1. A badminton teaching assistant App

我們在前節中已有論述,許多研究顯示運動教學輔助工具非常有益學員的學習,而目前市面上已經有棒球揮棒動作的分析器Zepp Baseball [8]，供學員知道自己揮棒的軌跡，並且提供知名選手的揮棒資料。在研究上則有高爾夫球的揮杆動作分析教學[9]，學員姿勢與標準姿勢進行比對與同部播放，讓學員知道自己需要改進的地方。綜觀上述兩項運動的共同點為：都是靜態分析,即學員(或被拍攝者)的位置是固定的。而本計畫的研究項目羽球，被拍攝者的位置則不斷地改變,因此教學輔助工具的開發更具挑戰性與創新性。本計畫的其中一項目標就是開發一套具有回饋功能的羽球教學輔助軟體,而所謂回饋功能是指這套軟體能指出學員的揮拍錯誤,讓學員在課後也能自主地利用本工具學習。

一般來說,因為學員看不見自已揮拍的動作,時常無法領會專業教練指出的揮拍錯誤，因此我們開發的羽球教學工具將利用攝影機拍攝學員與專業羽球選手的揮拍影像,讓學員比對。揮拍的動作非常快速、短暫,專業的教練能從短暫的過程中分辨出細微的揮拍錯誤,但一般的學員就必須由慢動作的撥放,才能體會自己揮拍的錯誤。因此，提供慢動作的撥放是本工具的一項重要功能。另外，為了達到回饋的功能進而判斷學員的揮拍正確與否,電腦必須有學員身體各部位的相對位置，如此才能在揮拍過程中的一些關鍵時刻，判斷學員身體的一些部位是否在合理的範圍之內。也就是說，為了達到回饋的功能，電腦必須有景深的資訊。傳統攝影機會將三度空間的場景紀錄在一張二度空間的圖片(frame)上,因此失去場景的景深。人類感官的感知功能以及大腦對場景的認識，可以在腦海中將二度空間圖片中的物體,重建回三度空間的場景。也就是說專業的羽球教練可以單憑觀看一段二度空間的揮拍影片，便指出學員的錯誤，但電腦卻不行。這個感官感知的功能非常複雜,科學家也才開始探索這個領域 [13] 。一般來說，從二度空間的圖片重建回三度空間的場景是個非常困難的電腦影像處理問題，目前學者已經提出的方法也只能適用特定的場景[14,15]。因此，我們決定運用較新的kinect攝影機來幫助我們實做出羽球的動態分析，因為kinect攝影機能讓我們紀錄三度空間的影片。

我們把羽球教學輔助軟體的實做分成三部分說明，分別為Kinect 骨架追蹤、慢動作播放與使用者介面、 比對評分。下面先簡介三個部份，之後再依序詳細的介紹。

* Kinect 骨架追蹤：利用kinect錄製影像，用微軟提供的API來進行骨骼追蹤，取得我們需要的資料。
* 慢動作播放與使用者介面：透過慢動作播放讓學員可以清楚知道自己的動作。我們目標是同步播放學員與標準動作，讓學員可以更清楚知道自己與標準動作的不同之處，更明確知道自己的缺失。
* 比對評分：透過Kinect所提供學員的骨架資訊與標準動做的差異性進行評分。比對不是單純的frame by frame比對，而是透過時間軸去分析學員的反應與動作。最後再與標準動作進行比對，分析差異性後並給予分數。

**3.1 Kinect 骨架追蹤**

在不久以前，為了建造出一個物體3D的模型，需要至少2台以上的攝影機，而為了得到 一個物體的骨架圖，物體則需要穿戴額外的感應裝置。同樣的實驗透過Kinect cameras則可以相對輕鬆地達到我們的目的，一台Kinect cameras包含三個攝影機，且可以偵測物體與攝影間的距離。此外，Microsoft(製造商)提供骨架資訊的API，方便使用者建構所攝物件的3D骨架圖，Kinect cameras已經廣泛被運用在研究電腦視覺的各種問題，舉例來說，作者[2,3]使用Kinect cameras來研究交響樂指揮的動作， 然後對於一個指定的音樂片段，合成電腦產生的指揮的影片。作者[4]則利用Kinect cameras收集3D動作的資料來研究posed-based 3D人體動作認知的分類方法。

透過kinect，我們可以直接使用現成的函式庫來進行骨骼追蹤，甚至生成相關的骨骼資訊來做後續的比對處理，以下是簡介骨骼追蹤的運作原理：

1.利用CMOS紅外線感應器，透過黑白光譜的方式來感知環境。純黑代表無窮遠，純白代表無窮近。黑白間的灰色地帶則對應物體到感應器的物理距離。

2. 收集視野範圍內的每一點，形成一幅環境景深圖像。以每秒30幀的速度生成景深圖像流，產生3D圖像以再現周圍環境。

3. 透過遮罩來減輕機器計算量。

4. 一個玩家有多個關節座標(x,y,z)。關節的物件型態為Joint，Joint物件的屬行為Position，而Position的型態為vector類別，裡面有X,Y,Z值。而每個關節有對應的狀態：

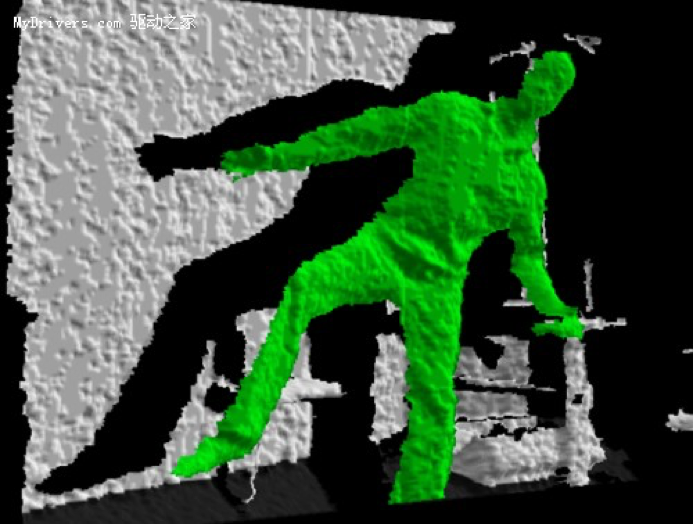
Tracked：表示被追蹤到了，關節資料可使用

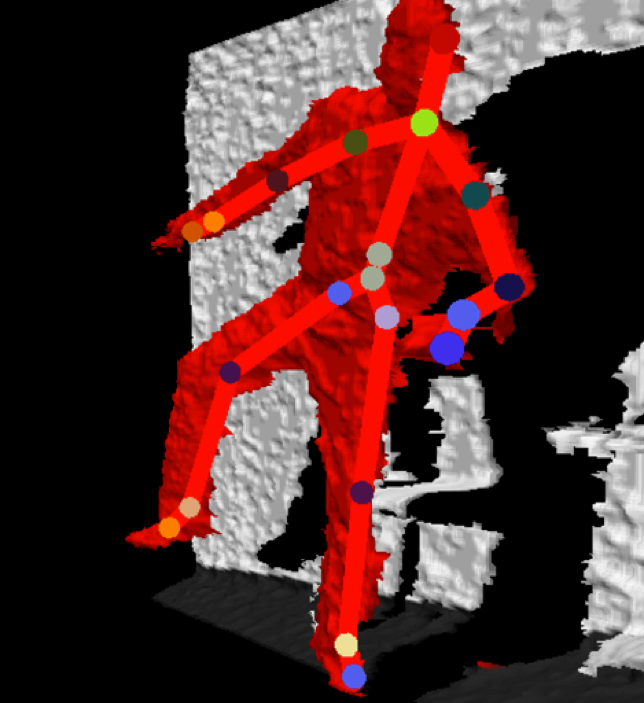
PositionOnly：表示被裁切掉、難以判斷、無法100%確定

NotTracked：表示該關節沒被追蹤到(很少發生，但需注意此狀

態)

5. 透過向量與矩陣運算，把關節座標轉換成螢幕座標。

6. 根據追蹤到的20個關節點來生成一幅骨架系統圖。

 Figure 3.1 (環境景深圖，綠色為加工過) [10] Figure 3.2 (輸出骨架圖) [10]

Kinect可以為我們的專案帶來很多好處。我們可以透過RGB camera取得的色彩影像，就像一般的攝影機錄製影片，並透過後端運算實現慢動作播放。在慢動做播放下進行比對，可以讓學員知道自己動做與標準動做的差異，以提升學習效率。

透過Kinect建構3D的空間資料，易於我們取得立體的空間資訊，而非一般錄影機將3D的空間資訊壓縮成2D的平面資料，這樣可以大幅減少我們花費時間在空間轉換上。如此，我們不需使用2台以上的攝影機，便可以透過座標轉換得到物件的各個角度的視角。

在先前介紹kinect所提供的骨骼追蹤，可以簡化我們實做出比對系統。我們不需要透過複雜的圖像演算法來取的關節節點座標。我們只需要呼叫定義好的函式即可以取得我們所需要的資料，把資料交給後端運算進行比對，告知學員的評分。甚至我們可以把骨骼資料鑲在慢動作播放，讓學員可以更清楚地知道自己的骨骼動作為何。

**3.2 慢動作與使用者介面**

在本小節中我們先介紹慢動作撥放的實作原理,再介紹本軟體的使用者介面。

本計畫將要加以分析的大部分都是揮拍這種瞬間性的動作，若不能將影片以慢動作的方式呈現，學員則無法清楚的看見自己細微的錯誤，因此慢動作的應用在本計畫也是一項主要的技術之一。如眾所皆知的,一段影片是一連串的畫面(frame)所組成。慢動作撥放的技術主要是在原有的frame跟frame之間插入額外的frame達到將影片放慢的效果 [5, 6, 7]。舉例來說,欲達成1/4的慢速撥放，就要在連續每兩張原有的frame之間插入3張額外的frame。而插入的frame上面畫素(pixel) 的決定方式又分為兩種，一為較簡略的「線性插入(linear interpolation)」，二為較精密的「移動插入(motion interpolation)」 [5, 6, 7]。線性插入僅利用插入的frame的前、後原始frame做加權平均來計算插入frame上的pixel的值。假設連續的前、後兩張原有frame的時間點分別為t0 和t1 ，而在這兩張原有frame中欲插入的第i張frame的時間點為t。我們利用t在t0 和t1之間的位置來決定加權的係數

wi = (t - t0) / (t1 - t0) 。

有了加權係數後，就可依照以下加權公式計算第i個插入frame上位於(x, y)位置的pixel的值Fi,x,y。

Fi,x,y = (1 - wi)F0,x,y + wiF1,x,y

本計畫希望實作的則為較精密的移動插入。移動插入主要會用到「光流(optical flow)」的概念。光流(參見文獻[11])，簡單來說，就是在畫面中物體的移動；較為精確的說法則為，在一個向量場內，於前一個frame裡每一個可見的點都能透露出下一個frame相對應的點的位置。假設在第i個frame時A點的位置是(x1, y1)，若能找到第(i+1)個frame上對應的A點並確定其位置為(x2, y2)，那麼就能找出A點在第i個frame與第i+1個frame之間的移動向量：

(ux, vy) = (x2, y2) - (x1, y1)。

不過該如何預測出A點在第i+1個frame上的位置呢？OpenCV有許多函式能達到目的，包括calcOpticalFlowPyrLK能計算出稀疏光流（某些點集）、calcOpticalFlowFarneback能計算出稠密光流（圖像上所有像素點的光流都計算出來）、CalcOpticalFlowBM、CalcOpticalFlowHS、calcOpticalFlowSF......等。而除了既有的函式，也有很多計算光流的方法，如相互關係法(correlation method)、區塊比對法(block matching method)、特性追踪法(feature tracking method)、以及能量法(energy-based method) [11] 。我們現在簡單地介紹區塊比對法：將一個frame分割為許多方塊，再將每個方塊與下一個frame一定範圍內的每個方塊相比，相差最少的那個方塊則為最佳位置。而最佳位置與目前位置的差值，就是目前位置上物體的光流。

移動插入即為利用區塊比對法或其他函式庫所找出的光流，製作欲插入的frame。以下為求簡化，用Vx,y表示光流。想像新的frame上的某一個點A由上一個原始frame(假設時間為t0)沿著一個向量Vx,y而來，並將沿著相同的向量Vx,y而去到下一個原始frame(假設時間為t1)。假設新的插入frame的時間點為t，而前、後原始frame的時間點分別為t0 和t1。我們取與線性插入同樣的加權係數wi，即

wi = (t - t0) / (t1 - t0)

本方法不同於線性插入法的地方在於我們會找出每個pixel「由哪來」，也就是順著Vx,y往回推，找到A在前一張原始frame上對應點的位置(xb, yb)，

(xb, yb) = (x, y) - wi•Vx,y

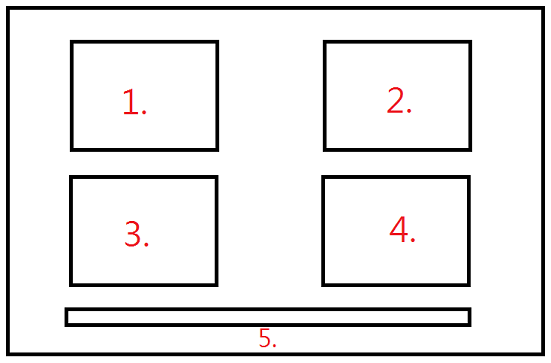
利用同樣的方法我們可以計算A點「會往哪去」，也就是順著Vx,y往後推，找到A在後一張原始frame上對應點的位置(xf, yf)

(xf, yf) = (x, y) + (1-wi)•Vx,y

最後我們利用加權比例再計算、組合成新的第i個frame上位於(x, y)位置的pixel的值，即

Fi,x,y = (1-wi)F0,xb,yb + wi•F1,xf,yf

以下我們介紹本軟體的使用者介面。本軟體的使用者介面如下圖所示,有四個影片撥放視窗。另外,下方有提供使用者控制的工具列。使用者可以透過工具列選擇觀看正常速度的撥放或慢速的撥放, 使用者也可以選擇同步觀看、並比較自己的揮拍動作與一位專業選手的揮拍動作。



1：播放標準動作(正面視角)

2：播放學員動作(正面視角)

3：播放標準動作(側身視角)

4：播放學員動作(側身視角)

5：工具列包含播放鈕、暫停鈕、時間軸

另外,雖然我們只使用一台kinect攝影機，而且是從正面錄製影像，但我們可以透過座標轉換產生側身圖，好像我們在學員的側邊也架設一台虛擬錄影機一樣(如下圖所示)，進而產生側身骨架圖。使用者可透過工具列選擇在視窗3和視窗4中,觀看側邊虛擬攝影機所對應自己的影像及專業選手的影像。

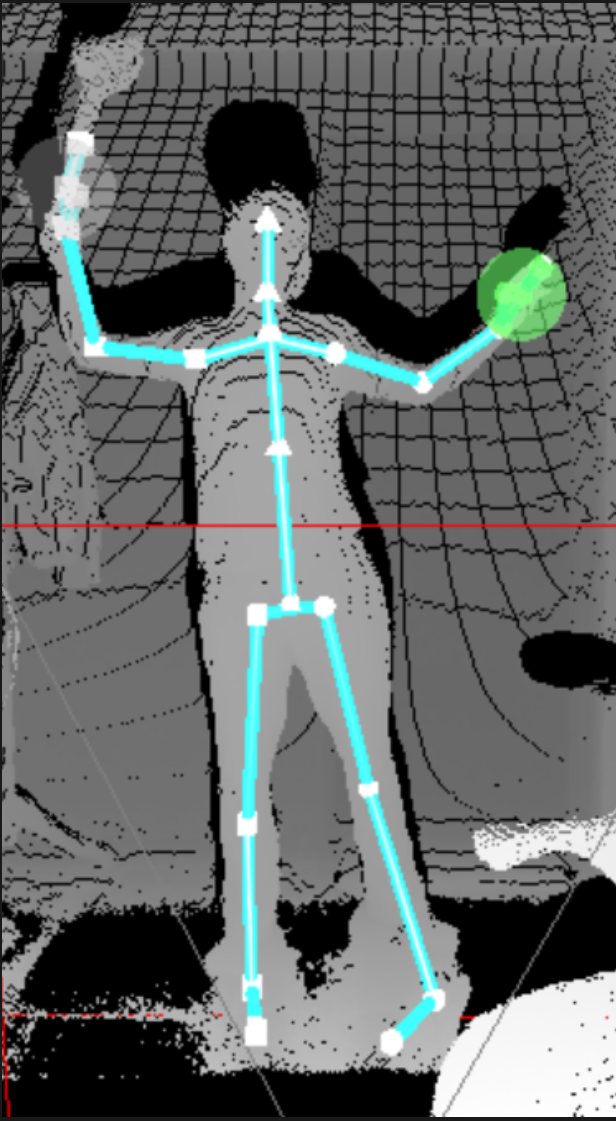
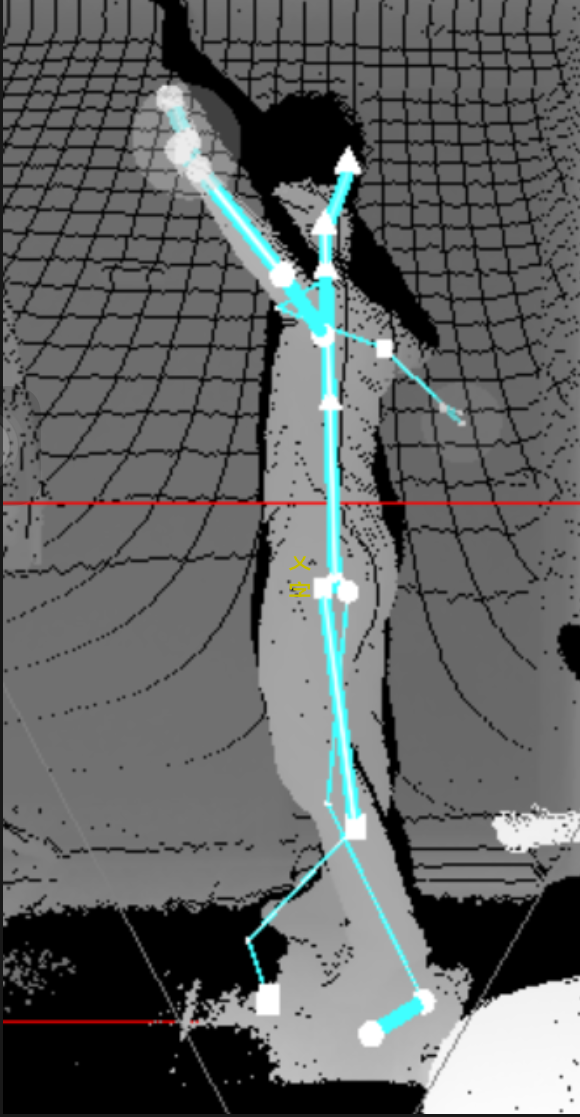
虛擬攝影機

實際攝影機

**3.3 評分**

想像在一個教學應用程式中，加入自動評分的功能，勢必能讓老師、學生省下許多心力，而這正是本計畫的核心技術之一。

本計畫希望藉由判斷學生的動作來給予適當的評分與建議，讓學員能夠及時了解自己的缺點並記錄自己的成長。本計畫實作的方式是預先錄好15位專業球員的標準姿勢作為多個樣本，抽出選手們**關鍵動作**的影像並分析出各選手姿勢的差距進而得到容許範圍，再將學員關鍵動作的影像抽出來比較，評斷出學員的姿勢是否正確。本計畫書以**殺球**的關鍵動作為例，做較詳細的解說。

關鍵動作1.手肘抬高與側身（準備擊球時的預備動作）：

(a) (b)

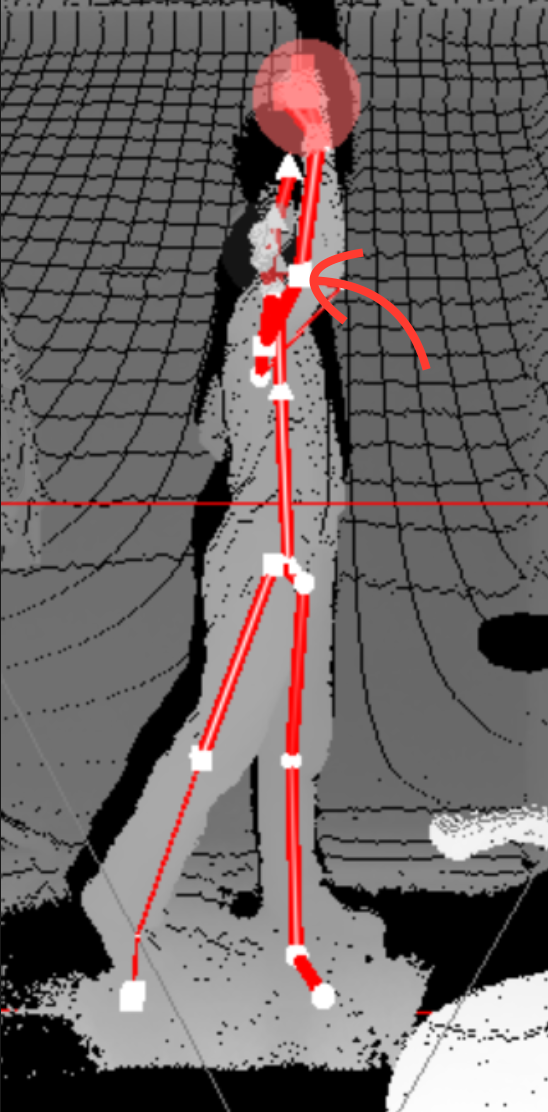
Figure 3.3 (a) 關鍵動作：側身，由正面拍攝。(b) 關鍵動作；手肘抬高，由側面拍攝。

側身：

分析骨架（脊椎、大腿）是否在前方攝影機的視角下(Figure 3.3 (b))呈現趨近一直線的形狀、或在側面攝影機的視角下是否呈現與前方視角看過去正常站立的骨架相似，來判斷是否有側身這個動作。

手肘抬高：

藉由Figure 3.3 (a) 分析手肘與水平面是否平行或夾一個容許範圍內的夾角（手肘高於水平或低於水平皆可）來判斷動作是否有將手肘先抬起來。

關鍵動作2.手肘轉向前（擊球前手肘應該向前延伸而不是緊縮）

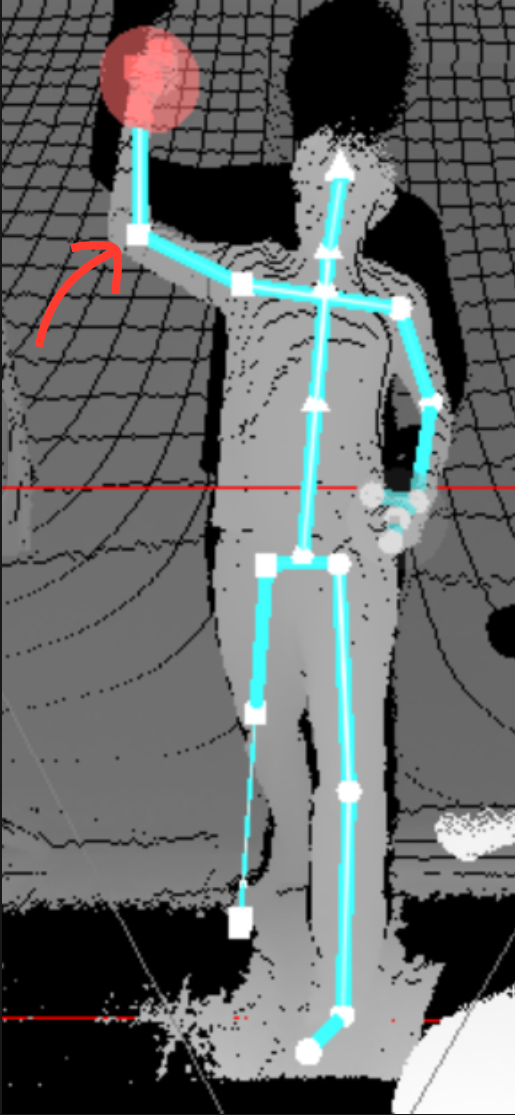
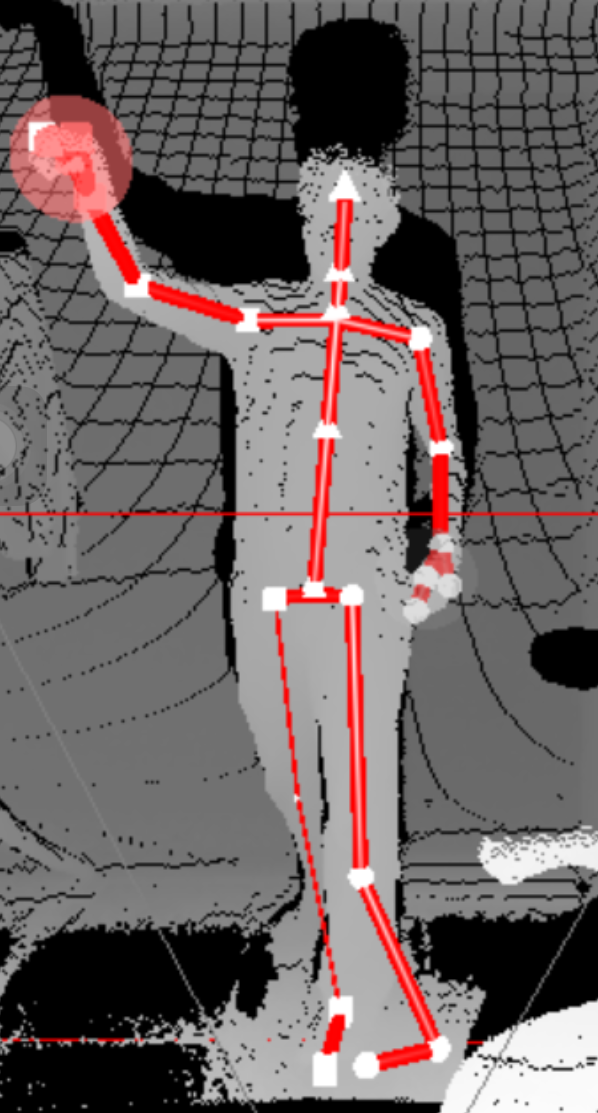
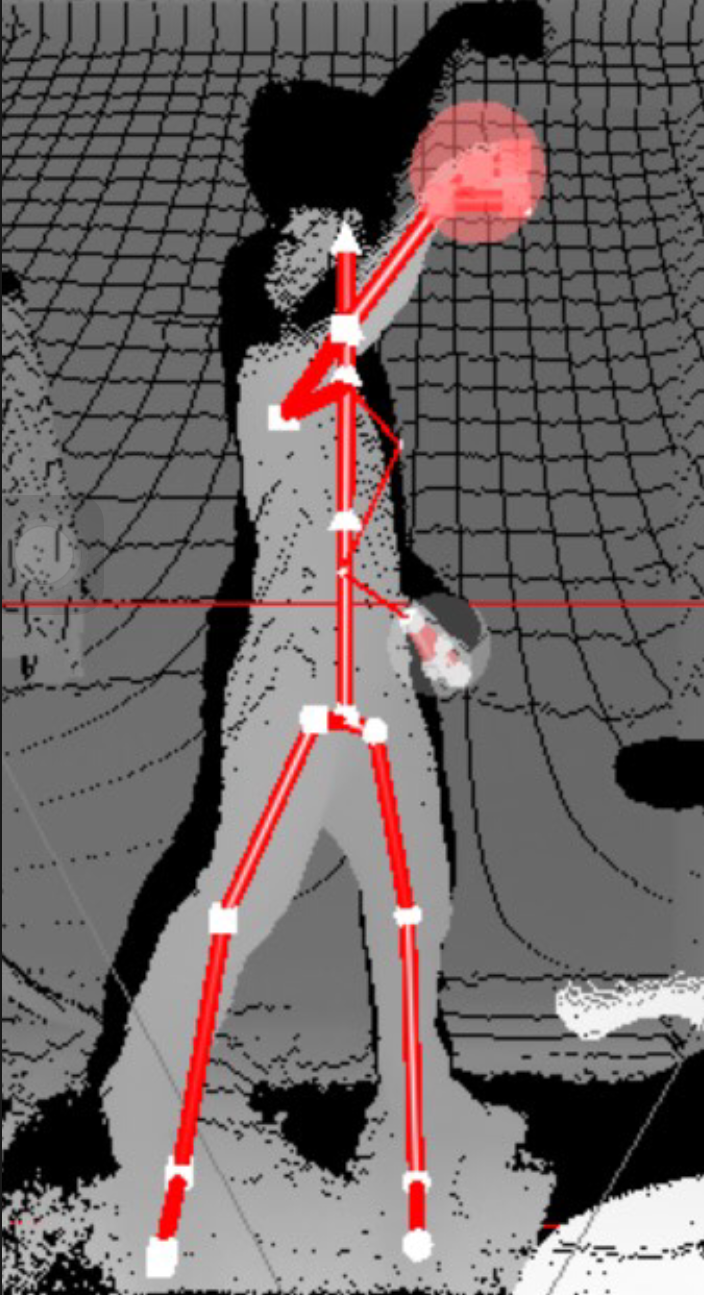
 (a) (b)

Figure3.4 關鍵動作：手肘轉向前 (a)由正面拍攝。 (b)由側面拍攝。

手肘是否轉向前的部分，主要是利用側面視角(Figure3.4(b)) 觀察出手肘的位移狀況，是否由脊椎的左方位移至脊椎的右方，也就是測試者及球的方向（平面上的左至右，及為測試者的後至前）。

關鍵動作3.手腕發力（擊球的瞬間）

(a) (b)

Figure3.5 關鍵動作：手腕發力 (a)由正面拍攝。 (b)由側面拍攝。

手腕是否發力是一個相當重要的評分項目。這個動作由側面拍攝(Figure3.5(b)) 較能明顯的觀察出手腕部分的變化。與前一個手肘抬高的截圖相比，小臂必須先向前延伸且手肘固定，手腕再做出瞬間向前的位移。所以判斷手腕發力這個關鍵動作，必須從上一個關鍵動作（手肘轉向前）停止後馬上開始取得連續的影像，藉由側面視角(Figure3.5(b))觀察手腕是否以手肘為圓心順時針旋轉，來判斷小臂（手腕—手肘）是否先向前延伸。要得知是否有做出手腕瞬間發力的動作，可以藉由在小臂（手腕—手肘）順時針旋轉的過程判斷手掌部分是否與小臂的夾角在短時間內有快速的變化，而非「手指—手腕—手肘」一路上都是趨近直線。

關鍵動作4.肩膀向前

肩膀向前是最後收拍的部分，可以藉由由前方拍攝的視角判斷右肩膀那個點在手腕發力後是否向左下延伸。

其餘還有一些下手的基本球路包括挑球、發球......等動作也會納入學生的學習目標內，發球的評分步驟如下：

發球：

1.重心角在右腳：左腳是否伸直（膝蓋夾角 = 180度）、右腳應為直立。

2.重心轉移到左腳：右腳是否些微彎曲（膝蓋夾角 < 180度）、左腳應為直立。

3.轉腰：原本為側身（前方視角：脊椎與大腿骨架近似直線），轉腰爲正身向前（側面視角：脊椎與大腿骨架近似直線）。

4.手腕發力：藉由在小臂（手腕—手肘）往上帶的過程判斷手掌部分是否與小臂的夾角在短時間內有快速的變化，而非「手指—手腕—手肘」一路上都是趨近直線。

5.肩膀轉向前：肩膀向前是最後收拍的部分，可以藉由由前方拍攝的視角判斷右肩膀那個點在手腕發力後是否向左上延伸。

**References**

[1] Loren Arthur Schwarz, Diana Mateus, and Nassir Navab, “Discriminative Human Full-Body Pose Estimation from Wearable Inertial Sensor Data,” 3DPH 2009, LNCS 5903, pp. 159–172, 2009.

[2] Hung-Kuo Chu, Che-Hua Yeh, Nick C. Tang, Hsiao-Rong Tyan, Yu-Shuen Wang, Hong-Yuan Mark Liao, “Motion Manipulation for Characters in Videos,”

[3] Che-Hua Yeh, Yi-Hsuan Yang, Ming-Hsu Chang and Hong-Yuan Mark Liao, “Music Driven Human Motion Manipulation for Characters in a Video,” 2014 IEEE International Symposium on Multimedia.

[4] Arif Budiman, Mohamad Ivan Fanany, “A Multiclass ELM Strategy in Pose-based 3D Human Motion Analysis,” ICACSIS 2013.

[5] ISO/IEC-13818-2, Generic Coding of Moving Pictures and Associated Audio, International Organization for Standardization, Jan. 1995, International Standard (MPEG-2 Video).

[6] F. Halsall, Multimedia Communications. New Delhi: Pearson Education, 2005.

[7] K.G. Jackson and G.B. Townsend, “TV and Video Engineer’s Reference Book,” Butterworth-Heinemann Ltd, 1991.

[8] http://www.zepp.com/baseball/zepp-sensor-specifications/

[9] <http://cjsb.org/paper/2003TSES/09.pdf>

[10] <http://forum.gamer.com.tw/Co.php?bsn=60001&sn=357966>

[11] <http://alex-phd.blogspot.tw/2014/03/optical-flowopencv.html>

[11] N. Paragios, Y. Chen and O. Faugeras, “Mathematical Models in Computer Vision: The Handbook”, p. 239-258, Springer, London, 2005.

[12] <http://monochrome.sutic.nu/2010/10/12/motion-interpolation.html>

[13] Ken-Ichiro Tsutsui, Hideo Sakata, Tomoka Naganuma and Masato Taira, “Neural Correlation for Perception of 3D Surface Orientation from Texture Gradient”, Science, 11, Volume 298, No. 5592, pages 409-412, October 2002.

[14] Feng Han and Song-Chun Zhu, “Bayesian Reconstruction of 3D Shapes and Scenes From A Single Image”, the First IEEE International Workshop on Higher-Level Knowledge in 3D Modeling and Motion Analysis (HLK’03), 2003.

[15] Volker Blanz and Thomas Vetter, “A morphable model for the synthesis of 3D faces”, ACM SIGGRAPH 1999, pages 187-194.