

以特徵線為基礎的影像變形方法研究 A Study on Feature-based Morphing Techniques

林永青
Yung-Ching Lin

李同益
Tong-Yee Lee

孫永年
Yung-Nien Sun

成功大學資訊工程研究所
Department of Computer Science and Information Engineering,, National Cheng-Kung University
Tainan, Taiwan, ROC

Abstract

Image-morphing between two images over time is a useful visual technique. In 1992, T. Beier and S. Neely developed a feature-based morphing technique offering natural transitions between images. However, their technique suffers from highly computational cost. In this paper, we propose several techniques to improve its computation efficiency but without expense of image quality too much. Our preliminary results show that our best performance (by the weighted scheme) is faster than that of T. Beier et al.'s scheme by a factor of 20.

Keyword: image morphing, scan-line, subdivision, weighted

摘要

影像變形(image morphing)是一種可以在兩張影像間做出平滑轉換的技術，在 1992 年，T. Beier 和 S. Neely 發表以特徵點為基礎的影像變形[1]，可說是這研究領域中的重要里程碑。不過其缺點在於處理變形時以整體方式(global)來考慮各種特徵點的影響，花費大量的計算時間，本篇論文主要是以此論文為基礎對影像變形的的方法作一個分析，探討其過程，並提出改進效能的方法。實驗結果顯示我們提出的方法對我們的測試影像，最佳可由原本 T. Beier 的方法 40 秒降低到 2 秒，增快了 20 倍之多，然而影像的品質與原先方法相差不多。

1. 簡介

近年來，在兩張影像間作影像變形已在各期刊間被廣泛的討論，並且已經有很多方法被提出來([1]-[6])，而且廣泛的應用到各方面，或許您對電影中狼人的變身還不陌生，在麥克傑克森黑與白的 MTV 中，電腦圖學影像變形技術已被這些電影製片家所採用，這些驚人的

效果，使得影像變形漸漸受到高度的重視，本實驗室一直致力於 3D 立體影像快速顯示與虛擬手術的研究，精確且快速的顯示技術一直是我們所追求的目標，在本篇論文中，我們嘗試研究影像變形的技術，將來準備應用到我們的手術模擬系統中。簡單的例子如巨大體積資料(large volume data)的快速顯示，通常計算一張影像，利用傳統光跡投射法(ray-casting)，時間可能需要數秒甚至數分鐘，因此若要達成交談式操作(interactive manipulation)這些資料。實在有些困難，但是若我們首先計算與預存一些特定角度(view-point)圖像，然後在交談當中，利用適當視角(view-point)內差技術，如視角變形(view-morphing)[3]，再配合本文所提的快速影像變形技術，可大大地減少計算時間，而達到交談式的效果。本文所提出的方法，很容易的就可以延伸至處理 3-D 資料的變形上，如處理肌肉體積資料，受力時會產生局部的收縮變形的模擬。如何提高速度及兼顧影像變形所得品質一直是本文所努力的方向，在此我們介紹 feature-based image metamorphosis[1]的方法，然後提出我們的改進方法，試著從實驗結果分析探討改進的原因。

2 影像變形的的方法

本論文乃是討論改善以特徵點為基礎的影像變形的的方法，所以在這個章節中我們首先簡單的介紹一下以特徵線為基礎的影像變形的的方法並且提出以這個方法為基礎的一些改進方法，2.1 簡單的回顧一下 T. Beier 和 S. Neely 所提出的影像變形的的方法，2.2 介紹以掃描線演算法來改進速度，2.3 介紹以區域分割來改進速度，2.4 介紹考慮權重(weight)來加速。

2.1 以特徵線為基礎的影像變形

在這個小節裏我們將簡單的介紹 T. Beier 和 S. Neely 所提出的影像變形的的方法。2.1.1 說明單一特徵線的扭曲，2.1.2 說明多對特徵線的扭曲。影像變形的步驟，如圖十所示，首先有

兩張原始影像，在兩張影像中選取一些互相對應的特徵線，以內差算出兩張影像中特徵線的位置，所得到的內差特徵線的位置的影像我們稱作目的影像，然後兩張原始影像分別扭曲到目的影像，再分別對兩張扭曲過的影像作混合(Blending)，這裡的混合乃是灰階值的內差，最後可得到我們所要的介於兩張影像作平滑轉化的中間影像。

2.1.1 單一特徵線的扭曲

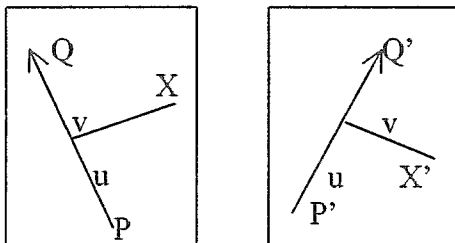
在來源影像中的一對特徵線段，有對應關係到目的影像中的另一特徵線段，則其對應關係如圖一，本圖中為了解決子點(subpixel)的問題，所以採用逆向對應關係，如圖一，目的影像中的 X 點，將被扭曲到來源影像中的 X' 點。

$$u = \frac{(X - P) \cdot (Q - P)}{\|Q - P\|^2} \quad (1)$$

$$v = \frac{(X - P) \cdot \text{Perpendicular}(Q - P)}{\|Q - P\|} \quad (2)$$

$$X' = P' + u \cdot (Q' - P') + \frac{v \cdot \text{Perpendicular}(Q' - P')}{\|Q' - P'\|} \quad (3)$$

Perpendicular 代表依照順時鐘方向旋轉九十度。



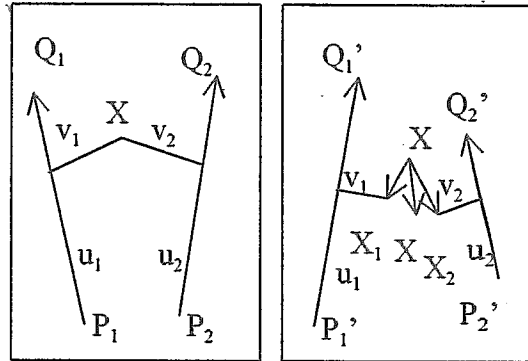
目的影像 來源影像
圖一：單一特徵線

2.1.2 多對特徵線的扭曲

在實際的影像中，物體或場景並非簡單一條特徵線可代表其輪廓特徵，因此我們以多對特徵線來趨近物體的輪廓，所以我們必須要能應付多對特徵線的扭曲對應關係，在多對特徵線的對應關係中，每對特徵線皆應被考慮，目的影像中 X 點分別針對每對特徵線的影響而被扭曲到來源影像位置 X_i，D_i=X_i-X，所以其每個對應點皆佔一定權重，權重函數為：

$$\text{weight} = \left[\frac{\text{length}^p}{(a + \text{dist})} \right] \quad (4)$$

length 為特徵線的長度，dist 為該點到特徵線的距離，a,b,p 是用來控制特徵線的效應。



目的影像 來源影像

圖二：多對特徵線

多對特徵線的演算法如下：

```

For each pixel X in the destination
  DSUM = (0,0)
  Weightsum = 0
  For each line PiQi
    Calculate u,v based on PiQi
    Calculate Xi' based on u,v and Pi'Qi'
    Calculate displacement
    Di = Xi' - X for this line
    dist = shortest distance from X to PiQi
    weight = (lengthp / (a+dist))b
    DSUM += Di * weight
    weightsum += weight
  X' = X + DSUM / weightsum
  DestinationImage(X) = sourceImage(X')
```

2.2 掃描線演算法 (scan-line algorithm)

考慮上面 2.1 所提的方法，在計算多對特徵線作扭曲時，影像中每一個點皆必須對每對特徵線作扭曲轉換，然後在根據它們所佔的權重來結合計算其真正位置，如此一來所需的計算將是十分費時的，因此我們在這篇論文中試著針對速度提出改進的方法，我們觀察上述數學式(1)(2)(3)，在只考慮單一對特徵線的影響下，無論其對應的特徵線經過位移，旋轉，縮放，發現其對應關係依舊是線性的，亦即影像中的一條直線，經過數學式(1)(2)(3)扭曲轉換的運算以後還是一條直線，如圖三，利用這個幾何特徵，我們可以一次以一條掃描線為單位，引用電腦圖學中的掃描線演算法(Scan-line conversion)，比照多對特徵線的演算法，掃描線演算法主要在加速步驟一的計算過程，而只

要針對掃描線起點和終點以數學式(1)(2)(3)求得正確位置，其中線上的每個點再以快速掃描線轉換演算法求得，步驟二會依照所有利用掃描線演算法所得到的對應點，依照不同權重(weight)，給予結合，此種結合關係並不是線性關係，可能會扭曲成曲線(curve)，參考圖四。因此本方法無法針對步驟二加速。假如現在影像大小 512×512，原本 512×512 個點皆要經過數學式(1)(2)(3)的運算，現在只需要 512×2 個點需要(1)(2)(3)的運算，而剩下的點皆由掃描線演算法快速求得，所以必可達到加速的效能。以下，我們將證明一條特徵線的影響是線性的關係。

假設 A, B 為掃描線的兩端點，C 為掃描線上任意的一點，在單一特徵線的扭曲中，若 A, B, C 變形的位罝分別為 A', B', C' 且

$$C = (1 - s)A + sB \quad s \in (0, 1) \quad (5)$$

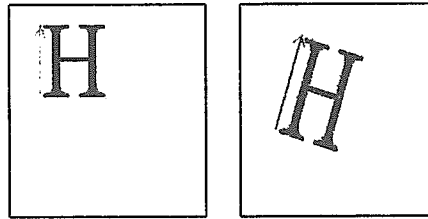
由 2.1.1 數學式(1)(2)(3)得知

$$\begin{aligned} C' &= P' + u \cdot (Q' - P') + \frac{v \cdot \text{Perpendicular}(Q' - P')}{\|Q' - P'\|} \\ &= P' + P' - P' + \frac{(A - sA + sB - P + P - P) \cdot \text{Perpendicular}(Q - P)}{\|Q - P\|^2} \cdot (Q' - P') \\ &\quad + \frac{(A - sA + sB + P - P + P) \cdot \text{Perpendicular}(Q - P) \cdot \text{Perpendicular}(Q' - P')}{\|Q - P\| \cdot \|Q' - P'\|} \\ &= A' - sA' + sB' = (1 - s)A' + B' \end{aligned} \quad (6)$$

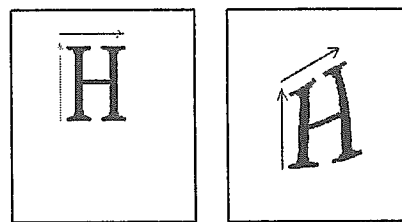
由上述(5)(6)我可證明得知數學式(1)(2)(3)有線性的關係

2.3 改良式的區域分割演算法 (Subdivision Algorithm)

本章節所提的改進方法，乃是參考以特徵點為基礎的體積變形[2]中所提到的最佳化方法，稱作區域分割，並加以改良進，觀察 2.1.2 中數學式(4)我們所使用的權重函數，就如數學式(4)所表示，權重大小和距離的 b 次方成反比，亦即他們所受到的權重與其位置有關，而且鄰近位置的所受到特徵線的扭曲力是相似的，利用影像中的點所受的扭曲力具有區域一致性(coherence)的特徵，發展出區域分割演算法，這個演算法以方形為區域單位，偵測方形中所受的扭曲力是否為均勻，偵測的方式為以 2.1.2 多對特徵線的演算法精確計算出四個頂點的位置，並在方形中選取幾個樣品點，分別以 2.1 提出的方法計算其精確位置，比較利用四個頂點以內差方式求得的估計位置，若這些樣本點兩種方法所求得值的差小於某個閾值



圖三：左圖影像在單一特徵線經過平移，旋轉，放大後的扭曲得到右圖，由圖可發現此一轉換為線性的。

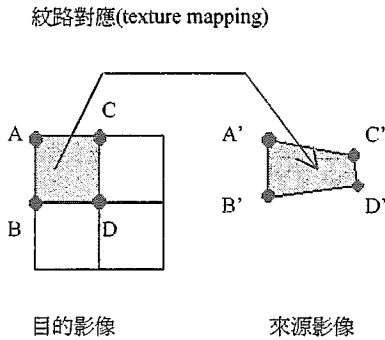


圖四：左圖影像在多對特徵線經過平移，旋轉，放大後的扭曲得到右圖，由圖可發現此一轉換為非線性的。

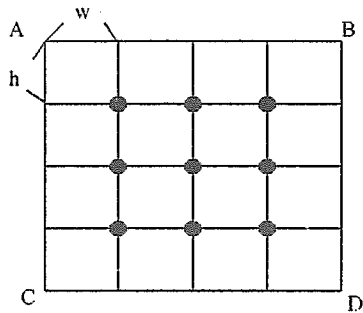
(threshold) T，那麼代表這個區域所受到的扭曲力很平均，所以這個區域所有點的扭曲位置以四個頂點位置以紋路對應(texture mapping)方式快速求得，如圖五若是有一個點的差值大於閾值(threshold)，那麼代表這個區域所受的扭曲力不平均，所以我們以遞迴的方式繼續將這個區域均分為四個等分區域，然後每個區域再重複上述的動作。請注意，如果有硬體可以支援紋路對應，本方法可以執行地十分快速。

我們所提出的改良式區域分割法，著重在有關於樣本點的選取，我們所選取的樣本點首先要能代表整個區域，而且最好能減少不必要的重複計算，所以我們所選取的樣本點如圖六，因為樣本點必須以 2.1.2 所提出的多對特徵線演算法步驟計算精確位置，所以十分費時，圖五中九個黑點所代表的樣本點，均勻廣佈在此區域中，故能代表此區域，再者因為方格的區域的頂點，若需要細分到該區域時，則必須依照 2.1.2 所提的的步驟計算，圖六大區域 ABCD 中每個小方格面積皆為 w×h，若大區域內所受的扭曲力不均勻，則那些黑點皆是以後所分出來的小區域的頂點，我們在之前測試樣本點時已經求過一次精確位置，此時當作頂點亦將需要精確位置，因此求過精確位置的樣本點若儲存起來，將來用到的機會很大，可以省下重複的複雜運算。值得注意的是這個演算法的加速是會造成誤差的，所以控制閾值 T 的大小取得速度及影像品質最佳平衡點亦非常重要，通常這個閾值我們設為二，即兩個色素(pixel)距離。

本方法，可以預期的是，當變形的程度相當劇烈時，本方法可得到相當不錯的效果，若是相當緩和，本方法的效能將回歸至原來的區域分割方法。



圖五：目的影像方形 ABCD 以紋路對應(texture mapping)對應到來源影像中的 A'B'C'D'。



圖六：圖中黑點為我們所選取的樣本點。

2.4 權重 (weight)

這個小節我們將以上一小節所提出的區域分割的方法，加上對權重的考量，提出一個更有效率的方法。

在 2.1.2 的數學式(4)中我們得知權重的大小與距離 b 次方成反比，所以我們可以把特徵線對影像中點的影響看做是磁鐵，雖然每個點皆會受到每個特徵線的影響，但若相距距離較遠，那麼其影響是微乎其微的，甚至可以忽略考慮不予計算，一個點若所考慮的特徵線愈少，由 2.1.2 所描述的演算法得知，它的扭曲精確位置的計算速度將更快，將這個觀念結合上一節的區域分割，每個區域記錄著將被考慮的特徵線，當然一開始整張影像將考慮全部的特徵線，隨這區域被分割，計算這些區域與所要考慮的特徵線的距離，當距離大於某個閾值 D ，那麼

將這個特徵線由這個區域去除，如此愈分愈小的區域所要考慮的特徵線將愈來愈少。

若原始影像中 X 扭曲到 X' 的位置， $D_i = X_i' - X$ ，參考 2.1.2 的多對特徵線扭曲演算法， $weight(i)$ 表示第 i 條特徵線的權重， N 為特徵線的總數。

$$X' = X + \sum_{i=1}^N \left(\frac{weight(i) \times D_i}{\sum_{j=1}^N weight(j)} \right) \quad (7)$$

所以當有一條特徵線的權重遠小於任何其它一條特徵線，它的權重就可以忽略不計，觀察一般我們要作影像變形的影像，我們會在物體四周劃上特徵線，所以物體上的點，一定至少有一個相較之下距離比其它的小，而權重又與距離的 b 次方成反比，這也就是我們可以把距離比較遠的特徵線的影響去除掉的原因，如圖七。

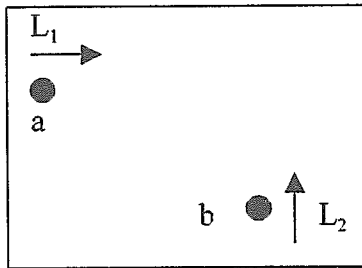
若 A_i 表示第 i 條特徵線， $dist(i)$ 為該點到第 i 條特徵線的距離，則

$$S = \{A_1, A_2, A_3, \dots, A_k, \dots, A_N\}, \text{ if } dist(k) < D$$

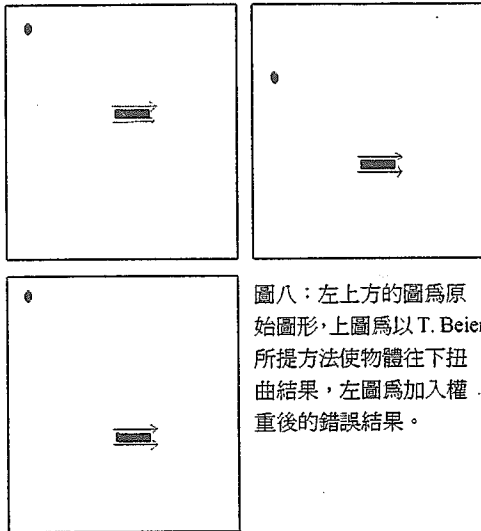
所以把太小的權重去除後，數學式(7)可用下面取代

$$X' = X + \sum_{k \in S} \left(\frac{weight(k) \times D_k}{\sum_{j \in S} weight(j)} \right)$$

但值得注意的是，若是所有特徵線都集中某一特定區域時，在計算距離所有特徵線都很遠的點時，我們的方法可能會把所有特徵線的影響去除，如此便會造成錯誤，幸好那些錯誤的點在絕大部份的應用都是無關緊要的背景點，因為大部份一般的影像變形背景都是一致且無關緊要的，所以我們的方法適用於絕大部份的案例，只有少數一些關心背景部份會造成錯誤。如圖八所示，我們關心的物體是中間的長方形，所以特徵線劃在長方形四周，我們以特徵線讓長方形往下移，T. Beier 的方法，在遠端左上方的圓形會跟著往下移，而若以我們加入權重後的方法後，圓形因距離所有特徵線都太遠而將會維持不動。造成這個現象的只要原因是，考慮數學式(7)，變形位置是各權重的相對比重而非權重的絕對值，注意將除以權重的總和，因此欲免此現象，我們可以再加入一個新的特徵線，如圖八接近圓形處，我們可以保證此圓形不被長方形四周的特徵線影響，而依然保持原來不動。



圖七：a點受 L_1 的相對權重遠比 L_2 大，故計算 a 的變形位置時，可以忽略 L_2 ，相同的在計算 b 時可以忽略 L_1 。



圖八：左上方的圖為原始圖形，上圖為以 T. Beier 所提方法使物體往下扭曲結果，左圖為加入權重後的錯誤結果。

3. 實驗結果與分析

目前我們的實作是在 Petium 133 的 PC 上，由於目前特徵線的選取仍採手動，所以使用者介面的部份亦非常重要，故我們選擇以 Visual C++ 4.0 來作為發展我們影像變形系統的程式語言，本系統的簡單介面如圖九，特徵線的拉法可用滑鼠拖曳得到，右下角狀態列顯示滑鼠指標目前在該影像中的位置，工具列的按鈕可決定特徵線的確定和取消，選單中有幾束種方法的實作，另外上有輔助工具功能，如儲存特徵線位置，由檔案載入特徵線等功能。

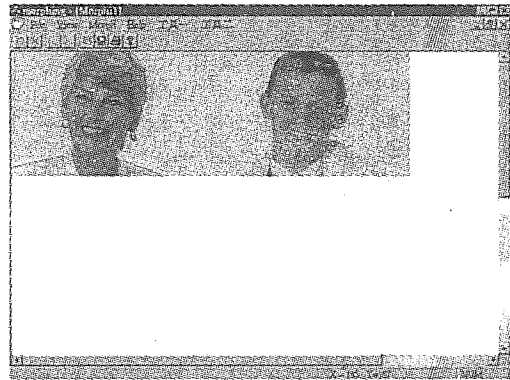
表一為我們實驗各種方法的結果，測試影像分別為圖 10.1 和圖 10.2，請注意在圖中的黑線條為特徵線。

在 T. Beier 等所提的方法中，計算時間和影像的大小及特徵線的個數成正比，若影像的長和寬為 L 和 W ，特徵線個數為 N ，則時間複

雜度為 $O(L \times W \times N)$ 。我們所得結果如圖十一左上所示，影像大小 320×200 ，共選取 22 對特徵線，至於我們參數的選取 $a=0.01, b=2, p=0$ ，所需時間為 40 秒。

	320×200 22 對特徵線
T. Beier 等所提的方法	40 秒
掃描線演算法	37 秒
區域分割演算法	15 秒
區域分割加權重去除演算法	2 秒

表一：各種方法之所需時間比較



圖九：我們實作的影像變形的系統平台

在掃描線的演算法中，實驗結果如圖十一右上所示，所需時間 37 秒，此結果並未如我們所預期一般大大的改善速度效率，原因乃是雖然我們以掃描線轉換取代數學式(1)(2)(3)的計算，但是 v 乃是點到掃描線的距離，所以計算權重時仍會用到，再則我們是用反向對應，所以兩張影像的扭曲計算中，相同的點其 u 和 v 參數是一樣的，故可避免重複計算，最重要的原因是整個轉換演算法中，如 2.1.2 的演算法所示，步驟一占的份量較小，約為 20%，這些原因使得掃描線演算法的改良效果有限，而不顯著。

在改良式的區域分割的演算法中，結果如圖十一左下所示，所需時間 15 秒，我們測試區域所受的扭曲力的閾值取 2，所需時間 15 秒，因為區域的一致性，大大減少需要計算精確位置的點數，因此而提昇了速度效能。比較此法和掃描線方法，此方法可視為以二維內差方式來進行加速，而掃描線方式為一維方式（一條掃描線），而且掃描線方式，只有考慮單一特徵線的一維內差，但是改良式區域方法，它在進行二維內差時，已經考慮到所有特徵線對於

某特定方形區域的影響進行二維內差。綜合上述，可以知道，此法比掃描線方法快了將近三倍，其實是有跡可尋的。

在區域分割加上權重考量的演算法中，所得結果如圖十一右下所示，我們取距離超過閾值 $D=30$ 即不考慮特徵線的影響，所需時間為 2 秒，它不但繼承了區域分割的好處，而且大部份所需要精確計算位置的點的所需考慮特徵線數目都減少許多，尤其是比較遠的背景部份，幾乎不受任何特徵線的影響，所以那些背景部份計算時間幾乎為零，而即是物體部份，它只受局部較近特徵線影響，所以進一步簡化了每個區域所要考慮的特徵線數目，並且也兼顧到影像的品質，可說是本篇論文中實驗出來效能最好的方法。

4. 結論與未來發展

這篇論文中，我們以現有的影像變形的方加以改良，分析幾種改良方法，最後成功的大大改善了原來影像變形的技術，現在我們只是對 2D 的影像作變形，未來將嘗試對 3D 的立體資料直接作體積變形，而且現在特徵線乃是以手動的方式，未來將結合影像處理的技巧，努力嘗試自動找出特徵線的方法。

REFERENCES

- [1] T. Beier and S. Neely. Feature-based image metamorphosis. In Computer Graphics, vol 26(2), pp 35-42, New York, NY, July 1992. Proceeding of SIGGRAPH '92.
- [2] Apostolos Leros, Chase D. Garfinkle, and Marc Levoy. Proceeding of SIGGRAPH 96'
- [3] Steen M. Seitz and R. Dyer. View Morphing. Processing of SIGGRAPH 96'.
- [4] Barr, A.H., Global and Local Deformation of Solid Primitives. In "Proc. SIGGRAPH '84" (Minneapolis, July 23-27, 1984). Published as "Computer Graphics", 18(3)(July 1984), pp. 21-30.
- [5] Kent, J., Parent, R., Carlson, "Establishing Correspondences by Topological Merging: A New Approach to 3-D Shape Transformation", Proceeding
- [6] Oka, M., Tsutsui, K., Akio, O., Yoshitaka, K., Takashi, T., Realtime Manipulation of Texture-

Mapped Surface. In "Proc. SIGGRAPH '87"(Anaheim, July 27-31, 1987). Published as "Computer Graphics", 21(4) (July 1987), pp. 181-188.

[7] Wolberg, G., "Digital Image Warping". IEEE Computer Society Press, 1990.

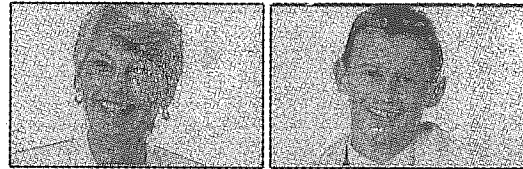


圖 10.1

圖 10.2

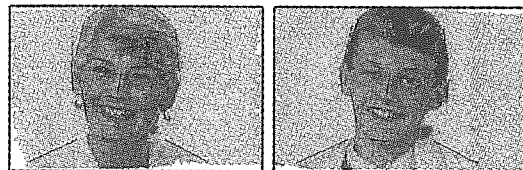


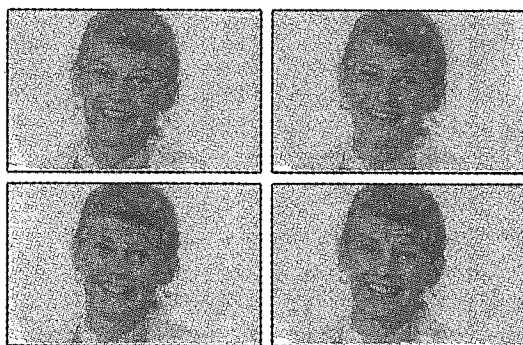
圖 10.3

圖 10.4



圖 10.5

圖十：圖 10.1 和圖 10.2 為兩張 320×200 的原始圖形，其所選取對應的 22 條特徵線如圖所示，圖 10.3 和圖 10.4 為其內差的特徵線，所扭曲後的影像分別如圖所示，圖 10.5 為圖 10.3 和圖 10.4 內差混合所得。



圖十一：輸入兩張原始影像圖 10.1 和圖 10.2，左上為 T. Beier 結果，右上為掃描線演算法的結果，左下為區域分割的結果，右下為區域分割和權重的結果。此四種方法所得變形結果，影像品質相差不大。