用動態時間校正演算法於冰凍肩患者復健之研究
Using Dynamic Time Warping to Help Rehabilitation for Patients with Frozen Shoulder

侯曉婷, 蔡長均, 李炯三, 江蔚文
Hsiao-Ting Hou,Tsai Chang-Chun,Chiung-San Lee, Jiang, Wey-Wen
國立臺北護理健康大學
Email: kenny61017@gmai.com

摘　要

在近幾年來隨著現代的科技進步與人們生活型態模式的改變，在長期工作中無形產生壓力與生活步調緊湊繁忙，久而久之容易造成慢性病症狀的累積與發生，接連可能導致相關後遺症與併發症，像是常見的疾病，如：糖尿病、關節疼痛、肩頸痠痛、冰凍肩症狀等，經由長期的壓力累積而導致這些相關疾病，而冰凍肩症狀為門診復健中最為常見的疾病項目之一，冰凍肩患者年齡層大多介於50~60 歲之間的族群，近來發現患有冰凍肩症狀的患者有逐年下降的趨勢，在 30 歲以下男女就醫的比例增加至 2~3 成，如此多的人需要透過復健醫學治療幫助，但因一般復健時需長時間且必須定期至醫院或診所進行治療復健，居住在較為偏遠地區的人口，則因地域關係較難全程配合，而使得患者在時間上較無法準時的控制，還需要在治療時度過程冗長且乏味的時間，導致冰凍肩患者長時間治療復健成效不佳，而時間久了病患最終選擇放棄治療，因此本文使用 Kinect 偵測骨架來算出人體身上的關節之間的角度和相對座標位置特徵值，再透過動態時間校正演算法(Dynamic Time Warping, DTW)來比對所做的各項復健動作與姿勢是否有到位。實作系統結合 Kinect、Processing、SimpleOpenNI 建立一套輔助冰凍肩患者用來進行復健之系統，針對肩關節運動(前屈、外展、內收、外旋、後伸及水平前屈)等設計姿勢判斷的規則，可以讓更多的冰凍肩患者不用在特定的醫院或診所之類的相關地點就診，且在任何時間、居家，均能復健。

本文修改動態時間校正演算法以適合於復健系統中，此改良式 DTW 演算法能有效縮短計算次數，結果顯示兩種演算法所產生的最短路近是非常相近，經由改良式方法的 DTW 統計出計算次數，比原 DTW 演算法大幅度的減少，相對於原 DTW 演算法的總計次數僅佔 2.37%。

關鍵詞：Kinect、復健動作偵測、動態時間校正、Processing、SimpleOpenNI。

一、緒論

依財團法人全民健康基金會指出，日本國家於西元 1996 年，已正式將「慢性病」更名為「生活習慣病」[1]。多數的慢性疾病皆由日常生活中的生活習慣及飲食習慣而得來，包含心臟病、高血壓、糖尿病及關節炎等慢性病，皆由長期間的習慣累積而成[2]。在日常生活中也因姿勢不良而產生的後遺症，和慢性病引起許多的併發症，例如：關節疼痛、肩頸痠痛及冰凍肩等症狀等等症狀，多數的症狀都必須要透過復健來改善。肩關節疼痛相關症狀為復健科中為多數，其中以「冰凍肩」症狀門診最為常見。

隨著現代的科技發展，近年來也被運用在醫療復建相關上整合科技的運用，如穿戴式與非穿戴式裝置透過攝影機感測器結合虛擬實境等等技術，來針對不同的復建動作、姿勢，經由電腦輔助設計設計各種復建項目，改變傳統的復建流程，不須到達特定的醫院、診所，在居家中就可以透過此方式進行，除了減輕金錢上的花費負擔，也增加患者復建的便利性。

二、相關知識與文獻探討

2.1 冰凍肩症候群

冰凍肩（frozen shoulder）此症狀是屬於一種症侯群，又稱「五十肩」，在醫學上學名稱之為「沾黏性關節囊炎」[3]，主要以肩關節劇烈疼痛及肩盂肱骨關節活動的角度及方向受到限制[4,5]，對於冰凍肩患者的影響包含拿物品、穿衣服、吃飯、洗澡，在生活中許多的不便及麻煩。

冰凍肩最主要的症狀就是及肩關節活動程度受到影響以疼痛，在醫學臨床裡將冰凍肩症狀分為三個期間階段，分別為第一期「疼痛期」、第二期「冰凍期」與第三期「恢復期」，疼痛期為肩部部分開始漸進似地產生疼痛，疼痛因素為關節囊的體積減少，關節活動程度目前屬於正常狀態，此階段持續 10 週至 36 週時間不等；冰凍期為肩部位置疼痛持續，肩部關節變得僵硬且活動程度降低，將嚴重影響到病患之日常生活，此階段持續 4 至 12 個月；恢復期肩部疼痛感逐漸減少，關節活動範圍也漸進地增加及恢復，但完全康復的機率很低，此階段持續 5 至 26 個月。

對於冰凍肩症狀的復健動作，針對肩部關節來調整活動的範圍，肩關節(shoulder joint)是杵臼關節，由肩胛骨與肱骨的圓頭的淺窩所形成的，所以肩關節有著不同面向的運動姿勢，因此肩關節的活

動範圍相當廣，肩關節復健動作可分為以下幾種，如圖 1 所示:

(1)前屈(forward flexion)：

肱骨沿著身體正前方向舉高

(2)外展與內收(abduction and adduction)：

肱骨沿身體側向上舉高再側向下壓

(3)外旋(external rotation)：

係指由肱骨軸旋轉的動作，此動作與上臂的位置有關，內外轉各約90度，共旋轉180度的活動範圍

(4)後伸(backward extension)：

肱骨沿著身體正後方舉高

(5)水平前屈(horizontal flexion)：

手臂由身體側向水平面向前移動至身體正向

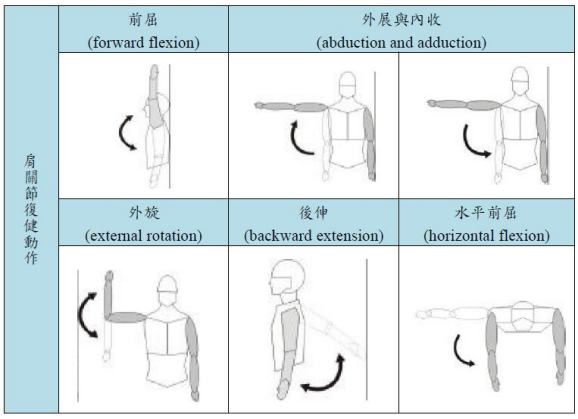

圖 1 肩關節復健動作[6,7]

2.2 Kinect 技術

Kinect 感應器能透過影像、骨架、聲音，做為一個人與電腦之間溝通的互動，分別為 RGB 攝影鏡頭、紅外線投影機以及紅外線攝影機，中間鏡頭為 RGB 彩色攝影機可擷取到彩色之影像，左側鏡頭為紅外線投影機可把特殊排列後的點投射到鏡頭前方的物體，來提供給右側鏡頭的紅外線 CMOS 攝影機來取得深度影像。如圖 2 所示:

圖 2 Kinect 感測器

而 Kinect 底部備有馬達可供感測器鏡頭垂直擺動，透過不同視角來觀測，且在 Kinect 底部感測器還建置四個陣列式麥克風提供聲音辨識，本論文 Kinect 主要以骨架追蹤方式來擷取特徵點取得到 15 個人體骨架關節點位置，包含頭部(head)、頸部(neck)、肩部(shoulder)、肘部(elbow)、手部(hand)、中央軀幹(torso)、髖部(hip)、膝部(knee)及腳部(feet)等 15 個關節座標位置資訊。

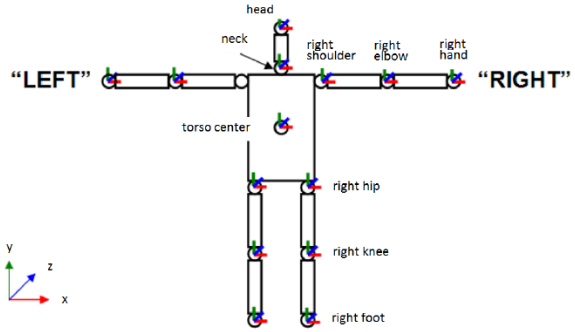

圖 3 擷取的 15 個關節資訊

2.3 動態時間校正演算法

動態時間校正(dynamic time warping，簡稱 DTW) 演算法是屬於動態規劃(dynamic programming)的方法，以比對序列的排序參考值與輸入值為主要資訊，其 DTW 演算法的特點在於能夠比較兩個長度不同的序列，有效解決辨識上時間扭曲的偏差，並且計算兩個序列之間的距離來判斷動作的相似程度，尋找點跟點與向量之間差距，透過歐基里得距離(euclidean distance)公式來作計算，假設有兩個向量分別為 U 和 V，長度分別為 m 和 n，則透過 DTW 演算法運算後，即找出一組路徑(X1, Y1), (X2, Y2), ..., (Xk, Yk)，經由路徑點對點的對應找出累積距離為最小之數值，並且此路徑需符合以下條件，如圖 3 所示:

(1)限制端點關係：(X1, Y1) = (1, 1), (Xk, Yk) = (m, n)，此端點關係則表示須頭對頭、尾對尾的進行比對

(2)局部距離關係：假設點和點之間的距離為 d(i,j)，距離值為｜U(Xi) – V (Yi)｜

(3)累積距離關係：假設最佳路徑上任一點可以表示為D(i, j)，那麼其前一點路徑只有三種可能：D(i-1, j), D(i, j-1), D(i-1, j-1)。此距離關係定義路徑的連續性

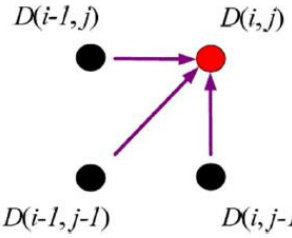

圖4 DTW演算法計算連續路徑[8,9]

以下為動態時間校正法之計算：

(1)定義D(i, j)為U與V之間的距離

(2)利用遞迴找出向量U與V之間最小累積距離值，D(i,j)為原點(1,1)到(i,j)的最小累積距離值，d(i,j)為點和點之間的距離之局部距離，其遞迴關係計算如公式1所示：

$$D(i,j) = \min\_of \begin{cases} D(i-1,j) + d(i,j) \\ D(i-1,j-1) + d(i,j) \\ D(i,j-1) + d(i,j) \end{cases}$$

DTW 演算法是先將兩序列中的所有兩點之間的 d(i,j)局部距離值圖 5 全部依依計算出來，再計算所有點跟點之間的 D(i,j)累積距離值，假設計算 D(1,1)之累積距離值圖 6，其值是由 D(0,1)、D(0,0)及 D(1,0)選擇其中最小值的 D(0,0)其數值為 2，再加上局部距離值 d(1,1)其數值為 1，累加後其 D(1,1)之累積距離值為 3。最後再由終點 D(11,12)開始往回推算至 D(0,0)尋找最短路徑值。

	0	1	2	3	4	5	6	7	8	9	10	11
Data	25	25	26	28	27	29	30	34	37	36	37	40
0	23	2	2	3	5	4	6	7	11	14	13	17
1	24	1	1	2	4	3	5	6	10	13	12	13	16
2	25	0	0	1	3	2	4	5	9	12	11	12	15
3	26	1	1	0	2	1	3	4	8	11	10	11	14
4	28	3	3	2	0	1	1	2	6	9	8	9	12
5	29	4	4	3	1	2	0	1	5	8	7	8	11
6	30	5	5	4	2	3	1	0	4	7	6	7	10
7	35	10	10	9	7	8	6	5	1	2	1	2	5
8	37	12	12	11	9	10	8	7	3	0	1	0	3
9	38	13	13	12	10	11	9	8	4	1	2	1	2
10	39	14	14	13	11	12	10	9	5	2	3	2	1

圖5 局部距離值

	0	1	2	3	4	5	6	7	8	9	10	11
Data	25	25	26	28	27	29	30	34	37	36	37	40
0	23	2	4	7	12	16	22	29	40	54	67	81	98
1	24	3	3	5	9	12	17	23	33	46	58	71	87
2	25	3	3	4	7	9	13	18	27	39	50	62	77
3	26	4	4	3	5	6	9	13	21	32	42	53	67
4	28	7	7	5	3	4	5	7	13	22	30	39	51
5	29	11	11	8	4	5	4	5	10	18	25	33	44
6	30	16	16	12	6	7	5	4	8	15	21	28	38
7	35	26	26	21	13	14	11	9	5	7	8	10	15
8	37	38	38	32	22	23	19	16	8	5	6	6	9
9	38	51	51	44	32	33	28	24	12	6	7	7	8
10	39	65	65	57	43	44	38	33	17	8	9	9	8

圖6 累積距離值

	0	1	2	3	4	5	6	7	8	9	10	11
Data	25	25	26	28	27	29	30	34	37	36	37	40
0	23	2	4	7	12	16	22	29	40	54	67	81	98
1	24	3	3	5	9	12	17	23	33	46	58	71	87
2	25	3	3	4	7	9	13	18	27	39	50	62	77
3	26	4	4	3	5	6	9	13	21	32	42	53	67
4	28	7	7	5	3	4	5	7	13	22	30	39	51
5	29	11	11	8	4	5	4	5	10	18	25	33	44
6	30	16	16	12	6	7	5	4	8	15	21	28	38
7	35	26	26	21	13	14	11	9	5	7	8	10	15
8	37	38	38	32	22	23	19	16	8	5	6	6	9
9	38	51	51	44	32	33	28	24	12	6	7	7	8
10	39	65	65	57	43	44	38	33	17	8	9	9	8

圖7 由後往回推尋找方式之最短路徑

原DTW演算法在尋找最短路徑之前，已將每一個點對點的局部距離值與累積距離值完整計算出來，所以能夠由最終點往回推算至初始端點來尋找最短路徑值如圖7，本論文所提出以即時方式來計算動態時間校正演算法之最短路徑，必須由最初數值開始計算，所以將原DTW演算法尋找最短路徑的方式改為由端點D(0,0)開始尋找最短路徑值至最終點D(11,12) ，其結果顯示兩種尋找最短路徑值之方法所產生的最短路徑值皆大致相同，所以本論文後續研究將會針對以即時尋找最短路徑的方式來作為比較依據。

	0	1	2	3	4	5	6	7	8	9	10	11
Data	25	25	26	28	27	29	30	34	37	36	37	40
0	23	2	4	7	12	16	22	29	40	54	67	81	98
1	24	3	3	5	9	12	17	23	33	46	58	71	87
2	25	3	3	4	7	9	13	18	27	39	50	62	77
3	26	4	4	3	5	6	9	13	21	32	42	53	67
4	28	7	7	5	3	4	5	7	13	22	30	39	51
5	29	11	11	8	4	5	4	5	10	18	25	33	44
6	30	16	16	12	6	7	5	4	8	15	21	28	38
7	35	26	26	21	13	14	11	9	5	7	8	10	15
8	37	38	38	32	22	23	19	16	8	5	6	6	9
9	38	51	51	44	32	33	28	24	12	6	7	7	8
10	39	65	65	57	43	44	38	33	17	8	9	9	8

圖8 由前往後推尋找方式之最短路徑

此方法為動態時間校正演算法之基礎，雖然最短路徑值是最準確的，但將可看出在計算兩序列之間的距離值時，發現實際上在尋找最短路徑有許多不需計算到的數值，若兩序列資料量增加，相對在計算上將會多出許多在計算不必要之距離值的時間，本論文根據此點來改進動態時間校正演算法之尋找最短路徑方法，將解決多於計算距離值的時間，以增加演算法之運算能力。

三、研究方法

3.1關節位置擷取

基於前述所擷取出的深度影像且偵測到復健者後，接下來就要針對復健者進行骨架追蹤(skeletal tracking)之階段，骨架追蹤是由Kinect來辨識復健者人體身上的特徵點並進行後續動作追蹤的程序，直接透過OpenNI追蹤每位復健者身上的各部位關節之位置，一旦偵測到使用者時，將會根據Kinect之人體骨架資訊，如圖11所示，擷取出頭部(head)、頸部(neck)、肩部(shoulder)、肘部(elbow)、手部(hand)、中央軀幹(torso)、髖部(hip)、膝部(knee)及腳部(feet)等15個關節座標位置資訊。如圖9所示

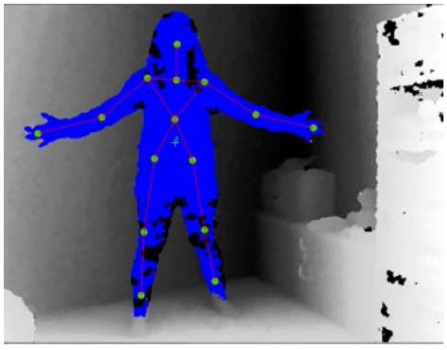

圖 9 完整關節骨架資訊

回傳個關節點在空間裡的三維座標(x,y,z)如圖10以從Kinect感測器為中心透過鏡頭看過去，x軸向左為正，向右為負；y軸向上為正，向下為負；z軸則是越靠近Kinect感測器值就越小，距離越遠值就越大。本研究需將關節位置呈現於二維之影像上，亦即是要將各部位關節位置在深度影像中顯示於相對應的身體部位，為了要達到此目的須將實際環境之座標值轉換成螢幕之座標值，即可與二維之

影像相對應之二維座標(x,y,z)

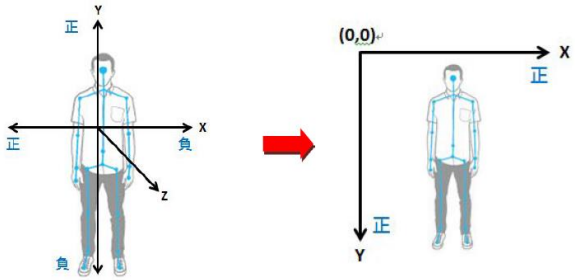

圖10 三維座標轉換二維座標系統[10]

3.2 改良式動態時間校正演算法

針對前一章節所敘述的原DTW演算法之運算過程，必須花費到大量的運算時間來計算不必要的比對數值，本章節將提出動態時間演算法之計算距離值與尋找最短路徑之改進方法，重新限制演算法比對的運算過程，來提升演算法的運算速度。改良式動態時間校正演算法分為三個階段，首先輸入資料為標準動作數據及復健者動作數據，兩筆輸入數據皆為一維序列，序列長度分別為n與m，則兩筆輸入數據中的每個Xi與Yj之特徵向量包含肩部角度及肘部角度。接下來是取得距離值，計算出局部距離再累加出最小距離值，根據計算出的距離值尋找出最短路徑。如圖11所示

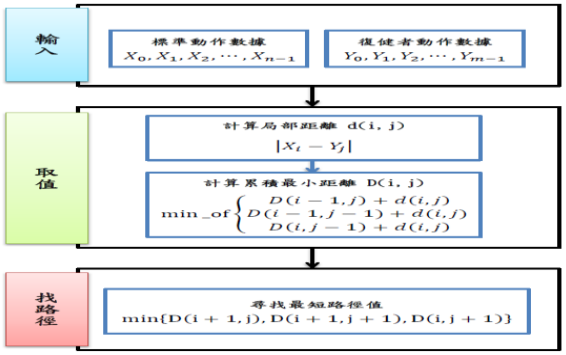

圖 11 改良式動態時間校正演算法之路徑

根據原DTW演算法，必須先將所有數據中的每一個局部距離值與累積距離值都需要計算出來，才能夠再進行接下來的尋找最短路徑階段。本章節所提出了適合本研究之復健動作比對的改良式動態時間校正演算法，針對尋找最短路徑的大量運算過程加以改進。

1. 為了減少不必要計算的數值，先將參考樣本與輸入樣本之兩序列間的累積距離值設定為最大值99999

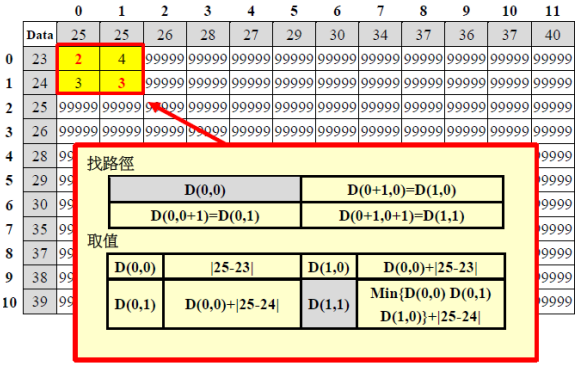

圖11 改良式DTW演算法之路徑

2. 由 D(0,0)開始尋找路徑，找到路徑值後再計算局部距離值與累積距離值，直到尋找至 D(n,m)為止，因計算過程相同將僅呈現此路徑的前兩段過程如圖 11，與最後兩段找路徑與取值的結果，如圖 12

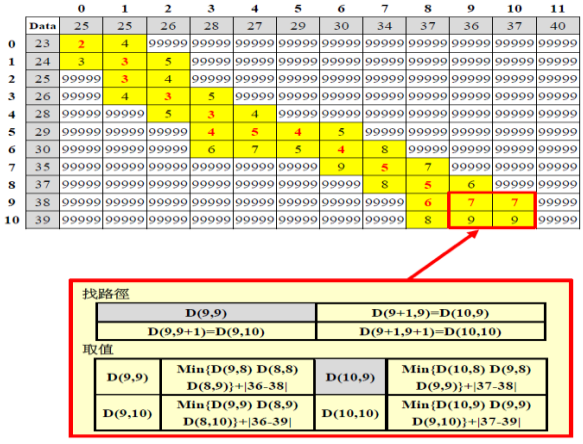

圖 12 原 DTW 演算法與改良式 DTW 演算法之比較結果

根據前章節所述，根據兩序列分別為序列{23,24,25,26,28,29,30,35,37,38,39}與序列{25,25,26,28,27,29,30,34,37,36,37,40}完成比較運算後，原 DTW 演算法所計算到的數值次數共為 132 次，而改良式動態時間校正演算法所計算到的數值次數僅為 36 次，並且顯示出兩個演算法所產生的最短路徑結果非常相似，因此透過改良式動態時間校正演算法明顯地能夠將計算量大幅的減少，且產生的最短路徑也能夠近似於原 DTW 演算法，將透過此改良式動態時間校正演算法運用至本論文復健系統之比對復健動作，以增加動作比較的運算速度。

3.3 復健系統

本論文實作之復健系統設計為五個階段，分別為初始階段、預備階段、復健階段、顯示結果與播放模式。以下為各階段介面顯示：

(1)初始階段：偵測復健者骨架。確認擷取到骨架後將顯示肩部角度及肘部角度。首先選擇復健者需進行復健的右手(Righthand)或左手(Lefthand)，如圖

33所示，再選擇需復健之動作類型分別為前屈(Forward)、外展與內收(Abduction)、外旋(External)、後伸(Backward)及水平前屈(Horizontal)

(2)預備階段：於初步階段選擇完成手部與復健動作類型後，在進行復健階段前的準備時間，給予復健者在復健前有個能夠緩衝的準備時間，此為防止若復健一開始復健者動作錯亂，導致系統誤判復健動作之數據。系統預設復健者準備之時間為10秒鐘，畫面將以倒數10~0秒鐘的方式來顯示，顯示於介面中的「Ready Time」，並且在預備階段將開始顯示出復健者之骨架，選擇右手復健動作，如圖13

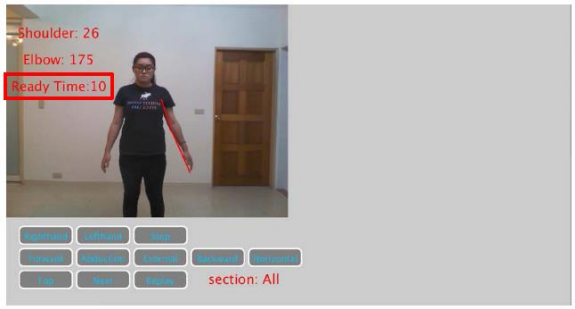

圖13選擇右手復健預備階段之介面

(3)復健階段：開始進行復健療程，每組復健療程皆設計為30秒，左側畫面將以「Run Time」顯示進行時間1~30秒鐘，右側畫面為標準動作示範畫面，讓復健者能夠跟著標準動作進行復健。由於本系統設計為讓復健者跟自己前一次的復健動作作比較，所以在進行第一次復健動作時畫面將會先顯示為「First Time」以提醒復健者，如圖14

圖14 復健階段之介面

(4)顯示結果：當復健者在進行第二次的復健動作後，畫面將會顯示出與前一次復健動作比較後之結果，若復健結果比前一次動作好的情況下，畫面將顯示為「Improvement!!」，若復健者動作與原先標準動作差異太大或者錯誤動作的情況下，畫面將顯示為「Action Error!!」如圖15、16所示

圖15 復健顯示進步結果之介面

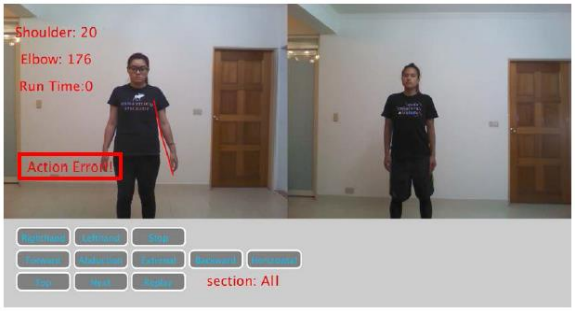

圖16 復健顯示動作錯誤結果之介面

(5)播放模式：當復健者或醫護人員需追蹤復健者的復健情形，復健者為連續進行復健療程，將可點選「Top」及「Next」需要觀看的第幾次復健療程記錄，或者選擇觀看全部復健療程畫面，選擇完成後點選「Replay」將可重覆播放復健過程，如圖17所示

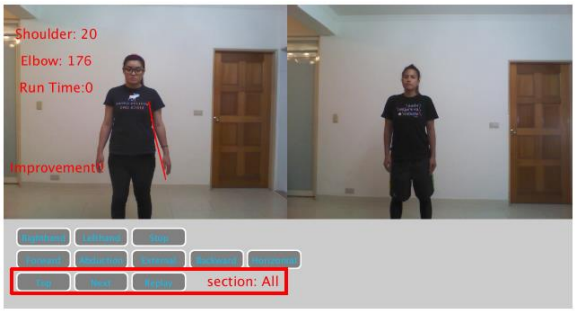

圖17 播放模式之介面

四、研究結果討論

4.1 復健動作之標準閥值

針對所設計的前屈、外展與內收、外旋、後伸及水平前屈的五個復健動作之標準動作閥值，設立此閥值是為了防止復健者在進行復健療程時，做出與標準動作完全不相同的動作，亦或做出錯誤動作還無法得知自己的復健動作已有所偏差。此閥值能夠以防復健者在錯誤的動作情況下所產生的數據，一直持續與自己錯誤的數據進行比較。如圖18所示

Data	外展內收	前屈	外旋	後伸	水平前屈
1	1054.67	2062.67	1086.50	297.83	2606.83
2	513.00	702.67	1057.80	668.83	2376.67
3	1192.67	1318.50	917.33	597.33	1538.67
4	743.17	1738.00	1353.67	796.83	2321.00
5	715.67	1847.83	1771.00	677.33	2153.67
6	1176.50	923.50	1129.50	911.80	1662.17
7	1045.33	1635.83	664.67	538.67	2259.33
8	1319.33	1553.83	2040.00	596.00	1869.00
9	892.50	1828.17	1330.83	683.17	2377.50
10	752.50	617.83	1729.83	634.50	1888.83
閥值	1400	2100	2000	1000	2700

圖18復健動作之標準閥值

本復健系統將根據改良式動態時間校正演算

法的計算結果，所建立出外展內收、前屈、外旋、後伸及水平前屈五個復健動作的標準閥值作為復健者動作判斷的初步依據。復健者在進行復健療程階段，有時候會因動作不熟悉亦或動作不準確導致做出錯誤的復健動作，當復健者做出錯誤的復健動作此數據結果將一定會超出所設立的標準閥值，系統將會告知復健者動作錯誤之訊息。設立此復健動作標準閥值即為了能夠告知復健者所做的復健動作是否在正確動作的前提下進行比較，以免復健者在復健療程中一直重複與自己錯誤復健動作的數據結果進行比較。

4.2 動態時間校正演算法最短路徑之實證結果

藉由Kinect技術所擷取到的復健者關節骨架，經由計算出的肩部角度與肘部角度作為演算法之數值依據，透過本論文針對冰凍肩患者所設計的五個復健動作外展內收、前屈、外旋、後伸及水平前屈。這五個復健動作的原DTW演算法與改良式DTW演算法之路徑比較，五個復健動作透過原DTW演算法之計算次數共皆為22500次，其經由改良式DTW演算法所統計出的五個復健動作所運用到的計算次數及相較於原DTW演算法比較下的總計算量。如圖19所示

Data	外展內收		前屈		外旋		後伸		水平前屈	
	計算次數	總計算量	計算次數	總計算量	計算次數	總計算量	計算次數	總計算量	計算次數	總計算量
1	564	3%	501	2%	537	2%	544	2%	535	2%
2	540	2%	530	2%	548	2%	562	2%	503	2%
3	558	2%	529	2%	548	2%	551	2%	531	2%
4	551	2%	492	2%	548	2%	540	2%	551	2%
5	546	2%	519	2%	557	2%	459	2%	561	2%
6	574	3%	538	2%	545	2%	356	2%	536	2%
7	537	2%	521	2%	538	2%	546	2%	546	2%
8	558	2%	556	2%	558	2%	463	2%	554	2%
9	557	2%	516	2%	558	2%	552	2%	527	2%
10	561	2%	524	2%	550	2%	466	2%	547	2%

圖19改良式DTW演算法計算次數比較結果

五、結論與未來展望

本文主要以改進動態時間演算法來節省計算時間與電腦效能，透過動作以骨架偵測的方式計算肩部關節角度與肘部關節角度，針對外展內收、前屈、外旋、後伸及水平前屈五個復健動作，建立出個動作之定點姿勢，定義每組復健動作的各關節點相對位置之判斷規則，利用肩部關節、肘部關節及手部關節座標值的相對位置來顯示骨架顏色，讓復健者能夠即時的調整復健動作，藉此也可提升復健動作之準確度。

在復健療程進行比較計算出的肩部關節角度與肘部關節角度之數據，進行動態時間校正演算法之運算，將所擷取出的復健動作串成一個連續的動作序列，提出改良式動態時間校正演算法之運算流程，解決了原DTW演算法必須花費大量運算時間之問題，排除計算多餘不必要的比對數值之運算，並且大幅減少計算次數，建立出尋找最短路徑之改良式DTW演算法，原DTW演算法與改良式DTW演算法之最短路徑非常相似，已重新限制演算法比對的運算過程，並且提升演算法的運算速度。

透過改良式動態時間校正演算法所產生的結果，建立出外展內收、前屈、外旋、後伸及水平前屈五個復健動作之標準動作閥值，復健者可根據此標準閥值得知復健動作是否正確，能夠在正確動作的情況下與自己的復健數據作比較，並且記錄下復健過程之數據，將復健者在復健過程中的動作記錄下來，提供醫護人員追蹤復健者的動作及姿勢，可透過重複播放復健動作之模式，回顧復健者的復健情形，以作為醫護人員追蹤及調查之參考依

六、參考文獻

[1] 財團法人全民健康基金會，2013，破解生活習慣病，Available: http://www.twhealth.org.tw/index.php?option=com_zoo&task=item&item_id=305&Itemid=19

[2] C. Chun-Ming, C. Yen-Ching, and H. Bing-Yu, "The design of a shoulder rehabilitation game system," in Frontier Computing. Theory, Technologies and Applications, 2010 IET International Conference on, 2010, pp. 151-156.

[3] 財團法人全民健康基金會，2013，破解生活習慣病，Available: http://www.twhealth.org.tw/index.php?option=com_zoo&task=item&item_id=305&Itemid=19

[4] 黃建銘、黃巧雯、王昱菱、宋姿穎，2008，『冰凍肩的診斷及治療』，基層醫學，第23卷，132~136頁

[5] 李君碩，2007，『物理治療對於冰凍肩的療效』，北市醫學雜誌，第4卷，787~799頁

[6] V. H. Frankel and M. Nordin, Basic biomechanics of the musculoskeletal system. Philadelphia, Penn.: Lippincott Williams & Wilkins, 2001.

[7] 張若菡，2002，應用肩部動作形態分析於電腦使用者肩部保健產品設計，國立成功大學工業設計學系碩博士班碩士論文

[8] 吳俊龍，2008，一個計算動態空間扭曲演算法之VLSI架構，國立臺灣海洋大學資訊工程學系碩士論文。

[9] 張恆誌，2011，使用動態時間校正演算法於國語數字語者辨識系統之研究，義守大學電子工程學系碩士班碩士論文。

[10]王森，2013，Kinect體感程式設計入門與應用(第二版)，台北市: 基峰資訊股份有限公司