

# 一種利用物件形狀來進行 MPEG-4 鏡頭變化偵測之技術

劉志俊 傅佳源 王志浩 喻仲平

中華大學資訊工程學系

ccliu@chu.edu.tw

## 摘要

隨著科技的日新月異，數位化的多媒體格式也越來越多，如 MPEG-1、MPEG-2 和最新的 MPEG-4 等視訊格式，都累積了大量的影片資料。由於整個影片長度太長，所以一般都以鏡頭 (Shots) 作為視訊資料索引查詢的單位。在傳統上，以往的鏡頭變化偵測技術是以顏色為基礎，比較連續兩張畫面的顏色或顏色統計差異來判斷鏡頭變換的位置。在本文中，我們將以 MPEG-4 格式的視訊資料為主，利用其中物件的形狀資訊，來作鏡頭變化偵測。由於形狀資訊遠小於顏色資訊，本方法充分利用 MPEG-4 編碼器在做物件偵測計算過程中所產生的形狀資訊，故可比以往之方法更有效率的完成 MPEG-4 影片的鏡頭變化偵測。

## 1. 序論

在影片中測出鏡頭的變換(shot change)位置是一項重要且基本的技術。因為鏡頭是一部影片拍攝的基本單位，舉凡視訊資料的內涵式索引與查詢、影片內涵分析、視訊摘要等研究都要用到此技術。早期的鏡頭變換偵測技術，如 Nagasaka 與 Tanaka [1] 以及 Zhang 等人[2][3]所提出的方法，是以兩畫面的像素差異或是畫面的顏色直方圖統計(color histogram)差異作為判斷的基準，將差異大於門限值(threshold)的畫面所在視作鏡頭變換位置。Otsuji 與 Tonomura [4]則是綜合使用像素亮度差異、像素變化面積、亮度直方圖統計差異、區塊顏色關連性等四種參數來計算兩

相鄰畫面的差異，經由分析畫面差異的統計分布(distribution)來找出鏡頭變換的門限值設定。此外，影片格式轉換、慢動作攝影、動畫與閃光等因素所產生的干擾，亦通過一映射偵測過濾器(projection detecting filter)予以降低。

早期的鏡頭變換偵測技術是針對一般未壓縮的數位視訊資料上來進行。但隨著 MPEG-1 與 MPEG-2 的日漸普遍，研究的重點逐漸強調要在壓縮的視訊資料上來進行(compressed domain)。MPEG 壓縮的視訊資料有兩種資訊：離散餘弦轉換係數(DCT, Discrete Cosine Transform)以及移動補償向量(MV, Motion Vector)。前者是為了減少空間上的多餘資訊(spatial redundancy)，而後者是為了減少時間軸上的多餘資訊(temporal redundancy)。因此，有許多鏡頭變換的技術是利用離散餘弦轉換係數[5][6][7][8][9][10]或移動補償向量[11][12][13][14]來進行。Armen 等人觀察到對於一個典型的影片來說，使用傳統的方法比對一個畫面需要  $10^5$  個比較運算，而使用離散餘弦轉換係數的則僅需  $10^3$  個比較運算[5]。Armen 等人的方法是對每張畫面選取若干個相連的區塊(blocks)，對每個區塊取出一定數量的離散餘弦轉換係數作為代表此畫面的特徵向量，再以兩特徵向量的內積結果來判斷兩張畫面的差異度。Yeo 與 Liu 則是利用離散餘弦轉換係數中的 DC 值來作鏡頭變換偵測[6]。在其方法中，每張影像先轉換為以區塊為單位的 DC 影像，再以兩張 DC 影像的差異來判斷兩張畫面的差異度。Kang 等人則是計算 DC 影像的累積差異統計(AHIM, Accumulated Histogram Intersection Measure)

---

\*本論文研究為國科會補助之研究成果，  
計劃編號 NSC 90-2213-E-216-010

來作鏡頭變換偵測[7]。Fernando 等人對漸變型鏡頭變換採取統計的方式來進行偵測[8]。在其論文中，他們首先對淡入(fade-in)、淡出(fade-out)、溶接(dissolve)、搖攝(wiping)等漸變型鏡頭變換以數學公式予以正式定義。每一張畫面被轉換為一組區塊的集合，對每個區塊分別記載其平均能量與能量標準差。若有淡入、淡出或溶接的情況發生時，其平均能量與能量標準差會呈現一些規則。而對搖攝的情況可以利用 Hough 轉換的方式來偵測。Lee 等人的方法相當特殊[9]。他們使用區塊的 DCT 係數將區塊轉為一張二元邊緣圖(binary edge map)並計算此區塊內的邊緣的方向(orientation)、強度(strength)與偏移量(offset)。接著再統計每張畫面中的邊緣方向與強度作為判斷鏡頭變換的依據。清華大學黃仲陵教授在[10]一文中提出對突然型鏡頭變更(abrupt scene-change)以及漸變型鏡頭變更(gradual scene-change)的一套偵測方法。對突然型鏡頭變更的衡量先以兩畫面中對應區塊的平均 DC 值差異來作初步判斷(類似 Yeo 等人的方法[6])，接著再以亮度直方圖的 $\chi^2$ 測試( $\chi^2$  test)作進一步確認，以補足兩種方法各自的缺點，減少誤判的次數。而對漸變型鏡頭變更的衡量亦是以 DC 值差異為基礎，將變化的過程分為四個階段，設計出適當的演算法來計算各階段的 DC 值差異特性，以偵測出漸變型鏡頭變更的開始畫面與結束畫面。在使用移動補償向量的方法方面，Bhandarkar 與 Khombhadia 利用 B 畫面(B frames)與 P 畫面(P frames)中前向參考移動向量(forward predicted motion vector)與後向參考移動向量(backward predicted motion vector)來計算每個區塊的移動距離，一個畫面中所有區塊的移動距離再經過加總而得到此畫面的移動邊緣值(motion edge magnitude)，由此值的大小來估算發生鏡頭變換的可能性[11]。Nang 等人也採用類似的方法計算 B 畫面中區塊參考狀

況來決定是否為鏡頭變換點[12]。Sugano 等人利用一個畫面中各種類型的區塊個數以及 DC 值來作 MPEG-2 影片鏡頭變換偵測，並對最棘手的突然變換(abrupt scene change)、閃光(flashlight)、溶出特效(dissolve)、捲入特效(wipe)等困難問題提出解決方法[13]。Shin 等人提出一種階層式的 MPEG-2 影片鏡頭變換偵測方法[14]。首先，一個畫面群組(GOP, group of picture)的平均 DC 值標準差被用來計算此畫面群組中是否有鏡頭變換。接著一個畫面中的前向參考移動向量與後向參考移動向量比值被用來計算此畫面是否有鏡頭變換。

Zhang 等人的方法則是同時利用到離散餘弦轉換係數與移動補償向量的資訊以提高鏡頭變換偵測的正確率[15][16]。Gamaz 等人亦是使用 B 與 P 畫面中參考到外部畫面的區塊個數比例與離散餘弦轉換係數中的 DC 值來對 MPEG-2 影片進行鏡頭變換偵測[17]。Bouthemy 等人提出的方法將攝影機移動(camera motion)的性質納入，提出一種整合攝影機移動與鏡頭變換偵測的綜合方法[18]。有關各種鏡頭變換偵測方法的比較與效能分析可以參考[19][20]。

在以往所提出的鏡頭偵測技術中，皆是對未壓縮視訊，或是 MPEG-1 和 MPEG-2 格式的視訊資料為主。鏡頭偵測所需要的特徵值(如顏色、顏色統計直方圖、質地、小波轉換係數、移動補償向量等)，是取自一整張畫面(frame)，因此其計算較為複雜，不易發展出快速的分鏡技術。而在 MPEG-4 視訊壓縮方法中，其最大的特色就是以物件為基礎來進行壓縮。根據 MPEG-4 標準[21]，每一畫面都是由視訊物件平面(VOP)所構成。視訊物件平面則包含了物件形狀和材質的資料，提供了 MPEG-1 及 MPEG-2 標準中沒有的額外物件與形狀資訊。因此，我們可以利用這些資訊針對 MPEG-4 格式的視訊資料來設計一種鏡

頭變化的快速偵測演算法。

本論文的結構說明如下：在第二節中我們將說明所提出之 MPEG-4 鏡頭變化偵測系統的系統架構，在第三節中說明 MPEG-4 對物件形狀的編碼方式及我們鏡頭變化偵測方法所需要的特徵值的擷取方式。以形狀為基礎的鏡頭變化衡量函數在第四節中討論。第五節說明本方法實際應用在 MPEG-4 電影片段的實驗結果。最後，第六節總結本論文並提出我們的未來研究方向。

## 2. 系統架構

如圖 1 所示，本文所提出之 MPEG-4 鏡頭變化偵測系統 (MPEG-4 shot change detector) 是由 MPEG-4 解碼器 (MPEG-4 decoder)、MPEG-4 形狀特徵值擷取器 (MPEG-4 feature extractor)、鏡頭變化偵測器 (shot change Detector) 等三部份所構成，各部份的功能及運作方式說明如下。

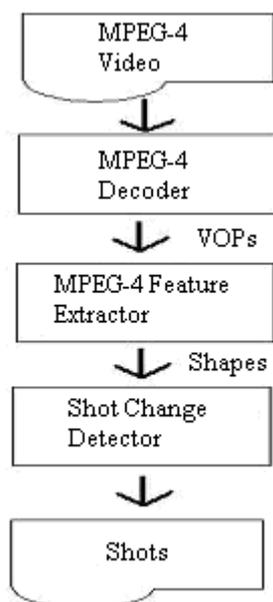


圖 1. MPEG-4 鏡頭變化偵測系統

- MPEG-4 解碼器：本文所提出的方法是利用 MPEG-4 視訊物件平面的物件形狀資訊來作鏡頭變換的判斷。所以我們需要使用 MPEG-4 解碼器[22]在解碼過程中取出每

一畫面中的物件形狀資訊及其移動向量 (對 P-VOP 與 B-VOP)。

- MPEG-4 形狀特徵值擷取器：MPEG-4 形狀資料基本上都是以算術編碼法來作壓縮。我們必須將算術編碼後的形狀資料解碼，並計算其數量、大小、位置、形狀等特徵值。
- 鏡頭變化偵測器：在同一個鏡頭裡的連續兩張畫面，其中所含的 VOP 形狀、大小、數量、位置等資料十分相似。我們可以利用此項特性來作鏡頭變化偵測器。所需的計算量會遠小於傳統的鏡頭變化偵測演算法。

## 3. 物件形狀特徵值的擷取

### 3.1 MPEG-4 物件形狀解碼

MPEG-4 是第一個以物件為基礎(object-based)的多媒體壓縮標準[21]，也是第一個涵蓋形狀編碼(shape coding)的國際標準。有關 MPEG-4 的簡介可參考[23][24]；有關 MPEG-4 形狀編碼的介紹可參考[25][26]。有關 MPEG 組織選擇算術編碼法來作形狀資訊壓縮的考量可參考[27][28]。

視訊物件平面則包含了物件形狀和材質的資料，如圖 2 所示，提供了 MPEG-1 及 MPEG-2 標準中沒有的額外物件與形狀資訊。

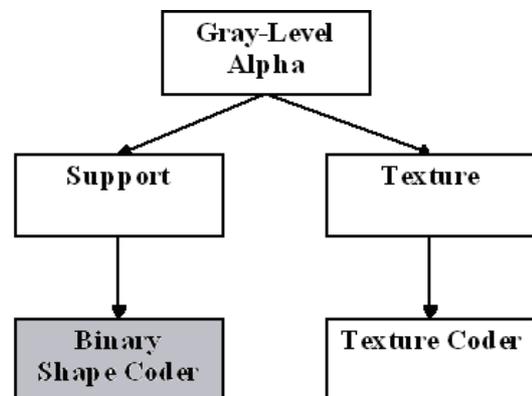


圖 2. MPEG-4 的灰階式形狀編碼法  
形狀編碼基本上有曲線法 (curve-based

approach)與像素法(pixel-based approach)兩大類，而 MPEG-4 標準最後決定採用像素法來作形狀編碼。MPEG-4 的形狀資料有兩種形式：灰階(grey scale)形狀與黑白二元(binary) 形狀。在灰階形狀資料中，對每個像素(pixel)儲存一個 0 到 255 的正整數，代表此像素的透明度 (degree of transparency)，其中 0 代表完全透明；255 代表完全不透明。灰階形狀資料也就是傳統影像處理技術中的 $\alpha$ 頻道(alpha channel)。而在黑白二元形狀資料中，對每個像素在 0 或 255 兩數值中擇一儲存。灰階與黑白二元形狀資料基本上都是以基於背景的算術編碼法(CAE, Context-Based Arithmetic Encoding)來進行壓縮。因此，在一個 MPEG-4 影片中的每一個視訊物件平面，其黑白二元形狀資料可以由其原始資料(raw data)，透過算術編碼器/解碼器[29][30]解出。本文所使用之 MPEG-4 解碼器為 Microsoft 所發展[22]。

### 3.2 MPEG-4 物件形狀特徵值擷取

MPEG-4 視訊和以往 MPEG-1、MPEG-2 相比最大的特色即在 MPEG-4 是以物件為基礎 (Object-Based) 的壓縮方法。MPEG-4 標準將畫面中的每個視訊物件平面(VOP)分為形狀和紋路分別來儲存。我們取出並計算每一畫面中的 VOP 形狀、大小、數量、位置等作為判斷鏡頭變換用的特徵值。



圖 3. MPEG-4 物件形狀範例

舉例來說，圖 3 為電影”美麗境界”中的一個畫

面，其中包含了兩個 VOP。根據 MPEG-4 黑白二元形狀解碼後之 VOP 如圖 4 所示。



圖 4. MPEG-4 黑白二元形狀範例

表 1. MPEG-4 物件形狀特徵值範例

物件代號	大小	位置	形狀
VOP1	144 x 400	(220, 300)	
VOP2	224 x 256	(410, 470)	

### 4. MPEG-4 鏡頭變換偵測技術

在同一個鏡頭裡的連續兩張畫面，其中所含的 VOP 形狀、大小、數量、位置等資料十分相似。相對地，鏡頭變換處的連續兩張畫面，即使是換一個角度拍攝同一場景(scene)，其中所含的 VOP 形狀、大小、數量、位置等資料亦會有很大的差異。舉例來說，在 VOP 數量大幅變更之處，如圖三所示，VOP 由 1 個變為 6 個時，我們可以依據變更的多寡加重鏡頭變化的權重值。

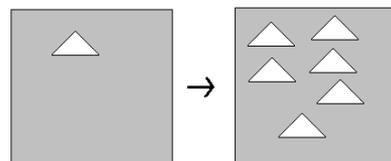


圖 5. 兩鏡頭的 VOP 數量差異

同一個鏡頭中的 VOP 個數基本上是不太會變化的。但也有可能在中途插入一個或數個新的 VOP，或者是一個或數個舊有 VOP 移出鏡頭之外。所以我們必須先能判斷

連續兩張畫面中的兩個 VOP 群組間，是否有對應關係，換句話說，兩個 VOP 是否代表同一個物件的 VOP 識別技術 (VOP identification)。此外，如果兩張連續畫面各有  $m$  個及  $n$  個 VOPs，則我們需要  $m*n$  個交叉比對。為了減少交叉比對的複雜度，提昇系統的效能，我們也需要知道前後兩個 VOPs 是否為同一 VOP。

VOP 識別技術有兩種做法：一種是計算兩個 VOP 的顏色、質地與形狀間的相似度，滿足一定的相似度門限值即可認為此兩個 VOP 為同一個物件。第二種做法是把問題丟給 MPEG-4 編碼器。也就是說，MPEG-4 在編碼過程中本來就必須作 VOP 識別並計算 VOP 參考移動補償向量來提高壓縮比。在本論文中，我們認為應該採用第二種做法以充分利用 MPEG-4 的物件特性以及 MPEG-4 編碼器在 VOP 識別所花費的努力。根據標準，形狀亦可有移動補償 (motion compensation)。所以我們可利用形狀移動補償向量 (MV, Motion Vector of shape) 來作物件追蹤 (object tracking)。也就是若一個黑白二元形狀是以移動補償的方式編碼，則我們視為此形狀與參考的形狀為同一個物件。

對連續兩張 MPEG-4 畫面  $X$  與  $Y$  而言，假設前一張畫面有  $L$  個 VOP ( $x_1, x_2, \dots, x_L$ ) 與後一張畫面的  $L$  個 VOP ( $y_1, y_2, \dots, y_L$ ) 兩兩對應；前一張畫面有  $M$  個 VOP ( $x_{L+1}, x_{L+2}, \dots, x_{L+M}$ ) 由後一張畫面中消失、後一張畫面比前一張畫面新出現  $N$  個 VOP ( $y_{L+1}, y_{L+2}, \dots, y_{L+N}$ )。則此兩張畫面的相似度可以定為

$$\text{Similarity}(X, Y) = \sum_{i=1}^L \text{Sim}(x_i, y_i) + \sum_{i=L+1}^{L+M} \text{Sim}(x_i) + \sum_{i=L+1}^{L+N} \text{Sim}(y_i) \quad (1)$$

在(1)式中

$$\sum_{i=1}^L \text{Sim}(x_i, y_i) \text{ 代表 } L \text{ 個共同 VOP 間的相似度;}$$

$$\sum_{i=L+1}^{L+M} \text{Sim}(x_i) \text{ 代表消失 } M \text{ 個 VOP 的相似度;}$$

$$\sum_{i=L+1}^{L+N} \text{Sim}(y_i) \text{ 代表新增 } N \text{ 個 VOP 的相似度。}$$

兩個 VOP ( $x$  與  $y$ ) 間的相似度函數  $\text{Sim}(x, y)$ ，需考量其位置、大小與形狀。

在 VOP 位置方面，如圖四所示，圖中 VOP 的位置由中央變到了角落，如此我們便可依移動的距離長短來調整權重值。

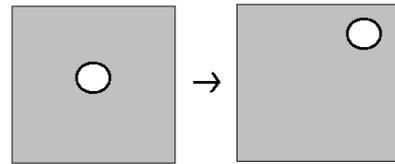


圖 6. 兩鏡頭的 VOP 位置差異

兩個 VOP ( $x$  與  $y$ ) 間的位置差異計算分為兩種情形：若  $x$  與  $y$  為同一物件，即  $y$  參考  $x$ ，則其位置差異為  $y$  的所有構成 BAB 的移動補償向量的平均值。若  $x$  與  $y$  為不同物件，我們計算離  $y$  最近的物件的質心座標與  $y$  的質心座標的歐幾里得距離。

在 VOP 大小方面，如圖六所示，VOP 大小劇烈變化，也可能是兩個不同的鏡頭。

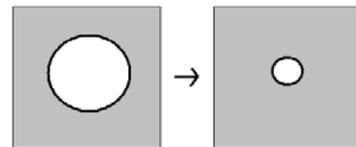


圖 7. 兩鏡頭的 VOP 大小差異

VOP 大小可以用其最小包含矩形 (MBR, Minimal Bounding Rectangle) 的面積來表示。事實上，MPEG-4 形狀編碼器本來就是以最小包含矩形作為 VOP 的編碼區域。不過 MPEG-4 的最小包含矩形的長與寬一定是 16 的倍數。理由是 BAB 是  $16 \times 16$  的區塊。因此，我們可以直接由 MPEG-4 解碼中取得物件的高 ( $\text{vop\_height} * 16$ ) 與寬 ( $\text{vop\_width} * 16$ )，面積即可直接由高寬相乘而得。計算其最小包含矩形面積的差異即可得到兩 VOP 大小的差異。

最後是 VOP 形狀變化，由圖五可以看的出來，圖中 VOP 的形狀由圓形變成了多邊形，這樣我們就判斷可能是兩個不同的鏡頭。

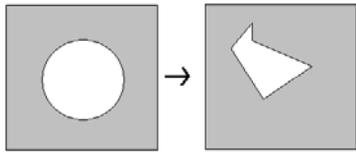


圖 8. 兩鏡頭的 VOP 形狀差異

形狀的差異可以經由大小正規化後以簡單計算重疊比例，例如修正式 Hamming 距離(modified Hamming distance)[31]，或是由其他較複雜的形狀相似性比對演算法來計算，例如 Hausdorff 距離(Hausdorff distance)[32]。在本文中，我們使用修正式 Hamming 距離來計算。值得注意的是，我們的計算單位為 BAB，也就是 16 x 16 的區塊而非像素，以減少計算量。而進行大小正規化的理由是我們已在 VOP 大小比較時考慮過其影響。

因此，我們可以設計一個連續畫面相似性函數，輸入連續兩張畫面中所有 VOPs 的大小、位置、形狀、數量，來計算其間的相似度。並設定一門限值來決定此兩張畫面間是否為一鏡頭變換點。

## 5. 實驗

為了證明本文所提出方法的可行性，我們實際應用在兩部電影”臥虎藏龍”與”美麗境界”。由於目前並無 MPEG-4 格式的電影，我們的電影資料取自 DVD 格式，再由 MPEG-4 編碼器轉為 MPEG-4 格式。對每部電影先以人工方式取出四個片段，每個片段包含 10 次連續的鏡頭變換。因此，實驗總共使用 88 個鏡頭，其中有 80 次鏡頭變換。

圖 9 為鏡頭變換前與鏡頭變換時的物件個數差異統計。由統計結果可以得知，在同一鏡頭內物件個數大抵不會變更，但”臥虎藏龍”的鏡頭發生 3 次新增物件的狀況，造成一些誤判的狀況。而鏡頭變換時的物件個數差異

就比較明顯。

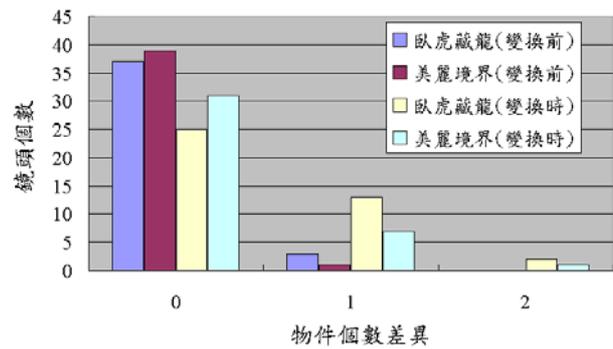


圖 9. 鏡頭變換前與鏡頭變換時的物件個數差異統計

圖 10 為鏡頭變換前與鏡頭變換時的物件移動位置統計(以像素單位，質心為計算根據)。由結果可以得知，在同一鏡頭內物件移動距離在 10 像素以內佔 80%以上，而鏡頭變換時的物件物件移動距離大多數在 32 像素以上。因此可以將物件移動距離的門限值定在 32 像素。

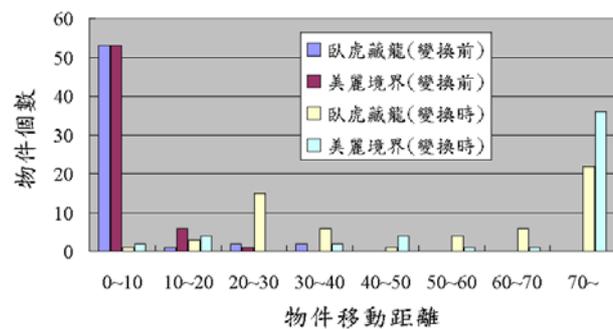


圖 10. 鏡頭變換前與鏡頭變換時的物件移動距離統計

圖 11 為鏡頭變換前與鏡頭變換時的物件大小差異統計。由結果可以得知，在同一鏡頭內物件大小變化大致不會超過 8 個區塊，而鏡頭變換時的物件物件大小變化多在 32 個區塊以上。因此可以將物件大小變化的門限值定為 32 區塊。

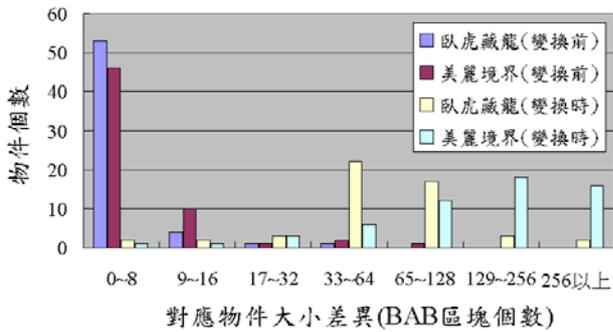


圖 11. 鏡頭變換前與鏡頭變換時的物件大小差異統計

圖 12 為鏡頭變換前與鏡頭變換時的物件形狀差異統計，計算的單位為每個區塊的平均 Hamming 距離。由結果可以得知，在同一鏡頭內物件形狀變化不大，而鏡頭變換時的物件物件大小變化約呈常態分布，平均值為 0.4。

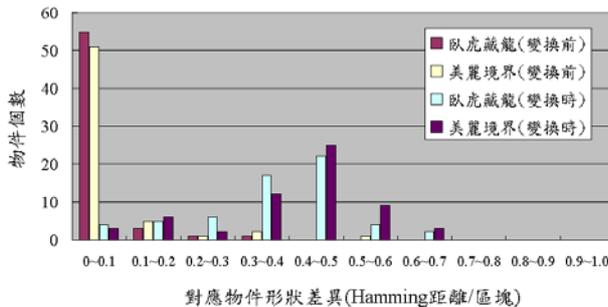


圖 12. 鏡頭變換前與鏡頭變換時的物件形狀差異統計

表 2 為兩部電影的最終鏡頭變換判斷結果。如前所述，誤判(false alarm)發生在有新的物件加入鏡頭內的時候。而回覆率(recall)相當不錯，在 92.5%以上。

表 2. 鏡頭變換判斷結果

	臥虎藏龍	美麗境界
鏡頭變換數	40	40
誤判個數	2	4
未判出個數	3	2
正確率(Precision)	95%	90%
回覆率(Recall)	92.5%	95%

## 6. 結論

本文提出一種利用視訊中的物件形狀資訊來對 MPEG-4 格式的視訊資料進行鏡頭變化偵

測的方法。其應用包含：

### (1) 視訊資料的內涵式查詢

傳統的查詢方式係利用文字的模式來下達查詢，並不能針對影音資料的內涵直接設定查詢條件。而內涵式資料庫查詢則是運用多媒體資料本身的特徵來下達查詢。一般視訊資料的內涵式查詢亦是以鏡頭作為查詢與索引的單位。

### (2) 視訊摘要

若能將資料龐大的視訊資料，擷取出視訊摘要，以精要的內容呈現出視訊資料的大致內容，則使用者可以透過視訊摘要，快速地瀏覽視訊資料。視訊摘要的擷取需要使用鏡頭變化偵測的技術。

### (3) 視訊內容分析

我們可以統計各種不同影片，如文藝片、動作片、運動、廣告，或是依照導演統計各種影片中鏡頭變化的頻繁度與鏡頭長度試著設計自動的影片分類的方法。

## 7. 參考文獻

- [1] Nagasaka, A. and Y. Tanaka, "Automatic video indexing and fullvideo search for objects appearances," *Visual Database Systems II*, E. Knuth and L. M. Wegner, Eds. New York: Elsevier Science, pp.113-127, 1992.
- [2] Zhang, H. J., A. Kankanhalli, and S. W. Smoliar, "Automatic partitioning of full-motion video," *Multimedia Systems*, Vol.1, No.1, pp.10-28, June 1993.
- [3] Smoliar, S.W. and H. J. Zhang, "Content based video indexing and retrieval," *IEEE Multimedia*, Vol. 1, No. 2, pp. 62-72, Summer 1994.
- [4] Otsuji, K., and Y. Tonomura, "Projection detecting filter for video cut detection," in *Proc. of ACM Intl. Conf. on Multimedia*, pp. 251-257, 1993.
- [5] Arman, F., A. Hsu, and M. Y. Chiu, "Image processing on compressed data for large video databases," in *Proc. of ACM Intl. Conf. on Multimedia*, pp.267-272, August, 1993.
- [6] Yeo, B.-L., and Liu, B., "Rapid scene analysis on compressed videos," *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 5, No. 6, pp. 533-544, Dec. 1995.
- [7] Kang, E. K., S. J. Kim, and J. S. Choi "Video

- retrieval based on scene change detection in compressed streams,” *IEEE Transactions on Consumer Electronics*, Vol. 45, No. 3, pp. 932–936, Aug. 1999.
- [8] Fernando, W.A.C., C.N. Canagarajah, and D.R. Bull, “A unified approach to scene change detection in uncompressed and compressed video,” *IEEE Transactions on Consumer Electronics*, Vol. 46, No. 3, pp. 769–779, Aug. 2000.
- [9] Lee S.W., Y. M. Kim, and S. W. Choi, “Fast scene change detection using direct feature extraction from MPEG compressed videos,” *IEEE Transactions on Multimedia*, Vol. 2, No. 4, pp. 240–254, Dec. 2000.
- [10] Huang C. L. and B. Y. Liao, “A robust scene-change detection method for video segmentation,” *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 11, No. 12, pp. 1281–1288, Dec. 2001.
- [11] Bhandarkar, S.M. and A.A. Khombhadia, “Motion-based parsing of compressed video,” in *Proc. of IEEE Intl. Workshop on Multi-Media Database Management Systems*, pp. 80–87, 1998.
- [12] Nang, J., S. Hong, and Y. Ihm, “An efficient video segmentation scheme for MPEG video stream using macroblock information,” *Proc. of ACM Intl. Conf. on Multimedia*, Vol. 1, pp. 23-26, Oct. 1999.
- [13] Sugano, M., *et al.* “A fast scene change detection on MPEG coding parameter domain,” in *Proc. IEEE Intl. Conf. on Image Processing*, Vol. 1, pp. 888-892, 1998.
- [14] Shin, T., *et al.*, “Hierarchical scene change detection in an MPEG-2 compressed video sequence,” in *Proc. of IEEE International Symposium on Circuits and Systems*, Vol. 4, pp. 253-256, 1998.
- [15] Zhang, H. J., C. Y. Low, and S. W. Smoliar, “Video Parsing and Browsing Using Compressed Data,” *Multimedia Tools and Applications*, Vol. 1, No. 1, pp. 89-111, 1995.
- [16] Zhang, H. J., *et al.*, “A video parsing, indexing and retrieval system,” in *Proc. of ACM Intl. Conf. on Multimedia*, 1995.
- [17] Gamaz, N., X. Huang, and S. Panchanathan, “Scene change detection in MPEG domain,” in *Proc. of IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 12–17, 1998.
- [18] Bouthemy, P., M. Gelgon, and F. Ganansia, “A unified approach to shot change detection and camera motion characterization,” *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 9, No. 7, pp.1030–1044, Oct. 1999.
- [19] Lupatini, G., C. Saraceno, and R. Leonardi, “Scene break detection: a comparison,” in *Proc. of IEEE International Workshop on Research Issues In Data Engineering*, pp. 34–41, 1998.
- [20] Gargi, U., R. Kasturi, and S.H. Strayer, “Performance characterization of video-shot-change detection methods,” *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 10, No. 1, pp. 1–13, Feb. 2000.
- [21] “ISO/IEC 14496-2 Information Technology-Generic Coding of Audio-Visual Objects,” *ISO/IEC* 1998.
- [22] Simon A. J. Winder, “ISO/IEC MPEG-4 Video Reference Software,” Version Microsoft-FPDAM1-1.0-000403, Feb. 2002.
- [23] Battista, S., F. Casalino and C. Lande, “MPEG-4: A Multimedia Standard for the Third Millennium. 1,” *IEEE Multimedia*, Vol.6, No.4, pp. 74–83, Oct., 1999.
- [24] Battista, S., F. Casalino and C. Lande, “MPEG-4: A Multimedia Standard for the Third Millennium. 2,” *IEEE Multimedia*, Vol.7, No.1, pp. 76–84, Jan., 2000.
- [25] Brady, N., “MPEG-4 standardized methods for the compression of arbitrarily shaped video objects,” *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 9, No. 8, pp. 1170–1189, Dec. 1999.
- [26] Brady, N., F. Bossen, and N. Murphy, “Context-based arithmetic encoding of 2D shape sequences,” in *Proc. IEEE Intl. Conf. on Image Processing*, Vol. 1, pp. 29–32, 1997.
- [27] Ostermann, J., “Efficient Encoding of Binary Shapes Using MPEG-4,” in *Proc. IEEE Intl. Conf. on Image Processing*, Vol. 1, pp. 295–298, 1998.
- [28] Katsaggelos, A.K. *et al.*, “MPEG-4 and Rate-Distortion-Based Shape Coding Techniques,” *Proceedings of the IEEE*, Vol. 86, No. 6, pp. 1126–1154, June 1998.
- [29] Moffat, A., R. M. Neal, and I. H. Witten, “Arithmetic Coding Revisited,” *ACM Transactions on Information Systems*, Vol. 16, No. 3, pp. 256–294, July 1998.
- [30] Witten, I. H., R. M. Neal, and J. G. Cleary, “Arithmetic coding for data compression,” *Communications of the ACM*, Vol. 30, No. 6, pp. 520–540, June 1987.
- [31] Erol, B. and F. Kossentini, “Automatic key video object plane selection using the shape information in the MPEG-4 compressed domain,” *IEEE Transactions on Multimedia*, Vol. 2, No. 2, pp. 129–138, June 2000.
- [32] Huttenlocher, D. P., G. A. Klanderman, and W. J. Rucklidge, “Comparing images using the Hausdorff distance,” *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 15, pp. 850–863, Sept. 1993.