

用Rabboni來畫畫

Fun
Learning

臺北市龍山國中洪瑞甫
臺北市民族國小戴貝羽



抽象創作：自動性技法

畫布釘在地板上或牆上，然後隨意
在畫布上潑灑顏料，任其在畫布上
滴流，創造出縱橫交錯的抽象線條
效果。



課堂實際操作困難

- 人數與場地限制
- 材料工具準備不易

平板或滑鼠模擬

- 沒有創作時的肢體動感
- 無法體驗顏料潑灑的快感



實作工具

軟體：**Python**

硬體：**rabboni**



使用模組

rabboni_multi_sdk

asyncio

mediapipe

pyautogui

time

random

numpy

第一版:用**rabboni**姿態來控制

開始拖拉畫線
Accx=0.1
Accy=-0.9
Accz=0.1

往上轉



往左轉

往右轉

上一頁
Shift+P
Accx=0.9
Accy=-0.2
Accz=-0.0

下一頁
Shift+N
Accx=-0.7
Accy=0.2
Accz=0.01

往下轉

結束拖拉畫線
Accx=0.1
Accy=0.9
Accz=-0.01

翻過來
Ctrl+p
Accx=0.1
Accy=-0.2
Accz=-0.9

第一版:用rabboni姿態來控制

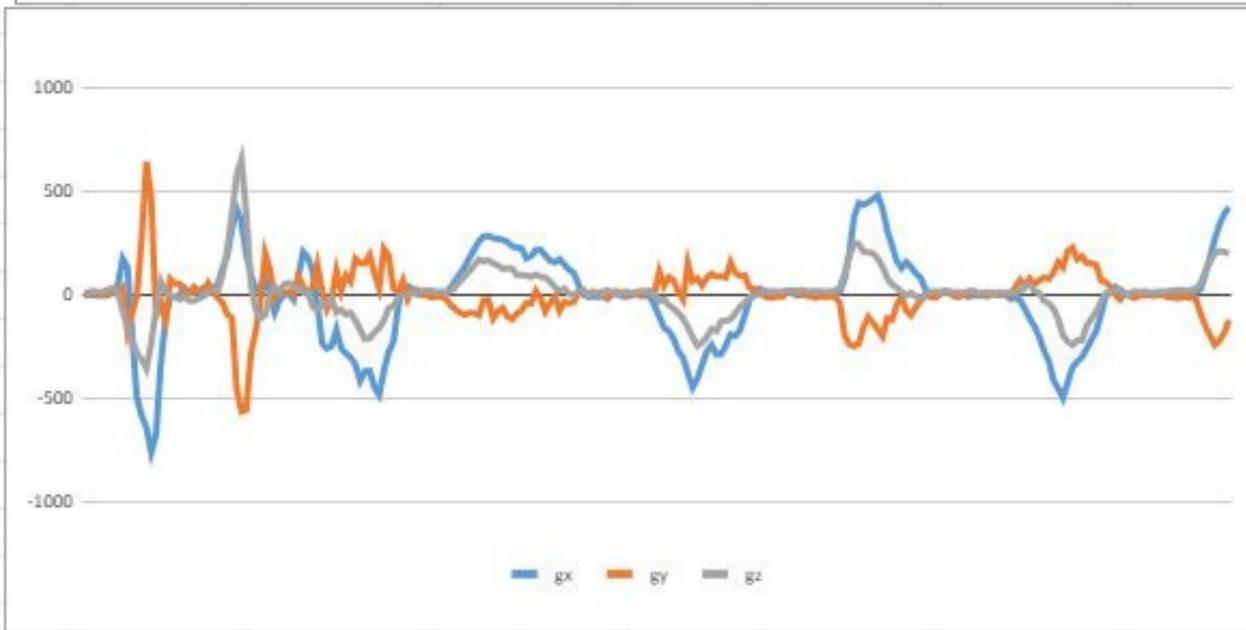
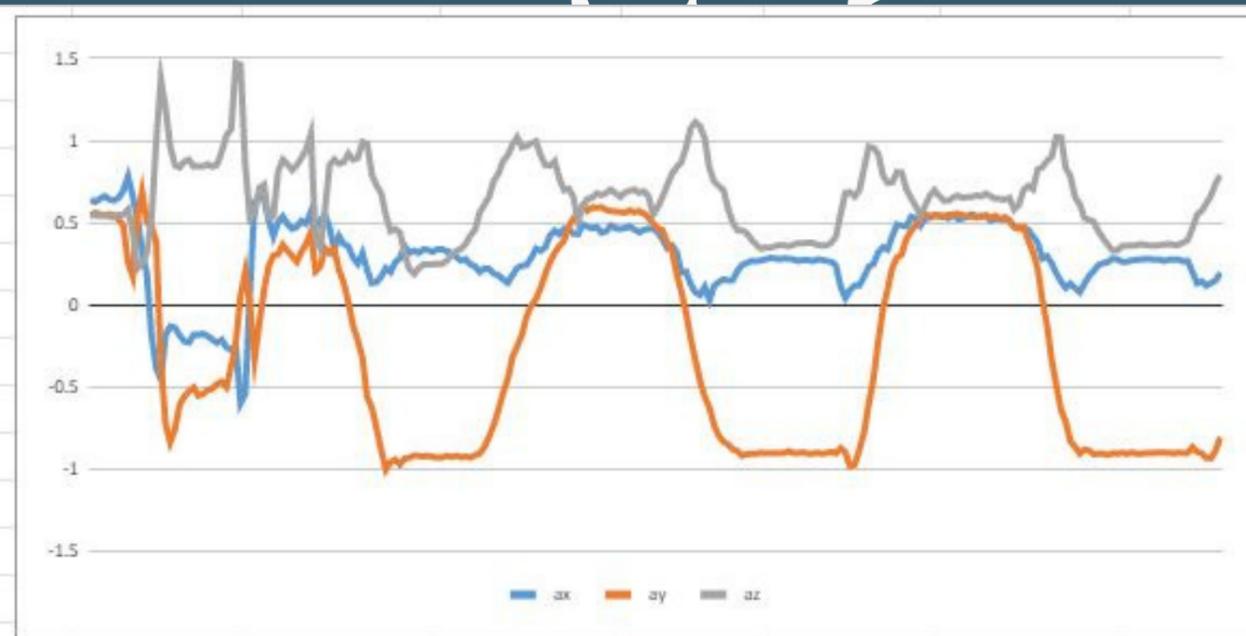
希望動作能 大一點



第二版:用**rabboni**移動來控制

1. rabboni 資料分析

| | ax | ay | az | gx | gy | gz |
|----|--------|--------|-------|----------|----------|----------|
| 左上 | 0.638 | 0.543 | 0.545 | -7.874 | -4.028 | 5.463 |
| | 0.626 | 0.56 | 0.544 | -2.319 | 1.099 | 12.421 |
| | 0.644 | 0.551 | 0.54 | -2.991 | 3.143 | 16.113 |
| | 0.661 | 0.542 | 0.548 | -6.012 | 4.456 | 10.101 |
| | 0.641 | 0.552 | 0.537 | -6.134 | -4.12 | 14.557 |
| | 0.634 | 0.543 | 0.531 | -3.632 | 2.045 | 22.4 |
| | 0.651 | 0.52 | 0.548 | 21.362 | 16.724 | 29.907 |
| | 0.694 | 0.481 | 0.549 | 44.464 | -4.7 | 14.313 |
| | 0.781 | 0.259 | 0.58 | 172.455 | 26.886 | -80.902 |
| | 0.662 | 0.174 | 0.445 | 131.714 | -176.422 | -151.093 |
| | 0.524 | 0.527 | 0.221 | -201.691 | -85.419 | -229.645 |
| 下甩 | 0.322 | 0.687 | 0.248 | -502.35 | -6.012 | -285.583 |
| | 0.192 | 0.499 | 0.316 | -588.257 | 274.292 | -314.453 |
| | -0.184 | 0.476 | 0.525 | -642.792 | 642.67 | -358.398 |
| | -0.385 | 0.384 | 0.99 | -763.428 | 461.945 | -235.779 |
| | -0.454 | -0.345 | 1.373 | -680.45 | -97.137 | -72.296 |
| | -0.184 | -0.726 | 1.195 | -338.074 | -40.497 | 61.157 |
| | -0.132 | -0.836 | 0.961 | -124.42 | -126.678 | 14.435 |
| 右下 | -0.139 | -0.758 | 0.848 | -11.505 | 74.066 | -6.256 |
| | -0.186 | -0.609 | 0.836 | -12.543 | 52.765 | -12.634 |
| | -0.223 | -0.559 | 0.871 | -3.937 | 50.415 | -23.041 |
| | -0.229 | -0.526 | 0.883 | 30.64 | 2.014 | -12.726 |
| | -0.181 | -0.503 | 0.841 | 7.141 | 5.28 | -30.182 |
| | -0.183 | -0.553 | 0.842 | 7.019 | 36.621 | -31.86 |
| | -0.176 | -0.543 | 0.841 | 12.421 | 16.968 | -19.348 |
| | -0.191 | -0.518 | 0.855 | 32.349 | 1.556 | -9.155 |
| | -0.212 | -0.511 | 0.84 | 36.072 | 52.979 | 7.111 |
| | -0.232 | -0.481 | 0.853 | 30.426 | 15.137 | 23.163 |
| | 0.213 | 0.468 | 0.934 | 44.25 | -18.463 | 16.296 |



第二版:用**rabbit**移動來控制

2.歸納出特徵值(一)

| | ax | ay | az | gx | gy | gz |
|----|-------------|------------|-----------|---------------|---------------|---------------|
| 右上 | -0.42~-0.65 | -0.4~-0.55 | 0.6~0.85 | abs() $<$ 100 | abs() $<$ 100 | abs() $<$ 100 |
| 左下 | -0.35~-0.58 | -0.6~-0.85 | 0.4~0.85 | abs() $<$ 100 | abs() $<$ 100 | abs() $<$ 100 |
| 左上 | 0.6~0.7 | 0.48~0.58 | 0.5~0.55 | abs() $<$ 100 | abs() $<$ 100 | abs() $<$ 100 |
| 右下 | -0.2~0 | -0.7~-0.4 | 0.8~1.1 | abs() $<$ 100 | abs() $<$ 100 | abs() $<$ 100 |

第二版:用**rabbit**移動來控制

2.歸納出特徵值(二)

| | ax | ay | az | gx | gy | gz |
|------|-----------|-----------|-----------|-----------|-----------|-----------|
| 慢速下甩 | -0.2~0.8 | -0.9~0.7 | 0~1.4 | -770~-300 | 400~500 | -200~-400 |
| 慢速上甩 | -0.6~0.1 | -0.5~0.25 | 0~1 | 150~450 | -400~-600 | 350~700 |

水平左右

ax, ay, az 變化不大，**gz**變化最大

慢速往右 < -150 ，往左 > 150

快速往右 < -800 ，往左 > 800

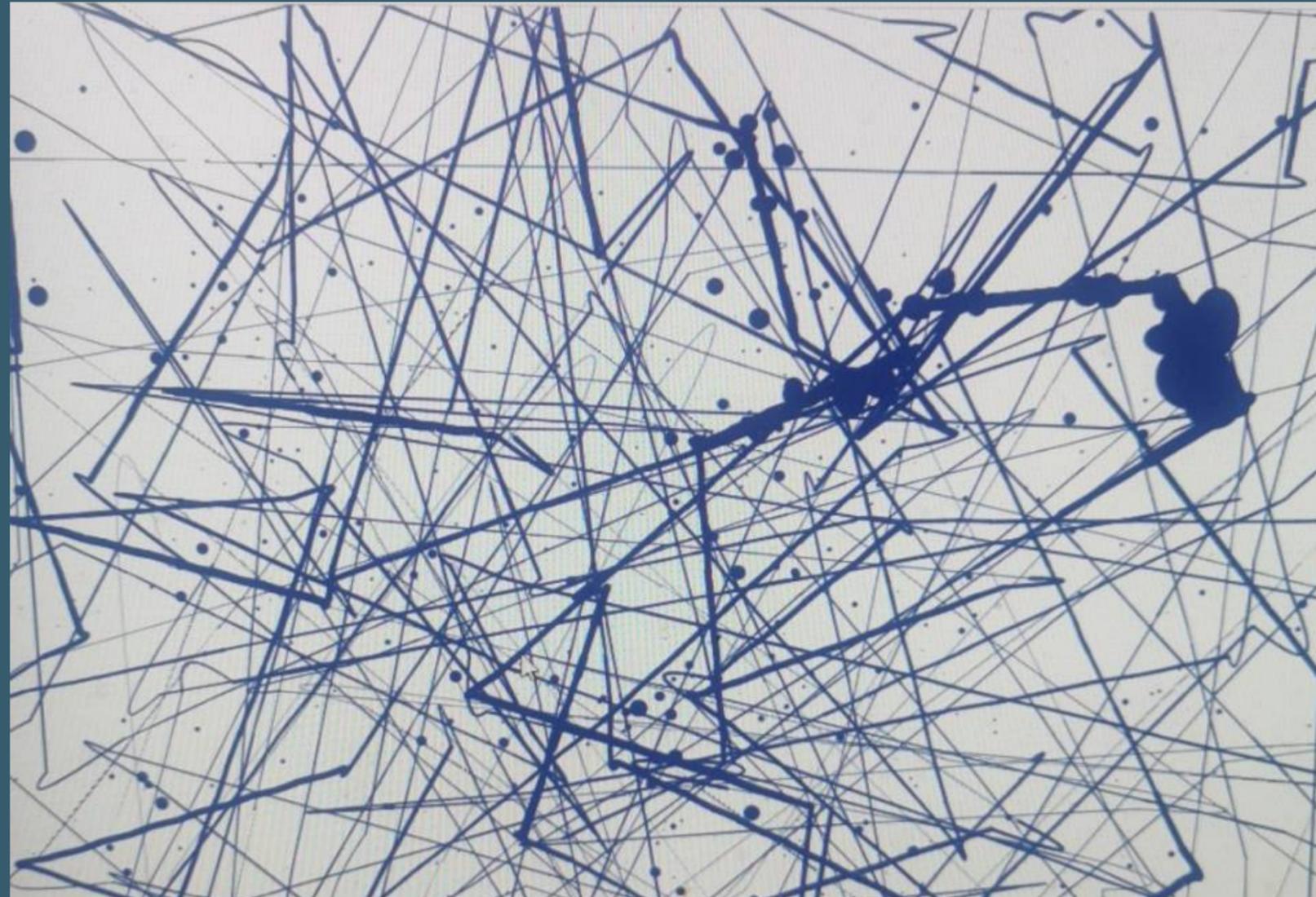
第二版:用**rabbit**移動來控制

3.依特徵值來撰寫程式

```
if (choice==1):
    if trigger:
        if send==0:
            send=1
            pyautogui.press('F5')
            stime=time.time()
        if (acc_list[0]<-0.7 and abs(acc_list[1])<0.3 and abs(acc_list[2])<0.3):
            if send==0:
                send=1
                pyautogui.keyDown('shift')
                pyautogui.press('n')
                pyautogui.keyUp('shift')
                stime=time.time()
            elif (acc_list[0]>0.7 and abs(acc_list[1])<0.3 and abs(acc_list[2])<0.3):
                if send==0:
                    send=1
                    pyautogui.keyDown('shift')
                    pyautogui.press('p')
                    pyautogui.keyUp('shift')
                    stime=time.time()
            elif (abs(acc_list[0])<0.3 and abs(acc_list[1])<0.3 and acc_list[2]<-0.7):
                if send==0:
                    send=1
                    pyautogui.keyDown('ctrl')
                    pyautogui.press('p')
                    pyautogui.keyUp('ctrl')
                    stime=time.time()
            elif (abs(acc_list[0])<0.3 and acc_list[1]<-0.7 and abs(acc_list[2])<0.3):
                drags=1
            elif (abs(acc_list[0])<0.3 and acc_list[1]>0.7 and abs(acc_list[2])<0.3):
                drags=0
        if send==1:
            if ((time.time()-stime)>1.5):
                send=0
```

第二版:用**rabboni**移動來控制

沒有甩動的感覺



第三版:用**rabboni**甩動來控制

1. 尋找自動性技法網站的規則

移動速度越慢，產生的甩痕越大



第三版:用**rabboni**甩動來控制

1.當偵測到**rabboni**往右甩時

螢幕左邊產生大點，線條往右邊隨機延伸1到3條

2.當偵測到往**rabboni**往左甩時

螢幕右邊產生大點，線條往左邊隨機延伸1到3條

3.當偵測到甩動時

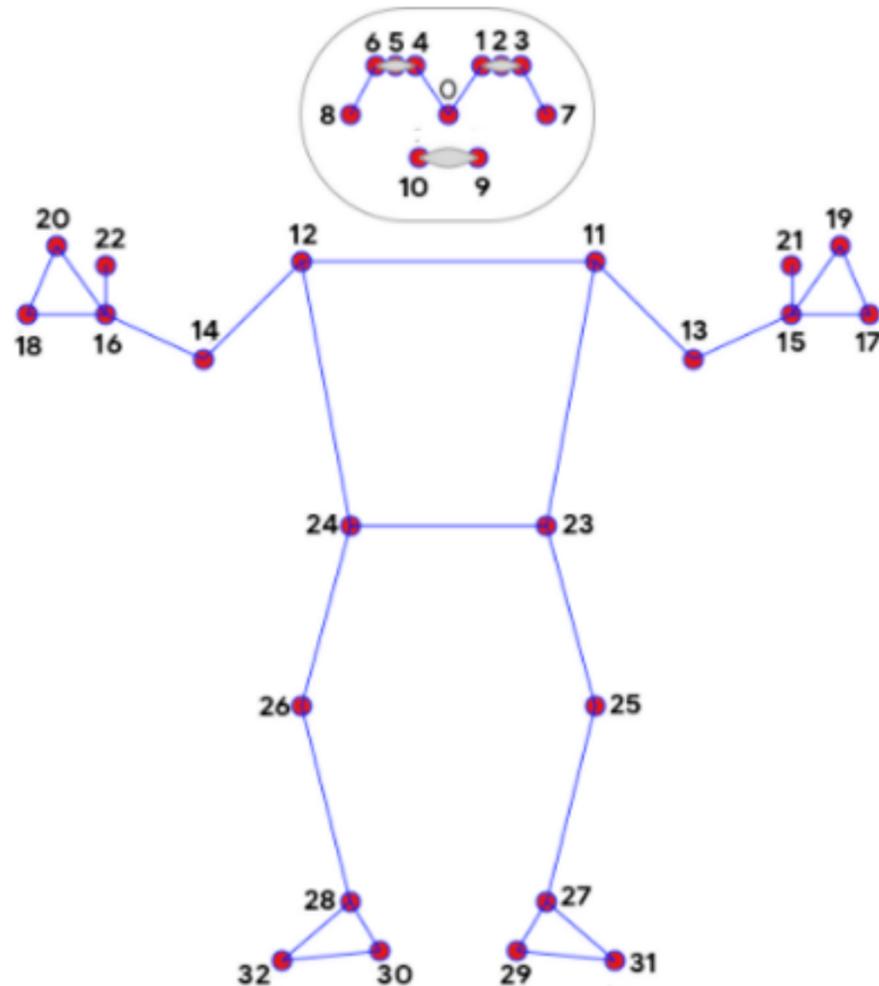
螢幕上方產生大點，線條往下邊隨機延伸1到3條

第四版: 用mediapipe肢態辨識
+rabboni甩動來控制

1. 只有使用rabboni，很難抓到較小的肢態變化
2. rabboni在加速度偵測很敏銳
3. 結合肢體辨識與rabboni兩者的優點來設計

第四版:用mediapipe肢態辨識

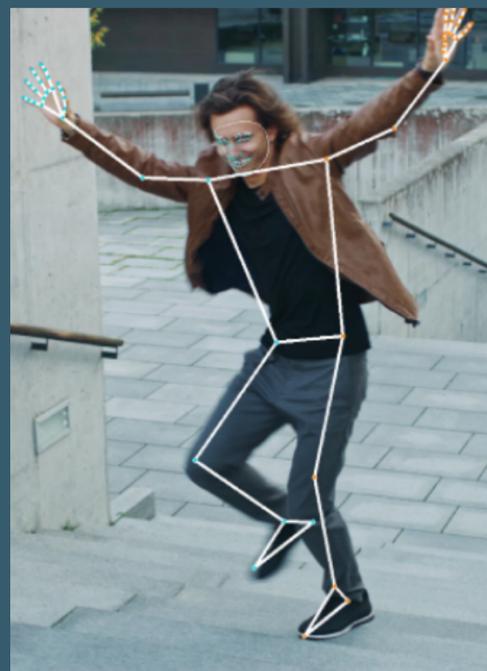
可以辨識出身體32個特徵點



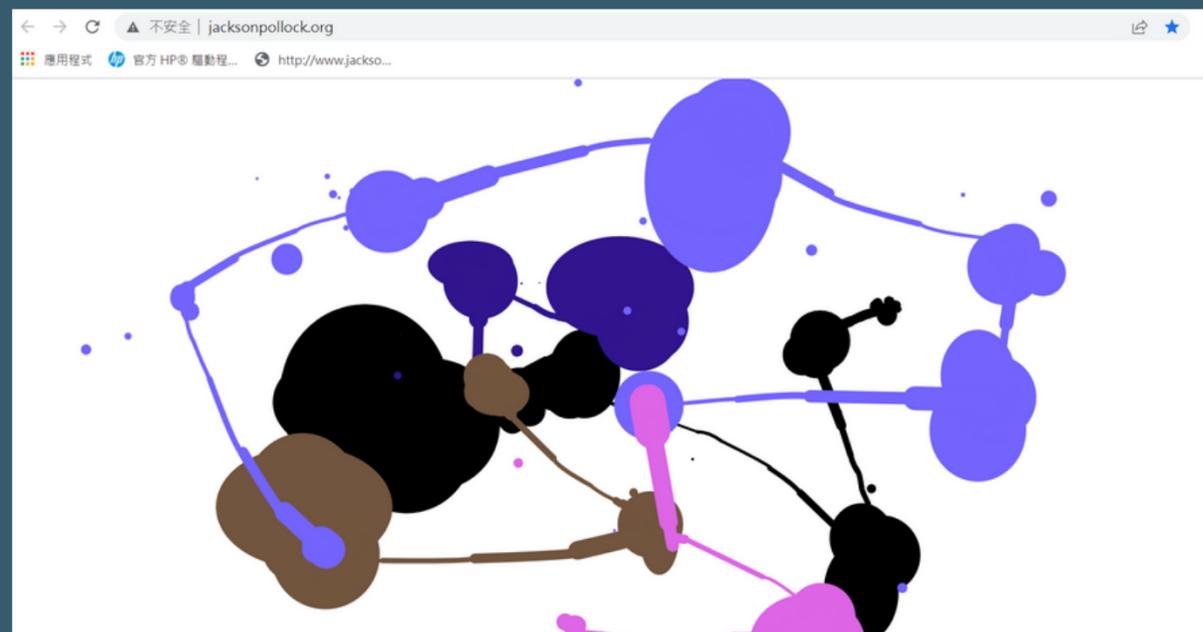
- | | |
|--------------------|----------------------|
| 0. nose | 17. left_pinky |
| 1. left_eye_inner | 18. right_pinky |
| 2. left_eye | 19. left_index |
| 3. left_eye_outer | 20. right_index |
| 4. right_eye_inner | 21. left_thumb |
| 5. right_eye | 22. right_thumb |
| 6. right_eye_outer | 23. left_hip |
| 7. left_ear | 24. right_hip |
| 8. right_ear | 25. left_knee |
| 9. mouth_left | 26. right_knee |
| 10. mouth_right | 27. left_ankle |
| 11. left_shoulder | 28. right_ankle |
| 12. right_shoulder | 29. left_heel |
| 13. left_elbow | 30. right_heel |
| 14. right_elbow | 31. left_foot_index |
| 15. left_wrist | 32. right_foot_index |
| 16. right_wrist | |

第四版:用mediapipe肢態辨識

將第16個特徵點(右手腕)，做座標轉換



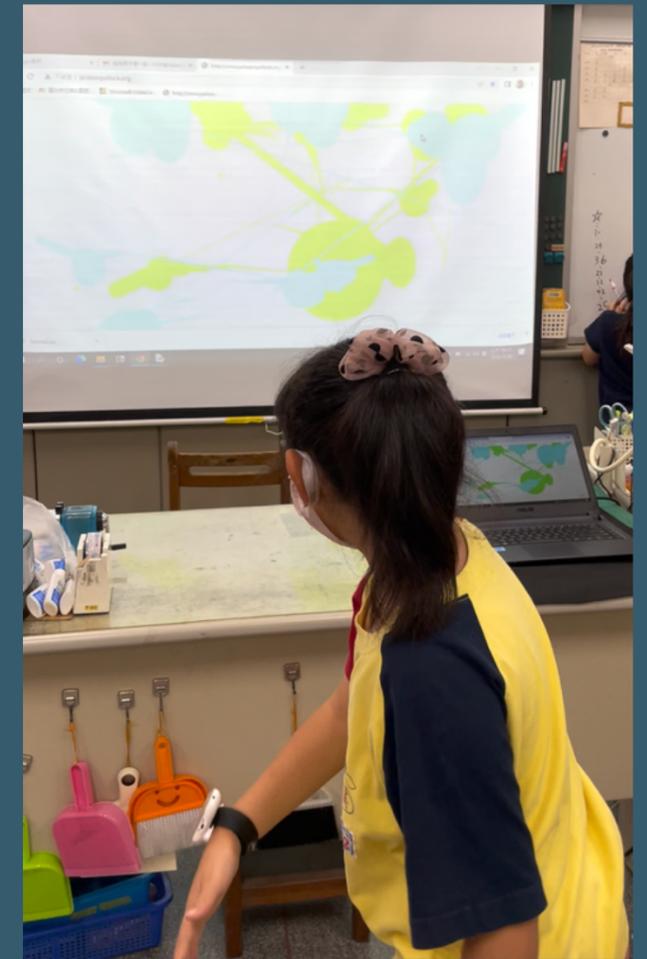
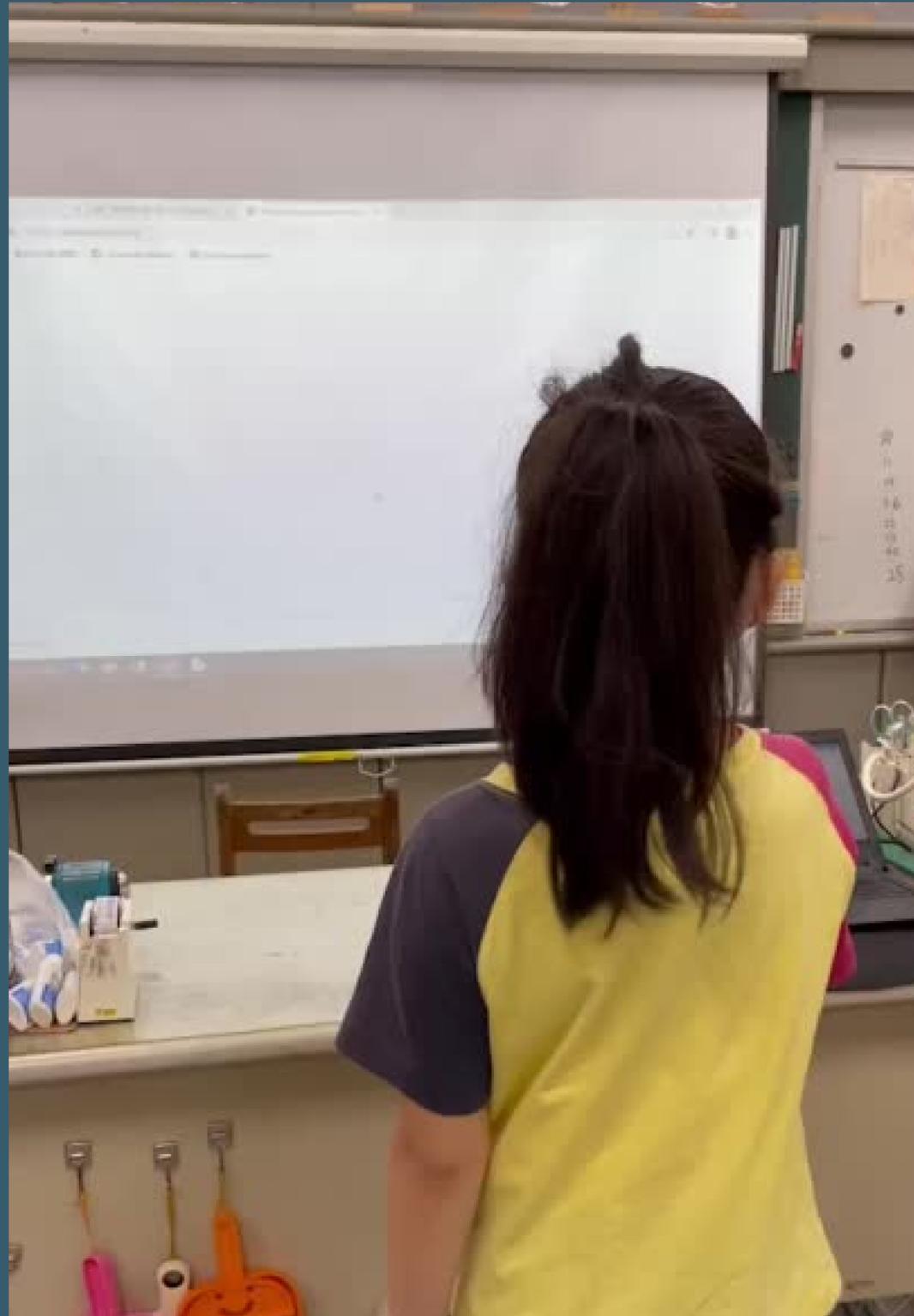
辨識畫面
640x480



自動性技法網頁1920x1080

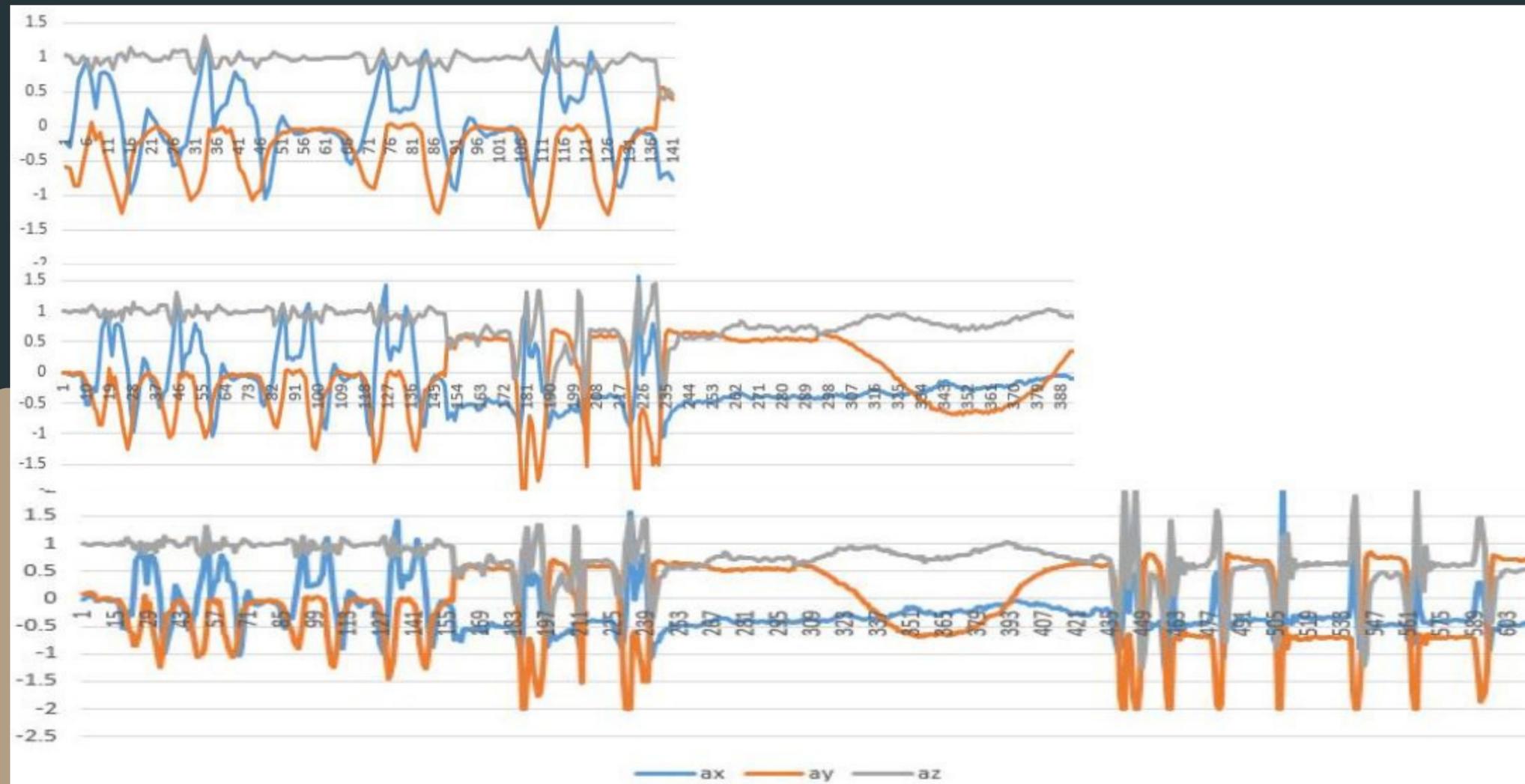
當rabboni有甩動，就加大甩動的點

上課實測



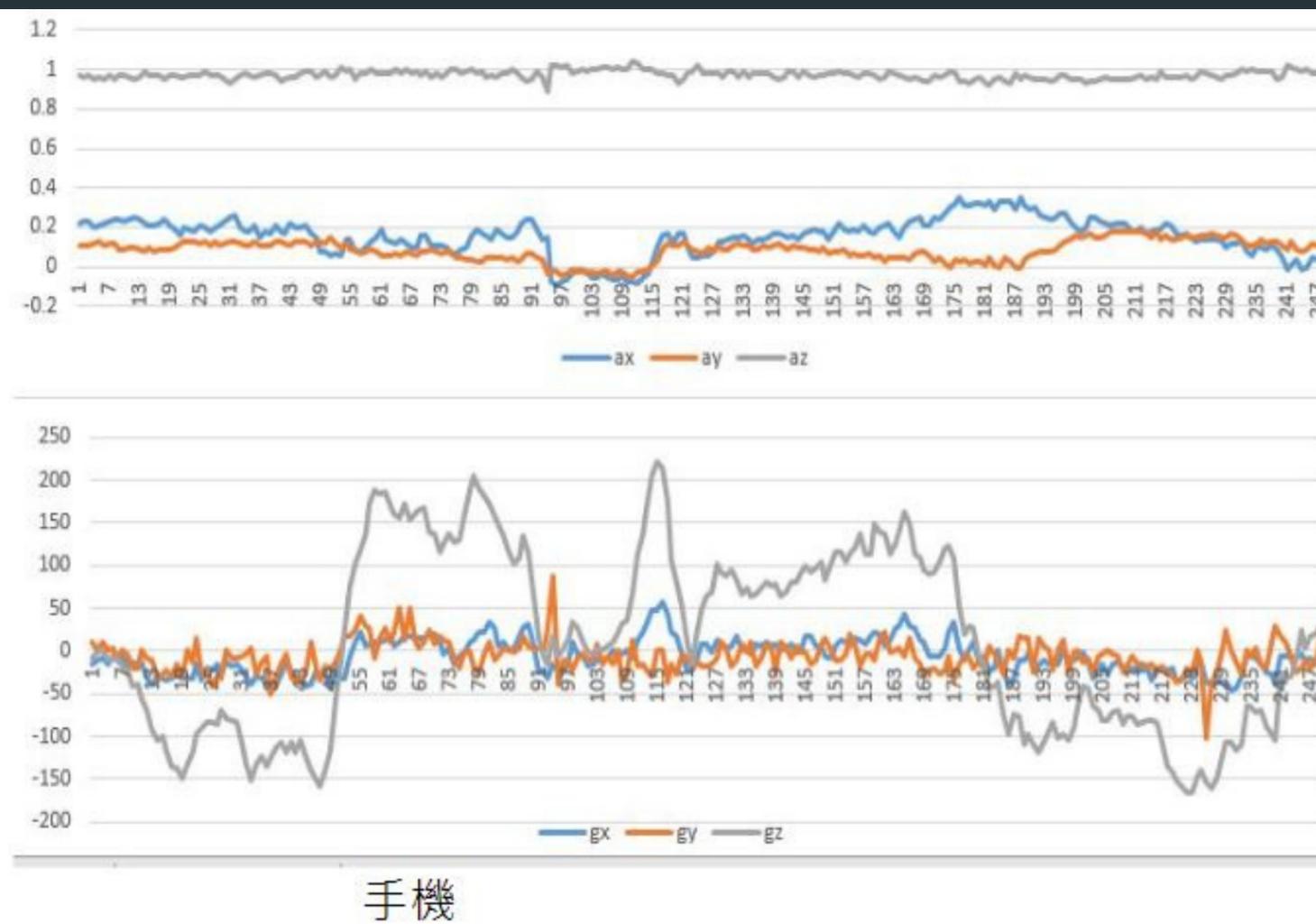
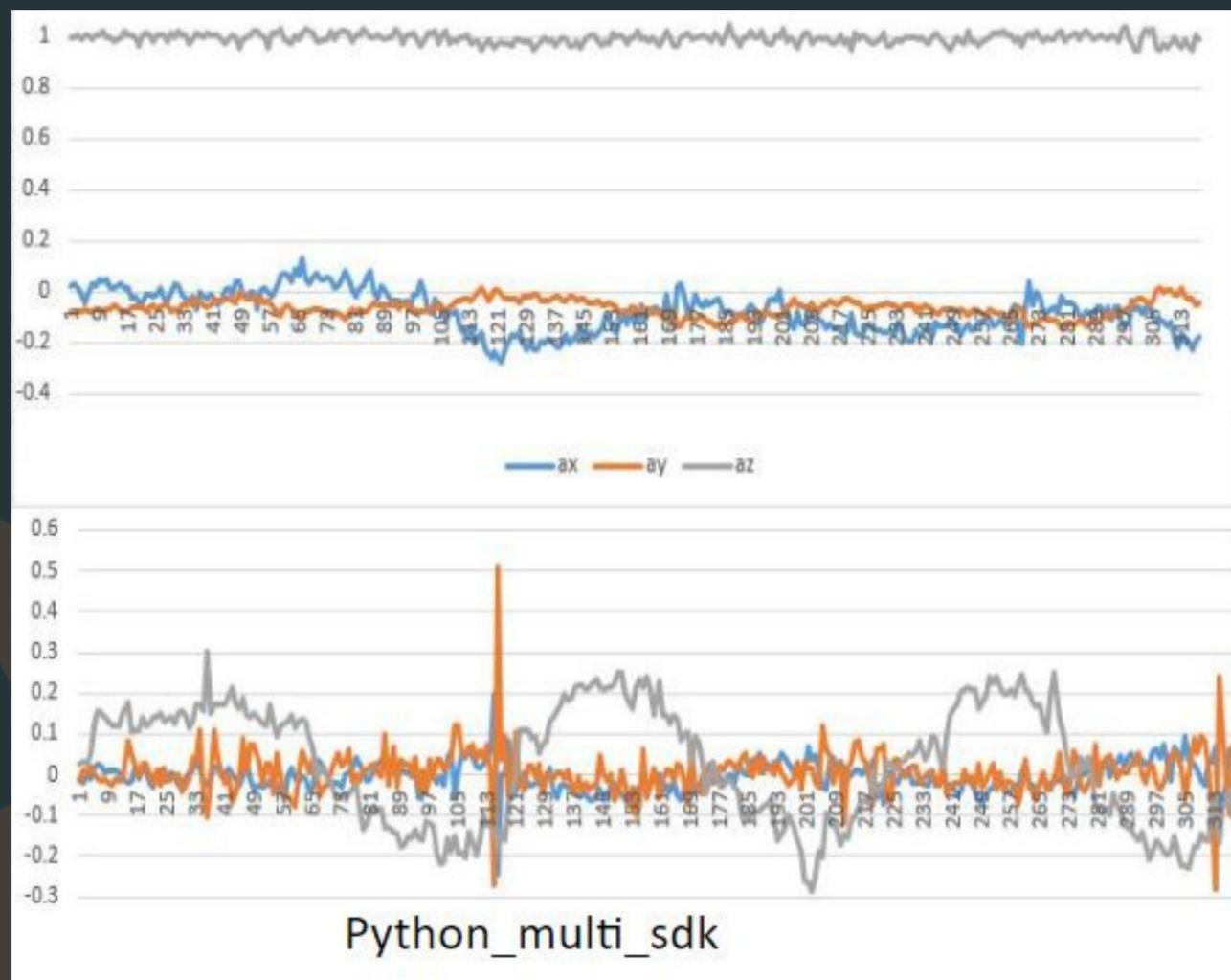
遇到的問題

1. 手機錄影檔案會越來越大，包含舊的資料



遇到的問題

2. 手機錄製跟python_multi_sdk 跑出來不一樣



3. 持續印出感測值會拖慢偵測的結果



Thank you for your attention.