

國立交通大學

電控工程研究所

碩士論文

人臉辨識及表情辨識之整合設計



An Integrated Design of Face and Facial Expression  
Recognition

研究 生：陳奕彥

指 導 教 授：宋開泰 博 士

中 華 民 國 九 十 九 年 七 月

# 人臉辨識及表情辨識之整合設計

An Integrated Design of Face and Facial Expression Recognition

研究 生：陳奕彥

Student: Yi-Wen Chen

指導教授：宋開泰 博士

Advisor: Dr. Kai-Tai Song



Submitted to Institute of Electrical Control Engineering

College of Electrical and Computer Engineering

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of Master

in

Electrical Control Engineering

July 2010

Hsinchu, Taiwan, Republic of China

中華民國九十九年七月

# 人臉辨識及表情辨識之整合設計

學生:陳奕彥

指導教授:宋開泰 博士

國立交通大學電控工程研究所

## 摘要

本論文發展一套應用於機器人互動之人臉辨識與表情辨識之整合設計。人臉影像先經由主動外觀模型(Active Appearance Model, AAM)計算出人臉形狀以及紋理模型；接著對輸入的人臉影像進行改良式 Lucas-Kanade 影像校正以找出人臉特徵，再利用 AAM 的人臉紋理模型建立出人臉紋理的特徵參數，並利用此特徵參數輸入倒傳遞類神經網路做辨識。在辨識過程中，我們提出一種整合設計，先經由人臉辨識找出使用者的身分，接著我們對於辨識出的已知使用者的表情資料庫做個人化表情辨識；由實驗結果可以看出，在使用 BU-3DFE 人臉表情資料庫做辨識，其人臉辨識成功率可達 98.3%。接著使用個人化表情辨識的成功率為 83.8%，相對於使用全體的表情辨識器之辨識率僅為 69.6%，其成功率可大幅提升。

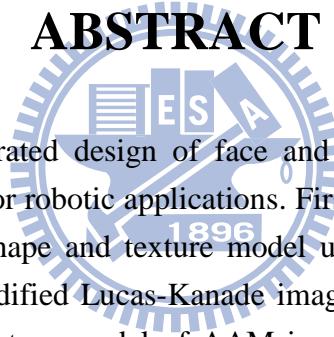
# An Integrated Design of Face and Facial Expression Recognition

Student: Yi-Wen Chen

Advisor: Dr. Kai-Tai Song

Institute of Electrical Control Engineering  
National Chiao Tung University

## ABSTRACT



In this thesis, an integrated design of face and facial expression recognition system has been developed for robotic applications. First, facial image from camera is exacted to compute facial shape and texture model using active appearance model (AAM). Second, we use modified Lucas-Kanade image alignment algorithm to find facial features. Third, the texture model of AAM is used to construct facial texture parameters. These parameters are used to train a back propagation neural network (BPNN) for face and facial expression recognition. In recognition process, we first use face recognition to find user's identity; then we use recognized user's facial expression database to recognize his/her facial expression. In experiments based on BU-3DFE database, a face recognition rate of 98.3% has been achieved. The facial expression recognition rate of the proposed integrated method (using a personal facial expression classifier) is 83.8%. It is a great improvement compared with using conventional facial expression classifier of 69.6%.

## 誌謝

謹向我的指導教授宋開泰博士致上感謝之意，感謝他兩年來在專業上的指導，以他豐富的學識與經驗，配合理論的應用，使得本論文得以順利完成。感謝各位口試委員的指導與意見，讓本論文能夠更加嚴謹。

感謝博士班學長姐孟儒、嘉豪、格豪、巧敏以及信毅，在理論上的指導與建議，並感謝各位與我共同努力的同學哲豪、宗暘、維峻與傑巽彼此之間相互的鼓勵與提供研究上的意見，以及學弟仕晟、建宏、上畯與碩成不時的提供許多的幫忙。

最後，特別要感謝我的父母，由於他們的辛苦的栽培以及在生活上細心地關懷與照料，使得我得以順利的完成此碩士論文，在此願將此論文獻給我最敬愛的父母。



# 目錄

摘要 .....	i
ABSTRACT .....	ii
誌謝 .....	iii
目錄 .....	iv
圖例 .....	vi
表格 .....	viii
<b>第一章、緒論 .....</b>	<b>1</b>
1.1. 研究動機 .....	1
1.2. 相關文獻回顧 .....	1
1.3. 問題描述 .....	7
1.4. 系統架構與章節說明 .....	8
<b>第二章、人臉偵測與特徵點擷取方法 .....</b>	<b>10</b>
2.1. 人臉偵測演算法 .....	10
2.1.1. 人臉區域決定 .....	11
2.1.2. 人臉偵測成果 .....	13
2.2. 主動外觀模型 .....	15
2.3. 人臉形狀模型 .....	15
2.3.1. 特徵點的設置 .....	15
2.3.2. 形狀對齊 .....	17
2.3.3. 模擬臉部形狀的變化 .....	20
2.4. 人臉紋理模型 .....	22
2.4.1. 人臉的紋理 .....	22
2.4.2. Piecewise Affine Warping .....	22
2.4.3. 模擬臉部紋理的變化 .....	25
2.5. 影像校正演算法 .....	27
2.5.1. Lucas-Kanade 演算法 .....	27
2.5.2. Inverse Compositional 演算法 .....	30
2.5.3. 整體形狀正規化轉換 .....	38
2.5.4. 梯度影像修正 .....	42
2.5.5. 直方圖等化 .....	45
2.5.6. 整體方法 .....	49

2.6. 結論與討論 .....	50
<b>第三章、人臉辨識與表情辨識演算法 .....</b>	<b>51</b>
3.1. 人臉特徵方法 .....	51
3.2. 倒傳遞類神經網路 .....	52
3.3. 表情辨識策略 .....	54
3.4. 人臉辨識與表情辨識訓練流程 .....	56
3.5. 結論與討論 .....	56
<b>第四章、實驗結果 .....</b>	<b>59</b>
4.1. 影像校正演算法迭代結果 .....	59
4.1.1. 使用資料庫影像.....	59
4.1.2. 使用自行擷取影像.....	64
4.2. 人臉辨識結果 .....	66
4.2.1. 使用資料庫影像.....	66
4.2.2. 使用自行擷取影像.....	70
4.3. 表情辨識結果 .....	72
4.3.1. 使用資料庫影像.....	72
4.3.2. 使用自行擷取影像.....	76
4.4. On-line 辨識結果 .....	80
4.5. 結論與討論 .....	82
<b>第五章、結論與未來展望 .....</b>	<b>83</b>
5.1. 結論 .....	83
5.2. 未來展望 .....	83
<b>參考文獻.....</b>	<b>85</b>

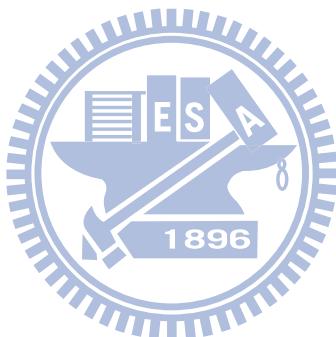
## 圖例

圖 1-1、一般人臉/表情辨識的流程圖 .....	2
圖 1-2、Template Selection Tracker(TST)方法的流程[4].....	3
圖 1-3、在 c 參數變化三個標準差的 Template[5] .....	4
圖 1-4、在眨眼時，所找出的 Template[5].....	4
圖 1-5、經由邊緣偵測的邊緣影像[9].....	5
圖 1-6、使用邊緣影像與灰階影像的 AAM 匹配正確率[9] .....	5
圖 1-7、SVM 表情辨識的流程[2].....	7
圖 1-8、系統架構 .....	9
圖 2-1、Haar-like 矩形特徵 .....	11
圖 2-2、人臉偵測在兩公尺時的結果 .....	12
圖 2-3、人臉偵測整體流程 .....	13
圖 2-4、人臉偵測的範圍 .....	13
圖 2-5、人臉偵測詳細流程範例 .....	14
圖 2-6、特徵點的選取[18].....	15
圖 2-7、人臉特徵點的設置 .....	16
圖 2-8、不同表情的特徵點標點情形(使用 AR Database[23]).....	16
圖 2-9、經過形狀對齊前的輸入形狀與經過形狀對齊後的形狀 .....	19
圖 2-10、由形狀對齊後的所有形狀計算出的平均形狀 .....	19
圖 2-11、人臉形狀模型的 mean shape(a)和前三個變形模型(b, c, d).....	21
圖 2-12、Delaunay triangulation 的外接圓特性[21] .....	23
圖 2-13、人臉形狀以 Delaunay triangulation 表示 .....	23
圖 2-14、三角形的形變 .....	24
圖 2-15、人臉形變的情形 .....	25
圖 2-16、人臉紋理模型的平均紋理(a)和前四個變化模型(b, c, d, e).....	26
圖 2-17、Lucas-Kanade 演算法流程 .....	30
圖 2-18、Inverse Compositional 演算法流程 .....	33
圖 2-19、 $(xi_0, yi_0)^T + (\Delta xi_0, \Delta yi_0)^T$ 經由 Affine warp 得到 $(xi, yi)^T + (\Delta xi, \Delta yi)^T$ .....	36
圖 2-20、Global shape normalizing transform 形狀變化模型 .....	39
圖 2-21、用於計算梯度的樣板影像 .....	42
圖 2-22、原本的水平與垂直梯度影像 .....	43
圖 2-23、修改後的水平與垂直梯度影像 .....	43
圖 2-24、梯度影像修正之前後影像校正結果 .....	44
圖 2-25、在每次迭代的過程中將形變影像進行亮度調整與直接對輸入影像進行亮度調整的結果.....	45

圖 2-26、直方圖等化的計算 .....	46
圖 2-27、直方圖等化過後的結果 .....	47
圖 2-28、使用不同方法所得到的人臉平均紋理 .....	47
圖 2-29、在每次迭代的過程中將形變影像進行直方圖等化調整與直接對輸入影像 進行亮度調整的結果.....	48
圖 3-1、人臉影像參數最為人臉特徵 .....	52
圖 3-2、類神經網路主要架構 .....	52
圖 3-3、[28]中所使用的類神經網路模型.....	54
圖 3-4、成員的人臉表情辨識範例 .....	55
圖 3-5、非成員的人臉表情辨識範例 .....	56
圖 3-6、人臉辨識與表情辨識的整體訓練流程 .....	57
圖 4-1、AR database 的部分影像.....	60
圖 4-2、所有 84 筆測試資料平均灰階 RMS 誤差收斂情形.....	60
圖 4-3、使用 AR database 的特徵點結果以及其形變出的人臉.....	61
圖 4-4、其中四位不同資料庫成員的表情影像.....	62
圖 4-5、所有 480 筆測試資料平均灰階 RMS 誤差收斂情形.....	63
圖 4-6、使用 BU-3DFE 的特徵點結果以及其形變出的人臉.....	64
圖 4-7、其中五位實驗室成員的各種表情資料 .....	65
圖 4-8、所有 300 筆測試資料平均灰階 RMS 誤差收斂情形.....	65
圖 4-9、其中五位實驗室成員的各種表情特徵點結果 .....	66
圖 4-10、人臉辨識類神經網路的收斂情形 .....	67
圖 4-11、人臉辨識類神經網路的收斂情形 .....	70
圖 4-12、全體表情辨識類神經網路的收斂情形 .....	72
圖 4-13、個人表情辨識類神經網路 20 位使用者的平均收斂情形 .....	73
圖 4-14、全體表情辨識類神經網路的收斂情形 .....	76
圖 4-15、個人表情辨識類神經網路 10 位使用者的平均收斂情形 .....	77
圖 4-16、on-line 辨識結果-生氣.....	80
圖 4-17、on-line 辨識結果-厭惡.....	80
圖 4-18、on-line 辨識結果-害怕.....	81
圖 4-19、on-line 辨識結果-高興.....	81
圖 4-20、on-line 辨識結果-傷心.....	82
圖 4-21、on-line 辨識結果-驚訝.....	82

# 表格

表 4-1、20 位成員的辨識率 .....	67
表 4-2、人臉辨識率與表情的關係 .....	68
表 4-3、人臉辨識錯誤的結果 .....	68
表 4-4、非成員的人臉辨識結果 .....	69
表 4-5、10 位實驗室成員的辨識率 .....	71
表 4-6、人臉辨識率與表情的關係 .....	71
表 4-7、使用 BU-3DFE 的使用全體的表情辨識結果.....	74
表 4-8、使用 BU-3DFE 在已知使用者的情況使用個人的表情辨識結果.....	75
表 4-9、使用 BU-3DFE 的整體人臉辨識與個人的表情辨識結果.....	75
表 4-10、使用實驗室成員的全體的表情辨識結果 .....	78
表 4-11、使用實驗室成員在已知使用者的情況使用個人的表情辨識結果 .....	79
表 4-12、使用實驗室成員的整體人臉辨識與個人的表情辨識結果 .....	79



# 第一章、緒論

## 1.1. 研究動機

近幾年來，機器人科技的發展越來越快速，而機器人已經逐漸的走入家庭。其中能與人互動的機器人之中，包括了看護機器人、保全機器人、接待機器人以及寵物機器人等等。在機器人與人互動(Human Robot Interaction, HRI)的設計中，我們希望機器人可以對使用者有更自然的互動，因此，我們希望機器人可以知道更多的使用者資訊，例如使用者身份，使用者目前的情緒為何，使用者目前在做什麼事，使用者想要做什麼等等。機器人必須運用所擁有的感測器來判別出使用者目前的狀態，然後決定出下一步的所需採取的動作，使得機器人越來越接近實用的功能。



在人類的臉上，包括了許多複雜單元，因此每個人的人臉都會有相當大的差異存在，因此，分析每個人臉上的不同，就可以得到許多不一樣的資訊，例如人臉辨識可以知道使用者是誰，表情辨識可以知道使用者目前的情緒，年齡估測以及性別辨識可以得到使用者的一些基本資料，機器人藉由辨識得知許多使用者資訊後，就可以針對使用者來進行各式各樣豐富的互動，讓機器人與人類更加生動的互動。

在目前辨識技術的應用當中，人臉辨識可以應用在許多不同的環境下，例如使用者身份的確認、防盜門禁系統等等，而在表情辨識的發展上，數位相機中的微笑快門可以說是應用最廣的一部分；而在機器人應用的開發當中，人臉辨識以及表情辨識都是讓機器人能夠與使用者產生互動的重要技術。

## 1.2. 相關文獻回顧

人臉以及表情辨識是利用影像中的人臉區域，對於資料庫中的人臉以及表情

資料進行比對，以得知使用者的身分以及表情。目前在人臉以及表情辨識方面的過程主要可以分為三部份 1.人臉偵測 2.特徵點擷取 3.人臉/表情辨識如圖 1-1 所示。

在人臉表情分析的文獻中，Ekman 等人 [1]所定義的 Facial Action Coding System (FACS)，這是根據人在表現出不同表情時，所帶動臉部肌肉的運動的原理，定義出了 44 種不同的 Action Unit，根據不同的 Action Unit 組合，即可以表現出人的各種不同表情。另外，在基本表情的定義下，可以定義出六種不同的基本表情，亦即為生氣(sadness)、厭惡(disgust)、害怕(fear)、高興(happiness)、驚訝(surprise)和傷心(sadness)六種。

在特徵辨識的方面，主要可分為兩種方法，一是整體特徵，一是局部特徵。整體特徵直接將整張人臉當作單一特徵來做辨識；局部特徵方法是先找出關於人臉/表情變化的重要局部特徵，例如眼睛、鼻子和嘴巴或嘴角、眼角等[2][3]，然後分別針對這些局部特徵的變化做辨識，最後將個別局部特徵的結果統合而得到最後結果。

Cristinacce 和 Cootes [4]提出了 Template Selection Tracker(TST)臉部特徵偵測的方法，先由手動對訓練資料庫進行標點，對於這些標點產生形狀(Shape)，並對之取平均而得到平均形狀(Mean shape)，當一張新進影像進入時，先將平均形狀套用到這張圖上，然後(1)使用 K 最近鄰居法(K-Nearest Neighbor, KNN)找出在資料庫中 Shape 最相近的 K 個 Template，並在這 K 個 Template 中對其紋理資

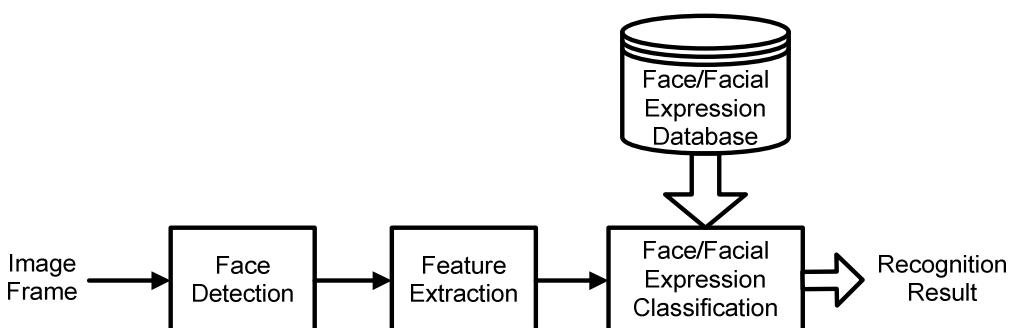


圖 1-1、一般人臉/表情辨識的流程圖

料進行正規化影像相關(normalized correlation)找出最相似的一個 Template；下一步(2)則是使用 Shape Constrained Template Search 方法來預測出新的特徵點位置，這是使用訓練資料的參數對影像作最佳化，更新 shape 參數；重複(1)(2)兩步驟直到座標點收斂，圖 1-2 即為此方法的流程。

之後 Cristinacce 和 Cootes [5] 對上述的方法進行修改，將 K 最近鄰居法 (K-Nearest Neighbor, KNN) 改為 Constrained Local Appearance Models 方法，它是將訓練資料先進行參數化，將所有的資料以(1-1)表示：

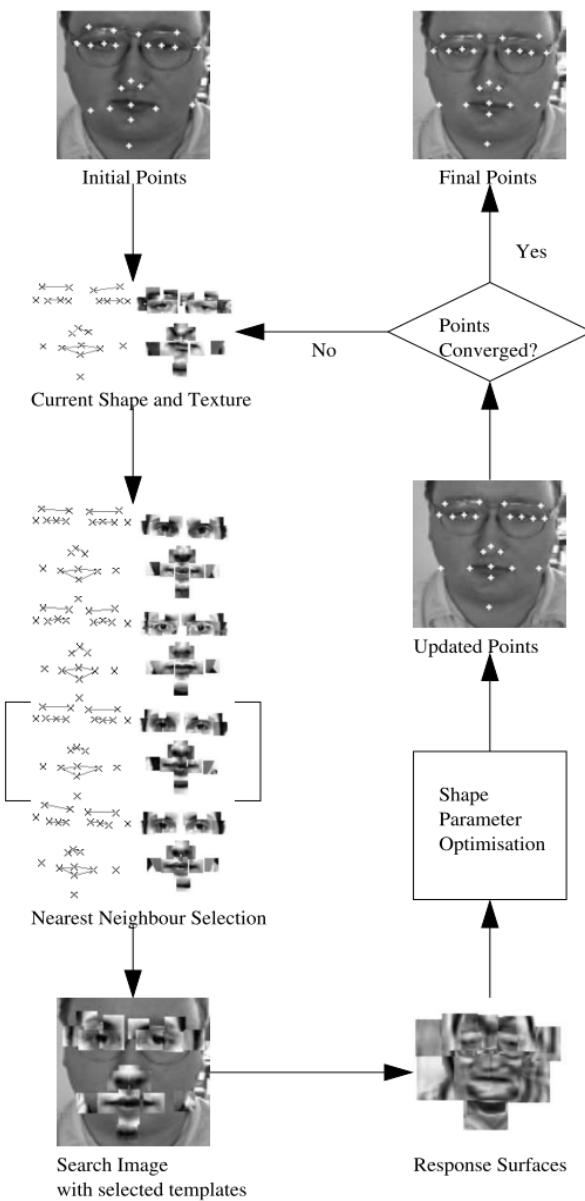


圖 1-2、Template Selection Tracker(TST)方法的流程[4]

$$x = \bar{x} + P_s b_s \quad g = \bar{g} + P_g b_g \quad (1-1)$$

表示，其中  $\bar{x}$  是平均形狀， $\bar{g}$  是平均灰階向量， $P_s$  是正交變化量， $b_s$  是形狀參數， $P_g$  是正交變化量， $b_g$  是灰階參數，將兩者結合，可以得到

$$b = P_c c \quad P_c = \begin{pmatrix} P_{cs} \\ P_{cg} \end{pmatrix} \quad b = \begin{pmatrix} W_s b_s \\ b_g \end{pmatrix} \quad (1-2)$$

因此，當我們改變  $c$  的權重時，即可得到各種不同的人臉模型，如圖 1-3。而圖 1-4 則是使用這個方法來對人臉的不同 frame 做 feature 的抓取，可以看出在眨眼時可以找出對應的 Template。

這個方法因為特徵的數量不多，在運算的速度上可以有較好的效果，但是其只計算部分的人臉特徵，屬於局部特徵方法，不能得到完整的人臉資訊，應用於人臉辨識系統當中較不理想。

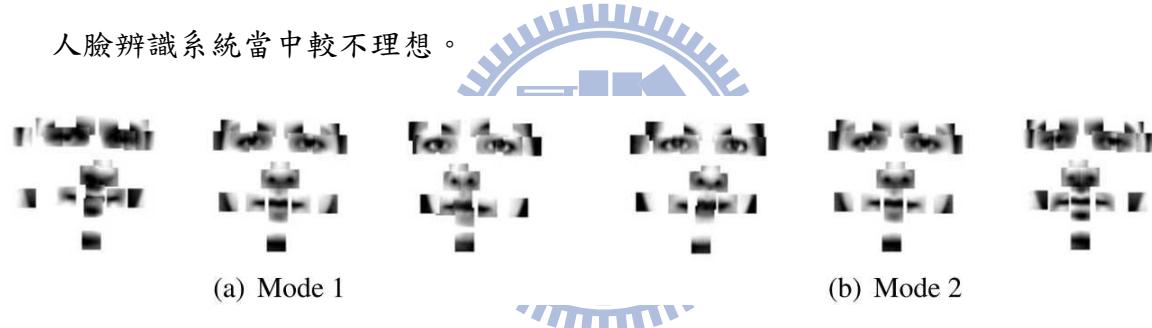


圖 1-3、在  $c$  參數變化三個標準差的 Template[5]

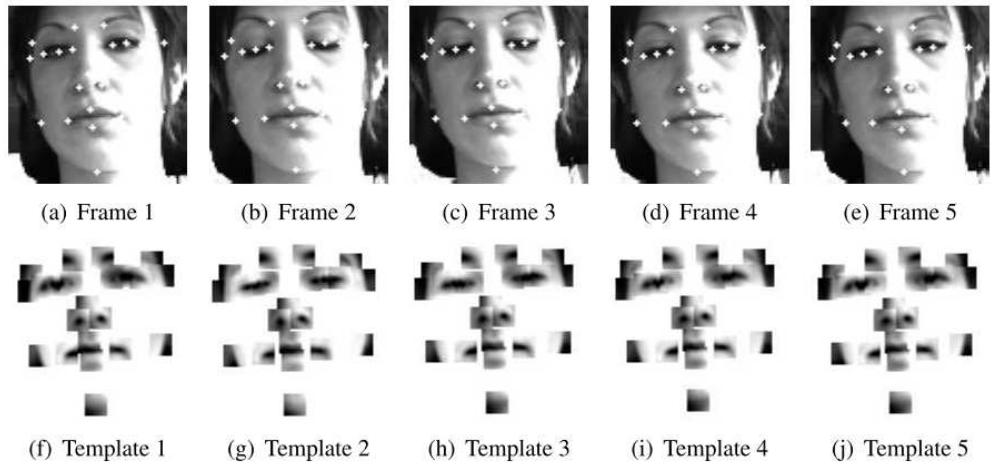


圖 1-4、在眨眼時，所找出的 Template[5]

Martin 等人 [9] 提出了不使用灰階影像 (Gray image) 於主動外觀模型 (Active Appearance Model, AAM) 上而使用了邊緣影像 (Edge image) 的方法，作者認為邊緣影像可以將笑線 (Laugh line) 和額頭的皺紋比灰階影像清楚的抽取出來。因此在 AAM 的 appearance model 上，作者先將影像經由影像邊緣偵測找出邊緣影像，將正規化的影像經由  $G_x$ ,  $G_y$  分別找出 x、y 方向的邊緣，

$$S_x = I(W(x, p)) * G_x, S_y = I(W(x, p)) * G_y \quad (1-3)$$

再計算出整體的邊緣影像  $S$ ，如圖 1-5，

$$S = \sqrt{S_x^2 + S_y^2} \quad (1-4)$$

以此邊緣影像來建立 appearance model，圖 1-6 顯示了分別使用邊緣影像與灰階影像的 AAM 匹配正確率，可以看出邊緣影像正確率明顯的較灰階影像來的高；在實驗上，作者先用 Viola 和 Jonse [10]的人臉偵測演算法將人臉部分框出，再用 AAM 來進行匹配，最後分別使用多層感知器類神經網路 (Multi-Layer-Perceptron, MLP) 以及狀態向量機 (Support-Vector-Machine, SVM) 分類器進行分類，使用邊緣影像可以使部分特徵更明顯，但是也會遺失部分的紋理資訊。

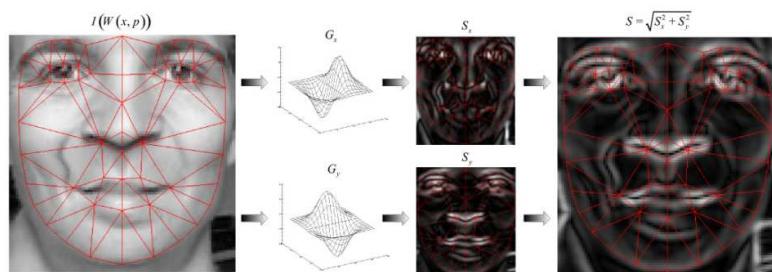


圖 1-5、經由邊緣偵測的邊緣影像[9]

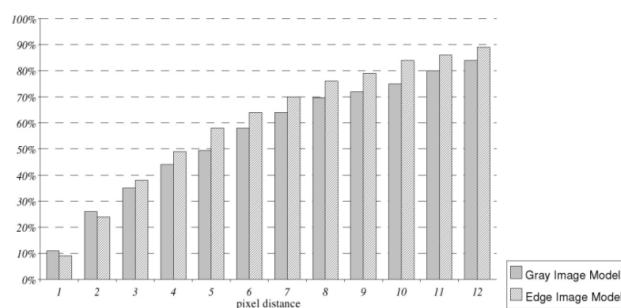


圖 1-6、使用邊緣影像與灰階影像的 AAM 匹配正確率[9]

Tang 和 Huang [6]提出了一套自動選擇特徵的方法用於 3D 的表情辨識上，他們利用了 MPEG-4 standard [7]的 Facial Animation Parameter Units(FAPUs)來對臉部做正規化，以不同兩點的連線作為特徵庫，然後再使用 Kullback-Leibler diverge (KLD) [8]機率分布方法來找出特徵值的 Discrimination power (DP(X))，選出前 K 個較大的作為特徵值，再以正規 AdaBoost 分類器呼叫三種不同的 weak 分類器 Nearest Neighbor、Naive Bayes、Linear Discriminant Analysis(LDA)來進行分類。這個方法最大的優點就在於它是利用三維的臉部幾何資訊來做特徵的擷取，因此不會受到光線的變化的影響。

在[3]中，作者利用一些觀察到表情變化的一些線索來當作模糊規則，例如驚訝時會張嘴、可能出現抬頭紋、屬於垂直走向的嘴型，生氣時再眉心區域會出現皺眉、嘴巴閉合、屬於平走向的嘴型.....等，並使用了 Minimum inference engine 來進行模糊推論；而[11]中，作者使用了 Mamdani's inference method 來進行模糊推論；最後使用重心解模糊化法 (Center of gravity defuzzification) 來解模糊化，可以得到每一種表情與其所對應的表情權重值，選取權重值為最大者的表情做為辨識結果的輸出，文獻[3]中進一步對於輸出時可能有兩個最高表情權重值過於相近時的情況做探討，將相近( $\leq 0.1$ )的兩個結果對其特徵值再進一步增設幾個判斷條件，以增加其辨識的正確率，其正確率可達 81.76%。使用模糊推論的一些規則可以讓系統擁有一些彈性存在，但是由於每個人的表情元素可能不太相同，若規則定的不夠完備可能造成辨識錯誤。

許晉懷 [12]使用了 SVM 分類器對表情進行分類。利用了 SVM 的雙模辨識策略：透過兩種資訊辨識情緒，對辨識的可靠度設定不同的權重，用以決定要該採用何種資訊。雙模資訊權重的數值，是透過 SVM 理論中測試資料距超平面的距離，以及訓練資料之標準差，接著再經由訓練資料距超平面的平均距離正規化後決定，此權重係數即可以判斷分類的可靠度；並以此權重係數對於設定好 SVM 分類器的兩兩表情進行多階層的分類，最後即可決定出輸出的表情為何。如圖 1-7 所示，如此可以減少對於所有兩兩表情的 SVM 分類所需的運算量。

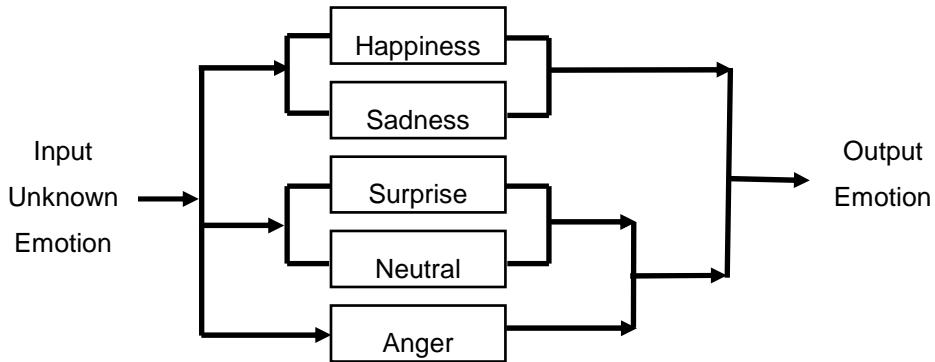


圖 1-7、SVM 表情辨識的流程[2]

### 1.3. 問題描述

在一般的情形下，每個人的自然表情以及當表情發生變化時的變化元素都不完全相同，例如有些人的中性表情在一般不認識的人來看會覺得他是處在生氣的情緒中，有些人的中性表情在一般不認識的人來看會覺得他是處在快樂的情緒中；當我們熟識這個人的情形下，我們就會大概知道這個人的中性表情為何，而不會把他的中性表情誤認為生氣或快樂。

在機器人的應用上，我們希望將機器人比擬為人類，當這個機器人在家庭中對於家中的成員或是機器人所熟識的人，可以知道這些人的表情變化因素，如此，就可以讓機器人針對已知的這個人，讓機器人可以知道這個人確切的表情，而不會誤認使用者的情緒。

在一般表情辨識的方法當中，是使用大部分人所擁有的臉的特徵，對於所有的使用者皆使用相同的特徵來做辨識。因此，很容易忽略掉每個人所擁有的表情特徵是不完全相同的，而使得有些人的表情無法正確辨識出來。

在本論文中，我們希望藉由針對每位使用者建立其個人的表情資料庫，讓機器人在辨識使用者的表情時，可以更確切的知道使用者目前真正的情緒為何，並希望能夠在未來讓機器人能針對不同的使用者，有不一樣的互動行為，讓機器人可以達到更生動的互動行為。

## 1.4. 系統架構與章節說明

對於應用於機器人上的人臉與表情辨識系統，本論文提出了一個辨識系統的架構如圖 1-8 所示，這個架構下大致分為三個部分，分別是人臉偵測、人臉的特徵擷取以及人臉以及表情辨識部分；人臉偵測是為了要找出影像中人臉的位置，而特徵擷取的部份是為了要找出可以做為人臉以及表情辨識的特徵，在人臉以及表情辨識部分是為了要辨識出輸入的使用者以及其表情出來。

在人臉偵測部分，主要是使用 Viola [10] 以及 Lienhart[13] 的 Haar-like 人臉偵測來整出可能的人臉位置，接著再加入幾項判斷條件將最接近人臉的影像篩選出來。在人臉的特徵擷取部分，我們先正規化影像大小，並使用了主動外觀模型 (AAM) 以及 Lucas-Kanade 演算法找出人臉部分並轉正，並取出人臉的紋理特徵。我們將人臉的紋理特徵放入倒傳遞類神經網路(Back Propagation Neural Network, BPNN)進行辨識。在辨識的過程中，我們先進行人臉辨識，找出使用者的身分，接著再針對我們辨識出來的成員進行個人化的表情辨識，以得到個人化的表情辨識結果。

本論文一共分為五章，第一章先介紹本研究的動機、相關研究以及問題描述，接著，在第二章會說明人臉的偵測以及特徵點的擷取，在第三章，我們會提出人臉以及表情辨識的方法，第四章展示一些實驗的結果，最後在第五章是結論以及未來展望。

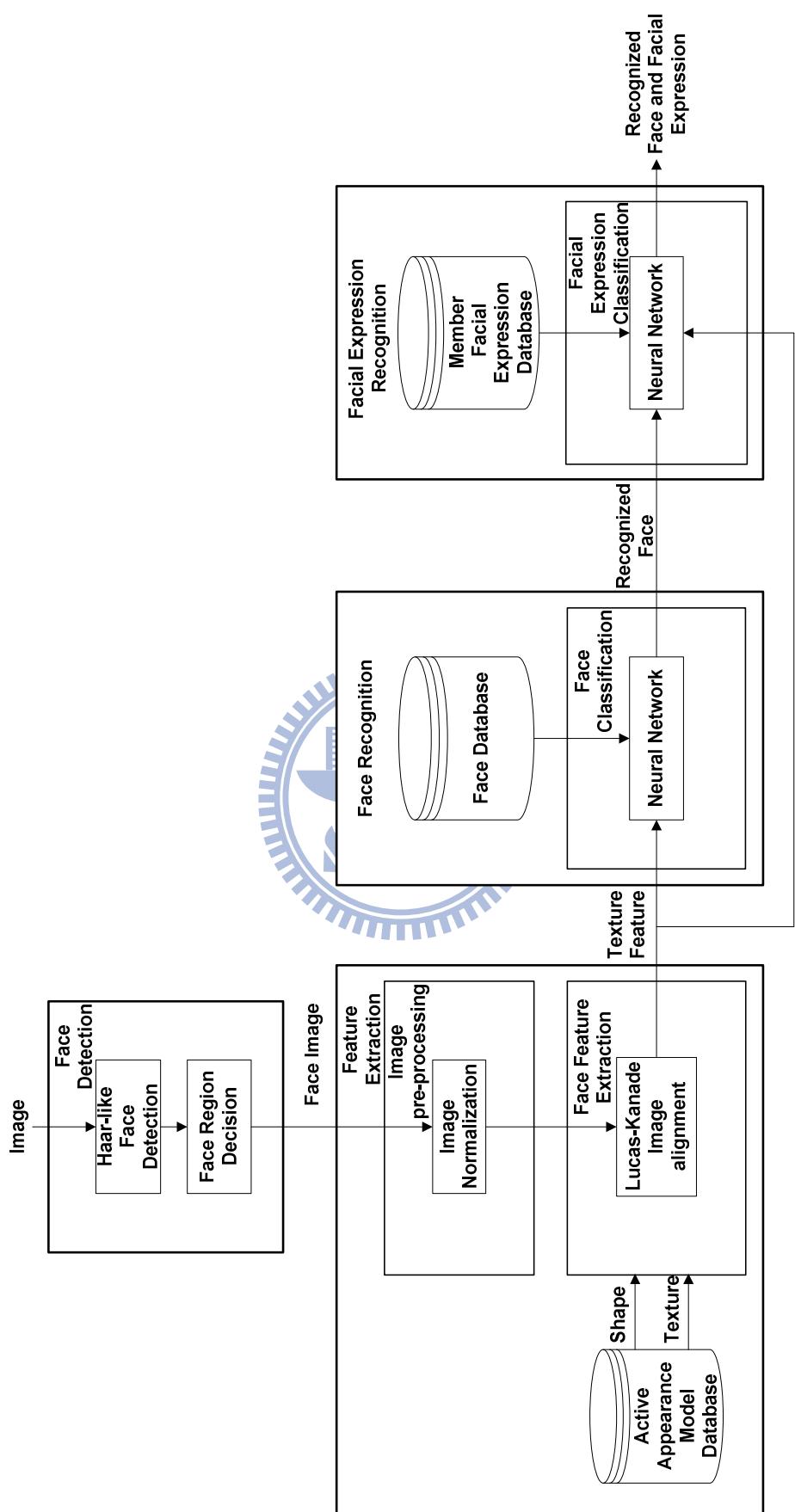


圖 1-8、系統架構

## 第二章、人臉偵測與特徵點擷取方法

在人臉與表情辨識系統中，我們首先需要知道人臉所在的位置以及擷取出做為辨識用所需的人臉特徵。因此接下來將介紹如何由攝影機取得影像並進行人臉偵測，取得正確的人臉位置；並接著使用主動外觀模型(Active Appearance Model)及 Lucas-Kanade 演算法找出人臉的特徵部位並將人臉轉正，取得人臉的紋理特徵值。

### 2.1. 人臉偵測演算法

在本系統中，我們先經由人臉偵測出人臉的位置，以利接下來進行的人臉特徵擷取；在人臉偵測部分，我們希望能準確的找出人臉的位置，而直接使用膚色偵測的方法來偵測人臉容易造成人臉區域偵測錯誤的情形發生，因此在這裡首先先由攝影機(Webcam)取得 640\*480 大小的影像，並在一開始採用 Viola [10] 以及 Lienhart[13] 的 Haar-like 人臉偵測來找出可能的人臉位置。

Haar-like 特徵的人臉偵測方法是不同形式的特徵如圖 2-1 所示，1.為邊緣特徵的四種、2.為線特徵八種、3.為中心包圍特徵兩種，將白色部分的影像像素值總和減去黑色部分的影像像素值總和來計算其特徵，利用改變矩形特徵的大小和位置，即可在影像中取出大量的特徵。

在影像當中的感興趣區域(Region of interest, ROI)中，使用了 Cascade 方法來對多個階層的分類器進行判斷，每一個感興趣區域在所有的階層都通過之後，就被分被為正確的結果；反之，在任一階層中被拒絕了就被淘汰。而在每一階層當中的分類器中，對基本的分類器採用了 Boosting 方法(AdaBoost)，每個基本的分類器是至少 2 元的決策樹分類器，他們的輸入即為所有前述的 Haar-like 特徵。本論文使用了 OpenCV(Open Source Computer Vision)所提供之 Haar-like 物體偵測方法來達成人臉偵測目的。

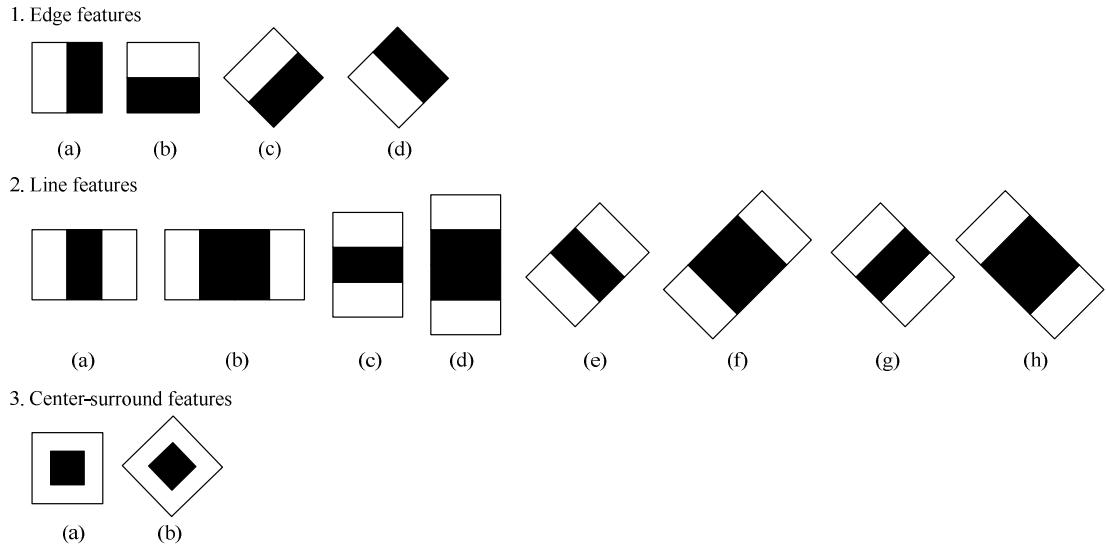


圖 2-1、Haar-like 矩形特徵

由於 Haar-like 人臉偵測可能找到一些較遠而造成影像畫質較低的人臉或是非人臉的區域，因此在 Haar-like 人臉偵測找出的幾個可能的人臉區域中，我們利用人臉影像區域大小以及人臉影像區域膚色密度的大小來決定出何者為真正的人臉區域，找出最適合的人臉。

### 2.1.1. 人臉區域決定

由於 Haar-like 人臉偵測可能找到一些較遠而造成影像畫質較低的人臉或是非人臉的部位，因此在這裡需要經由一些判斷來找出是人臉的區域；首先我們先針對找出人臉影像的大小進行過濾，這裡根據偵側出人臉影像大小的畫質最低可容忍範圍，在人距離攝影機兩公尺時，人臉影像大小大約是 50\*50 個像素如圖 2-2。在此距離所得的人臉影像已經不夠清楚來進行人臉的辨識，因此我們就設定偵側人臉大小最小的限制為 50\*50 個 pixel，小於這個大小的人臉影像就過濾掉。

過濾太小的人臉影像之後，我們對於人臉影像中的膚色進行偵測，由於 RGB 色彩模型在光源變化時會有非常大的改變，因此將 R、G 轉換到 Normalized color coordinates [14] 可減少 R、G 對於光源變化的靈敏度，R、G 轉換的公式如(2-1)(2-2)：

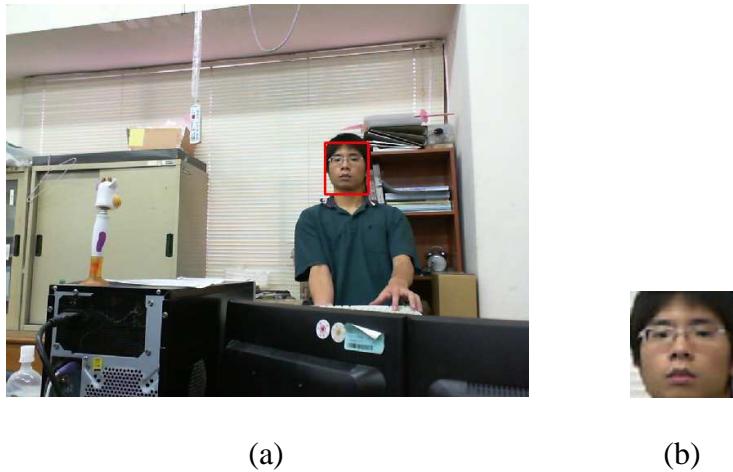


圖 2-2、人臉偵測在兩公尺時的結果，(a)框出人臉部分，(b)取出的人臉

$$r = R/(R + G + B) \quad (2-1)$$

$$g = G/(R + G + B) \quad (2-2)$$

其中 R、G、B 為原始 RGB 人臉影像中的紅、綠、藍影像畫素，而 r 與 g 為正規化以後的紅、綠部分，由正規化以後的人臉膚色範圍於 r、g 的座標表示，可以找出人體膚色的分布範圍公式，其中 g 的上限  $g_{up}$  為[15]：

$$g_{up} = -1.376r^2 + 1.0743r + 0.1452 \quad (2-3)$$

g 的下限  $g_{down}$  為[15]：

$$g_{down} = -0.776r^2 + 0.5601r + 0.1766 \quad (2-4)$$

其中白色的色彩也會包含於其中( $r=0.33, g=0.33$ )，因此我們必須將白色的部分過濾出來，白色的表示為：

$$W_r = (r - 0.33)^2 + (g - 0.33)^2 \quad (2-5)$$

因此，我們可以計算出膚色的範圍為：

$$Skin(r, g) = \begin{cases} 1, & (g < g_{up}) \text{ and } (g > g_{down}) \text{ and } W_r > 0 \\ 0, & \text{otherwise} \end{cases} \quad (2-6)$$

其中  $Skin(r, g) = 1$  為膚色， $Skin(r, g) = 0$  非膚色。

找到膚色範圍後，我們計算出整張人臉影像的膚色範圍大小，並將膚色範圍大小除以人臉影像大小即可得出膚色密度(2-7)，利用每個人臉影像的膚色密度的

不同，我們就可以將膚色密度最大的區域定義為最後找到的人臉影像，整體流程如圖 2-3 所示。

$$Density_{skin} = \frac{\sum Skin(r,g)}{width \cdot height} \quad (2-7)$$

其中  $Density_{skin}$  為人臉影像的膚色密度， $\sum Skin(r,g)$  為人臉影像膚色範圍大小， $width$  和  $height$  分別為人臉影像的寬和長。

### 2.1.2. 人臉偵測成果

在這個人臉偵測方法中可以偵測到最近的人臉距離為 20 公分圖 2-4(a)，最遠可以達到 2 公尺圖 2-4(c)，圖 2-5 顯示出人臉偵測系統的詳細的過程，在這個方法中，所找出的人臉為影像當中的一個人臉。

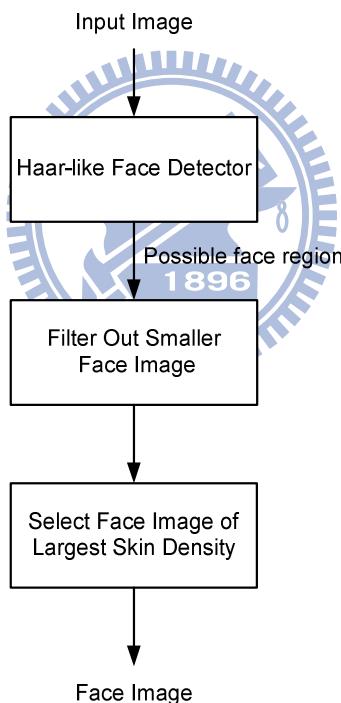


圖 2-3、人臉偵測整體流程

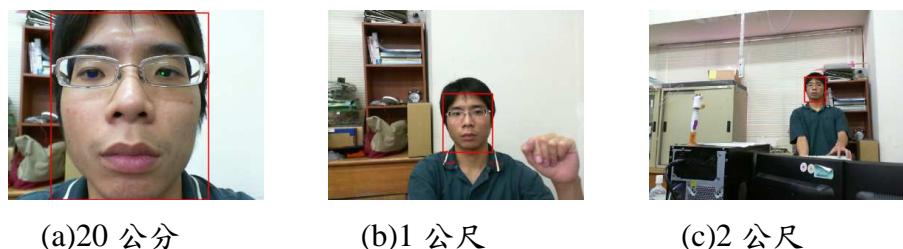


圖 2-4、人臉偵測的範圍

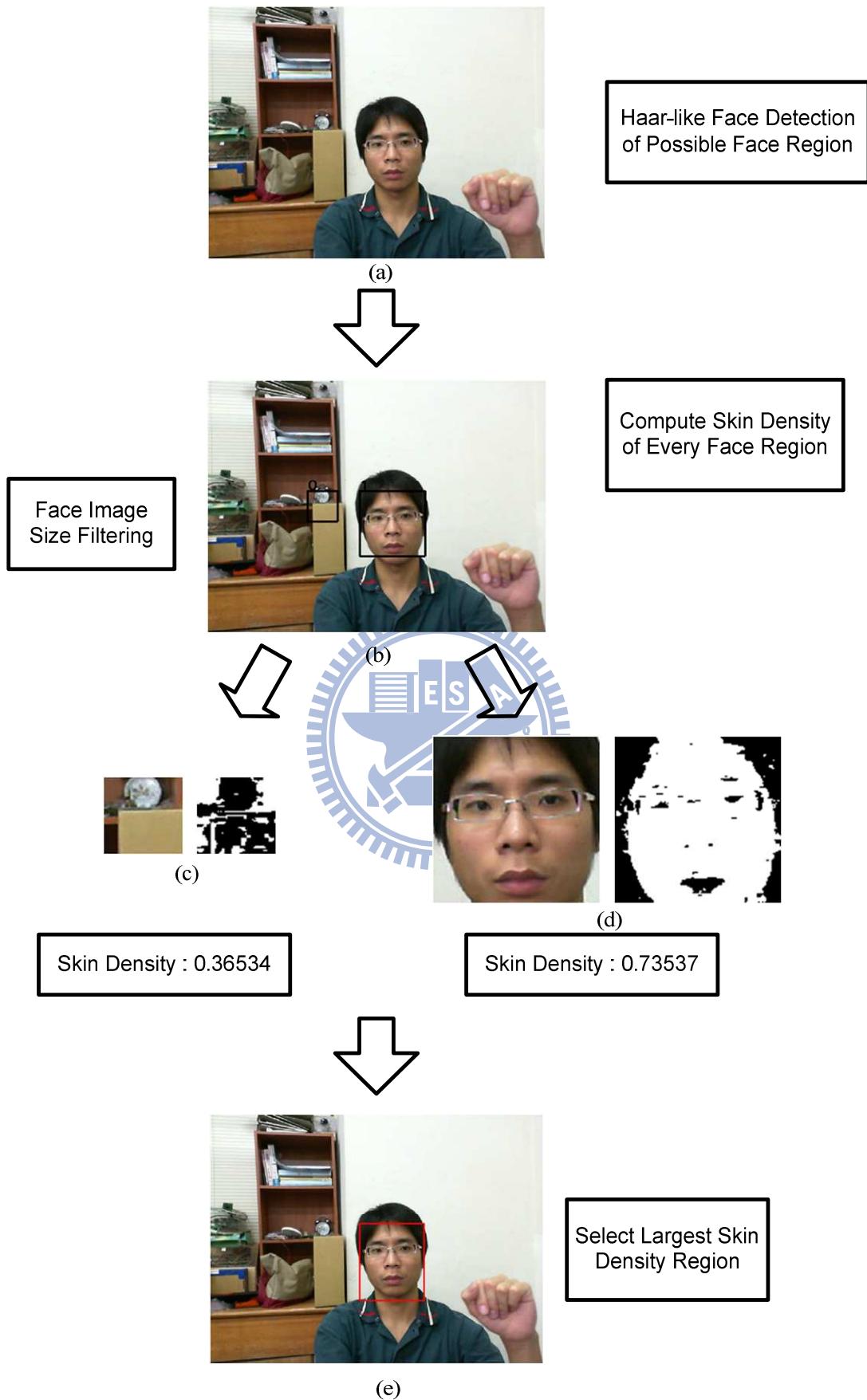


圖 2-5、人臉偵測詳細流程範例

## 2.2. 主動外觀模型

在人臉特徵的擷取上，我們希望能將人臉的特徵找出來，由於主動外觀模型(Active Appearance Model, AAM)包含了一個形狀模型(Shape Model)以及一個紋理模型(Texture Model)，利用這兩個模型可以對輸入的人臉影像來進行模擬，並可以把人臉的特徵準確的找出，因此這裡我們利用了主動外觀模型來匹配到輸入的人臉上，並藉此而得到我們想要的人臉特徵。

主動外觀模型是由 Cootes 等人[16-18]所提出，它是由作者之前的研究 Active Shape Model (ASM)[19]發展而來的，其主要是由一組形狀模型與一組紋理模型所組成，接下來就詳述其模型的建置方法。

## 2.3. 人臉形狀模型

在人臉的主動外觀模型中，包含了人臉形狀模型(Shape Model)。人臉形狀模型中，包含了一個平均人臉形狀(mean shape)以及平均人臉形狀可容許的變化向量組成，利用人臉模型就可以模擬出不同的人臉形狀出來。

### 2.3.1. 特徵點的設置

在決定人臉特徵點的位置時，特徵點是由曲線的高曲率(High Curvature)部分以及曲線間的 T 型交點(‘T’ Junction)，為了要定義出更確切的邊界，另外定義了以上兩種特徵點的中間點，如圖 2-6 所示。

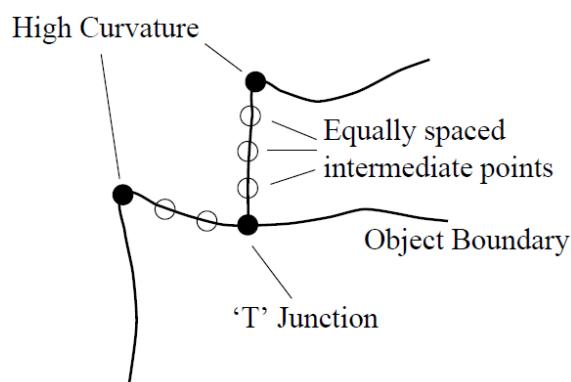


圖 2-6、特徵點的選取[18]

在特徵點的選擇上，這裡參考了文獻[18][20-22]的特徵點定義，定義出 70 個人臉的特徵點如圖 2-7，分別是眉毛兩邊各五個點、鼻子共九個點、眼睛輪廓每邊各八個點、人臉輪廓共 19 個點，特別值得一提的是，為了要能模擬出人嘴巴的開闊，固在嘴巴的外緣設置了 12 個點，在內緣設置了 6 個點，圖 2-8 顯示了在不同表情變化下的特徵點位置。

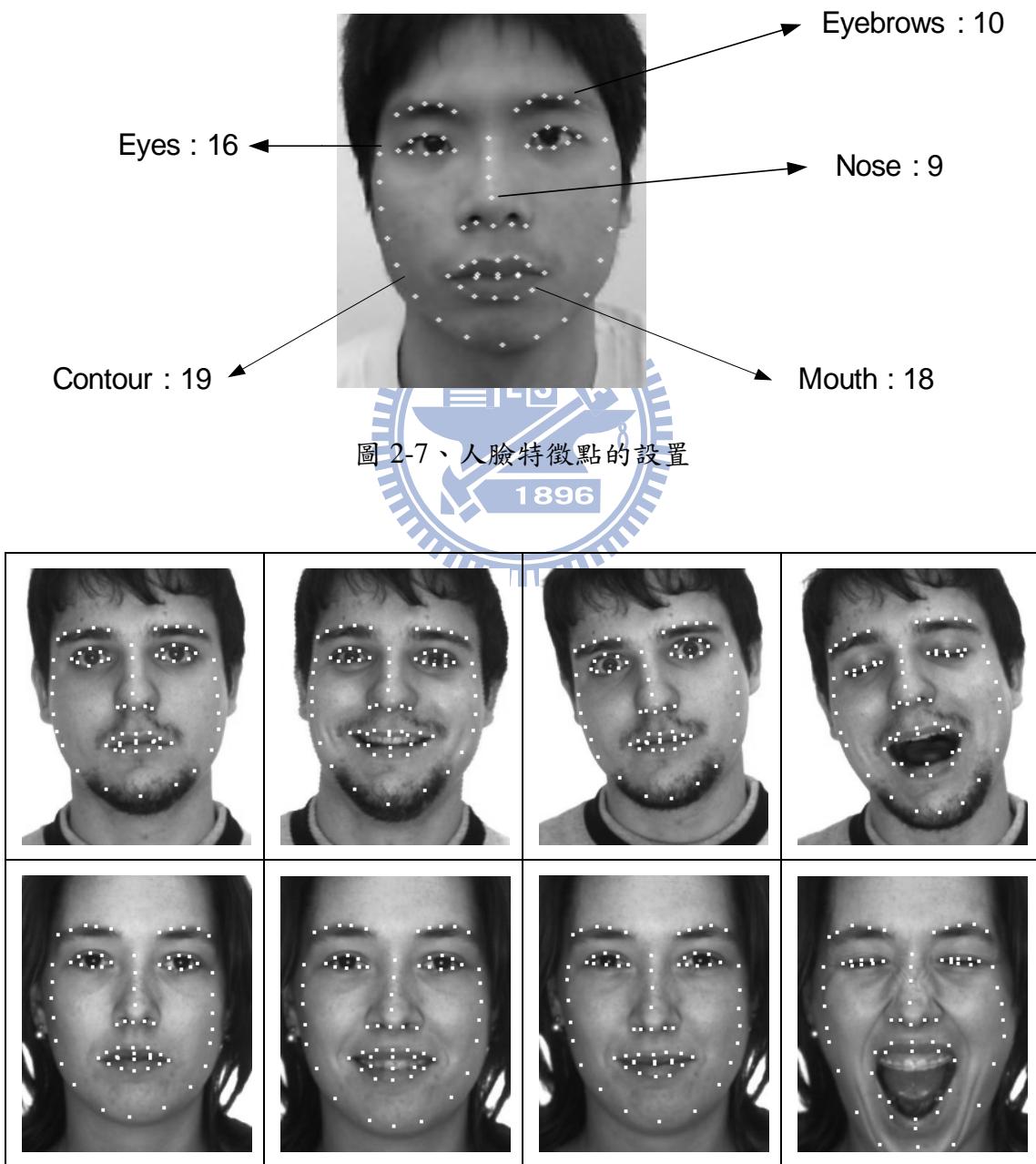


圖 2-8、不同表情的特徵點標點情形(使用 AR Database[23])

### 2.3.2. 形狀對齊

在一開始，我們需要藉由輸入的人臉特徵座標來組成人臉的形狀，由於訓練資料中的眾多人臉形狀包含了許多不同的變化，我們希望計算出他們的一組平均形狀來代表這些形狀的一個基底。

首先我們要定義出一個人臉的形狀，根據上一節所定義出的 70 個特徵點座標，我們將它依序來排列，就可以形成一個形狀(Shape)，如下所示：

$$s = (x_1, y_1, x_2, y_2, \dots, x_v, y_v), v = 70 \quad (2-8)$$

其中  $s$  是所得到的形狀， $x_i, y_i$  是特徵點的  $xy$  座標位置。

對所有 AAM 模型的訓練資料設置完成特徵點以後，由於每個輸入人臉形狀的大小、位置、旋轉角度皆不一致，我們首先需要對齊(Aligning Shapes)所有輸入的形狀，以得到我們想要的平均形狀(mean shape)，這裡是使用 Procrustes Analysis[18][24]來對齊形狀；我們必須對於所有輸入的形狀做縮放、位移以及旋轉來對齊，如(2-9)所示， $T_t(\mathbf{x})$  是經由縮放、位移以及旋轉的輸出， $a, b, c, d$  是縮放及旋轉的參數，而  $t_x, t_y$  則是  $x, y$  方向的位移。

$$T_t(\mathbf{x}) = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \mathbf{x} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \quad (2-9)$$

首先我們先對所有形狀取一個平均做為我們想要對齊形狀的目標形狀  $x'_i, y'_i$ ， $i=1, 2, \dots, 70$ ，而任一個輸入的形狀為  $x_i, y_i$ ， $i=1, 2, \dots, 70$ ，我們透過兩者的線性轉換關係(2-10)可以得到  $a, b, c, d, t_x, t_y$  的矩陣參數，將之帶入(2-9)式可以計算出轉正的輸入形狀。

$$\begin{bmatrix} S_{xx} & S_{xy} & S_x \\ S_{xy} & S_{yy} & S_y \\ S_x & S_y & n \end{bmatrix} \begin{bmatrix} a & c \\ b & d \\ t_x & t_y \end{bmatrix} = \begin{bmatrix} S_{xx'} & S_{xy'} \\ S_{yx'} & S_{yy'} \\ S_{x'} & S_{y'} \end{bmatrix}$$

$$S_{xx} = \sum x_i^2 \quad S_{yy} = \sum y_i^2 \quad S_x = \sum x_i \quad S_{yy} = \sum y_i$$

$$S_{xx'} = \sum x_i x'_i \quad S_{yy'} = \sum y_i y'_i \quad S_{x'} = \sum x'_i \quad S_{y'} = \sum y'_i$$

$$n = 70 \quad (2-10)$$

在這個過程中，我們一開始先計算出一組平均形狀來代表我們想要的目標形狀，接著藉由上述的方法來將縮放、平移、旋轉的效果去除，並計算出新的平均形狀出來，再重複同樣的步驟直至收斂。

在計算收斂的判定方面，我們藉由計算 Procrustes distance(2-11)來確定估測出的平均形狀的上一次與這次的變化是否夠小而收斂，在(2-11)中  $x_{j1}, y_{j1}$  為新的平均形狀中特徵點的座標  $x_{j0}, y_{j0}$  為前一個平均形狀中特徵點的座標。

$$P_d^2 = \sum_{j=1}^v [(x_{j1} - x_{j0})^2 + (y_{j1} - y_{j0})^2] \quad (2-11)$$

其整個流程如下：

1. 先將形狀座標中心點移到形狀中心(鼻子中心)
2. 由所有輸入形狀算出一個估測出的平均形狀(mean shape)
3. 藉由(2-9)與(2-10)計算出形變的形狀
4. 再由形變後的形狀估測出一個平均形狀(mean shape)
5. 迭代步驟 2~3 直到收斂。

訓練的形狀當中必須包含一些人臉的變化，在這裡我們取了 30 個人臉形狀做為範例。一開始我們將所有的人臉形狀重疊再一起顯示於圖 2-9(a)，其中每一個點所代表的是在這 30 個訓練人臉形狀當中的 x、y 座標，接著我們利用 Procrustes Analysis 形狀對齊去除縮放、平移、旋轉效果後將所有形狀重疊在一起的情形顯示於圖 2-9(b)，可以看到其中的眉毛以及眼睛的部分有較明顯的對齊結果，最後我們將對齊以後的所有形狀座標取平均值所算出的平均形狀(mean shape)顯示於圖 2-10。

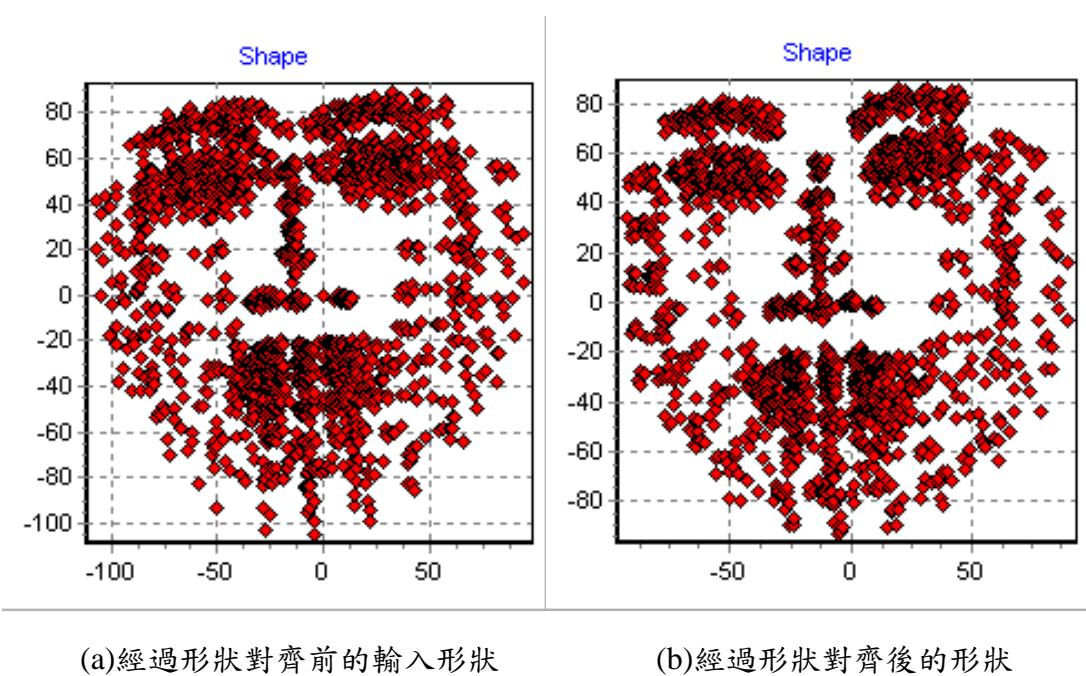


圖 2-9、經過形狀對齊前的輸入形狀與經過形狀對齊後的形狀

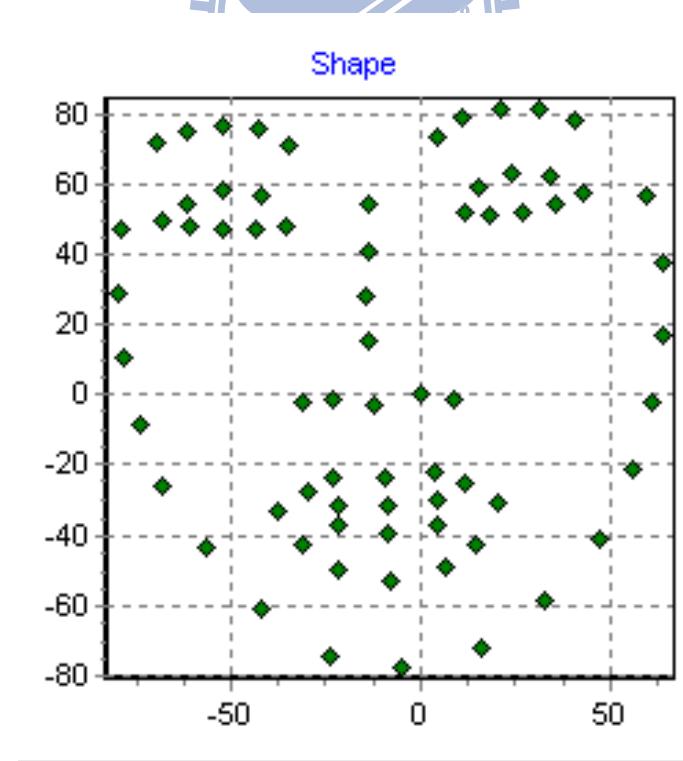


圖 2-10、由形狀對齊後的所有形狀計算出的平均形狀

### 2.3.3. 模擬臉部形狀的變化

在上一節中，我們找出對齊後的形狀(shape)以後，我們可以由式(2-8)得知每一個形狀(shape)可以組成一  $2v$  維的向量，每個形狀(shape)都是分布於這  $2v$  維的向量空間中的一個點，這些點所組成的一個範圍稱為允許形狀域(Allowable Shape Domain, ASD) [19]，在這個範圍裡所有的點都會與訓練資料相似，因此如果我們能夠建立出這些點分佈的模型，我們就可以建立出由這個模型所產生的一個位於 ASD 中的新的點，這些點就代表了各種不同的形狀。

為了要能夠找出每個人臉形狀的變化情形，來建立出人臉形狀變化的模型，我們使用了主成分分析(Principal Component Analysis, PCA)來簡化這  $2v$  綴度的資料以找出所有形狀之間的變化情形[18]，在這  $2$  綴的資料中，平均人臉形狀(mean shape)  $s_0$  是將所有的人臉相加取平均(2-12)也就是這個 ASD 的主軸，

$$s_0 = \frac{1}{n} \sum_{i=1}^n x_i, \quad x_i = \text{training shape set} \quad (2-12)$$

接著將所有的形狀減去平均形狀  $s_0$ ，計算出其整體的共變異矩陣(covariance matrix)S：

$$S = \frac{1}{n} \sum_{i=1}^n (x_i - s_0)(x_i - s_0)^T \quad (2-13)$$

接著由共變異矩陣 S 可以計算出它的特徵向量(eigenvector)和特徵值(eigenvalue)，其中特徵向量會由大到小來排列，而所有的特徵向量皆會是單範正交(orthonormal)的形式。

假設  $\Phi$ 是由前  $t$  大的特徵值對映到的特徵向量所組成，而我們就可以將任意訓練資料  $x$  趨近表示為

$$x \approx s_0 + \Phi b \quad (2-14)$$

其中  $\Phi = (\phi_1 | \phi_2 | \dots | \phi_t)$ ， $b$  是  $t$  綴的向量由(2-15)所得

$$b = \Phi^T(x - s_0) \quad (2-15)$$

由此(2-14)(2-15)的概念我們可以對於 Allowable Shape Domain 範圍中的任一形狀  $s$ ，建立出一個形狀的變化模型，

$$s = s_0 + \sum_{i=1}^n p_i s_i \quad (2-16)$$

其中  $s_i$   $i=1,2,\dots,n$  為前  $n$  大特徵值對映到的特徵向量， $p_i$   $i=1,2,\dots,n$  為權重值所組成的形變參數(shape parameter)。

在 Shape Model 所能表示整個變化的比率  $f_v$ ，是一個用來選擇特徵向量( $\lambda_i$ )的數目的依據，其計算方式是由選擇的前  $t$  大特徵值的總和除以所有特徵值的總和如(2-17)所示：

$$f_v = \frac{\sum_{i=1}^t \lambda_i}{\sum \lambda_i} \quad (2-17)$$

圖 2-10 顯示了人臉形狀模型的平均形狀(a)和前三個變形(b)(c)(d)的正負向量和大小(前三大特徵值的特徵向量)，其中在這裡為了方便觀察，這裡的向量大小放大 70 倍顯示。

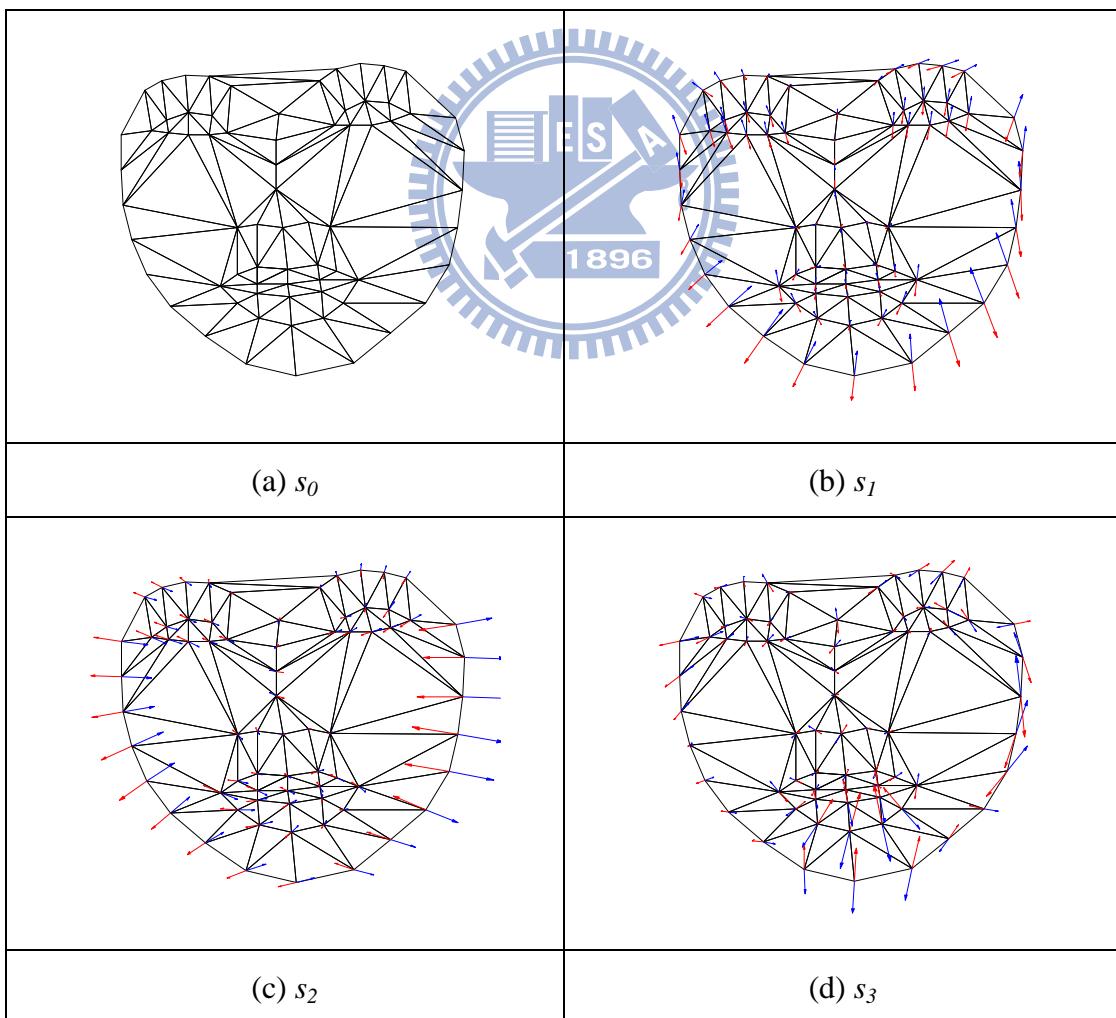


圖 2-11、人臉形狀模型的 mean shape(a)和前三個變形模型(b, c, d)

## 2.4. 人臉紋理模型

人臉的主動外觀模型中，還包括了紋理模型(Texture Model)，人臉紋理模型中，包含了一個平均人臉紋理(mean texture)以及平均人臉紋理可容許的變化組成，利用人臉紋理模型就可以模擬出不同的人臉紋理出來。

本節將探討人臉紋理的定義，以及對於不同人臉形狀的紋理如何形變來產生相同人臉形狀的紋理，以及最後是如何來模擬臉部紋理的變化情形。

### 2.4.1. 人臉的紋理

人臉的紋理特徵可以是原始影像的色彩像素、灰階像素或是經由其他影像處理之後所得到的像素數值所組成，(2-18)表示了紋理模型所組成的向量， $g_i$  為每個像素裡的紋理像素數值， $r$  為紋理模型總共的像素數量大小，這裡我們使用形狀模型裡的 mean shape 範圍所包含的所有像素數量(Texture pixel number)為紋理模型總共的像素數量。

$$g = (g_1, g_2, \dots, g_r) \quad r = \text{texture pixel number} \quad (2-18)$$



### 2.4.2. Piecewise Affine Warping

由於訓練人臉影像中，每個人臉的形狀、位置、大小都不盡相同，我們在建立人臉紋理模型時就必須將所有訓練資料中的人臉影像進行分段仿射變形(Piecewise Affine Warping)，在這個步驟中，我們使用了 Delaunay triangulation[25]來對人臉形狀做切割，在對切割後的各個對映的三角形將每張訓練資料的人臉紋理由原始形狀轉至平均形狀中。

Delaunay triangulation 是利用許多的三角形將一組輸入的點連接起來，它的特性就是每個三角形的外接圓內不會有其他點的存在，我們將這個方法將人臉形狀進行切割，並修改嘴巴部分以容許模擬出嘴唇開合的形狀以及將人臉形狀左右三角形對稱，如圖 2-13 所示。

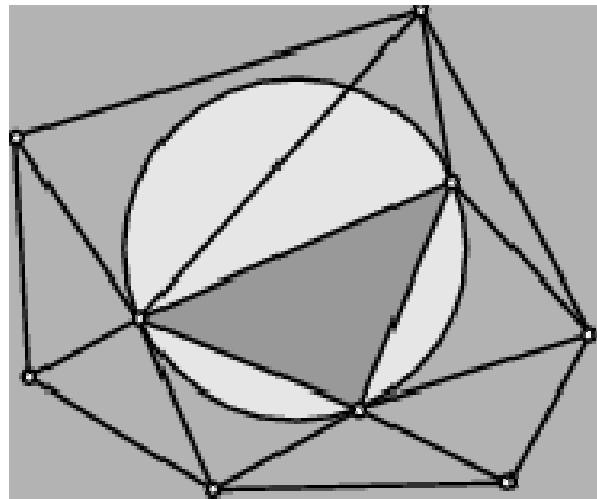


圖 2-12、Delaunay triangulation 的外接圓特性[21]

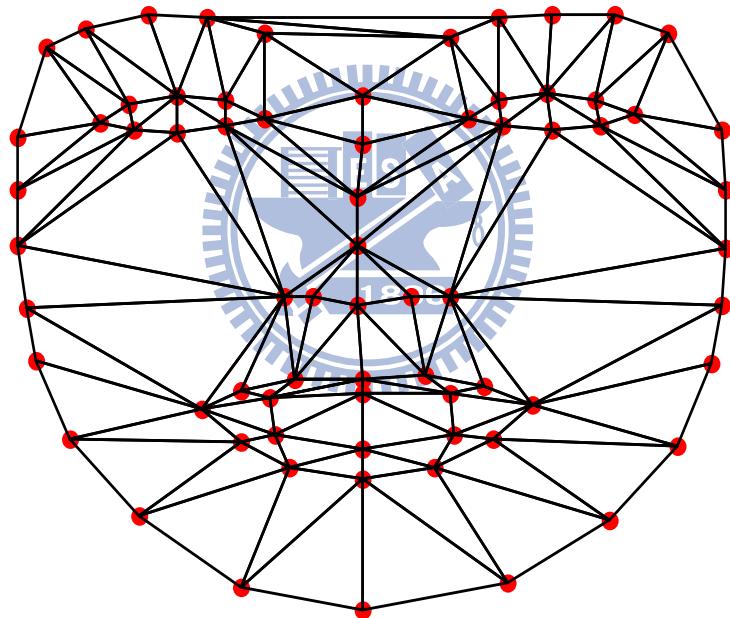


圖 2-13、人臉形狀以 Delaunay triangulation 表示

使用 Delaunay triangulation 將人臉形狀切割完成以後，我們要將人臉紋理值轉正，這裡要將訓練的人臉形狀依每個切割出的三角形與平均形狀中所對應的三角形進行轉換，將每個訓練的人臉形狀中三角形中的紋理像素值轉換到平均形狀中的對應三角形中。

在輸入形狀中， $\mathbf{x}_1(x_i, y_i)^T$ 、 $\mathbf{x}_2(x_j, y_j)^T$ 、 $\mathbf{x}_3(x_k, y_k)^T$ 是形狀的任一個 Delaunay 三角形中的三個頂點座標，三角形中的任意一點座標 $\mathbf{x}(x, y)^T$ 可以表示為：

$$\mathbf{x} = \mathbf{x}_1 + \alpha(\mathbf{x}_2 - \mathbf{x}_1) + \beta(\mathbf{x}_3 - \mathbf{x}_1) \quad (2-19)$$

其中  $0 \leq \alpha, \beta \leq 1$ ， $\alpha$  與  $\beta$  如下：

$$\begin{aligned} \alpha &= \frac{(x-x_i)(y_k-y_i)-(y-y_i)(x_k-x_i)}{(x_j-x_i)(y_k-y_i)-(y_j-y_i)(x_k-x_i)} \\ \beta &= \frac{(y-y_i)(x_j-x_i)-(x-x_i)(y_j-y_i)}{(x_j-x_i)(y_k-y_i)-(y_j-y_i)(x_k-x_i)} \end{aligned} \quad (2-20)$$

因此，我們利用  $\alpha$  與  $\beta$  值可以算出紋理像素值 $\mathbf{x}(x, y)^T$ 在平均形狀中的位置 $\mathbf{x}'(x', y')^T$ ，如圖 2-14：

$$\mathbf{x}' = \mathbf{x}_1' + \alpha(\mathbf{x}_2' - \mathbf{x}_1') + \beta(\mathbf{x}_3' - \mathbf{x}_1') \quad (2-21)$$

圖 2-15 是四張訓練影像的紋理形變結果，最上一排是四張經過手動標記特徵點的人臉影像；第二排是將輸入的人臉形狀以 Delaunay 三角形切割出來的人臉形狀結果；第三排是將輸入人臉影像灰階值在手動標記的特徵點範圍內取出，可以看到每張影像的形狀、位置、大小都不盡相同；最後第四排是將所有的人臉灰階影像轉換至平均形狀中，其中人臉的特徵位置(眼睛、鼻子、嘴巴……)都被轉至相同的地方。

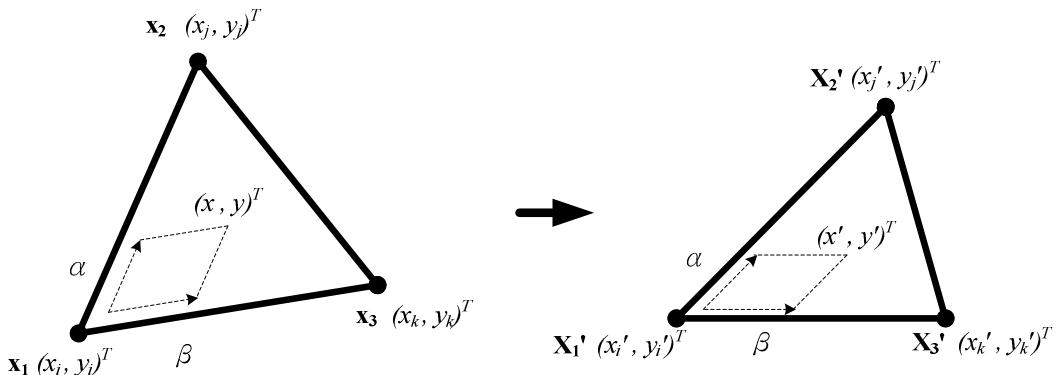


圖 2-14、三角形的形變

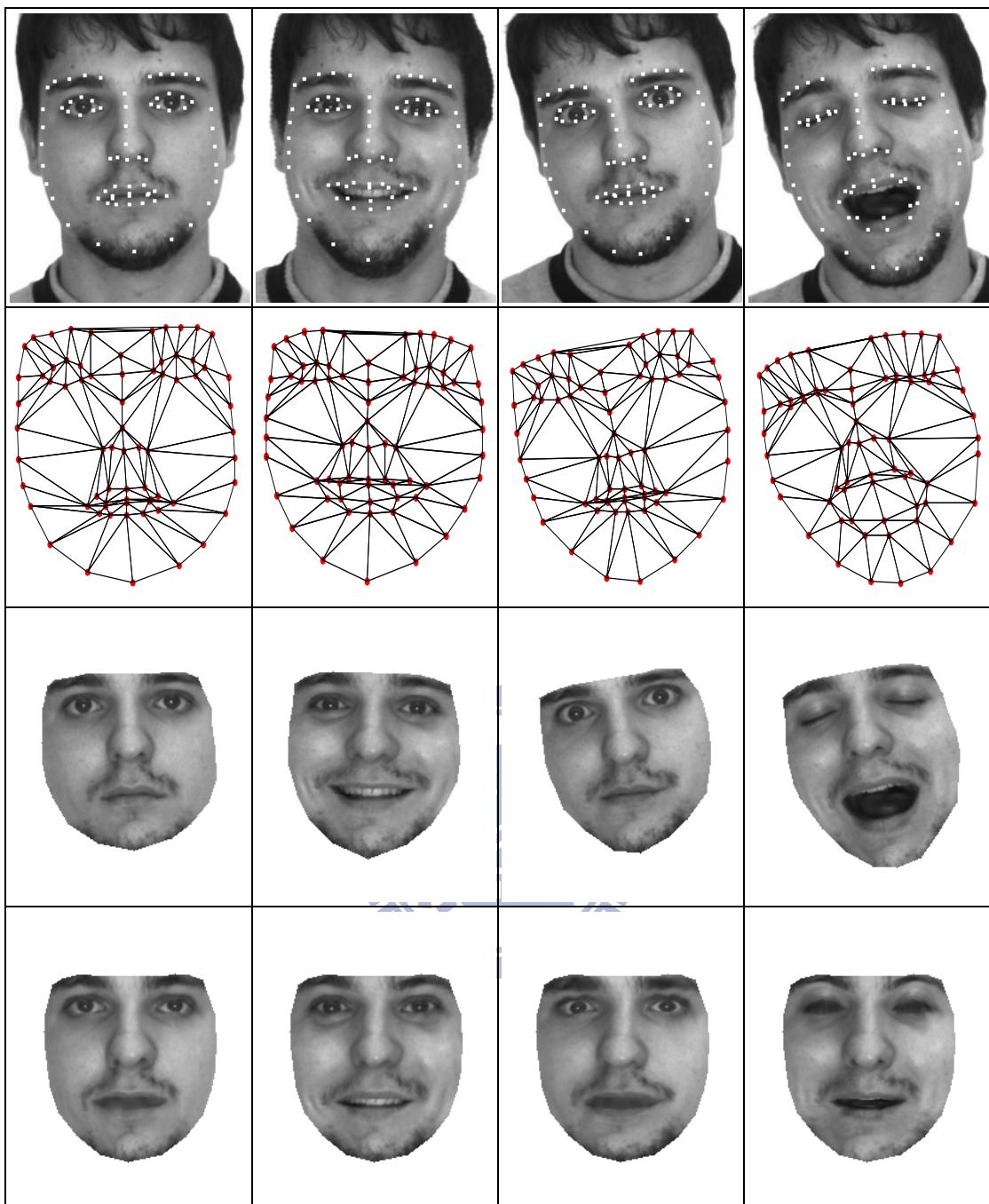


圖 2-15、人臉形變的情形

### 2.4.3. 模擬臉部紋理的變化

將人臉紋理影像轉正後，我們需要找出所有訓練人臉影像資料之中的紋理變化以建立臉部紋理模型，因此利用 Principal Component Analysis(PCA)來將訓練資料中的人臉紋理資料進行化簡，以找出人臉紋理的變化，首先先將所有轉正的人臉影像紋理相加，計算出平均紋理(mean texture)  $A_0$ ：

$$A_0 = \frac{1}{t} \sum_{i=1}^t g_i, \quad g_i = \text{training texture set} \quad (2-22)$$

接著將所有的紋理減去平均紋理  $A_0$ ，計算出其整體共變異矩陣(covariance matrix)G，

$$G = \frac{1}{t} \sum_{i=1}^t (g_i - A_0)(g_i - A_0)^T \quad (2-23)$$

計算出共變異矩陣 G 的特徵向量(eigenvector)和特徵值(eigenvalue)，特徵向量會由大到小來排列，其中所有的特徵向量會是單範正交(orthonormal)的形式。

利用與形狀變化模型相同的概念我們可以建立出一個紋理的變化模型：

$$A(x) = A_0(x) + \sum_{i=1}^m \lambda_i A_i(x) \quad \forall x \in S_0 \quad (2-24)$$

其中  $A_i(x) \quad i=1,2,\dots,n$  為前 n 大特徵值對映到的特徵向量， $\lambda_i \quad i=1,2,\dots,n$  為權重值所組成的形變參數 Texture parameter。

圖 2-16 顯示了(a)平均紋理  $A_0$  以及(b)、(c)、(d)、(e)前四個臉部紋理的變化模型  $A_1$ 、 $A_2$ 、 $A_3$ 、 $A_4$ (前三大特徵值的特徵向量)。



圖 2-16、人臉紋理模型的平均紋理(a)和前四個變化模型(b, c, d, e)

## 2.5. 影像校正演算法

到目前為止，我們已經建立出了主動外觀模型中的形狀模型以及紋理模型，接著，我們對於一個輸入的人臉影像，要找出這張影像的特徵出來，接下來將敘述輸入人臉的特徵擷取方法。

我們利用前面訓練出來的人臉形狀模型以及人臉紋理模型，即可藉由調整形狀參數(shape parameter)以及紋理參數(texture parameter)，即可得到模擬出的人臉；在這裡，我們可以利用這個模型來對輸入人臉影像做模擬，並可得到對準之人臉特徵位置。

在主動外觀模型的校正演算法中，大致分為 Independent AAM[22] 以及 Combine AAM[17]兩種方法，Independent AAM 是將形狀模型與紋理模型分開，分別計算出形狀參數(shape parameter)以及紋理參數(texture parameter)即可得到最後的模擬結果；而 Combine AAM 是將形狀模型與紋理模型整合為一個模型，

$$s = s_0 + \sum_{i=1}^l c_i s_i \quad (2-25)$$

$$A(x) = A_0(x) \sum_{i=1}^l c_i A_i(x) \quad (2-26)$$

藉由計算一個整合的參數  $c = (p_1, p_2, \dots, p_n, \lambda_1, \lambda_2, \dots, \lambda_m)^T$ ， $l \leq m+n$ ，來得到模擬的結果，Combine AAM 所需要的  $l \geq \max(m, n)$ ；而在 Independent AAM 主要是以形狀參數的計算為主，在計算量上 Combine AAM 所需要更多的計算量，因此，這裡選擇 Independent AAM 做為我們使用的方法，其中 Independent AAM 即是使用了 Lucas-Kanade 人臉校正演算法。

### 2.5.1. Lucas-Kanade 演算法

Lucas-Kanade 人臉校正演算法[22][26]主要是以影像梯度(Gradient)為基礎的影像變化估測演算法，這個方法常用於影像形變校正、物體追蹤等，我們將參考文獻[22][26]的方法再作改進，以達更佳的效果，接下來一開始將敘述在[22]以及[26]中所提出的 Lucas-Kanade 人臉校正演算法。

假定一個樣板影像(Template Image)  $A_0$  為形變前的影像而一個輸入影像  $I$  代表形變以後的目標影像，其中  $I$  與  $A_0$  存在有一形變的關係可以使  $I(W(x;p)) = A_0(x)$ ，因此我們希望透過的最小化來算出形變參數  $p$ 。

$$\sum_x [A_0(x) - I(W(x; p))]^2 \quad (2-27)$$

由於影像的變化為非線性，因此形變參數  $p$  無法直接求出，因此我們利用迭代的方式(2-28)，來將參數  $p$  逐漸找出最佳值。

$$\sum_x [A_0(x) - I(W(x; p + \Delta p))]^2, \quad p \leftarrow p + \Delta p \quad (2-28)$$

在最佳化計算的過程中， $\Delta p$  的估測值是以目前的  $p$  值為依據，因此將式(2-28)的  $I(W(x; p + \Delta p))$  對  $\Delta p \approx 0$  做泰勒展開可得：

$$I(W(x; p + \Delta p)) = I(W(x; p + 0)) + \nabla I \frac{\partial W}{\partial p} \Delta p + O(\Delta p^2) \quad (2-29)$$

其中  $\nabla I$  為  $I$  的影像梯度(gradient)值， $\frac{\partial W}{\partial p}$  為  $W$  的 Jacobian， $\nabla I \frac{\partial W}{\partial p}$  為 Steepest descent images；忽略高次項，將此代入式(2-28)中可得：

$$\sum_x \left[ A_0(x) - I(W(x; p)) - \nabla I \frac{\partial W}{\partial p} \Delta p \right]^2 \quad (2-30)$$

對  $\Delta p$  微分求極值可得：

$$2 \sum_x \left[ \nabla I \frac{\partial W}{\partial p} \right]^T \left[ A_0(x) - I(W(x; p)) - \nabla I \frac{\partial W}{\partial p} \Delta p \right] = 0 \quad (2-31)$$

$$\sum_x \left[ \nabla I \frac{\partial W}{\partial p} \right]^T \left[ \nabla I \frac{\partial W}{\partial p} \right] \Delta p = \sum_x \left[ \nabla I \frac{\partial W}{\partial p} \right]^T \left[ A_0(x) - I(W(x; p)) \right] \quad (2-32)$$

可求出  $\Delta p$ ：

$$\Delta p = H^{-1} \sum_x \left[ \nabla I \frac{\partial W}{\partial p} \right]^T \left[ A_0(x) - I(W(x; p)) \right] \quad (2-33)$$

其中  $H$  為 Hessian matrix：

$$H = \sum_x \left[ \nabla I \frac{\partial W}{\partial p} \right]^T \left[ \nabla I \frac{\partial W}{\partial p} \right] \quad (2-34)$$

Lucas-Kanade 演算法的迭代過程可分以下迭代九個步驟[26]，其整體流程圖如圖 2-17 所示：

迭代：

(1) 根據  $W(x; p)$  形變影像  $I$ ，計算出形變影像  $I(W(x; p))$

(2) 計算出誤差影像 (Error Image)  $A_0(x) - I(W(x; p))$

(3) 根據  $W(x; p)$  形變梯度影像  $\nabla I$

(4) 求出  $(x; p)$  的 Jacobian  $\frac{\partial W}{\partial p}$

(5) 計算出 steepest descent images  $\nabla I \frac{\partial W}{\partial p}$

(6) 由(2-34)計算出 Hessian matrix

(7) 計算  $\sum_x \left[ \nabla I \frac{\partial W}{\partial p} \right]^T [A_0(x) - I(W(x; p))]$

(8) 由(2-33)式計算出  $\Delta p$

(9) 更新參數  $p \leftarrow p + \Delta p$

直到  $\|\Delta p\|$  小於某閾值  $\epsilon$

在 Lucas-Kanade 演算法中，由於在每次的迭代過程中，Warping Jacobian，Steepest descent image 以及 Hessian matrix 都和形變參數  $p$  有關，需要在每一次的迭代過程中一再的進行計算，因此在運算量上會非常龐大，需要花費許多時間，因此下一小節將探討改良式的 Inverse Compositional Algorithm[20][22][26]。

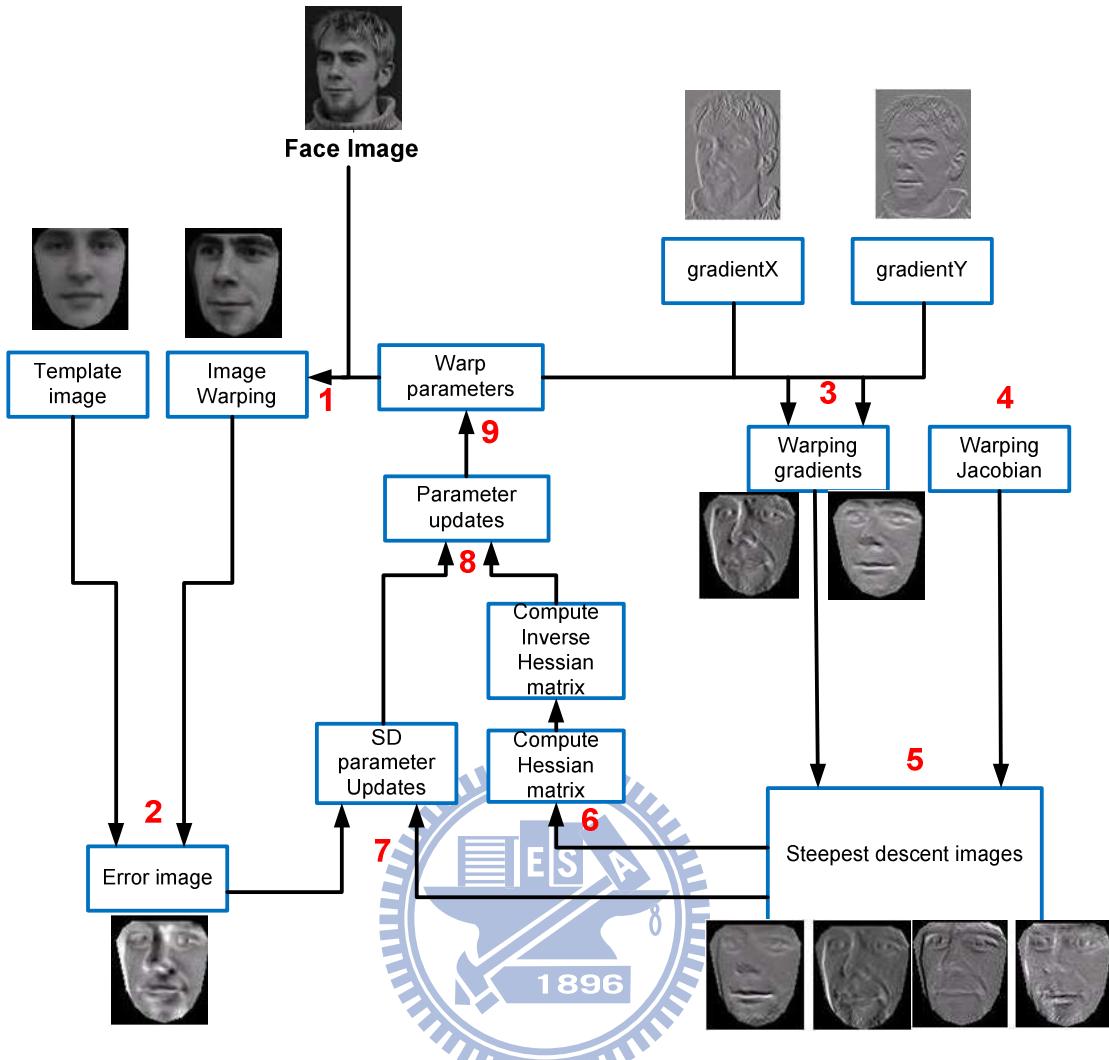


圖 2-17、Lucas-Kanade 演算法流程

## 2.5.2. Inverse Compositional 演算法

Inverse Compositional (IC) Algorithm [22][26]的是為了要避免讓所有要計算的過程均與形變參數  $p$  相依，這裡一開始先介紹[22]及[26]中所提出的方法。

這裡將式(2-28)改成：

$$\sum_x [I(W(x; p)) - A_0(W(x; \Delta p))]^2 \quad (2-35)$$

$$W(x; p) \leftarrow W(W(x; \Delta p)^{-1}; p) \quad (2-36)$$

我們將  $A_0(W(x; \Delta p))$  做泰勒展開可得：

$$A_0(W(x; \Delta p)) = A_0(W(x; 0)) + \nabla A_0 \frac{\partial W}{\partial p} \Delta p + O(\Delta p^2) \quad (2-37)$$

其中  $\nabla A_0$  為樣版影像的影像梯度(gradient)值， $\frac{\partial W}{\partial p}$  為  $W$  的 Jacobian， $\nabla A_0 \frac{\partial W}{\partial p}$  為

Steepest descent images；忽略高次項，將此代入式(2-35)中可得：

$$\sum_x \left[ I(W(x; p)) - A_0(W(x; 0)) - \nabla A_0 \frac{\partial W}{\partial p} \Delta p \right]^2 \quad (2-38)$$

對  $\Delta p$  微分求極值可得：

$$2 \sum_x \left[ \nabla A_0 \frac{\partial W}{\partial p} \right]^T \left[ I(W(x; p)) - A_0(W(x; 0)) - \nabla A_0 \frac{\partial W}{\partial p} \Delta p \right] = 0 \quad (2-39)$$

$$\sum_x \left[ \nabla A_0 \frac{\partial W}{\partial p} \right]^T \left[ \nabla A_0 \frac{\partial W}{\partial p} \right] \Delta p = \sum_x \left[ \nabla A_0 \frac{\partial W}{\partial p} \right]^T \left[ I(W(x; p)) - A_0(W(x; 0)) \right] \quad (2-40)$$

其中令  $A_0(W(x; 0)) = A_0(x)$  可求出  $\Delta p$ ：

$$\Delta p = H^{-1} \sum_x \left[ \nabla A_0 \frac{\partial W}{\partial p} \right]^T \left[ I(W(x; p)) - A_0(x) \right] \quad (2-41)$$

其中  $H$  為 Hessian matrix：

$$H = \sum_x \left[ \nabla A_0 \frac{\partial W}{\partial p} \right]^T \left[ \nabla A_0 \frac{\partial W}{\partial p} \right] \quad (2-42)$$

  
在 Inverse Compositional 演算法與 Lucas-Kanade 演算法的差異中，Lucas-Kanade 演算法要計算出輸入影像的梯度值(gradient)，接著再計算出 Warping Jacobian 並求出 Steepest descent image 以及 Hessian matrix，在每次迭代過程中都需要重複計算這一個部分；而在 Inverse Compositional 演算法當中，是使用樣版影像  $A_0$  去計算出影像的梯度值，由於樣版影像是已知的，因此之後的 Warping Jacobian 以及 Steepest descent image 和 Hessian matrix 都是已知的；IC 演算法只要先在迭代前計算出樣版影像的梯度值，Warping Jacobian 以及 Steepest descent image 和 Hessian matrix 都可以預先求出，而迭代時就可以省去這一部分的計算量，因此迭代的速度可以提升許多，並且在迭代的結果也不會降低它的成功率，因此我們論文選擇此改良方法來做我們的影像校正方法。

Inverse Compositional 演算法的迭代過程可分前處理 4 個步驟(I~IV)以及迭代 5 個步驟((1)~(5))[22]，為了方便區分前處理以及迭代的步驟，前處理 4 個步驟以 (I~IV) 表示，而迭代 5 個步驟以 ((1)~(5)) 表示，其整體流程圖如圖 2-18 所示：

前處理：

I. 根據樣板影像  $A_0(x)$  計算出梯度影像  $\nabla A_0$

II. 求出  $(x; 0)$  的 Jacobian  $\frac{\partial W}{\partial p}$

III. 計算出 steepest descent images  $\nabla A_0 \frac{\partial W}{\partial p}$

IV. 由(2-42)計算出 Hessian matrix

迭代：

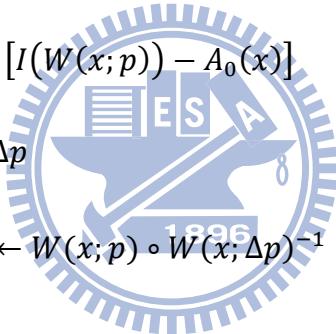
(1). 根據  $W(x; p)$  形變影像  $I$ ，計算出形變影像  $I(W(x; p))$

(2). 計算出誤差影像(Error Image)  $I(W(x; p)) - A_0(x)$

(3). 計算  $\sum_x \left[ \nabla A_0 \frac{\partial W}{\partial p} \right]^T [I(W(x; p)) - A_0(x)]$

(4). 由(2-41)式計算出  $\Delta p$

(5). 更新參數  $W(x; p) \leftarrow W(x; p) \circ W(x; \Delta p)^{-1}$



在這裡使用的 Inverse Compositional 演算法由於把部分要計算的東西拿到了前處理的步驟中，所以在迭代的過程中較 Lucas-Kanade 演算法少了四項計算，因此可以減少不少的運算量。

接下來介紹將 Inverse Compositional 演算法套用在 AAM 的方法[22]，由於形變的部分已於 2.4.2 介紹過，因此這裡將要介紹 Warp Jacobian[27]，假定

$W(x; p) = W(W_x(x; p), W_y(x; p))^T$ ，則：

$$\frac{\partial W}{\partial p} = \begin{bmatrix} \frac{\partial W_x}{\partial p_1} & \frac{\partial W_x}{\partial p_2} & \cdots & \frac{\partial W_x}{\partial p_M} \\ \frac{\partial W_y}{\partial p_1} & \frac{\partial W_y}{\partial p_2} & \cdots & \frac{\partial W_y}{\partial p_M} \end{bmatrix} \quad (2-43)$$

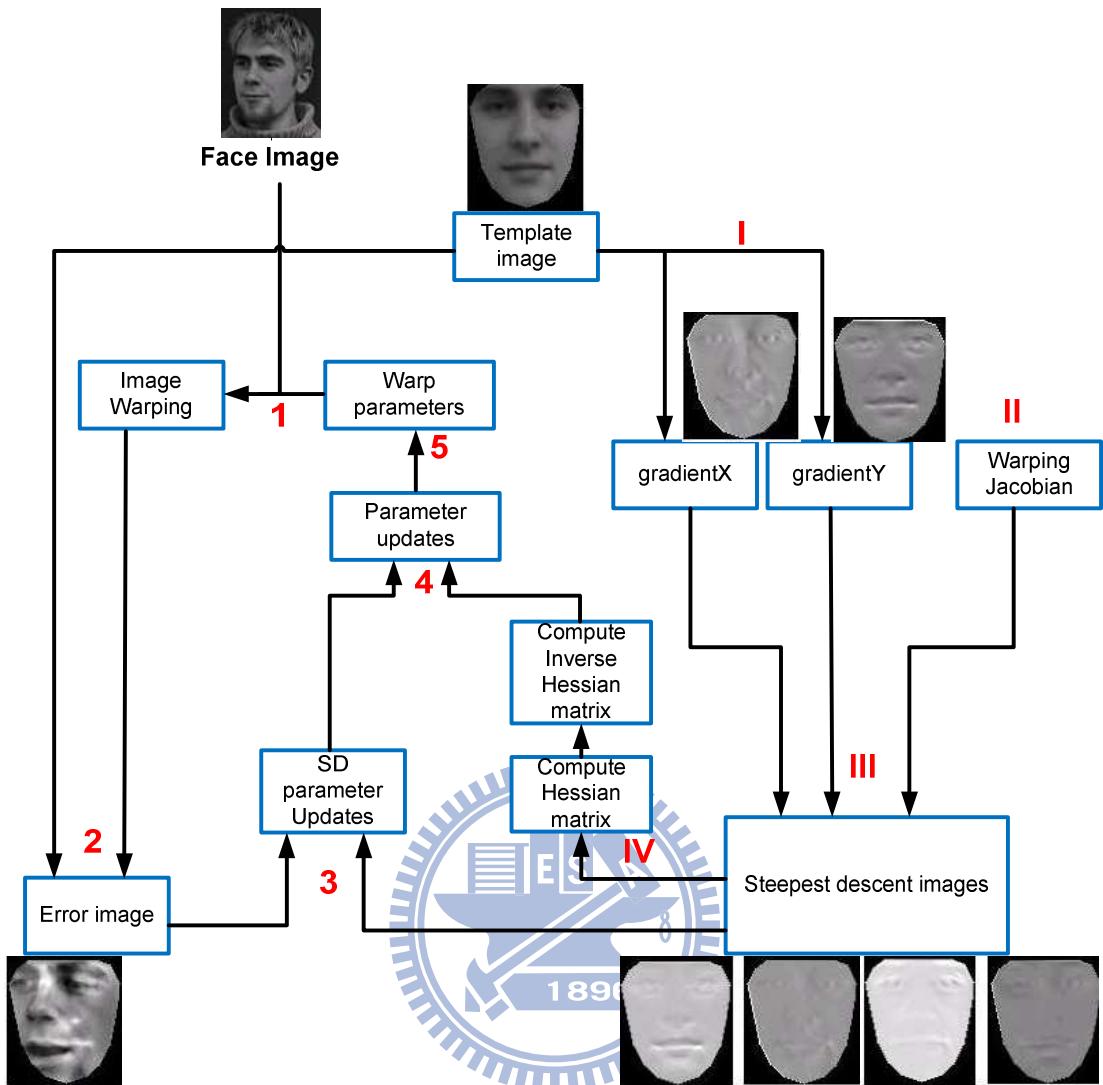


圖 2-18、Inverse Compositional 演算法流程

其中  $W_x$  表示為形變位置  $W(x; p)$  在  $x$  軸的座標， $p_m$  是代表形變參數，利用 chain rule 可以將  $\frac{\partial W_x}{\partial p_m}$  表示為：

$$\frac{\partial W_x}{\partial p_m} = \sum_{t=1}^v \left[ \frac{\partial W_x}{\partial x_t} \frac{\partial x_t}{\partial p_m} + \frac{\partial W_x}{\partial y_t} \frac{\partial y_t}{\partial p_m} \right] \quad (2-44)$$

同樣的將  $\frac{\partial W_y}{\partial p_m}$  表示為：

$$\frac{\partial W_y}{\partial p_m} = \sum_{t=1}^v \left[ \frac{\partial W_y}{\partial x_t} \frac{\partial x_t}{\partial p_m} + \frac{\partial W_y}{\partial y_t} \frac{\partial y_t}{\partial p_m} \right] \quad (2-45)$$

我們改寫(2-19)式可得：

$$\mathbf{x} = (x^0, y^0) = (1 - \alpha - \beta)(x_i^0, y_i^0)^T + \alpha(x_j^0, y_j^0)^T + \beta(x_k^0, y_k^0)^T$$

其中  $(x_i^0, y_i^0)^T$ 、 $(x_j^0, y_j^0)^T$ 、 $(x_k^0, y_k^0)^T$  為形變前影像中三角形的三個頂點。

同理，當形變到另一個影像時，可表示為：

$$W(x; p) = (1 - \alpha - \beta)(x_i, y_i)^T + \alpha(x_j, y_j)^T + \beta(x_k, y_k)^T$$

其中  $(x_i, y_i)^T$ 、 $(x_j, y_j)^T$ 、 $(x_k, y_k)^T$  為三角形形變後影像，分別對三個頂點微分可得：

$$\frac{\partial W_x}{\partial x_i} = \frac{\partial W_y}{\partial y_i} = 1 - \alpha - \beta \quad (2-46)$$

$$\frac{\partial W_x}{\partial x_j} = \frac{\partial W_y}{\partial y_j} = \alpha \quad (2-47)$$

$$\frac{\partial W_x}{\partial x_k} = \frac{\partial W_y}{\partial y_k} = \beta \quad (2-48)$$

$$\frac{\partial W_x}{\partial y_i} = \frac{\partial W_y}{\partial x_i} = \frac{\partial W_x}{\partial y_j} = \frac{\partial W_y}{\partial x_j} = \frac{\partial W_x}{\partial y_k} = \frac{\partial W_y}{\partial x_k} = 0 \quad (2-49)$$

由上式可知，當  $z \neq i, j, k$  時， $\frac{\partial W_x}{\partial x_z} = \frac{\partial W_y}{\partial y_z} = 0$ ，因此 (2-44)(2-45) 可改寫為：

$$\frac{\partial W_x}{\partial p_m} = \sum_{t=1}^v \left[ \frac{\partial W_x}{\partial x_t} \frac{\partial x_t}{\partial p_m} + \frac{\partial W_x}{\partial y_t} \frac{\partial y_t}{\partial p_m} \right] = (1 - \alpha - \beta) \frac{\partial x_i}{\partial p_m} + \alpha \frac{\partial x_j}{\partial p_m} + \beta \frac{\partial x_k}{\partial p_m} \quad (2-50)$$

$$\frac{\partial W_y}{\partial p_m} = \sum_{t=1}^v \left[ \frac{\partial W_y}{\partial x_t} \frac{\partial x_t}{\partial p_m} + \frac{\partial W_y}{\partial y_t} \frac{\partial y_t}{\partial p_m} \right] = (1 - \alpha - \beta) \frac{\partial y_i}{\partial p_m} + \alpha \frac{\partial y_j}{\partial p_m} + \beta \frac{\partial y_k}{\partial p_m} \quad (2-51)$$

我們將多個訓練人臉形狀模型(2-16)以矩陣的方式排列，可得：

$$[s_1 \ s_2 \ \dots \ s_n] = \begin{bmatrix} s_1^{x_1} & s_2^{x_1} & \dots & s_n^{x_1} \\ s_1^{y_1} & s_2^{y_1} & \dots & s_n^{y_1} \\ \vdots & \ddots & \ddots & \vdots \\ s_1^{x_v} & s_2^{x_v} & \dots & s_n^{x_v} \\ s_1^{y_2} & s_2^{y_2} & \dots & s_n^{y_2} \end{bmatrix} \quad (2-52)$$

其中  $s_m^{x_t}$  與  $s_m^{y_t}$  為一個人臉形狀在第  $m$  個形狀變化特徵向量的第  $t$  個  $x$  與  $y$  特徵點的位置， $\frac{\partial x_t}{\partial p_m}$  與  $\frac{\partial y_t}{\partial p_m}$  可由微分(2-16)得到：

$$\frac{\partial x_t}{\partial p_m} = s_m^{x_t} \quad (2-53)$$

$$\frac{\partial y_t}{\partial p_m} = s_m^{y_t} \quad (2-54)$$

寫成矩陣型式為：

$$\begin{bmatrix} \frac{\partial x_1}{\partial p_1} & \frac{\partial x_1}{\partial p_2} & \dots & \frac{\partial x_1}{\partial p_n} \\ \frac{\partial y_1}{\partial p_1} & \frac{\partial y_1}{\partial p_2} & \dots & \frac{\partial y_1}{\partial p_n} \\ \vdots & \ddots & \vdots & \vdots \\ \frac{\partial x_v}{\partial p_1} & \frac{\partial x_v}{\partial p_2} & \dots & \frac{\partial x_v}{\partial p_n} \\ \frac{\partial y_v}{\partial p_1} & \frac{\partial y_v}{\partial p_2} & \dots & \frac{\partial y_v}{\partial p_n} \end{bmatrix} = \frac{\partial \begin{bmatrix} p_1 s_1^{x_1} + p_2 s_2^{x_1} + \dots + p_n s_n^{x_1} \\ p_1 s_1^{y_1} + p_2 s_2^{y_1} + \dots + p_n s_n^{y_1} \\ \dots \\ p_1 s_1^{x_v} + p_2 s_2^{x_v} + \dots + p_n s_n^{x_v} \\ p_1 s_1^{y_v} + p_2 s_2^{y_v} + \dots + p_n s_n^{y_v} \end{bmatrix}}{\partial \mathbf{p}} = \begin{bmatrix} s_1^{x_1} & s_2^{x_1} & \dots & s_n^{x_1} \\ s_1^{y_1} & s_2^{y_1} & \dots & s_n^{y_1} \\ \vdots & \ddots & \ddots & \vdots \\ s_1^{x_v} & s_2^{x_v} & \dots & s_n^{x_v} \\ s_1^{y_v} & s_2^{y_v} & \dots & s_n^{y_v} \end{bmatrix} \quad (2-55)$$

將其帶入(2-50)(2-51)可以計算出 Warping Jacobian :

$$\frac{\partial W_x}{\partial p_m} = (1 - \alpha - \beta) s_m^{x_i} + \alpha s_m^{x_j} + \beta s_m^{x_k} \quad (2-56)$$

$$\frac{\partial W_y}{\partial p_m} = (1 - \alpha - \beta) s_m^{y_i} + \alpha s_m^{y_j} + \beta s_m^{y_k} \quad (2-57)$$

在 Inverse Compositional 演算法中我們要更新形變參數時，我們是由反向  $W(x; \Delta p)$  來計算出  $W(x; \Delta p)^{-1}$ ，其中：

$$W(x; \Delta p) = W(x; 0) + \frac{\partial W}{\partial p} \Delta p = x + \frac{\partial W}{\partial p} \Delta p + O(\Delta p^2) \quad (2-58)$$

其中  $W(x; 0) = x$  是單位形變(identity warp)，

$$W(x; \Delta p) \circ W(x; -\Delta p) = x - \frac{\partial W}{\partial p} \Delta p + \frac{\partial W}{\partial p} \Delta p = x + O(\Delta p^2) \quad (2-59)$$

因此根據第一項，我們可得：

$$W(x; \Delta p)^{-1} = W(x; -\Delta p) \quad (2-60)$$

當輸入人臉影像時，我們會先將平均形狀  $s_0$  放入影像中，接這再由更新形變參數的方式，來從平均形狀做參數的調整，因此(2-60)可以由(2-16)的線性組合方式表示：

$$\Delta s_0 = - \sum_{i=1}^n \Delta p_i s_i \quad (2-61)$$

其中  $\Delta s_0 = (\Delta x_1^0, \Delta y_1^0, \dots, \Delta x_v^0, \Delta y_v^0)^T$  是  $W(x; \Delta p)^{-1}$  由平均形狀所的改變量，為了要將  $W(x; \Delta p)^{-1}$  更新到  $W(x; p)$ ，我們要計算出目前形狀位置的變化量

$\Delta s = (\Delta x_1, \Delta y_1, \dots, \Delta x_v, \Delta y_v)^T$ ，知道置些位置以後，我們就可以由(2-16)計算出  $W(x; p) \circ W(x; \Delta p)^{-1}$  的參數：

$$p'_i = s_i \cdot (s + \Delta s - s_0) \quad (2-62)$$

其中  $p'_i$  是  $W(x; p) \circ W(x; \Delta p)^{-1}$  的第  $i$  個形狀參數，二這裡是內積相乘，而形狀向量  $s_i$  是單範正交 (orthonormal)。

我們要從  $\Delta s_0$  計算出  $\Delta s$  考慮一個人臉形狀中的第  $i$  個頂點，我們必須由平均形狀  $s_0$  中的  $(\Delta x_i^0, \Delta y_i^0)^T$  計算出目前形狀  $s$  中的  $(\Delta x_i, \Delta y_i)^T$ ，如圖 2-19 是利用 Delaunay 三角形的方式將平均形狀  $s_0$  透過 affine warp 轉換至目前形狀  $s$ ，要由  $(\Delta x_i^0, \Delta y_i^0)^T$  計算  $(\Delta x_i, \Delta y_i)^T$  必須由三角形 affine warp 從  $(x_i^0, y_i^0)^T + (\Delta x_i^0, \Delta y_i^0)^T$  得到  $(x_i, y_i)^T + (\Delta x_i, \Delta y_i)^T$ ，但是這裡每個特徵點所相鄰的三角形不會只有一個，使用不同三角形所計算出的結果會不相同，因此，在這裡我們是使用計算所有相鄰三角形所得的結果取平均如此可以使人臉形狀結果較平順。

之前所描述的 Inverse Compositional 演算法沒有考慮到 independent AAM 裡的紋理變化，在這裡使用了相同的演算法加入紋理的變化模式[22]：

$$\sum_{x \in s_0} \left[ A_0(x) + \sum_{i=1}^m \lambda_i A_i(x) - I(W(x; p)) \right]^2 = \left\| A_0(x) + \sum_{i=1}^m \lambda_i A_i(x) - I(W(x; p)) \right\|^2 \quad (2-63)$$

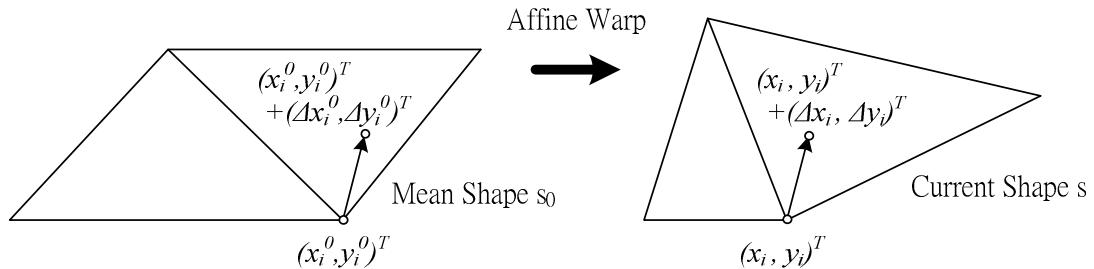


圖 2-19、 $(x_i^0, y_i^0)^T + (\Delta x_i^0, \Delta y_i^0)^T$  經由 Affine warp 得到  $(x_i, y_i)^T + (\Delta x_i, \Delta y_i)^T$

我們就由轉化上式以滿足 Inverse Compositional 演算法的定義，因此我們要轉化為求最小化[26]：

$$\left\| A_0(x) + I(W(x; p)) \right\|_{\text{span}(A_i)^\perp}^2 + \left\| A_0(x) + \sum_{i=1}^m \lambda_i A_i(x) - I(W(x; p)) \right\|_{\text{span}(A_i)}^2 \quad (2-64)$$

其中  $\text{span}(A_i)$  表示由紋理的特徵向量形成的線性子空間，而  $\text{span}(A_i)^\perp$  則是表示有正交補集的線性子空間，(2-64)的前半部分只與形狀變化參數  $p$  有關，而後半部分對於形狀參數  $p$  的最小值均為 0，因此採用 Inverse Compositional 演算法先求出前半部分的形變參數  $p$ ，接著再將  $p$  當作常數，而求出後半部的紋理參數  $\lambda_i$ ，由於  $A_i$  是單範正交的(orthonormal)，因此紋理參數可表示為[22]：

$$\lambda_i = \sum_{x \in s_0} A_i(x) \cdot [I(W(x; p)) - A_0(x)] \quad (2-65)$$

由於 Inverse Compositional 演算法是藉由  $\text{span}(A_i)^\perp$  來運算必須先將  $\nabla A_0 \frac{\partial W}{\partial p}$  投影至  $\text{span}(A_i)^\perp$  來做運算，而 steepest descent image 投影至  $\text{span}(A_i)^\perp$  如下式：

$$SD_j(x) = \nabla A_0 \frac{\partial W}{\partial p_j} - \sum_{i=1}^m \left[ \sum_{x \in s_0} A_i(x) \cdot \nabla A_0 \frac{\partial W}{\partial p_j} \right] A_i(x) \quad (2-66)$$

其 Inverse Compositional 演算法整體流程如下[22]，為了方便區分前處理以及迭代的步驟，前處理 4 個步驟以(I~IV)表示，而迭代 5 個步驟以((1)~(5))表示：

前處理：

I. 根據樣板影像  $A_0(x)$  計算出梯度影像  $\nabla A_0$

II. 求出  $(x; 0)$  的 Jacobian  $\frac{\partial W}{\partial p}$

III. 由(2-66)計算出改變的 steepest descent images  $SD(x)$

IV. 由 III. 的  $SD(x)$  計算出 Hessian matrix =  $\sum_x SD(x)^T SD(x)$

迭代：

(1). 根據  $W(x; p)$  形變影像  $I$ ，計算出形變影像  $I(W(x; p))$

(2). 計算出誤差影像(Error Image)  $I(W(x; p)) - A_0(x)$

(3). 計算  $\sum_x SD(x)^T [I(W(x; p)) - A_0(x)]$

(4). 由計算出  $\Delta p = H^{-1} \sum_x SD(x)^T [I(W(x; p)) - A_0(x)]$

(5). 更新參數  $W(x; p) \leftarrow W(x; p) \circ W(x; \Delta p)^{-1}$

迭代結束：

由(2-65)計算出紋理變化參數  $\lambda_i$ 。

### 2.5.3. 整體形狀正規化轉換

在一般的傳統 AAM 演算法中，對人臉偏轉的角度容忍度十分有限，而容易對於有偏轉角度的人臉造成迭代的錯誤；而另一方面，我們在訓練人臉形狀變化模型之前，會先經由 Procrustes Analysis[18][24]來對人臉做對齊，而移除了一些位移、縮放和旋轉的資訊，因此，在這裡我們必須進行整體形狀正規化轉換(Global Shape Normalizing Transform)[22]來得得到實際人臉所擁有的位移、縮放和旋轉資訊。

這裡定義了  $N(x; q)$  為訓練資料的 Global shape normalizing transform，平移為  $(t_x, t_y)$ ，旋轉角度  $\theta$ ，縮放  $k$  倍，因此可以利用一個參數  $q = (a, b, t_x, t_y)^T$  來代表其形變參數，其中  $a = k \cos \theta - 1$ ， $b = k \sin \theta$ ， $N(x; q)$  如下所示：

$$N(x; q) = \begin{bmatrix} (1+a) & -b \\ b & (1+a) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \quad (2-67)$$

在包含之前的形變與 Global shape normalizing transform，其最後得到的人臉形狀為  $N(W(x; p); q) \forall x \in s_0$ ，對於平均形狀(Mean shape) $s_0$  中的任一點  $x_0$ ，其在最後的人臉形狀位置也是  $N(W(x_0; p); q)$ ，而其所對應到的紋理值為  $A_0(x_0)$ ，在  $N(W(x_0; p); q)$  中也應當是  $A_0(x_0)$ ，因此 AAM 模型可表示為：

$$M(N(W(x_0; p); q)) = A(x) = A_0(x) + \sum_{i=1}^m \lambda_i A_i(x) \quad (2-68)$$

其中  $M$  代表和形狀  $N(W(x_0; p); q)$  相同大小的二維影像。

當給定一輸入影像  $I(x)$ ，需要讓  $M(N(W(x_0; p); q))$  與之匹配，讓兩者的差異最小化，因此必須讓其最小化：

$$\sum_{x \in s_0} [A_0(x) + \sum_{i=1}^m \lambda_i A_i(x) - I(N(W(x; p); q))]^2 \quad (2-69)$$

為了與之前的形狀模型(2-16)搭配運算，需要建立一組線性組合的係數[22]。

若 mean shape  $s_0 = (x_1^0, y_1^0, \dots, x_v^0, y_v^0)^T$ ，則假定  $s_1^* = s_0 = (x_1^0, y_1^0, \dots, x_v^0, y_v^0)^T$ ，  
 $s_2^* = (-y_1^0, x_1^0, \dots, -y_v^0, x_v^0)^T$ ， $s_3^* = (1, 0, \dots, 1, 0)^T$ ， $s_4^* = (0, 1, \dots, 0, 1)^T$ ，如圖 2-20，  
並將  $N(x; q)$  表示為：

$$N(x; q) = s_0 + \sum_{i=1}^4 q_i s_i^* \quad (2-70)$$

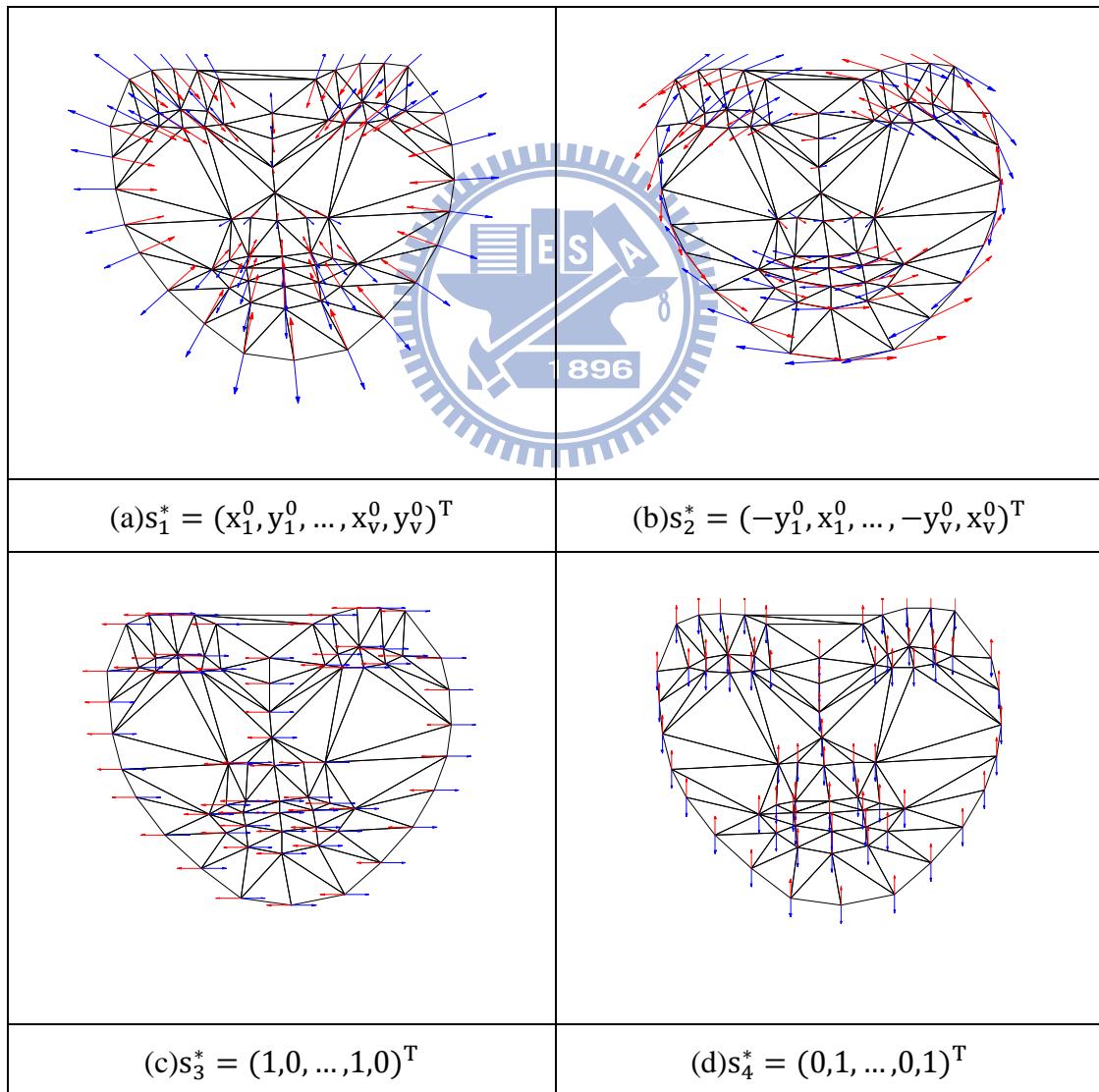


圖 2-20、Global shape normalizing transform 形狀變化模型

可以將(2-70)來計算 Piecewise affine warping 以及 Warping Jacobian，其計算方法與(2-16)計算的方法是相同的。

在形狀參數更新的方面，在(2-60)以知  $W(x; \Delta p)^{-1} = W(x; -\Delta p)$ ，這裡需要產生  $N \circ W$  的形狀參數，則參數可表示為  $(\Delta q, \Delta p)$  可表示為：

$$N \circ W(x; \Delta q, \Delta p)^{-1} = N \circ W(x; -\Delta q, -\Delta p) \quad (2-71)$$

故合成形式為：

$$\begin{aligned} (N \circ W)(x; q, p) \circ (N \circ W)(x; \Delta q, \Delta p)^{-1} \\ \approx (N \circ W)(x; q, p) \circ (N \circ W)(x; -\Delta q, -\Delta p) \end{aligned} \quad (2-72)$$

要找出  $(N \circ W)(x; -\Delta q, -\Delta p)$ ，其方法類似於之前的方法，根據(2-61)可以將此部分改寫為：

$$\Delta s_0 = -\sum_{i=1}^n \Delta p_i s_i - \sum_{j=1}^4 \Delta q_j s_j^* \quad (2-73)$$

這裡定義了：

$$N \circ W(s_0; q, p) = s^\dagger \quad (2-74)$$

則

$$N \circ W(s_0; q, p) = s^\dagger = N \left( s_0 + \sum_{i=1}^n p_i s_i; q \right) = s_0 + \sum_{i=1}^n p_i s_i + \sum_{j=1}^4 q_j s_j^* \quad (2-75)$$

利用正交的關係，可以計算出  $(N \circ W)(x; q, p) \circ (N \circ W)(x; \Delta q, \Delta p)^{-1}$  的形狀參數：

$$q_i = s_i^* \cdot (s^\dagger - s_0) \quad (2-76)$$

$$p_i = s_i \cdot (N(s^\dagger; q)^{-1} - s_0) \quad (2-77)$$

在紋理方面，這裡使用和之前一樣的方法計算  $(q_1, q_2, q_3, q_4)$  的 steepest descent image：

$$SD_j(x) = \nabla A_0 \frac{\partial N}{\partial q_j} - \sum_{i=1}^m \left[ \sum_{x \in S_0} A_i(x) \cdot \nabla A_0 \frac{\partial N}{\partial q_j} \right] A_i(x) \quad (2-78)$$

與  $p$  參數的 steepest descent image ,  $j=1,2,\dots,n$  :

$$SD_{j+4}(x) = \nabla A_0 \frac{\partial W}{\partial p_j} - \sum_{i=1}^m \left[ \sum_{x \in S_0} A_i(x) \cdot \nabla A_0 \frac{\partial W}{\partial p_j} \right] A_i(x) \quad (2-79)$$

同樣的 Hessian matrix 的計算為 :

$$H = \sum_x SD(x)^T SD(x) \quad (2-80)$$

同樣的紋理參數可表示為 :

$$\lambda_i = \sum_{x \in S_0} A_i(x) \cdot [I(N(W(x; p); q)) - A_0(x)] \quad (2-81)$$

其結合 Global shape normalizing transform 的 Inverse Compositional 演算法整體流程如下[22]，為了方便區分前處理以及迭代的步驟，前處理 4 個步驟以(I~IV)表示，而迭代 5 個步驟以((1)~(5))表示：

前處理：

- I. 根據樣板影像  $A_0(x)$  計算出梯度影像  $\nabla A_0$
- II. 求出  $(x; 0)$  的 Jacobian  $\frac{\partial W}{\partial p}$  及  $\frac{\partial N}{\partial q}$
- III. 由(2-78)與(2-79)計算出改變的 steepest descent images  $SD_j(x)$  和  $SD_{j+4}(x)$
- IV. 由(2-80)計算出 Hessian matrix

迭代：

- (1). 根據  $N(x; q)$  與  $W(x; p)$  形變影像  $I$ ，計算出形變影像  $I(N(W(x; p); q))$
- (2). 計算出誤差影像(Error Image)  $I(N(W(x; p); q)) - A_0(x)$
- (3). 計算  $\sum_x SD(x)^T [I(N(W(x; p); q)) - A_0(x)]$
- (4). 由 inverse Hessian matrix 計算出  $\Delta p$  與  $\Delta q$
- (5). 更新參數  $(N \circ W)(x; p, q) \leftarrow (N \circ W)(x; p, q) \circ (N \circ W)(x; \Delta p, \Delta q)^{-1}$

迭代結束：

由(2-81)計算出紋理變化參數  $\lambda_i$

#### 2.5.4. 梯度影像修正

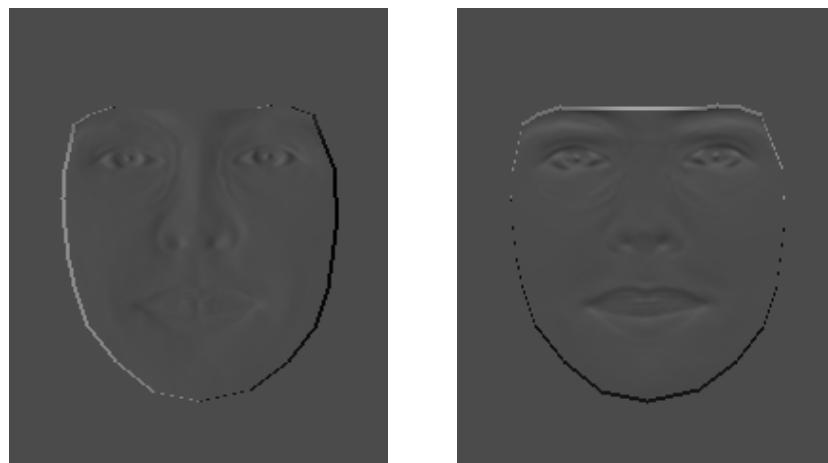
在上一節我們已經介紹了結合 Global shape normalizing transform 的 Inverse Compositional 演算法，在這個演算法的過程中，一開始需要計算出樣版影像  $A_0(x)$  的梯度影像  $\nabla A_0$ ，梯度影像一共包括了垂直方向的梯度(gradient X)與水平方向的梯度(gradient Y)兩種，在計算這個梯度影像時，由於我們所需要的部分只有平均人臉形狀(mean shape)所包含的影像，而我們實際在計算影像梯度時通常是將人臉樣版影像整張下去算出垂直方向的梯度與水平方向的梯度，然後再取出人臉平均人臉形狀所包含的影像部分為計算出的梯度值。

如此一來，因為我們計算整張人臉樣版影像如圖 2-21，而圖 2-21 的邊緣黑色部分不是我們所需要人臉部分，因此計算出來的梯度值在人臉形狀邊緣會是錯誤的如圖 2-22，而造成某些影像在影像校正中人臉形狀邊緣的特徵點發生錯誤的現象。

因此，在這裡我們將計算出來的梯度影像 gradient X 與 gradient Y 經過調整，對於 gradient X 與 gradient Y，我們將形狀邊緣錯誤的梯度值以其形狀往內一個像素的梯度值結果所取代之如圖 2-23，如此一來，在影像校正中的特徵點計算結果即會有所改善。



圖 2-21、用於計算梯度的樣板影像



(a) Gradient X

(b) Gradient Y

圖 2-22、原本的水平與垂直梯度影像



(a) Gradient X

(b) Gradient Y

圖 2-23、修改後的水平與垂直梯度影像

圖 2-24 顯示了進行梯度影像修正之前後的結果，圖 2-24(a)的部分是梯度影像修正之前所進行影像校正的結果，而圖 2-24(b)的部分則是梯度影像修正之後的結果所進行影像校正的結果；可以看到其影像校正的結果，在人臉形狀的邊緣，進行梯度影像修正之前較易發生特徵點校正錯誤的情形，而在進行梯度影像修正之後，特徵點校正錯誤的情形會減少。

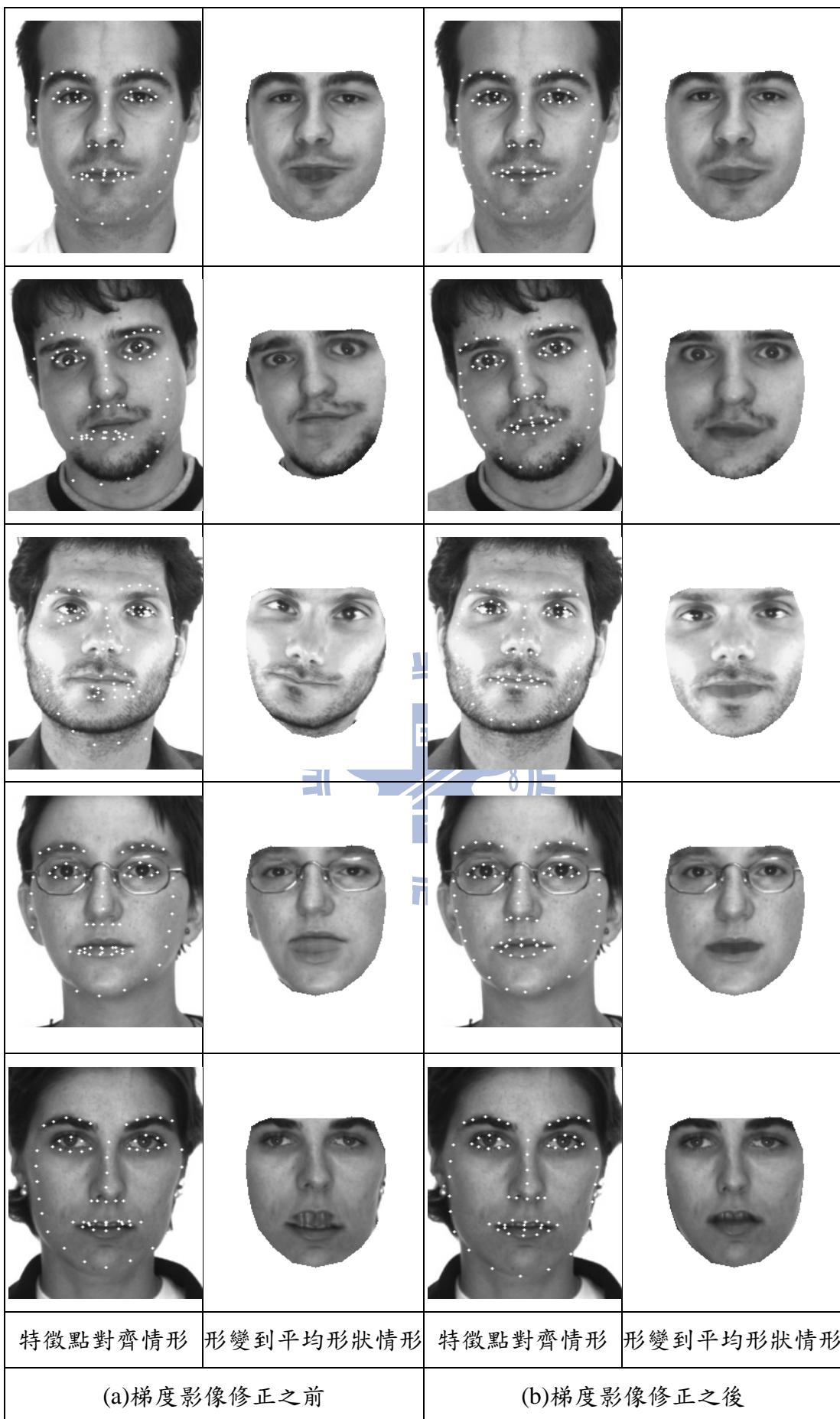


圖 2-24、梯度影像修正之前後影像校正結果

### 2.5.5. 直方圖等化

在訓練人臉模型的時候，我們是先將人臉影像形變至平均形狀中，並進行亮度調整，然後再將調整過後的人臉影像經過 PCA 來計算出其紋理參數模型；我們在影像校正演算法中迭代的時候，會計算出形變到平均形狀的人臉影像  $I(N(W(x; p); q))$ ；而我們在每一次迭代的過程中，我們將形變到平均形狀的目前迭代結果影像  $I(N(W(x; p); q))$  進行並進行亮度調整，將會較對於直接輸入影像進行亮度調整來的符合實際的情形，在人臉影像校正結果方面會有較佳的結果，如圖 2-25，圖中(a)為利用輸入人臉影像先經由亮度調整到平均灰階 128 的情形；而(b)則為在每次迭代的過程中將形變影像進行亮度調整到平均灰階 128 的情形。

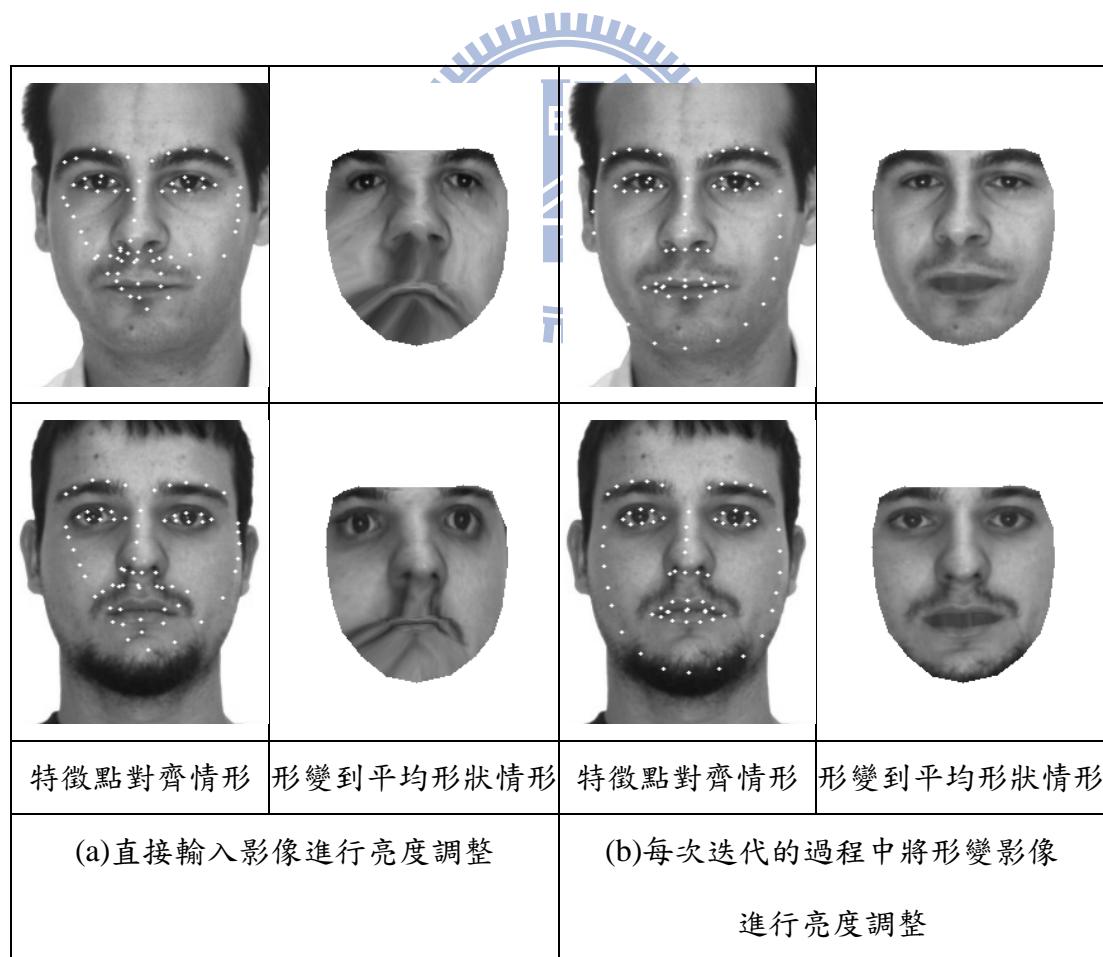


圖 2-25、在每次迭代的過程中將形變影像進行亮度調整與直接對輸入影像進行亮度調整的結果

另外，我們在訓練人臉模型與迭代過程中的亮度調整，將其使用直方圖等化(Histogram Equalization)的方式來調整影像，其優點就是較能凸顯出人臉的特徵部位。在直方圖等化的計算上，我們會先將輸入的影像計算出他的直方圖，圖2-26為訓練影像中的一個人臉紋理影像，圖2-26(b)為其人臉形狀區域中的灰階直方圖，接著由直方圖除以人臉形狀範圍中所有像素的數量，計算出機率密度函數(Probability Density Function, PDF)如圖2-26(c)，並計算出累積分佈函數(Cumulative Distribution Function, CDF)如圖2-26(d)，接著將累積分佈函數由0-1放大到0-255的範圍，並將原始灰階對映到新的灰階值來取代，以得到直方圖等化的影像，圖2-27即為直方圖等化後的結果。

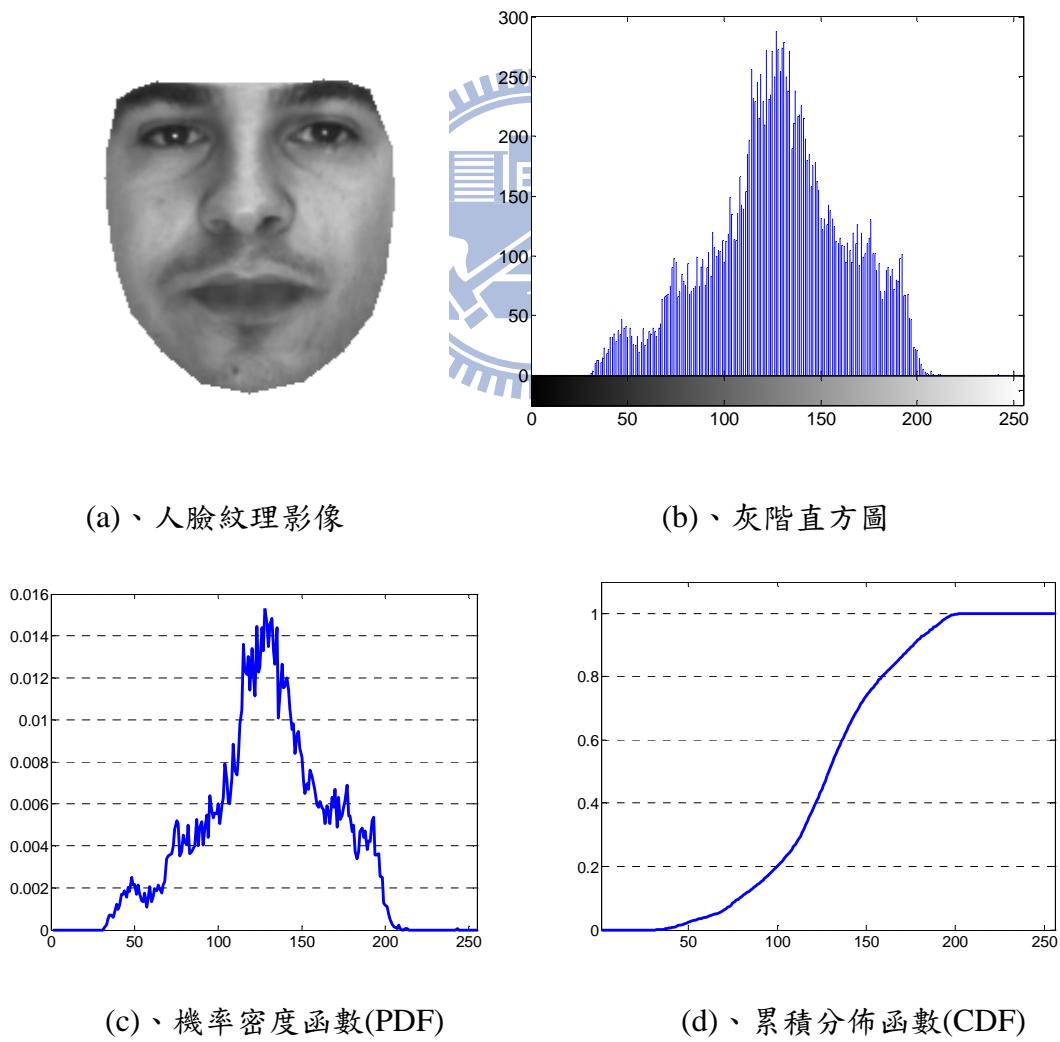
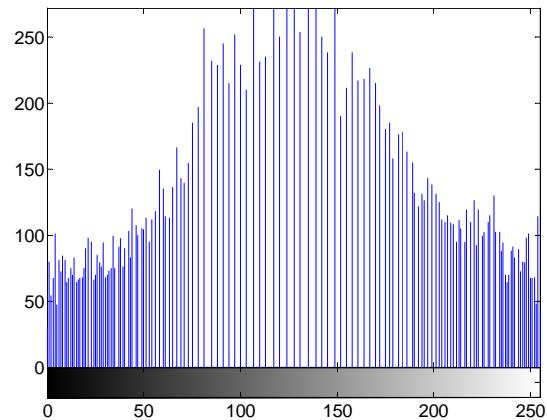


圖 2-26、直方圖等化的計算



(a)、人臉紋理影像



(b)、灰階直方圖

圖 2-27、直方圖等化過後的結果

圖 2-28(a)是將平均紋理經由亮度調整所得到的影像，而圖 2-28(b)是將平均紋理經由直方圖等化所得到的影像，可以得到人臉特徵更清楚的紋理影像，在圖 2-29(a)顯示了直接對輸入影像進行亮度調整以及圖 2-29(b)每次迭代的過程中將形變影像進行直方圖等化調整，可以看出在使用迭代的過程中將形變影像進行直方圖等化調整可以使得人臉影像校正結果方面會有較佳的結果。



(a) 原始亮度調整到平均 128



(b) 使用直方圖等化

圖 2-28、使用不同方法所得到的人臉平均紋理



圖 2-29、在每次迭代的過程中將形變影像進行直方圖等化調整與直接對輸入影像  
進行亮度調整的結果

## 2.5.6. 整體方法

在整個影像校正演算法方面，我們結合了 Global shape normalizing transform 的 Inverse Compositional 演算法和梯度影像修正以及直方圖等化，整體的方法步驟如下：

前處理：

I. 根據樣板影像  $A_0(x)$  計算出梯度影像  $\nabla A_0$

II. 將梯度影像  $\nabla A_0$  進行梯度影像修正

III. 求出  $(x; 0)$  的 Jacobian  $\frac{\partial W}{\partial p}$  及  $\frac{\partial N}{\partial q}$

IV. 由(2-78)與(2-79)計算出改變的 steepest descent images  $SD_j(x)$  和

$SD_{j+4}(x)$

V. 由(2-80)計算出 Hessian matrix

迭代：



(1). 根據  $N(x; q)$  與  $W(x; p)$  形變影像  $I$ ，計算出形變影像  $I(N(W(x; p); q))$

(2). 將形變影像  $I(N(W(x; p); q))$  進行直方圖等化

(3). 計算出誤差影像(Error Image)  $I(N(W(x; p); q)) - A_0(x)$

(4). 計算  $\sum_x SD(x)^T [I(N(W(x; p); q)) - A_0(x)]$

(5). 由 inverse Hessian matrix 計算出  $\Delta p$  與  $\Delta q$

(6). 更新參數  $(N \circ W)(x; p, q) \leftarrow (N \circ W)(x; p, q) \circ (N \circ W)(x; \Delta p, \Delta q)^{-1}$

迭代結束：

由(2-81)計算出紋理變化參數  $\lambda_i$

由此我們可以計算出紋理變化參數 $\lambda_i$ ，由於不同的紋理變化參數可以模擬出各種的人臉，因此我麼選擇其做為我們所要的人臉特徵，之後在第三章會更加詳細的說明。

## 2.6. 結論與討論

本系統利用 Haar-like 人臉偵測來找出影像畫面的人臉，接者使用膚色的密度與人臉影像的大小來過濾畫面中不是人臉的部分來找出確切的人臉。

在人臉的特徵擷取中，我們利用主動外觀模型(AAM)來找出具人臉的特徵，首先在訓練資料方面，我們需要先對訓練的資料進行特徵點的標記，接著找出其人臉平均形狀以及利用 PCA 方法來找出特徵向量的人臉形狀變化量；接著，在紋理的部分是將訓練人臉形變到平均形狀中，接著同樣的找出其人臉平均紋理以及利用 PCA 方法來找出特徵向量的人臉紋理變化量，用這些方法來得到 AAM 的人臉形狀以及紋理模型。

在影像校正方面，Lucas-Kanade 影像校正演算法可用於 AAM 的人臉校正，但由於其在每次迭代的過程中需要重複計算許多數值，因而造成其運算量較大的缺點；因此使用改良的 Inverse Compositional 演算法可以達到相同的效果，而且計算量也可以減少許多。

在 Inverse Compositional 演算法中，對人臉偏轉的角度容忍度十分有限，因此加入 Global shape normalizing transform 更可讓人臉偏轉的角度容忍度變大；而加入梯度影像修正可以改善人臉校正的結果而直方圖等化可以加強人臉的特徵而提升人臉校正的結果，對於接下來的辨識更有幫助。

## 第三章、人臉辨識與表情辨識演算法

在人臉以及表情辨識的方法中，由於每個人的表情變化元素不完全相同，而且我們希望機器人可以同時辨識出使用者及其的表情為何，以增加機器人與使用者的互動。在這裡，我們要辨識出使用者以及他們的表情，我們採用了倒傳遞類神經網路來做為辨識器，而在類神經網路當中，我們採用了人臉的紋理特徵來做為我們的輸入，為了要提高表情辨識的準確性，我們提出了一套個人表情辨識的策略來加強辨識的結果。

### 3.1. 人臉特徵方法

為了要辨識出使用者以及他們的表情，我們採用了人臉的特徵來當作我們辨識器的輸入，在人臉的特徵取得，我們是使用主動外觀模型(AAM)裡的紋理模型參數來做為我們做辨識的人臉特徵，因為紋理模型裡包含了許多的人臉紋理的變化向量，可以來模擬出輸入的各種人臉影像，並且使用紋理模型的參數相對於使用影像可以減低不少數據量。

在取得紋理模型的參數時，由第二章我們知道紋理模型參數為：

$$\lambda_i = \sum_{x \in S_0} A_i(x) \cdot [I(N(W(x; p); q)) - A_0(x)] \quad (3-1)$$

其中  $A_i(x)$  為人臉紋理變化特徵向量， $A_0(x)$  為樣板影像， $I(N(W(x; p); q))$  為形變到人臉平均形狀的人臉影像  $I(N(W(x; p); q)) - A_0(x)$  即為誤差影像(Error Image)， $\lambda_i$  為紋理模型的參數， $i=1,2,\dots,m$ ，我們將  $\lambda_i$  經過 Normalize 之後得到  $\lambda'_i$  做為人臉特徵，當作辨識的倒傳遞類神經網路的輸入特徵參數。

$$\lambda'_i = \frac{\lambda_i}{\sum_{i=1}^m \lambda_i} \quad i = 1, 2, \dots, m \quad (3-2)$$

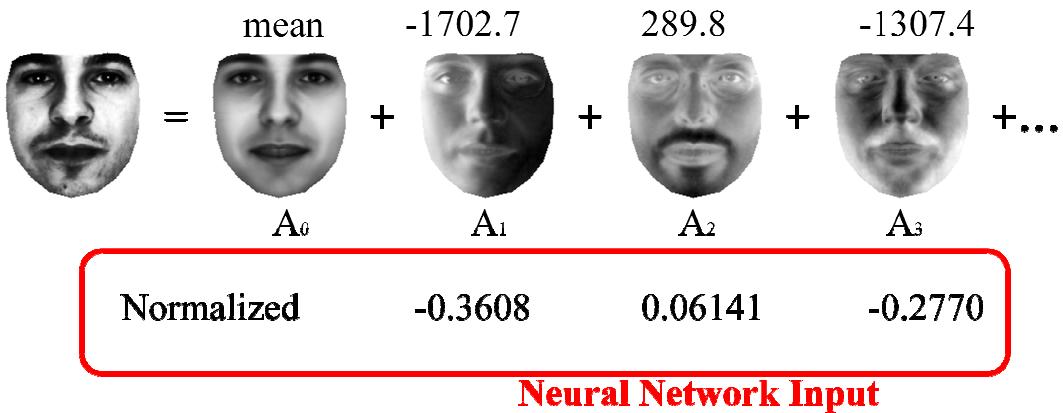


圖 3-1、人臉影像參數最為人臉特徵

### 3.2. 倒傳遞類神經網路

在這裡，我們使用了倒傳遞類神經網路(Back Propagation Neural Network, BPNN)為辨識系統，其主要目的是對於輸出入的映射問題化為非線性最佳化的問題，倒傳遞網路具有偏權值，隱藏層具有雙彎曲的轉移函數，輸出層具有線性的轉移函數[28]，我們可以利用迭代的方式，來優化每一個權重值，來找出輸出入的關係。

類神經網路的主要架構如圖 3-2 所示，其共分為輸入層、隱藏層以及輸出層，其層與層之間有權重值與偏權值兩個參數，我們即是藉由調整這兩個參數，來訓練出類神經網路的輸出入關係。

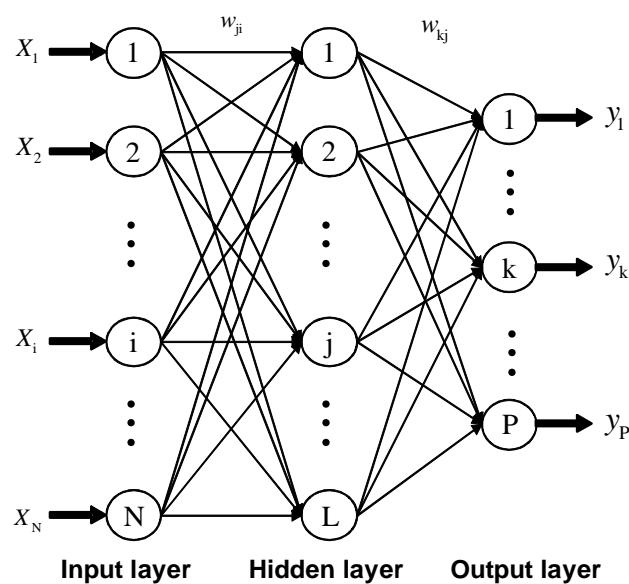


圖 3-2、類神經網路主要架構

在本論文中，我們是採用[28]所描述的倒傳遞類神經網路方法，他將網路模型表示為圖 3-3，其中  $W$  是權重值， $b$  為偏權值，輸入層大小為  $I$ ，隱藏層的大小為  $H$ ，輸出層大小為  $O$ ，這裡在訓練時，利用了倒傳遞的技巧，將靈敏度由最後輸出層計算回輸入層，再對於每一層的權重值與偏權值做修改，其輸出層的靈敏度為：

$$s^M = -2\dot{F}^M(n^M)(t - a) \quad (3-3)$$

其中  $s^M$  為輸出層的靈敏度、 $\dot{F}^M(\cdot)$  為對輸出層轉移函數的微分， $n^M$  為經由權重值與偏權值所計算出的輸出層轉移函數的輸入， $t$  為我們想要的輸出結果， $a$  為由類神經網路實際跑出的結果，而接下來希望藉由輸出層算出的靈敏度往前傳，計算出隱藏層中的靈敏度。

$$s^M \rightarrow s^{M-1} \rightarrow \dots \rightarrow s^2 \rightarrow s^1 \quad (3-4)$$

而中間第  $m$  層的靈敏度  $s^m$  為：

$$s^m = \dot{F}^m(n^m)(W^{m+1})^T s^{m+1} \quad (3-5)$$

其中  $s^m$  為第  $m$  層的靈敏度、 $\dot{F}^m(\cdot)$  為對第  $m$  層轉移函數的微分， $n^m$  為經由權重值與偏權值所計算出的第  $m$  層轉移函數的輸入， $W^{m+1}$  為第  $m+1$  層的權重值， $s^{m+1}$  為第  $m+1$  層的靈敏度。

接著，我們就可以利用靈敏度來更新所有的權重值與偏權值，其中權重值的更新為：

$$W^m(k+1) = W^m(k) - \alpha s^m(a^{m-1})^T \quad (3-6)$$

偏權值的更新為：

$$b^m(k+1) = b^m(k) - \alpha s^m \quad (3-7)$$

其中  $W^m(k)$  與  $b^m(k)$  分別為第  $k$  次迭代所計算出的權重值與偏權值， $\alpha$  為訓練速率，而  $a^{m-1}$  為第  $m-1$  層的輸出結果。

使用多次迭代以後，即可以優化所有的權重值與偏權值，進而訓練出網路的輸出入關係。

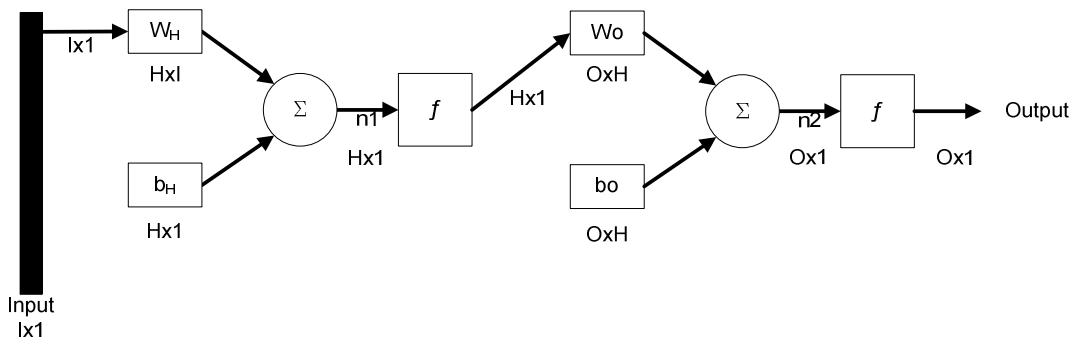


圖 3-3、[28] 中所使用的類神經網路模型

### 3.3. 表情辨識策略

在表情辨識的方法中，由於每個人的中性表情以及表情變化元素皆不完全相同，例如有些人的中性表情看起來就是嘴角往上類似於開心的樣子，而有些人的中性表情看起來就是嘴角往下類似於傷心的樣子，而造成在使用一般表情辨識方法中使用大多數人表情特徵來辨識，可能造成被錯誤辨識的結果，因此，在機器人與人互動的應用中，我們希望機器人對於一個它所認識的人，能夠了解到使用者的確切表情變化，進而提升表情辨識的辨識率。

以下將描述我們的個人表情辨識的策略，首先在表情資料庫中，我們建立了每一位成員自己的個人表情資料庫，以及一個利用所有成員的表情資料所建立的整體表情資料庫；當一位使用者出現在系統中時，我們會先以人臉辨識來辨識出這位出現在系統當中的使用者是成員中的哪位還是非成員，接著再利用所們所建立出不同的表情資料庫來辨識出使用者的表情出來，以下分為兩種情形來進行說明，首先第一種情況是當人臉辨識器辨識出來的結果為系統中的成員，而在第二種情形當中就是當人臉辨識器辨識出來的結果為非成員，以下將對這兩種情形分開來進行說明。

首先在第一種情形當中，我們一開始先對輸入人臉進行人臉辨識，當我們的人臉辨識器辨識出來的輸入人臉屬於系統當中的某位成員時，我們會在表情資料庫中找出這位使用者所建立出來的表情資料，並使用這位使用者的表情資料來進

行表情辨識，來辨識出這位使用者的表情。

接著在第二種情形當中，我們一開始先對輸入人臉進行人臉辨識，當我們的人臉辨識器辨識出來的輸入人臉不屬於系統當中的某位成員也就是非成員(Nonmember)時，我們會在表情資料庫中找出利用所有成員的表情資料所建立的整體表情資料來對這位使用者進行表情辨識，來辨識出這位使用者的表情。

以下為個人表情辨識的範例，假設我們的系統當中擁有 A、B、C、...N 人的人臉資料，我們利用每一位成員的表情資料建立出 A、B、C、...N 的表情資料庫，並利用所有成員的表情資料所建立的整體(A+B+C+...+N)表情資料；首先在第一種情形當中，當使用者進入系統時我們先以人臉辨識來辨識出使用者為哪一個成員，圖 3-4 中，使用者是資料庫中的成員 A，一開始經由人臉辨識辨識出是資料庫中的成員 A 之後，接著我們在表情資料庫中找出 A 的表情資料來進行辨識，以辨識出 A 的表情出來。

接著在第二種情形當中，當使用者進入系統時我們先以人臉辨識來辨識出使用者並不屬於系統當中為非成員，圖 3-5 中，使用者是不在資料庫中的  $\Omega$ ，一開始經由人臉辨識辨識出是不屬於資料庫中的非成員之後，接著我們在表情資料庫中找出 A+B+C+...+N 的整體表情資料來進行辨識，以辨識出這位使用者的表情出來。

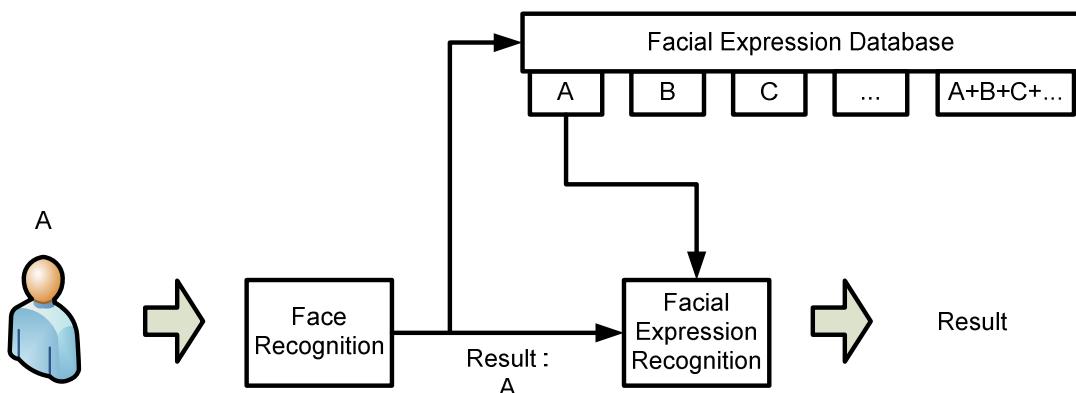


圖 3-4、成員的人臉表情辨識範例

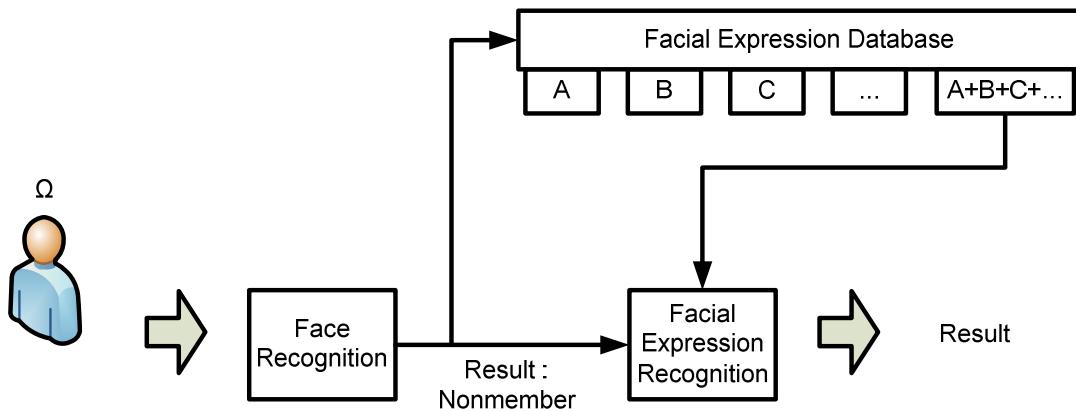


圖 3-5、非成員的人臉表情辨識範例

### 3.4. 人臉辨識與表情辨識訓練流程

在整體人臉與表情辨識訓練的過程上，首先我們可以將輸入人臉影像進行手動的特徵點標點得到人臉所有的人臉形狀；另外，如果 AAM 資料庫已經建立完成，將可以利用 AAM 的影像校正演算法(見第二章)找出輸入人臉的形狀，接著，我們將所有找到人臉形狀的人臉影像形變到平均形狀後，就可利用 AAM 的紋理模型找出人臉紋理變化向量的參數，並計算出人臉的特徵值，接著我們將人臉特徵值放入人臉辨識的類神經網路中計算出人臉辨識的類神經網路資料庫，再來針對所有的成員進行個別的表情辨識類神經網路以及全部成員表情辨識類神經網路進行訓練，最後就可以得到所有類神經網路的資料庫，如圖 3-6 所示。

### 3.5. 結論與討論

本章討論有關人臉辨識與表情辨識的方法，在人臉特徵的選取方面，是以 AAM 人臉模型中的紋理模型，計算出輸入人臉所得到的紋理模型參數，將此做為人臉的特徵值做為辨識，其優點是可以減低人臉特徵的資料量。

在辨識器的部分，這裡使用了倒傳遞類神經網路，在這個方法之中，使用到了一個學習參數  $\alpha$ ，如果學習速率設定太小，則收斂過程會越緩慢，所需要達到

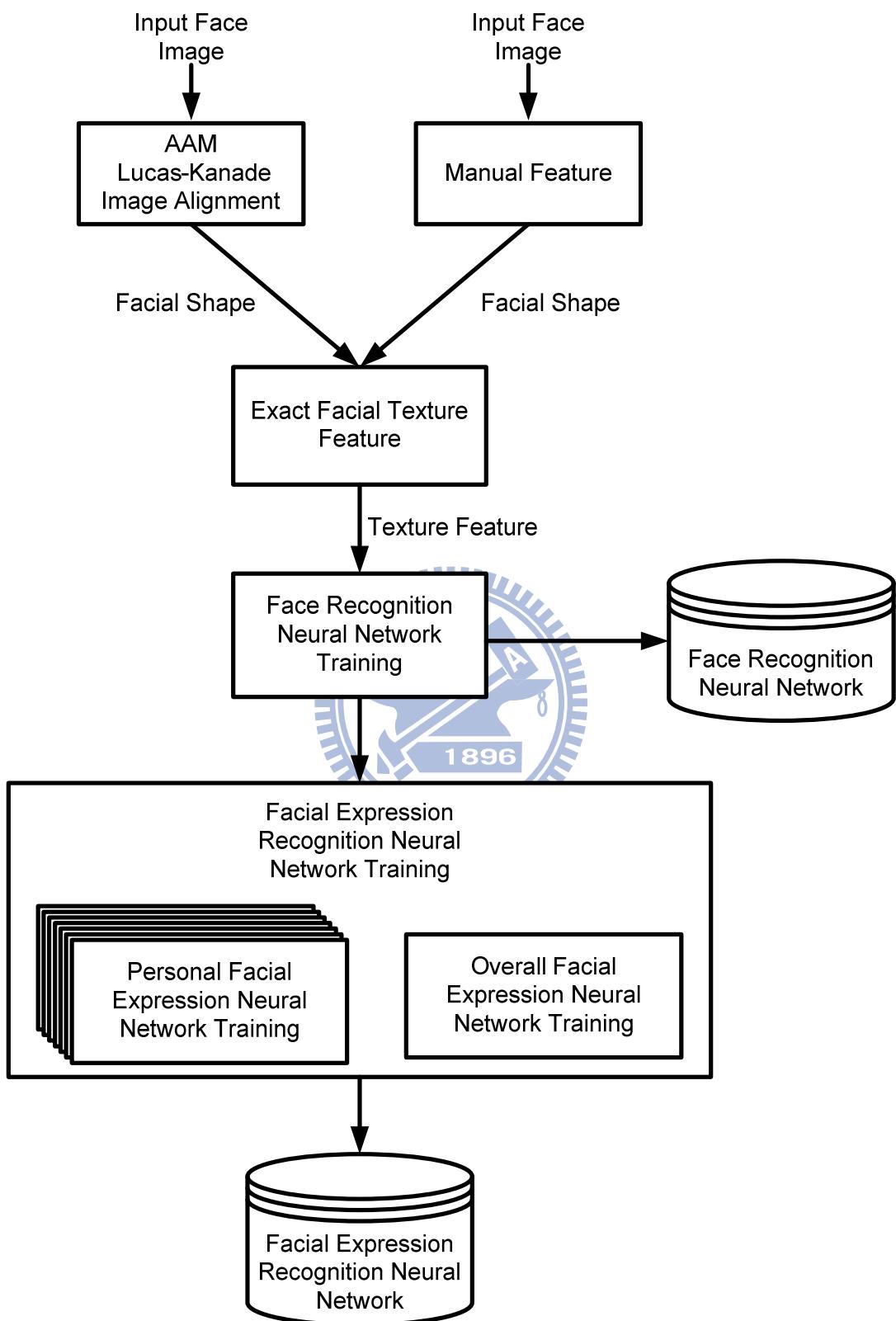


圖 3-6、人臉辨識與表情辨識的整體訓練流程

收斂的迭代次數會相對地提高；但如果學習速率太大，雖然可以加速搜尋效率，但容易造成搜尋過程中，目標函數產生不穩定的震盪情形，故需要多次的嘗試，才可找出適合此訓練資料的學習速率。

在表情辨識的方法中，我們藉由先進行人臉辨識，辨識出使用者的身分以後，再針對他的個人表情資料庫作辨識，如此一來，就可以減少需多在不同人之間的表情元素的差異所造成的辨識錯誤情形發生。



## 第四章、實驗結果

為了驗證本論文的人臉影像特徵擷取，以及成員人臉辨識與表情辨識系統，我們將藉由以下幾個步驟來驗證：

1. 建立影像資料庫
2. 利用 AAM 以及 Lucas-Kanade 影像校正演算法，對於輸入影像進行校正的結果，這裡將分別使用資料庫以及實驗室建立的影像進行影像校正。
3. 使用人臉辨識方法，將 AAM 以及 Lucas-Kanade 影像校正出來的結果進行辨識，分別使用資料庫以及實驗室建立的影像資料庫進行辨識。
4. 使用表情辨識方法，將 AAM 以及 Lucas-Kanade 影像校正出來的結果進行辨識，分別使用資料庫以及實驗室建立的影像資料庫進行辨識，在這裡將會對於使用相同的人臉特徵於一般情形下使用整體表情辨識與本論文中的個人表情辨識的結果進行比較。



### 4.1. 影像校正演算法迭代結果

在這個部分，我們將使用資料庫以及自行擷取的影像進行影像校正演算法的迭代，一開始先對於資料庫的結果，再來是由實驗室擷取的影像資料做影像校正的結果，在資料庫的部分，我們會先經由人臉偵測找出人臉的部分。

#### 4.1.1. 使用資料庫影像

在資料庫方面，首先我們使用 AR face database[23]，在這個測試裡，我們共採用了 12 位不同的資料人員，每個人的影像中包括了 14 張影像，14 張影像中，有不同表情的變化，包括了中性、高興、生氣、尖叫以及對於中性表情的左方、右方、左加右方打光的情形，每種影像各兩張，圖 4-1 是 AR face database 的幾個影像範例；在測試中，我們使用一半每人 7 張(共 84 張)做訓練，另一半每人 7 張(共 84 張)做測試，接著將形變影像與 AAM 模擬出的影像相減的 RMS 灰階影

像誤差來表示迭代的變化，如圖 4-2，其中縱軸是灰階誤差，橫軸是迭代次數，第一筆是平均形狀位置紋理誤差，一共迭代 21 次，其中可以看出整個平均的灰階誤差有明顯的下降趨勢，而由於訓練資料並無法完全的表示所有人臉的詳細紋理情況，所以收斂的結果還是會有部分的誤差存在。圖 4-3 為使用 Lucas-Kanade 影像校正出來的特徵點結果與將取得的人臉影像形變至平均形狀的結果。

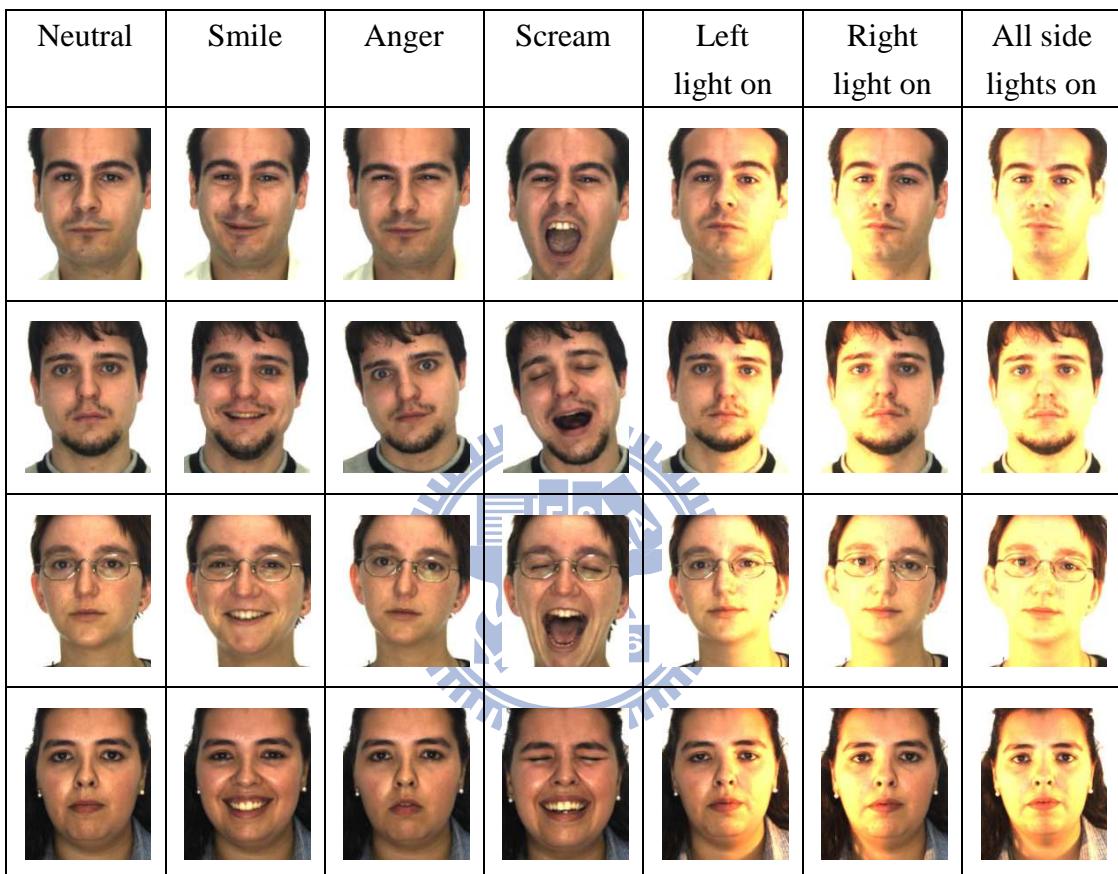


圖 4-1、AR database 的部分影像

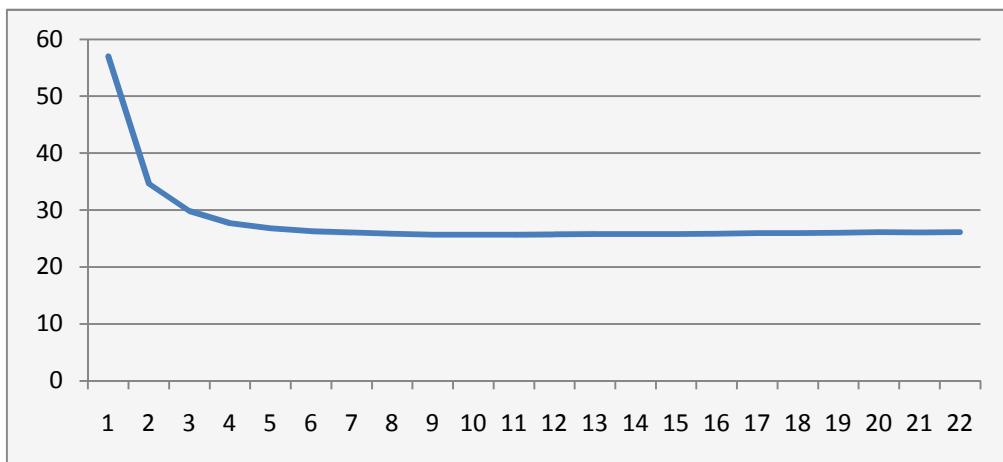


圖 4-2、所有 84 筆測試資料平均灰階 RMS 誤差收斂情形



圖 4-3、使用 AR database 的特徵點結果以及其形變出的人臉

由於 AR database 的人臉表情影像不足以讓我們取得足夠的表情影像來進行個人的表情辨識方法，而且其表情的種類也較缺乏，因此，我們接者採用 BU-3DFE[29]人臉表情資料庫，它包含了六種不同的表情，包括了生氣、厭惡、害怕、高興、傷心和驚訝(Angry, Disgust, Fear, Happy, Sad and Surprise)，我們在這裡利用了 10 位不同的人臉，每人每種表情各 2 張(共 12 張)，總共 240 張人臉影像來做為 AAM 模型的訓練，測試時，我們測試了 20 位不同的人臉，每人各 12 張，總共 480 張作為測試影像，如圖 4-4。

接著使用形變影像與 AAM 模擬出的影像相減的 RMS 灰階影像誤差來表示迭代的變化，如圖 4-5，其中縱軸是灰階誤差，橫軸是迭代次數，第一筆是平均形狀位置紋理誤差，一共迭代 21 次，可以看出整個平均的灰階誤差有明顯的下降趨勢，而由於訓練資料並不是使用全部使用者的人臉影像去做訓練，因此無法完全的表示所有人臉的詳細紋理情況，並且由於 BU-3DFE 人臉表情資料庫裡面的人臉影像包含了不同膚色的人臉變化，所以收斂的結果會有部分的誤差存在。



圖 4-4、其中四位不同資料庫成員的表情影像

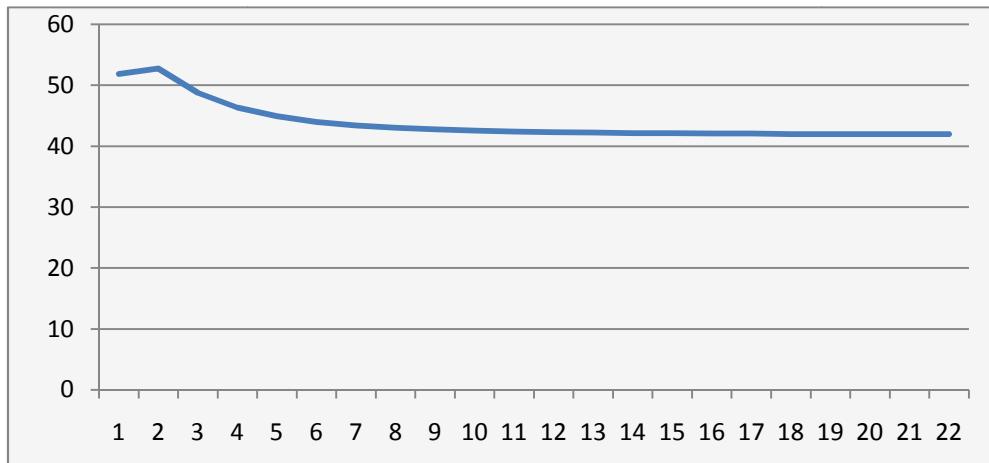


圖 4-5、所有 480 筆測試資料平均灰階 RMS 誤差收斂情形

圖 4-6 為使用 Lucas-Kanade 影像校正出來的特徵點結果與將取得的人臉影像形變至平均形狀的結果，其中在人臉表情太大的情況下，還是會有部分結果不夠理想。

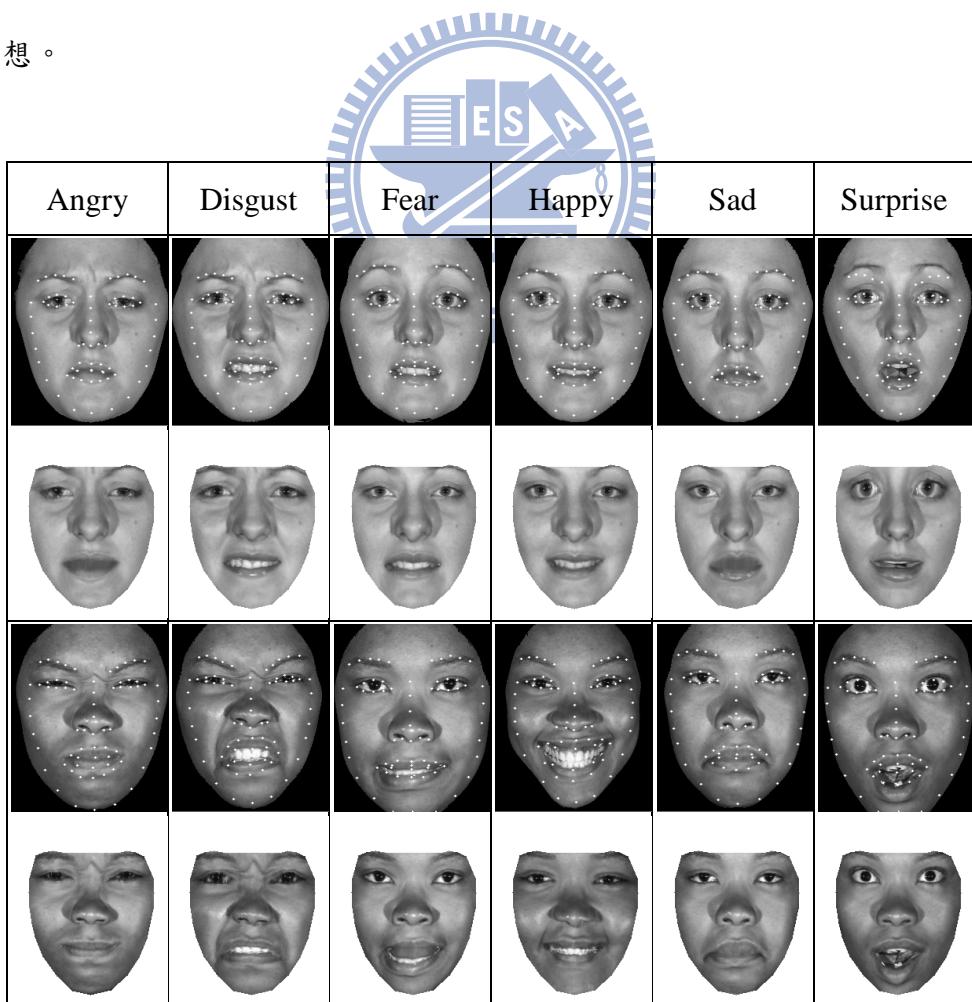




圖 4-6、使用 BU-3DFE 的特徵點結果以及其形變出的人臉

#### 4.1.2. 使用自行擷取影像

為了切合實際上的使用情形，我們對實驗室的成員取出影像來進行實驗，在這裡，我們同樣的取出每人六種表情包括了生氣、厭惡、害怕、高興、傷心和驚訝(Angry, Disgust, Fear, Happy, Sad and Surprise)，共 10 位使用者的資料，測試時，我們測試了 10 位不同的人臉，每人各 60 張，共 600 張，圖 4-7 是其中 5 位使用者的影像範例。

接著使用形變影像與 AAM 模擬出的影像相減的 RMS 灰階影像誤差來表示迭代的變化，如圖 4-8，其中縱軸是灰階誤差，橫軸是迭代次數，第一筆是平均形狀位置紋理誤差，一共迭代 21 次，可以看出整個平均的灰階誤差有明顯的下降趨勢，而由於訓練資料並不是使用全部使用者的人臉影像去做訓練，因此無法完全的表示所有臉的詳細紋理情況，所以收斂的結果會有部分的誤差存在。

圖 4-9 為使用 Lucas-Kanade 影像校正出來的特徵點結果與將取得的人臉影像形變至平均形狀的結果，其中在人臉表情太大的情況下，還是會有部分結果不夠理想。

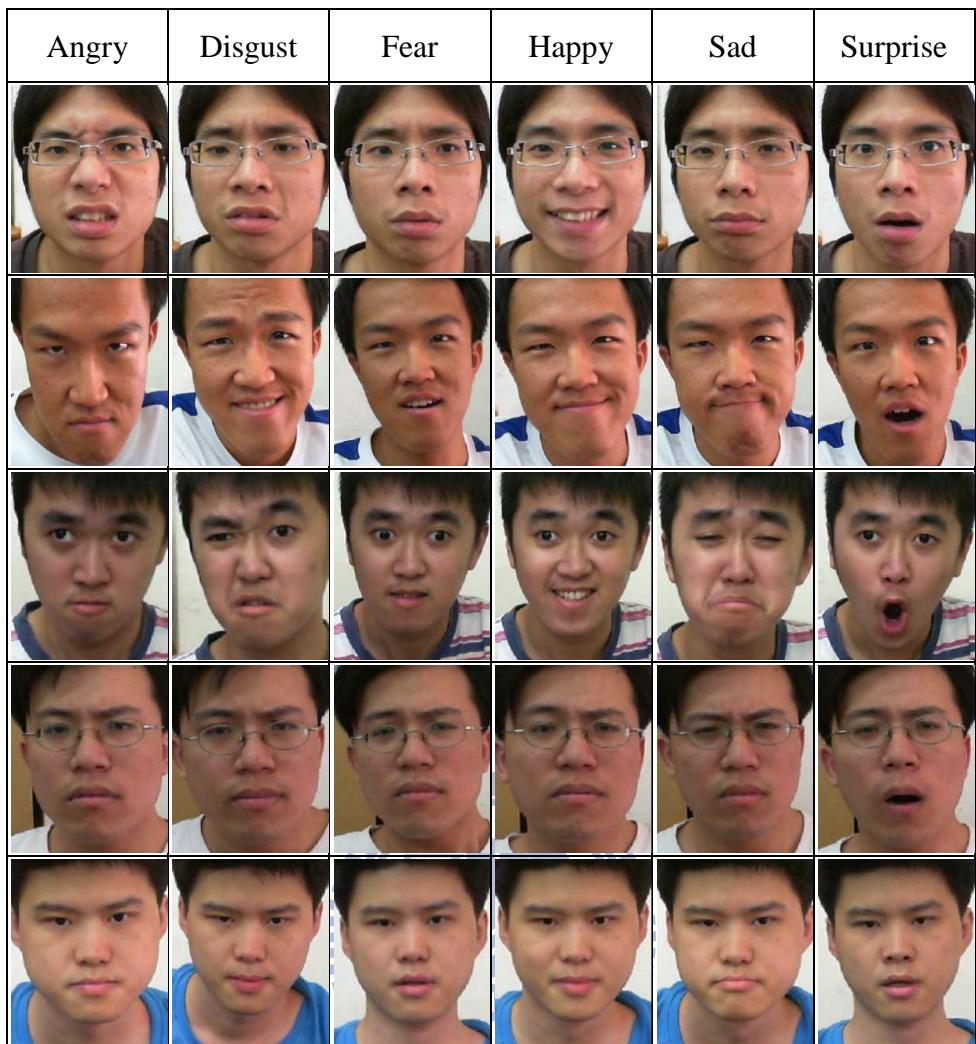


圖 4-7、其中五位實驗室成員的各種表情資料

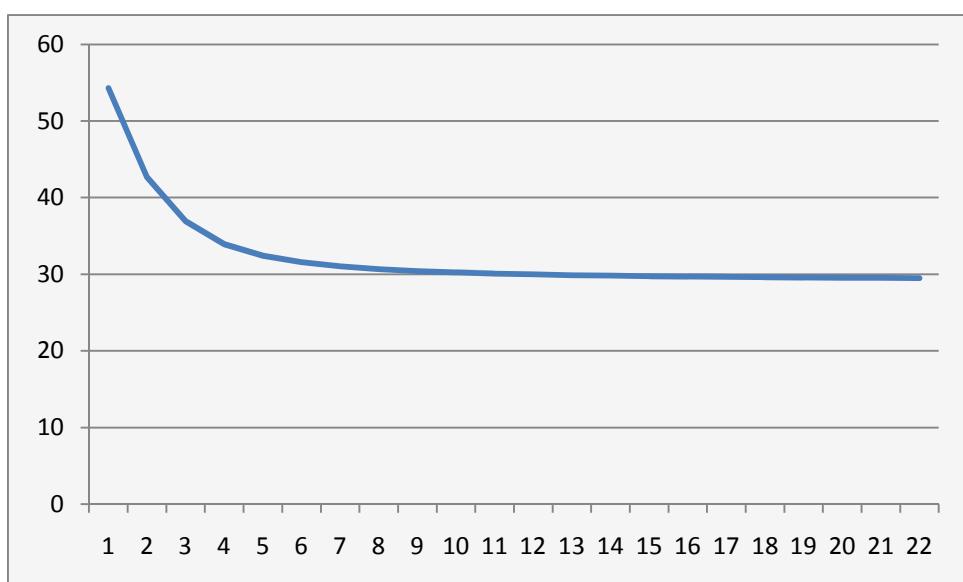


圖 4-8、所有 300 筆測試資料平均灰階 RMS 誤差收斂情形

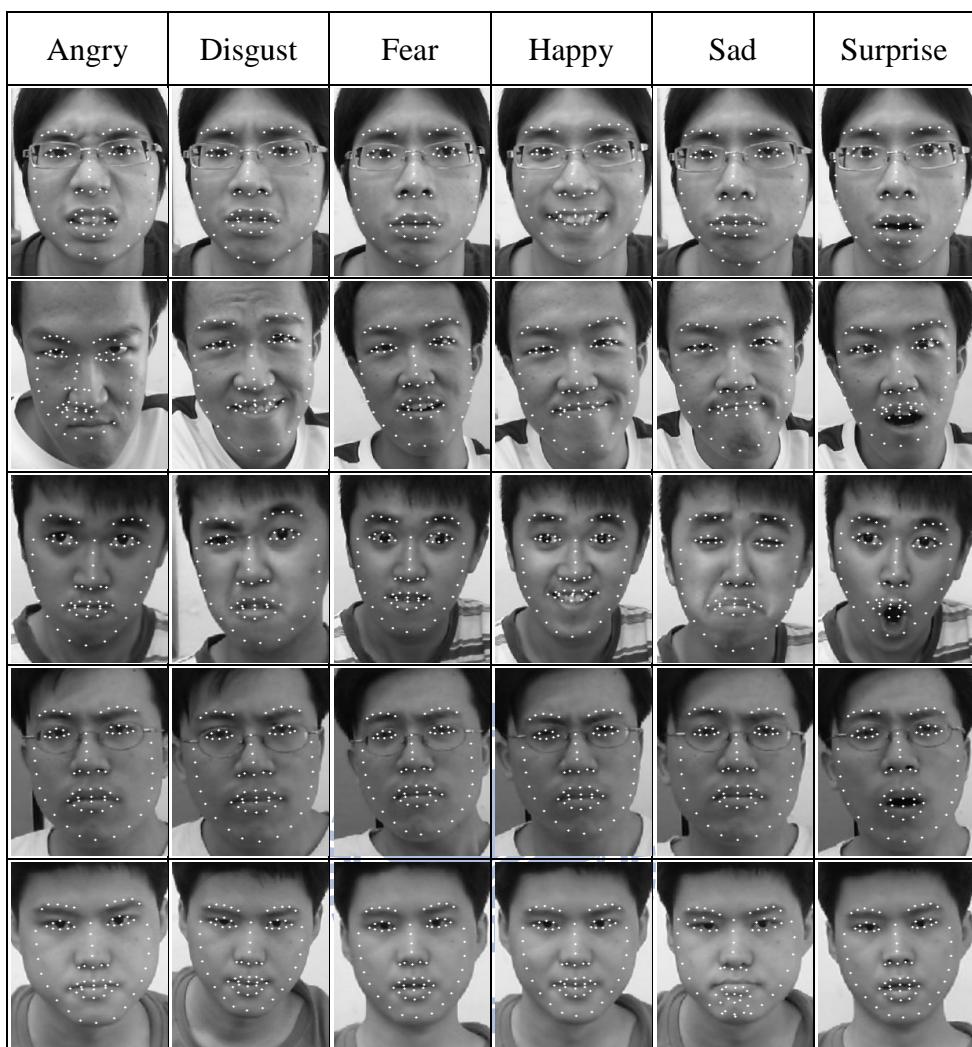


圖 4-9、其中五位實驗室成員的各種表情特徵點結果

## 4.2. 人臉辨識結果

在這個部分我們要驗證本論文的人臉辨識方法，我們同樣的分為兩部分的實驗，第一部分是使用資料庫進行辨識，而第二部分是使用自行擷取的影像進行辨識，一開始先對於資料庫的結果做驗證，再來是採用實驗室擷取的影像資料人臉辨識的結果做驗證。

### 4.2.1. 使用資料庫影像

在人臉辨識的部分，首先我們使用 BU-3DFE 的人臉資料庫進行辨識，輸入

的資料共有 20 個人，每人有六種表情，每種表情各兩張做為訓練資料，另兩張做為測試資料，每人訓練資料共有 12 張，總共訓練資料為 240 張，每人測試資料共有 12 張，總共測試資料為 240 張，首先經由 Lucas-Kanade 影像校正，接著利用 3.1 所提到的方法找出人臉特徵並使用倒傳遞類神經網路進行辨識。

使用倒傳遞類神經網路進行辨識，這裡一共訓練了 5000 次，圖 4-10 顯示了前 1000 次的收斂結果，可以看出其有正確的收斂到 0。

表 4-1 是 20 位成員的辨識率，總共測試影像為 240 張，共有 236 張正確，4 張辨識錯誤，辨識率最低有 91.7%，最高有 100%，平均辨識率為 98.3%。

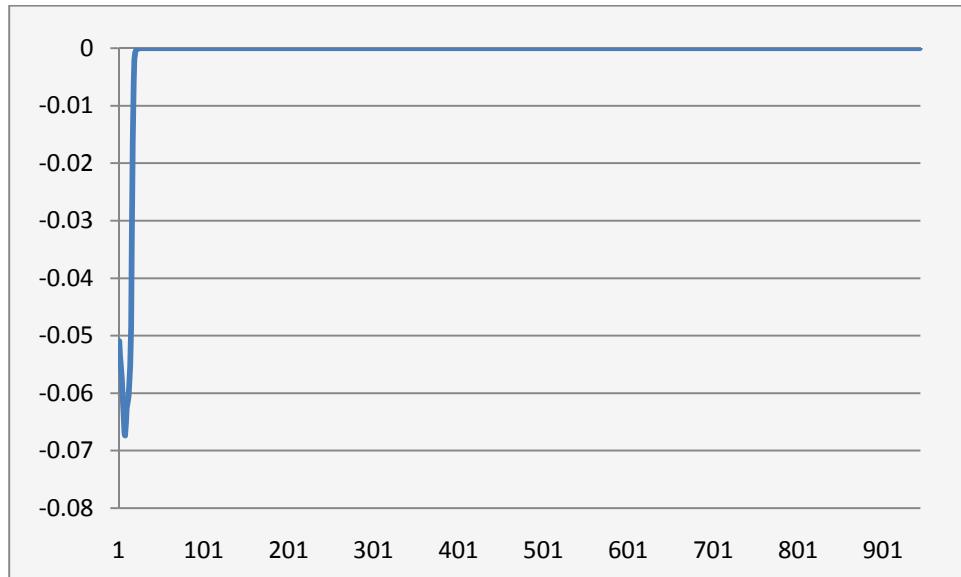


圖 4-10、人臉辨識類神經網路的收斂情形

表 4-1、20 位成員的辨識率

USER	1	2	3	4	5	6	7	8	9	10
Rate	100%	100%	100%	100%	100%	91.7%	100%	100%	100%	91.7%
USER	11	12	13	14	15	16	17	18	19	20
Rate	100%	100%	100%	100%	91.7%	100%	100%	91.7%	100%	100%
平均辨識率										
98.3%										

表 4-2 是人臉的辨識率與各種表情變化的關係，上方代表各種表情的輸入資料，第一列 Correct 代表辨識正確的資料數，第二列 Wrong 代表辨識錯誤的資料數，第三列 Total 代表每種表情的全部資料數，第四列 Rate 代表每種表情的人臉辨識正確率，其中六種表情中最低錯誤為一張的 97.4%，最高為厭惡及害怕的 100%，平均為 98.3%，每一種表情之下的人臉辨識率都差不多為 95% 以上。

表 4-3 為所有人辨識錯誤的結果，共有 4 張，其中最左行是辨識錯誤的使用者代號，而中間是辨識錯誤到哪位使用者，其中 0 表示非訓練資料中的成員，最後一行是這個錯誤的資料是在哪種表情之下出現的；由結果來看各種表情的錯誤比例都差不多，而辨識錯誤的皆是分類到不明使用者的部分。

表 4-2、人臉辨識率與表情的關係

		INPUT						
		Angry	Disgust	Fear	Happy	Sad	Surprise	Total
OUTPUT	Correct	39	40	40	39	39	39	236
	Wrong	1	0	0	1	1	1	4
	Total	40	40	40	40	40	40	240
	Rate	97.4%	100.0%	100.0%	97.4%	97.4%	97.4%	98.3%

表 4-3、人臉辨識錯誤的結果

USER	Wrong Recognition Results	Belong to Facial Expression
6	0	Sad
10	0	Happy
15	0	Angry
18	0	Surprise

為了要分別出不是成員中的使用者，這裡進行了非成員的人臉辨識，表 4-4 是非成員的人臉辨識結果，上方代表非成員的使用者 N1~N8 的輸入資料，每個人各有 24 張輸入測試資料，左方前 20 列代表辨識錯誤為成員(1~20)的資料，NonMember 代表辨識為非成員的正確結果，Rate 代表每位非成員所辨識得到的正確率。

其中 N1~N8 八位非成員中，辨識率最低為 62.5%，最高為 95.8%，平均為 80.7%，而辨識錯誤的大多是辨識為另一個人(例如 N4 有 6 張辨識為成員 13，N5 有 5 張辨識為成員 16，N8 有 6 張辨識為成員 16)。

表 4-4、非成員的人臉辨識結果

		INPUT								<b>TOTAL</b>
		<b>N1</b>	<b>N2</b>	<b>N3</b>	<b>N4</b>	<b>N5</b>	<b>N6</b>	<b>N7</b>	<b>N8</b>	
<b>OUTPUT</b>	<b>1</b>	0	0	0	0	0	0	0	0	0
	<b>2</b>	0	0	0	0	0	0	0	0	0
	<b>3</b>	0	0	0	0	0	0	0	0	0
	<b>4</b>	0	0	0	3	0	0	0	0	3
	<b>5</b>	0	0	2	0	0	0	0	0	2
	<b>6</b>	0	0	1	0	0	0	0	0	1
	<b>7</b>	0	2	0	0	0	0	0	0	2
	<b>8</b>	0	0	0	0	0	0	0	0	0
	<b>9</b>	0	0	0	0	1	1	0	0	2
	<b>10</b>	0	1	0	0	0	0	2	0	3
	<b>11</b>	0	0	0	0	0	0	0	1	1
	<b>12</b>	0	0	0	0	0	0	0	0	0
	<b>13</b>	0	0	0	6	0	0	0	0	6
	<b>14</b>	3	0	0	0	0	0	0	0	3
	<b>15</b>	0	0	0	0	0	0	0	0	0
	<b>16</b>	0	0	0	0	5	0	0	6	11
	<b>17</b>	0	0	0	0	0	0	2	0	2
	<b>18</b>	0	0	0	0	0	0	0	1	1
	<b>19</b>	0	0	0	0	0	0	0	0	0
	<b>20</b>	0	0	0	0	0	0	0	0	0
	<b>NonMember</b>	<b>21</b>	<b>21</b>	<b>21</b>	<b>15</b>	<b>18</b>	<b>23</b>	<b>20</b>	<b>16</b>	<b>155</b>
	<b>Rate</b>	<b>87.5%</b>	<b>87.5%</b>	<b>87.5%</b>	<b>62.5%</b>	<b>75.0%</b>	<b>95.8%</b>	<b>83.3%</b>	<b>66.7%</b>	<b>80.7%</b>

#### 4.2.2. 使用自行擷取影像

接下來，我們要驗證人臉辨識在實際的實驗室的成員取出影像來進行人臉辨識方面，我們使用 10 位使用者的資料來做辨識，每人六種表情包括了生氣、厭惡、害怕、高興、傷心和驚訝(Angry, Disgust, Fear, Happy, Sad and Surprise)，每個人每種表情各 5 張，總共訓練資料每人 30 張，測試資料同樣的也是 30 張，總共測試資料為 300 張，訓練資料為 300 張。方法同樣是經由 Lucas-Kanade 影像校正，接著利用 3.1 所提到的方法找出人臉特徵並使用倒傳遞類神經網路進行辨識。

使用倒傳遞類神經網路進行辨識，這裡一共訓練了 5000 次，圖 4-10 顯示了前 1000 次的收斂結果，可以看出其有持續的收斂到 0。

表 4-5 是 10 位實驗室成員(A,B,C,D,E,F,G,H,I,J)的辨識率，上方(A~J)代表各個成員的輸入資料，左方(A~J)代表輸入資料辨識得到的成員結果，None 表示為沒有辨識出來為資料庫的成員(A~J)的結果，而在下面的 Correct 表示為辨識正確的結果，Wrong 表示辨識錯誤的結果，Total 為每個人的影像張數，在這裡每個人各有 30 張，最後 Rate 是各種表情的辨識率。總共測試影像為 300 張，共有 290 張正確，10 張辨識錯誤，其中最低為 86.7%，最高為 100%，平均辨識率為 96.7%。

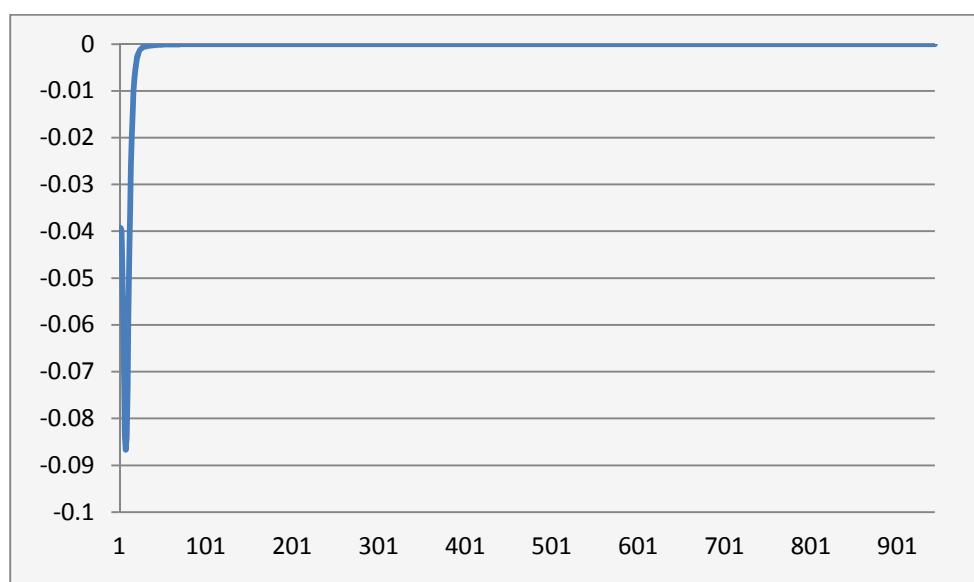


圖 4-11、人臉辨識類神經網路的收斂情形

表 4-5、10 位實驗室成員的辨識率

			INPUT								
		A	B	C	D	E	F	G	H	I	J
OUTPUT	A	30	0	0	0	0	0	0	0	0	0
	B	0	29	0	0	1	0	0	0	0	0
	C	0	0	30	1	0	0	0	0	0	0
	D	0	0	0	27	0	0	0	0	0	0
	E	0	1	0	0	26	0	0	0	0	0
	F	0	0	0	0	0	30	0	0	0	0
	G	0	0	0	0	0	0	30	0	0	0
	H	0	0	0	0	0	0	0	30	0	0
	I	0	0	0	0	0	0	0	0	29	0
	J	0	0	0	0	0	0	0	0	0	29
	None	0	0	0	2	3	0	0	0	1	1
	Correct	30	29	30	27	26	30	30	30	29	29
	Wrong	0	1	0	3	4	0	0	0	1	1
	Total	30	30	30	30	30	30	30	30	30	30
	Rate	100%	96.7%	100%	90.0%	86.7%	100%	100%	100%	96.7%	96.7%

表 4-6 是人臉的辨識率與各種表情變化的關係，上方代表各種表情的輸入資料，第一列 Correct 代表辨識正確的資料數，第二列 Wrong 代表辨識錯誤的資料數，第三列 Total 代表每種表情的全部資料數，第四列 Rate 代表每種表情的人臉辨識正確率，其中六種表情中最低為錯誤 3 張的 94.0%，最高為生氣及害怕的 100%，平均為 96.7%，每一種表情之下的人臉辨識率都差不多為 90% 以上。

表 4-6、人臉辨識率與表情的關係

		INPUT						
		Angry	Disgust	Fear	Happy	Sad	Surprise	Total
OUTPUT	Correct	50	49	50	47	47	47	290
	Wrong	0	1	0	3	3	3	10
	Total	50	50	50	50	50	50	300
	Rate	100.0%	98.0%	100.0%	94.0%	94.0%	94.0%	96.7%

### 4.3. 表情辨識結果

在這個部分我們要驗證本論文的表情辨識方法，我們同樣的分為兩部分的實驗，第一部分是使用資料庫進行表情的辨識，第二部分自行擷取的影像進行表情的辨識，一開始先對於資料庫的結果，再來是由實驗室擷取的影像資料表情辨識的結果，在結果呈現上，會展示出使用全體的表情辨識方法與已知使用者的情況下使用個人表情辨識方法以及整體人臉辨識及個人表情辨識的結果比較。

#### 4.3.1. 使用資料庫影像

在表情辨識的部分，我們使用 BU-3DFE 的人臉資料庫進行辨識，輸入的資料共有 20 個人，每人有六種表情，每種表情各兩張做為訓練資料，另外兩張做為測試資料，每人訓練資料共有 12 張，總共訓練資料為 240 張，每人測試資料共有 12 張，總共測試資料為 240 張，計算方是同樣是先經由 Lucas-Kanade 影像校正，接著利用 3.1 所提到的方法找出人臉特徵，並使用導傳遞類神經網路進行辨識。

首先一開始先進行全體表情辨識方法，使用倒傳遞類神經網路進行辨識，這裡一共訓練了 5000 次，圖 4-12 顯示了前 1000 次的收斂結果，可以看出其有正確的收斂到 0。

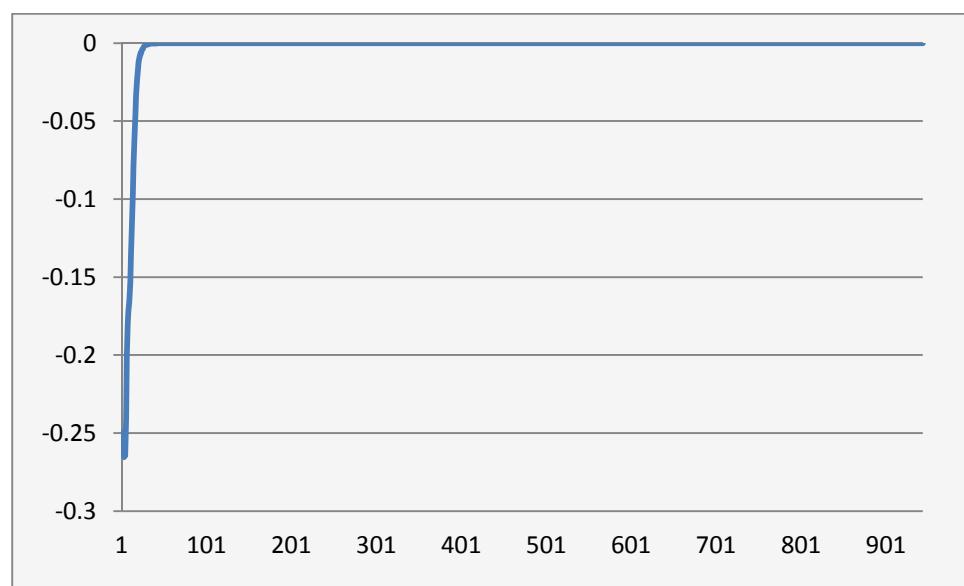


圖 4-12、全體表情辨識類神經網路的收斂情形

接著進行個人表情辨識方法，使用倒傳遞類神經網路進行辨識，這裡一共訓練了 5000 次，圖 4-13 顯示了 20 位成員前 1000 次的收斂結果的平均，可以看出有正確的收斂到 0。

表 4-7 到表 4-9 分別是使用全體表情辨識結果、在已知使用者情況下進行個人表情辨識結果、先進行人臉辨識再使用整體個人表情辨識結果，這裡總共採用了 20 位成員的資料輸入，每個人共有 24 張測試影像，每個人的每種表情各有 2 張輸入影像，共 6 種表情，總共輸入影像為 240 張；在表格中上方為輸入的人臉表情影像，左邊前六列是不同表情輸入所辨識出的輸出結果，Error 表示為沒有辨識出來為資料庫的六種表情的結果，而在下面的 Correct 表示為辨識正確的結果，Wrong 表示辨識錯誤的結果，Total 為表情的影像張數，在這裡每種表情各有 40 張，最後 Rate 是各種表情的辨識率。

表 4-7 是使用全體的表情辨識的辨識率，總共測試影像為 240 張，共有 167 張正確，73 張辨識錯誤，辨識率最低為生氣有 45.0%，最高為快樂有 95.0%，平均辨識率為 63.8%。

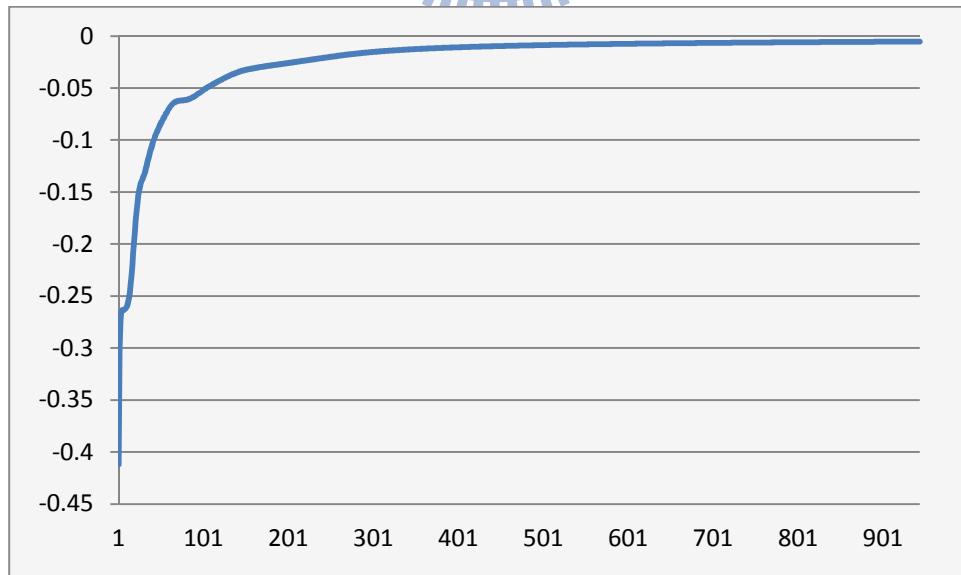


圖 4-13、個人表情辨識類神經網路 20 位使用者的平均收斂情形

表 4-8 是使用個人的表情辨識的辨識率，這裡是為了要驗證使用個人的表情辨識的辨識率可以優於使用全體的表情辨識的辨識率，這裡總共測試影像為 240 張，共有 202 張正確，38 張辨識錯誤，辨識率最低為害怕有 70.0%，最高為快樂有 97.5%，平均辨識率為 84.2%。

最後表 4-9 是使用本論文的個人的表情辨識方法的整體流程結果，這裡先進行了人臉辨識，再由辨識出來的成員進行個人的表情辨識，總共測試影像為 240 張，共有 201 張正確，39 張辨識錯誤，辨識率最低為害怕和生氣有 70.0%，最高為快樂有 97.5%，平均辨識率為 83.8%。

由表 4-7 到表 4-9 的結果可以看出，使用個人的表情辨識方法可以讓表情辨識的辨識率大幅的提升，而在各種表情的辨識方面，害怕和生氣這兩種表情較難有效的辨識出來，而快樂的表情較能有效的被正確的辨識出來。

表 4-7、使用 BU-3DFE 的使用全體的表情辨識結果

		INPUT						Total
		Angry	Disgust	Fear	Happy	Sad	Surprise	
OUTPUT	Angry	18	0	0	0	0	0	
	Disgust	5	30	1	0	0	2	
	Fear	1	0	27	0	2	1	
	Happy	0	0	1	38	0	0	
	Sad	4	0	0	0	23	0	
	Surprise	0	0	0	0	0	31	
	Error	12	10	11	2	15	6	
	Correct	18	30	27	38	23	31	167
	Wrong	22	10	13	2	17	9	73
	Total	40	40	40	40	40	40	240
	Rate	45.0%	75.0%	67.5%	95.0%	57.5%	77.5%	69.6%

表 4-8、使用 BU-3DFE 在已知使用者的情況使用個人的表情辨識結果

		INPUT						
		Angry	Disgust	Fear	Happy	Sad	Surprise	Total
OUTPUT	Angry	29	0	0	0	0	1	
	Disgust	4	37	1	0	1	0	
	Fear	1	1	28	0	0	2	
	Happy	0	1	1	39	0	0	
	Sad	1	0	2	1	34	0	
	Surprise	0	0	0	0	0	35	
	Error	5	1	8	0	5	2	
	Correct	29	37	28	39	34	35	202
	Wrong	11	3	12	1	6	5	38
	Total	40	40	40	40	40	40	240
	Rate	72.5%	92.5%	70.0%	97.5%	85.0%	87.5%	84.2%

表 4-9、使用 BU-3DFE 的整體人臉辨識與個人的表情辨識結果

		INPUT						
		Angry	Disgust	Fear	Happy	Sad	Surprise	Total
OUTPUT	Angry	28	0	0	0	0	1	
	Disgust	4	37	1	0	1	0	
	Fear	2	1	28	0	0	2	
	Happy	0	1	1	39	0	0	
	Sad	1	0	2	1	34	0	
	Surprise	0	0	0	0	0	35	
	Error	5	1	8	0	5	2	
	Correct	28	37	28	39	34	35	201
	Wrong	12	3	12	1	6	5	39
	Total	40	40	40	40	40	40	240
	Rate	70.0%	92.5%	70.0%	97.5%	85.0%	87.5%	83.8%

### 4.3.2. 使用自行擷取影像

接下來我們要驗證在實際實驗室成員的表情辨識結果，我們使用實驗室中 10 位使用者的人臉影像資料來做辨識，每人六種表情包括了生氣、厭惡、害怕、高興、傷心和驚訝(Angry, Disgust, Fear, Happy, Sad and Surprise)，每種表情各 5 張作為訓練資料，5 張做為測試資料，訓練資料每人共 30 張，測試資料同樣的也是 30 張，總共測試資料為 300 張，訓練資料為 300 張。計算方是同樣是先經由 Lucas-Kanade 影像校正，接著利用 3.1 所提到的方法找出人臉特徵，並使用導傳遞類神經網路進行辨識。

首先一開始先進行全體表情辨識方法，使用倒傳遞類神經網路進行辨識，這裡一共訓練了 5000 次，圖 4-14 顯示了前 1000 次的收斂結果，可以看出其有正確的收斂到 0。

接著進行個人表情辨識方法，使用倒傳遞類神經網路進行辨識，這裡一共訓練了 5000 次，圖 4-15 顯示了 10 位成員前 1000 次的收斂結果的平均，可以看出有正確的收斂到 0。

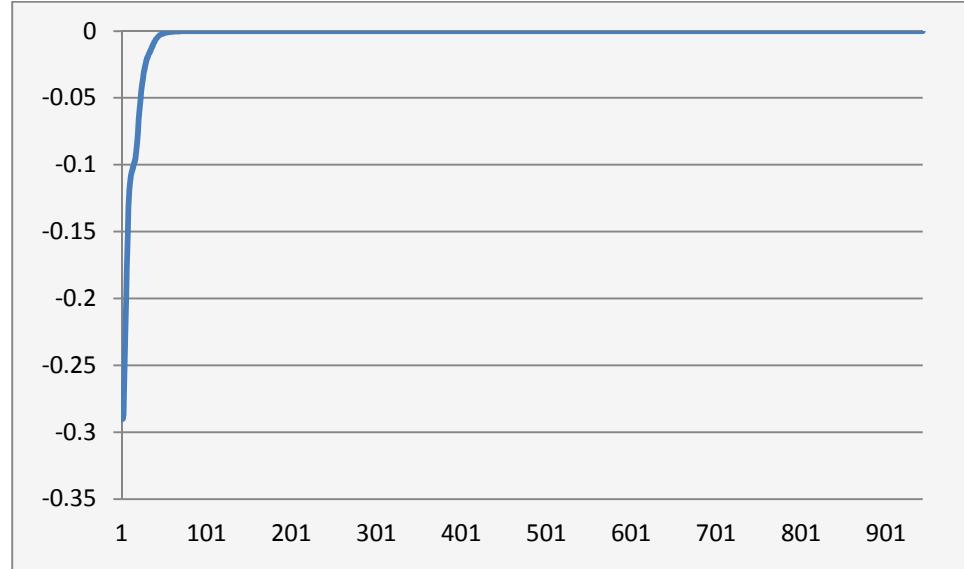


圖 4-14、全體表情辨識類神經網路的收斂情形

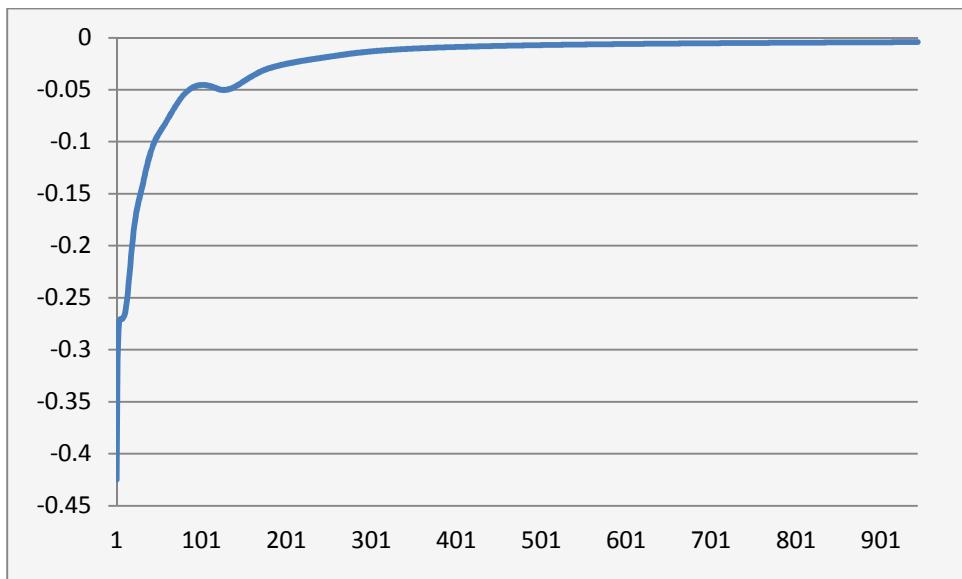


圖 4-15、個人表情辨識類神經網路 10 位使用者的平均收斂情形

表 4-10 到表 4-12 分別是使用全體表情辨識結果、在已知使用者情況下進行個人表情辨識結果、先進行人臉辨識再使用整體個人表情辨識結果，這裡總共採用了 10 位成員的資料輸入，每個人共有 30 張測試影像，每個人的每種表情各有 5 張輸入影像，共 6 種表情，總共輸入影像為 300 張；在表格中上方為輸入的人臉表情影像，左邊前六列是不同表情輸入所辨識出的輸出結果，Error 表示為沒有辨識出來為資料庫的六種表情的結果，而在下面的 Correct 表示為辨識正確的結果，Wrong 表示辨識錯誤的結果，Total 為表情的影像張數，在這裡每種表情各有 45 張，最後 Rate 是各種表情的辨識率。

表 4-10 是使用全體的表情辨識 10 位實驗室成員的辨識率，總共測試影像為 300 張，共有 175 張正確，95 張辨識錯誤，辨識率最低為傷心有 58.0%，最高為高興有 82.0%，平均辨識率為 67.0%。

表 4-11 是使用個人的表情辨識的辨識率，這裡是為了要驗證使用個人的表情辨識的辨識率可以優於使用全體的表情辨識的辨識率，這裡總共測試影像為 300 張，共有 261 張正確，39 張辨識錯誤，辨識率最低為害怕有 84.0%，最高為驚訝有 90.0%，平均辨識率為 87.0%。

最後表 4-12 是使用本論文的個人的表情辨識方法的整體流程於 10 位實驗室成員的辨識率，這裡先進行了人臉辨識，再由辨識出來的成員進行個人的表情辨識，總共測試影像為 300 張，共有 261 張正確，39 張辨識錯誤，辨識率最低為害怕有 84.0%，最高為驚訝有 90.0%，平均辨識率為 87.0%。

由表 4-10 到表 4-12 的結果可以看出，使用個人的表情辨識方法可以讓表情辨識的辨識率顯著的提升，而在各種表情的辨識方面，害怕表情較難有效的辨識出來，而驚訝的表情較能有效的被正確的辨識出來。

表 4-10、使用實驗室成員的全體的表情辨識結果

		INPUT						
		Angry	Disgust	Fear	Happy	Sad	Surprise	Total
OUTPUT	Angry	33	0	1	0	0	1	
	Disgust	1	30	1	0	2	2	
	Fear	0	2	31	1	0	0	
	Happy	1	0	0	41	0	0	
	Sad	2	3	0	2	29	0	
	Surprise	0	3	1	0	1	37	
	Error	13	12	16	6	18	10	
	Correct	33	30	31	41	29	37	201
	Wrong	17	20	19	9	21	13	99
	Total	50	50	50	50	50	50	300
	Rate	66.0%	60.0%	62.0%	82.0%	58.0%	74.0%	67.0%

表 4-11、使用實驗室成員在已知使用者的情況使用個人的表情辨識結果

		INPUT						Total
		Angry	Disgust	Fear	Happy	Sad	Surprise	
OUTPUT	Angry	44	1	0	0	0	1	
	Disgust	1	44	0	0	1	0	
	Fear	2	2	42	0	0	1	
	Happy	0	0	1	43	0	0	
	Sad	1	0	4	0	43	0	
	Surprise	0	1	1	0	1	45	
	Error	2	2	2	7	5	3	
	Correct	44	44	42	43	43	45	261
	Wrong	6	6	8	7	7	5	39
	Total	50	50	50	50	50	50	300
Rate		88.0%	88.0%	84.0%	86.0%	86.0%	90.0%	87.0%

表 4-12、使用實驗室成員的整體人臉辨識與個人的表情辨識結果

		INPUT						Total
		Angry	Disgust	Fear	Happy	Sad	Surprise	
OUTPUT	Angry	44	1	0	0	0	0	
	Disgust	1	44	0	1	1	1	
	Fear	2	2	42	0	0	0	
	Happy	0	0	1	43	0	0	
	Sad	1	0	4	0	43	0	
	Surprise	0	1	1	1	1	45	
	Error	2	2	2	5	5	4	
	Correct	44	44	42	43	43	45	261
	Wrong	6	6	8	7	7	5	39
	Total	50	50	50	50	50	50	300
Rate		88.0%	88.0%	84.0%	86.0%	86.0%	90.0%	87.0%

#### 4.4. On-line 辨識結果

為了符合在機器人的應用需求，能夠及時的辨識出使用者的人臉與表情，因此需要開發出一套整體的及時辨識系統，在這裡展示了 On-line 的辨識結果，圖 4-16 到圖 4-21 為生氣、厭惡、害怕、高興、傷心和驚訝的辨識結果。

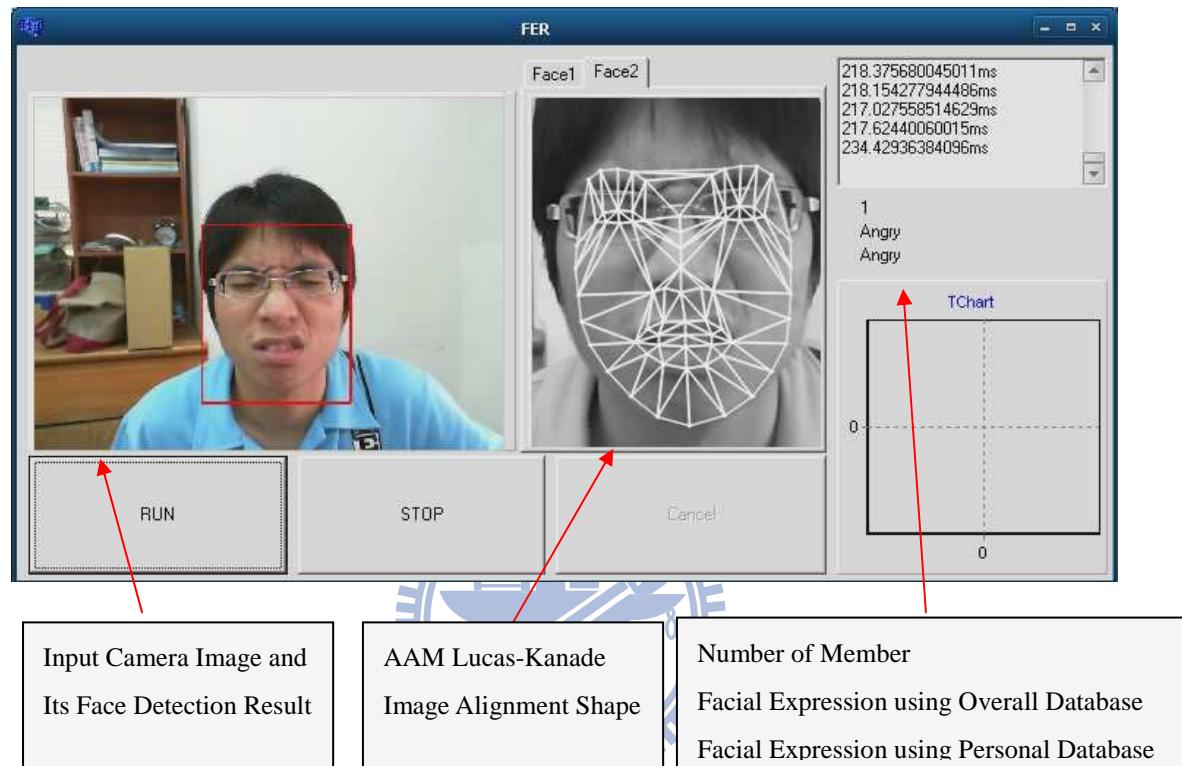


圖 4-16、on-line 辨識結果-生氣

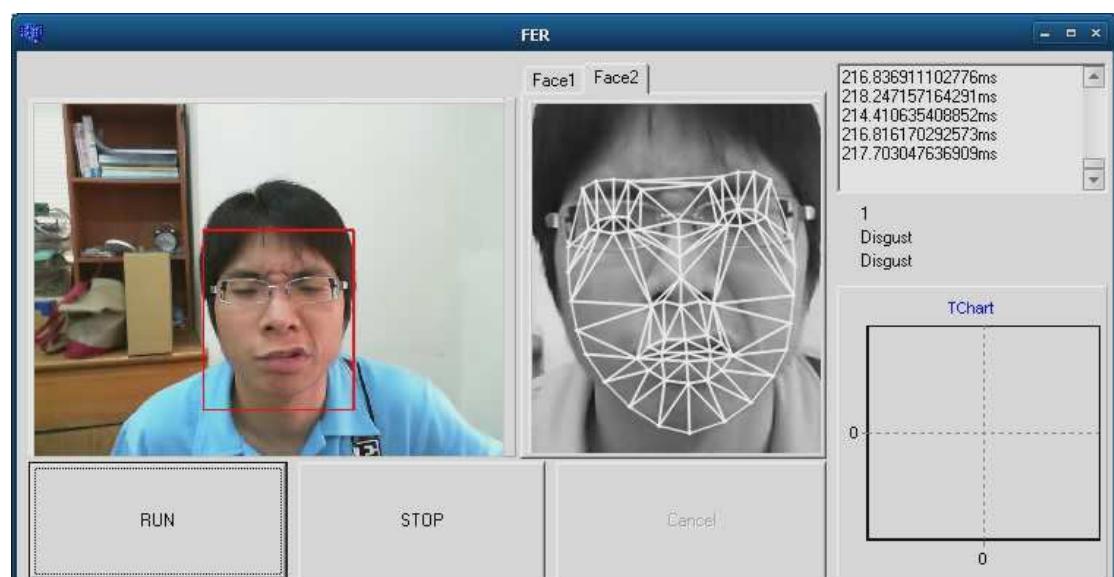


圖 4-17、on-line 辨識結果-厭惡

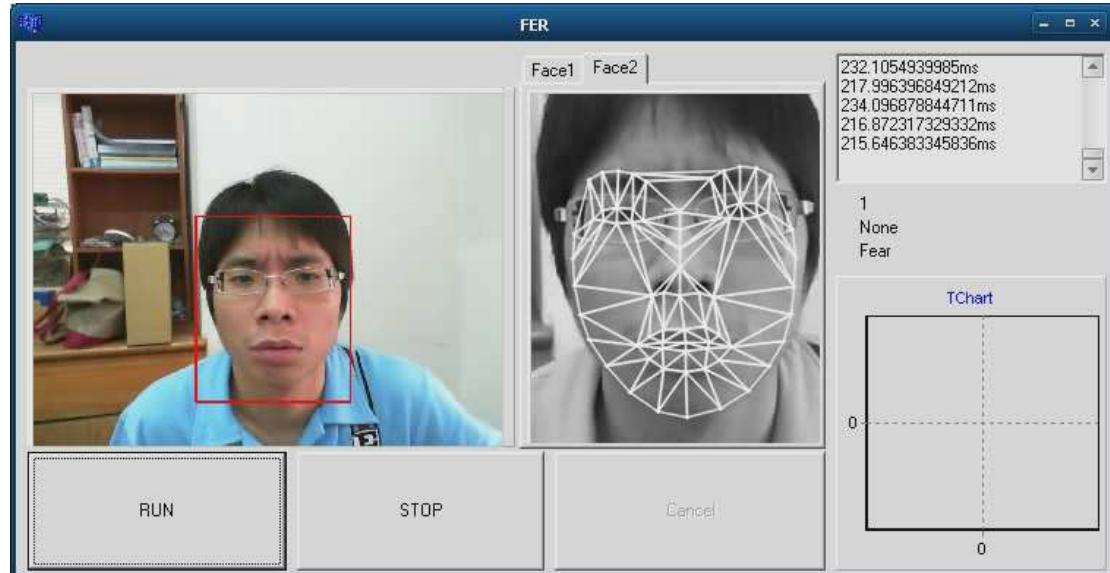


圖 4-18、on-line 辨識結果-害怕

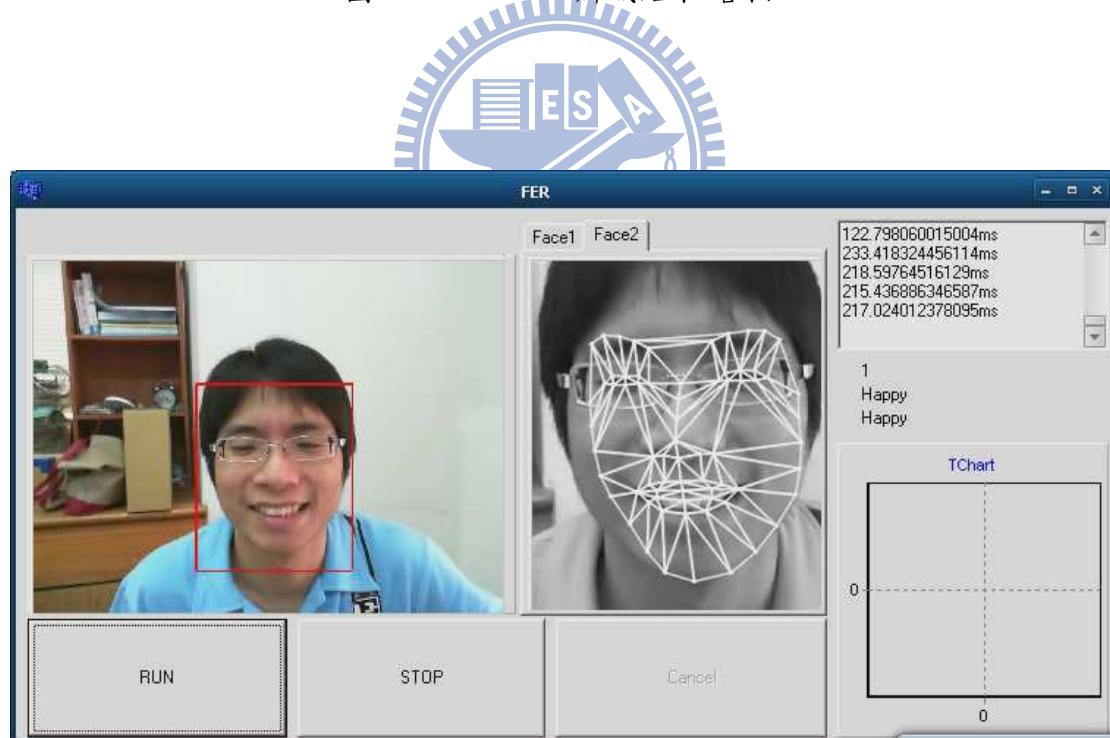


圖 4-19、on-line 辨識結果-高興



圖 4-20、on-line 辨識結果-傷心

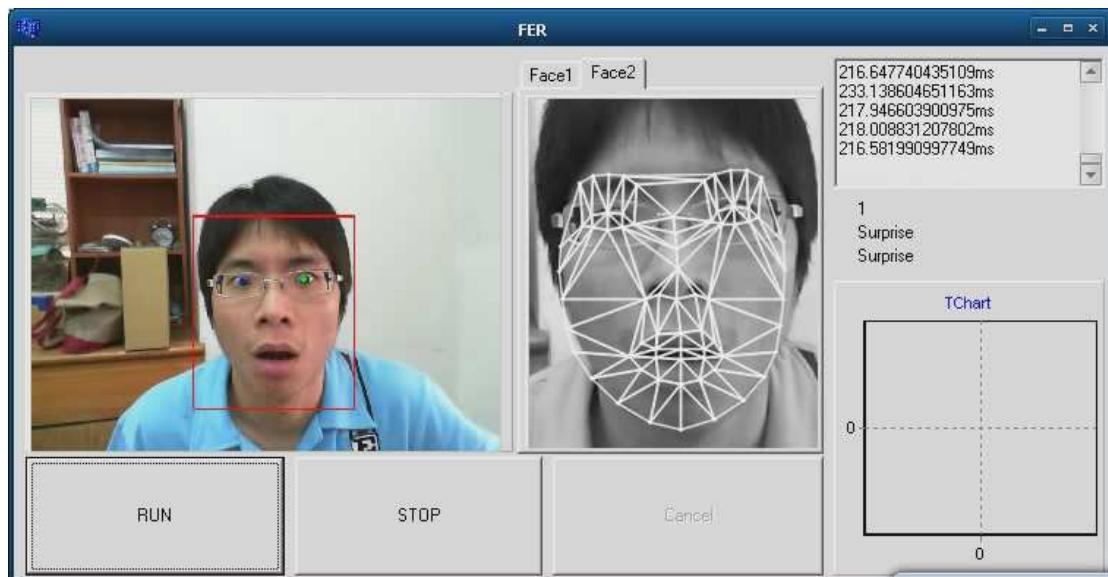


圖 4-21、on-line 辨識結果-驚訝

#### 4.5. 結論與討論

在本實驗中，使用了 BU-3DFE 以及實驗室的人員來進行人臉以及表情的辨識，在 BU-3DFE 資料庫中，使用了 20 位成員的人臉的辨識率可以達到 98.3%，而在表情辨識的部分，使用個人表情辨識方法的結果可以讓辨識率由全體表情辨識的 69.6% 提升到 83.8%；而在實驗室成員的測試中，使用了 10 位成員的人臉的辨識率可以達到 96.7%，而在表情辨識的部分，使用個人表情辨識方法的結果可以讓辨識率由全體表情辨識的 67.0% 提升到 87.0%。

## 第五章、結論與未來展望

### 5.1. 結論

在本論文中，我們提出了一套機器人應用上的人臉與表情辨識系統，可以讓機器人能夠同時的辨識出目前互動的使用者與他的表情。

它首先先進行人臉偵測，並將找到的人臉大小正規化到 200\*240 大小的人臉影像；藉由 PCA 方法來建立出 AAM 人臉的形狀與紋理的模型，其中人臉的模型使用了直方圖等化的方式來加強人臉的特徵與改善亮度；並接著使用改良的 Lucas-Kanade 人臉校正方法找出人臉的特徵，最後先經由倒傳遞類神經網路人臉辨識系統找出目前看到的人是成員中的何人，並接著利用這個人所建立的表情資料庫來辨識出這位使用者目前的表情為何。

使用這個方法，對於系統熟識的使用者，可以減低由於每個人的表情變化因素的不盡相同所產生的辨識錯誤結果，在實驗結果上在使用 BU-3DFE 人臉表情資料庫做辨識，其人臉辨識成功率可達 98.3%，而使用個人化表情辨識的成功率為 83.8% 相對於使用全體的表情辨識的 69.6%，其成功率可大幅提升。



### 5.2. 未來展望

由於目前的辨識系統並未達到完全令人滿足，因此以下有幾點是可以改進的地方：

在人臉特徵點的部分，由於在表情動作太大時，會發生特徵點迭代錯誤的結果，因此或許可以利用建立更豐富的資料庫或是加入一些局部特徵辨識的方法，來改善這個問題。

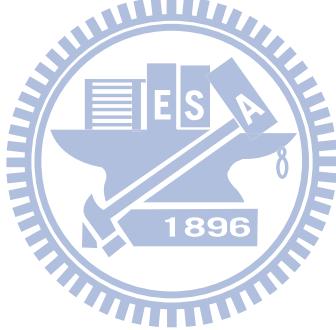
在目前的系統當中，是在每一個 frame 當中先找出影像中的人臉區域來進行 AAM 的 Lucas-Kanade 影像校正，因此可以修改為在當有人臉剛出現在攝影機影像中時，先對於這個人臉區域來進行 AAM 的 Lucas-Kanade 影像校正，然後在接

下來的每個 frame 中，只要對於在前一個 frame 裡已經找出的人臉形狀和位置再進行 AAM 的 Lucas-Kanade 影像校正，如此的話將可以提升 Lucas-Kanade 影像校正的結果，並可以降低一些迭代的次數，以降低運算量。

在人臉角度的變化方面，希望能參考文獻[30]中結合二維與三維的方法，以提升人臉邊轉角度過大時所可能無法正確匹配的問題。

在目前的系統當中，是先找出影像中的唯一一張人臉來做即時的辨識，未來希望可以同時對影像中的多人臉進行辨識，可以達到更有效的互動行為。

最後希望將本論文所提出之人臉辨識與表情辨識之整合設計方法應用於在實際的機器人與人的互動應用中，在實際的人與機器人互動中，對於變化的人臉能夠準確的進行辨識，並加入機器人互動行為的設計，以發展出有高互動性的機器人。



## 參考文獻

- [1] Paul Ekman and Wallace V. Friesen, *Unmasking the face*, Prentice-Hill Inc. , Englewood Cliffs, New Jersey, 1975.
- [2] 洪濬尉, 機器人之表情辨識快速學習法則, 碩士論文, 國立交通大學電機與控制工程學系, 2007.
- [3] 鐘仁厚, 基於模糊邏輯之臉部表情辨識, 碩士論文, 國立中央大學電機工程學系, 2008.
- [4] D. Cristinacce and T. F. Cootes, “Facial Feature Detection and Tracking with Automatic Template Selection,” in *Proc. of 7th IEEE International Conference on Automatic Face and Gesture Recognition*, 2006, Southampton, UK , pp. 429-434.
- [5] D. Cristinacce and T.F.Cootes, “Feature Detection and Tracking with Constrained Local Models,” in *Proc. of British Machine Vision Conference*, 2006, Edinburgh, UK, pp.929-938.
- [6] T. Hao and T. S. Huang, “3D Facial Expression Recognition Based on Automatically Selected Features,” in *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2008, Anchorage, Alaska, USA, pp. 1-8.
- [7] Igor S. Pandzic and Robert Forchheimer (Eds), *MPEG-4 Facial Animation: The Standard, Implementation and Applications*, John Wiley & Sons, Inc., 2002.
- [8] S. Kullback, “The Kullback-Leibler distance,” *The American Statistician* 41:340-341, 1987.
- [9] C. Martin, U. Werner and H. M. Gross, “A Real-Time Facial Expression Recognition System Based on Active Appearance Models Using Gray Images and Edge images,” in *Proc. of 8th IEEE International Conference on Automatic Face & Gesture Recognition*, 2008, Amsterdam, Nederland, pp. 1-6.

- [10] P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," in *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001, Kauai Marriott, Hawaii, pp. I-511-I-518.
- [11] M. Mufti and A. Khanam, "Fuzzy Rule Based Facial Expression Recognition," in *Proc. of International Conference on Computational Intelligence for Modelling, Control and Automation, and International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, 2006, Vienna, Austria, pp.57-61.
- [12] 許晉懷, .結合影像及語音之雙模情緒辨識系統, 碩士論文, 國立交通大學電機與控制工程學系, 2006.
- [13] Rainer Lienhart and Jochen Maydt. "An Extended Set of Haar-like Features for Rapid Object Detection," in *Proc. of International Conference on Image Processing*, 2002, Rochester, New York, USA , pp. 900-903
- [14] M. Soriano, B. Martinkuuppi, S. Huovinen and M. Laaksonen, "Skin Detection in Video Under Changing Illumination Conditions," in *Proc. of 15th International Conference on Pattern Recognition*, 2000, Kauai Marriott, Hawaii, pp. 839-842.
- [15] C. C. Chiang, W. K. Tai, M. T. Yang, Y. T. Huang and C. J. Huang, "A Novel Method for Detecting Lips, Eyes and Faces in Real Time," *Real-Time Imaging*, vol. 9, no. 4, pp. 277-287, 2003.
- [16] T. F. Cootes, G. J. Edwards and C. J. Taylor, "Active appearance models." in *Proc. of 5th European Conference on Computer Vision*, 1998, Springer, Berlin, pp 484–498.
- [17] T. F. Cootes, G. J. Edwards and C. J. Taylor, "Active Appearance Models," In *IEEE Transactions on Pattern Analysis And Machine Intelligence*, Vol. 23, No. 6, 2001, pp. 681-685
- [18] T. F. Cootes and C. J. Taylor, *Technical Report: Statistical Models of Appearance for Computer Vision*, The University of Manchester School of Medicine, 2004.

- [19] T. F. Cootes, C. J. Taylor, D. H. Cooper and J. Graham, "Active Shape Models - Their Training and Application," *Computer Vision and Image Understanding*, Vol. 61, no.1, pp.38-59, 1995.
- [20] 王仕傑, 多姿態人臉辨識及其在機器人與人互動之應用, 碩士論文, 國立交通大學電機與控制工程學系, 2008.
- [21] <http://www2.imm.dtu.dk/~aam/>
- [22] I. Matthews and S. Baker, "Active Appearance Models Revisited," *International Journal of Computer Vision*, vol. 60, no.2, 2004, pp.135-164.
- [23] A.M. Martinez and R. Benavente, *The AR Face Database*, CVC Technical Report #24, June 1998.
- [24] C. Goodall, "Procrustes Methods in the Statistical Analysis of Shape," *Journal of the Royal Statistical Society B*, Vol 53, no.2, pp.285-339, 1991.
- [25] J.R. Shewchuk, "Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator," In *Applied Computational Geometry, FCRC'96 Workshop*, pp. 203-222. Springer-Verlag, 1996.
- [26] I. Matthews and S. Baker, "Lucas-Kanade 20 Years On: A Unifying Framework," *International Journal of Computer Vision*, Vol. 56, No. 3, 2004, pp. 221 - 255.
- [27] P. Jia, "Active Appearance Model," Online technical report available from [http://www.visionopen.com/members/jiapei/publications/pei\\_aamreport2010.pdf](http://www.visionopen.com/members/jiapei/publications/pei_aamreport2010.pdf)
- [28] 羅華強, 類神經網路—MATLAB 的應用, 高立圖書, 2005
- [29] Lijun Yin, Xiaozhou Wei, Yi Sun, Jun Wang and Matthew J. Rosato, "A 3D Facial Expression Database For Facial Behavior Research," in *Proc. of 7th IEEE International Conference on Automatic Face and Gesture Recognition*, 2006, Southampton, UK , pp. 211-216.
- [30] X. Jing, et al., "Real-time combined 2D+3D active appearance models," in *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004, Washington, DC, USA, pp. II-535-II-542 Vol.2.