

虛擬環境使用情境感知人物臉部表情形變技術

A Situation-Aware Mesh Morphing of Facial Expression for Virtual Environments

盧天麒 (Tainchi Lu)
國立嘉義大學資訊工程學系
tclu@mail.ncyu.edu.tw

張宏彰 (Hungchang Chang)
國立嘉義大學資訊工程學系
s0940306@mail.ncyu.edu.tw

摘要

本論文主要是探討在虛擬環境中人物角色的表情如何動態表現的更加生動活潑。人物在虛擬環境裡根據不同的環境而產生不同的表情，主要分為兩大部份，首先我們利用基因演算法本身所具有最佳化搜尋的特性，針對人物角色的所在環境做判斷，並搜尋出最適合的參數值。當最佳解被計算產生後，就利用三維形變的技術將找到其相對應的表情做網格形變，藉由搜尋表情資料庫來分析擷取出適合此刻情境的人臉表情網格，以快速即時的形變技術進行人臉表情的網格形變，並配合階段式形變技術讓人臉表情生動不僵硬，且因網格形變的原始形狀和目的形狀差異不大，不至於佔用大部分電腦計算資源運算；此項技術的目的可將三維電腦遊戲的人物角色表情，由傳統前置處理二維影像貼圖提升至情境感知三維網格形變的階段，如此一來就可達到即時生動的人臉表情變化效果。

關鍵詞：虛擬環境、基因演算法、三維形變、情境感知、人臉表情

1. 前言

三維電腦圖學[1]在多媒體產業的發展中，在許多方面皆會運用到動畫的技術，不管是在娛樂、商業、教學許許多多方面皆會使用到，尤以三維動畫技術使用最為廣泛。以多媒體娛樂產業為例，如電影、電視、卡通、遊戲許多的特效都和動畫有著密不可分的關係存在。國內數位內容的起步也代表著將朝向多媒體的繽紛世界發展，數位內容產業中受到大眾所喜愛的便是電腦遊戲動畫，以三維電腦遊戲及動畫產業為例，國內市場蓬勃發展，不管是代理國外動畫或國內自創遊戲，都代表電腦遊戲動畫產業將成為電腦圖學技術的核心應用

範疇。

大部分電腦遊戲的人物幾乎沒有使用人臉表情技術，即使有顯示表情也是採用事前準備的二維影像材質進行貼圖或二維影像形變，並不能即時根據不同人物或場景的變化進行人臉表情的動態轉換，近年來 facial animation 技術突飛猛進，包含使用 morphing between libraries of key poses, blend shapes, motion capture, motion dynamics simulations 以及 goal-oriented techniques 等技術，但有些專業技術必須投入龐大經費建置軟硬體設施，甚至建造大規模真實模型，因此人力及物力成本所費不貲。

於是我們利用虛擬人物對環境場景的動態變化，皆會對環境的變化而有其相對應的情緒反應出現，所以我們使用基因演算法的搜尋作用，來針對環境中的每個可能會對人物產生反應的環境因素做一個搜尋的動作，並找出其相對應的表情出來，並以快速即時的三維形變技術進行人臉表情的網格形變，進而達到不同的虛擬環境皆可產生動態的臉部表情變化。

本研究主要是在探討遊戲人物在虛擬環境中如何表現的更加生動、活潑，人物在遊戲裡會根據不同的環境而產生不同的表情。本研究主要分為兩大部份，首先，我們利用基因演算法本身所具有最佳化搜尋的特性，來對角色人物的所在環境做判斷，並搜尋出最適合的表情出來。當我們找到最佳解時，就利用三維形變的技術將找到其相對應的表情做表情形變，反應在角色人物身上。如此一來就可達到生動的效果而不是靜態表情的狀況。

2. 相關工作

本研究目的主要是在虛擬環境中如何使角色人物能夠依據環境的不同進而產生多變且生動的表情，我們利用基因演算法具有最佳解搜尋的特性和三維形變的技術來表達角色

人物的表情變化。此章節將針對基因演算法和三維形變進行簡要的說明及介紹。

首先由 C. Burgy [2]所提出的 Interaction Constraints Model 系統架構是在說明在不同的周遭環境中的各種情況認知判斷。由於我們主要是對虛擬環境做整個場景的辨別，以便產生出各種表情出來，所以 Interaction Constraints Model 將會有助於我們對整個場景的分析。

2.1 基因演算法

具有演化式計算所建立的計算模式統稱為演化式演算法[3][4]，如演化式規劃、演化策略、基因演算法。在整個演化式計算研究中，範圍最為廣泛的研究主要為基因演算法，基因演算法可有效的搜尋出問題最佳解並產生較為穩定的結果，其主要運作方式為模仿生物界中的生存法則，適者生存，不適者淘汰，這種物競天擇的自然演進法則，可有效的選擇出具有較好特性的物種並繼續繁衍下去而產生出比上一代更優秀的子代，如此一直反覆的演化之後以產生適應性最高的子代。進而找到最理想的解答。

基因演算法(Genetic algorithm, GA)是由 Holland 於 1975 年所提出，並於往後的研究中[5]陸續有學者將基因演算法使用在生物學、經濟學、管理學、模式辨式、國防研究、設計與規劃...等等，在生物學領域的應用是在研究如何在蛋白質結構中預測出蛋白質二級結構的準確率，主要是做為準確度預測上。經濟學領域的應用是用來做為投資組合計算的預測系統。管理學領域的應用是用來處理排程問題或是求得最佳路徑、最短路徑等旅行推銷員問題。模式辨式領域的應用是利用基因演算法所具有的評量比對、局部保留、重組的功能具有逐步修正接近解答的能力因此可用來做為模式方面的比對工作，或是在模式比對的過程中去除雜質以產生較為純粹的模式。國防研究上的應用是用來研究在戰術與戰略的選定與武器系統的研發兩方面。設計與規劃上的應用是來解決在空間、環境設計問題上如何求得最佳參數解，在這些研究中發現當問題過大或過於複雜時，基因演算法可有效的求解出最佳解。

基因演算法共可分成五個步驟來執行，分別為染色體編碼、計算適應性函數、選擇、交配、突變，以下將對這五大步驟來做介紹。

(I)染色體編碼

在做基因演算法之前我們必需針對問題

和目標訂定出一串連續字串來做為執行基因演算法的工作，也就是對參數做編碼的動作來做為搜尋的依據，常見的編碼主要有二進位和非二進位二種。二進位編碼就是對參數轉換成 0、1 型態，如圖 1 所示。

二元編碼

染色體 A	01010011010011110011
染色體 B	10011100010101001100

圖 1 二進位編號

而非二進位編碼又可分成實數編碼、符號編碼主要是針對比較複雜的問題所設計，其表示方式如圖 2 所示。

實數編碼

染色體 A	1.234 2.345 3.456 4.567
-------	-------------------------

符號編碼

染色體 A	ABCDEFGHIJKLMNQRST
-------	--------------------

圖 2 非二進位編碼

在編碼時可對問題所需而自訂出不同的編碼方式，好的編碼方式將可有效的提昇整個工作效率。

(II)計算適應性函數

基因演算法在求解最佳解之前必需對每一個染色體也就是母代，給予一個適應值，我們可對問題的目標設計出適應值，適應值的給定將會影響到最終的結果，所以在設計適應值時必需很细心的設計才不會影響到結果的好壞。適應值愈大被選中或複製的機率就愈大，表示該染色體具有較好的特質可演化到下一代，反之，就不會被選上進而遭到淘汰。因此適應值的好壞便會間接影響搜尋的結果，假若適應值給定太小就有可能會導致太早收斂的情況發生，而得不到全體的最佳解，所以適應值的良劣將是整個基因演算法的成功關鍵所在。

(III)選擇

挑選出具有較佳特質的染色體來交配進而產生下一代而得到最終解，然而我們在選取(Selection)或複製(Reproduction)時可根據擁有較高適應值的染色體來選取，其挑選機制有輪盤式選擇、等機率法選擇、競爭式選擇。目前較為人所採用的方式為輪盤式選擇。以下將對每一種方法做簡介：

a. 輪盤式選擇：

我們先設想有一個輪盤，當輪盤中有一數值具有較大面積時被選中的機率就會相對的提高，因此輪盤式選擇的做法便是對所有染色體的適應值做一個加總在分別對單一染色體佔所有值的高低分配出相對應的權重，權重愈高被選中的機率就愈高，這麼做的好處就可以選出較好的染色體來產生下一子代，就可保證每一子代都是由最佳特質的母代所產生而不會遺失任一個可能的最佳解，相對其缺點也是有可能會因為每一次都是選出適應值高的染色體出來，進而大量複製交配，而造成群體過早收斂的情形發生。

b. 等機率法選擇

等機率法顧名思義就是分配每一染色體都擁有平等的機率會被選中，不管染色體的適應值高低皆有相同的機會被選中，然而這樣做就有相當大的可能會錯失任一次具有較高適應值的染色體，而導致太早收斂不能得到最佳解。

c. 競爭式選擇

競爭式選擇就是直接從所有的染色體中隨機選出幾個染色體出來，再由這此染色體來進行競爭最終得勝者就是雙親，在由雙親來進行演化，而不是使用機率的方式來選擇。因為染色體都是用隨機的方式來產生，這樣一來就有可能會發生和輪盤式選擇一樣有可能會造成最佳適應值的染色體沒被選上，因此就有一種精英挑選的策略，以確保最佳的染色體一定會被選上。

(IV) 交配

選擇、交配、突變可說是整個基因演算法的主要運算部份，不管缺少那一部份都不行，所以當我們選出兩個母體要來做交配以產生下一子代時，需要給定何種交配方法，如單點交配、雙點交配、多點交配、均一交配，使用者可依需求自行挑選出所需的交配方式。許多研究指出兩點交配方式優於單點交配方式，只有當群體接近收斂時。兩點交配的效果才會較差於單點交配方式，而多點交配方式和均一交配方式有極差的效果，只有在一些特殊的情況下才能有好的執行效率。均一交配方式因不會受到染色體編碼位的影響，但可能使的子代染色體結構受破壞，而無法保留雙親的特性。

(V) 突變

為了避免太早過於收斂使得無法得到最佳解，因此我們便對染色體給定一個突變率，

當達到突變率時便會對染色體的基因做突變，已防止過早收斂。至於突變率的給定方式可依問題所需而給予不同的機率，可強制每幾個染色體就突變一次或是高於突變率值時才做突變的動作。同時突變率也不用給定的太大，否則就會一直在運算而無法收斂，如此一來就無法得知最佳解為何。

(VI) 終止條件：

整個基因演算法達到以下幾個條件時，便可結束流程。

a. 當搜尋到指定的最大代數。

b. 達到收斂及飽和的狀態。

c. 找到目標最佳解。

2.2 形變簡介

二維形變方式通常都是由一張圖片要變換成另外一張圖時產生漸變效果的一種技術 [6][7]，將兩張圖片給予代表其圖片的特徵點 [8]，能讓圖片在漸變時保有其特徵資訊，而不會在過程中失去一致性。N. Arad [9]提出的便是由使用者在影像上圈選出特徵點，然後再對特徵點的起始位置和終點位置代入 Radial Basis Function 中，以求出對應的轉換矩陣，再算出其非特徵點的位置，來完成變形的效果。在二維形變的應用中，S. Neely [10]兩張影像之間做形變，F. Galpin [11]在 video 上的壓縮，便是將每個像素的灰階值做內插，但所產生的影像結果可能會有重疊和不平滑的情況發生。

三維形變方法是指一種將多邊型轉換成另一個不同的多邊型產生連續變化效果的技術 [12][13]。Kent [14]在他提出的演算法中提到，如何使兩個不相同的多邊型上的點彼此建立起關聯性，並且利用這些關聯點的資訊來讓在做連續變化的過程中能有一致性的效果。在 3 維模型變形中，A. Lee [15][16]，將原始兩物體之間對映關係的尋找，將兩物體分別簡化至粗糙的物體，再將兩粗糙的物體，找出對應關係。然後將原先 (source) 粗糙物體投影至目的 (target) 粗糙物體最初的對應，再以 Relaxation 的方式將目的物體的來源點作最佳化的調整以得到兩個模型物體的對應關係。

Marc Alexa [17][18]所提出的一套形變方法，是將兩個屬性 (genus) 為 "0" 的多面體做合併 (merge)，其做法是先將兩個屬性為 "0" 的物體分別映射到一個單位球上，然後再作 warping 的動作以對齊球上的特徵點，之後再將兩顆單位球作合併 (merge)，以產生多面體之間的對應關係。T. Michikawa [19]所提出的一個多階層

式架構的演算法，首先先由使用者在輸入的模型中，選出一些頂點，用來建立基礎模型，然後再透過網格細分法 (subdivision) 的技術，依序建立出其他階層的模型。此方法可避免使用合併的動作所衍生來的問題，但卻須由使用者建立基礎模型，對於複雜的模型通常需要很多時間來建立，對一般使用者來說並不適用。J. Parus [20] 提出了一套簡單的方法，主要是將兩物體直接做合併，一般作法是先將物體轉換成 mesh 的方式，而作者則是將兩物體合併直接找出相對特徵點，再利用內差做形變。首先將兩物體做結合，然後找出交錯的點出來，在其外圍再覆蓋上一層圓圈，主要是將點投射到圓圈上，再對圈上的點做內插的動作。

形變的過程主要可分成下列三大步驟：

Step 1：分別找出來源端(source)模型和目的端(target)模型的特徵點。

Step 2：將模型特徵點轉換成 supermesh，supermesh 主要是意指兩模型間特徵點的相對關係。

Step 3：對特徵點做內插的動作，以完成形變的轉換過程。

3. 方法

在整個虛擬環境情境感知系統上我們分為兩個部份來做介紹，分別是環境偵測和表情呈現，系統架構圖如圖 3 所示。在環境偵測方面，我們是採用具有最佳解搜尋特性的基因演算法，而在表情變化方面則是採用三維形變的技術。因此，將這兩個方法結合在一起就可為遊戲添增出另一種別有樂趣的新風格。

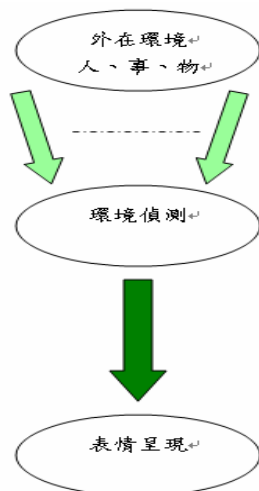


圖 3 系統架構圖

3.1 情境感知

在情境感知方面本研究是利用基因演算法所具有的搜尋功能來做為表情的產生依據，將角色人物所在的場景依所接收到的周圍環境做整個搜尋判斷，以找出最合適的表情來加以變化。首先在基因演算法部份我們主要是區分為幾個部份來完成，分別為產生初始母體、計算染色體的適應函數、複製、染色體交配、突變、下一世代選擇。

Step 1：產生初始族群

族群的產生我們所採用的染色體編碼是非二進位編碼方式，以符號來做為每一個染色體的基因，亦就是每一個符號都是代表一種可能的表情，主要根據人物的所在環境先做分類，依人、事、物三種關係來對週遭環境做規劃，如圖 4 所示，再針對所有可能的環境設定出其相對應的表情，如表 1 所示，如此一來便可對整個虛擬環境做出分類。

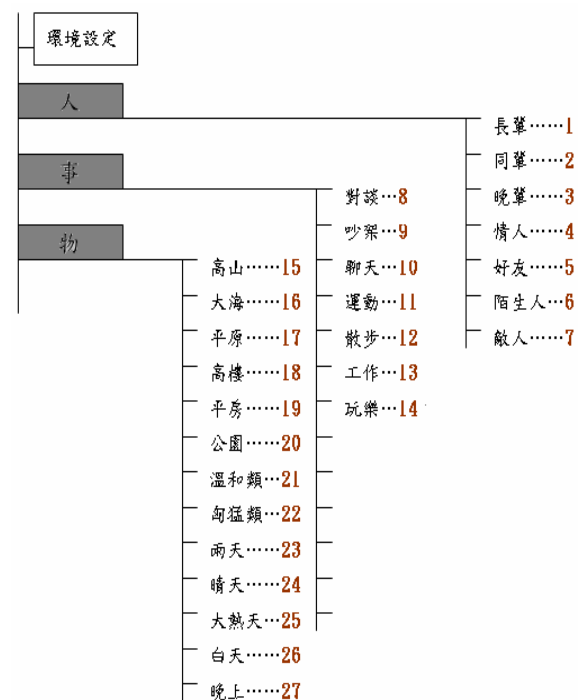


圖 4 環境設定

編號	表情名稱	種類	分類
1	嚇	IND	26
2	害怕	GRI	26
3	大笑	HAP	15
4	愉快	LIK	4, 10, 28
5	生氣	IND	9
6	哭	GRI	9
7	高興	HAP	2, 5
8	微笑	LIK	3, 7, 9
9	不耐煩	IND	27
10	悲傷	GRI	9
11	煩躁	IND	27, 29
12	憂慮	GRI	26
13	害羞	LIK	4
14	驚訝	IND	19
15	沉默	GRI	13

IND：怒，GRI：哀，HAP：樂，LIK：喜

表 1 表情分類

然後針對角色人物所在環境可能產生的表情，選出染色體，染色體長度為六，並以亂數分佈的方式分配六個角色人物的表情代碼如圖 5 所示，並對這六個場景做搜尋以找出最佳解出來。

F_1	F_2	F_3	F_4	F_5	F_6
-------	-------	-------	-------	-------	-------

圖 5 染色體示意圖

Step 2：適應函數設定

適應值的設定是依據表 1 中每一個表情間的關聯性來定義，而每個表情彼此間的相對關係透過公式(1)可求得如表 2 所示的相似矩陣。其表情權重關係式為 $w_i > w_g > w_h > w_l$ ，根據公式 (2) 可計算出 $w_i (k=1) = (4+1-1)/(4+1) = 0.8$ ，同理可知 $w_g (k=2) = 0.6$ ， $w_h (k=3) = 0.4$ ， $w_l (k=4) = 0.2$ ，故 f_1 和 f_2 的關相性為：
 $f_{12} = 1 \times (0.8 + 0.6 + 0.4 + 0.2) / 4 + (0.8 \times 0 + 0.6 \times 0 + 0.4 \times 0 + 0.2 \times 0) = 0.5$

	1	2	3	4	5	6	7	8	9	10	...
1	0	A	A	A	B	A	A	A	B	A	
2		0	A	A	A	C	A	A	A	C	
3			0	A	A	A	D	A	A	A	
4				0	A	A	A	E	A	A	
5					0	A	A	A	B	A	
6						0	A	A	A	C	
7							0	A	A	A	
8								0	A	A	
9									0	A	
10										0	
...											

其中 A=0.5，B=1.3，C=1.1，D=0.9，E=0.7

表 2 相似矩陣

$$\sum_{n=1}^6 f_{i,j} = f_{\alpha} + f_{\beta} \quad (1)$$

$$f_{\alpha} = f_{i,j} \times (w_i + w_g + w_h + w_l) / 4$$

$$f_{\beta} = w_i \times i_{i,j} + w_g \times g_{i,j} + w_h \times h_{i,j} + w_l \times l_{i,j}$$

其中

$f_{i,j}$ ：第 i 個和第 j 個表情關聯性，若權重

$i > j$ ，則 $f_{i,j} = 1$ ，否則 $f_{i,j} = 0$ 。

w_i ：怒之權重

w_g ：哀之權重

w_h ：樂之權重

w_l ：喜之權重

$i_{i,j}$ ：怒之關係式，若種類 $i = j$ ，則 $i_{i,j} = 1$ 。

$g_{i,j}$ ：哀之關係式，若種類 $i = j$ ，則 $g_{i,j} = 1$ 。

$h_{i,j}$ ：樂之關係式，若種類 $i = j$ ，則 $h_{i,j} = 1$ 。

$l_{i,j}$ ：喜之關係式，若種類 $i = j$ ，則 $l_{i,j} = 1$ 。

$$w_x = \frac{(n+1-k)}{(n+1)} \quad (2)$$

w_x ：第 x 個表情參數，

k ：表情權重大小之排名， $1 \leq x \leq 4$

n ：表情分類之個數

Step 3：選擇

由一開始的染色體中算出適應值最高的染色體來做為交配的因子，並執行菁英法則，保留適應性最高的染色體，主要是為了將有可能是最佳解的染色體保留到下一次繼續做交配，以達到最佳解，並可避免整個運算過程太

早進入收斂，而導致結果不正確。其染色體的計算方式根據公式(1)中 $\sum_{n=1}^6 f_{i,j}$ ，即可得知每一個染色體的適應值大小。每次都從母代族群中選出最佳的染色體，再加上菁英化的選擇方式即可確保每一次都是最佳化選擇。

Step 4：交配

在這裡我們是使用最大關聯性保留方式的動態交配，以確保染色體可維持在最佳狀態下。將染色體內的基因依公式找出具有最大關聯性的兩個區塊保持不變，並對其他的區塊做交配。

其區塊保留是指將染色體中任兩個具有最大關聯性的區塊保留不變以做為交配點，在將剩餘的染色體重新分配，如圖 6 所示。

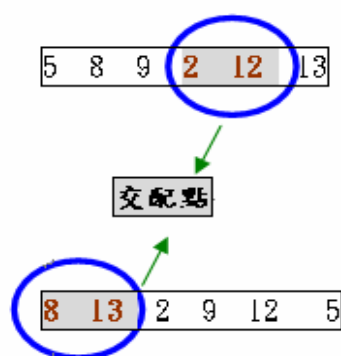


圖 6 區塊保留交配

當確定交配點之後，以不重覆為原則對所要交配對向進行染色體重填。將染色體 parent 1 交配點區塊內的數值複製到子代 child 1 中的相對應位置上，並從染色體 parent 2 中刪除 parent 1 保留區塊中的數值，依據染色體 parent 2 前後順序複製到染色體 child 1，如圖 7 所示。同理將染色體 parent 2 交配點內的保留區塊複製到染色體 child 2，在將染色體 parent 1 中的基因複製到染色體 child 2。

為了避免過早收斂或錯失可能的最佳解情況下，只要發生交配後產生的子代具有相同適應值時即對染色體做突變。其突變設定為交配的兩個染色體中關聯相似度最小的兩基因，即針對各個染色體中最小關聯性的基因為切點做交換，如圖 8 所示。染色體 parent 1 中 $f_{8,2}$ 和染色體 parent 2 中 $f_{12,5}$ 之相似度為最小以此兩染色體為切點各做突變。

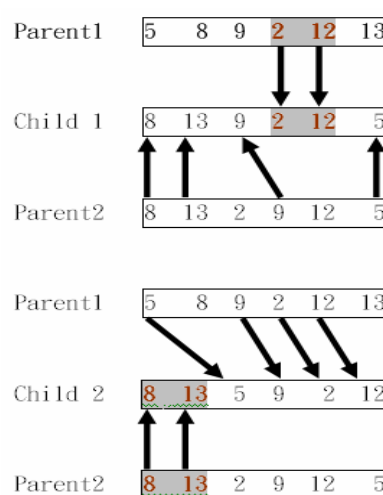


圖 7 基因交配示意圖

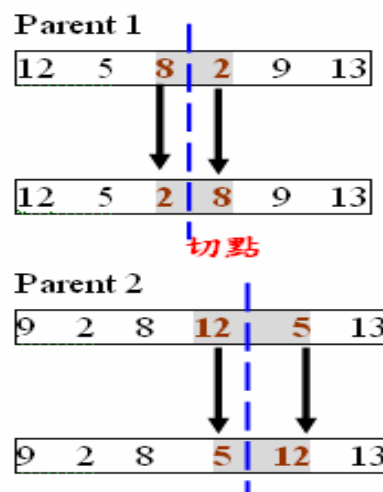


圖 8 突變之基因交換圖

Step 5：終止族群演化

經過 h 代演化之後，在這裡我們設定最高為 500 代演化。有些染色體的適應值會趨向一致進而達到收斂，而不會有更好的適應值出現，因此終止搜尋，若否，則回 step 2 繼續執行演化下一世代。

3.2 三維形變：

在三維形變上我們提出一個簡單且有效率的方法，來將兩個不同表情的人臉做形變以達到漸變的效果，使用以本文提出的 Ripple Search Algorithm 找尋特徵點為基礎並且以 semi-automatic 比對對應點作法，最後利用 Bezier curve 來做內插，實現 forward warping 的效果。

首先我們定義一個符合 Euler-valid topology 的幾何物件，多邊型的 structure 必須

包含剛好三個變數，並指定到其頂點，即是三角形三點成一面的道理。使用這個方式，不同編號的多邊型指定到同一個頂點，這對於系統的設計有幫助，多邊型上的每一個三角面皆是由圖 9 所示的結構所構成。

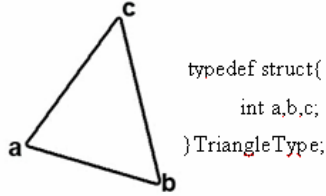


圖 9 三角面結構

本方法一共分為五個步驟：

Step 1.Head decomposition: front and back patches

將一個人頭的 3D mesh 物件，延著 X 軸一分為二，分別命名為 front patch 以及 back patch，如圖 10 所示，將 front patch 做為我們臉部表情開發的基本物件，使用 front patch 來創建所需要的表情，在不破壞性的修改前提下，產生 M_s 以及 M_t 兩個人臉表情物件。

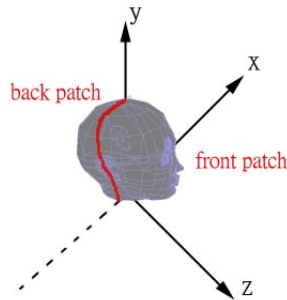


圖 10 頭部分解

Step 2.Parameterization: front patch

將 M_s 以及 M_t 兩者的中心與 unit sphere 的球心重疊。並利用公式(3)將 mesh 上的每一個點投影到 unit sphere 球面上。如圖 11 所示，可以發現兩個 mesh 的形狀不完全相似。本研究可以根據這兩個 mesh 的 topology 資訊來做形變。

$$\frac{\overrightarrow{OV_s} \cdot \vec{R}}{\|\overrightarrow{OV_s}\|} = (v_s)_p \quad \frac{\overrightarrow{OV_t} \cdot \vec{R}}{\|\overrightarrow{OV_t}\|} = (v_t)_p \quad (3)$$

$\overrightarrow{OV_s}$:圓心到 source 頂點的向量。

$\overrightarrow{OV_t}$:圓心到 target 頂點的向量。

R : 球的半徑。

$(v_s)_p$: source 投射之後的頂點位置。

$(v_t)_p$: target 投射之後的頂點位置。

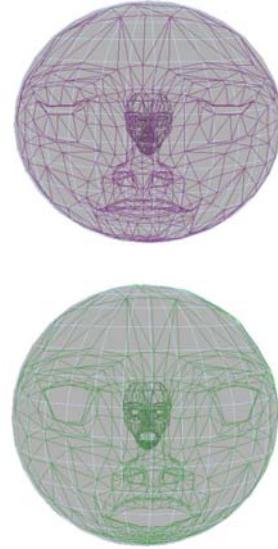


圖 11 物件投射至球上

Step 3.Patch with feature points

本步驟的特徵點是由使用者所點取，當 M_s and M_t 投影到 unit sphere's surface 上時，如圖 12 所示，可以發現彼此不相同的 mesh topology 將會很明顯地被突顯出來。由使用者針對想要對照的區域來做選取。當點選到該點半徑碰撞範圍時，該點會被標記起來，並且存進 P_f 。而另外一個物件會藉由此物件所存取的多邊型編號來比對，使用 semi-automatic 作法，同時也會被標記起來，搭配論文中所要提出的『Ripple Search Algorithm』便會產生如圖 13 所示的結果。

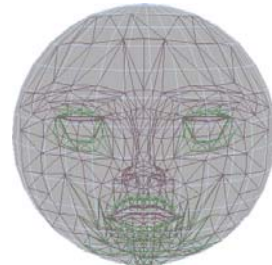


圖 12 Spherical projection of M_s and M_t

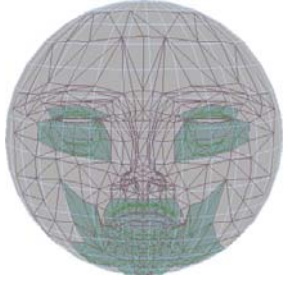
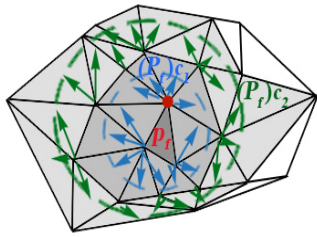


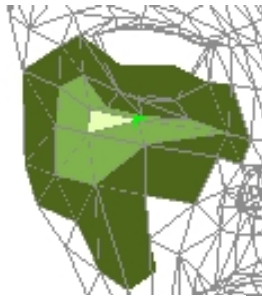
圖 13 The different fragment
between M_s and M_t

Step 4. A correspondence between the source and target patches

此小節將介紹如何找尋對應點，本研究使用『Ripple Search Algorithm』，我們利用所定義的 topology 來根據每個三角面的點 a、點 b、點 c，放入 candidate list，做點座標比對，如此會找到 P_f 第一圈所符合的多邊型，將這些多邊型編號存入 $(P_f)c_1$ 矩陣中。再將 P_{f-1} 中每個多邊型進行第二次比對，如此將會有第二圈被標示起來，並存入 $(P_f)c_2$ 矩陣。每點一個特徵點就根據此『Ripple Search Algorithm』如圖 14 所示，就可以找出該點周圍的兩圈對應點，並且 M_s 與 M_t 將會把本演算法所找尋到的資訊使用 semi-automatic 應用在下一節所要介紹的內插，即可以達到臉部表情形變的效果。



(a) 示意圖



(b) 執行結果

圖 14 Ripple Search Algorithm

Step 5. Interpolation of corresponding vertices

將原始物件與目標物件相異的對應點作貝茲內插，原始物件為起始點，目標物件終點，將點的位置慢慢轉移到目標物件的位置。

$$B(u) = \sum_{k=0}^n P_k \frac{N!}{k!(N-k)!} u^k (1-u)^{n-k} \quad \text{for } 0 \leq u \leq 1 \quad (4)$$

本論文選用 Bezier curve 公式(4)來作為內差，原因在於 Bezier 曲線的平滑度與可改變的控制點，一開始套用兩個控制點的 Bezier curve，用起始點 v_s 與終點 v_t 和中間的混合點 v_a 來定義出 morphing 的內插公式(5)，此時的內插仍為線性。

$$v_a = (1-a) \times v_s + a \times v_t \quad 0 \leq a \leq 1 \quad (5)$$

接著將 a 以 3 個控制點以上的控制點帶入 Bezier curve 的公式，如此一來能讓 a 值以曲線方式成長，而本論文所做的實驗則以 4 個控制點 (P_0, P_1, P_2, P_3) 來做控制，如公式(6)，在將 a 套到公式(5)，便能在 morphing 時有較自然的表情產生。必須注意的是起始點與終點的數值， P_0 必須為 0， P_3 必須為 1，才能確保所有的 P_f 都能位移到 P_c 的位置。

$$\begin{aligned} a &= P'_0 + P'_1 + P'_2 + P'_3 \\ P'_0 &= P_0 * (1-i)^3 \\ P'_1 &= P_1 * 3 * i * (1-i)^2 \\ P'_2 &= P_2 * 3 * i^2 * (1-i) \\ P'_3 &= P_3 * i^3 \end{aligned} \quad (6)$$

其中 $0 \leq i \leq 1$

其中 a 是一個變動參數，隨著時間遞增，以達到 mesh 漸變的目的，而遞增的曲線度為使用者自定(本論文實驗設定是四個控制點 0,0.2,0.9,1)，越多控制點，就能讓表情在 morphing 過程中有越多不同的速度感，看起來較為生動。具有 $n+1$ 個控制點的內插公式(7)，同樣的， P_0 與 P_n 也必須為 0 與 1。

$$a = \sum_{k=0}^n P_k \frac{n!}{k!(n-k)!} i^k (1-i)^{n-k} \quad 0 \leq i \leq 1 \quad (7)$$

4. 實驗結果

實驗包含兩個部份，分別為情境感知和三維形變，情境感知方面本研究是使用具有最佳解搜尋特性的基因演算法針對場景部份所做的搜尋分析，三維形變上則是使用本研究所提出的 Ripple Search 來對人臉表情部份做特徵點的對應方式。

圖 15 所示為利用基因演算法所需的搜尋時間，以每一百代演化為一個單位所做的比較，平均一百代演化的所需時間大約為一秒鐘。

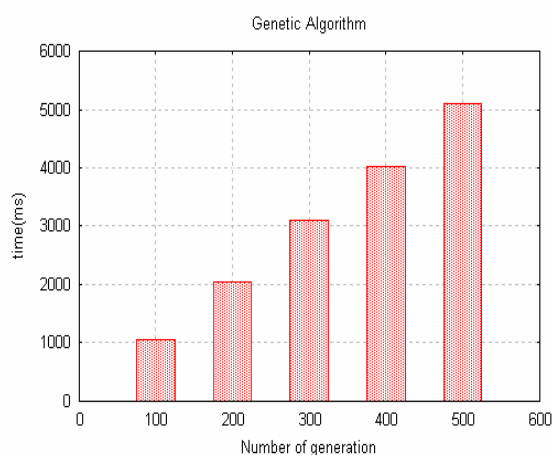


圖 15 演化時間

圖 16、17 所示為人物臉部表情之三維形變方式。圖 18 所示為 Polygons 與 feature points 數量以及 elapsed time 相對關係。



圖 16 三維形變方式



圖 17 三維形變方式

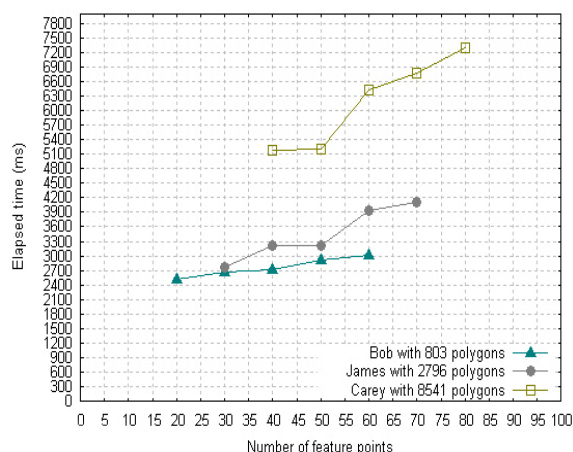


圖 18 特徵點數量和 elapsed time 關係圖

5. 結論

本篇論文描述在虛擬環境中人物臉部表情可以動態的表現而不是使用靜態的轉換。在虛擬環境中人物的臉部表情的呈現方式總是利用靜態的材質貼圖方式來表現，於是我們利用三維形變的技術並對虛擬環境偵測選出一個最合適的表情，使得人物的臉部表情可以更加生動、活潑；我們採用 morphing between libraries of key poses 技術，並結合在網路行動通訊領域中已廣為採用的 context-awareness, location-based service 及 sensor networks 技術，便可延伸應用在電腦遊戲場景中進行情境感知判斷，情境感知模型採用的是 Interaction Constraints Model，藉由在已完成的三維電腦場景中嵌入軟體感應器，把環境參數輸入人物角色的軟體接收器中，在 library of key poses 表情資料庫中分析擷取出適合此刻情境的人臉表情 mesh，以快速即時的 3D mesh morphing 技術進行人臉表情的網格形變，並配合階段式形變技術讓人臉表情生動不僵硬。

6. 誌謝

本論文由國科會計畫編號 NSC95-2221-E-415-015 補助辦理執行。

參考文獻

- [1] Kerlow, I. V., *The Art of 3D Computer Animation and Effects*, 3rd ed., John Wiley & Sons, Inc., 2003.
- [2] Burgy, C., Garrett, J., "Situation-Aware Interface Design: An Interaction Constraints Model for Finding the Right Interaction for

- Mobile and Wearable Computer Systems,” In *Proceedings of International Symposium on Automation and Robotics in Construction*, pp.563-568, 2002.
- [3] 林豐澤, “演化式計算上篇：演化式演算法的三種理論模式”, 文化大學用數學系。
- [4] 林豐澤, “演化式計算下篇：基因演算法以及三種應用實例”, 文化大學用數學系。
- [5] 林峰田、黃衍明, “基因演算法之基本概念、方法與國內相關研究概況”, 國立成功大學建築研究所博士班。
- [6] Heckbert, P. Master’s thesis, “Fundamentals of Texture Mapping and Image Warping,” UCB/CSD 89/516, CS Division, U.C. Berkeley, 86pp, June 1989.
- [7] Blanes, J. S., “Scanner morphing simulation with image warping,” In *Proceedings of the 14th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision’2004*, pp.233-240, 2006.
- [8] Beier, T., Neely, S., “Feature-based image metamorphosis,” In *Proceedings of SIGGRAPH*, Vol. 26, No. 2, pp. 35-42, 1992.
- [9] Arad, N., Reisfeld, D., “Image Warping using Few Anchor Points and Radial Functions,” *Computer Graphics Forum*, Vol.14, No. 1, pp.35-46, 1995.
- [10] Nelly, S., “Feature-Based Image Metamorphosis,” In *Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques*, Vol. 26, pp.35-42, 1999.
- [11] Galpin, F., Balter, R. and Deguchi, K., “3D Models Coding and Morphing for Efficient Video Compression,” In *Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition(CVPR 2004)*, pp.334-341, 2004.
- [12] Alexa, M., “Recent Advances in Mesh Morphing,” *Computer Graphics Forum*, Vol. 21, No. 2, pp.173-196, 2002.
- [13] Alexa, M., “Local Control for Mesh Morphing,” In *Proceedings of the International Conference on Shape Modeling & Applications*, pp.209-215, 2001.
- [14] Kent, J. R., Carlson, W. E. and Parent, R. E., “Shape Transformation for Polyhedral Objects,” *Computer Graphics*, Vol. 26, No. 2, pp.47-54, 1992.
- [15] Lee, A., Sweldens, W., Schorder, P., Cowsar, L. And Dobkin, D., “MAPS: Multiresolution Adaptive Parameterization of Surfaces,” In *Proceedings of SIGGRAPH’98*, pp.94-104, 1998.
- [16] Lee, A., Dobkin, D. and Sweldens, W. and Schroder, P., “Multiresolution Mesh Morphing,” In *Proceedings of SIGGRAPH’99*, pp.343-350, 1999.
- [17] Alexa, M., “Merging Polyhedral Shapes with Scattered Features,” *Visual Computer*, Vol. 16, No. 1, pp.38-46, 2002.
- [18] Alesa, M., “Differential Coordinates for Local Mesh Morphing and Deformation,” *Visual Computer*, Vol. 19, No. 2, pp.105-114, 2003.
- [19] Michikawa, T., Kanai, T., Fujita, M. and Chiyokura, H., “Multiresolution Interpolation Meshes,” In *Proceedings of 9th Pacific Graphics International Conference (Pacific Graphics 2001)*, pp.60-69, 2001.
- [20] Jindřich PARUS, Master Degree Programme. Dept. of Computer Science and Engineering.