

國立交通大學

多媒體工程研究所

碩士論文

肢體動作所產生肌肉形變之合成

**The Synthesis of Muscle Deformation Under Articulated  
Motions**



研究生：林逸晨

指導教授：施仁忠 教授

中華民國九十八年六月

肢體動作所產生肌肉形變之合成

The Synthesis of Muscle Deformation Under Articulated Motions

研究生：林逸晨

Student：I-Chen Lin

指導教授：施仁忠

Advisor：Zen-Chung Shih

國立交通大學

多媒體工程研究所



Submitted to Institute of Multimedia Engineering

College of Computer Science

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

June 2009

Hsinchu, Taiwan, Republic of China

中華民國九十八年六月

# 肢體動作所產生肌肉形變之合成

## The Synthesis of Muscle Deformation Under Articulated Motions

研究生:林逸晨

指導教授:施仁忠教授

國立交通大學多媒體工程研究所



本論文主要針對因肢體動作而產生的肌肉形變，提出一個新的肌肉形變合成方法。先利用骨骼子空間的形變 (Skeletal Subspace Deformation)，來控制整體人體的形變；接下來再針對身體上特定部位，作進一步地細緻化修改。當人體手臂的上臂以及下臂間的夾角越來越小時，此時肌肉會收縮，更進一步的二頭肌會很明顯的產生出肌肉膨脹的效果，因此我們根據上下臂間的夾角，以高斯函數 (Gaussian Function) 來合成不同程度肌肉膨脹的效果。本方法與骨骼子空間的形變相比，在些微的速度損失下，提升了肌肉形變的程度，產生更加逼真的效果。

# The Synthesis of Muscle Deformation Under Articulated Motions

Student: I-Chen Lin

Advisor: Dr. Zen-Chung Shih



Institute of Multimedia Engineering  
National Chiao Tung University

## ABSTRACT

This thesis proposes a new method of the deformation of body motion. Firstly, our work uses the Skeletal Subspace Deformation according to the skeleton to control the deformation of the human body. Secondly, we further modify the deformations at some specific muscles. When the angle of the upper arm and the lower arm reduce, the muscle will twist, and then the muscle bugle effect of the biceps will become more obvious. Consequently, we consider the angle of the upper arm and the lower arm, Gaussian function is applied to synthesis the effect of muscle bugles. Comparing to Skeletal Subspace Deformation, this method increases the degrees of the deformation of the muscle with slight loss of speed.

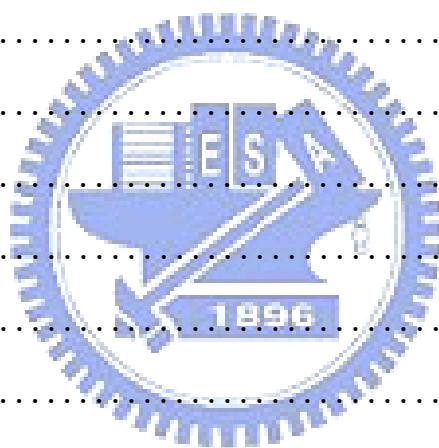
# 致謝

關於這篇論文，首先要感謝指導教授施仁忠老師，在他的幫助和指導之下，給了我很多的建議以及支持，才能順利的完成此篇論文著作。除此之外還要感謝幾個好朋友的幫忙。最後我要感謝我的家人，因為有你們的支持，我才能夠努力的完成論文。



# 目錄

摘要(中文).....	I
摘要(英文).....	II
致謝.....	III
目錄.....	IV
圖目錄.....	V
<b>第一章 緒論</b> .....	<b>1</b>
1.1 動機.....	1
1.2 系統概述.....	2
1.3 論文架構.....	4
<b>第二章 相關研究</b> .....	<b>5</b>
<b>第三章 演算法</b> .....	<b>9</b>
3.1 網格層次概述.....	9
3.2 骨骼子空間的形變.....	18
3.3 產生肌肉隆起的效果.....	26
<b>第四章 結果</b> .....	<b>33</b>
<b>第五章 結論與建議</b> .....	<b>41</b>
參考文獻.....	43

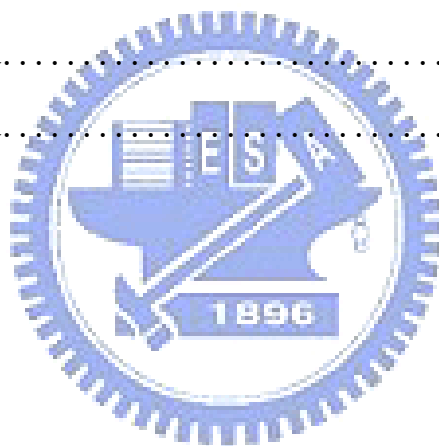


# 圖目錄

圖 1.1：系統架構圖.....	3
圖 2.1：由橢球體來模擬人物的四肢.....	6
圖 2.2：由橢球來模擬形變.....	6
圖 3.1：角色網格.....	10
圖 3.2：骨骼旋轉.....	12
圖 3.3：骨骼位移.....	12
圖 3.4：骨骼空間轉換.....	13
圖 3.5：手臂的骨架.....	14
圖 3.6：手臂旋轉(1).....	15
圖 3.7：手臂旋轉(2).....	16
圖 3.8：手臂旋轉(3).....	16
圖 3.9：手臂旋轉(4).....	17
圖 3.10：剛體動畫.....	20
圖 3.11：期望的結果：骨骼子空間的形變.....	21
圖 3.12：轉換過程.....	23
圖 3.13：二頭肌的位置.....	26
圖 3.14：肌肉隆起的方向.....	28
圖 3.15：皮膚網格頂點投影到平面.....	29
圖 3.16：二維高斯函數示意圖.....	29
圖 3.17：肌肉隆起的特性.....	30



圖 4.1: Woman 和 Skeleton 的預設姿勢.....	34
圖 4.2: Skeleton 與無肌肉隆起的形變.....	35
圖 4.3: 有肌肉隆起.....	35
圖 4.4: Skeleton 與無肌肉隆起的形變.....	36
圖 4.5: 有肌肉隆起.....	36
圖 4.6: Skeleton 與無肌肉隆起的形變.....	37
圖 4.7: 有肌肉隆起.....	37
圖 4.8: 小腿肌肉比較圖.....	38
圖 4.9: 其他形變結果.....	38
圖 4.10: 其他形變結果.....	39
圖 4.11: 其他形變結果.....	39
圖 4.12: 其他形變結果.....	40





# Chapter 1

## 緒論

### 1.1 研究動機



不論是動畫產業，或者是數位遊戲產業中，人物角色外表肌膚機制模擬 (Character skinning) 都扮演著相當重要的角色。首先，設計師利用繪圖工具建立角色的靜態模型 (Modeling)。然而在動畫以及遊戲中，該模型是一個會移動，並且改變姿勢的角色。因此當角色在姿勢改變的時候，需要讓模型上的頂點移動到適當的位置。根據人體的構造，角色姿勢的改變是源自於身體內骨架姿勢的改變，進一步牽動肌肉產生位置及形狀的改變 (Deformation)。因此，我們提出一個方法，可以讓模型經由骨架的變化驅動整體的形變。

此外在角色的形變產生之中，我們觀察到肌肉的變化，肌肉隆起 (Muscle Bugle) 的效果是最為明顯的部分。根據運動生理學[18]，肌肉隆起的原因是因為肌肉受刺激 (Twist) 收縮後所產生的物理現象。在人體中，上臂二頭肌以及小腿肌這兩部分是最容易觀察到這種現象的發生。

故在本論文中，我們在沒有適當肌肉形變範例的情況下，在角色模型產生形變之時，不僅讓肌膚上的頂點移動至適當的位置，並且在特定的部分，產生肌肉隆起的效果。

## 1.2 系統概述



本論文分為兩大部分：第一部分是角色模型整體肌肉的形變，這一部分是根據角色骨架改變，而影響皮膚頂點的位置，進而產生整體的變化。第二部分則專注於角色產生肌肉隆起的部分。我們將這一部分做另外的處理，利用高斯函數 (Gaussian Function) 以及骨骼之間夾角的大小，來達到不同程度肌肉隆起的效果。完成這兩個部分之後，我們便可以產生最後的結果，完成人物角色肌肉形變機制模擬之目的。

圖 1.1 展示了本系統的架構圖。在輸入角色的 3D 模型後，根據我們所要產

出的姿勢，首先我們先改變角色的骨架。骨架改變後，系統便依據新的骨架姿勢，利用骨骼子空間的形變 (Skeletal Subspace Deformation)，根據角色的骨架變化產生整體肌肉變形。接下來則針對特定的部分，利用高斯函數 (Gaussian Function)，以及骨骼之間的夾角大小，來模擬並且產生肌肉隆起的效果，這兩步驟完成之後，便可輸出最後的結果。

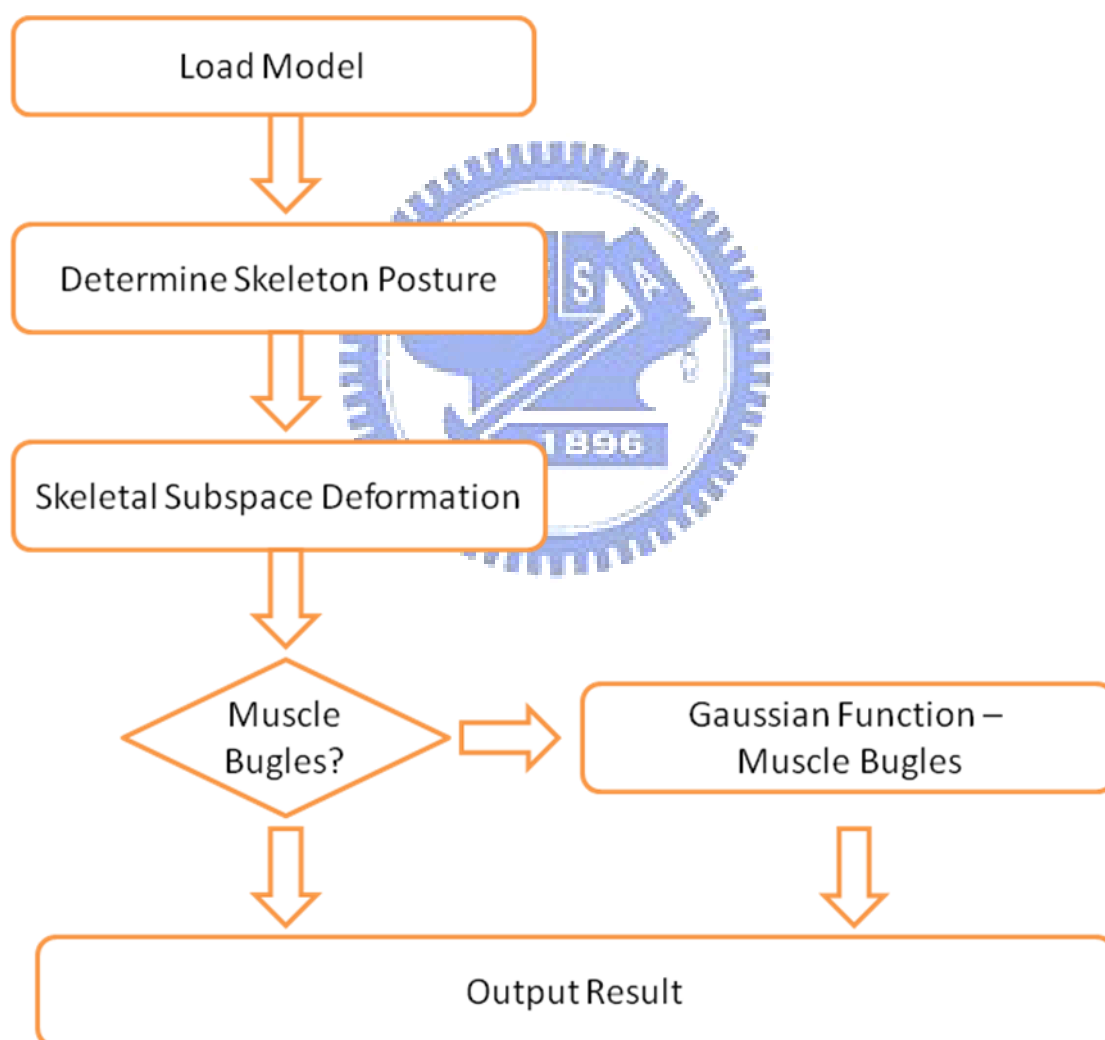


圖 1.1 系統架構圖

## 1.3 論文架構

本論文總共分為五個章節，第一章為緒論，描述我們的研究動機與目的。在第二章的部分，我們介紹相關文獻以及其方法概念。在第三章的部分我們說明骨架與肌肉的關係，以及形變之後的表皮機制處理，模擬肌肉隆起的效果的演算法。在第四章的部分會展示出本系統所產生出來的結果。最後在第五章的部分將會回顧本論文的方法及架構，並提出未來可能加強和改進的地方。



# Chapter 2

## 文獻回顧



本章將介紹與本論文方向及方法相關聯的研究文獻。

Dae-Eun Hyun et al. [7] 提出了利用橢球體來近似人體的四肢（手與腳），以達成四肢形變的結果。此方法對模型的四肢分別產生相對應的橢球體，伴隨骨架的改變，而進行四肢肌肉的形變（圖 2.1），然而，因為在橢球體的大小及位置改變之後，還需要利用對應表來查詢橢球體相對應的四肢部分，因此在速度上比較慢，無法達到即時的效果。而 Seung-Hyun Yoon [8] 利用相似的方法，不過將橢球體改變為橢圓，並且形變的範圍可以達到全身。此方法利用在關節點（Joint）上的關鍵橢圓（Key Ellipse），以及每一段骨骼上的三個橢圓來模擬肌肉（圖 2.2）。

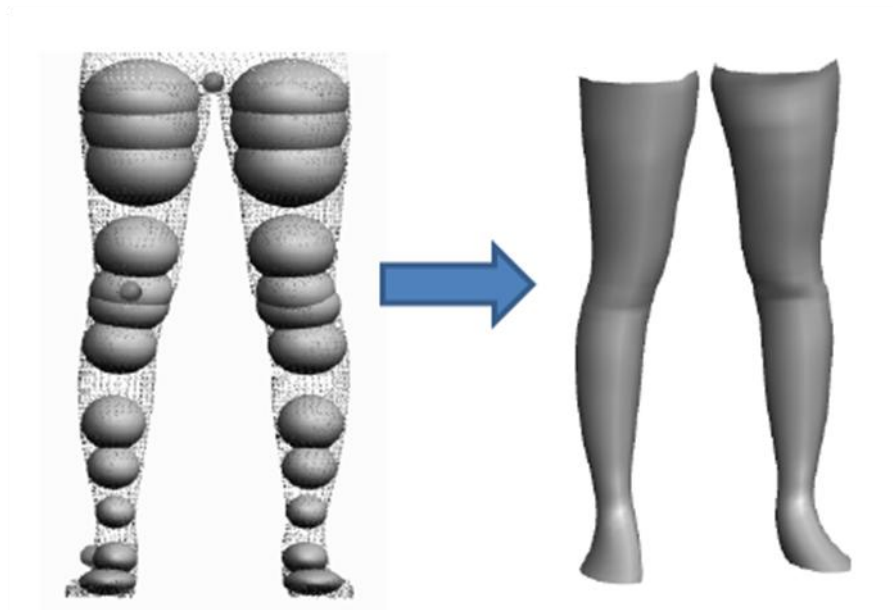


圖 2.1：由椭球體來模擬人物的四肢。

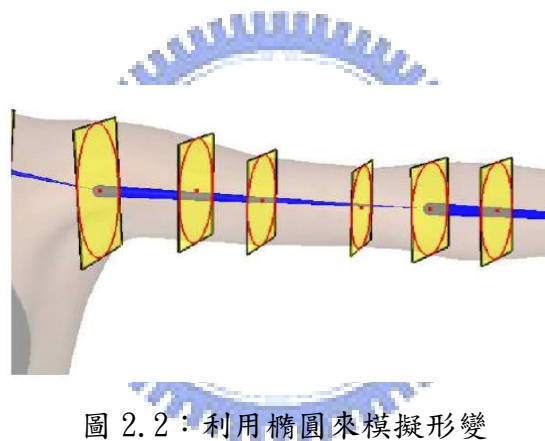


圖 2.2：利用橢圓來模擬形變

Tao Ju et al. [11] 以框架為基礎 (Case-Based) [9, 10, 13] 進行肌肉形變模擬。他們利用框架 (Cage) 來當作一個模型的概略描繪，每一個框架恰好包含了人體經由骨骼所隔開的一段人體表皮。經由骨骼的變動後，框架的位置跟著改變，伴隨的相對應角色表皮位置進而產生改變。本方法與上述三種方法的差異在於不需要產生額外的對應椭球體或橢圓，或者是模型的分段框架，而是在形變時，直接操作模型上的頂點，因此在速度的表現上，有比較大的優勢。

Li-Yi Wei et al. [6] 則是提出了經由梯度域 (Gradient Domain) 來進行肌肉形變模擬的方法。此方法使用梯度域的技術[14, 16, 23, 28, 29], 並且保持了角色的體積在形變後不發生改變, 並且保證了角色的四肢長度不會因為形變而發生被拉長的現象。

Shin Yoshizawa et al. [27]則結合了以骨骼為基礎的網格 (Skeleton-Based Mesh) 和離散微分座標 (Discrete Differential Coordinates) [1, 22]。此方法將人物模型分為兩個網格：骨架網格 (Skeletal Mesh) 和雙向的 Voronoi 基礎的近似網格。當骨架產生變化時, 再利用離散微分座標來重建模型形變後的形狀。



Peter Sand et al. [17] 則是使用連續擷取的方法。此方法擷取人體的輪廓, 經由輪廓產生可變的人體模型。動態方面則經由人體骨架的移動後, 再根據輪廓, 依序由每一個骨骼產生新的模型位置。而 Lewis et al. [12] 提出了以範例為基礎的形變方式。提供一個範例作為形變的基準, 利用 RBF (Radial Basis Functions) 內插出肌肉形變的模式。除此之外還有一些以範例為基礎的形變方法[15, 21, 25], 其中 Allen Brett et al. [2] 使用的方法較為不同。此方法所使用到的範例不只是僅僅一張圖片, 而是經由動作擷取 (Motion Capture) 所

得到的資料。而在上述三種方法中，皆使用到了形變範例，利用形變的範例來產生肌肉隆起的效果。但是通常形變範例與所使用的模型基本上都是相對應的。因此本方法希望在沒有取得適當的形變範例下，仍然可以模擬肌肉隆起的效果。

還有一些以物理為基礎的方法[3, 5, 26]。這部分的方法不只建構了皮膚模型，皮下組織的部分也作為考慮的因素，以物理為基礎的方式模擬真實的肌肉形變。上述的方法大體上都是以骨架為基礎的模型形變方式，此外還有一些不需要骨架來進行角色模型形變的方法，例如以空間為基礎的形變[ 19, 20 ]，還有以範例為基礎的表面形變技術[4, 24]。

總結上相關文獻，本系統希望，在利用骨架為形變基礎的情況下，首先進行腳色整體的肌肉形變，之後還希望對角色特定的部位，在不利用合適的形變範例之情況下，產生肌肉隆起的效果，並且能快速的產生人體形變。



# Chapter 3

## 演算法



在這個章節，我們將要講述整個系統的演算法。我們會將第三章分為三個部分，分別為：3.1 - 概述整體網格層次(Mesh Hierarchies)，3.2 - 骨骼子空間的形變 (Skeletal Subspace Deformation)，3.3 - 肌肉隆起的效果。

### 3.1 網格層次概述

圖 3.1 顯示了立體角色模型的網格。在這邊我們可以看到，位於網格的內部，有一段一段的箭頭，每一個箭頭代表骨骼 (Bone)，而將整個角色的骨骼串在一起，整串骨骼之為骨架。骨架是產生角色動畫最為重要，也是最為基礎的部分，因為整個角色的形變是由骨架的姿勢改變來驅動的。位於角色外部的網格則

稱為皮膚，皮膚代表整個立體角色的幾何形狀。

在骨架的姿勢改變的時候，骨骼會改變其位置，而骨骼位置的改變則會影響到皮膚上頂點位置，而產生形變。在數學上，我們將每塊骨骼用轉換矩陣 (Transformation Matrix) 來表示。透過轉換矩陣，我們可以將皮膚頂點的位置移動到發生形變後所該坐落的適當位置。簡單來說，當我們改變骨架的姿勢時，與骨架相對應的皮膚網格的頂點便會跟著變動。

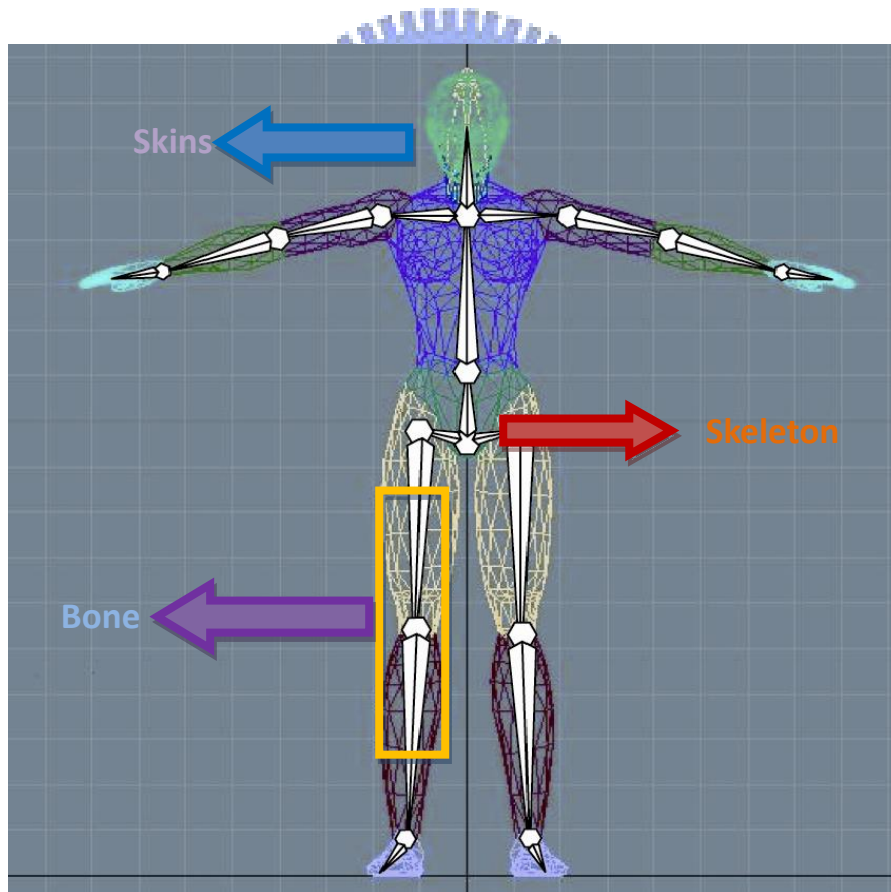


圖 3.1: 立體角色的網格。三個重要部分：表皮、骨骼、及骨架。

### 骨骼的階層架構

一開始，每個骨骼都分別在自己的空間內，此空間稱為骨骼空間 (Bone Space)，並且將其關節 (Joint) 放置於原點。每塊骨骼包含兩個變換矩陣，分別為：1) 局部的轉換矩陣 (Local Transformation Matrix)  $L$ ，以及 2) 複合轉換 (Combined Transformation Matrix)  $C$ 。在數學上座標轉換都以轉換矩陣來表示。其中轉換矩陣可以以  $4 \times 4$  的正交矩陣 (Homogeneous Transformation Matrix) 來表示：

$$T_j = \begin{bmatrix} T_j^{\text{rot}} & T_j^{\text{trans}} \\ \mathbf{0} & 1 \end{bmatrix}$$

$T_j$  代表的是一個轉換，

$T_j^{\text{rot}}$  所代表的是  $T_j$  的旋轉部分，是一個  $3 \times 3$  的旋轉矩陣，

$T_j^{\text{trans}}$  則代表了  $T_j$  的平移部分，是一個  $3 \times 1$  的向量。



由  $T_j$  可知，在局部的轉換矩陣的部分，包含了兩個資訊：一是旋轉，骨骼以本身的關節點為軸作旋轉，如圖 3.2 所示；另外一部分則是平移：這一部分則會根據父骨骼 (Parent Bone) 的位置來放置現在骨骼的位置，父骨骼代表再在整個骨架階層下，連接並且位於目前骨骼上一階層的骨骼。這麼做是為了要保留空間放置父骨骼，此部分如圖 3.3 所示。以下的圖皆以箭頭代表骨骼。

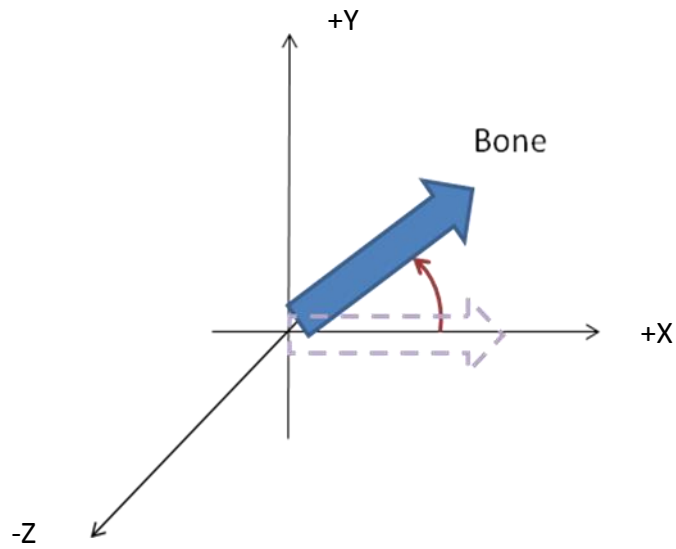


圖 3.2：骨骼根據本身的旋轉矩陣，以關節為軸來進行旋轉。

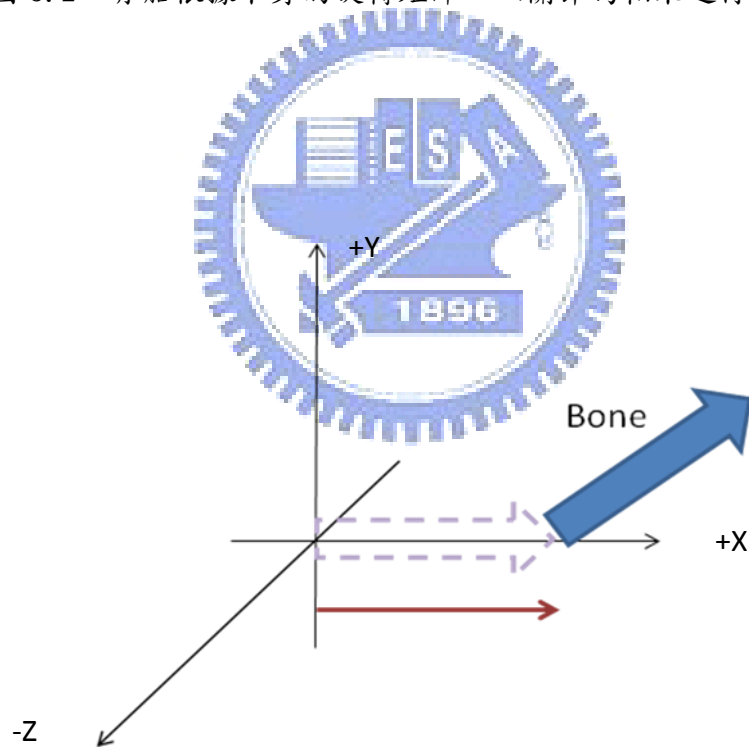


圖 3.3：骨骼依據本身的位移向量，預留空間給父骨骼，虛線為最初骨骼的位置，實線是經過位移後的改變。

相對於局部座標轉換，複合座標轉換則為了是表示整個骨骼連結完成，組成

骨架時，整個骨架的姿勢，如圖 3.4 所示。

複合座標轉換將把骨骼由自己的骨骼空間轉換到角色空間 (Character Space)。在骨架完成之後，我們就可以依據它的姿勢，來驅動之後所要產生的形變，移動皮膚上頂點的位置。

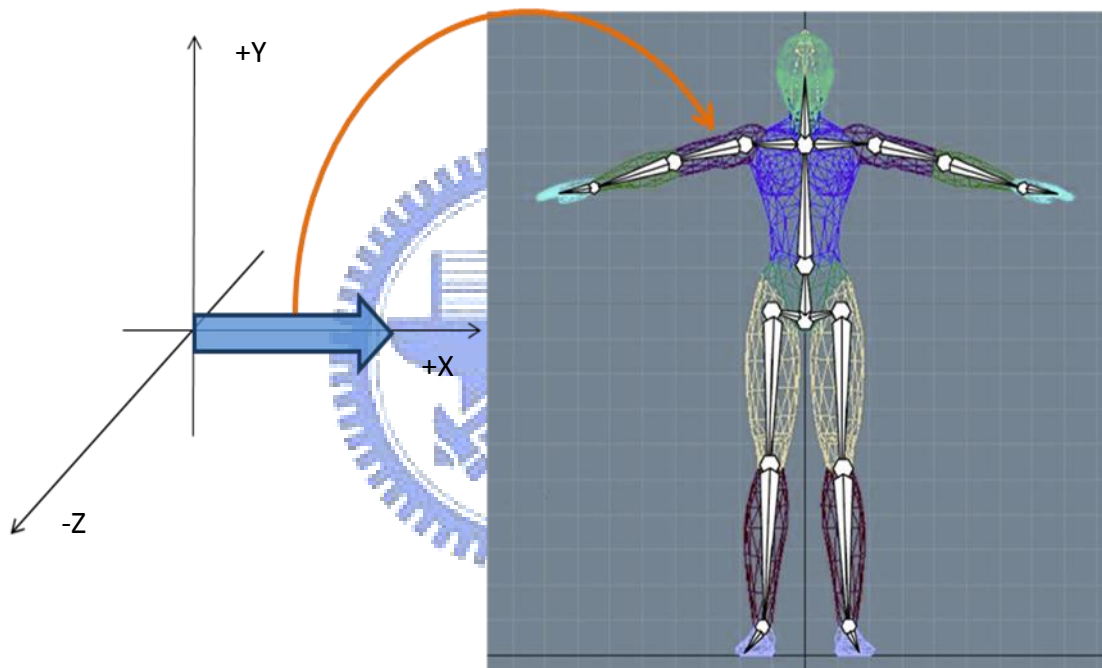


圖 3.4：借由複合轉換矩陣，我們可以將骨骼由骨骼空間轉換到角色空間。

以下將探討如何產生複合轉換。在每塊骨骼的局部轉換中，位移的部分包含了與父骨骼的相對關係，也就是與父骨骼的距離，因此骨骼之間並不是不相關連的，它們會受到父骨骼位置的影響，而改變其位置，以下用手臂例子來幫助我們了解如何形成複合轉換(圖 3.5)。

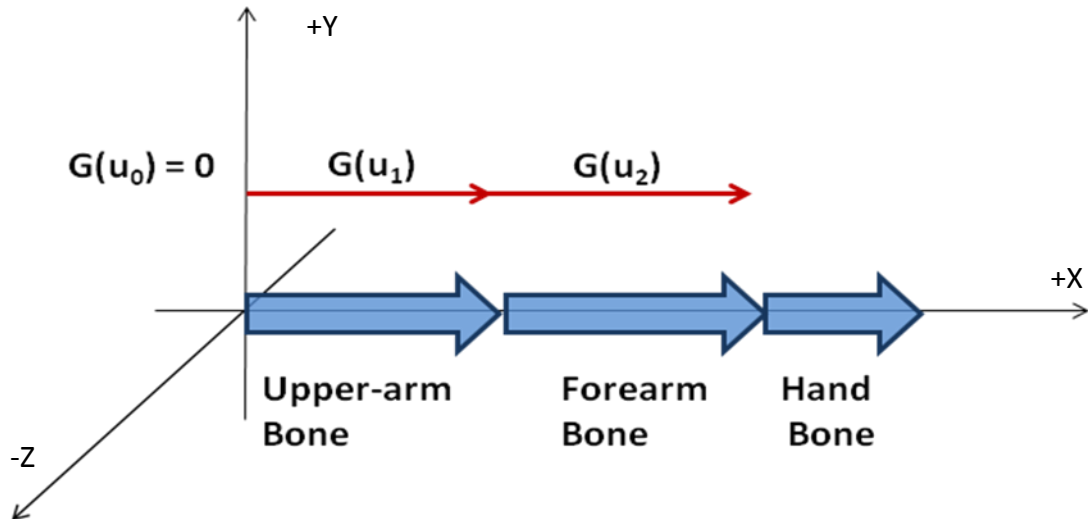


圖 3.5：手臂的骨架。在這裡可以看出來，我們複合了上臂骨骼，下臂骨骼，以及手骨骼，來完成手臂的骨架  $G(u_0)$  決定了上臂的位置， $G(u_1)$  決定了下臂以及  $G(u_2)$  決定了手的位置。而因為我們將上臂骨骼當作根(Root)骨骼，所以我們將其放置於原點，也就是為何  $G(u_0) = 0$ 。



手臂的骨架主要是由三塊骨骼來組成：上臂骨骼，下臂骨骼，以及手骨骼。

一開始，每塊骨骼分別位於自己的骨骼空間內。為了要完成手臂的骨架，此時我們必須要找出每塊骨骼的複合座標轉換矩陣，才能將骨骼轉換到角色空間，複合成骨架。

由前述所知，每塊骨骼局部轉換的位移部分會記錄著本身骨骼預留給父骨骼的空間，之後才是記錄自己的位置。我們現在從根骨骼開始放置骨骼，將其轉換至角色空間，接下來轉換骨骼的子骨骼到角色空間，因為骨骼上有父骨骼的位置資訊，因此根據此資訊我們便可以將子骨骼放到正確的位置。重複這個步驟，直

到骨骼全部轉換完成，便可以完成整個骨架。

以上是位移的部分，接下來我們根據圖 3.6 到圖 3.9 的部分，來說明旋轉部分的結合情況：

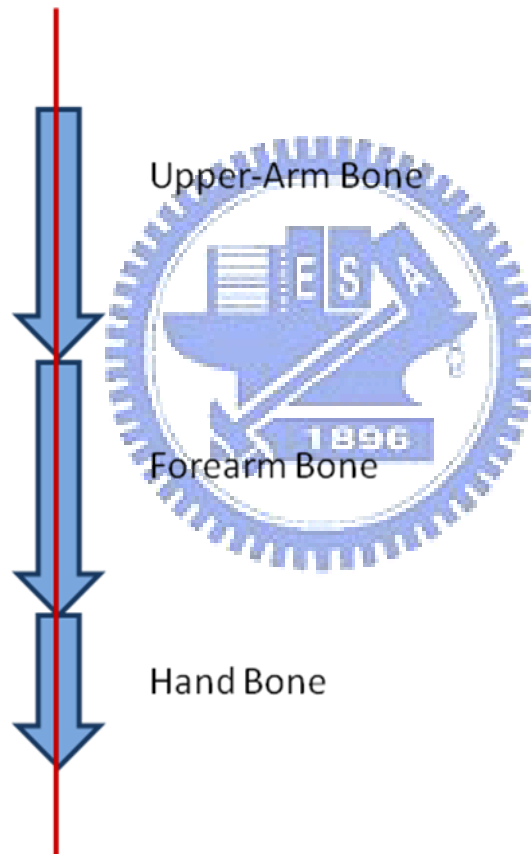


圖 3.6：初始的手臂骨架姿勢

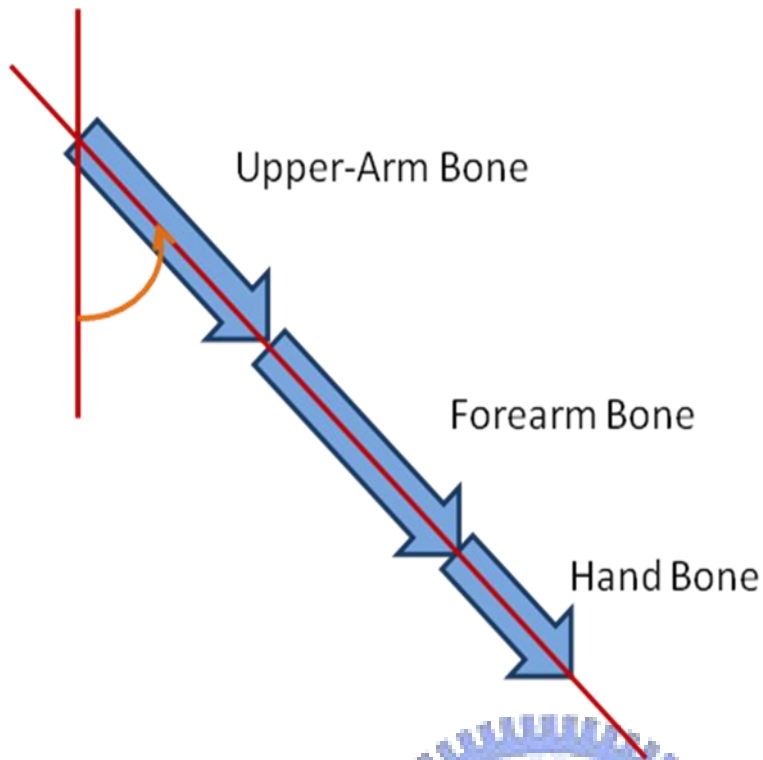


圖 3.7：第二步 手臂繞著肩膀關節旋轉

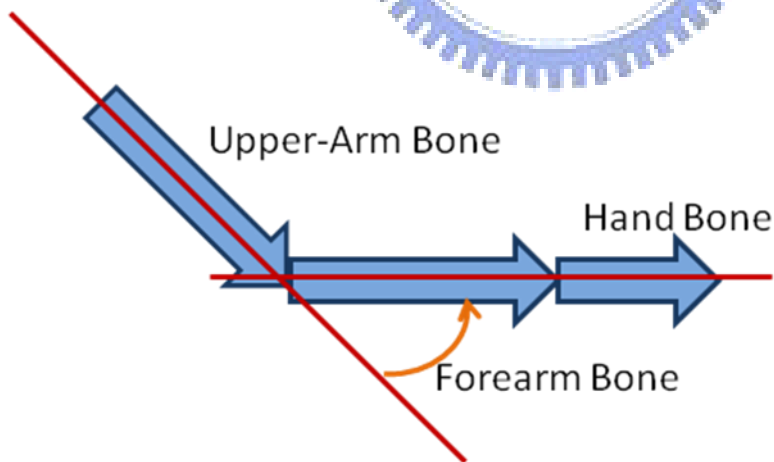


圖 3.8：第三步 下臂的部分依據手軸關節旋轉



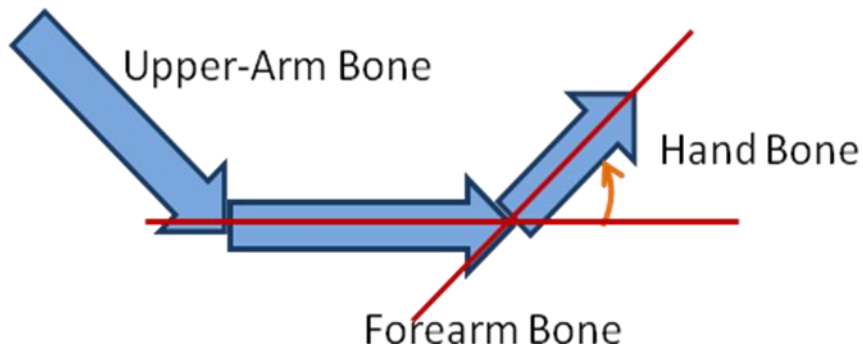


圖 3.9：第四步 手掌向上旋轉

依據人體物理學，如果將上臂以肩膀關節為軸來旋轉，除了上臂，下臂以及手掌也會跟著一起旋轉，如圖 3.7 所示。同樣的情況下，現在我們以肘關節為軸旋轉下臂，手掌也會跟著下臂一起旋轉，如圖 3.8 所示。最後旋轉手掌的時候，僅僅只有手掌會旋轉，如圖 3.9 所示。由這個例子可得知父骨骼的旋轉也會影響子骨骼，子骨骼的旋轉是所有祖先骨骼的旋轉累積之後，再加上自己的旋轉。

由以上兩個部分，我們可以了解骨架組成的方式。首先根據骨骼的局部轉換之位移部份，先決定尚未轉動後的骨骼，之後再以關節為軸心，利用旋轉部分來旋轉骨骼。重複這個步驟，從根骨骼一直轉換到最後一塊骨骼，完成我們所需的骨架。

以上的步驟表示在骨骼的連結中，不論是位移，或是旋轉的部分，都與父骨骼有關。整理之後，我們可以得到如何產生骨骼的複合轉換的步驟：將所有祖先

骨骼的複合轉換矩陣找出來，所有的轉換矩陣乘積起來，最後在加上自己的局部轉換，這樣就會是自己的複合轉換。數學上我們將複合轉換表示為 4x4 的矩陣：

$$C_i = \begin{cases} L_i & , \text{if } i = 0 \text{ (root)} \\ C_{i-1} L_i & , \text{otherwise} \end{cases}$$

$C_i$  表示第  $i$  根骨骼的複合轉換矩陣

$L_i$  表示第  $i$  根骨骼的局部轉換矩陣

$C_{i-1}$  表示第  $i$  根骨骼之父骨骼的複合轉換矩陣

在  $i = 0$  的情況下，表示這塊骨骼為根骨骼，它沒有任何的父骨骼，故它的局部轉換矩陣即為本身的複合轉換矩陣



## 3.2 骨骼子空間的形變

在上一節我們得到了如何去表示整個骨骼的結構，也就是骨架，接下來這一部分則要討論骨骼和皮膚之間的關係，以及如何模擬肌肉的形變。

一開始，我們將角色的皮膚網格依據骨骼來切割，使得每一個部分的網格都有一個相對應的骨骼。換句話說，我們在角色建模的時候，分別在每個骨骼空間建立模型的網格，再依據骨架結合成整個角色的網格，如此一來，因為用各個分段的網格與骨骼有相對應的關係，我們便可以利用每塊骨骼的複合轉換來建立出

整個角色的網格。

因此，每當角色的姿勢改變時，我們便重新計算出新的骨架姿勢，並重新計算每塊骨骼的複合轉換矩陣。利用重新計算得到的結合轉換矩陣，畫出每個骨骼相對應的角色分段網格。我們可以用簡易的數學式子來表示這個執行結果：

$$V' = C_i V$$

$V'$  表示角色網格之頂點新的位置

$C_i$  表示網格所對應到的骨骼之複合轉換矩陣

$V$  表示原本角色網格的頂點之位置



這個方法稱之為剛體動畫 (Rigid Body Animation)，基本上我們利用此方法，就可以輕鬆的操作角色動畫。不過剛體動畫有個嚴重的缺陷—此方法把每個分段網格視為不連續的部分，如圖 3.10 所示。

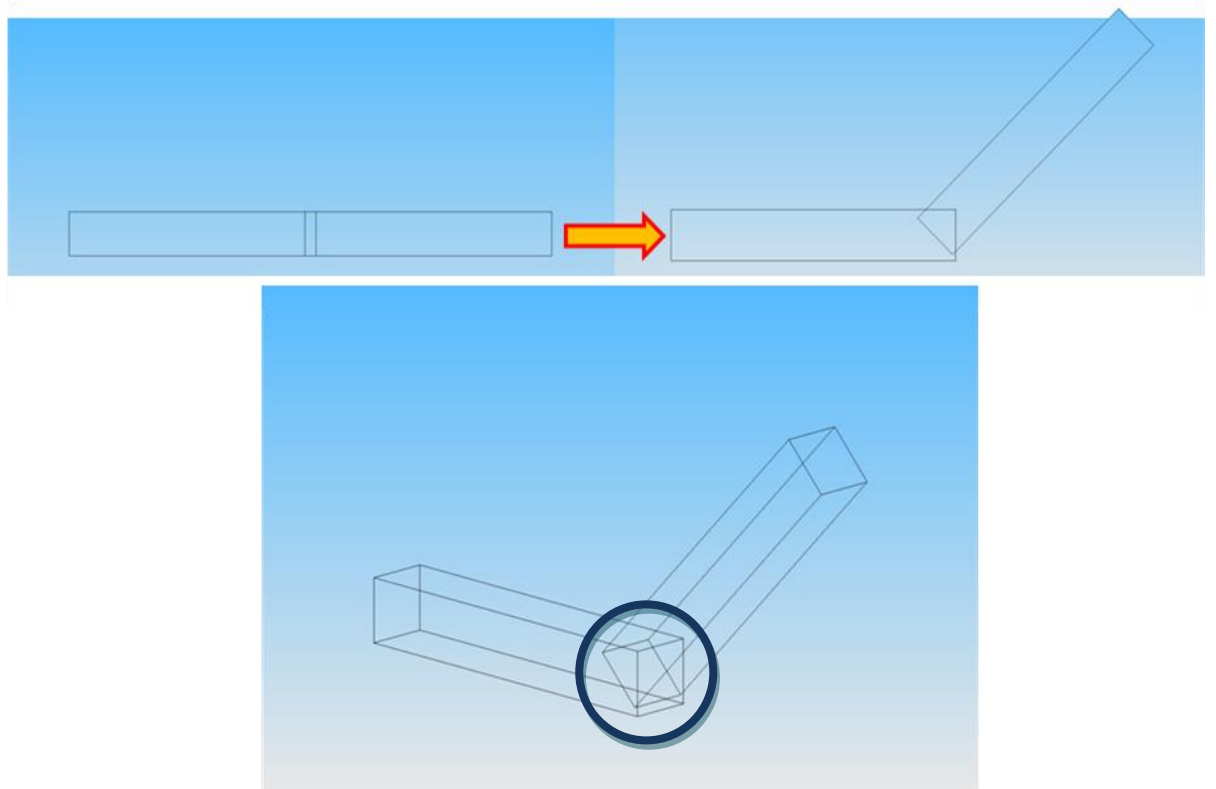


圖 3.10：這邊可以明顯的看出，此方法會把每個分段網格視為不連續的部分，圖中上下臂明顯是不連續的兩段網格，這在產生人物動畫的情況下，是很不自然的現象

因為剛體動畫把每一個分段的網格視為不連續的部分，但這不是我們所想要的結果，我們希望可以將每個分段的網格視為一連續的網格，如圖 3.11 所示：

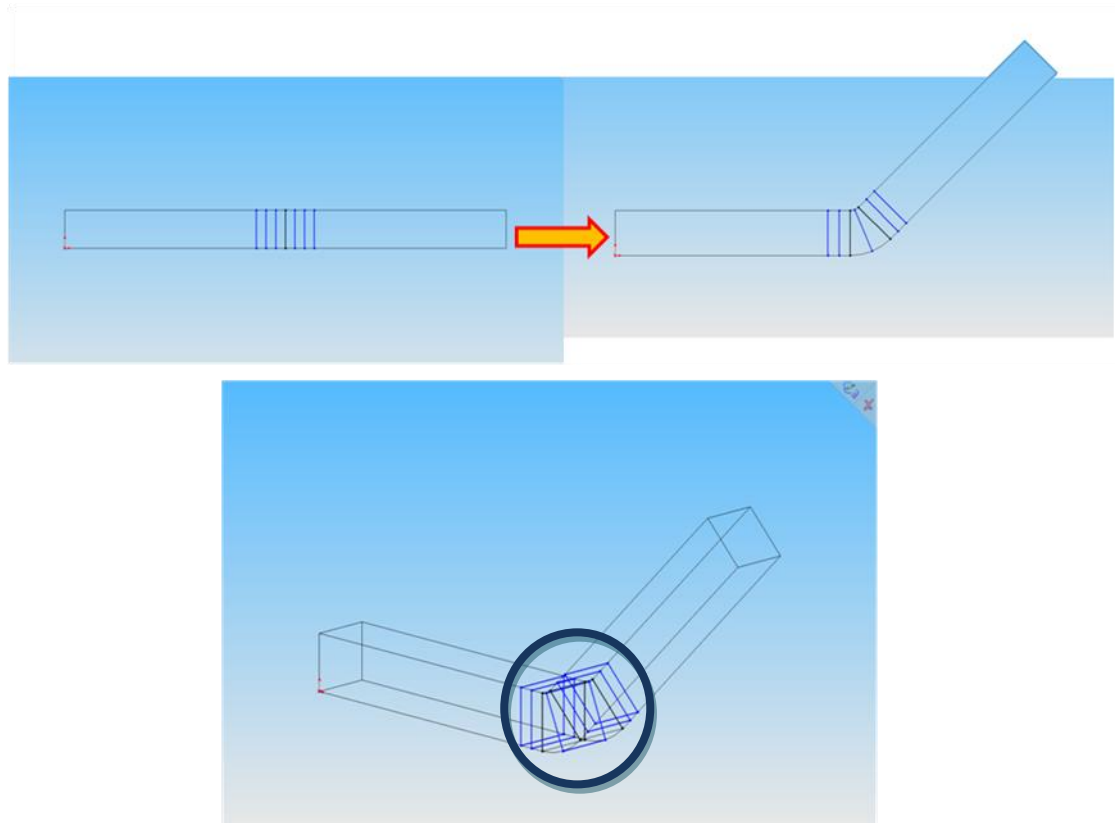


圖 3.11：觀察圈起來的部分，我們希望將每個分段網格視為連續網格

由上圖我們可以發現，在靠近關節部分的網格有一些被壓縮，有一些則被拉長，更重要的是，經由這結果，我們可以得到整個連續的網格。更精確的來說，在上臂以及下臂中，較為靠近關節點附近的網格，在骨架姿勢改變產生變形時，它的位置不僅會受到原本對應的骨骼影響，也會受到鄰近的骨骼影響，以上臂網格為例，不僅僅會受到上臂骨骼影響，也會受到下臂骨骼影響。

接著我們把焦點移到整個角色的網格，當我們要決定角色新的網格頂點時，我們不只該考慮原本對應的骨骼，還要考慮其他可能會影響該頂點的骨骼。此

外，由於網格頂點所受到每個骨骼影響的程度不同，我們根據影響的比率，產生大小不同的權重值以整合所有骨骼的影響量。在圖 3.11 的例子，網格上的頂點，會受到兩個骨骼的影響。這是骨骼子空間的形變 (Skeletal Subspace Deformation) 的關鍵點：一個頂點的位置會受到多重骨骼的影響。

在繼續談到骨骼子空間的形變演算法之前，我們還要處理一個問題。先前所提到的角色網格，它是依據每塊相對應的骨骼所分隔開的，因此，每個網格頂點的位置是相對於對應的骨骼，並且位於骨骼空間。之後我們利用骨骼的複合轉換矩陣轉換網格頂點的位置，將其由骨骼空間轉換到角色空間，建立整個角色。

而在骨骼子空間的形變這方面，我們則是將角色的皮膚視為位於角色空間的連續網格。我們需要的是原本角色模型的位在骨骼空間內的頂點座標，然而，由於角色模型建立時，本身會有一個預設的姿勢，因此我們沒有辦法直接利用複合轉換矩陣來使用骨骼子空間的形變。

為了解決這個問題，我們增加一個轉換機制，稱之為綁定轉換 (Binding Transform)。同樣的，每個骨骼都會有一個綁定轉換。這個綁定轉換矩陣是用來轉換在綁定姿勢 (Binding pose) 下的網格頂點到骨骼空間。所謂的綁定姿勢，就是在角色模型產生時的預設姿勢，此時網格上的頂點並未受到任何的骨骼轉換

矩陣所影響（即此時骨骼轉換矩陣皆為特徵矩陣（Identity Matrix））。

圖 3.12 的上半部表示處理的過程，綁定轉換將頂點由綁定空間轉換到骨骼空間。

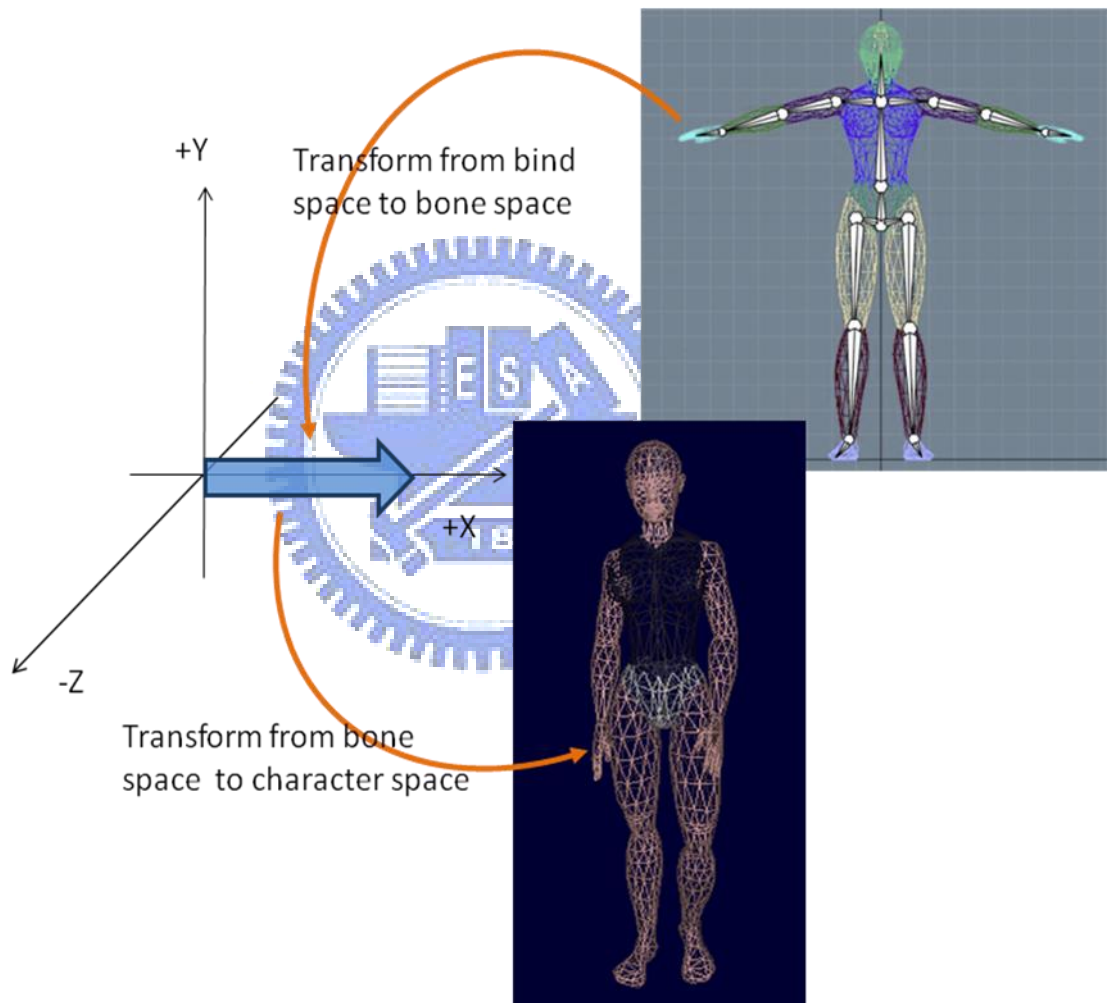


圖 3.12：轉換過程－綁定空間→骨骼空間→角色空間

現在我們要把網格上所有的頂點，都轉換到角色空間。整個流程如下：首先我們將其先由綁定空間轉換到骨骼空間，這個過程利用剛剛所提到的綁定轉換。

接下來，我們就可以利用先前的複合轉換，把頂點由骨骼空間轉換到角色空間，

圖 3.12 顯示整個轉換的過程。

我們可以把以上所提的綁定轉換及複合轉換組合起來，稱之為最終轉換。在

數學上我們依舊可用 4x4 的轉換矩陣來表示以上所提的兩個轉換：

$$F_i = C_i B_i$$

$F_i$  表示第  $i$  個骨骼的最終轉換矩陣

$C_i$  表示第  $i$  個骨骼的複合轉換矩陣

$B_i$  表示第  $i$  個骨骼的綁定轉換矩陣



接著我們把焦點轉回骨骼子空間的形變演算法上。我們需要角色的皮膚是一整個連續網格，為了達成目的，網格上的頂點在形變時，會受到不只一個骨骼影響。我們將影響該網格頂點的骨骼劃分為一個組合，稱為影響骨骼。每一個影響骨骼都有相對應的權重值，其中權重值表示該骨骼對頂點的影響量，對於每一組影響骨骼的權重值有以下的限制：

$$0 \leq w_i \leq 1, \text{ and}$$

$$\sum w_i = 1$$

$w_i$  表示第  $i$  個影響骨骼的權重。



現在我們可以利用最終轉換以及權重值來繼續接下來的步驟。我們將根據骨架姿勢的改變，轉換並改變角色的皮膚位置，且皮膚是連續的網格。一開始我們的頂點  $v$  是位於綁定空間的，我們接下來利用最終轉換、骨骼影響的權重值，將頂點移動到我們所要的位置，便完成骨骼子空間的形變。用數學式子來表示即為：

$$\begin{aligned}v' &= (w_1 v F_1 + w_2 v F_2 + \dots + w_{n-1} v F_{n-1} + w_n v F_n) \\&= (\sum_{i=1}^n w_i v F_i) \\&= (\sum_{i=1}^n w_i F_i) v\end{aligned}$$

$v'$  是頂點  $v$  在角色空間的位置

$w_i$  表示第  $i$  個影響骨骼的權重

$F_i$  表示第  $i$  個影響骨骼的最終轉換矩陣

$v$  是頂點  $v$  原先在綁定空間的位置

$n$  表示影響骨骼的數量

我們影響到頂點  $v$  的骨骼，將它的權重值乘上原本  $v$  在綁定空間的位置，再乘上骨骼的最終矩陣，最後把整組影響骨骼計算產生出的結果加總，得到的  $v'$  就是頂點  $v$  轉換到角色空間的位置。

### 3.3 產生肌肉隆起的效果

首先，我們利用骨骼子空間的形變來完成角色的整體形變。現在我們把目標放到更精細的描繪—肌肉隆起的效果。我們可以把肌肉隆起效果分為三個步驟：

1. 決定肌肉隆起的位置，2. 決定肌肉隆起的方向，3. 決定肌肉隆起的量。本部分將以上臂的肌肉部分作為範例。

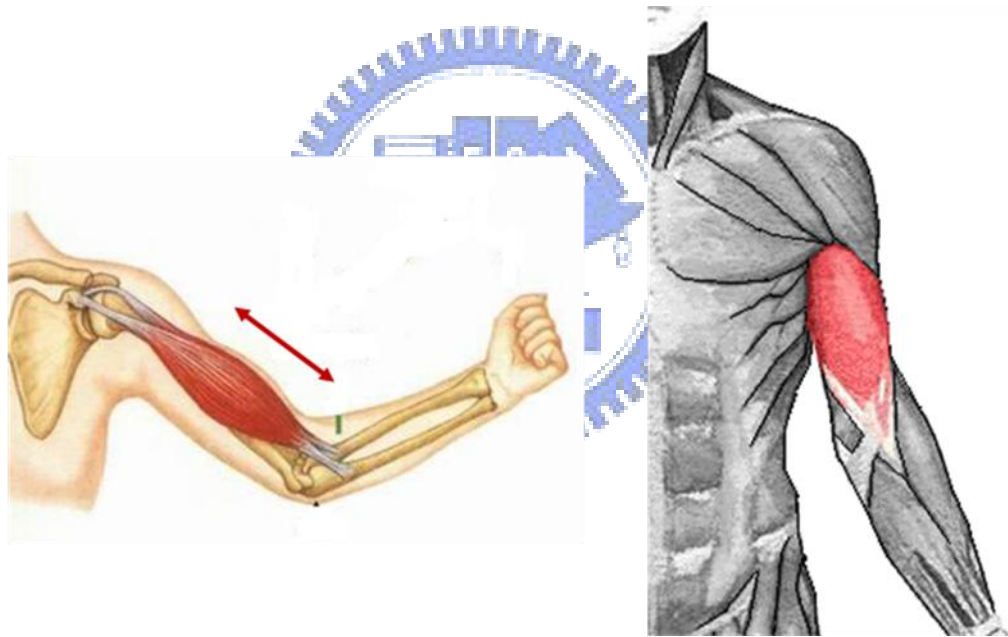


圖 3.13：二頭肌的位置

根據生物學[18]，我們知道肌肉隆起是因為肌肉受到刺激而產生收縮所造成的現象，而人體並不是每一塊肌肉都會有肌肉隆起的現象產生，上臂跟小腿的部

分是比較明顯會產生此種效果的部分。現在我們把目標放在上臂上，一樣由生物學我們可以得知上臂的二頭肌的位置是主要肌肉隆起產生的部位。因此若我們要模擬肌肉隆起的效果，我們首先需要界定出產生肌肉隆起效果的部位及範圍。觀察圖 3.13，我們可以發現二頭肌的位置大約是位在於上臂的下半部 2/3 的部分。利用這個觀察出來的結果，我們由此界定出肌肉隆起的位置以及範圍。

在我們找到隆起的範圍後，我們稱之為影響區域，肌肉隆起的效果只會發生在這一個區域內。對於角色模型而言，我們將會改變該區域內頂點的位置，以達到模擬肌肉隆起的效果，而在該區域範圍外面的頂點則不受影響，不改變其位置。



決定位置之後，接下來我們還要決定肌肉隆起的方向以及肌肉隆起的量。這三個部份都完成之後，我們就可以完成模擬肌肉隆起的效果。首先我們處理肌肉隆起的方向的這個部分。在觀察肌肉受到影響而收縮隆起時，其隆起的方向大致是向著表皮外膨脹，並且呈現一種相似於弧形的形狀。根據這一點，我們利用頂點法向量來作為肌肉隆起所依據的方向，如圖 3.14 所示。

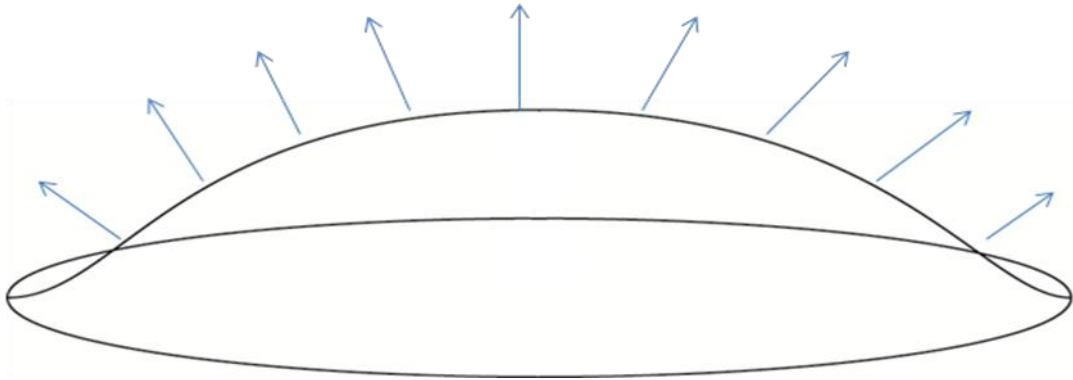


圖 3.14：肌肉隆起的方向

接下來要進行最後一個步驟了，也就是決定肌肉隆起的量。根據運動生理學 [18]，肌肉隆起時，二頭肌的中心部分，所隆起的量最多，而隨著與中心的距離越來越遠，隆起的量也越來越少。這個部份的特性與高斯函數 (Gaussian Function) 鐘形曲線相似。高斯函數也是一個平滑的函數，因此若用高斯函數來模擬肌肉線條的話，也不會有不連續的現象產生。在使用高斯函數前，還需要先進行一個步驟：首先將區域內皮膚網格的頂點，全部投影到以上下臂的關節，以及利用手掌關節到上下臂關節連線的垂直向量做為法向量所形成的平面上，並且找出該影響區域的中心點，如圖 3.15 所示。

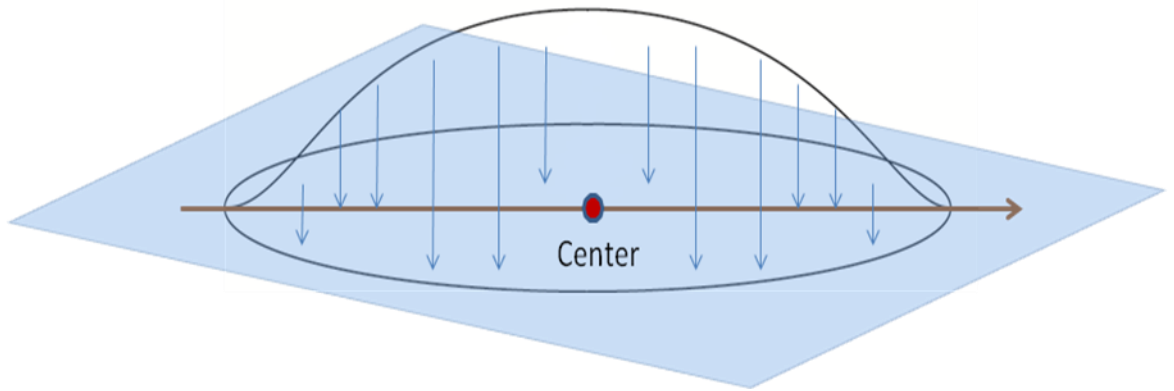


圖 3.15：將影響區域內的頂點投影到以上下臂的關節點連線形成的平面上，並且找出影響區域的中心點。

而肌肉的部分是一個立體的網格，此時我們已將影響區域範圍內頂點都投影到平面上，這邊就利用皮膚網格上的頂點投影到平面上位置，並且求出與影響區域中心點的距離，使用二維的高斯函數，來決定肌肉隆起的程度，並決定每個頂點位移的量：



$$1D : G(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-x^2/2\sigma^2}$$

$$2D : G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

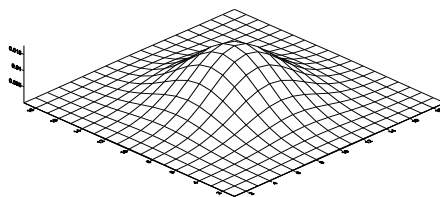


圖 3.16：二維高斯函數的示意圖

完成高斯函數的這個部分後，在進行動畫時，還要加上一個控制的參數，才能進行肌肉隆起模擬。肌肉隆起的量，不僅僅是受到與二頭肌中心的距離所影響，還會受到上臂以及下臂間夾角的差異所影響。在夾角逐漸縮小時，肌肉所受到的刺激會加大，隆起的量會加大，效果也會更明顯。因此除了高斯函數之外，我們還需要利用上下臂間的夾角大小，來決定肌肉所要隆起的量。

根據【運動生理學】[18]這本書裡面所提到的人體模型，我們可以由其中發現肌肉一些隆起的特性，見圖 3.17：

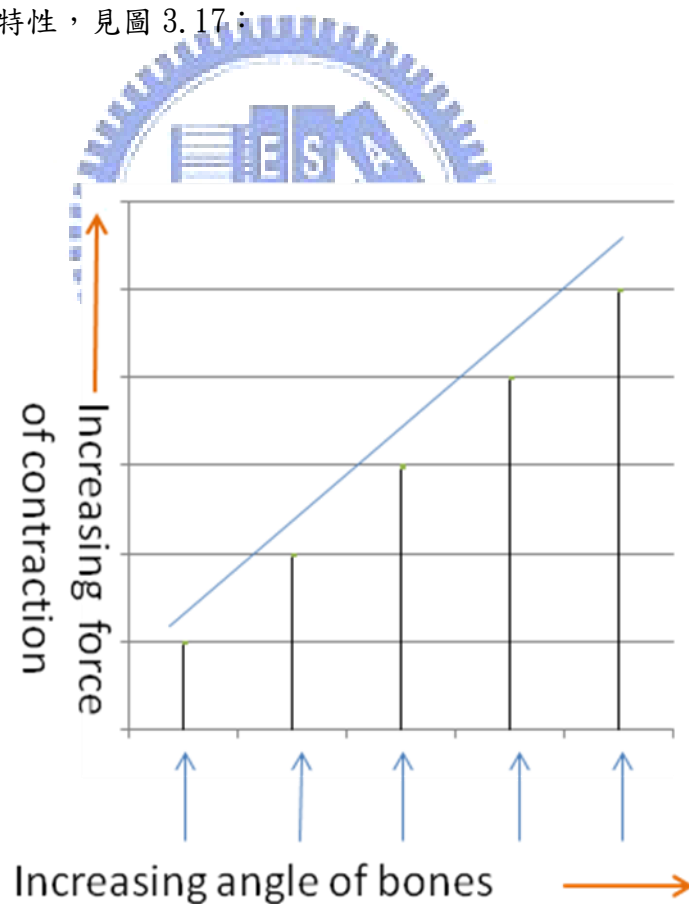



圖 3.17：肌肉隆起的特性

在肌肉受到刺激，力度由小加大時，肌肉所受到的力會呈現出線性成長的特性。而上下臂間夾角的改變，也會因角度的縮小，而肌肉產生呈現線性的膨脹。因此，我們在決定出在特定的角度的隆起量之後，其他變動較小的細微角度的隆起量就不用另外再重複上述的整個步驟，僅需要由與其它角度差異的比例，依線性調整隆起的量。

最後，結合 3.2 以及 3.3 兩部分，我們可以函式表示如下：

$$v' = \begin{cases} (\sum_{i=1}^n w_i F_i)v & \text{if } v' \text{ isn't in area} \\ (\sum_{i=1}^n w_i F_i)v + \delta G(x,y)v' & \text{otherwise} \end{cases}$$


$v'$  是頂點  $v$  在角色空間的位置

$w_i$  表示第  $i$  個影響骨骼的權重

$F_i$  表示第  $i$  個影響骨骼的最終轉換矩陣

$v$  是頂點  $v$  在綁定空間的位置

$n$  表示影響骨骼的數量

$\delta$  表示與我們決定出的特定夾角的差異比例

$G(x,y)$  表示高斯函數，其中  $x, y$  為頂點投影到平面後與影響區域中心點的

距離， $x$  是距離的  $x$  軸分量， $y$  是距離的  $y$  軸分量

$\vec{v}'$  表示  $v'$  法向量

綜合了 3.2 節及 3.3 節，我們就可以得到網格皮膚頂點最後的位置，我們就可以利用這些座標點，繪出我們想要的角色名模型。





# Chapter 4

## 結果



本系統是在 Intel Core 2 Duo 2GHz，4G 記憶體，顯示卡為 nVidia Geforce 8600M GT 的電腦上，利用 DirectX 9.0c 和 C++ 語言來實作。所使用的 Model 有兩個，分別為：

1. Skeleton：有 1844 個頂點，組成 3342 個三角形，全身則有 23 個關節點
2. Woman：有 2365 個頂點，組成 4412 個三角形，全身則有 37 個關節點

效能方面，在僅使用骨骼子空間的形變的情況下，FPS 大約為 202—211 格 (Frame) 之間，在增加了肌肉隆起效果之後，速度降低不多，FPS 大約在 174—181 格之間。在肌肉隆起的效果上，在不同的區域上，手臂的上臂部分大約影響了該區段 20% 的頂點 (在 232 個頂點中，影響約 51 個頂點)，小腿的部分則大約影響

該區段 15% (213 個頂點中，影響約 30 個頂點)。在操作角色骨架的變形方面，則是使用正向運動學(Kinematics)的方式，用不同的角度來轉動模型上的關節，以達成骨架的變形，之後再利用變形後的骨架，來完成角色模型的形變。

在結果圖的部分，圖 4.1 為 Woman 以及 Skeleton 的預設姿勢，圖 4.2 到圖 4.7 則為上下臂在不同角度的情況下，有無肌肉隆起的結果以及比較。圖 4.8 為小腿部分有無肌肉隆起的比較，圖 4.9 到圖 4.12 為其他的形變結果。

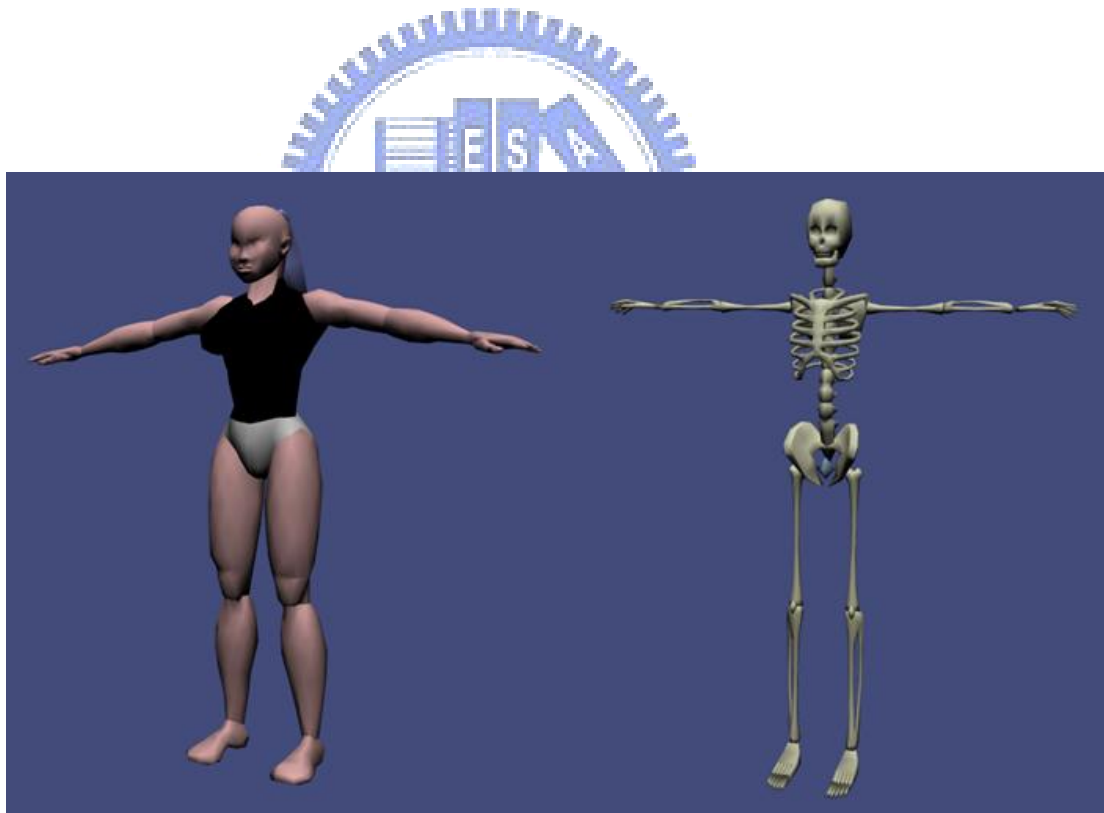


圖 4.1：Woman 與 Skeleton 的預設姿勢

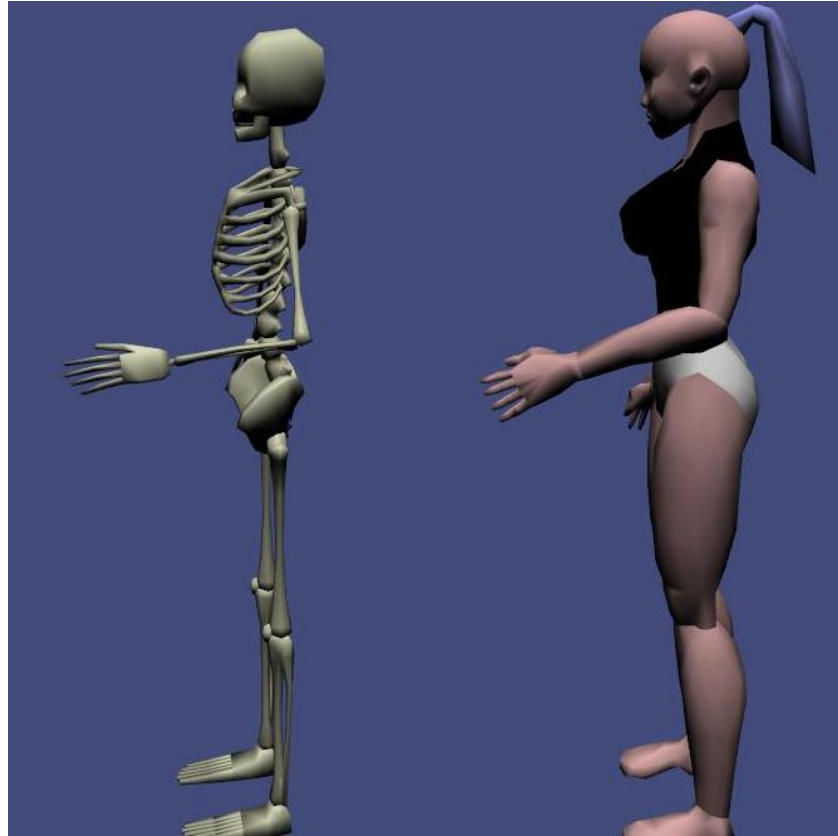


圖 4.2：Skeleton 與無肌肉隆起的形變

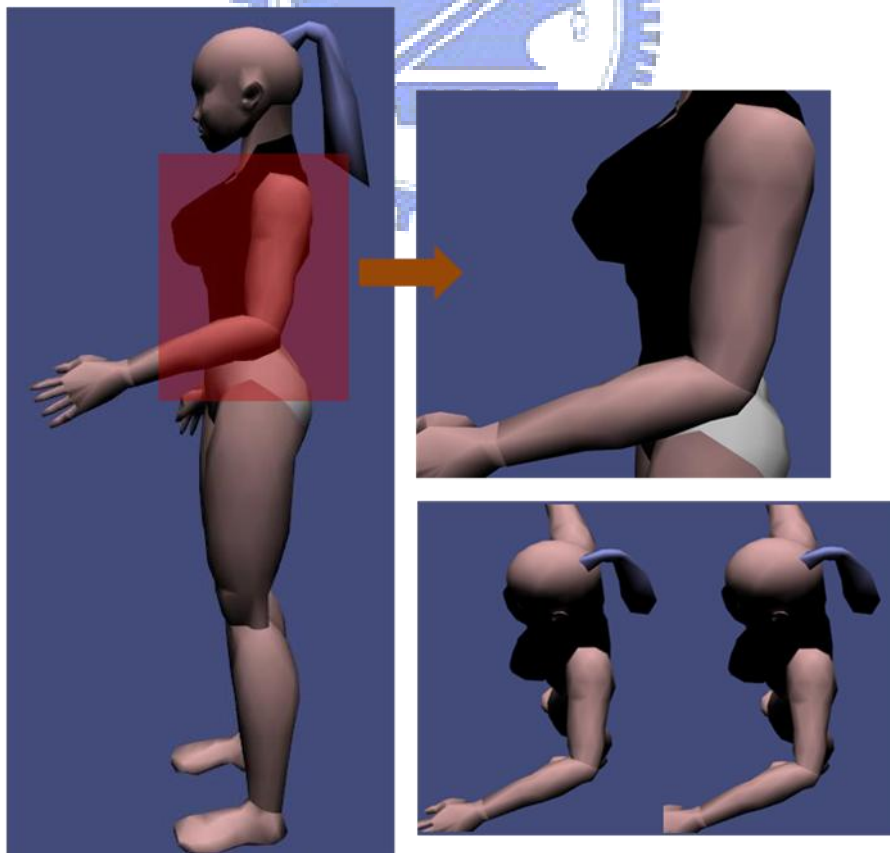


圖 4.3：有肌肉隆起，右上為放大圖，右下為與無肌肉隆起的比較

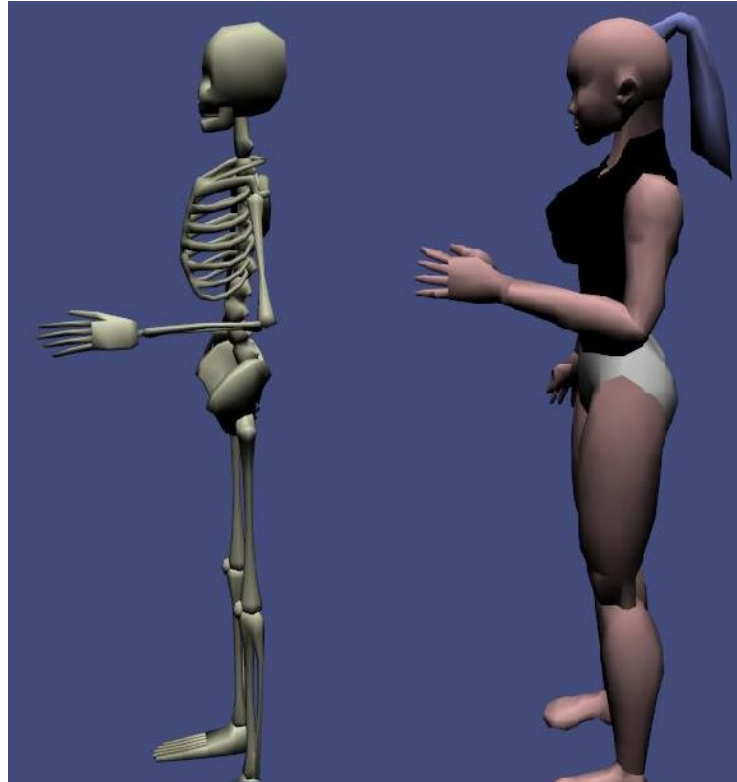


圖 4.4：Skeleton 與無肌肉隆起的形變

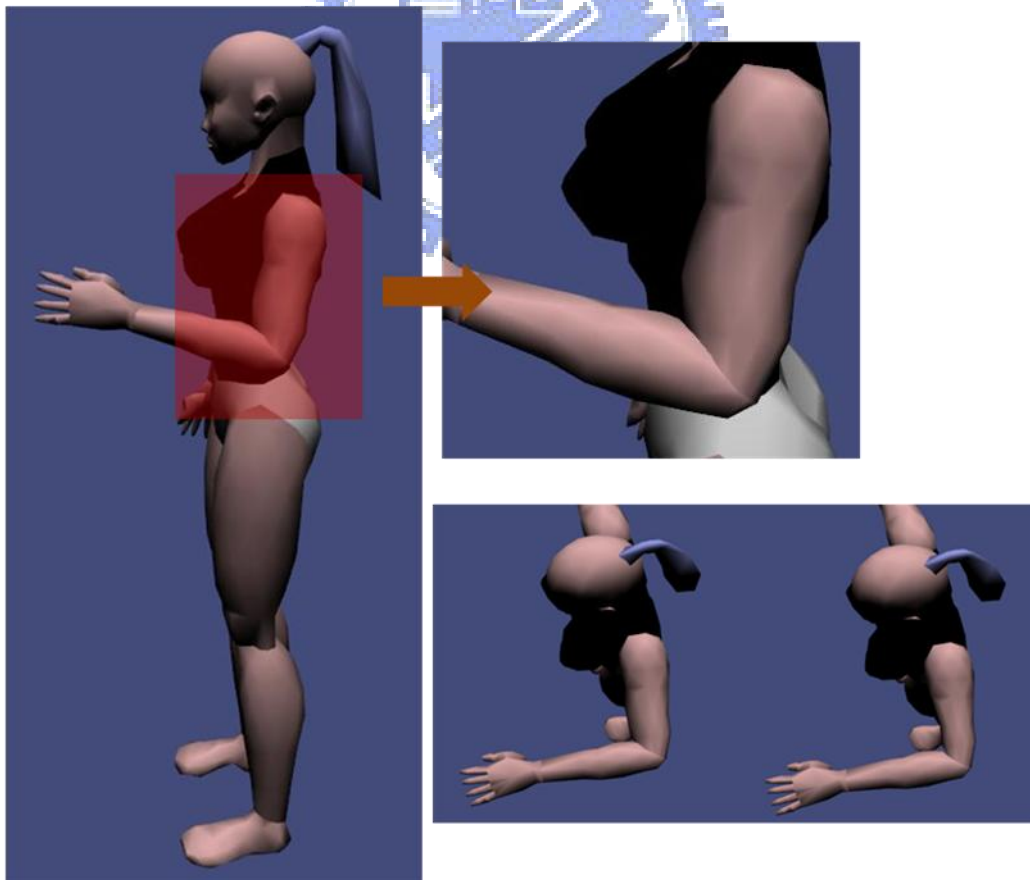


圖 4.5：有肌肉隆起，右上為放大圖，右下為與無肌肉隆起的比較

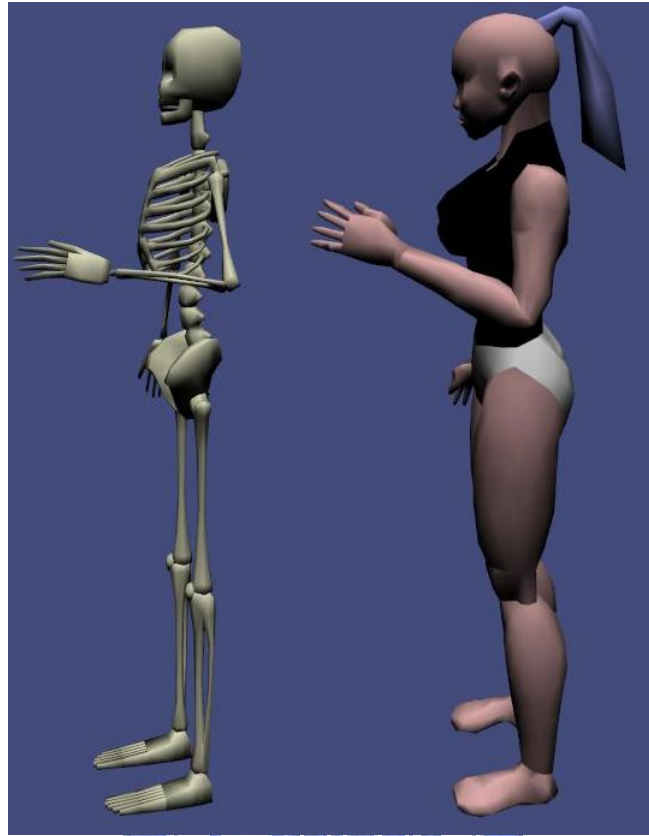


圖 4.6：Skeleton 與無肌肉隆起的形變

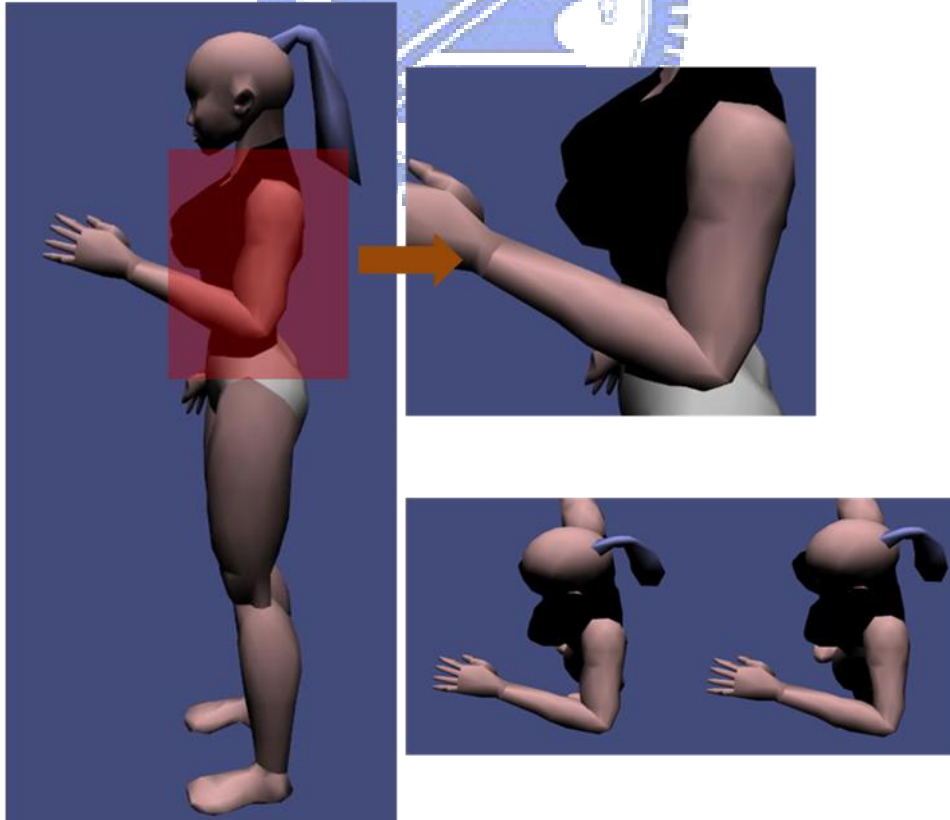


圖 4.7：有肌肉隆起，右上為放大圖，右下為與無肌肉隆起的比較

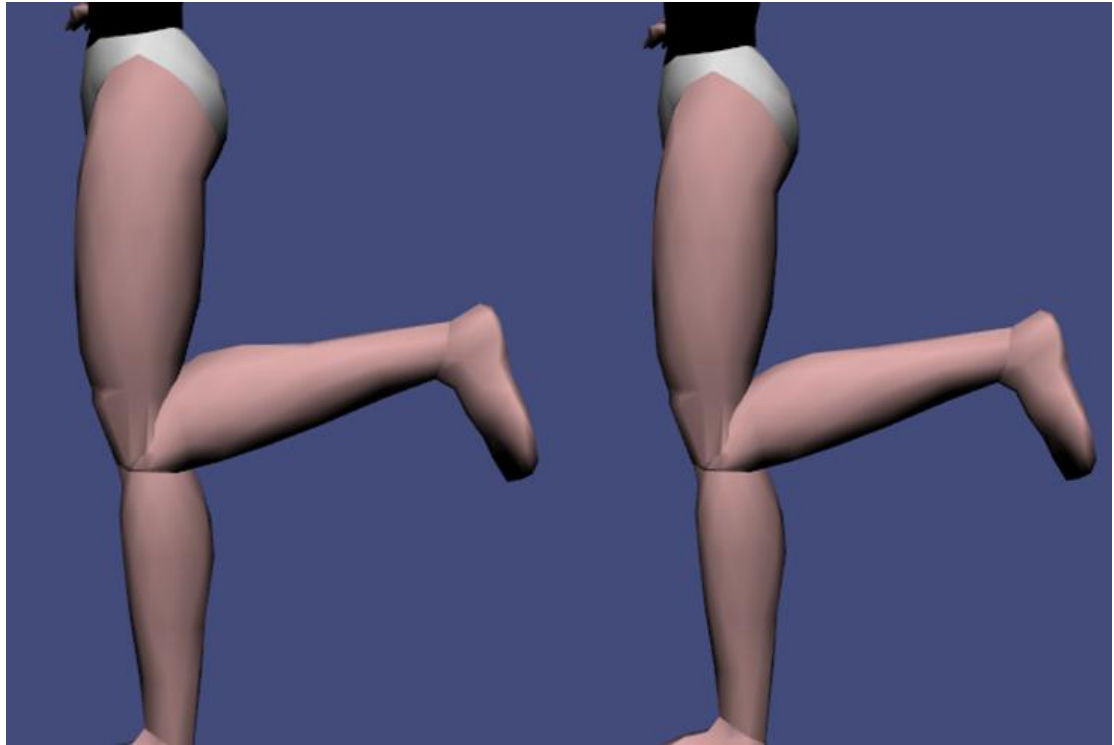


圖 4.8：小腿在有無肌肉隆起效果的比較



圖 4.9：其他形變結果

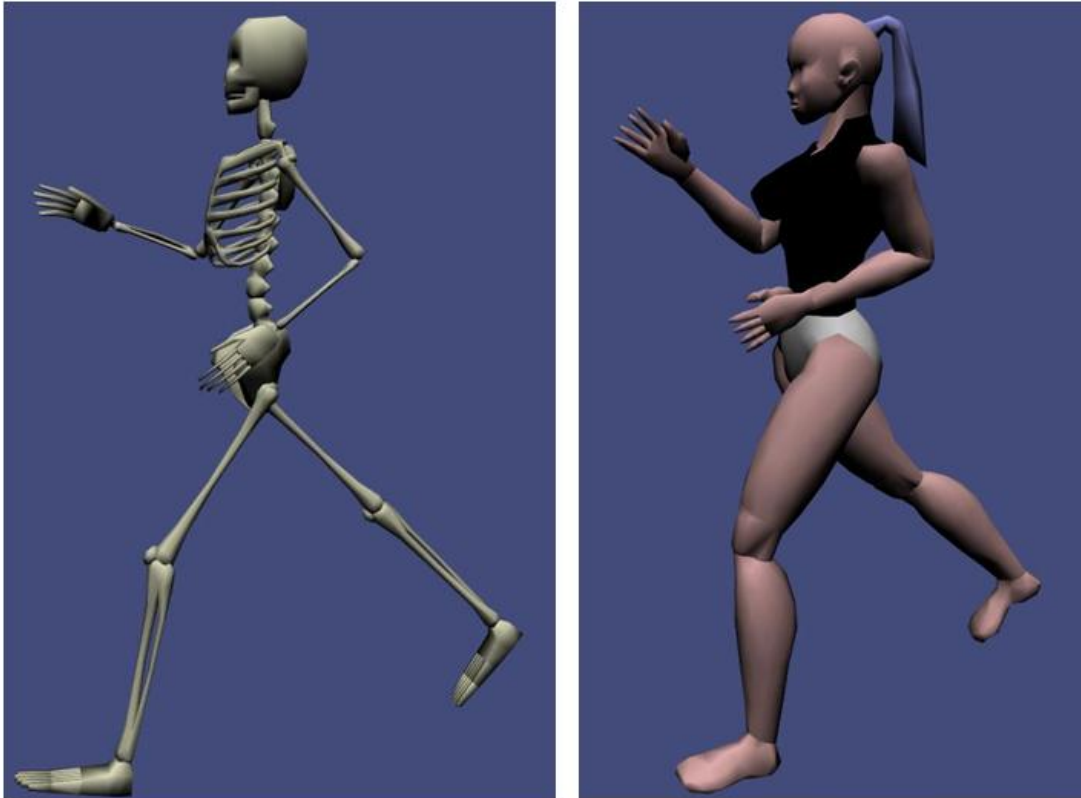


圖 4.10：其他形變結果

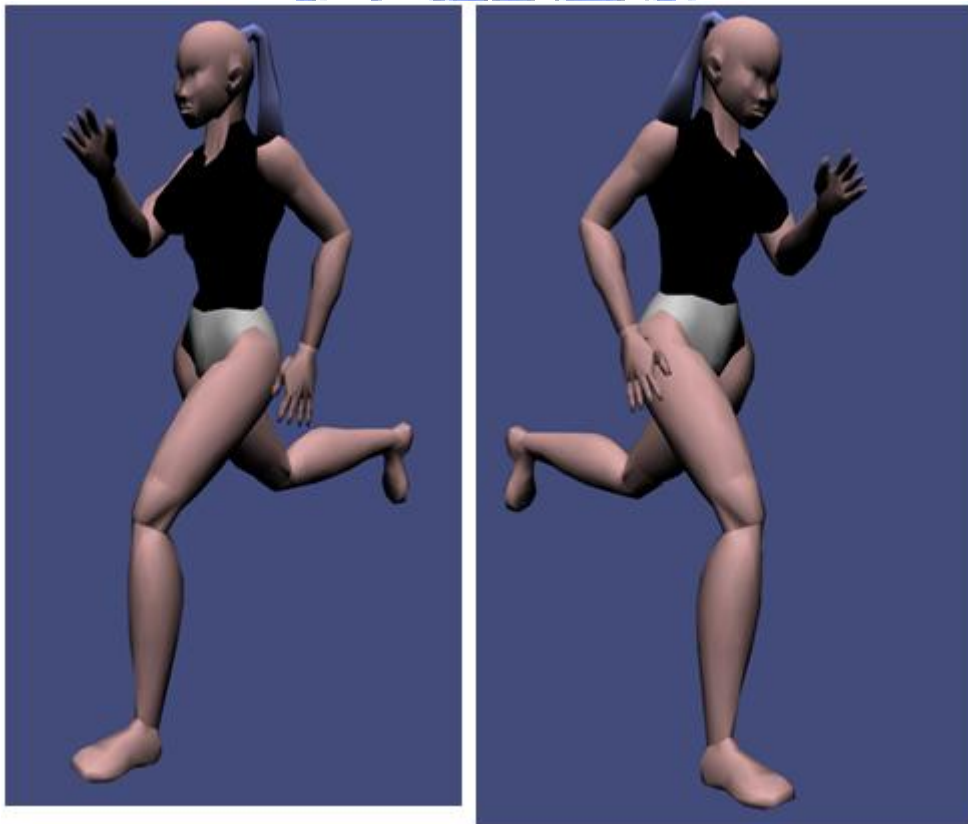


圖 4.11：其他形變結果





圖 4.12：其他形變結果





# Chapter 5

## 結論與建議



在本篇論文中，我們提出一種因肢體動作所產生的肌肉形變的合成方法。當一個角色模型的姿勢改變時，我們需要根據其骨骼，改變皮膚頂點的位置。我們利用骨骼子空間的形變為基礎，先完成整體的形變。接下來再做更細緻化的修改——加入肌肉隆起的效果。利用高斯函數以及關節之間的夾角，來模擬肌肉隆起的效果。依序完成上述兩個步驟之後，便可描繪出角色模型因為姿勢的改變，所形成的肌肉變化效果，而且在不需要肌肉隆起範例的情況下，亦能產生肌肉隆起的效果。

另外還有幾個問題和可以發展的方向，留待未來的研究解決。

1. 加入碰撞偵測：在骨骼間夾角持續的縮小的情況下，角色的表皮可能會

出現交錯而產生錯誤的情況，加入了碰撞偵測可以有效的改善這一部分的問題。

2. 還有一些其他的模擬形變方法可以來模擬肌肉隆起的效果，並且改善在旋轉角度過大的情況下，骨骼子空間的形變所產生的不自然現象。

3. 在肌肉隆起的效果方面，可以增加一些個人化的效果，如血管浮起的特殊效果。

4. 在執行速度方面，可以嘗試將此演算法移動到 GPU 上面來實作，加快整體的繪製速度。



## 參考文獻

- [1] Alexa M., “Differential coordinates for local mesh morphing and deformation,” *The Visual Computer* 19, 2-3, 2003.
- [2] Allen, B., Curless, B., and Popvic, Z., “Articulated Body Deformation from Range Scan Data,” *Proceedings of ACM SIGGRAPH*, 2002.
- [3] Angelidis, A., and Singh, K., “Kinodynamic skinning using volume-preserving deformations,” *In Proc. Symposium on Computer Animation*, 2007.
- [4] Botsch, M., and Sorkeine, O., “On linear variational surface deformation methods,” *IEEE Transactions on Visualization and Computer Graphics*, 2008.
- [5] Capell, S., Burkhart, M., Curless, B., Duchamp, T., and Popovic, Z., “Physically based rigging for deformable characters,” *Graphical Models* 69, 2007.

- [6] Huang, J., Shi, X., Liu, X., Zhou, K., Wei, L. Y., Teng, S. H., Bao, H., Guo, B., Shum, H. Y., “Subspace Gradient Domain Mesh Deformation,” *Proceedings of ACM SIGGRAPH*, 2006.
- [7] Hyun, D.E., Yoon, S.H., Kim, M.S., and Juttler, B., “Modeling and Deformation of Arms and Legs Based on Ellipsoidal Sweeping,” *In Proceedings of the 11<sup>th</sup> Pacific Conference on Computer Graphics and Applications*, 2003.
- [8] Hyun, D.E., Yoon, S.H., Chang, J.W., Seong, J.K., Kim, M.S., and Juttler, B., “Sweep-Based Human Deformation,” *The Visual Computer*, vol. 21, no. 8-10, pp.542-550, 2005.
- [9] Joshi, P., Meyer, M., DeRose, T., Green, B., and Sanocki, T., “Harmonic coordinates for character articulation,” *ACM Trans. Graph*, 2007.
- [10] Ju, T., Schaefer, S., and Warren, J., “Mean value coordinates for closed triangular meshes,” *ACM Trans. Graph*, 2005.
- [11] Ju, T., Zhou, Q., van de Panne, M., Cohen-Or, D., and Neumann, U., “Reusable skinning templates using cage-based deformations,” *ACM Trans. Graph. Article No. 122, SCI*, 2008.
- [12] Lewis, J. P., Corder, M., and Fong, N., “Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation,” *In*

*SIGGRAPH*, 2000.

- [13] Lipman, Y., Kope, J., Cohen-OR, D., and Levin, D., “Gpu-assisted positive mean value coordinates for mesh deformations,” *In Symp. on Geometry processing*, 2007.
- [14] Lipman, Y., Sorkine, O., Levin, D., and Cohen-OR, D., “Linear rotationinvariant coordinates for meshes.” *ACM Trans. Graph.* 2005.
- [15] Merry, B., Marais, P., and Gain, J., “Animation space: A truly linear framework for character animation,” *ACM Trans. Graph.*, 2006.
- [16] Nealen, A., Sorkine, O., Alexa, M., and Cohen-OR, D., “A sketch-based interface for detail-preserving mesh editing,” *ACM Trans. Graph.*, 2005.
- [17] Sand, P., McMillan L., and Popović J., “Continuous capture of skin deformation,” *ACM Trans. Graph.* 22 (3) (2003), pp. 578–586.
- [18] Scott K. Powers, and Edward T. Howly., “Exercise Physiology – Theory and Application to Fitness and Performance 4<sup>th</sup> edition.”
- [19] Sederberg, T. W., and Parry, S. R., “Free-form deformation of solid geometric models,” *SIGGRAPH Comput. Graph.* 20.
- [20] Singh, K., and Kokkevis, E., “Skinning characters using surface oriented free-form deformations,” *In Proceedings of Graphics Interface*, 2000.
- [21] Sloan, P., Rose III, C., and CohenN, M., “Shape by example,” *Symp. on*

*Interactive 3D graphics*, 2001.

[22] Sorkine O., “Differential representations for mesh processing,” *Computer Graphics Forum*, 2006.

[23] Sorkine, O., Cohen-OR, D., Lipman, Y., Alexa, M., Rossl, C., and Seidel, H.P., “Laplacian surface editing,” *In Proceedings of the symposium on Geometry processing*, 2004.

[24] Summer, R. W., Zwicker, M., Gotsman, C., and Popovic, J., “Mesh-based inverse kinematics.” *ACM Trans. Graph.*, 2005.

[25] Wang, R., Pulli, K., and Popovic, J., “Real-time enveloping with rotational regression,” *ACM Trans. Graph.*, 2007.

[26] Wilhelms, J., and Gelder, A.V., “Anatomically based modeling,” *In SIGGRAPH*, 1997.

[27] Yoshizawa, S., Belyaev, A. G., and Seidel, H.P., “Skeleton-based variational mesh deformations,” *Computer Graphics Forum*, pp.255-264. 2007.

[28] Yu, Y., Zhou, K., Xu, D., Shi, X., Bao, H., Guo, B., and Shum, H.Y., “Mesh editing with poisson-based gradient field manipulation,” *ACM Trans. Graphic*, 2004.

[29] Zhou, K., Huang, J., Snyder, J., Liu, X., Bao, H., Guo, B., and Shum, H.Y.,

“Large mesh deformation using the volumetric graph laplacian,” *ACM Trans. Graph*, 2005.

