國立交通大學

多媒體工程研究所

碩士論文

影片中人物分群方法之研究

The Investigation of Clustering Algorithms for Clustering People in Video

研究生：魏良佑

指導教授：王才沛　教授

中華民國　一百年八月

影片中人物分群方法之研究

The Investigation of Clustering Algorithms for Clustering People in Video

研 究 生：魏良佑　　Student：Liang-you Wei

指導教授：王才沛　　Advisor：Tsai-pei Wang

國 立 交 通 大 學

多 媒 體 工 程 研 究 所

碩 士 論 文

A Thesis
Submitted to Institute of Multimedia Engineering
College of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in
Computer Science
Aug 2011
Hsinchu, Taiwan, Republic of China

中華民國一百年八月

影片中人物分群方法之研究

學生：魏良佑　　指導教授：王才沛

國立交通大學多媒體工程所

摘要

本文主要是研究影片中人物分群的方法，人物分群最明顯的特徵就是臉部影像，但是影像的解析度、光影、拍攝角度、膚色、場景、分鏡 (shot) 卻會大大影響分群的結果，因此本文還利用身體影像以及影片的時間資訊作為分群輔助，我們把相似的人物先合併成演員串列，如此可避免過多雜亂的臉部影像降低效能，並計算串列間的人物相似度矩陣。

上述是利用一些客觀的條件處理人物分群，本文還結合叢集整合的概念，試著將這些演員串列分群，可得到較細膩的叢集整合相似度矩陣，藉此了解串列間的相似程度，人物相似度矩陣與叢集整合相似度矩陣兩者依據串列的時間差距決定權重值，分別乘上權重值加總過後，即是最終用來分群的相似度矩陣，搭配階層式凝聚法，我們可以得到最終的分群結果。

The Investigation of Clustering Algorithms for Clustering People in Video

Student：Liang-you Wei　　Advisor：Tsai-pei Wang

Institute of Multimedia Engineering

College of Computer Science

National Chiao Tung University

Abstract

We investigated clustering algorithm for clustering people in video in this paper. Face image is the most obvious feature of people, but its resolution, luminance, shadow, shooting angle, skin color, and shot, will greatly affect clustering, so we also used the body image and movie time information as the auxiliary. We aggregated the similar people image to the same actor sequence, it can avoid that many disorderly face images reduced performance, and then we computed person similarity matrix between sequences for use.

Above-mentioned use some objective condition to cluster people, and we also integrated the concept of cluster ensemble, and we tried to cluster the actor sequences. The ensemble similarity matrix is more exquisite than the person similarity matrix. It can help us to realize the similarity between actor sequences. Person similarity matrix and ensemble similarity matrix product their own weight which be computed according to the difference of time, and we sum up the two products of similarity matrix and their own weight, taking it as the final similarity matrix for clustering. We used the final similarity matrix on hierarchical agglomeration to find the final clustering.

誌謝

本文能夠完成，首先得感謝我的指導教授 王才沛老師，老師的細心指導，讓我在研究受挫時能知道明確的目標在哪，與老師討論總能有新的體悟，對於實驗也能即時給我不同的想法。也很感謝陳祝嵩教授、林文杰教授、黃雅軒教授抽空擔任我的口試委員，對於論文的內容以及實驗的方法給我許多很好的建議。

另外，也要感謝實驗室的三位學長，蘇偉誌、徐崇桂、林俞邦，碩一有你們的照顧，當自己有不懂的專業問題詢問你們時，你們總能不厭其煩地教導我，實在感到很窩心，還有我的同學蘇裕傑、林俞丞、邱俊予，跟你們一起渡過這兩年打球、趕報告、吃大餐、CVGIP 得獎、作論文的日子，實在很多彩多姿，與學弟楊堡評的相處就像好朋友一樣，這個碩士生涯肯定讓我回味無窮。

最後，要感謝的是我的母親，在這兩年的碩士生涯妳讓無後顧之憂，全心全意地完成我的碩士學位，每當自己做了決定跟妳報備，妳總是支持著我，真的很感謝妳。

目錄

圖例

表格

第一章　簡介

1.1　研究目的

科技進步的今日，從人手一支的可錄影手機、大街小巷的都裝設監視器，不難發現影片早已成為記錄生活上點點滴滴的主要媒介，原本影片需要較大空間存放的限制，拜科技之賜，記憶體的快速發展，此束縛逐漸被解放，因此影片的使用頻率大增；另外一個原因，是因為影片比圖片包含更多的資訊，例如：時間資訊，有了時間資訊我們可以判斷影像中變化的先後次序。

現今，觀賞影集或電影早已成為主要的生活娛樂之一，然而，當我們在挑選想要觀賞的影片時，或是突然想喚起影片中某個情節的回憶之際，在眾多的影片中瀏覽漫長的片段，使我們不易尋找也耗費相當多的時間成本，這時如果有一套系統可以標記片段索引輔助我們快速瀏覽，則我們可以有效地得到我們想要的影片資訊。除了視聽娛樂上的需求，警政單位也常常需要瀏覽監視器的影片協助辦案，因此發展一套快速瀏覽影片的工具是必要的。目前已有許多研究是依據影片內容而發展的技術，例如：視訊瀏覽（video browsing）、影片摘要（video summarization）等等，可提供使用者影片內容的資訊，尚未觀賞影片的使用者，也可藉由影片呈現的資訊快速篩選出自己想觀賞的影片。

上述的視訊瀏覽以及影片摘要，主要是針對影片中的整個畫面作處理以及辨識，而本文為了想達到快速瀏覽影片的目的，因此必須對人物出現的時間建立一個時間軸，時間軸資訊可供觀賞者快速瀏覽片段，也可瀏覽指定演員演出的精采片段。

人物在影片中出現的時間軸將被當成人物索引，而人物索引最基本的作法即是利用最具特徵的臉部影像，但臉部影像的解析度、光影、拍攝角度、膚色、場景、分鏡將會大大影響索引的結果，因此本文中還使用身體影像以及影像產生的時間資訊作為輔助。首先，我們把畫面裡所有被偵測到的臉部影像，以及正下方

長寬為兩倍臉體影像大小的身體影像，以及串列產生的時間資訊，當作是輸入的資訊，依據客觀的條件建立演員串列，試著將演員串列分群，分群的結果搭配演員串列所產生的時間，即可得到人物索引。圖 1-1 為人物索引的示意圖。

圖 1-1　演員索引示意圖

分群時，除了利用臉部影像、身體影像以及時間資訊產生人物相似度矩陣，我們也加入叢集整合的概念，試著讓這些臉部影像分群，找出他們的叢集整合相似度矩陣，最後依據串列產生的時間差距決定兩個相似度矩陣的權重，以合併過後的相似度矩陣作為最後分群的依據，執行凝聚演算法，最終由實驗可得知，加入叢集整合相似度矩陣確實有助於提升分群效能。

1.2　論文架構

在本篇的第二章節敘述人物分群以及叢集整合的相關文獻，第三章介紹如何產生演員串列，包含如何生成以及篩選演員串列的條件，第四章則是敘述從演員串列到分群的實驗步驟，第五章是呈現實驗結果以及探討，第六章是總結本篇的結論以及未來展望。

第二章 文獻探討

2.1 人物分群（People Clustering）

不論是在圖片或是影片中，臉部影像都是人物的主要特徵，因此使用臉部影像或是其他資訊執行分群，達到辨識人物的效果，就稱為人物分群。實際上利用臉部影像執行分群，時常會因為圖片或影片中的解析度、光影、臉部姿勢、膚色、場景、分鏡等等，各種因素影響分群的效能，因此〝以臉部影像為主軸利用其他條件輔助人物分群的方法〞才陸陸續續被提出。

在[1]中提到，圖片中的臉部影像資訊與影片中的臉部影像資訊有所不同，影片中可以帶來更多除了臉部資訊外的訊息，例如：時間資訊，正確使用可提升分群效能；[2]也提到在影片中利用畫面的相依性以及時間軸資訊，把系統偵測到的所有臉部影像利用條件限制產生演員串列（actor sequence），再使用分群演算法把相似的演員串列聚集在同一群；[2]也提出重要的兩個觀點，第一，產生演員串列時，串列內的臉部影像在時間軸上不能產生重疊（overlap），而且在同一個串列裡的臉部影像我們將認定他們是相同的演員；第二，串列使用分群法聚集時，若兩個演員串列裡的臉部影像時間軸發生重疊，則此兩個串列必定屬於不同的演員，因此將不被聚集在同一群內。

對於影片中時間資訊的使用，在[3]文中有提出更深入的作法，作者使用前後總共拍攝 11 年的影集作為人物分群的資料庫（database），這些影集並非在完全固定環境的情況下拍攝而成，使得人物分群的難度增加；但是在這麼長的影集中，演員的年齡、髮型、外觀等等，都會隨著時間逐漸改變，我們也擁有更長的時間軸資訊，因此作者建議除了利用臉部影像外，頭髮以及衣服資訊都是可以輔助分群的利器。對於時間軸的使用，作者提出三個概念，首先，在小範圍的時間差距內，例如在同一個分鏡，我們可以利用人物的移動（motion）以及對人物的追蹤（tracking），把相似的臉部影像聚集成一個演員串列，第二，在中範圍的時

間差距內，例如不同的分鏡但仍屬同一集，可利用人物的頭髮以及衣服影像把不同分鏡的演員串列聚集起來，最後，在大範圍的時間差距內，例如不同集，由於演員在不同集身著的衣服資訊早已大不相同,因此不能再使用衣服資訊作為分群的提示。相同演員在同一集中的頭髮與衣服資訊參考原則是，時間差距愈小參考價值就愈大，因此[3]提到使用權重的方式調整輔助資訊的參考值。[4]也提出相同的概念，先在每個演員串列中挑選一個關鍵臉（keyface），再利用關鍵臉的SIFT (Scale Invariant Feature Transform)特徵、膚色、身體影像的色彩直方圖，搭配分群演算法執行人物分群。

影像本身的特性會影響分群效能，而臉部姿勢也會造成分群上極大的困擾，[5]提到〝不同人物在類似的臉部姿勢下，臉部影像間的相似度〞比〝相同人物在不同臉部姿勢下，臉部影像間的相似度〞還來得高，這表示針對臉部姿勢的分群比針對人物的分群還容易，因此[5]的作者提出兩階段分群法，首先針對臉部姿勢分群，再來才對相同臉部姿勢的影像人物分群，如此確實能提升分群效能。

本節第三段有提到分鏡資訊可作為人物分群的輔助資訊，但是它能夠直接拿來作分群嗎？[6]提到，儘管基於人臉辨識（face recognition)的分群法可以有不錯的效能，但卻要花費較大的時間成本，因此[6]文中就提出兩個較快速的臉部影像分群法，SSC (Similar Shot-Based)與 SSC+FTC (Face Thumbnail Clustering)，它們之所以比較快速，是利用人物出現所在場景的相似度取代傳統臉部特徵向量的距離計算，而實驗數據顯示，僅管 SSC 與 SSC+FTC 處理長度為一小時影片的分群效能，比人臉辨識的分群法平均降低 6%與 0.9%，但卻只花費了 0.35 秒與 31 秒，如此快的速度若使用在大型資料庫上，肯定能節省不少時間成本。

2.2 叢集整合 (Cluster Ensemble)

在已知資料的分佈情況下，我們可以採用現今對其分佈有較好效能的分群法執行分群；但是實際上在很多時候我們處理的是未知的資料，或者，資料的分佈

根本雜亂無章，甚至即使知道資料的分布情況也沒有最佳分群法可使用的狀況下，這時我們不得不一一嘗試各種分群法尋找最佳解，但這是沒有效率的，也並非一個完整且完善的解決之道。

叢集整合即是解決上述情況的一個概念，它主要的作法是參考各種不同演算法或是相同演算法在不同條件下產生的〝分割〞(本文中我們稱一個分群結果為〝分割〞(partition))，從這些分割中挑出或是組合出最佳的整合結果 (ensemble)，如此一來，演算法對於某些特定資料的適用性之缺點就可加以改善[7]。另外，在叢集整合中使用分割時也只是使用其分群結果，而不會存取到最原始的資料特徵向量或是萃取特徵的方式，甚至也不需理會是何種演算法所產生的分割，這就是知識重用 (knowledge reuse) 的概念[8]。

圖 2-1 是叢集整合的示意圖，我們以 X 代表一個包含 n 個物件 (object) 的資料集 (dataset)，$\Phi^{(1,\ldots,r)}$ 為 r 個產生分割的函數，$\lambda^{(1,\ldots,r)}$ 為 r 個標籤 (label)，每一個標籤都是 n 維的向量，向量的值即代表物件所屬的群號，Γ 是共識函數 (consensus function)，主要目的是整合 $\lambda^{(1,\ldots,r)}$ 形成最終整合的結果 λ。

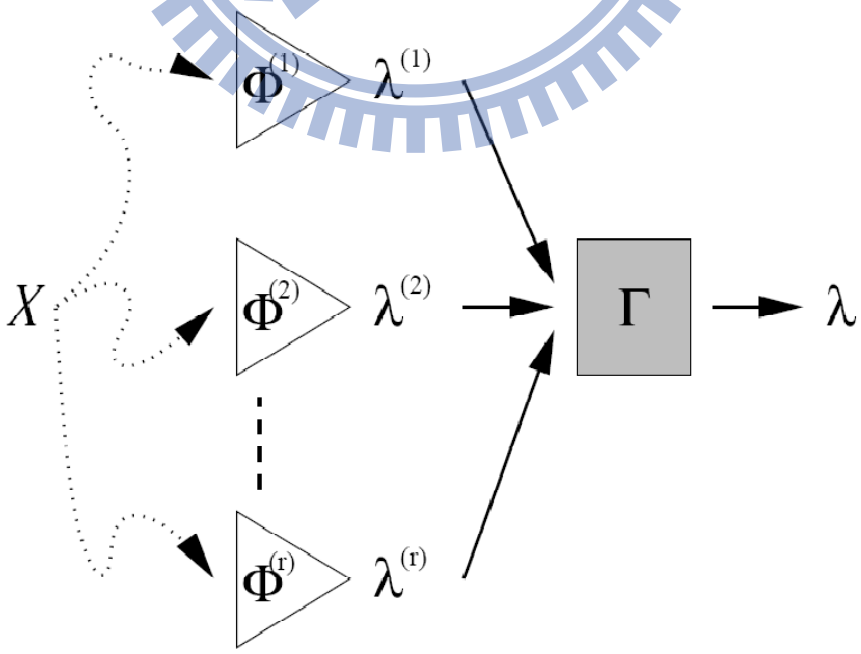


圖 2-1　叢集整合示意圖
資料來源[8]

叢集整合需要許多擁有高質量（quality）且多樣性（diversity）的分割[9]，高質量指的是分割具有高正確率，而這些分割也不能完全相同，必須有差異性才能創造出新的分群結果，這裡所說的差異性即多樣性，有這兩項條件才有充足的資訊產生最正確的整合結果，也由於如此，叢集整合的計算量頗大，且具有無法使用在高維度以及大型資料集（dataset）的缺陷，針對這兩點缺陷，[9]使用 RP（Random Projection），將特徵降維至〝有興趣〞的子空間，改善第一點；[10]則是利用 CBEC（Centroid Based Ensemble Clustering Algorithm）來產生整合結果，如此可改善第二點，並且在整合過程中濾除干擾（outlier）。[11]藉由〝單次瀏覽〞（single scan）的方式產生整合結果，進而提高執行速度，改善叢集整合時速度過慢的缺點。

叢集整合的第一個步驟就是要產生相異的分割，產生的過程中有許多變數，我們若改變其中一個變數使得分群結果改變即可得到新的分割，其中包含幾個重點，選擇不同的分群演算法、在相同演算法中選擇不同的參數值、不同的初始化[7]，另外，如改變物件的描述方式、改變物件的前置作業、重新取樣（resampling）、部分取樣（subsampling）都有可能產生擁有高質量和多樣性的分割；[12]就是以重新取樣來產生分割，並證實重新取樣更健全（robust）與穩定（stable），[13]也認為拔靴重複抽樣法（bootstrap resampling）可避免原始資料的小變動影響分群結果。

收集許多高質量與多樣性的分割之後，需要整合他們產生最終的整合結果，這就是共識函數的工作，[14]提出兩種評估共識函數好壞的方法。共識函數最常使用證據累積（evidence accumulation）法，證據累積即是把多個分割視為各自獨立的證據，利用共相關函數[7]（co-association function）統計兩兩物件在不同群數的分群結果中之相似程度，最後利用相似度矩陣搭配分群演算法產生最終的分群結果，其中分群法以階層式凝聚法（hierarchical agglomeration）最為常見。[15]也以證據累積為主要的概念，設計 EBSC（Evidence-Based Spectral Clustering）針對於資料中有混著數字與文字的物件分群；此外，[16]也提出另一種新的共識函

數的概念，機率累積（probability accumulation），與證據累積差別在於共識函數不同，此方法實驗在三個人造的資料集上，效能顯示優於證據累積。

產生叢集整合的分群結果，除了上述利用共識函數的方式外；另外一種是把叢集整合的問題轉換為圖形切割的問題，[8]提出 CSPA（Cluster-based Similarity Partitioning Algorithm）、HGPA（HyperGraph-Partitioning Algorithm）、MCLA（Mera-CLustering Algorithm）產生最終的分群結果，而[17]也提出把叢集整合轉換成二分圖形分割（bipartite graph partition）的問題，利用 HBGF（Hybrid Bipartite Graph Formulation）產生分群結果。最終[18]提出一個藉由收集雜訊（noise）至雜訊族群（noise cluster）的想法，可以把效能提升。對於整合的結果我們利用[19]、[20]計算分群的效能，藉此觀察並改善變數對於分群結果的影響。

第三章　前置作業

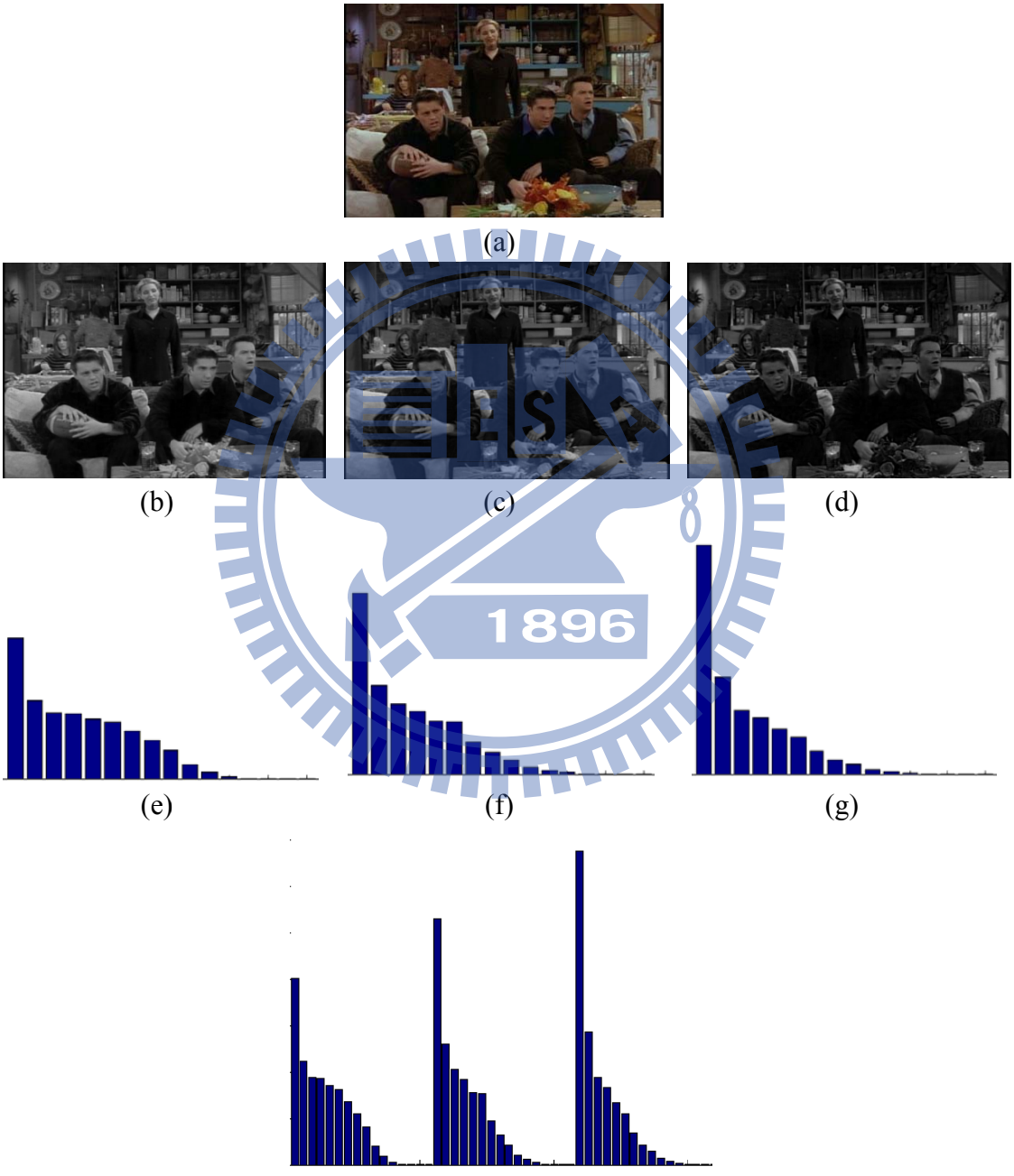
實驗時，我們模擬[21]提到的產生演員串列的規則。3.1 節說明如何從影片到產生臉部影像以及產生分鏡的資訊，3.2 節介紹建立粗略演員串列的條件，3.3 節介紹臉部影像的前處理以及如何分割演員串列，3.4 節則敘述如何篩選已建立的演員串列並產生重疊（overlap）資訊，3.5 節則說明如何計算人物的相似度。

3.1　產生臉部影像與分鏡偵測

本文探討在影片中的人物分群，因此要把影片中的臉部影像全部擷取出來，另外我們也需要分鏡資訊輔助臉部影像結合成為演員串列，因此得事先產生每個臉部影像的分鏡資訊。測試影片是以 30fps（frames per second）的速度播放，為了避免過多相同的影像拉長程式執行時間，因此我們以 5fps 的速度，利用“Free Video To JPG Converter”[22]這套軟體，從影片中擷取影像作為後續處理的資料。把影片切割成影像後，我們利用 OpenCV [23]快速地偵測每一張影像中可能的人臉位置。

影片中的攝影鏡頭常常圍繞在故事焦點的人物身上，因此當主角在同一個鏡頭（take）中出現一段時間，以 5fps 的速度擷取影像，經由 opeCV 的臉部偵測，我們將會得到許多同一演員且很相似的臉部影像，為了降低許多同演員且相似臉部影像的獨立處理時間，我們將這些連續出現的同演員影像合併為演員串列，後續以演員串列為單位處理人物分群。由於影片中的場景或鏡頭轉換將使影像背景明顯變化，因此我們使用色彩直方圖（color histogram）的方法，計算前後張影像的直方圖向量，兩者的歐氏距離（Euclidean distance）若小於臨界值則被認定是同一場景。色彩直方圖的計算可幫助我們了解影像內像素值（pixel value）的分佈。實驗的影像皆以無符號八位元的數字來表示，且以 16 為一個區間（interval），總共可分成 16 個區間，統計影像中像素值落在各區間的個數即為

色彩直方圖，若把影像的 RGB 成分都計算直方圖向量並合併，則一張影像可以得到1×48的向量，我們定義場景相異度為兩張影像的直方圖向量之歐氏距離，若超過5000，則我們把這兩個影像視為不同場景。下圖 3-1 為色彩直方圖的範例，圖 3-1(a)為原始圖，(b)~(d)分別為原始圖的 RGB 成分影像，(e)~(g)分別為(b)~(d)的色彩直方圖，最後(h)為合併(e)~(g)的色彩直方圖。



(a)

(b) (c) (d)

(e) (f) (g)

(h)

圖 3-1　色彩直方圖範例

(a)為原始圖，　(b)~(d)分別為(a)的 R、G、B 成分影像

(e)(f)(g)分別為(b)、(c)、(d)的色彩直方圖，　(h)為(e)(f)(g)的合併圖

3.2 建立粗略的演員串列

依照[21]我們不僅利用 openCV 偵測到的臉部方框，也利用連接在臉部方框底下，長寬為兩倍臉部影像大小的範圍作為身體影像的資訊。首先，我們認為，要被蒐集在同一個演員串列內的影像必須屬於同一個場景，如此可避免分鏡時納入不同演員的影像；影像是連續性的，因此我們也必須對於臉部影像的大小與位置作篩選，如果按照上述三個條件所形成的演員串列成員個數太少，我們會無法從中得到足夠的串列資訊來處理人物分群，因此四個條件被設計如下[12]，唯有四個條件都通過才能成為串列裡的一員。

- 演員串列裡的每個臉部影像必須是同一場景且出現在連續的影像中。
- 前後兩張臉部的中心位置相差距離必須在 35 個像素內。
- 每張臉部大小必須為前一張臉部大小的 0.5 至 1.5 倍之間。
- 經由上述三點所建立的演員串列中，影像個數必須超過 3 個。

3.3 臉部影像處理及分割演員串列

為了避免太亮或太暗的臉部影像造成計算相似度的差異性太大，因此我們必須統一臉部影像的特性，在[21]中影像的前處理嘗試了，光影平衡加上高斯低通濾波器 (Gaussian low-pass filter)，以及 LoG (Laplacian of Gaussian)兩種方式，而我們選擇效能較好的前者。公式(1)的 x 是待為調整的像素值，μ 與 δ 分別為該張影像所有像素值的平均值與標準差，μ_0 與 δ_0 為期望整張影像調整過後的平均值與標準差，實驗時測試的影像皆以無符號的八位元表示，因此我們設定 μ_0 為 127、δ_0 為 50，最後得到 x 轉換過後的像素值為 x' 。

$$x' \to (x-\mu)\times\frac{\delta_0}{\delta}+\mu_0 \tag{1}$$

調整完影像特性之後，為了影像處理的方便性，我們把所有 openCV 所偵測

到的臉部彩色影像，以公式(2)把 RGB 色彩空間的像素值(p_r, p_g, p_b)轉換成一維的灰階影像值 p_{gray}。

$$p_{gray} = 0.299 \times p_r + 0.587 \times p_g + 0.114 \times p_b \tag{2}$$

灰階影像經由光影平衡之後，再利用標準差為 1，5×5 的高斯低通遮罩 (mask) (3)過濾，最後調臉部整影像大小為 70×70 的大小，而圖 3-2 是從 openCV 偵測擷取的臉部影像到降維至 70 維度影像的過程。

$$\frac{1}{273}\begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix} \tag{3}$$

(a) (b) (c) (d)

(e) (f) (g) (h)

圖 3-2　臉部影像處理範例圖

(a)為原始圖 1，(b)為(a)之光影平衡，(c)為(b)之灰階影像，(d)為(c)之高斯影像
(e)為原始圖 2，(f)為(e)之光影平衡，(g)為(f)之灰階影像，(h)為(g)之高斯影像

儘管每張影像已經調整至 70×70，但是運算量仍然過大，因此我們先把每張 70×70 的影像擺放成 1×4900 的向量，本文中以$\boldsymbol{\upsilon}_{\mathrm{F}}$表示，之後利用 PCA (Principle Component Analysis) 的方法，計算$[\boldsymbol{\upsilon}_{\mathrm{F1}}{}^{\mathrm{T}}, \ldots, \boldsymbol{\upsilon}_{\mathrm{Fi}}{}^{\mathrm{T}}, \ldots]$的特徵值 (eigenvalue) 與特徵向量 (eigenvector)，一個 1×4900 的特徵向量記作 $\boldsymbol{e}$，實驗中我們取出前 70 大特徵值所對應的特徵向量，作為臉部影像的降維矩陣$\mathrm{B_F} = [\boldsymbol{e}_1{}^{\mathrm{T}}, \ldots, \boldsymbol{e}_{70}{}^{\mathrm{T}}]$，利

用公式(4)，$\boldsymbol{v}_F$被降維至1×70 的向量$\boldsymbol{v}_f$；考慮臉部與身體降維向量在計算相似度／相異度時能有相同的比重，因此我們只先把140×140 的身體影像調整大小至70×70，並擺放成1×4900 的$\boldsymbol{v}_B$向量。

$$\boldsymbol{v}_F \, B_F = \boldsymbol{v}_f \tag{4}$$

[21]提到為了避免在接連的分鏡上有不同的演員佇立在相同的位置上，造成串列裡的人物不一致，將會降低分群效能，因此我們利用兩個條件排除此種狀況。若在同一個演員串列中的前後$\boldsymbol{v}_F$與$\boldsymbol{v}_B$分別超過臉部相異度門檻值以及身體相異度門檻值，則我們將以兩者之間作為界線，分割成兩個子串列。

臉部相異度被我們定義為，前後兩個$\boldsymbol{v}_f$向量差距之歐氏距離，而身體相異度的計算我們則是使用三維色彩直方圖（3D color histogram），計算前後兩個$\boldsymbol{v}_B$的三維色彩直方圖向量差距之歐氏距離即為身體相異度。實驗中，我們設定臉部相異度門檻值為 2000，身體相異度門檻值為 2500，圖 3-3 是一個串列分割的範例，圖 3-3(a)為測試的演員串列，圖 3-3(b)為串列分割的結果，表 3-1 為前後影像的臉部以及身體相異度。

(a)

(b)

圖 3-3　分割演員串列範例圖
(a)演員串列，(b)為(a)的分割結果

三維色彩直方圖（3D color histogram）的概念與色彩直方圖相當類似，只不過前者事先把三維的像素值轉換成一維，再作直方圖統計。舉例來說，一個無符號八位元的數字可表示 0~255，實驗中我們以 16 為一個區間間隔，可把像素值轉換至 0~15 的索引值，假若有一個像素在 RGB 色彩空間的像素值為 $(255, 31, 10)$，則它的索引值為 $(15, 1, 0)$，把這索引值看作是 16 進位的數值即為 $F10_{16}$，因此這個像素的像素值代表 3856，再調整範圍從 1 開始，則此像素最終得到的像素值為 3857，最後再統計所有像素值的個數即為三維色彩直方圖，圖 3-4 為三維色彩直方圖的範例，圖 3-4(a)為原始圖，圖 3-4(b)為(a)圖的三維色彩直方圖。

(a) (b)

圖 3-4 三維色彩直方圖範例

(a)為原始， (b)為(a)之三維色彩直方圖

色彩直方圖與三維色彩直方圖，由於在直方圖的計算方式不同，若 v 為區間個數，前者產生 $1\times 3v$ 的直方圖向量，後者則是 $1\times v^3$ 的直方圖向量，兩者之間前者可快速且粗略地以直方圖向量描述畫面，後者則是可區分兩個些微差距的畫面，因此在使用上，場景的轉換由於變化較大且容易區分，因此我們使用粗略描述畫面的色彩直方圖即可，若使用三維色彩直方圖，則當人物走動，它將會很敏感的反映出來，反而會造成困擾；而當在處理臉部或是身體影像時，由於膚色佔據影像的大部分，因此需要更細膩地區分兩張圖，故選擇三維色彩直方圖來處理。

表 3-1　演員串列之臉部以及身體的相異度

串列影像	與前一$\boldsymbol{v}_F$的臉部相異度	與前一$\boldsymbol{v}_B$的身體相異度
1		
2	490	124
3	647	162
4	3517	3375
5	617	545
6	936	387
7	474	151

範例中的演員串列夾雜著兩個演員的影像，觀察表中串列影像 4 與串列影像 3 的$\boldsymbol{v}_F$相異度為 3517，明顯大於臉部相異度門檻值 2000，而且$\boldsymbol{v}_B$相異度也超過身體相異度門檻值 2500，因此藉由分割影像串列，可以使在串列中前後相異度較大的兩張影像分割成為兩個子串列。

3.4　篩選演員串列及建立重疊資訊

要建立演員串列時，必須經過一系列條件的篩選，以下 3.5.1~3.5.4 就是在不同階段所考量的篩選條件。

3.4.1　膚色篩選演員串列

為了減少 openCV 偵測臉部的錯誤，因此在分割完演員串列之後，利用洪詩祐[24]所提出的計算膚色的公式(5)，以及不同人種的參數，對演員串列作進一步的篩選。洪詩祐[24]提出的方法，先把影像的色彩空間轉換成 YCbCr，並利用 Cb 與 Cr 的數值計算膚色的機率 P，$\vec{x}$ 代表受測像素的 Cb 與 Cr，即為 $[\text{Cb}\ ,\ \text{Cr}]^T$，$\vec{\mu}$ 與 Σ 分別為 $\vec{x}$ 的平均值以及共變異矩陣。

$$P = \exp\left(-\frac{1}{2}(\vec{x}-\vec{\mu})^T \Sigma^{-1}(\vec{x}-\vec{\mu})\right) \tag{5}$$

$$\vec{\mu} = \begin{bmatrix} 113.17 \\ 149.03 \end{bmatrix},\ \delta = \begin{bmatrix} 44.98 & -31.01 \\ -31.01 & 46.60 \end{bmatrix}$$

受測像素的膚色機率 P 若是超過門檻值，即視為膚色像素，實驗中我們使用白種人的參數，並且設定膚色機率的門檻值為 0.1，統計整個$\boldsymbol{v}_{\mathrm{F}}$被視為膚色像素的比例，並移除小於 50%比例的$\boldsymbol{v}_{\mathrm{F}}$，假若串列中沒有任何一張臉部影像的膚色比例大於50%，極有可能是openCV的偵測錯誤，因此這個演員串列將被刪除。

3.4.2 色彩空間篩選演員串列

經過串列分割以及膚色篩選後，儘管被保留的$\boldsymbol{v}_{\mathrm{F}}$都符合篩選條件，但為了後續要計算$\boldsymbol{v}_{\mathrm{F}}$的相似度，我們不希望$\boldsymbol{v}_{\mathrm{F}}$的像素值標準差太大，主要原因是，當$\boldsymbol{v}_{\mathrm{F}}$的像素值標準差很大時，極有可能包含很多的背景資訊，但是膚色像素的比例又符合系統要求，這樣的$\boldsymbol{v}_{\mathrm{F1}}$與一個完全都是臉部資訊的$\boldsymbol{v}_{\mathrm{F2}}$計算相似度時，$\boldsymbol{v}_{\mathrm{F1}}$的背景資訊像素值將會影響描述兩者之間相似度的精確性，因此在這一步驟我們使用$\boldsymbol{v}_{\mathrm{F}}$的 RGB 色彩空間的標準差作為篩選限制。

首先，單一$\boldsymbol{v}_{\mathrm{F}}$的 RGB 分量標準差分別為$\delta_r$、$\delta_g$、$\delta_b$，而所有$\delta_r$、$\delta_g$、$\delta_b$之平均值及標準差分別為$\bar{\delta}_r$、$\bar{\delta}_g$、$\bar{\delta}_b$，與$std(\delta_r)$、$std(\delta_g)$、$std(\delta_b)$，我們以$\left[\bar{\delta}_r \quad \bar{\delta}_g \quad \bar{\delta}_b\right]$為基準把$S_E$、$\delta_g$、$\delta_b$的合理範圍界定在平均值的正負兩倍標準差範圍內，若$\boldsymbol{v}_{\mathrm{F}}$的$\delta_r$、$\delta_g$、$\delta_b$〝都〞超過該分量的合理範圍，則此$\boldsymbol{v}_{\mathrm{F}}$不合格；演員串列中若合格的$\boldsymbol{v}_{\mathrm{F}}$小於三個，則此串列將被刪除，代表此串列所包含的$\boldsymbol{v}_{\mathrm{F}}$與其他所有串列的$\boldsymbol{v}_{\mathrm{F}}$相差甚大，若把此演員串列保留下來，除了極有可能把此演員串列錯誤分群外，也有可能會造成其他演員串列錯誤分群的連鎖效應。以下是利用 RGB 分量標準差篩選演員串列的範例，圖 3-5 為兩個不同的演員串列，表 3-2 為每個$\boldsymbol{v}_{\mathrm{F}}$的 RGB 分量標準差，表 3-3 為兩個演員串列最後的篩選結果，由表 3-3 可得知，此步驟確實有效地篩選出擁有過多背景影像但膚色像素比例又符合系統預期的演員串列。

(a)

(b)

圖 3-5　兩個演員串列，(a)串列 1，(b)　串列 2

首先，先計算每個影像的 RGB 分量的標準差

表 3-2　RGB 分量標準差

υ_F	δ_r	δ_g	δ_b	是否合格
υ_{F11}	38.6	26	22.8	是
υ_{F12}	37.6	25.5	22.2	是
υ_{F13}	37.3	25.3	22.1	是
υ_{F14}	37.7	25.8	22.9	是
υ_{F15}	38.3	26.2	22.7	是
υ_{F21}	35.8	27.4	27.2	否
υ_{F22}	34.6	26.7	26.8	否
υ_{F23}	34.5	26.7	26.7	否
υ_{F24}	33.9	26.3	26.1	否
υ_{F25}	34.3	26.2	26.1	否

假設所有影像的δ_r、δ_g、δ_b之平均值與標準差為

$$\begin{bmatrix} \bar{\delta}_r & \bar{\delta}_g & \bar{\delta}_b \end{bmatrix} = \begin{bmatrix} 37 & 25 & 22 \end{bmatrix}$$
$$\begin{bmatrix} std(\delta_r) & std(\delta_g) & std(\delta_b) \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 & 2.0 \end{bmatrix}$$

則 RGB 標準差的合理範圍是$\begin{bmatrix} 36 \sim 38 & 24 \sim 26 & 18 \sim 26 \end{bmatrix}$

表 3-3　串列篩選結果

	演員串列	影像合格數	是否保留
串列 1		5	是
串列 2		1	否

3.4.3　演員串列的重疊資訊

當建立好演原串列後，一般而言，若兩個串列所包含的臉部影像在時間軸上有重疊，表示兩個串列是以不同的演員所構成，我們稱為〝重疊〞(overlap)，利用此資訊我們可以避免不同的演員被分在同一群的情況，規則為公式(6)。

$$\mathrm{OV}(i,j)=\begin{cases}1, \text{ if } I_i \text{ and } I_j \text{ are overlap}\\ 0, \text{otherwise}\end{cases} \tag{6}$$

3.5　人物相似度

實驗中，為了使分群擁有更多的資訊，我們不僅使用臉部影像，也加上身體影像作為輔助，而這兩個資訊的相似度矩陣我們以權重值的方式結合成為人物相似度矩陣。3.3 提到臉部相異度被定義為，兩個$\boldsymbol{v}_{\mathrm{f}}$向量差距之歐氏距離，而身體相異度則是兩個$\boldsymbol{v}_{\mathrm{B}}$的三維色彩直方圖向量差距之歐氏距離，擁有這兩者之後，我們可以計算影像間的臉部影像相似度s_P，以及身體影像相似度s_B，利用兩個影像的時間差距Δt來評估s_B的權重ω_B，最後帶入公式(7)就可產生兩影像間的人物相似度s_P，作為叢集整合的基礎。

$$s_P(i,j)=\left(1-\omega_B(i,j)\right)\cdot s_F(i,j)+\omega_B(i,j)\cdot s_B(i,j)\cdot r_B(i,j) \tag{7}$$

3.5.1 臉部相似度

我們以兩個$\boldsymbol{v}_{\mathrm{f}}$差距之歐氏距離作為臉部相似度$d_F$，而兩個演員串列間的臉部相異度我們取當中最小的數值代表，δ_F為所有d_F的標準差，利用公式(8)我們把臉部相異度d_F轉換成臉部相似度s_F。

$$s_F(i,j)=\exp\left(-\frac{{d_F}^2(i,j)}{2{\delta_F}^2}\right) \tag{8}$$

3.5.2 身體相似度

臉部影像對於演員串列的分群是最重要的，而身體影像也可以輔助分群，兩個影像間的身體相異度我們定義為，兩個$\boldsymbol{v}_{\mathrm{B}}$的三維色彩直方圖向量差距之歐氏距離$d_B$，同樣地，兩個演員串列間的身體相異度我們取當中最小的數值代表，δ_B為所有d_B的標準差，利用公式(9)我們把身體相異度d_B轉換成身體相似度s_B。

$$s_B(i,j)=\exp\left(-\frac{{d_B}^2(i,j)}{2{\delta_B}^2}\right) \tag{9}$$

3.5.3 身體影像之權重值

儘管影片裡的演員會隨著劇情而變更服裝內容，但在時間上，若兩個影像的時間差距很短時，兩個影像中的演員很有機會是同一人，而他們的衣服也很有可能是一樣的，因此身體影像的權重值ω_B我們參考了兩個影像的時間差距Δt，若是兩個串列，則以所有影像產生的平均時間作為該串列的生成時間，另外還有兩個可以控制身體權重值變化的參數h、σ，帶入公式(10)即可得到ω_B，實驗中設定 h 為 0.2，σ 為 1500。

$$\omega_B(i,j)=h\cdot\exp\left(-\frac{\Delta t^2(i,j)}{2\sigma^2}\right) \tag{10}$$

3.5.4 身體權重的參考值

實驗中基於〝在相近的時間內同一演員極有可能穿著相同的服裝〞的觀點，我們納入身體影像輔助分群，並且在時間差距 Δt 愈大時，參考身體影像的權重值就愈小。在實驗中可能會加入不同的測試影片，因此若兩個是來自不同影片的影像，則身體影像的相似度就不需要也不能參考，因此參考值 r_B 就被定義如公式(11)，I_i 與 I_j 代表兩個不同影像。

$$r_B(i,j)=\begin{cases}1, \text{ if } I_i \text{ and } I_j \text{ in the same movie}\\ 0, \text{otherwise}\end{cases} \tag{11}$$

第四章　實驗方法

第三章基本上是仿照[21]的作法，主要目的是利用一些客觀的條件產生演員串列以及人物相似度矩陣 S_P，這一章節則是說明如何運用這些已建立的演員串列產生最終分群的結果，實驗中利用叢集整合的概念設計實驗。首先，在 4.1 節說明利用 PCA 轉換，產生新的向量，並說明如何產生演員串列的領導臉，4.2 節則說明如何產生叢集整合相似度矩陣，以及說明如何把與叢集整合相似度矩陣與人物相似度矩陣合併，4.3 節介紹實驗中所使用的四種凝聚演算法，最後，在 4.4 節介紹實驗中選用哪些參數。

4.1　PCA 轉換及產生演員串列的領導臉（leader face）

在 3.3 節中已介紹過利用所有 openCV 偵測到的臉部影像，以 PCA 的方法把 1×4900 的 $\boldsymbol{v}_{\mathrm{F}}$ 降維至 1×70 的 $\boldsymbol{v}_{\mathrm{f}}$ 作演員串列的篩選；然而，現在我們已經完成所有演員串列的篩選，演員串列裡的影像成員早已大不相同，因此我們重新把目前被保留在演員串列內的所有 $\boldsymbol{v}_{\mathrm{F}}$ 依照 3.3 節的作法，取出前 70 大特徵值所對應的新的臉部降維矩陣為 B'_{F}，同樣地，$\boldsymbol{v}_{\mathrm{F}}$ 以公式(12)也被降維到新的 1×70 向量 $\boldsymbol{v}'_{\mathrm{f}}$，之後臉部影像的運算都從降維過後的 $\boldsymbol{v}'_{\mathrm{f}}$ 為基礎。

$$\boldsymbol{v}_{\mathrm{F}}\,\mathrm{B}'_{\mathrm{F}} = \boldsymbol{v}'_{\mathrm{f}} \tag{12}$$

[11]說明一套在眾多物件（object）中選 leader 的方法，[4]也提到，可在演員串列當中選擇關鍵臉（keyface），以關鍵臉作為往後處理串列運算的基礎.。在實驗中，我們也使用[4]的相同概念，但是選用臉部影像的條件與[4]不相同，因此在文中我們選出來的臉部影像稱為領導臉（leader face），主要目的是想藉由減少串列中的臉部影像換取時間成本，並試著用最少的領導臉使效能達到可接受的範圍內，我們稱這些串列中屬於領導臉的臉部影像集合為領導臉串列，往後只

利用領導臉串列的成員處理該串列的運算。

選擇領導臉的作法，首先，先計算演員串列內$\boldsymbol{v}'_{\mathrm{f}}$對$\boldsymbol{v}'_{\mathrm{f}}$的距離矩陣$D$，用公式(13)式把範圍平移縮放至 01 之間接著運用D'在階層式凝聚演算法上，並設定一個 0 到 1 的閥值θ_{dyn}，凝聚過程中假若即將被合併的兩群之間的距離超過θ_{dyn}，則停止凝聚，這樣一來我們可以得到對演員串列內的所有$\boldsymbol{v}'_{\mathrm{f}}$的分群結果，之後在每一群中，挑出一個〝與群內其他的$\boldsymbol{v}'_{\mathrm{f}}$差距總和最小的$\boldsymbol{v}'_{\mathrm{f}}$〞作為該群的領導臉。

$$D' = \frac{D - \min(D)}{\max(D) - \min(D)} \tag{13}$$

θ_{dyn}的選用將會影響領導臉的產生以及領導臉的個數，當θ_{dyn}為 0，沒有一個$\boldsymbol{v}'_{\mathrm{f}}$被合併，形成各自一群的情況，因此演員串列裡的每個$\boldsymbol{v}'_{\mathrm{f}}$都被選為領導臉，我們稱此情形為「all face」；而θ_{dyn}為 1 時，所有$\boldsymbol{v}'_{\mathrm{f}}$都被合併為一群，因此以〝與群內其他的$\boldsymbol{v}'_{\mathrm{f}}$差距總和最小的$\boldsymbol{v}'_{\mathrm{f}}$〞當作領導臉，我們稱此情形為「one face」；all face 與 one face 是兩種極端的情形，之間還有其他的可能性，因此我們另外也嘗試用其他 0 與 1 之間的θ_{dyn}來調整領導臉的個數。

4.2 產生叢集整合相似度矩陣及合併相似度矩陣

在上述 3.5 節中，串列產生的人物相似度是利用串列本身的時間特性、臉部特徵以及身體特徵來計算串列間的相似度，除了利用這些客觀的條件外，我們也試著讓所有領導臉串列進行分群，利用分群的結果了解串列之間的相似度。實驗中我們以串列為單位，採用 k-medoid 分群法，而 k 的數值是從一個固定的範圍隨機挑選，每個範圍都作 100 次取平均的，以下是 k-medoid 的虛擬碼（pseudo code）：

k - medoid：

step1. Initialization

$Q = \{ q_1, q_2, \ldots, q_r \}$, r sequences

$C = \{ c_1, c_2, \ldots, c_k \}$, assign k random sequences to k centers

$L = \{ l_1, l_2, \ldots, l_r \}$, label of r sequences

$L' = \phi$

step2. Compute label： $l_i = \arg\min_{j=1\sim k} d(q_i, c_j)$, $\forall i = 1 \sim r$

step3. Update center： $c_t = \arg\min_{q_i} \sum_{j \neq i, l_j = l_i} d(q_i, q_j)$, $\forall t = 1 \sim k$, $1 \le i, j \le r$

step4. Repeat

IF $(L' \neq L)$

$L' = L$;

repeat step2 and step3 ;

ELSE

stop ;

END

k-medoid 與 k-means 的演算法過程相當類似，兩者的主要差別是在於更新群中心的方式不同，k-means 是把群內所有串列的平均值作為群中心；而 k-medoid 則是以〝與群內其他串列差距總和最小的串列〞當作群中心。在計算兩個領導臉串列的距離時，由於串列內的$\boldsymbol{v}'_{\mathrm{f}}$是原先演員串列分群的各群領導臉，因此單一$\boldsymbol{v}'_{\mathrm{f}}$不太可能與另一個串列裡的全部領導臉都很相近，所以兩的領導臉串列之間我們只在乎他們差距最小的那幾個配對組合的距離。

我們以歐氏距離計算兩個$\boldsymbol{v}'_{\mathrm{f}}$的差距，實驗中設定 minimum k-top 的 k 值為 10，若兩個演員串列的$\boldsymbol{v}'_{\mathrm{f}}$配對數小於 k，則會以全部配對的平均距離作為兩個串列的距離，以下是計算兩個串列間距離的虛擬碼，兩個演員串列q_i與q_j分別擁有個 g 和 h 個領導臉。

$$q_i = \{LF_{i1}, LF_{i2}, \ldots, LF_{ig}\}, \quad q_j = \{LF_{j1}, LF_{j2}, \ldots, LF_{jh}\}$$

$$d(q_i, q_j) = \frac{1}{k}\left(\text{minimum k- top of } d(LF_{is}, LF_{jt})\right), \forall s = 1 \sim g, \forall t = 1 \sim h$$

$$d(LF_1, LF_2) = \text{Euclidean distance of } LF_1 \text{ and } LF_2$$

為了方便觀察，圖 4-1 我們使用被調整至 70×70 大小的 openCV 偵測臉部影像，圖 4-1 是一個計算 minimum k-top 的範例，圖 4-1(a)與(b)為兩個演員串列，串列 1 的臉部有稍微轉動，串列 2 的臉部也是轉動中，但只轉到正面，速度比串列 1 更快。若是在兩個串列的臉對臉的距離矩陣中取 minimum 3-top 平均，則所取出的配對結果為圖 4-1(c)，相較於其他配對組合，此三組配對確實較容易被辨認為同一個演員。實際上，兩個串列的 minimum 3-top 平均為 5.78，反觀，若是計算所有臉對臉的平均距離則為 8.09，上升幅度高達 39.9%，如此高的比例，極有可能會導致這兩個串列被錯分開來，因此，比起計算兩個串列的所有臉部影像配對的平均距離，minimum k-top 反而更容易看出兩個串列間的關係。

(a) 串列 1

(b) 串列 2 (c) 配對結果

圖 4-1 兩串列的 minimum 3-top

選擇使用 k-medoid 而非 k-means 的原因，是由於使用 k-means 會牽涉到更新群中心時，是以串列為單位，抑或是以串列內的領導臉為單位，但是在產生領導串列時，所作的 PCA 和選擇領導臉時，都已經把原先的串列資料簡化了，並且我們執行 k-medoid 或是 k-means 只是為了產生串列間的相似度矩陣，因此不必計算到如此細微，而且 k-medoid 的中心更新方式是挑選〝與群內其他串列差距

總和最小的串列″當作群中心，這樣的群中心與 k-means 的群中心兩者誤差仍在可接受的範圍內。

我們使用 4.1 所產生的領導臉串列執行 k-medoid，而群數 k 則是在某個固定範圍的隨機數，經由隨機 k 值得到的分割（partition）可以幫助我們了解串列之間的相關程度，在這裡我們以 P_r 作為第 r 個分割的標記，定義兩個串列在同一個分割裡的相似度是以共相關（co-association）[7]的計算方式，統計$\boldsymbol{v}'_f$間在 P_r 的叢集整合相似度為 $s_E^{(r)}$，另外計算總共 d 個分割的$\boldsymbol{v}'_f$間平均相似度為 s_E^*，如公式(14)。

$$s_E^{(r)}(i,j)=\begin{cases}1, \text{ if } q_i \text{ and } q_j \text{ in the same cluster in a partition}\\ 0, \text{otherwise}\end{cases}$$
$$s_E^*(i,j)=\sum_r^d s_E^{(r)}(i,j) \tag{14}$$

實驗時我們考量3.5節的人物相似度，s_P，以及本節敘述的叢集整合相似度，s_E，兩者以加權的方式結合，θ_ω 為叢集整合的權重值，最終形成我們使用的相似度 s，定義如公式(15)，s 不僅僅包含各演員串列的客觀條件(s_P)，也包含一些較細膩的領導臉串列的分群結果(s_E)，θ_ω 是重要的參數，以最好結果來觀察 θ_ω，可瞭解 s_P 與 s_E 的重要關係。

$$s(i,j)=(1-\theta_\omega)\cdot s_P(i,j)+\theta_\omega \cdot s_E(i,j) \tag{15}$$

4.3 凝聚演算法

實驗中並非每個領導臉串列都能找到與它很類似的串列，可能有些串列與大部分的串列相差甚遠，例如 openCV 可能把很類似膚色的背景影像當作演員串列，最終形成領導臉串列；因此，對所有領導臉串列分群時，第一階段我們可以先凝

聚相似度很高的大部分串列，第二階段才把剩下的串列以 1NN（1 nearest neighbor）的方式歸類，如此可避免過多的干擾（outlier）影響分群的結果，這種方法在本文稱做為〝2step〞。另外，我們也試著把重疊資訊加入演算法中，所以實驗中我們嘗試了四種組合的凝聚演算法：

4.3.1 弱凝聚法（Weak Agglomeration）

我們利用 S 以及 3.4.3 的重疊資訊矩陣的否定，$\overline{\mathrm{OV}}$，組成弱相似度矩陣，S_W，S_W 的產生如公式(16)

$$S_W = S \times \overline{\mathrm{OV}} \tag{16}$$

4.3.2 強凝聚法（Strong Agglomeration）

執行凝聚法時，假若目前步驟將要合併的兩個串列 q_i 與 q_j 產生重疊，即 $OV(i,j)=1$，則兩個串列將不允許被合併，直接合併相似度第二高的串列對（sequence pair），之後也不再合併 q_i 與 q_j。

4.3.3 兩階段弱凝聚法（2step Weak Agglomeration）

假設第一階段要求凝聚 θ_{rat} 比例的領導臉串列，則第一步驟是取出〝與其他串列差距總和為前 θ_{rat} 比例小的串列〞執行弱凝聚法，第二步驟才把剩下的串列以 1NN 的方式歸類。

4.3.4 兩階段強凝聚法（2step Strong Agglomeration）

假設第一階段要求凝聚 θ_{rat} 比例的領導臉串列，則第一步驟是取出〝與其他串列差距總和為前 θ_{rat} 比例小的串列〞執行強凝聚法，第二步驟才把剩下的串列以 1NN 的方式歸類。

弱凝聚法與強凝聚法的差別在於，當凝聚到很接近設定的最終群數時 C，前者仍有可能把兩個有重疊的領導臉串列合併；而後者則永遠不會，但是後者卻有可能因為剩下的任兩個串列都產生重疊而停止在非設定的群數上，如此將無法達到我們要求的群數。

4.4 叢集整合參數選用

實驗的後半段主要是結合叢集整合的方法得到不同的分割（partition），來測試人物分群的效能，因此在叢集整合的實驗中我們選定六個參數很有可能影響結果的參數進行實驗，六種參數共產生上萬筆數據，這樣的變化讓我們擁有充滿多樣性的分群結果，可提升叢集整合的分群效能，以下為參數的詳細介紹：

4.4.1 領導臉－θ_{dyn}

在 4.1 節就有提到 θ_{dyn} 會直接影響領導臉的選取以及個數，實驗中我們測試了 0、0.1、0.2、…、1.0，共 11 個數值。

4.4.2 k-medoid 隨機範圍－θ_{rg}

k 值是從一個固定範圍隨機挑選，它的變化會影響演員串列選領導臉的分群結果，間接地影響領導臉的選取以及個數，[7]中提到若由較大的範圍裡隨機挑選 k 值，所得到的叢集整合效果會優於小範圍，因此我們設計五個隨機挑選的範圍觀察其效能的變化，五個範圍是 2~10、2~20、2~40、2~80、2~160。

4.4.3 叢集整合相似度矩陣之權重值－θ_w

θ_ω 的改變直接影響了最終的相似度矩陣 S，我們以 0、0.1、0.2、…、1.0，共 11 個數值去測試 S 與效能的關係，θ_ω 為 0 時，代表僅由人物相似度矩陣執行凝聚法；反之，θ_ω 為 1 則由叢集相似度矩陣執行凝聚法，藉由效能的變化可以

看出叢集整合對於效能的影響與重要性。

4.4.4 凝聚法－θ_{agg}

依照 4.3 介紹的四種凝聚法，我們分別把弱凝聚法、強凝聚法、兩階段弱凝聚法、兩階段強凝聚法，以 θ_{agg} 用 1~4 來表示。

4.4.5 兩階段凝聚法比例－θ_{rat}

θ_{rat} 為兩階段凝聚法中，第一階段凝聚串列的比例，領導臉串列是經由演員串列降維並且挑出其中較具代表性的領導臉所組成，而演員串列的產生都是經過一系列的篩選，因此並不會有太多太雜亂的的演員串列，領導臉串列也是如此，因此若選用較低的 θ_{rat} 數值反而會出現反效果，因此實驗中設定 θ_{rat} 為 0.7、0.75、0.8、0.85、0.9、0.95 來觀察效能，另外，θ_{agg} 為 1 與 3 與同樣都是弱凝聚法，兩者只差在第一步的凝聚比例罷了，而 θ_{agg} 為 1 即是 θ_{agg} 為 3 且 θ_{rat} 為 1.0 的特殊案例（special case）；同理，θ_{agg} 為 2 即是 θ_{agg} 為 4 且 θ_{rat} 為 1.0 的特殊案例。

4.4.6 凝聚群數－C

我們測試的群數，只針對四個數值作測試，影片真正的演員個數 α、10、20、30，α 是影片中主要演員的個數，每個影片中的數值都不盡相同，假若把兩個測試影片放入一起執行，則影片真正的 α 會選擇兩個 α 中的較大值，作為最終凝聚 α 群的結果。

第五章 實驗結果

在這一章節中，5.1 節介紹實驗所使用的資料集，5.2 節介紹四種的計算分群結果的公式，5.3 至 5.7 針對單一變數的參數值效能分析與討論，最後在 5.8 比較數據。

5.1 資料集

我們使用的資料集為美國影集，六人行（Friends），影片內的主要的演員有六人，但是會依照各集的劇情加入客串演員，我們使用的測試影片有三集，第三季第九集、第三季第二集、第七季第一集，我們以 M1、M2、M3 表示，另外也使用兩兩合併的影集，測試系統在多影片的輸入下是否能得到同樣的效能，藉此推測系統是否適用在多影片的處理，合併影片M1+M2、M1+M3 以及 M2+M3 我們以 M4~M6 來表示，其中第三季第九集即是[21]中的測試影片，為了與[21]比較，因此實驗時我們同樣地把測試影片的片頭刪除，表 5-1 是測試影片刪除片頭後的影片特性：

表 5-1 測試影片的特性

影片	片長	影像數	場景數	演員串列數	全部串列的人臉總數	串列中最多 / 最少人臉數
M1	21:44	6520	479	504	4096	53 / 3
M2	21:44	6520	369	453	3739	55 / 3
M3	21:10	6350	411	462	4594	72 / 3
M4	43:28	13040	848	946	7765	55 / 3
M5	42:54	12870	890	969	8686	72 / 3
M6	42:54	12870	780	907	8249	72 / 3

由於合併影片時 PCA 的臉部降維矩陣B_F'已改變，因此 M1~M3 中與 M4~M6 中的的同一張臉部影像經過降維後的$\boldsymbol{v}_f'$也會有所不同，所以在演員串列數、全部串列的人臉總數都並非原先兩個測試影片的總和。

5.2 效能計算

為了展示叢集整合的結果，本文中利用下列四個數值來觀察，假設 $O=\{o_1,o_2,o_3,...,o_n\}$ 代表測試資料集，共包含 n 個物件；$U=\{u_1,u_2,u_3,\cdots,u_R\}$ 與 $V=\{v_1,v_2,v_3,\cdots,v_T\}$ 代表兩個測試樣本的分群結果，群數分別為 R 與 T，並且 $\bigcup_{i=1}^{R} u_i = O = \bigcup_{j=1}^{T} v_j$，$u_i \cap u_{i'} = \phi = v_j \cap v_{j'}$，$1 \le i, i' \le R$，$1 \le j, j' \le T$，把每一個物件從 U 對應至 V 的分群結果作一個統計表格，就可得到表 5-2。

表 5-2 U 對應 V 統計表

U \ V	v_1	v_2	$\cdots$	v_C	$\sum_j n_{ij}$
u_1	n_{11}	n_{12}	$\cdots$	n_{1C}	$n_{1.}$
u_2	n_{21}	n_{22}	$\cdots$	n_{2C}	$n_{2.}$
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
u_R	n_{R1}	n_{R2}	$\cdots$	n_{RC}	$n_{R.}$
$\sum_i n_{ij}$	$n_{.1}$	$n_{.2}$	$\cdots$	$n_{.C}$	n

資料來源[19]

5.2.1 RI (Rand Index)

在物件的分群結果中，我們可以把所有對應關係分成四種對應組合[19]：

I. 在 U 被分為同一群且在 V 也被分為同一群的物件，其對（pair）數為

$$a = \sum_{i,j} \binom{n_{ij}}{2}$$

II. 在 U 被分為同一群但在 V 卻被分為不同群的物件，其對 (pair)數為

$$b = \sum_i \binom{n_{i\cdot}}{2} - \sum_{i,j} \binom{n_{ij}}{2} = \sum_i \binom{n_{i\cdot}}{2} - a$$

III. 在 U 被分為不同群但在 V 卻被分為同一群的物件，其對 (pair)數為

$$c = \sum_j \binom{n_{\cdot j}}{2} - \sum_{i,j} \binom{n_{ij}}{2} = \sum_j \binom{n_{\cdot j}}{2} - a$$

IV. 在 U 被分為不同群且在 V 也被分為不同群的物件，其對 (pair)數為

$$d = \binom{n}{2} - a - b - c$$

由於 a 與 d 是兩個分群結果中共同認同的部分，因此 RI 的計算是如公式(17) RI 的值介於 0 與 1 之間，當 RI 為 1 即代表兩個分群結果完全相同。

$$RI(U,V) = \frac{a+d}{a+b+c+d} = (a+d) \Big/ \binom{n}{2} \tag{17}$$

5.2.2 ARI (Adjusted Rand Index)

[20]提到 ARI 的計算方式是從 RI 衍伸而來，由於計算兩個隨機的分群結果其 RI 的期望值，*expected index* 並非為 0，因此 Hubert 和 Arabie 提出 ARI，希望把期望值調整至 0，利用公式(18)把原本 RI 的數值 *index* 平移與縮放。

$$\frac{index - expected\ index}{maximum\ index - expected\ index} \tag{18}$$

經由推導，ARI 的公式如公式(19)

I. 在 U 被分為同一群且在 V 也被分為同一群的物件，其對 (pair)數為

$$a = \sum_{i,j} \binom{n_{ij}}{2}$$

II. 在 U 被分為同一群組的物件，其對 (pair)數為 $d = \sum_i \binom{n_{i\cdot}}{2}$

III. 在 V 被分為同一群組的物件，其對 (pair)數為 $e = \sum_j \binom{n_{\cdot j}}{2}$

$$ARI(U,V) = \frac{a - (d \times e) \Big/ \binom{n}{2}}{(d+e)/2 - (d \times e) \Big/ \binom{n}{2}} \tag{19}$$

為了讓讀者更明白這兩種公式的計算，以下有個簡單的範例，十個物件被分成$U=\{1,1,1,2,2,2,2,2,3,3\}$和$V=\{1,1,2,2,2,2,3,3,1,3\}$，經由統計可得到表 5-3。

表 5-3　分群範例統計表

U \ V	v_1	v_2	v_3	$\sum_j n_{ij}$
u_1	2	1	0	3
u_2	0	3	2	5
u_3	1	0	1	2
$\sum_i n_{ij}$	3	4	3	10

$$\text{RI}:\ \text{a}=\binom{2}{2}+\binom{3}{2}+\binom{2}{2}=1+3+1=5$$

$$\text{b}=\binom{3}{2}+\binom{5}{2}+\binom{2}{2}-5=3+10+1-5=9$$

$$\text{c}=\binom{3}{2}+\binom{4}{2}+\binom{3}{2}-5=3+6+3-5=7$$

$$\text{d}=\binom{10}{2}-5-9-7=45-21=24$$

$$\text{RI}(U,V)=\frac{5+24}{5+9+7+24}=\frac{29}{45}=64.44\%$$

$$\text{ARI}:\ \text{a}=\binom{2}{2}+\binom{3}{2}+\binom{2}{2}=1+3+1=5$$

$$\text{b}=\binom{3}{2}+\binom{5}{2}+\binom{2}{2}=3+10+1=14$$

$$\text{c}=\binom{3}{2}+\binom{4}{2}+\binom{3}{2}=3+6+3=12$$

$$\text{ARI}(U,V)=\frac{5-(14\times12)\Big/\binom{10}{2}}{(14+12)/2-(14\times12)\Big/\binom{10}{2}}=\frac{5-3.73}{13-3.73}=13.67\%$$

上述範例中可以看出，同樣的兩個分割 U 和 V，兩種數據的差距卻很大，實驗中我們希望能突顯兩個分割結果的差異，因此我們以 ARI 作為主要觀察效能的對象。4.6 節中提到，實驗中總共有六個變數，一個影片使用六個變數總共產生上萬筆數據，若直接拿來分析，恐怕不是那麼容易，因此我們在 5.3 到 5.7 小節中，每次都只針對一個參數的參數值來討論，討論的順序是依據趨勢的明顯度由大而小編排，亦即，愈先被討論的變數，代表愈容易看出它的規律性；反之，愈無規律性。

5.3 C 之效能

首先，針對最終凝聚的群數 C 來探討，我們固定領導臉的選擇方式 θ_{dyn} 為 0、θ_{rg} 的範圍是 2~80、θ_w 為 0.2、θ_{agg} 為兩階段強凝聚法、θ_{rat} 是 0.7，觀察圖 5-1 中六個影片(M1~M6)對於群數 C 之 ARI 曲線變化。圖中可發現，M1~M4 中以凝聚到 α (6 或 7)與 10 群的效果較 20、30 群來得好，這是由於 α 是代表每個影片中確切的演員個數，而 ARI 容易受群數影響，因此當群數接近 α 時，計算 ARI 將會有比較好的優勢，這是可以預見的，然而 M5 與 M6 從 C 為 10、20、30 看來，合併之後的效能確實被嚴重地影響了，比原來 M1~M3 的效能矮了一截，我認為這是因為合併影片之後把兩個影片原有的相似度打散了，串列變得更複雜了，才連最篤定〝愈靠近 α 群效能愈好〞的趨勢也都沒有顯現；[3]作者利用前後 11 年的影集作實驗，也提到在不同的影集當中，主角的面孔、頭髮也都會隨著年紀不太一樣，因此我認為，如果能把時間軸拉到每一集、每一季甚至每一年之外，就算是同一人在不同年紀所拍攝的影集，也可以調整影集之間相互參考的權重值，以這觀點來看，或許是很有機會可以提升效能的。

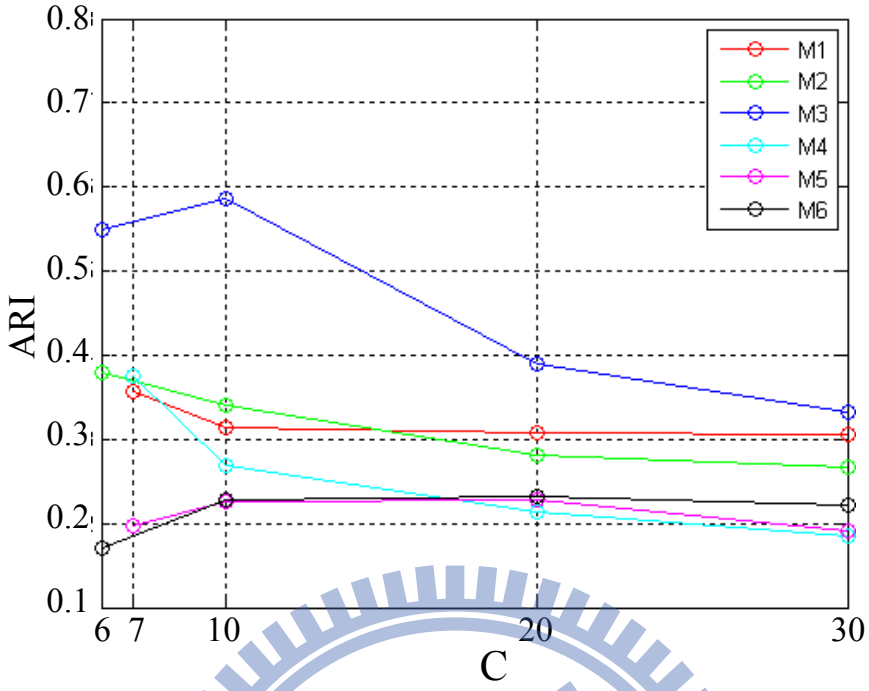


圖 5-1　M1~M6 對 C 的效能曲線

5.4　θ_{rg} 之效能

接下來我們針對 k-medoid 的範圍參數 θ_{rg} 來討論，下圖中是固定領導臉的選擇方式 θ_{dyn} 為 0.2、θ_{w} 為 0.4、θ_{agg} 為強凝聚法、θ_{rat} 是 1.0、θ_{c} 是 10，產生的結果，我們若只觀察 M1~M3 的曲線可以發現，愈後面的範圍效能愈好，這是由於在眾多的串列中，儘管是兩個相同人物的串列，也會因為臉部表情、臉部旋轉角度、影像明亮度和背景不同而有所差異，藉由執行較大範圍的 k-medoid，可得到兩個串列間較細膩的相似度關係，因此若在 θ_{rg} 不知從何選起，可設定較大範圍的是比較好的選擇，這個推論也和[7]文中所提到 k 值要選〝數值大以及範圍較廣的隨機範圍〞一樣，而 M4~M6 就如同 5.3 提到的，因為合併影片而降低了相似度，因此在圖 5-2 上看出趨勢。

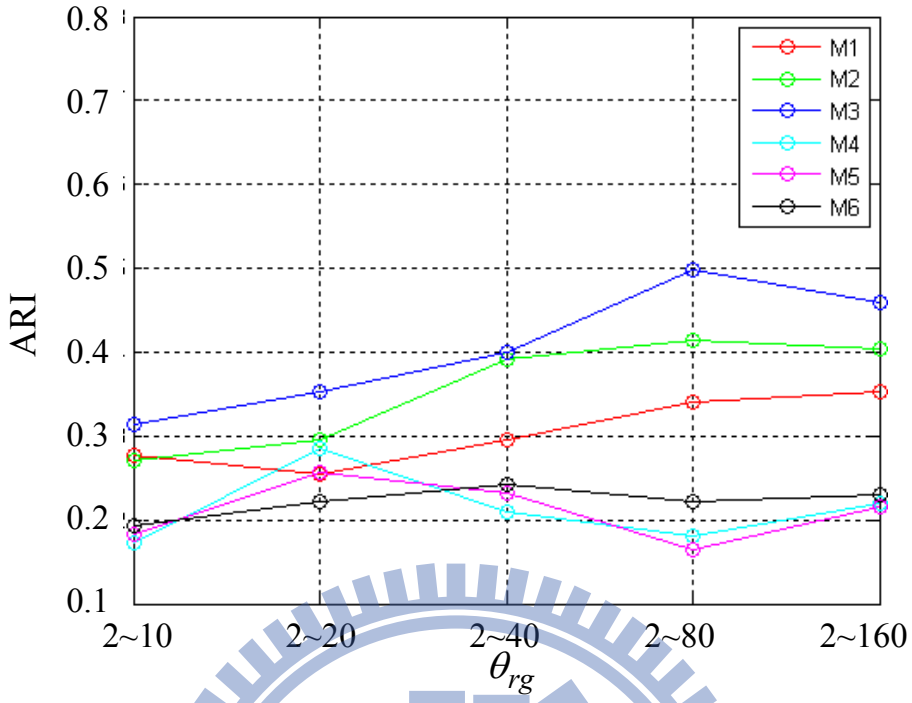
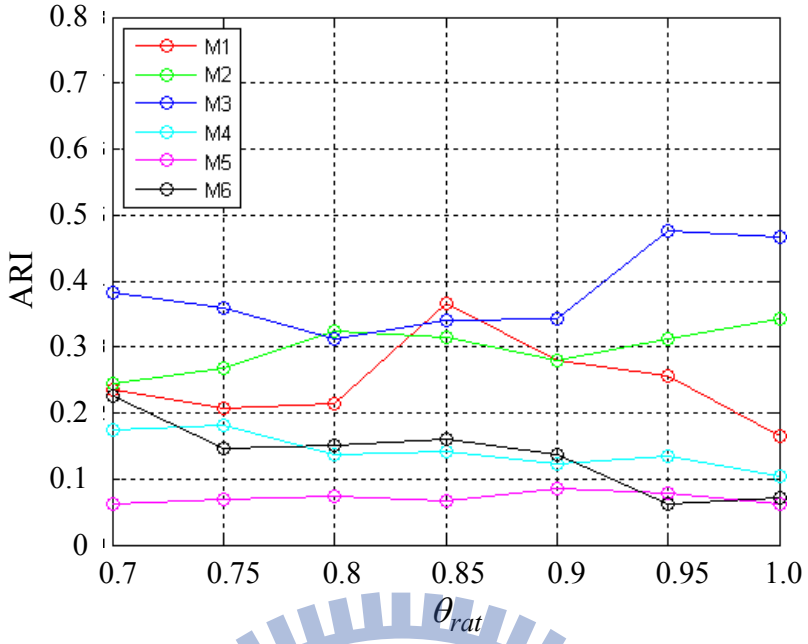


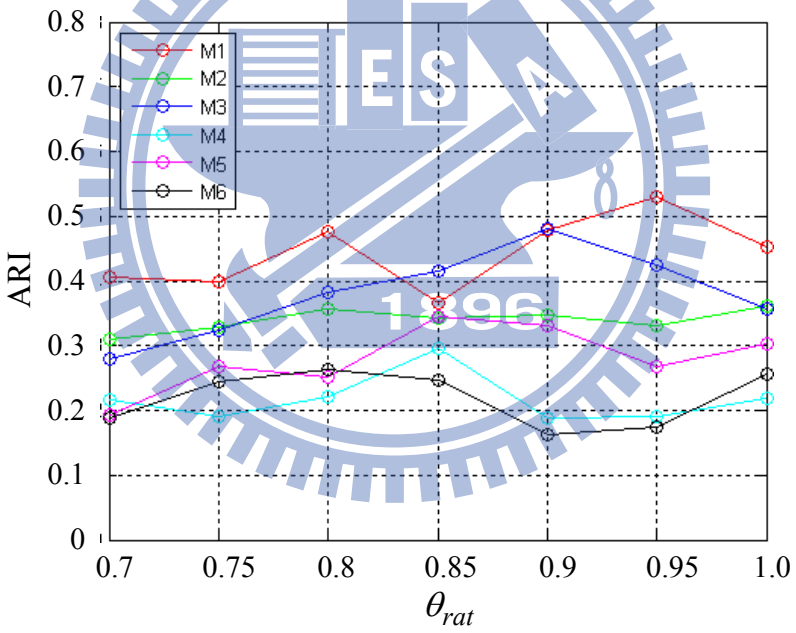
圖 5-2　M1~M6 對 θ_{rg} 的效能曲線

5.5　θ_{agg} 與 θ_{rat} 之效能

在這個小節中，我們討論四種 θ_{agg} 以及 θ_{rat} 對於 ARI 曲線的變化，θ_{agg} 為凝聚法的四種模式，θ_{rat} 為兩階段分群中首要階段分群的比例。本文 4.6.5 提到 θ_{agg} 為 1 與 2 分別是 θ_{agg} 為 3 與 4 在 θ_{rat} 為 1 的特殊案例，因此我們在比較兩階段強凝聚法時會把 θ_{agg} 為 1 的效能一起討論，同樣地，θ_{agg} 為 2 的效能也會與兩階段弱凝聚法一起討論，圖 5-3 是我們固定 θ_{dyn} 為 0.3、θ_{rg} 的範圍是 2~160、θ_{w} 為 0.1、θ_{c} 是 person，六個影片對於 ARI 的變化曲線，圖 5-3(a)是兩階段弱凝聚法的比較，圖 5-3(b)為兩階段強凝聚法的比較。



(a)



(b)

圖 5-3　　M1~M6 對 θ_{agg} 與 θ_{rat} 的效能曲線

(a)　兩階段弱凝聚法,(b)　兩階段強凝聚法

(a)(b)兩圖中相同顏色的曲線代表相同的影片，首先，觀察六組左右兩個相同顏色的曲線會發現，兩階段強凝聚法曲線的效能確實比弱凝聚法好，以 M1、M4、M5 以及 M6 特別明顯，平均效能也能領先一成左右，而這結論從我們設計

的限制條件就可找出端倪，由於強凝聚法硬性規定兩個發生時間軸重疊的串列永不合併，將可避免不合理的分群錯誤。

再來我們觀察使用兩階段強凝聚法的六個影片與θ_{rat}的關係，六條曲線儘管沒有太明顯的趨勢，但是以大方向來看，θ_{rat}在 0.85 到 1.05 之間時，平均效能比起 0.7 到 0.8 之間有稍微往上的趨勢，再者，若以曲線最高的範圍來說，六條曲線的最好效能皆落在 0.85 到 1.0 之間，自己另外觀察其他參數值組合時，大略也有此現象，然而這現象並非偶然，這是因為當初在設計θ_{rat}這個參數時，是考量到並不是所有串列都能找到與自己很相近的另一個串列而聚集在一起，因此我們才利用兩階段的分群法，使相似度比較高的大部分串列先分群，剩下的再依照 1NN 的方式歸類，因此θ_{rat}的選取數值也與產生的演員串列的息息相關。

假若只有極少數的串列是由非單一人物的影像所組成，就代表系統產生的演員串列純度極高，那麼θ_{rat}就可以大膽地挑選較高的數值甚至可直接執行強凝聚法；另外一個觀點是，θ_{rat}也跟演員串列的辨識度有關，若演員串列都有足夠的特徵使自己被分在對的群集裡，則θ_{rat}可挑選較的數值；反之，則應從較低的數值選起。除此之外，我們也可由較高效能的θ_{rat}值反推演員串列的品質好壞，觀察圖 5-3(b)，我們發現一個現象，M1 到 M3 的最高效能的位置都在 0.9 到 1.0 之間，而 M4 和 M5 卻都在 0.9 之下，M5 的最高效能也可被θ_{rat}為 0.8 所取代，M1~M3 θ_{rat}偏高而 M4~M6 的θ_{rat}較低，我們認為這是由於 M4~M6 是由多個影片合併的測試資料，因此影像特徵間的相似度並不如 M1~M3，所以產生較多的干擾 (outlier) ，也把最高效能的θ_{rat}直往下拉，避免直接分群造成效能降低。由此範例的最高θ_{rat}可以推論，此系統所產生的演員串列品質是不錯的，因為只要事先屏除一些 (0%~15%) 干擾的串列就可以使分群有不錯的效能。

5.6 θ_w 之效能

我們利用相似度矩陣 S 執行凝聚法決定最終的分群結果，在 S 中除了一般的

依照臉部、身體影像或是輔助的條件所建構的人物相似度矩陣 S_P 外，我們加入叢集整合相似度矩陣 S_E，想了解 S_E 對分群結果的幫助，而 W_E 為 S_E 的權重值，也代表 S_E 的重要性，因此 W_E 對 ARI 的數值曲線可以幫助我們了解 S_E 對分群結果的影響。圖 5-4 我們固定 θ_{dyn} 為 0.2、θ_{rg} 的範圍是 2~40、θ_{agg} 為強凝聚法、θ_{rat} 為 1.0、θ_c 是 person，觀察 W_E 對於 ARI 曲線的變化。

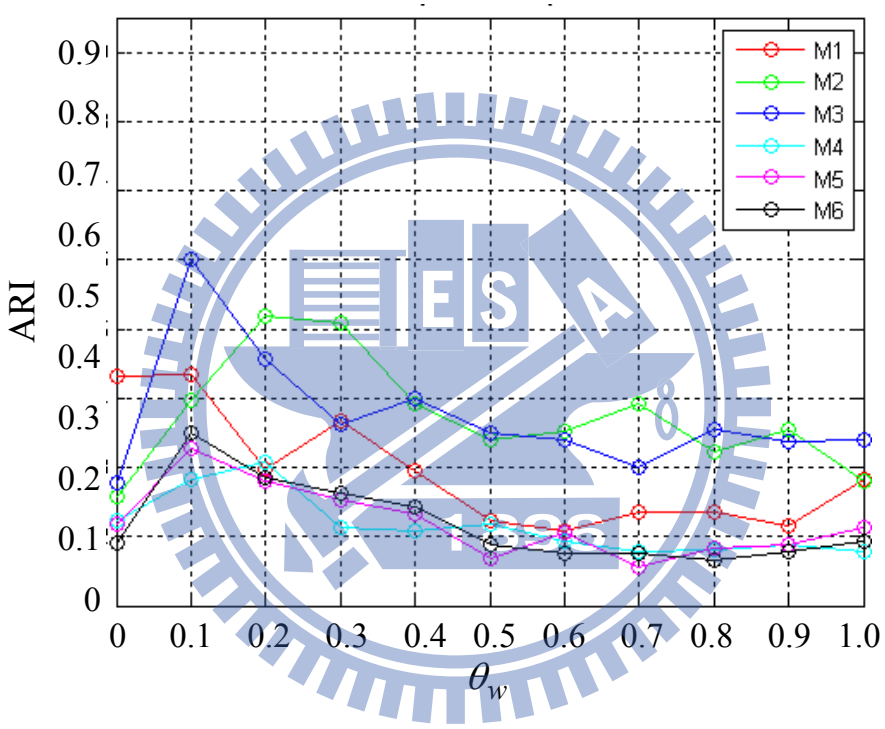


圖 5-4　M1~M6 對 θ_w 的效能曲線

實驗中 θ_w 一共挑選 11 個參數值，從 0 至 1 之間以 0.1 為間隔測試，當 θ_w 為 0，代表只以 S_P 執行凝聚演算法；反之，若 θ_w 為 1 則代表只以 S_E 執行凝聚演算法。首先，我們對六條曲線的效能走勢觀察，發現曲線從 0 出發至 θ_w 在 0.1 至 0.4 之間有明顯的提升，0.4 之後即開始走下坡，並開始小幅震盪，若以最高效能的 θ_w 值來觀察，M1~M6 都落在 0.1~0.3 之間，這代表加入 S_E 的最好效能確實能獲得提升。再者，我們觀察曲線的兩個端點，θ_w 為 0 與 1 的效能，在六條曲線

中互有領先，因此我們無法對於S_P與S_E哪一項的相似度資訊較正確的問題下定論，但觀察整體的曲線，M1 有 1 個θ_w的數值比θ_w為 0 的效能還要高，M2 有 10 個，M3 有 10 個，M4 有 2 個，M5 有 4 個，M2 也 4 有個，整體上看來加入S_E是有機會提升效能的，而挑選的原則是以0.1~0.3之間的θ_w為較好的選擇，因此，雖然我們無法得知S_P與S_E哪一項的相似度資訊較正確，但以最佳效能而言，S_P的影響力仍為S_E的三倍以上（θ_w以 0.3 計算），若再加強S_E的可靠度，效能提升的幅度勢必更可觀，因此在影像中處理人物分群時，我們並不能捨棄S_P只用S_E，然而S_E也會是個提升效能的重要輔助資訊，兩者能相輔相成。

5.7 θ_{dyn}之效能

θ_{dyn}的值從 0 到 1 之間，在 4.1 節中就提到，它直接影響了選出的領導臉以及領導臉的個數，這個參數設置的用意，是在於〝利用較少的領導臉代表整個串列，而能使效能保持在一定的水準，運算量卻大大降低〞，以下我們以三個方面來討論θ_{dyn}，5.7.1 是觀察θ_{dyn}與領導臉個數的關係，5.7.2 是以時間的觀點觀察θ_{dyn}的變化，5.7.3 則是以效能的觀點觀察θ_{dyn}。

5.7.1 θ_{dyn}與領導臉個數

在這一小節中，我們要觀察的是θ_{dyn}與領導臉個數的關係，在相異度矩陣（dissimilarity matrix）中，當θ_{dyn}愈大領導臉的個數就愈少，也愈接近〝one face〞的情況，但領導臉的個數與θ_{dyn}的數值並非是嚴格正比的關係，下圖 5-5 是 M1 的串列 1 在不同θ_{dyn}情況下的領導臉數量，圖中當θ_{dyn}為 0 時，演員串列擁有最多的 48 個領導臉，也就是〝all face〞；當θ_{dyn}為 0.9 與 1 時，演員串列僅有一個

領導臉，即為〝one face〞的情況。圖 5-5 中，當 θ_{dyn} 下降時領導臉個數也隨之下降，但下降的趨勢逐漸平緩，θ_{dyn} 在 0.4 之後領導臉個數的變化就不大，此時我們也應當要注意，當領導臉的個數愈來愈少時，計算量也愈小，但是分群效能是否能維持在一定水準呢？另外還有穩定性的問題，我們將在 5.7.2 來討論。

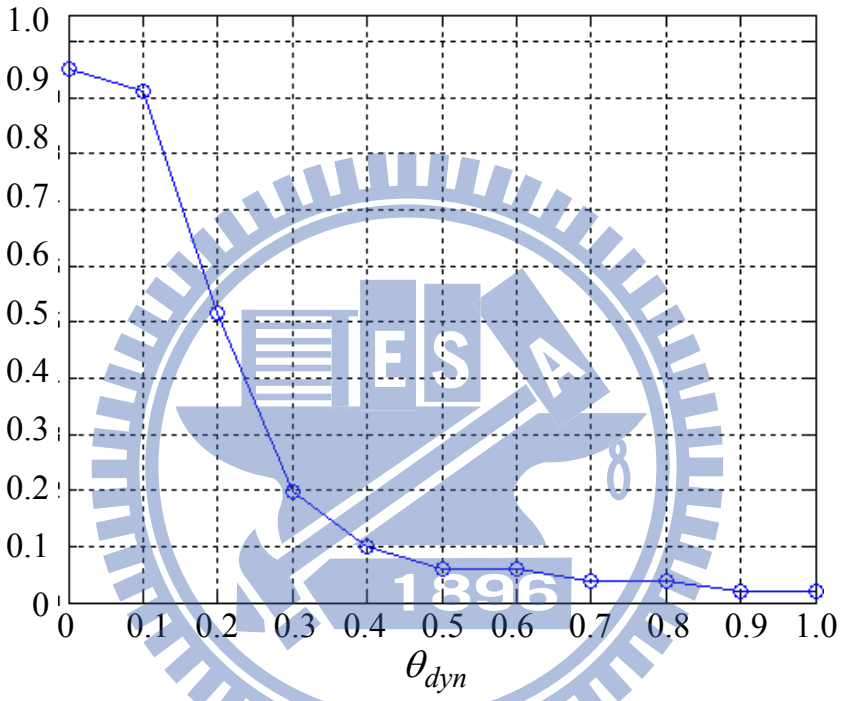


圖 5-5　串列對 θ_{dyn} 的領導臉比例曲線

5.7.2　θ_{dyn} 與時間、效能

當初在設計 θ_{dyn} 這個參數主要就是因為，影片會擷取許多影像，而影像的運算也很繁瑣，因此運算量頗大，若輸入的資料是多個合併的影片時，運算量會爆增，因此我們希望能以最簡化的運算達到不錯的效能。但若只追求運算快速，而不顧效能也是本末倒置，因此速度以及效能之間我們想取得一個平衡點，而效

能是我們優先考慮的因素，圖 5-6 是 M1~M6 對θ_{dyn}的效能曲線，圖中的六條效能曲線的趨勢並非很一致，因此我們無法下很斬斬釘截鐵的結論，只能發現當θ_{dyn}愈大時，曲線變動範圍很大的機率就愈大，換句話說，愈大的θ_{dyn}，穩定性就比較不足，我們愈無法掌控效能的變化，圖 5-5 中是以θ_{dyn}為 0.4 為界線θ_{dyn}比 0.4 大，變動幅度較大的趨勢就慢慢浮現，而這情況又以單一影片的 M1~M3 最為明顯，因為在 M1~M3 每個資料中串列彼此的平均相似度比 M4~M6 高，一旦我們調整θ_{dyn}的數值，直接影響領導臉的個數，計算相似度也不再如此細膩，因此好壞落差較大，但反觀 M4~M6，由於串列間的平均相似沒那麼高，改變領導臉個數的落差就沒 M1~M3 大，因此顯現比較平穩的曲線變化。

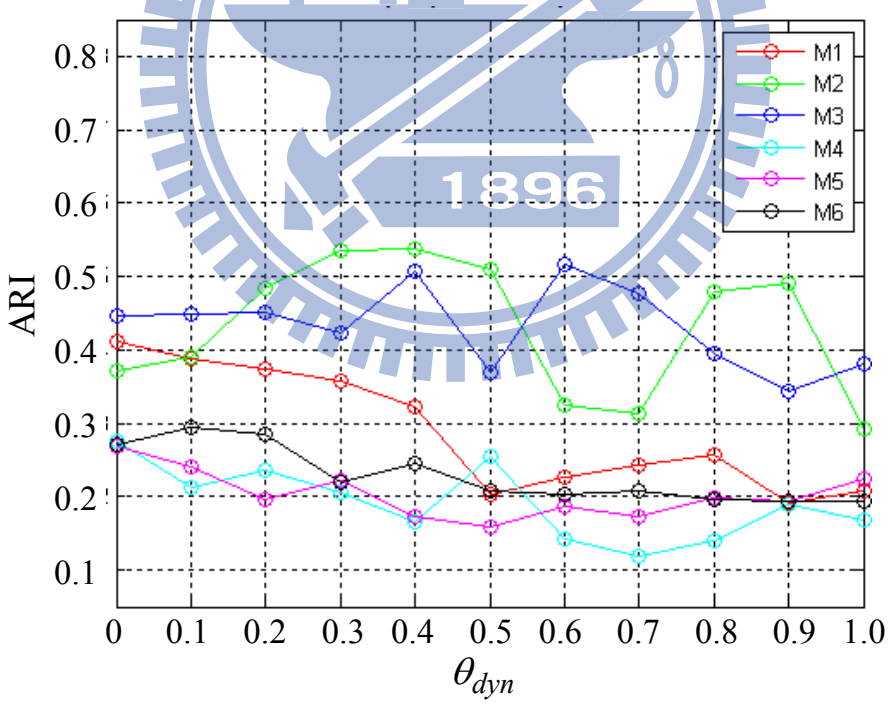


圖 5-6　M1~M6 對θ_{dyn}的效能曲線

擁有好的效能之後，再來是考慮運算量的問題，由於輸入的資料未知，有可

能是單一影片資料、或許是漫長的影片資料，又或者是多個影片合併的資料，不論怎樣，它們的運算量都不容小覷，因為這是讓使用者耗費相當大時間成本的關鍵。4.1 節提到，改變 θ_{dyn} 數值直接影響領導臉的個數，也間接影響運算量，因此我們希望以效能為第一考量點之後，減少領導臉的個數以達到〝利用些微的效能換取更多的時間成本〞為目標，實驗中以執行 k-medoid 的時間最為漫長，圖 5-7 是使用 M1 影片，調整 θ_{dyn} 的數值改變串列中領導臉的個數，並執行隨機 100 次的 k-medoid 的平均耗費時間比值。

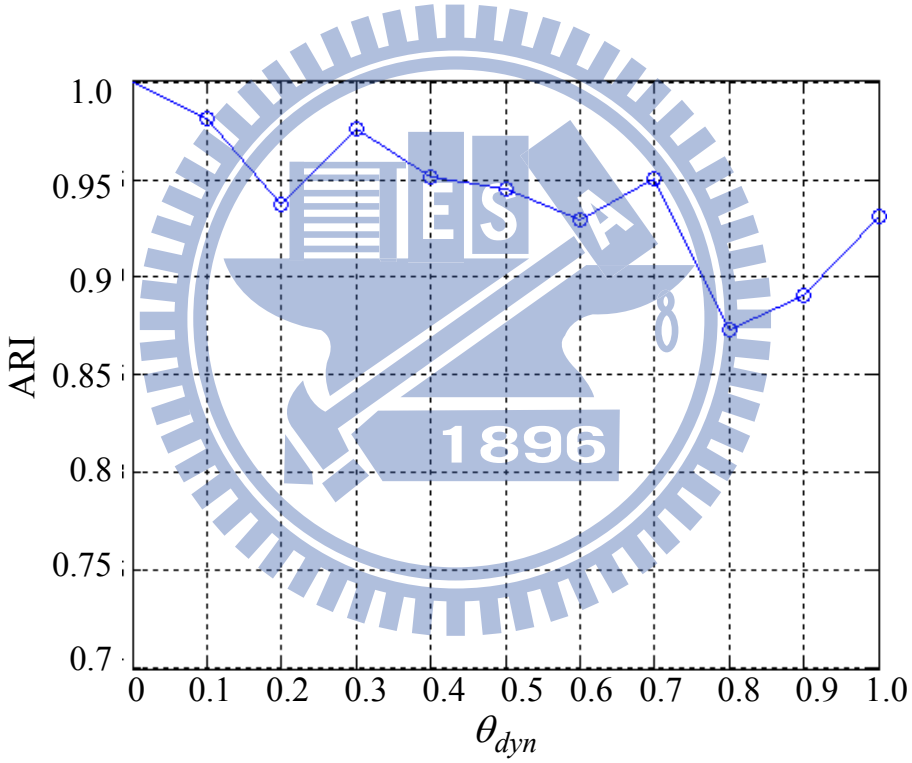


圖 5-7　在 M1 底下 θ_{dyn} 的效耗費時間比值

由於 k-medoid 初始化（initialization）的條件不同，因此曲線並為很平滑，但仍能看出 θ_{dyn} 對於時間的影響，θ_{dyn} 為 1 造成「one face」的情形，因此有效地降低時間成本。把圖 5-6 與 5-7 相互對照下，θ_{dyn} 在 0~0.3 之間的效能穩定性

較高，而較低的 θ_{dyn} 值可以降低時間成本，因此我們會建議使用 θ_{dyn} 時可以選擇較接近 0.3 的數值，如此可以取得效能與時間的平衡。

5.8 數據比較

我們把[21]的數據與我們實驗的數據以 M1 的影片作比較，依照上面各個參數值數的趨勢，表 5-4 為本實驗與[21]的效能以 ARI 的數據作比較，我們固定領導臉的選擇方式 θ_{dyn} 為 0.2、θ_{rg} 的範圍是 2~160、θ_{w} 為 0.1、θ_{agg} 為強凝聚法、θ_{rat} 是 1.0，每個群數 C 底下共有兩欄，左邊欄位是我們實驗中仿效[21]文中所敘述的條件產生的效能，右邊欄位為本實驗的實驗結果，我們把依照不同的 h 與 σ 的組合比較[21]與本實驗的實驗數據，h 與 σ 的特性是，h 愈大則身體的權重值就愈大，而 σ 愈大則身體群權重值也愈大但上升的趨勢變小。

表 5-4　與[21]數據比較表

M1	C	C = 7		C = 10		C = 20		C = 30	
h	σ	Su's	Wei's	Su's	Wei's	Su's	Wei's	Su's	Wei's
h = 0.2	σ= 1500	0.566	0.558	0.448	**0.482**	0.466	**0.521**	0.426	**0.451**
	σ= 3000	0.447	**0.544**	0.427	**0.572**	0.502	**0.527**	0.452	**0.454**
	σ= 6500	0.375	**0.485**	0.465	0.412	0.485	0.455	0.414	**0.42**
h = 0.4	σ= 1500	0.463	**0.507**	0.419	**0.458**	0.452	**0.462**	0.339	**0.391**
	σ= 3000	0.408	0.405	0.412	**0.416**	0.458	0.402	0.359	**0.361**
	σ= 6500	0.255	**0.447**	0.358	**0.447**	0.375	**0.382**	0.346	**0.378**
h = 0.6	σ= 1500	0.306	**0.412**	0.414	0.383	0.452	0.425	0.333	**0.344**
	σ= 3000	0.398	0.358	0.406	0.323	0.422	0.387	0.38	0.353
	σ= 6500	0.262	**0.316**	0.331	0.316	0.357	**0.388**	0.319	**0.359**

首先，針對本實驗的效能而言，我們觀察 h 對於效能的變化，不論是固定 σ 或是群數 C，我們發現當 h 愈大則效能就降低，這是可理解的，因為 h 上升將會提升身體的權重值，反倒抑制臉部影像對於結果的貢獻，而在人物分群中，最重要的資訊就是人臉特徵，其他的資訊不論是身體影像、場景資訊、相似度或是重疊資訊，都只是輔助分群的工具，因此三個 h 值中以 0.2 表現最好的效能，這是可以理解的，而 h 也並非選擇小的就是好，因為如果沒有身體資訊那一切都只使用臉部資訊來分群，又將回到人物分群的起始點，因此適當地使用身體資訊確實能幫助效能提升；同樣地，表格中，σ 愈大則效能愈低，這也是同樣的道理，只是數據的改變量沒有 h 那麼大，因此這也是為什麼我們實驗一開始就設定 h 為 0.2、σ 為 1500 的原因，而群數 C 對於效能的趨勢，就如同 5.3 節提到的，因為不同的群數對於 ARI 的計算影響很大，因此表中也以〝較接近實際群數 C 條件擁有較高的效能〞的趨勢。

接下來，我們針對本實驗仿造[21]作法所呈現的效能，與加入叢集整合觀念的人物分群結果相比較，表格中紅色粗體字為本實驗優於[21]的情形，仔細觀察會發現，當 h 比較小時，σ 在小值的部分明顯優於[21]，且相差較大；h 比較大時，儘管與[21]差距不大，但就是無法提升效能，這是因為兩個實驗的差異就在於，本實驗中加入叢集整合的相似度矩陣，而由於我們認為臉部向量是人物分群最重要的向量，因此我們以串列的領導臉向量作為產生叢集相似度矩陣的來源，無非是想再次藉由臉部向量得到更可靠的資訊，因此在小 h 與小 σ 能有很好的表現，全是因為在人物相似度矩陣中，身體的權重值偏大，再加上以身體影像為主要特徵的叢集整合相似度，兩者結合更能充分表達分群的特徵，達到提升效能的目的。

第六章 結論與未來展望

本文使用了美國影集作為人物分群的實驗對象，它的場景轉換、背景雜亂、變動的人物特徵以及衣服資訊等特性，使得人物分群困難度增加，因此我們不僅利用臉部影像，也納入身體影像輔助分群搭配串列時間軸產生的權重，以及分鏡資訊，產生人物相似度矩陣 S_P，另外加入叢集整合的概念作為輔助分群的資訊，在這麼雜亂的環境中，ARI 的最好效果可以達到六成，這是由於凝聚法所使用的相似度矩陣 S，不只包含演員串列特性的人物相似度矩陣，還增加了叢集整合相似度矩陣 S_E，確實有助於提升分群效能。另外，針對叢集整合的參數使用，我們會建議 θ_{dyn} 挑選接近 0.3 的數值可取得效能與時間的平衡點，θ_{rg} 則選較大的隨機範圍，θ_w 可選 0.1~0.3 之間作為輔助分群的資訊，θ_{agg} 選擇兩階段強凝聚法，θ_{rat} 要視資料雜亂性而定，過於雜亂可往低數值挑選起，但是盡量不要低於 0.5。

實驗中的效能符合我們所期待，而變數值也透漏更多思考的方向，在未來的時間裡，系統有些地方仍可以作加強，首先，我們知道從數據得知，加入 S_E 有助於提升效能，但是其影響結果的能力仍小於 S_P，因此若能增加 S_E 的可靠度，使得最佳效能的 W_E 上升，勢必能提升最佳效能；第二，系統中嘗試許多變數的數值，得到效能的某些規律以及結論，而我們可以更深入的探討每一個變數對於結果的穩定性以及敏感度，這樣將有助於系統評估變數值的好壞以及使用或取代；第三，本文中除了根據文獻得到的觀點，也從數據的規律中得到結論，因此在未來若能發展一套讓系統對於影片本身的特性，自動產生或推測對於結果最有利的參數值，輔助系統找出最正確的分群結果，這樣有助於減少耗費在找尋規律性的時間成本；第四，在實驗中我們只知道當最終凝聚的群數接近真正演員個數時，效能會提升，卻未對影片中人物個數產生預測，因此在未來的研究工作上可增加系統在群數的推測。最後，承如 2.2 節提到的，影像的運算是很繁複的，如果辨識影像的系統能夠克服無法使用在高維度以及大型資料集的缺陷，那麼辨識系統將會被更廣泛地運用在日常生活中。

參考文獻

[1] W. Y. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: a literature survey," ACM Computing Surveys (CSUR), vol. 35, no. 4, pp. 399-458, 2003.

[2] J. Tao and Y. P. Tan, "Efficient clustering of face sequences with application to character-based movie browsing," *Proc. IEEE International Conference on Image*, pp.1708~1711, 2008.

[3] D. Ramanan, S. Baker, and S. Kakade, "Leveraging archival video for building face datasets," *Proc. IEEE International Conference on Computer Vision*, pp. 1-8, 2007.

[4] E. El-Khoury, C. Senac, and P. Joly, "Face-and-Clothing Based People Clustering in Video Content," *Proc. International Multimedia Conference on Multimedia Information Retrieval*, pp. 295-304, 2010.

[5] P. Huang, Y. Wang, and M. Shao, "A New Method for Multi-view Face Clustering in Video Sequence," *Proc. IEEE International Conference on Data Mining Workshops*, pp. 869-873, 2008.

[6] K. Yamamoto, O. Yamaguchi, and H. Aoki, "Fast face clustering based on shot similarity for browsing video," Progress in Informatics, pp. 53-62, 2010.

[7] A.L.N. Fred and A.K. Jain, "Combining multiple clusterings using evidence accumulation," IEEE Transactions On Pattern Analysis And Machine Intelligence, vol. 27, no. 6, pp. b835-850, 2005.

[8] A. Strehl and J. Ghosh, "Cluster ensembles - a knowledge reuse framework for combing multiple partitions," J. Machine Learning Research, vol. 3, pp. 583-617, 2002.

[9] X.Z. Fern and C.E. Brodley, "Random projection for high dimensional data clustering- a cluster ensemble approach," *Proc. 20th Int'l Conf. Machine Learning(ICML)*, 2003.

[10] P. Hore, L. Hall, and D. Goldgof, "A cluster ensemble framework for large data sets," *Proc. 2006 IEEE Int'l. Conf. System, Man, and Cybernetics*, pp. 3342-3347, 2006.

[11] P. Viswanath and K. Jayasurya, "A fast and efficient ensemble clustering method," *Proc. 2006 Int'l Conf. Pattern Recognition(ICPR)*, vol. 2, pp.720-723, 2006.

[12] B. Minaei-Bidgoli, A. Topchy and W. F. Punch, "Ensembles of partitions via data resampling," *Proc. 2004 Int'l. Conf Information Technology*, vol. 2, pp. 118-192, 2004.

[13] B. Fischer and J.M. Buhmann, "Bagging for path-based clustering," IEEE Trans. Pattern Analysis Machine Intelligence, vol. 25, no. 11, pp. 1411-1415, 2003.

[14] A.P. Topchy, M.H.C. Law, A.K. Jain, and A.L. Fred, "Analysis of consensus partition in cluster ensemble," *Proc. 4th IEEE Int'l Conf. Data Mining(ICDM)*, pp. 225-232,

2004.
[15] H. Luo, F. Koug, and Y. Li, “Clustering mixed data based on evidence accumulation,” LNCS, vol. 4093, pp. 348-355, 2006.
[16] X. Wang, C. Yang, and J. Zhou, “Clustering aggregation by probability accumulation,” Pattern Recognition Letters, vol. 42, no. 5, pp. 668-675 , 2009.
[17] X.Z. Fern and C.E. Brodley, “Solving cluster ensemble problems by bipartite graph partitioning,” Proc. 21th Int’l Conf. Machine Learning(ICML), ACM International Conference Proceeding Series, vol.69, pp.281-288, 2004.
[18] R.N. Dave, “Characterization and detection of noise in clustering,” Pattern Recognition Letters, vol. 12, no. 11, pp. 657-664, 1991.
[19] L. Hubert and P. Arabie, “Comparing partitions,” Journal of Classification, vol. 2, no. 2-3, pp.193-218, 1985.
[20] K. Y. Yeung, W. L. Ruzzo, “Details of the Adjusted Rand index and clustering algorithms supplement to the paper “an empirical study on principal component analysis for clustering gene expression data”,” vol. 17, no. 9, pp. 763-774, 2001.
[21] 蘇偉誌, “Video indexing by information of face images,” 交通大學多媒體工程研究所碩士論文, 2009.
[22] http://dvdvideosoft.com/download/FreeVideoToJPGConverter.exe
[23] http://opencv.willowgarage.com/wiki/Welcome
[24] 洪詩祐, “Automatic skin detection using face information,” 交通大學多媒體工程研究所碩士論文, 2009.