 1
            if(forwardTime%10 == 0):
                if(nowMode == 0):
                    if(nowForward == 0):
                        nowForward = 1
                    else:
                        nowForward = 0
                #see up 2 and down 3
                else:

                    if(nowForward == 2):
                        nowForward = 3
                    else:
                        nowForward = 2
    else:
        break
```

結果

透過照著螢幕的指示和攝影頭做的互動，讀取動作並判斷。

預期與使用者對應方式	得分
系統提示之方向(上,下)(左,右)圖片出現時使用者和提示方向相同。Ex:系統顯示「左」，使用者跟著轉左邊	50
系統提示之方向(上,下)(左,右)圖片出現時使用者和提示方向相反。Ex:系統顯示「左」，使用者跟著轉右邊	10
系統提示之方向(上,下)(左,右)圖片出現時使用者和提示方向完全不同。Ex:系統顯示「左」，使用者卻動上下	0

使用者依據不同得分會呈現不同結果

使用者分數	得到結果
>=10000	Health
<10000	Not Bad

使用提示圖:

超過 10000 分時得到 Health 結果:

Youtube 連結:
https://youtu.be/zpJCgC5SY6s

結論

這次是訓練用攝影頭做互動的 AR 小遊戲，一開始想用 PIL 去貼圖片發現那圖片是 nparray 所以嘗試很多種方式去貼圖片，其中嘗試過 paste, mix, blend 但是都因為格式不一樣造成無法貼上，最後想到圖片是陣列，乾脆直接用陣列的改法[50:128, 400, 450]這種方式去修改就成功了，算是繞了一大圈才完成。
其中遊戲也需要蠻多全域變數的使用，以及遊戲每一個瞬間都需要做判斷，需要透過一輪一輪去控制模式。
最重要的大概是真的理解 calcOpticalFlowFarneback 這個光流的向量判斷法的演算步驟，去解析透過兩張前後圖判別光源變化產生的向量。
分析向量分布的 hist 其實應該讓他更集中化一點或是要改判斷式，不然其實有時候讀到的誤差還是有點大(ex:往上動卻印出 left)。
這個專案應該還可以有更多的擴展性，像是每一輪 loop 要切換模式可以加入提示音樂，或是加入其他特殊元素增加遊戲互動性和變化性。

參考文獻

全域變數使用
https://dotblogs.com.tw/chris0920/2010/10/21/18489
putText 編碼問題

http://hant.ask.helplib.com/python/post_3228604

PIL 相關研究

https://blog.csdn.net/sunny2038/article/details/9057415

https://blog.csdn.net/GZHermit/article/details/72758641

https://blog.csdn.net/icamera0/article/details/50706615

https://stackoverflow.com/questions/902761/saving-a-numpy-array-as-an-image

http://www.scipy-lectures.org/advanced/image_processing/

https://medium.com/@Syashin/python-%E7%85%A7%E7%89%87-%E7%B0%BD%E5%90%8D%E6%AA%94%E5%9C%96%E7%89%87%E5%90%88%E6%88%90%E5%B7%A5%E5%85%B7-e4df88f99994

https://stackoverflow.com/questions/19098104/python-opencv2-cv2-wrapper-get-image-size/19098258

calcOpticalFlowFarneback

https://blog.csdn.net/ironyoung/article/details/60884929