

國立台北大學資訊工程學系專題報告

擴增實境戶外廣告看板之穿戴式裝置互動

專題組員：鄭力瑋 410285002 姚定緯 410285057

專題編號：PRJ-NTPUCSIE-104-009

執行期間：104/7/1 ~ 105/6/4

1.摘要

隨著科技的興起，穿戴式裝置已經普遍於市場，但以往的穿戴式裝置皆以取得穿戴者身體資料為主要功能，而不是讓穿戴者與裝置實體互動。然而在現代大多數的智慧裝置多以「觸控」面板為互動介面，若能將裝置的「觸控」面板互動方式更深入進化到「影像辨識」與「手勢辨識」來操作，那將會是讓穿戴式裝置更具人性化的一大進步。

目前智慧眼鏡的操作方式依然是以觸控面板為主，但若是使用觸控裝置去控制智慧眼鏡，我們將會為了控制畫面資訊而需佔用一隻手來操控，這樣反而沒有達成智慧眼鏡的原始設計目的——為我們騰出雙手來應付周邊事物，我們的目標是結合智慧眼鏡與智慧手環，讓使用者能透過智慧手環以手勢來控制擴增資訊。

2.簡介

實驗室原本已經有能夠顯示景點資訊的擴增實境瀏覽器，本專題是要在原本系統擴增影片播放功能，也就是讓智慧眼鏡上的數位相機取得周遭影像，進行影像辨識並追蹤戶外的靜態廣告看板，智慧眼鏡在辨識出廣告看板後，從雲端下載使用者所感興趣的多媒體廣告影片至眼鏡，然後以擴

增技術浮現在該廣告看板上。本專題要以串流方式來即時播放影片，因此在APP中加入了影片串流的功能，並架設Darwin Streaming Server(DSS)來實現即時串流的功能。然而下載的影片可能不只一部，如何快速下載多部影片讓使用者選取所欲觀看的影片是本專題的一個重點。

此外，如何讓使用者可以操控想要觀看的影片，即為本專題的另一個研究重點。經過探討，本專題要讓使用者可以用智慧手環操作影片的播放，現代的手環皆具備各式感測器，包括陀螺儀、加速規、六軸感測器，可以偵測使用者的手勢動作，因此智慧眼鏡可以藍牙技術讀取手環資料來得知使用者的手勢變化，進而操作景點資訊的影片播放，及切換影片等等功能。

市面上的智慧手環都是透過藍芽協定傳遞資料至手機，而在近四年來藍芽協定已發展到BLE 4.0，提供更省能源與更方便控制的無線傳輸技術。GATT是BLE 4.0將之前許多的ATT組合而成一個新的協定，GATT使得藍芽Service更加方便與精簡。

GATT主要有三大要點所構成：

- 1.Handle：索引
- 2.UUID：唯一識別碼
- 3.Value：儲存的值，可以是純數或是

Uuid等。

在使用 BLE 4.0 的 Service 之前，我們要先了解什麼是 GATT profile? GATT profile 是藍芽協會定義的一份公開文件，裡面描述了每個 Uuid 代表著哪一個 Service 與每個 Service 裡的 Value 代表著什麼意思，而使用者能透過文件去解讀每一個被定義在 GATT profile 下的 Service 與 Value。

表一、GATT 的 profile 架構範例

Handle	UUID	Value (meaning)
0x0009	0X2800	180D (Heart Rate)
0x000a	0X2803	10 000b 2a37
0x000b	0X2a37	(Heart rate measurement, only notify)
0x000c	0X2902	(Control channel)
0x000d	0X2803	02 000e 2a38
0x000e	0X2a38	01 (Chest)
0x000f	0X2803	08 0010 2a39
0x0010	0X2a39	(Heart rate control point)

其中，Handle 指的就是 profile 每個欄位的索引，每一個欄位一定只有一個獨特的 Handle，而為什麼 Uuid 被稱作唯一識別碼而不是 Handle 呢？答案在於擷取 Service 數值時，所有 Service 欄位在 GATT 的 profile 結構下 Uuid 一定都是以 0X2800 起頭，這就意味著，Uuid 0X2800 這個欄位一定就是 Service 的開頭標示，也因此 Uuid 又被稱為唯一識別碼。而每個欄位都會對應到一組 Value，若 Value 是另一個 Uuid，我們能再從 GATT profile 裡得知這個 Uuid 是什麼樣的 Service？以上

圖為例，Handle 0X0009 的 Value 為 180D，只要查詢 GATT profile 文件就會知道這是 Heart Rate Service。

下一欄位的 Uuid 為 0X2803，這 Uuid 被稱作 Characteristic 又被稱為「特徵」，每一個 Service 一定至少包含一個 Characteristic，使用者可以透過 Characteristic 得知是否可以取得 Service 的資料，而此例 Characteristic 的 Value 為三組資料 10 000b 2a37。第一個數字 10 代表後面數字的權限，000b 代表指定欄位的 Handle，2a37 代表該欄位的 Uuid，整合以上三組資料對應到 GATT profile 能得知，Handle 000b、Uuid 2a37 的權限為 10 (notification-only，通知)，所以只能透過 Handle 000b、Uuid 2a37 獲得通知，並不能取裡面的 Value，那要怎麼辦呢？我們再看到下一個欄位，Handle 000c、Uuid 2902，這 Uuid 又被稱作 CCC (Client Characteristic Configuration)，使用者要透過更改這欄位的值，才能取得 Handle 000b、Uuid 2a37 的 Value，CCC 裡有 [bit0] enable/disable notification、[bit1] enable/disable indication，我們要將 [bit0] notification 更改為 0X1，這樣才能從 Handle 000b、Uuid 2a37 欄位取得 Value (亦即 Heart rate measurement)，如此才能取得這 Service 所傳來的心跳數值。

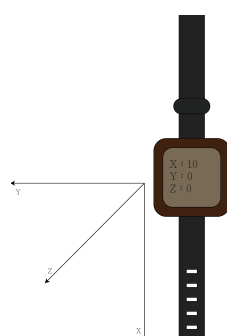
以上表格舉例來說，今天使用者想要取得 Heart Rate 那怎麼辦呢？在 GATT profile 中已經知道 Heart Rate Service 的 value 為 180D，因此，我們首先要搜尋所有 Uuid 為 0X2800 的欄位，並比對該欄位的 Value 直到找到 Value 為 180D 的欄位為止。這裡要注意一件事，所有的 Service Uuid 都是 0X2800，那為何又要找 180D 的 Service 呢？所謂的 180D 的 Service 是指 Uuid 為 0X2800 且

其Value為180D，並不是說Service的Uuid是180D，在GATT profile的定義裡180D是Heart Rate Service，若有Service沒有被定義在GATT profile文件內，那麼除了開發者外沒有人能知道這Service是什麼，也沒有人能知道他的Value代表什麼，以本專題為例，藍芽手環的三軸加速器Service就是開發者自己定義的，因此需要花時間去分析所獲得數值的意義。

找到Service起始欄位後，接著找這Service的Characteristic，Characteristic的Uuid都被定義為0X2803與Service的Uuid都被定義為0X2800是同樣的道理。當我們取得Characteristic的Value時卻沒獲得心跳值，而是一段字串10 000b 2a37，其中第一個數值10就是指這個Characteristic的權限（權限10為「通知」，上面已有提過），如果要更改權限，使用者要進而往下查詢Uuid 0X2902，而這Uuid就是上述所提的CCC (Client Characteristic Configuration)，必須要更動CCC的Value才能真的取得心跳值。

而本專題研究就是要取得手環上三軸加速器的Service資料，而這Service的資料權限就為「通知」，因此我們必須先去更改這Service的CCC才能獲得手環三軸加速器的數據進而判斷我們的手勢。

因為本專題所使用的藍芽手環內三軸加速器Service就是開發者自己定義的，因此我們必須先瞭解手環三軸X、Y、Z軸所代表的意義，示意圖如下圖一。



圖一、手環的三軸座標示意

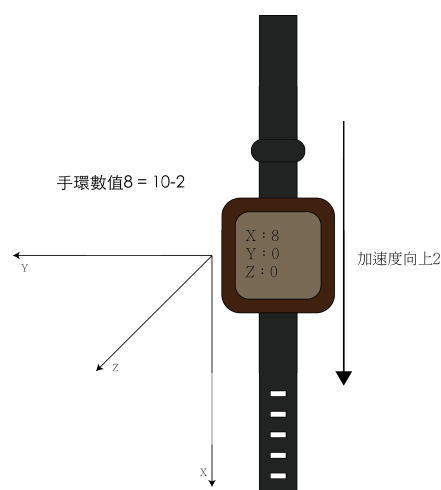
圖一的手環畫面上顯示著X：10、Y：0、Z：0，代表著目前只有X軸方向有加速度，而為何會是10呢？因為地球上有重力加速度9.8，因此垂直於地面的那一軸會顯示為10。

而當使用者移動手，手環偵測的重力加速度會改變，數值變化如圖二所示：



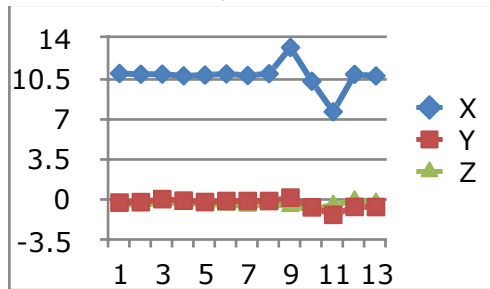
圖二、手環向上移動過程

手環向上移動後停止瞬間，因為慣性作用，因此手環三軸加速器會有反向的加速度，數值變化如圖三所示：



圖三、手環向上移動之停止瞬間

因此，上述動作會產生如下圖四的曲線圖。我們主要就是依上圖數據值的震盪來判斷手勢變化。



圖四、三軸向上移動的振盪示意圖

即時串流技術有幾個重要的關鍵元件：DSS(Darwin Streaming Server)、RTSP(Real Time Streaming Protocol)、與Android的MediaPlayer。運作上是MediaPlayer透過RTSP向DSS請求即時影片的串流服務。

Real Time Streaming Protocol (RTSP)，也就是即時串流協定，提供影片即時串流的服務，允許同時提供多個串流需求控制。

Darwin Streaming Server (DSS)，是第一個提供open source的RTP/RTSP伺服器的，是由APPLE公司開發，原本也只能用在OS X上使用，之後才有改成Linux及Windows的版本。

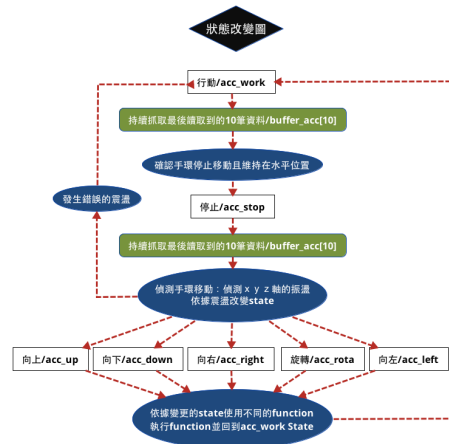
而MediaPlayer是Android系統所提供用來控制影片或音訊的class，Android系統還有另一個VideoView class來播放影片，操作上更簡單，不過因為本專題是要提供即時串流播放，因此選擇可以實現此功能的MediaPlayer來實作。

3.專題進行方式

我們使用Android系統下的GPS感測器獲得使用者位置資訊，進而向POI伺服器取得存在伺服器的景點資料，每個景點資料裡面主要包含有店家名稱、簡介、電話、經緯度與影片

等內容，擴增實境瀏覽器再將每一個景點資訊擴增顯示在眼鏡螢幕上。

瀏覽器接著透過BLE 4.0的功能取得手環的數據，進而分析數據取得手環內三軸的資料，再依據三軸數值的震盪來判斷手勢去操控介面。



圖五、手環的狀態變化圖

我們讀取最後10個連續數值去確認手環是否靜止不動並且符合我們規定的啟始手勢，這部分大約需要使用者靜止手勢約1到2秒，之後再依據數據震盪來判別手勢為哪一種狀態，而使用者的手勢與下一手勢之間必須間隔約為5秒，系統才能準確判斷。

但實際上，我們取得手環的數據卻是BA FE、E2 03等16進位的數值，因此，我們第一步驟就是要解讀這些數值，我們依據數值的跳動與手環畫面數值互相比對，大致上歸納出如下三點：

(1) 手環正負號是依據低位元的第一個bit而定，以BA FE為例，F二進位為1111，可知BA FE為負數，反之，若是數值為BA 0E，則所取得數值為正。

(2) 根據觀察可知FF FF為 -0，而00 00為 +0，而且數值依據16進位制跳動，正號時如 00 00, 01 00, 02 00

... FF 00, 00 01, 01 01, ...依此類推；而負號為FF FF, FE FF, FD FF ... 00 FF, FF FE, FE FE ...依此類推。

(3) 與手環畫面所顯示的數值對應比較，可觀察發現，當低位元進位時，手環數值會跳大約2.5，因此可以推測00 00為+0時，00 01為+2.5、00 02為+5、FF FF為-0、FF FE為-2.5……依此類推。

統整上面觀察到的數據，可以推出以下公式：

當正號時， $E2\ 03 = ((E2) / 255 * 2.5) + (03) * 2.5 = 9.71568$

當負號時， $E2\ F3 = ((255 - (E2)) / 255 * (-2.5)) + (255 - F3) * (-2.5) = -30.28431$

擴增廣告看板偵測部分則是使用OpenCV 2.4.10的OTSU先做影像二質化，然後使用Findcontour尋找邊界，再使用apollo Prox去辨識圖形，若是符合我們預設圖形就會擴增顯示影片在該看板上。

影片streaming部份的進行方式，首先要架好DSS並進入管理者頁面作一些基本的設定，像是設定廣播的密碼，設定要播放影片檔案的存放位置等等，DSS架好後，把想要在景點地標播放的影片檔放到DSS指定的路徑，並對檔案作一些處理：首先要將檔案轉成DSS可以傳送的格式，本專題是採用MP4檔案格式，然後要為MP4檔案加入hint track，此動作是為了讓server在即時串流時可以順利取得傳輸影片資料。當DSS架好，影片也處理好後，就可以透過景點資訊的server將user所想要觀看的景點資訊相關影片清單送到user端，之後user端就可以透過MediaPlayer在行動裝置(手機或AR眼鏡)上播放影片了，並且可以配合簡單的手勢的操作來控制影片。

MediaPlayer的運作方式可以參考

下一節「遭遇困難與解法」的MediaPlayer的生命週期圖(圖八)，這裡就簡單介紹一下MediaPlayer的一些狀態。MediaPlayer有幾個主要的狀態，分別為Idle、Initialized、Preparing、Prepared、Started、Stopped、以及End。當MediaPlayer剛被宣告出來時，會是Idle的狀態，此時我們得設定它所播放的檔案位置，因為我們要作多部影片的即時串流，並且希望在播影片的同時，其他影片可以同時loading，所以要有個MediaPlayer同時作Prepare的動作，因此要讓MediaPlayer先進入Preparing的狀態，當Prepare好之後MediaPlayer會進入Prepared的狀態，這時的MediaPlayer已經可以播放了，不過在播放之前，我們還得將MediaPlayer設定要播在哪個surface上，都設定好之後就可以開始在行動裝置上播放了，MediaPlayer播放後就會進入Started的狀態。當不播放時，可以讓MediaPlayer進入Stopped的狀態並重新prepare一次。而當我們要切換影片時，必須要把MediaPlayer給released掉才可以讓下一個MediaPlayer在同一個surface上播放，而released掉的MediaPlayer就會進入End的狀態。

DSS是透過RTSP(Real Time Streaming Protocol)傳輸影片給MediaPlayer來播放。RTSP是應用層的通信協定，定義於RFC2326中(RFC是由網際網路工程任務組(IETF)所發簿的一系列備忘錄，收集了網際網路的相關資訊，如規範，協定等等)，RTSP是用來控制具有即時與同步特性的資料，例如音樂和影片，這裡我們就是用RTSP的影片即時串流功能來播放影片。RTSP沒有連線的概念，而是在伺服器維護ID以識別播放期間(session)，RTSP也沒有限制底下傳

輸層的通訊協定，可以用 TCP 或 UDP 來傳送 RTSP 訊息。USER端是透過URL來讀取及播放想要播放的影音媒體，以rtsp://開頭表示用TCP來傳輸，而rtspu://則是以UDP來傳輸，沒有指定port number的話，RTSP 預設的port number是用 port 554。

開發過程中遭遇的主要困難與解決

- 困難 1：

定義手勢種類與手勢的初始位置。

- 解決：

三軸數值只能表示一個二維空間—加速度向量，若要升階為三維空間，必需要清楚規定其初始位置，我們設定手臂必須平行於地面，程式才能進入初始狀態開始偵測手勢，而手勢種類分別為向上與向下、向左與向右、以及手軸旋轉，依據人類的習慣，向前與向後的手勢動作並不流暢因此不納入考量。

- 困難 2：

手環並無左右之分，因此很難分辨左與右手勢。

- 解決：

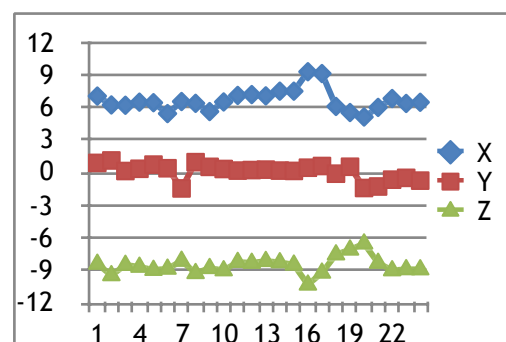
先規定使用者哪一隻手戴手環，再依據使用者的習性來判斷左與右，畢竟沒有人會反著戴手錶。

- 困難 3：

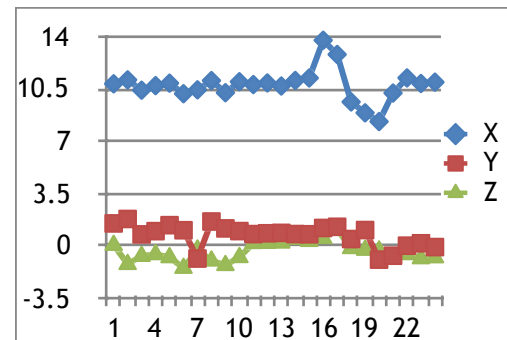
手環三軸的數值震盪不夠明顯。

- 解決：

透過將三軸座標系校正為其中一個軸向上，震盪更為明顯與精準。



圖六、手環三軸加速器的原始值



圖七、手環三軸加速器數值校正後

- 困難 4：

架設DSS server時遇到了一些問題：像是在安裝失敗、或是安裝成功後，server的管理頁面文字都是亂碼顯示。

- 解決：

後來發現是作業系統的版本及位元問題，因此最後採用12.04版本的ubuntu，並且可能得加上一些32位元的library，包括 lib32stdc++, lib32z1, lib32z1-dev。

- 困難 5：

影片即時串流，一開始使用VideoView來播放影片，操作及設定都簡單，不過一次只能準備單一影片，想要播放第二部影片就要等系統重抓影片並準備好後才可以播放，而這一動作會耗費數秒到數十秒。

- 解決：

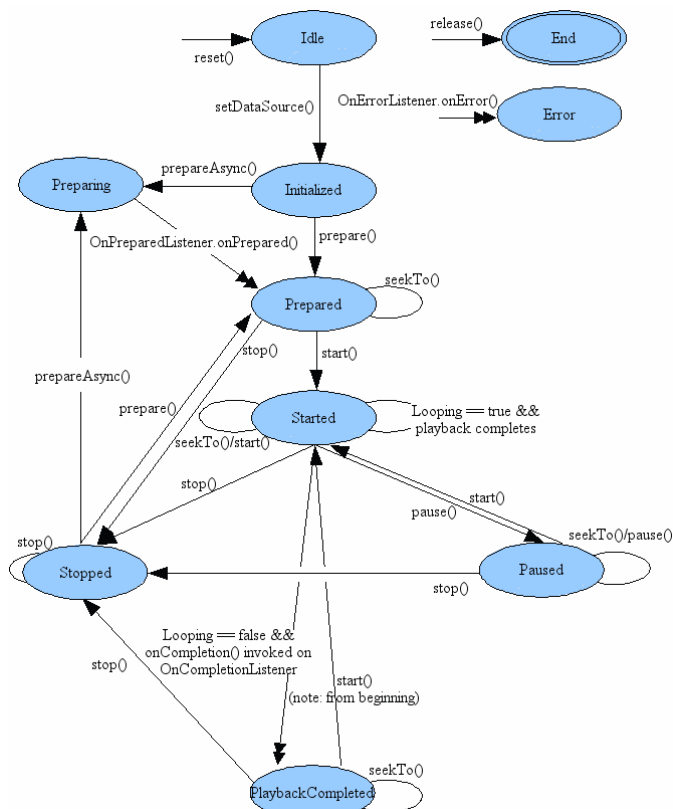
改用MediaPlayer來播放，使用MediaPlayer直接指定什麼時候讀檔，什麼時候播放影片，因此可以準備多個MediaPlayer來準備影片，並讓準備好的影片在指定的時間播出。

- 困難 6：

要重新播放上一部影片時要等很久。

- 解決：

由於影片要切換到下一步影片時必須要把整個MediaPlayer release掉，因此當切換到下一部影片時，原本的MediaPlayer被release掉了，若這時想播放原影片會發現MediaPlayer正在重抓，因此多準備了預備用的MediaPlayer供往回播放使用。
(MediaPlayer的運作如下圖所示)



圖八、Media player生命週期圖

4.主要成果與評估

我們使用的智慧手環內建的感測器只有三軸加速器，因此手勢判斷起來依然會有些許誤差，但根據我們測試的結果，三十次內只會發生低於一次的錯誤判斷。

影片的部份如果正常播放都沒問題，但是影片切換太快的話，就要等待影片prepare好才可以播放，解決方

法大概只能增加MediaPlayer個數，畢竟一部影片從prepare到ready還是要好幾秒的時間。

5.結語與展望

手勢與影像辨識的操作將會是未來的一種趨勢，而在應用的部分，本次專題只使用了手勢與影片的結合，這還只是現在體感操作的冰山一角，若能實際應用在許多穿戴式裝置上，相信能帶給更多人便利與福利。

6.銘謝

感謝指導教授的教導，告訴我們如何結合裝置與應用，並且給予我們許多建議，也感謝實驗室的楊博宇學長，從零開始教導我們如何使用Android系統。

7.參考資料

參考:

MediaPlayer :

<http://developer.android.com/intl/zh-tw/reference/android/media/MediaPlayer.html>

hint track :

http://www.courseschevaux.fr/medias/videos/mpeg4ip-1.5.0.1/doc/about_hint_tracks.txt

<http://www.race604.com/gatt-profile-intro/>