

跨平台FHD即時視訊使用者介面的體實作與研究

郭祐任¹ 溫竣皓² 林映任³

¹美和科技大學資訊科技系 學士生

²美和科技大學資訊科技系 助理教授

³美和科技大學資訊科技系 學士生

¹ d40414104@go.meiho.edu.tw ² x00007351@meiho.edu.tw ³ d40514141@go.meiho.edu.tw

摘要

無線影像傳輸通常有著延遲上的問題，不夠即時的畫面會導致視訊互動不協調，甚至造成該應用無法穩定可靠的使用。為了改善此問題，我們將實作無線影像串流視訊軟體，然後以此為基礎進行相關的優化研究。我們實作重點在於進行無線串流視訊的應用軟體開發，在評估比較數項軟體編輯器（IDE）後選擇 Unity 遊戲引擎設計實作出友善的使用者介面，並結合開源的 VLC 媒體編解碼 API 與 RTSP 影音串流技術為核心，針對如何應用聲音影像串流的編解碼器做研究。

實作結果顯示我們的視訊軟體能夠提供符合即時影像使用需求的低延遲串流，並且我們還發現不同畫質、快取值都對串流延遲時間有顯著的影響。該研究的成果更於協助我們了解使用者介面設計、視訊編碼技術以及無線網路傳輸協議的應用，並為後續無線視訊通訊應用計畫完成先期的準備。

關鍵字： Unity, VLC, RTSP, 即時影像串流, 媒體編解碼 API

Abstract

Most of the wireless video applications suffer high latency problem. The problem prohibits the possibility to support real-time video interaction. To the worse, this can make the video application unusable. To solve this problem, we build a prototype of wireless video streaming video application for testing the various approaches to optimize latency. The focus of this research is to build a software application for real-time wireless video streaming. The development of software starts with a comparison of various Integrated Development Environment (IDE), and we choose Unity game engine for its capability to construct a friendly user interface. The software utilizes an open source VLC media codec API and RTSP streaming architecture. The development of this software helps us to understand the video codecs and wireless network transmission protocol.

The results show that our video software can provide low-latency streams that meet the needs of real-time video application. In addition, it appears that different video resolution and cache size have a significant impact on the latency. These findings help us to understand the user interface design, video coding API, and wireless network transmission protocol applications. Moreover, this is a preparation for a forthcoming wireless video surveillance application project.

Key Words: Unity, VLC, RTSP, real-time streaming, media codec API

¹美和科技大學資訊科技系學士生(聯絡地址：屏東縣內埔鄉屏光路 23 號，聯絡電話：(08)779-9821 轉 8570，E-mail: d40414104@go.meiho.edu.tw)。

²美和科技大學資訊科技系助理教授

³美和科技大學資訊科技系學士生

前言

即時視訊傳輸的延遲的問題，已經存在許久，為了解決此問題，我們將使用即時串流協定實作出無線影像串流視訊軟體。

本文主體分為研究準備、實作方法、實作結果、結論與討論這四個主要章節，在各個章節依序說明如何對符合研究目的工具進行優缺點的評估比較，接著解釋我們是如何將最合適的媒體編解碼的API[1]應用於本次軟體開發，然後是實作結果的展示說明與成果測量比較，最後在本研究的結尾進行相關的結論與討論。

實作準備

現在有許多影音編解碼的插件可供應用在無線影音播放，同時需要考慮的還有開發環境、語言、與執行平台也有不少可供軟體開發者使用。下面將對先今幾個知名支援無線即時影音編解碼的軟體插件(表1)與開發環境(表2)進行功能比較，從中選出最合適的且應用平台最廣泛的開發工具。

表 1：媒體編解碼 API 的選擇比較表

API 名稱	開發語言	費用	普遍性
VLC	C[2]	開源免費	高
UVLC	C#[3]	US\$400	中
UMP	C#	US\$45	低

表 2：開發環境、支援 API 與平台的選擇比較表

開發環境	API	費用	執行平台	普遍性
Unity[4]	有支援	開源免費	Almost all	高
Visual studio [5]	有支援	開源免費	Windows	中
Android studio[6]	有支援	開源免費	Android	低

分析表中的普遍性指的是使用的普及程度，結果顯示VCL API是普遍性較高，同時且是免費且開源，能允許我們後續的優化，相對是較合適的選擇。而Unity開發環境是近年來應用廣泛的程式開發工具，並且支援表一篩選後的API，並且可在廣泛平台上執行。所以評估結果的選擇是以Unity開發環境搭配VLC API作為本次實作研究的基礎。

實作方法

RTSP[7]即時串流協定(Real Time Streaming Protocol, RTSP)是一種網路應用協定，專為娛樂和通訊系統的使用，以控制串流媒體伺服器。

VLC[8]是一個開源的跨平台多媒體播放器，可以播放大多數的媒體檔案，以及DVD、音樂CD、VCD、和各種串流協定。

在這裡我們是以RAD[9]快速應用程式開發的方式，來進行快速的軟體開發，如圖1所示是我們進行快速應用程式開發的流程，在準備後，開始實

作，並在完成後進行測試，如過測試失敗將重新進行實作，直至測試成功，才算完成。

雖然說VLC是一個擁有多功能與跨平台的軟體，但我們只需要簡單的視訊串流的功能，因此我們選擇Unity作為我們的開發工具，結合VLC插件單獨取用此功能，並藉由友善的介面設計來簡化串流時的設定。

圖2所示是我們初期構思的使用者介面，為了未來我們無線視訊通訊於應用計畫，我們構思出此介面。

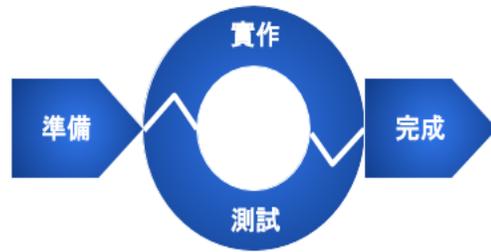


圖1 快速應用程式開發流程圖



圖2 介面構思

實作結果

下圖3-5是我的實作結果，我們成功實作出一個視訊串流軟體，並以RTSP預設的Port:554作為接收端口，接收串流影像，只要是以RTSP發送串流的影像，我們的視訊串流軟體均可接收。圖3為未開始串流的畫面，圖4為串流開始的畫面，圖5為串流時使用縮放功能的畫面。

在測量面我們進行延遲[10]測量的環境其Wi-Fi[11]規格為802.11n[12](廠牌:EDIMAX 型號:BR-6430ns,最快速度:300Mbps)，測量距離1公尺，快取設置250ms，實際測量方法是藉由Raspberry Pi[13] camera拍攝碼表畫面分別使用解析度FHD 1080p與HD 720p來進行視訊串流，並用手機同時錄製碼表與視訊串流軟體的畫面，拍攝完後，將碼表的時間減掉視訊串流畫面裡碼表的時間，就是我們所求的延遲。圖6所示為測量串延遲的實際畫面。

我們對最初實作出的第一版到現在的第三版進行延遲時間測量比較，如圖7所示，可以看到第一版延遲時間達1500ms，而在第二版我們將程式中的快取[15]值從1000ms下降到500ms，第三版再修改成250ms後其串流延遲時間也下降至479ms最低甚至到HD 720p的319ms。

圖8串流延遲測量結果所示，我們在FHD 1080p與HD 720p上個進行30次的延遲時間測量，結果平均值分別為479ms與319ms，由此測量結果可知畫質不同會影響到串流延遲時間。

國外文章[14]測量內容分別為480p Netbook、480p Laptop、HD 720p Netbook、HD 720p Laptop，由於我是使用MacBook來做測量，所以在這裡我們只看HD 720p Netbook的測量數據，在文顯示HD 720p Netbook測量結果的延遲平均值為194ms，比起我們的HD 720p的319ms來說相對來的低，由此可知我們的串流延遲時間仍有優化的空間，未來我們會持續優化來降低延遲時間。

另一方面在修改快取的過程中我發現，我們實作的串流軟體其快取值有一個平衡的值在，當快取低於平衡值將會造成畫面流暢度下降，因此我們最後選擇以250ms作為平衡值來測量，在測量期間此快取值在測量時都未造成畫面的流暢度下降。



圖5 實作結果串流畫面縮放

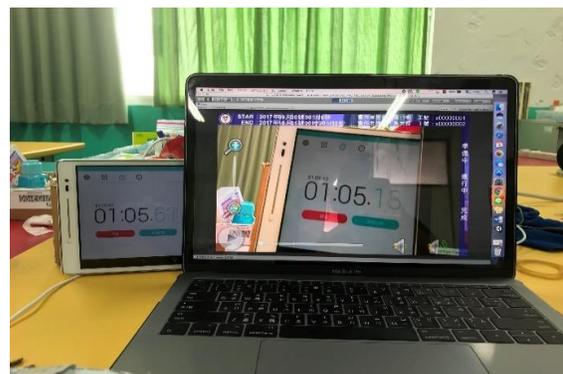


圖6 串流延遲測量畫面



圖3 實作結果起始畫面



圖4 實作結果串流畫面

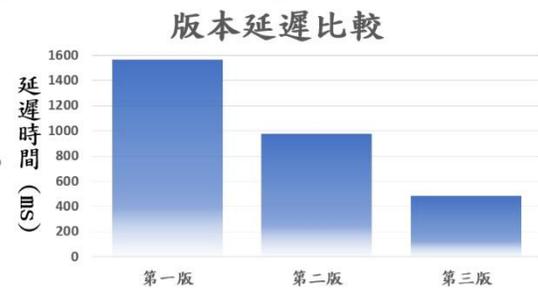


圖7 版本延遲比較長條圖

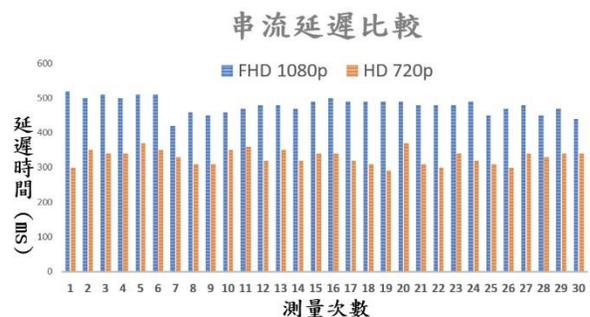


圖8 串流延遲比較長條圖

結論與討論

我們選擇以Unity作為本研究的開發工具並結合VLC媒體編碼器API，成功實作出一視訊串流軟體，並在FHD 1080p畫質下將串流延遲時間從一開始的1500ms下降至479ms，而在HD 720p畫質方面延遲也降到319ms，此數值仍然有改善的空間，但已符合使用條件。

現階段我們完成核心的串流功能，使用者介面設計、音量控制、畫面縮放也都已能正常使用，未來將會對步驟認證、資訊存留等功能作出完善外將持續優化使用者介面，並為後續無線視訊通訊應用計畫做出進一步的貢獻。

參考文獻

- [1] S.Browne, J.Dongarra, N.Garner, G.Ho, and P.Mucci, "A portable programming interface for performance evaluation on modern processors," *Int. J. High Perform. Comput. Appl.*, vol. 14, no. 3, pp. 189–204, 2000.
- [2] B. W.Kernighan and D. M.Ritchie, *The C programming language*. 2006.
- [3] A.Hejlsberg, S.Wiltamuth, and P.Golde, *The C# programming language*. Adobe Press, 2006.
- [4] J.Craighead, J.Burke, and R.Murphy, "Using the unity game engine to develop sarge: a case study," in *Proceedings of the 2008 Simulation Workshop at the International Conference on Intelligent Robots and Systems (IROS 2008)*, 2008.
- [5] S.Cook, G.Jones, S.Kent, and A. C.Wills, *Domain-specific development with visual studio dsl tools*. Pearson Education, 2007.
- [6] B. C.Zapata, *Android studio application development*. Packt Publishing Ltd, 2013.
- [7] H.Schulzrinne, "Real time streaming protocol (RTSP)," 1998.
- [8] L. A. N.Video, "VLC media player." 2013.
- [9] J.Martin, *Rapid application development*. Macmillan Publishing Co., Inc., 1991.
- [10] Y.Liu, Y.Guo, and C.Liang, "A survey on peer-to-peer video streaming systems," *Peer-to-peer Netw. Appl.*, vol. 1, no. 1, pp. 18–28, 2008.
- [11] J.-S.Lee, Y.-W.Su, and C.-C.Shen, "A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi," in *Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE, 2007*, pp. 46–51.
- [12] Y.Xiao, "IEEE 802.11 n: enhancements for higher throughput in wireless LANs," *IEEE Wirel. Commun.*, vol. 12, no. 6, pp. 82–91, 2005.
- [13] E.Upton and G.Halfacree, *Raspberry Pi user guide*. John Wiley & Sons, 2014.

- [14] Pilotgary, "Raspberry Pi Camera Latency Testing – Part 2," *wordpress*, 2014. [Online]. Available: <https://sparkyflight.wordpress.com/2014/02/22/raspberry-pi-camera-latency-testing-part-2/>.
- [15] A. J.Smith, "Cache memories," *ACM Comput. Surv.*, vol. 14, no. 3, pp. 473–530, 1982.