Chapter 3 Image Clustering and Image Arrangement

透過增添式環場技術(Augmented Panorama)所建立 3D 虛擬博物館，可以自然地融合場景和物體的影像來呈現逼真的虛擬展示效果。然而爲了提供觀賞者與文物良好的互動性，增添式環場往往需要大量的影像資料，譬如一個環物影片(Object Movie)常常需要上百張的影像，倘若一個展場有多個環物影片時，整體的資料量就相當可觀了。這麼龐大的影像資料要如何快速地傳遞到使用者的電腦，同時又能夠根據使用者與文物的互動作即時播放，都是增添式環場應用在博物館網路虛擬展示所會遭遇到的問題。

一般來說，有兩種壓縮方式可以減少環物影片資料量的大小，第一種方式是將每張環物影像以靜態影像的壓縮方式獨立處理，如 JPEG、PNG…等等，然而這樣的方式無法利用環物影像之間的高相似性導致壓縮的效率有限；第二種方式將環物影片視作一般的視訊影片，利用視訊壓縮的方法來處理，如 MEPG-2、H.263…等等，這種方式的優點是利用影像之間的相似性大幅降低檔案的大小。但是利用視訊壓縮的方式來處理環物影片會遇到下列三個主要的問題：1.如何將環物影片從二維的排列關係轉換成一維的影像序列，以方便用來視訊壓縮；2.如何能夠快速地從壓縮後的 bitstream 中解出指定的影像；3.克服觀賞環物影片時隨機存取(random-access)的問題。

爲了克服以上的問題，我們根據一般使用者瀏覽環物影片的習慣，同時考量環物影片壓縮品質與解壓縮時間，提出一套環物影片排列方法，以利於使用目前

最新的 H.264/AVC 視訊壓縮技術壓縮環物影片。

3.1 Problem Formulation

環物影片乃是具有空間上二維關係的影像集合(如圖 3.1)，但是視訊壓縮則是將依照時間順序排列的一維影像序列進行壓縮，因此如何將二維關係的影像集合排列成一維的影像序列，是進行視訊壓縮之前先要解決的問題。

視訊壓縮技術會使用影像間的相似性來提升壓縮的效能，經由 motion estimation /compensation 消除影像之間冗餘，以減少壓縮後檔案的大小。早期的壓縮標準通常只允許參考前/後一個 frame，但 H.264 具有多重參考 frame 運動補償(multiple reference frame motion compensation)的特性(如圖 3.2)，在壓縮一張影像時不同的 macroblock 可以參考不同的 frame，藉此提升影像的壓縮品質。然而要達到這樣的目的編碼器與解碼器必須儲存多張的參考影像並維持相同的順序 ，如此會造成解壓縮時 CPU 或記憶體額外的負擔。因此壓縮影像排列順序的不同會造成壓縮時參考影像的不同，會影響壓縮後影像品質的好壞，如果能把越相似的影像排列在一起，藉由 multiple reference frame motion compensation 的特性能使壓縮之後的影像品質更好，所以要找出一個比較好的排列順序來壓縮環物影片。

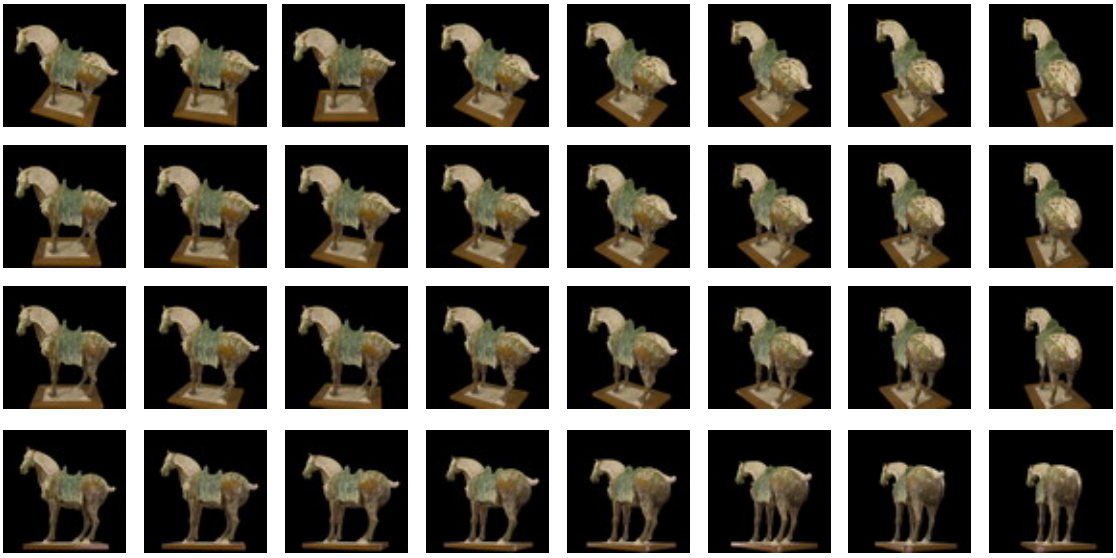

圖 3.1 環物影片的例子(資料來源：歷史博物館)

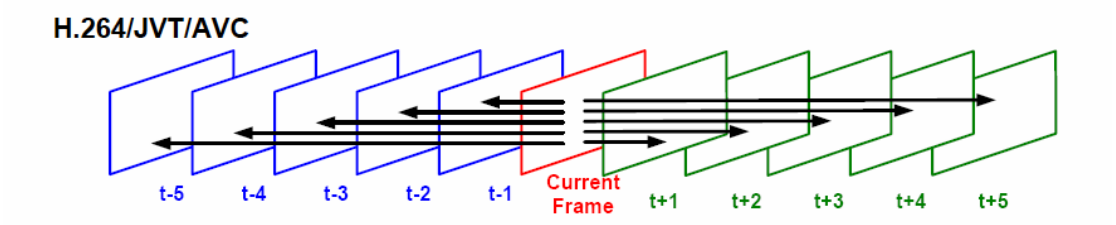

圖 3.2 H.264 multiple reference frame motion compensation

有別於一般影片依照時間順序播放的方式，環物影片的播放是根據使用者觀看的角度，然後從 bitstream 解出相對應的影像。這樣的過程中除非每張影像均採用靜態影像的壓縮法，或是該張影像恰好爲 I-frame，不然大多數的情況下無法直接解出所需要的影像。因此除了 I-frame 之外，不管是 P-frame 或 B-frame 都需要參考的影像已經被解壓縮之後才有辦法跟著解出。也就是說當解壓縮某一張影像時，如果參考影像不存在則必須先解出來後，才能進行 motion estimation / compensation。譬如 H.264 reference model software 預設前面 5 張影像都可以當作所參考影像，所以爲了正確還原影像必須維持跟編碼端相同的參考順序，當解壓縮某張影像時它前面的 5 張影像都必須先被解壓縮並存放在參考序列裡

(reference list)，而且不管它是不是有參考到這麼多張影像。這些額外需要解壓縮的影像就會造成播放上的延遲，而這個延遲會受到兩個因素影響，第一是 I-frame 的個數，當 I-frame 的數量越多的時候，延遲時間就越短但 bitstream 的資料量也會增加；第二是 bitstream 中影像排列的方式是否符合使用者操作環物影片的模式，如果使用者觀看的影像排列越相近則延遲時間也就會縮短，但我們很難事先預測使用者觀賞的方式，而且每次觀看的次序也不見相同，因此要有效衡量排列的好壞是一件困難的工作。

以 QTVR 所採取的方式爲例，環物影片的排列是以行優先 (row major)，具有相同垂直角度(tilt)的影像會優先串接在一起，這樣的優點是水平轉動環物時，可以獲得平順的播放效果。但是觀賞者突然從某一列跳到另一列的時候，可能因爲 P 或 B frame 的出現造成播放延遲的現象，如圖 3.3，假設黃色框起來的影像表示爲 I-frame 其餘均爲 P-frame，目前觀看爲藍色框起來的影像，而使用者下一張要觀看的爲紅色框起來的影像，此時，解碼端必須從 I-frame 開始先解出欲觀看影像前面 5 張的影像，接著才能解出使用者欲觀看的影像。

本文爲了解決上述的問題，同時配合 H.264 rate-distortion 的機制，根據一般使用者在瀏覽環物影片時會先觀看對應於物體某一面之影像與其附近的影像之習慣，提出一套環物影片排列方法。利用分群的技術(clustering)配合 I-frame 將環物影片分成若干群，然後再根據每張影像對群內其他影像執行 motion compensation 的好壞將影像排成一維序列。如此一來，把越相似的影像排列在一起會使得壓縮之後的影像品質會更好。

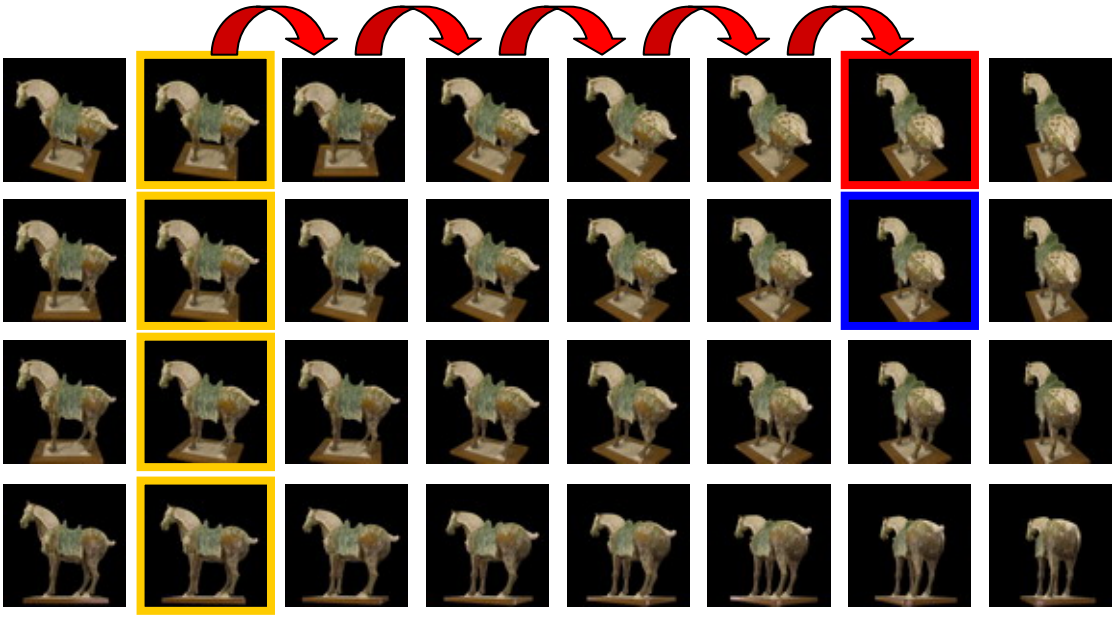

圖 3.3 播放延遲示意圖

(黃色代表 I-frame、藍色代表目前觀賞的 frame、紅色代表下個時間觀賞的 frame)

3.2 Image Clustering

爲了透過 H.264 壓縮環物影片，必須先將環物影片的二維關係透過適當的方式排列成一維序列。假設將環物影片中每張影像都壓縮成 I-frame，不管影像序列如何排列都不會影響到壓縮的品質或播放時解壓縮的速度，但如此一來 bitstream 的大小勢必增加許多，不利於數位博物館的網路虛擬展示。相對來說，若序列的 I-frame 的個數太少，則 P-frame/B-frame 解壓縮時因爲參考影像所造成的播放延遲勢必會增加。在此兩難的情況下，如何在 I-frame 的個數與環物影片播放的順暢度取得平衡也是需要解決的問題。

在 H.264 如果壓縮時影像中每個 slice 或 macroblock (MB)都採用 intra prediction 的話，則該張影像爲 I-frame。H.264 進一步支援一種特別的 I-frame 形式稱之爲 instantaneous decoding refresh (IDR) picture，當 IDR 影像出現時會把目前在 reference list（參考影像序列）裡的影像會標示成“unused for reference”，則在 IDR 影像之後的 P-frame/B-frame 不能使用 IDR 影像之前的影像當作參考影像。我們利用此一特性，在選定適當的 I-frame 個數之後，把每個 I-frame 設定成 IDR picture，則每個 I-frame 之後所帶領的數個 P-frame 可以視爲一個獨立的群組，於是就把整個環物影片分成若干個群組。接下來討論如何透過 clustering 使得每個群組內的影片彼此之間具有很高的相似性，讓後續壓縮時 ME / MC 能夠獲得較好的結果進而增加壓縮的效能。

在 clustering 的過程中，首先我們先由環物影片中選定 M 張影像當作 IDR picture，然後經由相似性評估(similarity measure)，在剩下的環物影片中找尋與這些 IDR pictures 相似的影像。similarity measure 的選擇我們採用 H.264 motion

estimation / compensation 的結果來評估。將非 IDR picture 的環物影像對 IDR picture 進行 motion compensation 之後的 residual 當作跟該 IDR picture 之間的相似值，所有 MB 的 residual 加起來的和越小表示該張影像跟 IDR picture 的相似度越高，把相似度高的影像分成同一群，將來壓縮時這些相似性高的影像就有可能互相參考，而得到較佳的壓縮結果。詳細的演算法如下：

1. 從環物影片中選定 *M* 張影像當作 IDR picture。
2. 根據下列公式(1)計算某影像 *n* 與所有 IDR picture 的相似度 J_n，找出相似度最大的 IDR picture *k*，則影像 *n* 屬於第 *k* 群。MC_k 爲 H.264 計算 motion estimation 的方式(採用跟 reference software 一樣的方式，motion estimation 爲 full search)。若該群的個數超過預設的值 *I*，則將該群中相似度最低的影像移出該群，並從執行第二步找出次佳的 IDR picture。

$$J_n = \min_k \sum_{mb_i \in n} MC_k(mb_i), n \notin M \qquad (1)$$

3. 依序執行完所有影像，使得每一張影像都會與某 IDR picture 同一群，且每一群包含的影像個數相同。

如此一來，由於相似的影像都排列在一起，所以相互參考的效果也會變好。同時由於相近的影像都歸類於同一群，當使用者瀏覽這群影像中的某一張影像時，這張影像與該影像解壓縮所需要之 IDR picture 與 reference frames 都已經解好，當使用者瀏覽附近其他的環物影像時，便有機會可以重複使用這些已經解好的影像，而不需要由 IDR picture 再次重新解壓縮。

3.3 Image Arrangement

在上一節中，我們已經將環物影片根據與 IDR picture 的相似性分成 M 個群組，接下來將探討如何將每個群組的影像排列成一維的序列。將相似性高的影像分成同一群有助於 H.264 MC/ME 的結果，爲了確保能夠得到最好的結果，在排列影像時我們考慮對於其他的影像能夠有較佳 MC/ME 結果的影像優先排列，也就是說，排列越前面的影像對於後面的影像能夠提供良好的 MC/ME 結果。以圖 3.4 爲例，紅色影像爲序列的第一張影像也就是 IDR picture，假設黃色影像對於除了紅色影像之外剩下的影像能夠提供較佳的 MC/ME 結果，也就是說以黃色影像當作參考影像去預測其他影像所得到的 residual 和爲最小，所以排列在序列的第二個位置；綠色影像則對於紅色和黃色之外的剩下影像執行 MC/ME 之後所得到 residual 的和爲最小，所以排列在序列的第三個位子，以此類推直到完成整個序列爲止。

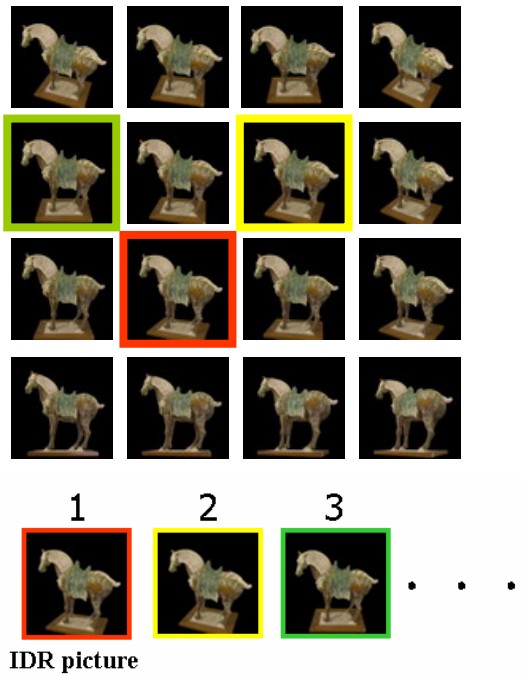

圖 3.4 根據每張影像對於其他影像執行 ME/MC 的結果來決定排列的先後次序

詳細的演算法如下

1. 在某一個群組 *k* 中假設所有影像屬於集合 E，首先將 IDR picture 提出來當成壓縮序列 L 的第一張。接下來從 E 中選定一張影像 e_n，計算其他影像以該張影像爲參考影像 MC/ME 之後的 residual，並將所有的 residual 加總起來得到 R_n，如公式(2)。

$$R_n = \sum_{i=1, i\neq n}^{N} MC_n(i), i \in E \qquad (2)$$

2. 對集合 E 中所有影像依序執行第一步，將 R_n 最小的影像移出集合 E 並放入序列 L 中。

3. 反覆執行 1 跟 2 直到所有影像都放入序列 L 中。

如此一來，能夠提供較佳的 MC/ME 的影像都會排列在前面，所以參考的效果也會變好，此外由於每一群包含的影像個數相同，所以我們也可預期最長解壓縮時間，在實驗中我們將 IDR picture 的個數訂爲 9 使得檔案大小與環物影片播放時的順暢度之間能夠取得平衡。

3.4 Rate Distortion Optimization (RDO)

在這個小節中，將進一步描述 H.264 的 motion compensation / motion estimation 的方式，並探討應用在 clustering 可能遇到的問題與解決的方式。H.264 在壓縮時若控制的參數(e.g. motion estimation search area、quantization parameter, etc.)保持不變的話，根據影像內容的不同每個 macroblock (MB)壓縮後產生出來的 bits 數目也會不同，因此爲了在維持一定的 bit rate 和最小化 distortion 之間取得平衡，編碼器必須採用 rate-control 機制控制在一定 bit rate 之下盡量提高視訊的品質。此外，H.264 提供了許多不同的 MB 模式，像是 INTER 16x16、INTER 16x8、INTER 8x16、INTER 8x8、INTRA 16x16、INTRA 4x4，透過 rate distortion optimization (RDO)針對每個 MB 的內容選擇適當的模式，在 bit rate 和 distortion 之間取得平衡。然而一開始 H.264 標準裡並沒有規定 RDO 實際的方式，但後來 Li et. al.所提出的方式[22]被 JVT reference model software 採用，他們將原本 constraint optimization 問題利用 lagrangian multiplier 轉換成 unconstraint problem，如公式(3)。

$$J = D + \lambda \times R \qquad (3)$$

J 爲 RD 的最小值，λ 爲 lagrangian multiplier，R 爲每個 MB 的 rate，D 爲每個 MB 的 distortion。在 H.264 的 motion estimation 和 mode decision 都是採用上述的方式決定。Motion estimation 利用下面的公式(4)去找出 motion vector 使得 cost J_{MV} 爲最小，

$$J_{MV} = SAD_{MV} + \lambda_{Motion} \times R_{MV} \qquad (4)$$

SAD_{MV} 爲做過 motion compensation 之後差的絕對直總和，R_{MV} 爲 motion vector

bits，λ_{Motion} 爲 motion Lagrangian multiplier.。在決定了 motion vector 之後，再計算每種 inter mode 的 cost J_M ，如(5)，J_M 由每個 MB 壓縮所需要的 bits，和重建後的 MB 跟原始 MB 的 sum of squared differences (SSD)所決定，測試所有的 mode 後選擇最小 J_M 的 mode。

$$J_M = SSD_M + \lambda_{Mode} \times R_M \qquad (5)$$

上述最佳化的過程，λ 由下列公式(6)、(7)決定，

$$\lambda_{Mode} = 0.85 \times 2^{(QP-12)/3} \qquad (6)$$

$$\lambda_{Motion} = \sqrt{\lambda_{Mode}} \qquad (7)$$

不管在 clustering 搜尋 nearest neighbor 或是轉化成一維序列，均採用跟上述相同的方式評估影像之間的相似性，使得排列後的結果使用 H.264 壓縮時能得到最佳的效果。但這樣做的缺點是需要花費大量運算時間，根據我們的測試平均一張影像執行 MC/ME 所需要的時間要 20 sec，因此光一張影像要決定屬於哪一群就需花費 13 分鐘。爲了能節省時間，在 clustering 階段，限定每張影像僅跟最鄰近的 8 張 IDR picture 做計算計算。在排列一維序列時，只要紀錄下找第一張影像時所計算每張影像跟其他影像 MC 之後的結果，之後根據第一次的結果就可以找出第二、三…影像，不需要再做額外計算。