



# 自由視角影視服務系統研發與應用

國立陽明交通大學電信所 / 賴宇葳

國立陽明交通大學電信所 / 李睿彬

國立陽明交通大學電信所 / 黃紹華

中華電信研究院數位創新研究所 / 白 莊

中華電信研究院數位創新研究所 / 高曉雯

中華電信研究院數位創新研究所 / 朱國華

關鍵字：5G、串流影音、自由視角

## 摘要

隨著5G的普及，其高頻寬、低延遲及大連結的特性讓全新的應用如雨後春筍般出現，而以往因頻寬限制而發展受限的應用也得以開枝散葉，自由視角影視（Free Viewpoint Video, FVV）服務即是得力於此的串流影音服務。

以韓國為例，自由視角影視服務主要應用於體育賽事轉播，傳統的轉播僅有少數幾台攝影機拍攝不同的區域，由導播決定當下要播出哪一個視角，而自由視角則是架設多台攝影機，由用戶自行決定想觀賞的視角，甚至若圍繞同一中心（如：棒球打擊者）

架設的攝影機足夠密集，利用暫停後快速切換各個視角觀看，就能達到時間凍結（time freezing）的效果。此服務需要充分利用5G高頻寬、低延遲的特性，除此之外也有許多技術困難點需要克服與釐清。

本篇文章乃介紹自由視角影視服務所需要的各項技術，並分析其可行性與困難點，希望未來很快能夠在國內各種體育賽事上真實呈現。

自由視角影視服務所使用或需克服的技術，包含：高速相機/介面傳輸、相機間時間精準同步、相機間色彩校正、相機間角度/位置校正、高速影像壓縮與可任意切換視角之



串流協定等。本篇文章將針對小型的自由視角系統架構作介紹，內容包括：多台相機間PTP (Precision Time Protocol) 同步技術、SIFT演算法調整相機間空間角度技術、影像壓縮/視訊封裝、以及串流/視角切換等技術。

## 一、簡介

### (一) 自由視角影視簡介

自由視角影視 (Free Viewpoint Video, FVV) 是一種能讓用戶以互動的方式控制視角的系統，透過視角控制，用戶能自行選擇從3D空間中的任何角度觀看影像，甚至若圍繞同一中心架設的攝影機足夠密集，利用暫停後快速切換各個視角，就能達到時間凍結 (time freezing) 的效果。若此應用漸趨成熟，未來每個人在觀看影片時，都能成為自己的導播，大大改變了傳統影視的服務型態。

### (二) 自由視角影視技術瓶頸

自由視角影視是一項新穎的應用，技術尚不普及也未有標準可遵循，因此各家廠商與研究團隊分別利用各自擅長的技術，打造出不同的架構，提供不同的效果。例如韓國的4DReplay[1]就是其中的佼佼者，其產品特色為可支持較高品質的自由視角播放效果，因此主要是提供精采片段的自由視角回放而非即時的直播互動；Intel True View[2]也是提供此類服務，此種做法主要是密集架設相

機拍攝場景，再利用拍攝出的影像，合成出滑動畫面時，相機與相機間的畫面。此方式雖然能生成高品質的多視角影片且完美呈現出時間凍結的效果，但犧牲了即時性，使用者不能自行控制自己想觀看的角度，減低了沉浸式體驗的交互性與樂趣。而FVV Live-GTI-UPM [3]則是使用了另一種方式，他們將少量相機所拍攝的影像，先透過影像合成處理技術製造出虛擬視角，並利用RTP傳送串流，因此延遲時間極少，不須預傳其他視角的影像就能達到real-time 的效果，因此能有很高的互動性，但觀看虛擬視角時，可以發現在物體邊緣以及色彩上會有許多瑕疵，在觀賞品質的部分大打折扣。

### (三) 小結











由以上介紹可以大致了解，自由視角影視服務的目標以及目前的技術瓶頸，從這些資訊可以了解到，若我們要發展自己的自由視角影視技術，首先要確定所要應用的場域，以及較重視的是品質或是互動性。除此之外也要考慮到建構系統的成本、難易度以及服務的可擴展性等議題。

## 二、研究方法

### (一) GenICam SDK[4]

由於現今的相機功能越來越複雜，除了擷取影像以外還要即時傳輸、串流控

表 1 常見傳輸介面比較表

Interface	Cable length in m	Bandwith max. in MB/s	Multi-camera	Cable costs	"Real-time"	"Plug & Play"
 USB 2.0	5m	40				
 FireWire	4.5m	64				
 GigE Vision	100m	100				
 USB 3 Vision	8m	350				
 Camera Link	10m	850				

制、高精確同步等功能，因此歐洲機器視覺協會（EMVA, European Machine Vision Association）提出了一套相機通用接口標準（Generic Interface for Camera, GenICam），也就是GenICam SDK，讓所有不同之硬體設備能夠透過GenICam SDK達成隨插即用之目標，使所有程式設計師有個共通標準介面去面對不同廠牌型號的相機設備。

本研究藉由GenICam SDK研發設計相機參數管理、多相機同步攝影、多相機影像擷取等模組。

## （二）GigE Vision 相機

現有高速網路介面中，包括USB 3 Vision、GigE Vision、IEEE 1394b、Camera

Link、CoaXPress 2.0和3G-SDI等，均稱得上高速網路介面，常見傳輸介面比較可見表1，表上的顏色代表支援程度，綠色為支援度最高，紅色則不支援。其中，GigE頻寬高達100 MB/s，cable長度最長可到100公尺，且在多相機功能方面提供最廣泛的技術靈活性，另外，若結合Power over Ethernet（PoE）能使用單一cable進行影像傳輸與供電，使得多攝影機建置變得非常簡單，因此可將多台GigE Vision相機拍攝多視角影像，透過高速Switch（>10 Gbps），同時傳輸至Server進行處理。

## （三）精確時間協定（Precision Time Protocol, PTP）[5][6]

精確時間協定（以下簡稱PTP）用於同



步網路上各個終端設備的內部時鐘，其準確度最高可達sub-microsecond等級。PTP原始版本為2002年發布的IEEE 1588-2002標準中，正式名稱為「Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems」，2008年發布修訂版IEEE 1588-2008，也稱為PTP version 2（簡稱PTPv2），第二版相較於原始版本提高了準確度、精度以及穩健性。

在PTP時間同步的系統中，各設備的時鐘可分為三類：普通時鐘（ordinary clock）、邊界時鐘（boundary clock）以及透明時鐘（transparent clock），普通時鐘是網路中的終端設備，只有一個port參與PTP時間同步，可作主時鐘（master clock）提供時鐘源，或是作從時鐘（slave clock）從其他的時鐘源獲取時鐘資訊，不能作中間節點把時鐘向其他節點傳遞。邊界時鐘可以存在兩個以上的port參與PTP時間同步。其中一個port從上游設備同步時間，其餘的port向下游設備發佈時間。透明時鐘作中間傳輸節點，收到PTP封包之後其內部會計算封包在本節點的駐留時間，以此來修正時間標籤再向下傳遞。不同的設備會使用不同的PTP message types，普通時鐘和邊界時鐘使用Sync、Follow\_Up、Delay\_Req和Delay\_Resp傳遞時間訊息，透明時鐘使用Pdelay\_Req、Pdelay\_Resp和Pdelay\_Resp\_Follow\_Up來測量通訊設備上的延遲，以讓系統對其進行補償。

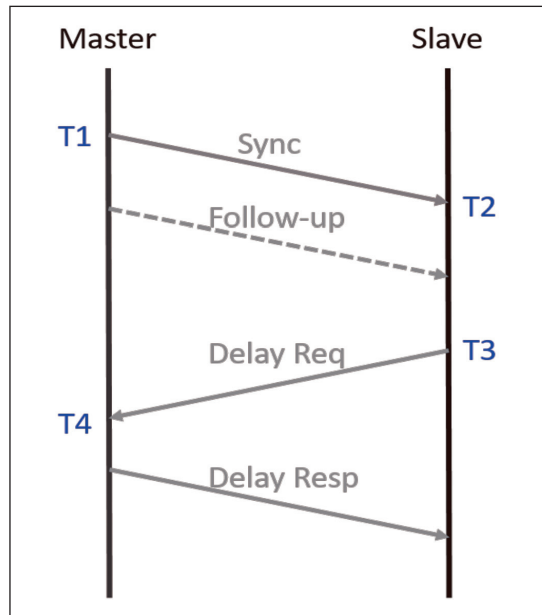


圖 1 PTP 同步作法示意圖

PTP進行同步的方式如圖1，T1為Master發送Sync封包給Slave的時間，T2為Slave收到Sync封包的時間（只由Slave紀錄），T3為Slave發送Delay Req封包的時間（只由Slave紀錄），T4為Master收到Delay Req封包的時間（由Delay\_Resp）。假設傳輸對稱，slave取得這四個時間後，利用式1可算出自己與master的delay，並利用此結果將自己的時鐘向master校正。詳細作法可分為Step-1與Step-2，Step-1做法是將發送Sync封包的時間戳記timestamp T1記錄下來，而Step-2則會利用發送第二個封包（Follow-up）攜帶送出Sync封包的timestamp T1給slave。

本研究利用PTP協定，同步了所有相機

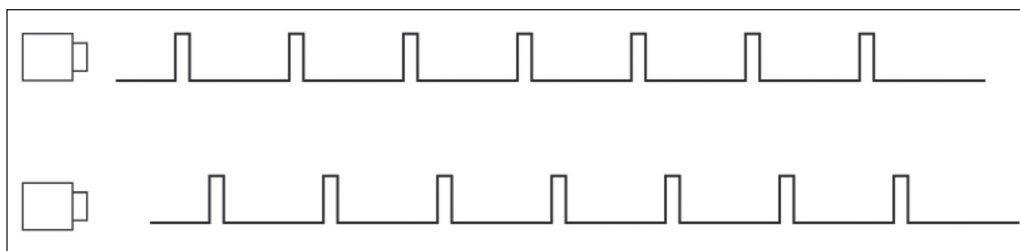


圖 2 未經 PTP 同步之相機 free run 示意圖

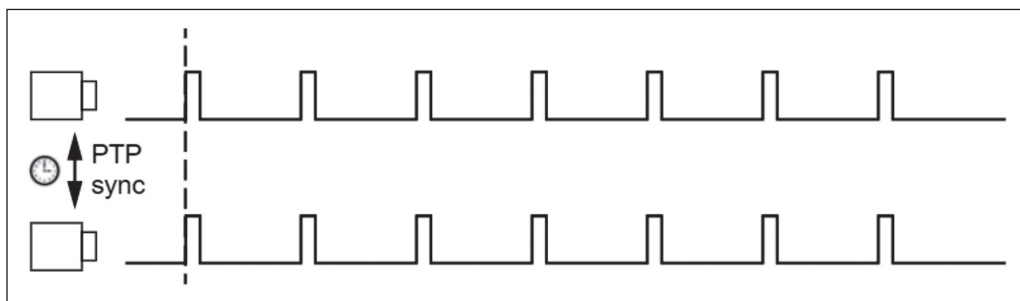


圖 3 經 PTP 同步之相機 free run 示意圖

內部時鐘，讓每台相機能精準同步拍攝，圖2與圖3為PTP同步前與同步後之相機free run狀態示意圖。

$$\text{Delay} = \frac{(T2-T1)+(T3-T4)}{2} \quad (\text{式 1})$$

#### (四) 尺度不變特徵轉換 (Scale-Invariant Feature Transform, SIFT) [7]

尺度不變特徵轉換 (以下簡稱SIFT) 演算法，是一種用來偵測與描述影像中的局部性特徵的電腦視覺演算法，其取出的特徵值對於：位置 (location)、尺度 (scale)、旋轉 (rotation)、…具有不變 (invariant) 的特性，對於：雜訊、光線變化以及視角些微

改變的影像也具有相當穩健 (Robust) 的特性。該技術之應用範圍很廣，特別是針對兩張照片中共同相似點的偵測配對 (如下圖4與5所示)。其他應用領域還包括：物件檢測、影像縫合、建立3D模型、影像追蹤和動作比對等。且由於現今電腦硬體速度較以往大幅提升，SIFT的運算速度經過一定優化可接近 real-time，因此像機器人視覺感知這類對時間要求較嚴格的應用也能使用此演算法。

此演算法的範例結果如圖4，此範例圖為同一拍攝主體不同角度的兩張圖片，正好符合自由視角應用的需求，由此範例可看出此演算法在此情景之下可以很好的找到對應點，故本文章後續關於相機校正的部分，會

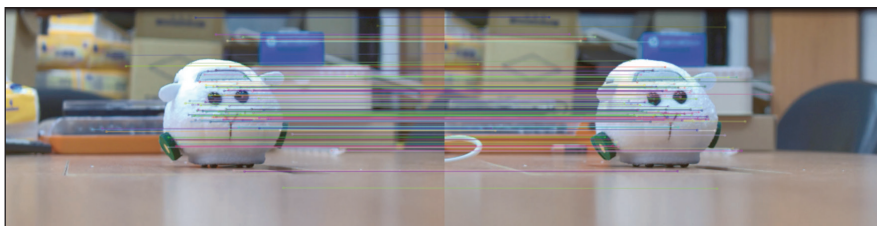


圖 4 SIFT 計算對應點範例

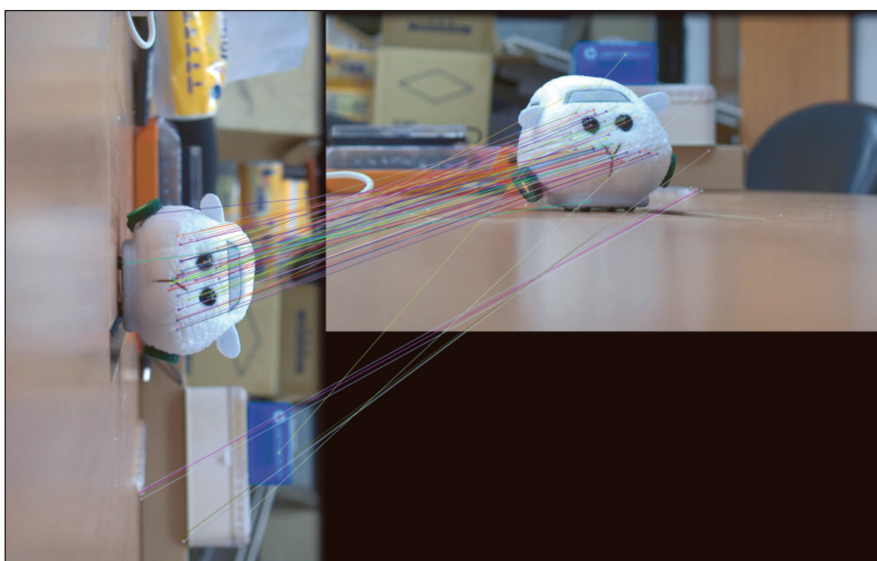


圖 5 SIFT 計算歪斜圖片的對應點範例

利用SIFT計算出對應點，並根據此相對應點之座標，計算空間參數並達到協助校正相機方位的目標。而圖5則能看出即使兩張圖片有一定角度的歪斜或旋轉，SIFT仍能正確地找到對應點。

#### (五) 即時訊息協定 (Real-Time Messaging Protocol, RTMP) [8]

即時訊息協定 (以下簡稱RTMP) 為

Adobe公司所開發，此協定主要用在影音串流傳輸，包括YouTube與FaceBook的直播均採用RTMP做影音傳輸。RTMP傳輸層使用TCP，此協定一開始需要經過Handshake建立連線，並可保證後續的傳輸品質。RTMP使用Port 1935，使用AMF (Action Message Format) 與FLV (Flash Video) 規範，支援的聲音編碼為AAC、MP3，支援的影像編碼為H.264



## （六）H.264 壓縮 [9]

H.264也稱作MPEG-4 AVC (Advanced Video Codec, 高級影像編碼), 是一種視訊壓縮標準, 同時也是一種被廣泛使用的高精度影像的錄製、壓縮和發佈格式。H.264因其是藍光光碟的一種編解碼標準而著名, 所有藍光播放機都必須能解碼H.264, 成為了目前影像編碼領域的絕對霸主, 超過80%商用市場的滲透率。

H.264比其他大部分編碼標準有著更高的影像品質和更低的編碼率, 因此也被廣泛用於網路流媒體資料、各種高清晰度電視廣播, 以及衛星電視廣播等領域。而後續推出的H.265相較於H.264的編碼率更低, 且經過主觀視覺測試得出的資料顯示, 在編碼率較H.264減少51-74%的情況下, H.265編碼影像的品質能與H.264編碼影像近似甚至更好, 但目前RTMP尚未支援H.265編碼, 故實驗部分還是使用H.264編碼。

## （七）HTTP 即時串流 (HTTP Live Streaming, HLS) [10]

HLS為Apple Inc.於2009年時所提出, 為基於HTTP(S)的串流媒體協定, 可支援直播與點播。與另一個常見的串流協定RTP相比, 其優勢是, 可以穿越所有允許HTTP通過的防火牆或代理伺服器, 能手動切換不同畫質的影片, 或是配合用戶頻寬自動切換到對應畫質, 且由於HLS會預載至少三個新的

影片檔案 (ts檔), 雖然播放前須要花一些等待時間, 但可避免觀賞期間卡頓, 能夠提供使用者較高的觀賞品質; 缺點則是即時性較差, 由於需要預載ts檔, 因此播放出的畫面會與實際畫面有較長的時間差, 短則3秒, 長則30秒, 不同的頻寬與ts檔長度會影響時間差的長短。

HLS 的作法如下:

1. 將檔案分割成多個 ts 檔, 一個 ts 檔通常不超過 10 秒、或不短於 1 秒。
2. 用一個 index file 記錄這些 ts 檔的順序與位址, 即 m3u 檔。常見的 m3u 檔的副檔名有 m3u8 (UTF-8) 與 m3u (Windows-1252)。
3. Client 向 Server 請求 index file。
4. Client 取得 index file 後, 會解析並按照順序逐一向 Server 請求指定路徑的 ts 檔。
5. Client 預載完一定數量的 ts 檔後, 影片會開始播放, 並持續請求下一個 ts 檔, 若該 index file 內記錄的 ts 檔下載完後, Client 會再請求下一個 index file, 繼續重複步驟 4。

## 三、實驗結果

### （一）實驗架構

本研究之實驗架構如圖6, 將四台GigE Vision相機連接至Switch的1G port, Switch的10G port連接Server, Server會將相機擷取到



的影像分別儲存，再串流成為多視角影片。

圖7為本研究之軟體架構，分為Camera Server、監控端、Media Server與Web Browser。Camera Server負責所有跟相機有關的工作，包含管理修改相機參數、開啟PTP同步、相機校正以及推流至Media Server。監控端的工作包括：顯示有連線的相機編號、時間同步誤差、以及定時儲存Log檔，以便後續查看相機狀況，雖然監控端並非FVV服務所需的主要功能，但若未來增加相機數量到數十台甚至上百台，則一定會需要監控系統來管理多台相機的狀態。Media Server的工作很單純，就是接收Camera Server的推流與回應Web Browser拉流的請求，雖然僅有這兩

項功能但由於FVV需要切換視角的功能，如何能同時接受多用戶連線，並讓每個用戶能各自切換自己想要的視角，且不能有過長的Delay，這些都是現行標準難以實現的功能，故這部分將會是未來研究的重點。

## (二) 多相機 PTP 同步擷取影像

多台相機利用PTP同步後，將其內部時鐘的時間顯示同步誤差低於1  $\mu\text{s}$ 。為了證明此同步誤差數據，我們使用兩台相機同步拍攝轉速高達95.88 km/h的電風扇，拍攝結果如圖8，可以觀察到在這個運動速率下的確能拍攝到同步影像。

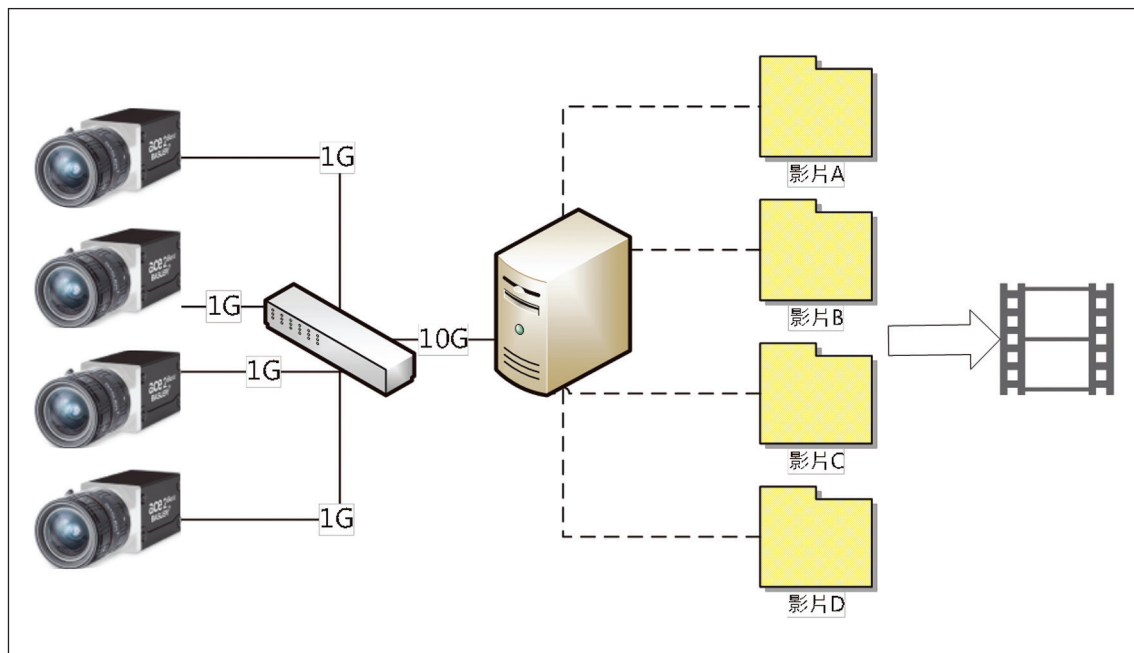


圖 6 實驗架構圖



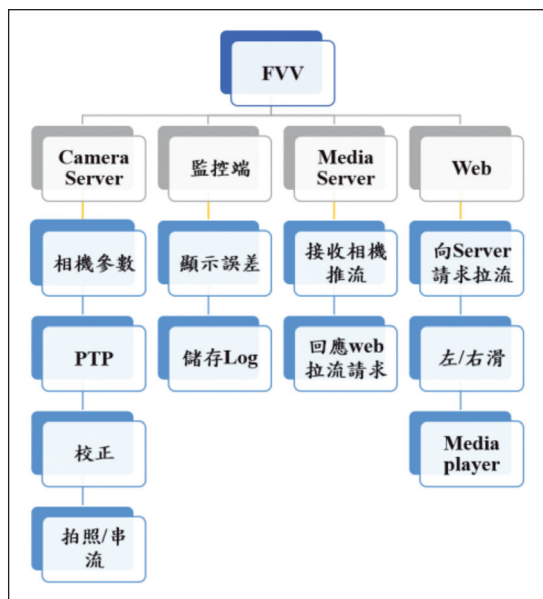


圖 7 軟體架構圖

### (三) SIFT 計算圖片對應點

由於本研究需要調整不同視角畫面的色差，故實驗過程使用兩台參數設定不同的相機拍攝不同角度的照片（約相差15度），圖9為兩張圖利用SIFT找出對應點的圖片，圖上以直線兩兩相連的點即為計算出的對應點，可以觀察到所標示出的點，的確為對應點。

利用這些對應點座標進行幾何運算可以算出兩台相機畫面間的誤差以達到協助校正的目的，以圖10為例，圖片分別為不同台

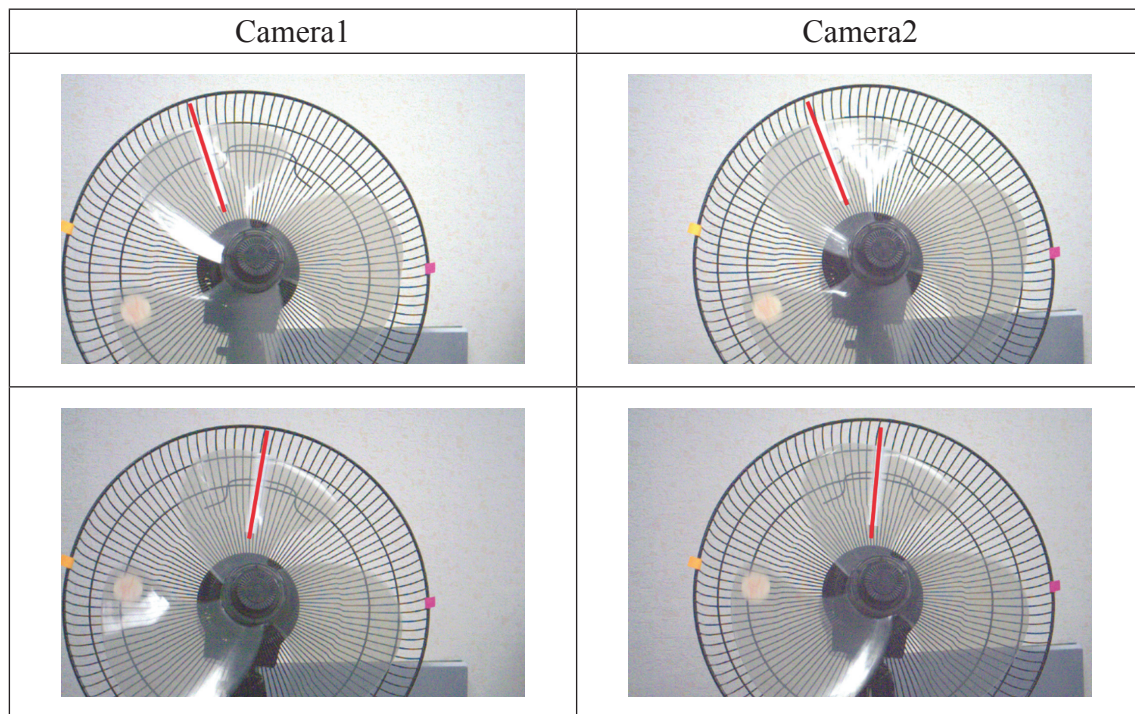


圖 8 同步拍攝結果



圖 9 SIFT 計算對應點結果



圖 10 旋轉圖片範例

相機拍攝，編號依序為40115493（左上）、40115495（右上）、40115498（左下）、40115506（右下），其中右上的圖片與其他圖相比有明顯的旋轉，圖11即為利用對應點計算出的旋轉角度誤差，架設相機陣列時即可利用此數值輔助調整。另外，垂直、水平、放大縮小等也可以利用對應點座標計算誤差值。

#### （四）Linear Regression 計算色彩校正矩陣

色彩校正的方式參考論文《Color calibration for multi-camera imaging systems》[12]的作法，用SIFT找出的對應點，其RGB值經過Linear Regression算出校正矩陣，方程式為式2，使用式2對要校正的圖片做矩陣運

相機編號	PTP status	PTP 誤差	旋轉
40115493	Master	0	
40115495	Slave	-8	32.763317
40115498	Slave	-330	0.000000
40115506	Slave	18	0.508604

圖 11 計算旋轉誤差結果

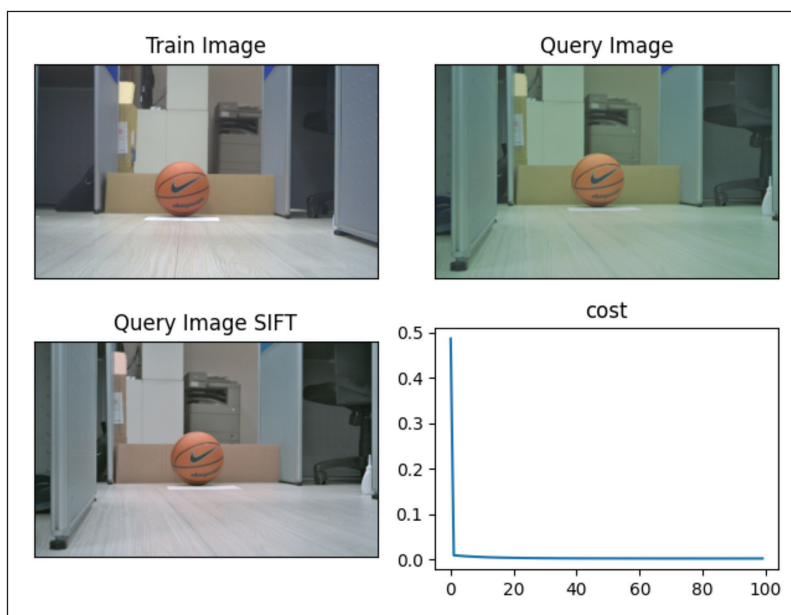


圖 12 色彩校正結果

算，其結果如圖12，Query Image為須校正的圖片，Train Image為目標圖片，這兩張圖是由不同角度拍攝的圖片，Query Image SIFT為校正結果，可以觀察到其色彩的確更接近Train Image。另外，圖中cost的部分顯示出在做Linear Regression的過程中epoch與誤差值的關係，可由此觀察大概需要做多少個epoch達到收斂。

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} Gain00 & Gain01 & Gain02 \\ Gain10 & Gain11 & Gain12 \\ Gain20 & Gain21 & Gain22 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (\text{式 2})$$

### (五) 多視角影像串流呈現

本研究使用Web Browser呈現多視角影像。由相機拍攝的影像會經RTMP推流至Media Server，當用戶需觀看時，再利用Web Browser發出HLS請求，Media Server回應此



請求後即可觀看串流影像，此Web Browser畫面中可看到有左滑/右滑的按鍵，如圖13、圖14，按下該按鍵後會切換至左/右的相機影像，如此即為FVV影像串流的簡易呈現。

#### 四、結論與展望

本研究初步建立了小角度的自由視角系統，能夠在較小的場域看出小角度變化的效果，但此系統要擴展到大場域（如運動賽場）且大角度拍攝，仍有許多障礙需要克服，例如：大量串流影像的頻寬限制、串流方式設計、跨Server的PTP同步等，若想要達

到高品質的視角轉換效果，大約每三度角就要設置一台相機，如何達到頻寬與影片品質的平衡尤其重要，本研究未來將著重於研究適合FVV系統的串流方式，讓FVV系統兼顧頻寬以及切換視角的即時性，達成技術自主及可商用化之目的。

誌謝：本研究於中華電信研究院計畫TL-110A307支助下完成。

#### 參考文獻

1. 4DReplay Corp., <https://4dreplay.com/?ckattempt=1>
2. Intel Corp., <https://www.intel.com.tw/content/www/tw/zh/sports/technology/true-view.html>

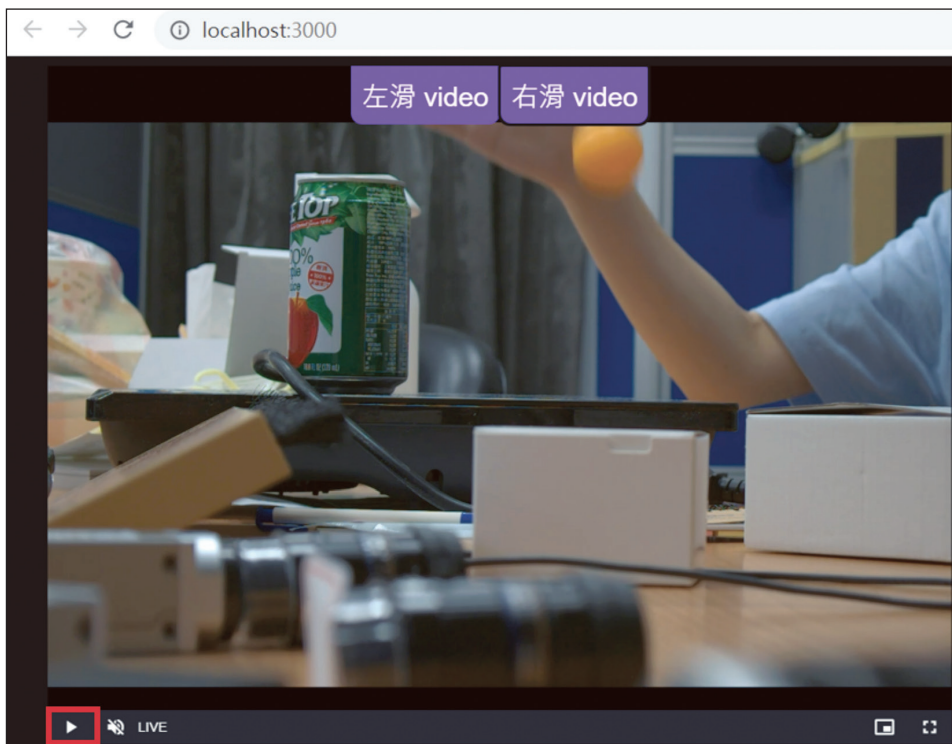


圖 13 Web Browser 呈現 FVV 影像串流 1

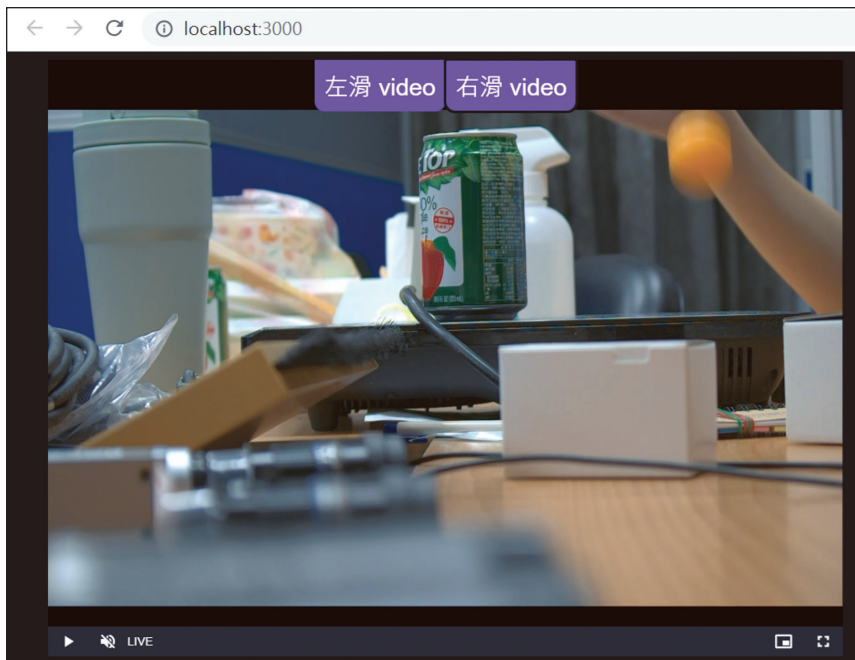


圖 14 Web Browser 呈現 FVV 影像串流 2

3. Daniel Berjón; Pablo Carballeira; Julián Cabrera; Carlos Carmona; Daniel Corregidor; César Díaz; Francisco Morán; Narciso, “FVV Live: Real-Time, Low-Cost, Free Viewpoint Video”, 2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW).
4. GenICam Standard, <https://www.emva.org/standards-technology/genicam/>
5. IEEE, “1588-2019 – IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems”, IEEE Std 1588-2019, 2019
6. Steve T. Watt, Shankar Achanta, Hamza Abubakari, and Eric Sagen, “Understanding and Applying Precision Time Protocol”, 14th Annual Clemson University Power Systems Conference, March 2015.
7. David G. Lowe , “Distinctive Image Features from Scale-Invariant Keypoints”, International Journal of Computer Vision 60(2), 91-110, 2004.
8. Adobe Corp., “RTMP: Real Time Messaging Protocol”, Adobe Flash Lite 4 Help. Adobe. Retrieved 29 December 2013.
9. T. Wiegand, G. J. Sullivan, G. Bjontegaard, A. Luthra, “Overview of the H.264/AVC video coding standard”, IEEE Transactions on Circuits and Systems for Video Technology, Volume: 13, Issue: 7, July 2003.
10. R. Pantos, Ed., and W. May, “HLS: HTTP Live Streaming Specification”, RFC-8216, ITU, 2017.
11. Hideaki Kimata , Masaki Kitahara , Kazuto Kamikura , Yoshiyuki Yashima , Toshiaki Fujii , and Masayuki Tanimoto, “System Design of Free Viewpoint Video Communication”, Proceedings of the Fourth International Conference on Computer and Information Technology (CIT’04), 2004.
12. Sabri Gurbuz; Masahiro Kawakita; Hiroshi Ando, “Color calibration for multi-camera imaging systems”, 2010 4th International Universal Communication Symposium.