

# 第十二屆旺宏科學獎

## 成果報告書

參賽編號：SA12-092

作品名稱：手機加速度感測器之動作辨識與低頭族應用

姓名：張郢展

關鍵字：感測器、動作辨識、低頭族

# 摘要

不管是在交通工具上，還是走路行進間使用手機或平板電腦，熱潮延續至今衍生出「低頭族」的新名詞。醫師發現每十名成人近視患者就有一到兩人『假性近視』，較行動裝置流行前人數**暴增十倍**；近來骨科門診發現，後天姿勢不良造成『**脊椎側彎**』加重的患者**激增2成**，甚至以前只有在老年人身上發生的眼疾『**黃斑部病變**』，最近也發生在年輕人身上，推估和長期不當使用3C產品有關。最近手機中置入微機電感測器，可以做出很多有用的應用程式，因此，我希望研究感測器與低頭族的應用。

## 一、健康方面：使用過久或光線過強下使用產生眼睛與脊椎的問題

撰寫「**背景執行程式**」來計算使用者是否使用手機超過30分鐘，此程式完全不影響使用者使用手機，例如：玩遊戲、上網…等等，當使用時間剩下1分鐘時，會跳出提示訊息，使用者可以先把目前玩的遊戲或網路先暫時告一段落，時間到後，透過「**加速度感測器**」檢測使用者必須做的手臂運動，例如：30次上下伸展手臂健康操，健康操的動作1為「拿著手機舉起手臂過肩膀」，動作2為「讓手臂垂直放下」，完成後才可繼續使用手機，藉此『休息一下眼睛與僵化的手臂』，兼顧手機使用者的健康與玩樂手機的權益。

我也會透過「**環境光感測器**」得知有人在環境光很暗的情況下使用手機，因此，使用手機15分鐘後就會啟動要求使用者做手臂運動的功能，**讓使用者休息眼睛**，除此之外，還可自動調整螢幕亮度，**不讓螢幕光太強而造成眼睛更容易損傷**。

## 二、安全方面：邊走邊滑手機造成的意外發生的問題

我透過「**加速度感測器**」與「**陀螺儀**」來辨別使用者是否邊走路邊使用手機，如果在3秒內走了3步，就會跳出提醒視窗告知「走路使用手機危險」，因為暫時中斷了使用者使用手機，使用者或許就會停下腳步或是抬起頭來，也就恢復對環境的觀察能力以避免意外發生，按下「**確定鈕**」後，就可繼續使用。

## 三、加速度感測器的動作辨識：

可分辨出移動、轉動、傾斜角度。

## 預見未來：

如果，有一天有手機業者可以把本作品融入系統中，讓爸爸媽媽可以設定此功能，當兒童玩遊戲超過正常時間就休息眼睛，運運動，更不讓兒童邊走邊玩，相信，一定會讓小朋友更能兼顧玩與安全與健康與家長的需求。

## 壹、研究動機

最近報導關於手機的新聞很多著墨在微機電感測器上，讓手機能做更多的事，因此，我也想要來了解並做出一些應用程式。

低頭族也是最近很熱門的話題，不管是在交通工具上，還是走路行進間使用手機或平板電腦，熱潮延續至今衍生出「低頭族」的新名詞，因此，我就針對這兩方面做相關的研究。

### 一、六成五家長「低頭族」，親子溝通少

最新調查發現，七成八的家長回家使用網路時間比在工作場所還要高，高達六成五家長為所謂的「低頭族」，習慣使用智慧型手機上網與玩遊戲，已經超越筆電。專家呼籲家長應以身作則，每天「休網」1小時來增加親子互動時間。【註1】

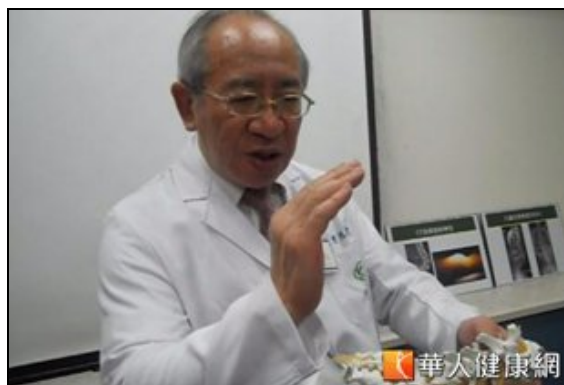


圖 1-1 兒童有樣學樣 也很會【滑】 圖 1-2 康寧醫院骨科陳博光教授說明低頭族現象

### 二、低頭族 成人假性近視多 10 倍

醫師發現每十名成人近視患者就有一到兩人假性近視，較行動裝置流行前人數暴增十倍。馬偕醫院眼科醫師鄭惠川說，望遠時眼球肌放鬆，看近時緊繃，人們因近距離看書或看電視太久，睫狀肌會緊繃而造成假性近視。如何防治假性近視，醫生建議【在連續看螢幕 30 分鐘，應起身活動、望遠休息 5 分鐘並做伸展運動】。【註 2】

### 三、低頭族恐成忍者龜 5 招防脊椎側彎

低頭族要小心了！長時間低頭彎腰滑手機恐導致脊椎側彎變成「忍者龜」。近來骨科門診發現，後天姿勢不良造成脊椎側彎加重的患者激增 2 成，主要集中在青少年族群，推估和

長期不當使用 3C 產品有關。避免脊椎側彎的方法為注意姿勢、均衡飲食、少用電腦、多運動，建議每 40 分鐘就伸展筋骨，以避免肌肉僵硬。【註 3】

#### 四、只顧看手機！低頭族掉鐵軌險送命

手機讓人生活更便利，但是，也讓人降低對周圍危險的反應能力。一位低頭族忙著打手機，走著走著竟然跌落到火車月台下的鐵軌，整個過程被監視器拍了下來。

除此之外，低頭族忘情地上網、玩遊戲，玩出意外的情況越來越多。有人邊走邊滑手機，不小心滑下樓梯，如圖 1-3；有人在逛百貨公司時直接跌進噴水池，如圖 1-4；有人走在公園裡不看自然美景，只顧玩手機，下場就是一路滾下小山坡。【註 4】



圖 1-3 邊走邊【滑】手機【滑】下樓梯



圖 1-4 邊走邊滑手機 摔進噴水池

#### 五、睡前關燈滑手機！「黃斑部病變」增 3 成

醫師發現越來越多 30 到 40 歲患者求診，因為在就寢關燈後還使用智慧型手機，如圖 1-5，手機強光直射眼睛 30 分鐘以上，造成眼睛黃斑部病變，視力急速惡化，原本老年人才出現的症狀也出現在年輕人身上，黑暗中用手机還可能導致白內障，若不及時治療，恐怕有失明危險。英國針對 2000 名大學生的調查發現 75% 在手機陪伴下入眠。【註 5】



圖 1-5 黃斑部病變 嚴重可導致失明

## 貳、研究目的

分析許多的報章雜誌的報導與身邊親朋好友的經驗，我針對低頭族的健康與安全兩方面提出對策：

### 一、健康方面：

撰寫背景執行程式來計算使用者是否使用超過 30 分鐘，超過後，透過手機加速度感測器檢測使用者必須做的手臂運動，完成後才可繼續使用手機，藉此休息一下眼睛與僵化的手臂，除了做運動時無法使用手機之外，其餘時間本軟體均不會影響使用者使用手機。

我也會透過環境光感測器得知有人在環境光很暗的情況下使用手機，因此，使用手機 15 分鐘後就會啟動要求使用者做手臂運動的功能，讓使用者休息眼睛，除此之外，還可自動調整螢幕亮度，不讓螢幕光太強而造成眼睛更容易損傷。

### 二、安全方面：

我透過手機加速度感測器與陀螺儀監控，不讓使用者在走路時可以使用手機。

### 三、加速度感測器與陀螺儀的動作辨識：

因為我想要針對低頭族的健康與安全兩方面做出貢獻的關鍵是使用加速度感測器與陀螺儀，因此，必須先針對此兩種感測器做基本的了解，再整合出能辨別人的動作的程式。

## 參、研究設備與軟體與材料

### 一、研究設備與軟體如表 3-1

表 3-1 研究設備與軟體

硬體設備名稱	規格	軟體
個人電腦	記憶體 4GB	Eclipse(開發 java 手機程式)
智慧型手機	有加速度感測器、陀螺儀、環境光感測器	HTC driver

## 肆、研究過程及方法

### 一、研究過程

Step1：Java 程式設計訓練。

Step2：android 程式設計入門。

Step3：背景執行程式設計、感測器程式設計、廣播與接收程式設計。

Step4：程式整合設計。

Step5：程式整合後測試與除錯。

### 二、研究方法

本作品撰寫一個手機 app:透過背景程式([Service](#))與廣播接收訊息([BroadcastReceiver](#))來監控使用者使用手機的情況，看看使用者是否超時使用手機。

透過感測器 ([Sensor](#)) 信號偵測，第一：不讓使用者在走路時使用手機，讓自己更安全。第二：判斷使用者是否有完成手臂運動讓自己更健康。第三：環境光亮度偵測與螢幕亮度調整。安排的研究如下：

- (一) 加速度感測器信號偵測實驗與動作辨識研究：加速度感測器判斷走路與運動動作。
- (二) 陀螺儀感測器信號偵測實驗與整合動作辨識研究：陀螺儀搭配加速度感測器做動作辨識，以增加正確動作的成功率與減少誤動作被判成正確動作的機會。
- (三) 環境光亮度偵測與螢幕亮度調整研究：讓使用者在暗的環境下更能避免強光傷眼。
- (四) 背景執行程式研究：研究不干擾使用者使用手機情況下，又能計算手機使用時間。
- (五) 廣播接收訊息程式研究：研究接收 android 作業系統的廣播訊息接收。



圖 4-1 作品系統架構圖

### 三、加速度感測器信號偵測實驗與動作辨識研究：

#### (一) 文獻探討：

微機電加速度感測器早在 1970 年代中即已問世，最大的應用領域為汽車市場，主要用在安全氣囊；隨著成本降低與尺寸縮小，加速度感測器開始被導入消費性的市場。【註 6】

加速度感測器由兩大晶片組成：感測部分為物理裝置 (MEMS 傳感器)，信號處理部分為把感測到的類比信號轉換成數位訊號 (ASIC)，如圖 4-2。【註 7】

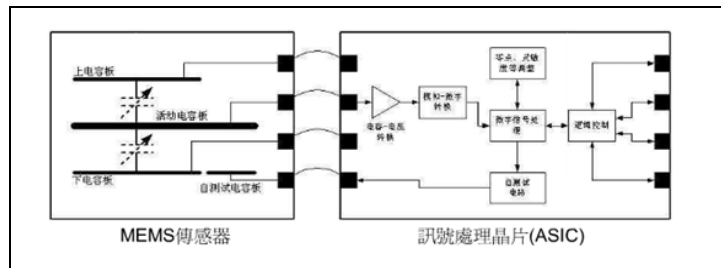


圖 4-2 MEMS 架構圖

加速度感測器常被用來測量距離、衝擊力…等等，裝置的架構設計成兩個電容及一個可導電、可晃動的懸臂系統，如圖 4-3，當感測器受力移動就會改變兩個電容值，電容值變化量轉成的數位訊號可讓我得知感測器的移動狀況，例如：物體往右移，左邊電容因為極板距離變小而增加，右邊電容值減少，產生了電容值差的變化量，如圖 4-4。【註 8】

受力公式： $F = kx$  ；

F：物體所受外力合、k：物體的彈性係數、x：位移量

電容值公式： $C = \epsilon \frac{A}{d}$

C：電容量、 $\epsilon$ ：介電常數、d：極板距離、A：極板面積

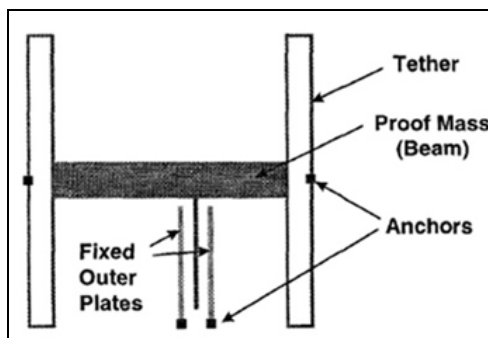


圖 4-3 加速度感測器結構

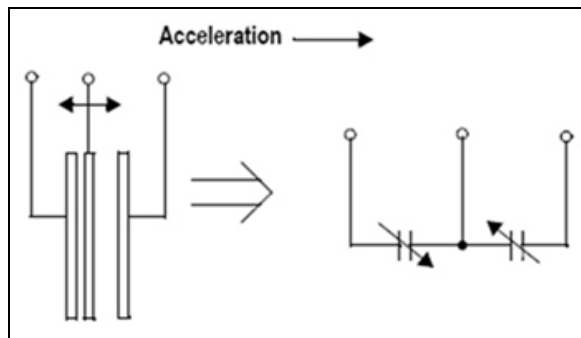


圖 4-4 右移發生的加速度感測器變化

加速度感測器可以感測的五種基本動作分別為加速度（位移）、震動、衝擊、傾斜及旋轉，如圖 4-5。【註 9】



圖 4-5 五種動作模式示意圖

## (二) 感測器測試參考程式：

Android 平台手機支援多種感測器，有加速度感測器（[accelerometer](#)）、陀螺儀（[gyroscope](#)）等等，參考程式如圖 4-6。實測加速度感測器 x、y、z 軸值如圖 4-7。【註 10】

第 1 行：實作 [SensorEventListener](#) 感測器監聽介面來取得感測器數值。

第 3 行：取得手機上的感測器管理元件，命名為 sm。

第 4 行：設定監聽的感測器為加速度感測器。

第 5~12 行：透過實作 [onSensorChanged](#) 方法來獲得最新感測器數值。

```
1 public class Main implements SensorEventListener
2 {
3     SensorManager sm = (SensorManager) getSystemService(SENSOR_SERVICE);
4     sm.registerListener(this,
5                         sm.getDefaultSensor(Sensor.TYPE_ACCELEROMETER),
6                         SensorManager.SENSOR_DELAY_FASTEST);
7     public void onSensorChanged(SensorEvent event) {
8         StringBuilder sb = new StringBuilder();
9         sb.append("values[0]:" + event.values[0] + "\n");
10        sb.append("values[1]:" + event.values[1] + "\n");
11        sb.append("values[2]:" + event.values[2] + "\n");
12        textView.setText(sb.toString());
13    }
```

圖 4-6 感測器參考程式

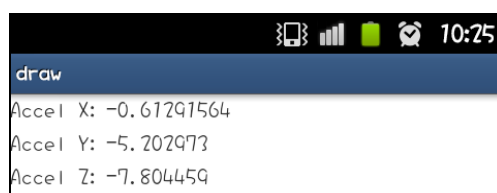


圖 4-7 加速度感測器實測值



### (三)「靜止」研究與實驗：可得知「傾斜」動作

手機的座標軸如圖 4-8，螢幕短邊為 X 軸、長邊為 Y 軸、垂直螢幕面的方向為 Z 軸，靜止時，先針對單軸做加速度感測器值的量測，再對兩軸座測量，如圖 4-9，我用三角尺規設計成 45 度傾斜的基座，把手機放上去可以測得 Y 軸與 Z 軸的加速度分量，最後，針對三軸做測量，我用兩組三角尺規設計成 30 度傾斜及 45 度傾斜的基座，手機放上去後就可以測得 X、Y 與 Z 三軸的加速度分量，因為是靜止的，所以分量的相量總合會等於重力加速度值(約 9.8)，因此，我可以使用下列公式來計算加速度感測器的訊號向量強度 (SVM)，其中，x 為 X 軸的分量、y 為 Y 軸的分量、z 為 Z 軸的分量。

$$SVM = \sqrt{x^2 + y^2 + z^2}$$



圖 4-8 手機座標

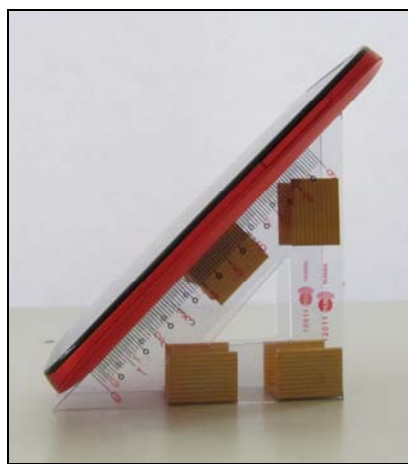


圖 4-9 兩軸測試圖

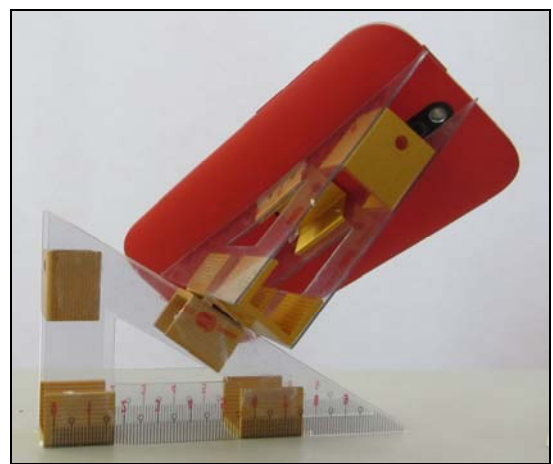


圖 4-10 三軸測試圖

經由上面的實驗器材與數學公式，我測得如表 4-1 的數據，SVM 值與重力加速度值的誤差值最大約±0.3，我也發現，Y 軸誤差值較大，我在想是否是單一手機的問題，再用另外 4 台手機測試發現，Y 軸誤差值還是最大，因為所測手機均一樣，因此，我懷疑是否是手機裡使用的感測器硬體規格並不是那麼好。

表 4-1 加速度感測器「靜止」動作的數據

					
	單 X 軸	單 Y 軸	單 Z 軸	Y、Z 軸	X、Y、Z 軸
X 軸分量	-9.806	0.681	0.027	0.145	-4.862
Y 軸分量	0.065	9.952	0.102	-4.440	-5.625
Z 軸分量	0.217	-0.491	-9.807	8.608	-5.965
SVM	9.809	9.971	9.807	9.679	9.533

#### (四)「走路」研究與實驗：可得知「加速度」動作

從走路的實驗數據轉成圖形，可讓我了解每次踏出步伐時，手機感測器除了感測到重力加速度之外，還有走路瞬間的加速度值（ $SVM \geq 11.5$ ），每一個步伐在要踏回地面時，也因為地面的阻力讓速度變慢（ $SVM \leq 8$ ），如果沒有在走路，就會回到約 9.8 的情況，如圖 4-11，我把邊走邊玩手機的 SVM 值記錄下來並轉成圖形顯示，看得出走了 12 步，圖 4-12 為走樓梯的圖，跟平地走路很類似，只不過，加速度值的變動更劇烈，不過，還是可以看得出下了 13 個樓梯階。

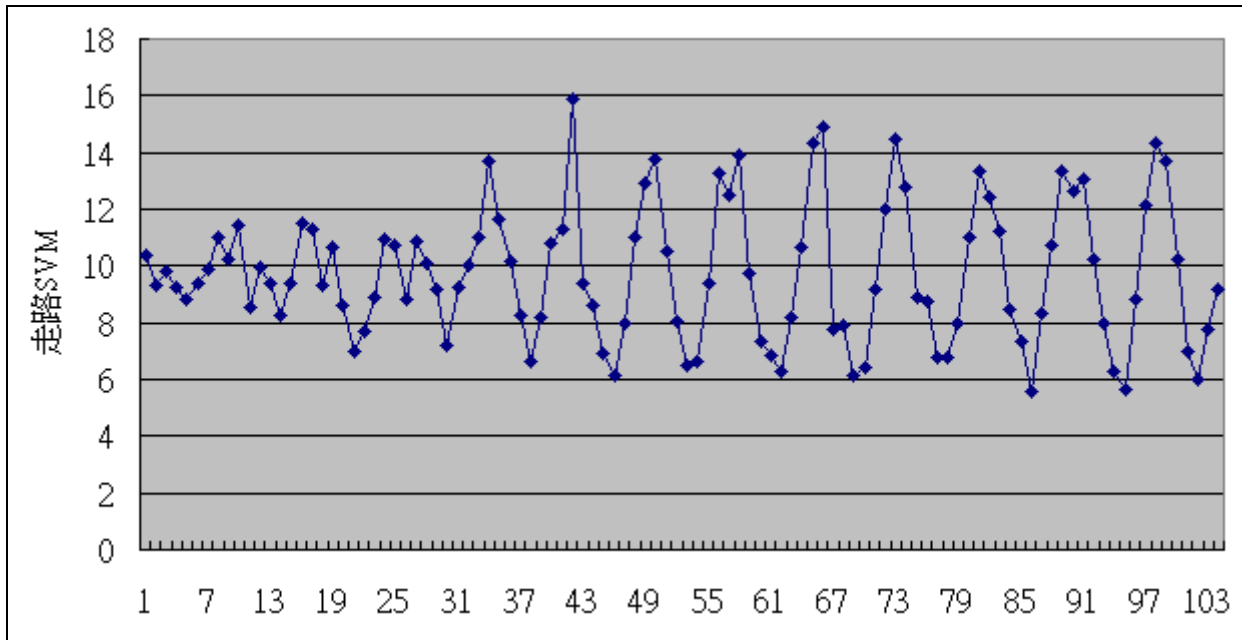


圖 4-11 走路的 SVM

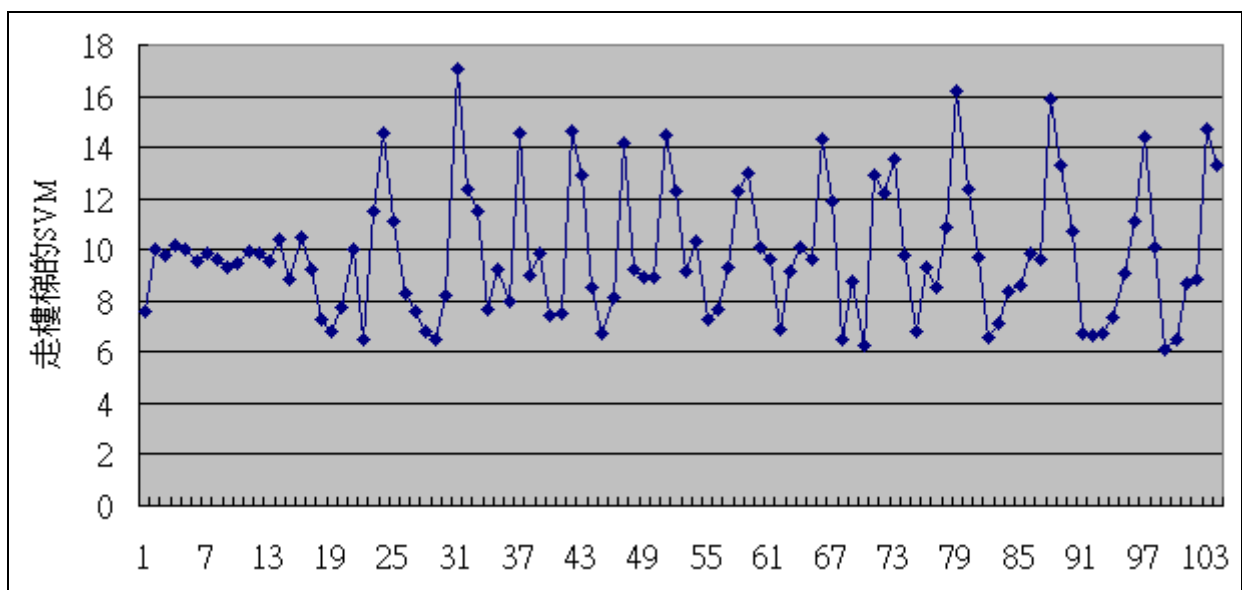


圖 4-12 走樓梯的 SVM

在我實驗後確定可以透過加速度感測器得知是否在走路後，遇到一個問題是：如果只是傳遞手機給別人呢？在公車上呢？我希望大家走路時不要玩手機，我會跳出提醒視窗，可是，如果連傳遞手機也會造成跳出視窗，這樣，手機使用者會覺得這個作品不好，一直在干擾我使用手機，於是，我針對傳遞與公車上的 SVM 做了實驗。

傳遞手機雖然 SVM 值也會增加到跟走路差不多，不過，較沒有規律且波動的頻率比較高，如圖 4-13。公車上的 SVM 值有點出乎我預料，SVM 還蠻平穩的，或許是因為駕駛技術好也安穩駕駛車輛，因此，光利用 SVM 的大小值就可以判斷出，在公車上而非走路，如圖 4-14。

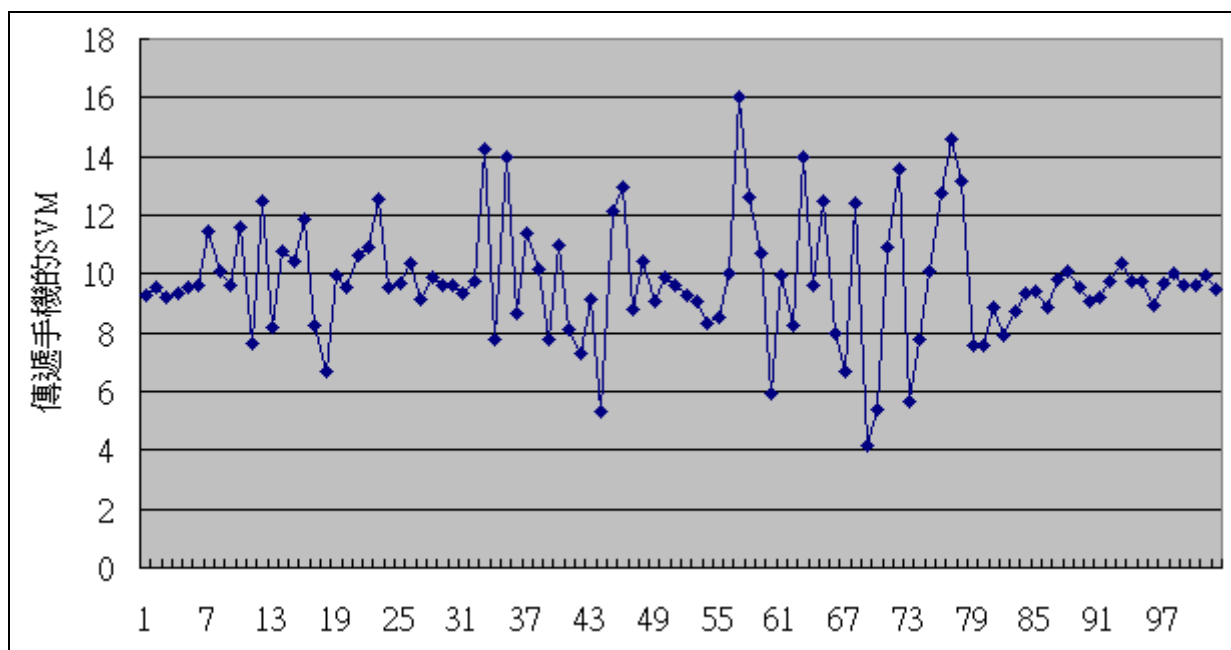


圖 4-13 傳遞手機的 SVM

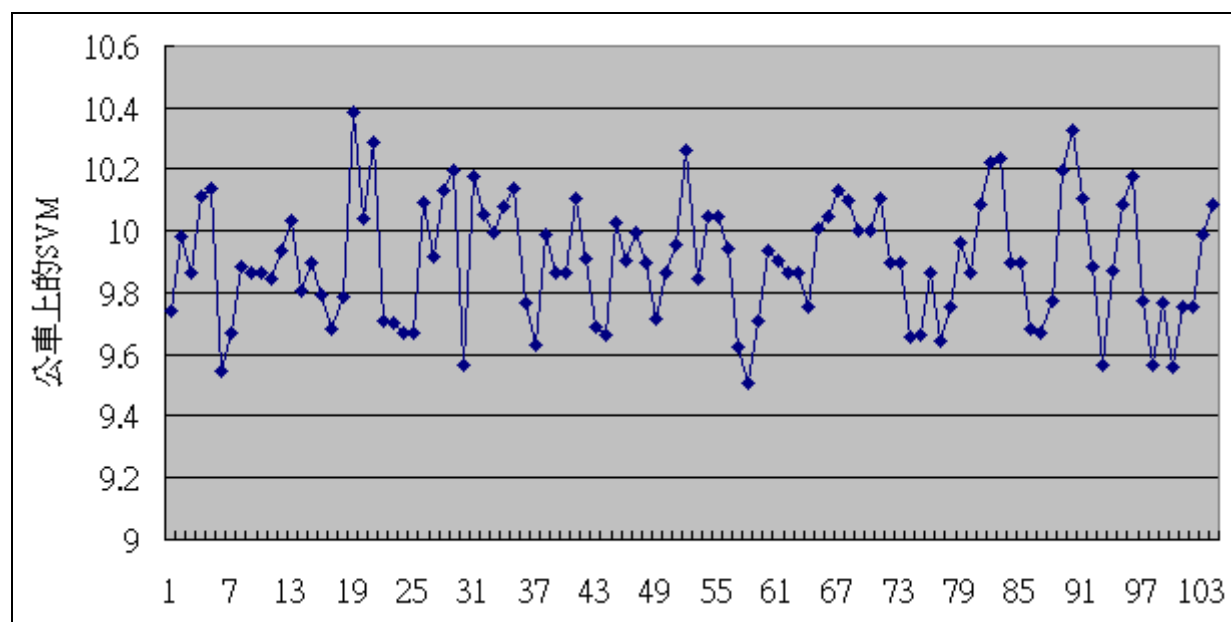


圖 4-14 公車上的 SVM

### (五)「手臂運動」研究：可得知「旋轉」動作

從多次的實驗數據可得知，無論使用者在拿手機時是拿長邊或短邊，都是 Z 軸會從負值轉成正值（手臂從放下到舉過肩），因此，我們只要透過 Z 軸的值即可辨識手機使用者是否有做手臂健康操，圖 4-15 可看出，我規律的做了 5 下健康操。

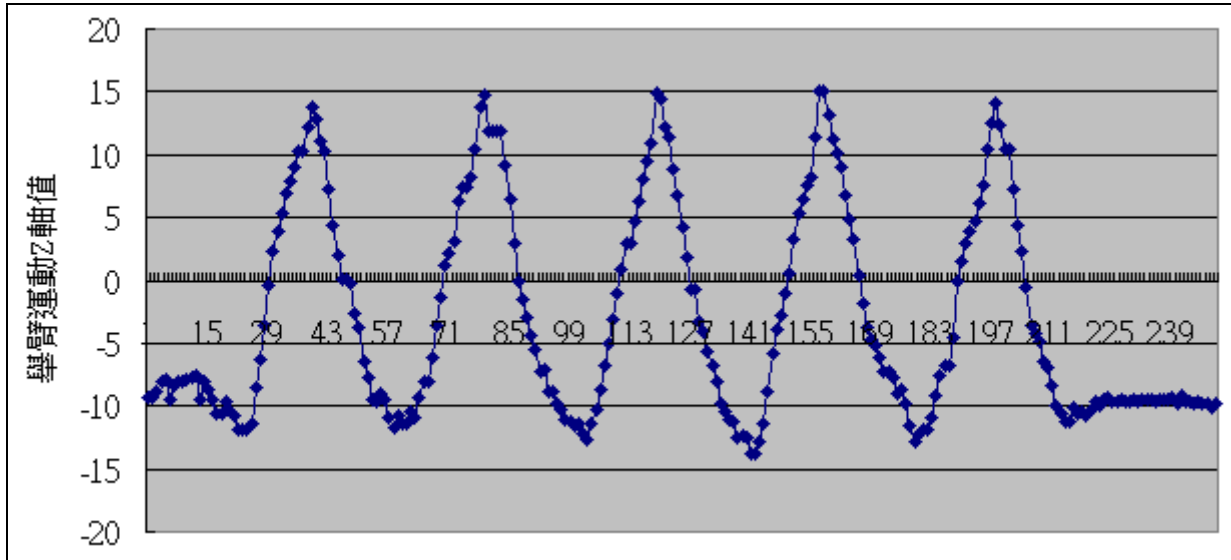


圖 4-15 舉臂運動的 Z 軸加速度值

### (六) 使用 SVM 與單軸值做動作辨識的結論

我針對本應用的兩個動作：走路與舉臂運動，做了正確率的測試，結果如表 4-2：

- 1、手臂運動：舉臂運動正確率高 99.68%，因此不做修正。
- 2、走路：每個人走路的速度與強度嚴重影響判斷結果，因此，我們期望找尋更好的動作辨識方法。

表 4-2 使用加速度感測器做動作辨識結果

	正確率	個人最大誤差率
走路	71/100 → 71%	0/16 → 100%
舉臂運動	314/315 → 99.68%	1/30 → 3.33%

## 四、陀螺儀感測器信號偵測實驗：

### (一) 陀螺儀文獻探討：

陀螺儀是利用高速轉動飛輪所具有的特性而設計出一種儀錶。1850 年法國的物理學家 J. Foucault 首先發現高速轉動中的轉子 (rotor) 由於慣性，它的旋轉軸永遠指向一固定方向，他用希臘字 gyro (旋轉) 和 skopein (看) 合為 gyroscopei 來命名這種儀錶。【註 11】

陀螺儀主要是一個對旋轉軸以極高角速度旋轉的轉子裝在一支架內，如圖 4-16-a，在中心軸  $XX_1$  上加一內環架，如圖 4-16-b，然後，在內環架外加上外環架，如圖 4-16-c，這個陀螺儀就可以環繞三軸作自由運動，如圖 4-16-d。

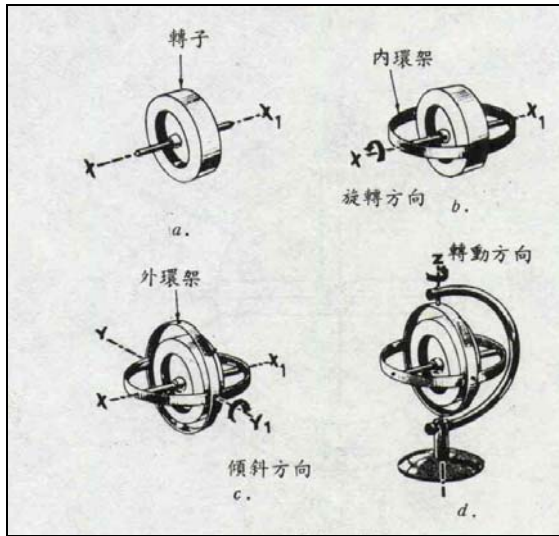


圖 4-16 陀螺儀架構

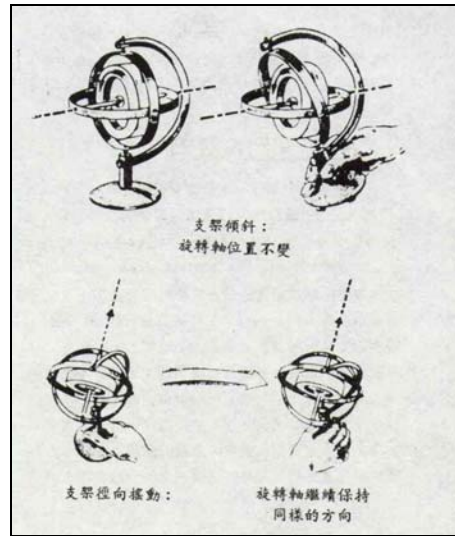


圖 4-17 定軸性

陀螺儀的兩個基本特性：定軸性 (inertia or rigidity) 與逆動性 (precession)。

### 1、定軸性

當陀螺轉子以極高速度旋轉就產生了慣性，使得陀螺轉子的旋轉軸保持在一個固定的方向，同時也反抗任何改變轉軸方向的力量，這種物理現象稱為陀螺儀的定軸性，如圖 4-17。其定軸性受轉子質量、旋轉半徑與旋轉速度影響。

### 2、逆動性

運轉中的陀螺儀，如果外界施力或力矩在轉子轉軸上，則轉軸並不沿施力方向運動，而是順著轉子旋轉向前 90 度垂直施力方向運動。逆動方向取決於施力方向及轉子旋轉方向，如圖 4-18。逆動性的大小受外界作用力、轉子的慣性力矩、轉子的角速度影響。

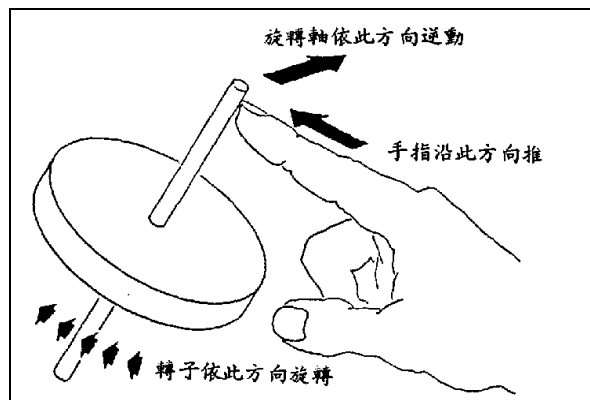


圖 4-18 逆動性

## (二) 陀螺儀測試參考程式：

手機獲取感測器數據的方法都一樣，透過實行 `SensorEventListener` 介面，產生管理元件後，透過 `onSensorChanged` 方法來得到數據，針對陀螺儀的數據說明如下：圖 4-19

`value[0]`：手機繞著 Z 軸的旋轉角度 (yaw)，表示方位。

0：北方、90：東方、180：南方、270：西方。

`value[1]`：手機繞著 X 軸的傾斜度 (pitch)。

`value[2]`：手機沿著 Y 軸的滾動角度 (roll)。【註 12】

```

1 public class Main implements SensorEventListener
2 {
3     public void onSensorChanged(SensorEvent event)
4     {
5         DangyroX.setText("X:" + values[0]);
6         DangyroY.setText("Y:" + values[1]);
7         DangyroZ.setText("Z:" + values[2]);
8     }
9 }

```

圖 4-19 陀螺儀參考程式

## (三) 陀螺儀三軸的基本實驗：

我們發現，手機若是指向北方，繞著 Z 軸的旋轉角度值約為 0，因此，這個參數可以告訴我們，我們目前往哪個方向走。

手機水平放在桌子上，往上往前繞著 X 軸產生的傾斜度為負，傾斜角度從  $0 \rightarrow -90 \rightarrow -180$ ，相反的，若往上往後繞著 X 軸產生的傾斜角度從  $0 \rightarrow 90 \rightarrow 180$ 。

手機水平放在桌子上，沿著 Y 軸向右滾動產生的角度為正，滾動角度從  $0 \rightarrow 90 \rightarrow 0$ ，相反的，沿著 Y 軸向左滾動產生的角度為負，滾動角度從  $0 \rightarrow -90 \rightarrow 0$ 。

表 4-3 陀螺儀基本實驗數據

陀螺儀特性 \ 手機單軸	 手機 Z 軸垂直桌面	 手機 Y 軸垂直桌面	 手機 X 軸垂直桌面
繞著 Z 軸的旋轉角度	依手機方向決定	依手機方向決定	依手機方向決定
繞著 X 軸的傾斜度	0.0	0.0	-90.0
沿著 Y 軸的滾動角度	0.0	-87.0	0.0

## 五、陀螺儀與加速度感測器整合做動作辨識研究：

### (一) 動作辨識演算法：

#### 1、Longest Common Subsequence (LCS)

在一堆序列當中，每個序列都有出現且最長的子序列，就是「最長共同子序列」：LCS。

例如：下面兩個序列 s1、s2 的 LCS 為 30

s1: 7 3 0 1 2 6

s2: 9 4 1 0 3 0

兩個序列分別為 s1、s2，要找出 s1 和 s2 的 LCS。首先，將 s1 的最後一個元素拆出來，這元素稱為 a1，剩餘部分稱為 sub1，s1=sub1+ a1；同樣的 s2=sub2+b1，s1 和 s2 的 LCS 會等於 sub1+a1 和 sub2+b1 的 LCS。現在來分析吧！所有情況分做四種：

#### (1) a1 是 LCS 的一部份，b1 不是。

對 a1 來說，在 sub2 一定要能找到一個元素和 a1 相同，才可能形成包含了 a1 的 LCS。b1 則是沒有用的元素，要找 LCS 只需要去找 sub2 就可。也就是說 s1 和 s2 的 LCS，其實就是 s1 和 sub2 的 LCS。

#### (2) b1 是 LCS 的一部份，a1 不是。

此情況和第一種類似，s1 和 s2 的 LCS 就是 sub1 和 s2 的 LCS。

#### (3) a1 和 b1 都不是 LCS 的一部份。

直接找 sub1 和 sub2 的 LCS 就好啦！

#### (4) a1 和 b1 都是 LCS 的一部份。

a1 和 b1 都是 LCS 的一部份，且 a1 和 b1 都在最後，a1 和 b1 一定會相等！a1 亦會是 LCS 的最後一個元素。LCS 剩下的部份，只需從 sub1 和 sub2 著手即可，因此，s1 和 s2 的 LCS，必定是 sub1 和 sub2 的 LCS 在尾端加上 a1（或 b1）。

眾合以上分析的結論如下。【註 13】

$$LCS(s1, s2) = \begin{cases} \max(LCS(sub1, s2), LCS(s1, sub2)), & \text{when } a1 \neq b1 \\ LCS(sub1, sub2) + a1 & , \text{when } a1 == b1 \end{cases}$$

#### 2、空間向量特徵值

我們必須先把加速度感測器感測到的 3 個值分成 8 個空間，如圖 4-20，例如：X=9.8、Y=0.2、Z=1.6，則此空間向量特徵值為 1，每個動作在一段時間內，都會有擁有一連串的加

速度值序列，我們可以把這個序列轉換成空間向量特徵值，例如：往前走路的特徵向量值序列可為 1222665551，我們後續即可透過立即得到的序列值與正確序列值比對來判定是否有做出某個動作。【註 14】

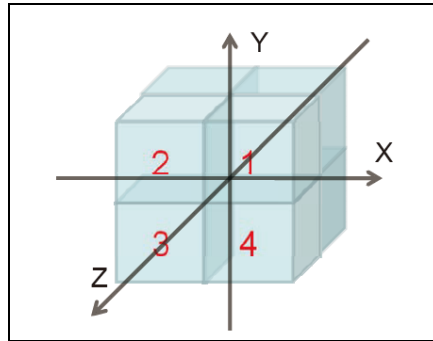


圖 4-20 空間向量特徵值

(二) 單獨使用加速度感測器做動作辨識的盲點實驗：

我做的第一個實驗，如果手機水平握著如圖 4-21，把加速度感測器所測到的重力加速度值都加在 Z 軸，做水平的移動動作，希望看看各種動作的加速度值為多少 (X 軸)，我測試結果如表 4-4，如果是邊走邊玩手机，會比正常走路還慢些，我發現，加速度值瞬間最大值為 2。

表 4-4 移動變化量



	X 軸加速度最大變化值
正常走路	2
快速走路	2.7
慢跑	3
快跑	6 以上

圖 4-21 水平握著走路

圖 4-22 傾斜握著走路

我做的第二個實驗，把手機握著如邊走邊玩的角度 (Y 軸約呈 45 度左右)，如圖 4-22，此時因為加速度感測器測到的重力加速度值已經分成兩分量： $g_x = -9.8 \times \sin(45^\circ) = -4.9$ 、 $g_z = -9.8 \times \cos(45^\circ) = -4.9$ ，因此，走路的加速度值最大只有 2 的變化量，也就是說，X 軸量到的值最大為  $-4.9 + 2 = -2.9$ ，最小值為  $-4.9 - 2 = -6.9$ ，如果使用演算法去分析時，無論走路或靜止，空間向量特徵值都是 7，因此會造成無法辨識，所以，我希望透過陀螺儀找到的手機傾斜角度來解決此問題。



我把陀螺儀測到的傾斜角拿來換算重力加速度在 3 軸的分量，因此，就可以把重力加速度的影響排除，表 4-5 就是測試數據，開始走路前，原本加速度感測器測到的  $g_x = -7.035$ ，考慮陀螺儀測得的角度  $g_x = 0.419$ ，走路時，只有加速度感測器測到的值從  $-7 \rightarrow -8 \rightarrow -5$ ，空間向量特徵值永遠都一樣，也跟靜止時一樣，因此，無法判斷使否有走路；考慮陀螺儀測得的角度  $g_x$ ，一開始會因為慣性而產生負值，之後會產生正值（因為往前走路），真的可以偵測得到走路本身的特性。

表 4-5 有陀螺儀的優點

X 軸走路加速度值(無陀螺儀)	-7.035	-6.778	-7.575	-7.715	-8.134	-7.372	-7.780	-5.150	-5.336	-6.156
空間向量特徵值	7	7	7	7	7	7	7	7	7	7
X 軸走路加速度值(陀螺儀加入)	0.419	-0.399	-0.538	-0.957	-0.195	-0.603	1.642	1.456	0.636	0.230
空間向量特徵值	1	2	2	2	6	6	5	5	5	1

### (三) 整合陀螺儀與加速度感測器做動作辨識研究：

#### 1、走路：

**比對率 (matching rate)**：為兩個動作的相似程度，例如：2522152625 代表往前走路的空間向量特徵序列，實際往前走路的某一個序列 2212156665，兩個序列經過 LCS 計算為後的比對率為  $\frac{7}{10} \times 100\% = 70\%$ ，實際從包包拿出手機的某一個序列 1155662112，與 2522152625 經過

LCS 計算為後的辨識率為  $\frac{4}{10} \times 100\% = 40\%$ ，由此可知，比對率越高，相似程度越高。

**辨識率 (recall rate)**：為任何一個動作產生的空間向量特徵序列能否被判定成某一個動作，藉由設定一比對率門鑑值，當超過此門鑑值就算辨識成功，門鑑值越低，辨識率會越高。例如：若設定比對率的門鑑值為 40%，延續上面的例子，往前走的動作（70%）與從包包拿出手機的動作（40%）都會當成辨識成功。

**精準率 (precision rate)**：為任何一個動作產生的空間向量特徵序列被辨識成某一個動作是否正確。若我們把是否成功辨識的比對率門鑑值定的越低，雖然辨識率會增加，但是，精準度就會越差。

例如：延續上例，因為辨識門鑑值低造成誤把手機從包包拿出來的動作辨識成走路。

走路運動單單使用加速度感測器的 SVM 值來辨識，是有困難的，第一：大家走路的速度與強度都不一樣，第二：很多其它的動作也會影響辨識，造成不是做走路動作也辨識成走路，因此，我們使用另一種方法：LCS，透過多人的走路測試，找到最佳辨識走路的空間向量特徵

序列，當辨識成功的比對率門鑑值定在 50% 以上時，可以成功達到  $\frac{5}{6} \times 100\% = 83.3\%$  的辨識

率，精準率為  $\frac{11}{12} \times 100\% = 91.6\%$ ，期望能達成辨識率與精準率都能 95% 以上。

表 4-6 LCS 辨識數據—以 2656556422 為往前走比對序列

測試者	往前走 空間向量序列	LCS	比對率	測試者	從包包拿手機 空間向量序列	LCS	比對率
正皓	6125656525	6	60%	正皓	2266251111	4	40%
鄧展	6267774823	4	40%	鄧展	6666443344	4	40%
翊宸	2256555212	7	70%	翊宸	6735738438	3	30%
宥勳	2556561611	5	50%	宥勳	1125638434	4	40%
哲旭	5225151525	5	50%	哲旭	2216221252	4	40%
靖靖	2115211556	5	50%	靖靖	2112216112	3	30%

## 2、手臂運動：

手臂運動使用加速度感測器，期間透過幾十人的測試→問題產生→數據蒐集與分析→反覆驗證與測試，我把辨識從一個臨界點 (0) 改成兩個臨界點 (-1, -10)，如圖 4-23，也就是把史密特電滯的觀念引入 (大於 -1、小於 -10 才轉態)，辨識時就不會因為任何的雜訊而誤判，例如：若只有一個臨界點 0，第 2 步因為抖動會判斷成走兩步，因此，辨識成功率從 95 進步到 99.68%，單人最大誤差也從 40% 降低到 3.3%。

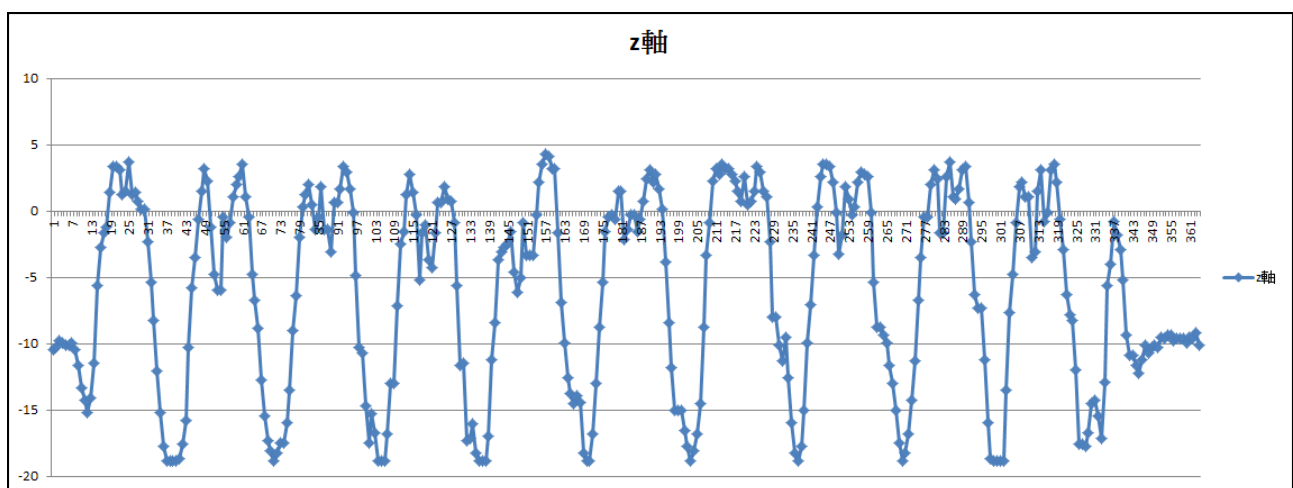


圖 4-23 走 10 步路時的加速度感測器 Z 軸的變化

## 六、環境光亮度偵測與螢幕亮度調整研究

### (一) 文獻探討：

Android 平台提供的光感測器用來偵測光的照度 (lux)，因此，可以拿來做照度計、調整手機螢幕亮度、支援照相時的閃光功能。光的照明基本知識如表 4-7，圖 4-24 以圖示來表達這四個名詞的觀念【註 15】。

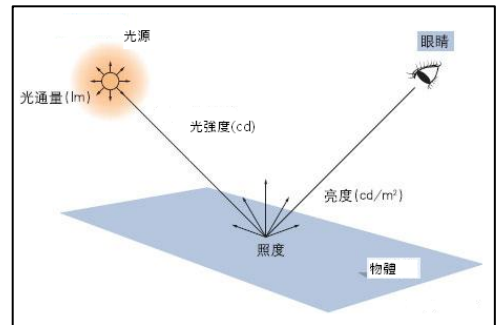


圖 4-24 光的概念

表 4-7 光的照明基本知識

名稱	說明	單位
光通量	光源每秒發出之總光量，簡單說就是發光量。	lm
照度	單位面積的入射光量，即環境的明亮度。	lux ( $\text{lm}/\text{m}^2$ )
光強度	在某一特定方向所放射的光量	cd
亮度	某一特定方向所看到物體反射光強度	$\text{cd}/\text{m}^2$
光效率	光通量除以消耗電量所得之值	lm/W

### (二) 環境光亮度測試參考程式：

基本上，android 手機要得之任何感測器值，做法都一樣，要先有感測器管理元件 (mr)，在來就是要取得手機內的哪些感測器硬體 (sr)，最後，就可透過 onSensorChanged 的方法來獲得感測值，我們這邊透過 value[0] 來取得環境光照度值，參考程式碼如圖 4-25。

```

1 public class Main implements SensorEventListener
2 {
3     mr = (SensorManager) this.getSystemService(SENSOR_SERVICE);
4     sr = this.mr.getDefaultSensor(Sensor.TYPE_LIGHT); //取得光感測器
5     public void onSensorChanged(SensorEvent event){
6         String msg = "現在的亮度是 " + arg0.values[0] + " Lux"; //arg.values[0]取得光感值
7     }
8 }

```

圖 4-25 環境光感測器參考程式

### (三) 螢幕亮度調整參考程式：

開始前請先加上 android.permission.WRITE\_SETTINGS 權限，我們這邊使用 brightness 來做亮度值的處理，最低為 10，避免 android 系統將整個螢幕關閉，我們透過第 9 行來重新

設定螢幕的亮度，參考程式碼如圖 4-26。【註 16】

```
1 public class Main extends Activity
2 {
3     public int brightness = 10; //設定螢幕基準亮度為10
4     //亮度最低為10，避免android系統將整個螢幕關閉
5     brightness = Math.min(brightness, 255);
6     brightness = Math.max(brightness, 10);
7     WindowManager.LayoutParams lp = getWindow().getAttributes();
8     lp.screenBrightness = brightness/255f;
9     getWindow().setAttributes(lp); // 設定螢幕亮度
10 }
```

圖 4-26 調整螢幕亮度參考程式

## 七、背景執行 (Service) 程式研究：

### (一) 文獻探討：

Service 是一個背景程序，透過 Service，我們可以做一些不需要使用者介面 (UI) 的功能，例如：在背景撥放音樂。Service 並非一個獨立的 Process，不會脫離 Main Thread 獨立運作，如果需要做長時間的運算，可以讓 thread 幫你作一些工作。【註 17】

### (二) 背景執行測試參考程式：

Service 是執行在背景的服務程式，沒有提供畫面與使用者互動。Service 服務的過程有三個階段，(1) 建立服務。(2) 開始服務。(3) 停止服務。在應用程式中要能使用此服務，需要在 AndroidManifest.xml 檔案中加入 <Service> 標籤。圖 4-27 為參考程式碼。【註 10】

```
1 public class EchoService extends Service {
2     public void onCreate() { //建立服務
3         super.onCreate();
4     }
5     public void onStart(Intent intent, int startId) { //開始服務
6         super.onStart(intent, startId);
7     }
8     public void onDestroy() { //停止服務
9         super.onDestroy();
10    }
11 }
```

圖 4-27 Service 參考程式碼

當服務寫成一個類別後，若要在主類別中開啟與關閉服務，參考程式碼如圖 4-28：

第 1 行：使用 Intent 要求從 Main 類別去啟動 Service.class。

第 2 行：啟動 Service。

第 3 行：關閉 Service。

```
1 Intent intent = new Intent(Main.this, Service.class);
2 Main.this.startService(intent);
3 Main.this.stopService(intent);
```

圖 4-28 啟動與關閉服務參考程式碼

## 八、廣播接收訊息程式研究：

### (一) 文獻探討：

廣播接收者 (BroadcastReceiver) 用於接收與廣播 Intent，通常一個廣播可以被多個廣播接收者所接收，例如：電池電量變化、系統時間都可以透過廣播接收得知。【註 12】廣播有兩種類型：普通廣播和有序廣播。普通廣播可以在同一時刻被所有接收者接收到，消息傳遞效率較高，缺點是不能將處理結果傳遞給下一個接收者；有序廣播是按照接收者聲明的優先級別依次接收廣播。例如：級別高低為 A>B，廣播會先傳給 A，再傳給 B。

廣播接收者的生命周期是很短暫的，在收到廣播時創建，onReceive() 方法結束之後銷毀，廣播接收中不可做一些耗時的工作，否則會彈出 Application No Response 錯誤對話視窗，耗時的工作放在服務 (Service) 中完成較好。

### (二) 廣播接收訊息測試參考程式：

系統的資訊可以透過 BroadcastReceiver 來廣播給有註冊欲處理這個訊息的接收程式，例如：電池的電力、系統時間…等等。圖 4-29 為參考程式碼。【註 18】

第 1 行：撰寫一個繼承 android.content.BroadcastReceiver 的類別。

第 2 行：當有廣播服務時，會由 onReceiver 方法接收廣播訊息。

第 3 行：根據接收的廣播訊息，決定相對應的執程式，例如：intent.getAction 中會傳來廣播訊息，如果你要知道電量變化的話，就要判定此廣播訊息是否為 Intent.ACTION\_BATTERY\_CHANGED。

```
1 public class StartupReceiver extends BroadcastReceiver{
2     public void onReceive(Context context, Intent intent){
3
4     }
5 }
```

圖 4-29 BroadcastReceiver 參考程式碼

## 伍、研究結果

經過感測器的實驗後，我確認可以使用加速度感測器與陀螺儀做走路與運動的辨別，因此，就把它拿來做為保護低頭族健康與安全的功能，程式架構如圖 5-1。

程式一開始會：

- (1) 啟動背景執行：因為沒有使用者介面，所以，使用者還是可以正常使用手機。
- (2) 啟動廣播接收：讓我知道目前手機螢幕是否有關閉，用來判斷是否有使用手機。
- (3) 計算時間：如果螢幕是亮著的，就當成在使用並且計算使用時間。

之後程式保持對下面兩件事做監測：

- (1) 加速度值的監測：若 SVM 大於 10.5 後，會判別是否在走路，如果是，就跳出提醒視窗告知，走路時不要使用手機，提醒視窗按掉後還是可正常使用手機。
- (2) 是否關閉螢幕：只要螢幕關閉超過 3 分鐘，計時就歸零，若螢幕亮著超過 30 分鐘，就必須做手臂健康操，完成後還是可以繼續使用手機。

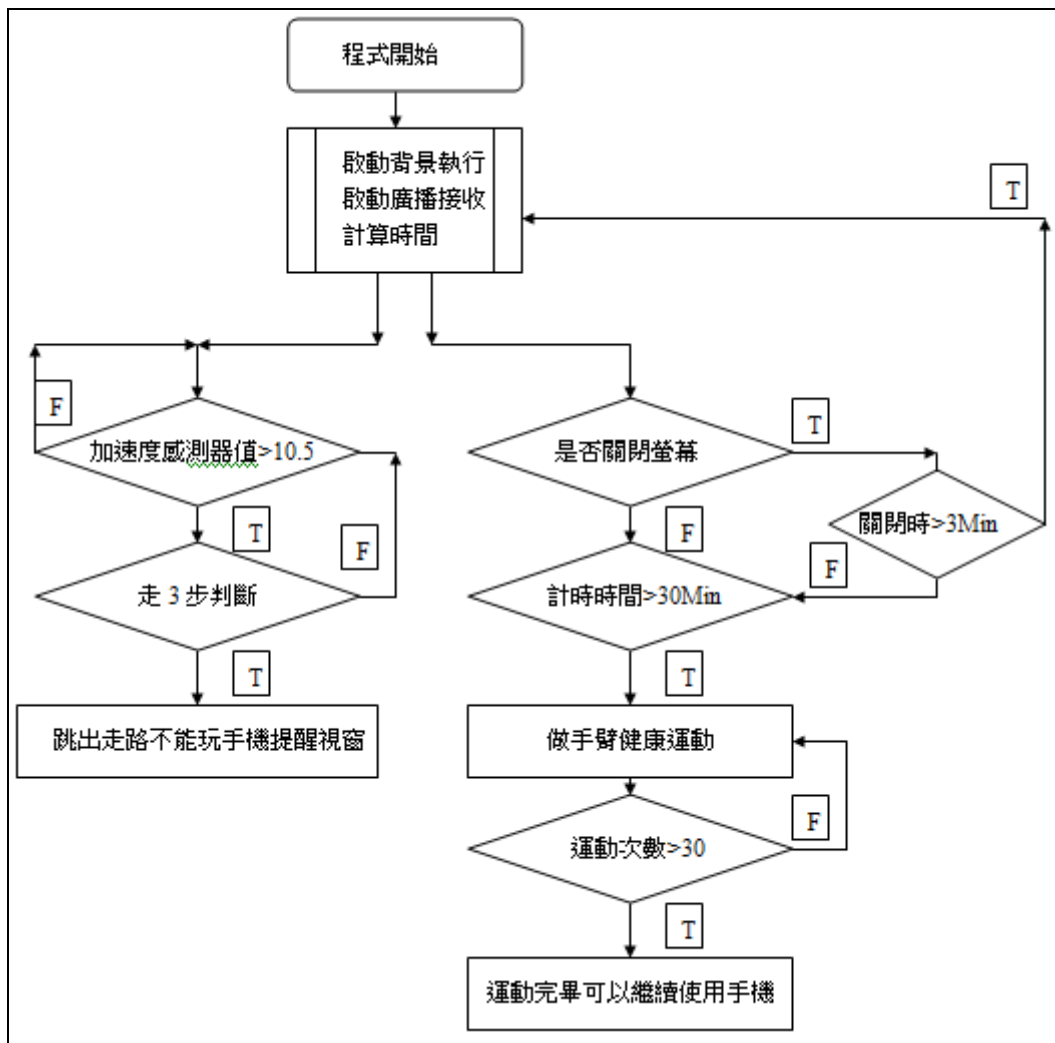


圖 5-1 低頭族應用的程式架構圖

## 陸、討論

一、別的遊戲軟體在執行，我的程式如何同時執行？

解決辦法：使用背景服務程式即可。

二、不知道要用哪種感測器？

解決辦法：每種感測器實際測試，發現加速度感測器最符合需求。

三、要作檔案記錄感測器值時發現書上的內容過舊？

解決辦法：eclipse 有相關除錯提示，還蠻實用的，也解決問題了。

四、程式一測試就立即終止，不好除錯？

解決辦法：可以從 DDMS 與 LogCat 看訊息或是把程式先縮小測試。

五、不知道如何才知道人在走路？

解決辦法：找了許多資料，也測了許多方法，終於找到平方開根號的方法最好。

六、Java 物件導向很難懂？

解決辦法：多看不同的書的解說，多寫練習題。

七、運動要偵測方向使用陀螺儀還是加速度感測器？

解決辦法：兩個都可用。

八、動作辨識的正確率低？

解決辦法：除了加速度感測器外，加速陀螺儀測得之角度值共同辨識。

九、陀螺儀測得的角度並非加速度感測器的 3 個軸的角度？

解決辦法：找出數學關係式來做轉換。

十、沒有執行表單畫面的程式，無法調整亮度值？

解決辦法：使用跳出小表單的方式就可以調整。

十一、如何增加正確率？

解決辦法：我們會找很多人來做測試，測試過程與感測器值使用檔案記錄下來，以方便我們事後分析失敗原因，來增加做品的正確率與穩定性。

十二、使用 LCS 的動作辨識率，看起來相似成度都不高？

解決辦法：找出可真正代表大多數人的序列，很重要。

## 柒、結論與應用

### 一、作品結論說明：

本作品可以透過**手機感測器辨識走路**（精準率：91.6%）跟**手臂運動**（精準率：99.68%），透過這兩個辨識讓低頭族在使用手機時能更安全與更健康，這兩個目的的實現作法如下：

#### （一）健康方面：環境光線充足

本作品一執行後的畫面如圖 7-1，「happy 低頭族」顯示幾秒後消失，使用者就可以正常使用手機，完全不受本軟體在背景執行所影響，當使用者連續使用時間超過一定的時間，例如：20 分鐘，本軟體會跳出提醒訊息「已使用 19 分鐘，將於 6 秒後健康運動 :)」，如圖 7-2，這時，使用者可選擇在 3 分鐘內作「手機健康操」幾次，如圖 7-3，選完次數後就會跳出一個計量 bar，如圖 7-4，使用者開始做健康操，**健康操的動作 1 為拿著手機舉起手臂過肩膀**，如圖 7-5，**動作 2 為讓手臂垂直放下**，如圖 7-6，讓自己的眼睛休息一下，讓自己的肩膀舒壓一下，感測器偵測到做的次數會在計量 bar 上用黃色條顯示，如圖 7-7，最後，如果健康操做完，代表有休息過，使用者即可再使用手機。



圖 7-1 本軟體執行初始畫面



圖 7-2 使用超過時間後的顯示訊息



圖 7-3 健康操次數選擇畫面



圖 7-4 健康操計量 bar





圖 7-5 健康操動作 1



圖 7-6 健康操動作 2



圖 7-7 使用者正在做健康操中



圖 7-8 使用者做完健康操

### (二) 健康方面：環境光線很暗

我們使用環境光感測器隨時偵測使用者目前所在環境的光線強度，當我們的 app 發現手機使用者在很暗的環境下使用手機時（小於 50 lux），通常手機的亮度都太亮，如圖 7-9，這樣的情況下使用手機更傷眼睛，因此，我們會自動把手機亮度調整成更適合的亮度而較不傷眼，如圖 7-10，更把使用 30 分鐘後要做手臂運動的時間縮短成 15 分鐘，讓使用者少用多休息，可為兼顧健康與使用者使用手機的權益。



圖 7-9 傷眼睛手機亮度



圖 7-10 調整後較不傷眼手機亮度

### (三) 安全方面：

當手機的加速度感測器三個值的平方加起來再開根號後，如果約在 9.8，代表使用者沒有在走路，若 3 秒內經由 LCS 的判斷有走 3 步路的話，這時，我們就會判定使用者在邊

走邊玩，就會跳出如圖 7-11 的視窗畫面，手機就會呈現暫停的狀況，當按下同意鈕後，會在跳出一個提醒視窗「走路時不要玩手機」，如圖 7-12，使用者不能邊走邊使用手機時，或許就會抬頭看看環境而不會發生意外。



圖 7-11 感測器偵測到走路時的畫面 1



圖 7-12 感測器偵測到走路時的畫面 2

## 二、作品創新所在：

目前雖然有台灣的法律規定，開車者不可使用手機，但是，全世界沒有針對低頭族『使用過久』或『邊走邊滑』的問題提出任何解決辦法，我的作品可謂是一大創舉。

### (1) 使用過久方面：

使用過久造成的眼睛疲勞與肩頸痠痛，我不是把手機關掉，不讓使用者滑手機，我會在時間快到前提示使用者要先把遊戲或網路先暫時告一段落，再作「手機健康操」，讓眼睛與身體休息一下後可再玩，兼顧手機使用者的健康與玩樂手機的權益。在暗的環境下使用手機，我更加了調整螢幕亮度功能，讓眼睛較不易受傷害。

### (2) 邊走邊滑方面：

當手機使用者走路時，我會跳出訊息視窗干擾使用者使用手機，讓他覺得邊走邊玩不行後，他就不會邊走邊玩，本身安全就更多分保障。

## 三、作品應用：

目前越來越多兒童都已經隨身攜帶手機，電視劇兩個爸爸裡的溫蒂就是，常常兒童無法自我克制，造成長時間使用手機玩遊戲，身體變得較不健康，父母也不希望這樣，可是，手機卻又可以隨時都聯絡到自己的孩子，造成利弊均有。如果，有一天有手機業者可以把本作品融入系統中，讓爸爸媽媽可以設定此功能，當兒童玩遊戲超過正常時間就休息眼睛，運運動，更不讓兒童邊走邊玩，相信，一定會讓小朋友更能兼顧玩與安全與健康與家長的需求。

## 捌、參考資料及其他

註 1：六成五家長「低頭族」親子溝通少。日期：2013.01.18.。

取自網址：[http://news.rti.org.tw/index\\_newsContent.aspx?nid=401499&id=8&id2=1](http://news.rti.org.tw/index_newsContent.aspx?nid=401499&id=8&id2=1)

註 2：低頭族，成人假性近視多 10 倍。日期：2012.10.23.。取自網址：

<http://www.appledaily.com.tw/appledaily/article/headline/20121023/34592668>

註 3：低頭族恐成忍者龜 5 招防脊椎側彎。日期：2013.04.17.。取自網址：

<http://tw.news.yahoo.com/%E4%BD%8E%E9%A0%AD%E6%97%8F%E6%81%90%E6%88%90%E5%BF%8D%E8%80%85%E9%BE%9C-5%E6%8B%9B%E9%98%B2%E8%84%8A%E6%A4%8E%E5%81%B4%E5%BD%8E-054212298.html>

註 4：只顧看手機！低頭族掉鐵軌險送命。日期：2012.12.24.。取自網址：

<http://news.cts.com.tw/cts/international/201207/201207311061065.html>

註 5：睡前關燈滑手機！「黃斑部病變」增 3 成。日期：2013.06.05.。

取自網址：[http://www.dreyes.com.tw/med\\_det.php?ArticleId=641](http://www.dreyes.com.tw/med_det.php?ArticleId=641)

註 6：MEMS 運動感測器技術現況與系統設計要領。檢索日期：2013.01.26.。

取自網址：<http://www.ctimes.com.tw/Art/Show2.asp?0=200812031740057863>

註 7：Baidu 文庫（MEMS 加速度计与陀螺仪的原理介绍）。日期：2013.01.26.。

取自網址：<http://wenku.baidu.com/view/9ee56aee856a561252d36fcf.html>

註 8：結合 GPS/重力感測器/陀螺儀 DR 功能進駐導航裝置。日期：2013.04.17.。

取自網址：[http://www.mem.com.tw/article\\_content.asp?sn=0801290009](http://www.mem.com.tw/article_content.asp?sn=0801290009)

註 9：加速度計搭配陀螺儀 MEMS 應用如虎添翼。檢索日期：2013.01.13.。

取自網址：[http://www.2cm.com.tw/zoomin\\_content.asp?sn=0910050011](http://www.2cm.com.tw/zoomin_content.asp?sn=0910050011)

註 10：李昇暉、詹智安（2012）。Android 雲端實務程式設計。台北市：碁峯。

註 11：飛機的眼睛——淺談陀螺儀。檢索日期：2013.07.07.。

取自網址：<http://210.60.224.4/ct/content/1986/00090201/0012.htm>

註 12：李寧（2011）。Android 案例開發完全講義。台北市：碁峯。

註 13：演算法筆記。檢索日期：2013.09.09.。取自網址：

<http://www.csie.ntnu.edu.tw/~u91029/LongestCommonSubsequence.html>

註 14：鍾哲民。加速度動作辨識系統之研究與應用 P37。檢索日期：2013.07.07.。取自網址：

[http://etdncku.lib.ncku.edu.tw/ETD-db/ETD-search-c/view\\_etd?URN=etd-0901108-164902](http://etdncku.lib.ncku.edu.tw/ETD-db/ETD-search-c/view_etd?URN=etd-0901108-164902)

註 15：LED 燈光之家。檢索日期：2013.09.16.。

取自網址：<http://www.ledgood.com/knowledge-of-light/>

註 16：在 Android 調整螢幕亮度 (Brightness)。檢索日期：2013.06.08.。

取自網址：<http://cw1057.blogspot.tw/2011/11/android-brightness.html>

註 17：Android Developer 安卓官方開發網站。檢索日期：2013.05.10.。取自網址：

<http://developer.android.com/reference/android/app/Service.html>

註 18：StackDOC 網站。檢索日期：2012.10.30.。取自網址：

<http://rritw.com/a/bianchengyuyan/xml/2012/0628/177130.html>