

摘要

在現今這忙碌的社會中，休閒娛樂是身心俱疲的我們一項不可或缺的舒壓活動，而其中體感遊戲可說是相當的熱門一種，在某次機緣巧合下，我們在同學家玩過 XBOX360 後，發現了大部分遊戲都是藉由對人體姿態的辨識來進行比對，判斷出所相對應的結果，並在遊戲中呈現出相對的動作反應，我們對這一部份感興趣，於是投入研究中，研究中我們預備發展出一套可以利用 Kinect 辨識人體行為、動作，目前要應用於辨識連續動作。

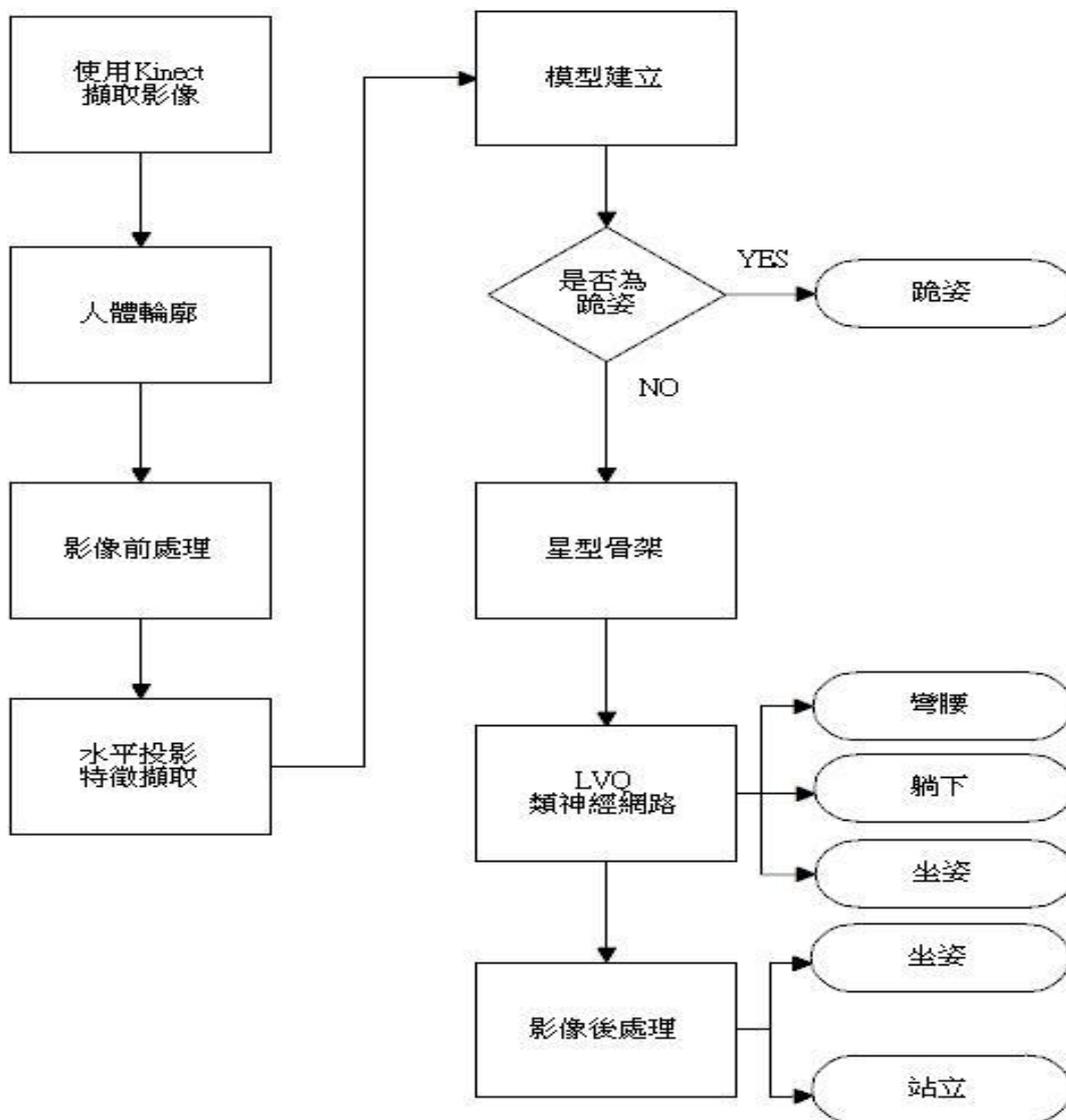
研究中我們學習到應用 Kinect 來做人體動作的辨識，並應用背景相減法、星型骨架、類神經網路、相似特徵提取等技術，來辨識動作。當 Kinect 擷取到人體影像後，藉由 Kinect 深度的資訊，然後經過前處理(侵蝕或膨脹)取得較佳的人體輪廓。接著用水平投影量來辨識運動情形，之後使用星型骨架取得由人體重心點至輪廓特徵點的特徵向量，此特徵向量與人體深度資訊為 Learning Vector Quantization (LVQ)類神經網路的輸入；接著 LVQ 網路的輸出便可辨識出人體動作，此外也可利用 Support Vector Machine(SVM)進行分類。如有相似特徵，會藉由水平與垂直投影，計算出人體長度與寬度比率再加以辨識，以提高辨識率。

目 錄

一、製作進度概況.....	3
流程圖.....	4
1. Kinect 影像擷取	5
2. 人體輪廓	8
3. 執行進度	9
二、目前成果與預訂進比較.....	15
三、遭遇困難與解決方法.....	16
四、指導老師意見(簽名).....	17

一、 製作進度概況

首先，我們資訊大致上都掌握了，並且取得了Kinect SDK的彩色、深度、骨架相關資訊；從資訊中提到了人體運動姿勢的辨識方法，大致上可以分為兩類，第一類為模型建立方法，會將人體影像套上一個定義好的幾何模型或是輪廓模型。第二類則是不需套入定義的人體模型，主要是利用輪廓比對，或是在輸入影像後，擷取特徵點來進行訓練和辨識。在輪廓比對上，我們必須要取得輪廓，而使用的方法為背景相減法，如連續影像相減法(Frame Differencing)等…，這種方法是使用事先建立場景做為參考背景影像，利用連續兩張或相隔一段距離之影像進行相減，本方法不易受到光線的影響，在動態的環境的偵測上有不錯的偵測效果；但在物體靜止時就無法找出其中的差異，所以我們尋求另外一種方法為背景相減法(Static Background Subtraction)，這個方法是使用事先建立好的背景影像和目前所輸入的影像做相減的動作，但容易受雜訊影響。之後我們修改Kinect SDK的應用程式，可以替代前者兩種方法，大幅減少我們擷取輪廓資訊的速度，但是，取得人體輪廓後，在某些地方會有雜訊，邊緣也不是那麼平滑，這時利用型態學中的侵蝕或膨脹進行前處理，可使影像看起來更加平滑。



圖(1). 辨識流程圖

1. 使用 Kinect 擷取影像

1.1 Kinect 硬體與原理介紹



圖(2). 硬體

Kinect 可以取得下列資訊：彩色影像(中間的鏡頭)、3D 深度影像(左右 2 顆鏡頭)、紅外線發射器和紅外線 CMOS 攝影機(左右 2 顆鏡頭)、聲音(麥克風陣列)。Kinect 也支援追焦功能，底座馬達會隨著人物轉動。

感應項目	有效範圍
顏色與深度	1.2~3.6 公尺
骨架追蹤	1.2~3.6 公尺
視野角度	水平 57 度、垂直 43 度
底座馬達旋轉	左右各 28 度
每秒畫格	30FPS(Frame Per Second)
深度與解析度	QVGA(320×240)
顏色解析度	VGA(640×480)
聲音格式	16KHz, 16 位元 mono pulse code modulation(PCM)
聲音輸入	四麥克風陣列、24 位元類比數位轉換(ADC)、聲音雜訊

Kinect 詳細規格表

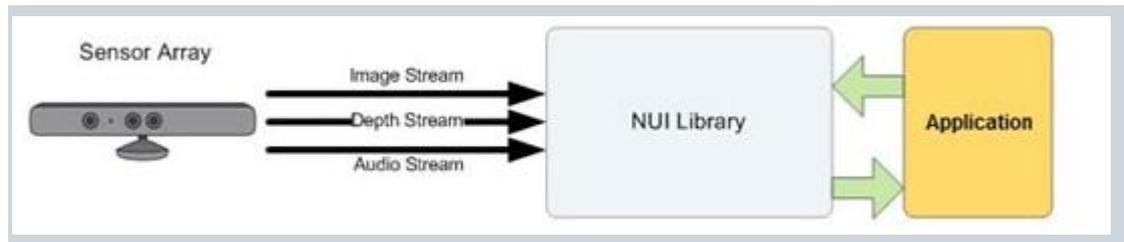
1.2 Kinect 開發環境與設定

在進行 Kinect 程式開發之前，需要準備的軟硬體如下：

- ◆ 作業系統：Windows7(x86/x64)
- ◆ 硬體：CPU：雙核心 2.66GHz 以上
RAM：2GB 以上
顯示卡：支援 DirectX 9.0c 以上
Kinect 感應器
- ◆ 軟體：Visual Studio 2010 或 Visual C# 2010 Express
NET Framework 4.0
Kinect SDK for Windows
- ◆ 安裝 Kinect For Windows SDK 步驟：
 1. 先不要接上 Kinect 感應器
 2. 不要開啓 Visual Studio 2010
 3. 安裝對應的(64bit or 32bit) Kinect SDK
 4. 安裝後請重新開機，以便識別 SDK 所需的環境變數
- ◆ 安裝 Coding4Fun Kinect Toolkit：

這個工具包主要是將一些在開發 Kinect 應用程式時會使用到的程式碼整理成擴充方法，可在開發時減少程式碼的撰寫。

1.3 Kinect for Windows 架構



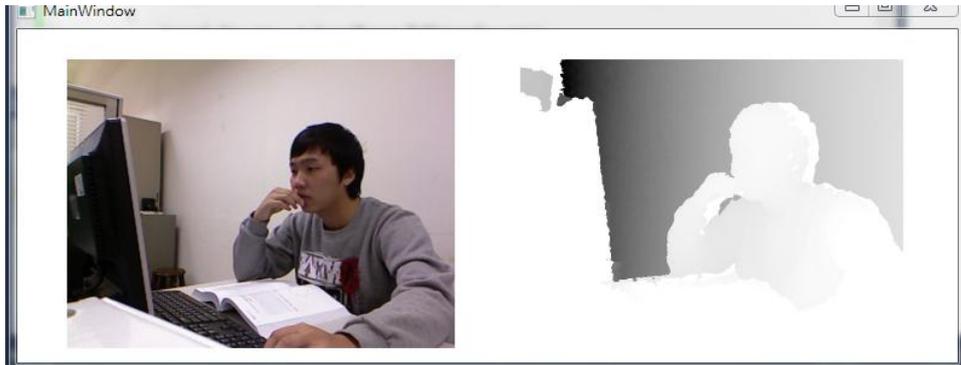
圖(3).Kinect 系統架構

Kinect 的 NUI 程式庫提供應用程式取得 Kinect 感應器傳送至主機的三種訊號串流，必須在初始化 API 時指定要接收哪幾種串流，彩色影像串流、深度影像串流、聲音串流。

1.4 NUI API 初始化

要使用 Kinect API 接收感應器的資訊，是透過 Runtime 物件，因此 Kinect 應用程式的第一步就是建立一個 Runtime 物件來準備接收感應器的資料，並呼叫 Initialize 方法進行初始化(指定接收哪種類型的資料)，在應用程式結束時要呼叫 Uninitialize 方法，關閉 Kinect 設備。

1.5 用 Kinect 取得彩色影像



圖(4). 彩色影像(左)、深度影像(右)

2. 人體輪廓

輪廓是在亮度不同的區域之間有一個明顯的變化，利用 Kinect SDK 應用程式擷取出人體輪廓後，針對其進行行為分析；亦或透過輪廓取得身體各部位的資訊。

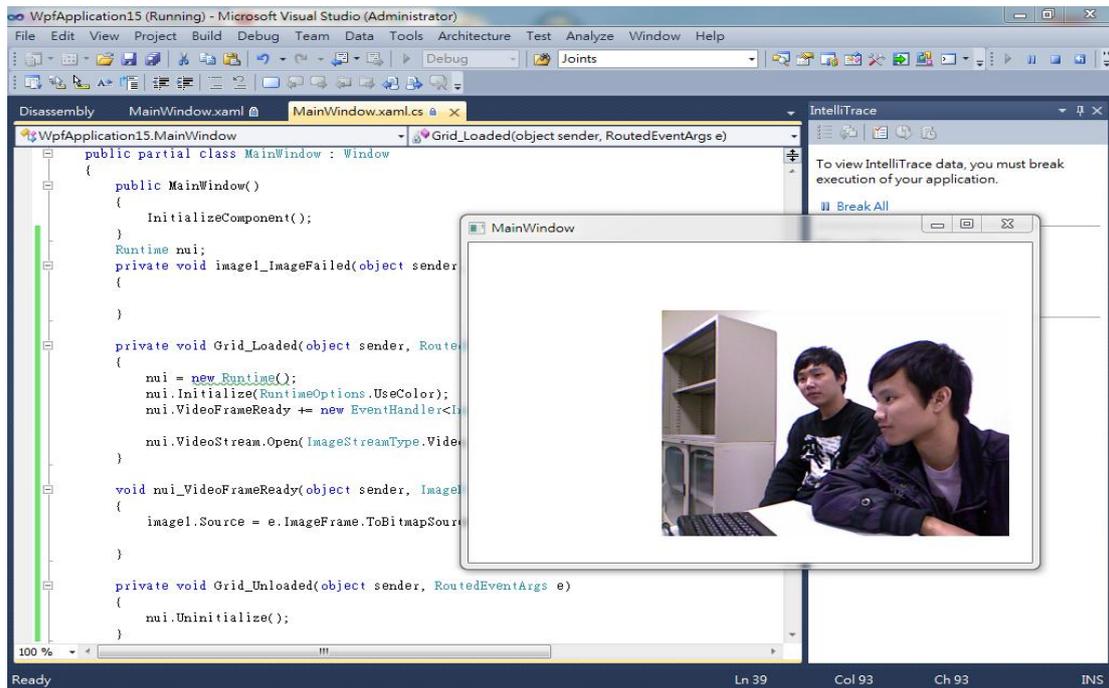


圖(5). 人體輪廓

3. 執行進度

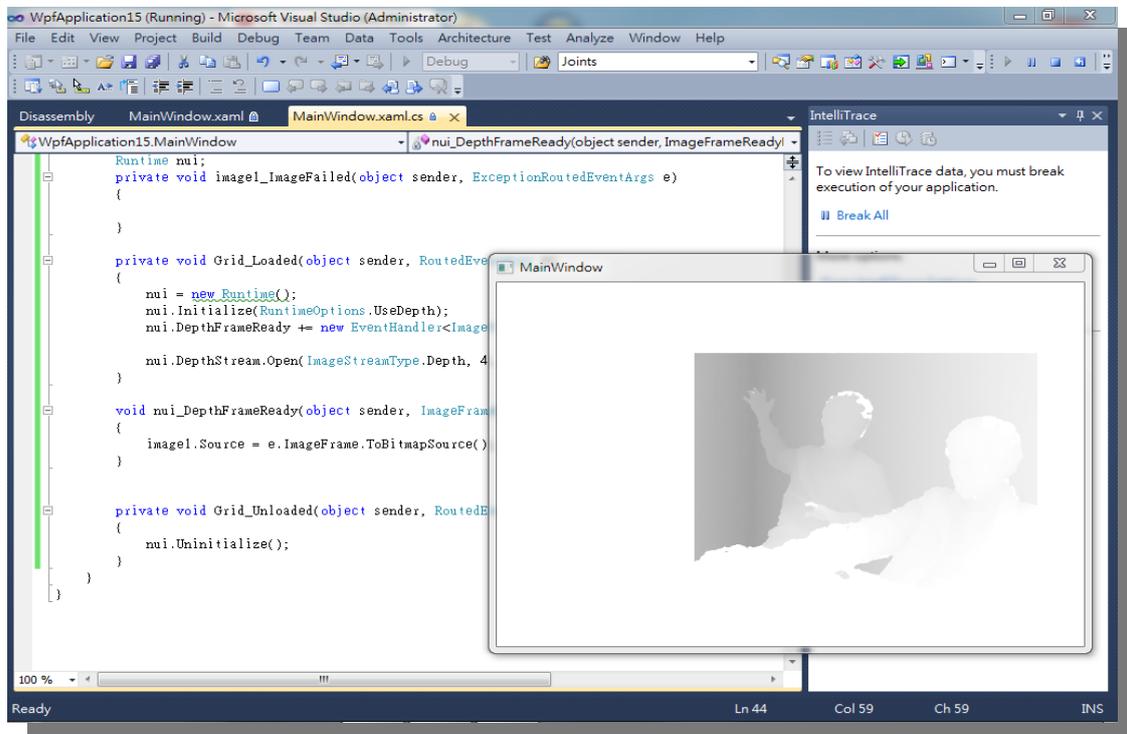
我們將 Kinect SDK 的環境架設完成，使用 Kinect 取得了人體的深度，追蹤使用者的動作、輪廓、骨架資訊，找了許多範例程式，藉此熟悉 Kinect 的操作和從中找尋有幫助的程式，從結果中討論有關輪廓或解析度等變化，在過程中我們嘗試將燈都關起來，也能擷取到影像，這是因為影像是透過紅外線發射器和紅外線 CMOS 擷取到的紅外線會發出人們肉眼看不到的紅外線光線去照亮被拍攝的物體，紅外線經物體反射後進入鏡頭進行成像，這時我們所看到的是由紅外線反射所成的影像，而不是可見光反射所成的影像，即此時可拍攝到黑暗環境下肉眼看不到的影像，而當遇到透明物體時，會因為反射的困難，無法正確取得深度資訊，當人物與 Kinect 距離超過 4 公尺也無法擷取到輪廓。學習 OpenCV 的電腦視覺技術，架設 OpenCV 的環境，由於 Kinect SDK 只能用 C# 取像，我們並將程式與 OpenCV 結合，運行 OpenCV 的範例程式，並運用 OpenCV 與 OpenNI 獲得 Kinect 資訊。找尋關於特徵的資訊。學習關於辨識方面的 paper，並找尋演算法和程式碼。

參考蒐集到的網路範例程式所取的彩色影像



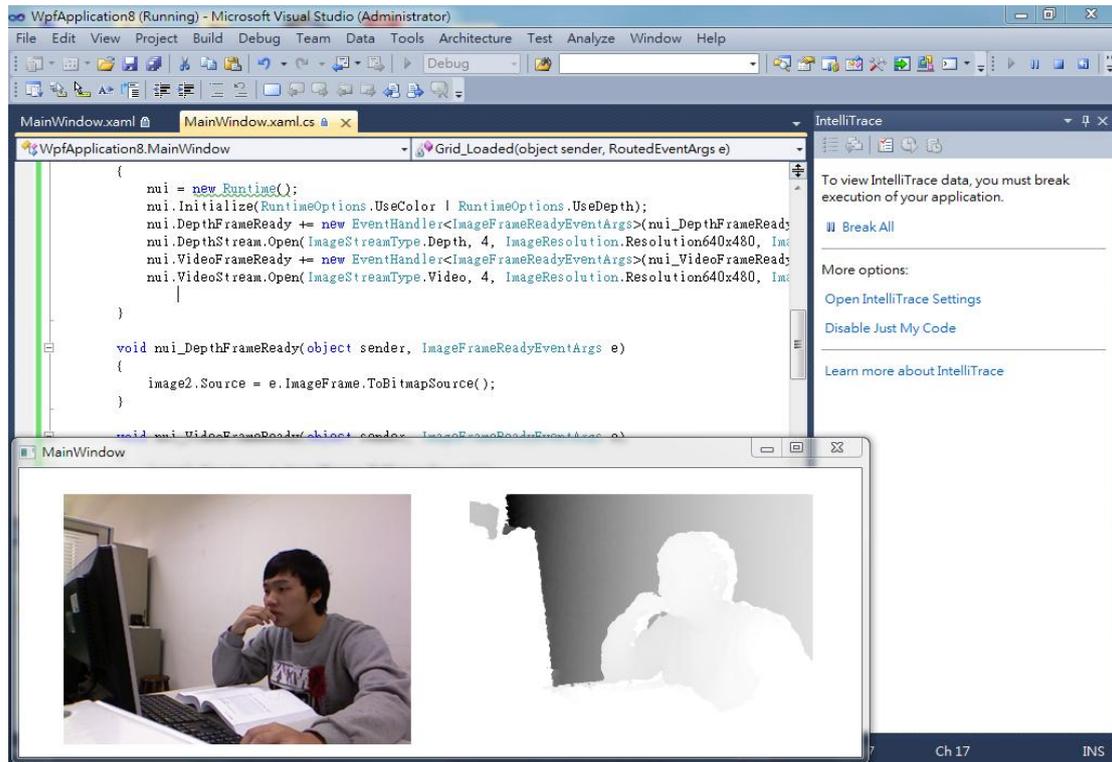
圖(6). 彩色影像

參考蒐集到的網路範例程式所取的深度影像



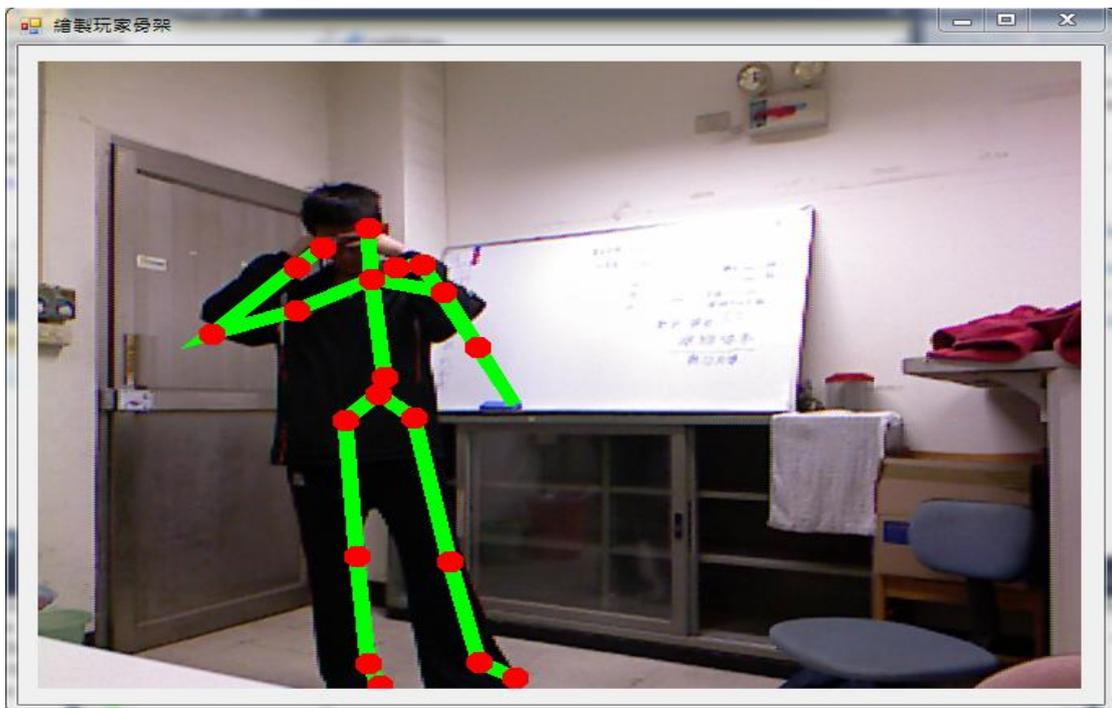
圖(7). 深度影像

將彩色與深度影像一起呈現



圖(8). 彩色與深度影像

參考蒐集到的網路範例程式所取的骨架資訊



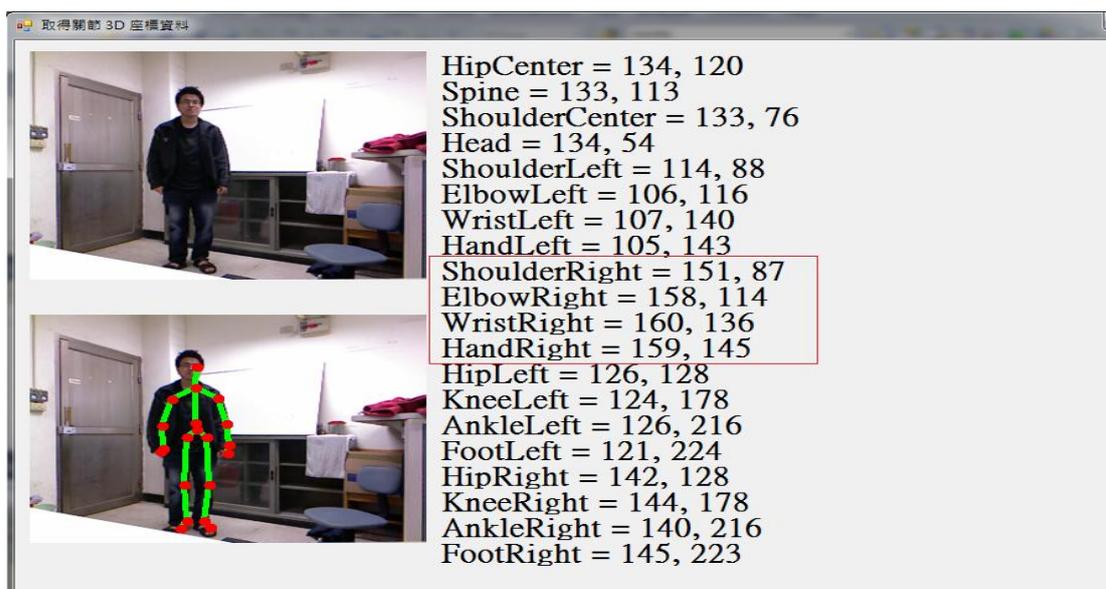
圖(9). 玩家骨架

結合了深度、骨架和我們所修改的座標與距離資訊，從影像可發現深度與骨架有些微的 Delay，距離也有錯誤。



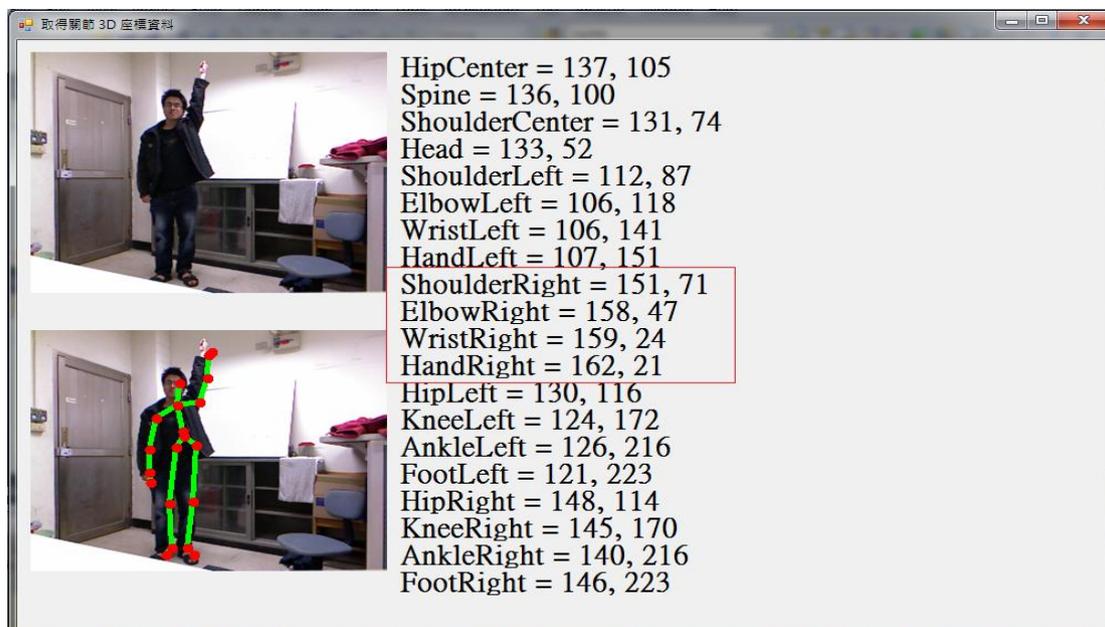
圖(10). 座標與距離資訊

將圖(10)的程式經過修改，得到正確的座標資訊，框起來的部份前項為 x 軸，後項為 y 軸，與圖(12)做比較。



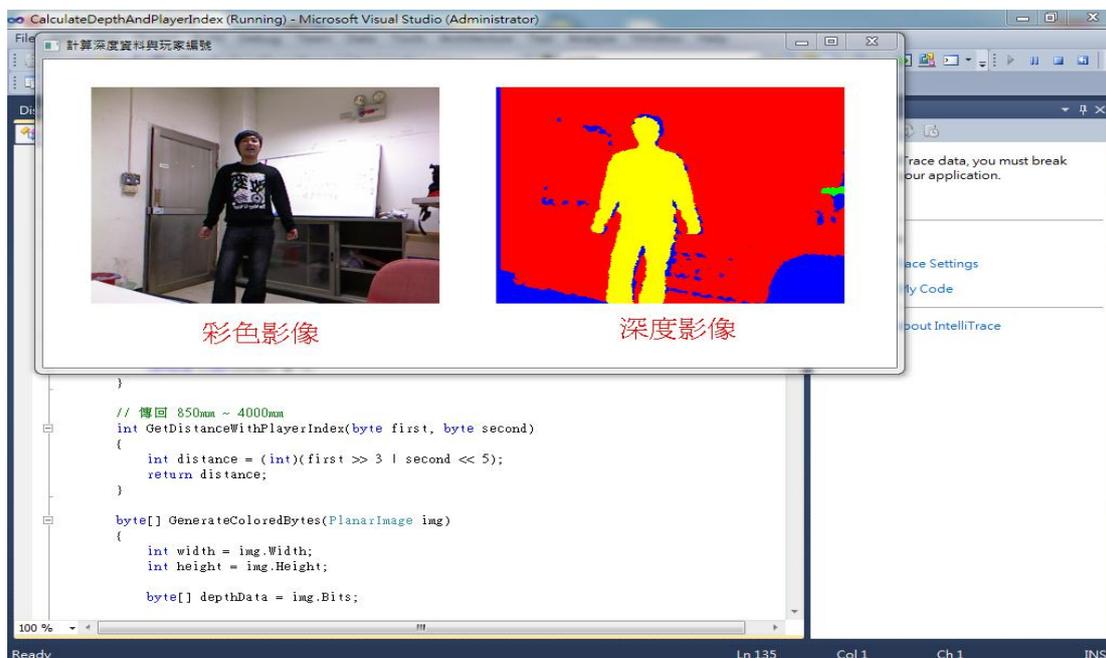
圖(11). 座標資訊

與圖(11)做比較後，可發現 y 軸部分因左手的抬起有明顯的減少，這是因為此圖座標的原點為左上角，x 軸越往右越大，y 軸越往下越大。



圖(12). 座標資訊

將深度範例程式加以修改後的深度資訊，每種顏色代表不同的深度。



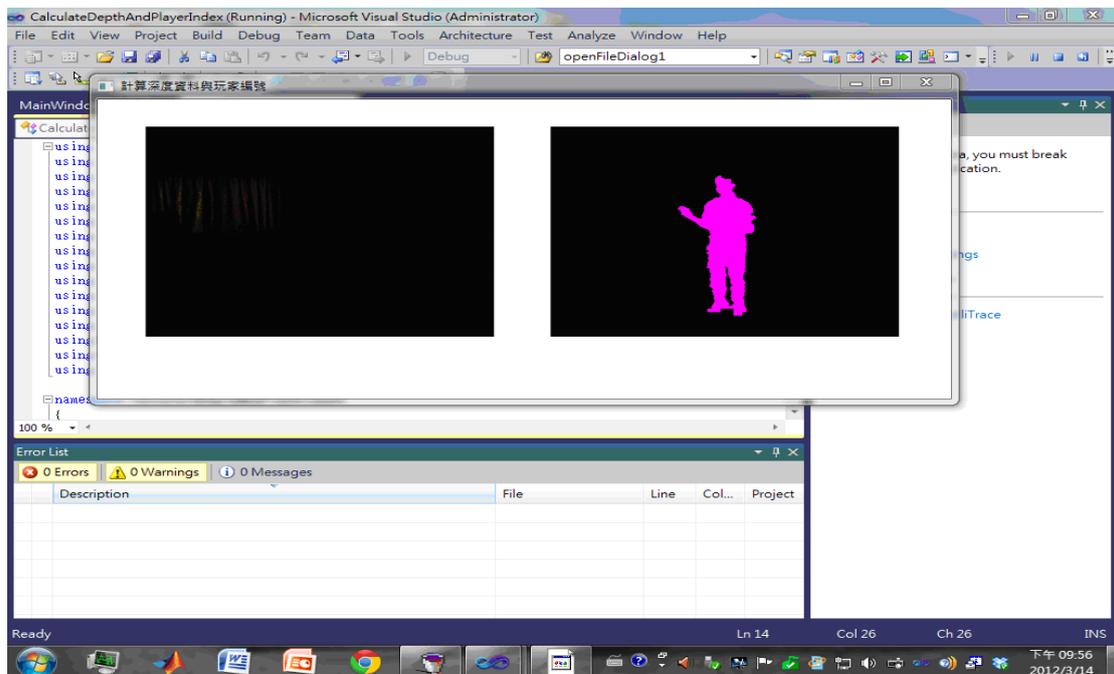
圖(13). 彩色與深度影像

將圖(13)的程式加以修改，使深度資訊能夠呈現出背景物跟前景物。



圖(14). 彩色影像與人體輪廓

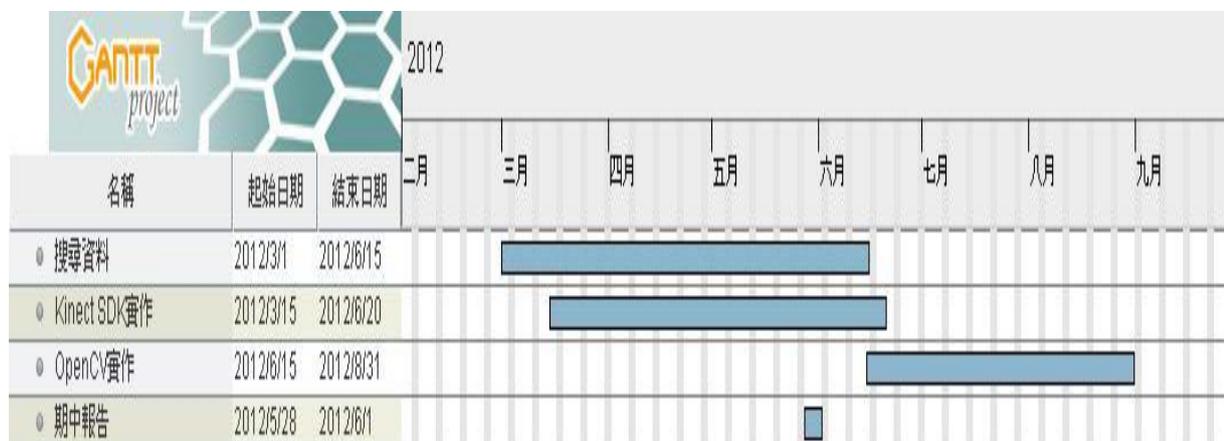
將室內的燈關閉後，因為影像由紅外線鏡頭所擷取，我們依然能夠取得人體輪廓。



圖(15). 將燈熄滅後的人體輪廓

二、 目前成果與預訂進度比較

預定進度表：



各項任務預定時間表

編號	任務名稱	開始時間	完成時間	完成比例
1	搜尋資料	2012/3/1	2012/6/17	80%
2	Kinect SDK 實作	2012/3/15	2012/5/31	80%
3	OpenCV 實作	2012/6/15	2012/8/31	5%
4	期中報告	2012/5/28	2012/6/1	100%

目前進度大概說明如下:搜尋資料與 Kincet SDK 實做仍在持續進行中，OpenCV 已安裝完成，其環境大致已架設完畢，原定進度與目前進度相差不大。

三、 遭遇困難與解決方法

最困難的部分還是在 Kinect 的程式碼，包括如何與 OpenCV 和 OpenNI 整合，以及如何儲存連續影像、如何取得特徵點…等，這些都還要持續的上網搜尋資訊與相關的程式。此外運動模型以及訓練 data 的建立，還有辨識的方法等，也是要上網找一些相關的 paper 來看。另外在 OpenCV 的安裝也費了很大的功夫，尤其是在環境的建立、路徑的設置、如何運行，也從老師、網路和書本上取得了資訊。而另外的五項困難就是：

1. 光線因素：雖然擷取影像是在室內但室外光線投射入室內所造成之陰影變化，以及人體運動過程中各部位造成之陰影都會造成辨識的困難。
2. 背景因素：當人物跳動影響地面引起的 Kinect 震動會產生背景的變化，背景的改變會造成影像校正困難。
3. 人物的衣著：如果所穿著的服裝與背景顏色太過雷同，則會影響系統的辨識率。
4. 即時性：必須快速的對影像做分析處理，否則將造成系統的實用性下降。
5. 由於人體所作之運動屬於形變物體的運動，要對形變物體進行動作偵測，也是這個研究的困難。

四、 指導老師意見(簽名)

1. 在期中報告前須完成專題題目相關背景知識的理論分析並完成基礎系統程式的撰寫。
2. 其透過第一學期的研討，了解專題题目的困難點，擬定可改進的突破點。
- 3 在未來的第二學期，針對困難點進行研討與改進，並期許同學發想新的創意。

指導老師簽名：_____