

基於時間空間可調性之可擴展高效率視訊編碼自適應串流傳輸演算法

王姿穎、*許子衡

南臺科技大學資訊工程系

*hsuth@stust.edu.tw

摘要

可擴展高效率視訊編碼 (Scalable High-Efficiency Video Coding, SHVC) 最大的特色在於能支援多層的視訊解析度、畫面更新率和畫面品質,使不同的接收者能依據本身條件和需求,來播放不同位元之視訊串流;可擴展高效率視訊編碼可以有效的提升視訊可擴展性,相對的運算複雜度也大幅增加。此外,可擴展高效率視訊編碼相關編解器及播放器設計較為複雜,導致支援可擴展高效率視訊編碼影片的硬體播放裝置並不普及。本論文主要針對可擴展高效率視訊編碼,探討網路頻寬擁塞時所導致的視訊傳輸品質不佳狀況,並發展基於時間空間可調性之可擴展高效率視訊編碼自適應串流傳輸演算法,進行效能改進。實驗結果顯示本論文所提之時間空間可調演算法可隨影片性質動態調整傳輸視訊幀,可以提供給使用者更好的影片觀賞品質。

關鍵詞: 可擴展高效率視訊編碼 (SHVC)、高效率視訊編碼 (HEVC)

A Scalable High-Efficiency Video Coding Adaptive Streaming Transmission Algorithm Based on Time and Space Adjustment

Zi-Ying Wang, *Tz-Heng Hsu

Dept. of Computer Science and Information Engineering, Southern Taiwan University of Science and Technology

Abstract

Scalable High-Efficiency Video Coding (SHVC) can support multi-layer video encoding, including temporal scalability, spatial scalability and quality scalability; Different receivers can play different videos based on their own conditions and needs, i.e., network conditions and quality requirements. SHVC can effectively improve video scalability, however, the relative computational complexity is also greatly increased. In addition, the design of encoders and players related to SHVC is relatively complex, resulting in the fact that hardware playback devices that support soft-scalable high-efficiency video coding videos are not popular. The goal of this paper is to investigate the coding efficiency of existing SHVC codecs and the compatibility of SHVC encoded videos, discusses the poor quality of video transmission caused by network bandwidth congestion, and develops adaptive streaming of scalable high-efficiency video coding based on temporal and spatial transmission adjustment algorithm to enhance the user watching experience. Experimental results show that the proposed temporal and spatial adjustment algorithm can dynamically adjust the transmission video frame according to the network conditions, which can provide users with better video viewing quality.

Keywords: Scalable High-Efficiency Video Coding(SHVC), High Efficiency Video Coding(HEVC)

壹、前言

隨著社群網路服務 (social network service, SNS) 的普及, 愈來愈多的使用者上傳自製內容 (user generated content, UGC) 到社群網站上, 並且在社群網路上進行影音推薦及意見分享。社群網路如 Facebook 在 2015 年第一季宣布使用者每日影片觀看次數已超過 40 億次, 已有超過 8 萬部使用者上傳的影片嵌入在第三方網站。各種高畫質影音服務的增加, 高解析度視訊格式, 如 4K×2K 或 8K×4K 解析度的出現, 以及行動裝置和平板電腦視訊傳輸服務的流量大幅增加, 視訊傳輸的流量需求使得現在的網路傳輸骨幹面臨頻寬不足的考驗。高效率視訊編碼 (high efficiency video coding, HEVC/H.265) 受到國際電信聯盟 (international telecommunication union, ITU) 的認可, 發佈為 ISO/IEC 23008-2 MPEG-H Part 2 and ITU-T H.265 標準, 其主要目的就是要進一步降低影片所需的資料量, 以降低視訊儲存與傳輸的成本[1]。高效率視訊編碼只要 MPEG-4/H.264 AVC[2]一半資料量, 即可展現等於或優於 H.264 畫質的視訊內容[3]。高效率視訊編碼利用許多新壓縮技術來達到高編碼效率, 使其適用於超高解析度的視訊, 可支援 4K 解析度超高畫質電視 (ultra high definition television, UHD TV), 最高解析度可達到 8192×4320 (8K 解析度), 將來可能全面取代 MPEG-4/H.264 編碼技術。然而, 高效率視訊編碼高壓縮率來自遠較 MPEG-4/H.264 AVC 複雜的視訊運算邏輯, 大幅增加編解碼運算複雜度[4–[5]。

長期演進技術 (long term evolution, LTE) 及 5G 等無線傳輸技術來提高無線鏈路的頻寬, 來自行動用戶的視訊流量需求正在快速的消耗無線鏈路的容量。異質行動裝置的發展, 使得高解析度、高頻寬和大規模的數位視訊串流傳輸需求快速增加。目前, 大部份的數位內容供應商使用固定非適應性 (non-adaptive codec) 的編碼方式來進行不同裝置的視訊編碼, 這種方式降低了視訊傳輸的彈性。傳統上針對不同視訊解析度的行動視訊傳輸要求, 視訊伺服器會儲存多個不同檔案, 每個檔案儲存不同傳輸率 (bit rate) 的相同視訊內容, 由於時間 (temporal) 及空間 (spatial) 的冗餘 (redundancy), 會佔用很多儲存空間。針對不同種解析度裝置產生不同傳輸率的視訊編碼檔案, 造成多版本不同解析度影片維護上的困難。

為了適應異質網路的傳輸與不同裝置的使用者, 進一步發展出可擴展高效率視訊編碼技術[6]。可擴展高效率視訊編碼是由 Joint Collaborative Team on Video Coding (JCT-VC) of ISO/IEC JTC 1 SC 29/WG 11 MPEG and ITU-T Q6/16 VCEG 工作小組開發。藉由採用高階描述語法 (High-Level Syntax only, HLS-only), 通過圖層間 (inter-layer) 參考圖片處理模組, SHVC 不需要改變單層高效率視訊編碼 (single-layer HEVC core) 核心的區塊層級的編碼邏輯 (block-level coding logic), 即可進行可擴展視訊編碼 (scalable video coding, SVC), 大幅簡化實作複雜度[7]。可擴展高效率視訊編碼架構設計能夠使用多個重用的單層 HEVC 編解碼器來實現 SHVC 編解碼, 其中增加了層間參考圖像處理模組, 包括紋理和運動重取樣和顏色映射。SHVC 主要是由多層 HEVC 組成, 包括基礎層 (base layer, BL) 和多個加強層 (enhancement layer, EL)。在可擴展高效率視訊編碼解碼中, 層間預測 (inter-layer prediction, ILP) 採用分層為基礎, 如圖 1, 加強層於編碼時, 除了可參考時間軸上已編碼的畫面, 另外也參考來自基礎層的層間預測資訊, 包含紋理資訊 (texture)、移動向量 (motion vector, MV) 與殘餘值 (residue), 來提高加強層的編碼效率。

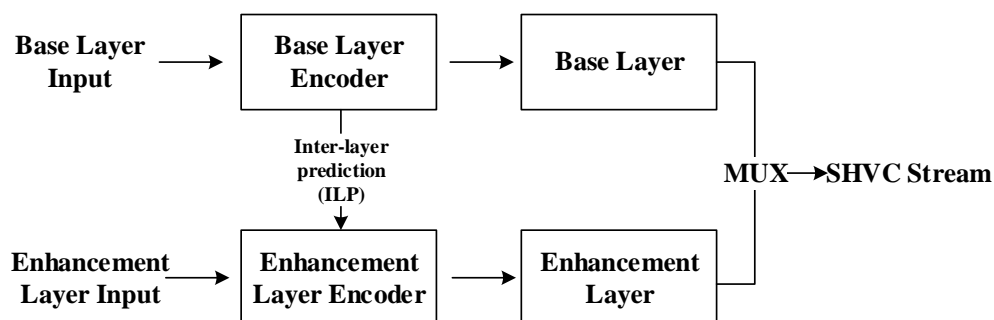


圖 1 SHVC 編碼架構圖[1]

藉由超高解析度螢幕及 5G 行動寬頻上網，使用者可以隨時隨地觀看高品質的視訊串流服務，伴隨而來的是更多電源消耗。此外，同一行動裝置在不同的時間和地點，由於訊號強度、其他用戶所佔用的頻寬及無線鏈路狀態的動態變化，導致網路可用鏈路容量隨著時間改變。使用者觀看串流視訊時需要較長的視訊緩衝時間（buffering time），同時不穩定的無線傳輸以及使用者的移動行為，容易導致觀看時的間歇性中斷（intermittent disruptions）。

為了適應異質網路的傳輸與不同裝置使用者，本論文提出基於時間空間可調性之可擴展高效率視訊編碼自適應串流傳輸演算法，探討網路頻寬擁塞時所導致的視訊傳輸品質不佳狀況，進而藉由頻寬預測來發展基於時間空間可調性之可擴展高效率視訊編碼自適應串流傳輸演算法，進行效能改進，滿足使用者視訊播放需求，給予使用者良好的視訊品質。可擴展高效率視訊編碼改善了傳統可擴展視訊編碼實作上的複雜度，藉由重覆利用現有單層高效率視訊編碼技術核心，減少可擴展高效率視訊編碼實現的複雜度。除了已經由高效率視訊編碼提供的時間可擴展性之外，可擴展高效率視訊編碼還提供空間，信噪比，位元深度和色域可縮放性等功能。可擴展高效率視訊編碼雖然能有效的提升視訊編碼效能，但運算複雜度也大幅增加，本論文也會探討現有 SHVC 編解碼器的效能，進行編解碼實驗步驟、了解 SHVC 影片的相容性。

貳、相關研究探討

一、高效率視訊編碼（High Efficiency Video Coding）

視訊傳輸的流量需求使得現在的網路傳輸骨幹面臨頻寬不足的考驗，高畫質視訊串流需要高位元率壓縮演算法來進行壓縮及傳送；高效率視訊編碼受到國際電信聯盟的認可，發佈為 ISO/IEC 23008-2 MPEG-H Part 2 and ITU-T H.265 標準，其主要目的就是要進一步降低影片所需的資料量，以降低視訊儲存與傳輸的成本。高效率視訊編碼被設計用來改善 264/MPEG-4AVC 位元壓縮率，高效率視訊編碼著重在兩個關鍵問題：改善視訊解析度和視訊平行處理架構的設計。高效率視訊編碼標準要實現多個目標，包括編碼效率、資料遺失容錯性（data loss resilience）以及使用平行處理架構進行編碼。以下針對高效率視訊編碼的細節進行描述：

視訊編碼層（video coding layer）：高效率視訊編碼的視訊編碼層採用跟 H.261 視訊壓縮標準相同的混合式架構-畫面間/畫面內預測和 2-D 轉換編碼（inter/intrapicture prediction and 2-D transform coding）[8]。高效率視訊編碼演算法產生位元資料流（bitstream）的過程描述如下：首先，每一畫面被分成塊狀區域（block-shaped regions），一個視訊影片序列（video sequence）的第一張畫面（picture）會先使用畫面內（Intrapicture）預測編碼，在同一畫面內進行從區域到區域的空間預測編碼。之後，視訊影片序列的所有剩餘畫面，則採用畫面間（interpicture）預測編碼模式來進行區塊編碼，畫面間預測（interpicture prediction）模式會根據編碼器所選擇的參考畫面（reference picture），計算出目前畫面的移動向量，進行編碼區塊移動預測。最後，編碼器和解碼器產生藉由移動向量來進行移動補償（MC）[1]，產生殘差信號（residual signal）資料，進行編碼及資料傳輸。高效率視訊編碼畫面內或畫面間預測經由線性空間轉換（linear spatial transform）的殘差信號，代表原始畫面區塊與其預測區塊之間的差異量，之後對轉換係數（transform coefficients）進行縮放、量化、熵編碼，並與預測資訊一起進行編碼及資料傳輸。編碼器會複製解碼處理循環，產生預測後數據。因此，量化後的轉換係數被逆轉換（inverse transformed）到接近原始訊號，再與預測殘差相加，送入環路濾波器（loop filters）來平滑區塊處理上的失真。還原後的最後一張畫面被儲存在解碼器畫面緩衝記憶體（decoded picture buffer, DPB）中，用於後續畫面的解碼預測處理。

高效率視訊編碼的特性描述如下：（1）編碼樹單位（coding tree unit, CTU）和編碼樹區塊（coding tree block, CTB）結構：高效率視訊編碼編碼器選擇的巨圖塊（macroblock, MB）大小，可以比傳統 H.264 所採用的巨圖塊大。編碼樹單元（CTU）包含了一個亮度（luma）編碼樹區塊和相應的彩度（chroma）編碼樹區塊以及語法元素（syntax elements）。亮度編碼樹區塊的尺寸可選擇為 $L \times L$ ， $L = 16, 32$ 或 64 個像素，較大的區塊尺寸可達到較好的壓縮率。高效率視訊編碼支援將編碼樹區塊分割成四叉樹（quadtree）

結構的更小區塊，在進行畫面內/間預測時，可以預測到更精確的位置，降低資料量。(2)編碼單元(coding unit, CU)和編碼區塊(coding block, CB)：編碼樹單元的四叉樹語法指定亮度和彩度編碼區塊(CBs)的大小和位置。一個編碼樹單元分割成亮度和彩度編碼區塊信號。一個亮度編碼區塊(CB)通常和兩個彩度編碼區塊，形成一個編碼單元。一個編碼樹區塊可以含有一個或多個以上的編碼單元(CUs)，每個編碼單元具有相關聯的預測單元(prediction units, PUs)和一個轉換單元(transform unit, TU)樹。(3)預測單元(prediction units, PU)和預測區塊(prediction block, PB)：無論是畫面間或畫面內預測來進行畫面區域的編碼，都由編碼單元根據基本預測型態(prediction-type)決定如何編碼；亮度和彩度的編碼區塊可以被進一步分割成較小的區塊，並且根據亮度和彩度預測區塊(PBs)進行預測。高效率視訊編碼支援的可變預測區塊尺寸從 64×64 到 4×4 大小。(4)轉換單元和轉換區塊(transform block, TB)：預測殘差(prediction residual)使用區塊轉換(block transform)進行編碼。亮度跟彩度的編碼區塊殘差可以進一步分割成更小的亮度轉換區塊(TBs)。利用類似於離散餘弦轉換(DCT)的整數基礎函數定義方形轉換區塊，尺寸可以為 4×4, 8×8, 16×16 和 32×32 大小。針對 4×4 亮度畫面內(intrapicture)預測殘差的轉換，可使用離散正弦轉換(discrete sine transform, DST)來進行編碼。(5)移動向量信號(motion vector signaling)：使用進階移動向量預測(advanced motion vector prediction, AMVP)，從相鄰的預測區塊和參考畫面選擇幾個最可能的候選區(candidates)進行預測，允許使用繼承自時間上或空間上相鄰預測區塊的移動向量來進行預測。相對於 H.264 / MPEG-4 AVC，也可採用改良的跳躍(skipped)和直接移動推論(direct motion inference)方式進行預測。(6)移動補償(motion compensation)：移動向量可使用四分之一樣本精度(quarter-sample precision)，使用 7-tap 或 8-tap 濾波器用於分數樣本位置(fractional-sample positions)進行訊號插補(Interpolation)。與 H.264 / MPEG-4 AVC 相似，可使用多個參考畫面進行移動補償。如同在 H.264 / MPEG-4 AVC 中，縮放(Scaling)和偏移(Offset)操作可採用加權預測方式的來預測信號。(7)畫面內預測(intrapicture prediction)：相鄰區塊(adjacent blocks)的解碼邊界樣本可被用來當作參考資料，進行區域中的空間預測。畫面內預測(intrapicture prediction)支援 33 種定向模式加上平面(planar)和 DC 預測模式。根據先前解碼相鄰預測區塊，決定最後畫面內預測模式，導出最適合的預測方向模式進行編碼。(8)量化控制(quantization control)：與 H.264/MPEG-4 AVC 相同，高效率視訊編碼採用均勻重建量化(uniform reconstruction quantization, URQ)及量化縮放矩陣(quantization scaling matrices)來支援各種轉換區塊大小(transform block sizes)的量化。(9)熵編碼(entropy coding)：高效率視訊編碼採用文本自適應二進制算術編碼(context adaptive binary arithmetic coding, CABAC)進行熵編碼。相似於 H.264/MPEG-4 AVC 的 CABAC 方案，但做了一些改進，提高其吞吐量速率和壓縮性能，並減少記憶體需求。(10)迴路中除塊濾波(in-loop deblocking filtering)：類似於 H.264/MPEG-4 AVC 使用的畫面間預測環路內(interpicture prediction loop)除塊濾波。然而，高效率視訊編碼簡化除塊濾波決策和濾波過程，使其更易於進行平行處理。(11)取樣自適應偏移(sample adaptive offset, SAO)：畫面間預測環路內除塊濾波(deblocking filtering)後採用非線性振幅映射(nonlinear amplitude mapping)方式來有效降低取樣失真，其目的是可以用更佳方式的重構原始信號。

二、可擴展高效率視訊編碼(Scalable Extensions of the High Efficiency Video Coding)

近年來許多學者研究可擴展視訊編碼並發表許多成果，然而可擴展視訊編碼並不如傳統視訊編碼技術般被廣泛使用。主要的原因之一在於傳統可擴展視訊編碼實作上的複雜度，阻礙了可擴展視訊編碼技術的普及[9]。可擴展高效率視訊編碼是由 JCT-VC of ISO/IEC JTC 1 SC 29/WG 11 MPEG and ITU-T Q6/16 VCEG 工作小組開發。在高效率視訊編碼第二版草案中，討論了可擴展高效率視訊編碼技術，Annex F 包含了 SHVC 高階描述語法，Annex H 包含了 SHVC 解碼器操作和一致性描述。可擴展高效率視訊編碼有鑑於以往的經驗，試圖重覆利用現有單層高效率視訊編碼技術核心，如圖 2，以減少可擴展高效率視訊編碼實現的複雜度。藉由採用高階描述語法，通過圖層間參考圖片處理模組，SHVC 不需要改變單層高效率視訊編碼核心的區塊層級的編碼邏輯，即可進行可擴展視訊編碼，大幅簡化實作複雜度[10]。可擴展高效率視訊編碼提供了視訊多層編碼機制，每個編碼層表示相同畫面但不同品質的視訊內容[11]。底層具有

最低的品質，一個或多個增強層可以透過參考基礎層進行解碼，強化視訊呈現品質；播放端接收愈多編碼子層，視訊呈現品質愈佳。當進行網路視訊串流時，如網路頻寬不足時，伺服器可僅傳送基礎層視訊，保持最低播放效果。與傳統非可擴展視訊編碼技術比較，可擴展視訊編碼可以提供更好的速率控制（rate control）機制，讓視訊串流不會突然因為頻寬不足，而嚴重下降播放品質[12]。

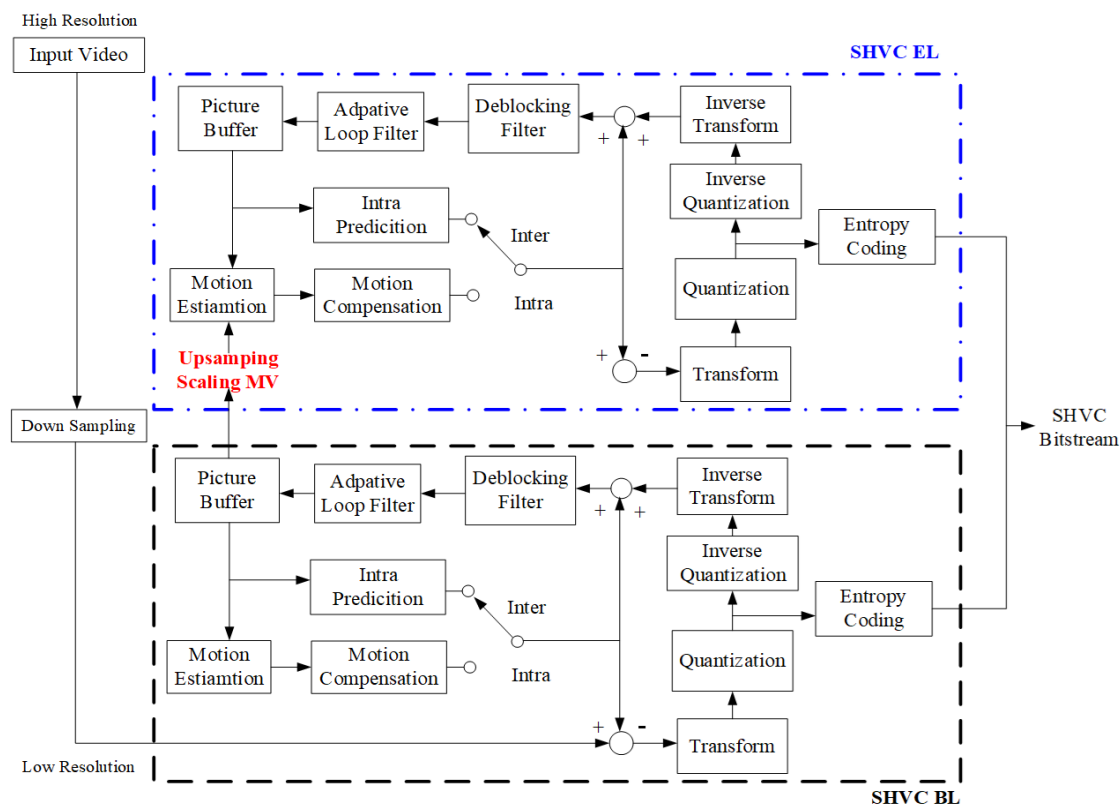


圖 2 SHVC 編碼器細部架構圖[13]

可擴展視訊編碼通常花費更多的位元資料（bits）記錄額外資訊，才能達到傳統視訊編碼技術的視訊品質[14]。可擴展視訊編碼定義了不同擴展性編碼技術，主要為：（1）時間擴展性（temporal scalability），調整畫面幀播放速率，（2）空間擴展性（spatial scalability），調整畫面幀解析度，及（3）品質擴展性（quality scalability），調整畫面幀量化壓縮品質。近年來，由於超高畫質（UHD）及高畫質（HD）影音內容的出現，可擴展高效率視訊編碼支援了新型態的擴展性編碼技術，包含影像位元深度擴展性（bit depth scalability）及色飽和度擴展性（color gamut scalability）編碼技術。在可擴展高效率視訊編碼解碼中，層間預測採用分層為基礎，加強層於編碼時，除了可參考時間軸上已編碼的畫面，另外也參考來自基礎層的層間預測資訊，包含紋理資訊、移動向量與殘餘值，來提高加強層的編碼效率。由參考層（reference layer）重建的畫面幀可被當做長期參考畫面幀（long-term reference picture），長期參考畫面幀會被分配一個參考索引值（reference index），連同其它時間當前層中的參考畫面幀放入一參考畫面幀清單（reference picture list）中。然後，藉由設定參考索引（ref_id）語法元素（syntax element）的值來對應層間預測參考畫面，進行預測單元解碼[15-16]。

Le 等學者提出一種在毫米波（mmWave）通訊系統中進行即時自適應視訊串流的方法[17]，它包括兩個部分：（1）通過 mmWave 802.11ad 60 GHz 無線鏈路的即時視訊串流；（2）使用可擴展高效率視訊編碼進行 360 度視訊解碼、後處理和顯示。mmWave 通信是一種高速無線技術，可以提高視訊傳輸的容量。高速即時視訊串流的同步是確保服務質量（QoS）的大問題，特別是行動裝置正在移動時。因此，作者將可擴展高效率視訊編碼內的基礎媒體層通過 mmWave 傳輸到行動裝置。實驗結果顯示通過 mmWave 傳輸機制，可以確保高解析度即時視訊串流傳輸。在[18]，Singhal 等學者提出了一種系統架構，可以基於

網路頻寬和最終使用者設備規範，通過無線網路自適應地動態編碼及傳輸超高畫質（UHD）視訊內容。作者提出一個位元率自適應的參數速率模型，在給定頻寬條件下進行自適應地動態編碼及傳輸，以改善用戶體驗。接收器把網路通道條件和設備功能傳送給發送器，發送器自適應地改變發送的視訊串流位元率。通過自適應編碼方式，即使在較差的網路傳輸條件下，也可以確保將視訊傳送給使用者戶並具有可接受的視訊播放品質。

Rakngan 等學者針對可擴展高效率視訊編碼視訊串流應用，提出了一個動態視訊內容自適應媒體感知網路元件（media aware network element, MANE）架構的實現[19]。MANE 使用基於空間可調性的自適應傳輸機制，依監控的網路頻寬變化，動態調整要傳輸的媒體層。實驗討論在變動的網路頻寬條件下，可擴展高效率視訊編碼中隨著網路變動而調整播放品質和解碼功耗。作者所提的 MANE 架構可以根據實際網路頻寬變化，在各個媒體層之間快速切換。由於所需層切換次數減少，MANE 架構表現良好。在[20]，Yang 等學者實現了混合單播和多播自適應視訊串流傳輸系統，可以同時滿足單播和多播用戶的需求。作者使用可擴展高效率視訊編碼產生不同位元率的高質量視訊串流，提出自適應算法動態分配單播和多播用戶的位元率，獲得最佳的平均收看品質。實驗結果顯示與傳統方法相比，該演算法在各種頻寬模型下均能有效提高系統的 QoE（體驗品質）。

參、論文演算法

一、視訊串流網路傳輸架構

當多媒體串流物件在網路中傳遞時，網路頻寬下降和網路封包遺失，是造成使用者終端裝置播放品質不良主要的原因。傳統的非可調式視訊編碼（non-scalable video coding）方法只有考慮到將預先決定好的影片品質，編碼成特定的位元傳送率。當編碼位元率（encoding bit-rate）確定並編碼完成後，就不再改變視訊本身的品質。使用非可調式視訊編碼的方式時，使用者觀賞到的影片品質會有以下兩種情況：（a）只有當客戶端與伺服器端連線頻寬大於或等於影片編碼位元率時，使用者觀賞到的影片品質是最好的。（b）而當兩端的連線頻寬小於影片編碼位元率時，使用者觀賞到的影片品質開始變的很差，因為網路可能處於壅塞狀況。明顯的，非可調式視訊編碼並不能依不同的網路狀況進行傳輸品質的調適（Adaptation）。

可擴展視訊編碼的優點之一是對於特定的網路視訊呈現裝置不需要把串流的所有部份都傳送，只需依裝置特性傳送所需的部份，滿足該裝置的呈現需求即可。當多媒體串流系統支援 SVC 可擴展視訊編碼，可同時傳送視訊串流給多個不同能力的網路視訊裝置，把可擴展視訊串流的基本層傳給所有的裝置，低運算能力的網路視訊呈現裝置只接受基本層串流，而高效能的裝置可接收高品質的增強層串流。此外，支援 SVC 可擴展視訊編碼的視訊串流系統，可以根據不同裝置的網路接收狀況，來傳送相對應品質的視訊串流。因為可擴展視訊編碼的每一層有不同的重要性，從基本層開始到其他多層的增強層的重要性遞減。在網路傳輸最差的情形下，只需要傳送基本層的資料，若在接收環境條件允許下，每多一層增強層重建回來的視訊的品質就會增加。在 SVC 中，如果同空間層或粗糙（coarse granular）的 SNR 層之間有多餘的資料，可以額外的用層與層之間的預測來壓縮，其中包括移動參數的預測機制。相對應的網路抽象層（network abstraction layer, NAL）單位包含了移動的資訊和網路的資料，這些基本圖層的重建品質可以藉著額外的漸進精煉切片（progressive refinement slices）編碼技術來加強。依照目前的圖層切片（slice）資料及網路抽象層（NAL）資訊，SVC 增強層可以任意調適來符合網路頻寬的變化，支援細紋（fine granular）視訊品質的可調適應性。對於一個低運算能力的網路視訊呈現裝置，可以藉著從 SVC 位元串流中丟棄 NAL 單位，達到資料傳輸及呈現品質的調整。本論文的基於時間空間可調性之可擴展高效率視訊編碼自適應串流傳輸演算法，系統客戶端（client）及伺服器端（server）視訊串流網路互動傳輸流程描述如下：首先客戶端對伺服器端發送影片請求，然後伺服器端在存取視訊串資料的同時會監測及記錄網路的傳輸狀態，再藉由已紀錄的網路狀態去做頻寬預測，透過自適性傳輸演算法，將串流影片的資料量降低至符合頻寬後再傳送給客戶端。

二、傳輸品質調適（Adaptation）

(一) 時間可調性

為了適應網路傳輸狀況，視訊串流傳輸常採用 I、P Frame 進行前向編碼，假設一個 GOP 順序如表 1，解碼的第 1 張圖像為 GOP 裡的第 1 張 Frame，Frame Type 是 I，又因最先解碼的圖像會被後解碼的圖像拿去參考，所以丟棄的 Frame 不可能是 Frame Type 為 I 或是第一張解碼的圖像，而最後解碼的圖像並不會被參考，因此會先從最慢被解碼的圖像依序開始丟棄，第一張被丟棄的就是 GOP 解碼順序 8、Frame Type 為 P 的第 8 張圖像。

表 1 GOP 編碼參數設置

Decoding Order	1	2	3	4	5	6	7	8
Frame Type	I	P	P	P	P	P	P	P
Play Order	1	2	3	4	5	6	7	8

時間可調性是將 FPS 降低，也就是降低每秒播放的 Frame 張數，減少資料量，通常即時視訊串流所傳送的 Frame 基本上為 I、P Frame，因為 P-Frame 只做向前預測，意指解碼只參考前一張圖像，避免串流傳輸發生延遲，而 B-Frame 需要參考前後圖像去做解碼，會延遲串流傳輸的時間，如果 Frame 當中有 B-Frame 也是可以進行時間可調性，只是可能會造成時間上的延遲。本論文中，傳送週期以 Δt 為單位，GOPsize 為 N，其中每個 Frame 的 NALU 長度都不一，當頻寬不足時，依解碼順序刪除 Frame，刪除後的當前 GOP 傳輸量須小於可用頻寬，若不小於可用頻寬則繼續做刪除 Frame 的動作。

(二) 空間可調性

空間可調性是指將不同解析度的影像做編碼，將高解析度的原始影片下取樣 (down sampling)，會產生出二種以上的解析度，SHVC 再將多種解析度的影像去做編碼。假設此影像有三種解析度分別為 640x480、960x540、1280x720，那麼 640x480 的解析度會為基本層，960x540、1280x720 是為第 1 加強層與第 2 加強層，加強層是高解析度與低解析度之間的差值影像，當頻寬足夠時就可以獲得基本層加上加強層的影像，此影像品質較為清晰，若頻寬不足時僅獲得基本層之影像。

假設編碼過後的串流檔最大層數為 N，其中 N 至少為 2 層最多為 7 層，在串流檔中的 SPS 有紀錄最大編碼層數，其欄位名稱為 sps_max_sub_layers_minus1，將其讀取出來便可得知串流檔編碼最大層數，當頻寬不足時，可依序從最高層慢慢刪減，將影片的解析度降低，來減少資料量進而使影片播放過程順暢。

(三) 時間加空間可調性

而時間加空間可調性，是使用降低解析度以及調整播放頻率方法，以配合頻寬不足的狀況，假設影片為動作變化較大的影片，那麼一開始會先選擇刪除 SpatialLayer，如圖 3，當 SpatialLayer 只剩 BaseLayer 且頻寬仍然不足時，反之，影片為動作變化較少的影片，那麼一開始會先選擇刪除 TemporalLayer，如圖 4，當 TemporalLayer 或是 SpatialLayer 只剩第 0 層且頻寬仍然不足時，下一步 SpatialLayer 將會變換成刪除 TemporalLayer，而 TemporalLayer 會變換成刪除 SpatialLayer，如圖 5。

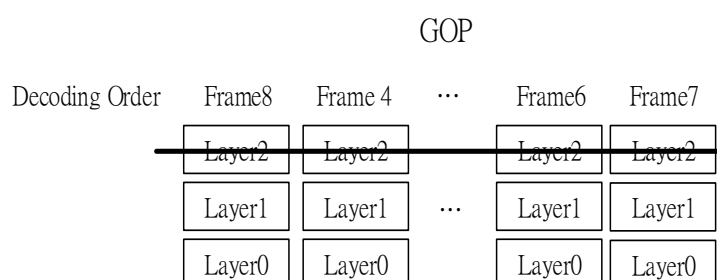


圖 3 GOP Drop Layer 示意圖

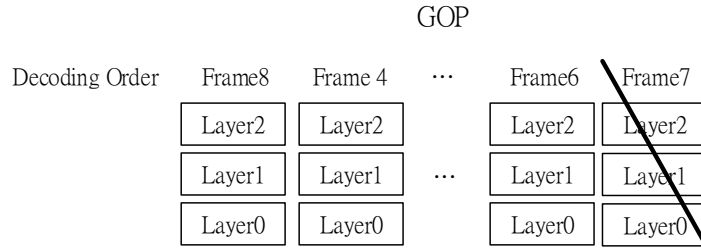


圖 4 GOP Drop Frame 示意圖

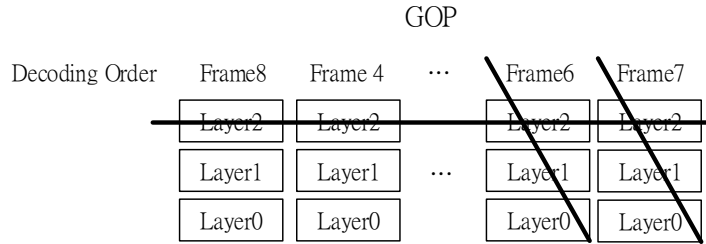


圖 5 GOP Drop Frame 示意圖

三、時間空間可調性自適性傳輸演算法

本論文設計之傳輸適應性演算法考慮影片的動、靜態程度之不同而採取不同的時間空間可調性視訊資訊丟棄策略。論文中將播放速率高且畫面幀間差異量大的影片視為動態影片，將播放速率低且畫面幀間差異量低的影片視為靜態影片。論文所提之視訊串流內容動、靜態程度的判斷方法說明如下：當視訊中有大量物體進行移動時，會有比較多的畫面幀之間差異量。當視訊中只有少量物體進行小範圍移動時，會有較少的畫面幀之間差異量。因此，本論文利用畫面幀間差異量（以 PSNR 為單位），計算第 i 秒畫面幀之間的累加像素差異量 $SecTotalDiff_i$ ，詳如公式（1）， $FrameRate$ 代表影片播放速率， $FrameDiff_{j,j-1}$ 代表二張畫面幀之間的差異量；接著，進行平均每秒累加畫面幀差異量 $AvgSecFrameDiff$ 計算，詳如公式（2）， $VLength$ 代表視訊長度（以秒為單位）。當影片平均每秒累加畫面幀差異量 $AvgSecFrameDiff$ 大於等於特定的差異量閥值 $AvgSecDiffThreshold$ ，就把該影片定義成動態影片，反之把該影片定義成靜態影片，詳如公式（3）（4）。考慮到不同動態影片的差異量很大，需要大量的影片進行結構化的統計分析，才能進行閥值 $AvgSecDiffThreshold$ 最佳化定量計算。視訊串流供應商如 YouTube、Netflix 等，擁有足夠資源可進行量化分析，得出最佳 $AvgSecDiffThreshold$ 定量數值。

$$SecTotalDiff_i = \sum_{j=i*FrameRate}^{(i+1)*FrameRate-1} FrameDiff_{j,j-1} \quad (1)$$

$$AvgSecFrameDiff = \frac{\sum_{k=0}^{VLength} SecTotalDiff_k}{VLength} \quad (2)$$

$$Dynamic\ video\ if\ AvgSecFrameDiff \geq AvgSecDiffThreshold \quad (3)$$

$$Static\ video\ if\ AvgSecFrameDiff < AvgSecDiffThreshold \quad (4)$$

本論文設計之時間空間可調性演算法如圖 6 至圖 8，假設目前偵測頻寬 BW_{now} 大於最低需求頻寬 $BW_{minimum_require}$ ，就會傳送資料，如果 $BW_{now} < BW_{minimum_require}$ ，先判斷影片為動態影片還是較為靜態的影片，如果是動態影片，例如：運動賽事、動作片…等，會先選擇空間可調，因為如果選擇了時間可調而刪除某些 Frame 怕影片會有動作不連貫的情形，反之如果為較靜態影片，例如：新聞影片，因主播移動的動作較少，刪除 Frame 動作而發生動作不連貫的問題機率比較低。

確定影片型態後， $SpatialLayerLimit$ 、 $TemporalLayerLimit$ 為限制的層級，也就是層級刪除至的層級後，轉變成另一種可調方式，如果是動態影片就先執行空間可調 $SpatialLayerScalable(SpatialLayerLimit)$ ，當 Layer 刪除至所設定的最低層 $SpatialLayerLimit$ ，便轉換成刪除時間層 $TemporalLayer$ ；靜態影片則先

執行 TemporalLayerScalable (TemporalLayerLimit)；，當刪除至所設定的最低層 TemporalLayer 時，便轉換成刪除空間層 SpatialLayer，讀取影片片段當前的 GOP 資料，DecodeList 儲存 GOP 的解碼順序， GOP_{data_length} 為當前 GOP 的資料長度、 $Index_{FrameDecode}$ 為 DecodeList 資料的索引值。

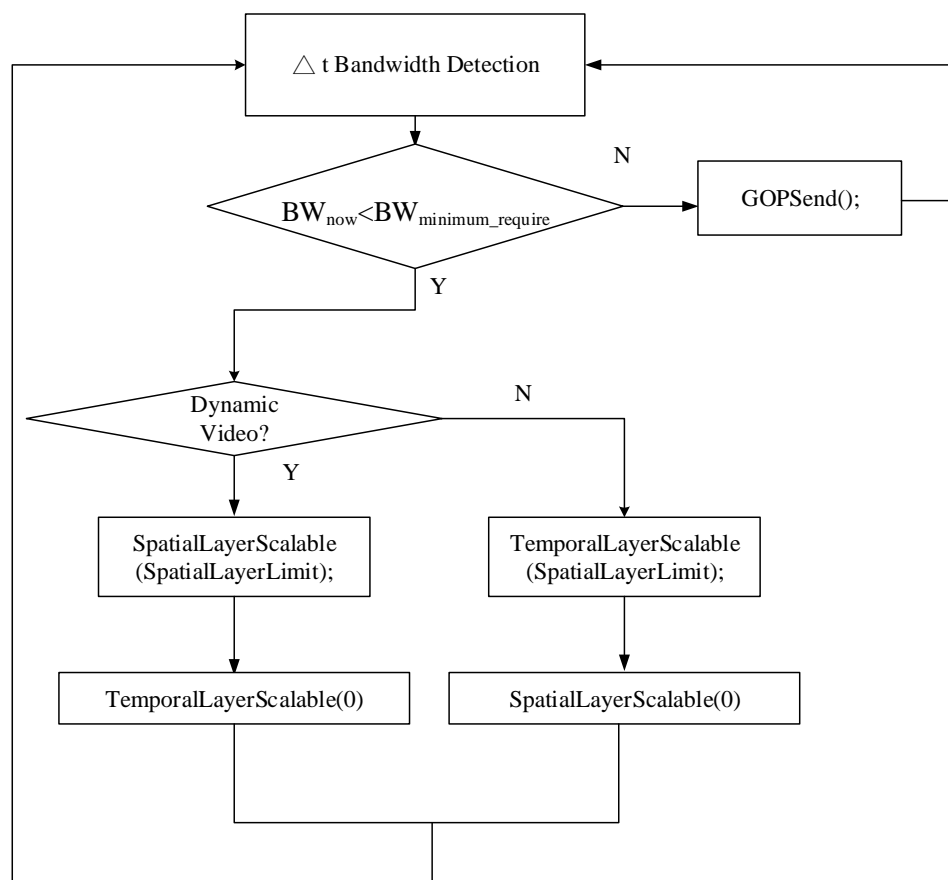


圖 6 時間空間可調性流程圖

空間可調以 $DecodeList[Index_{FrameDecode}].SpatialLayer$ 此變數來記錄當前 $Index_{FrameDecode}$ 為哪一層 SpatialLayer，當資料讀取完成後，進行自適應傳輸空間可調性部分，一開始會先從 DecodeList 裡最後的 $Index_{FrameDecode}$ 進行刪減的動作，以 $DecodeList[Index_{FrameDecode}].SpatialLayer$ 判斷其是否與當前欲指定刪除的 $MaxSpatialLayer_{current}$ 相符，直到刪除至所設定的限制層級 SpatialLayerLimit，假設頻寬依然不足則會繼續執行 TemporalLayerScalable(0)。

時間可調以 $DecodeList[Index_{FrameDecode}].TemporalLayer$ 此變數來記錄當前 $Index_{FrameDecode}$ 為哪一層 TemporalLayer，當資料讀取完成後，進行自適應傳輸時間可調性部分，一開始會先從 DecodeList 裡最後的 $Index_{FrameDecode}$ 進行刪減的動作，以 $DecodeList[Index_{FrameDecode}].TemporalLayer$ 判斷其是否為當前欲指定刪除的 TemporalLayer，直到刪除至所設定的限制層級 TemporalLayerLimit，假設頻寬依然不足則會繼續執行 SpatialLayerScalable(0)。

在 DecodeList 的資料排序就是解碼順序，根據 Frame 的相依性將解碼順序及層數依序儲存於 DecodeList 中，時間或空間可調後判斷當前 GOP 總資料傳輸量 GOP_{data_length} 是否小於 BW_{now} ，如果小於就進行傳輸，否則 $Index_{FrameDecode}$ 將會往前移動，持續進行空間或是時間可調的動作，直至 GOP_{data_length} 小於 BW_{now} ，或是當前 GOP 都完全被刪減掉而無法傳輸。當空間可調降低至最低門檻仍然頻寬不足時，就會開始執行時間可調；反之，時間可調 Frame 刪除至最低數量後，則是執行空間可調。

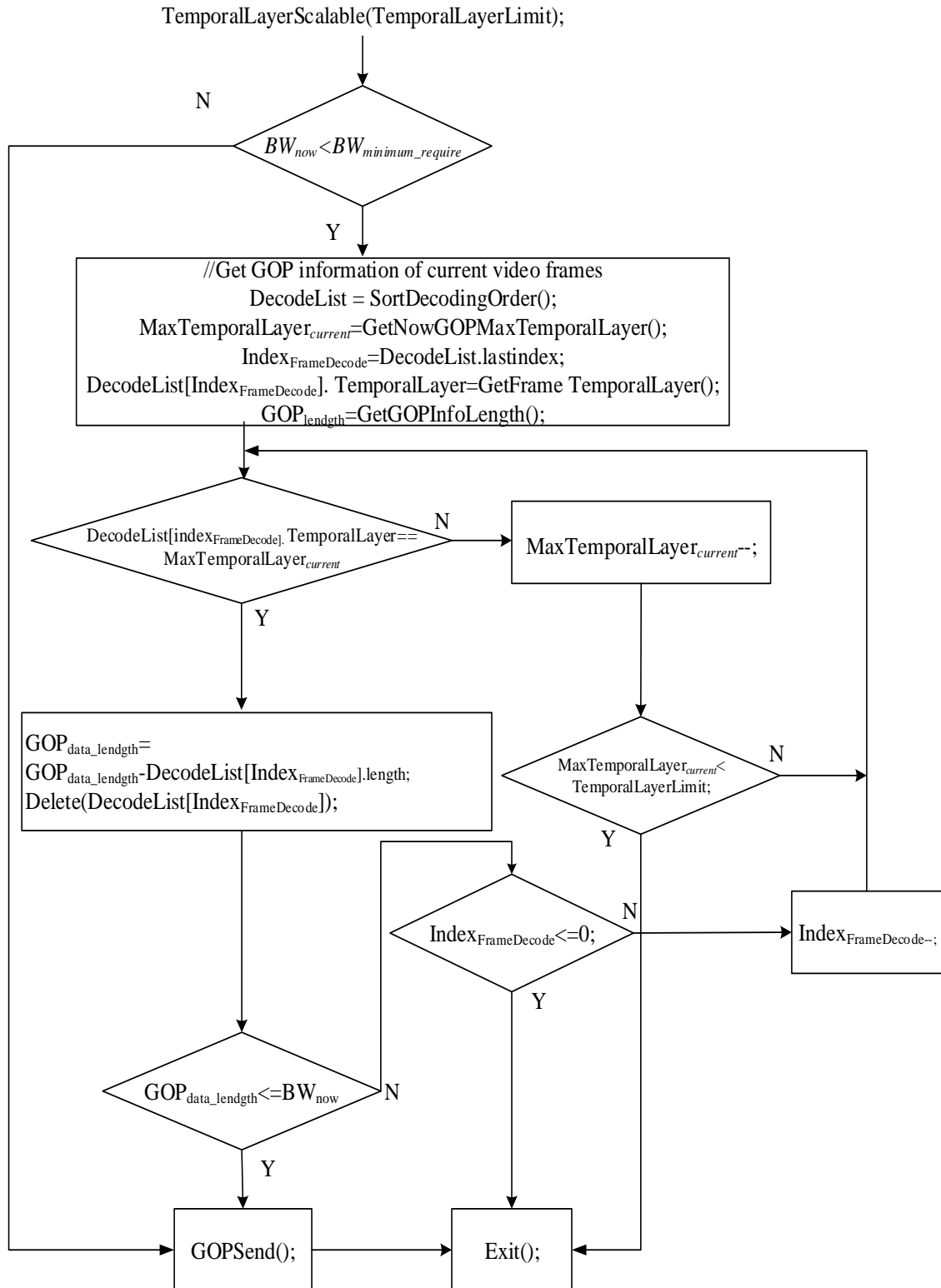


圖 7 時間空間可調性演算法-時間可調部分

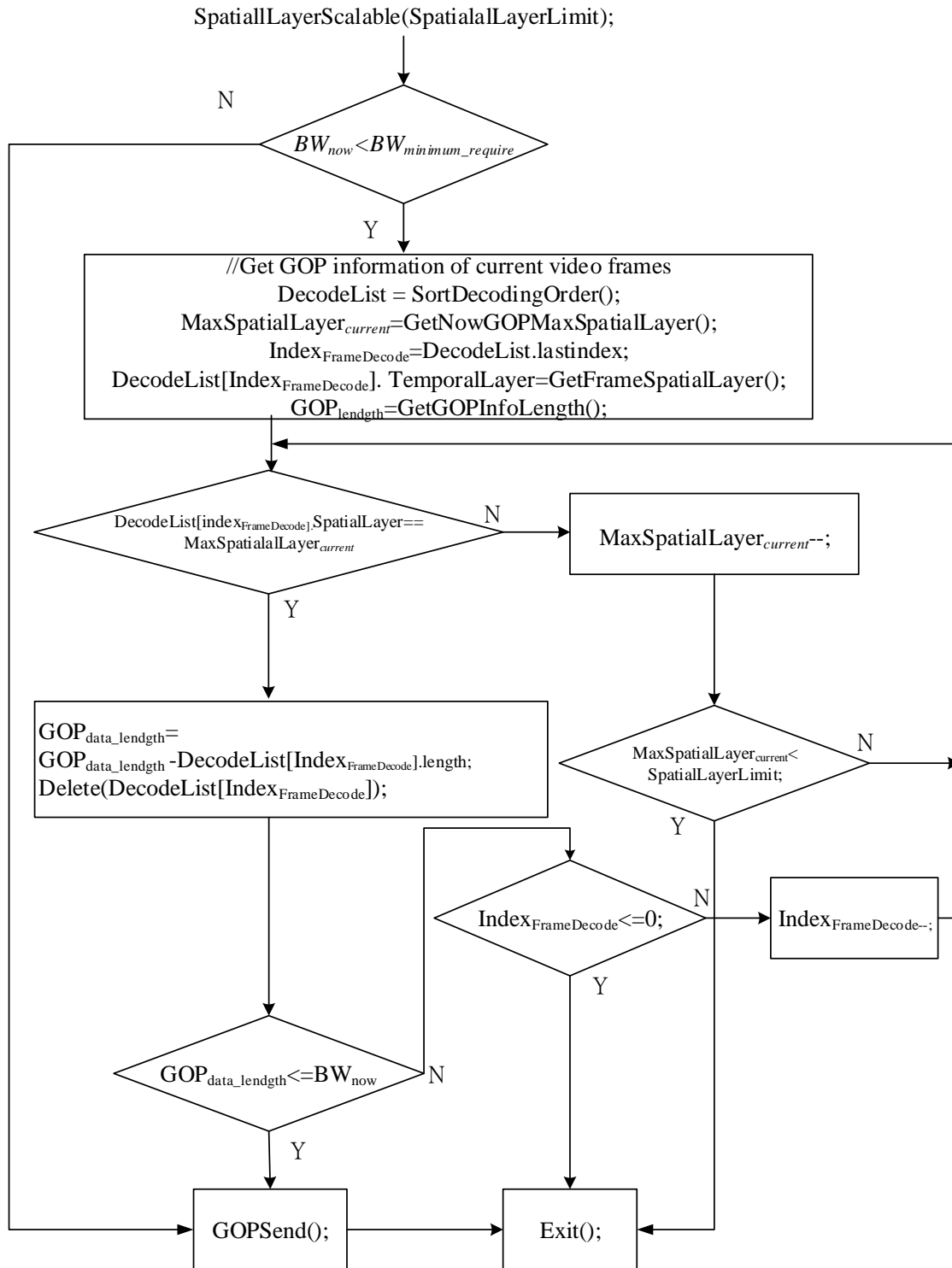


圖 8 時間空間可調性演算法-空間可調部分

肆、實驗步驟、模擬及效能分析

本論文藉由實驗模擬，透過網路頻寬預測方式來決定視訊串流影片的傳輸策略；本實驗模擬會將調整過後的影片與原始影片做比較，比較 HEVC 與 SHVC 的編碼效能及影片品質是否在接受範圍內。高效率視訊編碼使用之編解碼工具為 HM16.0 [21]，可擴展高效率視訊編碼使用之編解碼工具為 SHM12.0 [22]。實驗使用的電腦設備作業系統為 Ubuntu14.04，使用 Linux 建置 SHM 與 HM 模組，iPerf 為用於測量 IP 網路上最大可用頻寬的工具[23]。

一、編碼參數設置

測試影片為 BasketballDrive、PeopleOnStreet、Vidyo；圖 9 為 BasketballDrive 影片，Frame Rate 為每秒 50 張，球員大範圍高速移動；圖 10 為 PeopleOnStreet 影片，Frame Rate 為每秒 15 張，在街景有許多人員慢速移動；圖 11 為 Vidyo 影片，Frame Rate 為每秒 60 張，三位人員在討論時有許多肢體上的動作。三種影片的主要參數設置都一樣，如表 2 所示，QP 為量化參數，GOP Size 為 GOP 內所含的圖片數量，Decoding Refresh Type 為隨機訪問 GOP 的第 1 張圖片類型，Decoding Refresh Type 的值 1 為 CRA，2 為 IDR，Search Range 為 Frame 搜尋範圍，Intra Period 為每幾張就設一個 I-Frame，Input Bit Depth 為圖片位元深度。論文利用二張畫面幀之間的差異量 (PSNR)，把影片內容分類成動態及靜態影片。三種影片先調整成相同解析度 1280 x 720，依不同播放速率 (Frame Rate) 計算累加畫面幀之間的差異量 *SecTotalDiff*，再計算每秒平均累加畫面幀差異量 *AvgSecFrameDiff*；最後，把 *AvgSecDiffThreshold* 設定為 1,000 進行分類，Vidyo、BasketballDrive 分類為動態影片，PeopleOnStreet 分類為靜態影片。BasketballDrive、PeopleOnStreet、Vidyo 影片的參考畫面，如圖 9-11。

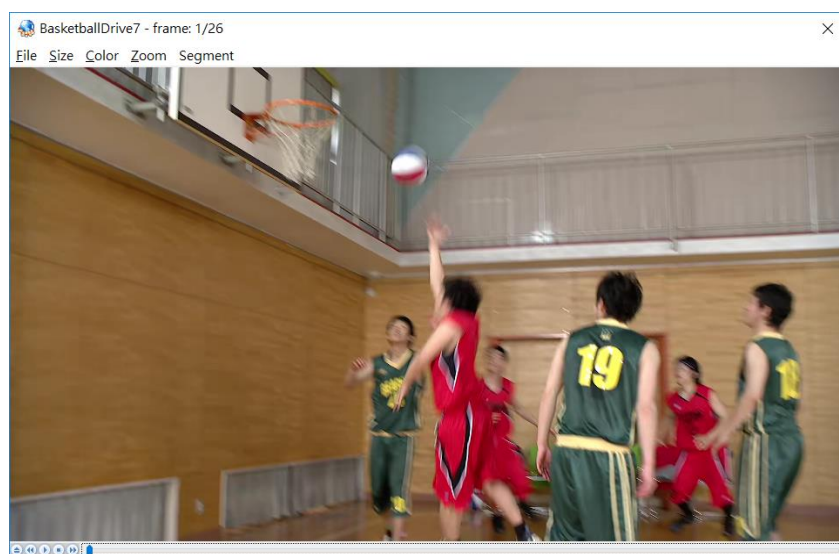


圖 9 BasketballDrive 影片播放畫面

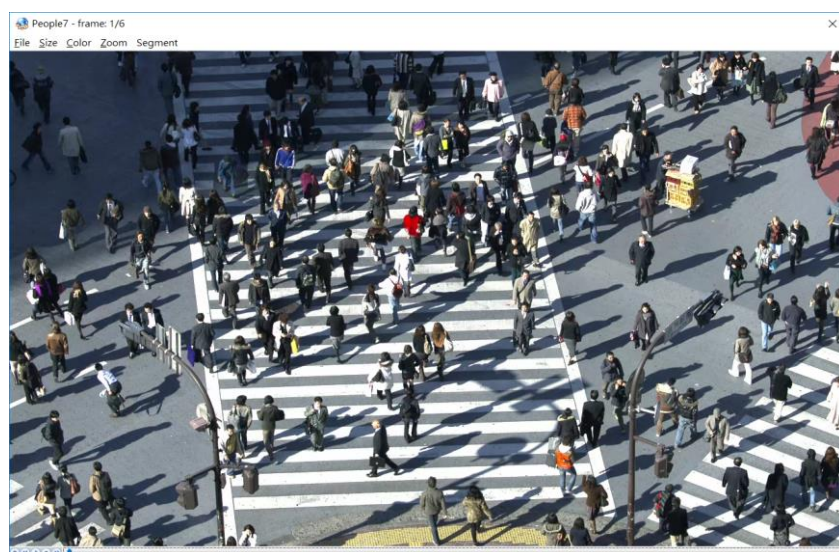


圖 10 PeopleOnStreet 影片播放畫面

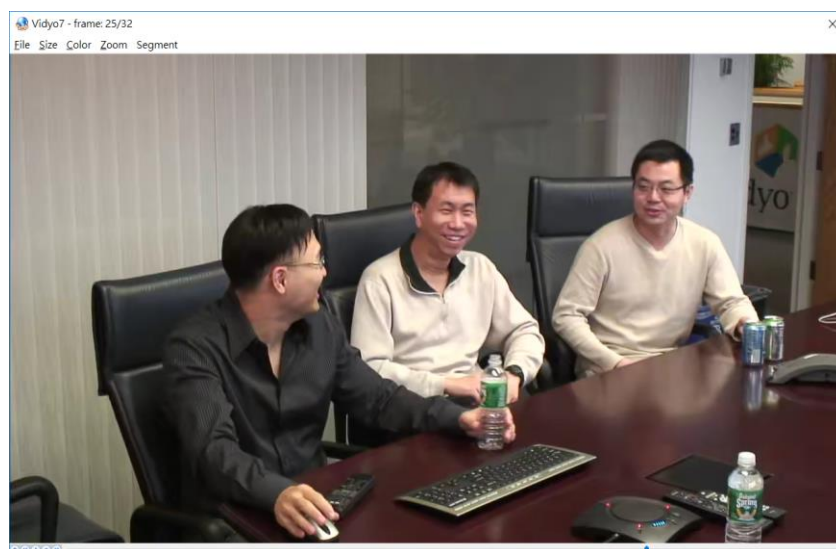


圖 11 Vidyo 影片播放畫面

表 2 主要參數設置

QP	32	Search Range	64
GOP Size	8	Intra Period	8
Decoding Refresh Type	1	Input Bit Depth	8

各影片解析度及其他參數設置如表 3 至表 5 所示，Frame Rate 為每秒播放的張數，Intra Period 為每幾張就設一個 I-Frame，將原始影片分割成數片段以模擬影片變換解析度及刪除 Frame。

表 3 BasketballDrive 影片參數設置

Resolution	1920 x1080	1280 x720	960 x540
Total number of frames		500	
Number of segments		10	
Number of frames in each segment		50	
Frame Rate		50	
Intra Period		8	

表 4 PeopleOnStreet 影片參數設置

Resolution	2560x1600	1920x1080	1280x720
Total number of frames		150	
Number of segments		10	
Number of frames in each segment		15	
Frame Rate		15	
Intra Period		8	

表 5 Vidyo 影片參數設置

Resolution	1280x720	960x540	640x480
Total number of frames		600	
Number of segments		10	
Number of frames in each segment		60	
Frame Rate		60	
Intra Period		8	

視訊網路串流使用 I、P Frame 的編碼可以減少解碼延遲，因為 B-Frame 需要前後參考，而 P-Frame 只須向前參考，在解碼上 B-Frame 花費的時間會多一點，進而影響傳輸時間，所以模擬採用 I、P Frame 的編碼，而 P-Frame 都需要參考前一張圖片，假設要進行時間可調的動作，拿掉某個 Frame，那其 Frame 後面的圖片就都無法解碼，因為沒有前一張圖片可以參考，所以我們在 GOP 的 8 張的圖片將其設為 IRAP 中的 IDR 或是 CRA 圖片，如圖 12 所示，當 GOP8 設為 IRAP，那麼 GOP1 至 GOP7 都是可以由後之前慢慢丟棄，不會因為丟棄 GOP7 而導致 GOP8 後面的圖片都無法解碼。

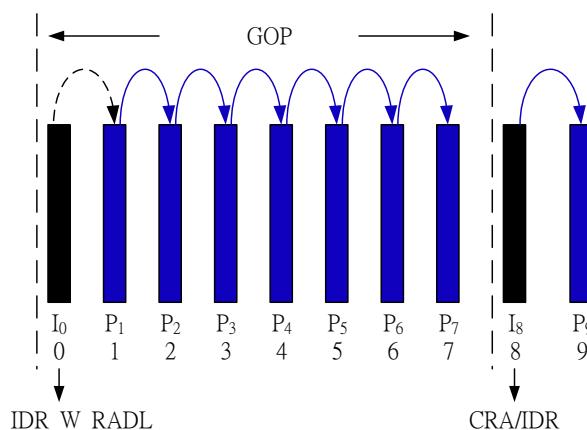


圖 12 模擬實驗 GOP 編碼設置

本實驗使用 evalvid-2.7 [24]來比較 SSIM 與 PSNR，SSIM 為結構相似性 (structural similarity)，是比較兩張圖片之間的相似程度，用來當成影像失真的品質衡量指標，結構相似性在影像品質的衡量上更符合人眼對影像品質的判斷，數值最大為 1，當 SSIM 等於 1 代表兩張圖片相同，數值越低相似度就越低，但對於影像出現位移、縮放、旋轉 (皆屬於非結構性的失真) 的情況無法有效的運作。PSNR 為峰值信噪比 (peak signal-to-noise ratio)，是在影像壓縮中測量重建影像質量的方法，使用均方誤差 MSE 計算，PSNR 值越高越好，通常介於 30 至 40dB 之間。

二、SHVC 與 HEVC 編碼效能評估比較

表 6 為 QP20 (quantization parameter) 與 QP32 的比較，壓縮影片為 BasketballDrive，原始解析度為 1920*1080，Frame 數量為 500 張，QP 量化參數較大檔案容量就會越小，search range 與 GOP Size 越大則會增加運算時間，因為搜尋的範圍變得更大更廣；很明顯地，因為 SHVC 運算複雜度比 HEVC 大幅增加，因此編碼時間上多過於 HEVC，QP20 時 SHVC 壓縮時間比 HEVC 多了約 42476 秒，編碼過後的檔案容量小 HEVC 約 2MB；以壓縮比例來進行觀察，SHVC 在二層的影片上進行 QP20 壓縮 (表 6)，比起 HEVC 約節省 $(34.2+7-39.1) / (34.2+7) = 5\%$ 的資料量。QP32 時 SHVC 壓縮時間比 HEVC 多了約 36830 秒，編碼過後的檔案容量小 HEVC 約 0.6MB。以壓縮比例來進行觀察，SHVC 在二層的影片上進行壓縮 QP32 (表 6)，比起 HEVC 約節省 $(4.4+1.3-5.1) / (4.4+1.3) = 10\%$ 的資料量。

三、SHVC 適應性傳輸效能評估

在頻寬變化下，將影片分割成數個 Segment，假設當前最高頻寬可在 1 秒完整傳送 1 個 Segment，如果頻寬下降至限制的頻寬後，就會啟用自適性傳輸。

表 7 為相同 QP，相同 search range，只有解析度的不同，解析度設有 1920*1080、1280*720、960*540 三種，壓縮影片為 BasketballDrive，原始解析度為 1920*1080，Frame 數量為 500 張，可以發現在眾多參數相同情形下，因運算複雜度的關係，SHVC 所花費的時間還是大於 HEVC，約多了 9782 秒，而檔案大小比 HEVC 少了約 4.4MB。在多層的影片上，如 SHVC 在三層的影片上 (表 7) 進行壓縮，比起 HEVC 約節省 $(8.444-5.151) / 8.444 = 38.9\%$ 的資料量，壓縮利益佳，可節省大量資料儲存空間。

表 6 SHVC 與 HEVC 編碼結果比較 (2 層)

Standard	QP20				QP32			
	SHVC		HEVC		SHVC		HEVC	
Layer	L0	L1	1920*	960*	L0	L1	1920*1080	960*540
Resolution	1920*1080	960*540	1080	540	1920*1080	960*540	1920*1080	960*540
Frame	0-499							
Intra period	32							
Search range	256							
GOP size	16							
Y-PSNR	42.85	40.02	39.93	44.01	35.1	36.3	35.7	35.46
U-PSNR	46.06	44.0	44.4	46.44	41.1	41.53	41.5	41.5
V-PSNR	45.885	45.367	45.8	46.18	39.8	41.62	41.56	40.33
YUV-PSNR	44.575	41.0	40.85	44.74	36.2	37.37	36.85	36.5
Bitrate	5796.6	25457.65	26578.6	5584.6	1015.7	3037	2714.77	1077.55
Total time(s)	89222.3		38015.6	8730.8	66252		23397.2	6024.6
Bytes written to file(kbps)	31304.09		26578.6	5584.6	4101.5		2714.77	1077.55
Original YUV size (MB)	1600	392.5	1600	392.5	1600	392.5	1600	392.5
Compressed size (MB)	39.1		34.2	7.0	5.1		4.4	1.3

表 7 SHVC 與 HEVC 編碼結果比較 (3 層)

Standard	SHVC			HEVC		
	L0	L1	L2	960*540	1280*720	1920*1080
Layer	L0	L1	L2	960*540	1280*720	1920*1080
Resolution	960*540	1280*720	1920*1080	960*540	1280*720	1920*1080
Frame	0-499					
QP	32					
Intra period	32					
Search range	64					
GOP size	8					
Y-PSNR	35.3217	35.6699	35.5265	36.102	36.633	36.12
U-PSNR	41.1783	41.2688	40.9531	41.495	41.92	41.452
V-PSNR	39.9059	40.2354	40.8032	40.47	41.174	41.612
YUV-PSNR	36.3375	36.6888	36.6153	37.1	37.64	37.23
Bitrate	1191.51	675.8352	2278.584	1392.544	2001.345	3527.414
Total time(s)		46241.677		5604.225	9724.235	21134.24
				Total: 36459.6		
Bytes written to file		5273937		1740680	2501681	4409267
Original YUV size (MB)	392.5	692.6	1600	392.5	692.6	1600
Compressed size (MB)		5.151		1.7	2.444	4.4
				Total: 8.444		

(一) 實驗模擬 (a)

將影片 Vidyo 共有 600 張 Frame 並分成 10 個 Segment，也就是 1 秒傳輸 60 張 Frame。實驗限制頻寬變化如圖 13，使用自適應傳輸機制-時間可調性、空間可調演算法、時間加空間可調演算法進行效能評估。

時間可調性演算法實驗結果：因為 Vidyo 影片整體的解析度較低，畫面人物動作多且 Frame 張數較多，所以當自適應傳輸機制刪除 Frame 後，在時間可調後整體的影片品質仍然是人眼可以接受的範圍內。

空間可調性演算法實驗結果：因為空間可調是直接將 Layer 刪除，因此 Frame 並不會有不連續的問題，關於解析度的可接受範圍因人而異，空間可調後因失真的關係而看不出襯衫的紋路，但基本輪廓仍然算清晰可見，在空間可調後的影片品質基本上仍然是人眼可以接受的範圍內。

時間加空間可調演算法實驗結果：藉由刪除 Frame 或是 Layer 來減少傳輸量，直到符合當前可用頻寬。在頻寬不足的情況下使用自適應性傳輸，Vidyo 為動態影片，一開始先用空間可調，當達到所設定的層級後，執行時間可調，此影片模擬 SpatialLayer 的 SpatialLayerLimit 值設為 1，TemporalLayer 的 TemporalLayerLimit 值設為 0，其 SSIM 如圖 18，原始與壓縮後每張圖片之間的相似度最低約為 0.86，代表影片的品質在接近人眼可接受範圍的最低極限值；當 PSNR 之間的比較值等於 0 時，代表兩張圖片完全相同，如果 PSNR 越高代表量化品質越好，PSNR 比較如圖 19，在 5-6 秒時頻寬低於 50%時，影片 Layer 降至最低層時頻寬仍然不足，因此下一步驟則是刪除 TemporalLayer 來達到頻寬，此時 PSNR 約 24dB 左右，理論上代表除了 5-6 秒間的畫面可能是無法接受的品質外，其他畫面壓縮後的影像品質在人眼可接受範圍內。

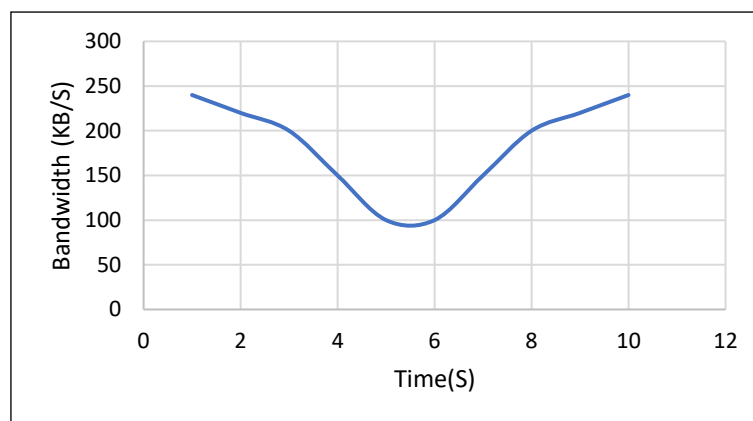


圖 13 Vidyo 實驗限制頻寬變化圖

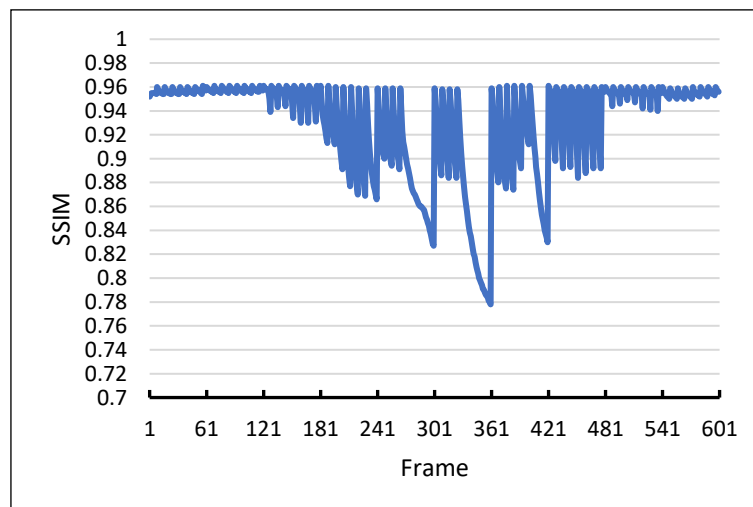


圖 14 Vidyo 時間可調 SSIM 比較

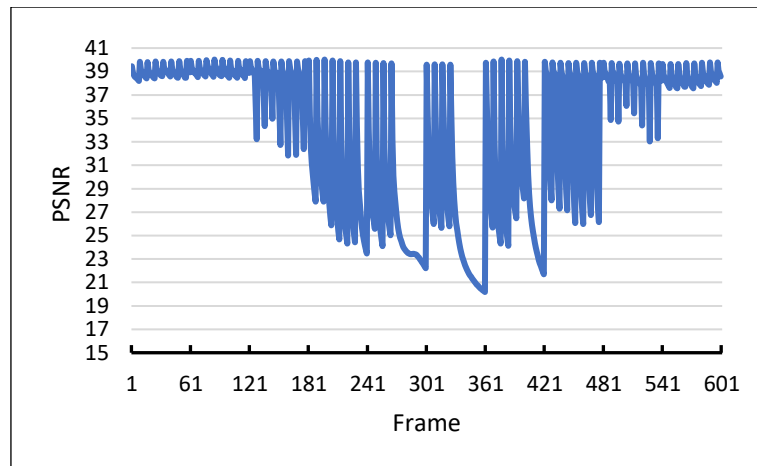


圖 15 Vidyo 時間可調 PSNR 比較

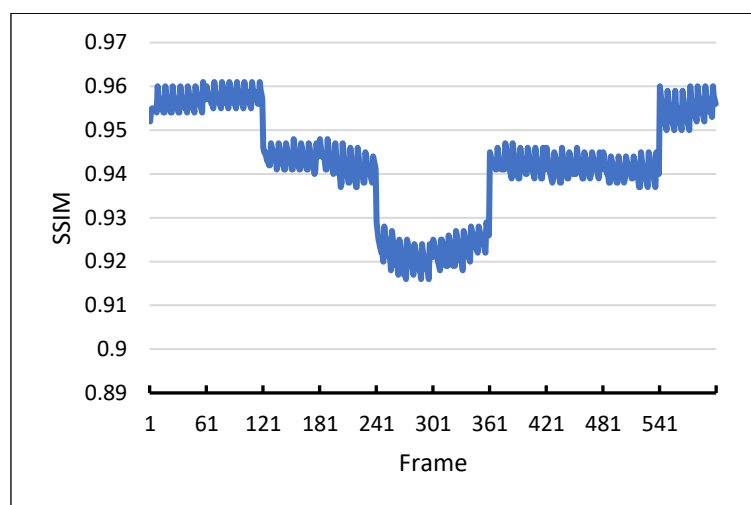


圖 16 Vidyo 空間可調 SSIM 比較圖

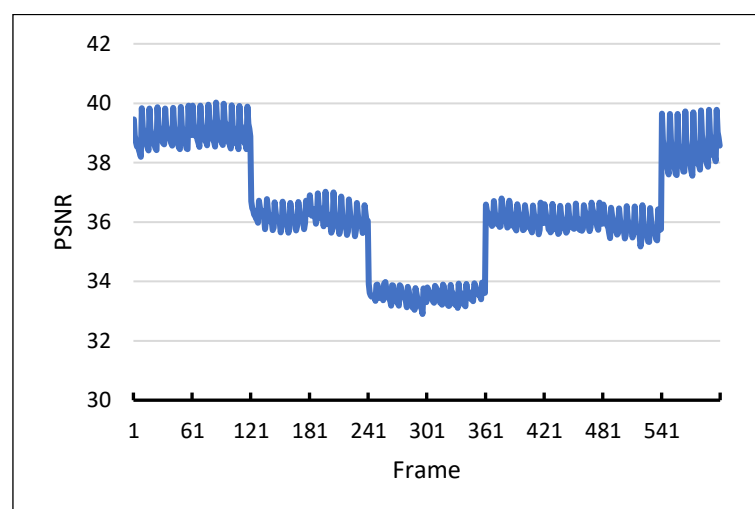


圖 17 Vidyo 空間可調 PSNR 比較圖

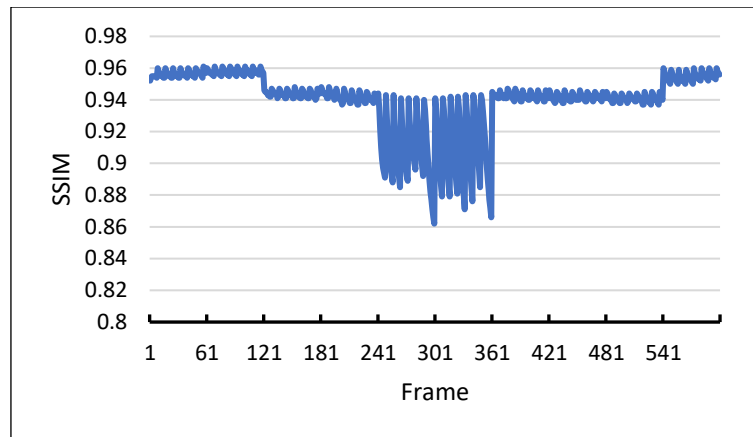


圖 18 Vidyo 時間空間可調 SSIM 比較圖

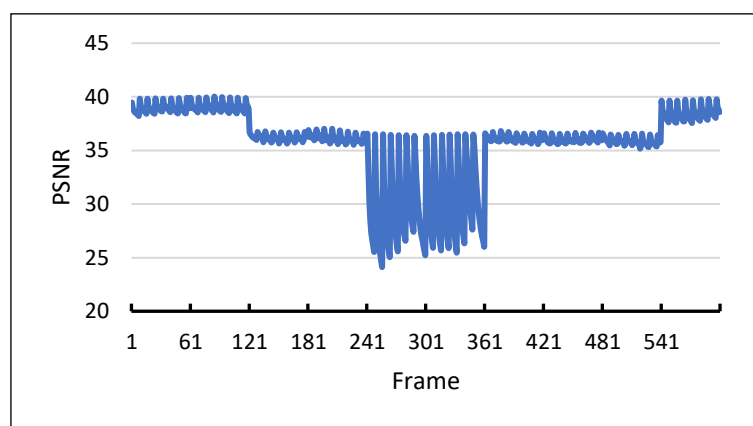


圖 19 Vidyo 時間空間可調 PSNR 比較圖

(二) 實驗模擬 (b)

將影片 PeopleOnStreet 共有 150 張 Frame 並分成 10 個 Segment，也就是 1 秒傳輸 15 張 Frame。實驗限制頻寬變化如圖 20，使用自適應傳輸機制-時間可調性、空間可調演算法、時間加空間可調演算法進行效能評估。

時間可調性演算法實驗結果：因為 PeopleOnStreet 影片整體的解析度較高，且 Frame 張數較少，也就是每張 Frame 之間的變化較大，所以當自適應傳輸機制刪除 Frame 後，在時間可調後整體的影片品質人眼可以觀察的出來影片的不連續性。

空間可調性演算法實驗結果：因為空間可調是直接將 Layer 刪除，因此 Frame 並不會有不連續的問題，而導致畫面中斷，第 3 秒因頻寬太低，該區間品質最低；而關於解析度的可接受範圍因人而異，PeopleOnStreet 的最低解析度為 1280x720，是一般人可接受的品質，但 PSNR 卻不高，空間可調後因失真的關係而看不太清楚指示牌文字，柏油路上的一些紋路也被模糊掉了，空間可調後的影片品質基本上仍然是人眼可以接受的範圍內。

時間加空間可調演算法實驗結果：藉由刪除 Frame 或是 Layer 來減少傳輸量，直到符合當前可用頻寬。在頻寬不足的情況下使用自適性傳輸，其為靜態影片，一開始先用 TemporalScalable (TemporalLayerLimit)；此影片模擬 TemporalLayerLimit 的值設為 1，SpatialLayerLimit 的值設為 0，當 TemporalLayer 刪至第 1 層時，便轉換成 SpatialScalable(0)；其 SSIM 如圖 25，PSNR 如圖 26，SSIM 與 PSNR 波幅大的原因是時間可調後，畫面差異大，無法由前張畫面補償，在第 3 秒頻寬最低時，SSIM 與 PSNR 是最低的，此區間最高只有約 27dB。

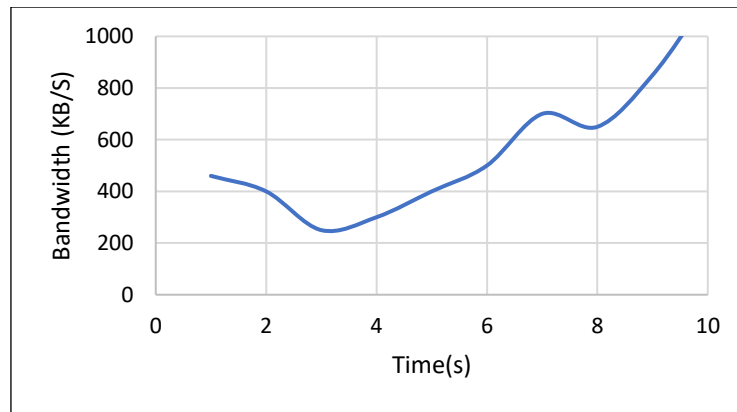


圖 20 PeopleOnStreet 實驗限制頻寬變化圖

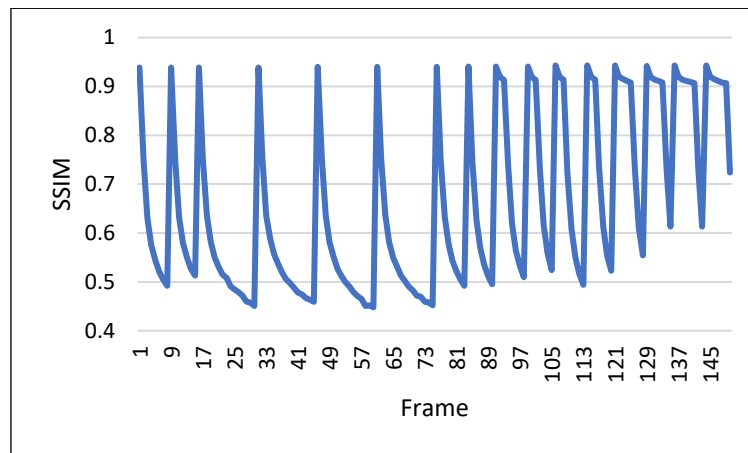


圖 21 PeopleOnStreet 時間可調 SSIM 比較

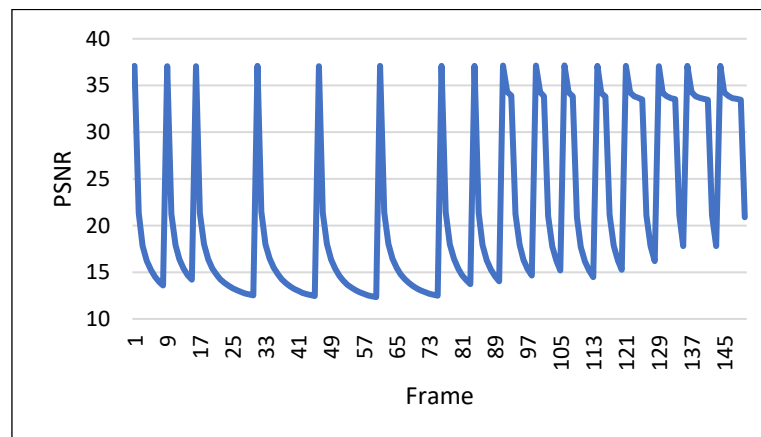


圖 22 PeopleOnStreet 時間可調 PSNR 比較

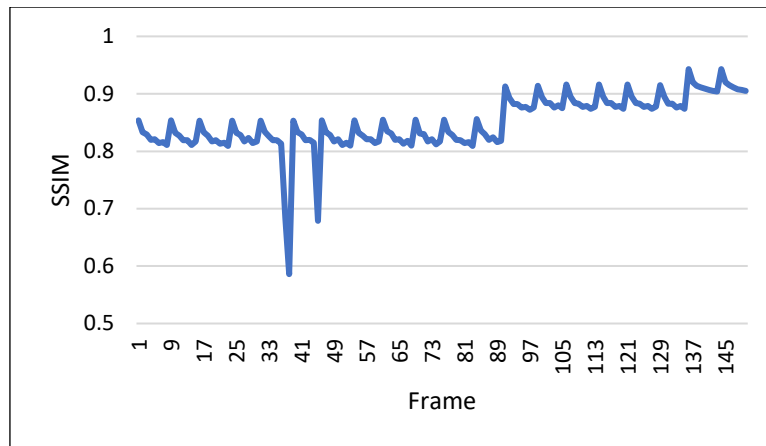


圖 23 PeopleOnStreet 空間可調 SSIM 比較圖

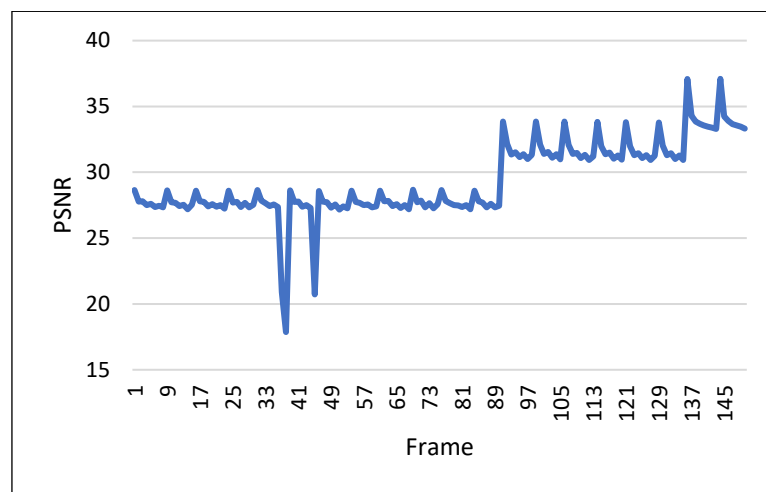


圖 24 PeopleOnStreet 空間可調 PSNR 比較圖

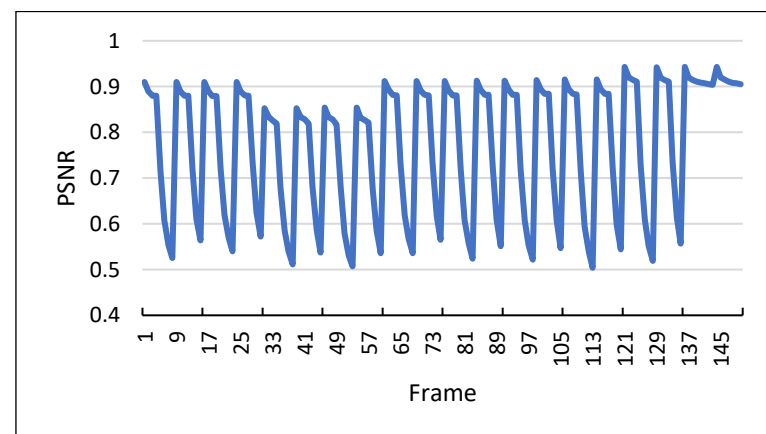


圖 25 PeopleOnStreet 時間空間可調 SSIM 比較圖

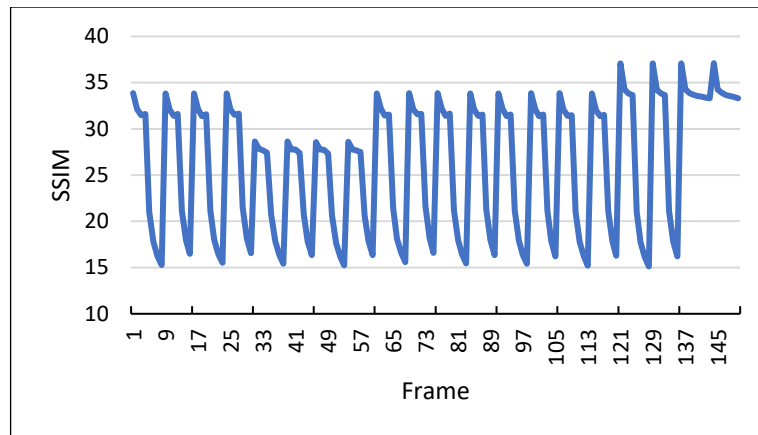


圖 26 PeopleOnStreet 時間空間可調 PSNR 比較圖

(三) 實驗模擬 (c)

將影片 BasketballDrive 共有 500 張 Frame 並分成 10 個 Segment，也就是 1 秒傳 50 張 Frame。實驗限制頻寬變化如圖 27，使用自適應傳輸機制-時間可調性、空間可調演算法、時間加空間可調演算法進行效能評估。

時間可調性演算法實驗結果：雖然 BasketballDrive 影片的 Frame 張數較多，由於是籃球比賽畫面，也就是每張 Frame 之間的變化較大，所以當自適應傳輸機制刪除 Frame 後，在時間可調後整體的影片品質人眼可以觀察的出來影片的不連續性。

空間可調性演算法實驗結果：因為空間可調是直接將 Layer 刪除，因此 Frame 並不會有不連續的問題，在最前 2 秒與最後 2 秒，該區間品質最低；而關於解析度的可接受範圍因人而異，BasketballDrive 的最低解析度為 960x540，是一般人可接受的最低品質，但可調後的 PSNR 卻不高，空間可調後的籃網輪廓不如原始影像那般銳利，空間可調後的影片品質基本上仍然是人眼可以接受的範圍內。

時間加空間可調演算法實驗結果：藉由刪除 Frame 或是 Layer 來減少傳輸量，直到符合當前可用頻寬。在頻寬不足的情況下使用自適性傳輸，其為動態影片，一開始先用 SpatialScalable(SpatialLayerLimit)；此影片模擬 SpatialLayerLimit 的值設為 1，TemporalLayerLimit 的值設為 0，當 SpatialLayer 刪至第 1 層時，便轉換成 TemporalScalable(0)；其 SSIM 如圖 32，PSNR 如圖 33，SSIM 與 PSNR 波幅大的原因是時間可調後，畫面差異大，無法由前張畫面補償，在最前與最後 2 秒頻寬最低時，SSIM 與 PSNR 是最低的，此區間最高只有約 17dB，保持解析度的品質而導致畫面不連續性。

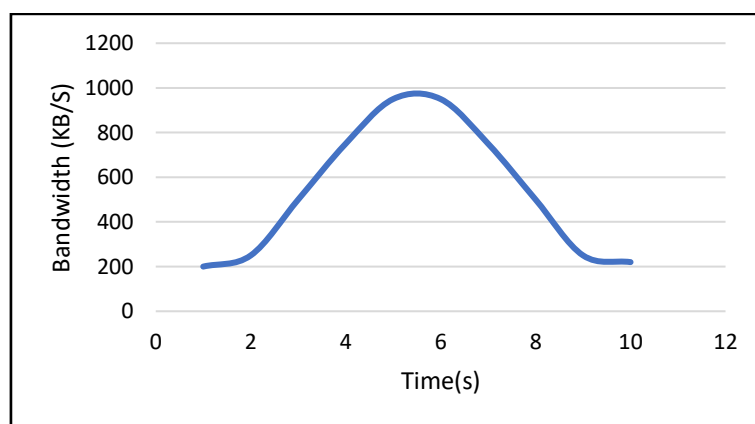


圖 27 BasketballDrive 實驗限制頻寬變化圖

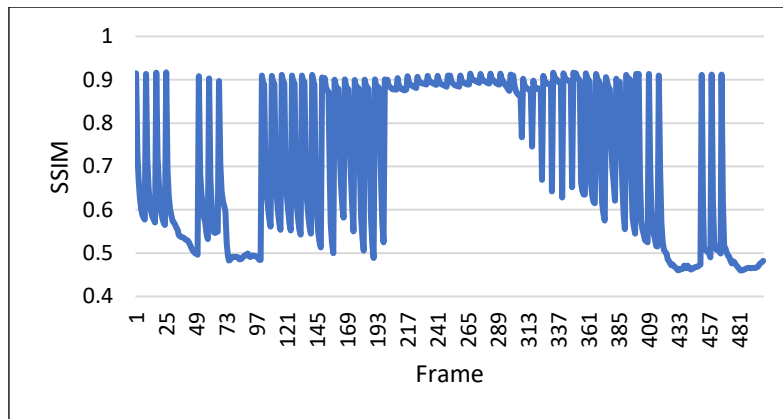


圖 28 BasketballDrive 時間可調 SSIM 比較

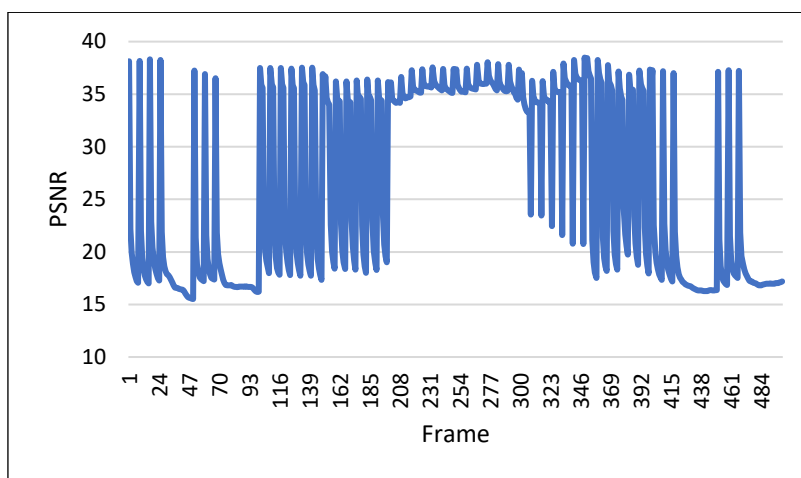


圖 29 BasketballDrive 時間可調 PSNR 比較

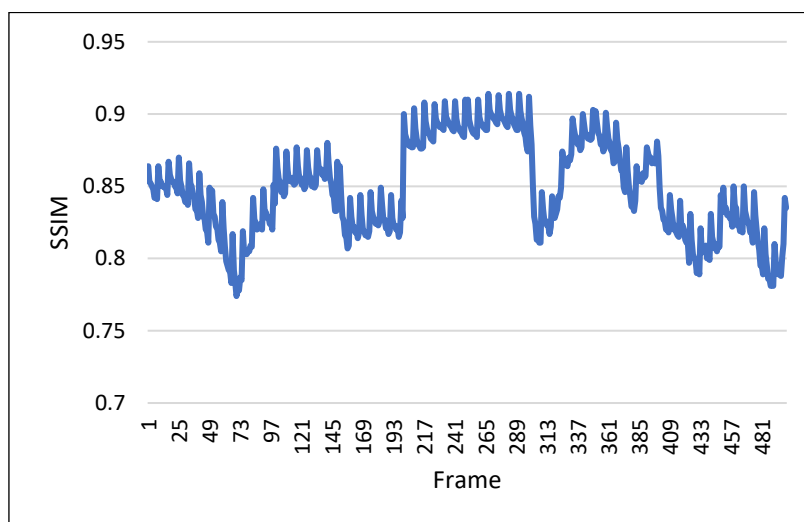


圖 30 BasketballDrive 空間可調 SSIM 比較圖

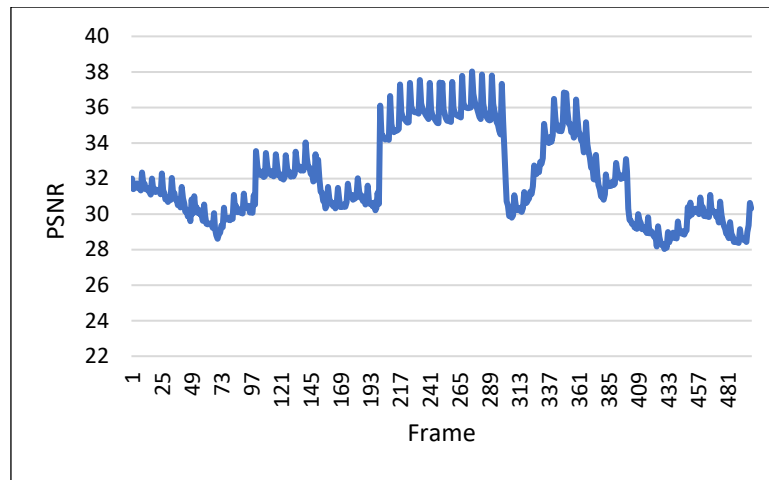


圖 31 BasketballDrive 空間可調 PSNR 比較圖

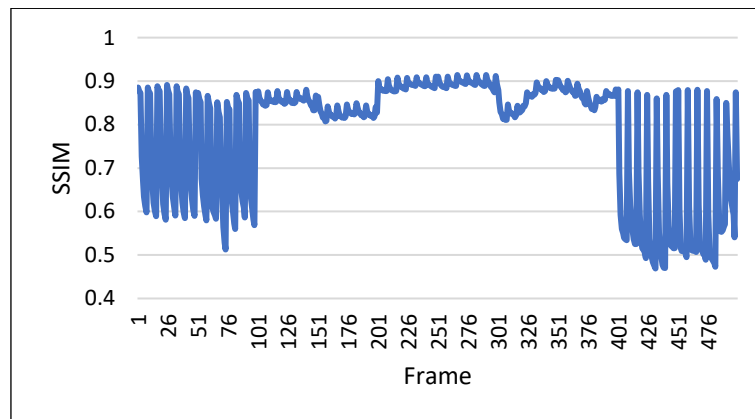


圖 32 BasketballDrive 時間空間可調 SSIM 比較圖

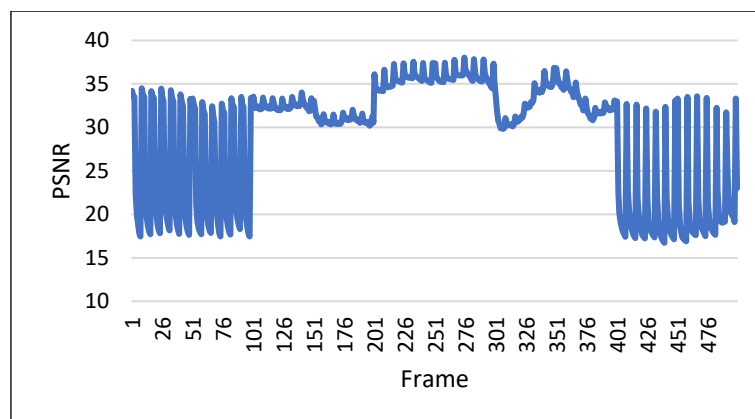


圖 33 BasketballDrive 時間空間可調 PSNR 比較圖

本論文所提出的時間空間可調性自適性傳輸演算法，在頻寬不足的情況下，可以有效改善使用者播放影片的觀看效果。使用者一開始因為頻寬不足而無法觀看影片，透過本論文所提出的時間空間可調性自適性傳輸演算法，經由時間可調、空間可調後，再藉由調整影片的播放速率、影片的解析度、以及影片性質動態調整影片，達到提高使用者的觀看品質的目的。

伍、結論

本論文提出基於可擴展高效率視訊編碼之自適應串流傳輸策略，藉由頻寬預測來決定串流影片的傳輸策略，所提之傳輸策略有時間可調、空間可調、時間空間可調。時間可調利用時間層（TemporalLayer）的層級關係，使影片畫面張數播放頻率下降。空間可調利用空間層（SpatialLayer）的層級關係調整解析度的高低，使影片資料量下降得以符合頻寬。時間空間可調則是配合影片型態，依影片型態，是動態影片還是靜態影片，若是動態影片則是採取先空間後時間可調策略；反之，靜態影片則是採先時間後空間可調策略。使用以上這三種可調方法可降低資料傳輸量以符合頻寬限制，讓影片使用者即使在頻寬不足的情況下仍然可以順暢觀賞影片，滿足觀看需求。本論實驗結果發現：（1）可擴展高效率視訊編碼的主要缺點是運算複雜度高，編碼時間長；（2）可擴展高效率視訊編碼的優點為檔案容量變小，檔案數量不因空間層數量增多而增加，方便檔案管理；（3）本論文所提之時間空間可調演算法可隨網路頻寬變化動態調整，根據不同裝置的網路封包接收狀況，來傳送相對應品質的視訊串流。在網路傳輸最差的情形下，只需要傳送基本視訊串流資料，若在接收環境條件允許下，增加傳送畫面幀的數量，如此重建回來的視訊的品質就會增加，可以提供給使用者更好的影片觀賞品質。

參考文獻

- [1] G.J. Sullivan, J.R. Ohm, W.J. Han and T. Wiegand. (2012). Overview of the high efficiency video coding(HEVC) standard. *IEEE Trans. Circuits Syst. Video Technol.*, 22(12), 1649–1668.
- [2] T. Wiegand, G.J. Sullivan, G. Bjontegaard and A. Luthra. (2003). Overview of the H.264/AVC video coding standard. *IEEE Trans. Circuits Syst. Video Technol.*, 13(7), 560–576.
- [3] J.R. Ohm, G.J. Sullivan, H. Schwarz, T.K. Tan, and T. Wiegand. (2012). Comparison of the coding efficiency of video coding standards including high efficiency video coding (HEVC). *IEEE Trans. Circuits Syst. Video Technol.*, 22, 1669–1684.
- [4] G.J. Sullivan, J.M. Boyce, Y. Chen, J.R. Ohm, C. Andrew Segall, and A. Vetro. (2013). Standardized extensions of high efficiency video coding (HEVC). *IEEE J Sel Top Signal Process*, 7(6), 1001–1016.
- [5] C. Ram and S. Panwar. (2017, Dec. 4–7). Performance comparison of high efficiency video coding (HEVC) with H.264 AVC. *2017 13th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*(pp.4–7). Jaipur, India. <https://doi.org/10.1109/SITIS.2017.58>
- [6] H. Schwarz, D. Marpe and T. Wiegand. (2007). Overview of the scalable video coding extension of the H.264/AVC Standard. *IEEE Trans. Circuits Syst. Video Technol.*, 17(9), 1103–1120.
- [7] P. Helle, H. Lakshman, M. Siekmann, J. Stegemann, T. Hinz, H. Schwarz, D. Marpe, and T. Wiegand. (2013). A scalable video coding extension of HEVC. *IEEE Conference on Data Compression* (pp. 201–210). <https://doi.org/10.1109/DCC.2013.28>
- [8] V. Sze, M. Budagavi and G.J. Sullivan. (2014). *High efficiency video coding (HEVC): Algorithms and architectures*. Springer.
- [9] Z. Shi, X. Sun, and F. Wu. (2012). Spatially scalable video coding For HEVC. *IEEE Trans. Circuits Syst. Video Technol.*, 22(12), 1813–1826.
- [10] I. Wali, A. Kessentini, M.A. Bohamed Ayed, and N. Masmoudi. (2015, Sep. 19–21). Scalable extension of the high efficiency video coding SHEVC performance study. *2015 World Symposium on Computer Networks and Information Security (WSCNIS)* (pp.19–21). Hammamet, Tunisia. <https://doi.org/10.1109/WSCNIS.2015.7368285>

- [11] C.S. Park, T.J. Kim, J.H. Lee, and B.G. Kim. (2016, Nov. 22–25). Adaptive inter-layer prediction algorithm for scalable extensions of high efficiency video coding. *2016 IEEE Region 10 Conference (TENCON)* (pp.22–25). Singapore. <https://doi.org/10.1109/TENCON.2016.7848610>
- [12] C. Singhal and A.K. Sahu. (2017, Dec. 17–20). UHD video transmission using adaptive SHVC in wireless networks. *2017 20th International Symposium on Wireless Personal Multimedia Communications (WPMC)* (pp.17–20). Bali, Indonesia. <https://doi.org/10.1109/WPMC.2017.8301815>
- [13] J.M. Boyce, Y. Ye, J. Chen, and A.K. Ramasubramanian. (2016). Overview of SHVC: Scalable extensions of the high efficiency video coding standard. *IEEE Trans. Circuits Syst. Video Technol.*, 26(1), 20–34.
- [14] I. Wali, A. Kessentini, M.A. Bohamed Ayed, and N. Masmoudi. (2016, Nov.). Statistical analysis of SHVC encoded video. *2016 International Image Processing, Applications and Systems (IPAS)* (pp.5–7).
- [15] C.C. Wang and J.T. Lin. (2014). An effective coding unit size assignment for scalable high-efficiency video coding (SHVC) encoder (Master thesis). <http://ir.lib.isu.edu.tw/handle/987654321/17363?locale=en-US>
- [16] C.C. Wang and P.H. Su. (2014). Analysis and study of decoded picture buffer (DPB) for scalable high-efficient video coding (SHVC) (Master thesis). <https://ndltd.ncl.edu.tw/cgi-bin/gs32/gsweb.cgi?o=dnclcdr&s=id=%22102ISU05428022%22.&searchmode=basic>
- [17] T.T. Le, D.N. Van, and E.S. Ryu. (2018, Jan. 10–12). Real-time 360-degree video streaming over millimeter wave communication. *2018 International Conference on Information Networking (ICOIN)* (pp. 857–862). Chiang Mai, Thailand. <https://doi.org/10.1109/ICOIN.2018.8343244>
- [18] C. Singhal and A.K. Sahu. (2017, Dec. 17–20). UHD video transmission using adaptive SHVC in wireless networks. *2017 20th International Symposium on Wireless Personal Multimedia Communications (WPMC)* (pp. 234–240). Bali, Indonesia. <https://doi.org/10.1109/WPMC.2017.8301815>
- [19] S. Rakngan, T. Mallikarachchi, and A. Fernando. (2019, Jan. 11–13). An adaptive video streaming framework for scalable HEVC (SHVC) standard. *2019 IEEE International Conference on Consumer Electronics (ICCE)* (pp. 1–2). Las Vegas, NV, United States. <https://doi.org/10.1109/ICCE.2019.8662075>
- [20] S.H. Yang and T.W. Liu. (2020, Sep. 28–30). Quality control for hybrid unicast and multicast video transmission systems. *2020 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-Taiwan)* (pp. 1–2). Taoyuan, Taiwan. <https://doi.org/10.1109/ICCE-Taiwan49838.2020.9258044>
- [21] High efficiency video coding HEVC test model 16.0 (HM.16.0). [online] https://HEVC.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.0/.
- [22] Scalable HEVC extension software. [online] https://HEVC.hhi.fraunhofer.de/svn/svn_SHVCSoftware/tags/SHM-12.0/.
- [23] iPerf - The ultimate speed test tool for TCP, UDP and SCTP. <https://iperf.fr/iperf-download.php>
- [24] EvalVid - A Video Quality Evaluation Tool-set. <http://www2.tkn.tu-berlin.de/research/evalvid/>